

Situational Awareness in Autonomous Vehicles: Learning to Read the Road

Bonolo Mathibela
New College



Supervisor:
Professor Paul Newman

Robotics Research Group
Department of Engineering Science
University of Oxford

January 2014

Situational Awareness in Autonomous Vehicles: Learning to Read the Road

Abstract

This thesis is concerned with the problem of situational awareness in autonomous vehicles. In this context, situational awareness refers to the ability of an autonomous vehicle to perceive the road layout ahead, interpret the implied semantics and gain an awareness of its surrounding - thus reading the road ahead.

Autonomous vehicles require a high level of situational awareness in order to operate safely and efficiently in real-world dynamic environments. A system is therefore needed that is able to model the expected road layout in terms of semantics, both under normal and roadwork conditions.

This thesis takes a three-pronged approach to this problem: Firstly, we consider *reading the road surface*. This is formulated in terms of probabilistic road marking classification and interpretation. We then derive the road boundaries using only a 2D laser and algorithms based on geometric priors from Highway Traffic Engineering principles. Secondly, we consider *reading the road scene*. Here, we formulate a roadwork scene recognition framework based on opponent colour vision in humans. Finally, we provide a data representation for situational awareness that unifies reading the road surface and reading the road scene. This thesis therefore frames situational awareness in autonomous vehicles in terms of both static and dynamic road semantics - and detailed formulations and algorithms are discussed.

We test our algorithms on several benchmarking datasets collected using our autonomous vehicle on both rural and urban roads. The results illustrate that our road boundary estimation, road marking classification, and roadwork scene recognition frameworks allow autonomous vehicles to truly and meaningfully read the semantics of the road ahead, thus gaining a valuable sense of situational awareness even at challenging layouts, roadwork sites, and along unknown roadways.

Statement of Authorship

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

Bonolo Mathibela, New College

Funding

Bonolo Mathibela is very grateful to the numerous bodies that funded this research. The work reported in this thesis was funded by:

- The Rhodes Trust, Oxford
- Department of Engineering Science Bursary, Oxford
- New College Graduate Scholarship, Oxford
- The National Research Foundation (NRF), South Africa

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Acknowledgements

First and foremost, thanks are due to my supervisor, Professor Paul Newman for his advice, time, direction, and guidance during my DPhil studies.

I would also like to thank Dr Ingmar Posner for his advice and time, particularly in the early years of my studies. Also, thanks go to the members of the Mobile Robotics Group (MRG) who made the lab a stimulating environment to be.

Thanks to Barry Houter for providing the Geograph image dataset (under the Creative Commons License).

I would like to especially thank my friends - Daniel John, Hasna Zaher, Mark Irwin and Victor Prisăcariu - for their encouragement and for sharing both the joys and frustrations of the DPhil journey with me.

Finally, but most importantly, thanks to my family, especially my father (Andrew), mother (Julia) and brother (Mpho) for their unfailing support, for always believing in me, for their many selfless sacrifices and for cheering me on throughout my doctoral studies. My parents led by example, showing me the path to true freedom through education and developing a passion for learning.

I came to truly understand, during my four years spent at Oxford, that “The fear of the LORD is the beginning of wisdom, and knowledge of the Holy One is understanding.” - Proverbs 9:10.

What an incredible journey it’s been!

To God be the glory!

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Outline of the Problem	2
1.3	Contributions	3
1.4	Structure of the Thesis	5
1.5	Publications	5
2	Background and Context	6
2.1	Situational Awareness in Mobile Robotics	6
2.2	Data Representation for Situational Awareness	7
2.2.1	Google Protocol Buffer Representation	7
2.2.1.1	Protocol Buffer Message Type: Road Markings	9
2.2.1.2	Protocol Buffer Message Type: Road Boundaries	10
2.2.1.3	Protocol Buffer Message Type: Roadworks	11
3	Reading the Surface: Road Marking Classification	12
3.1	Literature Review	12
3.1.1	Road marking arrows and painted road signs	12
3.1.2	Boundary type markings	14
3.1.3	Zebra crossings and intersections	14
3.2	Representation of Road Markings	16
3.2.1	Seven Road Marking Classes	17
3.2.2	Line Segment Representation	18
3.3	Inverse Perspective Mapping	19

3.4	Conditional Random Fields: Road Marking Classification	21
3.4.1	Conditional Random Fields	22
3.4.2	Graphical Structure	23
3.4.3	Feature Functions	24
3.4.3.1	Visual Feature Functions	24
3.4.3.2	Unary Score Features	28
3.4.3.3	Binary Features	30
3.4.4	Parameterisation	30
3.4.5	Parameter Estimation	31
3.4.5.1	Conditional Log Likelihood Training	31
3.4.5.2	Variational Inference	32
3.5	Performance Evaluation	32
3.5.1	Data Sets	32
3.5.2	Results	34
3.5.2.1	No Relational Learning	34
3.5.2.2	Classification with Relational Learning	35
3.5.2.3	Road Scene Semantics from Road Markings	37
3.5.2.4	Understanding Misclassification	43
3.6	Chapter Summary	45
4	Reading the Surface: Road Boundaries	46
4.1	Literature Review	46
4.1.1	Vision Based Approaches	47
4.1.2	Clothoid Models	48
4.1.3	LIDAR and Velodyne Based Approaches	48
4.2	Domain Knowledge: Geometric Priors	51
4.2.1	Cross-Sectional Elements	52
4.2.2	Vertical Alignment	53
4.2.3	Horizontal Alignment	55
4.2.3.1	Simple Curves	56
4.2.3.2	Compound, Reverse and Transitional Curves	57
4.3	Sensor Configuration	58

4.3.1	Absolute Sensor Calibration	59
4.4	Road Boundary Point Detection	61
4.4.1	Surface Segmentation	61
4.4.2	Feature Extraction	66
4.4.3	Boundary Point Detection	68
4.5	Parametric Models for Representing Road Boundaries	73
4.5.1	Implicit versus Parametric Form	73
4.5.2	Interpolation versus Approximation	74
4.6	BaryArc Model for Representing Road Boundaries	74
4.6.1	Geometry	75
4.6.1.1	Bulge	75
4.6.1.2	Barycentric Coordinates	76
4.6.2	BaryArc Derivation	77
4.6.3	Differentiability	79
4.6.4	Estimating the BaryArc weight w_1^2	79
4.6.5	Estimating the Road Boundary Transition Points	80
4.6.6	Plotting a BaryArc curve	82
4.7	Performance Evaluation	83
4.7.1	Data Sets	83
4.7.2	Results	84
4.7.2.1	Centre Line Continuity	86
4.7.2.2	Mean Squared Error for BaryArc	88
4.7.2.3	Road Curvature	88
4.7.2.4	Intersections	90
4.7.2.5	Road Boundary Estimation in Challenging Scenarios	90
4.7.2.6	Comparison	93
4.7.2.7	Overall System Performance	99
4.8	Chapter Summary	99
5	Reading the Scene: Roadwork Scene Recognition	106
5.1	Literature Review	106
5.1.1	Roadwork Object Recognition	107

5.1.2	Roadwork Scene Recognition	109
5.2	Understanding Roadworks	112
5.2.1	Colour in Scene Recognition	112
5.3	The Opponent Colour Model	115
5.3.1	Colour Vision in Humans	115
5.3.2	Roadwork Scenes and the Opponent Colour Model	120
5.4	An Opponent Colour Roadwork Scene Signature	122
5.4.1	Opponent Colour Spatial Prior	122
5.4.2	Image Descriptor Extraction	122
5.4.3	Codebook Generation and Vectorization	124
5.4.4	Classifier Training and Validation	126
5.5	Performance Evaluation	128
5.5.1	Data Sets	128
5.5.2	Results and Comparisons	128
5.6	Chapter Summary	134
6	Closure: Situational Awareness	135
6.1	Reading and Writing Data	135
6.2	Message Type Implementation	136
6.2.1	Road Markings	136
6.2.2	Road Boundaries	137
6.2.3	Roadworks	137
7	Discussion	138
7.1	Summary of Contributions	138
7.2	Future Work	139
7.3	Concluding Remarks	140
A	Geometric Design of Roads	141
A.1	Design Vehicle, Speed and Hourly Volume	141
A.2	Cross-Sectional Elements	142
A.3	Vertical Alignment	143

List of Figures

1.1	Typical complex road semantics (from left to right): worn out or ambiguous road markings, roadworks changing road layouts and worn tarmac with patched potholes.	3
3.1	An example of Inverse Perspective Mapping (IPM) to remove the perspective effect in images. Notice how, in the IPM images, parallel lines no longer intersect at the vanishing point.	16
3.2	Zig-zag approach road markings lead up to a traffic light controlled pedestrian crossing. .	17
3.3	Line segment representation of road markings. The line segment is the geometric model of a road marking, parameterised by endpoints p_A and p_B . The midpoint $p_C \triangleq \frac{(p_A + p_B)}{2}$ represents the node location for our graphical structure.	19
3.4	The frontal pinhole camera model showing how points in the real 3D world are projected into the 2D camera imaging plane. Specifically, for world point $\mathbf{p} = [X \ Y \ Z]^T$ and image point $\mathbf{x} = [x \ y]^T$ it is straightforward to show, using similar triangles, that $\mathbf{x} = [x \ y]^T = \frac{f}{Z} [X \ Y]^T$ [53].	20
3.5	CRF model, $p(\mathbf{x} \mathbf{z})$, for road marking classification. The hidden states $\mathbf{x}_i$ are the unknown road marking classes and the observations $\mathbf{z}_i$ (shaded orange) are line segment endpoints extracted from each road marking.	24
3.6	Some graph structures of road marking segments created using Delaunay Triangulation. Notice how each example (image) gives rise to a different graphical structure since the number of road markings varies. The node locations are taken to be the midpoint of each road marking segment (see Figure 3.3).	25
3.7	A simple example of a decision tree with two predictor variables x_1 and x_2 : the root node is shaded orange and the leaf nodes are shaded green. The blue node is a condition node. .	29

3.8	Data collection vehicle: the vehicle is shown with a camera rig mounted on the roof [24].	33
3.9	Histogram showing the training and test set class distributions of Tables 3.2 and 3.3. . . .	34
3.10	Confusion matrices for our test set produced using only the RUSBoost classification framework (no relational information)	35
3.11	Confusion matrices for our test set produced using a relational learning framework	36
3.12	Special lane and intersection type road markings can often have very similar visual appearance. Both these classes are marked by short repetitive road markings.	37
3.13	Classification results where each colour represents a different class (see Table 3.1). The graphical model structure is also superimposed on each image.	38
3.14	Classification results from our system (green lines are correctly classified, whilst red lines represent misclassifications). These images correspond to Figure 3.13.	39
3.15	Examples of zig-zag approaches correctly classified by our algorithm. Our classification framework achieves a prediction accuracy of 88% for identifying zig-zag approaches and hence anticipating upcoming traffic light controlled pedestrian crossings.	40
3.16	The two types of road markings we class as “boxed junctions”. These describe areas of the road that a vehicle should not be stationary in (but possibly quickly traverse) such as separator boxed junctions (left) and intersection boxed junctions (right).	41
3.17	Examples of boxed junctions correctly classified by our algorithm. Our classification framework achieves a prediction accuracy of 86% for identifying boxed junctions and hence finding areas of the road where no stopping or traversal is allowed.	42
3.18	Understanding misclassifications. The colours represent road marking classes according to Table 3.1. Refer to the text for explanations.	44
3.19	Worst case misclassifications. The red markings are misclassified and the green markings are correctly classified. Refer to the text for explanations	44
4.1	Road surface segmentation and curb point detection: (a) Detection of the road surface using three sliding windows from either side using: $\delta h_{pavement} \leq \delta z_1 \leq \delta h_{filter}$, $\delta z_2 \leq \delta h_{pavement}$ and $\delta z_3 \leq \delta h_{pavement}$ where z_i is the height range of the points in the i^{th} sliding window. (b) Curb points are detected using two 1D sliding windows and testing for the elevation ($\delta z_{curb1} \leq \delta h_{pavement}$ and $\delta z_{curb2} \geq \delta h_{curb}$), point density ($Length_2 < L < Length_1$) and slope changes ($\delta angle = abs(\alpha_1 - \alpha_2)$ and $\delta angle = \delta \theta$). These parameters are manually tuned by [153] to give good results. Image reproduced from [153].	51

4.2	Three possible 2D SICK LIDAR and 3D Velodyne sensor configurations. The left vehicle shows a forward facing SICK LIDAR sensor which is well suited for object detection and mapping applications. This is by far the most commonly adopted configuration. The central vehicle shows our vertically downward facing SICK LIDAR configuration where the scanning plane of the sensor is perpendicular to the ground plane. This unique configuration is seldom adopted in previous work - we therefore develop new feature detection algorithms for this case. The right vehicle shows a Velodyne sensor which scans around the vehicle generating a complete 3D point cloud. Although the cost of Velodynes is decreasing, they remain prohibitively expensive for low cost autonomous driving applications.	52
4.3	Typical cross sections of British roads: a rural all purpose dual carriageway bounded by grass (left) and an urban road bounded by curbs (right). Images taken from [4], Volume 6.	53
4.4	Typical features accommodated in cross sections. HS = hard shoulder and H/S = Hard strip. Image from [4], Volume 6. Regions between verges or footways represent areas to be bounded on the left and right by the continuous curves computed by our road boundary estimation algorithm.	54
4.5	Typical cross section for a two lane motorway: the gradients shown are of the form (horizontal:vertical) and the red dot represents the road centre line. Reproduced from [44].	55
4.6	Typical vertical (leftmost) and sloping curbs (right) found on highways. Dimensions are shown in inches. Diagrams reproduced from [44].	55
4.7	The alignment geometry for roads (simple curve) taken from [44].	57
4.8	A compound curve with two radii R_1 and R_2 and common tangent at C (left). A reverse curve with parallel tangents (centre) and an example of a clothoid (transitional curve) with changing radius (right).	58
4.9	The location of the 2D SICK LIDAR (yellow dot) and the GPS/INS unit (green dot) on our autonomous vehicle; and the laser scanline (shown in red). To create a 3D point cloud as the vehicle traverses the environment, a transformation matrix $\mathbf{T}_{\text{INS-LASER}}$ from the laser coordinate frame to the vehicle's INS coordinate frame is estimated. The red lines show some of the vertical scan lines from the laser which are approximately perpendicular to the ground plane. The road surface model and Old Factory Building is taken from Google Sketchup (Trimble 3D Warehouse)[145][144].	60

4.10	A 3D point cloud created using $\mathbf{T}_{\text{INS-LASER}}$. The segmented road surface (yellow), road boundary points (red) and road markings (blue) are results from our algorithms. The bottom row shows the corresponding images for this section of road. Notice how even pedestrian intersections are correctly segmented as road surface points.	60
4.11	(a) A projection of the laser sensor data (raw output from our boundary estimation algorithm) at Begbroke Science Park onto Google Maps, resulting from the transform $\mathbf{T}_{\text{G-LASER}}$. The blue, magenta and yellow points represent the detected road boundaries, road markings and vehicle GPS trajectory. (b) The 3D point cloud generated from the transformation matrix $\mathbf{T}_{\text{INS-LASER}}$. Corresponding points between the two figures are marked A and B.	61
4.12	A block diagram showing the main modules making up our road boundary point detection system: Surface Segmentation, Feature Extraction and Boundary Point Detection.	62
4.13	Estimating the vehicle pitch angle, θ . (a) A single LIDAR scanline where the red dot represents the known sensor origin and the green points are the ground points considered for approximating the pitch angle. (b) A magnified view of (a). (c) A close up simplified schematic where point C is the sensor origin and AB is the line fitted to the local ground points. The vehicle pitch angle is given by $\angle ABD$	63
4.14	Examples showing the estimation of vehicle pitch using a small region (between the dashed vertical lines) around the sensor origin (0,0) to estimate the point cloud rotation matrix $\mathbf{R}_{\text{p}}$. The blue dots show the original scanlines and the green dots are the resulting pitch corrected laser scanline outputs.	63
4.15	Minimum Spanning Tree Clustering for groundstrike estimation: each laser point (circle) is a node in the graph and the red lines are the connected clusters after the clustering threshold is applied. Only 2 connected clusters (with more than one member) remain and the largest one is the groundstrikes.	65
4.16	Single scanlines from the LIDAR sensor (each laser scan point is shown as a dot). The red region is the output of our groundstrike estimation algorithm based on Greedy Clustering. Notice how trees, buildings, bushes and other vehicles are excluded from the groundstrike regions, and most of the points in a scanline belong to the ground region which is a special property of our sensor configuration.	65
4.17	A 3D model of a road with and without curbs with road boundary points superimposed in pink. Model taken from Google Sketchup (Trimble 3D Warehouse)[145]	66

4.18	A comparison of our Range Filter output (left) to a derivative operation output (right). Notice how the derivative captures the noise characteristics of the sensor and is therefore not a good candidate for reliable feature extraction. Our Range Filter approach however, is able to detect curbs whilst suppressing most of the noise.	67
4.19	The output of our range filter is the feature function $f_{range}(z)$. (a) A scanline with two curbs (b) No curbs present but there is an object at the edge of the road	67
4.20	Tangent Normals (black arrows) on ground plane points (green). Normals on the curbs differ in direction from those on the ground. The direction magnitude feature function, $f_{direction}(y, z)$ is plotted in blue. The left and right portion of the road is split at the known sensor origin point, hence the two separate plots for a single scanline.	68
4.21	(a) The functions $f_{direction}(y, z)$ and $\alpha f_{range}(z)$ are shown in red and green respectively. (b) The transformed feature functions f_d (red) and f_r (blue) and their combination F (black) are shown. F is the input to our change point detection step.	69
4.22	To assign confidence levels to our road boundary estimates (where curbs are present), we perform change point analysis on the combined feature function F shown on the left. On the right, we show how the associated confidence level for each detected change point is computed using a bootstrapping technique. For clarity, only the first 10 bootstrapping results are shown in red and the peaks of the cumulative sums (blue) denote the detected change points. The principle behind this bootstrapping procedure is that by randomly sampling F without replacement, the resulting cumulative sums capture the behaviour of this feature function supposing no change occurred. Thus, if a significant change occurs, the bootstrapping cumulative sums (shown in red) are closer to zero than the original function cumulative sum (shown in blue).	71
4.23	Top: Final road boundary point estimation outputs for a single scanline where (a) no curbs are present and (b) both curbs are present. (c) The final road boundary points detected over a sequence of laser scans, shown in magenta on the corresponding 3D point cloud. Point correspondences, A and B, are also shown.	72
4.24	Representation of arc for road curvature ($\theta = \theta_1 = \theta_2$).	75

4.25	The bulge estimation procedure: Points C and B are the known endpoints of the green line. The road boundary detected points are projected onto the green line and the lengths h (perpendicular) and t (parallel) of Figure 4.24 are shown as dark blue lines. The bulge parameter is given by $b = \frac{h}{t}$.	80
4.26	The Wildcat autonomous car is equipped with front and rear (not shown) vertically downward facing SICK LMS1xx range sensors and mono and stereo (bumblebee) cameras. The differential OxTS GPS/INS unit and rotating 3D laser (not Velodyne) are mounted on the vehicle rooftop.	84
4.27	The operating range diagram of the SICK LMS1xx sensor on the Wildcat autonomous car. Figure taken from [64].	85
4.28	Dataset 1: Left: The road center line determined from the detected road boundaries. Right: The x and y position of the center line as a function of distance travelled. The repetitions result from when the vehicle drove over the same section of motorway (between two roundabouts) multiple times. The identical repetitions show that the center line is consistently estimated correctly each time.	87
4.29	Dataset 2: Left: The road center line determined from the detected road boundaries. Right: The x and y position of the center line as a function of distance travelled. The repetitions arise from repeated traversals of sections of the road through rural, urban and residential areas.	87
4.30	The cumulative distribution function of the Mean Squared Error for the (a) left road boundary in Dataset 1 (b) right road boundary in Dataset 1 (c) left road boundary in Dataset 2, and (d) right road boundary in Dataset 2	89
4.31	Dataset 1: Our estimated BaryArc road boundaries plotted on Google Maps, where the boundary colour represents the curvature values (see colour bar). Here, semantics such as roundabouts are automatically detected as regions of high curvature. The extracted OpenStreetMap intersections are also shown in the bottom right image (red crosses).	91
4.32	Dataset 2: Detected BaryArc road boundaries plotted on Google Maps, where the boundary colour represents the road curvature values.	92
4.33	Dataset 2: The extracted OpenStreetMap intersections shown as red crosses along the vehicle path	93

4.34	Dataset 1: (a) Roadworks found in Dataset 1 are outlined by traffic cones and traffic signs that occlude the laser scanner's view of road boundaries. Furthermore, the placement of traffic cones varies in density. (b) Left: Successful road boundary detections despite the presence of roadworks (traffic cones). Right: successful road boundary detection when another vehicle is on the road and traffic cones are present. (c) Two failure cases. Left: The presence of several vehicles on the road results in inaccurate boundary estimation. Right: Densely placed traffic cones result in our algorithm falsely detecting the cones as road boundary locations. (d) Left: The output of our road boundary point estimation algorithm (blue dots). Here, the sensed traffic cone points in the middle of the road were successfully ignored by our BaryArc boundary estimation algorithm (magenta curve). Right: The corresponding 3D output showing correctly estimated road boundaries in the presence of roadworks.	94
4.35	Dataset 2: (a) Incorrect road boundary estimates due to a series of parked vehicles. (b) Driveways are level with the road surface thus they are also segmented by our road surface detection algorithm (red shading). However, even in this challenging scenario, the correct road boundaries are still estimated because the driveway points are detected as outliers by our algorithm. (c) Correct road boundary estimation even when other vehicles are present on the road.	95
4.36	Road section considered for comparison experiment. Notice that the curb geometry narrows to accommodate a small pedestrian crossing and widens to accommodate temporary parking in front of the building.	96
4.37	Overhead view of the road boundary points detected using (a) our algorithm and (b) the competing algorithm. The changing curb geometry is more accurately captured using our method.	97
4.38	The final road boundary estimates using (a) our algorithm and (b) the competing algorithm. The changing curb geometry (increase and decrease of the road width) is accurately captured by our method.	98

4.39	Dataset 1 results. Left: The laser point cloud under consideration, coloured by intensity. Middle: Projected 2D BaryArc Model (magenta) fit for road boundaries with the raw laser estimated boundary points (blue dots) and curve transition points (black crosses) from Section 4.6.5 shown. Right: The final output of our algorithm is the road surface segmentation (red) and the BaryArc road boundary estimation (green).	100
4.40	Dataset 2 results. Left: The laser point cloud under consideration, coloured by intensity. Middle: Projected 2D BaryArc Model (magenta) fit for road boundaries with the raw laser estimated boundary points (blue dots) and curve transition points (black crosses) from Section 4.6.5 shown. Right: The final output of our algorithm is the road surface segmentation (red) and the BaryArc road boundary estimation (green).	101
4.41	Road boundary estimation results for Dataset 1. The road surface segmentation results are shown in red and the BaryArc boundaries in green.	102
4.42	Road boundary estimation results for Dataset 1. The road surface segmentation results are shown in red and the BaryArc boundaries in green.	103
4.43	Road boundary estimation results for Dataset 2. The road surface segmentation results are shown in red and the BaryArc boundaries in green.	104
4.44	Road boundary estimation results for Dataset 2. The road surface segmentation results are shown in red and the BaryArc boundaries in green.	105
5.1	The typical flow of traffic sign detection systems: the segmentation step identifies image regions potentially containing traffic signs. The detection steps identifies candidate traffic sign blobs which then pass through a classification step to assign a class label to the sign. Finally, classifications are refined by a tracking step which is for temporal smoothing of predictions.	108
5.2	Some of the workmen signs considered in roadwork zone recognition work done by [123]. Image taken from source.	111
5.3	Typical roadwork objects (from left to right): Construction sign, merge sign, traffic light ahead sign, keep left sign, keep right sign, plastic barrier, traffic cone, traffic light, concrete barrier and beacon. The occurrences of such objects is typically exploited for roadwork scene recognition tasks.	111

5.4	Our proposed autonomous vehicle interface for situational awareness at roadwork sites: when roadworks are detected (top), manual control is offered to the driver (bottom). The roadworks shown here were successfully recognised using our roadwork scene signature.	113
5.5	A diagram showing the cross-sectional elements of the human eye. Image taken from [29].	115
5.6	(a) Spectral sensitivity functions for cone (ρ, γ, β) photoreceptors. (b) The generalised spectral luminous efficiency functions where $V(\lambda)$ and $V'(\lambda)$ represents the weighted average of the three cone and the rod spectral response, respectively. Image taken from [46].	116
5.7	A simplified schematic of the Hering based opponent colour model for human vision proposed by Hurvich and Jameson [62]. The polarities indicate two possible modes of responses that are physiologically opponent in nature. The three pairs B-Y, G-R, and W-Bl are antagonistic (i.e. the concept of red-green is foreign to the human visual system, while yellow-red is a well perceived colour - orange). Here, α, β, γ are photosensitive biological retina photopigments aiding the formation of colour.	117
5.8	The afterimage effect which can only be fully explained by the opponent colour process in human vision: if you stare at the dot in the middle of the coloured pattern for about one minute and then quickly switch your gaze to a homogenous sheet of white paper, you will see the opponent colour pairs appearing in place of the squares in the afterimage.	118
5.9	The opponent chromatic response functions measured by Hurvich and Jameson in [67], to derive quantitative measures for opponent colour vision in humans. For this observer, the yellow, green and blue hues appear unique (at the arrows). Deriving the unique red hue at 700nm requires a mixture with blue to cancel out the yellow there (hue cancellation). The chromatic valence polarities in the figure illustrate the opposing nature of this colour space and are arbitrarily assigned. Image adapted from [67].	119
5.10	Typical roadwork scenes found in our datasets. The dominant co-occurrences of red-green and blue-yellow hues in the roadwork scenes inspire our opponent colour vision formulation of a roadwork scene signature.	120
5.11	Left: The axes of the CIELab opponent colour space. Right: Automatic pixel labelling characteristics: the red-green pattern is highlighted by channel a (bottom left), with the opponent colour intensities at opposite extremes. Similarly, the blue-yellow pattern is highlighted by channel b (bottom right). The black-white pattern is highlighted by the intensity (achromatic) channel.	121

5.12	Results from our opponent colour spatial prior showing the automatic labelling of pixels in CIELab channels a and b. The intensity scale is shown at the bottom of the figure. Notice how blue traffic signs, and barriers/traffic cones are well segmented in channels b and a respectively.	123
5.13	Our image descriptor comprises of a 128 bit SURF descriptor [6] where the scale and orientation of the SURF descriptor is denoted by the green circle. A descriptor is computed at each keypoint in the image for both opponent chromatic colour channels, a and b. . .	124
5.14	An illustration of our soft assignment vectorization for the simple case of a codebook $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ with six visual word centroids. Three descriptors (shown as crosses) were detected in the image. The histogram on the right shows the final vectorized image representation. Notice how the descriptor with the shortest distance from the centroid receives a higher weighting in the vectorized histogram representation (see Equation 5.3). This allows us to capture the distribution of the descriptors with respect to the nearest neighbour centroid (or visual word) rather than incrementing the histogram with the same value for each nearest neighbour regardless of similarity.	125
5.15	Examples of visual words (detected SURF descriptors from both opponent colour channels) superimposed on roadwork scenes found in our datasets. The SURF threshold remains constant throughout. (a) Visual words extracted without exploiting the spatial prior of the opponent colour model. (b) Visual words extracted using our opponent colour based formulation. Notice how significantly fewer visual words are extracted by incorporating the opponent colour model and importantly, these visual words are only extracted in regions most likely to contain roadwork objects.	127
5.16	The number of SURF keypoints detected without using the opponent colour spatial prior is significantly higher than when the transform is applied. Although fewer descriptors are extracted by our opponent colour framework, these are located at relevant regions in the image, thus accuracy is not compromised (see Figure 5.15).	129
5.17	In general, fewer keypoints are detected for negative examples which demonstrates the visual saliency of roadworks i.e. our descriptor focuses on roadwork objects highlighted by our spatial prior, while suppressing non-opponent noise.	129

5.18	ROC curves showing the TPR and FPR achieved for the datasets described in Table 5.1. Each curve represents a different dataset collected using our autonomous vehicle. The Area Under Curve (AUC) for each dataset is also shown. Good discriminative performance is achieved across all datasets.	130
5.19	ROC curves showing the TPR and FPR achieved for the datasets described in Table 5.1 without the opponent spatial prior. Each curve corresponds to Figure 5.18. The Area Under Curve (AUC) for each dataset is also shown. The performance is significantly worse without the opponent colour prior.	131
5.20	Some examples of traffic cone and keep right sign detector outputs.	132
5.21	A comparison showing ROC performance using our roadwork scene signature compared to roadwork object detection within a Naive Bayes scene recognition framework. Our method outperforms the naive approach and has the added advantage of not explicitly modelling individual roadwork objects in the scene. Dataset 3 was used.	133
A.1	Technical drawing showing the embankments (channels formed by curbs) taken from [3], Volume 3.	142
A.2	(a) A vertical crest curve. The line of sight for the crest curve is shown in red, given a driver at P of height H and an object h located at point Q. (b) A vertical sag curve. The headlight beam is shown in red for a driver of height H. In practice, standard heights are used for H and care should be taken to design roads with adequate sight distance S, the projection of the line of sight or headlight beam onto the horizontal, to ensure safe travel.	143

List of Tables

3.1	Seven road marking classes we classify in our framework (examples shown in Figure).	18
3.2	Training set class distribution.	33
3.3	Test set class distribution.	33
3.4	Percentage improvements in terms of precision and recall when relational learning is used.	36
4.1	Parameters of road boundary point detection algorithm.	73
4.2	Vehicle GPS trajectory for Dataset 1 and 2 shown on Google Maps. The trajectories total approximately 32km and are described in the Table.	85
5.1	A summary of the datasets used for training and evaluation.	128
A.1	Minimum design speeds for rural collector roads. Reproduced from [44].	141

Chapter 1

Introduction

1.1 Motivation

The overall aim of this thesis is to contribute to the development of semantic understanding and situational awareness in autonomous road vehicles. For several decades now, situational awareness has been limited to lane detection and estimation, traffic signal recognition, obstacle recognition, and lane departure warning systems. Although such systems offer a degree of situational awareness, they fail to model the complexities of road surfaces, the underlying geometric models underpinning the design of road layouts, and dynamic environments such as roadwork sites, where prior assumptions about road layout fail. The intelligent transport community continues to make steady progress, however the critical levels of robustness required for functioning in unconstrained environments has yet to be attained. Modelling complex semantics of the road will open up a huge field of possibilities in semantically driven navigation, local scene understanding and interpretation, and adaptation to road dynamics - leading to lifelong autonomy in robotic vehicles. This is the goal we pursue.

In particular, this thesis is concerned with *reading the road* by directly modelling useful road semantics in both static environments and dynamic roadwork conditions. To be truly autonomous, a vehicle should be able to detect the road surface and interpret interesting features present on the road such as road markings, road boundaries, intersections and objects like traffic cones, traffic signs, barriers and entire roadwork scenes. Great care has been taken by National Highway Authorities worldwide to formulate geometric models for road design. Such models dictate important considerations such as the alignment, superelevation, design speed, maximum volume capacity and drainage elements of roads. Highway Authorities also publish

technical reports detailing the meanings and correct placements of road markings, traffic signs, signals and cones in various typical scenarios. Such reports are essentially manuals for human drivers detailing how to properly read the road scene ahead. An autonomous vehicle however must be taught how to read road semantics by deriving a geometric road model and its associated semantic information in order to gain a sense of awareness of the driving situation in which it finds itself.

This thesis aims to create a unified, semantically complete situational awareness framework for autonomous vehicles and its associated data representation. Our approach goes beyond lane estimation and allows autonomous vehicles to automatically construct a model of the road layout and to infer meaningful semantics about the road scene: road surface markings, road boundaries, intersections, boxed junctions, pedestrian crossings, roundabouts and roadworks.

1.2 Outline of the Problem

Before formulating our problem statement, let's begin by briefly imagining a scenario where a motorist is driving home, from work, along an urban road. The motorist's ability to safely navigate home is aided by three things: a prior mental map of the road layout between home and work, an understanding of traffic signal posts and an awareness of his surroundings (other vehicles, pedestrians and road maintenance or construction). This prior map is a mental route representing an accumulation of many traversals of this same route over the months. This map is however not static, since temporary roadworks due to road maintenance might cause the motorists to take an alternative street or obey the new temporary road markings and signs around the roadworks. Furthermore, while driving home, the motorist obeys the rules dictated by the road markings and traffic signal posts he encounters and is careful to give way to other motorists or pedestrians along the way. In other words, the motorist remains in a constant state of semantic situational awareness whilst driving.

Now consider for a moment this same scenario, but this time the motorist is replaced by an autonomous vehicle. Maintaining situational awareness (while performing the simple task of driving home from work) now becomes a very difficult problem, and one that is of interest to the mobile robotics community. Concretely, the problem that will be addressed in this thesis is the following:

“How can an autonomous vehicle automatically discover and interpret road semantics in order to truly learn to read the road ahead, thus gaining a sense of what’s going on (situational awareness)?”



Figure 1.1: Typical complex road semantics (from left to right): worn out or ambiguous road markings, roadworks changing road layouts and worn tarmac with patched potholes.

This is a hard problem for a number of reasons. Firstly, the world is dynamic. Although the overall geometric road layout largely remains constant, the presence of other vehicles and roadwork construction dramatically alter the road layout. This poses a challenge both when building a road layout model and when using such a layout as a prior for navigation. Secondly, environmental conditions such as poor illumination, camera over exposure, noisy LIDAR intensity and range returns, and wet road surfaces make accurate feature extraction a challenging problem. Thirdly, roads deteriorate over time resulting in ambiguous or faded road markings which again violate assumptions about the physical road layout. Figure 1.1 illustrates some challenging scenarios found in our datasets. Without situational awareness, driverless vehicles are prone to accidents such as a recent one in Adelaide where a driverless car crashed into a restaurant, injuring a woman in the process [97].

In this thesis, learning to read the road ahead includes formulating probabilistic models for road marking classification, road marking scene interpretation, road boundary estimation based on Highway Traffic engineering geometric road models, and roadwork scene recognition based on neuroscience and biological vision principles.

1.3 Contributions

Situational awareness from road semantics can be decomposed into several sub-problems, each with their own challenges. Feature detectors for road boundaries and road surface semantics should have high discriminatory value, and high precision and recall in real-world conditions. Descriptors for scene recognition need to be invariant and robust to scale and colour changes in road scenes. Road markings should be meaningfully categorised and relationally classified. Choosing a geometric curve representation for parameterising road boundaries requires consideration of both the representational power and computational costs of reasoning about and computing the curves. Finally, progress measures should be well defined and allow for meaningful and objective assessment of system performance.

The scope of each of these sub-problems varies with application and previous systems have been limited to traffic signal recognition and lane estimation, roadwork barrier mapping, and lane keeping problems. In this work we extend the scope to allow for holistic situational awareness. In order to achieve this, the following contributions to the field are made:

- A probabilistic road marking joint classification system for recognising seven distinct classes of road markings on the road: single and double boundary, boxed junctions, separator lanes, zig-zag, special lanes and intersections. Our framework accommodates a varying graphical model structure where each road scene gives rise to a unique structure. These markings are also jointly classified, allowing the vehicle to read and interpret the road scene.
- 2D LIDAR-based feature detectors for road boundaries based on local orientations and ranges in the sensor frame were developed. These algorithms were especially developed for our unique vertically downward facing sensor configuration.
- A fast greedy clustering algorithm for extracting the ground plane from a 3D point cloud generated from a vertically downward facing LIDAR sensor was developed. This algorithm can readily segment the road surface from a point cloud without plane fitting (non-parametric).
- A geometric barycentric “BaryArc” model for road boundary approximation was developed, based on the Department for Transport’s geometric road models.
- Demonstrating the link between opponent colour vision in humans and the design of roadwork objects. Developing a biologically inspired opponent colour roadwork scene signature for roadwork recognition in images.
- Benchmarking datasets for road marking classification, boundary estimation and roadwork scene recognition. The road marking dataset contains 2068 images with 13099 unique hand labeled annotations of over seven distinct classes of road markings. The road boundary dataset contains 32km of road data. The roadwork scene recognition datasets contains 10473 images of roadwork and road scenes with ground truth annotations. All three benchmarking datasets comprise of city, residential and motorway roads.
- A data structure for efficiently representing Situational Awareness in autonomous vehicles.

1.4 Structure of the Thesis

In Chapter 2 we define the meaning of situational awareness in autonomous vehicles as well as introduce the data structure representation that summarises the contributions of this thesis.

The algorithms used to extract and interpret road semantic information derived from images and LIDAR point clouds form the body of this thesis. Specifically, road marking classification is discussed in Chapter 3, and road boundary estimation in Chapter 4. Chapter 5 introduces a biologically inspired roadwork scene recognition framework based on opponent colour vision in humans.

The semantics extracted between Chapters 3-5 are summarised in our situational awareness data structure which we return to in Chapter 6. Finally, conclusions and an outline for future research directions are given in Chapter 7.

1.5 Publications

The roadwork scene recognition system described in Chapter 5 was published at IROS 2013 [92]. We also developed a Roadwork Prior framework, for assessing the probability of encountering roadworks along a vehicle trajectory, which was published at ITSC 2012 [91].

Chapter 2

Background and Context

2.1 Situational Awareness in Mobile Robotics

For life long autonomous driving to occur, there needs to be some degree of situational awareness on the part of the robotic car, allowing it to safely navigate its environment. Formally, situational awareness is defined as a three level process involving “the perception of elements in the environment within a volume of time and space, and the comprehension of their meaning, and the projection of their status in the near future” [34]. In the context of autonomous driving, the ultimate goal of situational awareness is to assist in informative decision making. Within the intelligent transportation community, different approaches for achieving situational awareness have been proposed before in a host of sub-problems including robust lane estimation [56], moving obstacle detection [126, 9], vehicular communication systems [87, 129] and road marking recognition [72, 138]. These approaches solve important problems in intelligent transportation but are limited in terms of overall situational awareness scope.

In the ergonomics science community, three main perspectives on situational awareness exist, namely the individual, computing and systems approach. The individual approach focuses on analysing drivers whilst the computational approach considers information sources such as displays, road signage and in-vehicle systems. Contrastingly, a systems perspective considers the overall road transport system and interactions between motorists, cyclists, pedestrians and infrastructure, vehicles and signage [119] when assessing situational awareness. Salmon et al. argue that true situational awareness is not driver-centric (or in our case, autonomous vehicle-centric) but rather distributed based on infrastructure, road environment, vehicle to vehicle communication, and other vehicle and pedestrian interactions [119]. This lays a good

foundation for formulating a semantically complete description of situational awareness for autonomous vehicles, which we draw inspiration from. In extending related work, this thesis describes a distributed situational awareness approach which considers road infrastructure, road environment and roadworks using the three level approach of perception, interpretation and projection. Simply put, we frame situational awareness in autonomous vehicles in terms of reading the road semantics in order to automatically derive a sense of what's going on.

2.2 Data Representation for Situational Awareness

In the previous section, we motivated a need for situational awareness on a semantic level in autonomous driving. However, the question of data representation still remains to be addressed. Assuming the successful extraction of road semantics from image and laser data, in what form could this data be stored for easy online retrieval by the vehicle? Also, exactly what information should be stored in such a data structure? The first question touches on the issue of representation which we briefly introduce here. The second, more challenging question, forms the body of this thesis and is tackled in two tiers. Firstly, we consider *reading the road surface* by extracting and interpreting road semantics characterising the road itself: road markings and road boundaries. Secondly, we consider *reading the road scene* through recognising roadworks.

We introduce here a data structure which can be seen and understood as binding the Chapters and contributions of this thesis together. As we progress through the Chapters, we will describe the details of how the fields of this data structure can be automatically filled in. However, we are well served by first giving a big picture view of the totality of our ambition and a versatile and expressive data structure is a good pedagogical medium to achieve this. One such data structure is a Google Protocol Buffer.

2.2.1 Google Protocol Buffer Representation

A Google protocol buffer is a way of encoding structured data in an efficient (in terms of speed and storage) yet extensible format similar to XML [48]. The protocol buffers are platform independent and language neutral and enable the serialisation of structured data. Protocol buffers (protobufs) have the added benefit of a simplified programmers interface: after defining the data structure, generated source code is used to read and write to this data structure using Java (and hence MATLAB [93]), C++, or Python programming languages [48].

This standard representation therefore simplifies remote procedural calls between the vehicle's processors

whilst allowing for compact storage and representation of relevant extracted road semantics.

In order to use a protocol buffer, the structure must first be defined in the relevant `.proto` protobuf message type file. We define a hierarchical situational awareness protocol buffer message type file that contains the contributions of this thesis:

```
message SituationalAwareness {  
    required float position = 1;  
    required float orientation = 2;  
    repeated Road road = 3;  
    repeated Boundaries boundaries = 4;  
    repeated Works works = 5;  
}
```

Here, the “position” and “orientation” fields describe the pose of the vehicle in a known reference coordinate frame. The sequence of positions and orientations form the smooth vehicle path or trajectory thus creating a path length parameter. The “repeated” modifier signifies that the field may be repeated any number of times (including zero) similar to a dynamically sized array [48]. It’s worthwhile noting that the numbers equated to each field name are not field assignment values but rather placeholders - these can be ignored.

In order for the SituationalAwareness data structure to be useful to a vehicle at a later time, a sensor configuration file `Sensor.txt` is also made available. This file contains information about:

- the type of vehicle used to create the data structure
- the exact location of all the sensors on the vehicle (GPS, cameras, LIDAR etc.)
- the camera intrinsic and extrinsic calibration parameters
- the sensor extrinsic transformation parameters ($T_{\text{INS-LASER}}$, $T_{\text{CAMERA-LASER}}$ etc.)

Thus the semantic information stored in the SituationalAwareness data structure can be meaningfully retrieved by a vehicle and used to acquire a sense of what’s going on.

In summary, the SituationalAwareness message defined above contains information about road markings, road boundaries and roadworks represented as “Road”, “Boundaries”, and “Works” message types respectively. In the next sections we define these message types in detail.

2.2.1.1 Protocol Buffer Message Type: Road Markings

The road marking protobuf message type stores the detected road marking classes and associated interpreted road scene situations (semantics). Algorithm details are discussed in Chapter 3.

```

message Road {
  enum RoadmarkingType {
    SGL = 0; DBL = 1; DIV = 2; ZIG = 3;
    INT = 4; BJ = 5; SPL = 6; NONE = 7;
  }
  message Roadmarkings{
    optional boolean condition = 1;
    required RoadmarkingType type = 2 [default=NONE];
    required float location = 3;
  }
  enum SituationType {
    PEDESTRIAN_APPROACH = 0; NO_DRIVE = 1;
  }
  message Situations{
    optional string info = 1;
    optional SituationType sit = 2;
  }
  repeated Roadmarkings = 1 [packed=true];
  repeated Situations = 2 [packed=true];
}

```

The “condition” in the Roadmarkings message captures whether the marking is clearly visible or faded. The “enum” type assigns a set of predefined values to a field, in this case the road marking classes we consider (“RoadmarkingType”) and the road scene interpretations we infer (“SituationType”). Table 3.1 explains the road marking class abbreviations shown above. SituationTypes “PEDESTRIAN_APPROACH” and “NO_DRIVE” refer to the detection of pedestrian crossings and boxed junctions respectively. The “location” field for the Roadmarkings message stores the local position of each road marking in the image.

2.2.1.2 Protocol Buffer Message Type: Road Boundaries

The boundaries protobuf message stores the BaryArc parameters necessary to draw the curve describing a road boundary. Relevant semantics such as road curvature, center line, intersection locations, and boundary association (left or right) are also stored. The boundary extraction algorithms used to generate this semantic information are derived in detail in Chapter 4.

```
message Boundaries {  
  
    enum BoundaryType {  
        LEFT = 0;  
        RIGHT = 1;  
    }  
  
    message Edges{  
        required BoundaryType type = 1;  
        repeated float param = 2;  
        repeated float centerline = 3;  
        repeated float curvature = 4;  
    }  
  
    message Intersections{  
        required float location = 1;  
    }  
  
    repeated Edges = 1 [packed=true];  
    repeated Intersections = 2 [packed=true];  
}
```

2.2.1.3 Protocol Buffer Message Type: Roadworks

The roadworks protobuf message structure captures the location and type of roadwork scene, including the detection prediction score. Implementation details for our roadwork scene classification framework are given in Chapter 5.

```
message Works {  
  
    enum RoadworkType {  
        URBAN = 0;  
        CITY = 1;  
        MOTORWAY = 2;  
    }  
  
    message Roadworks{  
        optional RoadworkType type = 1;  
        required string score = 2;  
    }  
  
    repeated Roadworks = 1 [packed=true];  
}
```

The Roadworks message “score” is the predicted classification score output of our binary boosting classifier. This score is proportional to the likelihood of the presence of roadworks in the scene and can be calibrated to give the posterior probability using Platt Scaling, Isotonic Regression or Logistic Correction [98]. The calibrated score could be input for a decision making step which revokes autonomy (hands vehicle control over to a human operator) if the probability of roadworks in the scene is above a learned threshold.

We now turn our discussion, over the next three Chapters, to the methods used in this thesis to extract the relevant semantic information required for situational awareness in autonomous vehicles. In particular, we consider reading the road surface (markings and boundaries) followed by reading the road scene (roadworks). Whilst this is hard, we now progress to considering exactly how we might extract the information required to fill in the template data structure presented in this Chapter.

Chapter 3

Reading the Surface: Road Marking Classification

Road marking classification refers to the problem of recognising distinct classes of road markings. Road markings encode the rules of the road whilst capturing the upcoming road layout. Classifying road markings is therefore useful for guiding the search for other markings, detecting the road type (one-way or two-way), deciding what manoeuvres are permitted [21], and over taking and lane changes [39].

This Chapter introduces a probabilistic framework for joint classification of road markings in images. Our algorithm is able to accurately and efficiently classify road markings into one of seven distinct classes, as well as infer road scene semantics such as pedestrian approaches on the road.

3.1 Literature Review

In the literature, the scope of road marking classification includes detecting arrows, painted road signs and letters, boundary type lanes, zebra crossings (crosswalks) and intersections.

3.1.1 Road marking arrows and painted road signs

In earlier work, Charonnier et al. extracted parameters such as dash modulation and line width by representing lane markings as polygons of grouped detections, and their system could also detect arrow type road markings [19]. More recently, the simplest approaches to road marking recognition involve some form of template matching. Wu and Ranganathan proposed a template matching based system for finding

road marking signs such as speed limits (“35” and “40”), “left”, “right” and “forward” turn arrows, “bike”, “stop”, “ped”, “xing” and “rail signs” [152] and achieved 90.1% and 0.9% true positive and false positive rates respectively, based on two optimally set thresholds for matching. They extracted Histograms of Oriented Gradients (HOG) [26] features at Features from Accelerated Segment Test (FAST) [118] interest points. Feature matching, followed by structural matching to capture the 2D geometry of the FAST interest points was used for robust classification at runtime [152]. Interestingly, unlike much of the related work [141], Maximally Stable Extremal Regions (MSER) [90] which finds extremal connected regions (pixels in a region that are much brighter/darker than those on the boundary) was used for finding road marking pixels (regions of interest for the template matching). This approach was found to be quite robust to lighting conditions due to the repeatability of the interest points, however the method is very sensitive to shadows due to contrast differences. Although the method worked well for classifying complex road markings, simpler markings such as forward arrows produced more false positives. Similarly, Vacek et al. detected lanes using a particle filter (one tracker per lane). Scan lines were used to classify the type of lane (solid or dashed) and template matching, using sum of squared differences between a template and example, was used to detect arrow markings [138]. Template matching approaches such as [152, 90] suffer from inaccuracies arising when road markings are partially observed due to obstacles or faded road paint.

In recent years, machine learning approaches to road marking classification have become popular [27, 155, 72]. Such approaches are often coupled with off-the-shelf classifiers (Neural Networks, Support Vector Machines etc.). A contour representation of road markings was extracted using multi-level thresholding by Kheyrollahi and Breckon and the authors trained an Artificial Neural Network (ANN) classifier to recognise six arrow marking types (achieving 85.2% accuracy) and 17 alpha numeric road marking characters (achieving 80.7% accuracy) [72]. A feature vector of aspect ratio, central moments, Hu spatial moments, and vertical and horizontal histogram projections was used [72]. The system also performed word recognition against a dictionary to recognise markings such as “BUS ONLY”, “NO” and “SLOW”. This method however relies on extracting road marking glyphs, and worn markings produce incomplete glyphs whilst unbounded markings such as boundaries cannot be easily represented using glyphs.

Road markings typically found at intersections (i.e. different types of painted arrows) have been classified based on a decision tree (with hand tuned thresholds) and size constraints given a set of six numerical features based on relationships between the road marking border points and extracted RANSAC line fit [27]. The disadvantage of this approach is misclassifications due to worn or partial road markings

and the implementation of the decision tree in the form of nested if statements. In similar work, [155] classify five kinds of markers (arrows and straight lanes) using moments to capture shape information.

Finally, a generative learning method based on eigenvectors was used by [99] to recognise arrow and numbered road markings.

3.1.2 Boundary type markings

Boundary type road marking classification involves the recognition of horizontal road markings and their semantics (dashed, solid etc.). Fourier analysis was used by Collado et al. [21] to build a lane interpretation system for classifying continuous, discontinuous and merge type lanes. Lanes were extracted using an edge filter in a bird's eye-view image by detecting dark-bright-dark transitions in the horizontal direction [21]. Markings were then parameterised as straight lines extracted using the Hough transform. Fourier analysis on the intensity line profile of the thresholded image was used for classification, which involved applying thresholds to the resulting Fourier power spectrums within heuristically determined frequency bands. This system was successfully tested on motorway roads [21, 22].

A three level cascade of binary classifiers was used by [28] to classify horizontal road markings into dashed, dashed-solid, solid-dashed, and double solid classes for lane change applications. A Support Vector Machine was used as the first classifier in the cascade to distinguish between solid and dashed road markings, and subsequent classifiers used decision rules. This method was essentially a basic way of building a multi-class boosted classifier but only achieved a classification rate of 71.53%.

3.1.3 Zebra crossings and intersections

A template matching approach for the recognition of crosswalks (zebra crossings) and painted arrows (5 types) based on comparisons of extracted road markings with single or repetitive rectangular patterns was proposed by [39]. Extracted road markings were compared with single/repetitive rectangular patterns and crosswalks were detected if at least two components had the same orientation, centroid and predefined separation. The arrow markings were detected based on the Hamming distance between the model image and candidate image following the application of a threshold. Although 90% detection and 78% accuracy was achieved for classifying crosswalks and arrows respectively, the disadvantage of this method is that several heuristically set thresholds are required for the matching step.

Prior information about known dimensions of specific road marking types has also been exploited in the literature. Zebra crossings (crosswalks) and dashed lines were reconstructed in [131] where the

authors exploited prior information in the form of known 3D size and shapes of dashed lines and zebra crossing road markings. Specifically, Canny-Deriche edge detection was performed over stereo pairs and edge matching was used to discard false detections. A signature-based recognition approach exploiting the alignment of edges with the road axis was used: projections of the edges along horizontal and longitudinal axes provided accumulation signature features [131] (e.g. repetitive peaks for zebra crossings versus a pair of peaks for dashed lines in the transverse axis) where the road axis was estimated to be the most frequently occurring orientation of 2D line edges. The system was tested on just 1km of data (150 stereo pair images) and achieved a detection rate of above 86% [131] for both broad class types and 1.7cm accuracy for reconstruction. However, partially occluded strips resulted in errors exceeding 10cm.

Interesting work was done by Duchow and Koertner who detect individual lane segments and aggregate these into intersection lanes (to automatically discover which markings belong to which lanes at intersections) [32]. A machine learning approach is proposed where a set of features (width, length, orientation, position, intensity differences and deviations) is extracted from triplet groups of lane marking segments. The features are fed into a Support Vector Machine (SVM) which classifies them as belonging to the same lane or not [32]. Triplets are then combined to form lane boundaries. Importantly, this approach requires no use of prior information (digital maps or geometric road models) and neighbourhood information is captured by considering triplets of road markings at a time (whilst computing features using a single, two, or three segments). Although [32] considers only intersection classes, their work is the most related to ours as relationships between neighbouring road marking segments are exploited by the classification framework and no explicit geometric priors are used, similar to our approach. However, the neighbourhood information is captured based on spatial information only unlike our work where a relational classifier captures neighbourhood information such that the predicted class of a road marking influences that of its neighbours.

With the exception of [19, 27], most of the work in road marking recognition requires a preprocessing step involving the creation of a bird’s-eye-view of the road’s surface [72, 131, 21, 152, 65, 99, 155, 32, 39] (see Figure 3.1) using Inverse Perspective Mapping in order to remove the perspective distortion resulting in parallel lines meeting at the image vanishing point (on the horizon line). This has been shown to make classification more robust and we adopt this approach in our work.

Although much work exists in the road marking detection and tracking literature, road marking classification research is limited to classifying boundary type, intersection, arrows or traffic sign road markings. Our approach extends previous work in three ways: Firstly, we classify seven distinct classes of

Original Images



Inverse Perspective Transformed Images



Figure 3.1: An example of Inverse Perspective Mapping (IPM) to remove the perspective effect in images. Notice how, in the IPM images, parallel lines no longer intersect at the vanishing point.

road markings within a unified probabilistic framework that is flexible to include more classes. Secondly, we propose road marking classification within a relational learning framework thus allowing us to capture the underlying structure of road markings, which embody the rules of the road. Finally, our approach is probabilistic thus capturing uncertainty (unlike template matching approaches), and no explicit geometric priors are required.

Large road marking databases are hard to come by, apart from [141] which does not contain class labels (and has just 116 road images) and [152] which has bounding box ground truth labels of painted road signs in 1208 images. We therefore manually create our own benchmarking ground truth class annotated dataset containing 2068 images and over 13099 unique annotations, the largest of its kind available to date. The rest of this Chapter outlines our road marking classification framework and the results that will be stored in our Situational Awareness data structure introduced in Chapter 2.

3.2 Representation of Road Markings

Attempting to model the distribution of road marking classes on roads is a difficult problem that has only been partially addressed in the literature. Much research has focused on finding arrows and painted traffic signs on roads [152, 19, 27, 155, 72, 99], using feature-based and template matching recognition approaches. Although some authors have classified solid and dashed lines [21], intersections [32] and zebra crossings [39, 131], to the best of our knowledge, we are the first to address the issue of road marking classification for semantic awareness. In other words, our goal is for a vehicle to read and interpret the



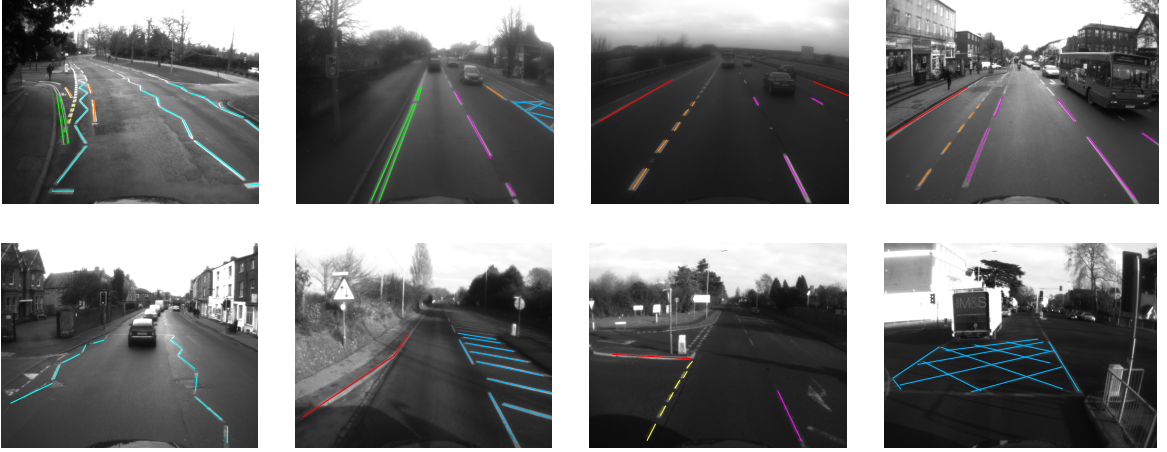
Figure 3.2: Zig-zag approach road markings lead up to a traffic light controlled pedestrian crossing.

rules of the road encoded in road markings in order to gain a sense of situational awareness of the current and upcoming road topology. For example, observing zig-zag approach road markings (see Figure 3.2) informs a vehicle that there is an upcoming traffic light controlled pedestrian crossing. Successful detection of zig-zag approaches therefore provides strong priors for the location of traffic lights and pedestrians which is useful also for cuing the appropriate detector modules. We formulate situational awareness on the road, first in terms of reading the road surface. Specifically, we consider reading (classifying) seven distinct road marking types and present a probabilistic framework to do so. Importantly, there is a confidence associated with each classification allowing the vehicle to reason further about what it reads.

3.2.1 Seven Road Marking Classes

We consider classifying seven types of road markings shown in Table 3.1 and the accompanying figure. Several observations can be made about road marking classes from the images:

- The number of lanes influences the type of road markings present (notice how the number of separator markings depend on the number of lanes which is not known a priori). We presently do not rely on having prior knowledge of the number of lanes present, however in future work this is a parameter we will learn from combining local vehicle observations with web-based sources.
- The class of a specific road marking is strongly influenced by that of its neighbours (notice how the different colour labels tend to be spatially clustered into groups). We therefore model the relationships (connectivity) between road markings based on proximity.
- The perspective effect in the images distorts the true lengths, orientations and positions of road markings. It is therefore necessary to either recover these true properties or explicitly model this perspective effect. Our method thus begins by removing the perspective distortion to recover the true geometric properties of road markings.



Class	Description	Abbreviation	Colour
1	Single boundary	SGL	Red
2	Double Boundary	DBL	Green
3	Separator	DIV	Magenta
4	Zig-zag	ZIG	Cyan
5	Intersection	INT	Yellow
6	Boxed Junction	BJ	Blue
7	Special Lane	SPL	Orange

Table 3.1: Seven road marking classes we classify in our framework (examples shown in Figure).

3.2.2 Line Segment Representation

Road marking extraction in images is generally a two step process involving feature extraction to detect the road paint pixels followed by assigning a geometric model to the markings [141]. In this work, we assume that the markings have already been detected. Choosing a geometric model is a design consideration dependant upon the system application. For lane detection and tracking applications for example, smoothness and continuity is an important consideration, thus spline [56] based geometric models are often used. For our application, we use a line segment geometric model similar to [21, 32]. In particular, a painted road marking segment is represented by a line segment (or series of line segments) parameterised by two endpoints p_A and p_B as shown in Figure 3.3. The midpoint locations, $p_C \triangleq \frac{(p_A + p_B)}{2}$, are important because they represent the node locations for our road marking connectivity graphical structure. Our road marking classification problem can now be framed in terms of assigning the most likely road marking class to each node.

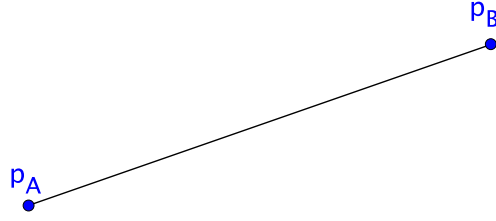


Figure 3.3: Line segment representation of road markings. The line segment is the geometric model of a road marking, parameterised by endpoints p_A and p_B . The midpoint $p_C \triangleq \frac{(p_A + p_B)}{2}$ represents the node location for our graphical structure.

3.3 Inverse Perspective Mapping

Given the line segment representation, we now wish to extract discriminative features using these segment endpoints. However, before proceeding with feature extraction, it is necessary to remove the perspective distortion in images so that true (scaled) lengths and orientations of road markings can be recovered. Details of our feature functions are discussed later in Section 3.4.3.

Inverse Perspective Mapping (IPM) refers to the process of removing the linear perspective effect from images. Table 3.1 and Figure 3.1 illustrate how perspective distortion affects the true underlying properties (length, orientation etc.) of road marking segments. This effect is especially evident in the zig-zag type markings (cyan colour) where the lengths of each road marking segment appear to be different. In reality however, each road marking segment in a zig-zag approach is the same length and one of two opposing orientations according to the UK Highway Code design specifications [38]. To address this, the linear perspective can either be implicitly modelled [19, 27] or Inverse Perspective Mapping (IPM) can be performed [72, 131, 21, 152, 65, 99, 155, 32, 39] to create a perspective free bird’s-eye-view image. We adopt the latter approach in our work.

Projective geometry describes objects as they appear (lengths, angles and parallelism is distorted) and provides a mathematical model for describing how images of the 3D world are formed. To reverse the perspective effect, we require a homography matrix $\mathbf{H}$ which defines the one-to-one relations between the camera image pixels and the synthesised bird’s-eye-view pixels as $x' = \mathbf{H}x$. We therefore begin by making the important assumption of a locally planar road surface. To estimate the homography matrix we require the camera’s intrinsic parameters $\mathbf{K}$, the relative virtual rotation angle of the camera about its center (from the current frontal view to the synthesised overhead view) θ , and the height above the ground of the

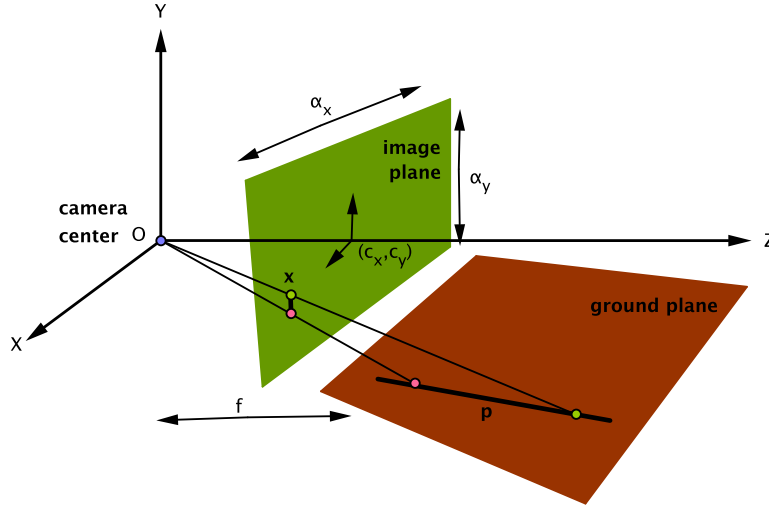


Figure 3.4: The frontal pinhole camera model showing how points in the real 3D world are projected into the 2D camera imaging plane. Specifically, for world point $\mathbf{p} = [X \ Y \ Z]^T$ and image point $\mathbf{x} = [x \ y]^T$ it is straightforward to show, using similar triangles, that $\mathbf{x} = [x \ y]^T = \frac{f}{Z} [X \ Y]^T$ [53].

camera h . The camera intrinsic matrix is given by:

$$\mathbf{K} = \begin{bmatrix} \alpha_x & \gamma & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where $\alpha_x = fm_x$ and $\alpha_y = fm_y$ represent the focal lengths in pixels where m_x and m_y are scale factors relating pixels to distance [53]. The principle point (image centre) is represented by c_x and c_y in the intrinsic matrix. Finally, γ is the skew coefficient between the x and y axis and is measured to be zero for our camera set up [53]. Our camera intrinsic parameters are known and they explain the camera image formation by characterizing the perspective projection from the 3D world frame to the 2D camera frame as shown in Figure 3.4.

The virtual rotation matrix we apply to the camera position, scaled by the height, is given by:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & h * 0 \\ 0 & \cos(\theta) & -h \sin(\theta) \\ 0 & \sin(\theta) & h \cos(\theta) \end{bmatrix}$$

Therefore, to create the homography, we first perform the scaled rotation in the world coordinate frame where the camera is located, then we translate this into the camera coordinate frame via the intrinsic

matrix [53] using

$$\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1} \quad (3.1)$$

The camera intrinsic matrix and height h are known, and we determine θ empirically to achieve the bird's-eye-view results shown in Figure 3.1.

Other methods for computing the homography matrix $\mathbf{H}$ have been proposed before, including solving a system of linear equations. Here, the idea is to estimate a homography matrix of the form

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (3.2)$$

and solve this by creating a system of linear equations [53] from

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

By setting $h_{33} = 1$ (since multiplying the h_{ij} 's by a non zero factor k does not affect the equations) we get eight degrees of freedom [53]. The system of linear equations can therefore be solved for unknowns h_{ij} 's if four point correspondences are known [53] (four points from the road image and four matching points from the virtual overhead bird's-eye-view image). The next section explains our road marking classification framework using these new transformed bird's-eye-view images.

3.4 Conditional Random Fields: Road Marking Classification

Conditional Random Fields (CRFs) are powerful undirected probabilistic graphical models for relational learning. They were initially proposed by Lafferty et al. for Parts Of Speech (POS) tagging in natural language processing [83]. A CRF directly models the conditional probability distribution of the latent variable (hidden state) given the observation, thus avoiding the challenging task of specifying a generative model for observations. This approach contrasts Hidden Markov Models (HMMs) or Markov Random Fields (MRFs) that apply Bayes rule to infer hidden states [114].

In particular, our road marking classification framework employs a CRF such that each road marking

segment represents a node with a fixed number of possible states per node (i.e. 7 states for the classes of road markings we consider). The CRF then models geometric information about the road marking segments and their neighbours, and parameters are learnt discriminatively from the data.

3.4.1 Conditional Random Fields

Conditional Random Fields are undirected graphical models that directly model the conditional distribution over hidden variables $\mathbf{x}$, given observations $\mathbf{z}$ as $p(\mathbf{x}|\mathbf{z})$. A CRF is a graphical model where each node represents the hidden states and observations. Thus, given a set of n hidden states $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ and data $\mathbf{z}$, the nodes $\mathbf{x}_i$ and their associated graph structure (connectivity) define the conditional distribution $p(\mathbf{x}|\mathbf{z})$ over the hidden states (Figure 3.5 shows an example CRF, probability distribution $p(\mathbf{x}|\mathbf{z})$, for road marking classification). The hidden states $\mathbf{x}_i$ are the unknown road marking classes, the observations $\mathbf{z}_i$ correspond to line segment endpoints extracted from each road marking, and the edges of the graph represent conditional dependencies between random variables at the nodes. Each fully connected subgraph of this graph structure represents a clique in the graph. Let C be the set of cliques in the graph, then a CRF factorizes the conditional distribution $p(\mathbf{x}|\mathbf{z})$ into a product of clique potentials $\phi_c(\mathbf{z}, \mathbf{x}_c)$ where $c \in C$, and $\mathbf{x}_c$ and $\mathbf{z}$ are the hidden nodes and data of clique c , respectively. Each clique potential $\phi_c(\mathbf{z}, \mathbf{x}_c)$ is a strictly positive function of the variable configuration and captures the compatibility amongst variables in the clique (thus large values imply a more likely configuration) [115]. The Conditional Random Field model can therefore be written as:

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} \prod_{c \in C} \phi_c(\mathbf{z}, \mathbf{x}_c) \quad (3.3)$$

where the normalisation factor (partition function) is $Z(\mathbf{z}) = \sum_{\mathbf{x}} \prod_{c \in C} \phi_c(\mathbf{z}, \mathbf{x}_c)$, which sums over all possible label assignments. For a large number of nodes and labels (typical of many problems), this summation can grow exponentially in size. Thus, exact inference is possible only for a limited class of CRF models such as linear chain CRFs [74]. In the log-linear parameterisation of CRFs, the potentials are described using a log-linear combination of feature functions:

$$\phi_c(\mathbf{z}, \mathbf{x}_c) = \exp(\mathbf{w}_c^T \cdot \mathbf{f}_c(\mathbf{z}, \mathbf{x}_c)) \quad (3.4)$$

where $\mathbf{w}_c^T$ is the weight vector we need to learn and $\mathbf{f}_c(\mathbf{z}, \mathbf{x}_c)$ is a feature function extracting a vector of features in a clique. Here, large weights imply high compatibility and low weights a less likely configuration.

Thus, the log-linear parameterisation of a CRF can be written as

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} \exp \left\{ \sum_{c \in C} \mathbf{w}_c^T \cdot \mathbf{f}_c(\mathbf{z}, \mathbf{x}_c) \right\} \quad (3.5)$$

where $Z(\mathbf{z}) = \sum_{\mathbf{x}} \exp \left\{ \sum_{c \in C} \mathbf{w}_c^T \cdot \mathbf{f}_c(\mathbf{z}, \mathbf{x}_c) \right\}$ is the new partition function. In particular, we consider pair wise potentials such that cliques have at most two nodes. Thus our proposed CRF model is given by:

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} \prod_{\langle i, j \rangle} \psi_{ij}(x_i, x_j, \mathbf{z}) \prod_i \psi_i(x_i, \mathbf{z}) \quad (3.6)$$

for unary (node) and binary (edge) potentials given by $\psi_i(x_i, \mathbf{z})$ and $\psi_{ij}(x_i, x_j, \mathbf{z})$ respectively. Each potential function is defined as the exponentiated sum of the product of the appropriate weight and feature:

$$\psi_i(x_i, \mathbf{z}) = \exp \left\{ \sum_k \lambda_k g_k(x_i, \mathbf{z}) \right\}$$

$$\psi_{ij}(x_i, x_j, \mathbf{z}) = \exp \left\{ \sum_t \mu_t h_t(x_i, x_j, \mathbf{z}) \right\}$$

where $g_k(x_i, \mathbf{z})$ and $h_t(x_i, x_j, \mathbf{z})$ are the node and edge feature functions respectively. Similarly λ_k and μ_t are the node and edge weights we wish to learn via parameter estimation using the training data.

CRFs avoid encoding distributions over just the variable $\mathbf{z}$ which allows great flexibility in choosing a rich set of continuous observed variables (feature functions) whose dependencies may be difficult to model (no simple parametric form) or understand. We can therefore exploit domain knowledge to define a set of discriminative features without needing to model their joint distribution [74].

Furthermore, the definition of a CRF in Equation 3.5 induces a different value for the partition function for every assignment z to $\mathbf{z}$ - this is illustrated by the orange coloured observations in Figure 3.5.

3.4.2 Graphical Structure

The graph structure (connectivity) is a way of relating the relationships between nodes (road marking segments) present in an image. When designing a graph structure, it is important for the connectivity to somehow capture the underlying relationships between adjacent nodes. From Table 3.1 we see that road markings belonging to the same class tend to be spatially clustered into groups, which suggests that nodes should be connected to their closest neighbours. Similar to [31], we use Delaunay triangulation to create a graph structure of road markings on the road, assuming a locally planar road surface.

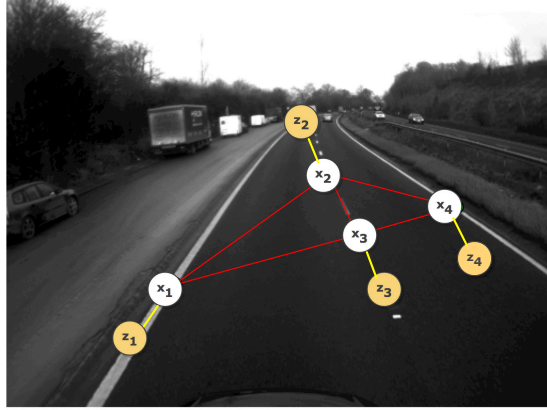


Figure 3.5: CRF model, $p(\mathbf{x}|\mathbf{z})$, for road marking classification. The hidden states $\mathbf{x}_i$ are the unknown road marking classes and the observations $\mathbf{z}_i$ (shaded orange) are line segment endpoints extracted from each road marking.

The nearest neighbour graph is a subgraph of the Delaunay triangulation [85], thus the closest neighbour j to any point p is on an edge jp . This implies that, for our application, road marking segments will be connected to their optimal nearest neighbours allowing relational information to be propagated throughout the road. It is worthwhile noting that the number of painted road markings (and hence nodes) on the road is mostly different for every frame unlike in [108, 82]. Therefore a unique connectivity graph structure exists per training example (or image) as demonstrated by Figure 3.6.

3.4.3 Feature Functions

Our feature functions for road marking classification describe geometric and spatial properties of line segments and their endpoints. The CRF learning algorithm associates an appropriate weight with each feature function given the training data. In CRFs, local features depend only on a single hidden state i and the associated data (node endpoints p_{Ai} and p_{Bi} ; and midpoint p_{Ci} as in Figure 3.3). Global features depend on two hidden states i and j . We begin by defining visual feature functions which are used to train a multiclass classifier, and later show how the classifier output scores are used as CRF features.

3.4.3.1 Visual Feature Functions

We formulate eleven geometric and spatial features capturing the properties of a single road marking. These local features are described as follows:

Length: this feature measures the length of a line segment. This is useful for separating typically long road markings (boundary, separator), medium length (zig-zag, special lane) and short markings

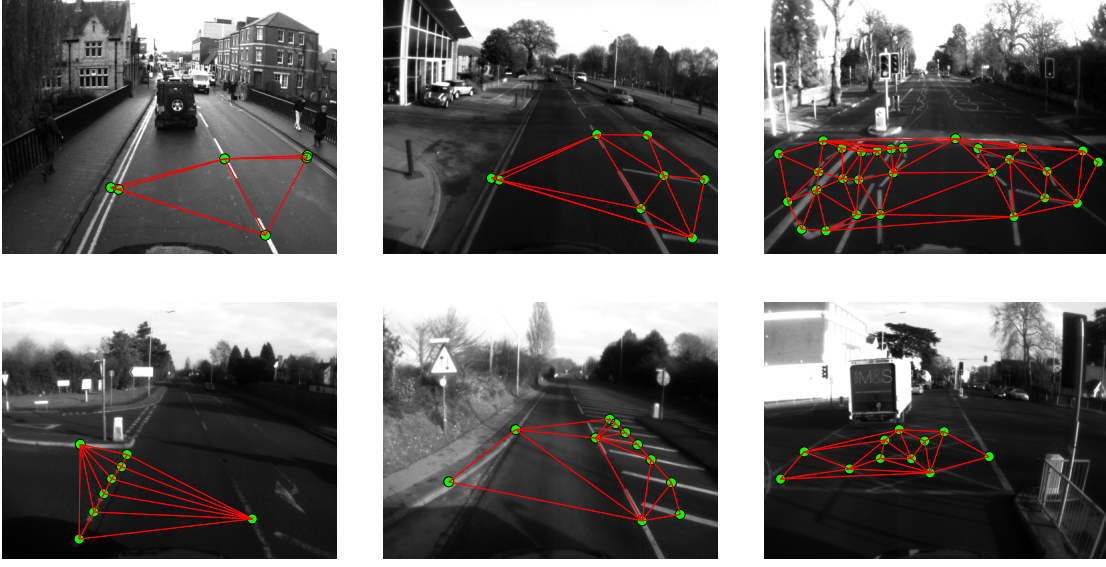


Figure 3.6: Some graph structures of road marking segments created using Delaunay Triangulation. Notice how each example (image) gives rise to a different graphical structure since the number of road markings varies. The node locations are taken to be the midpoint of each road marking segment (see Figure 3.3).

(intersection and special lane). If we denote the line segment endpoints for node i as p_{Ai} and p_{Bi} , then the length is given by:

$$f_{length}(i, \mathbf{z}_i) = \| p_{Ai} - p_{Bi} \| \quad (3.7)$$

X-midpoint location: this feature measures the spatial x location of the road marking in the image. This captures the fact that boundary or zig-zag type markings tend to be located towards the edges of the image (see Figure 3.2) whilst boxed junctions tend to take up the entire image. Similarly, special lanes and intersections are spatially located near the road boundaries as well. Given a line segment midpoint p_{Ci} , the x-midpoint location is given by:

$$f_{midx}(i, \mathbf{z}_i) = p_{Ci,x} \quad (3.8)$$

Y-midpoint location: for completion we also consider the y-midpoint location of the road marking. Although we removed the perspective distortion in the image, this feature is useful particularly at intersections and roundabouts where the overall orientation of road markings appears rotated. Formally, the feature is given by:

$$f_{midy}(i, \mathbf{z}_i) = p_{Ci,y} \quad (3.9)$$

Delta-x: in addition to capturing the location and length of a road marking line segment, it is also important to have features that capture the regularities. The change in x coordinates (delta x) feature captures the regular pattern of zig-zag approaches. Here markings of the same length and one of two orientations denote zig-zags. The delta x (and delta y) feature is constant for zig-zag type markings. For vertical type markings (boundaries, special lanes, intersections etc.) this feature is close to zero. If we denote the x segment endpoints for node i as $p_{Ai,x}$ and $p_{Bi,x}$ then the delta-x is given by:

$$f_{deltax}(i, \mathbf{z}_i) = \| p_{Ai,x} - p_{Bi,x} \| \quad (3.10)$$

Delta-y: this feature captures similar geometric properties to delta x. Specifically, for zig-zag markings, this feature is constant. For a vertical marking it simply reduces to the approximate length of the markings and for boxed junctions this feature varies depending on the type of boxed junction (small on the side or large spanning an entire intersection - see Table 3.1). If we denote the y segment endpoints for node i as $p_{Ai,y}$ and $p_{Bi,y}$ then the delta-y is given by:

$$f_{deltay}(i, \mathbf{z}_i) = \| p_{Ai,y} - p_{Bi,y} \| \quad (3.11)$$

Orientation: the orientation of a line segment is measured anticlockwise relative to the x-axis. This feature roughly evaluates to one of two values for zig-zag type markings and, with the exception of the boxed junction class, is approximately constant for road markings belonging to the same class in an image. It also roughly captures the vehicle orientation (which influences the relative orientation of the road markings). The orientation feature is given by:

$$f_{angle}(i, \mathbf{z}_i) = \angle(p_{Ai} - p_{Bi}, \mathbf{x} - \text{axis}) \quad (3.12)$$

Furthest vertical distance: this feature is simply a measure of how far away from the top of the image a road marking is. Let the y segment endpoints for node i be given by $p_{Ai,y}$ and $p_{Bi,y}$, then the furthest vertical distance feature is given by:

$$f_{vert}(i, \mathbf{z}_i) = \max(p_{Ai,y}, p_{Bi,y}) \quad (3.13)$$

Proportion segments to the left: this feature measures the percentage of road markings located to the left of the i^{th} node. This captures the overall distribution of road markings in the image relative to the position of node i . Intuitively, we expect this class distribution to be different across classes with boundary type markings having low proportions and separator markings proportions close to 0.5. Let N be the total number of nodes in an image and K_{left} be the number of nodes to the left of node i , then the feature is given by:

$$f_{propleft}(i, \mathbf{z}_i) = \frac{K_{left}}{(N - 1)} \quad (3.14)$$

Proportion segments to the right: this feature measures the percentage of road markings located to the right of the i^{th} node. It captures the overall distribution of road markings in the image relative to the position of node i similar to the previous feature. The symmetry of the road markings on the road is also captured by this and the previous proportion features. Let N be the total number of nodes in an image and K_{right} be the number of nodes to the right of node i , then the feature is given by:

$$f_{propriht}(i, \mathbf{z}_i) = \frac{K_{right}}{(N - 1)} \quad (3.15)$$

Closest midpoint distance: this feature is especially important for double boundary type road markings since they comprise of two road markings of similar length and orientation, running parallel. Thus the midpoint distance between double boundary road markings is small compared to the midpoint distances between other marking types. Let p_{Ci} denote the midpoint of node i and p_{Ck} the midpoint of the node closest to node i , then the feature is given by:

$$f_{midist}(i, k, \mathbf{z}_i) = \| p_{Ci} - p_{Ck} \| \quad (3.16)$$

Number of intersecting line segments: this feature is most important for boxed junction markings occurring at intersections/junctions. This is because for other classes, mostly no intersections occur. The feature function counts the number of line segments that intersect with the line segment associated with node i . Formally, given a function $isintersect()$ that returns 1 if there is an intersection or 0 otherwise, let l_i and l_k denote the line segments formed by joining the endpoints of the i^{th} and k^{th} nodes respectively, the feature function is then given by:

$$f_{int}(i, k, \mathbf{z}_i) = \sum_{k=1}^{N-1} isintersect(l_i, l_k) \quad (3.17)$$

Conditional Random Fields are able to incorporate continuous local features like the ones described above. CRF potentials are formed by exponentiating a linear function of the feature functions. To model the more complex non linear relationships between hidden states and feature values however, it is better to discretise (or binarise) the features [55].

3.4.3.2 Unary Score Features

Instead of directly using the local features described in Section 3.4.3.1 in our CRF model, we first train a RUSBoost classifier and use the output scores as unary features for our CRF. This approach was shown to be effective by Friedman et al. [43] who proposed first learning AdaBoost classifiers from trained labeled data. Other variants of this approach include using SVM scores and CRFs [55] to build a two stage classifier. We choose RUSBoost because it is robust to skewed training data - the number of examples per class (see Table 3.2) is not balanced since some classes have almost six times the number of training examples compared to others. This is true in general on roads - separator and boundary markings are more commonplace than boxed junctions for example.

Random Under Sampling Boost (RUSBoost) was first proposed by [121] for classifying imbalanced or skewed data. Given training data with K classes, let N be the number of examples in the class with the fewest members in the training data. Then classes with more members are under sampled (without replacement) by randomly selecting only N observations of every class. Thus RUSBoost only takes a subset of data with N observations from each of the K classes in the training set [122]. Like other boosting methods, RUSBoost works by creating a strong classifier by aggregating a combination of weak learners. This multi-class boosting framework uses weighted pseudo-loss for N observations and K classes to define a training error given by:

$$\frac{1}{2} \sum_{n=1}^N \sum_{k \neq y_n} d_{n,k}^{(t)} (1 - h_t(x_n, y_n) + h_t(x_n, k)) \quad (3.18)$$

where t indexes the training steps. The confidence of prediction by a learner at step t into class k is given by $h_t(x_n, k)$ where 0 indicates no confidence and 1 high confidence. The observation weights at step t for class k are given by $d_{n,k}^{(t)}$ and y_n represents the true class label. Typically, the pseudo-loss behaves similar to the weighted classification error in that the first few learners give low pseudo-loss errors then

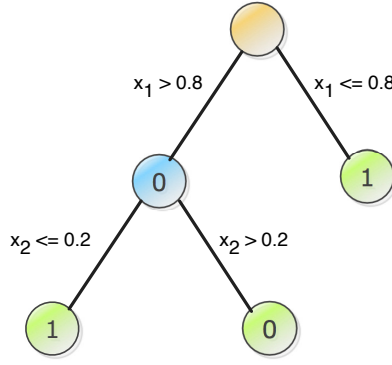


Figure 3.7: A simple example of a decision tree with two predictor variables x_1 and x_2 : the root node is shaded orange and the leaf nodes are shaded green. The blue node is a condition node.

the boosting classifier starts learning at a slower pace resulting in the pseudo loss value approaching 0.5 from below [122]. RUSBoost extends AdaBoost [41] for binary classification where a misclassification error given by Equation 5.5 is used.

The choice of a weak learner determines the performance of a boosting classifier. Simple decision stumps were initially proposed for AdaBoost [41], and regression learners have been used for GentleBoost [42]. We choose tree learners for our implementation. Tree learners refer to classification trees which predict an output based on the path followed from the root node to a leaf node of a decision tree. Figure 3.7 shows an example of a decision tree where the root, leaf, and condition nodes are shaded orange, green and blue respectively (classification trees give nominal responses of true or false whereas regression trees give numeric responses). In our example, we consider two predictor variables x_1 and x_2 . To classify a given example, follow the decision tree from root to leaf choosing the path that is true for each predictor variable condition.

A classification tree learnt from the data is used as the learners in our RUSBoost classification framework. For our application we choose tree learners with a minimum leaf span of five (each leaf has at least five observations resulting in deep trees). This means that if a split leads to a child node having observations less than the minimum leaf span, when creating the decision tree, a split with the best optimisation criterion subject to the leaf span constraint is chosen.

We train a RUSBoost classifier using the eleven features described in Section 3.4.3.1 for $t = 600$ steps using 70% of the data for training and 30% of the training data as a hold out set to test the quality of the resulting classifier. The classification output for each training example is a $K \times 1$ vector assigning a score to each class. For our road marking classification application, $K = 7$. The RUSBoost classifier learns an independent model which does not account for relationships between adjacent nodes. In fact, the model

considers each node (road marking) in isolation when classifying it. For this reason, instead of directly classifying road markings using the RUSBoost scores, we instead use this score vector as input features for our Conditional Random Field which learns statistical dependencies between adjacent hidden states.

3.4.3.3 Binary Features

Given the CRF node feature functions defined in the previous section, we also define binary feature functions to capture conditional dependencies between the hidden states. This allows the model to take into account the influence of the geometric and spacial properties of road markings on the transition between states (for example a neighbour node transitioning from a zig-zag to a boxed junction class). Thus for an edge connecting two nodes with unary feature functions $g_{k,1}(x_1, \mathbf{z})$ and $g_{k,2}(x_2, \mathbf{z})$ respectively, the edge feature is defined as $h_{t,12}(x_1, x_2, \mathbf{z}) = g_{k,1}(x_1, \mathbf{z}), g_{k,2}(x_2, \mathbf{z})$, the corresponding node features from both edge ends [120]. In theory, any real valued feature functions could be used to construct the edge features. Possible examples include the differences between node features on the edge, or edge features independent of node features such as distances between road markings connected by an edge. In our experiments we found that using edge features corresponding to node features, as defined above, produced very good classification results.

3.4.4 Parameterisation

When building a graphical model, the modelling issue of parameterising (or tying) the potential functions must be addressed. Ising like potentials assume that states are interchangeable, thus state transitions $a \rightarrow b$ are as likely as $b \rightarrow a$. For our application, this may not necessarily be the case. The probability of transitioning from class zig-zag to class boxed junction, for example, is not the same as the reverse transition probability. This is evidenced by the fact that zig-zag road markings usually occur before a traffic light controlled pedestrian crossing (in both directions of travel) [38]. Thus, transitioning from class zig-zag to boxed junction is associated with a low probability. On the other hand, transitioning from class boxed junction to zig-zag has a slightly higher probability because boxed junction markings at an intersection could be followed by a pedestrian crossing, denoted by a zig-zag approach, should the vehicle turn at the intersection. For our road marking classification model, we use full potentials to ensure that a different parameter for each combination of states in the edge potentials is used. Furthermore, the potentials are tied so that all nodes share node potentials and edges share edge potentials. Practically, this means that each road marking (node or edge) shares parameters.

3.4.5 Parameter Estimation

3.4.5.1 Conditional Log Likelihood Training

In parameter estimation, we are given the training dataset of M examples and associated graphical structures. Our aim is to estimate the parameters $\mathbf{w}_c^T$ in Equation 3.5 that best model the data. Let $\theta = [\boldsymbol{\lambda}, \boldsymbol{\mu}]$ represent the combined parameters of the node and edge features and $F(\mathbf{z}, \mathbf{x})$ represent all the features. The CRF learns the weights discriminatively by maximising the conditional log likelihood of the training data. Specifically, the log likelihood for M training examples is given by:

$$\log(p(\mathbf{x}|\mathbf{z})) = l_{\mathbf{x}|\mathbf{z}}(\mathbf{w} : \mathcal{D}) = \sum_{m=1}^M \theta^T F(\mathbf{z}_m, \mathbf{x}_m) - \sum_{m=1}^M \log \mathbf{Z}(\theta, \mathbf{z}_m) \quad (3.19)$$

This log likelihood function can be shown to be concave because $\log \mathbf{Z}(\theta, \mathbf{z})$ is concave: the Hessian is a covariance matrix of the features distributed according to θ and the first term is linear in the feature functions [74]. Thus the sum of a linear function and concave function is concave. This implies that the log-likelihood (Equation 3.19) is unimodal but the uniqueness of the global optimum is not necessarily guaranteed since the parameterisation can be redundant giving rise to multiple representations of the same distribution [74]. Because the log-likelihood function is concave, we know that the gradient must be zero at the maximum point. Computing the gradient of the log likelihood function, we get:

$$\begin{aligned} \frac{\partial}{\partial \theta} (l_{\mathbf{x}|\mathbf{z}}(\mathbf{w} : \mathcal{D})) &= \sum_m \left(F(\mathbf{z}_m, \mathbf{x}_m) - \frac{1}{\mathbf{Z}(\theta, \mathbf{z})} \sum_{\mathbf{x}'} \frac{\partial}{\partial \theta_i} \exp \{ \theta^T F(\mathbf{z}, \mathbf{x}') \} \right) \\ &= \sum_m \left(F(\mathbf{z}_m, \mathbf{x}_m) - \frac{1}{\mathbf{Z}(\theta, \mathbf{z})} \sum_{\mathbf{x}'} F(\mathbf{z}, \mathbf{x}') \exp \{ \theta^T F(\mathbf{z}, \mathbf{x}') \} \right) \\ &= \sum_m (F(\mathbf{z}_m, \mathbf{x}_m) - \mathbf{E}_\theta[F|\mathbf{z}_m]) \end{aligned} \quad (3.20)$$

Here different values of the conditioning variable $\mathbf{z}_m$ give rise to a new model. Also, the computation of the expected value under the model parameters $\mathbf{E}_\theta[F|\mathbf{z}_m]$ requires running inference over the model to compute the required probabilities. This implies that, in general, inference must be run for every single data case, conditioning on $\mathbf{z}_m$ however, the conditioned model is simpler [74].

No analytical form for this log likelihood maximum exists so one possibility is to use an iterative method to search for the global optimum. In our implementation, we perform parameter estimation using an L2-regularised log-likelihood and estimate the maximum using an iterative line search method (quasi-Newton approach) which takes steps in parameter space to improve the objective. L2-regularisation is necessary because as more features are added to the CRF model, the negative log-likelihood function decreases which could lead to over-fitting. The regularisation essentially adds a penalty for large parameter

values [74].

An alternative to optimising the negative log-likelihood is to optimise the pseudo-likelihood [8], where the product of conditional likelihoods is optimised instead of the joint likelihood.

3.4.5.2 Variational Inference

For the CRF model in Equation 3.5, inference refers to estimating the marginal distribution of each hidden variable $\mathbf{x}_i$ or the most likely configuration of all hidden variables $\mathbf{x}$ (decoding via MAP estimation). These tasks rely on using belief propagation, a message passing algorithm for estimating marginal distributions over a graph [10]. In belief propagation, messages are passed through the graph structure of the model and each node passes on received messages to its neighbours. This inference method is exact for trees, forests and other graphs not containing loops. For our application however, Delaunay triangulation creates loopy graphs, therefore we apply loopy belief propagation (LBP) [96] with the Bethe free energy approximation to the log-partition function, an approximate inference algorithm not guaranteed to necessarily converge to the true probability distribution (inference is intractable). We however found the performance of the LBP algorithm for inference in our CRF model to produce good results, even though convergence is not necessarily guaranteed.

We now discuss situational awareness results for road marking classification in the next section. These results will form the content of our Situational Awareness data structure already defined in Chapter 2.

3.5 Performance Evaluation

3.5.1 Data Sets

The image dataset was collected by [24] using the vehicle setup shown in Figure 3.8, and contained road images spanning city, residential and motorway roads. We created a benchmarking dataset containing 2068 images and manually labelled the images with 13099 ground truth annotations of over seven classes of road markings.

The unbalanced class distribution of the training and test images is shown in Tables 3.2 and 3.3 respectively, and the histogram in Figure 3.9. Here it's clear that some road marking types occur more frequently than others. Our road marking classification framework was implemented in MATLAB [93] using the Undirected Graphical Models (UGM) Toolbox by Mark Schmidt [120] for probabilistic graphical models.



Figure 3.8: Data collection vehicle: the vehicle is shown with a camera rig mounted on the roof [24].

Class	Description	Abbr.	Number of Examples	Proportion
1	Single Boundary	SGL	794	12.19%
2	Double Boundary	DBL	1012	15.54%
3	Separator	DIV	1491	22.89%
4	Zig-zag	ZIG	523	8.03%
5	Intersection	INT	287	4.41%
6	Boxed Junction	BJ	1566	24.04%
7	Special Lane	SPL	841	12.91%

Table 3.2: Training set class distribution.

Class	Description	Abbr.	Number of Examples	Proportion
1	Single Boundary	SGL	778	11.81%
2	Double Boundary	DBL	961	14.59%
3	Separator	DIV	1471	22.34%
4	Zig-zag	ZIG	549	8.34%
5	Intersection	INT	395	6.00%
6	Boxed Junction	BJ	1573	23.89%
7	Special Lane	SPL	858	13.03%

Table 3.3: Test set class distribution.

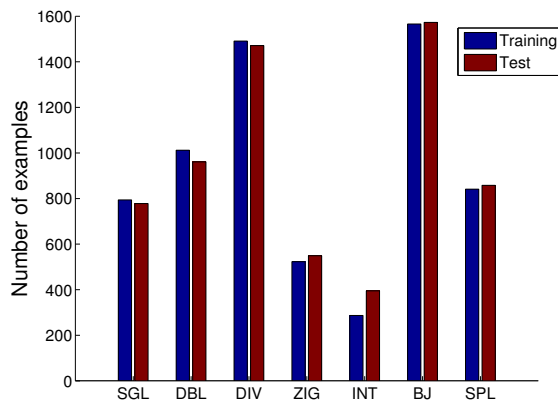


Figure 3.9: Histogram showing the training and test set class distributions of Tables 3.2 and 3.3.

3.5.2 Results

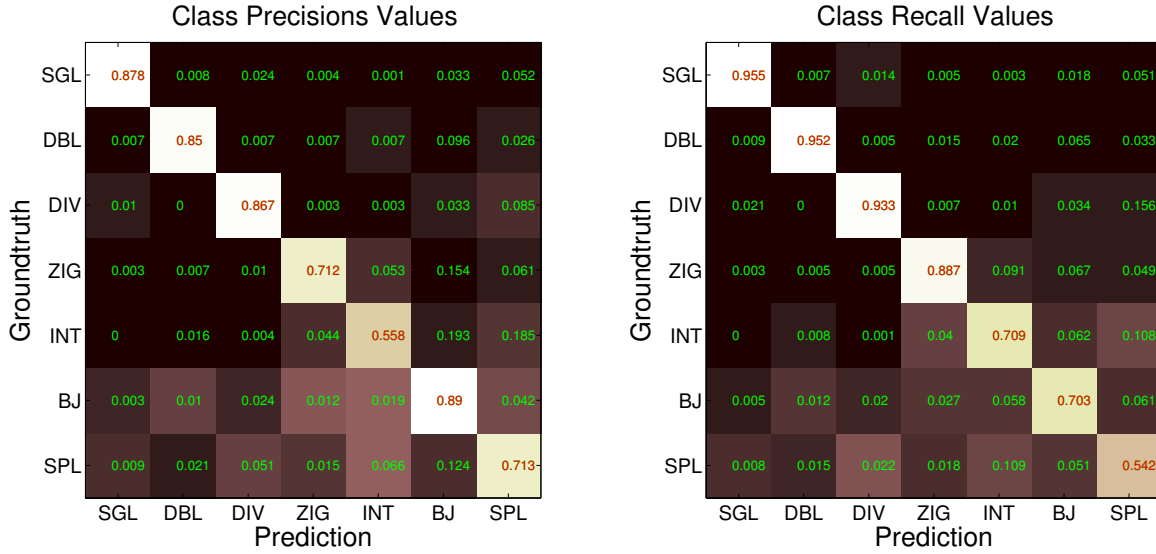
This section details our experimental results based on the datasets and classification framework described previously. We also demonstrate applications of road marking classifications for situational awareness in vehicles. We evaluate our system in two ways: (a) without relational learning (similar to Duchow and Kortner [32]) and (b) with relational learning (our proposed method).

3.5.2.1 No Relational Learning

First, we consider the performance of our road marking classification framework with no relational learning. To do this, we train a RUSBoost classifier (see Section 3.4.3.2) using the features described in Section 3.4.3.1. Although the three features, $f_{propleft}$, $f_{propright}$, and f_{midist} take other road markings into account, the relationships between adjacent node labels (or cliques) is not explicitly modelled.

The resulting per class precision and recall confusion matrices are shown in Figures 3.10a and 3.10b respectively. For a given confusion matrix, the diagonal elements are large ratios (close to 1) for good classification performance (i.e. the ground truth and predicted classes correspond exactly). Thus, a strong bright (using our colour map) diagonal in the confusion matrix illustrates good classification performance.

Our experiments show per class precision values ranging between 0.56 and 0.89 and per class recall ranging from 0.54 to 0.96. In particular, we note that the intersection class precision is very low. Examples belonging to this class are commonly misclassified as special lanes, boxed junctions or sometimes zig-zag approach. The per class recall values for the special lane class are also very low (only slightly better than a random guess). Classes predicted as separator, single boundary, zig-zag approach, intersection



(a) Per class precision values using only boosting classifier (b) Per class recall values using only boosting classifier

Figure 3.10: Confusion matrices for our test set produced using only the RUSBoost classification framework (no relational information)

or boxed junction actually belong to special lane. These misclassifications are partly due to the fact that the labels (class assignments) of neighbouring nodes are not explicitly taken into account by the RUSBoost classification framework. Thus, the fact that a road marking next to one already classified as boxed junction class is very likely to also belong to that same class is not taken into account. The RUSBoost model is discriminative and assigns road marking classes based on a strong learner given the feature functions. We show in the next section how introducing a relational learning framework, which models how adjacent markings are likely to belong to the same class, reduces the misclassification errors.

3.5.2.2 Classification with Relational Learning

We now consider road marking recognition performance when relational information between the nodes is explicitly modelled using a Conditional Random Field formulation. The resulting per class precision and recall confusion matrices are shown in Figures 3.11a and 3.11b respectively.

When relational information is taken into account, the per class precision values range between 0.74 and 0.93 whilst the per class recall ranges from 0.69 to 0.94. The precision is lowest for the intersection and special lane classes as their visual appearance is similar. The intersection class is also under represented in the training data (accounting for just 4.41% of the training examples). Similarly, the recall is lowest for the intersection and special lane classes. Figure 3.12 explains why confusion between these two classes occurs

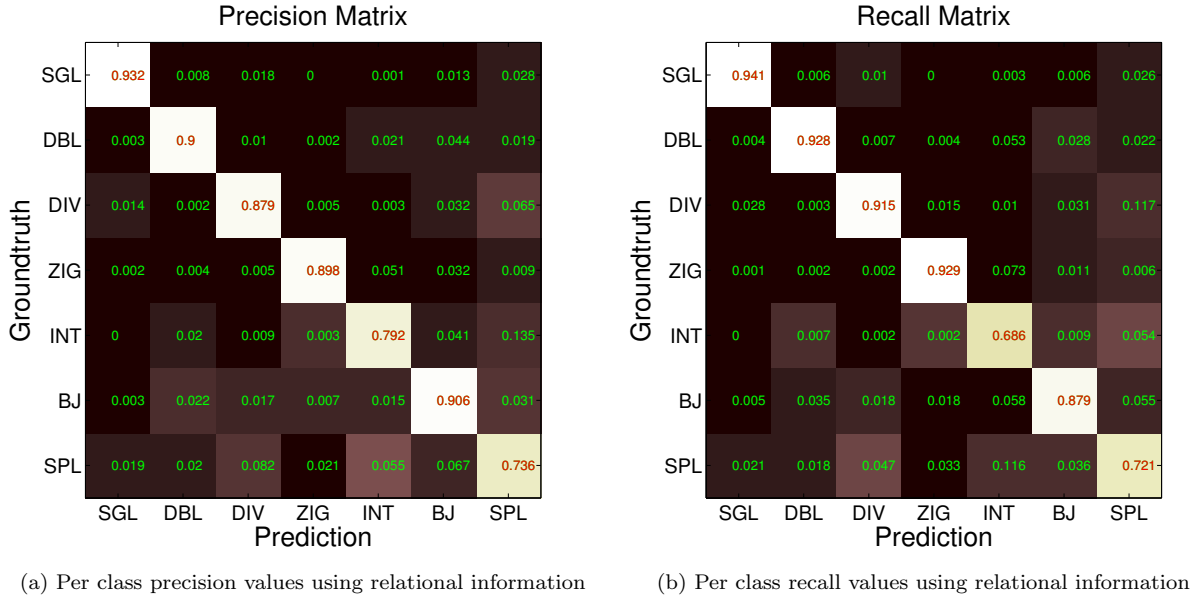


Figure 3.11: Confusion matrices for our test set produced using a relational learning framework

Class	Description	Change in Precision (%)	Change in Recall (%)
1	Single Boundary (SGL)	5.4	-1.4
2	Double Boundary (DBL)	5	-2.4
3	Separator (DIV)	1.2	-1.8
4	Zig-zag (ZIG)	18.6	4.2
5	Intersection (INT)	23.4	-2.3
6	Boxed Junction (BJ)	1.6	17.6
7	Special Lane (SPL)	2.3	17.9

Table 3.4: Percentage improvements in terms of precision and recall when relational learning is used.

since their visual appearance is very similar (both classes are marked by short repetitive road markings). The other classes show very good performance with precision and recall values of between 0.88 to 0.94. We quantify the per class percentage improvement gained by adding relational information to our road marking classification framework in Table 3.4. Here, performance improvements of up to 23% are achieved across classes.

Some example images showing classification results, with the road markings coloured to reflect the classes described in Table 3.1, are shown in Figure 3.13. To better visualise the errors, we plot these same images in Figure 3.14 where green represents correctly classified road markings, and red lines show the misclassifications.

Thus far, we have only considered the classification of individual road markings in a given image. For complete situational awareness however, it is important to also interpret the entire road scene in the



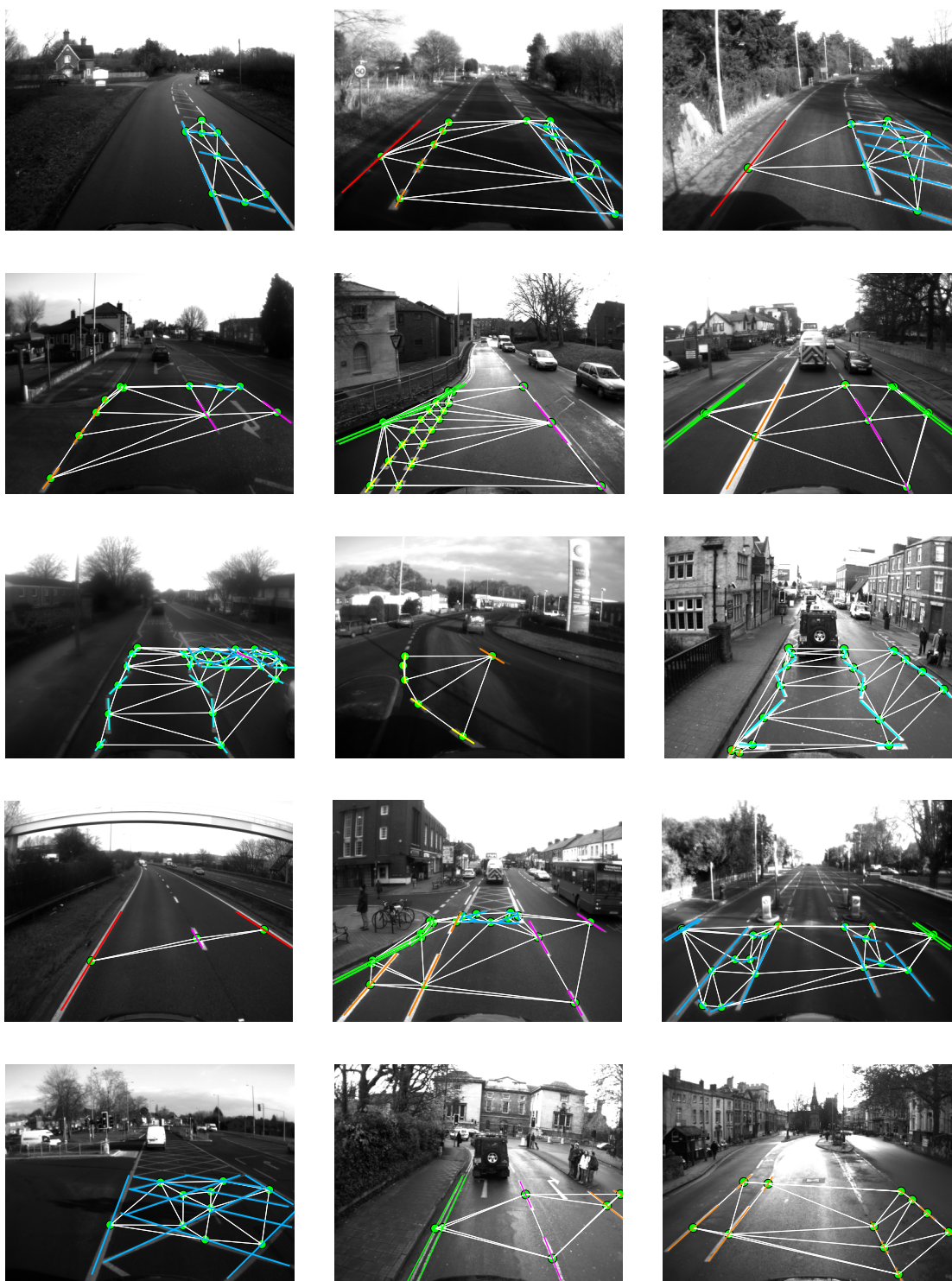
Figure 3.12: Special lane and intersection type road markings can often have very similar visual appearance. Both these classes are marked by short repetitive road markings.

context of present road markings thus reading the road scene. We address this in the next section where we consider recognising pedestrian approaches as well as no drive areas based on the classified road markings.

3.5.2.3 Road Scene Semantics from Road Markings

In order to fully read and understand the road, we perform a further classification decision step to detect zig-zag approaches. We define a zig-zag approach (or boxed junction) to exist if a proportion of over 0.7 of the classified road marking segments belong to class zig-zag (or boxed junction). This threshold was empirically determined to capture the fact that zig-zag approaches (and boxed junctions) occur when the majority of road markings in the road scene belong to that class. The total number of zig-zag approaches in our test dataset is 40 and our classifier was able to correctly recognise 35 of these, some of which are shown in Figure 3.15. This 88% prediction accuracy achieved for zig-zag approaches allows a vehicle to anticipate traffic-light controlled pedestrian crossings thus predicting the upcoming road semantic topology, preparing for the possibility of encountering humans on the road, and gaining a sense of situational awareness beyond what is on the road.

Another important semantic feature of roads is areas in which a vehicle may not be stationary (but may possibly momentarily drive through). Figure 3.16 shows examples of these where two types are considered: intersection boxed junctions and separator boxed junctions. We collectively classify these as belonging to class boxed junction and our test dataset contains 140 unique examples of these. Of these examples, our road marking classification framework was able to correctly recognise 121 boxed junctions, achieving a



COLOUR KEY: SGL-red DBL-green DIV-magenta ZIG-cyan INT-yellow BJ-blue SPL-orange

Figure 3.13: Classification results where each colour represents a different class (see Table 3.1). The graphical model structure is also superimposed on each image.



Figure 3.14: Classification results from our system (green lines are correctly classified, whilst red lines represent misclassifications). These images correspond to Figure 3.13.



Figure 3.15: Examples of zig-zag approaches correctly classified by our algorithm. Our classification framework achieves a prediction accuracy of 88% for identifying zig-zag approaches and hence anticipating upcoming traffic light controlled pedestrian crossings.



Figure 3.16: The two types of road markings we class as “boxed junctions”. These describe areas of the road that a vehicle should not be stationary in (but possibly quickly traverse) such as separator boxed junctions (left) and intersection boxed junctions (right).

prediction accuracy of 86%. Examples of correctly classified boxed junctions are shown in Figure 3.17.

Classifying boxed junctions in addition to zig-zag approaches further enhances the sense of situational awareness a vehicle acquires from interpreting the meaning of road markings. This allows the vehicle to not only read road semantics but to also comprehend their meaning (i.e. a zig-zag approach implies an upcoming traffic light controlled pedestrian crossing and a boxed junction marks an area where the vehicle should not be stationary).

The system presented in this Chapter classifies road markings in images using a Conditional Random Field (CRF) at each timestamp, independent of previous timestamps. For improved situational awareness, the temporal information should also be modelled by classifying road markings over consecutive frames. Two possible approaches to achieve this can be explored in future work. Firstly, the classified road markings could be tracked using an Extended Kalman Filter with a motion model based on the vehicle dynamics. This tracking step is useful for smoothing the CRF road marking class predictions over consecutive frames. This approach is similar to [79] - multiclass multimodal detection and tracking in urban environments. The second, more sophisticated approach also uses information about the vehicle’s ego motion to estimate the position, geometry and class of road markings at time step t based on the previous time step $t-1$. A temporally integrated Conditional Random Field can be formulated by connecting the CRF to the results of the previous time step and transferring the information contained in the previous node marginals into the current labelling task [128]. Siegemund et al. showed that this temporal connection can be provided via a Kalman filter and modelled using an additional potential function in the CRF [128].



Figure 3.17: Examples of boxed junctions correctly classified by our algorithm. Our classification framework achieves a prediction accuracy of 86% for identifying boxed junctions and hence finding areas of the road where no stopping or traversal is allowed.

3.5.2.4 Understanding Misclassification

Although our system performs very well, misclassifications still occur. Figure 3.18 shows examples of the labelling ambiguity which accounts for some of the misclassified classes. Specifically, (a) and (b) show examples of different types of separator markings (magenta). However, the short markings in (b) actually represent a special lane for buses whilst the ones in (a) are ordinary separator lanes. These look very similar and can be difficult even for a human to distinguish. In (c), the special lane markings (orange) and boxed junction markings (blue) are somewhat ambiguous since they could equally be correctly labeled as all belonging to the same boxed junction class. The same applies for (f), where the separator marking (magenta) could equally be a boxed junction marking (blue). The rounded curve road shape in (d) means that the road markings are labelled in piece-wise line segments (see the green double lines). Finally, no special line markings are present in (e) even though a special lane exists.

Some worst case misclassifications from our system are shown in Figure 3.19 where, (a) special lane markings occurring in the middle of the road are misclassified as separator markings (b) a boxed junction is misclassified (c) separator markings are misclassified as special lane markings (d) special lane markings viewed from a difficult sideways orientation were misclassified as boxed junction (e) boundary and special lane markings are misclassified as separator markings, and (f) the two adjacent single boundary markings are misclassified as a special lane.

The road marking classification system detailed thus far exploits the underlying structure of road markings in order to classify them within a Conditional Random Field (CRF) statistical relational learning framework which captures uncertainty. In recent years, Recursive Neural Networks (RNN) have gained some popularity in scene understanding literature and were shown by [113] to outperform CRFs for multi-class natural scene segmentation on the Stanford background dataset. An alternative relational learning framework for road marking classification could be implemented with a Graph Neural Network (GNN) which extends RNNs and directly processes undirected cyclic graphs [89]. Nodes in the graph represent objects/concepts whilst edges represent their relationships, captured by a parametric transition function expressing the dependence of a node on its neighbourhood [89]. Importantly, CRFs directly model the conditional probability distribution and output meaningful marginal probabilities for the class assignment of each node, thus uncertainty is captured and further reasoning about confidence in the labels can be made. Neural Networks however enjoy linear time computational complexity for classifying an example given a trained network.

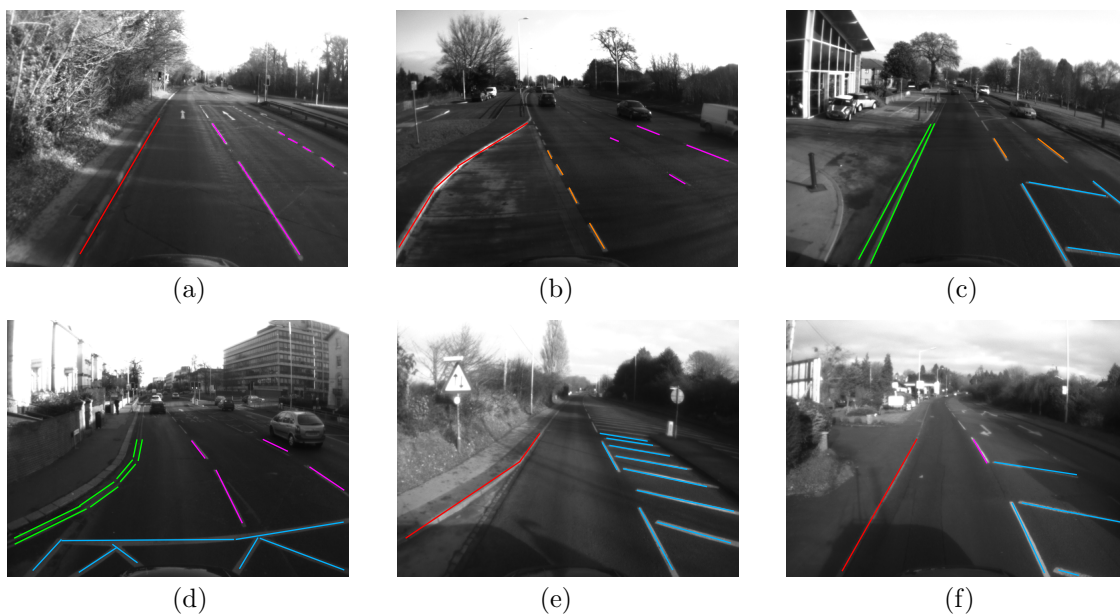


Figure 3.18: Understanding misclassifications. The colours represent road marking classes according to Table 3.1. Refer to the text for explanations.

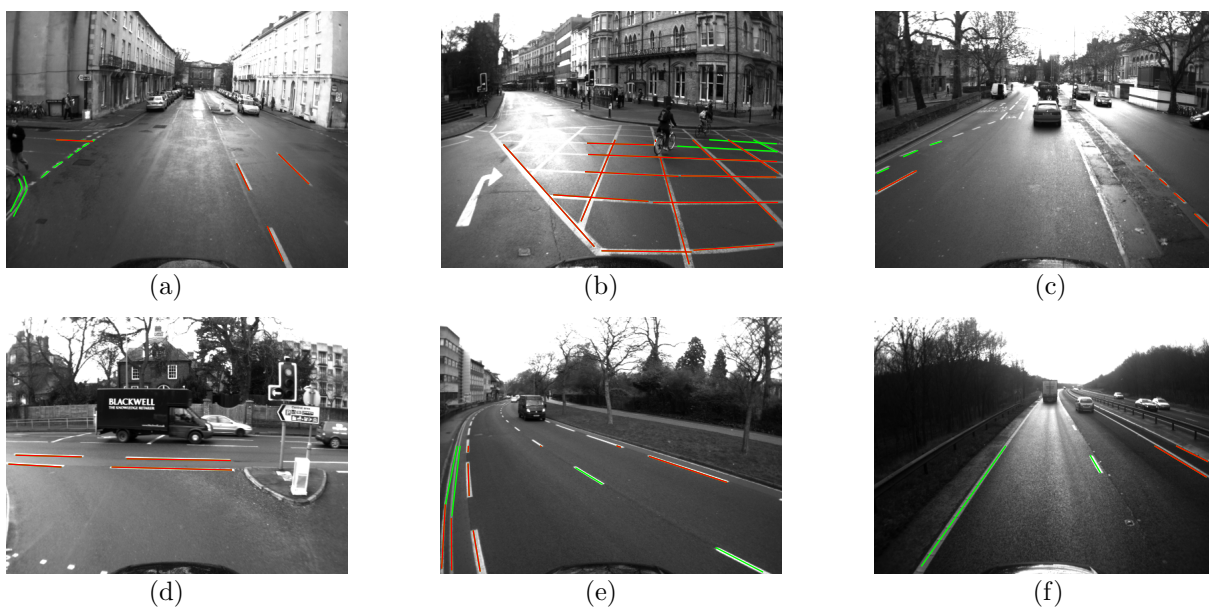


Figure 3.19: Worst case misclassifications. The red markings are misclassified and the green markings are correctly classified. Refer to the text for explanations

3.6 Chapter Summary

In this Chapter, we presented a framework for situational awareness based on reading and interpreting road markings. Specifically we classified road markings into seven distinct classes: single boundary, double boundary, separator, zig-zag, intersection, boxed junction and special lane. Our system achieved a precision of between 74% and 93% and corresponding recall of 69% to 94% across all classes.

Furthermore, we used our road marking classification output for vehicle road scene semantic interpretation in two ways. Firstly, we showed how a vehicle can anticipate traffic light controlled pedestrian crossings based on recognising zig-zag approaches. Our experimental results reported a prediction accuracy of 88% for zig-zag approach recognition and pedestrian crossing anticipation. This allows a vehicle to prepare for the possibility of encountering pedestrians on the road thus gaining a dynamic sense of situational awareness, like humans. Secondly, we demonstrated the recognition of boxed junctions at intersections and as separator lanes allowing a vehicle to detect areas where no stopping is permitted (no drive regions), achieving a prediction accuracy of 86%. Overall, our system allows a vehicle to read and interpret the road by extracting the rules of the road encoded in the type of painted road markings visible. In this manner, both the local and upcoming road topology can be inferred, resulting in true semantic situational awareness for our vehicle.

Thus far, our situational awareness Google Protocol Buffer data structure described in Chapter 2 contains semantics pertaining to road marking classes in a given image and, where applicable, the associated situations (pedestrian approaches or no drive regions). In the next Chapter, we consider the algorithms used to add information about road boundaries to this data structure.

Chapter 4

Reading the Surface: Road Boundaries

In this Chapter, we continue with the previous theme of reading the road surface, by extracting the road boundaries. A road boundary refers to the edge of a carriageway and could be interfaced by curbs, grass, shrubs, or simply the tarmac edge. At intersections, road boundaries are absent, however pseudo boundaries can be defined there. Specifically, we formulate road boundary estimation as a curve fitting problem using a novel barycentric formulation based on geometric prior information from Highway Traffic Engineering literature. Furthermore, our unusual sensor configuration comprises of just a single vertically downward facing 2D SICK LIDAR thus requiring the development of new algorithms for road boundary point detection and surface segmentation. Finally, we evaluate our road boundary estimation algorithm by computing relevant road semantics such as continuity, curvature, Mean Squared Error for curve fitting, and with the help of OpenStreetMap priors, the locations of intersections. This boundary related semantic information will also be captured in our Situational Awareness data structure introduced in Chapter 2.

4.1 Literature Review

Road boundary estimation is important for geometric situational awareness in autonomous driving. Knowledge of road boundaries allows a vehicle to understand where right of way should be given, regions where driving speeds should be reduced and areas in which the vehicle may safely drive. For road perception, range sensors such as LIDARs are preferred because they are not affected by changes in illumination,

shadows and poor lighting conditions. These factors adversely effect cameras resulting in potentially poor algorithm performance. Furthermore, imaging at dusk or night time is possible with a LIDAR thus opening a new exciting field of night time autonomous navigation. Images, on the other hand, provide rich colour and texture information which can be exploited for feature detection at both local and global image scales. Also, although 3D depth information can be estimated using stereo vision [76], a LIDAR sensor calibrated with a GPS; or a Velodyne provides accurate xyz point clouds.

Nearly two decades of road tracking work exists in the literature, thus we limit our review mainly to the most relevant work in LIDAR based road boundary estimation (such as summarised in [54]).

4.1.1 Vision Based Approaches

Vision based approaches for road boundary estimation can generally be divided into model based segmentation, Digital Elevation Mapping (DEM), or vanishing point [75], based algorithms. The first approach involves extracting the road boundaries by segmenting the road surface from images. Typically, such approaches model the road, using binary synthetic images, as either a triangle [17], a probability map created from averaging road images [5], or a trapezoid [117]; and refine this model based on a modified distance transform and morphological operations [17], a road scene SVM classifier built using SIFT descriptors combined with a pixel-based road detection algorithm [5], or surface segmentation results from a relevant algorithm [117]. Broggi simply assumes that road boundaries coincide with the locations of road markings on the edges [15]. Road segmentation approaches are insufficient for situational awareness applications, as they make no attempt to assign a geometric model to the detected road boundaries.

The second major approach involves the extraction of dense stereo 3D point clouds followed by the creation of Digital Elevation Maps [106, 128, 105, 58]. Here, grids are created and the local height value is computed for each grid using 3D triangulation or averaging. This captures the elevation of the road, and edge detection [106], machine learning approaches such as CRFs and MRFs [128, 127, 51], or RANSAC approaches [105, 104] are applied to extract the boundaries. The advantage of DEM approaches is that the curb heights can also be estimated, however the false positive rate depends on the quality of the stereo reconstruction [106, 104, 127, 105, 128, 51]. Machine learning methods for finding boundaries in DEM have been shown to outperform edge detection methods. Also, cubic polynomial representations of curbs (such as those used in [104, 127, 128]) fail to reconstruct all road boundary geometries and cannot model transitions of arcs to tangents well.

The third approach uses the vanishing point to approximate the road boundaries, such as [75].

4.1.2 Clothoid Models

A clothoid describes a curve with curvature linearly dependant on arc length l , given by $c(l) = c_0 + c_1 l$ where c_0 and c_1 are the initial curvature and curvature rate respectively. Clothoid based models for road boundary estimation are popular and have been used, for example, by [136, 110, 142]. In [136], the road boundaries are estimated from laser scan data as a sequence of joined clothoid curves, where a Kalman filter estimates the clothoid parameters. In similar work, texture features are used within a binary road/non-road classification framework to detect road boundary points which are tracked by a particle filter that estimates the parameters of a clothoid model [133]. However, clothoid models for describing road curvature have two limitations: firstly, for usage in Kalman filters, they are approximated by polynomial functions thus the torsion of the vehicle relative to the highway, occurring during lane changes, cannot be accurately modelled [142]. Secondly, accuracy is compromised at high curvature regions thus Corridori and Zanin proposed a two-clothoid model [23] for road boundary estimation which was only tested on simulated flat roads where the camera is assumed to be moving exactly over the road centre.

4.1.3 LIDAR and Velodyne Based Approaches

Laser based approaches to road boundary estimation can be divided into sequential algorithms that process each laser scanline in turn, and global algorithms that process the entire 3D point cloud at once. Algorithms that process each laser scanline focus on extracting road boundary features such as curbs, berms or shoulders using wavelet-based edge detection [110], Gaussian differential fitting [156], Haar wavelets [137], a linear road surface model [69, 147, 52, 73] or a parabolic surface model [130].

Peterson et al. proposed a wavelet-based edge detector for extracting curbs, berms and shoulders from single laser scanlines. The road shape was modelled as a clothoid where a Particle filter was used to estimate the clothoid parameters [110]. In the DARPA Urban Challenge, team Little Ben used downward angled LIDAR and side-mounted Hokuyo sensors for ground surface segmentation. To fit a ground plane, the range values from the scan were median filtered to remove noise and the LIDAR scan (x_i, y_i) points were minimised according to $\min_{m,b} \sum_i f(z_i - mx_i - b)$ for an error function f that was quadratic close to zero but became more linear for larger values [12]. Given this ground plane, the curb was detected based on obstacle height above ground using the side mounted laser.

In order to deal with multiple variants of curb structures, DARPA's team Boss assume that the road surface is relatively flat and process each LIDAR scan independently, thus considering data along a single height dimension only [137]. Before extracting features, the data was clustered to remove outliers and

only dense ground points were linearly resampled and considered for feature extraction. The heights were analysed by the discrete Haar wavelet to estimate the scaled average slopes (proportional to Haar coefficients) followed by post processing heuristics to extract the curb points. Later, [156] process the laser elevation signal using Gaussian differential filtering techniques to find the road region, and a SVM classifier with variance of ground points features to refine this ground plane segmentation. The line representation of the projected signal on the ground plane is compared to a simple road model to find the curb points [156].

Line fitting road boundary extraction approaches are based on estimating the ground plane in a laser scanline as a flat surface which can be approximated by a line segment. The Hough transform was used in [69] for line fitting and three distance and angle thresholds were manually set to extract the road edge points which were tracked using a Kalman filter and probabilistic decision rule. This approach assumes that curbs are always present in each scanline resulting in the need for multiple Kalman filter tracking to discard incorrect road boundary points. In similar work, [147] formulate road curb detection and tracking using an Extended Kalman Filter (EKF) with a straight line process model, assuming a minimum curb height of 25mm. Han et al. [52] used a downward angled LIDAR to extract line segments from a scanline and classify them as either road or obstacle using roll and pitch angle heuristic measures to increase accuracy. No parametric model was used for boundaries. Finally, Kim et al. built a localisation system based on the detection of curbs [73]. The authors propose that the longest straight line in a single laser scanline represents the road surface, with the curbs located at the endpoints. The Hough transform was used to extract the lines and three thresholds, based on Hough parameters, were manually set to determine the curb points.

The line fitting approaches described above make the assumption of a flat ground plane therefore ignoring the cross slope or cant of roads which allow for the proper drainage of water on the road surface. Smadja et al. use four SICK lasers for road extraction and account for the cant by modelling the road as a parabola instead [130]. Road segmentation was achieved by simply fitting a parabola to each scanline using RANSAC and classifying the inliers as road surface points based on slope break points. The road boundary points were recursively estimated by inlier voting using a third order polynomial RANSAC approximation and a Non Uniform Rational Bezier Spline (NURBS) was used to parameterise the final road boundary output.

Some previous work combines both LIDAR and vision. Multiple lane detection using five cameras, thirteen LIDARS and a Velodyne was proposed by Huang et al. [58] where road markings were detected using vision and curbs using LIDAR and combined to estimate centrelines of travel lanes which are tracked

and fused with a prior map (Route Network Definition File - RNDF) to output the final lane estimates. A Velodyne and SICK LIDARs were used to detect road boundaries by creating a DEM and casting a ray from the vehicle outwards throughout this map. The first small obstacle encountered by a ray was assumed to be a curb or berm and the road boundaries were parameterised by a spline model. This work placed a strong prior on the road width and poor performance was noted in shadow conditions. In later work, Huang et al. formulate a novel probabilistic model of curvature to track lane boundary curves with lateral uncertainties [57].

The entire 3D point cloud was also used in [95], where DARPA’s team Junior compared the range values returned between neighbouring Velodyne beams since the sensor scans in concentric circles of different constant radii. To detect curbs close to the vehicle, the IBEO and SICK LIDAR sensors were used to classify as obstacles all objects detected in close range, including curbs.

The most related previous work is by Yang et al. who implemented curb detection based on 1D filtering of estimated individual scan lines [153]. The GPS corrected time difference between consecutive points is used to partition point clouds into a sequential set of scan lines. A road area is then estimated based on thresholding height difference of points within three sliding windows along a single scan line (see Figure 4.1a). Curb points are detected by combining elevation, point density and slope change metrics (each with their associated thresholds) within two perpendicularly oriented sliding windows (see Figure 4.1b). A final clustering and merging step is done to remove noise and a B-spline model is then interpolated to ensure continuity. This approach depends on six hand tuned parameters: the maximum time between two sequent points δt , the number of points in the moving window n , the minimum height of curbs δ_{curb} , the maximum height of the curbs δh_{filter} , the maximum height difference of the pavement $\delta h_{pavement}$, the minimum slope difference between the curb and road surfaces $\delta \theta$. The authors also note that poorly chosen parameters can result in poor performance, although very good performance was reported with their chosen parameters and dataset.

In general, per laser scanline processing algorithms are popular due to their computational speed advantages, accuracy, and simplicity. Also, the cost of 2D SICK LIDARs is significantly less than a 3D Velodyne sensor. Our proposed road boundary estimation algorithm differs from the previous work in four ways: firstly, unlike other approaches that rely on multiple sensors, we only use a single SICK LIDAR. Secondly, we mount our sensor vertically downward facing in a push-broom fashion resulting in most of the sensor points falling on the actual road surface as shown in Figure 4.2. This unique sensor mounting allows us to develop new robust, yet elegant algorithms for road surface and curb detection. Thirdly,

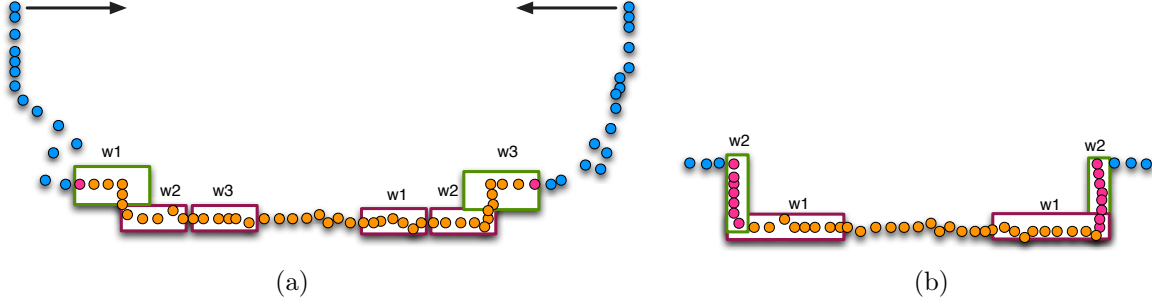


Figure 4.1: Road surface segmentation and curb point detection: (a) Detection of the road surface using three sliding windows from either side using: $\delta h_{pavement} \leq \delta z_1 \leq \delta h_{filter}$, $\delta z_2 \leq \delta h_{pavement}$ and $\delta z_3 \leq \delta h_{pavement}$ where z_i is the height range of the points in the i^{th} sliding window. (b) Curb points are detected using two 1D sliding windows and testing for the elevation ($\delta z_{curb1} \leq \delta h_{pavement}$ and $\delta z_{curb2} \geq \delta h_{curb}$), point density ($Length_2 < L < Length_1$) and slope changes ($\delta angle = abs(\alpha_1 - \alpha_2)$ and $\delta angle = \delta \theta$). These parameters are manually tuned by [153] to give good results. Image reproduced from [153].

unlike [156, 69, 153], we adopt a principled feature transformation and automatic change point detection (with confidence levels) procedure that does not require us to manually set thresholds for detecting road boundary points in a single laser scanline. Finally, we base our formulation of a parametric model for road boundaries on Highway Traffic Engineering geometric road design principles rather than approximations using high order polynomials or clothoids, and extract 3D boundaries.

We now turn our discussion to the geometric prior knowledge we exploit in our work.

4.2 Domain Knowledge: Geometric Priors

Domain knowledge, in the context of road boundary estimation, refers to the the rules and manuals used by Highway Engineers to design and build roads. The geometric design of roads is a branch of traffic and highway engineering concerned with the principles and rules for positioning physical elements - such as vertical and horizontal curves, lane widths, cross sections and parking bays - on roadway. Such standards exist to maximise safety and traffic efficiency whilst minimising environmental cost and efficiency of road construction. In the UK, the Highway Agency publishes the Design Manual for Roads and Bridges (DMRB) [4] and the Manual of Contract Documents for Highway Works (MCHW) [3] detailing the geometric and design requirements for road construction. The DMRB sets a standard of good practice, principally developed for truck roads and it is up to local authorities to decide to what extent the documents in the manual are appropriate in any particular situation whilst ensuring that local road schemes do not compromise health and safety, have negative environmental impact, or result in poor value for money [4].

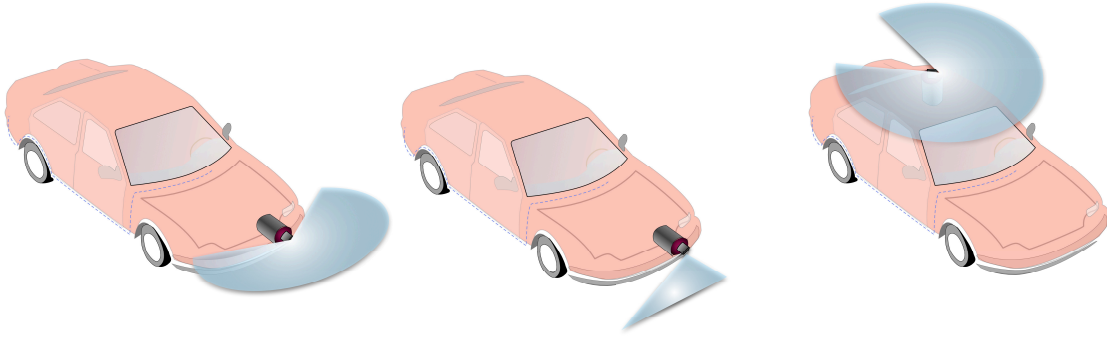


Figure 4.2: Three possible 2D SICK LIDAR and 3D Velodyne sensor configurations. The left vehicle shows a forward facing SICK LIDAR sensor which is well suited for object detection and mapping applications. This is by far the most commonly adopted configuration. The central vehicle shows our vertically downward facing SICK LIDAR configuration where the scanning plane of the sensor is perpendicular to the ground plane. This unique configuration is seldom adopted in previous work - we therefore develop new feature detection algorithms for this case. The right vehicle shows a Velodyne sensor which scans around the vehicle generating a complete 3D point cloud. Although the cost of Velodynes is decreasing, they remain prohibitively expensive for low cost autonomous driving applications.

Other countries worldwide also publish technical documents detailing geometric road design policies, such as the Guide to Road Design published by AUSTROADS in Australia, Characteristics of Roads and Motorways document published by the Director Generale des Autoroutes et des Routes in Belgium, the Manual of Geometric Design Standards for Canadian Roads published by the Transportation Association of Canada, and the Geometric Design Policy for Rural Roads published by the South African National Roads Agency [77]. Many countries' geometric road design standards were initially based on the American Association of State Highway and Transportation Officials (AASHTO) document, thus standards around the world display similarities in fundamental alignment design principles and philosophies, and in quantitative guidelines on basic design parameters [77]. However, Krammes and Garnham note that differences in policy emphasis and concern have led to more advanced guidelines on certain geometric elements, whilst some quantitative guidelines have been adapted to account for differences in local conditions and experience [77].

Importantly, domain knowledge provides geometric priors useful for developing models for road boundary estimation.

4.2.1 Cross-Sectional Elements

The cross section of a road is in the direction perpendicular to the driving lanes (see Figure 4.3). The main cross sectional elements are the travel lanes, shoulders and medians. The marginal elements are median and roadside barriers, gutters, sidewalks, side slopes, curbs, drains and ditches which we are interested in



Figure 4.3: Typical cross sections of British roads: a rural all purpose dual carriageway bounded by grass (left) and an urban road bounded by curbs (right). Images taken from [4], Volume 6.

automatically extracting for road boundary estimation. Figure 4.5 shows a typical cross section for a two lane motorway. Volume 6 of the DMRB document publishes typical important cross sectional features occurring on roads summarised by Figure 4.4. We are specifically interested in deriving the road boundaries as continuous curves outlining the carriageway (or drivable area). The distance between the boundaries of the road is the road width which depends on the number of lanes the carriageway supports.

On motorways, medians are sometimes used to separate lanes travelling in opposite directions and can be raised, flush or depressed. These barriers are longitudinal systems used to prevent an out of control vehicle from possibly crossing over into traffic travelling in the the opposite direction.

One of the most prominent cross-sectional structures used on urban roads are curbs: raised vertical or sloping concrete structures used to delineate pavement edges and pedestrian walkways, control drainage, reduce right of way and show road boundaries in an aesthetically appealing way [44]. The elevation changes resulting from curbs make them ideal candidates for automatic detection of the road edge from LIDAR data (see Figure 4.6) and we exploit both the height and orientation changes of curb geometric structure in our feature detection algorithms. Our algorithm is however ignorant to the type of curb present and only extracts features for road boundary estimation. Further background about cross-sectional elements and the design vehicle, speed and hourly volume is presented in Appendix A.

4.2.2 Vertical Alignment

The vertical alignment of a road consists of straight sections (tangents) connected by vertical curves in the direction of travel. The design of these directly influences the line of sight and headlight distance of vehicles. Appendix A discusses vertical alignment in further detail.

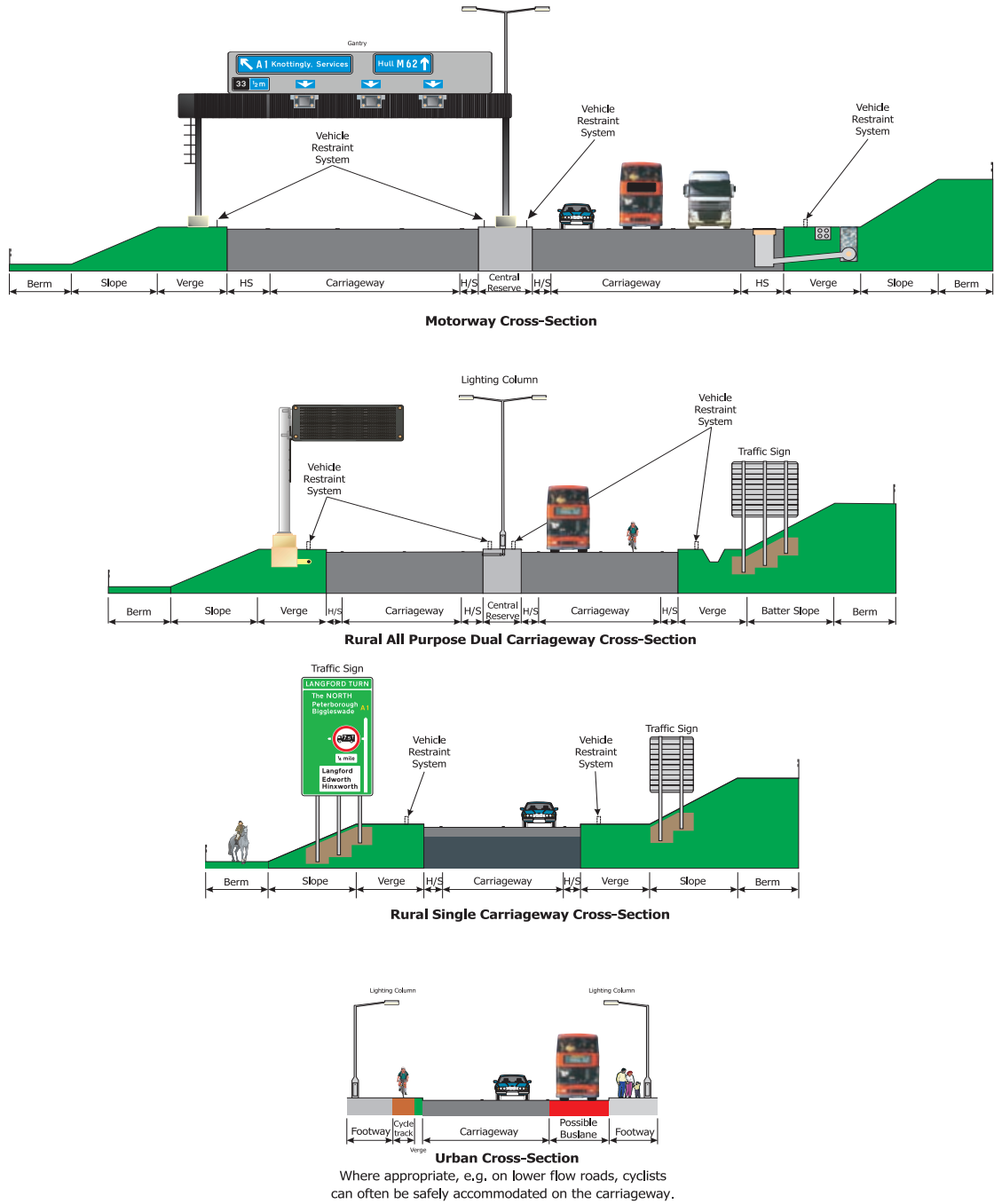


Figure 4.4: Typical features accommodated in cross sections. HS = hard shoulder and H/S = Hard strip. Image from [4], Volume 6. Regions between verges or footways represent areas to be bounded on the left and right by the continuous curves computed by our road boundary estimation algorithm.

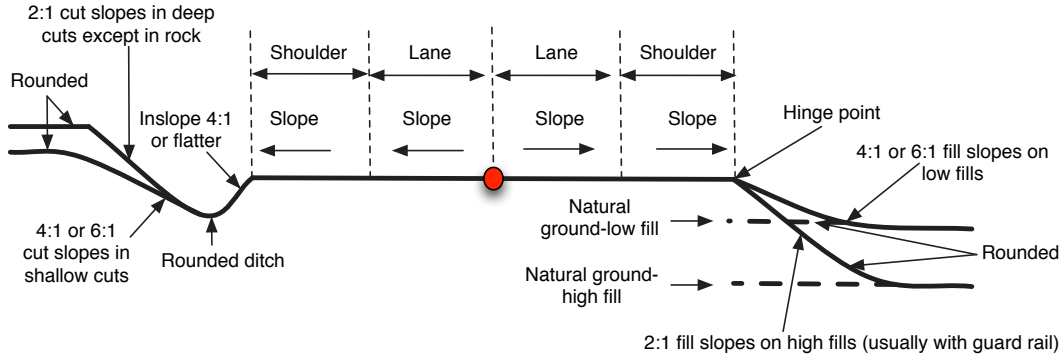


Figure 4.5: Typical cross section for a two lane motorway: the gradients shown are of the form (horizontal:vertical) and the red dot represents the road centre line. Reproduced from [44].

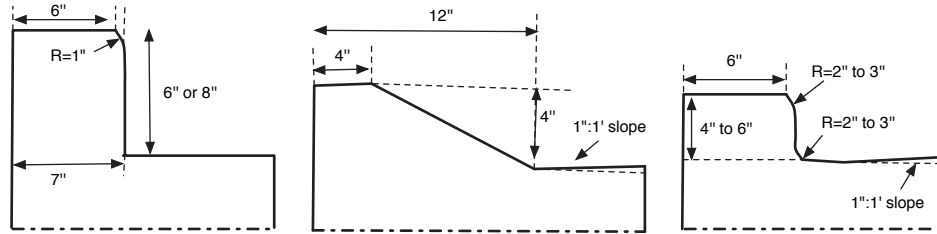


Figure 4.6: Typical vertical (leftmost) and sloping curbs (right) found on highways. Dimensions are shown in inches. Diagrams reproduced from [44].

4.2.3 Horizontal Alignment

The horizontal alignment describes the road boundaries we wish to estimate. Formally, horizontal alignment is the route of the road, defined by *straight tangents*, horizontal *arc curves* and *transitional curves* joining these. These three types of curves represent useful prior information when formulating our road boundary estimation model because it is sufficient for our road boundary model to represent only these types of curves, without compromising on accuracy. Thus a simple but accurate formulation can be derived based on this domain knowledge.

In practice, each section of road has an associated minimum radius of curvature for the circular arc portions which is a function of the design speed u , coefficient of side friction f_s and superelevation e and is thus given by:

$$R = \frac{u^2}{127(e + f_s)}$$

The transitional curve has varying radius in order to avoid the sudden change from a tangent (straight line) with infinite radius to an arc with finite radius. Typically, clothoids have been used as transitional

curves [136, 110, 142]. Furthermore, the degree of curvature is given by $D = \frac{30.48}{R} \frac{180}{\pi}$ for a 30.48m (100ft) long arc. In practice, the Design Manual for Roads and Bridges (DMRB) provides interpolation tables for determining the superelevation rate given a specific road design speed and radius [4]. Four types of horizontal curves exist: simple, compound, reversed and spiral [44] but we focus our discussion mainly on simple curves which form the basis of our road boundary model.

4.2.3.1 Simple Curves

Simple curves are segments of a circle (arcs) and are defined by their radius (tightness) and deflection (extent) which are determined by the speed limit, curve length and line of sight of the driver. Such arcs also have constant superelevation, curvature and radius of curvature.

The alignment geometry is shown in Figure 4.7 where CEB represents the arc shaped road with radius R . The circular arc is either described in terms of its radius or degree of curvature and the degree is given by the arc CEB (angle subtended at the centre by a 100ft arc) or the chord CB . C represents the point of curvature (where the curve begins) and B is the point of tangent (where the curve ends). The point of intersection or vertex is given by D .

Let T represent the length of tangents $\overline{BD}$ and $\overline{CD}$, C^* the chord length of $\overline{CB}$, L_c the curve length, l the curve length from C to any point P , and θ the deflection angle subtended by the curve. If the deflection angle and radius are known, the following geometric relations can be derived:

$$\begin{aligned} T &= R \tan\left(\frac{\theta}{2}\right) & C^* &= 2R \sin\left(\frac{\theta}{2}\right) \\ L_c &= \pi R \left(\frac{\theta}{180}\right) & l &= 2\pi R \left(\frac{\beta}{360}\right) \\ \overline{EM} &= R(1 - \cos(\frac{\theta}{2})) & \overline{ED} &= R\left(\frac{1}{\cos(\frac{\theta}{2})} - 1\right) \\ \overline{PN} &= x = R \sin(\beta) & \overline{LP} &= y = R(1 - \cos(\beta)) \end{aligned}$$

Crucially, the horizontal alignment influences the distance the driver is able to see ahead. Care must therefore be taken to ensure a safe sight distance for the driver so that approaching vehicles can be seen in good time and changing road curvature can be anticipated. The stopping sight distance on horizontal curves (clear distance from the centerline of the lane nearest the obstruction) is given by $R(1 - \cos(\frac{28.65S}{R}))$ [44], where S is the sight distance and R is the radius of the centerline of the lane nearest the obstruction.

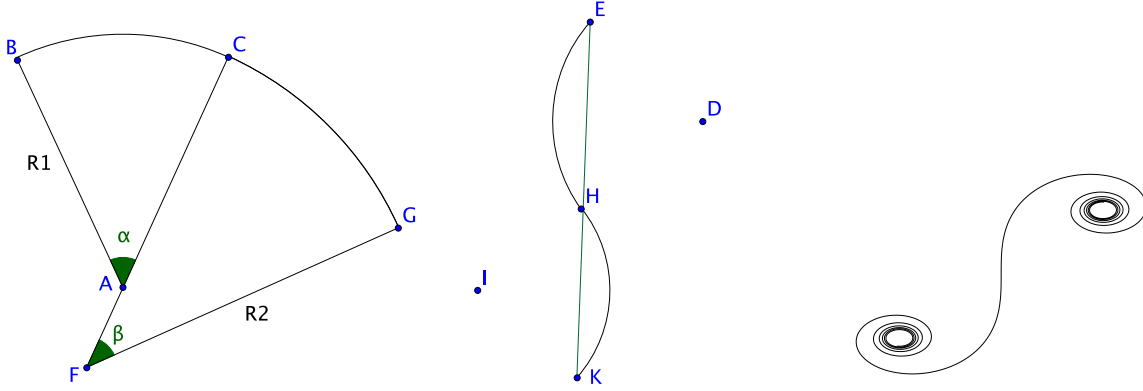


Figure 4.8: A compound curve with two radii R_1 and R_2 and common tangent at C (left). A reverse curve with parallel tangents (centre) and an example of a clothoid (transitional curve) with changing radius (right).

4.3 Sensor Configuration

Our road boundary estimation system only requires a single vertically downward orientated SICK LIDAR sensor mounted on the front of the vehicle. A GPS unit mounted on the roof is used for global referencing purposes. We develop our algorithms in the laser sensor frame only and use the GPS to create a 3D point cloud referenced in the global world coordinate frame. The vertically mounted laser scanning configuration illustrated by Figure 4.2 is an unusual but elegant simplification for our work. In the Figure, three possible configurations are presented: the left vehicle shows a forward facing SICK LIDAR sensor well suited for object detection and mapping applications (by far the most commonly adopted configuration). The central vehicle shows our unique vertically downward facing SICK LIDAR configuration where the scanning plane of the sensor is perpendicular to the ground plane (seldom adopted in previous work), and the right vehicle shows a Velodyne sensor which scans around the vehicle generating a complete 3D point cloud.

The UK Highways Agency Road Research Information System (HARRIS-1 and HARRIS-2 vehicles) also adopt similar vertically mounted laser scanner configurations for the collection of road surface data (rutting, deformation etc.) used for manual assessment of pavement conditions at traffic speed, measurement of road shape and visual condition [81]. Such a configuration results in the majority of the sensed points lying on the ground plane (road surface) as shown in Figure 4.9, however such points are generally more noisy than those further out (due to the sensor characteristics). The advantage of sensing many road surface points is that the ground plane can be segmented using clustering methods as we later show. This

allows for the development of new simple, yet elegant algorithms for road boundary feature detection. The vertical sensor configuration does not “look ahead” onto the road and is therefore unsuitable for online applications such as obstacle or road marking detection. We are however interested in populating the Situational Awareness data structure introduced in Chapter 2, which represents road semantic information that can be queried online as the vehicle drives along a previously surveyed route. Thus this unusual sensor configuration is optimal and affordable for our application.

4.3.1 Absolute Sensor Calibration

Although our boundary estimation algorithm requires only 2D LIDAR sensor data, for situational awareness, a vehicle should have knowledge of 3D road layout geometry relative to its own coordinate frame. The vehicle’s coordinate frame origin is taken to be at the GPS/INS unit on the roof. The 2D LIDAR is mounted vertically (perpendicular to the ground plane) at the front of the vehicle. To generate the 3D point cloud, a transformation $\mathbf{T}_{\text{INS-LASER}}$, is applied to project laser points from the sensor frame into the GPS/INS frame, as shown in Figure 4.9. For simplicity, we hand measure this extrinsic transformation and adopt the standard NASA SE(3) transform convention with the sensor base frame at the sensor origin. The transform is therefore given by:

$$\mathbf{T}_{\text{INS-LASER}} = \begin{bmatrix} \mathbf{R}_{AB} & \mathbf{t}_{AB} \\ \mathbf{0} & 1 \end{bmatrix}$$

where $\mathbf{R}_{AB}$ and $\mathbf{t}_{AB}$ are the rotation and translation matrices respectively. Examples of a 3D point cloud generated using this transformation are shown in Figures 4.10 and 4.11b.

The global coordinate frame is taken to be the World Geodetic System (WGS 84), a standard coordinate frame for the earth used by GPS devices. We calculate a transformation matrix from the laser sensor frame to the global coordinate frame, $\mathbf{T}_{\text{G-LASER}}$ as:

$$\mathbf{T}_{\text{G-LASER}} = \mathbf{T}_{\text{G-INS}} * \mathbf{T}_{\text{INS-LASER}}$$

where $\mathbf{T}_{\text{G-INS}}$ is just the Euler transformation matrix derived from the vehicle pose (X, Y, Z, roll, pitch and yaw) at time t , and $\mathbf{T}_{\text{INS-LASER}}$ is defined as previously. Knowing $\mathbf{T}_{\text{G-LASER}}$ allows us to visualise the projection of the LIDAR sensor output onto Google Maps as shown in Figure 4.11a, where the detected laser boundaries and road markings are shown. The GPS/INS accuracy is however not perfect because exact location estimates are impossible to obtain from the sensor due to drift, and atmospheric

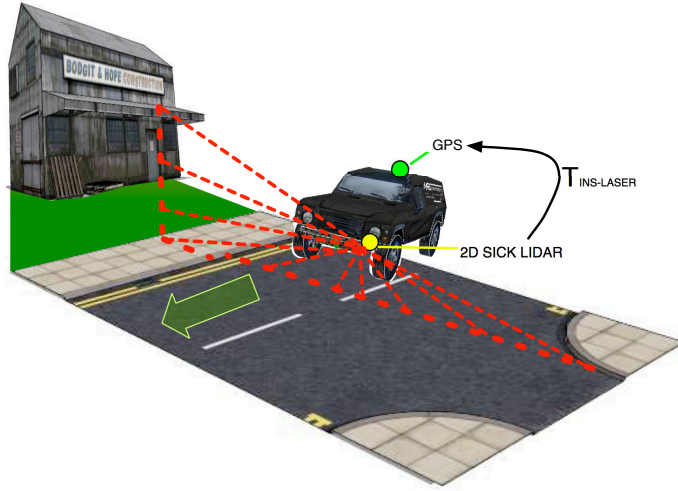


Figure 4.9: The location of the 2D SICK LIDAR (yellow dot) and the GPS/INS unit (green dot) on our autonomous vehicle; and the laser scanline (shown in red). To create a 3D point cloud as the vehicle traverses the environment, a transformation matrix $\mathbf{T}_{\text{INS-LASER}}$ from the laser coordinate frame to the vehicle's INS coordinate frame is estimated. The red lines show some of the vertical scan lines from the laser which are approximately perpendicular to the ground plane. The road surface model and Old Factory Building is taken from Google Sketchup (Trimble 3D Warehouse)[145][144].

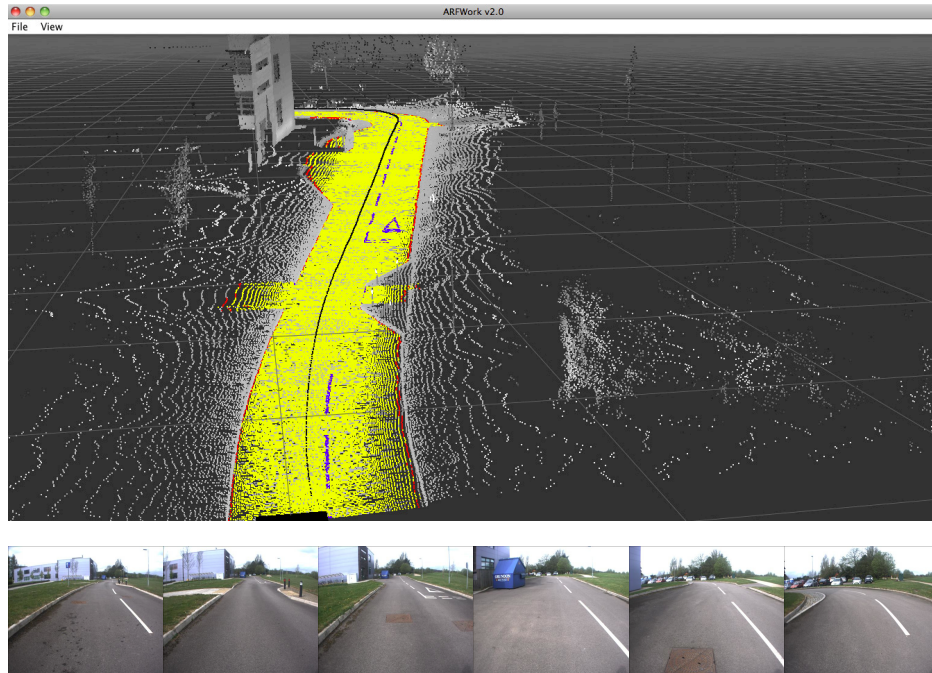


Figure 4.10: A 3D point cloud created using $\mathbf{T}_{\text{INS-LASER}}$. The segmented road surface (yellow), road boundary points (red) and road markings (blue) are results from our algorithms. The bottom row shows the corresponding images for this section of road. Notice how even pedestrian intersections are correctly segmented as road surface points.

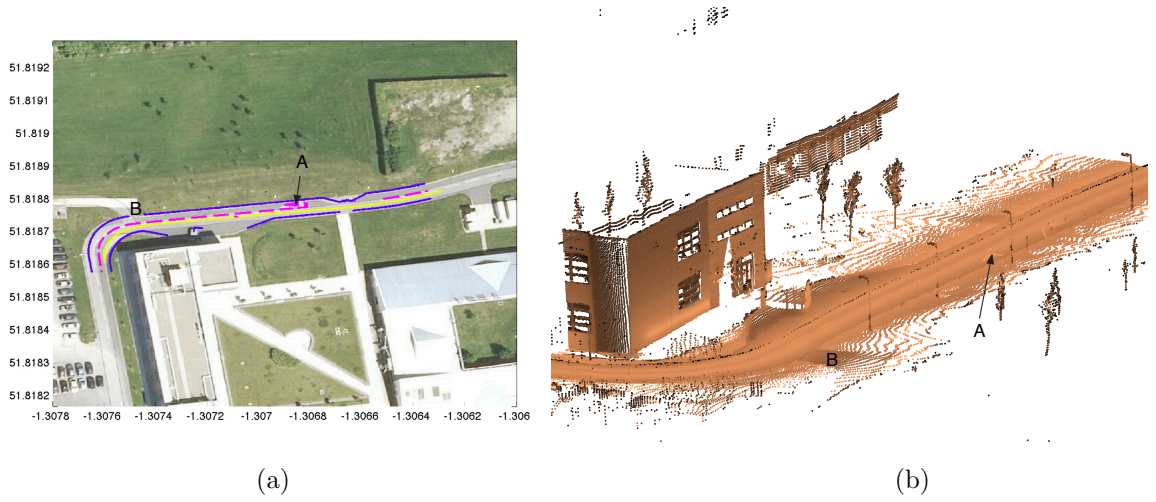


Figure 4.11: (a) A projection of the laser sensor data (raw output from our boundary estimation algorithm) at Begbroke Science Park onto Google Maps, resulting from the transform $\mathbf{T}_{\mathbf{G}-\mathbf{LASER}}$. The blue, magenta and yellow points represent the detected road boundaries, road markings and vehicle GPS trajectory. (b) The 3D point cloud generated from the transformation matrix $\mathbf{T}_{\mathbf{INS}-\mathbf{LASER}}$. Corresponding points between the two figures are marked A and B.

and multipath effects as evidenced by the shift in Figure 4.11a. Therefore projecting laser points onto Google Maps is useful only for visualisation purposes.

We now discuss our feature detection algorithms in detail.

4.4 Road Boundary Point Detection

Boundary point detection is the first step for road boundary estimation and involves extracting 2D laser points belonging to the road boundaries. We define a road boundary as the region on the interface between a carriageway and curbs, grass, sand etc. Coleman et al. define three types of edges present in range data (images): step edges, crease edges and smooth edges. Step edges occur at range discontinuities, crease edges when two surfaces meet and smooth edges in regions where the curvature is discontinuous but the surface is smooth [20]. Step edges therefore occur when curbs are present.

Our boundary point detection algorithm consists of three main connected modules summarised by Figure 4.12. In this section, we discuss the implementation of each of these modules in detail.

4.4.1 Surface Segmentation

The first module for road boundary point detection is surface segmentation which refers to the automatic extraction of road surface points in a given laser point cloud. Because our LIDAR sensor is mounted

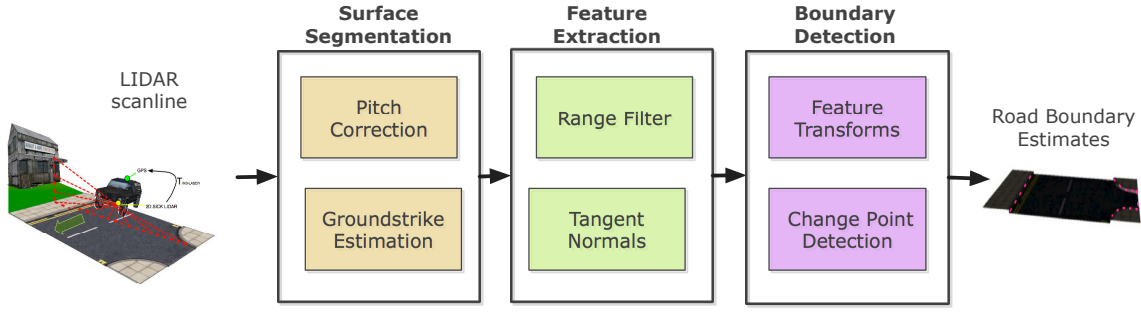


Figure 4.12: A block diagram showing the main modules making up our road boundary point detection system: Surface Segmentation, Feature Extraction and Boundary Point Detection.

on the vehicle itself, its global position (and hence the sensor origin in the global frame) varies with the vehicle’s pitch. This means that simple vertical thresholding of laser point heights taken on real roads will not result in accurate road surface segmentation. We therefore exploit our sensor configuration already described in Section 4.3 to develop a simple, yet elegant, sequential surface segmentation algorithm based on the observation that the vast majority of our laser points are clustered on the road surface plane. We begin by performing pitch correction on the laser scanlines.

Pitch Correction Pitch correction refers to the process of compensating for errors arising from the changing pose of the vehicle as it traverses the environment. Our road boundary estimation algorithm implements per scanline processing similar to Yang, Bisheng et al. [153]. As the vehicle traverses the environment it experiences forward translation motion and rotation about three orthogonal axes referred to as the roll, pitch and yaw. In developing our algorithms, we make the assumption that the ground plane is flat thus we need to estimate the rotation matrix $\mathbf{R}_p$ to correct the for the vehicle’s changing pitch. In order to achieve this, we use prior knowledge of the sensor’s location in each laser scanline (i.e. the local sensor origin) and assume that points in a small region directly below the sensor must lie on the ground plane. This assumption is valid because the SICK LIDAR sensor is mounted vertically down in line with the vehicle bumper which always lies above the road (see Figure 4.26). We fit a straight line through these points to estimate the pitch rotation angle θ and associated rotation matrix as:

$$\mathbf{R}_p = \begin{bmatrix} \cos(\theta) & b\sin(\theta) \\ c\sin(\theta) & \cos(\theta) \end{bmatrix}$$

where $(b, c) = (-1, 1)$ for counterclockwise rotation or $(b, c) = (1, -1)$ for clockwise rotation. Figure 4.13 illustrates the pitch correction process, showing how the pitch angle is estimated from the data.

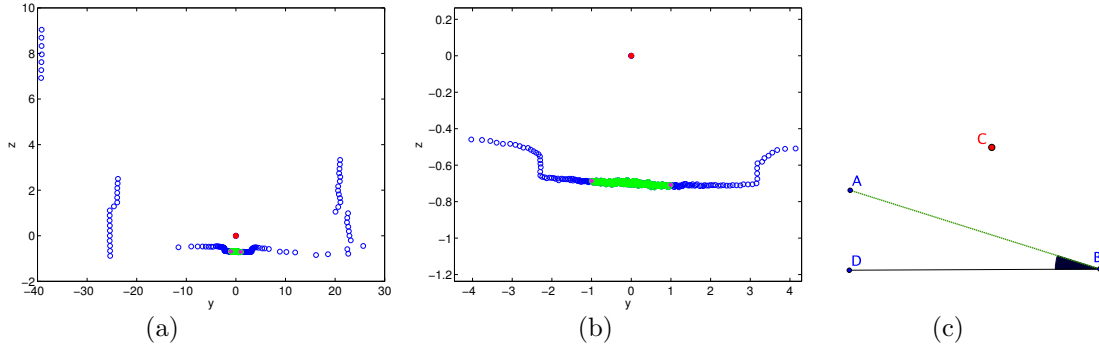


Figure 4.13: Estimating the vehicle pitch angle, θ . (a) A single LIDAR scanline where the red dot represents the known sensor origin and the green points are the ground points considered for approximating the pitch angle. (b) A magnified view of (a). (c) A close up simplified schematic where point C is the sensor origin and AB is the line fitted to the local ground points. The vehicle pitch angle is given by $\angle ABD$.

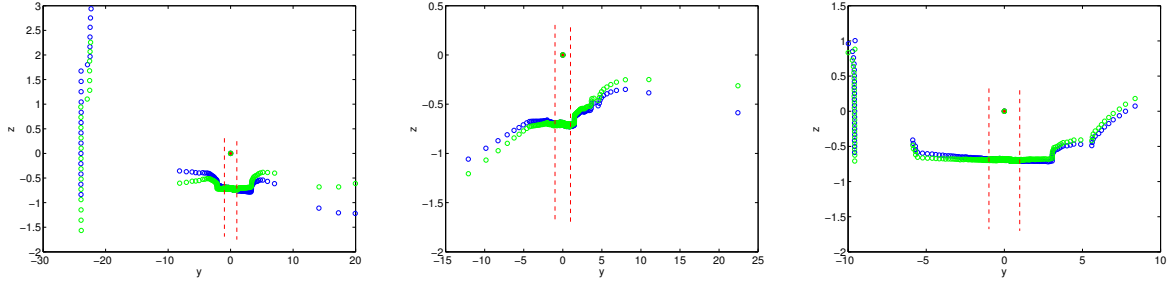


Figure 4.14: Examples showing the estimation of vehicle pitch using a small region (between the dashed vertical lines) around the sensor origin (0,0) to estimate the point cloud rotation matrix $\mathbf{R}_p$. The blue dots show the original scanlines and the green dots are the resulting pitch corrected laser scanline outputs.

Figure 4.14 shows the resulting point cloud rotation on three distinct scanlines, where the dashed vertical lines mark the region of interest for the pitch correction.

Groundstrike Estimation Groundstrike estimation refers to the extraction of laser scanline points belonging to the ground surface. Thus, given a pitch corrected laser scanline, groundstrike estimation reduces to a clustering problem as a result of our vertically downward facing sensor configuration. Our LIDAR sensor (sensor origin) is positioned such that the region below it always falls on the ground surface, where the vehicle drives (see Section 4.3 for a discussion). In our experiments, we considered two clustering algorithms: a standard Minimum Spanning Tree Clustering (MSTC) algorithm and a new Greedy Clustering algorithm (GC) which we propose. Both these algorithms cluster points based on distances and require a threshold distance th as input. The value of this threshold is sensor specific - it depends on the SICK LIDAR operating range and the height at which it is mounted. We empirically determined this

threshold using a small holdout dataset. The proposed Greedy Clustering algorithm exploits the fact that the sensor location/origin in a laser scanline is always known and proceeds by initialising the groundstrikes as a small region directly below the sensor origin. The algorithm then proceeds to grow this initial region in the left and right directions by adding points whose neighbouring distances to the region boundaries are below the threshold th . Algorithm 4.1 provides implementation details.

Algorithm 4.1 Greedy Clustering Algorithm for Groundstrike Estimation

input: (y, z) (scanline) and th (neighbourhood threshold)

output: (yN, zN) (groundstrikes) and $indices$ (corresponding indices)

$groundstrikes \leftarrow initialiseGroundstrikes(y)$

$gs \leftarrow initialiseGs();$

for $i=1$:points right of region **do**

$d = findNearestPoint(y, z, i, i + 1)$

if $(d < th)$ **then**

$gs(i) \leftarrow i + 1$

else break

end if

end for

$groundstrikes \leftarrow groundstrikes \cap gs$

$gs \leftarrow initialiseGs();$

for i :points left of region **do**

$d = findNearestPoint(y, z, i, i - 1)$

if $(d < th)$ **then**

$gs(i) \leftarrow i - 1$

else break

end if

end for

$groundstrikes \leftarrow gs \cap groundstrikes$

$yN \leftarrow y(groundstrikes) \quad zN \leftarrow z(groundstrikes) \quad indices \leftarrow groundstrikes$

Similarly, the groundstrikes can be estimated using a MSTC algorithm. First, a graph is created by connecting points (nodes) in the scanline to form a minimum spanning tree (no loops). Clusters in this graph are formed by removing graph edges with length exceeding th , the given threshold. Due to our unique sensor configuration, it follows that the majority of laser scanline points belong to the road surface thus the largest cluster from the MSTC is in fact the groundstrikes as demonstrated by Figure 4.15.

In our experiments, we found our new proposed greedy algorithm (that exploits the sensor configuration) to have comparable performance to the MSTC algorithm in terms of accuracy but enjoy a 5.5 times faster processing time of 11.23 milliseconds per scanline. Some examples of the resulting groundstrike estimation output, based on the new proposed Greedy Clustering algorithm, are shown in Figure 4.16 where trees, buildings, bushes and other vehicles are correctly excluded from the groundstrike regions.

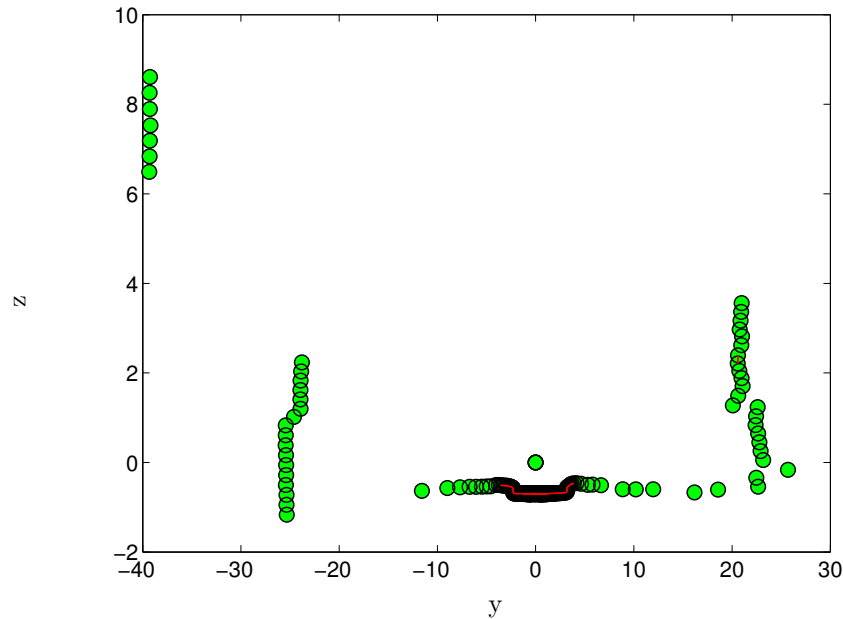


Figure 4.15: Minimum Spanning Tree Clustering for groundstrike estimation: each laser point (circle) is a node in the graph and the red lines are the connected clusters after the clustering threshold is applied. Only 2 connected clusters (with more than one member) remain and the largest one is the groundstrikes.

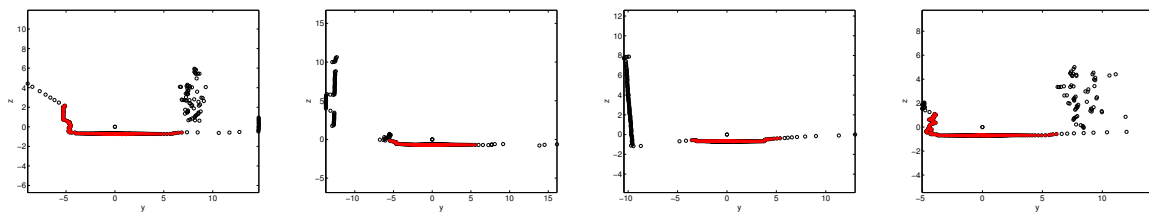


Figure 4.16: Single scanlines from the LIDAR sensor (each laser scan point is shown as a dot). The red region is the output of our groundstrike estimation algorithm based on Greedy Clustering. Notice how trees, buildings, bushes and other vehicles are excluded from the groundstrike regions, and most of the points in a scanline belong to the ground region which is a special property of our sensor configuration.



Figure 4.17: A 3D model of a road with and without curbs with road boundary points superimposed in pink. Model taken from Google Sketchup (Trimble 3D Warehouse)[145]

In this Section we showed how our sensor configuration allows us to develop simple yet elegant clustering algorithms for road surface segmentation. We now discuss the second module of our road boundary point detection system - Feature Extraction.

4.4.2 Feature Extraction

The second module for road boundary point detection is feature extraction. We define road boundary points as the curbs, and in the absence of curbs, the visible boundaries of the road as illustrated by Figure 4.17. The inputs to the two feature extraction algorithms (Range Filter and Tangent Normals) we develop are the groundstrikes estimates from the previous section.

Range Filter The range filter captures the height distribution of a laser scanline. Specifically, given a set of overlapping windows containing height information, we compute the range within each window. The largest range is then assigned to each overlapping window. In order to guarantee a filter that is sensitive to near vertical height changes present at curb regions, we choose a window size $nwind = 3$ and overlap size $noverlap = 2$. Algorithm 4.2 summarises our implementation.

Algorithm 4.2 Range Estimator Filter for height change detection

input: $nwind$ (window size), $noverlap$ (window overlap size), zN (groundstrike points)

output: zR (range estimations)

```

 $idx \leftarrow findOverlapWindowIndices(nwind, noverlap)$ 
 $slidingWindows \leftarrow zN(idx)$      $r \leftarrow range(slidingWindows)$ 
 $zR \leftarrow zeros$ 

for  $i=1$ :all  $idx$  elements do
    for  $p = 1$ :  $noverlap$  do
        if  $zR(idx_{p,i}) < r_i$  then
             $zR(idx_{p,i}) \leftarrow r_i$ 
        end if
    end for
     $zR(idx_{noverlap+1:end,i}) \leftarrow r_i$ 
end for

```

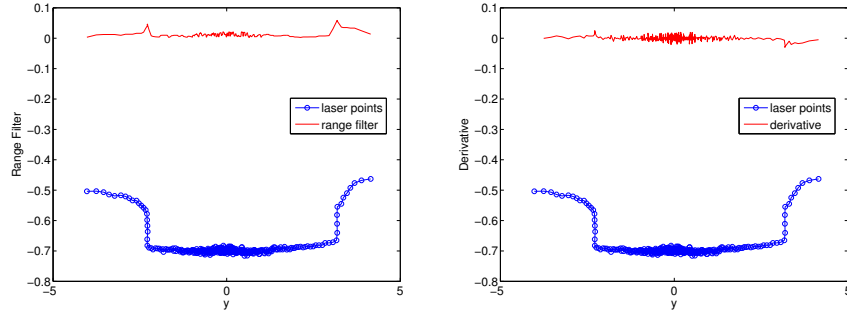


Figure 4.18: A comparison of our Range Filter output (left) to a derivative operation output (right). Notice how the derivative captures the noise characteristics of the sensor and is therefore not a good candidate for reliable feature extraction. Our Range Filter approach however, is able to detect curbs whilst suppressing most of the noise.

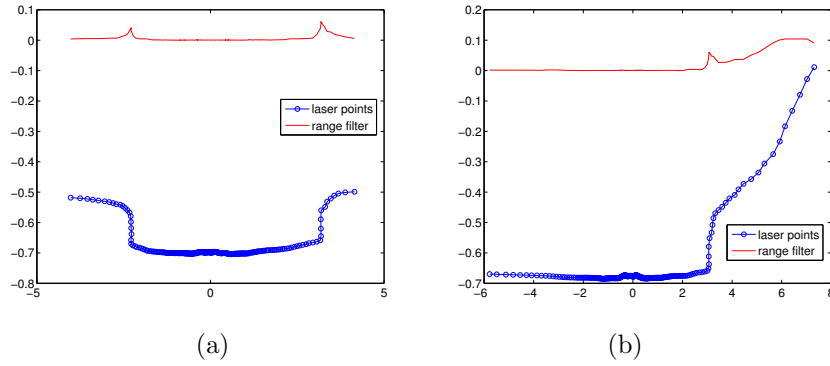


Figure 4.19: The output of our range filter is the feature function $f_{range}(z)$. (a) A scanline with two curbs (b) No curbs present but there is an object at the edge of the road

This feature differs from a differentiation operator which considers the height differences between adjacent or nearby points in a scanline. Because our SICK LIDAR sensor has associated noise characteristics, a differentiation operator will simply capture the sensor noise characteristics resulting in noise amplification as illustrated by Figure 4.18. Contrastingly, our Range Filter exploits the fact that it is the range between local points that changes when a curb occurs - even though the actual sensor measurements themselves are locally noisy. Figure 4.19 illustrates example outputs from our range filter which can detect changes in height along the ground (at potential curb or road edge locations). The output of the filter is the feature function $f_{range}(z)$.

Tangent Normals The Tangent Normals feature captures the local orientations of each point in a laser scanline. Specifically, for a given point, a small neighbourhood of points is considered and the tangent vector joining these points is calculated. The normal to this vector is the feature we use. Care must be taken

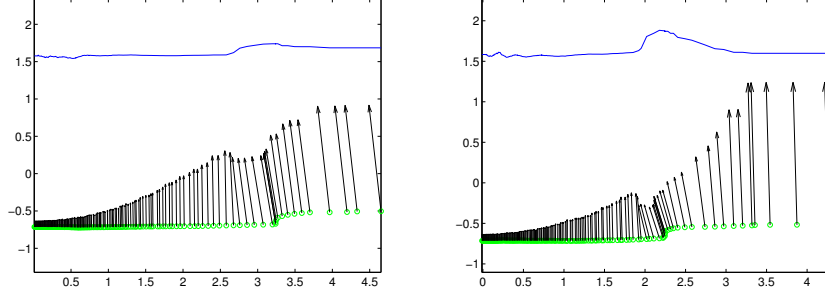


Figure 4.20: Tangent Normals (black arrows) on ground plane points (green). Normals on the curbs differ in direction from those on the ground. The direction magnitude feature function, $f_{direction}(y, z)$ is plotted in blue. The left and right portion of the road is split at the known sensor origin point, hence the two separate plots for a single scanline.

when choosing the neighbourhood size - a very small neighbourhood will not capture the local orientations, whilst a large neighbourhood tends to over smooth the tangent estimates. We empirically determine that a neighbourhood size of $k = 15$ provides discriminative performance for our sensor configuration. Because curbs are vertical structures in the road, the normal directions change near curb points.

To ensure consistency in the normal directions we split the groundstrikes at the known sensor location, flip the resulting right side about the y axis and process each half separately. Figure 4.20 illustrates the Tangent Normals and corresponding direction magnitude. Specifically, this direction magnitude is the Tangent Normal feature function $f_{direction}(y, z)$.

4.4.3 Boundary Point Detection

The third and final step for our road boundary point detection system is boundary point detection. This final stage detects the road boundary points given the tangent normal feature function $f_{direction}(y, z)$ and the range filter feature function $f_{range}(z)$. Both these feature functions have different units (radians and meters respectively). Thus a transform is needed to map these feature functions to a domain where they have comparable magnitudes and can effectively be combined. These feature transforms are discussed next.

Feature Transforms We transform the $f_{direction}(y, z)$ and $f_{range}(z)$ feature functions as a pre-processing step for our final boundary point detection step. First, we scale $f_{range}(z)$ by a factor α , according to

$$\alpha = R(f_{direction}(y, z)) / R(f_{range}(z))$$

where R is the range operator which takes as input a function, and outputs the difference between

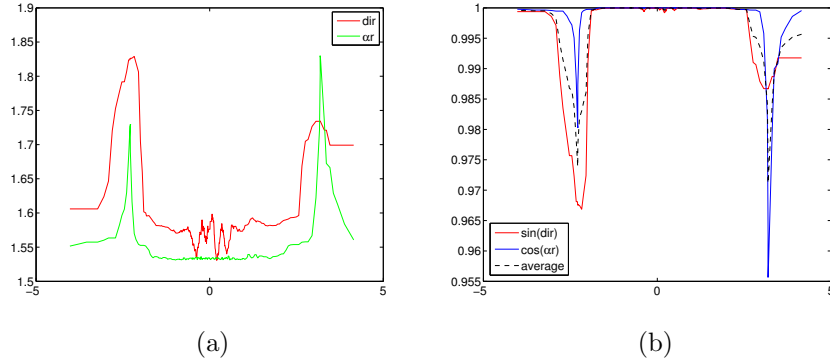


Figure 4.21: (a) The functions $f_{direction}(y, z)$ and $\alpha f_{range}(z)$ are shown in red and green respectively. (b) The transformed feature functions f_d (red) and f_r (blue) and their combination F (black) are shown. F is the input to our change point detection step.

the maximum and minimum value of the function. Given this scale factor the feature transforms are given by $f_d = \sin(f_{direction}(y, z))$ and $f_r = \cos(\alpha f_{range}(z))$. This transform ensures that f_d and f_r have a maximum value of one and can thus be combined to form a new feature function F given by

$$F = \frac{1}{2}(f_d + f_r)$$

Consequently, the new feature function F captures both range and orientation characteristics of the road surface and is used as input to the final Change Point Detection step of our algorithm which produces the road boundaries. Figure 4.21 visualises the feature transforms discussed in this section.

It is worthwhile discussing why the *sine* and *cosine* transform functions are chosen. Curbs are physically defined by vertical changes along the approximately flat road surface, therefore the Tangent Normal feature function $f_{direction}(y, z)$ is $\frac{\pi}{2}$ radians along the flat road surface. The *sine* function transforms the $\frac{\pi}{2}$ directions corresponding to the flat road surface to the value one using $f_d = \sin(f_{direction}(y, z))$. Similarly, for the Range Filter feature function $f_{range}(z)$, small range changes (of magnitude less than $\frac{\pi}{4}$) corresponding to the flat road surface are transformed by the *cosine* transform to the value one using $f_r = \cos(\alpha f_{range}(z))$. Thus our feature transforms result in the road surface points being mapped to the value one as illustrated by Figure 4.21b.

Change Point Detection This section describes the final step of our road boundary point detection algorithm summarised in the block diagram of Figure 4.12. Rather than simply thresholding F , the combined feature function, to extract the road boundary points, we seek a principled approach for determining the road boundary points. Change Point Detection is a procedure for determining whether a

significant change in a given function has occurred, and is a suitable method for detecting road boundary points. Thus, given the combined feature function F , the change point detection algorithm extracts points corresponding to road boundaries (where significant changes occur). To implement this, we follow the technique for Change Point Detection proposed by Taylor et al. in [134]. Given F , the cumulative sum over the elements of F is computed and the change point is the location corresponding to its maximum absolute value. Our trust in this change point is reflected by the confidence value computed via bootstrapping this cumulative sum. The principle behind bootstrapping is that by randomly sampling without replacement the input F , the resulting cumulative sums capture the behaviour of this feature function supposing no change occurred at all. Algorithm 4.3 summarises our implementation of Change Point Detection and specifically, we estimate the variation of S_{diff} if no change occurred and compare this to the original S_{diff} over 100 bootstrapping rounds. Finally, to estimate the road boundary points we begin by assuming that the boundaries occur at the endpoints of the groundstrike region. Then for bootstrapping confidence levels exceeding 99%, the change point is taken to be the road boundary point location.

Algorithm 4.3 Change Point Detection Algorithm with Bootstrapped Confidence Levels

input: F (combined feature function) , $nbootstraps$ (number of bootstrapping rounds),

$left_side$ (left boundary)

output: $Confidence$ (confidence level), $change$ (changepoint)

$\bar{F} \leftarrow ave(F)$

$S_0 \leftarrow 0$

$S_i \leftarrow S_{i-1} + (F_i - \bar{F}), i \in \{1, 2, \dots, size(F)\}$

$S_{max} \leftarrow max(S) \quad S_{min} \leftarrow min(S) \quad S_{diff} \leftarrow S_{max} - S_{min}$

for $k = 1$ to $nbootstraps$ **do**

$\bar{F}' \leftarrow ave(F')$

$F' \leftarrow randsample(F)$

$S_{0b} \leftarrow 0$

$S_{ib} \leftarrow S_{ib-1} + (F'_i - \bar{F}')$

$S_{bmax} \leftarrow max(S_b) \quad S_{bmin} \leftarrow min(S_b) \quad S_{bdiff} \leftarrow S_{bmax} - S_{bmin}$

if $S_{bdiff} < S_{diff}$ **then**

$X \leftarrow increment(X)$

end if

end for

$Confidence \leftarrow X/nbootstraps*100$

if $left_side$ **then** $change \leftarrow index(min(S))$ **else** $change \leftarrow index(max(S))$ **end if**

Our bootstrapping Change Point Detection procedure can be visualised by Figure 4.22 which shows that the bootstrap cumulative sums (shown in red) tend to be of smaller magnitude than the original cumulative sum (shown in blue) when a change has occurred. Similarly, the bootstrapping cumulative sums are of similar magnitude to the original cumulative sum when no change has occurred.

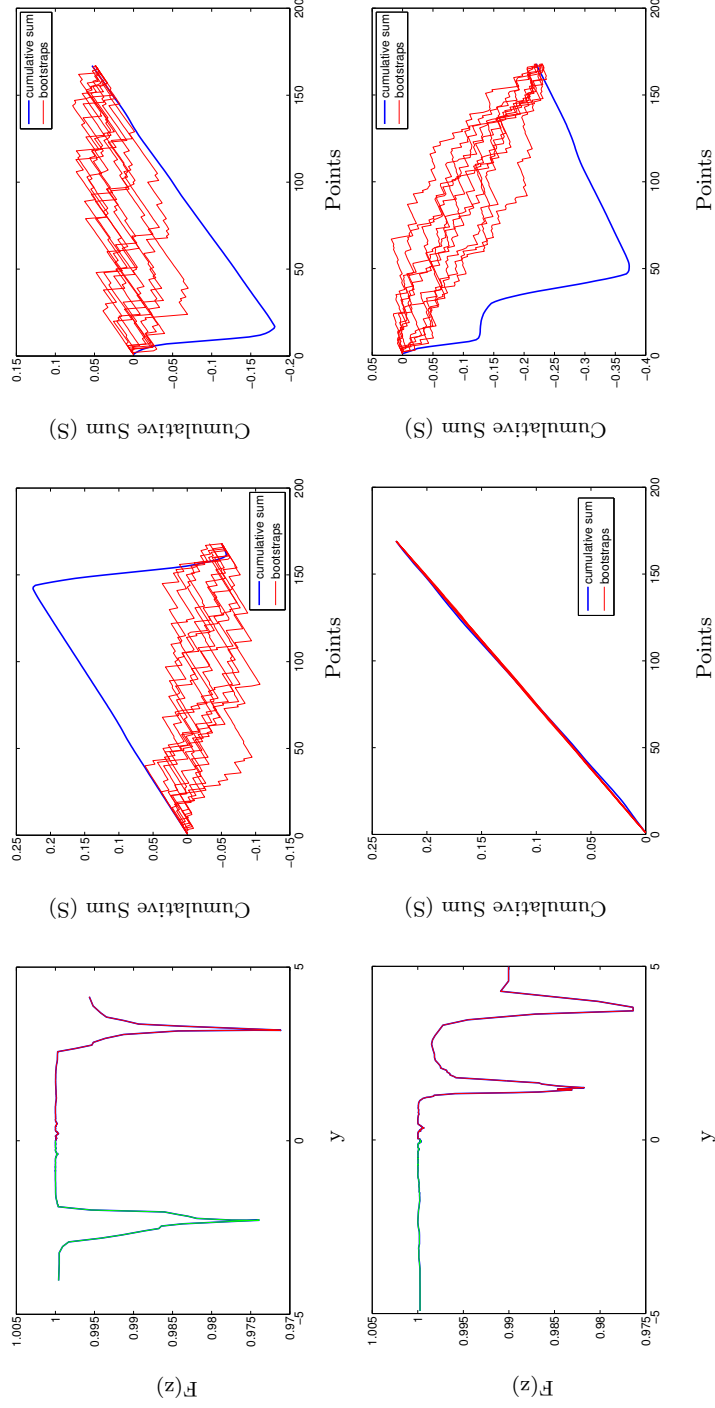


Figure 4.22: To assign confidence levels to our road boundary estimates (where curbs are present), we perform change point analysis on the combined feature function F shown on the left. On the right, we show how the associated confidence level for each detected change point is computed using a bootstrapping technique. For clarity, only the first 10 bootstrapping results are shown in red and the peaks of the cumulative sums (blue) denote the detected change points. The principle behind this bootstrapping procedure is that by randomly sampling F without replacement, the resulting cumulative sums capture the behaviour of this feature function supposing no change occurred. Thus, if a significant change occurs, the bootstrapping cumulative sums (shown in red) are closer to zero than the original function cumulative sum (shown in blue).

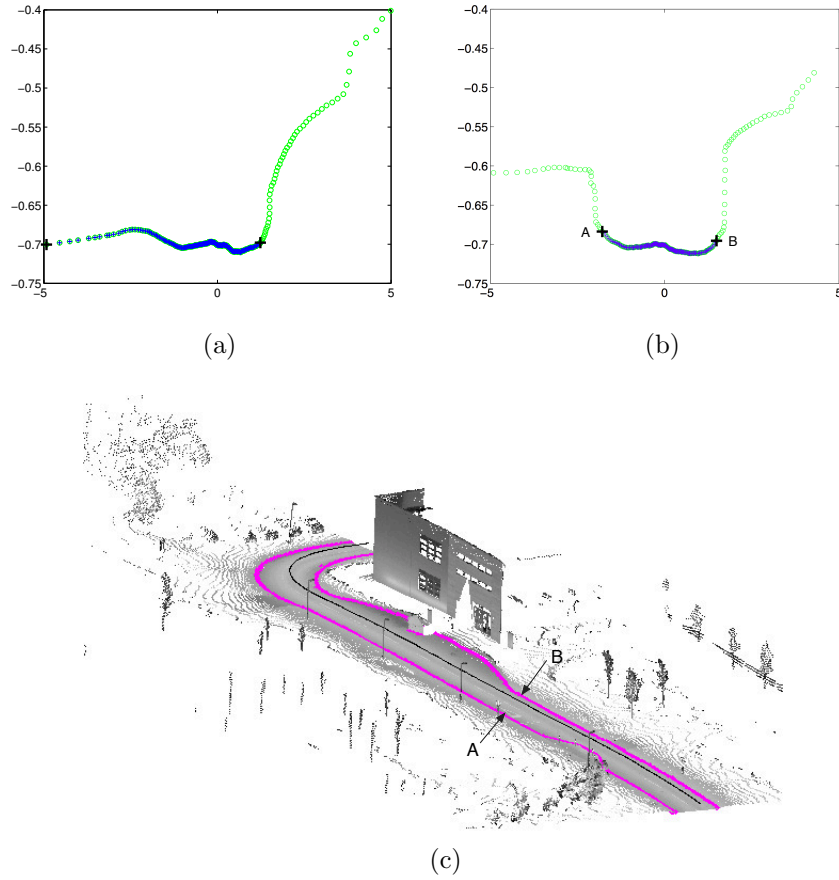


Figure 4.23: Top: Final road boundary point estimation outputs for a single scanline where (a) no curbs are present and (b) both curbs are present. (c) The final road boundary points detected over a sequence of laser scans, shown in magenta on the corresponding 3D point cloud. Point correspondences, A and B, are also shown.

Our road boundary detection system is therefore able to extract both curb and non-curb road boundary points within the LIDAR sensor’s field of view. An example scanline boundary point detection result is shown in Figure 4.23, along with a pointcloud showing boundary points detected over a sequence of scanlines.

The five parameters used in our road boundary point detection algorithm, summarised in Figure 4.12, are listed in Table 4.1 - notice how, unlike in much of the previous work, our algorithm does not rely on any thresholds for the final curb point extraction, given the feature functions. The extracted sequence of road boundary points form the input to our BaryArc curve fitting road boundary estimation model explained in the next part of this Chapter.

Parameter	Meaning of Parameter
th	Greedy Clustering (GC) algorithm threshold for groundstrike estimation
$nwind$	Range Filter window size
$noverlap$	Range Filter window overlap size
k	Tangent Normals neighbourhood size for computing local orientations
$nbootstraps$	Change Point Detection number of bootstrapping rounds

Table 4.1: Parameters of road boundary point detection algorithm.

4.5 Parametric Models for Representing Road Boundaries

In the previous section we described our algorithms for road boundary point detection summarised by Figure 4.12. Given these detected road boundary points, we want to determine a continuous representation for the left and right boundaries respectively. This curve fitting process allows us to represent the shape of each road boundary and to store its associated parameters in our Situational Awareness data structure described in Chapter 2. Formally, curve fitting is the process of estimating the curve that best fits a series of data points subject to specified constraints. In general curves can be represented using either implicit or parametric forms.

4.5.1 Implicit versus Parametric Form

Implicit curve functions take the form $f(x, y) = 0$ and describe the relationship between the x and y points located on the curve. Contrastingly, parametric curve functions represent each coordinate point separately, as an explicit function of an independent parameter within a specified interval. Let us consider the first quadrant unit circle as a simple example. Its implicit form is given by $f(x, y) = x^2 + y^2 - 1 = 0$ and two possible corresponding parametric forms are:

$$\begin{aligned} x(u) &= \cos(u) \\ y(u) &= \sin(u) \quad 0 \leq u \leq \frac{\pi}{2} \end{aligned}$$

or alternatively, substituting $t = \tan(\frac{u}{2})$ gives

$$x(t) = \frac{1 - t^2}{1 + t^2} \quad y(t) = \frac{2t}{1 + t^2} \quad 0 \leq t \leq 1 \quad (4.1)$$

Thus, as the parameters t or u vary, the curve is traced out and $x(u), y(u)$ and $x(t), y(t)$ are the coordinate functions. Notice how the parametric representation is not necessarily unique. The parametric form can be extended to specify curves in 3D space by adding a z coordinate. The implicit form however,

only specifies curves in the xy , yz or xz plane [112]. Bounded curves are easier to represent in parametric form since the parameter intervals dictate the curve bounds. Unbounded geometry such as a straight line to infinity is more simply represented by $ax + by + c = 0$ in implicit form, but the corresponding parametric representation is more difficult requiring infinity parameter intervals $(-\infty, \infty)$. Furthermore, computing a point on a curve is more difficult in implicit form whilst determining whether a point lies on a curve is more difficult in parametric form [112].

4.5.2 Interpolation versus Approximation

In general, the two main approaches to curve fitting are interpolation and approximation. In piece-wise or global interpolation, a function (of known degree) passes through the data points in a given order. This approach is simple but very sensitive to outlier points and noise, making it more suitable for random-noise free systems [100]. Curve approximation on the other hand involves estimating the best fit curve, given a model, that passes through some of the points whilst minimising some criteria (error). Least squares, natural smoothing and smoothing criteria are most commonly used [100]. In our work, we adopt an implicit curve representation using approximation curve fitting to derive the function parameters.

4.6 BaryArc Model for Representing Road Boundaries

Previously, in Section 4.2 we discussed the geometric priors, relevant to road boundary estimation, found in the Highway Engineering literature and the Design Manual for Roads and Bridges (DMRB). Recall that the road horizontal alignment can be divided into three parts: straight segments, curved segments and connecting curves. Typically, the straight segments are modelled as lines, the curved segments as arcs (with a radius of curvature 6m/10m for rural/urban roads) and clothoids can be used as connecting curves to ensure a smooth transition. In practice however, straight line segments are often directly joined to arcs because drivers tend to guide their vehicles gradually into circular curves thus compensating for transitions between arcs and tangents (See Section 4.2.3.2). The road boundary model we seek therefore need only represent arcs and tangent lines in order to model the horizontal road alignment (and hence road boundaries) according to Highway Traffic Engineering principles. In this section we formulate a BaryArc curve fitting model to solve this problem.

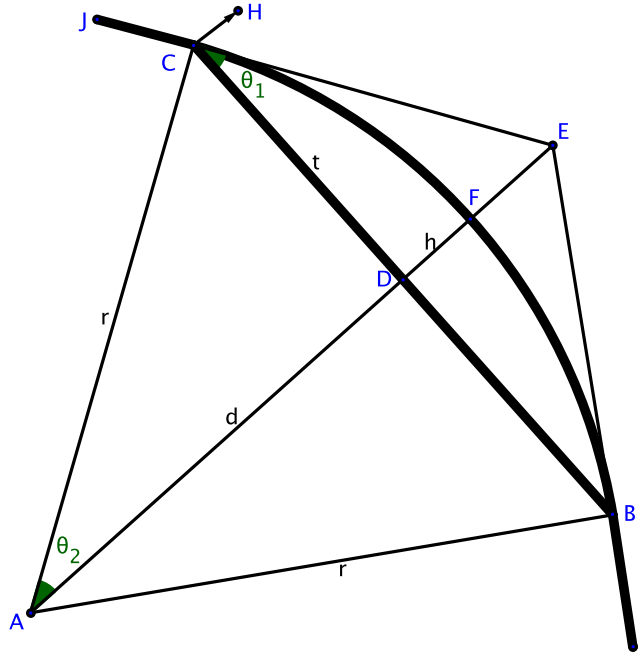


Figure 4.24: Representation of arc for road curvature ($\theta = \theta_1 = \theta_2$).

4.6.1 Geometry

The geometry for this problem is shown in Figure 4.24, which is derived from the horizontal alignment simple curve geometry of Figure 4.7. We represent a circular arc CFB by three points: C, E and B. A is the centre of the circle (of radius r) that forms the arc and θ is the (green shaded) half-angle at the centre of the circle. The midpoint of chord BC is D and $|CD| = |DB| = t$, $|AD| = d$ and $|DF| = h$. Finally, H represents the direction vector perpendicular to chord BC. BI and JC are straight line segments joined to either side of the arc. The coordinates of point E can be derived relative to point D. First, let's derive some important geometric relationships in the next subsection.

4.6.1.1 Bulge

A circular arc is characterised by its displacement from its chord. We call this the bulge which is defined as: $b = \frac{h}{t}$. In general $h \ll t$ for roads since road curvature is gradual and must fulfil line-of-sight requirements.

Using the geometric relations shown in Figure 4.24, we derive the following relationships:

$$\begin{aligned}
 b &= \tan\left(\frac{\theta}{2}\right) \\
 d &= t \frac{1-b^2}{2b} \\
 \tan(\theta) &= \frac{t}{d} = \frac{2b}{1-b^2} \\
 H &= \frac{(B-C)^{-\perp}}{\|B-C\|}
 \end{aligned} \tag{4.2}$$

Using the above expressions, the coordinates of point E, in terms of the bulge, are given by

$$E = D + H\left(\frac{2b}{1-b^2}\right)t \tag{4.3}$$

4.6.1.2 Barycentric Coordinates

Barycentric coordinates are a form of homogeneous coordinates in which a point is represented relative to the vertices of a triangle. The advantage of such a coordinate system is that it is defined on the curve itself. Consider the simple example of a straight line between endpoints $\mathbf{q}_1$ and $\mathbf{q}_2$. Any point $\mathbf{q}$ on the line is given by $\mathbf{q} = (1-t)\mathbf{q}_1 + t\mathbf{q}_2$ such that for $t \in [0, 1]$ the line lies between the points. For t values outside this interval, the line extends beyond the endpoints. The coefficients, $(1-t)$ and t , of the line equation sum to one and are in fact the barycentric coordinates for this line. Thus $\mathbf{q} = (1-t, t, 0)$ represents the barycentric coordinates of any point on the line.

Triangle $\triangle CEB$ of Figure 4.24 forms a barycentric system. In general, a point $\mathbf{p}$ can be represented by barycentric coordinates, in this triangular coordinate frame, as:

$$\mathbf{p} = \lambda_0 \mathbf{C} + \lambda_1 \mathbf{E} + \lambda_2 \mathbf{B} \quad \text{where } \lambda_0 + \lambda_1 + \lambda_2 = 1 \tag{4.4}$$

Here, the barycentric coordinates of the point are $\mathbf{p} = (\lambda_0, \lambda_1, \lambda_2)$ and $\mathbf{p} = \lambda_0 \mathbf{C} + \lambda_1 \mathbf{E} + (1 - \lambda_0 - \lambda_1) \mathbf{B}$. In other words, given $\mathbf{p}$, we find the weights λ_i such that $\mathbf{p} = \sum_i \lambda_i \mathbf{p}_i$ and the smoothness $\frac{\partial \lambda_i}{\partial p}$ is constant. The sum of barycentric coordinates equal to one implies translational invariance. It is well known that these coordinates can be interpreted as ratios of triangle areas. Furthermore to find the barycentric coordinates of a point involves solving a linear system of equations:

$$\begin{bmatrix} \mathbf{C} & \mathbf{E} & \mathbf{B} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}$$

which can be solved using Cramer's rule as follows:

Let

$$\mathbf{A} = \begin{bmatrix} \mathbf{C} & \mathbf{E} & \mathbf{B} \\ 1 & 1 & 1 \end{bmatrix}$$

then

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{p} & \mathbf{E} & \mathbf{B} \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} \mathbf{C} & \mathbf{p} & \mathbf{B} \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{A}_3 = \begin{bmatrix} \mathbf{C} & \mathbf{E} & \mathbf{p} \\ 1 & 1 & 1 \end{bmatrix}$$

and the barycentric coordinates can be found to be:

$$\lambda_0 = \frac{\det(\mathbf{A}_1)}{\det(\mathbf{A})} \quad \lambda_1 = \frac{\det(\mathbf{A}_2)}{\det(\mathbf{A})} \quad \lambda_2 = \frac{\det(\mathbf{A}_3)}{\det(\mathbf{A})} \quad (4.5)$$

Circular arcs therefore naturally lend themselves to a barycentric coordinate representation.

4.6.2 BaryArc Derivation

Higher order polynomials are able to represent a wide range of functions except conic sections (such as circular arcs) which can only be represented exactly using rational functions. In general, a rational parametric equation is given by quotients of the form:

$$x(t) = \frac{V(t)}{R(t)} \quad y(t) = \frac{W(t)}{R(t)}$$

such that the denominators, $R(t)$ are identical. For a rational Bezier curve, Bernstein polynomial functions [7], $B_{i,n}(t)$, are used for representing $V(t)$, $W(t)$, and $R(t)$. An n^{th} -degree rational Bezier function takes the form

$$\mathbf{f}(t) = \frac{\sum_{i=0}^n B_{i,n}(t) \omega_i \mathbf{P}_i}{\sum_{i=0}^n B_{i,n}(t) \omega_i} \quad 0 \leq t \leq 1$$

where the control points are $\mathbf{P}_i = [x_i, y_i]^T$ and associated positive scalar weights are ω_i . In this work, we consider only the quadratic rational Bezier spline of degree $n=2$. The weights are in fact not necessarily unique for quadratic rational Beziers, but are related by $\frac{\omega_0 \omega_2}{(\omega_1)^2} = \frac{\omega'_0 \omega'_2}{(\omega'_1)^2}$ [109]. To make the weights unique up to a multiplication factor the standardised form assigns $\omega_0 = \omega_2 = 1$ and the weight ω_1 is variable [112]. We adopt this standardised form in our work. Thus, when $\omega_1 = 1$ we get the polynomial Bezier curve and

for $w_1 \rightarrow 0$ a straight line segment between the first and last point.

Derivation We are now ready to derive our BaryArc model for representing road boundaries as arcs and tangents in barycentric coordinates. For a rational quadratic Bezier curve, the equation is given by:

$$c(t) = \frac{w_0(1-t)^2\mathbf{C} + 2w_1t(1-t)\mathbf{E} + w_2t^2\mathbf{B}}{w_0(1-t)^2 + 2w_1t(1-t) + w_2t^2} \quad (4.6)$$

and the Barycentric coordinates of a point on the arc is equally given by:

$$c(\lambda) = \lambda_0\mathbf{C} + \lambda_1\mathbf{E} + \lambda_2\mathbf{B} \quad (4.7)$$

Equating the coefficients of equations 4.6 and 4.7 we can derive the BaryArc equation. Let $\alpha = w_0(1-t)^2 + 2w_1t(1-t) + w_2t^2$, then

$$\frac{w_0(1-t)^2}{\alpha} = \lambda_0 \quad \text{and} \quad \frac{2w_1t(1-t)}{\alpha} = \lambda_1 \quad \text{and} \quad \frac{w_2t^2}{\alpha} = \lambda_2 \quad (4.8)$$

by substituting $\alpha = \frac{w_2t^2}{\lambda_2}$ we get

$$\frac{w_0\lambda_2(1-t)^2}{w_2t^2} = \lambda_0 \quad \text{and} \quad (1-t) = \frac{\lambda_1w_2t^2}{2w_1t\lambda_2} \quad (4.9)$$

combining these equations and simplifying yields

$$\lambda_1^2w_0w_2 = 4w_1^2\lambda_0\lambda_2 \quad (4.10)$$

The implicit BaryArc equation is therefore given by:

$$f(\lambda, \mathbf{w}) = \lambda_1^2w_0w_2 - 4w_1^2\lambda_0\lambda_2 = 0 \quad (4.11)$$

For the standardised form we know that $w_0 = w_2 = 1$, so in this model we need only estimate the value of the weight of point E (i.e. w_1). When $w_1 \rightarrow 0$, the arc degenerates to a line segment between C and B. Therefore this representation is able to model both arcs and tangents precisely - a property we desire for road boundary estimation. The points C, B, and E are the BaryArc anchor points.

Using the standardised form, Equation 4.11 can be simplified to:

$$f(\lambda, \mathbf{w}) = \lambda_1^2 - 4w_1^2\lambda_0\lambda_2 = 0 \quad (4.12)$$

and using $\lambda_0 + \lambda_1 + \lambda_2 = 1$ from the definition of barycentric coordinates (see Equation 4.4) it can be rewritten as:

$$f(\lambda, w) = \lambda_1^2 - 4w_1^2\lambda_0(1 - \lambda_0 - \lambda_1) = 0 \quad (4.13)$$

which is the implicit equation of the BaryArc. Note that for points inside the triangle, $\lambda_i \geq 0$. Vertices occur when two barycentric coordinates are zero. The form of the arc equation derived above has just one parameter, w_1^2 . Points on the curve itself have barycentric coordinates such that $f(\lambda, w) = 0$.

4.6.3 Differentiability

The BaryArc implicit Equation 4.13 is differentiable and therefore continuous. The partial derivatives are given by:

$$\begin{aligned} \frac{\partial f}{\partial \lambda_0} &= -4w_1^2 + 8w_1^2\lambda_0 + 4w_1^2\lambda_1 \\ \frac{\partial f}{\partial \lambda_1} &= 2\lambda_1 + 4w_1^2\lambda_0 \end{aligned} \quad (4.14)$$

and the second order partials are

$$\begin{aligned} \frac{\partial^2 f}{\partial \lambda_0^2} &= 8w_1^2 \\ \frac{\partial^2 f}{\partial \lambda_1^2} &= 2 \\ \frac{\partial f}{\partial \lambda_0 \lambda_1} &= \frac{\partial f}{\partial \lambda_1 \lambda_0} = 4w_1^2 \end{aligned} \quad (4.15)$$

Thus the curvature of the BaryArc, which is proportional to the second derivative, depends only on the value of the parameter w_1^2 .

4.6.4 Estimating the BaryArc weight w_1^2

Given the implicit curve Equation 4.13 and a set of N noisy data points $\{p_i\}_{i=1}^N$ in barycentric coordinates, we want to approximate the parameter w_1^2 for the BaryArc that best fits these data. We use a least squares fit of the data, minimising the following constrained optimisation function subject to non-negative x :

$$\min_x \| \mathbf{Ax} - \mathbf{d} \|^2_2 \text{ where } x \geq 0$$

where

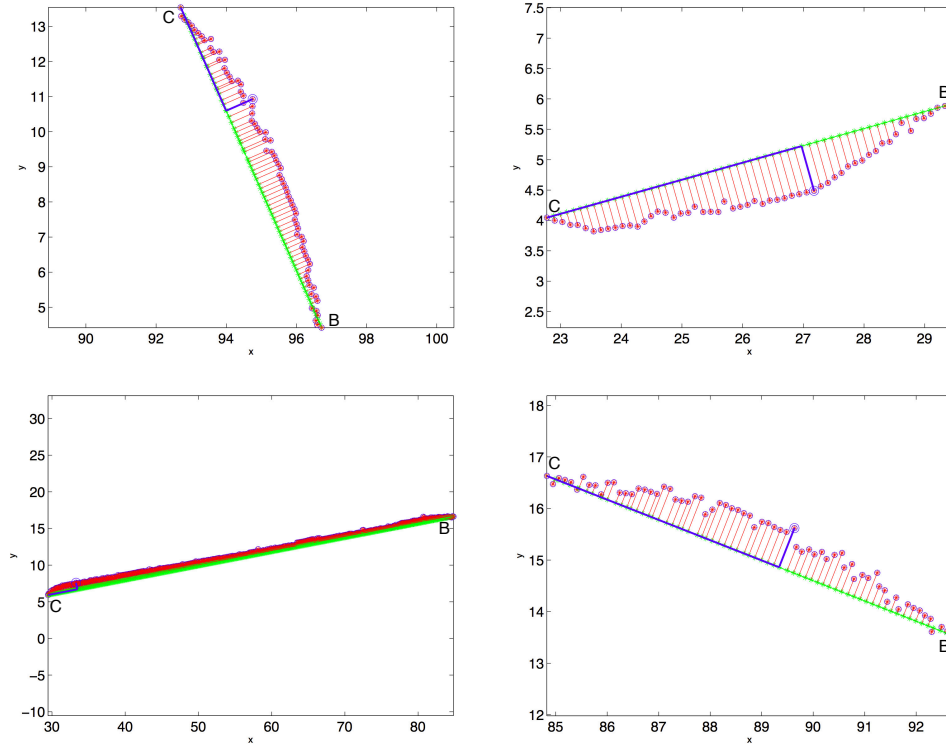


Figure 4.25: The bulge estimation procedure: Points C and B are the known endpoints of the green line. The road boundary detected points are projected onto the green line and the lengths h (perpendicular) and t (parallel) of Figure 4.24 are shown as dark blue lines. The bulge parameter is given by $b = \frac{h}{t}$.

$$\mathbf{A} = [4\lambda_0\lambda_2]; \mathbf{x} = w_1^2; \mathbf{d} = \lambda_1^2$$

This approximation, being least squares, suffers from sensitivity to noise however our boundary point extraction framework uses prior information about the width of the road and RANSAC [37] to minimise outliers.

4.6.5 Estimating the Road Boundary Transition Points

Our BaryArc formulation is parameterised by the bulge b , described in Section 4.6.1.1. In practice, we take this bulge to be the ratio of the distance from a point on the arc furthest from the line joining the endpoints; and the distance from the corresponding projected point to the initial arc point. Figure 4.25 illustrates some results of our bulge parameter estimation procedure.

Recall from Figure 4.24 that the initial point C and final point B are the known endpoints of the given data points. Point E is however unknown and must be estimated from the data using Equation 4.3 which

is a function of the bulge value.

The BaryArc model estimates the road boundaries as either tangent or curved however, given a sequence of detected road boundary data points, it is not known a priori which segments are curved and which are straight tangents. To estimate these transition points for piece-wise BaryArc curve fitting we use a recursive algorithm based on the Mean Squared Error (MSE) of the BaryArc fit which is summarised by Algorithm 4.4.

Algorithm 4.4 Recursive Estimation of Curvature Transition Points for BaryArc Curve Fitting

1. Given the road boundary data points $[\mathbf{x}, \mathbf{y}]$, let $patch = 50$ be the known patch size for the number of points to add to the accumulator at each iteration.
2. Initialise or update the accumulators with $patch$ number of points and associated errors
3. Fit the BaryArc to $points$ and find the associated MSE of the fit $mse(points)$. The MSE is an estimator of the variance of random disturbances, which is assumed constant. Given the least squares residuals r , and $p = 1$ unknown parameters from the least square estimation described in previous section, the MSE is given by

$$MSE = r' * r / (n - p) \quad (4.16)$$

where n is the number of observations.

4. Repeat (1) to (3) until the error (MSE) no longer decreases between iterations. The index of the current iteration is the transition point index for the BaryArc fit.
 5. Set transition point indice as new initial starting point and repeat steps (1) to (4) until termination (no points remain).
-

Given the vector of transition points calculated using Algorithm 4.4, the BaryArc curve is fit to the data points using a piece-wise approach that is first-order continuous. Specifically, we first use RANSAC to find inliers, and then fit a BaryArc road boundary model to these inlier points. This makes our method more robust to outliers resulting from the presence of traffic cones and vehicles on the road, or false road boundary points from our feature detection.

RANdom SAmple Consensus (RANSAC) is an iterative algorithm proposed by Fischler and Bolles [37] to estimate the parameters of a model from experimental data containing outliers. In our case, the model is the BaryArc and the data are the output of our road boundary point detection. RANSAC begins by randomly selecting a subset, s of points from the data. A model M is then fit to this subset and the set of points s' that is within an error tolerance tol of this model is termed the inliers or consensus set. If the number of inliers s' is above some threshold t , these inliers are used to estimate a new model M' . If however, s' is less than t , the algorithm randomly selects a new subset of points and repeats the above

procedure. The RANSAC algorithm is repeated until the best (largest) consensus set is found, or up to a predefined r times terminating with a model for the largest found consensus set or in failure. The three parameters of RANSAC are therefore tol , t , and r .

Most importantly, we formulate road boundaries as piece-wise continuous BaryArc curves based on geometric domain prior information from Highway Traffic Engineering literature.

4.6.6 Plotting a BaryArc curve

Now that the parameter w_1^2 is known, the BaryArc curve can be plotted. In general, the implicit equation is good for determining intersection points and whether a point lies on the curve. For plotting the curve, the parametric representation is preferred. The process of converting an implicit equation to its equivalent parametric form is known as parameterisation and is a classic topic in algebraic geometry. Parameterisation is more difficult than the inverse problem of implicitization which generally involves the elimination of the single parameter variable from the parametric form [2].

Recall that a general second degree conic implicit equation is given by $C(x, y) = ax^2 + by^2 + cxy + dx + ey + f = 0$ and the corresponding rational parametric equations are $x(t) = \frac{u(t)}{w(t)}$ and $y(t) = \frac{v(t)}{w(t)}$ where $u(t)$, $v(t)$, and $w(t)$ are polynomials of at most degree two. This is the parametric form we wish to derive for our implicit BaryArc Equation 4.13 via a parameterisation procedure.

We use the algorithm of parameterisation by lines [11] for the special case of conic parametric curves to achieve this. In particular, for the parameterisation of any conic, given an arbitrary point (simple point) $P = (p_1, p_2)$ on the curve, we can use a system of lines with equation

$$y = (x - p_1)t + p_2 \tag{4.17}$$

to parametrize the curve. Formally, the idea of parameterizing (rational) conics with lines was described by Abhyankar and Bajaj in [2] as fixing a point and taking a pencil of lines through that point which meet the conic at one additional point. Thus the parameterisation of conics is always rational with the slope of the line being the single parameter [2]. For our application, we are specifically interested in the parameterisation leading to a Rational Parametric Equation (RPE) since our BaryArc represents conic sections (arcs). To derive one such parametric representation, we consider the point $P = (0, 0)$ on the BaryArc which when substituted into Equation 4.17 yields the line $y = tx$ passing through the origin. This line intersects the graph at a specific t value and by moving the line as the parameter varies, the complete curve can be sketched out. Using this method, we can therefore derive the parametric BaryArc to be:

$$f(\lambda, w) = \lambda_1^2 - 4w_1^2\lambda_0 + 4w_1^2\lambda_0^2 + 4w_1^2\lambda_1\lambda_0 = 0 \quad (4.18)$$

let $\lambda_0 = x$; $\lambda_1 = y$; $w_1^2 = w$

then we have

$$f(x, y, w) = y^2 - 4wx + 4wx^2 + 4wxy = 0 \quad (4.19)$$

Now substitute $y = tx$ and we get,

$$(tx)^2 - 4wx + 4wx^2 + 4wx(tx) = 0 \quad (4.20)$$

$$x(t^2 + 4wt + 4w) = 4w \quad (4.21)$$

The BaryArc parametric equations are therefore given by:

$$x(t) = \frac{4w}{t^2 + 4wt + 4w} \quad y(t) = \frac{4wt}{t^2 + 4wt + 4w} \quad (4.22)$$

Or in barycentric coordinates

$$\lambda_0(t) = \frac{4w_1^2}{t^2 + 4w_1^2t + 4w_1^2} \quad \lambda_1(t) = \frac{4w_1^2t}{t^2 + 4w_1^2t + 4w_1^2} \quad \lambda_2(t) = 1 - \lambda_0(t) - \lambda_1(t) \quad (4.23)$$

It is worthwhile noting that in general parametric representations are not unique. The choice of the simple point influences the complexity of the resulting parameterisation. Finally, to plot the BaryArc road boundaries, we convert the barycentric coordinates to a cartesian representation.

4.7 Performance Evaluation

4.7.1 Data Sets

We collected two large datasets using our autonomous vehicle, the Bowler Wildcat, which is equipped with a mono camera, Bumblebee camera, 3D laser, GPS/INS and SICK LIDAR sensors for environment perception (see Figure 4.26).

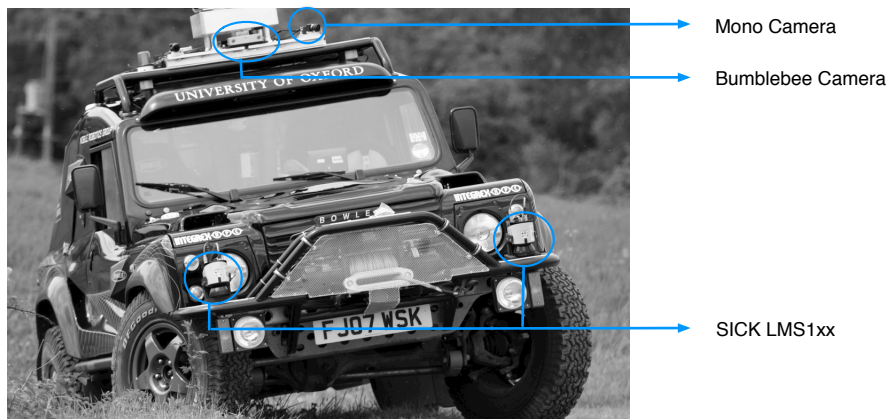


Figure 4.26: The Wildcat autonomous car is equipped with front and rear (not shown) vertically downward facing SICK LMS1xx range sensors and mono and stereo (bumblebee) cameras. The differential OxTS GPS/INS unit and rotating 3D laser (not Velodyne) are mounted on the vehicle rooftop.

The SICK short range laser emits light from the sensor to the object and estimates the range of the object (relative to the sensor) using time-of-flight [64]. The SICK LIDAR sensors on the Wildcat vehicle have a 270° field of view (see Figure 4.27) and a 25Hz/50Hz scanning frequency with a 0.25° angular resolution and are mounted pointing vertically downwards in a push broom configuration. Under good conditions a typical operating range of 0.5m - 20m and associated systematic and statistical errors of 30mm and 12mm have been observed respectively. Thus the operation error is larger for smaller operating ranges. The Point Grey Bumblebee2 BB2-7400627 colour stereo camera was used for the roadwork scene recognition (Chapter 5) work described in this thesis. This firewire colour camera records images at a 640x480 or 1024x768 resolution at a 20 FPS recording rate. The two datasets we collected spanned 32km and are summarised in Table 4.2.

4.7.2 Results

This section details our road boundary estimation results, evaluated on the datasets described in the previous section.

Obtaining ground truth for road boundary labels in 3D laser point clouds is challenging because manual labelling is time consuming, labour intensive and expensive. In scene interpretation literature, 3D point cloud benchmarking datasets containing manually labeled semantics such as pedestrians, vehicles, trees, and buildings have been created before. The Oakland 3-D Point Cloud Dataset [25] and the Urban Scene 3D Point Cloud Dataset (Velodyne) [84] are two such datasets, useful for benchmarking object segmentation algorithms. Labelling road boundaries in 3D point clouds poses new challenges because

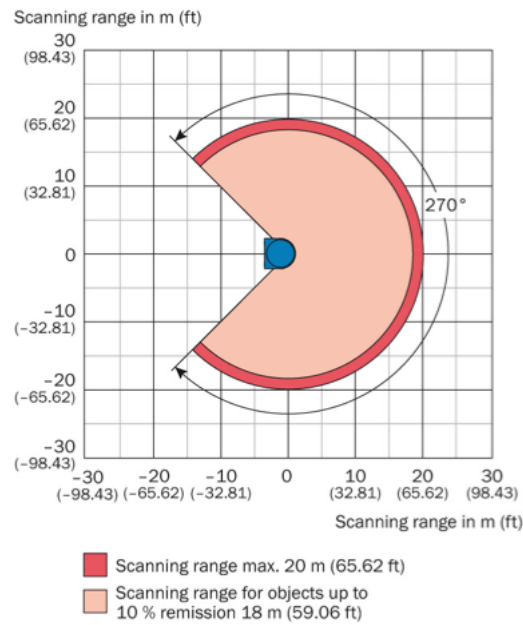


Figure 4.27: The operating range diagram of the SICK LMS1xx sensor on the Wildcat autonomous car. Figure taken from [64].



	Type	Distance	Features
Dataset 1	Science Park + Dual Carriageway	15km	Roadworks
Dataset 2	Urban and Residential	17km	None

Table 4.2: Vehicle GPS trajectory for Dataset 1 and 2 shown on Google Maps. The trajectories total approximately 32km and are described in the Table.

individual data points (rather than groups or regions of points) must be labeled. In the absence of a perfectly accurate GPS under all driving conditions, obtaining groundtruth data from overhead images is a difficult problem requiring accurate registration between the vehicle laser sensor data and overhead image. Furthermore, the boundaries in the overhead image must first be manually labeled to provide ground truth. One possible way to tackle this problem in future is to run a road boundary detection algorithm (such as the one proposed in this thesis) on a benchmarking dataset, and later have a human operator verify, and if necessary alter, the automatically labeled boundaries.

In our work, we compute quantitative metrics to concretely assess the quality and performance of our system.

4.7.2.1 Centre Line Continuity

We begin by evaluating the continuity of the estimated road boundaries. The estimated road centre line (i.e. the midpoint between the detected road boundaries), as a function of distance travelled, gives a compact measure of continuity. In reality, the center line of the road should be continuous and smooth because the road itself varies smoothly. Features such as narrowing lanes, bus lanes, emergency stops, road island lanes, junctions and intersections affect the width of the road and center line however, these variations are gradual.

In our experiments, we compute the center line of the road at 2m intervals along the road. The results for Dataset 1 are shown in Figure 4.28, where the path traced out by the center line resembles the one driven by the vehicle. On the right, the x and y components of the center line are plotted as a function of the distance travelled. Notice how the curves vary gradually showing that our estimated center line does not suddenly change but varies smoothly. Also, the repetitions in the curves are as a result of the vehicle making multiple traversals of the same road area between two roundabouts. Thus the repetitions should look near identical because the same road is being traversed and the same center line should be estimated each time. This is indeed the case, as shown in Figure 4.28.

Similarly, Figure 4.29 summarises the results for Dataset 2 where the smoothness of the curves show that the center line estimation varies gradually even along less structured roads (not motorways like in Dataset 1). The repetitions on the curve also result from multiple traversals of parts of the road during data collection. Here, the minor errors are due mainly to the presence of parked vehicles along the road.

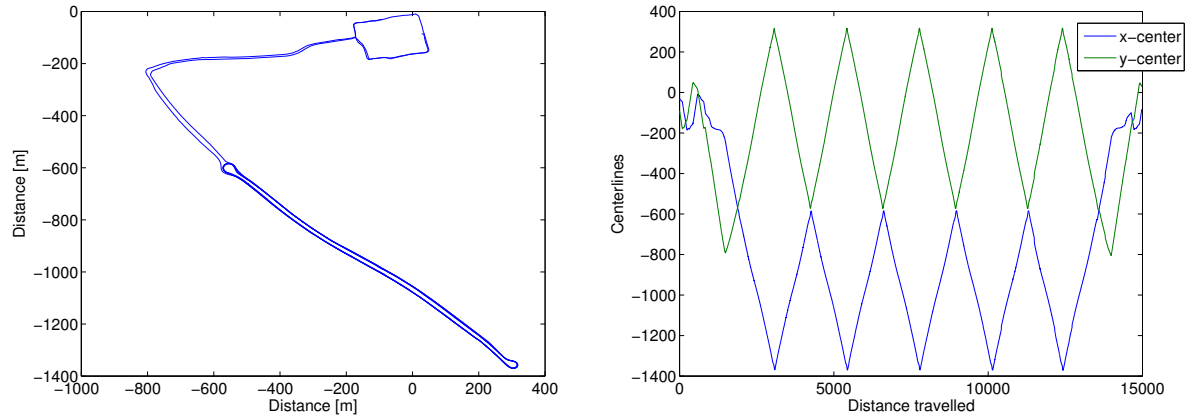


Figure 4.28: Dataset 1: Left: The road center line determined from the detected road boundaries. Right: The x and y position of the center line as a function of distance travelled. The repetitions result from when the vehicle drove over the same section of motorway (between two roundabouts) multiple times. The identical repetitions show that the center line is consistently estimated correctly each time.

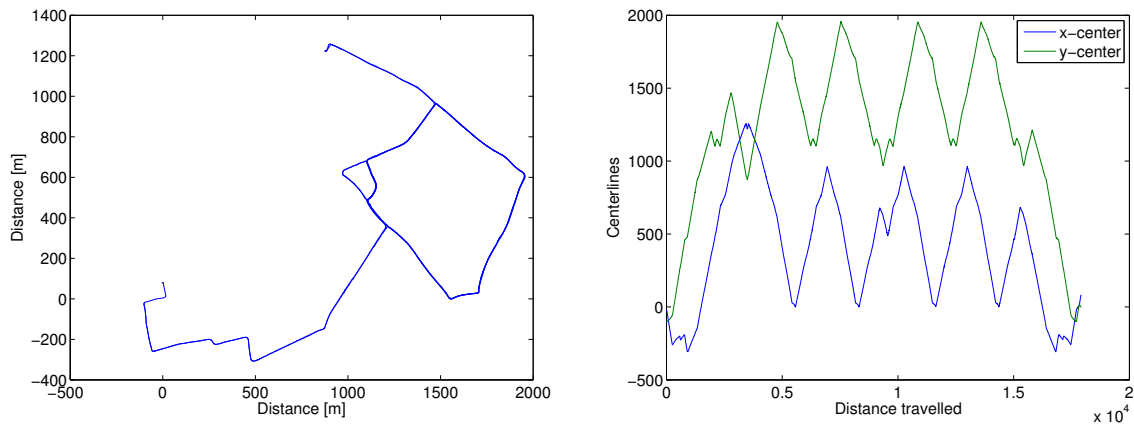


Figure 4.29: Dataset 2: Left: The road center line determined from the detected road boundaries. Right: The x and y position of the center line as a function of distance travelled. The repetitions arise from repeated traversals of sections of the road through rural, urban and residential areas.

4.7.2.2 Mean Squared Error for BaryArc

An important metric to consider is how well our BaryArc model fits the detected road boundary data points. In Section 4.6.4, we used a least squares fit to determine the parameters of the curve, thus the Mean Squared Error (MSE) is one possible good metric to use to assess the accuracy of our BaryArc fit. The MSE is an estimator of the variance of random disturbances, which is assumed constant. Given the least squares residuals r , and $p = 1$ unknown parameters, the MSE is given by

$$MSE = r' * r / (n - p) \quad (4.24)$$

where n is the number of observations. Recall that our BaryArc fit is piecewise (since the road is comprised of tangent and arc regions). Thus we compute the MSE for each piecewise segment of the curve for both left and right boundaries. The average MSE for the left and right boundary BaryArc fits is $1.7204e-04m^2$ and $0.0015m^2$ respectively for Dataset 1. Similarly, Dataset 2 has an average MSE of $0.0056m^2$ and $0.0025m^2$ for the left and right boundaries respectively. These values confirm that our BaryArc model is a good approximation for continuous road boundary estimation. The empirical cumulative distribution functions (CDFs) of the MSE for parts of Datasets 1 and 2 are also plotted in Figure 4.30. These CDFs capture the probability that the random variable $X = MSE$ is found to have a value less than or equal to x .

4.7.2.3 Road Curvature

In this section we derive the road curvature given the road boundaries. Our BaryArc Model discussed in Section 4.6 defines a curve in barycentric coordinates given three anchor points (denoted by C, E, B in Figure 4.24). The curve lies within the triangle formed by these points and the parameter w_1^2 controls the shape of the curve. Tangent (straight) sections of the road have small w_1^2 and curved sections larger w_1^2 . Furthermore, the second order partial derivatives of the BaryArc in Equation 4.15, which are proportional to the local curvature, depend only on the parameter w_1^2 . In practice, roads cannot just be straight because they must accommodate the topology of the surrounding area [44]. For situational awareness, we are specifically interested in allowing a vehicle to infer the ease of drivability of a road based on whether it's a curved, straight or circular (like a roundabout). In general, more care should be taken in terms of speed, maintaining a good line of sight, making U-turn manoeuvres and overtaking, when driving along curved road segments.

To partition the road boundaries into curved and straight segments, we considered the sum of the areas

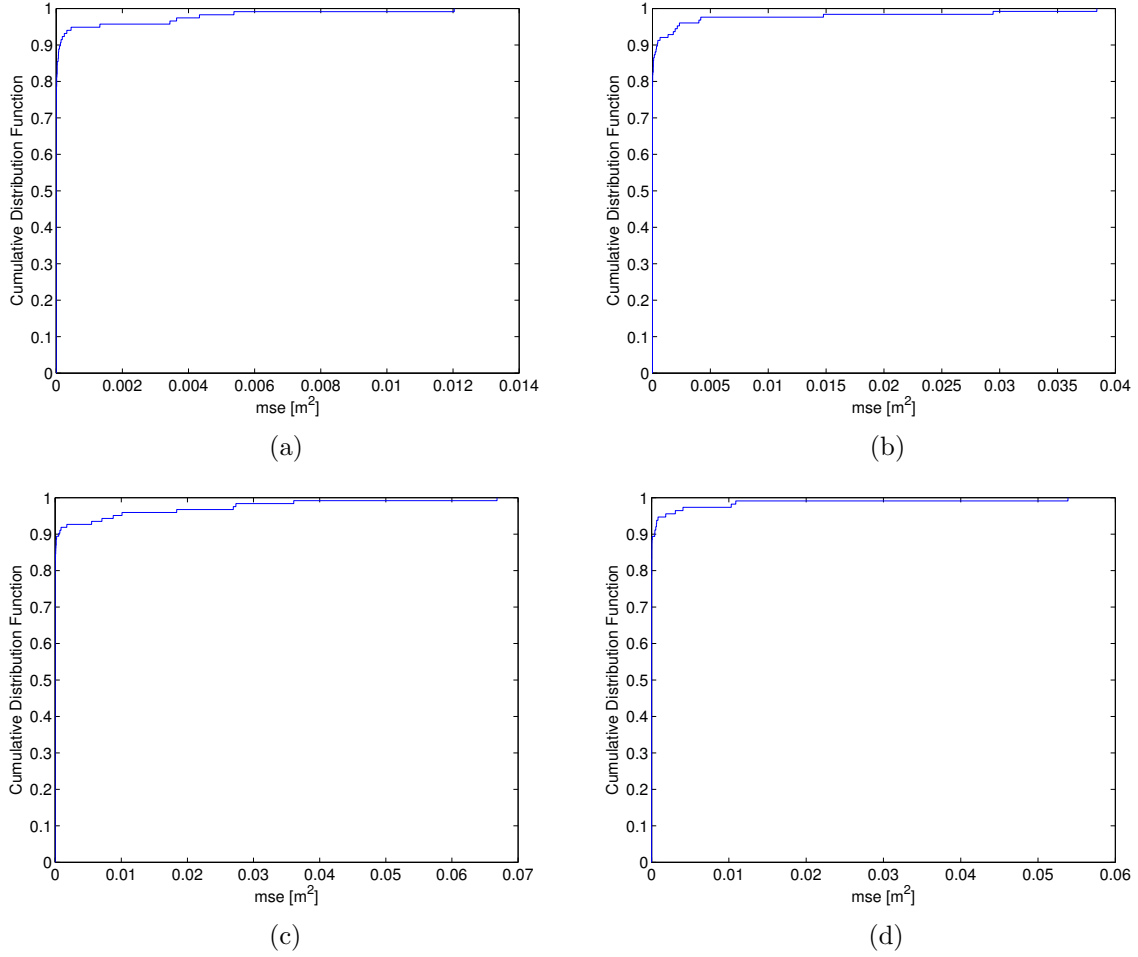


Figure 4.30: The cumulative distribution function of the Mean Squared Error for the (a) left road boundary in Dataset 1 (b) right road boundary in Dataset 1 (c) left road boundary in Dataset 2, and (d) right road boundary in Dataset 2

of the polygons formed by the road boundaries. Thus tangent regions of roads have much smaller area sums than arc regions. Figure 4.31 illustrates our detected road boundaries for Dataset 1, on Google Maps, colour coded based on the calculated curvature (see colour bar). Notice how we are able to automatically detect interesting road semantics such as roundabouts (traffic circles) based on road curvature. Similar results for Dataset 2 are shown in Figure 4.32 where the estimated boundaries and curvature is along rural and residential roads.

4.7.2.4 Intersections

Intersections influence the accuracy of detected road boundaries because defined road edges are absent at junctions and intersections. Thus an estimated boundary, using the laser data, at an intersection may indeed be traversable. We use OpenStreetMap, an online community project that creates and distributes free geographic data (maps and their associated annotations) [107] to extract both major and minor intersections along the vehicle’s trajectory. The fundamental element in OpenStreetMap is a node which is a single geospatial latitude longitude point. A node represents a single point feature and an ordered sequence of nodes defines a way [107]. Specifically, roads/streets/paths are defined by ways with semantic highway tags such as “highway=primary”, “highway=pedestrian”, and “highway=motorway”. The unique points where ways cross therefore define intersection locations. OpenStreetMap provides manually labeled data (by a set of volunteers) with true locations of ways in WGS84 latitude longitude coordinates (we extract intersections from the ways). Our vehicle GPS/IMU unit provides the vehicle’s global pose which is subject to drift (due to atmospheric effects) thus we interpolate the location of the intersection on the vehicle’s GPS trajectory. The intersections detected along the vehicle path are shown in the bottom right image of Figure 4.31, and Figure 4.33 for Dataset 1 and Dataset 2 respectively.

4.7.2.5 Road Boundary Estimation in Challenging Scenarios

In this section, we consider the performance of our road boundary estimation algorithm in difficult situations. Firstly, challenges arise when obstacles such as traffic cones, roadwork barriers or vehicles are present on the road. These objects occlude the laser scanline (road points beyond the object are missed), resulting in a partial sensing of the road surface and potential boundaries. The second challenging scenario occurs at intersections where no definite road boundary is observable. Thirdly, due to our vertically downward facing sensor configuration, the laser point density decreases dramatically with distance from the sensor origin resulting in sparse points deep into the intersection. In the case of motorway roadworks, found in

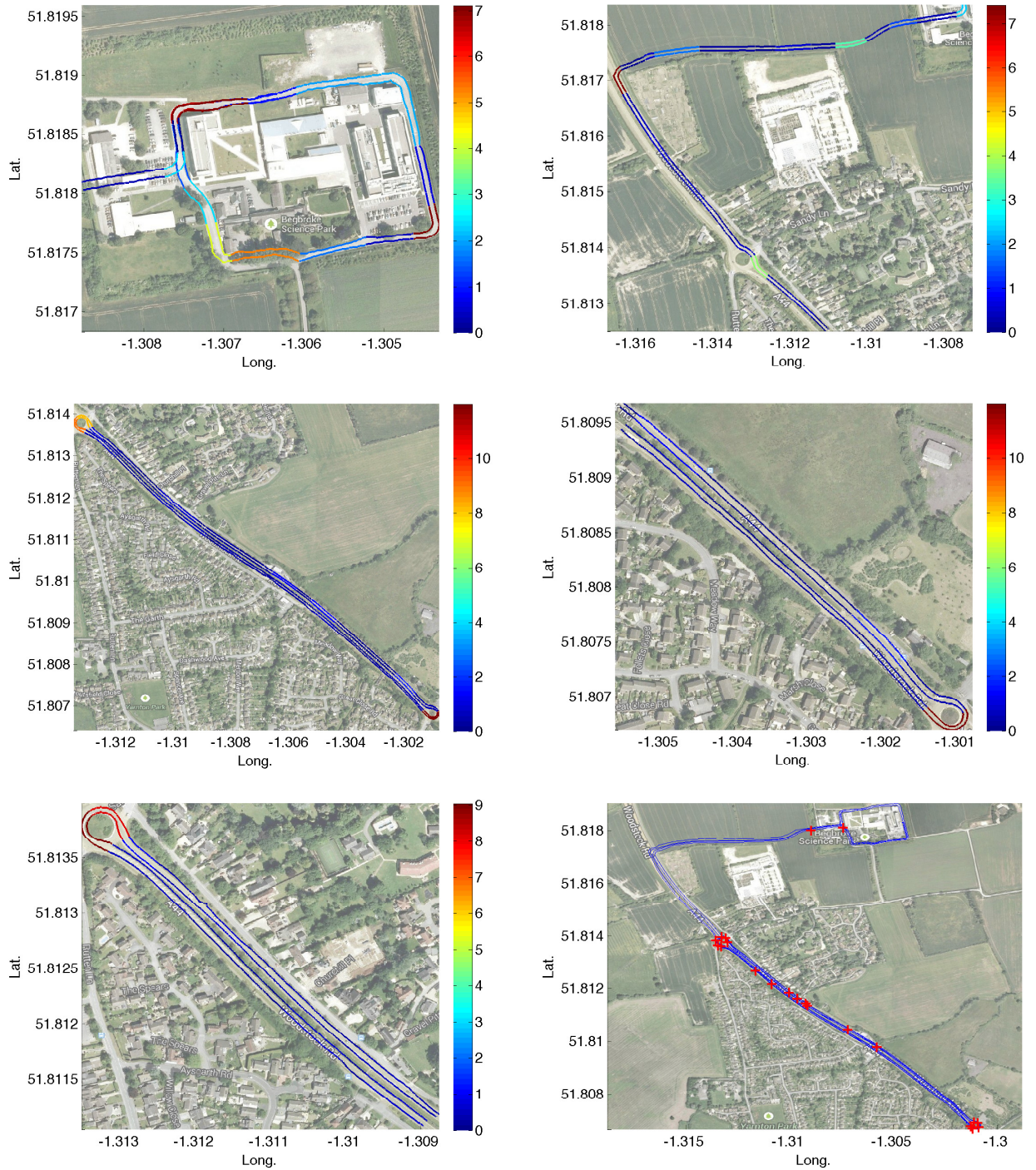


Figure 4.31: Dataset 1: Our estimated BaryArc road boundaries plotted on Google Maps, where the boundary colour represents the curvature values (see colour bar). Here, semantics such as roundabouts are automatically detected as regions of high curvature. The extracted OpenStreetMap intersections are also shown in the bottom right image (red crosses).

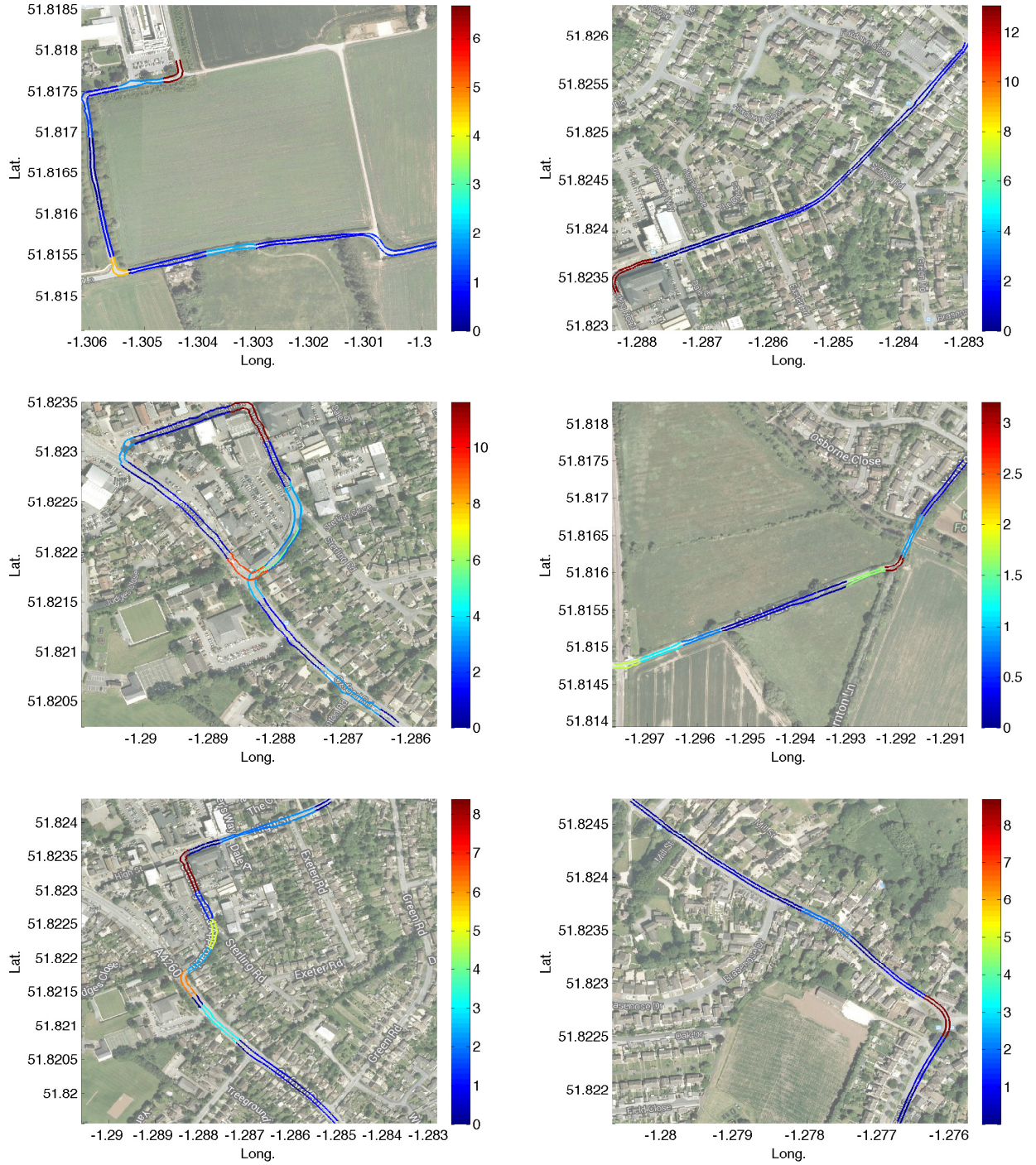


Figure 4.32: Dataset 2: Detected BaryArc road boundaries plotted on Google Maps, where the boundary colour represents the road curvature values.

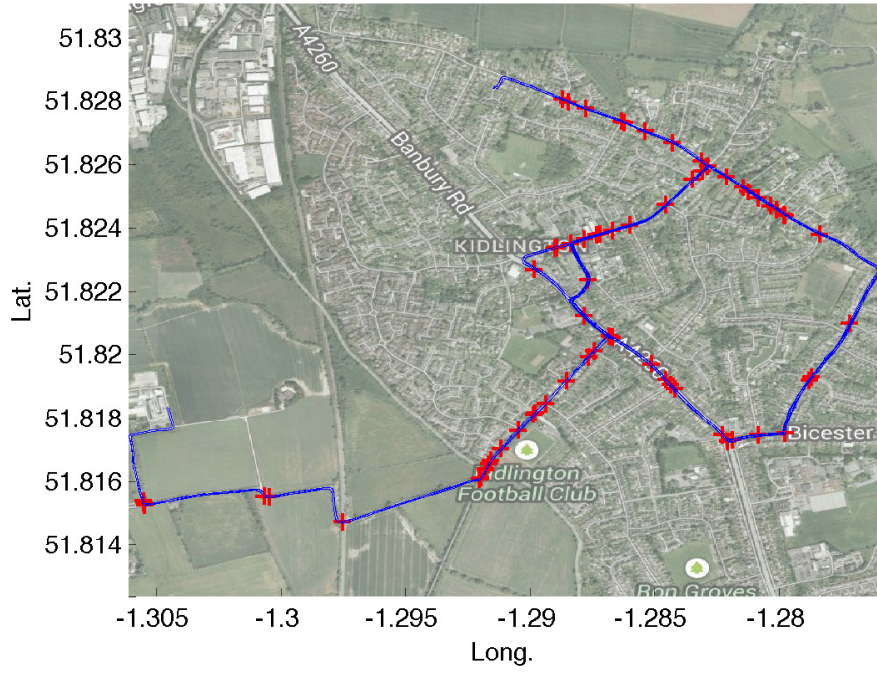


Figure 4.33: Dataset 2: The extracted OpenStreetMap intersections shown as red crosses along the vehicle path

Dataset 1, our algorithm performs well when the density of the roadworks is low (i.e. sparse placement of traffic cones and signs). Figure 4.34 visually summarises our system performance when other vehicles or roadworks are present in Dataset 1.

The fourth and final challenging scenario is found in Dataset 2 where residential area roads have driveways level with the road, leading up to houses. The estimated road boundaries must exclude these residential driveways even though they form part of the road surface (i.e. no curbs are present at driveways). Figure 4.35 illustrates our system performance when driveways are present, when a series of vehicles are parked on the street, and when other vehicles occur on the road in Dataset 2.

4.7.2.6 Comparison

We compare our road boundary detection system to the state of the art line fitting approach based on approximating the ground plane in a single laser scanline as a line segment. This method was shown to be very effective by [52, 69, 73, 147]. Specifically, we follow the method outlined by Kim et al. who propose that the longest straight line in a single laser scanline represents the road surface, with curbs located at the endpoints [73]. We compare this line segment approach to our method, which uses Range Filter and

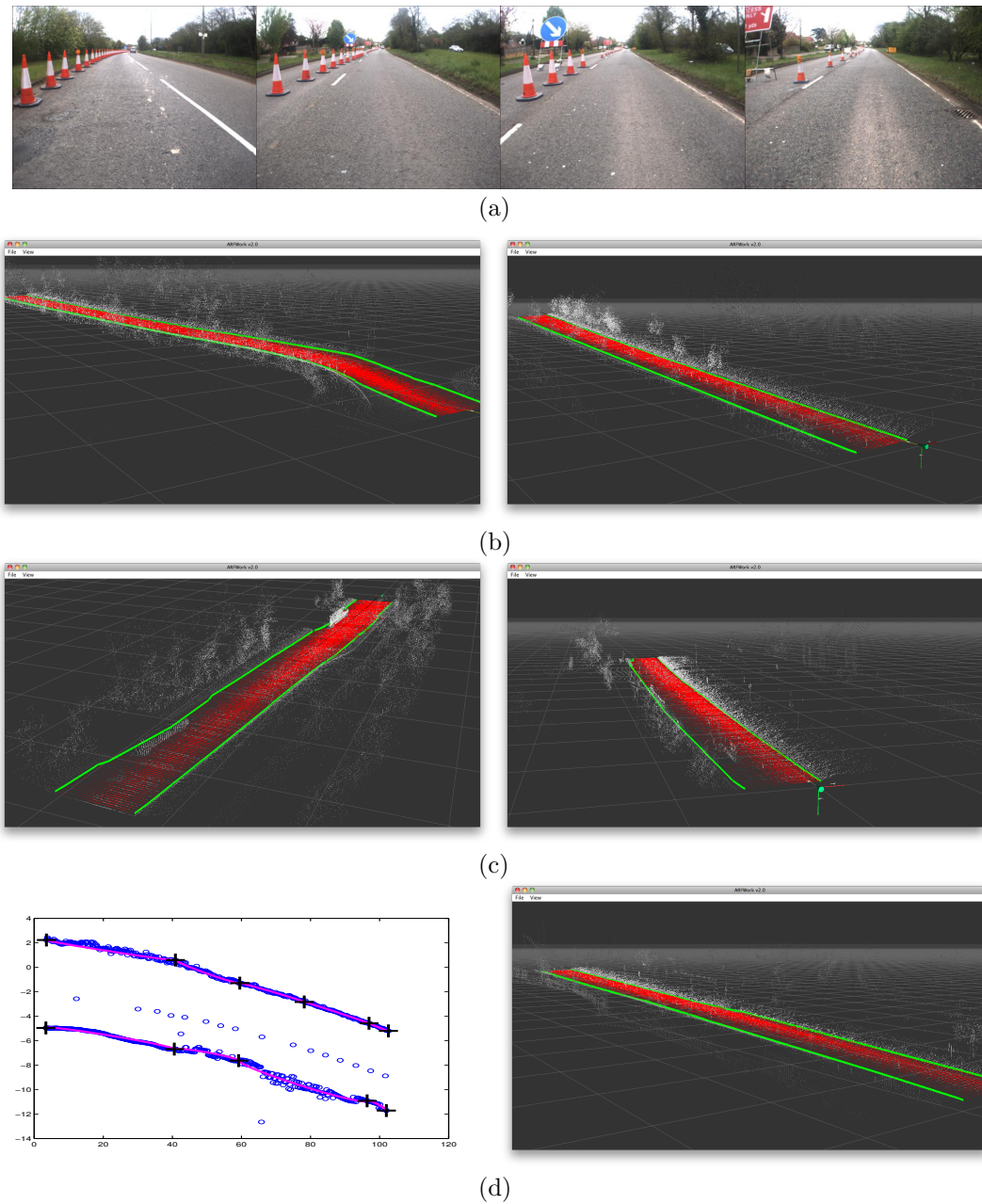
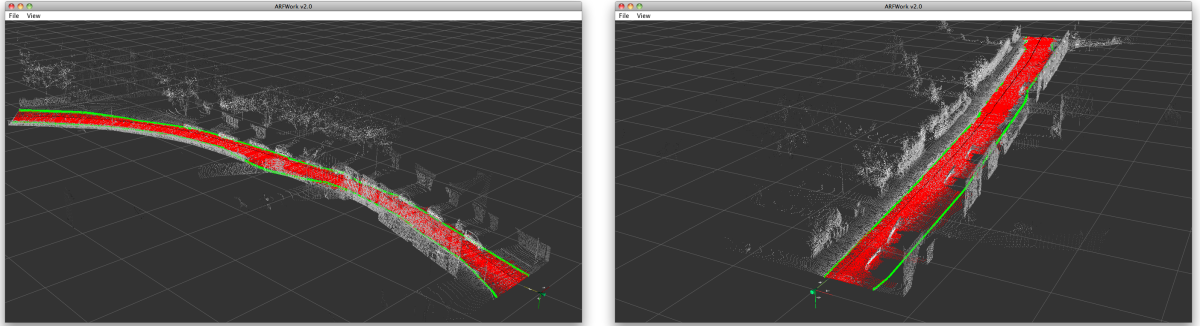
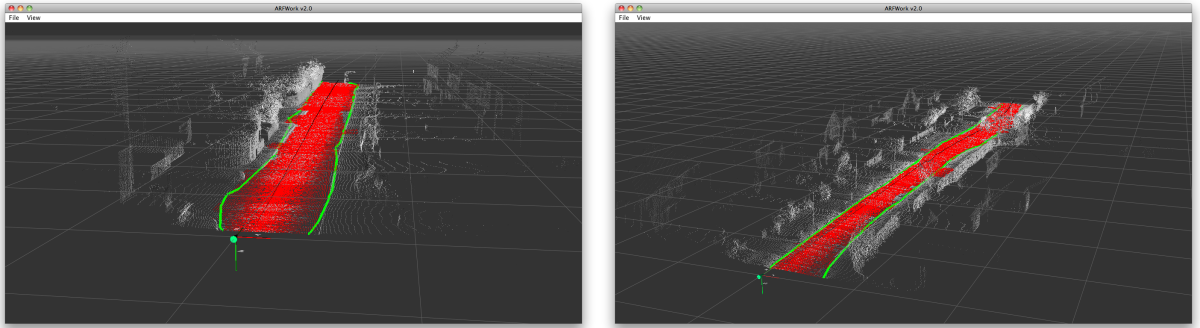


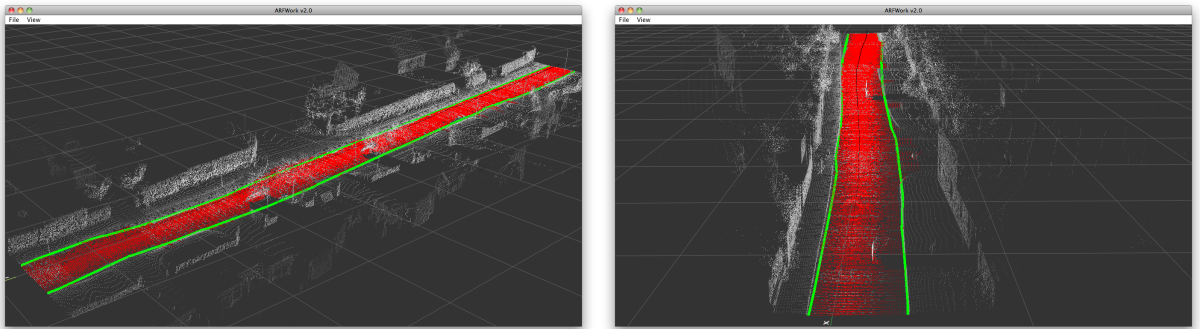
Figure 4.34: Dataset 1: (a) Roadworks found in Dataset 1 are outlined by traffic cones and traffic signs that occlude the laser scanner’s view of road boundaries. Furthermore, the placement of traffic cones varies in density. (b) Left: Successful road boundary detections despite the presence of roadworks (traffic cones). Right: successful road boundary detection when another vehicle is on the road and traffic cones are present. (c) Two failure cases. Left: The presence of several vehicles on the road results in inaccurate boundary estimation. Right: Densely placed traffic cones result in our algorithm falsely detecting the cones as road boundary locations. (d) Left: The output of our road boundary point estimation algorithm (blue dots). Here, the sensed traffic cone points in the middle of the road were successfully ignored by our BaryArc boundary estimation algorithm (magenta curve). Right: The corresponding 3D output showing correctly estimated road boundaries in the presence of roadworks.



(a)



(b)



(c)

Figure 4.35: Dataset 2: (a) Incorrect road boundary estimates due to a series of parked vehicles. (b) Driveways are level with the road surface thus they are also segmented by our road surface detection algorithm (red shading). However, even in this challenging scenario, the correct road boundaries are still estimated because the driveway points are detected as outliers by our algorithm. (c) Correct road boundary estimation even when other vehicles are present on the road.

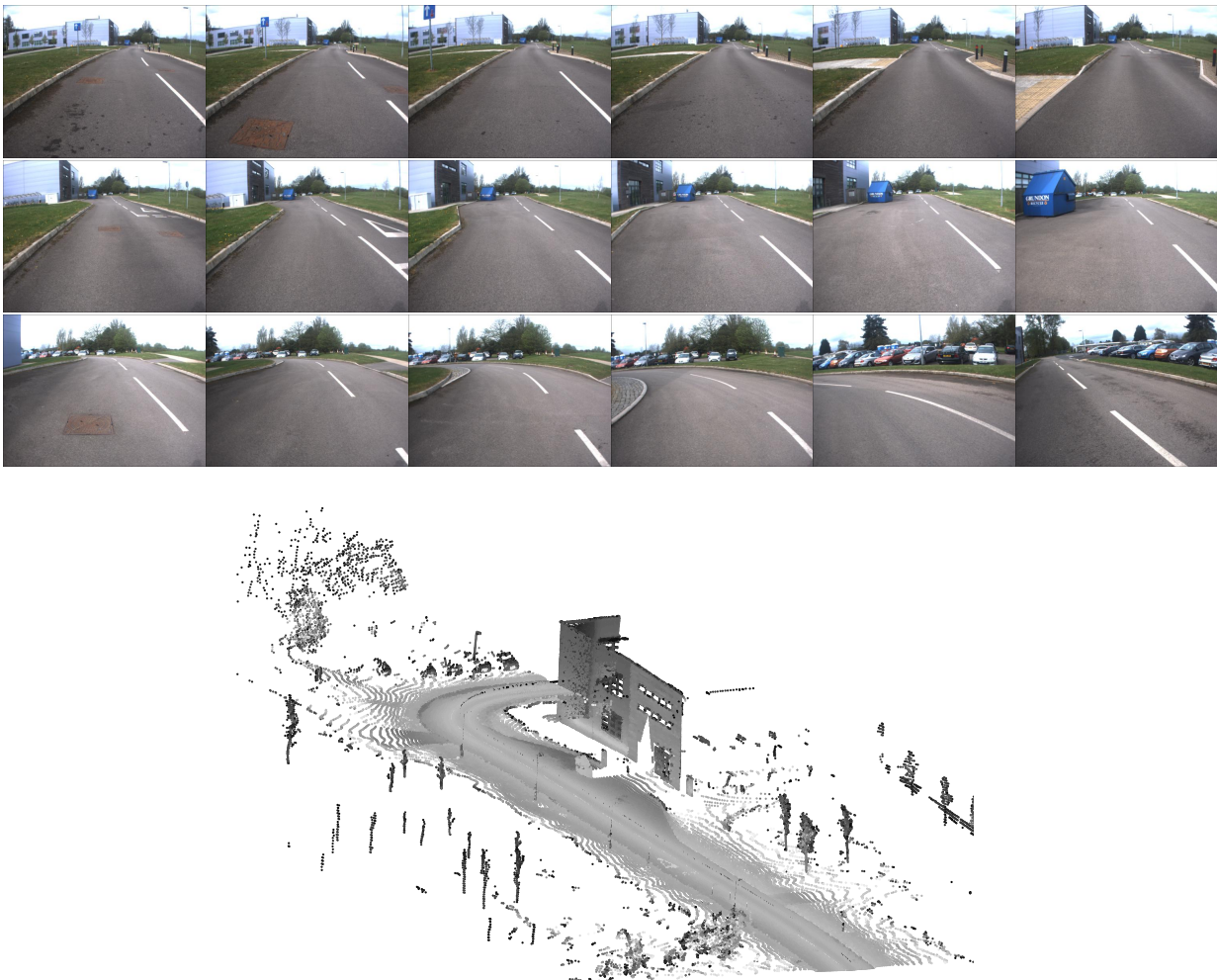
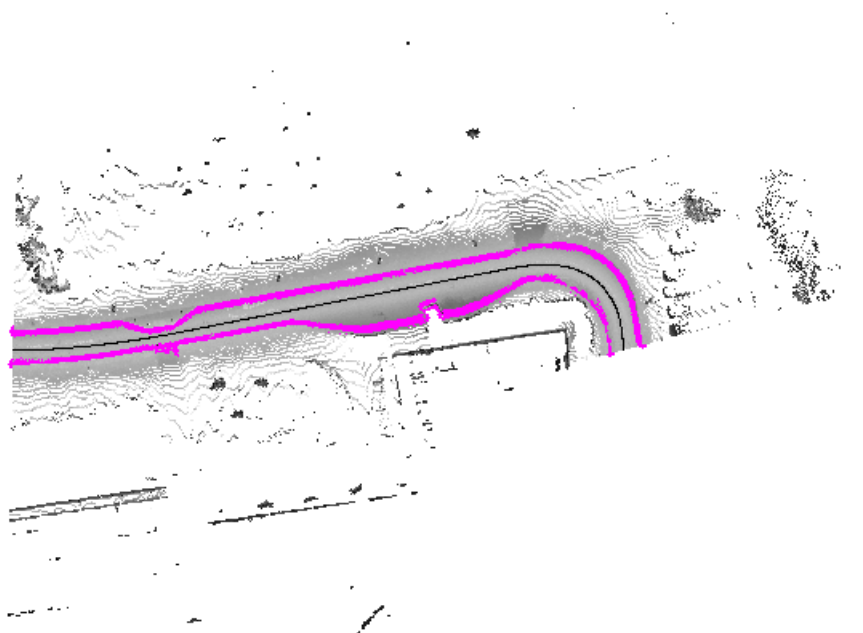
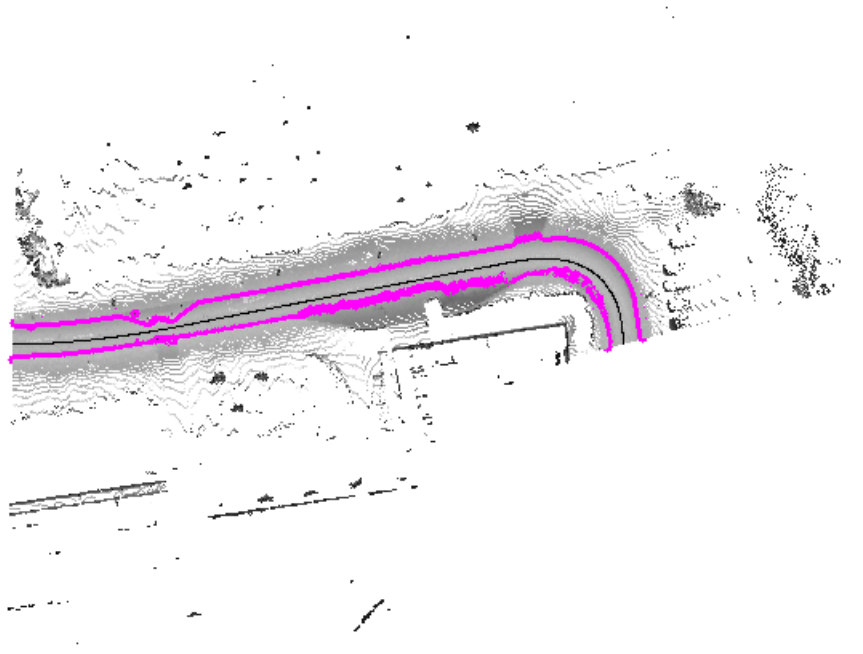


Figure 4.36: Road section considered for comparison experiment. Notice that the curb geometry narrows to accommodate a small pedestrian crossing and widens to accommodate temporary parking in front of the building.

Tangent Normals for feature extraction and automatically detects the road boundary points based on change point analysis [134]. Figure 4.36 shows the data used for the comparison - a long section of road with curb boundaries that narrow (to accommodate a pedestrian crossing) and widen (to accommodate a parking area next to a building). Our algorithm is able to capture this change in geometry by correctly detecting the road boundary points, whereas the line fitting algorithm fails to detect the widening road in front of the building as shown in Figure 4.37. The final extracted 3D road boundaries are shown in Figure 4.38, which demonstrates that our algorithm outperforms the line fitting approach in terms of curb detection accuracy and capturing changing road boundary semantics.

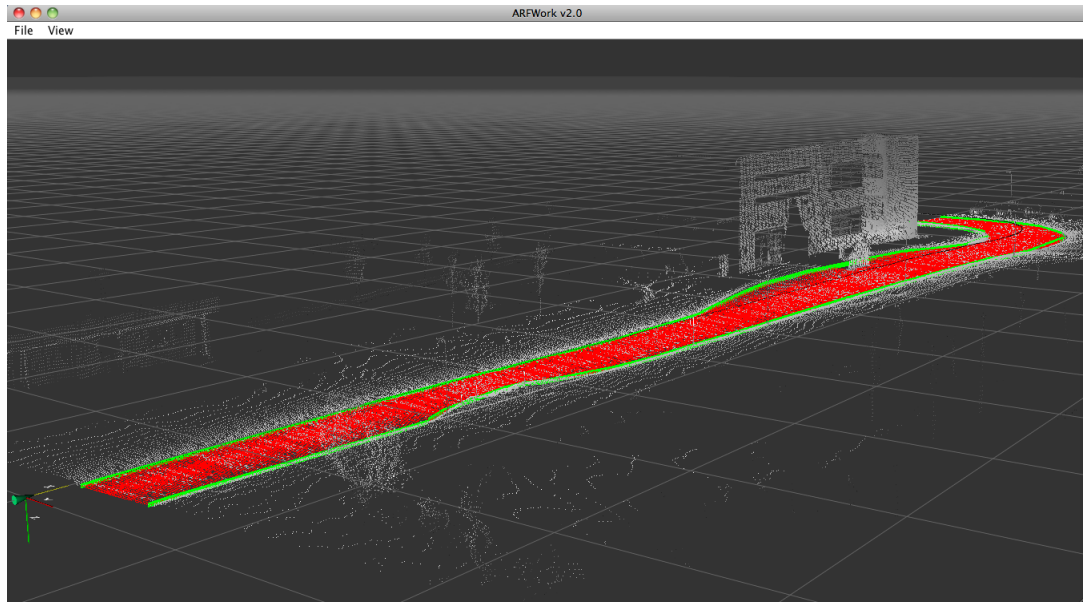


(a)

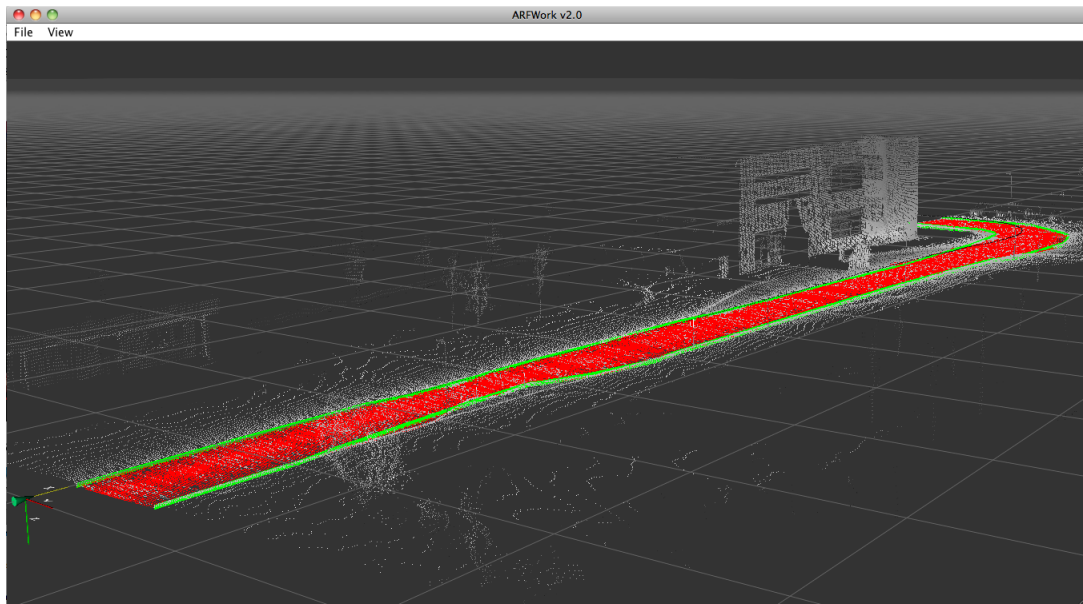


(b)

Figure 4.37: Overhead view of the road boundary points detected using (a) our algorithm and (b) the competing algorithm. The changing curb geometry is more accurately captured using our method.



(a)



(b)

Figure 4.38: The final road boundary estimates using (a) our algorithm and (b) the competing algorithm. The changing curb geometry (increase and decrease of the road width) is accurately captured by our method.

4.7.2.7 Overall System Performance

The screenshots presented in our qualitative analysis show the automatically segmented road surface (red) and estimated BaryArc boundaries (green). The entire laser point cloud is plotted to capture the context in which the detected boundaries should be interpreted. Our datasets contain structured environments (with curbs), less structured environments (with driveways or intersections), and challenging scenarios such as roadworks and other vehicles. Figures 4.34 - 4.44 give a qualitative feel of how well our system works across different contexts. Notice the smoothness, curvature, and constantly varying width of the estimated road boundaries within the whole point cloud containing buildings, trees, pedestrians and other vehicles.

For Dataset 1, Figure 4.39 shows the 3D point cloud, associated projected 2D BaryArc road boundaries with the transition points (see Section 4.6.5) drawn as crosses, and the final road boundaries shown in the 3D world. Our BaryArc model can accommodate a wide range of road boundary types because it is based on Highway Engineering geometric road design principles. Similarly, Figure 4.40 illustrates the same principles for Dataset 2. Furthermore, a montage of final results for our BaryArc road boundary estimation algorithm is shown in Figures 4.41 and 4.42 for Dataset 1, and Figures 4.43 and 4.44 for Dataset 2. Our algorithm performs well across a broad range of roads including urban, rural and residential roads.

4.8 Chapter Summary

In this Chapter, we presented algorithms for road boundary estimation based on a single vertically mounted push-broom 2D SICK laser scanner. Our algorithms processed each scanline individually and we exploited the road surface point density property of our unique sensor configuration to develop simple, yet elegant algorithms for road surface (groundstrike) segmentation and road boundary point detection. We formulated a new curve fitting model, the BaryArc, based on geometric prior information from Highway Traffic Engineering. Our algorithms were evaluated on data from our autonomous vehicle. Furthermore, we used our final BaryArc road boundaries to extract the road continuity and curvature semantics (e.g. roundabouts), and OpenStreetMap to extract intersection locations which we add to our Situational Awareness data structure described in Chapter 2. Thus far, this data structure contains road marking and road boundary semantics which describe *reading the road surface*. In the next Chapter, we introduce a biologically inspired framework for *reading the road scene*, namely roadwork recognition which is the final road semantics we will add to our Situational Awareness data structure.

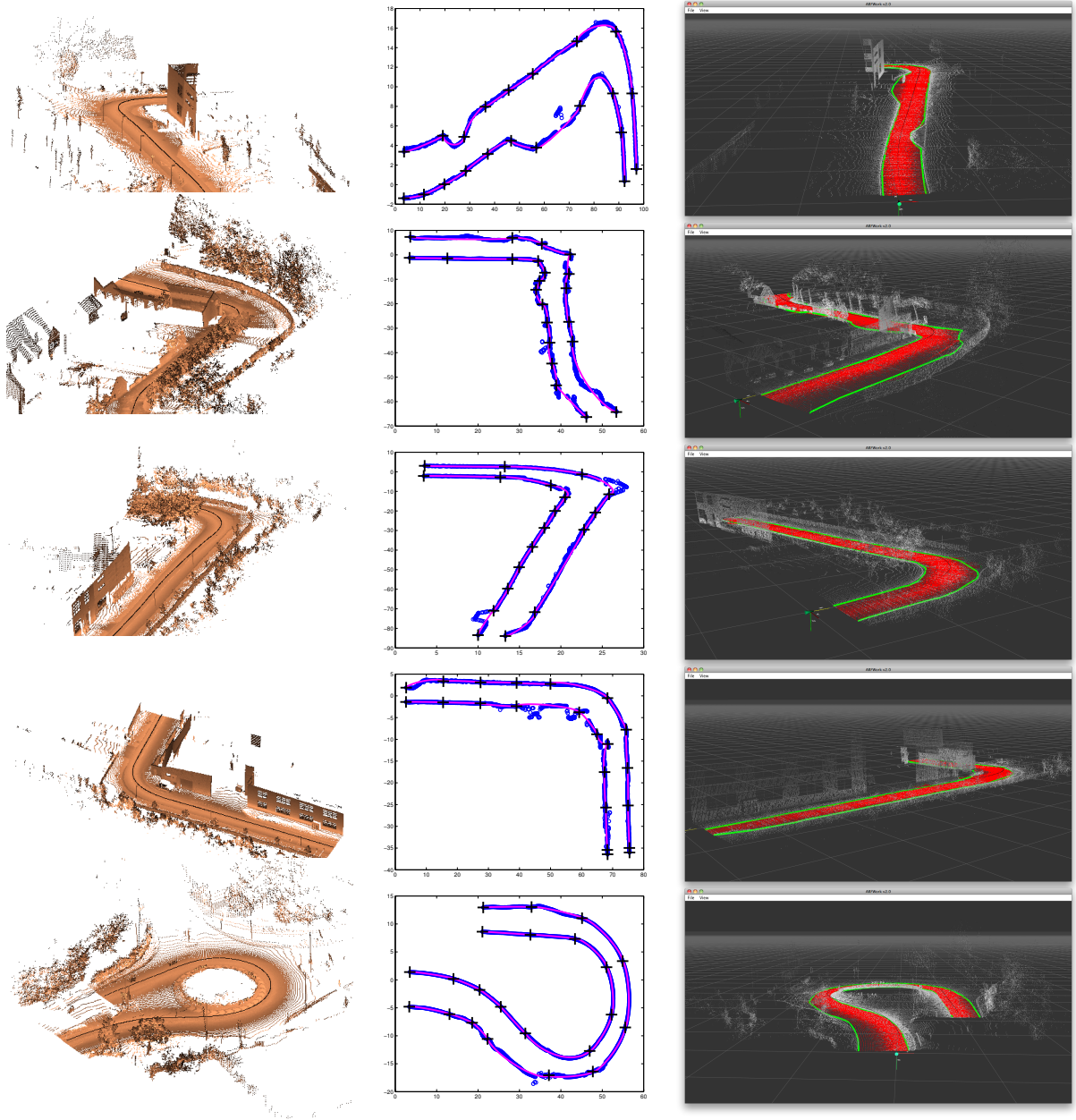


Figure 4.39: Dataset 1 results. Left: The laser point cloud under consideration, coloured by intensity. Middle: Projected 2D BaryArc Model (magenta) fit for road boundaries with the raw laser estimated boundary points (blue dots) and curve transition points (black crosses) from Section 4.6.5 shown. Right: The final output of our algorithm is the road surface segmentation (red) and the BaryArc road boundary estimation (green).

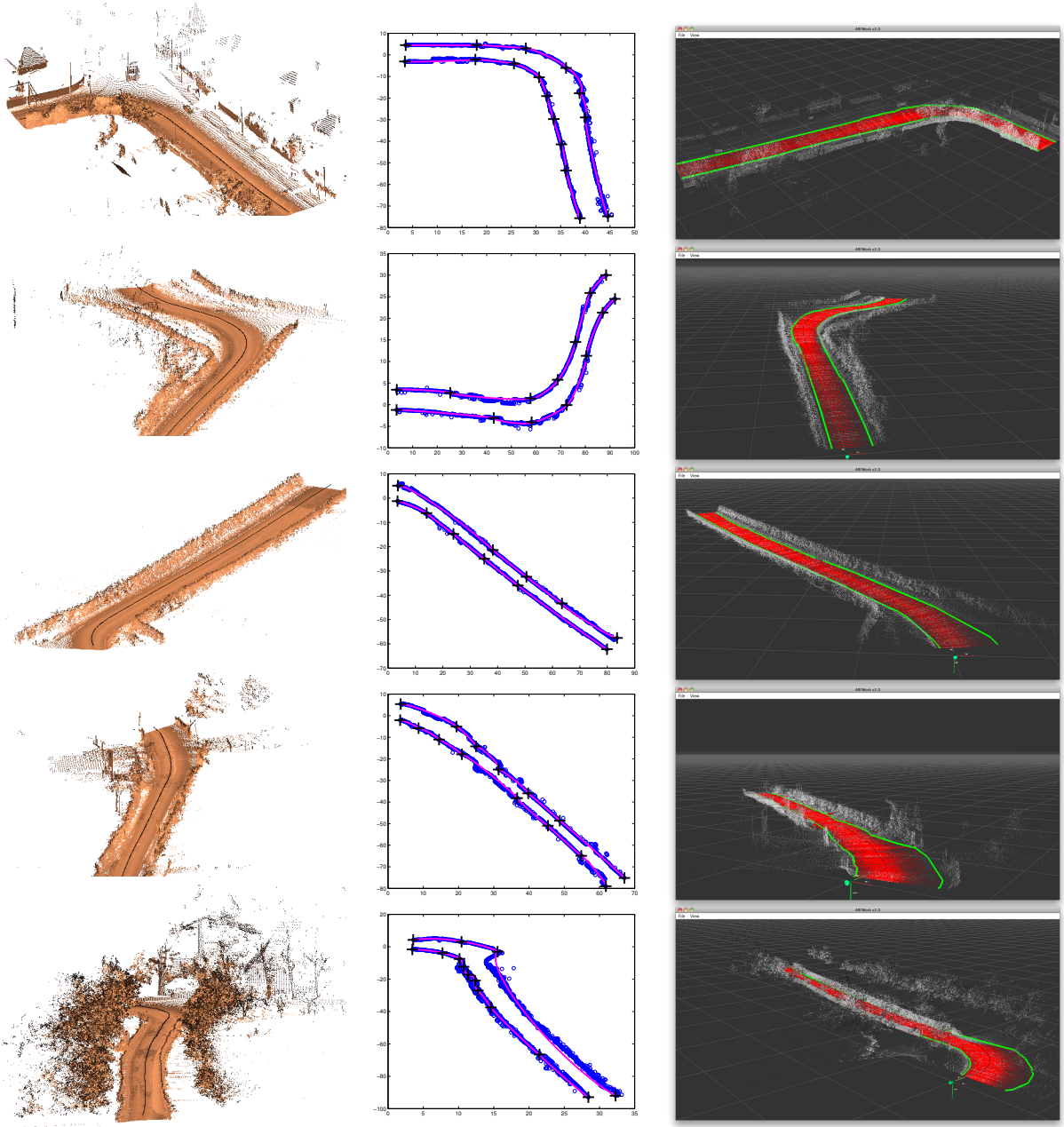


Figure 4.40: Dataset 2 results. Left: The laser point cloud under consideration, coloured by intensity. Middle: Projected 2D BaryArc Model (magenta) fit for road boundaries with the raw laser estimated boundary points (blue dots) and curve transition points (black crosses) from Section 4.6.5 shown. Right: The final output of our algorithm is the road surface segmentation (red) and the BaryArc road boundary estimation (green).

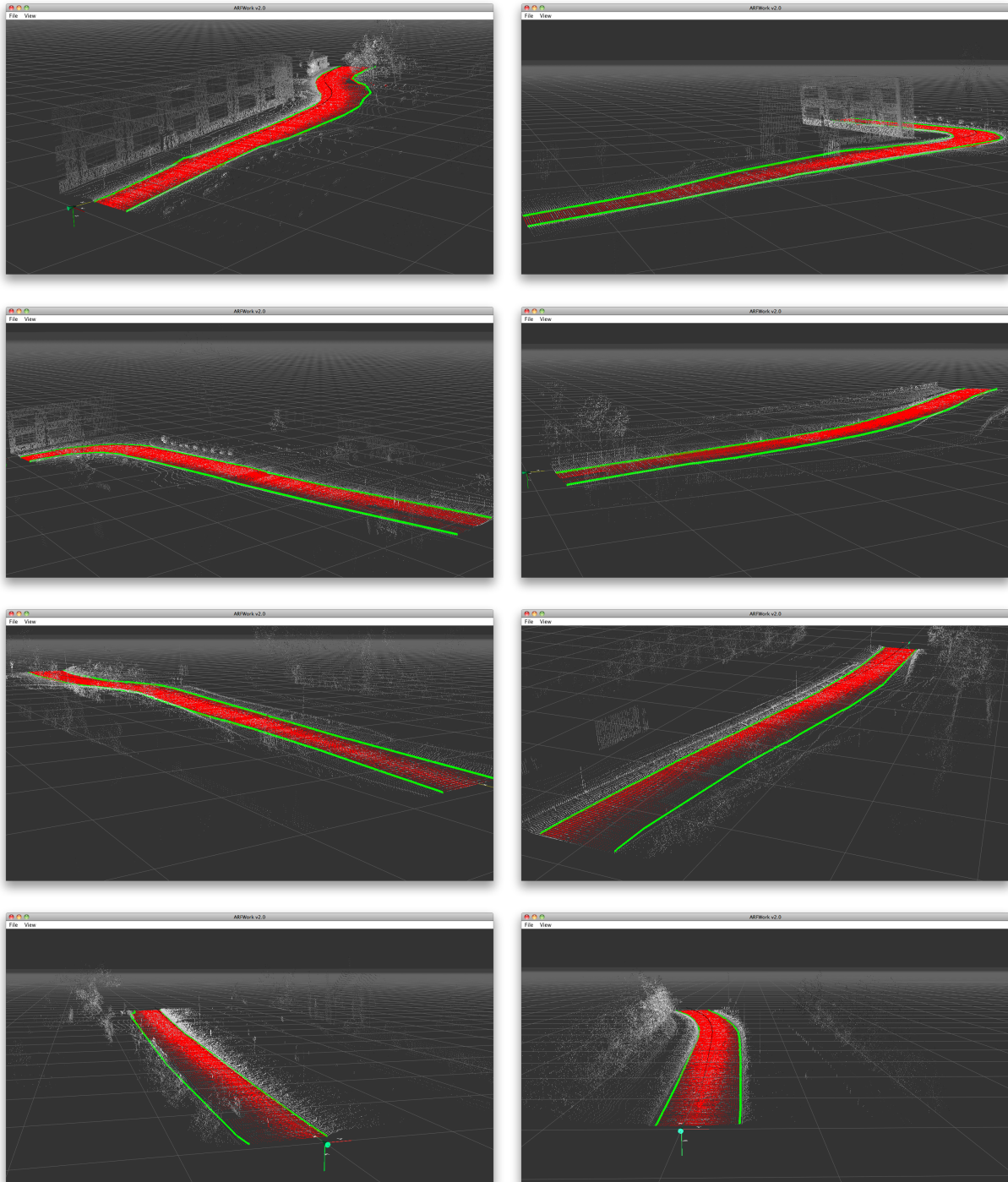


Figure 4.41: Road boundary estimation results for Dataset 1. The road surface segmentation results are shown in red and the BaryArc boundaries in green.

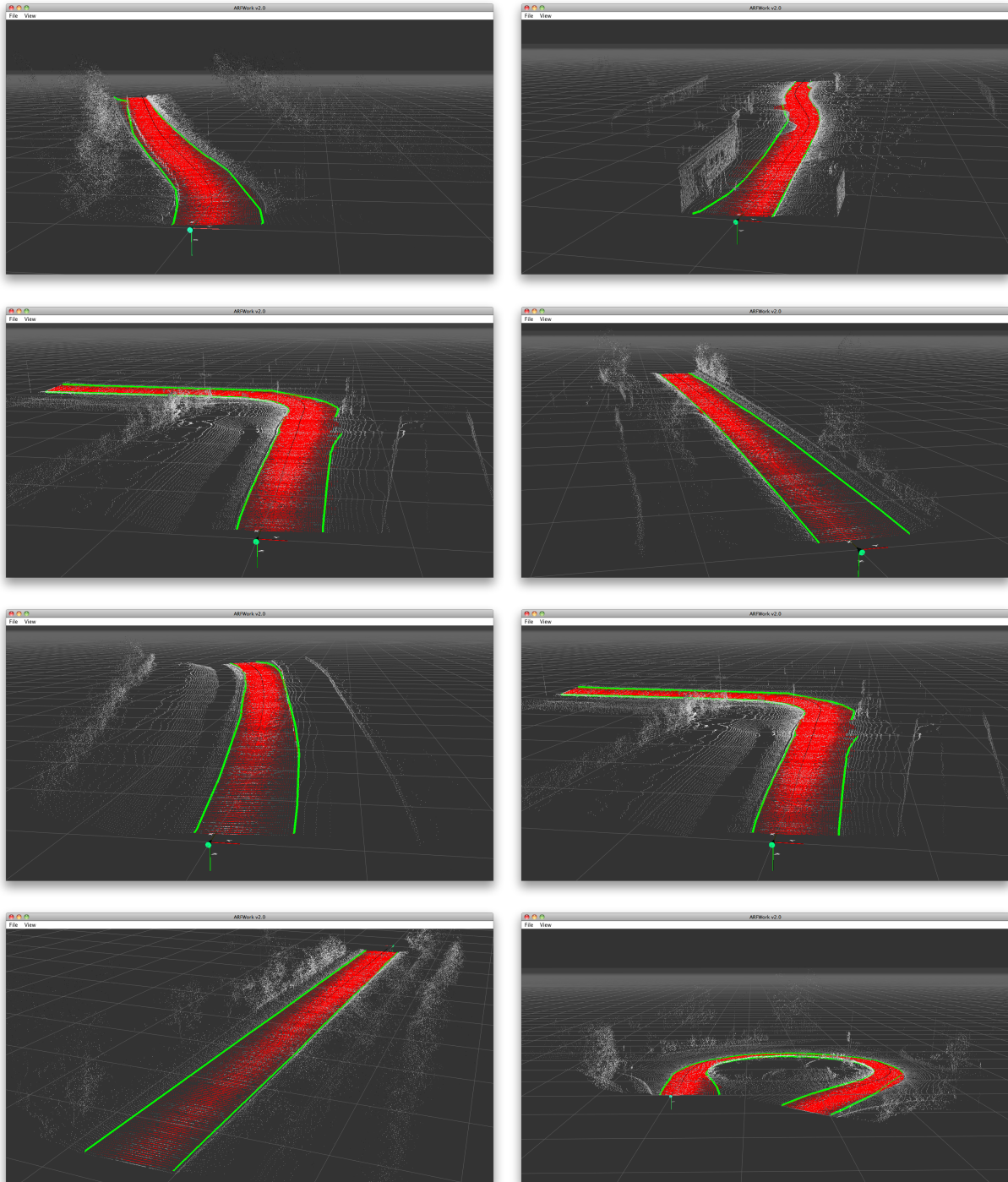


Figure 4.42: Road boundary estimation results for Dataset 1. The road surface segmentation results are shown in red and the BaryArc boundaries in green.

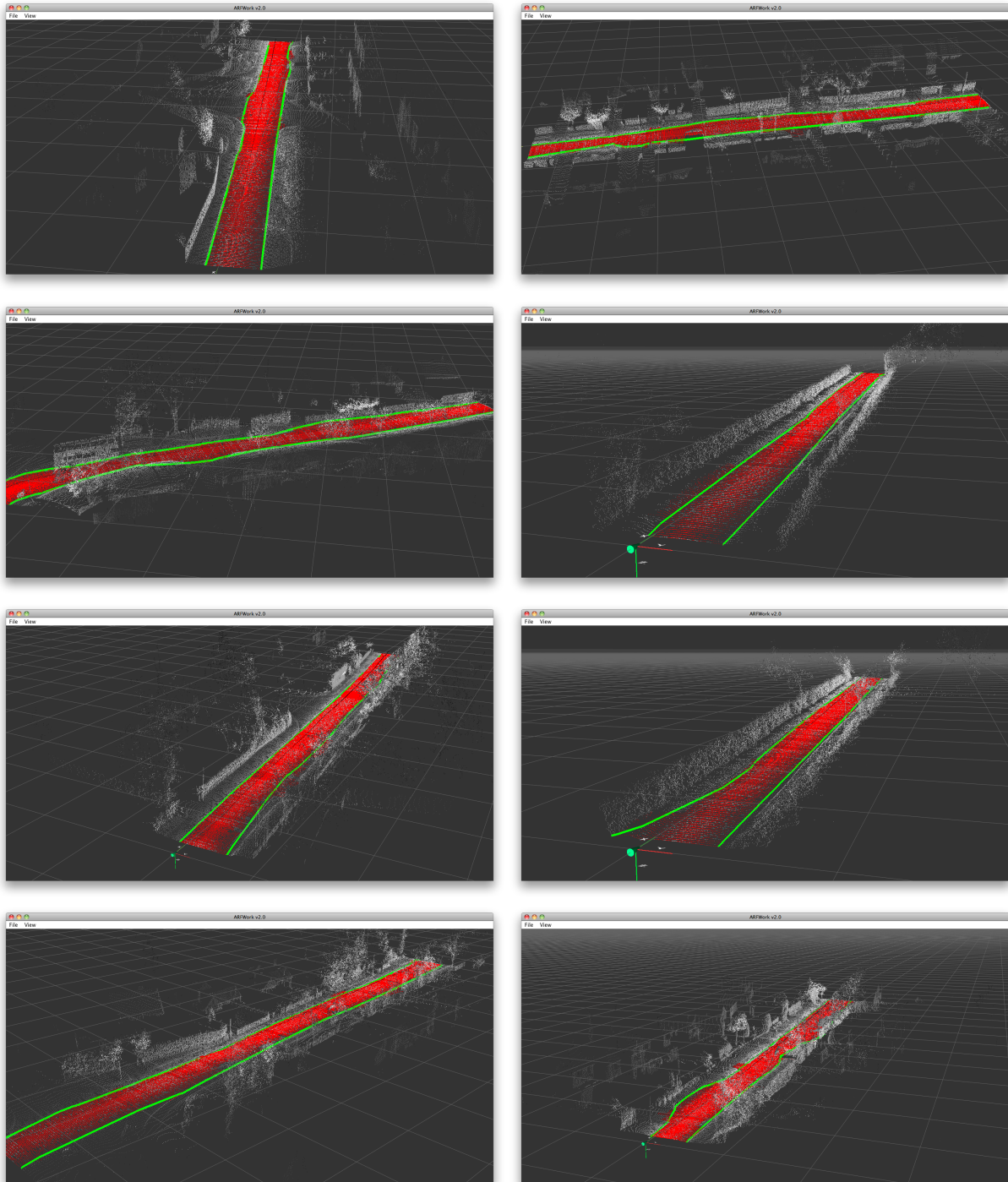


Figure 4.43: Road boundary estimation results for Dataset 2. The road surface segmentation results are shown in red and the BaryArc boundaries in green.

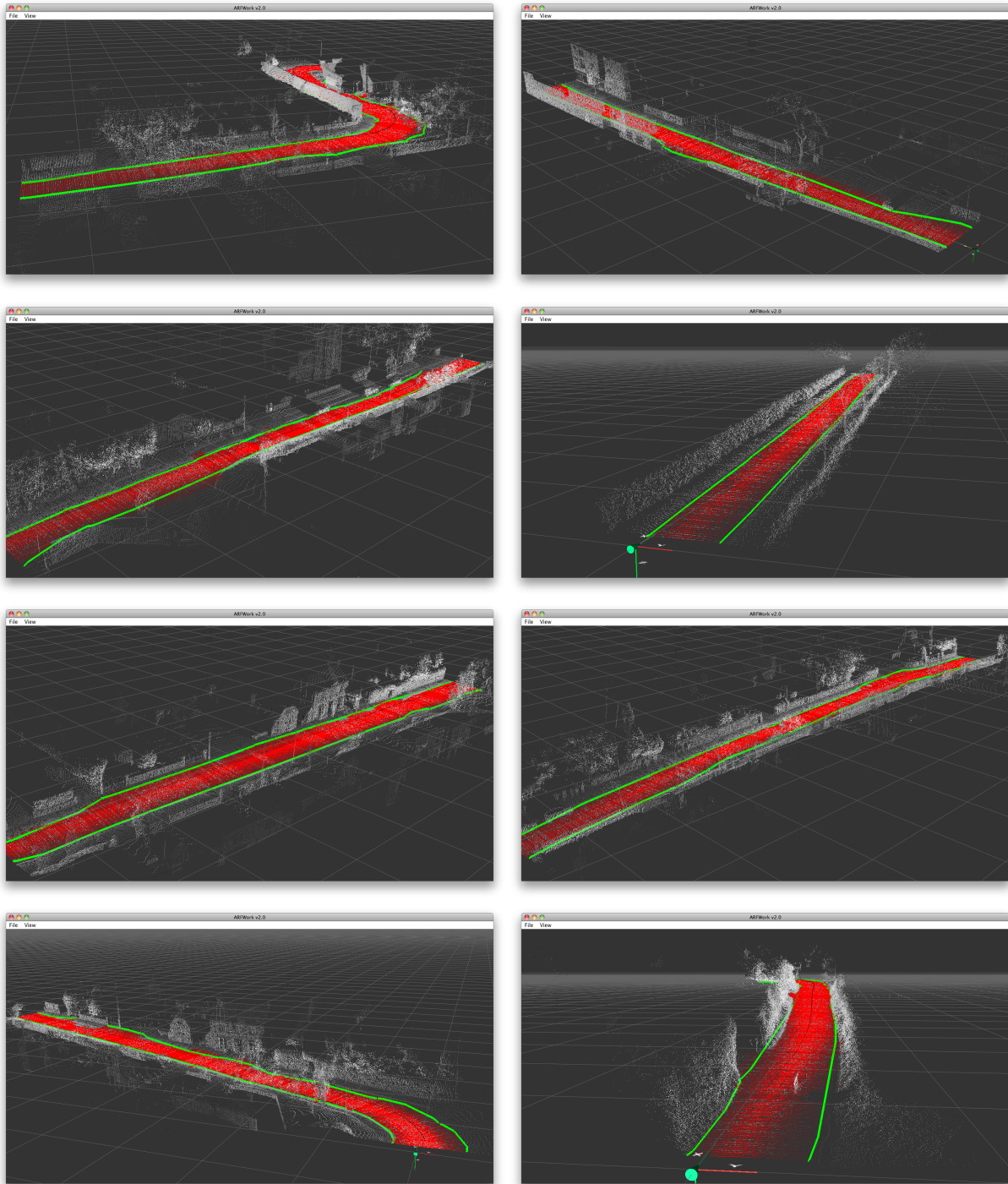


Figure 4.44: Road boundary estimation results for Dataset 2. The road surface segmentation results are shown in red and the BaryArc boundaries in green.

Chapter 5

Reading the Scene: Roadwork Scene Recognition

In the previous Chapters, we discussed the algorithms used for deriving semantic information about road markings and road boundaries which we added to our Situational Awareness data structure. We now turn our attention to situational awareness in dynamic road environments. Specifically, we consider scenarios involving road construction, road surface maintenance, and accident scenes bounded by roadwork objects. The results of this Chapter will be captured in the “Works” Google protobuf message type of our Situational Awareness data structure introduced in Chapter 2.

The presence of roadworks has important implications regarding the validity of prior maps used for navigation by autonomous vehicles. This Chapter addresses the problem of quickly and robustly assessing the gist of traffic scenes for whether roadworks might be present. Our method exploits the engineered visual saliency of traffic cones, construction barriers and traffic signs without explicitly modelling them individually. We draw inspiration from opponent colour vision in humans to formulate a novel roadwork scene signature. Finally, we apply our roadwork scene signature to the task of roadwork scene recognition, within a classification framework based on soft assignment vectorization and RUSBoost.

5.1 Literature Review

The presence of roadworks on a road alter both the road geometry and previously known traffic rules. Roadworks result in new temporary road markings, the placement of traffic cones and barrier objects on

the road, new temporary traffic signs (or lights) and traffic congestion. Although human drivers can learn to quickly adapt to dramatic scene changes, autonomous vehicles are faced with the challenging task of automatically perceiving and understanding semantic cues present at roadwork sites, and interpreting their meaning. Situational awareness at roadwork sites can therefore be viewed in terms of three sub-problems:

- The detection of roadwork objects such as traffic cones, specialised traffic signs, traffic lights and different coloured road markings
- The reliable recognition of roadwork scenes (sites) based on changing scene semantics
- The mapping of roadwork zones based on prior information and local observations

We address the second sub-problem in this Chapter, and the first and third sub-problems were previously addressed in our work [91], published at ITSC 2012.

Thus in this review, we consider the roadwork object and scene recognition problem. The frequency of occurrence of roadworks is surprisingly high. In 2009/2010 alone, Transport for London reported an estimated 370 000 roadworks [33], a figure that is typical for most major cities worldwide. In spite of this, limited research has been conducted in the area of autonomous driving at roadwork sites and this thesis makes a key contribution in this area, namely roadwork scene recognition.

5.1.1 Roadwork Object Recognition

Object recognition plays a huge role in the perception layer of semantic awareness in autonomous vehicles. A large corpus of literature exists in this area so we limit our discussion to the detection of roadwork objects (see Figure 5.3) in images. Generally speaking, roadwork object recognition can be divided into the detection of traffic signs, barriers, cones/beacons and temporary road markings. Traffic sign detection for autonomous vehicles is often posed as a four step process: a pre-processing background subtraction or colour segmentation [47] step to identify a region of interest most likely to contain traffic signs. This is followed by a detection step involving feature extraction at the location of the sign using either colour-based or shape based methods. A classification step then associates the sign with a specific class label. In order to discount false detections, a final tracking step is often done for temporal smoothing using a predictive model (Kalman filter for example) that predicts the location of the sign making traffic sign detection more robust to occlusions (see Figure 5.1). Several challenges arise in traffic sign detection work: varying lighting conditions, shadows or occlusions, changing pose and scale, shape deformations and reflections, inter and intra class variations, faded or dirty signs and background clutter [80, 94], making it currently impossible

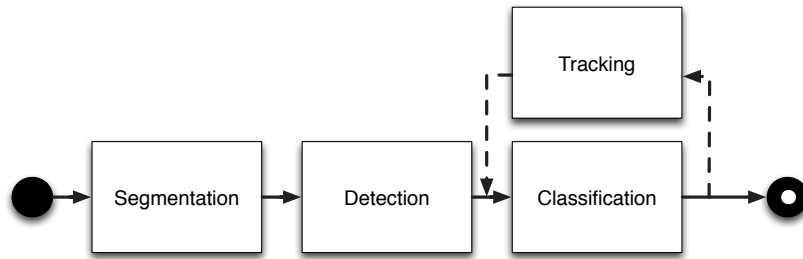


Figure 5.1: The typical flow of traffic sign detection systems: the segmentation step identifies image regions potentially containing traffic signs. The detection steps identifies candidate traffic sign blobs which then pass through a classification step to assign a class label to the sign. Finally, classifications are refined by a tracking step which is for temporal smoothing of predictions.

to design perfect detectors. Furthermore, true situational awareness goes beyond sign recognition. The relevance of a sign (based on its orientation relative to the vehicle) must be taken into account as well as focusing computational resources on signs, such as pedestrian or wildlife signs, which are more easily ignored [94].

Another important consideration for situational awareness in autonomous vehicles is real-time detector performance. To achieve this objective, simplifications such as colour segmentation (rather than slower model shape based segmentation), linear colour spaces, pattern matching and edge based shape detection techniques are preferred [16]. A purely feature matching GPU based implementation has also been proposed before [30].

A good review of current state of the art vision-based traffic sign detection and analysis can be found in [94], where open research questions such as the role of context and localisation in traffic sign recognition and the over representation of European traffic signs are discussed. These challenges are not unique to traffic sign recognition but apply to object detection problems in general.

For mapping applications, preserving 3D information is crucial, therefore laser sensors are sometimes preferred over cameras for roadwork object detection. Wimmer et al. [148] developed laser based algorithms (contour features derived from hull curves) for automatically detecting and classifying safety barriers (traffic pylons, guard rails, and Jersey barriers) and vision based algorithms for temporary road marking detection [150] at roadwork sites on motorways. Their work focused specifically on creating a “Road Work Map” for motorways. They detect concrete barriers which are not often used to denote roadworks on city roads (bright coloured plastic barriers are used instead) thus new algorithms must be developed for inner city or urban roadworks. In later work, [151] use this “Road Work Map” as a prior map for localisation at motorway construction sites. The localisation is based on refining a GPS location estimate

by matching detected beacons to ones in the prior map. The prior map is transformed to the local sensor frame which reduces the problem to a 2D point set matching problem. Crucially, the algorithms presented by [149, 151, 150] ignore colour information when detecting roadwork objects.

Roadwork object recognition also includes the challenging task of detecting valid road markings appearing at road work zones. Such markings are typically temporary and coloured bright yellow to distinguish them from previously valid (and still visible) white markings. A probabilistic method based on geometric analysis is used by [49] for estimating and tracking temporary lanes at roadwork sites on motorways. Interestingly, colour information is completely ignored as a grayscale camera is used for greater robustness to illumination changes, even though temporary markings are yellow. This method relies on estimating the width of each lane exactly and does not easily extend to urban city roadworks where traffic cones and barriers placed on the road make road marking detection more difficult, especially since these objects violate the planar road surface assumptions of Inverse Perspective Mapping approaches (generating a birds-eye-view of the road) to road marking extraction. Earlier work by Gump et al. in this area [50] presented a new method for detecting traffic beacons at roadwork sites. Beacons were extracted from images by exploiting the dominant diagonal red white stripe pattern and using a Hough transform along the horizontal axis to group neighbouring edges forming a vertical pile. The aspect ratio, size and texture estimates of beacon candidate regions were linearly combined to produce a confidence measure for detecting beacons. This approach involved no statistical machine learning and we note that template matching is an attractive and possibly more robust (less heuristics driven) alternative to the proposed method. To date, there has been no work exploiting the colour information of roadwork sites to develop classifiers to detect roadwork objects - this is where we make our contribution.

5.1.2 Roadwork Scene Recognition

Roadwork scene recognition is the task of automatically recognising the presence of roadworks along a road. In general, scene recognition can be framed as a global scene understanding problem where a descriptor is used to classify an entire image as belonging to a particular class. Alternatively, the problem can be viewed in terms of local scene recognition involving the classification of a scene based on co-occurrences of specific objects within the scene. We begin by discussing the latter approach.

Specialised roadwork signs and objects are typically used to denote roadworks (Figure 5.3 summarises British roadwork objects). The occurrences of such objects can be used to frame roadwork scene recognition as a classic object detection problem. Such an approach has been used for recognising temporary changes

due to roadworks on highways [123], based on the detection of workzone signs in order to identify roadwork bounds (beginning/end points) on highways - see Figure 5.2. Hue and saturation was used to learn a colour model to segment road signs, followed by an Adaboost based classifier. Such an approach has obvious advantages such as the ability to infer temporary driving conditions such as decreased speed or a blocked lane based on the interpreted meanings of recognised signs. The robustness of the object recognition approach is however limited to very high class precision/recall performance on the part of the object detectors used. In fact, Seo et al. note that incorrect or missed sign detections result in misclassified temporary highway changes and roadwork zones even when temporal smoothing and classification score propagation is used to reduce false negatives. Furthermore, the discrete placement of traffic signs at roadwork sites motivated the Gaussian smoothing step to interpolate the confidence values - we note that our Roadwork Prior work [91] formulates a Gaussian Process model which outputs an inherently smooth function describing the probability of roadworks being traversed given local in-situ and web based observations.

A similar approach to roadwork scene recognition on motorways was adopted by Gump et al. [50] who first proposed the joint detection of traffic beacons and yellow temporary road markings in order to classify construction sites (see Figure 5.3). The start and end of construction sites on motorways was automatically triggered based on the co-occurrences of beacons and yellow road markings.

These object recognition approaches to roadwork scene detection, described above, do not account for context. Oliva and Torralba proposed that context plays a huge role in object recognition [103] because real-world objects do not occur in isolation but co-vary with other objects and the environment. In fact, the presence of a particular object constrains the identity and location of other nearby objects in the scene. Context therefore affects scene semantics and this is especially true at roadwork sites where a useful statistical summary of the road scene could highlight the roadwork region.

In computer vision, recent work has shown how the class (e.g. beach, street, playground) of an image scene can be inferred from the aggregated statistics of low level features (such as local image gradients, oriented edges or image patches) [102, 36] in order to recognise entire scenes. The scene is therefore represented as a whole using image descriptors - such models represent the state of the art in scene recognition and context based object recognition. For these recognition frameworks, the first step is to identify salient and stable regions (e.g. Fast-Hessian Detector [6], Features from Accelerated Segment Test [118], or Convolution Filters [116]) in the image, known as keypoints. Global features (image descriptors) such as local image gradients or binary patches can then be computed at these keypoints - well known



Figure 5.2: Some of the workmen signs considered in roadwork zone recognition work done by [123]. Image taken from source.



Figure 5.3: Typical roadwork objects (from left to right): Construction sign, merge sign, traffic light ahead sign, keep left sign, keep right sign, plastic barrier, traffic cone, traffic light, concrete barrier and beacon. The occurrences of such objects is typically exploited for roadwork scene recognition tasks.

examples of such descriptors include Scale-Invariant Feature Transform (SIFT) [88], Speeded Up Robust Features (SURF) [6], Binary Robust Independent Elementary Features (BRIEF) [18], and Finite Element Laplacian Feature Detector (FELF) [71].

By clustering many such descriptors over a vast number of images, a visual word dictionary of cluster centroids can be learnt - this describes the well known Bag-Of-Words model [154]. Using Bag-Of-Words, an image can be represented as a vector summarising the frequency of occurrence of each visual word, in the dictionary, for that image. Attempts to preserve context information (at a coarse level) have included spatially organised texture models where the image is divided into sub regions and for each sub-region, a stationary texture histogram vector is computed [103].

In other related work, colour information has also been exploited in object scene recognition. A good review of recent colour descriptors and the advantages/disadvantages of different colour space transformations can be found in [139], where experiments show improved performance compared to when colour information is ignored. We draw inspiration from the literature in colour descriptors and context based scene recognition in order to develop an opponent colour vision based framework for fast recognition of roadwork scenes without explicitly modelling any objects in the scene.

5.2 Understanding Roadworks

The layout of a roadwork scene typically depends on the underlying cause of the disruption. This in turn directly influences the objects used to denote a roadwork scene. For example, highway maintenance is typically denoted by traffic cones and traffic signs, while city works are denoted by traffic signs, construction barriers and traffic cones. Directly modelling a roadwork scene requires a formulation for the relationships between these different objects within the scene. A naive approach is to detect individual objects in the scene and classify a scene based on occurrences of roadwork objects. Such an approach is both computationally expensive and requires training multiple object detectors. Our more sophisticated model however, describes the entire roadwork scene using a global signature without explicitly modelling individual objects within the scene. This approach is inspired by the relationship between opponent colour vision and the design of roadwork objects.

Roadworks introduce new infrastructure on the road, thus dramatically changing its semantic layout. The presence of roadworks therefore negatively affects the accuracy of prior maps [91] which have recently been exploited to constrain tasks such as traffic light detection and mapping [35, 86]. Such prior maps can no longer be trusted at roadwork sites and autonomy should potentially be revoked in a safe manner in order to safely navigate roadworks. There therefore exists a need for an efficient mechanism to quickly recognise roadwork scenes in order for an autonomous vehicle to update prior assumptions it has about the world and gain a true sense of situational awareness. Such a system has several important applications such as offering manual control to a human driver when autonomy is limited or difficult, and cuing the automatic switch from a “normal” to a “roadwork” autonomous navigation algorithm - this motivates our work. Figure 5.4 illustrates our proposed interface for roadwork scene recognition.

5.2.1 Colour in Scene Recognition

Previous work on roadwork scene recognition focuses on the detection of roadwork objects (traffic cones and concrete barriers) as described in Section 5.1.2. Here, we focus on the role of colour in scene recognition literature.

Colour information plays a vital role in object and scene recognition, and many colour descriptors have been proposed over the years (see [139] for an overview). The Hue Saturation Value (HSV) colour space has gained popularity within the vision community where it has been applied to a large corpus of problems such as face detection [63], traffic sign recognition [124], and pedestrian detection [132]. Scene recognition is traditionally posed as a similar image retrieval problem by the computer vision community. This



Figure 5.4: Our proposed autonomous vehicle interface for situational awareness at roadwork sites: when roadworks are detected (top), manual control is offered to the driver (bottom). The roadworks shown here were successfully recognised using our roadwork scene signature.

approach requires a representation of the image as a fixed-size vector created from quantised descriptors. Many approaches have been proposed for this: Bosch et al. apply probabilistic Latent Semantic Analysis (pLSA) to a bag of words representation of each image to create a descriptor and a multiway classifier is then trained on the topic distribution of each image [13]. In similar work, Siagian and Itti [125] use a multi-scale set of early-visual features to represent an image as a low-dimensional signature vector. In scene classification literature, contextual information is often captured by modelling spatial dependencies between classified segmented regions of the image to aid scene recognition [78].

Our method differs from previous work in that spatial dependencies between regions in the image are captured by an opponent colour spatial prior resulting in features being extracted only at relevant locations in the image. Furthermore, we extend the Bag-of-words approach to implement soft assignment vectorization based on standard deviations from codebook centroids. This allows us to capture the degree of similarity between features. Our approach uses colour texture (gradient) information to formulate a roadwork scene signature.

Other colour spaces have been used before: Bratkova et al. present a practical opponent colour space, known as oRGB, useful for graphics applications such as colour adjustment, transformation, and transfer [14]. Ishizuka and Hirai segment red road signs using the HSV colour space and later perform traffic sign recognition based on opponent colour filters [66]. Other colour descriptors such as colour GIST [101] and colour SIFT [139] have also been proposed before.

The most relevant related work is the Bag-of-Colors (BOC) descriptor proposed by Wengert et al. [143] for improved image search. This descriptor is based on the generation of a CIELab colour dictionary (codebook) which is used to represent an image as a normalised histogram colour signature. This approach does not however explicitly capture spatial information and at runtime, the closest visual colour in the codebook is computed for each pixel in the image.

Biologically inspired approaches to image processing have also been proposed before. Recent work by Kerr et al. demonstrated a new method for edge detection using Spiking Neural Networks on synthesised hexagonal pixel images (which are arranged to mimic photoreceptors on the human retina) [70]. We draw inspiration from this in our work, by exploiting the biological opponent colour vision model to show that roadworks are engineered to be seen by human drivers and therefore best recognised using the colour model underlying their design. Our roadwork scene signature does not explicitly model any objects in the scene and the opponent colour spatial prior captures dependencies between regions, thus reducing the computation required but increasing accuracy.

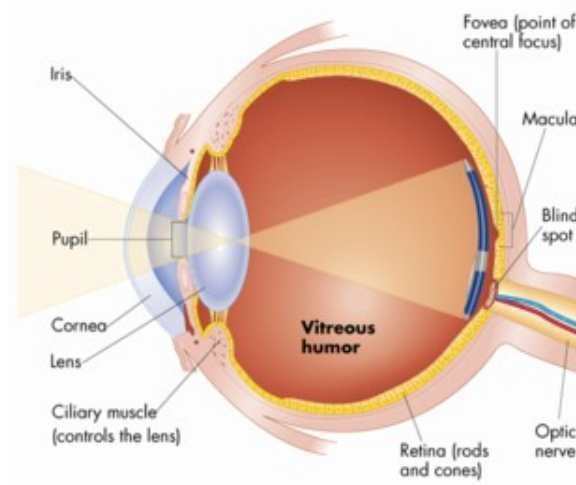


Figure 5.5: A diagram showing the cross-sectional elements of the human eye. Image taken from [29].

5.3 The Opponent Colour Model

5.3.1 Colour Vision in Humans

The perception of colour depends on three factors: the illumination source, the material of the object being illuminated and the observer (human eye or camera) [59]. In this section we focus on colour vision in humans which underlines the development of our biologically inspired roadwork scene recognition framework. The human eye is approximately spherical shaped with several bio-mechanical, bio-chemical and neurological components crucial for accurate perception of the environment. Figure 5.5 shows a diagram of the eye.

Light from the environment enters the eye through an aperture known as the pupil. The coloured region around the pupil, known as the iris, changes shape to control the amount of light entering the eye. In low light conditions, the pupils dilates and conversely constricts under bright illumination. The actual optical power of the eye is provided by the cornea and the lens mechanically adjusts the strength. For viewing objects far away, the shape of the lens is thin and becomes thicker to accommodate near sighted vision. An inverted image of the perceived world is then formed on a photoreceptive region at the back of the eye known as the retina.

This retina lines the back of the eye providing a large field of view. However, colour vision is limited for stimuli beyond about 40° on either side of the eye's axis with the remainder of the area mainly responsible for movement detection [60]. The fovea provides the most detailed and sharpest vision and is off axis around 4° giving just a 1° field of view [60]. The foveola region only contains photoreceptors known as

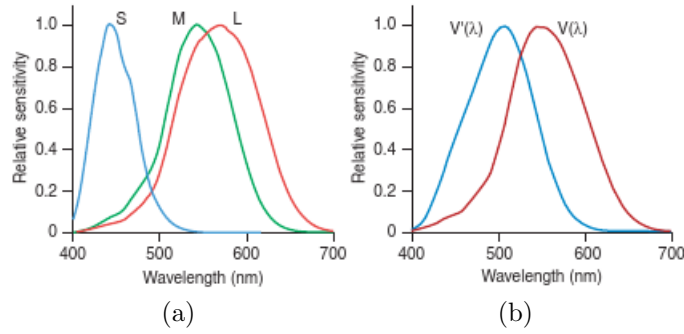


Figure 5.6: (a) Spectral sensitivity functions for cone (ρ, γ, β) photoreceptors. (b) The generalised spectral luminous efficiency functions where $V(\lambda)$ and $V'(\lambda)$ represents the weighted average of the three cone and the rod spectral response, respectively. Image taken from [46].

cones and outside this area both cone and rod photoreceptors can be found with the ratio of rods to cones gradually increasing outwards. These receptors are named due to their different shapes and are typically five hundredths of a millimetre in diameter with different spectral sensitivities [59].

Rod photoreceptors allow humans to perceive monochromatic vision in dim conditions (scotopic vision) and the cones provide colour vision under normal illumination (photopic vision). The spectral response functions of Figure 5.6, show the three types of cone receptor spectral sensitivities - Long (L or ρ), Medium (M or γ) and Short (S or β) wavelength - as well as the generalised spectral luminous efficiency function for photopic $V(\lambda)$ and scotopic $V'(\lambda)$ vision representing the combined spectral sensitivity of the eye [46].

Finally, optic nerves connected to the photoreceptors leave the eye at the blind spot (no light sensitivity) and send electro-chemical signals via neurons to the brain - the processing centre for vision. Opponent colour theory explains the perception of colour on the human visual cortex in the brain. While not studied in robotics, the opponent colour model has widely been studied by the neuroscience and psychology communities to explain the perception of colour in humans.

In general, there exists two main theoretical accounts for the formation of colour on the human visual cortex: the Young-Helmholtz and the Hering theories [40]. The former model is a theory of trichromatic colour vision based on three primary colours (red, green, blue) and their combinations to produce the full spectrum of colours perceived by humans. In contrast, Hering's theory postulates opponent colour vision involving three types of paired visual processes occurring in the visual cortex, such that each process has two opponent members. These sensory processes are antagonistic within each channel [61] and are summarised graphically by the colour model of Figure 5.7. Hering also observed the effect of afterimages to prove the concept of opponent colour vision. An afterimage is an optical illusion resulting in an image continuing to appear briefly after exposure to the actual image has ended. For example, if you stare at

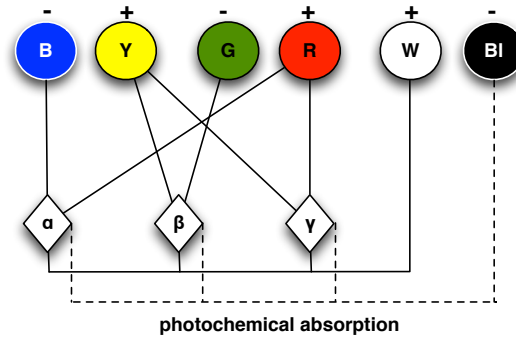


Figure 5.7: A simplified schematic of the Hering based opponent colour model for human vision proposed by Hurvich and Jameson [62]. The polarities indicate two possible modes of responses that are physiologically opponent in nature. The three pairs B-Y, G-R, and W-Bl are antagonistic (i.e. the concept of red-green is foreign to the human visual system, while yellow-red is a well perceived colour - orange). Here, α , β , γ are photosensitive biological retina photopigments aiding the formation of colour.

the dot in the centre of the pattern shown in Figure 5.8 for about one minute, then switch your gaze immediately to a homogenous sheet of white paper, you will see the same pattern but with the opponent colour pairs appearing where the original colours were (i.e. the red square appears green, the blue square appears yellow, the yellow square appears blue and the green square appears red). Hurvich and Jameson developed Hering's opponent colour theory further and were the first to propose an experimental method on test subjects (observers) for quantitatively measuring the opponent process response [67] using hue cancellation to produce the opponent process curves shown in Figure 5.9. Specifically, for each observer, there is a wavelength for which each yellow, green and blue hue appear unique (the single non-mixed hue) - this is shown by the arrows in Figure 5.9. To obtain unique red is a harder process involving selecting a long wavelength (such as 700nm) and mixing it with a bit of blue to cancel out the yellow also present at this wavelength. This process is known as hue cancellation [67]. Thanks to this pioneering experimental work in neuroscience, the opponent colour model is widely accepted as an explanation for colour vision in humans. We draw inspiration from these results when formulating situational awareness for reading roadwork scenes.

In this thesis, we limit our discussion of the opponent colour model to its effect on the design and manufacture of roadwork objects and how this knowledge can be exploited when formulating a roadwork scene signature. Figure 5.7 shows neuron colour responses of the visual cortex which are grouped into three antagonistic processes: yellow-blue (Y-B), red-green (R-G), and white-black (W-Bl) intensity. These responses are linked to three photochemical substances isolated in individual receptor units (denoted by α , β , γ). Hurvich and Jameson [62] showed that each opponent colour channel can be derived according to:

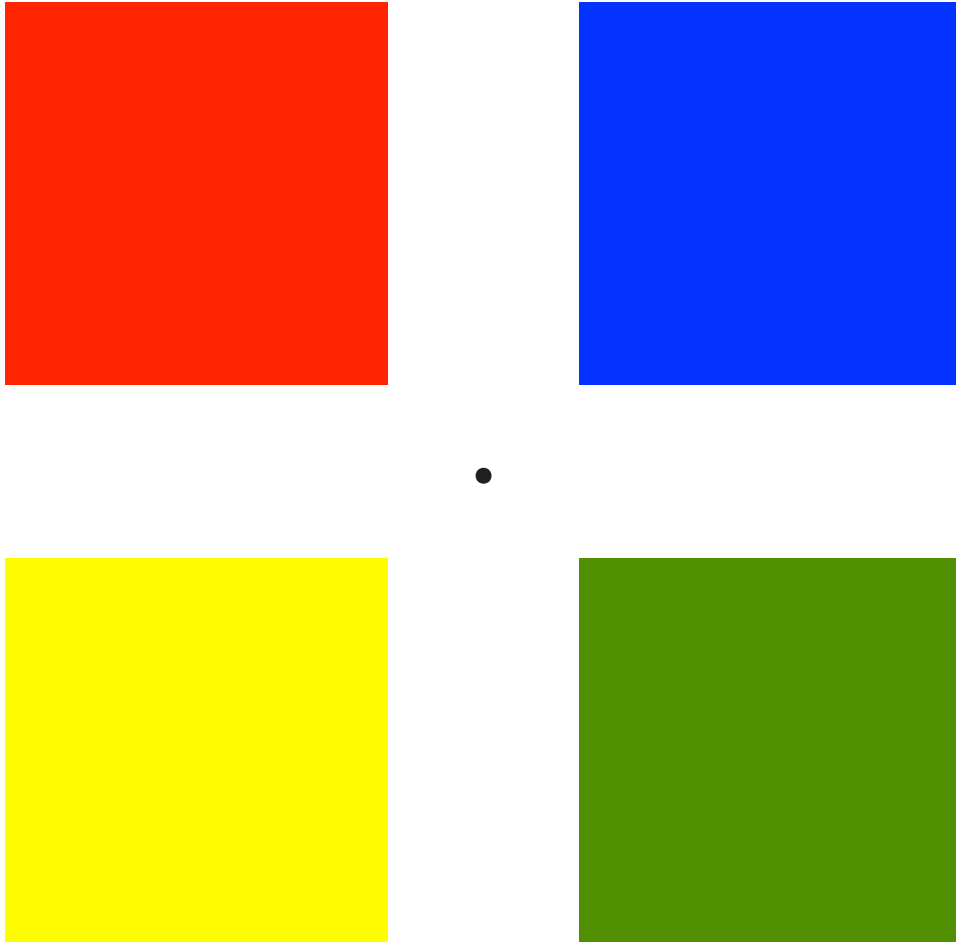


Figure 5.8: The afterimage effect which can only be fully explained by the opponent colour process in human vision: if you stare at the dot in the middle of the coloured pattern for about one minute and then quickly switch your gaze to a homogenous sheet of white paper, you will see the opponent colour pairs appearing in place of the squares in the afterimage.

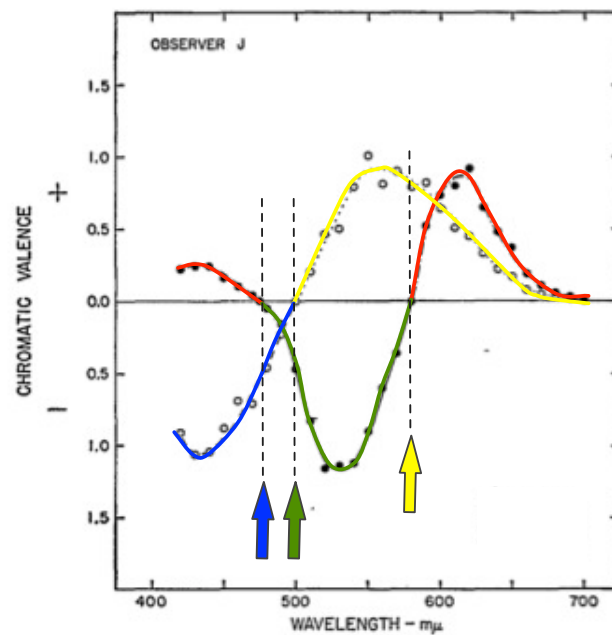


Figure 5.9: The opponent chromatic response functions measured by Hurvich and Jameson in [67], to derive quantitative measures for opponent colour vision in humans. For this observer, the yellow, green and blue hues appear unique (at the arrows). Deriving the unique red hue at 700nm requires a mixture with blue to cancel out the yellow there (hue cancellation). The chromatic valence polarities in the figure illustrate the opposing nature of this colour space and are arbitrarily assigned. Image adapted from [67].



Figure 5.10: Typical roadwork scenes found in our datasets. The dominant co-occurrences of red-green and blue-yellow hues in the roadwork scenes inspire our opponent colour vision formulation of a roadwork scene signature.

$$\begin{aligned}
 C_{y-b} &= k_1(\beta + \gamma - 2\alpha) \\
 C_{r-g} &= k_2(\alpha + \gamma - 2\beta) \\
 C_{w-bl} &= k_3(\alpha + \gamma + \beta) - k_4(\alpha + \beta + \gamma)
 \end{aligned} \tag{5.1}$$

where C_{x-y} represents an opponent colour channel and each k_i is a physiological constant.

In practice, the opponent colour model implies that certain combinations of colours cannot be perceived together. For example, blue-green or yellow-red describe well known colours cyan and orange. The concept of blue-yellow or red-green is however foreign to the human perception system. In Figure 5.10, we observe red-green roadwork barriers, blue traffic signs, bright orange traffic cones and barriers, and yellow coloured roadwork barriers. Roadworks are therefore designed to be easily perceived by the opponent colour process of the human visual system.

5.3.2 Roadwork Scenes and the Opponent Colour Model

Different combinations of roadwork objects, shown in Figure 5.3, such as construction barriers, traffic cones, pylons, keep left/right signs and roadwork signs can be found at roadwork sites. By studying a large amount of roadwork scenes (see Figure 5.10 for examples) typically traversed by our autonomous vehicle, an interesting and important observation is made: roadwork objects are predominantly coloured red, green, yellow or blue (or some closely related hue). In other words, roadworks are purposely designed to be salient to the human visual processing system, which perceives colour in an opponent-like fashion. This observation underpins our formulation of a roadwork scene signature.

For our application, we seek an equivalent computer vision colour model that interprets red-green and blue-yellow antagonistically, and therefore perceives roadworks the same way a human would as discussed in

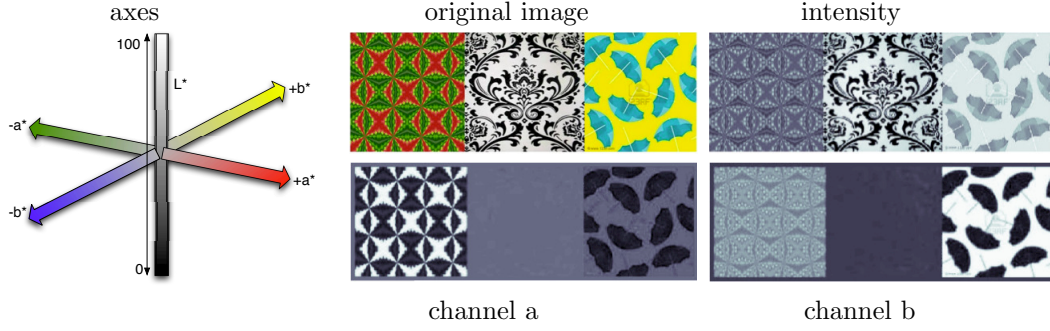


Figure 5.11: Left: The axes of the CIELab opponent colour space. Right: Automatic pixel labelling characteristics: the red-green pattern is highlighted by channel a (bottom left), with the opponent colour intensities at opposite extremes. Similarly, the blue-yellow pattern is highlighted by channel b (bottom right). The black-white pattern is highlighted by the intensity (achromatic) channel.

Section 5.3.1. In computer vision, the CIELab colour space is an opponent space that models perceptually uniform human vision by approximating the logarithmic response functions of the eye. For a given set of CIE XYZ tristimulus values with normalised reference white point (X_n, Y_n, Z_n) , the CIELab colour transform is given by:

$$\begin{aligned}
 L &= 116f(Y/Y_n) - 16 \\
 a &= 500[f(X/X_n) - f(Y/Y_n)] \\
 b &= 200[f(Y/Y_n) - f(Z/Z_n)]
 \end{aligned}$$

Where $f(t) = t^{\frac{1}{3}}$ if $t > \tau$ and $f(t) = vt + \frac{4}{29}$ otherwise, for CIE constants τ and v .

As in human vision, the CIELab colour space has an achromatic channel (L) and two chromatic red-green (a), and blue-yellow (b) colour channels. Perceptual uniformity implies that changes to any of the Lab components induces similar changes in the *perceived colour*. Converting from classic RGB to CIELab involves a nonlinear transform. This opponent colour model is useful for pattern recognition because it naturally segments patterns into perceptually meaningful opposing colour channels. This effect is illustrated by the righthand Figure 5.11 where the red-green and yellow-blue patterns are highlighted by the opponent colour transformation. This automatic pixel labelling characteristic is computationally efficient and a natural property of the colour model. Importantly, the CIELab colour space is consistent with the Hering colour model used to explain human vision and each channel encodes exactly two colours in an antagonistic fashion.

5.4 An Opponent Colour Roadwork Scene Signature

A roadwork scene signature is a compact image representation useful for characterising a road scene. We formulate this as a global signature based on the opponent colour model described in the previous section.

5.4.1 Opponent Colour Spatial Prior

In Section 5.3.2, we showed the relationship between colour vision in humans and the design of roadwork objects. Furthermore, we discussed why the CIELab opponent colour space is a good approximation of human opponent vision for the task of roadwork scene recognition. A good roadwork scene signature should be discriminative and robust to scene clutter, which motivates the first step of the signature creation process: transforming the image to CIELab colour space. This transformation allows us to exploit the automatic pixel labelling properties of the opponent colour space in order to implicitly derive a spatial prior for the location of roadwork objects in a given image. Only the chromatic CIELab colour channels (a and b) are considered in this formulation. Figure 5.11 shows that, for each of the chromatic channels, the opponent colours are at opposite intensity extremes (yellow is positive and blue is negative, similarly red is positive and green is negative). This implies that each CIELab chromatic channel reduces to a pixel wise spatial prior for the observation of red, green, yellow and blue hues of roadwork objects which is clearly illustrated by Figures 5.11 and 5.12. The peak gradient information is therefore concentrated at these opponent extreme regions (very high/low channel intensity regions) and an image keypoint detector peaks at such regions. We use this fact to efficiently extract descriptors only at relevant image locations. It is this computationally efficient spatial prior which allows us to avoid naively detecting individual roadwork objects in a given scene. Our formulation stands in stark contrast to traditional approaches where many keypoints are desired per image (and often densely sampled) and blocks are used to capture spatial information [154] resulting in more complex image representations.

5.4.2 Image Descriptor Extraction

Following the opponent based spatial prior described in the previous section, we extract a descriptor at each detected image keypoint. For a given colour channel and detected keypoint, a 128 bit SURF descriptor [6] is extracted as shown in Figure 5.13. We choose the SURF descriptor for speed and robustness [6]. Other descriptors such as RGB histograms, SIFT [88], colour moments or moment invariants are also valid possible candidates for our proposed framework, however a comparison of different descriptors is beyond



Figure 5.12: Results from our opponent colour spatial prior showing the automatic labelling of pixels in CIE Lab channels a and b. The intensity scale is shown at the bottom of the figure. Notice how blue traffic signs, and barriers/traffic cones are well segmented in channels b and a respectively.

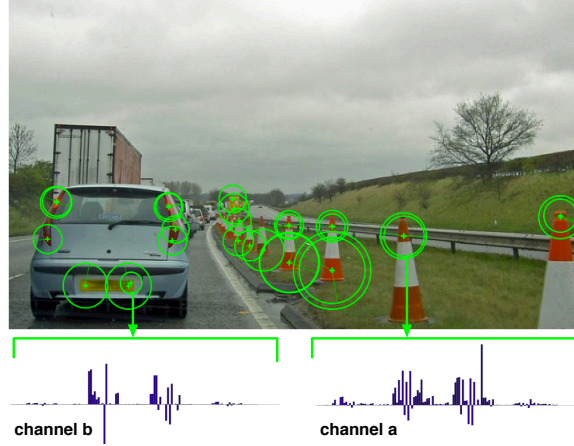


Figure 5.13: Our image descriptor comprises of a 128 bit SURF descriptor [6] where the scale and orientation of the SURF descriptor is denoted by the green circle. A descriptor is computed at each keypoint in the image for both opponent chromatic colour channels, a and b.

the intended scope of this work, and we refer the reader to [139] for a good comparison.

Our proposed descriptor differs from the Bag-of-Colors model [143] even though both methods exploit the CIELab colour space. Firstly, SURF keypoints are only detected at points most likely to contain roadworks due to our initial spatial prior. In contrast, the Bag-of-Colors approach creates a colour descriptor by considering every pixel in the image to increment a k_c -dimensional vector, where k_c is the colour codebook length. Secondly, in addition to colour, our descriptor also accounts for texture information.

5.4.3 Codebook Generation and Vectorization

For a given image, the number of extracted descriptors depends on the number of detected keypoints. This number varies with each image, thus a method to represent every image as a single vector is required. To achieve this, a codebook is generated and each image is vector quantised against this codebook in order to derive a single vector representation. Our codebook generation and vectorization procedure is an improved version of the Bag-of-Words approach for scene recognition [36, 154] and is summarised by Figure 5.14. Codebook generation therefore proceeds as follows:

1. Collect training images containing different road (non-roadwork) and roadwork scenes
2. Create a codebook $C = \{c_1, c_2, \dots, c_k\}$: for all the training images extract image descriptors as described in Section 5.4.2. K-means cluster these descriptors to create a codebook vocabulary, choosing k experimentally based on precision recall performance on a holdout dataset. Each cluster

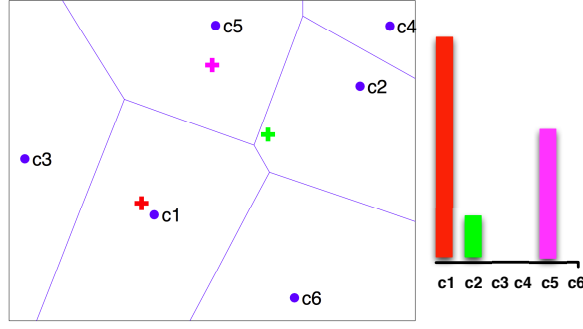


Figure 5.14: An illustration of our soft assignment vectorization for the simple case of a codebook $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ with six visual word centroids. Three descriptors (shown as crosses) were detected in the image. The histogram on the right shows the final vectorized image representation. Notice how the descriptor with the shortest distance from the centroid receives a higher weighting in the vectorized histogram representation (see Equation 5.3). This allows us to capture the distribution of the descriptors with respect to the nearest neighbour centroid (or visual word) rather than incrementing the histogram with the same value for each nearest neighbour regardless of similarity.

centroid, c_i then represents a visual word in this codebook.

Given a codebook, we can now vectorize a new image by quantising it against the codebook. The Bag-of-Words model represents each image as a histogram vector of length C . Each histogram bin is incremented by one when a detected descriptor is closest to its associated visual word centroid. This is an example of a hard assignment of descriptors to their nearest neighbour visual word. Thus for a given descriptor x , and nearest neighbour $NN(x) = c_i$ the i^{th} histogram bin h_i is updated according to:

$$h_i = h_i + 1 \text{ if } NN(x) = c_i \quad (5.2)$$

In our work, we adopt a soft assignment approach instead. Here, instead of updating the histogram by one, we update it with the inverse standard deviation from the closest visual word. Define a set N_i such that $x \in N_i$, iff $NN(x) = c_i$ then:

$$h_i = h_i + \left[\sqrt{\frac{1}{|N_i|} \sum_{x_m \in N_i} (x_m - c_{im})^2} + \varepsilon \right]^{-1} \quad (5.3)$$

where $m = \{1, 2, \dots, k\}$ indexes the elements of a visual word (or descriptor) and ε is a small number ensuring that the denominator is nonzero if the descriptor and codebook visual word match exactly. This soft assignment approach captures the degree of similarity between a descriptor and its closest visual word

and indirectly models the feature distribution (see Figure 5.14 for an example). We can adopt this approach because the perceptually uniform CIELab colour model represents a non-linear transformation into a new space that is better able to predict visual colour difference between two points using the Euclidean distance metric [146]. Furthermore, van Gemert et al. showed that explicitly modelling visual word assignment decreases classification ambiguity when compared to the traditional hard assignment model [140].

Many soft assignment schemes exist, however our approach is inspired by work done by Jegou et al. [68] to aggregate local descriptors into a single image representation for image search: a difference between a descriptor and visual word is calculated then dimensionality reduction is performed to represent the image using a single vector. Other methods such as probabilistic visual word voting where each image descriptor feature contributes to multiple codebook visual word assignments, based on the probability of the image given the visual word, have also been proposed [111]. Here the authors assigned weights proportional to $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ where d is the distance to the nearest visual word centroid and the model parameters are the spatial scale σ and number of neighbours to consider, r .

The key advantage of our formulation is that the opponent colour based spatial prior directs our descriptor extraction algorithm to consider only relevant regions in the image. These are the regions most likely to contain roadworks because hues not encoded by the opponent colour channels are suppressed. This phenomenon is demonstrated by Figure 5.15 which shows visual words detected with and without our spatial prior superimposed on the original image. Our roadwork signature, h therefore comprises of fewer but highly relevant visual words resulting in a compact discriminative vectorized image representation.

5.4.4 Classifier Training and Validation

The codebook generation and vectorization scheme described in Section 5.4.3 results in a roadwork scene signature (feature vector) for a given image. This signature is a compact vector representation of length C . Given a set of training images (containing non-roadwork and roadwork scenes) we extract roadwork scene signatures for all images and use these input features to train a classifier for roadwork scene recognition. Our classifier choice is influenced by the fact that, although roadworks are common [33], they are finite and occupy limited sections of the road. This results in a skewed number of examples for roadwork versus non-roadwork classes. The RUSBoost algorithm [122] handles class imbalance by combining Random Under Sampling (RUS) with AdaBoost [41] resulting in an improved model of the minority class by removing majority class examples. Details of this algorithm, with tree learners, was discussed in Section 3.4.3.2. We train a RUSBoost classifier using tree learners and 500 rounds within a five-fold cross validation scheme.

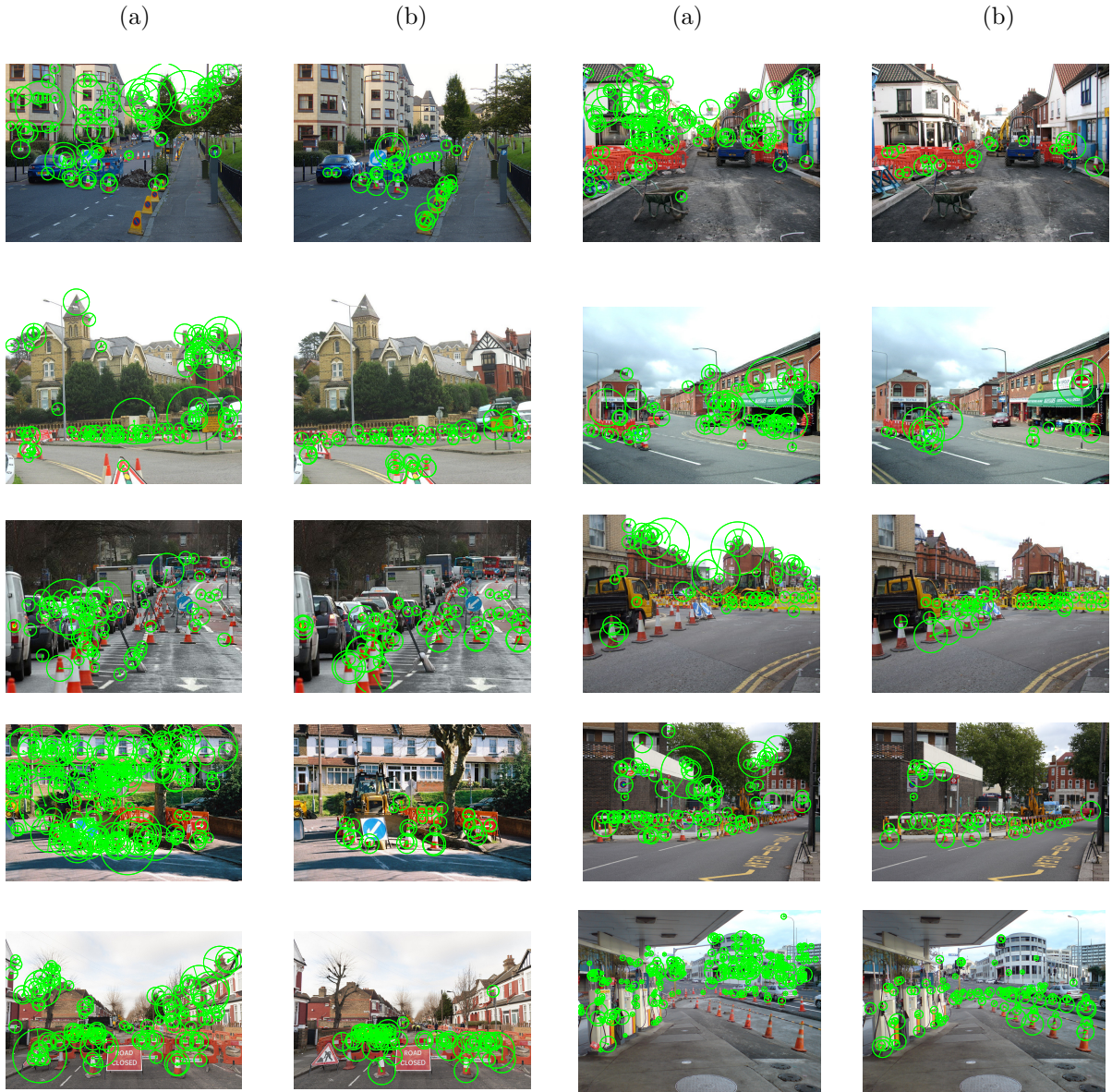


Figure 5.15: Examples of visual words (detected SURF descriptors from both opponent colour channels) superimposed on roadwork scenes found in our datasets. The SURF threshold remains constant throughout. (a) Visual words extracted without exploiting the spatial prior of the opponent colour model. (b) Visual words extracted using our opponent colour based formulation. Notice how significantly fewer visual words are extracted by incorporating the opponent colour model and importantly, these visual words are only extracted in regions most likely to contain roadwork objects.

	Source	Description	# Pos Examples	# Neg Examples
Train	Geograph	Codebook	536	1002
	Geograph	Classifier	1032	2042
Test	Town	Dataset 1	57	89
	Residential	Dataset 2	60	139
	Motorway	Dataset 3	71	28
	City	Dataset 4	1443	2075
	Motorway + city	Dataset 5	1195	704

Table 5.1: A summary of the datasets used for training and evaluation.

5.5 Performance Evaluation

5.5.1 Data Sets

Our roadwork scene signature was evaluated on data collected over several months using our autonomous vehicle. The dataset contains 10473 images of roadwork and road scenes with ground truth annotations. The training data was sourced online from the Geograph Project database for Britain and Ireland [45] containing images submitted by the public. Table 5.1 summarises our roadwork dataset.

5.5.2 Results and Comparisons

To evaluate the effectiveness of our spatial prior, we compare the number of SURF keypoints extracted in 100 images with and without the opponent spatial prior. From Figure 5.16, it is clear that the number of SURF keypoints is significantly reduced when using our opponent colour spatial prior formulation without compromising on accuracy, since features are extracted only at relevant locations (see Figure 5.15 for examples). Furthermore, Figure 5.17 shows that, for our opponent colour based formulation, the number of SURF keypoints is in general smaller for images not containing roadworks which confirms our intuition about the visual saliency of roadworks in images. In other words, roadworks are designed to be detected by the human visual system which operates in an opponent like fashion (see Section 5.3.2 for a discussion). To understand this further, we computed the Bayes optimal threshold for minimising the misclassification error based on the number of SURF keypoints detected in a given image. A misclassification rate of 30% was achieved with a simple feature count threshold classifier.

An important design consideration for our roadwork scene signature is how to select an appropriate codebook size, k . Care should be taken not to overfit the data whilst ensuring that a representative and discriminative number of visual words is used. We experimentally determined the value of k by computing the precision and recall on a holdout dataset for various values of k , and found $k = 100$ to give the best

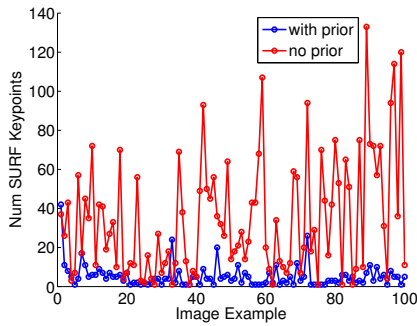


Figure 5.16: The number of SURF keypoints detected without using the opponent colour spatial prior is significantly higher than when the transform is applied. Although fewer descriptors are extracted by our opponent colour framework, these are located at relevant regions in the image, thus accuracy is not compromised (see Figure 5.15).

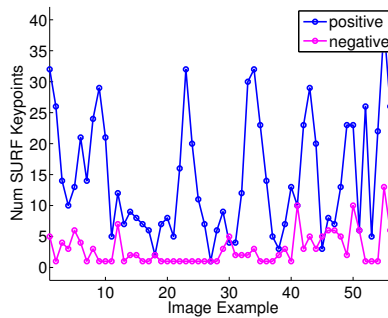


Figure 5.17: In general, fewer keypoints are detected for negative examples which demonstrates the visual saliency of roadworks i.e. our descriptor focuses on roadwork objects highlighted by our spatial prior, while suppressing non-opponent noise.

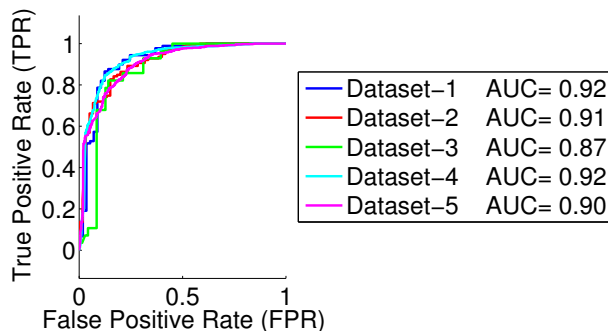


Figure 5.18: ROC curves showing the TPR and FPR achieved for the datasets described in Table 5.1. Each curve represents a different dataset collected using our autonomous vehicle. The Area Under Curve (AUC) for each dataset is also shown. Good discriminative performance is achieved across all datasets.

performance. Our codebook size is indeed very small, which is again a property of the opponent colour spatial prior discussed in Section 5.4.1 and demonstrated by Figure 5.12.

We apply our roadwork scene signature to the complex task of roadwork scene recognition. We also deliberately trained and tested on completely different datasets (online sourced training set and test set from our autonomous vehicle) in order to test the generalisation and robustness of our system. For the test datasets collected by our autonomous vehicle, we compute Receiver Operating Characteristic (ROC curves) to illustrate the performance of our classifier with varying RUSBoost discrimination threshold. For ROC curves, peak performance occurs at the top left hand corner of the graph where the False Positive Rate (FPR) is zero and the True Positive Rate (TPR) is one. The area under the curve is known as Area Under Curve (AUC) and can be interpreted as the probability that the classifier will assign a higher score to a randomly chosen positive example than to a randomly chosen negative example. This metric captures the accuracy of the classifier, thus $AUC = 1$ for a perfect classifier and $AUC \leq 0.5$ for a worthless classifier. Figure 5.18 shows ROC performance results (with AUC values) using our opponent colour formulation for the roadwork scene signature. The ROC curves show good performance illustrating the discriminative property of our roadwork scene signature.

To test the effectiveness of our opponent colour spatial prior, we conducted experiments without using the opponent colour spatial prior (keeping everything else the same). Considerable effort was made to give the comparison algorithm the best possible chance by tuning the SURF threshold and selecting the best

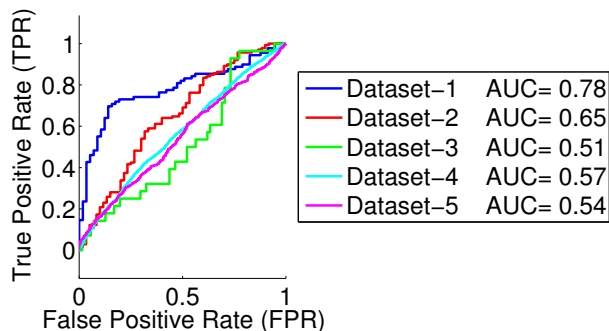


Figure 5.19: ROC curves showing the TPR and FPR achieved for the datasets described in Table 5.1 without the opponent spatial prior. Each curve corresponds to Figure 5.18. The Area Under Curve (AUC) for each dataset is also shown. The performance is significantly worse without the opponent colour prior.

codebook size, $k = 1000$ based on precision and recall performance on a holdout dataset. Notice how the codebook size is ten times larger than when the opponent colour spatial prior is used - this shows that our opponent colour roadwork signature is more compact whilst achieving better performance in terms of accuracy. Figure 5.19 shows the ROC results for the comparison experiment which demonstrate poor performance across all datasets (notice the low AUC values) compared to our opponent colour formulation (Figure 5.18). This demonstrates the effectiveness of using our roadwork scene signature based on the opponent colour model, which achieves between 14% - 36% improved performance across all datasets compared to the traditional approach.

Figure 5.4 shows examples of successful roadwork detections across all datasets using our roadwork scene signature and the associated user interface in our autonomous vehicle. To the best of our knowledge, there is no similar roadwork scene recognition work to compare against, however we implement a naive alternative to our method based on object recognition for comparison, similar to work done by Gump et al. [50] who proposed the joint detection of traffic beacons and yellow temporary road markings in order to classify construction sites. To do this, we train traffic cone and keep right sign classifiers (see Figure 5.20) using dictionary template matching features [135] trained within a GentleBoost framework [42].

GentleBoost (or Gentle Adaboost) is a binary classification framework first proposed by Friedman et al. [42]. Here, every weak learner fits a regression model to response values $y_n \in \{-1, +1\}$ and the exponential loss is minimised. Specifically, the mean squared error is given by:

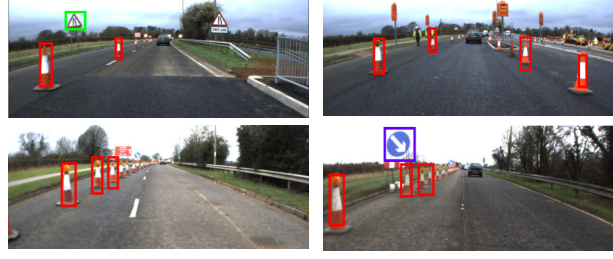


Figure 5.20: Some examples of traffic cone and keep right sign detector outputs.

$$\sum_{n=1}^N d_n^{(t)} (\tilde{y}_n - h_t(x_n))^2 \quad (5.4)$$

where t indexes the learners, $d_n^{(t)}$ is the normalised (sums to one) observation weight at step t and $h_t(x_n)$ represent the predictions of the regression model h_t fitted to the response values y_n . x_n is the vector of predictor values for the n^{th} observation. The weighted mean squared error of Equation 5.4 approaches 1 as the strength of individual learners weakens.

GentleBoost is an improvement on AdaBoost [41], a sequential algorithm which computes the weighted classification error given by:

$$\sum_{n=1}^N d_n^{(t)} \mathbb{I}(y_n \neq h_t(x_n)) \quad (5.5)$$

where y_n is the true class label, h_t is the predictor (hypothesis) commonly decision stumps, $\mathbb{I}$ is the indicator function and $d_n^{(t)}$ is the observation weight at step t . AdaBoost works by increasing the weights of observations (examples) misclassified by learner t (and reduces weights of correctly classified “easy” observations). These updated weights $d_n^{(t+1)}$ are used to train the next $(t+1)$ th learner. A new example is classified using $f(x) = \sum_{t=1}^T \alpha_t h_t(x)$ where $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$ where ϵ_t is given by Equation 5.5 [41].

Following classifier training, an image is then classified based on the presence of roadwork objects in the scene. For this comparison, we assume that the occurrence of traffic cones in a given scene is conditionally independent of the presence of keep right traffic signs given the class label. Each image is therefore represented by Naive Bayes:

$$p(C|f_1, f_2) = \frac{1}{Z} p(C) p(f_1, f_2|C) \quad (5.6)$$

where the features, f_1 and f_2 denote the presence of traffic cones and keep right signs respectively, Z is the normalisation constant, and classes C are roadwork and non-roadwork. We use a maximum likelihood

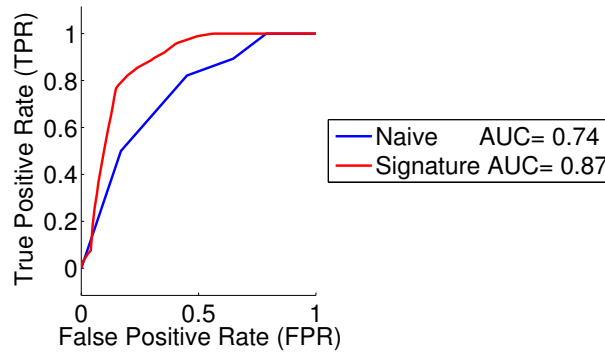


Figure 5.21: A comparison showing ROC performance using our roadwork scene signature compared to roadwork object detection within a Naive Bayes scene recognition framework. Our method outperforms the naive approach and has the added advantage of not explicitly modelling individual roadwork objects in the scene. Dataset 3 was used.

estimate of the parameters. Results of our comparison are shown in Figure 5.21 demonstrating that our roadwork scene signature outperforms the object detection approach. The reasons for this is twofold: firstly, the presence of multiple traffic cones (or keep right signs) does not necessarily imply a roadwork scene since keep right signs are not unique to roadwork scenes and traffic cones can appear for a number of reasons such as marking off where children are playing or advance warning of hazards. Secondly, a roadwork scene recognition system based on individual detectors is sensitive to detector performance even if multiple roadwork object detectors are trained. Contrastingly, our proposed roadwork scene signature does not require the detection of individual roadwork objects in the scene and is hence faster to train, apply to an image and has the added advantage of being more accurate.

Our current Naive Bayes Classifier implementation considers the number of traffic cones and keep right signs, in an image, as features. A potential variant approach could include a binary feature vector capturing the presence of a single traffic cone, more than one traffic cone, a single keep right sign, and more than one keep right sign. Such a feature vector is useful for discriminating between scenes containing roadworks and ones where a random cone/sign appears on the road. Also, a feature computing the average pairwise distances between the detected objects could be added to capture some spatial relationships between the detected roadwork objects.

In this Chapter we presented a roadwork scene signature based on an opponent colour spatial prior. This spatial prior has two main advantages: computational efficiency and increased accuracy, because

although fewer SURF features are detected, these are detected at relevant locations in the image (see Figure 5.15). One alternative approach to creating a spatial prior capturing the most likely locations of roadworks in an image, is to use the GentleBoost roadwork object classifier described above to select regions of interest (the bounding boxes shown in Figure 5.20). SURF features could then be computed within these regions of interest. This approach has the advantage of providing further useful semantic information regarding the type of roadwork objects present in an image (motorway roadworks tend to only be delineated by traffic cones and signs whilst residential and city roadworks tend to be delineated by barriers also). Three main disadvantages of this alternative approach exist: Firstly, the opponent colour spatial prior (an efficient colour transform) is replaced by a more complex object detection stage which is computationally expensive to run and requires training a multi-class object detector. Secondly, the prediction accuracy of the detectors greatly affects the regions of interest identified in the image - false detections result in incorrect regions of interest. Thirdly, the Bag-of-Words model creates a distribution over the SURF features across all image classes (roadworks and non-roadworks) thus an object detector based region of interest operator must rely on false detections to capture non-roadwork regions.

5.6 Chapter Summary

In this Chapter, we presented a framework for semantic situational awareness in dynamic roadway conditions. Specifically, we demonstrated that roadworks are purposely engineered to be salient to the human opponent colour visual system. We used this fact to formulate a spatial prior for the location of roadwork objects in a given image, based on the CIELab opponent colour space which approximates human vision. We then derived a unique roadwork scene signature, a global image representation that does not explicitly model individual roadwork objects in a given scene. We demonstrate the usefulness of this roadwork signature for the task of roadwork scene recognition for an autonomous vehicle - resulting in semantic situational scene awareness. Our system is evaluated on real data from an autonomous vehicle and is shown to perform well across multiple datasets containing varied types of roadwork images. We also show that our method outperforms an alternative roadwork object detection approach.

Our Situational Awareness data structure of Chapter 2 now contains all the relevant semantic information described in the original message type template: road marking classes and road scene situations, road boundaries and boundary semantics, and roadwork types and scene recognition scores. In the next Chapter, we discuss some implementation issues for our Google protobuf Situational Awareness data structure, which summarises the contributions of this thesis.

Chapter 6

Closure: Situational Awareness

At the start of this thesis, in Chapter 2, we introduced a data type representation for Situational Awareness based on Google Protocol Buffers. Although in theory, any appropriate data structure could be used, we continue this discussion focusing on Google Protocol Buffers as an example.

Specifically, we introduced the hierarchical “SituationalAwareness” protobuf message type which contained “Road”, “Boundaries” and “Works” messages. The core of this thesis, Chapters 3-5 provided details of the algorithms we formulated for extracting the semantic information that is stored in this Situational Awareness data structure. The intended application of such a data structure is to enable the easy storage and retrieval of relevant road semantics that capture and interpret the road scene, thus enabling a vehicle to truly *read the road* as it drives along a path.

In this Chapter we return to complete our discussion of the Google Protocol Buffer data structure for Situational Awareness templated in Chapter 2. Specifically, we discuss exactly what output, from our algorithms is stored in these buffers and how to read/write data to them.

6.1 Reading and Writing Data

Google Protocol Buffers [48] simplify the generation of data access classes given the `.proto` file defining the protobuf message types. The SituationalAwareness protocol buffer message type in Chapter 2 can be compiled to produce a main class named “SituationalAwareness”. As an example, to populate the data structure with semantics about a zig-zag road marking, in good condition, at a pedestrian approach this pseudo code could be used:

```

SituationalAwareness situation;
situation.set_position = [-1.307,51.818];
situation.set_orientation = 0.912;
situation.Road road;
road.Roadmarkings.set_condition = 1;
road.Roadmarkings.set_type = ZIG;
road.Roadmarkings.set_location = [0.53,0.89];
road.Situations.set_sit = PEDESTRIAN_APPROACH;

```

Similarly, to retrieve this message when the vehicle is at a known location, the following pseudo code could be used:

```

SituationalAwareness situation;
situation.Road road;
cout << "Road Marking Type: " << road.Roadmarkings.type() << endl;
cout << "Situation Type: " << road.Situations.sit() << endl;

```

6.2 Message Type Implementation

In this Section we discuss which outputs of our algorithms constitute the values of the fields in the protobuf messages.

6.2.1 Road Markings

The semantics associated with road markings were presented in Chapter 3 which described a probabilistic framework for road marking classification. We considered jointly classifying seven distinct classes of road markings, namely: single boundary, double boundary, separator, zig-zag, intersection, boxed junction and special lane. The classification output from our Conditional Random Field is captured by the *RoadmarkingType* field of the protobuf data structure. Similarly, the *SituationType* field in the protobuf message type is derived from our road scene semantics classification step to detect traffic light controlled pedestrian crossings (*PEDESTRIAN_APPROACH*) and boxed junctions (*NO_DRIVE*).

6.2.2 Road Boundaries

Given a vertically downward facing 2D SICK LIDAR, we derived a BaryArc model based on Highway Traffic Engineering principles in Chapter 4. In the protobuf message type, the *Intersection* message is derived from OpenStreetMap [107] interpolated locations for the intersections of ways. The data association of the laser points (*BoundaryType*) can automatically be derived from our road boundary extraction algorithm which fits the left and right boundaries separately. The *param* protobuf field contains the BaryArc anchor points and parameter w_1^2 which are necessary for the reconstruction of a BaryArc road boundary. Finally, the derived boundary semantics, namely the continuity and curvature, are stored in the *centerline* and *curvature* fields respectively.

6.2.3 Roadworks

In Chapter 5, we developed a novel roadwork scene recognition framework based on the opponent colour model for human vision. Specifically, we tested the algorithm on data from our autonomous vehicle which traversed urban, city and motorway roadworks. The *RoadworkType* field of the protobuf message type can be inferred from the GPS location of the vehicle and the topology associated with the surrounding area (urban, city, or motorway). The protobuf *score* field is the output of the GentleBoost classifier of our roadwork scene recognition framework.

The next and final Chapter of this thesis is a discussion summarising our contributions and future work.

Chapter 7

Discussion

This Chapter summarises the main contributions of this thesis and discusses possibilities for future research.

7.1 Summary of Contributions

This thesis has made the following contributions to the state of the art in situational awareness for autonomous vehicles:

1. **A data structure for efficiently representing Situational Awareness in autonomous vehicles (Chapters 2 and 6).** We introduced the idea of a semantic data structure for reading the road surface and reading the road scene. This data structure can easily be written to and read from in real-time in order to recall previously stored semantic information about the road ahead.
2. **A probabilistic framework for the joint classification of road markings (Chapter 3).** We introduced a framework for classifying seven distinct classes of road markings: single boundary, double boundary, separator, zig-zag, intersection, boxed junction, and special lane. We also inferred pedestrian approaches and no-drive road scene semantics based on the classified road markings.
3. **A principled method for detecting road boundary points in 2D laser point clouds (Chapter 4).** The method exploits our unique vertically downward facing push-broom laser sensor configuration to develop novel simple, yet elegant, algorithms for road surface segmentation and road boundary point extraction.
4. **An implicit barycentric model, the BaryArc, for road boundary estimation based on**

Highway Traffic Engineering geometric road design principles (Chapter 4). We evaluated the accuracy and suitability of our BaryArc model for road boundary curve fitting by measuring the Mean Squared Error (least squares), curvature, and continuity of the estimated road boundaries.

5. **A biologically inspired roadwork scene recognition system based on opponent colour vision in humans (Chapter 5).** We showed that roadworks are purposely engineered for human visual saliency and that detecting them using a computer vision approximation to opponent colour vision in humans (CIELab colour space), is efficient and effective. Our framework does not require the detection of individual roadwork objects in the scene but instead exploits an opponent spatial prior to capture the gist of a scene. Roadwork scene recognition is an under researched area but crucial for true autonomy and situational awareness in vehicles.
6. **Benchmarking datasets for road marking classification, boundary estimation and roadwork scene recognition.** The road marking dataset contains 2068 images with 13099 unique hand labeled annotations of over seven distinct classes of road markings. The road boundary dataset contains 32km of road data. The roadwork scene recognition datasets contains 10473 images of roadwork and road scenes with ground truth annotations . All three benchmarking datasets comprise of city, residential and motorway roads.

In summary, this thesis has made contributions to situational awareness in autonomous vehicles by capturing semantics derived from both *reading the road surface* as well as *reading the road scene*. Specifically, we considered semantics of the road itself (road markings and boundaries) and semantics of objects on the road (roadworks). Thus a vehicle can read *and* interpret the semantics of the road ahead.

7.2 Future Work

Many possibilities for future work exist and some interesting examples are listed here:

1. **Learning the drivable regions of the road based on the classification of present road markings.** Road markings encode the rules of the road and define where a vehicle may safely drive. Thus road marking classification outputs could be used to learn a model to intelligently segment the road surface into drivable and non-drivable regions.
2. **Semantic road scene interpretation based on classified road markings.** Using the classified road markings, learn a model for the distribution of road markings in road scenes. This model could

be used to classify roads into different road scene semantics such as motorway, crossing etc.

3. **Model driven road boundary feature extraction.** Two possible avenues for model-based feature extraction exist: (a) a model informing the expected locations of future feature points (by, for example, using a Bayes predictive model such as the Kalman filter) or (b) the features being independently extracted followed by fitting a geometric model to the feature points to discard noisy points. We adopt the latter approach in our work. Future work could consider, given a set of initial points, estimating the expected feature point locations of road boundaries based on the BaryArc model and current local observations (estimated boundary point locations).
4. **Learning a roadwork spatial prior from opponent colour channels.** Currently, in Chapter 5, the two separate Red-Green and Blue-Yellow opponent colour channels are processed independently to provide spatial prior information for keypoint detection in each roadwork image. Alternatively, a combined roadwork spatial prior could be learnt capturing the general layout of roadworks in the scene.

7.3 Concluding Remarks

Situational awareness for autonomous vehicles has been studied for two decades, primarily focusing on road marking detection, road surface extraction, traffic sign detection, and driving lane recognition. Much work has focused on camera based techniques, sometimes combined with LIDAR. To frame our task, we offered a Situational Awareness data structure on which to motivate our work. This data structure captures road semantic information which we automatically extract. Specifically, we developed a probabilistic road marking joint classification framework, a road boundary estimation system for 2D LIDAR based on a barycentric BaryArc model, and an opponent colour vision roadwork scene signature useful for roadwork scene recognition.

We hope that our approach to reading the road surface and road scene for semantic situational awareness in autonomous vehicles will be useful for improving a vehicle’s ability to truly read, understand, and interpret the road ahead.

Appendix A

Geometric Design of Roads

A.1 Design Vehicle, Speed and Hourly Volume

When designing a road, a design vehicle, to represent all vehicles on the highway, is chosen such that the physical characteristics (weight, dimensions, operation) and geometric parameters (intersection and turning radii) of the road are worst case [44]. The corresponding design speed (miles/hour) is a selected speed used to determine the various geometric features of the roadway [44] and is influenced by the functional class of the roadway, topography for the area (level, rolling or mountainous terrain) and the land use of the adjacent area. Table A.1 shows design speeds for rural collector roads.

The Design Hourly Volume (DHV) is a measure of the projected hourly volume (vehicles per day) used in design and is often taken to be a percentage of the Average Daily Traffic (ADT) or as the average of the maximum afternoon peak hour volume for each week in the year [44].

	Design Speed given Design Volume		
Type of Terrain	0-400	400-2000	Over 2000
Level	40	50	60
Rolling	30	40	50
Mountainous	20	30	40

Table A.1: Minimum design speeds for rural collector roads. Reproduced from [44].

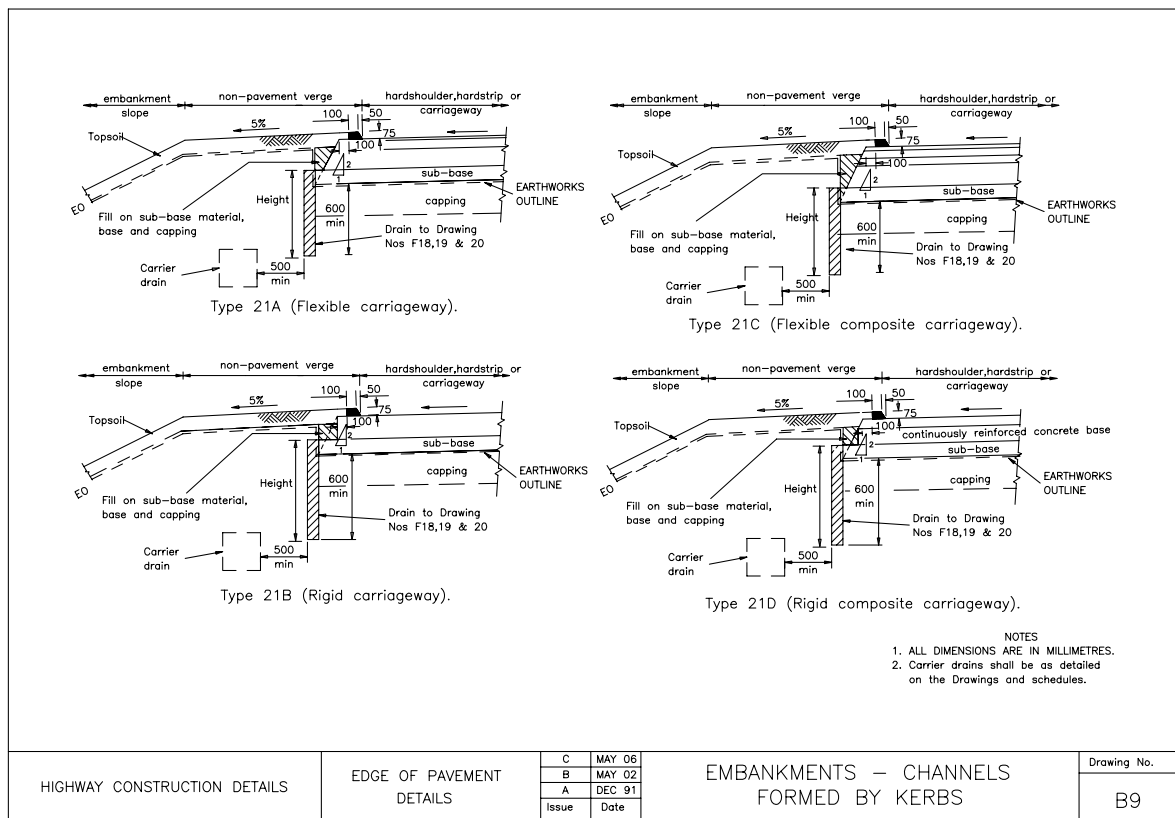


Figure A.1: Technical drawing showing the embankments (channels formed by curbs) taken from [3], Volume 3.

A.2 Cross-Sectional Elements

Highway engineers optimise the cost of additional lanes and the added performance gained, cross slope, superelevation and expected traffic volume, when choosing the number of lanes. The shoulder of a lane is continuous with the travel lane and provides a safe area in which vehicles may stop in emergencies, or bicycles may travel (if a bicycle lane exists). Furthermore, a vehicle should be at least 1 ft from the edge of the pavement when stopped, thus motorways typically have 10-12ft shoulder widths to support trucks. In practice however, topographical and low volume constraints result in minimum shoulder widths of just 2ft. In the case of a cycle lane, the minimum width should be 4ft. In order to facilitate adequate water drainage, shoulders tend to be sloped as shown in Figure 4.5.

The UK Department for Transport's Manual for Contract Documents for Highway Work (MCHW) [3] publishes technical drawings showing the construction of embankments resulting from the presence of curbs in Volume 3 (See Figure A.1 for an example).

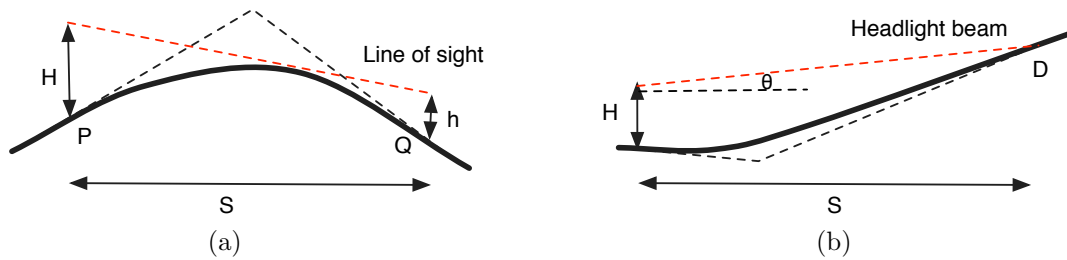


Figure A.2: (a) A vertical crest curve. The line of sight for the crest curve is shown in red, given a driver at P of height H and an object h located at point Q. (b) A vertical sag curve. The headlight beam is shown in red for a driver of height H . In practice, standard heights are used for H and care should be taken to design roads with adequate sight distance S , the projection of the line of sight or headlight beam onto the horizontal, to ensure safe travel.

The final cross-sectional design element worth discussing is the cross slope (cant or superelevation) of the roadway. The cross section of a cross slope is either curved or plane or a combination of the two and is generally modelled as a parabola. The cross slope at straight sections of the road is typically 1 – 2% [44] and on curved sections the outside edge of the road is super elevated (4 – 10%) above the centreline to allow gravity to assist with providing the necessary centripetal acceleration force to traverse the curved section. In icy conditions, low cross slopes could lead to loss of control whilst high cross slopes make driving slowly difficult. Cross slopes also ensure efficient draining of water from rain or melted snow. In practice, superelevation is often quoted as a rate and each road pavement either slopes in two ways or in one way only.

A.3 Vertical Alignment

The vertical alignment of a road can be summarised by Figure A.2, and consists of straight sections (tangents or grades) connected by vertical curves in the direction of travel. Vertical curves provide the gradual smooth transition from one tangent grade to another and are parameterised as either crest (concave up) or sag (concave down) parabolic curves. Vertical curves also dictate the minimum stopping distance (for crest curves) and the drainage, appearance and general ease of driving along the road. When selecting a vertical crest curve, the line of sight is important and similarly, the headlight sight distance is an important consideration for sag curve selection as shown in Figure A.2. The selection of maximum and minimum grades depend on the design speed/vehicle drainage conditions of the road respectively. In practice, for uncurbed pavements with adequate cross slopes, a 0% grade is sufficient. For curved pavements however, a longitudinal grade of 0.3-0.5% is typically used to encourage longitudinal flow of surface water [44].

Bibliography

- [1] AASHTO. American Association of State Highway and Transportation Officials, June 2013. URL <http://www.transportation.org/Pages/default.aspx>.
- [2] S.S. Abhyankar and C.L. Bajaj. Automatic Parameterization of Rational Curves and Surfaces III: Algebraic Plane Curves. *Computer Aided Geometric Design*, 5(4):309–321, 1988.
- [3] Highways Agency. *Manual of Contract Documents for Highway Works (MCHW)*, June 2013. URL <http://www.dft.gov.uk/ha/standards/ghost/mchw/vol3/section1.htm>.
- [4] Highways Agency. *Design Manual for Roads and Bridges (DMRB)*, June 2013. URL <http://www.dft.gov.uk/ha/standards/dmr/index.htm>.
- [5] J.M. Álvarez, T. Gevers, and A.M. López. Vision-based road detection using road models. In *IEEE International Conference on Image Processing*, pages 2073–2076, 2009.
- [6] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008.
- [7] S.N. Bernstein. Démonstration du théoreme de Weierstrass fondée sur le calcul des probabilités. *Comm. Soc. Math. Kharkov*, 13(1), 1912.
- [8] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, pages 179–195, 1975.
- [9] A. Bhoite, N. Beke, S. Nanduri, T. Duffy, and M. Torres. Advanced situational awareness and obstacle detection using a monocular camera. In *IEEE Image Processing Workshop (WNYIPW), Western New York*, pages 30–33, 2010.
- [10] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

-
- [11] M. Bizzarri. Rational Parameterization of Algebraic Curves. Master's thesis, University of West Bohemia, Department of Mathematics, 2008.
- [12] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. Lee, A. Stewart, P. Vernaza, J. Derenick, J. Spletzer, and B. Satterfield. Little Ben: The Ben Franklin racing team's entry in the 2007 Darpa Urban Challenge. *Journal of Field Robotics*, 25(9):598–614, 2008.
- [13] A. Bosch, A. Zisserman, and X. Munoz. Scene Classification Using a Hybrid Generative/Discriminative Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:712–727, April 2008.
- [14] M. Bratkova, S. Boulos, and P. Shirley. oRGB: A Practical Opponent Color Space for Computer Graphics. *IEEE Computer Graphics and Applications*, 29(1):42–55, 2009.
- [15] A. Broggi. Robust real-time lane and road detection in critical shadow conditions. In *Proceedings of the International Symposium on Computer Vision*, pages 353–358, 1995.
- [16] A. Broggi, P. Cerri, P. Medici, P.P. Porta, and G. Ghisio. Real Time Road Signs Recognition. In *IEEE Intelligent Vehicles Symposium*, pages 981–986, June 2007.
- [17] Alberto Broggi and Simona Berte. A morphological model-driven approach to real-time road boundary detection for vision-based automotive systems. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pages 73–80, 1994.
- [18] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2012.
- [19] P. Charbonnier, F. Diebolt, Y. Guillard, and F. Peyret. Road markings recognition using image processing. In *IEEE Conference on Intelligent Transportation Systems*, pages 912–917, 1997.
- [20] S.A. Coleman, S. Suganthan, and B.W. Scotney. Gradient operators for feature extraction and characterisation in range images. *Pattern Recognition Letters*, 31(9):1028–1040, 2010.
- [21] J.M. Collado, C. Hilario, A. de la Escalera, and J.M. Armingol. Detection and classification of road lanes with a frequency analysis. In *Proceedings of the Intelligent Vehicles Symposium*, pages 78–83, 2005.

-
- [22] J.M. Collado, C. Hilario, A. De La Escalera, and J.M. Armingol. Adaptive road lanes detection and classification. In *Advanced Concepts for Intelligent Vision Systems*, pages 1151–1162. Springer, 2006.
- [23] C. Corridori and M. Zanin. High curvature two-clothoid road model estimation. In *Proceedings of the International Conference on Intelligent Transportation Systems*, pages 630–635, 2004.
- [24] M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.
- [25] N. Vandapel D. Munoz, J.A. Bagnell and M. Hebert. Contextual classification with functional max-margin markov networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 975–982, 2009.
- [26] N. Dalal and B. Triggs. Histograms of Oriented Gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.
- [27] R. Danescu and S. Nedevschi. Detection and classification of painted road objects for intersection assistance applications. In *IEEE Conference on Intelligent Transportation Systems*, pages 433–438, 2010.
- [28] M.B. de Paula and C.R. Jung. Real-time detection and classification of road lane markings. In *IEEE Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 83–90, 2013.
- [29] M. Delahunt. Cross Section, June 2013. URL <http://www.artlex.com/ArtLex/c/crosssection.html>.
- [30] D. Ding, J. Yoon, and C. Lee. Traffic sign detection and identification using SURF algorithm and GPGPU. In *International SoC Design Conference*, pages 506–508, 2012.
- [31] B. Douillard, D. Fox, and F. Ramos. Laser and Vision Based Outdoor Object Mapping. In *Robotics Science and Systems*, 2008.
- [32] C. Duchow and B. Kortner. Aggregating lane markings into lanes for intersection assistance. In *IEEE Intelligent Vehicles Symposium*, pages 722–727, 2007.
- [33] G. Emmerson. Minimising disruption from roadworks, May 2010. Transport for London Surface Transport Panel.

-
- [34] M.R. Endsley. Towards a Theory of Situation Awareness in Dynamic Systems. *Human Factors*, 37(1):32–64, 1995.
- [35] N. Fairfield and C. Urmson. Traffic light mapping and detection. In *IEEE International Conference on Robotics and Automation*, pages 5421–5426, May 2011.
- [36] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531, 2005.
- [37] 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*, 24(6):381–395, 1981.
- [38] Department for Transport. The Highway Code. URL <https://www.gov.uk/highway-code>.
- [39] P. Foucher, Y. Sebsadji, J-P. Tarel, P. Charbonnier, and P. Nicolle. Detection and recognition of urban road markings using images. In *International Conference on Intelligent Transportation Systems*, pages 1747–1752, 2011.
- [40] C.L. Franklin. *Colour and Colour Theories*. Arno Press (New York), 1973.
- [41] Y. Freund and R.E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. *Computer and System Sciences*, 55:119–139, 1997.
- [42] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 41(5):337–407, 2000.
- [43] S. Friedman, H. Pasula, and D. Fox. Voronoi Random Fields: Extracting Topological Structure of Indoor Environments via Place Labeling. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 7, pages 2109–2114, 2007.
- [44] N.J. Garber and L.A. Hoel. *Traffic and Highway Engineering*. Thompson Publishing Company, 2002.
- [45] Geograph. The Geograph Project, August 2012. URL <http://www.geograph.org.uk>.
- [46] T. Gevers, A. Gijsenij, J. van de Weijer, and J-M. Geusebroek. *Color in Computer Vision: Fundamentals and Applications*, volume 24. Wiley, 2012.

-
- [47] H. Gomez-Moreno, S. Maldonado-Bascon, P. Gil-Jimenez, and S. Lafuente-Arroyo. Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition. *IEEE Transactions on Intelligent Transportation Systems*, 11(4):917–930, 2010.
- [48] Google. Google Protocol Buffers, October 2013. URL <https://developers.google.com/protocol-buffers/>.
- [49] R. Graf, A. Wimmer, and K.C.J. Dietmayer. Probabilistic estimation of temporary lanes at road work zones. In *International Conference on Intelligent Transportation Systems*, pages 716–721, 2012.
- [50] T. Gumpp, D. Nienhuser, R. Liebig, and J.M. Zollner. Recognition and tracking of temporary lanes in motorway construction sites. In *IEEE Intelligent Vehicles Symposium*, pages 305–310, 2009.
- [51] C. Guo, S. Mita, and D. McAllester. Robust road detection and tracking in challenging scenarios based on Markov Random Fields with unsupervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1338–1354, 2012.
- [52] J. Han, D. Kim, M. Lee, and M. Sunwoo. Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor. *IEEE Transactions on Vehicular Technology*, 61(3):971–985, 2012.
- [53] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*, volume 2. Cambridge Univ Press, 2000.
- [54] A.B. Hillel, R. Lerner, D. Levi, and G. Raz. Recent progress in road and lane detection: a survey. *Machine Vision and Applications*, pages 1–19, 2012.
- [55] G. Hoefel and C. Elkan. Learning a two-stage SVM/CRF sequence classifier. In *Proceedings of the ACM Conference on Information and Knowledge Management*, pages 271–278, 2008.
- [56] A.S. Huang and S. Teller. Lane estimation for autonomous vehicles using vision and lidar, 2010. URL http://rvsn.csail.mit.edu/Pubs/phd_ashuang_2010feb_laneestimation.pdf.
- [57] A.S. Huang and S. Teller. Lane boundary and curb estimation with lateral uncertainties. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1729–1734, 2009.
- [58] A.S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller. Finding multiple lanes in urban road networks with vision and Lidar. *Autonomous Robots*, 26(2-3):103–122, 2009.
- [59] R.W.G. Hunt and M.R. Pointer. *Measuring Colour*. Wiley, 2011.

-
- [60] L.M. Hurvich. *Color Vision*. Sinauer Associates, 1981.
- [61] L.M. Hurvich and D. Jameson. An opponent-process theory of color vision. *Psychological Review*, 64 (6, Part 1):384–404, 1957.
- [62] L.M. Hurvich and D. Jameson. Perceived color, induction effects, and opponent-response mechanisms. *The Journal of General Physiology*, 43(6):63–80, 1960.
- [63] O. Ikeda. Segmentation of faces in video footage using HSV color for face detection and image retrieval. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 913–916, Sept. 2003.
- [64] SICK Sensor Intelligence. *Scanners laser LMS1xx / LMS111 / Outdoor / Short Range*. SICK, 2013. URL www.mysick.com.
- [65] H. Ishida, K. Kidono, Y. Kojima, and T. Naito. Road marking recognition for map generation using sparse tensor voting. In *International Conference on Pattern Recognition*, pages 1132–1135, 2012.
- [66] Y. Ishizuka and Y. Hirai. Segmentation of road sign symbols using opponent-color filters. In *Intelligent Transportation Systems World Congress*, 2004.
- [67] D. Jameson and L.M. Hurvich. Some quantitative aspects of an opponent-colors theory. I. Chromatic responses and spectral saturation. *Journal of the Optical Society of America*, 45(7):546–552, 1955.
- [68] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, June 2010.
- [69] Y. Kang, C. Roh, S-B. Suh, and B. Song. A Lidar-based decision-making method for road boundary detection using multiple Kalman filters. *IEEE Transactions on Industrial Electronics*, 59(11): 4360–4368, 2012.
- [70] D. Kerr, S. Coleman, M. McGinnity, Q. Wu, and M. Clogenson. Biologically inspired edge detection. In *IEEE International Conference on Intelligent Systems Design and Applications*, pages 802–807, 2011.
- [71] D. Kerr, S.A. Coleman, and B.W. Scotney. Finite Element Laplacian Feature Detector. In *IAPR Conference on Machine Vision Applications*, pages 381–384, 2011.

-
- [72] A. Kheyrollahi and T.P. Breckon. Automatic real-time road marking recognition using a feature driven approach. *Machine Vision and Applications*, 23(1):123–133, 2012.
- [73] S-H. Kim, C-W. Roh, S-C. Kang, and M-Y. Park. Outdoor navigation of a mobile robot using differential GPS and curb detection. In *IEEE International Conference on Robotics and Automation*, pages 3414–3419, 2007.
- [74] D. Kollar and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.
- [75] H. Kong, J-Y. Audibert, and J. Ponce. Vanishing point detection for road detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 96–103, 2009.
- [76] K. Konolige, M. Agrawal, R.C. Bolles, C. Cowan, M. Fischler, and B. Gerkey. Outdoor mapping and navigation using stereo vision. In *Experimental Robotics*, pages 179–190. Springer, 2008.
- [77] R. A. Krammes and M. A. Garnham. Worldwide review of alignment design policies. Technical report, Transportation Research Board, 1998.
- [78] S. Kumar and M. Hebert. Discriminative Random Fields. *International Journal of Computer Vision*, 68(2):179–201, 2006.
- [79] R. Triebel L. Spinello and R. Siegwart. Multiclass multimodal detection and tracking in urban environments. *The International Journal of Robotics Research*, 29(12):1498–1515, 2010.
- [80] A. De la Escalera, J.M. Armingol, and M. Mata. Traffic sign recognition and analysis for intelligent vehicles. *Image and Vision Computing*, 21(3):247–258, 2003.
- [81] Transport Research Laboratory. Highways Agency Road Research Information System (HARRIS), June 2013. URL http://www.trl.co.uk/facilities/mobile_test_equipment/highways_agency_road_research_information_system.htm.
- [82] L. Ladicky, C. Russell, P. Kohli, and P.H.S. Torr. Associative hierarchical CRFs for object class image segmentation. In *International Conference on Computer Vision*, pages 739–746, 2009.
- [83] J. Lafferty, A. McCallum, and F.C.N. Pereira. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001.

-
- [84] Kevin Lai and Dieter Fox. Object recognition in 3D point clouds using web data and domain adaptation. *The International Journal of Robotics Research*, 29(8):1019–1037, 2010.
- [85] D-T. Lee and B.J. Schachter. Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer and Information Sciences*, 9(3):219–242, 1980.
- [86] J. Levinson, J. Askeland, J. Dolson, and S. Thrun. Traffic light mapping, localization, and state detection for autonomous vehicles. In *IEEE International Conference on Robotics and Automation*, pages 5784–5791, May 2011.
- [87] T.D.C. Little, A. Agarwal, J. Chau, M. Figueroa, A. Ganick, J. Lobo, T. Rich, and P. Schimitsch. Directional communication system for short-range vehicular communications. In *Vehicular Networking Conference*, pages 231–238, 2010.
- [88] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.
- [89] G. Monfardini M. Gori and F. Scarselli. A new model for learning in graph domains. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, volume 2, pages 729–734, 2005.
- [90] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.
- [91] B. Mathibela, M.A. Osborne, I. Posner, and P. Newman. Can Priors Be Trusted? Learning to Anticipate Roadworks. In *IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, USA, 2012.
- [92] B. Mathibela, I. Posner, and P. Newman. A roadwork scene signature based on the opponent colour model. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4394–4400, 2013.
- [93] MATLAB. *version 7.14.0.739 (R2012a)*. The MathWorks Inc., Natick, Massachusetts, 2012.
- [94] A. Mogelmose, M.M. Trivedi, and T.B. Moeslund. Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1484–1497, 2012.

-
- [95] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, A. Stavens, A. Vogt, and S. Thrun. Junior: The stanford entry in the Urban Challenge. *Journal of Field Robotics*, 25(9): 569–597, 2008.
- [96] K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc., 1999.
- [97] ABC News. Driverless car hits diner outside Gouger Street restaurant after owner’s remote control was activated. *ABC News Australia*, October 2013. URL <http://www.abc.net.au/news/2013-10-17/driverless-car-hits-diners-after-remote-misfires/5028950>.
- [98] A. Niculescu-Mizil and R. Caruana. Obtaining Calibrated Probabilities from Boosting. In *Uncertainty in Artificial Intelligence*, page 413, 2005.
- [99] M. Noda, T. Takahashi, I. Ide, H. Murase, Y. Kojima, and T. Naito. Recognition of Road Markings from In-Vehicle Camera Images by a Generative Learning Method. In *Machine Vision Application*, pages 514–517, 2009.
- [100] G.O. Ogonda. *Recovery of the geometric road design elements using low-cost sensors: a feasibility study*. PhD thesis, Stuttgart University, 2009.
- [101] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.
- [102] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155:23–36, 2006.
- [103] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527, 2007.
- [104] F. Oniga and S. Nedeveschi. Polynomial curb detection based on dense stereovision for driving assistance. In *International Conference on Intelligent Transportation Systems*, pages 1110–1115, 2010.

-
- [105] F. Oniga and S. Nedevschi. Curb detection for driving assistance systems: A cubic spline-based approach. In *Intelligent Vehicles Symposium*, pages 945–950, 2011.
- [106] F. Oniga, S. Nedevschi, and M.M. Meinecke. Curb detection based on a multi-frame persistence map for urban driving scenarios. In *IEEE International Conference on Intelligent Transportation Systems*, pages 67–72, 2008.
- [107] OpenStreetMap. OpenStreetMap Wiki, 2013. URL http://wiki.openstreetmap.org/w/index.php?title=Main_Page&oldid=941300.
- [108] D. Parikh and D. Batra. CRFs for Image Classification. 2008.
- [109] R.R. Patterson. Projective transformations of the parameter of a Bernstein-Bézier curve. *ACM Transactions on Graphics (TOG)*, 4(4):276–290, 1985.
- [110] K. Peterson, J. Ziglar, and P.E. Rybski. Fast feature detection and stochastic parameter estimation of road shape using multiple LIDAR. In *International Conference on Intelligent Robots and Systems*, pages 612–619, 2008.
- [111] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
- [112] L.A. Piegl and W. Tiller. *The NURBS Book*. Springer, 1997.
- [113] A.Y. Ng R. Socher, C.C. Lin and C.D. Manning. Parsing natural scenes and natural language with Recursive Neural Networks. In *Proceedings of the 28th International Conference on Machine Learning*, pages 129–136, 2011.
- [114] L.R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [115] F.T. Ramos, D. Fox, and H.F. Durrant-Whyte. CRF-Matching: Conditional Random Fields for Feature-Based Scan Matching. In *Robotics Science and Systems*, 2007.
- [116] A. Richardson and E. Olson. Learning Convolutional Filters for Interest Point Detection. In *Proceedings of the International Conference on Robotics and Automation*, May 2013.

-
- [117] S. Richter and D. Wetzel. A robust and stable road model. In *Proceedings of the IAPR International Conference on Pattern Recognition, Conference A: Computer Vision and Image Processing*, volume 1, pages 777–780, 1994.
- [118] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, pages 430–443. Springer, 2006.
- [119] P.M. Salmon, N.A. Stanton, and K.L. Young. Situation awareness on the road: review, theoretical and methodological issues, and future directions. *Theoretical Issues in Ergonomics Science*, 13(4): 472–492, 2012.
- [120] M. Schmidt. UGM: A Matlab Toolbox for Probabilistic Undirected Graphical Models, 2010. URL <http://www.di.ens.fr/~mschmidt/Software/UGM.html>.
- [121] C. Seiffert, T.M. Khoshgoftaar, J. van Hulse, and A. Napolitano. RUSBoost: Improving classification performance when training data is skewed. In *International Conference on Pattern Recognition*, pages 1–4, 2008.
- [122] C. Seiffert, T.M. Khoshgoftaar, J. van Hulse, and A. Napolitano. RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(1):185–197, Jan. 2010.
- [123] Y-W. Seo, D. Wettergreen, and W. Zhang. Recognizing temporary changes on highways for reliable autonomous driving. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 3027–3032, 2012.
- [124] W.G. Shadeed. Road traffic sign detection in color images. In *IEEE International Conference on Electronics, Circuits and Systems*, 2003.
- [125] C. Siagian and L. Itti. Rapid Biologically-Inspired Scene Classification Using Features Shared with Visual Attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2), Feb 2007.
- [126] M. Siegel and R. MacLachlan. Radar obstacle detection: finding moving targets in noisy range data. In *Proceedings of the Instrumentation and Measurement Technology Conference*, volume 1, pages 145–150, 2002.
- [127] J. Siegemund, D. Pfeiffer, U. Franke, and W. Forstner. Curb reconstruction using Conditional Random Fields. In *IEEE Intelligent Vehicles Symposium*, pages 203–210, 2010.

-
- [128] J. Siegemund, U. Franke, and W. Forstner. A temporal filter approach for detection and reconstruction of curbs and road surfaces based on Conditional Random Fields. In *IEEE Intelligent Vehicles Symposium*, pages 637–642, 2011.
- [129] A. Skordylis and N. Trigoni. Efficient data propagation in traffic-monitoring vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):680–694, 2011.
- [130] L. Smadja, J. Ninot, and T. Gavrilovic. Road extraction and environment interpretation from Lidar sensors. *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38:281–286, 2010.
- [131] B. Soheilian, N. Paparoditis, and D. Boldo. 3D road marking reconstruction from street-level calibrated stereo pairs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(4):347–359, 2010.
- [132] W. Stefan, M. Nikodem, S. Konrad, and B. Schiele. New features and insights for pedestrian detection. In *Conference on Computer Vision and Pattern Recognition*, San Francisco, June 2010.
- [133] S. Strygulec, D. Muller, M. Meuter, C. Nunn, S. Ghosh, and C. Wohler. Road boundary detection and tracking using monochrome camera images. In *International Conference on Information Fusion*, pages 864–870, 2013.
- [134] W.A. Taylor. Change-Point Analysis: A Powerful New Tool For Detecting Changes. Technical report, Taylor Enterprises Inc, 2012. URL <http://www.variation.com/cpa/tech/changepoint.html>.
- [135] A. Torralba, K.P. Murphy, and W.T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 762–769, July 2004.
- [136] M. Tsogas, N. Floudas, P. Lytrivis, A. Amditis, and A. Polychronopoulos. Combined lane and road attributes extraction by fusing data from digital map, laser scanner and camera. *Information Fusion*, 12(1):28–36, 2011.
- [137] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M.N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T.M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y-W. Seo, S. Singh, J. Snider, A. Stentz, W.R.L. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and

- D. Ferguson. Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.
- [138] S. Vacek, C. Schimmel, and R. Dillmann. Road-marking analysis for autonomous vehicle guidance. In *European Conference on Mobile Robots*, 2007.
- [139] K.E.A. van de Sande, T. Gevers, and C.G.M. Snoek. Evaluating Color Descriptors for Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.
- [140] J.C. van Gemert, C.J. Veenman, A.W.M. Smeulders, and J-M. Geusebroek. Visual Word Ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271–1283, July 2010.
- [141] T. Veit, J-P. Tarel, P. Nicolle, and P. Charbonnier. Evaluation of road marking feature extraction. In *International Conference on Intelligent Transportation Systems*, pages 174–181, 2008.
- [142] H. Weigel, H. Cramer, G. Wanielik, A. Polychronopoulos, and A. Saroldi. Accurate road geometry estimation for a safe speed application. In *IEEE Intelligent Vehicles Symposium*, pages 516–521, 2006.
- [143] C. Wengert, M. Douze, and H. Jégou. Bag-of-colors for improved image search. In *ACM Multimedia*, Scottsdale, United States, October 2011.
- [144] I. West. Old Factory Building, . URL <https://3dwarehouse.sketchup.com/model.html?id=d9dfc9a14c075f1ab48d1a35afd3934>.
- [145] I. West. Urban Street Junction (Small), . URL <http://sketchup.google.com/3dwarehouse/details?mid=f056257731486bd54aa2c9ef64f3f582>.
- [146] S. Westland, C. Ripamonti, and V. Cheung. *Computational Colour Science Using MATLAB*. Wiley, 2012.
- [147] W.S. Wijesoma, K.R.S. Kodagoda, and A.P. Balasuriya. Road-boundary detection and tracking using ladar sensing. *IEEE Transactions on Robotics and Automation*, 20(3):456–464, 2004.
- [148] A. Wimmer, T. Weiss, F. Flogel, and K. Dietmayer. Automatic detection and classification of safety barriers in road construction sites using a laser scanner. *IEEE Intelligent Vehicles Symposium*, June 2009.

- [149] A. Wimmer, T. Weiss, F. Flogel, and K. Dietmayer. Automatic detection and classification of safety barriers in road construction sites using a laser scanner. In *Intelligent Vehicles Symposium*, pages 578–583, June 2009.
- [150] A. Wimmer, T. Jungel, M. Glück, and K. Dietmayer. Automatic generation of a highly accurate map for driver assistance systems in road construction sites. In *IEEE Intelligent Vehicles Symposium*, pages 281–286, June 2010.
- [151] A. Wimmer, R. Graf, and K.C.J. Dietmayer. Matching highly accurate maps to local environmental perception at road construction sites. In *Intelligent Vehicles Symposium*, pages 975–980, 2011.
- [152] T. Wu and A. Ranganathan. A practical system for road marking detection and recognition. In *IEEE Intelligent Vehicles Symposium*, pages 25–30, 2012.
- [153] B. Yang, L. Fang, and J. Li. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 79:80–93, 2013.
- [154] J. Yang, Y-G. Jiang, A.G. Hauptmann, and C-W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the International Workshop on Multimedia Information Retrieval*, pages 197–206, New York, NY, USA, 2007.
- [155] L. Yunchong, H. Kezhong, and J. Peifa. Road markers recognition based on shape information. In *Intelligent Vehicles Symposium*, pages 117–122, 2007.
- [156] W. Zhang. LIDAR-based road and road-edge detection. In *IEEE Intelligent Vehicles Symposium*, pages 845–848, 2010.