Route Boundary Inference with Vision and LiDAR

Tarlan Suleymanov

Mansfield College

University of Oxford

Supervisor:

Professor Paul Newman

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2019

Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Paul Newman, for his insights, encouragement, endless support and guidance throughout my time in Oxford. Being part of the Oxford Robotics Institute (ORI) has been a privilege and I wish to thank everyone at ORI for being so welcoming and kind. I extend my deepest gratitude towards Lina Maria Paz and Pedro Piniés for all of their support and guidance during my first year at ORI. I am incredibly grateful to Lars Kunze for our valuable discussions regarding the research topics presented in this thesis. I would also like to thank my co-authors Paul Amayo, Tom Bruls and Geoff Hester. I want to thank Lars Kunze and Matthew Gadd for proof reading and constructive criticism of the manuscript. Finally, but most importantly thanks to my parents, Nizami and Elza, and my brother Habib, for their years of love and support; none of this would have been possible without them.

Abstract

The purpose of roads is to carry vehicles. Human drivers can easily distinguish roads and their components (e.g., surfaces, boundaries) using direct and indirect (contextual) clues as they are designed with driving in mind. The colour of road tarmacs, shape of road turns, smoothness of road surfaces, traffic signs, road boundaries, buildings, or even other vehicles provide clues about roads. For autonomous vehicles to safely navigate to a desired location in complex driving scenarios they are required to perceive their surrounding environment even in the presence of occlusions. This requires the use of contextual information in a similar fashion to human perception.

In this thesis, we focus primarily on road boundary detection and present a deep learning based approach to capture contextual information for dealing with occlusions. Many scenes present large-scale occlusion by other road users, preventing direct approaches from fully detecting road boundaries. Conventional neural network architectures fail to infer the exact location of an occluded, narrow, continuous curve running through the image. We tackle this problem with a coupled approach that generates multi-scale parameterised outputs in a discrete-continuous form. We combine the power of deep learning with the data obtained from our novel annotation framework to detect and infer road boundaries irrespective of whether or not the boundaries are visible by taking inspiration from human perception, which uses contextual information to perceive beyond the visible spectrum. Our semi-supervised data annotation framework leverages visual localisation and facilitates the use of deep networks by providing an efficient way to generate thousands of training samples. We present two road boundary detection approaches, camera-based and LiDAR-based, that capture scene context and achieve accurate results. We also demonstrate that the presented approaches have utility in scene understanding and localisation.

Contents

List of Figures

List of Abbreviations

ADAS . . . . . Advanced Driver Assistance Systems
CNN . . . . . Convolutional Neural Network
CORAL . . . . Convex Relaxation Algorithm
CPU . . . . . Central Processing Unit
CRF . . . . . Conditional Random Field
DEM . . . . . Digital Elevation Map
DNN . . . . . Deep Neural Network
GNSS . . . . . Global Navigation Satellite System
GPS . . . . . Global Positioning System
GPU . . . . . Graphics Processing Unit
ICP . . . . . Iterative Closest Point
IPM . . . . . Inverse Perspective Mapping
LDW . . . . . Lane Departure Warning
LiDAR . . . . Light Detection and Ranging
LKA . . . . . Lane Keeping Assist
LTS . . . . . Least Trimmed Squares
ML . . . . . Machine Learning
ORBI . . . . . Occluded Road Boundary Inference
ORI . . . . . Oxford Robotics Institute
RANSAC . . . Random Sampling and Consensus
RF . . . . . Random Forest
SLAM . . . . . Simultaneous Localisation and Mapping
TGV . . . . . Total Generalised Variation
TV . . . . . Total Variation
VGA . . . . . Video Graphics Array
VO . . . . . Visual Odometry
VRBD . . . . . Visible Road Boundary Detection

1 Introduction

Contents

1.1 Motivation

Major achievements in recent years in the fields of computer vision, machine learning and robotics are changing our interaction with cars from "driving" to "being driven". As a result, self-driving cars (Figure 1.2) are becoming a reality, but autonomous cars that are equipped with multiple sensors must tackle complex, real-world driving scenarios by accomplishing three main goals. First, these cars must know where they are located in the world. Second, they must perceive their surrounding environment and any obstacles within it. Finally, they must formulate a plan to safely navigate to a desired location. Accomplishing these goals requires tackling perception, localisation, and path planning challenges. This thesis mainly addresses perception. More specifically, this thesis is concerned with road boundary detection.

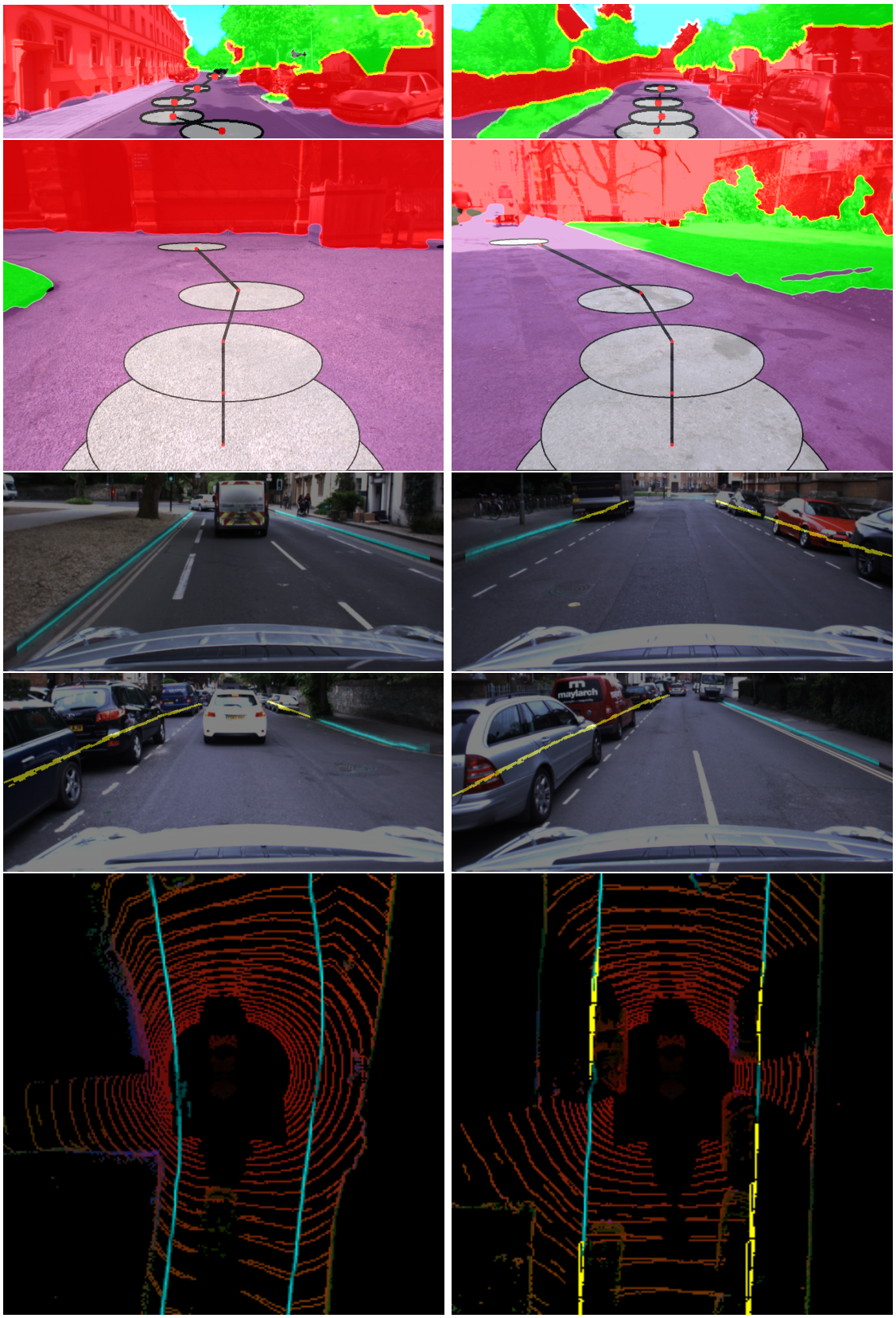


Figure 1.1: Examples of scene segmentation (first two rows) and road boundary detection (last three rows) outputs generated by approaches presented in this thesis.

The purpose of roads is to carry vehicles and they are designed with driving rules in mind. Although there are many types of roads with different physical characteristics (e.g., tarmac, dirt, gravel), they are easily distinguishable by human perception. The colour of tarmac, smoothness of the road surface, shape of the road turns, proximity to traffic signs or light poles, height differences with pavements, buildings or even other cars provide clues about roads. All of these direct or indirect (contextual) clues can be used to detect and infer roads or their boundaries that can be used for navigation in autonomous cars.

In this thesis we first addressed the problem using direct clues: colour, texture, and depth features to segment drivable areas that are directly observed from a stereo camera (Chapter 3). We used a combination of a Random Forest and a variational approach to segment collision-free, traversable paths from input images (Figure 1.1). In our second approach, we used both direct and contextual clues to be able to reason not just about visible areas, but also non-observable areas from an input sensor. Instead of segmenting the road surface we detected and inferred visible and occluded road boundaries from a camera (Chapter 5) or laser (Chapter 6) inputs. We present efficient methods for generating training data that allowed us to leverage the power of deep learning for capturing contextual information from inputs. This work was motivated by human perception, which is based on using context to reason about our surroundings.

1.2 Contributions

This thesis makes the following contributions:

- An on-line light-weight system that discovers and drives collision-free traversable paths using combination of a Random Forest and a variational approach for scene segmentation (Chapter 3),
- An annotation framework to easily generate road boundary masks for hundreds of camera images or LiDAR-based bird's-eye view images with one hour annotation (Chapter 4), which also enables to automatically obtain

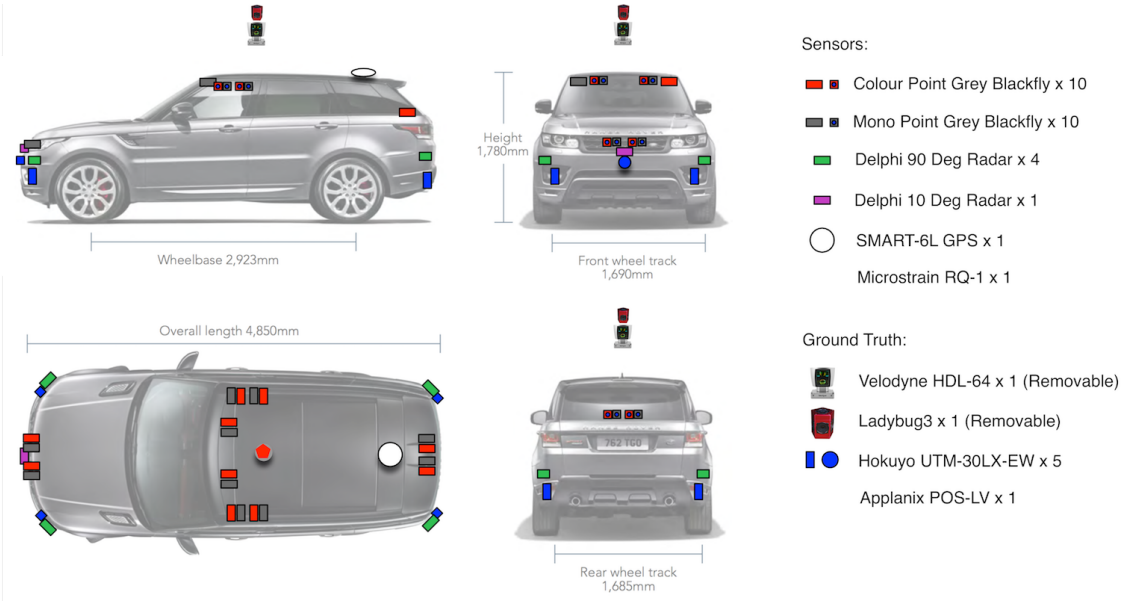


Figure 1.2: Example Autonomous Vehicle Platforms in use by ORI.

annotations for other traversals of a consistent route using a vision-based localiser (Chapter 4) and automatically partition raw camera-based road boundary masks into visible and occluded classes using a fully convolutional network (Chapters 5) or LiDAR-based road boundary masks using a hidden point removal algorithm (Chapter 6),

- A deep learning based approach to detect visible road boundaries using a camera or LiDAR without making any assumptions about their 3D structure, shape, or appearance (Chapters 5 and 6),

- A new deep learning architecture based on convolutional layers that captures contextual information using intra-layer convolutions and estimates occluded road boundaries in a multi-scale, parameterised, discrete-continuous form based on a single camera or LiDAR-based bird's-eye view image (Chapters 5 and 6),

- A model selection step to estimate geometric representation of road boundaries without making any assumption on the number of road boundaries in the scene (Chapter 5),

- Experimental lateral localisation based on road boundaries (Chapter 6), and
- Integration of the camera-based road boundary detection for the application of high-level semantic scene understanding (Chapter 5).

1.3 Publications

The following papers were published based on work presented in this thesis:

- T. Suleymanov, L. Kunze, and P. Newman, "Online Inference and Detection of Curbs in Partially Occluded Scenes with Sparse LIDAR," in IEEE International Conference on Intelligent Transportation Systems (ITSC), Auckland, New Zealand, 2019.
- T. Suleymanov, P. Amayo, and P. Newman, "Inferring Road Boundaries Through and Despite Traffic," in IEEE International Conference on Intelligent Transportation Systems (ITSC), Maui, Hawaii, USA, 2018.
- L. Kunze, T. Bruls, T. Suleymanov, and P. Newman, "Reading between the Lanes: Road Layout Reconstruction from Partially Segmented Scenes," in IEEE International Conference on Intelligent Transportation Systems (ITSC), Maui, Hawaii, USA, 2018.
- L. Kunze, T. Bruls, T. Suleymanov, and P. Newman, "Reading between the Lanes: Road Layout Reconstruction from Partially Segmented Scenes (Extended Abstract)," in Robotics Science and Systems (RSS) Workshop on Learning and Inference in Robotics: Integrating Structure, Priors and Models, 2018.
- T. Suleymanov, L. M. Paz, P. Piniés, G. Hester, and P. Newman, "The Path Less Taken: A Fast Variational Approach for Scene Segmentation Used for Closed Loop Control," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea, 2016.

- L. M. Paz, T. Suleymanov, P. Piniés, G. Hester, and P. Newman, "On-line Scene Understanding for Closed Loop Control," in Robotics: Science and Systems, Workshop on Geometry and Beyond: Representations, Physics, and Scene Understanding for Robotics, 2016.

1.4 Thesis Roadmap

In Chapter 2 we describe datasets that were used in training and testing of the presented approaches. Our scene segmentation method with a variational approach is presented in Chapter 3. In Chapter 4, we describe our annotation framework for obtaining thousands of samples to train our models and we introduce a camera and deep learning based approach that detects and infers road boundaries irrespective of boundary visibility in Chapter 5. A LiDAR-based deep learning approach for road boundary detection is presented in Chapter 6. To demonstrate the practical usability of our proposed approaches we applied the LiDAR-based approach to localisation in Chapter 6 and the camera-based road boundary detection approach to scene understanding in Chapter 5. Finally, Chapter 7 summarises the thesis and its contributions and suggests future work.

2 Datasets

Contents

2.1 Introduction

In this chapter we describe datasets that were used for training, validation, and testing of approaches presented in this thesis. Our first approach (Chapter 3), which combines a light-weight Random Forest and variational approach, was designed to rely on small amounts of training data for image segmentation. This enabled us to avoid time-consuming hand-annotation process of thousands of images, which is usually required for training of deep segmentation models. We trained our image classifier with small amounts of training data consisting of the publicly available KITTI [1] dataset and 65 manually annotated images from our Keble dataset (Section

Figure 2.1: Five sensors used in this thesis. From left to right: Point Grey Bumblebee XB3 colour stereo camera (top), Point Grey Bumblebee2 colour stereo camera (bottom), Velodyne VLP-32C 3D LiDAR, Velodyne HDL-32E 3D LiDAR, SICK LMS-151 2D LiDAR.

2.5). However, to take the advantage of deep learning techniques we developed an efficient way of semi-automating the annotation process of laser pointclouds and camera images to train our proposed road boundary detection models (Chapters 5 and 6). We present the detailed description of the annotation framework in Chapter 4. In the following sections we describe the sensors and datatsets that were used for the research in this thesis. For the summary of the datasets, see Table 2.1.

2.2 Sensor Overview

Sensors act as the "eyes" of robots and as such, without them, robots are "blind". Although there are many types of sensors that are used for autonomous driving, we used only cameras and lasers for our experiments in this thesis (Figure 2.1). In the following subsections (2.2.1 and 2.2.2), we discuss the inherent advantages and disadvantages when using these sensors.

2.2.1 Cameras

Cameras provide a reliable means of sensing the environment for a robot. They can perceive in a similar fashion to human vision, capturing images of a scene including such details as colours and texture. This enables the reading of street signs and interpretation of road markings. Cameras are inexpensive and cheaper than LiDARs, but they cannot directly provide the 3-dimensional (3D) structure

of a scene as LiDARs do. 3D structure can be computed using computer vision and machine learning techniques. Cameras are less affected by fog, rain, and snow, but highly dependent on lighting conditions. Therefore, strong shadows, bright sunlight, headlights of oncoming traffic, or an absence of light (e.g., at night) could potentially "blind" cameras. In our experiments we used Point Grey Bumblebee XB3 and Point Grey Bumblebee2 colour stereo cameras (Figure 2.1).

2.2.2 LiDAR

LiDAR (Light Detection and Ranging) is a time-of-flight sensor. It emits light pulses and measures the time it takes the light to travel. It calculates the distance to the nearest surface using the following formula:

$$d = \frac{ct}{2} \tag{2.1}$$

where c is the speed of light and t is the time taken for the light pulse to return. Unlike cameras, LiDARs yield a 3-dimensional image of an environment with high accuracy and precision. LiDARs are not affected by bright sunlight, headlights of other cars, shadows or darkness. However, fog, snow and rain adversely affect LiDAR beams. LiDARs do not provide the colour and texture information that cameras see, such as the words on a traffic sign or the colour of traffic lights. In our experiments we used VLP-32C 3D, Velodyne HDL-32E 3D and SICK LMS-151 2D LiDARs (Figure 2.1).

Table 2.1: Datasets that were used for training, validation, and testing of approaches presented in this thesis.

Dataset	Modality			Num. of frames	Annotation	Chapter	Used for
	Camera	2D LiDAR	3D LiDAR				
Oxford RobotCar 26-05-15	Yes	Yes	No	15K	Semi-automatic	4 & 5	Training
Oxford RobotCar 17-03-15	Yes	Yes	No	4K	Automatic	4 & 5	Training
Oxford RobotCar 08-05-15	Yes	Yes	No	1K	Automatic	4 & 5	Testing
Oxford RobotCar 19-05-15	Yes	Yes	No	4K	Automatic	4 & 5	Training
Oxford RobotCar 30-04-18	No	Yes	Yes	17K	Semi-automatic	4 & 6	Training
Oxford RobotCar 18-01-19	No	Yes	Yes	2K	Semi-automatic	4 & 6	Testing
Oxford RobotCar 24-08-18	No	No	Yes	1.8K	Semi-automatic	4 & 6	Train./Test.
Oxford 28-07-16	Yes	Yes	No	550	Semi-automatic	4 & 5	Testing
Cornbury	Yes	No	No	105	Manual	3	Train./Test.
Keble College	Yes	No	No	65	Manual	3	Train./Test.
KITTI	Yes	No	No	60	Manual	3	Train./Test.

2.3 The Oxford RobotCar Dataset

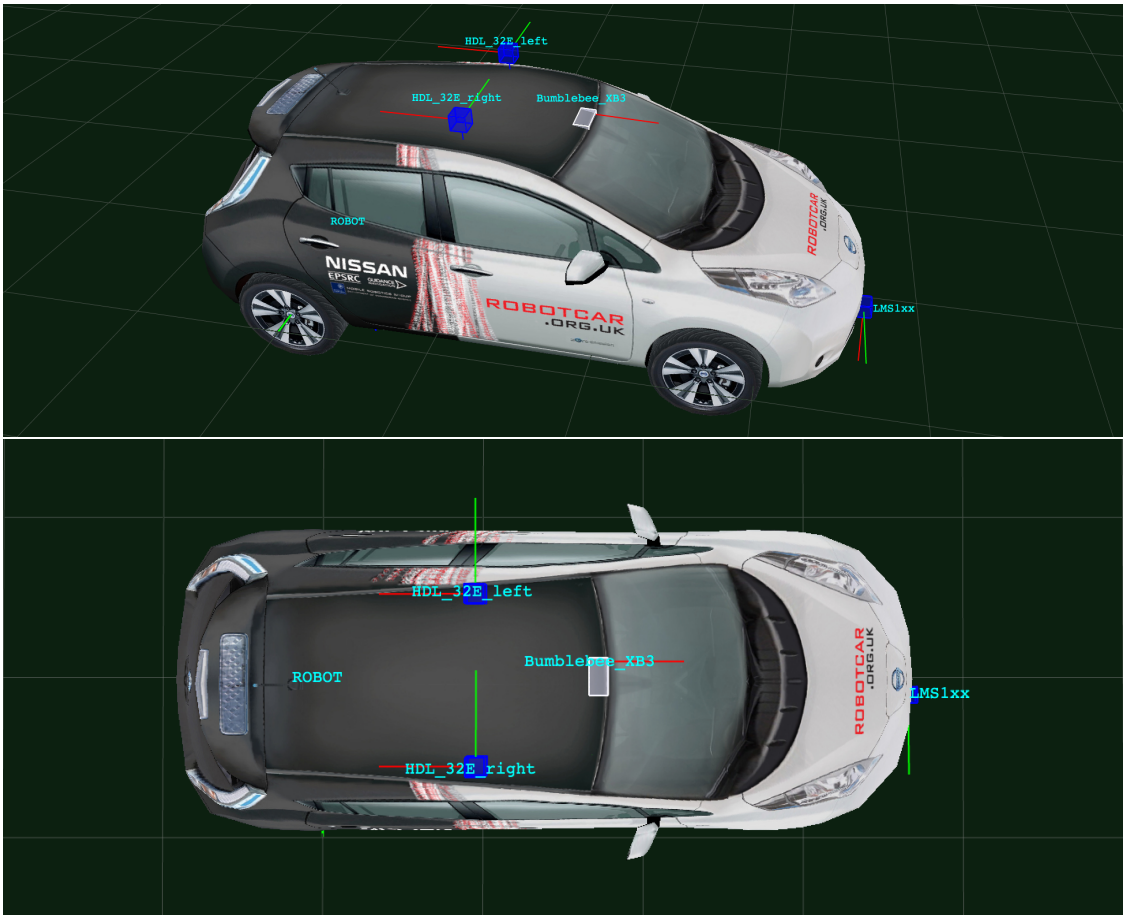


Figure 2.2: The modified Nissan LEAF data collection platform for the Oxford RobotCar Dataset and its sensor configuration. Note, only sensors that are relevant to this thesis are shown: a forward facing Point Grey Bumblebee XB3 colour stereo camera, two Velodyne HDL-32E 3D LiDARs on top and a SICK LMS-151 2D LiDAR vertically attached to the front of the car.

The Oxford RobotCar Dataset [2] consists of over 100 repetitions each of which is a 10 kilometre route through the Oxford city centre. It was collected throughout a year to capture weather, lighting, traffic, and other environmental changes as well as long-term changes (e.g., roadworks). The dataset was collected with a modified Nissan LEAF that was fitted with cameras and lasers (Figure 2.2). During the data collection the car was driven manually and the speed of the car did not exceed 30mph. There are other publicly available datasets of greater or similar size, but the uniqueness of the Oxford RobotCar Dataset is in its great number of traversals of a

Figure 2.3: The Oxford RobotCar Dataset examples. Each image is a view of the same location from different traversals of a 10 kilometre long consistent route in Oxford City.

consistent route in the city (Figure 2.3). We used 7 traversals from this dataset for our research from which approximately 45K ground truth samples were generated, a process which is described in Chapter 4, for training and testing of our proposed road boundary models. 15K semi-annotated camera images were generated from 26-05-15 dataset, which was used for training the camera-based road boundary approach. Further 8K semi-annotated camera images were automatically obtained for the training by projecting annotations from the 26-05-15 dataset to two other traversals: 17-03-15 and 19-05-15. Another traversal, 08-05-15, was automatically annotated for testing the camera-based road boundary approach, but only samples with fully annotated road boundaries were selected. For training and testing of the LiDAR-based road boundary approach, three datasets (30-04-18, 24-08-18 and 18-01-19) were annotated. Again, only samples with fully annotated road boundaries were selected for testing. Note that the training and testing samples for the LiDAR-based road boundary approach were from the different streets of the Oxford RobotCar Dataset. However, training and testing datasets for the camera-based approach included images from the same streets of Oxford but in different conditions and only 08-05-15 dataset was used for testing which was not included in the training set (see Table 2.1). Additionally, 550 samples were annotated from a separate dataset

(28-07-16) that was collected with the same Nissan LEAF platform but did not include the streets from the Oxford RobotCar Dataset. We used this dataset for quantitative evaluation of the camera-based approach.

Figure 2.4: Cornbury Dataset examples: RGB images with corresponding pixel-wise semantic class labels. There are 5 classes: tarmac road (purple), dirt road (cyan), grass (green), obstacles (maroon), and sky (grey). Pixels in black were not annotated and were not used during the training.

2.4 Cornbury Dataset

The Cornbury dataset was collected in a private area of Oxfordshire. It was used for testing our road segmentation approach as the dataset contained both tarmac and dirt roads. There were 105 images of the dataset that were hand-annotated to train our proposed road segmentation model (see Figure 2.4 for dataset samples). The dataset was collected with the same Nissan LEAF platform that was used for the Oxford RobotCar Dataset.

2.5 Keble College Dataset

The Keble College dataset was collected in a courtyard of one of Oxford University's constituent colleges, where an outdoor experiment for our road segmentation

Figure 2.5: Clearpath Husky UGV - the data collection platform for the Keble College Dataset. It has a maximum speed of 1.0 m/s.

approach was conducted. The dataset was collected with a Clearpath Husky UGV that was fitted with a tilted forward-facing Point Grey Bumblebee2 colour stereo camera, as shown in Figure 2.5. There were 65 images manually annotated for training our models; see Figure 2.6 for examples from this dataset and the corresponding annotated masks. Note that these images were coarsely annotated to save time and only annotated pixels were used during the training.

2.6 KITTI

From publicly available datasets, we made use of the KITTI dataset [1] for the proposed image segmentation approach. The KITTI Vision Benchmark Suite provides a benchmark system for various tasks in computer vision and mobile robotics. The KITTI datasets were captured in a mid-size city in Germany with

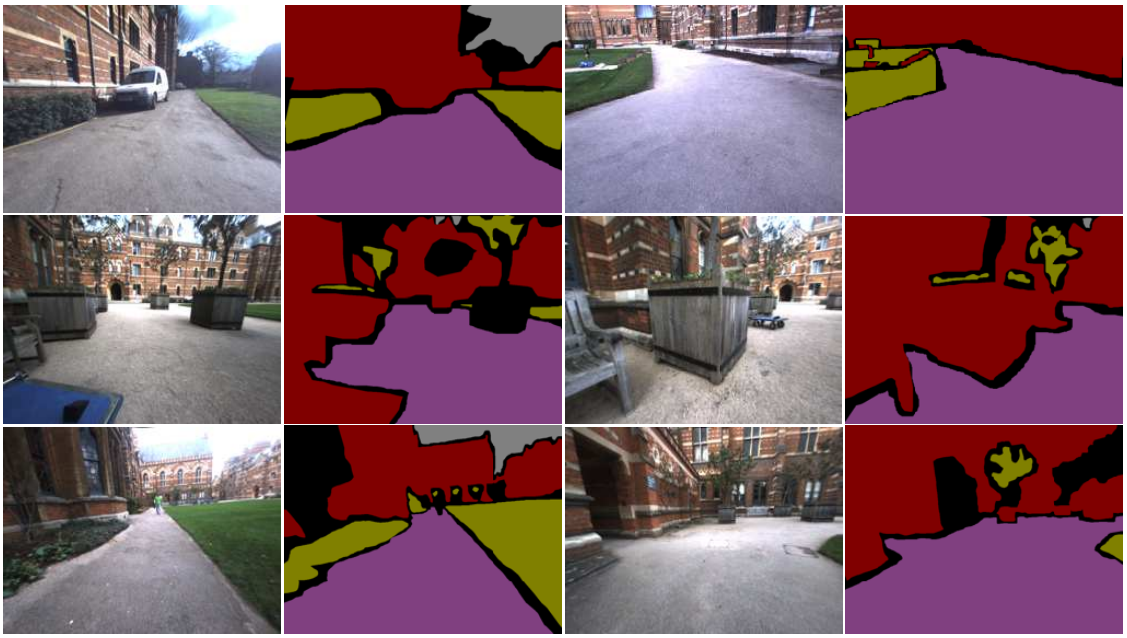


Figure 2.6: Keble College Dataset examples: RGB images with corresponding pixel-wise semantic labels. There are 4 classes: traversable area (purple), vegetation (green), obstacles (maroon), and sky (grey). Pixels in black were not annotated and were not used during the training.

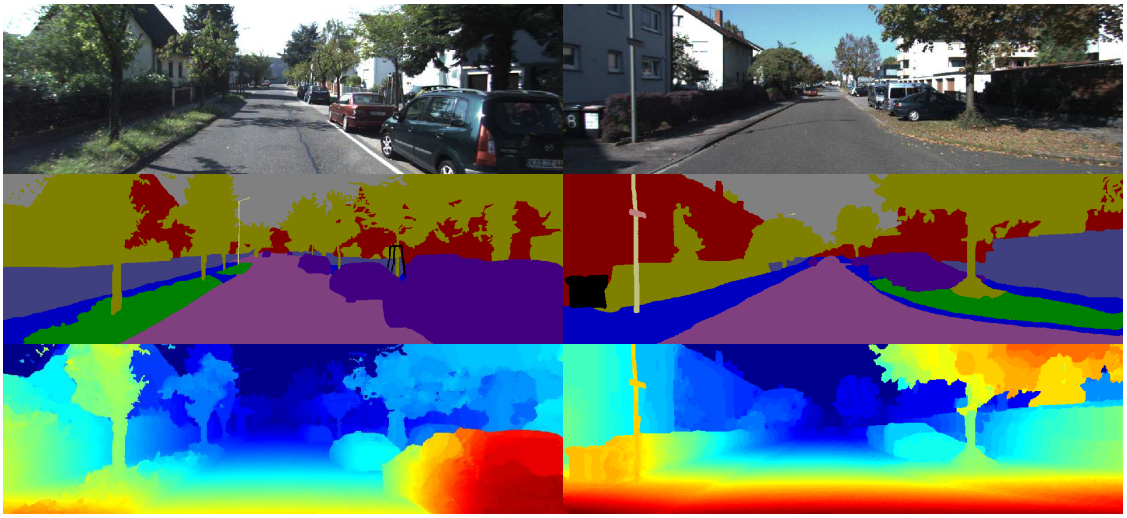


Figure 2.7: KITTI Dataset examples: RGB images (first row), semantic labels with 12 classes (second row) and depth maps (third row).

an autonomous driving platform. There were 60 stereo pairs from the KITTI dataset with perfect semantic class labels and ground truth depth maps were used for training and evaluation.

2.7 Conclusions

This chapter has presented four datasets that were used in the training and testing of our road segmentation and boundary detection systems. The largest of these datasets is the Oxford RobotCar dataset, consisting of over 100 traversals of a 10km route, but only 7 traversals were used here. Finally, 45K camera- and LiDAR-based ground truth samples were obtained from these traversals using our road boundary annotation framework that we presented in Chapter 4.

3

The Path Less Taken: A Fast Variational Approach for Scene Segmentation Used For Closed Loop Control

Contents

3.1 Introduction

This chapter is about our initial approach that examined the role of classical, non-deep learning approaches for road segmentation, which leveraged energy-based methods. The proposed approach integrates several modules including dense depth

estimation, semantic label prediction, image segmentation, route calculation, and robot control. It combines the use of a Random Forest with convex regularisation in a multi-labelling problem to obtain semantically segmented images. Beyond the obvious case of autonomous exploration, this path-following approach aims to provide a safety-net for a system which delivers a safe and coherent path to traverse in the temporary absence of a localiser. Therefore, generated collision-free paths are suitable for pure exploration and fall-back planning. The system relies on a camera alone and extracts feature channels from stereo images to encode depth, colour, and texture information which facilitates pixel-wise classification. The proposed approach is presented in Section 3.6, which was published and presented at International Conference on Intelligent Robots and Systems (IROS) 2016 [3].

The proposed approach consists of the following steps:

- Feature extraction from input images,
- Pixel-wise classification with a Random Forest,
- Regularisation via convex relaxation, and
- Collision-free local path planning and plan execution.

In the following section, we provide a review of and equations for feature extraction used in our proposed system.

3.2 Feature Extraction

Given a pair of stereo images as input, we extract feature channels to capture colour, texture, and 3D information for the pixel-wise classification as described below in subsections 3.2.1, 3.2.2, and 3.2.3.

3.2.1 Colour

RG Chromaticity. Normalising RGB values of pixels as in [4]:

$$r = R/(R+G+B) \qquad g = G/(R+G+B) \qquad b = B/(R+G+B) \tag{3.1}$$

where R, G, B are raw values and r, g, b are normalised, creates a chromaticity colour space in which red, green and blue colours are highlighted. This makes sky and vegetation more apparent and attenuates shadows and strong intensity changes. An example of chromaticity image is shown in Figure 3.1 (middle).

Figure 3.1: Input image (left) and extracted colour channels: RG chromaticity (middle), illumination invariant (right).

Illumination Invariant. Illumination Invariant (F) is a single-channel image that reduces the effect of shadows by applying eq.(3.2), given in [5]:

$$F = log(G) - \alpha log(B) - (1 - \alpha) log(R) \tag{3.2}$$

The parameter α in the above equation should satisfy the following equation:

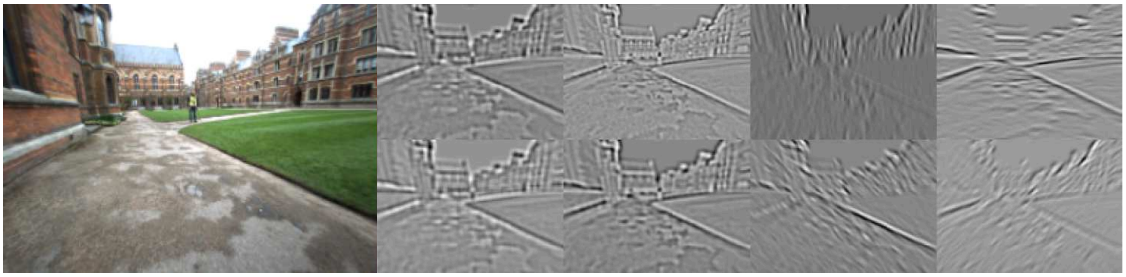
$$\frac{1}{\lambda_2} = \frac{\alpha}{\lambda_1} + \frac{(1 - \alpha)}{\lambda_3} \tag{3.3}$$

where λ_i is the peak spectral response of each sensor channel provided by the manufacturer. An example of the output is shown in Figure 3.1 (right).

3.2.2 Texture

To encode texture information, a subset of eight edge and blob filter kernels (Figure 3.2a) from the filter bank given in [6] - which includes Gaussian and Laplacian of Gaussian filters at various scales - was applied to the input images. See Figure 3.2b for examples of generated feature channels.

(a) Filter kernels.



(b) Texture feature channels.

Figure 3.2: Eight filter kernels (a) from a Leung-Malik filter bank [6] are applied to grayscale images that create eight texture feature channels (b).

3.2.3 3D Information

3D geometry of a scene is a rich source of information for the pixel-wise classification because some parts of a scene (e.g., road, path, grass) have flat horizontal surfaces, where the other parts (e.g., trees, obstacles) have flat vertical or non-flat surfaces. Using 3D information helps to distinguish between labels and to avoid physically unreasonable segmentation. To obtain depth feature channels a disparity map needs to be generated first (top right in Figure 3.3).

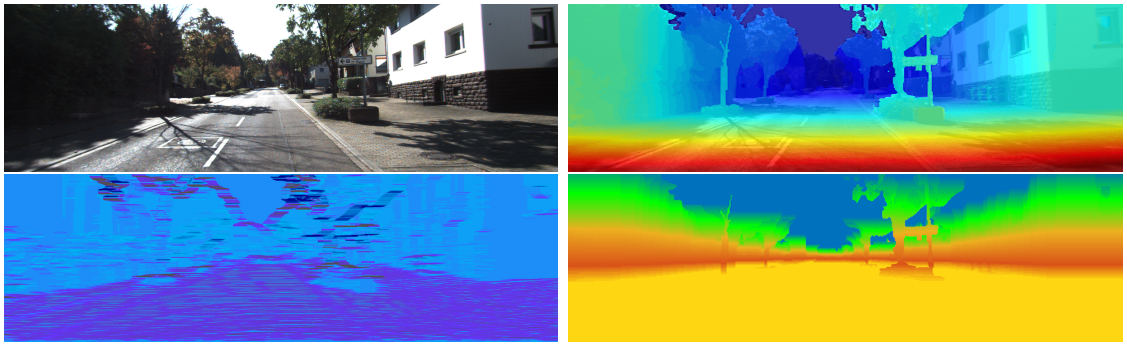


Figure 3.3: Input image (top left) and extracted depth channels: vertical disparity gradient (bottom left), disparity map (top right), and height above the ground plane (bottom right).

Depth Map Estimation. A depth map is an image in which the pixel values represent the distance from the viewpoint of a camera to the surface of objects.

As part of the feature extraction step, we used the depth map estimation method presented in [7] to obtain depth maps of a scene and extract 3D feature channels from them. This method formulates a global energy function that jointly optimises regularisation and data terms, where the regularisation term is a prior model that captures the structure of the scene and the data term represents the similarity between corresponding pixels in the stereo images.

Height above the ground plane. Physical objects are placed on top of each other or the ground and vertical ordering of an object is encoded by transforming each pixel to its height from the ground plane [4]. The resulting visualisation of a result of this transformation is shown in Figure 3.3 (bottom right).

Vertical Disparity Gradient. The vertical gradient of the disparity map is another strong feature channel that provides information helpful for ground plane detection, as gradient remains constant for a flat surface [8]. A 3 x 31 window was applied to obtain robust estimations. Figure 3.3 (bottom left) shows an example of the estimated feature channel.

3.3 Pixel-wise Classification

Having extracted the feature channels, the pixel-wise classification with a Random Forest (RF) [9] was applied to predict labels. RF is a popular ML method for regression and classification and it consists of an ensemble of decision trees. Combining separate decision trees as an ensemble improves performance and prevents over-fitting [9]. Trees in our RF were trained separately with a deterministic procedure, but to prevent them becoming identical we used a bootstrap procedure to train the trees on different sets of features. During the training, we randomly selected feature channels for each tree. Additionally, features were randomly selected for each node. As a result, these two procedures prevent the trees in the RF model becoming identical. During inference, a voting strategy was used to calculate the probability of an input pixel belonging to a particular label.

3.4 Regularisation via Convex Relaxation

After obtaining per-pixel classification probabilities, a regularisation step was applied to improve the output results. We formulated a global energy function that jointly optimised the per-pixel probabilities yielded by the RF model and the perimeter of the labelled segments. Applying the regularisation balanced the smoothness of the segments and per-pixel probabilities which resulted in denoised and improved segmentation results.

3.5 Path Planning and Execution

Table 3.1: Hardware specifications and average running time per task

	Server	Laptop
Hardware		
OS	Ubuntu 14.01	OSX Mavericks
Processor	Intel(R) Core(TM) i7 CPU @ 3.50GHz	Intel(R) Core(TM) i7 CPU @ 2.3GHz
Graphics Card	GeForce GTX TITAN Black, 6144 MB 2880 CUDA Cores	Geforce 750M, 2048 MB, 384 CUDA Cores
Running time		
Depth Map Estimation	190 ms (5.26 Hz)	1180 ms (0.85 Hz)
Feature Extraction	25 ms (40 Hz)	22 ms (45.45 Hz)
Label Probability prediction	10 ms (100 Hz)	6 ms (166 Hz)
Label Regularisation	155 ms (6.45 Hz)	1020 ms (0.98 Hz)
Route calculation	$\approx$ 5 ms (200 Hz)	$\approx$ 5 ms (200 Hz)

The final step in the system was the extraction of collision-free paths based on the segmented ground label. A simple strategy was applied to derive a drivable path. The ground label was tessellated in cells and the centre of mass was calculated for each cell. Then, we connected the valid centre of mass of the points to obtain the collision-free paths. This strategy generated paths by considering orientation and shape of ground label. In our live outdoor experiment with the Clearpath Husky UGV (Figure 2.5), the generated paths were executed with a constant linear velocity of 1.0 m/s and an angular velocity that was derived from the path segments. The overall system was running at $\approx$1Hz on the laptop in the experiment and was only limited by the frame rate of the dense depth estimation. Hardware specifications and average frame rate per task are given in Table 3.1.

The nature of the system as described here can be explored in more detail in the paper that follows. The paper presents the proposed scene segmentation approach in more detail and provides experimental results. The updated pipeline

of the system and further experimental results were investigated in the discussion that concludes this chapter.

The Path Less Taken: A Fast Variational Approach for Scene Segmentation Used For Closed Loop Control

Tarlan Suleymanov† Lina Maria Paz† Pedro Piniés† Geoff Hester† Paul Newman†

***Abstract*— In this paper we propose an on-line system that discovers and drives collision-free traversable paths, using a variational approach to dense stereo vision. Our system is light weight, can be run on low cost hardware and is remarkably quick to predict the semantics. In addition to the scene's path affordance it yields a segmentation of the local scene as a composite of distinctive labels – e.g, ground, sky, obstacles and vegetation. To estimate the labels, we combine a very fast and light weight (shallow) image classifier which considers informative feature channels derived from colour images and dense depth maps estimates. Unlike other approaches, we do not use local descriptors around pixel features. Instead, we encompass label-predicted probabilities with a variational approach for image segmentation. Akin to dense depth map estimation, we obtain semantically segmented images by means of convex regularisation. We show how our system can rapidly obtain the required semantics and paths at VGA resolution. Extensive experiments on the KITTI dataset support the robustness of our system to derive collision-free local routes. An accompanied video supports the robustness of the system at live execution in an outdoor experiment.**

I. Introduction

A fundamental task for a mobile robot is the ability to find and follow drivable or collision-free paths. In this paper, we propose a vision-based system that, via a variational approach, is able to segment and label semantically distinctive parts of the local scene including paths through it. Our motivation, beyond the obvious case of autonomous exploration, is the creation of a safety-net process which in the temporary absences of a localiser can still execute a safe and coherent path through its workspace. Figure 1 illustrates our approach for a single image frame.

Our system integrates different modules including dense local mapping, semantic label prediction, image segmentation, route calculation and robot control. A stereo camera is used as the primary sensing modality in this paper. Stereo cameras can provide an inexpensive and reliable means of sensing the environment for a robot at true scale if appropriately fast reliable processing schemes are deployed. Over the years, novel theoretical foundations of continuous optimisation [1], [2] and machine learning [3] for image analysis, upon which the most advanced algorithms rely, have become accessible for robotics and computer vision applications. In addition, the continuous development in parallel computing allows us to build systems that can respond in soft real time. Our algorithm for path discovery works in outdoor environments taking advantage of multiple, complementary depth and colour cues. We use these cues in a multi-label image segmentation approach. The problem is formulated in a probabilistic framework that combines machine learning with convex regularisation. In order to learn distinctive scene labels, we rely on shallow classifiers such as Random Forests (RF) [4]. This choice is driven by the intrinsic property of the RFs as low variance classifiers. As a result, they provide better generalisation by preventing from the undesirable overfitting problem. In addition, RFs explicitly allow us to model pixel-wise label probabilities with frequentist inference [5]. Moreover, RFs can easily adapt themselves to architectures supporting parallel computing and multi-threading to rapidly predict the per-pixel label probabilities. We summarise our contributions as follows:

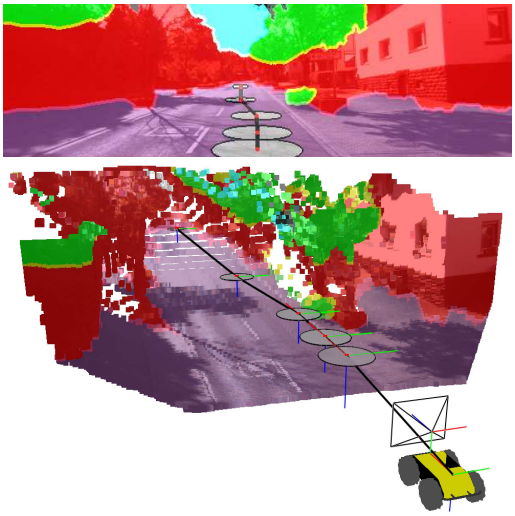


Fig. 1: Our approach combines the use of a shallow classifier with convex relaxation in a multi-labelling problem to obtain semantically segmented images. Given the labels describing drivable regions, we deliver collision-free local routes to the robot controller. Top, a segmented image with the plausible path. Bottom, a dense representation of the local scene.

- We demonstrate the ability of our system to run continuous optimisation at two different tasks in reasonable execution times –i.e. dense depth map estimation and multi-labeling image segmentation.
- We derive plausible routes by analysing the image semantics corresponding to drivable regions (e.g. road, ground).

We analyse the ability of RFs to combine multiple features leading to a further increase in performance when colour

†Mobile Robotics Group
Department of Engineering Science
University of Oxford
17 Parks Road, Oxford
OX1 3PJ, United Kingdom
tarlan,linapaz,ppinies,gjh,pnewman@robots.ox.ac.uk

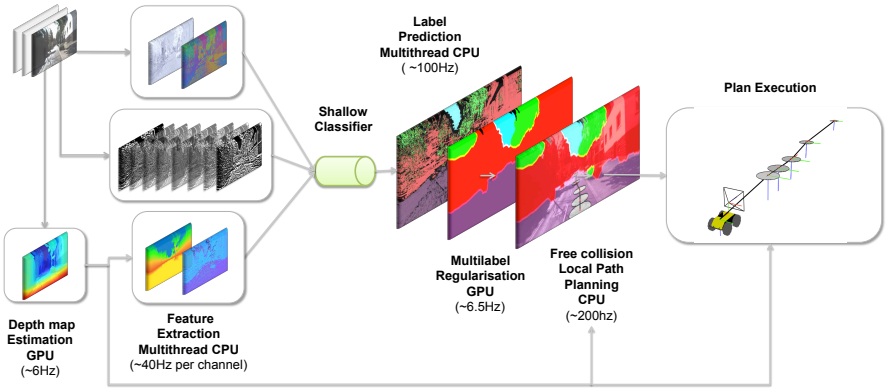


Fig. 2: Scene understanding pipeline for collision-free route following. The system considers different modules running in a main CPU multiple thread process. Stereo pairs are first processed in a dense depth map estimation task. In parallel, several CPU threads can process the rectified images to extract different colour and depth contextual feature channels. A different task is used to train our shallow Random Forest. During on-line mode, the output of the Random Forest produces the predicted per-pixel label probabilities. A new task uses this information to produced a regularised image segmentation. Given the image segmentation solution, we analyse the ground label and extract a feasible path. Finally we send the path to the robot controller.

and depth features are used simultaneously. We show how our system can rapidly obtain the required semantics – and therefore paths– at VGA resolution. Extensive experiments on the KITTI dataset support the robustness of our system to derive collision-free local routes. An accompanied video supports the robustness of the system at live execution in an outdoor experiment with a wheeled robot exploring over hundreds of metres of trajectory.

II. Related Work

Over the past few years, there has been an increasing development of path-following algorithms. Many of these algorithms are not necessarily adaptive. Some rely on a priori knowledge of specific visual characteristics such as lane markers or road boundaries of the road surface [6] or their geometric structure using complementary sensor modalities such as LIDAR [7]. Other approaches employ supervised learning techniques to learn to recognise a desired class of roads by exploiting colour cues unique to the road surface in combination with segmentation algorithms [8].

In this paper, we take advantage of the two frameworks by combining dense local geometry and image colour cues. We note, however, that our ultimate goal is to find a drivable path with no assumption of any particular structure – i.e, no lane or border information is used as prior. Therefore, we deliver only collision-free paths that are suitable for pure exploration, fall-back planning (localiser failure) and off-road applications.

Related approaches have been presented in the past. For instance, in [9] a fast path following strategy for unstructured scenes is formulated as a posteriori distribution for the path given semi-dense stereo disparity, image texture and orientation features. The texture feature described is designed to adopt colour information, starting at reliably classified seed points which are provided by simple disparity based segmentation into road plane and obstacle. A different algorithm is proposed in [10] where a 1D trifocal tensor is used to estimate parameters required for the path-following controller, through a structure from motion approach, without having to explicitly recover the 3D structure of the environment.

The focus of much of our work is the development of a path-following algorithm using scene understanding through image segmentation. Common approaches use information from dense stereo maps with Conditional Random Fields (CRFs) [11] or Convolutional Neural Nets (CNNs) [12], [13] to obtain a reasonable image segmentation – at the expense of higher computational cost to predict the per-pixel labels. A good assumption is that many scenes are a composite of vertical surfaces –e.g. buildings, vehicles, pedestrians– w.r.t the horizontal ground –e.g., road and sidewalk– with possible parts of the sky [12]. Analogously, we model the appearance of the ground using cues at pixel–level, such as colour and texture, together with contextual information from dense depth maps – in fact, they play an important role in our image segmentation task. In this work, because we require realtime performance, and in contrast to [12], [13], we use a shallow classifier rather than a deep classifier [14] to provide the data term into a down stream semantic regularisation formulated as continuous convex relaxation. Similar solutions are encountered in the literature. In [15] for instance, a set of five channels are extracted from depthmaps, however the approach does not take advantage of other colour features. Moreover, the multilabel segmentation problem is addressed as a combination of a RF classifier and a graph cut MRF based approach. In this paper, we present a solution that exploits continuous multilabel optimisation that has been shown to be superior in terms of parallelisation and runtime performance [16].

III. System Overview

Our intermediate (but welcome) goal is to provide per-pixel semantics for the simple application of exploration with a mobile robot at near real time. To this end, we design

a system consisting of several tasks running in a multi-thread process as illustrated in Figure 2. Each left and right image of the stereo pair I_{lr}, is processed in a parallel task to estimate a dense depth map ξ. In this paper we extend the approach presented in [17] –whose solution relies on continuous energy minimisation– to estimate stereo depth maps with TGV regularisation. This choice allows us to approximate locally the ground label as an affine surface. Such approach also exploits the use of the Augmented Lagrangian (AL) method to accelerate the convergence of the primal-dual algorithm. The algorithm supports per-pixel calculations, therefore allowing us to run the task on an available GPU.

Simultaneously, several CPU threads process the left image to extract different features channels $\mathbf{Z}$ consisting of colour $\mathbf{Z}_{rgb}$, location $\mathbf{Z}_{loc}$, filter-banks $\mathbf{Z}_f$ and depth-context features $\mathbf{Z}_\xi$.

The channels are received by a different task in charge of training our shallow Random Forest. During on-line mode, the output of the Random Forest produces per-pixel u probabilities $P_{\mathfrak{T}}(u \in L_i|\mathbf{z}_u)$ of u belonging to a set of labels $L_i,\ i \in \{1,\ldots,K\}$ where K is the number of labels. A final task uses this information to estimate the regularised image segmentation. Analogous to the depth map estimation task, we run the regularisation on the same GPU. Given the segmentation solution, we analyse the ground label and extract a feasible path. Finally we send the path to the robot controller.

IV. Predicting Labels with a Random Forest

A Random Forest (RF) is a popular machine learning method for classification and regression, which consists of an ensemble of decision trees $\mathfrak{T}_j,\ j \in \{1 \cdots T\}$ with predefined tree depth $d_{\mathfrak{T}}$. It has been shown that combining separate decision trees to form a forest improves performance of prediction and prevents over-fitting [4]. In our RF, each tree $\mathfrak{T}_j$ is trained individually. We follow the classical construction of the decision tree as a deterministic procedure. In order to prevent having identical trees, our trees are trained on different set of per-pixel features $\mathbf{Z}$ using a bootstrap procedure. For each tree, the same number of pixels as in the original set is randomly selected by sampling with replacement. As a result, some feature samples may appear several times, while some others could be absent. In addition, we randomly select features in each node inducing several node searching splits. Figure 3 illustrates this process.

A. Scene Features

A set of feature channels is used in order to obtain informative information about the scene by applying various transformations on the colour and depth map images. Let $\mathbf{z}_u \in \mathbf{Z}$ be a per-pixel feature vector defined as

$$\mathbf{z}_u = \begin{bmatrix} \mathbf{z}_{rgb}^T & \mathbf{z}_{loc}^T & \mathbf{z}_f^T & \mathbf{z}_\xi^T \end{bmatrix} \tag{1}$$

where $\mathbf{z}_{rgb}$ comprises an illumination invariant transform $\mathbf{z}_{ill\_inv}$ and a rg-chromaticity transform $\mathbf{z}_{rg-chroma}$ applied over the rgb pixel channels. $\mathbf{z}_\xi$ is represented by two contextual transforms over the depth: first, the height of the 3D back-projection of the pixel w.r.t the ground $\mathbf{z}_{hg}$; second, the vertical disparity gradient $\mathbf{z}_{vg}$. In addition, we estimate the distance from the pixel to the horizon line $\mathbf{z}_{loc}$. Finally, we use the Leung-Malik (LM) filter bank $\mathbf{z}_f$, a collection of Gaussian and Laplacian of Gaussian filters at various scales and orientations to represent the local texture.

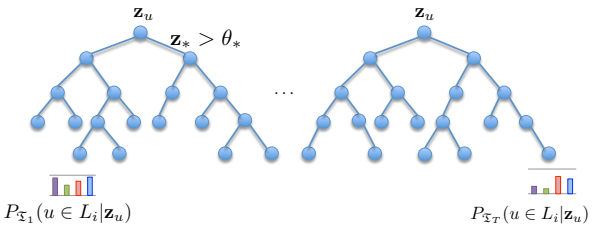


Fig. 3: The RF classifier uses per-pixel features derived from colour and depth cues. Given a new pixel sample $\mathbf{z}$, each tree classifies it in a different leaf. Each leaf saves a histogram modelling the class distribution over the samples.

B. Estimation of label distribution

The probability $P_{\mathfrak{T}}(u \in L_i|\mathbf{z}_u)$ of a pixel u belonging to a particular label L_i is the result of a voting strategy. For each tree in the forest $\mathfrak{T}_j$, a subset of the components of the feature vector $\mathbf{z}_u$ are compared at each node to a given threshold θ. The comparison determines the next branch to follow until a leaf node is reached. As can be seen in Figure 3, histograms learnt during the training phase are stored at the leaves of the trees. For a given tree, the histograms contain the number of pixels per label in the training set that end up in that leaf. These histograms aim to approximate the probability $P_{\mathfrak{T}_j}(u \in L_i|\mathbf{z}_u)$. During on-line mode, the label distribution of a test pixel is given by the average of the histograms stored at the corresponding leaf of each tree in the forest:

$$P_{\mathfrak{T}}(u \in L_i|\mathbf{z}_u) = \frac{1}{T}\sum_{j=1}^{T} P_{\mathfrak{T}_j}(u \in L_i|\mathbf{z}_u) \tag{2}$$

V. Regularisation via Convex Relaxation

With the initial per-pixel classification results in hand, greatly improved results can be obtained by formulating the complete image segmentation as a labelling problem with a global energy function that balances "smoothness" of the labelled segments (a prior) and per-pixel probabilities (a data term) coming from the random forest. The energy function is given by:

$$\min_{\Omega_i} \left\{ \frac{1}{2}\sum_{i=1}^{K} Per(\Omega_i) + \sum_{i=1}^{K} \int_{\Omega_i} f_i(u)du \right\} \tag{3}$$
$$s.t.\ \Omega = \bigcup_{i=1}^{K} \Omega_i, \quad \Omega_i \cap \Omega_j = \emptyset, \quad \forall i \neq j$$

where $\Omega \in \mathbb{R}^2$ represents all the pixels in the image assigned to K disjoint regions Ω_i (e.g. ground, vegetation, obstacles and sky).

In Eq.(3) the data term is given by the sum of the costs of the unary potentials $f_i(u) = -\log(P_{\mathfrak{T}}(u \in L_i|\mathbf{z}_u))$ per segmented region Ω_i. The intuition behind $f_i(u)$ is that when pixel u belongs to region Ω_i with a high probability ($P_{\mathfrak{T}}(u \in L_i|\mathbf{z}_u) \approx 1$) the cost added is negligible ($f_i(u) \approx 0$), on the contrary, low probabilities produce an increasingly unbounded cost. The main effect of the smoothness term is to reduce the perimeter of the regions $Per(\Omega_i)$ such that it tends to smooth the boundary between neighbours and delete small regions surrounded by bigger ones. In order to obtain a more convenient expression of the energy for optimisation we represent each region instead by its indicator function:

$$\phi_i(u) = \begin{cases} 1 & \text{if } u \in \Omega_i \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The energy function in Eq.(3) can then be described by:

$$\min_{\phi_i(u)} \left\{ \frac{1}{2} \sum_{i=1}^{K} \int_\Omega |\nabla \phi_i(u)| du + \sum_{i=1}^{K} \int_\Omega \phi_i(u) f_i(u) du \right\} \tag{5}$$
$$s.t.\ \phi_i(u) \in \{0,1\}, \quad \sum_{i=1}^{K} \phi_i(u) = 1$$

where $\int_\Omega |\nabla \phi_i(u)|$ is the Total Variation (TV) of the indicator function $\phi_i(u)$ that can be shown to be equal to the perimeter of the segment.

The constraint $\phi_i(u) \in \{0,1\}$ makes the problem combinatorial and NP-hard so it can only be approximately solved. We use a known fast relaxation approach [18] that transforms the original problem into a convex one. While this relaxation is not the tightest, it produces good results in practice. The relaxation is based on allowing $\phi_i(u)$ to take values in the interval $\phi_i(u) \in [0,1]$. Since in addition we know that $\sum_{i=1}^{K} \phi_i(u) = 1$, the constraint can be relaxed to:

$$\phi_i(u) \geq 0, \ \forall i \tag{6}$$

as a result, Eq.(5) becomes a convex optimisation problem.

Unfortunately the energy cost is non-smooth due to the L1 norm that appears in the TV term. The Legendre-Fenchel Transform [19] allows us to trade the non-smoothness of the prior term for a smooth convex constrained maximisation:

$$\int_\Omega |\nabla \phi_i(u)| du = \max_{\Psi_i(u)} \int_\Omega \nabla \phi_i(u) \cdot \Psi_i(u) du \tag{7}$$
$$s.t.\ |\Psi_i(u)|_{2,1} \leq 1 \tag{8}$$

where $\Psi_i(u) : \Omega \to \mathbb{R}^2$ is known as the dual function of $\phi_i(u)$. Although this transformation seems to apparently increase the complexity, the counterpart is that we can now use well known first order methods available for smooth problems to find the global solution of the relaxed energy.

As explained below, we can easily deal with the box constraints given in Eqs.(6, 8) by projecting the solution of the optimisation at each iteration to the feasible set when it fails to meet the constraints. The equality constraint $\sum_{i=1}^{K} \phi_i(u) = 1$ can be included into the energy by introducing Lagrange Multipliers $\Gamma(u)$. The relaxed problem to be solved is then:

$$\min_{\phi_i(u)} \max_{\Psi_i(u), \Gamma(u)} \left\{ \frac{1}{2} \sum_{i=1}^{K} \int_\Omega \nabla \phi_i(u) \cdot \Psi_i(u) du + \sum_{i=1}^{K} \int_\Omega \phi_i(u) f_i(u) du \right.$$
$$\left. + \int_\Omega \Gamma(u) \left(\sum_{i=1}^{K} \phi_i(u) - 1 \right) du \right\} \tag{9}$$
$$s.t.\ \phi_i(u) \geq 0, \quad |\Psi_i(u)|_{2,1} \leq 1$$

To solve the convex saddle point problem in Eq. (9) an iterative primal dual algorithm [1] is applied. Basically we just need to interleave gradient ascent steps for the maximisations with gradient descent steps for the minimisation at each iteration of the algorithm. In both cases we project the solution to the feasible set in case the box inequality constraints are not met. Therefore at each iteration t and per each label L_i we perform the following steps:

- Maximising $\Psi_i(u)$:
$$\tilde{\Psi}_i^{t+1} = \Psi_i^t + \sigma \nabla \tilde{\phi}_i^t \quad \text{gradient ascent}$$
$$\Psi_i^{t+1} = \frac{\tilde{\Psi}_i^{t+1}}{\max(1, |\tilde{\Psi}_i^{t+1}|_{2,1})} \quad \text{projection to feasible set}$$
- Minimising $\phi_i(u)$:
$$\tilde{\phi}_i^{t+1} = \phi_i^t - \tau(\nabla^T \Psi^{t+1} + f_i + \Gamma_i^t) \quad \text{gradient descent}$$
$$\phi_i^{t+1} = \max(0, \tilde{\phi}_i^{t+1}) \quad \text{projection to feasible set}$$
- Maximising $\Gamma(u)$:
$$\Gamma^{t+1} = \Gamma^t + \mu \left(\sum_{i=1}^{K} \phi_i^{t+1} - 1 \right) \quad \text{gradient ascent}$$
- Over-relaxation:
$$\bar{\phi}^{t+1} = \phi^{t+1} + \theta_r(\phi^{t+1} - \phi^t)$$

where σ, τ and μ control the step size of the gradient steps. In practice, each variable is updated by performing pixel wise calculations while the gradient operator ∇ is approximated by finite differences. The parameters are set up to $1/2, 1/4$ and $1/5$ respectively through the use of preconditioning [20]. The last over-relaxation step allows faster convergence of the algorithm with $0 \leq \theta_r \leq 1$. Analogous to the depth map estimation problem, the primal dual approach followed in this section allows us to take advantage of general purpose GPU hardware for parallel computing. For a detailed derivation of the update equations, we refer the interested reader to [20].

VI. Experiments

This section provides quantitative results for experiments carried out over two different datasets. Our heterogeneous pipeline (CPU/GPU) was tested in two different hardware architectures whose details are summarised in Table I. In addition, we show qualitative results from a live experiment carried out in an outdoor environment. Our experiments consider a parameter selection analysis as well as an assessment that guides the importance of the feature channels over the RF-based dataterm. Additionally, we make a comparison of predicted labels before and after label regularisation.

TABLE I: Hardware architectures

Architecture	OS	Processor	Graphics Card
Server	Ubuntu 14.01	Intel(R) Core(TM) i7 CPU @ 3.50GHz	GeForce GTX TITAN Black, 6144 MB 2880 CUDA Cores
Laptop	OSX Mavericks	Intel(R) Core(TM) i7 CPU @ 2.3GHz	Geforce 750M, 2048 MB, 384 CUDA Cores

A. *Quantitative experiments*

In order to evaluate the proposed image segmentation approach, we make use of a subset of the *KITTI dataset* for which ground truth is provided [21]. The dataset consists of 60 stereo pairs at resolution 1241×376 with perfect annotations for 12 semantic class labels and ground truth depth maps. For the purpose of our evaluation, we synthesise the annotations into ground, vegetation, obstacles and sky. In this case, we employ 50% of the images for training and 50% for cross-validation. An additional dataset was collected and manually labelled, *Keble College dataset*, consisting of 65 stereo frames at high resolution (1280×960). Optimised depth maps are also provided with centimetre accuracy.

1) Parameter selection for RF-based dataterm: The accuracy and computational complexity of our RF-based dataterm depends on two major parameters: the maximum tree depth and the number of trees in the forest. In order to choose the best parameters, we analyse their impact over the RF performance. For the KITTI dataset, we carry out an exhaustive search in a 2-dimensional grid representing the parameter domain. For each parameter configuration (number of trees, tree depth) we train a model. Figure 4 shows the average F1 score on the configuration space of parameters over 200 RF models. Although the performance can be optimised to find the maximum score, we found that a classifier consisting of 5 trees and 10 tree levels provides a good trade off between performance and speed.

2) Contribution of feature channels: Our assessment also takes into account the contribution of each feature channel

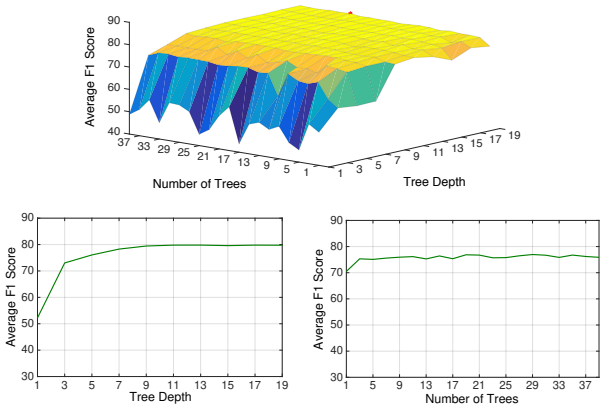


Fig. 4: Analysis of the performance of several RF-based classifiers for parameter selection. We carried out an exhaustive search in a 2-dimensional array representing the parameter space domain. For each parameter configuration (number of trees, tree depth), we train the classifier and evaluate its average F1-score. Although the performance can optimised by finding the maximum score, an a classifier consisting of 10 trees and 15 tree levels can result in a good trade off between performance and speed.

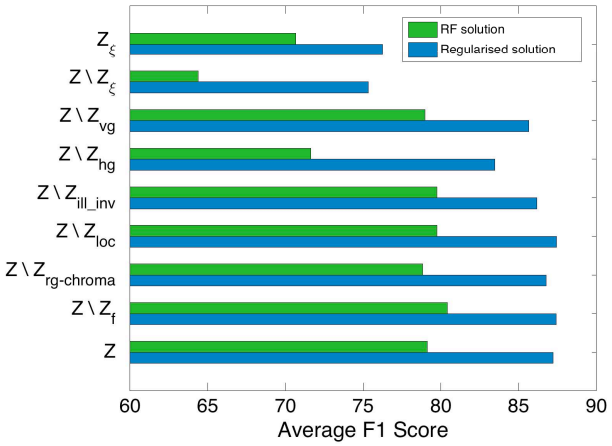


Fig. 5: Impact of the feature channels over the performance before (green) and after (blue) multi-label regularisation. In all cases, the smoothed solution outperforms the RF-classifier solution. Much worse results are obtained when depth features are left out.

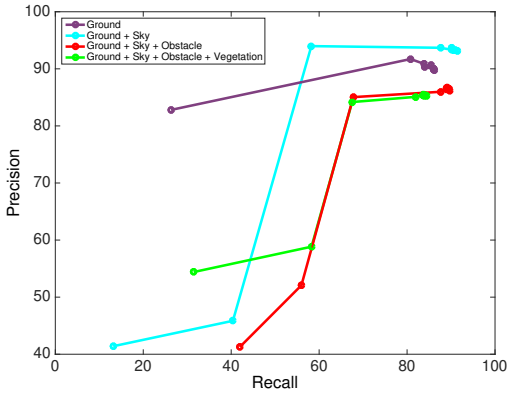


Fig. 6: Precision-Recall curves obtained by increasing the complexity of the RFs for different combination of labels. As expected, the curves exhibit better accuracy for more complex models preventing at the same time the undesirable over-fitting problem.

to the overall performance. We calculate the precision, recall and F1 score for different models. Table II details this values per label. An important part of the evaluation is to measure the impact of the depth channels (i.e, $\mathbf{z}_\xi$). Moreover, we emphasise the importance of each depth channel by leaving one channel out at a time (e.g, $\mathbf{Z} \setminus \mathbf{z}_{hground}$). This test is also applied to the colour channels allowing us to find strong features by observing whether or not the performance drops significantly when a particular channel is missing. Figure 5 summarises this information showing the average F1 score for the KITTI dataset. Note that the overall performance significantly decreases when the depth channels are not considered. In fact, a model that uses only depth channels

performs better than a model using the rest of the channels. We can also see that texture features do not appreciably affect the performance.

Table II provides information about the accuracy achieved per label. For instance, when only depth channels are used, the ground and the sky labels already exhibit high accuracy. This is not surprising if we consider that these labels are associated to very distinctive depths. In contrast, the accuracy obtained for the obstacle and vegetation labels when just depth channels are used is lower as there is much higher variability in their depth.

Figure 6 shows how the precision and recall curves vary when we increase the complexity of the RFs for different combination of labels. As expected, the curves exhibit better generalisation (better accuracy) for increasing complex models preventing at the same time the undesirable over-fitting problem.

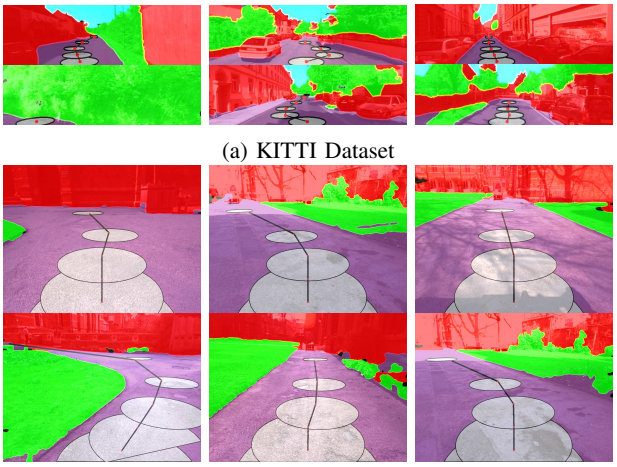
3) Prediction labels before and after regularisation: Table II and Figure 5 indicates the difference between the performance of the models before and after applying regularisation. Note that in all the cases, the regularised solution outperforms the RF-classifier solution.

4) Running Time: Many of the tasks involved in our pipeline can run in real time. Table III summarises the running times per task for the two testing architectures. All images are at VGA (480×640) resolution. Despite stereo depth estimation and label smoothing require more time than other tasks, the frame rates are still acceptable to provide reliable paths at live execution.

B. Live outdoor experiments

Fig. 7: Illustration of the collision-free algorithm to extract a route from the ground label.

For our live outdoor experiment we use a Clearpath Husky UGV equipped with a forward-facing PGR Bumblebee2 camera. In order to provide reliable collision-free paths, we implement an algorithm that analyses the ground label in a bottom-up direction. Figure 7 illustrates the process. The algorithm tessellates the ground label in cells with adjustable dimensions. Note that each cell contains only a sub-region of the segmented label. For each cell, we calculate the centre of mass. This simple strategy allows us to consider the shape and orientation of the drivable regions. All the points with valid centre of mass are concatenated together to form the desired path. In addition, we impose a safety margin over the robot dimensions. The robot is modelled as a circular object of 1.5 meters of diameter. Each feasible point of the circle is projected into the segmented image. We check for possible collisions if the projection intersects other labels. The path is back-projected to 3D space using the available depth map and the intrinsic camera parameters. For simplicity, our



(a) KITTI Dataset

(b) Live experiment

Fig. 8: Qualitative results of the path following approach. We first evaluated the collision-free approach in different scenarios. In some situations the ground segment extends in front of the robot, thus our algorithm succeeds to find a route to control the robot forwards. We show that in the presence of obstacles our approach provides collision-free paths. When the ground segment is small – for example when the robot reaches a wall– the estimated path can only provide one or no points. In this case, the robot performs pure rotation until it finds a ground region for which a plausible path can be estimated.

controller assumes a differential platform such that the path is executed with a constant linear velocity of 1 m/s. The angular velocity is derived from the path segments.

The collision-free path approach has been tested in different scenarios. Figure 8 shows qualitative results on the KITTI dataset and on our live experiment with a robot moving autonomously in a quad along hundreds of meters. When the ground segment extends in front of the robot and there are no obstacles, the algorithm succeeds to find the simplest route possible and plans to move the robot forwards following a straight line. When obstacles such as cars are present, our approach is able to over take the obstacles following a collision-free path. Finally, when the ground segment in front of the robot is not big enough – for example when the robot reaches a wall– no path can be provided. In this case, the robot performs a pure rotation until it finds a ground region for which a plausible path can be estimated.

VII. Conclusions

In this paper, we presented a general framework that combines a light weight (shallow) image classifier with convex regularisation for the general problem of image scene understanding. While recent approaches rely on the use of complex and deep classifiers, we demonstrate that a random forest can inform our variational formulation with very reliable label probabilities. In fact, our system requires small amounts of data during the training phase and yet produces high accurate results during testing. Finally, we showed that our system is remarkably fast to provide semantics form image data allowing a mobile robot to discover and drive collision-free traversable paths. We are also interested in comparing to alternative solutions based on deep classifiers

TABLE II: Recall, precision and F1 score for different RF models

	Ground			Obstacle			Vegetation			Sky		
Channels	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
KITTI Dataset												
$\mathbf{Z}_{\xi}$	91.83	87.41	89.56	69.59	65.03	67.23	57.24	63.65	60.27	98.20	98.20	98.20
	91.40	87.23	89.27	82.36	70.28	75.84	58.27	76.58	66.19	93.36	97.39	95.34
$\mathbf{Z} \setminus \mathbf{Z}_{\xi}$	57.29	53.25	55.20	66.49	59.72	62.93	65.98	79.15	71.97	62.89	60.02	61.42
	69.38	63.16	66.12	76.86	70.05	73.30	77.20	92.21	84.04	61.03	79.58	69.08
$\mathbf{Z} \setminus \mathbf{z}_{vg}$	89.49	89.75	89.62	83.56	72.00	77.35	66.68	79.70	72.61	93.86	98.52	96.13
	91.28	89.77	90.52	90.71	79.56	84.77	75.74	91.88	83.03	86.80	98.43	92.25
$\mathbf{Z} \setminus \mathbf{z}_{hg}$	71.03	74.24	72.60	78.63	66.36	71.97	66.16	78.60	71.85	58.40	66.86	62.35
	80.66	89.84	85.01	90.66	76.79	83.15	78.78	91.71	84.76	60.00	82.38	69.43
$\mathbf{Z} \setminus \mathbf{z}_{ill_inv}$	90.45	90.46	90.46	82.91	73.98	78.19	69.04	78.64	73.53	93.59	98.50	95.98
	92.61	90.78	91.68	92.89	79.73	85.81	73.99	93.10	82.45	88.53	98.49	93.24
$\mathbf{Z} \setminus \mathbf{z}_{loc}$	90.34	90.26	90.30	83.35	73.64	78.19	68.77	79.37	73.69	92.44	98.08	95.17
	92.02	91.92	91.97	91.37	82.47	86.69	79.18	91.74	85.00	88.84	98.21	93.29
$\mathbf{Z} \setminus \mathbf{z}_{rg-chroma}$	90.62	89.94	90.28	82.41	72.63	77.21	66.58	77.85	71.77	96.33	98.31	97.31
	92.71	90.99	91.84	91.18	81.78	86.22	77.35	90.98	83.62	89.06	98.26	93.44
$\mathbf{Z} \setminus \mathbf{z}_{f}$	90.49	91.01	90.75	82.65	75.04	78.66	70.61	78.81	74.48	97.58	98.28	97.93
	92.45	92.20	92.33	91.02	82.67	86.64	78.87	91.29	84.63	92.17	96.87	94.46
$\mathbf{Z}$	89.62	90.80	90.20	84.67	72.07	77.86	65.68	79.66	72.00	95.32	98.42	96.85
	92.01	92.32	92.17	91.13	82.24	86.46	78.60	91.34	84.49	90.61	97.32	93.85
Keble College Dataset (Full Resolution)												
$\mathbf{Z}$	99.10	94.90	96.95	89.06	93.22	91.10	78.02	98.68	87.14	—	—	—
Keble College Dataset (VGA Resolution)												
$\mathbf{Z}$	98.48	94.55	96.48	86.65	94.13	90.24	73.96	98.71	84.56	—	—	—

We compared the impact of the channels over the performance before (white rows) and after (gray rows) applying regularisation. For a better analysis, we show the independent precision-recall and F1 score per label. The first column describes the channels used for training. For instance, $\mathbf{Z}$ indicates that all channels have been used, while $\mathbf{Z} \setminus \mathbf{z}_{*}$ means that a particular channel has been left out. $\mathbf{z}_{ill_inv}$ is the illumination invariant transform, $\mathbf{z}_{rg-chroma}$ is the rg-chromaticity transform, $\mathbf{z}_{\xi}$ is represented by two contextual transforms over the depth. $\mathbf{z}_{hg}$ is the height of the 3D back-projection of the pixel w.r.t the ground. $\mathbf{z}_{vg}$ is the vertical disparity gradient. $\mathbf{z}_{loc}$ the distance from the pixel to the horizon line. $\mathbf{z}_{f}$ are the Leung-Malik (LM) filter bank $\mathbf{z}_{f}$.

TABLE III: Average Running time per task

Task	Server (ms (Hz))	Laptop (ms (Hz))	CPU Threads
Depth map estimation	190 ms (5.26 Hz)	1180 ms (0.85 Hz)	1
Feature Extraction	25 ms (40 Hz)	22 ms (45.45 Hz)	14
Label Probability prediction	10 ms (100 Hz)	6 ms (166 Hz)	10
Label Regularisation	155 ms (6.45 Hz)	1020 ms (0.98 Hz)	1
Route calculation	$\approx$ 5 ms (200 Hz)	$\approx$ 5 ms (200 Hz)	1

in combination with our variational approach. This will be part of our future work.

REFERENCES

[1] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, May 2011.

[2] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM J. Img. Sci.*, vol. 3, no. 3, pp. 492–526, Sep. 2010.

[3] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests." in *BMVC*, 2008, pp. 1–10.

[4] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] H. Boström, "Estimating class probabilities in random forests," in *Sixth Int. Conf. on Machine Learning and Applications, 2007. ICMLA 2007*. IEEE, 2007, pp. 211–216.

[6] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. on Image Processing*, vol. 19, no. 8, pp. 2211–2220, 2010.

[7] P. Y. Shinzato, D. F. Wolf, and C. Stiller, "Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion," in *IEEE Proc. Intelligent Vehicles Symposium*, 2014, pp. 687–692.

[8] R. Mohan, "Deep deconvolutional networks for scene parsing," *arXiv preprint arXiv:1411.4101*, 2014.

[9] B. Hummel, S. Kammel, T. Dang, C. Duchow, and C. Stiller, "Vision-based path-planning in unstructured environments," in *IEEE Proc. Intelligent Vehicles Symposium*, 2006, pp. 176–181.

[10] D. Sabatta and R. Siegwart, "Vision-based path following using the 1d trifocal tensor," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 3095–3102.

[11] L. Ladickỳ, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. Torr, "Joint optimization for object class segmentation and dense stereo reconstruction," *Int. Journal of Computer Vision*, vol. 100, no. 2, pp. 122–133, 2012.

[12] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 376–389.

[13] A. Kendall, V. Badrinarayanan, , and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[14] T. Scharwachter and U. Franke, "Low-level fusion of color, texture and depth for robust road scene understanding," in *IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 599–604.

[15] C. Zhang, L. Wang, and R. Yang, "Semantic segmentation of urban scenes using dense depth maps," in *ECCV 11th European Conference on Computer Vision*, 2010.

[16] C. Nieuwenhuis, E. Toeppe, and D. Cremers, "A survey and comparison of discrete and continuous multi-label optimization approaches for the potts model," *Int. Journal of Computer Vision*, vol. 104, no. 3, pp. 223–240, 2013.

[17] P. Piniés, L. M. Paz, and P. Newman, "Dense and swift mapping with monocular vision," in *Int. Conf. on Field and Service Robotics (FSR). Toronto, ON, Canada*, 2015.

[18] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer, "Fast global labeling for real-time stereo using multiple plane sweeps." in *VMV*, 2008, pp. 243–252.

[19] R. T. Rockafellar, *Convex Analysis*. Princeton, New Jersey: Princeton University Press, 1970.

[20] T. Pock and A. Chambolle, "Diagonal preconditioning for first order primal-dual algorithms in convex optimization," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2011, pp. 1762–1769.

[21] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2014, pp. 89–96.

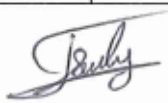
Statement of Authorship for joint/multi-authored papers for PGR thesis

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 Path Less Taken: A Fast Variational Approach for Scene Segmentation Used for Closed Loop Control
Publication Status	☑ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	T. Suleymanov, L. M. Paz, P. Piniés, G. Hester, and P. Newman, "The Path Less Taken: A Fast Variational Approach for Scene Segmentation Used for Closed Loop Control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, 2016.

Student Confirmation

Student Name:	Tarlan Suleymanov		
Contribution to the Paper	My contributions to the paper were: Developing the initial idea behind the paper. Dataset annotation and preparation. Running the experiments. Performing the analysis. Writing the paper, making the figures.		
Signature		Date	12 September 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: Prof Paul Newman		
Supervisor comments I AGREE		
Signature 	Date	12/9/2019

This completed form should be included in the thesis, at the end of the relevant chapter.

3.8 Summary of the Paper's Results

The experimental results presented in the paper demonstrated that the proposed approach required small amounts of training data in order to operate in the predefined environments in which it was trained. The scene segmentation approach produced highly accurate results with the F1 score greater than 0.85 overall. The quantitative experiments demonstrated that the regularisation step always improved the accuracy of the outputs. The live outdoor experiments and qualitative results showed the practical use of the approach in the real-world scenario in which the system could provide collision-free paths for navigation of the autonomous platform.

3.9 Further Improvements and Experiments

The analyses of the system described in the paper allowed us to improve different aspects of the system. First, we observed that there was significant improvement when adding the raw RGB channels as input feature channels. We removed texture and pixel location features as they do not appreciably affect the performance of the classification. The depth map estimation method was switched from the approach presented in [10] to the method described in [7] that generates better depth maps in shorter time. Additionally, the delay caused by consecutive processes in the pipeline was significantly reduced by changing the structure of connections between tasks.

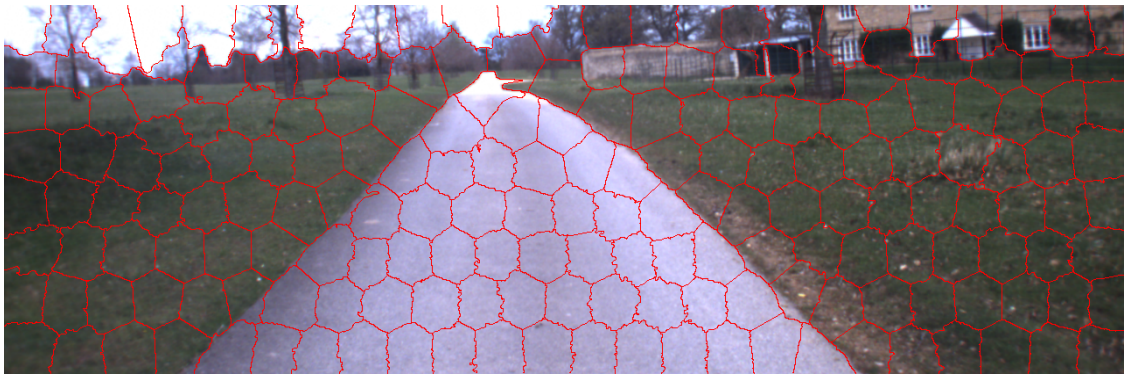


Figure 3.4: Extracted superpixels in an unstructured scene using the SLIC algorithm [11].

Although the pixel-wise classification scores provided by the RF were regularised, the optimisation required many iterations to produce a highly accurate segmentation. This is due in part to the calculation of the features channels, as no local information was being exploited. Local information around pixels produces more consistent features to train the classifier with the added benefit of reducing the noise [12]. A popular choice to encode local information is the extraction of superpixels from images as they maintain true region boundaries. The use of superpixels implicitly removes the noise and reduces the complexity by capturing redundancy of similar image regions. The updated version of the path following system leverages this property. The SLIC algorithm [11] was chosen for this purpose and Figure 3.4 illustrates the superpixels extracted in an unstructured scene. This results in a practical change that significantly reduced the number of predictions performed by the classifier. For instance, a 640 x 380 image requires 2,000 predictions instead of 243,200 assuming that 2,000 superpixels are extracted (Figure 3.5).

Figure 3.6 shows the results of the segmentation over images from the Cornbury Dataset (see Chapter 2). Two types of traversable paths were identified: asphalt road and dirt road. In order to detect both types of roads separately, a fifth label was introduced in this experiment. From the Cornbury dataset, 105 images were hand-annotated to train new RF models. Overall, the updated version ran at 5Hz frame rate and was only limited by the frame rate of the dense depth estimation.

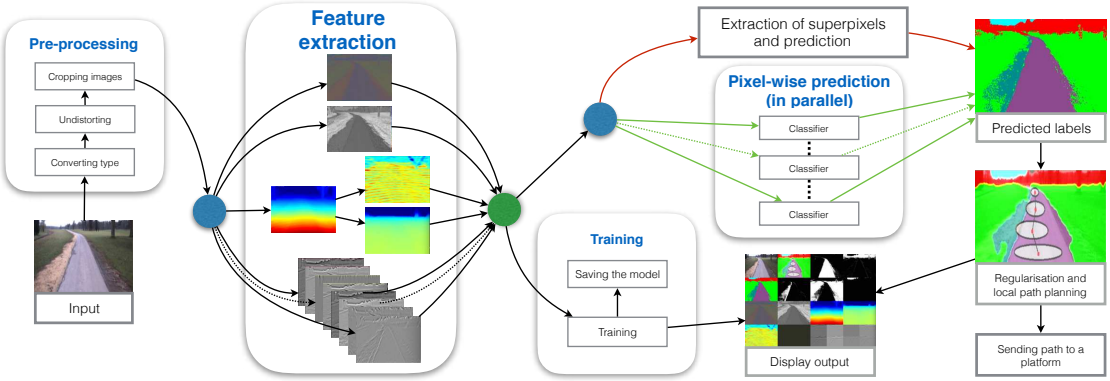


Figure 3.5: Updated scene understanding pipeline for collision-free route following. Note that prediction of class labels can be performed either with pixel-wise (green arrows) or with superpixels (red arrows).

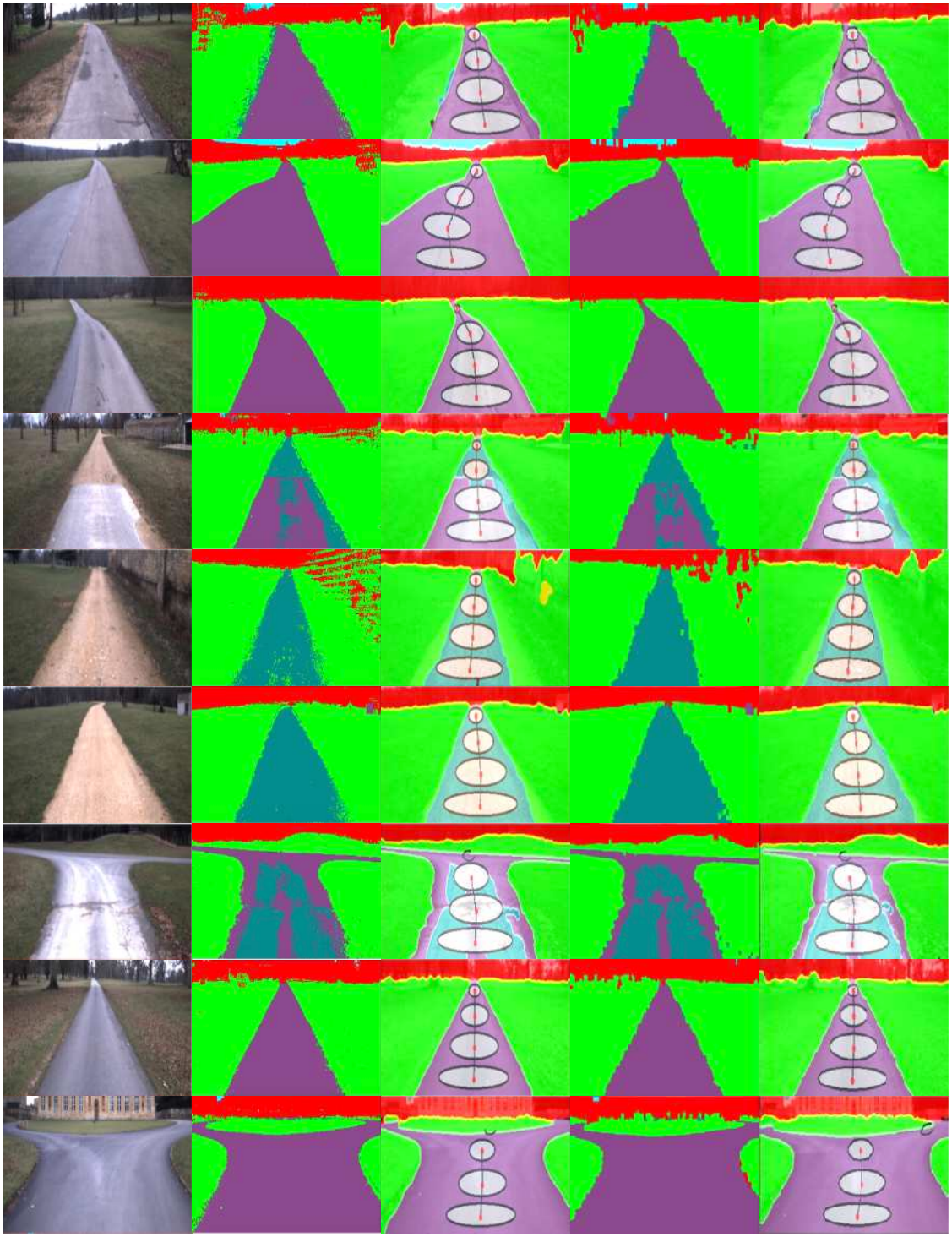


Figure 3.6: Qualitative results of the improved system running on 640 x 380 images. Columns from left to right: RGB inputs, pixel-wise predicted labels, regularised labels of pixel-wise predictions, superpixel predicted labels, and regularised labels of superpixel predictions. Note that 2,000 superpixels are extracted in the these examples.

Figure 3.7: Example in which the RF model fails to produce reasonable segmentation in an unusual environment.

Qualitative results of the improved system are shown in Figure 3.6. Note that when a road bifurcates (e.g., last row of Figure 3.6), at every frame the system chooses which way to drive depending on the centre of mass of the segmented traversable area. For our autonomous outdoor experiment with the Husky UGV platform, we added an extra behaviour to the controller that enabled the robot to rotate and find traversable paths when the robot was facing obstacles (e.g., wall) perpendicularly.

3.10 Discussion and Conclusions

The presented road segmentation approach had some advantages and disadvantages that were crucial for defining the future direction of the research. The results presented showed that the system required small amounts of training data to operate in environments in which it was trained. The system provided a short-term solution for the robot to operate in a small area. However, it did not generalise well with small amounts of training data and it failed when it was tested outside of the trained environments as shown in Figure 3.7, where the RF model was trained with the Cornbury Dataset but tested in a garage with a parked van blocking most of the view. Obtaining a model that generalises well usually requires large amounts of training data and high-capacity models, where deep learning models are more suitable. Additionally, using regularisation increases the overall accuracy by removing noise but it does not provide exact boundaries between road and

non-road areas (as the regularisation smoothes outputs), which is very crucial in the autonomous driving domain (e.g., parking, lateral localisation). Another important aspect for self-driving cars is the ability of a model to capture context of a scene for more accurate segmentation results and for inferring what is behind occluded areas. Roads are shared between vehicles and those very vehicles create occlusions for one another. While driving, humans can easily infer occluded road boundaries, but for a model to infer true boundaries of a road it needs to capture contextual information from a scene. The input feature channels of the RF model provided local information only without capturing the context. To overcome the disadvantages of the presented RF-based approach we required a model that:

- had high capacity to learn variations in shape, appearance, and structure,
- could capture contextual information from inputs,
- could infer what was behind occluding obstacles, and
- could provide visible and occluded road boundaries.

Considering these requirements, we decided to tackle the problem using deep learning approaches, which provided us with high capacity models and tools to design a new network architecture that could capture contextual information. Moreover, we concentrated on road boundaries alone rather than trying to segment the road surface. However, before we could start designing and training deep models, we had one important question to answer: "how to obtain annotated training data easily and within a reasonable amount of time?" In the following chapter we present a solution for this question: a framework to easily obtain annotated training data for hundreds of images within an hour to train deep models for road boundary detection.

4

Road Boundary Data Annotation

Contents

4.1 Introduction

Deep neural networks (DNNs) often require large amounts of training data to achieve high-performance, well-generalised models that can cope with changes due to colour, appearance, illumination, background clutter, perspective, scale, and occlusion. In this thesis, one of our goals was to find true road boundaries as they legally and intentionally delimit drive-able space. Tackling this problem had two major difficulties besides aforementioned ones: (1) road boundaries are often narrow and long with no clear structure and appearance, and (2) other road users (vehicles, pedestrians and cyclists) occlude the road boundaries. Often, detecting only visible road boundaries is not enough in urban scenarios as can be seen from examples in

Figure 4.1: Examples of road boundary occlusions: partial occlusion (first column), full occlusion (second and third columns).

Figure 4.1, where road boundaries are partially or fully occluded. For this reason, we needed to obtain training data that would include a large number of samples to capture the variability of road boundaries and contain annotations for both visible and occluded road boundaries. To this end, we needed a framework to carry out annotations efficiently within a reasonable amount of time.

Fine-grained hand-annotation of road boundaries from images would be a very time-consuming process and it would be impossible to exactly annotate the position of occluded road boundaries. To avoid this, we decided to annotate a 3D point cloud data that was collected by a 2D laser. As shown in Figure 2.2, the laser was attached vertically to the front of the test vehicle. In this sensor configuration, laser light pulses mostly hit the road boundaries perpendicularly, which made the road boundaries easily distinguishable for annotation (Figure 4.2, top left). Moreover, laser light pulses reached road boundaries behind parked vehicles as the pulses passed under the vehicles and hit the road boundaries (Figure 4.2, top right). Note that we can use a 2D laser for data annotation, but we cannot use it for online inference as we need to see and detect at least what is in front of the car if not all around it.

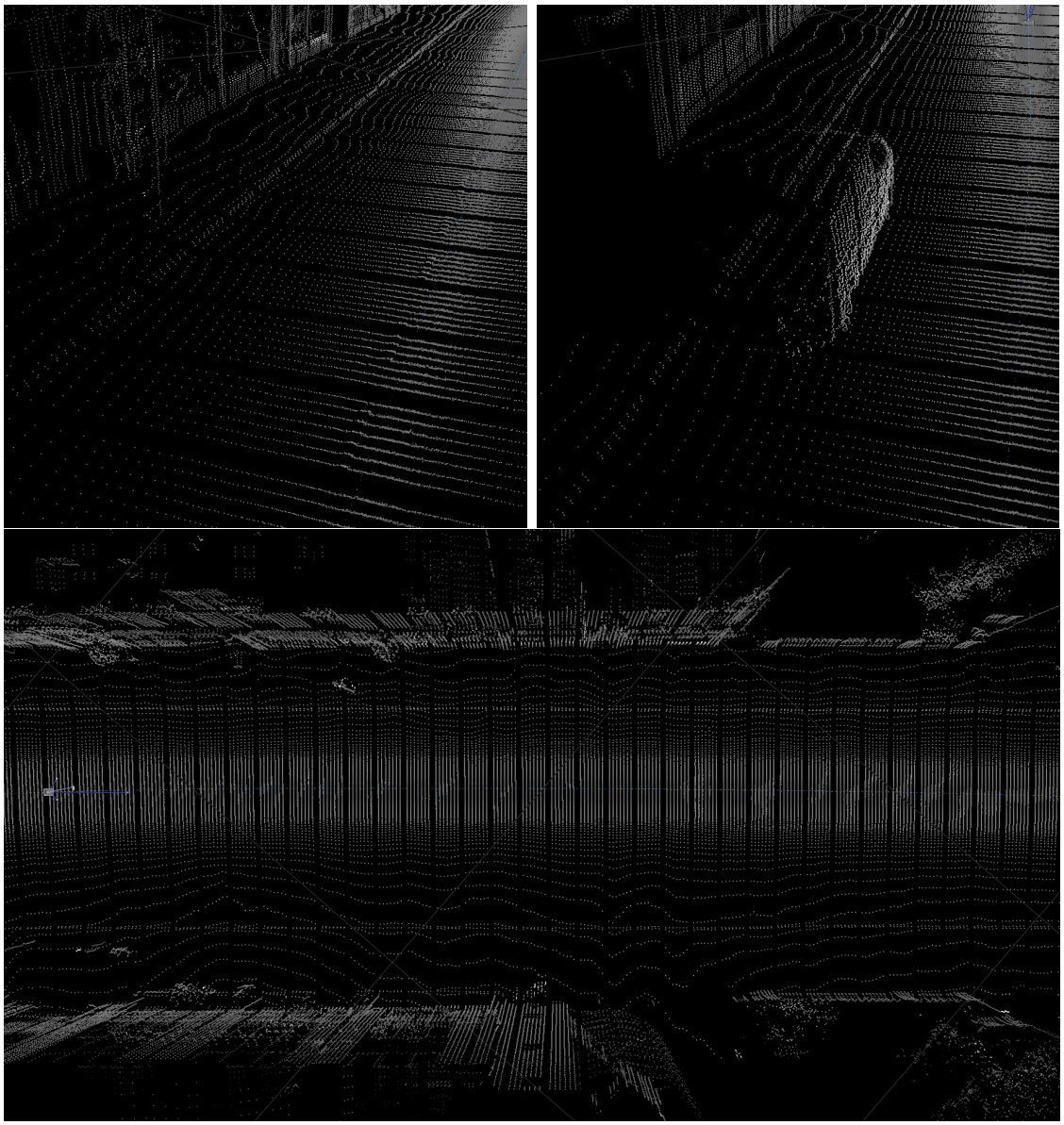


Figure 4.2: 3D point cloud from 2D laser data. The road boundary on the left side of the road is easily distinguishable (top left), laser light pulses reach the road boundary behind the parked car as they pass under the car (top right), bird's-eye view of the point cloud of a street where both left and right boundaries are clearly distinguishable.

4.2 Annotation Tool: Map Builder

To obtain a 3D point cloud of the Oxford RobotCar Dataset from 2D laser, we integrated subsequent vertical laser scans in a coherent coordinate frame as shown in Figure 4.2 (bottom). We used VO to estimate the vehicle's motion and compute transformations between subsequent scans. Note that, as VO uses camera images to calculate the vehicle's ego-motion, it provides transformations between camera timestamps. We use interpolation to obtain transformations between subsequent 2D laser scans at time frames t_l and t'_l as follows:

$$T(t'_l, t_l) = T(t'_l, t'_{vo}) \cdot T(t'_{vo}, t_{vo}) \cdot T(t_{vo}, t_l) \tag{4.1}$$
$$s.t.\ t'_{vo} \leq t'_l, \quad t_{vo} \geq t_l$$

where t'_{vo} and t_{vo} are the closest time steps of the VO with respect to the laser frames and where $T(t'_{vo}, t_{vo})$ is defined as follows:

$$T(t'_{vo}, t_{vo}) = \prod_{i=t'_{vo}}^{t_{vo}-1} T(i, i+1) \tag{4.2}$$

To obtain $T(t'_l, t'_{vo})$ and $T(t_{vo}, t_l)$, we interpolate in $[t'_{vo}, t'_{vo}+1]$ and $[t_{vo}-1, t_{vo}]$ respectively. Having generated 3D point clouds of the datasets, we annotated points corresponding to road boundaries using the Map Builder tool. We assigned the same IDs to the points lying on the same continuous boundary. When projecting the annotated points, this enabled us to connect consecutive points according their IDs and fully annotate road boundary segments between the points, as shown in Figure 4.3.

To train our deep models as presented in the following chapters, we selected and annotated the 2D laser integrated point clouds of three datasets from the Oxford RobotCar Datasets. The first dataset, which was collected in May, 2015 (Oxford RobotCar 26-05-15 dataset), was annotated from the beginning until the end (10 kilometres long), but for the second dataset, which is from April, 2018 (Oxford RobotCar 30-04-18 dataset), approximately 50% (5 kilometres) was annotated.

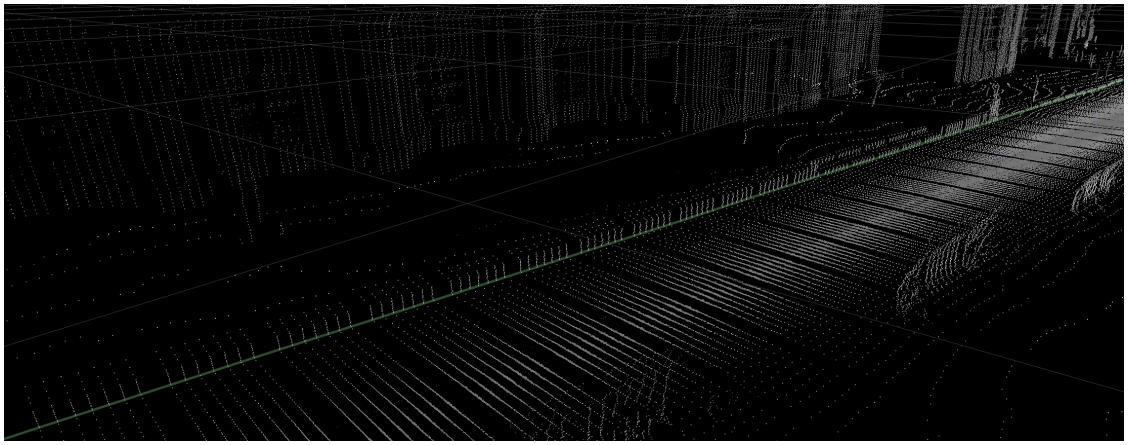


Figure 4.3: Lines are drawn between consecutive points with the same ID to annotate road boundary segments between the points.

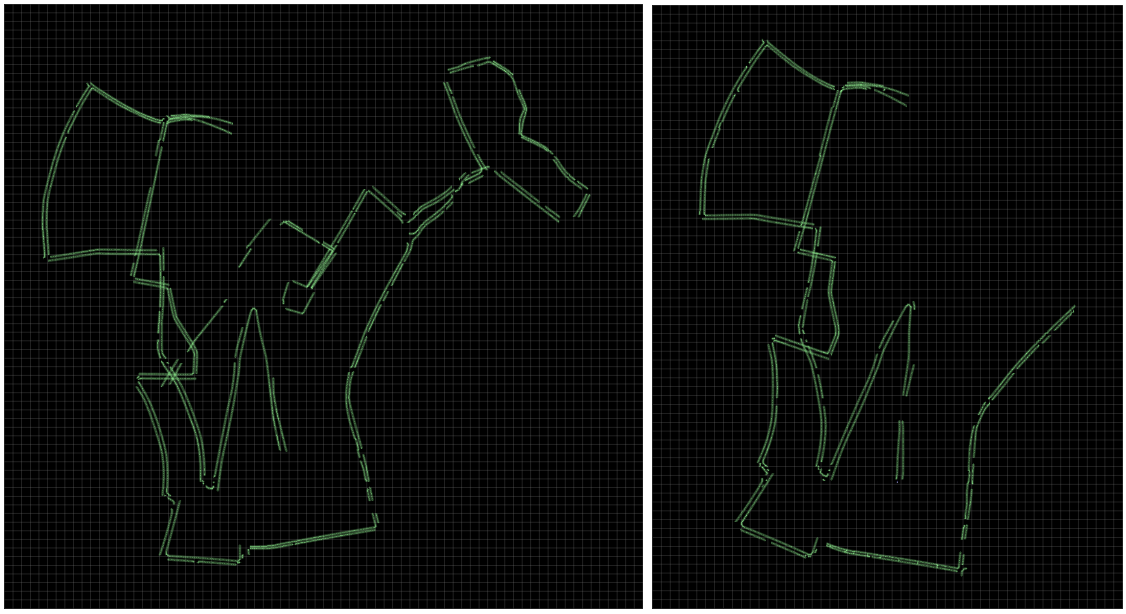


Figure 4.4: Bird's-eye view of the annotations of the datasets. Left: 26-05-15 dataset, 10 kilometres were annotated. Right: 30-04-18 dataset, 5 kilometres were annotated.

Bird's-eye view of the annotations of the datasets are shown in Figure 4.4. One more dataset (Oxford RobotCar 18-01-19) from the Oxford RobotCar Dataset was annotated in the same way for testing the proposed models.

4.3 Training Data Generation Tool

Having annotated the datasets we projected the annotations to generate the desired training data, which involved applying three steps:

- Projecting annotations to generate ground truth for camera images,
- Boosting the number of camera-based training samples by projecting labels from the annotated datasets to other traversals of the same route, and
- Projecting annotations to generate ground truth for LiDAR-based bird's-eye view images.

The overall pipeline of the proposed road boundary training data generation framework, which we describe in the following three subsections, is shown in Figure 4.5.

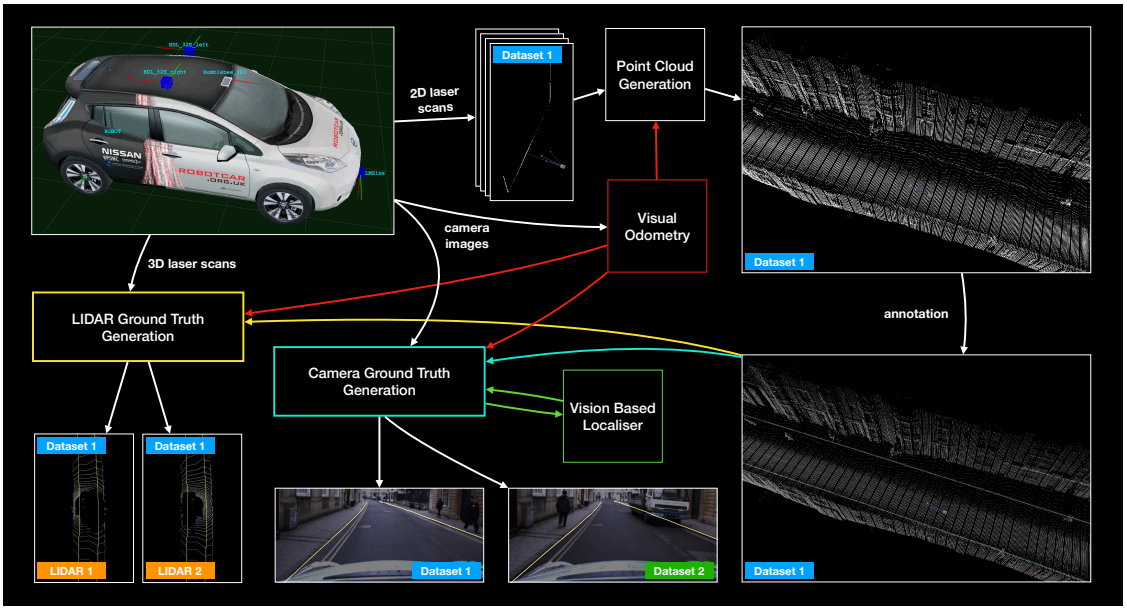


Figure 4.5: Data Annotation Framework: an efficient way of annotating road boundaries to obtain ground truth for camera- and LiDAR-based training samples.

4.3.1 Ground truth generation for camera images

To generate camera-based annotated training data, we project annotations to images of a forward-facing camera. The Map Builder tool outputs a list of annotated points, where each point corresponds to a point from a 2D laser scan. Thus, it has the timestamp of the scan, 3D coordinates in the laser frame, and the ID of the continuous road boundary annotation that it belongs to. Using VO, each point is projected from its timestamp to the timestamp of a given camera image. Then,

the points are transformed from the laser frame to the camera frame and projected into the given image. Lines are drawn between projected points with the same road boundary annotation IDs to annotate road boundary segments between the points. As a result, road boundary ground truth masks for camera images are generated, see Figure 4.6. Note that we call these masks "raw" masks as they contain annotations for both visible and occluded road boundaries, which we split into two classes later (see Chapter 5).

Figure 4.6: Generated "raw" road boundary training data examples: road boundary masks overlaid on top of RGB images. These masks are generated by projecting labels from annotated 3D point clouds into the corresponding images and they contain both visible and occluded road boundaries.

When annotating road boundaries we did not assume that every road boundary was a curb and we annotated all road boundaries even if they were on a flat surface with no height differences between roads and pavements. Although seeing flat road boundaries in the laser pointcloud was harder than curbs, we were simultaneously displaying projections of annotations of road boundaries on camera images in the annotation tool and using the projections as a reference to precisely position the annotations. This makes the proposed road boundary annotation framework applicable to any types of roads (e.g., urban roads, dirt rural roads, highways and motorways). An example of annotated road boundary with no height difference is shown in Figure 4.7.

Figure 4.7: The road boundary on the left hand side of the road has no height difference between the road and pavement and it was precisely annotated in the laser pointcloud by simultaneously displaying its projection on the camera image in the annotation tool and using it as a reference while annotating.

4.3.2 Boosting training samples

Annotating road boundaries in laser point clouds and projecting them into images enables us to easily generate hundreds of ground truth masks within a short period of time. Approximately 750 raw ground truth masks can be generated with one hour annotation. To further increase the number of training samples, we projected labels from the annotated datasets to other traversals using a vision-based localiser. As we mention in Section 2.3, the Oxford RobotCar Dataset contains over 100 repetitions of a 10 kilometre long consistent route, which enables us to manually annotate only one dataset and then obtain over 100 annotated datasets automatically, as shown in Figure 4.8. We obtained 15K annotated images by projecting labels into the images of the annotated dataset (26-05-15) and an additional 9K images by projecting the labels from the annotated dataset to three other datasets from the Oxford RobotCar Dataset (17-03-15, 08-05-15, and 19-05-15). See Table 2.1 for more details of the number of annotated samples for each dataset.

4.3.3 Ground truth generation for LiDAR-based IPMs

In sections 4.3.1 and 4.3.2, we described two steps to generate camera-based training samples. To generate LiDAR-based annotated training data we project labels from the annotated point clouds to 3D LiDAR scans. Then, the labels and 3D LiDAR scans are transformed into 2D bird's-eye view images (IPM) to obtain input images

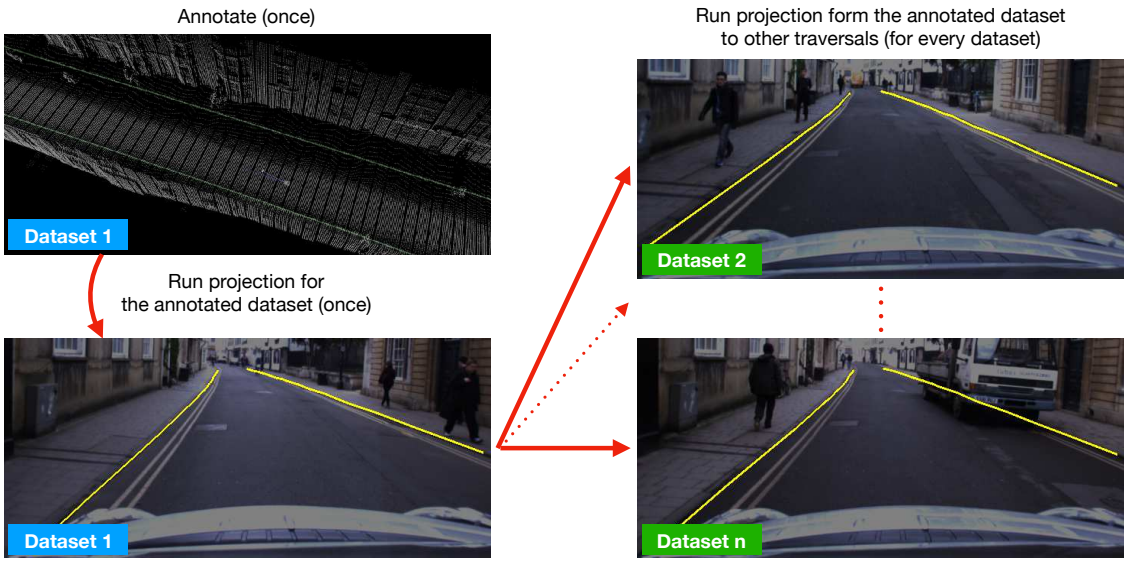


Figure 4.8: Boosting training data: annotate a dataset once, project the annotations to the images of the same dataset, then run the training data generation tool using outputs from the vision-based classifier to automatically project the annotations to other traversals.

and their road boundary masks. Note that the 3D LiDAR scans are trimmed to keep only the points that are close to the road surface before transforming them into IPM images. Our test vehicle has two 3D LiDARs (see Figure 2.2) and we generate IPM images for both of them. Note that, before projecting the scans to the IPMs, we transform the scans of both LiDARs to the "ROBOT" frame, which has its origin at the rear axle of the test vehicle (as shown in Figure 2.2), in order for them to have the same reference frame. We obtained 17K training samples from the 30-04-18 dataset, only half of which was annotated, and 2K samples from the 18-01-19 dataset that was used for testing only.

These ground truth generation steps take their annotations from the integrated point clouds of 2D laser scans of the 26-05-15, 18-01-19, 28-07-16, and 30-04-18 datasets. In addition to these four datasets, we annotated a fifth one (24-08-18) that did not have 2D laser scans. This was collected with another test vehicle that had only one 3D LiDAR and no 2D lasers. Similarly to the above framework, we generated integrated point clouds to annotate road boundaries, but this time from 3D LiDAR scans. Then, we generated IPM images for each scan and projected annotated road boundaries to obtain their ground truth masks. Examples of

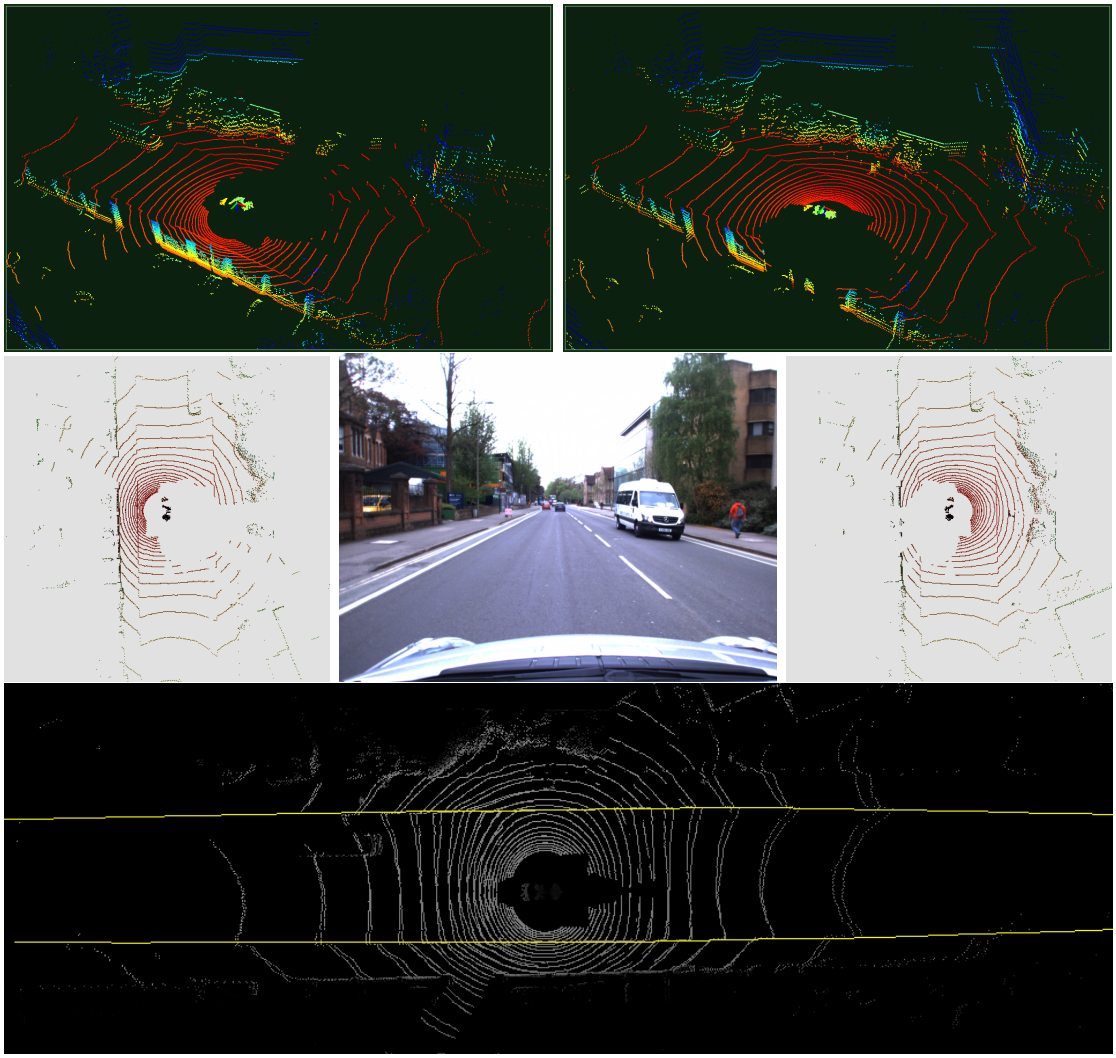


Figure 4.9: LiDAR training data. Left LiDAR scan's point cloud (left top), right LiDAR scan's point cloud (right top), IPM of the left LiDAR scan (middle row, left), IPM of the right LiDAR scan (middle row, right), annotated road boundaries overlaid on top of a combination of left and right IPMs (bottom).

generated training samples, which were generated by integrating five consecutive laser scans before being transformed into IPMs, are shown in the Figure 4.10. 1,800 IPM images that were generated using this dataset were only used to train and test models presented in our paper [13] that we present in Section 6.5.

4.4 Conclusions

In this chapter we presented the framework for easy generation of a large number of camera- and LiDAR-based training masks for road boundaries within a reasonable

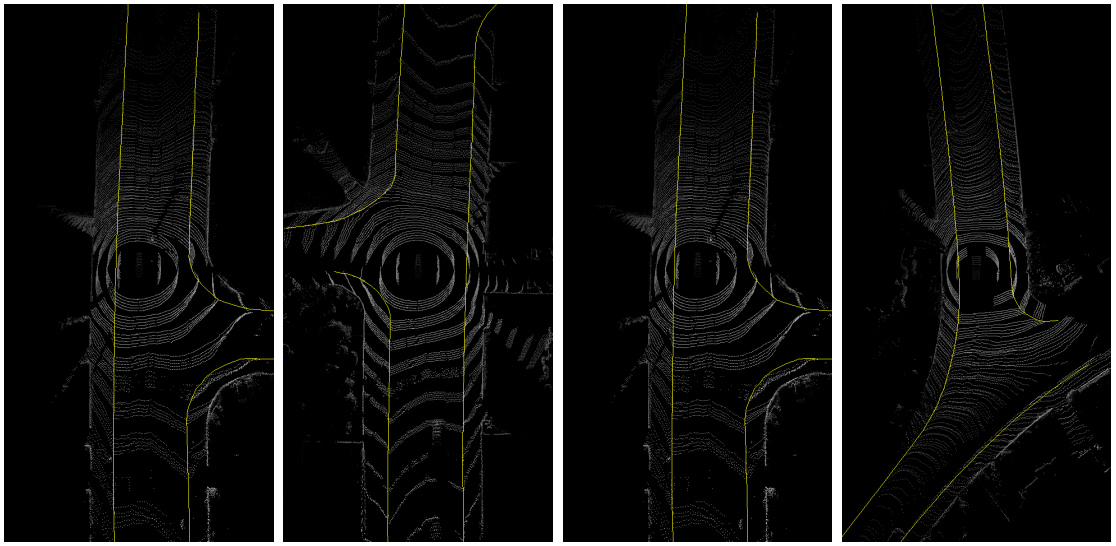


Figure 4.10: Generated "raw" LiDAR-based road boundary training data examples: road boundary masks overlaid on top of IPM images. These masks contain both visible and occluded road boundaries.

amount of time. The framework enables us to avoid fine-grained hand annotation of images and provides "raw" road boundary masks for both visible and occluded boundaries. In the following chapters we will talk about how we partition "raw" road boundary masks into two classes of visible and occluded boundaries. Being able to easily generate thousands of annotated training data for both camera- and LiDAR-based images enables us to apply deep learning techniques for road boundary detection.

5

Inferring Road Boundaries Through and Despite Traffic

Contents

Figure 5.1: Road boundaries appear in different shapes, colours and structures, which makes road boundary detection a challenging task.

5.1 Introduction

Starting from this chapter we present deep learning based methods for road boundary detection and their applications in the domain of autonomous driving. In Chapter 3 we presented machine learning based scene segmentation method with a variational approach, in which we did not apply deep learning techniques in order to avoid the need for time consuming and fine grained annotation of hundreds of training images. Instead, we used a Random Forest to segment collision-free, traversable paths with small amounts of training data. However, the road boundary annotation framework that we presented in Chapter 4 gave us an opportunity to exploit the

capacity of deep learning approaches by easily generating thousands of training samples for the challenging road boundary detection problem. Road boundaries appear in different shapes, structures, and colours and the small width and long thin shape of road boundaries (Figure 5.1) make their detection challenging. Presence of occluding obstacles aggravates this problem further.

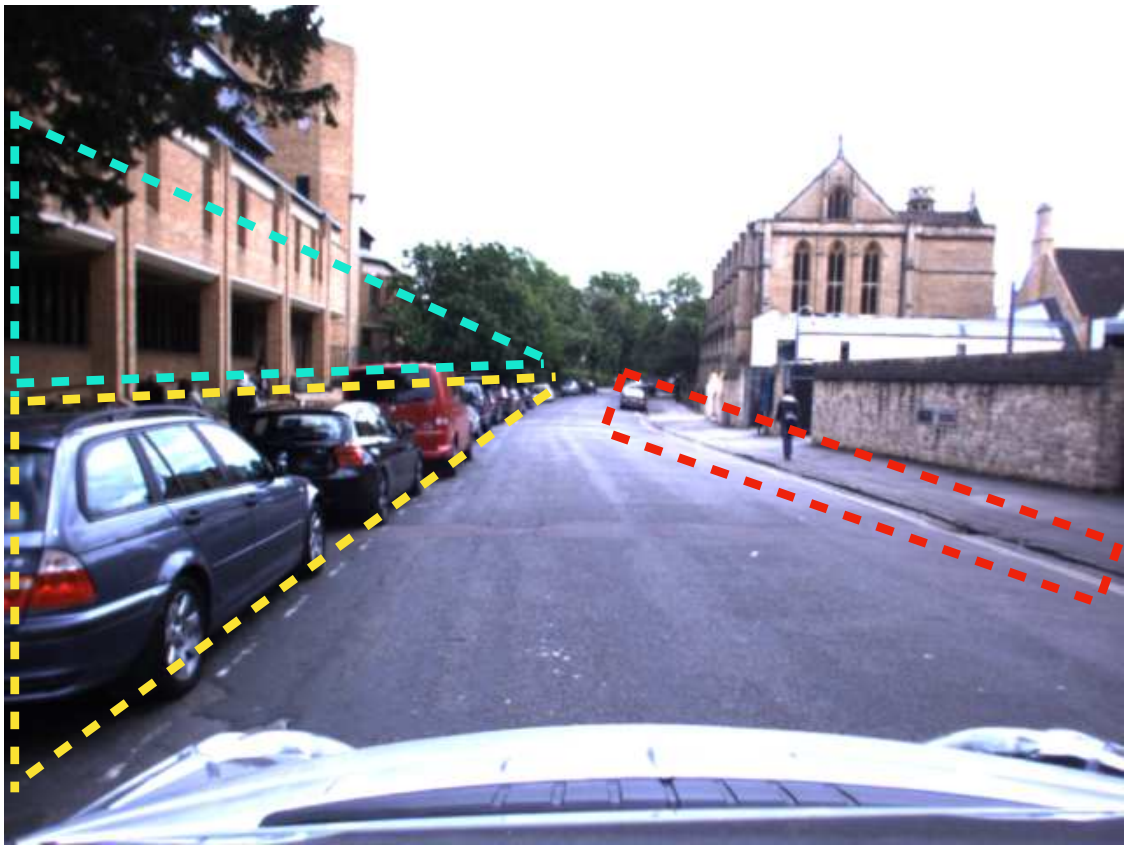


Figure 5.2: Parked cars on one side of the road, buildings visible behind them, and a road boundary on the other side provide clues for the occluded road boundary in the scene.

As we mentioned in Section 4.1 detecting only visible road boundaries is often not sufficient in urban scenarios as other road users create partial or full occlusion of road boundaries (see Figure 4.1). There is a need to estimate the correct orientation and position of occluded road boundaries in crowded areas, which is a challenging task and requires a model with a large receptive field to capture contextual information. A road boundary on one side of a road can be a clue about the road boundary on the other side of the road, or buildings, parked cars, a glimpse of a visible road

boundary in between parked cars. For example, in Figure 5.2, there are parked cars on one side of the road, buildings visible behind them, and a road boundary on the other side. All of these clues allow us to infer the occluded road boundary on the left side of the road. To capture all that contextual information, we need to relate pixels from the entire image plane, and need to pass information from one side the image to another in all directions. Such a problem is well suited to CNN-based approaches, but conventional NN architectures fail to infer the exact location of an occluded, narrow, continuous curve running through the image. To tackle the challenge of inferring occluded road boundaries in crowded urban environments, we presented a new approach in our paper [14] (Section 5.5) with a novel deep model architecture. Our proposed approach was designed with following points in mind:

(a) Detecting visible road boundaries and inferring occluded ones,

(b) Having outputs separately in two classes to know which parts are detected and which are inferred,

(c) Relating visible and occluded road boundaries with a continuity constraint,

(d) Capturing contextual information,

(e) Forcing outputs to have thin long shapes for occluded boundaries, and

(f) Having outputs in multiple scales.

We already mentioned that (a) detecting visible and inferring occluded road boundaries is crucial in urban scenarios as other road users create occlusions and many scenes present 100% occlusion. Splitting the road boundary detection problem into two tasks and (b) having outputs in two classes has a safety perspective as it is good to know which boundaries are directly observed and which ones are inferred. Having a continuity constraint to (c) relate visible and occluded road boundaries provides clues for inferring occluded road boundaries. To obtain even more clues, the model needs to be able to (d) capture contextual information by passing information from one side of an input image to another in all directions.

Road boundaries have a long thin shape with sinuous structure and to ensure that the model predicts reasonable outputs we need to (e) "force" the model to infer the outputs with the same shape and structure for hidden road boundaries. And finally, (f) having the outputs in multiple scales is necessary to cope with occluding obstacles in different sizes and shapes, ranging from traffic cones to trucks and busses. Taking these requirements into account, we designed the road boundary detection models as shown in Figure 5.3, which we explain below in detail.

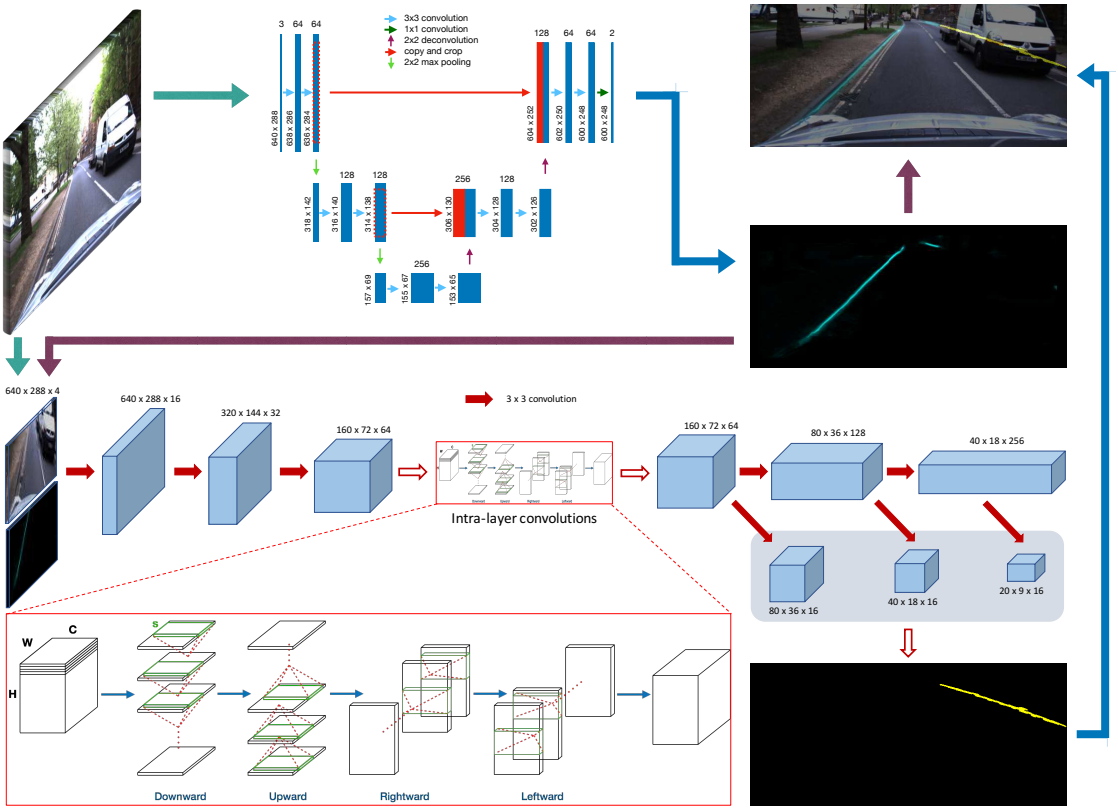


Figure 5.3: Camera-based road boundary detection and inference pipeline. Given an RGB input image, first, visible road boundaries are detected with a fully convolutional network. Then, the output mask of the detected visible road boundaries are passed to the second network to infer occluded road boundaries. The second network contains convolutional base layers, intra-layer convolutions, and multi-scale prediction layers that infer occluded road boundaries in a hybrid, discrete-continuous form. In contrast to other approaches [15–21], we do not assume that road boundary planes are orthogonal to the road plane.

5.2 Partitioning Training Data

Obtaining outputs in two classes for safety/operational and algorithmic advantages requires partitioning the raw training road boundary masks into two classes: visible and occluded. To separate the data into two classes we trained the U-net architecture [22], which is a fully convolutional network, with raw masks. The network can localise detected objects precisely by concatenating features from convolutional layers with outputs from deconvolutional layers. Higher resolution input-side features provide information to up-sampled outputs for better localisation of the detected objects. In this way, the network can detect visible road boundaries, but fails to infer correct structure and position of occluded road boundaries for two reasons: (1) the network does not have a large enough receptive field to capture contextual information in the scene and to estimate position and structure of occluded road boundaries and (2) the network does not have any structure to force/bias outputs to be in a thin long shaped form for the occluded road boundaries. As a result, the network can detect visible road boundaries when trained with raw masks, but fails to infer occluded ones and outputs blurry masks over occluding obstacles as shown in Figure 5.4. This enabled us to automatically partition the raw training data into two classes, see Figure 5.5 for examples of partitioned training data masks.

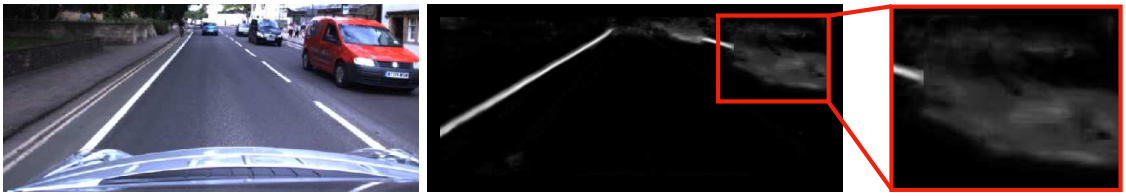


Figure 5.4: The U-net architecture can detect visible road boundaries when trained with raw masks, but fails to infer occluded road boundaries and outputs blurry masks over the occluding obstacles.

To detect visible road boundaries, we decided to leverage the U-net architecture as it provided reasonable outputs for visible road boundaries even when trained with raw masks. Having partitioned the training data, we re-trained the U-net architecture with visible road boundary masks only. With 24K samples of training data that we obtained from four Oxford RobotCar datasets using our annotation

Figure 5.5: Partitioned training data examples with two classes: visible (green) and occluded (red).

framework (Chapter 4) we trained the U-net architecture to detect visible road boundaries. Note, moving forward we name this model Visible Road Boundary Detection (VRBD) model.

5.3 Occluded Road Boundary Inference Model

As discussed above, visible road boundaries provide clues about positions of occluded ones. To take advantage of that we designed our approach with the continuity constraint between sub-tasks of detecting visible road boundaries and inferring occluded ones. As shown in Figure 5.3, the output of detected visible road boundary masks from the VRBD network together with the input RGB image becomes the input to the second network: the Occluded Road Boundary Inference (ORBI) model. The model was designed with requirements (d), (e) and (f) in mind. The network consists of convolutional layers and can be divided into three parts: "base" convolutional layers, intra-layer convolutions [23] and parameterised multi-scale

predictions. The first part has three base layers that process inputs and then is followed by intra-layer convolutions. Traditional convolutions are applied between feature maps, but intra-layer convolutions are applied slice-by-slice within feature maps. They are applied in four directions (top to bottom, bottom to top, right to left, and left to right) that propagates information across rows and columns in all directions. As a result, the ORBI model can capture contextual information in the scene by having spatial relationships across rows and columns.

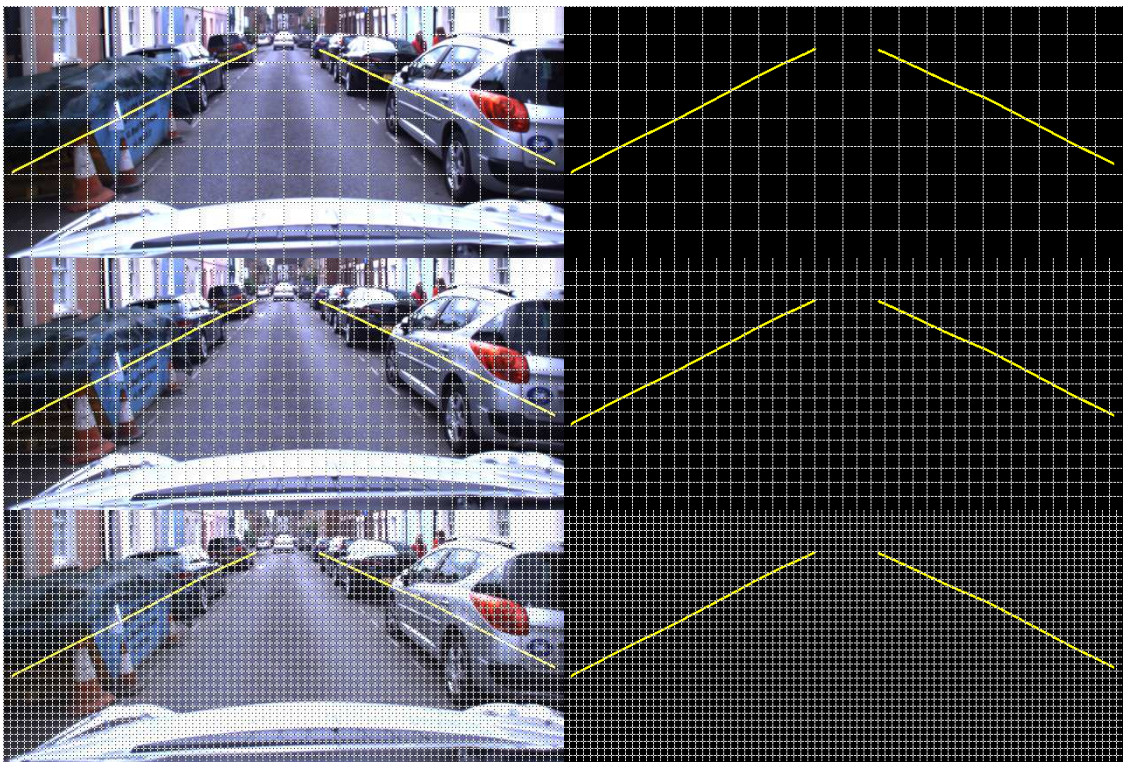


Figure 5.6: Multi-scale parameterisation of road boundaries. Pixel-wise masks are divided into a grid of squares at each scale. Each cell of the grid at each scale is parameterised in a discrete-continuous form.

The final part of the network consists of convolutional layers to estimate multi-scale predictions in a discrete-continuous form. Having predictions in multiple scales is important for the model due to different shapes and sizes of occluding obstacles. To bias the network towards estimating thin, long-shaped occluded road boundaries we approached the problem as an object detection problem (instead of segmentation) where the model estimates parameter-wise outputs instead of pixel-wise outputs. To parameterise road boundaries, pixel-wise masks are divided

into a grid of squares at each scale, as shown in Figure 5.6, where grid sizes are 32 x 32, 16 x 16 and 8 x 8 pixels. For each cell of the grid at each scale lines are fitted and then assigned to one of four anchor lines categories (Figure 5.7). Then, two offsets from the fitted lines to the anchor lines are calculated: (1) distance offset from the fitted line to the centre of the cell ($\beta^k_{i,j,gt}$) and (2) angle offset between anchor lines and fitted lines ($\omega^k_{i,j,gt}$), where k, i, j and gt are category number, row number, column number and ground truth, respectively. For example, $\beta^2_{3,5,gt}$ is the distance offset in the second category for the ground truth road boundary corresponding to the cell at the third row and the fifth column.

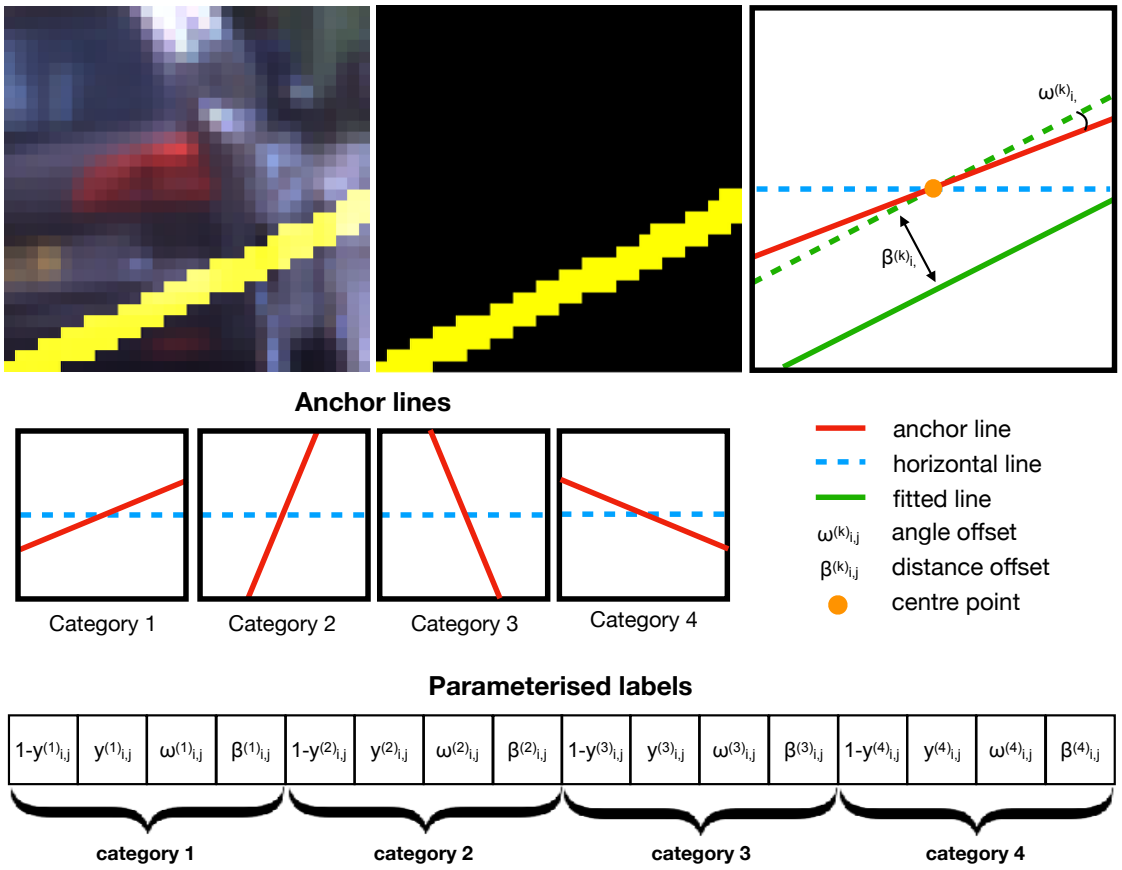


Figure 5.7: Parameterisation of pixel-wise road boundary masks in a discrete-continuous form (similar to [24]). For each cell of the grid at each scale lines are fitted and then assigned to one of four anchor lines categories. Then, offsets (distance offset $\beta^k_{i,j,gt}$ and angle offset $\omega^k_{i,j,gt}$, where k, i, j and gt are category number, row number, column number and ground truth, respectively) from the anchor lines to the fitted lines are calculated. $y^k_{i,j}$ in the parameterised labels indicates the presence of a road boundary for the k^{th} category in i^{th} row and j^{th} column.

As a result, each cell is represented with 16 parameters (4 parameters for each anchor line category) in a discrete-continuous form. The parameters represent the category of lines in a discrete form and offsets from the anchor lines in a continuous form. To infer occluded road boundaries the model needs to learn to correctly estimate these parameters, which means learning to perform classification of categories and regression of offsets at the same time. We achieve this by applying a discrete-continuous loss to our model during the training process, where the total loss is defined as the sum of discrete and continuous losses. The discrete loss is a cross-entropy loss and the continuous loss is a smooth L1 loss [25] between the predicted line and the ground truth line parameters. The smooth L1 loss combines the advantages of L1 and L2 losses. It is more robust to outliers than L2 loss and gradient decreases as it gets closer to zero to make it more precise. See our paper in Section 5.5 for the equations of the losses and for more details.

5.4 Geometric representation of road boundaries

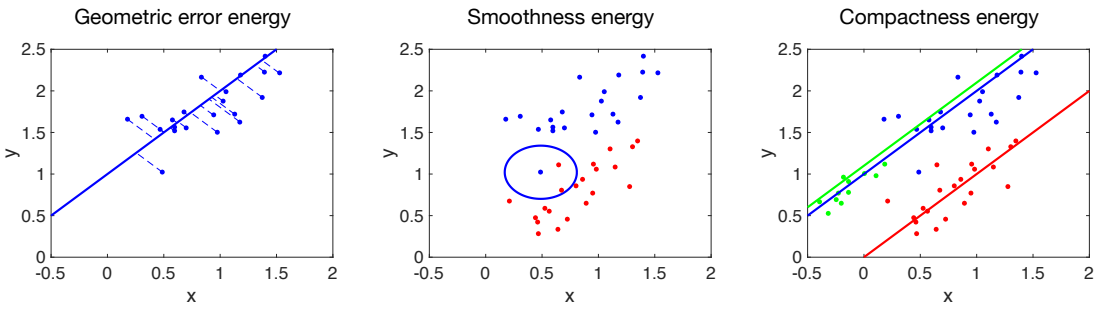


Figure 5.8: The global energy function has three terms: a data term (left), a smoothness term (middle) and a compactness term (right).

Once we have obtained the outputs from the networks, we need to transform the pixel-wise outputs into a set of semantically and geometrically meaningful road boundaries. We formulate a global energy function that jointly optimises the overall assignment of points to models while seeking a smooth and compact solution using Convex Relaxation Algorithm (CORAL) [26]. This allows us to obtain a continuous representation of road boundaries. The global energy function

has three terms: a data term, smoothness term and compactness term. The data term accounts for the distance between a point and a curve model. The smoothness term promotes a homogeneous assignment of labels to neighbouring points. It penalises points that belong to the same neighbourhood but do not share the same model. The compactness term penalises the number of models by adding a constant cost per model. This eliminates redundancies in models resulting in a compact solution. Figure 5.8 visualises all three terms.

The following paper describes the proposed approach in more detail and provides experimental results. The limits of the system's performance were further investigated with experiments and results in the discussion which concludes this chapter, which also guides the narrative of the thesis in subsequent chapters.

Inferring Road Boundaries Through and Despite Traffic

Tarlan Suleymanov Paul Amayo Paul Newman

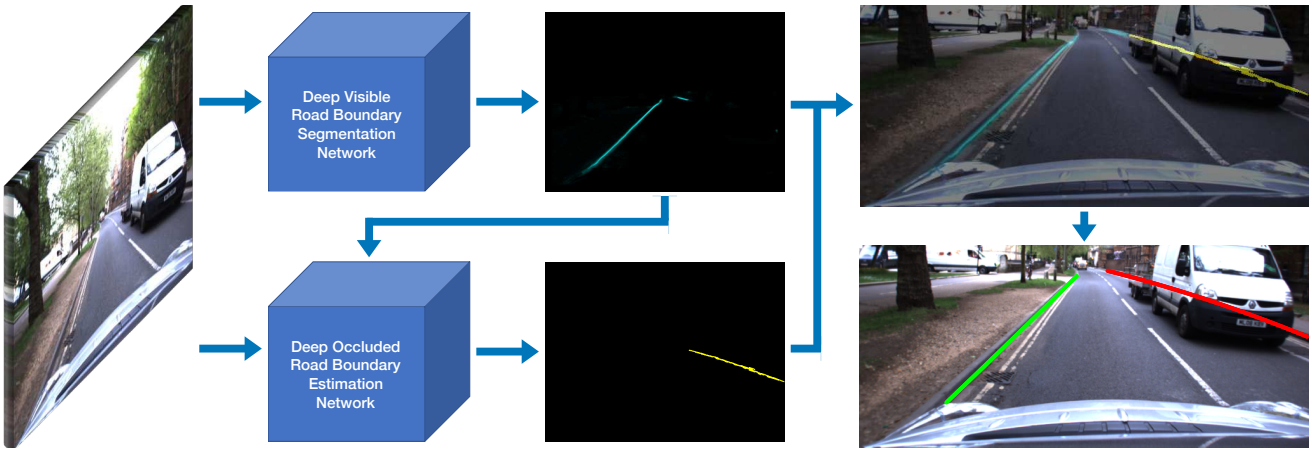


Fig. 1: Given an input image, road boundaries are inferred irrespective of whether or not the boundaries are actually visible. Our coupled approach first segments visible road boundaries with a fully convolutional network and then passes output to our deep network to infer occluded road boundaries. Our network contains intra-layer convolutions and produces outputs in a hybrid discrete-continuous form.

***Abstract*— This paper is about the detection and inference of road boundaries from mono-images. Our goal is to trace out, in an image, the projection of road boundaries irrespective of whether or not the boundary is actually visible. Large scale occlusion by vehicles prohibits direct approaches - many scenes present 100% occlusion and so we must infer the boundary location using scene context. Such a problem is well suited to CNN derived approaches but the sinuous structure of a hidden narrow continuous curve running through the image presents challenges for conventional NN-architectures. We approach this as a coupled, two class detection problem -solving for occluded and non-occluded curve partitions with a continuity constraint. Our network output is in a hybrid discrete-continuous form which we interpret as measurements of segments of the true road boundary. These measurements are passed to a model selection stage which associates measurements to minimal number of *a-priori* unknown set of geometric primitives (cubic curves) representing road boundaries. We present a semi-supervised method which leverages a visual localisation to generate 25 thousand labelled images for training and testing - the results of which are presented in the conclusion of the paper.**

I. Introduction

In the context of autonomous driving, curbs (road boundaries) play an important role as they delimit, legally and intentionally, drive-able space. They provide information for mapping, path planning and navigation, and can be used as reference structure for accurate lateral vehicle positioning on a road. Curb detection is a crucial component of ADAS (Advanced Driving Assistance Systems) such as parking assist systems. Knowing where the road ends is always good. However the purpose of roads is to carry vehicles and those very vehicles occlude the road boundaries. Our goal here is to infer road boundaries despite the occlusion. Our motivating observation is the orientation and location of occluding objects (overwhelmingly vehicles) is an observation of a hidden state namely the road boundary. Although our own camera cannot see the road boundary explicitly, we assume that an observer in the occluding vehicle does and is driving and positioning the vehicle accordingly. Note, moving forward we will use "road boundary" and the shorter "curb" synonymously.

In recent years, machine learning has achieved state-of-the-art performance in segmentation and object detection problems. However, many existing image segmentation or object detection methods do not explicitly infer geometry of road boundaries, rather segment the road [1]. Curb detection using deep learning approaches from mono images hasn't been addressed deeply in the literature. The small width and elongated shape of curbs make curb detection challenging for state-of-the-art deep models. Presence of occluding obstacles makes this even more challenging. We propose a deep learning based approach that relies on a single camera image and capable of detecting visible curbs and estimating positions of occluded curbs behind other road users (cars, cyclists, pedestrians). To train our deep models we propose a framework that enables swift accumulation of ground truth masks of visible and occluded target. We make

Authors are from the Oxford Robotics Institute, Dept. Engineering Science, University of Oxford, UK. {tarlan,pamayo pnewman}@robots.ox.ac.uk

no assumptions about structure, shape or colour of curbs or occluding obstacles.

The main contributions of this paper are as follows:

- An image annotation framework to easily generate curb masks for hundreds of images within an hour.
- A way to detect curbs without making any assumptions about their 3D structure, shape or appearance.
- A new deep model architecture based on convolutional layers to estimate curbs that are occluded by other road users.
- Inferring occluded road boundaries using a single image without temporal information.
- Performing a model selection step to return cubic representation of road boundaries without placing assumption on the number of continuous curbs within the scene.

This paper is structured as follows. Section II provides an overview of curb/road boundary detection methods using different sensors. We explain our framework for generating ground truth data to train our models in Section III. A detailed description of our model is provided in Section IV, which is followed by road boundary classification and evaluation of our approach through various experiments. A summary of our contributions is given in Section VII.

II. Related Work

Curb detection using single frame images is a hard problem, curbs are narrow, long and have no clear form in either appearance or geometry. If one assumes curbs always have a change in height (we don't) then the 3D structure can be modelled as a 2D step function in the image. This assumption is used in many curb detection approaches that rely on laser sensors or stereo cameras, where the 3D information is extracted to detect curbs.

a) Camera-based methods: In [2], a camera-based approach is used which exploits appearance, temporal information and 3D geometry to detect curbs with the underlying assumptions that the road surface is flat and curb planes are orthogonal to the road plane. A support vector machine is used for patch classification using histogram of gradients as image descriptor and the classifier is evaluated to detect curbs up to 2 metres away. 3D point clouds and intensity images are obtained from stereo cameras and used for curb detection in [3]. Curbs are detected independently of their orientation and geometry in relation to the car. The intensity images are used for correction of extracted curbs from 3D point cloud. In [4], normal vector information extracted from stereo images is used to determine boundary areas. Three Bayes models are established based on the surface normal vector, height and colour cues. A Naive Bayes framework, using these cues, provides a confidence level for each point in the boundary area. A Digital Elevation Map (DEM) is built in [5] from stereo images to detect curbs and height variation is used to detect edges. A multi-frame persistence map is used to reduce 3D-noise by performing temporal filtering and selecting only persistent points. Straight and curved curbs are extracted via a Hough accumulator. A curb detection algorithm based on a DEM is presented in [6], where different mapping techniques are compared. Parameters of a 3D curb model are estimated based on 3D point cloud obtained from dense stereo vision in [7]. 3D points are assigned to the curb surfaces using a temporally integrated Conditional Random Field (CRF) and then parameters of curb and road surface are estimated. The method can reconstruct only some part of partial occluded curbs. A multi-cue image-based curb classifier using Local Receptive Field (LRF) features is presented in [8]. Here cues from intensity images and three dimensional height profile data are used for curb classification.

b) LIDAR-based methods: A 3D LIDAR that provides dense point cloud data is used in [9]. Ring compression analysis followed by false positive filters are applied to detect curb points on input data. Then curb models are estimated using Least Trimmed Squares (LTS) that estimates road shape on occluded curbs. However, the presented occluded curb estimation examples are from simple scenarios, where there are curbs on both sides of the road, and this method would likely fail in more complex scenarios, such as junctions or roads with fully occluded curbs. Range and intensity information from 3D LIDAR is used in [10] and visible curbs are detected using elevation data, which again fails in the presence of occluding obstacles. Similarly, a LIDAR-based method presented in [11] detects visible curbs using sliding-beam segmentation followed by segment-specific curb detection, but fails to detect curbs behind obstacles.

In this work we opt for a machine-learning approach to curb detection. Unlike many works reported in the literature, in this work visible and occluded curbs are detected from a single monocular camera frame. We do this without making any assumptions on the structure or shape of the curbs. The large amounts of samples needed to train this network are swiftly accumulated using the module described in the following section.

III. Obtaining Ground Truth and Training Data

A. 3D Annotation

Obtaining well generalised, high-performance deep networks often requires large amounts of training samples. To cope with the variability of curbs, the required data should equally incorporate great variability changes in environment due to scale, appearance, colour and background clutter, occlusion, perspective and illumination. Fine-grained annotation of data requires time-consuming human interaction where labels of different classes must be assigned to outlined distinct regions. To avoid time-consuming image by image hand labelling process, we annotated points corresponding to curbs in a 3D point cloud data that was collected by a 2D laser attached vertically to the rear of a test car. During the annotation, points lying on the same continuous curb are given the same ID. The annotated points are projected to images that are collected using forward facing camera of the car. Between consecutive points with the same IDs, lines are drawn to annotate curb regions in-between the points. While projecting the points to the images, we apply distance and time constraints (e.g. project points that are within 100 metres of the car) to obtain reasonable annotations. This method enables us to easily obtain hundreds of images within an hour (approximately 750 images).

Fig. 2: An annotated 3D point cloud (top) used for generating the training data. Points lying on the same continuous road boundary are given the same ID and lines are drawn between consecutive points to annotate road boundary regions in-between the points. The raw road boundary mask (bottom) is generated by projecting 3D annotations into the corresponding image. The mask contains both visible and occluded road boundaries.

a) Leveraging hi-fidelity localisation: Additionally, we obtain labels for the curbs that are occluded by other road users by leveraging multiple passes through the same scene. As above, we annotated one of the 10 kilometres long datasets from the OxfordRobotcar Dataset introduced by [12] and generated several thousand images with semi-annotated curb masks. To boost the number of training samples, we used a vision based localiser [13] to project labels from the annotated dataset to other traversals at different times of data and weather conditions. As a result, we obtained 25K labelled images. Our data contains images from a diverse set of scenarios such as straight roads, parked cars, junctions and etc. (Figure 3).

B. Partitioning Training Data

We split our task into two sub-tasks: detecting visible curbs and hallucinating occluded ones. Beyond an algorithmic advantage discussed later, this has an operational/safety perspective - it's good to know when a solution is directly observed as opposed to hallucinated/inferred. Our raw curb masks, which are generated by projecting 3D annotations into images (see above), contain both visible and occluded curbs as a single class. To separate our training data into two classes, we trained U-net architecture [14] with the raw masks. U-net is a fully convolutional network that we used here to detect and precisely localise *visible* curbs. The network concatenates higher resolution "input-side" features from convolution layers with up-sampled outputs form deconvolution layers as illustrated in Figure 4. This typically enables the network to localise detected objects more precisely. Although U-net can segment visible curbs on an image, it is not able to estimate correct position and structure of occluded curbs (reasons are explained later). As a result, the U-net trained with the raw labels generates blurry outputs over occluding obstacles, which enables us to obtain masks for visible curbs as illustrated in Figure 5.

IV. Our approach

Having visible and occluded curbs as two separate classes enables sees us tackle the curb detection problem in two steps. But the steps are coupled; visible curbs provide clues about occluded ones. A glimpse of a curb in-between parked cars is a clue about occluded curbs behind the cars. Likewise and as an example of global context curb in one side of a road is a clue about the location and geometry of the partner curb on opposite side.

A. Detecting visible curbs

To detect visible curbs, we straightforwardly leverage the U-net architecture which yielded reasonable performance of detecting visible curbs even when we trained it with both visible and occluded curb labels for training data partitioning (see above). Of course post partition we re-trained with visible boundaries only.

B. Hallucinating occluded curbs

Although U-net can segment visible curbs on an image, it is not able to estimate correct position and structure of occluded curbs, because (1) the network has small receptive field, which is not big enough to capture context around large obstacles and to estimate position of curbs behind them, and (2) the network doesn't have any structures to bias it towards detecting thin space curves across an image. As a result, when the U-net is trained to segment occluded curbs, it produces blurry outputs over occluding obstacles, even if masks of segmented visible curbs are given as an input to the network. To tackle this problem, we approach it as an object detection problem with parameter-wise outputs instead of segmentation problem with pixel-wise outputs. Similar to [15], our proposed model consists of convolutional layers that produce output of detected curbs as discrete lines in multiple scales as illustrated in Figure 6. The network estimates parameters of lines that correspond to each cell of the grid at each scale. The network discretises the output space of lines into a set of default (anchor) lines over different orientation angles. At inference time, the network generates probabilities for the presence of occluded curbs for each anchor line orientation and estimates adjustments to the lines to estimate orientation of the curbs more precisely.

Having predictions of occluded curbs at multiple scales is important due to the different sizes and shapes of occluding obstacles. After experimenting, taking into account running time and accuracy of the model, we settled one 3 scales of parameterised outputs (Fig 6). To convert pixel-wise curb labels to parameterised labels, we divided curb masks into grid of squares in each scale and fitted lines for each cell as illustrated in Figure 7. The lines are parameterised in discrete-continuous form: fitted lines are assigned to one of 4 anchor line categories and then offsets from the anchor lines to the fitted lines are calculated. The anchor lines pass

Fig. 3: "Raw" road boundary mask examples from our dataset overlaid on top of RGB images. The dataset includes masks of semi-annotated *visible* and *occluded* road boundaries from various scenarios.

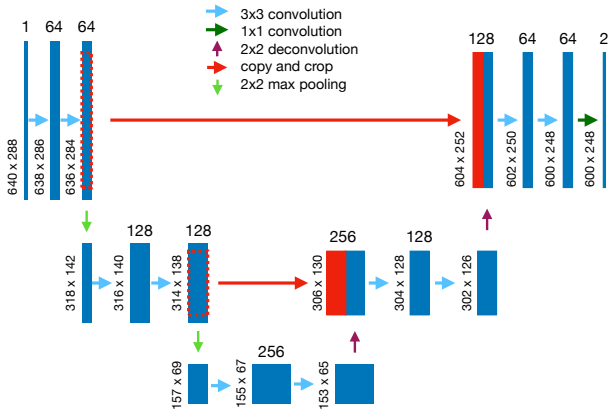


Fig. 4: 3 layers deep fully convolutional U-net architecture. To localise detected objects more precisely, the network concatenates higher resolution "input-side" features from convolution layers with up-sampled outputs form deconvolution layers as illustrated with red arrows.

from the centre point of cells and form an angle under 22.5, 67.5, 112.5 or 157.5 degrees with an imaginary horizontal line (see Figure 8). Lines are assigned to the category of the closest anchor line (e.g. lines with angles between 0 and 45 degrees are assigned to the category 1). Once the fitted line is discretised, two continuous parameters are calculated: (1) angle offset between fitted and anchor lines ($\omega^k_{i,j,gt}$), and (2) distance from the centre point of the cell to the fitted line ($\beta^k_{i,j,gt}$). As a result, we obtain 16 numbers for each cell, 4 numbers for each line category.

The model has 3 layers at the end of the network that progressively decrease in size and allow multi-scale predictions. The output scales have 80 x 36, 40 x 18 and 20 x 9 grids, where each cells on the grids corresponds to 8 x 8, 16 x 16 and 32 x 32 pixels on the input image respectively. For each cell the network estimates 16 numbers that represent presence of curb lines in one of 4 categories and their adjustments. Estimating presence of a curb line is a classification problem, but estimating adjustments to that line is a regression problem. To teach the network to perform the classification and regression at the same time, a discrete-continuous loss is applied during the training process.

1) Discrete-continuous loss: Total loss of the model L_t is defined as:

$$L_t = L_d + \alpha L_c \tag{1}$$

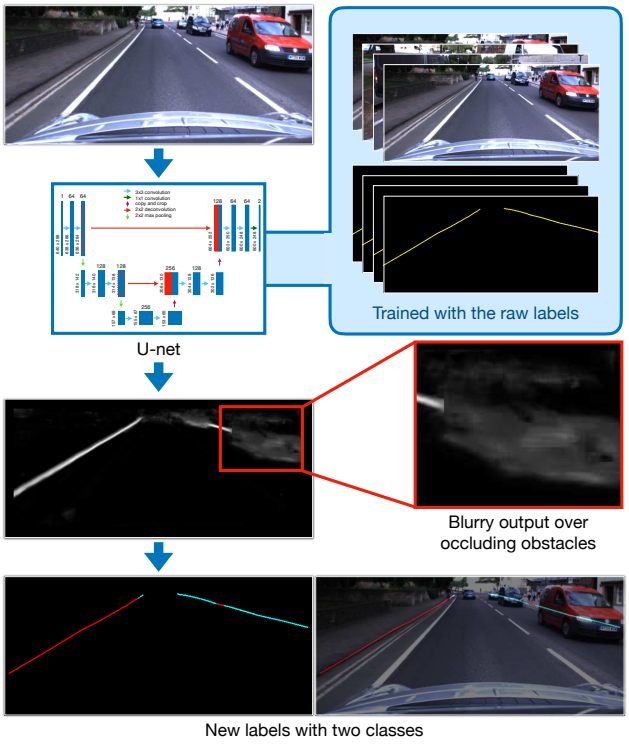


Fig. 5: Given input images, the U-net model trained with the raw labels detects *visible* road boundaries, but generates blurry outputs over occluding obstacles. We obtain masks for detected *visible* boundaries by applying threshold to the outputs. AND operation between the raw labels and thresholded outputs give us labels for *visible* road boundaries. Labels for *occluded* road boundaries are obtained by subtracting labels for *visible* from the raw labels.

where L_d is discrete loss of curb line category classification, L_c is continuous loss of curb line parameters regression and α is the weight term. The discrete and continuous losses are defined as:

$$L_d = \sum_{i=1}^{S} L_{d_i} \tag{2}$$

and

$$L_c = \sum_{i=1}^{S} L_{c_i} \tag{3}$$

respectively, where S is the number of scales (there are 3 scales). Let $\hat{p}^k_{i,j}$ be a softmax output of the network for the k-th anchor line category in j-th cell of the i-th scale, then

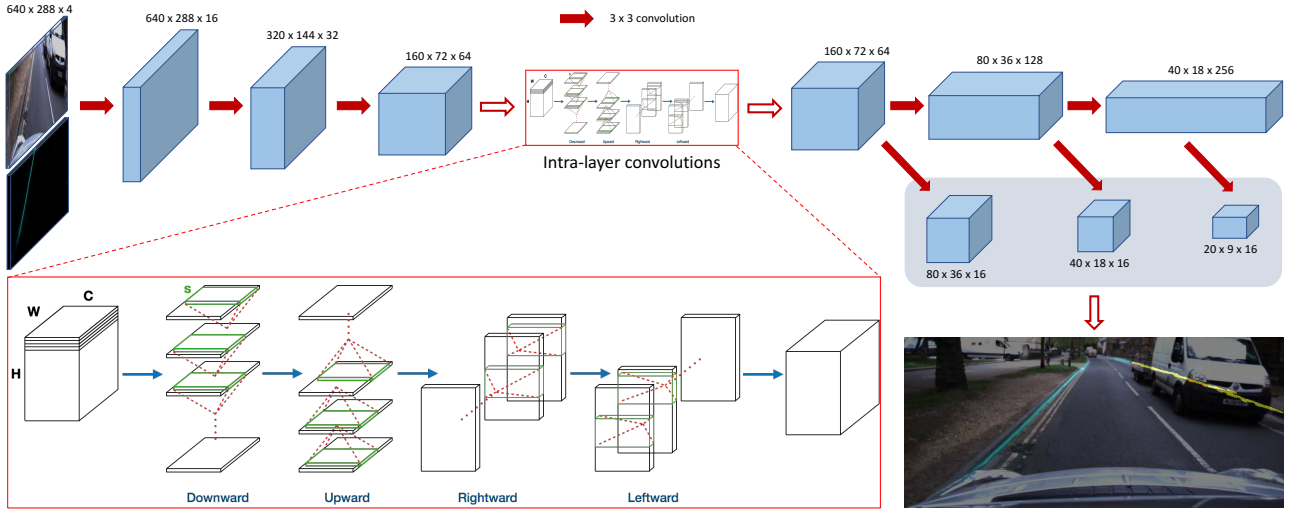


Fig. 6: Our model architecture for inferring *occluded* road boundaries. The model takes an RGB image and mask of detected *visible* road boundaries as inputs. The model consists of convolutional layers and there are 3 "base" layers followed by intra-layer convolutions that are slice-by-slice convolutions within feature maps. The intra-layer convolutions increase the capacity of the model to capture information from all over the image and spatial relationships across columns and rows. They are applied in 4 directions: downward, upward, rightward and leftward. Last 3 layers of the model progressively decrease in size and allow multi-scale predictions.

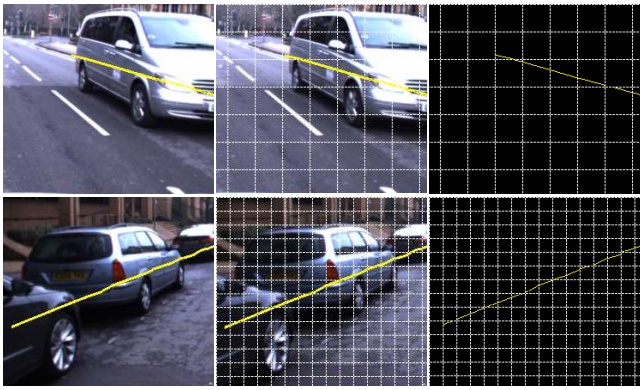


Fig. 7: Examples of parameterisation of *occluded* road boundary labels. In the left column pixel-wise labels are shown followed by the division of pixel-wise masks into a grid of squares at different scales. The final *occluded* road boundary masks are drawn based on parameterised labels in the right column. The grids on the first and second rows have sizes of 32 x 32 and 16 x 16 pixels respectively.

the discrete loss for the i-th scale is:

$$L_{d_i} = -\sum_{j=1}^{C_i}\sum_{k=1}^{A}(y_{i,j}^k \log(\hat{p}_{i,j}^k) + (1-y_{i,j}^k)\log(1-\hat{p}_{i,j}^k)) \quad (4)$$

where A is the number of anchor line categories (there are 4 categories), C_i is the number of cells in the i-th scale and $y_{i,j}^k$ is the ground truth for the k-th anchor line category in j-th cell of the i-th scale. The continuous loss is a smooth $L1$ loss between the predicted line ($\omega_{i,j,pr}^k$, $\beta_{i,j,pr}^k$) and the ground truth line ($\omega_{i,j,gt}^k$, $\beta_{i,j,gt}^k$) parameters. The continuous loss for the i-th scale is defined as:

$$L_{c_i} = \sum_{j=1}^{C_i}\sum_{k=1}^{A}(y_{i,j}^k(smooth_{L1}(\omega_{i,j,pr}^k - \omega_{i,j,gt}^k) + smooth_{L1}(\beta_{i,j,pr}^k - \beta_{i,j,gt}^k)) \quad (5)$$

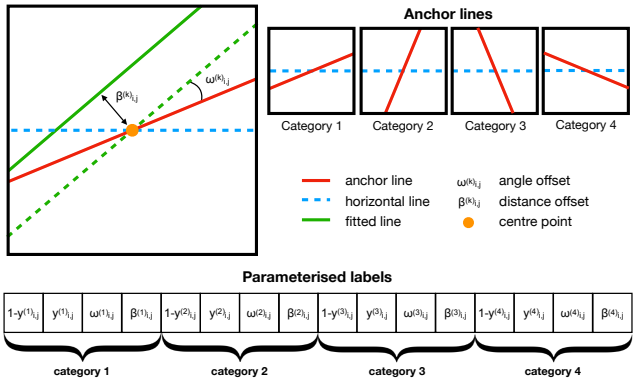


Fig. 8: Parameterisation of road boundary lines in discrete-continuous form. Each cell of the grid at each scale is represented with 16 parameters: 4 numbers for each line category. Lines are fitted to road boundaries in each cell and are assigned to one of 4 anchor line categories. Then offsets from the anchor lines to the fitted lines are calculated: angle offset between fitted and anchor lines ($\omega_{i,j,gt}^k$), and distance offset from the centre point of the cell to the fitted line ($\beta_{i,j,gt}^k$).

where the $smooth_{L1}$ is [16]:

$$smooth_{L1}(d) = \begin{cases} 0.5d^2 & \text{if } |\mathrm{d}| \leq 1 \\ |\mathrm{d}| - 0.5 & \text{otherwise} \end{cases} \quad (6)$$

2) Intra-layer convolutions: The length of the occluded curbs clearly depends on the size of occluding objects, ranging from 10-15 pixels long curbs occluded by traffic cones to 200-300 pixels long ones occluded by cars parked one after another. Estimating the correct orientation and position of occluded curbs in crowded areas requires the model to have a large receptive field. To increase the capacity of the model to capture information from all over the image and spatial relationships across columns and rows, we added intra-layer convolutions [17] before the multi-scale parameter

estimation layers. Traditional layer-by-layer convolutions are applied between feature maps, but intra-layer convolutions are slice-by-slice convolutions within feature maps. This enables the model to propagate spatial information across rows and columns as illustrated in Figure 6.

Given a 3D tensor, C x H x W (C - channels, H - height, W - width), intra-layer convolutions are applied in four directions, downward, upward, rightward and leftward. To apply convolutions downward, the tensor is split into H slices, where H in the number of rows. Starting from the top slice, convolution with kernel size C x s, where C is the number of channels and s is the kernel width, is applied to the first row and the output is added to the second row. Then the convolution is applied to the updated second row and output is added to the next row. This process continues until reaches to the bottom row. Similarly, intra-layer convolutions are applied upward, rightward and leftward.

V. Road Boundary Set Formation

From the output of the network, we need to transform pixel-wise output into a set, with unknown cardinality, of semantically and geometrically meaningful road boundaries. When the motion of the vehicle is parallel to the road boundaries, simple approaches such as using the pixels to the left and right to form two separate boundaries would lead to a usable set. However, the motion of the vehicle in comparison to the road boundaries set is not known *a-priori*. And of course road junctions present a larger number of road boundaries where this simple approach fails.

Instead we opt for a robust global energy based formulation based on a Convex Relaxation Algorithm (CORAL) [18] that has been shown to be superior to greedy sampling techniques such as the widely used Random Sampling and Consensus (RANSAC) [19] algorithm at detecting multiple geometric primitives. This approach jointly optimises the overall assignment of points to models by while seeking compact solution that explains the data with as few models as possible. This allows in our case for a minimal number of best-fit cubic curves (boundaries) to be associated to the network output with the global energy shown in Equation 7:

$$\sum_{l=1}^{L}\Bigg(\underbrace{\sum_{i=1}^{n}(\|D(\mathbf{A}_l\mathbf{u}_i)\|)\phi_l(\mathbf{u})}_{\text{Data Term}}+\lambda\underbrace{\sum_{i=1}^{n}|\nabla_{\mathscr{N}}\phi_l(\mathbf{u})|}_{\text{Smoothness Term}}\Bigg)+\underbrace{\beta\|L\|}_{\text{Compactness Term}} \tag{7}$$

The data term in Equation 7 accounts for the distance between a point and a curve model. Here A is the curve equation $\mathbf{A}=(a_0,a_1,a_2,a_3)$ and we refer to D as the Euclidean distance between a point $\mathbf{u}_i=(x,y)$ and the curve A. The assignment of data points to their respective models is encapsulated through an indicator function

$$\phi_l(\mathbf{u})=\begin{cases}1 & \mathbf{u}\in L_l\\ 0 & \text{otherwise}\end{cases} \tag{8}$$

where the uniqueness in the label assignment can be achieved by adding the constraint $\sum_{l=1}^{L}\phi_l(\mathbf{u})=1$. To account for outliers –where some data points might not be explained by a geometric model– a special label $\emptyset$, representing the outlier model is added. In this way a constant cost, γ, is assigned to points that cannot be explained by any geometric model. The model cost for the outlier model is simply given by $D(\mathbf{A}_\emptyset,\mathbf{u})=\gamma$.

The smoothness term in Equation 7 promotes a homogeneous assignment of labels to neighbouring points. The $\nabla_{\mathscr{N}}$ operator calculates the gradient of the indicator function over the neighbourhood $\mathscr{N}$ of a point and penalises points that belong to the same neighbourhood but do not share the same model. The parameter λ controls the trade-off between the smoothness cost and the data cost. Finally, the third term in Equation 7 penalises the number of models by adding a constant cost β per model. This eliminates redundancies in models resulting in a compact solution.

For the minimisation of this energy CORAL leverages a primal dual optimisation that utilises a parallel approach implementable on a General Purpose Graphical Processing Unit (GPGPU) is able to achieve real-time performance on geometric model detection. Due to space constraints, we refer the reader to [18] for further implementation details. The minimisation thus reveals a minimal set of cubic curves that encapsulate the road boundaries.

VI. Experimental Results

In this section we provide qualitative and quantitative results for experiments carried out to test the performance of our approach. Due to the lack (to the best of our knowledge) of a public road boundary detection benchmark, comparison with other existing approaches could not be undertaken. Nevertheless, we present quantitative results based on evaluations with our ground truth test data. Our experiments consider an assessment that demonstrates the importance of the intra-layer convolutions for inferring occluded road boundaries.

In order to evaluate the proposed road boundary detection approach, we used one of the datasets from OxfordRobotcar Dataset [12] that wasn't included in the training process. Qualitative results (Figure 10) show that our approach is able to produce accurate pixel-wise visible road boundary detection and infer occluded ones even the percentage of occlusions increases. This is followed by cubic curve fitting to reveal the road boundary models using CORAL. Figure 11 shows that CORAL is able to reveal the minimum set of road boundary models that represent the viewed scene without making any assumptions of the number of geometric models available over a diverse set of viewed scenes.

As mentioned in Section III, the training and ground truth data contain semi-annotated curb masks as they were generated by projecting the 3D annotated points to the images under some constraints. To calculate the accuracy of the trained models, we selected 1000 images that have all the road boundaries annotated ignoring the 50 px height area on top of the images as illustrated in Figure 9. Note that the width of curb line annotations on the ground truth masks are always the same regardless of the height of curbs as the annotated points don't contain any such information. To compensate that we have 4 px tolerance when calculating precision, recall and F1 score. Table I summarises the accuracy for the whole system and separately for U-net and our model.

Fig. 9: Left: the area inside the green box is taken into consideration during the evaluation. Right: the width of the ground truth curb line is always the same regardless the height of the curb.

TABLE I: Precision, recall and F1 score of the model

Labels	Precision	Recall	F1 Score
Visible road boundaries only	97.01	92.64	94.77
Occluded road boundaries only	90.47	88.24	89.34
All road boundaries	96.17	92.68	94.40

In Section IV we emphasised the importance of the intra-layer convolutions for inferring occluded road boundaries. They increase the capacity of the model to capture information from all over the image and spatial relationships across columns and rows, which enables the model to infer occluded boundaries behind obstacles in any size. To demonstrate the importance of the intra-layer convolutions in practice, we trained our model by taking out intra-layer convolutions from the network and evaluated against our ground truth data. Table II summarises the results, where 12.26% performance drop can be seen for occluded road boundaries.

TABLE II: Precision, recall and F1 score of the model without intra-layer convolutions

Labels	Precision	Recall	F1 Score
Occluded road boundaries only	78.19	76.00	77.08
All road boundaries	94.37	90.62	92.45

Our proposed approach is implemented in Python with TensorFlow library. Running times for the whole pipeline and for some of its tasks are presented in Table III. With input images of size 640 x 288, the system runs at 8.33 Frames Per Second (FPS) on a NVIDIA 1080 Ti GPGPU.

TABLE III: Average Running time per task

Tasks	Milliseconds	FPS
U-net	50	20.0
Our Model	65	15.3
U-net + Our Model	104	9.56
U-net + Our Model + Post Processing	120	8.33
CORAL	90	11.11

VII. Conclusions

In this paper, we presented a method to detect and infer road boundaries from mono-images irrespective of whether or not the boundaries are actually visible. We demonstrated that our coupled approach first segmented visible road boundaries with U-net and then inferred occluded road boundaries with our our CNN-based network that contained the intra-layer convolutions and produced outputs in a hybrid discrete-continuous form. Our approach worked without any assumptions about 3D structure, shape or appearance of road boundaries and didn't use any temporal information to infer occluded road boundaries. To easily generate training data for our models, we presented an image annotation framework that enabled us to generate *visible* and *occluded* road boundary masks for hundreds of images within an hour. Through our experiments we demonstrated that our approach achieved high performance for both *visible* and *occluded* road boundaries. Finally, we performed the model selection step to return cubic representation of road boundaries without placing assumption on the number of continuous road boundaries within the scene.

References

[1] T. Suleymanov, L. M. Paz, P. Piniés, G. Hester, and P. Newman, "The Path Less Taken: A Fast Variational Approach for Scene Segmentation Used for Closed Loop Control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, October 2016.

[2] V. Prinet, J. Wang, J. Lee, and D. Wettergreen, "3D road curb extraction from image sequence for automobile parking assist system," *Proceedings - International Conference on Image Processing, ICIP*, pp. 3847–3851, 2016.

[3] M. Kellner, U. Hofmann, M. E. Bouzouraa, and N. Stephan, "Multi-cue, model-based detection and mapping of road curb features using stereo vision," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 1221–1228.

[4] L. Wang, T. Wu, Z. Xiao, L. Xiao, D. Zhao, and J. Han, "Multi-cue road boundary detection using stereo vision," in *2016 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, July 2016.

[5] F. Oniga, S. Nedevschi, and M. M. Meinecke, "Curb detection based on a multi-frame persistence map for urban driving scenarios," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, Oct 2008, pp. 67–72.

[6] M. Kellner, M. E. Bouzouraa, and U. Hofmann, "Road curb detection based on different elevation mapping techniques," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 1217–1224.

[7] J. Siegemund, U. Franke, and W. Förstner, "A temporal filter approach for detection and reconstruction of curbs and road surfaces based on conditional random fields," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, June 2011, pp. 637–642.

[8] M. Enzweiler, P. Greiner, C. Knöppel, and U. Franke, "Towards multi-cue urban curb recognition," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 902–907.

[9] A. Y. Hata and D. F. Wolf, "Feature detection for vehicle localization in urban environments using a multilayer lidar," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 420–429, Feb 2016.

[10] W. Yao, Z. Deng, and L. Zhou, "Road curb detection using 3d lidar and integral laser points for intelligent vehicles," in *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, Nov 2012, pp. 100–105.

[11] Y. Zhang, J. Wang, X. Wang, and J. M. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3d-lidar sensor," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–11, 2018.

[12] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498

[13] C. Linegar, W. Churchill, and P. Newman, "Work Smart, Not Hard: Recalling Relevant Experiences for Vast-Scale but Time-Constrained Localisation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *ECCV 2016*, 2016.

[16] R. B. Girshick, "Fast R-CNN," in *ICCV*, 2015.

Fig. 10: Sample outputs from the network under different levels of road boundary occlusion. The network is able to seamlessly deal with and predict the position of the road boundaries in scenarios of low to no occlusion (left column), some occlusion (middle column) and full occlusion (right column). The blue pixels are the visible road boundaries while the yellow pixels represent the occluded boundaries.

Fig. 11: Sample results of geometric road boundary model extraction from the network output. CORAL is able to obtain the minimum set of cubic curves that represent the road boundaries, without any prior information of the number of models, in a diversity of viewed scenes.

[17] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," *arXiv preprint arXiv:1712.06080*, 2017.

[18] P. Amayo, P. Piniés, L. M. Paz, and P. Newman, "Geometric Multi-Model Fitting with a Convex Relaxation Algorithm," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, June 2018.

[19] 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.

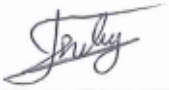
Statement of Authorship for joint/multi-authored papers for PGR thesis

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	Inferring Road Boundaries Through and Despite Traffic
Publication Status	☑ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	T. Suleymanov, P. Amayo, and P. Newman, "Inferring Road Boundaries Through and Despite Traffic," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Maui, Hawaii, USA, 2018.

Student Confirmation

Student Name:	Tarlan Suleymanov		
Contribution to the Paper	My contributions to the paper were: Developing the initial idea behind the paper. Dataset annotation and preparation. Designing the models and framework. Running the experiments, analysis and interpretation of the data. Drafting and writing the paper. Designing and making the figures.		
Signature		Date	12 September 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: Prof Paul Newman		
Supervisor comments I AGREE		
Signature P. Newman	Date	12/9/2019

This completed form should be included in the thesis, at the end of the relevant chapter.

5.7 Summary of the Paper's Results

The experimental results presented in the paper demonstrated that the proposed approach achieved high accuracy in detecting visible and inferring occluded road boundaries. Overall, the F1 score for detecting all road boundaries reached 0.9440, where it was 0.9477 for detecting visible road boundaries only and 0.8934 for inferring occluded road boundaries. Our experiments showed that the intra-layer convolutions play an important role in inferring occluded road boundaries as they capture contextual information across rows and columns. Taking out the intra-layer convolutions resulted in a performance drop of 0.1226 from 0.8934 to 0.7708.

5.8 Further Experimental Results

We conducted further quantitative and qualitative experiments to evaluate the performance of the models and their components. We present the results here that were omitted from the paper due to space constraints.

5.8.1 Importance of multi-scale predictions

Table 5.1: Accuracy of the camera-based ORBI models depending on the number of scales. The highest F1 score is achieved when using three scales for the parameterised outputs.

Number of scales	Precision	Recall	F1 Score
3 scales	0.9335	0.9063	**0.9197**
Last 2 scales	0.9051	0.8332	0.8677
Last 1 scale	0.9372	0.7922	0.8587
First 2 scales	0.9459	0.8100	0.8727
First 1 scales	0.9807	0.7868	0.8731

In our ITSC 2018 paper (Section 5.5, [14]) we briefly mentioned that it is important to have predictions of occluded road boundaries in multiple scales due to different sizes and shapes of occluding obstacles, such as other road users ranging from motorcycles and cars to buses and trucks. Additionally, depending on how far these obstacles are located from the camera, they can appear in any size in the input image as shown in Figure 5.16 (left) where the bus occludes everything.

In the paper, we mentioned that we decided to use three scales of parameterised outputs taking into account accuracy and size of the model. However, the details of the experiment were omitted to conserve space in the paper. We present the results here in Table 5.1, which shows that the highest F1 score is achieved when using three scales for the parameterised outputs. Note that the F1 score presented here for the ORBI model with three scales (0.9197) is higher than the F1 score presented in the paper (0.8934). This is due to the model that was trained for the multi-scale experiment being trained with more data (~1K).

5.8.2 Quantitative Results

Table 5.2: Precision, recall and F1 score of the camera-based VRBD and ORBI models.

Labels	Precision	Recall	F1 Score
Visible road boundaries only	0.9800	0.8898	0.9327
Occluded road boundaries only	0.9198	0.8618	0.8899
All road boundaries	0.9790	0.8912	0.9330

In Chapter 2, we mentioned that 550 images were annotated from the 28-07-16 dataset, which did not include the streets from the Oxford RobotCar Dataset. We used these samples to further evaluate our camera-based road boundary detection approach and the results are presented in Table 5.2. Similar to the results presented in the paper, the F1 score for detecting all road boundaries reached 0.9330, 0.9327 for detecting visible road boundaries only and 0.8899 for inferring occluded road boundaries.

5.8.3 Qualitative Results

To demonstrate the ability of the model to infer outputs in a variety of scenarios, we present further qualitative examples here. Figure 5.9 shows examples where only visible road boundaries are present and detected by the VRBD model and no boundaries were hallucinated by the ORBI model. Output samples with both visible and occluded road boundaries are shown in Figure 5.10. For more examples, see Appendix A.

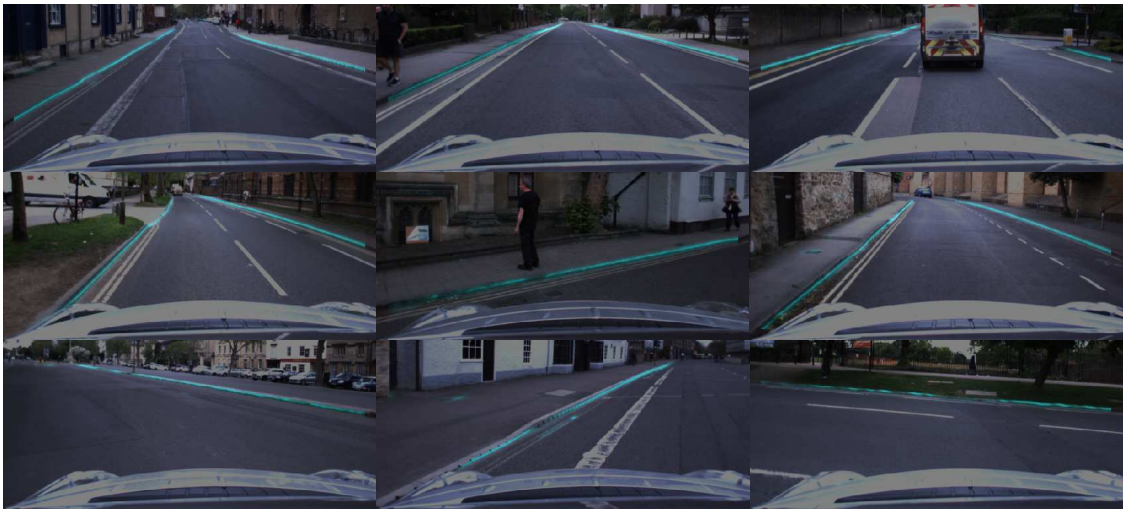


Figure 5.9: Camera-based visible only road boundary detection examples.

In some cases we observe that the ORBI model tries to detect visible road boundaries when the VRBD model fails to fully detect them. The VRBD model only captures context locally and takes into account the appearance and structure of visible road boundaries, but the ORBI model learns how to infer position of occluded road boundaries from context and never "sees" the actual appearance of those boundaries. This enables the ORBI model to detect visible road boundaries (using contextual information) that are not detected by the VRBD model as shown in Figure 5.11. Note that the ORBI model receives the mask of detected visible road boundaries as an input that provides information to the model about the location of detected boundaries.

Similar behaviour may manifest when lighting conditions change. In Figure 5.12, two consecutive frames are shown in which the visible road boundaries are detected first, but in the next frame the VRBD model failed to detect the visible road boundary on the right hand side of the road due to underexposure. The visible boundary that became "invisible" due to underexposure was inferred by the ORBI model.

To further test this behaviour, an experiment was conducted by masking out visible road boundary sections in various colours. As shown in Figure 5.13, the masked out sections of the road boundaries cannot be detected as visible road

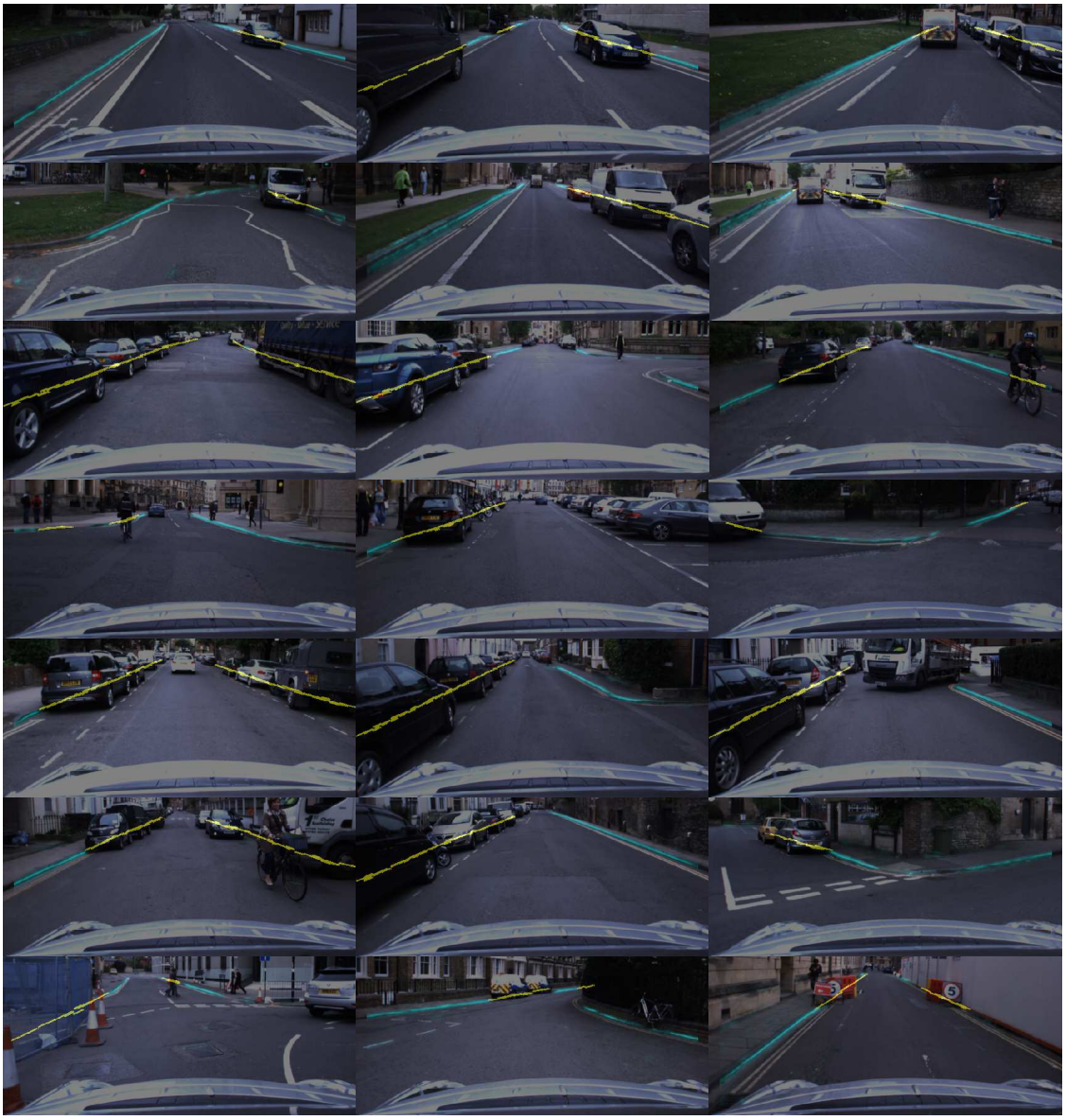


Figure 5.10: Camera-based road boundary detection examples. Visible road boundaries (cyan) are detected by the VRBD model and occluded road boundaries (yellow) are hallucinated by the ORBI model.

Figure 5.11: Visible boundaries are detected by the ORBI model as the VRBD model failed to detect them.

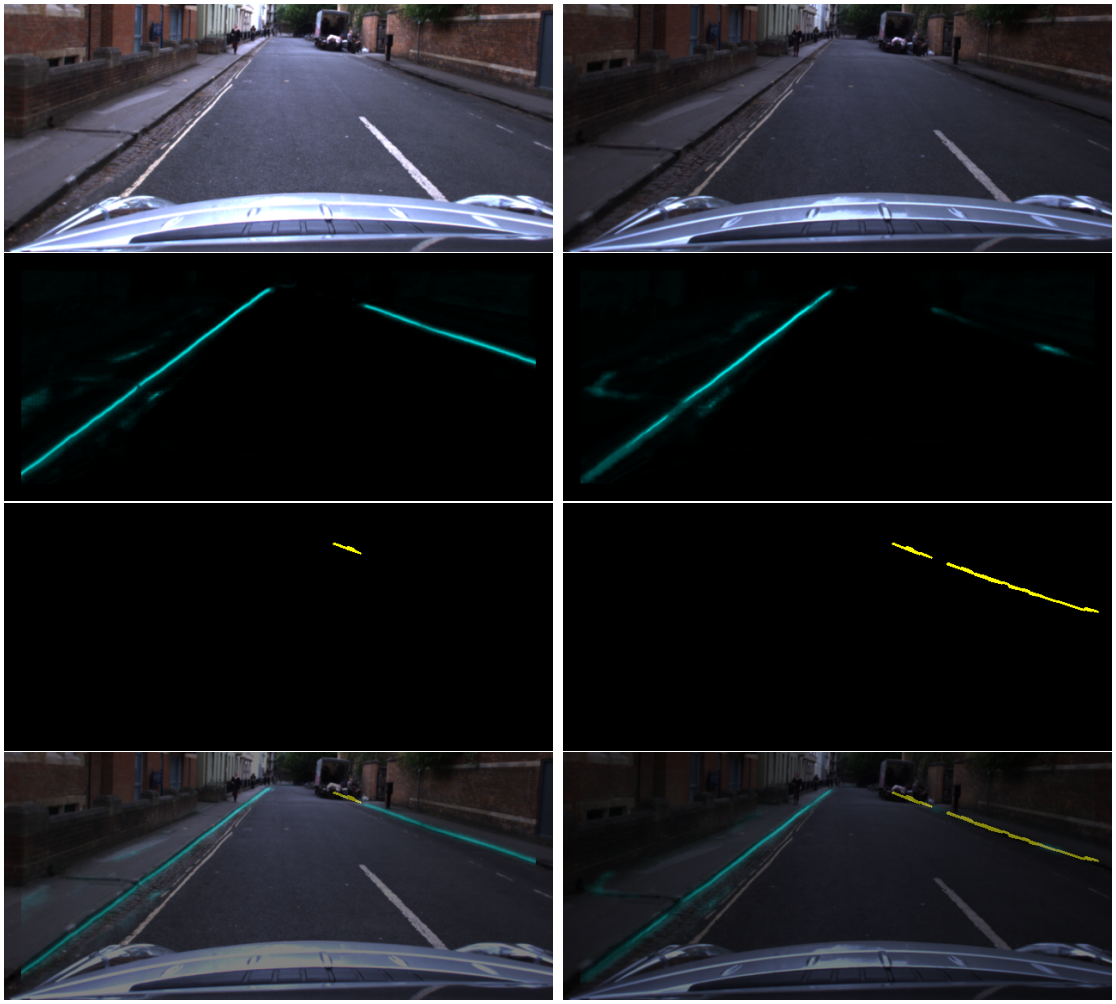


Figure 5.12: Visible road boundary on the right hand side of the road is not detected due to underexposure, but inferred as an occluded boundary.

boundaries. Instead, those sections are inferred by the ORBI model as occluded road boundaries. This experiment and the examples above demonstrate that the ORBI model can infer occluded road boundaries using contextual information regardless of structure, shape, or colour of occluding obstacles.

Another similar experiment was conducted by cropping out a car from one image and inserting it into other images to observe the changes in outputs. First, we selected an image with a car and passed it through our networks to detect road boundaries, as shown in Figure 5.14a, where visible road boundaries were detected and the occluded section behind the cars was inferred as expected. We selected another image without any occluding obstacles and obtained full detection of visible

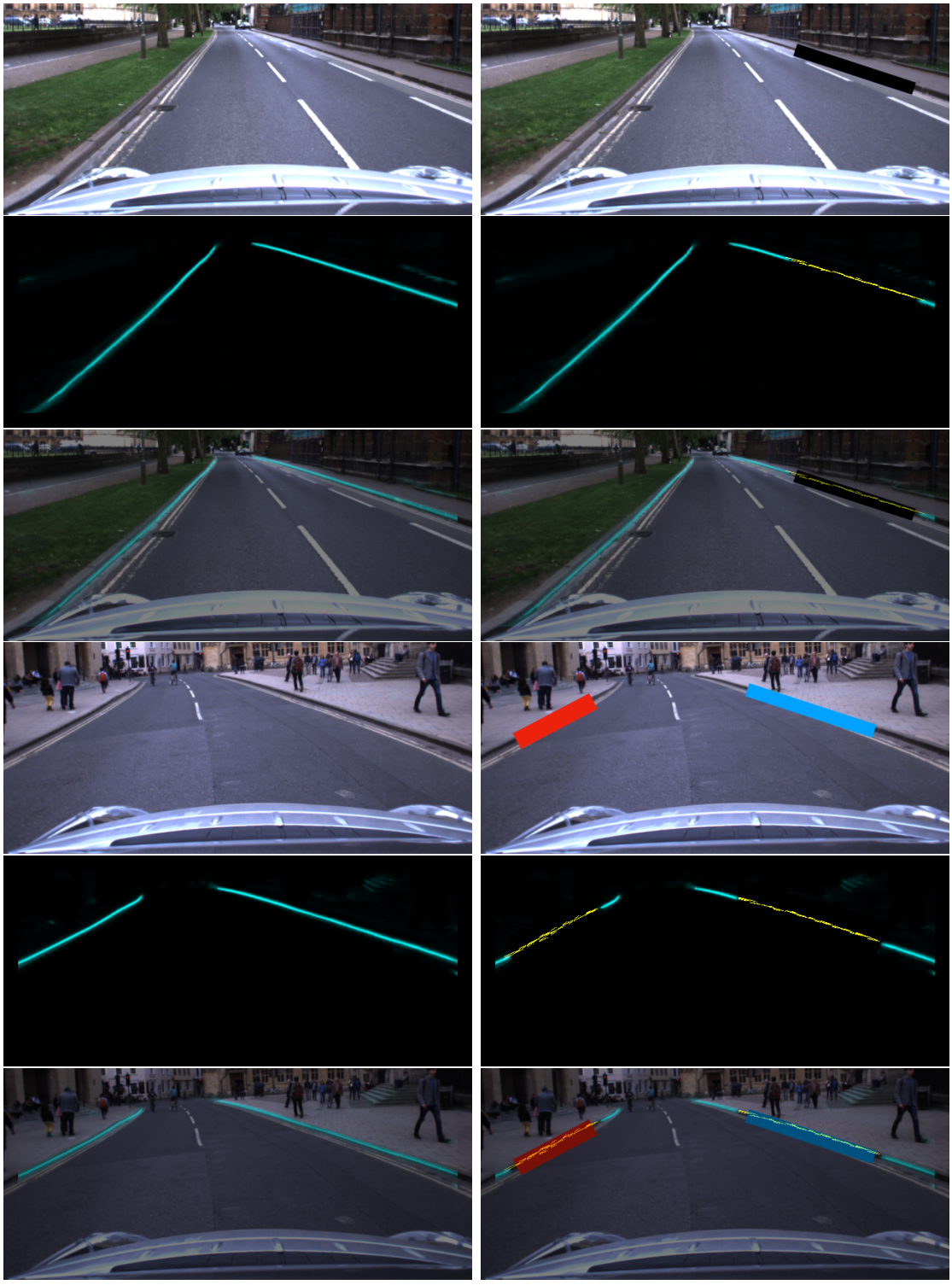


Figure 5.13: Masking out sections of visible road boundaries to experiment the ability of the ORBI model to infer occluded road boundaries regardless of structure, shape, or colour of occluding obstacles. Road boundaries were fully detected as visible road boundaries in the original images (left column), but the occluded sections in the edited images were inferred as occluded boundaries (right column).

(a) An image with the occluding black SUV car.

(b) An image without occlusions.

(c) The SUV car from the image (a) overlaid on top of the image (b).

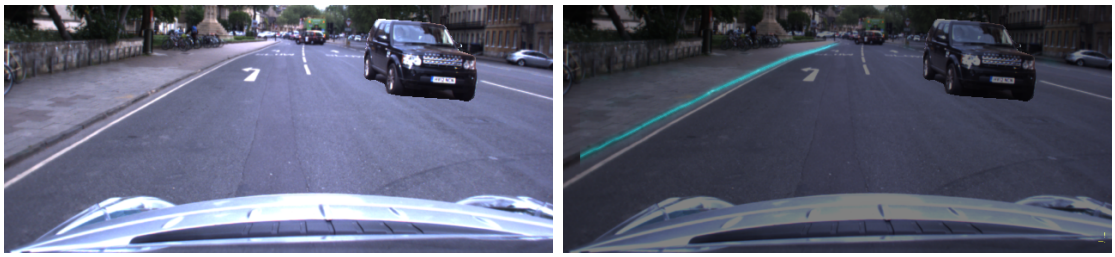
(d) The yellow coloured mask in the shape of the car overlaid on top of the image (b).

Figure 5.14: To demonstrate the ability of the ORBI model, a car from one image was cropped out and inserted into other images. (a): The occluded road boundary was inferred as expected over the occluding SUV car. (b): All road boundaries were detected by the VRBD model as expected. (c) and (d): The section of the road boundaries occluded by the inserted car or by its coloured mask were inferred by the ORBI model.

road boundaries as can be seen in Figure 5.14b. Then, we cropped the black SUV car from the first image and added it to the second image. As shown in Figure 5.14c, this resulted in a change of the road boundary section overlaid by the car from being detected as visible to being inferred as occluded. The same happened when the image was overlaid with a coloured mask in the shape of the car. As expected,

the VRBD model could not detect the section of the road boundary that became occluded, where the ORBI model inferred it by capturing the contextual information. This is again, regardless of the structure, shape, or colour of the occluding obstacles.

(a) An image without occlusions and no visible road boundary on the right hand side.



(b) The SUV car overlaid on top of the image (a).

Figure 5.15: When the car is added to the image, we do not observe any changes in the outputs. This demonstrates that the ORBI model did not "blindly" learn to infer occluded road boundaries over obstacles, but learned to look for clues to generate reasonable outputs.

We experimented with the same scenario using an image with no boundaries on the right hand side, where the networks do not detect or infer anything (Figure 5.15a). When the car is added to the image, we do not observe any changes in the outputs (Figure 5.15b). This demonstrates that for the network, the presence of cars does not always mean that there must a boundary behind it and the ORBI model infers occluded road boundaries considering road structure and other clues. The network did not "blindly" learn to infer occluded road boundaries over obstacles, but learned to look for clues to generate reasonable outputs.

Further examples showed that in the absence of road boundaries, the models did not detect or infer anything to avoid any unreasonable outputs. In the left image of Figure 5.16 everything was occluded by a bus passing in front of the test vehicle, where the models did not infer any road boundaries. Similarly, in the right image of the same figure the presence of cars and the complexity of

Figure 5.16: Left: full occlusion and no contextual information to rely on. Right: impossible to infer any road boundaries without temporal information or any prior knowledge due to complexity of the scene and presence of occluding vehicles.

the scene makes it impossible to infer any road boundaries without temporal information or any prior knowledge.

5.8.4 Failure Cases

As we discussed in Chapter 2, cameras are highly dependent on lighting conditions and bright sunlight, headlights of oncoming traffic, strong shadows, or absence of light during nights could potentially "blind" the cameras. We conducted experiments to demonstrate some failure cases due to lighting conditions.

Night. Cameras depend on lighting conditions and absence of light at night often makes camera-based methods fail. We tested our VRBD model with samples from a night-time dataset and unsurprisingly the model could not detect road boundaries (Figure 5.17). Note that the model was trained with samples only from day-time.

Over- and Underexposure. Similarly to an absence of light, overexposure and underexposure effect the performance of camera-based approaches. We present examples of overexposure in Figure 5.18, where the both models failed to detect visible or occluded road boundaries to a reasonable level.

Lack of contextual information. In Section 5.8.3, we demonstrated the importance of capturing contextual information for estimating the correct position and structure of occluded road boundaries. The ORBI model infers occluded road boundaries from a single frame without temporal information or prior knowledge

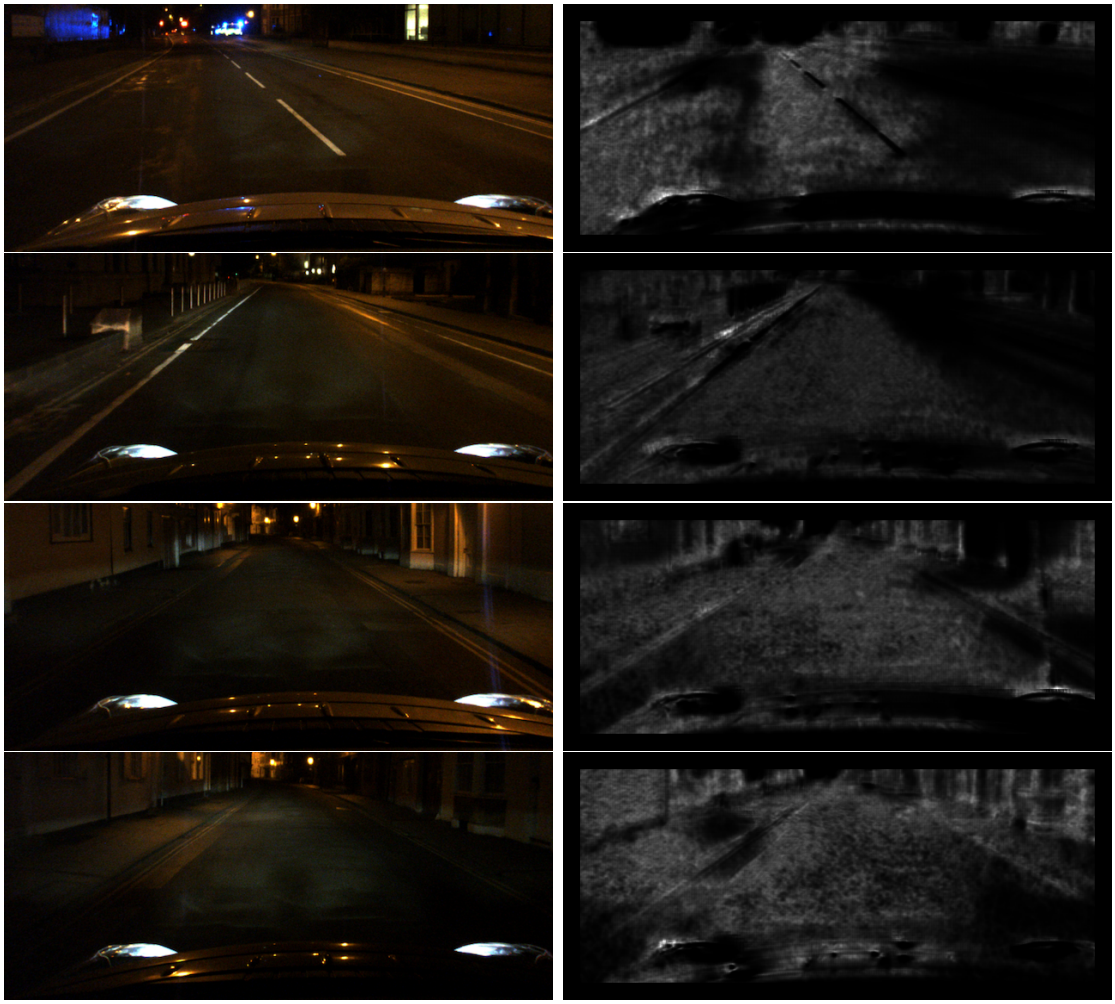


Figure 5.17: Output samples from a night-time dataset. The VRBD model fails to detect road boundaries as it was trained with only day-time samples.

and in some cases lack of contextual information may lead to inaccurate results. In Figure 5.19, the ORBI model failed to accurately estimate correct position of the occluded road boundary on the left hand side of the road due to the car on the left. The car blocked the view and created the misleading scenario, in which the ORBI model inferred the occluded road boundary as the continuation of the detected visible road boundary on the left hand side of the road. Although the ORBI model correctly inferred the existence of the occluded road boundary behind the car it could not accurately infer the position of the boundary due to lack of contextual information. This could be fixed by integrating temporal information or prior knowledge about the scene, such as OpenStreetMap, but this is beyond

Figure 5.18: Overexposure output samples, where the camera-based models failed to detect road boundaries to a reasonable level.

Figure 5.19: Example of a failure case due to lack of contextual information, where the camera-based models failed to estimate correct position of the occluded road boundary on the left hand side of the road.

the scope of this thesis and we will address this issue in future work.

5.9 Scene Understanding Experiment

In the context of autonomous driving, understanding of the surroundings and accurate representations are very important for planning and decision making. Self-driving cars are required to make complex manoeuvres in real-world scenarios that require high-level reasoning of the environment. Deep learning methods have achieved impressive results in semantic segmentation in recent years, but they mostly provide pixel-wise outputs which are not sufficient for navigation in complex urban environments. To address this problem, in our Intelligent Transportation Systems Conference (ITSC) 2018 paper [27] (Section 5.10) we introduced the scene graph, which is a graph-based hierarchical representation of a scene from a partially

segmented image. We used background information and a logical representation to model the structure of roads. We segmented road boundaries and road markings [28] from input images using deep learning based detection networks. Having obtained partially segmented, pixel-wise outputs, we found clusters that represented entities in the scene. Then, from the set of segmented entities, scene graphs were generated using a learnt probabilistic grammar and an adapted version of the Earley parser [29]. The image was parsed from left to right in image space to generate the structure of the scene using this Earley algorithm. As a proof-of-concept, we used the Highway Code as domain knowledge to demonstrate that parts of a scene can be inferred based on scene graphs. Road boundaries intentionally and legally delimit drive-able space, which makes them an important entity for scene understanding as they define the borders of roads. As part of the proposed system, we applied our camera-based VRBD model for the detection of road boundaries that were used as semantic entities in reconstruction of road layout. The work presented in this thesis contributed to the proposed scene understanding approach by providing segmented masks of detected road boundaries.

Reading between the Lanes: Road Layout Reconstruction from Partially Segmented Scenes

Lars Kunze, Tom Bruls, Tarlan Suleymanov, and Paul Newman

***Abstract*— Autonomous vehicles require an accurate and adequate representation of their environment for decision making and planning in real-world driving scenarios. While deep learning methods have come a long way providing accurate semantic segmentation of scenes, they are still limited to pixel-wise outputs and do not naturally support high-level reasoning and planning methods that are required for complex road manoeuvres. In contrast, we introduce a hierarchical, graph-based representation, called *scene graph*, which is reconstructed from a partial, pixel-wise segmentation of an image, and which can be linked to domain knowledge and AI reasoning techniques.**

In this work, we use an adapted version of the Earley parser and a learnt probabilistic grammar to generate scene graphs from a set of segmented entities. Scene graphs model the structure of the road using an abstract, logical representation which allows us to link them with background knowledge. As a proof-of-concept we demonstrate how parts of a parsed scene can be inferred and classified beyond labelled examples by using domain knowledge specified in the Highway Code. By generating an interpretable representation of road scenes and linking it to background knowledge, we believe that this approach provides a vital step towards explainable and auditable models for planning and decision making in the context of autonomous driving.

I. Introduction

Autonomous vehicles need to perceive their surroundings accurately for safe decision making and navigation in complex urban environments. These highly-structured environments can be described by hierarchical graphs containing semantic and spatial constraints. Such graphical representations can be employed for (cost-based) planning, inferring object classes, or reasoning about missing or occluded parts. More importantly, they provide a way to explain the behaviour and decision making of the vehicle which is paramount for real-world deployment and adoption. In this paper, we introduce such a representation, which is generated from partially segmented scenes and allows us to reason about the environment.

Recently, deep semantic segmentation networks have achieved impressive results for pixel-wise scene understanding of images [1], [2]. However, these methods suffer from interpretation and debugging difficulties and often fail to include prior information or dependencies/constraints (in the output space). More importantly, they do not naturally support high-level reasoning which is required for planning and navigation.

Authors are from the Oxford Robotics Institute, Dept. Engineering Science, University of Oxford, UK. {lars, tombruls, tarlan, pnewman}@robots.ox.ac.uk

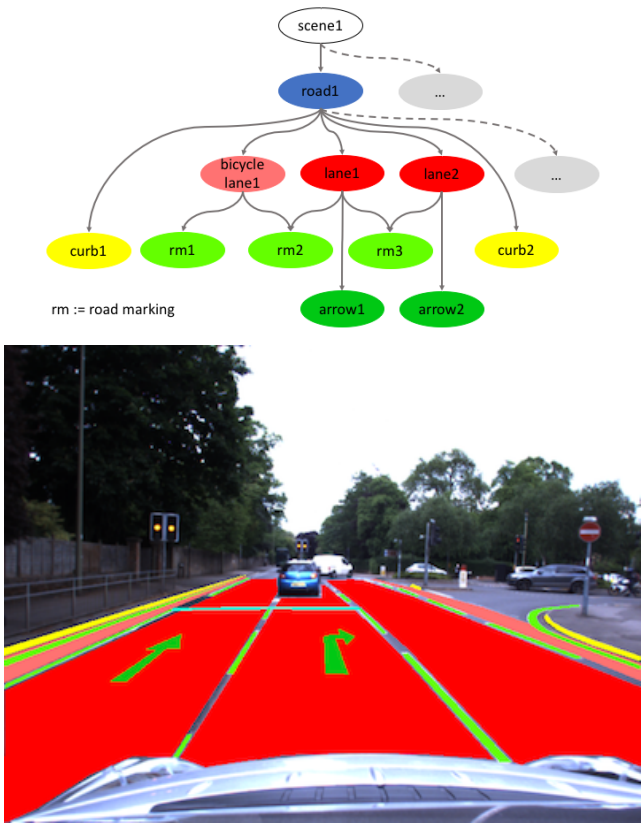


Fig. 1. Hierarchical scene graph representation (top) that was reconstructed from a partially segmented image (bottom). In this work we present a probabilistic scene parser that reconstructs the layout of road scenes from partial segmentations of road markings and curbs.

In contrast, all important (static or dynamic) objects influencing the decision making are detected separately in the mediated approach [3], [4]. This produces a world representation which can be employed more directly for planning and navigation. Interestingly, most approaches focus on detecting a single type of object or perform detection of several types of objects independently. Thereby they neglect that urban traffic scenes are highly structured and that there exist spatial and semantic constraints between objects, since these scenes are built and function according to specified rules.

Therefore, we introduce *scene graphs*, a hierarchical, graph-based representation, to model road layouts (i.e. lane geometries). Fig. 1 shows an example scene graph for a segmented road scene. We focus on the reconstruction of scene graphs from partial, pixel-wise segmentation. In par-

ticular, we consider segmented entities of road markings and curbs to reconstruct the semantic structure of road scenes. The road layout is reconstructed from these entities using both a learnt probabilistic context-free grammar and a learnt spatial, relational model. A road layout is chosen from a set of competing hypotheses by estimating the maximum a posteriori probability (MAP) of each model. Furthermore, we show that scene graphs can be refined by linking them with domain knowledge about the road construction, e.g. from the Highway Code.

In this paper, we make the following contributions:

- we introduce *scene graph*, a formal logic-based description of road scenes using a graph-based representation;
- we present an approach based on dynamic programming for parsing road scenes and reconstructing scene graphs from partial, segmentations and a learnt probabilistic grammar; and
- we demonstrate how scene graphs can be further refined and used for reasoning when linked to domain knowledge.

The remainder of the paper is structured as follows. We first discuss related work in Sec. II. In Sec. III, we provide and overview of the approach and explain how scene graphs are generated from both object segmentations and learnt prior models. In Sec. IV, we explain how scenes are partially segmented using deep networks for road markings and curbs. In Sec. V we explain how we represent a scene, learn both a probabilistic context-free grammar and a spatial relational model to describe scenes, and how scenes are parsed and interpreted using an adapted version of the probabilistic Earley parser. In Sec. VI, we showcase and discuss several examples of scene graphs and explain how they can be further refined. Lastly, we discuss possible application in Sec. VII before we conclude in Sec. VIII.

II. Related Work

In this section, we review different approaches for scene understanding in the context of autonomous vehicles. We mainly focus on graph based methods, since these are closest to the scene graph.

1) Graph-based Approaches: Representing the contents of scenes using graph-based approaches is not novel. In the context of urban traffic scenes, however, there exist only a few papers that take the spatial and semantic constraints into account by introducing graphs.

In [5], different sensor modalities and hierarchical graphs containing relational knowledge are fused to model traffic scenes. The output is still a pixel-wise segmented image not directly employable for automated driving.

Several other papers implement more high-level reasoning to infer the lane geometries. The authors of [6] introduce a theoretical, hierarchical framework including uncertainties to reason about multiple hypotheses for the lane geometry. Similar methods that work on real-world data are introduced in [7], [8]. From linear patches of lane markings a graph is built including their spatial relationship represented by continuous distributions and non-parametric belief propagation is used to infer the different lanes in the scene. However, these methods are not guaranteed to work in urban environments.

In [9], the lane separators are modelled as latent variables without linear constraints so that the framework becomes applicable to more complex scenes. By encoding geometric relationships at different levels (i.e. lane markings, lane separators, lanes, and road), the authors show that they improve inference of the lane geometries even in case of many false detections at the root nodes. This work is similar to our approach as we also represent the geometric relationships of different entities according to the hierarchy.

The driving rules of a traffic scene are given by the type of road markings that often appear in similar configurations. Therefore, [10] connects them as a graph and optimises a CRF with handcrafted spatial features of the road markings to predict their class. Similarly, we learn a distribution of geometric and relation features to predict and evaluate the type and the role of an entity within the hierarchy.

Work by [11] is most similar to our approach. In their work, they learn a probabilistic grammar based on a set of features and use a dynamic programming approach to generate a scene graphs which describe the furniture layout of synthetic indoor scenes. Whereas their approach considers full object knowledge from CAD models, our approach reconstructs scenes from partial observations of real-world environments.

2) Mediated Approaches: Proposed solutions differ widely in terms of the objects that are taken into account, used sensors, required computation time, usage of prior information, and abstraction level of the output. In general, our approach is flexible to consider different kinds of information from various resources. In this work, we consider segments of road markings and curbs as input.

In [12] a coarse road geometry/scene analysis is estimated from the acquired semantic segmentation. This framework is significantly extended in [3] where the precise intersection geometry is inferred from vanishing points, semantic labels, and tracklets of traffic participants. However, these methods cannot be used for navigation directly as they do not map to precise lane geometries and do not include the road rules. The former is solved in [13] by looking more closely into the tracks of the surrounding vehicles. Our also approach models the geometry of high-level concepts based on the low-level image segmentations. Thereby, information about lanes and boundaries can potentially be used for navigation planning. Through advancements in deep learning we have now come to a point where even reasoning of the space behind occluded parts of the images is possible for inferring road geometries [14]. In future work, we also plan to extent our work in this direction.

3) Deep Networks: All of above mentioned methods require handcrafted features/probabilities in some way to optimise the graph. It has been shown by now that deep networks with learned feature maps achieve much better semantic (instance) segmentation [1], [2] and thus understanding of the scene. Besides, they are able to generalise better when auxiliary output tasks are employed [15]. How-

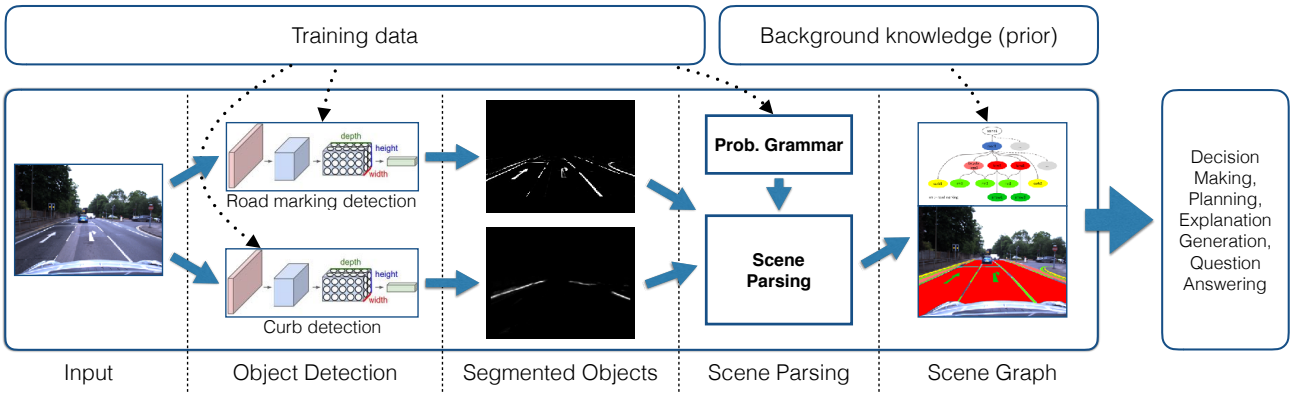


Fig. 2. Scene parsing approach based on road marking and curb detections. The approach has two main steps: (1) given an image, road marking and curb segments are detected by deep networks, and (2) given a set of detected segments, the scene is parsed using an adapted version of the Earley algorithm and a learnt probabilistic grammar. The resulting scene graph is integrated with domain knowledge and can used for planning and decision making.

ever, these networks suffer from interpretation and debugging difficulties and often fail to include prior information, high-level reasoning, or constraints (in the output space). Recently, some works have tried to improve some of these disadvantages by introducing spatial and semantic reasoning frameworks that can be trained in an end-to-end way [16]–[18]. In this work, we simply use deep networks as an effective way for segmenting an input image. However, our future goal is to extend this approach and to feed geometric, spatial, and semantic constraints back to the deep networks during learning.

III. Approach Overview

Our approach constructs a symbolic, graph-based description of the road layout given an image of a road scene (see Fig. 1). When interpreting the image, our approach considers two types of information: object detections and common road configurations based on learnt prior models.

Fig. 2 depicts the overall pipeline of our approach. We first segment the image by detecting curbs and road markings using trained deep networks (Sec. IV). These pixel-wise segmented images are clustered and the resulting entities are considered as input for a parsing process which generates a hierarchical scene representation (*scene graph*) (Sec. V). The parser takes object detections (and their uncertainty) and prior information of road scenes into account. Our probabilistic approach is in particular suitable for integrating incomplete and uncertain information from object detection pipelines. Each valid parse tree is scored by a probability which allows us to disambiguate between alternative representations. Intuitively, the score captures three aspects: (1) hierarchy (2) geometric features of detected entities, and (3) spatial relations between entities in the hierarchy. As we represent scene graphs using logical representations they can be linked to background knowledge and used for auditable planning and decision making.

IV. Scene Perception

This section describes how road markings and curbs are detected in a given image of a road scene. The resulting pixel-based images are clustered and segmented entities are obtained which are considered as input for the scene interpretation process described in Sec. V.

A. Road Marking Detection

Road markings are a critical component for (autonomous) driving especially in urban environments. The road rules are captured by their underlying meaning and they guide all traffic participants through potentially dangerous situations. Therefore, real-time detection and interpretation of road markings is an important cue for high-level scene understanding and aids planning and decision making.

Detecting all painted road markings (not just lane separators) on the road surface, which dictate the traffic rules for that particular urban setting, is a challenging problem for several reasons. Firstly, there are visual challenges such as occlusions, varying lighting, and changing weather conditions. Secondly, road markings vary from country to country and are often degraded. Lastly, there are no large datasets available for training with accurate ground-truth labels for road markings.

Road marking detection in images can be seen as a semantic segmentation problem. State-of-the-art methods for these tasks implement deep networks, which are able to learn specific scene context and thereby cope with the challenges stated above, as long as sufficient training data is available. Manually generating training data is extremely labour expensive, because of the required pixel-level detail in combination with the aforementioned visual issues. Therefore, we create road marking annotations in a weakly-supervised way, by leveraging complementary sensor modalities (i.e. LiDAR).

For generating the annotations, we exploit the property that road markings are highly reflective and must lie on the road surface. Firstly, we utilise the LiDAR point cloud to coarsely segment the road surface from the image. A dense CRF is then optimised to identify the road marking image pixels by

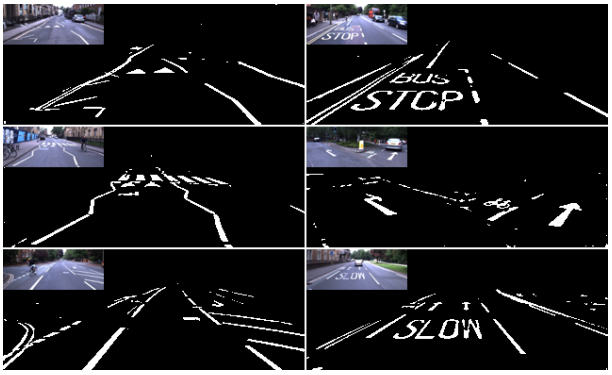


Fig. 3. Road marking detection performed by a deep semantic segmentation network in real-time. Before the detections are employed to generate the scene graph, they are mapped to top-down view.

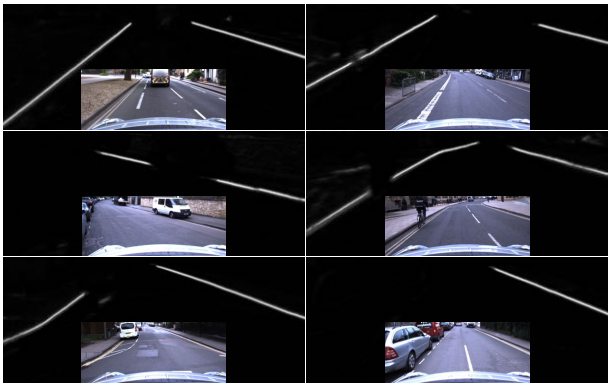


Fig. 4. Curbs are detected by a fully convolutional network. The network can detect visible curbs without making any assumptions about their 3D structure, shape or appearance.

corresponding them with the high-reflectance LiDAR points, which are not affected by varying lighting.

We employ these annotations to train a deep semantic segmentation network (inspired by U-net [19]) for road marking detection using only a monocular camera. The results demonstrate that the network segments the road markings from the image without any preprocessing steps, as shown in Fig. 3.

We direct the reader to [20] for a more detailed description of this method.

B. Curb Detection

Curbs (road boundaries) play an important role for autonomous cars as they intentionally and legally delimit driveable space. Curb detection using monocular images is a challenging problem. Road boundaries have narrow and long shapes which are not easily detectable. Deep networks often require large amounts of training data to obtain high-performance, well-generalised models. Due to colour, appearance, shape, perspective, illumination and background clutter, the training data should incorporate great variability changes. However, image by image hand labelling of the ground truth data is a time-consuming process. To avoid this problem and obtain a large amount of training samples, we generated 3D points cloud from 2D laser data and annotated points in the point cloud corresponding to road boundaries. Note that the 2D laser is attached vertically to the rear of a test car, which makes road boundaries easy to spot and annotate in the point cloud. The annotated points are projected to images of forward facing camera of the car. Lines are drawn between consecutive points to annotate road boundaries in-between the points. This way, hundreds of labelled images are obtained within an hour (approximately 750 images). A 10 kilometres dataset from the Oxford RobotCar Dataset introduced by [21] was annotated to generate several thousand semi-annotated masks. A vision based localiser was used to boost the number of training images by projecting labels from the annotated dataset to other traversals. However, some of the generated curbs masks contain annotations for occluded areas of curbs, such as over parked cars. To remove those redundant annotations, we trained U-net [19] with the raw masks and then run the inference with RGB images from the training data to generated output of detected curbs. The trained U-Net model can segment visible areas of curbs, but produces blurry outputs over occluding obstacles. Applying a threshold to the outputs gives us masks for detected visible curbs. We obtain labels for visible curbs by applying an AND operation between the thresholded outputs and raw labels. Finally, we train the U-net with visible curbs only (Fig. 4). A detailed description of our work on curb detection is given in [22].

V. Scene Interpretation

In the previous section, we explained how an input image is segmented into two classes: road markings and curbs. Before we describe how we learn a probabilistic grammar to parse these segmentations and construct a scene graph from them, we first introduce scene graphs formally.

A. Representation

Our motivation with this work is to support autonomous vehicles in their decision making, planning, and explanation generation. In particular, we aim at a representation that is interpretable (by machines and humans alike), extendable, and suitable for different inference tasks. To this end, we introduce *scene graphs* as a way to represent road scenes semantically using well-defined concepts and relations which are grounded in the vehicle's perception system.

Formally, scene graphs are represented in Description Logic; an overview is given in [23]. A scene is described by a set of instances of meaningful classes and their relations. For example, a *scene* is composed of a *road* which has two *curbs* and several *lanes* which in turn are bounded by several *road markings*. This hierarchical decomposition of a scene is important as we will explain later in Sec. V-C. In general, however, scene graphs can be linked flexibly to other information resources due to its underlying logical representation as we have shown in previous work [24]. For example, they can be linked to the outcome of detection and tracking algorithms of traffic participants and/or domain knowledge

TABLE I
SCENE GRAPH TAXONOMY

Class	Description
Scene	Root node of a scene graph. A *Scene* has at least one road (*Road*), but can have multiple.
Road	A road is delimited by at most two curbs (*Curb*) and has one or more lanes (*Lane*).
Curb	A curb is composed of one or multiple curb segments (*CurbSeg*).
Lane	A lane is bounded by road markings along the carriage way (*RMAlong*). Additionally, lanes can have road markings that are across the carriage way (*RMAcross*), and other road markings such as symbols and text (*RMOther*).
RMAlong	Road marking along the carriage way.
RMAcross	Road marking across the carriage way.
RMOther	Road marking of a symbol or text.
RMSeg	A road marking segment is a set of clustered pixels detected by the network described in Sec. IV-A. It can be one of three types: *RMALong*, *RMAcross*, or *RMOther*.
CurbSeg	A curb segment is a set of clustered pixels detected by the network described in Sec. IV-B.

TABLE II
LEARNT PROBABILISTIC CONTEXT-FREE GRAMMAR

Production rule
Scene $\rightarrow$ *Road*
Road $\rightarrow$ *Curb Lane*
Lane $\rightarrow$ *RMAlong RMAcross RMOther*
RMAlongCW $\rightarrow$ *RMSeg*
RMAcrossCW $\rightarrow$ *RMSeg*
RMOther $\rightarrow$ *RMSeg*
Curb $\rightarrow$ *CurbSeg*

defined by the Highway Code. This kind of knowledge can be encoded as logical rules within Description Logic.

A brief description of the most important concepts is given in Tab. I. It is important to note that entities that represent road marking segments (*RMSeg*) and curb segments (*CurbSeg*) are both linked to the output of the segmentation networks described in the previous section. Hence, instances of these types are grounded in image space. This is important as it allows us to reconstruct concepts higher-up in the hierarchy (e.g. Lanes) based on those low-level segmentations. In particular, we represent detected segments using axis-aligned and minimal area bounding boxes. More high-level concepts are represented as the bounding box of their children. Note that all other concepts are assigned based on the learnt grammar.

In the next section, we explain how we learn a probabilistic grammar for road scenes based on the introduced concepts.

B. Probabilistic Grammar

We adopt the approach by [11] and learn a probabilistic context-free grammar for road scenes from a set of annotated examples. To this end, we consider a set of scene graphs that have been manually annotated according to the concepts introduced in the previous section and based on the detections of road markings and curbs (Sec. IV). We learn the structure of the production rules and their probability from the frequency observed in the annotated set. The production rules are shown in Tab. II[1]

For each annotated scene graph we compute a set of geometric properties and spatial relations between instances that share the same parent node. We start the computation at the leaf nodes and propagate the results up the hierarchy. In our implementation, we consider several geometric properties including: length, width, and area for both axis-aligned and minimal bounding boxes. Furthermore, we consider the ratios between these properties to compute scores for the *axis-alignedness*, *alongness*, and *acrossness* of an instance. We also consider spatial relations between instances that share the same parent node (e.g. two boundaries of a lane). For these instances, we compute several relations including: the connectivity of the bounding boxes based on the Region Connection Calculus [25] and their relative angle and distance based on the Ternary Point Calculus [26]. In total we consider 18 geometric properties and 14 spatial relations. However, the details of how these properties and relations are not described here for brevity. Overall, the individual features are not critically important (and can be replaced). However, they provide us with the ability to assess the overall probability of the scene by considering all instances of a tree t given its geometric description and its relations. For each geometric property and relation we learn a probability distribution, namely $P_{geo}(x)$ and $P_{rel}(x)$, based on the annotated data using Kernel Density Estimation (based on Gaussian kernels). By computing the probability of each individual property and relation we can compute the overall probability of a tree based on the grounded representation as follows:

$$P(s|t,g) = \prod_{x \in t} P_{geo}(x) P_{rel}(x) \tag{1}$$

whereby s denotes a scene, t a tree, and g a grammar.

C. Scene Parsing

To reconstruct the layout of a road scene we use an extended version of a probabilistic Earley parser [27]. In general, the Earley algorithm is a dynamic programming approach that is able to handle ambiguous grammars. It combines top-down predictions and bottom-up recognitions to effectively parse its input. The algorithm has three main steps: *predict*, *scan*, and *complete*. In the predict step, rules are expanded according to the grammar. This step guides the overall search in a top-down way (initially the root node is expanded). In the scan step, the next input symbol is read and compared to the next one that was predicted. If a production rule is completed, the complete step has found a valid parse of a subtree and overall search is advanced. This type of

[1]Note, that we have omitted the learnt probabilities as we have learn different rules for different cardinalities.

hybrid search using top-down down reasoning and bottom-up perception for scene understanding can be very effective in real-world scenarios as we have shown earlier [28].

Our adapted version of the parser takes the learnt probabilistic grammar and a sequence of curb and road marking segments as input. The segments form the lexicon of our grammar and their probabilities are determined according to $P_{geo}(X)$ as defined in the previous section.

After the parser has recognised the input, a forest of parse trees can be retrieved. In our implementation we use a shared packed parse forest (SPPF) to store the ambiguous parse trees [29]. Parse trees are evaluated according their probabilities computed as follows:

$$P(t|s,g) = P(t|g)P(s|t,g) \tag{2}$$

whereby t denotes a parse tree, s the scene, and g the grammar. $P(t|g)$ is the product of all probabilities according to the production rules and $P(s|t,g)$ represents the data likelihood of seeing this scene given the tree and the grammar. Eventually, the best parse tree t^* can be chosen according to the overall probability:.

$$t^* = \arg\max_{t \in \mathcal{T}} P(t|s,g) \tag{3}$$

whereby t denotes a parse tree in the parse forest $\mathcal{T}$, s the scene, and g the grammar.

VI. Experiments

In this section we present the experimental setup and discuss qualitative results of our approach.

A. Experimental Setup

In this work, we evaluated the overall pipeline as depicted in Fig. 2. A given input image is processed by the road marking and the curb detection networks. The output of these networks is a probability distribution of segments in the image space. Using Inverse Perspective Mapping (IPM), we transform each of the segmented images into a birds eye view (see Tab. III). For each class, we then find clusters that represent these entities by their bounding boxes and compute a set of geometric features. Based on their visual and geometric probability these segmented entities are added to the lexicon of the grammar.

The Earley algorithm predicts the structure of the scene based on the learnt grammar and parses the segments from left to right in image space. We evaluated the generated parse trees according to their probability. However, given the high ambiguity of rules in the learnt grammar, we have selected a few examples manually (Tab. III). In the next section we discuss several theses examples and point to interesting and/or problematic aspects.

B. Qualitative Results

Tab. III depicts the qualitative results for several scenes. The table shows the input image; the different segments produced by the networks and the clustering step (road markings in green; curbs in orange); and the generated scene graphs (or parts of it).

Scene (a) In this scene (see Fig. 1), the segmentation captures curbs on both sides of the road as well as road markings along the carriage way. However, a stop line as well as the bicycle symbol are not detected. By integrating some domain knowledge from the Highway Code in form of rules, we can refine the scene graph by inferring that there is a bicycle lane on the left-hand side as the lane's width is too narrow for a standard car lane. These rules are encoded within Description Logic and can infer classes which were not labelled in any of the examples. However, we are not able to infer the same on the right-hand side as we do not have any meaningful segment that describes the boundary of the bicycle lane on the right-hand side. The detection of road markings and curbs in roads other than the main road is typically more challenging as they are perceived at the edge of the camera's field of view.

Scene (b) In this scene the parser detects two road markings on a lane. Given their size and spatial relation we can infer that these entities are road markings that introduce speed humps on the road.

Scene (c) This scene is interesting as there are curb structures in the middle of the road. Furthermore, the left lane has two stop-lines. However, it is important for an autonomous vehicle to infer that is has to stop in front of the first one. Note, that such an inference can only be drawn when local context of the scene is considered, but not from the single segment alone. These are situations in which we believe that background knowledge and AI reasoning techniques can have a great impact when interpreting scenes.

Scene (d) In this scene both curbs and road markings are well detected (except for the degraded dotted line across the road). However, this scene provides an interesting and rare case as the road markings for the car (zig-zag line) and the bicycle lane overlap. Momentarily this cannot be represented by our grammar as we made the assumption that lanes are next to each other.

In future work we will also perform a quantitative analysis of our approach, in particular with respects to its real-time capabilities. In general the Early algorithm is well-suited for real-time applications as its worst time complexity is $O(n^3)$. However, retrieving and processing a potential exponential number of parse trees might be challenging.

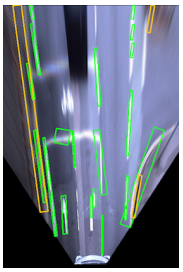
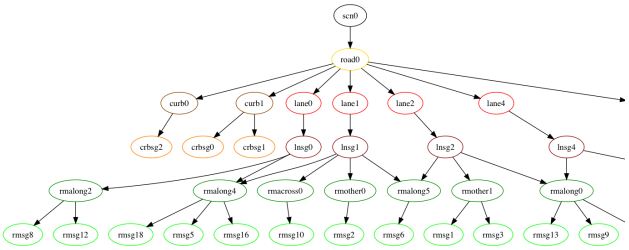
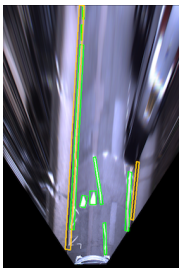
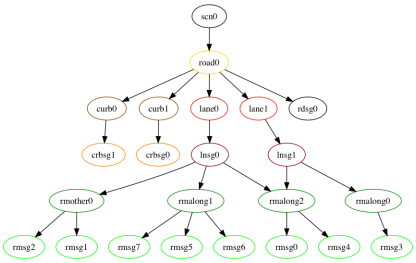
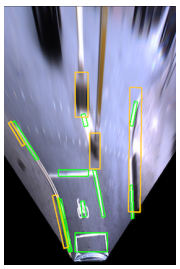
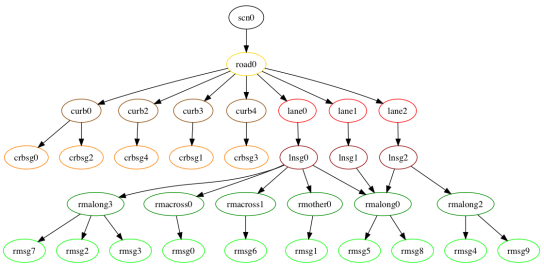
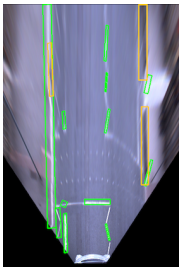
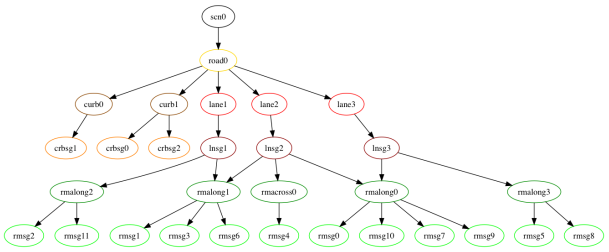
VII. Discussion

In this section we would like to provide a brief overview of the application space of the scene graph.

1) Urban traffic scenes are highly structured since they are built consistently according to specified road rules. By incorporating these rules, certain nodes in the scene graph can be classified. For instance, a bicycle lane is easily distinguished from a car lane by comparing the width. In this way, the scene graph allows for classification of road objects/segments without requiring expensive manual labels.

2) The segmented scenes given by the scene graph can be employed to bootstrap deep learning models. As stated above classification labels which can be used for training

TABLE III
QUALITATIVE RESULTS

ID	Original (RGB)	Segments (IPM)	Scene Graph (partial)
(a)			
(b)			
(c)			
(d)			

purposes can be acquired without expensive manual annotation. Furthermore, the scene graph provides an informed indication about the likely location of road objects (e.g. curbs, road markings). This could be used when training deep networks for instance to guide attention or to adjust the loss and thereby improve performance. In this way, important prior information about the environment is included in a deep learning approach (which is non-trivial).

3) Scene graphs can be used for (cost-based) planning for autonomous vehicles as they reason about the lane geometry and can infer road marking classes based on contextual spatial relations. For instance, a solid boundary of a bicycle lane should only be crossed in case of emergency. Besides, actions are now interpretable because we can review the representation inferred from the segmentation.

4) The scene graph is able to predict/hallucinate missing objects because of the learned spatial and semantic constraints. For example, two-way roads with missing lane markings in the middle will not fit the learned representations (nor the road rules). The scene graph can predict the most likely lane geometry in that case.

We think that these examples are interesting uses cases with exciting technological challenges for applications of scene graphs.

VIII. CONCLUSION

In this paper we presented an approach for scene understanding of complex urban environments. To this end, we

proposed *scene graph*, a hierarchical, graph-based representation, and a parsing pipeline that generates and evaluates scenes graphs based on partially segmented images, a learnt probabilistic grammar, as well as geometric and relational models. Furthermore, we have presented and discussed several example scenarios in which scene graphs can provide meaningful insights in the overall structure of the environment. The construction and interpretation of interpretable and auditable scene graphs can play essential role in many tasks of autonomous vehicles including planning, decision making, and explanation generation. Hence we believe that this functionality can have wide impact in the context of autonomous driving and mobile robotics in general.

ACKNOWLEDGMENT

The work has been supported by the EPSRC/UK Research and Innovation Programme Grant EP/M019918/1 (Mobile Autonomy: Enabling a Pervasive Technology of the Future).

We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.

[3] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 1012–1025, 2014.

[4] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller *et al.*, "Making bertha drive an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[5] J.-B. Bordes, F. Davoine, P. Xu, and T. Denœux, "Evidential grammars: A compositional approach for scene understanding. application to multimodal street data," *Applied Soft Computing*, vol. 61, pp. 1173–1185, 2017.

[6] F. Dierkes, M. Raaijmakers, M. T. Schmidt, M. E. Bouzouraa, U. Hofmann, and M. Maurer, "Towards a multi-hypothesis road representation for automated driving," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 2497–2504.

[7] D. Töpfer, J. Spehr, J. Effertz, and C. Stiller, "Efficient road scene understanding for intelligent vehicles using compositional hierarchical models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 441–451, 2015.

[8] J. Spehr, D. Rosebrock, D. Mossau, R. Auer, S. Brosig, and F. M. Wahl, "Hierarchical scene understanding for intelligent vehicles," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 1142–1147.

[9] S. Kashetty Venkateshkumar, M. Sridhar, and P. Ott, "Latent hierarchical part based models for road scene understanding," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.

[10] B. Mathibela, P. Newman, and I. Posner, "Reading the road: Road marking classification and interpretation," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2072–2081, 2015. [Online]. Available: http://dx.doi.org/10.1109/TITS.2015.2393715

[11] T. Liu, S. Chaudhuri, V. G. Kim, Q.-X. Huang, N. J. Mitra, and T. Funkhouser, "Creating consistent scene graphs using a probabilistic grammar," *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, vol. 33, no. 6, Dec. 2014.

[12] A. Ess, T. Mueller, H. Grabner, and L. van Gool, "Segmentation-based urban traffic scene understanding," in *Proceedings of the British Machine Conference, pages*, 2009, pp. 84–1.

[13] A. Joshi and M. R. James, "Generation of accurate lane-level maps from coarse prior maps and lidar," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 19–29, 2015.

[14] S. Schulter, M. Zhai, N. Jacobs, and M. Chandraker, "Learning to look around objects for top-view representations of outdoor scenes," *arXiv preprint arXiv:1803.10870*, 2018.

[15] 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.

[16] X. Chen, L.-J. Li, L. Fei-Fei, and A. Gupta, "Iterative visual reasoning beyond convolutions," *arXiv preprint arXiv:1803.11189*, 2018.

[17] X. Liang, H. Zhou, and E. Xing, "Dynamic-structured semantic propagation network," *arXiv preprint arXiv:1803.06067*, 2018.

[18] N. Nauata, H. Hu, G.-T. Zhou, Z. Deng, Z. Liao, and G. Mori, "Structured label inference for visual understanding," *arXiv preprint arXiv:1802.06459*, 2018.

[19] 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.

[20] T. Bruls, W. Maddern, A. A. Morye, and P. Newman, "Mark yourself: Road marking segmentation via weakly-supervised annotations from multimodal data," in *Robotics and Automation (ICRA), 2018 IEEE International Conference on*. IEEE, 2018.

[21] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498

[22] T. Suleymanov, P. Amayo, and P. Newman, "Inferring road boundaries through and despite traffic," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, November 2018.

[23] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., *The Description Logic Handbook: Theory, Implementation, and Applications*. New York, NY, USA: Cambridge University Press, 2003.

[24] M. Tenorth, L. Kunze, D. Jain, and M. Beetz, "Knowrob-map - knowledge-linked semantic object maps," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, Dec 2010, pp. 430–435.

[25] D. A. Randell, Z. Cui, and A. G. Cohn, "A spatial logic based on regions and connection," in *KR*. Morgan Kaufmann, 1992, pp. 165–176.

[26] R. Moratz and M. Ragni, "Qualitative spatial reasoning about relative point position," *Journal of Visual Languages & Computing*, vol. 19, no. 1, pp. 75–98, 2008, spatial and Image-based Information Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1045926X06000723

[27] J. Earley, "An efficient context-free parsing algorithm," *Commun. ACM*, vol. 13, no. 2, pp. 94–102, Feb. 1970. [Online]. Available: http://doi.acm.org/10.1145/362007.362035

[28] L. Kunze, C. Burbridge, M. Alberti, A. Tippur, J. Folkesson, P. Jensfelt, and N. Hawes, "Combining top-down spatial reasoning and bottom-up object class recognition for scene understanding," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, Illinois, US, September, 14–18 2014.

[29] E. Scott, "Sppf-style parsing from earley recognisers," *Electronic Notes in Theoretical Computer Science*, vol. 203, no. 2, pp. 53 – 67, 2008, proceedings of the Seventh Workshop on Language Descriptions, Tools, and Applications (LDTA 2007). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1571066108001497

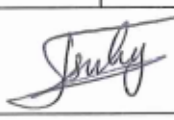
Statement of Authorship for joint/multi-authored papers for PGR thesis

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	Reading between the Lanes: Road Layout Reconstruction from Partially Segmented Scenes
Publication Status	☑ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	L. Kunze, T. Bruls, T. Suleymanov, and P. Newman, "Reading between the Lanes: Road Layout Reconstruction from Partially Segmented Scenes," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Maui, Hawaii, USA, 2018.

Student Confirmation

Student Name:	Tarlan Suleymanov		
Contribution to the Paper	My contributions to the paper were: Contributed to the refinement of the initial idea behind the paper. Dataset annotation and preparation for curb detection. Running the curb detection experiments. Writing the curb detection section. Designing and making the figures. The overall paper was done in discussion and collaboration with Lars Kunze, Tom Bruls and Paul Newman.		
Signature		Date	12 September 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: Prof Paul Newman			
Supervisor comments I AGREE			
Signature		Date	12/9/2019

This completed form should be included in the thesis, at the end of the relevant chapter.

5.12 Summary of the Paper's Results

In this paper we presented an approach to scene understanding using a hierarchical graph-based representation for complex urban scenarios. The proposed approach included a parsing pipeline for generating scene graphs based on partially segmented images which were obtained from deep learning networks. Besides segmented images, we used prior knowledge and learnt a probabilistic grammar to reconstruct road layouts and enable high-level reasoning that can be applied to planning, navigation, and decision making in the context of autonomous driving. The proposed approach also demonstrated that the road boundaries detected via the camera-based VRBD model provided important information for scene understanding. Detected road boundaries and road markings were linked to the entities that represented road boundary/curb segments and road marking segments, respectively. Hence, instances of these segments were grounded in image space, allowing us to reconstruct scene graphs with concepts higher-up in the hierarchy.

Based on the experimental results presented in the paper we observe that the representation of a road layout using scene graphs is beneficial for classification of road segments without the need for time-consuming, manual labelling. Moreover, scene graphs can be used to obtain classification labels for deep learning models without expensive manual annotation. Scene graphs provide information about likely location of road makings, lanes, and curbs that can be employed to bootstrap deep learning models and improve their performance. Additionally, scene graphs can be adopted for planning and decision making as they provide information based on contextual information. Finally, using the learnt grammar and domain knowledge, scene graphs can hallucinate missing objects in the scene.

As part of our future work we are interested in using inferred, occluded road boundaries for constructing scene graphs and evaluating how this changes the performance. In the presence of occlusions of road boundaries, inferring occluded road boundaries will help to generate a complete graph of the scene. Additionally, we will integrate LiDAR-based road boundary detection, which is presented in Chapter 6, into the pipeline to provide outputs in a bird's-eye view format and

compare camera and LiDAR-based approaches, evaluating the performance gain available from a combination of these two modalities.

5.13 Conclusions

In this chapter we presented a deep learning based road boundary detection approach that inferred road boundaries irrespective of whether or not the boundaries are actually visible. The quantitative and qualitative experiment results demonstrated that our approach could tackle even very challenging scenarios. Our ORBI model captured contextual information for estimating the correct position and structure of occluded road boundaries. After carefully examining different types of scenarios, we decided to explore another modality for road boundary detection. Although the camera-based approach failed in certain scenarios (such as night-time), those cases could be fixed by (1) generating more training data from different scenarios and/or (2) using systems (as a pre-processing step) that could transform weather/time conditions of images (which we will do in future work). In the next chapter we address the road boundary detection problem based on LiDAR data.

6

Online Inference and Detection of Curbs in Partially Occluded Scenes with Sparse LiDAR

Contents

6.1 Introduction

In Chapter 5, we presented a deep learning-based road boundary inference and detection method using camera images. Our camera-based VRBD and ORBI models achieved 0.9477 and 0.9197 F1 scores, respectively. However, as we discussed in Section 2.2, camera and LiDAR present distinct and opposing advantages and disadvantages in certain conditions. Relying on one sensor can be risky, and having an alternative "opinion" from another source is very helpful in case of failure of the first modality. In this chapter, we present an alternative method for road boundary detection that uses LiDAR as an input modality. Having achieved more than 0.9 F1 score for the camera-based models, we decided to adopt the camera-based road boundary detection approach for the LiDAR-based approach.

Table 6.1: Datasets that were used for the training and testing of LiDAR-based VRBD and ORBI models.

Dataset	Modality		No. of frames	Size	Area	Used for
	2D LiDAR	3D LiDAR				
Oxford RobotCar 24-08-18	No	Yes	1.8K	480×960	48×96	Train./Test.
Oxford RobotCar 30-04-18	Yes	Yes	17K	480×480	48×48 and 24×24	Training
Oxford RobotCar 18-01-19	Yes	Yes	2K	480×480	48×48 and 24×24	Testing

The framework for obtaining the ground truth data for training the LiDAR-based VRBD and ORBI models is presented in Section 4.3.3, where we described how 2D or 3D LiDAR-based point clouds were annotated and then projected into 2D bird's-eye view images (IPM) to generate input images and their road boundary masks. We used three datasets (24-08-18, 30-04-18 and 18-01-19) from the Oxford RobotCar Dataset to train and test our models. The 24-08-18 dataset did not contain 2D LiDAR and was annotated using 3D LiDAR data (see Table 6.1). A total of 1800 IPM samples were generated from this dataset and were used for the training and testing of the models presented in our ITSC 2019 paper [13] (Section 6.5). The generated IPM images were of 480×960 pixels resolution and covered a 48×96 squared metre area (see Figure 6.1 for an example). They were obtained by projecting 5 consecutive scans from the Velodyne VLP-32C 3D LiDAR into IPM images.

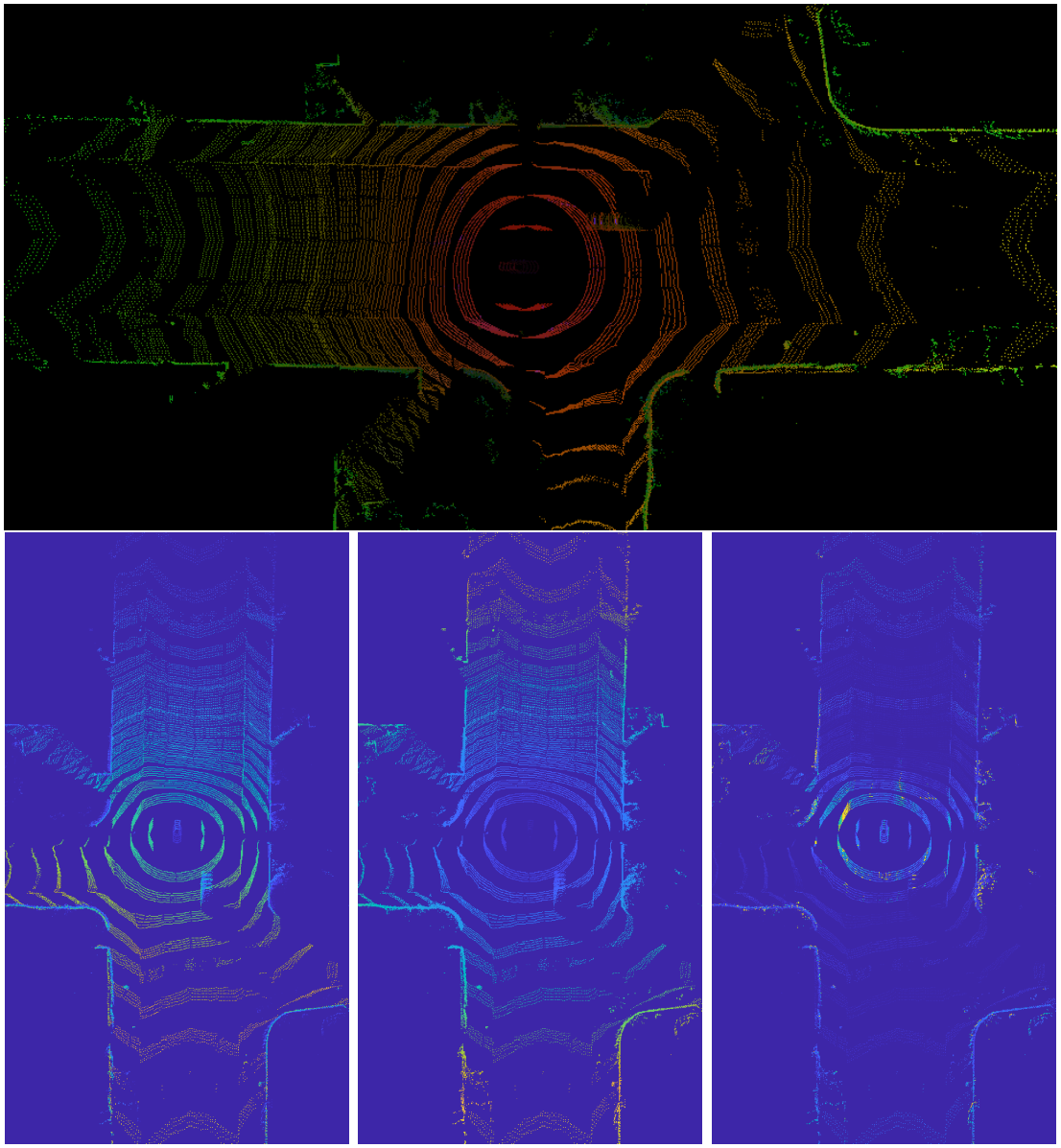


Figure 6.1: A training sample from the 24-08-18 dataset. An input IPM image (above) which consists of three channels: height (left), range (middle) and intensity (right). Note that the channel images are displayed with scaled colours for better visualisation.

The other two datasets (30-04-18 and 18-01-19) contained 2D LiDAR data that was annotated for generating road boundary masks. The 30-04-18 dataset was half annotated and used to generate 17K samples for training the models. The 18-01-19 contained 2K samples and was used for testing the models (see Table 6.1). Note that these datasets had two Velodyne HDL-32E 3D LiDARs, and IPM images were generated for both of them. Additionally, the IPM images were generated at two scales covering 48×48 or 24×24 squared metre area but had the same

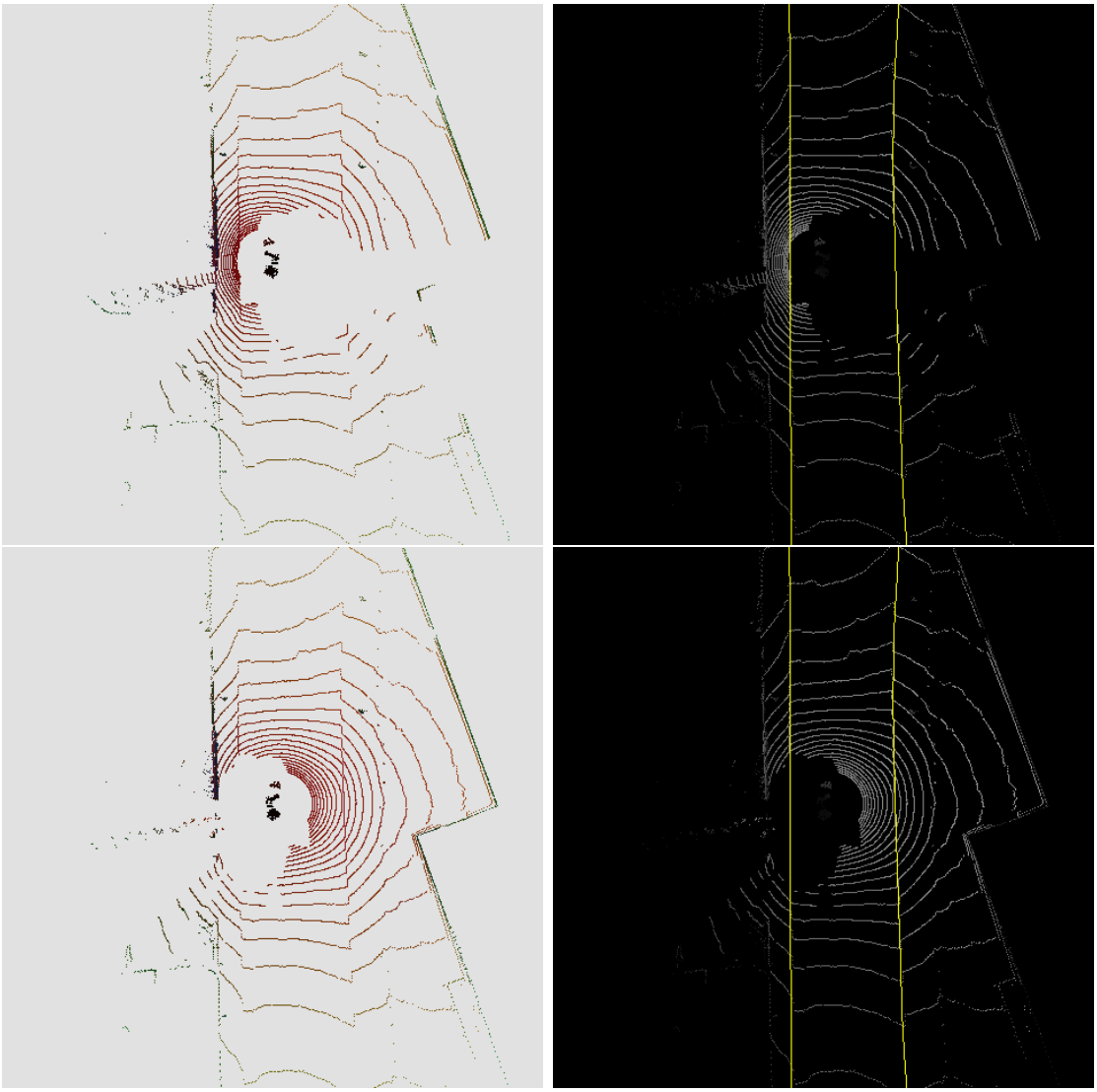


Figure 6.2: A training sample with 48×48 squared metre ROI generated from the pair of Velodyne HDL-32E 3D LiDARs. Top row: left LiDAR IPM. Bottom row: right LiDAR IPM.

resolution of 480×480 pixels. See Figure 6.2 and Figure 6.3 for examples of the 48×48 and 24×24 images, respectively.

6.2 Comparison of Datasets

Before moving on to describe the LiDAR-based road boundary detection approach in detail, we make a comparison between single LiDAR and dual LiDAR datasets. The single Velodyne VLP-32C 3D LiDAR-based data had sparse point clouds around the vehicle even within 24 metres range, but the laser beams could reach to 48

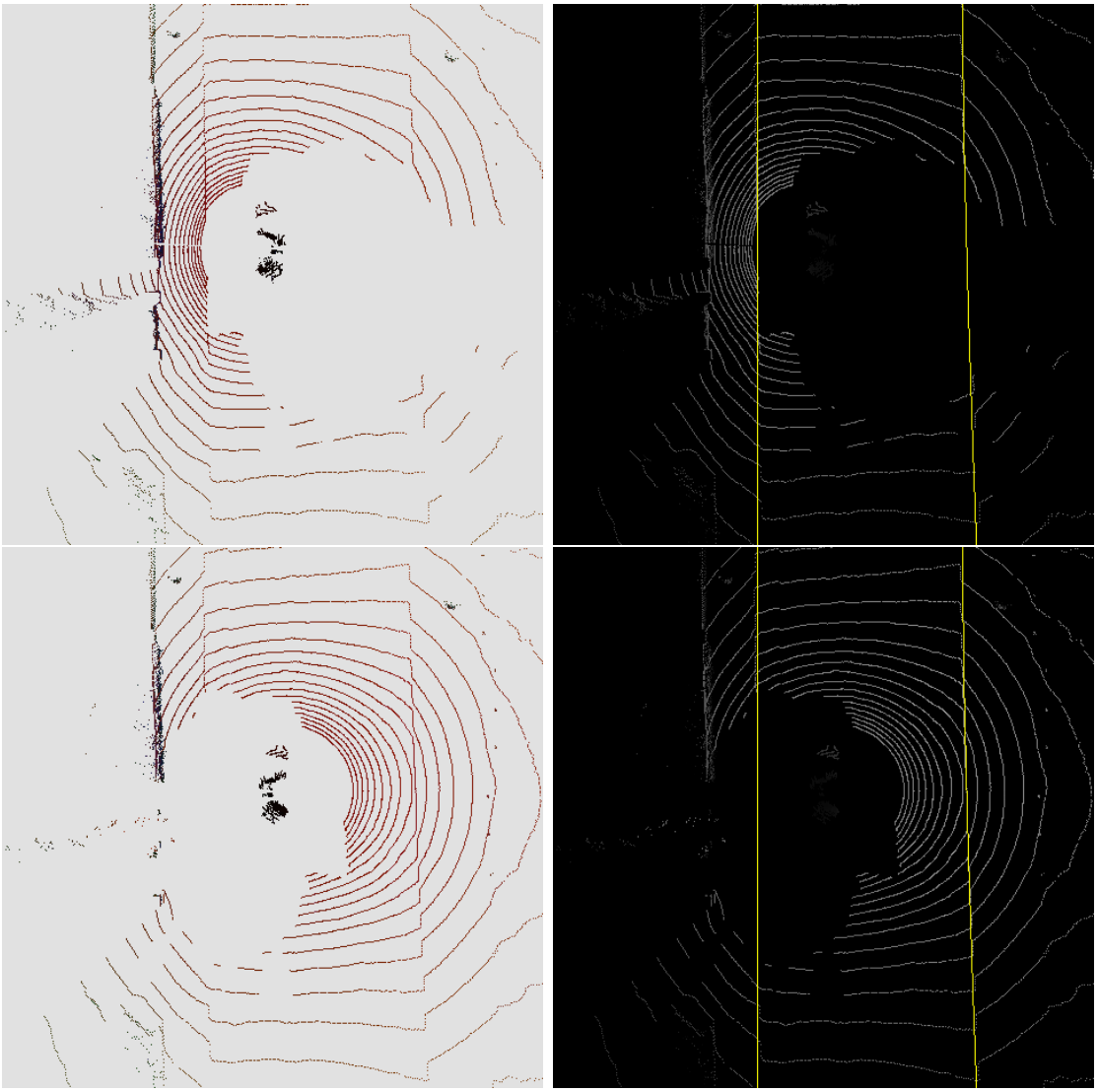


Figure 6.3: A training sample with a 24×24 squared metre ROI generated from the pair of Velodyne HDL-32E 3D LiDARs. Top row: left LiDAR IPM. Bottom row: right LiDAR IPM.

metres range and could capture the shape and structure of the roads, as can be seen in Figure 6.4 (middle). In contrast, the point clouds that were generated based on pair of Velodyne HDL-32E 3D LiDARs were dense within the range of 24 metres but could not capture the shape or structure of roads beyond that. For our initial attempt, we used the datatset with VLP-32C LiDAR in order to exploit the longer range, and we integrated five consecutive scans when projecting point clouds into IPM images in order to compensate for the sparsity of the point clouds. The IPM images in our first attempt had a 480×960 pixels resolution covering a

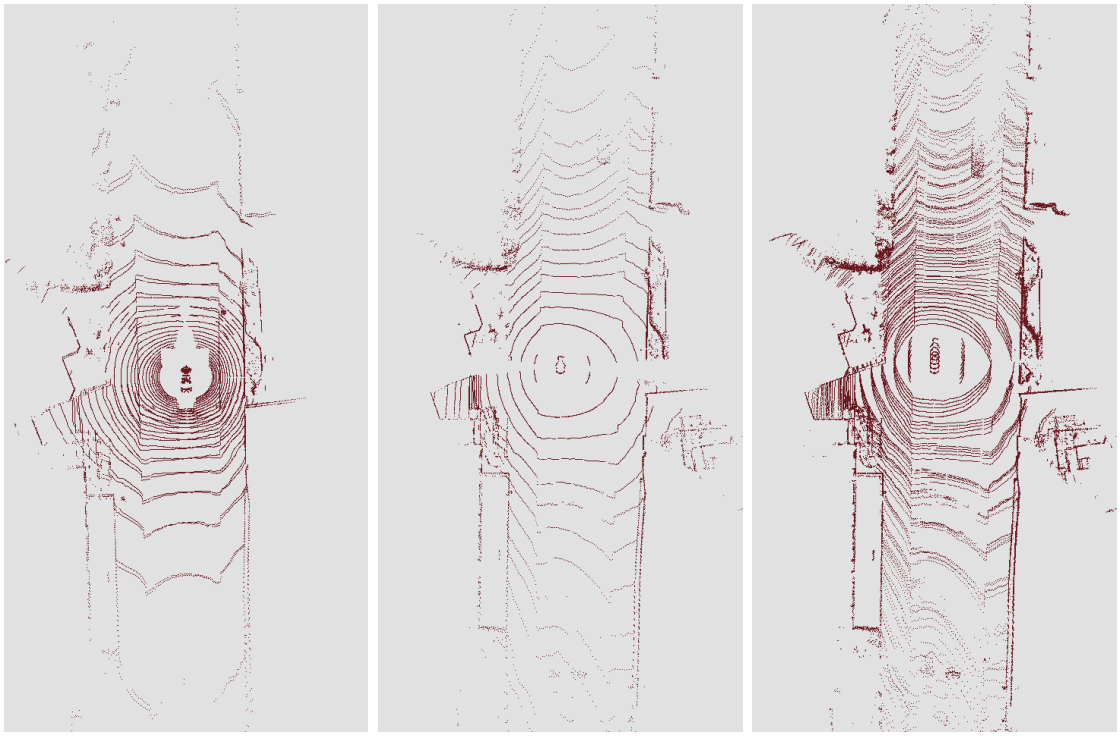


Figure 6.4: Comparison of laser scans of different LiDAR sensors: combined scans from a pair of Velodyne HDL-32E 3D LiDARs (left), single scan (middle) and five consecutive scans (right) of Velodyne VLP-32C 3D LiDAR. All three IPMs cover 48×96 metres area.

48×96 squared metre area. However, the lack of a 2D LiDAR sensor in this dataset (24-08-18) made it more difficult to annotate the point clouds. Approximately 750 samples could be obtained with one hour of annotation of 2D LiDAR-based point clouds, but obtaining the same quantity of training samples by annotating 3D LiDAR-based point clouds took more than three hours. As a result, we generated only 1800 samples for the training and testing of our first approach, which was presented in the ITSC 2019 paper (Section 6.5). For our second attempt, we used datasets that included 2D LiDARs that were easier to annotate, which allowed us to obtain 19K samples for training and testing (Table 6.1). However, as those datasets (30-04-18 and 18-01-19) had a pair of Velodyne HDL-32E 3D LiDARs, we reduced the resolution of the IPM images to 480×480 pixels due to the range of the sensors. Additionally, integrating consecutive scans was not required as the scans were sufficiently dense within a 24 metres range.

6.3 Partitioning Training Data

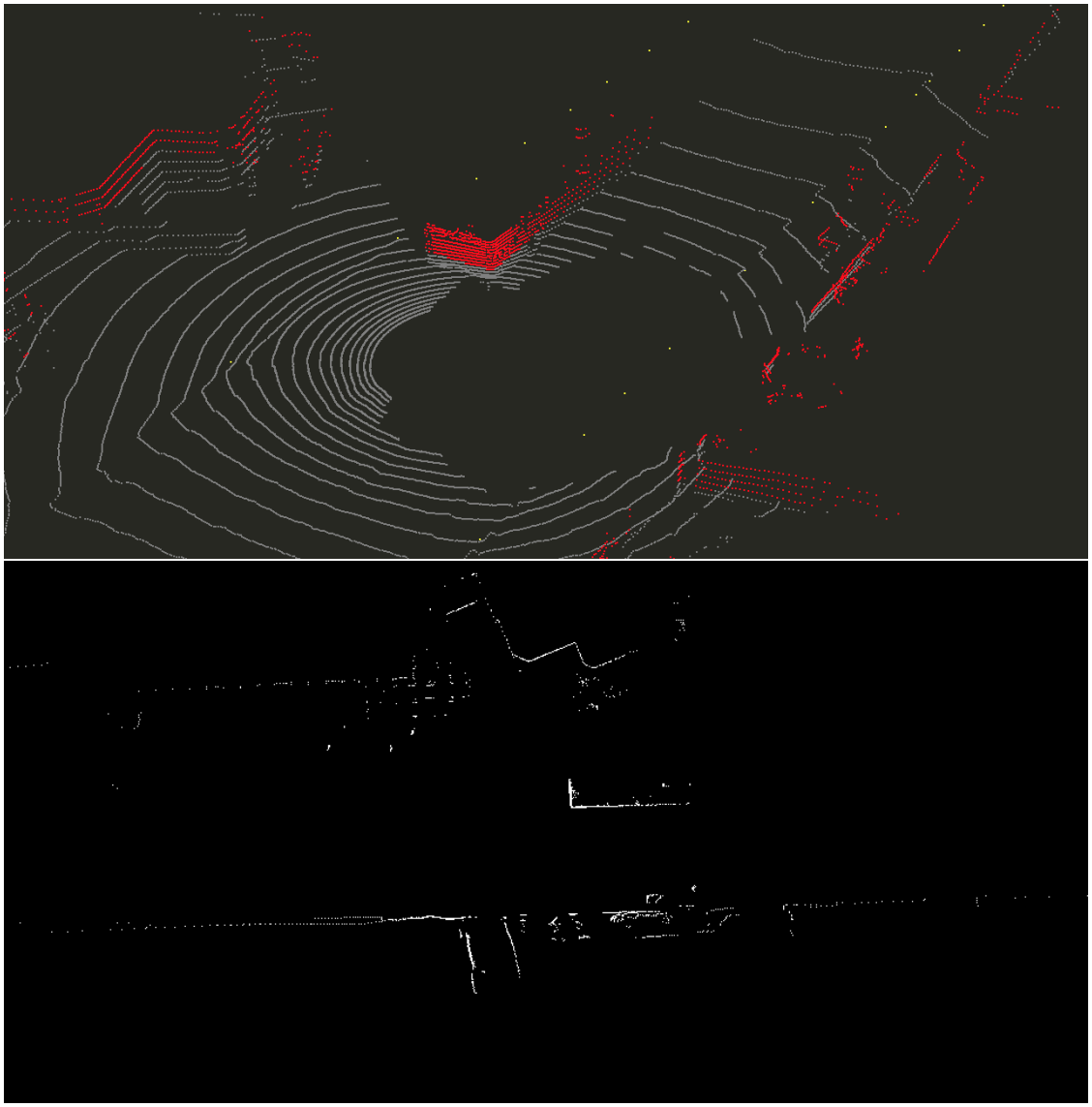


Figure 6.5: Top: A point cloud of a LiDAR scan where the points that are within the predefined height difference from the sensor are coloured in red. Bottom: IPM mask generated from the above red-coloured points representing occluding obstacles.

In Chapter 5, we approached the road boundary detection problem as a coupled, two-class detection problem, detecting visible road boundaries and inferring occluded ones with a continuity constraint. The camera-based road boundary training data was partitioned into two classes to solve the problem with two sub-tasks. Adopting the same approach for the LiDAR-based road boundary detection required partitioning the raw training data, but instead of using the U-net-based approach

for partitioning (Section 5.2) we used the 3D information of the points provided by LiDAR. Obstacles that are above the ground surface potentially prevent the laser beams from reaching the road boundaries and create occlusions. To obtain masks of potentially occluding obstacles, LiDAR scans were processed such that the points that were within the predefined height difference from the LiDARs were selected. As can be seen in Figure 6.5, the red-coloured points, which included points on a bus, appeared above the ground surface and created occlusions. The generated masks of occluding obstacles (Figure 6.5, bottom) were used to identify which sections of the raw labels were not visible from the sensor using a hidden point removal algorithm presented in [30]. This algorithm determines which points are visible from a given viewpoint by extracting points residing on the convex hull. As a result, the raw road boundary masks were partitioned into two classes, see Figure 6.6 for examples.

Partitioning the raw training data of the 24-08-18 dataset consisted of applying only one step of the hidden point removal algorithm as the IPMs were generated from one LiDAR sensor. However, the other two datasets (30-04-18 and 18-01-19) had two 3D LiDARs and consisted of applying (twice) the hidden point removal algorithm to obtain masks of road boundaries for each LiDAR followed by a final step which combined those masks. Each LiDAR had slightly different viewpoints, and the sections of visible/occluded road boundaries had slight differences. The combined masks of visible road boundaries were obtained by applying the logical AND operation between the visible masks of the two LiDARs. To obtain the combined masks of occluded road boundaries the logical OR operation was applied between the pair of occluded masks. Having partitioned the training data, the VRBD and ORBI models were adopted from the camera-based approach to fit the requirements of LiDAR-based IPMs. We present the models adopted for the single Velodyne VLP-32C 3D LiDAR in the following section and the models for the pair of Velodyne HDL-32E 3D LiDARs in Section 6.8.

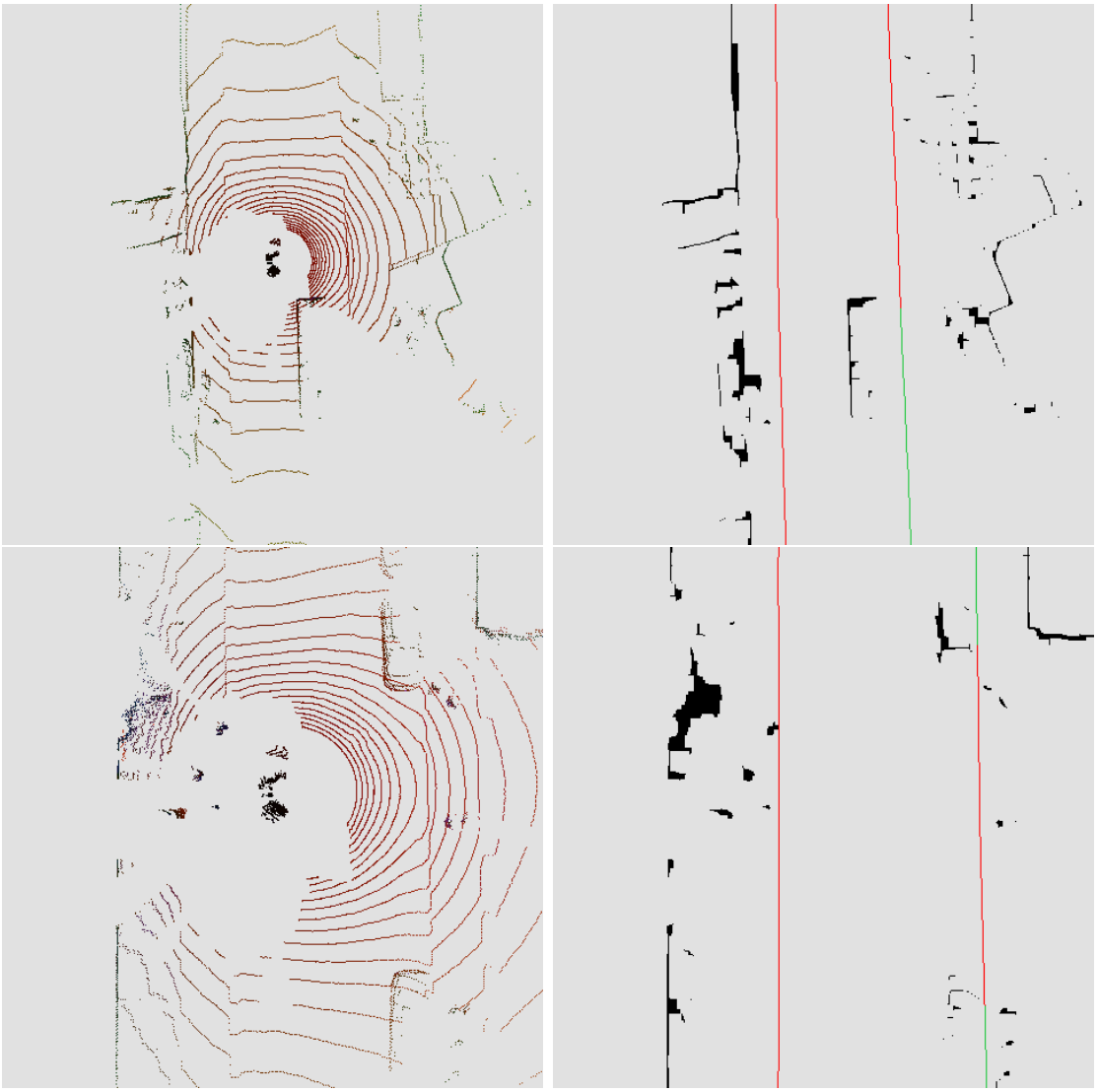


Figure 6.6: Partitioned training data examples. IPM images (left column) and their corresponding obstacle masks with partitioned road boundary labels overlaid (right column), where red-coloured labels are visible road boundaries and green labels are occluded.

6.4 Single VLP-32C LiDAR-based models

We adopted the VRBD and ORBI models from our camera-based road boundary detection approach to fit the size of the input IPM images. The VRBD model takes the 480×960 pixels, 3-channel IPM images as inputs in order to detect visible road boundaries. The two models are linked by the continuity constraint. Masks of detected road boundaries are fed to the ORBI model together with the IPM images. The ORBI model for the LiDAR approach had a similar architecture to

the camera approach and consisted of convolutional layers, intra-layer convolutions for capturing contextual information and parameterised multi-scale predictions for forcing the network to output long thin lines. A total of 1.8K samples were generated from the 24-08-18 dataset, 1.3K of which were used for training the models and the reminder for testing. Experimental results and the accuracy of the models are given in our paper presented in the following section. Note that a post-processing step was applied to the output of the models to track the detected and inferred road boundaries and fill in the gaps of missing detections.

The nature of the system as described here can be explored in more detail in the paper which follows. The paper presents the proposed LiDAR-based road boundary detection approach in more detail with an additional post-processing step and experimental results. The results of the initially conceived system were improved, and the limits of the system's performance were further investigated in the discussion which concludes this chapter, which also guides the narrative of the thesis in subsequent chapters.

Online Inference and Detection of Curbs in Partially Occluded Scenes with Sparse LIDAR

Tarlan Suleymanov Lars Kunze Paul Newman

Abstract— **Road boundaries, or curbs, provide autonomous vehicles with essential information when interpreting road scenes and generating behaviour plans. Although curbs convey important information, they are difficult to detect in complex urban environments (in particular in comparison to other elements of the road such as traffic signs and road markings). These difficulties arise from occlusions by other traffic participants as well as changing lighting and/or weather conditions. Moreover, road boundaries have various shapes, colours and structures while motion planning algorithms require accurate and precise metric information in real-time to generate their plans.**

In this paper, we present a real-time LIDAR-based approach for accurate curb detection around the vehicle (360 degree). Our approach deals with both occlusions from traffic and changing environmental conditions. To this end, we project 3D LIDAR pointcloud data into 2D bird's-eye view images (akin to Inverse Perspective Mapping). These images are then processed by trained deep networks to infer both visible and occluded road boundaries. Finally, a post-processing step filters detected curb segments and tracks them over time. Experimental results demonstrate the effectiveness of the proposed approach on real-world driving data. Hence, we believe that our LIDAR-based approach provides an efficient and effective way to detect visible and occluded curbs around the vehicles in challenging driving scenarios.

I. Introduction

Autonomous vehicles are required to detect road boundaries (or curbs) for understanding their surroundings [1] and for generating behaviour plans. Road boundaries separate drivable road areas from non-drivable areas in the environment. Knowing the boundaries of a road is paramount for many applications, such as autonomous parking, navigation, mapping and path planning. Road boundaries can be used for lateral guidance as part of Advanced Driver Assistant Systems (ADAS), e.g. during parking. However, road boundaries have various shapes, colours and structures, which makes road boundary detection a challenging task. Moreover, road boundaries are often occluded by other traffic participants which makes their detection even more challenging.

In this work, our goal is to address the problem of road boundary detection using methods of deep learning. Recent advances in deep learning have shown that neural networks can be used for various tasks in robotics, such as segmentation and object detection, which play important roles in solving challenging tasks for operation of self-driving cars. Here we use deep networks to detect visible and infer occluded road boundaries around the vehicle (360°) from bird's-eye view images that are generated from 3D LIDAR pointclouds.

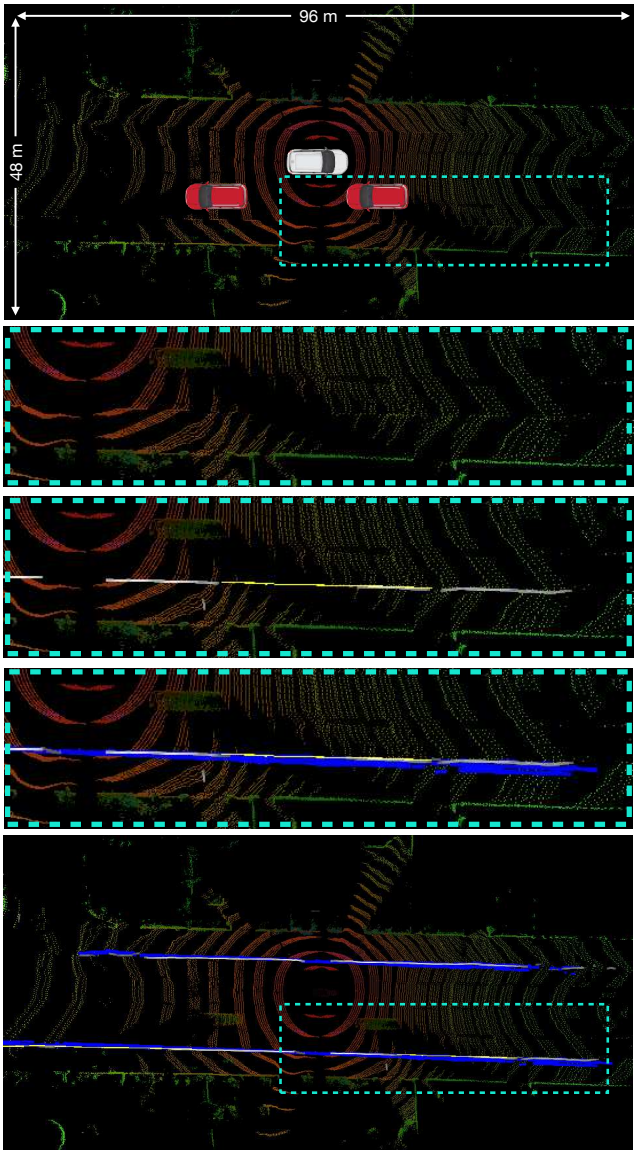


Fig. 1: Our 360° LIDAR-based curb detection approach. First, LIDAR data from the ego-vehicle (white) is transformed in bird's-eye view images which are then processed by trained deep networks to detect visible (white) and occluded (yellow) curbs. Finally, post-processing steps filters out outliers and tracks curbs over time (blue). The result is a robust curb detection around the vehicle over a total distance of 96 metres.

To this end, the paper makes the following contributions:

- a framework to annotate 3D pointclouds and transform

Authors are from the Oxford Robotics Institute (ORI), Dept. of Engineering Science, University of Oxford, Oxford, UK. {tarlan,lars,pnewman}@robots.ox.ac.uk

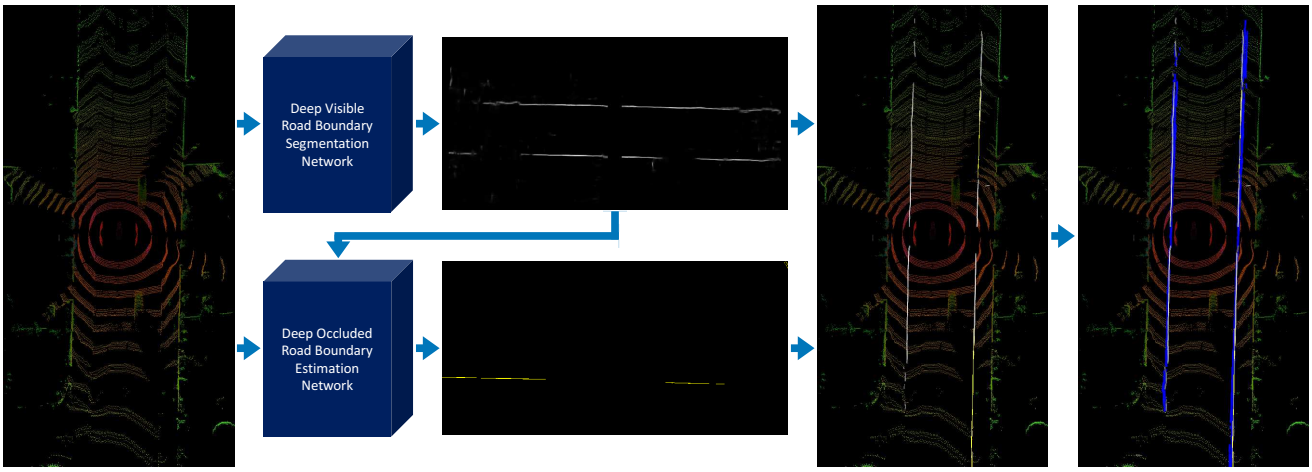


Fig. 2: Our 360° LIDAR-based curb detection approach. A pre-processing step integrates several subsequent laser scans into a coherent coordinate frame and projects them into a bird's-eye view image with height information (left). This image is then processed by two deep segmentation networks to detect both visible and occluded road boundaries (middle). Note that the network responsible for occluded boundaries additionally considers the output of the network for visible boundaries as its input. Finally, the output of both networks is combined and tracked in order to improve the overall performance over a series of processed images (right).

them into a set of bird's-eye view images and *raw* road boundary masks;
- an automatic method for partitioning/labelling *raw* training data into two classes (visible and occluded);
- a 360°, real-time deep learning method to detect visible and occluded road boundaries from bird's-eye images without making any assumptions about structure and shape of the road boundaries; and
- an experimental evaluation which provides qualitative and quantitative results of our approach on real-world LIDAR data acquired while driving through Oxford.

This remainder of the paper is structured as follows. In Section II, we discuss related work on road boundary detection. An overview of our approach is given in Section III. In Section IV, we first explain how we obtained a data set for training our deep segmentation/detection networks before we describe our proposed approach in more detail in Section V. Finally, we provide a thorough qualitative and quantitative evaluation in Section VI before we conclude in Section VII.

II. Related Work

Work on road boundary detection can be divided into two categories: *camera-based* and *LIDAR-based methods*.

Most camera-based methods address the problem using stereo cameras and 3D geometry to identify road boundaries in the scene [2], [3], [4], [5], [6], [7], [8]. In contrast, our previous work [9] used only a single monocular camera and deep convolutional neural networks for image processing to detect visible and occluded road boundaries. In this paper, we follow a similar machine-learning approach to infer both visible and occluded road boundaries, but we use LIDAR data as input. We do this without making any assumptions on the structure or shape of the curbs.

LIDAR-based methods often rely on more traditional information engineering techniques. In [10], a ring compression analysis on dense 3D LIDAR data followed by false-positive filters was used to detect curb points. Curb models are estimated using Least Trimmed Squares (LTS) and describe the road shape on occluded curbs. However, the approach mostly considers simple examples where curbs are on both sides of the road. Hence, this method would likely fail in more complex scenarios, such as intersections and/or roads with fully occluded curbs. Work by [11] uses range and intensity information from 3D LIDAR to detect visible curbs on elevation data, which fails in the presence of occluding obstacles. Similarly, [12] presents a LIDAR-based method to detect visible curbs using sliding-beam segmentation followed by segment-specific curb detection, but fails to detect curbs behind obstacles.

Our proposed approach combines advantages from both research directions camera-based and LIDAR-based methods. In particular, we obtain highly accurate, 3D data about the world from LIDAR sensors which we process in real-time using deep convolutional neural networks (CNNs). 3D LIDAR data allows us to have a larger view angle than a single camera (here 360°). At the same time, LIDAR data is not restricted by lighting or weather conditions, which allows us to operate the system under various environmental conditions (e.g. rain and fog). However, LIDAR-based methods are very data intensive. To circumvent long processing times, we project 3D LIDAR data (i.e. pointclouds) into 2D bird's-eye view images which we then process using CNNs, similar to [13], [14]. Thereby the network robustly detects road boundaries around the vehicle (360°) in urban traffic scenarios under different weather conditions.

III. LIDAR-based Curb Detection: An Overview

In this work, we detect and infer visible and occluded road boundaries based on sparse LIDAR data using deep learning methods.

LIDAR data is used for both generating training examples and inferring road boundaries. To this end, we propose a novel framework to semi-automatically generate and boost the training data (Section IV). Moreover, we use 3D LIDAR

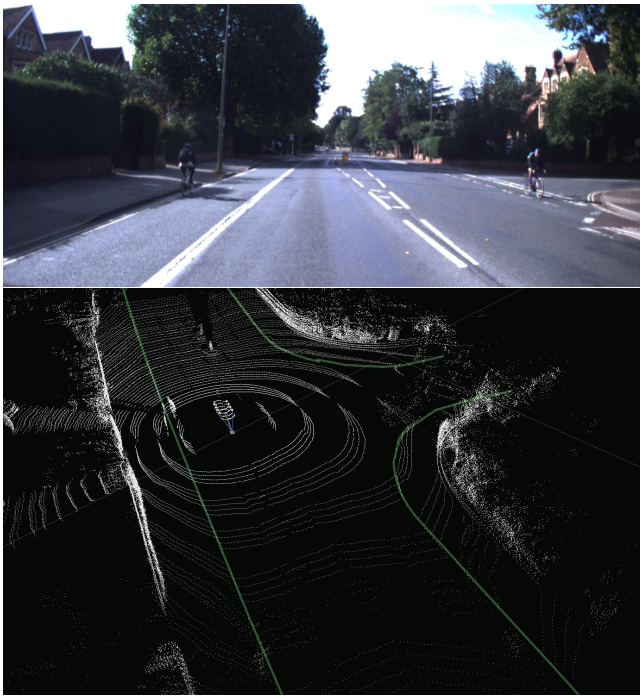


Fig. 3: Integrated 3D pointcloud data acquired by driving through Oxford and manually annotated with road boundaries.

data to detect and infer road boundaries around the vehicle (Sections V).

An overview of our 360° curb detection approach is given in Figure 2. In a pre-processing step, a sequence of LIDAR scans are integrated and projected into a bird's-eye view image. Using this image, a first network detects visible road boundaries with a fully convolutional neural network and passes the output to a second network. The purpose of the second network is to infer road boundaries given both the original bird's-eye view image and a mask of detected visible road boundaries. The generated output represents segments of road boundaries in a hybrid, discrete-continuous form. To improve the overall performance, a post-processing step filters out noise by consolidating detections in subsequent images and by tracking detected curbs over time.

IV. Obtaining Training Data and Ground Truth

Training deep networks to obtain high-performance and well generalised models requires large datasets with ground truth labels. Moreover, the variability of road boundaries in shape and structure requires training data to include samples from different environments. Although obtaining *raw* data is relatively simple, fine-grained annotation of data requires human interaction and can be extremely time consuming. For example, outlined distinct regions in an image must be associated with corresponding class labels. To obtain ground truth lables we annotated road boundaries in 3D integrated pointcloud data, which was collected by a test car fitted with a 3D LIDAR (Velodyne VLP-32C) on the roof (Figure 3). To register and integrate a set of subsequent LIDAR scans in a coherent coordinate frame, we estimate the vehicle's motion using Visual Odometry (VO) [15]. In this work, we used images acquired by a Point Grey Bumblebee XB3

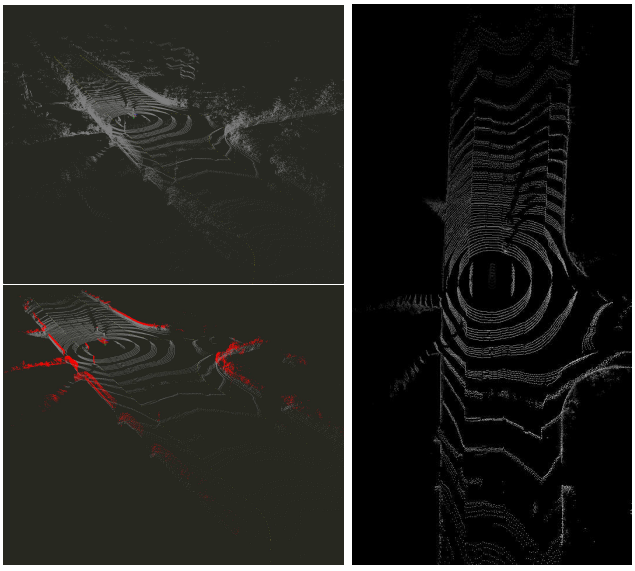


Fig. 4: Top left: integrated LIDAR pointcloud. Bottom left: Filtered and trimmed pointcloud. Right: Projected bird's-eye view image.

camera, mounted on the front of the platform facing towards the direction of motion. In particular, our implementation of VO uses FAST corners [16] combined with BRIEF descriptors [17], RANSAC [18] for outlier rejection, and nonlinear least-squares refinement.

Based on integrated 3D pointclouds we then generate 2D projections which are transformed into bird's-eye view (IPM) images. To do this, we take a laser scan and remove all points above the LIDAR device. As the LIDAR is mounted on top of the car, points above the LIDAR cannot be part of road boundaries. Similarly, we remove points that are below the LIDAR by more than 3.55 metres because some points may appear below the road surface as reflections. Also, we only keep points if they are located within 24 metres away from the car in x direction and within 48 metres in z direction. Thus we obtain trimmed pointclouds that we transform into bird's-eye view images (Figure 4). In similar way, we obtain image masks by projecting annotations of road boundaries. Note that input bird's-eye view images consist of three channels, range, intensity and height. In this work, we used LIDAR data from the OxfordRobotcar dataset [19] to obtain a new data set of bird's-eye view images and semi-annotated (labelled) road boundary masks.

A. Partitioning Training Data: Visible and Occluded Curbs

To detect all road boundaries in a given scene we split the problem into two tasks: first we detect road boundaries that are visible from the laser and then infer road boundaries that are occluded by other road users. To achieve this, we partition the training data into visible and occluded classes.

However, note that the obtained masks contain both visible and occluded road boundaries as a single class as they were generated by projecting 3D annotations into bird's-eye view images. To determine which points are visible and which are occluded we use the hidden point removal operator as described in [20]. The operator determines all visible points in a pointcloud when observed from a given viewpoint. This

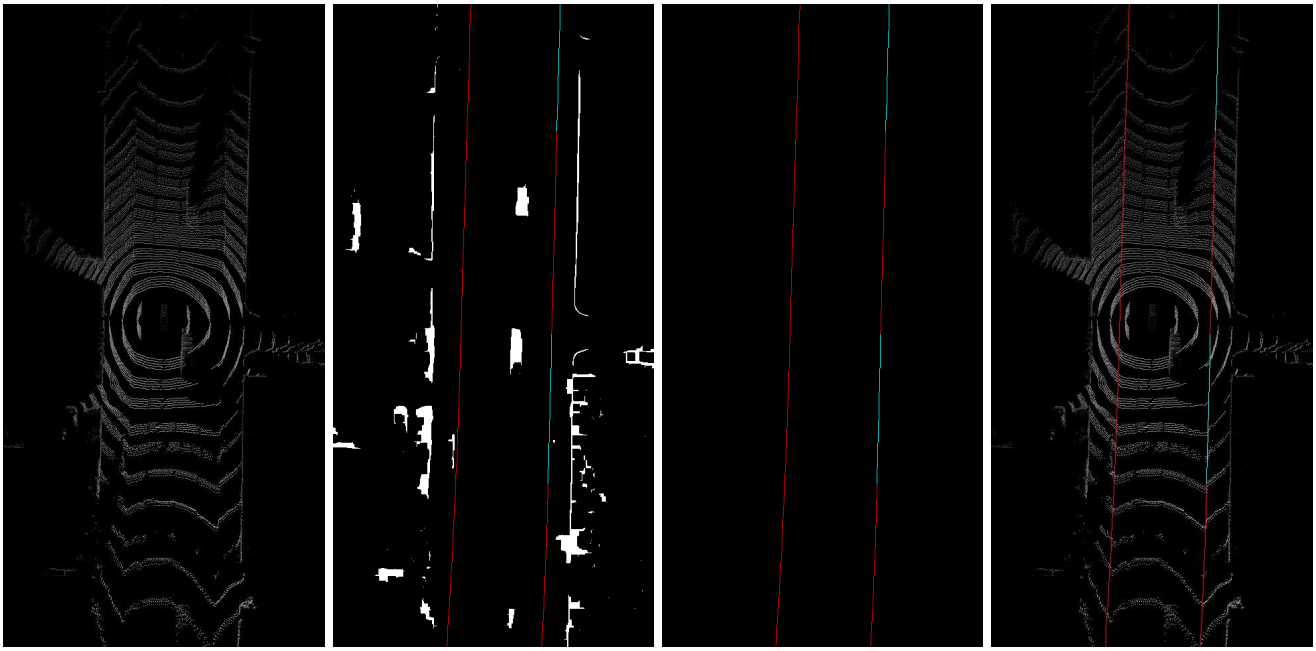


Fig. 5: Partitioning of training data. From left to right: bird's-eye view image, detected obstacles as well as visible and occluded curbs, curbs labels only, bird's-eye view image with labels.

is achieved by extracting all points residing on the convex hull of a transformed pointcloud. These points resemble the visible points, all other (labeled) points are considered as *hidden* (or occluded). We take the previously trimmed pointclouds and create binary bird's-eye view images by taking the height of points from the ground into account. The points that are within a predefined height difference from the LIDAR roughly correspond to the points (obstacles) that are blocking the view. By putting together raw labels and binary masks of obstacles, obtained by running the hidden point removal algorithm, we obtain separate masks for visible and occluded road boundaries (Figure 5).

V. Our approach

Separating the training data into two classes (visible and occluded curbs) allows us to address the problem in two steps. Note, however, that these steps are linked (see Figure 2). In a first step we detect only visible curbs. These detected curbs then provide additional information for the second step, the detection of occluded curbs. In the following we briefly describe the architectures used for both steps.

A. Detecting visible curbs

To detect visible curbs we use the U-net architecture [21]. U-net is a fully convolutional network which concatenates higher resolution "input-side" features from convolution layers with up-sampled outputs form deconvolution layers. This enables the network to detect and segment objects such as curbs more precisely. Although this approach has been successful for visible curbs, it did not generate the desired outcome for occluded curbs. The reasons for this are twofold: first, the network's limited receptive field, which is not big enough to capture context around large obstacles to estimate the position of curbs behind them, and second, the lack of structure (model-free) which prevents the network to infer very thin curves of occluded road boundaries within an image. Hence, we have employed a different approach for the detection of occluded curbs which we explain in the next section.

B. Inferring occluded curbs

In this section, we explain our approach on inferring curbs in partially occluded scenes.

Our model consists of several convolutional layers that produce an output of detected curbs as discrete lines at multiple scales. Instead of pixels, a trained network estimates parameters of lines that correspond to cells in a grid (at different scales). It discretises the output space of lines into a set of default (anchor) lines over different orientation angles. At inference time, the network then generates probabilities for the presence of occluded curbs for each anchor line and its orientation. Making predictions of occluded curbs at multiple scales is important due to the different sizes and shapes of occluding obstacles.

In this work, we have selected three scales of parameterised outputs due to run-time constraints and a given accuracy target. Pixel-wise curb labels are converted to parameterised labels by dividing curb masks (at each scale) into a grid. Lines in each grid cell are parameterised in a discrete-continuous form: first, fitted lines are assigned to one of four types of anchor lines, and secondly, offsets between fitted and anchor lines are calculated. Anchor lines pass through the centre of a grid cell at different angles (22.5°, 67.5°, 112.5° and 157.5°). During fitting, lines are assigned to the closest anchor line. Once a fitted line is discretised, two continuous parameters are calculated: (1) an angle offset between a fitted and the respective anchor line ($\omega^k_{i,j,gt}$), and

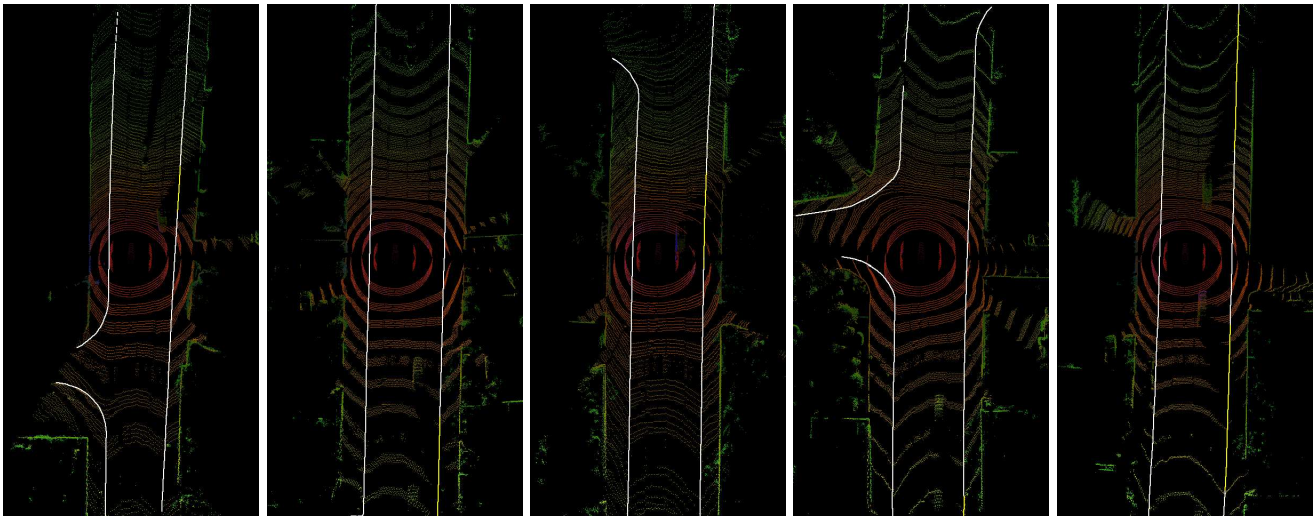


Fig. 6: Examples of labelled training data. Visible curbs are marked in white, while occluded curbs are marked in yellow.

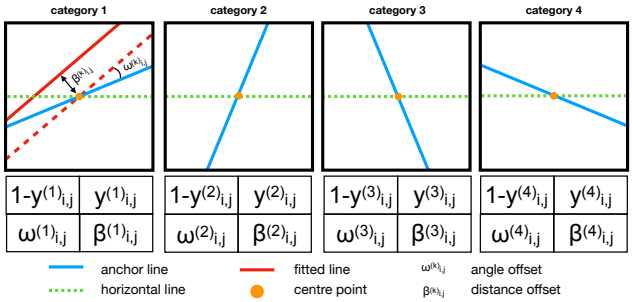


Fig. 7: Parameterisation of curb lines in discrete-continuous form. The four categories correspond to the four anchor lines.

(2) a distance from the centre of the cell to the fitted line ($\beta^k_{i,j,gt}$). As a result, we obtain 16 numbers for each grid cell, 4 numbers for each line category.

Estimating the presence of a curb line is a classification problem, but estimating adjustments to that line is a regression problem. To teach our network to perform the classification and regression at the same time, a discrete-continuous loss is applied during the training process. The total loss of the model L_t is defined as:

$$L_t = L_d + \alpha L_c = \sum_{i=1}^{S} L_{d_i} + \alpha \sum_{i=1}^{S} L_{c_i} \tag{1}$$

where L_d is a discrete loss of the curb line category classification, L_c is a continuous loss of the curb line parameters' regression, α is a weight term, and S denotes the number of scales (here 3).

Let $\hat{p}^k_{i,j}$ be a softmax output of the network for the k-th anchor line category in the j-th cell of the i-th scale, then the discrete loss for the i-th scale is:

$$L_{d_i} = -\sum_{j=1}^{C_i} \sum_{k=1}^{A} (y^k_{i,j} \log(\hat{p}^k_{i,j}) + (1 - y^k_{i,j}) \log(1 - \hat{p}^k_{i,j})) \tag{2}$$

where A is the number of anchor line categories (there are 4 categories), C_i is the number of cells in the i-th scale and $y^k_{i,j}$ is the ground truth for the k-th anchor line category in j-th cell of the i-th scale.

The continuous loss is a smooth $L1$ loss between the predicted line ($\omega^k_{i,j,pr}$, $\beta^k_{i,j,pr}$) and the ground truth line ($\omega^k_{i,j,gt}$, $\beta^k_{i,j,gt}$) parameters. The continuous loss for the i-th scale is defined as:

$$\begin{aligned} L_{c_i} = \sum_{j=1}^{C_i} \sum_{k=1}^{A} (y^k_{i,j} (&smooth_{L1}(\omega^k_{i,j,pr} - \omega^k_{i,j,gt}) \\ &+ smooth_{L1}(\beta^k_{i,j,pr} - \beta^k_{i,j,gt})) \end{aligned} \tag{3}$$

where $smooth_{L1}$ is define as in [22].

To increase the receptive field of the model we added intra-layer convolutions [23] before the multi-scale parameter estimation layers. Traditional layer-by-layer convolutions are applied between feature maps, but intra-layer convolutions are slice-by-slice convolutions within feature maps. Hence, intra-layer convolutions capture aspects across the whole image and can thereby capture spatial relationships over longer distances. For example, there is a strong correlation between the length of the occluded curbs and the size of objects which are obstructing the view (ranging from 10-15 pixels through occlusions by traffic cones to 200-300 pixels through occlusions by several parked cars).

C. Post-processing

Although the generated bird's-eye view images integrate five subsequent laser scans, our models do not take any temporal information into account explicitly. That is, each inference step is independent from the previous one. However, using temporal information can be useful in two ways: filtering out false positives and tracking true positives. To achieve this, we transform inputs at different time steps, t_l and t'_l, into a common reference frame using the following transformation:

$$T(t'_l, t_l) = \begin{bmatrix} R(t'_l, t_l) & t(t'_l, t_l) \\ 0 & 1 \end{bmatrix} \tag{4}$$

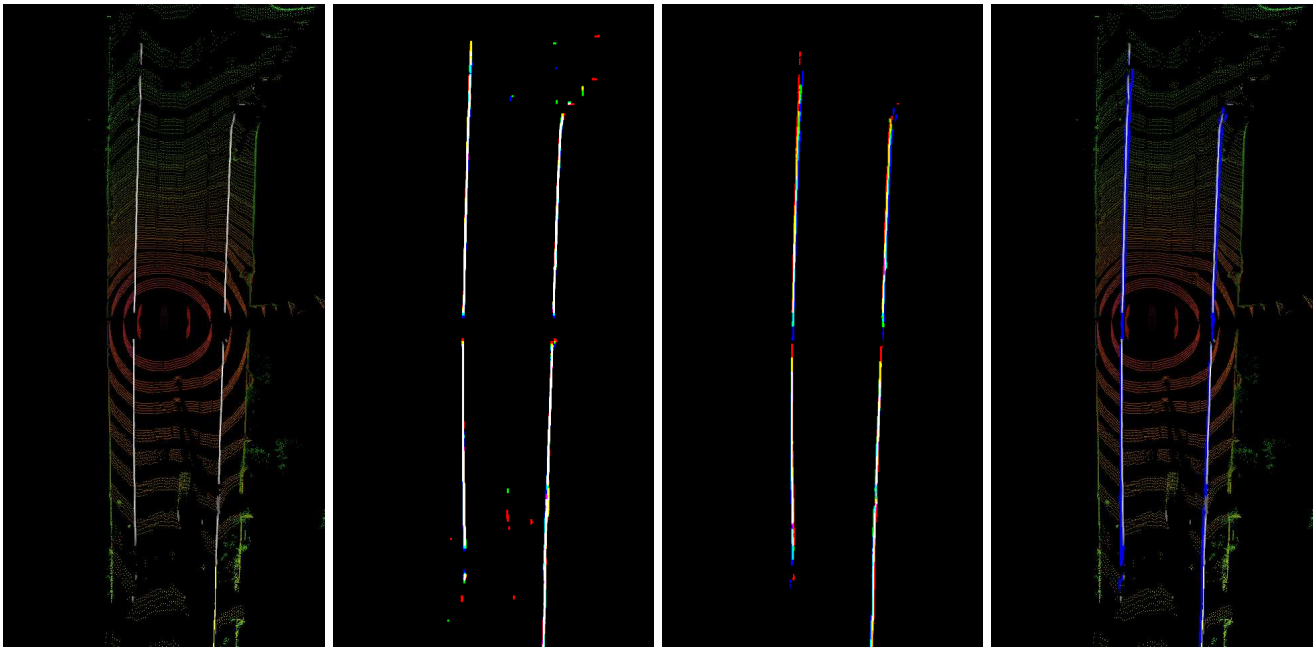


Fig. 8: Post-processing. From left to right: Output of curb detection networks (visible and occluded). Filtering step in which noise (highlighted in colour) is removed. Tracking step in which features tracked over time (in colour) are added. Finally, curb detection result after post-processing.

where $R(t'_l,t_l)$ is a rotation matrix and $t(t'_l,t_l)$ is a translation vector. As we explained in Section IV, VO is used to estimate the vehicle's ego-motion that allows us to obtain transformations between the vehicle's poses at different time steps. However, VO provides transformations corresponding to a camera frame rate. To obtain transformations between successive laser scans we use interpolation and calculate the transformation between time frames t_l and t'_l as follows [24]:

$$T(t'_l,t_l) = T(t'_l,t'_{vo}) \cdot T(t'_{vo},t_{vo}) \cdot T(t_{vo},t_l) \quad (5)$$
$$s.t.\ t'_{vo} \leq t'_l, \quad t_{vo} \geq t_l$$

where t'_{vo} and t_{vo} are the closest time steps of the VO with respect to laser frames and where $T(t'_{vo},t_{vo})$ is defined as follows:

$$T(t'_{vo},t_{vo}) = \prod_{i=t'_{vo}}^{t_{vo}-1} T(i,i+1) \quad (6)$$

To obtain $T(t'_l,t'_{vo})$ and $T(t_{vo},t_l)$ we interpolate in $[t'_{vo},t'_{vo}+1]$ and $[t_{vo}-1,t_{vo}]$ respectively. Using these transformations we apply two post-processing steps as follows:

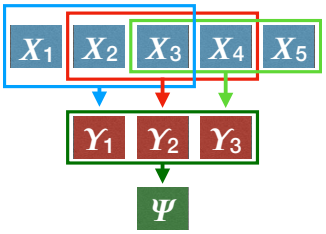


Fig. 9: Post-processing steps: A first step consolidates detection results of several subsequent scans (X_i) and generates a filtered output (Υ_i). A second step, tracks detected segments over a series of filtered images (Υ_i) and generates a final output (Ψ).

Filtering. In the first step, we transform the last three output masks of detected road boundaries into a common reference attached to the current frame. Then we construct a histogram of output mask size (480x960) by counting the number of overlapping pixels with a value grater than threshold of 0.7 (which was determined experimentally). We keep the detected road boundaries that appear in all three frames above the threshold and disregard the rest. As a result noise is filtered out from the output mask as shown on histogram in Figure 8.

Tracking. In the second step, we perform a similar procedure as outlined above. However, this time we consider road boundary masks from the last three frames that were generated by the first step (as shown in Figure 9). By taking the union of these masks we track the detected road boundaries over the time. Integrating temporal information helps to close gaps between boundary segments (Figure 8).

VI. Experimental Results

In this section, we show some qualitative results and provide an extensive quantitative evaluation of our approach. Qualitative results are presented in Figure 10 where outputs of curb detection networks and post-processing steps are included. The examples show that our approach is able to generate accurate masks for detected visible and inferred occluded road boundaries. To the best of our knowledge there is no public road boundary detection benchmark. Hence, we could not undertake a quantitative comparison of our approach with other existing approaches. However, we present an extensive quantitative evaluation based on our ground truth data.

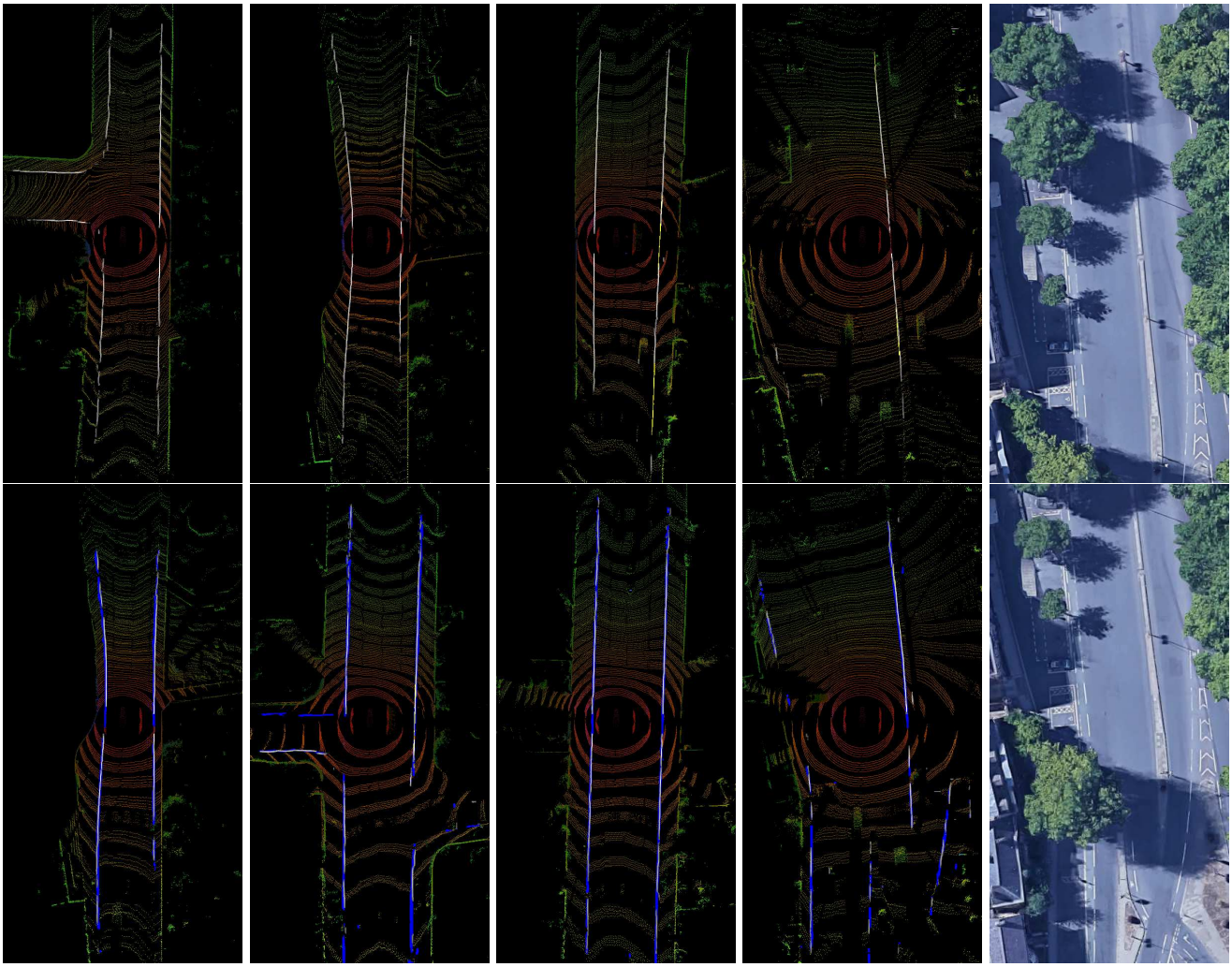


Fig. 10: First row: Sample outputs from the networks of detected and inferred road boundaries. Second row: Sample outputs after post-processing steps. The satellite images on the right depict the complexity of the scenes next to them.

1) Visible only: as we described in Section V, our approach tackles the detection of visible and inference of occluded road boundaries in two steps with a continuity constraint. The visible detector model is trained with 21K samples, which are generated by augmenting 1300 bird's-eye view images. Table III summarises the accuracy of the model with respect to 500 test images. We present precision, recall and F1 score for different areas around the vehicle (images size) and different tolerances. With minimum tolerance and maximum distance of 96 metres the model achieves 0.8168 F1 score. The model performs better in areas close to the car and achieves up to 0.9437 F1 score.

2) Visible and occluded: the occluded road boundary inference model is also trained with 21K samples that are obtained from 600 bird's-eye images through augmentation. The model infers road boundaries that are occluded by other road users and achieves 0.7867 F1 score. Note that for occluded road boundaries having higher tolerance is acceptable as their exact location is unknown. Combining outputs of both networks gives 0.893 F1 score with respect to all road boundaries.

TABLE I: Precision, recall and F1 score of the visible road boundary detection model

Image size / area	Tolerance	Precision	Recall	F1 Score
480x960 / 48x96 metres	1px	0.8371	0.7974	0.8168
	2px	0.9193	0.8387	0.8772
	3px	0.9455	0.8572	0.8992
	4px	0.9579	0.8698	0.9117
480x720 / 48x72 metres	1px	0.8631	0.8721	0.8676
	2px	0.9310	0.9050	0.9178
	3px	0.9520	0.9183	0.9349
	4px	0.9618	0.9272	0.9442
480x480 / 48x48 metres	1px	0.8819	0.8921	0.8870
	2px	0.9343	0.9021	0.9179
	3px	0.9541	0.9157	0.9345
	4px	0.9632	0.9250	0.9437

3) Post-processing: The importance of the post-processing step is to track detected and inferred road boundaries and fill in the gaps that were not detected initially. This happens on both sides of the vehicle as can be seen in Figure 8 (left). Those gaps are closed after the

TABLE II: Precision, recall and F1 score of the visible and occluded road boundary detection and inference models (size: 480x960)

Model	Labels	Tolerance	Precision	Recall	F1 Score
Occluded	Occluded	1px	0.5099	0.6420	0.5684
		2px	0.6414	0.7441	0.6889
		3px	0.7090	0.7938	0.7490
		4px	0.7523	0.8245	0.7867
Vis + Occ	Vis + Occ	1px	0.8318	0.7531	0.7905
		2px	0.9184	0.7980	0.8540
		3px	0.9477	0.8191	0.8787
		4px	0.9619	0.8333	0.8930

TABLE III: Precision, recall and F1 score with and without post-processing step (size: 480x480)

Pipeline	Tolerance	Precision	Recall	F1 Score
Vis + Occ	1px	0.8922	0.8387	0.8646
	2px	0.9470	0.8649	0.9041
	3px	0.9627	0.8784	0.9186
	4px	0.9706	0.8887	0.9279
Vis + Occ + Post	1px	0.5961	0.9246	0.7249
	2px	0.8773	0.9524	0.9133
	3px	0.9542	0.9594	0.9568
	4px	0.9741	0.9630	0.9685

post-processing step (right). Note that when generating filtered outputs (Υ_i) dilation is applied to the output masks to increase the overlap between output masks. As a result, the thickness of detected and inferred road boundaries increases, which leads to a decrease in precision and F1 score but an increase in recall for the minimum tolerance. However, whenever the tolerance is above 1px, the F1 score increases as the post-processing step fills in the gaps.

4) Running time: The system runs at 8.53 Frames Per Second (FPS) with input images of size 480x960 on a NVIDIA 1080 Ti GPU.

Overall, the experiments demonstrate that our approach is capable of detecting and inferring all road boundaries in complex road scenes over a total distance of 96 metres around the vehicle and in real-time.

VII. Conclusions

In this paper, we have presented a LIDAR-based approach for curb detection around the vehicle. Integrated LIDAR pointclouds are transformed into bird's-eye images which are then processed by trained convolutional networks. Instead of processing the whole pointcloud we only process a projected pointcloud in form of an image which allows our method to operate in real-time. Finally, we post-process the network outputs, i.e. the detected curbs (visible and occluded), by filtering out noise and tracking detections over time. This last step increases the overall performance of curb detection as we have shown in an extensive evaluation (Section VI).

Overall, we conclude that LIDAR data can be used to accurately detect curbs at high processing rates. Compared to camera-based based approaches, LIDAR has not only a greater coverage and range around the vehicle, but it is also more robust to environmental changes due to lighting and/or weather. The experiments demonstrated that our approach achieves a high performance in detecting and inferring visible and occluded road boundaries around the vehicle and achieves an F1 score of 0.9685 with post-processing. Hence, we strongly believe that our proposed deep learning approach based on LIDAR data can have a wide impact on a range of different applications in the autonomous driving domain.

Acknowledgment: The work has been supported by the EPSRC/UK Research and Innovation Programme Grant EP/M019918/1 (Mobile Autonomy) and the NVIDIA Corporation with the donation of Titan Xp and Titan V GPUs.

References

[1] L. Kunze, T. Bruls, T. Suleymanov, and P. Newman, "Reading between the lanes: Road layout reconstruction from partially segmented scenes," in *IEEE ITSC*, Maui, Hawaii, USA, November 2018.

[2] V. Prinet, J. Wang, J. Lee, and D. Wettergreen, "3D road curb extraction from image sequence for automobile parking assist system," *IEEE ICIP*, pp. 3847–3851, 2016.

[3] M. Kellner, U. Hofmann, M. E. Bouzouraa, and N. Stephan, "Multi-cue, model-based detection and mapping of road curb features using stereo vision," in *IEEE ITSC*, Sept 2015, pp. 1221–1228.

[4] L. Wang, T. Wu, Z. Xiao, L. Xiao, D. Zhao, and J. Han, "Multi-cue road boundary detection using stereo vision," in *IEEE ICVES*, July 2016.

[5] F. Oniga, S. Nedevschi, and M. M. Meinecke, "Curb detection based on a multi-frame persistence map for urban driving scenarios," in *IEEE ITSC*, Oct 2008, pp. 67–72.

[6] M. Kellner, M. E. Bouzouraa, and U. Hofmann, "Road curb detection based on different elevation mapping techniques," in *IEEE IV*, June 2014, pp. 1217–1224.

[7] J. Siegemund, U. Franke, and W. Förstner, "A temporal filter approach for detection and reconstruction of curbs and road surfaces based on conditional random fields," in *IEEE IV*, June 2011, pp. 637–642.

[8] M. Enzweiler, P. Greiner, C. Knöppel, and U. Franke, "Towards multi-cue urban curb recognition," in *IEEE IV*, June 2013, pp. 902–907.

[9] T. Suleymanov, P. Amayo, and P. Newman, "Inferring road boundaries through and despite traffic," in *IEEE ITSC*, November 2018.

[10] A. Y. Hata and D. F. Wolf, "Feature detection for vehicle localization in urban environments using a multilayer lidar," *T-ITS*, vol. 17, no. 2, pp. 420–429, Feb 2016.

[11] W. Yao, Z. Deng, and L. Zhou, "Road curb detection using 3d lidar and integral laser points for intelligent vehicles," in *SCIS/ISIS*, Nov 2012, pp. 100–105.

[12] Y. Zhang, J. Wang, X. Wang, and J. M. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3d-lidar sensor," *T-ITS*, vol. 19, no. 12, pp. 3981–3991, 2018.

[13] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," *IEEE CVPR*, Jul 2017.

[14] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast lidar-based road detection using fully convolutional neural networks," *IEEE IV*, Jun 2017.

[15] D. Nistér, O. Naroditsky, and J. R. Bergen, "Visual odometry," in *IEEE CVPR*. IEEE Computer Society, 2004, pp. 652–659.

[16] E. Rosten, G. Reitmayr, and T. Drummond, "Real-time video annotations for augmented reality," in *Advances in Visual Computing*. Springer, 2005, pp. 294–302.

[17] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "Brief: Computing a local binary descriptor very fast," *IEEE TPAMI*, vol. 34, no. 7, pp. 1281–1298, July 2012.

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

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

[20] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[22] R. B. Girshick, "Fast R-CNN," in *ICCV*, 2015.

[23] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," *arXiv preprint arXiv:1712.06080*, 2017.

[24] R. Aldera, D. De Martini, M. Gadd, and P. Newman, "Fast radar motion estimation with a learnt focus of attention using weak supervision," *IEEE ICRA*, 2019.

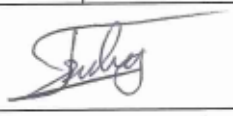
Statement of Authorship for joint/multi-authored papers for PGR thesis

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).**

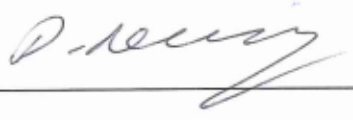
Title of Paper	Online Inference and Detection of Curbs in Partially Occluded Scenes with Sparse LIDAR
Publication Status	☐Published ☑ Accepted for Publication ☐Submitted for Publication ☐Unpublished and unsubmitted work written in a manuscript style
Publication Details	T. Suleymanov, L. Kunze, and P. Newman, "Online Inference and Detection of Curbs in Partially Occluded Scenes with Sparse LIDAR," in IEEE International Conference on Intelligent Transportation Systems (ITSC), Auckland, New Zealand, 2019

Student Confirmation

Student Name:	Tarlan Suleymanov		
Contribution to the Paper	My contributions to the paper were: Developing the initial idea behind the paper. Design and implementation of the research. Dataset annotation and preparation. Designed and performed the experiments. Analysing the results. Writing the paper, designing and making the figures.		
Signature		Date	12 September 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: Prof Paul Newman			
Supervisor comments AS AGREED.			
Signature		Date	12/9/2019

This completed form should be included in the thesis, at the end of the relevant chapter.

6.7 Summary of the Paper's Results

The experimental results presented in the paper demonstrate that the VRBD model trained with the IPM images based on the single VLP-32C LiDAR achieved 0.8168 F1 score with minimum tolerance and a maximum distance of 96 metres. The model performed better within the 48×48 squared metre region of interest and achieved 0.9437 F1 score. The ORBI model achieved 0.7867 F1 score with 4px tolerance for inferring occluded road boundaries. The combination of outputs from both models achieved 0.8930 F1 score for the detection and inference of all road boundaries without the post-processing step. The F1 score reached 0.9279 within the 48×48 square metre region of interest.

6.8 Pair of HDL-32E LiDARs-based models

After our first attempt, we decided to use the other two datasets (30-04-18 and 18-01-19) with a 2D LiDAR sensor because (1) annotating the 2D LiDAR-based point clouds was more than 3 times faster, and (2) the pair of Velodyne HDL-32E 3D LiDARs generated dense point clouds within a 24 metres range. We generated 17K samples from the 30-04-18 dataset for training and 2K samples from the 18-01-19 dataset for testing (see Table 6.1). The samples were generated with a size of 480×480 pixels and in two configurations of region of interest: 48×48 metres and 24×24 metres. New models, which were adopted to the input size of 480×480 pixels, as shown in Figure 6.7, were trained and tested. Evaluation results are presented in Table 6.2 which shows that the new models in both configurations outperformed the previous single LiDAR-based models. The F1 score for the detection of visible road boundaries within the region of interest (48×48 metres) increased from 0.8870 to 0.9272 with 1px tolerance and from 0.9437 to 0.9645 with 4px tolerance. Similarly, the F1 score for the detection of all road boundaries increased from 0.8646 to 0.9068 with 1px tolerance and from 0.9279 to 0.9589 with 4px tolerance. The results clearly show that the new VRBD and ORBI models outperform the previous models within the region of interest of 48×48 metres. Note that the models trained with the

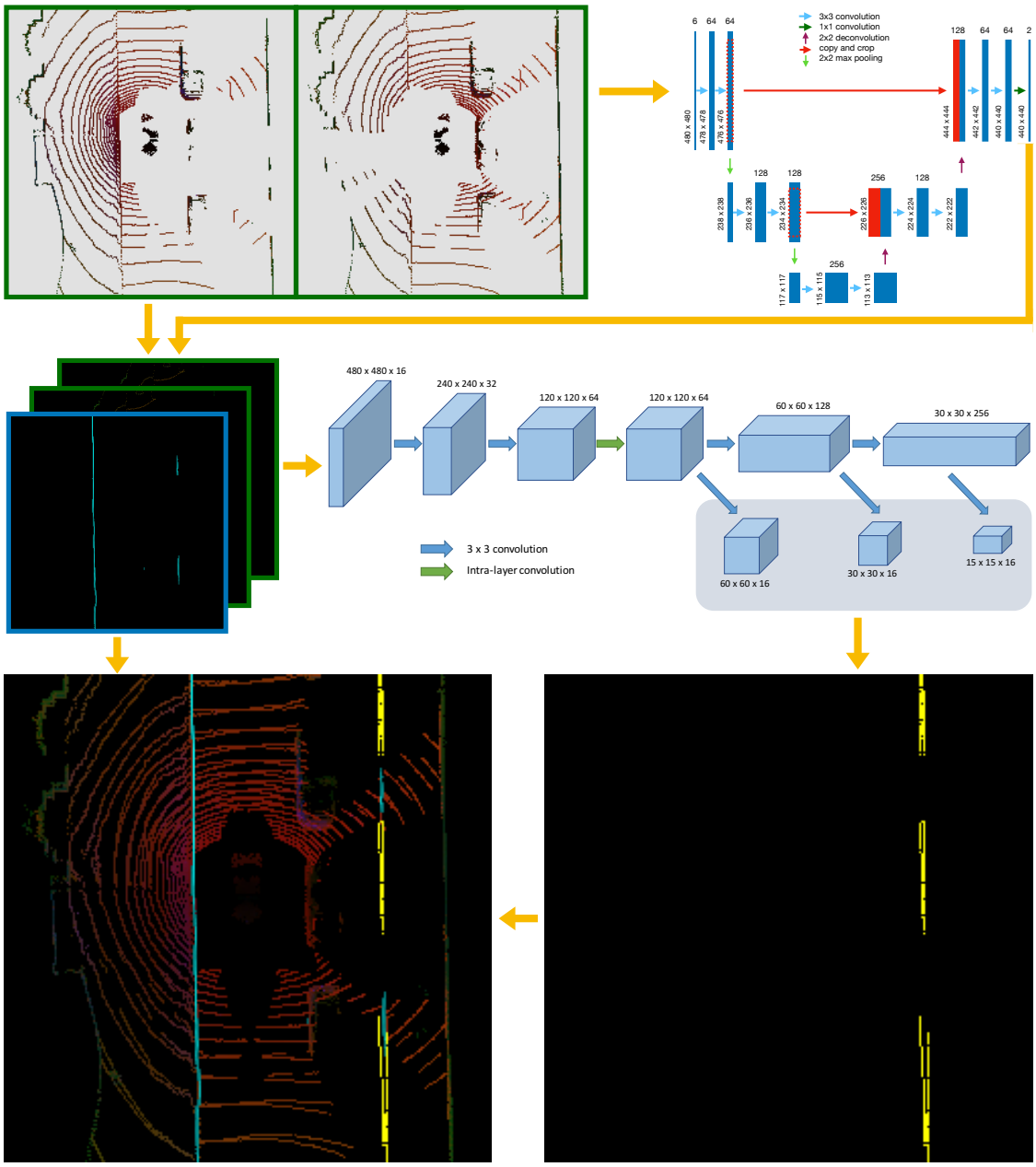


Figure 6.7: Our LiDAR-based coupled approach for road boundary detection. Given a pair of IPM images, the fully convolutional VRBD model detects visible road boundaries and then passes to the ORBI model for the inference of occluded road boundaries. The second model contains 3 base layers, intra-layer convolutions and 3 layers of parameterised multi-scale predictions at the end.

24×24 metres configuration also demonstrated similar performance to the models with the 48×48 configuration: a 0.9548 F1 score for visible road boundaries and 0.9524 for all road boundaries (with 4px tolerance).

Table 6.2: Precision, recall and F1 score of the visible and occluded road boundary detection and inference models (image size: 480 × 480 pixels)

Model	Area	Tolerance	Precision	Recall	F1 Score
Visible	48 × 48 metres	1px	0.9384	0.9163	0.9272
		2px	0.9733	0.9330	0.9527
		3px	0.9792	0.9413	0.9599
		4px	0.9817	0.9480	0.9645
Occluded	48 × 48 metres	1px	0.4987	0.6305	0.5569
		2px	0.6486	0.7417	0.6920
		3px	0.7322	0.7937	0.7617
		4px	0.7925	0.8277	0.8097
Vis + Occ	48 × 48 metres	1px	0.9103	0.9032	0.9068
		2px	0.9546	0.9254	0.9397
		3px	0.9665	0.9364	0.9512
		4px	0.9733	0.9449	0.9589
Visible	24 × 24 metres	1px	0.9086	0.9266	0.9175
		2px	0.9530	0.9385	0.9457
		3px	0.9598	0.9432	0.9514
		4px	0.9627	0.9470	0.9548
Occluded	24 × 24 metres	1px	0.5232	0.6151	0.5654
		2px	0.7058	0.7478	0.7262
		3px	0.8013	0.8046	0.8030
		4px	0.8508	0.8303	0.8404
Vis + Occ	24 × 24 metres	1px	0.8932	0.9156	0.9043
		2px	0.9445	0.9323	0.9383
		3px	0.9557	0.9393	0.9474
		4px	0.9608	0.9441	0.9524

6.9 Further Experimental Results

To demonstrate the ability of the VRBD and ORBI models to infer and detect outputs in a variety of scenarios, we present further qualitative examples here. Figure 6.8 shows examples where only visible road boundaries are present and detected by the VRBD model and no boundaries were hallucinated by the ORBI model. Output samples with both visible and occluded road boundaries are shown in Figure 6.9, where the road boundaries occluded by other road users were successfully inferred. Note that these examples are from the models with an ROI of 48×48 square metres. Similarly, we present examples of outputs of the models with an ROI of 24×24 square metres in Figure 6.10 and Figure 6.11. For more examples, see Appendix B.

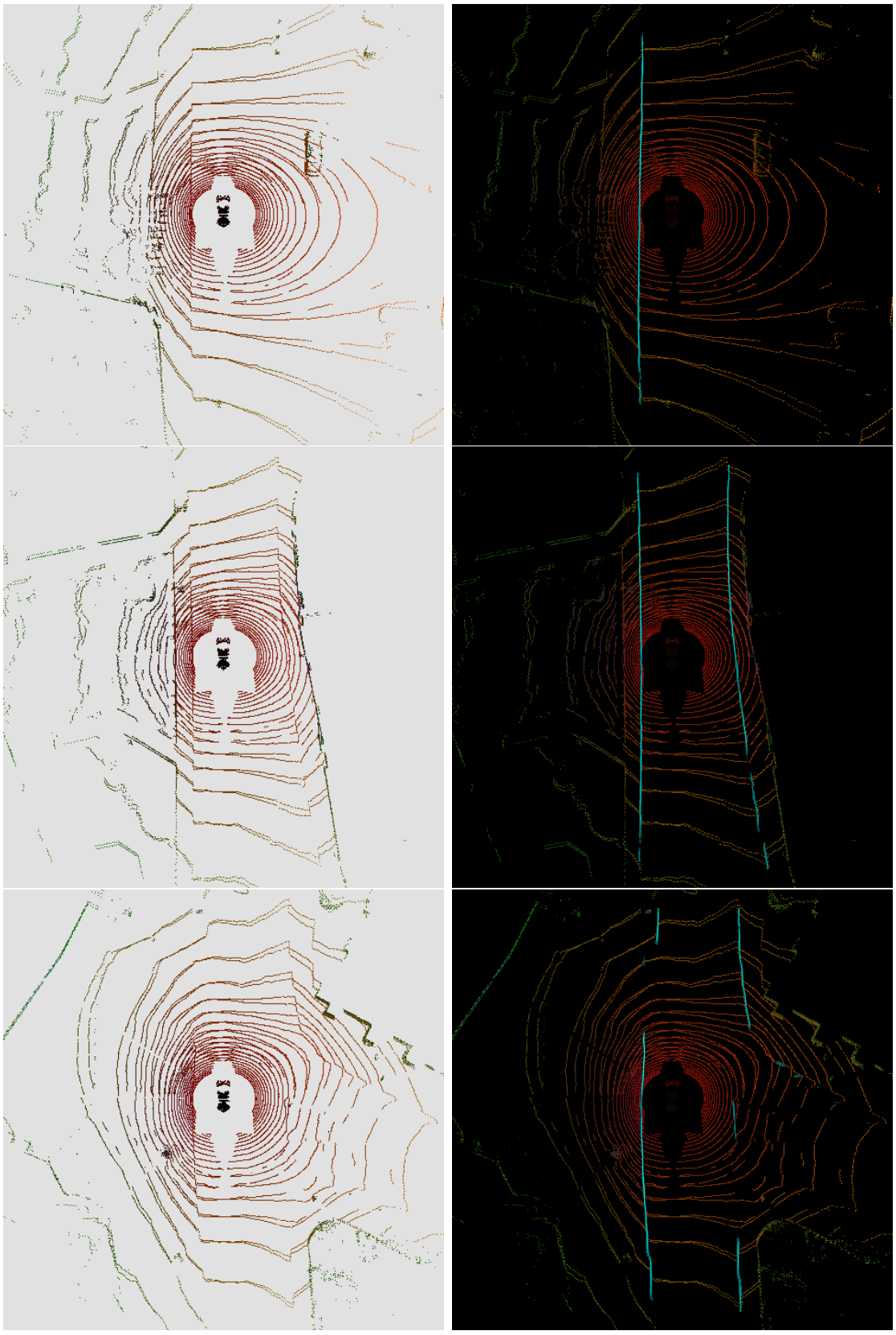


Figure 6.8: Output samples of detected visible road boundaries by the VRBD model with an ROI of 48×48 square metres.

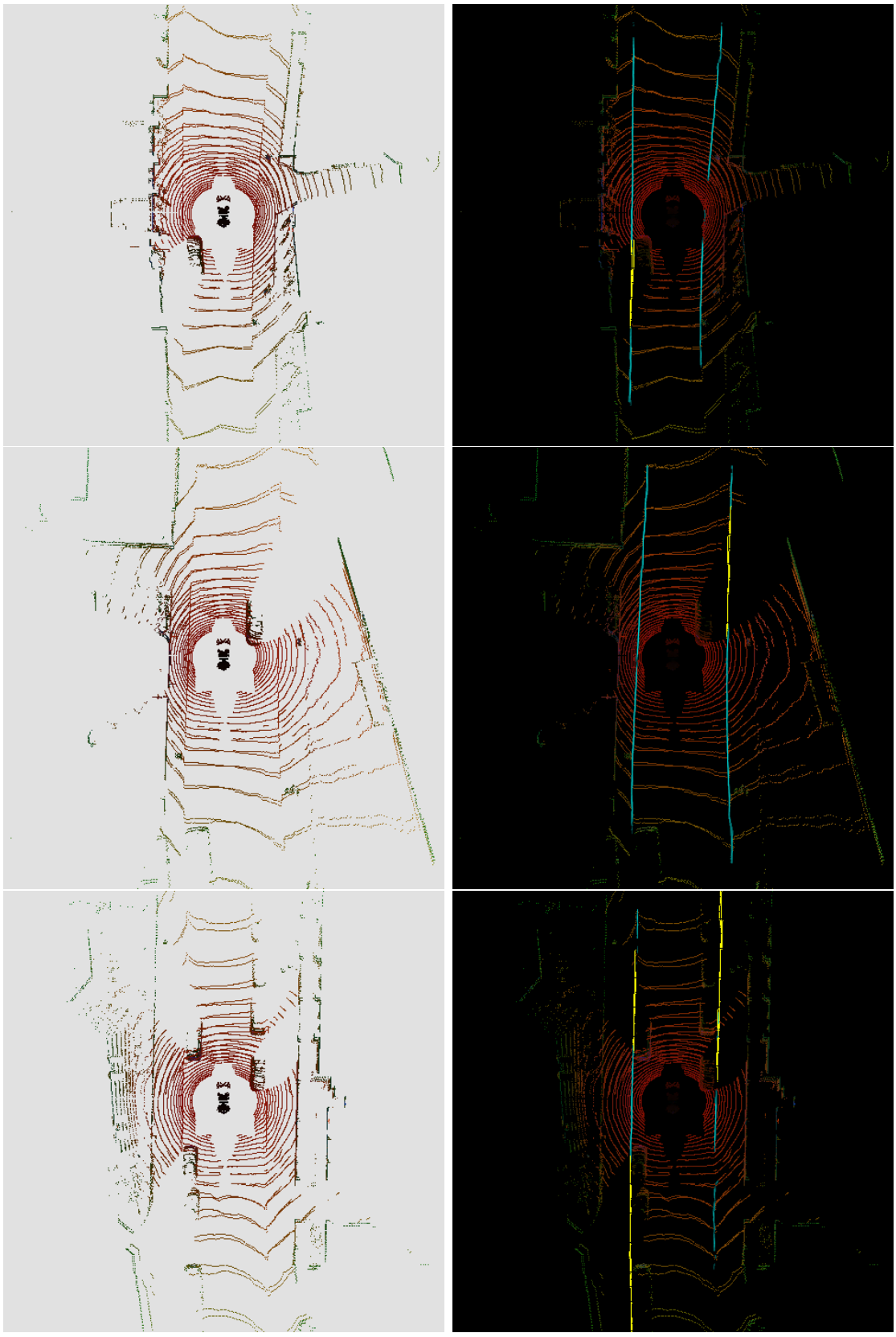


Figure 6.9: Output samples of detected visible and inferred occluded road boundaries by the VRBD and ORBI models with an ROI of 48×48 square metres.

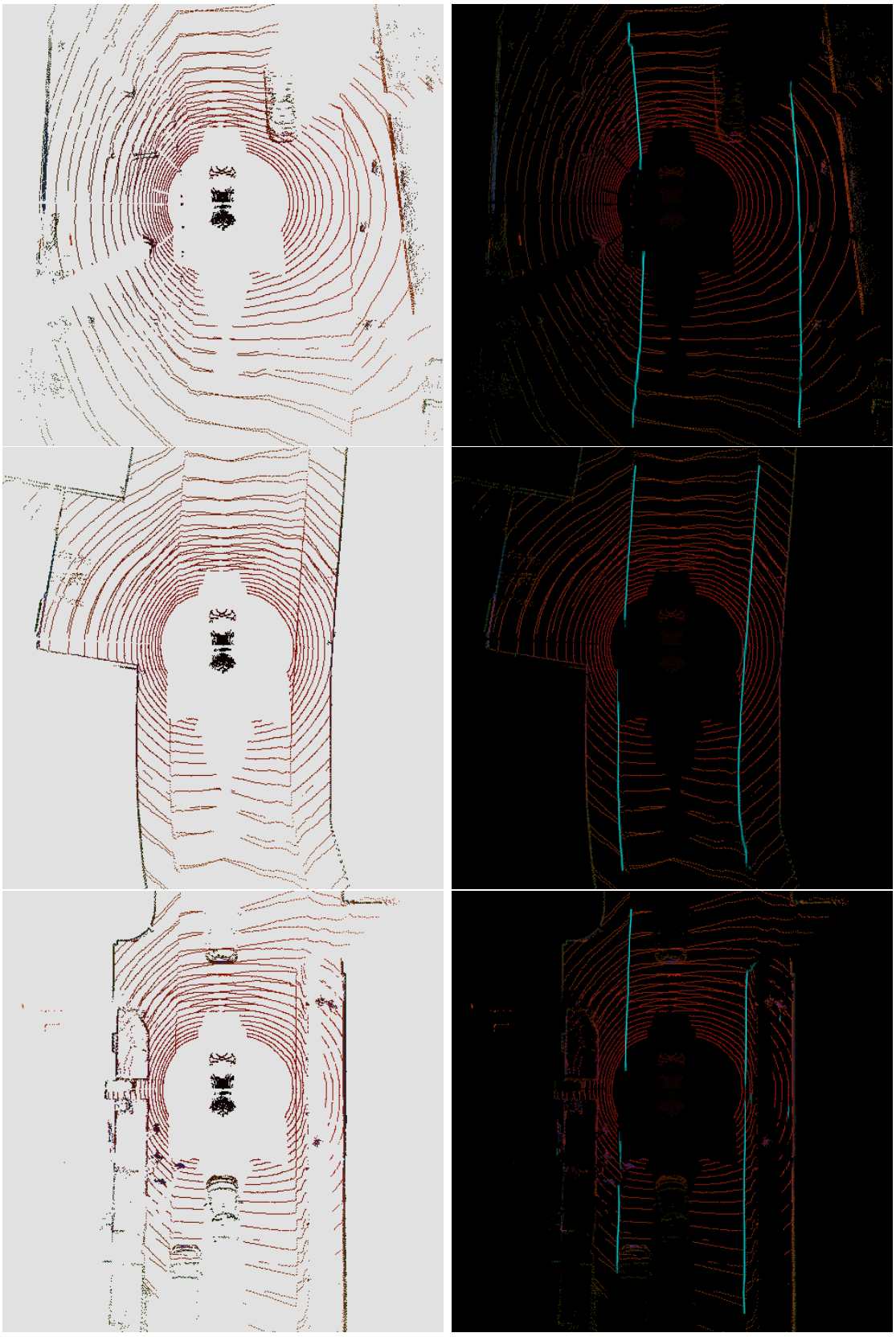


Figure 6.10: Output samples of detected visible road boundaries by the VRBD model with an ROI of 24×24 square metres.

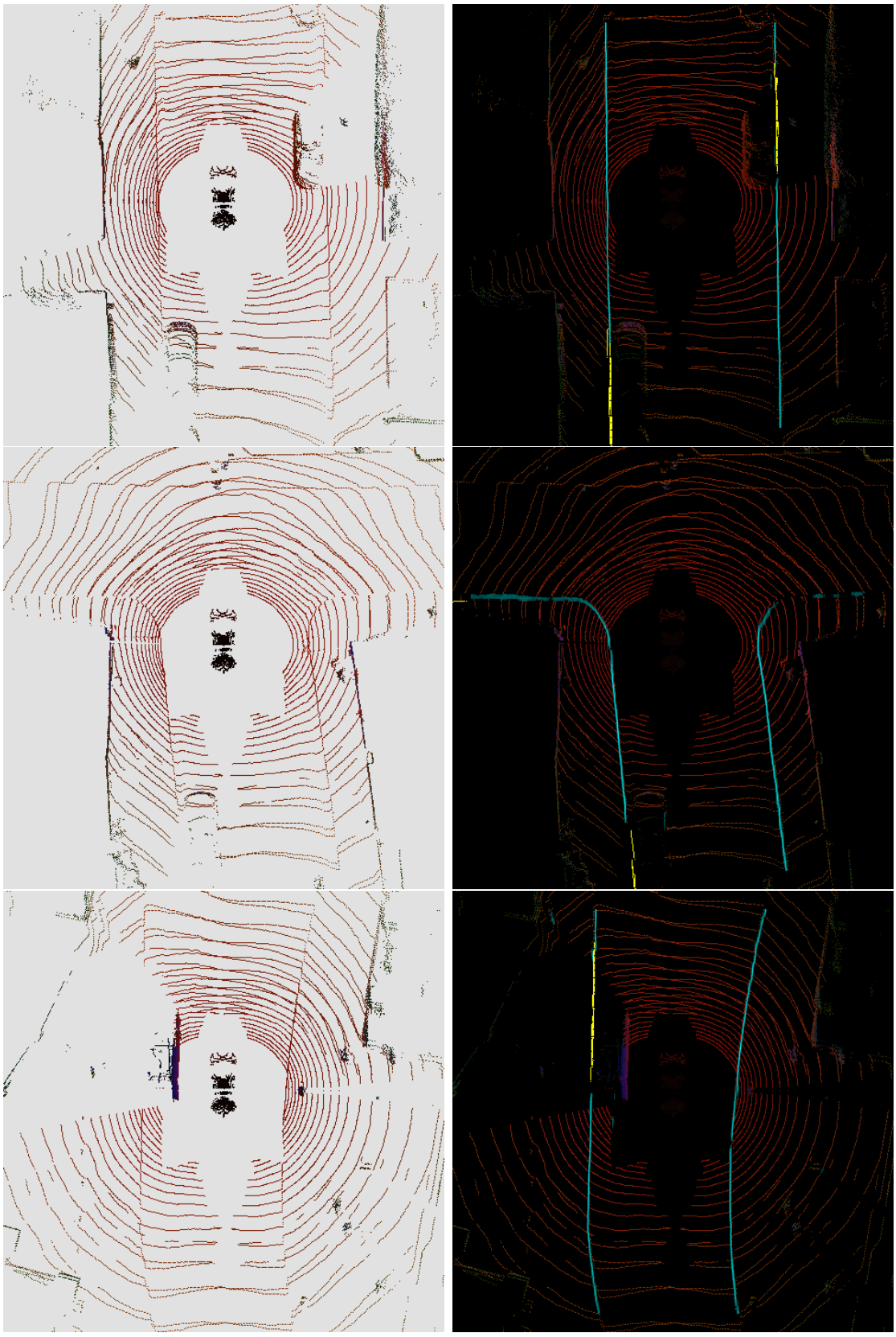


Figure 6.11: Output samples of detected visible and inferred occluded road boundaries by the VRBD and ORBI models with an ROI of 24×24 square metres.

6.10 Road Boundary Detection Failure Cases

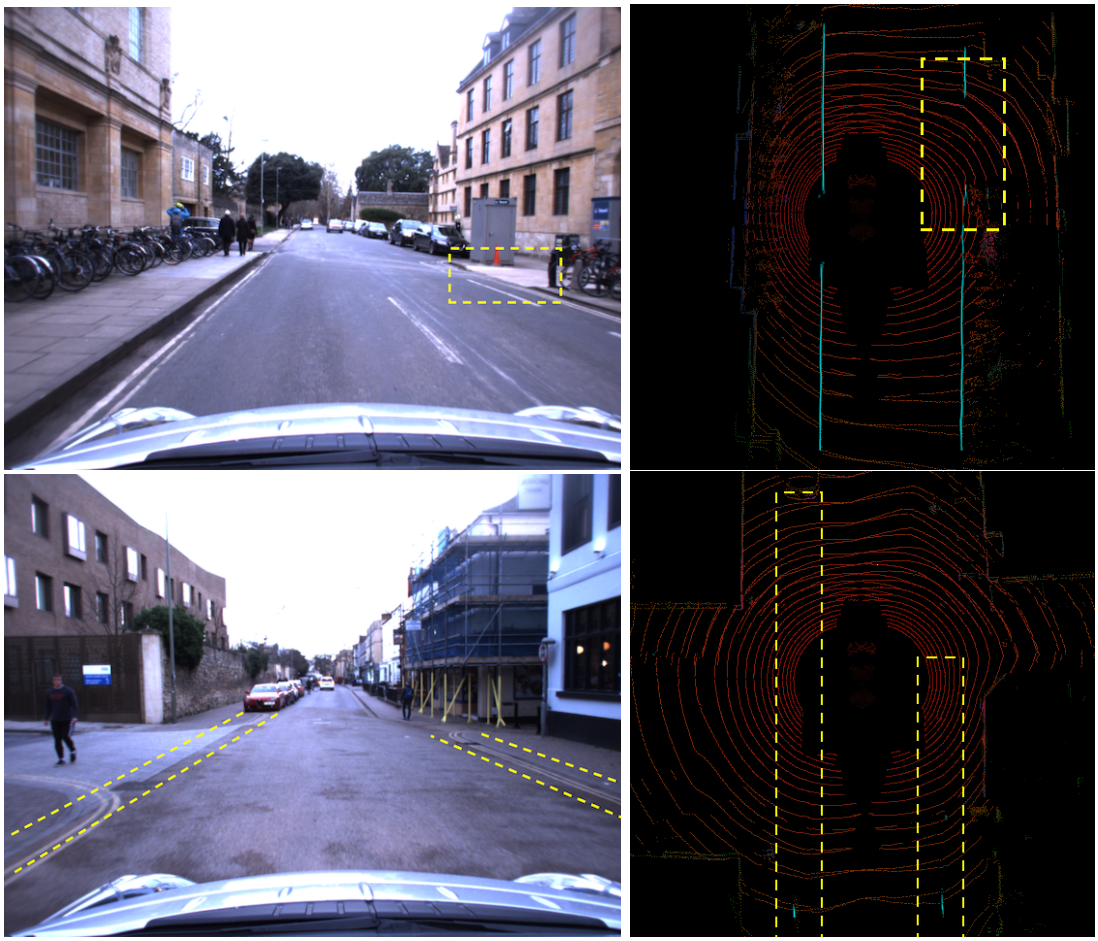


Figure 6.12: Road boundaries that do not have a height difference between the road surface and pavement are not distinguishable in the LiDAR scan. The VRBD model fails to detect those boundaries, and the ORBI model cannot infer them due to lack of contextual information.

LiDARs capture the 3D structure of a scene and provide valuable information for autonomous driving, but in some cases only having access to the 3D structure is not enough for road boundary detection. LiDARs can "see" road boundaries that present a change in height between road surface and pavement (curbs). However, road boundaries do not always have that distinguishable height difference, and the lack of structural difference makes those road boundaries invisible for LiDARs. In Figure 6.12 (top), there is a curb ramp on the right side of the road, which does not have a height difference with the road surface. The VRBD model cannot detect that section as it is invisible in the IPM image. The ORBI model cannot

infer, either, as the non-occluded gap could be a side road, and there is no obvious contextual information to rely on. In the bottom image of Figure 6.12, the road surface is at the same level with the pavements on both sides, making the road boundaries indistinguishable. Both models fail to detect the road boundaries. Again, there is not enough contextual information to infer the exact location and orientation of the road boundaries.

6.11 Lateral Localisation Experiment

To further evaluate our LiDAR-based road boundary detection and inference approach, we present an experimental application of its outputs to localisation. As mentioned at the beginning of this thesis, autonomous vehicles are required to perceive their surrounding environment and know their location in the world before they can plan a path to safely navigate to a desired location. We presented work on road segmentation, road boundary detection and scene understanding, which were about perceiving the environment of autonomous vehicles. Now, we take this perceived information - in this case detected road boundaries - and apply it to solve one of the fundamental tasks of autonomous driving and Advanced Driver Assistance Systems (ADAS): localisation, or more specifically, lateral localisation. Indeed, accurate lateral localisation is crucial for many ADAS, such as Lane Departure Warning (LDW), Lane Keeping Assist (LKA/LKS) and Parking Assist System. However, these systems are susceptible to cluttered environments, where traffic and other obstructions are likely to disrupt the robust performance required for safe operation of the vehicle.

Localisation techniques can be classified into two groups: global and relative [32]. Global localisation of a robot can be achieved using Global Navigation Satellite Systems (GNSS), but it is not precise enough for autonomous driving as the accuracy is worse than 2-3 metres in an open-sky environment [33]. In Figure 6.13, we observe that there are many gaps along the route when samples from the 18-01-19 dataset are overlaid on a digital map using GPS coordinates. Visual Odometry, Laser Odometry and SLAM-based approaches are relative localisation techniques that

Figure 6.13: 31100 samples of the 18-01-19 dataset are overlaid on a digital map using GPS coordinates. Although the dataset is a complete loop, we observe many gaps along the route.

mainly focus on ego-motion. An alternative approach to localisation is to use map-based techniques as the maps can be built in advance and accurately, which makes them suitable for autonomous driving. Depending on the type of input sensor, these approaches can be categorised into two groups: camera-based (passive sensor) and LiDAR-based (active sensor). Camera-based methods are sensitive to lighting conditions, shadows, illumination and under- and overexposure. Using dense maps in LiDAR-based localisation approaches is usually computationally expensive for

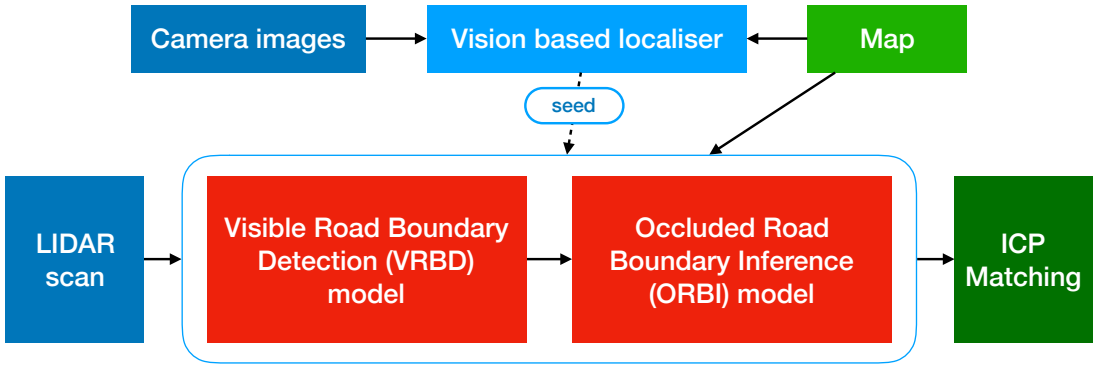


Figure 6.14: An overview of the localisation pipeline. After localisation is coarsely initialised, the live LiDAR scan is passed through our VRBD and ORBI models. This gives us not only visible road boundaries but also inferred locations for the occluded parts of those road boundaries. These are then matched to a map which has been similarly processed for visible and occluded road boundaries. Note that localisation is coarsely initialised by a camera stream, but this part of the system is interchangeable with e.g. LiDAR place recognition systems, such as in [31].

running in real-time. Regardless of the sensor used, feature matching is often used to match inputs from the sensors to maps. Lane markings, traffic signs, feature points or road boundaries can be used as features for localisation. The long and continuous shape of road boundaries makes them stable and robust features for localisation in the lateral direction as they capture the structure of roads, and, as such, in our experiment we used LiDAR-based detected road boundaries.

6.11.1 Experimental Setup

Our cross-track localisation experimental approach, the workflow for which is shown in Figure 6.14, is aimed to (1) demonstrate the usability of the outputs of our LiDAR-based road boundary detection approach for localisation and (2) demonstrate the gain in performance that the inferring of occluded road boundaries could bring. To run the experiment, we first selected two datasets, one dataset as a map and the other as a live input. Secondly, we ran the inference over the map dataset using our LiDAR-based VRBD and ORBI models with a region of interest of 24×24 squared metre. The model with the smaller region of interest was used to increase the accuracy of lateral localisation as 1px corresponded to 5 cm in the real world, while the model with the 48×48 squared metre of region of interest represented 10 cm.

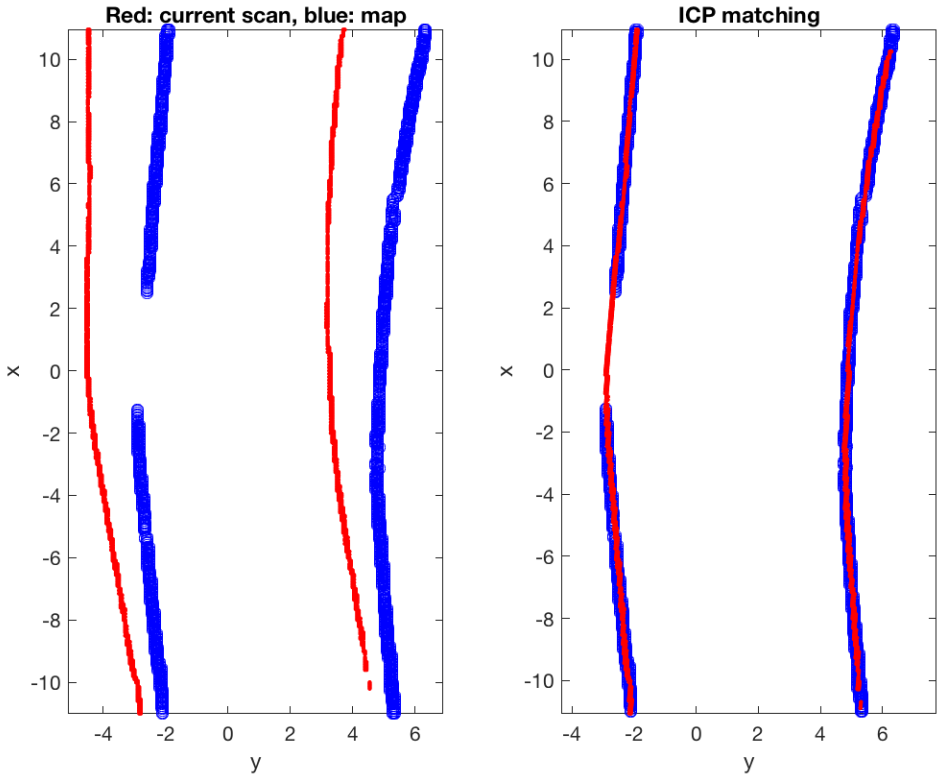


Figure 6.15: Given the detected road boundaries mask of a scan, it was binarised and transformed into a point cloud to match with a scan from the map. ICP was used to match the road boundaries and estimate the transformations between the frames. This example shows the accurate estimation of the transformation despite the undetected section of the road boundaries in the map frame.

Outputs of the detected road boundaries were stored in the map with corresponding timestamps. Thirdly, the second dataset was used as a live input, and the detected road boundaries from the second dataset were matched against the map dataset to perform lateral localisation. We assumed that the initial guess of the location of the vehicle in the map was provided by a vision-based localiser [34]. Note that the vision-based localiser only provided timestamps of corresponding images from the map without providing initial pose. We used the Iterative Closest Point (ICP) algorithm [35] to perform the matching process and estimate the transformations between the live inputs and the map. ICP is a well-known algorithm that is used for matching point clouds, where the algorithm iteratively updates the transformation between two point clouds to minimise the distance between them. We adopted

ICP to estimate the transformation between live and map samples. The detected road boundary masks were binarised and converted into point clouds and then matched with ICP, as shown in Figure 6.15.

6.11.2 Qualitative ICP Results

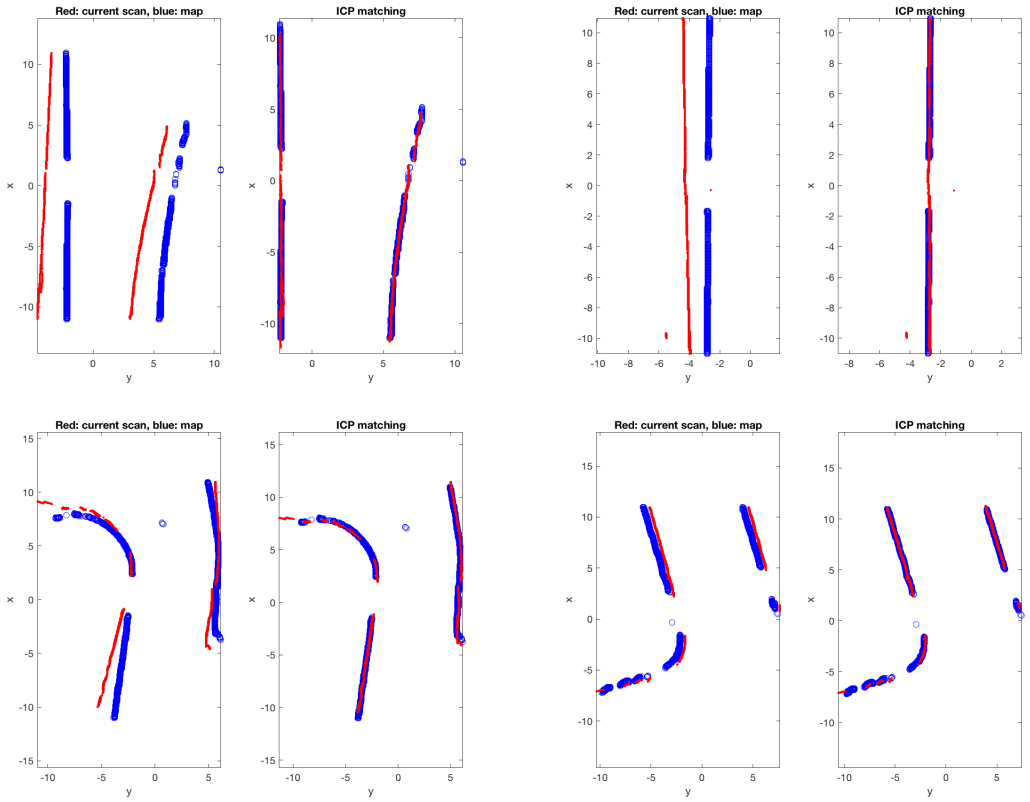


Figure 6.16: Examples of road boundary based ICP matching for localisation. These examples demonstrate that ICP accurately estimated the transformations between samples irrespective of the structure of the detected road boundaries since the detected road boundaries between samples were balanced over the sections of the true boundaries.

We present some examples of road boundary based ICP matching for localisation in Figure 6.16. The examples demonstrate that ICP accurately estimated the transformations between samples irrespective of the structure of the detected road boundaries. Small amounts of noise in the detection did not change the overall estimation of the transformations given that the road boundaries were detected in a balanced way over the sections of the true boundaries. ICP could accurately estimate the transformation in the presence of detected road boundaries on one

side of the road, as shown in Figure 6.16 (top right). For more examples of ICP matching, see Appendix C.

6.11.3 ICP Failure Cases

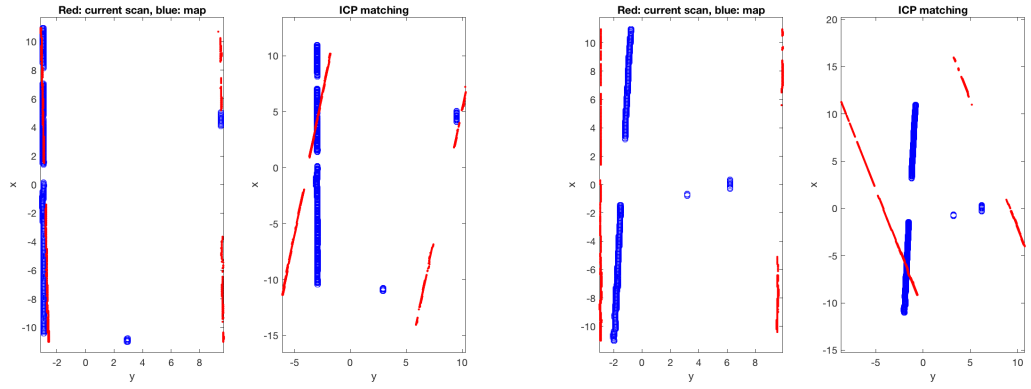


Figure 6.17: ICP matching failure examples, where the detected road boundaries on the right-hand side of the road were unbalanced between samples.

In some cases, ICP failed to match the road boundaries accurately to generate transformations between samples. This happened when the detected road boundaries were unbalanced. As shown in Figure 6.17, in both examples only a small section of road boundaries was detected on the right side of the road from live inputs. This forced ICP to rotate the live inputs as keeping them parallel was more costly. This can be fixed using more sophisticated matching techniques, but this is beyond the scope of this thesis. The goal here is to illustrate how road boundary detection has utility in localisation. Consider Figure 6.18, in which we included in the matching detections of occluded road boundaries. Without more sophisticated matching techniques, we were now able to obtain sane transformations.

6.11.4 Experimental Localisation Results

To evaluate our approach, we used the vision-based localiser, which we also used in our data annotation process, to obtain near ground truth transformations between the camera images of the datasets. We compared the estimated road boundary based lateral localisation results with the lateral localisation of the vision-based

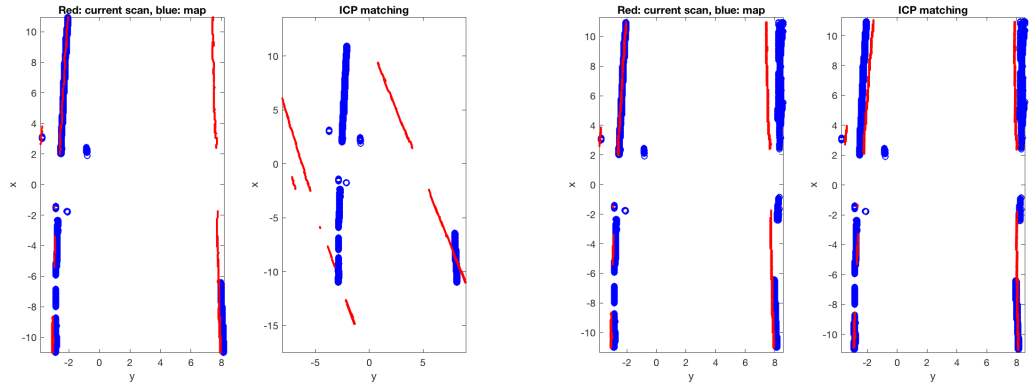


Figure 6.18: Comparison: ICP matching based on detected visible road boundaries only (left) and based on all road boundaries (right).

Table 6.3: Road boundary based lateral localisation results comparing localisation based on visible road boundaries only with localisation based on a combination of visible and occluded road boundaries. The results show that using the inferred occluded road boundaries always improved performance.

Error range	Visible only		Visible and occluded	
	Number of samples	Percentage	Number of samples	Percentage
Within 0.1 metre	20389	65.56%	21934	70.53%
Within 0.3 metre	27479	88.36%	28349	91.15%
Within 0.5 metre	29005	93.26%	29617	95.23%
Within 1 metre	30124	96.86%	30504	98.08%

localiser. We ran the experiment based on 31100 samples. Note that the IPM images that were used for estimating the lateral localisation were interpolated to match the timestamps of the camera images that the vision-based localiser used. We calculated average lateral error (mean absolute error) and yaw error for the lateral localisation based on (1) visible road boundaries only and (2) a combination of all road boundaries. The average lateral localisation error based on visible road boundaries was only 18.95 cm. Including the inferred occluded road boundaries decreased the error by 4.23 cm to 14.72 cm. Similarly, the yaw error decreased from 0.0332 rad to 0.0206 rad.

We further analysed the output results by counting the number of samples that had a lateral error within 1, 0.5, 0.3 and 0.1 metres. Table 6.3 shows that using the inferred occluded road boundaries increased the percentage of the number of samples within 1 metre from 96.86% to 98.08%. Similar gains were achieved for the number of

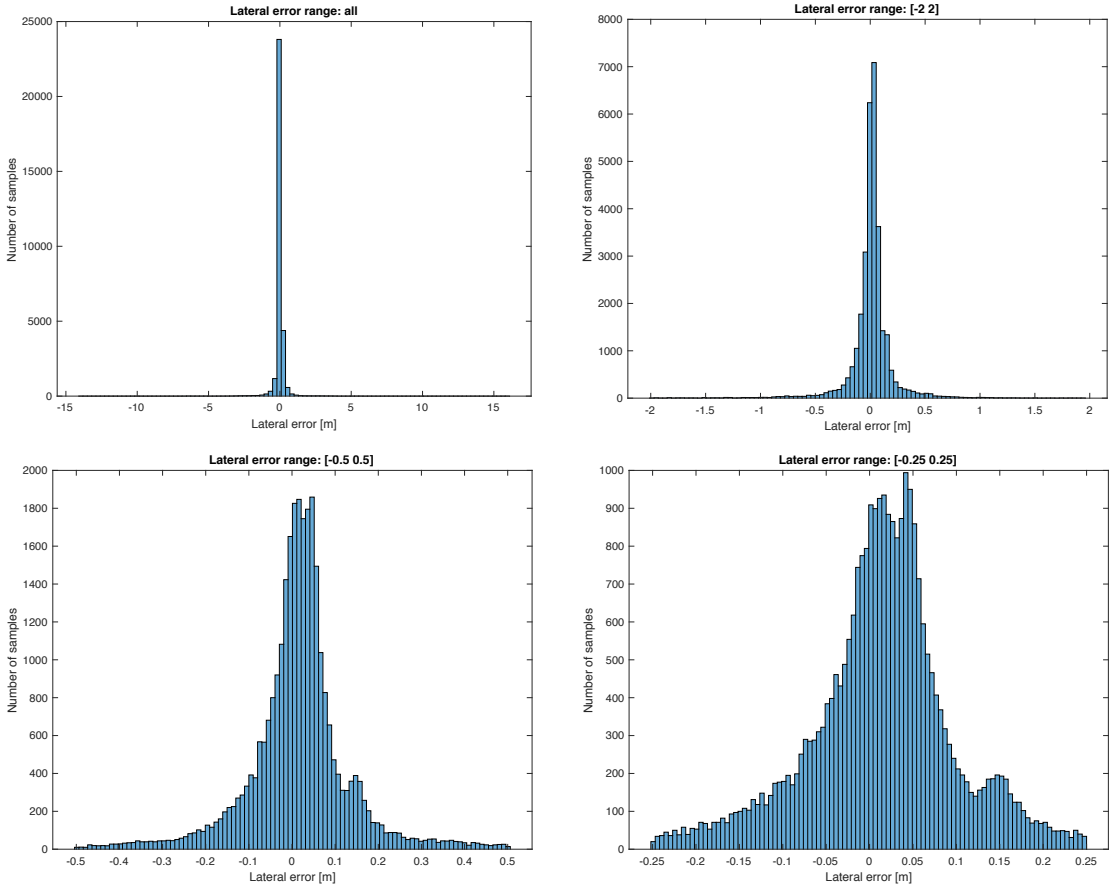


Figure 6.19: Histograms of road boundary based lateral error. The top left histogram includes all samples from the dataset, while the remaining histograms progressively narrow the displayed error range (horizontal axis). We observe that the majority of the samples (70.53%) have a maximum lateral error smaller than 10 cm.

samples within 0.5, 0.3 and 0.1 metres. Overall, 70.53% of the samples (21934) had a lateral error less than 10 cm in contrast with the vision-based localiser. To visualise the lateral error of the samples, we plotted them as histograms in Figure 6.19.

Figure 6.20 (top) displays the lateral error of all samples in a timeline. We observe that the majority of the samples (98.08%) have an error less than one metre and that there are only a small number of peaks where localisation failed. Zooming in to a section of the timeline shows that the large number of samples (70.53%) has a lateral error less than 10 cm.

Another timeline, plotted in Figure 6.21, contains only 1000 samples and shows the importance of occluded road boundaries. The blue points in the figure represent the lateral error based on only visible road boundaries, where the red points are

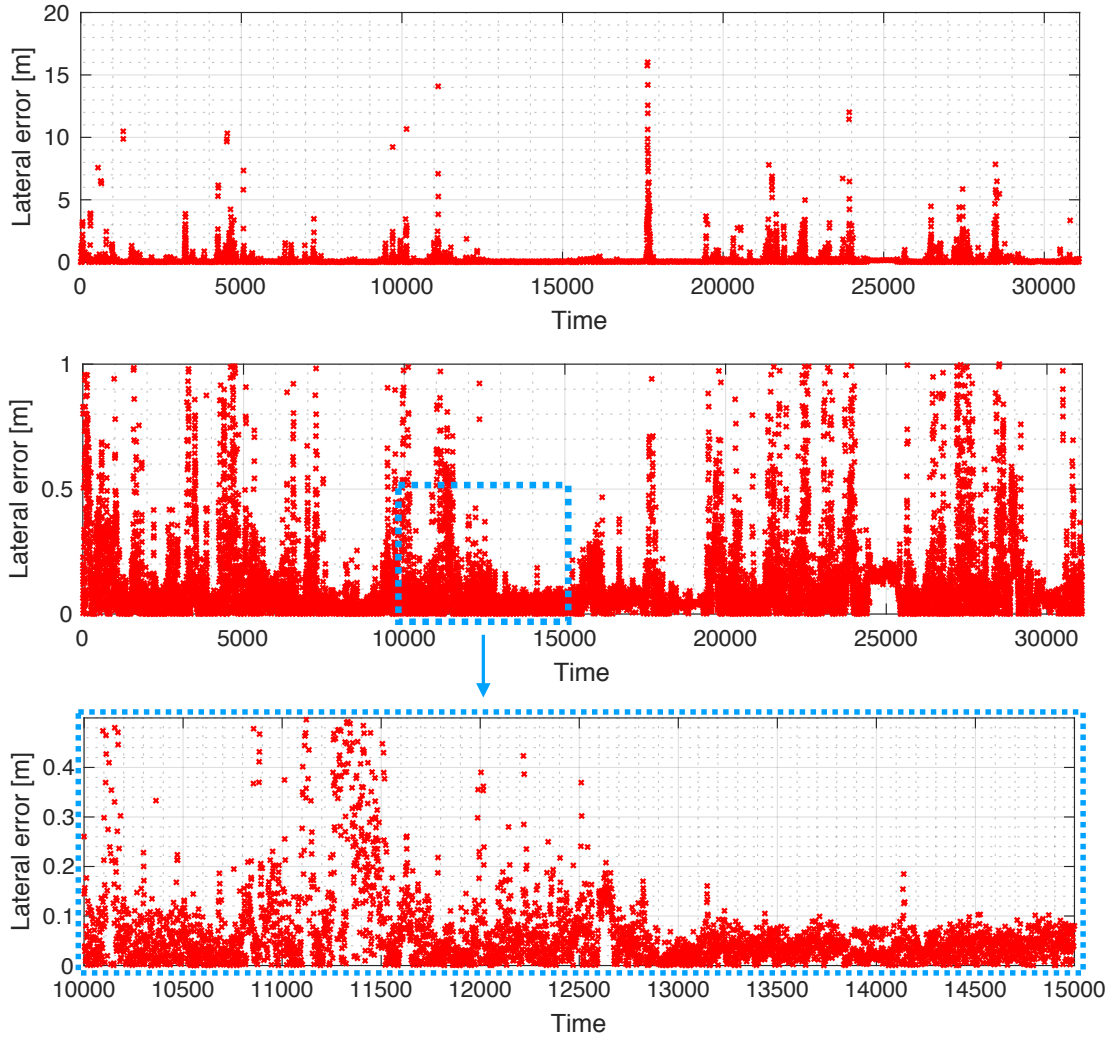


Figure 6.20: Lateral error of all samples displayed in a timeline (top), where we observe that there are only a small number of peaks where localisation failed. Progressively narrowing the displayed error range (vertical axis) shows that the majority of the samples (70.53%) have a maximum lateral error within 10 cm.

the localisation errors with all road boundaries. Overall, we observe that the blue points have higher values than the red ones.

Finally, we performed an experiment with another pair of datasets and calculated the average lateral localisation error. The same experiment was conducted using ICP with worst rejection, and a similar performance gain was observed when using occluded road boundaries. The average lateral localisation error for both pairs of datasets is presented in Table 6.4.

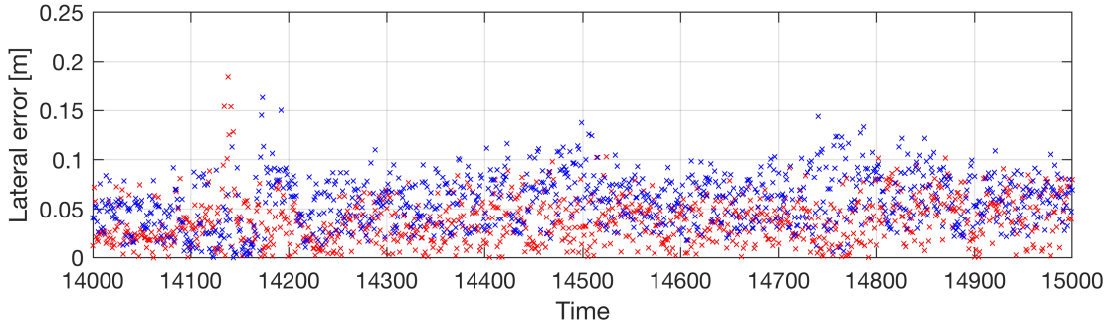


Figure 6.21: A timeline of 1000 samples displaying lateral errors based on only visible road boundaries (blue points) and errors based on all road boundaries (red points). We observe that the blue points are generally larger than the red ones, indicating that using all road boundaries is better for performance.

Table 6.4: Average lateral localisation error for both pairs of datasets.

Matching method	Map dataset	Live dataset	Visible only	Visible and occluded
ICP	2018-04-30	2019-01-18	18.95 cm	14.72 cm
ICP	2019-01-10-11	2019-01-10-12	18.54 cm	12.18 cm
ICP with worst rejection	2019-01-10-11	2019-01-10-12	9.33 cm	7.51 cm

6.12 Conclusions

In this chapter, we presented a deep learning-based road boundary detection approach that relies on LiDARs and infers road boundaries irrespective of boundary visibility. The quantitative and qualitative experimental results demonstrated that the approach is capable of detecting road boundaries in various scenarios and achieved more than 0.9 F1 score. We investigated the limits of the system's performance and presented experimental results. In future work, we aim to extend the lateral localisation experiment to a full system.

7 Conclusions and Future Work

Contents

This chapter summarises the motivation, major ideas, approaches, contributions and results presented in Chapters 3 to 6. We finish the thesis by presenting a discussion of the future directions of research and some closing remarks.

7.1 Summary

This thesis has presented several approaches to address multiple challenges faced by self-driving vehicles in real-world driving scenarios. In Chapter 3, we discussed scene segmentation for the collision-free navigation of a robot. The proposed approach was an on-line, light-weight system that combined a Random Forest with a variational approach for regularisation via convex relaxation. The system required small amounts of training data and performed well in predefined environments by providing reasonable segmentation results for path planning and safe navigation of the robot. To address the problem of the generalisation of machine learning models, we shifted to deep learning-based models. Often, however, large numbers of training

samples are required to obtain well-generalised, high-performance deep models. Our solution to this problem was presented in Chapter 4, where we proposed a road boundary annotation framework that enabled us to easily obtain thousands of annotated training samples for camera- and LiDAR-based input data. One of the main contributions of this thesis was presented in Chapter 5, where we combined the power of deep learning with the data obtained from our framework to detect and infer road boundaries irrespective of boundary visibility by taking inspiration from human perception, which uses contextual information to perceive the beyond visible spectrum. Our camera-based road boundary detection approach consisted of VRBD and ORBI models that were linked with a continuity constraint, where the outputs of detected visible road boundaries from the VRBD model were given as inputs to the ORBI model and provided clues about occluded road boundaries. We added intra-layer convolutions to our new deep learning architecture to pass information across the rows and columns of inputs to capture other contextual clues in the scene, which significantly improved the performance. The model estimated multi-scale parameterised outputs in a discrete-continuous form, enabling the network to correctly estimate the position of occluded road boundaries irrespective of the size and shape of occluding obstacles. Having achieved high performance in camera-based road boundary detection, we adapted this approach in LiDAR-based road boundary detection, presented in Chapter 6. We achieved similarly high performance in detecting visible road boundaries and inferring occluded ones using LiDARs. Finally, we applied detected road boundaries from our camera-based approach to scene understanding in Chapter 5 and our LiDAR-based approach to localisation in Chapter 6.

7.2 Future Work

The work presented in this thesis primarily contributes to the understanding on image segmentation and road boundary detection problems and brings new opportunities for future research. Our directions for future work, including a variety of ways for improving the proposed approaches, are as follows:

- A pre-processing step to combine camera and LiDAR data as inputs to the VRBD and ORBI models,
- A post-processing step to combine the outputs of detected road boundaries of the camera-based and LiDAR-based approaches,
- Using prior maps, such as OpenStreetMap, and integrating temporal information to prevent unreasonable road boundary detections,
- Making the road boundary dataset publicly available for benchmarking,
- Extending the lateral localisation experiment to a full system, and
- Using the camera-based ORBI model to improve scene understanding results in the presence of occlusions and integrating the outputs of the LiDAR-based approaches.

7.3 Closing Remarks

This thesis sought to contribute to the understanding on image segmentation and road boundary detection problems in robotics. Our primary contributions are the camera-based and LiDAR-based road boundary detection approaches that are motivated by human perception and use contextual information to infer road boundaries irrespective of whether or not the boundaries are actually visible. We hope that the presented approaches here will serve as valuable tools for the robotics community and contribute to the development of self-driving cars.

Appendices

A

Camera-based Road Boundary Detection Examples

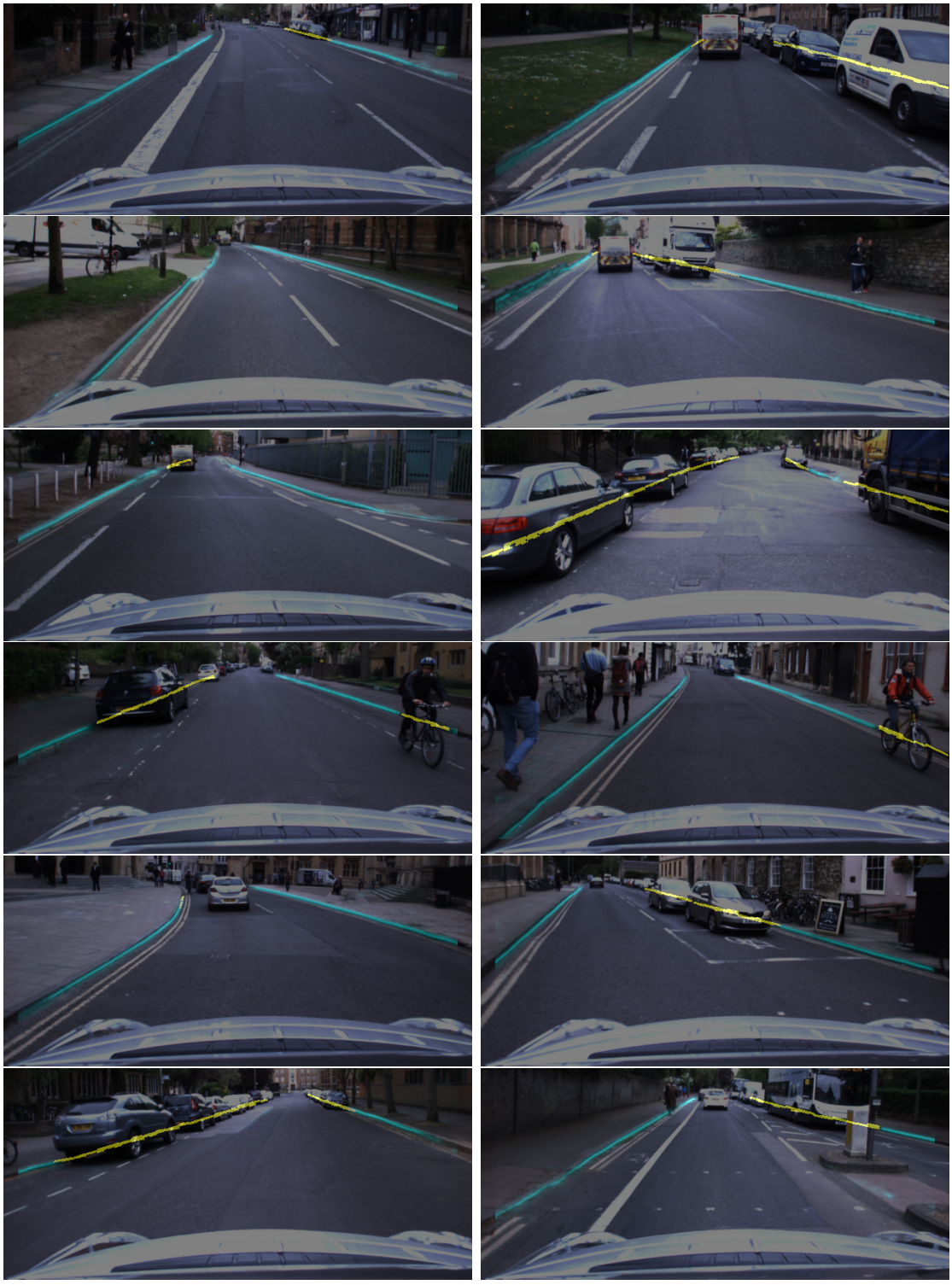


Figure A.1: Camera-based road boundary detection examples. Visible road boundaries (cyan) are detected by the VRBD model, and occluded road boundaries (yellow) are hallucinated by the ORBI model.

Figure A.2: Camera-based road boundary detection examples. Visible road boundaries (cyan) are detected by the VRBD model, and occluded road boundaries (yellow) are hallucinated by the ORBI model.

B

LiDAR-based Road Boundary Detection Examples

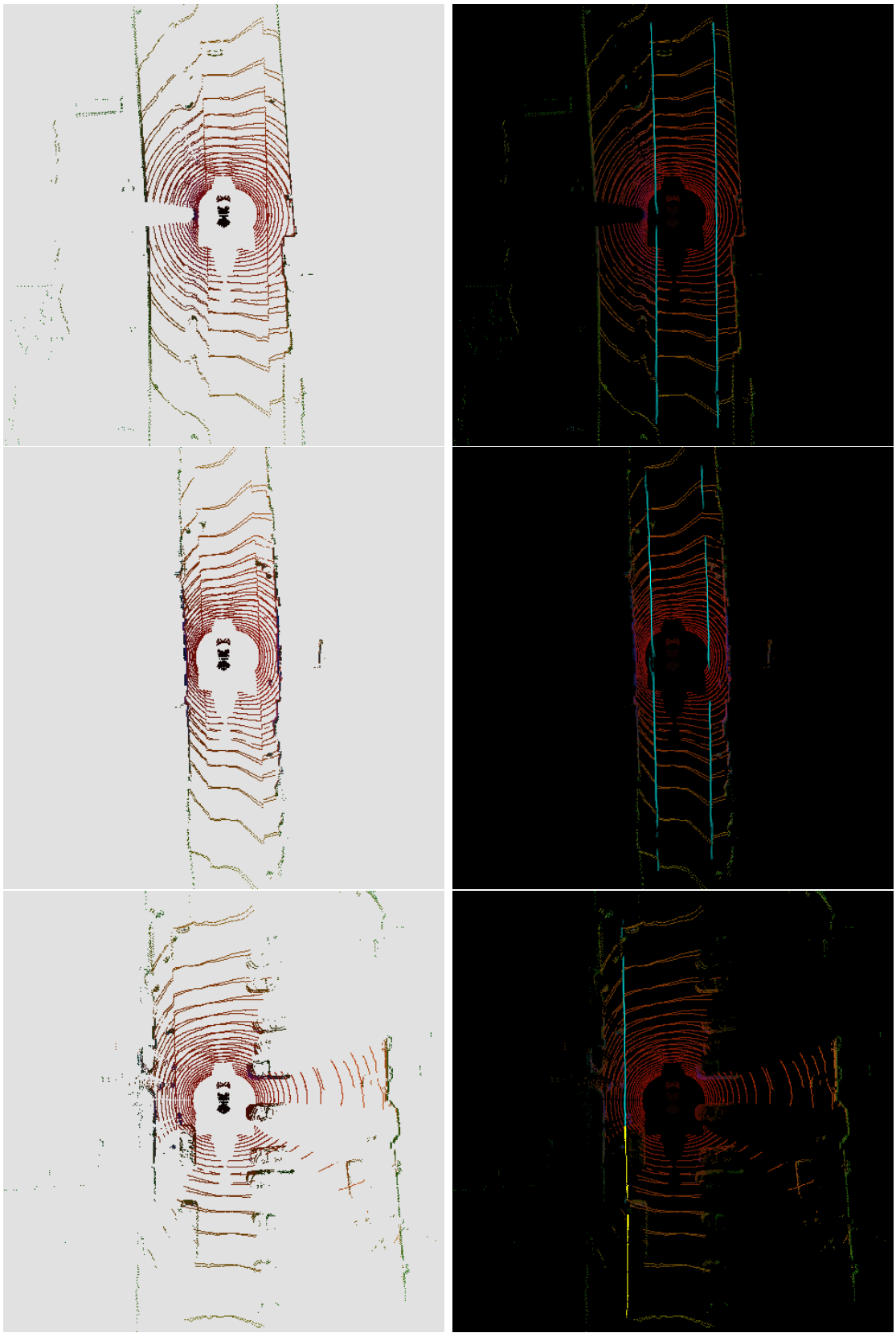


Figure B.1: Output samples of detected visible (cyan) and inferred occluded (yellow) road boundaries by the VRBD and ORBI models with an ROI of 48x48 metres.

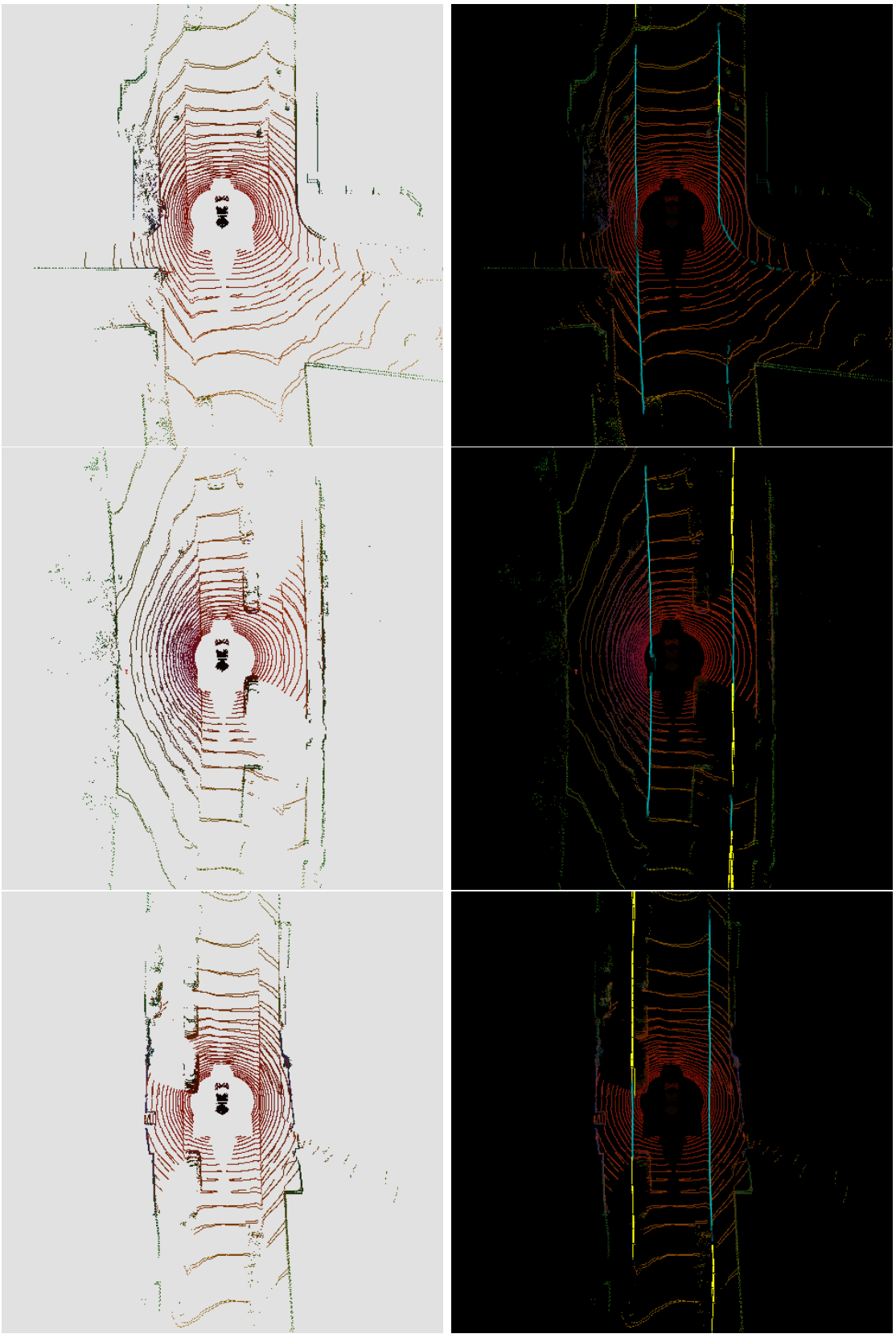


Figure B.2: Output samples of detected visible (cyan) and inferred occluded (yellow) road boundaries by the VRBD and ORBI models with an ROI of 48x48 metres.

C
ICP Matching Examples

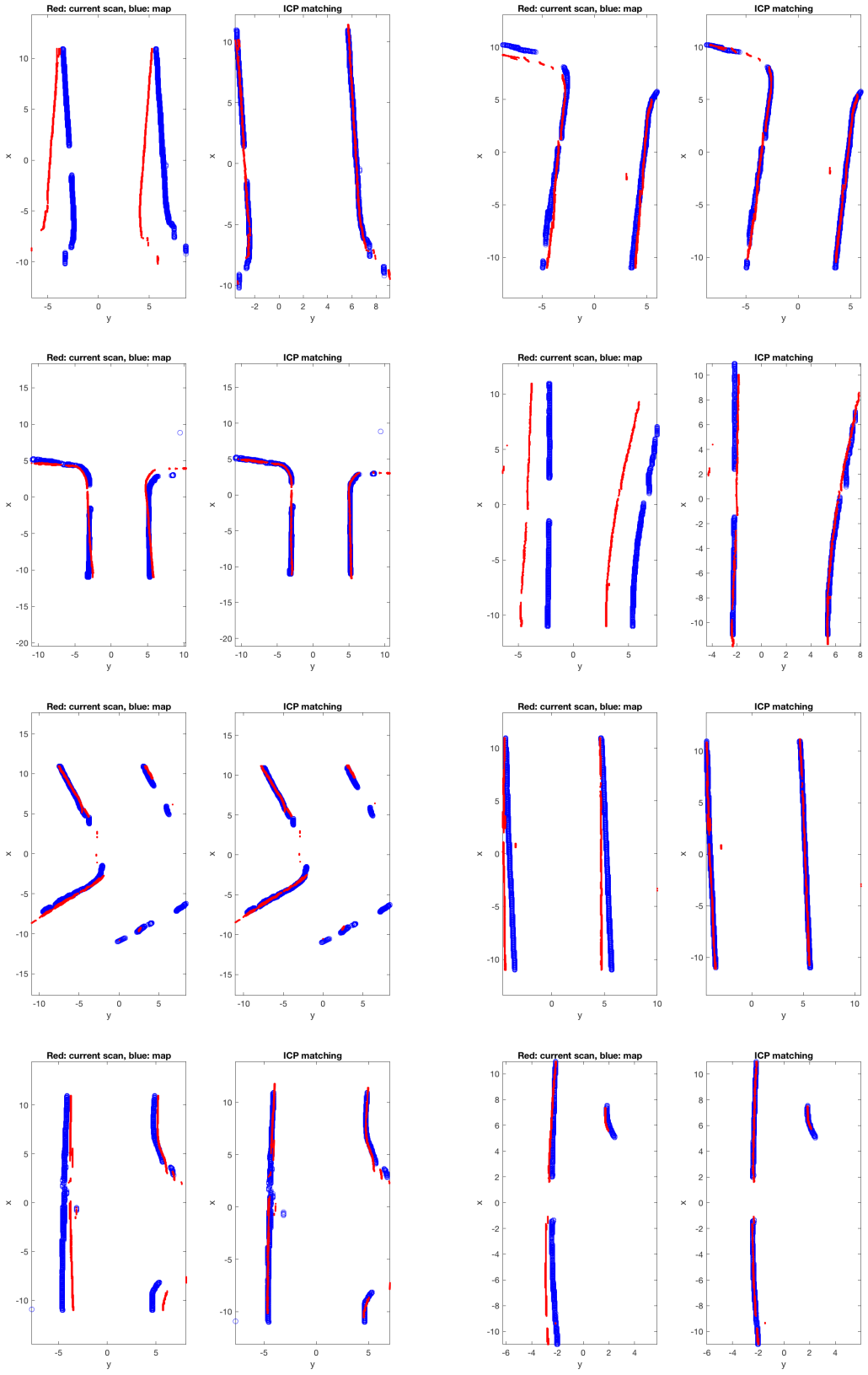


Figure C.1: Examples of road boundary based ICP matching for localisation.

References

[1] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. “Pulling things out of perspective”. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*. 2014, pp. 89–96.

[2] Will Maddern et al. “1 Year, 1000km: The Oxford RobotCar Dataset”. In: *IJRR* 36.1 (2017), pp. 3–15. eprint: `http://ijr.sagepub.com/content/early/2016/11/28/0278364916679498.full.pdf+html`.

[3] Tarlan Suleymanov et al. “The Path Less Taken: A Fast Variational Approach for Scene Segmentation Used for Closed Loop Control”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, South Korea, Oct. 2016.

[4] Timo Scharwachter and Uwe Franke. “Low-level fusion of color, texture and depth for robust road scene understanding”. In: *IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2015, pp. 599–604.

[5] Ben Upcroft et al. “Lighting Invariant Urban Street Classification”. In: *ICRA* (2014).

[6] Thomas Leung and Jitendra Malik. “Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons”. In: *IJCV* 43.1 (2001).

[7] Michael Tanner et al. “DENSER Cities: A System for Dense Efficient Reconstructions of Cities”. In: *ArXiv e-prints* (Apr. 2016). arXiv: `1604.03734 [cs.CV]`.

[8] R. Labayrade, D. Aubert, and J. -. Tarel. “Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation”. In: *Intelligent Vehicle Symposium, 2002. IEEE*. Vol. 2. June 2002, 646–651 vol.2.

[9] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[10] Pedro Piniés, Lina Maria Paz, and Paul Newman. “Dense and Swift Mapping with Monocular Vision”. In: *Int. Conf. on Field and Service Robotics (FSR). Toronto, ON, Canada*. 2015.

[11] Radhakrishna Achanta et al. *SLIC Superpixels**.

[12] Peter Kontschieder et al. “Structured Class-Labels in Random Forests for Semantic Image Labelling”. In: *ICCV* (2011).

[13] T. Suleymanov, L. Kunze, and P. Newman. “Online Inference and Detection of Curbs in Partially Occluded Scenes with Sparse LIDAR”. In: *The 22nd IEEE International Conference on Intelligent Transportation Systems*. Oct. 2019.

[14] T. Suleymanov, P. Amayo, and P. Newman. “Inferring Road Boundaries Through and Despite Traffic”. In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. Nov. 2018.

[15] Veronique Prinet et al. “3D road curb extraction from image sequence for automobile parking assist system”. In: *IEEE ICIP* (2016), pp. 3847–3851.

[16] M. Kellner et al. “Multi-cue, Model-Based Detection and Mapping of Road Curb Features Using Stereo Vision”. In: *IEEE ITSC*. Sept. 2015, pp. 1221–1228.

[17] L. Wang et al. “Multi-cue road boundary detection using stereo vision”. In: *IEEE ICVES*. July 2016.

[18] F. Oniga, S. Nedevschi, and M. M. Meinecke. “Curb Detection Based on a Multi-Frame Persistence Map for Urban Driving Scenarios”. In: *IEEE ITSC*. Oct. 2008, pp. 67–72.

[19] M. Kellner, M. E. Bouzouraa, and U. Hofmann. “Road curb detection based on different elevation mapping techniques”. In: *IEEE IV*. June 2014, pp. 1217–1224.

[20] J. Siegemund, U. Franke, and W. Förstner. “A temporal filter approach for detection and reconstruction of curbs and road surfaces based on Conditional Random Fields”. In: *IEEE IV*. June 2011, pp. 637–642.

[21] M. Enzweiler et al. “Towards multi-cue urban curb recognition”. In: *IEEE IV*. June 2013, pp. 902–907.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597.

[23] Xingang Pan et al. “Spatial As Deep: Spatial CNN for Traffic Scene Understanding”. In: *arXiv preprint arXiv:1712.06080* (2017).

[24] Wei Liu et al. “SSD: Single Shot MultiBox Detector”. In: *ECCV*. 2016.

[25] Ross B. Girshick. “Fast R-CNN”. In: *ICCV*. 2015.

[26] Paul Amayo et al. “Geometric Multi-Model Fitting with a Convex Relaxation Algorithm”. In: *IEEE CVPR*. Salt Lake City, USA, June 2018.

[27] Lars Kunze et al. “Reading between the Lanes: Road Layout Reconstruction from Partially Segmented Scenes”. In: *IEEE ITSC*. Maui, Hawaii, USA, Nov. 2018.

[28] Tom Bruls et al. “Mark Yourself: Road Marking Segmentation via Weakly-Supervised Annotations from Multimodal Data”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia, May 2018.

[29] Jay Earley. “An Efficient Context-free Parsing Algorithm”. In: *Commun. ACM* 13.2 (Feb. 1970), pp. 94–102. URL: http://doi.acm.org/10.1145/362007.362035.

[30] Sagi Katz, Ayellet Tal, and Ronen Basri. “Direct Visibility of Point Sets”. In: *ACM Trans. Graph.* 26.3 (July 2007).

[31] Giseop Kim, Byungjae Park, and Ayoung Kim. “1-day learning, 1-year localization: Long-term LiDAR localization using scan context image”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1948–1955.

[32] Sebastian Thrun et al. “Robust Monte Carlo Localization for Mobile Robots”. In: *Artificial Intelligence* 101 (2001), pp. 99–141.

[33] Farouk Ghallabi et al. “LIDAR-Based Lane Marking Detection For Vehicle Positioning in an HD Map”. In: *IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*. Maui, Hawaii,, United States, Nov. 2018.

[34] Chris Linegar, Winston Churchill, and Paul Newman. “Work Smart, Not Hard: Recalling Relevant Experiences for Vast-Scale but Time-Constrained Localisation”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA, May 2015.

[35] P. J. Besl and N. D. McKay. “A method for registration of 3-D shapes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (Feb. 1992), pp. 239–256.