AN OBJECT-BASED CLASSIFICATION FRAMEWORK FOR ALS POINT CLOUD IN URBAN AREAS

E. Hasanpour Zaryabi [1]*, M. Saadatseresht [1], E. Ghanbari Parmehr [2]

[1] School of Surveying and Geospatial Engineering, University of Tehran, Tehran, Iran - (erfan.hasanpoor, msaadat)@ut.ac.ir
[2] Babol Noshirvani University of Technology, Dept. of Geomatics, Faculty of Civil Engineering, Babol, Iran - parmehr@nit.ac.ir

Commission IV, WG IV/3

KEY WORDS: Point Cloud, Segmentation, Classification, Supervoxel, Voxel, Local Graph

ABSTRACT:

This article presents an automated and effective framework for segmentation and classification of airborne laser scanning (ALS) point clouds obtained from LiDAR-UAV sensors in urban areas. Segmentation and classification are among the main processes of the point cloud. They are used to transform 3D point coordinates into a semantic representation. The proposed framework has three main parts, including the development of a supervoxel data structure, point cloud segmentation based on local graphs, and using three methods for object-based classification. The results of the point cloud segmentation with an average segmentation error of 0.15 show that the supervoxel structure with an optimal parameter for the number of neighbors can reduce the computational cost and the segmentation error. Moreover, weighted local graphs that connect neighboring supervoxels and examine their similarities play a significant role in improving and optimizing the segmentation process. Finally, three classification methods including Random Forest, Gradient Boosted Trees, and Bagging Decision Trees were evaluated. As a result, the extracted segments were classified with an average precision of higher than 83%.

1. INTRODUCTION

In the mid-1990s, ALS (airborne laser scanning) systems were developed for bathymetric and topographical applications. As a result of the laser scanning sensor installed on the airplane, 3D coordinates (XYZ) are calculated from a cloud of laser range measurements (Yan et al., 2015). During the development of UAVs and the design of lightweight laser scanner sensors installed on UAVs, LiDAR-UAV technology was developed. LiDAR-UAV is a type of airborne laser scanner (ALS) that can fly at lower altitudes and measure 3D points on an object's surface in more detail. The output of a LiDAR sensor is 3D coordinates, and these 3D coordinates represent the geometry of the surface from which the laser beams are reflected. A Point cloud as raw data does not have classified geometric, topological and descriptive information and it is not easy to distinguish different objects from each other. Therefore, it needs processes to transfer these 3D information to a meaningfully higher-level space (Xu et al., 2017). In order to extract semantic information from a point cloud, segmentation and classification are widely used. Both segmentation and classification processes can be performed in three levels, including point-based, segment-based (over segmentation), and object-based. Due to the high computational cost of point-based methods and their sensitivity to noise, a technique called supervoxel has attracted many attentions in recent years. The primary purpose of this research is to present a framework for the segmentation and classification of LiDAR-UAV point clouds in urban areas by combining spectral and geometric information and utilizing supervoxel structure.

2. RELATED WORK

Point cloud segmentation is a 3D partitioning of points into areas where the points share one or more similar characteristics (geometric, spectral, etc.). (Sithole & Vosselman, 2004). On the other hand, In classification, points are classified with different labels based on defined criteria that specify the object type (Grilli et al., 2017). Challenges such as irregular sampling, differences in density of points, objects with complex structure, etc. made point cloud segmentation and classification as significant topics. Segmentation methods are generally classified into three groups including model-based, region growing, and segment-based segmentation. Moreover, Classification methods based on segmentation level generally fall into three groups including point-based, segment-based, and object-based classification. Model-based segmentation determines the connectivity between points by fitting specific mathematical models based on surface geometric parameters (normal vectors, curvature, etc.). RANSAC and its extended versions are among the popular methods in this group, used to fit various shape models and extract different objects from point cloud in the presence of noise and errors. (Schnabel et al., 2007). 3D Hough Transform is another widely used method that successfully extracts lines and planes from the point cloud (Tarsha-Kurdi et al., 2007). The main disadvantages of these methods are the high computational cost of modeling, the high memory requirements, and the inability to segment objects that do not follow a specific mathematical shape. Region growing methods are iterative processes that evaluate neighboring points around a seed point to determine whether those points belong to that seed point. Studies (Belton & Lichti, 2006) and (Klasing et al., 2009) used two features of normal vector and curvature to find the connection of smooth regions. In 2015, a region growing method was introduced with an octree structure that uses only the geometric features of the point cloud

* Corresponding author

https://doi.org/10.5194/isprs-annals-X-4-W1-2022-279-2023 | © Author(s) 2023. CC BY 4.0 License.

to define the growth criteria. The main problem for this group is finding the seed points to start the process. Segment-based methods determine connectivity between adjacent points by examining the spectral and geometric similarity. Euclidean distance (Aldoma et al., 2012), and normal vector (Vo et al., 2015) are examples of similarity criteria used to form the initial segments. The best-known methods of this group are Min Cut (Golovinskiy & Funkhouser, 2009), Normalize Cut (Shi & Malik, 2000), Graph-based segmentation (Felzenszwalb & Huttenlocher, 2004), Markov Random Field (Hackel et al., 2016), and Conditional Random Field (Rusu et al., 2009). The computational cost of these methods depends on the complexity of the similarity criteria.

As explained for all above methods, the complexity of objects in a point cloud and the computational cost are challenging for segmentation methods. Therefore, using 3D segments instead of points as main units for segmentation process attracted many attentions. Voxel (Wang & Tseng, 2011), flat planes (Vosselman et al., 2017), patch (Iman Zolanvari & Laefer, 2016), and supervoxel (Papon et al., 2013) are examples that have been considered in various studies as the primary units for segmentation.

3. PROPSED METHOD

In this study, a framework for segmentation and classification of point clouds is presented. First, an initial segmentation is performed on the point cloud using the supervoxel structure. A local graph-based segmentation (Xu et al., 2017) are then utilized to combine supervoxels. After segmenting the supervoxels, the final segments are classified using three classification methods including Random Forest, Gradient Boosted Trees, and Bagging Decision Trees. The general framework of this study is shown in Figure 1.

3.1 Supervoxel Generation

The main idea of supervoxel structure derives from how superpixels are formed in images (Achanta et al., 2012). The method presented in this section is taken from the two processes of forming superpixels (Achanta et al., 2012) and octree-structured supervoxels (Papon et al., 2013). The first step in forming supervoxels is to create a regular structure and select initial points as seed points. The voxel structure is used to create this regular structure. After voxelizing the point cloud, the voxels within which there are fewer points than a certain threshold are discarded. For the remaining voxels, the centroid of the points within each voxel is calculated. These centroids are considered as starting points for voxel growth. Assuming that the size of each voxel is S, only points located in a cube of dimensions 2*S around a seed point are considered as candidate points for growing that seed point.

After selecting the initial seed points, three types of features are selected to create a general criterion for the growing of seed points. The first feature is the spatial distance between seed points and their neighboring points. This feature limits the distance between seed points and their neighbors. Therefore, far-distance points cannot be selected for connection. The second feature is the colour distance between points. This feature measures how similar seed points and their neighbors are in RGB space. The third feature is the angular difference between the normal vectors. After calculating these three features, they should be normalized using their maximum values. In addition, for each feature, there is an additional coefficient ($W_{spatial}, W_{spectral}, W_{normal}$) that indicates the importance of each feature. Therefore, the following formula is the final criterion for growing supervoxels, which is based on the research (Papon et al., 2013).

$$D_{\boldsymbol{final}} = \sqrt{\left(W_{spatial} * \frac{D_{spatial}}{\sqrt{3}S}\right)^2 + \left(W_{spectral} * D_{spectral}\right)^2 + (W_{normal} * D_{normal})^2} \quad (1)$$

3.2 Feature Extraction

The characteristics of a supervoxel are defined using the points within it. For each supervoxel, three types of features are used including spatial distance, geometric similarity, and surface connectivity. The spatial distance is the distance between the centroids of two neighboring supervoxels. Geometric similarity is defined using eigenvalue-based features that express the 3D distribution of points within a supervoxel. Four geometric features were calculated, including degree of linearity, flatness, point, and degree of curvature change (C). (Weinmann et al., 2015). The third feature for segmentation is surface connectivity (Stein et al., 2014). This criterion examines the smoothness and convexity of surfaces for points within adjacent supervoxels. According to research (Xu et al., 2017). The connection between surfaces is assumed only four types of smooth, stair-like, convex and concave. The formula for calculating this criterion is given in Equation (2). The first term of this criterion, examines the smoothness of two surfaces. The second term is the convexity criterion, which determines the type of connection between two surfaces. The connection between two supervoxels is likely to be convex when the value of this term is high. The type of connection between two supervoxels V_i and V_j is determined based on the relationship between their normal vectors N_i, N_j and the vector connecting the two centres of the supervoxels X_i and $X_j(d_{ij}$ vector). As illustrated in Fig1, α is the angle between the normal vector N and the vector d_{ij}. According to Equation (2), The high probability of continuity (smaller value D_{ij}^c**)** belongs to the surfaces that the connection types between them are convex, smooth or stair-like**,** which occurs when $\alpha_i - \alpha_j > \Theta$ (Xu et al., 2017)**.** According to the research (Stein et al., 2014), ϴ is calculated using a sigmoid function shown in Equation (3).

$$\begin{cases} (\alpha_i - \alpha_j)^2 + (\pi - \alpha_i - \alpha_j)^2 & if\ \alpha_i - \alpha_j > \Theta \\ (\alpha_i - \alpha_j)^2 + \pi^2 & else \end{cases} \quad (2)$$

In the equation, the threshold ϴ is calculated using the research (Stein et al., 2014):

$$\Theta = -\Theta_{max} \cdot \left(1 + \exp\left(-\alpha \cdot \left(\beta - \beta_{off}\right)\right)\right)^{-1} \quad (3)$$

3.3 Supervoxel-Based Local Graphs

At this stage, for each supervoxel V_i, all its N neighbors are selected as candidates to construct the local graph G = (V, E), where V and E are nodes and edges of the graph G, respectively. Among these N supervoxels, a sphere of radius R is considered as a search area for each supervoxel and distances between all N supervoxels are calculated from each other. Two supervoxels V_i and V_j are selected as adjacent supervoxels if the distance between them is less than √3S (S, the initial size of voxels). After selecting the neighboring supervoxels, three features including spatial distance, geometric similarity, and surface continuity are calculated, and the following formula is used to calculate the weight of edges for constructed graph (Xu et al., 2017).

$$W(i \text{،} j) = \prod_{k \in [p \text{،} s \text{،} c]} e^{\left(-\frac{(D_{ij}^k)^2}{2\lambda_k^2}\right)} \quad (4)$$

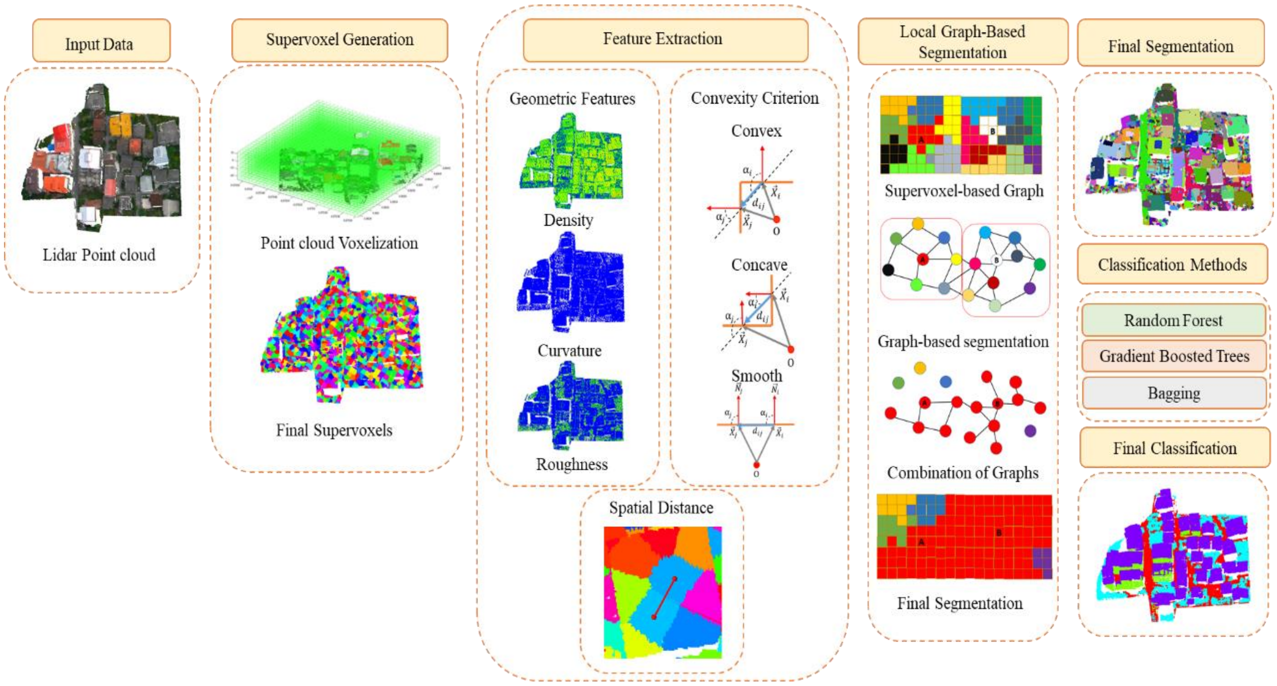

Figure 1. The general framework of the proposed method.

The three parameters λ_p, λ_s, and λ_c control the importance of spatial distance, geometric similarity, and surface continuity, respectively.

3.4 Segmentation of Local Graphs

The segmentation method used for graph segmentation is based on two studies (Felzenszwalb & Huttenlocher, 2004) and (Xu et al., 2017). Once local graphs are formed for all supervoxels and their neighbors, it is possible to determine the final connection of each supervoxel to other supervoxels by individually segmenting each local graph and forming the final segmentation. First, each supervoxel V is an initial segment S and the number of segments and supervoxels are the same and $S \in C$ (C indicates the final segmentation). Then all edges are arranged in descending order of their weights and an initial threshold is defined to merge supervoxels whose weights are greater than the threshold. This stage is an iterative process and each time a new segment is merged with another segment if one or both segments have been merged at least once. The two segments are then merged if the minimum weight between them is greater than the minimum internal difference. In the following, we explain the mathematical terms. First, the internal difference $Int(C)$ for each segment S is calculated. This term is defined as the maximum weight among the supervoxels placed in a segment (Xu et al., 2017).

$$Int(C) = \max_{e \in E(C)} W(e) \quad (5)$$

The external difference between the two segments C_1 and $C_2$is defined as the minimum weight connecting a supervoxel from the first segment to a supervoxel from another one (Xu et al., 2017).

$$Dif(C_1 ‹ C_2) = \min_{V_i \in C_1 ‹ V_j \in C_2 ‹ (V_i ‹ V_j) \in E} W(V_i ‹ V_j) \quad (6)$$

If there is no edge between C_1 and C_2, $(C_1, C_2) = \infty$, and the value of $Dif(C_1, C_2)$ should be larger than one of $Int(C_1)$ or $Int(C_2)$. A threshold τ is also used to control the value of Int (Xu et al., 2017).

$$D(C_1, C_2) = \begin{cases} true & if\ Dif(C_1, C_2) > MInt(C_1, C_2) \\ false & otherwise \end{cases} \quad (7)$$

In the above equation, the minimum internal difference $MInt(C_1 ‹ C_2)$ is defined as follows:

$$MInt(C_1 ‹ C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)) \ (8)$$

The threshold τ is initially equal to a constant value k. Each time a new segment S_i is connected to C_i, the threshold is updated as follows:

$$\tau(C_j) = k / |C_j| \ (9)$$

In the above formula $|C_j|$ is equal to the number of supervoxels S that form C_j.

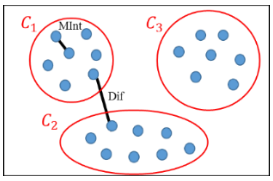

Figure 2. Visual representation of MInt and Dif.

3.5 Combination of local graphs

After segmenting the local graphs individually, according to research (Xu et al., 2017) , the local graphs are connected to form final graphs. As shown in Figure 3, two segmented local graphs with centres V_i and V_j are common in two supervoxels V_K and V_e. these two supervoxels connect two local graphs. Consequently, each final graph is known as individual segment. Figure 3 shows the general steps of the segmentation method.

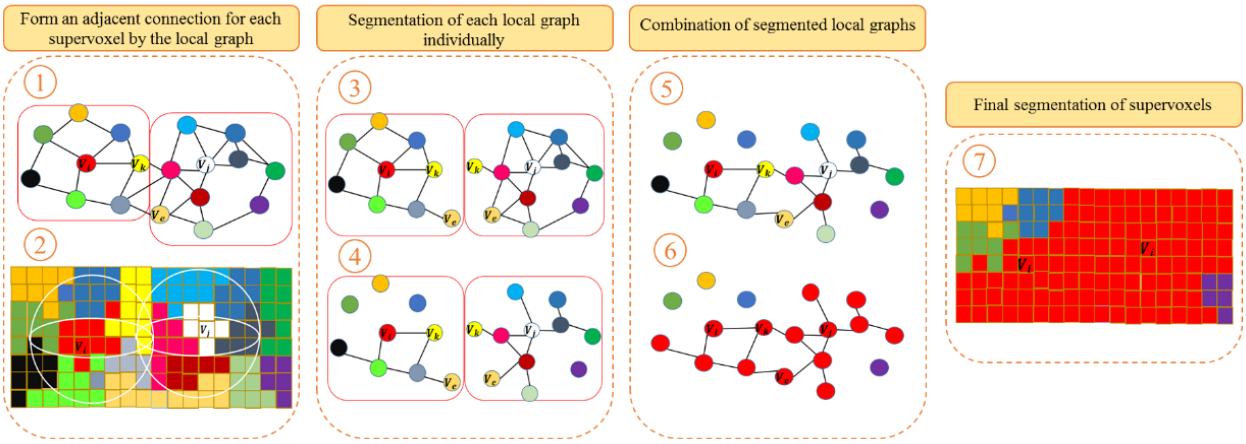
Form an adjacent connection for each supervoxel by the local graph
Segmentation of each local graph individually
Combination of segmented local graphs
Final segmentation of supervoxels
1
2
3
4
5
6
7
V_i
V_k
V_l
V_e

Figure 3. The general steps of the segmentation method based on local graphs (Xu et al., 2017).

3.6 Classification of Segments

In this stage, three classification methods including Random Forest, Gradient Boost Trees, and Bagging Decision Trees are used to classify the segments based on 18 geometric and spectral features.. The Random Forest algorithm was proposed by Breiman in 2001 (Breiman, 2001). This algorithm is one of the Ensemble Learning algorithms that combine outputs from multiple classifiers to achieve more powerful classification (Zhou, 2012). This method is performed in two stages, the first stage is the formation of learners, and the second stage is the combination of learners. The Gradient Boosted Trees algorithm is another classification method used in this research. This algorithm is a reinforcement learning framework using tree-based learning algorithms proposed by Friedman in 2006 (Hastie et al., 2009). This algorithm has a faster training speed, higher efficiency, and less memory usage than other tree-based learning methods. Since Gradient Boosted Trees is based on decision tree algorithms, the growth of this algorithm is vertical while the growth of other algorithms is horizontal. The last classification method is The Bagging Decision Trees, introduced by Breiman in 1996 to improve classification accuracy. Like the Random Forest algorithm, this algorithm is part of the tree-based ensemble learning algorithms. This method assigns each point a label based on the maximum number of votes received from a group of classifiers.

4. EXPERIMENTAL RESULTS

4.1 Dataset

In this study, Point cloud data collected by a LiDAR-UAV sensor with an average density of 13 $^{p}/_{m^3}$ was used. The data were collected in the city of Tonekabon in Iran. Three urban parts of the data were chosen for the experiment. The data has four classes including building roof, tree, wall and ground.

	Part 1	Part 2	Part 3
Number of Points	327592	516061	240870
Area	3886750 km^2	6722508km^2	342805$\boldsymbol{km^2}$

Table 1. The general properties of the dataset.

4.2 Evaluation metric

An essential point to evaluate the result of segmentation is the ability of the method to extract edges between different objects. Since the segmentation results are used as input to the classification stage, their evaluation metric is based on identifying the correct edges between objects. Therefore, the results are not analysed in terms of subdividing the objects into small parts. The segmentation results and the ground truth are projected onto a two-dimensional plane, since the detection of edges in three-dimensional space is imprecise. Then edges are detected for both the reference image and the segmentation image using morphological operators. The dilation operator extracts the outer edges, and the erosion operator extracts the inner edges. Finally, the edges of each class are individually compared in the segmentation image with the reference image and the following equations are used to calculate the segmentation error.

$$S = \frac{FE}{(FE+TE)} \quad (10)$$

$$\bar{S} = \frac{\sum_{1}^{n} S}{n} \quad (11)$$

In the above formula, FE is the number of edge points in the reference image that is not in the result image, and TE is the number of edge points in both images. In addition, S and n are the segmentation error for each class and the number of classes, respectively. Figure 4 shows the general steps of this metric with an example for the roof class.

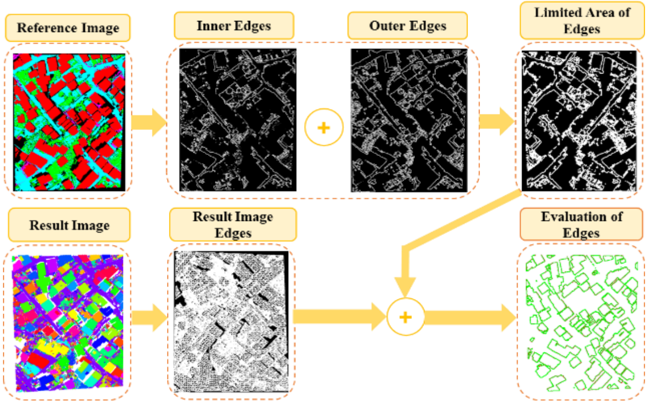

Figure 2. The general steps of the segmentation evaluation metric.

Using the confusion matrix, four main metrics for the classification evaluation are calculated as follows:

$$sensitivity = \frac{TP}{(TP+FP)} = \frac{Number\ of\ True\ Postive}{Number\ of\ all\ Postive} \quad (12)$$

$$Specificity = \frac{TN}{TN+FP} = \frac{Number\ of\ True\ Negative}{Number\ of\ all\ Negative} \quad (13)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (14)$$

$$Percision = \frac{TP}{TP+FP} \quad (15)$$

4.3 Segmentation Results

In this study, the segmentation method has only one parameter for segmenting the supervoxels (Xu et al., 2017). This parameter specifies the number of neighbors for each supervoxels to form a local graph. As shown in Table 2, segmentation results are evaluated based on different values for the number of neighbour's parameter. These values range from 3 to 12 supervoxels. Since each supervoxels is generated based on the initial voxel size of 1.2 and a point spacing of 0.3, each supervoxels contains many points. Therefore, constructing the local graph with more neighbors generates high computational cost. As shown in Table 2, changing the number of neighbors does not cause significant changes in segmentation error, and all values have an acceptable result. However, the computational cost for these values is different. In general, large the number of neighbors parameter for supervoxels generates large graphs. Therefore, segmentation process takes longer to segment local garphs. Consequently, according to the results in Table 2 and considering the computational cost, the number of 5 neighbors for each supervoxels was considered to form the local graph. Visualization of segmentation result are shown in Figure 5.

Number of Adjacent	S1 (Roof Class)	S2 (Tree Class)	S3 (Wall Class)	S4 (Ground Class)	S (Average Error)
12	0.2252	0.1171	0.1548	0.15	0.1618
11	0.225	0.1175	0.1541	0.1503	0.1617
10	0.2251	0.1171	0.1539	0.15	0.1615
9	0.2242	0.1167	0.1512	0.1498	0.1605
8	0.2214	0.1164	0.1512	0.1496	0.1603
7	0.2231	0.1146	0.1498	0.1486	0.159
6	0.2221	0.1134	0.1478	0.1481	0.1579
5	**0.2181**	**0.111**	**0.1424**	**0.1469**	**0.1546**
4	0.2183	0.1115	0.1431	0.1472	0.155
3	0.2186	0.1122	0.1433	0.1481	0.1558

Table 2. Experimental results for the number of neighbors parameter.

4.4 Classification Results

In this step, three classification methods, including Random Forest, Gradient Boosted Trees, and Bagging Decision Trees are used for final classification based on 18 geometric and spectral features. 30 percent of the data is considered for training. Table 4 is the results of these three classification methods according to the data part and classifier. As you can see, the classification results are roughly similar in all three methods, and the Random Forest is slightly better than the other two methods. In the random forest method, the roof, tree, wall, and ground classes have an average accuracy of 93.1, 89.13, 67.03, and 93.08, respectively. The reason for the similarity of the results of the three methods lies in the object-based classification. Since the segmentation step extracts individual objects and each object has specific properties, all three methods have similar results. Among the results obtained for the four classes, the two classes' tree and roof show higher overall accuracy due to their unique structure. Nevertheless, due to the vertical geometry of the data collection, the wall class has a low density and an irregular structure. Therefore, the wall class has the lowest accuracy among the classes. Classification results for 3 parts of data using these three methods are shown in Figure 6.

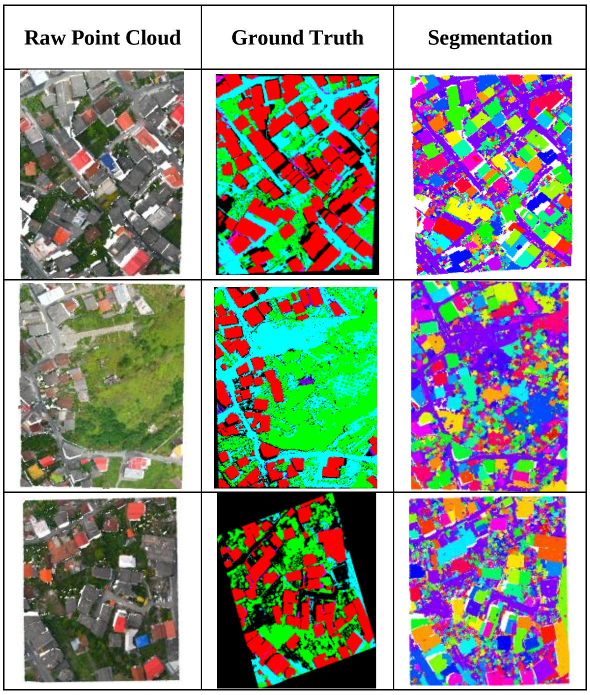

Figure 5. The results of segmentation method.

Geometric Features		Spectral Features	
Linearity	$L = \frac{e_1 - e_2}{e_1}$	**Mean R**	$(\sum_{i=1}^{n} \boldsymbol{R_i})/\boldsymbol{n}$
Planarity	$P = \frac{e_2 - e_3}{e_1}$	**Mean G**	$(\sum_{i=1}^{n} \boldsymbol{G_i})/\boldsymbol{n}$
Scattering	$S = \frac{e_3}{e_1}$	**Mean B**	$(\sum_{i=1}^{n} \boldsymbol{B_i})/\boldsymbol{n}$
Local curvature	$C = \frac{e_3}{e_1 + e_2 + e_3}$	**R ratio**	$\overline{\boldsymbol{R}} / \sum_{i=1}^{n} (\overline{\boldsymbol{R}} + \overline{\boldsymbol{G}} + \overline{\boldsymbol{B}})$
Omnivariance	$O = \sqrt[3]{e_1 \cdot e_2 \cdot e_3}$	**B ratio**	$\overline{\boldsymbol{B}} / \sum_{i=1}^{n} (\overline{\boldsymbol{R}} + \overline{\boldsymbol{G}} + \overline{\boldsymbol{B}})$
Anisotropy	$A = (e_1 - e_3)/e_1$	**G ratio**	$\boldsymbol{G} / \sum_{i=1}^{n} (\overline{\boldsymbol{R}} + \overline{\boldsymbol{G}} + \overline{\boldsymbol{B}})$
Eigen Entropy	$E = -\sum_{i=1}^{3} e_i \cdot \ln(e_i)$	**Maximum R difference**	$\boldsymbol{R_{max}} - \boldsymbol{R_{min}}$
Mean Z	$\overline{Z} = (\sum_{i=1}^{n} Z_i)/\boldsymbol{n}$	**Maximum B difference**	$\boldsymbol{B_{max}} - \boldsymbol{B_{min}}$
Z Variance	$(\sum_{i=1}^{n} Z_i - \overline{Z})/\boldsymbol{n}$	**Maximum G difference**	$\boldsymbol{G_{max}} - \boldsymbol{G_{min}}$

Table 3. Geometric and spectral features that are used in classification stage.

	Part1			**Part2**			**Part3**		
Method	Random Forest	Gradient Boosted trees	Bagging	Random Forest	Gradient Boosted trees	Bagging	Random Forest	Gradient Boosted trees	Bagging
Roof	**88.13**	87.38	88.64	**96.92**	96.55	96.79	94.54	**98**	94.25
Tree	**94.52**	93.52	90.29	89.99	87.19	**90.19**	82.88	**87.04**	82.48
Wall	**73.56**	72.45	71.44	**73.88**	64.24	68.97	**53.65**	48.70	51.69
Ground	**84.43**	83.77	83.93	**98.42**	97	98	96.40	95.52	**96.5**
Precision	**85.16**	84.05	83.57	**89.80**	86.48	88.49	81.87	**82.32**	81.23
Sensitivity	88.13	87.38	**88.64**	**61.44**	54.10	58.06	**93.20**	90.72	93.09
Specificity	**94.01**	93.13	92.16	**98.93**	98.5	98.72	97.13	**98.7**	96.9
Accuracy	**91.47**	90.64	90.63	**97.17**	96.4	96.81	95.78	**96.13**	95.63

Table 4 Experimental results of the classification methods

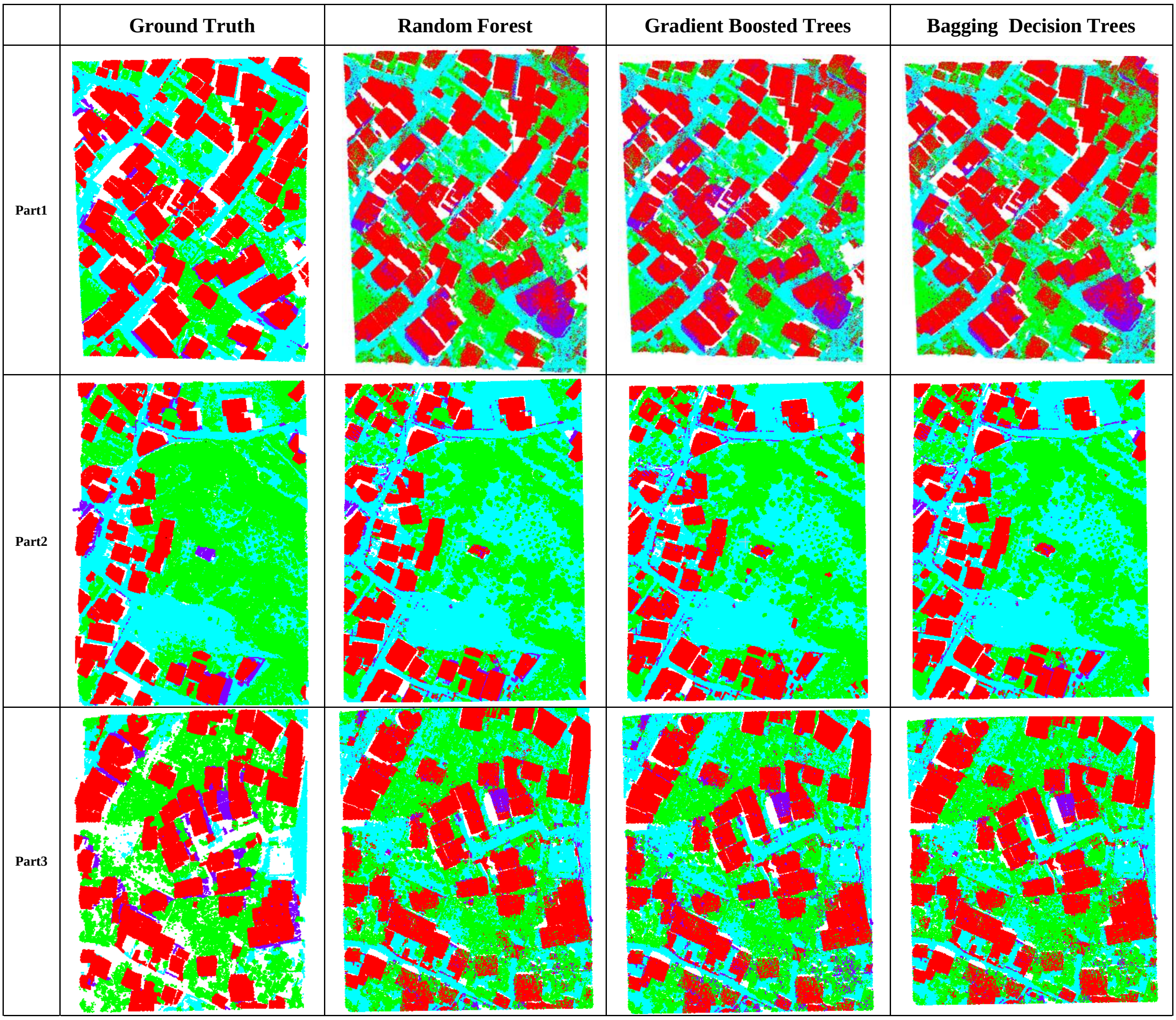

Figure 6. Visualization results of different parts with three classification methods. (Red: Roof, Green: Tree, Purpule: Wall, Cyan: Ground)

5. CONCLU/SION

In this research, we have presented a general framework for segmentation and classification of airborne laser scanning (ALS) point clouds obtained from a LiDAR-UAV sensor in urban area. Since point-based segmentation and classification processes need high computational cost, we have presented a framework based on supervoxel structure. supervoxel structure could reduce sensitivity to noise for segmentation and classification stages. For segmentation, a method based on local graphs was used (Xu et al., 2017), which has only one parameter, the number of neighbors for each supervoxel to form local graphs. This parameter depends on the dataset and must be specified with regard to point spacing and density. Experiments indicate the number of five neighbors has the best result for the segmentation stage, and the larger number of neighbors causes the higher segmentation error and computational cost. For classification stage, the segments are classified based on three methods, including Random Forest, Gradient Boosted Trees, and Bagging Decision Trees. The classification results indicate Random Forest is slightly better than the other two methods. Moreover, experiments prove object-based classification causes similar result for different classifiers. There are still points to improve the proposed framework. The first point is to use more efficient methods to classify the extracted segments. Currently, many methods using deep learning algorithms for semantic segmentation in 3D space have been reported. Features calculated by deep learning networks have significant advantages in terms of uniqueness over manual features. Moreover, some modern networks (like generative adversarial networks) make it possible to generate information in areas of data loss. Finally, the local graph structure has shown acceptable results for point cloud segmentation. Nevertheless, the computation cost of graph-based methods has always been considered as a disadvantage of these methods. Therefore, providing a graph-based structure with the optimal number of edges is vital for future research.

REFERENCES

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(11), 2274–2281. https://doi.org/10.1109/TPAMI.2012.120

Aldoma, A., Marton, Z. C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R., Gedikli, S., & Vincze, M. (2012). Tutorial: Point cloud library: Three-dimensional object recognition and 6 DOF pose estimation. *IEEE Robotics and Automation Magazine*, *19*(3), 80–91. https://doi.org/10.1109/MRA.2012.2206675

Belton, D., & Lichti, D. (2006). Classification and segmentation of terrestrial laser scanner point clouds using local variance information. *ISPRS Commission V Symposium 'Image Engineering and Vision Metrology*, *5*(1), 44–49. https://modelisation-environnement-forestier.googlecode.com/files/segmentation classification variance.pdf

Breiman, L. (2001). Random Forests. *Mach. Learn*.
Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, *59*(2), 167–181. https://doi.org/10.1023/B:VISI.0000022288.19776.77

Golovinskiy, A., & Funkhouser, T. (2009). Min-cut based segmentation of point clouds. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009*, 39–46. https://doi.org/10.1109/ICCVW.2009.5457721

Grilli, E., Menna, F., & Remondino, F. (2017). A review of point clouds segmentation and classification algorithms. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2W3), 339–344. https://doi.org/10.5194/isprs-archives-XLII-2-W3-339-2017

Hackel, T., Wegner, J. D., & Schindler, K. (2016). Contour detection in unstructured 3D point clouds. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 1610–1618. https://doi.org/10.1109/CVPR.2016.178

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Springer Series in Statistics The Elements of Statistical Learning - Data Mining, Inference, and Prediction. *Springer*, *2nd*. http://www.springerlink.com/index/D7X7KX6772HQ2135.pdf

Iman Zolanvari, S. M., & Laefer, D. F. (2016). Slicing Method for curved façade and window extraction from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *119*, 334–346. https://doi.org/10.1016/j.isprsjprs.2016.06.011

Klasing, K., Althoff, D., Wollherr, D., & Buss, M. (2009). Comparison of surface normal estimation methodsfor range sensing applications. *Proceedings - IEEE International Conference on Robotics and Automation*, 3206–3211. https://doi.org/10.1109/ROBOT.2009.5152493

Papon, J., Abramov, A., Schoeler, M., & Worgotter, F. (2013). Voxel cloud connectivity segmentation - Supervoxels for point clouds. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2027–2034. https://doi.org/10.1109/CVPR.2013.264

Rusu, R. B., Blodow, N., Marton, Z. C., & Beetz, M. (2009). Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, 1–6. https://doi.org/10.1109/IROS.2009.5354683

Schnabel, R., Wahl, R., & Klein, R. (2007). Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, *26*(2), 214–226. https://doi.org/10.1111/j.1467-8659.2007.01016.x

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905. https://doi.org/10.1109/34.868688

Sithole, G., & Vosselman, G. (2004). Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *59*(1–2), 85–101. https://doi.org/10.1016/j.isprsjprs.2004.05.004

Stein, S. C., Schoeler, M., Papon, J., & Worgotter, F. (2014). Object partitioning using local convexity. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 304–311. https://doi.org/10.1109/CVPR.2014.46

Tarsha-Kurdi, F., Landes, T., & Grussenmeyer, P. (2007). Hough-Transform and Extended Ransac Algorithms for Automatic Detection of 3D Building Roof Planes From Lidar Data. *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, *XXXVI*(1), 407–412.

Vo, A. V., 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. https://doi.org/10.1016/j.isprsjprs.2015.01.011

Vosselman, G., Coenen, M., & Rottensteiner, F. (2017). Contextual segment-based classification of airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *128*, 354–371. https://doi.org/10.1016/j.isprsjprs.2017.03.010

Wang, M., & Tseng, Y. H. (2011). Incremental segmentation of lidar point clouds with an octree-structured voxel space. *Photogrammetric Record*, *26*(133), 32–57. https://doi.org/10.1111/j.1477-9730.2011.00624.x

Weinmann, M., Jutzi, B., Hinz, S., & Mallet, C. (2015). Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, *105*, 286–304. https://doi.org/10.1016/j.isprsjprs.2015.01.016

Xu, Y., Hoegner, L., Tuttas, S., & Stilla, U. (2017). VOXEL- and GRAPH-BASED POINT CLOUD SEGMENTATION of 3D SCENES USING PERCEPTUAL GROUPING LAWS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*(1W1), 43–50. https://doi.org/10.5194/isprs-annals-IV-1-W1-43-2017

Yan, W. Y., Shaker, A., & El-Ashmawy, N. (2015). Urban land cover classification using airborne LiDAR data: A review. *Remote Sensing of Environment*, *158*, 295–310. https://doi.org/10.1016/j.rse.2014.11.001

Zhou, Z. H. (2012). Ensemble methods: Foundations and algorithms. *Ensemble Methods: Foundations and Algorithms*, 1–218. https://doi.org/10.1201/b12207