

RESEARCH

Open Access



Classifying 3D objects in LiDAR point clouds with a back-propagation neural network

Wei Song^{1*†} , Shuanghui Zou^{1†}, Yifei Tian^{2†}, Simon Fong^{2†} and Kyungeun Cho^{3†}

*Correspondence:

sw@ncut.edu.cn

[†]Wei Song, Shuanghui Zou, Yifei Tian, Simon Fong and Kyungeun Cho contributed equally to this work

¹ North China University of Technology, Beijing, China
Full list of author information is available at the end of the article

Abstract

Due to object recognition accuracy limitations, unmanned ground vehicles (UGVs) must perceive their environments for local path planning and object avoidance. To gather high-precision information about the UGV's surroundings, Light Detection and Ranging (LiDAR) is frequently used to collect large-scale point clouds. However, the complex spatial features of these clouds, such as being unstructured, diffuse, and disordered, make it difficult to segment and recognize individual objects. This paper therefore develops an object feature extraction and classification system that uses LiDAR point clouds to classify 3D objects in urban environments. After eliminating the ground points via a height threshold method, this describes the 3D objects in terms of their geometrical features, namely their volume, density, and eigenvalues. A back-propagation neural network (BPNN) model is trained (over the course of many iterations) to use these extracted features to classify objects into five types. During the training period, the parameters in each layer of the BPNN model are continually changed and modified via back-propagation using a non-linear sigmoid function. In the system, the object segmentation process supports obstacle detection for autonomous driving, and the object recognition method provides an environment perception function for terrain modeling. Our experimental results indicate that the object recognition accuracy achieve 91.5% in outdoor environment.

Keywords: 3D object recognition, Back-propagation neural network, Feature extraction, LiDAR point cloud

Introduction

Autonomous driving technologies enable motor vehicles to drive themselves safely and reliably, and are being widely researched for smart cities and urban services [1]. The ability to perceive their surroundings is essential for unmanned ground vehicles (UGVs) to achieve autonomous driving [2]. Autonomous UGVs need to obtain a large amount of accurate environmental data to support automatic object avoidance and local path planning [3].

Several types of environment sensors, such as fisheye, binocular, and depth cameras, are widely used to obtain real-time information about a vehicle's surroundings so it can be aware of its environment [4–6]. Compared with these, the clearest advantage of Light Detection and Ranging (LiDAR) is that it can rapidly collect high-precision, wide-range point clouds [7]. Classifying and recognizing the features of individual objects based on these point clouds is a crucial challenge, and involves exploiting their unique properties,

such as their non-uniform densities and non-structural distributions [8]. In traditional point cloud analysis methods, the accuracy and speed performance are interfered by the unorganized distribution features of LiDAR point cloud [9]. Besides, uniformity density is also a difficult bottleneck for computer to allocate memory in point cloud storage [10]. To analyze the types of obstacles found in outdoor scenes, highly efficient pre-process of point cloud is urgent before executing classification and recognition steps [11].

In object recognition domain, machine learning algorithms are widely used in sensor-based object classifying and recognizing to increase the accuracy rate [12]. Thus, this paper proposes an urban object feature extraction and classification method that uses a back-propagation neural network (BPNN) to partition the original LiDAR point cloud into individual objects. In most urban environments, objects are always perpendicular to the ground surface, a feature we exploit. After filtering out the ground points from the LiDAR data, the remaining non-ground points are projected onto the x - z plane and clustered into different objects. This plane is rasterized to divide it into neatly-arranged cells containing the corresponding scattered points. By grouping connected cells together, the disorganized 3D points are split into different objects with unique labels by an inverse mapping from the x - z plane. This way, the redundant iteration in object segmentation is avoided through in the proposed method. In addition, the objects in the training and testing datasets are manually labelled with their categories.

During training, the BPNN model receives the objects' features and their labels as input and the parameters of all neurons in all layers are adjusted by back-propagation [13]. After a large number of training iterations, the parameters (namely the weights and bias) have been optimized and training is considered to be complete. Our non-linear classifier's recognition accuracy is then evaluated on the testing dataset by comparing the predicted and true object categories. Here, we consider five object types, namely trees, bushes, pedestrians, poles, and walls. The proposed feature extraction and classification method can be utilized in most urban environments to support UGV decision-making and hence realize autonomous driving in unknown environments.

The remainder of this paper is organized as follows. "Related works" section gives an overview of related work. "Feature extraction and object classification system" section describes our object feature extraction and classification method, including how we extract basic object characteristics, train the BPNN model, and test it. "Experiments" section presents our experimental results. Finally, "Conclusions" section concludes the paper.

Related works

Accurate object recognition and classification information was crucial for UGVs to understand their environments and make driving decisions via automatic traversable road planning [14]. In outdoor environments, different obstacles had different spatial distributions (meaning, for example, different sizes, shapes, and topologies), which were important features for distinguishing object types [15]. To obtain accurate object recognition and classification results, UGVs sensed their surroundings by using LiDAR to collect 3D point clouds.

The most common urban environments UGVs encounter were streets, which include buildings and other infrastructure, trees, brushes, pedestrians, and vehicles [16]. Before

classifying these objects, semantic segmentation was applied as a pre-processing step to divide the full point cloud for the current scene into individual parts [17]. However, the disorganized feature distributions in LiDAR point clouds hindered the accuracy and speed of traditional segmentation methods [9].

Indexing and pooling methods were used prior to object segmentation to order the original point cloud, but the labelling process required a large number of iterations [18, 19]. Yang et al. [20] developed a semantic object registration system that calculates the parameters of vertical lines extracted from pole-like and planar objects in 3D point clouds. Based on both geometric and semantic constraints, the extracted vertical lines provided important information about the external environment. They were used to assess objects' characteristics, which were recorded in a hash table with several descriptive attributes. Based on the hash table, specific objects containing vertices, straight lines, and planes were perceived after several scans from different static views. As an initial step, before object classification and recognition, this LiDAR point cloud registration process provided UGVs with basic perception information. By storing the LiDAR point clouds in a structured way, it dealt with the dispersed and non-sequential nature of their spatial distributions. Broggi et al. [21] used a stereo camera to capture point clouds around an unmanned car, and then grouped them into clusters with a flood fill method. This method employed a linear Kalman filter to analyze the movement and orientation of obstacles, classifying them as either moving or stationary. Due to only considering neighboring points when analyzing the large-scale dataset, this approach required for rapid iteration and traversal.

Next, the point clouds were split into groups representing individual objects for categorization. One method of extracting the objects' characteristics involves analyzing the outlines, edges, and vertical spatial distribution of each group [22]. Geometric properties and density distributions could also be used as basic object features for classification. Zhao et al. [23] installed multiple sensors, including camera and LiDAR, on an unmanned vehicle to collect 2D images and 3D point clouds of its surroundings. Then, they segmented the ground points by extracting geometrical features from the 3D LiDAR points while simultaneously utilizing color and texture information captured by the cameras to categorize the objects in detail. To deal with incorrect results caused by measurement error, they employed a Markov random field model to reduce detection mistakes when combining the data from multiple sensors.

Recognition algorithms used these extracted basic features to classify objects into types, such as planes, poles, or balls [24]. Several categorization methods exploited specific descriptions to find certain object types in the scenes [25, 26]. Wang et al. [27] developed a 3D object matching system that used an octree structure to describe the spatial distributions of the unstructured point clouds representing each separate object. They applied principal component analysis algorithm to each octree node to compute the eigenvectors and eigenvalues based on an icosahedron model. This enables them to extract the objects' features so as to divide the objects into different categories. Zeng et al. [28] proposed a kind of novel multiscale 3D keypoint detection method by using the double Gaussian weighted dissimilarity measure. The shape index value and the Gaussian weighted value of each 3D point were computed to select the most suitable 3D multi-scale key points. In this method, key points of uniform distribution were obtained

with excellent anti-noise ability. But, the method cost lots of time consumption in the multi-scale key point selection process so that the algorithm is not time efficiency to support environment information for UGV's automatic driving.

Subdivision was essential for object recognition to implement local path planning without collisions [29]. Compressing adjacent points with the same distance made less accurate in different object features identifying. After identifying certain LiDAR point orders, the approach of Choe et al. [30] abstracted geometrical features from the points to classify them into four types, namely horizontal, slope, vertical, and scatter. They then clustered points of the same type based on a nearest neighbors rule and trained a Gaussian mixed model to estimate the confidence levels when classifying the objects into types such as buildings, trees, cars, and other urban objects. However, the limited coverage of 2D LiDAR meant only two sides of the vehicle were in the detection domain and the front was ignored.

Combining these techniques with machine learning algorithms had significantly improved the accuracy and speed of semantic perception in urban areas [31, 32]. Wang et al. [25] developed a real-time pedestrian perception and tracking method using LiDAR point clouds collected from moving UGVs. This method first projected the point cloud onto a horizontal plane and then divided the plane into regular grid of cells. It then clustered connected cells into groups, considers all points in a given group to belong to the same object, and used this to recognize and classify the object. This method used a support vector machine (SVM) to separate the points into clusters and classify the grouped points. To determine whether or not a cluster represents a pedestrian, all the points in the cluster were input to the trained SVM model. This approach therefore recognized pedestrians from LiDAR point clouds using a trained model.

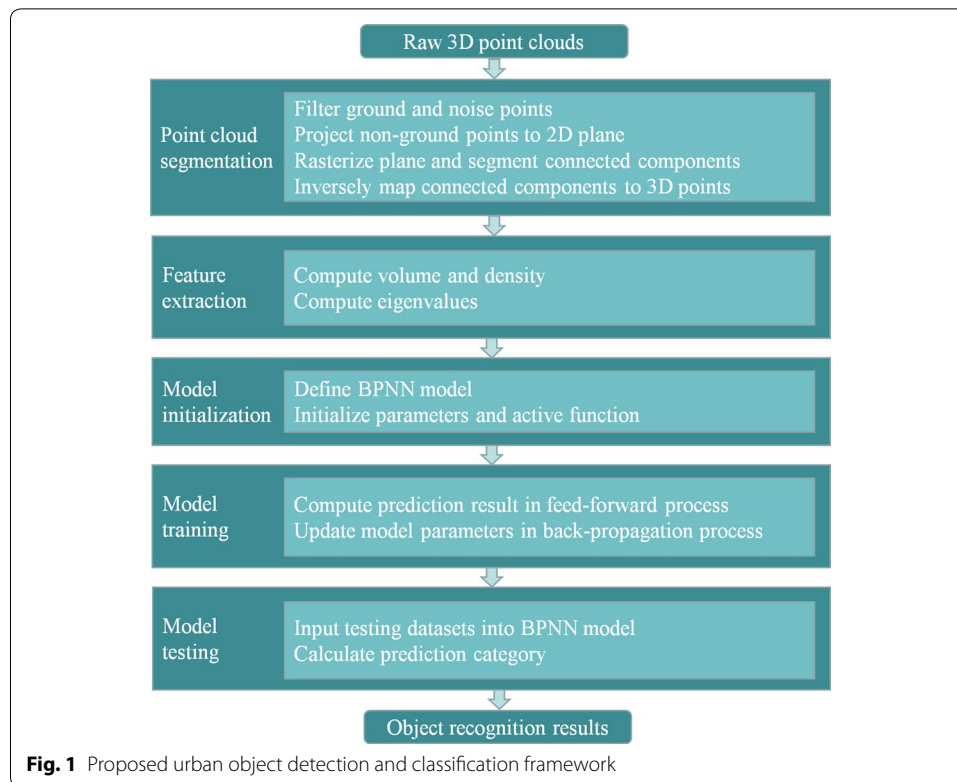
Zeng et al. [33] proposed a multi-feature fusion learning approach to 3D object recognition that used a convolutional neural network (CNN). This computed the heat and wave kernel signature descriptors to describe the 3D objects' shape geometry distributions, then used the CNN to learn fused features based on these descriptors to describe the objects' features. However, unlike with their high-resolution test model, the LiDAR point cloud's spatial distribution was so diffuse that it was difficult to describe the 3D surfaces accurately using these descriptors.

To improve object type recognition accuracy, this paper proposed an urban object feature extraction and classification method that identified different object categories from LiDAR point clouds. A BPNN model was trained to classify five common object types found in urban areas based on geometrical features extracted from segmented point clusters.

Feature extraction and object classification system

The proposed urban object feature extraction and classification method uses 3D LiDAR point clouds to enable dynamic environment perception for autonomous UGV decision-making. As illustrated by Fig. 1, our method consists of five steps, namely point cloud segmentation, feature extraction, model initialization, model training, and model testing.

To gain information about the road conditions in urban areas, the UGV utilizes high-precision LiDAR to generate raw 3D point clouds. Since most of the objects in urban



regions are perpendicular to the ground surface, segmenting them in the x - z plane is a reliable and reasonable approach. Here, the ground points form a connected plane, so all the objects would be recognized as a single connected component without ground filtering. Thus, we use a histogram-based threshold in the x - z plane to filter out the ground points [34]. To segment all the non-ground points into separate connected clusters, the projected points are rasterized into identically-sized square cells, and the cells are grouped into separate objects. Then, we apply an inverse projection to these clusters to form 3D objects with corresponding labels.

From the m -point sub-cloud D corresponding to a given object, we extract geometrical features including the volume, density, and eigenvalues in the three principal directions as a basis for classification. Here, the object's volume is computed by multiplying its length, width, and height together. The object's density is the quotient of its total effective point count and the effective count for the projected grid cells in the rasterized 2D plane, as illustrated above. The three eigenvalues are obtained by decomposing the point cloud's covariance matrix, providing estimates of the object's distribution in each dimension. Thus, by comparing these three eigenvalues, we can divide the objects into three different types based on their distributions.

In the object point cloud D , the values of each point in x , y , z coordinates are stored in the matrix X . This consists of n rows and m columns, where m is the number of 3D points in the object and n is the number of data dimensions, i.e., 3 (x , y , and z). To simplify the eigenvalue calculations, we normalize X to create X' by subtracting the mean values of the three coordinates.

Using the normalized matrix X' , we obtain the covariance matrix H as

$$H = \frac{1}{m} X' X'^T = \begin{Bmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{Bmatrix} \tag{1}$$

The diagonal elements of H are the variances of x , y , and z , and the other elements are the covariances. Because H is symmetric, the eigenvalues and eigenvectors can be calculated using the eigen decomposition method. The three resulting pairs of eigenvectors and eigenvalues represent the principal directions and the object's dimensions in these directions, respectively. The three eigenvalues thus roughly describe the object's point distribution, and are important features for object classification.

Next, we create a BPNN model that uses these three extracted features to recognize different object types. As illustrated in Fig. 2, the model has three layers, namely a 5-neuron input layer, a 20-neuron hidden layer, and a 5-neuron output layer. The BPNN model is trained via feed-forward and back-propagation steps.

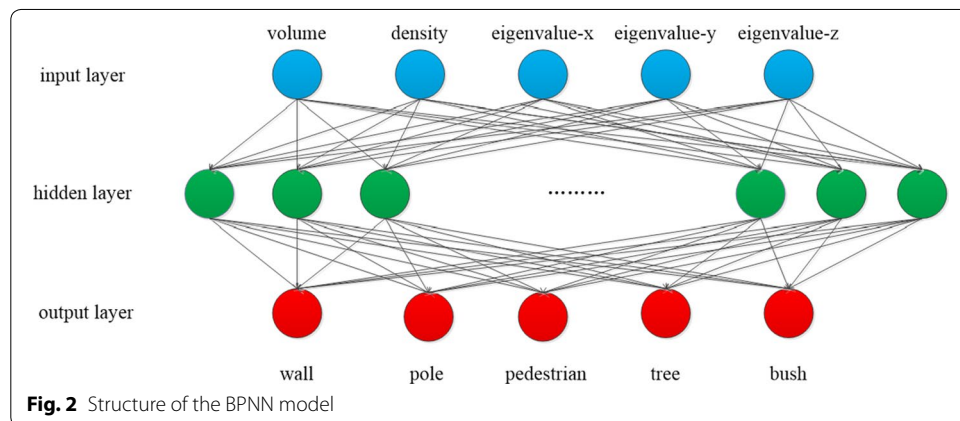
During the feed-forward step, all the hidden and output neurons x' are updated according to the weights w_i and values x_i of the neurons in the previous layer, as follows:

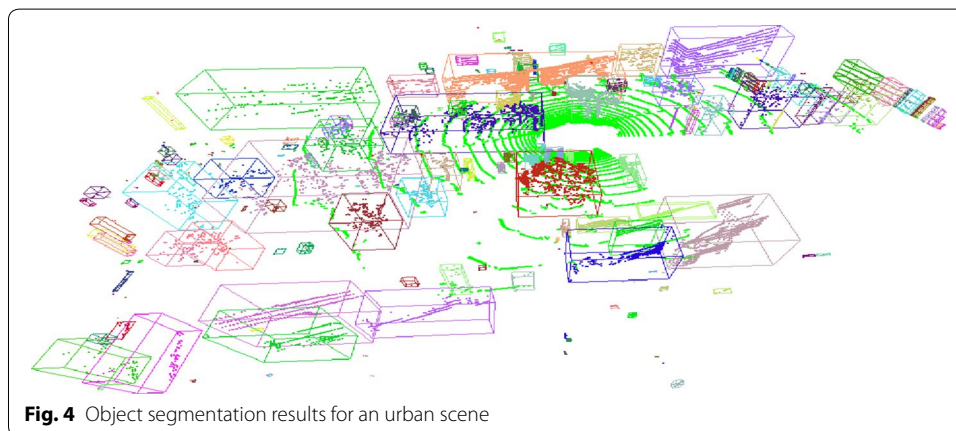
$$x' = \sum_{i=0}^n w_i x_i + b \tag{2}$$

$$y' = \frac{1}{1 + e^{-x'}} \tag{3}$$

where n is the number of neurons in the previous layer and b is the bias of neuron x' . We use a sigmoid activation function, obtaining the neuron's output y' . Next, the BPNN model's output vector Y' ($y' \cdot Y'$) as the prediction is computed using Eqs. (2) and (3). Here, the model has five output neurons, so the vectors Y and Y' are five dimension vectors. The error is calculated by comparing the prediction Y' with the true output vector Y as follows:

$$E = \frac{1}{2} \sum (Y - Y')^2 \tag{4}$$



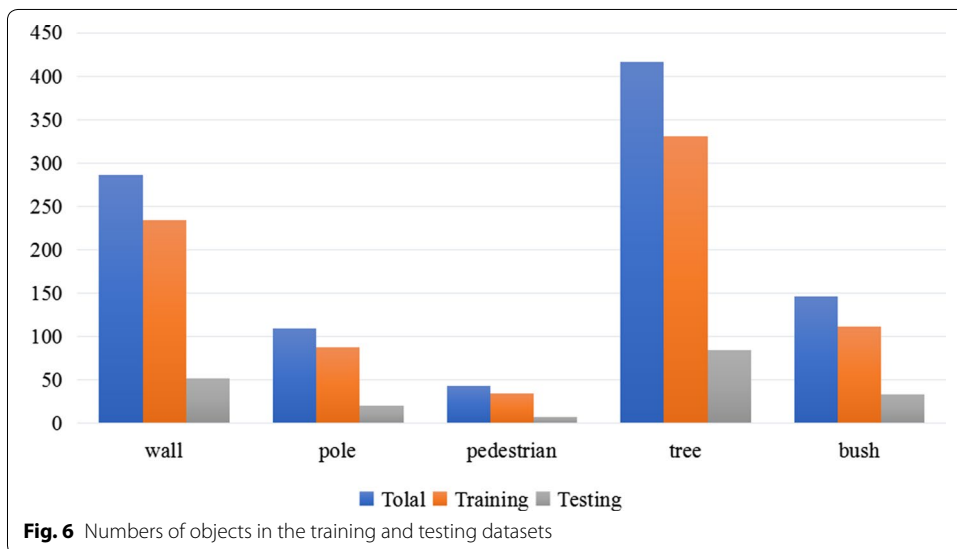
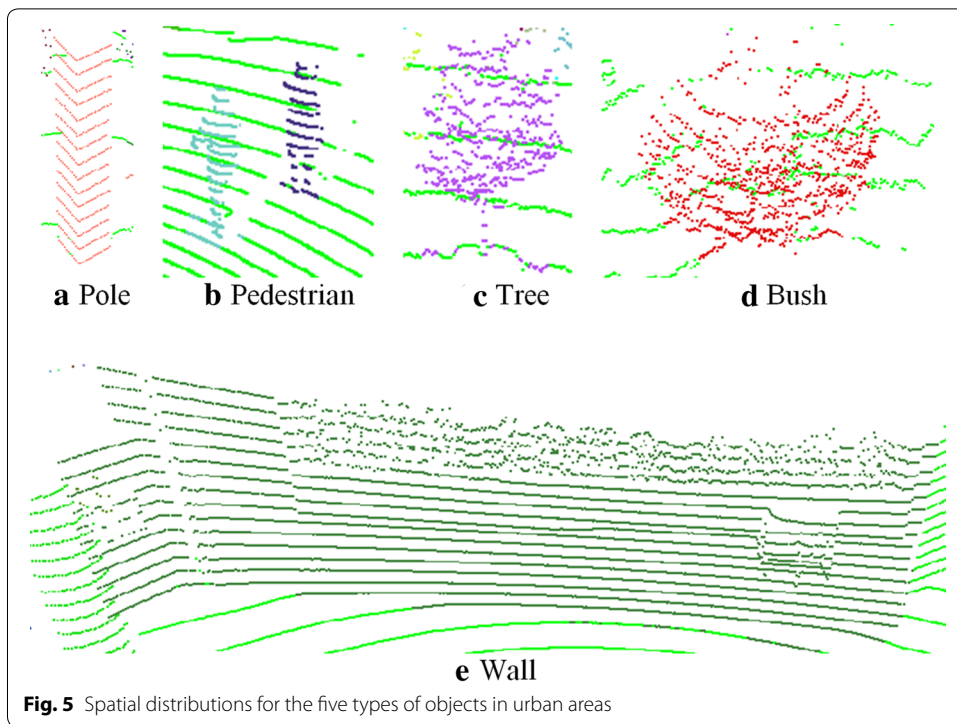


Back-propagation is then utilized to minimize the error by iteratively modifying the model's weight and bias parameters. Finally, after the weight and bias parameters have been optimized, the training process is complete. A testing dataset is then used to evaluate the model's object recognition accuracy using these basic geometric and spatial distribution features.

Experiments

Figure 3a shows our UGV, equipped with a roof-mounted LiDAR scanner to gather information about the urban environment within 70 m. The LiDAR device (Fig. 3b) collected about seven hundred thousand 3D points per second. The developed system ran on a PC with a 3.20 GHz Intel Core i5-5200 CPU @ 2.20 GHz, an NVIDIA GeForce GTX 770 GPU, and 4 GB RAM.

Figure 4 shows object segmentation results for 3D LiDAR point clouds recorded in urban environments, where different types of non-ground object, and their corresponding bounding boxes, are rendered in different colors. The objects in most urban environments typically consist largely of trees, pedestrians, walls, poles, and bushes. Thus, we aimed to classify the objects into these five types, as this is considered to be the main challenge for UGVs to sense their surroundings. Figure 5 illustrates the spatial distributions of point clouds corresponding to these five object types.



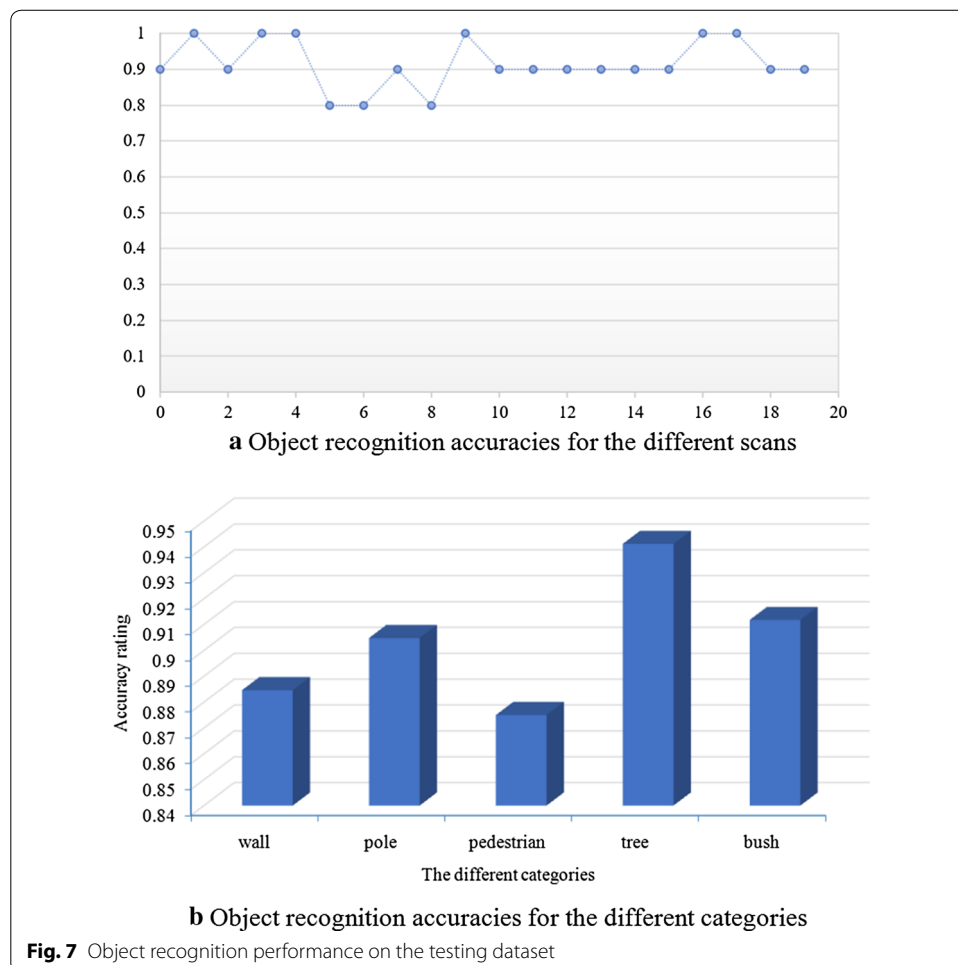
Based on the segmentation results, the object features, namely the volume, density, and eigenvalues in the three principal directions, were extracted by traversing all the object points. The training and testing datasets were created by collecting and manually labelling data for 1000 objects from 20 LiDAR scans taken in different urban environments. These consisted of 286 walls, 109 poles, 43 pedestrians, 416 trees, and 146 bushes. Of these, 800 objects were utilized for training and the other 200 for testing the BPNN model, distributed as shown in Fig. 6.

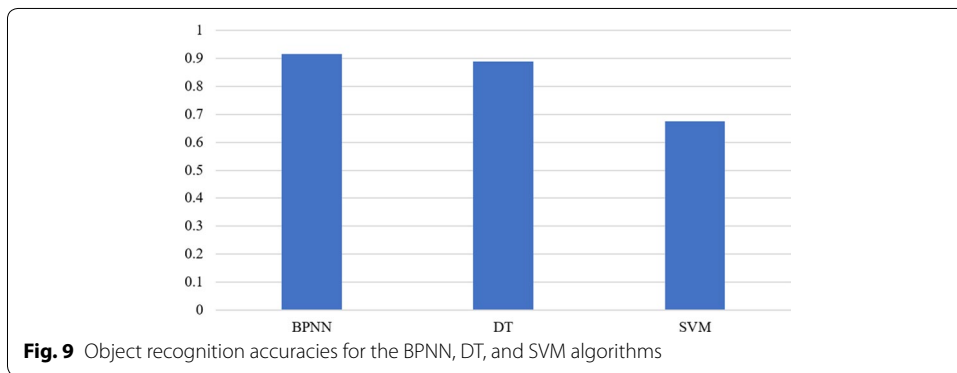
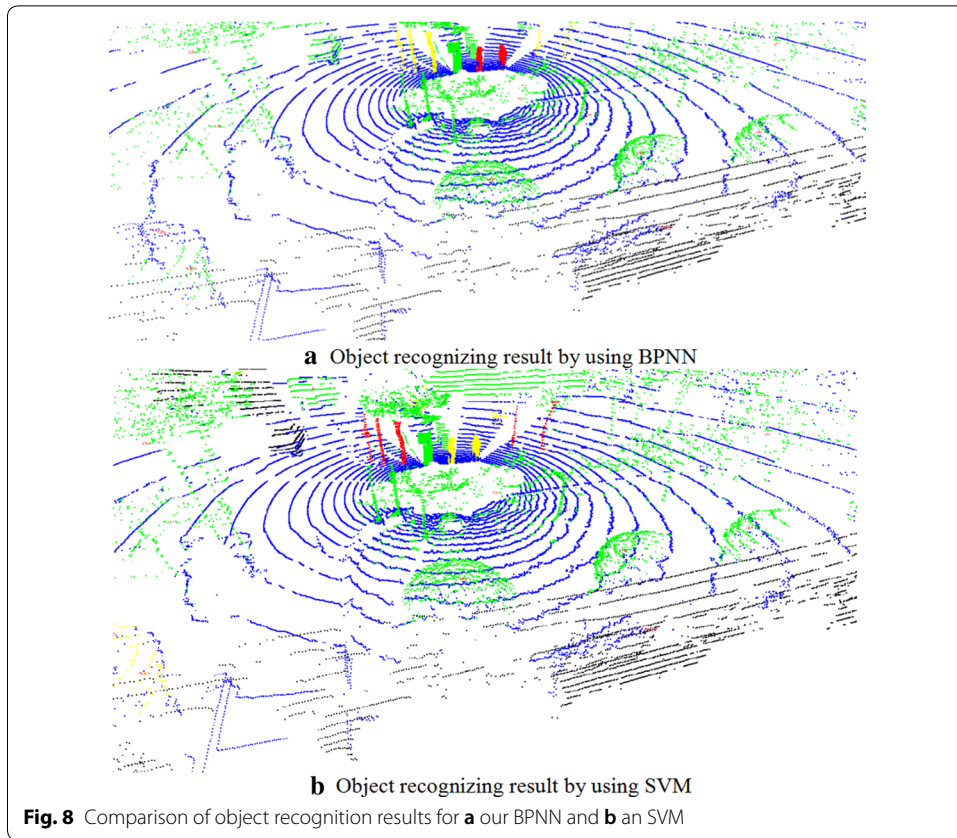
The BPNN training process was iterated 10,000 times. Figure 7a shows the trained BPNN’s object recognition accuracies for each of the 20 scans on the testing dataset; its average accuracy was 91.5%.

Figure 7b shows the BPNN’s performance for the different object types. Our datasets included many more trees and walls than pedestrians, so the model recognized trees more accurately than pedestrians. In addition, the simple and similar structures of poles meant they were identified relatively accurately despite the low number of training examples.

Figure 8 compares the object recognition results of our BPNN and an SVM. Here, poles, pedestrians, vegetation, and walls are supposed to be render in yellow, red, green, and black, respectively. Without the benefit of back-propagation, the SVM (Fig. 8b) found it hard to classify the pedestrian and vegetation objects, as their surface shapes were complex and variable.

For comparison, we also evaluated the performance of Decision Tree (DT) and Support Vector Machine (SVM) algorithms on our experimental datasets. Figure 9 compares the object recognition accuracies of the three methods (BPNN, DT, and SVM). This shows that our BPNN algorithm was the most accurate, achieving an average





accuracy of about 91.5%, compared with 89% and 67.5% for the DT and SVM algorithms, respectively.

Conclusions

In this paper, we have developed an object feature extraction and classification system that enables UGVs to analyze and perceive their environment. Our system uses LiDAR to scan the UGV's surroundings and gather information about the urban environment. Then, it segments the 3D point cloud data and extracts geometrical and distribution-related features (the volume, density, and eigenvalues in three principal directions) from the resulting object clouds. Training data that combines these features with manual

labels is used to train a BPNN to recognize five types of outdoor objects, namely walls, poles, pedestrians, trees, and bushes. The BPNN model has 5 input neurons, 20 hidden-layer neurons, and 5 output neurons. After 10,000 training iterations, the model's object classification accuracy averaged 91.5% on our testing dataset. In comparison, DT and SVM algorithms yielded accuracies of 89% and 67.5% respectively, indicating that our BPNN was more suitable for the UGV object classification task. In future work, we plan to increase recognition accuracy by gathering more manually-labelled object feature datasets to better train our BPNN model.

Authors' contributions

WS and KC described the proposed algorithms and wrote the whole manuscript. YT and SZ implemented the experiments. SF revised the manuscript. All authors read and approved the final manuscript.

Author details

¹ North China University of Technology, Beijing, China. ² Dept. Computer and Information Science, University of Macau, Macau, China. ³ Dept. Multimedia Engineering, Dongguk University, Seoul, South Korea.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (61503005), NCUT "The Belt and Road" Talent Training Base Project, NCUT "Yuyou" Project, the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2013-1-00684) supervised by the IITP (Institute for Information & communications Technology Promotion) and by Beijing New Star Project of Interdisciplinary Science and Technology (XXJC201709).

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The individual data are available in the archives of the laboratory and can be obtained from the corresponding author on request.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 16 June 2018 Accepted: 24 September 2018

Published online: 12 October 2018

References

1. Tirkolaee EB, Hosseinabadi AAR, Soltani M, Sangaiah AK, Wang J (2018) A hybrid genetic algorithm for multi-trip green capacitated arc routing problem in the scope of urban services. *Sustainability* 10:5
2. Zhong Z, Lei M, Cao D, Fan J, Li S (2017) Class-specific object proposals re-ranking for object detection in automatic driving. *Neurocomputing* 242:187–194
3. Vaquero V, Pino I D, Moreno-Noguer F, Solà J, Sanfeliu A, Andrade-Cetto J (2017) Deconvolutional networks for point-cloud vehicle detection and tracking in driving scenarios. In: *European conference on mobile robots (ECMR)*. pp 1–7
4. Häne C, Heng L, Lee GH, Fraundorfer F, Furgale P, Sattler T, Pollefeys M (2017) 3D visual perception for self-driving cars using a multi-camera system: calibration, mapping, localization, and object detection. *Image Vis Comput* 68:14–27
5. Cao L, Wang C, Li J (2015) Robust depth-based object tracking from a moving binocular camera. *Signal Process* 112:154–161
6. Deng T, Cai J, Cham TJ, Zhen J (2017) Multiple consumer-grade depth camera registration using everyday objects. *Image Vis Comput* 62:1–7
7. Vieira AW, Drews PLJ, Campos MFM (2014) Spatial density patterns for efficient change detection in 3D environment for autonomous surveillance robots. *IEEE Trans Autom Sci Eng* 11(3):766–774
8. Kang Z, Yang J (2018) A probabilistic graphical model for the classification of mobile LiDAR point clouds. *ISPRS J Photogramm Remote Sens.* <https://doi.org/10.1016/j.isprsjprs.2018.04.018>
9. Hao W, Wang Y (2016) Structure-based object detection from scene point clouds. *Neurocomputing* 191:148–160
10. Khatamian A, Arabnia HR (2016) Survey on 3D surface reconstruction. *J Inf Process Syst* 12(3):338–357
11. Yoo H, Son J, Ham B, Sohn K (2016) Real-time rear object detection using reliable disparity for driver assistance. *Expert Syst Appl* 56:186–196
12. Simon F, Song W, Kyungeun C, Raymond W, Kelvin K (2017) Training classifiers with shadow features for sensor-based human activity recognition. *Sensors* 17(3):476
13. Li D, Wu G, Zhao J, Niu W, Liu Q (2017) Wireless channel identification algorithm based on feature extraction and bp neural network. *J Inf Process Syst* 13(1):141–151

14. Himmelsbach M, Luettel, T, Wuensche HJ (2009) Real-time object classification in 3D point clouds using point feature histograms. In: the 2009 IEEE/RSJ international conference on intelligent robots and systems. pp 994–1000
15. Yang B, Dong Z, Zhao G, Dai W (2015) Hierarchical extraction of urban objects from mobile laser scanning data. *ISPRS J Photogramm Remote Sens* 99:45–57
16. Kim P, Chen J, Cho YK (2018) SLAM-driven robotic mapping and registration of 3D point clouds. *Autom Constr* 89:38–48
17. Ma H, Xiong R, Wang Y, Kodagoda S, Shi L (2017) Towards open-set semantic labeling in 3D point clouds: analysis on the unknown class. *Neurocomputing* 275:1282–1294
18. Yousefhussein M, Kelbe DJ, lentilucci EJ, Salvaggio C (2018) A multi-scale fully convolutional network for semantic labeling of 3D point clouds. *ISPRS J Photogramm Remote Sens*. <https://doi.org/10.1016/j.isprsjprs.2018.03.018>
19. Cura R, Perret J, Paparoditis N (2017) A scalable and multi-purpose point cloud server (PCS) for easier and faster point cloud data management and processing. *ISPRS J Photogramm Remote Sens* 127:39–56
20. Yang B, Dong Z, Liang F, Liu Y (2016) Automatic registration of large-scale urban scene point clouds based on semantic feature points. *ISPRS J Photogramm Remote Sens* 113:43–58
21. Broggi A, Cattani S, Patander M, Sabbatelli M, Zani P (2013) A full-3D voxel-based dynamic obstacle detection for urban scenario using stereo vision. In: 16th international IEEE annual conference on intelligent transportation systems. pp 71–76
22. Hamraz H, Contreras MA, Zhang J (2017) Vertical stratification of forest canopy for segmentation of understory trees within small-footprint airborne LiDAR point clouds. *ISPRS J Photogramm Remote Sens* 130:385–392
23. Zhao G, Xiao X, Yuan J, Ng GW (2014) Fusion of 3D-LIDAR and camera data for scene parsing. *J Vis Commun Image R* 25:165–183
24. Li Q, Xiong R, Vidal-Calleja T (2017) A GMM based uncertainty model for point clouds registration. *Robot Auton Syst* 91:349–362
25. Wang H, Wang B, Liu B, Meng X, Yang G (2017) Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle. *Robot Auton Syst* 88:71–78
26. Koo KM, Cha EY (2017) Image recognition performance enhancements using image normalization. *Human-centric Comput Inf Sci* 7:33
27. Wang J, Lindenbergh R, Menenti M (2017) SigVox – A 3D feature matching algorithm for automatic street object recognition in mobile laser scanning point clouds. *ISPRS J Photogramm Remote Sens* 128:111–129
28. Zeng H, Wang H, Dong J (2016) Robust 3D keypoint detection method based on double Gaussian weighted dissimilarity measure. *Multimedia Tools Appl* 76(2):1–13
29. Hyo HJ, Kim S, Chung MJ (2012) Object and ground classification for a mobile robot in urban environment. In: 2012 12th international conference on control, automation and systems. pp 2068–2070
30. Choe Y, Ahn S, Chung MJ (2014) Online urban object recognition in point clouds using consecutive point information for urban robotic missions. *Robot Auton Syst* 62(8):1130–1152
31. Zeng D, Dai Y, Li F, Sherratt R, Wang J (2018) Adversarial learning for distant supervised relation extraction. *Comput Mater Contin* 55(1):121–136
32. Zhang R, Li G, Li M, Wang L (2018) Fusion of images and point clouds for the semantic segmentation of large-scale 3D scenes based on deep learning. *ISPRS J Photogramm Remote Sens*. <https://doi.org/10.1016/j.isprsjprs.2018.04.022>
33. Zeng H, Liu Y, Li S, Che J, Wang X (2018) Convolutional neural network based multi-feature fusion for non-rigid 3D model retrieval. *J Inf Process Syst* 14:178–192
34. Song W, Liu L, Tian Y, Sun G, Fong S, Cho K (2017) A 3D localisation method in indoor environments for virtual reality applications. *Human-centric Comput Inf Sci* 7(1):39

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
