SCIR-Net: Structured Color Image Representation Based 3D Object Detection Network from Point Clouds

Qingdong He[1 2*], Hao Zeng[1], Yi Zeng[1], Yijun Liu[1]

[1] University of Electronic Science and Technology of China,Chengdu,China
[2] Aibaba Group,Hangzhou, China
heqingdong@alu.uestc.edu.cn,haozeng@std.uestc.edu.cn,zengyii@std.uestc.edu.cn,yijunliu@std.uestc.edu.cn

Abstract

3D object detection from point clouds data has become an indispensable part in autonomous driving. Previous works for processing point clouds lie in either projection or voxelization. However, projection-based methods suffer from information loss while voxelization-based methods bring huge computation. In this paper, we propose to encode point clouds into structured color image representation (SCIR) and utilize 2D CNN to fulfill the 3D detection task. Specifically, we use the structured color image encoding module to convert the irregular 3D point clouds into a squared 2D tensor image, where each point corresponds to a spatial point in the 3D space. Furthermore, in order to fit for the Euclidean structure, we apply feature normalization to parameterize the 2D tensor image onto a regular dense color image. Then, we conduct repeated multi-scale fusion with different levels so as to augment the initial features and learn scale-aware feature representations for box prediction. Extensive experiments on KITTI benchmark, Waymo Open Dataset and more challenging nuScenes dataset show that our proposed method yields decent results and demonstrate the effectiveness of such representations for point clouds.

Introduction

The ability of perception and understanding in 3D environment is vital in autonomous driving (Geiger, Lenz, and Urtasun 2012) and virtual/augmented reality (VR/AR) scenarios (Park, Lepetit, and Woo 2008). In the domain of 3D sensing, 3D object detection is crucial and indispensable. In the last decades, with the advanced 3D sensing technology, the point clouds from LiDAR scanner have become a mainstream due to the rich spatial information and geometric features. However, the sparsity, irregularity and non-Euclidean structure of point clouds make it a tricky problem to apply the conventional deep convolutional neural networks (CNNs) to 3D object detection.

Current methods for processing point clouds can be divided into three streams, i.e., projection-based approaches, voxelization-based approaches and PointNet-based approaches. Projection-based approaches (Simon et al. 2019; Chen et al. 2017; Yang, Liang, and Urtasun 2018; Ku et al. 2018; Liang et al. 2018; Yang, Luo, and Urtasun 2018; Premebida et al. 2014) project point clouds into front view or birds eye view and employ 2D CNN to directly predict 3D bounding box. However, projection alone will lose the spatial information greatly and cannot represent geometry information of point clouds inevitably, so this type of method often combines semantic information from RGB images to compensate for the information loss, as shown in Fig. 1(a).

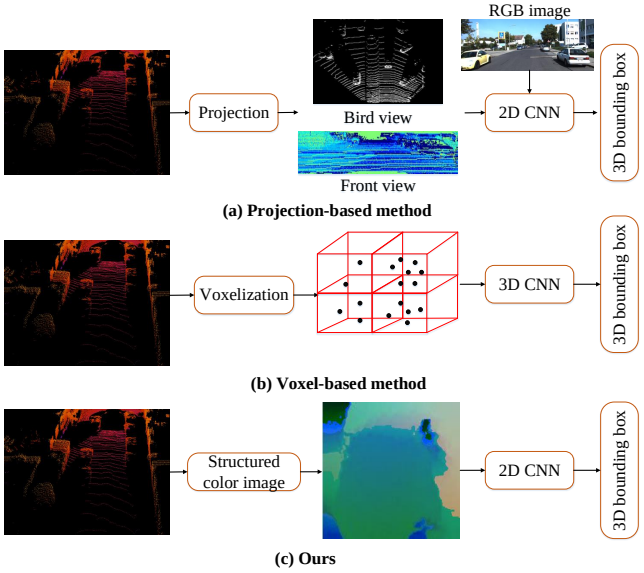

Figure 1: Comparison of different representations for point clouds. (a) Projection-based method: project point clouds to bird view or front view and combine RGB image to generate 3D box by 2D CNN, (b) Voxelization-based method: divide point clouds into equally spaced 3D voxels but use 3D CNN to generate 3D box, (c) our method: encode point clouds into structured color image and adopt 2D CNN to generate 3D box.

Compared to projection-based approaches, voxelization-based approaches (Zhou and Tuzel 2018; Yan, Mao, and Li 2018; Lang et al. 2019; Shi et al. 2020a) divide 3D point clouds into regular spaced voxels and apply 3D CNN (Li 2017; He et al. 2020a) to generate feature maps for 3D de-

*Corresponding Author

tection, as shown in Fig. 1(b). However, even though the later sparse convolution and submanifold sparse convolution (Graham, Engelcke, and Van Der Maaten 2018; Shi et al. 2020b) are applied, the problem of the huge amount of calculation (Liu et al. 2019) caused by the 3D CNN still has not been well solved. Powered by the remarkable feature extractor PointNet (Qi et al. 2017a,b), some works (Qi et al. 2018; Wang and Jia 2019; Shi, Wang, and Li 2019; Yang et al. 2019, 2020) try to predict 3D box from raw point clouds directly. But these methods achieve better performances with the uncontrolled receptive fields and non-compact neighborhood relations.

In order to apply the well-developed 2D CNN and avoid the irreversible lossy conversion, we consider a different paradigm that convert the 3D point clouds into regular 2D structured color image representation without aligning other data format (see Fig. 1(c)). Studies (Bommes, Zimmer, and Kobbelt 2009; Campen, Bommes, and Kobbelt 2015; Lyon et al. 2019) in computer graphics domain have looked into using geometry image representation and grid parameterization for the classification and semantic segmentation of point clouds. However, little studies have devoted to using structured image representation for 3D object detection task from point clouds. Our work demonstrates the feasibility of using structured image representation for accurate 3D object detection.

In this paper, we propose the structured color image representation based 3D object detection network from point clouds (SCIR-Net), which is capable of transforming the disorganized 3D point clouds into completely structured 2D color image representation and utilize 2D CNN to generate the category and bounding boxes information of the object. Specifically, the whole SCIR-Net is composed of structured color image encoding module, which aims to transform the global feature into structured color image, followed by the enforced detection network. Consuming the global feature from the feature extractor and the raw point clouds as input, the structured color image encoding module encodes the 3D points as the 2D tensor image through series of MLPs and then the feature normalization module is designed to generate the final regular color image representation. Such representation is differentiable and lossless which means the raw point clouds can be processed directly without carrying much computation burden. For regressing the final bounding box on these 2D dense color image, we design the enforced detection network with repeated multi-scale fusion scheme. Experiments on KITTI detection benchmark, Waymo Open Dataset and more challenging nuScenes dataset demonstrates that our proposed method can achieve comparable results with the state-of-the-art approaches, which validate the feasibility of such representation.

The main contributions of our work can be summarized as follows:

- We propose SCIR-Net, a new 3D object detection network which contains a new point clouds encoding method and enforced detection network.
- We design a new point clouds representation method that encodes the irregular point clouds in 3D space into 2D structured color image representation.
- We design the enforced detection network which augment the initial features with repeated multi-scale fusion scheme at different levels, further improving the 3D localization accuracy.
- Our proposed SCIR-Net achieves comparable results with the state-of-the-art methods on the KITTI 3D detection dataset, Waymo Open Dataset and more challenging nuScenes dataset.

Related Work

3D Object Detection with Multiple Sensors. There are several methods utilizing semantic information from RGB images and spatial information from point clouds for 3D object detection. Considering the properties of 2D CNN, many methods project the 3D points into different 2D views in order to align with RGB images. Among them, MV3D (Chen et al. 2017) projects point clouds into LIDAR bird view and LIDAR front view and fuses 3D proposals from bird view with RGB images to predict the final bounding box in the RPN network. AVOD (Ku et al. 2018) extends the fusion method by performing multimodal feature fusion on high resolution feature maps. The latest SRDL (He et al. 2020b) generates candidate boxes from stereo images and use edge convolution and MLP to process point clouds in parallel to improve the initial performance in F-pointnet (Qi et al. 2018) and IPOD (Yang et al. 2018). MMF (Liang et al. 2019) jointly reasons about 2D and 3D object detection, ground estimation and depth completion by utilizing depth maps, LIDAR point clouds and RGB images.

3D Object Detection with LiDAR Only. Dealing with 3D object detection, there are several streams of methods which use point clouds only. The first one is voxelization-based methods. VoxelNet (Zhou and Tuzel 2018) divides 3D point clouds into equally spaced 3D voxels and generated unified feature representation through 3D CNN after the newly designed VFE layer. In order to accelerate the calculation speed, SECOND (Yan, Mao, and Li 2018) applies sparse convolution layers and angle loss regression to improve the orientation estimation performance. SA-SSD (He et al. 2020a) also uses 3D CNN in the divided non-empty voxels and proposes an auxiliary network to exploit point-wise supervisions. Part A2 (Shi et al. 2020b) applies submanifold sparse convolution to estimates intra-object part locations and conducts the proposed RoI-aware point cloud pooling operation.

Thanks to the groundbreaking work PointNets (Qi et al. 2017a,b) in processing the irregular point clouds data, series of methods attempt to take raw point clouds as input to make the 3D predictions. PointRCNN (Shi, Wang, and Li 2019) utilizes PointNet++ as point cloud encoder-decoder to generate point-wise feature vector and proposes a two-stage RPN network to predict bounding boxes. To further reduce the large computation cost, 3DSSD (Yang et al. 2020) removes FP layers and the refinement module and propose a fusion sampling strategy in downsampling process to make detection on less representative points feasible. Considering the advantages of both point-based and voxel-based method-

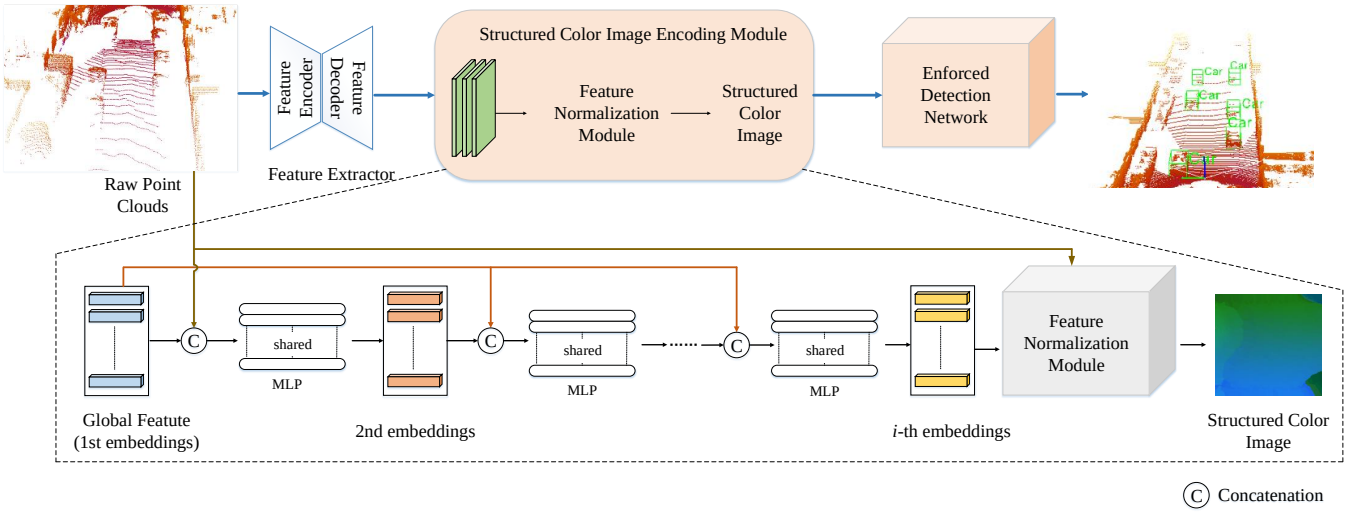

Figure 2: Illustration of the architecture of the proposed SCIR-Net. The whole network contains two sub-networks, (a) the structured color image encoding network to transform global feature into structured color image with series of feature embeddings and feature normalization module, and (b) the enforced detection network with repeated multi-scale fusion to learn scale-aware feature representations for classification and box prediction.

s, STD (Yang et al. 2019) proposes spherical anchors for accurate proposal generation, then applies the newly designed PointsPool to generate feature representations from sparse to dense for reducing inference time.

Parameterization on irregular data representations. In the field of digital geometry processing, parameterization is a typical method of transforming irregularly represented data (such as point clouds) into regular representations. Since the deformation from 3D surface to 2D plane inevitably occurs, parameterization is very popular (Gu and Yau 2003). Since the regional distortion of parameterization is closely related to the singularity of parameterization, many methods usually describe parameterization as mixed integer programming (Bommes, Zimmer, and Kobbelt 2009), where the positions of the singular points are continuous. Once the parameterization is calculated, the model can be divided into graphs to generate multi-graph geometric images (Campen, Bommes, and Kobbelt 2015). These methods are theoretically reasonable and elegant, but due to the high computational cost, they are impractical for curved surfaces with complex geometric shapes and topologies.

Proposed Method

In this paper, we propose the structured color image representation based 3D object detection network from point clouds (SCIR-Net), which encodes irregular 3D point clouds into structured 2D color image representation and apply 2D CNN architecture to accomplish accurate 3D object detection task. As shown in Fig. 2, SCIR-Net consists of the structured color image encoding module and a enforced detection network. The former aims to transform global feature into structured color image with series of feature embeddings and feature normalization module. Given the normalized 2D color image, the enforced detection network learns scale-aware feature representations with repeated multi-scale fusion for final classification and box prediction.

Feature Embedding Generation

Formally, considering a point cloud set with N points $\mathbf{P} = \{p_1, ..., p_N\}$, where each $p_i = (x_i, y_i, z_i, r_i)$ is a spatial point with 3D coordinates and the reflected laser intensity r_i. Given a point cloud $\mathbf{P}$, we propose to encode the Cartesian coordinates of spatial points into a 2D image color channels representation with depth, which denoted as $\mathbf{F} \in \mathbb{R}^{m \times m \times 4}$, dubbed as $(r, g, b; d)$, where d denotes the depth information in each 2D color point.

Feature Extractor. There are many methods processing point clouds to generate compact features (Qi et al. 2017b; Wang et al. 2019; Kaul, Pears, and Manandhar 2019; He et al. 2020b). In order to learn discriminative compact pointwise features for describing the raw point clouds, we utilize the combined edge convolution and MLPs in SRDL (He et al. 2020b) as our backbone network as the residual attention learning mechanism can extract deeper geometric features of different levels from the original irregular 3D point clouds. Further ablation studies will validate this choice for improving our performance.

Feature Embedding. Given the initial extracting feature $\mathbf{g} \in \mathbb{R}^S$ from the input point clouds $\mathbf{P}$, we first duplicate the initial feature N times and arrange them in a row-wise matrix $\mathbf{G} \in \mathbb{R}^{N \times S}$. To obtain the 1st embedding, we concatenate $\mathbf{G}$ with the input Cartesian coordinates $\mathbf{P}$, which denotes as $\mathbf{H_1} \in \mathbb{R}^{N \times (S+4)}$. Then, we apply a four-layer MLPs to $\mathbf{H_1}$ to generate the intermediate 2D embeddings $\mathbf{T_1}$.

For the 2nd embeddings, we apply the operation in a sim-

ilar fashion. We concatenate $\mathbf{T_1}$ with $\mathbf{G}$ to form $\mathbf{H_2}$ and feed $\mathbf{H_2}$ into another four-layer MLPs to generate the next embeddings $\mathbf{T_2}$. For the i-th embeddings, we repeat this process in the same way.

Putting it all together, we can express the feature embedding procedure as:

$$\mathbf{T_1} = \sigma(W_1(\mathbf{H_1})) = \sigma(W_1([\mathbf{P}, \mathbf{G}])) \quad (1)$$

$$\mathbf{T_{i+1}} = \sigma(W_i(\mathbf{H_{i+1}})) = \sigma(W_i([\mathbf{T_i}, \mathbf{G}])) \quad (2)$$

where $i = 1, ..., n$, $[\cdot, \cdot]$ denotes channel-wise concatenation between two feature matrices, W_i is the weights of the two MLPs and σ is the sigmoid activation function to normalize the embedded 2D points in the range of [0, 1]. We observe that three times embeddings have led to satisfactory results.

We should note that each 2D points $\mathbf{t_i} = (u_i, v_i)$ in $\mathbf{T_i}$ has a one-to-one correspondence with the point cloud in $\mathbf{P}$. To prevent the embedded 2D points from being over-clustered, we design a separation loss to extend the point distribution by imposing penalties on the clustered points until they are separated by the distance threshold D. Mathematically, the separation loss for each point $\mathbf{t_i}$ can be formulated as:

$$l_{fe}(\mathbf{t_i}) = \begin{cases} 0 & if \quad d_i \geq D \\ -log(d_i - D + 1) & otherwise \end{cases} \quad (3)$$

where $d_i = min\|\mathbf{t_i} - \mathbf{t_j}\|_2$ for $t_j \in \mathbf{T} \setminus \{\mathbf{t_i}\}$. In our implementation we set the distance threshold $D = \frac{1}{m-1}$.

For the feature embedding generation, we define the total loss function as:

$$L_{fe} = \frac{1}{N} \sum_{i}^{N} l_{fe}(\mathbf{t_i}) \quad (4)$$

By optimizing L_{fe}, the feature embedding generation part can encode the 3D points into the 2D tensor image square. However, the embedded 2D points are not distributed on the grid positions of the regular image lattice. In other words, the embedded 2D lattice does not conform to the European structure, which cannot be processed for the subsequent CNNs. Therefore, we design the feature normalization module to produce the regular dense color image.

Feature Normalization Module

Consuming the irregular 2D embeddings $\mathbf{T}$ as input, the feature normalization module parameterizes the 2D tensor image onto a regular dense color image $\mathbf{F}$ and the generated regular dense image can be directly reshaped into a point cloud $\mathbf{Q} \in \mathbb{R}^{M \times 4}$, where $M = m \times m$. As shown in Fig. 3, we design two strategies in the normalization process, which denoted as ball query normalization and bilinear interpolation normalization.

Ball Query Normalization. In order to maintain the consistency of the dimensions and reduce the loss of information, we construct a uniform canonical 2D square $\mathbf{V} \in \mathbb{R}^{M \times 2}$ in a unit grid domain. Each row of matrix $\mathbf{V}$ corresponds to a certain 2D point $\mathbf{v_i}$ within the $m \times m$ grid structure. For each $\mathbf{v_i}$, we search its nearest neighbour within a fixed radius r to form a local ball with n points in $\mathbf{T}$. Based

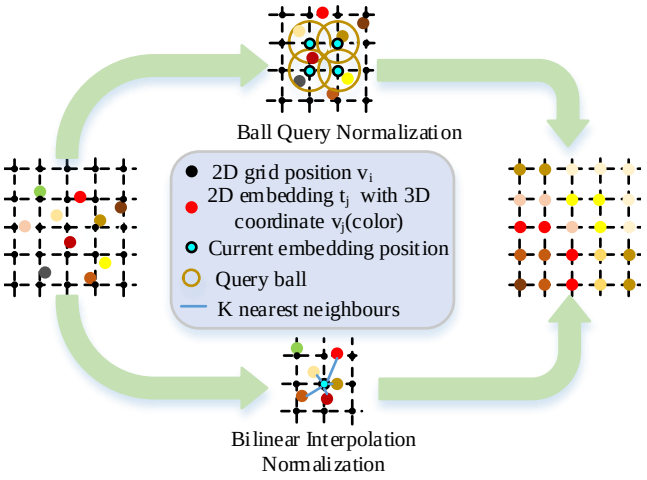

Figure 3: The feature normalization module parameterizes the 2D tensor image onto a regular dense color image using ball query normalization(the upper) and bilinear interpolation normalization(the bottom).

on the exact matching between $\mathbf{P}$ and $\mathbf{T}$, we can deduce a 2D array $\mathbf{Q}$:

$$\mathbf{q_i} = \sum_{j=1}^{n} \varrho_j \cdot \mathbf{p_j}, \quad where \quad \varrho_j = \frac{\mathbf{v_i^T} \cdot \mathbf{t_j}}{\sum_{j=1}^{n} \mathbf{v_i^T} \cdot \mathbf{t_j}} \quad (5)$$

Then, we reshape the $M \times 4$ array $\mathbf{Q}$ into a $m \times m \times 4$ regular dense color image. By increasing the resolution m of the regular dense color image sufficiently, the final regular dense color image is a lossless representation which includes all points of the input point clouds.

Bilinear Interpolation Normalization. In order to further realize an end-to-end optimization, we design an annealing strategy to generate the regular dense color image in a differentiable manner.

For each grid point $\mathbf{v_i}$, we select its K nearest neighbours from $\mathbf{T}$, denoted as $\{\mathbf{t}_k\}_{k=1}^{K}$, and their corresponding 3D points in $\mathbf{P}$ can be denoted as $\{\mathbf{p}_k\}_{k=1}^{K}$. Then, we determine series of weights using the distances between $\mathbf{v_i}$ and its K neighbours:

$$\hbar_{ik} = \frac{exp(-d_{ik}/|\omega|)}{\sum_{j=1}^{K} exp(-d_{ij}/|\omega|)} \quad (6)$$

where $d_{ik} = \|\mathbf{v_i} - \mathbf{t_k}\|_2$, and ω is a temperature coefficient. The i-th entry of $\mathbf{Q}$ can be formulated as

$$\mathbf{q_i} = \sum_{k=1}^{K} \hbar_{ik} \cdot \mathbf{p_k} \quad (7)$$

During annealing, we can approximate the nearest neighborhood point and the distribution of weights gradually converges to the Kronecker delta function when ω approaches 0. This can be easily achieved by optimizing the parameter term:

$$L_{fn} = \frac{|\omega|}{2} \quad (8)$$

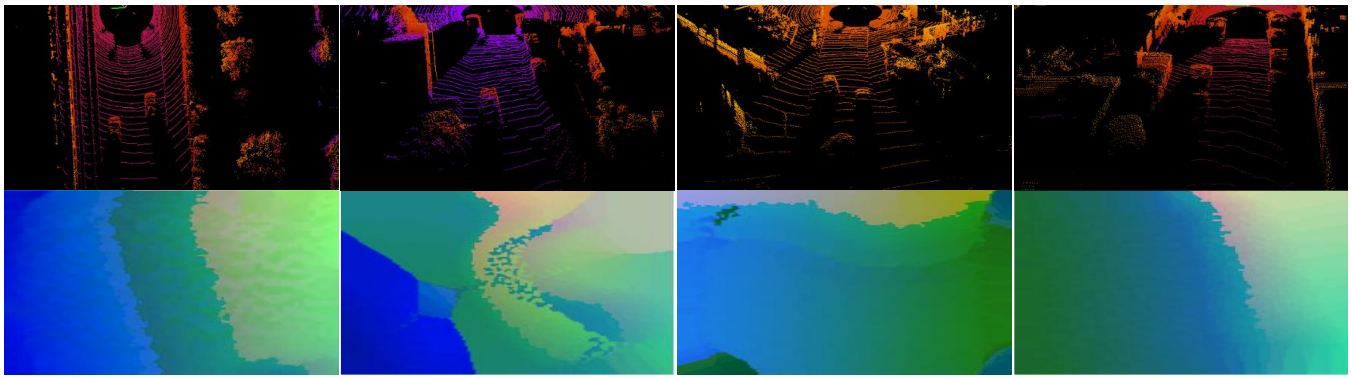

Figure 4: Examples of generated regular dense color images. The first row of each pair is the corresponding point cloud scene, and the second row is the generated regular dense color image.

By averaging the results of the ball query and bilinear interpolation normalization, we get the final regular dense color image. Fig. 4 shows some generated visual regular dense color image examples by our structured color image encoding module after sufficient iterations. We can observe that the regular dense color image can accurately represent the original point clouds while maintaining satisfactory smoothness.

Enforced Detection Network

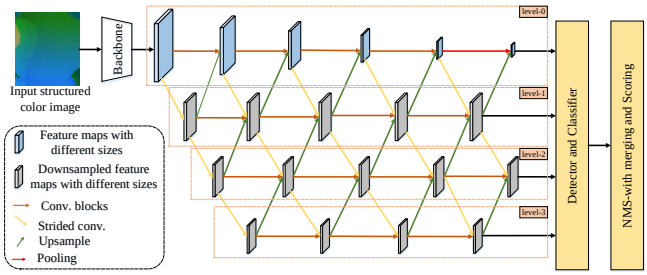

Figure 5: Architecture of the enforced detection network. The feature maps of each level are fused by other features at different scales and the whole network conducts repeated multi-scale fusion to make the final prediction.

After obtaining the structured dense color image with Euclidean structure from the raw point clouds, we aim at predicting the box sizes, orientations, locations and categories based on the 2D CNN detection network. To estimate more refined proposals and learn more specific local features, we propose to conduct repeated multi-scale fusion with different levels such that each of the feature maps in different scales receives information from other parallel scales over and over, leading to rich multi-scale representations.

As shown in Fig. 5, the enforced detection network takes the generated structured color image as input and transforms into the backbone network to make the initial feature maps generation. Apart from the original scale at level-0 with series of convolution blocks as SSD (Liu et al. 2016), we conduct three downsampling operations on the different feature maps from level-1 to level-3. For the feature maps at the same level, they are serialized and processed by the 1×1 and 3×3 convolution blocks. At the same time, each feature map from each level completes two sampling operations, the first one is to down-sampling by $\frac{1}{2}$ stride convolution to the low-scale, and the second is to up-sampling to the high-scale. The results of each up-sampling and down-sampling are added to the feature map of the corresponding level. As a result, the features for the parallel subnetworks of a later layer consists of the feature information from the previous previous layer, the higher scale and the lower one.

For the feature maps in these different levels, default boxes of different scales are constructed. Then they are detected and classified separately, and a number of default boxes that initially meet the conditions are generated. Finally, we combine the default boxes obtained from different feature maps, and use the NMS with box merging and scoring (Shi and Rajkumar 2020) to suppress some overlapping or incorrect boxes to generate the final boxes set.

Experiments

Datasets and Evaluation Metrics

KITTI dataset. We first evaluate our method on the widely used KITTI 3D object detection benchmark (Geiger, Lenz, and Urtasun 2012). It includes 7481 training samples and 7518 test samples with three categories: car, pedestrian and cyclist. For each category, results are evaluated based on three levels of difficulty: easy, moderate and hard. Furthermore, we divide the training data into a training set (3712 images and point clouds) and a validation set (3769 images and point clouds) at a ratio of about 1: 1. For evaluation, the average precision (AP) metric is to compare with different methods and the 3D IoU of car, cyclist, and pedestrian are 0.7, 0.5, and 0.5 respectively.

Waymo Open dataset. The Waymo Open Dataset (Sun et al. 2020) is by far the largest public data set for autonomous driving. There are a total of 1,000 sequences in the dataset. The training set contains 798 sequences with approximately 158000 point cloud samples, and the validation

Modality	Method	$3D_{car}$			$3D_{cyclist}$			BEV_{car}			$BEV_{cyclist}$		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Projection	MV3D(Chen et al. 2017)	71.09	62.35	55.12	-	-	-	86.02	76.90	68.49	-	-	-
	F-Pointnet(Qi et al. 2018)	81.20	70.39	62.19	71.96	56.77	50.39	88.70	84.00	75.33	75.38	61.96	54.68
	AVOD-FPN(Ku et al. 2018)	81.94	71.88	66.38	64.00	52.18	46.61	88.53	83.79	77.90	68.09	57.48	50.77
	MMF(Liang et al. 2019)	86.81	76.75	68.41	-	-	-	89.49	87.47	79.10	-	-	-
Voxel-based	Voxelnet(Zhou and Tuzel 2018)	77.47	65.11	57.73	61.22	48.36	44.37	89.35	79.26	77.39	66.70	54.76	50.55
	SECOND(Yan, Mao, and Li 2018)	83.13	73.66	66.20	70.51	53.85	46.90	88.07	79.37	77.95	73.67	56.04	48.78
	PointPillars(Lang et al. 2019)	79.05	74.99	68.30	75.78	59.07	52.92	88.35	86.10	79.83	79.14	62.25	56.00
	3DSSD(Yang et al. 2020)	**88.36**	79.57	74.55	-	-	-	-	-	-	-	-	-
Ours	SCIR-Net	87.53	**80.62**	**76.00**	**76.32**	**60.89**	**54.48**	**92.11**	**90.04**	**85.63**	**80.95**	**67.17**	**60.55**

Table 1: Performance comparison on KITTI 3D object detection and bird's eye view(BEV) for car and cyclists. The evaluation metrics is the average precision (AP) on the official test set.

set contains 202 sequences with approximately 40000 point cloud samples. Different from KITTI, which only provides annotations in the camera's FOV, Waymo Open Dataset provides annotations for objects throughout 360 degree. We adopt the official released evaluation tools for evaluating, where the mean average precision (mAP) and the mean average precision weighted by heading (mAPH) are used for evaluation. The rotated IoU threshold is set as 0.7 for vehicle detection.

nuScenes dataset. The nuScenes dataset (Caesar et al. 2020) is a recently released and more challenging dataset. It contains 1000 scenes, each of which is 20 seconds long, and is fully annotated with 3D bounding boxes of 23 categories and 8 attributes. The number of annotations and images are 7 times and 100 times that of the KITTI dataset, respectively. It provides us with 1.4 million 3D objects in 10 different categories, as well as their properties and speeds. There are about 40k points in each frame, and in order to predict the speed and attributes, all previous methods combine the key frames and the points in the last 0.5s frame to get about 400000 points. The evaluation metric on this dataset is called nuScences detection score (NDS), which is a weighted sum between mean average precision (mAP), the mean average errors of location (mATE), size (mASE), orientation (mAOE), attribute (mAAE) and velocity (mAVE).

Network Architecture. In our implementation, we employ SRDL (He et al. 2020b) as our feature extractor for the raw input point clouds. The initial extracted global feature for an input point cloud is a 512-dimensional feature vector, i.e., $\mathbf{g} \in \mathbb{R}^{512}$. In feature embedding, each embedding is processed by MLP layers and we employ three times in total. The user-specified resolution m is set as 128. For the annealing regularization term in bilinear interpolation normalization, we initiated the learnable temperature coefficient as $\omega = 10^{-6}$. Besides, the number of nearest neighbors selected for each grid point is set as $K = 5$. In practice, we discover that the bilinear interpolation normalization module is not sensitive to the value of K due to the effective annealing process. For car(vehicles), the number of the point clouds N is set as 1024. For cyclist and pedestrians, N is set as 512. In the enforced detection network, we apply VGG16 (Simonyan and Zisserman 2014) as our backbone, replace fc6 and fc7 with convolutional layers and subsample parameters from fc6 and fc7. The IoU threshold of NMS is set to 0.6.

Training. The whole network is end-to-end optimized with the Adam optimizer (Kingma and Ba 2014) on GTX 1080 GPU. The loss weights are $\alpha = 2$, $\beta = 0.5$, $\gamma = 1$ and $\xi = 1.5$. For KITTI dataset, the network is trained for 120 epoches with batchsize 16, learning rate 0.015. For Waymo Open dataset, the network is trained for 50 epoches with batchsize 32, learning rate 0.01. For nuScenes dataset, the network is trained for 60 epoches with batchsize 32.

Main Results

Method	Modality	$3D_{car}$		
		Easy	Moderate	Hard
Projection	MV3D (Chen et al. 2017)	71.29	62.68	56.56
	F-Pointnet (Qi et al. 2018)	83.76	70.92	63.65
	AVOD-FPN (Ku et al. 2018)	84.41	74.44	68.65
	F-ConvNet(Wang and Jia 2019)	89.02	78.80	77.09
Voxel-based	Voxelnet (Zhou and Tuzel 2018)	81.97	65.46	62.85
	SECOND (Yan, Mao, and Li 2018)	87.43	76.48	69.10
	SA-SSD(He et al. 2020a)	90.15	79.91	78.78
	3DSSD(Yang et al. 2020)	89.71	79.45	78.67
Ours	SCIR-Net	**92.47**	**85.07**	**82.74**

Table 2: Performance comparison of 3D object detection and bird's eye view(BEV) detection on KITTI val set for car class.

Results on KITTI Dataset. We evaluate our method on the 3D detection and the bird's eye view detection benchmark of the KITTI test server. As shown in Table 1, we compare our results with state-of-the-art RGB+LIDAR and LIDAR only methods for both tasks on car and cyclist. By only taking the point clouds as input, our proposed approach outperforms the most effective RGB+LIDAR methods MMF(Liang et al. 2019) for car category on three difficulty levels. By applying 2D CNN, our SCIR-Net can also achieve decent results with the Voxel-based methods, which utilize 3D CNN, especially on moderate and hard difficulties. Besides, the AP comparison for 3D object detection of our SCIR-Net on KITTI val set is presented in Table 2. Our proposed SCIR-Net achieves the best performance on all difficulty levels on the val set for car class.

Results on Waymo Open dataset. We evaluate our SCIR-Net on both LEVEL_1 and LEVEL_2 objects for 3D and BEV mAP. As shown in Table 3, our method outperforms previous methods with remarkable margins on all ranges of both LEVEL_1 and LEVEL_2. Specifically, with the commonly used LEVEL_1 objects detection, our method achieves new performance with 75.63% and 88.45% on 3D and BEV mAP evaluation metric. The whole experimental

	Car	Ped	Bus	Barrier	TC	Truck	Trailer	Moto	Cons. Veh.	Bicycle	mAP
SECOND (Yan, Mao, and Li 2018)	75.35	59.86	29.04	32.21	22.49	21.88	12.96	16.89	0.36	0	27.12
PointPillars(Lang et al. 2019)	70.5	59.9	34.4	33.2	29.6	25.0	20.0	16.7	4.5	1.6	29.5
3DSSD(Yang et al. 2020)	81.20	70.17	61.41	47.94	31.06	47.15	30.45	35.96	12.64	8.63	42.66
SCIR-Net(ours)	**85.21**	**76.45**	**74.86**	**52.17**	**37.65**	**53.37**	**40.12**	**47.64**	**26.65**	**29.38**	**52.35**

Table 5: AP comparison on nuScenes dataset.

Difficulty	Method	Overall	0-30m	30-50m	50m-Inf
LEVEL_1 (3D mAP)	PointPillars(Lang et al. 2019)	56.62	81.01	51.75	27.94
	MVF(Zhou et al. 2020)	62.93	86.30	60.02	36.02
	PV-RCNN (Shi et al. 2020a)	70.30	91.92	69.21	42.17
	SCIR-Net(ours)	**75.63**	**92.55**	**72.42**	**49.17**
LEVEL_2 (3D mAP)	PV-RCNN (Shi et al. 2020a)	65.36	91.58	65.13	36.46
	SCIR-Net(ours)	**66.73**	**91.84**	**67.22**	**39.54**
LEVEL_1 (BEV mAP)	PointPillars(Lang et al. 2019)	75.57	92.10	74.06	55.47
	MVF(Zhou et al. 2020)	80.40	93.59	79.21	63.09
	PV-RCNN (Shi et al. 2020a)	82.96	97.35	82.99	64.97
	SCIR-Net(ours)	**88.45**	**97.71**	**88.41**	**76.09**
LEVEL_2 (BEV mAP)	PV-RCNN (Shi et al. 2020a)	77.45	94.64	80.39	55.39
	SCIR-Net(ours)	**81.65**	**96.88**	**81.34**	**62.56**

Table 3: Performance comparison on the Waymo Open Dataset with 202 validation sequences for the vehicle detection.

	mAP	mATE	mASE	mAOE	mAVE	mAAE	NDS
PointPillars(Lang et al. 2019)	29.5	0.54	0.29	0.45	0.29	0.41	44.9
3DSSD(Yang et al. 2020)	42.6	0.39	0.29	0.44	0.22	0.12	56.4
SCIR-Net(ours)	**52.35**	**0.32**	**0.23**	**0.41**	**0.19**	**0.11**	**58.7**

Table 4: NDS comparison on nuScenes dataset.

results on the large-scale Waymo Open dataset further validate that our proposed structured dense color images are able to effectively encode more accurate information for improving the 3D detection performance and demonstrate the generalization ability of our proposed network.

Results on nuScenes dataset. We show the comparison of our SCIR-Net with state-of-the-arts on mAP and NDS in Table 4, and compare their APs of each class in Table 5. Our method acquires better performance compared to others by a large margin. Not only on mAP, it also outperforms those methods on AP of each class. The results show that our model can handle different objects even for huge scenes and large scale differences.

Ablation Studies

fe	fn	EDN	Easy	Moderate	Hard
–	–	–	85.64	81.59	77.48
×	✓	✓	87.77	83.32	79.25
✓	×	✓	88.46	83.71	80.69
✓	✓	×	89.63	83.55	81.32
✓	✓	✓	92.47	85.07	82.74

Table 6: Performance of removing different part of our network. × denotes removing and ✓ denotes retaining. EDN is the enforced detection network.

As shown in Table 6, we illustrate the importance of different components of our network by removing each part and keeping all the others unchanged. Table 7 details how each proposed module influences the accuracy and efficiency of our SCIR-Net. The results are evaluated with AP for car class.

		$3DAP_{car}(\%)$			$BEVAP_{car}(\%)$		
		Easy	Moderate	Hard	Easy	Moderate	Hard
FE	Pointnet++	83.45	72.36	70.02	82.47	83.12	80.28
	DGCNN	86.27	77.54	76.29	87.83	86.02	85.17
	SAWNet	90.13	81.65	79.44	91.23	89.72	88.17
	SRDL	92.47	85.07	82.74	95.53	91.23	90.65
No.	1	82.57	76.29	73.43	87.12	84.43	83.65
	2	88.91	80.74	78.19	91.87	88.56	88.32
	3	92.47	85.07	82.74	95.53	91.23	90.65
	4	89.63	83.22	80.32	93.47	90.02	89.27
level	0	77.54	70.82	68.56	82.29	80.37	79.08
	1	84.65	76.62	72.34	86.25	85.44	83.21
	2	90.38	82.92	79.52	91.34	89.66	88.17
	3	92.47	85.07	82.74	95.53	91.23	90.65

Table 7: Performance of proposed method with different design choice on KITTI val set. 'FE', 'No.' and 'level' stands for feature extractor, number of feature embeddings and number of levels in enforced detection network.

(a) To extract the initial global feature, we experiment with different extractor for processing the raw point clouds. SRDL (He et al. 2020b) performs best which indicates the local and global feature extractors to capture richer deep features from point clouds are most suitable for our module.

(b) In feature embedding, we conduct three times embeddings to generate the 2D tensor image by successively increasing the number of embeddings. Finally, we find that in our model, three times is the number which can achieve the highest accuracy.

(c) In enforced detection network, we start from level-0 and increase the down-sampled level one by one to the previous level. Final implementation demonstrates that three levels have been able to fuse enough feature information to generate accurate detection boxes.

Conclusion

In this paper, we have proposed a new framework SCIR-Net to perform accurate 3D object detection from point clouds. We introduce a novel encoding method to convert the irregular 3D point clouds into structured 2D dense color image. This newly proposed point representation makes it feasible to apply 2D CNNs to fulfill the final box prediction. In the enforced detection network, a repeated multi-scale fusion scheme with different levels is designed to learn scale-aware features to boost the performance of our model. All of above delicate designs enable our SCIR-Net to show decent accuracy on the public KITTI dataset, Waymo Open dataset and more challenging nuScenes dataset.

References

Bommes, D.; Zimmer, H.; and Kobbelt, L. 2009. Mixed-integer quadrangulation. *ACM Transactions On Graphics (TOG)*, 28(3): 1–10.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Campen, M.; Bommes, D.; and Kobbelt, L. 2015. Quantized global parametrization. *Acm Transactions On Graphics (tog)*, 34(6): 1–12.

Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1907–1915.

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

Graham, B.; Engelcke, M.; and Van Der Maaten, L. 2018. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9224–9232.

Gu, X.; and Yau, S.-T. 2003. Global conformal surface parameterization. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 127–137.

He, C.; Zeng, H.; Huang, J.; Hua, X.-S.; and Zhang, L. 2020a. Structure Aware Single-stage 3D Object Detection from Point Cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11873–11882.

He, Q.; Wang, Z.; Zeng, H.; Liu, Y.; Liu, S.; and Zeng, B. 2020b. Stereo RGB and Deeper LIDAR Based Network for 3D Object Detection. *Neurocomputing*.

Kaul, C.; Pears, N.; and Manandhar, S. 2019. Sawnet: A spatially aware deep neural network for 3d point cloud processing. *arXiv preprint arXiv:1905.07650*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12697–12705.

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

Liang, M.; Yang, B.; Chen, Y.; Hu, R.; and Urtasun, R. 2019. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7345–7353.

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

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*, 21–37. Springer.

Liu, Z.; Tang, H.; Lin, Y.; and Han, S. 2019. Point-Voxel CNN for efficient 3D deep learning. In *Advances in Neural Information Processing Systems*, 963–973.

Lyon, M.; Campen, M.; Bommes, D.; and Kobbelt, L. 2019. Parametrization quantization with free boundaries for trimmed quad meshing. *ACM Transactions on Graphics (TOG)*, 38(4): 1–14.

Park, Y.; Lepetit, V.; and Woo, W. 2008. Multiple 3D Object Tracking for Augmented Reality. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, 117–120. Washington, DC, USA: IEEE Computer Society. ISBN 978-1-4244-2840-3.

Premebida, C.; Carreira, J.; Batista, J.; and Nunes, U. 2014. Pedestrian detection combining RGB and dense LIDAR data. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4112–4117. IEEE.

Qi, C. R.; Liu, W.; Wu, C.; Su, H.; and Guibas, L. J. 2018. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 918–927.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, 5099–5108.

Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020a. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10529–10538.

Shi, S.; Wang, X.; and Li, H. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–779.

Shi, S.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020b. 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*.

Shi, W.; and Rajkumar, R. 2020. 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*, 1711–1719.

Simon, M.; Amende, K.; Kraus, A.; Honer, J.; Samann, T.; Kaulbersch, H.; Milz, S.; and Michael Gross, H. 2019. Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. In *Advances in Neural Information Processing Systems*, 440–144.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2446–2454.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5): 1–12.

Wang, Z.; and Jia, K. 2019. Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1742–1749. IEEE.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

Yang, B.; Liang, M.; and Urtasun, R. 2018. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, 146–155.

Yang, B.; Luo, W.; and Urtasun, R. 2018. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 7652–7660.

Yang, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11040–11048.

Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; and Jia, J. 2018. Ipod: Intensive point-based object detector for point cloud. *arXiv preprint arXiv:1812.05276*.

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

Zhou, Y.; Sun, P.; Zhang, Y.; Anguelov, D.; Gao, J.; Ouyang, T.; Guo, J.; Ngiam, J.; and Vasudevan, V. 2020. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning*, 923–932. PMLR.

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