Diversity Matters: Fully Exploiting Depth Clues for Reliable Monocular 3D Object Detection

Zhuoling Li[1]*, Zhan Qu[2]*†, Yang Zhou[2], Jianzhuang Liu[2], Haoqian Wang[1], Lihui Jiang[2]

[1]Tsinghua University [2]Huawei Noah's Ark Lab

lzl20@mails.tsinghua.edu.cn {quzhan, zhouyang116, liu.jianzhuang, jianglihui1}@huawei.com

wanghaoqian@tsinghua.edu.cn

Abstract

As an inherently ill-posed problem, depth estimation from single images is the most challenging part of monocular 3D object detection (M3OD). Many existing methods rely on preconceived assumptions to bridge the missing spatial information in monocular images, and predict a sole depth value for every object of interest. However, these assumptions do not always hold in practical applications. To tackle this problem, we propose a depth solving system that fully explores the visual clues from the subtasks in M3OD and generates multiple estimations for the depth of each target. Since the depth estimations rely on different assumptions in essence, they present diverse distributions. Even if some assumptions collapse, the estimations established on the remaining assumptions are still reliable. In addition, we develop a depth selection and combination strategy. This strategy is able to remove abnormal estimations caused by collapsed assumptions, and adaptively combine the remaining estimations into a single one. In this way, our depth solving system becomes more precise and robust. Exploiting the clues from multiple subtasks of M3OD and without introducing any extra information, our method surpasses the current best method by more than 20% relatively on the Moderate level of test split in the KITTI 3D object detection benchmark, while still maintaining real-time efficiency.

1. Introduction

Significant attention has been drawn by 3D object detection due to its widespread applications in autonomous driving and robotic navigation [2, 13, 14, 42]. Inaccurate detection affects the motion planning process directly and could lead to serious accidents. Therefore, the industry has great demand for precise and robust 3D object detection systems.

(a) Direct estimation (1 depth): 0.6220   (b) Depth from height (3 depths): 0.5831

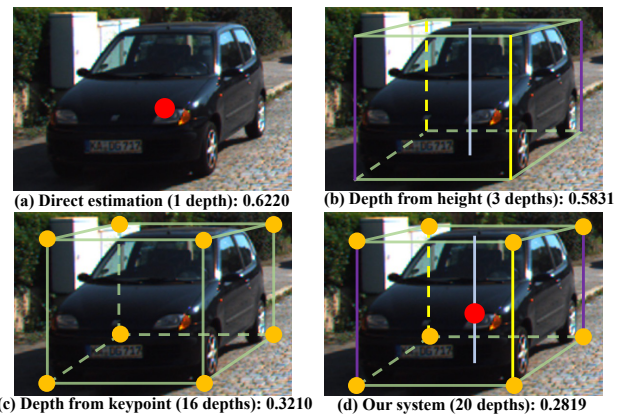(c) Depth from keypoint (16 depths): 0.3210   (d) Our system (20 depths): 0.2819

Figure 1. Comparison among various depth solving strategies with different diversity levels. The value below each image is the mean absolute error of depth estimation on the KITTI validation set [12] if we always select the most accurate depth from multiple produced estimations. As shown, the error decreases as the diversity of estimations increases.

Many recently proposed 3D object detection algorithms heavily rely on LiDARs [47] and stereo cameras [18], because they are able to perceive the depth information of surroundings directly. Nevertheless, LiDAR sensors are expensive while stereo cameras require exact online calibration [22]. These limitations make 3D perception using only monocular images promising because it is economical and flexible for deployment.

The monocular 3D object detection community has achieved prominent progress in recent years. However, there still exists a huge performance gap between the monocular and LiDAR-based methods. This gap is caused by the fact that accurate localization of 3D objects relies on precise depth estimation, and predicting depth from monocular images is an inherently ill-posed problem [25], which means the information contained in a single image is insufficient for determining the depths of objects. To compensate the lack of information, current detectors usually resort to some preconceived assumptions. For example, SMOKE [21] assumes that depth can be inferred from visual pixels

---

*Zhuoling Li and Zhan Qu contributed equally. This work was done when Zhuoling Li was an intern at Huawei Noah's Ark Lab.

†Corresponding author.

directly. MonoRCNN [39] hypothesizes the height of a target can be estimated precisely and the camera is an ideal pinhole imaging model [37]. Nevertheless, these assumptions do not always hold. When the assumptions fail, the single depth produced by a method becomes unreliable.

To address the aforementioned problem, we develop a depth solving system that provides diverse depth estimations for every target. Different from MonoFlex [45] that only utilizes limited information (direct estimation and the heights of objects) and generates similar depths, our method fully exploits various attribute combinations (direct estimation, keypoint, orientation and dimension of an object) to produce 20 depths, which present diverse distributions. Besides, since the 20 depths are separately obtained by solving 20 equations built upon different assumptions, part of the depths are still precise when some assumptions collapse. Figure 1 illustrates the importance of diversity for monocular depth estimation in the condition that the most accurate one can be selected from predicted depths. When only direct estimation (1 depth) is applied, the mean absolute error (MAE) of depth estimation is 0.6220. In contrast, utilizing our depth solving system, the MAE decreases to 0.2819.

Although the depths produced by our depth solving system include promising estimations, they also contain outliers. The following problem is how to select promising estimations and combine them into a single value. To that end, we devise a strategy that removes outliers iteratively and integrates the remaining depths based on uncertainty. The experimental results in Section 5.3 suggest that this strategy is crucial for the overall performance.

Last but not least, considering the uncertainty of both the combined depth and 3D box vertexes, we propose a new scheme, named 3D geometry confidence, to model the conditional 3D confidence. Compared with existing strategies such as modeling the confidence with 3D IOU [6], our scheme generalizes better.

Incorporating all the techniques, the resulting Monocular 3D detector with diverse depth estimations, named MonoDDE, fully exploits depth clues in monocular images and produce reliable 3D detection boxes in practical applications. Our main contributions are summarized as follows:

• We point out that the diversity of depth estimation is critical for monocular 3D object detection. Correspondingly, a novel depth solving system that produces 20 depths for every target is developed.

• We devise a strategy that removes outliers caused by collapsed assumptions and combines the remaining reliable estimations into a single depth. Besides, a new scheme for modeling the conditional 3D confidence is developed.

• Using a single model, MonoDDE outperforms the current best method by 20.96% relatively on the Moderate level of the Car class in KITTI, and ranks 1st and 2nd on the Cyclist and Pedestrian classes, respectively.

2. Related Work

Monocular 3D object detection. According to the form of generated depth, recent monocular 3D object detection algorithms can be mainly categorized into two classes: dense-depth and sparse-depth methods.

Dense-depth 3D detectors generate depth values for every pixel in an image. The generated dense depth map can be combined with the original RGB image as input to a model for producing 3D object detection boxes [25,28,38]. Alternatively, it can also be converted to pseudo 3D point clouds firstly and then a LiDAR-based 3D detector is applied on them to derive the results [26,33,35]. Although the dense-depth methods have achieved impressive results, estimating pixel-wise depths is challenging and requires more complex backbones compared with the strategy that only predicts the depths of several keypoints. This issue has hindered dense-depth methods from further improvement to some extent [51].

Sparse-depth methods only produce one valid depth for every recognized target. Their network structures mostly follow some outstanding 2D detectors, such as Faster RCNN [36] and CenterNet [48]. Early sparse-depth methods rely on generating numerous anchors heavily and utilize the information contained in the anchors to regress desired object properties [7,8,28]. However, the anchor-generating process introduces non-negligible noise and increases computation burden [21]. Recent sparse-depth 3D detectors are mainly center-based [22,49], which represent objects by their 2D centers [21] or projected 3D centers [27]. This anchor-free structure has led to simpler model structures, fewer hyper-parameters and better detection precision [20]. Our proposed MonoDDE is also center-based.

Sparse depth estimation. Experimental results in previous works have shown that depth estimation is the most crucial step in center-based methods [45], and existing sparse depth estimation can be roughly divided into 3 strategies, *direct depth estimation* [21], *depth from height* [39] and *pespective-n-point* (PnP) [19].

Among the three strategies, *direct depth estimation* is the easiest for implementation. Taking monocular images as input, it completely relies on a deep neural network to explore visual clues and infer depths [21,49]. Besides, since *direct depth estimation* does not require manual annotation, its precision can be improved conveniently via large-scale self-supervised pre-training without labels [31]. Nevertheless, since monocular depth estimation is an ill-posed problem, the estimated values are not reliable when there exists a significant domain gap between training and testing images.

Depth from height computes depths based on the pixel heights and estimated physical heights of targets [39]. Since the physical heights of objects belonging to the same category are similar, *depth from height* generalizes better than

direct depth estimation [22]. However, estimating physical heights is still an ill-posed problem.

In contrast to *direct depth estimation* and *depth from height*, PnP incorporates all the dimension, orientation and keypoint information of an object to construct geometric constraints [19, 20, 22] and uses the *least squares method* [29] to obtain its location. Therefore, PnP exploits the information more efficiently. However, all the equations in PnP are closely coupled with each other [22]. This issue causes the difficulty to model the uncertainty of every depth individually.

3. Preliminary

To present our method clearly, we first review the target of monocular 3D object detection. Afterwards, the mathematical forms of the three depth estimation strategies mentioned in Section 2 are given, which are *direct depth estimation*, *depth from height* and PnP.

3.1. Monocular 3D Object Detection

Given a single image, monocular 3D object detection aims to find every object of interest, identify its category and estimate a 3D box B that contains the object properly. The 3D box B can be further divided into 3 properties, i.e., the 3D center location (x, y, z), dimension (h, w, l) and orientation (yaw angle) θ. The roll and pitch angles of objects are set to 0 following the KITTI [12] setting.

Among these properties, the dimension and orientation are strongly related to the visual appearance and can be learned by a network [15], while the 3D location is challenging to obtain. This is because producing an accurate 3D location is built upon the premise of precise depth estimation. Thus, how to estimate the depth correctly is the most important research topic in monocular 3D object detection.

3.2. Depth Estimation Strategies

Direct depth estimation. Given an input image I, *direct depth estimation* relies on the appearance of an object and its surrounding pixels to regress depth z directly. Afterwards, utilizing the projected 3D center estimation (u_c, v_c), x and y are determined as:

$$x = \frac{(u_c - c_u)z}{f_x}, \quad y = \frac{(v_c - c_v)z}{f_y}, \quad (1)$$

where (c_u, c_v) represents the coordinate of the principle point, and f_x and f_y are the horizontal and vertical focal lengths, respectively.

Depth from height. The *depth from height* strategy tackles depth estimation by decoupling it as predicting the physical height h and pixel height h' of an object. The process of computing z given h and h' can be formulated as:

$$z = \frac{f_y h}{h'}. \quad (2)$$

After obtaining z, x and y are calculated using Eq. (1).

Pespective-n-point. Since objects in 3D object detection are represented as cuboids, we can use their geometric constraints to obtain their 3D locations based on the *least squares method*.

Denoting the position of a 3D keypoint under the object coordinate system as $\mathbf{P}^o = (x^o, y^o, z^o)^T$, it can be transformed to the camera coordinate system with respect to the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$ as:

$$[x^c, y^c, z^c]^T = \mathbf{R}[x^o, y^o, z^o]^T + \mathbf{T}, \quad (3)$$

where $\mathbf{P}^c = (x^c, y^c, z^c)^T$ represents the location of this 3D point under the camera coordinate system, and

$$\mathbf{R} = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} x, & y, & z \end{bmatrix}^T. \quad (4)$$

Afterwards, given the camera intrinsic matrix $\mathbf{K}$, we can project $\mathbf{P}^c$ to a point in the 2D pixel coordinate system as (u, v):

$$\lambda[u, v, 1]^T = \mathbf{K}[x^c, y^c, z^c]^T, \quad (5)$$

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_u \\ 0 & f_y & c_v \\ 0 & 0 & 1 \end{bmatrix}, \quad \lambda = z^c. \quad (6)$$

Hereby, the geometric relations between any point in the object coordinate system and its corresponding pixel on the 2D imaging plane are described by Eqs. (3)–(6). In these relations, $\mathbf{P}^o$ is pre-defined manually, $\mathbf{K}$ is known, and $\mathbf{R}$ and (u, v) are estimated by a network. Thus, $\mathbf{T} = [x, y, z]^T$ contains the only variables waiting to be computed. Since every 3D keypoint provides 2 geometric constraints, we can obtain x, y and z simultaneously using the *least squares method* if we have at least 2 keypoints.

4. Method

This section details our proposed method and how MonoDDE is implemented.

4.1. Overall Framework

The overall framework of MonoDDE is illustrated in Figure 2. MonoDDE employs CenterNet [48] as the base model for producing discriminative representation. Specifically, for any input image I, DLA34 [44] is adopted as the backbone of CenterNet for extracting features. We establish several network heads to regress object properties, including categorical heatmap, 2D bounding box, dimension, keypoint offsets, orientation, depth and multiple uncertainty items. Based on the regressed properties, our proposed depth solving system produces 20 diverse depths in different ways. Subsequently, the developed robust depth
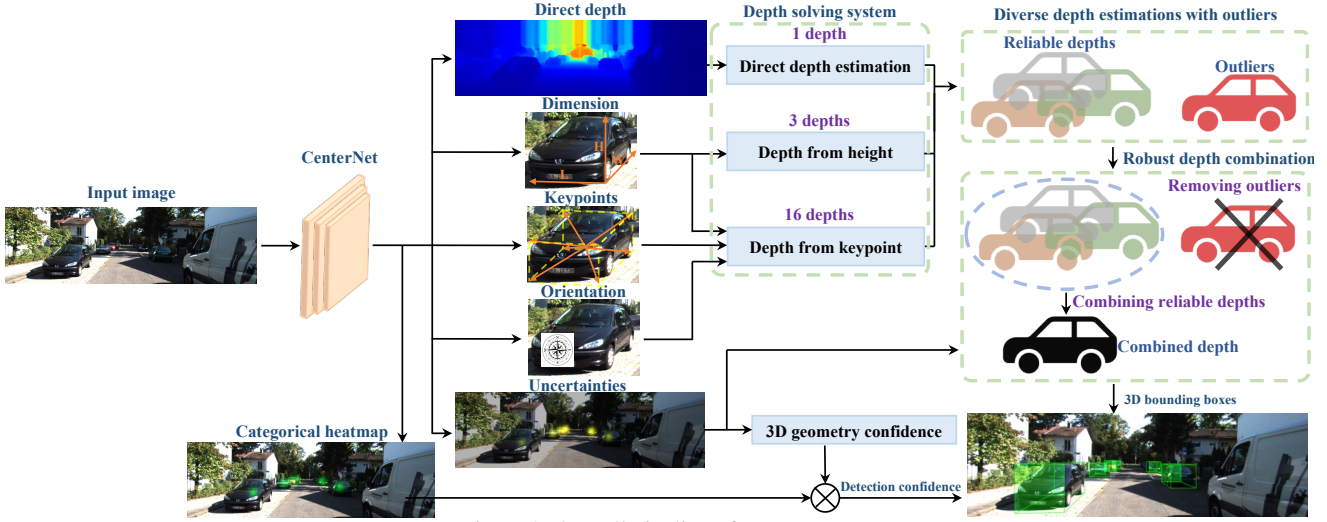
Figure 2. Overall pipeline of MonoDDE.

combination module filters out outlier values and combines the remaining estimations as a single depth. Taking this depth value into Eq. (1), we get the location of the target and further its 3D box with the regressed dimension and orientation. In addition, the detection confidence obtained by our 3D geometry confidence (Section 4.4) is responsible for modeling the probability that a target is recognized correctly.

4.2. Diverse Depth Estimations

In this work, we expect our developed depth solving system to possess three key characteristics: (1) It should concentrate on obtaining depth z rather than computing x, y and z together. (2) In contrast to existing methods, it should produce multiple and diverse estimation values. (3) It should make full use of all available information, including visual clue, estimated target center, dimension, orientation and keypoints.

To realize the above goal, we first revisit the geometric constraints described in Section 3.2. Combining Eqs. (3)–(6), we can simplify the relation between a 3D keypoint under the object coordinate system $P^o = (x^o, y^o, z^o)$ and its corresponding pixel (u, v) as:

$$
\begin{bmatrix} -1 & 0 & \tilde{u} \\ 0 & -1 & \tilde{v} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} \mathbf{A} + \mathbf{B}, \tag{7}
$$

where

$$
\tilde{u} = \frac{u - c_u}{f_x}, \quad \tilde{v} = \frac{v - c_v}{f_y}, \tag{8}
$$

$$
\mathbf{A} = \begin{bmatrix} x^o sin\theta - z^o cos\theta \end{bmatrix}, \tag{9}
$$

$$
\mathbf{B} = \begin{bmatrix} x^o cos\theta + z^o sin\theta \\ y^o \end{bmatrix}. \tag{10}
$$

It can be observed from Eq. (7) that x, y and z appear in the same equation, which hinders this system from only obtaining z. In order to solve this problem, we need to resort to some extra prior knowledge.

Through experiments, we observe that most centers of objects can be recognized precisely. More than 85% of estimated object centers fall within 1 pixel around their corresponding ground truth points. Hence, Eq. (1) can be used as the prior. By inserting Eq. (1) into Eq. (7), Eq. (7) can be reformulated as:

$$
(\tilde{u} - \tilde{u}_c)z = \mathbf{A}\tilde{u} + x^o \cos \theta + z^o \sin \theta, \tag{11}
$$

$$
(\tilde{v} - \tilde{v}_c)z = \mathbf{A}\tilde{v} + y^o, \tag{12}
$$

where $\tilde{u}_c = \frac{u_c - c_u}{f_x}$ and $\tilde{v}_c = \frac{v_c - c_v}{f_y}$.

In this way, Eq. (7) is decoupled into two independent equations, Eqs. (11) and (12), which focus on solving for z. The geometric relation between every 3D vertex and its corresponding projected pixel can result in 2 separate depths. In our implementation, as shown in Figure 3 (a), we select 8 vertexes of a 3D box as the keypoints to calculate the depths, which provide 16 diverse estimation values.

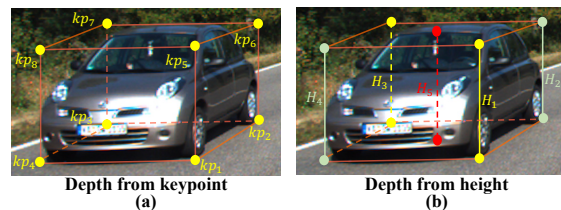Figure 3. Depths from keypoints and heights.

Furthermore, we incorporate the *direct depth estimation* and *depth from height* strategies into the depth solving system. Specifically, *direct depth estimation* regresses 1 depth value of the projected 3D center like [21]. For *depth from height*, as shown in Figure 3 (b), we split the heights of the center vertical line and corner vertical lines into three

groups, $\{H_5\}$, $\{H_1, H_3\}$, and $\{H_2, H_4\}$, which is similar to [45]. The depth of an object can be obtained using the center vertical line H_5 and Eq. (2) or by averaging the depths generated using the opposite corner vertical lines ($\{H_1$ and $H_3\}$ or $\{H_2$ and $H_4\}$).

Hereby, we have established a depth solving system that can output 20 diverse depths, 16 from our newly proposed geometric constraints (*depth from keypoint*), 1 from *direct depth estimation*, and 3 from *depth from height*. The following problem is how to select reliable depths from them.

4.3. Robust Depth Combination

In this subsection, we present the strategies for selecting and combining promising depths.

Output distribution. Assuming each estimated depth follows the Gaussian distribution [24], the model learns to predict the mean and variance of this distribution by minimizing:

$$L_\sigma = \frac{|p - p^*|}{\sigma} + log\sigma, \tag{13}$$

where p and σ are the predicted mean and standard deviation of the output distribution, respectively, and p^* represents the ground truth. Note that σ is learned implicitly from Eq. (13) without the need of ground truth. More details are given in [10, 16] about why the distribution can be captured by the network in this way.

Moreover, we define the distribution of a set $S = \{s_i\}_{i=1}^N$, which contains N Gaussian distribution variables $s_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, as a new Gaussian distribution, because all the heads predict the depths of the same target. It is the weighted sum of $\{s_i\}_{i=1}^N$ and the weights $\{\omega_i\}_{i=1}^N$ are derived via the *weighted least squares method* [3]:

$$\omega_i = \frac{1/\sigma_i^2}{\sum_{j=1}^N 1/\sigma_j^2}. \tag{14}$$

Hence, the mean μ_s and variance σ_s^2 of S are calculated as:

$$\mu_s = \sum_{i=1}^N \omega_i\mu_i, \ \sigma_s^2 = \sum_{i=1}^N \omega_i^2\sigma_i^2. \tag{15}$$

Selecting and combining reliable depths. We first train our model to predict the means and variances of the 20 depth distributions using Eq. (13), and compose the 20 distributions as the set S. Since both S and its contained variables $\{s_i\}_{i=1}^N$ are treated as Gaussian distributions, we can filter out outliers based on the 3σ rule [32], and devise a robust algorithm similar to the expectation-maximization (EM) algorithm [11].

Algorithm 1 Robust depth selection and combination.

Input: Estimated depths $\{z_i\}_{i=1}^N$ and their corresponding variances $\{\sigma_i^2\}_{i=1}^N$
1: Initialize an empty set $S = \varnothing$
2: Put z_k to S if $\sigma_k^2 = \min\{\sigma_1^2, \sigma_2^2, ..., \sigma_N^2\}$
3: **while** True **do**
4:     Update μ_s and σ_s^2 according to Eqs. (14)–(15)
5:     $S_{new} = \varnothing$
6:     **for** $i = 1$ to N **do**
7:         $S_{new} \leftarrow S_{new} \cup \{z_i\}$ if $z_i \in (\mu_s - 3\sigma_s, \mu_s + 3\sigma_s)$ and $z_i \notin S$.
8:     **end for**
9:     **if** S_{new} is $\varnothing$ **then**
10:         break
11:     **end if**
12:     $S \leftarrow S \cup S_{new}$
13: **end while**
Output: μ_s

In this algorithm, we first initialize S as an empty set and put the depth with minimum variance to S. For the maximization step, μ_s and σ_s^2 are updated using Eqs. (14)–(15). During the expectation step, the depths that fall into $(\mu_s - 3\sigma_s, \mu_s + 3\sigma_s)$ are added to S. We repeat the maximization and expectation steps until μ_s and σ_s^2 converge. Afterwards, all depths falling out of $(\mu_s - 3\sigma_s, \mu_s + 3\sigma_s)$ are regarded as outliers and removed.

In this way, the reliable depths are contained in S. We directly employ the final μ_s as the combined depth z_c for subsequent operations. The pseudo code of the robust depth combination is given in Algorithm 1.

4.4. 3D Geometry Confidence

Let P_m be the probability (also called confidence) that a target is detected correctly. Following [20] with the probability chain rule, it is factorized into two items:

$$P_m = P_{3d|2d} \cdot P_{2d}, \tag{16}$$

where P_{2d} is represented by the categorical heatmap score and $P_{3d|2d}$ denotes the conditional 3D confidence. Previous methods often model $P_{3d|2d}$ with 3D IOU [6, 40, 43]. However, since the training images are used to train the model and the validation images are unseen, the mean 3D box IOU of the model on the training images is significantly higher than that on the validation images. Due to the large IOU gap, directly employing 3D IOU in the training stage to train the network and regarding the predicted IOU as $P_{3d|2d}$ lead to poor results in the validation stage. Meanwhile, some works have indicated that models trained with implicit supervision generalize better [46]. Hence, we model $P_{3d|2d}$ based on the estimated variance in Eq. (13), which is implicitly learned. Specifically, following [45], we define the

confidence d of an estimation item with respect to its variance σ^2 as:

$$d = 1 - \min\{\sigma^2, 1\}. \tag{17}$$

In this work, we model $P_{3d|2d}$ as the weighted sum of two items, the combined depth confidence d_c and the 3D box confidence d_b:

$$P_{3d|2d} = \omega_c d_c + \omega_b d_b, \tag{18}$$

where ω_c and ω_b are calculated based on σ_c^2 and σ_b^2 using Eq. (14). $P_{3d|2d}$ in Eq. (18) is our devised 3D geometry confidence.

The combined depth variance σ_c^2 for determining d_c is learned with Eq. (13). We do not directly use σ_s^2 as σ_c^2 because we observe that the estimated σ_c^2 leads to a more precise value. Meanwhile, similar to Eq. (13), the variance of the 3D box σ_b^2 is obtained via minimizing:

$$L_b = \frac{\sum\limits_{i=1}^{8} |v_i - v_i^*|}{\sigma_b} + log\sigma_b, \tag{19}$$

where $\{v_i\}_{i=1}^8$ denote the coordinates of the 8 3D box vertexes and $\{v_i^*\}_{i=1}^8$ are their ground truth.

4.5. Network Heads

This subsection describes the implementation of the detection heads briefly. Each head comprises two convolutional layers and one batch normalization layer.

Categorical heatmap. It is responsible for distinguishing the categories of objects and localizing target points. In this work, we employ projected 3D centers as the ground truth of the target points, and the representation decoupling strategy devised in MonoFlex [45] is adopted to tackle truncated objects. The loss function follows [21].

Orientation. Similar to [30], we regress the observation angle α instead of the yaw angle θ, and train the network with the MultiBin loss. α is split into 4 bins like [5], and then θ is obtained based on α.

Dimension. To be consistent with existing works, we predict the log-scale offsets of dimensions rather than directly outputting absolute sizes. Refer to [48] for details.

Keypoints. Following [45], MonoDDE regresses the offsets from target points to 10 pre-defined 2D keypoints, which include 8 vertexes, the bottom center and top center of a 3D bounding box.

Depth. This head is responsible for producing the direct estimation depth z. Notably, instead of estimating the absolute value of z directly, MonoDDE learns to fit its exponentially transformed form in [9].

Uncertainty. Based on Eq. (13), we enforce the network to capture the uncertainties (variances) of the 20 depth values, the combined depth z_c, and the 3D box.

5. Experiments

Dataset. Our method is evaluated on the KITTI 3D object detection benchmark [12], which comprises 7481 images for training and 7518 images for testing. Since the annotations of the testing data are not available, following [50], we further divide the training data into the training set (3712 images) and validation set (3769 images). Our reported detection classes include Car, Pedestrian and Cyclist. Besides, the objects in KITTI have been categorized into three difficulty levels (Easy, Moderate and Hard) according to their pixel heights, occlusion ratios, etc.

Evaluation metrics. The average precision (AP) of 3D bounding boxes and bird's-eye view (BEV) map are main metrics for comparing performance. Following [41], 40 recall positions are sampled to calculate AP. The IOU thresholds are 0.7 for Car, and 0.5 for Pedestrian and Cyclist.

Implementation details. MonoDDE is trained for 100 epochs with the initial learning rate 3e-4. The weights of the model are updated using the AdamW optimizer [23] and the learning rate is decayed at the 80th and 90th epochs [34]. The batch size is set to 8 and the whole training process is conducted on a single Tesla V100 GPU. Random horizontal flipping is the only augmentation operation.

5.1. Quantitative Results

We compare our method with recent SOTA counterparts of monocular 3D object detection on the KITTI benchmark. The detection results of the Car category are reported in Table 1, and the comparison on Pedestrian and Cyclist is given in Table 2. For the convenience of observation, the best and second-best results are in bold and underlined, respectively.

As shown in Table 1, taking monocular images as input, MonoDDE outperforms all other methods by large margins on both the testing and validation sets without introducing any extra information. For instance, MonoDDE surpasses Autoshape, a very recent SOTA method that utilizes CAD models as an extra clue, by 2.97% for $\text{AP}_{3D}70$ on the Moderate level. In other words, MonoDDE outperforms AutoShape by 20.96% (2.97÷14.17) relatively.

In Table 2, MonoDDE outperforms all sparse-depth methods (M3D-RPN, MonoPair, DFR-Net and MonoFlex) significantly. Although MonoDDE is slightly weaker than CaDDN (a pseudo-LiDAR method) for the Pedestrian class, MonoDDE is much faster (MonoDDE: 0.04s/image vs. CaDDN: 0.63s/image). We speculate that MonoDDE does not behave the best for Pedestrian because pedestrians are non-rigid and much smaller compared with cars. Therefore, it is difficult to recognize the keypoints of pedestrians, while the pseudo-LiDAR methods do not suffer from this issue.

5.2. Ablation Study on Depth Estimation

This subsection aims to study how various depth estimation methods affect the 3D object detection precision. To

Method	Depth	Extra	Test, $AP_{3D}70$ (%)			Test, $AP_{BEV}70$ (%)			Val, $AP_{3D}70$ (%)			Time (s)
			Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	
M3D-RPN [4]	E	-	14.76	9.71	7.42	21.02	13.67	10.23	14.53	11.07	8.65	0.16
SMOKE [21]	E	-	14.03	9.76	7.84	20.83	14.49	12.75	-	-	-	0.03
MonoPair [9]	E	-	13.04	9.99	8.65	19.28	14.83	12.89	16.28	12.30	10.42	0.06
Monodle [27]	E	-	17.23	12.26	10.29	24.79	18.89	16.00	17.45	13.66	11.68	0.04
GrooMeD-NMS [17]	E	-	18.10	12.32	9.65	26.19	18.27	14.05	19.67	14.32	11.27	0.12
Kinematic3D [5]	E	Video	19.07	12.72	9.17	26.69	17.52	13.10	19.76	14.10	10.47	0.12
CaDDN [35]	E	Depth	19.17	13.41	11.46	27.94	18.91	17.19	23.57	16.31	13.84	0.63
DFR-Net [51]	E	Depth	19.40	13.63	10.35	28.17	19.17	14.84	<u>24.81</u>	<u>17.78</u>	14.41	0.18
MonoEF [49]	E	-	21.29	13.87	11.71	29.03	19.70	<u>17.26</u>	-	-	-	0.03
MonoRCNN [39]	H	-	18.36	12.65	10.03	25.48	18.11	14.10	16.61	13.19	10.65	0.07
RTM3D [20]	P	-	14.41	10.34	8.77	19.17	14.20	11.99	-	-	-	0.05
KM3D [19]	P	-	16.73	11.45	9.92	23.44	16.20	14.47	-	-	-	0.03
Autoshape [22]	P	CAD	<u>22.47</u>	<u>14.17</u>	11.36	<u>30.66</u>	<u>20.08</u>	15.95	20.09	14.65	12.07	0.04
MonoFlex [45]	EH	-	19.94	13.89	<u>12.07</u>	28.23	19.75	16.89	23.64	17.51	<u>14.83</u>	0.03
MonoDDE (ours)	EHK	-	**24.93**	**17.14**	**15.10**	**33.58**	**23.46**	**20.37**	**26.66**	**19.75**	**16.72**	0.04

Table 1. Performance comparison between MonoDDE and recent SOTAs on the Car class of KITTI. They are sorted according to their depth solving strategies shown in the 2nd column (E: Direct depth estimation, H: Depth from height, P: PnP, and K: Depth from keypoint).

Method	Test, $AP_{3D}50$ (%)					
	Pedestrian			Cyclist		
	Easy	Moderate	Hard	Easy	Moderate	Hard
M3D-RPN [4]	4.92	3.48	2.94	0.94	0.65	0.47
MonoPair [9]	10.02	6.68	5.53	3.79	2.12	1.83
CaDDN [35]	**12.87**	**8.14**	**6.76**	**7.00**	3.41	<u>3.30</u>
DFR-Net [51]	6.09	3.62	3.39	5.69	<u>3.58</u>	3.10
MonoFlex [45]	9.43	6.31	5.26	4.17	2.35	2.04
MonoDDE (ours)	<u>11.13</u>	<u>7.32</u>	<u>6.67</u>	<u>5.94</u>	**3.78**	**3.33**

Table 2. Performance comparison on the Pedestrian and Cyclist classes of KITTI.

Select	Combine	Val, $AP_{3D}70$ (%)			Val, $AP_{BEV}70$ (%)		
		Easy	Moderate	Hard	Easy	Moderate	Hard
None	Hard	25.71	19.13	16.39	34.30	25.72	22.39
None	Mean	18.08	14.31	12.34	24.60	19.10	16.71
None	Weighted	25.81	19.26	16.34	34.25	25.83	22.50
Min	Weighted	26.31	19.59	16.58	34.79	26.09	22.78
Iterative	Weighted	26.66	19.75	16.72	35.51	26.48	23.07
Oracle	None	49.96	38.73	33.06	58.69	43.96	37.65

Table 4. Analysis of the depth selection and combination strategies.

this end, we compare the performance of the model that predict depths based on different combinations of three strategies (*direct depth estimation*, *depth from height* and *depth from keypoint*). The model is trained on the KITTI training set and evaluated on the Car class of KITTI validation set. The results are reported in Table 3.

E	H	K	Val, $AP_{3D}70$ (%)			Val, $AP_{BEV}70$ (%)		
			Easy	Moderate	Hard	Easy	Moderate	Hard
✓			24.20	18.01	15.88	32.53	24.52	21.33
	✓		25.01	18.36	15.32	33.15	24.83	21.40
		✓	24.48	18.74	15.88	32.89	25.29	21.51
✓	✓		25.26	18.74	16.26	33.68	25.26	21.95
	✓	✓	24.48	18.82	15.96	33.69	25.47	22.22
✓		✓	25.64	19.18	16.29	34.14	25.65	22.43
✓	✓	✓	26.66	19.75	16.72	35.51	26.48	23.07

Table 3. Ablation study on depth estimation strategies (E: *direct depth estimation*, H: *depth from height*, and K: *depth from keypoint*). We highlight the strategy adopted by MonoDDE in pink.

As reported in the 1st–3rd rows of results in Table 3, when the three depth estimation strategies are applied separately, the model based on *depth from keypoint* achieves the best performance on Moderate and Hard, and the one with only *direct depth estimation* performs the worst. The underlying reason is that *depth from keypoint* brings the most clues (16 depths) for every target while *direct depth estimation* only produces 1 depth.

According to the results in the last 4 rows of Table 3, if we combine two of the depth solving strategies to pro-

duce depths, better results are obtained because the diversity of the estimations is enhanced. The best performance is achieved when we combine all the three strategies, which totally generates 20 depths for every detected object.

5.3. Analysis on Depth Selection and Combination

In this subsection, we analyze how the depth selection and combination strategies affect the results. We compare the performance of the model that tackles estimations in various ways. The results are presented in Table 4. The 1st column indicates how reliable depths are selected. Specifically, "None" means no selection is applied. "Min" indicates that the minimum estimated variance is regarded as the variance of the set S. "Iterative" refers to the proposed iterative strategy described in Algorithm 1. In the 2nd column, "Hard" denotes that we use the value with the minimum variance as the combined depth z_c. "Mean" and "Weighted" represent that z_c is the mean and the weighted sum of the depth estimations, respectively. Notably, the last row (in gray) of Table 4 shows the performance if the best one is always selected from the set of the 20 depths. The strategy employed by MonoDDE is highlighted in pink.

Comparing the 2nd and 3rd rows of the results in Table 4, we can notice that it is necessary to model the variance of the network output and combine estimations with the weighted sum operation in Eq. (15). Besides, according to the 3rd and 5th rows of Table 4, removing outliers with Algorithm 1 boosts the detection precision effectively.

Figure 4. Visualization of some examples on the validation (left) and test (right) sets of KITTI. Failure cases are presented in the last row. The 3D green boxes are produced by MonoDDE and the orange boxes are the ground truth. We hightlight the target failed to be labeled by the annotator with a yellow ellipse and the undetected targets with red ellipses.

Notably, as presented in the last row of Table 4, if we develop a perfect strategy that always selects the most accurate one from the 20 depths, the $AP_{3D}70$ on the Moderate level arrives 38.73%. This phenomenon indicates how to select accurate depths deserves further study in the future work.

5.4. Analysis on the 3D Geometry Confidence

In this subsection, we study how various ways of modeling $P_{3d|2d}$ affect the performance of MonoDDE. We compare the models based on different strategies, and the results are presented in Table 5. In the 1st column of Table 5, "None" means we directly regard the 2D categorical heatmap score as the detection confidence P_m. For "3D IOU", we train a specific network head to regress $P_{3d|2d}$ defined based on 3D IOU. Denoting the 3D IOU between an estimated box and its ground truth as I_{3D}, $P_{3d|2d} = \min\{\max\{2I_{3D} - 0.5, 0\}, 1\}$ following [19]. "d_1-d_{20}" indicates that $P_{3d|2d}$ is computed based on the confidences of the 20 depth estimations through the weighted sum ($\sum_{i=1}^{20} \omega_i d_i$) like Eq. (18). "$d_c$" and "$d_b$" mean we model $P_{3d|2d}$ using the combined depth confidence d_c and the 3D box confidence d_b, respectively. "3D Confidence" is the strategy employed by MonoDDE (marked in pink).

From Table 5, we can mainly observe two facts: (1) Comparing the 1st and 2nd rows of the results, it can be found that modeling $P_{3d|2d}$ with 3D IOU does not always boost the performance. (2) According to the values in the 3rd–6th rows, modeling $P_{3d|2d}$ with our proposed strategy leads to the best results.

Strategies	Val, $AP_{3D}70$ (%)			Val, $AP_{BEV}70$ (%)		
	Easy	Moderate	Hard	Easy	Moderate	Hard
None	23.67	18.15	15.41	31.59	24.57	21.45
3D IOU	22.67	18.54	16.06	30.30	24.14	21.17
d_1-d_{20}	25.32	19.08	16.12	33.37	25.39	22.16
d_c	25.58	19.12	16.17	33.76	25.72	22.34
d_b	26.02	19.48	16.43	34.14	25.87	22.88
3D Confidence	26.66	19.75	16.72	35.51	26.48	23.07

Table 5. Analysis of modeling conditional 3D confidence strategies.

5.5. Qualitative Results and Limitation

We show some 3D boxes and BEV maps produced by MonoDDE on both the KITTI validation and testing sets in Figure 4. As shown, although some targets are not labeled by the annotators, they are still detected by MonoDDE correctly. However, as illustrated in the last row of Figure 4, similar to other works, the performance of MonoDDE on detecting seriously occluded targets is limited.

6. Conclusion

In this paper, we have proposed a robust monocular 3D detector that can produce diverse depth estimations for every target and combine the reliable estimations into a single depth. Besides, a new way for modeling the conditional 3D confidence is developed. The experimental results indicate that all our proposed techniques are effective, which establish new SOTA in monocular 3D object detection. We hope this work can shed light on how to tackle the problem of missing depth information in monocular images. We thank MindSpore [1] for the partial support to this work, which is a new deep learning computing framework.

References

[1] Mindspore. https://www.mindspore.cn. 8

[2] Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE T-ITS*, 20(10):3782–3795, 2019. 1

[3] Tihomir Asparouhov and Bengt Muthén. Weighted least squares estimation with missing data. *Mplus Technical Appendix*, 2010:1–10, 2010. 5

[4] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, pages 9287–9296, 2019. 7

[5] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *ECCV*, pages 135–152, 2020. 6, 7

[6] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *CVPR*, pages 10379–10388, 2021. 2, 5

[7] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, pages 2147–2156, 2016. 2

[8] 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 *NeurIPS*, pages 424–432, 2015. 2

[9] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *CVPR*, pages 12093–12102, 2020. 6, 7

[10] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *ICCV*, pages 502–511, 2019. 5

[11] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *J R Stat Soc Series B Stat Methodol*, 39(1):1–22, 1977. 5

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012. 1, 3, 6

[13] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *J. Field Robot.*, 37(3):362–386, 2020. 1

[14] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 1

[15] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 3

[16] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017. 5

[17] Abhinav Kumar, Garrick Brazil, and Xiaoming Liu. Groomed-nms: Grouped mathematically differentiable nms for monocular 3d object detection. In *CVPR*, pages 8973–8983, 2021. 7

[18] Chengyao Li, Jason Ku, and Steven L Waslander. Confidence guided stereo 3d object detection with split depth estimation. In *IROS*, pages 5776–5783, 2020. 1

[19] Peixuan Li and Huaici Zhao. Monocular 3d detection with geometric constraint embedding and semi-supervised training. *IEEE Robot.Autom. Lett.*, 6(3):5565–5572, 2021. 2, 3, 7, 8

[20] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *ECCV*, pages 644–660, 2020. 2, 3, 5, 7

[21] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Singlestage monocular 3d object detection via keypoint estimation. In *CVPR*, pages 996–997, 2020. 1, 2, 4, 6, 7

[22] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. Autoshape: Real-time shape-aware monocular 3d object detection. In *ICCV*, pages 15641–15650, 2021. 1, 2, 3, 7

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[24] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *ICCV*, pages 3111–3121, 2021. 5

[25] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *ECCV*, pages 311–327, 2020. 1, 2

[26] 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 *ICCV*, pages 6851–6860, 2019. 2

[27] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *CVPR*, pages 4721–4730, 2021. 2, 7

[28] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *CVPR*, pages 2069–2078, 2019. 2

[29] William Menke. Review of the generalized least squares method. *Surv Geophys*, 36(1):1–25, 2015. 3

[30] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, pages 7074–7082, 2017. 6

[31] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *ICCV*, pages 3142–3152, 2021. 2

[32] Friedrich Pukelsheim. The three sigma rule. *Am Stat*, 48(2):88–91, 1994. 5

[33] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. In *CVPR*, pages 5881–5890, 2020. 2

[34] Zhan Qu, Huan Jin, Yang Zhou, Zhen Yang, and Wei Zhang. Focus on local: Detecting lane marker from bottom up via key point. In *CVPR*, pages 14122–14130, 2021.

[35] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, pages 8555–8564, 2021. 2, 7

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28:91–99, 2015. 2

[37] Carlos Ricolfe-Viala and Antonio-José Sánchez-Salmerón. Using the camera pin-hole model restrictions to calibrate the lens distortion model. *Opt. Laser Technol.*, 43(6):996–1005, 2011. 2

[38] Xuepeng Shi, Zhixiang Chen, and Tae-Kyun Kim. Distance-normalized unified representation for monocular 3d object detection. In *ECCV*, pages 91–107, 2020. 2

[39] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3d object detection. *arXiv preprint arXiv:2104.03775*, 2021. 2, 7

[40] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, and Elisa Ricci. Are we missing confidence in pseudo-lidar methods for monocular 3d object detection? In *ICCV*, pages 3225–3233, 2021. 5

[41] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *ICCV*, pages 1991–1999, 2019. 6

[42] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Elisa Ricci, and Peter Kontschieder. Towards generalization across depth for monocular 3d object detection. In *ECCV*, pages 767–782, 2020. 1

[43] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *CoRL*, pages 1475–1485, 2022. 5

[44] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, pages 2403–2412, 2018. 3

[45] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *CVPR*, pages 3289–3298, 2021. 2, 5, 6, 7

[46] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? In *ICLR*, 2021. 5

[47] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Sessd: Self-ensembling single-stage object detector from point cloud. In *CVPR*, pages 14494–14503, 2021. 1

[48] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2, 3, 6

[49] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. Monocular 3d object detection: An extrinsic parameter free approach. In *CVPR*, pages 7556–7566, 2021. 2, 7

[50] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018. 6

[51] Zhikang Zou, Xiaoqing Ye, Liang Du, Xianhui Cheng, Xiao Tan, Li Zhang, Jianfeng Feng, Xiangyang Xue, and Errui Ding. The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3d object detection. In *ICCV*, pages 2713–2722, 2021. 2, 7