Det6D: A Ground-Aware Full-Pose 3-D Object Detector for Improving Terrain Robustness

Junyuan Ouyang* and Haoyao Chen*, Member, IEEE

Abstract—Accurate 3-D object detection with LiDAR is critical for autonomous driving. All existing research studies were based on the flat-world assumption. However, the actual road can be complex with steep sections, which breaks the premise. Current methods suffer from performance degradation in this case due to difficulty correctly detecting objects on sloped terrain. This work presents the first full-degree-of-freedom 3-D object detector, Det6D, without spatial and postural limitations to improve terrain robustness. We choose the point-based framework because of its flexible detection range. A ground-aware orientation branch leveraging the local ground constraints is designed to predict full-degree poses, that is, including pitch and roll. Given the difficulty of long-tail nonflat scene data collection and 6-D pose annotation, we present Slope-Aug, a data augmentation method for synthesizing nonflat terrain from existing datasets recorded in flat scenes. Experiments on various datasets demonstrate the effectiveness and robustness of our method on different terrains. The proposed modules are plug-and-play for existing point-based frameworks. The code will be available at https://github.com/HITSZ-NRSL/De6D.

Index Terms—3-D object detection, autonomous driving, complex terrain, point cloud.

I. INTRODUCTION

IN RECENT years, LiDAR sensors have been widely used for scene understanding in robotics and autonomous vehicles. 3-D object detection as an upstream task of these applications has attracted researchers’ attention. Most of them can be divided into grid-based methods [1], [2], [3], [4], [5], [6], [7], [8] and point-based methods [9], [10], [11], [12], [13]. These methods are based on the flat-world assumption that objects are located on flat ground. Thus, current approaches can only detect objects close to the $x$-$o$-$y$-plane and their orientation along the gravity axis. Nevertheless, this assumption does not always hold in the real world, such as when driving through overpasses, underground parking lots, or other urban scenes with steep slopes. In these cases, objects have an extensive range of $z$-coordinates, and their pitch and roll can no longer be ignored. An example of failure cases is shown in Fig. 1, where the previous methods suffer from wrong object detection and incorrect pose estimation. This kind of corner case should be considered, as severe performance degradation can lead to potential traffic accidents.

In this work, we present a point-based one-stage detector to achieve robust perception in complex terrains. The proposed method named Det6D is not dependent on the flat-world assumption by fully considering the nonflat driving scene existing in the real world. It aims to improve conventional 3-D object detection, which is more like a 2.5-D detection, to the full-space and full-pose 3-D object detection.

There are three main challenges in reaching a terrain-robust detector. The first is full-space 3-D proposal generation, that is, how to detect objects at any position in space to avoid missing objects on the slope or high places. This problem has not been considered previously because the assumption of the existing methods limits the detection range to near the ground. For example, the classic voxel-based pipelines [2], [3] place anchor boxes and detect objects in bird’s-eye view (BEV), ignoring height. A naive approach is to expand anchor boxes and voxelization range along height direction, but this makes the method less efficient. The second is a full pose prediction. Incorrect orientation prediction also causes the predicted object center to shift. However, all the previous...
methods ignore the pitch and roll of objects. Note that the full pose prediction in this work is similar to 6-D pose estimation. Methods [14], [15], [16], [17], [18], [19], [20], [21], [22] in the 6-D pose estimation task based on depth information can predict full pose well, typically capturing data with an RGB-D camera and having prior computer aided diagnosis (CAD) models of objects for feature learning or result refinement. Unfortunately, these methods are unsuitable for detection due to the gaps in task domains, sensors, and real-time requirements. The third challenge is the lack of data to train the network. Existing large-scale open datasets [23], [24], [25] for autonomous driving are all recorded in flat scenes and have no 6-D pose annotation of objects. Furthermore, laboriously collecting long-tail sloped scenes and annotating 6-D poses make it harder to build a suitable dataset. Thus, the last problem to be solved is how to take advantage of a large number of existing datasets without much effort. It should be emphasized that the performance decline of previous methods is not only caused by the lack of similar nonflat scenes in the dataset, but also because of inaccurate pose estimation.

This work addresses the above challenges. We first exploit the point-based framework and anchor-free center point generation module to detect objects without spatial limitation. Then the ground-aware orientation branch assists in full pose prediction by leveraging the results from the ground segmentation module. Finally, a new data augmentation method termed Slope-Aug is proposed for non-flat scene generation and further alleviates the imbalanced distribution of pitch and roll.

The main contributions are summarized as follows.

1) To the best of our knowledge, this work is the first attempt to consider a 6-D pose prediction in 3-D object detection, which is nontrivial. To improve the performance in complex terrains, we design Det6D with a ground-aware orientation branch to lift conventional detectors from 2.5-D-limited into true 3-D.

2) The proposed data augmentation strategy, Slope-Aug, leverages the existing datasets recorded in ordinary flat scenes to train the network, avoiding long-tail scene collection and laborious 6-D pose annotations.

3) We benchmark our method against state-of-the-art methods in both flat and non-flat scene datasets. The results show that our model maintains high performance on various terrains. Furthermore, extended verification indicates that the network predicts object poses by learning the local ground constraints on poses.

II. RELATED WORKS

A. Point Cloud 3-D Object Detection

The purpose of existing point cloud 3-D object detection methods is to predict objects’ position, dimension, and heading from point clouds. According to the data representation of the unordered 3-D point cloud, the current 3-D object detector can be mainly divided into grid-based methods and point-based methods. Some works [26], [27], [28] handle the points projected to image views with highly optimized 2-D convolutional neural networks (CNNs). Other methods divide space into pillars [1] or voxels [2], [3], [4], [5], [6], [7], [29], called voxel-based methods. Zhou and Tuzel [2] used learned voxel features instead of artificial ones. Yan et al. [3] stacked sparse convolution [30] for efficient feature extraction by utilizing the sparsity of point cloud. Most voxel-based methods compressed the feature volume along the z-axis into a BEV feature map. And then, an anchor-based [2], [3], [5], [6], [29] or anchor-free [4], [7], [8] detection head is employed to predict boxes.

The point-based methods use points directly as input without any representation conversion. The pioneering work [31] learned order-invariant global features from points set via multilayer perceptron (MLP) and channel-wise max pooling. PointNet++ [32] is a hierarchical network that can learn local and global features from points. As a fundamental module in point-based detectors, it is widely used to extract semantic features in various networks [9], [10], [11]. Shi et al. [11] first proposed a bottom-up proposal generator to get proposals from points directly. Yang et al. [13] remove the feature propagation layer to reduce the time consumption. Chen et al. [33] introduce learnable parameters in the downsampling layer to retain more foreground points adaptively.

B. Object Pose Estimation from Depth Information

These methods [14], [15], [16], [17], [18], [19], [20], [21], [22] aim to recover the position and orientation of objects from depth information captured by RGB-D cameras. Since CAD models of objects are known, there is no need to predict the dimension of objects. Guo et al. [14] proposed a method based on point pair features (PPFs) to estimate object 6-D poses from points in the depth map. Gao et al. [16], [19] utilized two network branches to estimate position and orientation and proposed a novel geodesic loss function for more accurate rotation regression. Other voting-based [17] and keypoint-based [15] methods fed the point clouds into a PointNet-like network to predict key points. They recovered 6-D poses by estimating the transformation between predicted keypoint sets and predefined point sets. Most of these methods perform iterated closest point (ICP) based on the known CAD model to refine the estimation results further.

In the detection task, the existing methods only consider predicting the horizontal orientation, that is, yaw, from a point cloud. Zhou and Tuzel [2] directly regress the residual between the proposal and its corresponding anchor box. Zhou and Tuzel [2] encode the yaw into sin–cos code for better regression. Other works [9], [11], [13] divide the yaw into several bins, then predict which bin it belonged to and the residual in the bind.

C. Limitations of the Existing Methods

The existing detection methods assume the world to be flat, resulting in only detecting objects near the x-o-y-plane and only predicting yaw. Projection-based methods have no detection range limitation but lower accuracy. While expanding the detection range in voxel-based methods will lead to less efficiency. The 6-D pose estimation methods based on depth information cannot be used to predict full pose in the detection task. These methods utilize the dense and small-scale depth data from the RGB-D camera. While in the detection
task, the point cloud from LiDAR is nonuniform, sparse, and large-scale. Moreover, in the 6-D pose estimation task, networks can only handle an instance point cloud obtained by image segmentation at a time and involve time-consuming postprocesses. Thus, multiple objects need to be processed sequentially. However, the detection task needs to process multiple objects quickly in parallel. Furthermore, these 6-D pose estimates methods get accurate predictions because the CAD models are known, which are not available in the detection task.

### III. Method

This section first analyzes the method selection, introduces our network design, and then describes the differences between yaw, pitch, and roll in pose prediction. The overall pipeline is illustrated in Fig. 2. It consists of three main parts: 1) a center point generator for full-space proposal generation; 2) a multitask detection head with the ground-aware branch for full orientation prediction; and 3) an augmentation method Slope-Aug for synthesizing nonflat scenes.

#### A. Problem Definition and Method Analysis

We denote \( P = \{p_i\}_{i=1}^{n} \) as the input point clouds, where each point \( p_i = (x_i, y_i, z_i, \ldots) \in \mathbb{R}^{3+n} \) consists of three coordinates and optional features with \( c_0 \) initial channels. The existing methods parameterize the bounding box of the detected object with \( b_{2.5-D} = (x_c, y_c, z_c, l, w, h, \theta_z) \), which stands for the position, dimension, and yaw, respectively. Nevertheless, \( z_c \) of \( b_{2.5-D} \) is limited to a small value, that is, near the ground. We denote the full-pose bounding box by \( b_{3-D} = (x_c, y_c, z_c, l, w, h, \theta_x, \theta_y, \theta_z) \) to describe the object in nonflat scenes, where \( z_c \) is unconstrained and \( (\theta_x, \theta_y, \theta_z) \) are Euler angles in the \( x-y-z \) order. This work aims to detect objects in \( P \) represented by \( b_{3-D} \) instead of \( b_{2.5-D} \) used in the previous works.

1) Method Analysis: We explore what limits current methods to detect objects with full pose in the entire space. In voxel-based methods, the voxelization range of the \( z \)-axis tends to be no more than 3 m, while the height of target objects in nonflat scenes can be up to 10 m, causing points outside the range to be ignored. A simple approach is to expand the \( z \)-axis range, but this greatly increases the time and memory consumption. In addition, since nonflat scenes are corner scenes, such a large \( z \) range is meaningless in most cases and results in lots of empty voxels. Although sparse convolution alleviates this problem, the sparse features need to be converted into a dense presentation and compressed along the \( z \)-axis to obtain BEV feature maps before dense detection. This causes the dimension of BEV features to grow and introduces many zero values in features. Anchor boxes with different categories, sizes, and orientations need to be placed at each location in the BEV maps for anchor-based methods. Even for flat scenes, regardless of pitch, roll, and \( z \), the number of anchors can reach a million. Thus, anchor-based methods are not feasible for our task.

#### B. Feature Encoding and Center Point Generation

Based on the above analysis, for efficient feature encoding and full-space full-pose object detection in nonflat scenes, we design a point-based backbone and an anchor-free proposal generator similar to the previous works [13], [33].

1) Point Backbone: The backbone only remains the encoder part of the U-shaped network [32] to reduce overhead. The input points \( P \) are gradually aggregated into a small set of representative points \( P_r = \{(x_i, y_i, z_i)\}_{i=1}^{n_r} \), each with semantic feature \( f_r \in \mathbb{R}^{c_1} \) by multiple furthest point sampling layers and multiscale grouping set abstraction layers [32]. The set abstraction layer first groups sampled points and then extracts features at each grouping center by PointNet [31] as

\[
  f = \gamma \left( \text{MAX} \left( h(x_i) \right) \right)
\]

where \( f \) is the extracted feature from grouped points \( \{x_i\}_{i=1}^{n_r} \) and \( \gamma(\cdot) \) and \( h(\cdot) \) are MLPs, respectively. The distance metric of furthest point sampling is weighted by feature distance, foreground score, and local point density to preserve more foreground points in downsampling.

---

**Fig. 2.** Overall framework of the proposed Det6D. In the training phase, the raw point cloud is randomly synthesized as a sloped scene via Slope-Aug to leverage the existing dataset recorded in the flat scene, while it is fed directly into the network in the test phase. The point cloud backbone downsamples the input and extracts multiscale semantic features. These semantic features are used to generate a little set of coarse center points of objects. And then, the center points and their features are segmented into different classes by a ground segmentation module. Finally, the detection head combines (detailed in Fig. 4) the ground segmentation results and the features of center points to predict boxes with full poses.
b) Distribution of object dimension.

Fig. 3. Distribution of object attributes in the SlopedKITTI dataset. (a) Distribution of object centers in BEV. (b) Distribution of object dimension. (c) Distribution of object poses (log-10 scale).

2) Center Point Generator: The offset from each point in \( \mathcal{P}_t \) to the center of its corresponding object is predicted and used to obtain coarse centers \( \mathcal{P}_{cc} \) (the golden balls shown in Fig. 2) for further refinement. A set abstraction layer is also applied on \( \mathcal{P}_{cc} \) to extract the feature \( f_{cc} \in \mathbb{R}^{c^2} \) of each coarse center.

C. Box Prediction via Full Pose Detection Head

Our detection head consists of multiple branches to predict the attributes of objects corresponding to coarse centers \( \mathcal{P}_{cc} \) with their features \( F_{cc} = \{ f_{cc} \} \). Concretely, these branches predict the class label \( c \), offset to fine center \((\Delta x, \Delta y, \Delta z)\), dimensions \((l, w, h)\), and orientations \((\theta_x, \theta_y, \theta_z)\) of objects. Finally, apply nonmaximum suppression (NMS) to remove redundancy.

1) Ground-Aware Orientation Head: Unlike the existing methods, our orientation head predicts both the yaw \( \theta_z \) and the two previously ignored poses, that is, pitch \( \theta_y \) and roll \( \theta_x \). It is worth noting that this extension is nontrivial. Since the data distribution difference, copying the method for yaw to these two poses does not work. Fig. 3 shows the distribution of object attributes in SlopedKITTI containing 6-D pose annotations for better explanation. Both \((l, w, h)\) and \(\theta_z\) approximately obey the mixed Gaussian distribution, making them easy to be regressed. However, despite this dataset containing up to 10% of nonflat scenes, higher than most cases, the points with zero \( \theta_z \) and \( \theta_y \) are still 3–4 orders of magnitude more than the nonzero ones. If we directly predict \( \theta_x \) and \( \theta_y \) like \( \theta_z \), the results tend to be zero since the highly imbalanced distribution leads to nonzero values being ignored like noise. Furthermore, it is intuitive from the BEV that the shape of points varies with noise. Furthermore, it is intuitive from the BEV that the shape of points varies with \( \theta_x \) and \( \theta_y \), making the network hard to capture the explicit relationship between shape and these two poses. The above reasons explain why the prediction methods for different poses are not the same.

A novel view is that objects in autonomous driving scenes all lie on the ground, which well constrains \( \theta_x \) and \( \theta_y \). Thus, awareness of the ground can help the regression of these two poses. With this prior, we introduce a lightweight ground segmentation module composed of only several full connection layers. It classifies the terrain of each point located into two categories: flat and sloped, as represented by blue and red balls in Fig. 2, respectively. Specifically, we feed the feature \( f_{cc} \) of each coarse center in \( \mathcal{P}_{cc} \) to the segmentation module and predict the probability \( s_g \) of being on sloped terrain via

\[
s_g = \text{Sigmoid}(\text{MLPs}(f_{cc})).
\]  

By doing this, the unbalanced regression problem is decomposed into two well-addressed problems: unbalanced classification and balanced regression. We use focal loss [34] with the default setting to handle this imbalanced classification since there are fewer points belonging to sloped terrain. And smooth-L1 loss is applied to supervise the pose predictions \( \theta_x \) and \( \theta_y \) for points on sloped terrain only. We denote \( b_{3-D} = (x^g, y^g, z^g, l^g, w^g, h^g, \theta_x^g, \theta_y^g, \theta_z^g) \) as a ground-truth box. The ground segmentation label \( c_g \) and the pose regression targets \( \theta_x^t \) and \( \theta_y^t \) of each coarse center can be generated naturally from ground-truth bounding boxes by

\[
c_g = \mathbb{1}(\theta_x^g \geq \theta_x^t \text{ or } \theta_y^g \geq \theta_y^t),
\]

\[
\theta_x^t = \frac{\theta_x^g - \theta_x^0}{\pi/2}, \quad \theta_y^t = \frac{\theta_y^g - \theta_y^0}{\pi/2}
\]  

where \( \mathbb{1} \) is the indicator function, and \( \theta_x^0 \) and \( \theta_y^0 \) are the thresholds used to determine the terrain category. The orientation branch then utilizes the segmentation results to assist the network in recognizing the latent relationship between pose and terrain, as detailed in Fig. 4. Specifically, we obtain the final prediction \( \theta^c \) and \( \theta^t \) by associating with \( s_g \), as

\[
\theta^c = \begin{cases} 
\theta^0, & \text{if } s_g > 0.5 \\
0, & \text{otherwise}
\end{cases} \quad \text{for } a \in \{x, y\}
\]  

where \( \theta^0 \) is decoded from branch predicted output \( \hat{\theta}_a \) via (3).

As for heading \( \theta_z \), the Decoder 2 branch in the right of Fig. 4 details this part. We first divide \( \theta_z \) into \( N_{\theta_z} \) discrete bins. A cross-entropy loss and smooth-L1 loss are then employed to classify the bin and predict its residual, respectively. The bin label \( \theta_z^b \) and the residual in a bin \( \Delta \theta_z^t \) for heading prediction are calculated as in the previous works [11] by

\[
c_{\hat{\theta}_z} = \left\lfloor \frac{\theta_z^t}{\Delta \theta_z^t} \right\rfloor
\]

\[
\Delta \theta_z^t = \frac{\theta_z - c_{\hat{\theta}_z} \Delta \theta_z + \hat{\theta}_z}{\Delta \theta_z}
\]  

where \( \Delta \theta_z = 2\pi/N_{\theta_z} \) represents the size of each bin in radian and \( \lfloor \cdot \rfloor \) means floor operation to get an integer value.

2) Position Head: The offset \( \Delta \mathcal{P}_{cc} \) of each coarse center in \( \mathcal{P}_{cc} \) to its actual center is further predicted via a smooth-L1 loss to refine object localization.

3) Dimension Head: We also regress the dimension of objects. The dimensions are log-mapped, and the results are used as the regression target \( u' = \log u \), where \( u \in \{l, w, h\} \).

4) Classification Head: Benefiting from the downsampling strategy [33], the categories distribution of coarse centers is more balanced. Thus, we directly apply cross-entropy to calculate the classification loss from prediction \( \hat{c} \) and label \( c \).
The part containing the origin is denoted as \( \tau \) vector \( P \) of \( \theta \) and \( \theta \) orientation branch. The ground segmentation module predicts which classes of terrains the coarse centers are located via ground-truth boxes supervising. The coarse centers are located via ground-truth boxes supervising. The orientation branch predicts bin index \( \hat{\theta}_p \) and \( \hat{\theta}_t \) by (3), and \( \hat{\theta}_p \) and \( \hat{\theta}_t \) by bin-based methods (5), respectively. Additionally, the ground segmentation scores are utilized to postprocess \( \theta \) and \( \theta \) via (4). Finally, these three predicted orientations are concatenated and treated as output.

5) Loss: The overall training loss of bounding boxes \( \{b_{1-D}\} \) consists of different branch loss terms, as

\[
L_{\theta,\gamma} = \frac{1}{N_{p}} \sum \mathbb{1}(c > 0) \mathcal{L}_{\text{seg}}(s_{x}, c_{g}) + \frac{1}{N_{p}} \sum \mathbb{1}(c > 0) \mathcal{L}_{\text{seg}}(\theta_{a}, \theta_{a}^{0})
\]

\[
L_{\theta} = \sum \frac{1}{N_{p}} \left[ \mathcal{L}_{\text{ch}}(c_{0}, c_{a}) + \mathcal{L}_{\text{seg}}(\Delta \theta_{s}, \Delta \theta_{t}) \right]
\]

\[
L_{\text{box}} = L_{\text{ch}} + L_{\text{dim}} + L_{\text{pos}} + L_{\theta} + L_{\theta}
\]

where the normalization factor \( N_{p} \) is the number of foreground points in \( P_{cc} \), and \( N_{p} \) is the number of foreground points in \( P_{cc} \) that lie on sloped terrain; \( \mathbb{1}(c_{x} > 0) \) indicates that only foreground points on sloped terrain contribute to the loss item; the smooth-L1 is used for the regression item \( \mathcal{L}_{\text{reg}} \); \( \mathcal{L}_{\text{seg}} \) is focal loss for terrain segmentation and \( \mathcal{L}_{\text{ch}} \) is cross-entropy.

D. Slope Augmentation to Help Full Pose Learning

The existing open datasets [23], [24], [25] contain only flat scenes and lack the required \( \{(\theta_{s}, \theta_{t})\} \) annotations. Moreover, it is laborious to collect data from nonflat scenes and manually annotate 3-D bounding boxes with full poses in 3-D space. Therefore, Slope-Aug is proposed to address this problem by synthesizing random slope scenes in the input point clouds and generating full pose annotations during the training phase.

Specifically, we perform the steps shown in Fig. 5 with an enabled probability \( p_i \) for each input. We first select an anchor point \( \tau = (r, \alpha, 0) \) in cylindrical coordinates with the distance of \( r \) and direction of \( \alpha \) to the origin. The tangent direction of vector \( \tau \) is the desired rotation axis \( \psi \). The input point cloud \( P \) is divided into two parts by the anchor \( \tau \) and the axis \( \psi \). The part containing the origin is denoted as \( P_1 \), and the other part \( P_2 \) can be formulated as

\[
P_2 = \left\{ p_i \mid r^{T}(\tau - p_i) < 0, \forall p_i \in P \right\}.
\]

Part \( P_2 \) is rotated by the axis angle \( (\psi, \gamma) \) at \( \tau \) to simulate the points on slope \( P_2 \), and the scene with a pseudo-slope \( P_2 = P_1 \cap P_2 \) is obtained. Full-pose bounding box annotations \( B_{3-D} \) are generated simultaneously from \( \{b_{2,5-D}\} \) as

\[
B_{3-D} = \left\{ b_{i,7} \mid b_{i} \in \{b_{2,5-D}\}, c_{i} = r^{T}(\tau - b_{i,3}) < 0 \right\}
\]

where the second subscript of \( b_{i,11} \) means the slice operation in vector \( b_{i} \), that is, to obtain elements of a vector.

In this way, we can leverage \( P \) and \( \{b_{2,5-D}\} \) from the existing datasets to synthesize slope scene \( P_2 \) and generate full-pose box annotations \( B_{3-D} \). Despite the domain gaps between synthetic and natural data, the following experiments show that the models trained with these pseudo-slope scenes can be generalized to actual scenes. Moreover, the highly unbalanced distribution of yaw and roll is alleviated by this augmentation.

IV. EXPERIMENTS

This section conducts extensive experiments to verify the effectiveness of our framework. The benchmark results indicated that the proposed Det6D outperforms all other methods in nonflat scenes. We further reveal the decisive role the ground plays in pose prediction by simulation. Summarily, our method achieves a more robust perception in complex terrains.

A. Dataset

1) KITTI: The competitive 3-D object detection dataset KITTI [23] is used to evaluate the performance of Det6D in flat scenes. It contains 7481 training and 7518 test samples collected with \( \{b_{2,5-D}\} \) annotated. The training set is divided into train split and val split with 3712 and 3769 samples, respectively. We refer to strategy [29] to generate the train split for submission to the KITTI test server.
2) *SlopedKITTI*: Since the existing open datasets [23], [24], [25] do not contain nonflat scenes and 6-D-pose box annotations, the SlopedKITTI dataset is built based on KITTI val split by synthesizing pseudo-slope as described in Section III-D. It contains sloped terrains and 6-D-pose box annotations and is used to benchmark the performance of our method against others in nonflat scenes to show its robustness.

3) *GTA-V and GAZEBO*: Although SlopedKITTI contains nonflat urban scenes, it is similar to the original KITTI with Slope-Aug applied. We also test Det6D on two different environments for fair comparison and to demonstrate the generalization performance. GTA-V is a commercial video game with realistic urban scenes in which we collect real nonflat scenes. GAZEBO is a common robot simulation software used for additional experiments and verification. It is worth noting that the point clouds in these two environments are very similar to those in the real world, as the geometric information is easier to simulate than the rich textured images.

B. Metrics

The same as official KITTI, all results are evaluated by AP, which is calculated using 11 recall positions on val split and 40 positions on test set. Regarding vehicle detection, we use an intersection over union (IoU) of 0.7 to determine the true positive (TP). Rotated BEV IoU and rotated BEV 3-D IoU are used for BEV and 3-D detection, respectively.

Since the IoU used above is calculated based on BEV, the relationship between the two full-pose bounding boxes in 3-D space cannot be correctly described. Inspired by NDS [24], we provide an additional rotated 3-D metric (R3DM) to reflect the performance in 3-D space accurately. It replaces the IoU threshold of 0.7 with the center distance of 1.0 m, and the corresponding average precision (AP) is denoted as APcd.

For TPs, we additionally compute the average translation score (ATS), average scale score (ASS), and average full orientation score (AOS) from their corresponding error item. The composite rotated object detection score (RODS) is computed by

\[ \text{RODS} = \frac{1}{6} (3 \text{AP}_{cd} + \text{ATS} + \text{ASS} + \text{AOS}) \]  

where factor 1/6 normalizes the sum of weights to be 1.

C. Implementation Details

Our Det6D and other compared models are implemented based on the OpenPCDet [37] toolbox and run on the same GTX2080Ti GPU for a fair comparison.

1) Network Architecture: We first randomly sample 16384 points for each raw point cloud and shuffle them as input. In the backbone, we gradually sample (4096, 1024, 512) points using three downsampling layers. The output feature channels of each SA layer are 64, 128, and 256, respectively.

The detection head contains the backbone, we gradually sample (4096, 1024, 512) points using three downsampling layers. The output feature channels of each SA layer are 64, 128, and 256, respectively.

The detection head contains the backbone, we gradually sample (4096, 1024, 512) points using three downsampling layers. The output feature channels of each SA layer are 64, 128, and 256, respectively.

During the testing phase, we use the IoU threshold of 0.1 for NMS.

2) Training: The model was trained on KITTI train split for 7 h using a batch size of 4 and a one-cycle turning strategy with a 0.1 learning rate. We first use GT-Aug [3] to randomly copy some foreground objects from other scenes into the current scene and then use the proposed Slope-Aug with probability \( p_v = 0.1 \). Other strategies have also been applied, such as global flipping, scaling, and rotation. Except for testing on the test set, models for all experiments were trained on train split and then validated on val split.

D. Main Results

We quantitatively compare the proposed method with state-of-the-art methods [1], [3], [4], [5], [11], [13], [29], [33], [35], [36] to evaluate the performance in nonflat scenes. All methods...
are trained on the same KITTI train split and validated on the SlopedKITTI val split. Some results are visualized in Fig. 6. It can be found that the pitch and roll of detected objects can be different in the proposed method since our approach does not need to assume there is only one ground plane in the scene as in the existing works. As shown in Table I, the proposed method outperforms all the previous methods by a large margin on all metrics in sloped scenes. In terms of 3-D object detection, Det6D dramatically improves the easy, moderate, and hard APs by 25.76%, 34.0%, and 33.26%, respectively. For the most concerned rotated 3-D object detection, we lead by 12.76%, 1.17%, 0.04%, 1.36%, and 11.54% in AP cd, each TP score and RODS, respectively.

It can be found in Table I that point-based methods outperform voxel-based methods in BEV and AP cd because of the capability of detecting objects anywhere the point exists, that is, no detection range limitation. While in terms of TP scores, the previous point-based methods perform worse than voxel-based methods. It is mainly because all TPs on sloped terrains are detected with incorrect poses by the previous point-based methods. Thus, except for ours, the point-based methods fall behind voxel-based methods in each TP score. Nevertheless, all these methods perform poorly in 3-D AP since full pose prediction is not considered, resulting in low IoU. In addition, it is worth noting that the previous point-based methods detected lots of false positives on the slope because the scan lines from slopes are similar to those on the side of vehicles. Our method can accurately predict all poses besides correctly detecting objects on slopes and some unlabeled vehicles.

We also report results on KITTI as shown in Table II to evaluate performance in flat scenes. The results show that our method performs comparably to other point-based methods in flat scenes. Comparing the results in Tables I and II shows that existing methods have a significant performance drop in nonflat scenes. We further report the results on SlopedKITTI with different slopes in Fig. 7 to explore the effect of terrain on detector performance. In the scene with a 30° slope, the existing methods drop dramatically by up to 45% AP, with less than 40% AP remaining. Such poor performance is unacceptable for real-world applications. In contrast, our method keeps a higher performance of around 70% AP, with merely a relatively slight reduction of 15% AP. That is, the degradation of the existing methods is nearly 300% of ours. Although there is still a 15% performance drop, mainly due to the sensitivity of AP to slight pitch and roll error, our method is sufficient to handle complex terrain, as shown in Figs. 6(e) and 8(e).

It can be found that the performance improvement is more pronounced at a slope of 10° than at −10°, which may be related to the unbalanced distribution of the regression labels for positive and negative degrees. Overall, we correctly detect objects with full pose in nonflat scenes, improving perception robustness.
The inference speed of all models is measured in milliseconds on a single RTX2080Ti for a fair comparison. As shown in the last column of Table I, the proposed method can detect in real-time with a latency of 32 ms, similar to most state-of-the-art point-based methods. Overall, our approach achieved a better balance between performance and efficiency.

E. Effectiveness

We test these methods on GTA-V and GAZEBO, containing actual sloped terrain, to verify the effectiveness of the proposed method. All models are trained on KITTI, and some results are visualized in Fig. 8. Consistent with the previous conclusions, voxel-based methods cannot detect objects on slopes, and the previous point-based methods cannot accurately detect objects on slopes. In contrast, the proposed method can detect objects correctly and predict full poses accurately. It proves that the model trained by Slope-Aug can effectively handle real slopes.

However, for sparse distant point clouds, the ground segmentation module may fail and make pose predictions wrong, as the two rightmost cars shown in the last row of Fig. 8(e). The accuracy of pose estimation is part of the detection performance, and inaccurate ground segmentation can affect the detection quality through inaccurate pose estimation. In this case, the proposed method, like the existing ones, cannot correctly estimate the pose of the object due to the insufficient surrounding point clouds.

1) What Did the Network Learn?: We designed an extended verification in GAZEBO to verify our conjecture that the predictions of pitch and roll depend on the ground while yaw depends on the object itself. When the vehicle is on a slope [Fig. 9(a)], regardless of the actual poses, the predicted pose is consistent with the local ground normal. When the vehicle is on flat ground [Fig. 9(b)], the predicted poses vary with the local terrain. The above experimental results demonstrate that the proposed ground-aware orientation branch accurately predicts pitch and roll with an awareness of the ground, which is inherently different from yaw prediction.

F. Ablation Studies

The result in Table III shows how each module affects the performance. The first two rows show that Slope-Aug improves detection performance in nonflat scenes significantly. D.R. and G.O.B. are different methods for predicting pitch and roll. The second and third rows show that direct regression has some effects. Pitch and roll errors decrease while yaw errors increase. Since the vehicle is usually a cuboid, yaw has a much more impact on IoU than pitch and roll. That is why the third row has 10.70% less 3-D AP than the second row. The last two rows indicate that the proposed ground-aware pose head can better predict the full poses than direct regression. In all configurations, the last row achieves the best performance, reflecting the effectiveness of the proposed framework.

Table III: Ablation Study of Proposed Framework on SlopedKITTI Validation Split. “D.R.” “G.O.B.”, and “S.A.” Mean Direct Regression, Ground-Aware Orientation Branch, and Slope-Aug, Respectively.

| G.O.B. | D.R. | S.A. | 3D AP_R@0.7 Moderate | R3Dm_R@0.7 Moderate | Mean Error (rad) Yaw | Pitch(Roll) |
|--------|------|------|----------------------|---------------------|----------------------|--------------|
| -      | -    | -    | 36.34                | 69.01               | 0.15                 | 0.33         |
| -      | √    | -    | 70.27                | 65.56               | 0.15                 | 0.34         |
| √      | √    | -    | 59.57                | 72.09               | 0.20                 | 0.15         |
| √      | √    | √    | 73.55                | 84.36               | 0.12                 | 0.05         |

V. Conclusion

This work presents the first full-degree-of-freedom 3-D object detector to improve the terrain-robustness in widespread robotic applications. We uncover that the ground well constraints the pitch and roll of objects on it, essential for predicting both poses. Thus, the ground-aware orientation branch, which benefited from the ground segmentation module, is designed to regress the full-degree poses by leveraging this prior. The Slope-Aug data augmentation method is further proposed to alleviate extreme imbalances in pose distribution, making it possible to train the network using the existing flat scene datasets. We conducted extensive experiments to
evaluate our framework. With comparable performance in flat terrain, the proposed method outperforms all the previous methods based on the flat-world assumption by a large margin of 34% AP in the nonflat scene. The exploratory verification confirms our view that ground constraints are critical for pose prediction. Future work will focus on addressing the efficiency of voxel-based methods and exploring general detection frameworks in 3-D space.

REFERENCES

[1] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “PointPillars: Fast encoders for object detection from point clouds,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12697–12705.
[2] Y. Zhou and O. Tuzel, “VoxelNet: End-to-end learning for point cloud based 3D object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4490–4499.
[3] Y. Yan, Y. Mao, and B. Li, “SECOND: Sparcely embedded convolutional detection,” Sensors, vol. 18, no. 10, p. 3337, 2018.
[4] T. Yin, X. Zhou, and P. Krähenbühl, “Center-based 3D object detection and tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 11784–11793.
[5] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, “Voxel R-CNN: Towards high performance voxel-based 3D object detection,” in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 2, 2021, pp. 1201–1209.
[6] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, “CIA-SSD: Confident iou-aware single-stage object detector from point cloud,” in Proc. AAAI Conf. Artif. Intell., 2021, vol. 35, no. 4, pp. 3555–3562.
[7] R. Ge et al., “AFDet: Anchor free one stage 3D object detection,” 2020, arXiv:2006.12671.
[8] Y. Hu et al., “AFDetV2: Rethinking the necessity of the second stage for object detection from point clouds,” in Proc. AAAI Conf. Artif. Intell., 2022, vol. 36, no. 1, pp. 969–979.
[9] 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/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 918–927.
[10] C. R. Qi, O. Litany, K. He, and L. Guibas, “Deep Hough voting for 3D object detection in point clouds,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9277–9286.
[11] S. Shi, X. Wang, and H. Li, “PointRCNN: 3D object proposal generation and detection from point cloud,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 770–779.
[12] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, “STD: Sparse-to-dense 3D object detector for point cloud,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1951–1960.
[13] Z. Yang, Y. Sun, S. Liu, and J. Jia, “3DSSD: Point-based 3D single stage object detector,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11040–11048.
[14] J. Guo et al., “Efficient center voting for object detection and 6D pose estimation in 3D cloud data,” IEEE Trans. Image Process., vol. 30, pp. 5072–5084, 2021.
[15] W. Chen, J. Duan, H. Basevi, H. J. Chang, and A. Leonardis, “Point-PoseNet: Point pose network for robust 6D object pose estimation,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2020, pp. 2824–2833.
[16] G. Gao, M. Lauri, Y. Wang, X. Hu, J. Zhang, and S. Frintrop, “6D object pose regression via supervised learning on point clouds,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2020, pp. 3643–3649.
[17] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, “PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11632–11641.
[18] F. Hagelskjaer and A. G. Buch, “PointVotenet: Accurate object detection and 6 DOF pose estimation in point clouds,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Oct. 2020, pp. 2641–2645.
[19] G. Gao, M. Lauri, X. Hu, J. Zhang, and S. Frintrop, “CloudAAE: Learning 6D object pose regression with on-line data synthesis on point clouds,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2021, pp. 11081–11087.
[20] Y. Zhang, Y. Liu, Q. Wu, J. Zhou, X. Gong, and J. Wang, “EANet: Edge-attention 6D pose estimation network for texture-less objects,” IEEE Trans. Instrum. Meas., vol. 71, pp. 1–13, 2022.

Junyuan Ouyang
Haoyao Chen

Junyuan Ouyang received the B.Eng. degree in automation from the Harbin Institute of Technology, Shenzhen, China, in 2021, where he is currently pursuing the master’s with the School of Mechanical Engineering and Automation. His research interests include computer vision and mobile robots.

Haoyao Chen (Member, IEEE) received the B.Eng. degree in mechatronics and automation from the University of Science and Technology of China, Hefei, China, in 2004, and the Ph.D. degree in robotics and automation from the University of Science and Technology of China and the City University of Hong Kong, Hong Kong, in 2009. He is currently a Professor with the Harbin Institute of Technology, Shenzhen, China, and the State Key Laboratory of Robotics and System, Heilongjiang, China. His research interests include visual servoing, multirobot systems, motion control, and aerial manipulation.