RangeRCNN: Towards Fast and Accurate 3D Object Detection with Range Image Representation

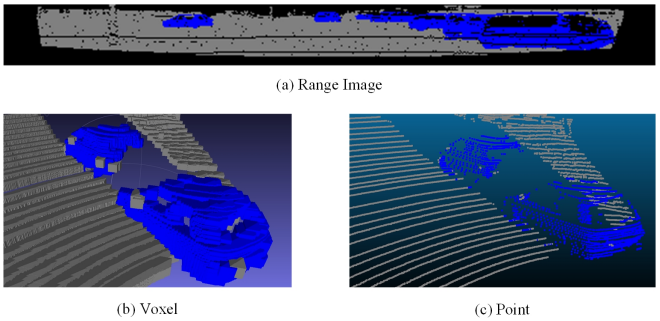
Zhidong Liang[1], Ming Zhang[1], Zehan Zhang[1], Xian Zhao[1] and Shiliang Pu[1]

***Abstract*— We present RangeRCNN, a novel and effective 3D object detection framework based on the range image representation. Most existing 3D object detection methods are either voxel-based or point-based. Though several optimizations have been introduced to ease the sparsity issue and speed up the running time, the two representations are still computationally inefficient. Compared to these two representations, the range image representation is dense and compact which can exploit the powerful 2D convolution and avoid the uncertain receptive field caused by the sparsity issue. Even so, the range image representation is not preferred in 3D object detection due to the scale variation and occlusion. In this paper, we utilize the dilated residual block to better adapt different object scales and obtain a more flexible receptive field on range image. Considering the scale variation and occlusion of the range image, we propose the RV-PV-BEV (Range View to Point View to Bird's Eye View) module to transfer the feature from the range view to the bird's eye view. The anchor is defined in the BEV space which avoids the scale variation and occlusion. Both RV and BEV cannot provide enough information for height estimation, so we propose a two-stage RCNN for better 3D detection performance. The point view aforementioned does not only serve as a bridge from RV to BEV but also provides pointwise features for RCNN. Extensive experiments show that the proposed RangeRCNN achieves state-of-the-art performance on the KITTI 3D object detection dataset. We prove that the range image based methods can be effective on the KITTI dataset which provides more possibilities for real-time 3D object detection.**

I. INTRODUCTION

In recent years, 3D object detection has been paid more and more attention to in many fields. The well-studied 2D object detection can only tell the object position in the 2D pixel space instead of the 3D physical space. However, the 3D information is extremely important for several applications, such as autonomous driving. Compared to 2D object detection, 3D object detection remains challenging since the point cloud is irregular and sparse. The suitable representation for the 3D point cloud is worthy of research.

Existing methods are mostly divided into two categories: the grid-based representation and the point-based representation. The grid-based representation can further be classified into two classes: 3D voxels and 2D BEV (Bird's Eye View). Such representations can utilize the 3D/2D convolution to extract features, but simultaneously suffers from the information loss of quantization. The 3D convolution is not efficient and practical in large outdoor scenes even though several optimizations have been proposed [1], [2]. The 2D BEV suffers from a more severe information loss than the 3D voxel which limits its performance. The point-based representation retains more information than the voxel-based methods. But the point-based methods are generally inefficient when the number of points is large. Downsampling points can reduce the computation cost but simultaneously degrades the localization accuracy. In summary, both of the two representations cannot retain all original information for feature extraction while being computationally efficient.



(a) Range Image

(b) Voxel

(c) Point

Fig. 1. Different representations of point clouds. (a) Range image representation (dense). Use 2D convolution to extract features. (b) 3D Voxel Representation (sparse). Use 3D convolution to extract features. (c) Point Representation (sparse). Use point-based convolution to extract features.

Although we mostly regard the point cloud as the raw data format, the range image is the native representation of the rotating LIDAR sensor (e.g. Velodyne 64E, etc). It retains all original information without any loss. Beyond this, the dense and compact properties make it efficient to process. Fig. 1 shows the three representations of point clouds. As a result, we think that it is beneficial to extract features from the range image. Several methods [3], [4] directly operates on the range image, but have not achieved similar performance as the voxel-based and point-based methods. [4] attributes the unsatisfied performance to the small size of the KITTI dataset which makes it difficult to learn from the range image, and conducts the experiment on their private dataset to prove the effectiveness of the range image. In this paper, we present a range image based methods and prove that the range image representation can also achieve state-of-the-art performance in the KITTI dataset.

Though several advantages of the range image are pointed out above, its essential drawbacks are also obvious. The large scale variation makes it difficult to decide the anchor size in the range view and the occlusion makes the bounding boxes easily overlap with each other. The two issues do not

*Shiliang Pu is the corresponding author.

[1]Zhidong Liang, Ming Zhang, Zehan Zhang, Xian Zhao, Shiliang Pu are with Hikvision Research Institute, Hangzhou Hikvision Digital Technology Co. Ltd, China (e-mail: liangzhidong@hikvision.com; zhangming15@hikvision.com; zhangzehan@hikvision.com; zhaoxian@hikvision.com; pushiliang.hri@hikvision.com)

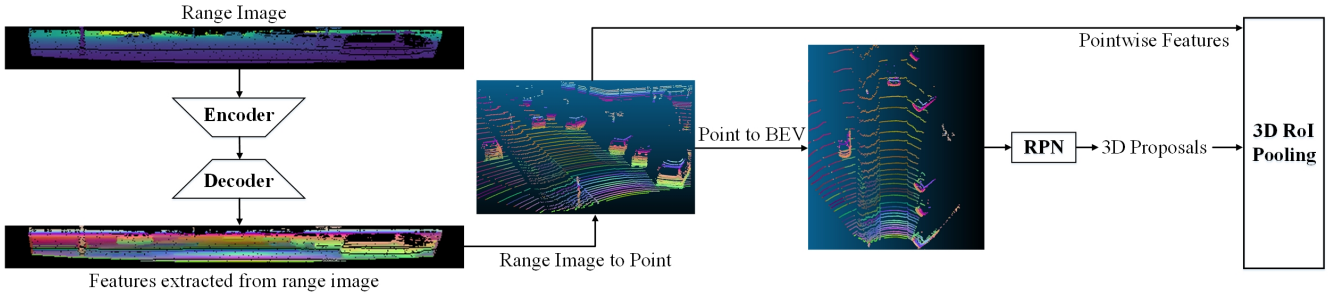


Fig. 2. Illustration of the framework of RangeRCNN. The range image is visualized using the pseudo color according to the range value. After the range image backbone, the features extracted from the range image are transferred to the point view and the bird's eye view in turn. The region proposal network (RPN) is used to generated 3D proposals from the BEV. The 3D proposal and the pointwise feature are input into the 3D RoI pooling for proposal refinement.

exist in the 2D BEV space. Considering these properties, we propose a novel framework named RangeRCNN. First, we extract features from the range image for its compact and lossless representation. To better adapt the scale variation of the range image, we utilize the dilated residual block which using the dilated convolution [5], [6] to achieve a more flexible receptive field. Then, we propose the RV-PV-BEV module to transfer the feature extracted from the range view to the bird's eye view. As the high-level feature is well extracted, the influence of the quantitative error caused by the BEV is not great anymore. The BEV mainly plays the role of anchor generation. Neither the range image nor the bird's eye image cannot explicitly supervise the height of the 3D bounding box. As a result, we propose to refine the 3D bounding box using a two-stage RCNN. The point view in the RV-PV-BEV module does not only serve as the bridge from RV to BEV, but also provides pointwise features for RCNN refinement.

In summary, the key contributions of this paper are as follows:

- We propose RangeRCNN framework which takes the range image as the initial input to extract dense and lossless features for fast and accurate 3D object detection.
- We design a 2D CNN utilizing the dilated convolution to better adapt the flexible receptive field of the range image.
- We propose the RV-PV-BEV module for transferring the feature from the range view to the bird's eye view for easier anchor generation.
- We propose an end-to-end two-stage pipeline that utilizes a region convolutional neural network (RCNN) for better height estimation. The whole network does not use 3D convolution or point-based convolution which makes it simple and efficient.
- Our proposed RangeRCNN achieves state-of-the-art performance on the competitive KITTI 3D detection benchmark.

II. RELATED WORK

A. 3D Object Detection

3D Object Detection with Grid based Methods. Most state-of-the-art methods in 3D object detection project the point clouds to the regular grids. [7]–[10] directly projects the original point clouds to the 2D bird's eye view to utilize the efficient 2D convolution for feature extraction. They also combine RGB images and other views for deep feature fusion. [11] is a pioneer in 3D voxel based object detection. Based on [11], [1] increases the efficiency of the 3D convolution using the sparse optimization. Several following methods [12]–[14] utilize the sparse operation [1], [2], [15] to develop more accurate detectors. For real-time 3D object detector, [16] proposes the pillar-based voxel to significantly improve the efficiency. However, the grid-based methods suffer from the information loss in the stage of initial feature extraction. The sparsity issue of the point clouds also limits the effective receptive field of 3D convolution. These problems will be more serious if processing the large outdoor scene.

3D Object Detection with Point based Methods. Compared to the grid-based methods, the point-based methods are limited by the high computation cost in early researches. [17], [18] project 2D bounding boxes to the 3D space to obtain 3D frustums and conduct the 3D object detection in each frustum. [19], [20] directly process the whole point cloud using [21] and generate proposals in a bottom-up manner. [22] introduces a vote-based 3D detector which is more suitable to process indoor scenes. [23] uses graph neural network for point cloud detection. [24] proposes a fusion sampling strategy to speed up the point-based method.

3D Object Detection with Range Image. Compared to the grid-based and point-based methods, fewer researches utilize the range image in 3D object detection. [7] takes the range image as one of its inputs. [3], [4] directly process the range image for 3D object detection. These methods have not matched the performance of the grid-based or point-based methods. We think that the main reason for this phenomenon is that the range image is a good choice for extracting initial features, but not a good choice for generating anchors. In this paper, we design a better framework utilizing the range

image representation for 3D object detection.

B. 3D Semantic Segmentation

3D semantic segmentation task is chosen by many methods [2], [15], [21], [25]–[29] as the touchstone for evaluating the ability to extract features from point clouds. Early researches mainly focus on indoor scenes due to the lack of outdoor datasets. SemanticKITTI [30] is a recent semantic segmentation benchmark for autonomous driving scenes. In the benchmark, [31] prove the effectiveness of extracting features from the range image and simultaneously runs at a high speed. We believe that the range image can also provide rich and useful information for the 3D object detection task.

III. METHOD DESCRIPTION

In section III-A, we first present the whole network architecture of our method. In section III-B, we introduce the backbone for extracting features from the range image. In section III-C, we introduce the RV-PV-BEV module for transferring features between different views. In section III-D, we utilize the 3D RoI pooling to refine the generated proposals. In section III-E, we describe the loss function used in our network.

A. Network Architecture

Our proposed network is illustrated in Fig. 2. The 3D point cloud is represented as the native range image which is fed into an encoder-decoder 2D backbone to efficiently and effectively extract features. We upsample the deep feature map to the original resolution of the range image using a decoder for retaining more spatial information. Then we transfer features from the range image to each point. We do not extract features based on the point view using the point-based convolution [21], [25]. Actually, the point view has two functions. First, it serves as the bridge from the range image to the bird's eye image. Second, it provides the pointwise feature to the 3D RoI pooling module for refining the proposals generated by the region proposal network (RPN). After obtaining the pointwise feature, we can easily get the BEV feature by projecting the 3D point to the x-y plane. Since we have well extracted high-level features from the range image, the BEV mainly plays the role of proposal generation. So this projection is different from projecting the 3D point to the BEV at the beginning which is extremely dependent on the feature extraction from the BEV. We use a simple RPN to generate proposals from the BEV and refine the proposals using the 3D RoI Pooling module. We name the one-stage network without the 3D RoI pooling RangeDet, and name the whole two-stage framework RangeRCNN.

B. Range Image Backbone

The KITTI dataset provides the point cloud as the LIDAR data format, so we need to convert the points to the range image via spherical projection. As described in [31], the conversion formula is as follows:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2}[1 - arctan(y,x)\pi^{-1}] \times w \\ [1 - (arcsin(z,r) + f_{down})f^{-1}] \times h \end{pmatrix} \quad (1)$$

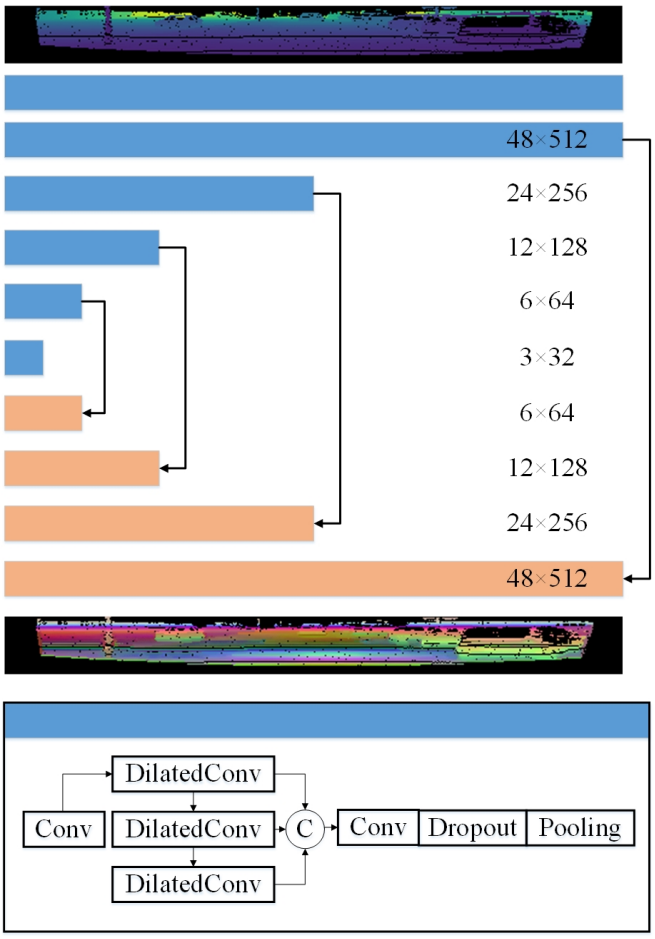


Fig. 3. Illustration of the backbone for the range image. The encoder-decoder structure is adopted. We utilize the dilated convolution for better feature extraction.

where (x, y, z) is the point coordinate in the 3D space. (u, v) is the pixel coordinate in the range image. $r = \sqrt{x^2 + y^2 + z^2}$ is the range of each point. w and h are the predefined width and height of the range image. $f = f_{up} + f_{down}$ is the vertical field-of-view of the LIDAR sensor. For each pixel position, we encode its range, coordinate, and intensity as the input channel. As a result, the size of the input range image is $5 \times h \times w$. In [31], the categories of semantic segmentation are labeled for all points, so the range image contains the 360-degree information. The LIDAR used by the KITTI dataset is the Velodyne 64E LIDAR with 64 vertical channels. Each channel generates approximately 2000 points. So in their task, h and w are set as 64 and 2038 respectively. In the KITTI 3D detection task, the LIDAR and the camera are jointly calibrated. Only the objects in the front view of the camera are labeled which contains approximately 90-degree of the whole scene. Also, some vertical channels are filtered by the FOV of the camera. So we set $h = 48$ and $w = 512$ which are already enough to contain the front view scene. The size of the range image used in this paper is $5 \times 48 \times 512$.

The range image provides dense and compact representation for utilizing the 2D CNN but simultaneously brings

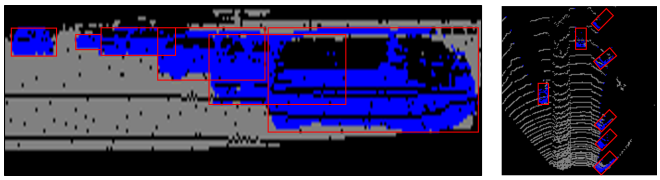


Fig. 4. Comparison of the range image and the bird's eye image. The bounding boxes in the range image differs a lot in the scale and easily overlap with each other. In contrast, the bounding boxes in the bird's eye image maintain a similar size and do not have overlapping areas.

the scale variation issue. The scale of objects with different distances has significant differences. To better adapt different scales and obtain a more flexible receptive field, we insert the dilated convolution into the normal residual block. In each dilated residual block, we first use a 1×1 convolution to extract features across channels. Then three 3×3 convolutions with different dilated rates are applied to extract features with different receptive fields. Then the three branches are concatenated followed by a 1×1 convolution to fuse the features from the three branches. The dilation rates of the three dilated convolutions are set as $\{1, 2, 3\}$. The dilation rate of 1 indicates normal convolution. Then the dropout operation and the pooling operation are used for better generalization performance and downsampling the feature map, respectively. The structure of the dilated residual block is illustrated in Fig. 3. The 2D backbone for extracting features from the range image is an encoder-decoder structure. In each layer of the encoder, the feature is extracted by the dilated residual block. In the first two blocks of the encoder, we do not use the pooling operation. In the decoder, we use a similar block to fuse features from the last layer and the corresponding layer in the encoder. We remove the pooling operation and add the bilinear interpolation operation in the decoder. Finally, we output the high-level features with the same resolution as the input range image. The output feature dimension is 64. We visualize the output feature in Fig. 3 by t-SNE dimension reduction.

C. RV-PV-BEV Module

The range image representation is suitable for feature extraction by utilizing the 2D convolution. However, it is difficult to assign anchors in the range image plane due to the large scale variation. The severe occlusion also makes it difficult to remove redundant bounding boxes in the Non-Maximum Suppression (NMS) module. Fig. 4 shows a typical example. The size of the bounding boxes varies greatly in the range image. Some bounding boxes have a large overlap. In contrast, these bounding boxes have a similar shape in the BEV plane because most cars are of a similar size. It is also impossible for different cars to overlap with each other in the BEV place even though they are very close. So we think that it is more suitable to generate anchors in the BEV plane. Thus, we transfer the feature extracted from the range image to the bird's eye image.

For each point, we record its corresponding pixel coordinate in the range image plane, so we can obtain the pointwise feature by indexing the output feature of the range image backbone. Then, we project the pointwise feature to the BEV plane. For points falling into the same pixel in the BEV image, we use the average pooling operation to generate the representative feature for the pixel. Here the point view only serves as the bridge to transfer features from the range image to the BEV image. We do not use the point-based convolution to extract features from points.

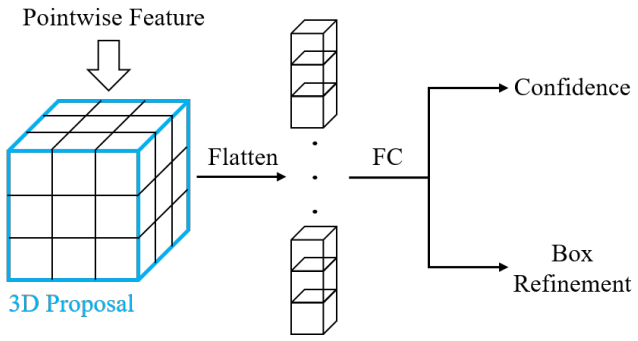


Fig. 5. Illustration of the 3D RoI pooling. The blue box is the 3D proposal. It is divided into the regular grids along three aligned axes. Each grid obtains the feature from the point view aforementioned. The max pooling operation is applied to pool multiple point features within a grid. All 3D grids are flattened to a vectorized feature. Several fully connected layers are applied to predict the refined boxes and the confidences.

Discussion. Different from projecting point clouds to the BEV plane at the beginning of the network [7], [8], we do the projection after extracting high-level features. If projecting at the beginning, the BEV serves as the main feature extractor. The information loss caused by the discretization leads the inaccurate features. In our framework, the BEV mainly plays the role of anchor generation. As we have extracted features from the lossless range image, the quantization error caused by the discretization has a minor influence. Experiments also show the superiority of our methods compared with those methods projecting at the beginning.

D. 3D RoI Pooling

Based on the bird's eye image, we generate 3D proposals using the region proposal network (RPN). However, neither the range image nor the bird's eye image does not explicitly learn features along the height direction of the 3D bounding box, which causes our predictions to be relatively accurate in the BEV plane, but not in the 3D space. As a result, we want to explicitly utilize the information of the 3D space. We conduct a 3D RoI pooling based on the 3D proposals generated by RPN similar to [12], [32]. The proposal is divided into a fixed number of grids. Different grids contain different parts of the object. As these grids have a clear spatial relationship, the height information is encoded among the positions of these grids. We directly vectorize these grids from three dimensions to one dimension sorted by their 3D positions (illustrated in Fig. 5). We apply several fully connected layers to the vectorized features and predict the refined bounding boxes and the corresponding confidences. We do not use the 3D convolution or other

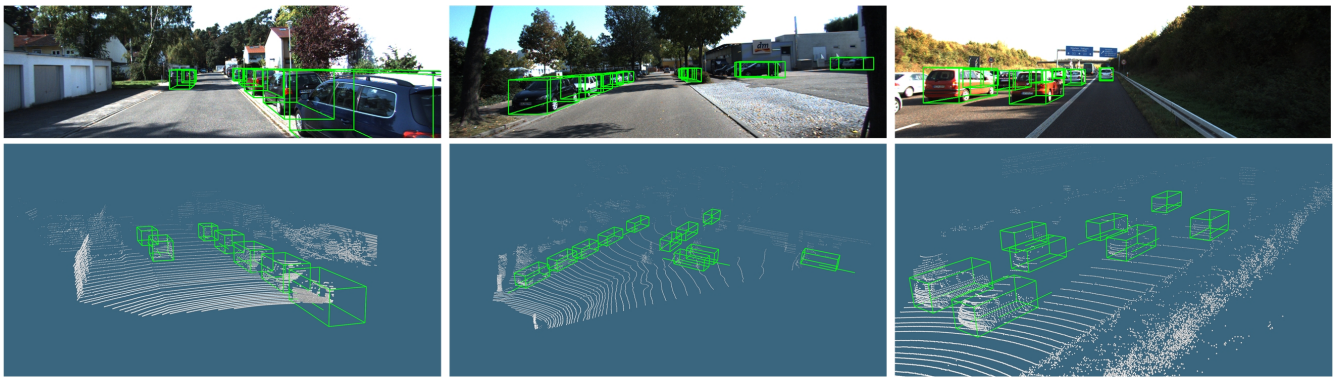


Fig. 6. Visualization of our predictions on the KITTI dataset.

complex operations, which makes the pipeline simpler.

E. Loss Function

The whole network is trained with an end-to-end fashion. The loss function contains two main parts: the region proposal network loss L_{rpn} and the region convolutional neural network loss L_{rcnn}, which is similar with [1], [12], [14]. The RPN loss L_{rpn} includes the focal loss L_{cls} for anchor classification, the smooth-L1 loss L_{reg} for anchor regression and the direction classification loss L_{dir} for orientation estimation [1]:

$$L_{rpn} = L_{cls} + \alpha L_{reg} + \beta L_{dir} \tag{2}$$

where α is set to 2 and β is set to 0.2. We use the default parameters for focal loss [33]. The smooth-L1 loss regresses the residual value relative to the predefined anchor [1].

The RCNN loss includes the confidence prediction loss L_{score} guided by the IoU [12], the smooth-L1 loss L_{reg} for refining proposals and the corner loss L_{corner} [17]:

$$L_{rcnn} = L_{score} + L_{reg} + L_{corner} \tag{3}$$

The total training loss is the sum of the above losses:

$$L_{total} = L_{rpn} + L_{rcnn} \tag{4}$$

IV. EXPERIMENTS

In this section, we evaluate our proposed RangeRCNN on the challenging KITTI dataset [38].

Dataset. KITTI dataset [38] contains 7481 training samples and 7518 test samples. We follow the general split of 3712 training samples and 3769 validation samples. The KITTI dataset provides a benchmark for 3D object detection. We compare our proposed RangeRCNN with other state-of-the-art methods on this benchmark.

Metrics. The detection result is evaluated using the mean average precision (mAP) with the IoU threshold 0.7. For the official test benchmark, the mAP with 40 recall positions is reported. To fairly compare with some previous methods, we also report the mAP with 11 recall positions on the validation set. We note the used metric in the title of each table.

A. Implementation Details

Network Details. The input point cloud is converted to the range image representation with a size of $5 \times 48 \times 512$. The backbone for the range image is illustrated in Fig. 3. The feature map is downsampled in the encoder by six dilated residual blocks, and gradually upsampled in the decoder by four corresponding upsampled layers. The feature dimensions of the encoder are 32-64-128-256-256-256, and the feature dimensions of the decoder are 128-128-64-64. The size of the output features extracted from the range image is 48×512 with 64 dimensions. The resolution of BEV is 0.16^2 m^2, and the initial spatial size of BEV is 496×432. We use three convolution blocks to downsample the BEV to 248×216, 124×108 and 62×54, and upsample the three sizes to 248×216. Then the three features with the same size are concatenated along the channel axis. The 3D proposals are predicted based on the concatenated feature. In the two-stage RCNN, the 3D proposal generated by the RPN is divided into a fixed number of grids along with the local coordinate system of the 3D proposal. The spatial shape is set as $12 \times 12 \times 12$ in our implementation. We reshape the $12 \times 12 \times 12$ grids to a vectorized format with a $12^3 \times C$ dimension, where C is the feature dimension of each grid. In our implementation, the feature of each grid is obtained from the point features. So C is equal to 64. If multiple points fall into the same grid, the max pooling operation is used. Then three fully connected layers are applied to the vectorized feature. Finally, the confidence branch and the refinement branch are used to output the final result.

Training and Inference Details. We implement the proposed RangeRCNN with Pytorch1.3. The network can be trained in an end-to-end fashion with the ADAM optimizer. We train the entire network with the batch size 32, learning rate 0.01 for 80 epochs on 8 NVIDIA Tesla V100 GPUs, which takes about 1.5 hours. We adopt the cosine annealing learning rate strategy for the learning rate decay.

We use the data augmentation strategy similar with [1], [12], including random flipping along the x axis, random global scaling with a scaling factor sampled from $[0.95, 1.05]$, random global rotation around the vertical axis with a sampled angle from $[-\frac{\pi}{4}, \frac{\pi}{4}]$, and the ground-

TABLE I

PERFORMANCE COMPARISON WITH PREVIOUS METHODS ON THE KITTI ONLINE TEST SERVER. THE AP WITH 40 RECALL POSITIONS ($R40$) IS USED TO EVALUATE THE 3D OBJECT DETECTION AND BEV OBJECT DETECTION.

Method	Reference	Modality	3D			BEV		
			Easy	Moderate	Hard	Easy	Moderate	Hard
Multi-Modality:								
MV3D [34]	CVPR 2017	RGB+LIDAR	74.97	63.63	54.00	86.62	78.93	69.80
ContFuse [10]	ECCV 2018	RGB+LIDAR	83.68	68.78	61.67	94.07	85.35	75.88
AVOD-FPN [8]	IROS 2017	RGB+LIDAR	83.07	71.76	65.73	90.99	84.82	79.62
F-PointNet [17]	CVPR 2018	RGB+LIDAR	82.19	69.79	60.59	91.17	84.67	74.77
UberATG-MMF [9]	CVPR 2019	RGB+LIDAR	88.40	77.43	70.22	93.67	88.21	81.99
EPNet [35]	ECCV 2020	RGB+LIDAR	89.91	79.28	74.59	94.22	88.47	83.69
LIDAR-only:								
PointRCNN [19]	CVPR 2019	Point	86.96	75.64	70.70	92.13	87.39	82.72
Point-GNN [23]	CVPR 2020	Point	88.33	79.47	72.29	93.11	89.17	83.90
3D-SSD [24]	CVPR 2020	Point	88.36	79.57	74.55	92.66	89.02	85.86
SECOND [1]	Sensors 2018	Voxel	83.34	72.55	65.82	89.39	83.77	78.59
PointPillars [16]	CVPR 2019	Voxel	82.58	74.31	68.99	90.07	86.56	82.81
3D IoU Loss [36]	3DV 2019	Voxel	86.16	76.50	71.39	91.36	86.22	81.20
Part-A2 [12]	TPAMI 2020	Voxel	87.81	78.49	73.51	91.70	87.79	84.61
Fast Point R-CNN [37]	ICCV 2019	Voxel+Point	85.29	77.40	70.24	90.87	87.84	80.52
STD [20]	ICCV 2019	Voxel+Point	87.95	79.71	75.09	94.74	89.19	**86.42**
SA-SSD [13]	CVPR 2020	Voxel+Point	88.75	79.79	74.16	**95.03**	**91.03**	85.96
PV-RCNN [14]	CVPR 2020	Voxel+Point	**90.25**	**81.43**	76.82	94.98	90.65	86.14
LaserNet [4]	CVPR 2019	Range	-	-	-	79.19	74.52	68.45
RangeRCNN (ours)	-	Range	88.47	81.33	**77.09**	92.15	88.40	85.74

truth sampling augmentation to randomly "paste" some new ground-truth objects to current training scenes.

For RCNN, we choose 128 proposals with a 1:1 ratio for positive and negative samples during training. During inference, we retain the top 100 proposals according to the confidence with the NMS threshold 0.7. We apply the 3D NMS to the refined bounding boxes with a threshold of 0.1 to generate the final result.

B. Performance of RangeRCNN w.r.t State-of-the-art

We submit our results to the online KITTI benchmark to compare with other state-of-the-art methods. For evaluating the test set, we use all provided labeled samples to train our model. Table I shows the results evaluated on the KITTI online test server. Our RangeRCNN almost outperforms all previous approaches except PV-RCNN [14] on the commonly used moderate level for 3D car detection. We surprisingly observe that our method achieves the highest accuracy on the hard level. We think that the performance is beneficial to two aspects. First, some hard examples are very sparse in the 3D space, but they have more obvious features in the range image thanks to the compact representation. So these objects can be detected using the range image representation. Second, RCNN further refines the 3D position of the bounding box, which boosts the 3D performance. The ablation study also proves the value of RCNN.

Following the tradition of previous methods, we also compare the mean average precision with 11 recall positions on the validation set. Table II shows the results. Our method also achieves state-of-the-art performance. While achieving high precision, our method also runs at a high speed.

C. Ablation Study

In this section, we conduct the ablation study on the validation set of the KITTI dataset.

Effects of 3D RoI Pooling. As the entire framework is two-stage, we compare the result of the single-stage model RangeDet and the two-stage model RangeRCNN to better analyze the value of RCNN. From Table III, we can find that RangeDet and RangeRCNN have similar performance for BEV detection. But for 3D detection, RangeRCNN outperforms RangeDet by a large margin. The better 3D performance comes from the 3D information encoded by 3D RoI pooling.

We further evaluate the influence of the grid size in 3D RoI pooling. We compare a set of grid sizes in $\{6, 8, 10, 12, 14\}$. Table IV shows the results. It can be found that the grid size has no great influence on the metric. We choose 12 as the grid size which is a relatively better one.

Effects of the Range Image Backbone. We conduct a comparison experiment based on the single-stage model RangeDet to prove the effectiveness of the dilated convo-

TABLE II

PERFORMANCE COMPARISON WITH PREVIOUS METHODS ON THE MODERATE LEVEL OF KITTI VALIDATION SPLIT SET. THE 3D DETECTION AP WITH 11 RECALL POSITIONS (R11) IS USED.

Method	Reference	3D	FPS
MV3D [34]	CVPR 2017	62.68	2.8
AVOD-FPN [8]	IROS 2017	74.44	10
F-PointNet [17]	CVPR 2018	70.92	5.9
PointRCNN [19]	CVPR 2019	78.63	10
Point-GNN [23]	CVPR 2020	78.34	1.6
3D-SSD [24]	CVPR 2020	79.45	26
SECOND [1]	Sensors 2018	76.48	20
Part-A2 [12]	TPAMI 2020	79.47	14
Fast Point R-CNN [37]	ICCV 2019	79.00	15.4
STD [20]	ICCV 2019	79.80	10
SA-SSD [13]	CVPR 2020	79.91	25
PV-RCNN [14]	CVPR 2020	**83.90**	-
RangeDet (ours)	-	78.55	**43**
RangeRCNN (ours)	-	80.14	15

TABLE III

COMPARISION OF THE ONE-STAGE MODEL RANGEDET AND THE TWO-STAGE MODEL RANGERCNN. THE AP WITH 40 RECALL POSITIONS (R40) IS USED.

Method	3D			BEV		
	Easy	Moderate	Hard	Easy	Moderate	Hard
RangeDet	89.87	80.72	77.37	92.07	88.37	87.03
RangeRCNN	**91.41**	**82.77**	**80.39**	**92.84**	**88.69**	**88.20**

lution. We implement a simple baseline using the normal residual block without the dilated convolution. The result is shown in Table V. Using the dilated residual block brings approximately a 2% improvement on the 3D performance. It means that the flexible receptive field brought by the dilated convolution is helpful for the range image. Considering the characteristic of the range image, we believe that the specially designed backbone for the range image is valuable for 3D object detection. We leave this as future work.

D. Runtime Analysis

The inference time of the one-stage model RangeDet and the two-stage model RangeRCNN is 23 ms and 66 ms respectively, tested with an NVIDIA Tesla V100. We compare the inference time with other state-of-the-art methods in Table II. The one-stage model RangeDet is faster than all existing methods except Pointpillars [16]. The two-stage model RangeRCNN can also run at 15 fps. The high computation performance is beneficial from the compact representation of the range image.

TABLE IV

COMPARISION OF DIFFERENT POOLING SIZES IN $\{6, 8, 10, 12, 14\}$. THE AP WITH 40 RECALL POSITIONS (R40) IS USED.

Method	3D			BEV		
	Easy	Moderate	Hard	Easy	Moderate	Hard
RangeRCNN-6	89.55	82.33	80.01	92.56	88.47	87.78
RangeRCNN-8	89.23	82.35	79.96	92.44	88.49	88.00
RangeRCNN-10	89.48	82.62	80.36	92.76	88.60	88.11
RangeRCNN-12	91.41	**82.77**	**80.39**	**92.84**	**88.69**	**88.20**
RangeRCNN-14	**91.54**	82.61	80.29	92.67	88.49	88.16

TABLE V

COMPARISION OF DIFFERENT BACKBONES FOR THE RANGE IMAGE. THE AP WITH 40 RECALL POSITIONS (R40) IS USED.

Method	3D			BEV		
	Easy	Moderate	Hard	Easy	Moderate	Hard
RangeDet-normal	87.67	78.77	75.83	91.90	88.08	86.65
RangeDet-dilated	**89.87**	**80.72**	**77.37**	**92.07**	**88.37**	**87.03**

V. CONCLUSIONS

In this paper, we explore the potential of the range image representation and present a novel framework called RangeRCNN for fast and accurate 3D object detection. Our method achieves state-of-the-art performance on the KITTI dataset. The compact representation of the range image provides more possibilities for real-time 3D object detection in the large outdoor scene.

REFERENCES

[1] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[2] B. Graham, M. Engelcke, and L. van der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9224–9232.

[3] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," in *Proceedings of Robotics: Science and Systems (RSS)*, 2016.

[4] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "Lasernet: An efficient probabilistic 3d object detector for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 12 677–12 686.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[6] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 1451–1460.

[7] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6526–6534.

[8] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–8.

[9] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 7345–7353.

[10] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 641–656.

[11] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[12] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[13] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3d object detection from point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 873–11 882.

[14] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.

[15] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3075–3084.

[16] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 12 697–12 705.

[17] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 918–927.

[18] Z. Wang and K. Jia, "Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1742–1749.

[19] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object progposal generation and detection from point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 770–779.

[20] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "Std: Sparse-to-dense 3d object detector for point cloud," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1951–1960.

[21] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[22] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep hough voting for 3d object detection in point clouds," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9276–9285.

[23] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1711–1719.

[24] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 040–11 048.

[25] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[26] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in neural information processing systems*, 2018, pp. 820–830.

[27] Z. Liang, M. Yang, L. Deng, C. Wang, and B. Wang, "Hierarchical depthwise graph convolutional neural network for 3d semantic segmentation of point clouds," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8152–8158.

[28] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6411–6420.

[29] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1887–1893.

[30] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9297–9307.

[31] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4213–4220.

[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[34] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *IEEE CVPR*, vol. 1, no. 2, 2017, p. 3.

[35] T. Huang, Z. Liu, X. Chen, and X. Bai, "Epnet: Enhancing point features with image semantics for 3d object detection," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[36] D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, and R. Yang, "Iou loss for 2d/3d object detection," in *2019 International Conference on 3D Vision (3DV)*, 2019, pp. 85–94.

[37] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast point r-cnn," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[38] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.