Probabilistic and Geometric Depth: Detecting Objects in Perspective

Tai Wang¹,² Xinge Zhu¹ Jiangmiao Pang¹,²* Dahua Lin¹,²,³
¹CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
²Shanghai AI Laboratory ³Centre of Perceptual and Interactive Intelligence
{wt019, zx018, dhlin}@ie.cuhk.edu.hk, pangjiangmiao@gmail.com

Abstract: 3D object detection is an important capability needed in various practical applications such as driver assistance systems. Monocular 3D detection, as a representative general setting among image-based approaches, provides a more economical solution than conventional settings relying on LiDARs but still yields unsatisfactory results. This paper first presents a systematic study on this problem. We observe that the current monocular 3D detection can be simplified as an instance depth estimation problem: The inaccurate instance depth blocks all the other 3D attribute predictions from improving the overall detection performance. Moreover, recent methods directly estimate the depth based on isolated instances or pixels while ignoring the geometric relations across different objects. To this end, we construct geometric relation graphs across predicted objects and use the graph to facilitate depth estimation. As the preliminary depth estimation of each instance is usually inaccurate in this ill-posed setting, we incorporate a probabilistic representation to capture the uncertainty. It provides an important indicator to identify confident predictions and further guide the depth propagation. Despite the simplicity of the basic idea, our method, PGD, obtains significant improvements on KITTI and nuScenes benchmarks, achieving 1st place out of all monocular vision-only methods while still maintaining real-time efficiency. Code and models will be released at https://github.com/open-mmlab/mmdetection3d.

Keywords: Probabilistic and Geometric Depth, Monocular 3D Detection

1 Introduction

3D object detection is an essential task for many robotic systems such as autonomous vehicles. Recent advanced methods in this field typically resort to various sensors, such as LiDAR [1, 2, 3, 4, 5, 6], Radar [7], binocular vision [8, 9], or their combinations for accurate depth information. Nevertheless, these perceptual systems are complicated, expensive, and difficult to maintain in complex environments. In contrast, monocular 3D detection, a setting that aims at perceiving 3D objects from 2D monocular images, has drawn increasing attention due to its low costs. However, as the depth information is not directly manifest in the input, this task is inherently ill-posed, making the problem particularly challenging.

This paper starts from a systematic study about this problem on two authoritative benchmarks in a quantitative way. Although we already knew the depth information is critical to this task, the study surprisingly shows that inaccurate depth estimation blocks all the other localization predictions from improving the final results. As instance depth has shown to be the bottleneck, we can simplify monocular 3D detection as an instance depth estimation problem to tackle it essentially.

Previous methods [10, 11] first use an extra cumbersome depth estimation model to complement 2D detectors on depth information. The following methods [12, 13, 14] simplify the frameworks by directly regarding depth as one dimension of the 3D localization task. However, they still use simple methods that estimate depth from isolated instances or pixels in a regression manner. We observe that aside from each object itself, other objects are co-existing in an image and the geometric relations across them can be valuable constraints to guarantee accurate estimation.

Motivated by these observations, we propose Probabilistic and Geometric Depth (PGD) that jointly leverages probabilistic depth uncertainty and geometric relationships across co-existed objects for accurate depth estimation. Specifically, as the preliminary depth estimation of each instance is usually inaccurate in this ill-posed setting, we incorporate a probabilistic representation to capture the uncertainty of the estimated depth. We first bucket the depth values into a set of intervals and calculate the depth by the expectation of the distribution (Fig. 1(a)). The average of top-k confidence scores from the distribution is taken as the uncertainty of the depth. To model the geometric relations,
In this paper, to tackle the dominating depth estimation problem in the monocular 3D detection, we first (a) predict the depth of each instance with a probabilistic representation to capture the uncertainty, and (b) further construct a geometric relation graph to enhance the estimation from contextual connections. (c) The proposed method outperforms the other work in terms of both performance and speed significantly on the KITTI 3D car detection benchmark.

We further construct a depth propagation graph to enhance the estimations with their contextual relationship. The uncertainty of each instance depth provides useful guidance for the propagation therein. Benefiting from this overall scheme, we can easily identify the predictions with higher confidence, and more importantly, estimate their depths more accurately with the graph-based synergistic mechanism.

We implement the methods on a simple monocular 3D object detector FCOS3D [15]. Despite the simplicity of the basic idea, our PGD results in significant improvements on KITTI [16] and nuScenes [17] with different benchmark settings and evaluation metrics. It achieves 1st place out of all monocular vision-only methods while still maintaining real-time efficiency. The simple yet effective method proves that with only designs tailored to depth, a 2D detector can be capable of detecting objects in perspective.

2 Related Work

2D Object Detection

According to the base of initial guesses, modern 2D detection methods can be divided into two branches, anchor-based and anchor-free. Anchor-based methods [18, 19, 20, 21] benefit from the predefined anchors in terms of much easier regression, while anchor-free methods [22, 23, 24, 25, 26] do not need complicated prior settings and thus have better universality. For simplicity, we take FCOS3D [15], the 3D adapted version of FCOS [24], as the baseline considering its capability of handling overlapped ground truths and scale variance problem.

Monocular Depth Estimation

Monocular depth estimation is also a challenging ill-posed problem like monocular 3D detection. It aims at predicting dense and global depth field at pixel level given an RGB image. Early works [27, 28] predict depth from hand-crafted features with non-parametric optimization methods. With the rapid progress of CNNs, fully supervised methods [29, 30, 31], self-supervised methods based on stereo pairs [32, 33] and monocular videos [34, 35] gradually emerged. Although this problem has been explored for a long time, there are very few works [36] studying it in a specific task, like detection, where the dense depth supervision is always not guaranteed and we only care about the accuracy of instance depth instead of the global depth field.

As for the reformulation of depth learning problems, there are a few attempts in this field. For example, DORN [37] recasts the depth learning problem as ordinal regression and proposes a spacing-increasing discretization (SID) strategy to improve network training and reduce computations. It is similar to the underlying idea of our probabilistic representation for uncertainty modeling while different in terms of motivation and design details.

Monocular 3D Object Detection

Monocular 3D detection is more complicated than the 2D case. The underlying problem is the inconsistency of input 2D data modal and the output 3D predictions. Methods involving sub-networks Earlier work uses sub-networks to assist 3D detection. 3DOP [38] and MLFusion [10] use a depth estimation network while Deep3DBox [39] uses a 2D object detector. They rely on the design and performance of these sub-networks, even external data and pre-trained models, which makes the training inconvenient and introduces additional system complexity.

Transform to 3D representation Another category is to convert the RGB input to 3D representations like OFTNet [40] and Pseudo-Lidar [11]. Although these methods have shown promising
Figure 2: Oracle analyses with different datasets and metrics. From left to right: 3D IoU based mAP on KITTI, NuScenes Detection Score (NDS) and distance-based mAP on nuScenes. We replace our predictions with ground truth values step by step and observe the performance improvements. It can be seen that an accurate depth can bring significant performance improvement (green lines), and only with accurate depth can the improvements brought by other oracles be realized.

performance, they actually rely on dense depth labels and hence are not regarded as pure monocular approaches. There are also domain gaps between different depth sensors, making them hard to generalize smoothly to a new practical setting. Furthermore, the efficiency of processing a large number of point clouds is also a significant issue to deal with in practical applications.

End-to-end designs like 2D detection Recently work notices these drawbacks, and end-to-end frameworks are thus proposed. M3D-RPN [13] implements a single-stage multi-class detector with an end-to-end region proposal network and depth-aware convolution. SS3D [41] proposes to detect 2D key points and further predicts object characteristics with uncertainties. MonoDIS [12] introduces a disentangling loss to reduce the instability of the training procedure. Some of them still have multiple training stages or post-optimization phases. In addition, they all follow anchor-based manners, and thus the consistency of 2D and 3D anchors is needed to be determined. In contrast, anchor-free methods [26, 14, 42, 15] do not need to make statistics on the given data and have better generalized ability to more various classes or different intrinsic settings, so we choose to follow this paradigm.

Nevertheless, all of these works rarely have customized designs for instance depth estimation in particular, and only take it as one common regression target for isolated points or instances. It actually hinders the breakthrough of this problem, which will be discussed in our quantitative study and specifically addressed in our approach.

3 Preliminary and Motivating Study

In this section, we aim at making an in-depth quantitative error analysis on top of a basic adapted monocular 3D detector to investigate the key challenge in the specific 3D detection setting.

Typically, conventional 2D detection expects the model to predict 2D bounding boxes and category labels for each object of interest, while a monocular 3D detector needs to predict 7-DoF 3D boxes given the same input. From this perspective of problem formulation, the main difference lies on the regression targets. An intuitive reason for the much worse performance of monocular 3D detection compared to 2D is that there exist much more difficult targets to regress in the localization. Hence, we choose a simple detector FCOS3D [15] to study the specific problem, which keeps the well-developed designs for 2D feature extraction and is adapted for this 3D task with only basic designs for specific 3D detection targets. As shown in the left part of Fig. 3, there are overall two branches for classification and localization respectively. Formally, for the regression branch, the detector predicts 3D attributes, including offsets $\Delta x, \Delta y$ to the projected 3D center, depths $d$, 3D size $w^{3D}, l^{3D}, h^{3D}$, sin value of rotation $\theta$, direction class $C_\theta$, center-ness $c$, and distances to four sides of 2D boxes $l, r, t, b$, for each location on the output dense map. We further equip it with a basic consistency loss between 3D and 2D localization, which will be detailed in supplemental materials.

On this basis, we apply this baseline on two representative benchmarks, KITTI and nuScenes, and replace the predictions with ground truths step by step to identify the performance bottleneck (Fig. 2). Unexpectedly, the inaccurate depth blocks all the other sub-task predictions from improving the overall detection performance, on both datasets under different metrics. Hence, current monocular 3D detection, especially 3D localization, can be reduced to the dominating instance depth estimation problem to a great extent, which will be the focus of our method to be presented next. See more details about the explanation of oracle analyses in supplemental materials.
Figure 3: An overview of our framework. We start from a basic monocular 3D detector, FCOS3D, while focus on tackling the difficulty of instance depth estimation with our proposed customized module in the head. With the feature map from the regression branch as the input, we first introduce a branch for probabilistic depth estimation to model the uncertainty, then derive the geometric depth with the depth propagation graph and finally integrate them to get the final depth prediction.

4 Our Approach

Given images collected from similar cameras, previous work typically resorts to direct regression for instance depth estimation and expects the model to directly learn that objects with certain appearances and sizes always exist at locations with certain depths. Our baseline also follows this way. However, it is hard to learn due to the large variance and also obviously not enough for the accuracy needed in 3D detection. Given the inherent downside of hard regression for isolated points, in our approach, we aim at constructing an uncertainty-aware depth propagation graph to enhance the estimation from contextual connections among instances. Next, we will first elaborate on the adopted probabilistic representation and technical details of the constructed geometric graph, and finally present how we integrate these obtained depth estimations.

4.1 Uncertainty Modeling with Probabilistic Representation

For a one-stage detector, a general design for direct depth estimation is a small head along the regression branch expected to output a dense depth map. Formally, as shown in Fig. 3, suppose the input feature map has shape $H \times W$, then the direct depth regression output $D_R$ can be denoted as $D_R \in \mathbb{R}^{H \times W}$. On this basis, to establish an effective depth propagation mechanism, modeling the uncertainty of depth estimation for each instance is an important preliminary, which can provide useful guidance for weighing the propagation among instances. We adopt a simple yet effective probabilistic representation to achieve this: Considering the depth value is continuous in a certain range, we uniformly quantize the depth interval into a set of discrete values and represent the prediction with the expectation of the distribution. Suppose the detection range is $0 \sim D_{\text{max}}$, the discretized unit is $U$, then we have $C = \lfloor D_{\text{max}} / U \rfloor + 1$ split points. Denote the set of points as a weight vector $\omega \in \mathbb{R}^C$, and then we introduce a new head parallel with direct regression to produce a probabilistic output map $D_{\text{PM}}$, which will be decoded with:

$$D_P = \omega^T \text{softmax}(D_{\text{PM}})$$

where $D_P$ is the so-called probabilistic depth. It is equivalent to compute the expectation of the probabilistic distribution formed by $\text{softmax}(D_{\text{PM}})$. Apart from the $D_P$, we can further obtain the depth confidence score, denoted as $s^d \in S_D$, from the depth distribution of each instance. In practice, we take the average of top-2 confidence as the depth score for $U = 10m$. It will be multiplied by the center-ness and classification score as the final ranking criterion for predictions during inference.

Subsequently, we fuse $D_R$ and $D_P$ with the sigmoid response of a data-agnostic single parameter $\lambda$:

$$D_L = \sigma(\lambda)D_R + (1 - \sigma(\lambda))D_P$$

Here $D_L$ is regarded as a local depth estimation for each isolated instance, which together with the depth score derived from $D_{\text{PM}}$ serve as the foundation of constructing the depth propagation graph.

It is worth noting that our implementation is different from the typical way used in monocular depth estimation [37], which usually adopts a fine-grained quantization for the depth interval and further estimates the value with classification and residual regression. In comparison, our method is more memory-efficient, more straightforward for regressing continuous value, and provides a natural indicator for uncertainty estimation. Please refer to the supplemental for empirical results about comparison with other depth interval division methods.

\footnote{To make learning easier, the output of direct regression branch is applied an exponential transformation.}
With the depth prediction $D_d$ with which we can predict $f$ where will first derive the perspective relationship between two instances, and then present the details of the with this constraint to put objects in the context of the overall 3D scene by modeling the relationship (corresponding with $v$ (Down is the positive direction in the Fig. 4). Then we get:

The relation for $u$ given $v$ yields errors. Suppose

To simplify the result, we replace $v'$ with $v + c_v$, then $v$ represents the distance to the horizon line (Down is the positive direction in the Fig. 4). Then we get:

The relation for $u$ is similar. Considering the constraint that all the objects are on the ground, the bottom centers of objects always share the same $y$ (height in the camera coordinates), so we mainly consider this relation for $v$ next. Given two objects 1 and 2, the relationship between their center depths can be derived from Eqn. 5:

with which we can predict $d_2$ given $d_1$ precisely with the height difference between 3D centers. Besides, we can also leverage an approximation of this relationship given the assumption that objects share the same bottom height, then $y_2 - y_1$ can be substituted by the difference of half heights of 3D boxes $\frac{1}{2}(h_1^{3D} - h_2^{3D})$, defined as $d_{1\rightarrow 2}^r$.

In this relation, when $h_1^{3D} = h_2^{3D}$, $v_1d_1 = v_2d_2$, which is easy to understand, i.e., an object closer to vanishing line is farther away. It is a clear relationship connecting different instances but also yields errors. Suppose $|(y_2 - y_1) - \frac{1}{2}(h_1 - h_2)| = \delta$, the error of depth will be $\Delta d = \frac{1}{2}\delta$. When $\delta = 0.1m$, $v_2 = 50$ (pixels), $\Delta d$ can be about 1.5$m$. Although it is acceptable for objects 30$m$ away (corresponding with $v_2 = 50$), we also need a mechanism to avoid possible large errors. It consists of the edge pruning and gating scheme to be described next and the location-aware weight map to be mentioned in the Sec. 4.3.

Graph-Based Depth Propagation With the pairwise perspective relationship, we can estimate the depth of any object from the cues of other objects. Then we can construct a dense directed graph with two bidirectional edges between any two objects representing the depth propagation.
We evaluate our method on two datasets, KITTI [16] and nuScenes [17]. There are 7481/7518 samples for training/testing respectively on KITTI, and the training samples are generally divided into 3712/3769 samples as training/validation splits. We first validate our method on this popular benchmark. Nevertheless, the variety of scenes and categories is limited on KITTI, so we further test our approach on the large-scale nuScenes dataset. NuScenes consists of multi-modal data collected from 1000 scenes, including RGB images from 6 cameras, points from 5 Radars, and 1 LiDAR. It is split into 700/150/150 scenes for training/validation/testing. There are overall 1.4M annotated 3D bounding boxes from 10 categories. In addition, nuScenes uses different metrics, distance-based mAP and NDS, which can help evaluate our method from another perspective. See more explanations about metrics in supplemental materials.

5 Experiments

In this section, we present our experimental setup and implementation details, and then make the quantitative analysis on the KITTI and nuScenes dataset with details of both performance and efficiency. Finally, detailed ablation studies are conducted to show the efficacy of each component in our method. Refer to supplemental materials for more qualitative analysis.

5.1 Datasets & Evaluation Metrics

We evaluate our method on two datasets, KITTI [16] and nuScenes [17]. There are 7481/7518 samples for training/testing respectively on KITTI, and the training samples are generally divided into 3712/3769 samples as training/validation splits. We first validate our method on this popular benchmark. Nevertheless, the variety of scenes and categories is limited on KITTI, so we further test our approach on the large-scale nuScenes dataset. NuScenes consists of multi-modal data collected from 1000 scenes, including RGB images from 6 cameras, points from 5 Radars, and 1 LiDAR. It is split into 700/150/150 scenes for training/validation/testing. There are overall 1.4M annotated 3D bounding boxes from 10 categories. In addition, nuScenes uses different metrics, distance-based mAP and NDS, which can help evaluate our method from another perspective. See more explanations about metrics in supplemental materials.

5.2 Implementation Details

Network Architectures As shown in the Fig. 3, our baseline framework basically follows the design of FCOS3D [15]. Given the input image, we utilize ResNet101 [44] as the feature extraction backbone followed by FPN [45] for generating multi-level predictions. Detection heads are shared among multi-level feature maps except that three scale factors are used to differentiate some of their
Table 1: Results on the KITTI validation dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>Venue</th>
<th>Extra Labels</th>
<th>Time</th>
<th>AP BEV</th>
<th>IOU ≥ 0.7</th>
<th>AP 3D</th>
<th>IOU ≥ 0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mono3D [47]</td>
<td>CVPR 2016</td>
<td>Mask</td>
<td>4.2s</td>
<td>5.22</td>
<td>5.19</td>
<td>4.13</td>
<td>2.53</td>
</tr>
<tr>
<td>3DOP [38]</td>
<td>TPAMI 2017</td>
<td>Stereo</td>
<td>3s</td>
<td>12.63</td>
<td>9.49</td>
<td>7.59</td>
<td>6.55</td>
</tr>
<tr>
<td>MF3D [10]</td>
<td>CVPR 2018</td>
<td>Dense Depth</td>
<td>22.03</td>
<td>13.63</td>
<td>11.60</td>
<td>10.53</td>
<td>5.69</td>
</tr>
<tr>
<td>Mono3D++ [48]</td>
<td>AAAI 2018</td>
<td>Dense Depth+Shape</td>
<td>&gt;0.6s</td>
<td>16.70</td>
<td>11.50</td>
<td>10.10</td>
<td>7.60</td>
</tr>
<tr>
<td>PL [11, 36]</td>
<td>(AVOD)</td>
<td>CVPR 2019</td>
<td>19.0</td>
<td>15.3</td>
<td>13.0</td>
<td>7.5</td>
<td>6.1</td>
</tr>
<tr>
<td>ForeSee [36]</td>
<td>(A VOD)</td>
<td>AAAI 2020</td>
<td>-</td>
<td>23.4</td>
<td>17.4</td>
<td>15.9</td>
<td>15.0</td>
</tr>
</tbody>
</table>

Table 2: Results on the nuScenes dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Split</th>
<th>Modality</th>
<th>mAP</th>
<th>mAEE</th>
<th>mASE</th>
<th>mAER</th>
<th>mAVE</th>
<th>mAAE</th>
<th>NDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PointPillars (Light) [2]</td>
<td>test</td>
<td>LiDAR</td>
<td>0.305</td>
<td>0.517</td>
<td>0.290</td>
<td>0.500</td>
<td>0.316</td>
<td>0.368</td>
<td>0.453</td>
</tr>
<tr>
<td>CenterFusion [7]</td>
<td>test</td>
<td>Cam. &amp; Radar</td>
<td>0.326</td>
<td>0.631</td>
<td>0.261</td>
<td>0.516</td>
<td>0.614</td>
<td>0.115</td>
<td>0.449</td>
</tr>
<tr>
<td>CenterPoint v2 [52]</td>
<td>test</td>
<td>Cam. &amp; LiDAR &amp; Radar</td>
<td>0.671</td>
<td>0.249</td>
<td>0.236</td>
<td>0.350</td>
<td>0.250</td>
<td>0.136</td>
<td>0.714</td>
</tr>
<tr>
<td>LRM0</td>
<td>test</td>
<td>Camera</td>
<td>0.294</td>
<td>0.752</td>
<td>0.265</td>
<td>0.603</td>
<td>1.582</td>
<td>0.14</td>
<td>0.371</td>
</tr>
<tr>
<td>MonoDIS [12]</td>
<td>test</td>
<td>Camera</td>
<td>0.304</td>
<td>0.738</td>
<td>0.263</td>
<td>0.546</td>
<td>1.553</td>
<td>0.134</td>
<td>0.384</td>
</tr>
<tr>
<td>CenterNet [26]</td>
<td>test</td>
<td>Camera</td>
<td>0.338</td>
<td>0.658</td>
<td>0.255</td>
<td>0.629</td>
<td>1.629</td>
<td>0.142</td>
<td>0.4</td>
</tr>
<tr>
<td>Noah CV Lab</td>
<td>test</td>
<td>Camera</td>
<td>0.331</td>
<td>0.660</td>
<td>0.262</td>
<td>0.354</td>
<td>1.663</td>
<td>0.198</td>
<td>0.418</td>
</tr>
<tr>
<td>FCOS3D [15]</td>
<td>test</td>
<td>Camera</td>
<td>0.358</td>
<td>0.690</td>
<td>0.249</td>
<td>0.452</td>
<td>1.434</td>
<td>0.124</td>
<td>0.428</td>
</tr>
<tr>
<td>PGD (Ours)</td>
<td>test</td>
<td>Camera</td>
<td>0.386</td>
<td>0.626</td>
<td>0.245</td>
<td>0.451</td>
<td>1.509</td>
<td>0.127</td>
<td>0.448</td>
</tr>
<tr>
<td>CenterNet [26]</td>
<td>val</td>
<td>Camera</td>
<td>0.306</td>
<td>0.716</td>
<td>0.264</td>
<td>0.609</td>
<td>1.426</td>
<td>0.658</td>
<td>0.328</td>
</tr>
<tr>
<td>FCOS3D [15]</td>
<td>val</td>
<td>Camera</td>
<td>0.343</td>
<td>0.725</td>
<td>0.263</td>
<td>0.422</td>
<td>1.292</td>
<td>0.153</td>
<td>0.415</td>
</tr>
<tr>
<td>PGD (Ours)</td>
<td>val</td>
<td>Camera</td>
<td>0.369</td>
<td>0.683</td>
<td>0.260</td>
<td>0.439</td>
<td>1.268</td>
<td>0.185</td>
<td>0.428</td>
</tr>
</tbody>
</table>

Training Parameters
For all the experiments, we trained randomly initialized networks from scratch following end-to-end manners. Models are trained with SGD optimizer, in which gradient clip and warm-up policy are exploited with learning rate 0.001, number of warm-up iterations 500, warm-up ratio 0.33 and batch size 32/12 on 16/4 GTX 1080Ti GPUs for nuScenes/KITTI.

Data Augmentation
We only implement image flip for augmentation, where offset and 2D targets are flipped for the 2D image while 3D boxes are transformed correspondingly in 3D space. No other augmentation (right image augmentation, cropping, resizing, etc.) methods are utilized.

5.3 Quantitative Analysis
We make quantitative analyses both on KITTI (Tab. 1 and 4, Fig. 1(c)) and much harder, less commonly validated nuScenes dataset (Tab. 2). It can be seen that our method achieves the state-of-the-art on both benchmarks with different settings and metrics while maintains outstanding speed.

Then for the nuScenes dataset, we also compare the results on the test set and validation set, respectively. On the test set, we first compared all the methods using RGB images as the input data. Our single model achieved the best performance among them with mAP 37.0% and NDS 43.2%, in which we particularly exceeded the previous best method more than 3% in terms of mAP. We also list benchmarks based on other data modality, including lightweight, real-time PointPillars [2] with LiDAR, CenterFusion [7] with RGB image and Radar, and CenterPoint [52] ensemble results with all the sensors. It can be seen that although our method has a certain gap with the high-performance CenterPoint, it even surpasses PointPillars and CenterFusion on mAP, which shows that this ill-posed...
problem can be solved decently with enough data. At the same time, the methods using other modal
data usually yield better NDS, mainly because the mAVE is smaller. The reason is that they can predict the speed of objects from continuous multi-frame point clouds or velocity measurement of Radar. In contrast, we only use the single-frame image in our experiments. So how to mine the speed information from consecutive frame images will be a direction worthy of exploring in the future. On the validation set, we compare our method with the best open-source center-based detector, CenterNet. Our method is not only much more efficient to train and inference (3 days to train the CenterNet vs. only one day to train our model with comparable performance), but also achieves better performance, especially in terms of the mAP and mAOE. On this basis, we finally achieved an improvement of about 9% on NDS. See more detailed results about depth estimation accuracy and per-class detection performance in supplementary materials.

### Table 3: Ablation study on KITTI.

<table>
<thead>
<tr>
<th>Method</th>
<th>AP&lt;sub&gt;T&lt;/sub&gt;</th>
<th>mIOU&lt;sub&gt;T&lt;/sub&gt;</th>
<th>mAOE</th>
<th>mAAE</th>
<th>NDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCOS3D [15]</td>
<td>0.319</td>
<td>0.743</td>
<td>0.265</td>
<td>0.543</td>
<td>0.155</td>
</tr>
<tr>
<td>+Local cons.</td>
<td>0.325</td>
<td>0.723</td>
<td>0.266</td>
<td>0.546</td>
<td>0.164</td>
</tr>
<tr>
<td>+Prob. depth</td>
<td>0.339</td>
<td>0.716</td>
<td>0.265</td>
<td>0.511</td>
<td>0.163</td>
</tr>
<tr>
<td>+Dep. prop.</td>
<td>0.348</td>
<td>0.709</td>
<td>0.268</td>
<td>0.452</td>
<td>0.166</td>
</tr>
</tbody>
</table>

### Table 4: Results on the KITTI test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>AP&lt;sub&gt;T&lt;/sub&gt;</th>
<th>mIOU&lt;sub&gt;T&lt;/sub&gt;</th>
<th>mAOE</th>
<th>mAAE</th>
<th>NDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoVi-3D [53]</td>
<td>15.19</td>
<td>10.90</td>
<td>9.26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTM3D [14]</td>
<td>14.41</td>
<td>10.34</td>
<td>8.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PGD (Ours)</td>
<td>19.05</td>
<td>11.76</td>
<td>9.49</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 5: Ablation study on nuScenes.

<table>
<thead>
<tr>
<th>Method</th>
<th>AP&lt;sub&gt;T&lt;/sub&gt;</th>
<th>mIOU&lt;sub&gt;T&lt;/sub&gt;</th>
<th>mAOE</th>
<th>mAAE</th>
<th>NDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy Mod. Hard</td>
<td>20.58</td>
<td>16.30</td>
<td>14.99</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 6: Ablation study for the depth score.

<table>
<thead>
<tr>
<th>Method</th>
<th>AP&lt;sub&gt;T&lt;/sub&gt;</th>
<th>mIOU&lt;sub&gt;T&lt;/sub&gt;</th>
<th>mAOE</th>
<th>mAAE</th>
<th>NDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-2 score</td>
<td>19.05</td>
<td>11.76</td>
<td>9.49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norm. Entropy</td>
<td>20.15</td>
<td>15.89</td>
<td>14.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>± Std</td>
<td>20.68</td>
<td>16.26</td>
<td>14.94</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 7: Ablation study for probabilistic depth.

<table>
<thead>
<tr>
<th>prop. branch</th>
<th>w/direct</th>
<th>depth score</th>
<th>Easy</th>
<th>Mod.</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>19.10</td>
<td>16.04</td>
<td>14.83</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>16.58</td>
<td>13.82</td>
<td>13.49</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>18.54</td>
<td>15.44</td>
<td>13.94</td>
<td></td>
</tr>
</tbody>
</table>

### Table 8: Ablation study for geometric depth.

<table>
<thead>
<tr>
<th>Method</th>
<th>AP&lt;sub&gt;T&lt;/sub&gt;</th>
<th>mIOU&lt;sub&gt;T&lt;/sub&gt;</th>
<th>mAOE</th>
<th>mAAE</th>
<th>NDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy Mod. Hard</td>
<td>18.54</td>
<td>15.44</td>
<td>13.94</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5.4 Ablation Studies

Finally, we conduct ablation studies to validate the efficacy of our proposed key components on KITTI (Tab. 3) and nuScenes (Tab. 5). We can observe that local constraints can basically enhance the baseline, and our probabilistic and geometric depth further boost the performance significantly, especially in terms of mAP and translation error (mATE). Tab. 7 and Tab. 8 show more details of two core components for improving depth estimation with the metrics average precision under IOU≥0.7. It can be seen that combining the probabilistic representation (prop. branch in the Tab. 7) with direct regression (w/direct) and leveraging the depth score in the inference (depth score) can finally make the most of this design. For geometric depth, the basic fusion with local estimation can not bring the desirable gain. Improving the propagation graph via edge pruning and gating (edge gating) and cutting off the unexpected gradients propagation (cut off grad.) can help remove possible noises and prompt the learning more focused on the final integration, thus making the overall scheme much more effective. As for alternative implementations, we compare feasible methods of computing the depth score from the probabilistic distribution (Tab. 6). Compared to other more complicated ways, normalized entropy and standard deviation, our exploited top-2 score can achieve decent results with better efficiency. See more results about different depth division methods and detailed analyses for these two datasets from other perspectives like the Precision-Recall curve in the supplemental.

### 6 Conclusion

This paper targets the key challenge lying in monocular 3D object detection, instance depth estimation. Started from a basic adapted 3D detector, we firstly make in-depth oracle analyses. We surprisingly find that depth estimation is the dominating bottleneck for current 3D detection, especially in terms of localization. To tackle the discovered challenge, we propose a novel approach, Probabilistic and Geometric Depth (PGD), which leverages the geometric relationship in perspective to construct a graph connecting instance estimations with uncertainty and thus predicts depths more accurately. The efficacy of this solution is demonstrated on both KITTI and large-scale nuScenes datasets. In the future, we will further extend the geometric depth scheme to more general cases by relaxing the “ground” assumption via 2D height regression or ground normal estimation, and validate this pipeline on other 2D detectors. How to better leverage temporal geometry information to address the difficulty of instance depth estimation is also a promising direction worthy of further exploration.
Acknowledgments

This work is supported in part by Centre for Perceptual and Interactive Intelligence Limited, in part by the GRF through the Research Grants Council of Hong Kong under Grants (Nos. 14208417, 14207319 and 14203518) and ITS/431/18FX, in part by CUHK Strategic Fund and CUHK Agreement TS1712093, in part by the Shanghai Committee of Science and Technology, China (Grant No. 20DZ1100800).

References


dan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. CoRR, 


[22] L. Huang, Y. Yang, Y. Deng, and Y. Yu. Densebox: Unifying landmark localization with end to 


[27] A. Saxena, S. Chung, and A. Ng. Learning depth from single monocular images. In Advances 

[28] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still 
doi:10.1109/TPAMI.2008.132.

[29] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a 

monocular depth estimation. In IEEE Conference on Computer Vision and Pattern Recognition, 
2018.

[31] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang. Learning monocular depth by distilling cross-domain 


[33] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with 

[34] T. Zhou, M. Brown, N. Snavely, and D. Lowe. Unsupervised learning of depth and ego-motion 

In IEEE Conference on Computer Vision and Pattern Recognition, 2018.


monocular depth estimation. In IEEE Conference on Computer Vision and Pattern Recognition, 
2018.


