Monocular 3D Object Detection via Feature Domain Adaptation

Xiaoqing Ye†[1][0000−0003−3268−880X]⋆, Liang Du†[2][0000−0002−7952−5736]⋆⋆, Yifeng Shi[1], Yingying Li[1], Xiao Tan[1], Jianfeng Feng[2], Errui Ding[1], and Shilei Wen[1]

[1] Baidu Inc., China
[2] Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai, China, Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence (Fudan University), Ministry of Education, China

Abstract. Monocular 3D object detection is a challenging task due to unreliable depth, resulting in a distinct performance gap between monocular and LiDAR-based approaches. In this paper, we propose a novel domain adaptation based monocular 3D object detection framework named DA-3Ddet, which adapts the feature from unsound image-based pseudo-LiDAR domain to the accurate real LiDAR domain for performance boosting. In order to solve the overlooked problem of inconsistency between the foreground mask of pseudo and real LiDAR caused by inaccurately estimated depth, we also introduce a context-aware foreground segmentation module which helps to involve relevant points for foreground masking. Extensive experiments on KITTI dataset demonstrate that our simple yet effective framework outperforms other state-of-the-arts by a large margin.

Keywords: Monocular, 3D Object Detection, Domain Adaptation, Pseudo-Lidar.

1 Introduction

3D object detection is in a period of rapid development and plays a critical role in autonomous driving [16] and robot vision [4]. Currently, methods [34, 39, 47] based on LiDAR devices have shown favorable performance. However, the disadvantages of these approaches are also obvious due to the high cost of 3D sensors. Alternatively, the much cheaper monocular cameras are drawing increasing attention of researchers to dig into the problem of monocular 3D detection [3, 9, 10, 32, 37, 44]. Monocular-based methods can be roughly divided into two categories, one is RGB image-based approaches incorporating with geometry constraints [32] or semantic knowledge [9]. Unsatisfactory precision is observed due to the variance of the scale for an object caused by perspective projection and the lack of depth information. The other category leverages depth estimation to convert pixels into artificial point clouds, namely, pseudo-LiDAR [42,43],

⋆ yexiaoqing@baidu.com
⋆⋆ duliang@mail.ustc.edu.cn † The first two authors contributed equally to this work.

so as to boost the performance by borrowing benefits of approaches working for real-LiDAR points. In this case, 3D point cloud detection approaches [34] can be adopted for pseudo-LiDAR. Although recent works [42, 43, 46] have proven the superiority of pseudo-LiDAR in 3D object detection, the domain gap between pseudo-LiDAR and real LiDAR remains substantial due to their physical differences, which limits the higher performance of pseudo-LiDAR.

To solve the problem, most approaches [41, 46] investigate more accurate depth estimation methods such as DenseDepth [2], DispNet [31], PSM-Net [8], or take advantage of semantic segmentation [9] as well as instance segmentation [37] to obtain an unalloyed object point cloud with background filtered out. However, the extra information makes the framework too heavy for real scene application. As shown in Figure 1, compared against real LiDAR, pseudo-LiDAR point cloud has a weaker expression of the object structure.

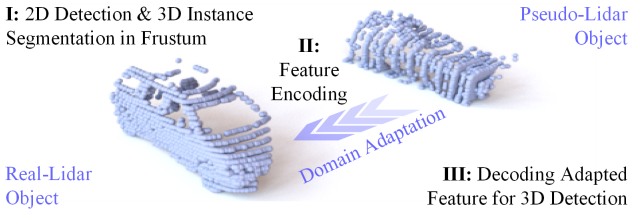

Fig. 1. A comparison of real and pseudo-LiDAR object point clouds. The real-LiDAR point cloud of the object has more accurate and crisper representation than pseudo-LiDAR, leading to a performance discrepancy. Domain adaptation approach is utilized to bridge the domain gap between these two modalities for further boosting the performance of monocular 3D object detection.

In contrast, we propose a simple yet effective way to boost the performance of pseudo-LiDAR by bridging the domain inconsistency between real LiDAR and pseudo-LiDAR with adaption. We build up a siamese network based on the off-the-shelf LiDAR based 3D object detection framework [34]. Two branches of the siamese network take real and pseudo-LiDAR as input, respectively. The difference between the two feature domains is minimized by the proposed network so as to encourage the pseudo-LiDAR feature being similar to the real-LiDAR feature. By narrowing the gap between high-dimensional feature representations of LiDAR and pseudo-LiDAR, our work surpasses previous state-of-the-arts on the KITTI 3D detection benchmark. The main contributions of our work are summarized as follows:

– We are the first to leverage domain adaptation approach to customize the features from pseudo-LiDAR domain to real-LiDAR domain, so as to bridge the performance gap between monocular-based and LiDAR point-based 3D detection approaches.

- To fully exploit the context information for feature generation, we investigate a context-aware foreground segmentation module (CAFS), which allows network using the both foreground and the context point clouds to map the pseudo features to discriminative LiDAR features.
- We achieve new state-of-the-art monocular 3D object detection performance on KITTI benchmark.

2 Related Works

2.1 LiDAR-based 3D object detection

Current LiDAR Based 3D object detection methods can be divided into three categories: (1) Multi-view based methods [11,23] project the LiDAR point clouds into bird's eye view (BEV) or front view to extract features, and a fusion process is applied to merge features. (2) Voxel-based. LiDAR point clouds are first divided into voxels and then learned by 3D convolutions [13, 47]. However, due to the sparsity and non-uniformity of point clouds, voxel-based methods suffer from high computation cost. To tackle this problem, sparse convolutions are applied on this form of data [12, 45]. Besides, (3) direct operation on raw points are also investigated recently [34–36, 39], since some researchers believe that data representation transformation may cause data variance and geometric information loss. For instance, F-PointNet [34] leverages both mature 2D object detectors and advanced 3D pointnet-based approach for robust object localization.

2.2 Monocular 3D object detection

Monocular-based 3D detection [5, 25, 28, 30, 33, 38, 40] is a more challenging task due to a lack of accurate 3D location information. Most prior works [3, 9, 10, 24, 37, 44] on monocular 3D detection were RGB image-based, with auxiliary information like the semantic knowledge or geometry constraints and so on. AM3D [29] designed two modules for background points segmentation and RGB information aggregation respectively in order to improve 3D box estimation. Some other works involved 2D-3D geometric constraints to alleviate the difficulty caused by scale variety. Mousavian et al. [32] argued that 3D bounding box should fit tightly into 2D detection bounding box according to geometric constraints. Deep MANTA [7] encoded 3D vehicle information using key points of vehicles.

Another recently introduced approach for monocular 3D detection is based on pseudo-LiDAR [42, 43, 46], which utilizes depth information to convert image pixels into artificial point clouds, i.e., pseudo-LiDAR, and employs LiDAR-based frameworks for further detection. PL-MONO [42] was the pioneer work that pointed out the main reason for the performance gap is not attributed to the inaccurate depth information, but data representation. Their work achieved impressive improvements by converting image-based depth maps to pseudo-LiDAR representations. Mono3D-PLiDAR [43] trained a LiDAR-based 3D detection network with pseudo-LiDAR; therefore the LiDAR-based methods can work with

a single image input. They also point out the noise in pseudo-LiDAR data is a bottleneck to improve performance. You et al. [46] believed pseudo-LiDAR relies heavily on the quality of depth estimation, so a stereo network architecture to achieve more accurate depth estimation is proposed. While pseudo-LiDAR has largely improved the performance of monocular 3D detection, there is still a notable performance gap between pseudo-LiDAR and real LiDAR.

2.3 Domain adaptation

As shown in Figure 1, given the same object, distinct point distribution discrepancy can be noticed between the depth-transformed pseudo-LiDAR and real LiDAR, which leads to a large domain gap between two modalities. Domain adaptation [1] is a machine learning paradigm aiming at bridging different domains. The critical point lies in how to reduce the distribution discrepancy across different domains while avoiding over-fitting. Thanks to deep neural networks that are able to extract high-level representations behind the data, domain adaptation [22] has made tremendous progress in object detection [19] and semantic segmentation [18]. In order to keep from overly dependent on the accuracy of depth estimation, we take advantage of LiDAR guidance in training stage to adapt the pseudo-LiDAR features to act as real-LiDAR features do. Following [27], our DA-3Ddet employs the L2-norm regularization to calculate the feature similarity for domain adaptation.

3 Methodology

3.1 Overview

The proposed framework DA-3Ddet is depicted in Figure 2. It consists of two main components, the siamese branch for feature domain adaption and the context-aware foreground segmentation module. First the overall pipeline is introduced, then the two critical modules are elaborated in detail, and finally the training loss is given.

Recent research works [42] have verified the superiority of adopting pseudo-LiDAR representations from estimated depth to mimicking LiDAR point clouds for 3D object detection. However, there is still a large performance gap between pseudo-based and real LiDAR detection methods due to two reasons. For one thing, the generated pseudo-LiDAR representations heavily rely on the accuracy of the estimated depth. For another, the distribution and density are physically different between the two representations, as shown in Figure 1. In light of such domain inconsistency issue, we propose a simple yet effective method to boost the performance of 3D object detection from only a monocular image. Rather than hunting for expensive multi-modal fusion strategies to improve the pseudo-LiDAR approaches, we build up a siamese network leveraging off-the-shelf LiDAR-based 3D detection frameworks as the backbone. Further, domain adaptation approach between different modal features is adopted to guide the pseudo-LiDAR representations to be closer to real-LiDAR representations.

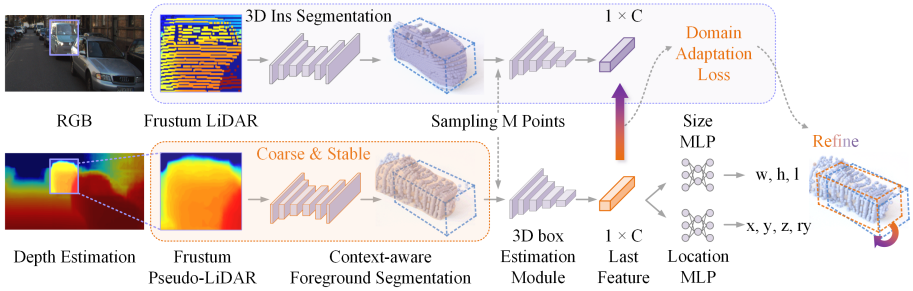

Fig. 2. Illustration of the proposed DA-3Ddet. A 2D object detector is first utilized to recognize and localize the objects before lifting the concerned 2D regions to 3D frustum proposals. Given a point cloud in a frustum ($N \times 3$ with N points and three channels of XYZ for each point), the object instance is segmented by binary classification of each point. The segmented foreground point cloud ($M \times 3$) is then encoded by 3D box estimation network. Domain adaptation is performed between the last layer ($1 \times C$) of real and pseudo encoded features. Finally, the box estimation net is employed to predict 3D bounding box parameters.

The input of our framework is a monocular image, and during the training process, real-LiDAR data is also utilized for feature domain adaption. Only a single image is required during the inference stage. First, the depth map is estimated given a monocular image and then transformed into point clouds in the LiDAR coordinate system, namely, pseudo-LiDAR. After that, real and pseudo LiDAR data of the same scene are simultaneously fed into the siamese branches, respectively, to obtain their high-dimensional feature representations. The features of pseudo-LiDAR domain are adapted to real-LiDAR feature domain during the training process. Finally, the aligned pseudo feature is decoded to regress the 3D parameters of detected objects. More technical details of our approach will be explained in the following sections.

3.2 Siamese framework for adapting pseudo-LiDAR to LiDAR

To narrow the gap between pseudo-LiDAR generated by depth maps and real LiDAR based methods, we propose a naive yet effective adaption method. Any off-the-shelf LiDAR-based 3D object detection networks can be utilized as the backbone for encoding 3D points data. To fairly compare with most existing pseudo approaches like [42], we adopt the same framework frustum PointNet (F-PointNet) [34] as our baseline.

First, we briefly review the pipeline of frustum PointNet [34]. A 2D object detection network is applied to the monocular image to detect the objects. Next, each region within the 2D bounding box is lifted to 3D frustum proposals. Each frustum point cloud is then fed into a PointNet encoder for 3D instance segmentation to perform foreground and background classification. Based on the masked object point cloud after binary segmentation, a simplified regression

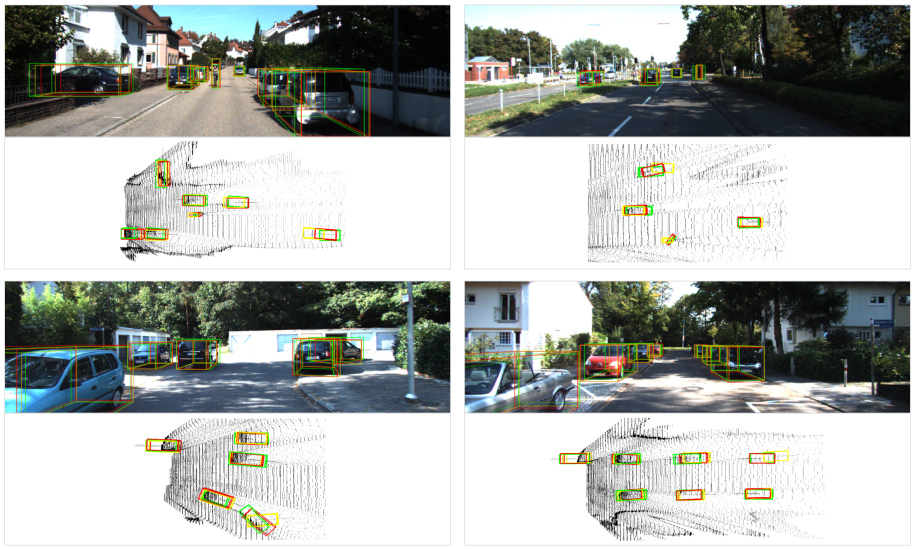

Fig. 3. The qualitative comparison of the ground truth (green), the baseline (yellow) and our method (red) on KITTI val set. The first row shows RGB images, and the second row shows the bird's-eye view, respectively. Our method effectively predicts reliable 3D bounding box of objects even with inaccurate depth estimation.

PointNet (T-Net) is further applied to translate the mask center to amodal box center. Finally, another PointNet module is followed to regress 3D box parameters. More details can be found in [34]. The advantage of the chosen baseline lies in its employment of 2D detector to restrain interested regions as well as operation on raw point clouds, which makes it robust to strong occlusion and sparsity at low cost.

In our work, we adopt a siamese network consisting of two branches of frustum PointNet, as depicted in Figure 2. For the upper branch corresponding to the real LiDAR, we utilize the pretrained model as prior, and it is only forwarded during the training process with parameters fixed. For the lower depth-transformed pseudo-LiDAR branch, the extracted frustum is fed into the frustum PointNet-based module, which is exactly the same architecture with the upper branch. Similar to F-PointNet [34], we also adopt a PointNet-based 3D instance segmentation network to filter out background or irrelevant instance point clouds in the frustum. Differently, due to inaccurate estimated depth, pseudo-LiDAR points within the ground truth 3D bounding box (provided by 3D LiDAR) can be inconsistent with the foreground points derived from 2D images. In consequence, a context-aware foreground segmentation module is proposed to alleviate the adverse effects caused by inaccurate depth estimation, which will be further explained in the following subsection.

The generated features of the two branches before the final head of 3D bounding box regression are encoded into $(1 \times C)$, respectively, where C is the channel

number of the encoded feature. To make the domain adaptation more focused, it is restricted to segmented foreground points only. We calculate the $\mathcal{L}2$ distance between the pseudo and real LiDAR high-dimensional features to perform domain adaptation so that the engendered representations of pseudo-LiDAR resemble real-LiDAR features. After that, the amodal 3D box estimation network is adopted to decode the features after adaptation, so as to regress the 3D box parameters of the object. After the pseudo-LiDAR branch network is tuned to achieve an aligned feature domain, we can simply discard the upper real-LiDAR branch at inference time.

3.3 Context-aware foreground segmentation

In real LiDAR-based 3D detection approaches, LiDAR points within the ground truth 3D bounding box are utilized as the supervising signal for 3D instance segmentation to filter out background points. However, for pseudo-LiDAR point clouds, points computed from 2D foreground instance pixels may be inconsistent with 3D foreground mask ground truth due to inaccurate depth estimation, as shown in Figure 4. To be more specific, if the estimated depth differs from the ground truth depth to some degree, there can be fewer points within the 3D ground truth box, increasing the difficulty for regress the 3D object parameters. In contrast, relevant points (regions colored in light green in Figure4) that contain useful structural information are excluded by the ground truth 3D foreground mask. An extreme case can be no valid pseudo-LiDAR object points found in the ground truth 3D box in far-away regions due to large depth estimation offset to actual distance.

Although neglected by previous works, we argue that it must not be overlooked. To tackle this problem, we investigate a context-aware foreground segmentation module (CAFS). We first train a baseline model with 3D instance segmentation loss like F-PointNet [34] does. The pretrained model serves as a prior to help CAFS module recognize and segment the foreground point clouds in a coarse manner. During the end-to-end whole siamese network training, the foreground segmentation loss is then discarded to let the CAFS select both foreground and background points to generate stable and abundant features for domain adaptation.

3.4 Training Loss.

In order to align the 3D parameters with projected 2D bounding boxes according to projective transformation, we define the ground truth 2D bounding box for our 2D detector: $[x_1, y_1, x_2, y_2]$ according to projected 3D ground truth boxes. In specific, we project all ground truth 3D bounding boxes onto image space given the camera intrinsic, as Equation 1 shows.

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_{P_c} \tag{1}$$

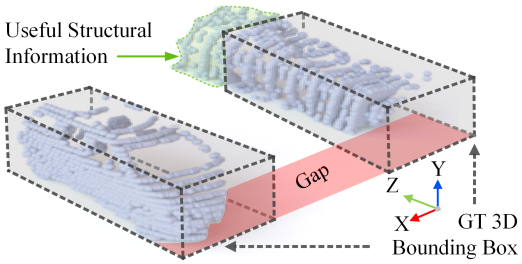

Fig. 4. An example of pseudo-LiDAR and real-LiDAR instance point cloud. Adopting GT 3D bounding box as supervising signal of pseudo-LiDAR can lead to the lost of structural information due to inaccurate depth estimation.

Where the left point $[u,\ v,\ 1]$ denotes the projected 2D image coordinate and Z is the depth, whereas the right part denotes 3D point in camera coordinate and K is the camera intrinsic computed in advance and used during the training and inference stage. Rather than directly adopting 2D annotations provided by KITTI, the computed minimum bounding rectangle of the projected eight 3D vertices are served as the ground truth of 2D detector. By means of 2D-3D alignment, the 3D prediction is jointly optimized.

For each input frustum point cloud, the outputs are parameterized as follows: $[c_x, c_y, c_z]_{3D}$, $[h, w, l]_{3D}$, $[R_y^{(m)}, R_{offset}^{(m)}]$, $C^{(s)}$, $\text{Corner}_i, i \in 1, ..., 8$, where $[c_x, c_y, c_z]_{3D}$ and $[h, w, l]_{3D}$ are the regressed center and size of 3D box, respectively. As is proved in previous works [34], a hybrid of classification and regression formulations makes heading angle estimation more robust. Thus the network predicts the scores of each equally split angle bins as well as their offset to the center of each bin, i.e., $[\text{R}_y^{(m)}, R_{offset}^{(m)}]$, where m is the predefined number of bins. $C^{(s)}$ denotes the class of the given object, with s refers to the categories.

Following [34], for the sub-network of training pseudo-LiDAR data we adopt the same loss function, including cross-entropy foreground segmentation loss $\mathcal{L}_{seg}$, smooth-L1 3D box regression loss $\mathcal{L}_{3D_{reg}}$, cross-entropy classification loss $\mathcal{L}_{cls}$, as well as corner loss $\mathcal{L}_{corner}$. For corner loss, we compute the minimum one of the two mean distances between the eight corners derived from the predicted angle and its flipped angle with respect to ground truth:

$$\mathcal{L}_{corner} = \tfrac{1}{8} min(\sum_{i=1}^{8} |C_i - C_i^*|, \sum_{i=1}^{8} |C_i - C_i^{*'}|) \tag{2}$$

where C_i^* and $C_i^{*'}$ are predicted corners and the corners with flipped angle, C_i is the ground truth corner.

$$\mathcal{L}_{3D} = \mathcal{L}_{seg} + \mathcal{L}_{3D_{reg}} + \mathcal{L}_{cls} + \mathcal{L}_{corner} \tag{3}$$

For the domain adaption loss, we compute the $\mathcal{L}_2$ loss for feature alignment:

$$\mathcal{L}_{DA} = \mathcal{L}_2(\mathcal{F}_{real}, \mathcal{F}_{pseudo}) \tag{4}$$

Table 1. Comparison on KITTI val and test set. The average precision (in %) of "Car" on 3D object detection (AP_{3D}) at IoU = 0.7 is reported. Our proposed DA-3Ddet achieves new state-of-the-art performance.

Method	Val			Test		
	Mod.	Easy	Hard	Mod.	Easy	Hard
SS3D [20]	13.2	14.5	11.9	7.7	10.8	6.5
RT-M3D [26]	16.9	20.8	16.6	10.1	13.6	8.2
Pseudo-LiDAR [42]	17.2	19.5	16.2	/	/	/
M3D-RPN [5]	17.1	20.3	15.2	9.7	14.8	7.4
Decoupled-3D [6]	18.7	27.0	15.8	7.3	11.7	5.7
Mono3D-PliDAR [43]	21.0	31.5	17.5	7.5	10.8	6.1
AM3D [29]	21.1	32.2	17.3	10.7	16.5	**9.5**
Ours	**24.0**	**33.4**	**19.9**	**11.5**	**16.8**	8.9

The overall loss is then computed by Equation 5, where α is a hyperparameter to balance these two terms.

$$\mathcal{L}_{all} = \mathcal{L}_{3D} + \alpha\mathcal{L}_{DA} \tag{5}$$

4 Experiment

4.1 Implementation

Dataset. The proposed approach is evaluated on the KITTI 3D object detection benchmark [15,16], which contains 7,481 images for training and 7,518 images for testing. We follow the same training and validation splits as suggested by [11], i.e., 3,712 and 3,769 images for train and val, respectively. For each training image, KITTI provides the corresponding LiDAR point cloud, right image from stereo cameras, as well as camera intrinsics and extrinsics.

Metric. We focus on 3D and bird's-eye-view (BEV) object detection and report the average precision (AP) results on validation and test set. Specifically, for "Car" category, we adopt IoU = 0.7 as threshold following [11]. Besides, to validate the effectiveness on the other two categories - "Pedestrian" and "Cyclist", we also include corresponding experiments on the two categories with IoU = 0.5 for fair comparison. AP for 3D and BEV tasks are denoted by AP_{3D} and AP_{BEV}, respectively. Note that there are three levels of difficulty defined in the benchmark according to the 2D bounding box height, occlusion and truncation degree, namely, easy, moderate and hard. The KITTI benchmark ranks algorithms mainly based on the moderate AP.

Monocular depth estimation. Different depth estimation approaches can have influence on the transformed pseudo-LiDAR. Thanks to the proposed feature domain adaptation and context-aware foreground segmentation module, our framework works on various depth predictors regardless of the degree of accuracy. For fair comparison with other works, we adopt the open-sourced monocular depth estimator DORN [14] to obtain depth maps. Note that our proposed

framework observes improvements on various depth estimation strategies and experiment result in Table7 validates the effectiveness.

Results on other categories. Due to the small sizes and non-rigid structures of the other two classes - "Pedestrian" and "Cyclist", it is much more challenging to perform 3D object detection from monocular image than cars. We guess it could be the reason that most of the previous monocular methods simply report their results on "Car" only. Nevertheless, we still conduct self-compared experiments with respect to the baseline on the given two classes. Table 2 reports the AP_{3D} results on KITTI val set at IoU = 0.5. Although the results seem worse than "Car", compared to the baseline pseudo-LiDAR, we observe an improvement on both categories at all difficulties due to our domain adaptation and context-aware 3D foreground segmentation module.

Table 2. Bird's eye view detection (AP_{BEV}) / 3D object detection (AP_{3D}) performance for "Pedestrian" and "Cyclist" on KITTI val split set at IoU = 0.5.

Method	Cyclist			Pedestrian		
	Mod.	Easy	Hard	Mod.	Easy	Hard
Baseline	10.8/10.6	11.7/11.4	10.8/10.6	9.3/6.0	11.7/7.2	7.8/5.4
Ours	**12.2/11.5**	**15.5/14.5**	**11.8/11.5**	**10.6/7.1**	**13.1/8.7**	**9.2/6.7**

Pseudo-LiDAR frustum generation. First the estimated depth map is back-projected into 3D points in LiDAR's coordinate system by the provided calibration matrices. Second, utilizing the 2D detector which is trained with the minimum bounding rectangle of the projected vertices of 3D boxes as ground truth, the frustum is lifted to serve as the input of our siamese network.

Training details. The network is optimized by Adam optimizer [21] with initial learning rate 0.001 and a mini-batch size of 32 on TITAN RTX GPU. The number of points of the network input is fixed to 1024. Frustum with less than 1024 points will be sampled repeatedly and otherwise be randomly down-sampled. For training the coarse foreground segmentation module with loss, we trained for 150 epochs, and after that we further trained for 150 epochs with segmentation loss discarded. α in Equation 5 is set to 1.0. And for final output, in order to decouple the size and location properties, we add a three-layer MLP to regress the size and location parameters of 3D box, respectively.

4.2 Comparison with state-of-the-art methods

Results on KITTI test and val. The 3D object detection results on KITTI val and test set are summarized Table 1 at IoU threshold = 0.7. Compared with the top-ranked monocular-based 3D detection approaches in KITTI leader board, our method consistently outperforms the other methods and ranks 1st. In specific, (1) Both on validation and test set, our method achieves the highest performance on the moderate set, which is the main setting for ranking on the KITTI benchmark. Large margins are observed over the second top-performed

Table 3. Ablative analysis on the KITTI val split set for AP_{3D} at IoU = 0.7. Experiment group (a) is our baseline method. Different experiment settings are applied: using domain adaptation (DA), using GT bounding box to supervise the 3D instance segmentation (SP), unsupervised segmentation (UnSp), using our context-aware foreground segmentation method (CAFS), single decoder (D1), two-branch decoder (D2) and three-branch decoder (D3).

Group	DA	SP	UnSP	CAFS	D1	D2	D3	Mod.	Easy	Hard
(a)		✓			✓			21.9	28.6	18.4
(b)			✓		✓			15.7	18.5	14.8
(c)				✓	✓			22.4	29.4	18.8
(d)		✓				✓		22.1	28.8	18.5
(e)		✓					✓	21.4	27.6	18.2
(f)	✓	✓			✓			23.1	31.8	19.2
(g)	✓		✓		✓			16.1	19.0	14.9
(h)	✓			✓	✓			23.6	32.9	19.7
(i)	✓			✓		✓		**24.0**	**33.4**	**19.9**
(j)	✓			✓			✓	23.2	31.7	19.4

method (AM3D [29]), that is, 13.6% and 4.7% on val and test set, respectively. (2) Some top-ranked monocular methods utilize extra information for 3D detection. For example, Mono3D-PliDAR [43] uses the instance mask instead of the bounding box as the representation of 2D proposals and AM3D [29] designs two extra modules for background points segmentation and RGB information aggregation. In contrast to the above-mentioned methods that utilize extra information, our simple yet effective method achieves appealing results.

4.3 Ablation study

Main ablative analysis. We conduct ablation study by making comparison among ten variants of the proposed method as shown in Table 3. "DA" means applying our domain adaptation approach for pseudo-LiDAR. "SP" and "UnSP" represent the network with or without 3D instance segmentation loss when selecting foreground points during the whole training process, respectively. CAFS denotes the proposed module that segment the foreground points with context information for further domain adaptation and 3D bounding box regression. Furthermore, to compare the effect of decoupling size and location parameters of 3D object, three different head decoders are experimented. "D1", "D2" and "D3" indicate adopting single, double ("$xyzr$" and "whl") and triple ("xyz", "whl", "r") decoding branches and each branch is composed of a three-layer MLP, respectively. The baseline (a) is constructed following F-PointNet [34], with supervised instance segmentation loss during the whole training process and only one decoder branch, no domain adaptation is utilized. Note that for the baseline, we use the modified 2D detector with a minimum bounding rectangle of the projected eight 3D corners as ground truth.

As depicted in Table 3, we can observe that: (1) Compared (a) with (b), (c), we found that without domain adaptation, the performance can be deteriorated

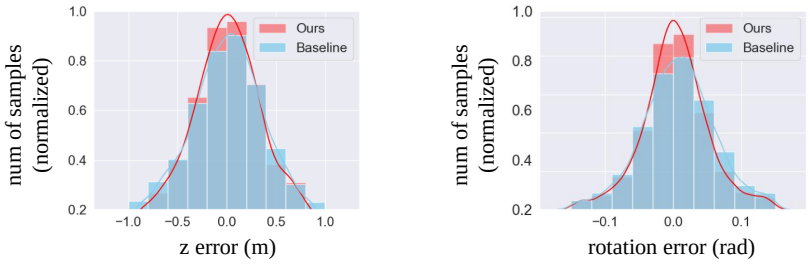

Fig. 5. The statistic analysis and comparison of the baseline method (blue) and our DA-3Ddet (red). The y axis of the chart represents the number of samples after normalization. Our method effectively improves the critical metrics "z" and rotation "ry" in 3D object detection.

by simply removing the foreground segmentation loss, since the regression relies on the useful foreground points rather than background noises. If we divide the whole training process into two stages and apply our context-aware foreground segmentation (CAFS) module, the performance of the baseline can be improved from 21.9 to 22.4 in moderate setting. (2) Compared (a) with (f), a noticeable gain is achieved due to our feature domain adaptation (21.9 vs. 23.1). (3) In addition, changing from supervised instance segmentation loss to our adaptive context-aware foreground segmentation further improves the performance, reflected from (g) to (h). (4) Finally, we also compare different decoder branch settings and find that employing separate heads for "$xyzr$" and "whl" achieves the best performance. We believe that "whl" are of size parameters, whereas "$xyzr$" is related to location and is estimated in residual form. Decoupling the two groups can slightly benefit to the regression task.

Statistic analysis on 3D metric. For monocular 3D detection, the depth (i.e., "z" in distance) and rotation "ry" around Y-axis in camera coordinates are the most challenging parameters, which have significant influence on the 3D detection precision. As a result, for further detailed explanation of the improvement over the baseline, we compare the errors on the above-mentioned two metrics of the baseline and our proposed DA-3Ddet. As shown in Figure 5, we can clearly see that our proposed method improves the baseline method in "z" and "ry", which results in more accurate monocular 3D object detection.

Table 4. Comparison of different sampling rates on val set at IoU = 0.7. The number of sampling points during the training and testing process is the same.

Sampling Num	AP_{3D}			AP_{BEV}		
	Mod.	Easy	Hard	Mod.	Easy	Hard
768	23.5	32.4	19.6	32.1	45.0	26.6
1024	**24.0**	**33.4**	**19.9**	32.7	45.5	27.1
1536	23.9	32.3	19.8	33.1	45.6	27.2
2048	23.8	32.4	19.7	**33.1**	**46.0**	**27.2**

Impact of different point cloud densities. Real LiDAR point clouds are sparse and non-uniform whereas the depth-transformed pseudo-LiDAR data can be denser. As a result, to compare the influence of points number within the lifted frustum, we conduct the experiments with different point sampling rates from the frustum. As shown in Table 4, for 3D detection task, 1024 points perform best. For detection in bird's eye view, more points (2048) achieves better results. We claim that our framework is robust to point numbers to some degree since the performance gap of different densities is small.

Impact of different loss functions for domain adaptation. For feature domain adaptation, we experimented with two kinds of losses, namely, the $\mathcal{L}1$ and $\mathcal{L}2$. Table 5 shows that $\mathcal{L}2$ performs better than $\mathcal{L}1$.

Table 5. Comparison of different loss functions for adaptation on val set at IoU = 0.7.

Adaptation Loss	AP_{3D}			AP_{BEV}		
	Mod.	Easy	Hard	Mod.	Easy	Hard
$\mathcal{L}1$	23.3	32.1	19.4	32.5	44.8	26.9
$\mathcal{L}2$	**24.0**	**33.4**	**19.9**	**32.7**	**45.5**	**27.1**

4.4 Generalization ability

For generalization ability validation, we include two kinds of experiments. The first aims to verify that feature adaptation can generalize to other data modalities, such as monocular to stereo, stereo to LiDAR. The second aims to prove that our approach gains improvement on different depth estimation methods.

Domain adaptation between different modalities. To validate the effectiveness and generalization ability of our domain adaptation based method, we perform feature adaptation between different data modalities. As illustrated in Table 6, the "LiDAR" and "Stereo" indicate adopting the same baseline for real-LiDAR and stereo-based pseudo-LiDAR method. Stereo $\Rightarrow$ LiDAR, "Mono" $\Rightarrow$ "Stereo" and "Mono" $\Rightarrow$ "LiDAR" denote the feature adaptation between different modalities, respectively. The experiment results demonstrate that the feature domain adaptation from a less accurate feature representation to a more reliable feature representation could largely improve the 3D detection performance for both stereo and monocular approaches.

Impact of different depth estimators In this experiment, we choose the unsupervised depth estimator MonoDepth [17] as well as the supervised monocular DORN [14] for comparison. As is known, supervised approaches have higher accuracy in depth prediction than unsupervised methods. As shown in Table 7, the 3D detection precision is positively correlated with the accuracy of estimated depth. Besides, improvements can be noticed both in unsupervised and supervised depth predictors.

Table 6. Results on AP_{3D} and AP_{BEV} via domain adaptation between different data modalities on KITTI val at IoU = 0.7. "LiDAR", "Stereo" and "Mono" represent using the single modality data for training without domain adaptation. "⇒" denotes the adaptation direction between different modalities. DORN [14] and PSMNet [8] are adopted to generate the pseudo-LiDAR of "Mono" and "Stereo".

Modality	AP_{3D}			AP_{BEV}		
	Mod.	Easy	Hard	Mod.	Easy	Hard
LiDAR	67.9	84.8	58.8	79.0	88.5	69.5
Stereo	44.0	59.2	36.4	55.2	73.0	46.3
Stereo ⇒ LiDAR	**46.1**	**66.7**	**38.2**	**56.1**	**73.8**	**47.3**
Mono	21.9	28.4	18.4	30.7	42.7	25.5
Mono ⇒ Stereo	23.4	32.0	19.5	32.0	44.8	26.7
Mono ⇒ LiDAR	**24.0**	**33.4**	**19.9**	**32.7**	**45.5**	**27.1**

Table 7. Comparison of different depth estimators on val split set at IoU = 0.7.

Depth	Method	AP_{3D}			AP_{BEV}		
		Mod.	Easy	Hard	Mod.	Easy	Hard
MonoDepth [17]	Baseline	16.4	20.4	15.2	22.6	31.2	18.6
	Ours	**18.1**	**23.9**	**16.6**	**23.8**	**33.5**	**19.7**
DORN [14]	Baseline	21.9	28.4	18.4	30.7	42.7	25.5
	Ours	**24.0**	**33.4**	**19.9**	**32.7**	**45.5**	**27.1**

5 Conclusions

In this paper, we present a monocular 3D object detection framework based on domain adaptation to adapt features from the noisy pseudo-LiDAR domain to accurate real LiDAR domain. Motivated by the overlooked problem of foreground inconsistency between pseudo and real LiDAR caused by inaccurate estimated depth, we also introduce a context-aware foreground segmentation module which uses both foreground and the certain context points for foreground feature extraction. In future work, Generative Adversarial Networks is considered for feature domain adaptation instead of simple $\mathcal{L}2$ and RGB information may be incorporated with pseudo-LiDAR.

Acknowledgments

This work was supported by National Key R&D Program of China (2019YFA0709502), the 111 Project (NO.B18015), the key project of Shanghai Science & Technology (No.16JC1420402), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01) and ZJLab, National Key R&D Program of China (No.2018YFC1312900), National Natural Science Foundation of China (NSFC 91630314).

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. arXiv preprint arXiv:1707.02392 (2017)
2. Alhashim, I., Wonka, P.: High quality monocular depth estimation via transfer learning. arXiv preprint arXiv:1812.11941 (2018)
3. Atoum, Y., Roth, J., Bliss, M., Zhang, W., Liu, X.: Monocular video-based trailer coupler detection using multiplexer convolutional neural network. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5477–5485 (2017)
4. Biegelbauer, G., Vincze, M.: Efficient 3d object detection by fitting superquadrics to range image data for robot's object manipulation. In: Proceedings 2007 IEEE International Conference on Robotics and Automation. pp. 1086–1091. IEEE (2007)
5. Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9287–9296 (2019)
6. Cai, Y., Li, B., Jiao, Z., Li, H., Zeng, X., Wang, X.: Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation. arXiv preprint arXiv:2002.01619 (2020)
7. Chabot, F., Chaouch, M., Rabarisoa, J., Teuliere, C., Chateau, T.: Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2040–2049 (2017)
8. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5410–5418 (2018)
9. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2147–2156 (2016)
10. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals for accurate object class detection. In: Advances in Neural Information Processing Systems. pp. 424–432 (2015)
11. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1907–1915 (2017)
12. Du, L., Ye, X., Tan, X., Feng, J., Xu, Z., Ding, E., Wen, S.: Associate-3ddet: Perceptual-to-conceptual association for 3d point cloud object detection. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
13. Engelcke, M., Rao, D., Wang, D.Z., Tong, C.H., Posner, I.: Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 1355–1361. IEEE (2017)
14. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2002–2011 (2018)
15. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
16. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361. IEEE (2012)

17. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 270–279 (2017)
18. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
19. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5001–5009 (2018)
20. Jörgensen, E., Zach, C., Kahl, F.: Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss. arXiv preprint arXiv:1906.08070 (2019)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
22. Kouw, W.M., Loog, M.: An introduction to domain adaptation and transfer learning. arXiv preprint arXiv:1812.11806 (2018)
23. Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L.: Joint 3d proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–8. IEEE (2018)
24. Ku, J., Pon, A.D., Waslander, S.L.: Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11867–11876 (2019)
25. Li, B., Ouyang, W., Sheng, L., Zeng, X., Wang, X.: Gs3d: An efficient 3d object detection framework for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1019–1028 (2019)
26. Li, P., Zhao, H., Liu, P., Cao, F.: Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. arXiv preprint arXiv:2001.03343 (2020)
27. Li, X., Grandvalet, Y., Davoine, F.: Explicit inductive bias for transfer learning with convolutional networks. arXiv preprint arXiv:1802.01483 (2018)
28. Liu, L., Lu, J., Xu, C., Tian, Q., Zhou, J.: Deep fitting degree scoring network for monocular 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1057–1066 (2019)
29. Ma, X., Wang, Z., Li, H., Zhang, P., Ouyang, W., Fan, X.: Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6851–6860 (2019)
30. Manhardt, F., Kehl, W., Gaidon, A.: Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2069–2078 (2019)
31. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4040–4048 (2016)
32. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3d bounding box estimation using deep learning and geometry. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7074–7082 (2017)
33. Naiden, A., Paunescu, V., Kim, G., Jeon, B., Leordeanu, M.: Shift r-cnn: Deep monocular 3d object detection with closed-form geometric constraints. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 61–65. IEEE (2019)

34. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 918–927 (2018)
35. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
36. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017)
37. Qin, Z., Wang, J., Lu, Y.: Monogrnet: A geometric reasoning network for monocular 3d object localization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8851–8858 (2019)
38. Roddick, T., Kendall, A., Cipolla, R.: Orthographic feature transform for monocular 3d object detection. arXiv preprint arXiv:1811.08188 (2018)
39. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–779 (2019)
40. Simonelli, A., Bulo, S.R., Porzi, L., López-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1991–1999 (2019)
41. Vianney, J.M.U., Aich, S., Liu, B.: Refinedmpl: Refined monocular pseudolidar for 3d object detection in autonomous driving. arXiv preprint arXiv:1911.09712 (2019)
42. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8445–8453 (2019)
43. Weng, X., Kitani, K.: Monocular 3d object detection with pseudo-lidar point cloud. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
44. Xu, B., Chen, Z.: Multi-level fusion based 3d object detection from monocular images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2345–2353 (2018)
45. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
46. You, Y., Wang, Y., Chao, W.L., Garg, D., Pleiss, G., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. arXiv preprint arXiv:1906.06310 (2019)
47. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4490–4499 (2018)