LEARNING TO PLAN SEMANTIC FREE-SPACE BOUNDARY

Ziyi Yin, Ziyang Song, Zejian Yuan

Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University, China
{yzy19980922@stu.xjtu.edu.cn}, {songziyang@stu.xjtu.edu.cn}, {yuan.ze.jian@xjtu.edu.cn}

ABSTRACT

Recently, free-space detection has attracted widespread attention. Most existing methods treat free-space detection as a semantic segmentation task. In this paper, we propose a novel approach to directly infer the boundary of the semantic free-space from a single image. Firstly, we design a multi-stage CNN to produce 2D belief maps with high resolution for boundary segments of different semantic classes, such as road boundary, vertical obstacles on road and so on. The proposed CNN architecture can implicitly learn boundary structure and long-range spatial context. Then, based on the 2D belief maps we address the semantic free-space detection as a dynamic programming problem to ensure the spatial smoothness of the predicted boundary. The experimental results on our dataset show that our method has a convincing performance on various quantitative metrics.

Index Terms— semantic free-space, multi-stage CNN, dynamic programming

1. INTRODUCTION

As a crucial component in Advanced Driver Assist Systems (ADAS), free-space detection requires a prediction of the ground plane in a particular traffic scene, which can provide useful information, such as latent hazardous and drivable space, for the trajectory programming.

Free-space detection is widely considered as a semantic segmentation task. Several strategies have been developed to detect a fine-grained free-space, such as occupancy grids [1][2][3], stixel world algorithm [4][5][6][7]. Recently, with the widespread application of fully convolutional networks(FCNs) [8], free-space detection can be efficiently realized in the pixel-level without the constrains of input size. For example, paper [9] proposed an approach with FCNs to achieve free-space detection in the pixel-level.

However, there are still several defects of the above methods. A common flaw is that most of these methods [10][11][12] lack the semantic information of free-space boundary. Although these methods can detect the free-space finely, not all detected free-space are drivable. Other methods [13][14][15][16] only segment the pixels belong to free-space and lack the planning for the boundary of these pixels, thus is still hard to describe the surrounding drivable situation clearly. We aim to predict semantic free-space boundary, which can provide useful and applicable information for trajectory programming in ADAS. As shown in Fig. 1, given an input image of a traffic scene, our model localizes a spatially smooth boundary of free-space and segment it into several parts semantically belonging to vertical, step and flat, which represent road boundary (Step), vertical obstacles on road (Vertical), and partial flat road (Flat) respectively. Step and vertical are easier to detect owing to their clearer structures in the traffic scene, while flat detection remains a challenging task. To address this problem, we make use of the sufficient context information in traffic scenes to provide cues and accurately infer semantic free-space boundary in the pixel level.

Fig. 1. A free-space boundary with semantic information, where red, green and yellow areas denote the vertical, flat and step respectively.

More specifically, our approach starts with a multi-stage CNN to produce a 2D belief map with high resolution. Then the belief map is fed into a dynamic programming algorithm, resulting in a reasonable boundary. We also exploit a coarse-to-fine training strategy to help the network learn to regress free-space boundary more efficiently and accurately. The experimental results demonstrate that our approach can infer a semantic free-space with satisfying semantic accuracy and spatial smoothness.

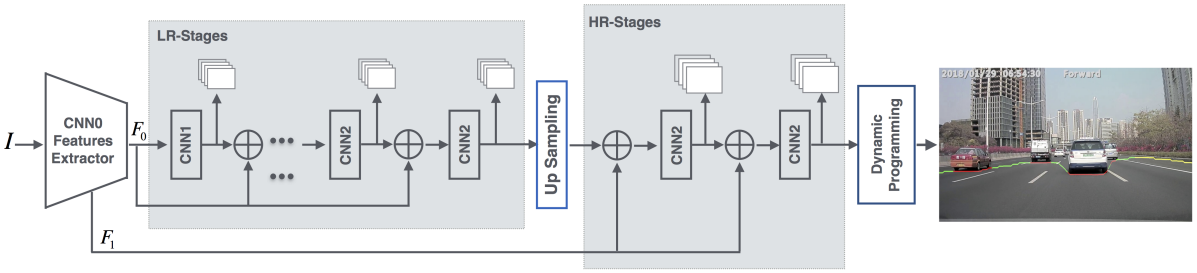

Fig. 2. The system of semantic free-space boundary detection. F_0 represents the output of CNN0 Features Extractor with 1/8 input size, F_1 represents a feature map of 4 times downsampling in CNN0 Features extractor.

2. OUR APPROACH

2.1. Multi-stage CNN

Overall Network: Our network for semantic free-space boundary detection follows the architecture proposed by [17], in which we use a simple CNN structure as our feature extractor and a sequence of convolution-based predictors for belief map regression. For an input image of a traffic scene, multi-stage CNN outputs a belief map with higher spatial resolution, denoted as H. The overall network of our approach is illustrated in Fig. 2.

CNN0 Features Extractor: We adopt 2× Conv (3x3x16), 2× Conv (3x3x32) and 3× MaxP (2x2) in CNN0 Feature Extractor. The feature maps are spatially downsampled by a factor of two for three times to ensure a rapidly-growing receptive field, resulting in a feature map with 32 channels and 1/8× size of the input image.

Low Resolution Stages (LR-Stages): LR-Stages consist of 5 stages. The configuration of the first stage CNN1 is as follows: 3× Conv (3x3x32), Conv(1x1x4)+sigmoid and the rest four stages are composed of CNN2 as follows: 3× Conv (7x7x16), Conv (1x1x4)+sigmoid. Since each stage in LR-Stages can produce a belief map, we concatenate the output of the previous stage and feature map output by the feature extractor (F_0 in Fig. 2) as the input to the next stage. Finally, LR-Stages give a feature map with 1/8× input size and 4 channels.

The first LR-Stage regresses a belief map based on features produced by the feature extractor, while the following stages can improve their predictions by combining the original feature map with the spatial context information provided by the noisy belief map from its previous stage.

High Resolution Stages (HR-Stages): HR-Stages can promote the resolution of the belief map produced by LR-Stages. We use 2 stages owning the same configuration as CNN2. We upsample the output of LR-Stages and concatenate the results with F_1 (in Fig. 2) as input to HR-Stages. The HR-Stages finally generate a belief map with 4 channels and 1/4× input size for the free-space boundary, denoted by H. Each channel represents background (H_B), flat (H_F), vertical (H_V) and step (H_S). HR-stages can also be extended by adding more stages and applying to higher spatial resolution.

To simplify the process of subsequent inference, we integrate H_F, H_V, H_S into a single-channel belief map C by adding them on channels: $C = H_S + H_V + H_F$.

2.2. Semantic free-space boundary planning

Enlightened by [18], we propose a strategy to infer the semantic free-space boundary with precise locating and spatial smoothness.

For belief map C, we first store the pixels of C in columns, like $\{C_1, C_2....C_N\}$, in which N represents the width of the belief map. Our task is to find the best boundary with the greatest confidence and spatial smoothness. More precisely, we need select a pixel from each column to form the boundary. Mathematically, the optimization solution can be described as follows:

$$(p_1{}^* \dots p_N{}^*) = \arg\max_{p_1 \dots p_N} [\sum_{n=1}^{N} C_n(p_n) + \sum_{n=2}^{N} S(p_n, p_{n-1})] \quad p_n \in \{1 \dots H\}, \tag{1}$$

$$S(p_n, p_{n-1}) = \alpha(p_n - p_{n-1})^2, \tag{2}$$

where p_n represents the row coordinate of a pixel in the n-th column, M represents height of the image and $C_n(p_n)$ represents the confidence of pixel (p_n, n) in C. $S(p_n, p_{n-1})$ is a smoothness constraint to prevent the discontinuity of two adjacent pixels. We use dynamic programming to solve the above problem. Specifically, we use the following recursive equation as:

$$D_n(p_n) = C_n(p_n) + \max_{p_{n-1}} [S(p_n, p_{n-1}) + D_{n-1}(p_{n-1})], \tag{3}$$

the programming result $\{p_1{}^* ... p_N{}^*\}$ is obtained by the tracing back algorithm in dynamic programming. For each row

coordinate ${p_n}^*$ in sequence, we use the max confidence in C to give pixel $({p_n}^*, n)$ a semantic label:

$$l({p_n}^*, n) = \underset{d \in \{S,V,F\}}{\arg\max}\ H_d({p_n}^*, n). \tag{4}$$

Additionally, the semantic information of each pixel in $\{{p_1}^* ... {p_N}^*\}$ can also be inferred by dynamic programming.

2.3. Training

We propose a coarse-to-fine strategy to guide the network to incorporate large scale context information for semantic prediction at first, and then turn to details for more accurate localization in the training process. To be specific, we train our network for 160 epochs, which are equally divided into 4 phases. In each phase, we use Gaussian kernels with size β to blur the original labels. As the training epochs increase, β gradually decreases as $\{11, 9, 7, 5\}$ at the start of each phase.

We use L2 loss in the output of each stage including HR-stages and LR-Stages, i.e.,

$$L = \sum_{t=1}^{T} l_t(\beta), \tag{5}$$

where $l_t(\beta)$ represent the L2 distance between the predicted belief maps and the blur labels.

3. EXPERIMENTS

3.1. Dataset and Evaluation

Dataset: We construct a dataset by ourselves to evaluate our method. All images are captured from urban and highway traffic scenes through a car camera. To enhance the robustness of our dataset, we specially collect images of some adverse conditions such as shadowed or reflective scenes.

The dataset consists of 1093 training examples and 172 test examples with 1280×720 size and RGB channels. For each image, we roughly annotate step, flat and vertical separately on three channels with broken lines.

Evaluation: We propose Distance Loss (DL) and Semantic Accuracy (SA) to evaluate our approach according to boundary similarity and the accuracy of semantic prediction respectively.

Distance Loss is defined as the similarity between our predicted and the labelled free-space boundary. Firstly, we use distance transformation [19] to derive the closest distance of each pixel in the predicted boundary to the annotated boundary. Then we sum and average the result. We use $T(p_n, n)$ to represent the distance from the pixel (p_n, n) in the predicted boundary to the nearest pixel in the annotated boundary, and the distance loss can be represented as: $DL = \left[\sum_{n=1}^{N} T(p_n, n)\right] /N.$

Semantic Accuracy is defined as the accuracy of semantic prediction by comparing each pixel on the predicted boundary to the nearest label pixel. We use $S(p_n, n)$ to represent whether the predicted class of the pixel (p_n, n) is the same as the class of label pixel closest to it. The semantic accuracy can be counted as: $SA = \left[\sum_{n=1}^{N} S(p_n, n)\right] /N$, where $S(p_n, n) \in \{0, 1\}$.

3.2. Setup

In training procedure, we use Adam Optimizer with the default parameter settings suggested in the paper [20]. We set the learning rate to 0.0001 and the size of mini-batch to 2.

In prediction procedure, we experiment with 172 testing examples in our dataset. Given an RGB image of size 720×1280, we scale it down to 360×640 and use it as the network input. The network outputs a 4-channel belief map of size 90×160 and scales it to 720×1280 for dynamic programming to get the final result.

3.3. Quantitative evaluation results

Table 1. Performance on different structures and strategies. DL: Distance Loss, SA: Semantic Accuracy.

	DL(pixel)	SA
Base	9.36	0.75
BASE+HR	7.32	0.82
BASE+CF	8.73	0.79
BASE+HR+CF	**6.72**	**0.85**

We compare our approach in terms of network structure and training strategy on our test data. Our baseline model (BASE) is a multi-stage CNN only with LR-Stages, and adopts the training strategy in [17]. Besides the baseline model, we also test the performance of coarse-to-fine (CF) training strategy and HR-Stages (HR).

With the same configurations of other variables, we experiment on our test data and average the results, which are presented in Table. 1. The application of HR-Stages and coarse-to-fine strategy both improve the performance of the baseline approach concerning all metrics. In addition, the benefits of using both the HR-Stages structure and the coarse-to-fine strategy are significant, reducing the distance loss by 2 pixels and increasing the semantic accuracy by 10%.

We also test the benefits of HR-Stages and coarse-to-fine strategy regarding the networks ability to fit with most of the traffic scenes. In Fig. 3, X-axis denotes the distance loss while Y-axis represents the proportion occupied by images of which distance loss is lower than the threshold value corresponding to the X-axis in the test samples. As shown in Fig. 3, with HR-Stages the network performs better in capturing those explicit

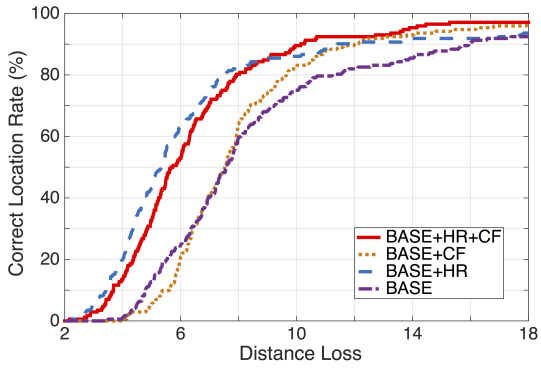

Fig. 3. Comparisons of different training strategies and network structures on the metric of distance loss.

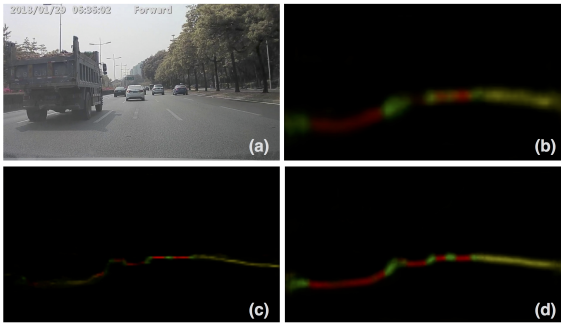

Fig. 4. Output of our multi-stage CNN with different setups. a) Input Image; b) BASE+CF; c) BASE+HR; d) BASE+HR+CF.

structural features, therefore resulting in more high-quality prediction results under a strict distance loss threshold. As the threshold increases, traffic scenes lacking clear structures and thus difficult to plan emerges. Our coarse-to-fine training strategy helps the network raise the correct location rate on such samples. The improvements should be attributed to large scale context information captured in the early training phases, which can guide the network to locate the boundary more accurately in traffic scenes without clear structures.

Qualitative evaluations: We show an intermediate result from our multi-stage CNN to illustrate the advantages of our HR-Stages structure and coarse-to-fine strategy. Comparison between Fig. 4(b) and 4(d) proves that the network with HR-Stages localizes the boundary more precisely and results in a belief map with higher resolution. What's more, by comparing Fig. 4(c) and 4(d), we observe that the network trained with our coarse-to-fine strategy can better learn the boundary structure and long-range spatial context.

Some test samples processed through dynamic programming are also depicted in Fig. 5. Owing to HR-Stages and coarse-to-fine strategy, our approach can give a semantically accurate and spatially smooth free-space boundary prediction. The approach also exhibits robustness in various conditions, even on shadowed or reflective roads. However, not all traffic scenes can be well handled, due to the lack of spatial context features in some extremely adverse conditions. As shown in Fig. 6, our approach can only correctly infer a few intervals of the semantic free-space boundary in the image. There are some extra results at https://youtu.be/FrhR4VPeg58.

Fig. 5. inferential results in different traffic scenes. a) shadowed; b) highway with fork; c) urban; d) reflective.

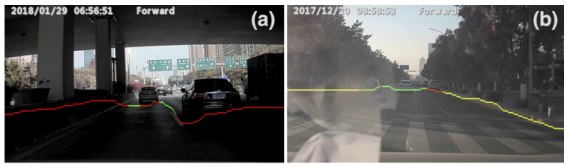

Fig. 6. Some examples with poor results. a) extremely shadowed; b) Large area of reflection in the image.

4. CONCLUSION

We propose an approach to infer the semantic free-space boundary directly. Making use of the spatial context features of a traffic scene, multi-stage CNNs can regress a belief map with high resolution, accurate location and semantic information for the free-space boundary. A semantic free-space boundary can be finally produced through dynamic programming based on the belief map. Furthermore, the specially-designed training strategy ensures an efficient learning process. The experiments demonstrate our approach can achieve a convincing performance on various traffic scenes. For the future, we plan to apply our approach to image sequences to realize the robustness in those extremely challenging scenes.

Acknowledgement: This work was supported by the National Key RD Program of China (No.2016YFB1001001), the National Natural Science Foundation of China (No.91648121, No.61573280), and Tencent Robotics X Lab Rhino-Bird Joint Research Program (No.201902, No.201903).

5. REFERENCES

[1] Hernán Badino, Uwe Franke, and Rudolf Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*. Citeseer, 2007, vol. 20.

[2] Ali Harakeh, Daniel Asmar, and Elie Shammas, "Ground segmentation and occupancy grid generation using probability fields," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 695–702.

[3] Matthias Schreier and Volker Willert, "Robust free space detection in occupancy grid maps by methods of image analysis and dynamic b-spline contour tracking," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 514–521.

[4] Willem P Sanberg, Gijs Dubbelman, and Peter HN de With, "Extending the stixel world with online self-supervised color modeling for road-versus-obstacle segmentation," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 1400–1407.

[5] David Pfeiffer and Uwe Franke, "Modeling dynamic 3d environments by means of the stixel world," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 3, pp. 24–36, 2011.

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

[7] Friedrich Erbs, Alexander Barth, and Uwe Franke, "Moving vehicle detection by optimal segmentation of the dynamic stixel world," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 951–956.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[9] Willem P. Sanberg, Gijs Dubbelman, and Peter H. N. de With, "Free-space detection with self-supervised and online trained fully convolutional networks," *CoRR*, vol. abs/1604.02316, 2016.

[10] Dan Levi, Noa Garnett, Ethan Fetaya, and Israel Herzlyia, "Stixelnet: A deep convolutional network for obstacle detection and road segmentation.," in *BMVC*, 2015, pp. 109–1.

[11] Jian Yao, Srikumar Ramalingam, Yuichi Taguchi, Yohei Miki, and Raquel Urtasun, "Estimating drivable collision-free space from monocular video," in *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2015, pp. 420–427.

[12] Hui Kong, Jean-Yves Audibert, and Jean Ponce, "General road detection from a single image," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2211–2220, 2010.

[13] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1013–1020.

[14] Suvam Patra, Pranjal Maheshwari, Shashank Yadav, Subhashis Banerjee, and Chetan Arora, "A joint 3d-2d based method for free space detection on roads," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 643–652.

[15] Jose M Alvarez, Theo Gevers, Yann LeCun, and Antonio M Lopez, "Road scene segmentation from a single image," in *European Conference on Computer Vision*. Springer, 2012, pp. 376–389.

[16] Vijay John, Nithilan Meenakshi Karunakaran, Chunzhao Guo, Kiyosumi Kidono, and Seiichi Mita, "Free space, visible and missing lane marker estimation using the psinet and extra trees regression," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 189–194.

[17] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[18] Pedro F Felzenszwalb and Ramin Zabih, "Dynamic programming and graph algorithms in computer vision," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 4, pp. 721–740, 2011.

[19] Pedro F Felzenszwalb and Daniel P Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.