Automated Extraction of Urban Trees from Mobile LiDAR Point Clouds

Fan W.[a], Chenglu W.[a*], and Jonathan L.[ab]

[a]Fujian Key Laboratory of Sensing and Computing for Smart City and the School of Information Science and Engineering, Xiamen University, Xiamen, China
[b]Senior Member, IEEE, Dept. of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada

ABSTRACT

This paper presents an automatic algorithm to localize and extract urban trees from mobile LiDAR point clouds. First, in order to reduce the number of points to be processed, the ground points are filtered out from the raw point clouds, and the un-ground points are segmented into supervoxels. Then, a novel localization method is proposed to locate the urban trees accurately. Next, a segmentation method by localization is proposed to achieve objects. Finally, the features of objects are extracted, and the feature vectors are classified by random forests trained on manually labeled objects. The proposed method has been tested on a point cloud dataset. The results prove that our algorithm efficiently extracts the urban trees.

Keywords: Point clouds, localization, segmentation, classification, urban trees

1. INTRODUCTION

Urban trees are an important part of a city. They bring benefits in five aspects: social benefits, architectural benefits, climatic benefits, ecological benefits, and economic benefits. They improve home and work environment, add beauty for architectural building, bring more pleasure to the environment, offer biotopes for flora and fauna, and boost the development of economy in a city.[1,2] Thus, the monitoring of urban trees is very essential.

Traditional way for monitoring urban trees is dependent on an inspection method, which consumes much manual labor work and time. In recent years, some automated methods based on mobile LiDAR systems have been proposed and they are based on the Euclidean clustering,[3],[4],[5] supervoxels clustering,[6] and the graph clustering,[7].[8]

In,[3] raw point clouds were partitioned along road directions as sub-regions called road parts. Then the segmented points were roughly labeled into ground, on-ground and off-ground by a surface growing algorithm.[9] Next, a connected component analysis was used to obtain the unique IDs. Finally, prior knowledge was employed to classify them. Rutzinger et al.[4] first removed planar and large regions with the method in.[9] Next, Rutzinger et al. clustered objects by a connected components technology. Yu et al.[5] first used Euclidean clustering method to obtain objects, and employed the pair wise 3D shape context[10] to describe detected feature points. Next, the template matching was employed to find urban trees. Pu et al. used Euclidean clustering method and prior knowledge to extract urban trees. In,[6] point clouds were first partitioned into multi-scale supervoxels according to the attributes of points and a spatial distance between points. Next, Yang et al. obtained the geometric structure information of each supervoxel by Principle Connectivity Analysis (PCA). Then the predefined rules are applied to merge adjacent supervoxels to become a complete object. Finally, semantic knowledge was employed to classify the objects. Livny et al.[7] first removed the ground points by projection density of raw point clouds, and the remaining points were clustered by the graph clustering. Golovinskiy et al. localized objects by a technology based on a graph; then, the min cut segmentation[11] was employed to obtain the objects. Finally, the classifiers trained on manually labeled objects were used to classify the objects.

Further author information: (Send correspondence to Chenglu Wen)
Chenglu Wen: E-mail: clwen@xmu.edu.cn

2nd ISPRS International Conference on Computer Vision in Remote Sensing (CVRS 2015), edited by Cheng Wang, Rongrong Ji, Chenglu Wen, Proc. of SPIE Vol. 9901, 99010P · © 2016 SPIE · CCC code: 0277-786X/16/$18 · doi: 10.1117/12.2234795

However, the processing unit of above most algorithms is a point. As a result, the time complexity of these algorithms is too high. On the other hand, most popular segmentation method, such as Euclidean clustering, is not robust to an occluded environment. Thus, the algorithms based on the segmentation method is limited. To solve the two problems, we develop an efficient algorithm to extract urban trees. First the processing unit of point clouds becomes a supervoxel rather than a point and it extremely accelerates process of extraction of urban trees. Second our proposed segmentation method is dependent on the localized positions and robust for the complex environment. Therefore, the classification of urban trees based on our segmentation method can obtain satisfying results.

Our algorithm can be decomposed into five steps: (1) Removing points close to the ground by iterative plane; (2) Segmenting un-ground points into supervoxels by the Voxel Cloud Connectivity Segmentation method; (3) Localizing the urban trees; (4) Segmenting by localization (5) Finally, the random forests trained by manually labeled objects is used to classify objects.

2. METHOD

2.1 Preprocessing

2.1.1 Ground removal

To reduce the number of points to be processed, ground points should be filtered out from raw point clouds, the the Random sample consensus (RANSAC) algorithm[12] that is modified by us is employed to remove ground points. For each RANSAC iteration, the points near the fitting plane are classified into ground points. The ground height h_{ground} is obtained in the first RANSAC algorithm. The iteration continues until one of the point near the fitting plane is higher than the $h_{ground} + 1$. Figure. 1 shows the result of ground removal.

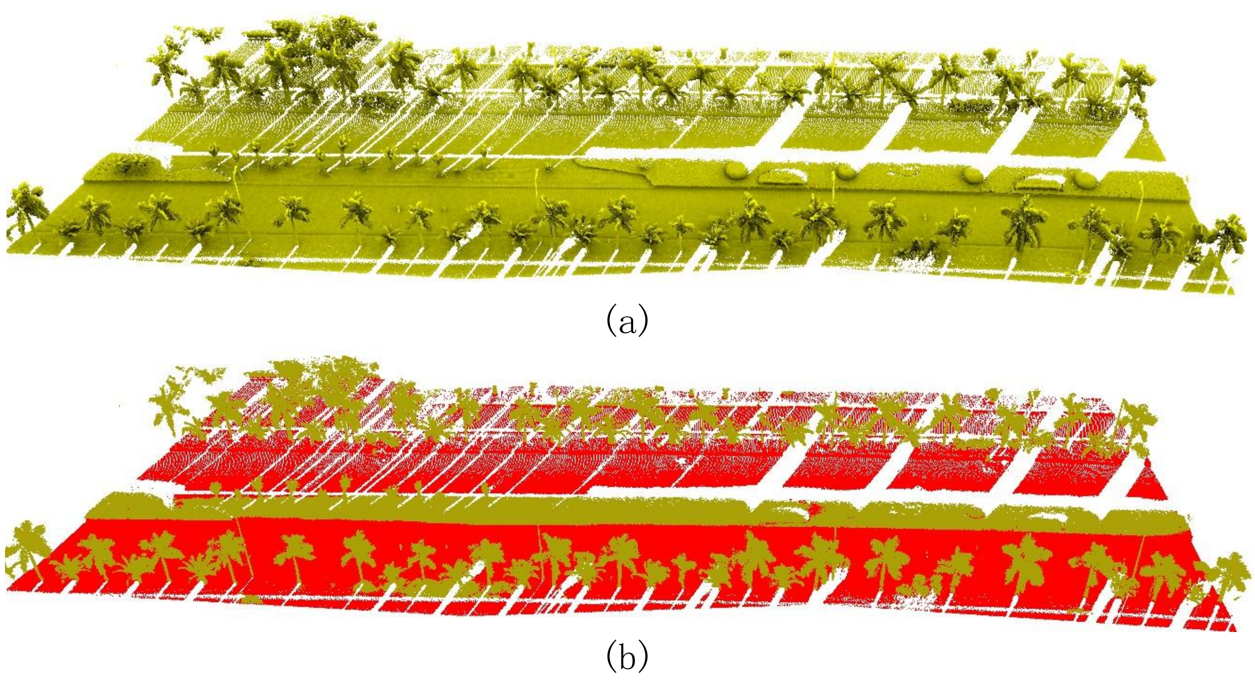

Figure 1. results of filtering ground. (a) raw point clouds. (b) filtering result

2.1.2 Generating supervoxels of non-ground points

When the ground points are separated from the raw point clouds, the un-ground points are segmented into supervoxels through the Voxel Cloud Connectivity Segmentation method[13] and the result can be shown in Figure. 2.

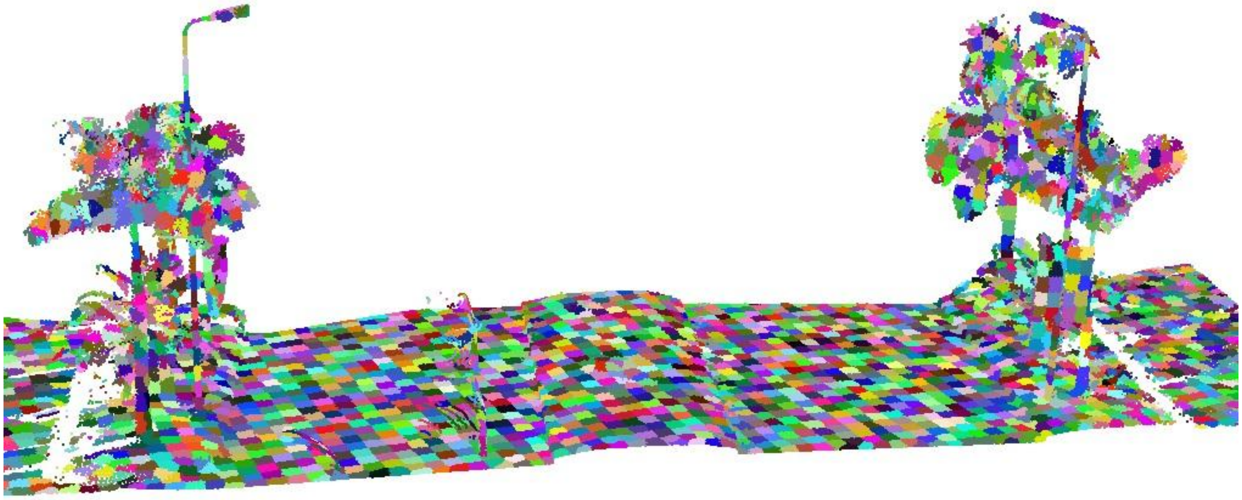

Figure 2. Supervoxel examples of a point cloud

After the un-ground points are segmented into supervoxels, the points in the supervoxel are projected onto the XY plane. Then, the algorithm in[14] is used to obtain the convex hull for the projected points. After the convex hull is achieved, the directed area of triangular is employed to obtain the area of the polygon of the convex hull.

2.2 Localization

The raw point clouds are partitioned into cells along the x and y coordinate, and the maximum z coordinate of the points in each cell can be obtained. Only the cells whose maximum z coordinate is in a pre-defined range (h_{low}, h_{high}) are retained. For each point $p(p_x, p_y, p_z)$ in the cell c, $s(p_z)$ is acquired by (1). The function $s(x)$ is defined as follows:

$$s(x) = \frac{1}{exp(-(x - \frac{h_{tree}}{2}) + 1)} \tag{1}$$

where h_{tree} is the estimated height of urban trees. Let $P(c)$ calculate the sum of $s(p_z)$ of the point p in the cell, c. Next the maximum of $P(c)$ can be obtain and it is γ.

Afterwards the localization image Img can be achieved as following:

$$Pixel(i,j) = \frac{P(c)}{\gamma} \cdot 255 \tag{2}$$

i and j is the index of cell c. Then a threshold gray value is set to obtain the connected areas. The position of the maximum gray value of a connected area is taken as the position of a object.

2.3 Segmentation

Before the segmentation starts, it is necessary to calculate the highest point, lowest point, bounding box, and barycenter of a supervoxel. After the un-ground points are segmented into supervoxels, the plane coordinate of the gravity center, $p_g(x, y)$, for each supervoxel is calculated as follows:

$$\begin{aligned} x &= \frac{\sum_{i=1}^{n} x_{p_i}}{n} \\ y &= \frac{\sum_{i=1}^{n} y_{p_i}}{n} \end{aligned} \tag{3}$$

where $p_i(x_{p_i}, y_{p_i}, z_{p_i})$ belongs to the supervoxel, β; n is the number of points in the supervoxel, β. There are three steps in our segmentation method. The order of merging supervoxels in the three steps decreases according to the corresponding pixel value of Img for the detected positions.

Step 1. The first step is to obtain tree trunks. It is believed that if the plane coordinate of the gravity center of a supervoxel is near a detected position and if more points in the supervoxel is near a detected position, this supervoxel would be more likely to belong to the detected position. And this is the thought of the first step. The function $dis(a, b)$ measures the plane distance between point a and point b and is defined as follows:

$$dis(a,b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2} \tag{4}$$

Firstly it is need to obtain the unclassified supervoxel, P_{patch}, whose plane distance between the gravity center, $p_g(x, y)$ of the supervoxel, P_{patch} and the plane coordinate of a detected position, p_{pole} is less than a predefined value, d_g. Afterwards, the ratio ϕ for P_{patch} can be obtained. The ratio, ϕ, is defined as follows:

$$\begin{aligned} &P_{in} = \{p | dis(p, p_{pole}) < d_{in},\ p \ \in \ P_{patch}\} \\ &\phi = \frac{Count(P_{in})}{n} \end{aligned} \tag{5}$$

where d_{in} is less than d_g. $Count(x)$ measures the number of elements in the set, x; n is the number of the points in P_{patch}. If the value of ϕ is greater than λ, the supervoxel, P_{patch}, would temporiably belong to the tree trunk object at the detected position, p_{pole}. If the number of supervoxels, like P_{patch}, belonging to the object is less than N_{own}, these supervoxels .become unclassified supervoxels again. When the pole is separated from the scene, the number of the points in the pole and the height of the tree trunk can be achieved. If the number of the points in the pole is less than a pre-defined value or the height of the pole is less than a pre-defined value, the pole is not taken as the tree trunk. The highest point of the tree trunk is P_{summit}.

Step 2. The second step is to filter out the vegetation near the tree trunk. First the supervoxels whose plane center is in the pre-defined plane range at each position of the detected tree trunk are found. Next, the supervoxels whose highest point is lower than a pre-defined value are removed.

Step 3. The third step is to achieve the complete tree. This step is motivated by the breadth-first search algorithm. The supervoxels can expand towards up and down. The patch with the peak point, $P_{peak}(x_{peak}, y_{peak}, z_{peak})$, is taken as the seed patch, and added into the queue. Next, the top element of the queue is taken as the growing patch, and the top element is removed from the queue. Sequentially, the neighboring patches whose barycenter is achieved in the r_{grow} meter range at the growing patch center. If the maximum in distances from the eight plane projection corners of the bounding box of the neighbor patch to the p_{pole} is less than r_{safe}, then the neighboring patch is added to the queue. After all the neighboring patches have been judged, the first element of the queue becomes the new growing patch. This process continues until the queue is empty.

2.4 Feature Extraction

After the segmentation process terminates, the features of the object are calculated. There are two kinds of features: pole features and whole object features.

Tree trunk features. When the first step of segmentation ends, the following nine features that describe the pole are computed: (1) the height; (2) the average height; (3) the standard deviation in height; (4) the average area of the convex hull of supervoxel's plane projection points; (5) the standard deviation in the area of the convex hull of supervoxel's plane projection points; (6) the area of the convex hull of the whole object's plane projection points; (7) the estimated volume; (8) the number of pole points; and (9) the number of the supervoxel whose area of the convex hull of the supervoxel's plane projection points is greater than the standard area of the convex hull of supervoxel's plane projection points of an urban tree, s_{base}.

Whole tree features. After the second step of segmentation is completed, the following ten features that describe the whole object are computed: (1) the height; (2) the average height; (3) the standard deviation in height; (4) the pixel value in the corresponding location image position; (5) the projection convex hull area; (6) the estimated volume; (7) the height difference of the barycenter and geometry center; (8) the number of points; (9) the number of neighboring patches in one meter range at the peak position of the first segmented pole.

2.5 Classification

In the final stage, random forests trained on manually labeled objects are employed to classify the objects.

3. RESULTS AND DISCUSSION

The point clouds were acquired by a RIEGL VMX-450 system. They were acquired on the Ring Road South in Xiamen, China. This is a typical urban scene and many trees are beside street light poles, which causes that the extraction of trees is difficult.

First, the ground points were removed from the raw point clouds by multiple RANSAC methods. Next, the non-ground points are segmented into supervoxels. Subsequently, a novel localization method was proposed to localize urban trees with $h_{low} = 2.0m$, $h_{high} = 8.0m$, and $h_{tree} = 8.0m$. Next, a segmentation by localization was proposed to obtain the urban trees with $d_g = 1.5m$, $d_{in} = 0.5m$, $\lambda = 0.5$, $r_{grow} = 1.1m$, and $r_{safe} = 2.4m$. The localization and extraction results are shown in Figure. 3. Finally, the random forests trained on manually labeled objects were used to classify objects. The results of extracting urban trees are shown in Figure. 4 and the ground true, detection result, and quantitative evaluation is presented in Table. 1.

Figure 3. localization and segmentation results

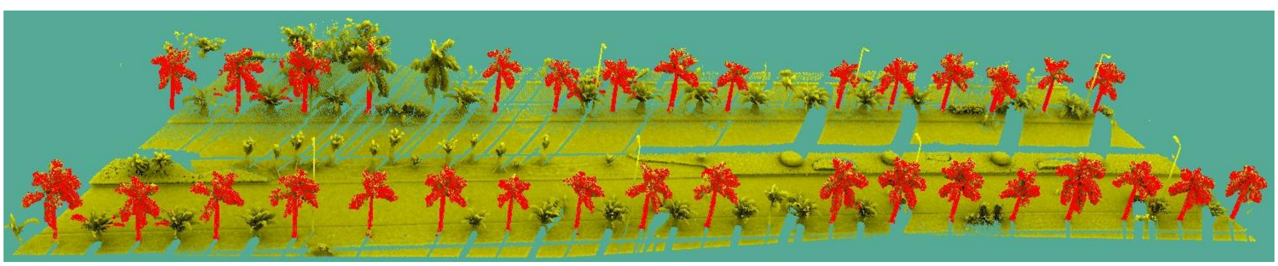

Figure 4. extraction of urban trees

Table 1. Quantitative Evaluation Result using random forests

Ground Truth	Detection Result		Quantitative Evaluation			
Urban Trees	Urban Trees	False Positive	Completeness	Correctness	Quality	F_1-measure
311	268	33	86.2%	89.0%	77.9%	87.6%

4. CONCLUSION

In this paper, we have proposed a novel and efficient method to extract urban trees from mobile LiDAR point clouds. Even though our algorithm was tested in a complex environment, the segmentation result is still good. The segmentation is the key step for the classification of urban trees, for the extracted features largely rely on the segmentation result. On the other hand, the proposed features are suitable and they can be employed to classify trees and non-tree objects. Therefore, our algorithm about extraction of urban trees is very promising.

REFERENCES

[1] Burden, D., "Urban street trees: 22 benefits, specific applications." http://www.michigan.gov/documents/dnr/22_benefits_208084_7.pdf.

[2] Tyrväinen, L., Pauleit, S., Seeland, K., and de Vries, S., "Benefits and uses of urban forests and trees," in [*Urban forests and trees*], 81–114, Springer (2005).

[3] Pu, S., Rutzinger, M., Vosselman, G., and Elberink, S. O., "Recognizing basic structures from mobile laser scanning data for road inventory studies," *ISPRS Journal of Photogrammetry and Remote Sensing* **66**(6), S28–S39 (2011).

[4] Rutzinger, M., Pratihast, A., Oude Elberink, S., and Vosselman, G., "Detection and modelling of 3d trees from mobile laser scanning data," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci* **38**, 520–525 (2010).

[5] Yu, Y., Li, J., Guan, H., and Wang, C., "Automated extraction of urban road facilities using mobile laser scanning data," *Remote Sensing of Environment* (2015).

[6] Yang, B., Dong, Z., Zhao, G., and Dai, W., "Hierarchical extraction of urban objects from mobile laser scanning data," *ISPRS Journal of Photogrammetry and Remote Sensing* **99**, 45–57 (2015).

[7] Livny, Y., Yan, F., Olson, M., Chen, B., Zhang, H., and El-Sana, J., "Automatic reconstruction of tree skeletal structures from point clouds," *ACM Transactions on Graphics (TOG)* **29**(6), 151 (2010).

[8] Golovinskiy, A., Kim, V. G., and Funkhouser, T., "Shape-based recognition of 3d point clouds in urban environments," in [*Computer Vision, 2009 IEEE 12th International Conference on*], 2154–2161, IEEE (2009).

[9] Vosselman, G., Gorte, B. G., Sithole, G., and Rabbani, T., "Recognising structure in laser scanner point clouds," *International archives of photogrammetry, remote sensing and spatial information sciences* **46**(8), 33–38 (2004).

[10] Yu, Y., Li, J., Yu, J., Guan, H., and Wang, C., "Pairwise three-dimensional shape context for partial object matching and retrieval on mobile laser scanning data," *Geoscience and Remote Sensing Letters, IEEE* **11**(5), 1019–1023 (2014).

[11] Golovinskiy, A. and Funkhouser, T., "Min-cut based segmentation of point clouds," in [*Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*], 39–46, IEEE (2009).

[12] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM* **24**(6), 381–395 (1981).

[13] Papon, J., Abramov, A., Schoeler, M., and Worgotter, F., "Voxel cloud connectivity segmentation-supervoxels for point clouds," in [*Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*], 2027–2034, IEEE (2013).

[14] Andrew, A. M., "Another efficient algorithm for convex hulls in two dimensions," *Information Processing Letters* **9**(5), 216–219 (1979).