Multi-Task Multi-Sensor Fusion for 3D Object Detection

Ming Liang[1*] Bin Yang[1,2*] Yun Chen[1†] Rui Hu[1] Raquel Urtasun[1,2]
[1]Uber Advanced Technologies Group [2]University of Toronto
{ming.liang, byang10, yun.chen, rui.hu, urtasun}@uber.com

Abstract

In this paper we propose to exploit multiple related tasks for accurate multi-sensor 3D object detection. Towards this goal we present an end-to-end learnable architecture that reasons about 2D and 3D object detection as well as ground estimation and depth completion. Our experiments show that all these tasks are complementary and help the network learn better representations by fusing information at various levels. Importantly, our approach leads the KITTI benchmark on 2D, 3D and bird's eye view object detection, while being real-time.

1. Introduction

Self-driving vehicles have the potential to improve safety, reduce pollution, and provide mobility solutions for otherwise underserved sectors of the population. Fundamental to its core is the ability to perceive the scene in real-time. Most autonomous driving systems rely on 3-dimensional perception, as it enables interpretable motion planning in bird's eye view.

Over the past few years we have seen a plethora of methods that tackle the problem of 3D object detection from monocular images [2, 31], stereo cameras [4] or LiDAR point clouds [36, 34, 16]. However, each sensor has its challenge: cameras have difficulty capturing fine-grained 3D information, while LiDAR provides very sparse observations at long range. Recently, several attempts [5, 17, 12, 13] have been developed to fuse information from multiple sensors. Methods like [17, 6] adopt a cascade approach by using cameras in the first stage and reasoning in LiDAR point clouds only at the second stage. However, such cascade approach suffers from the weakness of each single sensor. As a result, it is difficult to detect objects that are occluded or far away. Others [5, 12, 13] have proposed to fuse multi-sensor features instead. Single-stage detectors [13] fuse multi-sensor feature maps per LiDAR point, where local nearest neighbor interpolation is used to densify the correspondence. However, the fusion is still limited when LiDAR points become extremely sparse at long range. Two-stage detectors [5, 12] fuse multi-sensor features per object region of interest (ROI). However, the fusion process is typically slow (as it involves thousands of ROIs) and imprecise (either using fix-sized anchors or ignoring object orientation).

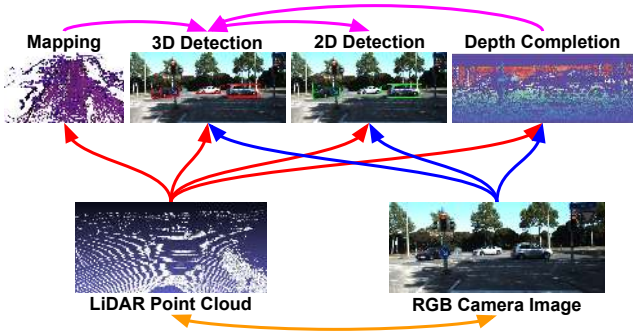

Figure 1. Different sensors (bottom) and tasks (top) are complementary to each other. We propose a joint model that reasons on two sensors and four tasks, and show that the target task - 3D object detection can benefit from multi-task learning and multi-sensor fusion.

In this paper we argue that by solving multiple perception tasks jointly, we can learn better feature representations which result in better detection performance. Towards this goal, we develop a multi-sensor detector that reasons about 2D and 3D object detection, ground estimation and depth completion. Importantly, our model can be learned end-to-end and performs all these tasks at once. We refer the reader to Figure 1 for an illustration of our approach.

We propose a new multi-sensor fusion architecture that leverages the advantages from both point-wise and ROI-wise feature fusion, resulting in fully fused feature representations. Knowledge about the location of the ground can provide useful cues for 3D object detection in the context of self-driving vehicles, as the traffic participants of interest are restrained to this plane. Our detector estimates an accurate voxel-wise ground location online as one of its auxiliary tasks. This in turn is used by the bird's eye view

*Equal contribution.
†Work done as part of Uber AI Residency program.

(BEV) backbone network to reason about relative location. We also exploit the task of depth completion to learn better cross-modality feature representation and more importantly, to achieve dense point-wise feature fusion with pseudo LiDAR points from dense depth.

We demonstrate the effectiveness of our approach on the KITTI object detection benchmark [8] as well as the more challenging TOR4D object detection benchmark [34]. On the KITTI benchmark, we show very significant performance improvement over previous state-of-the-art approaches in 2D, 3D and BEV detection tasks. Meanwhile, the proposed detector runs over 10 frames per second, making it a practical solution for real-time application. On the TOR4D benchmark, we show detection improvement from multi-task learning over previous state-of-the-art detector.

2. Related Work

We review related works that exploit multi-sensor fusion and multi-task learning to improve 3D object detection.

3D detection from single modality: Early approaches to 3D object detection focus on camera based solutions with monocular or stereo images [3, 2]. However, they suffer from the inherent difficulties of estimating depth from images and as a result perform poorly in 3D localization. More recent 3D object detectors rely on depth sensors such as LiDAR [34, 36]. However, although range sensors provide precise depth measurements, the observations are usually sparse (particularly at long range) and lack the information richness of images. It is thus difficult to distinguish classes such as pedestrian and cyclist with LiDAR-only detectors.

Multi-sensor fusion for 3D detection: Recently, a variety of 3D detectors that exploit multiple sensors (*e.g.* LiDAR and camera) have been proposed. F-PointNet [17] uses a cascade approach to fuse multiple sensors. Specifically, 2D object detection is done first on images, 3D frustums are then generated by projecting 2D detections to 3D and PointNet [18, 19] is applied to regress the 3D position and shape of the bounding box. In this framework the overall performance is bounded by each stage which is still using single sensor. Furthermore, object localization from a frustum in LiDAR point cloud has difficulty dealing with occluded or far away objects as LiDAR observation can be very sparse (often with a single point on the far away object). MV3D [5] generates 3D proposals from LiDAR features, and refines the detections with ROI feature fusion from LiDAR and image feature maps. AVOD [12] further extends ROI feature fusion to the proposal generation stage to improve the object proposal quality. However, ROI feature fusion happens only at high-level feature maps. Furthermore, it only fuses features at selected object regions instead of dense locations on the feature map. To overcome this drawback, ContFuse [13] uses continuous convolution [30] to fuse multi-scale convolutional feature maps from each sensor, where the correspondence between image and BEV spaces is achieved through projection of the LiDAR points. However, such fusion is limited when LiDAR points are very sparse. To address this issue, we propose to predict dense depth from multi-sensor data, and use the predicted depth as pseudo LiDAR points to find dense correspondences between multi-sensor feature maps.

3D detection from multi-task learning: Various auxiliary tasks have been exploited to help improve 3D object detection. HDNET [33] exploits geometric ground shape and semantic road mask for BEV vehicle detection. SBNet [21] utilizes the sparsity in road mask to speed up 3D detection by > 2 times. Our model also reasons about a geometric map. The difference is that this module is part of our detector and thus end-to-end trainable, so that these two tasks can be optimized jointly. Wang *et al*. [29] exploit depth reconstruction and semantic segmentation to help 3D object detection. However, they rely on 3D rendering, which is computationally expensive. Other contextual cues such as the room layout [23, 26], and support surface [24] have also been exploited to help 3D object reasoning in the context of indoor scenes. 3DOP [3] exploits monocular depth estimation to refine the 3D shape and position based on 2D proposals. Mono3D [2] uses instance segmentation and semantic segmentation as evidence, along with other geometric priors to reason about 3D object detection from monocular images. Apart from geometric map estimation, we also exploit depth completion which brings two benefits: it guides the network to learn better cross-modality feature representations, and its prediction serves as pseudo LiDAR points for dense fusion between image and BEV feature maps.

3. Multi-Task Multi-Sensor Detector

One of the fundamental tasks in autonomous driving is to perceive the scene in real-time. In this paper we propose a multi-task multi-sensor fusion model for the task of 3D object detection. We refer the reader to Figure 2 for an illustration of the model architecture. Our approach has the following highlights. First, we design a multi-sensor architecture that combines point-wise and ROI-wise feature fusion. Second, our integrated ground estimation module reasons about the geometry of the road. Third, we exploit the task of depth completion to learn better multi-sensor features and achieve dense point-wise feature fusion. As a result, the whole model can be learned end-to-end by exploiting a multi-task loss.

In the following, we first introduce the architecture of the multi-sensor 2D and 3D detector with point-wise and ROI-

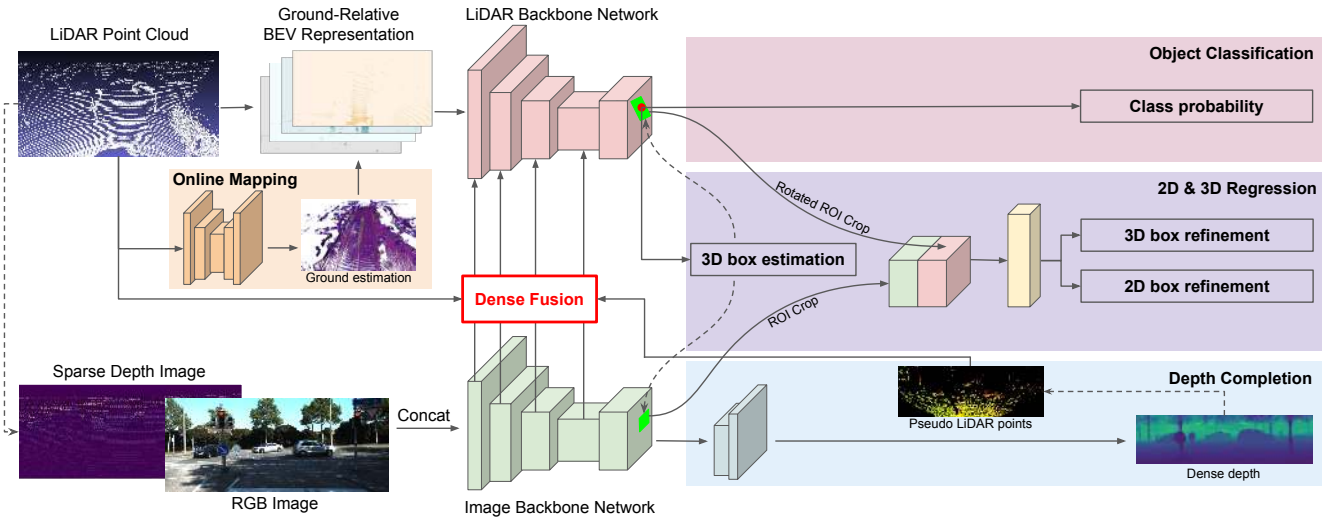

Figure 2. The architecture of the proposed multi-task multi-sensor fusion model for 2D and 3D object detection. Dashed arrows denote projection, while solid arrows denote data flow. Our model is a simplified two-stage detector with densely fused two-stream multi-sensor backbone networks. The first stage is a single-shot detector that outputs a small number of high-quality 3D detections. The second stage applies ROI feature fusion for more precise 2D and 3D box regression. Ground estimation is explored to incorporate geometric ground prior to the LiDAR point cloud. Depth completion is exploited to learn better cross-modality feature representation and achieve dense feature map fusion by transforming predicted dense depth image into dense pseudo LiDAR points. The whole model can be learned end-to-end.

wise feature fusion. We then show how we exploit the other two auxiliary tasks to further improve 3D detection. Finally we provide details of how to train our model end-to-end.

3.1. Fully Fused Multi-Sensor Detector

Our multi-sensor detector takes a LiDAR point cloud and an RGB image as input. The backbone network adopts the two-stream structure, where one stream extracts image feature maps, and the other extracts LiDAR BEV feature maps. Point-wise feature fusion is applied to fuse multi-scale image features to BEV stream. The final BEV feature map predicts dense 3D detections per BEV voxel via 2D convolution. After Non-Maximum Suppression (NMS) and score thresholding, we get a small number of high-quality 3D detections and their projected 2D detections (typically fewer than 20 when tested on KITTI dataset). We then apply a 2D and 3D box refinement by ROI-wise feature fusion, where we combine features from both image ROIs and BEV oriented ROIs. After the refinement, the detector outputs accurate 2D and 3D detections.

Input representation: We use the voxel based LiDAR representation [13] due to its efficiency. In particular, we voxelize the point cloud into a 3D occupancy grid, where the voxel feature is computed via 8-point linear interpolation on each LiDAR point. This LiDAR representation is able to capture fine-grained point density clues efficiently. We consider the resulting 3D volume as BEV representation by treating the height slices as feature channels. This allows us to reason in 2D BEV space, which brings significant efficiency gain with no performance drop. We simply use the RGB image as input for the camera stream. When we exploit the auxiliary task of depth completion, we additionally add a sparse depth image generated by projecting the LiDAR points to the image.

Network architecture: The backbone network follows a two-stream architecture [13] to process multi-sensor data. Specifically, for the image stream we use the pre-trained ResNet-18 [10] until the fourth convolutional block. Each block contains 2 residual layers with number of feature maps increasing from 64 to 512 linearly. For the LiDAR stream, we use a customized residual network which is deeper and thinner than ResNet-18 for a better trade-off between speed and accuracy. In particular, we have four residual blocks with 2, 4, 6, 6 residual layers in each, and the numbers of feature maps are 64, 128, 192 and 256. We also remove the max pooling layer before the first residual block to maintain more details in the point cloud feature. In both streams we apply a feature pyramid network (FPN) [14] with 1×1 convolution and bilinear up-sampling to combine multi-scale features. As a result, the final feature maps of the two streams have a down-sampling factor of 4 compared with the input.

On top of the last BEV feature map, we simply add a 1×1 convolution to perform dense 3D object detection. After score thresholding and oriented NMS, a small number of high-quality 3D detections are projected to both BEV

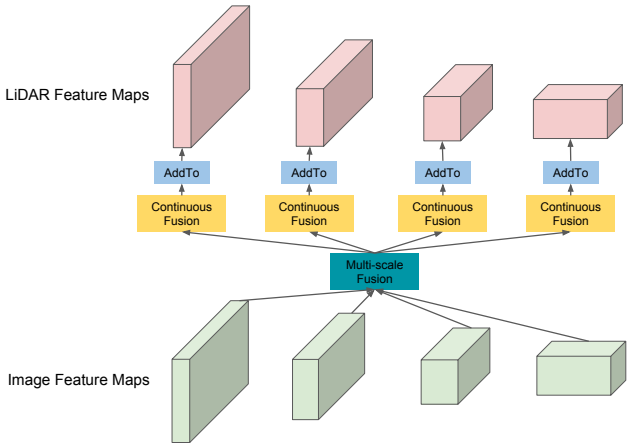

Figure 3. Point-wise feature fusion between LiDAR and image backbone networks. A feature pyramid network is used to combine multi-scale image feature maps, followed with a continuous fusion layer to project image feature map into BEV space. Feature fusion is implemented by element-wise summation.

space and 2D image space, and their ROI features are cropped from each stream's last feature map via precise ROI feature extraction. The multi-sensor ROI features are fused together and fed into a refinement module with two 256-dimension fully connected layers to predict the 2D and 3D box refinements for each 3D detection respectively.

Point-wise feature fusion: We apply point-wise feature fusion between the convolutional feature maps of LiDAR and image streams (as shown in Figure ref-fig:point). The fusion is directed from image steam to LiDAR steam to augment BEV features with information richness of image features. We gather multi-scale features from all four blocks in the image backbone network with a feature pyramid network. The resulting multi-scale image feature map is then fused to each block of the LiDAR BEV backbone network.

To fuse image feature map with BEV feature map, we need to find the pixel-wise correspondence between the two sensors. Inspired by [13], we use LiDAR points to establish dense and accurate correspondence between the image and BEV feature maps. For each pixel in the BEV feature map, we find its nearest LiDAR point and project the point onto the image feature map to retrieve the corresponding image feature. We compute the distance between the BEV pixel and LiDAR point as the geometric feature. Both retrieved image feature and the BEV geometric feature are passed into a Multi-Layer Perceptron (MLP) and the output is fused to BEV feature map by element-wise addition. Note that such point-wise feature fusion is sparse by nature of LiDAR observation. Later we explain how we exploit dense depth as pseudo LiDAR points to provide dense correspondence for dense point-wise fusion.

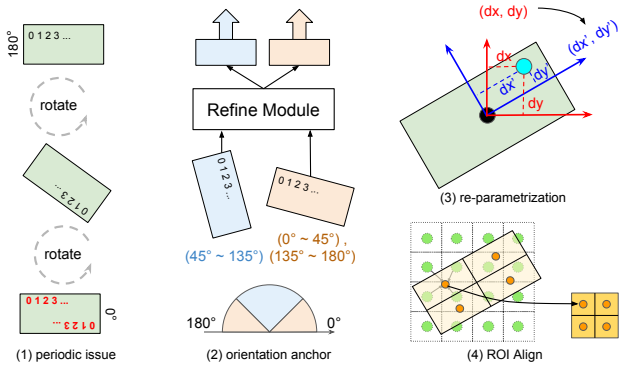

Figure 4. Precise rotated ROI feature extraction that takes orientation cycle into account. (1) The rotational periodicity causes reverse of order in feature extraction. (2) ROI refine module with two orientation anchors. An ROI is assigned to 0° or 90°. They share most refining layers except for the output. (3) The regression target of relative offsets are re-parametrized with respect to the object orientation axes. (4) A $n \times n$ sized feature is extracted using bilinear interpolation (we show an example with $n = 2$).

ROI-wise feature fusion: The motivation of the ROI-wise feature fusion is to further refine the localization precision of the high-quality detections in 2D and 3D spaces respectively. Towards this goal, the ROI feature extraction itself needs to be precise so as to properly predict the relative box refinement. By projecting a 3D detection onto the image and BEV feature maps, we get an axis-aligned image ROI and an oriented BEV ROI. We adopt ROIAlign [9] to extract features from an axis-aligned image ROI.

For oriented BEV ROI feature extraction, however, we observe two new issues (as shown in Figure 4). First, the periodicity of the ROI orientation causes the reverse of feature order around the cycle boundary. To solve this issue, we propose an oriented ROI feature extraction module with anchors. Given an oriented ROI, we first assign it to one of the two orientation anchors, 0 or 90 degrees. Each anchor has a consistent feature extraction order. The two anchors share the refinement net except for the output layer. Second, when the ROI is rotated, its location offset has to be parametrized in the rotated coordinates as well. In practice, we rotate the axis-aligned location offset to be aligned with the ROI orientation axes. Similar to ROIAlign[9], we extract bilinear interpolated feature into a $n \times n$ regular grid for the BEV ROI (in practice we use $n = 5$).

3.2. Multi-Task Learning for 3D Detection

In this paper we exploit two auxiliary tasks to improve 3D object detection, namely ground estimation and depth completion. They help in different ways: ground estimation provides geometric prior to canonicalize the LiDAR point clouds. Depth completion guides the image network

to learn better cross-modality feature representations, and further facilitates dense point-wise feature fusion.

3.2.1 Ground estimation

Mapping is an important task for autonomous driving, and in most cases the map building process is done offline. However, online mapping is appealing for that it decreases the system's dependency on offline built maps and increases the system's robustness. Here we focus on one specific subtask in mapping, which is to estimate the road geometry on-the-fly from a single LiDAR sweep. We formulate the task as a regression problem, where we estimate the ground height value for each voxel in the BEV space. This formulation is more accurate than plane based parametrization [3, 1], as in practice the road is often curved especially when we look far ahead.

Network architecture: We apply a small fully convolutional U-Net [25] on top of the LiDAR BEV representation to estimate the normalized voxel-wise ground heights at an inference time of 8 millisecond. We choose the U-Net architecture because it outputs prediction at the same resolution as the input, and is good at capturing global context while maintaining low-level details.

Map fusion: Given a voxel-wise ground estimation, we first extract point-wise ground height by looking for the point index during voxelization. We then subtract the ground height from each LiDAR point's Z axis value and generate a new LiDAR BEV representation (relative to ground), which is fed to the LiDAR backbone network. In the regression part of the 3D detection, we add the ground height back to the predicted Z term. Online ground estimation eases 3D object localization as traffic participants of interest all lay on the ground.

3.2.2 Depth completion

LiDAR provides long range 3D information for accurate 3D object detection. However, the observation is sparse especially at long range. Here, we propose to densify LiDAR points by depth completion from both sparse LiDAR observation and RGB image. Specifically, given the projected (into the image plane) sparse depth from the LiDAR point cloud and a camera image, the model outputs dense depth at the same resolution as the input image.

Sparse depth image from LiDAR projection: We first generate a three-channel sparse depth image from the LiDAR point cloud, representing the sub-pixel offsets and the depth value. Specifically, we project each LiDAR point (x, y, z) to the camera space, denoted as $(x_{cam}, y_{cam}, z_{cam})$ (the Z axis points to the front of the camera), where z_{cam} is the depth of the LiDAR point in camera space. We then project the point from camera space to image space, denoted as (x_{im}, y_{im}). We find the pixel (u, v) closest to (x_{im}, y_{im}), and compute $(x_{im} - u, y_{im} - v, z_{cam}/10)$[1] as the value of pixel (u, v) on the sparse depth image. For pixel locations with no LiDAR point, we set the pixel value to zero. The resulting sparse depth image is then concatenated with the RGB image and fed to the image backbone network.

Network architecture: The depth completion network shares the feature representation with the image backbone network, and applies four convolutional layers accompanied with two bilinear up-sampling layers afterwards to predict the dense pixel-wise depth at the same resolution as the input image.

Dense depth for dense point-wise feature fusion: As mentioned above, the point-wise feature fusion relies on LiDAR points to find the correspondence between multi-sensor feature maps. However, since LiDAR observation is sparse by nature, the point-wise feature fusion is sparse. In contrast, the depth completion task provides dense depth estimation per image pixel, and therefore can be used as "pseudo" LiDAR points to find dense pixel correspondence between multi-sensor feature maps. In practice, we use true and pseudo LiDAR points together in fusion and resort to pseudo points only when true points are not available.

3.3. Joint Training

We employ multi-task loss to train our multi-sensor detector end-to-end. The full model outputs object classification, 3D box estimation, 2D and 3D box refinement, ground estimation and dense depth. During training, we have detection labels and dense depth labels, while ground estimation is optimized implicitly by the 3D localization loss. There are two paths of gradient transmission for ground estimation. One is from the 3D box output where ground height is added back to predicted Z term. The other goes through the LiDAR backbone network to the LiDAR voxelization layer where ground height is subtracted from the Z coordinate of each LiDAR point.

For object classification loss L_{cls}, we use binary cross entropy. For 3D box estimation loss L_{box} and 3D box refinement loss L_{r3d}, we compute smooth $\ell 1$ loss on each dimension of the 3D object $(x, y, z, \log(w), \log(l), \log(h), \theta)$[2], and sum over positive samples. For 2D box refinement loss L_{r2d}, we similarly compute smooth $\ell 1$ loss on each dimension of the 2D object $(x, y, \log(w), \log(h))$, and sum over positive samples. For dense depth prediction loss L_{depth}, we sum $\ell 1$ loss over all pixels. The total loss is defined as follows:

$$Loss = L_{cls} + \lambda(L_{box} + L_{r2d} + L_{r3d}) + \gamma L_{depth}$$

where λ, γ are the weights to balance different tasks.

[1] We divide the depth value by 10 for normalization purpose.

[2] We normalize each dimension of the regression targets respectively.

Detector	Input Data		Time	2D AP (%)			3D AP (%)			BEV AP (%)		
	LiDAR	IMG	(ms)	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard
SHJU-HW [35, 7]		✓	850	90.81	90.08	79.98	-	-	-	-	-	-
RRC [20]		✓	3600	90.61	**90.23**	87.44	-	-	-	-	-	-
MV3D [5]	✓		240	89.80	79.76	78.61	66.77	52.73	51.31	85.82	77.00	68.94
VoxelNet [36]	✓		220	-	-	-	77.49	65.11	57.73	89.35	79.26	77.39
SECOND [32]	✓		50	90.40	88.40	80.21	83.13	73.66	66.20	88.07	79.37	77.95
PIXOR [34]	✓		**35**	-	-	-	-	-	-	87.25	81.92	76.01
PIXOR++ [33]	✓		**35**	-	-	-	-	-	-	89.38	83.70	77.97
HDNET [33]	✓		50	-	-	-	-	-	-	89.14	86.57	78.32
MV3D [5]	✓	✓	360	90.53	89.17	80.16	71.09	62.35	55.12	86.02	76.90	68.49
AVOD [12]	✓	✓	80	89.73	88.08	80.14	73.59	65.78	58.38	86.80	85.44	77.73
ContFuse [13]	✓	✓	60	-	-	-	82.54	66.22	64.04	88.81	85.83	77.33
F-PointNet [17]	✓	✓	170	90.78	90.00	80.80	81.20	70.39	62.19	88.70	84.00	75.33
AVOD-FPN [12]	✓	✓	100	89.99	87.44	80.05	81.94	71.88	66.38	88.53	83.79	77.90
Our MMF	✓	✓	80	**91.82**	90.17	**88.54**	**86.81**	**76.75**	**68.41**	**89.49**	**87.47**	**79.10**

Table 1. Evaluation results on the testing set of KITTI 2D, 3D and BEV object detection benchmark (car). We compare with previously published detectors on the leaderboard ranked by Average Precision (AP) in the moderate setting.

A good initialization is important for faster convergence. We use the pre-trained ResNet-18 network to initialize the image backbone network. For the additional channels of the sparse depth image at the input, we set the corresponding weights to zero. We also pre-train the ground estimation network on TOR4D dataset [34] with offline maps as labels and $\ell 2$ loss as objective function [33]. Other networks in the model are initialized randomly. We train the model with stochastic gradient descent using Adam optimizer [11].

4. Experiments

In this section, we first evaluate the proposed method on the KITTI 2D/3D/BEV object detection benchmark [8]. We also provide a detailed ablation study to analyze the gains brought by multi-sensor fusion and multi-task learning. We then evaluate on the more challenging TOR4D multi-class BEV object detection benchmark [34]. Lastly we provide qualitative results and discussions.

4.1. 2D/3D/BEV Object Detection on KITTI

Dataset and metric: KITTI's object detection dataset has 7,481 frames for training and 7,518 frames for testing. We evaluate our approach on "Car" class. We apply the same data augmentation as [13] during training, which applies random translation, orientation and scaling on LiDAR point clouds and camera images. For multi-task training, we also leverage the dense depth labels from the KITTI's depth dataset [28]. KITTI's detection metric is defined as Average Precision (AP) averaged over 11 points on the Precision-Recall (PR) curve. The evaluation criterion for cars is 0.7 Intersection-Over-Union (IoU) in 2D, 3D or BEV. KITTI also divides labels into three subsets (easy, moderate and hard) according to the object size, occlusion and truncation levels, and ranks methods by AP in the moderate setting.

Implementation details: We detect objects within 70 meters forward and 40 meters to the left and right of the ego-car, as most of the labeled objects are within this region. We voxelize the cropped point cloud into a volume of size $512 \times 448 \times 32$ as the BEV representation. We also center-crop the images of different sizes into a uniform size of 370×1224. We train the model on a 4 GPU machine with a total batch size of 16 frames. Online hard negative mining [27] is used during training. We set the initial learning rate to 0.001 and decay it by 0.1 after 30 and 45 epochs respectively. The training ends after 50 epochs. We train two models on KITTI: one without depth completion auxiliary task, and one full model. We submit the former one to the test server for fair comparison with other methods that are trained on KITTI object detection dataset only. We evaluate the full model in ablation study to showcase the performance gain brought by depth completion and dense fusion.

Evaluation results: We compare our approach with previously published state-of-the-art detectors in Table 1, and show that our approach outperforms competitors by a large margin in all 2D, 3D and BEV detection tasks. In 2D detection, we surpass the best image detector RRC [20] by 1.1% AP in the hard setting, while being $45\times$ faster. Note that we only use a small ResNet-18 network as the image stream backbone network, which shows that 2D detection benefits a lot from exploiting the LiDAR sensor and reasoning in 3D. In BEV detection, we outperform the best detector HDNET [33], which also exploits ground estimation, by 0.9% AP in moderate setting. The

Model	Multi-Sensor		Multi-Task			2D AP (%)			3D AP (%)			BEV AP (%)		
	pt	roi	map	dep	depf	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard
LiDAR only						93.44	87.55	84.32	81.50	69.25	63.55	88.83	82.98	77.26
+image	✓					+2.95	+1.97	+2.76	+4.62	+5.21	+3.35	+0.70	+2.39	+1.25
+map	✓		✓			+3.06	+2.20	+3.33	+5.24	+7.14	+4.56	+0.36	+3.77	+1.59
+refine	✓	✓	✓			+3.94	**+2.71**	+4.66	**+6.43**	+8.62	+12.03	+7.00	+4.81	+2.12
+depth	✓	✓	✓	✓		**+4.69**	+2.65	+4.64	+6.34	**+8.64**	**+12.06**	+7.74	+5.16	+2.26
full model	✓	✓	✓	✓	✓	+4.61	+2.67	**+4.68**	+6.40	+8.61	+12.02	**+7.83**	**+5.27**	**+2.34**

Table 2. Ablation study on KITTI object detection benchmark (car) training set with four-fold cross validation. *pt*: point-wise feature fusion. *roi*: ROI-wise feature fusion. *map*: online mapping. *dep*: depth completion. *depf*: dense fusion with dense depth.

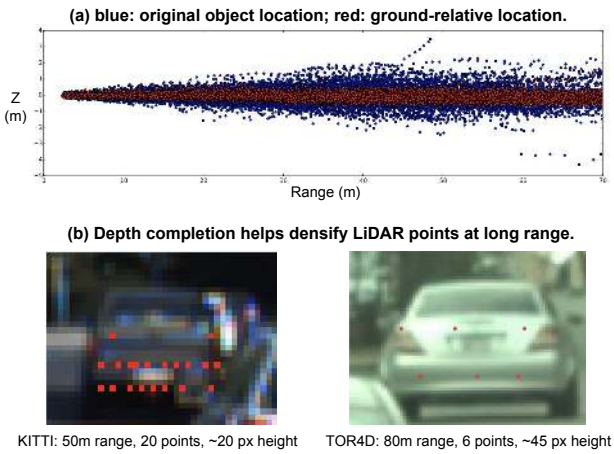

Figure 5. Object detection benefits from ground estimation and depth completion.

improvement mainly comes from multi-sensor fusion. In the most challenging 3D detection task (as it requires 0.7 3D IoU), we show an even larger gain over competitors. We surpass the best detector SECOND [32] by 3.09% AP in moderate setting, and outperform the previously best multi-sensor detector AVOD-FPN [12] by 4.87% AP in moderate setting. We believe the large gain mainly comes from the fully fused feature representation and the proposed ROI feature extraction for precise object localization.

Ablation study: To analyze the effects of multi-sensor fusion and multi-task learning, we conduct an ablation study on KITTI training set. We use four-fold cross validation and accumulate the evaluation results over the whole training set. This produces stable evaluation results for apple-to-apple comparison. We show the ablation study results in Table 2. Our baseline model is a single-shot LiDAR only detector. Adding image stream with point-wise feature fusion brings over 5% AP gain in 3D detection, possibly because image features provide complementary information on the Z axis in addition to the BEV representation of LiDAR point cloud. Ground estimation improves 3D and BEV detection by 1.9% and 1.4% AP respectively in moderate setting. This gain suggests that the geometric ground prior provided by online mapping is very helpful

Model	Vehicle		Pedestrian		Bicyclist	
	$AP_{0.5}$	$AP_{0.7}$	$AP_{0.3}$	$AP_{0.5}$	$AP_{0.3}$	$AP_{0.5}$
Baseline	95.1	83.7	88.9	80.7	72.8	58.0
+dep	95.6	84.5	88.9	81.2	74.3	62.2
+dep+depf	**95.7**	**85.4**	**89.4**	**81.8**	**76.3**	**63.1**

Table 3. Ablation study of BEV object detection with multi-task learning on TOR4D benchmark. The baseline detector is based on [13], with multi-sweep LiDAR and HD maps added to the input for better performance. *dep*: depth completion. *depf*: dense fusion using estimated dense depth.

for detection at long range (as shown in Figure 5 a), where we have very sparse 3D LiDAR observation. Adding the refinement module with ROI-wise feature fusion brings consistent improvements on all three tasks, which purely come from more precise localization. This proves the effectiveness of the proposed orientation aware ROI feature extraction. Lastly, the model further benefits in BEV detection from the depth completion task with better feature representations and dense fusion, which suggests that depth completion provides complementary information in BEV space. On KITTI we do not see much gain from dense point-wise fusion using estimated depth. We hypothesize that this is because in KITTI the captured image is at equivalent resolution of LiDAR at long range (as shown in Figure 5 b). Therefore, there isn't much juice to squeeze from image feature. However, on TOR4D benchmark where we have higher resolution camera images, we show in next section that depth completion helps not only by multi-task learning, but also dense feature fusion.

4.2. BEV Object Detection on TOR4D

Dataset and metric: The TOR4D BEV object detection benchmark [34] contains over 5,000 video snippets with a duration of around 25 seconds each. To generate the training and testing dataset, we sample video snippets from different scenes at 1 Hz and 0.5 Hz respectively, leading to around 100,000 training frames and around 6,000 testing frames. To validate the effectiveness of depth completion in improving object detection, we use images captured by camera with long-focus lens which provide richer information at long range (as shown in Figure 5 b). We evaluate on multi-class BEV object detection (*i.e.*

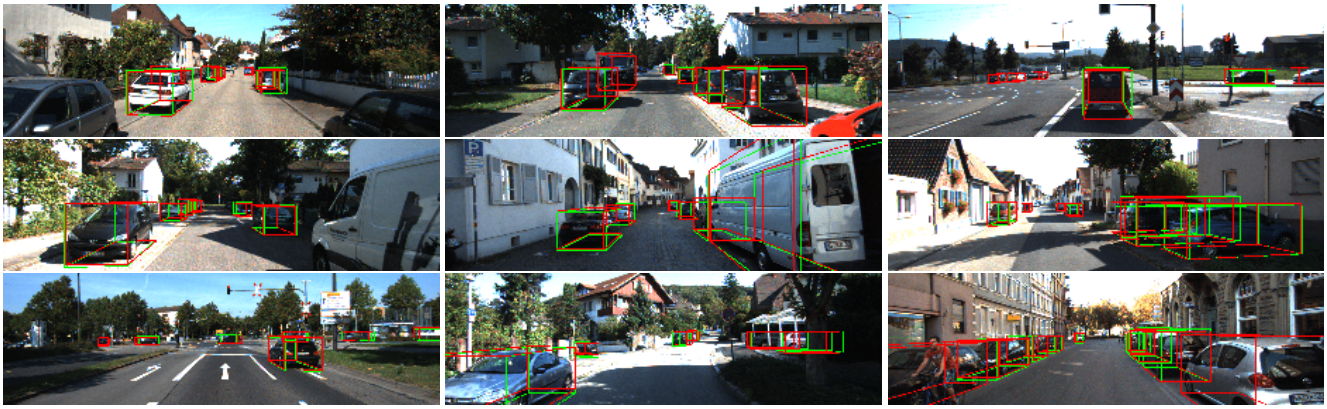

Figure 6. Qualitative results of 3D object detection (car) on KITTI benchmark. We draw object labels in green and our detections in red.

vehicle, pedestrian and bicyclist) with a distance range of 100 meters from the ego-car. We use AP at different IoU thresholds as the metric for multi-class object detection. Specifically, we look at 0.5 and 0.7 IoUs for vehicles, 0.3 and 0.5 IoUs for the pedestrians and cyclists.

Evaluation results: We re-produce the previously state-of-the-art detector ContFuse [13] on our TOR4D dataset split. Two modifications are made to further improve the detection performance. First, we follow FAF [16] to fuse multi-sweep LiDAR point clouds together at the BEV representation. Second, following HDNET [33] we incorporate semantic and geometric HD map priors to the detector. We use the improved ContFuse detector as the baseline, and apply the proposed depth completion with dense fusion on top of it. As shown in Table 3, the depth completion task helps in two ways: multi-task learning and dense feature fusion. The former increases the bicyclist AP by an absolute gain of 4.2%. Since bicyclists have the fewest number of labels in the dataset, having additional multi-task supervision is particularly helpful. In terms of dense fusion with estimated depth, the performance on vehicles is improved by over 5% in terms of relative error reduction (*i.e*. 1-AP) at 0.7 IoU. The reason for this gain may be that vehicles receive more additional feature fusion compared to the other two classes.

4.3. Qualitative Results and Discussion

We show qualitative 3D object detection results of the proposed detector on KITTI benchmark in Figure 6. The proposed detector is able to produce high-quality 3D detections of vehicles that are highly occluded or far away from the ego-car. Some of our detections are un-labeled cars in KITTI dataset. Previous works [5, 12] often follow state-of-the-art 2D detection framework (like two-stage Faster RCNN [22]) to solve 3D detection. However, we argue that it may not be the optimal solution. With thousands of pre-defined anchors, the feature extraction is both slow and inaccurate. Instead we show that by detecting 3D objects in BEV space, we can produce high-quality 3D detections via a single pass of the network (as shown in Table 2 by model variants without refinement), given that we fully fuse the multi-sensor feature maps via dense fusion.

Cascade approaches [17, 6] assume that 2D detection is solved better than 3D detection, and therefore use 2D detector to generate 3D proposals. However, we argue that 3D detection is actually easier than 2D. As we detect objects in 3D metric space, we do not have to handle the problems of scale variance and occlusion reasoning that would otherwise arise in 2D. Our model, which uses a pre-trained ResNet-18 as image backbone network and is trained from thousands of frames, surpasses F-PointNet [17], which exploits two orders of magnitude more training data (*i.e*. COCO dataset [15]), by over 7% AP in hard setting of KITTI 2D detection. Multi-sensor fusion and multi-task learning are highly interleaved. In this paper we provide a way to combine them together under the same hood. In the proposed framework, multi-sensor fusion helps learn better feature representations to solve multiple tasks, while different tasks in turn provide different types of clues to make feature fusion deeper and richer.

5. Conclusion

We have proposed a multi-task multi-sensor detection model that jointly reasons about 2D and 3D object detection, ground estimation and depth completion. Point-wise and ROI-wise feature fusion are applied to achieve full multi-sensor fusion, while multi-task learning provides additional map prior and geometric clues enabling better representation learning and denser feature fusion. We validate the proposed method on KITTI and TOR4D benchmarks, and surpass the state-of-the-art methods in all detection tasks by a large margin. In the future, we plan to expand our multi-sensor fusion approach to exploit other sensors such as radar as well as temporal information.

References

[1] Jorge Beltran, Carlos Guindel, Francisco Miguel Moreno, Daniel Cruzado, Fernando Garcia, and Arturo de la Escalera. Birdnet: a 3d object detection framework from lidar information. In *ITSC*, 2018. 5

[2] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, 2016. 1, 2

[3] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *NIPS*, 2015. 2, 5

[4] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *PAMI*, 2018. 1

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017. 1, 2, 6, 8

[6] Xinxin Du, Marcelo H Ang Jr, Sertac Karaman, and Daniela Rus. A general pipeline for 3d detection of vehicles. In *ICRA*, 2018. 1, 8

[7] Liangji Fang, Xu Zhao, and Shiquan Zhang. Small-objectness sensitive detection based on shifted single shot detector. *Multimedia Tools and Applications*, 2018. 6

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2, 6

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[11] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[12] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IROS*, 2018. 1, 2, 6, 7, 8

[13] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *ECCV*, 2018. 1, 2, 3, 4, 6, 7, 8

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 8

[16] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *CVPR*, 2018. 1, 8

[17] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018. 1, 2, 6, 8

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2

[19] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 2

[20] Jimmy Ren, Xiaohao Chen, Jianbo Liu, Wenxiu Sun, Jiahao Pang, Qiong Yan, Yu-Wing Tai, and Li Xu. Accurate single stage detector using recurrent rolling convolution. In *CVPR*, 2017. 6

[21] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun. Sbnet: Sparse blocks network for fast inference. In *CVPR*, 2018. 2

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 8

[23] Zhile Ren and Erik B Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *CVPR*, 2016. 2

[24] Zhile Ren and Erik B Sudderth. 3d object detection with latent support surfaces. In *CVPR*, 2018. 2

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 5

[26] Alexander G Schwing, Sanja Fidler, Marc Pollefeys, and Raquel Urtasun. Box in the box: Joint 3d layout and object reasoning from single images. In *ICCV*, 2013. 2

[27] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 6

[28] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *3DV*, 2017. 6

[29] Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Holistic 3d scene understanding from a single geo-tagged image. In *CVPR*, 2015. 2

[30] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *CVPR*, 2018. 2

[31] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *CVPR*, 2018. 1

[32] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018. 6, 7

[33] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *CoRL*, 2018. 2, 6, 8

[34] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *CVPR*, 2018. 1, 2, 6, 7

[35] Shiquan Zhang, Xu Zhao, Liangji Fang, Fei Haiping, and Song Haitao. Led: Localization-quality estimation embedded detector. In *ICIP*, 2018. 6

[36] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 1, 2, 6