Survey on RGB LiDAR Fusion

Ajay Chawda[1] and Michael Fürst[2]

[1] a_chawda19@cs.uni-kl.de
[2] michael.fuerst@dfki.de

Abstract. In this paper, we study the existing sensor fusion methods for RGB and Lidar data. Our goal is to categorize the methods into early, sequential, late and slow fusion, also providing recent examples for better insight to our categories. In addition to that we compare the methods based on empirical results and contributions to find the best available technique.

Keywords: RGB, LiDAR, Fusion, 3D object detection

1 Introduction

Self driving vehicles are an important area of research in the field of computer vision. They potentially increase safety by decreasing dependency on human factor. An important challenge is to perceive the scene in real time to make instantaneous decisions. A RGB image provides fine grained texture and color information but lacks depth information. On the other hand, a Lidar point cloud has accurate 3D localization but suffers from low resolution and texture information. 3D object detection is difficult using only image data due to depth loss and for Lidar it becomes tough at long distances due to sparsity of points. Therefore we overcome the drawbacks of each model by fusion of both modalities. Our study characterizes the existing methods into early, sequential, late and slow fusion. KITTI provides synced Lidar point clouds and front-view camera images while nuScenes is annotated with 3D bounding boxes, both datasets are used for evaluation. Our comparison provides us with evidence that sequential fusion is the current best method for fusion of RGB and Lidar data.

RGB only: YOLO [6] is a detection without proposal approach used in 2D object detection. Other methods with proposals include R-CNN, Faster R-CNN etc. which use Region proposal networks(RPN) to generate proposals in the image. The proposals are then used as input to perform object detection. R-CNN is slower in comparison to Faster R-CNN as generating proposals over the complete image takes a lot of time. In Faster R-CNN proposals are predicted from features of the image. YOLO is faster compared to proposal methods but it has a higher localization error of bounding boxes. For 3D object detection using only RGB information from image is not sufficient. We also need to estimate the depth information from the image which is an overhead while performing 3D detection. Using only RGB information is not desirable for autonomous driving as we want to localize objects in 3D space. Therefore we use RGB-D images for 3D object detection.

LiDAR only: Lidar object detection approaches discretize point clouds and performs convolutions either in bird's eye view (BEV) [2] or in native range view (RV) [4]. BEV is a 3 channel image encoding height, intensity and density information. The height channel represents the maximum height, the intensity channel encodes the mean intensity of all points, the third channel represents the density of points in each cell. For range view, we project the lidar point cloud onto the cylindrical plane to generate a dense front view map. BEV methods have higher performance in terms of 3D object detection compared to RV methods. RV methods are computationally efficient because of sparse representation of Lidar data in BEV. RV methods are also better in detection of small objects due to BEV voxelization removing fine-grain details. Lidar is also represented as PointNet [5]

computed on point cloud as an unordered set. At far distances the Lidar data becomes sparse due to which localization becomes difficult. As the data becomes sparse, it is tough to detect bikes, pedestrians and other small objects which are relevant to autonomous driving. In case of detection failure of our model for small objects can also lead to disastrous consequences.

RGB-LiDAR Fusion: The image information is merged with Lidar point cloud to improve upon the deficiencies of RGB and Lidar only data. Combining RGB information with Lidar points improves detection performance at far distances. Addition of image data improves the semantic information of the 3D points. Images project representation of world on the camera plane whereas Lidar points explain world's 3D structure. In elemental terms, addition of depth information as an extra channel to camera image can be defined as fusion, in contrast adding semantic information from camera image to 3D point cloud will be defined same.

Our survey categorizes fusion methods into four approaches : Early fusion, Sequential fusion, Late fusion and Slow fusion. Earlier categorizations in previous research papers include Meyer et al. [3] characterization of fusion methods into *2D to 3D* approaches, where first 2D object detection is performed and then using Lidar data, the 2D detection is converted to 3D boxes. *Proposal fusion*, methods propose 3D bounding boxes either by sampling over output space or predicting from Lidar data. Then 3D proposals are used to extract features from RGB model and Lidar model and combine them. *Dense fusion*, methods fuse image and Lidar features into a common frame and perform single stage 3D object detection. Similarly Vora et al. [7] characterizes into *Object-centric fusion*, fusion occurs at proposal level with ROI pooling from a shared set of 3D proposals in each modality. *Continuous feature fusion*, allows feature information to be shared across all levels of the network architecture. *Explicit transform*, methods transform the image to bird's eye view representation and perform fusion. *Detection Seeding*, semantics are extracted from the image and used to seed detection in the point cloud.

Based on our categories we have characterized Explicit transform as *Early fusion*. Detection Seeding, 2D to 3D as *Sequential fusion*. Proposal fusion, Dense fusion, Object-centric fusion as *Late fusion* and Continuous feature fusion as *Slow fusion*.

2 Fusion Approaches

2.1 Early Fusion

Early fusion is fusion of RGB and Lidar data in early stages of the detection network. Our data consists of raw values from image and Lidar point cloud. The process is depicted in Fig 1 a). A Lidar point cloud is transformed to RGB-D data. 3D detection is performed using RGB-D detection algorithms. Fig 1 b). Depth is estimated from RGB (Stereo/Mono) images. A 3D point cloud is generated based on the depth information extracted from image data. 3D object detection is performed by applying Lidar based detection algorithm.

Wang et al. [9] states that the difference between 3D object detection using Lidar and stereo images is not due to depth value estimation but poor representation of 3D information. To support the claim, the estimated depth map from stereo or monocular images is converted to 3D point

Fig. 1: *Early Fusion*. a) From Lidar to RGB-D b) From RGB to Lidar

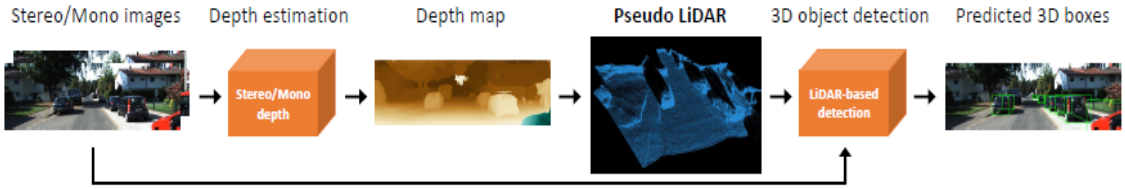

Fig. 2: Pseudo-LiDAR [9] predicts depth map from stereo or monocular images as input. The estimated depth-map is then projected onto the LiDAR co-ordinate system. The point cloud is then processed with Lidar based detection methods for object detection. Figure is taken from [9].

cloud, which is referred as pseudo-LiDAR. The pseudo-LiDAR representation is then trained on existing Lidar based detection methods. The use of pseudo-LiDAR increases the accuracy of 3D object detection. This paper proposes pseudo-LiDAR as a new representation of estimated depth. The proposed method is similar to Fig 1 a). Depth estimation is performed using stereo disparity estimation algorithm that takes a pair of left-right images as input, captured from a pair of cameras with a horizontal offset and outputs a disparity map of the same size as either one of the two input images. Without loss of generality, we assume the depth estimation algorithm treats the left image as reference and records the horizontal disparity to right image for each pixel. As Wang et al. [9] emphasizes more towards depth estimation, we can argue that it is not an exact representation of early fusion method. But it is relevant to our categorization because of similarity in pipelines of the proposed method (Fig 2) and early fusion category.

2.2 Sequential Fusion

Fusion of RGB and Lidar in successive steps in the network where features from convolution of either RGB or Lidar system are extracted, and on it a priori estimated points of the other system are projected. Fig 3 a) shows semantic information of Lidar point cloud is extracted using a CNN. Depth information from RGB images is estimated a priori and projected onto the output of CNN. The sequentially fused point cloud is used as input for Lidar detection algorithm for 3D object detection. b) Lidar point cloud information is projected on the output of image CNN. The fused point cloud is passed through Lidar detection method for object detection.

The challenge overcome by PointPainting [7] is viewpoint misalignment. Most of the sensor modalities are in range view but some methods use bird's eye view for Lidar point cloud. Due to which the fusion of RGB and Lidar features provides worse results than Lidar only methods. So, it does not have constraints for object detection architectures and also does not suffer from depth blurring. Depth blurring is poor representation of pixels at far distances. After projection of Lidar points onto the image, the segmentation scores for the relevant pixel are added to the Lidar point to create the painted Lidar point. If the field of view of two cameras overlap, there will be some points that will project on two images simultaneously and we randomly choose the segmentation score vector from one of the two images.

Fig. 3: *Sequential fusion*. Depending on input to network sequential fusion occurs from a) Lidar to RGB b) RGB to Lidar.

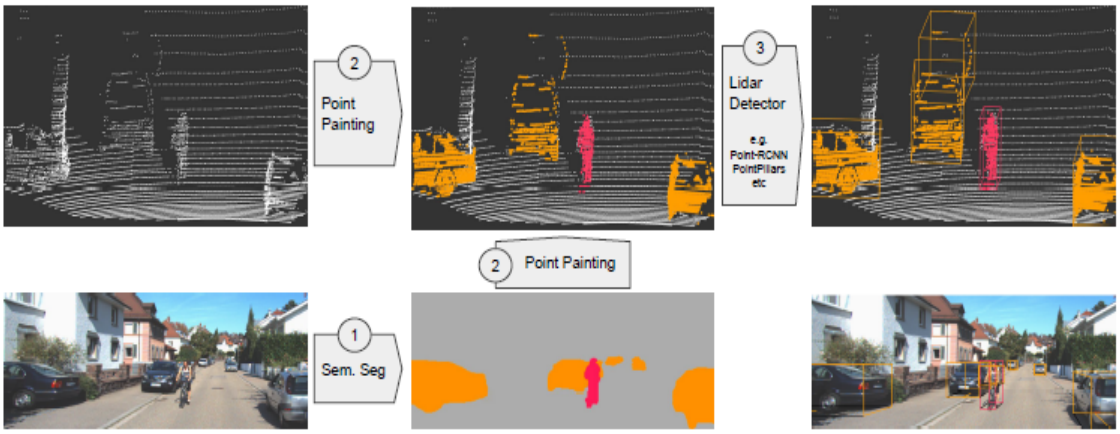

Fig. 4: (1) A semantic segmentation network is used to generate segmentation scores. (2) In Point-Painting step, Lidar points are projected into the output of first step. (3) A Lidar based detector is used to obtain 3D detection. Figure is taken from [7].

PointPainting [7] architecture in Fig 4 shows that it consists of three main stages and it is sequential by design, which means it cannot be optimized always for 3D object detection. Empirically PointPainting is more effective than other proposed fusion methods. The major contribution of the proposed method is *General* - achieving significant improvements when used with top Lidar-only methods on the KITTI and nuScenes benchmarks, *Accurate* - the painted version of PointRCNN achieves state of the art on the KITTI benchmark, *Robust* - the painted versions of PointRCNN and PointPillars improved performance on all classes on the KITTI and nuScenes test sets, respectively and *Fast*- low latency fusion can be achieved by pipelining the image and Lidar processing steps. The larger improvements were on the challenging scenarios of pedestrian and cyclist detection, but there was also improvement on car detection.

2.3 Late Fusion

Late fusion occurs late in the network at two different levels. Illustration in Fig 5 a) *Feature level fusion* - Features are extracted from image and Lidar data using convolutional neural network. The extracted features are merged together generating fused feature map. 3D object detection is performed using Lidar detection algorithm. b) *Output level fusion*. Generate 2D bounding boxes and 3D bounding boxes from image and Lidar data respectively. Fusion of bounding boxes provides 3D detection at output level.

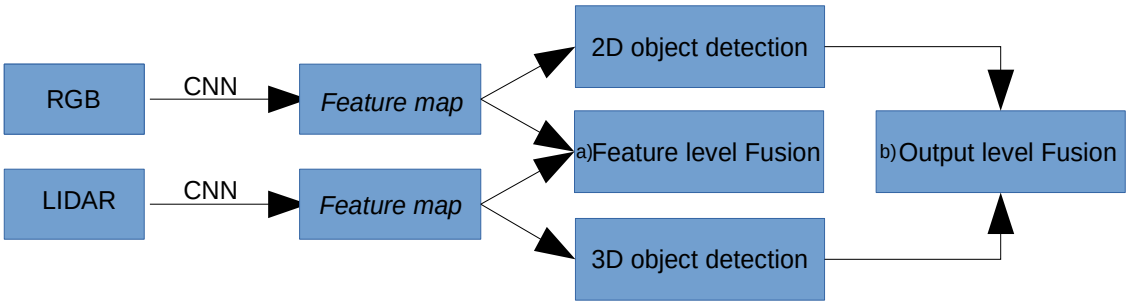

Fig. 5: *Late fusion*. Concatenation of RGB and Lidar modalities later in the architecture after feature extraction or bounding box detection.

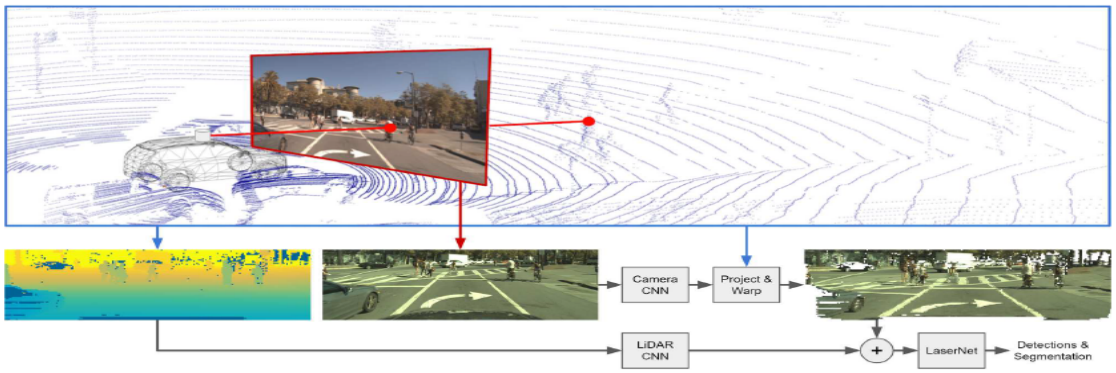

Fig. 6: Features are extracted by passing image points and Lidar points through respective CNN. The feature maps are fused together and passed to LaserNet for object detection. Figure is taken from [3].

Meyer et al. [3] argues that the existing fusion methods achieve good performance but are inefficient in runtime, making it difficult to integrate into autonomous driving system. This paper shows that concatenation of RGB values with their Lidar points does not improve performance. Therefore, features are extracted from RGB image using CNN and fused with Lidar points in range view as shown in Fig 6. Fusion of raw RGB data leads to a significant amount of information being discarded. So, Meyer et al. fuses learned features extracted by a CNN from the RGB image. This allows the network to capture higher level concepts from the image data, so that more information is conveyed when fused with the Lidar image. It improves the semantic segmentation of 3D points at long ranges. Performance at long ranges is attributed to use of supplemental 2D data where Lidar points are sparse. Smaller objects like pedestrians and bikes also observe increase in detection accuracy because of use of RV instead of BEV which requires voxelization of 3D points, resulting in removal of fine grain detail.

Yoo et al. [10] states that the spatial feature maps obtained from each modality are represented by significantly different views in the camera and world coordinates. So it is not an easy task to combine two heterogeneous feature maps without loss of information. To overcome the challenge 3D-CVF [10] employs a cross view feature fusion strategy depicted in Fig 7. The method uses *auto-calibrated projection*, for transformation of 2D features to spatial feature map with highest correspondence to Lidar features in BEV view. *Gated feature fusion network* is used to concatenate the camera and Lidar features according to region.The spatial attention maps are applied to both feature maps to weight the information from each modality depending on their contributions to the

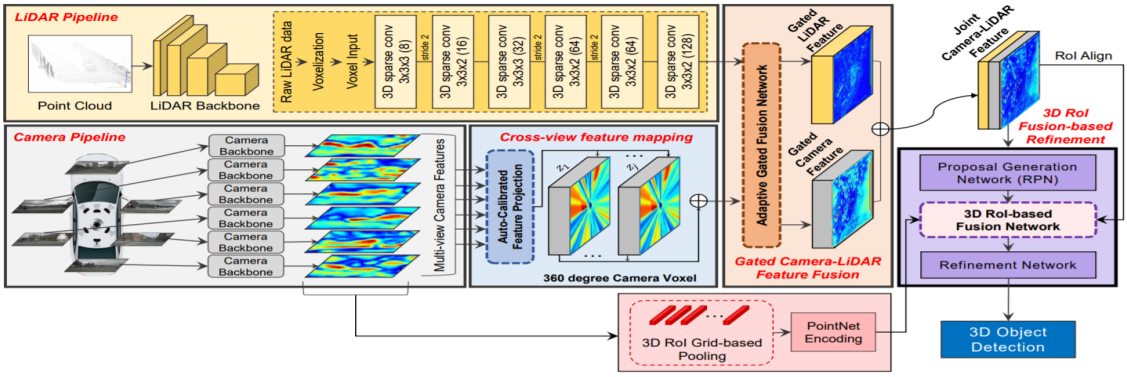

Fig. 7: Architecture of 3D-CVF.Figure is taken from [10].

detection task. The adaptive gated fusion network produces the joint camera-Lidar feature map, which is delivered to the 3D ROI fusion-based refinement block. Fusion also occurs in the proposal refinement stage. The camera feature obtained after 3D ROI grid pooling is fused with BEV feature in proposal refinement stage.

2.4 Slow Fusion

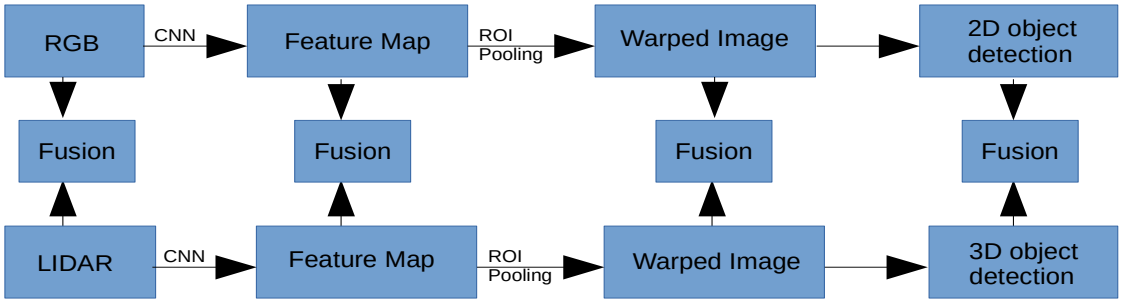

Fig. 8: *Slow fusion*. Sensor fusion from raw pixels to bounding box at each level.

Slow fusion refers to fusion of RGB data and Lidar point cloud at multiple levels in the network throughout the architecture. Multi-level fusion refers to concatenation of RGB and Lidar points at input using raw pixel values, then at feature level after passing through CNN, and later at output level after bounding box detection. Fig 8 shows slow fusion in a network.

ContFuse [2] proposes a 3D object detector that fuses image features by learning to project them into BEV space. This paper outlines challenges like output of fusion in camera space is not suitable for autonomous driving, voxelization is not efficient in memory and computation and color information is lost over pixels, loss of geometric information due to ROI pooling. In order to create a dense BEV feature map ContFuse uses continuous convolutions [8] to extract information from nearest image features in BEV space.Deep parametric continuous convolution is a learnable operator that operates over non-grid-structured data. The idea is to extend the standard grid-structured convolution to non-grid-structured data, while retaining high capacity and low complexity to exploit multi-layer perceptrons as parameterized kernel functions for continuous convolution. The continuous fusion

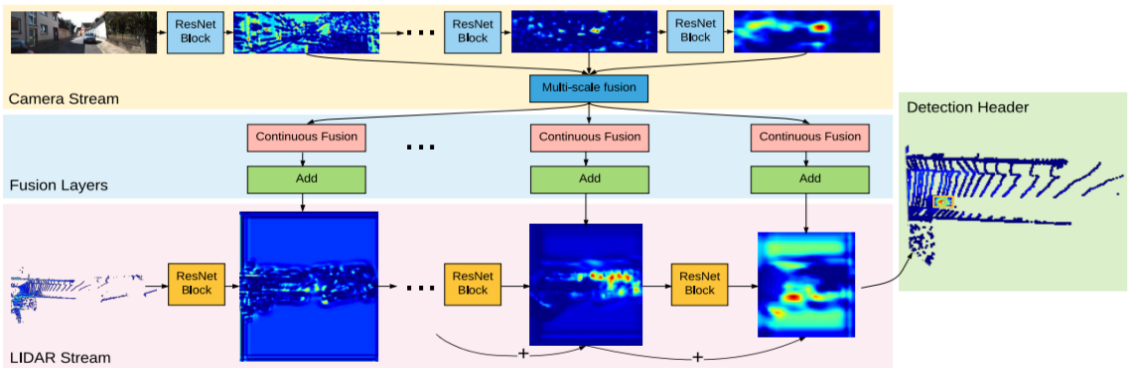

Fig. 9: Camera Stream is the backbone network of input image. Lidar stream is backbone of Lidar pointcloud. Continuous fusion layer is used to fuse image features into Lidar features. Detection header is used to perform object detection.Figure is taken from [2].

layer in Fig 9 plays an important role in overcoming the sparse observations and managing spatially discrete features in camera view image. The target of continuous fusion layer is to create a dense representation of BEV feature map which can be fused with feature maps from Lidar. The problem is that not all discrete pixels on BEV space are observable in the camera. Therefore, ContFuse uses information from K nearest Lidar points using euclidean distance to interpolate features at target pixel. The advantage of parametric continuous convolution is that it utilizes the concept of standard convolution without loss of geometric information. Contributions of ContFuse include more flexibility to collect information from multiple neighbours due to use of continuous convolutions and it is memory-efficient. Detection Header is used for real-time efficiency. A 1×1 convolutional layer is computed over the final BEV layer to generate the detection output. A Non-Maximum Suppression (NMS) layer follows to generate the final object boxes based on the output map.

2.5 Mixed Fusion

Combination of any two or more methods from above fusion methods can be termed as mixed fusion. MMF [1] is an example where early fusion and slow fusion are integrated to form a mixed fusion architecture. Early fusion is used for ground estimation and incorporate geometric ground information prior to Lidar data. Slow fusion performs object detection in the network.

3 Comparison

Table 1: Comparison on 3D average precision and BEV average precision for Car class.

Method	Fusion	Dataset	3D AP			BEV AP		
			easy	moderate	hard	easy	moderate	hard
Pseudo-LiDAR [9]	Early	KITTI	88.7	**84.0**	75.3	81.2	70.4	62.2
Painted PointRCNN [7]	Sequential	KITTI	88.38	77.74	**76.76**	92.45	**88.11**	**83.36**
3D-CVF [10]	Late	KITTI	**88.84**	79.72	72.80	-	-	-
LaserNet++ [3]	Late	AT4GD	-	-	-	94.96	85.42	70.31
ContFuse [2]	Slow	KITTI	86.32	73.25	67.81	**95.90**	87.39	82.43
MMF [1]	Mixed	KITTI	86.81	76.75	68.41	89.49	87.47	79.10

From Table 1 we observe best results for 3D average precision in Late fusion method in easy and hard difficulties for Car class in KITTI dataset. Similarly, our observation for BEV average precision is Sequential fusion method in the moderate and hard difficulties. ContFuse performs better than other methods for easy class in BEV, while PointPainting for hard class in 3D. Considering all values for KITTI dataset we narrow our choices upto Late and Sequential fusion for best method.

Table 2: Comparison on nuScenes dataset between 3D-CVF and Point Painting methods.

Method	mAP	Car	Ped.	Bus	Barrier	Traffic Cone	Truck	Trailer	Motorcycle
3D-CVF [10]	42.17	**79.69**	71.28	**54.96**	47.10	40.82	**37.94**	36.29	37.18
Painted PointPillars+ [7]	**46.4**	77.9	**73.3**	36.1	**60.2**	**62.4**	35.8	**37.3**	**41.5**

From Table 2 we observe mean average precision of Painted PointPillars+ using sequential fusion method is better compared to 3D-CVF using late fusion on nuScenes dataset. Precision values from

table 2 help us further in choosing best method for fusion as sequential fusion outperforms late fusion for smaller objects such as pedestrians, traffic cone, motorcycle and barrier.

4 Conclusion

Based on observations from KITTI and nuScenes dataset and contribution to fusion methods we would like to conclude sequential fusion as current best method for fusion of RGB and Lidar data. This is largely due to the fact that PointPainting is able to detect smaller objects with more precision than other methods while still performing with good accuracy for large objects in comparison to other methods studied in the survey.

Our categorizations are unique and proceed with comprehensive description of sensor fusion methods. We observed that challenges of early fusion due to fusion of raw pixels is eliminated by late fusion using feature level fusion. Sequential fusion overcomes difficulty in detection of small objects. Slow fusion is used by ContFuse for efficient detection. We might observe increased accuracy by using combination of fusion methods in future architectures. This is evident from MMF [1] in Table 1 which uses pseudo-LiDAR and slow fusion and had best results before 3D-CVF and PointPainting were introduced. We will leave the argument for future studies.

References

1. Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
2. Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
3. Gregory P. Meyer, Jake Charland, Darshan Hegde, Ankit Laddha, and Carlos Vallespi-Gonzalez. Sensor fusion for joint 3d object detection and semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.
4. Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and carl K. Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
5. Charles R Qi, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
6. Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
7. Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
8. Shenlong Wang, Simon Suo, Wei-Chu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
9. Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Killian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
10. Jin Hyoek Yoo, Yeocheol Kim, Ji Song Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. *Computing Research Repository (CoRR),abs/2004.12636v1*, 2020.