Leveraging Domain Knowledge for Self-Supervision in Scalable Robot Learning

Dan Barnes

April 2020

Leveraging Domain Knowledge for Self-Supervision in Scalable Robot Learning

Dan Barnes

Applied AI Lab
Oxford Robotics Institute
Department of Engineering Science
University of Oxford

Dan Barnes

Leveraging Domain Knowledge for Self-Supervision in Scalable Robot Learning

Supervisor: Professor Ingmar Posner
Examiners: Professor Paul Newman and Professor Andrew Davison

University of Oxford
Applied AI Lab
Oxford Robotics Institute
Department of Engineering Science
Parks Road
Oxford
OX1 3PJ
UK

Declaration

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. I declare that this thesis is entirely my own work, and except where otherwise stated, describes my own research.

Daniel H Barnes

Dan Barnes
Oxford, UK
April 2020

Acknowledgements

I first would like to thank my supervisor Professor Ingmar Posner, with whom I have worked on various projects over the last seven years. His guidance and patience over that time has undoubtedly helped shape my research career from that early stage and fostered my interest in robot learning. I am extremely grateful to have been able to receive his advice during my time at Oxford.

I would like to thank Dr Will Maddern, who also provided invaluable influence and support during my formulative years as a researcher. In addition to guidance on my studies, stimulating my interest in deploying robots into the real world is undoubtedly one of the most important lessons I have learned over the past four years.

I would like to thank everyone at the Oxford Robotics Institute (ORI): from the students, postdocs and senior researchers for academic discussions and constructive criticism, to the engineers and administrative staff for helping with all the difficult things that make real-world robotics possible. I look forward to hearing about all the amazing things that the ORI gets up to in the future.

I would like to thank my family Ruth, Steve and Spence, for their support during my studies and whatever comes next. In particular, I would like to thank Heli for being at my side over the course of my DPhil. There is no way I can express here my gratitude or put into words the true impact she has had. Suffice to say I consider myself extremely lucky and look forward to our adventures together in the future.

I would like to thank my examiners, Professor Paul Newman and Professor Andrew Davison, for taking time out of their busy schedules to provide valuable feedback and discussion on my research.

Finally, I would like to gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), through the Doctoral Training Award (DTA), in making my studies possible.

Abstract

Deep learning in robotics has a data problem.

Over the past decade, deep learning has revolutionised the application of robotics in the real world. From self-driving cars to drones and warehouse applications, deep learning has become an integral tool in deploying robots into complex domains. Despite many advances in training efficiency, operating performance, and inference speeds, there is one key problem that remains in utilising deep models in real-world applications; data efficiency due to the vast quantities of labelled data required for training.

Despite the clear successes that have emerged by using substantial manually annotated datasets to satisfy this need, human labelling, such as drawing bounding boxes around vehicles, is labour intensive, financially expensive and mentally unfulfilling. These datasets are extremely uneconomical and scale poorly to new geographic locations, weather conditions and robotic tasks. Even when deployed in well-labelled domains, trained models may fail when encountering novel data and have no means for self-improvement.

In this thesis we investigate the paradigm of *self-supervised learning* as a solution to these shortcomings, in which high fidelity training data is generated completely automatically without any human supervision. In particular, our goal is to utilise domain knowledge in expert systems for curating the data necessary to train deep models.

We apply this methodology on real-world tasks with clear commercial applications, such as predicting drivable paths in camera imagery for mapless urban navigation, learning ephemerality masks for robust monocular visual odometry, and learning keypoints for radar odometry and localisation. Not only do we meet or exceed state-of-the-art performance in these crucial tasks, but we do so with no human supervision. In sharing our approaches and ideas from the past four years, we lay the foundations for more efficient and scalable robot learning in the future.

Contents

Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
ADAS	Advanced Driver-Assistance System
ANN	Artificial Neural Network
BASD	Binary Annular Statistics Descriptor
CFAR	Constant False Alarm Rate
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DNN	Deep Neural Network
FAST	Features From Accelerated Segment Test
FFT	Fast Fourier Transform
FMCW	Frequency Modulated Continuous Wave
FMT	Fourier Mellin Transform
GPU	Graphics Processing Unit
lidar	Light Detection And Ranging
ORB	Oriented Rotated BRIEF
radar	Radio Detection And Ranging
RO	Radar Odometry
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localisation And Mapping
SURF	Speeded Up Robust Features
VO	Visual Odometry

Glossary

Egomotion	Motion of a sensor or robot in its environment.
Ephemeral	Lasting for a temporally short time.
Odometry	Estimate of egomotion.
Path Planning	Estimating the optimal path for a robot between two points.
Self-Supervised	Training regime where raw data is automatically labelled.

Introduction 1

1.1 Motivation

Deep Neural Networks (DNNs) have become the de facto method for many tasks in real-world robotic applications such as perception, planning, prediction and tracking. This is extremely noticeable in autonomous vehicles, where DNNs have been integrated deeply across all levels of the software stack including lower level tasks such as vehicle detection [1], planning with intermediate representations [2] and controlling vehicles directly from sensory inputs [3].

Despite the clear successes integrating and using DNNs in robotic applications, there are still numerous pitfalls to their use. No constraint is more apparent than the colossal amount of data required to learn meaningful representations which is typically addressed in practice by expensive, albeit effective, manually labelled datasets. Modern publicly available autonomous vehicle datasets contain up to hundreds of thousands of annotated sensor frames, including millions of individually labelled objects [4]–[8]; clearly annotation at such scales is infeasible for most robot learning practitioners.

The questions we must ask ourselves are: Is this a constraint we must accept? How much longer is this approach viable? What can we implement to alleviate this data limitation? Practical solutions to these issues could either look at reducing data requirements of training DNNs or by reducing the burden to create labelled training data. This thesis focuses on the latter.

We start with a general belief that continually collecting raw sensor data from a robot system is considered free when compared to the alternative of human annotation. Given this corpus of ever growing sensor data, continually encountering new lighting, weather and traffic conditions, we aim to employ prior domain knowledge to automatically label data in a semantically meaningful way, with no human supervision. A DNN trained on the resulting data can both run significantly faster than the originating labelling method, and be deployed within reason to new locations and conditions not previously encountered. We refer to this automatic data generation and DNN learning paradigm as *self-supervised* *learning*. By intelligently integrating these DNNs in existing systems or novel architectures, we can significantly improve performance in real-world robotics tasks.

1.2 Contributions

This thesis explores methods in which self-supervised learning can be used to reduce manual labelling effort, with the goal of improving the scalability of deep learning applications in robot learning. The solutions we present tackle fundamental real-world robotics problems with obvious commercial applications. Additionally, we surmise that these self-supervised approaches and techniques can readily be extended and applied to other tasks in the future.

The principal contributions of this thesis are in devising and deploying approaches for effective self-supervision in scalable robot learning. First, we conceive of high-fidelity automatic data annotation systems applicable to a diverse set of downstream robotics tasks. By using only raw sensor data as inputs and domain knowledge in the form of expert systems as the labelling mechanism, we collect and label unlimited sensor data from novel traffic, weather and lighting conditions with no human supervision. Second, we exploit this automatically labelled data in a self-supervised learning paradigm, to train DNNs which would normally require vast quantities of manual annotations. Third, we integrate these DNNs into robotic systems to significantly improve performance in several core real-world applications.

In applying this methodology, we present these additional significant contributions to specific robotics tasks:

- Learning to segment drivable paths in monocular camera imagery which follow the rules of the road in complex urban environments.

- Integrating DNNs within existing sparse and dense visual odometry (VO) systems to identify stable structure and enable robust monocular VO in cluttered urban scenes.

- Developing novel dense and sparse architectures to unify learning methods with classical approaches for state-of-the-art radar odometry (RO) and localisation.

- Releasing the largest urban radar dataset of its kind to the research community to spur academic interest in a promising yet underutilised modality.

1.3 Publications

The following publications implement the self-supervised learning methodology presented in Section 1.1 and form the core findings of this thesis:

- Find Your Own Way: Weakly-Supervised Segmentation of Path Proposals for Urban Autonomy
  Dan Barnes, Will Maddern and Ingmar Posner
  IEEE International Conference on Robotics and Automation (ICRA), 2017

- Driven to Distraction: Self-Supervised Distractor Learning for Robust Monocular Visual Odometry in Urban Environments
  Dan Barnes, Will Maddern, Geoffrey Pascoe and Ingmar Posner
  IEEE International Conference on Robotics and Automation (ICRA), 2018

- Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information
  Dan Barnes, Rob Weston and Ingmar Posner
  Conference on Robot Learning (CoRL), 2019

- Under the Radar: Learning to Predict Robust Keypoints for Odometry Estimation and Metric Localisation in Radar
  Dan Barnes and Ingmar Posner
  IEEE International Conference on Robotics and Automation (ICRA), 2020

In addition to theoretical contributions, considerable effort and resources have been invested to collect data suited for evaluating our hypothesis. The resulting dataset forms a key component to our narrative as detailed in the following publication, as well as a significant contribution to the academic community.

- The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset
  Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman and Ingmar Posner
  IEEE International Conference on Robotics and Automation (ICRA), 2020

Further information on all projects including presentations and videos can be found at: `https://dbarnes.github.io`

1.4 Thesis Outline

In this section we provide a brief outline of the thesis and our work in applying self-supervision to scalable robot learning.

1.4.1 Drivable Path Segmentation

Scene understanding and path planning, identifying where a robot can safely travel, is a critical component for safe navigation in both robot-only and shared settings. In the context of autonomous vehicles, given the structured nature of urban driving, all vehicles must follow the rules of the road. A typical Advanced Driver-Assistance System (ADAS) relies on simplified road models extracted from sensor data to plan in. However, without clear or consistent lane markings in ever more complex environments, this is becoming increasingly challenging and an alternative approach is required. In Chapter 3 we present a self-supervised approach to automatically segment proposed drivable paths in camera imagery without requiring any manual annotations. We achieve this by using human driving demonstrations, along with understanding of the scene geometry and other expert systems, to automatically label drivable paths and obstacles in camera imagery. We evaluate our method on two diverse datasets [7], [9] showing reliable path and obstacle segmentations in a wide variety of environments. To the best of our knowledge, ours was the first system to achieve such outputs in a self-supervised way without human involvement.

1.4.2 Egomotion with Learnt Ephemerality

Prior work has demonstrated the benefits of identifying distractor objects for improving VO estimation in highly dynamic, heavily cluttered urban environments [10]. For example sizeable moving agents in the scene, such as buses, can occupy the majority of a camera's field of view; hence accounting for this motion is essential for correct egomotion estimation and safe operation. A key limitation of the prior work is the inability to operate outside of previously mapped areas, as a Three-Dimensional (3D) map is used to estimate which objects are ephemeral, typically moving vehicles, and which objects belong to the underlying static scene, such as buildings and roads. In Chapter 4 we present a self-supervised approach for ignoring distractors in camera images. We do this by leveraging offline multi-session mapping to automatically label per-pixel ephemerality masks and depth maps. We show that at run-time, by using the predicted ephemerality masks to weight sparse and dense VO matches, we can recover the correct egomotion even with significant independently moving distractor objects. Furthermore, we show that by training our model to predict absolute depth from single image inputs, we are able to yield robust metric-scale VO using only a monocular camera in urban environments.

1.4.3 Large-Scale Urban Radar Dataset Collection

Safety critical applications such as autonomous vehicles employ various sensor modalities, such as vison, Radio Detection and Ranging (radar) and Light Detection and Ranging (lidar), for redundancy to ensure reliable operation in challenging conditions. A key limitation of both vision and lidar is vulnerability to common environmental conditions such as: low-light, sun glare, fog, rain and snow. Frequency Modulated Continuous Wave (FMCW) radar shows promise in tackling some of these shortcomings due to its wavelength. However, there are few publicly available datasets to evaluate this hypothesis. To this end, in Chapter 5 we present a radar extension to the Oxford RobotCar Dataset [9]. With a full sensor suite, we collected data over 280 km of urban city driving encompassing a variety of weather, traffic, and lighting conditions. Furthermore, ground truth trajectories were optimised completely automatically, enabling the self-supervised work discussed in Chapters 6 and 7. We share the entirety of the dataset with the community to foster and promote development in this interesting modality.

1.4.4 Distraction-Free Radar Odometry Estimation

Highly accurate RO is possible using FMCW radar [11]. However, there is still a noticeable gap to vision and lidar odometry performance [12], [13]. In Chapter 4 we demonstrate improving visual egomotion estimation by learning to mask distractor objects. Although this proved to be a successful approach, these masks were not necessarily optimal for pose estimation. The masks were instead trained on the proxy task of segmenting temporally stable objects such as buildings and walls. In Chapter 6 we improve on this approach by learning to mask for the *specific* motion estimation algorithm, rather than on some other proxy task, and in the process produce the state-of-the-art in RO. Our primary contribution is to embed a differentiable pose estimator within our architecture, thereby enabling us to learn radar masks optimised for the specific pose estimator. Due to ubiquitous sensing artefacts in radar scans and moving vehicles in urban environments, our architecture naturally learns to retain only the static structure in the scene with no explicit mask supervision. We additionally propose a strategy for producing calibrated pose uncertainties from our architecture, crucial for real-world robotics applications.

1.4.5 Self-Supervised Radar Keypoint Learning

In Chapter 6 we present a method to achieve state-of-the-art RO. Although trained on pose error alone, the formulation naturally learns to retain only the static structure in radar scans which can then be used for other applications such as dense mapping. Conversely, sparse odometry methods typically use keypoints that can be more readily used for other applications, such as place recognition and sparse optical flow. The prior state-of-the-art in sparse RO [11] devises hand-crafted keypoints to be used in motion estimation. In Chapter 7 we improve on these approaches by learning a keypoint detector for radar which is optimised for pose estimation. Our main contribution, in a similar fashion to Chapter 6, is to embed a differentiable keypoint-based pose estimator within our architecture; enabling us to learn keypoints reusable for other tasks without imposing any prior human assumptions. We show that the keypoint-based approach produces accurate motion estimates, outperforming the prior state-of-the-art in sparse RO. Finally, as the keypoints are reliably detected on the static structure they can be reused for place recognition, and we utilise this to present a full real-time mapping and localisation framework in radar.

1.5 Impact

The main contribution of this thesis is in developing system-level supervision for DNNs in real-world urban robotics. However, in addition to the academic benefits detailed in the following chapters, there are clearly commercial applications both for the scalable learning regimes and developed solutions. To that end, each of the projects in Chapters 3 to 7 are either licensed, or are undergoing licensing, by Oxford University Innovation for commercialisation. Furthermore, the works detailed in both Chapters 3 and 4 are undergoing patent approval and have been licensed by a commercial entity.

Background

2

The detailed contributions of this thesis are provided in Chapters 3 to 7. Each represents a publication with a targeted study of the literature which is not repeated here. This chapter instead aims to provide a broad overview of the overarching themes related to the applications we cover. The concepts and ideas presented in the following chapters have applicability across different sensor modalities and robotic platforms. Nevertheless, as the primary data collection and evaluation platform in this thesis, we frame our background discussion around urban mobile robotics and in particular autonomous vehicles.

2.1 Autonomous Vehicles

Urban driving is an extremely complex task even for humans to execute. Between predicting the intentions of other agents in the scene to controlling the vehicle itself, there is a lot for a driver to process. Poor weather and lighting conditions and high-level route planning all add cognitive strain. Although particularly demanding in autonomous driving, these key challenges are ubiquitous across robotics and can be summarised as: Where am I? What's around me? and What should I do next? For this reason, and the complexity in autonomous vehicle design and deployment, this application is a well-suited proxy for utilising robotics in other domains.

The Oxford RobotCar[1] platform, a modified Nissan LEAF, is used as the primary data collection and evaluation vehicle in this thesis. The vehicle sensor suite is comparable to commercial autonomous vehicles including position sensors (GPS/INS), range sensors (3D/2D lidar and radar) and image sensors (stereo/monocular) as well as using high accuracy calibration and time synchronisation tools detailed further in [9] and Chapter 5. Typical sensor data can be seen in Figure 2.1, with each sensor offering their own strengths and weaknesses; for example, vision is useful for detecting traffic light state but struggles with fog, whilst radar offer consistent returns but suffers significant sensing artefacts. When the different modalities are used jointly for sensor redundancy, the vehicle is privy to richer

[1] `https://ori.ox.ac.uk/robotcar-overview` - with additional 3D lidars and radar

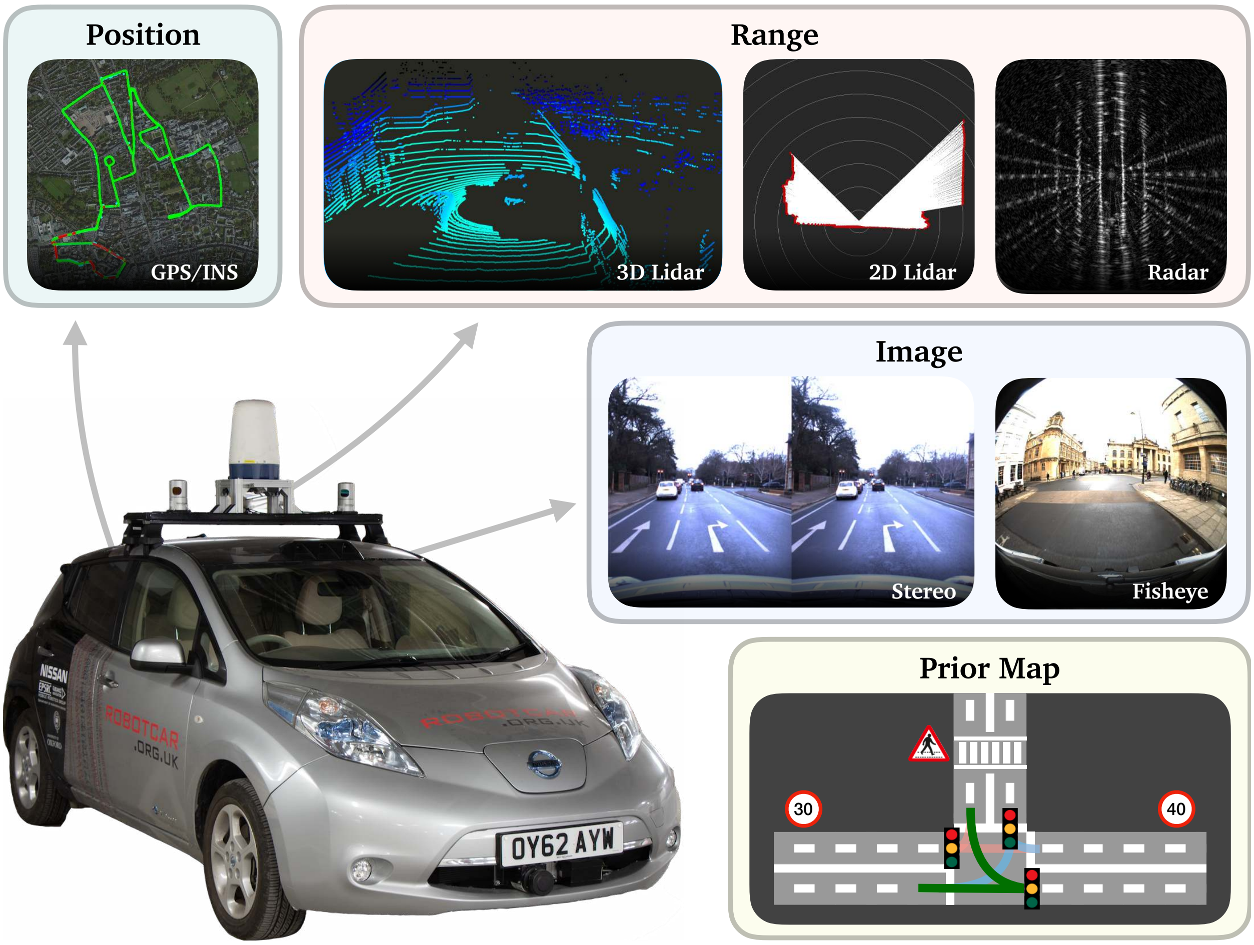


Figure 2.1: Example sensor data on the Oxford RobotCar[1], a modified Nissan LEAF used as the primary data collection and evaluation vehicle in this thesis. Although each modality has its own strengths and weaknesses, when used jointly they enable improved performance for safely navigating in the real world. In addition to online sensors, the car may have access to prior maps which can include detailed 3D information of the environment such as traffic lights, road layouts and lane connectivity, but are extremely expensive to create and challenging to maintain.

information and can navigate safely in the real world. The wealth of data on offer enables remarkable opportunities for analysing and extracting semantics from the world. This avenue is exploited in the coming chapters for automatic data annotation.

Given this wealth of sensor and map data, a simplified systems diagram for an urban autonomous vehicle is shown in Figure 2.2, where the green arrows represent data flow. The data is processed in order to understand both where the vehicle is and what is located nearby, before planning and executing the most optimal path to take. Notably, unlike lower cost robot applications, autonomous vehicles typically rely on extremely expensive inputs (both financially and in terms of human supervision) such as lidar [1] and prior maps [14]. Despite the high accuracy and semantically meaningful 3D data these sources can

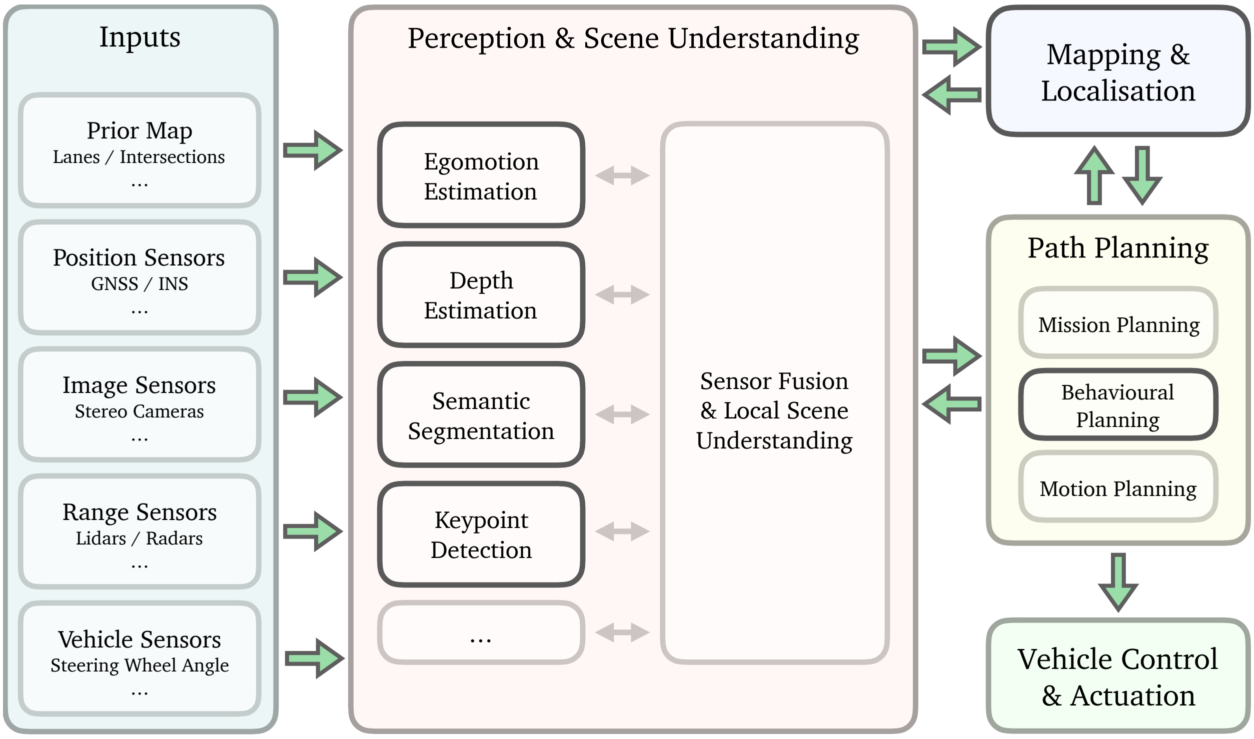


Figure 2.2: A simplified systems diagram for an urban autonomous vehicle where green arrows represent data flow. The vehicle receives input data (left) both from sensors mounted on the vehicle and an optional prior map. All the raw sensor and optional map data are fused together and processed by the perception and localisation modules (middle), so that the vehicle knows what is nearby and where it is located. Finally, with all this knowledge and a high level plan, the vehicle can safely plan and execute a trajectory in the real world (right). In this thesis we investigate a subset of these tasks as highlighted.

provide, they limit wider-scale deployment due to their cost and excessive labelling time requirements. The works presented in this thesis reduce some of these constraints through the use of self-supervised learning.

To validate our self-supervised methodology, we target a subset of the core tasks from Figure 2.2 as highlighted, which include: behavioural planning in monocular camera imagery in Chapter 3, semantic segmentation in the form of ephemerality mask prediction in vision and radar in Chapters 4 and 6, depth estimation in monocular camera imagery in Chapter 4, egomotion estimation in vision in Chapter 4 and radar in Chapters 6 and 7, and mapping and localisation in radar in Chapter 7.

In reality, these clear subsystem distinctions become blurred as individual components become more heterogeneous in nature, sharing computation between tasks. This is primarily as a consequence of joint inference proving to be more performant for certain tasks, for example when inferring both semantic segmentation and object detection. Nevertheless, this chapter provides background on two specific tasks which broadly cover the application-specific contributions presented in this thesis: path planning and egomotion estimation. In providing this complementary background we facilitate a smooth transition to the focused literature reviews in Chapters 3 to 7.

2.2 Path Planning

Path planning is extremely challenging in urban scenarios and requires balancing many constraints. An optimal path must traverse free space on the road, in the vehicle's own lane and in such a way as to convey future intentions to other agents in the environment, all whilst attempting to follow a higher-level plan. As such, path planning for autonomous vehicles can be conceptually thought of as three sequential tasks: *mission planning* (balancing high-level objectives such as determining the topological route to a final destination), *behavioural planning* (creating a local plan following the rules of the road and other agents in the scene), and *motion planning* (generating the appropriate vehicle commands).

If we presume mission planning is accounted for, the autonomous driving literature can be separated into three main paradigms [15]: mediated perception approaches that semantically parse a scene to make a driving decision, behaviour reflex approaches that directly map an input image to a driving action, and direct perception approaches that fall in between. Each of these has seen vast learning-based developments in recent years.

Inferring scene semantics in mediated perception approaches, such as lane boundaries, enjoyed many successes prior to learning-based methods due to the inherent on-road visual structure [16]–[18]. These methods typically employ shape, colour and brightness operations such as template matching, colour space mappings and thresholding. In well-marked, well-lit urban environments these perform well, yet generalise poorly under drastic changes in local lighting conditions or in areas devoid of road markings. DNN models have slowly replaced these methods to address some of these shortcomings such as for lane boundary estimation [19] and semantic segmentation [20], [21]. As an optional pre-planning step, per-frame semantics can be accumulated into a local map from which to generate a vehicle plan.

Conversely, robot learning has enabled behaviour reflex approaches where a direct mapping is learnt from sensor input to vehicle commands. This idea dates back to 1989 [22] where a 3-layer DNN named ALVINN (Autonomous Land Vehicle In a Neural Network) was trained for the task of road following by learning to regress to human steering commands. This idea of imitation learning has persisted to this day, and in 2016 [23] Nvidia demonstrated real-world driving using this approach in complex urban scenes amongst other vehicles. Recent approaches have also shown the ability to transfer trained models from simulation to real-world environments [24], as well as accept higher level plan conditioning [3]. However, arguments against reflex approaches cite both the large amounts of data required and the lack of generalisation/interpretability, a motivating factor in our work.

Chapter 3 presents an approach with a different narrative to these three paradigms. An imitation learning approach is used to replicate human expert driver trajectories which follow the rules of the road, but rather than regressing to steering commands, our method predicts vehicle trajectories in image space. As a consequence, this method retains the benefits of reflex approaches with additional interpretability, as well as providing a useful semantic representation for other downstream tasks, all without any manual annotation.

2.3 Egomotion Estimation

Egomotion estimation is a core task in deploying robots in the real world and has been an active field of research for decades. In the context of autonomous vehicles, without accurate egomotion, plans can not be safely executed no matter how optimal they may be. Approaches for estimating egomotion differ drastically based on the sensor modality (or combination of sensor modalities) used. In this thesis we investigate egomotion estimation in vision and radar limited only to processing sequential frames, and give an overview of these modalities in the following sections.

Broadly speaking, odometry approaches in vision and radar can be categorised as sparse/dense (describing how much of the input data is used) and indirect/direct (describing how the input data is used). Sparse odometry approaches use only a subset of the input data at certain locations whereas dense methods use all of the input data. Direct methods use sensor returns directly for estimating motion, such as pixel intensities in vision and power returns in radar. In contrast, indirect methods extract descriptors sparsely/densely from the data, usually computed from a local neighbourhood. Needless to say, these distinct categories are a simplification, as some hybrid methods may be described as semi-dense [25] or semi-direct [26], and the separation between these terms is somewhat continuous.

2.3.1 Vision

Egomotion estimation in vision, or visual odometry (VO), is one of the most researched tasks in computer vision. The natural world is abundant with texture and colour, making vision a well-suited modality for egomotion estimation. However, there are numerous potential challenges with using standard commercial cameras for VO in the real world. For example, challenges in automotive applications could include: poor visibility at night, sun glare at dawn/dusk or from vehicle headlights, vulnerability to weather conditions such as rain, snow and fog and no explicit measure of depth. Furthermore, common camera setups provide a limited field of view which is highly vulnerable to occlusion from large independently moving objects common to urban settings, such as buses, which drastically degrade VO performance.

A common approach to VO is indirect, sparse and revolves around descriptor matching. First, sparse interest points are extracted from an input image using feature detectors and descriptors such as Scale-Invariant Feature Transform (SIFT) [27], Speeded Up Robust Features (SURF) [28], Features from Accelerated Segment Test (FAST) [29] and Oriented Rotated BRIEF (ORB) [30]. Second, feature matching presents the most likely feature associations between frames using the individual interest point descriptors. Finally the matches are supplied to a robust motion estimator, for example using RANSAC [31], to estimate the motion between frames. This approach, although simple, can yield state-of-the-art results [30], [32].

Direct VO methods claim to be more robust to drastic viewpoint changes and motion blur as they are usually denser than indirect methods, hence calculating motion from a larger set of input pixels, and take advantage of other geometric structure in the scene such as lines and curves. A large majority of direct VO algorithms are semi-dense and select pixels in the camera frame to use for optimisation, for example regions with non-negligible intensity gradients [25], [33]. Various error metrics can then be used to optimise the motion estimate given this subset of sensor data such as: Photometric Error [34], [35], Mutual Information [36] and Normalised Information Distance (NID) [37]. Although direct methods have the potential to be vastly more computationally intensive depending on density, real-time Central Processing Unit (CPU) approaches [25], [33] exist and the highly parallelisable formulations can readily be implemented on Graphics Processing Units (GPUs).

Whereas most contributions in the literature focus on a subset of VO approaches, Chapter 4 presents a method for improving any VO system by masking out distractor objects in camera imagery without any human annotation. In doing so, poor regions for motion estimation, such as moving vehicles, are heavily downweighted and instead only the stable static structure in the scene, such as walls and buildings, is used. Additionally, the proposed method is completely model free and learns to ignore other sensor artefacts which degrade VO performance, such as water droplets and sun glare, all from raw sensor data alone. Finally, by learning to predict absolute depth from monocular camera imagery in a self-supervised manner, we present robust sparse and dense monocular VO implementations for distractor free egomotion estimation in urban environments.

2.3.2 Radar

Egomotion estimation in radar, or radar odometry (RO), is an exciting area of research in a modality promising a hardware-based solution to some of the shortcomings of other sensors. Radar data can vary greatly between sensors, from sparse to dense returns with/without Doppler information, to varying range and angular resolutions. In automotive applications, radar is often used as a core modality for adaptive cruise control and ADAS to measure the position and velocities of nearby vehicles.

In this thesis we investigate RO with dense FMCW scanning radar data without Doppler information around 77Ghz, the band commonly used for long-range automotive applications. Notably, this class of radar holds promise in addressing some of the challenges in vision due to the wavelength used, such as visibility in low light and challenging weather conditions. The publications in Chapters 5 to 7 present further in-depth discussions and visualisations of the radar data used which we do not reproduce here.

Indirect methods for RO can follow a similar regime to VO and indeed visual features have been directly applied to radar, such as SIFT [38] in maritime settings and FAST [39] in urban settings. Needless to say, radar data has vastly different sensor characteristics to camera data and so directly applying visual features may not be the most prudent approach. In contrast, radar-specific features such as Constant False Alarm Rate (CFAR) filtering [40] or grid-map features such as Binary Annular Statistics Descriptor (BASD) [41] have been used to mitigate some of these factors, for example by modelling the radar characteristics more closely. More recently the work of Cen et al. [11] finds correspondences between point features extracted from raw scans using a shape similarity metric at close to real-time, showing the viability of accurate RO in urban environments.

Nevertheless, due to the complex interactions between radar sensors and the local environment, predicting keypoint locations and descriptors with manually tuned algorithms and heuristics remains a challenging task. For example, visual feature detectors applied to radar are prone to false positives [11] and hence degrade RO performance. Chapter 7 provides a solution with an approach that imposes no human assumptions on what makes a suitable keypoint, instead learning a keypoint detector in a self-supervised manner from raw radar data alone. The learned keypoints are localised to the static structure in the scene (ignoring prevalent sensing artefacts), facilitating robust matching between frames and a new state-of-the-art in sparse RO. Furthermore, by additionally reusing the learned keypoints for place recognition, we present a real-time mapping and localisation system in radar.

Due to the aforementioned challenges in manually designed keypoint extraction in radar, direct, landmark-free approaches provide an attractive alternative. Rouveure et al. [42] use a 3D cross-correlation formulation with ground-based radar for Simultaneous Localisation and Mapping (SLAM) over multi-km trajectories showing that reliable direct matching is viable. Alternatively Checchin et al. [43] apply the Fourier Mellin Transform (FMT) in a similar experimental setup as an efficient way of computing the rigid transformation between consecutive scans.

Despite the success of these approaches they still struggle with the heterogeneous sensing artefacts present when using radar in urban settings. Chapter 6 proposes a method to mitigate this limitation by learning to mask out the adverse sensing artefacts using raw radar data alone. The learned dense masks suppress sensing artefacts, such as speckle noise, saturations and ghost reflections, whilst preserving the static structure in the scene optimal for dense RO methods (as well as other downstream tasks such as radar-based mapping). By doing so we present a fast and robust state-of-the-art RO system utilising the advantages of both learning and non-learning methods, all without any human supervision.

2.4 Neural Networks in Robot Learning

This thesis centers around the scalable and efficient training of DNNs in robot learning and their use in core urban mobile robotics tasks. Despite the widespread use of DNNs today, practical deployment of neural networks has been a fairly recent development, largely due to computational requirements. Conversely, academic interest spans decades to 1943 [44] when McCulloch et al. presented a simple model of how neurons function using electrical circuits, the building block of neural networks. Almost two decades later Window and Hoff developed the first Artificial Neural Networks (ANNs) deployed to a real-world problem [45], eliminating echoes on phone lines, and are still in service today.

By adding multiple hidden layers of neurons in DNNs and vision-inspired convolution layers in Convolutional Neural Networks (CNNs), we form the types of ANNs used today [46]. However, it was not until 1998 that the first of what can be considered a modern day DNN application occurred. LeCun et al. [47] used backpropagation to learn the coefficients of a CNN classifier for hand-written digits directly from images. The approach was fully automatic and gave better model parameters than manual design, reinvigorating the interest in CNNs for computer vision applications. Since then, DNNs and CNNs have largely taken the same form, albeit with numerous advances in model design varying from differentiable sampling kernels [48] to Long Short-Term Memory units (LSTMs) [49] as well as computational efficiency gains by utilising GPU hardware [50]–[52].

Spurred on by these discoveries in the two decades since, applied uses of DNNs have skyrocketed and applications can now be found ubiquitously in society from neural machine translation [53] and speech synthesis [54] to medical applications [55]. In robotics, and specifically autonomous vehicles, DNNs have been integrated right from lower level tasks such as vehicle detection [1], through planning with intermediate representations [2] and all the way to controlling vehicles directly from sensory inputs [3], [23].

These learning methods offer an alternative to more classical algorithm design, and often reach state-of-the-art performance in their respective tasks. Despite this, it is worth noting that there are many cases where DNNs are less favourable in practice, for example due to their lack of interpretability. As performance demands in the real world have increased, so have DNN model complexity and with it the huge amount of labelled data needed to train models for safe deployment [56]. While there are some large datasets available to the public in the context of autonomous vehicles [4]–[9], they are extremely expensive to create, are largely system/application specific and do not necessarily scale well to new environments or conditions.

A simplified process of DNN design and development in robotics applications is shown in Figure 2.3, which consists of three main stages: data curation (data collection and annotation), model design and training, and model deployment. During design and deployment a model's performance will continually be monitored to ensure safe operation. If at any point performance is insufficient, we are left with two options: changes to the model design and training, or collecting/generating more annotated data (which is often seen as an attractive choice [57]). Although raw data collection itself is relatively cheap and easy (and is often a side effect of other autonomous operations), the corresponding annotation can be extremely time consuming and expensive. For example, a common dataset used for DNN pretraining [56] consists of over a million images with class labels, each of which has been manually verified; at 2 images/sec [58] this is clearly infeasible for an individual. Even with highly streamlined commercial services, the financial cost limits most practical applications. For example semantic segmentation labelling costs over $5 per image [59] which hardly scales to tens or hundreds of thousands of images. Conversely, if there was a way to automate annotation with an expert system, we can significantly speed up this development cycle by requiring no manual supervision.

As discussed in Chapter 1, automatic data annotation using expert systems for robot learning is the core contribution of this thesis. It is this self-supervised learning paradigm that inspires us to address some of the shortcomings in deploying DNNs in real-world tasks.

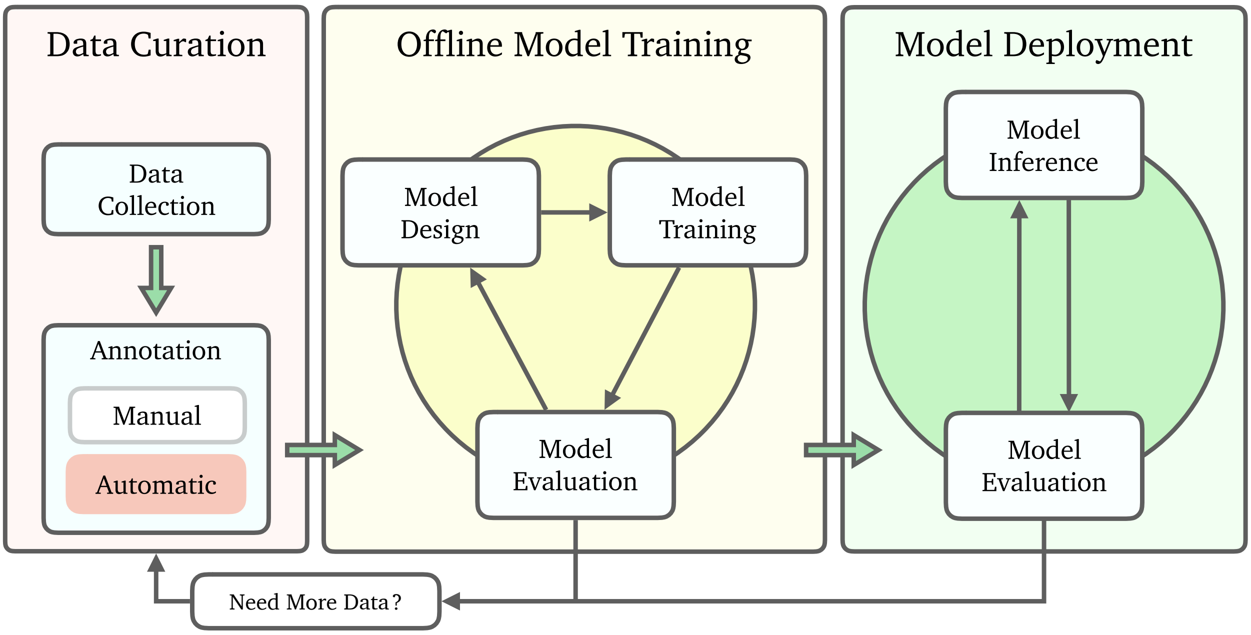


Figure 2.3: A simplified systems diagram for training DNNs in self-supervised robot learning. For a given task we first collect raw sensor data (left) and annotations to use as examples for training. Annotations can be created manually, such as labelling bounding boxes, but this can be extremely time consuming and expensive. Alternately, as is the focus of this thesis, we can use expert systems to annotate automatically, labelling a practically unlimited amount of data. With a now annotated dataset, we iterate though model design, training and evaluation (middle) until a model performs sufficiently for deployment (right). Whilst at any point automatic annotation methods can generate new data with zero human effort, for example if deployed to a new geographic location, manual annotation adds significant cost and scalability limitations to this learning cycle.

To truly scale robot learning in the future, we need expert systems that accurately and automatically annotate the data required to train and deploy DNNs in the real world.

3 Find Your Own Way: Weakly-Supervised Segmentation of Path Proposals for Urban Autonomy

The first application of self-supervised learning in this thesis concerns behavioural path planning, a fundamental task in autonomous driving. Knowing how to safely navigate in uncertain urban environments is an extremely challenging task – even for humans. As detailed in Sections 2.1 and 2.2 there are many factors for a driver to consider, from perceiving the local road layout to predicting the intentions of other vehicles.

Commercial driver assistance systems typically depend on visual recognition of road markings and explicit definitions of lanes and traffic rules, constraining them to well-maintained roads with basic road layouts. The task can be simplified using prior maps with exhaustively labelled road layouts and lane connectivity. However, this approach still needs to track local agents and mandates significant labelling effort to keep prior maps up to date when road layouts change. To address these limitations, we need a solution that learns to traverse complex road layouts with no human supervision or prior map.

In this publication, we present a self-supervised approach to segment drivable paths in camera imagery with the goal of mapless autonomous driving. By exploiting human driving, additional vehicle sensors and local scene geometry, an expert system can automatically label drivable paths and untraversable obstacles in monocular camera imagery with no manual annotation. In doing so, vast quantities of realistic vehicle paths are automatically labelled across a diverse range of road layouts, traffic configurations and weather conditions, only limited by the time spent driving the data collection vehicle. When deployed with a monocular camera alone, the system predicts drivable paths suitable for autonomous urban navigation, as well as untraversable obstacles, at well over real-time speeds.

This manuscript was presented at the IEEE International Conference on Robotics and Automation (ICRA), 2017 [60]. A video summary of the publication can be found at: `http://youtu.be/rbZ8ck_1nZk`

Find Your Own Way: Weakly-Supervised Segmentation of Path Proposals for Urban Autonomy

Dan Barnes, Will Maddern and Ingmar Posner

***Abstract*— We present a weakly-supervised approach to segmenting proposed drivable paths in images with the goal of autonomous driving in complex urban environments. Using recorded routes from a data collection vehicle, our proposed method generates vast quantities of labelled images containing proposed paths and obstacles without requiring manual annotation, which we then use to train a deep semantic segmentation network. With the trained network we can segment proposed paths and obstacles at run-time using a vehicle equipped with only a monocular camera without relying on explicit modelling of road or lane markings. We evaluate our method on the large-scale KITTI and Oxford RobotCar datasets and demonstrate reliable path proposal and obstacle segmentation in a wide variety of environments under a range of lighting, weather and traffic conditions. We illustrate how the method can generalise to multiple path proposals at intersections and outline plans to incorporate the system into a framework for autonomous urban driving.**

I. Introduction

Road scene understanding is a critical component for decision making and safe operation of autonomous vehicles in urban environments. Given the structured nature of on-road driving, all autonomous vehicles must follow the 'rules of the road'; crucially, driving within designated lanes in the correct direction and negotiating intersections.

Current commercial systems that perform driver assistance and on-road autonomy typically depend on visual recognition of lane markings and explicit definitions of lanes and traffic rules, and therefore rely on simple road layouts with clear markings (e.g. well-maintained highways) [1], [2]. To extend these systems beyond multi-lane highways to complex urban environments and rural or undeveloped locations without clear or consistent lane markings, an alternative approach is required.

In this paper we present a weakly-supervised approach to segmenting *path proposals* for a road vehicle in urban environments given a single monocular input image. Our approach is capable of segmenting the proposed path for a vehicle in a diverse range of road scenes, without relying on explicit modelling of lanes or lane markings. We define the term *path proposal* as a route a driver would be expected to take through a particular road and traffic configuration. We present a novel method of automatically generating labelled images containing path proposals. Our method leverages both the behaviour of the data collection vehicle driver and additional sensors mounted to the vehicle, illustrated in Fig. 1. Using this approach we can generate vast quantities of labelled training data without any manual annotation, spanning a wide variety of road and traffic configurations under a number of different lighting and weather conditions limited only by the time spent driving the data collection vehicle. We use this data to train an off-the-shelf deep semantic segmentation network (e.g. SegNet [3]) to produce path proposal segmentations using *only* a monocular input image.

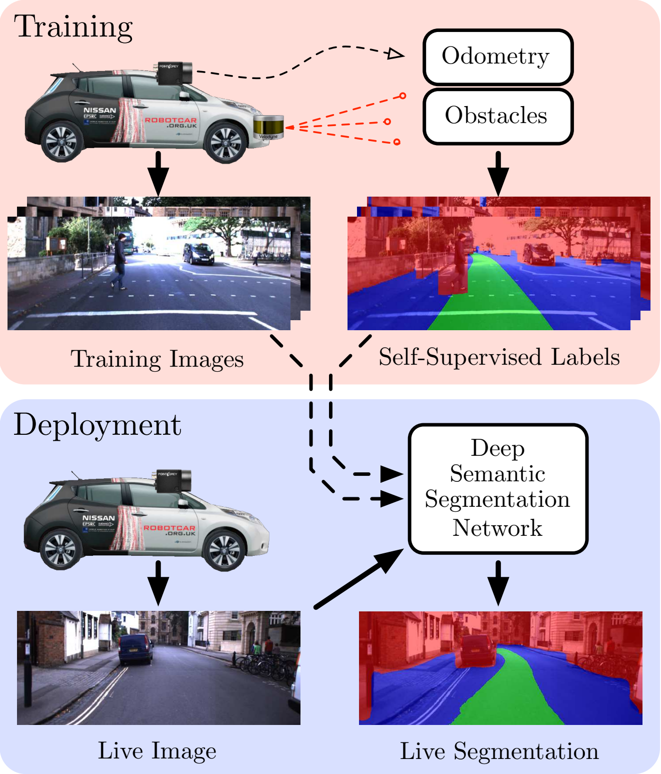


Fig. 1. Weakly-supervised path proposal segmentation using our approach. A data collection vehicle equipped with a camera as well as odometry and obstacle sensors is used to collect vast quantities of data during normal driving (top). The odometry and obstacle data is projected into the training images to generate weakly-supervised labels relevant for on-road autonomy, which are then used to train a deep semantic segmentation network. At run-time, a vehicle equipped with only a monocular camera can perform live segmentation of the drivable path and obstacles using the trained network (bottom), even in the absence of explicit lane markings.

We evaluate our approach using two large-scale autonomous driving datasets: the KITTI dataset [4], collected in Karlsruhe, Germany, and the large-scale Oxford RobotCar Dataset [5], consisting of over 1000km of recorded driving in Oxford, UK, over the period of a year. For each of these datasets we make use of the additional sensors on the vehicle and the trajectory taken by the driver as the weakly-supervised signal to train a pixelwise semantic classifier. We

Authors are from the Oxford Robotics Institute, Dept. Engineering Science, University of Oxford, UK. {dbarnes,wm,ingmar}@robots.ox.ac.uk

present segmentation results on the KITTI Road [6], Object and Tracking benchmarks and investigate the performance under different lighting and weather conditions using the Oxford dataset.

II. Related Work

Traditional methods of camera-based drivable path estimation for road vehicles involve preprocessing steps to remove shadow and exposure artefacts [7], [8], extraction of low-level road and lane features [9], [10], fitting road and lane models to feature detections [11], [12], and temporal fusion of road and lane hypotheses between successive frames [13], [14]. While effective in well-maintained road environments, these approaches suffer in the presence of occlusions, shadows and changing lighting conditions, unstructured roads, and areas with few or no markings [2]. Robustness can be significantly increased by combining images with radar [15] or LIDAR [16] but at an increased sensor cost.

More recently, advances in image processing using deep learning [17] have led to impressive results on the related problem of *semantic segmentation*, which aims to provide per-pixel labels of semantically meaningful objects for input images [18], [19], [3]. Deep networks make use of the full image context to perform semantic labelling of road and lane markings, and hence are significantly more robust than previous feature-based methods [3]. However, for automated driving these approaches depend on large-scale manually-annotated road scene datasets (notably CamVid [20] and Cityscapes [21], consisting of 700 and 5,000 labelled frames respectively), for which the labels are time-consuming and expensive to produce.

The challenges in building large-scale labelled datasets has led some researchers to consider virtual environments, for which ground-truth semantic labels can be rendered in parallel with synthetic camera images. Methods using customised video game engines have been used to produce hundreds of thousands of synthetic images with corresponding ground truth labels [22], [23]. While virtual environments allow large-scale generation of ground-truth semantic labels, they present two problems: firstly, rendering pipelines are typically optimised for speed and may not accurately reflect real-world images (both above approaches suggest rendered images are used only for augmenting real-world datasets and hence manual labelling is still necessary); secondly, the actions of the vehicle and all other agents in the virtual world must be pre-programmed and may not resemble real-world traffic scenarios. A recent method uses sparse 3D prior information to transfer labels to real-world 2D images [24] but requires sophisticated 3D reconstructions and manual 3D annotations.

Some approaches have proposed bypassing segmentation entirely and learning a direct mapping from input images to vehicle behaviour [25], [26]. These methods also use the driver of the data collection vehicle to generate the supervised labels for the network (e.g. steering angle) and have recently demonstrated impressive results in real-world driving tests [27], but it is not clear how this approach generalises to scenarios where there are multiple possible drivable paths to consider (e.g. intersections). Our approach instead uses the data collection vehicle driver to implicitly label proposed paths in the image, but still allows a planning algorithm to choose the best path for the current route.

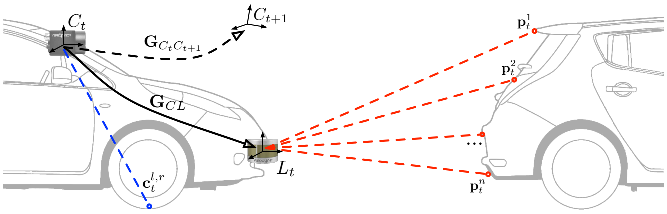


Fig. 2. Sensor extrinsics for weakly-supervised labelling. The survey vehicle (left) is equipped with a camera C and obstacle sensor L, e.g. a LIDAR scanner. The extrinsic transform $\mathbf{G}_{CL}$ between the camera and LIDAR is found using a calibration routine. The contact point $\mathbf{c}^{\{l,r\}}$ of the left and right wheels on the ground relative to the camera frame C is also measured at calibration time. At time t, the LIDAR scanner observes a number of points $\mathbf{p}_t^{1\cdots n}$ on obstacles, including other vehicles on the road (right). The relative pose $\mathbf{G}_{C_tC_{t+1}}$ of the camera between time t and $t+1$ is determined using vehicle odometry, e.g. using a stereo camera.

III. Weakly-Supervised Segmentation

In the following section we outline our approach for generating weakly-supervised training data for proposed path segmentation using video and sensor data recorded from a manually-driven vehicle.

A. Sensor Configuration

In addition to a monocular camera to collect input images, our approach depends on the following two capabilities for the data collection vehicle:

Vehicle odometry: a method of estimating the motion of the vehicle is required. For this we use stereo visual odometry [28], although other methods using inertial systems or wheel odometry would suffice.

Obstacle sensing: a method of detecting the 3D positions of impassible objects (both static and dynamic) in front of the vehicle is necessary to ensure that dynamic objects are not accidentally included in the drivable label area. For this we use a LIDAR scanner, though other methods that use dense stereo [29] or automotive radar would also be suitable.

Note that these additional sensing capabilities are only required for collecting training data; the resulting network only requires a monocular input image. Fig. 2 illustrates the sensor extrinsics for a vehicle equipped with a stereo camera and LIDAR sensor.

B. Weakly-Supervised Labelling

To generate class labels for pixels in the input image, we make use of large quantities of recorded data from the data collection vehicle driven by a human driver in a variety of traffic and weather conditions. We follow the general approach of methods that learn to drive by demonstration [30], [31], and assume the *proposed path* corresponds to the one chosen by the driver of the data collection vehicle in each scenario. Labels are then generated by projecting the *future* path of the vehicle into each image, over which object labels as detected by the LIDAR scanner are superimposed:

1) Proposed path projection: To project the future path of the vehicle into the current frame, it is necessary to know the size of the vehicle and the points of contact with the ground during the trajectory. We assume the position of the contact points $\mathbf{c}_{\{l,r\}}$ of the front left and right wheels on the ground relative to the camera C is determined as part of a calibration procedure. The position of the contact point $\mathbf{c}_{\{l,r\}}$ in the current camera frame C_t after k frames is then found as follows:

$$^{C_t}\mathbf{c}_{\{l,r\},k} = \mathbf{K}\mathbf{G}_{C_tC_{t+k}}\mathbf{c}_{\{l,r\}} \tag{1}$$

where $\mathbf{K}$ is the perspective projection matrix for the camera C and $\mathbf{G}_{C_tC_{t+k}}$ is the $\mathbb{SE}(3)$ chain of relative pose transforms formed by vehicle odometry from frame t to frame $t+k$ as follows:

$$\mathbf{G}_{C_tC_{t+k}} = \mathbf{G}_{C_tC_{t+1}} \times \mathbf{G}_{C_{t+1}C_{t+2}} \times \cdots \times \mathbf{G}_{C_{t+k-1}C_{t+k}} \tag{2}$$

Proposed path pixel labels are then formed by filling quadrilaterals in image coordinates corresponding to sequential future frames. The vertices of the quadrilateral are formed by the following points in camera frame C_t:

$$\left\{^{C_t}\mathbf{c}_{l,j},^{C_t}\mathbf{c}_{l,j-1},^{C_t}\mathbf{c}_{r,j-1},^{C_t}\mathbf{c}_{r,j}\right\} \tag{3}$$

where the index variable $j = \{1 \ldots k\}$. An illustration of the proposed path projection and labelling process is shown in Fig. 3. The choice of frame count k depends on the look-ahead distance required for path labelling and the accuracy of the vehicle odometry system used to provide relative frame transforms. In practice we choose k such that the distance between first and last contact points $\left\|\mathbf{G}_{C_tC_{t+k}}\mathbf{c}_{\{l,r\}} - \mathbf{c}_{\{l,r\}}\right\|$ exceeds 60 metres. Different camera setups with higher viewpoints may require greater path distances, but accumulated odometry error will affect far-field projections.

2) Obstacle projection: For some applications it may be sufficient to use just the proposed path labels to train a semantic segmentation network. However, for on-road applications in the presence of other vehicles and dynamic objects, a naive projection of the path driven will intersect vehicles in the same lane and label them as drivable paths as illustrated in Fig. 4. This may lead to catastrophic results when the labelled images are used to plan paths for autonomous driving, since vehicles and traffic may be labelled as traversable by the network.

We make use of the obstacle sensor mounted on the vehicle, in our case a LIDAR scanner. Each 3D obstacle point $\mathbf{p}_t^i$ observed at time t is projected into the camera frame C_t as follows:

$$^{C_t}\mathbf{p}_t^i = \mathbf{K}\mathbf{G}_{CL}\mathbf{p}_t^i \tag{4}$$

where $\mathbf{K}$ is the camera projection matrix and $\mathbf{G}_{CL}$ is the $\mathbb{SE}(3)$ extrinsic calibration between the camera and LIDAR sensor. For each camera-frame point $^{C_t}\mathbf{p}_t^i$, we take an approach inspired by "stixels" [29], [32] and label all pixels in the image on and above the point as an obstacle. This ensures all locations above and behind the detected obstacle are labelled as non-drivable, as illustrated in Fig. 3. Obstacle pixel labels take precedence over proposed path labels to ensure correct labelling of safe drivable paths as illustrated in Fig. 4.

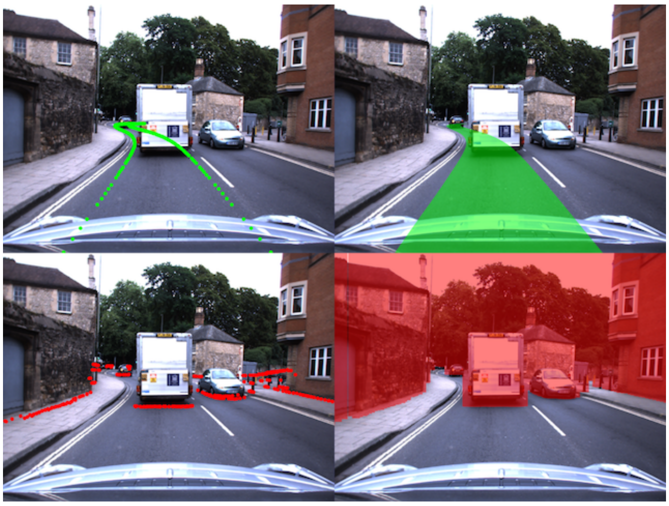


Fig. 3. Ground contact point (top) and obstacle point (bottom) projection into images. At time t, ground contact points $\mathbf{c}_{\{l,r\},j}$ (green) corresponding to the path of the vehicle up to k frames ahead are projected into the current image (top left). Pixel labels corresponding to drivable paths are filled in by drawing quadrilaterals between the left and right contact points between two successive frames (top right). At the same time, obstacle points $\mathbf{p}_t^i$ (red) from the current LIDAR scan are projected into the image (bottom left). Pixel labels corresponding to obstacles are formed by extending each of these points to the top of the image (bottom right). Note that the top and bottom sections of the image corresponding to the sky and vehicle bonnet are removed before training.

In most images, there will be locations labelled as neither proposed path nor obstacle. These correspond to locations which the vehicle has not traversed, and no positive identification of obstacles have been made. Typically these areas correspond to the road area outside the current lane (including lanes for oncoming traffic), kerbs, empty pavements and ditches. We refer to these locations as "unknown area" since it is not clear whether the vehicle should enter these spaces; this would be a decision for a higher-level planning framework as discussed in Section VII.

C. Semantic Segmentation

Once proposed path, obstacle and unknown area labels are automatically generated for a large number of recorded images, they can be used to train a semantic segmentation network to classify new images from a different vehicle equipped with only a monocular camera. We make use of SegNet [3], a deep convolutional encoder-decoder architecture for pixelwise semantic segmentation. Although higher-performing network architectures now exist (e.g. [19]), SegNet provides real-time evaluation on consumer GPUs, making it suitable for deployment in an autonomous vehicle.

The weakly-supervised labelling approach described in this section can generate vast quantities of training data, limited only by the length of time spent driving the data collection vehicle. However, the types of routes driven will also bias the input data, as most on-road driving is performed in a straight line; a random subsample of the training data will consist mostly of straight-line driving. In practice we subsample the data to 4Hz, before further subsampling based

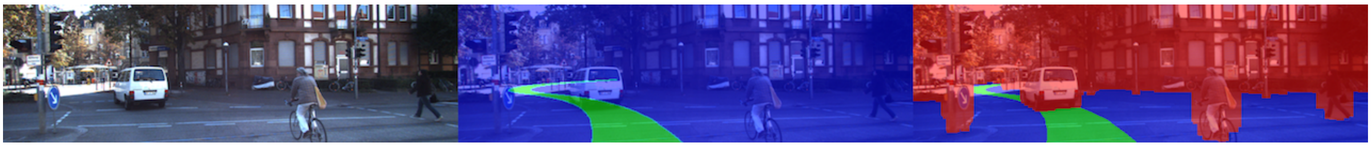


Fig. 4. Proposed path labels for an input image (left) before (middle) and after (right) applying obstacle labels from the LIDAR scanner. Without the obstacle labels, the proposed path (middle, green) intersects vehicles in the same lane as the path driven by the data collection vehicle, which in this case will erroneously label sections of the white van as drivable route. Adding labels for obstacles (right, red) ensures that dynamic objects including the van, cyclist and pedestrian are marked as non-drivable. Note that static obstacles such as the road sign and the building are also labelled as obstacles, which correctly handles occlusions (e.g. as the path turns right after the traffic lights).

TABLE I

VEHICLE AND SETUP SUMMARY

Vehicle	**Oxford RobotCar** **Nissan LEAF**	**KIT AnnieWAY** **VW Passat**
Camera Sensor	Point Grey Bumblebee XB3	2 x Point Grey Flea2
Input Resolution	640 x 256	621 x 187
LIDAR	SICK LD-MRS 4-beam	Velodyne HDL-64E 64-beam
Vehicle Width	2.43 m	2.2 m

TABLE II

TRAINING IMAGE SUMMARY STATISTICS

Dataset	**Condition**	**Training Images**
KITTI[1]	City	1264
	Residential	20734
	Road	2445
	Total	**24443**
Oxford[2]	Overcast	17085
	Sun	16299
	Rain	9822
	Night	4170
	Snow	2604
	Total	**49980**

on turning angle. For each frame we compute the average yaw rate $\bar{\Delta\psi}$ per frame for the corresponding proposed path as follows:

$$\bar{\Delta\psi} = \frac{1}{k}\sum_{i}^{k} \psi\left(\mathbf{G}_{C_{t+i-1}C_{t+i}}\right) \tag{5}$$

where $\psi(\mathbf{G})$ is a function that extracts the Euler yaw angle ψ from the $\mathbb{SE}(3)$ transform matrix $\mathbf{G}$. We then build a histogram of average yaw rates and randomly sample from the histogram bins to ensure an unbiased selection of different turning angles.

IV. EXPERIMENTAL SETUP

We build two different models for evaluation: one using the KITTI Raw dataset [4] and one using the Oxford RobotCar dataset. These datasets were collected using different vehicles with different sensor setups, summarised in Table I.

A. Platform Specifications

Both vehicles are equipped with stereo camera systems, and we use the stereo visual odometry approach in [28] to compute the relative motion estimates required in Eq. 2. The images from the cameras are cropped and downscaled to the resolutions listed in Table I before training. The Oxford RobotCar is equipped with a SICK LD-MRS LIDAR scanner, which performs obstacle merging and tracking across 4 scanning planes in hardware. We use points identified as "object contours" to remove erroneous obstacles due to noise and ground-strike. The Velodyne HDL-64E mounted on the AnnieWAY vehicle does not perform any object filtering, and hence we use the following approach to detect obstacles: we fit a ground plane to the 3D LIDAR scan using MLESAC [33], and treat all points more than 0.25m above this plane as obstacles, as illustrated in Fig. 6. This approach effectively identifies obstacles the vehicle may collide with even in the presence of pitching and rolling motions. The camera-LIDAR calibration $\mathbf{G}_{CL}$ for the RobotCar vehicle was determined using the method in [34]; for the AnnieWAY vehicle the calibration provided with the KITTI Raw dataset was used.

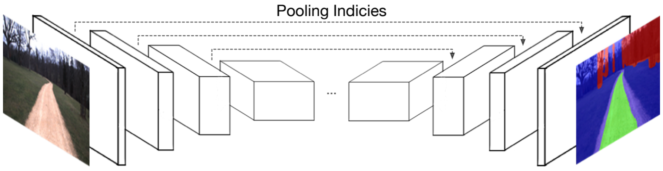


Fig. 5. Semantic segmentation is performed using a common encoder-decoder architecture (e.g. [3]) where the feature representation is progressively spatially compressed before being expanded to a full resolution per-pixel class prediction.

B. Network Training

For the KITTI model, we made use of the available City, Residential and Road data from the KITTI Raw dataset. For the Oxford model, we selected a diverse range of weather conditions for each traversal of the route, including 9 overcast, 8 with direct sun, 4 with rain, 2 at night and 1 with snow; each traversal consisted of approximately 10km of driving. The number of labelled images used to train each model is shown in Table II and some examples are shown in Fig. 7. In total we used 24,443 images to train the KITTI model, and 49,980 images for the Oxford model.

For both datasets we built semantic classifier models using the standard SegNet convolutional encoder-decoder architecture. The same SegNet parameters were used for both datasets, with modifications only to account for the differences in input image resolution. We randomly split the input data into 75% training and 25% validation sets, performed training for 100 epochs then selected the best-performing model according to the validation set results.

[1] http://www.cvlibs.net/datasets/kitti/
[2] http://robotcar-dataset.robots.ox.ac.uk

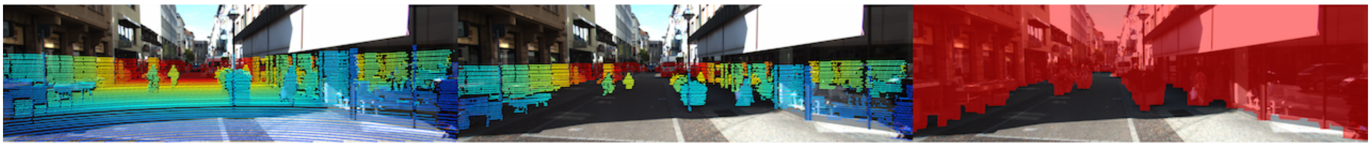


Fig. 6. Obstacle labelling using Velodyne data for the KITTI dataset. Raw Velodyne scans (left) contain returns from the road surface as well as nearby obstacles. We fit a ground plane using MLESAC and retain only points 0.25m above the plane (middle). We then label pixels using the approach in Section III-B.2 (right) to ensure accurate labels on obstacles while retaining drivable surfaces on the ground.

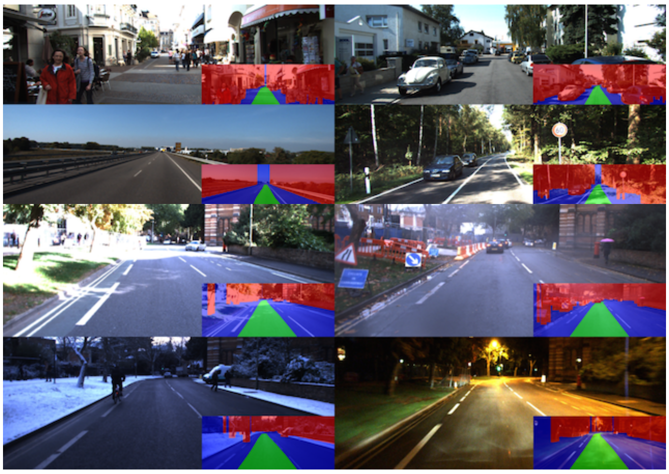


Fig. 7. Example training images with weakly-supervised labels from the KITTI (top) and Oxford (bottom) datasets. The weakly-supervised approach generates proposed path and obstacle labels for a diverse set of locations in the KITTI dataset, and a diverse set of conditions for the same location in the Oxford dataset. No manual annotation is required to generate the labels.

For the comparison using the KITTI Road benchmark presented in Section V-B.1, we trained an additional SegNet model on only the training images provided for the Ego-Lane Estimation Evaluation. Note that these ground truth images were not provided to the model trained using the weakly-supervised approach described above. For the object detection evaluation using the KITTI Object and Tracking datasets, we have ensured that there is no overlap between images selected to train the weakly-supervised labels and the images with ground truth labels used in the evaluation.

V. Results

For reliable on-road driving, the semantic segmentation must function in multiple environments under the range of lighting, weather and traffic conditions encountered during normal operation. In this section we evaluate the performance of both the KITTI model and Oxford model under a range of different test conditions.

A. Oxford Dataset

We evaluate the Oxford model by generating ground truth labels for a further four datasets not used for training, consisting of 2,718 images in sunny conditions, 2,481 images in cloudy conditions, 2,340 images collected at night and 1,821 images collected in the rain, for a total of 9,360 test images. Table III presents the segmentation results for the three classes in each of the four different conditions in the test datasets listed above, where the "All" column shows the mean of precision (PRE), recall (REC) and intersection-over-union (IoU) across all classes. The model provides good performance across the different conditions with mean IoU scores exceeding 80% in all cases, with the highest performance in cloudy weather and lowest at night, due to the reduced image quality in low-light conditions. Fig. 8 illustrates the output of the network for four images of the same location under different conditions. Despite significant changes in lighting and weather, the network correctly determines the proposed path through the crossing and identifies obstacles (e.g. construction barriers). This result demonstrates that the weakly-supervised approach can be used to train a single network that segments proposed paths and obstacles across a wide range of conditions without explicitly modelling environmental changes due to lighting, weather and traffic. Fig. 9 presents a number of locations where the network proposed a valid path in the absence of explicit road or lane markings, instead using the context of the road scene to infer the correct route.

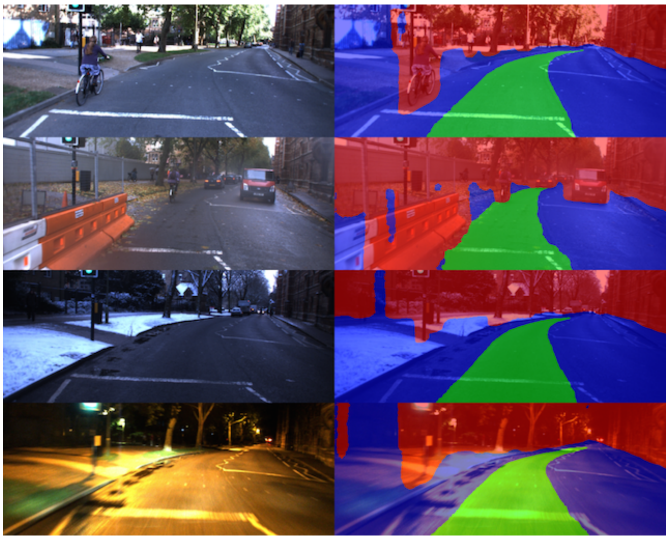


Fig. 8. Semantic segmentation on frames captured at the same location under different conditions. Despite significant changes in appearance between sunny, rainy, snowy and night-time conditions, the network correctly segments the proposed drivable path and labels obstacles including cyclists, other vehicles and road barriers.

B. KITTI Benchmarks

To demonstrate how our weakly-supervised labelling approach can lead to useful performance for autonomous driving tasks, we evaluate it on two different benchmarks from the KITTI Vision Benchmark Suite: ego-lane segmentation and object detection. However, neither of these benchmarks are an exact match for the segmentation results provided by the network, as they were designed for different purposes. Accordingly, we present alternative metrics based on the provided ground truth to quantitatively evaluate our sys-

TABLE III

SEGMENTATION RESULTS FOR OXFORD TEST DATA ACROSS VARYING CONDITIONS.

Condition	Proposed Path	Obstacle	Unknown Area	All
Night	PRE 86.50% REC 87.75% IoU 77.18%	PRE 93.60% REC 93.71% IoU 88.06%	PRE 88.88% REC 88.31% IoU 79.53%	PRE 89.66% REC 89.92% IoU 81.59%
Rain	PRE 89.55% REC 86.97% IoU 78.95%	PRE 94.04% REC 96.88% IoU 91.27%	PRE 91.41% REC 88.73% IoU 81.90%	PRE 91.66% REC 90.86% IoU 84.04%
Overcast	**PRE 91.13%** **REC 92.63%** **IoU 84.97%**	PRE 94.76% REC 96.68% IoU 91.77%	**PRE 93.41%** **REC 90.53%** **IoU 85.09%**	**PRE 93.10%** **REC 93.28%** **IoU 87.27%**
Sun	PRE 89.50% REC 89.53% IoU 81.02%	**PRE 94.85%** **REC 97.01%** **IoU 92.16%**	PRE 92.56% REC 90.05% IoU 83.97%	PRE 92.30% REC 92.20% IoU 85.72%

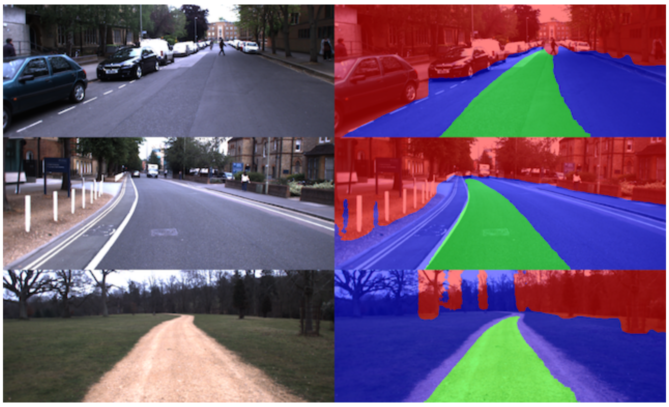


Fig. 9. Path proposals in locations without explicit lane dividers or road markings. Using the context of the road scene the network infers the correct proposed path (top, middle), even for gravel roads never seen in the training data (bottom).

tem. Note that the following two sections present different evaluation metrics for the same model trained on the same input data and should be interpreted in concert; the network produces both path and obstacle labels for each test image even when only one class is under evaluation.

1) Ego-lane Segmentation: The closest analogue to a *proposed path* in the KITTI benchmark suite is the ego-lane, consisting of the entire drivable surface within the lane the vehicle currently occupies [6]. The ego-lane dataset consists of 95 training and 96 test images, each with manually annotated ground truth labels. We trained an additional SegNet model on the provided ground truth training images to compare to our model trained on weakly-supervised labelled images, as detailed in Section IV-B. The results of both models on the KITTI website benchmark is shown in Table IV. Fig. 10 illustrates a sample network output for both models. The weakly-supervised model outperforms the model trained on the provided ground-truth images, with a 20% increase in max F-score and 15% increase in precision exceeding 90% in total, despite never making use of manually annotated ground truth images or explicit encoding of lane markings. Although the overall performance is not competitive with those generated by more sophisticated network architectures on the KITTI leaderboard (due to the different definition of ego-lane and proposed path), this result strongly indicates that the weakly-supervised approach generates segmentations useful for real-world path planning. The differences in the number of training images used for each model is illustrative of the fact that manually-annotated datasets will always be more be more time-consuming and expensive to produce than our weakly-supervised approach; even if manually annotated data is also available, for many tasks our approach could be used as pre-training to further improve results.

2) Object Detection: While the KITTI benchmark suite does not contain a semantic segmentation benchmark, it does contain object instance bounding boxes in both the Object and Tracking datasets. The definition of an *object* in the KITTI benchmark (an individual instance of a vehicle or person within a bounding box) differs significantly from our definition of an *obstacle* as part of the weakly-supervised approach (any part of the scene the vehicle might collide with). However, we can evaluate object detection performance by ensuring that every *object* instance provided by the KITTI Object and Tracking benchmarks was also classified as an *obstacle* by our segmentation approach; hence we aim for the highest *pixel-wise recall* score. For each object instance we evaluate the number of pixels within the bounding box classified as an obstacle using our weakly-supervised approach, as illustrated in Fig. 11. We present three different recall metrics: pixel recall, which includes all pixels under all bounding boxes for each object class, and two variants of instance recall, which requires a certain fraction of obstacle-labelled pixels within each bounding box instance before the object is considered as "detected" (thresholds of 50% and 75% are presented). We present recall results on the data provided as part of the Object and Tracking datasets (consisting of 15,047 images with 87,343 total object instances) in Table V, and an example detection is shown in Fig. 11. We have combined the object classes as follows: car, van, truck and tram labels are grouped as Vehicle; pedestrian, person sitting and cyclist labels are grouped as Person, and all others are grouped as Misc. The results show that the weakly-supervised segmentation approach is reliably labelling *objects* as *obstacles* regardless of object class (and performs especially well for an instance recall threshold of 50%); this is critical to avoid planning trajectories that intersect other vehicles or road users.

C. Limitations

Under some conditions the network fails to produce useful proposed path segmentations, as illustrated in Fig. 12. These

TABLE IV

EGO-LANE SEGMENTATION RESULTS ON THE KITTI ROAD BENCHMARK

Training	Benchmark	MaxF	AP	PRE	REC	FPR	FNR
Provided	UM_LANE	52.42%	37.85%	77.88%	39.50%	1.98%	60.50%
Weakly-Supervised	UM_LANE	**72.88%**	**64.49%**	**92.78%**	**60.01%**	**0.82%**	**39.99%**

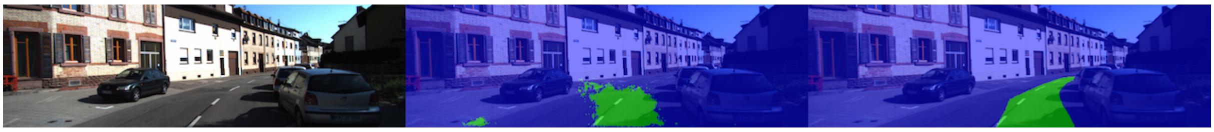


Fig. 10. Example ego-lane segmentation results using the KITTI Road dataset. For the given input image (left), a SegNet model trained on the small number of manually-annotated ground truth images (middle) performs poorly in comparison with the model trained on the much larger weakly-supervised dataset (right) generated without manual annotation.

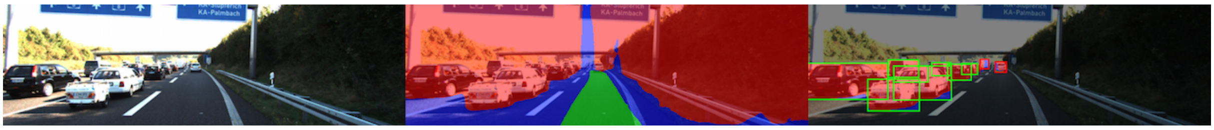


Fig. 11. Example object detection results using obstacle segmentation. For a given input image (left), the network labels areas corresponding to proposed path, obstacle and unknown area (middle). For each ground-truth bounding box provided in the KITTI Object and Tracking datasets, we compute the ratio of pixels labelled as obstacle by our method (right). For each object instance, we consider it detected (green outline) if more than 75% of the pixels within the bounding box are labelled as obstacles. Note that even for failed detections (red outline), a number of the pixels were still labelled as obstacle, and due to the tight obstacle outlines provided by our method we may miss portions of the bounding box (e.g. undercarriage of vehicles at bottom left).

TABLE V

OBSTACLE SEGMENTATION RESULTS ON THE KITTI OBJECT AND TRACKING DATASETS

Metric	Vehicle	Person	Misc	All
Pixel Recall	93.73%	92.47%	94.11%	93.53%
Instance Recall (>50%)	99.52%	99.65%	99.29%	99.55%
Instance Recall (>75%)	98.15%	97.38%	96.73%	97.93%

failure cases are mostly due to limitations of the sensor suite (poor exposure or low field of view), and could be addressed using a larger number of higher-quality cameras.

D. Route Generalisation

As the weakly-supervised labels are generated from the recording of a data collection trajectory, it can only provide one proposed path per image at training time. However, at intersections and other locations with multiple possible routes, at test time the resulting network frequently labels multiple possible proposed paths in the image as shown in Fig. 13; this is an important step towards decision-making for topological navigation within a road network. Currently we have no ground truth to evaluate route generalisation; we present qualitative results here for illustration only and plan to further characterise this effect in a future publication.

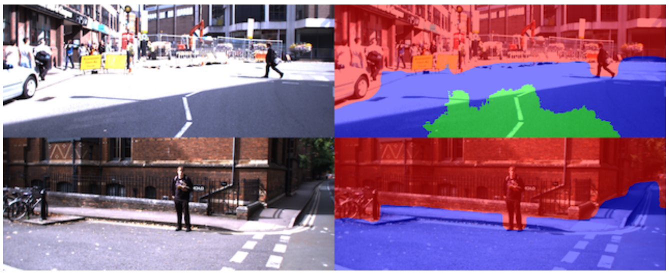


Fig. 12. Proposed path segmentation failures. (Top) Overexposed or underexposed images will lead to incorrect path segmentation; this could be addressed by using a high-dynamic-range camera. (Bottom) At some intersections during tight turns, there is no clear path to segment as it falls outside the field of view of the camera; using a wider field of view lens or multiple cameras in a surround configuration would address this limitation.

VI. CONCLUSIONS

In this paper we have outlined our approach for weakly-supervised labelling of images for proposed path segmentation during on-road driving using only a monocular camera. We have demonstrated that by leveraging multiple sensors and the behaviour of the data collection vehicle driver, we are able to generate vast quantities of semantically-labelled training data relevant for autonomous driving applications; crucially, we do not require any manual labelling of images in order to train our segmentation network. Our approach does not depend on specific road markings or explicit modelling of lanes to propose drivable paths. We evaluated the approach in the context of ego-lane segmentation and obstacle detection using the KITTI dataset, outperforming networks trained on manually-annotated training data and providing reliable obstacle detections. We also demonstrated the robustness of the trained network to changes in lighting, weather and traffic conditions using the large-scale Oxford RobotCar dataset, with successful proposed path segmentation in sunny, cloudy, rainy, snowy and night-time conditions. We plan to integrate the network with a planning framework that includes our previous work on topometric localisation across experiences [35] as well as our semantic map-guided approach for traffic light detection [36] to enable fully autonomous driving in complex urban environments.

VII. ACKNOWLEDGEMENTS

This work was supported by the UK EPSRC Doctoral Training Partnership and Programme Grant EP/M019918/1.

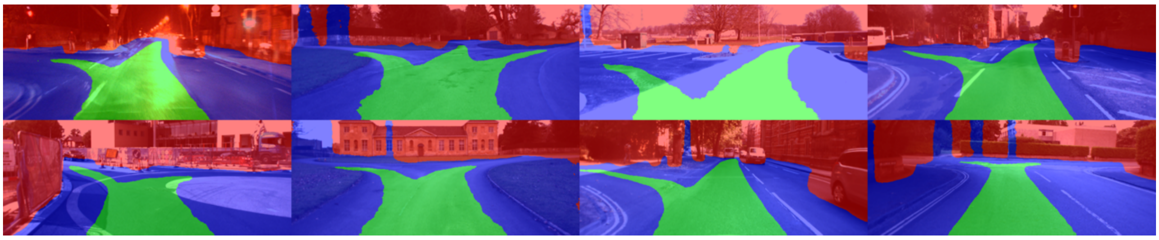


Fig. 13. Proposed path generalisation to multiple routes. At intersections and roundabouts the network will often label different possible paths, which can then be leveraged by a planning framework for decision making during autonomous navigation.

References

[1] S. Yenikaya, G. Yenikaya, and E. Düven, "Keeping the vehicle on the road: A survey on on-road lane detection systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, p. 2, 2013.

[2] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine vision and applications*, vol. 25, no. 3, pp. 727–745, 2014.

[3] V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[4] 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.

[5] 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.

[6] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 1693–1700.

[7] J. M. Álvarez, A. M. López, and R. Baldrich, "Shadow resistant road segmentation from a mobile monocular system," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2007, pp. 9–16.

[8] I. Katramados, S. Crumpler, and T. P. Breckon, "Real-time traversable surface detection by colour space fusion and temporal analysis," in *International Conference on Computer Vision Systems*. Springer, 2009, pp. 265–274.

[9] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, 2006.

[10] K. Yamaguchi, A. Watanabe, T. Naito, and Y. Ninomiya, "Road region estimation using a sequence of monocular images," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[11] R. Labayrade, J. Douret, J. Laneurit, and R. Chapuis, "A reliable and robust lane detection system based on the parallel use of three algorithms for driving safety assistance," *IEICE transactions on information and systems*, vol. 89, no. 7, pp. 2092–2100, 2006.

[12] A. S. Huang and S. Teller, "Probabilistic lane estimation for autonomous driving using basis curves," *Autonomous Robots*, vol. 31, no. 2-3, pp. 269–283, 2011.

[13] R. Jiang, R. Klette, T. Vaudrey, and S. Wang, "New lane model and distance transform for lane detection and tracking," in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2009, pp. 1044–1052.

[14] H. Sawano and M. Okada, "A road extraction method by an active contour model with inertia and differential features," *IEICE transactions on information and systems*, vol. 89, no. 7, pp. 2257–2267, 2006.

[15] B. Ma, S. Lakshmanan, and A. O. Hero, "Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 135–147, 2000.

[16] A. S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, "Finding multiple lanes in urban road networks with vision and LIDAR," *Autonomous Robots*, vol. 26, no. 2-3, pp. 103–122, 2009.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[18] 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.

[19] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a DCNN for semantic image segmentation," *arXiv preprint arXiv:1502.02734*, 2015.

[20] 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.

[21] 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.

[22] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234–3243.

[23] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," *arXiv preprint arXiv:1608.02192*, 2016.

[24] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger, "Semantic instance annotation of street scenes by 3D to 2D label transfer," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," DTIC Document, Tech. Rep., 1989.

[26] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun, "Off-road obstacle avoidance through end-to-end learning," in *Advances in neural information processing systems*, 2005, pp. 739–746.

[27] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

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

[29] D. Pfeiffer and U. Franke, "Efficient representation of traffic scenes by means of dynamic stixels," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010, pp. 217–224.

[30] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[31] D. Silver, J. A. Bagnell, and A. Stentz, "Learning autonomous driving styles and maneuvers from expert demonstration," in *Experimental Robotics*. Springer, 2013, pp. 371–386.

[32] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth, "Stixmantics: A medium-level model for real-time semantic scene understanding," in *European Conference on Computer Vision*. Springer, 2014, pp. 533–548.

[33] P. H. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[34] G. Pascoe, W. Maddern, and P. Newman, "Direct visual localisation and calibration for road vehicles in changing city environments," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 9–16.

[35] C. Linegar, W. Churchill, and P. Newman, "Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 787–794.

[36] D. Barnes, W. Maddern, and I. Posner, "Exploiting 3D semantic scene priors for online traffic light interpretation," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 573–578.

Statement of Authorship for joint/multi-authored papers for PGR thesis
To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Find Your Own Way: Weakly-Supervised Segmentation of Path Proposals for Urban Autonomy
Publication Status	Published
Publication Details	Dan Barnes, Will Maddern and Ingmar Posner Find Your Own Way: Weakly-Supervised Segmentation of Path Proposals for Urban Autonomy International Conference on Robotics and Automation (ICRA), 2017

Student Confirmation

Student Name:	Daniel Barnes		
Contribution to the Paper	My contributions to the paper were: Developed the idea behind the paper Collected and processed data Performed the experiments Wrote the paper with co-authors		
Signature	Daniel H Barnes	Date	18 / 11 / 2019

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Ingmar Posner		
Supervisor comments	The description above is accurate.		
Signature		Date	7/1/2020

4

Driven to Distraction: Self-Supervised Distractor Learning for Robust Monocular Visual Odometry in Urban Environments

In Chapter 3 we presented a behavioural path planning solution suitable for urban autonomous vehicles equipped only with a monocular camera. However, as mentioned in Section 2.3, if deployed without accurate egomotion, plans can not be followed safely no matter how optimal they may be. Precise monocular VO would provide the means for deployment with no additional hardware or cost, a core motivation in this thesis.

There has been considerable research in VO, as detailed in Section 2.3.1. The core challenges a VO system faces in urban environments are decomposing egomotion when observing large independently moving objects (such as buses), and in monocular setups, deriving absolute scale without additional sensing. Many approaches have been applied to these problems, such as estimating scale from perceived objects, and improved robustness with outlier rejection schemes and multi-motion estimation. However, fundamentally deriving scale and identifying preferable structure for motion estimation with monocular cameras remains a challenging task.

In this publication, we present a model-free and self-supervised approach to segment ephemeral objects in camera imagery, with the goal of improving VO in urban environments. By aligning data collected from repeated traversals and analysing consistently observed 3D structure, dense labels are automatically generated for both absolute depth and ephemerality (a measure of how beneficial each pixel is for motion estimation). A DNN trained on this data can be utilised to provide robustness, using ephemerality to suppress features likely to degrade VO, and scale in monocular VO with single-image depth prediction. Deployed in both sparse and dense VO systems, our approach provides robust metric-scale monocular VO in challenging urban environments at well over real-time speeds.

This manuscript was presented at the IEEE International Conference on Robotics and Automation (ICRA), 2018 [61]. A video summary of the publication can be found at: `https://youtu.be/ebIrBn_nc-k`

Driven to Distraction: Self-Supervised Distractor Learning for Robust Monocular Visual Odometry in Urban Environments

Dan Barnes, Will Maddern, Geoffrey Pascoe and Ingmar Posner

***Abstract*— We present a self-supervised approach to ignoring “distractors” in camera images for the purposes of robustly estimating vehicle motion in cluttered urban environments. We leverage offline multi-session mapping approaches to automatically generate a per-pixel *ephemerality mask* and depth map for each input image, which we use to train a deep convolutional network. At run-time we use the predicted ephemerality and depth as an input to a monocular visual odometry (VO) pipeline, using either sparse features or dense photometric matching. Our approach yields metric-scale VO using only a single camera and can recover the correct egomotion even when 90% of the image is obscured by dynamic, independently moving objects. We evaluate our robust VO methods on more than 400km of driving from the Oxford RobotCar Dataset and demonstrate reduced odometry drift and significantly improved egomotion estimation in the presence of large moving vehicles in urban traffic.**

I. Introduction

Autonomous vehicle operation in crowded urban environments presents a number of key challenges to any system based on visual navigation and motion estimation. In urban traffic where up to 90% of an image can be obscured by a large moving object (e.g. bus or truck), standard outlier rejection schemes such as RANSAC [1] will produce incorrect motion estimates due to the large consensus of features tracked on the moving object. The key to robust “distraction-free” visual navigation is a deeper understanding of which image regions are *static* and which are *ephemeral* in order to better decide which features to use for motion estimation.

In this paper we leverage large-scale offline mapping and deep learning approaches to produce a per-pixel *ephemerality mask* at run-time without requiring any semantic classification or manual labelling, as illustrated in Fig. 1 and the project video[1]. The ephemerality mask predicts stable image regions (e.g. buildings, road markings, static landmarks) that are likely to be useful for motion estimation, in contrast to dynamic or ephemeral objects (e.g. pedestrian and vehicle traffic, vegetation, temporary signage). In contrast to semantic segmentation approaches that explicitly label objects belonging to a-priori chosen classes and hence require manually annotated training data, our approach is trained using repeated traversals of the same route with a LIDAR-equipped survey vehicle producing per-pixel depth and ephemerality labels for a deep convolutional network as a fully self-supervised process.

We integrate the ephemerality mask as a component of a monocular visual odometry (VO) pipeline as an outlier rejection scheme. By leveraging the depth and ephemerality outputs of the network, we can produce robust metric-scale VO using *only a single camera* mounted to a vehicle. Our approach leads to significantly more reliable motion estimation when evaluated over hundreds of kilometres of driving in complex urban environments in the presence of heavy traffic and other challenging conditions.

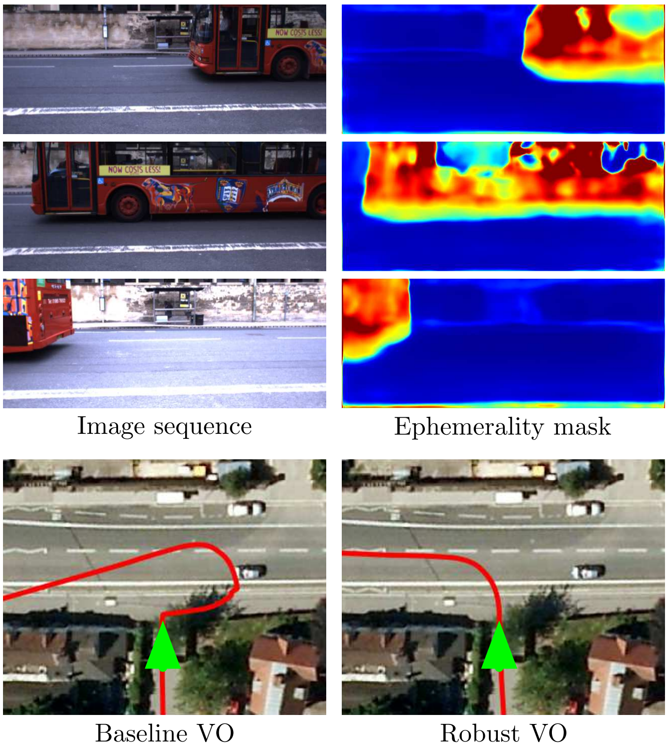


Fig. 1. Robust motion estimation in urban environments using a single camera and a learned ephemerality mask. When making a left turn onto a main road, a large bus passes in front of the vehicle (green arrow) obscuring the view of the scene (top left). Our learned ephemerality mask correctly identifies the bus as an unreliable region of the image for the purposes of motion estimation (top right). Traditional visual odometry (VO) approaches will incorrectly estimate a strong translational motion to the right due to the dominant motion of the bus (bottom left), whereas our approach correctly recovers the vehicle egomotion (bottom right).

II. Related Work

Estimating an ephemerality mask is closely related to background subtraction approaches [2], [3], which build statistics over background appearance based on training data from a static camera to identify discrepancies in live images. These methods are typically used in surveillance applications

Authors are from the Oxford Robotics Institute, Dept. Engineering Science, University of Oxford, UK. {dbarnes,wm,gmp,ingmar}@robots.ox.ac.uk

[1] https://youtu.be/ebIrBn_nc-k

and have limited robustness to general 3D camera motion in complex scenes, as experienced on a vehicle [4], [5].

Conversely, there is a significant body of work on detection and tracking of moving (foreground) objects [6], [7], [8], which has been applied to robust VO in dynamic environments [9] and scale references for monocular SLAM [10]. However, these approaches require large quantities of manually-labelled training data of moving objects (e.g. cars, pedestrians) and the chosen object classes must cover all possibly-moving objects to avoid false negatives. Recent 3D SLAM approaches have integrated per-pixel semantic segmentation layers to improve reconstruction quality [11], [12], but again rely on laboriously manually-annotated training data and chosen classes that encompass all object categories.

Unsupervised approaches have recently been introduced to estimate depth [13], egomotion [14] and 3D reconstruction [15]. These methods are attractive for large-scale use as they only require raw video footage from a monocular or stereo camera, without any ground-truth motion estimates or semantic labels. In particular, [14] introduces an “explainability mask”, which highlights image regions that disagree with the dominant motion estimate. However, the explainability mask differs from the ephemerality mask in that it only recognises non-dominant moving objects, and hence will still produce incorrect motion estimates when significantly occluded by a large, independently moving object.

Our approach is inspired by the distraction-suppression methods presented in [16], [17]. Both methods use a prior 3D map to estimate a mask that quantifies reliability for motion estimation, which is integrated into a VO pipeline. We significantly extend the map prior approach of [16] to multi-session mapping and quantify ephemerality using a structural entropy metric, and use the result to automatically generate training data for a deep convolutional network. As a result, our approach does not rely on live localisation against a prior map or live dense depth estimation from stereo, and hence can operate in a wider range of (unmapped) locations with a reduced (monocular-only) sensor suite.

III. Learning Ephemerality Masks

In this section we outline our approach for automatically building ephemerality masks by leveraging an offline 3D mapping pipeline. Note that LIDAR and stereo camera sensors are only required for the survey vehicle to collect training data; at run-time only a monocular camera is required. Our method takes the following steps:

1) Prior 3D Mapping: Using a survey vehicle equipped with a stereo camera and LIDAR scanner, we perform multiple traversals of the target environment. By analysing structural consistency across multiple mapping sessions with an entropy-based approach, we determine what constitutes the static (non-ephemeral) structure of the scene.

2) Ephemerality Labelling: We project the prior 3D static structure into every stereo camera image collected during the survey, and compare it to the structure computed by a dense stereo approach (similar to [16]). In the presence of traffic or dynamic objects these will differ considerably; we compute ephemerality as a weighted sum of disparity and normal difference between prior and true 3D structure.

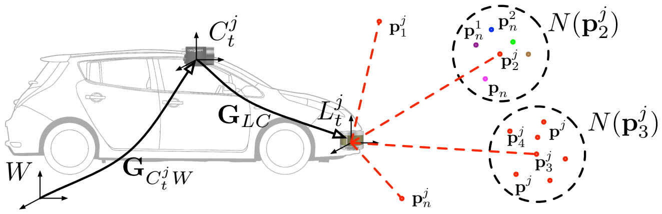


Fig. 2. Multi-session mapping and 3D pointcloud entropy computation. For each traversal j, we compute the global pose of the vehicle $\mathbf{G}_{C_t^j W}$ at each timestamp t and project points $\mathbf{p}$ into the global frame W. We then analyse the neighbourhood N of each point $\mathbf{p}_i^j$; in neighbourhoods where points are well distributed between traversals $\{1 \cdots j\}$ such as $N(\mathbf{p}_2^j)$ the scene is likely to be static, and where points are mostly derived from one traversal such as $N(\mathbf{p}_3^j)$ the structure is ephemeral. We quantify static scenes using an entropy metric applied to each neighbourhood $N(\mathbf{p})$.

3) Network Training: We train a deep convolutional network to predict the resulting pixel-wise depth and ephemerality mask using only input monocular images. At run-time we produce live depth and ephemerality masks even in locations not traversed by the survey vehicle.

In the following sections we describe these steps in detail.

A. Prior 3D Mapping

Given a survey vehicle equipped with a camera C and LIDAR L illustrated in Fig. 2 that has performed a number of traverses j of an environment, we recover each global camera pose $\mathbf{G}_{C_t^j W}$ at time t relative to world frame W with a large-scale offline process using the stereo mapping and navigation approach in [18]. We then compute the position of each 3D LIDAR point $\mathbf{p}_i^j \in \mathbb{R}^3$ in world frame W using the camera pose and LIDAR-camera calibration $\mathbf{G}_{LC}$ as follows:

$$ {}^{W}\mathbf{p}_i^j = \mathbf{G}_{C_t^j W} \mathbf{G}_{LC} \mathbf{p}_i^j \tag{1}$$

Given the pointcloud of all points $\mathbf{p}$ collected from all traversals j, we wish to compute the local entropy of each region of the pointcloud, to quantify how reliable the region is across each traversal. We define a neighbourhood function $N(\cdot)$, where a point $\mathbf{p}_t^k$ belongs to a neighbourhood if it satisfies the following condition:

$$\mathbf{p}_t^k \in N(\mathbf{p}_i) \iff \left\|\mathbf{p}_i - \mathbf{p}_t^k\right\|_2 < \alpha \tag{2}$$

where α is a neighbourhood size parameter, typically set to 0.5m in our experiments. For each query point $\mathbf{p}_i$, we then build a distribution $p_i(j)$ over the traverses j from which points fell in the neighbourhood of the query point as follows:

$$p_i(j) = \frac{1}{|N(\mathbf{p}_i)|} \sum_{\mathbf{p}_t^k \in N(\mathbf{p}_i)} \begin{cases} 1, & j = k \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Intuitively, neighbourhoods of points that are well-distributed between different traversals indicate static structure, whereas neighbourhoods of points that were only

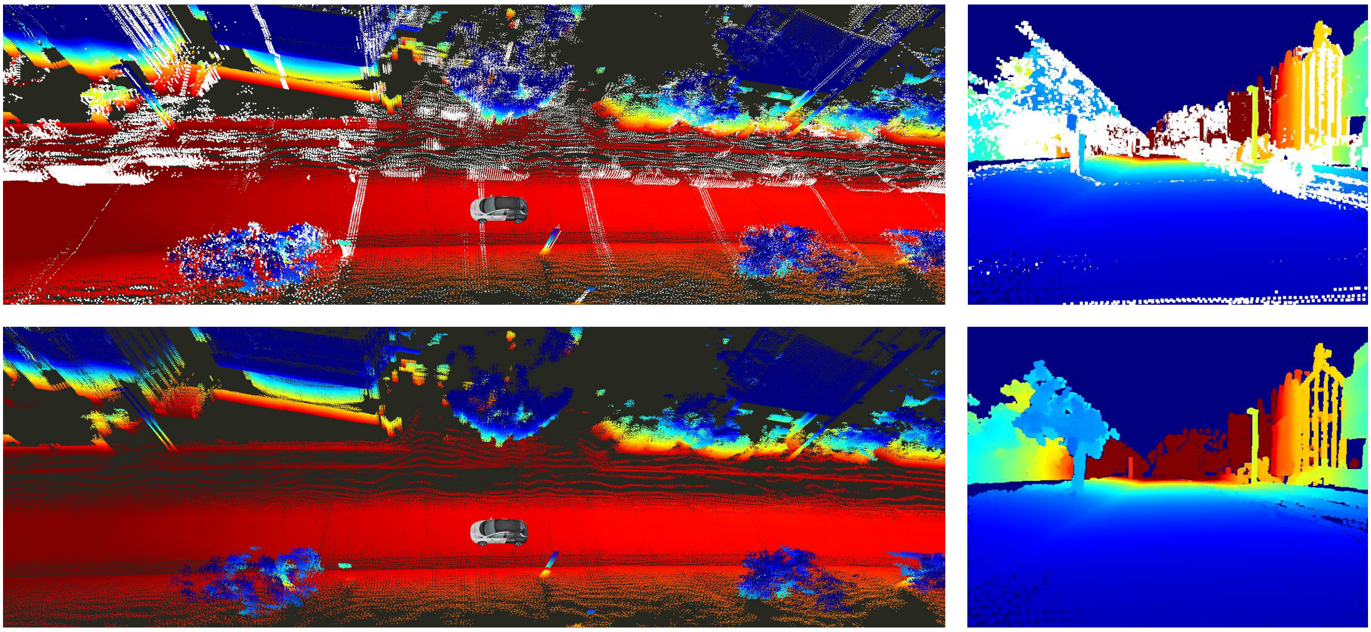


Fig. 3. Prior 3D mapping to determine the static 3D scene structure. Alignment of multiple traversals of a route (top left) will yield a large number of points only present in single traversals, e.g. traffic or parked vehicles, here shown in white. These points will corrupt a synthetic depth map (top right). Our entropy-based approach removes 3D points that were only observed in some traversals, and retains the structure that remained static for the duration of data collection (bottom left), resulting in high-quality synthetic depth maps (bottom right).

sourced from one or two traversals are likely to be ephemeral objects. We compute the neighbourhood entropy $\mathrm{H}(p_i)$ of each point across all n traversals as follows:

$$\mathrm{H}\left(p_i\right) = -\sum_{j=1}^{n} p_i\left(j\right) \log\left(p_i\left(j\right)\right) \tag{4}$$

We classify a point $\mathbf{p}_i$ as static structure $\mathbf{p}_i^S$ if the neighbourhood entropy $\mathrm{H}(p_i)$ exceeds a minimum threshold β; all other points are estimated to be ephemeral and are removed from the static 3D prior. The pointcloud construction, neighbourhood entropy and ephemeral point removal process are illustrated in Fig. 3.

B. Ephemerality Labelling

Given the prior 3D static pointcloud $\mathbf{p}^S$ and globally aligned camera poses C, we can produce a synthetic depth map for each survey image, as illustrated in Fig. 4. To handle visibility constraints we make use of the hidden point removal approach in [19]. For every pixel i into which a valid prior 3D point projects, we compute the expected disparity d_i^S and normal $\mathbf{n}_i^S$ using the local 3D structure of the pointcloud.

In the presence of dynamic objects, the scene observed from the camera will differ from the expected prior 3D map. We use the offline dense stereo reconstruction approach of [20] to compute the true disparity d_i and normal $\mathbf{n}_i$ for each pixel in the survey image, illustrated in Fig. 4. We define the ephemerality mask $\mathcal{E}_i$ as the weighted difference between the expected static and true disparity and normals as follows:

$$\mathcal{E}_i = \gamma \left\| d_i^S - d_i \right\|_1 + \delta \cos^{-1}\left(\mathbf{n}_i^S \cdot \mathbf{n}_i\right) \tag{5}$$

where γ and δ are weighting parameters, and $\mathcal{E}_i$ is bounded to $[0, 1]$ after computation.

C. Network Architecture

We adopt a convolutional encoder-multi-decoder network architecture to predict both disparity and ephemerality masks from a single image, as illustrated in Fig. 5, by adding an additional decoder to the architecture in [21].

To train the disparity output we use the stereo photometric loss proposed in [21], optionally semi-supervised using the prior LIDAR disparity d_i^S to ensure metric-scaled outputs. For the ephemerality output we use the $\mathcal{L}_1$ loss for each pixel with a valid ephemerality label. We balance these losses using the multi-task learning approach in [22], which continuously updates the inter-task weighting during training.

As in [21], we trained our model from scratch for 50 epochs, with a batch size of 8 using the Adam [23] optimiser, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We used an initial learning rate of $\lambda = 10^{-4}$ which we kept constant for the first 30 epochs before halving it every 10 epochs until the end.

IV. Ephemerality-Aware Visual Odometry

We leverage the live depth and ephemerality mask produced by the network to produce reliable visual odometry estimates accurate to metric scale. We present two robust VO approaches: a sparse feature-based approach and a dense photometric approach. Each integrates the ephemerality mask in order to estimate egomotion using only static parts of the scene, and uses the learned depth to estimate relative motion to the correct scale. This improves upon traditional monocular VO systems that cannot recover absolute scale [24]. Both our odometry approaches are optimised for real-time performance on a vehicle platform.

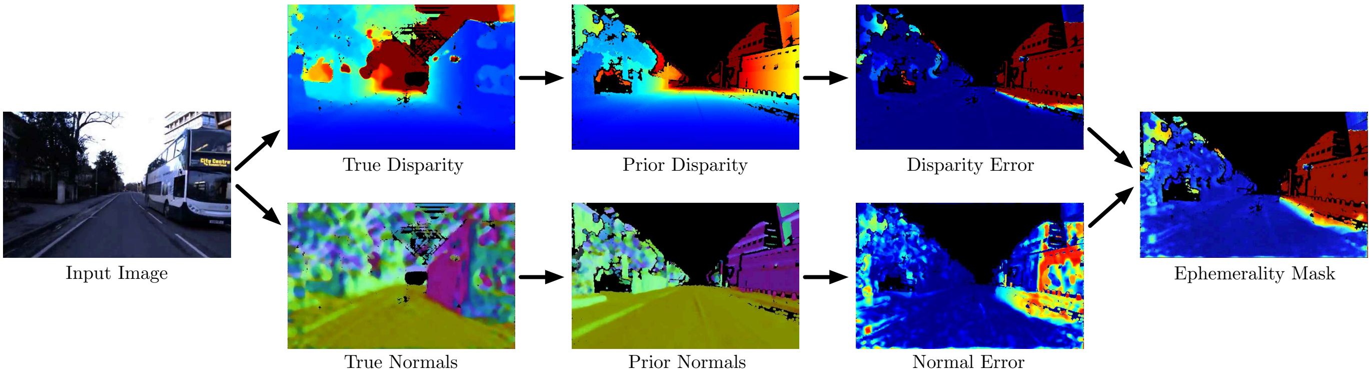


Fig. 4. Ephemerality labelling process. From input images (left) we compute the true disparity d_i and normals $\mathbf{n}_i$ using an offline dense stereo approach. We then project the prior 3D pointcloud $\mathbf{p}^S$ into the image to form the prior disparity d_i^S and prior normal $\mathbf{n}_i^S$. The disparity and normal error terms are combined to form the ephemerality mask (right).

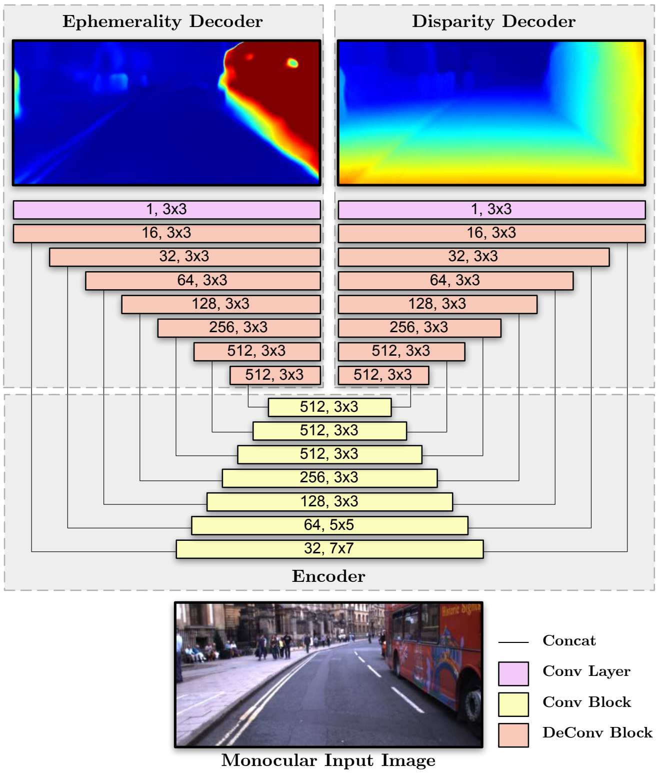


Fig. 5. Network architecture for ephemerality and disparity learning. The width of each block indicates the spatial dimensions of the feature map, which vary by a factor of 2 between blocks. The number of output channels and filter dimensions are also detailed for each block.

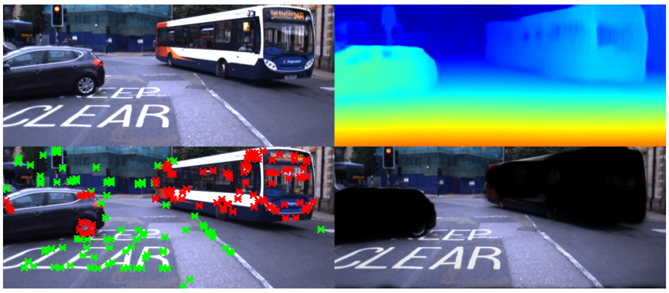


Fig. 6. Input data for ephemerality-aware visual odometry. For a given input image (top left), the network predicts a dense depth map (top right) and an ephemerality mask. For sparse VO approaches, the ephemerality mask is used to select which features are used for optimisation (bottom left), and for dense VO approaches the photometric error term is weighted directly by the ephemerality mask (bottom right).

A. *Sparse Monocular Odometry*

Our sparse monocular VO approach is derived from well-known stereo approaches [25], where sets of features are detected and matched across successive frames to build a relative pose estimate. Each feature $\mathbf{x}_i$ is parameterised as follows:

$$\mathbf{x}_i = \begin{bmatrix} u_i \\ v_i \\ d_i \end{bmatrix} \tag{6}$$

where (u_i, v_i) are the pixel coordinates and d_i is the disparity predicted by the deep convolutional network. The relative pose $\boldsymbol{\xi} \in \mathbb{SE}(3)$ is recovered by minimising the reprojection error between matched features $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$:

$$\underset{\boldsymbol{\xi}}{\arg\min} \sum_{i \in \mathcal{F}} s\left(\mathcal{E}_i\right) \left\|\mathbf{x}_i - \omega\left(\hat{\mathbf{x}}_i, \boldsymbol{\xi}\right)\right\|_2^2 \tag{7}$$

The warping function $\omega(\cdot) \rightarrow \mathbb{R}^2$ projects the matched feature $\hat{\mathbf{x}}_i$ into the current image according to relative pose $\boldsymbol{\xi}$ and the camera intrinsics. The set of all extracted features $\mathcal{F}$ is typically a small subset of the total number of pixels in the image. The step function $s(\mathcal{E}_i)$ is used to disable the residual according to the predicted ephemerality as follows:

$$s\left(\mathcal{E}_i\right) = \begin{cases} 1, & \mathcal{E}_i < \tau \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where τ is the maximum ephemerality threshold for a valid feature, typically set to 0.5. In practice we detect sparse features using FAST corners [26] and match using BRIEF descriptors [27] for real-time operation.

B. *Dense Monocular Odometry*

For the dense monocular approach, we adopt the method of [28] and combine our learned depth maps with the photometric relative pose estimation of [29]. Rather than a subset of pixels $\mathcal{F}$, all pixels i within the reference keyframe image $\mathcal{I}_r$ are warped into the current image $\mathcal{I}_c$ and the relative pose $\boldsymbol{\xi}$ is recovered by minimising the photometric error as follows:

$$\underset{\boldsymbol{\xi}}{\arg\min} \sum_{i \in \mathcal{I}_r} \left(1 - \mathcal{E}_i\right) \left\|\mathcal{I}_r\left(\mathbf{x}_i\right) - \mathcal{I}_c\left(\omega\left(\mathbf{x}_i, \boldsymbol{\xi}\right)\right)\right\|_2^2 \tag{9}$$

where the image function $\mathcal{I}(\mathbf{x}_i) \rightarrow \mathbb{R}^+$ returns the pixel intensity at location (u_i, v_i). Note that the ephemerality mask is used directly to weight the photometric residual; no thresholding is required. Fig. 6 illustrates the predicted depth, selected sparse features and weighted dense intensity values used for a typical urban scene.

V. Experimental Setup

We benchmarked our approach using hundreds of kilometres of data collected from an autonomous vehicle platform in a complex urban environment. Our goal was to quantify the performance of the ephemerality-aware visual odometry approach in the presence of large dynamic objects in traffic.

A. *Network Training*

We train our approach using eight 10km traversals from the Oxford RobotCar dataset [30] for a total of approximately 80km of driving. The RobotCar vehicle is equipped with a Bumblebee XB3 stereo camera and a LMS-151 pushbroom LIDAR scanner. For training we downsample the input images to 640 × 256 pixels and subsample to one image every metre before use; a total of 60,850 images were used for training. At run-time we produce ephemerality masks and depth maps at 50Hz using a single GTX 1080 Ti GPU.

B. *Evaluation Metrics*

We evaluate our approach on 42 further Oxford traversals for a total of over 400km. The evaluation datasets contain multiple detours and alternate routes, ensuring the method is tested in (unmapped) locations not present in the training datasets. To quantify the performance of the ephemerality-aware VO, we compute translational and rotational drift rates using the approach proposed in the KITTI odometry benchmark [31]. Specifically, we compute the average end-point-error for all subsequences of length $(100, 200, \ldots, 800)$ metres compared to the INS system installed on the vehicle.

In addition, we compare the instantaneous translational velocities of each method to that reported by the INS system (based on doppler velocity measurements). We manually selected 6,000 locations that include distractors, and evaluate velocity estimation errors in comparison to the average of all locations. This allows us to focus on dynamic scenes where independently moving objects produce erroneous velocity estimates in the baseline VO methods.

VI. Results

In addition to the quantitative results listed below, we present qualitative results for ephemerality masks produced in a range of different locations in Fig. 7.

A. *Odometry Drift Rates*

The end-point-error evaluation for each of the methods is presented in Table I. In both cases, the addition of the ephemerality mask reduced both average translational and rotational drift over the full set of evaluation datasets. Note that the metric scale for translational drift is derived from the depth map produced by the network, and hence both systems report translation in units of metres with low overall error rates using only a monocular camera. The sparse VO approach provided lower overall translational drift, whereas the dense approach produced lower orientation drift.

TABLE I
Odometry Drift Evaluation

VO Method	Translation [%]	Rotation [deg/m]
Sparse	6.55	0.0353
Sparse w/Ephemerality	**6.38**	**0.0321**
Dense	7.15	0.0373
Dense w/Ephemerality	**6.52**	**0.0307**

TABLE II
Velocity Error Evaluation

VO Method	All [m/s]	Distractors [m/s]
Sparse	0.0548	0.220
Sparse w/Ephemerality	**0.0406**	**0.0489**
Dense	0.0568	0.766
Dense w/Ephemerality	**0.0407**	0.424

B. *Velocity Estimates*

The velocity error evaluation for each of the methods is presented in Table II. Across all the evaluation datasets, the ephemerality-aware odometry approaches produce lower average velocity errors. However, in locations with distractors, the ephemerality-aware approaches produce significantly more accurate velocity estimates than the baseline approaches. In particular, the robust sparse VO approach is almost unaffected by distractors, whereas the baseline method reports errors 4 times greater. The dense VO approach generally produces poorer translational velocity estimates than the sparse approach, which corresponds with higher translational drift rates reported in the previous section. Fig. 8 presents the distribution of velocity errors for each of the approaches in the presence of distractors.

VII. Conclusions

In this paper we introduced the concept of an ephemerality mask, which estimates the likelihood that any pixel in an input image corresponds to either reliable static structure or dynamic objects in the environment, and can be learned using an automatic self-supervised approach. Crucially, we do not require any manual labelling or choice of semantic classes in order to train our approach, and at run-time we only require a single monocular camera to produce reliable ephemerality-aware visual odometry to metric scale. Over hundreds of kilometres our approach produces improved odometry resulting in lower drift rates, and significantly more robust velocity estimates in the presence of large dynamic objects in urban scenes.

The benefits of our approach are not restricted to improving motion estimation, and there are a number of avenues to explore in future work. Fig. 9 illustrates a foreground/background segmentation performed using the ephemerality mask; where we currently use the background to guide motion estimation, a detection and classification approach could be guided by the foreground mask to efficiently track dynamic objects in the scene. We plan to integrate the approaches in this paper for improved localisation, motion estimation, obstacle avoidance and scene understanding for fully autonomous vehicles operating in complex urban environments.

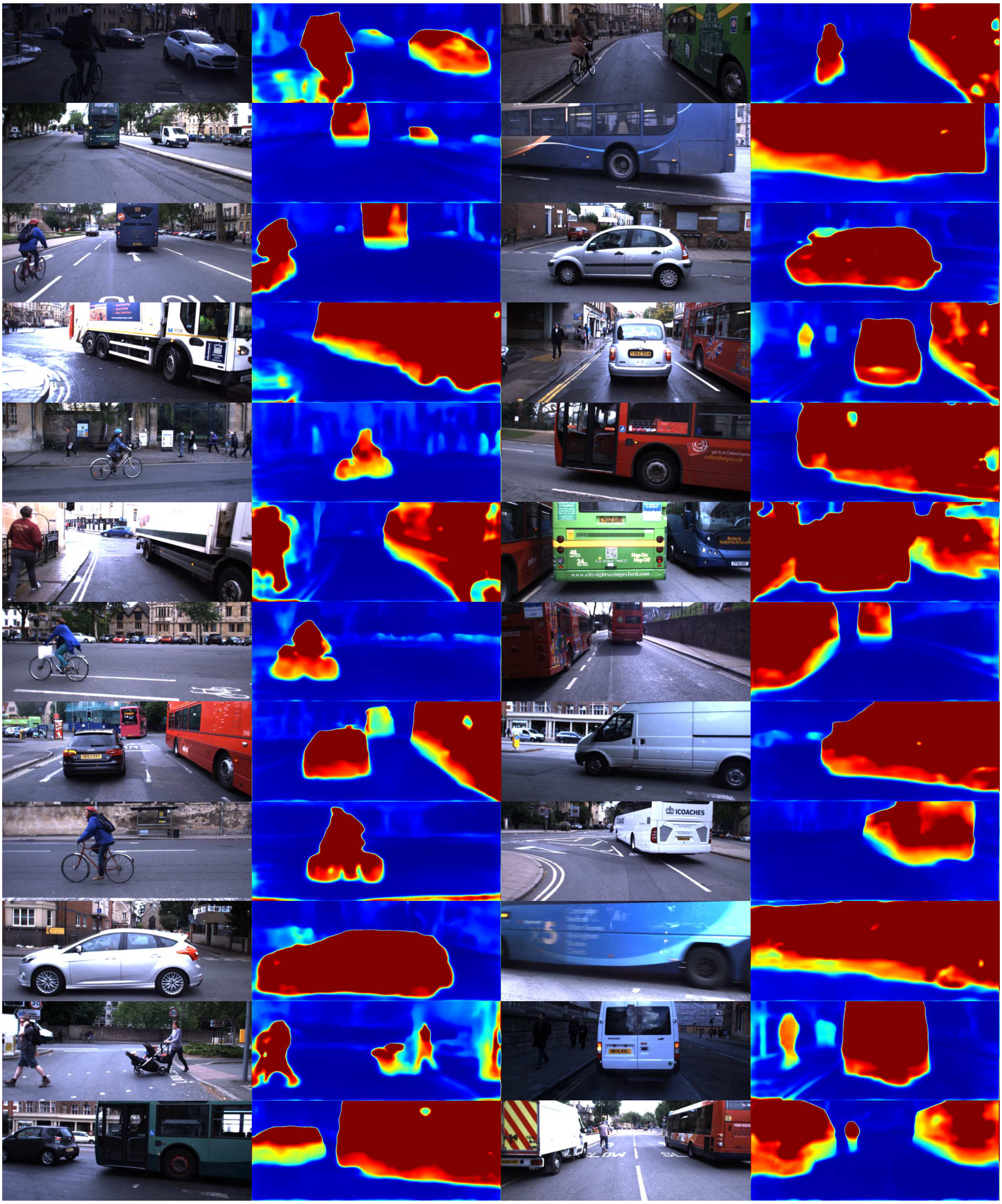


Fig. 7. Ephemerality masks produced in challenging urban environments. The masks reliably highlight a diverse range of dynamic objects (cars, buses, trucks, cyclists, pedestrians, strollers) with highly varied distances and orientations. Even buses and trucks that almost entirely obscure the camera image are successfully masked despite the lack of other scene context. Robust VO approaches that make use of the ephemerality mask can provide correct motion estimates even when more than 90% of the static scene is occluded by an independently moving object.

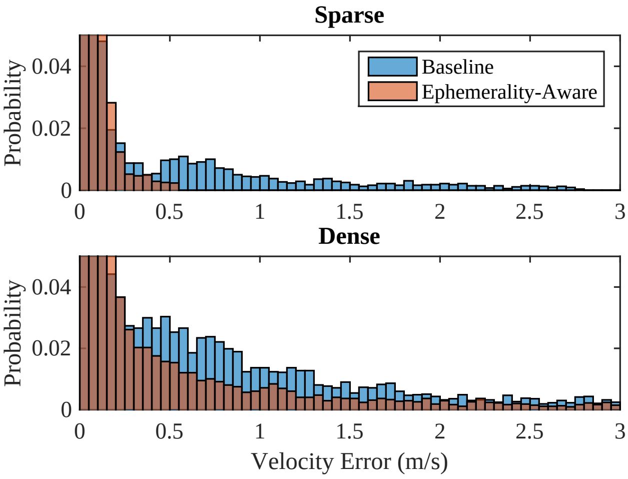


Fig. 8. Velocity estimation errors in the presence of distractors. The sparse ephemerality-aware approach significantly outperforms the baseline approach, producing far fewer outliers above 0.5 m/s. The dense ephemerality-aware approach does not perform as well, but still outperforms the baseline. The vertical axis is scaled to highlight the outliers.

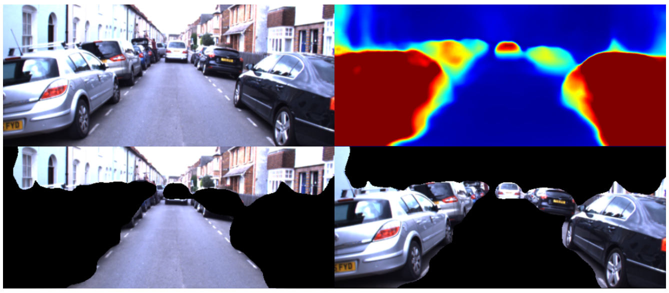


Fig. 9. Ephemerality masks are widely applicable for autonomous vehicles. In the above scene the ephemerality mask can be used to inform localisation against only the static scene (bottom left) whilst guiding object detection to only the ephemeral elements (bottom right).

VIII. Acknowledgements

This work was supported by the UK EPSRC Doctoral Training Partnership and Programme Grant EP/M019918/1.

References

[1] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[2] M. Piccardi, "Background subtraction techniques: a review," in *Systems, man and cybernetics, 2004 IEEE international conference on*, vol. 4. IEEE, 2004, pp. 3099–3104.

[3] S. Jeeva and M. Sivabalakrishnan, "Survey on background modeling and foreground detection for real time video surveillance," *Procedia Computer Science*, vol. 50, pp. 566–571, 2015.

[4] E. Hayman and J.-O. Eklundh, "Statistical background subtraction for a mobile observer," in *CVPR*. IEEE, 2003, p. 67.

[5] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1219–1225.

[6] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[9] A. Bak, S. Bouchafa, and D. Aubert, "Dynamic objects detection through visual odometry and stereo-vision: a study of inaccuracy and improvement sources," *Machine vision and applications*, pp. 1–17, 2014.

[10] S. Song and M. Chandraker, "Robust scale estimation in real-time monocular SFM for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1566–1573.

[11] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. Montiel, "Towards semantic SLAM using a monocular camera," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1277–1284.

[12] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1722–1729.

[13] R. Garg, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision*. Springer, 2016, pp. 740–756.

[14] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017.

[15] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "SfM-Net: learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[16] C. McManus, W. Churchill, A. Napier, B. Davis, and P. Newman, "Distraction suppression for vision-based pose estimation at city scales," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3762–3769.

[17] R. W. Wolcott and R. Eustice, "Probabilistic obstacle partitioning of monocular video for autonomous vehicles," in *BMVC*, 2016.

[18] C. Linegar, W. Churchill, and P. Newman, "Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 787–794.

[19] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 24.

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

[21] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[22] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *arXiv preprint arXiv:1705.07115*, 2017.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] H. Strasdat, J. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM," *Robotics: Science and Systems VI*, vol. 2, 2010.

[25] 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.

[26] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Computer Vision–ECCV 2006*, pp. 430–443, 2006.

[27] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," *Computer Vision–ECCV 2010*, pp. 778–792, 2010.

[28] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: real-time dense monocular SLAM with learned depth prediction," in *CVPR*, 2017.

[29] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1449–1456.

[30] 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.

[31] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3354–3361.

Statement of Authorship for joint/multi-authored papers for PGR thesis
To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Driven to Distraction: Self-Supervised Distractor Learning for Robust Monocular Visual Odometry in Urban Environments
Publication Status	Published
Publication Details	Dan Barnes, Will Maddern, Geoffrey Pascoe and Ingmar Posner Driven to Distraction: Self-Supervised Distractor Learning for Robust Monocular Visual Odometry in Urban Environments International Conference on Robotics and Automation (ICRA), 2017

Student Confirmation

Student Name:	Daniel Barnes		
Contribution to the Paper	My contributions to the paper were: Developed the idea behind the paper Collected and processed data Performed the experiments Wrote the paper with co-authors		
Signature	Daniel H Barnes	Date	18 / 11 / 2019

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Ingmar Posner		
Supervisor comments	The description above is accurate.		
Signature		Date	7/1/2020

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

Chapters 3 and 4 presented self-supervised approaches for urban behavioural planning and odometry systems in vision which require no manual annotation. However, routine conditions faced whilst driving (such fog, rain, snow and lens flare) present a serious challenge for vision-based systems. For safe autonomous navigation, a robot must operate consistently in any of these conditions, motivating our search for possible solutions. Millimetre-wave radar holds the promise of robustness to these conditions despite heterogeneous sensing artefacts. Therefore, with the goal of safer robot deployment in the real world, this modality presents a worthwhile avenue of research.

In this manuscript, we detail the Oxford Radar RobotCar Dataset (a radar extension to the Oxford RobotCar Dataset [9]) used throughout the remainder of this thesis. This dataset was also released to the public to advance research into the application of millimetre-wave FMCW scanning radar data to autonomous driving. The dataset covers 280 km of driving around Oxford, UK in January 2019, and includes ground truth RO for evaluating radar-based mapping and localisation research. Little data has been available in this modality until now, which presented a barrier for radar-based learning methods. By sharing this dataset with the community we hope to accelerate research in this domain.

In investigating radar data, we are able to demonstrate the benefits of large-scale self-supervision in a relatively under-utilised and under-researched sensor (in comparison to vision). Furthermore, the ground truth RO is automatically optimised with no manual intervention. Hence, this dataset forms an excellent platform for evaluating the self-supervised approaches to radar odometry and localisation in Chapters 6 and 7.

This manuscript has been accepted to the IEEE International Conference on Robotics and Automation (ICRA), 2020 [62]. A short video summary of the dataset can be found at: `https://youtu.be/rzDDDTNxhAo`

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`

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 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.

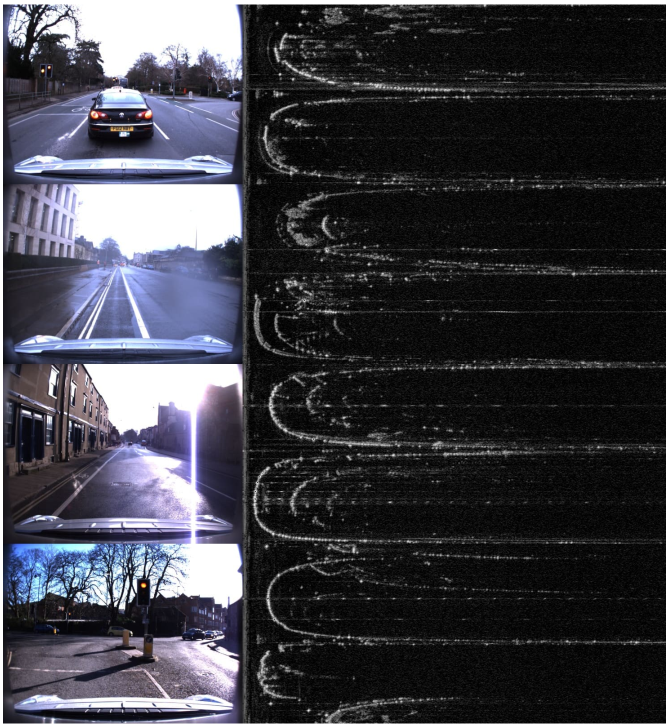


Fig. 1. The Oxford Radar RobotCar Dataset for complex and robust scene understanding with Millimetre-Wave FMCW scanning radar data. We collected 32 traversals of a central Oxford route with the Oxford RobotCar platform during the month of January, 2019. Despite weather conditions such as rain, direct 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.

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

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

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 $40\,\mathrm{m}$. 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, $4\,\mathrm{Hz}$, 400 measurements per rotation, $163\,\mathrm{m}$ range, $4.38\,\mathrm{cm}$ range resolution, $1.8°$ beamwidth
- 2 x Velodyne HDL-32E 3D LIDAR, 360° HFoV, 41.3° VFoV, 32 planes, $20\,\mathrm{Hz}$, $100\,\mathrm{m}$ range, $2\,\mathrm{cm}$ range resolution

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

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

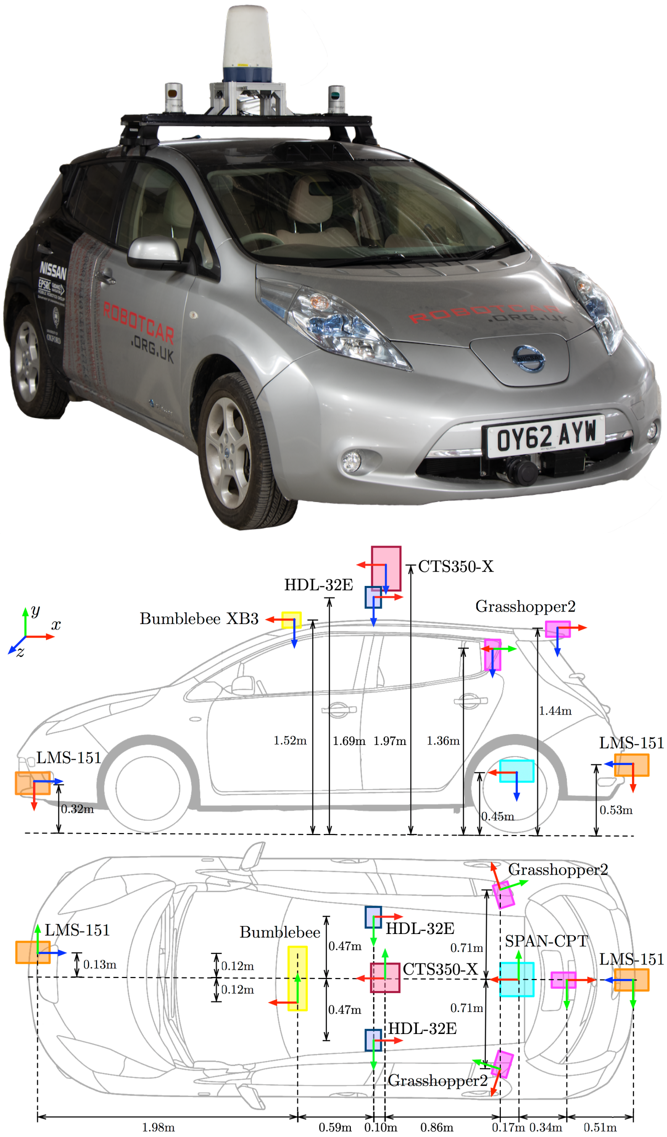


Fig. 2. The Radar RobotCar platform (top) and sensor location diagram (bottom) with the Navtech CTS350-X radar mounted in the centre. Coordinate frames show the origin and direction of each sensor mounted on the vehicle with the convention: x-forward (red), y-right (green), z-down (blue). Measurements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

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 $4.38\,\mathrm{cm}$ across 400 azimuths at a frequency of $4\,\mathrm{Hz}$ (corresponding to a maximum range of $163\,\mathrm{m}$ and $0.9°$ azimuth resolution). Other configurations of the Navtech CTS350-X are able to provide range in excess of $650\,\mathrm{m}$ or higher rotation frequencies. However, for this dataset shorter range, high resolution data was deemed most

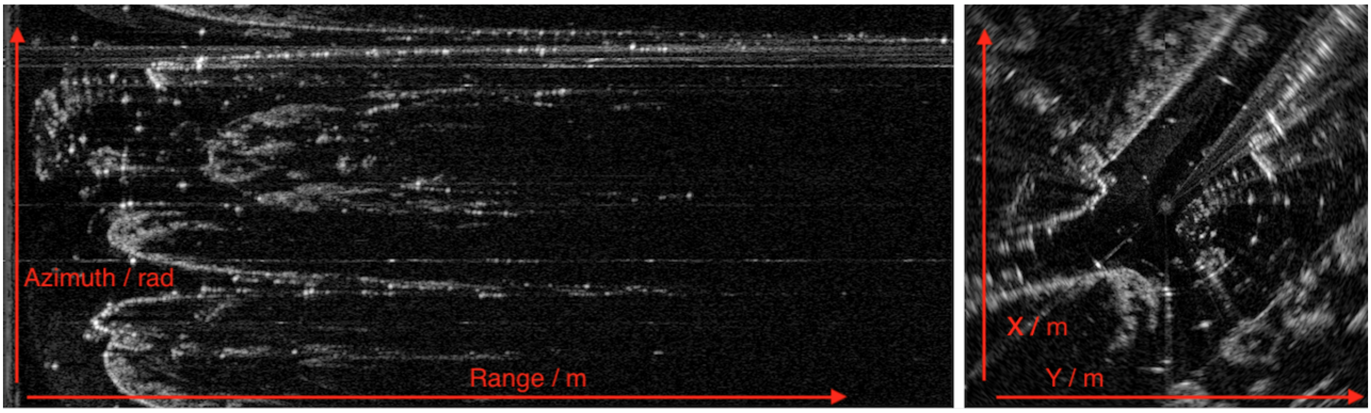


Fig. 3. Example sensor data from the Navtech CTS350-X radar. Raw radar power power returns in polar form (left) for a full sweep of $0 \rightarrow 2\pi$ over a range of $0 \rightarrow$ 163m and the corresponding scan in Cartesian form (right), with the vehicle in the center and axes from -50m → 50m. 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 $163\,\mathrm{m}$ 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 $76\,\mathrm{GHz}$ to $77\,\mathrm{GHz}$, ensuring consistent measurements through harsh local conditions such as dust, rain, and snow. The main beam spread is 1.8° between $-3\,\mathrm{dB}$ points horizontally and vertically; with an additional cosec squared fill-in beam pattern up to 40° below the horizontal which permits detection of objects beneath the main beam.

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.

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 $280\,\mathrm{km}$ 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 $4.7\,\mathrm{TB}$. 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 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,

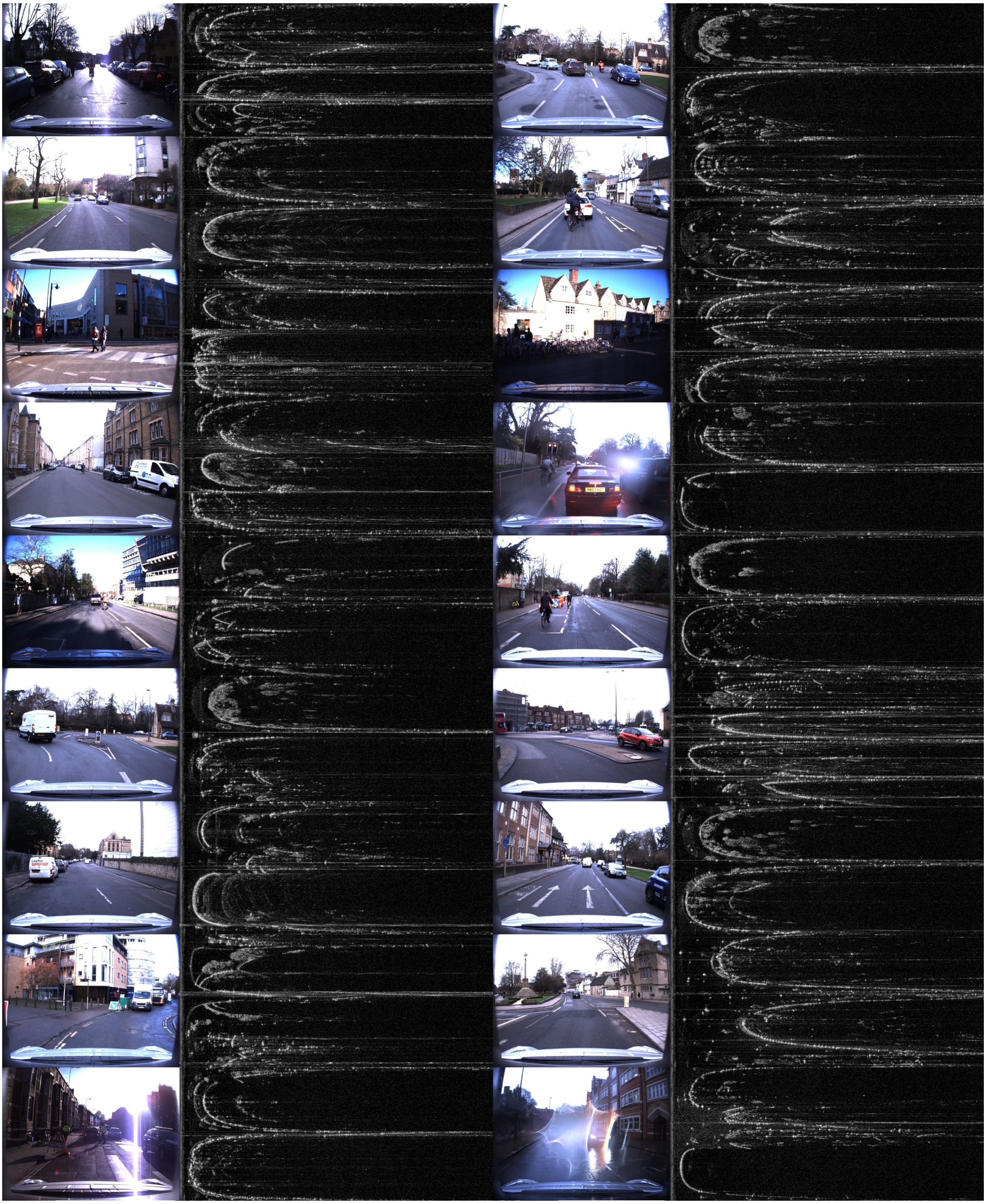
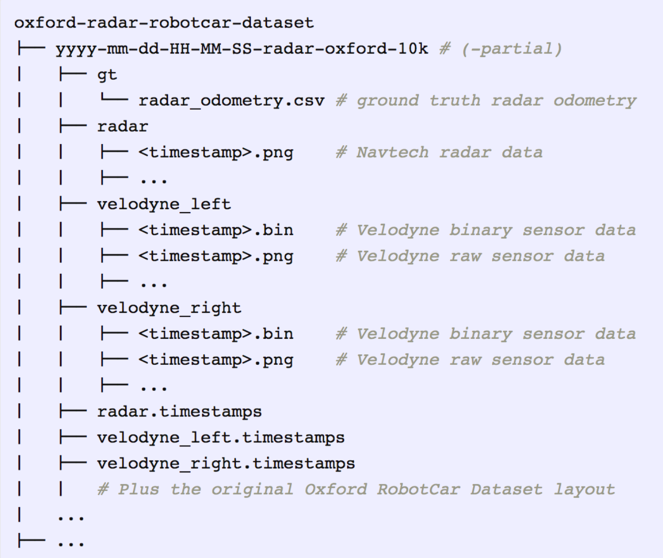


Fig. 4. Random pairs of Bumblebee XB3 images (left) with the temporally closest Navtech CTS350-X radar scan (right) from the Oxford Radar RobotCar Dataset, showing the challenging diversity of weather, lighting, and traffic conditions encountered during the period of data collection in Oxford, UK in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

  `angle = sweep_counter / 2800 * π`

- 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:

  `ranges (metres) = ranges_raw * 0.02`

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

  `angle = sweep_counter / 18000 * π`

- 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) x 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, α, β, γ) of the $SE(3)$ relative pose relating the two frames (where z, α, β 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.

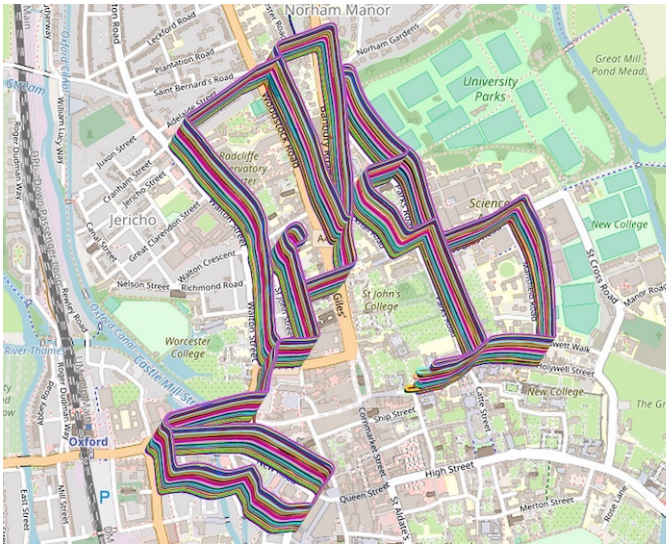


Fig. 6. Optimised radar odometry plotted on OpenStreetMap [19] for each of the 32 dataset traversals, where each run is offset for visualisation purposes. The trajectories were generated by optimising robust VO [20], visual loop closures [21], and GPS/INS as constraints. Map data copyrighted OpenStreetMap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

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[1]. 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 (µs), 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 $4.38\,\mathrm{cm}$.

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 $(\mathrm{uint8})$, azimuth angles (rad), and approximate timestamps (µs) as described previously.

The functions `VelodyneRawToPointcloud.m` and `velodyne_raw_to_pointcloud` take the ranges (m), intensities $(\mathrm{uint8})$, 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

[1] `github.com/ori-mrg/robotcar-dataset-sdk`

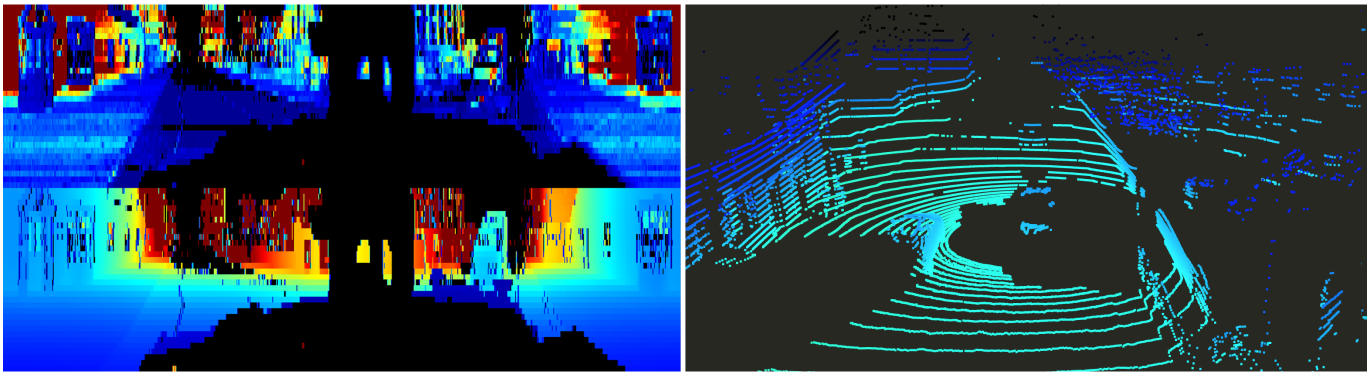


Fig. 7. Example sensor data from the Velodyne HDL-32E 3D LIDAR. A raw Velodyne scan (left) stores intensities (top) and ranges (bottom) for each azimuth (columns) whereas a binary scan stores the Cartesian pointcloud (right). Tools required to parse the data and perform the raw-to-pointcloud conversion 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. Scheding, 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. Gavrila, "An Extrinsic Calibration Tool for Lidar, Camera and Radar," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada, 2019*, 2019.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset
Publication Status	Submitted to the International Conference on Robotics and Automation (ICRA), 2020
Publication Details	Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman and Ingmar Posner The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset * International Conference on Robotics and Automation (ICRA), 2020

Student Confirmation

Student Name:	Daniel Barnes
Contribution to the Paper	My contributions to the paper were: Organised and coordinated dataset collection Processed raw sensor data Created dataset website and user management Wrote software development kit extensions to fully utilise dataset Wrote the paper with co-authors

Signature	Daniel H Barnes	Date	18 / 11 / 2019

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Ingmar Posner
Supervisor comments	The description above is accurate.

Signature		Date	7/1/2020

6 Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information

As shown in Chapter 5, millimetre-wave radar is a notoriously challenging sensor modality to process due to complex interactions between the sensor and the environment. However, if we are to utilise radar for safer deployment of robots in the real world (here as a redundant source of odometry), algorithms must adapt to handle the intricacies of radar rather than simply applying systems designed in other modalities.

As detailed in Section 2.3.2, performant RO systems exist but are either inaccurate in comparison to vision and lidar approaches, or too slow for real-time applications. Chapter 4 utilised a trained DNN to mask camera data to reduce the impact of sensing artefacts and distractor objects in VO. This presents an attractive proposition in radar, to learn modality-specific masks for improving RO without any manual supervision. However, the key issue with this approach is the human-imposed assumption of what is suitable for motion estimation.

In this publication, we present a model-free, self-supervised approach to mask sensing artefacts and distractor objects in radar data, with the goal of improving RO. By embedding a differentiable pose estimator inside the RO system, a masking network optimal for our specific formulation is learnt from pose error alone, rather than imposing any assumptions on what data to retain. In doing so, the masking network naturally learns to suppress sensing artefacts and distractor objects which would otherwise degrade pose estimation performance, whilst keeping static objects such as walls and buildings – all without any manual annotation. At run-time, we predict state-of-the-art RO at well over real-time speeds, as well as interpretable artefact and distraction-free radar scans. To further validate our approach for RO in urban environments, in extension experiments we quantitatively compare against a classical baseline and an alternative spectral pose matching formulation.

This manuscript was presented at the Conference on Robot Learning (CoRL), 2019 [63]. A video summary of the publication can be found at: `https://youtu.be/eG4Q-j3_6dk`

Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information

Dan Barnes, Rob Weston, Ingmar Posner
Applied AI Lab, University of Oxford
{dbarnes, robw, ingmar}@robots.ox.ac.uk

Abstract: This paper presents an end-to-end radar odometry system which delivers robust, real-time pose estimates based on a learned embedding space free of sensing artefacts and distractor objects. The system deploys a fully differentiable, correlation-based radar matching approach. This provides the same level of interpretability as established scan-matching methods and allows for a principled derivation of uncertainty estimates. The system is trained in a (self-)supervised way using only previously obtained pose information as a training signal. Using 280km of urban driving data, we demonstrate that our approach outperforms the previous state-of-the-art in radar odometry by reducing errors by up 68% whilst running an order of magnitude faster.

Keywords: Perception, Radar, Odometry, Localisation, Deep Learning, Autonomous Driving

1 Introduction

Robust ego-motion estimation and localisation are established cornerstones of autonomy. Emerging commercial needs as well as otherwise ambitious deployment scenarios require our robots to operate in ever more complex, unstructured environments and in conditions distinctly unfavourable for typical go-to sensors such as vision and lidar. Our robots now need to see further, through fog, rain and snow, despite lens flare or when directly facing the sun. Radar holds the promise of remedying many of these shortcomings. However, it is also a notoriously challenging sensing modality: radar applications are typically blighted by heterogeneous noise artefacts such as ghost objects, phase noise, speckle and saturation. In response, previous approaches to utilising radar for robot navigation have often tried to manually extract features from noise corrupted radar scans, commonly relying on simplifying assumptions on the distribution of power returns [1], manually designed heuristics [2], or features designed for different modalities [3, 4]. Nevertheless, the recent seminal work by Cen et al. [2] has firmly established radar as a feasible alternative to complement existing navigation approaches when it comes to ego-motion estimation.

Beyond the basic methodology for pose estimation, the prevalence of vision- and lidar-based approaches in this space has given rise to a number of useful methods beyond those currently utilised for radar. State-of-the-art visual odometry, for example, leverages learnt feature representations [5] as well as attention masks filtering out potentially distracting objects [6]. Lidar-based methods using correlative scan matching [7] typically achieve highly accurate and intuitively interpretable results.

Inspired by this prior art, the aim of our work is to provide a robust radar odometry system which is largely unencumbered by either the typical radar artefacts or by the presence of potentially distracting objects. Our system is explicitly designed to provide *robust*, *efficient* and *interpretable* motion estimates. To achieve this we leverage a deep neural network to learn an essentially artefact and distraction free embedding space which is used to perform efficient correlative matching between consecutive radar scans. Our matching formulation is fully differentiable, allowing us to explicitly learn a representation suitable for accurate pose prediction. The correlative scan matching approach further allows our system to efficiently provide principled uncertainty estimates.

Training our network on over 186,000 examples generated from 216km's of driving, we outperform the previous state of the art in challenging urban environments, reducing errors by over 68% and running an order of magnitude faster. Furthermore, our pose ground truth is gathered in a self-supervised manner, automatically optimising odometry, loop closure, and location constraints, enabling us to adapt to new locations and sensor configurations with no manual labelling effort.

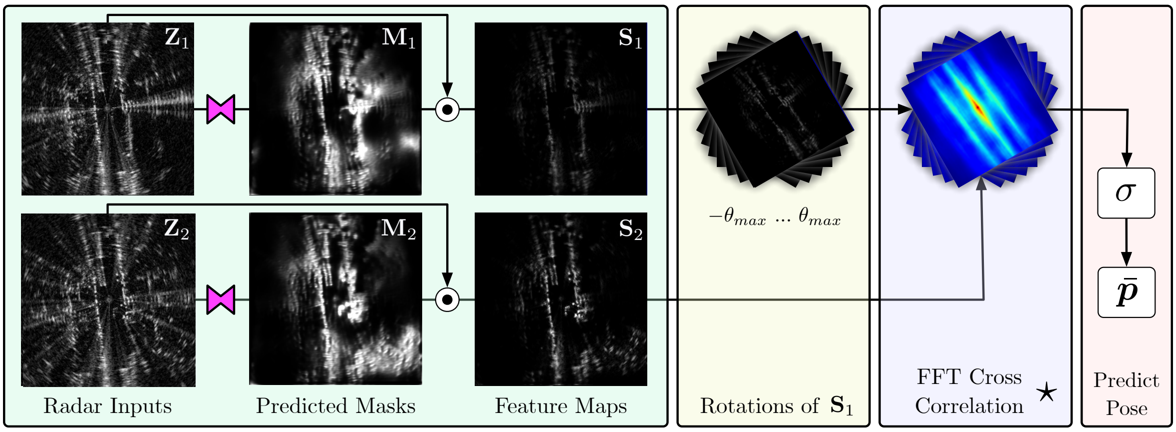


Figure 1: Using masked correlative scan matching to find the optimum pose. Each radar scan is passed to a CNN (pink) in order to generate a mask which is subsequently applied to the input radar scan generating sensor artefact and distractor free representations $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. We then calculate the 2D correlation between $\boldsymbol{S}_2$ and rotated copies of $\boldsymbol{S}_1$ using the fft2d in order to generate the correlation volume $\boldsymbol{C}$. Finally we perform a softargmax operation in order to retrieve the predicted pose. Crucially this pipeline is fully differentiable allowing us to learn the filter masks end to end. A video summary of our approach can be found at: `https://youtu.be/eG4Q-j3_6dk`

2 Related Work

Compared to other sensing modalities such as vision or lidar, radar has received relatively little attention in the context of robot navigation. Prior art in this area largely deploys a more traditional processing pipeline consisting of separate feature extraction, data association and loss minimisation steps, for example using the Iterative Closest Point (ICP) algorithm [8, 9]. For feature extraction some works deploy approaches developed in vision, such as SIFT and SURF [3, 4], others more bespoke methods such as CFAR filtering [10, 1], temporal-space continuity modelling [11, 12], and grid-map features such as Binary Annular Statistics Descriptor (BASD) [13]. Most recently the authors of [2] find point correspondences between point features extracted from raw scans using a shape similarity metric. The final pose is then found by minimising the mean squared error between point correspondences in close to real time.

By making use the Fourier transform correlation-based approaches are in contrast able to perform a dense search over possible point correspondences [14] yielding intuitively interpretable results. Similar approaches have also been applied successfully to lidar scan matching utilising efficient GPU implementations [15] [7]. In comparison to ICP, correlation-based methods have been shown to be significantly more robust to noise in pose initialisation [15]. While robustness and interpretability are desirable, correlation-based methods operate on the assumption that the power returns from a particular location are stationary over time so that a correlation operation produces meaningful results. In reality, this is often not the case – for example when dynamic objects are present in the scene. This problem is particularly pronounced in radar data due to the prevalence of noise artefacts.

Visual odometry systems, in contrast to radar-based ones, have a significant track record of successful application in robotics and beyond. While traditional processing pipelines similar to the one outlined for radar above have been widely deployed in this context (e.g. [16]) there has recently been significant interest in moving away from separate processing steps towards end-to-end approaches. Typically, a neural network is used to regress to a predicted pose directly from consecutive camera images, learning the relationship between features and point correspondences in an integrated manner (e.g. [5, 17]). In [18] the authors extend this approach by learning to predict the optimum pose from stereo images alone. As in many related fields, these end-to-end approaches demonstrate the potential for learning representations generally useful for odometry prediction. However, this comes at the expense of entangling feature representation and data association, which makes the resulting system significantly less interpretable. In contrast, the authors of [19] propose to learn a feature embedding for localising online lidar sweeps into a previously known map, whilst maintaining the interpretability, of a conventional correlative scan matching approach.

Due to the ubiquitous nature of vision-based systems researchers have also addressed challenges beyond the basic pose estimation task such as suppressing noise sources inherent in individual scenes. For example, both [20] and [6] try to mask areas of an image where non-stationary features might be found, which could corrupt the odometry estimate. Of particular relevance is [6], where a deep neural network is trained using data from other parts of the autonomous system in order to predict human interpretable ephemerality masks indicating the presence of distractor objects in a scene.

Given the large body of evidence that end-to-end approaches tend to outperform more traditional, hand-engineered processing pipelines it is tempting to conclude that our goal here is simply to deploy a deep network to radar odometry. And we do indeed leverage deep learning in our system. However, in doing so we are cognisant that we desire a system which ideally exploits the power of representation learning offered by end-to-end approaches while at the same time leveraging the efficiency, robustness and interpretability offered by correlation-based methods. Thus, inspired by [6] and similar to [19], we deploy a correlation-based matching method as part of an end-to-end system which learns a radar embedding used to produce largely artefact and distraction-free representations optimised for pose prediction. Both the masks obtained as well as the cost-volumes considered remain as interpretable as more traditional approaches.

3 Deep Correlative Scan Matching with Learnt Feature Embeddings

Given two consecutive radar observations $(\boldsymbol{Z}_t, \boldsymbol{Z}_{t-1})$ we wish to determine the relative pose $[\boldsymbol{R}|\boldsymbol{t}] \in \mathbb{SE}(2)$ giving the transformation between the two co-ordinate systems at each time step. In achieving this we aim to harness the efficiency, interpretability and robustness of correlative scan matching assuming that the power returned from each world location is independent of the co-ordinate system it was sensed in. In reality the power returns generated from real world scenes are far from stationary, as dynamic objects move into and out of the field of view of the sensor and pertinent, random noise artefacts obscure the true power returns, limiting the performance of an out-of-the-box correlative scan matching system applied to radar data.

To address this, and inspired by the recent successes of learnt masking for pose prediction in vision [6], we instead perform correlative scan matching over a learnt feature embedding, utilising a deep, fully convolutional network to mask each radar scan as illustrated in Figure 1 (described in Section 3.1). Through this approach we are able to harness the power of deep representation learning whilst ensuring the feature representation remains interpretable through the geometrical constraints imposed by the use of a correlative scan matching procedure. Crucially, we train our network by supervising pose prediction *directly*. In doing so, our network naturally learns to attenuate distractor objects such as moving vehicles and sensor noise as they degrade pose estimation accuracy, whilst preserving features which are likely to be consistent between scans such as walls and buildings. This leads to a 68% reduction in errors over the current state-of-the-art whilst, by making use of efficient correlation computations using the Fast Fourier Transform (FFT), running an order of magnitude faster.

Even in the limit of perfectly stationary power returns, uncertainty in our pose prediction still emanates from pathological solutions arising from the underlying scene topology. In Section 5.2 we show how we are additionally able to quantify the uncertainty in our pose prediction, further aiding the interpretability of our system.

3.1 Correlative Scan Matching with Learnt Feature Embeddings

Let $(\boldsymbol{Z}_{t-1}, \boldsymbol{Z}_t) \in [0,1]^{W \times H}$ denote consecutive observations made by single sweeps of the radar sensor, converted to Cartesian co-ordinates such that $\boldsymbol{Z}_t^{u,v}$ gives the power return at Cartesian co-ordinate (x, y) at time t. Let $\boldsymbol{p} = [\Delta x, \Delta y, \Delta\theta]^T$ denote the parameters of the relative pose $[\boldsymbol{R}|\boldsymbol{t}] \in \mathbb{SE}(2)$ between the co-ordinate frames at $t-1$ and t. We aim to predict the optimum pose from consecutive radar observations harnessing the efficiency, interpretability and robustness of correlative scan matching,

$$\bar{\boldsymbol{p}} = \operatorname*{arg\,max}_{\boldsymbol{p} \in \mathbb{SE}(2)} \boldsymbol{Z}_t \star \boldsymbol{Z}_{t-1} \tag{1}$$

where $\boldsymbol{Z}_t \star \boldsymbol{Z}_{t-1}$ is defined as the *discrete cross correlation* between $\boldsymbol{Z}_t$ (after being warped by the pose $\boldsymbol{p}$) and $\boldsymbol{Z}_{t-1}$.

In order to solve for the predicted pose $\bar{\boldsymbol{p}}$ we consider a brute force approach: we discretise our search search space, calculating the cross correlation score for each pose on a regular grid of pose candidates before utilising a soft-argmax operation to solve for the optimum pose to sub-grid resolution accuracy. This is achieved efficiently using Algorithms 2 and 3. By utilising bi-linear interpolation for all re-size and rotation operations, and computing the cross-correlation using the highly efficient 2D Fast Fourier Transform, we are able to search for the optimum pose over a large search area, efficiently solving (1) whilst still maintaining end-to-end differentiability.

Central to this approach is an assumption that the power returned from each world location is independent of the co-ordinate system it was sensed in. This assumption rarely holds in practice. Random noise artefacts, dynamic objects and changing scene occlusion cause fluctuations in the power field, degrading the accuracy of conventional correlation-based approaches applied to radar. To counter this, we propose to learn a feature representation $\boldsymbol{S}$ specifically optimised for correlative scan matching by filtering each radar scan $\boldsymbol{S} = \boldsymbol{M} \odot \boldsymbol{Z}$ with a mask $\boldsymbol{M} = f_\alpha(\boldsymbol{Z})$ generated by a neural neural network f_α (where $\odot$ denotes Hadamard product). By limiting each element of the mask to $[0, 1]$ (using an element wise sigmoid), the network is able to learn to filter out distractor objects and noise in each sensor observation, before correlative scan matching is applied to find the optimum pose. By leveraging the differentiability of our approach for predicting $\bar{\boldsymbol{p}}$, we are able to use Algorithm 1, to learn a radar feature embedding specifically optimised for correlative scan matching by minimising the Mean Squared Error (MSE) over the training set, $\mathcal{D} = \{(\boldsymbol{Z}_t, \boldsymbol{Z}_{t-1}, \boldsymbol{p})^n\}_{n=1}^N$,

$$\alpha^* = \arg\min_\alpha \mathbb{E}_{\boldsymbol{p}\sim\mathcal{D}}\left[||\bar{\boldsymbol{p}} - \boldsymbol{p}||^2\right] \tag{2}$$

to update our network parameters α using conventional stochastic gradient descent based optimisers.

3.2 Pose Uncertainty Estimation

Pathological solutions arising from the underlying scene topology increase the uncertainty in our pose prediction even in the case of perfectly stationary power returns. In the real world identifying such cases is important in order to ensure robust operation. To this end, our approach also affords us a principled mechanism to estimate the uncertainty in each element of the predicted pose.

In performing the soft-argmax operation, we first apply a temperature controlled softmax over the correlation scores for each candidate pose, to give weights $\boldsymbol{\omega} = \text{Softmax}(\beta \boldsymbol{C})$, interpreted as the probability that each pose candidate is optimum. Assuming that our predicted pose is Gaussian distributed we can quantify the uncertainty in each pose prediction by using the weights $\boldsymbol{\omega}$ to predict both the mean pose $\bar{\boldsymbol{p}}$ *and* the predicted co-variance $\bar{\boldsymbol{\Sigma}}$,

$$\bar{\boldsymbol{p}} = \sum_s \omega_s \boldsymbol{p}_s \quad \bar{\boldsymbol{\Sigma}} = \sum_s \omega_s \boldsymbol{p}_s \boldsymbol{p}_s^T - \bar{\boldsymbol{p}}\bar{\boldsymbol{p}}^T \quad p(\boldsymbol{p}|\boldsymbol{S}_t) \approx \mathcal{N}(\boldsymbol{p}|\bar{\boldsymbol{p}}, \bar{\boldsymbol{\Sigma}}) \tag{3}$$

where we sum over all pose candidates. The softmax temperature parameter β plays an important role here: for high β our system is biased to the pose candidate with highest correlation and a low co-variance, whilst for low β to a weighted mean over a greater number of pose candidates and high co-variance.

Algorithm 1: Training

```
Input:
$\mathcal{D}$ // Dataset
$\boldsymbol{r}$ // Search Region giving min and
    max range in $\Delta x, \Delta y, \Delta\theta$
$\boldsymbol{\delta}$ // Grid resolution in each
    dimension $\delta_x, \delta_y, \delta_\theta$
$\beta$ // Softmax Temperature Parameter
$\epsilon$ // Learning Rate
$\alpha$ // Initial Network Parameters
$\boldsymbol{G}_{xy\theta} = MeshGrid(\boldsymbol{r}, \boldsymbol{\delta})$
while not converged do
    $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{p} \leftarrow Sample(\mathcal{D})$
    $\boldsymbol{M}_1, \boldsymbol{M}_2 \leftarrow f_\alpha(\boldsymbol{Z}_1), f_\alpha(\boldsymbol{Z}_2)$
    $\boldsymbol{S}_1, \boldsymbol{S}_2 \leftarrow \boldsymbol{M}_1 \odot \boldsymbol{Z}_1, \boldsymbol{M}_2 \odot \boldsymbol{Z}_2$
    $\boldsymbol{C} \leftarrow GetCorrelation(\boldsymbol{G}_{xy\theta}, \boldsymbol{S}_1, \boldsymbol{S}_2)$
    $\bar{\boldsymbol{p}} \leftarrow SoftArgMax(\boldsymbol{G}_{xy\theta}, \boldsymbol{C}, \beta)$
    $\alpha \leftarrow \alpha - \epsilon\nabla_\alpha \mathcal{L}(\bar{\boldsymbol{p}}; \boldsymbol{p})$
end
```

Algorithm 2: Correlation

```
function GetCorrelation($\boldsymbol{G}_{xy\theta}$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$) :
    $n_x, n_y, n_\theta \leftarrow Shape(\boldsymbol{G}_{xy\theta})$
    $\boldsymbol{C} = Zeros([n_x, n_y, n_\theta])$
    $\boldsymbol{G}_{xy}, \boldsymbol{G}_\theta \leftarrow \boldsymbol{G}_{xy\theta}$
    $\boldsymbol{X}_1, \boldsymbol{X}_2 \leftarrow Resize(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{G}_{xy})$
    par for $i \leftarrow 1$ to $n_\theta$ :
        $\boldsymbol{X}_1^{\mathsf{R}} \leftarrow Rotate(\boldsymbol{X}_1, \boldsymbol{G}_\theta[i])$
        $\boldsymbol{C}[:, :, i] \leftarrow$
          $\mathsf{fft2d}^{-1}\big(\mathsf{fft2d}(\boldsymbol{X}_1^{\mathsf{R}}) \odot \mathsf{fft2d}(\boldsymbol{X}_2^{\mathsf{C}})\big)$
    return $\boldsymbol{C}$
```

Algorithm 3: Soft Arg Max

```
function SoftArgMax($\boldsymbol{G}_{xy\theta}$, $\boldsymbol{C}$, $\beta$) :
    $\boldsymbol{\omega} \leftarrow Softmax(\beta\boldsymbol{C})$
    $\boldsymbol{G}_x, \boldsymbol{G}_y, \boldsymbol{G}_\theta \leftarrow \boldsymbol{G}_{xy\theta}$
    $\Delta x \leftarrow \sum_{i,j,k} \big(\boldsymbol{\omega} \odot \boldsymbol{G}_x\big)[i, j, k]$
    $\Delta y \leftarrow \sum_{i,j,k} \big(\boldsymbol{\omega} \odot \boldsymbol{G}_y\big)[i, j, k]$
    $\Delta\theta \rightarrow \sum_{i,j,k} \big(\boldsymbol{\omega} \odot \boldsymbol{G}_\theta\big)[i, j, k]$
    return$[\Delta x, \Delta y, \Delta\theta]$
```

4 Experimental Setup

4.1 Dataset

To evaluate our approach we use the recently released Oxford Radar RobotCar Dataset [21], a radar extension to the Oxford RobotCar Datsset [22], which provides Navtech CTS350-X radar data as well as ground truth poses. The Navtech CTS350-X is a Frequency Modulated Continuous Wave (FMCW) scanning radar without doppler information, configured to return 3768 power readings at a resolution of 4.32cm across 400 azimuths at a frequency of 4Hz (corresponding to a maximum range of 163m). The beam spread is 2 degrees in azimuth and 25 degrees in elevation with a cosec squared beam pattern. We randomly split the traversals into training (80%) and evaluation (20%) partitions. We additionally run spatial cross validation experiments, where each split occupies a different real world region of the dataset. Further information on these results and the dataset can be found in the appendix B.2, C.1.

To validate the advantages of learning masks directly from pose supervision we compare against supervising the learnt masks directly on the proxy task of predicting temporally static occupied cells. Training data for this is generated using a similar approach to [6]. For each radar scan we warp the nearest radar sensor observation from each training traversal into the current pose before applying a static power threshold. We then form a 2D histogram counting the number of thresholded power returns that fall in each Cartesian grid cell. Any grid cell with more than 9 consistent observations is assumed to be temporally stable and is labelled with a 1, whilst every other cell is set to 0. This is repeated for every pose in every dataset. Examples of the masks generated by this approach can be found in the appendix B.1.

4.2 Network Architecture and Training

In all experiments we use a U-Net style architecture [23] in which we encode the input tensor through the repeated application of two convolutional layers (filter size 3x3) with ReLU activations before a max pooling operation. After each max pool the width and height of the tensor are reduced by a factor of 2 whilst the number of features is doubled, starting from 8 at the input to 256 at the bottleneck of the network (corresponding to 5 max pools). The feature tensor is then converted back to the original shape by the decoder through the application of bilinear upsampling followed by two convolutional layers increasing the width and height and decreasing the feature channels by a factor of 2. Skip connections at each level are implemented allowing information to flow from encoder to decoder by stacking each representation with the output from the bilinear upsampling layer in each case. The final convolutional layer has a single output channel with a sigmoid activation to limit the range to $[0, 1]$. We experiment with learning to mask both Cartesian and Polar radar representations, as well as both *single* and *dual* configurations. In the dual case radar observations are concatenated and passed as a single input producing two masks (instead of one) at the output. An architecture diagram can be found in the appendix A.1. In all cases we consider a search region of $[-50\text{m}, 50\text{m}]$ in Δx and Δy and $[-\pi/12, \pi/12]$ in $\Delta\theta$. We experiment with the three grid resolutions $[0.2\text{m}, 0.4\text{m}, 0.8\text{m}]$ for δ_x and δ_y whilst fixing δ_θ to $\pi/360$.

Our network is implemented in Tensorflow [24] and trained using the Adam Optimiser [25] (learning rate $1\text{e}-5$ and batch size 5) until the loss on a small validation set is a minimum. When training our network with pose supervision we minimise the loss proposed in (2). We performed a grid search over the optimum value of β and found setting it to 1 gave good performance.

4.3 Evaluation Metrics and Baselines

Our primary baseline is the current state of art for radar odometry [2] (implemented in C++) in which the authors extract point features from consecutive radar scans before scan matching using a global shape similarity score and refining by minimising mean squared error. Our radar was set to a range resolution of 4.32cm, whilst the original algorithm was developed for a 17.28cm resolution. As such we compare against [2] with full resolution radar scans and downsampled (with max pooling) to 17.28cm. For context we also provide visual odometry estimates (as in [2]). To assess the benefits of learning feature masks specifically optimised for pose prediction, we benchmark against scan matching on the raw radar scans without masking, as well as using the method proposed in [6] with mask labels generated as described in Section 4.1. In this setup, we supervise (using a binary cross entropy loss) the learnt masks directly (instead of supervising pose prediction). We also benchmark against taking an off the shelf deep odometry model and training this for the task of radar pose prediction. Specifically we use the UnDeepVO model proposed in [18].

For all evaluations we follow the KITTI odometry benchmark [26]. For each 100m offset up to 800m, we calculate the average residual translational and angular error for every example in the datastet normalising by the distance travelled. Finally, we average these values. Due to highly skewed error distributions we report Inter Quartile Range (IQR) for each method instead of the standard deviation. All timing statistics are calculated using a 2.7 GHz 12-Core Intel Xeon E5 CPU and Nvidia Titan Xp GPU by averaging across 1000 predictions.

4.4 Uncertainty Evaluation

To assess the quality of the uncertainty predicted by our approach we observe that if our pose distribution is Gaussian than the Mahalanobis error

$$d^2 = (\boldsymbol{p} - \bar{\boldsymbol{p}})^T \bar{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{p} - \bar{\boldsymbol{p}}), \quad d^2 \sim \chi^2(3) \tag{4}$$

should be chi-squared distributed with degrees of freedom equal to the state dimensionality of $\boldsymbol{p}$ (in this case three). As the mean of a chi-squared distribution is equal to the distributions degrees of freedom, by averaging the mean Mahalanobis distance over the test dataset $\bar{d^2} = \frac{1}{N}\sum_n d_n^2$ we can assess to what degree the uncertainties predicted by our approach are calibrated to the test errors [7]. Specifically, if $\bar{d^2} \ll 3$ then our model is overly conservative in its predictions whilst if $\bar{d^2} \gg 3$ it is overly confident. In Section 5.2 we use this result to tune the temperature parameter β to provide us with realistic uncertainties, that are calibrated to the true errors in our system.

5 Results

In this section we evaluate the performance of our approach. We find by utilising correlative scan matching in combination with a learnt radar feature embedding we are able to significantly outperform the previous state of art in both prediction performance and speed. Additionally, we show how, by tuning the temperature parameter of the softargmax, we are able to predict realistic and calibrated uncertainties, further increasing the interpretability of our system and allowing us to identify pathological cases, crucial for robust operation in the real world.

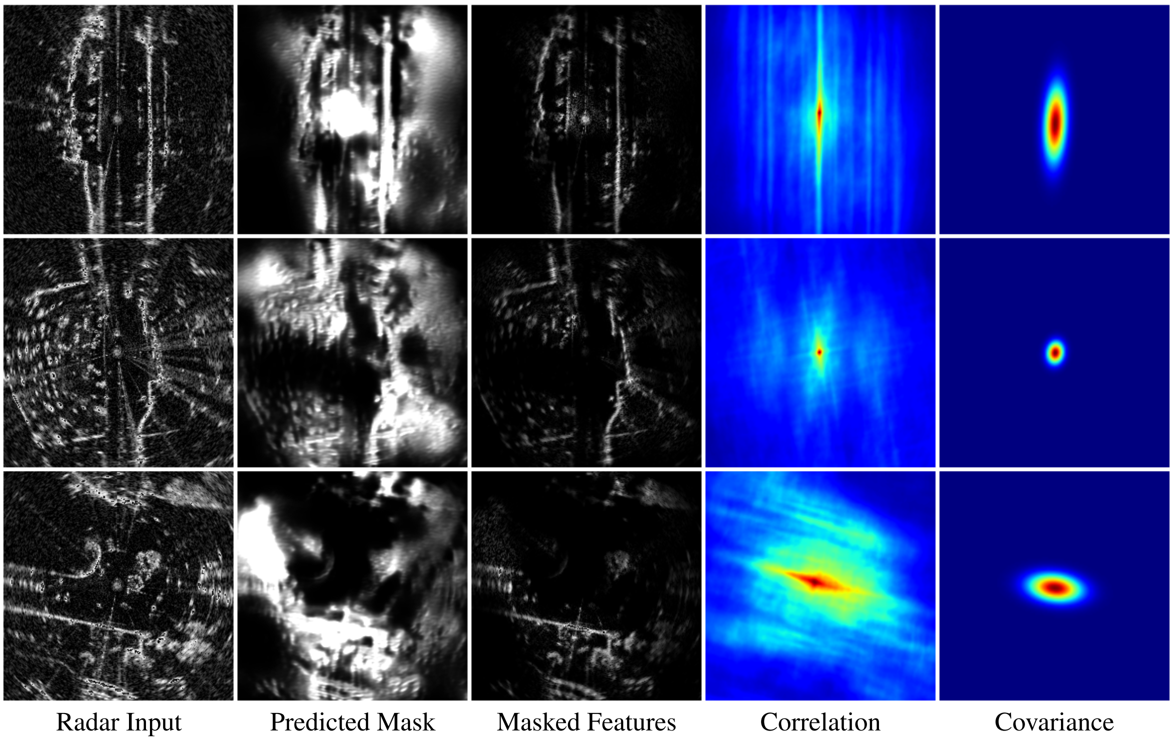


Figure 2: Qualitative examples generated from our best performing model. Our network learns to mask out noise and distractor objects whilst preserving temporally consistent features such as walls, well suited for pose prediction. Predicted co-variance is high for pathological solutions arising through a lack of constraints in the x-direction (top), whilst stationary well-constrained scenes result in low co-variance (middle). Motion blur increases the uncertainty due to ambiguous point correspondence (bottom). Further examples can be found in Figure 8 in the appendix.

	Resolution	Translational error (%)		Rotational error (deg/m)		Runtime (s)	
Benchmarks	(m/pixel)	Mean	IQR	Mean	IQR	Mean	Std.
RO Cen Full Resolution [2]	0.0432	8.4730	5.7873	0.0236	0.0181	*0.3059*	0.0218
RO Cen Equiv. Resolution [2]	0.1752	*3.7168*	3.4190	*0.0095*	0.0095	2.9036	0.5263
Raw Scan	0.2	8.3778	7.9921	0.0271	0.0274	0.0886	0.0006
Supervised Masks Polar	0.2	5.9285	5.6822	0.0194	0.0197	0.0593	0.0014
Supervised Masks Cart	0.2	5.4827	5.2725	0.0180	0.0186	0.0485	0.0013
Adapted Deep VO Cart [18]	0.2	4.7683	3.9256	0.0141	0.0128	0.0060	0.0003
Adapted Deep VO Polar [18]	-	9.3228	8.3112	0.0293	0.0277	0.0093	0.0002
Visual Odometry [16]	-	3.9802	2.2324	0.0102	0.0065	0.0062	0.0003
Ours							
Polar	0.8	2.4960	2.1108	0.0068	0.0052	0.0222	0.0013
	0.4	1.6486	1.3546	0.0044	0.0033	0.0294	0.0012
	0.2	1.3634	1.1434	0.0036	0.0027	0.0593	0.0014
Cartesian	0.8	2.4044	2.0872	0.0065	0.0047	0.0113	0.0012
	0.4	1.5893	1.3059	0.0044	0.0035	0.0169	0.0012
	0.2	1.1721	0.9420	0.0031	0.0022	0.0485	0.0013
Dual Polar	0.8	2.5762	2.0686	0.0072	0.0055	0.0121	0.0003
	0.4	2.1604	1.9600	0.0067	0.0053	0.0253	0.0006
	0.2	1.2621	1.1075	0.0036	0.0029	0.0785	0.0007
Dual Cart	0.8	2.7008	2.2430	0.0076	0.0054	**0.0088**	0.0007
	0.4	1.7979	1.4921	0.0047	0.0036	0.0194	0.0010
	0.2	**1.1627**	0.9693	**0.0030**	0.0030	0.0747	0.0005

Table 1: Odometry estimation and timing results. Here “RO Cen” [2] is our primary benchmark reported for 0.04m (full resolution) and, by downsampling, 0.17m (equivalent resolution for which the approach was originally developed). For comparison we also provide performance results for correlative scan matching on the *raw* power returns, for mask supervision (instead of supervising the predicted pose directly), and adapting the deep VO network proposed in [18], alongside visual odometry [16] for context. All baselines performed best at 0.2 m/pixel resolution where applicable and the rest are omitted for clarity. We experiment with both polar and Cartesian network inputs at multiple resolutions. Our approach outperforms the current state of the art, “RO Cen” (italics), for all configurations of Cartesian / polar inputs and independent / dual masking at all resolutions. Our best performing models in terms of speed and odometry performance are marked in bold.

5.1 Odometry Performance

Table 1 gives our prediction and timing results. We experiment with both Cartesian and Polar inputs to the masking network (converting the latter to Cartesian co-ordinates before correlative scan matching), as well as experimenting with single and dual configurations as detailed in Section 4.2.

At all resolutions and configurations we beat the current state of the art with our best model reducing errors by 68% in both translation and rotation, whilst running over 4 times faster. Our fastest performing model runs at over 100Hz whilst still reducing errors on the state of the art by 28% in translational and 20% in rotational error (further results exploring the accuracy-speed trade off can be found in A.2). We find that Cartesian network inputs typically outperform Polar (presumably because correlative scan matching is performed in Cartesian space). Dual input configurations also typically outperform passing single sensor observations to the masking network.

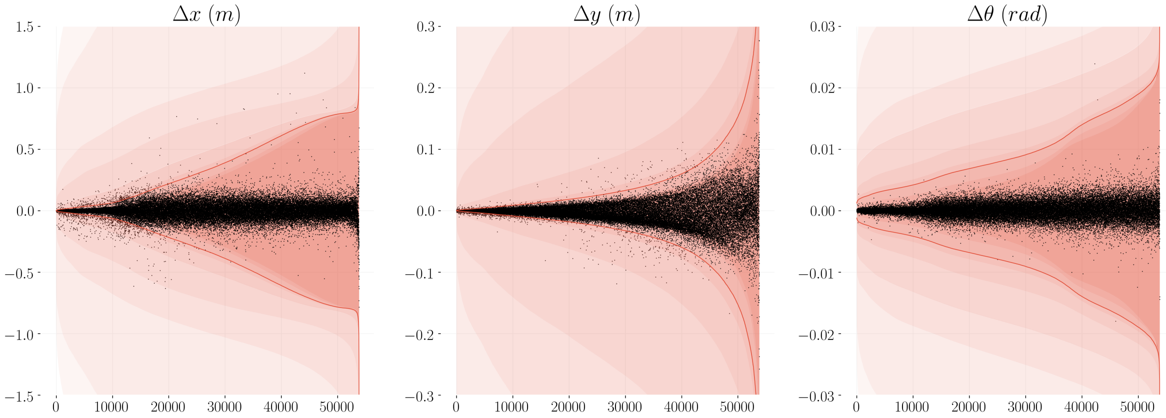
Key to our approach is learning a radar feature embedding that is optimised for pose prediction: compared to correlative scan matching on the raw radar power returns this allows us to reduce errors by over 85%. As predicted, optimising masks directly for pose prediction results in a higher prediction accuracy than mask supervision labelling the temporally stationary scene directly. We also find that simply adapting a deep odometry approach to radar results in significantly worse performance. Our approach in contrast makes use of the inherent top down representation of a radar observation which lends itself to a correlative scan matching procedure, whilst learning to mask out noise artefacts which make pose prediction in radar uniquely challenging. In addition, by adopting a correlative scan matching approach, our results remain interpretable: Figure 2 shows several qualitative examples in which the network learns to mask noise artefacts and dynamic objects in the scene whilst preserving features which are likely to be temporally stationary such as walls.

5.2 Uncertainty Prediction

In addition to the boosts in performance and speed afforded by our approach, we are also able to estimate the uncertainty in each pose prediction: by interpreting the weights generated through the temperature controlled softargmax operation as the probability that each pose candidate is optimum we predict the co-variance $\bar{\mathbf{\Sigma}}$ in our prediction as detailed in Section 3.2.

We now use the methodology proposed in Section 4.4 to tune the temperature parameter β such that the mean Mahalanobis distance $\bar{d}^2 \approx 3$ producing uncertainties $\bar{\mathbf{\Sigma}}$ that are calibrated to the errors in our system. Naively perturbing the temperature parameter away from its original value β_0 degrades pose prediction performance as the feature mask no longer corresponds to the β it was optimised for. Instead, we calculate the predicted pose using β_0, whilst varying β to tune the co-variance matrix. The results of this process (for the 0.8m resolution single mode Cartesian model from Table 1) are shown in Figure 5.2 alongside the marginal distributions for the uncertainty in each pose component plotted with the true errors in our system ordered by predicted uncertainty. For a temperature parameter $\beta = 2.789$ the mean Mahalanobis distance $\bar{d}^2$ is equal to 2.99 giving us well calibrated uncertainty predictions, whilst temperature parameters above and below this value are overly certain and conservative respectively. There is a clear correlation between error and uncertainty with most errors falling within the predicted uncertainty bounds.

Figure 2 shows Gaussian heat maps generated through our approach; the results are highly intuitive with feature embeddings well constrained in each dimension having smaller and symmetric co-variance, whilst pathological solutions arising from a lack of scene constraints increase the uncertainty in Δx.



Temperature Parameter β	4.467	3.548	**2.789**	2.512	1.995	1.413	1.122
Mean Mahalanobis Distance $\bar{d}^2$	4268.471	57.099	**2.992**	1.244	0.283	0.065	0.030

Figure 3: The marginal distributions and errors (black) in each pose component for each example in our test set ordered by predicted uncertainty. The colours correspond to 1.98 standard deviation bounds plotted for each of the temperature parameters given in the table with dark to light moving through the table left to right. The red line corresponds to the standard deviation bound plotted for $\beta = 2.789$ corresponding to a mean Mahalanobis distance of $\bar{d}^2 = 2.99$. For this temperature setting the majority of the errors fall within the 1.98 standard deviation bound. Note the y axis in each case has a different scale.

6 Conclusions

By using a learnt radar feature embedding in combination with a correlative scan matching approach we are able to improve over the previous state of the art, reducing errors in odometry prediction by over 68% and running an order of magnitude faster, whilst remaining as interpretable as a conventional scan matching approach. Additionally, our method affords us a principled mechanism by which to estimate the uncertainty in the pose prediction, crucial for robust real world operation.

Our approach for attaining calibrated uncertainties currently relies on tuning a pre-trained model. An interesting direction for future work would be to incorporate this tuning process into the training pipeline, learning not only a radar feature embedding optimised for pose prediction but also for uncertainty estimation. We leave this for future work.

Acknowledgments

This work was supported by the UK EPSRC Doctoral Training Partnership and EPSRC Programme Grant (EP/M019918/1). The authors also would like to acknowledge the use of Hartree Centre resources and the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work (http://dx.doi.org/10.5281/zenodo.22558).

References

[1] D. Vivet, P. Checchin, and R. Chapuis. Localization and mapping using only a rotating fmcw radar sensor. *Sensors*, 13(4):4527–4552, 2013.

[2] S. H. Cen and P. Newman. Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[3] J. Callmer, D. Törnqvist, F. Gustafsson, H. Svensson, and P. Carlbom. Radar slam using visual features. *EURASIP Journal on Advances in Signal Processing*, 2011(1):71, 2011.

[4] F. Schuster, C. G. Keller, M. Rapp, M. Haueis, and C. Curio. Landmark based radar slam using graph optimization. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 2559–2564. IEEE, 2016.

[5] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2043–2050. IEEE, 2017.

[6] 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)*, pages 1894–1900. IEEE, 2018.

[7] W. Maddern, G. Pascoe, and P. Newman. Leveraging experience for large-scale lidar localisation in changing cities. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1684–1691. IEEE, 2015.

[8] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.

[9] E. Ward and J. Folkesson. Vehicle localization with low cost radar sensors. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*. Institute of Electrical and Electronics Engineers (IEEE), 2016.

[10] H. Rohling. Ordered statistic cfar technique-an overview. In *Radar Symposium (IRS), 2011 Proceedings International*, pages 631–638. IEEE, 2011.

[11] E. Jose and M. D. Adams. An augmented state slam formulation for multiple line-of-sight features with millimetre wave radar. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3087–3092. IEEE, 2005.

[12] E. Jose and M. D. Adams. Relative radar cross section based feature identification with millimeter wave radar for outdoor slam. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 425–430. IEEE, 2004.

[13] M. Rapp, K. Dietmayer, M. Hahn, F. Schuster, J. Lombacher, and J. Dickmann. Fscd and basd: Robust landmark detection and description on radar-based grids. In *Microwaves for Intelligent Mobility (ICMIM), 2016 IEEE MTT-S International Conference on*, pages 1–4. IEEE, 2016.

[14] P. Checchin, F. Gérossier, C. Blanc, R. Chapuis, and L. Trassoudaine. Radar scan matching slam using the fourier-mellin transform. In *Field and Service Robotics*, pages 151–161. Springer, 2010.

[15] E. B. Olson. Real-time correlative scan matching. *Ann Arbor*, 1001:48109, 2009.

[16] W. Churchill. *Experience Based Navigation: Theory, Practice and Implementation*. PhD thesis, University of Oxford, Oxford, United Kingdom, 2012.

[17] D. DeTone, T. Malisiewicz, and A. Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016.

[18] R. Li, S. Wang, Z. Long, and D. Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7291. IEEE, 2018.

[19] I. A. Barsan, S. Wang, A. Pokrovsky, and R. Urtasun. Learning to localize using a lidar intensity map. In *Conference on Robot Learning*, pages 605–616, 2018.

[20] C. McManus, W. Churchill, A. Napier, B. Davis, and P. Newman. Distraction suppression for vision-based pose estimation at city scales. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3762–3769. IEEE, 2013.

[21] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner. The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset. *arXiv preprint arXiv:1909.01300*, 2019.

[22] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[23] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

A Implementation

A.1 Masking Network Architecture

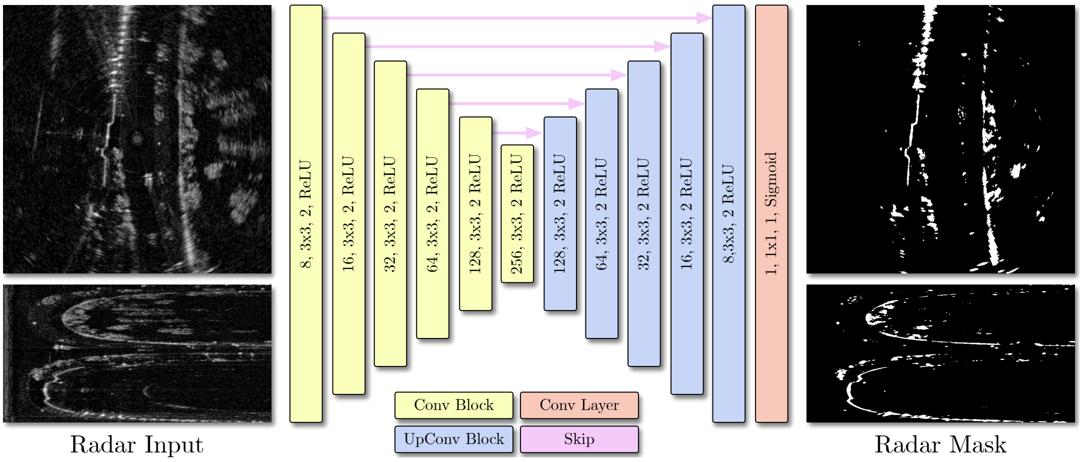


Figure 4: Architecture diagram of the radar masking network. Layers are detailed by output channels, kernel sizes, repetitions and activations respectively. The final network layer has a single output channel with a sigmoid activation to limit the masking range to $[0, 1]$. We experiment using the masking network in both Cartesian and Polar radar representations. Additionally we investigate the impact of modifying the *single* configuration shown to *dual* configuration, in which sequential radar observations used for odometry prediction are concatenated and passed as a single input producing two masks (instead of one) at the output. For more details please refer to the text in Section 4.2. The predictions shown are from a network directly supervised with baseline masks detailed in Section B.1.

A.2 Speed vs Accuracy Trade Off

By reducing our Cartesian grid resolution before calculating the correlation volume, for the same grid coverage we are able to predict the optimum pose in a shorter amount of time to the detriment of pose prediction accuracy. Estimating this trade off for our trained models is challenging and requires many training runs. Instead we investigate the speed-accuracy trade off by performing correlative scan matching on the raw power returns at a variety of grid resolutions according to Algorithms 2 and 3. The results for this process are displayed in Figure 5 which we use to choose the grid resolutions for the main results presented in Table 5.

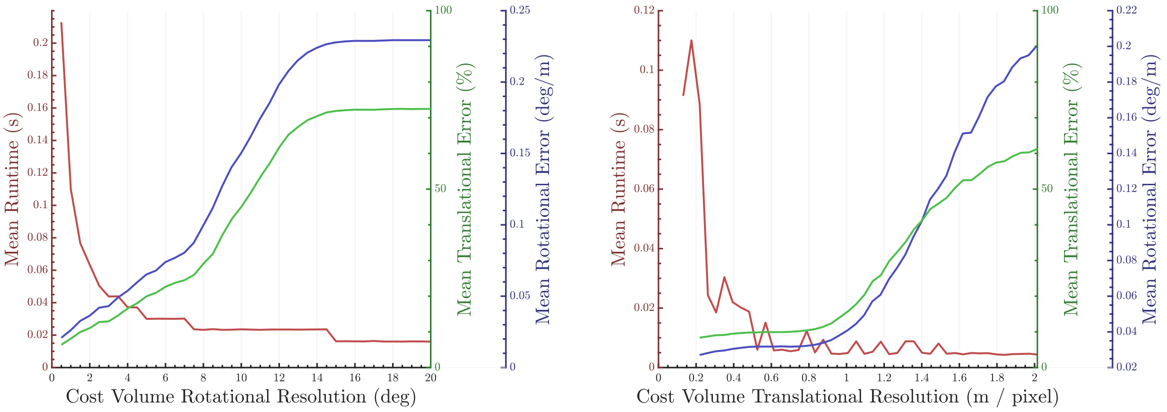


Figure 5: Translational error (green), angular error (blue) and run time (red) as a function of Cost volume resolution in degrees (left) and metres per pixel (right). In the case of limited computational resources or required pose estimate accuracy it is possible to flexibly trade off performance and computational speed.

B Data

B.1 Baseline Masks

To validate the advantages of learning masks directly from pose supervision we compare against supervising the learnt masks directly on the proxy task of predicting temporally static occupied cells. To generate static mask labels we use a similar approach to [6] as detailed in Section 4.1, whereby nearby radar scans from different traversals are warped into the current sensor frame to assess temporal stability. Even with a large corpus of accurately labelled masks identifying static structure suitable for estimating odometry, we observe increased performance by training directly on the task of pose estimation.

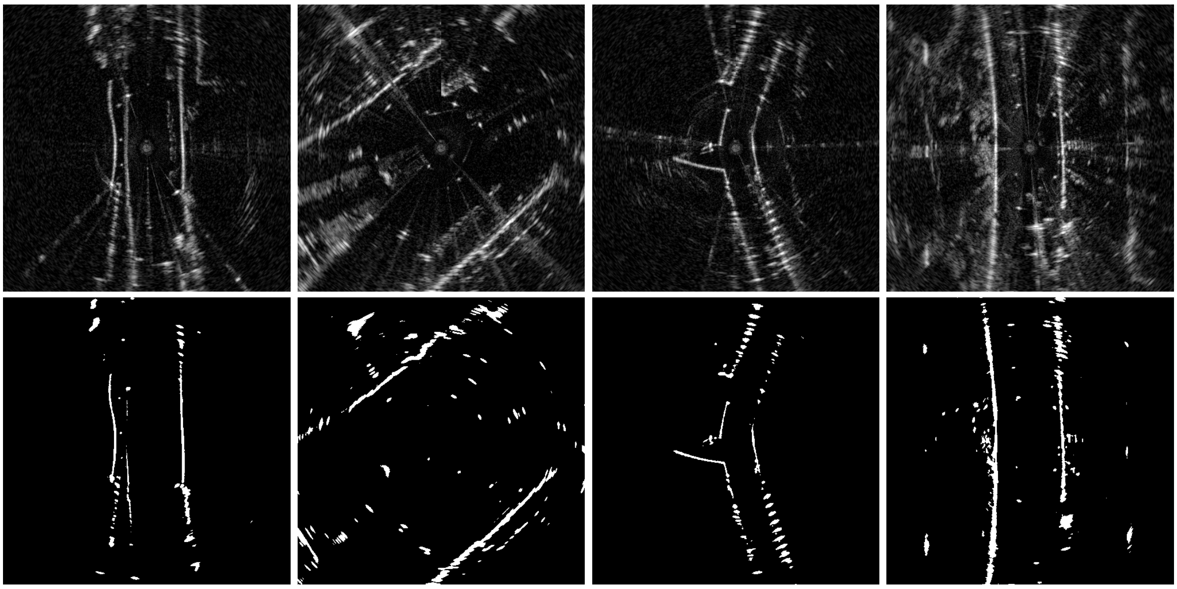
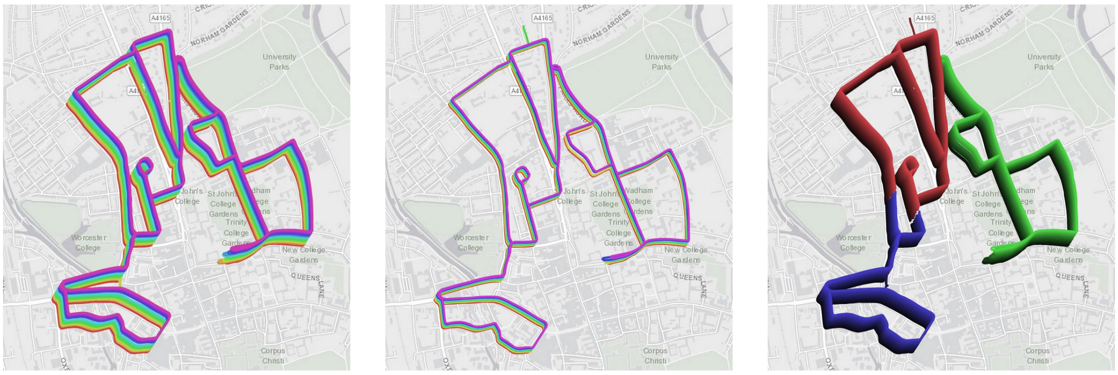


Figure 6: Example generated baseline masks used to supervise the radar masking network directly. For a given raw radar scan at time t (top) we can automatically generate high quality baseline masks identifying structure useful for pose estimation (bottom).

B.2 Dataset Splits



Training Traversals

Testing Traversals

Spatial Cross Validation

Figure 7: Trajectories of the ground truth optimised pose chains used for the 25 training (left) and 7 evaluation (middle) traversals from the Oxford Radar RobotCar Dataset [21] covering a wide variety of traffic and other challenging conditions in complex urban environments. In addition to splitting the dataset temporally we provide spatial cross validation results (right), detailed in Section C.1. Each traversal is incrementally offset with a unique colour for visualisation.

C Results

C.1 Spatial Cross Validation

In Section 5.1 we achieve radar odometry performance far exceeding the state of the art. However we train and evaluate on scenes from the same spatial locations. To assess how well our models generalise to previously un-seen scenes, in this section we train and evaluate our models using spatial cross validation: splitting our traversal loop into three, we train on two out of the three splits, evaluate performance on the third and average results across hold-out splits. Due to the computational demands of training models from scratch on each split, we train our medium resolution model (which is faster to train but has slightly worse performance than its higher resolution counterpart).

Our best model reduces average cross validation errors over the current state of the art by over 25% in translational and 11% in rotational error whilst running over 15x faster. Using this training paradigm we reduce the effective training data diversity by a third. We attribute this to the slight reduction in performance in comparison to the results presented in Section 5.1. We theorise we could significantly boost performance by moving to our highest resolution model also.

Benchmarks	Resolution (m/pixel)	Translational error (%) Mean	IQR	Rotational error (deg/m) Mean	IQR
RO Cen Full Res [2]	0.0432	6.3813	4.6458	0.0189	0.0167
RO Cen Equiv.* [2]	0.1752	*3.6349*	3.3144	*0.0096*	0.0095
Raw Scan	0.4	8.4532	8.0548	0.0280	0.0282
Adapted Deep VO Cart [18]	0.4	11.531	9.6539	0.0336	0.0307
Adapted Deep VO Polar [18]		14.446	11.838	0.0452	0.0430
Visual Odometry [16]		3.7824	1.9884	0.0103	0.0072
Ours					
Polar	0.4	2.8115	2.4189	0.0086	0.0084
Cart	0.4	3.2756	2.8213	0.0104	0.0100
Dual Polar	0.4	3.2359	2.5760	0.0098	0.0091
Dual Cart	0.4	**2.7848**	2.2526	**0.0085**	0.0080

Table 2: Spatial cross validation odometry estimation results. Our approach outperforms the benchmark (italics) in a large proportion of the experiments and we would expect a similar boost in performance to Section 5.1 by moving from our medium to highest resolution model. Our best performing model in terms odometry performance is marked in bold.

C.2 Additional Evaluation Examples

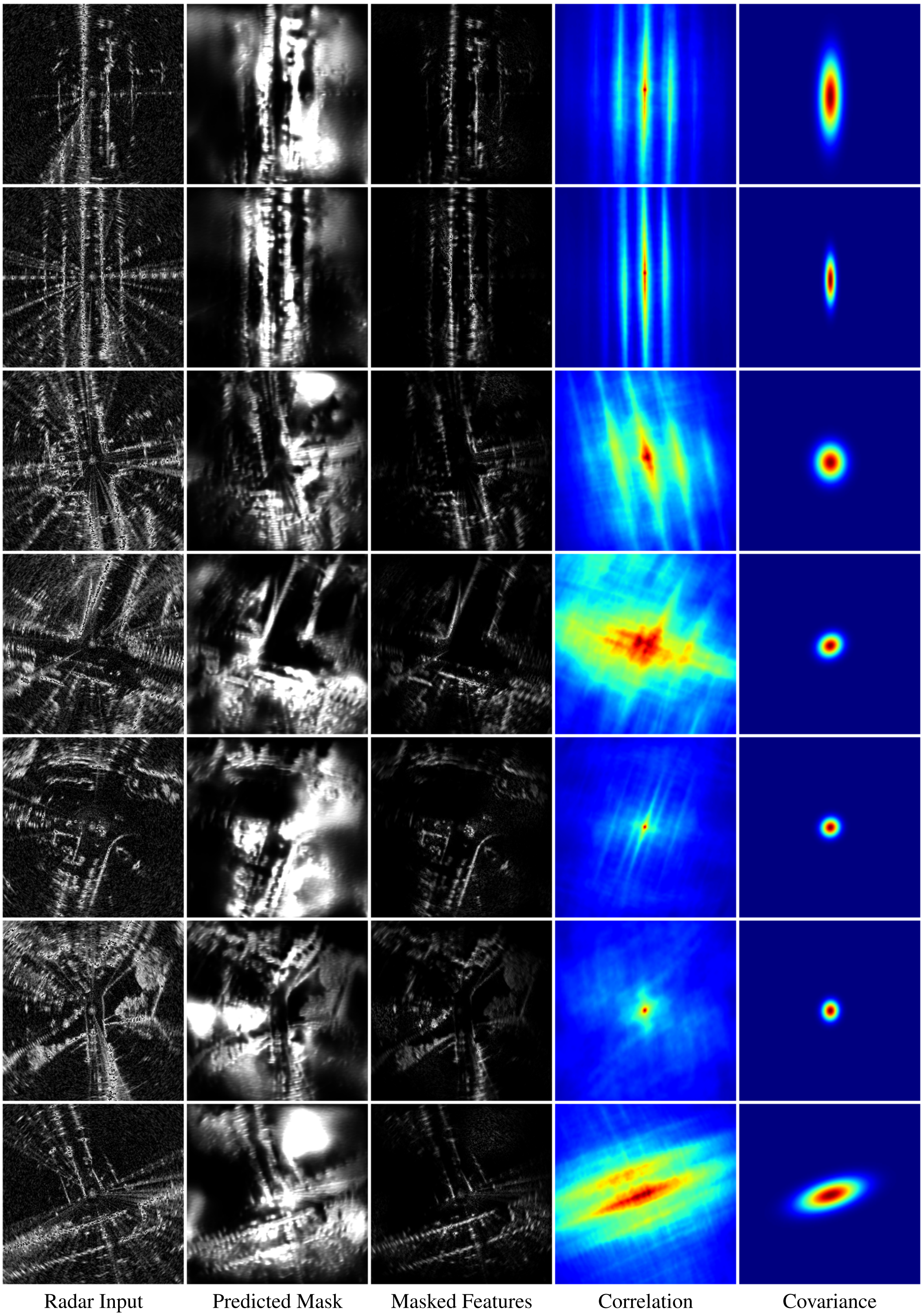


Figure 8: Additional qualitative examples generated from our best performing model. The masks generated from our network filter out noise and distractor objects in the scene whilst preserving temporally consistent features such as walls, well suited for pose prediction. From left to right the raw Cartesian radar scan, the predicted network mask, the masked radar scan, the correlation volume and the fitted gaussian to the correlation volume after temperature weighted softmax.

Statement of Authorship for joint/multi-authored papers for PGR thesis
To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information Dan Barnes, Rob Weston and Ingmar Posner Conference on Robot Learning (CoRL), 2019
Publication Status	Published
Publication Details	Dan Barnes, Rob Weston and Ingmar Posner Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information Conference on Robot Learning (CoRL), 2019

Student Confirmation

Student Name:	Daniel Barnes		
Contribution to the Paper	My contributions to the paper were: Developed the idea behind the paper Collected and processed data Performed the experiments Wrote the paper with co-authors		
Signature		Date	18 / 11 / 2019

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Ingmar Posner		
Supervisor comments	The description above is accurate.		
Signature		Date	7/1/2020

6.1 Extensions

This publication presents a new state-of-the-art RO system, reducing errors by up to 68%, whilst also running an order of magnitude faster than previous approaches. Fundamentally, the approach is facilitated by a Fast Fourier Transform (FFT) based RO architecture, accelerated on GPUs. Despite extensive evaluation of the formulation against the prior work, two additional experiments identify themselves as key in justifying the core design decisions.

First, although prior work in the literature compares against using more classical computer vision approaches for RO, there is limited quantitative analysis available in urban environments. To address this, Section 6.1.1 applies a simple VO estimation pipeline on radar data in the Oxford Radar RobotCar Dataset (detailed in Chapter 5). Secondly, despite the high accuracy RO produced by the final system, there are many other viable formulations for the differentiable RO that could have been used. To this end, Section 6.1.2 replaces the core 3D cross-correlation pose estimator (in x, y, θ) with a Fourier Mellin Transform (FMT) based approach used previously for RO in the literature [43]. By aggregating these results with the published work, we further justify the design decisions and architecture as the state-of-the-art in RO.

6.1.1 Simple Radar Odometry Baseline

The publication we present includes a thorough comparison against the prior state-of-the-art in RO and various other baselines. However, a notable additional comparison would be to apply a more conventional computer vision approach directly to RO.

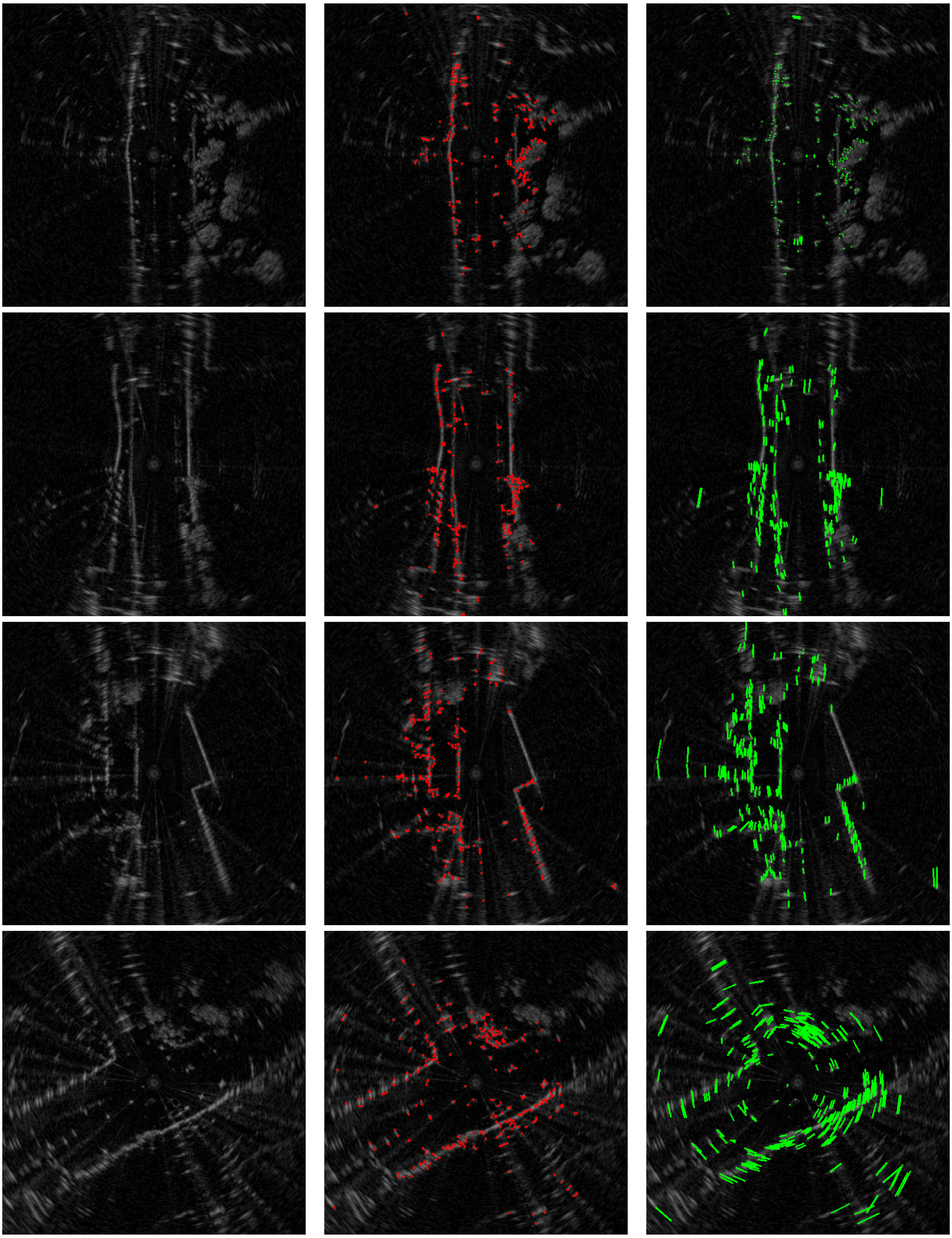
In [11], our primary benchmark for RO, Cen et al. present results comparing the authors' method for landmark extraction to 1D CFAR [40], [64] and SURF [28]. The authors show qualitatively that their landmark extraction does not contain as many redundant returns as 1D CFAR and finds consistently real objects unlike SURF. In order to quantify our gains over and above such traditional baselines, we implement and evaluate a simple VO approach applied to RO.

Computer vision features have been utilised for RO in the literature [38], for example using SIFT and FAST descriptors. These methods roughly follow the same approach of feature detection, descriptor computation, feature matching and finally motion estimation. Despite the successes shown by the prior work, using such approaches on the Oxford Radar RobotCar Dataset does not provide stable motion estimates (as correctly suggested by Cen et. al). The key failure mode of these approaches for RO is brittle and inconsistent feature matching due to ambiguous feature descriptors. This is unsurprising given the descriptors were designed for vision-based applications.

With this observation, for a simple RO baseline we opt to take a descriptor-less approach to motion estimation based on optical flow. First, candidate keypoint locations are extracted from a Cartesian radar scan before calculating sparse optical flow to the previous scan. Given these sparse tracks between sequential scans, a robust [31] motion estimator (constrained to $\mathbb{SE}(2)$) predicts the relative motion. This naive approach is often used as a starting point for VO estimation and serves as a suitable baseline when applying no radar-specific domain knowledge. This approach is visualised in Figure 6.1, with the RO results presented in Table 6.1 alongside the best performing architecture in the publication.

Two additional improvements to the baseline method are presented along with quantitative RO results. The first performs subpixel refinement on the candidate feature locations, in an attempt to better correspond to the real-world positions of the detected keypoints. The second performs motion compensation on the keypoint tracks using an initial motion prior, to account for the spinning nature of the radar sensor. Using this approach for RO is extremely fast, as detailed in Table 6.1, with the core algorithm taking less than 1ms on a CPU. For the interested reader, in its simplest form, this method can be implemented using only three OpenCV [65] functions (a popular computer vision library): `goodFeaturesToTrack` → `calcOpticalFlowPyrLK` → `estimateAffinePartial2D`.

As shown in Table 6.1, applying this naive approach to RO delivers poorer results than both the prior state-of-the-art [11] and the solution presented in the publication. However, despite a reduction in performance, the significantly reduced runtime of this approach is noteworthy, and may be extremely useful in a number of real-world robotics applications.



(a) Radar Input

(b) Features to Track

(c) Sparse Optical Flow

Figure 6.1: Visualisation of the naive radar odometry baseline (as detailed in Section 6.1.1) under stationary (top), straight (middle) and rotational (bottom) movements. Given a raw Cartesian radar scan (a), candidate feature locations are extracted (b) before estimating sparse optical flow to the preceding frame (c). These tracks are fed to a robust transform estimator constrained to $\mathbb{SE}(2)$ in order to calculate the radar motion.

	Resolution	Translational error (%)		Rotational error (deg/m)		Runtime (s)	
Benchamrk	(m/pixel)	Mean	IQR	Mean	IQR	Mean	Std.
Cen et al. [11]	0.1752	3.7168	3.4190	0.0095	0.0095	2.9036	0.5263
Masking By Moving [63] **Chapter 6 Publication**							
Dual Cart	0.8	2.7008	2.2430	0.0076	0.0054	0.0088	0.0007
	0.4	1.7979	1.4921	0.0047	0.0036	0.0194	0.0010
	0.2	**1.1627**	0.9693	**0.0030**	0.0030	0.0747	0.0005
OpenCV Baseline **Section 6.1.1**							
Ransac Affine Estimator	0.8	8.1064	6.8652	0.0251	0.0274	**0.0007**	0.0001
	0.4	5.2006	4.2427	0.0151	0.0164	0.0022	0.0002
	0.2	5.3618	4.5688	0.0151	0.0159	0.0065	0.0005
+ Subpixel Refinement	0.8	8.3847	7.0794	0.0268	0.0267	0.0013	0.0002
	0.4	5.1710	4.2063	0.0151	0.0166	0.0030	0.0004
	0.2	5.3941	4.5507	0.0153	0.0162	0.0092	0.0009
+ Motion Compensation	0.8	7.4603	6.1846	0.0233	0.0255	0.0008	0.0001
	0.4	4.4545	3.3339	0.0145	0.0148	0.0023	0.0002
	0.2	4.0060	3.1418	0.0138	0.0134	0.0067	0.0005
+ Subpixel Refinement + Motion Compensation	0.8	7.5387	6.4558	0.0233	0.0248	0.0014	0.0002
	0.4	4.4278	3.2731	0.0146	0.0148	0.0031	0.0004
	0.2	3.9635	3.0279	0.0137	0.0130	0.0093	0.0010
Fourier Mellin **Section 6.1.2**							
Dual Cart	0.8	8.8374	8.1958	0.0277	0.0290	0.0118	0.0011
	0.4	2.9823	2.6999	0.0093	0.0100	0.0208	0.0015
	0.2	2.6338	2.1862	0.0080	0.0079	0.0236	0.0013

Table 6.1: Radar odometry estimation results supplementary to Table 1 in the manuscript, comparing the literature benchmark and best proposed method against a naive baseline as detailed in Section 6.1.1 and an alternate spectral formulation as detailed in Section 6.1.2. Runtime for the OpenCV Baseline was computed using OpenCV 4.0.1 on a MacBook Pro with a 2.50GHz Intel Core i7-4870HQ and the Fourier Mellin baseline on a 2.7 GHz 12-Core Intel Xeon E5 CPU and NVidia Titan Xp GPU. The radar sensor collects raw scans at 4Hz and is detailed further in Chapter 5.

6.1.2 Fourier Mellin Transform Formulation

An alternate formulation for spectral scan-matching in radar is the Fourier Mellin Transform (FMT), which has been used extensively for image registration and also applied to RO [43].

The primary advantages to using this approach, and motivation for the evaluation in this section, include a less restricted output as well as a smaller computational footprint. When using the FMT, two candidate radar scans can be matched at any relative rotation, whereas the method proposed in the publication is constrained between user defined limits. For the task of odometry this limitation has no consequence, as there is a physical limit to the rotation between sequential scans when driving. However, the FMT formulation is attractive for combined mapping and localisation applications, as upon returning to the same location the sensor may be at any angle. Secondly, the computational complexity of the FMT approach scales more favourably with increased resolution than the 3D cross-correlation used in the publication.

The primary disadvantage when using the FMT for RO is the inability to estimate a full 3x3 covariance matrix for each predicted relative pose. Conversely, the formulation presented in the publication provides full and calibrated uncertainties alongside odometry estimates, which are essential for safe robot deployment in the real world.

Complementary to the publication, we implement RO estimation and mask learning using the FMT as detailed in Algorithm 1. Similar to the modifications made in the publication to ensure differentiability, the FMT becomes fully differentiable by replacing originally argmax operations on Lines 7 and 10 with soft-argmax equivalents.

Using the same training methodology as in the publication, the FMT based RO architecture is trained on pose prediction error alone. The resulting radar masks are clean, and follow the semantics of the scene as shown in Figure 6.2, suppressing sensing artefacts and leaving only walls and buildings useful for pose estimation. The corresponding quantitative RO results are included alongside the baseline in Table 6.1. By inspecting the runtime, we observe the improved complexity scaling with resolution in the penultimate column when compared to our approach. However, in terms of RO performance, the method presented in the publication (using 3D correlation for pose estimation instead of the FMT) is clearly the state-of-the-art approach and still operates at well over real-time speeds.

Algorithm 1: Fourier Mellin Transform for Motion Estimation and Mask Learning

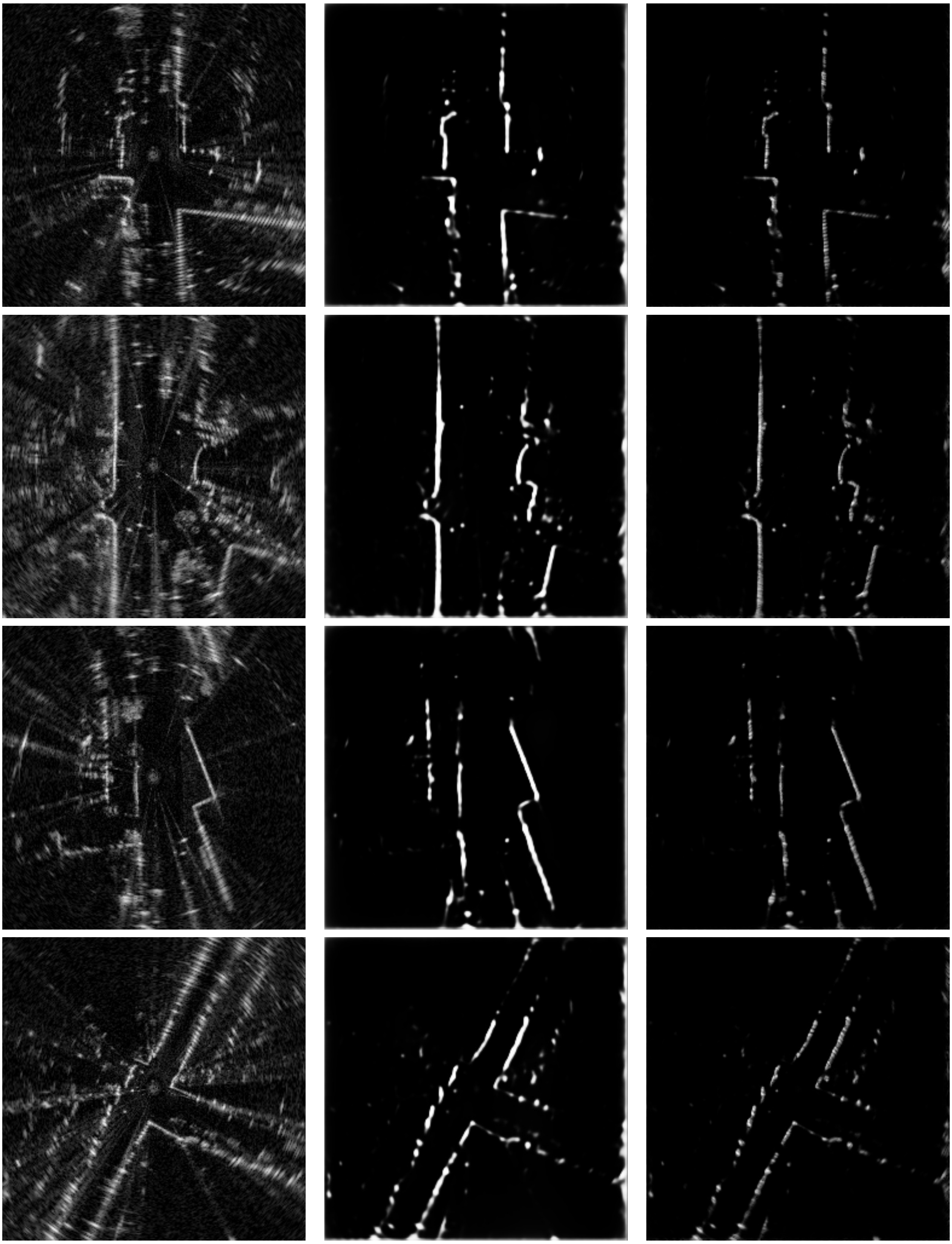
```
Input:
  $\boldsymbol{Z}_1,\ \boldsymbol{Z}_2$          // sequential radar frames in Cartesian form
Parameters:
  $f_\alpha$              // radar masking network
  $T_\theta, T_{xy}$          // rotational and translational softArgMax temperatures
Output:
  $\Delta x,\ \Delta y,\ \Delta\theta$   // relative translation and rotation between radar scans

// Predict radar masks with network $f_\alpha$
$\boldsymbol{M}_1, \boldsymbol{M}_2 \leftarrow f_\alpha(\boldsymbol{Z}_1), f_\alpha(\boldsymbol{Z}_2)$
// Mask each radar scan
$\boldsymbol{S}_1, \boldsymbol{S}_2 \leftarrow \boldsymbol{M}_1 \odot \boldsymbol{Z}_1, \boldsymbol{M}_2 \odot \boldsymbol{Z}_2$
// Attenuate edges to background to reduce high frequency artefacts
$\hat{\boldsymbol{S}}_1, \hat{\boldsymbol{S}}_2 \leftarrow \mathsf{hanning}(\boldsymbol{S}_1), \mathsf{hanning}(\boldsymbol{S}_2)$
// Calculate FFT and apply high pass filter
$\boldsymbol{F}_1, \boldsymbol{F}_2 \leftarrow \mathsf{hpf}(\mathsf{fft2d}(\hat{\boldsymbol{S}}_1)), \mathsf{hpf}(\mathsf{fft2d}(\hat{\boldsymbol{S}}_2))$
// Convert to log polar coordinates
$\boldsymbol{L}_1, \boldsymbol{L}_2 \leftarrow \mathsf{cart2polar}(\boldsymbol{F}_1), \mathsf{cart2polar}(\boldsymbol{F}_2)$
// Phase correlate for relative scale and rotation weights
$\boldsymbol{C}(s,\theta) \leftarrow \mathsf{phaseCorrelate}(\boldsymbol{L}_1, \boldsymbol{L}_2)$
// Estimate relative rotation assuming no change in scale
$\Delta\theta \leftarrow \mathsf{softArgMax}(\boldsymbol{C}(s,\theta), T_\theta),\ \text{ where } s = 1$
// Rotate FFT of first radar scan
$\boldsymbol{F}_{1r} \leftarrow \mathsf{rotate}(\boldsymbol{F}_1, \Delta\theta)$
// Phase correlate for relative translation weights
$\boldsymbol{C}(x,y) \leftarrow \mathsf{phaseCorrelate}(\boldsymbol{F}_{1r}, \boldsymbol{F}_2)$
// Estimate relative translation
$\Delta x, \Delta y \leftarrow \mathsf{softArgMax}(\boldsymbol{C}(x,y), T_{xy})$
```

Algorithm 2: Phase Correlation

```
Input:
  $\boldsymbol{f}_1,\ \boldsymbol{f}_2$          // signals related by a spatial translation $(\Delta x, \Delta y)$
Output:
  $\boldsymbol{C}(x,y)$          // relative translation weights

// Calculate FFT of each signal
$\boldsymbol{F}_1, \boldsymbol{F}_2 \leftarrow \mathsf{fft2d}(\boldsymbol{f}_1), \mathsf{fft2d}(\boldsymbol{f}_2)$
// Calculate Cross-Power Spectrum
$\boldsymbol{R} \leftarrow (\boldsymbol{F}_1 \odot \boldsymbol{F}_2^*)\ /\ |\boldsymbol{F}_1 \odot \boldsymbol{F}_2|$
// Convert spectral to spatial translation weights
$\boldsymbol{C}(x,y) \leftarrow \mathsf{fft2d}^{-1}(\boldsymbol{R})$
```



(a) Radar Inputs **(b)** Predicted Masks **(c)** Masked Radar Scans

Figure 6.2: Qualitative radar mask predictions on a held-out test set when trained using the Fourier Mellin Transform formulation. Similar to the results we present in the manuscript, this formulation naturally learns to attenuate sensing artefacts and distractor objects, leaving only the static structure in radar scans (such as walls and buildings) needed for accurate pose estimation.

7

Under the Radar: Learning to Predict Robust Keypoints for Odometry Estimation and Metric Localisation in Radar

Chapter 6 presented a dense, direct approach to RO which far exceeded the previous state-of-the-art [11]. In other modalities, keypoint-based approaches are often preferred due to their condensed representation and applicability to other tasks (such as place recognition), making them an interesting avenue for research in radar. In vision, keypoint detectors have been researched for decades and classical implementations such as SIFT [27], SURF [28] and ORB [30] are highly performant and reusable across applications and sensor modalities. Nevertheless, learning approaches to keypoint detection have closed the performance gap in recent years. In radar, where defining hand-crafted keypoints proves challenging due to the complex interactions present in the sensor data, this approach provides an attractive avenue to pursue.

This manuscript presents a self-supervised method for learning descriptive keypoints for motion estimation and localisation from ground truth odometry information alone. Similar to Chapter 6, we avoid imposing any human constraints or assumptions by instead incorporating a differentiable keypoint-based motion estimator within a RO architecture. In doing so, our network naturally learns to detect keypoint locations, scores and descriptors suitable for motion estimation and place recognition – all without any manual supervision. Despite only being trained for RO, the keypoint descriptors prove to be excellent for place recognition. Hence, with this added capability, we present a full mapping and localisation system in radar, capable of running at well over real-time speeds.

This manuscript has been accepted to the IEEE International Conference on Robotics and Automation (ICRA), 2020. A video summary of the publication can be found at: `https://youtu.be/L-PO7nxWpJU`

Under the Radar: Learning to Predict Robust Keypoints for Odometry Estimation and Metric Localisation in Radar

Dan Barnes and Ingmar Posner

Abstract— **This paper presents a self-supervised framework for learning to detect robust keypoints for odometry estimation and metric localisation in radar. By embedding a differentiable point-based motion estimator inside our architecture, we learn keypoint locations, scores and descriptors from localisation error alone. This approach avoids imposing any assumption on what makes a robust keypoint and crucially allows them to be optimised for our application. Furthermore the architecture is sensor agnostic and can be applied to most modalities. We run experiments on 280km of real world driving from the Oxford Radar RobotCar Dataset and improve on the state-of-the-art in point-based radar odometry, reducing errors by up to 45% whilst running an order of magnitude faster, simultaneously solving metric loop closures. Combining these outputs, we provide a framework capable of full mapping and localisation with radar in urban environments.**

I. Introduction

Robust egomotion estimation and localisation are critical components for autonomous vehicles to operate safely in urban environments. Keypoints are routinely used in these applications but are typically manually designed or not optimised for the task at hand. Keypoints represent repeatable locations under different viewpoints; hence conditions such as lighting (lens-flare), weather (rain) or time of day (night) can have drastic effects on their quality. The key to improving keypoint quality and robustness is to learn keypoints *specifically tailored* for these tasks and sensor modality.

There is increasing research into radar for urban robotics and because of it's wavelength and range holds the promise of directly addressing many of aforementioned challenges. However, it is also a notoriously challenging sensing modality: typically covered with noise artefacts such as ghost objects, phase noise, speckle and saturation. Hence using off-the-shelf keypoint detectors designed for other modalities is ill-advised, but makes radar an ideal, if challenging, candidate for learning a more optimal keypoint detector.

In this paper we present a self-supervised approach for learning to predict keypoint locations, scores and descriptors in radar data for odometry estimation and localisation. We achieve this by embedding a differentiable point-based motion estimator inside our architecture and supervise only with automatically generated ground truth pose information. This approach avoids imposing any assumption on what makes a robust keypoint; crucially allowing them to be optimised for our application rather than on some proxy task. Furthermore, the architecture itself is sensor agnostic as long as real world keypoint locations can be inferred.

Authors are from the Applied AI Lab, University of Oxford, UK. {dbarnes,ingmar}@robots.ox.ac.uk

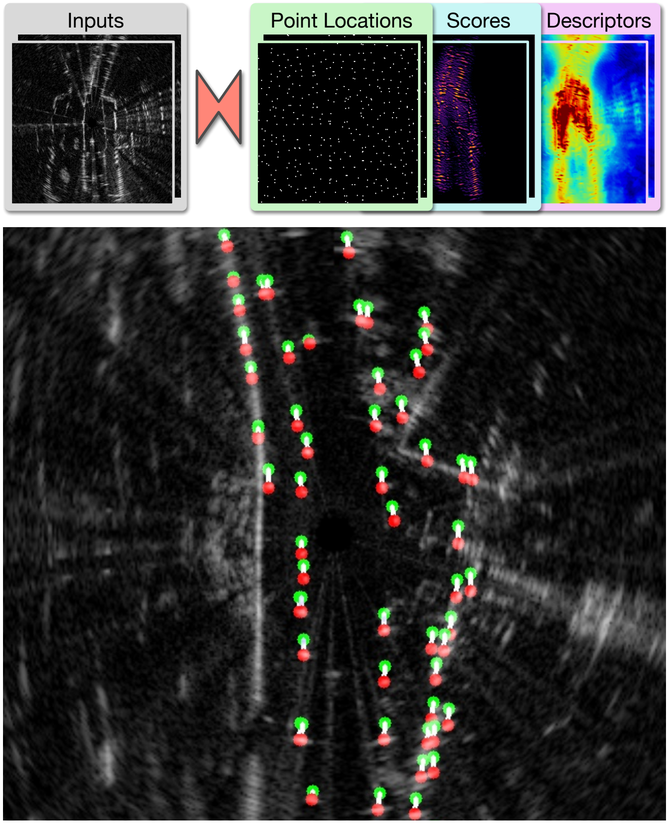


Fig. 1. Learned keypoints for localisation in radar. Given pair of input radar scans (top left) a trained CNN predicts keypoint locations, scores and descriptors (top right). We calculate point matches using descriptor cosine similarity; with final matches also weighted by keypoint scores allowing us to ignore points belonging to noise and unobservable regions (centre with only highest scoring points shown). Finally a pose estimator calculates the optimal transform from the point matches. Crucially the formulation is fully differentiable and can be supervised on odometry error alone, thereby learning keypoint locations, scores and descriptors that are optimal for localisation.

Our approach leads to a state-of-the-art in point-based radar odometry when evaluated on the Oxford Radar RobotCar Dataset [1] driving in complex urban environments. In addition the formulation detects metric loop closures, leading to a full mapping and localisation system in radar data.

II. Related Work

Extracting keypoints, such as SIFT [2], SURF [3] and ORB [4], from sensor data has historically been an initial step for egomotion estimation, place recognition, and simultaneous localisation and mapping (SLAM). Recently CNN based keypoint detectors have emerged predicting locations [5], [6] and also descriptors [7], [8]. However ground-truth supervision is challenging as any reliably detected location is

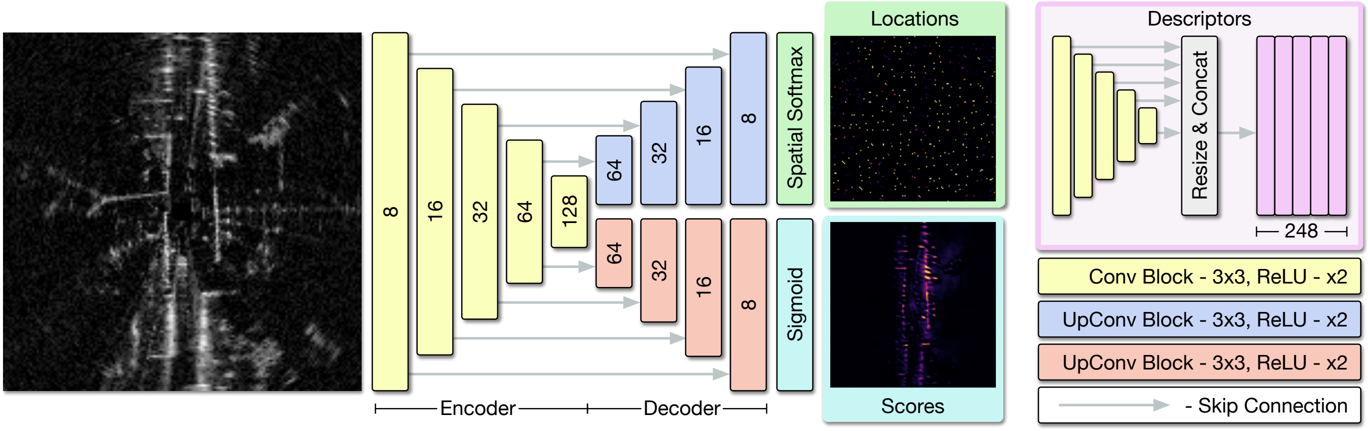


Fig. 2. Network architecture for predicting keypoint locations, scores and descriptors. The height of each block indicates the spatial dimensions of the feature map, which vary by a factor of 2 between blocks through max-pooling or bilinear-interpolation. A dense pixel-wise descriptor map (top right) is created by resizing the output of each encoder block to the size of the input before concatenation into a single feature map. For keypoint locations, spatial softmax is performed on a per cell basis with cell size chosen such that 400 keypoints are predicted. A pointwise convolution with no activation and single channel output precedes the sigmoid and spatial softmax operations. The number of output channels are detailed for each block.

a candidate keypoint. Typically, these keypoint detectors are trained with homography related losses to promote keypoint repeatability or use labels from other detectors. However, both these solutions are suboptimal given the alternative of learning keypoints tailored for a downstream task.

By embedding a differentiable point-based pose estimator [9] learns to predict keypoint locations for the task of rotation prediction; however the formulation predicts only object category specific keypoints which cannot generalise to new scenes. Conversely [10] registers two point clouds by predicting point-wise descriptors for matching, followed by the same pose estimation formulation.

Our target domain of radar is becoming an increasingly researched modality for mobile robotics and with the recently released Oxford Radar RobotCar Dataset [1], a radar extension to the Oxford RobotCar Dataset [11], we expect interest to continue to grow. The seminal work on egomotion estimation in this modality [12] extracts point features before predicting pose; however the point extraction is hand-crafted and may not be optimal for the task. The current state-of-the-art in egomotion estimation in radar [13] employs a correlation-based approach, with learned masking to ignore moving objects and noise artefacts, but is limited to a maximum rotation, making it unsuitable for metric localisation.

Inspired by [9], [10] we learn to predict keypoints specialised for localisation by embedding a pose estimator in our architecture and use only pose information as supervision. This avoids imposing any assumptions on what makes suitable keypoints and enables pose prediction at any angle, a limitation of the current state-of-the-art [13]. Furthermore training over a large dataset, we produce descriptors ideally suited for place recognition without tailored architectures [14] or training regimes designed for that task.

III. Learning Point-Based Localisation

In the following section we outline our approach for learning robust keypoints from ground truth pose information. No part of the approach or model design have been tailored for radar data and can be applied to other modalities such as vision or LIDAR. Our method takes the following steps:

1) Keypoint Prediction: From a raw radar scan we predict keypoint locations, scores and descriptors.

2) Pose Estimation: Given keypoints from two proximal scans we estimate the optimal transform between them and use the errors to train keypoint prediction.

3) Metric Localisation: Using the same keypoint descriptors as a summary of the local scene, we detect and solve metric loop closures.

A. Keypoint Prediction

We adopt a U-Net [15] style convolutional encoder-multi-decoder network architecture (with concatenation skip connections) as shown in Fig. 2 to predict full resolution point locations, scores and descriptors.

The Locations head predicts the sub-pixel locations of each keypoint. To achieve this, we divide the full resolution Locations output into equally sized square cells, with each producing a single candidate keypoint. We apply a spatial softmax on each cell followed by a weighted sum of pixel coordinates to return the sub-pixel keypoint locations.

The Scores head predicts how useful a keypoint is for estimating motion and is mapped to $[0, 1]$ by passing the full resolution logits through a sigmoid function. A perfect scores output would give all static structure in the scene, such as walls and buildings, a score of 1 and all noise, moving objects and empty regions a score of 0.

The Descriptors aim to uniquely identify real-world locations under keypoints so that we can relate points by comparing descriptor similarity. Dense descriptors are created by resizing the output of each encoder block (shown in yellow) to the input resolution before concatenation into a single 248 channel feature map.

B. Pose Estimation

Given a set of keypoint locations we can extract keypoint descriptors and scores using $@$, where $@$ is a sampling function so that $X \ @ \ y$ takes a bilinear interpolation of dense feature map X at coordinates y. The keypoint descriptors are then ℓ_2 normalised so that cosine similarities between any pair is in the range $[-1, 1]$ using: $\tilde{d} = \ell_2(d) = d \,/\, ||d||_2$.

Given two proximal radar scans and their predicted keypoint locations, scores and descriptors we can match keypoints using the differentiable formulation in Algorithm 1 producing keypoint matches ($\boldsymbol{P}_s, \boldsymbol{P}_d$) and weights ($\boldsymbol{w}$) in the range $[0, 1]$. The weights are a combination of the keypoint scores and descriptor cosine similarity; hence matches are only kept if part of the static scene, as predicted by keypoints scores, and identified the same real world location by comparing keypoint descriptors.

The matching is implemented as a dense search for optimum keypoint locations in the destination radar scan given keypoints in the source radar scan. Although matching keypoints directly would be computationally preferable, this formulation produces improved results while still running at well over real-time speeds.

Similar to [9], [10] given matched keypoints and weights we calculate the transform between them using singular value decomposition (SVD) as laid out in Algorithm 2 (detailed further in [16]). Crucially this pose estimation is differentiable, allowing us to backpropagate from transform error right through to the keypoint prediction network.

Given a ground-truth transform between two radar scans, we train with a loss penalising the error in translation and rotation, with weight $\alpha = 10$, learning keypoints optimal for motion estimation.

$$\mathcal{L} = ||\hat{t} - t||_2 + \alpha \; ||\hat{R}R^{\mathsf{T}} - \mathbf{I}||_2 \tag{1}$$

C. Metric Localisation

For each radar scan we assemble a dense descriptor map which enables the keypoint matching previously discussed. Although trained for the task of pose estimation, we reuse the descriptors to produce a location specific embedding $\boldsymbol{G}$ by max-pooling the dense descriptors $\boldsymbol{D}$ across all spatial dimensions, resulting in a single 248-D embedding. This process adds practically no overhead to the inference speed of the network. At run-time, we compare the cosine similarity of the current embedding to previously collected embeddings; when the similarity crosses a threshold, the pair is deemed to be a topological loop closure (at the same physical location). When a topological loop closure is detected, the respective keypoints are solved for a full metric loop closure using the same pose estimation formulation detailed in Section III-B.

IV. Experimental Setup

We aim to evaluate our approach for localisation on challenging radar data from the Oxford Radar RobotCar Dataset [1] through the tasks of odometry estimation and place recognition.

A. Network Training

We train our approach using 25 10km traversals from the Oxford Radar RobotCar Dataset [1] which provides Navtech CTS350-X radar data and ground truth radar poses. For

Algorithm 1: Differentiable Point Matching

```
Input:
  $\boldsymbol{P}_s$   // source point pixel locations
  $\boldsymbol{D}_s$, $\boldsymbol{D}_d$ // source and destination descriptor maps
  $\boldsymbol{S}_s$, $\boldsymbol{S}_d$ // source and destination score maps
Parameters:
  $T$ // descriptor cosine distance softmax temperature
  $\boldsymbol{X}$ // pixel locations map
Output:
  $\boldsymbol{P}_d$ // destination point locations
  $\boldsymbol{w}$ // point match weights
for $i \leftarrow 1$ to $n$ do    // For each source point
    // Extract and normalise source point descriptor
    $\boldsymbol{d}_{si} \leftarrow \ell_2 \; ( \; \boldsymbol{D}_s @ \; \boldsymbol{p}_{si} \; )$
    // Pixelwise cosine distance to dest. descriptor map
    $\boldsymbol{C}_i \leftarrow \boldsymbol{d}_{si} \odot \boldsymbol{D}_d$
    // Apply temperature weighted softmax
    $\boldsymbol{S} \leftarrow \sigma(T\boldsymbol{C}_i)$
    // Extract destination point pixel coordinates
    $\boldsymbol{p}_{di} \leftarrow \boldsymbol{S} \odot \boldsymbol{X}$
    // Extract and normalise destination point descriptor
    $\boldsymbol{d}_{di} \leftarrow \ell_2 \; ( \; \boldsymbol{D}_d @ \; \boldsymbol{p}_{di} \; )$
    // Extract source and destination point scores
    $s_{si} \leftarrow \boldsymbol{S}_s @ \; \boldsymbol{p}_{si} \; , \; s_{di} \leftarrow \boldsymbol{S}_d @ \; \boldsymbol{p}_{di}$
    // Compute weight for point match
    $w_i \leftarrow \frac{1}{2}(\boldsymbol{d}_{si} \odot \boldsymbol{d}_{di} + 1) \; s_{si} \; s_{di}$
end
```

Algorithm 2: Differentiable Pose Estimation

```
Input:
  $\boldsymbol{P}_s$, $\boldsymbol{P}_d$ // source and destination point pixel locations
  $\boldsymbol{w}$     // point match weights
Output:
  $\boldsymbol{t}$, $R$ // optimal translation and rotation that minimise:
          $\sum_{i=1}^{n} w_i \|(R\boldsymbol{q}_{si} + \boldsymbol{t}) - \boldsymbol{q}_{di}\|^2$
// Convert pixel locations to world locations
$\boldsymbol{Q}_s \leftarrow \text{pix2world}(\boldsymbol{P}_s) \; , \; \boldsymbol{Q}_d \leftarrow \text{pix2world}(\boldsymbol{P}_d)$
// Compute the weighted centroids of both point sets
$\bar{\boldsymbol{Q}}_s \leftarrow \sum_{i=1}^{n} w_i \; \boldsymbol{q}_{si} \; / \; \sum_{i=1}^{n} w_i$
$\bar{\boldsymbol{Q}}_d \leftarrow \sum_{i=1}^{n} w_i \; \boldsymbol{q}_{di} \; / \; \sum_{i=1}^{n} w_i$
// Compute the centred vectors
$\boldsymbol{x}_i \leftarrow \boldsymbol{q}_{si} - \bar{\boldsymbol{Q}}_s \; , \; \boldsymbol{y}_i \leftarrow \boldsymbol{q}_{di} - \bar{\boldsymbol{Q}}_d \; , \qquad i = 1, 2, ..., n.$
// Compute the d x d covariance matrix
$S \leftarrow XWY^T$
// Compute the singular value decomposition
$U, \Sigma, V \leftarrow SVD(S)$
// Compute optimal rotation
$R \leftarrow V \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & det(VU^T) \end{pmatrix} U^T$
// Compute optimal translation
$\boldsymbol{t} \leftarrow \bar{\boldsymbol{Q}}_d - R\bar{\boldsymbol{Q}}_s$
```

training we convert the polar radar scan to Cartesian at either 0.7 or 0.35 m/pixel resolution and apply additional data augmentation to the odometry ground truth so that we can reliably solve pose at any rotation between radar scans. For all training we use TensorFlow [17] and the Adam [18] optimiser with a learning rate of $\lambda = 10^{-3}$ until the task loss is minimised on a small validation set for at least 150k steps.

B. *Evaluation Metrics*

We evaluate our approach on 7 further dataset traversals for a total of approximately 70km.

1) Odometry: To quantify the performance of the odometry estimated by our point-based architecture, we compute translational and rotational drift rates using the approach proposed by the KITTI odometry benchmark [19] for various resolutions and network configurations. Specifically, we compute the average normalised end-point translational and rotational error for all subsequences of length (100, 200, . . . , 800) metres compared to the ground truth radar odometry.

2) Localisation: We follow the place recognition evaluation metrics as in [14], [20], [21]. The query radar scan is deemed correctly localised if at least one of the top N retrieved radar scans, according to descriptor cosine distance, is within $d = 5m$ from the ground truth position of the query. For the purposes of this evaluation, detecting loop closures to the same trajectory are ignored and results are plotted for localising to other test datasets. The metrics presented use $d = 5m$ rather than $d = 25m$ as in [14] because when nearer we can reliably solve for a full metric loop closure.

We further compare against the addition of a trainable NetVLAD layer [14] (512-D output / 64 clusters) to project location embeddings (Section III-C) onto a localisation specific metric space expecting this to improve performance. The highest performing model according to odometry metrics is frozen before the NetVLAD layer and fine-tuned for place recognition. We use the batch hard triplet loss in Eq. (2) with online hard negative mining, where $d(i, j)$ returns the ℓ_2 distance between descriptors i and j, sampling 5 positive locations ($\boldsymbol{p}$) closer than 5m and 5 negative locations ($\boldsymbol{n}$) further than 25m away per training sample.

$$\mathcal{L}_{place\ rec.} = max\left(\max_{p\in\boldsymbol{p}} d(a,p) - \min_{n\in\boldsymbol{n}} d(a,n) + m, 0\right) \quad (2)$$

V. Results

A. *Odometry Performance*

The end-point-error evaluation is presented in Table I. The key benchmark we compare to is ‘RO Cen’, the state-of-the-art in *point-based* radar odometry, where ‘Full Res.’ operates on the full resolution of the radar and ‘Equiv’ is downsampled through max pooling to the resolution the algorithm was designed for. As can be seen, our best performing model (shown in bold) outperforms these by 45% in translational and 29% in rotational error whilst running an order of magnitude faster at 28.5Hz. Increasing the resolution would likely boost performance further at the cost of runtime speed. Additionally we outperform pure CNN regression using a model designed and trained for pose estimation [22].

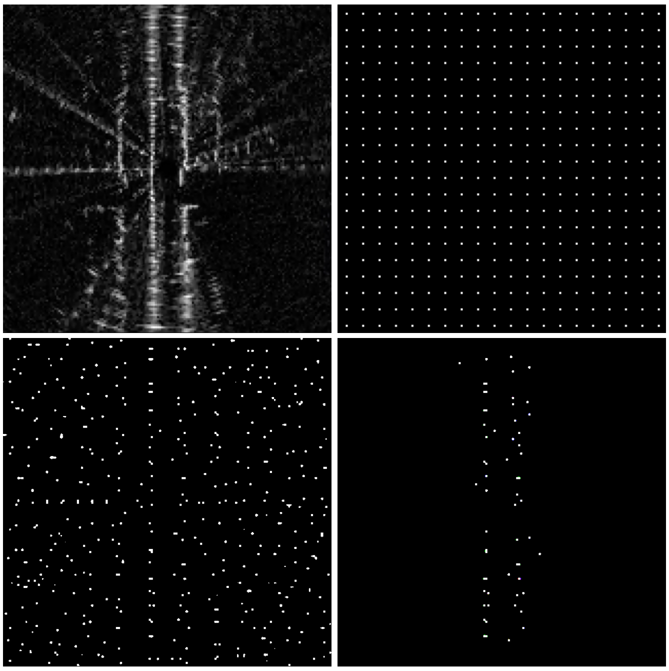


Fig. 3. Architecture design experiments. For a given radar input (top left) we compare against a baseline where points are distributed uniformly across the scan (top right) with Location and Score heads disabled. With Location head enabled we learn to predict per cell sub-pixel keypoint locations (bottom left). When the Score head is also enabled (bottom right) we are able to ignore points due to noise artefacts or in unobservable regions, leaving only points located on the static structure in the scene.

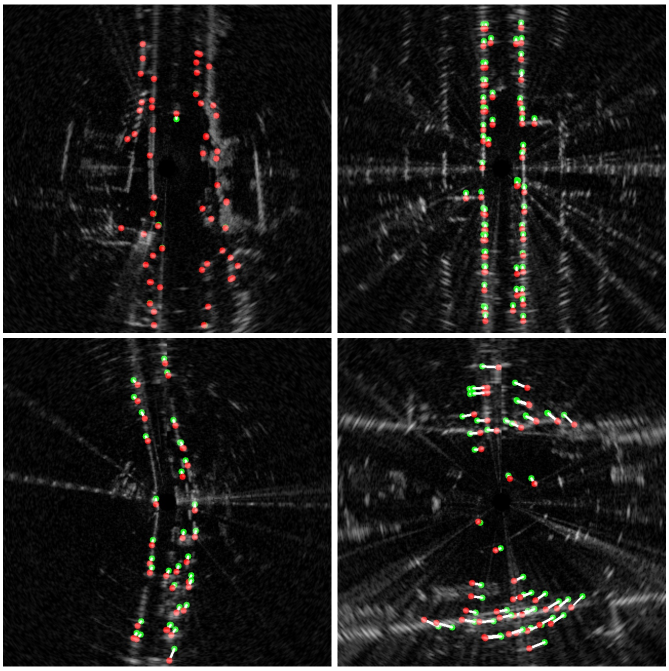


Fig. 4. Odometry keypoint matches when stationary (top left), travelling forward (top right), on a slight bend (bottom left) and performing an aggressive turn (bottom right). Points from sequential scans are shown in red or green and the match between them shown in white with only the highest scoring matches shown. In all situations the point locations and weights accurately capture the vehicle motion and are well localised to the static structure in the scene such as walls and buildings. Interestingly the only moving match in the stationary example belongs to a vehicle moving through the scene.

Benchmarks	Translational Error (%)	Rotational Error (deg/m)	Runtime (s)
RO Cen Full Res. [12]	8.4730	0.0236	0.3059
RO Cen Equiv. [12]	3.7168	0.0095	2.9036
CNN Regression [22]	4.7683	0.0141	0.0060
Masking By Moving Equiv. * [13]	1.5893	0.0044	0.0169
Stereo Visual Odometry * [23]	3.9802	0.0102	0.0062

Ours					
Resolution (m)	Localiser	Scores	Translational Error (%)	Rotational Error (deg/m)	Runtime (s)
0.6912			18.6996	0.0569	0.0111
	✓		8.3700	0.0253	0.0117
		✓	4.4153	0.0140	0.0125
	✓	✓	3.9518	0.0138	0.0134
0.3456			22.9889	0.0644	0.0291
	✓		9.0955	0.0278	0.0305
		✓	2.4607	0.0089	0.0323
	✓	✓	**2.0583**	**0.0067**	0.0340

TABLE I

ODOMETRY DRIFT EVALUTAION. BEST PERFORMING MODEL MARKED IN BOLD. METHODS NOT DIRECTLY COMPARABLE MARKED WITH *.

We provide two additional benchmarks not directly comparable to the method proposed. Firstly, odometry estimation in another modality using an off-the-shelf visual odometry system [23] as chosen by the prior state-of-the-art in radar odometry [12], which we exceed in performance by a significant margin. Secondly, the current state-of-the art in dense radar odometry estimation [13] at the most closely related configuration and resolution. Whilst we do not exceed the performance of [13], we are not limited by rotation, crucial for solving metric loop closures in Section V-C.

We evaluate our architecture design optionally disabling the Location and Score heads. The effect these have on predicted keypoints are visualised in Fig. 3. At both test resolutions, enabling both Location and Score heads lead to the best performance. As the majority of a radar scan is either: empty, unobserved, or contain noise artefacts; scores prove more essential to odometry performance than locations as these regions can be ignored. Odometry keypoint matches are visualised Fig. 4 in various locations and vehicle movements, showing points localise well to the static structure.

B. Localisation Performance

Place recognition results are shown in Fig. 5. We compare creating location embeddings from the full resolution dense descriptor map and from descriptors extracted at keypoint locations. Even when not trained on the task of place recognition, our location embeddings reliably allow us to detect topological loop closures ('Max Descriptors') far exceeding randomly initialised weights ('Rand.' with the keypoint variant off the bottom of the graph).

When fine tuning an additional layer for place recognition as described in Section IV-B.2, we freeze the best performing odometry estimation model (bottom row in Table I) before adding the NetVLAD layer. Despite a better training convergence, interestingly the NetVLAD layer based embeddings generalise worse to the test set than the embeddings trained on the task of odometry. Further experiments increasing the dimensionality of the core architecture descriptors, as well as the NetVLAD layer, showed negligible improvements at the cost of runtime speed. Qualitative topological localisation results can be seen in Fig. 6.

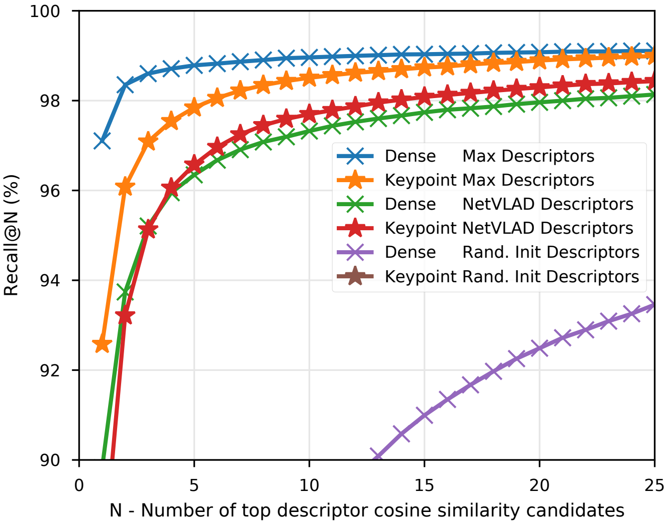


Fig. 5. Place recognition results between test datasets showing Recall vs Number of top candidates as per the results in [14]. The embeddings learnt by our architecture trained for odometry ('Max') exceed the performance of fine-tuning an additional layer to project onto a localisation specific metric space ('NetVLAD').

C. Mapping and Localisation

Given we now have a system that can reliably solve the pose between two proximal radar scans and a method for detecting topological loop closures, we can combine these into a full mapping and localisation stack running at well over real-time speeds.

For online applications we run three processes in parallel that output a fully optimised map. We run odometry estimation in a process to produce open loop trajectory edges. The second process detects topological loop closures by comparing against stored location embeddings, before solving the relative pose for metric loop closures. We store embeddings in an KDTree for fast lookup and set the cosine similarity threshold to give 100% loop closure precision according to a small validation set. The third process receives all edges and continuously optimises the underlying pose graph using g2o [24], producing a complete map of how the vehicle has travelled. A qualitative figure of our full mapping and localisation system can be seen in Fig. 7.

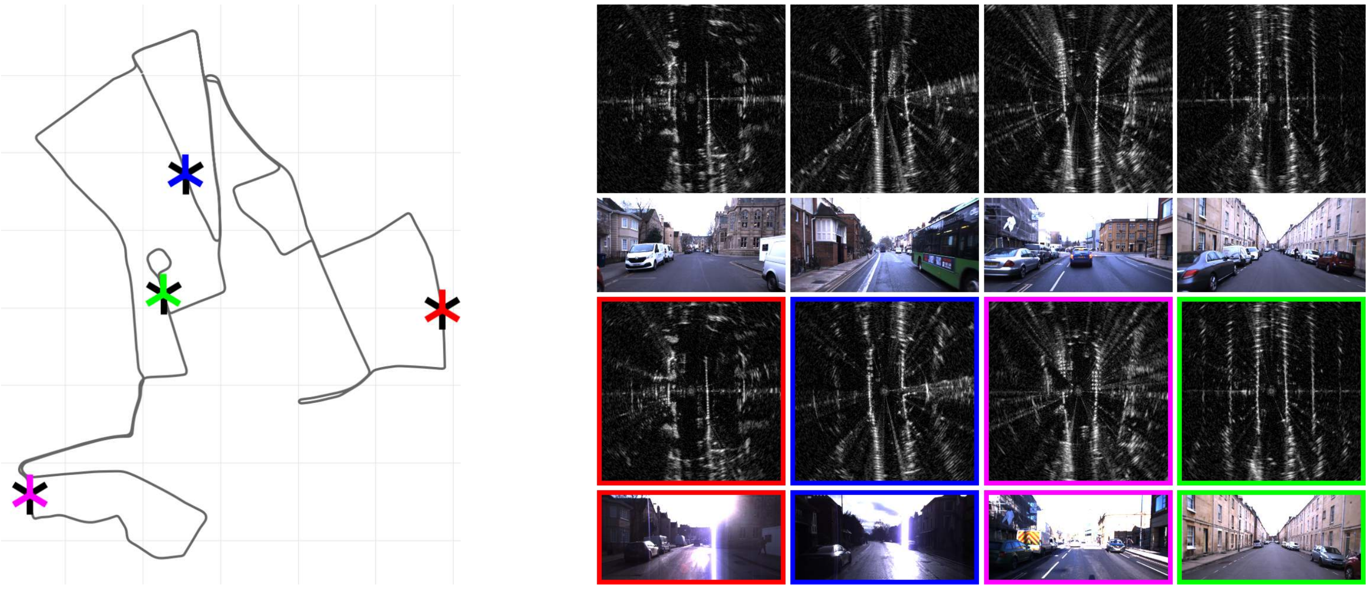


Fig. 6. Qualitative loop closure detections. For a given radar input, shown as Y on the map (left) and top row (right), our location specific embeddings enable us to detect loop closures from different traversals of the route, shown as ⅄⅄⅄⅄ on the map and the corresponding colour-coded scans in the third row. As can be seen from the temporally closest camera images, place recognition can be extremely challenging in vision due to limited field-of-view, lens-glare and other environmental conditions. Using radar data, we are not faced with the same challenges.

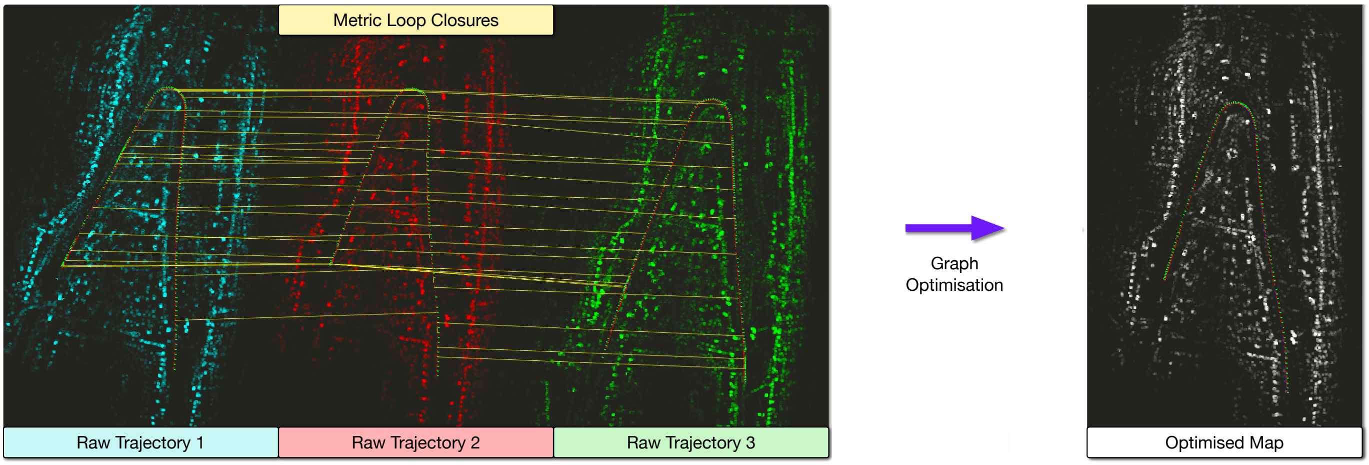


Fig. 7. Full localisation and mapping system. Given sequential radar frames we can estimate open loop trajectories by composing radar odometry and show three such sections from the test datasets on the left at approximately the same location. All keypoints are rendered for each traversal weighted by keypoint scores (in cyan, red and green) and clearly highlight static structure, such as walls and buildings, whilst attenuating empty and unobserved regions. When running our full system, as described in Section III-C, we detect metric loop closures shown as yellow lines (downsampled heavily for visualisation). All constraints are merged into a single map with pose graph optimisation, shown on the right in white, at well over real time speeds.

VI. Conclusions

In this paper we introduced the concept of learning keypoints for odometry estimation and localisation by embedding a differentiable pose estimator in our architecture. With this formulation, we learn to predict keypoint locations, scores and descriptors from pose information alone despite operating with extremely challenging radar data in complex environments. Over a large test set we improve on the state-of-the-art in *point-based* radar estimation by a large margin, reducing errors by up to 45%, whilst running an order of magnitude faster. Whilst we do not surpass the current state-of-the-art in *dense* radar odometry, we can solve poses at any rotation and detect metric loop closures, serving as a full system for radar-based mapping and localisation.

Furthermore, the benefits of our approach are not limited to radar or localisation tasks. The flexible architecture can be applied to most sensor modalities with few changes, and the detected points are readily reusable for other downstream tasks such as object velocity estimation. We plan to pursue these directions in the future, increasing radar based competencies for autonomous vehicles in urban environments.

VII. Acknowledgments

This work was supported by the UK EPSRC Doctoral Training Partnership and EPSRC Programme Grant (EP/M019918/1). The authors would like to acknowledge the use of Hartree Centre resources in carrying out this work.

References

[1] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset," *arXiv preprint arXiv: 1909.01300*, 2019.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523

[3] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *International Conference on Computer Vision*, Nov 2011, pp. 2564–2571.

[5] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Toward geometric deep slam," *arXiv preprint arXiv:1707.07410*, 2017.

[6] A. B. Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, "Key. net: Keypoint detection by handcrafted and learned cnn filters," *arXiv preprint arXiv:1904.00889*, 2019.

[7] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 224–236.

[8] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European Conference on Computer Vision*. Springer, 2016, pp. 467–483.

[9] S. Suwajanakorn, N. Snavely, J. J. Tompson, and M. Norouzi, "Discovery of latent 3d keypoints via end-to-end geometric reasoning," in *Advances in Neural Information Processing Systems*, 2018, pp. 2059–2070.

[10] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," *arXiv preprint arXiv:1905.03304*, 2019.

[11] 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.

[12] S. H. Cen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[13] D. Barnes, R. Weston, and I. Posner, "Masking by moving: Learning distraction-free radar odometry from pose information," *arXiv preprint arXiv: 1909.03752*, 2019.

[14] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[16] O. Sorkine-Hornung and M. Rabinovich, "Least-squares rigid motion using svd," https://igl.ethz.ch/projects/ARAP/svd_rot.pdf.

[17] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] 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.

[20] R. Arandjelović and A. Zisserman, "Dislocation: Scalable descriptor distinctiveness for location recognition," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 188–204.

[21] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla, "Learning and calibrating per-location classifiers for visual place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 907–914.

[22] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7286–7291.

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

[24] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," *2011 IEEE International Conference on Robotics and Automation*, pp. 3607–3613, 2011.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Under the Radar: Learning to Predict Robust Keypoints for Odometry Estimation and Metric Localisation in Radar
Publication Status	Submitted to the International Conference on Robotics and Automation (ICRA), 2020
Publication Details	Dan Barnes and Ingmar Posner Under the Radar: Learning to Predict Robust Keypoints for Odometry Estimation and Metric Localisation in Radar * International Conference on Robotics and Automation (ICRA), 2020

Student Confirmation

Student Name:	Daniel Barnes		
Contribution to the Paper	My contributions to the paper were: Developed the idea behind the paper Collected and processed data Performed the experiments Wrote the paper		
Signature Daniel Barnes		Date	18 / 11 / 2019

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Professor Ingmar Posner		
Supervisor comments	The description above is accurate.		
Signature		Date	7/1/2020

Summary and Future Work 8

In this thesis we demonstrated several real-world applications of leveraging domain knowledge for self-supervision in scalable robot learning. Not only do the presented approaches drastically improve existing systems or exceed the prior state-of-the-art, but the results are achieved without any manual supervision. In this chapter we summarise our contributions and discuss future work and extensions.

8.1 Drivable Path Segmentation

In Chapter 3 we demonstrated the ability of a self-supervised system to learn to segment drivable paths as well as obstacles in camera images. Despite being an easy task to conceptualise for humans due to both the informal and formal rules of the road, defining all possible road layouts and conditions as well as other agents in the scene is nearly impossible. By using human drivers as expert demonstrators and understanding the local scene geometry, the system learns a direct mapping of where to drive from raw camera data without any manual annotation. Furthermore, we demonstrated the application of this approach in different countries and in a wide variety of weather, lighting and traffic conditions; promising consistent drivable path prediction due to the large amount of training data generated in the self-supervised approach.

Predicting drivable paths and obstacles is only part of the puzzle for self-supervised planning and autonomy. An extension to this work would be to add depth prediction, and simply feed the outputs as additional cost terms into an existing planning framework. Furthermore, in the cases where multiple drivable paths are predicted, a high level controller (such as a satellite navigation device) could be used to navigate the vehicle along a long complex route. Finally, we wish to extend this approach to sensors with more complete observations of the environment, such as spherical cameras, lidar and radar. In doing so the system would have the capability to infer more complex local scene topology, and learn more realistic and human-like trajectories.

8.2 Egomotion with Learnt Ephemerality

In Chapter 4 we trained a system in a self-supervised fashion to segment ephemeral objects in camera images for improved visual odometry. In busy, cluttered, urban environments, large independently moving distractor objects, such as as buses, may occupy the majority of the scene. Using multi-session mapping to identify the static elements of the scene, our approach automatically labels and learns to predict good candidates for motion estimation, without requiring labelled prior maps or manual supervision. Furthermore, by training a model to predict metric depths from monocular camera imagery, our system produces robust real-time monocular VO in urban environments.

The benefits of ephemerality mask prediction are not limited to VO, and in future work we would like to exploit their use in other tasks. First, using ephemerality masks as a prior on foreground/background segmentation with the aim of using the background for longer timescale tasks, such as place recognition and dense mapping, and the foreground for more dynamic tasks such as object detection and tracking of agents in the scene. Secondly, since the approach makes no explicit assumption on using camera imagery, learning ephemerality masks in other modalities such as lidar presents an interesting avenue for research.

8.3 Large-Scale Urban Radar Dataset Collection

In Chapter 5 we released a large-scale millimetre-wave FMCW scanning radar dataset publicly to the research community. This class of radar holds promise in addressing some of the limitations other sensors face in challenging local conditions, and thus may enable safer operation through sensor redundancy and improved perception. Despite the existence of numerous large autonomous vehicle and robotics datasets, there are none that provide sufficient quantities of data in this modality for machine learning. We collected the dataset on over 280km of urban driving along with a full sensor suite including cameras, lidars and GPS/INS, to incentivise research (in particular robot learning) in this under-utilised modality. Furthermore, ground truth trajectories are automatically optimised without human supervision, providing a self-supervised pathway for the projects in Chapters 6 and 7.

There are numerous natural extensions to this work, both in terms of data collection and analysis of the dataset itself. Firstly, although the dataset is substantial compared to what was previously available to the research community, the data only includes urban city centre traversals. We would like to extend data collection to suburban, rural and even environments devoid of any infrastructure or roads. Secondly, we wish to update our radar to record Doppler to research accurate velocity estimation directly from raw sensor data. Lastly, we wish to further evaluate the application of radar against vision and lidar as a core sensor for autonomous robot navigation tasks, including: object detection, tracking, semantic segmentation, mapping and localisation.

8.4 Distraction-Free Radar Odometry Estimation

In Chapter 6 we showed that by embedding a radar masking network inside a differentiable pose estimation architecture, we achieve the state-of-the-art in RO in terms of both performance and speed. We took inspiration from learning ephemerality masks in Chapter 4 and the improvements to VO this brought. However, we noticed that the masks are learned for a proxy task rather than optimised for the pose estimator. We demonstrate that by using a differentiable pose estimator formulation, the radar masks are optimised directly for the task of RO. Furthermore, we qualitatively show the learned masks suppress sensing artefacts and distractor objects such as buses, leaving only the stable static structure in the scene, despite not having explicit Two-Dimensional (2D) mask supervision. Finally, we presented a method for producing calibrated uncertainties, crucial for real-world robot deployments, as well as alternative spectral and more classical RO implementations.

Despite delivering state-of-the-art radar odometry, there are some noticeable avenues we wish to investigate in the future. Firstly, training the RO system probabilistically, thereby explicitly learning calibrated pose uncertainties rather than calibrating uncertainties after training. Secondly, despite the clear inherent semantics in the predicted radar masks, there is no explicit constraint that they are temporally stable past sequential frames. One might find that by imposing explicit losses to promote mask consistency, the learned radar masks would become even more suitable for other applications. Finally, we aim to utilise these masks to suppress sensing artefacts in other radar applications such as place recognition and dense mapping, similar to the future applications of ephemerality masks in Section 8.2

8.5 Self-Supervised Radar Keypoint Learning

In Chapter 7 we showed that by by embedding a differentiable keypoint-based pose estimator inside a RO system, we can learn a keypoint detector in radar for improved radar odometry, mapping and localisation – all without manual supervision. We took inspiration from learning a masking network in Chapter 6 for state-of-the-art RO; however we noted that for a diverse range of tasks, keypoint representations are more reusable, and provide a condensed representation of the local scene. By taking this approach, we improved on the prior state-of-the-art in keypoint-based RO, whilst operating over an order of magnitude faster. Furthermore, the keypoint features are descriptive enough for place recognition, and we used this facility to build a real-time mapping and localisation system in radar.

Numerous directions present themselves when considering extensions to this work. First, as the formulation is not tailored specifically to radar, we would like to to apply the same technique to learning keypoints in other modalities such as vision and lidar. Secondly, unlike the dense RO formulation presented in Chapter 6, this method does not predict any notion of relative pose uncertainty, a crucial aspect for safe, real-world robot deployment. We hypothesise that by sampling an ensemble of pose predictions through dropping out point matches, the architecture could predict pose uncertainties without requiring architectural modifications. Finally, we would like to investigate learning regime and architectural design changes to improve training stability and convergence, likely by imposing more techniques from classical keypoint-based motion estimation approaches.

8.6 Discussion

In this thesis, we investigated the use of self-supervision for scalable robot learning on a variety of core tasks in urban autonomous driving. Through the intelligent use of domain knowledge, we apply expert systems to automatically label data where expensive human annotation would normally be required. By distilling this knowledge into DNNs used both in novel architectures and existing systems, we enable improved and state-of-the-art performance in numerous applications. Fundamentally, we believe this is the only practical approach that will enable the continued use of DNNs in robot systems in an ever growing number of applications and domains.

As identified in Chapter 1, our core contribution in this thesis has been to explore and present new applications of self-supervision to robot learning, rather than present an all-encompassing solution that eliminates the need for human annotation. Whilst the aim of avoiding manual labelling strongly aligns with the author's view on scalable robot learning, there will always be some need for it. In safety critical applications, the only method of evaluating true performance is to be equipped with a complete understanding of the world in which it operates; in some cases human annotation may be the only choice. Despite the successes in the applications we present, engineers must always be aware of the limits of self-supervision, and how best to evaluate systems developed this way before deploying them safely for use in the real world.

Nevertheless, we presently find ourselves in an extremely exciting and fast-paced time for applied robotics, and a large part of the reason for that has been the successful integration of DNNs for use in the real world. As robots are deployed more widely into the world and integrated more tightly within society there will be countless opportunities to further robot learning, and we believe self-supervision has an essential part to play.

References

[1] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection From RGB-D Data", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 918–927.

[2] M. Bansal, A. Krizhevsky, and A. Ogale, "ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst", in *Proceedings of Robotics: Science and Systems*, 2019.

[3] J. Hawke, R. Shen, C. Gurau, *et al.*, "Urban Driving with Conditional Imitation Learning", *arXiv preprint arXiv:1912.00177*, 2019.

[4] R. Kesten, M. Usman, J. Houston, *et al.*, *Lyft Level 5 AV Dataset 2019*, `https://level5.lyft.com/dataset/`, 2019.

[5] P. Sun, H. Kretzschmar, X. Dotiwalla, *et al.*, "Scalability in Perception for Autonomous Driving: An Open Dataset Benchmark", *arXiv preprint arXiv:1912.04838*, 2019.

[6] M. Cordts, M. Omran, S. Ramos, *et al.*, "The Cityscapes Dataset for Semantic Urban Scene Understanding", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[7] 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.

[8] H. Caesar, V. Bankiti, A. H. Lang, *et al.*, "nuScenes: A multimodal dataset for autonomous driving", *arXiv preprint arXiv:1903.11027*, 2019.

[9] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset", *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[10] C. McManus, W. Churchill, A. Napier, B. Davis, and P. Newman, "Distraction Suppression for Vision-Based Pose Estimation at City Scales", in *IEEE International Conference on Robotics and Automation*, 2013.

[11] S. H. Cen and P. Newman, "Precise Ego-Motion Estimation with Millimeter-Wave Radar Under Diverse and Challenging Conditions", in *IEEE International Conference on Robotics and Automation*, 2018.

[12] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time", in *Proceedings of Robotics: Science and Systems Conference*, 2014.

[13] J. Zhang and S. Singh, "Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast", in *IEEE International Conference on Robotics and Automation*, 2015.

[14] B. Yang, M. Liang, and R. Urtasun, "HDNET: Exploiting HD Maps for 3D Object Detection", in *Conference on Robot Learning*, 2018.

[15] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[16] P. Foucher, Y. Sebsadji, J.-P. Tarel, P. Charbonnier, and P. Nicolle, “Detection and Recognition of Urban Road Markings Using Images”, in *International IEEE Conference on Intelligent Transportation Systems*, 2011.

[17] S. Yenikaya, G. Yenikaya, and E. Düven, “Keeping the Vehicle on the Road: A Survey on On-Road Lane Detection Systems”, *ACM Computing Surveys*, vol. 46, no. 1, 2013.

[18] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, “Recent Progress in Road and Lane Detection - A survey”, *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014.

[19] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, “Spatial As Deep: Spatial CNN for Traffic Scene Understanding”, in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[21] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, “ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation”, *arXiv preprint arXiv:1606.02147*, 2016.

[22] D. A. Pomerleau, “ALVINN: An Autonomous Land Vehicle In a Neural Network”, in *Advances in Neural Information Processing Systems 1*, 1989.

[23] M. Bojarski, D. Del Testa, D. Dworakowski, *et al.*, “End to End Learning for Self-Driving Cars”, *arXiv preprint arXiv:1604.07316*, 2016.

[24] A. Bewley, J. Rigley, Y. Liu, *et al.*, “Learning to Drive from Simulation without Real World Labels”, in *IEEE International Conference on Robotics and Automation*, 2019.

[25] J. Engel, J. Sturm, and D. Cremers, “Semi-Dense Visual Odometry for a Monocular Camera”, in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[26] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast Semi-Direct Monocular Visual Odometry”, in *IEEE International Conference on Robotics and Automation*, 2014.

[27] D. G. Lowe, “Object Recognition from Local Scale-Invariant Features”, in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[28] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded Up Robust Features”, in *European Conference on Computer Vision*, 2006, pp. 404–417.

[29] E. Rosten and T. Drummond, “Machine Learning for High-Speed Corner Detection”, in *European Conference on Computer Vision*, 2006, pp. 430–443.

[30] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System”, *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[31] M. A. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[32] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras”, *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[33] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM”, in *European Conference on Computer Vision*, 2014, pp. 834–849.

[34] R. A. Newcombe, S. Izadi, O. Hilliges, *et al.*, “KinectFusion: Real-Time Dense Surface Mapping and Tracking”, in *IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[35] H. Alismail, M. Kaess, B. Browning, and S. Lucey, “Direct Visual Odometry in Low Light Using Binary Descriptors”, *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 444–451, 2017.

[36] G. Caron, A. Dame, and E. Marchand, “Direct Model Based Visual Tracking and Pose Estimation Using Mutual Information”, *Image and Vision Computing*, vol. 32, no. 1, pp. 54–63, 2014.

[37] G. Pascoe, W. Maddern, M. Tanner, P. Piniés, and P. Newman, “NID-SLAM: Robust Monocular SLAM Using Normalised Information Distance”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1435–1444.

[38] J. Callmer, D. Törnqvist, F. Gustafsson, H. Svensson, and P. Carlbom, “Radar SLAM using visual features”, *EURASIP Journal on Advances in Signal Processing*, no. 1, p. 71, 2011.

[39] F. Schuster, C. G. Keller, M. Rapp, M. Haueis, and C. Curio, “Landmark based Radar SLAM Using Graph Optimization”, in *IEEE International Conference on Intelligent Transportation Systems*, 2016, pp. 2559–2564.

[40] 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.

[41] M. Rapp, K. Dietmayer, M. Hahn, *et al.*, “FSCD and BASD: Robust Landmark Detection and Description on Radar-Based Grids”, in *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*, 2016, pp. 1–4.

[42] R. Rouveure, M. O. Monod, and P. Faure, “High Resolution Mapping of the Environment with a Ground-Based Radar Imager”, in *International Radar Conference "Surveillance for a Safer World"*, 2009, pp. 1–6.

[43] P. Checchin, F. Gérossier, C. Blanc, R. Chapuis, and L. Trassoudaine, “Radar Scan Matching SLAM Using the Fourier-Mellin Transform”, in *Field and Service Robotics*, A. Howard, K. Iagnemma, and A. Kelly, Eds., Springer Berlin Heidelberg, 2010, pp. 151–161.

[44] W. S. McCulloch and W. Pitts, “A Logical Calculus of the Ideas Immanent in Nervous Activity”, *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[45] B. Widrow and M. A. Lehr, “30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation”, *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[46] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning”, *Nature*, vol. 521, no. 7553, p. 436, 2015.

[47] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, “Gradient-Based Learning Applied to Document Recognition”, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[48] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, “Spatial Transformer Networks”, in *Advances in Neural Information Processing Systems 28*, 2015, pp. 2017–2025.

[49] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] K.-S. Oh and K. Jung, "GPU Implementation of Neural Networks", *Pattern Recognition*, vol. 37, no. 6, pp. 1311–1314, 2004.

[51] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition", *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[52] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification", in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[53] Y. Wu, M. Schuster, Z. Chen, *et al.*, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation", *arXiv preprint arXiv:1609.08144*, 2016.

[54] A. v. d. Oord, S. Dieleman, H. Zen, *et al.*, "WaveNet: A Generative Model for Raw Audio", *arXiv preprint arXiv:1609.03499*, 2016.

[55] P.-H. C. Chen, K. Gadepalli, R. MacDonald, *et al.*, "An augmented reality microscope with real-time artificial intelligence integration for cancer diagnosis", *Nature Medicine*, vol. 25, no. 9, pp. 1453–1457, 2019.

[56] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[57] A. Halevy, P. Norvig, and F. Pereira, "The Unreasonable Effectiveness of Data", *IEEE Intelligent Systems*, vol. 24, pp. 8–12, 2009.

[58] Fei-Fei Li, *ImageNet: crowdsourcing, benchmarking & other cool things*, `http://www.image-net.org/papers/ImageNet_2010.pdf`, [Online]. Accessed: 11-01-2020.

[59] Scale AI, *Scale AI Pricing*, `https://scale.com/pricing`, [Online]. Accessed: 11-01-2020.

[60] D. Barnes, W. Maddern, and I. Posner, "Find Your Own Way: Weakly-Supervised Segmentation of Path Proposals for Urban Autonomy", in *IEEE International Conference on Robotics and Automation*, 2017.

[61] 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 *IEEE International Conference on Robotics and Automation*, 2018.

[62] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset", *arXiv preprint arXiv: 1909.01300*, 2019.

[63] D. Barnes, R. Weston, and I. Posner, "Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information", in *Conference on Robot Learning*, 2019.

[64] H. Rohling, "Ordered Statistic CFAR Technique – an Overview", in *International Radar Symposium*, 2011, pp. 631–638.

[65] G. Bradski, "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, 2000.

Colophon

This thesis was typeset with a modified version of the *Clean Thesis* style developed by Ricardo Langner which can be downloaded at `http://cleanthesis.der-ric.de`.