A VOXEL-BASED FILTERING ALGORITHM FOR MOBILE LIDAR DATA

Hejuan Qin [1], Haiyan Guan [1,*], Yongtao Yu [2], and Liang Zhong [3]

[1] College of Remote Sensing and Geometics Engineering, Nanjing University of Information Science & Technology, Nanjing, 210044, China － *guanhy.nj@nuit.edu.cn , qhj20140826@outlook.com
[2] Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian, 223003, China － allennessy.yu@gmail.com
[3] Changjiang Spatial Information Technology Engineering Co., Ltd., Wuhan 410010, China －bluewind20000@163.com

KEY WORDS：Mobile LiDAR, filtering, voxelization, upward growing, curvature

Abstract:

This paper presents a stepwise voxel-based filtering algorithm for mobile LiDAR data. In the first step, to improve computational efficiency, mobile LiDAR points, in *xy*-plane, are first partitioned into a set of two-dimensional (2-D) blocks with a given block size, in each of which all laser points are further organized into an octree partition structure with a set of three-dimensional (3-D) voxels. Then, a voxel-based upward growing processing is performed to roughly separate terrain from non-terrain points with global and local terrain thresholds. In the second step, the extracted terrain points are refined by computing voxel curvatures. This voxel-based filtering algorithm is comprehensively discussed in the analyses of parameter sensitivity and overall performance. An experimental study performed on multiple point cloud samples, collected by different commercial mobile LiDAR systems, showed that the proposed algorithm provides a promising solution to terrain point extraction from mobile point clouds.

INTRODUCTION

Airborne Light Detection and Ranging (LiDAR) technology has been proven to be a useful tool for describing detailed digital terrain models (DTMs) with the advantages of no effects of relief displacement, penetration of vegetation, insensitivity to lighting conditions, large data coverage, etc. In many literatures, the existing DTM generation methods (also termed filtering) are classified into the following categories: slope-based filtering, linear prediction-based filtering, morphology-based filtering, statistics-based filtering, and other filtering methods.

Closely followed by airborne LiDAR, mobile LiDAR technologies have been rapidly developed. A mobile LiDAR system (i.e. a laser scanning system deployed on the top of a land-vased vehicle), can fully capture road environments along survey roads in a form of 3-D point clouds (Guan et al., 2016a; Yang and Dong, 2013). Compared to an airborne LiDAR system, mobile LiDAR acquires data at a much higher point density and more complete data coverage (Fang and Yang, 2013; Yang et al., 2010a; Guan et al., 2016b). Due to its “drive-by” data acquisition means, Mobile LiDAR technology has potentials to road-scene object inventory mapping, including any road-scene structure, road pavement, traffic signaling devices, buildings, trees, power-lines, etc. Most studies interpreted mobile LiDAR point clouds by developing different road detection/extraction algorithms, in terms of road geometric shape, the use of road geometric features and LiDAR data characteristics, data format, the use of classification methods, and eternal data sources.

Similar to airborne LiDAR point cloud processing, the primary task of mobile LiDAR point cloud processing is to extract terrain data from the primitive point clouds, that is, mobile LiDAR point cloud filtering. Usually, most filtering methods, which were originally developed for airborne LiDAR data, assume that the lowest point in a neighborhood is a terrain point. However, compared to the looking-down view patterns of airborne LiDAR systems, which are more likely to generate uniform point densities, mobile LiDAR systems with side view patterns collect very dense data close to the scanner path and less dense data farther away from the scanner path. Points belonging to road surface account for a great portion of the collected mobile LiDAR data. Thus, the established airborne LiDAR filtering algorithms are unsuitable for retrieving non-terrain points from mobile LiDAR data. Compared with airborne LiDAR, terrain extraction methods of mobile LiDAR point clouds are still under developing.

Most current mobile LiDAR filtering methods are based on the principles, theories, and assumptions of airborne LiDAR filtering methods. For example, by means of the widely used triangular irregular network (TIN) progressive filtering method, reference (Fang et al., 2015) extract terrain patch segmentation. The other TIN-based methods can be found in literature(Wei et al.,2014; Liu et al.,2015) ·Iterative processing is feasible for airborne LiDAR data because of relatively low point density. However, such iterative processing methods are impractical for mobile LiDAR data because their point density far away higher than airborne LiDAR’s.

In the light of the characteristics of mobile LiDAR data (e.g., high point cloud density (Shi et al., 2005; Andrés and Beatriz, 2013), side-view laser scanning mode (Yang et al., 2010a; Cabo et al., 2015), elevation difference of multiple echoes), many filtering algorithms were developed accordingly. Reference (Shi et al., 2005), based on point density information, transformed three-dimensional (3-D) mobile LiDAR data into two-dimensional (2-D) images, on which terrain features were extracted by a segmentation strategy. Yang et al., (2010a) and Cabo et al. (2015) proposed a scan-line based filtering method, which separated terrain and non-terrain points using elevation differences scan-line by scan-line. Wu et al., (2007) and Tian et al., (2013), by using grid structure, separated terrain from non-terrain points. Lu et al., (2014) proposed a mathematical morphology based method, where 3-D mobile LiDAR data were first interpolated into 2-D images, and then an open operation with different window sizes was performed on

https://doi.org/10.5194/isprs-archives-XLII-3-1433-2018 | © Authors 2018. CC BY 4.0 License.

mobile LiDAR data under different terrain conditions. However, the aforementioned methods might be feasible for relatively-flat urban areas, and is ineffective for high point-density and large survey-scene mobile LiDAR data. Thus, this study presented a voxel-based terrain extraction method, which contributes to the improvement of mobile LiDAR processing efficiency.

1. OUR METHOD

The proposed filtering algorithm is two-step strategy: (1) rough terrain points are determined from mobile LiDAR points using a voxel-based upward-growing method; (2) terrain points are refined by using voxel curvature computation. Before terrain filtering, a whole survey scene is first partitioned into a set of blocks, B_i (i = 0, 1, 2,…, N. N is the number of blocks) with a given data-block size Γ_b. The data division processing means contributes to computational efficiency improvement by making use of multi-thread and distributed computing methods. Each data block can be parallelly processed using the proposed filtering algorithm detailedly described in the following subsections.

1.1 Octree-based Voxelization

Although the whole scene is divided into a set of data blocks, each data block still contains a large volume of points due to the close-range scanning pattern of a mobile LiDAR system. Moreover, each data block contains different object types. If all points in each data block are calculated the geometrical parameters of scene objects, leading to unreliable and ambiguous description and representation of scene objects in complex environments.

An octree is a data structure represented by a tree, in which each branch contains eight nodes, and therefore is commonly used to partition 3-D space into voxels (Truong-Hong and Laefer, 2014; Su et al., 2016). In this work, the whole 3-D points are equally and recursively partitioned into eight voxels until reaching a predefined sub-division depth or all voxels containing less than the predefined point number. Note that the octree has been used for the storage, segmentation, and compression of huge 3-D point clouds (Wang and Tseng, 2011; Elseberg et al., 2013; Su et al., 2016), as well as visualization (Wurm et al., 2010). The input 3-D points are recursively partitioned into voxels with a given voxel size, v_p. The value of v_p is determined based on the average point density of the collected mobile LiDAR data. The selection of voxel size significantly influences computational complexity and segmentation performance. Vo et al. (2015) presented the formula of voxel size selection, and stated that voxel size is partially controlled by point density through a set of experiments performed on points with different point density. The partition process continues until termination criteria are satisfied. In this study, we choose two termination criteria – the residual threshold, r_{th}, and minimum voxel size, v_{min}. The residual threshold is used to adaptively split the octree according to local surface planarity; that is, larger voxels appear at smooth regions (e.g., terrain, traffic-sign plates, building facades, and roads), smaller voxels present at edges, undulate terrain, rough objects (e.g., trees, grass), corners, etc. The minimum voxel size is defined close to point density. Based on the given voxel size, Γ_v, a point cloud block is partitioned into a set of voxels, v_j, j = 1, 2, …, N_v, where N_v denotes the number of voxels.

1.2 Voxel-based Upward Growing

For each voxel, v_j, 26 neighbors (L_j, j = 1, 2, …, 26) are distributed along X-, Y-, and Z- directions. According to scanning characteristics of a mobile LiDAR system, objects above ground surfaces (e.g. traffic light poles, trees, etc.) contain a large number of laser points in the direction of Z- axis, leading to a clear profile representation of these objects. Starting with any a voxel, an upward growing algorithm is recursively performed; that is, voxel, v_j, grows up to its nine neighbors, L_1 - L_9 , which are located above the voxel; then, each neighbor continues to grow upward to its corresponding neighbors. The upward growing algorithm stops when no more voxel can be reached.

1.3 Rough Terrain Extraction

Normally, we assume that the lowest points in elevation has high possibility of terrain points. Through upward growing, voxels are grouped into a set of voxel clusters, C_k, k = 1, 2, …, N_c, where N_c denotes the number of voxel clusters. In a voxel cluster, the highest voxel, $v_{highest}$, in elevation is justify to determine whether the voxel cluster is labelled as "*terrain*" or "*non- terrain*" based on the following two parameters:
(1) Local elevation difference (e_{local}): which represents the difference between $v_{highest}$ and the minimum elevation of a voxel in a local neighborhood.
(2) Global elevation difference (e_{global}): which represents the difference between $v_{highest}$ and the minimum elevation of a voxel in the whole scene.

Accordingly, the following two criteria are used to determine whether the voxel cluster is labelled as "*terrain*" or "*non-terrain*":
(1) If e_{local} lies below a predefined local elevation threshold (h_e) and e_{global} is smaller than a given global elevation (h_g), the voxel cluster is labelled as "*terrain*".
(2) Otherwise, the voxel cluster is labelled as "*non-terrain*", and removed from the data.

The local elevation threshold, h_g, contributes to…. ; the global elevation threshold, h_e, helps…….

1.4 Voxel Curvature-based Terrain Refinement

The above steps can eliminate the majority of *non-terrain* points. To further remove non-terrain points from terrain points, we propose a voxel-based curvature-based terrain refinement strategy based on the assumption that a terrain surface is locally continuous. For each voxel, we calculate its curvature using all points within the voxel. Let $\boldsymbol{P_j} = \{\boldsymbol{p_i} \epsilon \boldsymbol{R}^3, \boldsymbol{i} = 0,1,2\ldots,\boldsymbol{m}\}$ denote a point cloud dataset for each voxel, v_j, where $\boldsymbol{p_i}$ represents a point and $\boldsymbol{m}$ is the number of points. Let $\overline{\boldsymbol{p}}_{\boldsymbol{j}}$ denote the centroid of each voxel, v_j. At the $\overline{\boldsymbol{p}}_{\boldsymbol{j}}$, a 3×3 covariance matrix is computed as:

$$C_i = \frac{1}{N_{P_i}} \sum_{p_i \epsilon N_{p_i}} (p_i - \overline{p_i})(p_i - \overline{p_i})^T \tag{1}$$

$$C_i \cdot e_i = \lambda_i \cdot e_i \tag{2}$$

Where C_i is the symmetric and semi-positive definite matrix; { e_1 , e_2 , e_3 } are the eigenvectors, { λ_1 , λ_2 ,

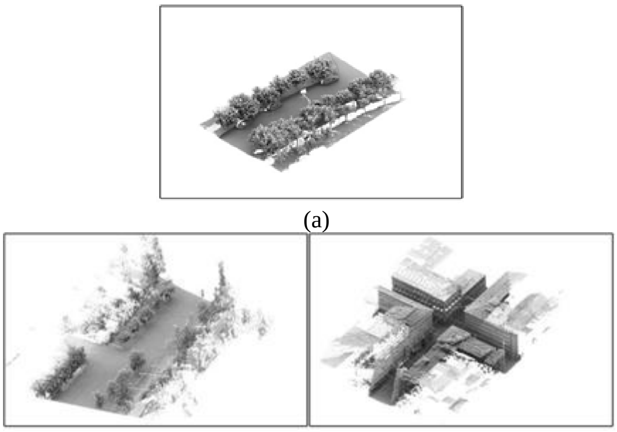
$\lambda_3\}(\lambda_1 \le \lambda_2 \le \lambda_3)$ are their corresponding eigenvalues. The curvature of a 3-D voxel is expressed as follows:

$$S_{p_j}=\frac{l_1}{l_1+l_2+l_3} \tag{3}$$

The value of $\sigma_{\boldsymbol{p}_j}$ reflects the deviation degree that represents the normal vector direction of the ground surface deviated from the tangent plane of the 3-D voxel. The larger the value of $\sigma_{\boldsymbol{p}_j}$, the greater the surface curvature change of the 3-D voxel, which represents a higher possibility of the existence of noise points in the 3-D voxel. The curvature change can eliminate discrete non-terrain points near-to /above the terrain surface, contributing to refinement of terrain extraction results.

2. EXPERIMENTAL DATASETS OF MOBILE LIDAR DATA

To verify the reliability of the proposed voxel-based terrain extraction algorithm, we conduct experiments on three experimental datasets collected by different mobile LiDAR systems. These datasets are detailed as follows.


(a)

(b) (c)

Fig. 3 Mobile LiDAR data samples: (a) MLSD-1, (b) MLSD-2, and (c) MLSD-3

Experimental dataset MLSD-1: the dataset was acquired along the urban roads in Xiamen City, China, by a RIEGL VMX-450 LiDAR system, which contains two RIEGL VQ-450 laser scanners. The specification of the RIEGL VMX-450 system claims that the system can achieve a maximum effective measurement rate of 1.1 million points per second and a scan speed of 400 lines per second. In this study, point density stands for the number of points per square meter and sharply drops perpendicular to the line of travel. With a vehicle driving speed of 30 km/h, the system collected ~7,000 points/m2 on the road surface within the range of 2.5 m, a much lower point density of 1,600 points/m2 on the pavements 20 m away from the scanning center. This complete survey was carried out once in a forward direction and once in a backward direction along the road, thus the collected data are an integration of the data collected from four scanners (Forward direction- two RIEGL VQ-450 scanners; backward direction- two RIEGL VQ-450 scanners). We selected a road section of about 48.57 meters, containing about 6.8 million points (see Fig. 3(a)).

Experimental dataset MLSD-2: the dataset was collected by a ROAMER mobile LiDAR system with a sampling frequency of 48 Hz. The average point density of this survey is around 200 points/m^2. The study area is a two-lane road section with a "T-type" crossing. We selected a road section of 71.28 meters, including 0.788 million points (see Fig. 3 (b)).

Experimental dataset MLSD-3: the dataset was collected by an Optech Lynx mobile LiDAR. This survey area is a typical urban street in Finland, Helsinki. Tall and large buildings are located along two sides of the surveyed street. We selected a road section of 131.69 meters, including 8.66 million points. The selected road section is featured by a crossroad, several tram tracks, and overhead trolley-bus wires. The average point density is 700 points/m^2 (see Fig. 3 (c)).

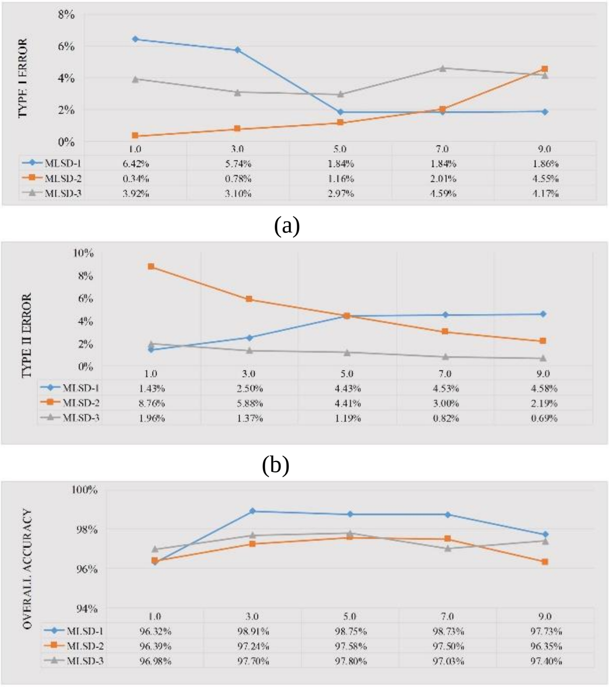
3. EXPERIMENTS AND ANALYSIS

To evaluate the performance of the proposed terrain extraction algorithm, two type errors were used: Type I and Type II errors. Type I error defines the percentage of terrain points incorrectly classified as non-terrain points; whereas Type II error defines the percentage of the non-terrain points are wrongly classified as terrain points. Type I error presents loss of accuracy and type II error presents deterioration of the quality of digital elevation mode. In addition, comprehensive accuracy refers to the percentage of the total number of terrain and non-terrain points and the total number of laser points in all the cloud data. Therefore, three indicators: Type I error, Type II error, and comprehensive accuracy are used for evaluating filter algorithms. The reference data used for precision evaluation of terrain and non-terrain points are mainly manually assisted marking. Experiments were tested on a personal computer with a 3.30-GHz Intel(R) Core(TM) i3-2120 central processing unit.

3.1 Parameter Sensitivity

The aforementioned three datasets are used to investigate the applicability of the proposed voxel-based terrain extraction method, in which the following five parameters are used: 2-D block size(Γ_b), 3-D voxel size (Γ_v), glocal elevation threshold (h_g), local elevation threshold (h_e), and voxel curvature (σ_{p_i}). The curvature histogram of all voxels suggests value of σ_{p_i} in the range of [0, 1/3]. In this section, we designed four groups of experiments to investigate the sensitivity of the proposed voxel-based terrain extraction method to the selection the size parameters Γ_b and Γ_v, as well as the elevation parameters h_g and h_e.

In the first group, we maintained Γ_v = 0.05 m, h_e = 0.5 m, and h_g = 4.5 m, and varied Γ_b from 1.0 m to 9.0 m with an interval of 2.0 m. Fig. 4 shows the experimental results for these three datasets. As shown in Figure 4, for MSLD-1, Type I error, Type II error, and overall accuracy dramatically vary with the parameter Γ_b increasing from 1.0 m to 3. 0 m, and tend to be stable as Γ_b changing from 5.0 m to 10.0 m. For MSLD-2, when Γ_b changes from 1.0 m to 10. 0 m, Type I error increases from 0.34% to 4.53%, whereas Type II error decreases from 8.76% to 2.18%. The overall accuracy slightly changes around 97%. Similarly, for MLSD-3, the optimal values of Type I error and overall accuracy, respectively, are 2.97% and 97.80% when Γ_b = 5.0. As such, in our study, the better terrain extraction results in Type I error, Type II error, and overall accuracy were obtained at Γ_b = 3.0 or 5.0.



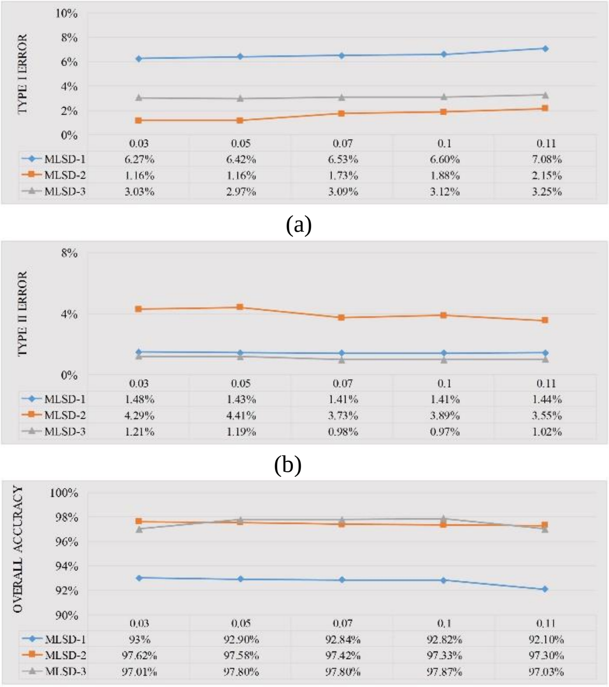
(a)

(b)

(c)

Fig.4 Sensitivity tests with the size of 2-D block (Γ_b): (a) Type I error, (b) Type II error, and (c) Overall accuracy

Next, we used Γ_b = 5.0 m，h_g = 0.5 m，and h_e = 4.5 m, and varied Γ_v from 3.0 cm to 12.0 cm with an interval of 2.0 cm. Fig.5 shows the results for these three datasets. As shown in Figure 5, with the parameter Γ_v from 3.0 cm to 12.0 cm, Type I errors for all three test datasets increase slightly and tend to be stable. For all datasets Type II error and overall accuracies tend to be stable with the increasing of Γ_v, and then decreases slightly. In this paper, the Γ_v value of 0.05 obtained the better terrain extraction performance.



(a)

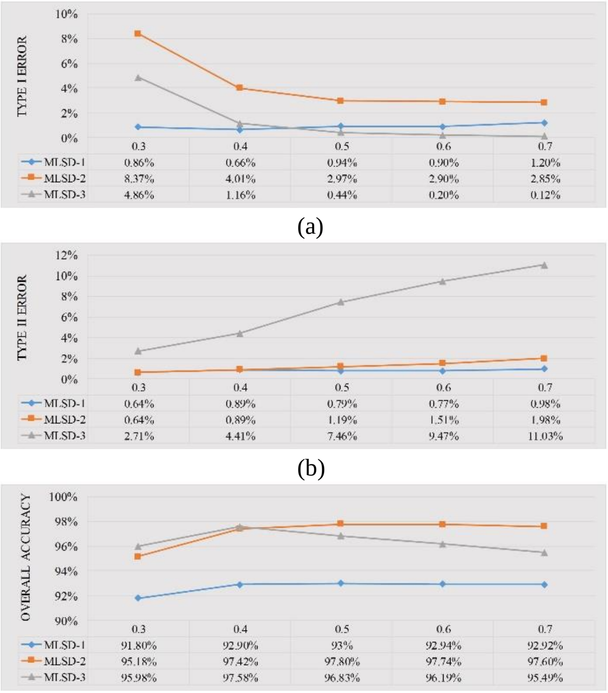
(b)

(c)

Fig.5 Sensitivity tests with the size of 3-D voxel (Γ_v): (a) Type I error, (b) Type II error, and (c) Overall accuracy

In the third group, we maintained Γ_b = 5.0 m, Γ_v = 5.0 cm, and h_e = 4.5 m and varied h_g from 0.3 m to 0.7 m with an interval of 0.1 m. Fig.6 shows the results for these three datasets. As shown in Fig. 6, Type I errors for all these three datasets gradually decreased with the increase of h_g. Type II errors for MLSD-1, -2, and -3 datasets tend to be stable. Type II errors for MLSD-3 greatly increase from 2.71% to 11.03% when the parameter h_g increases.

The main reason behind this phenomenon is that, for MSLD-3 covering a relatively flat urban area, an increasing value of h_g results in a large number of “non-terrain” points were misclassified as “terrain” points, leading to a rapid growth of Type II errors. As shown in Fig. 6, when h_g = 0.5 m, the terrain extraction results of all these four datasets exhibit good performance.



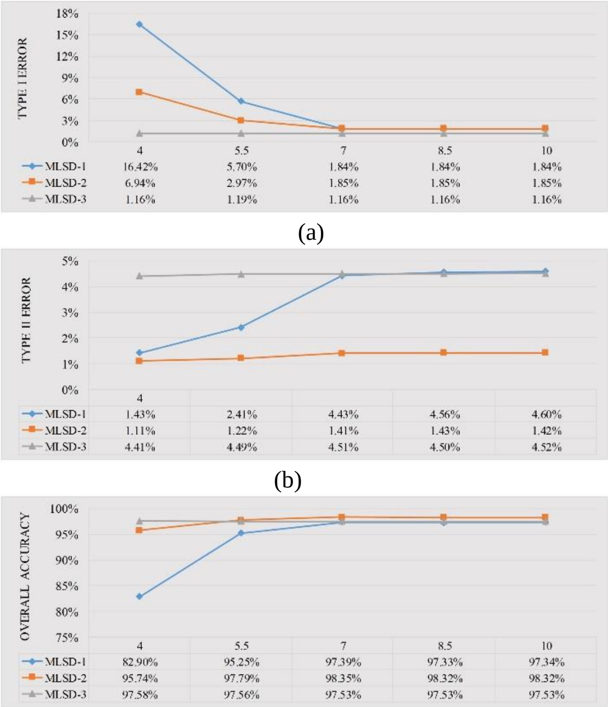
(a)

(b)

(c)

Fig. 6 Sensitivity tests with the global elevation threshold (h_g): (a) Type I error, (b) Type II error, and (c) Overall accuracy

Finally, we used Γ_b = 5.0 m, Γ_v = 5.0 cm, and h_g = 0.5 m and varied h_e from 4.0 m to 10.0 m with an interval of 1.5 m. Fig.7 shows the results for these three datasets. As shown in Fig. 7, For MSLD-1, Type I and II errors greatly decrease when the parameter h_e increases; conversely, overall accuracy, for MSLD-1, greatly increases when h_e changes from 4.0 to 7.0 m and tends to be stable since h_e = 7.0 m. For MSLD-2 and 3, Type I error, and Type II error, and overall accuracy tend to be stable. The reason behind this phenomenon is that h_e mainly controls terrain fluctuation in the whole scene of 3-D point clouds, which can maximally guarantee our method can deal with the scenes with varying degrees of terrain relief. In this study, the best terrain extraction performance obtained at h_e = 5.0 m or 6.0 m.



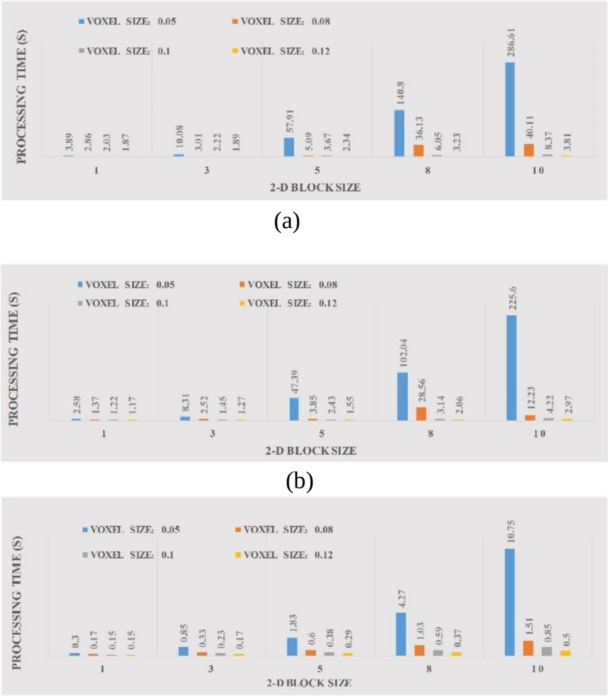
(a)

(b)

(c)

Fig. 7 S Sensitivity tests with the local elevation threshold (h_e): (a) Type I error, (b) Type II error, and (c) Overall accuracy

3.2 Computational Complexity



(a)

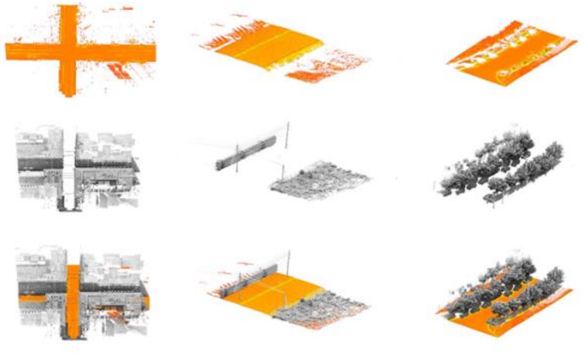
(b)

(c)

Fig. 8 The running time of our method with the sizes of 2-D block ($\mathit{\Gamma}_b$) and 3-D voxel ($\mathit{\Gamma}_v$):
(a) MLSD-1, (b) MLSD-2, and (c) MLSD-3.

Five parameters were used for terrain extraction from mobile LiDAR data. As aforementioned parameter sensitivity discussion, two parameters, namely, 2-D block size and 3-D voxel size, have great influence on terrain extraction efficiency from voluminous mobile LiDAR points. The processing parameters are hg = 0.5 m and he = 5.0 m. Therefore, we performed time complexity analysis by varying 2-D block size from 1.0 to 10.0 m with five different size settings (namely, 1.0 m, 3.0 m, 5.0 m, 8.0 m, and 10.0 m) and 3-D voxel size from 0.05 m to 0.12 m with four different size settings (namely, 0.05 m, 0.08 m, 0.10 m, and 0.12 m). Fig.8 shows the running time results for these three datasets. As shown in Fig.8, the runtime greatly grows as the 2-D block size increases at any a fixed 3-D voxel size. This is because the larger the 2-D block size, the more points to be processed. On the contrary, the runtime decrease as the 3-D voxel size increases at any a fixed 2-D block size. The main reason is that the smaller the 3-D voxel size, the more the number of the generated 3-D voxels, in each of which less points are included, leading to a decrease of the time complexity.

To qualitative access the proposed terrain extraction method, we used $\mathit{\Gamma}_b$ = 5.0 m, $\mathit{\Gamma}_v$ = 5.0 cm, h_g = 0.5 m, and h_e = 5.0 m based on the parameter sensitivity and time complexity analyses. Fig.9 shows the terrain extraction results from three different mobile LiDAR datasets. As shown in Fig.9, the separated “terrain” points and “non-terrain” are demonstrated at first and second rows, respectively; the third row shows the labeled “terrain” and “non-terrain” points.



(a) (b) (c)

Fig. 9 Sketch maps of the results of terrain extraction using space voxels: (a) MLSD-1, (b) MLSD-2, and (c) MLSD-3

4. CONCLUSION

We have proposed a novel method for extracting terrain points from mobile LiDAR point clouds. The presented terrain extraction method combined the following: 1) 2-D block segmentation, which c; 2) voxel-based terrain extraction using upward growing; 3) terrain point refinement. The performance of our terrain extraction method was validated by three different mobile LiDAR datasets acquired by RIEGL VMX450, ROAMER, and Lynx systems, respectively. Parameter sensitivity analyses showed that local and global terrain thresholds mainly control the quality of the extracted terrain; while 2-D block size and 3-D voxel size have a great impact on the efficiency of data processing. Qualitatively, our method demonstrated promising crack extraction performance. Our research on terrain extraction from mobile LiDAR data provides valuable insights into point-cloud processing at all levels of survey or survey-related agencies. We will make further efforts to extend our method to the applications in terrain extraction from all types of point clouds, such as airborne LiDAR points.

5. ACKNOWLEDGEMENT

This work was supported in part by the Natural Science Foundation of Jiangsu Province under Grant BK20151524, in part by the National Natural Science Foundation of China under Grant 41501501.

REFERENCE

Cabo, C., García Cortés, S., and Ordoñez, C., 2015. Mobile Laser Scanner data for automatic surface detection based on line arrangement. Automation in Construction.58,28-37.

Elseberg, J., Borrmann, D., Nuchter, A., 2013. One billion points in the cloud—an octree for efficient processing of 3-D laser scans. ISPRS Journal of Photogrammetry and Remote Sensing.76, 76–88.

Fang, H., Lin, X., Duan, M., Z., 2015a. Object-oriented filtering method for mobile LiDAR point clouds. Science of Surveying and Mapping.40(4),92﹣96.

Fang, L., Yang, B., 2013. Automated Extracting structural roads from mobile laser scanning point clouds. Acta Geodaetica et Cartographica Sinica. 42(2), 260-267.

Guan, H., Li, J., Cao, S., Yu, Y., 2016a. Use of mobile LiDAR in road information inventory: a review. International Journal of Image and Data Fusion. 7(3), 219–242.

Guan H., Yu, Y., Li, J., Ji, Z., Zhang Q., 2016a. Extraction of power transmission lines from vehicle-borne LiDAR data. International Journal of Remote Sensing. 37(1), 229-247.

Truong-Hong, L., Laefer, D. F., 2014. Octree-based, automatic building façade generation from LiDAR data, Computer-Aided Design.53, 46–61.

Liu, R., Tian, M., Xu, J., 2015. Expressway road surface point filtering for mobile laser scanning data. Geomatics and Information Science of Wuhan University.40(6),751﹣755.

Lu, X., Liu, R., Tian, M., Liu, B., 2014. Ground filtering of vehicle-borne laser point cloud based on adaptive mathematical morphology. Geomatics and Information Science of Wuhan University.39(5),514﹣518.

Serna Andrés and Marcotegui Beatriz., 2013. Urban accessibility diagnosis from mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing.84, 23-32.

Shi W., Li B., Li Q., 2005. A method for segmentation of range image captured by vehicle-borne laser scanning based on the density of projected points. Acta Geodaetica et Cartographica Sinica. 34(2):95-100.

Su, Y., Bethel, J., S. Hu, 2016. Octree-based segmentation for terrestrial LiDAR point cloud data in industrial applications, ISPRS Journal of Photogrammetry and Remote Sensing. 113, 59–74.

Tian, M.，Zhao, Q.，Liu, J.，Liu, Y., 2013. A filtering method of vehicle-borne laser scanning data. Journal of Geomatics Science and Technology.30(6),593﹣596.

Vo, A., Truong-Hong, L., Laefer, D. F., Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation, ISPRS Journal of Photogrammetry and Remote Sensing.104, 88–100.

Wang, M., Tseng, Y.-H., 2011. Incremental segmentation of LIDAR point clouds with an octree-structured voxel space. Photogrammetric Record.26 (133), 32–57.

Wei, S., Fang, H., Lin, X., 2014. Roads information extraction from mobile laser scanning point cloud based on prior knowledge. Science of Surveying and Mapping.39(10),81﹣84.

Wu, F., Li, Q., Xiong, Q., 2007a.Classification of vehicle-borne laser scanning data. Science of Surveying and Mapping. 32(4),75-79.

Wurm, K.M., Hornung, A., Bennewitz, M., Stachniss, C., Burgard, W., 2010, OctoMap: a probabilistic, flexible, and compact 3-D map representation for robotic systems. In: ICRA 2010 workshop best practice in 3-D perception and modeling for mobile manipulation. p. 8.

Yang, B., Dong, Z., 2013. A shape-based segmentation method for mobile laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing. 81, 19-30.

Yang, B., Wei, Z., Li, Q., Mao, Q., 2010a. A Classification-oriented Method of Feature Image Generation for Vehicle-borne Laser Scanning Point Clouds. Acta Geodaetica et Cartographica Sinica. 39(5),540-545.

Yang, Y., Zhang, Y., Ma, Y., Yang, J., 2010a. A point cloud filtering method of vehicle-borne lidar based on scanning beam. Journal of Geomatics Science and Technology.27(3),209﹣212.