Inferring Road Boundaries Through and Despite Traffic

Tarlan Suleymanov Paul Amayo Paul Newman

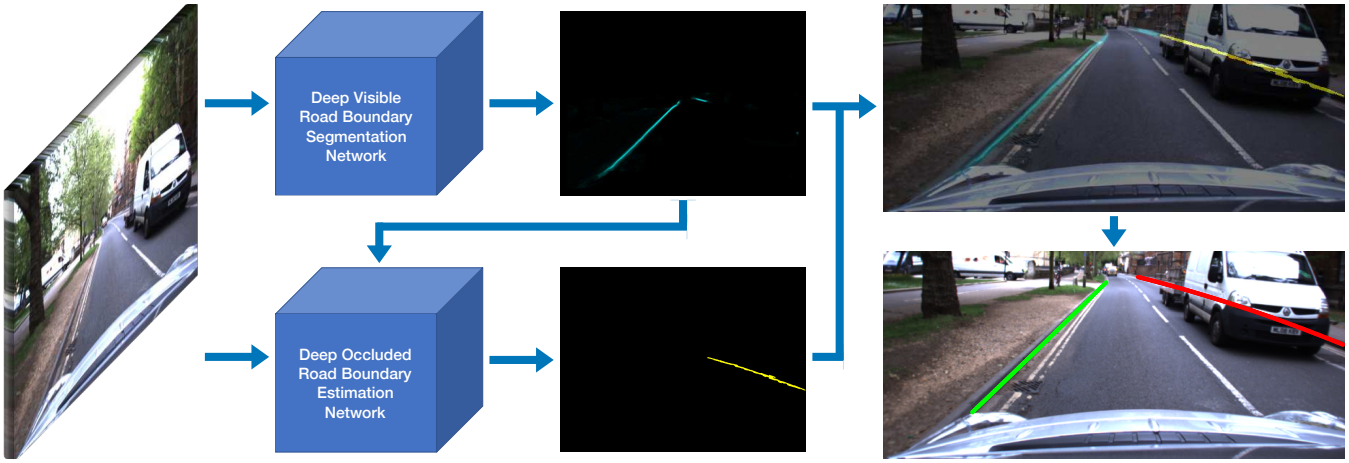

Fig. 1: Given an input image, road boundaries are inferred irrespective of whether or not the boundaries are actually visible. Our coupled approach first segments visible road boundaries with a fully convolutional network and then passes output to our deep network to infer occluded road boundaries. Our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

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

I. Introduction

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

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

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

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

The main contributions of this paper are as follows:

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

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

II. Related Work

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

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

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

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

III. Obtaining Ground Truth and Training Data

A. 3D Annotation

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

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

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

B. Partitioning Training Data

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

IV. Our approach

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

A. Detecting visible curbs

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

B. Hallucinating occluded curbs

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

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

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

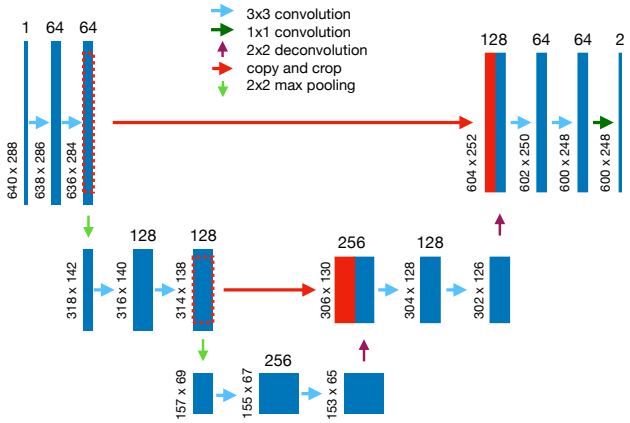

Fig. 4: 3 layers deep fully convolutional U-net architecture. To localise detected objects more precisely, the network concatenates higher resolution "input-side" features from convolution layers with up-sampled outputs form deconvolution layers as illustrated with red arrows.

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

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

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

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

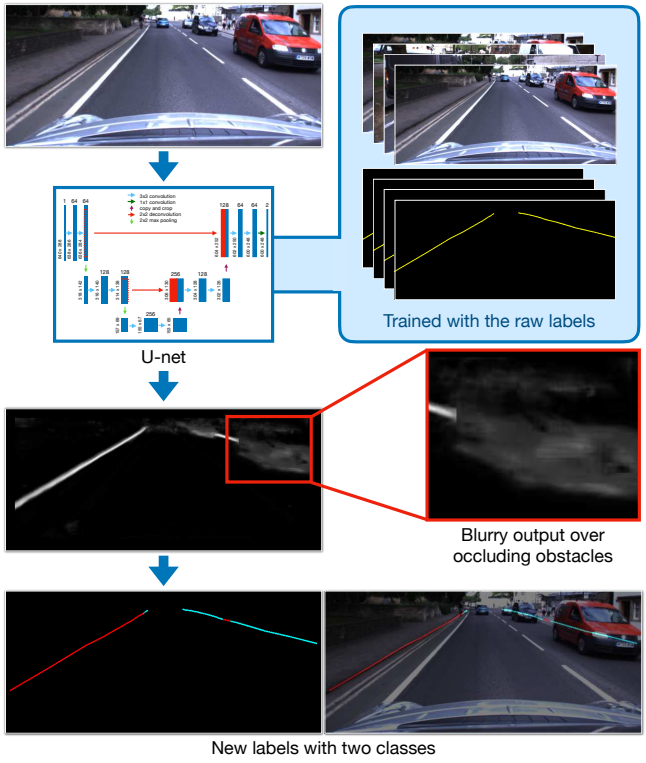

Fig. 5: Given input images, the U-net model trained with the raw labels detects *visible* road boundaries, but generates blurry outputs over occluding obstacles. We obtain masks for detected *visible* boundaries by applying threshold to the outputs. AND operation between the raw labels and thresholded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

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

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

and

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

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

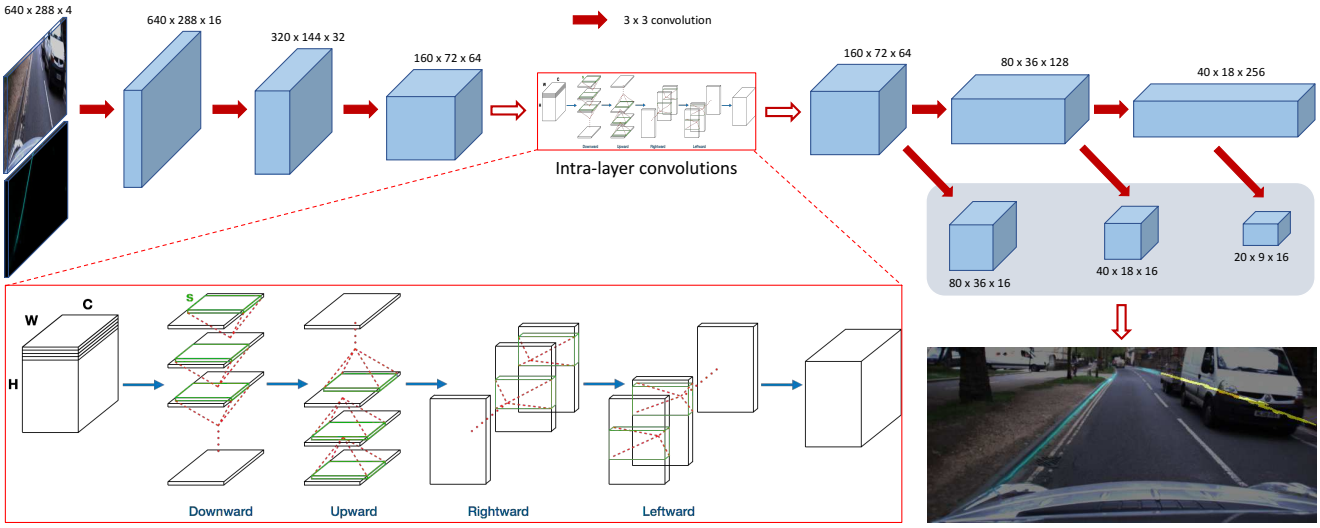

Fig. 6: Our model architecture for inferring *occluded* road boundaries. The model takes an RGB image and mask of detected *visible* road boundaries as inputs. The model consists of convolutional layers and there are 3 "base" layers followed by intra-layer convolutions that are slice-by-slice convolutions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

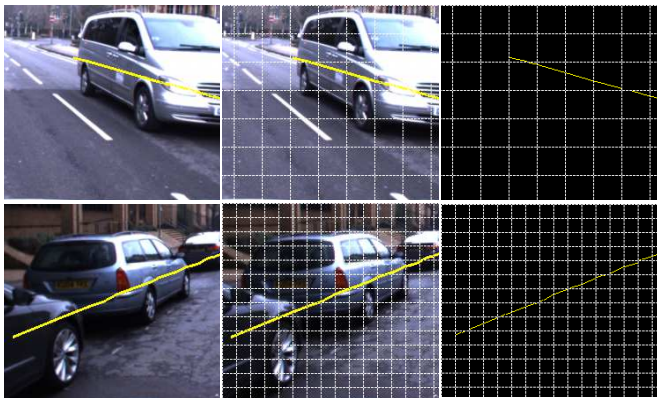

Fig. 7: Examples of parameterisation of *occluded* road boundary labels. In the left column pixel-wise labels are shown followed by the division of pixel-wise masks into a grid of squares at different scales. The final *occluded* road boundary masks are drawn based on parameterised labels in the right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

the discrete loss for the i-th scale is:

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

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

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

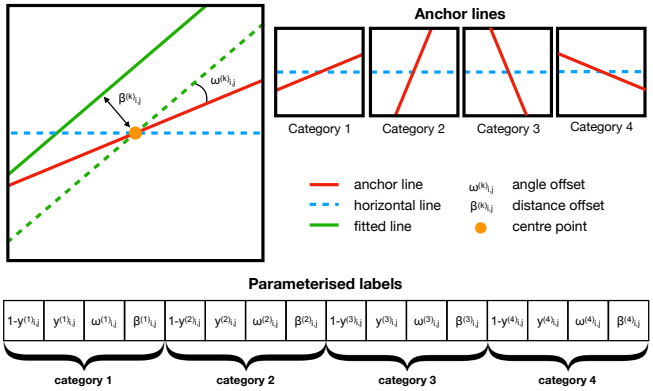

Fig. 8: Parameterisation of road boundary lines in discrete-continuous form. Each cell of the grid at each scale is represented with 16 parameters: 4 numbers for each line category. Lines are fitted to road boundaries in each cell and are assigned to one of 4 anchor line categories. Then offsets from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

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

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

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

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

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

V. Road Boundary Set Formation

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

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

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

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

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

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

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

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

VI. Experimental Results

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

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

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

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

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

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

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

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

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

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

TABLE III: Average Running time per task

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

VII. Conclusions

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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