Robotics: Science and Systems 2019
Freiburg im Breisgau, June 22-26, 2019

Segment2Regress: Monocular 3D Vehicle Localization in Two Stages

Jaesung Choe, Kyungdon Joo, Francois Rameau, Gyumin Shim, In So Kweon
RCV Lab, KAIST
jaesung.choe@kaist.ac.kr, {kdjoo369, rameau.fr}@gmail.com, {shimgyumin, iskweon77}@kaist.ac.kr

Abstract*—High-quality depth information is required to perform 3D vehicle detection, consequently, there exists a large performance gap between camera and LiDAR-based approaches. In this paper, our monocular camera-based 3D vehicle localization method alleviates the dependency on high-quality depth maps by taking advantage of the commonly accepted assumption that the observed vehicles lie on the road surface. We propose a two-stage approach that consists of a segment network and a regression network, called *Segment2Regress*. For a given single RGB image and a prior 2D object detection bounding box, the two stages are as follows: 1) The segment network activates the pixels under the vehicle (modeled as four line segments and a quadrilateral representing the area beneath the vehicle projected on the image coordinate). These segments are trained to lie on the road plane such that our network does not require full depth estimation. Instead, the depth is directly approximated from the known ground plane parameters. 2) The regression network takes the segments fused with the plane depth to predict the 3D location of a car at the ground level. To stabilize the regression, we introduce a coupling loss that enforces structural constraints. The efficiency, accuracy, and robustness of the proposed technique are highlighted through a series of experiments and ablation assessments. These tests are conducted on the KITTI bird's eye view dataset where Segment2Regress demonstrates state-of-the-art performance. Further results are available at *https://github.com/LifeBeyondExpectations/Segment2Regress

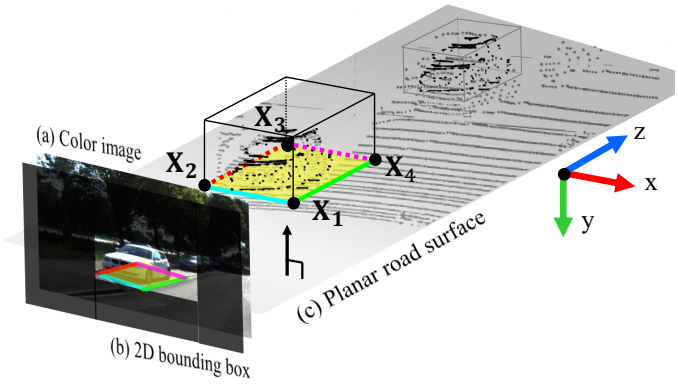


Fig. 1. **Overview.** Given (a) an RGB image and (b) a binary mask of a 2D bounding box, we localize a 3D vehicle under the road environment assumption. We segment the four line segments and a quadrilateral (vehicle segments $\mathcal{S}$) in image coordinate, where green, cyan, red, and magenta line segments indicate left, front, right, back line segments, respectively, and the yellow region describes the bottom quadrilateral. Then, we regress the four corners of the vehicle (vehicle points $\mathcal{X}$) from the vehicle segments fused with a compact depth estimated by (c) given plane parameters.

I. Introduction

Vehicle detection constitutes one of the fundamental elements required to analyze and understand dynamic road environments [7]. Recent LiDAR-based approaches have demonstrated reliable results by extracting semantic features from sparse point clouds [40, 39, 33]. In the context of intelligent vehicle technology, LiDAR, in particular, remains a privileged solution for its versatility (full 360°), accuracy ($\pm$5-15cm RMSE) and robustness (*e.g.* functional during day and night). In fact, its high-quality depth measurements facilitate the 3D vehicle localization. Despite all the advantages offered by LiDAR, such equipment is expensive, cumbersome to install and to calibrate, heavy and consumes a non-negligible quantity of energy. All these limitations make passive sensors, such as color cameras, very desirable to solve the task of 3D object localization in lieu of LiDAR. Therefore, a few attempts have been made to explore RGB-based 3D detection and localization [2, 1, 36, 25]. Relying on the advancement of the 2D object detectors [9, 13, 31, 22], camera-based 3D vehicle detection methods internally include the 2D object detector (usually two-stage 2D object detector [9]), and directly estimate vehicles' metric positions. Alternatively, segmentation information [2], shape retrieval [24] and explicit depth fusion [36] have been proposed, however, these works did not alleviate the depth dependency problem. The need for such depth maps is problematic for two major reasons. First, the computational time required to estimate the depth of a scene is a bottleneck for the entire 3D object detection module, while, for vehicular based application, the real-time aspect is critical. Second, high-quality depth map from a single image is a complex task which often requires additional data, such as LiDAR, RGB-D data [5], or stereo images [10, 23], to train the depth estimation network.

In this paper, we describe how to estimate the 3D positions of vehicles from a single RGB image (see Fig. 1). To alleviate the dependency on high-quality and high-computation depth (*scene depth*), we take advantage of the commonly admitted assumption that cars lie on a planar road surface. In this paper, we call this assumption the *road environment assumption*. Thus, our strategy includes localizing the vehicle at the ground level (*i.e.*, the bird's eye view). We utilize the depth from the known road plane, called the *plane depth* which is quickly estimated from a given planar equation that can be deduced using prior knowledge on the camera installation or via calibration with respect to the ground surface. Under these conditions, our

method achieves fast inference speed and obtains state-of-the-art performance in the KITTI bird's eye view benchmark [8]. Moreover, our method is robust to hard cases, such as occluded or truncated cars. To summarize, the key contributions of our work are as follows:

- Under the road environment assumption, we propose a two-stage approach for monocular 3D vehicle localization composed of the segment network and the regression network (called *Segment2Regress*), where we fuse the plane depth to regress the 3D metric location.
- We introduce the coupling loss, which jointly constrains the structural priors of the vehicle such as size, heading, and planarity.
- Our approach is validated through systematic experiments and an ablation study. Among monocular camera-based methods, our proposed two-stage approach achieves a state-of-the-art performance with a compact plane depth, not the full depth of the given scene.

II. Related Work

In this section, we briefly review the object detection approaches according to output types: 2D and 3D cases.

2D object detection The recent advances in deep-learning architectures [18, 12, 20] and the development of large-scale datasets [4, 19, 6] have enabled the success of the 2D object detection task. Existing 2D object detection approaches are categorized into two groups: single-stage and two-stage approaches. Two-stage 2D object detectors [9, 20, 13] are slower but demonstrate the higher performance in accuracy by differentiating the two steps; region proposal and classification, while single-stage approaches [31, 22, 21] incorporate the complicated steps into one straightforward network. In this work, we utilize a single-stage 2D object detector (we adapt YOLO [31]) to generate a 2D bounding box for a vehicle, which is the input of the proposed network. Note that any 2D object detector can be employed with our approach.

3D vehicle detection LiDAR-based 3D vehicle object detection methods have been extensively researched [3, 30, 40, 39]. Zhou *et al*. [40] introduce the voxel features to overcome the sparse point cloud. Yang *et al*. [39] focus on the bird's eye view 3D localization by projecting the 3D point clouds into accumulated planes. Recently, a few studies [3, 17, 30] suggest combining RGB images with LiDAR measurement. Specifically, Qi *et al*. [30] utilize the image-based 2D object detector to generate the frustum which minimized the search-space of the target object. Chen *et al*. [3] and Ku *et al*. [17] introduce a multi-view fusion approach to perform multi-modal feature fusion. Alongside the development of these LiDAR-based approaches, a few attempts for RGB-based 3D object detection are notable [2, 1, 25, 36]. Usually, RGB-based approaches [25, 2, 1] re-build the architecture based on the two-stage 2D object detector [9] to predict the location of cars in metric units. To overcome the lack of the depth information in an RGB image, Xu and Chen [36] propose a method that fused an RGB image and depth information, which is estimated by monocular depth estimation network [10]. However, estimation of the monocular depth information requires additional computation and the resulting accuracy is sensitive to the quality of the estimated depth. Following [36], we take advantage of depth information, but we utilize simplified depth information, plane depth under the road environment assumption. Thanks to the plane depth, we can decrease the dependency on the estimated scene depth and lessen the computational cost.

III. Segment2Regress for 3D Localization

In this section, we present the details of the proposed 3D vehicle localization approach which is composed of two networks: the segment network and the regression network. Under the road environment assumption, the segment network activates the pixels that correspond to the area under a vehicle in the image coordinate (Sec. III-A). We fuse the estimated segments with the plane depth to aid the estimation of the metric location of the car of interest (Sec. III-B). Then, the regression network predicts the 3D location of the vehicle (Sec. III-C). Thanks to this architecture, we avoid the scene depth estimation and localize the 3D vehicles at the ground level, *i.e*., bird's eye view localization, as shown in Fig. 2.

A. Segment network

The segment network takes an RGB image and a 2D bounding box of a target vehicle as input, where the 2D bounding box is pre-computed using any 2D object detector (YOLO [31] in this paper). Given this input data, the goal of the segment network is to estimate the area beneath the vehicle (the ground region occupied by a car) in the image domain. This bottom region usually forms a quadrilateral in the image domain and is expressed with a group of activated pixels (segments). Nevertheless, in this work, we estimate additional four line segments (left, front, right, and back line segments, see Fig. 2) alongside the bottom region to gain several advantages. Specifically, thanks to these additional four line segments, we can 1) estimate the heading of the vehicle with the line segments, 2) support the estimation of the bottom region via the observed line segments (typically two line segments are visible for truncated vehicles), and 3) disambiguate the physical attributes (the size) of the car, *i.e*., the width and the length. For simplicity, we refer to a set of segments including the bottom region and line segments as *vehicle segments* $\mathcal{S}=\{b_l, b_f, b_r, b_b, b_g\} = \{b_{(i)}\}_{i=1}^{5}$, where each segment $b_{(i)}$ follows left (l), front (f), right (r), back (b), and bottom ground (g) order.[1]

To estimate the vehicle segments, we design the segment network based on stacked encoder-decoder architectures [27, 28]. Specifically, we utilize stacked hourglass networks [28] as a base network, which shows superior performance on key-point estimation by refinement and noise filtering. For our target task, we stack four hourglass modules with two

[1]The index of $b_{(i)}$ directly maps to the output channel index of the segment network.

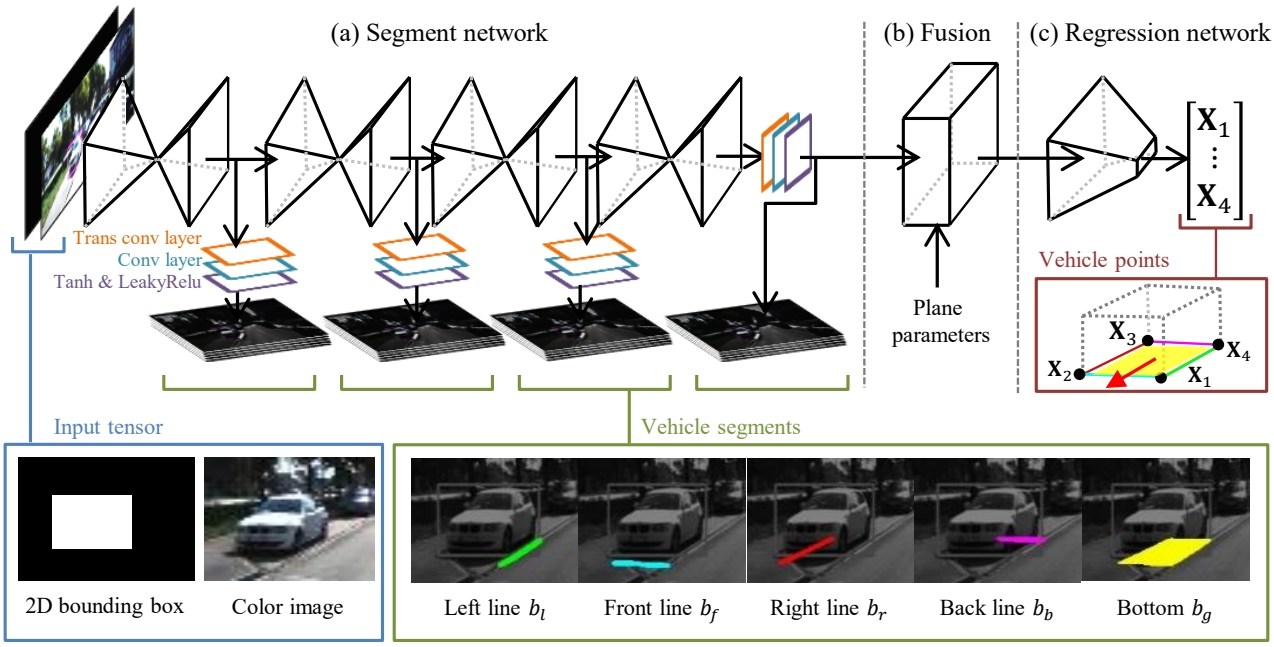


Fig. 2. **The overall architecture of the proposed Segment2Regress network.** Given an input tensor, (a) the segment network generates the vehicle segments $\mathcal{S}$ based on stacked hourglass networks [28]. (b) The estimated vehicle segments $\mathcal{S}$ and the plane depth computed by the plane parameters are combined by the fusion process. (c) The regression network predicts the four bottom corners of a vehicle in world coordinate, called vehicle points $\mathcal{X}$. For visualization purpose, we overlay a grayscale image with the colored vehicle segments, where green, cyan, red, magenta, and yellow indicate each vehicle segment, respectively.

additional layers (one transpose convolution and one convolution layers) at the end of each hourglass module to increase the resolution (four times higher than the output from the original hourglass network [28]), and generate sharp segments. In addition, we attach two activation functions (hyperbolic tangent function and Leaky Rectified Linear Units [37]) to estimate the confidence map (see Fig. 2(a)). With these modifications, the segment network filters the wrong vehicle segments consecutively, as shown in Fig. 7.

To train the segment network, we minimize the following L_2 loss:

$$L_{seg} = \sum_{k=1}^{4} \sum_{i=1}^{5} \left\| b_{(i)} - \widetilde{b}^{k}_{(i)} \right\|_2 , \tag{1}$$

where $\widetilde{b}^{k}_{(i)}$ is the i-th predicted vehicle segment from k-th hourglass network. The loss is computed in a pixel-wise manner.

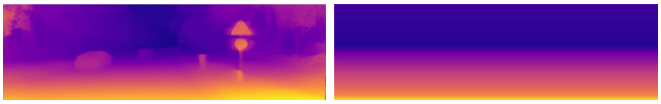
B. Fusion of plane depth

Recently, monocular 3D object detection has benefited from single image depth estimation techniques which significantly contributed to improved the accuracy [36]. To fully exploit the road environments assumption, we fuse the vehicle segments with an approximated depth map estimated from the road plane parameters (*i.e.*, plane depth) instead of relying on a highly accurate depth map (*i.e.*, scene depth) [10, 23] (see the example in Fig. 3). The plane parameters can be accurately estimated from geometric prior, *e.g.*, the elevation and the intrinsic parameters of the camera [11]. This strategy has the advantage of providing metric depth estimation faster than other depth estimation techniques.

To fuse the different data distribution, *i.e.*, vehicle segments and the plane depth, we introduce a fusion-by-normalization. First, we apply a batch normalization [15] to the vehicle segments, and then multiply them with the plane depth in a pixel-wise way. After the multiplication, we apply the instance normalization [35, 14] with learnable parameters (Fig. 4). Since batch normalization normalizes features along the mini batches, it maintains the instance-level responses (*i.e.*, vehicle segments). On the other hand, instance normalization normalizes each feature independently with the trainable parameters, so the plane depth is fused into each channel in an adaptive manner, like Nam *et al.* [26]. We validate the fusion-by-normalization method in our ablation study, Table II. All these steps are visualized in Fig. 4.

C. Regression network

After the fusion with the plane depth, the purpose of the regression network is to regress the 3D position of the observed vehicle, *i.e.*, 3D corners of the vehicle in metric units. We model these bottom corners as four 3D points $\mathcal{X} = \{\mathbf{X}_j\}_{j=1}^{4}$, where each point $\mathbf{X}_j = [x_j, y_j, z_j]^\top$ directly maps the absolute position of the vehicle (in the camera's referential), *e.g.*, the first corner $\mathbf{X}_1$ denotes the 3D intersection point between the left and front line segments. We refer to these four points $\mathcal{X}$ as *vehicle points*. Therefore, our regression network predicts a set of 12 variables which model the vehicle position. We



(a) Scene depth by Godard *et al.* [10] (b) Plane depth

Fig. 3. **Comparison of two different depth information**: (a) Scene depth estimated by single image depth estimation approach [10] and (b) plane depth computed by the given plane parameters, which is computationally cheaper than the scene depth estimation. We use an RGB image in the KITTI dataset [8] for inference in both cases.

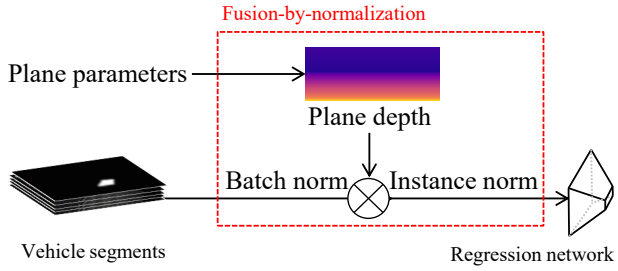


Fig. 4. **Illustration of fusion-by-normalization.** To facilitate the prediction of the metric locations in the following regression network, plane depth and the generated vehicle segments $\mathcal{S}$ — from the segment network — are combined in a normalized way.

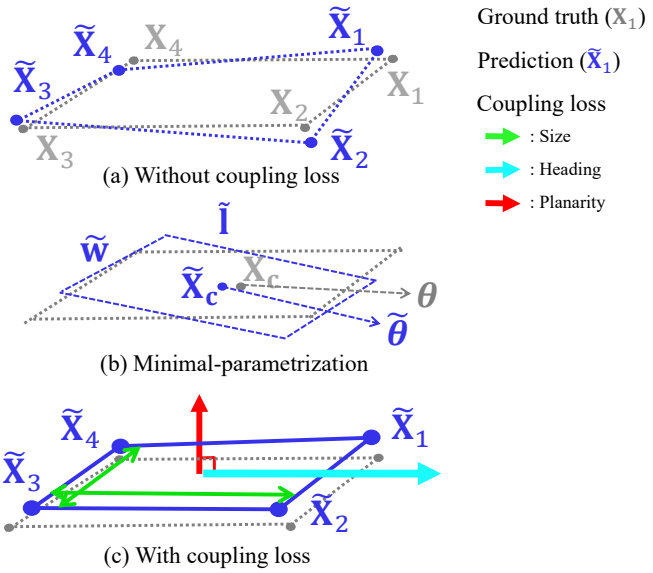


Fig. 5. **Illustration of coupling loss.** (a) Without coupling loss, the estimated vehicle points do not follow structural constraints. (b) The minimal-parametrization model (center $\mathbf{X}_c$, width w, length l, and orientation θ) suffers from local minimas [38]. (c) With coupling loss, we can regularize the vehicle points jointly while forcing structural conditions. We denote the ground truth of the vehicle point as $\mathbf{X}_j$ and prediction of the vehicle point as $\widetilde{\mathbf{X}}_j$.

intentionally employ such relaxed-parametrization instead of the more widely used minimal representation [3] (*e.g.*, center position, width, length, and rotation angle in a top-down view). This choice is motivated by recent studies, such as Xu *et al.* [38], where the authors underline the advantage of over-parametrization, *i.e.*, escape from spurious local optima. In addition, to enforce the structural constraints and to group the relaxed-regression variables, we introduce three losses respectively imposing the following geometric properties: size, heading, and planarity of vehicle. These constraints are combined through our *coupling loss* such that the relaxed-regression variables are coupled to other adjacent regression variables to satisfy the geometric properties. To estimate the vehicle points $\mathcal{X}$, we train our regression network such that it minimizes the absolute distance between the ground-truth 3D point $\mathbf{X}_j$ and its prediction $\widetilde{\mathbf{X}}_j$ as:

$$L_{reg} = \sum_{j=1}^{4} \left\| \mathbf{X}_j - \widetilde{\mathbf{X}}_j \right\|_1 + L_{couple}, \tag{2}$$

where $\|\cdot\|_1$ is the mean absolute error and L_{couple} indicates the coupling loss, which incorporates a set of structural constraints between points to ensure a shape-aware estimation of the 3D position of vehicles.

Coupling loss To apply the structural prior of the vehicle, we introduce a coupling loss, which consists of three constraints related to size, heading, and planarity of the vehicle (see Fig. 5).

1) Size loss. First, we exploit the size of the vehicle to jointly regularize adjacent vehicle points. In the size loss, we measure the size (width and length) at each vehicle point $\mathbf{X}_j$ and minimize the absolute distance as:

$$\begin{aligned} L^j_{size} = &\left| d(\mathbf{X}_j, \mathbf{X}_{j+1}) - d(\widetilde{\mathbf{X}}_j, \widetilde{\mathbf{X}}_{j+1}) \right| \\ &+ \left| d(\mathbf{X}_j, \mathbf{X}_{j-1}) - d(\widetilde{\mathbf{X}}_j, \widetilde{\mathbf{X}}_{j-1}) \right|, \end{aligned} \tag{3}$$

where $d(\cdot,\cdot)$ is the Euclidean distance between the two adjacent points[2] and $|\cdot|$ is the absolute value.

2) Heading loss. Second, the heading of vehicle is another important structural information because it can distinguish the front/rear (width) and left/right (length). The pair of adjacent points are related to the direction of the cars. Therefore, the heading constraint can be applied through the following formulation:

$$\begin{aligned} L^j_{head} = &\left| f(\overrightarrow{\mathbf{X}_j\mathbf{X}_{j+1}},\ \mathbf{e}_1) - f(\overrightarrow{\widetilde{\mathbf{X}}_j\widetilde{\mathbf{X}}_{j+1}},\ \mathbf{e}_1) \right| \\ &+ \left| f(\overrightarrow{\mathbf{X}_j\mathbf{X}_{j-1}},\ \mathbf{e}_3) - f(\overrightarrow{\widetilde{\mathbf{X}}_j\widetilde{\mathbf{X}}_{j-1}},\ \mathbf{e}_3) \right|, \end{aligned} \tag{4}$$

where $\overrightarrow{\mathbf{X}_j\mathbf{X}_{j+1}}$ is the unit direction vector from $\mathbf{X}_j$ to $\mathbf{X}_{j+1}$ and $f(\cdot,\cdot)$ denotes the cosine similarity between two vectors. The two vectors $\mathbf{e}_1{=}[1,0,0]^\top$ and $\mathbf{e}_3{=}[0,0,1]^\top$ are the x and z axes in the camera's referential, respectively.

3) Plane loss. Last, we impose the planarity loss to ensure the vehicle is located on the ground plane. In practice, the ground-truth value provided by the KITTI dataset [8] does not provide points lying exactly on a single ground plane.

[2]In the coupling loss, we utilize the mod operation to compute the two adjacent points in practice.

Therefore, we rectify the ground-truth to respect this constraint by adjusting their vertical positions (y-axis) according to the provided parameters of the plane from the previous study [17]. Considering the normal of the plane $\mathbf{n}$ and the elevation of the camera to the road surface d, the plane loss can be formulated as:

$$L^j_{plane} = \left| \mathbf{n}^\top \cdot \mathbf{X}_j + d \right|. \tag{5}$$

Using the above three loss functions related to the structural relationship, we define the coupling loss as follows:

$$L_{couple} = \sum_{j=1}^{4} \alpha L^j_{size} + \beta L^j_{head} + \gamma L^j_{plane}, \tag{6}$$

where each of the terms in the coupling loss is weighted by the balancing parameters α, β, and γ. We set α, β, and γ as 0.01, 1.0 and 1.0, respectively. We discuss the effect of the proposed coupling loss in an ablation study (see Table III).

IV. Experiments

In this section, we evaluate the proposed approach on the KITTI object detection benchmark (bird's eye view) [8]. First, we present a comparison of our method with recent state-of-the-art techniques, this assessment underlines the robustness and efficiency of Segment2Regress under challenging conditions. In addition, we conduct a systematic ablation study to validate the factors of coupling loss, fusion-by-normalization, and road plane assumption.

A. Implementation details

The size of the input tensor is $4 \times 256 \times 512$ pixels (C×H×W) which contains an RGB image and the binary mask of a given 2D car detection bounding box. With this input tensor, the segment network estimates the vehicle segments which have a shape of $5 \times 128 \times 256$ pixels. Then, the estimated vehicle segments are combined with the plane depth by the fusion-by-normalization process. It should be noted that this process preserves the shape of the original segment network output. Concerning the regression network, we use ResNet101 [12] with some slight adjustments. At the first layer, we add a dropout layer [34] and set the drop ratio to 0.1 for training. At the last layer, we configure 12 output parameters that represent the 3D location of the four corners of a car (vehicle points $\mathcal{X}$). The two networks (the segment network and the regression network) are trained independently. We do not load the pre-trained weight for the two networks. The learning rate is initiated at 0.0002 and decreases to 10 times smaller at 20 epochs and 5 times smaller at 40 epochs. This training stage is performed with an Adam optimizer [16] with batch size of 32. The training ends after 50 epochs for each training phase. We use three GPUs (GeForce GTX 1080 Ti) for training. The proposed method processes each vehicle of interest individually such that custom local planes can be utilized for each object to improve the localization accuracy. Moreover, with a single GPU, we can process 10 objects simultaneously (available batch size) without an additional computational load. In practice, an RGB image acquired in a road environment captures the limited number of cars. For example, the KITTI dataset contains less than 10 vehicles on average. Thus, we consider the computational time for 10 vehicles per frame, *i.e.*, FPS.

B. Evaluation

Dataset and metric The KITTI dataset [8] is a real-world public dataset captured for various traffic scenarios (highway and city scenes) for research in robotics and computer vision fields. This dataset also includes a variety of modalities: stereo cameras, a 3D Velodyne and GPS/IMU. It also provides online benchmarks for stereo, optical flow, object detection, and other tasks.

In this dataset, we exploit the bird's eye view benchmark that measures the 3D object detection on the top-down view, *i.e.*, exclude the vertical components (y-axis). It should be noted that the ground truth of 3D bounding boxes is not annotated to precisely respect the ground plane assumption. These rather inaccurate annotations violate our hypothesis, therefore, we adjust the ground truth data such that the vehicle points $\mathcal{X}$ lie on a single road plane via a projection along the y-axis.[3]

All the results presented in this paper strictly follow the KITTI's official metrics. For example, we exclusively consider cars while omitting trucks or buses — as stated in [8]. The dataset is divided into three cases: easy, moderate, and hard. We measure the Average Precision (AP) of the bird's eye view bounding box for each case, where we set the IoU thresholds as 0.7.

Comparison We compare our approach with recent monocular camera-based methods [2, 1, 36, 24, 32]. The resulting scores are summarized in Table I. Due to limited information provided by a single RGB image (absence of depth), most approaches suffer from limited performance. To cope with this problem, Xu and Chen [36] presented a multi-fusion method that incorporates depth information estimated by the single depth estimation approach [10]. This strategy improved the performance significantly but remains sensitive to hard cases. On the other hand, our proposed approach shows promising performances for every difficult category (see Fig. 8). We attribute this robustness to our ground plane based depth approximation which cannot be affected by a clutter environment or occlusions — because it is directly derived from the plane equation. Indeed, when the cars are occluded or truncated, the localization uncertainty of unknown points increases. In our method, the plane depth regularizes the estimated vehicle points at the ground level. Furthermore, our coupling loss further enforces structural priors jointly, and improves the robustness to hard cases. Qualitative results on the KITTI dataset are available in Fig. 6.

In our self-validation, we challenge our method with various types of bounding box inputs: ground truth with/without noise and estimation from another 2D object detection network. For

[3]Since we modify the ground truth along the y-axis, it does not affect the bird's eye view metric.

Method	Modality	Speed (FPS)	Car 2D AP IoU=0.7 [val/test]			Car BEV AP IOU=0.7 [val]		
			Easy	Moderate	Hard	Easy	Moderate	Hard
Mono3d [2]	Mono	3	93.89 / 92.33	88.67 / 88.66	79.68 / 78.96	5.22	5.19	4.13
3DOP [1]	Stereo	0.83	93.08 / 90.09	88.07 / 88.34	79.39 / 78.79	12.63	9.49	7.59
Multi-Fusion [36]	Mono	10.52**	- / -	- / -	- / -	11.14	6.59	5.43
	Mono+Depth [10]	8	- / 90.43	- / 87.33	- / 76.78	22.03	13.63	11.60
ROI-10D [24]	Mono+Depth [29]	5	85.32 / 75.33	77.32 / 69.64	69.70 / 61.18	14.76	9.55	7.57
OFT-NET [32]	Mono	2	- / -	- / -	- / -	11.06	8.79	8.91
Ours + GT 2D BBox	Mono	60*	100 / -	100 / -	100 / -	**22.73**	17.31	16.87
Ours + GT 2D BBox (noise)	Mono	60*	62.59 / -	55.16 / -	42.72 / -	22.61	**17.40**	**16.89**
Ours + YOLO [31]***	Mono	15	63.18 / -	55.59 / -	42.99 / -	19.21	15.35	14.51

TABLE I

3D VEHICLE LOCALIZATION. AVERAGE PRECISION (AP) OF BIRD'S EYE VIEW BENCHMARK ON KITTI DATASET [8]. WE DESCRIBE THE MODALITY AND THE SPEED FROM OTHER METHODS AND ADDITIONALLY PROVIDE THE 2D AP ON KITTI VAL/TEST DATASET [8]. OFT-NET [32] AND OUR SEGMENT2REGRESS DOES NOT PREDICT THE 2D BOUNDING BOXES. * MEANS THE PURE INFERENCE TIME OF OUR NETWORK. ** INDICATES THE EXPECTED SPEED. WE DID NOT FINE-TUNE YOLO [31]*** BUT FILTERED OUT THE FALSE-POSITIVE 2D VEHICLE PREDICTIONS TO MEASURE THE PURE ACCURACY OF 3D VEHICLE LOCALIZATION.

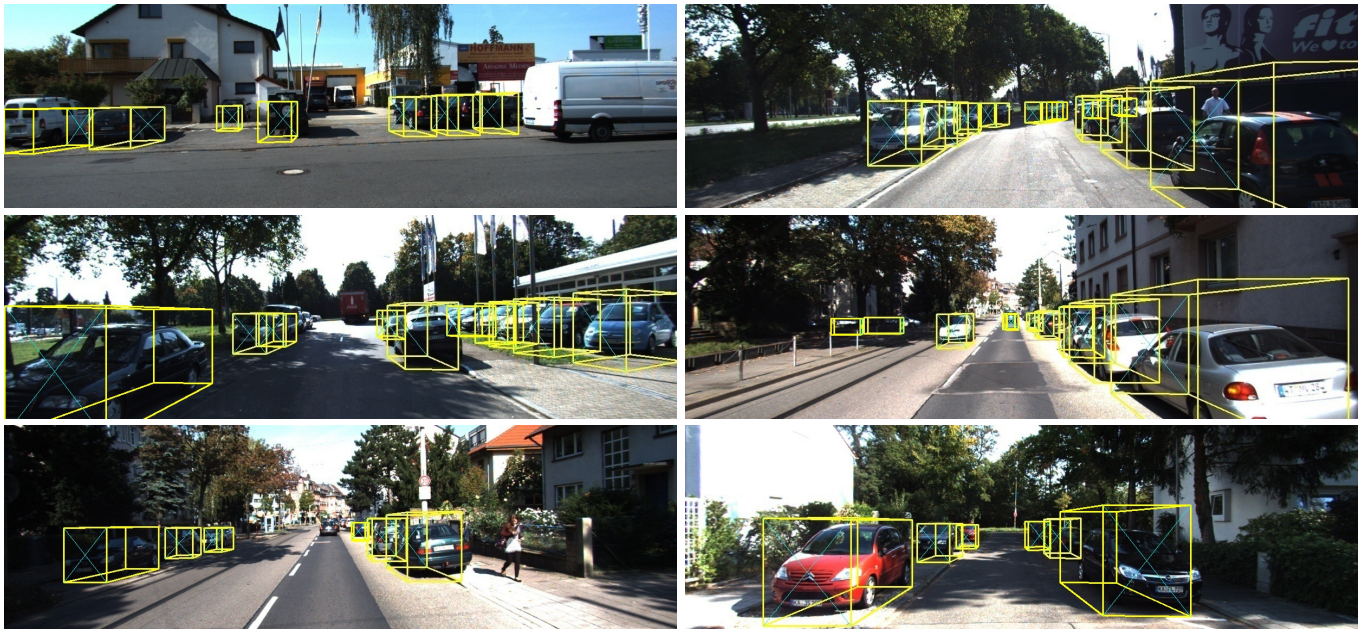


Fig. 6. **Qualitative results of the proposed Segment2Regress.** To validate the performance, we utilize the KITTI raw dataset [8], which only provides RGB images. We obtain the 2D bounding boxes from YOLO [31] and calculate the static plane from the sensor setup of KITTI [8]. With the given input data, we infer the vehicle points of the target objects through our Segment2Regress network. The yellow cube is the prediction by our network (for visualization, we set the height of cars as 2 meters) and cyan represents the front phase (heading) of the vehicles. We follow the KITTI's official rules [8] and only consider the single class, *i.e.*, cars, not vans or trucks.

the noise case, we randomly translated the ground truth up to 20 pixels under a uniform distribution. As for the network case, we use YOLO [31] to estimate the 2D bounding boxes. It is worth mentioning that YOLO [31] was pre-trained on COCO dataset [19] without fine-tuning on the KITTI 2D object benchmark [8]. Thus, it performed less accurately than the other fine-tuned networks [2, 36, 1, 24] on 2D detection, as shown in Table I. Nonetheless, our Segment2Regress network still demonstrates higher accuracy under hard cases, which highlights the robustness of the proposed approach against input 2D bounding boxes. On the other hand, other approaches [2, 36, 1, 24] internally include a two-stage 2D object detector network (Faster-RCNN [9]) and simultaneously fine-tune the whole system from the ground truth of the 2D/3D bounding boxes. Thus, they show higher accuracy in the KITTI 2D object detection benchmark.

C. Ablation study

In this section, we describe our extensive ablation study to confirm our three contributions that mainly increase the performance of our method. First, we analyze the effect of the fusion-by-normalization. Second, we evaluate the influence of the coupling loss. Third, we assess the robustness to the plane depth accuracy. The metrics in Tables II, III, and IV are the Average Precision (AP) of the bird's eye view benchmark on the KITTI validation dataset [8]. Since the contributions are related to both regression and fusion, the prediction is purely based on the regression network and the fusion-by-normalization process. Using the ground truth of the vehicle

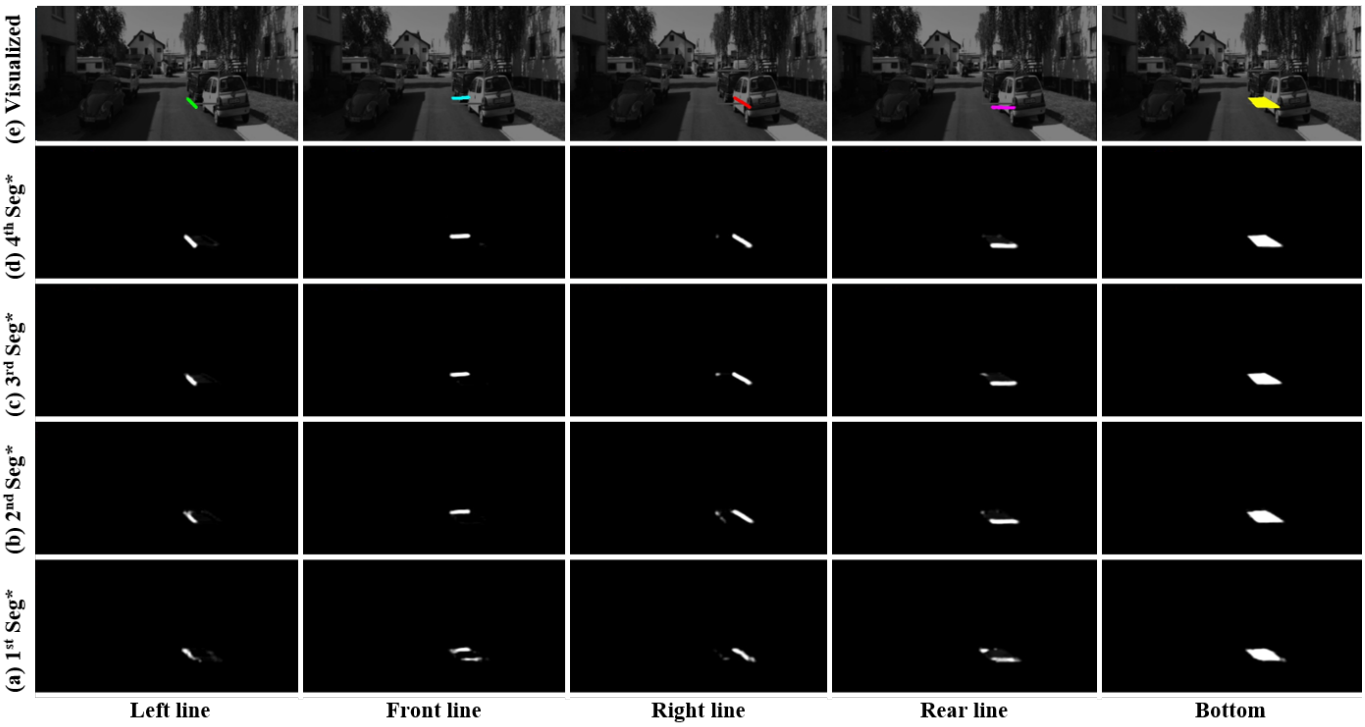


Fig. 7. **Visualization of vehicle segments from each hourglass network in segment network.** Fig. 7-(a) gives the results from the first hourglass network, Fig. 7-(b), Fig. 7-(c), and Fig. 7-(d) show the results from the following hourglass modules. The consecutive process filters out the wrongly segmented lines. Fig. 7-(e) is the visualization of the final estimation Fig. 7-(d), which is overlaid with an RGB image.

Coupling loss	Batch norm*	Plane depth	Inst norm**	Car BEV AP IoU=0.7 [val]		
				Easy	Moderate	Hard
✓				5.40	4.92	6.05
✓	✓			5.29	4.78	5.97
✓	✓	✓		0.99	1.16	1.46
✓	✓		✓	5.21	4.75	5.88
✓		✓	✓	13.87	10.75	12.88
✓	✓	✓	✓	**45.70**	**34.48**	**39.32**

TABLE II

ANALYSIS OF FUSION-BY-NORMALIZATION. WE INTENTIONALLY OMIT THE COMPONENTS IN THE FUSION-BY-NORMALIZATION PROCESS TO VERIFY EACH STEP. THE MARKS MEAN THAT WE APPLY THE CORRESPONDING CONDITION. BATCH NORMALIZATION LAYER [15] AND INSTANCE NORMALIZATION LAYER [35] ARE ABBREVIATED TO BATCH NORM* AND INST NORM**, RESPECTIVELY.

Fusion	Coupling loss			Car BEV AP IoU=0.7 [val]		
	Size	Heading	Planarity	Easy	Moderate	Hard
✓				29.75	22.41	21.80
✓	✓			34.96	26.32	30.12
✓		✓		35.70	31.45	31.02
✓			✓	42.63	33.20	37.95
✓	✓	✓		41.56	**35.05**	31.56
✓	✓	✓	✓	**45.70**	34.48	**39.32**

TABLE III

EVALUATION OF THE COUPLING LOSS. WE ACHIEVE THE UPPER-BOUND ACCURACY OF OUR METHOD WHEN WE TAKE INTO ACCOUNT THE ALL ELEMENTS OF COUPLING LOSS. MARK MEANS THAT WE APPLY THE CORRESPONDING CONDITION.

A static plane	Estimated planes [17]	Car BEV AP IoU=0.7 [val]		
		Easy	Moderate	Hard
		5.21	4.75	5.88
✓		38.60	33.83	39.03
	✓	**45.70**	**34.48**	**39.32**

TABLE IV

INFLUENCE OF PLANE DEPTH. WE ADDRESS THAT FUSION OF PLANE DEPTH IS NECESSARY FOR THE METRIC PREDICTION FROM REGRESSION NETWORK. MARK MEANS THAT WE APPLY THE CORRESPONDING CONDITION.

segments, we re-train the regression network and fusion-by-normalization process. The obtained metric becomes the upper-bound performance of our Segment2Regress network.

Fusion process We present a case study for the proposed fusion-by-normalization method. This method is effective for fusing vehicle segments and plane depth. For the comparison, we apply the coupling loss to all experiments, and change the components in Fig. 4. Table II shows the existence of plane depth itself cannot leverage the performance, but the association of two normalization layers (batch normalization [15] and instance normalization [35]) increase the performance significantly.

Coupling loss To highlight the relevance of the coupling loss, we test various combinations involving the different elements of the coupling loss. The results obtained through this ablation study are provided in Table III. Based on the fusion-by-normalization method, we train the regression network with

(a) Truncated sample **(b) Side-occluded sample** **(c) Rear-occluded sample** **(d) Far located sample**

Fig. 8. **Sampled 3D vehicle localization results for hard cases in the KITTI dataset:** (a) Truncated, (b) side-occluded, (c) rear-occluded, and (d) far located samples. The top row shows the predicted result (projection of the predicted 3D bounding box) in the image domain, where we denote the prediction by the proposed approach as a yellow cube with a cyan-colored front face, and the input 2D bounding box as a magenta box. The bottom row describes the corresponding results in the bird's eye view, where the red quadrilateral represents the prediction from our Segment2Regress, the green one is estimated purely from the regression network using the ground truth of vehicle segments, and the blue one is the ground truth of the 3D vehicle localization.

different variations of the coupling loss. From this test, we notice that the plane loss is the most effective to regularize the regression variables. When we apply the all elements of the coupling loss, the accuracy increases further. In other words, our coupling loss effectively regularizes the relaxed-regression variables by grouping the adjacent regression variables while considering its geometric properties: size, heading, and planarity of the vehicle.

plane parameters We demonstrate the influence of the plane parameters in Table IV. The static plane means that we calculate the static plane parameters from the sensor setup of the KITTI dataset [8], such as the elevation of the camera and the calibration parameters. In other words, the static plane is identical throughout all the different images. Even from these rough static plane parameters, the regression network performs better than when it does not utilize plane depth. When we achieve the more accurate road equations, the performance of the regression network increases further. The accuracy of plane parameters affects our vehicle localization results, and even the fusion of static plane parameters outperforms the non-fusion experiment.

V. Conclusion

We have presented a novel approach for 3D vehicle localization from a single RGB image and plane parameters. To fully exploit the road environment assumption (vehicles lie on the road surface), we formulate the 3D vehicle localization as two sub-tasks (two stages): 1) Segment the vehicle region in the image domain (segment network) and 2) regress the vehicle points in the 3D domain (regression network), where we newly introduce a coupling loss to enforce the structure and heading of the vehicles. In addition, we estimate the 3D vehicle localization in metric units through a fusion-by-normalization approach with the plane depth, which can be computed from simple plane parameters without heavy computation. We successfully validated our method on the bird's eye view KITTI dataset and by an ablation study. The proposed approach can be considered as an independent 3D localization module applicable to any 2D object detector.

Acknowledgments

This research was partially supported by the Shared Sensing for Cooperative Cars Project funded by Bosch (China) Investment Ltd. This work was also partially supported by Korea Research Fellowship Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2015H1D3A1066564).

References

[1] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2015.

[2] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016.

[3] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *IEEE CVPR*, volume 1, page 3, 2017.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[5] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[7] Andreas Geiger. *Probabilistic models for 3D urban scene understanding from movable platforms*, volume 25. KIT Scientific Publishing, 2013.

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237, 2013.

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[10] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017.

[11] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[14] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR, abs/1703.06868*, 2:3, 2017.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

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

[17] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3d proposal generation and object detection from view aggregation. *arXiv preprint arXiv:1712.02294*, 2017.

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[20] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017.

[21] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[23] Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 155–163, 2018.

[24] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. *arXiv preprint arXiv:1812.02781*, 2018.

[25] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Košecká. 3d bounding box estimation using deep learning and geometry. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5632–5640. IEEE, 2017.

[26] Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *Advances in Neural Information Processing Systems*, pages 2558–2567, 2018.

[27] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[28] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017.

[29] Sudeep Pillai, Rares Ambrus, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. *arXiv preprint arXiv:1810.01849*, 2018.

[30] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. *arXiv preprint arXiv:1711.08488*, 2017.

[31] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[32] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018.

[33] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. *arXiv preprint arXiv:1812.04244*, 2018.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[35] Dmitry Ulyanov, Andrea Vedaldi, and Victor S Lempitsky. Improved texture networks: Maximizing quality

and diversity in feed-forward stylization and texture synthesis. In *CVPR*, volume 1, page 3, 2017.

[36] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2345–2353, 2018.

[37] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[38] Ji Xu, Daniel Hsu, and Arian Maleki. Benefits of over-parameterization with EM. *CoRR*, abs/1810.11344, 2018. URL http://arxiv.org/abs/1810.11344.

[39] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[40] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *arXiv preprint arXiv:1711.06396*, 2017.