Lidar Point Cloud Compression, Processing and Learning for Autonomous Driving

Rashid Abbasi, Ali Kashif Bashir, *Senior Member, IEEE,* Hasan J. Alyamani, Farhan Amin, *Graduate Student Member, IEEE,* Jaehyeok Doh, Jianwen Chen, *Senior Member, IEEE,*

***Abstract*—As technology advances, cities are getting smarter. Smart mobility is the key element in smart cities and Autonomous Driving *(AV)* are an essential part of smart mobility. However, the vulnerability of unmanned vehicles can also affect the value of life and human safety. In this paper, we provide a comprehensive analysis of *3D* Point-Cloud *(3DPC)* processing and learning in terms of development, advancement, and performance for the *AV* system. *3DPC* has recently attracted growing interest due to its extensive applications, such as autonomous driving, computer vision, and robotics. Light Detection and Ranging Sensors *(LiDAR)* is one of the most significant sensors in *AV*, which collects *3DPC* that can accurately capture the outer surfaces of scenes and objects. Learning and processing tools in the *3DPC* are essential for creating maps, perceptions, and localization devices in *AV*. The intention behind *3DPC* learning and practical processing tools is to be considered the most essential modules to create, locate, and perceive maps in an *AV* system. The goal of the study is to know "what has been tested in *AV* system so far and what is necessary to make it safer and more practical in *AV* system." We also provide insights into the necessary open problems that are required to be resolved in the future.**

***Keywords*-1 Self-Driving Cars, Cybersecurity, *3D LiDAR* data, Object Detection and Tracking, Vehicle safety, Deep Learning**

I. Introduction

THE unprecedented impact of *Covid-19* has hard up the world to accelerate the process of digitization and automation at a quicker pace than expected. Avoid any physical contact that could be a factious touch to humans due to the cost of precious human lives. The idea of driverless smart cars is a rapidly evolving technology. However, the vulnerability of unmanned vehicles can also affect the value of life and human safety [1], [2], [3], [4]. Threats to Autonomous Vehicles *(AV)* can come from any system linked to *AV* sensors, processors, communications applications, and control systems, in addition to an external data source from vehicles, infrastructure, maps, roads, and *GPS* data systems. Technology advancements have influenced important improvements in transportation infrastructure. These days, several Intelligent Transportation Systems *(ITS)* are proposed to help travelers. In order to better educate users and facilitate safer, more coordinated, intelligent transportation networks and smarter use of transportation systems, [5], [6].

The smart city is a rapidly growing concept that monitors the physical world in real time and provides smart facilities to residents in the areas of the environment, entertainment, transportation, and energy. However, because smart cities collect sensitive data, there are concerns about data security that require high levels of privacy in a smart city network. As smart city systems need to act quickly, there is a growing need for algorithms that are computationally efficient. Reversible data hiding plays a very significant role in remote sensing data. The *LiDAR* sensor's data is processed, analyzed, and confirmed. Reversible data hiding is often less resource-intensive, and it can incorporate a perceptible, reversible, or non-reversible data hiding signature into existing data that the end process or application can handle natively. This is valuable for real-time data processing. Watermarking data hiding enables traceability and security for autonomous driving applications. The encoded signature remains in the payload until the data is received and analyzed [15], [20], [21], [22], [7].

Light Detection And Ranging *(LiDAR)* sensors are primarily used for navigation in *AV* because they are perceived to provide a better and more supportive awareness of the objects [7]. However, *3DPC* information reserves a significant detail of the surroundings during the process of navigation, but managing a substantial amount of required data in a real-time situation is quite complex. Consequently, the researchers have experimented with a lot of various algorithms to use sizeable data of *3DPC* during the operational process by applying different techniques of *3DPC* transformations. One of the most key compression approaches has been used to handle the enormous volume of *3DPC* data, certainly [7]. However, if captured data is stored in a compressed form, we need to decompress it before doing any processing. The de-compression process required a considerable cost in terms of space and computation during the real-time process, shown in Figure 1 2.

Consequently, the process of retrieving and decompressing data for processing without being decompressed is referred to as Compressed Domain Processing *(CDP)* [8], [9], [11]. Optimize both the computational and operational costs, as well as the storage costs. The phenomena of *CDP* have already

Manuscript received month date, year. This work was supported by the Sichuan Science and Technology Program under Grant 2018RZ0070

Rashid Abbasi, Jianwen Chen, Ali Kashif Bashir are with the School of Information and Communication Engineering, University of Electronics Science and Technology of China. (e-mail: rashid.abbasi@uestc.edu.cn), Corresponding author: Jianwen Chen. (chenjianwen@uestc.edu.cn)

Ali Kashif Bashir is with the Department of Computing and Mathematics, Manchester Metropolitan University Manchester, United Kingdom.

Hasan J. Alyamani is with teh Department of Information Systems, Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Rabigh 21911, Saudi Arabia.

Farhan Amin is with the Department of Computer Engineering, Gachon University, South Korea.

Jaehyeok Doh is with the School of Mechanical Engineering, Gyeongsang National University, Republic of Korea-52725.

TABLE II
ABBRIVATION

3DPC	3D Point Cloud Processing
AV	Autonomous Driving
PC	Point Cloud
CDP	Compressed Domain processing
V-PCC	Video Cloud Based Point Compression
G-PCC	Geometry-Based Point Cloud Compression
PSNR	Peak signal to Noice Ratio
KPConv	Kernel Point Convolution
SLP	Single Layer Perception
MLP	Multiple Layer Perception
FCN	Fully Convolution Network
FP	Feature Propagation
GPS	Global Positioning System
AVS	Autonomous Vehicle System
ADM	Autonomous Driving Map

been applied successfully [8], [9]. The main contributions of this work can be summarized as follows:

- Deep Learning: We did our best to cover all significant deep learning *DL* techniques used in *3DPC* for many tasks, including *3D* shape classification, *3D* object detection, object tracking, and *3DPC* segmentation comprehensively. Furthermore, the detailed comparisons of current methods have provided concise summaries of many publicly available data sets on the subject topics.
- Limited computing power: *AV* computing performance is often limited compared to the computational performance of computing power. This limitation is because *AV* has a longer lifespan, and its endurable temperature and vibration are higher than conventional computing systems. Limited *AV* computational performance is also the main reason that some vehicular cybersecurity solutions will have a prohibitively high overhead to execute.
- Significant risks to the lives of drivers or passengers: The few proscribed messages transferred or sensors misled eventually cause vehicle malfunctions, which put the lives of passengers, pedestrians, and drivers at risk.
- Artificial Intelligence and big data: The research trend on autonomous vehicles' safety shows that artificial intelligence combined with big data can be used to defend against attacks on self-driving cars.

Table II contains all of the abbreviations. The rest of the paper is organized as follows: Sect. 2 provided the evaluation of projection and mapping of *1D* and *3D* compression domains, while Sect.3 deliberated on the implications and present state of autonomous driving. Sect.4 discussed real-world challenges and conclusion in sect.5.

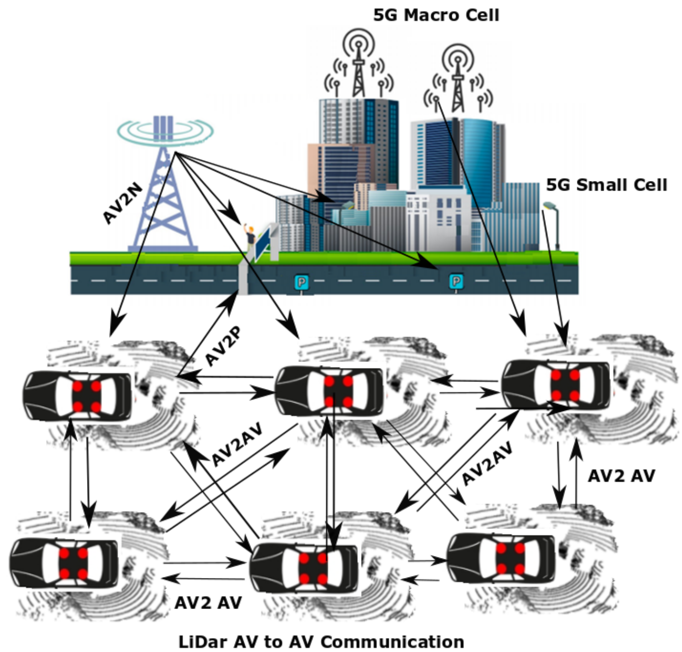


Fig. 1. The exchange of information is made more difficult by the high mobility of the vehicles and the fast variations in the network topology. *VANET* [12] security requirements include user authentication, data integrity, confidentiality, scalability, data protection, and portability. The *Lidar*-based *V2V* authentication mechanism can authenticate a vehicle even if it cannot connect to a dedicated group due to non-existent infrastructure. This protocol detects nearby vehicles by using sensors pre-installed in the *AV*. *AV2P* represents vehicle-to-pedestriancommunication, *AV2AV* represents vehicle-to-vehicle communication, and *AV2N* represents vehicle-to-network communication.

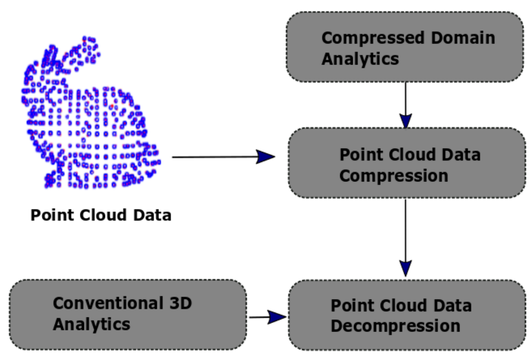


Fig. 2. The conventional model of *3D* analysis comprises the components of *3D* compressed data domain analytics, *3DPC* data compression, and *3DPC* data decompression for prediction of future base outcomes.

II. Projection and Mapping of $1D$, $2D$ Compression Domains

Today, *PC* are widely used in many applications, including surveying and *3D* modeling, environmental monitoring, agriculture and forestry, biomedical imaging, *CAD* and autonomous driving. Geometric information indicates the position of a point at given point coordinates as *(X, Y, Z)*. Attribute data labels the appearance of each point inversely. Geometric coordinates are usually expressed as floating-point values; conversely, they can be used as integer representations of the coordinates, which helps to save *CPU* calculation, time and improve memory efficiency [13], [14]. Lossless compression *(LC)* generates the compressed data by identifying and eliminating statistical redundancies and preserving the original information. *LC* reduces the size of the data by eliminating redundant and visually unusable data during the quantization process. Three different *3D* modes are (*3D* decorrelation, *2D* projection, and *1D* traversal). Overlapping with *2D* projection and *1D* traversal levels, *3D* decorrelation relates to the data structure that involves geometric information processing, as shown in Table I, III,IV.

In the one-dimensional estimation approach, the basic stan-

TABLE I
1D COMPRESSION TECHNIQUES

	[24]	[26]	[25]	[27]
Compression rates(bpp-Geometry)	16.12	15.51	10.47	11.08 Axial mode
Compression rates (bpp-Attributes)	-	2:11	-	-
Lossless Capable	√	×	×	×
Input Type	Static-float	Static-float	Static-float	Static-float
Outcomes	Poor with rigid surfaces	Real-time 5 to 6 fps Spatio-temporal correlations, AR/MR/VR content,	3D one-shot scanning module, Efficient using with video coding,	Efficient Compression,

dard is to make connections between trees based on the proximity relationship created by the actual geometric distance between the *PC* information. Furthermore, two additional methods, general predictive encoder [16] and kd-tree reference [17] approach, which is designed for *3D* mesh compression but can also be directly applying for *3DPC* geometry compression. A combination of octave-based geometric and color-based image coding has emerged in recent years to open a new avenue in ongoing research. In [18], [19] techniques, the main idea of *V-PCC* is to use existing video codecs to compress geometry and texture information dynamically; simultaneously, Voxel-based adaptive approach introduced for real-time[23].

Thus, the response time is mandatory by the application and the available network bandwidth is used to calculate the target speed and compression ratio. The *3DPC* is decomposed into *2D* images as in [20], [21], [22], [23] and then compressed using a *JPEG* image encoder. The pixel size is scaled to a higher or lower resolution in a way that ensures a trade-off between quality and compression ratio. Scattered voxel arrays are described as scattered *3DPC*, which are obtained by decomposition and placing *PC* space in blocks.

A. *Normalization of* 3DPC *compression*

After identifying the growing demand for *3DPC* compression technologies in the consumer electronics industry, the Moving Picture Experts Group *3D* Graphics Suite *(MPEG-3DG)* has now been commercialized. Within this framework, there are currently two types of compression applications in use: (a) Video Cloud Based Point Compression *(V-PCC)*, (b) Geometry-Based Point Cloud Compression *(G-PCC)*. *V-PCC* is a complex video compression technology that aims to provide low-complexity decoding capabilities for an application that requires real-time decoding. For example, virtual or augmented reality, immersive communication, *V-PCC* take advantage of current and future video compression technologies, as well as the comprehensive video ecosystem (hardware acceleration, streaming services, and infrastructure). The implementation of the *MPEG Meeting 124* reference model encoder exhibits compression ratios of $125:1$ with good cognitive quality. *G-PCC* is well-known for its efficient lossless and lossy compression technique, which is used in *AV*, *3D* maps, and other applications that rely on *LiDAR* generated *PC*. Several geometric-driven approaches are encompassed in the *G-PCC* framework.

B. *Analysis in practice techniques of* 3DPC *compression*

3DPC info comes from *LiDAR* which is affixed to the *AV* system. There are various challenges linked with the *LiDAR* produced data to carry out any processing. To counter such an issue, one of the core ideas that has been suggested by researchers as cited in [39] is to use the deep learning-based geometric technique to compress the unprocessed *3DPC* using a hierarchical structure method named the auto-encoder model. The prototype is precise new-fangled and has some similarities through *PointNet++*. The pattern uses an encoder to compress the original *3DPC* data, employing sparse coding. Similarly, reverse approaches are used during the decompression of data with the help of the decoder. They used the multi-metric scattered loss function (Sparse Multiscale Loss Function) and achieved a high compression ratio, and tested with the *ShapeNet40* dataset, in addition to achieving a high-end reconstruct quality, as shown in Table V. In the paper [58], the author proposed the concept of using *RNN* with residual blocks while compressing the captured *3DPC* data obtained from *3D-LiDAR*. Due to the compression proportion and the decompression error, the compression method is very adaptable. The original *2DPC* information is transformed by *LiDAR* into a *2D* matrix, then normalized further before *RNN* used for compression. The author used Bits Per Point *(BpP)* to evaluate the fraction of the information. Subsequently, the compression process measures Symmetric Nearest Neighbor Root Mean Squared Error to assess the loss by decompression technique.

The author [59] proposed a lossless approach to compress and optimize *3DPC* data that preserves the geometric information. They perform segmentation using the regional growth technique for all points within the sealed surface that are intentionally eliminated to achieve successful compression results. On the other hand, a polynomial equation is used to recover the data discarded during decompression. In summary, the raw data obtained from the *3DPC* split into different segments, and a level was assigned to each segment. The given level is erect by a polynomial equation of one degree. With a compression ratio of an *RMSE* value of 0.003 and a time range of $0.0643 - ms$ of processing time, performance is recorded with an accuracy of 89 percent; however, this approach has some limitations when dealing with complex *PC* data processing. This article [60] describes the current *3DPC* compression technology and focuses on design principles such as *1D* traversal, *2D*, clustering, mapping, and projection. However, *2D* is not suitable for high-precision applications such as self-driving cars. Therefore, it is recommended to fully

TABLE III
2D PROJECTION-BASED TECHNIQUES

	[28]	[29]	[30]	[19]
Compression rates(dB-Geometry)	1.20 to 3.41 (60 dB to 87dB)	5000:1 to 50:1 (28 dB to 31 dB)	10 to 18(55dB to 99dB)	0.01 to 0.07(29 dB to 39dB)
Compression rates (dB-Attributes)	-	5000:1 to 50:1	-	0.01 to 0.07(29 dB to 39dB)
Lossless Capable	√	√	×	×
Input Type	Static-float	Dynamic-float	Dynamic-float	Dynamic-int
Outcomes	Poor with rigid surfaces	Real-time 5 to 6 fps Spatio-temporal correlations, AR/MR/VR content,	3D one-shot scanning module, Efficient use with video coding,	Efficient Compression,

TABLE IV
3D COMPRESSION TECHNIQUES

	Compression rates(dB-Geometry)	Compression rates (dB-Attributes)	Lossless Capable	Input Type	Outcomes
Zhang[31]	-	0.16 to 5.36 (28 dB to 52 dB)	×	Static-Integer	Efficient
Golla Klein[32]	-	0.1 to 3.2 (65 dB to 86dB)	×	Dynamic-float	Efficient in storage, Efficient computational cost, Robotics applications
Thanou [33]	-	0,08 to 1.85 (34dB to 44.5dB)	×	Dynamic-Integer	Efficient Compression result, Graph Transformation, Precise Motion Estimation
Queiroz Chou[34]	-	0.85 to 2.5(31.7 dB to 39.8 dB)	×	Static-Integer	Efficient Computational performance, 3D video rate is 30fbs for real time performance
Queiroz Chou[35]	3.7	3.7	×	Dynamic-Integer	Low bit rate index, Optimized motion advantage
Queiroz Chou[36]	-	0.54 to 2.11 (34.2 dB to 41.8dB)	×	Satic Integer	Efficient Compression result, Gaussian Process model
Zhang [37]	-	2.0 (51dB)	√	Satic Integer	Intra Cluster prediction, Lossless Compression, Hierarchical segmentation
Garcia and Queiroz[38]	1.59	1.95	√	Satic Integer	Lossles coding, Enhanced contest used on octree

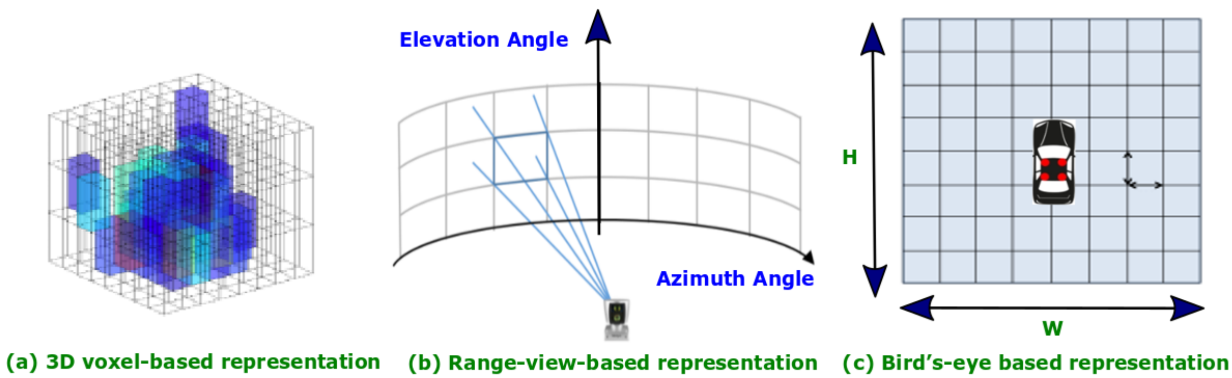


Fig. 3. Collective methods for estimating three-dimensional space into discretization [138], [143]. Voxel-based *3D* rendering aims to distinguish *3D* space into pixels that do not overlap and are evenly spaced for each of the three dimensions. A width based on the bandwidth is the determination of the three-dimensional space along the azimuth angle and the elevation angle; a representation based on the width of a bird's eye is the determination of three-dimensional space along the X and Y axes, neglecting the height dimension.

rely on the *3D* method, which offers the best precision through lossless *3DPC* compression.

The author [61], discussed the basic mechanism that is being used in *3DPC* compression. Subsequently, it evaluates the *TMC3*, *TMC2*, *TMC1* and *TMC13* besides their encoder architectures. It demonstrates that *TMC2* performance is on average for dense *PC* while *TMC13* is optimum outcome for sparse and noisy *PC* with lower time complexity. Furthermore, *TMC2* performs best on normal, dense *PC*, whereas *TMC13* performs best on sparse and noisy *PC* with less time complexity. The author [62] dealing with the compression of *3D* morphological data based on compressed detection using Shannon Nyquist's sampling theory and uses compressed sensing technology to model broadleaf point clouds. To simplify *PC* and eliminate outliers, *Voxel* and statistical filtering are used. Due to its larger size, the *3D* data is then partitionedinto *3D* data portions and *1D* data is organized into distinct arrays, as revealed in Figure 3. In addition, a sparse transformation and a partial Fourier matrix have been used to reduce the sample. To accurately reconstruct the data, Orthogonal Regular Match *(ROMP)* is being employed. In terms of costs, *ROMP* has advantages in terms of both storage and computational. A tree-based architecture has recently been used to compress *LiDAR* data, where the depth of the tree is proportional to the accuracy of the *LiDAR* data. *PC* is isolated from the tree with 8 children, and this process is continued to the specified depth. Another research work carried out by Chenxi [93] about the real-time compression of *3DPC* data transmission technique using a *Unet*-based deep learning network. In their proposed model, they converted the raw *LiDAR-PC* data into a *2D* matrix form. Furthermore, segment the data into *I-frame*and *B-frame*. Then, the *I-frame* would be integrated with the *Unet* architecture while the *Unet* output is combined with *B-frame* to process for the next stage.

In general, *2D* video compression algorithms employ ”motion” by examining similarity trends of pixels in an adjacent macro-block. A local property is determined by this macro-block motion. Local motion information may be used to

TABLE V
CURRENT DATASETS FOR *3DPC*, *3D* OBJECT DETECTION AND TRACKING AND *3D* SHAPE CLASSIFICATION, (−) OR (×) REPRESENT NOT APPLICABLE OR RESULT ARE UNKNOWN, *PC* REPRESENT POINT CLOUD WHILE √ REPRESENT RESULT ARE KNOWN.

		RGB	LiDar	RGB-D	Mesh	MLS	ALS	TLS	Urban	Indoor	Synthetic	Real-World	Classes	PC
3D Point Cloud Segmentation	VMR-Oakland[153]	×	-	-	-	√	×	-	×	×	×	×	44	-
	ISPRS[40]	×	-	-	-	×	√	-	×	×	×	×	9	-
	Pairs-rue-Madame[156]	×	-	-	-	√	×	-	×	×	×	×	17	-
	IQmulus[41]	×	-	-	-	√	×	-	×	×	×	×	22	-
	ScanNet[42]	√	-	√	-	×	×	-	×	×	×	×	22	-
	S3DIS[157]	√	-	√	-	×	×	-	×	×	×	×	13	-
	Semantic3D[154]	√	-	√	-	×	×	√	×	×	×	×	9	-
	Paris-Lille-3D[43]	×	-	×	-	√	×	×	×	×	×	×	50	-
	SemanticKITTI[44]	×	-	×	-	√	×	×	×	×	×	×	28	-
	Toronto-3D[45]	√	×	-	×	√	×	×	×	×	×	×	9	-
	DALES[46]	×	-	×	-	×	√	×	×	×	×	×	9	-
3D Object Detection and Tracking	KITTI[158]	√	√	-	-	-	-	-	√	×	×	×	8	-
	H3D[47]	√	√	-	-	-	-	-	√	×	×	×	8	-
	Argoverse[48]	√	√	-	-	-	-	-	√	×	×	×	15	-
	Lyft L5[49]	√	√	-	-	-	-	-	√	×	×	×	9	-
	A*3D[50]	√	√	-	-	-	-	-	√	×	×	×	7	-
	Waymo Open[51]	√	√	-	-	-	-	-	√	×	×	×	4	-
	nuScenes[52]	√	√	-	-	-	-	-	√	×	×	×	23	-
	SUN RGB-D[53]	×	×	√	-	-	-	-	×	√	×	×	37	-
	ScanNetV2[42]	×	×	×	√	-	-	-	×	√	×	×	18	-
3D Shape Classification	McGill Benchmark[54]	×	×	×	√	-	-	-	×	×	√	×	19	-
	ModelNet10[55]	×	×	×	√	-	-	-	×	×	√	×	10	-
	ModelNet40[55]	×	×	×	√	-	-	-	×	×	√	×	40	-
	ShapeNet[160]	×	×	×	√	-	-	-	×	×	√	×	55	-
	ScanNet[42]	×	×	√	×	-	-	-	×	×	×	√	17	-
	ScanObjectNN[56]	×	×	×	×	-	-	-	×	×	×	√	15	√
	Sydney Urban Objects[57]	×	×	×	×	-	-	-	×	×	×	√	14	√

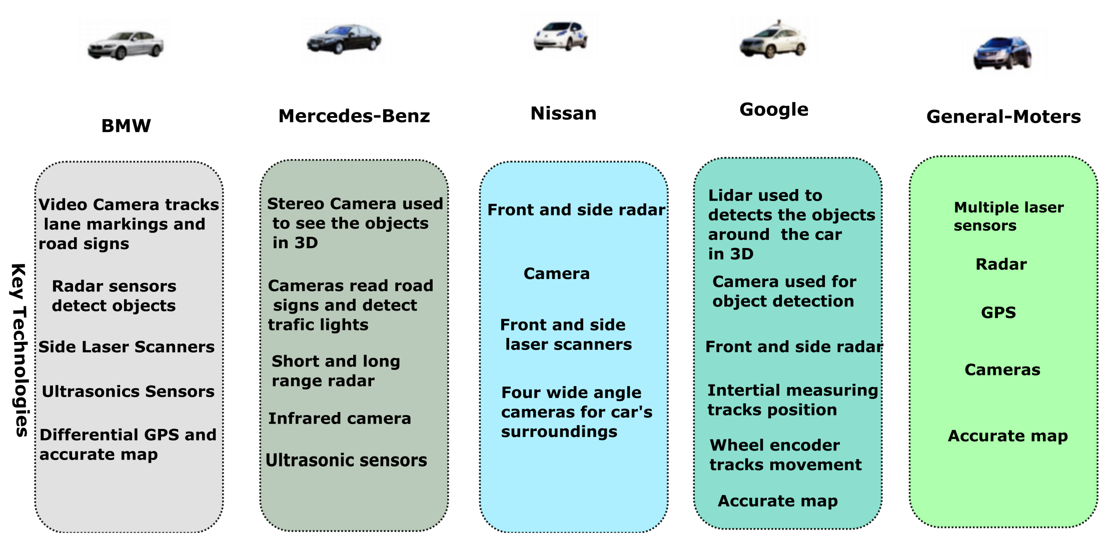


Fig. 5. Key Technologies of *AV* System

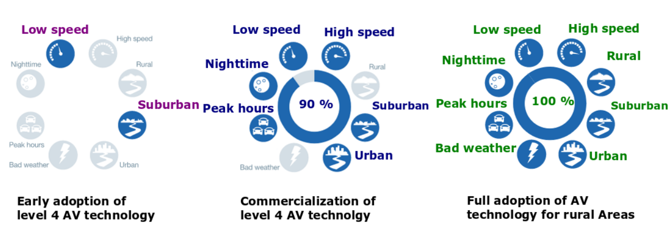


Fig. 4. Different levels of autonomous vehicles: cars represent a transition from high-level human intervention to low-level human intervention. A fully autonomous vehicle can sense its surroundings and operate without human intervention. A self-driving automobile can travel everywhere a car can. It can accomplish everything a human driver can.

compress a sparse point cloud acquired by a *LiDAR* sensor in a vehicle. Due to its sparse point composition, a higher *2D* macro-cubic is required to express redundancy information. Recent deep learning approaches, such as OctSqueeze [95], have been developed to improve *2D* scene compressibility. They keep accuracy after compression but involve point-level data processing, which may not be done in real time. In order to reduce computational complexity, researchers are considering bi-dimensional *2D* transformations of the point cloud using graph algorithms, such as Lossless JPEG *(J-LS)*, Portable Network Graphics *(PNG)*, as well as video-based compression techniques such as Motion JPEG 2000 (MJ2) and Lempel-Ziv-Welch *(LZW)* [96], [98]. The most promising schemes [97] provide accurate but efficient compression for point clouds. A point cloud's Peak signal to Noice Ratio *(PSNR)* and decompression time are two important features that determine how well the compressed point cloud visual aspects and how well it can be decompressed in a short amount of time.

C. 3D *Point Cloud Classification* (3DPCC)

The *3DPC* classification contains projection-based approaches in which data from *PC* is converted *2D* into *3D* using deep learning approaches. By using different direct *PC* processing algorithms, such as the convolution-based network or the graph-based network, they have performed much better in *3DPCC*, as revealed in Table VI. The *3DPC* object-detection [7] is the second most common issue. Object detection is considered the most challenging task in the autonomous car industry. There are two methods used, region-based and single-shot approaches. The first scheme creates potential information areas for the object and then applies to bounding box regression and classification algorithms. The second method relies on two single-layer grids that define the bounding box and the object's class. This technique is faster than the region-based one, as it does not work on the two-stage network. Segmentation is the third most common problem with *3DPC* data, further classified into instance segmentation, semantic segmentation, and part segmentation.

- Point-Based approaches: Depending on the network architecture used to find out the characteristics of each point, the methods in this category are segmented into point-based (e.g., *MLP*), convolution-based, data structure-based, hierarchical graph-based, etc.
- Point-based *MLP* approaches: *MLP* model, each point independently with multiple *MLP's* and then add a common feature using the symmetric grouping function. *3DPC* object detection is the most difficult challenge in autonomous vehicles. The *DL* methods for *2D* images cannot be applied directly to *3DPC* due to their inherent data skewness. PointNet [94] directly takes *PC* as input to verify, depending on the network structure used to discover the properties of each point. Camera sensor *2D* images are full of semantic information. However, only *2D* object detection can no longer provide all the essential information for border perception. For nearly limitless range, high throughput, and fast beam scanning Lidars may capture *2D* images of objects or their surroundings. Time-of-flight *(TOF)* is used in pulsed lidars to calculate range. They have an extensive range and quick detection by emitting repeating high-peak power pulses. This class can also be subdivided into *MLP* based on traditional techniques such as point, convolution, graph, and hierarchical data structure. PointNet uses multiple *MLP* layers to define point features independently, and uses the maximum set of layers to retrieve shared features. Depth set[99] properties are independently predictable for each point in PointNet [94], and it is not possible to get local hierarchical information between points. To capture the geometric structure of a neighborhood point, Qui et al. [100] proposed a PointNet++ hierarchical network. Local geometries features are exposed layer by layer in PointNet++'s hierarchical structure, which consists of three levels: the sample layer, the PointNet-based learning layer, and the aggregation layer.
- Convolution-based Methods: These methods locate convolutional cores in a continuous space. The weights of adjacent spatial distribution points are connected to the central point, while *3D* convolution is interpreted as a given subset's weighted sum. The base class of *RS-Conc* [102], *RS-CNN* [101], accepts both local and adjacent points as input; additionally, *CNN* is implemented using *MLP* to learn low-level map relationships (e.g., Euclidean distance and relative position). In DensePoint [103], the Single Layer Perception *(SLP)* is defined with nonlinear functions and learned through the sequence of tasks from all layers to adequately exploit relative information. Deformable Kernel Point Convolution (KPConv) was proposed by Thomas et al. [104]. Furthermore, by making use of learnable kernel points, ConvPoint [105] segments the convolution kernel into spatial and temporal while the locations of the spatial part are randomly chosen from a unit sphere and the weighting function to learn through a simple *MLP*. PointConv [106], which is defined as a Monte Carlo estimation of continuous *3D* convolution in terms of sampling importance. The convolutional kernel consists of a weighting function (which is learning with *MLP* layers) and a density function (which is learning by estimating the density of the kernel and the *MLP* layer). Estevez et al. [107] proposed a *3D-CNN* that takes multivalued spherical functions as input and local convolutional filters by specifying the spectral parameters of the anchor points in the spherical harmonic domain to learn the iso-rotation representation of *3D* shapes. Speed up the computation speed, Flex-Convolution [108] defines convolution kernel weights as standard nearest-neighbors, which are accelerated by *CUDA*. Thus, the results of the experiments showed that they could compete with each other on a small data set with fewer parameters and less memory use. Hua et al. [109] transformed *3D* irregular point clouds into uniform networks and defined convolutional kernels in each network, assigning the same weights to all points that lie on the same grid. The average properties of all networks are weighted and added together to make the output of the current layer, which is the result of this process. The spherical convolutional kernel is defined by Lee et al. [110] by dividing a *3D* spherical adjacency into multiple volumetric vessels and relating each container to a learnable weighting matrix. This is how the product of a spherical convolutional kernel is made. The nonlinear activation of its adjacent points' average weighted values is what makes this happen.
- Single Shot Methods: Single-shot methods directly predict the class probabilities and return the $3D$ bounding boxes of the objects using a single-stage grid. Consequently, they do not need to create an area proposal; consequently, they can run at high speed (depending on the input data type). The single-shot method is further cat-

TABLE VI

$ModelNet40/10$ STANDARDS COMPARATIVE OUTCOMES FOR *3D* SHAPE CLASSIFICATION. *nAcc* REPRESENT MEAN ACCURACY FOR EACH TEST INSTANCE, *mAcc* DEFINE FOR EACH SHAPE CLASS, $(-)$ OR $(\times)$ REPRESENT NOT APPLICABLE, OR RESULT IS UNKNOWN WHILE $\surd$ REPRESENT RESULT ARE KNOWN.

		Normals	Coordinates	ModelNet10(nAcc)	ModelNet10(mAcc)	ModelNet40(nAcc)	ModelNet40(mAcc)
Pointwise MLPMethods	PointNet [94]	×	√	-	-	89.2	86.2
	PointNet++ [100]	×	√	-	-	90.7	-
	Deep Sets[99]	×	√	-	-	87.1	-
	MO-Net [63]	×	√	-	-	89.3	-
	PointWeb [64]	×	√	-	-	92.3	-
	PointASNL[65]	√	√	95.9	-	93.2	-
	SRN-PointNet++ [66]	×	√	-	-	91.5	-
	PAT[67]	×	√	-	-	91.7	-
Graph-based Methods	PointGCN [68]	×	√	-	-	89.5	-
	Hassani et al.[69]	×	√	-	-	89.1	-
	DPAM [70]	×	√	94.6	94.3	91.9	89.9
	Grid-GCN [71]	×	√	97.5	97.4	93.1	91.3
	RGCNN [72]	√	√	-	-	95.5	97.3
	LocalSpecGCN [73]	√	√	-	-	92.1	-
	DGCNN [74]	×	√	-	-	92.2	90.2
	LDGCNN [75]	×	√	-	-	92.9	90.3
	ClusterNet [76]	×	√	-	-	87.1	-
	KCNet[77]	×	√	94.4	-	91.0	-
	ECC [78]	×	√	90.8	90.0	87.4	83.2-
	3DTI-Net [79]	×	√	-	-	91.7	-
Hierarchical Data Structure -based Methods	3DContextNet [80]	√	√	-	-	91.1	-
	A-SCN [81]	×	√	-	-	89.8	87.4
	KD-Net [82]	×	√	94.0	93.5	91.8	88.5
	SCN [82]	×	√	-	-	90.0	87.6
	SO-Net [108	×	√	94.1	93.9	90.9	87.3
Convolution-based Methods	ConvPoint[105]	×	×	√	-	91.8	88.5
	DensePoint [103]	×	√	96.6	-	93.2	-
	SFCNN [84]	√	√	-	-	92.3	-
	A-CNN [85]	×	√	-	-	92.6	90.3
	KPConv deform [104]	×	√	-	-	92.7	-
	KPConv rigid[104]	×	×	√	92.9	-	
	InterpCNN [86]	×	√	-	-	93.0	-
	GeoCNN [87]	×	×	√	93.4	91.1	
	Spherical CNNs [107]	×	×	√	88.9	-	
	PointConv [106]	√	√	-	-	92.5	-
	Pointwise-CNN [109]	×	√	-	-	86.1	81.4
	SpiderCNN [88]	√	√	-	-	92.4	-
	PointCNN [89]	×	√	-	-	92.2	88.1
	Flex-Convolution [108]	×	√	-	-	90.2	-
	MC Convolution [90]	×	√	-	-	90.9	-
	PCNN [91]	×	√	-	94.9	92.3	-
	Boulch [92]	×	√	-	-	91.6	88.1
	RS-CNN [101]	×	√	-	-	93.6	-

egorized into BEV-based, point-based, and discretization-based approach. Yang et al. [111] identified the scene's *PC* with evenly spaced cells and similarly encoded the reflection in a regular representation. *FCN* method is applied to approximate the positions and directions of the angled objects. *FCN* scheme outperforms most single-shot techniques while running at a speed of $28.6fps$. Yang et al. [112] employ the geometric approach as well as semantic information from *HD* maps to expand the consequences of robustness and detection[111]. Since *HD* maps are not presented everywhere, to go with this, an online map prediction module coupled with a single *LiDAR-PC*. This approach significantly exceeds its baseline approach on the $TOR4D$ [111], [113] and KITTI [114] data sets.

The point-based, instance-segmentation, and convolutional networks were used in the first, second, and third categories, respectively, but the various folds of three-dimensional shape make it difficult to generalize all parts of an object[115]. Discretization-based approaches use *CNN* to predict both classes and *3D* bounding boxes of objects from a point cloud. A Fully Convolution Network *(FCN)* was used for the first time by Li et al.[144]. They employed a *2D-FCN* to estimate object bounding boxes and a *3D* point map from a point cloud. VoxelNet is a voxel-based end-to-end trainable framework proposed by Zhou et al [138]. They divided a *PC* into voxels and stored each voxel's characteristics in a *4D* tensor. Then a regional proposal model is connected to the detector. Due to the sparsity of voxels and *3D* convolutions, the performance of this approach is slow. Yan et al. [111] used a sparse convolutional network to improve the significance efficiency of the Zhou scheme. Image features are combined with voxel characteristics to build precise *3D* boxes. It uses multi-modal information to minimise false positives and negatives, unlike [138], [111]. Point-based schemes use raw point clouds as input. *3DSSD* [147] is an original work. in order to eliminate the time-consuming Feature Propagation *(FP)* layers and the refinement part in [116]. An anchor-free regression through a *3D* centerness label is then exploited to predict *3D* object boxes using a Candidate Generation *(CG)* layer. *3DSSD* beats the two-phase point-based technique PointRCNN [116] while maintaining 25 fps-speed.

TABLE VII

KITTI-3D TEST STANDARDS COMPARATIVE OUTCOMES FOR *3D* OBJECT. $Best_1$, $avearge_1$ AND $worst_1$ REPRESENT WITH 0.5 THRESHOLD VALUE FOR CAR, $Best_2$, $Best_3$ $avearge_2$, $avearge_3$ AND $worst_2$, $worst_3$ REPRESENT FOR PEDESTRIANS AND CYCLISTS WITH 0.5 CLASS OF OBJECT, $(-)$ OR $(\times)$ REPRESENT NOT APPLICABLE OR RESULT ARE UNKNOWN WHILE $\surd$ REPRESENT RESULT ARE KNOWN.

		LiDar	Image	$Best_1$	$Average_1$	$Worst_1$	$Best_2$	$Average_2$	$Worst_2$	$Best_3$	$Average_3$	$Worst_3$	Speed(fbs)
Region Proposal-based Methods	PointRCNN [116]	√	×	86.96	75.64	70.70	47.98	39.37	36.01	74.96	58.82	52.53	10.0
	PointPainting [117]	√	√	82.11	71.70	67.08	50.32	40.97	37.87	77.63	63.78	55.89	2.5
	STD [118]	√	×	87.95	79.71	75.09	53.29	42.47	38.35	78.69	61.59	55.30	12.5
	PointRGCN [119]	√	×	85.97	75.73	70.60	-	-	-	-	-	-	3.8
	IPOD [120]	√	√	80.30	73.04	68.73	55.07	44.37	40.05	71.99	52.23	46.50	5.0
	AVOD[121]	√	√	76.39	66.47	60.23	36.10	27.86	25.76	57.19	42.08	38.29	12.5
	PointFusion[122]	√	√	77.92	63.00	53.27	33.36	28.04	23.38	49.34	29.42	26.98	-
	Patch Refinement[123]	√	×	88.67	77.20	71.82	-	-	-	-	-	-	6.7
	PV-RCNN[124]	√	×	90.25	81.43	76.82	-	-	-	-	-	-	12.5
	VoteNet[125]	√	×	-	-	-	-	-	-	-	-	-	-
	3D IoU loss[126]	√	×	86.16	76.50	71.39	-	-	-	-	-	-	12.5
	F-ConvNet[127]	√	√	87.36	76.39	66.69	52.16	43.38	38.80	81.98	65.07	56.54	2.1
	ImVoteNet[128]	√	×	-	-	-	-	-	-	-	-	-	-
	F-PointNets[129]	√	√	82.19	69.79	60.59	50.53	42.15	38.08	72.27	56.12	49.01	5.9
	RoarNet[130]	√	√	83.71	73.04	59.16	-	-	-	-	-	-	10.0
	MV3D[131]	√	√	74.97	63.63	54.00	-	-	-	-	-	-	2.8
	SCANet[132]	√	√	16.7	79.22	67.13	60.65	-	-	-	-	-	11.1
	ContFuse[133]	√	√	83.68	68.78	61.67	-	-	-	-	-	-	16.7
	RT3D[134]	√	√	23.74	19.14	18.86	-	-	-	-	-	-	11.1
	MMF[135]	√	√	77.43	70.22	12.5	-	-	-	-	-	-	88.40
	SIFRNet[136]	√	√	-	-	-	-	-	-	-	-	-	-
	Fast Point R-CNN[137]	√	×	84.80	74.59	67.27	-	-	-	-	-	-	16.7
Single Shot Methods	VoxelNet[138]	√	×	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37	2.0
	SECOND[139]	√	×	83.34	72.55	65.82	48.96	38.78	34.91	71.33	52.08	45.83	26.3
	Vote3Deep[140]	√	×	-	-	-	-	-	-	-	-	-	-
	3D FCN[141]	√	×	-	-	-	-	-	-	-	-	-	0.2
	3DBN[142]	√	×	83.77	73.53	66.23		-	-	-	-	-	7.7
	PointPillars[143]	√	×	82.58	74.31	68.99	51.45	41.92	38.89	77.10	58.65	51.92 62.0	
	VeloFCN[144]	√	×	-	-	-	-	-	-	-	-	-	1.0
	SA-SSD[145]	√	×	88.75	79.79	74.16	-	-	-	-	-	-	25.0
	MVX-Net[146]	√	√	84.99	71.95	64.88	-	-	-	-	-	-	16.7
	3DSSD[147]	√	×	88.36	79.57	74.55	54.64	44.27	40.23	82.48	64.10	56.90	25.0
Others	OHS-Dense[148]	√	×	88.12	78.34	73.49	47.14	39.72	37.25	79.09	62.72	56.76	33.3
	LaserNet[149]	√	×	-	-	-	-	-	-	-	-	- 83.3	
	Point-GNN[150]	√	×	88.33	79.47	72.29	51.92	43.77	40.14	78.60	63.48	57.08	1.7
	OHS-Direct[148]	√	×	86.40	77.74	72.97	51.29	44.81	41.13	77.70	63.16	57.16	33.3
	LaserNet++[151]	√	√	-	-	-	-	-	-	-	-	-	-26.3

III. AUTONOMOUS DRIVING: IMPLICATION AND PRESENT STATE

The world is intensively focusing on the emerging technology of autonomous driving to counter transportation issues in urban areas like road accidents, traffic congestion, parking space, and redundancy issues [152], [178]. *AV* began in the early $1980s$, primarily in the United States and Europe, leading to increasing advances in driving competence in a variety of situations [166], [174], [175], [176], [177]. Historically, we see tremendous efforts being made to achieve the desired goals of autonomous driving. *DARPA's* big urban challenge in $2007-2009$, *Google's*research project, and made the first *Waymo* autonomous vehicles, which capitalizes on their initial success. As a result, deep neural networks and computer vision are undergoing a revolution. As a result, the deep neural network and computer vision revolution led many people to believe that many of the technical obstacles to self-driving could be overcome in some solutions, while academia, the auto industry, and other high-tech companies are also working hard on autonomous technologies, as illustrated in Figure 4,5,6.

So far, progress toward autonomous vehicle goals remains elusive. The system consists of a series of units and complex interior/exterior dependencies in an autonomous vehicle. The complete auto drive is still far away due to technical bottlenecks and long-tail problems, [167]. Although *3D* image-based depth estimation and *3D*reconstruction techniques have greatly improved with the development of computer vision algorithms based on deep learning, the resulting estimates are still not always accurate. It's not just computational limitations that are a problem, but also poor light perception and bad weather are also big issues.

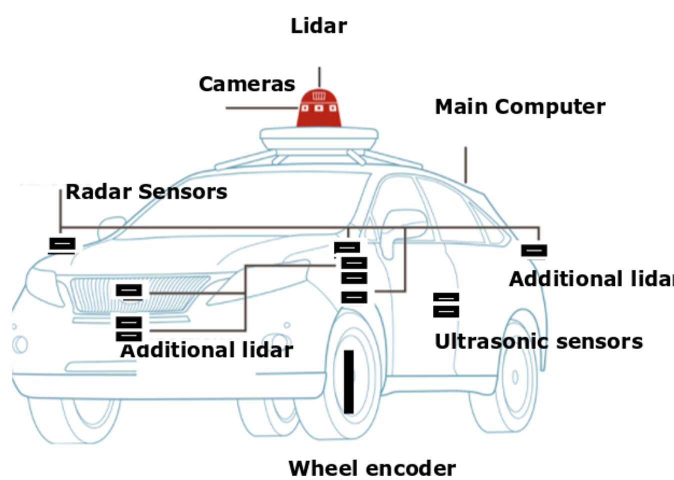


Fig. 6. Autonomous vehicle

1) Conceptual Composition of AV *system:* The *AV* system generally consists of various types of sensors as well as high definition maps, localization (required for geo-positioning), perception (required for navigation), prediction, orientation, tracking maps, object identification, and detection, and control

TABLE VIII

KITTI *3D* TEST *BEV* STANDARDS COMPARATIVE OUTCOMES FOR *3D* OBJECT. $Best_1$, $avearge_1$ AND $worst_1$ REPRESENT WITH 0.5 THRESHOLD VALUE FOR CAR, $Best_2$, $Best_3$ $avearge_2$, $avearge_3$ AND $worst_2$, $worst_3$ REPRESENT FOR PEDESTRIANS AND CYCLISTS WITH 0.5 CLASS OF OBJECT, $(-)$ OR $(\times)$ REPRESENT NOT APPLICABLE OR RESULT ARE UNKNOWN WHILE $\surd$ REPRESENT RESULT ARE KNOWN.

		LiDar	Image	$Best_1$	$Average_1$	$Worst_1$	$Best_2$	$Average_2$	$Worst_2$	$Best_3$	$Average_3$	$Worst_3$	Speed(fbs)
Region Proposal-based Methods	PointRCNN [116]	√	×	92.13	87.39	82.72	54.77	46.13	42.84	82.56	67.24	60.28	10.0
	PointPainting [117]	√	√	92.45	88.11	83.36	58.70	49.93	46.29	83.91	71.54	62.97	2.5
	STD [118]	√	×	94.74	89.19	86.42	60.02	48.72	44.55	81.36	67.23	59.35	12.5
	PointRGCN [119]	√	×	91.63	87.49	80.73	-	-	-	-	-	-	3.8
	IPOD [120]	√	√	89.64	84.62	79.96	60.88	49.79	45.43	78.19	59.40	51.38	5.0
	AVOD[121]	√	√	89.75	84.95	78.32	42.58	33.57	30.14	64.11	48.15	42.37	12.5
	PointFusion[122]	√	√	-	-	-	-	-	-	-	-	-	-
	Patch Refinement[123]	√	×	92.72	88.39	83.19	-	-	-	-	-	-	6.7
	PV-RCNN[124]	√	×	94.98	90.65	86.14	-	-	-	82.49	68.89	62.41	12.5
	VoteNet[125]	√	×	-	-	-	-	-	-	-	-	-	-
	3D IoU loss[126]	√	×	91.36	86.22	81.20	-	-	-	-	-	-	12.5
	F-ConvNet[127]	√	√	91.51	85.84	76.11	57.04	48.96	44.33	84.16	68.88	60.05	2.1
	ImVoteNet[128]	√	×	-	-	-	-	-	-	-	-	-	-
	F-PointNets[129]	√	√	91.17	84.67	74.77	57.13	49.57	45.48	77.26	61.37	53.78	5.9
	RoarNet[130]	√	√	88.20	79.41	70.02	-	-	-	-	-	-	10.0
	MV3D[131]	√	√	86.62	78.93	69.80	-	-	-	-	-	-	2.8
	SCANet[132]	√	√	90.33	82.85	76.06	-	-	-	-	-	11.1	
	ContFuse[133]	√	√	94.07	85.35	75.88	-	-	-	-	-	-	16.7
	RT3D[134]	√	√	56.44	44.00	42.34	-	-	-	-	-	-	11.1
	MMF[135]	√	√	93.67	88.21	81.99	-	-	-	-	-	-	12.5
	SIFRNet[136]	√	√	-	-	-	-	-	-	-	-	-	-
	Fast Point R-CNN[137]	√	×	90.76	85.61	79.99	-	-	-	-	-	-	16.7
Single Shot Methods	VoxelNet[138]	√	×	89.35	79.26	77.39	46.13	40.74	38.11	66.70	54.76	50.55	2.0
	SECOND[139]	√	×	89.39	83.77	78.59	55.99	45.02	40.93	76.50	56.05	49.45	26.3
	Vote3Deep[140]	√	×	-	-	-	-	-	-	-	-	-	-
	3D FCN[141]	√	×	70.62	61.67	55.61	-	-	-	-	-	-	0.2
	3DBN[142]	√	×	89.66	83.94	76.50	-	-	-	-	-	-	7.7
	PointPillars[143]	√	×	90.07	86.56	82.81	57.60	48.64	45.78	79.90	62.73	55.58	62.0
	VeloFCN[144]	√	×	0.02	0.14	0.21	-	-	-	-	-	-	1.0
	SA-SSD[145]	√	×	95.03	91.03	85.96	-	-	-	-	-	-	25.0
	MVX-Net[146]	√	√	92.13	86.05	78.68	-	-	-	-	-	-	16.7
	3DSSD[147]	√	×	92.66	89.02	85.86	60.54	49.94	45.73	85.04	67.62	61.14	25.0
Others	OHS-Dense[148]	√	×	93.73	88.11	84.98	50.87	44.59	42.14	82.13	66.86	60.86	33.3
	LaserNet[149]	√	×	79.19	74.52	68.45	-	-	-	-	-	- 83.3	
	Point-GNN[150]	√	×	93.11	89.17	83.90	55.36	47.07	44.61	81.17	67.28	59.67	1.7
	OHS-Direct[148]	√	×	93.59	87.95	83.21	55.90	49.48	45.79	79.66	67.20	61.04	33.3
	LaserNet++ [151]	√	√	-	-	-	-	-	-	-	-	-	26.3

units [168]. In the first step, a high-resolution offline map and its surroundings are developed without the internet; then, the online system receives a destination for a specific location. *LiDAR* technology was first activated in 1960 for light and range detection and remote sensing to measure the exact distance of an object on the surface of the Earth. In addition to that, the Global Positioning System *(GPS)* was introduced in 1980, and it later became a popular method for computing accurate geospatial measurements. *LiDAR* technology, each *3D* point indexes the range from *LiDAR* to the outer surface of an object and turns them into accurate *3D* coordinates. As shown in Figure 7, *3DPC* is extremely useful to the autonomous vehicle for locating and detecting surrounding objects in the *3D* world.

2) Functioning Methodology of AV *system:* The *AV* system localizes itself to the map, senses its environment and perceives the world around it, and calculates corresponding potential trajectories for the future motion of these objects. The *AV* system uses motion sensors and predictions to plan a safe trajectory to follow the high-level route from initial to end as executed by the controller. In the *AV* system, two approaches to *3DPC* are used. For instance, *PC* map is generated through a map-generated localization unit. In addition, a real-time *LiDAR* sweep developed by localization and perception units

3) 3DPC *processing and learning:* In *AVS* processing and learning techniques convert raw measurements into useful information, and *LiDAR* provides essential *3D* data for *AV*. The

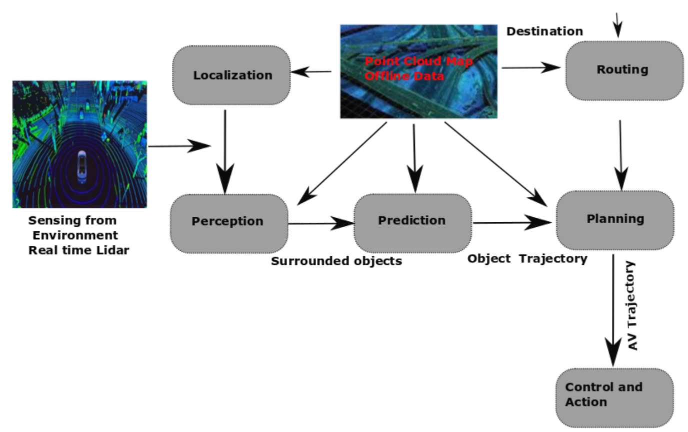


Fig. 7. A typical autonomous vehicle system *(AVS)* schematic includes a high-resolution offline map. When the object is in motion, the online system gets the target parameters. First, the system takes input data related to its surroundings, determines its location on a map, marks the surroundings around the object, and then makes similar predictions for the next move. The perceived predictions are then sent to the motion planner to organize a safe route for the *(AVS)*. The controller follows the specified movement according to the specified path to the target destination. In the standalone system, two types of *3DPC* functions are used: a mapping unit linked to a *GPS* unit and *LiDAR* for real-time scanning. The assembly process is then carried out by the location and perception unit together with the detection unit.

3DPC index is used to locate *AV* and the *PCM* perception units are used to distinguish between foreground and background. The *PCM* provides information on the environment. The locator unit uses a *PCM* as a reference in the *PC* log to determine the position of the *AV* system. The advancement ofcommunication networks and acoustic sensors based on *RADAR* has seen rapid growth over the past century, which has taken a revolutionary consequence on digital communication networks.

The proliferation of cameras and televisions into *2D* image processing has increased over the past 30 years, taking photography, entertainment, and surveillance to the next level of scientific usage. At the same time, *3DPC* processing and learning algorithms are gaining more attention in academia. Cooperation of academic researchers and the auto industry would make rapid progress in achieving the desired *AVS* goals, but would have a prosperous socio-economic and environmental impact.

4) 3DPC*: Properties and Characteristic:* There are two typical types of *3DPC* work in *AVS*, real-time *LiDAR* spans and *PC* mapping. *LiDAR* data is displayed in real-time as a *2D* image with *x-axis* and *y-axis* timestamp records. Each combined *3DPC* which is associated with *GPS* timestamp, band scale, and intensity value. However, for real-time *LiDAR* images are together at diverse timestamps. In some cases, the combined *3DPC* is not completely aligned in a normal *2D* mesh.

- *PC* Mapping: The *PC* mapping adds several *LiDAR* scans from distinct views, which are similar to the generated *3DPC* data through different sensing units. A *PC* mapping captures data about the surfaces of objects, providing a more intense and detailed *3D* representation. Irregularity, as it comes from multiple *LiDAR* scans and loses laser identification, causes disordered *3DPC*.
- Dense *PC*: Dense point clouds collect the information about the *3D* object that contains full detail, while *3DPC* includes semantic labels for the *3D* scene to improve precision.
- Conventional and Non-Conventional: There have been several conventional approaches to dealing with *3DPC* for different tasks, besides deep learning tools to manipulate *3DPC* and applying a convolutional neural network to analyze images. The input data is used to produce the filter's activation map to set a learnable filter by cover layer. The benefit of using *CNN* to interact with a *3DPC* is that it includes local spatial relationships. *CNN* will invariably replace accurate *3D* position information, but it still provides reliable and promising experimental results [169], [170], [171].
- PointNet: PointNet-based methods manipulate the primary *3DPC* by applying deep neural networks to generate the equivalent outcomes regardless of the order of the input data, as well as surprisingly influential performance in *3DPC* recognition and segmentation.
- Graph-based methods: The impulse to use the chart Based methods make use of the spatial relationships between *3D* dots to accelerate all-around learning of deep neurotechnology networks. A graph-based operation is usually a schematic filter that moves the classic filter to the histogram space and takes properties from the graph signal.

5) 3DPC *for higher definition map:* The *HD* Autonomous Driving Map *(ADM)* is an accurate and heterogeneous representation of the static *3DPC* surrounding map. The foremost intention for generating a high-resolution offline map is to understand the traffic rules and surroundings in real-time, which is a considerable challenge by adopting the *AV* system. A high-resolution map provides robust and multi-unit designs in the autonomous system, as displayed in Figures 8,9,10,13.

- $HD-Map$ localization: The role of the *HD*-Map localization is to locate the *AV* on a high-resolution map, *PC* map, and semantic features related to traffic rules, such as lane markers and posts, which generally function as pre-map locations.
- Perception: The role of perception is to pre-reveal and identify all elements of the landscape and their inner points. The *PC* map and real-time *LiDAR* scanning process are separated into real-time foreground and background points. These measures can considerably increase the recognition accuracy and reduce the computational cost.
- Prediction: Prediction is used to figure out how the landscape will look in the future for each individual unit. For example, it can use it to direct the projected paths of units to follow traffic paths.
- Planning of Actions: The measure of scheduling and action is to conclude the track of *AV* system. In a high-resolution map, semantic characteristics related to traffic rules, such as lane geometry, junctions, traffic lights, traffic signals, and speed limits for lanes, are indispensable preconceptions of a module.
- *3DPC* stitching: The aim of *3DPC* is to generate high-resolution *PC* images from sensor information generated by *AV* systems. The mapping module includes *3DPC* mapping and semantic feature extraction; furthermore, the *PC* map leads the precision of all map inputs. Optimal accuracy is mandatory at all local units in the *PC* map, which instantly generates and updates high-resolution citywide maps for *AV* system.
- Local and global accuracy: Local accuracy refers to the precision of the *LiDAR* location in the corresponding local area. At the same time, the global resolution shows all *LiDAR* modes on the entire *HD* map and is precise concerning the standard world framework.
- Feature Extraction: The limitations of training and complex traffic conditions make it difficult to extract complex semantic properties, such as traffic sign control information and road lane information, which still rely heavily on human supervision and are certainly expensive and time-consuming.

6) 3DPC *processing for localization:* A comprehensive *AV* system, high sensitivity, and durability are the most important building blocks for transforming performance constraints. Endurance specifies that the *GPS* devices must function in all driving situations, including changes in lighting, weather,

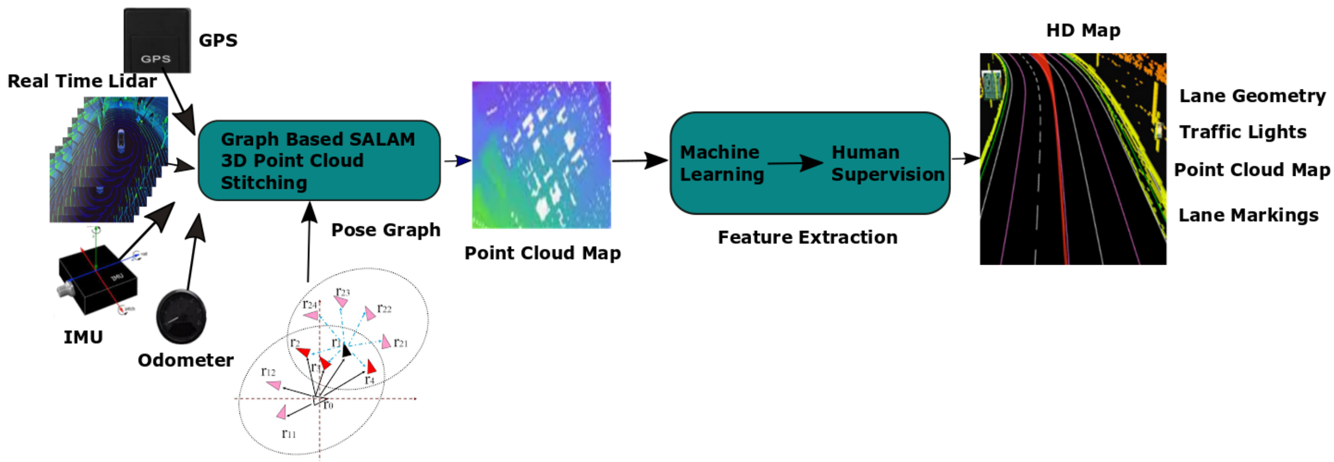


Fig. 8. A standard *HD* map system comprises two main components: semantic feature extraction and *3DPC* stitching. *3DPC* is generally based on graphics-based *SLAM* along with a hierarchical optimization system; the other feature, the semantic feature extraction system, comprises iterative actions of human supervision and machine learning. An essential component of graphics-based *SLAM*, which formulates relationships between *LiDAR* modes which reflects the level of misalignment between the two *LiDAR* modes. The outcomes include a *PC* map, which is referred to as a dense *3DPC* and a semantic feature map associated with traffic rules, which contains the positions of ground signs, road signs, and road signs of traffic.

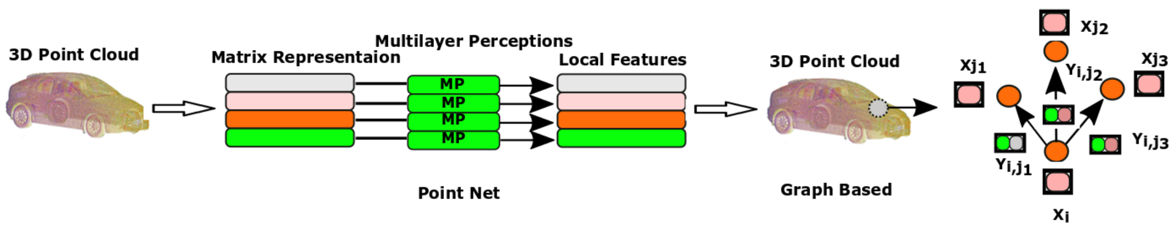


Fig. 9. (a) The PointNet extracts geometric features with the static permutation property of raw *3DPC* using a set of multilayer perceptions *(MLPs)* followed by maximum aggregation.(B) shows that graph-based methods provide a graphical structure to capture local relationships between three-dimensional points. Each node is a three-dimensional point in the graph, and each edge reflects the relationship between each pair of three-dimensional points.

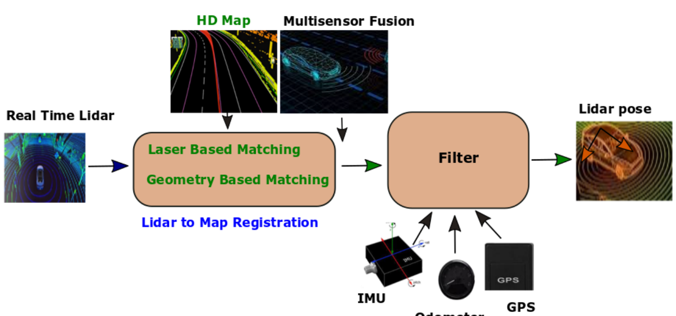


Fig. 10. The standard map-based location system is made up of two major components: *LiDAR* for log mapping and multi-sensor integration. *LiDAR* systems use record geometry-based mapping and laser inversion-based matching for high precision and searching. Multi-sensor fusion uses a Bayesian filter to combine different methods.

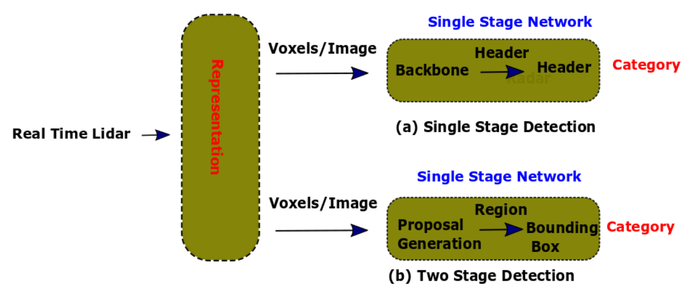


Fig. 11. Single-step detection and two-stage detection frames figure out the bounding box right away, while two-stage detection first suggests a large area that might have objects and then works out the bounding box.

traffic, and road conditions. Currently, deep neural networks are being used to build robust feature matching and visual localization under harsh conditions. Wang proposed high-accuracy visual localization that leverages a preceding *LiDAR* point cloud to restrict visual location. However, the new local feature extraction approaches must go through a time-consuming deep neural network computing process. Terrestrial or static *LiDAR* sensors can make dense point clouds in a single frame, but they can't do the same in larger frames (like mechanical rotary *LiDAR*.

- Map-based localization: Localization can be divided into two categories: optimization and Bayesian fil-

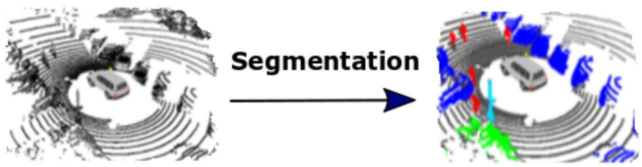


Fig. 13. *3DPC* segmentation intentions to categorize respective point in *3DPC* are a distinct class

tering approaches. Based on current sensor measurements, optimization-based algorithms frequently outperform Bayesian-filtering methods in terms of estimation accuracy. Since it reduces the cost function, it is vulnerable to some extent of noise. Such Bayesian filtering techniques are impervious to such noises and can estimate a smooth trajectory[173]. The key indicator of map-based localization is the evaluation of *LiDAR* placement by corresponding *LiDAR* survey with *PC-HD* map while generating dimensions from *IMU*, *GPS*, and cameras to generate a robust position estimation. A map-based location system generally comprises of two modules: the *LiDAR* to map recording, which calculates the *LiDAR* mode by recording the *LiDAR* scan of the *PC* map.

A map-based location system typically includes *LiDAR*, which computes the *LiDAR* mode by recording the *LiDAR* scan on the PC map. At the same time, the second module is the integration of multiple sensors, which estimates the final position of the *IMU* and the *GPS*, in addition to the approximation of *LiDAR* recording to the map.

- Geometry-based approach computes high-resolution. Operating *AV* and autonomous systems in unfavourable weather conditions such as rain, snow, and fog is currently a major challenge. In severe weather, human eyes are impaired, making a driver assistance system even more essential. *LiDAR* sensors have recently been presented as an important part of a high-performance perception device for better driver assistance features. A *LiDAR* scan with a *PC* map based on an *ICP* algorithm [172] generally performs well in heavy traffic flow and harsh weather conditions, as well as engineering scenes such as tunnels, bridges, roads, etc.
- Multi-sensor synthesis: The multi-sensor fusion component is used to estimate the robust and safe location from the dimensions of various sensors comprising cameras and *GPS*, in addition to evaluating the *LiDAR*-to-map recording module.
- Real-world challenges: Bring adaptation drive to extreme scenes as well as the *AV* system as a straight channel without a broken lane marking, there are few geometric and textual features that fail to register *LiDAR* to the map. Furthermore, when the *AV* is surrounded by large trucks, the *LiDAR* can be blocked entirely, which can also cause the *LiDAR* to map process failure. The *LiDAR* recording failure on the map continues for a few minutes. The *LiDAR* mode is predictable by the multi-sensor fusion units that will be severely skewered, and the positioning device will lose precision.

7) 3DPC *processing for perception:* An overview of the perception module, the general description of the unit of perception is an optical unit of an *AV* system that allows the perception of the neighboring in *3D* environment. The output from the perception units is generally assessed from the *LiDAR*, cameras, ultrasound, and *RADAR*, as well as the output from the ego movement mode in the positioning unit, the outcomes of the perception unit of traffic lights, and *3D* boxes for objects with tracks shown in Table VII, VIII.

- Detection of *3D* objects: The operation of *3D* object detection is to perceive and locate objects in *3D* space through the representation of surrounding squares based on a single measurement by various sensors. The detection of a *3D* object usually results in a bounding box for the *3D* object, which means the association of object and tracking components; In addition, we can use the sensor dimension to classify the detection of *3D* objects through detection based on *LiDAR*, shown in Figure 11.
- *LiDAR*-based object detection: *LiDAR*-based detection task is usually implemented using architectures based on deep neural networks. The key transformation between the detection of *3D* and *3D* objects lies in the input data representation; compared to *3D* images, real-time *LiDAR* scanning is done in several ways.
- Fusion-based object detection: Real-time *LiDAR* scanning provides exceptional *3D* representation of the scene. However, sparse sizes often return instantaneous locations and intricacy. A *LiDAR*-based recognition process for estimating object speed and recognizing small objects (e.g. pedestrians). *RADAR* offers real-time motion information, while *3D* images offer dense dimensions to improve overall reliability, as shown in Figure 12.

IV. Real world challenges and discussion

The self-driving industry is overgrowing. Many technologies are relatively mature; conversely, the final solution for *AV* system has yet to be found. Furthermore, advanced *3DPC* learning and processing technologies are critical components of *AV* driving. In this article, we looked at the latest developments in *3DPC* processing and learning and presented applications for *AV* driving. We elucidate in what way *3DPC* processing and learning play a role in three critical units of *AV* driving: mapping, perception, and localization.

The rapid expansion of *3DPC* learning and processing and the inclusive performance of mapping, perception, and localization units in *AV* system have improved significantly. However, there are still many challenges ahead of us. Here we will concisely highlight some of the major unsolved problems. In both projection-based and discretization-based approaches, *3D* image representations can benefit from a well-established network architecture. Conversely, the fundamental constraint of projection-based approaches is the loss of data affected by *3D-2D* projection; however, the key barrier for discretization-based approaches is higher computational and memory overheads induced by the upsurge in resolution. It is possible to achieve this goal using sparse convolution based

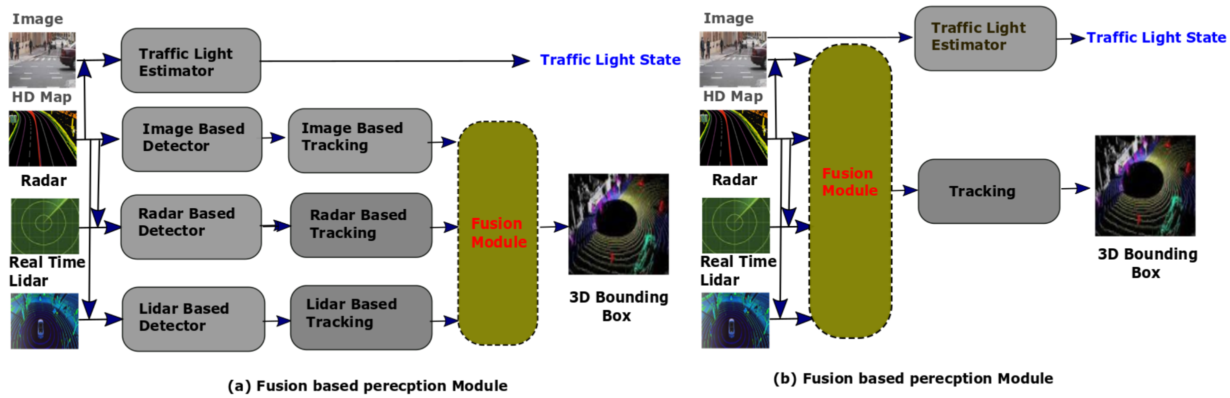


Fig. 12. The perception module takes multiple detection modes and generates clues by generating a *3D* bounding box of traffic light states and objects. According to the fusion modality mechanism, perception modules are classified as late mergers because they merge in the semantic space and combine in the functional space.

on indexing structures. There are the most common techniques of investigation; most existing point-based approaches rely on expensive contiguous-searching procedures since point representation naturally lacks explicit nearby information (e.g., *KNN* [104], ball query [155]. It is for this reason that point-voxel mutual representation, which was recently proposed, might be a promising new research avenue.

Meyer et al. proposed LaserNet [149], a fast *3D* object detector to build precise *3D* object boxes and estimate a probability distribution across bounding boxes for each point. LaserNet outperforms prior approaches from 0 to 50 meters and has a shorter runtime. Meyer [151] enhanced LaseerNet [149] to leverage *RGB* images' rich texture (e.g., 50 to 70 meters). They linked *LiDAR* points to image pixels by projecting *3DPC* onto *2D* images, then combining *RGB* data into *3D* points. To figure out how to make better representations, they examined *3D* semantic segmentation and looked at how to make both long-range ($50 - 70$ meter) object identification and semantic segmentation more efficient while keeping LaserNet's high efficiency.

MLP-Pointwise networks are commonly used as the foundation for other kinds of networks to acquire knowledge of pointwise functions. In terms of performance on irregular *3DPC*, convolution- based networks are a usual *DL* architecture. In the case of irregular data, continuous and discrete convolution networks should get more attention. Graph-based networks have attracted a lot of interest recently because of their inherent strength in dealing with irregular data. But, extending graph-based spectral networks to different graph architectures is still a challenge.

- In what way do we make learning and processing algorithms scalable and efficient? We are still in the development stage and will test *AV* on an insufficient number of standard roads. In the near forthcoming, self-driving cars can confirm on an urban/rural scale, which will require high resolution urban/rural scale *HD* maps. Today, self-driving cars are often equipped with 64 *LiDAR* lines, which still generate relatively trivial *3DPC*; furthermore, *LiDAR* could contain more ribbons and create denser *3DPC*. *AV* system needs additional proficient algorithms to locate *LiDAR* map in real-time and recognize *3D* targets.
- By what means do we make learning and processing algorithms influential enough to cope with extreme situations? We assemble huge volumes of real-time sensor information and generate simulated sensor information. However, it's essential to consciously choose the best representative statistics to improve the versatility of the algorithm. Simultaneously, we have to recognize all objects based on training algorithms and data that cannot cover all prospects. The main area of research is to expand the algorithm's improbability estimate; consequently, the system reacts cautiously when the component precision is uncertain. So, this means that we need to look at both known uncertainties from the training information and more complicated uncertainties that aren't covered by the training information.
- How do we progress faster with iterative learning and processing algorithms? We need more statistics and complex algorithms to attain enhanced *AV* driving performance. We need efficient and real-world algorithms to speed up the development of new products; industry practitioners and academic researchers should work together to increase the proportion of research that is used in the real world.
- In what way should processing and learning algorithms be evaluated? Most of the learning and processing algorithms for specific metrics are assessed at the model level to meet the benchmarks of the task; conversely, these model-level indicators are often not fully correlated with system-level indicators that reflect general behavior. The research community frequently focuses on average, optimizing performance; however, it should pay more attention to those rare cases where optimization is critical for real-world systems.
- The evolution of deep learning, the unit of perception,

has improved tremendously. However, the component of perception is far from perfect. There are several units of perception challenges that remain costly. An autonomous vehicle is generally equipped with one or more *LiDAR* and computers, such as *GPU* and other expensive generalized treatments.

- The compromise between effectiveness and efficiency is that the *AV* must interact with its environment in real-time. It would not make sense to go for high-fidelity perception when the drive provides so much latency.
- Training data deluge: the modern display unit relies heavily on machine learning approaches that generally require as much training information as possible.
  However, it requires a lot of processing and computational resources, and yet there are countless driving circumstances because large-scale training information cannot cover every possible situation.
  Finding and managing the angular state is still an unsolved problem, especially when it comes to detecting objects that never appear in the training information.
- Standardization experimental setup: there is no standard for *3DPC* sampling. Researchers create training and test data sets based on the $Vision-Air$, $ShapeNet$, $ModelNet40$, and $SHREC15$ repositories. The extensive outdoor *3DPC* technology evaluates and records the performance by evaluating the difference in position and orientation between the location and the calculated based data.
- There is an increasing demand for *3DPC* in the different application fields, as well as robots, *AV* systems, virtual and augmented reality, infrastructure scanning, animation, and monument preservation. *3DPC* learning and processing relatively extend most *1D* signal processing, *2D* machine learning, image processing, and computer vision tasks into a *3D* space. The following are the significant obstacles causing hurdles in the development of a fully autonomous and reliable *AV* system:

V. Conclusions

The recent evolution of *3D* Point Cloud learning and processing has significantly improved the overall performance of localization, mapping creation, perception, and recognition modules in AV systems. How could we construct more scalable and more proficient learning and processing algorithms? As we already know that, we are still in the emerging stage and *AV* system, which has tried over some degree of established routes. It's likely that in the future, a lot of *AVs* will be tested on a national level, and they'll need the wide-range *HD* map'. To achieve this, it needs a scalable approach to produce *HD-map* at runtime. We need to develop the processing and learning algorithms at their maximum iterative speeds. *3DPC* data and the complex algorithms would enable the performance of autonomous driving. The auto industry's owners are working together with the researchers to upsurge the research transformation rate. Still, they need to emphasize more on improving long-tail rare cases, which are essential to the existing system.

Acknowledgment

Competing interests

The authors declare that they have no competing interests.

References

[1] Lou K, Yang Y, Wang E, et al. Reinforcement Learning Based Advertising Strategy Using Crowdsensing Vehicular Data[j]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(7): 4635-4647.

[2] Raja G, Anbalagan S, Subramaniyan AG, et al. Efficient and Secured Swarm Pattern Multi-UAV Communication[j]. IEEE Transactions on Vehicular Technology, 2021, 70(7):7050-7058.

[3] Raja G, Anbalagan S, Vijayaraghavan G, et al. Energy-Efficient End-to-End Security for Software Defined Vehicular Networks[j]. IEEE Transactions on Industrial Informatics. 2020.

[4] Raja G, Anbalagan S, Vijayaraghavan G, et al. 2020. Energy-Efficient End-to-End Security for Software-Defined Vehicular Networks[j]. IEEE Transactions on Industrial Informatics, 2020, 17(8): 730-5737.

[5] Asghar I, James C, William W, Oche Alexander Egaji, Mark Griffiths, and Shelly Barratt. A Smart Transportation Management System for Managing Travel EventsC. In Proceedings of the 2020 10th International Conference on Information Communication and Management, pp. 61-65. 2020.

[6] Juma M, and Khaled S. Cyberphysical systems in the smart city: challenges and future trends for strategic research[J]. In Swarm Intelligence for Resource Management in Internet of Things, pp. 65-85. Academic Press, 2020.

[7] Javed M, Meraz M D, Chakraborty P. A Quick Review on Recent Trends in 3D Point Cloud Data Compression Techniques and the Challenges of Direct Processing in 3D Compressed Domain[J]. arXiv preprint arXiv:2007.05038, 2020.

[8] Javed M, Nagabhushan P, Chaudhuri B B. A review on document image analysis techniques directly in the compressed domain[J]. Artificial Intelligence Review, 2018, 50(4): 539-568.

[9] Mei J, Gao B, Xu D, et al. Semantic segmentation of 3D LiDAR data in dynamic scene using semi-supervised learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(6): 2496-2509.

[10] Zhong Y, Zhang H, Jain A K. Automatic caption localization in compressed video[J]. IEEE transactions on pattern analysis and machine intelligence, 2000, 22(4): 385-392.

[11] Chen S, Liu B, Feng C, et al. 3d point cloud processing and learning for autonomous driving[J]. arXiv preprint arXiv:2003.00601, 2020.

[12] Zeadally, S, Hunt, R, Chen, Irwin, A, Hassan, A. Vehicular ad hoc networks (VANETS): Status, results, and challenges[j]. Telecommunication Systems, 2012. 50(4), 217-241.

[13] Cao C, Preda M, Zaharia T. 3D point cloud compression: A survey[C]//The 24th International Conference on 3D Web Technology. 2019, 1-9.

[14] Guo Y, Wang H, Hu Q, et al. Deep learning for 3d point clouds: A survey[J]. IEEE transactions on pattern analysis and machine intelligence, 2020.

[15] Abbasi R, Chen J, Al-Otaibi Y, et al. RDH-based dynamic weighted histogram equalization using for secure transmission and cancer prediction[J]. Multimedia Systems, 2021, 27(2): 177-189.

[16] Waschbusch M, Gross M H, Eberhard F, et al. Progressive Compression of Point-Sampled Models[C]//PBG. 2004: 95-102.

[17] Merkle P, Smolic A, Muller K, et al. Multi-view video plus depth representation and coding[C]//2007 IEEE International Conference on Image Processing. IEEE, 2007, 1: I-201-I-204.

[18] Mekuria R, Blom K, Cesar P. Design, implementation, and evaluation of a point cloud codec for tele-immersive video[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 27(4): 828-842.

[19] Schwarz S, Preda M, Baroncini V, et al. Emerging MPEG standards for point cloud compression[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2018, 9(1): 133-148.

[20] Abbasi R, Luo B, Rehman G, et al. A new multilevel reversible bit-planes data hiding technique based on histogram shifting of efficient compressed domain[J]. Vietnam Journal of Computer Science, 2018, 5(2): 185-196.

[21] Abbasi R, Xu L, Amin F, et al. Efficient lossless compression based reversible data hiding using multilayered n-bit localization[J]. Security and Communication Networks, 2019, 2019.

[22] Abbasi R, Faseeh Qureshi N M, Hassan H, et al. Generalized PVO-based dynamic block reversible data hiding for secure transmission using firefly algorithm[J]. Transactions on Emerging Telecommunications Technologies, 2019: e3680.

[23] Golla T, Klein R. Real-time point cloud compression[C]//2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015: 5087-5092.

[24] Gandoin P M, Devillers O. Progressive lossless compression of arbitrary simplicial complexes[J]. ACM Transactions on Graphics (TOG), 2002, 21(3): 372-379.

[25] Gumhold S, Kami Z, Isenburg M, et al. Predictive point-cloud compression[M]//ACM SIGGRAPH 2005 Sketches. 2005: 137-es.

[26] Waschbusch M, Gross M H, Eberhard F, et al. Progressive Compression of Point-Sampled Models[C]//PBG. 2004: 95-102.

[27] Merry B, Marais P, Gain J. Compression of dense and regular point clouds[C]//Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa. 2006: 15-20. https://doi.org/10.1145/1108590.1108593

[28] Ochotta T, Saupe D. Compression of point-based 3D models by shape-adaptive wavelet coding of multi-height fields[M]. 2004.

[29] Lien J M, Kurillo G, Bajcsy R. Multi-camera tele-immersion system with real-time model driven data compression[J]. The Visual Computer, 2010, 26(1): 3-15.

[30] Ismael D, Ryo F, Ryusuke S,et al. and Naoki Asada. 2012. Efficient rate-distortion compression of dynamic point cloud for grid-pattern-based 3D scanning systems[j]. 3D Research, 3(1) 2012.

[31] Zhang C, Florencio D, Loop C. Point cloud attribute compression with graph transform[C]//2014 IEEE International Conference on Image Processing (ICIP). IEEE, 2014: 2066-2070.

[32] Golla T, Klein R. Real-time point cloud compression[C]//2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015: 5087-5092.

[33] Thanou D, Chou P A, Frossard P. Graph-based motion estimation and compensation for dynamic 3D point cloud compression[C]//2015 IEEE International Conference on Image Processing (ICIP). IEEE, 2015: 3235-3239.

[34] De Queiroz R L, Chou P A. Compression of 3D point clouds using a region-adaptive hierarchical transform[J]. IEEE Transactions on Image Processing, 2016, 25(8): 3947-3956.

[35] de Queiroz R L, Chou P A. Motion-compensated compression of dynamic voxelized point clouds[J]. IEEE Transactions on Image Processing, 2017, 26(8): 3886-3895.

[36] de Queiroz R L, Chou P A. Transform coding for point clouds using a Gaussian process model[J]. IEEE Transactions on Image Processing, 2017, 26(7): 3507-3517.

[37] Zhang K, Zhu W, Xu Y. Hierarchical segmentation based point cloud attribute compression[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 3131-3135.

[38] Garcia D C, de Queiroz R L. Intra-frame context-based octree coding for point-cloud geometry[C]//2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018: 1807-1811.

[39] Huang T, Liu Y. 3d point cloud geometry compression on deep learning[C]//Proceedings of the 27th ACM International Conference on Multimedia. 2019: 890-898.

[40] Rottensteiner F, Sohn G, Jung J, et al. The ISPRS benchmark on urban object classification and 3D building reconstruction[J]. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1, 2012, 1(1): 293-298.

[41] Vallet B, Bredif M, Serna A, et al. TerraMobilita/iQmulus urban point cloud analysis benchmark[J]. Computers and Graphics, 2015, 49: 126-133.

[42] Dai A, Chang A X, Savva M, et al. Scannet: Richly-annotated 3d reconstructions of indoor scenes[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 5828-5839.

[43] Roynard X, Deschaud J E, Goulette F. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification[J]. The International Journal of Robotics Research, 2018, 37(6): 545-557.

[44] Behley J, Garbade M, Milioto A, et al. Semantickitti: A dataset for semantic scene understanding of lidar sequences[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 9297-9307.

[45] Tan W, Qin N, Ma L, et al. Toronto-3D: A large-scale mobile lidar dataset for semantic segmentation of urban roadways[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020: 202-203.

[46] Varney N, Asari V K, Graehling Q. DALES: a large-scale aerial LiDAR data set for semantic segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020: 186-187.

[47] Patil A, Malla S, Gang H, et al. The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes[C]//2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 9552-9557.

[48] hang M FC, Lambert J, Sangkloy P, et al. Argoverse: 3D tracking and forecasting with rich maps[c] //2019 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2019.

[49] Houston J, Zuidhof G, Bergamini L, et al. One thousand and one hours: Self-driving motion prediction dataset[J]. arXiv preprint arXiv:2006.14480, 2020.

[50] Pham Q H, Sevestre P, Pahwa R S, et al. A 3D dataset: Towards autonomous driving in challenging environments[C]//2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 2267-2273.

[51] Sun P, Kretzschmar H, Dotiwalla X, et al. Scalability in perception for autonomous driving: Waymo open dataset[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 2446-2454.

[52] Caesar H, Bankiti V, Lang A H, et al. nuscenes: A multimodal dataset for autonomous driving[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11621-11631.

[53] Song S, Lichtenberg S P, Xiao J. Sun rgb-d: A rgb-d scene understanding benchmark suite[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 567-576.

[54] Siddiqi K, Zhang J, Macrini D, et al. Retrieving articulated 3-D models using medial surfaces[J]. Machine vision and applications, 2008, 19(4): 261-275.

[55] Wu Z, Song S, Khosla A, et al. 3d shapenets: A deep representation for volumetric shapes[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1912-1920.

[56] Uy M A, Pham Q H, Hua B S, et al. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1588-1597.

[57] De Deuge M, Quadros A, Hung C, et al. Unsupervised feature learning for classification of outdoor 3d scans[C]//Australasian Conference on Robitics and Automation. 2013, 2: 1.

[58] Tu C, Takeuchi E, Carballo A, et al. Point cloud compression for 3D LiDAR sensor using recurrent neural network with residual blocks[C]//2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 3274-3280.

[59] Imdad U, Asif M, Ahmad M T, et al. Three dimensional point cloud compression and decompression using polynomials of degree one[J]. Symmetry, 2019, 11(2): 209.

[60] Cao C, Preda M, Zaharia T. 3D point cloud compression: A survey[C]//The 24th International Conference on 3D Web Technology. 2019: 1-9.

[61] Liu H, Yuan H, Liu Q, et al. A comprehensive study and comparison of core technologies for MPEG 3-D point cloud compression[J]. IEEE Transactions on Broadcasting, 2019, 66(3): 701-717.

[62] Xu R, Yun T, Cao L, et al. Compression and Recovery of 3D Broad-Leaved Tree Point Clouds Based on Compressed Sensing[J]. Forests, 2020, 11(3): 257.

[63] Joseph Rivlin M, Zvirin A, Kimmel R. Momen (e) t: Flavor the moments in learning to classify shapes[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019: 0-0.

[64] Zhao H, Jiang L, Fu C W, et al. Pointweb: Enhancing local neighborhood features for point cloud processing[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 5565-5573. 2019.

[65] Yan X, Zheng C, Li Z, et al. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 5589-5598.

[66] Duan Y, Zheng Y, Lu J, et al. Structural relational reasoning of point clouds[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 949-958.

[67] Yang J, Zhang Q, Ni B, et al. Modeling point clouds with self-attention and gumbel subset sampling[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 3323-3332.

[68] Zhang Y, Rabbat M. A graph-cnn for 3d point cloud classification[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 6279-6283.

[69] Hassani K, Haley M. Unsupervised multi-task feature learning on point clouds[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 8160-8171.

[70] Liu J, Ni B, Li C, et al. Dynamic points agglomeration for hierarchical point sets learning[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 7546-7555.

[71] Xu Q, Sun X, Wu C Y, et al. Grid-gcn for fast and scalable point cloud learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 5661-5670.

[72] Te G, Hu W, Zheng A, et al. Rgcnn: Regularized graph cnn for point cloud segmentation[C]//Proceedings of the 26th ACM international conference on Multimedia. 2018: 746-754.

[73] Wang C, Samari B, Siddiqi K. Local spectral graph convolution for point set feature learning[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 52-66.

[74] Wang Y, Sun Y, Liu Z, et al. Dynamic graph cnn for learning on point clouds[J]. Acm Transactions On Graphics (tog), 2019, 38(5): 1-12.

[75] Zhang K, Hao M, Wang J, et al. Linked dynamic graph cnn: Learning on point cloud via linking hierarchical features[J]. arXiv preprint arXiv:1904.10014, 2019.

[76] Chen C, Li G, Xu R, et al. Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 4994-5002.

[77] Shen Y, Feng C, Yang Y, et al. Mining point cloud local structures by kernel correlation and graph pooling[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4548-4557. 2018.

[78] Simonovsky M, Komodakis N. Dynamic edge-conditioned filters in convolutional neural networks on graphs[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 3693-3702.

[79] Pan G, Wang J, Ying R, et al. 3DTI-Net: Learn inner transform invariant 3D geometry features using dynamic GCN[J]. arXiv preprint arXiv:1812.06254, 2018.

[80] Zeng W, Gevers T. 3dcontextnet: Kd tree guided hierarchical learning of point clouds using local and global contextual cues[C]//Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 2018: 0-0.

[81] Xie S, Liu S, Chen Z, et al. Attentional shapecontextnet for point cloud recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4606-4615.

[82] Klokov R, Lempitsky V. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 863-872.

[83] Li J, Chen B M, Lee G H. So-net: Self-organizing network for point cloud analysis[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 9397-9406.

[84] Rao Y, Lu J, Zhou J. Spherical fractal convolutional neural networks for point cloud recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 452-460.

[85] Komarichev A, Zhong Z, Hua J. A-cnn: Annularly convolutional neural networks on point clouds[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 7421-7430.

[86] Mao J, Wang X, Li H. Interpolated convolutional networks for 3d point cloud understanding[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1578-1587.

[87] Lan S, Yu R, Yu G, et al. Modeling local geometric structure of 3d point clouds using geo-cnn[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 998-1008.

[88] Xu Y, Fan T, Xu M, et al. Spidercnn: Deep learning on point sets with parameterized convolutional filters[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 87-102.

[89] Li Y, Bu R, Sun M, et al. Pointcnn: Convolution on x-transformed points[J]. Advances in neural information processing systems, 2018, 31: 820-830.

[90] Hermosilla P, Ritschel T, Vquez P P, et al. Monte carlo convolution for learning on non-uniformly sampled point clouds[J]. ACM Transactions on Graphics (TOG), 2018, 37(6): 1-12.

[91] Atzmon M, Maron H, Lipman Y. Point convolutional neural networks by extension operators[J]. arXiv preprint arXiv:1803.10091, 2018.

[92] Boulch A. Generalizing Discrete Convolutions for Unstructured Point Clouds[C]//3DOR Eurographics. 2019: 71-78.

[93] Tu C, Takeuchi E, Carballo A, et al. Real-time streaming point cloud compression for 3d lidar sensor using u-net[J]. IEEE Access, 2019, 7: 113616-113625.

[94] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.

[95] [12] Huang L, Wang S, Wong K, Liu J, Urtasun R. Octsqueeze: Octree-structured entropy model for lidar compression[C]//2020 InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2020 (pp. 1313-1323).

[96] [13] Weinberger MJ, Seroussi G, Sapiro G. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS[J]. IEEE Transactions on Image processing. 2000 Aug;9(8):1309-24.

[97] Nardo F, Peressoni D, Testolina P, Giordani M, Zanella A. Point Cloud Compression for Efficient Data Broadcasting: A Performance Comparison[J]. arXiv preprint arXiv:2202.00719. 2022 Feb 1.

[98] Kim J, Rhee S, Kwon H, Kim K. LiDAR Point Cloud Compression by Vertically Placed Objects based on Global Motion Prediction. IEEE Access. 2022 Jan 31.

[99] Zaheer M, Kottur S, Ravanbakhsh S, et al. Deep sets[J]. arXiv preprint arXiv:1703.06114, 2017.

[100] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. arXiv preprint arXiv:1706.02413, 2017.

[101] Liu Y, Fan B, Xiang S, et al. Relation-shape convolutional neural network for point cloud analysis[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 8895-8904.

[102] Boulch A. Generalizing Discrete Convolutions for Unstructured Point Clouds[C]//3DOR@ Eurographics. 2019: 71-78.

[103] Liu Y, Fan B, Meng G, et al. Densepoint: Learning densely contextual representation for efficient point cloud processing[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 5239-5248.

[104] Thomas H, Qi C R, Deschaud J E, et al. Kpconv: Flexible and deformable convolution for point clouds[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 6411-6420.

[105] Boulch A. ConvPoint: Continuous convolutions for point cloud processing[J]. Computers and Graphics, 2020, 88: 24-34.

[106] Wu W, Qi Z, Fuxin L. Pointconv: Deep convolutional networks on 3d point clouds[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 9621-9630.

[107] Esteves C, Allen-Blanchette C, Makadia A, et al. Learning so (3) equivariant representations with spherical cnns[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 52-68.

[108] Groh F, Wieschollek P, Lensch H P A. Flex-convolution[C]//Asian Conference on Computer Vision. Springer, Cham, 2018: 105-122.

[109] Hua B S, Tran M K, Yeung S K. Pointwise convolutional neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 984-993.

[110] Lei H, Akhtar N, Mian A. Octree guided cnn with spherical kernels for 3d point clouds[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 9631-9640.

[111] Wang Y, Xu S, Zell A. Real-time 3D Object Detection from Point Clouds using an RGB-D Camera[C]//ICPRAM. 2020: 407-414.

[112] Yang B, Liang M, Urtasun R. Hdnet: Exploiting hd maps for 3d object detection[C]//Conference on Robot Learning. PMLR, 2018: 146-155.

[113] Luo W, Yang B, Urtasun R. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2018: 3569-3577.

[114] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite[C]//2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012: 3354-3361.

[115] Tu C, Takeuchi E, Carballo A, et al. Point cloud compression for 3D LiDAR sensor using recurrent neural network with residual blocks[C]//2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 3274-3280.

[116] Shi S, Wang X, Li H P. 3d object proposal generation and detection from point cloud[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA. 2019: 16-20.

[117] Vora S, Beijbom O O, Lang A H, et al. Sequential fusion for 3d object detection: U.S. Patent Application 17/096,916[P]. 2021-5-20.

[118] Yang Z, Sun Y, Liu S, et al. Std: Sparse-to-dense 3d object detector for point cloud[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1951-1960.

[119] Zarzar J, Giancola S, Ghanem B. PointRGCN: Graph convolution networks for 3D vehicles detection refinement[J]. arXiv preprint arXiv:1911.12236, 2019.
[120] Yang Z, Sun Y, Liu S, et al. Ipod: Intensive point-based object detector for point cloud[J]. arXiv preprint arXiv:1812.05276, 2018.
[121] Ku J, Mozifian M, Lee J, et al. Joint 3d proposal generation and object detection from view aggregation[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 1-8.
[122] Xu D, Anguelov D, Jain A. Pointfusion: Deep sensor fusion for 3d bounding box estimation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 244-253.
[123] Lehner J, Mitterecker A, Adler T, et al. Patch Refinement–Localized 3D Object Detection[J]. arXiv preprint arXiv:1910.04093, 2019.
[124] Shi S, Guo C, Jiang L, et al. Point-voxel feature set abstraction for 3d object detection. 2020 IEEE[C]//CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020: 10526-10535.
[125] Qi C R, Litany O, He K, et al. Deep hough voting for 3d object detection in point clouds[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 9277-9286.
[126] Zhou D, Fang J, Song X, et al. Iou loss for 2d/3d object detection[C]//2019 International Conference on 3D Vision (3DV). IEEE, 2019: 85-94.
[127] Wang Z, Jia K. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 1742-1749.
[128] Qi C R, Chen X, Litany O, et al. Imvotenet: Boosting 3d object detection in point clouds with image votes[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 4404-4413.
[129] Qi C R, Liu W, Wu C, et al. Frustum pointnets for 3d object detection from rgb-d data[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 918-927.
[130] Shin K, Kwon Y P, Tomizuka M. Roarnet: A robust 3d object detection based on region approximation refinement[C]//2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019: 2510-2515.
[131] Chen X, Ma H, Wan J, et al. Multi-view 3d object detection network for autonomous driving[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2017: 1907-1915.
[132] Lu H, Chen X, Zhang G, et al. SCANet: Spatial-channel attention network for 3D object detection[C]//ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 1992-1996.
[133] Liang M, Yang B, Wang S, et al. Deep continuous fusion for multi-sensor 3d object detection[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 641-656.
[134] Zeng Y, Hu Y, Liu S, et al. Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 3434-3440.
[135] Liang M, Yang B, Chen Y, et al. Multi-task multi-sensor fusion for 3d object detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 7345-7353.
[136] Zhao X, Liu Z, Hu R, et al. 3D object detection using scale invariant and feature reweighting networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 9267-9274.
[137] Chen Y, Liu S, Shen X, et al. Fast point r-cnn[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 9775-9784.
[138] Zhou Y, Tuzel O. Voxelnet: End-to-end learning for point cloud based 3d object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4490-4499.
[139] Yan Y, Mao Y, Li B. Second: Sparsely embedded convolutional detection[J]. Sensors, 2018, 18(10): 3337.
[140] Engelcke M, Rao D, Wang D Z, et al. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks[C]//2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 1355-1361.
[141] Li B. 3d fully convolutional network for vehicle detection in point cloud[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 1513-1518.
[142] Li X, Guivant J E, Kwok N, et al. 3D backbone network for 3D object detection[J]. ArXiv, abs/1901.08373, 2019.
[143] Li X, Guivant J E, Kwok N, et al. 3D backbone network for 3D object detection[J]. ArXiv, abs/1901.08373, 2019.
[144] Li B, Zhang T, Xia T. Vehicle detection from 3d lidar using fully convolutional network[J]. arXiv preprint arXiv:1608.07916, 2016.
[145] He C, Zeng H, Huang J, et al. Structure aware single-stage 3d object detection from point cloud[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 11873-11882.
[146] Sindagi V A, Zhou Y, Tuzel O. Mvx-net: Multimodal voxelnet for 3d object detection[C]//2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 7276-7282.
[147] Yang Z, Sun Y, Liu S, et al. 3dssd: Point-based 3d single stage object detector[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11040-11048.
[148] Chen Q, Sun L, Wang Z, et al. Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots[C]//European Conference on Computer Vision. Springer, Cham, 2020: 68-84.
[149] Meyer G P, Laddha A, Kee E, et al. Lasernet: An efficient probabilistic 3d object detector for autonomous driving[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 12677-12686.
[150] Shi W, Rajkumar R. Point-gnn: Graph neural network for 3d object detection in a point cloud[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1711-1719.
[151] Meyer G P, Charland J, Hegde D, et al. Sensor fusion for joint 3d object detection and semantic segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2019: 0-0.
[152] Taeihagh A, Lim H S M. Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks[J]. Transport reviews, 2019, 39(1): 103-128.
[153] Munoz D, Bagnell J A, Vandapel N, et al. Contextual classification with functional max-margin markov networks[C]//2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009: 975-982.
[154] Hackel T, Savinov N, Ladicky L, et al. Semantic3d. net: A new large-scale point cloud classification benchmark[J]. arXiv preprint arXiv:1704.03847, 2017.
[155] Lawin F J, Danelljan M, Tosteberg P, et al. Deep projective 3D semantic segmentation[C]//International Conference on Computer Analysis of Images and Patterns. Springer, Cham, 2017: 95-107.
[156] Behley J, Steinhage V, Cremers A B. Performance of histogram descriptors for the classification of 3D laser range data in urban environments[C]//2012 IEEE International Conference on Robotics and Automation. IEEE, 2012: 4391-4398.
[157] Armeni I, Sener O, Zamir A R, et al. 3d semantic parsing of large-scale indoor spaces[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1534-1543.
[158] Gaidon A, Wang Q, Cabon Y, et al. Virtual worlds as proxy for multi-object tracking analysis[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 4340-4349.
[159] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: The kitti dataset[J]. The International Journal of Robotics Research, 2013, 32(11): 1231-1237.
[160] Yi L, Kim V G, Ceylan D, et al. A scalable active framework for region annotation in 3d shape collections[J]. ACM Transactions on Graphics (ToG), 2016, 35(6): 1-12.
[161] Armeni I, Sener O, Zamir A R, et al. 3d semantic parsing of large-scale indoor spaces[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1534-1543.
[162] Armeni I, Sener O, Zamir A R, et al. 3d semantic parsing of large-scale indoor spaces[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1534-1543.
[163] Silberman N, Hoiem D, Kohli P, et al. Indoor segmentation and support inference from rgbd images[C]//European conference on computer vision. Springer, Berlin, Heidelberg, 2012: 746-760.
[164] Douillard B, Underwood J, Kuntz N, et al. On the segmentation of 3D LIDAR point clouds[C]//2011 IEEE International Conference on Robotics and Automation. IEEE, 2011: 2798-2805.
[165] Liu C, Yuen J, Torralba A. Nonparametric scene parsing via label transfer[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(12): 2368-2382.
[166] Badue C, Guidolini R, Carneiro R V, et al. Self-driving cars: A survey[J]. Expert Systems with Applications, 2021, 165: 113816.
[167] Bansal M, Krizhevsky A, Ogale A. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst[J]. arXiv preprint arXiv:1812.03079, 2018.
[168] Urmson C, Anhalt J, Bagnell D, et al. Autonomous driving in urban environments: Boss and the urban challenge[J]. Journal of Field Robotics, 2008, 25(8): 425-466.
[169] Meyer G P, Laddha A, Kee E, et al. Lasernet: An efficient probabilistic 3d object detector for autonomous driving[C]//Proceedings of

the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 12677-12686.

[170] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.

[171] Chen X, Ma H, Wan J, et al. Multi-view 3d object detection network for autonomous driving[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2017: 1907-1915.

[172] Besl P J, McKay N D. Method for registration of 3-D shapes[C]//Sensor fusion IV: control paradigms and data structures. International Society for Optics and Photonics, 1992, 1611: 58

[173] Akai N, Yasui K, Arashi K, Saliou K, Tsubakino D, Hara S. Bayesian Filtering Fusion of Optimization-Based Monocular Visual Localization and Autonomous Quadcopter Navigation[C]// In2022 IEEE/SICE International Symposium on System Integration (SII) 2022 Jan 9 (pp. 754-759). IEEE.

[174] Javed M A, Nafi N S, Basheer S, et al. Fog-assisted cooperative protocol for traffic message transmission in vehicular networks[J]. IEEE Access, 2019, 7: 166148-166156.. 2019

[175] Shafiq M, Tian Z, Bashir A K, et al. Data mining and machine learning methods for sustainable smart cities traffic classification: A survey[J]. Sustainable Cities and Society, 2020, 60: 102177.

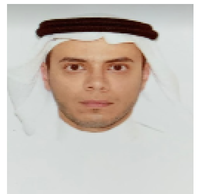
[176] Raja G, Ganapathisubramaniyan A, Anbalagan S, et al. Intelligent reward-based data offloading in next-generation vehicular networks[J]. IEEE Internet of Things Journal, 2020, 7(5): 3747-3758.

[177] Qiao F, Wu J, Li J, et al. Trustworthy edge storage orchestration in intelligent transportation systems using reinforcement learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(7): 4443-4456.

[178] Lou K, Yang Y, Wang E, et al. Reinforcement learning based advertising strategy using crowdsensing vehicular data[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(7): 4635-4647.

Rashid Abbasi (M'07) received the Ph.D. degree from the Anhui University Hefei, China, in 2019. He is currently Working as Post-Doctoral Researcher with the University of Electronic Science and Technology of China. His research interests include the 3D point cloud compression for autonomous driving, Image processing and Deep Learning.

Ali Kashif Bashir (Senior Member, IEEE) (M'07) received the Ph.D. degree in computer science and engineering from Korea University, South Korea. He is currently a Senior Lecturer at the Department of Computing and Mathematics, Manchester Metropolitan University, U.K. He is also an Adjunct Professor at the National University of Science and Technology, Pakistan. His past assignments include an Associate Professor of ICT, University of the Faroe Islands, Denmark, Osaka University, Japan, Nara National College of Technology, Japan, the National Fusion Research Institute, South Korea, Southern Power Company Ltd., South Korea, and the Seoul Metropolitan Government, South Korea. He has worked on several research and industrial projects of South Korean, Japanese, and European agencies and Government Ministries. He is also advising several start-ups in the field of STEM-based education, block chain, robotics, and smart homes. He has authored over 100 research articles and is supervising/cosupervising several graduate (M.S. and Ph.D.) students. His research interests include the Internet of Things, wireless networks, distributed systems, network/cyber security, and cloud/network function virtualization. He is an Invited Member of the IEEE Industrial Electronic Society, a member of ACM, and a Distinguished Speaker of ACM. He is serving as the Editor-in-Chief of the IEEE Future Directions Newsletter.



Hasan J. Alyamani (S'09) received his Bsc (Computer Science) from Umm Al-Qura University, Saudi Arabia in 2006, Ms (Computer Science) from The University of Waikato, New Zealand in 2012 and PhD (Computer Science) from Macquarie University, Australia in 2019. He is currently working as an Assistant Professor at the Department of Information Systems, King Abdulaziz University, Saudi Arabia.

Farhan Amin (S'09) received Ph.D. degree from the Department of Information and Communication Engineering, College of Engineering, Yeungnam University, Gyeongsan, South Korea, in October 2020. He is currently working as an Assistant Professor with the Department of Computer Engineering, Gachon University, South Korea. His research interests include the Internet of Things, AI, big data, and data science.

Jaehyeok Doh (S'09) received his Bachelor Degree from Inje University, a Master Degree from Kyungpook National University, and a Ph.D. degree from Yonsei University in South Korea, all in Mechanical Engineering. After his post-doctoral work at the Singapore University of Technology and Design (SUTD) for the past two years, he recently joined Gyeongsang National University as an Assistant Professor in the school of Mechanical Engineering. His research interests include structural analysis, reliability-based design optimization, design for additive manufacturing, uncertainty quantification, as well as prognostics and health management.

Jianwen Chen (Senior Member, IEEE) (S'09) received the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 2007. From 2007 to 2010, he was a Staff Researcher with IBM Research, where he conducted research on wireless communications systems and multi-core video coding architectures. Since, 2010, he has been with the Image Communications Lab, University of California at Los Angeles (UCLA), Los Angeles, where he is currently focusing on video signal processing/ enhancement, high efficiency video coding, and high performance computing architecture and application. He is currently a Professor with the University of Electronic Science and Technology of China. His research interests include signal processing for video and communication systems, and in particular video coding algorithm design, video quality assessment, 3-D video coding, low-complexity video codec optimization, and wireless communication protocols and systems.