Pointwise classification of mobile laser scanning point clouds of urban scenes using raw data

Qiujie Li,* Pengcheng Yuan, Yusen Lin, Yuekai Tong, and Xu Liu
Nanjing Forestry University, College of Mechanical and Electronic Engineering,
Xuanwu District, Nanjing, China

Abstract. Mobile laser scanning (MLS), which can quickly collect a high-resolution and high-precision point cloud of the surroundings of a vehicle, is an appealing technology for three-dimensional (3D) urban scene analysis. In this regard, the classification of MLS point clouds is a common and core task. We focus on pointwise classification, in which each individual point is categorized into a specific class by applying a binary classifier involving a set of local features derived from the neighborhoods of the point. To speed up the neighbor search and enhance feature distinctiveness for pointwise classification, we exploit the topological and semantic information in the raw data acquired by light detection and ranging (LiDAR) and recorded in scan order. First, a two-dimensional (2D) scan grid for data indexing is recovered, and the relative 3D coordinates with respect to the LiDAR position are calculated. Subsequently, a set of local features is extracted using an efficient neighbor search method with a low computational complexity independent of the number of points in a point cloud. These features are further merged to produce a variety of binary classifiers for specific classes via a GentleBoost supervised learning algorithm combining decision trees. The experimental results on the *Paris-rue-Cassette* database demonstrate that the proposed approach outperforms the state-of-the-art methods with a 10% improvement in the F_1 score, whereas it uses simpler geometric features derived from a spherical neighborhood with a radius of 0.5 m. [DOI: 10.1117/1.JRS.15.024523]

Keywords: classification; mobile laser scanning; point clouds; urban scene.

Paper 210233L received Apr. 20, 2021; accepted for publication Jun. 11, 2021; published online Jun. 29, 2021.

1 Introduction

With the development of light detection and ranging (LiDAR) technology, mobile laser scanning (MLS) systems, which deploy one or multiple LiDARs on a ground-based vehicle,[1] can quickly collect a high-resolution and high-precision point cloud of the surroundings of the vehicle and have gained increasing attention in three-dimensional (3D) urban scene analysis,[2] including urban 3D modeling[3] and automated urban driving.[4]

Classification of MLS point clouds, in which each point in an MLS point cloud is determined to belong to a specific class, e.g., ground,[5] road,[6] road markings,[7] vehicles,[8] power lines,[9] and street trees,[10,11] is a common and core task for various applications of 3D urban scene analysis.[12] Weinmann et al.[13] proposed a pointwise classification framework, whereby each individual point is classified by a binary classifier involving a set of local geometric features derived from the neighborhoods of the point. The framework does not require expert knowledge in the specific domain[12] and thus can be applied to label a variety of urban object classes. However, the main challenges of pointwise classification include a low distinctiveness of local geometric features and a high computational complexity of the neighbor search. The commonly used neighbor search approaches for MLS point clouds are based on the k-D tree algorithm, in which a k-dimensional index tree is constructed and the average complexity for the nearest-neighbor search is $\mathcal{O}(\log N)$ for a point cloud with N points. To enhance the discrimination of local low-level geometric features, multiple neighborhood scales[14–16] or a selected optimal neighborhood scale[13,17,18] are

*Address all correspondence to Qiujie Li, lqj@njfu.edu.cn

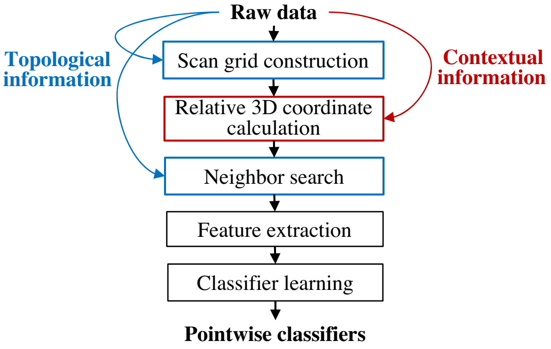


Fig. 1 Framework of the proposed pointwise classification approach.

recovered via the k-D tree, resulting in a higher computational cost. To address the above issues, Hackel et al.[14] downsampled the point cloud and built a multiscale pyramid of k-D trees to help improve the efficiency of neighbor searching.

An MLS point cloud is usually georeferenced by merging data from LiDAR and other sensors, such as inertial measurement units and global positioning systems.[1] However, more semantic information exists among the raw data acquired by the LiDAR, i.e., the relative positions of the measured points with respect to the LiDAR, which can be viewed as the relative positions of these points with respect to the road surface. This information can help to monitor the presence of most constructed objects since the spatial distributions of these objects along the road are generally regular. In addition, the raw data recorded in scan order are beneficial for organizing the point cloud.[19]

To reduce the neighbor search time and enhance the distinctness of local geometric features for pointwise classification, this study fully exploits the contextual and topological information in the raw data of the MLS point clouds, as shown in Fig. 1. First, a two-dimensional (2D) scan grid for data indexing is recovered, and the relative 3D coordinates with respect to the LiDAR position are calculated. Subsequently, a set of local features is extracted using an efficient neighbor search method with a low computational complexity independent of the number of points N in a point cloud. These features are further merged to produce a variety of binary classifiers for specific classes via a boosting supervised learning algorithm combining decision trees.

2 Methods

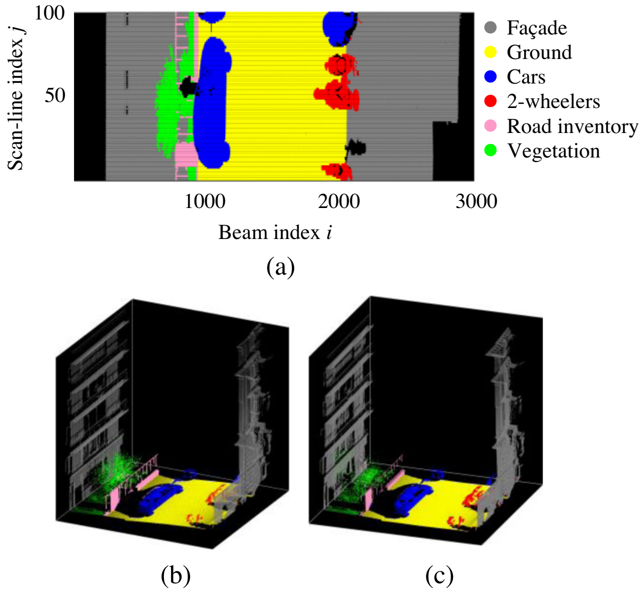
This study considers an MLS system with a single 2D LiDAR sensor used in push-broom mode;[20] i.e., the scan plane of the sensor is orthogonal to the direction of vehicle movement.

2.1 *Scan Grid Construction*

A scan line is defined as a 2D profile acquired by a single rotation of the 2D LiDAR mirror.[19,21] Thus, an MLS point cloud can be organized by constructing a scan grid in which each row represents a scan line, as shown in Fig. 2(a). In the grid, a point $\mathbf{p}$ measured by the i'th beam on the j'th scan line can be indexed by (i, j). The scan grid provides a compact representation of the point cloud with a size of $N_{\mathrm{sl}} \times N_b$, where N_{sl} is the number of scan lines and N_b is the number of laser beams per scan line.

To construct a scan grid from the MLS data, the scan angles of the point cloud should be recorded. Some LiDARs do not record the point if the laser beam does not return. In this case, we can segment the point cloud into scan lines to calculate the scan line index j for point $\mathbf{p}$ by finding a significant jump between the scan angles of adjacent points. Then, the beam index i for point $\mathbf{p}$ can be calculated as

$$i = \left[\frac{\theta - \theta_0}{\Delta\theta}\right] + 1, \tag{1}$$



(a)

(b)

(c)

Fig. 2 One hundred scan lines acquired by an MLS system with a single 2D LiDAR sensor, for which the scan plane is orthogonal to the direction of vehicle movement. The 2D LiDAR sensor performs a 360 deg scan at an interval of 0.12 deg, recording 3000 points per scan line. Points are colorized according to their class labels. (a) Scan grid, in which each row represents a scan line. The black points represent points in which the laser beam does not return. (b) Georeferenced 3D coordinates. (c) Relative 3D coordinates.

where θ is the scan angle of point $\mathbf{p}$, θ_0 is the start scan angle of a scan line, and $\Delta\theta$ is the scan resolution, i.e., the nominal angular increment between adjacent beams on a scan line.

If the scan resolution $\Delta\theta$ is not provided, it can be estimated using the mean or median of the differences between scan angles of adjacent points on the same scan line.

2.2 *Relative Coordinate Calculation*

To obtain more contextual information from LiDAR data, a relative 3D coordinate system, where x represents the moving distance from the origin along the trajectory of the LiDAR, y represents the horizontal displacement with respect to the LiDAR position, and z represents the vertical displacement with respect to the LiDAR position, is constructed for the MLS point cloud. The relative coordinates of point $\mathbf{p}(i,j)$ are calculated as

$$\begin{cases} x(i,j) = \sum_{k=1}^{j} v(k)\Delta t \\ y(i,j) = r(i,j)\cos\,\theta(i,j)\,, \\ z(i,j) = r(i,j)\sin\,\theta(i,j) \end{cases} \tag{2}$$

where $v(k)$ is the vehicle speed at the time when the k'th scan line is acquired by the LiDAR, Δt is the time interval between adjacent scan lines, $r(i,j)$ is the radial distance of point $\mathbf{p}(i,j)$, and $\theta(i,j)$ is the scan angle of point $\mathbf{p}(i,j)$.

Since the attitude of the LiDAR (vehicle) over a short period of time can be regarded as constant, the relative coordinates can describe a local spatial distribution of points as well as the georeferenced coordinates, as shown in Figs. 2(b) and 2(c).

2.3 *Neighbor Search*

The neighborhood of each point should be recovered for local feature extraction. A spherical neighborhood $\mathbf{S}(i,j)$ of point $\mathbf{p}(i,j)$ is defined as a set of points within a sphere centered at $\mathbf{p}(i,j)$ with a radius of δ. We propose a fast procedure for searching $\mathbf{S}(i,j)$ with a computational complexity of $\mathcal{O}(1)$, i.e., independent of the number of points N in a point cloud.

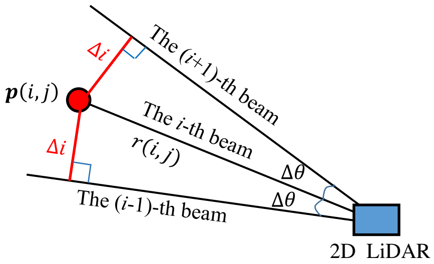


Fig. 3 Resolution Δi at point $p(i,j)$, i.e., the minimum distance between point $p(i,j)$ and its adjacent points on the j'th scan line.

Consider the measurement resolutions Δi and Δj of the scan grid at point $\mathbf{p}(i,j)$. The resolution Δi, the minimum distance between the point $\mathbf{p}(i,j)$ and its adjacent points $\mathbf{p}(i \pm 1, j)$ on the j'th scan line, can be estimated by the distance between the point $\mathbf{p}(i,j)$ and the $(i \pm 1)$'th beams on the j'th scan line, as shown in Fig. 3. The resolution Δj, the distance between the point $\mathbf{p}(i,j)$ and its adjacent points $\mathbf{p}(i, j \pm 1)$, can be estimated by the distance between the j'th scan line and the $(j \pm 1)$'th scan lines along the trajectory of the LiDAR. Thus, the measurement resolutions Δi and Δj are calculated as

$$\begin{cases} \Delta i = r(i,j)\sin\,\Delta\theta \\ \Delta j = v(j)\Delta t \end{cases}. \tag{3}$$

Then, a candidate neighborhood $\mathbf{R}(i,j)$ can be derived from the scan grid:

$$\mathbf{R}(i,j) = \left\{\mathbf{p}(i_R, j_R)|i_R \in \left[i - \frac{\delta}{\Delta i}, i + \frac{\delta}{\Delta i}\right], j_R \in \left[j - \frac{\delta}{\Delta j}, j + \frac{\delta}{\Delta j}\right]\right\}. \tag{4}$$

The spherical neighborhood $\mathbf{S}(i,j)$ is finally determined as

$$\mathbf{S}(i,j) = \{\mathbf{p}(i_S, j_S)|\|\mathbf{p}(i_S, j_S) - \mathbf{p}(i,j)\| \le \delta, \mathbf{p}(i_S, j_S) \in \mathbf{R}(i,j)\}. \tag{5}$$

The computational complexity of the proposed neighbor search method can be measured by the number of points in $\mathbf{R}(i,j)$:

$$N_R(i,j) = \frac{4\delta^2}{\Delta i \Delta j} = \frac{4\delta^2}{r(i,j)\sin\,\Delta\theta v(j)\Delta t}, \tag{6}$$

which depends on the measurement resolutions of the scan grid at point $\mathbf{p}(i,j)$ and is independent of the number of points in the whole point cloud. Note that affected by the variation of point density, the computational complexity of the proposed method becomes high with a small range $r(i,j)$.

2.4 *Feature Extraction*

To demonstrate the effectiveness of the relative 3D coordinates, a set of intuitive geometric features are extracted from the spherical neighborhood with a $\delta = 0.5$ m radius, as shown in Table 1. The density feature is corrected with the measurement resolutions to eliminate the influences of varying vehicle speeds and measurement ranges:

$$d(i,j) = N_S(i,j)\Delta i \Delta j, \tag{7}$$

where $N_S(i,j)$ is the number of points in the spherical neighborhood.

In addition, radiometric and penetrating features are derived based on the intensities I and numbers of echoes n measured by the LiDAR, providing further distinctive properties not covered by geometric features.

Table 1 Local features extracted from the spherical neighborhood.

Symbol	Description
x, y, z, I, and n	Relative 3D coordinates, intensity, and number of echoes of the central point
μ_x, μ_y, μ_z, μ_I, and μ_n	Means within the neighborhood
σ_x, σ_y, σ_z, σ_I, and σ_n	Standard deviations within the neighborhood
Δ_x, Δ_y, Δ_z, Δ_I, and Δ_n	Ranges within the neighborhood
L_λ, P_λ, S_λ, and O_λ	Shape features within the neighborhood[13]
d	Density corrected with measurement resolutions

2.5 *Classifier Learning*

The local features should be merged into a series of binary classifiers for a variety of specific classes by using supervised learning algorithms. The decision tree adopts a divide-and-conquer strategy by partitioning the feature space into subregions with a high class purity on the training set.[22] A top-down tree structure is constructed and each node chooses the best feature from the feature set to split the training data. Since a single decision tree has a very low bias and extremely high variance, we use the boosting framework to ensemble multiple decision trees to improve the classification accuracy and generalization ability.

We chose the GentleBoost algorithm, in which the total exponential loss on the training set is minimized using a functional Newton-like numerical optimization method.[23] The ensemble classifier is

$$F = \sum_{m=1}^{M} T_m, \tag{8}$$

where T_m is a decision tree generated as an incremental function in the m'th iteration and M is the number of iterations.

3 Results

3.1 *Dataset*

To compare our method with prior state-of-the-art methods, we use a publicly available and labeled database, namely *Paris-rue-Cassette*.[20] It contains 12 million points recorded on a street section in Paris with a length of ~200 m. A 2D LiDAR sweeps from −180 deg to 180 deg with $\Delta t = 10$ ms time interval. The starting scan angle θ_0 is −180 deg, i.e., the upward direction. All coordinates are geo-referenced (E,N,U) in Lambert 93 and altitude IGN1969 (grid RAF09) reference system. For the points with laser beam return, the range r, scan angle θ, intensity I, number of echoes n, and georeferenced LiDAR position (x_0, y_0, z_0) are recorded in scan order in addition to the georeferenced coordinates. The object classes for the experiments include façade, ground, cars, two-wheelers, road inventory, pedestrians, and vegetation. See Table 2 for details.

3.2 *Recovery of the Scan Grid*

The point cloud was first divided into scan lines at the positions where the sign of the scan angle changed from positive to negative. Then, the angle resolution $\Delta\theta = 0.12$ deg was estimated by analyzing the distribution of the difference between scan angles of adjacent points on the same scan line. Finally, a scan grid with a size of 4642×3000 was constructed, as shown in Fig. 4. Given the real-time speeds v of LiDAR estimated by the georeferenced LiDAR positions, the relative 3D coordinates were computed using Eq. (2).

Table 2 Object classes in the *Paris-rue-Cassette* database.

Class name	Urban objects	No. of points
Façade	Outside of a building	7,027,016
Ground	Roads, sidewalks, curbs, and other ground	4,229,639
Car	Cars	368,271
Two wheelers	Bicycles and other two-wheeled vehicles	40,331
Road inventory	Bollards, lampposts, traffic signs, meters, grids, and other road objects	46,105
Pedestrian	Still pedestrians, walking pedestrians, standing pedestrians, and other pedestrian	23,999
Vegetation	Trees and potted plants	212,131
Total		11,947,492

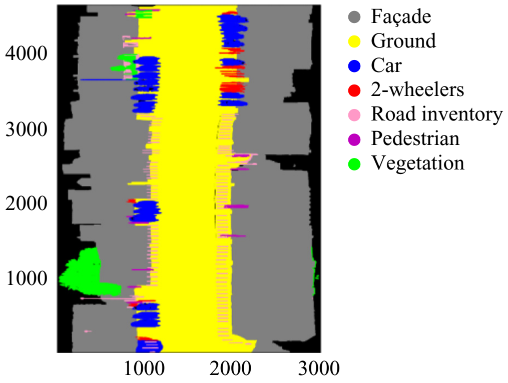


Fig. 4 Scan grid for the *Paris-rue-Cassette* database.

3.3 *Neighbor Search Efficiency*

To compare the computational complexities of the *k*-D tree algorithm and the proposed neighbor search method in searching spherical neighborhood, we run the two algorithms on a laptop PC with an AMD Ryzen 5 4600H CPU (hexa-core, 3.0 GHz) and 16 GB of RAM. The *k*-D tree algorithm was performed over the raw *Paris-rue-Cassette* dataset, while the proposed neighbor search method was performed over the scan grid of the *Paris-rue-Cassette* dataset. The Point Cloud Library was used for the *k*-D tree neighbor search. The search radius δ is set to 0.2, 0.5, and 0.8 m. As shown in Table 3, our method is much faster.

Table 3 Computational complexity of the neighbor search.

Paris-rue-Cassette		*k*-D tree	Proposed approach
Time of index creation		31.872 s	0.392 s
Time of neighbor search per 1000 pts	$\delta = 0.2$ m	1.944 s	0.010 s
	$\delta = 0.5$ m	3.314 s	0.041 s
	$\delta = 0.8$ m	5.568 s	0.088 s

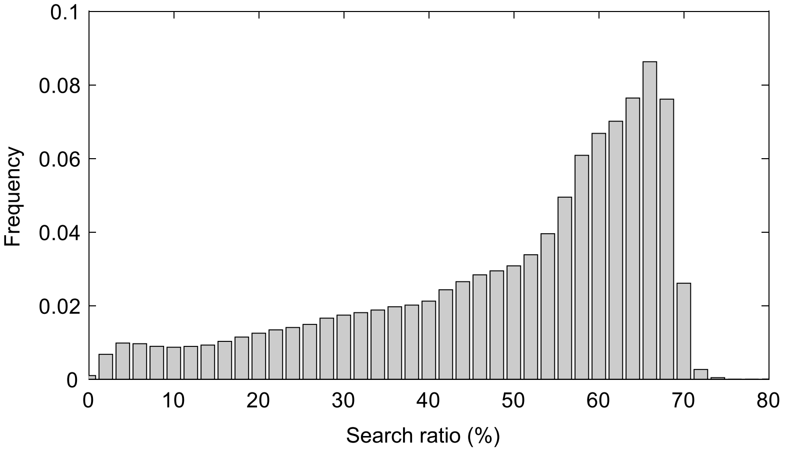


Fig. 5 Frequency distribution of search rate on *Paris-rue-Cassette*. The average search rate is 49.49%.

Figure 5 shows the frequency distribution of the search rate N_S/N_R on the *Paris-rue-Cassette* dataset with an average search rate of 49.49%, demonstrating the efficiency of the proposed neighbor search method.

3.4 *Effectiveness of the Relative 3D Coordinates*

To demonstrate the effectiveness of the relative 3D coordinates, we compare the distinctiveness of the geometric features derived from the georeferenced and relative 3D coordinates. The geometric features in Table 1 are divided into three kinds for comparison: (i) single coordinates of the central point; (b) basic features derived from the neighborhood using a single coordinate, i.e., the means, standards, and ranges within the neighborhood; and (c) shape features derived from the structure tensor of the neighborhood using three coordinates, i.e., the linearity L_λ, planarity P_λ, scattering S_λ, and omnivariance O_λ.

The Bayes error[22] for each feature is numerically estimated as follows:

$$e = \frac{1}{2}\sum_{n_h=1}^{N_h} \min[h^+(n_h), h^-(n_h)], \tag{9}$$

where $h^+(n_h)$ and $h^-(n_h)$ are the n_h'th bins of the probability histograms with $N_h = 100$ bins for a single feature, corresponding to an object class and a nonobject class, respectively. All features are mapped to the interval [0, 1].

The search radius δ is set to 0.5 m. Figure 6 shows the average Bayes error for each kind of geometric feature on the seven classes. Figures 6(a) and 6(b) show that single relative coordinates are more distinctive than single georeferenced coordinates, implying that the relative coordinates introduce more semantic information. Figure 6(c) shows that shape features using three relative coordinates have similar distinctiveness as those using three georeferenced coordinates, implying that the relative 3D coordinates can maintain the local spatial distribution of points as well as the georeferenced 3D coordinates.

3.5 *Pointwise Classification Accuracy*

The best results of pointwise classification we are aware of are those of Refs. 14, 24, and 25. In Ref. 25, a hierarchical framework composed of ground filtering, structural segmentation, and contextual classification was proposed. In Refs. 14 and 24, geometric features are derived at multiple scales or an optimal scale and then are combined into pointwise object classifiers. To improve the classification accuracy, statistical features derived from the 2D projection of

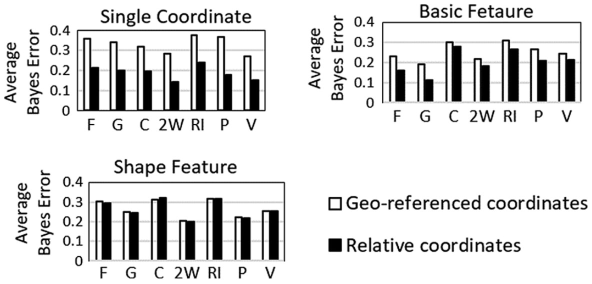


Fig. 6 Distinctiveness of the three kinds of geometric features using the georeferenced and relative coordinates, measured as the average of numerically estimated Bayes errors. (a) Single coordinates of the central point. (b) Basic features using a single coordinate, i.e., the means, standards, and ranges within the neighborhood. (c) Shape features L_λ, P_λ, S_λ, and O_λ using three coordinates. F, façade; G, ground; C, cars; 2W, two-wheelers; RI, road inventory; P, pedestrians; and V, vegetation.

the point cloud, the shape context 3D, and signature of histogram of orientations features are also utilized in Refs. 14 and 24 as well as fundamental geometric features.

The experiments use the same training and test sets as in Refs. 14 and 24; i.e., 1000 points per class are randomly selected as training samples and the remaining data are used as test samples. The number of iterations for GentleBoost is $M = 500$. Table 4 shows the classification results. Results of a variety of pointwise classification approaches are provided in Ref. 22, and the best result for each class is used for comparison. Compared with the prior state-of-the-art methods, our approach achieves an approximately 10% improvement in terms of the F_1 score. The improvement increased to 15% by adding radiometric and penetrating features mentioned in Sec. 2.4 into the classifiers.

The F_1 score depends on the decision threshold. When the decision threshold changes, the F_1 score changes. The area under the receiving operating characteristic (ROC) curve (AUC) does not depend on the decision threshold, so it is better than F_1 score to evaluate the performance of a classifier. Figure 7 shows the ROC curves for the proposed approach with all features. The AUC can summarize the relationship between the true- and false-positive rates of a binary classifier for different decision thresholds; hence, we also use the AUC to evaluate our approach in Table 4.

Table 4 Pointwise classification results.

				Proposed method			
	Hackel et al.[14]	Landrieu et al.[24]	Li et al.[25]	Geometric features		All features	
Class	F_1	F_1	F_1	F_1	AUC	F_1	AUC
F	0.9685	0.957	0.9860	0.9656	0.9923	0.9740	0.9941
G	0.9847	0.982	0.9815	0.9806	0.9986	0.9842	0.9987
C	0.8943	0.835	0.8401	0.8835	0.9936	0.9057	0.9951
2W	0.6784	0.667	0.5725	0.7723	0.9983	0.8594	0.9993
RI	0.3134	0.327	0.2540	0.5639	0.9917	0.6736	0.9932
P	0.3960	0.659	0.5454	0.7674	0.9984	0.8003	0.9988
V	0.6790	0.549	0.9099	0.7788	0.9951	0.8213	0.9963
Avg.	0.7020	0.711	0.7270	**0.8160**	0.9954	**0.8598**	0.9965

Note: F, façade; G, ground; C, cars; 2W, two-wheelers; RI, road inventory; P, pedestrians; V, vegetation. Bold values indicate our experimental results, which are better than the state-of-the-art (the second, third, and fourth columns).

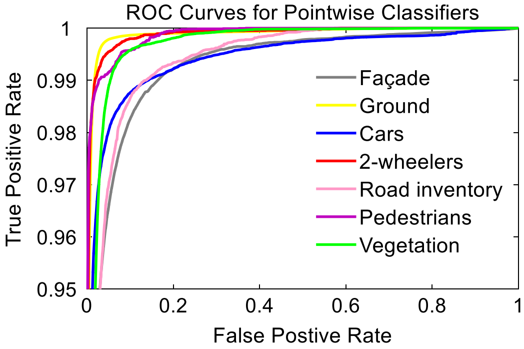


Fig. 7 ROC curves for the pointwise classifiers.

4 Discussion and Conclusion

This study aims to speed up the neighbor search and enhance feature distinctiveness for pointwise classification by exploiting topological and contextual information among raw data. Considering an MLS system with a single 2D LiDAR sensor, the cores of our approach are: (i) to construct a scan grid according to the scan pattern to organize an MLS point cloud; (ii) to compute the relative 3D coordinates with respect to the LiDAR position; and (iii) to recover the neighborhood by a fast search method. The computational complexity of the proposed neighbor search strategy is independent of the number of points in a point cloud with an average search rate of 49.49%. In terms of the Bayes error, geometric features using the relative coordinates are more distinctive than these features using georeferenced coordinates. Compared with the state-of-the-art methods, the proposed pointwise classification achieves an approximately 10% improvement in terms of the F_1 score, whereas it uses simpler geometric features derived for a search radius of 0.5 m. Furthermore, our approach is straightforward to parallelize and would be faster when taking advantage of parallel programming.

The proposed approach has several limitations: (1) the proposed neighbor search approach only works on an MLS system with a single 2D LiDAR sensor used in push-broom mode; (ii) to construct or recover a scan grid from the MLS data, the scan angles of the point cloud should be recorded, and the MLS data needs to be recorded in scan order or time stamps of the MLS data are recorded; (iii) because the proposed approach uses local features derived from several adjacent scan lines, it will not work well if the vehicle drives have multiple drive-runs in different directions (e.g., drive forward and backward along the road).

In future work, the proposed approach will be extended to MLS systems with a single 3D LiDAR or multiple 2D/3D LiDARs by exploring the measurement geometry of 3D LiDARs as well as the spatial relationship of multiple LiDARs. Furthermore, postprocessing, such as soft labeling,[24] will be considered to improve the classification accuracy.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant No. 31901239.

References

1. I. Puente et al., “Land-based mobile laser scanning systems: a review,” *Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **XXXVIII-5/W12**, 163–168 (2011).
2. Y. Wang et al., “A survey of mobile laser scanning applications and key techniques over urban areas,” *Remote Sens.* **11**(13), 1540 (2019).
3. C. Wang et al., “Urban 3D modeling with mobile laser scanning: a review,” *Virtual Real. Intell. Hardware* **2**, 175–212 (2020).
4. R. W. Wolcott and R. M. Eustice, “Visual localization within LIDAR maps for automated urban driving,” in *IEEE Int. Conf. Intell. Robots and Syst.* (2014).

5. H. Zhao et al., "Ground surface recognition at voxel scale from mobile laser scanning data in urban environment," *IEEE Geosci. Remote Sens. Lett.* **17**(2), 317–321 (2020).
6. C. Ye et al., "Robust lane extraction from MLS point clouds towards HD maps especially in curve road," *IEEE Trans. Intell. Transp. Syst.*, 1–14 (2020).
7. Y. Li et al., "Localization and extraction of road poles in urban areas from mobile laser scanning data," *Remote Sens.* **11**(4), 401 (2019).
8. J. Zhang et al., "Vehicle tracking and speed estimation from roadside Lidar," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **13**, 5597–5608 (2020).
9. S. Xu and R. Wang, "Power line extraction from mobile LiDAR point clouds," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **12**, 734–743 (2019).
10. Y. Q. Li et al., "Street tree information extraction and dynamics analysis from mobile lidar point cloud," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **XLIII-B2-2020**, 271–277 (2020).
11. S. Xu et al., "Automatic extraction of street trees' nonphotosynthetic components from MLS data," *Int. J. Appl. Earth Obs. Geoinf.* **69**, 64–77 (2018).
12. E. Che, J. Jung, and M. J. Olsen, "Object recognition, segmentation, and classification of mobile laser scanning point clouds: a state of the art review," *Sensors* **19**(4), 810 (2019).
13. M. Weinmann et al., "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS J. Photogramm. Remote Sens.* **105**, 286–304 (2015).
14. T. Hackel, J. D. Wegner, and K. Schindler, "Fast semantic segmentation of 3d point clouds with strongly varying density," *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* **III-3**, 177–184 (2016).
15. M. Weinmann et al., "Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas," *Comput. Graphics* **49**, 47–57 (2015).
16. T. Hackel, J. D. Wegner, and K. Schindler, "Joint classification and contour extraction of large 3D point clouds," *ISPRS J. Photogramm. Remote Sens.* **130**, 231–245 (2017).
17. M. Weinmann et al., "Geometric features and their relevance for 3d point cloud classification," *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* **IV-1/W1**, 157–164 (2017).
18. J. Demantké et al., "Dimensionality based scale selection in 3d lidar point clouds," *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **XXXVIII-5/W12**, 97–102 (2011).
19. E. Che and M. J. Olsen, "An efficient framework for mobile lidar trajectory reconstruction and *Mo-norvana* segmentation," *Remote Sens.* **11**(7), 836 (2019).
20. B. Vallet et al., "TerraMobilita/iQmulus urban point cloud analysis benchmark," *Comput. Graphics* **49**, 126–133 (2015).
21. Y. Zhou et al., "A fast and accurate segmentation method for ordered LiDAR point cloud of large-scale scenes," *IEEE Geosci. Remote Sens. Lett.* **11**(11), 1981–1985 (2014).
22. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., Comput. Complex, John Wiley & Sons, New York (1998).
23. J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Ann. Stat.* **28**(2), 337–407 (2000).
24. L. Landrieu et al., "A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds," *ISPRS J. Photogramm. Remote Sens.* **132**, 102–118 (2017).
25. Y. Li, B. Wu, and X. Ge, "Structural segmentation and classification of mobile laser scanning point clouds with large variations in point density," *ISPRS J. Photogramm. Remote Sens.* **153**, 151–165 (2019).

Qiujie Li is an associate professor working at the College of Mechanical and Electronic Engineering, Nanjing Forestry University. Her research focuses on mobile laser scanning, point cloud processing, and forestry informatization.

Biographies of the other authors are not available.