Learning Auxiliary Monocular Contexts Helps Monocular 3D Object Detection

Xianpeng Liu[1], **Nan Xue**[2] , **Tianfu Wu**[*] [1]

[1] Department of Electrical and Computer Engineering, North Carolina State University, USA
[2] School of Computer Science, Wuhan University, China
xliu59@ncsu.edu, xuenan@whu.edu.cn, tianfu_wu@ncsu.edu

Abstract

Monocular 3D object detection aims to localize 3D bounding boxes in an input single 2D image. It is a highly challenging problem and remains open, especially when no extra information (e.g., depth, lidar and/or multi-frames) can be leveraged in training and/or inference. This paper proposes a simple yet effective formulation for monocular 3D object detection without exploiting any extra information. It presents the **MonoCon** method which learns **Mono**cular **Con**texts, as auxiliary tasks in training, to help monocular 3D object detection. *The key idea is that with the annotated 3D bounding boxes of objects in an image, there is a rich set of well-posed projected 2D supervision signals available in training, such as the projected corner keypoints and their associated offset vectors with respect to the center of 2D bounding box, which should be exploited as auxiliary tasks in training.* The proposed MonoCon is motivated by the Cramèr–Wold theorem in measure theory at a high level. In implementation, it utilizes a very simple end-to-end design to justify the effectiveness of learning auxiliary monocular contexts, which consists of three components: a Deep Neural Network (DNN) based feature backbone, a number of regression head branches for learning the essential parameters used in the 3D bounding box prediction, and a number of regression head branches for learning auxiliary contexts. After training, the auxiliary context regression branches are discarded for better inference efficiency. In experiments, the proposed MonoCon is tested in the KITTI benchmark (car, pedestrian and cyclist). It outperforms all prior arts in the leaderboard on the car category and obtains comparable performance on pedestrian and cyclist in terms of accuracy. Thanks to the simple design, the proposed MonoCon method obtains the fastest inference speed with 38.7 fps in comparisons. Our code is released at https://git.io/MonoCon.

Introduction

3D object detection is a critical component in many computer vision applications in practice, such as autonomous driving and robot navigation. High performing methods often require more costly system setups such as Lidar sensors (Yan, Mao, and Li 2018; Lang et al. 2019; Qi et al. 2019; Shi et al. 2020) for precise depth measurements or stereo cameras (Li, Chen, and Shen 2019; Qin, Wang, and Lu 2019; Xu et al. 2020; Sun et al. 2020) for stereo depth estimation, and are often more computationally expensive. To alleviate those "burden" and due to the potential prospects of reduced cost and increased modular redundancy, monocular 3D object detection that aims to localize 3D object bounding boxes from an input 2D image has emerged as a promising alternative approach with much attention received in the computer vision and AI community (Chen et al. 2016; Manhardt, Kehl, and Gaidon 2019; Simonelli et al. 2019; Li et al. 2019; Brazil and Liu 2019; Liu et al. 2020; Wang et al. 2020; Ye et al. 2020; Shi, Chen, and Kim 2020; Luo et al. 2021; Kumar, Brazil, and Liu 2021; Wang et al. 2021c,d). In addition to potential advantages in practice, developing powerful monocular 3D object detection systems will facilitate addressing a fundamental question in computer vision: whether it is possible to recover 3D structures from only 2D images which have lost the depth information in the first place.

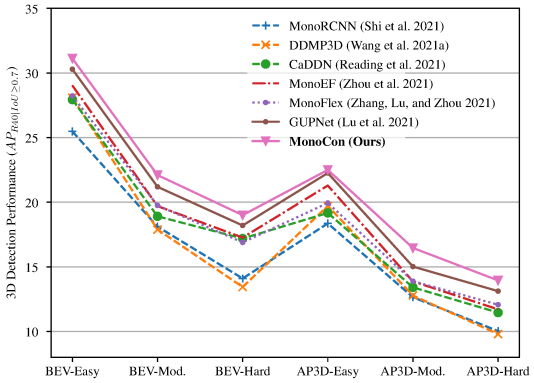

Figure 1: Performance comparisons on the car category in the KITTI 3D object detection benchmark. The proposed MonoCon shows consistently better performance. See text for detail.

This paper is interested in 3D object detection in the autonomous driving application. The objective is to estimate the 3D bounding box for each object instance such as a car in a 2D image. In the KITTI benchmark (Geiger et al. 2013), a 3D bounding box is parameterized by: (i) the 3D center location $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ in the camera 3D coordinates (in meters), (ii) the observation angle α of the object with respect to the camera based on the vector from the camera center to the 3D

[*]T. Wu is the corresponding author.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

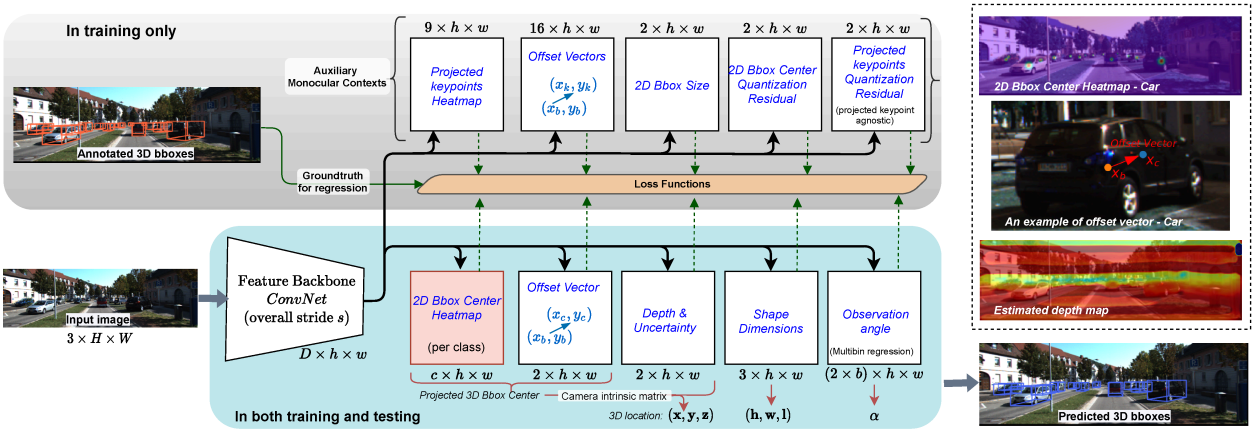

Figure 2: Illustration of the proposed MonoCon method for monocular 3D object detection without exploiting any extra information. It seeks a minimally-simple design. Given an input RGB image of dimensions $3 \times H \times W$, a convolution neural network feature backbone computes the output feature map of dimensions $D \times h \times w$, where D is the output feature map dimension, $h = H/s$ and $w = W/s$ with s the overall stride/sub-sampling rate of the feature backbone (e.g., $s = 4$). Then, light-weight regression head branches are used in a direct and straightforward way, including one set of the regression head branches for the essential parameters (3D locations, shape dimensions and observation angles) which will be used in inferring the 3D bounding box, and the other set for the auxiliary contexts. Only the heatmap of 2D bounding box centers is class specific, and the others are class-agnostic. The proposed MonoCon is trained end-to-end and the auxiliary branches will be discarded in testing. In the right-top, the intermediate results for three regression branches are shown (note that the depth map will only be used sparsely based on the detected 2D bounding box centers). Best viewed in color and magnification. See text for detail.

object center, and (iii) the shape dimensions $(\mathbf{h}, \mathbf{w}, \mathbf{l})$, i.e., height, width and length (in meters). Based on the extensive and insightful analyses made by the MonoDLE method (Ma et al. 2021) in the KITTI benchmark, **one main challenge of improving the overall performance in monocular 3D object detection lies in inferring the 3D center location with high accuracy**. To address the challenge, there are two main types of settings in state-of-the-art monocular 3D object detection, depending on whether there are extra information (Lidar depth measurements, monocular depth estimation results by a separately trained model or multi-frames) leveraged in training and/or inference. In practice, the 3D center location $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is often decomposed to the projected 3D center in the image plane (x_c, y_c) and the object depth $\mathbf{z}$. With the camera intrinsic matrix assumed to be known in both training and inference, the 3D location can be recovered with the inferred projected 3D center and object depth.

This paper focuses on end-to-end monocular 3D object detection without exploiting any extra information. It adopts the anchor-offset formulation proposed in the CenterNet (Zhou, Wang, and Krähenbühl 2019) in learning the projected 3D center based on the 2D bounding box center (i.e., the anchor), and proposes a simple yet effective method that facilitates better overall performance (Fig. 1). **The key idea is to leverage *Mono*cular *Con*texts as auxiliary learning tasks in training to improve the performance** (Fig. 2). The underlying rationale is that with the annotated 3D bounding boxes of objects in an image, there is a rich set of well-posed projected 2D supervision signals available in training, such as the projected corner keypoints and their associated offset vectors with respect to the anchor. They should be exploited in training to induce more expressive representations for monocular 3D object detection. The proposed method is thus dubbed as **MonoCon**.

Statistically speaking, the monocular contexts can be treated as marginal random variables in the image plane, which are projected from the 3D bounding box random variables. In measure theory, the Cramèr-Wold theorem (Cramér and Wold 1936) states that a Borel probability measure on $\mathbb{R}^k$ is uniquely determined by the totality of its one-dimensional projections. Motivated by the Cramèr-Wold theorem, the proposed MonoCon method introduces monocular projections as auxiliary tasks in training to learn more effective representations for monocular 3D object detection. In the meanwhile, it seeks a minimally-simple design of the overall detection system to justify the effectivenss of the underlying rationale and the high-level motivation.

In implementation, the proposed MonoCon utilizes a very simple design consisting of three components (Fig. 2): a Deep Neural Network (DNN) based feature backbone, a number of regression head branches for learning the essential parameters used in the 3D bounding box prediction, and a number of regression head branches for learning auxiliary contexts. After training, the auxiliary context regression branches are discarded. In experiments, the proposed MonoCon is tested in the KITTI benchmark (car, pedestrian and cyclist) (Geiger et al. 2013). It outperforms prior arts (including methods that use lidar, depth or multi-frame extra information) in the leaderboard on the car category and obtains comparable performance on pedestrian and cyclist in terms of accuracy. Thanks to the simple design, the proposed MonoCon obtains the fastest speed with 38.7 fps (on a single NVIDIA 2080Ti GPU card) in comparisons.

Related Work and Our Contributions

Auxiliary tasks and auxiliary learning: In machine learning, auxiliary tasks refer to tasks which are leveraged in training with the sole goal of better performing the primary

tasks of interest in inference. The learning procedure is thus called auxiliary learning, in contrast to multitask learning, for which all tasks in training will be of interest in inference too. Auxiliary tasks and auxiliary learning have shown many successful applications in computer vision (Zhang et al. 2014; Mordan et al. 2018; Liu, Davison, and Johns 2019; Ye et al. 2021; Valada, Radwan, and Burgard 2018). Although simple, exploiting comprehensive 2D auxiliary tasks has not been studied well in monocular 3D object detection. The proposed MonoCon showcases the advantage of auxiliary learning for both performance (on the car category) and inference efficiency in the KITTI benchmark (Geiger et al. 2013).

Monocular 3D detection *with* extra information: Monocular 3D detection falls behind lidar-based and stereo-image based counterparts significantly due to its ill-posed nature. Therefore, many monocular 3D detection methods seek solutions with the help of extra information, such as lidar data (Chen et al. 2021; Reading et al. 2021), off-the-shelf monocular depth estimation modules (pretrained using dense depth map) (Xu and Chen 2018; Ding et al. 2020; Wang et al. 2019; Ma et al. 2020, 2019; Wang et al. 2021a), multi-frames (Brazil et al. 2020), or CAD models (Xiang et al. 2015; Chabot et al. 2017; He and Soatto 2019), etc. Although these methods have shown promising results, however, most of these models heavily rely on extra modules (i.e. depth estimation modules, etc.), which entails extra computation cost. As a result, these methods are usually slow in inference (less than 10 fps), severely hindering their applications in real-time autonomous driving. The proposed MonoCon method does not use any extra information and seeks a minimally-simple design with very promising performance and real-time inference speed achieved. One motivation is that before exploiting the more computationally expensive settings using multi-view images or more costly settings with more sensors, we want to understand the "true" limit of purely monocular 3D detection methods.

Monocular 3D detection *without* extra information: Since the seminal work of Deep3DBox (Mousavian et al. 2017), many efforts (Liu et al. 2019; Li et al. 2020; Cai et al. 2020; Chen et al. 2020; Bao, Yu, and Kong 2020; Shi et al. 2021; Liu, Yixuan, and Liu 2021; Zhang et al. 2021; Lu et al. 2021; Zhang, Lu, and Zhou 2021) have been proposed to utilize 2D-3D geometric constraints to improve 3D detection performance, which are often posed as multi-task learning, rather than auxiliary learning. For example, in the RTM3D method (Li et al. 2020), all of the learned 2D tasks are used as optimization terms to calculate the 3D location of cars in the post-processing (using the PnP method) in inference. In the MonoRCNN method (Shi et al. 2021), one 2D task (i.e., 2D box) is used. The 2D box prediction is used to calculate the depth together with 3D box size in inference. The SMOKE method (Liu, Wu, and Tóth 2020) does not learn any 2D task. One main claim in SMOKE is that 2D tasks will interfere the learning of 3D tasks. In exploiting 2D-3D geometric constraints, existing work compute 3D locations explicitly based on 2D predictions, and thus often suffer from the well-known error amplification effect. So, more recent work try to use uncertainty modeling (e.g., the GUPNet (Lu et al. 2021)), or sophisticated model ensemble (e.g., in MonoFlex (Zhang, Lu, and Zhou 2021)). The goal of the proposed MonoCon is to investigate the effects of 2D auxiliary tasks in training, and to eliminate the potential error amplification effect in inference, with improved performance obtained. More recently, there are efforts exploring how to generate extrinsic-invariant (Zhou et al. 2021) or distance-invariant (Simonelli et al. 2020) representations to improve 3D detection performance. The proposed MonoCon is complementary to aforementioned methods by leveraging well-posed 2D contexts projected from 3D bounding boxes as auxiliary learning tasks. It has the potential to be easily extended using aforementioned methods with performance further improved.

Our contributions: This paper makes three main contributions to monocular 3D object detection as follows: (i) It presents a simple yet surprisingly effective method, MonoCon for purely monocular 3D object detection by learning auxiliary monocular contexts. At a high level, the proposed MonoCon formulation can be explained by the Cramèr-Wold theorem (Cramér and Wold 1936) in measure theory. (ii) It shows state-of-the-art performance on the car category in the KITTI 3D object detection benchmark, outperforming prior arts by a large margin. It obtains comparable performance on the pedestrian and cyclist categories. It can run at a speed of 38.7 fps, faster than prior arts. (iii) It sheds light on developing more powerful and efficient monocular 3D object detection systems by exploring and exploiting even more auxiliary contexts in general applications going beyond autonomous driving (e.g., robot navigation).

Approach

Problem Definition

Let Λ be the image lattice (e.g. 384×1280 in the KITTI benchmark), and I_Λ an image defined on the lattice. As aforementioned, the objective of monocular 3D object detection is to infer the label (e.g., car, pedestrian and cyclist) and the 3D bounding box for each object instance in I_Λ. The 3D bounding box is parameterized by the 3D center location $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ in meters, the shape dimensions $(\mathbf{h}, \mathbf{w}, \mathbf{l})$ in meters and the observation angle $\alpha \in [-\pi, \pi]$, all measured in the camera coordinate system. The observation angle is used in prediction due to its underlying stronger relationship with image appearance. The camera intrinsic matrix is assumed to be known in both training and inference.

Challenges. Typically, both the shape dimensions and the orientation are directly regressed using features computed by a feature backbone such as a Convolutional Neural Network (CNN). The direct regression methods also have shown good performance for them individually. In the meanwhile, the overall 3D bounding box prediction performance (e.g., the Average Precision (AP) based on the intersection-over-union) is relatively less sensitive to the shape dimensions and the orientation, in the sense that if the 3D center location can be inferred with high accuracy, the AP will not decrease dramatically even if the shape dimensions and the orientation are not accurately predicted. By contrast, even with very accurate estimate of shape dimensions and orien-

tations, *the AP will drop catastrophically if the 3D center location is perturbed*. The underlying reason is the significant gap between the shape dimensions (roughly between 1 and 3 meters) and the 3D location (roughly between 1 and 60 meters), and the uncertainty measured for both of them in monocular images will cause dramatically different effects for the overall AP.

There are two different formulations in learning the projected 3D center (x_c, y_c): *One* is to directly predict it by learning a heatmap representation, for which the projected centers falling outside the image plane are either simply discarded in training (Liu, Wu, and Tóth 2020; Ma et al. 2021) or cleverly handled with the help from the intersection point between the image edge and the line from the center of 2D bounding box to the outside projected 3D center (Zhang, Lu, and Zhou 2021). *The other* is to further decompose a projected 3D center into the center of 2D bounding box (x_b, y_b) (i.e., the anchor inside the image plane) and an offset/displacement vector $(\Delta x, \Delta y)$ with $x_c = x_b + \Delta x$ and $y_c = y_b + \Delta y$, following the CenterNet (Zhou, Wang, and Krähenbühl 2019) formulation. Due to the large variation of the offset vectors, it is difficult to learn them. Thus, the latter is often inferior to the former in terms of the overall performance (Zhang, Lu, and Zhou 2021; Ma et al. 2021), albeit it is an intuitively simple and generic representation for learning the projected 3D center. *The proposed method shows that the latter can work well when sufficient monocular contexts are exploited in training.*

The Proposed MonoCon Method

As illustrated in Fig. 2, the proposed MonoCon method is simple by design, consisting of three components:

Feature Backbone. Given an input RGB image I_Λ of dimensions $3 \times H \times W$, a feature backbone $f(\cdot; \Theta)$ is used to compute the output feature map F of dimensions $D \times h \times w$,

$$F = f(I_\Lambda; \Theta), \tag{1}$$

where Θ collects all the learnable parameters, D is the output feature map dimension (e.g., $D = 512$), and h and w are determined by the overall stride/sub-sampling rate s in the backbone (e.g. $s = 4$). We use the DLA network (Yu et al. 2018) (DLA-34) that is widely used in monocular 3D object detection for fair comparisons in experiments.

The 3D Bounding Box Regression Heads. We adopt the anchor-offset formulation in learning the projected 3D bounding box center (x_c, y_c) in the image plane. A regression head is used to compute the class-specific heatmap $\mathcal{H}^b$ of dimensions $c \times h \times w$ for the 2D bounding box center (x_b, y_b) for each of the c classes (e.g., $c = 3$ representing car, pedestrian and cyclist in the KITTI benchmark),

$$\mathcal{H}^b = g(F; \Phi^b), \tag{2}$$

where $g(\cdot; \Phi^b)$ is realized by a light-weight module with the learnable parameters Φ^b,

$$F \xrightarrow[d\times 3\times 3\times D]{Conv+AN+ReLU} \mathbb{F}_{d\times h\times w} \xrightarrow[c\times 1\times 1\times d]{Conv} \mathcal{H}^b_{c\times h\times w}, \tag{3}$$

where the first convolution also reduce the feature dimension to d (e.g., $d = 64$) to be light-weight, and AN represents the Attentive Normalization (AN) (Li, Sun, and Wu 2020), which is a light-weight module integrating feature normalization (e.g., BatchNorm (Ioffe and Szegedy 2015)) and channel-wise feature attention (e.g. the Squeeze-Excitation module (Hu, Shen, and Sun 2018)). Thanks to its mixture modeling formulation of the affine transformation in recalibrating the features after standardization, it is adopted in the regression head for learning more expressive latent feature representations $\mathbb{F}_{d\times h\times w}$. The light-weight module architecture $g(\cdot)$ (Eqn. 3) is used by all regression heads with different instantiations (i.e., different learnable parameters).

The regression head of computing the offset vector $(\Delta x_b^c, \Delta y_b^c)$ from the 2D bounding box center (x_b, y_b) to the projected 3D bounding box center (x_c, y_c) is defined by,

$$\mathcal{O}^c_{2\times h\times w} = g(F; \Theta^{b_c}). \tag{4}$$

Similarly, the depth and the shape dimensions are regressed respectively as follows,

$$\mathcal{Z}_{1\times h\times w} = \frac{1}{\text{Sigmoid}\left(g\left(F; \Theta^Z\right)[0]\right) + \epsilon} - 1, \tag{5}$$

$$\sigma^{\mathcal{Z}}_{1\times h\times w} = g(F; \Theta^Z)[1], \tag{6}$$

$$\mathcal{S}^{3D}_{3\times h\times w} = g(F; \Theta^{S^{3D}}), \tag{7}$$

where $g(F; \Theta^Z)$ estimates the depth and its uncertainty. The inverse sigmoid transformation is applied to handle the unbounded output of $g(F; \Theta^Z)[0]$, as done in (Zhou, Wang, and Krähenbühl 2019), and ϵ is a small positive constant to ensure numeric stability. $\sigma^{\mathcal{Z}}$ is used to model the heteroscedastic aleatoric uncertainty in the depth estimation as done in (Chen et al. 2020; Zhang, Lu, and Zhou 2021; Ma et al. 2021).

For the observation angle α, the multi-bin setting proposed by (Mousavian et al. 2017) is used. The angle range $[-\pi, \pi]$ is divided evenly into a predefined number of b non-overlapping bins (e.g., $b = 12$). The observation angle regression head is defined by,

$$\mathcal{A}_{2b\times h\times w} = g(F; \Theta^A), \tag{8}$$

where the observation angle α is predicted by computing its bin index, $\alpha_i \in \{0, 1, \cdots 11\}$ from the first b channels (using $\arg\max$ after softmax along the b channels) and the corresponding angle residual, α_r in the second b channels of $\mathcal{A}$, together with proper conversions to ensure $\alpha \in [-\pi, \pi]$.

Computing the predicted 3D bounding box. Based on the peaks in each channel of the heatmap $\mathcal{H}^b$ (Eqn. 2) after non-maximum suppression (NMS) and thresholding with a threshold τ (e.g., $\tau = 0.2$), a set of 2D bounding box centers are detected for each class. Without loss of generality, consider a detected 2D bounding box center (x_b, y_b) for a car, the offset vector is retrieved from $\mathcal{O}^c$ (Eqn. 4), $(\Delta x_b, \Delta y_b) = \mathcal{O}^c(x_b, y_b)$. Then, the projected 3D center for the car is predicted by $(x_c, y_c) = (x_b + \Delta x_b, y_b + \Delta y_b)$. The corresponding depth is predicted by $z = \mathcal{Z}(x_b, y_b)$. With the camera intrinsic matrix, the 3D location $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ will be computed in a straightforward way. Similarly, the shape dimensions $(\mathbf{h}, \mathbf{w}, \mathbf{l})$ and the observation angle α can be predicted for the car. With all these parameter inferred, the 3D bounding box is predicted.

The Auxiliary Context Regression Heads. The proposed MonoCon method exploits four types of projection information from 3D bounding boxes as auxiliary learning tasks.

i) The heatmaps of the projected keypoints. As done in computing the 2D bounding box center heatmap $\mathcal{H}^b$ (Eqn. 2), the first type of auxiliary contexts is the heatmaps of the 9 projected keypoints consisting of the projected 8 corner points and the projected center of the 3D bounding box, and we have,

$$\mathcal{H}^k_{9\times h\times w} = g(F;\Theta^k). \tag{9}$$

ii) The offset vectors for the 8 projected corner points. In addition to the offset vector from the 2D bounding box center to the projected 3D bounding box center, $\mathcal{O}^c$ (Eqn. 4), the second type of auxiliary contexts is the offset vectors from the 2D bounding box center to the 8 projected corner points of the 3D bounding box, and we have,

$$\mathcal{O}^k_{16\times h\times w} = g(F;\Theta^{b_k}). \tag{10}$$

Note that this is combined with Eqn. 4 with the first convolution block in $g(\cdot)$ shared in implementation.

iii) The 2D bounding box size. This is as done in the CenterNet (Zhou, Wang, and Krähenbühl 2019). The height and width of the 2D bounding box are regressed,

$$\mathcal{S}^{2D}_{2\times h\times w} = g(F;\Theta^{S^{2D}}). \tag{11}$$

iv) The quantization residual of a keypoint location. Due to the overall stride s (typically $s > 1$) in the feature backbone, there is a residual between the pixel location in the original input image I_Λ and its corresponding pixel location in the output feature map F after multiplying the stride s. Consider the 2D bounding box center (x^*_b, y^*_b) of a car in the original image, its pixel location in the feature map F is $(x_b = \lfloor \frac{x^*_b}{s} \rfloor, y_b = \lfloor \frac{y^*_b}{s} \rfloor)$, and the residual is defined by,

$$\delta x_b = x^*_b - x_b, \quad \delta y_b = y^*_b - y_b. \tag{12}$$

We model the residual of the 2D bounding box center (x_b, y_b) and that of the 9 projected keypoints (x_k, y_k) separately to account for the underlying difference of the nature of those points. The latter is modeled in a keypoint-agnostic way as shown in Fig. 2 for simplicity. We have,

$$\mathcal{R}^b_{2\times h\times w} = g(F;\Theta^{R^b}), \tag{13}$$

$$\mathcal{R}^k_{2\times h\times w} = g(F;\Theta^{R^k}). \tag{14}$$

Loss Functions

We use five loss functions which are widely used in monocular 3D object detection, consisting of (i) *the Gaussian kernel weighted focal loss* (Lin et al. 2017; Law and Deng 2018) function for the heatmaps (Eqn. 2 and Eqn. 9) as used in the CenterNet (Zhou, Wang, and Krähenbühl 2019), (ii) *the Laplacian aleatoric uncertainty loss function* for the depth estimation (Eqn. 5 and Eqn. 6), (iii) *the dimension-aware $L1$ loss function* for shape dimensions (Eqn. 7), (iv) *the standard cross-entropy loss function* for the bin index in observation angles (Eqn. 8), and (v) *the standard $L1$ loss function* for offset vectors (Eqn. 4 and Eqn. 10), the intra-bin angle residual in observation angles (Eqn. 8), 2D bounding box sizes (Eqn. 11) and the quantization residual (Eqn. 13 and Eqn. 14). We briefly discuss the first three as follows.

i) The Gaussian kernel weighted focal loss function for heatmaps (Lin et al. 2017; Law and Deng 2018; Zhou, Wang, and Krähenbühl 2019). Without loss of generality, consider a regressed heatmap $\mathcal{H}_{1\times h\times w}$ (e.g., the 2D bounding box centers of cars), the ground-truth heatmap $\mathcal{H}^*_{1\times h\times w}$ is also generated at the resolution of the regressed heatmap. For each ground-truth center point $(x^*_b, y^*_b) \in \mathbb{P}$ in the original image, its location in the ground-truth heatmap is $(x_b = \lfloor \frac{x^*_b}{s} \rfloor, y_b = \lfloor \frac{y^*_b}{s} \rfloor)$ (where s is the overall stride of the feature backbone). A Gaussian kernel $G(x,y) = \exp\left(-\frac{(x-x_b)^2+(y-y_b)^2}{2\cdot\sigma_b^2}\right)$ is used to model the center point, where σ_b is a predefined object-size-adaptive standard deviation as used in (Law and Deng 2018). If two Gaussian kernels overlap, the element-wise maximum is kept. All the $G(\cdot,\cdot)$'s are then collapsed to form the ground-truth heatmap $\mathcal{H}^*$. The loss function is defined by,

$$\mathcal{L}(\mathcal{H},\mathcal{H}^*) = \frac{-1}{N}\sum_{(x,y)} \begin{cases} (1-\mathcal{H}_{xy})^\gamma \log(\mathcal{H}_{xy}), & \text{if } \mathcal{H}^*_{xy} = 1, \\ (1-\mathcal{H}^*_{xy})^\beta (\mathcal{H}_{xy})^\gamma \log(1-\mathcal{H}_{xy}), \end{cases} \tag{15}$$

where $N = |\mathbb{P}|$ is the number of ground-truth points. β and γ are hyper-parameters (e.g., $\beta = 4.0$ and $\gamma = 2.0$).

ii) The Laplacian aleatoric uncertainty loss function for depth (Chen et al. 2020; Ma et al. 2021; Zhang, Lu, and Zhou 2021). Denote by $\mathcal{Z}^*_{1\times h\times w}$ the ground-truth (sparse) depth map in which the ground-truth depth of an annotated 3D bounding box is assigned to the corresponding ground-truth 2D bounding box center location in the lattice of $h\times w$, i.e., $\mathcal{Z}^*(x_b, y_b)$ (with the same inverse sigmoid transformation applied as in Eqn. 5). The Laplace distribution is used in modeling the uncertainty σ^Z (Eqn. 6). For the prediction depth $\mathcal{Z}$ (Eqn. 5), the loss function is defined by,

$$\mathcal{L}(\mathcal{Z},\mathcal{Z}^*) = \frac{1}{|\mathbb{P}|}\sum_{(x_b,y_b)\in\mathbb{P}} \frac{\sqrt{2}}{\sigma^Z_b}|z_b - z^*_b| + \log(\sigma^Z_b), \tag{16}$$

where $\mathbb{P}$ is the set of ground-truth 2D bounding box center points, $\sigma^Z_b = \sigma^Z(x_b, y_b)$, $z_b = \mathcal{Z}(x_b, y_b)$ and $z^*_b = \mathcal{Z}^*(x_b, y_b)$.

iii) The dimension-aware $L1$ loss function for shape dimensions (Ma et al. 2021), which is motivated by the IoU oriented optimization (Rezatofighi et al. 2019) and realizes a re-distribution of the standard $L1$ loss. Similarly, let $\mathcal{S}^{3D^*}$ be the ground-truth map of shape dimensions assigned to the ground-truth 2D bounding box center locations in the lattice of $h\times w$. For the predicted shape dimensions $\mathcal{S}^{3D}$ (Eqn. 7), the loss function is defined by,

$$\mathcal{L}(\mathcal{S}^{3D},\mathcal{S}^{3D^*}) = \lambda\cdot||\frac{\mathcal{S}^{3D}-\mathcal{S}^{3D^*}}{\mathcal{S}^{3D}}||_1, \tag{17}$$

where λ is the compensation weight to ensure the dimension-aware $L1$ loss has the same value as the standard $L1$ loss, which is by definition the ratio (without gradients in training) between the standard $L1$ loss and the dimension-aware loss before applying the compensation weight.

The Overall Loss is simply the sum of all loss terms each of which has a trade-off weight parameter. For simplicity, we use 1.0 for all loss terms except for the 2D size $L1$ loss which uses 0.1.

Experiments

In this section, we test the proposed MonoCon in the widely used and challenging KITTTI 3D object detection bench-

Methods, *Publication Venues*	Extra Info.	Runtime↓ (ms)	$AP_{BEV\|R40\|IoU\geq0.7}$ ↑ Easy	Mod.	Hard	$AP_{3D\|R40\|IoU\geq0.7}$ ↑ Easy	Mod.	Hard
PatchNet, *ECCV20* (Ma et al. 2020)	Depth	400	22.97	16.86	14.97	15.68	11.12	10.17
D4LCN, *CVPR20* (Ding et al. 2020)		200	22.51	16.02	12.55	16.65	11.72	9.51
DDMP-3D, *CVPR21* (Wang et al. 2021a)		180	28.08	17.89	13.44	19.71	12.78	9.80
Kinematic3D, *ECCV20* (Brazil et al. 2020)	Multi-frames	120	26.69	17.52	13.10	19.07	12.72	9.17
MonoRUn, *CVPR21* (Chen et al. 2021)	Lidar	70	27.94	17.34	15.24	19.65	12.30	10.58
CaDDN, *CVPR21* (Reading et al. 2021)		630	27.94	18.91	17.19	19.17	13.41	11.46
RTM3D, *ECCV20* (Li et al. 2020)	None	40	19.17	14.20	11.99	14.41	10.34	8.77
Movi3D, *ECCV20* (Simonelli et al. 2020)		45	22.76	17.03	14.85	15.19	10.90	9.26
IAFA, *ACCV20* (Zhou et al. 2020)		40	25.88	17.88	15.35	17.81	12.01	10.61
MonoDLE, *CVPR21* (Ma et al. 2021)		40	24.79	18.89	16.00	17.23	12.26	10.29
MonoRCNN, *ICCV21* (Shi et al. 2021)		70	25.48	18.11	14.10	18.36	12.65	10.03
Ground-Aware, *RAL21* (Liu, Yixuan, and Liu 2021)		50	29.81	17.98	13.08	21.65	13.25	9.91
PCT, - (Wang et al. 2021b)		45	29.65	19.03	15.92	21.00	13.37	11.31
MonoGeo, - (Zhang et al. 2021)		50	25.86	18.99	16.19	18.85	13.81	11.52
MonoEF, *CVPR21* (Zhou et al. 2021)		30	29.03	19.70	17.26	21.29	13.87	11.71
MonoFlex, *CVPR21* (Zhang, Lu, and Zhou 2021)		35	28.23	19.75	16.89	19.94	13.89	12.07
GUPNet, *ICCV21* (Lu et al. 2021)		34	*30.29*	*21.19*	*18.20*	*22.26*	*15.02*	*13.12*
MonoCon (Ours), *AAAI22*	None	**25.8**	**31.12**	**22.10**	**19.00**	**22.50**	**16.46**	**13.95**
	Improvement	v.s. Depth	+3.04	+4.21	+4.03	+2.79	+3.68	+3.78
		v.s. Multi-frames	+4.43	+4.58	+5.90	+3.43	+3.74	+4.78
		v.s. LiDAR	+3.18	+3.19	+1.81	+2.85	+3.05	+2.49
		v.s. None	+0.83	+0.91	+0.80	+0.24	+1.44	+0.83

Table 1: Comparisons with state-of-the-art methods on the car category in the KITTI official test set. Following the KITTI protocol, methods are ranked by their performance under the moderate difficulty setting. The best results are listed in bold and the second place in italic. The runtime of our MonoCon is measured using a single 2080Ti GPU card.

Methods	Extra	*Ped.*, $AP_{3D\|R40\|IoU\geq0.5}$ Easy	Mod.	Hard	*Cyc.*, $AP_{3D\|R40\|IoU\geq0.5}$ Easy	Mod.	Hard
DDMP-3D	Depth	4.93	2.55	3.01	4.18	2.50	2.32
CaDDN	Lidar	12.87	8.14	6.76	**7.00**	**3.41**	**3.30**
MonoDLE	None	9.64	6.55	5.44	4.59	2.66	2.45
MonoGeo		8.00	5.63	4.71	4.73	2.93	2.58
MonoEF		4.27	2.79	2.21	1.80	0.92	0.71
MonoFlex		9.43	6.31	5.26	4.17	2.35	2.04
GUPNet		**14.95**	**9.76**	**8.41**	*5.58*	*3.21*	*2.66*
MonoCon		*13.10*	*8.41*	*6.94*	2.80	1.92	1.55

Table 2: Comparisons with state-of-the-art methods on the pedestrian category and the cyclist category in the KITTI official test set.

mark (Geiger et al. 2013). We first present comparisons with prior arts in the leaderboard, and then analyze the proposed MonoCon method using ablation studies.

Data. The KITTI dataset consists of 7,481 images for training and 7,518 images for testing. There are three categories of interest: car, pedestrian and cyclist. The ground truth for the test set is reserved for evaluation on the test server. In comparison with prior arts, we train our MonoCon on all 7,481 images and the performance is evaluated by the KITTI official server. For ablation studies, we follow the protocol used by prior works (Chen et al. 2016, 2015, 2017) to split the provided whole training data into a training subset (3,712 images) and a validation subset (3,769 images).

Evaluation Metrics. We follow the protocol provided in the KITTI benchmark. The detection is evaluated by the average precision (AP) of 3D bounding boxes $AP_{3D|R40}$ and the AP of bird's eye view ($AP_{BEV|R40}$), both with 40 recall positions ($R40$) used and under three difficulty settings (easy, moderate, and hard). *The moderate difficulty level is used to rank methods in the KITTI leaderboard.* The APs are computed with the intersection-over-union (IoU) threshold 0.7, 0.5 and 0.5 for car, pedestrian and cyclist respectively.

Implementation Details. Our MonoCon is trained on a single GPU with a batch size of 8 in an end-to-end way for 200 epochs. The AdamW optimizer is used with $(\beta_1, \beta_2) = (0.95, 0.99)$ and weight decay 0.00001 (not applying to feature normalization layers and bias parameters). The initial learning rate is $2.25e-4$, and the cyclic learning rate scheduler is used (1 cycle), which first gradually increases the learning rate to $2.25e-3$ with the step ratio 0.4, and then gradually drops to $2.25e-4 \times 1.0e-4$ (i.e., the target ratio is $(10, 1.0e-4)$). The cyclic scheduler is also applied for the momentum with the target ratio $(0.85/0.95, 1)$ and the same step ratio 0.4. Due to the auxiliary context regression heads in training, the complexity of training our MonoCon is greater in terms of training time and memory footprint. After training, the auxiliary components will be discarded, resulting in faster speed in inference than prior arts. We adopt the commonly used data augmentation methods such as photometric distortion and random horizontal flipping following (Zhou, Wang, and Krähenbühl 2019; Ma et al. 2021; Zhang, Lu, and Zhou 2021). We also utilize random shifting to augment cropped instances at the edge of images.

Comparisons with State-of-the-Art Methods

Table 1 and Table 2 show the quantitative comparisons of our MonoCon with state-of-the-art methods. *We provide qualitative results in the supplementary materials.*

Comparisons on the car category. Cars are the dominant objects in the KITTI 3D object detection benchmark, and of the most interest in evaluation. Our MonoCon consistently outperforms all prior arts. In terms of the KITTI ranking pro-

	2D Context Heads					AN	Val, AP_{R40}, Car		
	$\mathcal{O}^k$	$\mathcal{R}^b$	$\mathcal{S}^{2D}$	$\mathcal{H}^k$	$\mathcal{R}^k$		Easy	Mod.	Hard
MonoDLE	-	-	✓	-	-	-	17.45	13.66	11.68
MonoDLE*	-	-	✓	-	-	✓	22.96	16.76	14.85
(a)	-	-	-	-	-	✓	16.57	10.20	8.14
(b)	-	✓	✓	✓	✓	✓	17.65	11.42	8.71
(c)	✓	✓	✓	-	-	✓	21.76	16.09	13.34
(d)	✓	✓	-	✓	✓	✓	22.72	16.68	13.88
(e)	✓	✓	✓	✓	-	✓	23.51	17.76	15.03
(f)	✓	-	✓	✓	✓	✓	24.11	18.28	15.35
(g)	✓	✓	✓	✓	✓	-	25.37	18.69	15.67
MonoCon	✓	✓	✓	✓	✓	✓	**26.33**	**19.01**	**15.98**

Table 3: Ablation studies on different auxiliary contexts (see Fig. 2 and Eqn. 9 to Eqn. 14) and the Attentive Normalization (Li, Sun, and Wu 2020) (AN in Eqn. 3) in our MonoCon. *For fair comparisons and to justify our method's effectiveness, we re-implement and train a modified and enhanced version of the vanilla MonoDLE (Ma et al. 2021) with the AN added and the exactly same training settings as our MonoCon. Note that the 2D size is used in both training and inference in MononDLE.

tocol based on the $AP_{3D|R40}$ under the moderate difficulty setting, our MonoCon achieves significant improvement by 1.44% **absolute increase** against the runner-up method, the GUPNet (Lu et al. 2021). It also runs faster than prior arts. The improvement justify the effectiveness of learning more auxiliary monocular contexts in monocular 3D object detection for the autonomous driving applications. *Our MonoCon also consistently outperform prior arts in the validation set with the results provided in the supplementary materials* due to the space limit.

Comparisons on the pedestrian and cyclist categories. Our MonoCon shows inferior performance than some of the prior arts. On the pedestrian category, our MonoCon shows 1.35% drop of $AP_{3D|R40}$ under the moderate setting comparing with the best model, the GUPNet (Lu et al. 2021), but outperforms all other methods in comparisons. On the cyclist category, our MonoCon shows 1.29% drop comparing with the best purely monocular model, the MonoDLE method (Ma et al. 2021). Overall, our MonoCon is less effective on the cyclist category among the three categories. We observe that the 3D bounding boxes of pedestrian and cyclist are much smaller than those of car, and the projected monocular contexts are often in the very close proximity in the feature map (of the $h \times w$ lattice). The close proximity may affect the learnability and effectiveness of the auxiliary contexts. One potential solution is to randomly sample a subset of auxiliary contexts that are spatially separate from each other, which we leave to future work.

Ablation Studies

We investigate the effects of learning auxiliary contexts and of the class-agnostic settings for regression heads in Fig. 2.

The Importance of Learning Auxiliary Contexts and the Attentive Normalization (AN). Table 3 shows the comparisons which show the effectiveness of the proposed MonoCon and justify the importance of the design choices. One the one hand, without the auxiliary components, our MonoCon is most similar to the MonoDLE method (Ma et al. 2021). We retrain an enhanced MonoDLE model which obtains significantly better performance than the vanilla MonoDLE. Our MonoCon still outperforms the enhanced MonoDLE by a large margin. On the other hand, we test 7 variants of our MonoCon model from (a) to (g). The auxiliary contexts are significantly more important than the Attentive Normalization: (g) vs (a) with a 8.49% absolution increase under the moderate difficulty settings, which clearly shows the significance of the proposed formulation against some implementation tuning. From (b) to (f), we rank the importance of the auxiliary contexts based on the performance of the model trained without them: the lower the performance is, the more important the context(s) are.

Data	CA	Val, AP_{R40}, Car			Val, AP_{R40}, Ped.			Val, AP_{R40}, Cyc.		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
All	✓	**26.33**	**19.01**	**15.98**	1.46	1.31	0.99	**7.60**	**4.35**	**3.55**
All	-	24.69	18.53	15.49	**9.21**	**6.85**	**5.49**	3.44	1.50	1.50
Car	N/A	24.60	18.15	15.36	-	-	-	-	-	-
Ped.	N/A	-	-	-	5.10	4.13	3.10	-	-	-
Cyc.	N/A	-	-	-	-	-	-	2.98	1.66	1.27

Table 4: Ablation studies on the design of regression heads (class-agnostic (CA) vs class-specific in Eqn. 4 to Eqn. 14) and the training settings (joint vs separate training of car, pedestrian and cyclist).

The effects of class-agnostic settings in regression heads and of training settings. Table 4 shows the comparisons. On the one hand, using the class-agnostic design shows better performance for the car and cyclist categories, while the class-specific design is significantly better for the pedestrian category. On the other hand, jointly training the three categories is beneficial, which indicates that some inter-category synergy may exist.

Conclusion

This paper proposes a simple yet effective formulation for monocular 3D object detection without exploiting any extra information. It presents the MonoCon method which learns auxiliary monocular contexts that are projected from the 3D bounding boxes in training. The proposed MonoCon utilizes a simple design in implementation consisting of a ConvNet feature backbone and a list of regression heads with the same module architecture for the essential parameters and the auxiliary contexts. In experiments, the proposed MonoCon is tested in the KITTI 3D object detection benchmark with state-of-the-art performance on the car category and comparable performance on the pedestrian and cyclist categories. At a high level, the effectiveness of the proposed MonoCon can be explained by the Cramèr–Wold theorem in measure theory. Ablation studies are performed to investigate, and the results support, the effectiveness of our MonoCon.

Acknowledgements

X. Liu and T. Wu were supported in part by NSF IIS-1909644, ARO Grants W911NF1810295 and W911NF2210010, NSF IIS-1822477, NSF CMMI-2024688, NSF IUSE-2013451 and DHHS-ACL Grant 90IFDV0017-01-00. N. Xue was supported by NSFC under Grant 62101390. The views presented in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Bao, W.; Yu, Q.; and Kong, Y. 2020. Object-Aware Centroid Voting for Monocular 3D Object Detection. In *IROS*, 2197–2204. IEEE.

Brazil, G.; and Liu, X. 2019. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, 9287–9296.

Brazil, G.; Pons-Moll, G.; Liu, X.; and Schiele, B. 2020. Kinematic 3d object detection in monocular video. In *ECCV*, 135–152. Springer.

Cai, Y.; Li, B.; Jiao, Z.; Li, H.; Zeng, X.; and Wang, X. 2020. Monocular 3D object detection with decoupled structured polygon estimation and height-guided depth estimation. In *AAAI*, volume 34, 10478–10485.

Chabot, F.; Chaouch, M.; Rabarisoa, J.; Teuliere, C.; and Chateau, T. 2017. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *CVPR*, 2040–2049.

Chen, H.; Huang, Y.; Tian, W.; Gao, Z.; and Xiong, L. 2021. MonoRUn: Monocular 3D Object Detection by Reconstruction and Uncertainty Propagation. In *CVPR*, 10379–10388.

Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; and Urtasun, R. 2016. Monocular 3d object detection for autonomous driving. In *CVPR*, 2147–2156.

Chen, X.; Kundu, K.; Zhu, Y.; Berneshawi, A. G.; Ma, H.; Fidler, S.; and Urtasun, R. 2015. 3d object proposals for accurate object class detection. In *NeurIPS*, 424–432. Citeseer.

Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 1907–1915.

Chen, Y.; Tai, L.; Sun, K.; and Li, M. 2020. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *CVPR*, 12093–12102.

Cramér, H.; and Wold, H. 1936. Some theorems on distribution functions. *Journal of the London Mathematical Society*, 1(4): 290–294.

Ding, M.; Huo, Y.; Yi, H.; Wang, Z.; Shi, J.; Lu, Z.; and Luo, P. 2020. Learning depth-guided convolutions for monocular 3d object detection. In *CVPR Workshops*, 1000–1001.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.

He, T.; and Soatto, S. 2019. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In *AAAI*, volume 33, 8409–8416.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *CVPR*, 7132–7141.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 448–456. PMLR.

Kumar, A.; Brazil, G.; and Liu, X. 2021. GrooMeD-NMS: Grouped Mathematically Differentiable NMS for Monocular 3D Object Detection. In *CVPR*, 8973–8983.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 12697–12705.

Law, H.; and Deng, J. 2018. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 734–750.

Li, B.; Ouyang, W.; Sheng, L.; Zeng, X.; and Wang, X. 2019. Gs3d: An efficient 3d object detection framework for autonomous driving. In *CVPR*, 1019–1028.

Li, P.; Chen, X.; and Shen, S. 2019. Stereo r-cnn based 3d object detection for autonomous driving. In *CVPR*, 7644–7652.

Li, P.; Zhao, H.; Liu, P.; and Cao, F. 2020. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *ECCV*, 644–660. Springer.

Li, X.; Sun, W.; and Wu, T. 2020. Attentive normalization. In *ECCV*, 70–87. Springer.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *ICCV*, 2980–2988.

Liu, L.; Lu, J.; Xu, C.; Tian, Q.; and Zhou, J. 2019. Deep fitting degree scoring network for monocular 3d object detection. In *CVPR*, 1057–1066.

Liu, L.; Wu, C.; Lu, J.; Xie, L.; Zhou, J.; and Tian, Q. 2020. Reinforced axial refinement network for monocular 3d object detection. In *ECCV*, 540–556. Springer.

Liu, S.; Davison, A. J.; and Johns, E. 2019. Self-supervised generalisation with meta auxiliary learning. *arXiv:1901.08933*.

Liu, Y.; Yixuan, Y.; and Liu, M. 2021. Ground-aware monocular 3d object detection for autonomous driving. *IEEE RA-L*, 6(2): 919–926.

Liu, Z.; Wu, Z.; and Tóth, R. 2020. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *CVPR Workshops*, 996–997.

Lu, Y.; Ma, X.; Yang, L.; Zhang, T.; Liu, Y.; Chu, Q.; Yan, J.; and Ouyang, W. 2021. Geometry Uncertainty Projection Network for Monocular 3D Object Detection. *arXiv:2107.13774*.

Luo, S.; Dai, H.; Shao, L.; and Ding, Y. 2021. M3DSSD: Monocular 3D single stage object detector. In *CVPR*, 6145–6154.

Ma, X.; Liu, S.; Xia, Z.; Zhang, H.; Zeng, X.; and Ouyang, W. 2020. Rethinking pseudo-lidar representation. In *ECCV*, 311–327. Springer.

Ma, X.; Wang, Z.; Li, H.; Zhang, P.; Ouyang, W.; and Fan, X. 2019. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *ICCV*, 6851–6860.

Ma, X.; Zhang, Y.; Xu, D.; Zhou, D.; Yi, S.; Li, H.; and Ouyang, W. 2021. Delving into Localization Errors for Monocular 3D Object Detection. In *CVPR*, 4721–4730.

Manhardt, F.; Kehl, W.; and Gaidon, A. 2019. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *CVPR*, 2069–2078.

Mordan, T.; Thome, N.; Henaff, G.; and Cord, M. 2018. Revisiting multi-task learning with rock: a deep residual auxiliary block for visual detection. In *NeurIPS*.

Mousavian, A.; Anguelov, D.; Flynn, J.; and Kosecka, J. 2017. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 7074–7082.

Qi, C. R.; Litany, O.; He, K.; and Guibas, L. J. 2019. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 9277–9286.

Qin, Z.; Wang, J.; and Lu, Y. 2019. Triangulation learning network: from monocular to stereo 3d object detection. In *CVPR*, 7615–7623.

Reading, C.; Harakeh, A.; Chae, J.; and Waslander, S. L. 2021. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 8555–8564.

Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; and Savarese, S. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 658–666.

Shi, S.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *TPAMI*.

Shi, X.; Chen, Z.; and Kim, T.-K. 2020. Distance-normalized unified representation for monocular 3d object detection. In *ECCV*, 91–107. Springer.

Shi, X.; Ye, Q.; Chen, X.; Chen, C.; Chen, Z.; and Kim, T.-K. 2021. Geometry-based Distance Decomposition for Monocular 3D Object Detection. *arXiv:2104.03775*.

Simonelli, A.; Bulo, S. R.; Porzi, L.; López-Antequera, M.; and Kontschieder, P. 2019. Disentangling monocular 3d object detection. In *ICCV*, 1991–1999.

Simonelli, A.; Bulo, S. R.; Porzi, L.; Ricci, E.; and Kontschieder, P. 2020. Towards generalization across depth for monocular 3d object detection. In *ECCV*, 767–782. Springer.

Sun, J.; Chen, L.; Xie, Y.; Zhang, S.; Jiang, Q.; Zhou, X.; and Bao, H. 2020. Disp r-cnn: Stereo 3d object detection via shape prior guided instance disparity estimation. In *CVPR*, 10548–10557.

Valada, A.; Radwan, N.; and Burgard, W. 2018. Deep auxiliary learning for visual localization and odometry. In *ICRA*, 6939–6946. IEEE.

Wang, L.; Du, L.; Ye, X.; Fu, Y.; Guo, G.; Xue, X.; Feng, J.; and Zhang, L. 2021a. Depth-conditioned Dynamic Message Propagation for Monocular 3D Object Detection. In *CVPR*, 454–463.

Wang, L.; Zhang, L.; Zhu, Y.; Zhang, Z.; He, T.; Li, M.; and Xue, X. 2021b. Progressive Coordinate Transforms for Monocular 3D Object Detection. arXiv:2108.05793.

Wang, T.; Zhu, X.; Pang, J.; and Lin, D. 2021c. FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection. arXiv:2104.10956.

Wang, T.; Zhu, X.; Pang, J.; and Lin, D. 2021d. Probabilistic and Geometric Depth: Detecting Objects in Perspective. arXiv:2107.14160.

Wang, X.; Yin, W.; Kong, T.; Jiang, Y.; Li, L.; and Shen, C. 2020. Task-aware monocular depth estimation for 3d object detection. In *AAAI*, volume 34, 12257–12264.

Wang, Y.; Chao, W.-L.; Garg, D.; Hariharan, B.; Campbell, M.; and Weinberger, K. Q. 2019. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 8445–8453.

Xiang, Y.; Choi, W.; Lin, Y.; and Savarese, S. 2015. Data-driven 3d voxel patterns for object category recognition. In *CVPR*, 1903–1911.

Xu, B.; and Chen, Z. 2018. Multi-level fusion based 3d object detection from monocular images. In *CVPR*, 2345–2353.

Xu, Z.; Zhang, W.; Ye, X.; Tan, X.; Yang, W.; Wen, S.; Ding, E.; Meng, A.; and Huang, L. 2020. Zoomnet: Part-aware adaptive zooming neural network for 3d object detection. In *AAAI*, volume 34, 12557–12564.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

Ye, J.; Batra, D.; Das, A.; and Wijmans, E. 2021. Auxiliary Tasks and Exploration Enable ObjectNav. *arXiv:2104.04112*.

Ye, X.; Du, L.; Shi, Y.; Li, Y.; Tan, X.; Feng, J.; Ding, E.; and Wen, S. 2020. Monocular 3d object detection via feature domain adaptation. In *ECCV*, 17–34. Springer.

Yu, F.; Wang, D.; Shelhamer, E.; and Darrell, T. 2018. Deep layer aggregation. In *CVPR*, 2403–2412.

Zhang, Y.; Lu, J.; and Zhou, J. 2021. Objects are Different: Flexible Monocular 3D Object Detection. In *CVPR*, 3289–3298.

Zhang, Y.; Ma, X.; Yi, S.; Hou, J.; Wang, Z.; Ouyang, W.; and Xu, D. 2021. Learning Geometry-Guided Depth via Projective Modeling for Monocular 3D Object Detection. *arXiv:2107.13931*.

Zhang, Z.; Luo, P.; Loy, C. C.; and Tang, X. 2014. Facial landmark detection by deep multi-task learning. In *ECCV*, 94–108. Springer.

Zhou, D.; Song, X.; Dai, Y.; Yin, J.; Lu, F.; Liao, M.; Fang, J.; and Zhang, L. 2020. IAFA: Instance-aware Feature Aggregation for 3D Object Detection from a Single Image. In *ACCV*.

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as points. *arXiv:1904.07850*.

Zhou, Y.; He, Y.; Zhu, H.; Wang, C.; Li, H.; and Jiang, Q. 2021. Monocular 3D Object Detection: An Extrinsic Parameter Free Approach. In *CVPR*, 7556–7566.