Image Recognition Based on High Accuracy 3D Depth Map Information

Yu-Cheng Fan [1,*], Chun Ju Huang [2], and Chitra Meghala Yelamandala [3]

[1] Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan
[2] Siemens EDA, Wilsonville, Oregon, USA (Headquartered)
[3] Iout Private Limited Company, Guntur, India
*Correspondence: skystar@ntut.edu.tw (Y.-C.F.)

***Abstract*—In recent years, self-driving cars have developed rapidly. Many academic research institutes have begun to develop self-driving cars. However, the object recognition rate of current self-driving cars is still not high, especially the recognition of pedestrians and small objects. In order solve this problem, we presented an image recognition system adopting high-accuracy 3D depth map information. This algorithm combines two kinds of sensor data for calculation, uses the 3D depth signal of the 3D point cloud map to segment, finds the location of the small objects, and uses the color image information to recognize the object. A series of experiments have proved that the proposed scheme can generate marked color images and point cloud images and improve the efficiency of the algorithm. Our algorithm improves the disadvantages of the traditional YOLO neural network in pedestrian recognition, reduces the input image range through point cloud image segmentation, and improves the recognition rate of small objects in the YOLO network. Our method could recognize randomly small objects with over 75% recognition accuracy, outperforming other methods in the literature.**

***Keywords*—3D image, depth map, high accuracy, image recognition, smart cities**

I. Introduction

In recent years, Artificial Intelligence (AI) has become a hot topic in the news. With Google AlphaGo defeating many famous Go players, the AI industry has reached unprecedented prosperity. Among them, the self-driving car industry is the most eye-catching. Before 2009, self-driving cars were only in the experimental stage. In 2010, the Google X laboratory developed the Toyota Prius self-driving car, which successfully drove on ordinary roads with a driving distance of up to 100,000 miles. Electric vehicle Tesla, car manufacturer Delphi, Ford, Benz, BMW, and other companies have successively invested in related technology research.

By 2013, four states in the United States had passed regulations related to self-driving cars, including Nevada, California, Florida, and Michigan, and allowed self-driving cars to be tested on open roads. Subsequently, Germany also passed relevant bills to allow self-driving cars equipped with black boxes to be tested on the road.

With the vigorous development of the self-driving car industry, the academic community has also invested in the research of self-driving car-related sensors. Among them, 3D Light Detection And Ranging (3D LiDAR) is the most researched project.

3D LiDAR can generate 360° real-time depth Information, and its sensing range can reach up to 100 m, which is one of the indispensable sensors for self-driving cars. Due to the low vertical resolution of LiDAR, the object recognition rate of current autonomous cars is still not high, especially the recognition of pedestrians and small objects. We designed an image recognition system adopting high-accuracy 3D depth map information and high resolution color image in order solve this problem. The high-resolution camera provides real-time color images ahead for the algorithm to detect objects in front. Combining the information of the two sensors can realize the planning and judgment of the driving route.

We presented the new fusion scheme of 3D LiDAR point cloud map and color image to improve the performance of the object recognition. The proposed approach overcomes the disadvantages of the traditional neural network in pedestrian recognition, reduces the input data through 3D point cloud image segmentation, and increase the recognition rate of small objects in the YOLO network. Besides, this paper considers the 2D Region of Interest (ROI) and combines difference regions to reduce computing complexity. The presented scheme could recognize randomly small objects with over 75% recognition accuracy and provide a well solution.

II. Overview of the Development of Autonomous Vehicles

After the automobile was developed, inventors began to have the idea of self-driving vehicles in 1920. In 1925, Francis Houdina invented the radio-controlled car. Although the car can be remotely controlled, it is still far away from true self-driving. In 1984, Carnegie Mellon University launched two projects, Navlab and ALV. In 1986, the prototype of the self-driving car, NavLab 1, was completed in NavLab that contains five computers, three Sun workstations, and a Warp supercomputer, with a speed

Manuscript received April 3, 2023; revised June 15, 2023; accepted June 30, 2023; published October 13, 2023.

of up to 20 mile/hr (32 km/hr). In 2001, NavLab of Carnegie Mellon University published the self-driving car of NavLab 11.

After the first self-driving car came out, many research institutions and companies began to invest heavily in related research, including Mercedes-Benz, General Motors, Continental Automotive Systems, Market America, Nissan Motor, Toyota Motor, Audi, University of Oxford, and Google. In 2010, Google had a breakthrough in the development of self-driving cars. The self-driving car Toyota Prius developed by the Google X laboratory successfully drove 100,000 miles on ordinary roads. Before that self-driving cars have not yet been successfully driven on general roads. LiDAR and color cameras are the most important sensors.

In 2014, Google released Firefly, a new self-driving car prototype. This prototype has no accelerator, steering wheel, or brakes, and is 100% autonomous. Its purpose is to be used as a learning and development platform to improve the later route planning algorithm.

Using LiDAR to obtain distance information and left and right cameras to obtain image information can simulate the information received by the human eye, and develop a route planning algorithm based on this information, which can greatly reduce the probability of misjudgment.

Currently, self-driving car classification is defined by the International Society of Automatic Machine Engineers (SAE International), which is divided into Levels 0 to 5. From Level 0 to Level 2, it is an auxiliary system that still needs to rely on driving to make decisions. Level 3 has some scenarios and reliable system solutions. Level 4 and above are so-called highly automated self-driving cars. Since the system needs to handle most situations, almost all of its sensors are equipped with LiDAR and cameras to obtain a large amount of information for judgment. It shows that the two are indispensable devices for highly automated systems.

III. 3D Point Cloud Segmentation

From the above, we can know that 3D LiDAR point cloud images and color images are indispensable information for self-driving cars, and they are also one of the most popular research projects in recent years. In the research projects of object recognition and detection, most papers only take a single piece of information used as the basis for judgment, such as object segmentation and classification for LiDAR point cloud images, or object detection and identification for color images. The reason is that research related to object detection is mostly implemented by neural networks. It is easier to train and design the network architecture, so this kind of paper does not use the characteristics of the two pieces of information to combine the algorithm. For example, the LiDAR point cloud image belongs to the depth information, which is easier to implement the object segmentation algorithm, while the color image is beneficial.

LiDAR point cloud image-cutting algorithms can be roughly divided into two methods. One is to use the marked point cloud image data with a neural network for training. This type of algorithm will use the point cloud image features to find a specific single object, and is mostly used to find points. Roads or vehicles in the cloud image [1–3], since the goal of this paper, is to use the depth information of the point cloud image to segment, and then use the corresponding color image to classify, so the above algorithm is not in the scope of this paper.

Another way is to use algorithms other than neural networks for segmentation. This type of algorithm can be divided into two categories. The first type is the ground extraction-oriented cutting algorithm. Douillard *et al*. [4], Himmelsbach *et al*. [5, 6], and Cheng *et al*. [7] all use the above-mentioned algorithm. The approaches are to segment the ground points and non-ground points, and then classify the non-ground points. The advantage is that most objects will be connected to other objects due to the ground. When the ground is filtered out, the object segmentation can become easier. However, the disadvantage of this method is that the ground point is assumed to be relatively flat terrain, which will not exist in actual applications.

The second category is to use a two-dimensional grid algorithm for object segmentation, such as the algorithm mentioned in Ref. [8–11], all use a two-dimensional grid as the basis for cutting. The two-dimensional grid is a commonly used algorithm for point cloud image segmentation. The grid is used for preliminary analysis of the point cloud image. The size of the grid determines the fineness of the segmentation and the calculation time. When the point cloud image is scattered in a two-dimensional grid, we can analyze the higher density points for cutting. The above algorithms all have excellent calculation speed, but over-cutting often occurs, resulting in a decrease in the recognition rate.

Most of the above studies use the KITTI Dataset [12]. The quality of 3D point cloud image reconstruction will affect the segmentation results [13–15].

In recent years, some researchers have begun to use neural networks to realize automatic driving, and this type of method has begun to have more accurate results [16–20]. However, the calculation time and training time are too long.

In order to solve the above problems, this paper proposes image recognition based on high accuracy 3D depth map information. For object recognition, the algorithm in this paper is divided into two stages. In the first stage, the LiDAR point cloud image is used for preliminary segmentation. In the second stage, a neural network is used to identify the segmented image to achieve a high-efficiency object detection algorithm.

IV. Image Recognition Based on High Accuracy 3D Depth Map Information

This paper uses 3D LiDAR point cloud images and color images to propose an accurate object recognition algorithm. We propose the fusion technology of 3D LiDAR point cloud image and color image and improve the efficiency of the object recognition algorithm by using the characteristics of these two kinds of data.

A. KITTI Dataset

This paper uses KITTI Dataset. KITTI Dataset is a database specially designed for autonomous vehicles [12], which uses the autonomous driving platform Annieway to collect relevant road information [21, 22].

Its autonomous driving platform is equipped with a Velodyne LiDAR scanner, a high-resolution color camera, and a GPS positioning system, which are used to collect LiDAR point cloud images, corresponding color images, and vehicle location information [23, 24]. The information categories include object detection, object Tracking, visual ranging, etc., and this paper will use the KITTI Dataset object detection database for experiments and tests [12].

There are three different types of data in the object detection database, which are color images, LiDAR point cloud images, and camera internal parameters. Since the color camera only collects front images, the provided LiDAR point cloud image is cut, and only the point cloud image information at 45° in front is taken and then enters the point cloud image object cutting stage [12].

Since the color camera only collects the front image, it is necessary to cut the LiDAR point cloud image provided by the database, and only take the point cloud image information of 45° in front and then start the point cloud image object segmentation stage [12].

B. Point Cloud Segmentation

The point cloud segmentation algorithm we designed includes layer segmentation, layer merging, and ground removal. Since the neural network in the next step will recognize vehicles, bicycles, and pedestrians, background objects will be removed together in the algorithm to reduce the recognition range at the back end to achieve the goal of acceleration.

C. 2D Region of Interest Extraction

After the point cloud object is segmented, we can select the area to be tested on the color image according to each object cluster [25]. Since the point cloud image is 3D information, and the color image is 2D information, we need to use the 3D to the 2D algorithm [26, 27].

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

$$u = \frac{m_{11}x + m_{12}y + m_{13}z + m_{14}}{m_{31}x + m_{32}y + m_{33}z + m_{34}}$$

$$v = \frac{m_{21}x + m_{22}y + m_{23}z + m_{24}}{m_{31}x + m_{32}y + m_{33}z + m_{34}}$$

To match point cloud images and color images, the algorithm is shown in Eq. (1), where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the output two-dimensional coordinates, $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ are the input three-dimensional coordinates and the middle is the camera parameters provided by KITTI Dataset, $\boldsymbol{f_u}$, and $\boldsymbol{f_v}$ are the focal length of the camera, $\boldsymbol{u_0}$ and $\boldsymbol{v_0}$ are the initial coordinates of the camera, $\boldsymbol{R}$ and $\boldsymbol{t}$ are the rotation matrices, by substituting the object cluster into formula 1, we can use the 3×4 $\boldsymbol{M}$ array to find the corresponding color image area, and send this area to the neural network for the next stage of judgment [28–30]. Fig. 1 is the result of two-dimensional ROI extraction. It can be seen that there are two selected areas in the figure.

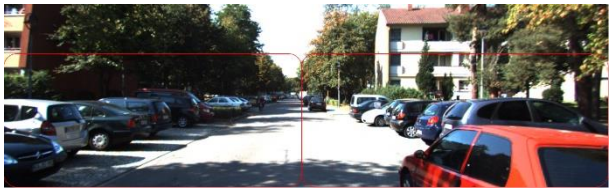

Figure 1. Two-dimensional ROI extraction result.

D. Neural Network Architecture

R-CNN, Faster-RCNN, FCN neural networks, etc. can be used during object detection. Since the algorithm in this paper is expected to be applied to self-driving cars, even if the accuracy is high, the speed does not meet the requirements, the output information often does not meet the current situation. This will cause the system to judge completely wrong. Therefore, the speed of the algorithm is the primary consideration of this paper.

At present, the fastest neural network architecture for object detection is the YOLO neural network. This paper considers the 2D ROI generated by the LiDAR segmentation algorithm and combines it with the YOLO network to construct the R-YOLO network.

We input the color image and 2D ROI information into the R-YOLO network and use the ROI information to slice the color image into several pieces. For the sake of computational efficiency and the possibility of overlapping ROI information, we combine several ROIs into one or two regions and input them into the back-end YOLO neural network. After obtaining the object detection result, merge it with the front-end ROI information to obtain the final output result.

V. Experimental Results

Firstly, the experimental environment, software, and database used in this system are introduced, as shown in Table I.

TABLE I. Simulation Environment

Simulation Environment	
Hardware	Personal Computer, Training Computer
Software	Matlab R2018a
Dataset	Pascal VOC, KITTI

TABLE II. PC Specification

PC Specification	
OS	Microsoft Windows 10 Professional 64-bit
CPU	Intel® Core™ i7-4770 Processor
RAM	8.00 GBytes

We use personal computers and train dedicated computers in the experimental environment. Since the back-end R-YOLO neural network requires a lot of calculations during training, the dedicated computer for training is equipped with a GPU with high computing performance to increase computing efficiency. Table II and Table III show the specifications of the personal computer and training computer respectively.

TABLE III. SPECIFICATIONS OF THE TRAINING COMPUTER

Specifications of the Training Computer	
OS	Ubuntu
CPU	Intel® Core™ i7-7700K Processor
RAM	32.00 GBytes
Graphic Processor	NVIDIA GeForce GTX 1080 Ti

In the implementation of the LiDAR segmentation algorithm, we use Matlab for development, which contains several simulation-specific function libraries, which can analyze and display the results of the algorithm. During the implementation process, we can also observe the result figures of the functions of each stage, and adjust the algorithm.

For the database, we use the Pascal VOC database to train the R-YOLO neural network and use the KITTI database to test the final result of the algorithm. The KITTI database contains color images and corresponding point cloud images. Table IV shows the specifications of the LiDAR. The scanning depth can reach up to 120 m, the horizontal scanning range is 360°, and the vertical scanning range is 26.9°.

TABLE IV. LIDAR SPECIFICATION

LiDAR Specification	
Name	HDL-64E
Range	120m
Data Rate	Up to 2.2 Million Points per Second
Vertical FOV	26.9º
Vertical Resolution	0.4º
Horizontal FOV	360º
Horizontal Resolution	0.08° Angular Resolution

The above is the experimental environment and the database and sensors used in this paper. In the experimental method part, we will first implement the LiDAR segmentation algorithm on a personal computer, and send the generated bounding box information to a dedicated training computer. We use R-YOLO for object detection and compare the detection results with the bounding box to produce the final result map.

Next, we will show the results of the experiment and analyze the advantages and disadvantages of the algorithm in this paper in different scenarios. Figs. 2–6 are the result pictures numbered 502. The first two are the processing steps of color images, and the last three are the processing steps of the point cloud image.

Figure 2. Original color image of No. 502.

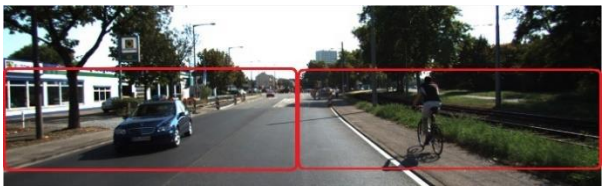

Figure 3. ROI of color image No. 502.

Fig. 4 is the point cloud image scanned by HDL-64E LiDAR in the KITTI database, and is positioned by GPS. The point cloud image of Fig. 4 was captured by driving around the mid-size city of Karlsruhe. This scene contains cars, bicycles, roads, trees etc. The effective range of LiDAR scanning is about 120m. Vertical FOV is 26.9º. Vertical Resolution is 0.4º. Horizontal FOV is 360º. Horizontal Resolution is 0.08° Angular Resolution.

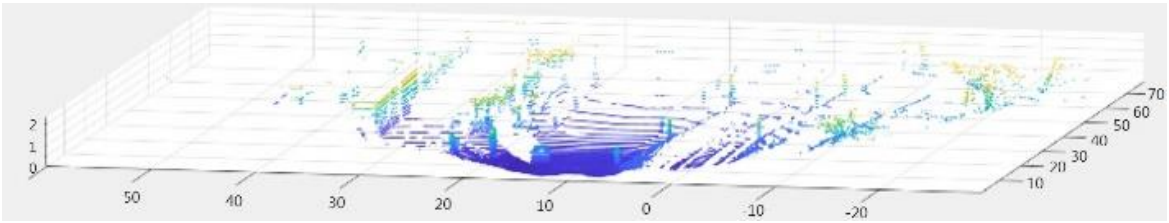

Figure 4. Point cloud image No. 502.

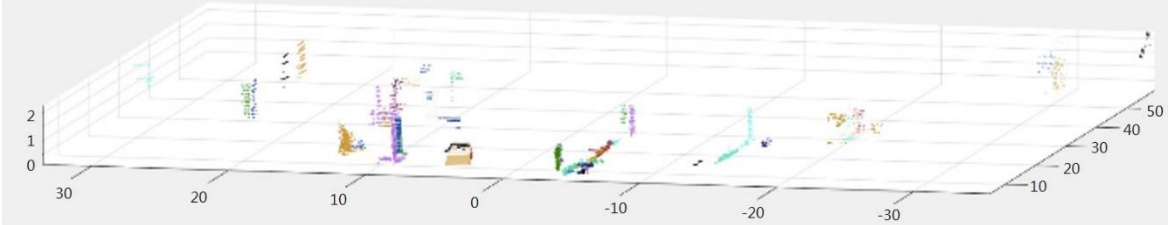

Figure 5. Segmentation result of the point cloud image No. 502.

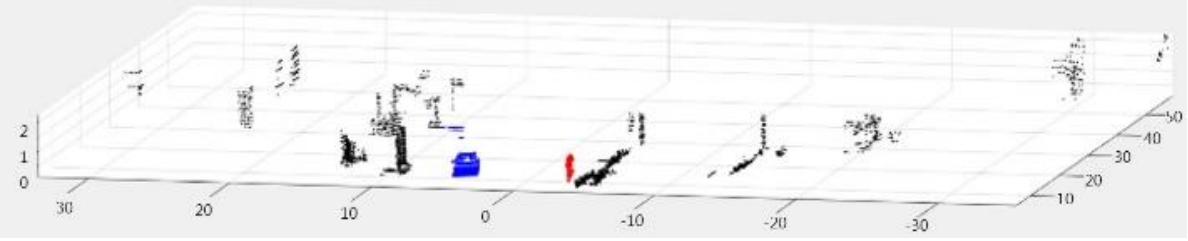

Figure 6. Marking result of point cloud image No. 502.

It can be seen that some backgrounds are marked as vehicles (blue) in the part of the point cloud image marking result. The reason is that when combining the bounding box and the detection frame, a detection frame may not only be selected as a single bounding box, so small objects behind the vehicle will also be framed together, but they will be judged as obstacles in the end, so there is no security concern. On the other hand, it also compensates for the excessive segmentation of the point cloud image due to the above situation problem, it can be seen that the vehicle in Fig. 5 is over-segmented, but after combining the detection results, all points of the vehicle are marked as the same object as shown in Fig. 6, and the red part in the figure represents a pedestrian.

VI. Conclusion

This paper proposes an algorithm for real-time object detection and greatly increases the recognition rate of pedestrians. However, it is easy to produce errors in the part where the final bounding box and the detection box are combined. The main reason for the error is that the algorithm in this paper takes immediacy as the main consideration, so the accuracy of point cloud image segmentation will be sacrificed, which will further cause errors in the final results. Therefore, if the accuracy of point cloud image segmentation can be improved, the error in the final data combination will be reduced, and the neural network has a similar situation. Due to the large reduction in the number of layers, the recognition rate tends to decline in more complex scenes. In the future, we could try to increase the number of network layers or structures to achieve a better trade-off between time and performance.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Yu-Cheng Fan has investigated the ideas, system, algorithm, and methodology of the proposed techniques, and wrote the manuscript; Chun Ju Huang and Chitra Meghala Yelamandala implemented the proposed system, conducted the experiments, analyzed the experimental data, and provided the analytical results, and wrote the manuscript with support from Yu-Cheng Fan. All authors discussed the results and contributed to the final manuscript. All authors had approved the final version.

Funding

This work was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 110-2221-E-027-084-MY3.

Acknowledgment

The authors gratefully acknowledge the Taiwan Semiconductor Research Institute (TSRI), for supplying the technology models used in IC design.

References

[1] Z. Chen, J. Zhang, and D Tao, "Progressive LiDAR adaptation for road detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, pp. 693–702, 2019.

[2] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "Lidar-camera fusion for road detection using fully convolutional neural networks," *Robotics and Autonomous Systems*, vol. 111, pp. 125–131, 2019.

[3] Y. C. Fan, Y. C. Liu, and C. A. Chu, "Efficient CORDIC iteration design of LiDAR sensors' point-cloud map reconstruction technology," *Sensors*, vol. 19, issue 24, 5412, pp. 1–28, Dec. 2019.

[4] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3D LIDAR point clouds," in *Proc. 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 2798–2805.

[5] M. Himmelsbach, F. V. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in *Proc. 2010 IEEE Intelligent Vehicles Symposium*, San Diego, CA, USA, Jun. 21–24, 2010, pp. 560–565.

[6] M. Himmelsbach and H. J. Wuensche, "Tracking and classification of arbitrary objects with bottom-up/top-down detection," in *Proc. 2012 IEEE Intelligent Vehicles Symposium*, Spain, Jun. 3–7, 2012, pp. 577–582.

[7] J. Cheng, Z. Xiang, T. Cao, and J. Liu, "Robust vehicle detection using 3D LiDAR under complex urban environment," in *Proc. 2014 IEEE International Conference on Robotics and Automation*, Hong Kong, China, May 2014, pp. 691–696.

[8] M. Himmelsbach, T. Luettel, and H. Wuensche, "Real-time object classification in 3D point clouds using point feature histograms," in *Proc. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, MO, USA, Oct. 10–15, 2009, pp. 994–1000.

[9] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito, and J. Miura, "Pedestrian recognition using high-definition LIDAR," in *Proc. 2011 IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, Jun. 5–9, 2011, pp. 405–410.

[10] Y. C. Fan, C. M. Yelamandala, T. W. Chen, and C. J. Huang, "Real-time object detection for LiDAR based on LS-R-YOLOv4 neural network," *Journal of Sensors*, vol. 2021, 5576262, pp. 1–11, May 2021.

[11] D. Held, J. Levinson, S. Thrun, and S. Savarese, "Combining 3D shape, color, and motion for robust anytime tracking," in *Proc. 2014 Robotics: Science and Systems Conference*, Berkeley, CA, USA, Jul. 12–16, 2014.

[12] KITTI dataset. [Online]. Available: http://www.cvlibs.net/datasets/kitti/

[13] Y. C. Fan, B. T. Wu, C. J. Huang, and Y. H. Bai, "Environment detection of 3D LiDAR by using neural networks," in *Proc. 2019*

IEEE International Conference on Consumer Electronics, 2019, pp. 1–2.
[14] Y. C. Fan, L. J. Zheng, and Y. C. Liu, "3D environment measurement and reconstruction based on LiDAR," in *Proc. 2018 IEEE International Instrumentation and Measurement Technology Conference*, 2018, pp. 1–4.
[15] B.-T. Wu, P.-C. Li, J.-H. Chen, Y.-J. Li, and Y.-C. Fan, "3D environment detection using multi-view color images and LiDAR point clouds," in *Proc. 2018 IEEE International Conference on Consumer Electronics-Taiwan*, 2018, pp. 1–2.
[16] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 712–733, Feb. 2021.
[17] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust Lane detection from continuous driving scenes using deep neural networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 41–54, Jan. 2020.
[18] K. Yu, L. Lin, M. Alazab, L. Tan, and B. Gu, "Deep learning-based traffic safety solution for a mixture of autonomous and manual vehicles in a 5G-enabled intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4337–4347, July 2021.
[19] Y. C. Fan, W. L. Mao, and H. W. Tsao, "An artificial neural network-based scheme for fragile watermarking," in *Proc. 2003 IEEE International Conference on Consumer Electronics*, 2003, pp. 210–211.
[20] Y. Zhang *et al*., "PolarNet: An improved grid representation for online LiDAR point clouds semantic segmentation," in *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9598–9607.
[21] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," *in Proc. 2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.
[22] H. Caesar *et al*., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11618–11628.
[23] J. Zhang, W. Xiao, B. Coifman, and J. P. Mills, "Vehicle tracking and speed estimation from roadside lidar," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5597–5608, 2020.
[24] Y. Zhang, H. Xu, and J. Wu, "An automatic background filtering method for detection of road users in heavy traffics using roadside 3-D LiDAR sensors with noises," *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6596–6604, 2020.
[25] Y. C. Fan, A. Chiang, and J. H. Shen, "ROI-based watermarking scheme for JPEG-2000," *Circuits, Systems, and Signal Processing*, vol. 27, no. 5, pp. 763–774, Oct. 2008.
[26] Y.-C. Fan, Y.-C. Chen, and S.-Y. Chou, "Vivid-DIBR based 2D–3D image conversion system for 3D display," *Journal of Display Technology*, vol. 10, no. 10, pp. 887–898, Oct. 2014.
[27] Y.-C. Fan, J.-C. Chiou, and Y.-H. Jiang, "Hole-filling based memory controller of disparity modification system for multi-view three-dimensional video," *IEEE Transactions on Magnetics*, vol. 47, no. 3, pp. 679–682, March 2011.
[28] C. R. Qi, X. Chen, O. Litany, and L. J. Guibas, "ImVoteNet: Boosting 3D object detection in point clouds with image votes," in *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4403–4412.
[29] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self-calibration of high-resolution LiDAR and camera in targetless environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7517–7524, Oct. 2021.
[30] L.-H. Wen and K.-H. Jo, "Fast and accurate 3D object detection for lidar-camera-based autonomous vehicles using one shared voxel-based backbone," *IEEE Access*, vol. 9, pp. 22080–22089, 2021.