AN IMPROVED AUTOMATIC POINTWISE SEMANTIC SEGMENTATION OF A 3D URBAN SCENE FROM MOBILE TERRESTRIAL AND AIRBORNE LIDAR POINT CLOUDS: A MACHINE LEARNING APPROACH

Xu-Feng XING [1] *, Mir Abolfazl Mostafavi [1], Geoffrey Edwards [1], Nouri Sabo [2]

[1] Dept. of Geomatics, Laval University, Québec, Canada - xufeng.xing.1@ulaval.ca, (mir-abolfazl.mostafavi, geoffrey.edwards)@scg.ulaval.ca
[2] Canada Centre for Mapping and Earth Observation, Natural Resources Canada, Canada - nouri.sabo@canada.ca

KEY WORDS: Semantic segmentation, 3D urban scene, 3D LiDAR point clouds

ABSTRACT:

Automatic semantic segmentation of point clouds observed in a 3D complex urban scene is a challenging issue. Semantic segmentation of urban scenes based on machine learning algorithm requires appropriate features to distinguish objects from mobile terrestrial and airborne LiDAR point clouds in point level. In this paper, we propose a pointwise semantic segmentation method based on our proposed features derived from Difference of Normal and the features “directional height above” that compare height difference between a given point and neighbors in eight directions in addition to the features based on normal estimation. Random forest classifier is chosen to classify points in mobile terrestrial and airborne LiDAR point clouds. The results obtained from our experiments show that the proposed features are effective for semantic segmentation of mobile terrestrial and airborne LiDAR point clouds, especially for vegetation, building and ground classes in an airborne LiDAR point clouds in urban areas.

1. INTRODUCTION

With the rapid development of LiDAR technologies, airborne and terrestrial LiDAR datasets are widely used as an important source of geospatial information for various applications ranging from 3D mapping to urban planning, land surveying, building reconstruction, 3D city modeling and digital heritage management (Yang, 2013). Generally, LiDAR data processing and modeling steps take tremendous time and operator efforts compared to the data acquisition step (Knaak, 2012). To address this issue, the automation of LiDAR data processing is very important to help to better benefit from the richness of data. Semantic segmentation is one of those important steps in LiDAR data processing that needs to be automated especially for real-time applications.

Semantic segmentation of point clouds directly gives semantic labels to points for better understanding scenes recorded in a point cloud. The efficiency of this process is important in applications such as self-driving cars which navigate themselves by integrating LiDAR scanners to observe the surrounding areas (Fisher, 2013), or on-the-fly decision making for secure navigation and localization. Furthermore, dynamic environment maps and real-time semantic 3D object maps are important prerequisites in motion planning for robots self-navigation as well (Rusu, 2010). In addition, semantic segmentation is required in applications such as cliff recognition to evaluate sea cliff changes (Young, 2010) or detecting transport network obstructions by comparing airborne LiDAR data before and after disasters to shorten the time of reaching disaster sites (Kwan, 2010). In some practical applications, identifying points representing terrain topography from airborne LiDAR point clouds is a fundamental requirement. However, extraction of the topography in urban areas is more complex as tunnels and bridges are not easy to be detected from airborne LiDAR point clouds. Semantic segmentation is promising to classify airborne LiDAR point clouds to deal with varying topography based on appropriate features.

In this paper, we present an improved pointwise semantic segmentation of an urban scene from mobile terrestrial and airborne LiDAR point clouds. Inspired by multi-scale features for classifying points defined in (Hackel, 2016), we propose features derived from Difference of Normal (DoN) for better identifying geometric properties of the surface of different objects. The feature “directional height above” that compares height difference between a given point and its neighbors is defined to improve semantic segmentation of airborne LiDAR point clouds especially for building and ground classes. This allows training models in an urban area with buildings and relatively flat ground. The method is robust enough to segment scenes with changing topography and buildings with different dimensions. Random forest classifier is chosen to classify points based on existed features and new proposed features. The results obtained from several experiments show that the proposed method with newly defined features is effective to improve semantic segmentation of airborne and mobile terrestrial LiDAR point clouds, especially to differentiate ground, buildings, and vegetation from airborne LiDAR point clouds in urban areas.

The remainder of this paper is structured as follows: we present related works in Section 2. Sections 3 presents the proposed method and define the features for semantic segmentation in details. Section 4 presents experiments on mobile and airborne LiDAR point cloud and the analysis of experimental results. Finally, Section 5 concludes this work and presents some perspectives on future work.

2. RELATED WORK

Automatic semantic segmentation for deriving information on individual objects from LiDAR point clouds is a difficult task (Hackel, 2016). Segmentation is the process of partitioning a point cloud into groups with homogeneous properties where all points belonging to a group have the same meaningful label (for

* Corresponding author

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-4-W8-139-2019 | © Authors 2019. CC BY 4.0 License.

example, points belonging to a geometric primitive such as a plane) (Awwad, 2010; Rabbani, 2006). Similarly, semantic segmentation of point clouds gives a semantic label to points representing the same object class (for example a wall or a building). Knowledge-based and machine learning methods are among the approaches that are proposed for the extraction of semantic information from point clouds in urban areas. Knowledge-based methods for extracting semantic information from point clouds have been explored in segmentation, feature extraction, and object recognition from point clouds. (Pu, 2009) extracted semantic features using semantic rules for the reconstruction of building facades from point clouds. (Boochs, 2011), (Hmida, 2012) and (Truong, 2013) used semantic knowledge in all point cloud processing stages for object detection based on three modules including a built knowledge module, an algorithm selection module, and a semantic qualification engine. (Xing, 2018) proposed a knowledge base for feature recognition from point clouds of urban scenes. The prior knowledge about objects is formalized as semantic rules based on ontology in which contains several modules for describing urban scenes in different perspectives. Among them, the spatial relations module allows formalizing possible topological relations among object components extracted from point clouds. The geometric properties and topological relations between object components extracted from point clouds are viewed as facts to infer semantic information of objects, such as recognizing complex geometry, building roof styles and building components.

Machine-learning algorithms are used to extract semantic information from point clouds. For the indoor environment, Rusu (Rusu, 2009b) used Conditional Random Field (CRF) to label small indoor point clouds based on Fast Point Feature Histograms (FPFH) derived from planar segments. Xiong (Xiong, 2013) employed region growing algorithms to detect planar patches from a voxelized point cloud of inner structures of buildings and then used the "Stacking" learning algorithm to classify patches. Then, the patches are annotated with semantic labels of building components. In their work, the features are designed for a group of points in planar segments and the classification is conducted based on the features of planar segments. Armeni (Armeni, 2016) proposed a hierarchical approach for semantic parsing point cloud of an entire building in an indoor space into semantically meaningful spaces at the first level, and spaces parsed into building elements wall, columns in the second level. For identifying building elements, 3D sliding windows are used to slide candidate windows from large-scale point clouds. Then, for each voxel, features including position, size, surface normal, curvature, occupancy, and ratio, were derived from points in the voxel. Structured SVM classifier is chosen to classify candidate windows. This method is effective to segment indoor environment of buildings. Semantic segmentation based on the features of candidate windows can fast segment large indoor scenes. However, the features of candidate windows will not perform well for semantic segmentation of complex urban scenes. For the outdoor environment, pointwise semantic segmentation of point clouds directly gives semantic labels to points, which is a straightforward way to understand scenes. Weinmann (Weinmann, 2013) studied on feature relevance assessment based on geometric 2D and 3D features and analyzed the impact on the semantic interpretation of 3D terrestrial LiDAR point cloud data using four classifiers. The experiments are conducted on a terrestrial LiDAR point cloud representing an urban environment containing smooth ground. For improving the distinctiveness of 2D and 3D geometric features, the optimal size of neighborhoods selection for individual points is explored based on the definition of Shannon entropy (Weinmann, 2015). The multi-scale features extended from (Weinmann, 2013) for dealing with varying point density are used in semantic segmentation of urban area observed by terrestrial LiDAR. This is a supervised pointwise classification of mobile LiDAR point clouds of urban area using a random forest classifier that is simple but powerful, and has good generalization ability (Hackel, 2016). Moreover, Niemeyer (Niemeyer, 2012) used a CRF to classify urban scenes with a flat ground in a point cloud. The fundamental element of machine-learning methods is the definition of features. The features are designed according to nature of segmentation (pointwise or voxel). Meanwhile, the design of features for semantic segmentation of mobile terrestrial LiDAR as well as airborne LiDAR point clouds should consider the variety of objects. For example, height difference in a local area is effective to distinguish ground and buildings, but not effective for tunnels and for varying change of ground. This is because the ground close to the edges of a tunnel will be classified into building class. Therefore, it is necessary to define new features that allow better distinction between tunnels, buildings in the topography itself.

In addition, other solutions based on deep learning for semantic segmentation include Pointnet (Qi, 2017), PointCNN (Li, 2018) and deep learning on multiple 2D image views (or snapshots) of the point cloud (Boulch, 2018). However, these methods need a massive volume of training sets. For practical applications, it is not easy to collect a good training set, especially for semantic segmentation of large-scale mobile terrestrial and airborne LiDAR point clouds.

In summary, knowledge-based methods for extracting semantic information on objects require pre-built rules or knowledge base to infer semantic information combining the information extracted from point clouds. This method is difficult to be applied in point level and if it is used in a large-scale urban scene, the knowledge base containing appropriate rules is essential to recognize different types of objects. A large volume of the pre-labeled training set is needful to deep learning algorithm for semantic segmentation of urban scenes. However, it is possible to obtain good results of semantic segmentation without the need for large volume of training set if the appropriate features are defined for machine learning methods of semantic segmentation of urban scenes.

3. METHOD

Machine learning based semantic segmentation includes three steps: define features for training a classification model, train classification models on a training set based on defined features and evaluate the classifier performance on a testing set. Pointwise semantic segmentation requires to define a feature vector for each point. Then the feature vectors derived from the training set are input into machine learning classifier to train a model. Similarly, feature vectors obtained from the testing set are given to the trained model to classify points for evaluating the classification results.

3.1 Definition of Features for Pointwise Semantic Segmentation

3.1.1 Normal Estimation: Within a point cloud, surface normal estimation at a given point requires the information on its neighbors in a local area (Klasing, 2009). There are several methods for selecting neighbors, including fixed number neighbors selection and fixed radius neighbors selection. Due to the presence of uneven density and occlusion in point clouds, the fixed number neighbors selection allows ensuring the selection of the required points for the estimation of the surface normal. Although, this may introduce some uncertainty in the estimation process. In this paper, the optimal neighbor size is selected using the general definition of the Shannon Entropy (Weinmann, 2014). When the neighbors are chosen, the Principal Component Analysis (PCA) is used to estimate the normal. According to the approach, the local surface covariance matrix C is expressed as:

$$C = \frac{1}{k}\sum_{i=1}^{k}(p_i - \overline{p})\bullet(p_i - \overline{p})^T \; ; C \bullet v_j = \lambda_j \bullet v_j \quad j \in \{0,1,2\} \qquad (1)$$

Where C is a 3*3 symmetric and positive semi-definite matrix.
$\overline{p}$ is the centroid
p_i indicates the neighboring point
λ_j is eigenvalue
v_j is eigenvector

After using Singular Value Decomposition (SVD), its eigenvalues λ_j and the corresponding non-zero eigenvectors v_j are solved. These eigenvectors are orthogonal to each other. In a point cloud with 3D coordinates, if eigenvalues $\lambda_2 > \lambda_1 > \lambda_0 > 0$, the two largest eigenvectors can approximately determine a plane and the eigenvector corresponding to the smallest eigenvalue is its orientation, or normal (Rusu, 2009a). Therefore, the eigenvector corresponding to λ_0 is the approximation of the normal ($+\vec{n}$ or $-\vec{n}$). The known viewpoint and the Riemannian graphs (Hoppe, 1992) can be used to make normal directions uniform. Additionally, the curvature (surface variation) at p_i is defined as:

$$\sigma(p_i) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \qquad (2)$$

3.1.2 Definition of Multi-scale Features for Semantic Segmentation: For a large-scale urban scene, the extraction of multi-scale features from dense point clouds requires huge computation capabilities within a given range of selected neighbors. Decreasing the point density in a large range is necessary to balance computation cost and time. The strategy of downsampling (Brodu, 2012; Hackel, 2016) point clouds makes it possible to select a fixed number of nearest neighbors for different scales (Figure 1). The point cloud is downsampled by generating a pyramid of scales using different voxel size. For a given point in the original point cloud, the fixed number of neighbors is picked up at each scale level. The voxel filter is a widely used method for downsampling 3D point clouds. The bounding volume of the point cloud is divided into small voxels of a given size, and the points in each voxel are replaced by the centroid of the point set. Based on the normal estimation in a point cloud, several features for the characterization of objects (as shown in Table 1) are derived (Gross, 2006; Hackel, 2016; Wang, 2015) at different scales.

We also propose to calculate height above feature based on sliced point cloud in 8 directions (east, south, west, north, southeast, southwest, northeast and northwest) (as shown in Figure 2). For each point P, we slice point cloud in 8 directions on P. The height difference is calculated based on the endpoints of smooth line segments. For example, in Figure 2(B), point P1 is an endpoint of the segment containing current point P because there is a sharp change on point P1 when the line segment grows from P to P1. Similarly, point P2 is an endpoint of another smooth line segment containing the lowest points. The sliced height above feature is computed by the height difference between two neighboring endpoints belonging to different line segments (P1 and P2). The sliced height above features are calculated in eight directions, and they are computed in the level 0 of the downsampled point cloud.

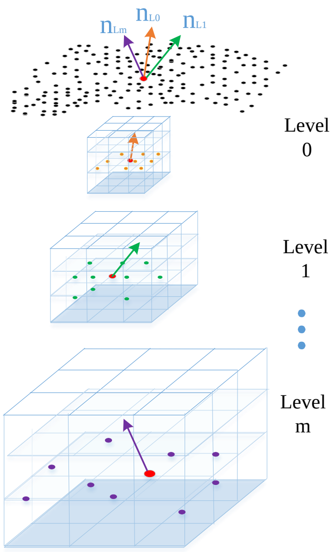


Figure 1. Illustration of downsampling a point cloud at different scales

Features in (Hackel, 2016)	Covariance	Sum	$\lambda_1+\lambda_2+\lambda_3$
		Omnivariance	$\sqrt[3]{\lambda_1 \cdot \lambda_2 \cdot \lambda_3}$
		Eigenentropy	$-\sum_{i=1}^{3}\lambda_i \cdot \ln(\lambda_i)$
		Anisotropy	$(\lambda_1-\lambda_3)/\lambda_1$
		Planarity	$(\lambda_2-\lambda_3)/\lambda_1$
		Linearity	$(\lambda_1-\lambda_2)/\lambda_1$
		Surface Variation	$\lambda_3/(\lambda_1+\lambda_2+\lambda_3)$
		Sphericity	λ_3/λ_1
		Verticality	$1-\lvert\langle n_z, e_3\rangle\rvert$ ($n_z = (0,0,1)$)
	Moment	1st order 1st axis	$\sum_{i \in Neg(P)}\langle p_i - p, e_1\rangle$
		2nd order 1st axis	$\sum_{i \in Neg(P)}\langle p_i - p, e_1\rangle^2$
		1st order 2nd axis	$\sum_{i \in Neg(P)}\langle p_i - p, e_2\rangle$
		2nd order 2nd axis	$\sum_{i \in Neg(P)}\langle p_i - p, e_2\rangle^2$
	Height	Height Range	$Z_{\max} - Z_{\min}$
		Height Below	$Z_p - Z_{\min}$
		Height Above	$Z_{\max} - Z_p$
Our proposed new features	Directional height	Sliced Height Above (8 directions)	$[Z_{d1}, Z_{d2}, Z_{d3}, Z_{d4}, Z_{d5}, Z_{d6}, Z_{d7}, Z_{d8}]$ Where $Z_{di} = Z_{p1} - Z_{p2}$
	DoN	Norm of DoN	$norm(\Delta n_d(p, P_{l\_i}, P_{l\_0}))$

Table 1. Features for semantic segmentation of urban scenes from LiDAR point clouds

Difference of Normal (DoN) is an arithmetic multi-scale operator for evaluating the geometric properties of a point cloud (Ioannou, 2012). Here, the term "scale" indicates the size of the radius used for normal estimation. It is defined as:

$$\Delta n_d(p, r_l, r_s) = \frac{n_l(p, r_l) - n_s(p, r_s)}{2} \qquad (3)$$

Where p is the given point
$n_l(p, r_l)$ is the normal estimated in a large radius r_l
$n_s(p, r_s)$ is the normal estimated in a small radius r_s

Surface normal estimation is dependent on the neighbors located in a sphere defined by radius r. In this paper, r_l is replaced by the downsampled fixed number of nearest neighbors at a larger scale and r_s is replaced by the downsampled fixed number of nearest neighbors at a smaller scale. Thus, the DoN based on the fixed number of nearest neighbors and the scale used for selecting neighbors are defined as follows:

$$\Delta n_d(p, P_l, P_s) = \frac{n_l(p, P_l) - n_s(p, P_s)}{2} \qquad (4)$$

Where P_l represents large-scale fixed number of neighbors
P_s represents small-scale fixed number of neighbors

Hence, the change of scales for picking up nearest neighbors to compute the DoN can reflect the properties of the geometric shape and size of objects. For a perfect plane, the DoN is a zero vector. For a planar segment, the values of DoN for most of points are near the zero vector. The histogram of the norm of DoN reveals the effectiveness of DoN for distinguishing between the objects having planar surfaces (such as buildings and roads) and those without regular geometric shapes (such as trees and bushes). Here the norm of a vector Δn_d is defined as:

$$norm(\Delta n_d) = \sqrt{(\Delta n_d.x)^2 + (\Delta n_d.y)^2 + (\Delta n_d.z)^2} \qquad (5)$$

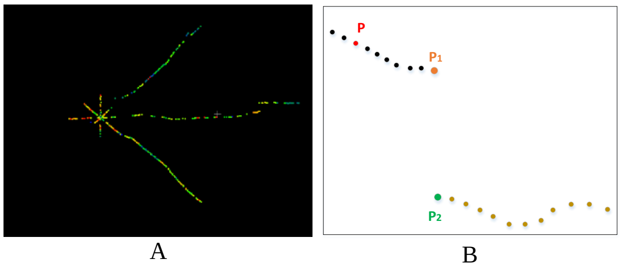


Figure 2. Illustration of “directional height above” feature in 8 directions

In Figure 3, an example is given to show the characteristics of the DoN for different types of objects in three different scales of r_s and r_l. The values of radius r_s and r_l are 0.5 and 1 meter, 0.5 and 1.5 meters, and 1 and 2 meters, respectively. In a point cloud of a building wall, over 98% of points fall in the range of the norm of DoN between 0 and 0.18 while most points have a greater value for a tree. This example reveals that the norm of DoN calculated between normals estimated using downsampled neighbors at different scale is effective for semantic segmentation of point clouds.

3.2 Random Forest for Pointwise Semantic Segmentation

Random forest (Breiman, 2001) is composed of a collection of decision trees constructed using random features sampled independently. Each tree is trained on the training set based on bootstraps that creates a random resampling on training set itself, and random features are selected to create trees (Svetnik, 2003). The prediction is decided by aggregating of the predictions of decision trees. Each node in a binary decision tree represents a feature selected for splitting samples into two classes. Gini impurity measures how well a potential split is in this node (Menze, 2009). The formula of Gini impurity is:

$$Gini(m) = 1 - \sum_{i=1}^{C} (p_i)^C \qquad (6)$$

Where $p_i = n_k/n$ is the fraction of n_k samples from C classes out of the total of n samples at node m.

The multi-scale features defined in the previous section are organized as a features vector which combines features obtained in different scales. The feature vector is produced on each point. The feature vector represents the learnable variables of objects. Then the feature vectors are input into machine learning classifier for learning the parameters for classification from the training set. Similarly, the feature vectors calculated from the testing set are used in the semantic segmentation on the testing set for the evaluation of the performance of the learned classifier. Generally, the precision, recall, and F1-Score are calculated to compare the performance of the classifier.

We choose a Random Forest classifier for pointwise semantic segmentation because it is straightforward to deal with multi-class problem and it is easy to parallelize its implementation. It demonstrated good results on large-scale point clouds in a reasonable time (Hackel, 2016; Weinmann, 2015). We use the random forest algorithm in Scikit-learn library with Gini-impurity (Menze, 2009) as the splitting criterion. In our application, the density of point clouds is uneven, and it has high density on the ground. Due to occlusion, the scanning angle and the viewpoint of the scanner, some parts of objects are missing or have low density. Thus, the uneven density of points leads to a distribution of class labels that do not conform to reality, which affects the training of the classifier parameters. For decreasing this negative effect, the dataset is downsampled with an appropriate resolution. After downsampling, the dataset better represents the true distribution of classes in point clouds. In fact, the high density of points in the local area is not better than the even point density because it is difficult to reflect the geometric properties in a small range with dense points. Considering the computation efficiency, the downsampled training set is economic to fit the capabilities of memory and to get the classification done in a reasonable time. In addition, the classifier trained on the point cloud with even density and reasonable resolution has good generalization capabilities.

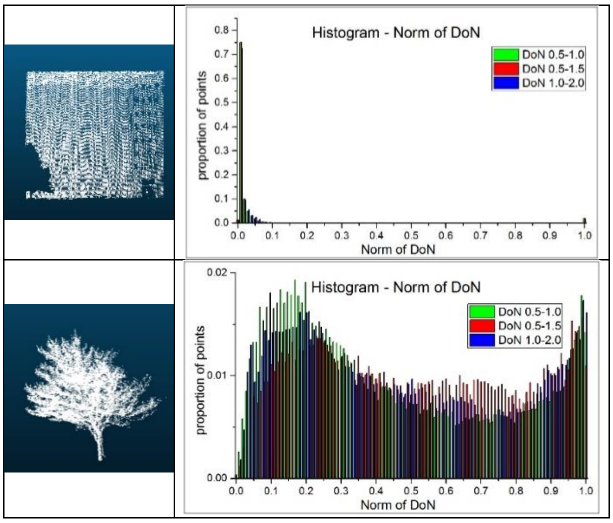


Figure 3. Histogram of the norm of DoN for distinguishing different objects.

4. EXPERIMENTS AND RESULTS

For the first experiment, a mobile LiDAR dataset is scanned at Laval University campus by a Terrapoint Titan mobile LiDAR system. The average point spacing in the point cloud is 0.089m. The point density is approximately 130 points/m2. A higher density of points is observed on the roads. Due to occlusions, scan angle, and objects properties, the point density is non-uniform and objects are incomplete in some parts of the point cloud. In the LAS files, the outlier points are cleaned, and moving objects and noisy points are removed from the raw dataset.

In addition, we have carried out second experiment with airborne LiDAR dataset from Montreal area. The point cloud contains flat urban terrain and changing topography following the both sides of railway. In the point cloud, there are buildings, vegetation, bridges, ground, and other classes (power lines, cars, poles, etc.)

All algorithms are implemented in C++ in QT with Point Cloud Library for computing features and scikit-learn for classification using random forest classifier. All experiments are run on a laptop with Intel Xeon E3-1505 v5 CPU (quad-core, 2.8GHz) and 48 GB of RAM. The process of computing features is parallelized across the available CPU cores. The training of model and the classification step are set to parallel as well.

4.1 Experiments on Mobile Terrestrial LiDAR Point Cloud

Based on the proposed pointwise segmentation method, the features of each point are composed of features derived from multi-scale neighbors selection. We chose 8 scales (0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4 and 12.8 (meters)) for downsampling point clouds to calculate the features at each scale. The first scale is computed based on the average point density of point clouds. Here we consider the first scale to be slightly greater than the value of average point density. Then the DoN is computed as the difference between the normal estimated at the smallest scale and the normals at other larger scales. All the features are combined as a feature vector for each point. When the feature vectors of points are extracted, they are input into a random forest classifier to train the classification model. To do so, we first need to have a training set that contains the defined classes of objects for the semantic segmentation. Based on this training set, other raw point clouds are classified using the trained model. The dataset in Figure 4(B) is chosen as the training set to train the model and the dataset shown in Figure 4(A) for testing. In the training set, there are classes of objects such as ground, bushes, trees, buildings, cars, curbs, light poles, sign poles, traffic indicators, benches, people, bus stations, etc. However, the points representing traffic indicators, benches, people and bus stations are low in proportion compared to the other object classes. In the testing dataset, we have similar distribution of the objects.

We have defined 50 trees and Gini index as the splitting criterion to train the random forest classifier. Then the trained model is used on the testing dataset presented in Figure 4(A). The results of the test are presented in Table 2. The classification results of the testing dataset are presented in Figure 5. The analysis of the results reveals that in the case where objects are too close to each other and have similar geometric properties (such as, tree and bush), or are absent in training set, the classification is not very efficient. For instance, in the training set, bushes and trees are located close to each other. In the testing set, some parts of bushes are misclassified into the tree class. Also, when the point cloud is downsampled for calculating the feature vectors, the geometric properties of curves are not clear extracted. As the curved walls do not occur in the training set, the curved parts of the building in the given data set are misclassified into tree class. Additionally, the density of points for traffic indicators, benches, and people is not high enough and the numbers of instances of these classes are all less than 3, which is not enough to make the classifier learn.

Classes	Our results (%)			Results based on features in (Hackel, 2016) (%)		
	P	R	F1	P	R	F1
Ground	93.92	99.88	96.81	93.85	99.90	96.78
Bush	45.31	50.57	47.80	39.27	49.84	43.92
Tree	78.81	98.59	87.60	80.51	96.61	87.83
Building	88.97	71.38	79.21	84.90	70.49	77.03
Car	93.46	45.89	61.56	95.07	45.57	61.61
Curb	72.89	3.34	6.39	77.42	2.81	5.42
Light pole	97.94	36.52	53.20	97.40	38.00	54.67
Sign pole	49.37	15.29	23.35	38.69	18.68	25.19
Traffic indicator	0	0	0	0	0	0
Bench	0	0	0	0	0	0
People	0	0	0	0	0	0
Bus station	0	0	0	0	0	0
Overall	90.29	90.43	88.92	90.24	90.16	88.74

Table 2. Quantitative results of testing on mobile LiDAR point cloud (Precision (P), Recall (R), F1-Score (F1))

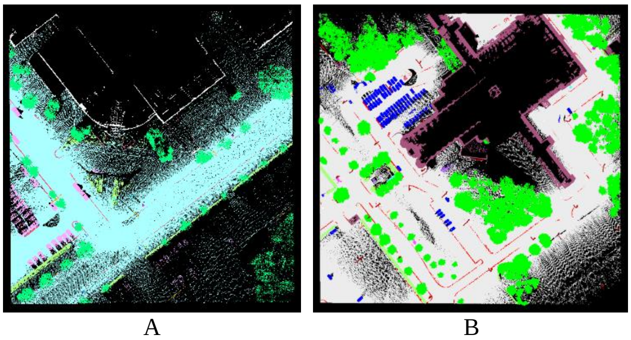


Figure 4. Dataset for testing (A) and for training the classification model (B)

4.2 Experiments on Airborne LiDAR Point Cloud

The airborne LiDAR point cloud of urban areas for the experiment contains ground, vegetation, building, bridge, power line, tower, fence, car, and pole. In this experiment, we classify point cloud into six classes: ground, low vegetation, middle vegetation, high vegetation, building, and others. The bridges are classified into building class. The rest of the objects are given as other classes, including power line, cars, fences, towers, and poles near to railway. In Figure 7, the left part is chosen as the training set and the right part for testing. In the training area, there are bridges and tunnels in the ground class. The testing area contains the railway environment, and there is a changing topography on both sides of the railway (Figure 6). In contrast, in the training area, the topography is relatively smooth and flat.

We chose seven scales (0.2, 0.4, 0.8, 1.6, 3.2, 6.4 and 12.8 (meters)) for downsampling point clouds to calculate the features at each scale. The first scale is decided as 0.2 due to the average point density of airborne LiDAR point clouds is near to 0.2 meters. After training the classification model from the training area of the point cloud, we compare our results and the result based on the features in (Hackel, 2016) (Table 3). Our proposed features for semantic segmentation of airborne LiDAR point clouds have good performance in building class and high vegetation class. As shown in Figure 8, our results are better than the results based on features in (Hackel, 2016) in building classes. However, we can still see some misclassifications in the results. As shown the results in Figure 8(A), some points in the central part of the building roof is misclassified as ground class.

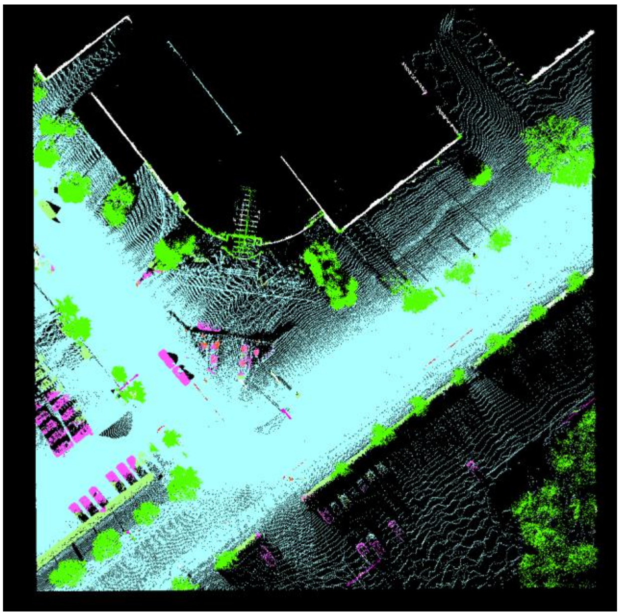
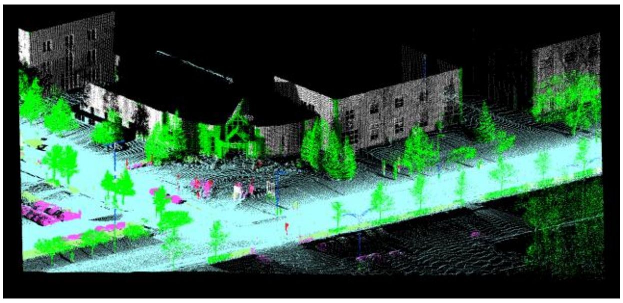

A

B

Figure 5. Classification result on testing point cloud (A) and its 3D view (B)

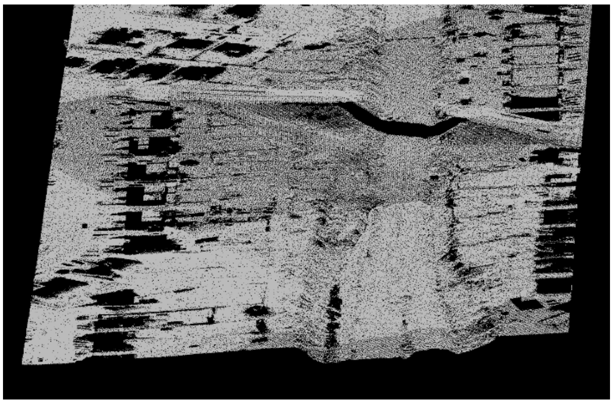


Figure 6. 3D view of points for creating topography in the testing area

Classes	Our results(%)			Results based on the features in (Hackel, 2016) (%)		
	P	R	F1	P	R	F1
Ground	96.74	98.99	97.85	94.26	99.14	96.64
Low vegetation	83.29	1.13	2.23	71.67	1.22	2.39
Middle vegetation	75.84	10.09	17.80	75.04	11.88	20.52
High vegetation	96.96	89.59	93.13	97.08	84.4	90.26
Building	94.70	90.28	92.44	91.44	75.52	82.72
Others	43.26	70.99	53.67	42.54	71.12	53.24
Overall	93.27	92.27	91.47	90.95	90.01	89.14

Table 3. Quantitative results of testing on airborne LiDAR point cloud (Precision (P), Recall (R), F1-Score (F1))

4.3 Discussion

In this experiment, the proposed features help to improve the overall semantic segmentation of an urban scene from mobile terrestrial LiDAR point clouds. More specifically, there is an obvious improvement of precision on the building class. For semantic segmentation of airborne LiDAR point cloud, our proposed new features are effective to ground, building and low vegetation classes. In the overall semantic segmentation, our results have 2.26% improvement from the comparison of recall between ours and Hackel's results. However, due to the unbalance of object classes and the limited number of examples, the bridge class is not learned from training sets. In this case, a greater data set can help to produce more examples in the training step. In addition, the computation time for semantic segmentation of airborne LiDAR point clouds using our method is about 10 minutes/million points. In general, more features require more computation time. In our work, the computation of directional height difference is easy to be parallelized based on the downsampling of point clouds. But the steps of downsampling and normal estimation are not parallelized. There is still a space to improve the computation time if the downsample and normal estimation are done using parallel computation on CPU or GPU.

5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an improvement to previously proposed methods for semantic segmentation by adding features derived from Difference of Normal (DoN) and "directional height above" neighbors for semantic segmentation of mobile and airborne LiDAR point clouds. The proposed features allow to improve semantic segmentation of mobile and airborne point clouds in urban scenes. We use a random forest classifier for pointwise segmentation of point clouds. After comparing our results and the results based on features in (Hackel, 2016), the newly proposed features produce slightly improved semantic segmentation results of vegetation and building classes in mobile LiDAR point clouds. However, there are significant improvements on the vegetation and building classes in the semantic segmentation of airborne LiDAR point clouds. As future work, we plan to integrate other features that can be extracted from supplementary data sources into the proposed approach that will allow to further improve semantic segmentation of a LiDAR point cloud.

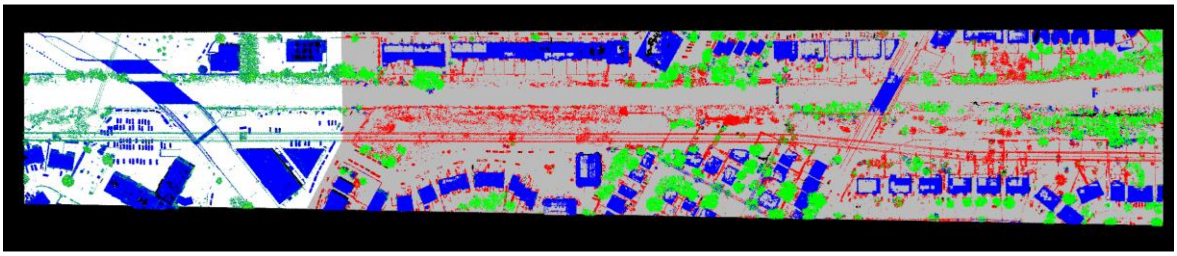
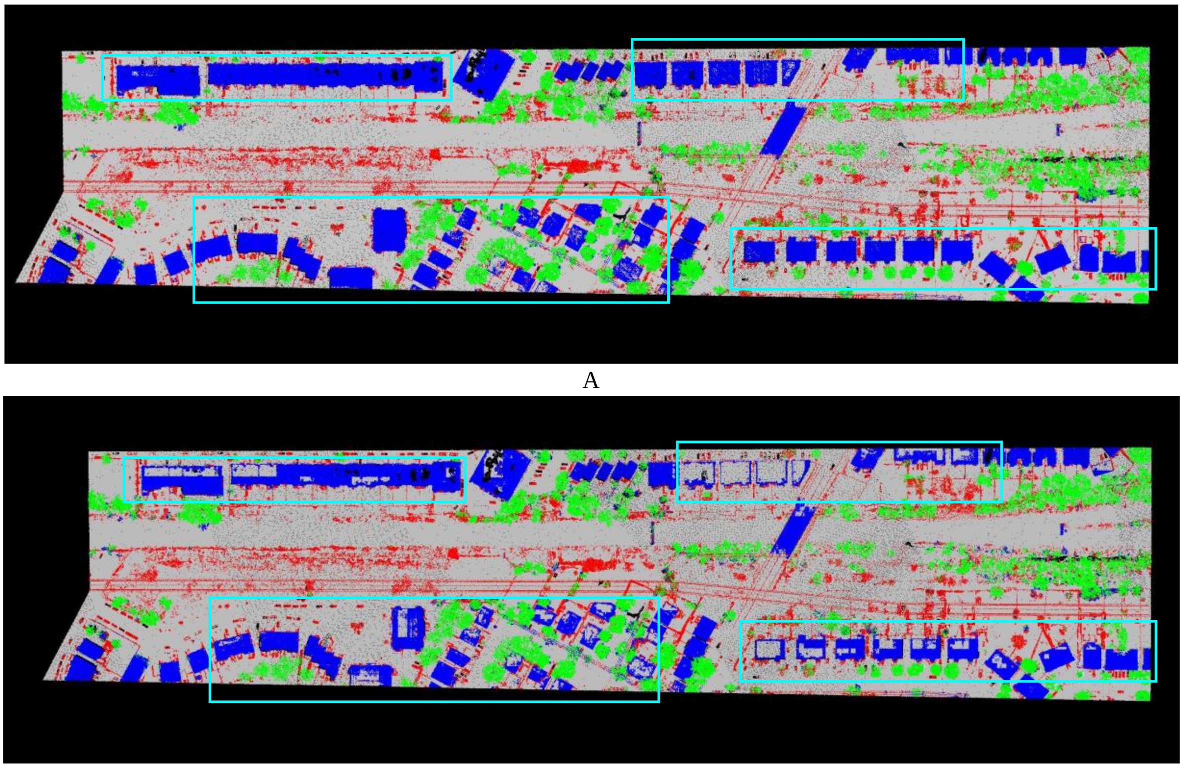


Figure 7. Datasets for training (left) and testing (right)



A

B

Figure 8. Comparison of semantic segmentation results of airborne LiDAR point cloud. (A) our result and (B) result based on features in (Hackel, 2016)

ACKNOWLEDGMENTS

This research is supported jointly by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the China Scholarship Council. The authors would like to gratefully acknowledge the dataset provider for the experimentations presented in this paper. We are also grateful to Sonia Rivest for her comments on the manuscript.

REFERENCES

Armeni, I., O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer and S. Savarese, 2016. 3d semantic parsing of large-scale indoor spaces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Awwad, T. M., Q. Zhu, Z. Du and Y. Zhang, 2010. An improved segmentation approach for planar surfaces from unstructured 3D point clouds. *The Photogrammetric Record* 25(129): 5-23.

Boochs, F., A. Marbs, H. B. Hmida, T. Hung, A. Karmachaiya, C. Cruz, A. Habed, C. Nicolle and Y. Voisin, 2011. Integration of knowledge to support automatic object reconstruction from images and 3D data. *2011 8th International Multi-Conference on Systems, Signals and Devices (SSD)*.

Boulch, A., J. Guerry, B. Le Saux and N. Audebert, 2018. SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics* 71: 189-198.

Breiman, L., 2001. Random Forests. *Machine Learning* 45(1): 5-32.

Brodu, N. and D. Lague, 2012. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. *ISPRS Journal of Photogrammetry and Remote Sensing* 68(0): 121-134.

Fisher, A, 2013. Google's Self-Driving Cars: A Quest for Acceptance. http://www.popsci.com/cars/article/2013-09/google-self-driving-car (18 July 2019).

Gross, H. and U. Thoennessen, 2006. Extraction of lines from laser point clouds. Symposium of ISPRS Commission III: Photogr ammetric Computer Vision PCV06. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences.

Hackel, T., J. D. Wegner and K. Schindler, 2016. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Prague, Czech Republic* 3: 177-184.

Hmida, H. B., C. Cruz, F. Boochs and C. Nicolle, 2012. Knowledge Base Approach for 3D Objects Detection in Point Clouds Using 3D Processing and Specialists Knowledge. *International Journal on Advances in Intelligent Systems* 5(1&2): 1-14.

Hoppe, H., T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, 1992. Surface reconstruction from unorganized points. *SIGGRAPH Comput. Graph.* 26(2): 71-78.

Ioannou, Y., B. Taati, R. Harrap and M. Greenspan, 2012. Difference of Normals as a Multi-scale Operator in Unorganized Point Clouds. *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*.

Klasing, K., D. Althoff, D. Wollherr and M. Buss, 2009. Comparison of surface normal estimation methods for range sensing applications. *2009 IEEE International Conference on Robotics and Automation*.

Knaak, T, 2012. Two Perspectives on LiDAR Technology Market Adoption. http://www.lidarmag.com/content/view/8864/199/ (12 September 2017).

Kwan, M.-P. and D. M. Ransberger, 2010. LiDAR assisted emergency response: Detection of transport network obstructions caused by major disasters. *Computers, Environment and Urban Systems* 34(3): 179-188.

Li, Y., R. Bu, M. Sun and B. Chen, 2018. PointCNN. *arXiv preprint arXiv:1801.07791*.

Menze, B. H., B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich and F. A. Hamprecht, 2009. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics* 10: 213-213.

Niemeyer, J., F. Rottensteiner and U. Soergel, 2012. Conditional random fields for lidar point cloud classification in complex urban areas. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences* 3: 263-268.

Pu, S. and G. Vosselman, 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing* 64(6): 575-584.

Qi, C. R., H. Su, K. Mo and L. J. Guibas, 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* 1(2): 4.

Rabbani, T., F. A. van den Heuvel and G. Vosselman, 2006. Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences*.

Rusu, R., 2010. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. *KI - Künstliche Intelligenz* 24(4): 345-348.

Rusu, R. B., 2009a. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. PhD, Tecnische Universitatet Muenchen.

Rusu, R. B., Z. C. Marton, N. Blodow, A. Holzbach and M. Beetz, 2009b. Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments. *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*.

Svetnik, V., A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan and B. P. Feuston, 2003. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *Journal of Chemical Information and Computer Sciences* 43(6): 1947-1958.

Truong, H. Q., H. B. Hmida, F. Boochs, A. Habed, C. Cruz, Y. Voisin and C. Nicolle, 2013. Automatic detection and qualification of objects in point clouds using multi-layered semantics. *Photogrammetrie - Fernerkundung - Geoinformation* 2013(3): 221-237.

Wang, Z., L. Zhang, T. Fang, P. T. Mathiopoulos, X. Tong, H. Qu, Z. Xiao, F. Li and D. Chen, 2015. A Multiscale and Hierarchical Feature Extraction Method for Terrestrial Laser Scanning Point Cloud Classification. *IEEE Transactions on Geoscience and Remote Sensing* 53(5): 2409-2425.

Weinmann, M., B. Jutzi and C. Mallet, 2013. Feature relevance assessment for the semantic interpretation of 3D point cloud data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 5(W2): 1.

Weinmann, M., B. Jutzi and C. Mallet, 2014. Semantic 3D scene interpretation: a framework combining optimal neighborhood size selection with relevant features. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2(3): 181.

Weinmann, M., S. Urban, S. Hinz, B. Jutzi and C. Mallet, 2015. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. *Computers & Graphics* 49: 47-57.

Xing, X.-F., M.-A. Mostafavi and S. Chavoshi, 2018. A Knowledge Base for Automatic Feature Recognition from Point Clouds in an Urban Scene. *ISPRS International Journal of Geo-Information* 7(1): 28.

Xiong, X., A. Adan, B. Akinci and D. Huber, 2013. Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction* 31(0): 325-337.

Yang, B. and Z. Dong, 2013. A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 81(0): 19-30.

Young, A. P., M. Olsen, N. Driscoll, R. Flick, R. Gutierrez, R. Guza, E. Johnstone and F. Kuester, 2010. Comparison of airborne and terrestrial LiDAR estimates of seacliff erosion in southern California. *Photogrammetric Engineering & Remote Sensing* 76(4): 421-427.