Remote Sensing Letters

ISSN: 2150-704X (Print) 2150-7058 (Online) Journal homepage: http://www.tandfonline.com/loi/trsl20

A supervoxel approach to the segmentation of individual trees from LiDAR point clouds

Sheng Xu, Ning Ye, Shanshan Xu & Fa Zhu

To cite this article: Sheng Xu, Ning Ye, Shanshan Xu & Fa Zhu (2018) A supervoxel approach to the segmentation of individual trees from LiDAR point clouds, Remote Sensing Letters, 9:6, 515-523, DOI: 10.1080/2150704X.2018.1444286

To link to this article: https://doi.org/10.1080/2150704X.2018.1444286

REMOTE SENSING LETTERS, 2018
VOL. 9, NO. 6, 515–523
https://doi.org/10.1080/2150704X.2018.1444286

Taylor & Francis
Taylor & Francis Group

A supervoxel approach to the segmentation of individual trees from LiDAR point clouds

Sheng Xu[a], Ning Ye[a], Shanshan Xu[a] and Fa Zhu[b]

[a]College of Information Science and Technology, Nanjing Forestry University, Nanjing, China; [b]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

ABSTRACT

This letter aims to propose an automatic method for the segmentation of street trees from LiDAR point clouds. The first step is to extract tree points to reduce the scale of data to be processed. The second step is to formulate supervoxels for grouping homogeneous points. The third step is to optimize the segmentation of individual trees based on the minimum distance rule. Each supervoxel will be assigned the same tree index as its spatially closest treetop supervoxel, therefore, points from the homogeneous region sharing the same tree index. Experiments show that we achieve the completeness of 78.5%, correctness of 94.5% and *F*-score of 0.85 in the airborne LiDAR data. For mobile LiDAR data our accuracy is 100, 93.8% and 0.96 in terms of the recall, precision and *F*-score, respectively.

ARTICLE HISTORY
Received 23 November 2017
Accepted 16 February 2018

1. Introduction

The individual tree segmentation aims to derive the structural attributes of the tree, which plays an important role in the biomass productivity estimation (Yun et al. 2016) and the forest management (Xia et al. 2015). Nowadays, the laser scanning technique succeeds to provide an accurate 3D information of high-density point clouds which allows for a better street tree segmentation. In the following, typical methods proposed for the tree segmentation from LiDAR point clouds will be reviewed and analyzed.

Li et al. (2012) proposed a top-to-bottom region growing approach to segment trees from LiDAR point clouds. First, they obtained the height from ground to each point by subtracting the ground points. Then, they segmented trees by starting from a treetop point and grouped its nearby points by setting a distance threshold. The problem is that points at the lower elevation are difficult to be classified using a fixed threshold in complex environments. Véga et al. (2014) proposed a method called PTrees to the segmentation from LiDAR point clouds. Similar to Li et al. (2012), they used the elevation value to process points from the highest to the lowest point. Their strategy is to search treetop points within a neighborhood window first and then perform a region-growing strategy to delineate tree crowns. Treetop points are generated at different segmentation scales, therefore, their algorithm can select an optimal treetop points in the

CONTACT Shanshan Xu shanshanxus@gmail.com College of Information Science and Technology, Nanjing Forestry University, Nanjing, China

segment generation. Mongus and Žalik (2015) proposed a clustering-based method for delineating individual trees from LiDAR data. First, they detected treetop points by defining a canopy height model using the shape information. Then, they achieved the delineation of additional tree crowns from different layers by estimating points' distributions. Their search for treetop points does not depend on the shape or size of tree crowns. Hamraz, Contreras, and Zhang (2016) proposed a non-parametric method for segmenting individual trees based on the crown shape and height of the vegetation. They segmented trees by starting from the tallest tree within a given area to the smallest until all trees have been segmented. The advantage is that they do not require priori assumptions of crown shape and size.

The target of this letter is to generate supervoxels to tree points and provide an automatic method for segmenting the individual tree from LiDAR point clouds.

2. Methodology

2.1. *Data preprocessing*

The outlier removal is based on the calculation of the mean μ and standard deviation σ of k-nearest neighbors. In our work, points between $\mu - t_1\sigma$ and $\mu + t_1\sigma$ are regarded as valid points. To improve the efficiency of data processing, ground points are filtered before the segmentation. In the road environment, ground points are more dense than off-ground points. Thus, peaks in the elevation histogram can be used to find the threshold for the filtering task. Details of selecting the peak for the ground point removal are shown in the work of Xu et al. (2018). The final task in the preprocessing step is the tree point extraction, which is implemented by classifying off-ground point into tree points and non-tree points. We split off-ground points into multilevel hierarchical cluster sets first, and then the shape features of the multilevel point clusters are extracted based on the work of of Zhang et al. (2016). In the classifier selection, we choose the support vector machine (SVM) (Cortes and Vapnik 1995; Zhu et al. 2016) to distinguish trees (positive) and others (negative).

2.2. *Supervoxel generation*

In our work, the supervoxel is defined as a polyhedral region consisting of homogeneous points. Each supervoxel has one center point and there are no overlapping points between different supervoxels. We define three matrices in the supervoxel generation. The first one is $\mathbf{S}_{n\times n}$ called the neighbor distance matrix, where n is the number of points. The element $S(p,q)$, which is indexed at (p,q) in the matrix $\mathbf{S}$, measures the similarity between the point p and q by

$$S(p,q) = \mathrm{e}^{-d(p,q)}, \tag{1}$$

where $d(p,q)$ means the Euclidean distances between the point p and q. If the point p and q are spatially close, the similarity $S(p,q)$ tends to be 1, otherwise, it approaches 0.

The second one is $\mathbf{M}_{n\times n}$ named the attraction matrix and its element $M(p,r)$ is calculated by

$$M(p,r) = S(p,r) - \max_{r' \neq r}(N(p,r') + S(p,r')). \tag{2}$$

We assume that $S(p,r)$ not only measures the similarity between p and r, but also shows the potential of choosing the point r as the supervoxel center of p. Besides, we define the third matrix $\mathbf{N}_{n\times n}$ as the assignment matrix and its element $N(p,r')$ shows the potential of assigning the point p to the supervoxel center r'. Then, the second term in Equation (2) is to calculate the potential of choosing the point r' as the center of a supervoxel containing the point p. Therefore, $M(p,r)$ shows the superiority of using the point r as the supervoxel center of p over the point r', i.e. the attraction of r as the center of a supervoxel containing p. The element $N(p,r)$ in the defined assignment matrix is calculated by

$$N(p,r) = \sum_{p' \neq p} M(p',r). \tag{3}$$

In our work, if the potential of choosing r as the supervoxel center for all other points p' is high, the potential of selecting r as the supervoxel center of p will be also high.

At the beginning, we regard each point as a candidate supervoxel center. Our goal is to assign non-center points to the optimized supervoxel centers to generate complete supervoxels. In the generation, supervoxel centers are updated iteratively based on the attraction and assignment matrix, which is similar to the update of centers in k-means clustering. The following Algorithm 1 shows the supervoxel generation from point clouds.

Algorithm 1. Supervoxel formulation.

1: Initialize the neighbor distance matrix **S** by Equation (1);
2: Initialize the assignment matrix **N** as zero;
3: Update the attraction matrix **M** by Equation (2);
4: Update the assignment matrix **N** by Equation (3);
5: Repeat 3–4 until the matrix **T** = (**M** + **N**) converges;
6: Find the point r^* to achieve the maximum element in the i-th row of **T**, i.e. $T(i,r^*) > T(i,r'), r' \neq r^*$;
7: Mark r^* as a supervoxel center;
8: Repeat 6–7 until each row of **T** has been decided;
9: Assign all other points to supervoxel centers based on the Euclidean distance.

2.3. *Tree segmentation*

In the search of the treetop location, we first use a non-overlapping sliding window (size: s m $\times$ s m) to extract the local highest supervoxel as a candidate treetop supervoxel. The elevation of a supervoxel is decided by its center point. Then, we use a larger non-overlapping sliding window (size: $2s$ m $\times$ $2s$ m) to remove treetop supervoxels that are lower than $(\mu_s + t_2\delta_s)$, where μ_s and δ_s are the mean and standard deviation elevation of candidate supervoxels in the current window, respectively. After this, if the distance between two treetop supervoxels' centers is smaller than s m, we will remove the lower one. The illustration of our filtering method is shown in Figure 1. A non-treetop supervoxel will be assigned the same tree index as its closest treetop supervoxel. Figure 2(a) shows an input scene consisting of several individual trees. Our generated supervoxels are visualized by different colors in point clouds as shown in Figure 2(b) and the red dots

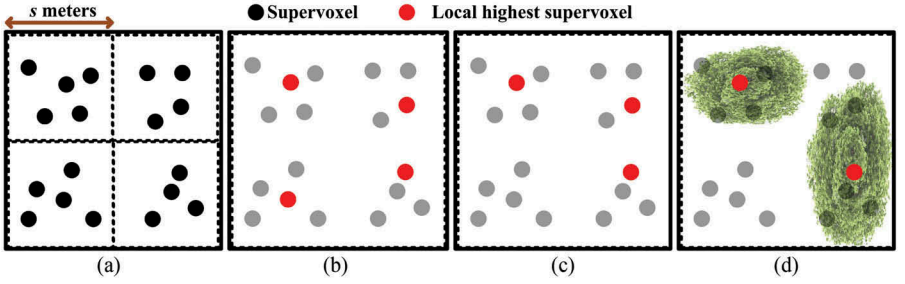

Figure 1. The filtering of treetop supervoxels. (a) Formulated supervoxels (black dot). (b) Found candidate treetop supervoxels (red dot) by filtering with a sliding window (size: s m $\times$ s m). (c) Updated candidate treetop supervoxels (red dot) by filtering with a sliding window (size: $2s$ m $\times$ $2s$ m). (d) Refined treetop supervoxels (red dot) by adding the constraint of Euclidean distances between centers.

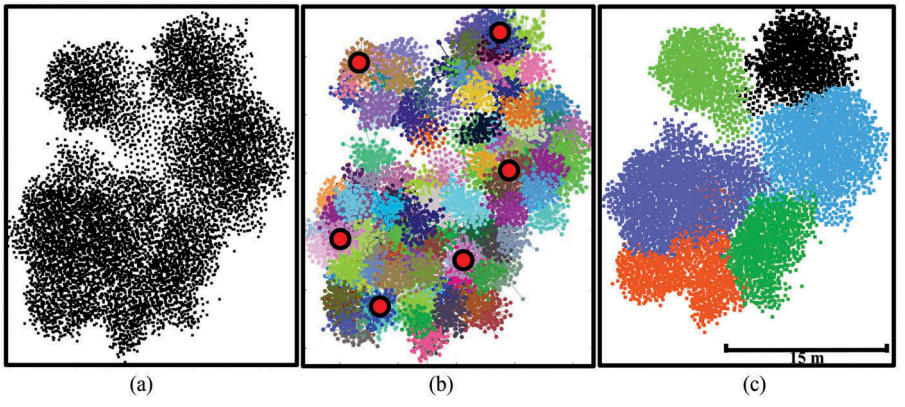

Figure 2. The assignment of supervoxels in the tree segmentation. (a) Input point clouds. (b) Detected treetop supervoxels (red dot). (c) Achieved tree segmentation results based on the found treetops.

show the detected treetop supervoxels achieved by filtering the elevation. Figure 2(c) is the result of the individual tree segmentation.

Figure 3 uses a simulated example to show the outcome of each step in the proposed method. Figure 3(a) shows the input LiDAR point clouds. Figure 3(b,c) are the extracted ground points and off-ground points, respectively. Figure 3(d) presents the classified tree points (green) and other points (black). Figure 3(e) shows the generated supervoxels for tree points. Figure 3(f) demonstrates the final results of the tree segmentation.

3. Experiments and evaluation

This section tests the proposed algorithm on two LiDAR point cloud sets. The first set is from the 2015 Dublin LiDAR project (doi:10.17609/N8MQ0N) collected by the airborne LiDAR scanning (ALS) system. Our segmented individual trees are illustrated by different colors as shown in Figure 4(a,b).

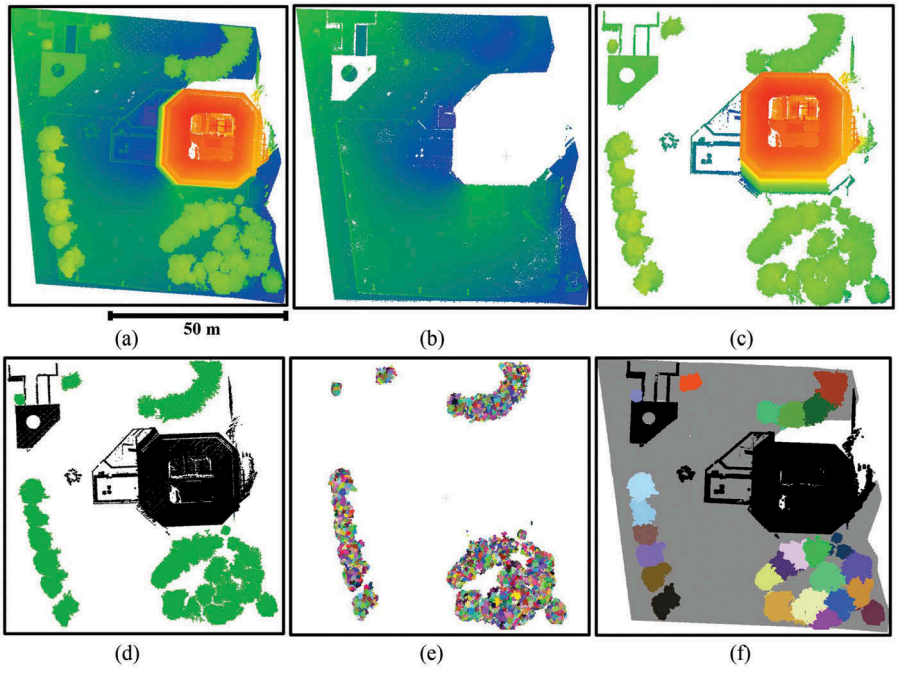

Figure 3. The outcome of each step in the proposed tree segmentation. (a) Input point clouds. (b) Extracted ground points. (c) Extracted off-ground points. (d) Classified tree points (green) and others (black). (e) Generated supervoxels for tree points. (f) Obtained tree segmentation results.

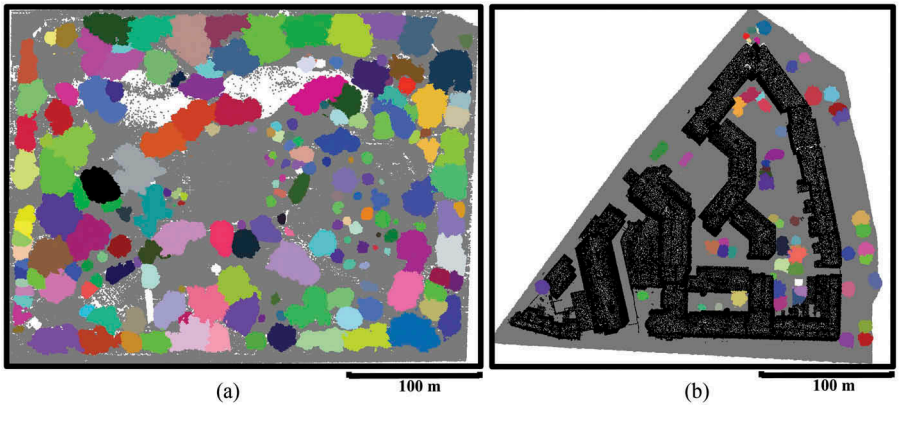

Figure 4. Test on the ALS data collected in Dublin, Ireland (location: 53°20′16″N, 6°15′32″W). (a) Achieved tree segmentation results from the St Stephen's Green park (area: 371 m × 292 m, number of points: 1,724,926). (b) Achieved tree segmentation results from the street scene (area: 278 m × 219 m, number of points: 244,831).

The second set is from a dense urban environment in Paris (Vallet et al. 2015) collected by the mobile LiDAR scanning (MLS) system. Performance of the proposed tree segmentation is shown in Figure 5(a–c) are results of the tree segmentation in the side view and top view, respectively. Our errors are caused by the tree point

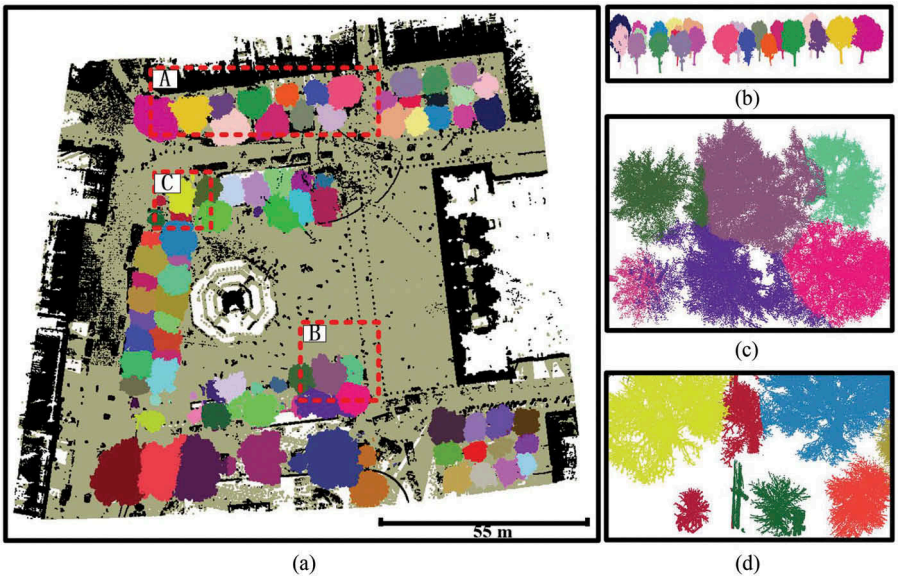

Figure 5. Test on the MLS data collected in Paris, France (location: 48°51′01.3″N, 2°19′57.9″E, area: 150 m × 134 m, number of points: 19,930,860). (a) Achieved tree segmentation results from the road scene. (b) The side view of the region A in (a). (c) The top view of the region B in (a). (d) The close-view of the region C in (a).

classification. As shown in Figure 5(d), pole-like objects are classified as trees and are wrongly detected as treetops.

In the evaluation, the reference of individual trees is manually segmented from input points. If a tree is correctly segmented, it is called true positive; if a tree is not segmented but assigned to a nearby tree, it is called false negative; if a tree does not exist but is wrongly segmented from the point cloud by us, it is called false positive. The evaluation is achieved by analyzing the recall R, precision P and F-score F as

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, F = 2 \times \frac{R \times P}{R + P}, \tag{4}$$

where TP, FN and FP are the number of segments in true positive, false negative and false positive, respectively. The recall measures the probability of trees that can be extracted by our method, and the precision measures the probability of our segments that are trees in the reference. The comparison of the proposed algorithm and others is shown in Table 1.

Experiments show that the proposed algorithm achieves very competitive results in the individual tree segmentation. We do not have complex neighbor information calculation, e.g. the voxelization in (Wu et al. 2013; Li et al. 2016) or the octree construction in (Zhong et al. 2017). Our supervoxel-based assignment is more efficient and robust than the point-based assignment strategy in (Tao et al. 2015; Li et al. 2012; Véga et al. 2014). Points from the same supervoxel are homogeneous and will be assigned the same index simultaneously. Moreover, we do not require providing the height model or the shape information in the segmentation as in (Mongus and Žalik

Table 1. Comparison of the accuracy of different methods.

Method	Scanner	*R* (%)	*P* (%)	*F*
Li et al. (2012)	ALS	86.0	94.0	0.89
Véga et al. (2014)	ALS (average)	82.7	89.7	0.86
Mongus and Žalik (2015)	ALS	56.0	77.0	0.64
Hamraz, Contreras, and Zhang (2016)	ALS	72.0	86.0	0.78
Proposed	ALS (Park scene)	74.0	92.0	0.82
	ALS (Street scene)	83.0	97.0	0.89
	ALS (average)	78.5	94.5	0.85
Wu et al. (2013)	MLS (average)	99.3	100	0.99
Tao et al. (2015)	MLS (average)	98.0	100	0.98
Li et al. (2016)	MLS (average)	98.0	96.0	0.96
Zhong et al. (2017)	MLS	94.0	93.7	0.93
Proposed	MLS	100	93.8	0.96

2015; Hamraz, Contreras, and Zhang 2016). In the ALS data test, we achieve the highest precision in the experimental scene. False negative errors, e.g. the short trees and spatially close trees, can be improved by using adaptive sliding windows for different scales of trees. In the MLS data test, we achieve the highest recall in the experimental scene. False positive errors, e.g. streetlights and traffic signs, caused by the tree point classification can be removed based on the point distribution.

4. Discussion

To evaluate the applicability and robustness of the proposed pipeline, we test it on the ALS and MLS data. It took us 477.74 seconds to extract 186 out of 250 trees in Figure 4 (a), 79.93 seconds to extract 58 out of 70 trees in Figure 4(b) and 212.51 seconds to extract 90 out of 90 trees in Figure 5(a). Experiments were done on a Windows 10 Home 64-bit, Intel Core i5-7200U 2.5GHz processor with 32 GB of RAM and computations were carried on Matlab R2017a.

Table 2 organizes all required parameters in the proposed method. In order to improve the efficiency of the segmentation, we set the number of neighbors k and the truncation thresholds t_1 and t_2 as constants. The $S(p,p)$ is set as the median value of $\mathbf{S}$ to adapt different scenes. We assume that two treetops in the road environment are over $s = 2.5$ m. Parameters can be varied in different scenes, therefore, one may need to tune them by experiments for achieving the optimal results.

The proposed supervoxel approach is a general method, which only requires setting the distance value $S(p,p)$. The size of the generated supervoxel depends on $S(p,p)$, which can be decreased by increasing the value of $S(p,p)$. Supervoxels have the similar size after setting an appropriate $S(p,p)$, e.g. the median value of $\mathbf{S}$ in our work. The efficiency of the supervoxel generation relies on the update of $\mathbf{M}$ and $\mathbf{N}$, which incurs

Table 2. Parameter setting used in the different stages of the proposed method.

Stage	Parameter	Value	Unit
Data preprocessing	k	40	points
	t_1	1.5	N/A
Supervoxel formulation	$S(p,p)$	median value of $\mathbf{S}$	N/A
Tree segmentation	s	2.5	meters
	t_2	1.0	N/A

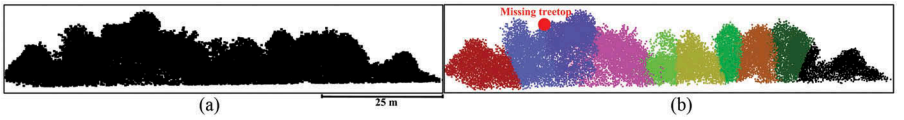

Figure 6. Illustration of the disadvantage in the treetop supervoxel detection. (a) Input tree points. (b) Tree segmentation results and the missing treetop (red dot) caused by the overlapping of trees.

very low computation complexity, i.e. $O(1)$ and $O(\log_2 m)$, respectively, where m is the number of elements in **M**. Points from the same supervoxel are homogeneous, which have a high similarity measured by **S**. The drawback in the supervoxel generation is the high memory request for the defined matrices **S, M** and **N**, which can be improved by sparse storage techniques. Another disadvantage lies in the search of treetop supervoxels. The filtering process uses a fixed window results in the missing of treetops as shown in Figure 6. Therefore, further research is necessary to develop an adaptive sliding window algorithm for the search of treetop supervoxels.

5. Conclusions

In this letter, we propose a general pipeline for the individual tree segmentation from urban environments. The pipeline consists of the preprocessing step to extract tree points, the supervoxel generation step to group homogeneous points and the segmentation step to delineate trees. In the supervoxel generation, we first optimize supervoxel centers, and then assign other points to them for obtaining complete supervoxels. We succeed to overcome two main drawbacks in the commonly used tree point assignment strategy, including the low efficiency caused by assigning the index to each point and the assignment of different tree indexes for homogeneous points. Our algorithm requires inputting point clouds only and is proceeded automatically. Experiments show that the proposed algorithm is competitive with other existing methods and is a promising method for the individual tree segmentation from LiDAR point clouds.

Funding

National Key Research and Development Plan of China (2016YFD0600101), National Natural Science Foundation of China (31770591, 41701510), China Postdoctoral Science Foundation (2016M601823).

References

Cortes, C., and V. Vapnik. 1995. "Support-Vector Networks." *Machine Learning* 20 (3): 273–297. doi:10.1007/BF00994018.

Hamraz, H., M. A. Contreras, and J. Zhang. 2016. "A Robust Approach for Tree Segmentation in Deciduous Forests Using Small-Footprint Airborne Lidar Data." *International Journal of Applied Earth Observation and Geoinformation* 52: 532–541. doi:10.1016/j.jag.2016.07.006.

Li, L., D. Li, H. Zhu, and Y. Li. 2016. "A Dual Growing Method for the Automatic Extraction of Individual Trees from Mobile Laser Scanning Data." *ISPRS Journal of Photogrammetry and Remote Sensing* 120: 37–52. doi:10.1016/j.isprsjprs.2016.07.009.

Li, W., Q. Guo, M. K. Jakubowski, and M. Kelly. 2012. "A New Method for Segmenting Individual Trees from the Lidar Point Cloud." *Photogrammetric Engineering & Remote Sensing* 78 (1): 75–84. doi:10.14358/PERS.78.1.75.

Mongus, D., and B. Žalik. 2015. "An Efficient Approach to 3d Single Tree-Crown Delineation in Lidar Data." *ISPRS Journal of Photogrammetry and Remote Sensing* 108: 219–233. doi:10.1016/j.isprsjprs.2015.08.004.

Tao, S., F. Wu, Q. Guo, Y. Wang, W. Li, B. Xue, X. Hu, et al. 2015. "Segmenting Tree Crowns from Terrestrial and Mobile Lidar Data by Exploring Ecological Theories." *ISPRS Journal of Photogrammetry and Remote Sensing* 110: 66–76. doi:10.1016/j.isprsjprs.2015.10.007.

Vallet, B., M. Brédif, A. Serna, B. Marcotegui, and N. Paparoditis. 2015. "Terramobilita/Iqmulus Urban Point Cloud Analysis Benchmark." *Computers & Graphics* 49: 126–133. doi:10.1016/j.cag.2015.03.004.

Véga, C., A. Hamrouni, S. El Mokhtari, J. Morel, J. Bock, J. P. Renaud, M. Bouvier, and S. Durrieu. 2014. "Ptrees: A Point-Based Approach to Forest Tree Extraction from Lidar Data." *International Journal of Applied Earth Observation and Geoinformation* 33: 98–108. doi:10.1016/j.jag.2014.05.001.

Wu, B., B. Yu, W. Yue, S. Shu, W. Tan, C. Hu, Y. Huang, J. Wu, and H. Liu. 2013. "A Voxel-Based Method for Automated Identification and Morphological Parameters Estimation of Individual Street Trees from Mobile Laser Scanning Data." *Remote Sensing* 5 (2): 584–611. doi:10.3390/rs5020584.

Xia, S., C. Wang, F. Pan, X. Xi, H. Zeng, and H. Liu. 2015. "Detecting Stems in Dense and Homogeneous Forest Using Single-Scan Tls." *Forests* 6 (11): 3923–3945. doi:10.3390/f6113923.

Xu, S., S. Xu, N. Ye, and F. Zhu. 2018. "Individual Stem Detection in Residential Environments with Mls Data." *Remote Sensing Letters* 9 (1): 51–60. doi:10.1080/2150704X.2017.1384588.

Yun, T., F. An, W. Li, Y. Sun, L. Cao, and L. Xue. 2016. "A Novel Approach for Retrieving Tree Leaf Area from Ground-Based Lidar." *Remote Sensing* 8 (12): 942. doi:10.3390/rs8110942.

Zhang, Z., L. Zhang, X. Tong, P. T. Mathiopoulos, B. Guo, X. Huang, Z. Wang, and Y. Wang. 2016. "A Multilevel Point-Cluster-Based Discriminative Feature for Als Point Cloud Classification." *IEEE Transactions on Geoscience and Remote Sensing* 54 (6): 3309–3321. doi:10.1109/TGRS.2016.2514508.

Zhong, L., L. Cheng, H. Xu, Y. Wu, Y. Chen, and M. Li. 2017. "Segmentation of Individual Trees from Tls and Mls Data." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10 (2): 774–787. doi:10.1109/JSTARS.2016.2565519.

Zhu, F., J. Yang, C. Gao, S. Xu, N. Ye, and T. Yin. 2016. "A Weighted One-Class Support Vector Machine." *Neurocomputing* 189: 1–10. doi:10.1016/j.neucom.2015.10.097.