THE PHOTOGRAMMETRIC RECORD

REMOTE SENSING AND PHOTOGRAMMETRY SOCIETY

The Photogrammetric Record 33(163): 315–340 (September 2018)
DOI: 10.1111/phor.12247

SCALABLE INDIVIDUAL TREE DELINEATION IN 3D POINT CLOUDS

JINHU WANG* (jinhu.wang@tudelft.nl)
RODERIK LINDENBERGH (r.c.lindenbergh@tudelft.nl)
MASSIMO MENENTI (m.menenti@tudelft.nl)
Delft University of Technology, Delft, The Netherlands

*Corresponding author

Abstract

Manually monitoring and documenting trees is labour intensive. Lidar provides a possible solution for automatic tree-inventory generation. Existing approaches for segmenting trees from original point cloud data lack scalable and efficient methods that separate individual trees sampled by different laser-scanning systems with sufficient quality under all circumstances. In this study a new algorithm for efficient individual tree delineation from lidar point clouds is presented and validated. The proposed algorithm first resamples the points using cuboid (modified voxel) cells. Consecutively connected cells are accumulated by vertically traversing cell layers. Trees in close proximity are identified, based on a novel cell-adjacency analysis. The scalable performance of this algorithm is validated on airborne, mobile and terrestrial laser-scanning point clouds. Validation against ground truth demonstrates an improvement from 89% to 94% relative to a state-of-the-art method while computation time is similar.

KEYWORDS: cuboid, individual tree delineation, laser scanning, point cloud, 3D clustering, tree, voxel

INTRODUCTION

TREES PLAY AN INDISPENSABLE ROLE in the urban environment and tree management is of great interest for biomass estimation and monitoring environmental changes (Cottone and Ettl, 2001; Zheng et al., 2007; van Deusen, 2010; Moskal and Zheng, 2012). Traditionally, trees are manually measured *in situ*, which is time-consuming, costly and susceptible to subjective errors. Besides, adverse site conditions can make access difficult (Hopkinson et al., 2004).

Lidar has become a well-established surveying technique for the acquisition of geospatial information (Vosselman and Maas, 2010). Combined with automatic point cloud processing techniques, this in principle enables the efficient extraction of geometric tree parameters. In recent years, many studies have investigated the application of lidar implemented in airborne laser scanning (ALS), mobile laser scanning (MLS) and terrestrial laser scanning (TLS), for tree and forest applications. A typical processing workflow based on registered point clouds consists of three steps: (i) separate tree points from non-tree points; (ii) identify individual trees among all tree points; and (iii) estimate parameters describing individual tree geometry.

Nowadays, large-scale urban tree inventories call for flexible and efficient methods to segment tree points from raw point clouds. Many algorithms have been developed for detecting and classifying trees from point clouds captured by different sensors. A conventional method is first to segment non-terrain points and then extract all tree points based on the height distribution of the points (Kraus and Pfeifer, 1998; Axelsson, 2000; Sithole and Vosselman, 2004; McDaniel et al., 2012). This methodology is robust and both digital elevation models (DEMs) and digital surface models (DSMs) can be generated in parallel. Other approaches are region growing (Pauling et al., 2009; Aijazi et al., 2013), feature-based tree classification (Rutzinger et al., 2010; Strom et al., 2010; Yang and Dong, 2013; Lin et al., 2014) and canopy model-fitting (Lahivaara et al., 2014). With the availability of small-footprint full-waveform lidar systems, algorithms such as those in Guo et al. (2011), Vaughn et al. (2012) and Lindberg et al. (2014) have been proposed to exploit the waveform features of the back-scattered wave to classify tree points. Different sensors can be integrated onto the same platform (Fieber et al., 2013). For example, hyperspectral and multispectral images were integrated into the classification and extraction of trees by Puttonen et al. (2011). There are also numerous methods available to extract tree points from MLS and TLS point clouds (Rutzinger et al., 2011; Yang et al., 2012; Zhong et al., 2013; Sirmacek and Lindenbergh, 2015; Wang et al., 2015).

Individual tree delineation, the second step in the processing chain, aims at separating individual trees from the segmented tree points. Since this is the primary focus of this work, existing methods will be discussed in more detail below. Tree delineation has been studied in fields such as computer graphics, forestry and remote sensing for various purposes. Tree parameters have been extracted and models of trees reconstructed from ALS point clouds (Vosselman et al., 2004). Based on TLS point clouds, forest geometry has been reconstructed for canopy radiative transfer models (Bremer et al., 2015). An automated workflow has been presented to extract 3D tree models from MLS point clouds by Rutzinger et al. (2011). An octree-based space-division procedure was introduced to extract tree skeletons by Bucksch and Lindenbergh (2008). Tang et al. (2013) proposed an algorithm to reconstruct the 3D surface of tree canopies from lidar point clouds. The method first obtains a stack of separated slices corresponding to different height levels; then the acquired boundaries are combined to form canopies of individual trees. However, these methods are either: (i) not scalable and are computationally expensive; or (ii) consider point clouds obtained from only one type of sensor. The algorithm proposed in this work is able to deal with different situations and is demonstrated to work in several case studies involving data from ALS, MLS and TLS systems including challenging scenarios, such as separation of trees on steep terrain.

This paper is structured as follows. Firstly, existing methods of tree separation in ALS, MLS and TLS point clouds are discussed. Next, the newly proposed algorithm for individual tree delineation is presented. Then, the results and evaluation of the algorithm are presented. Finally, concluding remarks and recommendations are made.

Related Work and Proposed Innovations

Existing methods used for individual tree delineation are categorised into two classes: (1) point-based approaches; and (2) cuboid- or voxel-based approaches. The first class deals with all tree points while the latter considers cuboid (volumetric) cells containing points. A detailed review of the two approaches is given below.

Point-based Approaches

Many available algorithms have already proved their feasibility to identify individual (single) trees and estimate their parameters from ALS, MLS and TLS point cloud data. Solberg et al. (2006) presented a method that first generated a canopy surface model from an ALS point cloud. Then, based on this surface model, single trees were segmented and characterised. Based on the vertical distribution of the ALS point clouds, shapes of spruce and pine trees were constructed to individualise and discriminate these two types of tree. In addition, tree positions and the height and crown diameter of individual trees were estimated (Persson et al., 2002; Holmgren and Persson, 2004). Hyyppä et al. (2001) proposed building a terrain model and a canopy model and then, from these, generating a 3D tree-height model. Based on this tree-height model, individual trees were extracted and parameters, such as tree height, crown area and stem diameter, were derived. Rahman et al. (2009) identified local maxima in 2D point densities to distinguish tree canopies from surrounding objects; trees were then delineated based on the obtained maxima.

Rutzinger et al. (2010) introduced an alpha-shape approach for point cloud reduction; tree models were generated consisting of a tree crown and a realistic trunk. The alpha-shape concept is a generalisation of the convex hull and is able to describe the shape of point entities, but requires tuning via the alpha parameter (Edelsbrunner et al., 1983). The accuracy of tree detection was 85%. Although this method reduced the volume of the point cloud data and partly preserved the geometry of trees, it introduced an extra step (to determine the alpha shape) which also affects its computational efficiency.

Li et al. (2012) presented a new region-growing method to segment individual trees from ALS point clouds by taking advantage of the open space between trees. The method segmented 94% of trees in the study correctly but was only tested on sparse discrete ALS point clouds; open space between trees is not always available, especially not in dense mixed forests. Vega et al. (2014) proposed an algorithm to segment single trees by analysing local maxima among the *k* nearest-neighbour points of a query point in 3D space. Points were processed from highest to lowest height and assigned to corresponding tree segments. This algorithm was tested on three different forest types, with 82% of trees being successfully detected. Duncanson et al. (2014) presented a method to delineate multi-layered crowns for assessing individual tree structure using a watershed-based canopy height model. The method could identify 70% of dominant trees and was able to determine tree parameters such as tree height, crown radius and crown area. Lu et al. (2014) developed a bottom-up approach to segment individual deciduous trees sampled when in leaf (so-called *leaf-on*, as opposed to *leaf-off* after shedding of leaves). Both lidar intensity and 3D structure were used in the segmentation. The approach was tested on a forest and correctly detected 84% of trees and correctly segmented 97% of them.

Raumonen et al. (2015) presented a massive-scale tree-modelling method for point clouds from TLS. This method takes tree points as input. Consecutive voxel partitioning is used to structure points and to modify regions of different point density. Next, tree stems are located by horizontally slicing the tree points. Finally, the remaining points are classified as either trees or non-trees. This method was tested on two plots of trees, consisting of English oak and Australian eucalyptus. The biomass overestimation rates were 17% and 8·5% for the two plots, respectively. The presented method is also able to determine tree diameters at breast height; in this case, the classification step is performed point by point.

Cuboid and Voxel Approaches

One way to speed up point cloud processing is to consider voxels rather than individual points. Wang et al. (2008) presented a voxel-based procedure to analyse vertical canopy structure and to obtain 3D models of single trees sampled by ALS. The algorithm first resampled the input point cloud to voxels and a series of horizontal 2D projected images at different height levels were generated in voxel space. Then the main tree-canopy layer and the height ranges of the layers were detected according to a statistical analysis of the height distribution of the normalised raw point clouds. Compared to point-based methods, this approach improved on efficiency, but did not consider the relationship between voxel cells. Bienert et al. (2010) introduced a voxel-based method to analyse wind-field models of a TLS forest scenario. The stems of trees were automatically detected before the 3D point cloud was translated into a voxel structure representing the forest. Then voxels were clustered using region growing and finally individual trees were interpreted.

Several algorithms that process point clouds using voxels have, however, not considered trees. Papon et al. (2013) presented a region-growing algorithm for point cloud segmentation. In this study, first a point cloud was segmented using the relationship between voxels. Next, the segmented results were merged to ensure consistency with the spatial geometry of the scene. However, the algorithm was only tested on indoor objects scanned by a TLS. Aijazi et al. (2013) presented a method to classify 3D urban MLS point clouds based on voxels, but again trees were not studied.

Other studies have included trees in straightforward situations. Cabo et al. (2014) detected pole-like objects from MLS point clouds using cuboid-based methods. The technique first simplified the imported huge high-density MLS point cloud by a regular voxelisation. Then, by assessing the local morphology of voxel cells, trees could be classified and individualised. Nevertheless, the considered street trees were mainly of the same size and did not overlap much. Babahajiani et al. (2015) presented an automated method to classify urban environments based on supervoxels. In this study, buildings, roads, trees and cars were successfully classified. However, delineation of overlapping trees was not studied. Wu et al. (2013) presented a voxel-based method to detect street trees from MLS point clouds. This method was able to extract single trees in a street scenario. However, this method focused on MLS scanned point cloud data; separating trees in ALS and TLS point clouds was not yet considered. This method searched for trees starting from the bottom layer of a 3D grid; therefore, it was unable to detect multiple-stemmed trees or trees that had their trunk occluded. Notably, this method shows its feasibility in separating street trees of comparable size which are located along a road and connected mainly in the road direction. However, urban trees which vary in size or were overlapping in different directions were not yet evaluated. This method employed an incremental competing region-growing algorithm proposed in Liu et al. (2006) to separate touching tree crowns. The strategy took the relative distance to tree centres into consideration rather than the actual connections between voxel cells. This introduced separation errors in cases where a larger tree touched a smaller one, as outer points of the bigger tree would be wrongly assigned to the smaller tree.

Proposed Innovations

The proposed cuboid-based individual tree delineation algorithm in this paper is different from the previously mentioned methods with respect to the following points:

(1) Assignment of cuboid cells to individual trees is based on a novel adjacency analysis, which improves the separation of trees of different size.
(2) In the resampling step, the size of the cuboid cell is different in the three Cartesian coordinate axes directions. Therefore, the term *cuboids* is used rather than *voxels*. The use of cuboids (rather than voxel cubes) makes the algorithm more flexible.
(3) A 3D clustering step groups non-empty cuboids and immediately separates non-connected components. This allows the algorithm to work efficiently.
(4) Individual tree delineation is performed both in a bottom-to-top and top-to-bottom scheme, which makes the algorithm more robust.

Methodology

The methodology presented in this paper consists of five steps:

(1) A preprocessing step classifies tree points from the raw point cloud. The tree points will be imported as input for the next steps.
(2) Resampling and clustering, which resamples the imported tree points to cuboid cells and clusters incident cuboid cells in 3D space.
(3) Seed selection over all the clustered cells.
(4) Individualise (separate) trees.
(5) Evaluate the overall individual tree delineation quality with respect to manually individualised trees as ground truth data.

Step 1: Preprocessing

In the first step, tree points are classified and segmented from the original raw imported point cloud. Since this is not the focus of this work, this step is achieved with existing methods. In this work, tree points are obtained in two steps. First, the imported raw point cloud is classified into ground and non-ground points by implementing the algorithm presented by Kraus and Pfeifer (1998). Next, tree points are extracted from the non-ground points using the methodology in Sirmacek and Lindenbergh (2015). This algorithm first generates 2D horizontal grids and then projects points onto each of the 2D grid cells. The point density of each grid cell is then obtained. Consecutively, by determining the local maxima of the 2D grid cells, the location of a tree is determined and the tree diameter is estimated. Finally, tree points are extracted based on their proximity to these tree locations.

Step 2: Resampling and Clustering

Resampling a point cloud in this work means gathering its points into cuboidal cells. The cell designed for the proposed algorithm is a cuboid rather than the standard cube voxel (where the three sides are of equal length). Cuboid cells have different edge lengths W_x, W_y and W_z in the three coordinate directions (Fig. 1) to improve flexibility. The non-empty cells in green are defined as *positive cells*, in contrast to empty *zero cells*.

To save computation time and deal with trees that have large height differences, connected cells are clustered after point cloud resampling. In this study, a 3D seed filling algorithm from Yu et al. (2010) is employed to perform the clustering on all positive cells before individual tree delineation. First, all positive cells are labelled as non-visited. Then,

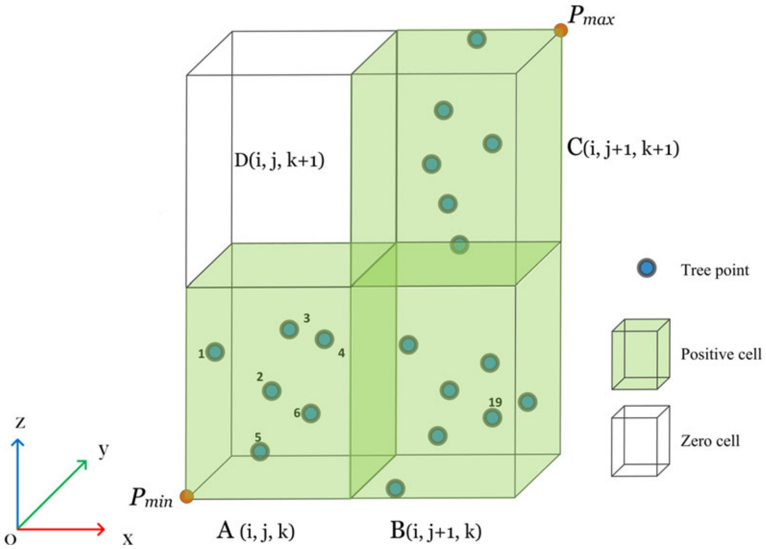

FIG. 1. Point cloud resampling. Points are assigned to a designated 3D cuboid cell according to its bin indices.

for the current cell, its 26 3D neighbouring cells are obtained. All positive neighbours, that are non-visited, are pushed into a cluster stack. This step is performed recursively until all positive cells are traversed. The output of this step is connected cells, which are represented as clusters. In this step, all the cells are traversed only once. Thus, the computational complexity of this step is linear in the number of cells, which is $O(N)$ in asymptotic notation (Cormen et al., 2001), if there are N cells.

Step 3: Select Seed Cells

The separation of the clustered tree cells into individual trees starts with seed cell identification. A seed will potentially result in one individual tree after separation. The procedure of separation from the bottom layer upwards is similar to the separation from the top layer downwards. The only difference lies in seed inheritance. In the downwards direction, the label of a cuboid is transferred to its neighbouring bottom-face cuboid. When traversing in the upwards direction, the label is transferred to its upper-face neighbour. This section will only describe the methodology in detail for the top layer downwards. First, a cell is defined as a top cell if and only if this cell has a bottom-face neighbour but has no top-face neighbouring cell. The bottom-face neighbour to a cell is the cell that connects to the bottom face of the query cell. For example, in Fig. 1 cell B is the bottom-face neighbour of cell C; conversely, cell C is the top-face neighbour of cell B. Fig. 2 depicts a scenario with two clusters: Cluster 1 contains two overlapping trees and Cluster 2 one individual tree. The coloured cells are the initially identified seed cells of the two clusters. Next, connected seed cells are clustered, resulting in the seeds S_1, S_2, S_3 and S_4 in Fig. 2. Note that, at this stage, one seed typically consists of a number of connected cells. For example, seed S_1 consists of five connected top cells. The location of a seed is defined as the centroid of all the points inside the cell.

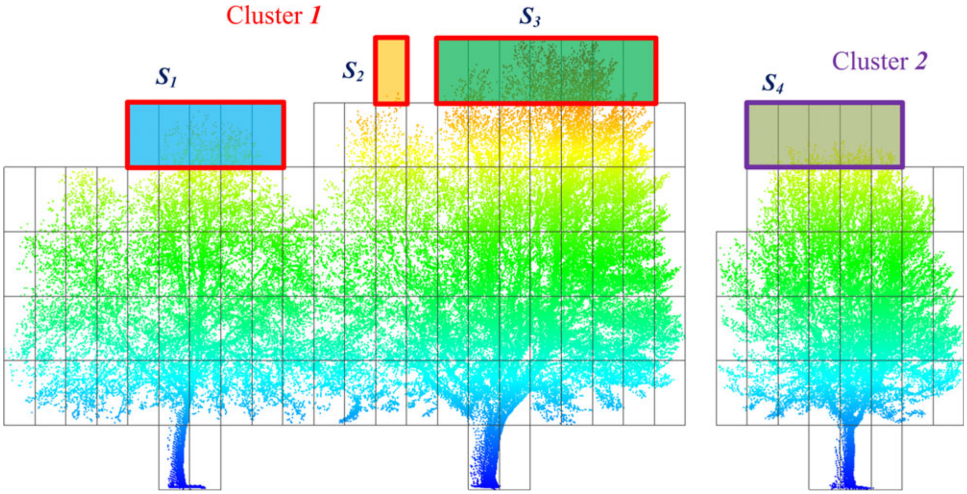

FIG. 2. Connected top cells are clustered as potential seeds of individual trees from the top layer downwards. S_1, S_2 and S_3 are potential seeds of Cluster 1, and S_4 is a potential seed of Cluster 2.

Step 4: Tree Separation

The tree-separation procedure starts with the potential seeds identified in the previous section. The strategy described in this section is to separate trees starting from the potential seeds in the top layer of the cluster downwards to the bottom layer. The following paragraphs describe the method in detail.

Step 4.1: Detect Multiple Component Clusters. This section categorises the 3D cuboid clusters into either multiple or single component clusters. A multiple component cluster is expected to contain more than one tree. A cluster is considered as having multiple components if and only if this cluster meets the following conditions:

(1) The cluster has at least two potential seeds.
(2) The minimum distance between any two potential seeds is larger than the pre-set minimum tree-canopy diameter.

If a cluster does not meet these conditions, it will be considered as an individual tree. A multiple component cluster will be forwarded to the tree-separation step.

Step 4.2: Seed Inheritance. Merging close by seeds avoids separating one individual tree with several high branches erroneously into two or more trees. After potential seed cells are identified, the separation of multiple component clusters starts with merging seeds separated by only a small distance. Fig. 3 shows a side view of the scenario in Fig. 2, which has two overlapping trees, illustrating that the resampling resulted in seven vertical layers. The identified potential seed cells are firstly clustered and labelled as S_1, S_2 and S_3. Points P_1, P_2 and P_3 are the centroids of the points in each of the respective seed clusters.

The horizontal distances between the centroids of all the potential seeds are computed. As shown in Fig. 3, the horizontal distances between the three selected seed cells are D_{12},

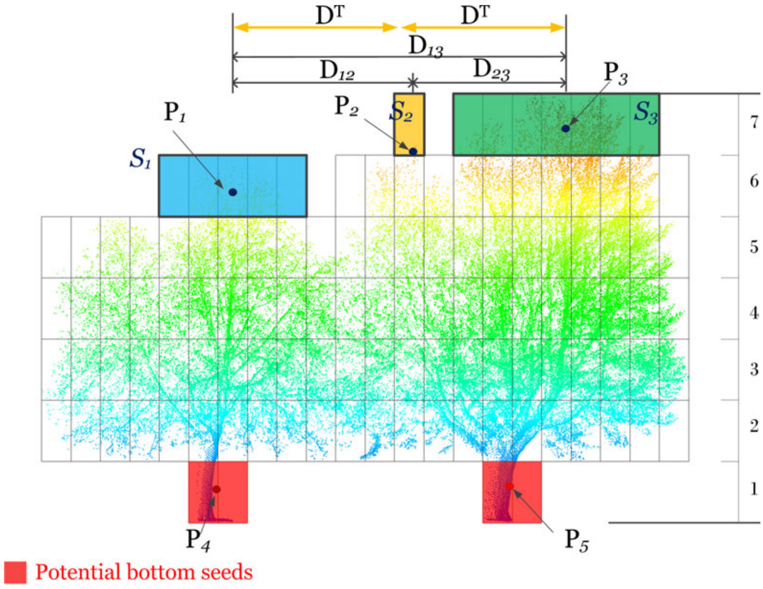

FIG. 3. Seed cell identification and merging. D_{13}, D_{12} and D_{23} are the distances between the three selected seeds, respectively. D^{T} is the distance threshold of the two trees. The distance D_{23} is smaller than the pre-set minimum tree canopy diameter D^{T}, thus potential seeds S_2 and S_3 are merged together as one larger seed.

D_{23} and D_{13}. Seeds in close proximity are merged in an iterative way as follows. First, distances between seeds are sorted in ascending order and, for the pair of closest seeds, the horizontal distance between them is tested to see if it is smaller than the pre-set minimum tree-canopy diameter. In Fig. 3, horizontal distance D_{23} is smaller than the pre-set minimum tree-canopy diameter D^{T}, thus seeds S_2 and S_3 are merged into one seed and labelled as S_2, as illustrated in Fig. 4. Next, distances between seed pairs are recomputed and the distance evaluation is repeated until no distance between seed cells is smaller than the tree-canopy diameter threshold.

After merging seeds in close proximity, their bottom-face neighbours inherit the cluster index from their top-face neighbours. As illustrated in Fig. 4, the bottom-face neighbours of seeds S_1 and S_2 in layers 6 and 7 are identified, which are coloured in cyan and orange, respectively. Those cells inherit the cluster index and form the new seeds of layers 5 and 6. The same operation is conducted when traversing to the next layer downwards.

Similar operations are performed when traversing from the bottom layer upwards. As shown in Fig. 3, the red cells are seed cells, and the same distance evaluation is conducted on the centroids denoted by points P_4 and P_5. Then similar seed-merging and index-inheritance operations are performed until the procedure has traversed to the top layer of the cuboid cells.

Step 4.3: Assign Cells to Individual Trees. The objective of this step is to assign non-seed cells to individual trees. After seed identification and merging, seed indices are inherited via bottom-face neighbouring cells and separation continues at the next layer of cuboid cells. If, at a given layer, cells are connected to only one seed cell, then these cells

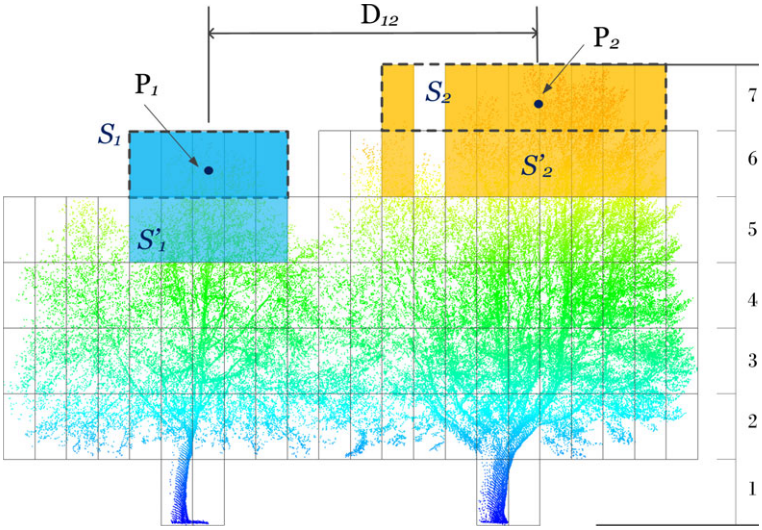

FIG. 4. Inherited labelling of trees from the top layer. Tree labels of the lower-layer cells S'_1 and S'_2 are inherited from their top-face neighbouring seeds S_1 and S_2, respectively.

are assigned to the corresponding tree. The situation becomes more complicated when traversing to a layer where all cells are connected and have at least two seed indices inherited from the top layer. As shown in Fig. 4, layer 5 will obtain two different seed indices from layer 6. To separate cells for such layers, a cell-assignment strategy based on adjacency analysis is presented in this study.

Rather than simply considering distances to different seed cells, connectivity-based separation takes the neighbourhood of the cells into account. Fig. 5 is a plan view of layer 5 in Fig. 4. The cells in cyan and orange have indices inherited from the top-face neighbouring seeds S_1 and S_2, respectively, as depicted in Fig. 4. The striped orange and cyan cells are boundary cells of the two trees in layer 5 and are identified first. Boundary seed cells are defined as seed cells that have at least one unassigned cell among its eight 2D neighbours. Unassigned cells will be assigned to a tree according to a *connectivity coefficient*. First, the connectivity coefficient is set to 1 for all boundary cells. Then the coefficient is propagated to its unassigned neighbouring cells. The connectivity coefficients are computed using the following equation:

$$c(t,s) = C(t,s) \times R(k) \tag{1}$$

where *R(k)* is the attenuation influence (see below); *c(t, s)* is the connectivity coefficient of a target cell t with regard to a source cell s; and *C(t, s)* is the neighbour type of the target cell with respect to the source cell, as determined as follows:

$$C(t,s) = \begin{cases} 0 \cdot 50, & t \text{ is a face neighbour of } s \\ 0 \cdot 25, & t \text{ is an edge neighbour of } s. \end{cases} \tag{2}$$

A cell t is defined as a face neighbour of source cell s if either t shares a face with s, or when all cells on the straight line connecting t with s are face neighbours of s. If a cell is

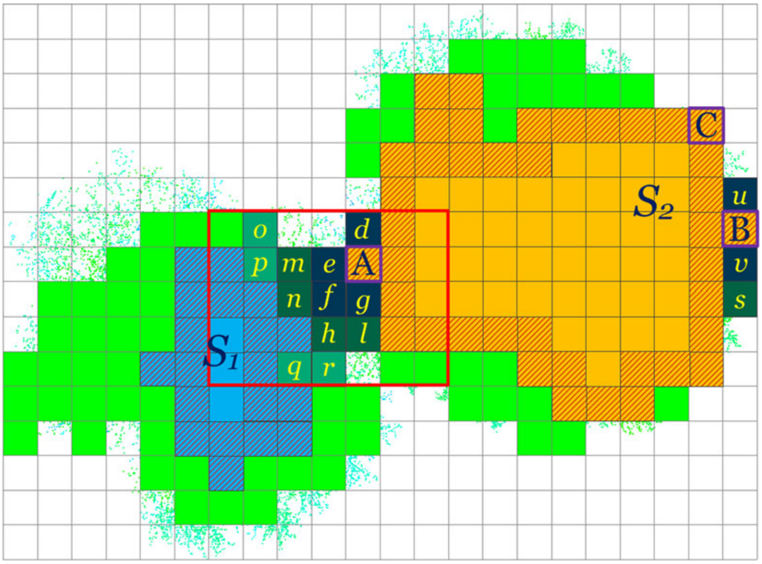

Fig. 5. Connectivity-based cell separation. Boundary cells of seeds S_1 and S_2 are obtained to compute the adjacent coefficient for all unassigned cells in the same layer, such as *d*, *e*, *f* and *g*. The striped orange and cyan cells are boundary cells of seeds S_2 and S_1, respectively. The red rectangle is used in Fig. 6.

not a face neighbour of a source cell, it is an edge neighbour of a source cell. For example, in Fig. 5 cells *d*, *e* and *g* are face-connected neighbouring cells, while *f* is an edge-connected neighbouring cell of boundary cell *A*.

In equation (1), *R(k)* denotes the attenuation of the influence from the source cell to its unassigned neighbouring cells: $R(k) = 1/k$ where *k* is the order of the cell with regard to the source cell, defined as the length (in number of cuboids) of the shortest path connecting the source and the target. For example, in Fig. 5, unassigned cells *d*, *e*, *f* and *g* are directly connected to cell *A*, so they are first-order cells of *A* and hence $k = 1$ Similarly, cells *m*, *n*, *h* and *l* are second-order cells so $k = 2$.

All the boundary cells propagate their connectivity coefficients to the connected unassigned neighbouring cells of different orders until there is no connected unassigned cell in the same layer. For example, in Fig. 5, cell *B* has cells *u* and *v* as its first-order neighbours and cell *s* as its second-order neighbour. After propagating its connectivity coefficient first to cells *u* and *v*, and then (consecutively) to cell *s*, cell *B* has finished its connectivity propagation procedure. Cell *C* has no unassigned connected cells, so no connectivity coefficient with respect to *C* needs to be computed. All seeds in one layer propagate their connectivity through their boundary cells to the unassigned cells in the same layer. The unassigned cells are then assigned to the tree that has the largest accumulated connectivity value. After all cells are assigned to an individual tree segment, the points inside these cells are exported as individual tree points.

Fig. 6 is an enlarged display of the red rectangle in Fig. 5, which shows the accumulated connectivity coefficients of each cell with respect to the boundary cells of the two trees S_1 and S_2. In Fig. 6, the value in the upper-right corner of each cell is the accumulated connectivity coefficient value of tree S_2 while the value in the lower-left corner corresponds to that of tree S_1. The accumulated connectivity coefficient value of an

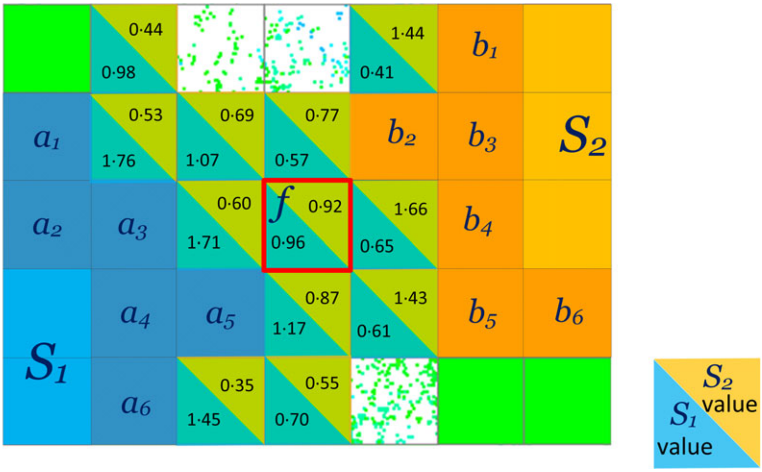

FIG. 6. Computed adjacent coefficients of the unassigned cells. An unassigned cell will be assigned to the tree that has the biggest adjacency coefficient.

unassigned cell to a tree in the same layer is computed as follows (note that in Fig. 6, those values are only computed based on the boundary cells in the rectangle for demonstration):

$$c\left(t, S_j\right) = \sum\nolimits_{i=1}^{n} c_i = \sum\nolimits_{i=1}^{n} C\left(t, s_i\right) \times R_i. \tag{3}$$

Here *c(t, S*$_j$*)* is the accumulated connectivity coefficient of an unassigned cell propagated from all boundary cells of tree S_j; n is the number of the boundary cells of the tree; c_i is the connectivity coefficient value of the ith boundary cell, *c(t, S*$_i$*);* and R$_i$ are the neighbour type and the inverse order of the unassigned cell with respect to the ith boundary cell s_i For example, the connectivity value of cell f in the red rectangle in Fig. 5 with respect to tree S_1 is computed as follows:

$$\begin{aligned} c_1 &= c_{a1} + c_{a2} + c_{a3} + c_{a4} + c_{a5} + c_{a6} \\ &= \frac{0{\cdot}25}{3} + \frac{0{\cdot}5}{3} + \frac{0{\cdot}5}{2} + \frac{0{\cdot}25}{2} + \frac{0{\cdot}25}{1} + \frac{0{\cdot}25}{3} = 0{\cdot}96. \end{aligned} \tag{4}$$

Here c_{a1} to c_{a6} are the connectivity coefficients for boundary cells a_1 to a_6, respectively. Take c_{a1} for example: cell f is an edge-connected, third-order neighbour of boundary cell a_1. The influence from cell a_1 is determined as 1/3. Therefore, the connectivity coefficient of cell a_1 with regard to f will be 0·25/3 according equation (1). Similarly, the connectivity values to tree S_2 are computed as follows:

$$\begin{aligned} c_2 &= c_{b1} + c_{b2} + c_{b3} + c_{b4} + c_{b5} + c_{b6} \\ &= \frac{0{\cdot}25}{3} + \frac{0{\cdot}5}{1} + \frac{0{\cdot}25}{2} + \frac{0{\cdot}5}{2} + \frac{0{\cdot}25}{2} + \frac{0{\cdot}25}{3} = 0{\cdot}92. \end{aligned} \tag{5}$$

Since $c_1 > c_2$ for cell f, it is assigned to tree S_1. The resulting cell assignments are illustrated in Fig. 7. It can be seen that cells o, p, m, n, h, q, r and f in light blue are assigned to tree S_1, and cells d, e, g and l in light orange are assigned to tree S_2.

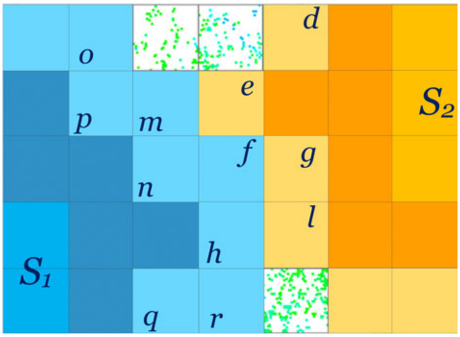

FIG. 7. Previously unassigned cells are assigned to a particular tree as their colour indicates.

Step 5: Overall Quality Analysis

This section describes the strategy and parameters to evaluate the individual tree delineation quality in this study. During the tree-separation process, whether from the top layer downwards or from the bottom layer upwards, each individualised tree has a quality flag assigned to it according to the size of the tree. If the individualised tree-canopy diameter is smaller than a pre-set threshold, the flag is set to 0; if it is larger than the threshold, the flag is set to 1. In this work, the threshold was set to five times the minimum tree diameter. For those trees that were flagged as 0, another separation is conducted in the opposite direction. A valid separation result (with a flag value of 1), is chosen as the final tree-separation result. If both traversed directions result in a quality flag of 0, this result is given as the output. Such negative results could be an indication for a human operator that further inspection is needed.

Expected Computational Effort

The computational effort of the proposed method depends on the size of the cuboid cell and the complexity of the algorithm. Suppose the edge lengths of the 3D bounding box of tree points is D_x, D_y and D_z, respectively. If the cuboid cell size used is d_x, d_y and d_z, then the number of cells is $N = D_xD_yD_z/(d_xd_yd_z)$. Since the cells will be traversed at least once, a lower bound for the computational efforts can be set as $\Omega(N)$. Also, the complexity of clustering connected cells can be assigned as $O(N)$. However, it is difficult to find an accurate index of tree-separation complexity, since the connectivity coefficient determination depends on the number of boundary cells, which varies in different scenarios. For example, trees with different canopy sizes, or trees having a circular shape in horizontal cross section, will have fewer boundary cells than a complex tree.

RESULTS

To verify the flexibility and reliability of the proposed algorithm, five tests were performed:

(1) Separating the same trees sampled by different sensor systems.
(2) Using different cuboid sizes to separate the same trees sampled by the same sensor.
(3) Separating trees with occluded trunks.

(4) Separating trees on steep terrain.
(5) Validated and comparison against manually processed ground-truth data and an existing method (Wu et al., 2013) using a patch of 11 trees scanned by TLS.

The presented algorithm was implemented in C++ and tested on a Dell desktop with four 3·6 GHz processors and 16·0 GB memory. The results of these tests are discussed below.

Same Trees Sampled by Different Sensors

In this section, the flexibility of the presented method is evaluated on a group of three trees, scanned by both MLS and TLS. The trees are all of the species *Aesculus hippocastanum* "Baumannii" (Delft City Trees, 2017). The datasets sample two almost overlapping trees and one isolated tree; details are given in Table I. The data was acquired by different companies hired by the Dutch government; the name of the ALS system is unknown, despite enquiries.

Fig. 8 shows three different point clouds sampling the same trees. Because of differences in scanning mechanism and acquisition geometry, the point density of the three datasets varies considerably (Table I). Compared to MLS and TLS (Figs. 8(b) and (c), respectively), the ALS point cloud (Fig. 8(a)) consists of far fewer points. Note that the two left-hand trees in green and blue) are almost touching (overlapping) whereas the right-hand one (in red) is isolated.

The delineation results of the ALS, MLS and TLS point clouds are illustrated in Fig. 8. In this test, the selected cuboid cell size was 1·0 m in the *x* and *y* directions and 2·5 m in the *z* direction. The minimum tree-canopy diameter was 7·5 m and the maximum tree bounding box size was set to five times of the minimum canopy diameter. The same cuboid size was used for all three point clouds. The processing time for ALS, MLS and TLS datasets was 0·241, 0·428 and 0·626 s, respectively. Note that although the number of points in the TLS point cloud is 644 times larger than that of the ALS dataset, the processing time is only three times longer. As can be distinguished from the enlarged views on the left of Fig. 8, the two overlapping trees are well separated in all three data sets. However, the point clouds of these deciduous trees were collected when they were in leaf (leaf-on). As a result, several branches were severely occluded and not all branches could be completely scanned, let alone separated using smaller-size cuboids.

Separating Trees after Leaf Shedding (Leaf-off) at the Branches Level

In this paragraph, tree individualisation is demonstrated on two nearly overlapping deciduous trees scanned after abscission (leaf shedding) and thus leaf-off. A smaller sized cuboid was used to test the capability of the proposed method to separate trees at the

TABLE I. Details of the three test datasets.

	ALS	*MLS*	*TLS*
Scanner	Unknown(van der Sande et al., 2010)	Fugro DriveMap	Leica C10
No. of tree points	3640	112 957	2 346 740
Density (points/m^3)	25·3	241·9	1062·5
Scanning date	30th November 2011	23rd November 2013	23rd July 2015

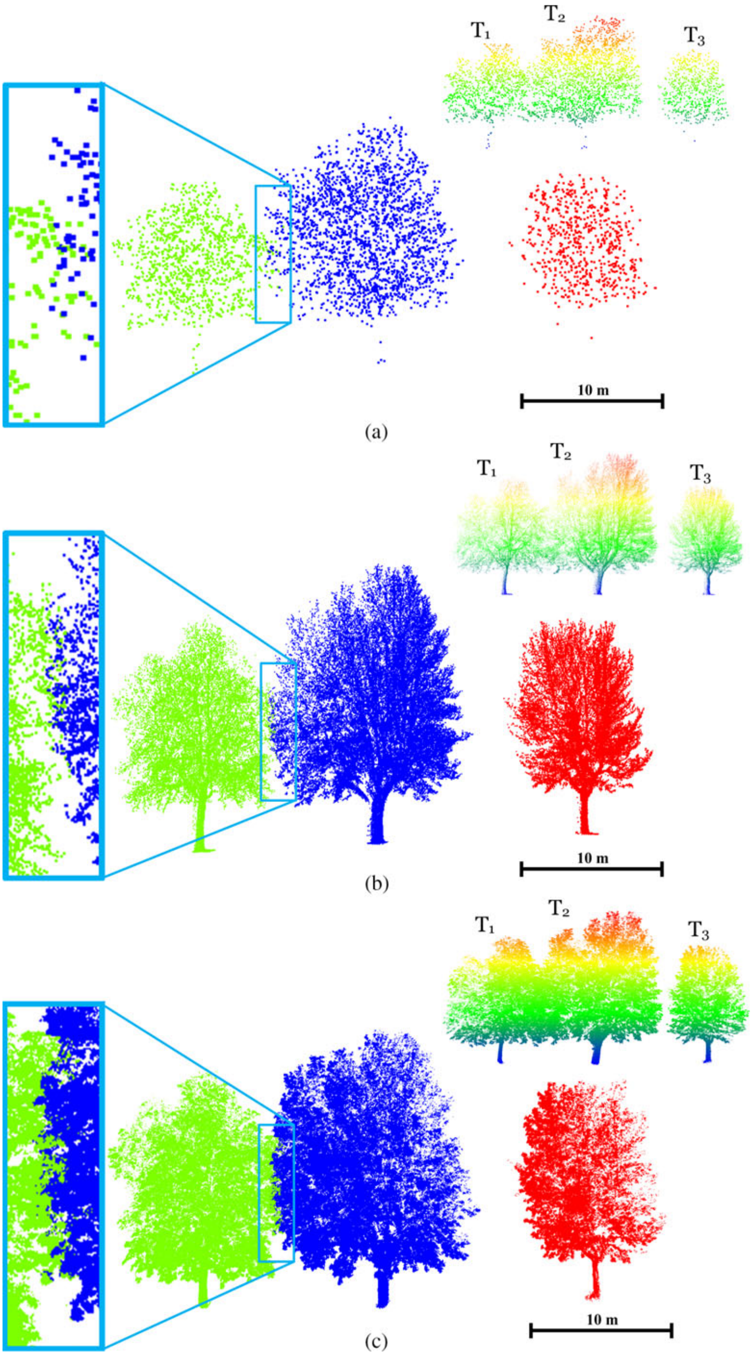

FIG. 8. Original point clouds sampled by three different sensors and delineation results of the overlapping trees: (a) ALS; (b) MLS; and (c) TLS. The enlargements show the overlap between the two left-hand trees.

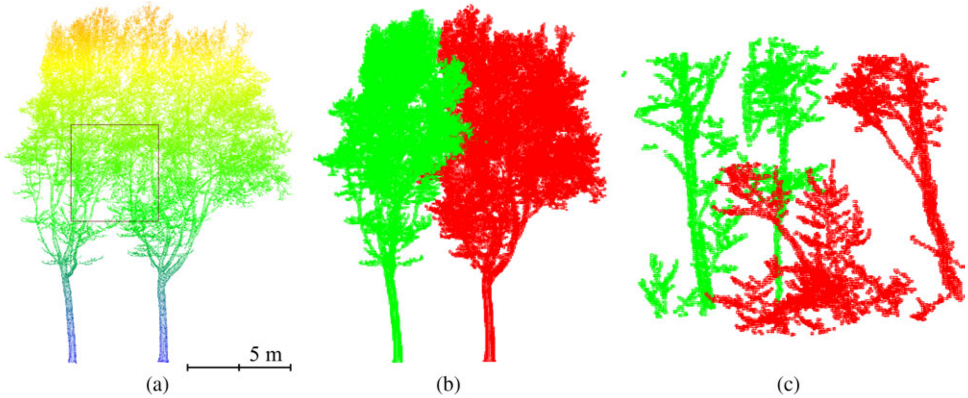

FIG. 9. Individual tree delineation, using a small cuboid size, of two overlapping trees scanned by TLS. (a) Original point cloud of the two trees. The points are coloured by height from red to blue. (b) Individualisation results of the proposed method. The cuboids in green and in red belong to the two different trees. (c) Close-up view of some branches and twigs.

branches level. Two overlapping trees were scanned by a Leica C10 TLS scanner. Fig. 9(a) shows the original point cloud consisting of 270 909 points. To demonstrate the proposed method, the size of the cuboid used for re-sampling was set to 10 cm in all three directions. The minimum tree diameter was set to 4·5 m. The two trees were individualised in the bottom-to-top direction. Fig. 9(b) shows the separation results at cuboid level. Fig. 9(c) is the area in the rectangle in Fig. 9(a), which clearly illustrates that the branches of the two trees are well separated. Note that this conclusion is based on visual inspection in a 3D point cloud viewer. It revealed that the green and red branches and twigs in Figs. 9(b) and (c) indeed belonged to different trees.

Small cuboids can also be applied to extract structure within an individual tree. This is demonstrated on a single tree, scanned by a Leica C10 laser scanner. To separate individual branches, a cube with 10 cm sides was used in combination with a minimum tree diameter threshold of 1·5 m. The separation was conducted in the top-to-bottom direction. The original point cloud consisted of 107 062 points and the height of the tree was approximately 26·2 m (Fig. 10(a)). Fig. 10(b) provides the separation results: the resulting five components are the main branches of the tree.

Trees on Steep Terrain

This section demonstrates the capability of the proposed algorithm to individualise trees on steep terrain. Fig. 11(a) shows steep terrain at Obergurgl, Austria. There are several spruce trees on the steep slope and some of them are touching or overlapping. This area was scanned on 7th July 2015 with a Riegl VZ-400 scanner. The distance of the TLS to the cliff is 95 m on average and the scanning resolution was set to "high". The area within the red rectangle was selected for testing. The original point cloud of this area has 121 039 points with 30 732 tree points remaining after segmentation.

After segmenting the tree points from the original point cloud, the proposed algorithm was applied. The edge length of the cuboid cells was set to 10 cm and the minimum tree diameter was set to 1·5 m. Fig. 11(c) shows the resulting trees while Fig. 11(d) provides the

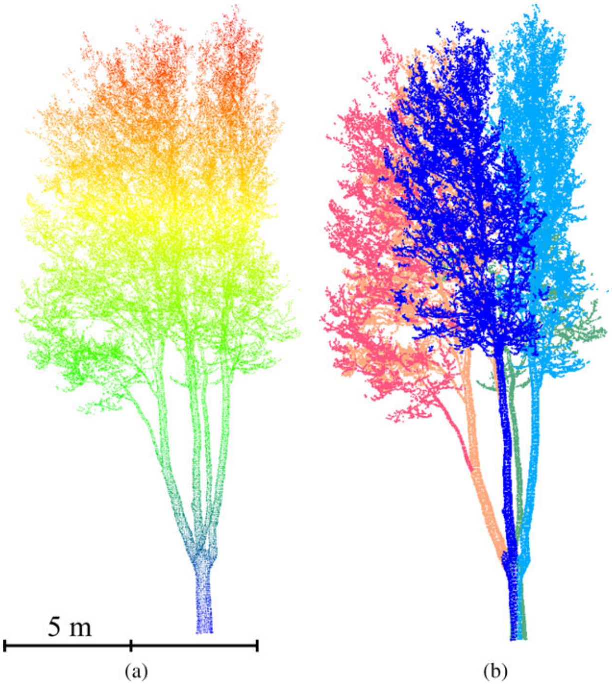

Fig. 10. Separation of branches. (a) Original TLS point cloud: the points are coloured from red to blue based on height. (b) Separation results using 10 cm cuboids in combination with a small (1·5 m) minimum tree diameter threshold in the top-to-bottom direction; different colours represent different individual branches.

results from the method of Wu et al. (2013). Ground points are dark red and individualised trees (36 in total) are depicted in random colours. As can be observed from the enlarged area in the black rectangle, the touching or overlapping trees are correctly individualised in Fig. 11(c), while Fig. 11(d) shows that the existing method (Wu et al., 2013) did not separate the overlapping trees using a voxel size of 10 cm.

Validation Against Ground Truth Data

To quantitatively evaluate the accuracy of the proposed method for individual tree delineation, a group of 11 trees in dense forest with very small gaps between the trees was scanned by a Leica C10 TLS scanner. The points of those trees were manually separated using the open-source CloudCompare software and the results were used as ground truth for accuracy evaluation. The original point cloud and the segmented ground truth are shown in Fig. 12 in both plan (top) and side views. Note that each of the 11 trees is numbered in both views of the original TLS point cloud. In this test, the method proposed by Wu et al. (2013) was implemented for comparison. For both methods (Wu's and the presented one), the same cuboid cell size of 30 cm was used and 1·5 m was set as minimum tree diameter.

The individual tree delineation results of the two methods are shown in Fig. 13, where (a) is a top view of the separation results from Wu's method and (b) is the result of the proposed method. The method in Wu et al. (2013) correctly assigned most of the points to a

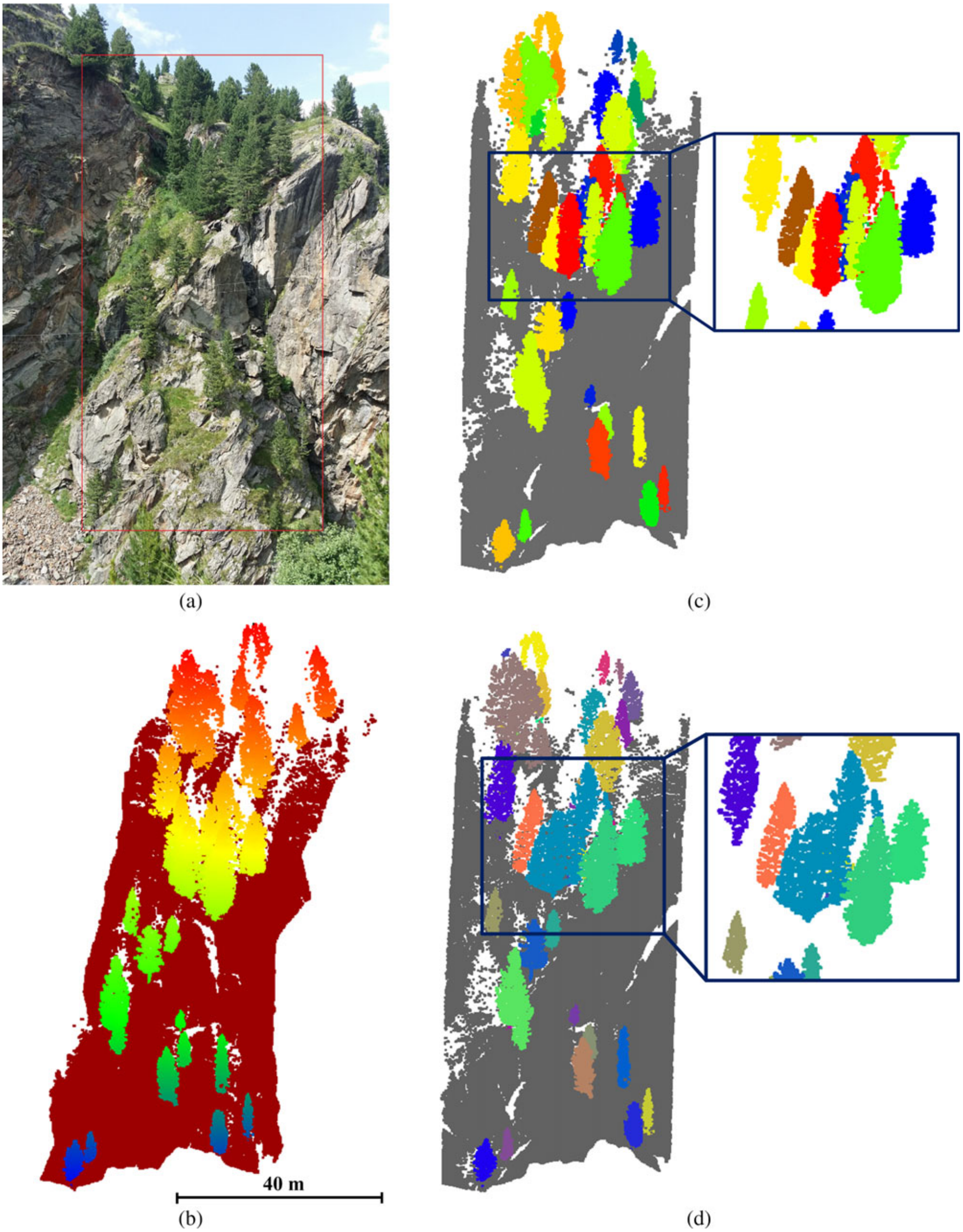

FIG. 11. Individual tree delineation on steep terrain at Obergurgl, Austria from a single TLS point cloud. (a) Test area. (b) Original segmented tree points and ground points. Terrain points are in dark red and the tree points are coloured from red to blue according to height. (c) Individual tree delineation results of trees on this steep terrain. (d) Tree separation results obtained by the method of Wu et al. (2013).

Table II. Confusion matrix of the tree separation results of the proposed method (%).

Individual tree delineation		*Trees (manually separated ground truth)*											*Row total (no. of points)*
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	
Trees	1	98·4	15·0	11·0	0·2								98 687
(separation results	2		85·0	1·2		0·2							44 151
of the proposed	3			87·6	3·5								54 872
method)	4	1·6		0·2	95·6			0·9	1·5				62 077
	5					92·3	0·5						39 263
	6					0·2	93·7	2·0					51 074
	7				0·7		0·5	89·4				0·2	55 168
	8							2·0	97·3				61 250
	9					7·3	0·2			98·1			89 989
	10						5·1	3·1		1·9	99·0	0·4	18 479
	11							2·6	1·2		1·0	99·4	123 012
Column total (no. of points)		85 653	50 969	60 161	61 811	42 220	53 067	60 658	61 716	88 524	11 854	121 389	698 022

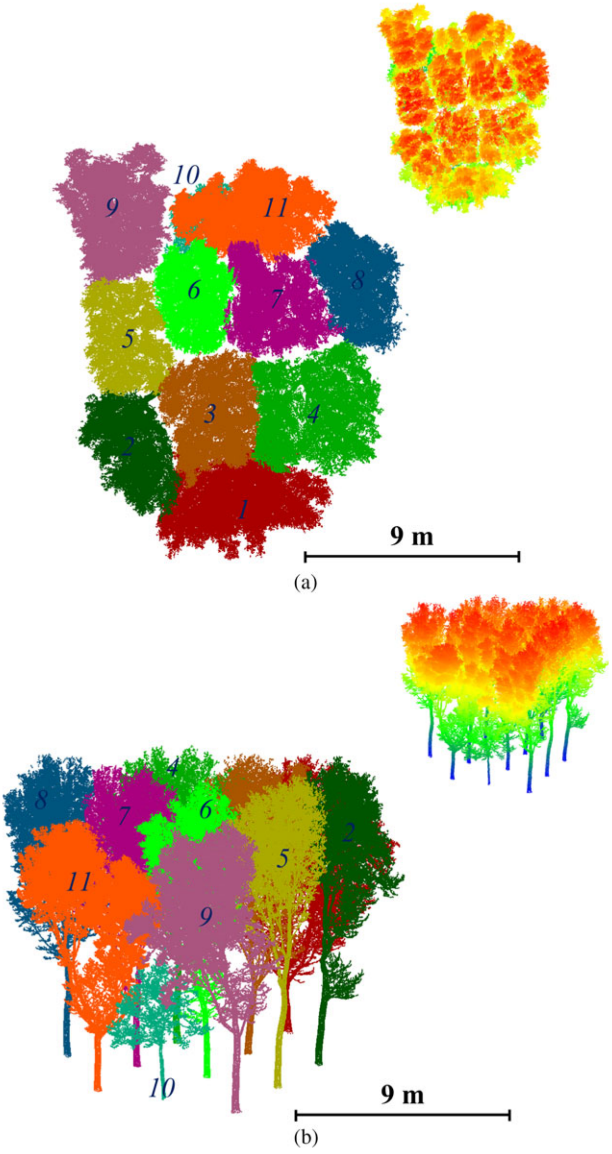

Fig. 12. Original TLS point cloud (bottom left) and manually separated ground truth (top right) of a group of 11 overlapping trees: (a) top view; (b) side view.

particular tree; however, the proposed method is superior at separating overlapping parts of trees and therefore generates better results. To quantify the delineation accuracy of the two methods, the manually separated ground truth of the trees was used as reference. The separation of the two methods was computed using Cohen's kappa coefficient (Cohen, 1960), based on the number of points that were correctly assigned. For this purpose, each of

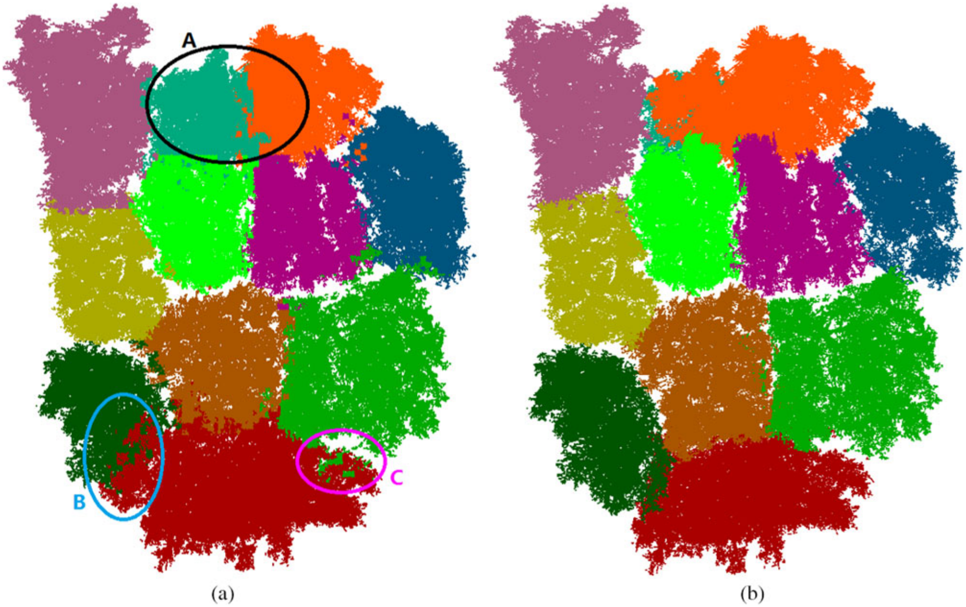

FIG. 13. Top view of the individual tree delineation results of the two methods applied on TLS point cloud: (a) Wu et al. (2013) method; (b) proposed method. The areas indicated by A, B and C are cases where Wu's method resulted in larger errors.

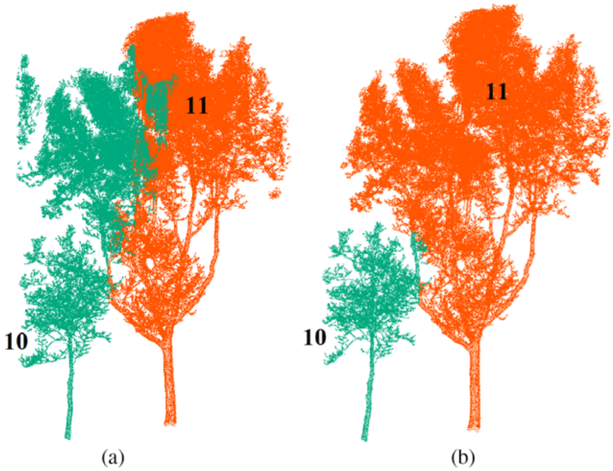

FIG. 14. Tree separation details of Wu's existing method (a) and the proposed method (b) from TLS point clouds on tree 10 and tree 11 in Fig. 10.

the 11 trees was considered as a classification class. Using the reference data, an 11 × 11 confusion matrix was determined as shown in Table II. From this confusion matrix, Cohen's kappa coefficient was determined for both Wu's method and the proposed method. The

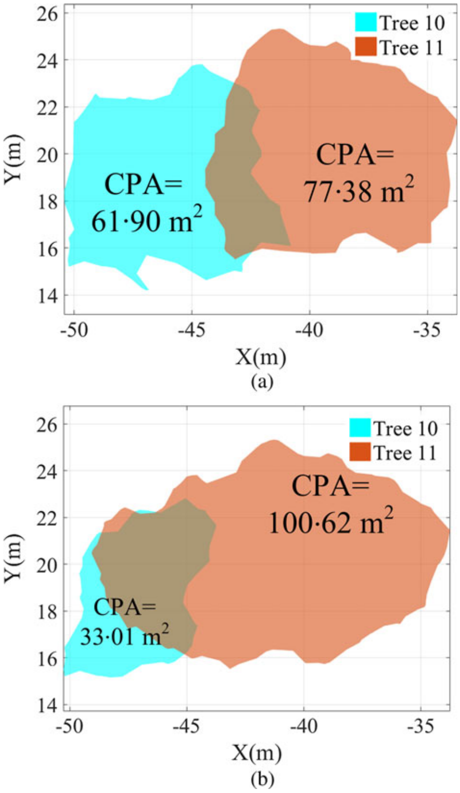

FIG. 15. Crown projection area (CPA) estimation from the resulting points of tree 10 and tree 11: (a) Wu's method; (b) proposed method.

kappa of Wu's method is 89% and for the presented method it is 94% for this specific group of trees.

Fig. 13(a) shows the results from Wu's method. In the highlighted (elliptical) areas, tree points were wrongly assigned to neighbouring trees. The proposed method shows an improvement of 5% with respect to Wu's method, although one part of a branch is wrongly assigned to an adjacent smaller tree (Fig. 12). As can be seen in Table II, the proposed method wrongly assigned 115 points from tree 10 to tree 11, and 508 points from tree 11 to tree 10. For Wu's method, the corresponding numbers were 196 and 86 236, respectively. Many of those points correspond to area A in Fig. 13(a). The subsequent benefit is significant when estimating the crown projection area (CPA) based on the individual tree delineation (separation) results of the two methods. Fig. 15 shows the estimation results of the CPA on points of tree 10 and tree 11 in Fig. 14. The areas of the two trees were approximated by an alpha shape with an alpha value of 0·8 m (see the section above on "Point-based Approaches"). Fig. 15 illustrates the CPA from the two results. The estimated CPA of tree 10 is 61·90 m^2 (Wu's method) versus 33·01 m^2 (proposed method), while for tree 11 the corresponding areas are 77·38 m^2 versus 100·62 m^2.

Conclusions and Recommendations

In this work a cuboid scalable individual tree delineation method has been proposed. The innovations as proposed in the "Introduction" section were implemented and validated as follows:

(1) Rather than simply considering the distance from unassigned individual cuboids to a seed cuboid, a novel adjacency analysis has been introduced that follows the shape of the canopy in assigning cuboids to individual trees. In the "Results" section it has been shown that this approach leads to a higher individual tree delineation accuracy than an existing method.
(2) The input tree points were first resampled using cuboids, whose edge lengths in each Cartesian axis direction were different (unlike conventional voxels). By using smaller edge lengths in the two horizontal directions, it is possible to increase the horizontal resolution while maintaining the vertical traversing time.
(3) Since most of the 3D space of a rectangular bounding box is not occupied by the input tree points, 3D clustering of the cuboids avoided redundancy in seed cuboid detection.
(4) Traversing cuboid layers in two directions makes the presented method more flexible to deal with point clouds obtained from different viewpoints, as implemented in airborne, mobile and static laser-scanning systems.

The presented method was demonstrated and tested on several datasets. Its applicability in individual tree delineation from point clouds obtained in different seasons was validated. In leaf-on seasons, branches are typically occluded by leaves and the continuous representation of branches in point clouds is problematic. The proposed method resamples point clouds using cuboids which reduce the influence of occlusions. The presented tests illustrate the advantages and reliability of the proposed algorithm, but also reveal some weaknesses. The input of this algorithm is tree points only, thus the tree points need to be classified and segmented from the original point clouds first by some existing method. The algorithm is tuneable by two types of input parameters. The first three parameters control the cuboid size in the three Cartesian directions; the fourth asks for a minimum tree-canopy diameter. Nevertheless, over- or under-individualisation may occur when trees vary considerably in size.

A recommendation is to tile input tree points according to approximate tree size. This is expected to generate more reliable delineation results. Also, the proposed method is based on cuboids, which correspond to uniform 3D grids. Since most space in a bounding box of the input point cloud data is not occupied by tree points, uniform 3D resampling to cuboids reduces memory redundancy in the processing. Therefore, an additional recommendation is to organise the cells using octrees rather than cuboids in a uniform 3D grid. The applicability of the proposed method in forest applications with dense trees and undergrowth has not been tested and needs further research. A possible application of the proposed method is tree-parameter extraction. Based on the organisation of tree points using cuboids, tree diameters at breast height can be obtained from the trunk layers. Given volumetric reference measurements and wood density values, a possible next study topic could be to investigate whether the entire volume of trees, or even biomass, could be estimated starting from the delineation results. The authors are also making efforts to maintain a webpage of this work, reporting on further improvements of the proposed method (Individual Tree Delineation, 2018).

REFERENCES

Aijazi, A. K., Checchin, P. and Trassoudaine, L., 2013. Segmentation based classification of 3D urban point clouds: a super-voxel based approach with evaluation. *Remote Sensing*, 5(4): 1624–1650.

Axelsson, P., 2000. DEM generation from laser scanner data using adaptive TIN models. *International Archives of Photogrammetry and Remote Sensing*, 33(B4): 110–117.

Babahajiani, P., Fan, L., Kamarainen, J. and Gabbouj, M., 2015. Automated super-voxel based features classification of urban environments by integrating 3D point cloud and image content. *IEEE International Conference on Signal and Image Processing Applications (ICSIPA),* Kuala Lumpur, Malaysia. Pages 372–377.

Bienert, A., Queck, R., Schmidt, A., Bernhofer, C. and Maas, H.-G., 2010. Voxel space analysis of terrestrial laser scans in forests for wind field modeling. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(5): 92–97.

Bremer, M., Schmidtner, K. and Rutzinger, M., 2015. Reconstruction of forest geometries from terrestrial laser scanning point clouds for canopy radiative transfer modelling. *Geophysical Research Abstracts*, 17: article no. 11819.

Bucksch, A. and Lindenbergh, R., 2008. CAMPINO – a skeletonization method for point cloud processing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1): 115–127.

Cabo, C., Ordóñez, C., Garcia-Cortés, S. and Martinez, J., 2014. An algorithm for automatic detection of pole-like street furniture objects from mobile laser scanner point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87: 47–56.

Cohen, J., 1960. A coefficient of agreement of nominal scales. *Educational and Psychological Measurement*, 20(1): 37–46.

Cormen, T. E., Leiserson, C. E. and Rivest, R. L., 2001. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, USA. 1312 pages.

Cottone, N. and Ettl, G., 2001. Estimating populations of whitebark pine in Mount Rainier National Park, Washington, using aerial photography. *Northwest Science*, 75(4): 397–406.

Delft City Trees, 2017. *Bomen in beheer door gemeente Delft*. https://www.arcgis.com/home/item.html?id=b8eee2fed05949d1a4926ba1e41ad3c6#visualize [Accessed: 9th December 2017].

Duncanson, L. I., Cook, B. D., Hurtt, G. C. and Dubayah, R. O., 2014. An efficient, multi-layered crown delineation algorithm for mapping individual tree structure across multiple ecosystems. *Remote Sensing of Environment*, 154: 378–386.

Edelsbrunner, H., Kirkpatrick, D. and Seidel, R., 1983. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4): 551–559.

Fieber, K. D., Davenport, I. J., Ferryman, J. M., Gurney, R. J., Walker, J. P. and Hacker, J. M., 2013. Analysis of full-waveform LiDAR data for classification of an orange orchard scene. *ISPRS Journal of Photogrammetry and Remote Sensing*, 82: 63–82.

Guo, L., Chehata, N., Mallet, C. and Boukir, S., 2011. Relevance of airborne lidar and multispectral image data for urban scene classification using random forests. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(1): 56–66.

Holmgren, J. and Persson, Å., 2004. Identifying species of individual trees using airborne laser scanner. *Remote Sensing of Environment*, 90(4): 415–423.

Hopkinson, C., Chasmer, L., Young-Pow, C. and Treitz, P., 2004. Assessing forest metrics with a ground-based scanning lidar. *Canadian Journal of Forest Research*, 34(3): 573–583.

Hyyppä, J., Kelle, O., Lehikoinen, M. and Inkinen, M., 2001. A segmentation based method to retrieve stem volume estimates from 3-D tree height models produced by laser scanners. *IEEE Transactions on Geoscience and Remote Sensing*, 39(5): 969–975.

Individual Tree Delineation, 2018. https://sites.google.com/site/treedelineation/ [Accessed: 30th January 2018].

Kraus, K. and Pfeifer, N., 1998. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(4): 193–203.

Lahivaara, T., Seppanen, A., Kaipio, J. P., Vauhkonen, J., Korhonen, L., Tokola, T. and Maltamo, M., 2014. Bayesian approach to tree detection based on airborne laser scanning data. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5): 2690–2699.

Li, W., Guo, Q., Jakubowski, M. K. and Kelly, M., 2012. A new method for segmenting individual trees from the lidar point cloud. *Photogrammetric Engineering & Remote Sensing*, 78(1): 75–84.

Lin, C., Chen, J., Su, P. and Chen, C., 2014. Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 94: 70–79.

Lindberg, E., Eysn, L., Hollaus, M., Holmgren, J. and Pfeifer, N., 2014. Delineation of tree crowns and tree species classification from full-waveform airborne laser scanning data using 3-D ellipsoidal clustering. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(7): 3174–3181.

Liu, H., Wang, L. and Jezek, K. C., 2006. Automated delineation of dry and melt snow zones in Antarctica using active and passive microwave observations from space. *IEEE Transactions on Geoscience and Remote Sensing*, 44(8): 2152–2163.

Lu, X., Guo, Q., Li, W. and Flanagan, J., 2014. A bottom-up approach to segment individual deciduous trees using leaf-off lidar point cloud data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 94: 1–12.

McDaniel, M. W., Nishihata, T., Brooks, C. A., Salesses, P. and Iagnemma, K., 2012. Terrain classification and identification of tree stems using ground-based LiDAR. *Journal of Field Robotics*, 29(6): 891–910.

Moskal, L. M. and Zheng, G., 2012. Retrieving forest inventory variables with terrestrial laser scanning (TLS) in urban heterogeneous forest. *Remote Sensing*, 4(1): 1–20.

Papon, J., Abramov, A., Schoeler, M. and Wörgötter, F., 2013. Voxel cloud connectivity segmentation – supervoxels for point clouds. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA. Pages 2027–2034.

Pauling, F., Bosse, M. and Zlot, R., 2009. Automatic segmentation of 3D laser point clouds by ellipsoidal region growing. *Australasian Conference on Robotics and Automation (ACRA 09)*, Sydney, NSW, Australia. Pages 11–20.

Persson, Å., Holmgren, J. and Söderman, U., 2002. Detecting and measuring individual trees using an airborne laser scanner. *Photogrammetric Engineering & Remote Sensing*, 68(9): 925–932.

Puttonen, E., Jaakkola, A., Litkey, P. and Hyyppä, J., 2011. Tree classification with fused mobile laser scanning and hyperspectral data. *Sensors*, 11(5): 5158–5182.

Rahman, M. Z. A., Gorte, B. G. H. and Bucksch, A. K., 2009. A new method for individual tree delineation and undergrowth removal from high resolution airborne LiDAR. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(3/W8): 283–288.

Raumonen, P., Casella, E., Calders, K., Murphy, S., Åkerblom, M. and Kaasalainen, M., 2015. Massive-scale tree modelling from TLS data. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Science*, 2(3/W4): 189–196.

Rutzinger, M., Pratihast, A. K., Oude Elberink, S. and Vosselman, G., 2010. Detection and modelling of 3D trees from mobile laser scanning data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(5): 520–525.

Rutzinger, M., Pratihast, A. K., Oude Elberink, S. J. and Vosselman, G., 2011. Tree modelling from mobile laser scanning data-sets. *Photogrammetric Record*, 26(135): 361–372.

Sirmacek, B. and Lindenbergh, R., 2015. Automatic classification of trees from laser scanning point clouds. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3/W5): 137–144.

Sithole, G. and 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.

Solberg, S., Naesset, E. and Bollandsas, O. M., 2006. Single tree segmentation using airborne laser scanner data in a structurally heterogeneous spruce forest. *Photogrammetric Engineering & Remote Sensing*, 72(12): 1369–1378.

Strom, J., Richardson, A. and Olson, E., 2010. Graph-based segmentation for colored 3D laser point clouds. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, Taipei, Taiwan. Pages 2131–2136.

Tang, S., Dong, P. and Buckles, B. P., 2013. Three-dimensional surface reconstruction of tree canopy from lidar point clouds using a region-based level set method. *International Journal of Remote Sensing*, 34(4): 1373–1385.

van der Sande, C., Soudarissanane, S. and Khoshelham, K., 2010. Assessment of relative accuracy of AHN-2 laser scanning data using planar features. *Sensors*, 10(9): 8198–8214.

van Deusen, P., 2010. Carbon sequestration potential of forest land: management for products and bioenergy versus preservation. *Biomass and Bioenergy*, 34(12): 1687–1694.

Vaughn, N. R., Moskal, L. M. and Turnblom, E. C., 2012. Tree species detection accuracies using discrete point Lidar and airborne waveform Lidar. *Remote Sensing*, 4(2): 377–403.

Vega, C., Hamrouni, A., El Mokhtari, S., Morel, J., Bock, J., Renaud, J.-P., Bouvier, M. and Durrieu, S., 2014. PTrees: a point-based approach to forest tree extraction from lidar data. *International Journal of Applied Earth Observation and Geoinformation*, 33: 98–108.

Vosselman, G., Gorte, B. G. H., Sithole, G. and Rabbani, T., 2004. Recognising structure in laser scanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(8/W2): 33–38.

VOSSELMAN, G. and MAAS, H.-G. (Eds), 2010. *Airborne and Terrestrial Laser Scanning*. Whittles, Dunbeath, Scotland. 336 pages.

WANG, Y., WEINACKER, H. and KOCH, B., 2008. A Lidar point cloud based procedure for vertical canopy structure analysis and 3D single tree modelling in forest. *Sensors*, 8(6): 3938–3951.

WANG, Z., ZHANG, L., FANG, T., MATHIOPOULOS, P. T., TONG, X., QU, H., XIAO, Z., LI, F. and CHEN, D., 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.

WU, B., YU, B., YUE, W., SHU, S., TAN, W., HU, C., HUANG, Y., WU, J. and LIU, H., 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.

YANG, B., WEI, Z., LI, Q. and LI, J., 2012. Automated extraction of street-scene objects from mobile lidar point clouds. *International Journal of Remote Sensing*, 33(18): 5839–5861.

YANG, B. and DONG, Z., 2013. A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 81: 19–30.

YU, W., HE, F. and XI, P., 2010. A rapid 3D seed-filling algorithm based on scan slice. *Computers and Graphics*, 34(4): 449–459.

ZHENG, G., CHEN, J. M., TIAN, Q. J., JU, W. M. and XIA, X. Q., 2007. Combining remote sensing imagery and forest age inventory for biomass mapping. *Journal of Environmental Management*, 85(3): 616–623.

ZHONG, R., WEI, J., SU, W. and CHEN, Y. F., 2013. A method for extracting trees from vehicle-borne laser scanning data. *Mathematical and Computer Modelling*, 58(3–4): 733–742.

Résumé

La détection et la documentation manuelle des arbres est une tâche fastidieuse. Le lidar offre une solution possible pour l'inventaire automatique des arbres. Les approches existantes pour la segmentation des arbres dans des nuages bruts de points ne proposent pas de méthodes efficaces et adaptées à toutes les échelles pour séparer des arbres individuels échantillonnés par différents systèmes lidar avec une qualité acceptable en toute circonstance. Cette étude propose et valide un nouvel algorithme pour la délimitation efficace d'arbres individuels à partir de nuages de points lidar. L'algorithme proposé commence par rééchantillonner les points dans des cellules cubiques (voxels), puis regroupe les cellules connexes en traversant verticalement les couches de cellules. Les arbres proches sont identifiés grâce à une nouvelle analyse d'adjacence de cellules. La performance de cet algorithme en termes d'adaptabilité au changement d'échelle est validée à partir de nuages de points issus de systèmes laser à balayage aérien, mobile et terrestre. Une validation basée sur des données de terrain de référence fait état d'une amélioration de 89% à 94% par rapport à des méthodes connues pour un temps de calcul comparable.

Zusammenfassung

Eine Überwachung und Dokumentation von Bäumen ist sehr arbeitsaufwändig. Lidar bietet das Potential für automatische Bauminventur. Es gibt Ansätze zur Segmentierung von Bäumen aus Punktwolken, die allerdings noch nicht in der Lage sind, einzelne Bäume in Punktwolken verschiedener Lidar-Systeme zuverlässig und mit ausreichender Qualität unter vielfältigen realen Bedingungen zu separieren. Diese Studie stellt einen neuen Algorithmus zur effizienten Erfassung von Bäumen in Lidar-Punktwolken dar. Der Algorithmus bildet Punkte mit Hilfe von quaderförmigen (Voxel) Zellen um. Nacheinander verbundene Zellen werden durch vertikale Traverse der Zellschichten akkumuliert. Bäume in nächster Nachbarschaft werden durch eine neuartige Zellanalyse identifiziert. Der Vorteil der Skalierbarkeit des Algorithmus wird an flugzeuggestützten, mobilen und terrestrischen Laserscanpunktwolken validiert. Anhand von Solldaten ist festzustellen, dass bei gleicher Rechenzeit, eine Verbesserung von 89% bis 94% im Vergleich zu aktuellen Verfahren erzielt werden kann.

Resumen

Monitorizar y documentar manualmente árboles es un trabajo intensivo. El lidar proporciona una posible solución para la generación automática del inventario de árboles. Los enfoques existentes para segmentar árboles a partir originalmente de nubes de puntos lidar carecen de métodos escalables y eficientes que separen árboles individuales muestreados por diferentes sistemas lidar con calidad suficiente bajo todas las circunstancias. En este estudio, se presenta y valida un algoritmo nuevo para la delimitación eficiente de árboles individuales a partir de nubes de puntos lidar. El algoritmo propuesto primero remuestrea los puntos usando células cuboides (vóxels). Los vóxels adyacentes se acumulan atravesando verticalmente las capas de vóxels. Basados en un nuevo análisis de adyacencia de vóxels se identifican árboles que están próximos. El rendimiento escalable de este algoritmo se valida con nubes de puntos lidar aerotransportados, móviles y terrestres. La validación con verdad terreno demuestra una mejora del 89% al 94% en comparación con un método de vanguardia, mientras que el tiempo de cálculo es similar.

摘要

人工监控和记录树木的信息是劳动密集的工作。激光扫描为树木清查数据库自动化提供了一种可能的解决方案。现有基于原始点云的树木分割方法缺乏可扩展性与效率，不能适用于不同情况下的不同激光扫描系统。本研究提出并验证了一种新的由激光扫描点云中提取单一树木的有效算法。该方法首先使用长方体单元(体元)将点云重新取样， 再连续联接垂直方向各层的相通单元。使用一种创新的单元邻接分析方法，根据邻近性以识别树木。利用机载、地面移动和固定式激光扫描所获得点云数据验证本算法的性能与可扩展性，与地面控制数据相比较，本文所提出的方法较现有的最佳方法在正确性上由 89% 提高到 94%，而所需计算时间接近。