Exploiting Ground Depth Estimation for Mobile Monocular 3D Object Detection

Yunsong Zhou, *Member, IEEE*, Quan Liu, *Member, IEEE*, Hongzi Zhu, *Senior Member, IEEE*, Yunzhe Li, *Member, IEEE*, Shan Chang, *Member, IEEE*, and Minyi Guo, *Fellow, IEEE*

***Abstract*—Detecting 3D objects from a monocular camera in mobile applications, such as on a vehicle, drone, or robot, is a crucial but challenging task. The monocular vision's *near-far disparity* and the camera's constantly changing position make it difficult to achieve high accuracy, especially for distant objects. In this paper, we propose a new Mono3D framework named *MoGDE*, which takes inspiration from the observation that an object's depth can be inferred from the ground's depth underneath it. MoGDE estimates the corresponding ground depth of an image and utilizes this information to guide Mono3D. We use a pose detection network to estimate the camera's orientation and construct a feature map that represents pixel-level ground depth based on the 3D-to-2D perspective geometry. To further improve Mono3D with the estimated ground depth, we design an RGB-D feature fusion network based on transformer architecture. The long-range self-attention mechanism is utilized to identify ground-contacting points and pin the corresponding ground depth to the image feature map. We evaluate MoGDE on the KITTI dataset, and the results show that it significantly improves the accuracy and robustness of Mono3D for both near and far objects. MoGDE outperforms state-of-the-art methods and ranks first among the pure image-based methods on the KITTI 3D benchmark.**

***Index Terms*—Monocular 3D object detection, ground depth estimation, vision transformer, autonomous driving.**

I. Introduction

AFTER making promising progress in 2D object detection in recent years, as evidenced by the success of techniques such as Faster R-CNN [49] and YOLOv3 [48], there has been growing interest from both industry and academia in 3D object detection, particularly in the context of moving agents. This is because 3D object detection plays a crucial role in a wide range of applications, including autonomous vehicles [18], drones, robotic manipulation, and augmented reality. While LiDAR-based [12], [28], [44], [50], [51], [72] and stereo-based [10], [11], [24], [43], [46], [63] methods have been extensively researched, the field of monocular 3D object detection (Mono3D) remains an open and challenging research area that offers a much cheaper, more energy-efficient, and easier-to-deploy alternative. To be practical for moving agents, a Mono3D detector must satisfy two key requirements: first, it must accurately produce 3D bounding boxes for both near and far objects, which is crucial for high-priority driving safety applications; and second, it must remain robust in mobile scenarios where camera pose changes along with the movement of the agent.

Recent Mono3D methods with complex network structures [39], [45], [55], [61], [73] have achieved high accuracy for near objects in the literature. However, the predicted 3D bounding boxes for far objects are still ill-posed due to the lack of depth cues. This disparity between near and far objects is due to the nature of monocular vision. Specifically, as shown in Fig. 1(a), equal distances of different depths from the camera have distinct numbers of pixels in the image, which can lead to non-negligible pixel rounding errors when detecting far objects (see Section II for detailed analysis). Additionally, as illustrated in Fig. 1(b), camera pose variance can cause a large offset in the form of 3D boxes and in the bird's eye view [16]. To the best of our knowledge, existing Mono3D methods, such as geometric constraint-based [6], [9], [40], [62], pseudo-LiDAR-based [17], [33], [34], [36], [45], [59], [63], and pure image-based [2], [3], [13], [25], [30], [32], [49], [53], [57], [69], [70], [71], [73], have not addressed the issue of inevitable camera pose changes in mobile scenarios.

Our paper introduces a new Mono3D method, named *MoGDE*, which aims to improve detection accuracy and robustness in mobile scenarios. Our key insight is that *the depth of an object in 3D space can be accurately determined based on the depth of the ground on which it stands*. Using the pinhole model and camera pose, we can derive the depth of each pixel corresponding to the ground. The core idea of MoGDE is to continuously estimate the ground depth while in motion and then use this information to guide the Mono3D detector.

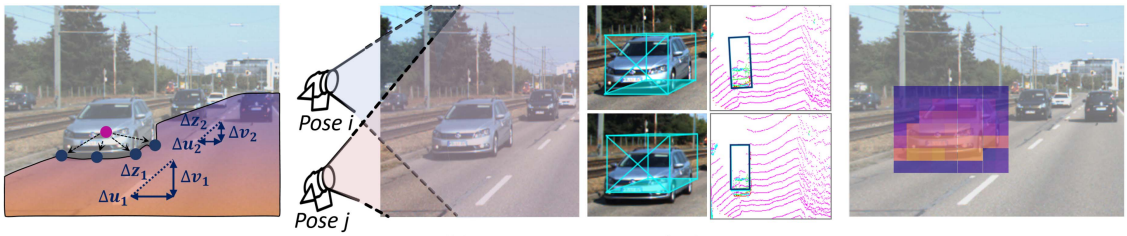
The design of MoGDE faces two primary challenges. The first is detecting the varying camera pose, such as the pitch and roll angles, in dynamic mobile scenes and obtaining accurate ground depth information, which is non-trivial. Different camera poses correspond to different ground depth estimates, as explained in Section III-C3, requiring careful consideration. To address this challenge, we propose using a pose detection network to extract vanishing point and horizon information from an image, allowing us to estimate the camera pose corresponding to the image. Once the view direction of the camera is determined,

Received 8 May 2023; revised 26 July 2024; accepted 9 January 2025. Date of publication 15 January 2025; date of current version 6 March 2025. This research was supported in part by the National Natural Science Foundation of China under Grant 62432008, in part by the Natural Science Foundation of Shanghai under Grant 22ZR1400200, and in part by the National Natural Science Foundation of China under Grant 62472083. Recommended for acceptance by P. Arbelaez. *(Corresponding author: Hongzi Zhu.)*

Yunsong Zhou, Quan Liu, Hongzi Zhu, Yunzhe Li, and Minyi Guo are with the Department of Computer Science and Technology, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: hongzi@cs.sjtu.edu.cn).

Shan Chang is with the Donghua University, Shanghai 201620, China.

Digital Object Identifier 10.1109/TPAMI.2025.3529084



(a) Near-far disparity phenomenon (b) Camera pose variation (c) Self-attention

Fig. 1. (a) Equal distances of different depth from the camera (e.g., $\Delta z_1 = \Delta z_2$) have distinct number of pixels in image (e.g., $\Delta u_1 > \Delta u_2$ and $\Delta v_1 > \Delta v_2$), which is referred to as the near-far disparity phenomenon of monocular vision, making the detection of far object susceptible to pixel rounding errors. (b) The camera pose variance, caused by the movement of a mobile agent, can eventually result in a large offset both in form of 3D boxes and in the bird's eye view. (c) Each color block represents its attention value with the centroid of the vehicle. The attention mechanism of the transformer network can be well leveraged for this long-range relationship modeling.

we construct a feature map that highlights pixel-level depth clues. Specifically, we create a virtual 3D scene containing only the sky and ground and project it onto an image, with each pixel associated with a unique depth derived from 3D-to-2D perspective geometry. By doing so, MoGDE can obtain dynamic ground depth information as prior knowledge to guide Mono3D.

The second challenge involves incorporating the estimated ground depth into the image features to improve detection accuracy, which is also challenging. To achieve this, it is crucial for the Mono3D detector to identify the *ground-contacting points* of an object in the image, as illustrated by the blue dots in Fig. 1(a). To address this, we propose an RGB-D feature fusion network based on the transformer structure that ties the ground depth feature to the image feature. As depicted in Fig. 1(c), the feature fusion network captures the features of pixels near an object's centroid, identifies the ground-contacting points using the attention mechanism, and attaches depth values with weights to compute a new feature map that contains location information. This allows for accurate 3D detection results using a conventional Mono3D detector with the fused feature map.

Experiments on KITTI dataset [18] demonstrate that our method outperforms the SOTA methods by a large margin. Such a framework can be applied to existing detectors and is practical for industrial applications. The proposed MoGDE is ranked *number one* on the KITTI 3D benchmark by submission. The whole suite of the code base will be released and the experimental results will be posted to the public leaderboard. We highlight the main contributions made in this paper as follows: 1) A novel Mono3D detector in a mobile setting is introduced, leveraging the dynamically estimated ground depth as prior knowledge to improve the detection accuracy and robustness for both near and far objects. 2) A transformer-based feature fusion network is designed, which utilizes the long-range attention mechanism to effectively identify ground-contacting points and pin the corresponding ground depth to the image feature map. 3) Extensive experiments on the real-world KITTI dataset are conducted and the results demonstrate the efficacy of MoGDE.

II. Theoretical Analysis on Monocular Vision

A general camera model can be characterized by a simple pinhole camera model plus lens distortions. In the pinhole camera model, as shown in Fig. 2, a point Q in the physical world is projected through a pinhole (referred to as the projection center)

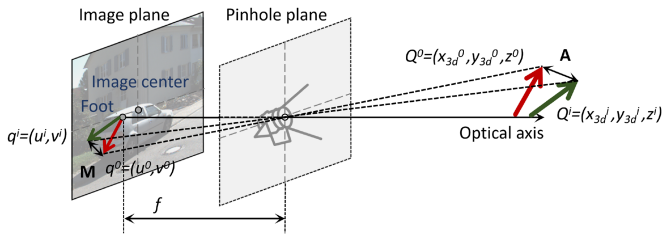


Fig. 2. The fundamental pinhole model of a camera. When the camera view changes from $\mathbf{P}^0$ to $\mathbf{P}^i$, if the camera coordinate system is re-established with the new view, the object Q^0 is moved to Q^i *w.r.t.* the camera through a transformation matrix $\mathbf{A}$. Correspondingly, the transformation matrix of the two points on the image is $\mathbf{M}$.

to a point q on the image plane. We have omitted the superscript for ease of writing. Given the coordinates of point Q in the world coordinate system, (x_{3d}, y_{3d}, z), if the camera coordinate system coincides with the world coordinate system, the pixel location of the projected point q on the virtual image plane, (u, v), satisfies the following equations:

$$u = f_x \cdot \frac{x_{3d}}{z} + c_x, \quad v = f_y \cdot \frac{y_{3d}}{z} + c_y, \tag{1}$$

where f_x and f_y are the focal lengths represented in the units of pixels along the x- and y-axis of the image plane and c_x and c_y are the possible displacements between the image center and the foot point. These are called the intrinsic parameters $\mathbf{K}$ of the camera. For simplicity, we can write it as a matrix multiplication in the form of $q = \frac{1}{z}\mathbf{K}Q$ (when necessary for matrix operations, q needs to be filled with 1 to form a 3×1 vector, i.e., $q = [u, v, 1]^T$).

From the above (1), it can be seen that the depth variation is inversely proportional to the pixel disparity, i.e., the further away the object is, the smaller the disparity would be. This makes this monocular vision system have a high depth resolution only for objects relatively close to the camera and a low depth resolution for objects far away from the camera, due to rounding errors in measuring the difference in pixels. We refer to this phenomenon as "near-far diversity" in depth estimation in the monocular vision system. To verify this, we measure the depth error of objects of different ranges and plot the cumulative distribution function (CDF) in Fig. 3. It can be seen that the error of the measurement increases with the depth of the object.

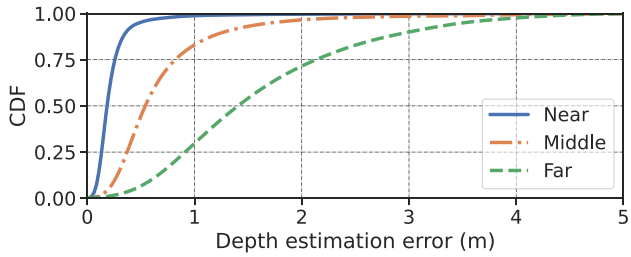


Fig. 3. CDF of depth estimation error with monocular vision in KITTI *val* set. Near (5 m–10 m), middle (25 m–30 m), and far (55 m–60 m) are three different distance intervals.

III. Design of MoGDE

A. Overview

The core idea of MoGDE is to utilize dynamically estimated ground depth information to improve Mono3D so that two goals can be achieved: 1) superior ground-aware image features are obtained to increase Mono3D accuracy for both near and far objects; 2) the impact of camera pose variation is diminished to enhance Mono3D robustness in mobile settings. Fig. 4 depicts the architecture of our framework. Specifically, MoGDE first adopts the DLA-34 [68] as its backbone, which takes a monocular image of size $(W \times H \times 3)$ as input and outputs a feature map of size $(W_s \times H_s \times C)$ after down-sampling with an s-factor. Then, the feature map is fed into three components as follows:

Ground Depth Estimation (GDE): GDF mainly integrates two functions, i.e., *camera pose detection* (CPD), and *virtual scene construction* (VSC). Specifically, CPD estimates the camera pose (i.e., the pitch and roll angles) based on the predicted vanishing point and ground plane extracted by a pose detection network. VSC establishes a 2D ground depth feature map based on a pose-specific virtual 3D scene containing only the sky and the ground.

Ground Depth Fusion (GDF): GDF leverages the attention mechanism of a transformer network to fuse the image features with the ground depth feature map, producing a superior ground-aware fused feature map.

Monocular 3D Detection (M3D): MoGDE employs GUPNet [32], a SOTA CenterNet [71] based SOTA monocular 3D object detector as its underlying detection core. For 2D and 3D detection, we follow standard procedures in this domain.

In the following sections, we will first introduce the principles of monocular vision (Section III-B), along with the Ground Depth Estimation (Section III-C) and Ground Depth Fusion (Section III-D) proposed in this paper. Finally, we will delve into the Extension on Transformer Paradigms (Section III-E).

B. Preliminary on Monocular Object Detection

MoGDE employs GUPNet [32] as its underlying detection core. There exist seven output branches with each having size of $(W_s \times H_s \times m)$, where m is the output channel of each branch. The model outputs are 2D and 3D bounding boxes describing the category, position, angle, and size information of the objects in the scene.

The 2D object detection is built on CenterNet [71], which includes a backbone network and three 2D detection subheads to compute the location, size, and confidence for each potential 2D box. The model outputs are 2D bounding box parameterized by $[x, y, w_{2d}, h_{2d}]$. The heatmap head computes a heatmap with the size of $(W_s \times H_s \times c)$ to indicate the coarse locations and confidences of the objects in the given image, where c is the number of categories. Based on that, a 2D offset regression branch computers the bounding box center (x, y), and a 2D size branch predicts the size (w_{2d}, h_{2d}) for each box. Their loss functions are denoted as $\mathcal{L}_{heatmap}$, $\mathcal{L}_{loc2d}$ and $\mathcal{L}_{size2d}$.

The 3D object detection constructs several sub-heads on top of RoI features to predict some basic 3D bounding box information. The model outputs are 3D bounding box parameterized by $[u, v, z, w_{3d}, h_{3d}, l_{3d}, \alpha]$. A 3D offset regression branch and a depth prediction branch aim to estimate the 3D center projection on the 2D feature maps (u, v) and depth z, respectively [13]. And the 3D size branch estimates the 3D dimension parameters $[w_{3d}, h_{3d}, l_{3d}]$, including height, width and length. The angle prediction branch predicts the relative yaw angle α using Multi-bin [39]. These predictions are supervised by $\mathcal{L}_{loc3d}$, $\mathcal{L}_{size3d}$ and $\mathcal{L}_{angle}$.

C. Ground Depth Estimation

1) Camera Pose Detection: In order to generate a ground depth estimate, it is key to detect the camera pose given an image feature map. We have the following proposition:

Proposition 1: Given a benchmark camera coordinate system $\mathbf{P}^0$, which is aligned with the ground plane coordinate systems, and the current camera coordinate system $\mathbf{P}^i$, which is not aligned with $\mathbf{P}^0$ due to camera movement, there exists a transformation matrix $\mathbf{A}$ between $\mathbf{P}^i$ and $\mathbf{P}^0$ that can be uniquely determined by pitch θ_p and roll θ_p angle changes of the camera. (see detailed analysis in Section III-C3.)

Therefore, we introduce the subsequent neural network to learn the pitch θ_p and roll θ_p angle changes of the camera when the camera coordinate system changes from $\mathbf{P}^0$ to $\mathbf{P}^i$. Specifically, in addition to the regular regression tasks in CenterNet [71] based network, we introduce a regression branch for pose detection following MonoEF [73]. Since the camera pose is a feature that is implicit in images, we chose two physical quantities with a clear meaning for detection: the ground plane (associated with roll angle) and the vanishing point (associated with pitch angle). Following the state-of-the-art odometer framework in DeepVP [7], we represent a regression task with L1 loss as:

$$[\hat{\mathbf{y}}_{\text{gp}}, \hat{\mathbf{y}}_{\text{vp}}] = f^{\text{pose}}(\mathbf{H}),$$
$$\mathcal{L}_{\text{pose}} = \|\mathbf{A} - \mathbf{g}(\hat{\mathbf{y}}_{\text{gp}}, \hat{\mathbf{y}}_{\text{vp}})\|, \tag{2}$$

where $\mathbf{H}$ is the input image feature; f^{pose} is the CNN architecture used for the horizon and vanishing point detection in the work [22]; $\hat{\mathbf{y}}_{\text{gp}}$ and $\hat{\mathbf{y}}_{\text{vp}}$ are the predicted ground plane and vanishing point; $\mathbf{g}$ is a mapping function $\mathbf{g} : (\mathbb{R}^2, \mathbb{R}^2) \mapsto \mathbf{A}_{3\times3}$ which turns pitch and roll angles into a matrix $\mathbf{A}$. The regression network is supervised by $\mathcal{L}_{\text{pose}}$ and can be trained jointly with other Mono3D branches.

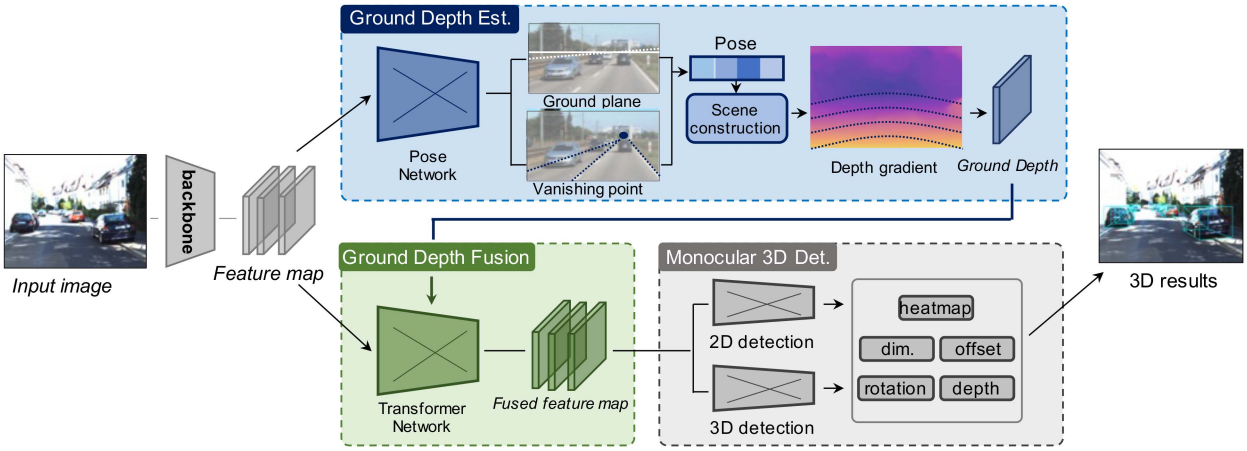


Fig. 4. MoGDE consists of three main components, i.e., *ground depth estimation* (GDE), *ground depth fusion* (GDF) and *monocular 3D detection* (M3D). In GDE, the pose network predicts the ground plane as well as the vanishing point. The derived pose information is then used to construct a virtual scene and obtain a pose-specific ground depth feature map. In GDF, a transformer network is leveraged to fuse the image features with the ground depth feature map, resulting a ground-aware fused feature map. M3D employs a standard Mono3D detector as the underlying detection core.

2) Virtual Scene Construction: We envision such a virtual scene, where there is a vast and infinite horizontal plane in the camera coordinate system $\mathbf{P}^0$, and have the following proposition:

Proposistion 2: Given the camera coordinate system $\mathbf{P}^i$, the virtual horizontal plane can be projected on the image plane of the camera according to the ideal pinhole camera model (see Section II) and the depth corresponding to each pixel on the image is determined by the camera intrinsic parameter $\mathbf{K}$ and pose matrix $\mathbf{A}$ from $\mathbf{P}^0$ to $\mathbf{P}^i$.

We first construct the ground depth feature map in the camera coordinate system $\mathbf{P}^0$. Specifically, as illustrated in Fig. 5, for each pixel on the depth image locating at (u^0, v^0) with an estimated depth $\hat{z}^0$, it can be back-projected to a point $(x^0_{3d}, y^0_{3d}, \hat{z}^0)$ in the 3D scene:

$$x^0_{3d} = \frac{u^0 - c_x}{f_x}\hat{z}^0 \quad y^0_{3d} = \frac{v^0 - c_y}{f_y}\hat{z}^0, \tag{3}$$

where f_x and f_y are the focal lengths represented in the units of pixels along the x- and y-axis of the image plane and c_x and c_y are the possible displacements between the image center and the foot point. These are referred to as the camera intrinsic parameters $\mathbf{K}$. We omit the camera extrinsic $\mathbf{T}$ for the sake of simplicity, and the depth corresponding to each pixel on the image is solely determined by the camera intrinsic parameter $\mathbf{K}$ under $\mathbf{P}^0$.

Assume that the elevation of the camera from the ground, denoted as EL, is known (for instance, the mean height of all vehicles in the KITTI dataset, including ego vehicles, is 1.65 m [18]), the depth of a point on the depth feature map (u^0, v^0) can be calculated as:

$$z^0 = \frac{f_y \cdot EL}{v^0 - c_y}. \tag{4}$$

Note that (4) is not continuous when the point is near the vanishing point, i.e., $v^0 = c_y$, and does not physically hold when $v^0 \leq c_y$. To address this problem, similar to the KITTI stereo setup, we encode the depth gradient value as an associated feature map using a virtual stereo setup with baseline $B = 0.54$m. We represent the ground depth d in the following form:

$$d = \mathrm{ReLU}\left(f_y \cdot B\frac{v^0 - c_y}{f_y \cdot EL + b}\right) \tag{5}$$

where b is a constant to prevent the value of d from being too large. The ReLU activation is applied to suppress ground depth values smaller than zero, which is not physically feasible for monocular cameras. As a result, the ground depth feature map becomes spatially continuous and consistent.

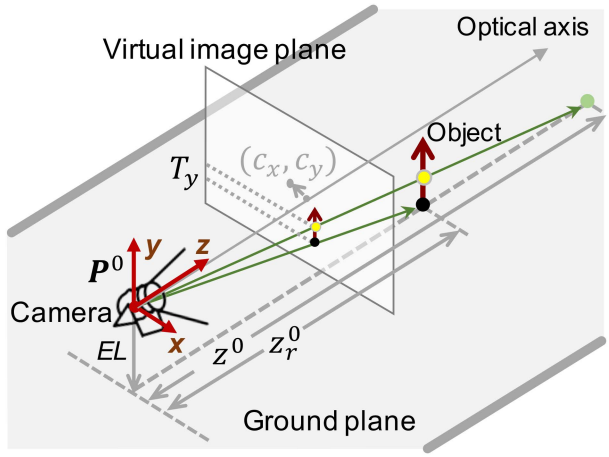


Fig. 5. Perspective geometry for ground depth estimation. In the camera coordinate system $\mathbf{P}^0$, given the camera intrinsic parameters $\mathbf{K}$ and the elevation of the camera from the ground EL, the depth of a point on the ground depth feature map can be calculated. Moreover, by estimating the transformation matrix $\mathbf{A}$ between $\mathbf{P}^0$ and an arbitrary $\mathbf{P}^i$, the ground depth feature map in $\mathbf{P}^i$ can be obtained. To utilize the ground depth feature, it is key to locate *ground-contacting* points of an object (e.g., the dark point) to get an accurate depth (e.g., z^0_r). On the contrary, misuse of the depth in the ground depth feature corresponding to other points on the object (e.g., the bright point) leads to obvious depth estimation error (e.g., z^0).

Finally, to obtain the ground depth feature map in $\mathbf{P}^i$, the model needs to convert the 3D coordinate system by (6) first, and then just apply (5). We omit the formula derivation due to page limitation. Please refer to (9) in the Section III-C3 for the detailed formula form.

3) Impact of Camera Pose Variance to Depth Estimation: As shown in Fig. 1(b), the current datasets and models assume that the Mono3D model works statically at an ideal fixed viewport which is always aligned with the horizon (top row). They simply assume that the pose of the camera does not change during travel for the convenience of data processing. When these methods are used in the dynamic scene shown in the bottom row, the object depth seen by the camera deviates from the real object depth, which will eventually result in a large offset in the form of both 3D boxes and bird's eye view. A fatal reason is that they tend to ignore a problem that often occurs in real scenes but rarely gets attention: changes in camera pose lead to changes in the depth distribution of the monocular scenario. According to the findings in [16], the estimation of object depth by mainstream models is heavily dependent on the longitudinal position of the object on the 2D image, so when the camera pose changes make the 2D position of the object change, which is common in driving scenes, the depth estimation is then severely biased.

As shown in Fig. 2, when the camera pose changes so that the camera coordinate system $\mathbf{P}^i$ is no longer consistent with $\mathbf{P}^0$, there is the following transformation relationship between the coordinate points of the same object in both coordinate systems:

$$\begin{bmatrix} x_{3d}^0 & y_{3d}^0 & z^0 \end{bmatrix}^T = \mathbf{A}^{-1} \cdot \begin{bmatrix} x_{3d}^i & y_{3d}^i & z^i \end{bmatrix}^T, \tag{6}$$

where the matrix $\mathbf{A}$ represents the transfer matrix of the camera coordinate system from $\mathbf{P}^0$ to $\mathbf{P}^i$ due to the change of pose. Specifically, it can be written as:

$$\mathbf{A} = \begin{bmatrix} \cos\theta_r & \sin\theta_r & 0 \\ \cos\theta_p \sin\theta_r & \cos\theta_r \cos\theta_p & \sin\theta_p \\ -\sin\theta_p \sin\theta_r & -\sin\theta_p \cos\theta_r & \cos\theta_p \end{bmatrix}, \tag{7}$$

where θ_p stands for pitch angle and θ_r for roll angle of ego car *w.r.t* the line of sight of $\mathbf{P}^0$ respectively. Now we are equipped with the pose variance being introduced spatially, where $Q^i = z^i(\mathbf{K}^i)^{-1}q^i = \mathbf{A}Q^0$ in the form of the matrix. On the feature map, the object point shifts correspondingly from q^0 to q^i. The transformation relationship $\mathbf{M}$ of points in a 3×3 shape can be represented as:

$$q^i = \mathbf{M}q^0 = \frac{z^0}{z^i}\mathbf{K}^i\mathbf{A}(\mathbf{K}^0)^{-1}q^0. \tag{8}$$

In the coordinate system $\mathbf{P}^i$, (3) no longer holds, and continuing to utilize this equation (as most existing work does) would cause a non-trivial error [73]. This shift in image coordinates would confuse the prediction of 3D position in CenterNet [71] based methods.

To avoid this problem, it is first necessary to let the model acquire the pose information, for which we designed the Ground Depth Generation module in Section III-C. Considering the change of camera pose, the original depth projection relation of (4) needs to be changed. Specifically, points related to the computation of z^i need to be transformed from the coordinate system of matrix $\mathbf{A}$ and $\mathbf{M}$ to $\mathbf{P}^0$ before being substituted into the formula. This transformation involves parameters from matrices $\mathbf{A}$ and $\mathbf{M}$. The process can be written in the following form:

$$\begin{aligned} z^i &= \frac{f_x f_y EL - a_{32} f_x EL \times \Upsilon}{\Upsilon(a_{31}(u^i - c_x) + a_{33} f_x)}, \\ \Upsilon &= m_{21}u^i + m_{22}v^i + m_{23} - c_y, \end{aligned} \tag{9}$$

where z^i is the depth of a point under e camera coordinate system $\mathbf{P}^i$, Υ represents the distance of a pixel from the optical center, a and m are the elements on the corresponding rows and columns of matrices $\mathbf{A}^{-1}$ and $\mathbf{M}^{-1}$, respectively. This gives the model an expression for the depth prior that is universal in any camera view.

D. Ground Depth Fusion

1) Ground-Aware Transformer: In real-world scenarios, as depicted in Fig. 5, objects have height. To fuse the image feature and the ground depth feature, it is key to locate *ground-contacting* points of an object (e.g., the dark point) to get an accurate depth (e.g., z_r^0). On the contrary, misuse of the depth in the ground depth feature corresponding to other points on the object (e.g., the bright point) leads to obvious depth estimation error (e.g., z^0). To formalize the analysis of the resulting errors, in Fig. 5, we can construct proportionality relationships of similar triangle side lengths, including pixel displacement, to calculate the estimated depth. Specifically, the relation between the estimated depth of an object and the pixel displacement in locating ground-contacting points can be calculated as:

$$\hat{z}^0 = \frac{EL \cdot f_y \cdot z_r^0}{EL \cdot f_y - z_r^0 \cdot T_y}. \tag{10}$$

where $\hat{z}^0$ is the estimated depth of an object, z_r^0 is the real depth, T_y is the vertical displacement in locating ground-contacting points. It can be seen from (10) that T_y can cause inaccurate $\hat{z}^0$. However, how to acquire T_y is non-trivial. Inspired by the great success of transformer [5], [37], [38], [60], [67], [74] in adaptive long-range relational modeling, we propose a *ground-aware* feature fusion method based on a transformer structure as depicted in Fig. 6, leveraging its attention mechanism to automatically locate ground-contacting points of an object and fuse the corresponding depth feature with the image feature of that object.

Encoder: Our transformer encoder aims to encode the correlation between image features using a self-attention mechanism. The input of the transformer encoder is the flattened image features $\mathbf{H}_{\text{img}} \in \mathbb{R}^{N \times C}$ with position encoding and the output is the embedding vectors $\mathbf{H}_e \in \mathbb{R}^{N \times C}$ to be sent to the decoder. Following the self-attention pipeline, given the input matrix calculated from the image features: query $\mathbf{Q} \in \mathbb{R}^{N \times C}$, key $\mathbf{K} \in \mathbb{R}^{N \times C}$, and value $\mathbf{V} \in \mathbb{R}^{N \times C}$ with sequence length $N = W \times H$, the output of $l + 1$th layer of self-attention can

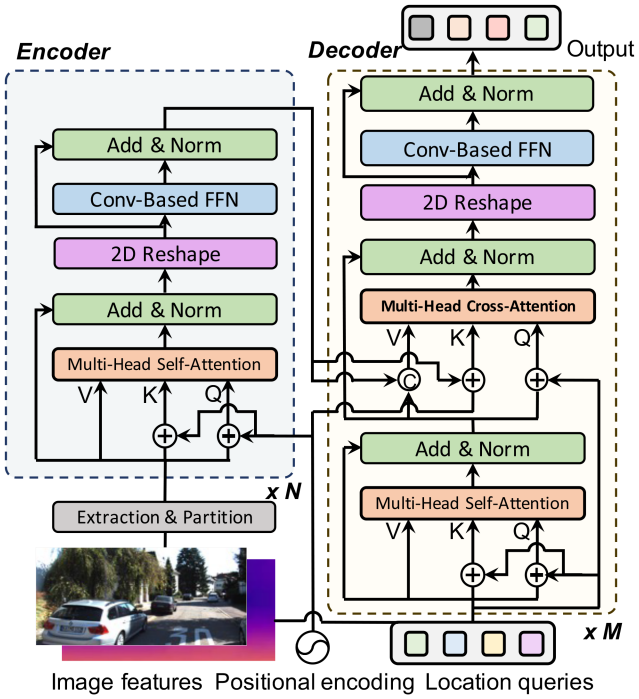


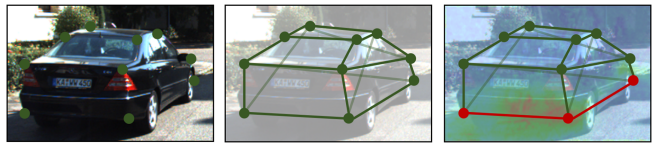
Fig. 6. The architecture of *ground-aware* transformer. The encoder uses self-attention to encode the non-local mutual correlation of image pixels (i.e., object centers and ground points). The ground depth estimate is used to generate location queries which are thereby fed to the decoder along with the location encoding. The cross-attention in the decoder prompts each query to consider the image and depth features of its associated points.

be briefly formulated as:

$$\begin{aligned}\mathbf{Q}^l, \mathbf{K}^l, \mathbf{V}^l &= \text{Embedding}(\mathbf{H}^l_{\text{img}}, \mathbf{W}^l_q, \mathbf{W}^l_k, \mathbf{W}^l_v),\\ \mathbf{H}^{l+1}_{\text{img}} &= \text{Attention}(\mathbf{Q}^l, \mathbf{K}^l, \mathbf{V}^l)\\ &= \text{softmax}\left(\mathbf{Q}^l {\mathbf{K}^l}^\top / \sqrt{C}\right)\mathbf{M}^l\mathbf{V}^l. \end{aligned} \tag{11}$$

Here, $\mathbf{M}^l$ is the mask used to constrain the visible range of attention. The introduction of $\mathbf{M}^l$ is to take advantage of the *ground-aware* property (i.e., the depth of each object should be related to the depth of the object's location) so that each pixel will only consider information within a window around that location. The encoded feature obtained through multi-head self-attention operation is then re-transformed into a 2D feature map format and fed into a convolution-based feed-forward network (FFN). The 2D reshape as well as convolution-based FFN are necessary because image data is two-dimensional, unlike one-dimensional serialized data.

Decoder: The proposed transformer decoder aims to determine for each location its depth information, using the cross-correlation between the ground depth and the image features. We propose utilizing the ground depth as the location query of the decoder instead of learnable embedding (object query), which is different from the common usage in previous encoder-decoder vision transformer works [5], [74]. The main reason is that the simple learnable embedding is hard to fully represent the object's property and handle complex depth variant situations in the Mono3D task. In contrast, plentiful distance-aware cues are hidden in the ground depth features, which will give the transformer a baseline estimate of the expected depth at each location. To this end, the decoder can leverage the power of cross-attention in the transformer to efficiently model the correlation between the target pixel point and the point of interest (i.e., the grounded point), thus achieving the *ground-awareness* for higher performance.



(a) Semantic key points (b) 2D-3D Edge Graph (c) Foot Point Focal Loss

Fig. 7. (a) Schematic diagram of the semantic key points detected by [31], e.g., the green points in the figure indicate the corner points associated with the vehicle contour. (b) The graph formed by giving regular concatenated edges from 2D semantic points lift to 3D space, where transparent concatenated edges indicate the part of the concatenation that is in invisibility. (c) The focal loss calculated between the foot points of the picked instances (i.e., the red edges and lines) and the attention map between the center of the instance and the pixels at other locations in the image as output by Transformer's encoder.

Specifically, the input of the decoder is the flattened ground depth $\mathbf{H}_{\text{dep}} \in \mathbb{R}^{N\times 1}$ with position encoding and embedding vectors $\mathbf{H}_e \in \mathbb{R}^{N\times C}$ obtained from the encoder. In addition, the output is the aggregated feature map $\mathbf{H}_d \in \mathbb{R}^{N\times C}$. The ground depth is first embedded upon the standard self-attention architecture following (11). For the cross-attention module, its input $\mathbf{Q}$ is derived from the self-attention part upstream in the decoder, and its $\mathbf{K}$ is derived from the encoder. The input $\mathbf{V}$ is a concatenation of two sources from both the encoder and decoder. The purpose of this concatenation is to make the decoder take into account both the information from the image and the depth during decoding.

2) Pseudo Foot Point Labeling: To help the transformer achieve convergence, we need to generate pseudo-labels for the foot points in the image to help the training of the encoder explicitly. Adapted from [31], the labeling starts from generating semantic key points of each object. Fig. 7(a) illustrates an object which has n semantic key points, the ith (i=1,2,...,n) key point's 2D coordinate is $(u, v)_i$. Then, we can get the depth z_i from the point cloud annotation corresponding to the image, and use (3) to lift the 2D coordinate points $(u, v)_i$ to 3D coordinate points $(x_{3d}, y_{3d}, z)_i$. A rule-based approach built on 3D spatial distances is used to establish a few instance graphs, which is shown in Fig. 7(b). By comparing it with some models built in [26] with 3D points segmented from the points cloud and 2D masks, we can pick out the parts that can be used for example foot points and bottom edges. It is worth mentioning that those points and edges that are not visible are partially excluded, considering the orientation and position of the target. As shown in Fig. 7, the points and edges in red are used to compute the focal loss with the attention map of the transformer's encoder to help the model focus on the grounded part of the target.

E. Extension on Transformer Paradigms

Inspired by the success of transformers in natural language processing, visual transformers with long-range attention between image patches have been developed to address Mono3D

tasks and achieve state-of-the-art (SOTA) performance [20], [69]. Because our ground depth fusion module is built on a transformer structure, it seamlessly integrates into the current design of the transformer architecture. Moreover, the pure transformer structure enables this model to be more easily scalable. Therefore, we introduce a new version of a model based on a transformer structure, called MoGDE†.

Specifically, the feature extraction component of the model has been updated from the original DLA-34 to ViT, which encodes images into tokens representing features. In Ground Depth Estimation, the model directly learns camera parameters from these tokens to generate a depth map, which is then divided into patches that can also serve as tokens. During Ground Depth Fusion, image features and depth tokens are concatenated and fed into the module described in Section III-D. Finally, the model leverages the network structure of MonoDETR [69] to achieve 3D object detection.

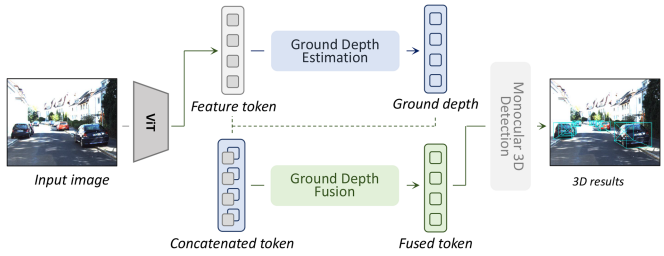


Fig. 8. An attempt to extend the conference version of the model with a transformer structure. We utilize ViT for extracting features from images and representing them as feature vectors, referred to as tokens. These tokens can seamlessly integrate into the ground depth fusion module and be utilized with a transformer-based decoder for 3D object detection.

IV. Performance Evaluation

We conduct experiments on the widely-adopted KITTI3D dataset and KITTI Odometry dataset [18]. We report the detection results with three-level difficulties, i.e., easy, moderate, and hard, in which the moderate scores are normally for ranking and the hard category is generally distant objects that are difficult to distinguish. For the detailed dataset statistics, training structure, learning rules, evaluation metrics, *etc.*, please refer to the Section IV-A.

A. Implementation Details

Following [32], we adopt the modified DLA-34 [68] as our backbone. The input image resolution is 380×1280 for the KITTI dataset. The downsampling factor s of the model is set to 4 and the number of feature map channels C is 64. The pose detection network follows the modified AlexNet's architecture [7] and the transformer follows the DETR [5] modifying 1-dimensional FFN to convolution-based FFN. The pose network needs to be pre-trained on the KITTI Odometry dataset beforehand. The ground truth labels for the pose detection module come from the given camera extrinsic calibration in the dataset. Following DCD [26], we set corresponding keypoint labels for each object to be detected. Then all of MoGDE's networks can be trained together in an end-to-end manner on the KITTI dataset. Alternatively, each module can be trained separately and then stitched together for fine-tuning. The whole model is trained for 140 epochs with a batch size of 32 on four NVIDIA 3090 GPUs simultaneously. The initial learning rate is 1.25e-4, dropped by multiplying 0.1 both at 90 and 120 epochs. We use the ADAM optimizer with a weight decay of 1e-5.

For the evaluation and ablation study, we show experimental results from two different setups. The **Baseline** is derived from GUPNet [32] and **MoGDE** is the final proposed method integrating Ground Depth Estimation and Ground Depth Fusion. The KITTI3D dataset uses a 40-point interpolated average precision metric AP_{40} that averages precision results on 40 recall positions except for the one where recall is 0. The precision is evaluated at both the bird's eye view 2D box AP_{BEV} and the 3D bounding box AP_{3D} in world coordinates. We report average precision with intersection over union (IoU) using 0.5 and 0.7 as thresholds.

Regarding the robustness test experiments against attitude changes, we have the following setup. Scene the KITTI3D dataset is initially without pose information, we simulate the camera pose variance in the mobile scenes using an artificial Gaussian function ($\theta_p, \theta_r \sim N(0, \sigma)$), where σ is the standard deviation. We establish three settings with different degrees of pose variation: tiny ($\sigma = 1$), medium ($\sigma = 2$), and large ($\sigma = 3$), respectively. The pitch θ_p and roll θ_r will determine a unique transformation matrix $\mathbf{A}$ and $\mathbf{M}$. For the evaluation, the original 3D coordinates are transformed using matrix $\mathbf{A}$ according to (8). The input image is also processed using matrix $\mathbf{M}$ following MonoEF [73].

B. Quantitative and Qualitative Results

We first show the performance of our proposed MoGDE on KITTI 3D object detection benchmark[1] for car. Comparison results with other state-of-the-art (SOTA) monocular 3D detectors are shown in Table I. For the official *test* set, we achieve the highest score for all kinds of samples and is ranked No.1 among all existing methods with different additional data inputs on all metrics. Compared to the second-best models, MoGDE surpasses them under easy, moderate, and hard levels respectively by +1.72, +1.20, and +1.27 in AP_{3D}, especially achieving a significant increase (8.7%) in the hard level. The comparison fully proves the effectiveness of the proposed oracle fusion for images with prior depth knowledge.

We conduct experiments for the car category on the KITTI *val* set in Table II. Our approach achieves superior performance over several methods, especially on far away objects (i.e., the hard category), benefiting from leveraging the ground plane information. Specifically, our method outperforms GUPNet [32] with an improvement of +3.99 in AP_{3D} on the hard category.

Fig. 9 shows the qualitative results on the KITTI Odometry dataset. Compared with the baseline model without the aid of ground depth, the predictions from MoGDE are much closer to the ground truth, especially for distinct objects. It shows that the

[1]http://www.cvlibs.net/datasets/kitti/eval object.php?obj benchmark=3d

TABLE I
AP_{40} SCORES(%) OF THE CAR CATEGORY ON KITTI *TEST* SET AT 0.7 IOU THRESHOLD REFERRED FROM THE KITTI BENCHMARK WEBSITE

Method	Extra data	Test, AP_{3D}			Test, AP_{BEV}		
		Easy	Mod.	Hard	Easy	Mod.	Hard
PatchNet [33]	Depth	15.68	11.12	10.17	22.97	16.86	14.97
D4LCN [17]		16.65	11.72	9.51	22.51	16.02	12.55
DCD [26]		23.81	15.90	13.21	32.55	21.50	18.25
DID-M3D [42]		24.40	16.29	13.75	32.95	22.76	19.83
MonoDDE [27]		24.93	17.14	15.10	33.58	23.46	20.37
Kinematic3D [3]	Multi-frames	19.07	12.72	9.17	26.69	17.52	13.10
DfM [56]		29.27	20.22	17.46	38.60	27.13	24.05
MonoRUn [8]	Lidar	19.65	12.30	10.58	27.94	17.34	15.24
CaDDN [47]		19.17	13.41	11.46	27.94	18.91	17.19
LPCG-Monoflex [41]		25.56	17.80	15.38	35.966	24.81	21.86
AutoShape [31]	CAD	22.47	14.17	11.36	30.66	20.08	15.59
SMOKE [30]	None	14.03	9.76	7.84	20.83	14.49	12.75
MonoDLE [35]		17.45	13.66	11.68	24.97	19.33	17.01
MonoFlex [70]		19.94	13.89	12.07	28.23	19.75	16.89
DFR-Net [75]		19.76	14.10	10.47	27.83	19.72	15.20
GUPNet [32]		20.11	14.20	11.77	-	-	-
MonoDTR [20]		21.99	15.39	12.73	28.59	20.38	17.14
MonoDETR [69]		23.65	15.92	12.99	32.08	21.44	17.85
MonoCD [64]		25.53	16.59	14.53	33.41	22.81	19.57
MonoUNI [21]		24.75	16.73	13.49	33.28	23.05	19.39
MoGDE		27.07	17.88	15.66	38.38	25.60	22.91
MoGDE†		**27.25**	**17.93**	**15.80**	**38.84**	**26.02**	**23.27**
Improvement	*v.s. second-best*	+1.72	+1.20	+1.27	+5.43	+2.97	+3.70

We utilize bold to highlight the best results and color the second-best ones and our performance gain over them in blue. Our model is ranked NO. 1 on the benchmark.†: the journal version with the extension on transformer paradigms.

Fig. 9. Qualitative results on KITTI Odometry dataset. The predicted 3D bounding boxes of our proposed MoGDE are shown in the first row. The second row shows the detection results in the bird's eye view (z-direction from right to left). The green dashed boxes are the ground truth, and the blue and red solid boxes are the prediction results of our MoGDE and the comparison baseline (GUPNet [32]), respectively. The third row visualizes the results of the attention map in the transformer's encoder, where the purple point is the location of the query point; the yellow dashed box is the range of the encoder's mask; and the brightness of the image represents the attention value between the query point and that pixel. The forth row illustrates the depth map generated from the ground depth fusion module.

TABLE II
DETECTION PERFORMANCE OF CAR CATEGORY ON THE KITTI *VAL* SET

Method	Val, AP_{3D}			Val, AP_{BEV}		
	Easy	Mod.	Hard	Easy	Mod.	Hard
M3D-RPN [2]	14.53	11.07	8.65	20.85	15.62	11.88
MonoPair [13]	16.28	12.30	10.42	24.12	18.17	15.76
Kinematic3D [3]	19.76	14.10	10.47	27.83	19.72	15.10
MonoRUn [8]	20.02	14.65	12.61	-	-	-
CaDDN [47]	23.57	16.31	13.84	-	-	-
GUPNet [32]	22.76	16.46	13.72	31.07	22.94	19.75
MonoFlex [70]	**23.64**	17.51	14.83	-	-	-
MoGDE (Ours)	23.35	**20.35**	**17.71**	**31.36**	**25.41**	**24.36**

We utilize bold to highlight the best results and color the second-best ones. The metric is AP_{40}.

TABLE III
DETECTION PERFORMANCE OF PEDESTRIAN AND CYCLIST CATEGORIES ON THE KITTI *TEST* SET AT 0.5 IOU THRESHOLD

Method	Test, AP_{3D}, Ped.			Test, AP_{3D}, Cys.		
	Easy	Mod.	Hard	Easy	Mod.	Hard
MonoDLE [35]	9.64	6.55	5.44	4.59	3.66	2.45
D4LCN [17]	4.55	3.42	2.83	2.45	1.67	1.36
MonoPair [13]	10.02	6.68	5.53	3.79	2.12	1.83
DDMP3D [13]	4.93	3.55	3.01	4.18	2.50	2.32
CaDDN [47]	**12.87**	8.14	6.76	7.00	3.41	3.30
MonoFlex [70]	9.43	6.31	5.26	4.17	2.35	2.04
MoGDE (Ours)	11.27	**8.33**	**7.67**	**7.02**	**3.96**	**3.41**

We utilize bold to highlight the best results and color the second-best ones. The metric is AP_{40}.

TABLE IV
EFFECTIVENESS OF DIFFERENT COMPONENTS OF OUR APPROACH ON THE KITTI *VAL* SET FOR THE CAR CATEGORY

	Pose -guided	Conv. Fusion	Tran. Fusion	Easy	Mod.	Hard
(a)	-	-	-	22.76	16.46	13.72
(b)	✓	-	-	22.78	16.93	14.04
(c)	-	✓	-	22.82	17.22	14.51
(d)	-	-	✓	22.93	18.42	15.46
(e)	✓	✓	-	23.07	18.66	15.73
(f)	✓	-	✓	**23.35**	**20.35**	**17.71**

The first column is whether the model takes into account the pose variance. The second and third columns show which way the model chooses to fuse the ground depth information.

consideration of sight-based supporting depth clues can help to locate the object precisely.

C. Ablation Study

Effectiveness of each proposed component: In Table IV, we conduct an ablation study to analyze the effectiveness of the proposed components: (a) Baseline: only using image features for 3D object detection, i.e., without concerning posed variance and proposed ground-aware modules. (b) Considering the camera pose variations implied in the images, we use the method described in [73] to apply a "projection transform" to the input image to remove the perturbations. (c) Considering the use of ground plane clues, we generate a depth oracle about the scene (assuming constant pose) and use a convolutional neural network in (d) with the proposed *ground-aware* transformer, which has the ability to model long-range relationships of pixels. (e) Full model except that we use a convolutional neural network for oracle fusion. (f) Full model (MoGDE).

First, we can observe from (a $\rightarrow$ b, c $\rightarrow$ e, and d $\rightarrow$ f) that there is an implicit uncalibrated pose variation in the KITTI dataset, and considering it is necessary to improve the detection accuracy. Besides, by observing (b $\rightarrow$ e), we illustrate that leveraging ground depth brings an improvement in accuracy at the hard level, but the improvement is limited because fusion by convolution is clumsy.

In contrast, (e $\rightarrow$ f) indicates the effectiveness of the transformer, which helps the model to understand the long-range attention relationship between pixel points and the ground plane.

Results in all categories: To further vilify the performance of MoGDE in all categories, the full results are released publicly on the KITTI benchmark and are shown in Table III. Our MoGDE has managed to outperform its peers in the Cyclist category and is comparable to SOTAs in the Pedestrain category. The reason for the model's failure to perform as expected on the Pedestrain category may come from the Ground Depth Fusion (GDF) module. Since humans have a slender shape, the GDF module may not work well to fully identify all the key points applicable to the Pedestrain category.

Visualization of attention: To facilitate the understanding of our *ground-aware* transformer, we visualize the depth self-attention map in the encoder and paint the query points red and mask region yellow in the third row of Fig. 9. As shown in the figure, it can be seen that, within the relevant region of each query, areas that are interfacing the object and the ground have the highest attention scores. In contrast, for non-ground pixels of the object, the lower attention values indicate that the query is not relevant to them, even if they have similar image features and are geographically adjacent. This implies that under the transformer's attention mechanism guidance, the query is able to borrow depth information from regions of interest (i.e., the ground plane), which helps the fused feature map produce more accurate prediction results.

Simulation experiments on robustness: In order to verify the robustness of our proposed MoGDE against camera pose variance, we set three cases of variances (tiny, medium, and large) to compare the accuracy degradation of MoGDE with that of the baseline. The detailed experimental setup can be found in Section IV-A. In Table V, it can be noticed that the baseline is quite sensitive to pose variance, with very severe performance degradation, while our model only has a slight performance drop. Moreover, our model performs more robustly especially in the hard case, gaining higher performance improvement. This demonstrates the effectiveness of our proposed pose-specific ground depth in handling camera pose variance for mobile scenes.

Plugging into existing methods: Our proposed approach is flexible to extend to existing image-only Mono3D detectors. We respectively plug the Ground Depth Estimation and the Ground Depth Fusion components to three popular Mono3D detectors, which is shown in Table VI. It can be seen that, with the aid of our proposed ground depth fusion, these detectors can

TABLE V
ROBUSTNESS TEST OF OUR APPROACH ON THE KITTI *VAL* SET FOR CAR CATEGORY

Pose Var.	Method	Val, AP_{3D} Easy	Mod.	Hard	Val, AP_{BEV} Easy	Mod.	Hard
Tiny	w/o	20.64	14.87	12.47	27.97	20.78	17.79
	w/	**22.30**	**19.42**	**16.84**	**29.86**	**24.28**	**23.15**
	Imp.	+1.66	+4.55	+4.37	+1.89	+3.50	+5.35
Medium	w/o	17.76	12.98	10.78	24.43	18.06	15.46
	w/	**21.86**	**19.08**	**16.61**	**29.23**	**23.70**	**22.84**
	Imp.	+4.10	+6.10	+5.83	+4.81	+5.64	+7.38
Large	w/o	13.29	9.60	8.05	18.05	13.34	11.57
	w/	**21.10**	**18.35**	**16.10**	**28.33**	**22.98**	**22.06**
	Imp.	+7.81	+8.75	+8.05	+10.28	+9.64	+10.48

Tiny, medium, large correspond to three different degrees of posture change, i.e., the camera pitch and roll angles vary with a Gaussian distribution with mean 0 and standard deviation 1, 2, and 3, respectively.

TABLE VI
EXTENSION OF MOGDE TO EXISTING IMAGE-ONLY MONOCULAR 3D OBJECT DETECTORS

Method	Val, AP_{3D} Easy	Mod.	Hard	Val, AP_{BEV} Easy	Mod.	Hard
M3D-RPN [2]	14.53	11.07	8.65	20.85	15.62	11.88
M3D-RPN + **Ours**	**19.85**	**16.84**	**14.62**	**25.16**	**20.65**	**17.39**
Imp.	+5.32	+5.77	+5.97	+4.31	+5.03	+5.51
MonoPair [13]	16.28	12.30	10.42	24.12	18.17	15.76
MonoPair + **Ours**	**19.20**	**15.42**	**13.16**	**27.33**	**21.71**	**18.68**
Imp.	+2.92	+3.12	+2.74	+3.21	+3.54	+2.92
Kinematic3D [3]	19.76	14.10	10.47	27.83	19.72	15.10
Kinematic3D + **Ours**	**21.59**	**16.54**	**12.77**	**29.80**	**22.80**	**17.96**
Imp.	+1.83	+2.44	+2.30	+1.97	+3.08	+2.86

We show the AP_{40} scores(%) evaluated on KITTI3D val set. +Ours indicates that we apply the GDE and GDF modules to the original methods. All models benefit from the MoGDE design.

TABLE VII
THE COMPARISON OF THE PERFORMANCE GAIN OF OUR MOGDE OVER EXISTING MODELS AT DIFFERENT DISTANCES

Method	Val, AP_{3D} Near	Mid.	Far	Val, AP_{BEV} Near	Mid.	Far
Kinematic3D [3]	34.52	16.50	5.82	246.86	22.81	7.35
+ **Ours**	**35.70**	**21.86**	**10.48**	**48.23**	**27.84**	**12.34**
Imp.	+1.18	+5.36	+4.66	+1.37	+5.03	+4.99
MonoDTR [20]	48.51	17.87	2.16	61.25	25.03	3.29
+ **Ours**	**49.69**	**22.49**	**10.95**	**63.24**	**29.81**	**11.65**
Imp.	+1.18	+4.62	+8.79	+1.99	+4.78	+8.36
MonoDETR [69]	48.66	17.91	2.35	61.21	25.25	3.38
+ **Ours**	**49.92**	**22.48**	**11.07**	**63.29**	**30.00**	**11.91**
Imp.	+1.26	+4.57	+8.72	+2.08	+4.75	+8.53

We show the AP_{40} scores(%) evaluated on KITTI3D val set. +Ours indicates that we apply our modules to the original methods. Near (5m-10m), middle (20m-25m), and far (40m-45m) are three different distance intervals. All models benefit from the MoGDE design, especially for far objects.

achieve further improvements on KITTI3D *val* set, demonstrating the effectiveness and flexibility of our approach. Particularly, MoGDE enabled models tend to achieve more performance gains in the hard category. For example, for Kinematic3D, the AP_{3D}/AP_{BEV} gain is +1.83/+1.97 in the easy category and +2.30/+2.86in the hard category.

Detection performance for objects at different distances: In Table VII, we compare the accuracy gain effect of our model for object detection at different distances. In this table, we present

TABLE VIII
COMPARISON OF DIFFERENT MODIFICATIONS ON THE TRANSFORMER FOR CAR CATEGORY ON THE KITTI *VAL* SET

Method	Val, AP_{3D} Easy	Mod.	Hard	Val, AP_{BEV} Easy	Mod.	Hard
Baseline	22.76	16.46	13.72	31.07	22.94	19.75
+DETR	22.81	16.67	13.92	31.12	23.10	20.11
+2D FFN	22.03	17.82	15.47	31.23	23.95	22.36
+Mask	23.22	19.26	16.68	31.30	24.62	23.25
+Concat. (Ours)	**23.35**	**20.35**	**17.71**	**31.36**	**25.41**	**24.36**

Note that '+DETR', '+2D FFN', '+Mask', and '+Concat.' stand for using origin DETR, replacing FFN to 2D FFN, adding masks to all of the inputs, and concatenating image feature and depth feature, respectively. The metric is AP_{40}.

TABLE IX
COMPARISON OF DIFFERENT APPROACHES ON HOW TO USE THE KEY POINTS OF THE INSTANCE FOR CAR CATEGORY ON THE KITTI *VAL* SET

Method	Val, AP_{3D} Easy	Mod.	Hard	Val, AP_{BEV} Easy	Mod.	Hard
Baseline	22.78	16.46	13.72	31.07	22.94	19.75
Box Cons.	22.88	16.71	13.96	31.16	23.33	20.74
Shape Graph	23.17	19.37	16.49	31.22	24.81	22.98
Trans. (Ours)	**23.35**	**20.35**	**17.71**	**31.36**	**25.41**	**24.36**

Note that 'Box Cons', 'Shape Graph', and 'Trans.' stand for using key points as the depth constraint [27], leveraging car models by PCA and refining the models to 3D graph points [26], and using the transformer to fuse foot point features, respectively. For a fair comparison, the semantic points selected by all three methods are located consistently on the image.The metric is AP_{40}.

the accuracy (%) of ours with Kinemantic3D, MonoDTR, and MonoDETR as the baselines in the KITTI *val* set for three distance ranges: near (5 m–10 m), middle (20–25 m), and far (40 m–45 m). For MonoDTR, the AP_{3D}/AP_{BEV} gain is +8.79/+8.36 on the far case. From this, we can see that our method has a significant improvement in the detection accuracy of distant objects.

Results on nuScenes val set. Table X shows the experimental results of deploying our proposed approach on nuScenes [4] *val* set. Specifically, mean Average Precision (mAP) is defined by matching predictions with the ground truth objects that have the smallest center distance up to a certain threshold. And nuScenes detection score (NDS) is derived from consolidating several metrics by computing a weighted sum, including translation error, scale error, orientation error, velocity error, and attribute error. Under the same configurations (e.g., backbone and training schedule), our model achieves better performance than two 3D object detection baselines (FCOS3D [57], and PGD [58]), which demonstrates the effectiveness of our approach.

Comparison of modifications on the transformer: Table VIII shows the different effectiveness of modifications for the transformer to make it more suitable for ground plane fusion. We can observe from the table that applying 2D FFN and adding masks to the input feature can achieve considerable improvement on AP_{3D}. Besides, combining the image information in the encoder with the depth information in the decoder using concatenation is essential for fusing the ground plane information.

Fig. 10. Qualitative results on the KITTI *val* set for multi-class 3D object detection. We utilize cyan, red, and yellow colors to indicate car, pedestrian, and cyclist categories, respectively.

TABLE X
DETECTION PERFORMANCE ON NUSCENES *VAL* SET

Method	NDS ↑	mAP ↑
FCOS3D	37.7	29.8
PGD	39.3	31.7
MoGDE	**41.3**	**34.7**

We build our approach based on FCOS3D. The experiments are conducted under the same training settings. The results of baselines are taken from MMDetection3D [15].

Comparison of various utilization of foot points: Table IX shows the different usage of foot points for ground feature fusion. The experiment set following [27] uses these semantic points as the depth regression constraints to facilitate the accuracy of the regression to the center point. [26] forms the key points into a diagram in 3D space and compares it with a pre-specified CAD model to aid depth prediction. In contrast, we use the transformer's long-range attention mechanism to encode the connection between centroids and foot points, adaptively allowing centroids to capture information related to foot point depth. The improvement on AP_{3D} indicates that the transformer is more suitable for establishing the connection between central and key points based on its excellent attention mechanism. Besides, using foot points to fuse ground depth information into instance centroids is a more beneficial task for object detection than using key points to help regress depth.

Visualization for multi-class 3D object detection: In Fig. 10, we provide the additional visualization results for multi-class (car, pedestrian, and cyclist) on the KITTI *val* set. It can be observed that MoGDE can achieve accurate detection of 3D bounding boxes benefit from the aid of ground depth estimation.

V. DISCUSSION ON LIMITATIONS

In this section, we will focus on some of the limitations of the proposed MoGDE and possible ideas for improvement in future work. After our analysis, MoGDE includes two main limitations, which we will explain separately below.

First, the proposed MoGDE relies heavily on pose detection, which will directly affect the accuracy of the ground depth estimation in the virtual scene. This process of ground depth estimation needs to be based on the implicit assumption that the ground plane is flat. On the one hand, it is affected by the error of pose detection itself resulting in the estimation of pitch and roll angles not being absolutely accurate. On the other hand, when the assumption that the ground plane is flat is not valid, the continued use of ground depth estimation will produce serious errors. In Fig. 11 we visualize the two bad cases where the assumption that the ground plane is flat no longer holds. In the left image, the turning of the ego-vehicle leads to two different sections of the road surface with their own vanishing points. In the right image, a sudden drop in gradient occurs on the ramp, resulting in different vanishing points for the red and yellow sections of the road surface. In both cases the ground plane is curved and our proposed MoGDE does not work properly. To solve this problem, we intend to take a segmented approach to build the ground plane to fit the virtual scene. We will leave it as a part of our future work.

Second, the model depends on the detection of the ground-contacting point. The proposed ground-aware feature fusion module does not work well when the ground-contacting points of the target to be detected are not visible due to truncation or occlusion. In Fig. 12 we give the attention map of the transformer encoder in the truncated case. We can see that the encoder does not seem to be able to find the correct location of the

Fig. 11. The visualization of the vanishing points when the road surface is no longer flat. The red and yellow lines are auxiliary lines for the visualization of vanishing points. The red and yellow ranges of the road surface are subordinate to the vanishing points indicated by the red and yellow points, respectively.

Fig. 12. The visualization of the attention map in the transformer's encoder in the truncation case. The purple point is the location of the query point, the yellow dashed box is the range of the encoder's mask, and the brightness of the image represents the attention value between the query point and that pixel.

ground-contacting point but is rather vaguely distributed near the lower edge of the image. In the right image, the encoder even incorrectly locates the point on an unrelated vehicle. This situation leads to serious errors in the depth estimation of the model for the objects to be detected. We will improve the idea by sieving out the truncated objects in the inference on the one hand, and by trying to make the model learn the depth information not only from the ground-contacting point but also from other relevant points.

VI. Related Work

Monocular 3D Object Detection: Existing Mono3D methods can be roughly divided into the following three categories. *1) Geometric constraint-based methods:* Extra information of prior 3D vehicle shapes is widely used, such as vehicle computer-aided design (CAD) models [6], [9], [31], [40], [62] or key points [1]. By this means, extra labeling cost is inevitably required. *2) Depth assist methods:* A stand-alone depth map of the monocular image is predicted at the first stage. Such prior knowledge can be derived in various ways, such as a depth map generated by LiDAR point cloud (or Pseudo-LiDAR) [8], [34], [47], [59], monocular depth predictors [17], [33], [45], or disparity map generated by stereo cameras [63]. However, such external data is not easily available in all scenarios. In addition, the inference time increases significantly due to the prediction of these dense heatmaps. *3) Pure image-based methods:* Without requiring extra side-channel information, such methods [19], [23], [29], [52] take only a single image as input and adopt center-based pipelines following conventional 2D detectors [49], [53], [71]. M3D-RPN [2] reformulates the monocular 3D detection problem as a standalone 3D region proposal network. SMOKE [30] and FCOS3D [57] predict a 3D bounding box by combining a concise one-stage keypoint estimation with regressed 3D variables based on CenterNet [71] and FCOS [53], respectively. MonoEF [73] first proposes a novel method to capture the camera pose in order to formulate detectors that are not subject to camera extrinsic perturbations. GUPNet [32] solves the error amplification problem by geometry-guided depth uncertainty and collocates a hierarchical learning strategy to reduce the training instability. MonoDTR [20] proposes to globally integrate context- and depth-aware features with transformers and inject depth hints into the transformer for better 3D reasoning. MonoDETR [69] introduces a simple monocular object detection framework that makes the vanilla transformer to be depth-aware and enforces the whole detection process guided by depth. The above geometrically dependent designs largely promote the overall performance of center-based methods [26], [27], [35], [41], [42], [56], [75], but the underlying problem still exists, namely, the detection accuracy for distant objects is still not satisfactory.

Object Detection with Transformer: 2D object detectors [53], [71] have achieved excellent performance in recent years but are equipped with cumbersome post-processing, e.g., non-maximum suppression (NMS) [49]. To circumvent it, the pioneering work DETR [5] constructs a novel and simple framework by adapting the powerful transformer [54] to the field of visual detection. DETR is further enhanced by designing deformable attention [74], placing anchors [60], setting conditional attention [37], embedding dense prior [67], and so on [38]. Some recent works have tried to apply transformers to some other tasks related to monocular scene reconstruction, depth prediction, *etc.* While methods have made a demonstration of how to apply a transformer to a monocular camera model [14], [20], [65], [66], [69], they all rely on other branches (either the environment reconstruction or the depth map), which will not be available in a typical Mono3D task based on RGB images.

Our MoGDE inherits DETR's superiority for non-local encoding and long-range attention. Specifically, we endow the transformer to be *ground-aware* by pinning ground depth to image features leveraging the encoder-decoder architecture to improve the detection accuracy for far objects.

VII. Conclusion

In this paper, we have proposed a Mono3D framework, called *MoGDE*, which can effectively utilize the estimated ground

depth as prior knowledge to improve Mono3D in mobile settings. The advantages of MoGDE are two-fold: 1) it can significantly improve the Mono3D accuracy, especially for far objects, which is an open issue for Mono3D; 2) it can improve the robustness of Mono3D detectors when applied in more appealing mobile applications. Nevertheless, MoGDE still has two main limitations as follows: 1) it heavily relies on pose detection, which directly affects the accuracy of the ground depth estimation; 2) it also counts on the detection of ground-contacting points. In cases when such points are uncertain or ambiguous due to occlusion and truncation, it is hard for the proposed ground-aware feature fusion method to obtain accurate results. These limitations also direct our future work. We have implemented Mono3D and conducted extensive experiments on the real-world KITTI dataset. MoGDE yields the best performance compared with the state-of-the-art methods by a large margin and is ranked number one on the KITTI 3D benchmark.

REFERENCES

[1] I. Barabanau, A. Artemov, E. Burnaev, and V. Murashkin, "Monocular 3D object detection via geometric reasoning on keypoints," May 2019, *arXiv: 1905.05618*.

[2] G. Brazil and X. Liu, "M3D-RPN: Monocular 3D region proposal network for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9286–9295.

[3] G. Brazil, G. Pons-Moll, X. Liu, and B. Schiele, "Kinematic 3D object detection in monocular video," 2020, *arXiv: 2007.09548*.

[4] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," 2019, *arXiv: 1903.11027*.

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 213–229.

[6] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, "Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2040–2049.

[7] C. K. Chang, J. Zhao, and L. Itti, "DeepVP: Deep learning for vanishing point detection on 1 million street view images," in *Proc. 2018 IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1–8.

[8] H. Chen, Y. Huang, W. Tian, Z. Gao, and L. Xiong, "MonoRUn: Monocular 3D object detection by reconstruction and uncertainty propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10379–10388.

[9] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 2147–2156.

[10] X. Chen et al., "3D object proposals for accurate object class detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 424–432.

[11] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals using stereo imagery for accurate object class detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1259–1272, May 2018.

[12] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1907–1915.

[13] Y. Chen, L. Tai, K. Sun, and M. Li, "MonoPair: Monocular 3D object detection using pairwise spatial relationships," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12093–12102.

[14] Z. Cheng, Y. Zhang, and C. Tang, "Swin-depth: Using transformers and multi-scale fusion for monocular-based depth estimation," *IEEE Sensors J.*, vol. 21, no. 23, pp. 26912–26920, Dec. 2021.

[15] M. Contributors, "MMDetection3D: OpenMMLab next-generation platform for general 3D object detection," 2020. [Online]. Available: https://github.com/open-mmlab/mmdetection3d

[16] T. V. Dijk and G. D. Croon, "How do neural networks see depth in single images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2183–2191.

[17] M. Ding et al., "Learning depth-guided convolutions for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 1000–1001.

[18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[19] T. He and S. Soatto, "Mono3D: Monocular 3D vehicle detection with two-scale 3D hypotheses and task priors," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8409–8416.

[20] K. C. Huang, T. H. Wu, H. T. Su, and W. H. Hsu, "MonoDTR: Monocular 3D object detection with depth-aware transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4012–4021.

[21] J. Jinrang, Z. Li, and Y. Shi, "MonoUNI: A unified vehicle and infrastructure-side monocular 3D object detection network with sufficient depth clues," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, Art. no. 514.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[23] A. Kundu, Y. Li, and J. M. Rehg, "3D-RCNN: Instance-level 3D object reconstruction via render-and-compare," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3559–3568.

[24] P. Li, X. Chen, and S. Shen, "Stereo R-CNN based 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7644–7652.

[25] P. Li, H. Zhao, P. Liu, and F. Cao, "RTM3D: Real-time monocular 3D detection from object keypoints for autonomous driving," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 644–660.

[26] Y. Li, Y. Chen, J. He, and Z. Zhang, "Densely constrained depth estimator for monocular 3D object detection," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, Springer, 2022, pp. 718–734.

[27] Z. Li, Z. Qu, Y. Zhou, J. Liu, H. Wang, and L. Jiang, "Diversity matters: Fully exploiting depth clues for reliable monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2791–2800.

[28] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss, Eds., Springer International Publishing, 2018, pp. 663–678.

[29] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, "Deep fitting degree scoring network for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1057–1066.

[30] Z. Liu, Z. Wu, and R. Tóth, "SMOKE: Single-stage monocular 3D object detection via keypoint estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 996–997.

[31] Z. Liu, D. Zhou, F. Lu, J. Fang, and L. Zhang, "AutoShape: Real-time shape-aware monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15641–15650.

[32] Y. Lu et al., "Geometry uncertainty projection network for monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3111–3121.

[33] X. Ma, S. Liu, Z. Xia, H. Zhang, X. Zeng, and W. Ouyang, "Rethinking pseudo-LiDAR representation," 2020, *arXiv: 2008.04582*.

[34] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, "Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6850–6859.

[35] X. Ma et al., "Delving into localization errors for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4721–4730.

[36] F. Manhardt, W. Kehl, and A. Gaidon, "ROI-10D: Monocular lifting of 2D detection to 6D pose and metric shape," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2064–2073.

[37] D. Meng et al., "Conditional DETR for fast training convergence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3651–3660.

[38] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2906–2917.

[39] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7074–7082.

[40] J. K. Murthy, G. S. Krishna, F. Chhaya, and K. M. Krishna, "Reconstructing vehicles from a single image: Shape priors for road scene understanding," in *Proc. 2017 IEEE Int. Conf. Robot. Automat.*, 2017, pp. 724–731.

[41] L. Peng et al., "LiDAR point cloud guided monocular 3D object detection," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, Springer, 2022, pp. 123–139.
[42] L. Peng, X. Wu, Z. Yang, H. Liu, and D. Cai, "DID-M3D: Decoupling instance depth for monocular 3D object detection," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, Springer, 2022, pp. 71–88.
[43] C. C. Pham and J. W. Jeon, "Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks," *Signal Process.: Image Commun.*, vol. 53, pp. 110–122, Apr. 2017.
[44] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 918–927.
[45] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A geometric reasoning network for monocular 3D object localization," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8851–8858.
[46] Z. Qin, J. Wang, and Y. Lu, "Triangulation learning network: From monocular to stereo 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7607–7615.
[47] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8555–8564.
[48] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv: 1804.02767*.
[49] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[50] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770–779.
[51] K. Shin, Y. P. Kwon, and M. Tomizuka, "RoarNet: A robust 3D object detection based on region approximation refinement," in *Proc. 2019 IEEE Intell. Veh. Symp.*, 2019, pp. 2510–2515.
[52] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1991–1999.
[53] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9627–9636.
[54] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[55] J. M. U. Vianney, S. Aich, and B. Liu, "RefinedMPL: Refined monocular PseudoLiDAR for 3D object detection in autonomous driving," 2019, *arXiv: 1911.09712*.
[56] T. Wang, J. Pang, and D. Lin, "Monocular 3D object detection with depth from motion," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 386–403.
[57] T. Wang, X. Zhu, J. Pang, and D. Lin, "FCOS3D: Fully convolutional one-stage monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 913–922.
[58] T. Wang, X. Zhu, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *Proc. Conf. Robot Learn.*, 2021, pp. 1475–1485.
[59] Y. Wang, W. L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8445–8453.
[60] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor DETR: Query design for transformer-based object detection," 2021, *arXiv:2109.07107*.
[61] X. Weng and K. Kitani, "Monocular 3D object detection with Pseudo-LiDAR point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 857–866.
[62] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," Mar. 2017, *arXiv:1604.04693*.
[63] B. Xu and Z. Chen, "Multi-level fusion based 3D object detection from monocular images," in *Proc. 2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 2345–2353.
[64] L. Yan, P. Yan, S. Xiong, X. Xiang, and Y. Tan, "MonoCD: Monocular 3D object detection with complementary depths," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 10248–10257.
[65] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci, "Transformer-based attention networks for continuous pixel-wise prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16269–16279.
[66] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci, "Transformers solve limited receptive field for monocular depth prediction," 2021, *arXiv:2103.12091*.
[67] Z. Yao, J. Ai, B. Li, and C. Zhang, "Efficient DETR: Improving end-to-end object detector with dense prior," 2021, *arXiv:2104.01318*.
[68] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2403–2412.
[69] R. Zhang et al., "MonoDETR: Depth-aware transformer for monocular 3D object detection," 2022, *arXiv:2203.13310*.
[70] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3289–3298.
[71] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," Apr. 2019, *arXiv: 1904.07850*.
[72] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.
[73] Y. Zhou, Y. He, H. Zhu, C. Wang, H. Li, and Q. Jiang, "Monocular 3D object detection: An extrinsic parameter free approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7556–7566.
[74] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv: 2010.04159*.
[75] Z. Zou et al., "The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2713–2722.

Yunsong Zhou (Member, IEEE) received the bachelor's degree in microelectronics science and engineering in 2020 from Shanghai Jiao Tong University, Shanghai, China, where he is currently working toward the PhD degree in computer science with the School of Electronic Information and Electrical Engineering. His research interests include robotics (perception, autonomous driving), embodied AI, and computer vision.

Quan Liu (Member, IEEE) received the bachelor's degree in IEEE Pilot Class and the master's degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2021 and 2024, respectively. His research interests include point cloud processing and autonomous driving.

Hongzi Zhu (Senior Member, IEEE) received the PhD degree in computer science from Shanghai Jiao Tong University, in 2009. He was a post-doctoral fellow with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology and the Department of Electrical and Computer Engineering, University of Waterloo, in 2009 and 2010, respectively. He is a professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include Internet of Things, mobile sensing and mobile computing. He received the Best Paper Award from IEEE Globecom'16.

Yunzhe Li (Member, IEEE) received the BS degree in computer science from Wuhan University, in 2021. He is currently working toward the PhD degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include mobile sensing and edge computing.

Shan Chang (Member, IEEE) received the BS degree in computer science and technology from Xi'an Jiaotong University, in 2004, and the PhD degree in computer software and theory from the Xi'an JiaoTong University, in 2013. From 2009 to 2010, she was a visiting scholar with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. She is now an associate professor with the Department of Computer Science and Technology, Donghua University, Shanghai. Her research interests include security and privacy in mobile networks and sensor networks.

Minyi Guo (Fellow, IEEE) received the PhD degree in computer science from the University of Tsukuba, Japan. He is a Zhiyuan chair professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He is currently Zhiyuan chair professor and head with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His present research interests include parallel/distributed computing, compiler optimizations, embedded systems, pervasive computing, Big Data and cloud computing. He is now on the editorial board for *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing* and *Journal of Parallel and Distributed Computing*.