2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China



# Ground-aware Monocular 3D Object Detection for Autonomous Driving

Yuxuan Liu[1], Yuan Yixuan[2], Lujia Wang[3] and Ming Liu[1]

***Abstract*— Estimating the 3D position and orientation of objects in the environment with a single RGB camera is a critical and challenging task for low-cost urban autonomous driving and mobile robots. Most of the existing algorithms are based on the geometric constraints in 2D-3D correspondence, which stems from generic 6D object pose estimation. We first identify how the ground plane provides additional clues in depth reasoning in 3D detection in driving scenes. Based on this observation, we then improve the processing of 3D anchors and introduce a novel neural network module to fully utilize such application-specific priors in the framework of deep learning. Finally, we introduce an efficient neural network embedded with the proposed module for 3D object detection. We further verify the power of the proposed module with a neural network designed for monocular depth prediction. The two proposed networks achieve state-of-the-art performances on the KITTI 3D object detection and depth prediction benchmarks, respectively. The code will be published in https://www.github.com/Owen-Liuyuxuan/visualDet3D**

## I. Introduction

Simultaneously estimating the position, orientation, and dimensions of an object in 3D with a single well-calibrated RGB camera image in an autonomous driving scene is generally an ill-posed problem. Lidar-based methods and stereo-vision-based methods, which respectively obtain depths and distance information from lidar measurements and triangulation, can achieve superior performance [1][2][3][4]. Monocular setups are cheaper and more versatile than LiDAR setups and are more robust to variations in extrinsic parameters than stereo cameras. As a result, 3D detection with a single camera is still a heated research direction despite a lack of depth information.

Recent developments in monocular 3D object detection mainly utilize geometric constraints between the 3D object and the its projection on a 2D image. ShiftRCNN [5], SS3D [6], and RTM3D [7] optimize the estimation of depth and orientation by solving a Perspective-n-Point problem with noisy observation.

*This work was supported by the National Natural Science Foundation of China (Grant No. U1713211), the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017, and No. 21202816, and Shenzhen Science, Technology and Innovation Comission (SZSTI) JCYJ20160428154842603, awarded to Prof. Ming Liu. And it was supported by the Guangdong Science and Technology Plan Guangdong-Hong Kong Cooperation Innovation Platform (Grant Number 2018B050502009) awarded to Lujia Wang. (Lujia Wang is the corresponding author).

[1]Yuxuan Liu and Ming Liu are with the Robotics and Multi-Perception Laborotary, Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology yliuhb@connect.ust.hk ,eelium@ust.hk

[2]Yuan Yixuan is with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China. yxyuan.ee@cityu.edu.hk

[3]Lujia Wang is with Cloud Computing Lab of Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. lj.wangl@siat.ac.cn

Most of these ideas come from a more general problem of monocular 6D pose estimation. Monocular 6D pose estimation benchmarks like LINEMOD [8] are based on the assumption that the CAD models of the objects of interest are known. However, we do not have access to accurate car models for each vehicle in autonomous driving scenes; thus, the performances of monocular 3D object detectors in autonomous driving scenes are limited.

In autonomous driving and mobile robotics applications, we can generally assume that most important dynamic objects are on a ground plane, and the camera is mounted at a certain height above the ground. Some traditional depth prediction methods also note the importance of ground planes and introduce a similar "floor-wall" assumption for indoor environments [9], [10]. Such perspective priors on the ground plane, not presented in *general* monocular 6D pose estimation problems, provide a significant amount of information for geometric reasoning for monocular 3D object detection in driving scenes. Few recent works *explicitly* inject the perspective priors on the ground plane into a neural network.

This paper proposes two novel procedures to allow a monocular object detector to reason over the ground plane explicitly.

The first procedure is anchor filtering, where we explicitly break the invariance in the neural network predictions. Given a prior distance between an anchor and its distance to the camera, we back-project the anchor to 3D. Since all objects of interest are located around the ground plane, we filter out 3D anchors far from the ground plane during training and testing. This operation focuses the network on positions where objects of interest are likely to appear. We will further introduce this procedure in Section III-A.

The second procedure is a ground-aware convolution module. The motivation of this module is illustrated by Figure 1. For a human, ground pixels around a car are useful to estimate the car's 3D position, orientation, and dimensions. For an anchor-based detector, features at the center are responsible for estimating all the car's 3D parameters. However, to infer depth with ground pixels like a human, the network model needs to perform the following steps from the center of an object (e.g. the red dot in the figure)

1) identifing the contact points of the object and the ground plane (e.g. the blue curve beneath the car).
2) computing the 3D position of the contact points with perspective geometry.
3) gathering information from these contact points with a receptive field focusing downwards.

A standard object detection or depth prediction network is built to have a uniform receptive field, and neither perspective

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

Fig. 1: Contact points with the ground plane are important in inferreing 3D information of an object. Predicting depths of background pixels (e.g., the brown point) also rely on the geometry of the ground plane. Best viewed in color.

geometry priors nor camera parameters are provided to the network. Thus, it is non-trivial to train a standard neural network to make inferences like a human.

The ground-aware convolution module is designed to guide the network to incorporate the ground-based reasoning in network inferencing. We encode each pixel point's prior depth value as an additional feature map, and we guide each pixel point of the feature map to incorporate features from pixels below them. Details of this module will be introduced in Section III-B.

Incorporating the two proposed procedures into the network, we propose a one-stage framework with explicit ground plane hypothesis usage. The network is fast thanks to its clean structure and can run at about 20 frames-per-second (FPS) on a modern GPU.

We further incorporate the ground-aware convolution module in a u-net-based structure on monocular depth prediction. Both networks achieve state-of-the-art (SOTA) performance on the KITTI dataset.

The contribution of the paper is three-fold.

- We identify the benefit of learning from the ground plane priors in urban scenes for 3D reasoning from images.
- We introduce a processing method and a ground-aware convolution module in monocular 3D object detection to use the ground plane hypothesis.
- We evaluate the proposed module and design methods on the KITTI 3D object detection benchmark and the depth prediction benchmark, and we achieve competitive results.

## II. Related Works

### A. Pseudo-LiDAR for Monocular 3D Object Detection

The idea of pseudo-LiDAR, reconstructing point clouds from mono or stereo images, has led to the recent advances in 3D detection [11][12][13][14][15]. Pseudo-LiDAR methods usually reconstruct the point cloud from a single RGB image with off-the-shelf depth prediction networks, which limit their performance. Moreover, the current SOTA monocular depth prediction networks generally take about 0.05s per frame, which significantly limits the inference speed of pseudo-lidar detection pipelines.

### B. One-Stage Detection for Monocular 3D Object Detection

Several recent advances in monocular 3D object detection directly regress 3D bounding boxes in a one-stage object detection framework.

*Optimization-based Methods:* SS3D [6] concurrently estimated 2D bounding boxes, depth, orientation, dimensions, and 3D corners. Nonlinear optimization was applied to merge all these predictions. Shift-RCNN[5] also estimated 3D information in a 2D anchor and applied a small subnetwork instead of a nonlinear solver. More recent methods, SMOKE[16] and RTM3D [7] incoperate the aforementioned optimization scheme into the anchor-free object detector CenterNet [17].

*3D Anchor-based Methods:* M3D-RPN [18] introduced 3D priors in 2D anchors, and also emphasized the importance of the ground plane hypothesis. It also introduced height-wise convolution while D4LCN [19] introduced depth-guided convolution. Both techniques came at a high cost to efficiency and only utilized the ground plane hypothesis implicitly.

We point out that anchor-based methods are still better than anchor-free methods in 3D detection. Anchor-free detectors implicitly require the network to learn the correlation between the object's apparent size and its distance value. In contrast, anchor-based detectors can embed this in an anchor's preprocessing. As a result, we develop our framework upon anchor-based detectors.

To our knowledge, our proposed framework is the first 3D anchor-based method to explicitly utilize the ground plane hypothesis of driving scenes in monocular 3D detection and achieves the SOTA performance at the time of writing.

### C. Supervised Monocular Depth Prediction With Deep Learning

Supervised monocular depth prediction is another hot research topic closely related to monocular 3D object detection.

DORN [20] and SoftDorn [21] proposed to treat the depth estimation problem as an ordinal regression problem to improve the convergence rate. BTS [22] proposed the local planar guidance module and incorporated normal information to constraint the depth prediction results in the scenes. BANet [23], meanwhile, proposed a bidirectional attention network to improve the receptive fields and global information understanding of depth prediction networks.

Many of the methods above focus on depth prediction for multiple datasets and scenarios. Images in datasets like NYUv2[24] and DIODE[25] are taken from various viewpoints, and it is hard to extract floor priors, unlike the cases in driving scenes. As a result, the neural networks mentioned above do not utilize the camera's extrinsic parameters to extract environment priors, and the absolute scale is lacking during the network inference process.

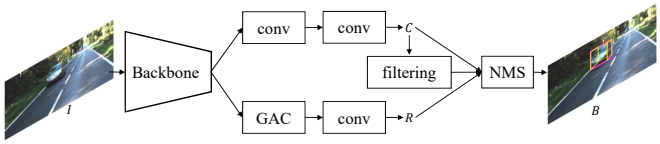


Fig. 2: Network structure for 3D object detection. We extract features from image $I$ and predict classfication tensor $C$ and regression tensor $R$. We filter anchors far from the ground before post-processing and produce the final bounding boxes $B$.

## III. Methods

In this section, we elaborate on the methods applied in this paper. First, we present the formulation of the detection network's inference results and the data preprocessing procedure. Second, we introduce the ground-aware convolution module that extracts depth priors from the ground plane hypothesis. Finally, we present the network's architecture with other major modifications in the training and inferencing process.

### A. Anchors Preprocessing

*1) Anchors Definition:* We follow the idea from YOLO [26] to densely predict bounding boxes with dense anchors. Each anchor on the image also acts as a proposal of an object in 3D. A 3D anchor consists of a 2D bounding box parameterized by $[x, y, w_{2d}, h_{2d}]$, where $(x, y)$ is the center of the 2D box and $(w_{2d}, h_{2d})$ is the width and height; 3D centers of an object are presented as $[cx, cy, z]$, where $(cx, cy)$ is the center of the object projected on the image plane and $z$ is the depth; $[w_{3d}, h_{3d}, l_{3d}]$ corresponds to the width, height and length of the 3D bounding box, and $[sin(\alpha), cos(\alpha)]$ is the sine and cosine value of the observation angle $\alpha$.

*2) Priors Extraction from Anchors:* The shape and size of an anchor or an object are highly correlated with the depth. In some prior methods [18], the mean of the depth is computed for each pre-defined anchor box, while variance is computed globally instead. The global variance is only computed to normalize the targets for the neural net.

We further observe that the variance of the depth $z$ of an anchor is inversely proportional to the object's size in the image. Thus, we consider each anchor as a distribution with individual mean and variance of the object proposal in 3D. To collect prior statistical knowledge in the anchors, we iterate through the training set and collect all objects sharing a large intersection-over-union (IoU) with the box for each anchor box with a different shape. Then we calculate the mean and variance of the depth $z$, $sin(\alpha)$ and $cos(\alpha)$ for each pre-defined anchor box. We can significantly lower the prior variance of the depth $z$ for large anchor boxes / close objects.

Since we have considered anchors as distribution of 3D proposals, the associated 3D targets should not deviate much from the expectation. We utilize the fact that most objects of interest should be on the ground plane. Each anchor, centering at $(u, v)$ with pre-computed mean depth $\hat{z}$, can be back-projected to 3D:

$$x_{3d} = \frac{u - c_x}{f_x}\hat{z} \qquad y_{3d} = \frac{v - c_y}{f_y}\hat{z}, \tag{1}$$

where $(c_x, c_y)$ is the camera's principal point and $(f_x, f_y)$ is the camera's focal length. Anchors with $y_{3d}$ too far from the ground will be filtered out from training and testing. Such a strategy allows the network to train with 3D anchors around the region of interest and simplify the classification problem.

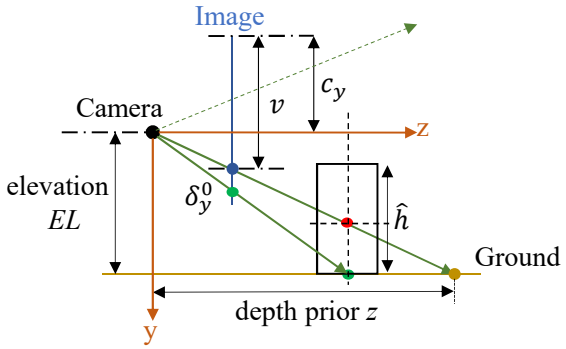


Fig. 3: Perspective geometry for the GAC module. When we calculate the vertical offsets $\delta_y^0$, we assume pixels are foreground object centers. When we compute the depth priors $z$, we assume pixels are on the ground because they are features to be queried.

### B. Ground-Aware Convolution Module

Ground-aware convolution is designed to guide the object center to extract features and reason the depth from its contact point; the structure is presented in Figure 4.

To first inject perspective geometry into the network, we encode the prior depth value $z$ of each pixel point, assuming that it is on the ground. The persepctive geometry foundation is presented in Figure 3.

According to the ideal pin-hole camera model, the relation between the depth $z$ and height $y_{3d}$ can be obtained as:

$$z \cdot v = f_y \cdot y_{3d} + c_y \cdot z + T_y, \tag{2}$$

where $f_y, c_y$, and $T_y$ are focal lengths, the principal point coordinate and relative translation respectively, and $v$ is the pixel's y-coordinate in the image.

Assume we know the expected elevation $EL$ of the camera from the ground (1.65 meters in the KITTI dataset [27]). The distance from the ground plane pixel to the camera in $z$ can be solved from Equation 2 as:

$$z = \frac{f_y \cdot EL + T_y}{v - c_y}. \tag{3}$$

We note that the function is not continuous around the vanishing line of the ground plane ($v = c_y$), and, as indicated in Figure 3, physically unachievable for $v < c_y$. To detour from such a problem, we first propose to encode the depth value as the disparity of a virtual stereo setup (baseline $B =$

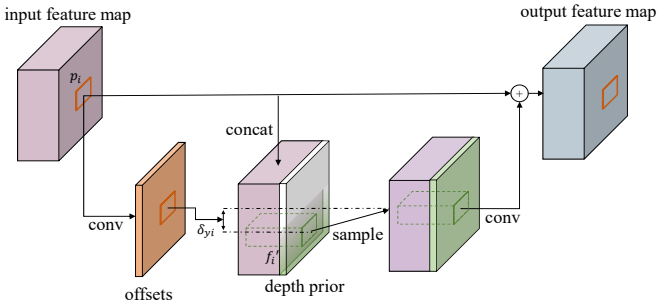


Fig. 4: Ground-aware convolution. The network predicts the offsets in the vertical direction, and we sample the corresponding features and depth priors from pixels below. Depth priors are computed with perspective geometry with ground plane assumption.

$0.54m$, similar to the KITTI stereo setup), and we derive the virtual disparity

$$d = f_y \cdot B \frac{v - c_y}{f_y \cdot EL + T_y} \tag{4}$$

based on the depth $z$ in Equation 3. Rectified Linear Unit (ReLU) activation ($max(x, 0)$) is then applied to suppress pixels with disparity smaller than zero, which is physically unachievable for forward-facing cameras. After these two steps, the depth priors of the image becomes spatially continuous and consistent.

Inspired by CoordinateConv [28], we treat this depth prior as an additional feature map with the same spatial size as the base feature map. Each element in the feature map is now encoded with depth priors assuming it is on the ground.

As motivated in Figure 1, pixels at the center of the object need to query the depth and image features from contact points, which are usually below the object centers.

Each point $p_i$ in the feature map will then dynamically predict an offset $\delta_{yi}$ as if it is the center of a foreground object

$$\delta_{yi} = \delta_{yi}^0 + \Delta_i = \frac{\hat{h}}{2EL - \hat{h}} \cdot (v - c_y) + \Delta_i,$$

, where $\hat{h}$ is the height of the object (we fix this to be the average height of foreground objects of the dataset), $\Delta_i$ is the residual predicted by the convolution networks.

Then, as shown in Figure 4, we extract features $f_i'$ at position $p_i + \delta_{yi}$ using linear interpolation. The extracted features $f_i'$ are merged back to the original point $p_i$ with a residual connection.

The ground-aware convolution module mimics how humans utilize the ground plane in depth perception. It extracts geometric priors and features from pixels beneath. The other part of the network is then responsible for predicting the depth residual between the priors and the targets. The module is differentiable and trained end-to-end with the entire network.

### C. Network Architecture for Monocular 3D Detection

The inference structure of the network is presented in Figure 2. We adopt ResNet-101 [30] as the backbone network, and we only take features at scale $1/16$. The feature map is then fed into the classification branch and regression branch.

The classification branch consists of two convolutional layers, while the regression branch is composed of a ground-aware convolution module followed by a convolutional output layer.

The shape of the output tensor from the classification branch $C$ is $(B, \frac{W}{16}, \frac{H}{16}, K * \#anchors)$, where $K$ represents the number of classes and $\#anchors$ means the number of anchors per pixel. The output tensor from the regression branch is $(B, \frac{W}{16}, \frac{H}{16}, 12 * \#anchors)$. There are nine parameters for each anchor: four for 2D bounding box estimation, three for object center predictions, three for dimension predictions, and two more for observation angle predictions.

*1) Loss Functions:* The total loss $\mathcal{L}$ is the aggregation of classification loss for objectness $L_{cls}$, and regression loss for other parameters $L_{reg}$:

$$\mathcal{L} = L_{cls} + L_{reg}.$$

We adopt focal loss [1], [31] for classification of objectness and cross-entropy loss for the multi-bin classification of width, height, and length. Other parameters, $[x_{2d}, y_{2d}, w_{2d}, h_{2d}, cx, cy, z, w_{3d}, h_{3d}, l_{3d}, sin(\alpha), cos(\alpha)]$, are normalized based on the anchors' prior parameters and optimized through smoothed-L1 loss [32].

*2) Post Optimization:* We follow [18] to apply hill-climbing algorithms as a post-optimization procedure. By perturbating the observation angle and depth estimation, the algorithm incrementally maximizes the IoU between the directly estimated 2D bounding box and the 2D bounding box projected from the 3D bounding box to the image plane.

The original implementation optimizes the depth and observation angle concurrently. With repeated experiments, we find that optimizing only the observation angle produces even better results in the validation set. Concurrently optimizing two variables could overfit to the sparse 3D-2D constraints and affect the accuracy of the 3D prediction.

### D. Network Architecture for Monocular Depth Prediction

We adopt a U-Net [33] structure for supervised dense depth prediction. We select a pretrained ResNet-34 [30] as the backbone encoder.

In the decoding phase, the features are bilinearly upsampled, followed by two convolution layers and concat with the skip connections. We add a ground-aware convolution module before the two convolution layers in the decoder.

The depth prediction network densely predicts the logarithm of depth from each image with a $(B, 1, H, W)$ tensor $y = \log z$. We provide supervision on each output scale $l$. The total loss is the sum of a scale-invariant (SI) loss $\mathcal{L}_{SI}$ [21] and a smoothness loss $\mathcal{L}_{smooth}$ [34] with hyperparameter $\alpha$:

$$\mathcal{L} = \sum_l (\mathcal{L}_{SI} + \alpha \mathcal{L}_{smooth}). \tag{5}$$

SI loss is commonly used to simultaneously minimize the mean-square-error (MSE) and improve global consistency.

TABLE I: 3D Object Detection Results of Car on KITTI Test Set

| Methods | 3D Easy | 3D Moderate | 3D Hard | BEV Easy | BEV Moderate | BEV Hard | Time |
|---|---|---|---|---|---|---|---|
| MonoPSR[15] | 10.76 % | 7.25 % | 5.85 % | 18.33 % | 12.58 % | 9.91 % | 0.2s |
| PLiDAR[14] | 10.76 % | 7.50 % | 6.10 % | 21.27 % | 13.92 % | 11.25 % | 0.1s |
| SS3D[6] | 10.78 % | 7.68 % | 6.51 % | 16.33 % | 11.52 % | 9.93 % | 0.05s |
| MonoDIS[29] | 10.37 % | 7.94 % | 6.40 % | 17.23 % | 13.19 % | 11.12 % | 0.1s |
| M3D-RPN[18] | 14.76 % | 9.71 % | 7.42 % | 21.02 % | 13.67 % | 10.42 % | 0.16s |
| RTM3D[7] | 14.41 % | 10.34 % | 8.77 % | 19.17 % | 14.20 % | 11.99 % | 0.05s |
| AM3D[12] | 16.50 % | 10.74 % | 9.52 % | 25.03 % | 17.32 % | **14.91** % | 0.4s |
| D4LCN[19] | 16.65 % | 11.72 % | 9.51 % | 22.51 % | 16.02 % | 12.55 % | 0.2s |
| **Ours** | **21.65 %** | **13.25 %** | **9.91 %** | **29.81 %** | **17.98 %** | 13.08 % | **0.05s** |

Smoothness loss is needed because the supervision from the KITTI dataset [27] is sparse and lacks local consistency. The SI loss and smoothness loss are computed with the following equations:

$$\mathcal{L}_{SI} = \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}(\sum_i d_i)^2 \tag{6}$$

$$\mathcal{L}_{smooth} = \frac{1}{N}\sum_i |\partial_x z_i^l| e^{-||\partial_x I_i^l||} + |\partial_y z_i^l| e^{-||\partial_y I_i^l||}, \tag{7}$$

where $d_i = \log z_i - \log z_i^*$, $n$ is the number of valid pixels, $\lambda \in [0, 1]$ is a hyperparameter balancing the absolute MSE loss and relative scale loss, $N$ is the number of total pixels, and $\partial_x I$ and $\partial_y I$ are the gradients of the input images.

## IV. Experiments

### A. Dataset and Training Setups

We first evaluate the proposed monocular 3D detection network on the KITTI benchmark [27]. The dataset consists of 7,481 training frames and 7,518 test frames. Chen *et al.* [35] further splits the training set into 3,712 training frames and 3,769 validation frames.

We first determine the hyperparameters of the network with a family of smaller networks fine-tuned on Chen's split [35]. Then, we retrain the final network on the entire training set with the same hyperparameters before uploading the result for testing on the KITTI server. The ablation study that follows is also conducted on the validation set of Chen's split.

Similar to RTM3D [7], we double the training set by utilizing images both from the left and right RGB cameras (only RGB images from the left camera are used in validation and final testing) and use random horizontal mirroring as data augmentation (not applied in validation and testing), which significantly enlarges the training set and improve performance. The top 100 pixels of each image are cropped to speed up inference, and the cropped input images are scaled to $288 \times 1280$ for the model submitted to the KITTI server, which is similar to the original scale of the images. The feature map produced by the backbone, therefore, has a shape of $18 \times 80$. Regression loss and classification loss that are too small in magnitude (1e-3) are clipped to prevent overfitting. The network is trained with a batch size of 8 on a single Nvidia 1080Ti GPU. During inference, the network is fed one image at a time, and the total average processing time, including file IO and post-optimization, is 0.05s per frame.

TABLE II: Depth Prediction Results on KITTI Test Set

| Methods | SILog | sqErrorRel | absErrorRel | iRMSE |
|---|---|---|---|---|
| PAP[36] | 13.08 | 2.72 % | 10.27 % | 13.95 |
| VNL[37] | 12.65 | **2.46** % | 10.15 % | 13.02 |
| SoftDorn[21] | 12.39 | 2.49 % | 10.10 % | 13.48 |
| Base U-Net | 12.78 | 3.11 % | 10.12 % | 13.46 |
| **Ours** | **12.13** | 2.61 % | **9.41** % | **12.65** |

### B. Evaluation Metric and Results for 3D Detection

As pointed out by Simonelli *et al.* [29] and the KITTI team, evaluating performance with 40 recall positions ($AP_{40}$) instead of the 11 recall positions ($AP_{11}$) proposed in the original Pascal VOC benchmark[38] could eliminate the problematic results presented in the lowest recall bin. Therefore, we present our results on the test set and also ablation study based on $AP_{40}$.

The results are presented in Table I alongside those of other SOTA monocular 3D detection methods based on the KITTI benchmark.

The proposed network significantly outperforms existing methods on easy and moderate vehicles. We do expect ground-aware convolutions to produce more accurate predictions for close-up vehicles with clear borders with the ground plane.

Qualitative results are presented in Figure 5. The model shown here shares the same hyperparameters as the model submitted to the KITTI server but is only trained on the training sub-split. In the images on the left-hand side of the figure, cars are mostly detected and estimated accurately. The effect of the GAC module is also visualized.

We present several typical failure cases on the right-hand side of Figure 5, and in the top-right image, the network does not detect a heavily obscured car. In the middle-right image, truncated cars and a car that is quite far away are not detected. We acknowledge that the network could still have trouble detecting small objects. We show the bottom-right image to demonstrate cases in which the network give an inaccurate estimation of the 3D dimensions of a car because, as stated in Section III, it is still difficult to estimate the width, length, and height of an object merely by semantic information in the image.

We provide an ablation study of the model in Section V.

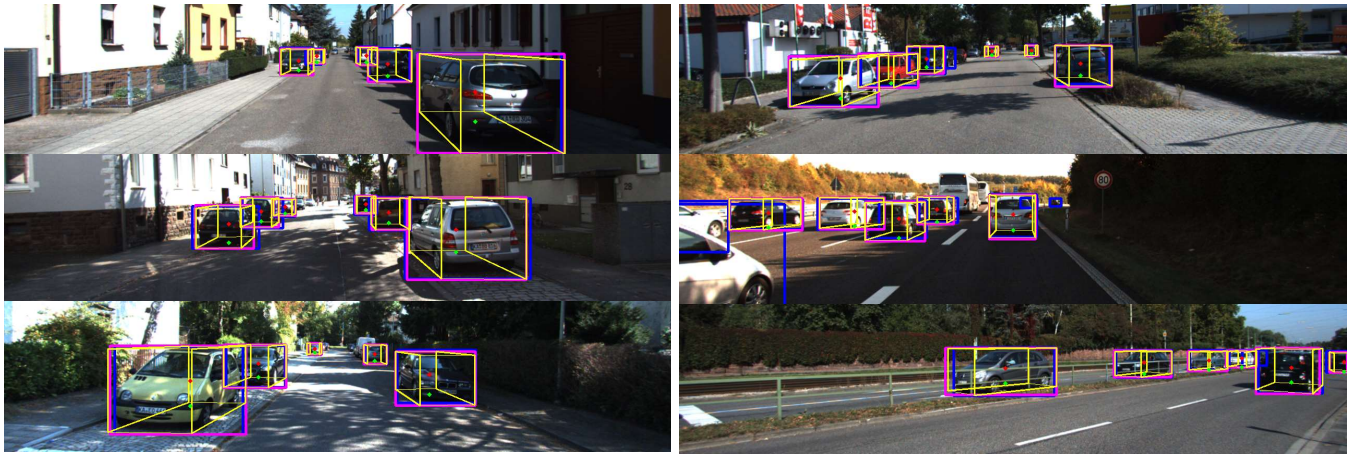

Fig. 5: Qualitative examples from validation sets. Blue boxes, pink boxes and yellow boxes indicate the ground truth 2D bounding box, estimated 2D bounding box, and estimated 3D bounding box respectively. Red points are object centers and green points visualize the offsets $\delta_{yi}$ in the GAC module.

### C. Experiments on Monocular Depth Prediction

We further evaluate the proposed depth prediction network in the KITTI depth prediction benchmark [27]. The dataset for monocular depth prediction consists of 42949 training frames, 1000 validation samples, and 500 test samples, annotated with sparse point clouds.

The input images are cropped to $352\times1216$ during training and testing. In the loss function, we applied $\alpha$= 0.3, $\lambda$=0.3 through grid-search on the validation set. The network is also trained with a batch size of 8 on a single Nvidia 1080Ti GPU.

Scale-invariant log error (SILog) is the primary metric used in the KITTI benchmark to evaluate depth prediction algorithms.

The results are presented in Table II. The proposed network produces one of the best performances on the KITTI dataset, providing competitive results compared with SOTA methods. We also show that the network improves significantly against the baseline U-Net Model.

Qualitative results are presented in Figure 6. Depth predictions inside the range of LiDAR are generally consistent. Depth predictions along long, vertical objects like trees are consistent thanks to the ground aware convolution module. We point out that there are still artifacts around the edge of objects and areas without supervision because the network receives no post-processing and little pre-training. The depth prediction results show that the proposed module and the proposed network can improve depth inferencing from monocular images in autonomous driving scenes.

## V. Model Analysis and Discussion

In this section, we further analyze the performance of the proposed method and discuss the effectiveness of each design choice. The experiments will focus more on monocular 3D detection. We conduct ablation studies to validate the contribution of anchor preprocessing and the ground-aware convolution module.

### A. Anchor Preprocessing

We first conduct experiments on anchor filtering. In the experiment, we do not filter out unnecessary anchors during training and testing. We notice that the proposed filtering will filter out half of the negative anchors, so we also conduct an experiment against Online Hard Example Mining (OHEM), where we filter out half of the easy negative anchors during training[39].

As shown in Table III, the baseline model outperforms the ablated one and OHEM. The baseline model performs better at 3D inference. We also point out that there is almost no difference in 2D detection between the two models.

Generally, a one-stage single-scale object detector not only needs to classify background from the foreground but also needs to select anchors with the correct scales at foreground pixels, which also means selecting the proper depth prior. Filtering off-the-ground anchors during training and testing significantly lowers the learning burden for the classification branch of the object detector. Thus the classification branch can focus more on selecting the right anchors for foreground pixels. Such a method, as a result, also outperforms position-invariant filtering methods like OHEM.

### B. Ground-Aware Convolution Module

Intuitively, basic convolutions provide a uniform receptive field for each pixel, and the network could implicitly learn to adjust its receptive field by fine-tuning the weights of multiple convolution layers. Deformable convolutions [40] further explicitly encourage the network to adapt its receptive field according to each pixel's surrounding context. Compared with deformable convolutions, the ground-aware convolution module fixes the search direction and allows a larger search range.

We substitute the proposed module with basic convolutions, disparity-conditioned convolutions (i.e., convolution with the depth prior as an additional feature map), and deformable convolutions to examine the performance. The

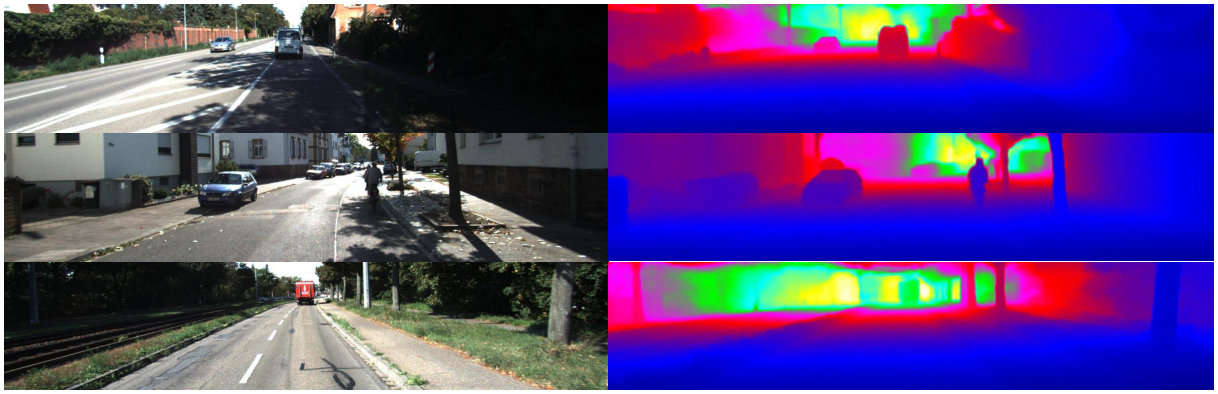

Fig. 6: Qualitative examples of depth prediction from validation sets. The depth maps on the right are rendered with the official color map.

TABLE III: 3D Detection Ablation Study Results of Car on KITTI Validation Set

| **Methods** | $IoU \geq 0.7$ **3D Easy/Moderate/Hard** | $IoU \geq 0.5$ **3D Easy/Moderate/Hard** |
|---|---|---|
| **Baseline Model** | **23.63 %/ 16.16 %/ 12.06 %** | **60.92 %/ 42.18 %/ 32.02 %** |
| w/o Anchor Filtering | 21.39 %/ 14.35 %/ 11.11 % | 59.76 %/ 41.00 %/ 31.10 % |
| w OHEM | 22.45 %/ 15.10 %/ 11.29 % | 60.71 %/ 42.01 %/ 31.88 % |
| w Conv | 21.57 %/ 15.26 %/ 11.35 % | 58.17 %/ 41.17 %/ 32.58 % |
| w DisparityConv | 22.13 %/ 15.42 %/ 11.34 % | 60.13 %/ 41.62 %/ 33.07 % |
| w Deformable Conv | 22.16 %/ 15.71 %/ 11.75 % | 62.24 %/ 43.93 %/ 33.76 % |

results are shown in Table III. The experiments with deformable convolutions demonstrate better 2D detection results.

Deformable convolution can enhance the performance with a generally larger receptive field. While disparity-conditioned convolution provides the network with prior depths, the receptive field of the network is lacking. These two modules improve the performance, but the proposed module has better results by a considerable margin.

## VI. Conclusion

In this paper, we presented ground-aware monocular 3D object detection for autonomous driving scenes. First, we improved the problem setup for monocular 3D detection and introduced an anchor filtering procedure to inject ground plane priors and statistical priors in anchors. Second, we introduced a ground-aware convolution module, providing sufficient hints and geometric priors for the network to reason based on ground plane priors. The proposed monocular 3D object detection network was tested on the KITTI detection benchmark and achieved SOTA performance among monocular methods. We further tested the ground-aware convolution module in the monocular depth prediction task, and it also produced competitive results on the KITTI depth prediction benchmark.

We note that the "floor-wall" assumption is limited to scenes with specific camera poses, and it only partially holds in a complex driving scene. The proposed methods still do not reason explicitly based on the boundaries of the ground and other objects. Instead, we encode sufficient information and priors into the network and adopt a data-driven approach.

Nevertheless, the proposed method pushes the boundary of 3D detection and depth inference from images and produces powerful neural network models for autonomous driving and mobile robotics scenes.

## References


[1] P. Yun, L. Tai, Y. Wang, C. Liu, and M. Liu, "Focal loss in 3d object detection," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1263–1270, April 2019.

[2] W. Sukai, S. Yuxiang, L. Chengju, and L. Ming, "Pointtracknet: An end-to-end network for 3-d object detection and tracking from point clouds," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 02 2020.

[3] Z. Chen, Q. Liao, Z. Wang, Y. Liu, and M. Liu, "Image detector based automatic 3d data labeling and training for vehicle detection on point cloud," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, June 2019, pp. 1408–1413.

[4] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[5] A. Naiden, V. Paunescu, G. Kim, B. Jeon, and M. Leordeanu, "Shift r-cnn: Deep monocular 3d object detection with closed-form geometric constraints," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 61–65.

[6] E. Jörgensen, C. Zach, and F. Kahl, "Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss," *arXiv preprint arXiv:1906.08070*, vol. abs/1906.08070, 2019. [Online]. Available: http://arxiv.org/abs/1906.08070

[7] P.-X. Li, H. Zhao, P. Liu, and F. Cao, "Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 644–660.

[8] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, , and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian Conference on Computer Vision*, 2012, pp. 548–562.

[9] E. Delage, Honglak Lee, and A. Y. Ng, "A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 2418–2428.

[10] C. Chun, D. Park, W. Kim, and C. Kim, "Floor detection based depth estimation from a single indoor scene," in *2013 IEEE International Conference on Image Processing*, 2013, pp. 3358–3362.

[11] Y. Wang, W. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8437–8445.

[12] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, "Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 10 2019, pp. 6850–6859.

[13] V. J. M. Uwabeza, A. Shubhra, and L. Bingbing, "Refinedmpl: Refined monocular pseudolidar for 3d object detection in autonomous driving," *arXiv preprint*, vol. abs/1911.09712, 11 2019.

[14] X. Weng and K. Kitani, "Monocular 3d object detection with pseudo-lidar point cloud," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 857–866.

[15] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3d object detection leveraging accurate proposals and shape reconstruction," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 859–11 868.

[16] Z. Liu, Z. Wu, and R. Tth, "Smoke: Single-stage monocular 3d object detection via keypoint estimation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 4289–4298.

[17] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," in *arXiv preprint arXiv:1904.07850*, 2019.

[18] G. Brazil and X. Liu, "M3d-rpn: Monocular 3d region proposal network for object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9286–9295.

[19] M. Ding, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu, and P. Luo, "Learning depth-guided convolutions for monocular 3d object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 669–11 678.

[20] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2002–2011.

[21] R. Daz and A. Marathe, "Soft labels for ordinal regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4733–4742.

[22] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *arXiv preprint arXiv:1907.10326*, 2019.

[23] S. Aich, J. M. U. Vianney, M. A. Islam, M. Kaur, and B. Liu, "Bidirectional attention network for monocular depth estimation," *arXiv preprint arXiv:2009.00743*, 2020.

[24] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision (ECCV)*, 2012, pp. 746–760.

[25] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, and G. Shakhnarovich, "DIODE: A Dense Indoor and Outdoor DEpth Dataset," *arXiv preprint*, vol. abs/1908.00463, 2019. [Online]. Available: http://arxiv.org/abs/1908.00463

[26] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[27] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[28] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution," in *Advances in Neural Information Processing Systems*, 11 2018.

[29] A. Simonelli, S. Rota Bulo, L. Porzi, M. Lopez Antequera, and P. Kontschieder, "Disentangling monocular 3d object detection: From single to multi-class recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 07 2018.

[32] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, vol. abs/1505.04597, 2015, pp. 234–241.

[34] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[35] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 424–432. [Online]. Available: http://papers.nips.cc/paper/5644-3d-object-proposals-for-accurate-object-class-detection.pdf

[36] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-affinitive propagation across depth, surface normal and semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4106–4115.

[37] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5683–5692.

[38] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[39] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 761–769.

[40] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9300–9308.