2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

MVX-Net: Multimodal VoxelNet for 3D Object Detection

Vishwanath A. Sindagi[1], Yin Zhou[2] and Oncel Tuzel[2]

***Abstract*— Many recent works on 3D object detection have focused on designing neural network architectures that can consume point cloud data. While these approaches demonstrate encouraging performance, they are typically based on a single modality and are unable to leverage information from other modalities, such as a camera. Although a few approaches fuse data from different modalities, these methods either use a complicated pipeline to process the modalities sequentially, or perform late-fusion and are unable to learn interaction between different modalities at early stages. In this work, we present *PointFusion* and *VoxelFusion*: two simple yet effective early-fusion approaches to combine the RGB and point cloud modalities, by leveraging the recently introduced VoxelNet architecture. Evaluation on the KITTI dataset demonstrates significant improvements in performance over approaches which only use point cloud data. Furthermore, the proposed method provides results competitive with the state-of-the-art multimodal algorithms, achieving top-2 ranking in five of the six birds eye view and 3D detection categories on the KITTI benchmark, by using a simple single stage network.**

Fig. 1. Example 3D detection result from the KITTI validation set projected onto an image. *Top row:* VoxelNet [43], where yellow boxes represent detections. The solid red circle and dashed red circles highlight a false negative and two false positives by VoxelNet, respectively. *Bottom row:* Proposed method, where green rectangles indicate detections.

I. INTRODUCTION

With the advent of 3D sensors and diverse applications of 3D understanding, there is an increased research focus on 3D recognition [40], object detection [43], [28], and segmentation [29]. A wide variety of applications such as augmented reality [26], robotics [25], and navigation [11], [13] rely heavily on 3D understanding. Among these tasks, 3D object detection is a fundamental problem and forms a crucial step in many 3D understanding pipelines. In this work, we focus on improving the 3D detection performance by fusing multiple modalities.

2D object detection is an extensively researched topic in the computer vision community. Convolutional neural network (CNN) based techniques [12], [22], [20], [30] have shown excellent performance on image-based detection datasets [21], [10], [7]. However, these methods cannot be applied directly to 3D detection since the input modalities are fundamentally different. LiDAR enables accurate localization of objects in the 3D space, and detection techniques based on LiDAR data often outperform the 2D techniques. Some of these methods convert 3D point cloud to hand-crafted feature representations, such as depth or bird's eye view (BEV) maps [5], [42] and then apply 2D-CNN based methods for vehicle detection and classification. However, these techniques suffer from quantization which leads to reduced performance for objects with fewer points or variable geometries. Another set of techniques represent 3D point cloud data in a voxel grid [24], [23] and employ 3D CNNs to generate detection results. These methods are often limited by the memory requirements, especially when processing full scenes.

Recent research on 3D classification has focused on enabling the use of end-to-end trainable neural networks that can consume point cloud data without transforming them to intermediate representations, such as depth or BEV formats. Qi *et al*. [29] designed a neural network architecture that directly takes point clouds as input and outputs class labels. With this design, one can learn the representations from the raw data. However, this work could not be applied to the problem of detection and localization due to the limitations in architecture design along with high computational and memory cost. Recently, Zhou and Tuzel [43] overcame this issue by proposing VoxelNet, which involves voxelization of a point cloud and encoding the voxels using stacks of Voxel Feature Encoding (VFE) layers. With these steps, VoxelNet enabled the use of a 3D region proposal network for detection. Although this method demonstrates encouraging performance, it relies on a single modality, *i.e*., point cloud data. In contrast to point clouds, RGB images provide much denser texture information and it is desirable to leverage both modalities to improve the detection performance.

In this paper, we propose Multimodal VoxelNet (MVX-Net), to augment LiDAR points with semantic image features and learn to fuse image and LiDAR features at early stages for accurate 3D object detection. The proposed approach extends the recently proposed VoxelNet algorithm [43]. Specifically, we develop two fusion techniques: (i) *PointFusion*: This is an early-fusion method where points from the LiDAR

[1]Vishwanath A. Sindagi is with the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore. vishwanathsindagi@jhu.edu

[2]Yin Zhou and Oncel Tuzel are with AI Research, Apple Inc. yzhou3@apple.com, otuzel@apple.com

978-1-5386-6027-0/19/$31.00 ©2019 IEEE

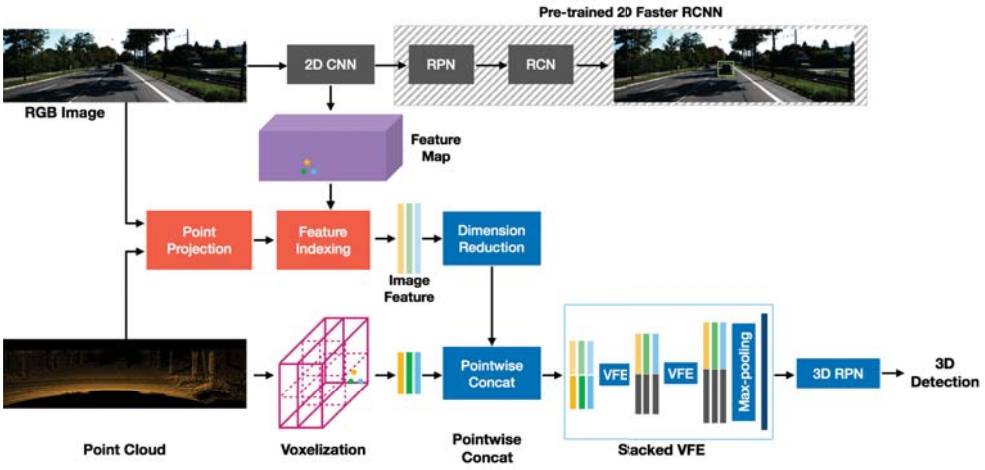

Fig. 2. Overview of the proposed MVX-Net *PointFusion* method. The method uses convolutional filters of pre-trained 2D faster RCNN to compute the image feature map. Note that the RPN and RCN (shown in shaded rectangle) are not part of the 3D inference pipeline. The 3D points are projected to the image using the calibration information and the corresponding image features are appended to the 3D points. The VFE layers and the 3D RPN process the aggregated data and produce the 3D detections.

sensor are projected onto the image plane, followed by image feature extraction from a pre-trained 2D detector. The concatenation of image features and the corresponding points are then jointly processed by the VoxelNet architecture. (ii) *VoxelFusion*: In this technique, non-empty 3D voxels created by VoxelNet are projected to the image, followed by extracting image features for every projected voxel using a pre-trained CNN. These features are then pooled and appended to the VFE feature encoding for every voxel and further used by the 3D region proposal network (RPN) to produce 3D bounding boxes. Compared to *PointFusion*, *VoxelFusion* is a relatively later fusion technique, which however, can be extended to handle empty voxels as well, thereby reducing the dependency on the availability of high-resolution LiDAR point cloud data. As illustrated in Fig. 1, the proposed MVX-Net effectively fuses multimodal information leading to reduced false positives and negatives compared to the LiDAR-only VoxelNet.

This paper is organized as follows. Section II describes related work on 3D object detection. Section III introduces the proposed Multimodal VoxelNet algorithm and two fusion techniques for effectively combining multimodal information. Section IV presents experimental results. Finally, Section V concludes the paper and points out future directions for improvement.

II. RELATED WORK

As discussed earlier, 3D understanding is an extensively researched topic. Earlier approaches ([6], [36], [1], [8], [37]) employ hand-crafted representations and achieve satisfactory results in the presence of rich and detailed 3D information. Some of the techniques ([38], [9], [35], [35], [17]) represent 3D point cloud data using voxel occupancy grid representation, followed by the use of 3D convolutions to compute the 3D bounding boxes. Due to the high computational and memory cost, several approaches based on BEV representation were developed ([27], [14], [42]). The BEV-based methods assume that point cloud data is sparse in one dimension, which is usually not the case in many scenarios. Different from these approaches, image-based methods ([3], [4], [39], [44], [45], [34], [2]) were developed to infer 3D bounding boxes from 2D images. However, they usually suffer from low accuracy in terms of depth localization. Recently, VoxelNet [43] proposed an end-to-end learning architecture that consumes point cloud data in its raw format.

Multimodal fusion by combining LiDAR and RGB data has been less explored as compared to single modality-based approaches. Recently, Chen *et al.* [5] proposed a multi-view 3D object detection network (MV3D), which takes multimodal data as input and produces 3D bounding boxes by incorporating region-based feature fusion. Although this method demonstrated encouraging results by using multimodal data, it has the following disadvantages: (i) the method converts point clouds into BEV representation, which loses detailed 3D shape information, and (ii) the fusion is performed at a much later stage as compared to the proposed fusion techniques (*i.e.*, after the 3D proposal generation stage), which limits the ability of the neural network to capture the interaction between the two modalities at earlier stages and hence, the integration is not necessarily seamless. Similar to [5], Ku *et al.* [16] proposed multimodal fusion by incorporating region-based features. They achieved better performance than [5] especially in the small object category by designing a more advanced RPN that employs high-resolution feature maps. This method also uses hand-crafted BEV representation and performs late fusion.

In a different approach, Qi *et al.* proposed Frustum PointNets [28] for 3D detection using LiDAR and RGB data. First, they use a 2D object detector on RGB data to generate 2D proposals, which are then converted to frustum proposals in the 3D space, followed by a point-wise instance segmentation using the PointNet architecture [29]. This method is an image-first approach and hence lacks the capability of utilizing both modalities simultaneously. Most recently, Liang *et al.* [19] proposed to aggregate the discrete BEV space with image features by projecting the

LiDAR points to image space. This approach interpolates each BEV pixel location with RGB features based on K nearest neighbor search, which may not satisfy real-time requirements as the density and coverage of LiDAR point clouds increase. In contrast to existing approaches that either use a complicated pipeline to process different modalities or perform late-fusion, our simple yet effective fusion strategies can learn interaction between modalities at early stages.

III. PROPOSED METHOD

The proposed fusion techniques, illustrated in Fig. 2 and Fig. 3, are based on the VoxelNet [43] architecture. In order to fuse information from RGB and point cloud data, we first extract features from the last convolutional layer of a 2D detection network. This network is first pre-trained on ImageNet [32], [33] and then fine-tuned for the 2D object detection [31] task. These high-level image features encode semantic information that can be used as prior knowledge to help infer the presence of an object. Based on the type of fusion described earlier (*PointFusion* or *VoxelFusion*), either points or voxels are projected onto the image and the corresponding features are concatenated with point features or voxel features respectively. Details of the 2D detection network, VoxelNet, and the proposed fusion techniques are described in the following subsections.

A. 2D Detection Network

Compared to LiDAR point clouds, RGB images capture richer color and texture information. In this work, to improve 3D detection accuracy, we extract high-level semantic features from RGB images and incorporate them into the VoxelNet algorithm.

Convolutional neural networks are highly effective at learning semantic information present in the images. Here, we propose to use an existing 2D detection framework which has shown excellent performance on various tasks [21], [10], [7]. Specifically, we employ the Faster-RCNN framework [31] which consists of a region proposal network (RPN) and a region classification network (RCN). We use VGG16 [33] pre-trained on ImageNet [32] as the base network and finetune the Faster-RCNN network using images from a 2D detection dataset and the corresponding bounding box annotations. More training details are described in Section III-D.

Once the detection network is trained, high-level features (from the conv5 layer of the VGG16 network) are extracted and fused either at the point or voxel level.

B. VoxelNet

We choose the VoxelNet architecture as the base 3D detection network for two main reasons: (i) it consumes raw point clouds and removes the need for hand-crafted features and (ii) it provides a natural and effective interface for combining image features at different granualities in 3D space, *i.e.,* points and voxels. We use the network as described in [43]. For completeness, we briefly revisit VoxelNet in this section. This algorithm consists of three building blocks: (i) a Voxel Feature Encoding (VFE) layer (ii) Convolutional Middle Layers, and (iii) a 3D Region Proposal Network.

VFE is a feature learning network that aims to encode raw point clouds at the individual voxel level. Given a point cloud, the 3D space is divided into equally spaced voxels, followed by grouping the points to voxels. Then each voxel is encoded using a hierarchy of voxel feature encoding layers. First, every point $\mathbf{p}_i = [x_i, y_i, z_i, r_i]^T$ (containing the XYZ coordinates and the reflectance value) in a voxel is represented by its co-ordinates and its relative offset with respect to the centroid of the points in the voxel. That is each point is now represented as: $\hat{\mathbf{p}}_i = [x_i, y_i, z_i, r_i, x_i - v_x, y_i - v_y, z_i - v_z]^T \in \mathbb{R}^7$, where x_i, y_i, z_i, r_i are the XYZ coordinates and the reflectance value of the point p_i, and v_x, v_y, v_z are the XYZ coordinates of the centroid of the points in the voxel which p_i belongs to. Next, each $\hat{\mathbf{p}}_i$ is transformed through the VFE layer which consists of a fully connected network (FCN) into a feature space, where information from the point features can be aggregated to encode the shape of the surface contained within the voxel. The FCN is composed of a linear layer, a batch normalization (BN) layer, and a rectified linear unit (ReLU) layer. The transformed features belonging to a particular voxel are aggregated using element-wise max-pooling. The max-pooled feature vector is then concatenated with point features to form the final feature embedding. All non-empty voxels are encoded in the same way and they share the same set of parameters in FCN. Stacks of such VFE layers are used to transform the input point cloud data into high-dimensional features.

The output of the stacked VFE layers are forwarded through a set of convolutional middle layers that apply 3D convolution to aggregate voxel-wise features within a progressively expanding receptive field. These layers incorporate additional context, thus enabling the use of context information to improve the detection performance.

Following the convolutional middle layers, a region proposal network [12] performs the 3D object detection. This network consists of three blocks of fully convolutional layers. The first layer of each block downsamples the feature map by half via a convolution with a stride size of 2, followed by a sequence of convolutions of stride 1. After each convolution layer, BN and ReLU operations are applied. The output of every block is then upsampled to a fixed size and concatenated to construct a high resolution feature map. Finally, this feature map is mapped to the targets: (1) a probability score map and (2) a regression map.

C. Multimodal Fusion

As discussed earlier, VoxelNet [43] is based on a single modality and adapting it to multimodal input enables further performance improvements. In this paper, we propose two simple techniques to fuse RGB data with the point cloud data by extending the VoxelNet framework.

PointFusion: This is an early fusion technique where every 3D point is aggregated by an image feature to capture a dense context.

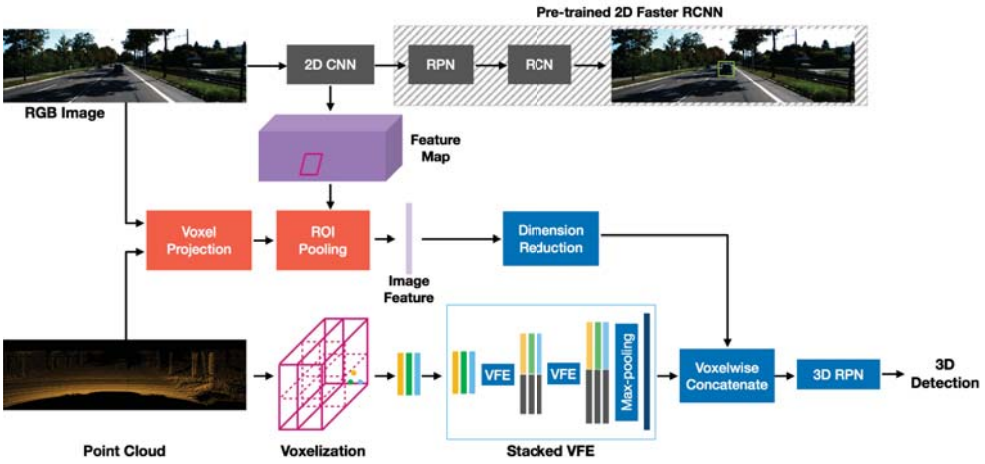

Fig. 3. Overview of the proposed MVX-Net *VoxelFusion* method. The method uses convolutional filters of pre-trained 2D faster RCNN to compute the image feature map. Note that the RPN and RCN (shown in shaded rectangle) are not part of the 3D inference pipeline. The non-empty voxels are projected to the image using the calibration information to obtain the ROIs. The features within each ROI are pooled and appended to the voxel features computed by VFE layers. The 3D RPN processes the aggregated data and produces the 3D detections.

The method first uses a pre-trained 2D detection network (described in Section III-A) to extract a high level feature map from the image which encodes image-based semantics. Then using the calibration matrix, it projects each 3D point onto the image and appends the point with the feature corresponding to the projected location index. This process associates information about the presence and, if it exists, the pose of the object from 2D images to every 3D point. Note that the features are extracted from the conv5 layer of the VGG16 network and are 512 dimensional. We first reduce the dimensionality to 16 through a set of fully connected layers and then concatenate them to the point features. The concatenated features are processed by a set of VFE layers in VoxelNet and then used in the detection stage. Fig. 2 provides an overview of this approach.

The advantage of this approach is that since the image features are concatenated at a very early stage, the network can learn to summarize useful information from both modalities through the VFE layer. Moreover, the approach exploits the LiDAR point cloud and lifts the corresponding image features to the coordinates of the 3D points.

VoxelFusion: In contrast to *PointFusion* that combines features at an earlier stage, *VoxelFusion* employs a relatively later fusion strategy where the features from the RGB image are appended at the voxel level. As described in [43], the first stage in VoxelNet involves dividing the 3D space into a set of equally spaced voxels. Points are grouped into these voxels based on where they reside, after which each voxel is encoded using a VFE layer. In the proposed *VoxelFusion* method, every non-empty voxel is projected onto the image plane to produce a 2D region of interest (ROI). Using the feature map from the pre-trained detector network (conv5 layer of VGG16), the features within the ROI are pooled to produce a 512-dimensional feature vector, whose dimensionality is first reduced to 64 and then appended to the feature vector produced by the stacked VFE layers at every voxel. This process encodes prior information from the 2D image at every voxel. Fig. 3 provides an overview of this approach.

Although *VoxelFusion* is a relatively later fusion strategy and produces slightly inferior performance as compared to *PointFusion*, it has the following advantages. First, it can be easily extended to aggregate image information to empty voxels where LiDAR points are not sampled due to reasons such as low LiDAR resolution or far objects, thereby reducing dependency on the availability of high-resolution LiDAR points. Second, *VoxelFusion* is more efficient in terms of memory consumption as compared to *PointFusion*.

D. Training Details

2D Detector: We use the standard Faster-RCNN detection framework [31], which is a two stage detection pipeline consisting of a region proposal network and a region classification network. The base network is VGG16 architecture and we use ROIAlign [15] operation to pool the features from the last convolutional layer before forwarding them to the second stage (RCNN). We use four sets of anchors with sizes $\{4,8,16,32\}$ and three aspect ratios $\{0.5,1,2\}$ on the conv5 layer. Anchors are labeled as positive if the intersection-over-union (IoU) with the ground truth boxes is greater than 0.7, and the anchors are labeled as negative if the IoU is less than 0.3. During training, the shortest side of the image is rescaled to 600 pixels. The training dataset is augmented with standard techniques such as flipping and adding random noise. For the RCNN stage, we use a batch size of 128 with 25% of the samples reserved for foreground ROIs. The network is trained using stochastic gradient descent with a learning rate of 0.0005 and momentum of 0.9.

Multimodal VoxelNet: We retain most of the settings of VoxelNet as described in [43] apart from a few simplifications to improve the efficiency. The 3D space is divided into voxels of sizes $v_D = 0.4$, $v_H = 0.2$, $v_W = 0.2$. Two sets of VFE layers and three convolutional middle layers are used. The input and output dimensionalities of these layers are different based on the type of fusion.

For *PointFusion*, the VFE stack has a configuration of VFE-1(7+16,32) and VFE-2(32,128). The input to the first

TABLE I

COMPARISON OF RESULTS ON KITTI VALIDATION SET USING MEAN AVERAGE PRECISION (IN %) WITH IOU=0.7. TOP-2 METHODS ARE HIGHLIGHTED IN BOLD. (S: SINGLE MODALITY, M:MULTIMODAL)

Method	AP_{BEV} (IoU=0.7)			AP_{3D} (IoU=0.7)		
	Easy	Med	Hard	Easy	Med	Hard
Mono3D [3] (S)	5.22	5.19	4.13	2.53	2.31	2.31
3DOP [4] (S)	12.6	9.49	7.5	6.55	5.07	4.10
VeloFCN [18] (S)	40.1	32.0	30.4	15.2	13.6	15.9
MV3D [5] (S)	86.2	77.3	76.3	71.2	56.6	55.3
MV3D [5] (M)	86.6	78.1	76.7	71.3	62.7	56.6
PIXOR [42] (S)	86.8	80.8	76.6	N/A	N/A	N/A
F-PointNet [28] (M)	88.2	84.0	76.4	**83.8**	70.9	63.7
VoxelNet [43] (S)	**89.6**	**84.8**	**78.6**	82.0	65.5	62.9
Baseline VoxelNet (S)	87.6	83.7	78.4	79.5	65.7	64.6
MVX-Net (VF) (M)	88.6	84.6	**78.6**	82.3	**72.2**	**66.8**
MVX-Net (PF) (M)	**89.5**	**84.9**	**79.0**	**85.5**	**73.3**	**67.4**

TABLE II

COMPARISON OF RESULTS ON KITTI VALIDATION SET USING MEAN AVERAGE PRECISION (MAP) WITH IOU=0.8.

Method	AP_{BEV} (IoU=0.8)			AP_{3D} (IoU=0.8)		
	Easy	Med	Hard	Easy	Med	Hard
Baseline VoxelNet (S)	72.4	62.2	56.5	32.8	28.1	24.6
MVX-Net (VF) (M)	72.2	62.3	61.0	39.5	30.8	29.8
MVX-Net (PF) (M)	74.2	64.5	61.6	43.6	33.2	31.3

VFE layer is a concatenation of point features which have 7 dimensions and CNN features which have 16 dimensions. Note that the features extracted from conv5 layer of the pre-trained 2D detection network have a dimensionality of 512. Their dimensions are first reduced to 96 and finally to 16 using two fully-connected (FC) layers with BN and ReLU.

For *VoxelFusion*, the VFE stack has a configuration of VFE-1(7,32) and VFE-2(32,64). Features extracted from conv5 layer of the pre-trained 2D detection network have a dimensionality of 512, and they are reduced to 128D and 64D using two FC layers, each followed by a BN and a ReLU non-linearity. These dimension reduced features are then concatenated to the output of VFE-2 to form a 128 dimensional vector for every voxel. By reducing the output dimensionality of VFE-2 to 64 (as compared to 128 in the original work), we ensure that the architecture of the convolutional middle layers remain unchanged.

To reduce the memory footprint, we trim the RPN by using only half of the number of ResNet blocks as in the original work. We employ the same anchor matching strategies as those in the original work. For both fusion techniques, the network is trained using stochastic gradient descent with a learning rate of 0.01 for the first 150 epochs, after which the learning rate is decayed by a factor of 10. Furthermore, since we use both images and point clouds, some of the augmentation strategies used in the original work are not applicable to the proposed multimodal framework, e.g., global point cloud rotation. Despite training with a trimmed RPN and using less data augmentations, the proposed multimodal framework is still able to achieve significantly higher detection accuracy compared to the original LiDAR-only VoxelNet [43].

IV. EXPERIMENTS AND RESULTS

A. Dataset

The proposed fusion techniques are evaluated on the KITTI 3D object detection dataset [11] that contains 7,481 training samples and 7,518 test samples. There are three difficulty levels: easy, moderate and hard which are determined based on the object size, visibility (occlusion) and truncation. We further split the training set into train/validation sets by avoiding samples from the same sequence being included in both sets [5]. After the split, the training set consists of 3712 samples and the validation set consists of 3769 samples.

We compare the proposed MVX-Net with previously published approaches on the car detection tasks. To analyze effectiveness of the proposed multimodal approaches we also trained a baseline VoxelNet model. Similar to the multimodal approaches, this model used the trimmed architecture and did not use the global rotation augmentation. By comparing the results to this baseline, we can directly attribute the gains to the proposed multimodal fusion techniques.

B. Evaluation on KITTI Validation Set

We follow the standard KITTI evaluation protocol (IoU=0.7) for measuring the detection performance. Table I shows the mean average precision (mAP) scores for *VoxelFusion* and *PointFusion* compared to the state-of-the-art methods on the KITTI validation set using 3D and bird's eye view (BEV) evaluation. In all fusion experiments, the detection performance improved significantly after fusion as compared to the baseline VoxelNet. The effectiveness of fusion is more pronounced in terms of 3D mAP scores than BEV mAP scores. It is also important to note that the proposed fusion techniques are able to obtain improved performance as compared to the original VoxelNet which has a more powerful RPN and uses more data augmentations. Moreover, our approach consistently outperforms other recent top-performing approaches [5], [42], [28]. Fig. 4 compares example detection results from the proposed approaches and the LiDAR-only VoxelNet [43].

It can be observed that *VoxelFusion* yields slightly lower performance as compared to *PointFusion* because *PointFusion* combines features at an earlier stage. It is worth pointing out that in contrast to *PointFusion* which is LiDAR-centric, *VoxelFusion* can exploit both modalities independently. For efficient training and inference, our current implementation only projects non-empty voxels onto the image. However, the *VoxelFusion* method can be extended by projecting all voxels onto the image. This strategy utilizes image-based information regardless of existence of the points within a voxel which could be helpful in far-range detection, where LiDAR has a very low resolution.

We conduct an ablation study by replacing the 2D CNN features by the cropped raw image patches and append them to the 3D points. We test image patch sizes 3x3 and 5x5 and find that even this simple strategy yields 0.5% to 1.0% higher AP as compared to the baseline VoxelNet model. However,

(a) (b) (c)

Fig. 4. Sample 3D detection results from KITTI validation dataset projected onto image for visualization. *(a)* VoxelNet [43], *(b)* MVX-Net with *VoxelFusion*, *(c)* MVX-Net with *PointFusion*. Green rectangles indicate detection results. Red rectangles highlight missed detections and false positives.

TABLE III
COMPARISON OF RESULTS ON KITTI TEST SET USING MEAN AVERAGE PRECISION (IN %) WITH IOU=0.7. TOP-2 METHODS ARE HIGHLIGHTED IN BOLD. (S: SINGLE MODALITY, M:MULTIMODAL)

Method	AP_{BEV} (IoU=0.7)			AP_{3D} (IoU=0.7)		
	Easy	Med	Hard	Easy	Med	Hard
MV3D [5] (S)	85.8	77.0	68.9	66.8	52.7	51.3
PIXOR [42] (S)	81.7	77.1	73.0	N/A	N/A	N/A
PIXOR++ [41] (M)	**89.4**	83.7	78.0	N/A	N/A	N/A
VoxelNet [43] (S)	**89.4**	79.3	77.4	77.5	65.1	57.7
MV3D [5] (M)	86.0	76.9	68.5	71.1	62.4	55.1
F-PointNet [28] (M)	88.7	84.0	75.3	81.2	70.4	62.2
AVOD [16] (M)	86.8	85.4	77.7	73.6	65.8	58.4
AVOD-FPN [16] (M)	88.5	83.8	77.9	81.9	**71.9**	**66.4**
HDNET [41] (M)	89.1	**86.6**	**78.3**	N/A	N/A	N/A
Cont-Fuse [19] (M)	88.8	85.8	77.3	**82.5**	66.2	64.0
MVX-Net (PF) (M)	89.2	**85.9**	**78.1**	**83.2**	**72.7**	**65.2**

this result is not as good as using a higher level feature computed through an image CNN.

We investigate the performance of different methods with a more rigorous evaluation criterion by increasing the IoU threshold to 0.8. The mAP scores for this configuration are summarized in Table II. As the IoU criterion increases, the performance improvement by multimodal fusion is more pronounced, which indicates that multimodal fusion helps to improve not only the detection but also the localization accuracy over approaches using a single modality.

C. Evaluation on KITTI Test Set

We evaluate the proposed MVX-Net with *PointFusion* on the KITTI test set by submitting detection results to the official server. The results are summarized in Table III. We observe that the MVX-Net with *PointFusion* achieves competitive results with the state-of-the-art 3D detection algorithms. Out of six birds eye view and 3D detection categories, the proposed approach achieves top rank in two categories, 2nd rank in three categories, and 3rd in one other category.

V. CONCLUSION

In this work, we present two feature fusion techniques, *PointFusion* and *VoxelFusion*, to combine RGB with LiDAR, by extending the recently proposed VoxelNet [43]. *PointFusion* involves projection of 3D points onto the image using a known calibration matrix, followed by feature extraction from a pre-trained 2D CNN and concatenation of image features at point level. *VoxelFusion* involves projection of 3D voxels onto the image, followed by feature extraction within 2D ROIs and concatenation of pooled image features at the voxel level. In contrast to existing multimodal techniques, the proposed methods are single stage detectors which are simple and effective. Experimental results on the KITTI dataset demonstrate significant improvements over approaches using a single modality. Furthermore, our approach yields results competitive with the state-of-the-art multimodal algorithms on KITTI test set. In the future, we plan to train a multi-class detection network, and compare the current two-stage training with end-to-end training.

REFERENCES

[1] P. Bariya and K. Nishino. Scale-hierarchical 3D object recognition in cluttered scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1657–1664, 2010.

[2] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3D vehicle analysis from monocular image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2040–2049, 2017.

[3] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3D object detection for autonomous driving. In *IEEE CVPR*, 2016.

[4] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3D object proposals for accurate object class detection. In *NIPS*, 2015.

[5] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3D object detection network for autonomous driving. In *IEEE CVPR*, 2017.

[6] C. S. Chua and R. Jarvis. Point signatures: A new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–85, Oct 1997.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[8] C. Dorai and A. K. Jain. Cosmos-a representation scheme for 3D free-form objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1115–1130, 1997.

[9] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361, May 2017.

[10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

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

[13] R. Gomez-Ojeda, J. Briales, and J. Gonzalez-Jimenez. Pl-svo: Semi-direct monocular visual odometry by combining points and line segments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4211–4216, Oct 2016.

[14] A. Gonzalez, G. Villalonga, J. Xu, D. Vazquez, J. Amores, and A. Lopez. Multiview random forest of local experts combining rgb and lidar data for pedestrian detection. In *IEEE Intelligent Vehicles Symposium (IV)*, 2015.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[16] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander. Joint 3D proposal generation and object detection from view aggregation. *arXiv preprint arXiv:1712.02294*, 2017.

[17] B. Li. 3D fully convolutional network for vehicle detection in point cloud. In *IROS*, 2017.

[18] B. Li, T. Zhang, and T. Xia. Vehicle detection from 3D lidar using fully convolutional network. In *Robotics: Science and Systems*, 2016.

[19] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3D object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[20] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 3, 2017.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

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

[23] D. Maturana and S. Scherer. 3D Convolutional Neural Networks for Landing Zone Detection from LiDAR. In *ICRA*, 2015.

[24] D. Maturana and S. Scherer. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *IROS*, 2015.

[25] Y.-J. Oh and Y. Watanabe. Development of small robot for home floor cleaning. In *Proceedings of the 41st SICE Annual Conference. SICE 2002.*, volume 5, pages 3222–3223 vol.5, Aug 2002.

[26] Y. Park, V. Lepetit, and W. Woo. Multiple 3D object tracking for augmented reality. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 117–120, Sept 2008.

[27] C. Premebida, J. Carreira, J. Batista, and U. Nunes. Pedestrian detection combining RGB and dense LIDAR data. In *IROS*, pages 0–1. IEEE, Sep 2014.

[28] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3D object detection from rgb-d data. 2018.

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[31] 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*, pages 91–99, 2015.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] S. Song and M. Chandraker. Joint sfm and detection cues for monocular 3D localization in road scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3734–3742, June 2015.

[35] S. Song and J. Xiao. Sliding shapes for 3D object detection in depth images. In *European Conference on Computer Vision*, pages 634–651, 2014.

[36] F. Stein and G. Medioni. Structural indexing: efficient 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.

[37] O. Tuzel, M.-Y. Liu, Y. Taguchi, and A. Raghunathan. Learning to rank 3D features. In *13th European Conference on Computer Vision, Proceedings, Part I*, pages 520–535, 2014.

[38] D. Z. Wang and I. Posner. Voting for voting in online point cloud object detection. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[39] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. Data-driven 3D voxel patterns for object category recognition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2015.

[40] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese. Objectnet3D: A large scale database for 3D object recognition. In *European Conference on Computer Vision*, pages 160–176. Springer, 2016.

[41] B. Yang, M. Liang, and R. Urtasun. HDNET: Exploiting HD maps for 3D object detection. In *2nd Conference on Robot Learning (CoRL)*, 2018.

[42] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3D object detection from point clouds. In *CVPR*, 2018.

[43] Y. Zhou and O. Tuzel. VoxelNet: End-to-end learning for point cloud based 3D object detection. 2018.

[44] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3D representations for object recognition and modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2608–2623, 2013.

[45] M. Z. Zia, M. Stark, and K. Schindler. Are cars just 3D boxes? jointly estimating the 3D shape of multiple objects. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3678–3685, June 2014.