DOI 10.1007/s10946-021-09986-x
Journal of Russian Laser Research, Volume 42, Number 4, July, 2021

LIDAR SENSOR-BASED OBJECT RECOGNITION USING MACHINE LEARNING

Rui Wang,[1] Mengyu An,[1] Sihan Shao,[1] Mingyang Yu,[1] Shifeng Wang,[1,2] and Xiping Xu[1*]

[1]*School of Optoelectronic Engineering*
Changchun University of Science and Technology
Weixing Road 7089, Changchun City, Jilin 130022, China

[2]*Key Laboratory of Optoelectronic Measurement and Optical Information Transmission Technology*
Changchun University of Science and Technology
Weixing Road 7089, Changchun City, Jilin 130022, China

*Corresponding author e-mail: xipingxu_opto@163.com

Abstract

Light detection and ranging (LIDAR) sensor provides complicated and large volume of environmental point cloud data that are essential for the target recognition. In this study, we present a novel LIDAR sensor-based target recognition approach of the point cloud data, using adaptive rounding algorithm and optimized support vector (SVM). First, the multilayer LIDAR is used to obtain three-dimensional point cloud data of the surrounding environment. Second, we use the grid-occupied method to cluster the point cloud data after reducing redundant point cloud data through a self-adaptive removal method. Third, the multiple features of target are extracted, which classified by a novel SVM based on block feature. Finally, the classifier achieves the best effect through parameter optimization, and the different target objects are distinguished. The experimental results show that the classification accuracy of the target recognition method proposed can reach 93.75% under the premise of reducing training features by at least 33.25%. Therefore, the target recognition method proposed in this study can complete the classification of objects in the target area with a significant increase in accuracy compared to previous approaches.

Keywords: LIDAR point cloud, target recognition, support vector machine, block feature.

1. Introduction

Light detection and ranging (LIDAR) is capable of obtaining three-dimensional spatial information on objects [1] accurately with reliable stability and improving the accuracy of obstacle detection and recognition. The perception and recognition of urban area environment play a prominent role in diverse LiDAR applications [2]. Traditional pedestrian and vehicle detection algorithms mostly use image data collected by cameras [3]. However, the complexity of the urban environment would have a great impact on the image quality of the camera, hence limiting the accuracy of target classification. Compared with images, LIDAR has the advantage of acquiring depth data, which are not adversely affected by weak signals and weather conditions. The method of supervised learning in machine learning [4] is used to ensure the accuracy of classification. Support vector machine (SVM) [5] is one of the most widely used

Manuscript submitted by the authors in English on April 20, 2021.

classifiers in supervised learning. It can achieve high classification accuracy under the condition that the required number of samples is relatively small. Nevertheless, as the training time increases, its performance decreases since the data set increases.

To improve the classification accuracy, this study proposes a LIDAR sensor-based object recognition method using adaptive rounding algorithm and block feature classification. First, the redundant point cloud data is reduced by a self-adaptive removal method, hence prevented effectively the performance degradation of SVM due to too large data set. Meanwhile, a new method based on block feature is used to make the effect of classifier reaches the best. Experimental results show that this method can reduce effectively the impact of large data set on SVM performance and improve classification accuracy.

2. LIDAR Point Cloud Data Acquisition and Preprocessing

The system used in the present study is based on a test car equipped with a Velodyne HDL-32E that facilitates LIDAR, which is an active form of remote sensing. It measures the distance from the sensor to the object by calculating the time interval between an emitted laser pulse and the detected reflected signal [6] and obtains angle and intensity information by combining the built-in laser arrangement angle and angle measurement technology. The parameters of Velodyne HDL-32E are shown in Table 1. To obtain effectively the point cloud features of pedestrians and vehicles, we adopt the center-to-center installation method [7]. In addition, the installation height determined to be 1.85 m by referring to the height of pedestrians and common vehicles [8]. The harness distribution and installation of LIDAR are shown in Fig. 1.

Table 1. The Parameters of Velodyne HDL-32E.

Height inch	Diameter inch	Weight kg	Number of Sensor Channels	Laser Wavelength	Sensor Rotation Rate	Power Consumption
5.7	3.4	1	32	903 nm	5 – 20 Hz	12 W

Measurement Range	Range Accuracy	Angular Resolution		Field of View	
		Vertical (τ)	Horizontal (η)	Vertical	Horizontal
Up to 100 m	< 2 cm (within 25 m)	1.33°	0.16°	$-30.67°$ to $+10.67°$	360°

Fig. 1. System configuration; here, the distribution of LIDAR harness (a) and the installation of LIDAR (b).

2.1. LIDAR Point Cloud Clustering

Multilayer LIDAR can obtain 700,000 scanning points per second [9], so it is necessary to remove redundant point clouds to improve the efficiency of clustering. The fewer interference points on the ground and the region of non-interest, the smaller impact on the clustering of pedestrians and vehicles. It can be assumed that the point cloud data of trees and buildings in the environment are invalid, so the redundant data can be filtered by setting a height threshold. The threshold is set as 2.5 m to ensure that the candidate pedestrians and vehicles are included in the region of interest (ROI), which is also known as the estimated short region [10]. The ground LIDAR cloud point is then removed, and an adaptive algorithm is introduced to address sloped-ground removal, since this is likely to be incorrectly identified as an object. In Fig. 2, we show a schematic diagram of the point cloud segmentation and de-redundancy process.

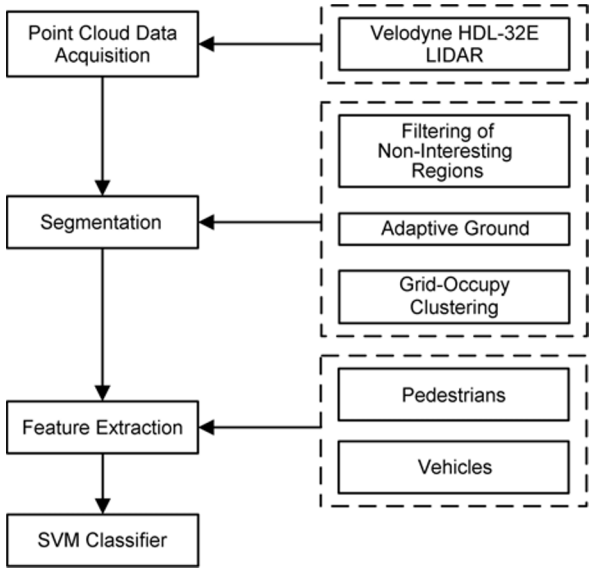

Fig. 2. Flow chart of segmentation and de-redundancy process of LIDAR data.

Finally, the LIDAR point cloud is projected on the $X-Y$ plane. The occupancy grid map [11] is built according to the density of the point cloud. In Fig. 3, the gray points in each grid represent the sum of the number of LIDAR point clouds that fall on a single grid. Use the four-direction searching rule [12] to search in the order from top-to-bottom and then from left-to-right. After data searching, a linked area is regarded as one separate category of data clustering. When the point number in the grid area is smaller than the pre-setting threshold, the clustering is discarded.

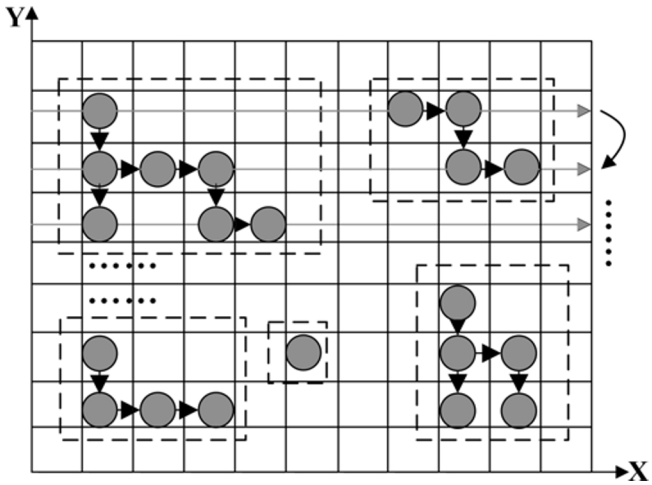

Fig. 3. Projection to the $X-Y$ plane and searching.

The resolution of grid is defined as follows:

Assume the original dataset is $D_S = \{L_{P_i} \in R^d \,|\, L_{P_i} = (x_i, y_i, z_i),\ i = 1, 2, \cdots, N\}$, where (x_i, y_i, z_i) is the coordinate of point clouds. Based on the distribution of points on the $X-Y$ plane, parameters $(x_{\min}, y_{\min})$, $(x_{\max}, y_{\max})$, and the grid size S can be used to show that the resolution of the grid is

$$m \times n, \qquad \text{where} \qquad m = \frac{x_{\max} - x_{\min}}{S} + 1, \qquad n = \frac{y_{\max} - y_{\min}}{S} + 1. \tag{1}$$

The grid size S is 0.1×0.1 m in this study; this parameter is determined by the LIDAR installation position and its internal parameters. Within the effective detection range of Velodyne HDL-32E used in this study (i.e., 25 m), the distance between two adjacent point clouds in the same plane is <0.1 m.

2.2. Self-Adaptive Removal Method

Pedestrian and vehicle point clouds are connected to each other through the ground. Therefore, the target point clouds will be separated from each other after the operation of ground removal is performed, thereby reducing the impact of redundant point clouds on clustering. The traditional ways to remove ground are to treat the ground as a fixed plane [13]. Whereas urban roads are undulating generally, so the ground with a certain inclination angle can be often misjudged as the target object. For this study, we propose a self-adaptive removal method for ground removal. Specific steps are as follows:

1. The point cloud characteristics of pedestrians and vehicles in the region of interest (ROI) are summarized and statistically analyzed to determine the relationship between the density of the target point cloud ρ_i in the region and the distance between the points in the neighborhood d_{ij} [14], which is expressed as $\rho_i = \dfrac{k}{\sum_{j=1}^{k} |d_{ij}|}$.

2. The point cloud density range and height range of pedestrians and vehicles in the region of interest are set.

3. The minimum Z value of the Z axis in the point cloud data in the region of interest is found out and a certain height is deleted.

4. The point cloud is projected into the $X-Y$ plane, and the corresponding interval parameter T and interval number N are determined in the projection plane [15]. If the point cloud density is within the density range, proceed to the next step. If not, the data in the interval will be removed as redundancy, and this step will be repeated N times.

5. Judge the change of point cloud density with the Z value in the above interval. If the point cloud density in the interval changes sharply with change in the Z value, hence indicates that there may be pedestrians or vehicles in the area. Otherwise, the data in the interval will be removed as redundancy.

The results of self-adaptive removal are shown in Fig. 4.

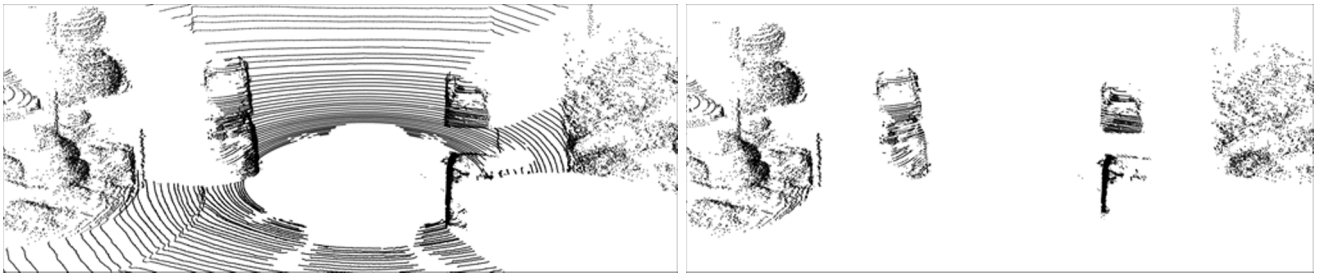

Fig. 4. Effect of using self-adaptive removal algorithm; here, without removal algorithm (left panel) and with removal algorithm (right panel).

3. Classification

The classification of point cloud data is the process of identifying point cloud attributes. The accurate classification has great significance for improving the reliability of modeling. After the process of segmentation, we apply a novel feature-based support vector machine (SVM) for classification [16].

3.1. SVM Classification Algorithm

SVM is a supervised learning algorithm that originated in statistical machine learning theory. The algorithm attempts to find a classification plane maximizing the margins on both sides of it.

The soft edge concept [17] is used to optimize the separation hyperplane,

$$\min(\|w\|^2) + C\sum_i \varepsilon_i y_i (w^T x_i + b) + \varepsilon_i \geq 1, \tag{2}$$

where ε_i is the slack variable introduced to avoid overfitting, w is the normal vector of the classification plane, x is the input eigenvector, C is the penalty term, (x_i, y_i) is a given data set, and b is a constant.

The point cloud distribution of pedestrians and vehicles is often nonlinear, and classification cannot be achieved through simple mapping. Consequently, it is necessary to use the kernel function to calculate the inner product function of the implicit mapping between the vectors [18]. The kernel function greatly simplifies the inner product operation of solving the mapping space. Commonly-used kernel functions include linear kernel function, radial basis function (RBF) kernel, and polynomial kernel function. The three different kernel functions have been tested under the experimental data in this study, and the results are shown in Table 2.

Table 2. Kernel Function Training Results.

Kernel function	Support vector number	Number of iterations	Recognition accuracy
Linear kernel function	426	1131	93.7505%
Radial basis function (RBF)	945	924	90.3461%
Polynomial kernel function	2456	1426	57.8259%

Since the simple structure of the linear kernel function, the number of iterations is relatively small, and the prediction speed is faster while improving the classification accuracy effectively. Therefore, in this study, we use the linear kernel function as the kernel function of the SVM classifier.

3.2. Feature Extraction

With the LIDAR data clusters, many variables can be used as an input for SVM training [19]. However, the data are often redundant and possess irrelevant variables. Consequently, a feature selection procedure is necessary to address this problem. To describe the characteristics of a clustered point cloud more accurately, we propose a concept of block area based on the outline feature and internal feature of the point cloud area.

3.3. Outline Features Extraction

We establish a bounding box [20] on the clustered point clouds to describe the external features more conveniently. The spatial shape and position information on the target object in the point cloud data is available and clear. The spatial geometric information including height, projected areas, and volume of bounding box are usually set as features for training.

To describe the size of target data set, we define Δh as the difference between maximum and minimum heights of the LIDAR point bounding box. Here, l and w represent the length and width of projected points of the bounding box in the $X-Y$ plane. Generally, the height between pedestrians and vehicles is of little difference. We amplify height difference through employing Δh^3 as training feature. Meanwhile, use the lw, $l\Delta h$, $w\Delta h$, $l\Delta hw$, and l/w to enlarge and shrink the border of the box, expecting to contribute to classification accuracy. These quantities are used to construct the spatial feature of the target point cloud

$$f_1 = \{\Delta h, \Delta h^3, lw, l\Delta h, w\Delta h, l\Delta hw, l/w\}. \tag{3}$$

The target object's centroid vector and centroid variance vector are defined as follows:

$$f_2 = (\bar{x} - x_{\min}, \bar{y} - y_{\min}, \bar{z} - z_{\min}), \qquad f_3 = \sqrt{\sum_{i=1}^{N} \frac{f_2^2}{n-1}}, \tag{4}$$

where x, y, and z are the average of the coordinate on the X, Y, and Z axes of the internal LIDAR spot of the target object, respectively.

3.4. Internal Feature Extraction

As the distance d increases, the number of point clouds N contained in the target point cloud decreases as

$$f_4 = \left\{ N = \frac{\alpha w + \beta h}{\gamma D^2} \right\}, \tag{5}$$

where α, β, and γ are the weight of these parameters separately. Since the vertical and horizontal resolutions of LIDAR are 1.33° and 0.16°, respectively, and the weight of width α is greater than the weight of the height β. Based on the characteristics of the LIDAR point-cloud density, we define density as another feature. Similarly, point-cloud data within the clustering is relatively denser; thus, the dispersion degree of point cloud could be defined as another feature. We define the spatial dispersion as the ratio of the number of LIDAR points to the bounding box volume,

$$f_5 = \frac{N}{l\Delta hw}, \tag{6}$$

where N represents the number of points within the clustered segmentation. The novel LIDAR point intensity feature [21] is invented to describe point cloud data. For a vehicle, it is recognized that the license plate's reflection intensity differs from other regions [22]. Graphical representations of vehicular plates are shown in Fig. 5. The surface material of vehicle is mostly steel and the difference in the laser reflection intensity is small, which is in stark contrast to the plate region. We select the mutation of the license plate's intensity as a training feature due to the distinct intensity information associated with the license plate.

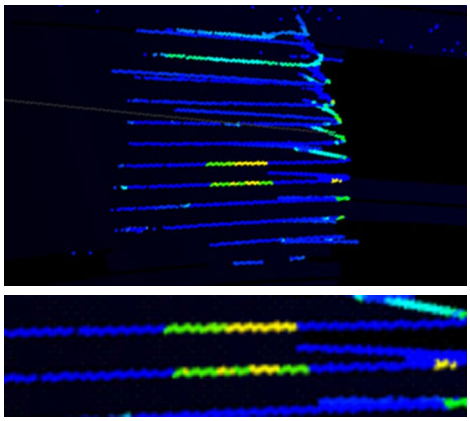

Fig. 5. License plate intensity feature: general view (top) and the plate enlarged (bottom).

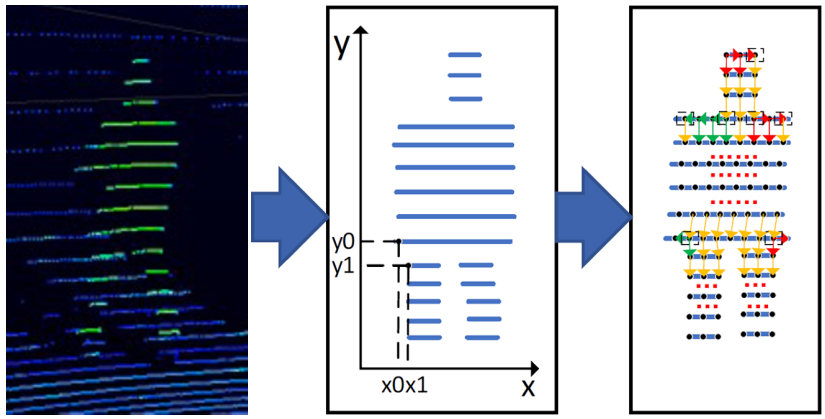

Fig. 6. Block feature of a pedestrian.

To describe the internal point cloud characteristics of pedestrians, the concept of a shape number [23] descriptor is introduced. This method is utilized to perform secondary clustering within the clustered point cloud. In this case, we take pedestrians as an example. After the process of pedestrian clustering, the body can be divided into such sections as head, upper body, and legs. Based on the morphological relationship between these individual clustering, pedestrians can be re-clustered into 3 to 4 categories. We use the number of categories after secondary clustering as another feature. The specific steps involved in LIDAR data re-clustering are as follows:

(1) Project the clustered data onto the plane which is assumed to have a maximum area

$$\{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}.$$

(2) Form a 2D lattice grid and search the area via four-neighborhood method.

(3) Analyze each point's search pattern of neighborhood based on the searching direction combination.

This process is illustrated in Fig. 6.

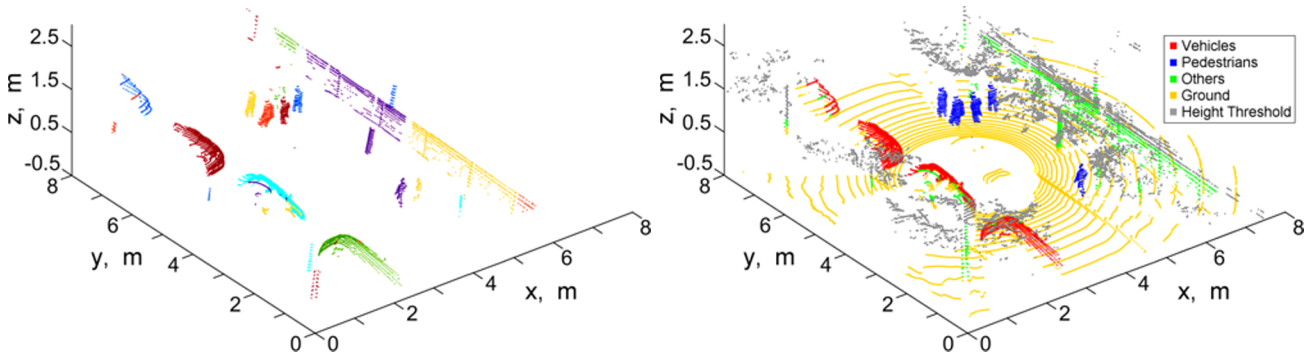

Fig. 7. Experimental results with segmentation result (a) and classification result (b).

4. Experiment and Results

In order to verify the efficiency of our method for clustering, we conduct experiments in a complex urban environment. The experimental platform is i7 8700K CPU, 16 GB memory, Windows 10 operating system, using a MATLAB R2019b environment to achieve classification results. The experimental results show that the classification accuracy is as high as 93.75%. The training sample library constructed in this study contains 1,327 positive samples and 1,742 negative samples of pedestrians and vehicles, which can basically meet various possible interferences. The clustering result is illustrated in Fig. 7 a. The different colors represent the various clustered point cloud data. In this scenario, we collectively cluster the scanned environment into 39 classes, which cover all the pedestrian and vehicle candidate clusters. A total of 70,648 points of data are included in the frame scene, and it takes 1.28 s to perform data classification.

In order to reduce the redundant features extracted in vehicle detections, the principle component analysis (PCA) [24] is introduced to convert a set of possibly correlated observations into a set of values of linearly uncorrelated variables. We are able to filter the input features and subsequently achieved at least 33.25% reduction of training features. The reduction is visualized in Fig. 8.

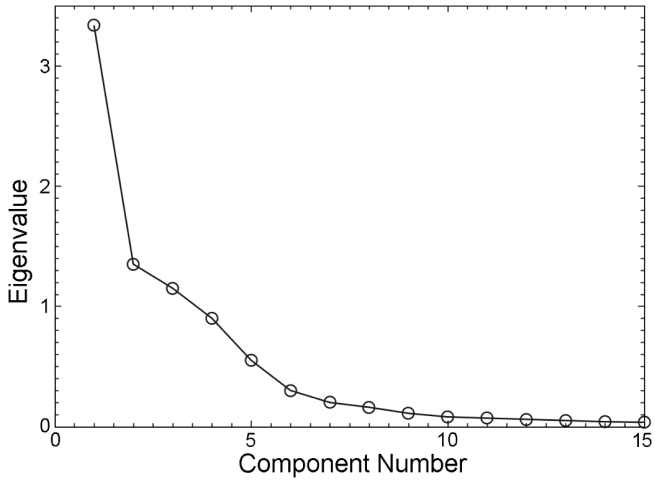

Fig. 8. Scree plot.

Meanwhile, in order to verify the contribution of block features to the recognition results, the data based on block features and non-block features are classified respectively. The results show that, after introducing block features, the classification accuracy can reach 93.7505%, which is 3.245% higher than that without this feature. The visual classification result is shown in Fig. 7 b.

We implement the same procedures on selected samples through KNN, Naïve Bayes, Multilayer Perception, Random Forest, Logistic Regression, and SVM as a comparison within classifiers. The classifier accuracy results are shown in Table 3.

Table 3. Comparison of Different Classifiers.

	Correctly Classified Rate	Kappa Coefficient	Root Mean Square Error	Relative Absolute Error
SVM	93.7505%	0.8964	0.1857	14.3023%
KNN(K=3)	89.0244%	0.8334	0.2325	17.1030%
Naïve Bayes	90.2439%	0.8526	0.2511	15.0269%
Multilayer Perception	93.0894%	0.8952	0.1873	14.3037%
Random Forest	92.6829%	0.8888	0.1880	18.7592%
Logistic Regression	90.6504%	0.8579	0.2475	14.6731%

5. Conclusions

Based on LIDAR sensor, in this study, we investigated a grid-occupied based segmentation method and proposed a novel block morphology feature based on pedestrian data for classification, using SVM. A more adaptive algorithm is presented for removing effectively the undulating road surface in the urban environment, reducing 33.25% training characteristics. Region growing is used in four-neighborhood to obtain clustering point clouds. The experimental results demonstrate that an accuracy of 93.7505% is achieved, which is an increase of 3.245% compared to the case where the feature is not used. Furthermore, when compared to several existing methods, the classification results indicate that the SVM approach yields good performance. Therefore, the method in this study is effective in improving accuracy of target recognition.

Acknowledgments

Rui Wang and Mengyu An contributed equally to this work. This work is funded by the Natural Science Foundation of Jilin Province under Grant No. 20150101047JC and is also supported by OptoBot Lab. National Demonstration Center for Experimental Optoelectronic Engineering Education, School of Optoelectronic Engineering.

References

1. A. Iavarone and D. Vagners, *Remote Sens.*, **34-5/W10**, 24 (2003).
2. K. Williams, J. M. Olsen, G. V. Roe, and C. Glennie, *Remote Sens.*, **5**, 4652 (2013).
3. S. Bakheet, *J. Russ. Laser Res.*, **38**, 61 (2017).
4. M. Zhou, Y. Tang, Z. Tian, and X. Geng, *IEEE Access*, **5**, 4388 (2017).
5. L. Bottou and C. J. Lin, "Support vector machine solvers," in: L. Bottou, O. Chapelle, D. DeCoste, and J. Weston (Eds.), *Large-Scale Kernel Machines*, MIT Press (2013), pp. 1–27.
6. M. N. Favorskaya and L. C. Jain, "Overview of LIDAR technologies and equipment for land cover scanning," in: *Remote Sensing and Geographic Information Systems*, Springer, Cham (2017), Vol. 122, p. 19.
7. B. J. Guerreiro, C. Silvestre, and P. Oliveira, *Rob. Auton. Syst.*, **62**, 1116 (2014).
8. J. K. Yang, P. Lövsund, C. Cavallero, and J. Bonnoit, *J. Crash Prevention and Injury Control*, **2**, 131 (2000).
9. Y. Chen and J. Chien, *J. Vis. Commun. Image Represent.*, **25**, 659 (2014).
10. A. Borcs, B. Nagy, and C. Benedek, *IEEE Geosci. Remote Sens. Lett.*, **PP(99)**, 1-5 (2017).
11. M. Salem, *Building an Efficient Occupancy Grid Map Based on Lidar Data Fusion for Autonomous Driving Applications*, PhD Theses, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, Stockholm (2019).
12. Y. Ao, L Wang, J. Wan, and K. Xu, *J. Wireless Com. Network*, **2019**, 214 (2019).
13. M. K. Lin, *Astrophys. J.*, **765**, 84 (2013).
14. D. Rato and V. Santos, "Detection of road limits using gradients of the accumulated point cloud density," in: *Robot 2019: Fourth Iberian Robotics Conference Proceedings, Advances in Robotics, Vol. 1*, Springer, Cham (2020), pp. 267–279.
15. A. Asvadi, C. Premebida, P. Peixoto, and U. Nunes, *Rob. Auton. Syst.*, **83**, 299 (2016).
16. I. M. Gorovyi and D. S. Sharapov, "Comparative analysis of convolutional neural networks and support vector machines for automatic target recognition," in: *IEEE Microwaves, Radar and Remote Sensing Symposium (MRRS) Proceedings* (2017), pp. 63–66.
17. S. Dai, M. Han, W. Xu, et al., "Soft edge smoothness prior for Alpha-channel super-resolution," in: *IEEE Conference on Computer Vision and Pattern Recognition, MN, USA* (2007), pp. 1–8.

18. J. P. Landers and A. De Mello, "Kernel Function" in D. Li (Ed.), *Encyclopedia of Microfluidics and Nanofluidics*, Springer, New York (2015), pp. 919–925.
19. F. Shen and J. Zhang, *Wuhan Uni. J. Nat. Sci.*, **16**, 475 (2011).
20. P. Deepak and S. Suresh, "Design and utilization of bounding box in human detection and activity identification," in: S. Satapathy, A. Govardhan, K. Raju, J. Mandal (Eds.), *Emerging ICT for Bridging the Future – Proceedings of the 49th Annual Convention of the Computer Society of India CSI*, Vol. 2, Springer, Cham (2015), pp. 59–70.
21. P. Zhang, S. Wang, Z. Wang, et al., "Improving vehicle detection in point cloud data with novel features," in: *IEEE International Conference on Mechatronics and Automation (ICMA) Proceedings, Changchun, China* (2018), pp. 2227–2231..
22. A. Asvadi, L. Garrote, C. Premebida, et al., "Real-time deep ConvNet-based vehicle detection using 3D-LIDAR Reflection Intensity Data" in: *ROBOT 2017: Third Iberian Robotics Conference Proceedings*, Springer, Cham (2018), pp. 475–486.
23. M. Ranzini, L. Lugli, F. Anelli, et al., *Front. Hum. Neurosci.*, **5**, 147 (2011).
24. M. Judith Leo and S. Suchitra, *Procedia Computer Sci.*, **143**, 619 (2018).