Cross-Modality 3D Object Detection

Ming Zhu[1]   Chao Ma[1] [†]   Pan Ji[2]   Xiaokang Yang[1]

[1]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2]NEC Laboratories America

{droplet-to-ocean, chaoma, xkyang}@sjtu.edu.cn, peterji1990@gmail.com

Abstract

In this paper, we focus on exploring the fusion of images and point clouds for 3D object detection in view of the complementary nature of the two modalities, i.e., images possess more semantic information while point clouds specialize in distance sensing. To this end, we present a novel two-stage multi-modal fusion network for 3D object detection, taking both binocular images and raw point clouds as input. The whole architecture facilitates two-stage fusion. The first stage aims at producing 3D proposals through sparse point-wise feature fusion. Within the first stage, we further exploit a joint anchor mechanism that enables the network to utilize 2D-3D classification and regression simultaneously for better proposal generation. The second stage works on the 2D and 3D proposal regions and fuses their dense features. In addition, we propose to use pseudo LiDAR points from stereo matching as a data augmentation method to densify the LiDAR points, as we observe that objects missed by the detection network mostly have too few points especially for far-away objects. Our experiments on the KITTI dataset show that the proposed multi-stage fusion helps the network to learn better representations.

1. Introduction

Object detection on 2D images has made great progress in recent years using deep neutral networks [7, 22]. In contrast, 3D object detection for 3D scene understanding, albeit being crucial and indispensable for many real-world applications such as autonomous driving, still faces great challenges. The most commonly used data in 3D object detection algorithms is point clouds scanned by LiDAR sensors. Although point clouds data could provide precise depth information, on the other hand, they are unordered, sparse, and unevenly distributed. Algorithms using images only for 3D object detection have also been proposed [3, 2, 27, 28, 13], but generally show very poor performance

---
[†]Corresponding author.

compared with the algorithms using point clouds, despite the rich color and semantic information in images. In view of complementary nature of LiDAR and images, we propose a method to deeply fuse both types of data and present an effective way to combine the best of both modalities.

1.1. Challenges

LiDAR points provide abundant geometry information and images possess rich semantic information. [4, 10] propose to first transform point clouds to various views to obtain compact representation, then apply 2D Convolutional Neural Networks(CNN) for feature maps of 2D views and finally fuse them with images feature maps. However, during projection, it will inevitably suffer from information loss. By fusing multi-sensor features of each region of interest(RoI) with predefined anchor boxes, the fusing process becomes very slow due to learning redundant information. [19] proposes to adopt a 2D detector first and conduct 3D object detection for points lying in the predicted frustum. However, the fusion only occurs on the input side, and thus the advantages of both modalities are not fully utilized. Another line of solutions chooses to fuse image and LIDAR feature maps at different levels of resolutions [16, 15]. [16] proposes a single-stage detector by fusing point-wise multi-sensor features but is still subject to the sparsity of points especially for distant objects. [15] utilizes images to produce dense depth as a compensation for sparse points while ignoring the strength of images on 2D object detection which we believe should be coupled together with the 3D object detection.

1.2. Our Contributions

In this paper, we propose a deeply fused multi-modal two-stage 3D object detection framework, taking full advantage of images and point clouds. During the first stage, instead of subdividing point clouds into regular 3D voxels or organising data into different views, we utilize Pointnet++ [21] to directly learn 3D representations from point clouds for classification and segmentation. For binocular images, we apply modified Resnet-50 [8] and FPN [17] as

our backbone network to learn discriminative feature maps for future point-wise feature fusion. The good performance of [19] has shown great power of images on the 3D detection task. Inspired by that, we also propose a new reprojection loss to tightly combine 2D detection and 3D detection together, which benefits the 3D proposals by leveraging the 2D constraints.

To further utilize the ability of multi-modal fusion, we gather point clouds and image areas in proposals generated by the first stage, and then adopt a RoI-wise fusion to associate interior feature maps. At last, a light-weight PointNet is applied to refine final predictions.

By point-wise feature fusion, we can enrich each point with abundant semantic information from binocular images. From RoI-wise feature fusion, we can achieve a robust local proposal representation. With the joint anchor mechanism, we can restrain 3D regression in a well constrained procedure. Besides, we propose to use pseudo LiDAR point clouds to enhance the sparsity of Ground Truth (GT) point clouds, as objects far away from the camera may only possess very few points while images could provide extra geometry information by generating pseudo-lidar point clouds. So we adaptively crop pseudo lidar points and add it to GT point clouds as a data augmentation method which also leads to a considerable improvement to our network.

We summarize the main contributions of this work as follows:

- We propose a two-stage fusion framework to combine the best of binocular image pairs and point clouds for 3D object detection.

- By projecting 3D bounding box to 2D image space, we propose a 2D-3D coupling loss to take full use of image information and constrain the 3D bounding box proposal to conform to the 2D bounding box.

- We compensate the sparsity of point clouds by adding pseudo-lidar points to the real 3D scene.

We evaluate our proposed network on the standard KITTI 2D/3D object detection dataset. By thorough ablation studies, we demonstrate the effectiveness of each novel building block of our method.

2. Related Works

Over recent years, we have witnessed a growing trend of algorithms [10, 4, 11, 14, 18, 23, 31, 1, 12, 26, 25, 24] on the 3D object detection task. These algorithms can be generally grouped in image-based, LiDAR-based and multi-sensor based methods, which we review in detail as follows.

2.1. Image Based 3D Object Detection

There are existing algorithms taking only monocular images as input to generate final 3D bounding boxes. Several works are inspired to explore the power of images on 3D object detection task. [2, 3] formulate the 3D geometric information of objects as an energy function to score the predefined 3D boxes, but meanwhile suffer from the inavailability of depth information. [5] proposes to estimate 3D boxes using the geometry relations between 2D box edges and 3D box corners. [28] formulates an end-to-end multi-level fusion method to predict 3D bounding boxes by concatenating the RGB image and the monocular-generated depth map.

A few other works focus on leveraging binocular images for 3D object detection. [27] proposes to produce the pseudo-LiDAR point cloud by depth predicted from different monocular or binocular depth estimation algorithms and then make use of point cloud-based frameworks to obtain the final prediction. They argue it is not the quality of the data but its representation that accounts for the majority of the difference between 2D images and 3D point clouds. [13] exploits the keypoint and binocular boxes constraints and uses a dense region-based photometric alignment method to ensure 3D localization accuracy.

2.2. Point Cloud Based 3D Object Detection

[30] proposes to represent the scene from the Bird's Eye View (BEV) and apply a single-stage detector that outputs oriented 3D objects. [32] divides a point cloud into equally spaced 3D voxels and transforms a group of points within each voxel to learn descriptive representations. By processing point clouds as voxel input or projecting to various views and applying 2D convolution or 3D convolution to make final prediction, these methods may ineluctably ignore information of one dimension. [23] formulates a different method that takes the whole point cloud as input and predicts final bounding boxes through a two-stage network. Although the LiDAR sensors could offer precise depth information, distant objects still possess very sparse points making it tough for detection while images could provide additional information.

2.3. Multi-sensor Based 3D Object Detection

MV3D [4] projects LiDAR point cloud to BEV to generates proposals, and then fuses BEV features, images features and front view features together to predict final 3D bounding boxes. AVOD [10] proposes a feature fusion Region Proposal Network(RPN) that utilizes multiple modalities to produce positive proposals. These methods still have a limited power when detecting small objects due to the loss of spatial information after projecting point cloud to different views. F-PointNet [19] proposes to generate frustum proposals from 2d object detection and then apply Point-Net [20] based on interior points in each proposal. But the 2D detector and PointNet are two separate cascaded stages and the final results heavily rely on the 2D detection results. [16, 15] utilize continuous convolution to fuse multi-scale
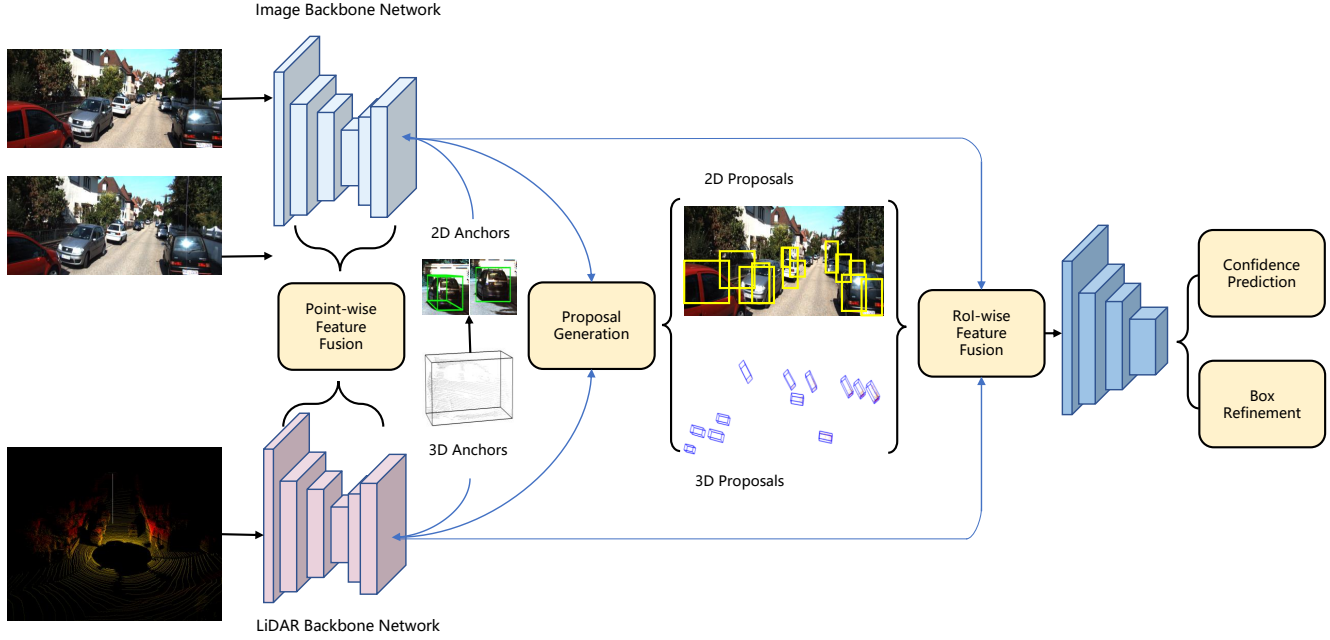
Figure 1. The architecture of our proposed network. It consists of two stages: region proposal stage and box refinement stage. The first stage aims at generating accurate proposals by applying point-wise feature fusion and 2D-3D coupling anchors. The second stage conducts RoI-wise feature fusion to learn more robust representations and predict final confidence and box refinement.

convolutional feature maps from each sensor but ignoring the 2D-3D constraints between images and LiDAR points.

3. Proposed Method

With the development of cameras, the resolution and quality of images have been greatly improved, which enables images to play a more and more important role on the 3D object detection task. But fusing information between LiDAR points and images is a tough process, because Li-DAR points represent the world's native geometry structure while images represent a RGB projection of the world onto the camera plane. Instead of projecting LiDAR points to multi views which leads to information loss, we propose to deeply fuse information between raw points and images in two stages.

The proposed method, depicted in Fig. 1, utilizes binocular images and corresponding LiDAR point cloud as input. In the first stage, the binocular images are passed to a feature extractor to extract their feature maps for following point-wise feature fusion. Given point clouds, we add a classification branch to predict confidence for each point and seed 3D anchors centered on foreground points. Those 3D anchors are then projected to images and adopted for 2D classification and regression. During the second stage, we apply RoI-wise fusion for each pair of 2D and 3D proposals generated by the first stage to learn more robust and discriminative representation for final predictions.

3.1. Feature Extractor

Our proposed architecture for feature extraction is depicted in Fig. 2. For images, we use a modified ResNet-50 [8] as the encoder which takes an image of size $(3, H, W)$ as input and produces a feature map of size $(1024, \frac{H}{32}, \frac{W}{32})$. The output feature map contains high level semantic information but has low resolution which is hard to be leveraged for our point-wise feature fusion. Inspired by FPN [17], a bottom-up decoder is applied to upsample the feature map back to multi-scale. Details are depicted in the upper part of Fig. 2. Feature maps upsampled from the decoder and corresponding feature maps from encoder are concatenated and then passed through a 3x3 convolutional layer. We choose the bilinear interpolation for upsampling.

For point cloud, we use Pointnet++ [21] as our backbone network. Four set-abstraction modules with multi-scale grouping are used to subsample points into groups with sizes of 4096, 1024, 256, 64 and then feature propagation modules are employed to obtain the point-wise feature vectors for segmentation and proposal generation.

3.2. Point-wise Feature Fusion

As shown in Fig. 2, the output multi-scale image feature maps are of size $(B, 1024, \frac{H}{32}, \frac{W}{32})$, $(B, 512, \frac{H}{16}, \frac{W}{16})$, $(B, 256, \frac{H}{8}, \frac{W}{8})$, $(B, 64, \frac{H}{4}, \frac{W}{4})$. Image feature maps of different scales contain different levels of semantic information and receptive fields. Similarly, set abstraction module is designed to build a hierarchical grouping of points
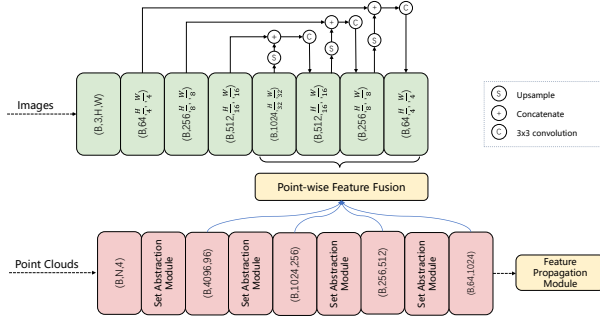
Figure 2. The image and LiDAR backbone networks that we apply in our proposed architecture. The upper part is a modified ResNet-50 and FPN feature extractor. The lower part is the four set-abstraction module and feature propagation module of Point-Net++.

and abstract larger and larger local regions along the hierarchy. Every time points are passed through the set abstraction layer, the number of points in the group reduces and each point is enriched with stronger representation of larger regions. So we propose to apply point-wise fusion at different levels of feature maps to enable points to possess high level semantic information from RGB images. Firstly, we extract XYZ coordinates of points in the four sized groups and project them to image feature maps of its corresponding size. For an example, given coordinates of points in the group with size 1024, we project it to image feature maps with size of $(1024, \frac{H}{32}, \frac{W}{32})$. In the same way, groups with sizes 4096, 256, 64 correspond to image feature maps of size $(B, 1024, \frac{H}{32}, \frac{W}{32}), (B, 256, \frac{H}{8}, \frac{W}{8}), (B, 64, \frac{H}{4}, \frac{W}{4})$.

3.3. Proposal Generation Network

3.3.1 Joint Anchor

Before generating proposals, we need to seed reasonably anchors for the scene. Inspired by [23], we segment the raw point cloud based on the point-wise fused features and generate 3D proposals from the segmented foreground points simultaneously to constrain the search space for 3D proposal generation. For each 3D anchor, its size is predefined as (L=3.9,W=1.6,H=1.5) meters which is obtained form the clustering of training dataset. These 3D anchors are located at the center of each foreground point and then projected to images for 2D anchors producing. Benefited from the projection, there is no need to seed extra 2D anchors of different ratios for images because it's scale-adapted benefited from projection. With the anchor generation mechanism, 2D and 3D regions are tightly connected for following classification and regression.
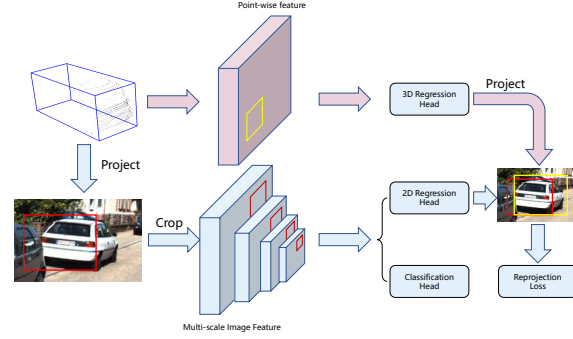

Figure 3. The process of reprojection with joint anchor mechanism.

3.3.2 Joint Proposal

With 2D-3D anchor pairs mentioned above, we propose a joint proposal generation scheme that generates both 2D and 3D bounding box proposals simultaneously.

A 3D bounding box is represented as $(x_p, y_p, z_p, h_p, w_p, l_p, \theta)$ in the LiDAR coordinate system, where (x_p, y_p, z_p) is the object center location, (h_p, w_p, l_p) is the object size, and θ is the object orientation from the bird's eye view. A 2D bounding box is represented as (x_i, y_i, h_i, l_i), where (x_i, y_i) is the 2D bounding box center and (h_i, l_i) is the box size. The projection from a point $\mathbf{x}$ in velodyne coordinate system to image coordinates $\mathbf{y}$ follows:

$$\mathbf{y} = \begin{pmatrix} f_u^{(i)} & 0 & c_u^{(i)} & -f_u^{(i)} b_x^{(i)} \\ 0 & f_v^{(i)} & c_v^{(i)} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R}_{rect}^{(0)}(\mathbf{R}_{velo}^{cam}\mathbf{x} + \mathbf{t}_{cam}^{velo}) \\ 1 \end{pmatrix},$$

(1)

where $f_{\cdot}^{(i)}, c_{\cdot}^{(i)}$ and $b_x^{(i)}$ are intrinsic parameters of camera i, $\mathbf{R}_{rect}^{(0)}$ is rectifying rotation matrix of camera 0, $\mathbf{R}_{velo}^{cam}$ is rotation matrix, and $\mathbf{t}_{cam}^{velo}$ is translation vector.

For the classification prediction branch, we force 2D-3D anchor pairs to share the same confidence score. Given the point-wise fused features, we append one segmentation head for estimating foreground mask. Then 3D anchors are seeded at the center of each foreground point and projected to images to perform 2D classification prediction. For the 2D anchors, the one with IoU less than 0.3 are considered background anchors, while the one with IoU greater than 0.5 are considered as foreground anchors during training. We can further reduce the number of anchors while not lowering the quality of proposals by keeping only foreground anchors verified by both 2D and 3D classification. We use the binary cross entropy loss for 2D segmentation and the focal loss for 3D segmentation as

$$L_{cls\_2D} = -\left(\frac{1}{N_{pos}} \sum log(p^{pos}) + \frac{\alpha}{N_{neg}} \sum log(1 - p^{neg})\right),$$
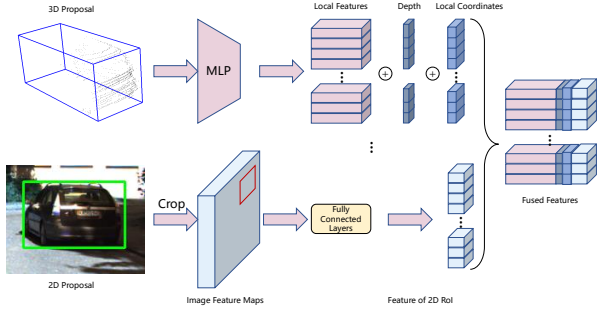
(2)

Figure 4. RoI-wise feature fusion module for robust and dense representation learning.

$$L_{cls\_3D} = -\beta(1-P)^\gamma log(P) \ , \qquad (3)$$

where P means the possibility of foreground points.

Since the 2D and 3D anchor pairs are naturally produced by our joint anchoring mechanism, we hope to dig deeper to make full use of their inter-connection. As mentioned before, images represent a RGB projection of the world onto the camera plane, and thus 2D anchors correspond to the frustum area of the real world. By regressing 2D proposals, we are regressing a frustum area to its Ground Truth regions. Although it cannot provide precise 3D locations for 3D proposals, it's still able to offer coarse directions for 3D regression. So we propose a reprojection loss to perform better proposal generation.

Firstly, we identically deliver 2D and 3D anchors to respective heads for their own regression. For 3D anchors, we compute center offsets $(\Delta x_1, \Delta y_1, \Delta z_1)$, predefined size offsets $(\Delta l_1, \Delta w_1, \Delta h_1)$ and the orientation θ. For 2D anchors, we compute 2D center offsets $(\Delta x_2, \Delta y_2)$ and predefined size offsets $(\Delta x_2, \Delta h_2)$. Then we obtain regressed 3D anchors $(x'_p, y'_p, z'_p, h'_p, w'_p, l'_p, \theta')$ and 2D anchors (x'_i, y'_i, h'_i, l'_i). Afterwards, we project regressed 3D anchors onto images to generate 2D anchors of size $(x''_p, y''_p, h''_p, l''_p)$. So we can compute our reprojection loss between (x'_i, y'_i, h'_i, l'_i) and $(x''_p, y''_p, h''_p, l''_p)$, which couples the 2D/3D boxes tightly together. Our experimental results (see Sec. 4.2) show the reprojection loss improves the quality and recall of the proposals. Our regression loss is composed of three parts as

$$L_{reg} = L_{reg}^{2D} + L_{reg}^{3D} + \alpha L_{reg}^{reprojection} \ . \qquad (4)$$

3.4. Box Refinement

3.4.1 Proposal-wise Feature Fusion

After obtaining high quality 3D bounding box proposals by the first stage, we aim at further refining the box loca-

tions and orientations for final predictions during the second stage.

Although we have conducted point-wise fusion to enrich points with high level semantic information, it's only point-to-pixel level fusion, which may be still too sparse for learning specific local features of each proposal. So we propose to apply RoI-wise feature fusion to learn denser representation for refinement, which is a region-to-region level as depicted in Fig. 4. For 3D proposals, following [23], we transform the points belonging to each proposal to the canonical coordinate system of the corresponding 3D proposal. Meanwhile, we also include the distance of each point to the sensor as a compensation for the loss of depth information. Then the interior points of canonical coordinate system are passed to several MLPs to encode local point features. For 2D proposals we firstly extract image feature maps from the last layer of the multi-scale features. As downsampling leads to low resolution of feature maps, the increase in receptive field caused by convolutions makes it hard for fusion of objects with very sparse LiDAR points. After obtaining feature maps of 2D regions, we deliver it through several fully-connected layers to encode to the same dimension of 3D local point features. At last, we concatenate local point features, local point coordinates, depth information and local 2D region features together to form a strong representation.

3.4.2 Final Prediction

Given the high quality 3D proposals and corresponding representative fused features, we adopt a light-wight Point-Net [20] consisting of two MLP layers to encode the features to a discriminative feature vector. Final confidence classification and 3D box refinement are achieved by two MLP layers.

3.5. Pseudo-LiDAR Fusion

Due to the sparseness of the point cloud, some objects we need to detect may contain only a very small number of points, so we hope to improve the detection performance by compensating this sparseness via pseudo-LiDAR point clouds. Through binocular images, we can predict pixel-wise disparity through stereo matching, so that the corresponding pseudo-LiDAR point of each pixel can be obtained by inverse projection. For an instance, a highly occluded or distant object of size 10x10px in images only possess 10 points in 3D scene, but we can produce 100 pseudo points by predicting pixel-wise depth.

Since we want to focus on the target that needs to be detected, we need to remove other unnecessary points to reduce the noise in the Pseudo-LiDAR points. During the training stage, we only fuse the pseudo point clouds within the GT 2D boxes. During the inference stage, we instead

Table 1. Performance comparison of 3D object detection with previous methods on the KITTI test split by submitting the results to the official test server. The evaluation metric is Average Precision(AP) with IoU threshold 0.7 for Cars.

Method	Modality	3D AP(%)			2D AP(%)		
		Easy	Moderate	Hard	Easy	Moderate	Hard
MV3D[4]	RGB+LiDAR	71.09	62.35	55.12	-	-	-
AVOD[10]	RGB+LiDAR	73.59	65.78	58.38	95.17	89.88	82.83
AVOD-FPN[10]	RGB+LiDAR	81.94	71.88	66.38	94.70	88.92	84.13
F-PointNet[19]	RGB+LiDAR	81.20	70.39	62.19	95.85	**95.17**	85.42
ContFuse[16]	RGB+LiDAR	82.54	66.22	64.04	-	-	-
VoxelNet[32]	LiDAR	77.49	65.11	57.73	-	-	-
Second[29]	LiDAR	83.13	73.66	66.20	93.72	90.68	85.63
PointPillars[11]	LiDAR	82.58	74.31	68.99	94.00	91.19	88.17
PointRCNN[23]	LiDAR	86.96	75.64	70.70	95.92	91.90	87.11
Ours	RGB+LiDAR	**87.22**	**77.28**	**72.04**	**96.21**	93.45	**88.68**

resort to fusing Pseudo-LiDAR points within the predicted 2D boxes. Due to the error of the depth estimation, the resulting pseudo point cloud is inaccurate, including long tails and local misalignment.The long tail effect is mainly due to the inaccurate depth prediction at the edge of the object, for which we design a point cloud statistical filter to filter out all the outliers. The local misalignment effect is caused by the error in the overall depth prediction. Although the predicted point clouds are similar in shape to the original objects, there will be a certain forward or backward deviation in the overall depth, so we use the GT point clouds to rectify the pseudo point clouds. For each predicted 2D bounding box, we extract corresponding GT point cloud P_G and pseudo point cloud P_P, and then calculate the average distance D_i from per pseudo point P_{P_i} to the nearest K GT points. Then we move the point cloud as a whole by a certain distance to minimize the sum of all distances $\sum D_i$ to achieve depth correction.

4. Experimental Results

We evaluate our proposed 3D object detector on the public KITTI [6] benchmark and compare it with previous state-of-the-art methods in both 3D object detection and 2D object detection tasks. Extensive ablation study is also conducted which evaluates how different components affect our model.

4.1. Experiment Setup

4.1.1 Dataset and Metric

The KITTI object detection benchmark provides 7481 training frames and 7518 testing frames but only offers labels for training frames in order to prevent overfitting. Since the access to the ground truth for the test set is not available, we follow the official setup to split the training samples into a training set consisting of 3712 frames and a validation set consisting of 3769 frames. The mean Average Precision (mAP) is utilized as our evaluation metric following the official evaluation protocol.

4.1.2 Implement Details

For our image backbone network, we resize binocular images to $(600, 2000)$ and feed them to extract multi-scale features simultaneously. For the LiDAR backbone network, the point cloud is firstly cropped to the range of [0., 70.]x[-40., 40.]x[-3., 1.] meters along (X, Y, Z) axes respectively, following [32, 4]. During training, a proposal is considered as positive if its maximum 3D IoU with ground-truth boxes is above 0.6 and negative below 0.45. We use 3D IoU 0.55 as the minimum threshold of proposals for the training of box regression head. Nonmaximum suppression (NMS) with IoU thread 0.85 is applied to remove the redundant proposals. We keep top 9000 proposals for regression in stage one and top 300 proposals in stage two during training.

4.1.3 Training Details

We train our models for 200 epochs with batch size 8 for stage one and 70 epochs with batch size 1 for stage two on one GTX 2080Ti GPU. We use the ADAM [9] optimizer with an initial learning rate 0.002 for the first 150 epochs and then decayed by 0.1 in every 10 epochs during the first stage, while the second stage is trained for 70 epochs with batch size 1 and learning rate 0.002. Considering the limited amount of training data, we also conduct point clouds data augmentation of random flipping, random scaling with a uniformly sampled scale from 0.95~1.05 and random rotation with a degree sampled from $-45° \sim 45°$ to allevi-

Table 2. Ablation study on KITTI object detection benchmark training set with four-fold cross validation for Car class.

Model	3D AP(%)			2D AP(%)		
	Easy	Moderate	Hard	Easy	Moderate	Hard
LiDAR only	83.25	76.65	73.91	94.22	86.39	83.15
+ Point-wise Fusion	+1.91	+1.37	+0.22	+1.19	+0.53	+0.12
+ RoI-wise Fusion	+2.82	**+3.73**	+2.75	+2.43	+2.57	+2.16
+ Reprojection Loss	**+4.07**	+3.29	**+3.26**	**+2.81**	**+3.15**	**+5.44**
+ Binocular Images	+0.02	+0.49	+0.55	+0.17	+0.75	+0.97
Full Model	91.08	83.19	77.12	98.43	93.85	90.39

Table 3. The number of mispredicted 2D object detection results on validation set.

Class	Before	After
Easy	83	**27**
Moderate	3672	**713**
Hard	906	**104**

Table 4. Recall of proposals with different numbers of ROIs before and after applying 2D-3D reprojection loss for the car class on the val split at moderate difficulty.

RoI	Before(IoU=0.7)	After(IoU=0.7)
50	30.59	**40.91**
100	68.04	**71.66**
150	70.66	**74.45**
200	73.23	**76.03**
250	77.79	**80.87**
300	79.55	**82.13**

ate the overfitting problem. Since per-object augmentations cannot be applied in the camera images, such augmentation strategies are used only for the LiDAR backbone pre-training process.

4.1.4 Main Results

As shown in Table 1, we compare our model with state-of-the-art approaches in 3D object detection and 2D object detection on the KITTI test dataset. Through deep fusion of images and point clouds, we have the best AP on the easy subset for 2D object detection, and our model outperforms all other methods measured by moderate and hard APs for 3D object detection. We show qualitative results in Figure 5.

4.2. Ablation Study

To further analyze the ability of our proposed deeply fused multi-modal 3D object detection method, we conduct extensive ablation studies on the KITTI train/val set to explore the effects of our components. We use the official training and validation split and accumulate the evaluation results over the whole training set. The ablation study results are shown in Table 2. Our baseline model only uses LiDAR as input without any fusion with images.

4.2.1 Effects of Multi-modal Feature Fusion

Results in Table 2 confirm that feature fusion over two stages helps improve the performance of our model in different degrees. By applying point-wise fusion, it brings 1.91%/1.37%/0.22% AP gain in 3D detection of easy, moderate and hard class respectively. It's a reward of providing every LiDAR points with different level 2D semantic information as a compensation for XYZ coordinates. But we can observe that there is less improvement for the hard class where the objects usually have fewer LiDAR points. It's most likely because that the fusion is in a point-wise level, so it's naturally unfair for those objects owning fewer LiDAR points to get more information from RGB images. That is why we propose to apply RoI-wise fusion during stage two. RoI-wise feature fusion improves 3D detection by 2.82%/3.73%/2.75%AP in easy/moderate/hard class respectively. Because the fusion is conducted in a region-to-region level, we can achieve a stable improvement in all three classes. This proves the dense image features are beneficial for learning robust and discriminative representation of local proposals.

4.2.2 Effects of Reprojection Loss

As is shown in Table 2, the 2D-3D reprojection loss plays the most important role for both 2D object detection and 3D object detection. We analyze that 2D regression and 3D regression are closely coupled together and mutually reinforcing. During 2D regression, 3D proposals could provide supervision from a higher dimension. During 3D regression, 2D regression could offer a coarse region where the 3D regression process must lie in. They complement each other and lead to a win-win situation. We also conduct extensive

Table 5. Results of 3D object detection in AP with IoU=0.7 for Cars using different types of fusion with Pseudo LiDAR Points on the validation split.

Data Source	Easy	Moderate	Hard
Without Fusion	91.08	83.19	77.12
By 2D boxes	90.27	84.51	77.94
By 3D boxes	**92.19**	**86.80**	**79.62**

experiments to explore its performance as shown in Table 3. We count the numbers of mispredicted objects which means those predictions have no interaction area(IoU=0) with GT 2D bounding boxes. There is an obvious decline we can observe from the results. It proves that we can rectify the final results by precise 2D information provided by images. We also calculate the recall of 3D bounding boxes with various numbers of proposals before and after applying the reprojection loss. We can achieve different gains as shown in Table 4, which reveals the proposed loss is helpful to proposals to regress to an ideal position. Our loss is complementary to other methods which take LiDAR and images as input.

4.2.3 Effects of Binocular Images

By taking binocular images as input, our model obtains a slight improvement of all classes. In most cases, monocular images can provide enough information we desire for objects, but in some extreme occasions, it can offer extra cues which are critical for detection. For an example, a highly or completely occluded car pictured by the left camera may get a clear view from the right camera. With the availability of binocular images, we will suffer less from the occlusions.

4.2.4 Effects of Pseudo LiDAR Points

We conduct two-fold cross validation and the results in Table 5 verify that fusion with Pseudo LiDAR Points could bring improvements for the final results at different levels. There are improvements we can observe for the moderate and hard classes by fusing pseudo points cropped by 2D predictions. We notice that there is an accuracy drop for the easy class, which is possibly because objects of easy class possess enough number of points for network and cannot squeeze much juice from pseudo points. In order to explore the upper limit of the fusion, we fuse the point cloud extracted through the GT 3D bounding boxes, which is the purest noise-free point cloud. We use pseudo points lying in GT 3D bounding boxes during training and validation. We can obtain a great gain by the fusion especially for objects of hard class which benefit most from denser input points. The experimental results verify pseudo LiDAR points could enhance the performance of network by this method of fusion. With the improvement of photography technology, we can obtain pictures which would be influenced less by the light and possess higher resolution. By that time, we can generate more preciser pseudo LiDAR points which could play a vital role in future image-based 3D object detection task.

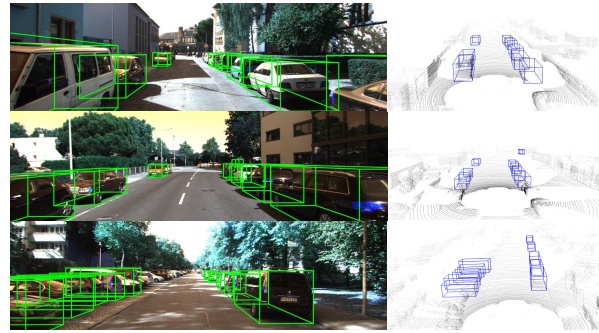### 4.3. Qualitative Results and Discussion

Figure 5. Visualization of final results.

We show qualitative 3D object detection results of the proposed detector on the KITTI benchmark in Figure 5. We can observe that the predicted 3D bounding boxes fit tightly to each object, even for the distant and highly occluded cars.

5. Concluding Remarks

In this paper, we have proposed an effective deeply fused multi-model framework for 3D object detection that aims at exploring the association between LiDAR and images to perform precise 3D localization. Our approach is realized by feature fusing during two stages. The first stage focuses on acquiring high quality 3D proposals for the next stage by taking raw LiDAR points and images as input and applies point-wise fusion with 2D-3D reprojection constraints in a sparse method. The second stage concentrates on learning representative and robust features of each proposals in a dense way to predict final refined 3D bounding boxes. We also propose to leverage pseudo LiDAR points as an augmentation for the sparse point clouds. Our experimental results show our approach outperforms previous baselines and is capable to combine both LiDAR's and images' strengths.

6. Acknowledgement

References

[1] Waleed Ali, Sherif Abdelkarim, Mahmoud Zidan, Mohamed Zahran, and Ahmad El Sallab. Yolo3d: End-to-end real-time 3d oriented object bounding box detection from lidar point cloud. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[2] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016.

[3] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2015.

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[5] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. 2017.

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[7] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[11] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[12] Bo Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518. IEEE, 2017.

[13] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7644–7652, 2019.

[14] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018.

[15] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019.

[16] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018.

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[18] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018.

[19] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[20] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[21] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[23] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[24] Kiwoo Shin, Youngwook Paul Kwon, and Masayoshi Tomizuka. Roarnet: A robust 3d object detection based on region approximation refinement. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2510–2515. IEEE, 2019.

[25] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. In *European conference on computer vision*, pages 634–651. Springer, 2014.

[26] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016.

[27] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. 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, pages 8445–8453, 2019.

[28] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2345–2353, 2018.

[29] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[30] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[31] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Ipod: Intensive point-based object detector for point cloud. *arXiv preprint arXiv:1812.05276*, 2018.

[32] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.