Recent Advances in 3D Object Detection in the Era of Deep Neural Networks: A Survey

Mohammad Muntasir Rahman, Yanhao Tan, Jian Xue, *Member, IEEE,* and Ke Lu, *Member, IEEE*

***Abstract*—With the rapid development of deep learning technology and other powerful tools, 3D object detection has made great progress and become one of the fastest growing field in computer vision. Many automated applications such as robotic navigation, autonomous driving, and virtual or augmented reality system require estimation of accurate 3D object location and detection. Under this requirement, many methods have been proposed to improve the performance of 3D object localization and detection. Despite recent efforts, 3D object detection is still a very challenging task due to occlusion, viewpoint variations, scale changes, and limited information in 3D scenes. In this paper, we present a comprehensive review of recent state-of-the-art approaches in 3D object detection technology. We start with some basic concepts, then describe some of the available datasets that are designed to facilitate the performance evaluation of 3D object detection algorithms. Next, we will review the state-of-the-art technologies in this area, highlighting their contributions, importance, and limitations as a guide for future research. Finally, we provide a quantitative comparison of the results of the state-of-the-art methods on the popular public datasets.**

***Index Terms*—3D object detection, deep neural networks, RGB-D data, LiDAR data, point cloud, deep learning.**

I. Introduction

OBJECT detection is one of the fundamental problems in computer vision. Recently, the rapid success of deep neural networks has significantly boosted the development of various automation-based systems such as mobile robots, autonomous driving and virtual or augmented reality systems that eagerly demand 3D understanding. 3D object detection helps to extract a geometric understanding of physical objects in 3D space. In the past years, although great progress has been made in image-based 2D object detection [1]–[3] and instance segmentation [4] tasks, 3D object detection is less explored in the literature. Beyond 2D detection, accurate 3D understanding in the 3D world remains an open challenge due to the complex interactions between objects, heavy occlusions, cluttered scene, viewpoint and scale variations, and limited information provided by 3D data. Besides, the representation of 3D data itself is more complicated, requiring higher computation and memory requirements due to an extra dimension is added. The recent advances in 3D sensing equipment (*e.g.*, LiDAR, Radar) sensors) and the availability of low-cost devices (*e.g.*, Microsoft Kinect, Xtion Pro-live, etc.) have made the capturing of 3D data more convenient than ever before. Leveraging the power of the 3D sensing technology along with the rapidly evolving deep learning technology, 3D object detection research has sparked a new interest in addressing challenges, making it a constantly changing research field in computer vision. This rapid growth has made it difficult to keep track methods dealing with the technologies of 3D object detection.

Our survey focuses on describing and analyzing recent competing deep learning-based 3D object detection methods. There exist few surveys on 3D data [5] and 2.5/3D indoor scene understanding [6], covering only existing methods in a specific domain, and since 3D object detection is a rapidly growing field, it may lack the idea of the state-of-the-art methods that could provide some new solutions and directions. In this paper, we systematically and comprehensively review up-to-date methods of 3D object detection and determine the potential pros and cons of these methods. We've listed the recently proposed solutions, but ignored the discussion of traditional approaches, so readers can see the cutting-edge technology of the 3D object detection more easily. Based on the input modality, we divide the existing methods into three categories: image-based processing methods, point cloud-based processing methods and multimodal fusion-based methods. Finally, we provide comparisons of 3D object detection methods and discuss current research issues and future research directions. Therefore, we believe that our work will be served as a helpful reference and make a significant contribution to the research community.

The key contributions of our work are as follows:

- We start with the basic concepts of 3D bounding box encoding techniques, 3D data representation and sensing modalities, and then introduce some existing datasets that might be helpful for future deep learning based 3D object detection projects.
- We provide in-depth systematic reviews of recent 3D object detection methods, their origins, contributions, and limitations.
- We present comparative results for the described methods on popular datasets and show their performances.
- We point out the potential research challenges, gaps and future research directions.

This work was supported by National Key R&D Program of China [Grant number 2017YFB1002203]; the National Natural Science Foundation of China [Grant number 61972375, 61671426, 61731022, 61572077, 61871258]; the Instrument Developing Project of the Chinese Academy of Sciences [Grant number YZ201670]; the Scientific Research Program of Beijing Municipal Education Commission [Grant number KZ201911417048]; the Beijing Natural Science Foundation [Grant number 4182071]; and the University of Chinese Academy of Sciences (Y95401YXX2). Thanks to CAS-TWAS Presidents Fellowship for supporting M. Muntasir Rahman as a doctoral student [fellowship number 2015CTF075].

M. Muntasir Rahman, Y. Tan, J. Xue, and Ke Lu are with the School of Engineering Sciences, University of Chinese Academy of Sciences, No.19A Yuquan Road, Beijing 100049, China. M. Muntasir Rahman is also with the Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh. E-mail: *muntasir@mails.ucas.ac.cn*, *tanyanhao15@mails.ucas.edu.cn*, *xuejian@ucas.ac.cn*, and *luk@ucas.ac.cn*. Corresponding author: Ke Lu.

Manuscript received April 04, 2019; revised September 30, 2019.

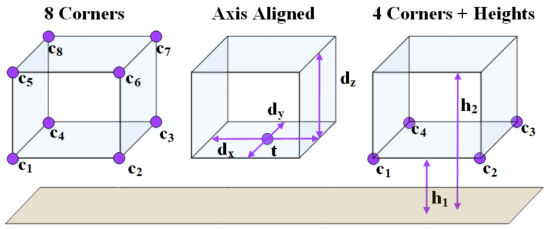

Fig. 1. A visual comparison between the 8-corners method [8], the axis aligned 3D center offset method [7], and 4-corner-2-height method [9].

II. Background

In order to properly understand how modern deep learning architectures solve 3D object detection problems, this section provides a background concepts about the 3D object detection technology. A typical object detector takes an image as input and draws a bounding box around the detected object along with the class level in the image plane. However, lifting from 2D object detection to 3D requires an additional estimation of the size, orientation and position of the object in 3D space. More specifically, the 3D object detection system is intended to place an oriented 3D bounding box on an object in 3D space, which should tightly fit the object of interest. In the following, first we define 3D bounding box encoding techniques. Next, we briefly summarize typical sensors with their sensing modalities. Then we introduce some popular dataset for 3D object detection.

A. 3D Bounding Box Encoding Methods

A 3D bounding box is nothing but an oriented rectangular cuboid drawn in 3D space. Three types of methods are commonly used to represent a 3D bounding box, such as axis aligned 3D center offset method [7], 8-corners method [8] and 4-corner-2-height method [9] as shown in Fig. 1. Similar to modern 2D object detector [1], [10], 3D center offset method [7] parameterized a 3D bounding box as $(\Delta x, \Delta y, \Delta z, \Delta h, \Delta w, \Delta l, \Delta\theta)$, where $(\Delta x, \Delta y, \Delta z)$ is the center coordinate of the 3D bounding box, $(\Delta h, \Delta w, \Delta l)$ is the height, width and length of the box, respectively, and $\Delta\theta$ is the yaw angle of the bounding box. The pitch and roll angles are considered to be zero, or to be of less importance for this task. An illustration of how a 3D bounding box is presented in 3D space is shown in Fig. 2. In 8-corners method [8], the 3D bounding box is parameterized as $(\Delta x_0 \dots \Delta x_7, \Delta y_0 \dots \Delta y_7, \Delta z_0 \dots \Delta z_7)$, where each (x, y, z) represent each corner of the 3D bounding box. The 4-corner-2-height method [9] encodes the 3D bounding box with 4 corners and 2 height values that represent the top and bottom corner offsets from the ground plane and defined as $(\Delta x_1 \dots \Delta x_4, \Delta y_1 \dots \Delta y_4, \Delta h_1, \Delta h_2)$.

B. 3D Data Representation and Sensing Modalities

Data plays a key factor in understanding the 3D world around us. 3D data can have multiple data representations, such as voxel representation [11], point cloud [12], multi-view images [8], depth channel encoding [13], polygonal mesh,

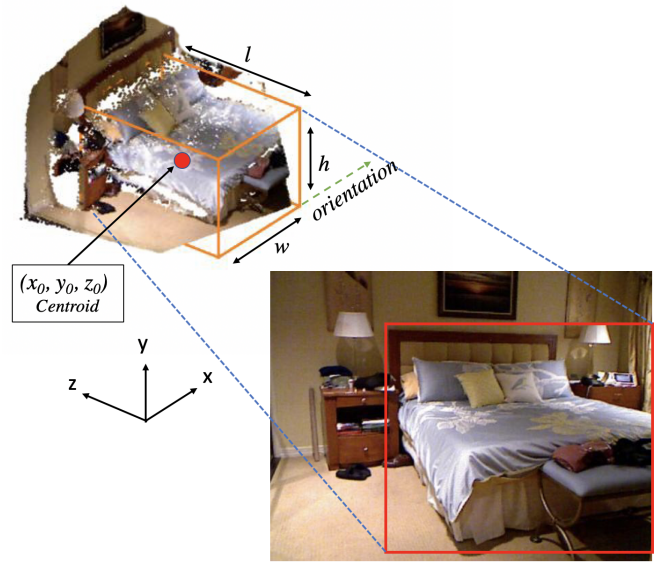

Fig. 2. An illustration of a 3D bounding box in 3D space.

truncated signed distance function (TSDF) [7], constructive solid geometry (CSG), stixels [14], octree [15], and primitive-based representation. Data is captured through various sensors that are now a standard feature in many automobiles, drones, robots and smartphones. Next, we describe the various sensors used in 3D object detection.

1) Monocular Cameras: Images captured by monocular cameras provides an RGB two-dimensional image of the three-dimensional world containing the object's color, texture and appearance information. Monocular cameras are passive and do not interfere with other systems because they do not emit signals for measurements. Furthermore, monocular camera images are widely used in 2D computer vision with superior state-of-the-art algorithms. However, these cameras are sensitive to lighting and weather conditions, and cannot differentiate an object when it has similar colors and textures in the foreground and background. Depth information is difficult to obtain from a single camera, but a stereo vision camera can solve this problem. Other types of cameras gaining interest for 3D object detection including thermal cameras, which are more robust to lightining conditions since they detect infrared radiation.

2) LiDARs: LiDARs (Light Detection And Ranging) are active sensors because they emit infrared light waves to calculate the surrounding depth information in the form of 3D points. They can calculate the distance accurately under 200 meters. LiDARs are robust against various light illuminations, and less affected by different weather conditions such as fog, snow or rain than visual cameras. However, LiDARs can only measure distances, and cannot capture texture information of objects. Recently, flash LiDARs were introduced which can provide similar information like a camera image.

3) RGB-D Cameras: Generally, a basic RGB-D camera consist of a RGB camera, an infrared camera and an IR projector, which can capture RGB images along with per-pixel depth information using infrared camera and IR projector. The

RGB-D cameras are active cameras and can provide both color, texture and depth information. Some popular RGB-D cameras are Microsoft Kinect, Asus Xtion Pro, Intel RealSense etc. RGB-D cameras are widely used in indoor 3D object detection and 3D reconstruction task. However, the limitation is that RGB-D cameras can measure distance below 8 miters.

4) Radars: Radars (Radio Detector And Ranging) are also active sensors that emit radio waves reflected back from an obstacle and estimate the time of each reflection by the Doppler effect. They are also robust to different lighting and weather conditions. The limitation is that Radar can interfere with other systems, and it is very challenging to classify objects via Radars due to their low resolution.

5) Ultrasonics: Ultrasonic sensors transmit high-frequency sound waves to estimate the distance to objects. Therefore, they are also active sensors and usually applied to close-range object detection at low speeds such as automated parking. These sensors are affected by temperature or humidity.

C. Datasets Used for 3D Object Detection

Data is an important part of a machine learning system. However, this importance is increased even more when dealing with deep neural networks. Therefore, in the followings, we describe few popular large-scale datasets currently available for 3D tasks with sufficient detailed information to allow others to replicate and build models based on these datasets. Table I shows a comparisons of popular datasets.

- **KITTI** [19][1]: KITTI is one of the well-known benchmark dataset collected for the autonomous driving platform, and its sensor configuration includes a wide-angle camera and a Velodyne HDL-64E LiDAR. The dataset contains both 2D and 3D annotations of cars, pedestrians, and cyclists in urban driving scenarios, which are suitable for several vision tasks; stereo, optical flow, visual odometry, 3D object detection, and 3D tracking. It provides 7,481 images for training and 7,518 images for testing, including camera calibration files. Detection is evaluated as: *easy*, *moderate* and *hard*, according to the occlusion and truncation levels of objects.
- **NYU-Depth V2** [20][2]: The dataset consists of 1449 densely labeled pairs of aligned RGB and depth images captured from Microsoft Kinect. The dataset represents 464 different scenes with 26 scene types. Gupta *et al.* [39] amalgamated the per-pixel dense labeling into 40 indoor object classes, where 795 images in training set and 654 images in the testing set. Later, Deng *et al.* [40] provide some improved annotation of this dataset for 3D object detection task.
- **SUN3D** [21][3]: The SUN3D dataset contains a large-scale RGB-D video on indoor scenes with camera pose, object labels, and point clouds registered into a global coordinate frame, which are reconstructed from RGB-D structure from motion (SfM). The dataset provides 8 annotated sequences, and a total of 415 sequences are captured for 254 different spaces in 41 different buildings.
- **Berkeley B3DO** [16][4]: B3DO dataset is a category-level 3D object dataset that exhibits significant variations in terms of objects and poses, containing 849 images taken in 75 different scenes with 50 different object classes. The dataset supplies basic object annotations in the form of 2D bounding boxes. Later, Barnea *et al.* [41] extended the dataset by adding annotations of the 3D center points of all objects.
- **SUN RGB-D** [25][5]: Song *et al.* [25] introduces SUN RGB-D dataset through coalesced RGB-D images from NYU depth v2 [20], Berkeley B3DO [16], and SUN3D [21], aimed to build a large-scale RGB-D dataset at a similar scale as PASCAL VOC [42]. It comprises 10,335 RGB-D images of indoor scenes with densely annotated in 2D and 3D, including 146,617 2D polygons and 58,657 3D bounding boxes with accurate object orientations, as well as a 3D room layout and category for scenes. The whole dataset captured from four different kinds of RGB-D sensors that contains 47 scene categories, 800 object categories, and roughly 14.2 objects annotated in each image. Therefore, the data is carefully divided into training (5,285 images) and testing (5,050 images) sets such that each split can allocate half data from each sensor. SUN RGB-D dataset have enabled several vision tasks, including 3D object detection, scene classification, semantic segmentation, object orientation, room layout estimation, and indoor scene understanding.
- **KAIST-CU** [38][6]: The KAIST-CU is a complex urban LiDAR dataset collected from metropolitan areas, large building complexes, and underground parking lots, which consists of multi-sensor data including RGB/thermal camera, RGB stereo, 3D lidar and GPS/IMU. The LiDAR data are provided in two-dimensional and three-dimensional forms. The dataset also provides nighttime data.
- **H3D** [37][7]: The Honda Research Institute 3D Dataset (H3D) is a large-scale 3D multi-object detection and tracking dataset for traffic scene understanding. It is collected from HDD [43], which is driving scene understanding dataset captures in San Francisco bay area. The data are taken from five sensors: 3 cameras, LiDAR and GPS/IMU. The H3D dataset contains full 360-degree view angle of LiDAR data with 160 crowded traffic scenes and annotated with 1,071,302 3D bounding box labels, which are categorized into 8 common classes.
- **nuScenes** [36][8]: The nuScenes is the first dataset that carries full autonomy vehicle sensor kit: 6 cameras, 5 radars and 1 lidar, all with a full viewing angle of 360 degrees. It consists of 1000 scenes with each 20s long and fully annotated with 3D bounding boxes that categorized into 23 classes and 8 attributes. As a whole,

[1]http://www.cvlibs.net/datasets/kitti
[2]https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html
[3]http://sun3d.cs.princeton.edu
[4]http://kinectdata.com
[5]http://rgbd.cs.princeton.edu
[6]http://irap.kaist.ac.kr/dataset
[7]http://usa.honda-ri.com/H3D
[8]https://www.nuscenes.org

TABLE I
COMPARISON OF VARIOUS PUBLICLY AVAILABLE 2.5D/3D DATASETS.

Dataset	Year	Scene type	Sources	Object classes	Annotation Image/Scenes			Synthetic/ Real
					Training	Validation	Test	
Berkeley B3DO [16]	2011	Indoor	RGB-D	50	849	N/A	N/A	R
Washington RGB-D [17]	2011	Indoor	RGB-D	51	250000	N/A	N/A	R
Cornell RGB-D [18]	2011	Indoor	RGB-D	N/A	52 (point cloud)	N/A	N/A	R
KITTI [19]	2012	Urban (Driving)	RGB & LiDAR	8	7481	N/A	7518	R
NYU-Depth V2 [20]	2012	Indoor	RGB-D	40	795	N/A	654	R
SUN3D [21]	2013	Indoor	RGB-D	-	19640	N/A	N/A	R
IKEA [22]	2013	Indoor	RGB & 3D models	90	800	N/A	N/A	R
Sydney-Urban Objects [23]	2013	Urban	LiDAR	26	41	N/A	N/A	R
PASCAL3D+ [24]	2014	Indoor & Outdoor	RGB & 3D models	12	24K	N/A	N/A	R
SUN RGB-D [25]	2015	Indoor	RGB-D	37	2666	2619	5050	R
CityScapes [26]	2015	Urban (Driving)	RGB	8 groups	25000	N/A	N/A	R
ObjectNet3D [27]	2016	Indoor & Outdoor	RGB & 3D models	100	90127	N/A	N/A	S
ShapeNet Part [28]	2016	Object/Part	3D models	16	31963	N/A	N/A	S
MVTec ITODD [29]	2017	Industrial	Grayscale & 3D models	28	800	N/A	N/A	S
T-LESS [30]	2017	Industrial	RGB-D & 3D models	30	39K	N/A	10K	S
ScanNet [31]	2017	Indoor	RGB-D	50	1205	N/A	312	S
2D-3D-S [32]	2017	Indoor	RGB-D	13	70469	N/A	N/A	R
SUNCG [33]	2017	Indoor	3D scenes	84	130K	N/A	N/A	S
FAT [34]	2018	Indoor	RGB & 3D models	21	61500	N/A	N/A	S
House3D [35]	2018	Virtual Indoor	Designed 3D scenes	80	45622	N/A	N/A	S
nuScenes [36]	2019	Urban (Driving)	Camera & LiDAR	23	1.4M	N/A	N/A	R
H3D [37]	2019	Urban (Driving)	Camera & LiDAR	8	50	30	80	R
KAIST-CU [38]	2019	Urban (Driving)	Camera & LiDAR	-	8.9K	N/A	N/A	R

the dataset provides approximately 1.4M camera images, 390K LiDAR scans, 1.4M RADAR scans and 1.4M object bounding boxes, which is 7× more annotations and 100× more images than popular KITTI dataset [19].

- **MVTec ITODD** [29][9]: MVTec ITODD is a 3D object detection and pose estimation dataset, which strongly focuses on industrial scenarios. The dataset contains 28 rigid objects with different shapes and surface characteristics, arranged in over 800 scenes and labeled with around 3500 rigid 3D transformations as ground truth. The scenes are observed by two industrial 3D sensors and three grayscale cameras from different angles, allowing to evaluate methods that operate on 3D, image, or combined modalities.
- **FAT** [34][10]: Tremblay *et al.* [34] present Falling Things (FAT), a synthetic dataset for 3D object detection and pose estimation generated by placing 3D household object models in virtual environments. The dataset consists of 61,500 annotated images of 21 household objects collected from the YCB dataset [44]. It supplies the annotations of 3D poses, per-pixel class segmentation, 2D/3D bounding box coordinates, and projected 3D bounding boxes for all objects. Additionally, it provides mono and stereo RGB images, along with registered dense depth images to facilitate testing on different input modalities. FAT dataset is suitable to accelerate research in 3D object detection and pose estimation, as well as segmentation, depth estimation, and sensor modalities.

III. 3D OBJECT DETECTION METHODS

A 3D object detection system takes different types of data as input and outputs 3D bounding boxes and class labels for all objects of interest in the sensor field of view. The input data potentially come from a combination of various sensors, such as monocular or stereo cameras, RGB-D sensors, LiDAR, and sonar. Most existing works either use only RGB images from visual cameras, or just use 3D point clouds from LiDAR, or combine RGB images with 3D LiDAR point clouds, or employ Kinect RGB-D camera to fuse RGB images and depth images. Based on input data representation, 3D object detection methods can be divided into three main categories: image-based methods, point cloud-based processing methods, and multimodal fusion-based methods. We summarize the comparison of these methods and their limitations in Table II.

A. Image-based Processing Methods

Image-based processing methods use only monocular images as input. Although these methods have achieved enormous success in 2D object detection, obtaining 3D bounding boxes only from the 2D image plane is a much more challenging task due to the absence of absolute depth information. Since there is no depth information available, most methods use two-stage approach for 3d object detection: first generate proposals and then perform detection. 2D proposal boxes extracted from hand-crafted features [45]–[48] or standard 2D object detectors using deep neural networks [1], which are then regress to oriented 3D bounding boxes using geometrical constraints [49], 3D models fitting [11], [50]–[54] or active shape modeling [55], [56].

One example of early works related to this category is 3DOP [57] proposed by Chen *et al.*. 3DOP [57] utilizes stereo imagery to reconstructs depth and exploits the Markov Random Field (MRF) ranking model to generate object proposals in the form of 3D bounding boxes. These 3D proposals are then projected onto the image as 2D bounding boxes and

[9]https://www.mvtec.com/company/research/datasets/mvtec-itodd
[10]https://research.nvidia.com/publication/2018-06_Falling-Things

TABLE II
COMPARISONS OF DIFFERENT 3D OBJECT DETECTION METHODS BASED ON INPUT MODALITY

Category		Methodology	Limitations	Research Gaps
Image-based		Use only RGB images as input, then generate 2D region proposals using hand-crafted features or standard 2D object detectors, and finally infer the 2D bounding boxes to 3D bounding boxes by re-projection or bounding box regression. Image-based methods are less expensive and more flexible.	The lack of depth information, which is critical to accurate object size and location estimation.	Methods for estimating depth information need to be investigated to improve the accuracy of object size and position.
Point cloud-based	View-based	Converts the point cloud into a 2D image views and processed using conventional CNN architecture to obtain 2D bounding boxes, which are then used to retrieve 3D bounding boxes by position and size regression.	Loss of information while projecting the 3D point cloud onto a 2D image plane.	Input images need to be encoded with the hand-engineered feature, which could be replaced by the learned representation.
	Voxel-grid based	Point cloud is discretized into 3D voxel representation and 3D convolutional network or Fully Convolutional Networks (FCNs) uses to predict detection. Shape information is preserved by the volumetric representation.	Volumetric representations are sparse containing many empty cells, which reduces efficiency of processing these empty cells. Operating 3D convolutions result in higher computational costs.	Due to its computational complexity, it is necessary to explore how to effectively use 3D CNN. Volumetric representation could be used to generate 3D region proposals.
	Unstructured point cloud-based	CNNs directly takes unordered raw point clouds without voxelization or rendering to learn pointwise features for classification and bounding box estimation.	The computational complexity is increased when considering the entire point cloud to be processed. Often required region proposals to limits the number of points.	More investigation is needed to operate with the entire point cloud. In addition, there has not been much research on the geometric relationships among the points, which can improve accuracy.
Multimodal fusion-based		Fuse point cloud and camera images by using muti-modal CNN architecture to generate detections. The fusion-based approach shows state-of-the-art detection performance because they can aggregate different features form different sensor modalities.	It is necessary to align features from different sensors. Difficult to represent and process different sensor modalities appropriately before feeding them into a fusion network.	More research is needed to investigate how to fuse different sensing modalities and align them temporally and spatially.

given to an extended Fast R-CNN [10] pipeline to jointly predict the class of the object proposal and estimate object orientation by using angle regression. 3DOP formulates the 3D as an energy minimization approach by encoding object size priors, ground plane and a variety of depth informed features such as free space, point densities inside the box, visibility and distance to the ground. Although the authors show a great performance of 3DOP for 2D detection and orientation estimation, they do not provide a quantitative evaluation of the 3D bounding box proposals. Chen *et al.* extended 3DOP in Mono3D [58] to obtain a monocular version. Unlike 3DOP, Mono3D generates 3D proposals from monocular images and use the assumption that all objects should lie close to the ground plane, which should be orthogonal to the image plane. Then each 3D proposal box is scored in the image plane via several intuitive potentials by utilizing semantic segmentation, contextual information, size and location priors, and typical object shape. Finally, the top-scoring regions are fed to the extended Fast R-CNN pipeline used in 3DOP to predict category labels and estimate 3D bounding box offsets and orientation. However, the limitation of these models is that it needs to be run separately for each object category and requires many proposals in order to achieve high recall, which leads to an increased processing time of the classification. To overcome this limitation, Pham *et al.* proposed DeepStereoOP [59], a class-independent object proposal re-ranking approach that uses both monocular images and depth maps in a lightweight two-stream CNN architecture. Their approach outperforms both 3DOP and Mono3D methods.

Xiang *et al.* introduces 3D Voxel Patterns (3DVP) [11] representation that jointly encodes the key object properties by the appearance through RGB luminance values, 3D shape as a set of 3D voxels, and occlusion masks and learn a large dictionary of 3DVP. Their approach follows the idea that by learning such a large 3DVP dictionary, it is possible to effectively simulate changes in the brightness of objects in an image due to intra-class variations and occlusions, where each 3DVP captures the specificity of the three attributes listed above (appearance, 3D shape and occlusion). Using this 3DVP dictionary, they train a set of object detectors through which each detector is trained from the appearance information associated with a specific 3DVP. These detectors allow to

localize objects in the image even when viewed from any viewpoint or visible under severe occlusion. In the testing phase, 3DVP detector can predict 2D segmentation mask, 3D pose or 3D shape and can be used with 3D CAD models to detect 3D objects.

In CNN-based object detection, Region Proposal Network (RPN) [10] performs better than the traditional region proposal methods. However, the bottleneck is that it cannot handle scale variation of object, occlusion and truncation. By observing this limitation, Xiang *et al.* extended 3DVP in SubCNN [51] where they use the concept of subcategory information for region proposal generation and object detection. A subcategory can be objects with similar properties or attributes, such as 2D appearance, 3D pose or 3D shape. The RPN is designed by using a *subcategory convolutional layer* that outputs heat maps for the existence of certain subcategories at a particular location and scale. The detection is performed by injecting subcategory information into the Fast R-CNN. By using 3DVPs as subcategories, the method can jointly detect objects, 3D shape, pose, and occluded or truncated regions. In addition, a *feature extrapolating layer* is used in both RPN and detection network, which takes image pyramids as input and efficiently computes *conv* features at multiple scales to detect objects with large-scale variations. Although the 3D voxel patterns representation is robust to object occlusion and truncation, it still relies on a class-wise dictionary learning process, which may fail for any object pose different from the existing patterns.

Another attempt by Chabot *et al.* in Deep MANTA [50], where 3D object parts (vehicle parts) are localized even if these parts are hidden due to occlusion, truncation or self-occlusion in the image. Deep MANTA uses a many-task CNN to regrate jointly 2D and 3D bounding boxes of vehicles with multiple refinement steps. First, the input monocular image is passed through a many-task CNN network that outputs 2D scored bounding boxes, vehicle part coordinates, 3D template similarity, and part visibility properties. These outputs are then fed to a second step inference in which two 3D vehicle datasets (3D shape and 3D template) are utilized to recover the 3D orientations and locations of the vehicle. These datasets consist of various 3D models of different types of vehicles together with their corresponding 3D bounding box dimensions and 3D parts coordinates. In the inference step, the 3D bounding box dimensions are first compared to the entries in the 3D template dataset to find the best-matching 3D template of the vehicle, then 2D/3D shape matching algorithm [60] is applied to estimate full 3D bounding box and 3D part coordinates. The authors report improved performance of the 3D localization accuracy compared to the 3DOP [57]. However, the limitation is that Deep MANTA training requires a large database of 3D models (geometry information, visibility, etc.) consisting of different types of vehicles, which makes the architecture difficult to generalize to the category where such models do not exist.

Recently, deep neural networks combining with geometric properties has shown more accurate results. Mousavian *et al.* first presented a much simplified monocular image only architecture in Deep3DBox [49] that combines visual appearance and geometric constraints into 3D object detection scenario. Deep3DBox utilizes the state-of-the-art 2D object detector to estimate 3D bounding boxes with the geometric constraints that the 3D bounding box fits tightly into 2D detection window requires that each side of the 2D bounding box to be touched by the projection of at least one of the 3D box corners. First, a 2D object detector [61] is extended by training a discrete-continuous CNN architecture to regress the orientation of the object's 3D bounding box and dimensions. In contrast to only regressing 3D orientation of an object, Deep3DBox uses a MultiBin regression for the estimation of the heading angle of the object. Finally, given the estimated object's heading angle, dimensions, and the aforementioned constraints, the center coordinates as well as complete 3D bounding box can be calculated using an optimization-based method. Deep3DBox has shown improved detection and orientation estimation performance compared to the more complicated architecture of 3DOP [57].

Following MultiBin architecture of [49], Xu *et al.* [62] present an end-to-end multi-level fusion-based framework from a single monocular image for 2D/3D object detection with a stand-alone fully convolutional network (FCN) based module to predict disparity information and compute 3D point cloud. The disparity information is then encoded with a front view feature representation and fused to the original RGB image to enhance the input, which is then fed into a Faster R-CNN based region proposal network to generate 2D region proposals. Based on the 2D region proposals, an RoI max-pooling layer is applied to the main convolutional branch and an RoI mean-pooling layer is introduced to convert the point cloud inside the proposal into a fixed-length feature vector in another stream. Then different levels of fusion are used to compute object classification, orientation, dimension, and location.

The main limitation of [57] and [59] is that each region proposal or bounding box is treated independently, preventing any joint reasoning about the 3D configuration of the scene. To overcome this, Roddick *et al.* introduce an orthographic feature transform (OFT) based architecture in OFT-Net [63] by following a similar feature aggregation techniques in Mono3D [58], but they apply a secondary convolutional network to the resulting proposals while retaining their spatial configuration. OFT-Net maps RGB image features into a top view representation, which are further processed by a secondary top-down convolutional network to predict confidence score, position offset, dimension offset and angle vector.

Currently, expensive LiDAR-based 3D object detection techniques perform highly accurate, while the cheaper image-based 3D object detection performs drastically lower accuracies. Wang et al [64] find the gap between these two techniques by arguing that a key component to closing the gap may be simply the representation of the 3D information. By observing this gap, they propose to convert image-based depth maps to pseudo-LiDAR representation since it mimics the LiDAR signal. Finally, they exploit existing LiDAR-based 3D object detection pipelines [9], [65] to detect the 3D object. By converting 3D depth representation to pseudo-LiDAR, they obtain an unprecedented performance gain in accuracy of the

image-based 3D object detection techniques.

Recently, 3D object detection from the 2D point of view has gained attention to the researcher. [66] proposes a 2D driven 3D object detection method to reduce the search space for objects in 3D, which uses a multistage pipeline to perform 2D object detection, 3D object orientation regression and object refinement based on context information. However, the limitation is that they use hand-crafted features (*i.e.,* histograms for the coordinates of the 3D points) to train a multilayer perceptron (MLP) network to regress the 3D bounding boxes. Similarly, [40] also propose a 3D object detection technique by inferring 3D bounding boxes from 2D. A limitation of their work is that it is not fully integrated architecture as there are two separate computations: first, they externally compute 2D bounding box proposals along with their segmentation mask by using extended multiscale combinatorial grouping (MCG) algorithm [13], then they use those 2D proposals and corresponding segmentation mask information to compute 3D boxes for the classification and 3D bounding box regression. Nevertheless, computing 2D proposals and segmentation mask from the offline algorithm (*i.e.,* MCG [13]) requires quite high computation cost and are not feasible to the automated system. In addition, they require to provide object segmentation information in both training and testing phases to predict 3D bounding boxes. Later, Rahman *et al.* extended the [40] in an integrated manner in [67], by proposing a multi-modal region proposal network to generate region proposals and a dilated 2D bounding box method to produce 3D bounding boxes.

2D image-based methods have been extensively studied because 2D cameras are less expensive and more flexible than complex 3D acquisition sensors. 2D images provide rich color and texture information of the object in the form of pixel intensity. However, the disadvantage of a 2D image is that it lacks depth information, which is necessary for accurate object size and location estimation especially in low light conditions, and the detection of distant and occluded objects. Despite this limitation, 3D object detection with 2D image-based methods becomes important for economic 3D object detection systems.

B. Point Cloud-based Processing Methods

Since 2D image-based methods do not provide depth information that is crucial for location aware applications such as robot navigation, autonomous driving and augmented/virtual reality, point cloud-based methods can provide solutions that can significantly improve 3D detection performance than 2D image-based approaches. Point clouds provide reliable depth information, which can be used to precisely localize objects and determine their shapes. In recent years, it is evident that there is a growing interest in using point clouds with the deep learning based 3D object detection. Unlike images, point clouds are unordered, unstructured, sparse, and unevenly distributed in space. Besides, existing deep learning algorithms are designed for structured regular input data such as images. In order to use point cloud for existing deep learning algorithms, there are mainly three ways for representing the point clouds. One way is to represent 3D point cloud by projecting them onto a 2D perspective view so that they can be processed by 2D convolutional layers [68]; the second way is to discretize point clouds into 3D voxel grids [69], and the third way is to directly learn over raw point clouds [12], [70]. Considering the point cloud representation, current point cloud-based 3D object detection methods can be divided into three sub-categories: view-based, voxel-grid based and unstructured point cloud-based methods.

*1) **View-based Methods**:* The view-based method converts the 3D point cloud into a projected 2D image views [71], spherical views [72], cylindrical views [68], [73] or top-views (bird's eye views), which are processed using conventional deep neural architecture to obtain 2D bounding boxes. These 2D bounding boxes are then regressed by position and dimension regression to retrieve the 3D bounding boxes.

One such technique is VeloFCN [68], where Li *et al.* project the 3D point cloud onto a cylindrical image to obtain a 2D depth map which is used as an input to a Fully Convolutional Network (FCN) that outputs objectness classification and bounding box regression. Similarly, Minemura *et al.* also adopt cylindrical projection of 3D point clouds in LMNet [73]. They project 3D point cloud onto five frontal-view representation namely: reflection, range, forward, side, and height and then use a dilated convolutional network to address real-time like 3D multi-class object detection. Another example is presented in [72], where irregularly distributed 3D point clouds are transformed to a spherical map, which is characterized by azimuth and zenith angles. The advantage of spherical projection is that it represents 3D point in a dense and 2D grid representation that is suitable for point cloud segmentation.

Compared to the aforementioned methods in which point cloud is encoded as a front-view representation, a Bird's Eye View (BEV) representation avoids occlusion problems since objects occupy different space on the map. Moreover, the BEV can retain the length and width of the object, and provide the position of the object directly on the ground plane, which makes the localization task simpler. Recently, BEV representation has been extensively used in 3D object detection [74]–[76]. Methods, such as DoBEM [74], BirdNet [75] transformed the point cloud into three channels grid cell representation of the bird's eye view projection. DoBEM [74] represent the bird's eye view as elevation images, where each pixel of the bird's eye view encodes into three channels, namely maximum height, median and minimum height. BirdNet [75] encodes the BEV image as three channels with height, intensity, and density information. This three channels representation of the bird's eye view allows them to utilize common image-based object detection network without modification. TopNet [76] encodes the top-view as multi-layer grid maps containing different features, such as height, intensity, detections, observations and decay rate per cell. Du *et al.* [77] present a flexible 3D detection pipeline to adopt any 2D detection network and fuse it with a 3D point cloud viewed from the bird's eye view to provide accurate 3D detection results. An effective model fitting algorithm is used to detect the 3D location and 3D bounding boxes. All these methods use the Faster R-CNN [1] style architecture that generates region proposals with an adjusted refinement network to predicts oriented 3D bounding boxes.

In contrast to a typical two-stage detector [1], PIXOR [78], FaF [79], Complex-YOLO [80], YOLO3D [81] and HDNET [82] show excellent performances in both speed and accuracy by exploiting single-stage detector on bird's eye view representation. Unlike two-stage detectors that require region proposal generation and require further processing of each region for finer predictions, this type of architecture aims to map the feature maps directly to classification scores and bounding boxes through a single-stage, unified CNN model. PIXOR [78] conducts a single-stage, proposal free, real-time 3D object detection that efficiently makes use of height-encoded BEV input by assuming that the objects are on the ground and outputs pixel-wise prediction corresponds to a 3D object estimate. Similarly, FaF [79] also takes advantages from BEV representation to conduct a single-stage detector that consumes a 4D tensor created from multiple consecutive temporal frames and perform 3D convolution. FaF [79] pioneered for jointly learn 3D detection, tracking and short-term motion forecasting from LiDAR point clouds in driving scenarios. Recently, Complex-YOLO [80] and YOLO3D [81] extended YOLOv2 [83] to perform 3D object detection and classification from the bird's eye view representation projected from the 3D LiDAR point cloud to achieve real-time performance. Complex-YOLO [80] uses a simplified YOLOv2 [83] architecture by expanding it to a specific complex regression strategy to increase speed and performance. In order to this, Complex-YOLO [80] utilizes a specific Euler-Region-Proposal Network (E-RPN) for reliable angle regression to detect accurate multi-class oriented 3D objects. However, it uses fixed height and z-center locations in the predicted 3D bounding boxes. In addition, as Complex-YOLO translates the orientation vector to real and imaginary values, angle regression does not guarantee or preserve any type of correlation between the two components. By observing this, YOLO3D [81] extended the loss function of YOLOv2 [83] to include yaw angle, the 3D box center in cartesian coordinates and the height of the box as a direct regression task. Similarly, M-YOLO [84] also expand the YOLO to improve the positioning accuracy of the 3D object detection. Similarly, HDNET [82] exploits high High-Definition (HD) maps information to boost the 3D detection performance in the context of autonomous driving. HDNET is a single-stage detector that operates on bird's eye view LiDAR representation in combination with the HD maps which contain geometric and semantic information about the environment. Although bird's eye view shows reasonable results, these methods often experiencing poor orientation angle regression.

*2) **Voxel-grid Based Methods**:* The voxel-grid based methods first discretized the 3D point cloud into a volumetric 3D grid, or voxel representation, where each entity attributed as binary occupancy or a continuous point density. The advantage is that this volumetric representation preserves the shape information and can be applied directly to the 3D convolutional networks. However, due to the sparsity and irregularity of the 3D point cloud, this method results in many empty voxels, which reduces efficiency when processing these empty cells. Moreover, since 3D convolutional networks result in high computational cost, this type of representation is constrained by its resolution.

Inspired by the success of 2D fully convolutional network (FCN) [85], [86], Li *et al.* extended their previous work [68] in [87], where 3D FCN based mechanism is used instead of 2D CNN by utilizing point cloud data discretized into a 3D grid. For input, they used binary encoding to discretize the point cloud into a voxelized 4D array with dimension of length, width, height and channels, and predict the objectness and object shape in 3D space.

The major difficulties in an object detection algorithm occur due to variation in object texture, illumination, shape, viewpoint, self-occlusion, clutter, and occlusion. To overcome this difficulty, Sliding Shapes [88] exploited depth information in a data-driven fashion and trained exemplar SVM [89] on 3D voxel grids encoded with hand-crafted geometric features by rendering a collection of 3D CAD models from different viewpoints into synthetic depth maps. At the testing time and hard-negative mining, a 3D detection window is slid on the 3D space to match the exemplar shape and each window by the learned SVMs.

Vote3D [90] adopt the same strategy of [88] to train the SVM classifier on 3D voxel and used a feature-centric voting scheme by applying sliding window approach, which made the exhaustive sliding window searching in 3D extremely efficient. The author first discretized the 3D space into a fixed resolution grid cell. Then, the points in the point cloud occupied into each cell are converted into a fixed-dimensional feature vector and the cells that are not occupied by any points map to zero feature vectors. Finally, a 3D detection sliding window exhaustively searches for an object with a linear Support Vector Machine (SVM). Vote3D [90] proves the voting scheme is mathematically equivalent to a dense convolution operation.

To obtain a more efficient implementation, another approach was later presented by Engelcke *et al.* in Vote3Deep [91] that extends the Vote3D [90] by replacing SVM with a 3D convolution networks on voxelized 3D grid cells. Vote3Deep fixes the bounding box sizes for each class and uses sparse convolutional layers based on voting and L1 regularization. However, it is challenging to handle very large volume of point clouds.

Inspired by Sliding Shapes [88], Bimodal DBM [92] also adopt 3D sliding window strategy in the 3D point cloud cells to get scores for all exemplar-SVMs. The 3D bounding boxes were projected into 2D bounding boxes on both channels and then the cross-modal features were extracted by feeding it into pre-trained R-CNNs and bimodal deep Boltzmann Machine, respectively. Finally, the exemplar-SVMs are used for detection. However, the uses of many exemplar classifier and hand-crafted features make the algorithm computationally intensive.

Ren *et al.* in [93] design a cloud of oriented gradients (COG) descriptor for 3D object detection and propose a "Manhattan voxel" representation in a point cloud to capture 3D room layout geometry for indoor environment. Later, [94] extended in [93] by using latent support surfaces to improve the detection of small objects. Although [93] and [94] archived significant gain on mAP, their model takes 10-30min processing time for each image while testing.

Since the orientation angle will directly affect the 3D

detection performance, the estimation of the object orientation is of great significance. Sedaghat *et al.* [95] proposed an orientation-boosted voxel-based 3D CNN that outputs orientation labels as well as classification labels with improved 3D object classification performance.

Zhou *et al.* proposed much improved LiDAR-only architecture named VoxelNet [69], which takes only raw point cloud as input without the need of manual feature engineering and simultaneously learns a discriminative feature representation to predicts accurate 3D bounding boxes using a single end-to-end trainable network. VoxelNet [69] has three functional blocks: (1) Feature learning network, (2) Convolutional middle layers, and (3) Region proposal network. Feature learning network separates the point cloud into equally spaced 3D voxels, transforms points within each voxel to a vector represented as 4D tensor characterizing the shape information through newly designed voxel feature encoding (VFE) layers, which is actually a small PointNet [12]. After that, the convolutional middle layers aggregate the neighborhood voxel features, adding more context to the shape description, and converting point cloud into a high-dimensional volumetric portrayal. Finally, the region proposal network takes the volumetric representation and obtains the 3D detection results. While the VoxelNet [69] performance is strong by utilizing the sparse 3D convolutional operation, the major bottleneck with these methods is the high computational cost of the 3D convolutional network as the computational complexity of 3D CNN increases cubically with the voxel resolution.

Recently, SECOND [96] offers a series of improvement to VoxelNet [69] to increase training and inference speed. SECOND [96] uses several improved sparse convolutional layers after the VFE in order to convert the sparse voxel data into 2D images and applies a new form of angle loss regression to improve the orientation estimation. These sparse convolution only operates on the locations associated with input points, resulting in much stronger 3D detection performance with improved speed. However, SECOND still remains the bottleneck of using the expensive 3D convolutions.

*3) **Unstructured Point Cloud-based Methods**:* A point cloud is an unordered set of vectors. Since typical CNNs require highly regular input data formats, most researchers transform the point cloud to regular 3D voxel grid or project them into perspective images due to the irregular nature of the point cloud. However, in this pre-processing, spatial information is always lost to some extent. On the other hand, 3D voxel grids require highly computationally intensive 3D convolution operations, which limits their application to practical applications. Motivated by these issues, some works [12], [70] have recently developed a learning-based architecture for dealing with unordered raw point clouds.

Qi *et al.* [12] proposed PointNet, a unified end-to-end deep neural network architecture that operates directly on unordered point clouds without lossy operations like voxelization or rendering and learns both global and local point-wise features. The point-wise transformation is performed by using Fully Connected layers and a max-pooling layer is used to aggregate the global features. The key idea of PointNet is that the classification network takes n points as input, applies point-wise feature transformations, and then use max pooling layer as a symmetric function to aggregate features from all points. These features are then used for 3D object recognition, 3D object part segmentation, and point-wise semantic segmentation tasks. Although PointNet has the strong ability to capture global structures, it cannot capture local structures induced by the metric space. Local structures are very important to the success of convolutional neural architecture for fine-grained classification and generalizability to complex scenes. To overcome this bottleneck, PointNet is further extended in PointNet++ [70] which enabled the network to learn the local structures by increasing contextual scales in distance metric space. PointNet++ first partition the set of points into overlapping local regions using farthest point sampling (FPS) algorithm, then capture the local feature using PointNet, and finally calculate higher features by grouping the local features into bigger unit.

The limitation of PointNet and PointNet++ is that they ignor the geometric relationships among points. To address these drawback, Wang et al. simplify the PointNet in EdgeConv [97] by generating edge feature that define the geometric point-wise relationships, rather than generating point features directly from point-wise transformation. However, the original PointNet architecture was firtly designed for 3D object recognition tasks and later extended in Frustum PointNet [65], PointFusion [98] and RoarNet [99] to 3D object detection tasks in combination with RGB images, which are described into next section.

C. Multimodal Fusion-based Methods

Image-based processing methods only provide texture information, not depth information. On the other hand, point cloud-based processing methods provide depth information but lack texture information. Texture information is important for object detection and classification, while depth information is critical for accurate estimation of 3D location and object size. Furthermore, as the distance from the sensor increases, the point cloud density decreases proportionally. Since texture and depth modalities are essential in 3D object detection, some methods use both images and point clouds to improve the overall performance by adopting a fusion schemes. Usually three fusion schemes are used [8]: early, late, and deep fusion, as shown in Fig. 3. Early fusion fuses raw or pre-processed sensor data, late fusion merges the last layer of feature maps, and deep fusion mixtures different modalities hierarchically. Most works found in the literature propose to fuse 3D point cloud and RGB image features extracted from 2D CNNs. For this purpose, they project point cloud onto 2D plane and use 2D convolutions to process it. Some works directly fuse bird's eye view features with RGB images [8], [9] and many works project 3D point clouds on the image plane or RGB images on the bird's eye view plane [100] in order to align the features from multiple sensors. Some works extract point cloud features by using PointNet [65], [98] or 3D convolutions [7], while the others generate 3D region proposals from the segment of 3D point cloud [101], [102] and make fusion with the 2D representation of the point cloud.

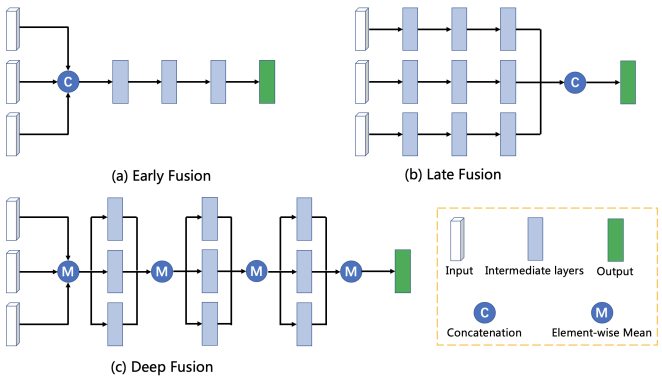

Fig. 3. An illustration of early fusion, late fusion, and deep fusion [8].

MV3D [8] takes the bird's eye view and front-view projection of 3D point cloud together with the RGB image as input, from which feature maps are extracted using multi stream CNNs. The LiDAR bird's view feature map is passed to an RPN, which is extended from Faster R-CNN [1], to generate highly accurate 3D region proposals. Each of these 3D region proposals is projected onto all three views of the feature maps. Finally, three types of fusion network is used to combine region-wise features from multiple views for object classification and oriented 3D box regression. The author conclude that the deep fusion approach provides the best performance since it aggregates the feature hierarchically.

In a similar way, AVOD [9] introduce an early fusion architecture that takes bird's eye view projected from LiDAR point cloud and RGB images to generate high-resolution feature maps which are shared by two subnetworks: a region proposal network and a second stage detection network. The architecture is thus similar to MV3D [8], the key difference is that AVOD [9] feeds the RPN with the three sources of information: color images, 3D voxel grid, and birds eye-view to perform multi-modal feature fusion to generate reliable 3D object proposals. These 3D proposals are then transferred to the second stage detector network for oriented 3D bounding box regression and classification. Unlike 8 corner box encoding [8] or axis aligned encoding [7] of a 3D bounding box, AVOD [9] encodes the 3D box by using top and bottom corner offsets from the ground plane. Besides, it reports two limitations to compute orientation from a 3D bounding box introduced in MV3D [8], where the orientation vector is assumed to be in the direction of the longer side of the box. First, this method fails in regards to pedestrians because of not obeying the rule. Secondly, the orientation information is lost as the corner order is not preserve in closest corner to corner matching. AVOD [9] resolve this ambiguity by converting the orientation vector to sine and cosine with four corner representation. The author also report that utilizing both color image and BEV features in the RPN, as compared to only using BEV features in [8], is not effective on the performance of the cars category, but a significant effect on pedestrians and cyclists category.

Information fusion between LiDAR data and image data is non-trivial because images represent the camera projection of the 3D world, while LiDAR captures the geometric structure of the native 3D world. Traditional solution is to project the LiDAR data onto the image which then append as an extra channel with depth information and perform 2D detection methods on that data. In contrast to this method, ContFuse [100] perform the opposite operation that exploits image features extracted by a convolution network and then project the image features into a bird's eye view. ContFuse takes advantages from continuous convolutions [103] to extract information from the nearest corresponding image features for each point in BEV space and fuse image and LiDAR feature maps at different levels of resolution for 3D object detection.

Inspired by the [104], Song *et al.* extended [88] in Deep Sliding Shapes [7], where a multi-modal deep learning framework is introduced for amodal 3D object detection. Unlike typical object detection algorithm which draws bounding box on object visible parts only on an image plane, amodal object detection aims to draw bounding box of an object in full extent even if part of the object is occluded or truncated. Following faster R-CNN [104], Deep Sliding Shapes [7] presented a 3D ConvNet based approach that converts a point cloud of an entire scene generated from depth map into a 3D volumetric grid and produces object category labels along with 3D bounding boxes. The approach has two modules. First, a 3D region proposal network (3D RPN) which takes a 3D volume to generate 3D region proposals. Second, a joint object recognition networks (3D ORN), where each 3D region proposal is feed into another 3D convolutional network, and 2D projection of the 3D proposal is fed to a 2D convolutional network to jointly learn geometric features in 3D and color features in 2D. However, operating 3D ConvNets on a 3D voxel grids in the large 3D search space is computationally expensive. Moreover, another limitation of this work is that the computation of the two modules have done separately, where the convolutional layers are not shared, and the object orientation is not explicitly measured.

Recently, PointNet [12] based 3D object detection becomes popular. However, the original PointNet formulation cannot be used for instance-level 3D object detection task. With this observation, Qi *et al.* presented Frustum PointNet (F-PointNet) [65], which leverage the mature 2D object detectors [105], [106] and PointNet/PointNet++ [12], [70] architecture to detect 3D objects. F-PointNet takes RGB image and depth. It consists of three main modules: 3D frustum proposal, 3D instance segmentation, and 3D bounding box estimation. In the frustum proposal module, F-PointNet first uses a 2D CNN based object detector to generate 2D region proposals in order to reduce the search space in 3D. Each 2D region proposal is then mapped to corresponding 3D frustum proposal that contains all points in the point cloud which lie inside the 2D region when projected onto the image plane. After that, these 3D frustum proposal is fed to the instance segmentation module, in which PointNet [12], [70] (T-Net) architecture is used to perform binary classification for each point, predicting whether or not the point belongs to the detected object. Finally, all positively classified points are fed into the 3D bounding box estimation module, in which another PointNet is used to regress the 3D bounding box parameters. The

network uses a "residual" approach to regresses the box center estimation. For box dimensions and heading angle, F-PointNet follows the [49], [104] and utilize a hybrid of classification and regression formulations. Although F-PointNet [65] gains pioneering results on both KITTI and SUN RGB-D 3D object detection benchmarks compared to other methods, its multi-stage architecture makes the end-to-end learning impractical. F-PointNet [65] also suffers synchronization problem between sensors. Moreover, it suffers more time complexity due to fusing lidar data with the camera and may lose local orientation information while using K-nearest searching method in PointNet++. In real scenarios, objects have relatively different scales.

By observing this, Zhao *et al.* introduce the architecture of Scale Invariant and Feature Reweighting Network (SIFRNet) [107], which utilize front-view images and frustum point clouds to generate 3D detection results. SIFRNet has three components: 3D instance segmentation network (Point-UNet), T-Net, and 3D box estimation network (Point-SENet). Point-UNet is designed by integrating PointSIFT module [108] for 3D instance segmentation which has the ability to learn different orientation information while adapting to scale invariance to the shape of point clouds. The second component is T-Net, which is used to learn the global features with additionally taking Lidar reflection intensity feature. Finally, Point-SENet is designed by extending the SENet [109] to estimate final 3D bounding boxes.

Recently, Shin *et al.* proposed RoarNet [99] to improve 3D detection performance and reduce the problem caused by sensor synchronization issue from the 2D image and 3D LiDAR point clouds. RoarNet [99] consists of two parts. First, it takes a monocular image as input to estimate 3D poses and derives multiple geometrically feasible candidate locations. After obtaining 3D region proposals from 2D image, a two-stage 3D object detection framework, analogous to [1], [10] with PointNet [12] as a backbone network, is used to gradually refine the search space from 3D point clouds. RoarNet [99] differs from F-PointNet [65] by taking whole point clouds that are located inside region proposals instead of filtering out point clouds by using 2D bounding box, that makes the RoarNet [99] being more robust to sensor synchronization.

Another architecture presented by Shi *et al.* named PointRCNN [110], where a two-stage 3D object detection framework is used which take only raw point cloud as input. Differ from other works that utilized fused feature maps of bird's view and front view [9], or RGB images [65], PointRCNN [110] directly generate 3D proposals from raw point cloud in a bottom-up manner in the first stage. The second stage conducts canonical 3D bounding box refinement by combining semantic features and local spatial features.

Many other networks have been proposed advanced algorithms utilizing both images and raw point clouds in a sensor fusion manner to enhance the performance of the 3D object detection. Among these, Xu *et al.* present PointFusion [98], which leverages both image data and raw point cloud data independently processed by a CNNs [111] and a PointNet [12] architecture, respectively. After that, a fusion sub-network used to combine the output of the two networks to predict 3D bounding boxes.

Another recent work, PointPillars [112] also utilizes PointNets [12] to design an encoder that learns a representation of point clouds organized in pillars, where a pillar is a vertical column that can extend infinitely up and down. By learning end-to-end on these pillars enabling the PointPillars to be used with any standard 2D convolution layers for 3D object detection. PointPillars has three main stages: pillar feature network to convert a point cloud to a sparse pseudo-image, a 2D CNN backbone network to learn high-level representation from the pseudo-image, and a detection head using SSD [3] to detects and regresses 3D oriented boxes. PointPillars gains the faster speed by using pillars which eliminate the 3D convolutions replaces with 2D convolution.

Compared to image-only or point cloud-based methods, fusion-based methods show the state-of-the-art performance because they can aggregate different features from different sensor modalities. Point clouds provide geometric information, while images provide texture information that is essential for class identification. Combine these two modalities allows to use discriminative information to improve performance.

IV. Evaluation Metric

In object detection, *Intersection over Union* (IoU) is commonly used as an evaluation metric which is represented by the mean average precision (*mAP*). The IoU is calculated as the overlap of the predicted bounding box and the ground truth bounding box, which is given in Eq. 1. 3D object detection is evaluated using 3D volume Intersection over Union (IoU) metric defined in [88]. To calculate the IoU of the 3D bounding box, the box is assumed to be aligned with the gravity direction, but there are no assumptions on the other two axes. A predicted bounding box is considered as true positive if IoU is greater than certain threshold (normally 0.25 for 3D bounding box and 0.5 for 2D bounding box), otherwise it is considered as a false positive.

$$\text{IoU} = \frac{BBox_{pred} \cap BBox_{gt}}{BBox_{pred} \cup BBox_{gt}} \tag{1}$$

In practice, the precision-recall curve is used to observe the best balance between precision and recall. The *Average Precision* (AP) numerically summarizes the shape of the precision-recall curve, and is defined as the mean precision over N equally spaced discrete levels of recall $\{r_n\}_{n=1}^{N}$ [113], which is given as

$$\text{AP} = \frac{1}{N} \sum_{n=1}^{N} \max_{n \leq i \leq N} p(r_i) \tag{2}$$

where $p(r_i)$ is the calculated precision at recall r_i. The value of N is set to 11. After calculating AP of each class, the mean Average Precision (*mAP*) is calculated for overall performance evaluation.

On the other hand, another type of evaluation method presented in [19] called *Average Orientation Similarity* (AOS). [19] suggest to evaluate using both AP_2D (IoU on the 2D image plane) and AOS metrics. AOS measures the performance of jointly 2D detection and 3D orientation by weighting

the 2D average precision (AP_2D) score on the image plane with cosine similarity between the predicted and ground-truth orientations. AOS is define as:

$$\text{AOS} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} \max_{\tilde{r}: \tilde{r} \geq r} s(\tilde{r}) \tag{3}$$

where $r = \frac{TP}{TP+FN}$ is the recall and the orientation similarity $s \in [0, 1]$ is normalized by the cosine similarity defined as

$$s(r) = \frac{1}{|\mathcal{D}(r)|} \sum_{i \in \mathcal{D}(r)} \frac{1 + cos\Delta_{\theta}^{(i)}}{2} \delta_i \tag{4}$$

where $\mathcal{D}(r)$ is the set of all object detection at recall rate r, $\Delta_{\theta}^{(i)}$ is the angle difference between estimated and ground truth orientation of detection i and δ_i defines if a detection i has been assigned to a ground truth bounding box or not.

Chen *et al.* [8] introduce bird's eye view metric (AP_BV) by projecting 3D detections on the bird's eye view. Ku *et al* [9] extend the AOS metric with Average Heading Similarity (AHS), which is basically a AOS but evaluated using 3D IoU and global orientation angle instead of 2D IoU and observation angle. AHS use the 3D volume Average Precision (AP_3D) metric weighted by the cosine similarity of predicted and ground-truth orientations.

V. Performance Comparisons of Existing Methods

We have reviewed the current state-of-the-art deep learning methods for 3D object detection. In this section, we compare the quantitative performance of the state-of-the-art 3D object detection methods on three publicly available benchmark datasets, including KITTI [19], SUN RGB-D [25] and NYUv2 [20] dataset.

KITTI [19] is an important dataset for outdoor 3D object detection that provides images as well as LiDAR point cloud data, focusing on autonomous driving platform. It contains 3 main object categories, (e.g. cars, pedestrians and cyclists) with 3 difficulty levels namely *easy*, *moderate* and *hard*. Table III and Table IV combine and summarize the performance comparisons of 3D object detection and bird's eye view detection (3D localization) on the KITTI *test* benchmark, respectively. Among these methods, some use only LiDAR data or RGB image as input, while others use RGB images and LiDAR data. There is a large performance gap between only image-based method and only LiDAR point cloud-based methods in Table III. Since image-based method cannot capture depth information, it shows very limited performance on 3D detection.

SUN RGB-D [25] is a large-scale dataset containing densely annotated indoor scenes captures from RGB-D sensors. The dataset is widely used for indoor object detection and segmentation tasks. In Table V, we summarize the comparisons of state-of-the-art 3D object detection approaches for 10-class evaluation on SUN RGB-D dataset. We also show the performance comparisons on the NYUv2 [20] dataset in Table VI. Among these methods, some use RGB images and point clouds, some use only point clouds and others use both RGB and depth images as inputs.

Overall, It can be seen that the recent work trend is to use point cloud data to make full use of 3D information. However, due to its sparse, unordered and unstructured nature, it is very difficult to process point clouds. Therefore, traditional architecture applies some sort of discretization process to make it structured forms, such as 3D voxel or 2D perspective views. However, after presenting an innovative idea by Qi *et al.* in [12] to handle the unordered raw point cloud, the current trend of 3D object detection is going to change. Although there are already some promising works that convert the point cloud to other representation, it is clearly seen that processing raw point cloud directly by CNNs can boost the 3D object detection performance without any other representations that require excessive time-consuming computations. In addition, it can also observe that most methods are not able to operate in real-time scenario considering the speed.

VI. Conclusion and Discussion

In this paper, we reviewed the state-of-the-art deep learning techniques for 3D object detection. Recent works demonstrate the state-of-the-art results using RGB image, point cloud and fusion-based techniques. We addressed the advantages and disadvantages of each core technique. In addition, we presented a performance comparison among the recent methods on the different datasets and highlighted with their input representations. Since deep learning methods for 3D object detection is not as mature as 2D object detection, there is still a large gap between them. The state-of-the-art method still not achieved that performance, which is needed for real-time operation. Therefore, significant improvement must be made in order to obtain a fast and reliable 3D object detection system operating in a wide set of real-time practical applications. Additionally, it can be seen that the recent trend is to use direct unordered point cloud processing, which provides a simple but potentially very effective solution for 3D object detection. However, these methods still face the problem of large empty values in the point cloud. Therefore, more exploration is needed to find the geometrical relationship among the points. Fusion-based approaches also become popular due to the increasing number of multi-modal datasets. Accordingly, most of the method we reviewed fuse RGB images either with LiDAR point cloud or depth images from RGB-D cameras. However, there is no certain confirmation that one fusion method is superior than the others. So, it needs to pay more focus on the multi-modal methods. More recently, only [114] proposed to fuse RGB images with the Radar points, but no large datasets available for this purpose. Due to the scarcity of large-scale annotated training data, more datasets and fusion methods related to Radar signals are expected in the near future. In addition, there are very few methods make a unified architecture that work in both indoor and outdoor scenarios. We believe that in this case, more research should be explored, how to combine different distributions of objects and different scene areas between indoor and outdoor to form a unified 3D object detection architecture. From this study, we can conclude that 3D object detection has been approached with many successes, but still it remains open problems that require more exploration. We

TABLE III
PERFORMANCE COMPARISON OF STATE-OF-THE-ART 3D OBJECT DETECTION ON THE KITTI [19] *test* BENCHMARK

Method	Input	Car			Pedestrian			Cyclist			mAP(Mod)	Speed(s)
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard		
Mono3D [58]	Image	2.53	2.31	2.31	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Deep3DBox [49]	Image	5.84	4.09	3.83	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
OFT-Net [63]	Image	3.28	2.50	2.27	1.06	1.11	1.06	0.43	0.43	0.43	1.35	0.50
3DOP [57]	Stereo Image	6.55	5.07	4.10	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
VeloFCN [68]	LiDAR	15.20	13.66	15.98	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
VoxelNet [69]	LiDAR	77.47	65.11	57.73	39.48	33.69	31.50	61.22	48.36	44.37	49.05	0.23
SECOND [96]	LiDAR	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90	56.69	0.038
PointPillars [112]	LiDAR	79.05	74.99	68.30	52.08	43.53	41.49	75.78	59.07	52.92	59.20	0.016
BirdNet [75]	LiDAR	14.75	13.44	12.04	14.31	11.80	10.55	18.35	12.43	11.88	12.56	0.11
PointRCNN-v1.1 [110]	LiDAR	85.94	75.76	68.32	49.43	41.78	38.63	73.93	59.60	53.59	59.05	0.10
MV3D [8]	Image & LiDAR	71.09	62.35	55.12	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.36
UberATG-ContFuse [100]	Image & LiDAR	82.54	66.22	64.04	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.06
RoarNet [99]	Image & LiDAR	84.25	74.29	59.78	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.10
AVOD [9]	Image & LiDAR	73.59	65.78	58.38	38.28	31.51	26.98	60.11	44.90	38.80	47.40	0.08
AVOD-FPN [9]	Image & LiDAR	81.94	71.88	66.38	50.80	42.81	40.88	64.00	52.18	46.61	55.62	0.10
F-PointNet [65]	Image & LiDAR	81.20	70.39	62.19	51.21	44.89	40.23	71.96	56.77	50.39	57.35	0.17

TABLE IV
PERFORMANCE COMPARISON OF STATE-OF-THE-ART BIRD'S EYE VIEW (BEV) DETECTION ON THE KITTI [19] *test* BENCHMARK

Method	Input	Car			Pedestrian			Cyclist			mAP(Mod)	Speed(s)
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard		
OFT-Net [63]	Image	9.50	7.99	7.51	1.93	1.55	1.65	0.79	0.43	0.43	3.32	0.50
UberATG-PIXOR [78]	LiDAR	81.70	77.05	72.95	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.035
VoxelNet [69]	LiDAR	89.35	79.26	77.39	46.13	40.74	38.11	66.70	54.76	50.55	58.25	0.23
SECOND [96]	LiDAR	88.07	79.37	77.95	55.10	46.27	44.76	73.67	56.04	48.78	60.56	0.038
PointPillars [112]	LiDAR	88.35	86.10	79.83	58.66	50.23	47.19	79.14	62.25	56.00	66.19	0.016
PointRCNN-v1.1 [110]	LiDAR	89.47	85.68	79.10	55.92	47.53	44.67	81.52	66.77	60.78	66.66	0.10
BirdNet [75]	LiDAR	75.52	50.81	50.00	26.07	21.35	19.96	38.93	27.18	25.51	33.11	0.11
UberATG-PIXOR++ [82]	LiDAR	89.38	83.70	77.97	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.035
UberATG-HDNET [82]	LiDAR & Map	89.14	86.57	78.32	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.05
MV3D [8]	Image & LiDAR	86.02	76.90	68.49	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.36
UberATG-ContFuse [100]	Image & LiDAR	88.81	85.83	77.33	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.06
RoarNet [99]	Image & LiDAR	88.19	79.77	69.83	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.10
AVOD-FPN [9]	Image & LiDAR	88.53	83.79	77.90	58.75	51.05	47.54	68.09	57.48	50.77	64.11	0.10
AVOD [9]	Image & LiDAR	86.80	85.44	77.73	42.51	35.24	33.97	63.66	47.74	46.55	56.14	0.08
F-PointNet [65]	Image & LiDAR	88.70	84.00	75.33	58.09	50.22	47.20	75.38	61.96	54.68	65.39	0.17

TABLE V
PERFORMANCE COMPARISONS OF STATE-OF-THE-ART 3D OBJECT DETECTION FOR 10-CLASSES EVALUATION ON SUN RGB-D [25] DATASET.

Method	Key Input Processing											mAP	Speed
COG [93]	Point Cloud	58.26	63.67	31.80	62.17	45.19	15.47	27.36	51.02	51.29	70.07	47.63	10-30m
LSS [94]	Point Cloud	76.20	73.20	32.90	60.50	34.50	13.50	30.40	60.40	55.40	73.70	51.00	10-30m
2D-driven [66]	Image & Depth	43.45	64.48	31.40	48.27	27.93	25.92	41.92	50.39	37.02	80.40	45.12	4.15s
Rahman *et al.* [67]	Image & Depth	44.10	78.10	12.00	54.40	19.70	33.10	44.50	52.10	37.80	80.90	45.70	0.30s
DSS [7]	Image & Point Cloud	44.20	78.80	11.90	61.20	20.50	6.40	15.40	53.50	50.30	78.90	42.10	19.55s
F-PointNet [65]	Image & Point Cloud	43.30	81.10	33.30	64.20	24.70	32.00	58.10	61.10	51.10	90.90	54.00	0.12s
PointFusion [98]	Image & Point Cloud	37.26	68.57	37.69	55.09	17.16	23.95	32.33	53.83	31.03	83.80	45.38	1.30s
SIFRNet [107]	Image & Point Cloud	64.00	84.40	38.40	57.90	34.10	32.20	67.70	67.30	51.40	86.20	58.40	N/A

TABLE VI
PERFORMANCE COMPARISONS OF STATE-OF-THE-ART 3D OBJECT DETECTION FOR 19-CLASSES EVALUATION ON NYUV2 [20] DATASET.

Method	Key Input Processing	mAP	Speed (s)
Deng *et al.* [40]	Image & Depth	40.9	0.74s
Rahman *et al.* [67]	Image & Depth	43.1	0.30s
DSS [7]	Image & Point Cloud	36.3	19.55s

hope that this survey will serve as a supportive reference and a significant contribution to the research community.

REFERENCES

[1] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 6, pp. 1137–1149, 2017.

[2] J. Redmon, S. K. Divvala, R. B. 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 (CVPR)*, 2016, pp. 779–788.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Proceedings of the 14th European Conference on Computer Vision ECCV*, 2016, pp. 21–37.

[4] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in

Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988.

[5] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "Deep learning advances in computer vision with 3d data: A survey," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 20:1–20:38, 2017.

[6] M. Naseer, S. Khan, and F. Porikli, "Indoor scene understanding in 2.5/3d for autonomous agents: A survey," *IEEE Access*, vol. 7, pp. 1859–1887, 2019.

[7] S. Song and J. Xiao, "Deep sliding shapes for amodal 3d object detection in RGB-D images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 808–816.

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

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

[10] R. B. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[11] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3d voxel patterns for object category recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1903–1911.

[12] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

[13] S. Gupta, R. B. Girshick, P. A. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proceedings of the 13th European Conference on Computer Vision (ECCV)*, 2014, pp. 345–360.

[14] H. Badino, U. Franke, and D. Pfeiffer, "The stixel world - A compact medium level representation of the 3d-world," in *Pattern Recognition, 31st DAGM Symposium, Jena, Germany, September 9-11, 2009. Proceedings*, 2009, pp. 51–60.

[15] H. P. Eberhardt, V. Klumpp, and U. D. Hanebeck, "Density trees for efficient nonlinear state estimation," in *13th Conference on Information Fusion, FUSION 2010, Edinburgh, UK, July 26-29, 2010*, 2010, pp. 1–8.

[16] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3-d object dataset: Putting the kinect to work," in *IEEE International Conference on Computer Vision Workshops, ICCV*, 2011, pp. 1168–1174.

[17] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1817–1824.

[18] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *Advances in Neural Information Processing Systems*, 2011, pp. 244–252.

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

[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, 2012, pp. 746–760.

[21] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using sfm and object labels," in *IEEE International Conference on Computer Vision, ICCV*, 2013, pp. 1625–1632.

[22] J. J. Lim, H. Pirsiavash, and A. Torralba, "Parsing IKEA objects: Fine pose estimation," in *IEEE International Conference on Computer Vision, ICCV*, 2013, pp. 2992–2999.

[23] M. De Deuge, A. Quadros, C. Hung, and B. Douillard, "Unsupervised feature learning for classification of outdoor 3d scans," in *Australasian Conference on Robitics and Automation*, vol. 2, 2013, p. 1.

[24] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3d object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision*, 2014, pp. 75–82.

[25] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576.

[26] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[27] Y. Xiang, W. Kim, W. Chen, J. Ji, C. B. Choy, H. Su, R. Mottaghi, L. J. Guibas, and S. Savarese, "Objectnet3d: A large scale database for 3d object recognition," in *European Conference on Computer Vision - ECCV*, 2016, pp. 160–176.

[28] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas *et al.*, "A scalable active framework for region annotation in 3d shape collections," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 210, 2016.

[29] B. Drost, M. Ulrich, P. Bergmann, P. Härtinger, and C. Steger, "Introducing mvtec ITODD - A dataset for 3d object recognition in industry," in *IEEE International Conference on Computer Vision Workshops, ICCV*, 2017, pp. 2200–2208.

[30] T. Hodan, P. Haluza, S. Obdrzálek, J. Matas, M. I. A. Lourakis, and X. Zabulis, "T-LESS: an RGB-D dataset for 6d pose estimation of texture-less objects," in *IEEE Winter Conference on Applications of Computer Vision, WACV*, 2017, pp. 880–888.

[31] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2432–2443.

[32] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," *CoRR*, vol. abs/1702.01105, 2017.

[33] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. A. Funkhouser, "Semantic scene completion from a single depth image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 190–198.

[34] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3d object detection and pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR*, 2018, pp. 2038–2041.

[35] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian, "Building generalizable agents with a realistic and rich 3d environment," in *International Conference on Learning Representations, ICLR*, 2018.

[36] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[37] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, "The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes," in *International Conference on Robotics and Automation*, 2019.

[38] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," in *The International Journal of Robotics Research*, 2019.

[39] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 564–571.

[40] Z. Deng and L. J. Latecki, "Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 398–406.

[41] E. Barnea and O. Ben-Shahar, "Depth based object detection from partial pose estimation of symmetric objects," in *European Conference on Computer Vision - ECCV*, 2014, pp. 377–390.

[42] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[43] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[44] B. Çalli, A. Singh, A. Walsman, S. S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *International Conference on Advanced Robotics, ICAR*, 2015, pp. 510–517.

[45] M. Cheng, Z. Zhang, W. Lin, and P. H. S. Torr, "BING: binarized normed gradients for objectness estimation at 300fps," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3286–3293.

[46] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 391–405.

[47] P. Krähenbühl and V. Koltun, "Learning to propose objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1574–1582.

[48] T. S. H. Lee, S. Fidler, and S. J. Dickinson, "Learning to combine mid-level cues for object proposal generation," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1680–1688.

[49] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5632–5640.

[50] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, "Deep MANTA: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1827–1836.

[51] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 924–933.

[52] B. Pepik, M. Stark, P. V. Gehler, T. Ritschel, and B. Schiele, "3d object class detection in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–10.

[53] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic, "Seeing 3d chairs: Exemplar part-based 2d-3d alignment using a large dataset of CAD models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3762–3769.

[54] J. J. Lim, A. Khosla, and A. Torralba, "FPM: fine pose parts-based model with 3d CAD models," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 478–493.

[55] M. Z. Zia, M. Stark, and K. Schindler, "Explicit occlusion modeling for 3d object class representations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3326–3333.

[56] Y. Lin, V. I. Morariu, W. H. Hsu, and L. S. Davis, "Jointly optimizing 3d model fitting and fine-grained classification," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 466–480.

[57] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, 2015, pp. 424–432.

[58] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2147–2156.

[59] C. C. Pham and J. W. Jeon, "Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks," *Sig. Proc.: Image Comm.*, vol. 53, pp. 110–122, 2017.

[60] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate $O(n)$ solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[61] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision - ECCV*, 2016, pp. 354–370.

[62] B. Xu and Z. Chen, "Multi-level fusion based 3d object detection from monocular images," in *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2018, pp. 2345–2353.

[63] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3d object detection," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.

[64] Y. Wang, W. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," 2019.

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

[66] J. Lahoud and B. Ghanem, "2d-driven 3d object detection in rgb-d images," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4632–4640.

[67] M. M. Rahman, Y. Tan, J. Xue, L. Shao, and K. Lu, "3D object detection: Learning 3d bounding boxes from scaled down 2d bounding boxes in rgb-d images," *Information Sciences*, vol. 476, pp. 147–158, 2019.

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

[69] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[70] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 5105–5114.

[71] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 945–953.

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

[73] K. Minemura, H. Liau, A. Monrroy, and S. Kato, "Lmnet: Real-time multiclass object detection on CPU using 3d lidar," *CoRR*, vol. abs/1805.04902, 2018.

[74] S. Yu, T. Westfechtel, R. Hamada, K. Ohno, and S. Tadokoro, "Vehicle detection and localization on bird's eye view elevation images using convolutional neural network," in *IEEE International Symposium on Safety, Security and Rescue Robotics(SSRR)*, 2017, pp. 102–109.

[75] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. de la Escalera, "Birdnet: A 3d object detection framework from lidar information," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3517–3523.

[76] S. Wirges, T. Fischer, C. Stiller, and J. B. Frias, "Object detection and classification in occupancy grid maps using deep convolutional networks," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3530–3535.

[77] X. Du, M. H. Ang, S. Karaman, and D. Rus, "A general pipeline for 3d detection of vehicles," in *IEEE International Conference on Robotics and Automation, ICRA*, 2018, pp. 3194–3200.

[78] B. Yang, W. Luo, and R. Urtasun, "PIXOR: real-time 3d object detection from point clouds," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7652–7660.

[79] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3569–3577.

[80] M. Simon, S. Milz, K. Amende, and H. Gross, "Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds," in *European Conference on Computer Vision - ECCV*, 2018, pp. 197–209.

[81] W. Ali, S. Abdelkarim, M. Zidan, M. Zahran, and A. E. Sallab, "YOLO3D: end-to-end real-time 3d oriented object bounding box detection from lidar point cloud," in *European Conference on Computer Vision - ECCV*, 2018, pp. 716–728.

[82] B. Yang, M. Liang, and R. Urtasun, "HDNET: exploiting HD maps for 3d object detection," in *Conference on Robot Learning (CoRL)*, 2018, pp. 146–155.

[83] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[84] X. Zhao, H. Jia, and Y. Ni, "A novel three-dimensional object detection with the modified you only look once method," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, p. 1729881418765507, 2018.

[85] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2015, pp. 3431–3440.

[86] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017.

[87] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in *International Conference on Intelligent Robots and Systems IROS*, 2017, pp. 1513–1518.

[88] S. Song and J. Xiao, "Sliding shapes for 3d object detection in depth images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 634–651.

[89] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 89–96.

[90] D. Z. Wang and I. Posner, "Voting for voting in online point cloud object detection," in *Robotics: Science and Systems*, 2015.

[91] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1355–1361.
[92] W. Liu, R. Ji, and S. Li, "Towards 3d object detection with bimodal deep boltzmann machines over RGBD imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3013–3021.
[93] Z. Ren and E. B. Sudderth, "Three-dimensional object detection and layout prediction using clouds of oriented gradients," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1525–1533.
[94] ——, "3d object detection with latent support surfaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 937–946.
[95] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox, "Orientation-boosted voxel nets for 3d object recognition," in *British Machine Vision Conference (BMVC)*, 2017.
[96] Y. Yan, Y. Mao, and B. Li, "SECOND: sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.
[97] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *CoRR*, vol. abs/1801.07829, 2018.
[98] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 244–253.
[99] K. Shin, Y. P. Kwon, and M. Tomizuka, "Roarnet: A robust 3d object detection based on region approximation refinement," in *IEEE Intelligent Vehicles Symposium, IV*, 2019.
[100] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *European Conference on Computer Vision - (ECCV)*, 2018, pp. 663–678.
[101] X. Du, M. H. Ang, and D. Rus, "Car detection for autonomous vehicle: LIDAR and vision fusion approach through deep learning framework," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, 2017, pp. 749–754.
[102] D. Matti, H. K. Ekenel, and J. Thiran, "Combining lidar space clustering and convolutional neural networks for pedestrian detection," in *14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017, Lecce, Italy, August 29 - September 1, 2017*, 2017, pp. 1–6.
[103] S. Wang, S. Suo, W. Ma, A. Pokrovsky, and R. Urtasun, "Deep parametric continuous convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2589–2597.
[104] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.
[105] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.
[106] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," *CoRR*, vol. abs/1701.06659, 2017.
[107] X. Zhao, Z. Liu, R. Hu, and K. Huang, "3d object detection using scale invariant and feature reweighting networks," in *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019, pp. 9267–9274.
[108] M. Jiang, Y. Wu, and C. Lu, "Pointsift: A sift-like network module for 3d point cloud semantic segmentation," *CoRR*, vol. abs/1807.00652, 2018.
[109] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.
[110] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
[111] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[112] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
[113] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
[114] S. Chadwick, W. Maddern, and P. Newman, "Distant vehicle detection using radar and vision," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8311–8317.

Mohammad Muntasir Rahman received Ph.D. degree in Computer Applied Technology from the School of Engineering Sciences, University of Chinese Academy of Sciences, Beijing, China, in 2019. He also received B.Sc and M.Sc degree in Computer Science and Engineering from the Islamic University, Kushtia, Bangladesh, in 2005 and 2006, respectively. Currently, he is an Associate Professor in the Dept. of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh. His research interest include computer vision, machine learning and pattern recognition.

Yanhao Tan is a Master-Docter combined program graduate student in Computer Applied Technology from the University of Chinese Academy of Sciences, Beijing, China, from 2015. His research interest includes deep learning and computer vision, RGB-D object recognition and detection.

Jian Xue was born in Jiangsu, China, in 1979. He received the Ph.D. degree in Computer Applied Technology from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2007. He is currently an Associate Professor with the University of Chinese Academy of Sciences, Beijing. Since 2003, he has long been engaged in the research work about out-of-core medical image analysis and processing, and visualization in scientific computing. His current research interests include image processing, computer graphics and scientific visualization.

Ke Lu was born in Ningxia on March 13th, 1971. He received master degree and Ph.D. degree from the Department of Mathematics and Department of Computer Science at Northwest University in July 1998 and July 2003, respectively. He worked as a postdoctoral fellow in the Institute of Automation Chinese Academy of Sciences from July 2003 to April 2005. Currently he is a professor of the University of the Chinese Academy of Sciences. His current research areas focus on computer vision, 3D image reconstruction and computer graphics.