MonoPair: Monocular 3D Object Detection Using Pairwise Spatial Relationships

Yongjian Chen Lei Tai Kai Sun Mingyang Li
Alibaba Group
{yongjian.cyj, tailei.tl, sk157164, mingyangli}@alibaba-inc.com

Abstract

Monocular 3D object detection is an essential component in autonomous driving while challenging to solve, especially for those occluded samples which are only partially visible. Most detectors consider each 3D object as an independent training target, inevitably resulting in a lack of useful information for occluded samples. To this end, we propose a novel method to improve the monocular 3D object detection by considering the relationship of paired samples. This allows us to encode spatial constraints for partially-occluded objects from their adjacent neighbors. Specifically, the proposed detector computes uncertainty-aware predictions for object locations and 3D distances for the adjacent object pairs, which are subsequently jointly optimized by nonlinear least squares. Finally, the one-stage uncertainty-aware prediction structure and the post-optimization module are dedicatedly integrated for ensuring the run-time efficiency. Experiments demonstrate that our method yields the best performance on KITTI 3D detection benchmark, by outperforming state-of-the-art competitors by wide margins, especially for the hard samples.

1. Introduction

3D object detection plays an essential role in various computer vision applications such as autonomous driving, unmanned aircrafts, robotic manipulation, and augmented reality. In this paper, we tackle this problem by using a monocular camera, primarily for autonomous driving use cases. Most existing methods on 3D object detection require accurate depth information, which can be obtained from either 3D LiDARs [8, 30, 34, 35, 23, 45] or multi-camera systems [6, 7, 20, 29, 32, 41]. Due to the lack of directly computable depth information, 3D object detection using a monocular camera is generally considered a much more challenging problem than using LiDARs or multi-camera systems. Despite the difficulties in computer vision algorithm design, solutions relying on a monocular camera can potentially allow for low-cost, low-power, and deployment-flexible systems in real applications. Therefore, there is a growing trend on performing monocular 3D object detection in research community in recent years [3, 5, 26, 27, 31, 36].

Existing monocular 3D object detection methods have achieved considerable high accuracy for normal objects in autonomous driving. However, in real scenarios, there are a large number of objects that are under heavy occlusions, which pose significant algorithmic challenges. Unlike objects in the foreground which are fully visible, useful information for occluded objects is naturally limited. Straightforward methods on solving this problem are to design networks to exploit useful information as much as possible, which however only lead to limited improvement. Inspired by image captioning methods which seek to use scene graph and object relationships [10, 22, 42] , we propose to fully leverage the spatial relationship between close-by objects instead of individually focusing on information-constrained occluded objects. This is well aligned with human's intuition that human beings can naturally infer positions of the occluded cars from their neighbors on busy streets.

Mathematically, our key idea is to optimize the predicted 3D locations of objects guided by their uncertainty-aware spatial constraints. Specifically, we propose a novel detector to jointly compute object locations and spatial constraints between matched object pairs. The pairwise spatial constraint is modeled as a keypoint located in the geometric center between two neighboring objects, which effectively encodes all necessary geometric information. By doing that, it enables the network to capture the geometric context among objects explicitly. During the prediction, we impose aleatoric uncertainty into the baseline 3D object detector to model the noise of the output. The uncertainty is learned in an unsupervised manner, which is able to enhance the network robustness properties significantly. Finally, we formulate the predicted 3D locations as well as their pairwise spatial constraints into a nonlinear least squares problem to optimize the locations with a graph optimization framework. The computed uncertainties are used to weight each term in the cost function. Experiments on challenging KITTI 3D datasets demonstrate that our method outperforms the state-of-the-art competing approaches by wide margins. We also note that for *hard* samples with heavier occlusions, our method demonstrates massive improvement. In summary, the key contributions of this paper are as follows:

- We design a novel 3D object detector using a monocular camera by capturing spatial relationships between paired objects, allowing largely improved accuracy on occluded objects.
- We propose an uncertainty-aware prediction module in 3D object detection, which is jointly optimized together with object-to-object distances.
- Experiments demonstrate that our method yields the best performance on KITTI 3D detection benchmark, by outperforming state-of-the-art competitors by wide margins.

2. Related Work

In this section, we first review methods on monocular 3D object detection for autonomous driving. Related algorithms on object relationship and uncertainty estimation are also briefly discussed.

Monocular 3D Object Detection. Monocular image is naturally of limited 3D information compared with multi-beam LiDAR or stereo vision. Prior knowledge or auxiliary information are widely used for 3D object detection. Mono3D [5] focuses on the fact that 3D objects are on the ground plane. Prior 3D shapes of vehicles are also leveraged to reconstruct the bounding box for autonomous driving [28]. Deep MANTA [4] predicts 3D object information utilizing key points and 3D CAD models. SubCNN [40] learns viewpoint-dependent subcategories from 3D CAD models to capture both shape, viewpoint and occlusion patterns. In [1], the network learns to estimate correspondences between detected 2D keypoints and 3D counterparts. 3D-RCNN [19] introduces an inverse-graphics framework for all object instances from an image. A differentiable Render-and-Compare loss allows 3D results to be learned through 2D information. In [17], a sparse LiDAR scan is used in the training stage to generate training data, which removes the necessity of using inconvenient CAD dataset. An alternative family of methods is to predict a stand-alone depth or disparity information of the monocular image at the first stage [25, 26, 38, 41]. Although they only require the monocular image at testing time, ground-truth depth information is still necessary for the model training.

Compared with the aforementioned works in monocular 3D detection, some algorithms consist of only the RGB image as input rather than relying on external data, network structures or pre-trained models. Deep3DBox [27] infers 3D information from a 2D bounding box considering the geometrical constraints of projection. OFTNet [33] presents a orthographic feature transform to map image-based features into an orthographic 3D space. ROI-10D [26] proposes a novel loss to properly measure the metric misalignment of boxes. MonoGRNet [31] predicts 3D object locations from a monocular RGB image considering geometric reasoning in 2D projection and the unobserved depth dimension. Current state-of-the-art results for monocular 3D object detection are from MonoDIS [36] and M3D-RPN [3]. Among them, MonoDIS [36] leverages a novel disentangling transformation for 2D and 3D detection losses, which simplifies the training dynamics. M3D-RPN [3] reformulates the monocular 3D detection problem as a standalone 3D region proposal network. Very recently, several concurrent works [24, 21] also adopt a keypoint detection strategy similar to our work. However, all the object detectors mentioned above focus on predicting each individual object from the image. The spatial relationship among objects is not considered. Our work is originally inspired by CenterNet [44], in which each object is identified by points. Specifically, we model the geometric relationship between objects by using a single point similar to CenterNet, which is effectively the geometric center between them.

Visual Relationship Detection. Relationship plays an essential role for image understanding. To date, it is widely applied in image captioning. Dai *et al.* [10] proposes a relational network to exploit the statistical dependencies between objects and their relationships. MSDB [22] presents a multi-level scene description network to learn features of different semantic levels. Yao *et al.* [42] proposes an attention-based encoder-decoder framework. through graph convolutional networks and long short-term memory (LSTM) for scene generation. However, these methods are mainly for tackling the effects of visual relationships in representing and describing an image. They usually extract object proposals directly or show full trust for the predicted bounding boxes. By contrast, our method focuses 3D object detection, which is to refine the detection results based on spatial relationships. This is un-explored in existing work.

Uncertainty Estimation in object detection. The computed object locations and pairwise 3D distances of our method are all predicted with uncertainties. This is inspired by the aleatoric uncertainty of deep neural networks [13, 15]. Instead of fully trusting the results of deep neural networks, we can extract how uncertain the predictions. This is crucial for various perception and decision making tasks, especially for autonomous driving, where human lives may be endangered due to inappropriate choices. This concept has been applied in 3D Lidar object detection [12] and pedestrian localization [2], where they mainly consider uncertainties as additional information for reference. In [39], uncertainty is used to approximate object

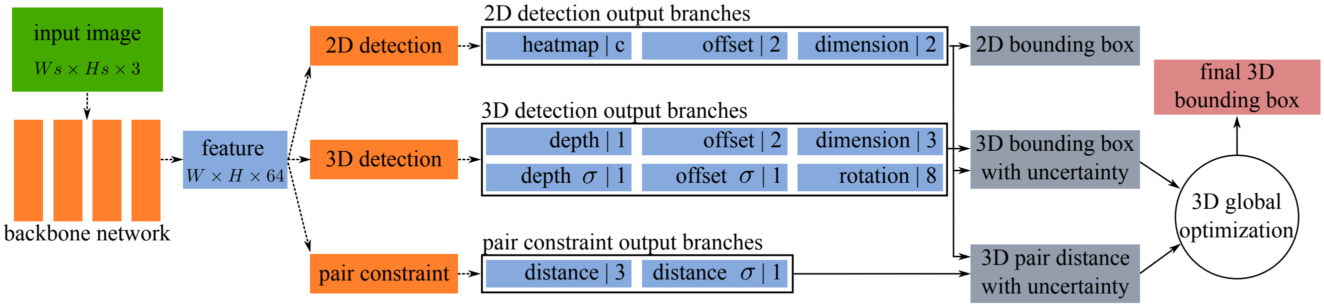


Figure 1: Overview of our architecture. A monocular RGB image is taken as the input to the backbone network and trained with supervision. Eleven different prediction branches, with feature map as $W \times H \times m$, are divided into three parts: 2D detection, 3D detection and pair constraint prediction. The width and height of the output feature (W, H) are as the same as the backbone output. Dash lines represent forward flows of the neural network. The heatmap and offset of 2D detection are also utilized to locate the 3D object center and the pairwise constraint keypoint.

hulls with bounded collision probability for subsequent trajectory planning tasks. Gaussian-YOLO [9] significantly improves the detection results by predicting the localization uncertainty. These approaches only use uncertainty to improve the training quality or to provide an additional reference. By contrast, we use uncertainty to weight the cost function for post-optimization, integrating the detection estimates and predicted uncertainties in global context optimization.

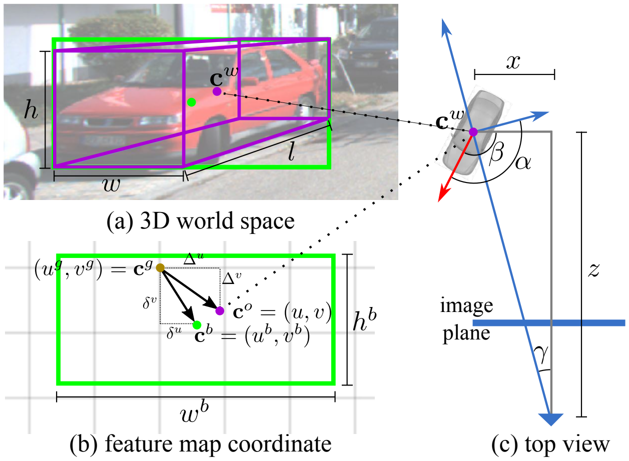
3. Approach

3.1. Overview

We adopt a one-stage architecture, which shares a similar structure with state-of-the-art anchor-free 2D object detectors [37, 44]. As shown in Figure 1, it is composed of a backbone network and several task-specific dense prediction branches. The backbone takes a monocular image I with a size of $(Ws \times Hs)$ as input, and outputs the feature map with a size of $(W \times H \times 64)$, where s is our backbone's down-sampling factor. There are eleven output branches with a size of $W \times H \times m$, where m means the channel of each output branch, as shown in Figure 1. Eleven output branches are divided into three parts: three for 2D object detection, six for 3D object detection, and two for pairwise constraint prediction. We introduce each module in details as follows.

3.2. 2D Detection

Our 2D detection module is derived from the CenterNet [44] with three output branches. The heatmap with a size of $(W \times H \times c)$ is used for keypoint localization and classification. Keypoint types include $c = 3$ in KITTI3D object detection. Details about extracting the object location $\mathbf{c}^g = (u^g, v^g)$ from the output heatmap can be referred in [44]. The other two branches, with two channels for each, output the size of the bounding box (w^b, h^b) and the offset vector (δ^u, δ^v) from the located keypoint $\mathbf{c}^g$ to the bounding box center $\mathbf{c}^b = (u^b, v^b)$ respectively. As shown in Figure 2, those values are in units of the feature map coordinate.

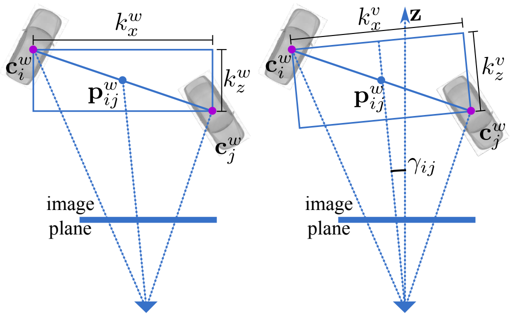


(a) 3D world space
(b) feature map coordinate
(c) top view

Figure 2: Visualization of notations for (a) 3D bounding box in world space, (b) locations of an object in the output feature map, and (c) orientation of the object from the top view. 3D dimensions are in meters, and all values in (b) are in the feature coordinate. The vertical distance y is invisible and skipped in (c).

3.3. 3D Detection

The object center in world space is represented as $\mathbf{c}^w = (x, y, z)$. Its projection in the feature map is $\mathbf{c}^o = (u, v)$ as shown in Figure 2. Similar to [26, 36], we predict its offset (Δ^u, Δ^v) to the keypoint location $\mathbf{c}^g$ and the depth z in two separate branches. With the camera intrinsic matrix



(a) camera coordinate (b) local coordinate

Figure 3: Pairwise spatial constraint definition. $\mathbf{c}_i^w$ and $\mathbf{c}_j^w$ are centers of two 3D bounding boxes where $\mathbf{p}_{ij}^w$ is their middle point. 3D distance in camera coordinate $\mathbf{k}_{ij}^w$ and local coordinate $\mathbf{k}_{ij}^v$ are shown in (a) and (b) respectively. The distance along $\mathbf{y}$ axis is skipped.

$\mathbf{K}$, the derivation from predictions to the 3D center $\mathbf{c}^w$ is as follows:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & a_x \\ 0 & f_y & a_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

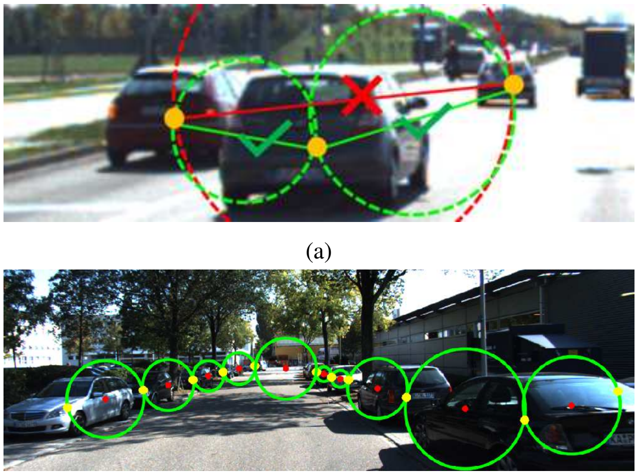
$$\mathbf{c}^w = (\frac{u^g + \Delta^u - a_x}{f_x} z, \frac{v^g + \Delta^v - a_y}{f_y} z, z) \tag{2}$$

Given the difficulty to regress depth directly, depth prediction branch outputs inverse depth $\hat{z}$ similar to [11], transforming the absolute depth by inverse sigmoid transformation $z = 1/\sigma(\hat{z}) - 1$. The dimension branch regresses the size (w, h, l) of the object in meters directly. The branches for depth, offset and dimensions in both 2D and 3D detection are trained with the L1 loss following [44].

As presented in Figure 2, we estimate the object's local orientation α following [27] and [44]. Compared to global orientation β in the camera coordinate system, the local orientation accounts for the relative rotation of the object to the camera viewing angle $\gamma = \arctan(x/z)$. Therefore, using the local orientation is more meaningful when dealing with image features. Similar to [27, 44], we represent the orientation using eight scalars, where the orientation branch is trained by $MultiBin$ loss.

3.4. Pairwise Spatial Constraint

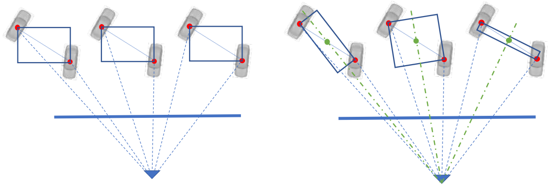
In addition to the regular 2D and 3D detection pipelines, we propose a novel regression target, which is to estimate the pairwise geometric constraint among adjacent objects via a keypoint on the feature map. Pair matching strategy for training and inference is shown in Figure 4a. For arbitrary sample pair, we define a range circle by setting the distance of their 2D bounding box centers as the diameter. This pair is neglected if it contains other object centers. Figure 4b shows an example image with all effective sample pairs.



(a)

(b)

Figure 4: Pair matching strategy for training and inference.



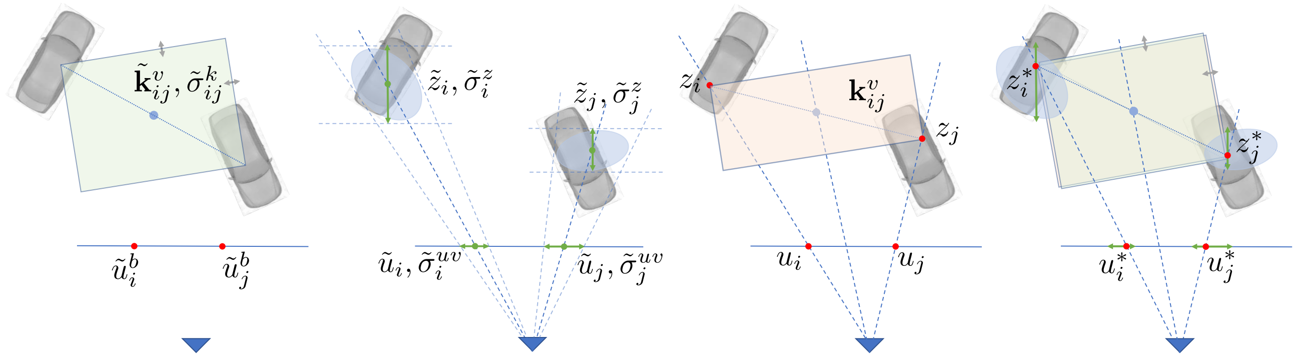
(a) camera coordinate (b) local coordinate

Figure 5: The same pairwise spatial constraint in camera and local coordinates from various viewing angles. The spatial constraint in camera coordinate is invariant among different view angles. Considering the different projected form of the car, we use the 3D absolute distance in local coordinate as the regression target of spatial constraint.

Given a selected pair of objects, their 3D centers in world space are $\mathbf{c}_i^w = (x_i, y_i, z_i)$ and $\mathbf{c}_j^w = (x_j, y_j, z_j)$ and their 2D bounding box centers on the feature map are $\mathbf{c}_i^b = (u_i^b, v_i^b)$ and $\mathbf{c}_j^b = (u_j^b, v_j^b)$. The pairwise constraint keypoint locates on the feature map as $\mathbf{p}_{ij}^b = (\mathbf{c}_i^b + \mathbf{c}_j^b)/2$. The regression target for the related keypoint is the 3D distance of these two objects. We first locate the middle point $\mathbf{p}_{ij}^w = (\mathbf{c}_i^w + \mathbf{c}_j^w)/2 = (p_x^w, p_y^w, p_z^w)_{ij}$ in 3D space. Then, the 3D absolute distance $\mathbf{k}_{ij}^v = (k_x^v, k_y^v, k_z^v)_{ij}$ along the view point direction, as shown in Figure 3b, are taken as the regression target which is the distance branch of the pair constraint output in Figure 1. Notice that $\mathbf{p}^b$ is not the projected point of $\mathbf{p}^w$ on the feature map, like $\mathbf{c}^w$ and $\mathbf{c}^b$ in Figure 2.

For training, $\mathbf{k}_{ij}^v$ can be easily collected through the groundtruth 3D object centers from the training data as:

$$\mathbf{k}_{ij}^v = \left|\overrightarrow{\mathbf{R}(\gamma_{ij})\mathbf{k}_{ij}^w}\right|, \tag{3}$$


$\tilde{\mathbf{k}}^v_{ij}, \tilde{\sigma}^k_{ij}$
$\tilde{u}^b_i$
$\tilde{u}^b_j$
$\tilde{z}_i, \tilde{\sigma}^z_i$
$\tilde{z}_j, \tilde{\sigma}^z_j$
$\tilde{u}_i, \tilde{\sigma}^{uv}_i$
$\tilde{u}_j, \tilde{\sigma}^{uv}_j$
z_i
$\mathbf{k}^v_{ij}$
z_j
u_i
u_j
z^*_i
z^*_j
u^*_i
u^*_j

(a) pair constraint prediction (b) object location prediction (c) variables of optimization (d) optimized results

Figure 6: Visualization of optimization for an example pair including. In (a), The predicted pairwise constraint $\tilde{\mathbf{k}}^v_{ij}$ and its uncertainty $\tilde{\sigma}^k_{ij}$ is located by predicted 2D bounding box centers $(\tilde{u}^b_i, \tilde{v}^b_i)$ and $(\tilde{u}^b_j, \tilde{v}^b_j)$ on the feature map. The 3D prediction results (green points) are shown in (b). All uncertainties are represented as arrows to show a confidence range. We show variables in (c) for this optimization function as red points. The final optimized results are presented in (d). Our method is mainly supposed to work for occluded samples. The relatively long distance among the paired cars is for simplicity in visualization. Properties along v direction is skipped.

where $\overrightarrow{|\cdot|}$ means extract absolute value of each entry in the vector. $\mathbf{k}^w_{ij} = \mathbf{c}^w_i - \mathbf{c}^w_j$ is the 3D distance in camera coordinate, $\gamma_{ij} = \arctan(p^w_x / p^w_z)$ is the view direction of their middle point $\mathbf{p}^w_{ij}$, and $\mathbf{R}(\gamma_{ij})$ is its rotation matrix along the $\mathbf{Y}$ axis as

$$\mathbf{R}(\gamma_{ij}) = \begin{bmatrix} \cos(\gamma_{ij}) & 0 & -\sin(\gamma_{ij}) \\ 0 & 1 & 0 \\ \sin(\gamma_{ij}) & 0 & \cos(\gamma_{ij}) \end{bmatrix}. \tag{4}$$

The 3D distance $\mathbf{k}^w$ in camera coordinate is not considered because it is invariant from different view angles, as shown in Figure 5a. As in estimation of the orientation γ, 3D absolute distance $\mathbf{k}^v$ in the local coordinate of $\mathbf{p}^w$ is more meaningful considering the appearance change through viewing angles.

In inference, we first estimate objects' 2D locations and extract pairwise constraint keypoint located in the middle of predicted 2D bounding box centers. The predicted $\tilde{\mathbf{k}}^v$ is extracted in the dense feature map of the distance branch based on the keypoint location. We do not consider offsets for this constraint keypoint both in training and reference, and round the middle point $\mathbf{p}^b_{ij}$ of paired objects' 2D centers to the nearest grid point on the feature map directly.

3.5. Uncertainty

Following the heteroscedastic aleatoric uncertainty setup in [15, 16], we represent a regression task with L1 loss as

$$[\tilde{\mathbf{y}}, \tilde{\sigma}] = f^{\theta}(\mathbf{x}), \tag{5}$$

$$L(\theta) = \frac{\sqrt{2}}{\tilde{\sigma}} \|\mathbf{y} - \tilde{\mathbf{y}}\| + \log \tilde{\sigma}. \tag{6}$$

Here, $\mathbf{x}$ is the input data, $\mathbf{y}$ and $\tilde{\mathbf{y}}$ are the groundtruth regression target and the predicted result. $\tilde{\sigma}$ is another output of the model and can represent the observation noise of the data $\mathbf{x}$. θ is the weight of the regression model.

As mentioned in [15], aleatoric uncertainty $\tilde{\sigma}(\mathbf{x})$ makes the loss more robust to noisy input in a regression task. In this paper, we add three uncertainty branches as shown as σ blocks in Figure 1 for the depth prediction σ^z, 3D center offset σ^{uv} and pairwise distance σ^k respectively. They are mainly used to weight the error terms as presented in Section 3.6.

3.6. Spatial Constraint Optimization

As the main contribution of this paper, we propose a post-optimization process from a graph perspective. Suppose that in one image, the network outputs N effective objects, and there are M pair constraints among them based on the strategy in Section 3.4. Those paired objects are regarded as vertices $\{\xi_i\}_{i=1}^{N^{\mathcal{G}}}$ with size of $N^{\mathcal{G}}$ and the M paired constraints are regarded as edges of the graph. Each vertex may connect multiple neighbors. Predicted objects not connected by other vertices are not updated anymore in the post-optimization. The proposed spatial constraint optimization is formulated as a nonlinear least square problem as

$$\underset{(u_i, v_i, z_i)_{i=1}^{N^{\mathcal{G}}}}{\arg\min} \; \mathbf{e}^T \mathbf{W} \mathbf{e}, \tag{7}$$

where $\mathbf{e}$ is the error vector and $\mathbf{W}$ is the weight matrix for different errors. $\mathbf{W}$ is a diagonal matrix with dimension $3N^{\mathcal{G}} + 3M$. For each vertex ξ_i, there are three variables (u_i, v_i, z_i), which are the projected center (u_i, v_i) of the 3D bounding box on the feature map and the depth z_i as shown in Figure 2. We introduce each minimization term in the following.

Pairwise Constraint Error For each pairwise constraint connecting ξ_i and ξ_j, there are three error terms $(\mathbf{e}^x_{ij}, \mathbf{e}^y_{ij}, \mathbf{e}^z_{ij})$ measuring the inconsistency between network estimated 3D distance $\tilde{\mathbf{k}}^v_{ij}$ and the distance $\mathbf{k}^v_{ij}$ obtained by 3D locations $\mathbf{c}^w_i$ and $\mathbf{c}^w_j$ of the two associated objects. $\mathbf{c}^w_i$ and $\mathbf{c}^w_j$ can be represented by variables (u_i, v_i, z_i), (u_j, v_j, z_j) and the known intrinsic matrix through Equation 2. Thus, error terms $(\mathbf{e}^x_{ij}, \mathbf{e}^y_{ij}, \mathbf{e}^z_{ij})$ are the absolute difference between $\tilde{\mathbf{k}}^v_{ij}$ and $\mathbf{k}^v_{ij}$ along three axis as following.

$$\mathbf{k}^v_{ij} = \left|\overrightarrow{\mathbf{R}(\gamma_{ij})(\mathbf{c}^w_i - \mathbf{c}^w_j)}\right| \tag{8}$$

$$(\mathbf{e}^x_{ij}, \mathbf{e}^y_{ij}, \mathbf{e}^z_{ij})^T = \left|\overrightarrow{\tilde{\mathbf{k}}^v_{ij} - \mathbf{k}^v_{ij}}\right| \tag{9}$$

Object Location Error For each vertex ξ_i, there are three error terms $(\mathbf{e}^u_i, \mathbf{e}^v_i, \mathbf{e}^z_i)$ to regularize the optimization variables with the predicted values from the network. We use this term to constraint the deviation between network estimated object location and the optimized location as follows.

$$\mathbf{e}^u_i = \left|\tilde{u}^g_i + \tilde{\Delta}^u_i - u_i\right| \tag{10}$$

$$\mathbf{e}^v_i = \left|\tilde{v}^g_i + \tilde{\Delta}^v_i - v_i\right| \tag{11}$$

$$\mathbf{e}^z_i = \left|\tilde{z}_i - z_i\right| \tag{12}$$

Weight Matrix The weight matrix $\mathbf{W}$ is constructed by the uncertainty output $\tilde{\sigma}$ of the network. The weight of the error is higher when the uncertainty is lower, which means we have more confidence in the predicted output. Thus, we use $1/\tilde{\sigma}$ as the element of $\mathbf{W}$. For pairwise inconsistency, the weights for the three error terms $(\mathbf{e}^x_{ij}, \mathbf{e}^y_{ij}, \mathbf{e}^z_{ij})$ are the same as the predicted $1/\tilde{\sigma}_{ij}$ as shown in Figure 6a. For object location error, the weight is $1/\tilde{\sigma}^z_i$ for depth error $\mathbf{e}^z_i$ and $1/\tilde{\sigma}^{uv}_i$ for both $\mathbf{e}^u_i$ and $\mathbf{e}^v_i$ as shown in Figure 6b. We visualize an example pair for the spatial constraint optimization in Figure 6. Uncertainties give us confidence ranges to tune variables so that both the pairwise constraint error and the object location error can be jointly minimized. We use g2o [18] to conduct this graph optimization structure during implementation.

4. Implementation

We conduct experiments on the challenge KITTI 3D object detection dataset [14]. It is split to 3712 training samples and 3769 validation samples as [6]. Samples are labeled from *Easy*, *Moderate*, to *Hard* according to its condition of truncation, occlusions and bounding box height. Table 1 shows counts of groundtruth pairwise constraints through the proposed pair matching strategy from all the training samples.

Count	object	pair	paired object
Car	14357	11110	13620
Pedestrian	2207	1187	1614
Cyclist	734	219	371

Table 1: Count of objects, pairs and paired objects of each category in the KITTI training set.

4.1. Training

We adopt the modified DLA-34 [43] as our backbone. The resolution of the input image is set to 380 × 1280. The feature map of the backbone output is with a size of $96 \times 320 \times 64$. Each of the eleven output branches connects the backbone feature with two additional convolution layers with sizes of $3 \times 3 \times 256$ and $1 \times 1 \times m$, where m is the feature channel of the related output branch. Convolution layers connecting output branches maintain the same feature width and height. Thus, the feature size of each output branch is $96 \times 320 \times m$.

We train the whole network in an end-to-end manner for 70 epochs with a batch-size of 32 on four GPUs simultaneously. The initial learning rate is 1.25e-4, dropped by multiplying 0.1 both at 45 and 60 epochs. It is trained with Adam optimizer with weight decay as 1e-5. We conduct different data augmentation strategies during training, as random cropping and scaling for 2D detection, and random horizontal flipping for both 3D detection and pairwise constraints prediction.

4.2. Evaluation

Following [36], we use 40-point interpolated average precision metric AP_{40} that averaging precision results on 40 recall positions except the one where recall is 0. The previous metric AP_{11} of KITTI3D average precision on 11 recall positions, which may trigger bias to some extent. The precision is evaluated at both the bird-eye view 2D box AP_{bv} and the 3D bounding box AP_{3D} in world space. We report average precision with intersection over union (IoU) using both 0.5 and 0.7 as thresholds.

For the evaluation and ablation study, we show experimental results from three different setups. **Baseline** is derived from CenterNet [44] with an additional output branch to represent the offset of the 3D projected center to the located keypoint. $+\sigma^z + \sigma^{uv}$ adds two uncertainty prediction branches on **Baseline** which consists of all the three 2D detection branches and six 3D detection branches as shown in Figure 1. **MonoPair** is the final proposed method integrating the eleven prediction branches and the pairwise spatial constraint optimization.

Methods	AP_{bv} IoU≥0.5			AP_{3D} IoU≥0.5			AP_{bv} IoU≥0.7			AP_{3D} IoU≥0.7			RT
	E	M	H	E	M	H	E	M	H	E	M	H	(ms)
CenterNet[44]*	34.36	27.91	24.65	20.00	17.50	15.57	3.46	3.31	3.21	0.60	0.66	0.77	**45**
MonoDIS[36]	-	-	-	-	-	-	18.45	12.58	10.66	11.06	7.60	6.37	-
MonoGRNet[31]*	52.13	35.99	28.72	47.59	32.28	25.50	19.72	12.81	10.15	11.90	7.56	5.76	60
M3D-RPN[3]*	53.35	39.60	31.76	48.53	35.94	28.59	20.85	15.62	11.88	14.53	11.07	8.65	161
Baseline	53.06	38.51	32.56	47.63	33.19	28.68	19.83	12.84	10.42	13.06	7.81	6.49	47
$+\sigma^z + \sigma^{uv}$	59.22	46.90	41.38	53.44	41.46	36.28	21.71	17.39	15.10	14.75	11.42	9.76	50
MonoPair	**61.06**	**47.63**	**41.92**	**55.38**	**42.39**	**37.99**	**24.12**	**18.17**	**15.76**	**16.28**	**12.30**	**10.42**	57

Table 2: AP_{40} scores on KITTI3D validation set for car. * indicates that the value is extracted by ourselves from the public pretrained model or results provided by related paper author. E, M and H represent *Easy*, *Moderate* and *Hard* samples.

Methods	AP_{2D}			AOS			AP_{bv}			AP_{3D}		
	E	M	H	E	M	H	E	M	H	E	M	H
MonoGRNet[31]	88.65	77.94	63.31	-	-	-	18.19	11.17	8.73	9.61	5.74	4.25
MonoDIS[36]	94.61	89.15	78.37	-	-	-	17.23	13.19	11.12	10.37	7.94	6.40
M3D-RPN[3]	89.04	85.08	69.26	88.38	82.81	67.08	**21.02**	13.67	10.23	**14.76**	9.71	7.42
MonoPair	**96.61**	**93.55**	**83.55**	**91.65**	**86.11**	**76.45**	19.28	**14.83**	**12.89**	13.04	**9.99**	**8.65**

Table 3: AP_{40} scores on KITTI3D test set for car referred from the KITTI benchmark website.

Cat	Method	AP_{bv}			AP_{3D}		
		E	M	H	E	M	H
Ped	M3D-RPN[3]	5.65	4.05	3.29	4.92	3.48	2.94
	MonoPair	10.99	7.04	6.29	10.02	6.68	5.53
Cyc	M3D-RPN[3]	1.25	0.81	0.78	0.94	0.65	0.47
	MonoPair	4.76	2.87	2.42	3.79	2.12	1.83

Table 4: AP_{40} scores on pedestrian and cyclist samples from the KITTI3D test set at 0.7 IoU threshold. It can be referred from the KITTI benchmark website.

5. Experimental Results

5.1. Quantitative and Qualitative Results

We first show the performance of our proposed MonoPair on KITTI3D validation set for car, compared with other state-of-the-art (SOTA) monocular 3D detectors including MonoDIS [36], MonoGRNet [31] and M3D-RPN [3] in Table 2. Since MonoGRNet and M3D-RPN have not published their results through AP_{40}, we evaluate the related values through their published detection results or models.

As shown in Table 2, although our baseline is only comparable or a little worse than SOTA detector M3D-RPN, MonoPair outperforms all the other detectors mostly by a large margin, especially for *hard* samples with augmentations from the uncertainty and the pairwise spatial constraint. Table 3 shows results of our MonoPair on the KITTI3D test set for car. From the KITTI 3D object detection benchmark[1], we achieve the highest score for *Moderate* samples and rank at the first place among those 3D monocular object detectors without using additional information. AP_{2D} and AOS are metrics for 2D object detection and orientation estimations following the benchmark. Apart from the *Easy* result of AP_bv and AP_{3D}, our method outperforms M3D-RPN for a large margin, especially for *Hard* samples. It proves the effects of the proposed pairwise constraint optimization targeting for highly occluded samples.

We show the pedestrian and cyclist detection results on the KITTI test set in Table 4. Because MonoDIS [36] and MonoGRNet [31] do not report their performance on pedestrian and cyclist categories, we only compare our method with M3D-RPN [3]. It presents a significant improvement from our MonoPair. Even though the relatively few training samples of pedestrian and cyclist, the proposed pairwise spatial constraint goes much deeper by utilizing object relationships compared with target-independent detectors.

Besides, compared with those methods relying on time-consuming region proposal network [3, 36], our one-stage anchor-free detector is more than two times faster on an Nvidia GTX 1080 Ti. It can perform inference in real-time as 57 ms per image, as shown in Table 2.

5.2. Ablation Study

We conduct two ablation studies for different uncertain terms and the count of pairwise constraints both on KITTI3D validation set through AP_{40}. We only show results from *Moderate* samples here.

[1] http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d

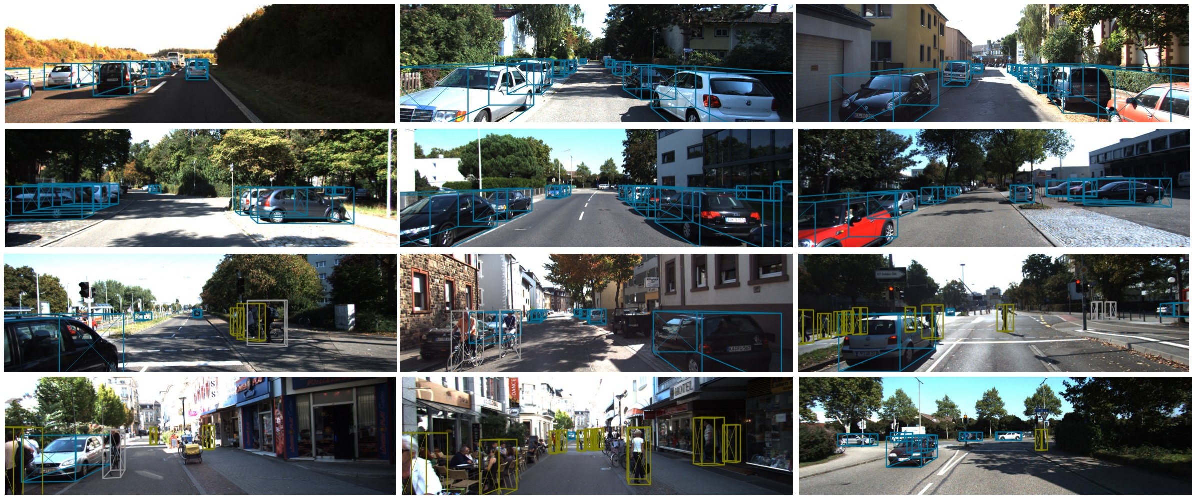


Figure 7: Qualitative results in KITTI validation set. Cyan, yellow and grey mean predictions of car, pedestrian and cyclist.

Uncertainty	IoU≥0.5		IoU≥0.7	
	AP_{bv}	AP_{3D}	AP_{bv}	AP_{3D}
Baseline	38.51	33.19	12.84	7.81
$+\sigma^{uv}$	42.79	38.75	14.38	8.96
$+\sigma^{z}$	45.09	40.46	15.79	10.15
$+\sigma^{z}+\sigma^{uv}$	46.90	41.46	17.39	11.42

Table 5: Ablation study for different uncertainty terms.

pairs	images	AP_{bv}		AP_{3D}	
		Uncert.	MonoPair	Uncert.	MonoPair
0-1	1404	10.40	10.44	5.41	6.02
2-4	1176	13.25	14.00	8.46	8.97
5-8	887	20.45	22.32	14.63	15.54
9-	302	25.49	25.87	17.98	18.94

Table 6: Ablation study for improvements among different pair counts through 0.7 IoU.

For uncertainty study, except the **Baseline** and $+\sigma^z + \sigma^{uv}$ setups mentioned above, we add σ^z and σ^{uv} methods by only predict the depth or projected offset uncertainty based on the **Baseline**. From Table 5, uncertainties prediction from both depth and offset show considerable development above the baseline, where the improvement from depth is larger. The results match the fact that depth prediction is a much more challenging task and it can benefit more from the uncertainty term. It proves the necessity of imposing uncertainties for 3D object prediction, which is rarely considered by previous detectors.

In terms of the pairwise constraint, we divide the validation set to different parts based on the count of groundtruth pairwise constraints. The **Uncert.** in Table 6 represents $+\sigma^z + \sigma^{uv}$ for simplicity. By checking both the AP_{bv} and AP_{3D} in Table 6, the third group with 5 to 8 pairs shows higher average precision improvement. A possible explanation is that fewer pairs may not provide enough constraints, and more pairs may increase the complexity of the optimization.

Also, to prove the utilization of using uncertainties to weigh related errors, we tried various strategies for weight matrix designing, for example, giving more confidence for objects closed to the camera or setting the weight matrix as identity. However, none of those strategies showed improvements in the detection performance. On the other hand, the baseline is easily dropped to be worse because of coarse post-optimization. It shows that setting the weight matrix of the proposed spatial constraint optimization is nontrivial. And uncertainties, besides its original function to enhance network training, is naturally a meaningful choice for weights of different error terms.

6. Conclusions

We proposed a novel post-optimization method for 3D object detection with uncertainty-aware training from a monocular camera. By imposing aleatoric uncertainties into the network and considering spatial relationships for objects, our method has achieved the state-of-the-art performance on KITTI 3D object detection benchmark using a monocular camera without additional information. By exploring the spatial constraints of object pairs, we observed the enormous potential of geometric relationships in object detection, which was rarely considered before. For future work, finding spatial relationships across object categories and innovating pair matching strategies would be exciting next steps.

References

[1] Ivan Barabanau, Alexey Artemov, Evgeny Burnaev, and Vyacheslav Murashkin. Monocular 3d Object Detection via Geometric Reasoning on Keypoints. *arXiv:1905.05618 [cs]*, May 2019. arXiv: 1905.05618.

[2] Lorenzo Bertoni, Sven Kreiss, and Alexandre Alahi. Monoloco: Monocular 3d pedestrian localization and uncertainty estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[3] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[4] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teulière, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2040–2049, 2017.

[5] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d Object Detection for Autonomous Driving. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2147–2156, Las Vegas, NV, USA, June 2016. IEEE.

[6] 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.

[7] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d Object Proposals Using Stereo Imagery for Accurate Object Class Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1259–1272, May 2018.

[8] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[9] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving. *arXiv:1904.04620 [cs]*, Apr. 2019. arXiv: 1904.04620.

[10] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3076–3086, 2017.

[11] 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.

[12] Di Feng, Lars Rosenbaum, and Klaus Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3266–3273. IEEE, 2018.

[13] Yarin Gal. *Uncertainty in deep learning*. PhD thesis, PhD thesis, University of Cambridge, 2016.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[15] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

[16] Alex Guy Kendall. *Geometry and uncertainty in deep learning for computer vision*. PhD thesis, University of Cambridge, 2019.

[17] Jason Ku, Alex D. Pon, and Steven L. Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[18] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613. IEEE, 2011.

[19] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3559–3568, 2018.

[20] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7644–7652, 2019.

[21] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. RTM3D: real-time monocular 3d detection from object keypoints for autonomous driving. *arXiv:2001.03343 [cs]*, 2020.

[22] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[23] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep Continuous Fusion for Multi-sensor 3d Object Detection. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pages 663–678. Springer International Publishing, 2018.

[24] Zechen Liu, Zizhang Wu, and Roland Tóth. SMOKE: single-stage monocular 3d object detection via keypoint estimation. *arXiv:2002.10111 [cs]*, 2020.

[25] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[26] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[27] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d Bounding Box Estimation Using Deep Learning and Geometry. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5632–5640, Honolulu, HI, July 2017. IEEE.

[28] J Krishna Murthy, GV Sai Krishna, Falak Chhaya, and K Madhava Krishna. Reconstructing vehicles from a single image: Shape priors for road scene understanding. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 724–731. IEEE, 2017.

[29] Cuong Cao Pham and Jae Wook Jeon. Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks. *Signal Processing: Image Communication*, 53:110–122, Apr. 2017.

[30] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[31] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8851–8858, 2019.

[32] Zengyi Qin, Jinglu Wang, and Yan Lu. Triangulation learning network: From monocular to stereo 3d object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

[34] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[35] Kiwoo Shin, Youngwook Paul Kwon, and Masayoshi Tomizuka. Roarnet: A robust 3d object detection based on region approximation refinement. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2510–2515. IEEE, 2019.

[36] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel Lopez-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[37] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[38] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3d Object Detection for Autonomous Driving. *arXiv:1812.07179 [cs]*, Dec. 2018. arXiv: 1812.07179.

[39] Sascha Wirges, Marcel Reith-Braun, Martin Lauer, and Christoph Stiller. Capturing Object Detection Uncertainty in Multi-Layer Grid Maps. *arXiv:1901.11284 [cs]*, Jan. 2019. arXiv: 1901.11284.

[40] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Subcategory-aware Convolutional Neural Networks for Object Proposals and Detection. *arXiv:1604.04693 [cs]*, Mar. 2017. arXiv: 1604.04693.

[41] Bin Xu and Zhenzhong Chen. Multi-level Fusion Based 3d Object Detection from Monocular Images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2345–2353, Salt Lake City, UT, USA, June 2018. IEEE.

[42] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[43] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018.

[44] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as Points. *arXiv:1904.07850 [cs]*, Apr. 2019. arXiv: 1904.07850.

[45] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.