Computers & Geosciences 156 (2021) 104900

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Tree segmentation and change detection of large urban areas based on airborne LiDAR☆

Anett Fekete, Mate Cserep *

Faculty of Informatics, ELTE Eötvös Loránd University, Budapest, Hungary

ARTICLE INFO

Keywords:
Tree segmentation
Vegetation
Change detection
LiDAR
Point cloud
Digital elevation model

ABSTRACT

As the utilization of LiDAR (Light Detection and Ranging) is getting more affordable and available for a wider audience, the analysis of point clouds constructed by laser scanning is earning more attention. Airborne LiDAR is especially useful in the analysis and classification of land objects. We are able to determine if they are natural or artificial objects and what changes occurred to them throughout time by examining multi-temporal data. The goal of our research was to define a completely automatized methodology for the segmentation of vegetation (specifically trees) in urban environment, followed by the qualification and quantification of change detection. Our proposed algorithm provides a robust approach designed to scale dynamically to large areas, in contrast to existing methods that require manual or semi-supervised human interaction and can only be applied on relatively small areas. The algorithm was tested on parts of the Dutch and the Estonian altimetry archives, point cloud datasets that provide several terabytes of data. It was proved to be an effective method for the qualified and quantified change detection of trees, including height and volume changes.

1. Introduction

Change detection of vegetation has become a highly important issue in our days since the alteration of the ecosystem affects urban life, such as living conditions and urban planning. Expansion of urban area and industrialization both have a great impact on vegetation growth, therefore it is important to continuously monitor and analyze the changes . Plants can be ranked by certain qualities, e.g. type, height, or in a more specific case canopy density of trees, etc.

Multispectral satellite imagery and aerial photography are still the main data source for land cover classification. LiDAR is a possible alternative of raster scanning although not as widespread due to the higher cost and limited availability. As a result there are no comprehensive LiDAR records of the Earth. One of the greatest advantages of LiDAR is that it makes it possible to compare 3D point clouds which not only makes classification, but filtering temporary objects, and change detection easier than using 2D imagery. On the other hand, the comparison of point clouds is definitely more complex algorithmically compared to raster images since the points are not likely to be located in the same spatial point when examining multiple point clouds from different epochs. In fact, significant differences may occur in the quality and density of point clouds when the same area is recorded in different epochs.

While change detection and land cover classification are well-researched topics for which several manual or semi-automatic methods exist (Hyyppä et al., 2001; Yu et al., 2006; Yang et al., 2020), most available algorithms are usually applicable and were tested only for smaller areas. The goal of this paper is to define a robust, automated tree segmenting method that is independent of tree species, targeting especially large urban and suburban areas. Through evaluating multiple scanning epochs of the same area, the main goal of our method is to quantify the changes of trees. This includes changes in canopy volume, tree height and tree presence.

The most important contributions of this paper are (*i*) defining a novel approach for segmenting trees with multiple local maximum points and overlapping trees based on both a horizontal and vertical distance between the formed clusters; (*ii*) comparing two different pairing algorithms, centroid distance and Hausdorff distance for matching detected trees between the evaluated epochs; and (*iii*) creating a robust, automatized, open-source algorithm pipeline and software framework capable of processing large datasets efficiently. To the best of our knowledge, no previous research or work targeted the automatized processing of larger territories. The implementation was tested on the Dutch and the Estonian national altimetry archives, which are public point cloud datasets acquired by airborne laser scanning.

☆ This work is supported by the Hungarian Government, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00001).
* Corresponding author.
E-mail addresses: afekete@inf.elte.hu (A. Fekete), mcserep@inf.elte.hu (M. Cserep).

https://doi.org/10.1016/j.cageo.2021.104900
Received 31 December 2020; Received in revised form 15 June 2021; Accepted 27 July 2021
Available online 5 August 2021

The rest of the paper is structured as follows: Section 2 overviews the related literature of land coverage classification, single tree segmentation and change detection methods based on LiDAR data. Section 3 presents the study dataset, the AHN point cloud. In Section 4 we describe our methodology of tree segmentation and pairing between multiple epochs, followed by the evaluation of change analysis. In Section 5 we test our methodology on multiple sample areas, discuss and validate the results against a reference dataset, and compare our results with other methods in literature. Finally, we conclude our work in Section 6 and discuss future work.

2. Related work

2.1. Classification of land coverage

Various approaches exist for processing point clouds and using them in classification, with special regard to vegetation. Antonarakis et al. (2008) describe two models for distinguishing natural and planted forestry, one of which includes ground hits while the other one does not. The former relies on the use of bimodal distribution skewness and kurtosis models. A skewness value and a kurtosis value was chosen to define which pixels belonged to a natural forest, and which ones belonged to a planted one. This method also proved to be useful in determining if the tree was younger or mature. Bellakaout et al. (2016) mention a classification method that identifies and utilizes different contour types by which objects can be classified. The paper describes four different object classes that contain terrestrial objects identified by contours: superior contour, inferior contour, uniform surface and non-uniform surface. These classes allow the extraction of soil, vegetation, buildings and roads. Song et al. (2002) describe a study whose aim was to evaluate the use of LiDAR data in land cover classification. The key of this method is to classify objects based on the intensity of reflection. The point clouds were converted into grid form by the IDW and the Kriging interpolation methods and the acquired intensity data was divided into four classes: grass, tree, asphalt road, and house roof. Beside the classification of vegetation, attribute estimation such as tree or canopy height can also be performed (Hamraz et al., 2017). The combined usage of aerial photography and airborne LiDAR to enable more precise classification and tree height estimation in forestry has also been studied (Suárez et al., 2005).

Machine learning has become a frequently utilized approach in classification methods, and land coverage classification is no exclusion. Shaker et al. (2019) used machine learning algorithms in order to automatize land–water classification. Such algorithms can also be used for more specific land coverage classification. Sun et al. (2019) compare three deep learning methods to determine if (and to what extent) they are suitable for the mapping of tree species in a tropical environment.

2.2. Single tree segmentation

Segmenting single (individual) trees is a well-researched topic with several somewhat similar approaches. Papers by Kaartinen et al. (2012) and Eysn et al. (2015) give very good summaries and comparison of the existing methods based on their matching rates among other aspects. Local maximum detection in trees is a widely used technique in tree segmentation which is applied along with other methods. Complementing local maximum detection, these methods include low-pass filtering with a convolution matrix (Monnet et al., 2010; Eysn et al., 2012; Kaartinen et al., 2012; Dalponte et al., 2014) for image blurring. Clustering with region growing (Dalponte et al., 2014) and the watershed algorithm are also frequently applied in tree segmentation in order to determine tree edges in the point cloud (Sambugaro et al., 2013; Lindberg et al., 2014; Yang et al., 2020). Besides pointwise segmentation, delineating tree contours is also applied in multiple papers (Lindberg et al., 2014; Jakubowski et al., 2013; Sambugaro et al., 2013).

2.3. Change detection in point clouds

There are two distinct types of change detection: binary and quantifiable change. The former is a simple approach which only determines whether there was a change in the scene. Its result is mostly a binary map where no change is indicated by 0, change is indicated by 1. Quantifiable change, on the other hand, strives to find a more complex answer to non-binary questions on the exact nature of the change. The binary type is often simply called change detection, while quantifiable change detection is called deformation analysis (Vosselman and Maas, 2010).

The existing methods can be grouped into the following categories.

$2\frac{1}{2}$ dimensional visibility maps: The dimension of the point cloud can be reduced to $2\frac{1}{2}$D if the observation happens from a fixed scanner position. This way objects can appear, disappear, and move in the point cloud, revealing contingent binary changes in the scene. A spherical coordinate system is used to determine if there are any changes in the point clouds: the dataset that was captured later (and perhaps from a different but fixed standpoint) is transferred into the coordinate system. Afterwards, it is checked whether the reference cloud points are present in the new cloud. If a point is there in the new cloud, it is checked whether it represents an object at this location or not. Points can also be invisible due to occlusion (Vosselman and Maas, 2010).

Direct DEM comparison: This binary method extends digital elevation models with a simple subtraction process and it adjusts the results to eliminate errors caused by misregistration (Vosselman and Maas, 2010). It can be used both in urban areas and in nature. In addition, it is worth noting that direct DEM comparison is a decent quantifiable method as well, since exact height and volume changes can be measured by comparing DEMs. dos Santos et al. (2020) use DSM comparison to produce a difference DSM for their method of building change detection, as a first step to determine height changes in the point cloud.

Pointwise deformation analysis: Most methods for change detection are based on DEMs constructed from point clouds. Other quantifiable methods examine raw point clouds in order to achieve more accurate results. Butkiewicz et al. (2008) describe a change detection method that finds deformations in urban environment over time , both in vegetation and buildings. It also uses raw LiDAR point clouds. It calculates bounds for what could be scanning error or geological variation and compares the distance of points in different scans. This method is capable of detecting changes in individual and grouped objects as well.

Object-oriented deformation analysis: This quantifiable method makes use of the observation that man-made objects are mostly constructed by geometric shapes like planes and cylinders (Lindenbergh and Pietrzyk, 2015). Although these shapes are easily recognizable by only a couple of points, the cloud still consists of hundreds of thousands to millions of points. This redundancy might be used to determine every facet of change in the scene the dataset represents.

LiDAR proves to be an effective technology for monitoring changes in vegetation . Vegetation is mostly represented by irregularly distributed points in the dataset. LiDAR-based monitoring is also efficient because the laser beams easily penetrate through leaves and the canopy of trees. The denser the point cloud, the easier it is to detect changes in height and land coverage. In addition to changes in trees and forestry, changes in the total biomass of an area can be monitored by analyzing multitemporal laser scanned data. The point clouds can be collected both by terrestrial and airborne laser scanning.

Meyer et al. (2013) tried to estimate the biomass change of a 0.5 km^2 tropical area. They collected canopy height metrics and used an importance analysis method to evaluate the importance of the variables. They demonstrated the use of spatial scales in biomass estimation and determined that they give more accurate results when used on finer scales. Kaasalainen et al. (2014) use the TIN model to detect quantitative changes in tree growth and litter production. They created a flexible surface model of each tree using the QSM method (Raumonen et al., 2013) and compared it to the model created with the TIN model and gave results of 10% accuracy.

3. Dataset description

Two multitemporal point cloud altimetry archives were selected as study datasets (Fekete and Cserép, 2021).

- The Dutch *Actueel Hoogtebestand Nederland (AHN)*,[1] which covers The Netherlands, 41.526 km^2 per each data acquisition. There are three different datasets in AHN, from which we chose the *AHN-2* and the *AHN-3* point clouds due to their higher density of points (6–10 points/m^2). The AHN-2 point cloud (PDOK, 2013) was scanned between 2007 and 2012, while the AHN-3 acquisition (PDOK, 2015) was started in 2014 and was finished in 2019.
- The *Estonian Elevation Dataset*[2] produced by the Estonian Land Board also contains three data acquisitions and covers the complete territory of Estonia, 45.339 km^2 per each dataset. Since larger cities were covered multiple times (typically each year) with low-altitude flights as part of the last scanning between 2017 and 2020, we used the point clouds from this time span, again due to their higher density of points (18-30 points/m^2).

Raw point clouds are extremely large in both datasets (Swart, 2010), e.g. AHN is built up of 1372 tiles each of which covers 31.25 km^2. A single tile is typically over 15 GB (in LAS format). AHN offers preprocessed DEMs with 0.5 m and 5 m resolution whose data size is 0.5 GB per tile. The vertices in the DEMs (Mukherjee et al., 2013; San and Suzen, 2005; Zhang and Montgomery, 1994) contain the accumulated height values, calculated by e.g. the inverse distance weighting (IDW) algorithm (Shepard, 1968), typically represented in GeoTiff format. This model also reduces the number of data points to analyze, by controlling the grid size of the DEM.

In order to spare time and computational cost, and produce the most accurate results possible, we used the 0.5 m resolution DEMs of AHN-2 and AHN-3 for our research. In case of the Estonian Elevation Dataset, preprocessed DEMs are only available for the latest acquired state, therefore surface and terrain models were generated from the raw point clouds with the open-source *CloudCompare* tool, with the same 0.5 m resolution.

3.1. Demonstration area

A demonstration area had to be selected to properly showcase the proposed methodology, which contains urban environment with plenty of vegetation, preferably with considerable changes between the two analyzed epochs of data acquisition. The campus of the Delft University of Technology and its surroundings were selected as a test area, which fulfills these criteria. The city of Delft was scanned in the years 2008 and 2014 for AHN-2 and AHN-3, respectively. The steps of the algorithm in Section 4 will be demonstrated on a sample area of the TU Delft Campus. This is presented with Google Satellite image in Fig. 1. We illustrate how the algorithm works by step-by-step snapshots from the processing of the AHN-2 dataset.

The Estonian Elevation Dataset will be used in Section 5 for evaluating our proposed methodology on various sample territories.

[1] Actueel Hoogtebestand Nederland: http://www.ahn.nl/index.html.
[2] Estonian Elevation Dataset: https://geoportaal.maaamet.ee/eng/.

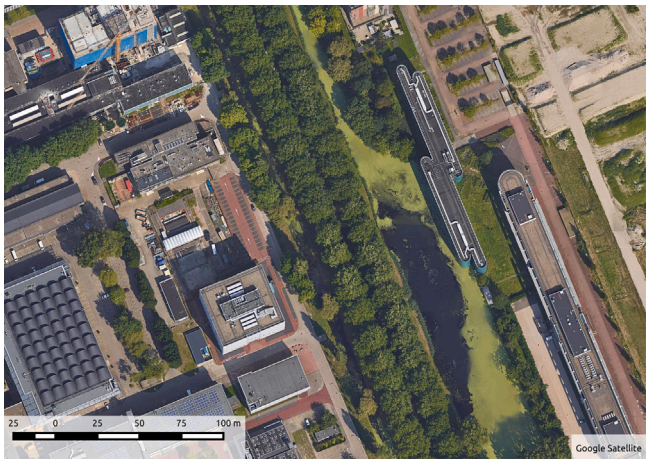

Fig. 1. Satellite image of the study area.

4. Methodology

We define an algorithm which, when executed on a preprocessed point cloud, produces a cluster map that covers every tree in the area by exactly one cluster. Upon evaluation on multitemporal point clouds, changes in tree presence, height and canopy volume can be calculated. We will describe the steps of the algorithm in the coming subsections as follows.

1. Produce a canopy height model of the DSM and DTM of the area in the same epoch.
2. Remove excess local maximum points from the CHM.
3. Interpolate the erroneous nodata points in the CHM.
4. Collect the remaining local maximums.
5. Construct a cluster map from the collection of seed points in which one cluster is equivalent to one tree.
6. Apply morphological opening on the cluster map to erode outlier points.
7. Pair up clusters of the same area in different epochs and seclude trees without a pair in both epochs.
8. Calculate change metrics in the vegetation:
    8.1. height difference of tree pairs;
    8.2. volume difference of tree pairs and epochs.

The steps of the algorithm are depicted in Fig. 2 and are described in detail in the following sections.

4.1. Producing canopy height models

The canopy height model (CHM) represents the height of individual trees that are present in the examined area. It is constructed by subtracting a digital terrain model (DTM) from a digital surface model (DSM) (Hyyppä et al., 2008). A DSM contains a point cloud of the top of the surface depicting the terrain and the natural and man-made environmental elements. On the other hand, a DTM represents the bare-earth surface, this is why their difference provides a fine model of the vegetation. The results of canopy height model production are illustrated by Fig. 3.

4.2. Low-pass filtering

Classification originates and expands from a distinctive *seed point* in the CHM. For trees, the obvious choice is to originate a point set (called *cluster*) from the highest point of the tree. However, canopy height models are not reliable for good clustering on their own

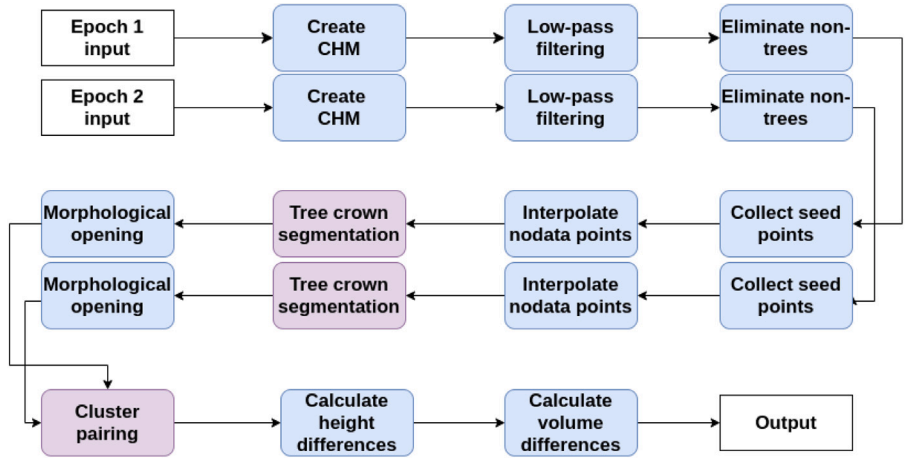

Fig. 2. Flowchart of the algorithm. Steps highlighted with purple background contain the main contributions of the proposed algorithm: segmenting trees with multiple local maximums and overlapping tree clusters; and comparing centroid and Hausdorff distance in the cluster pairing step..

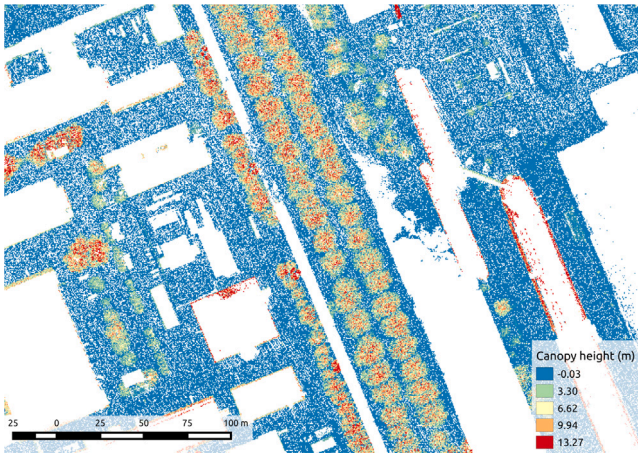

Fig. 3. AHN-2 canopy height model. All height values are represented in meters.

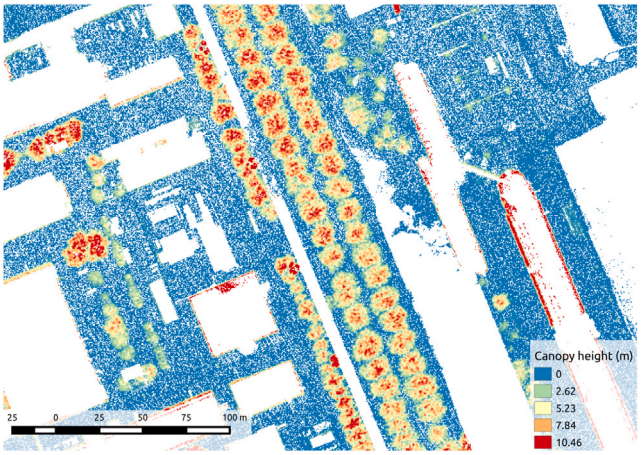

Fig. 4. Low-pass filtering performed on AHN-2.

because they might contain multiple local maximum points per tree. This makes future classification results have smaller clusters. This is why the number of local maximum points needs to be reduced by removing several unnecessary peaks. The elimination was done by a sweeping-window Gaussian blurring transformation, with the following convolution matrix:

$$\frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

This low-pass filtering is similar to the one used by Hyyppä et al. (2001), however we introduced a special rule for handling *nodata values* – points in the raster grid where the Z coordinate is missing.[3] These nodata values were treated as zero values when performing the multiplication, but the divisor 16 was also reduced accordingly in such cases. With this modification we managed to eliminate the distorting effect of nodata values on the low-pass filter. The results of low-pass filtering are illustrated by Fig. 4.

Our convolution matrix is a 3×3 Gaussian filter, commonly used in image blurring. We have experimented with a 5×5 Gaussian convolution matrix with varying results: while in some cases it eliminated more local maximums for larger trees, it also hindered the correct detection of small trees. Ultimately, the difference in the number of detected trees was below 1%, therefore the simpler 3×3 convolution matrix was used.

4.3. Elimination of low points

The previously filtered canopy height model still contains points that do not belong to trees. These points typically come from vegetation that is shorter than an average tree. In order to have only trees in the CHM, we executed a sweeping-window transformation on the model that erased every point below a threshold value[4] of 1.5 m. The results of the elimination of low points are illustrated in Fig. 5.

4.4. Collecting local maximum points

First of all, we needed the seed points of the prospective clusters. As mentioned before, the most reasonable decision was to set the tree

[3] For surfaces that absorb the laser pulse (e.g. water), the DEM sources contain nodata values. The CHMs generated in Section 4.1 also contain nodata values for the grid positions where one or both source datasets have a nodata value.

[4] The constant of 1.5 m was adapted to the trees of the temperate climate zone. It may vary depending on the average height of trees in an area.

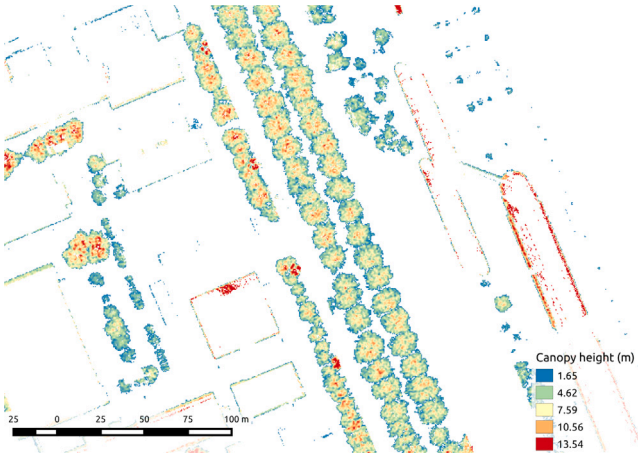

Fig. 5. Elimination of low points performed on AHN-2.

tops as seed points. This was achieved by executing a sweeping-window calculation on the point cloud which ran a 3×3 kernel matrix (*window*) through the points and saved the point to a container if it was located higher than all its neighboring points, meaning it was a local maximum point.

4.5. Interpolation of nodata points

In order to achieve more accurate results and eliminate holes, we need to fill the nodata points that are near a potential cluster. We achieved this by applying a type of majority filter (Gurney et al., 1983): the value of a nodata point is calculated as the arithmetic mean of the adjacent points with value.

4.6. Tree crown segmentation

Once the seed points are collected, the next step is to construct clusters of them that cover the same tree and extend them by classifying the remaining points. In order to define a cluster map that covers the vegetation tree by tree, 2 constant values are needed:

- *Maximum horizontal value,* max_h: the assumed greatest horizontal radius that a tree canopy can reach calculated from its seed point.
- *Maximum vertical value,* max_v: the assumed greatest vertical difference of the seed point and another arbitrary point in a cluster.

A cluster will be constructed by gradually expanding the set of points originating from the seed point, for which we define the *neighbors of a cluster* as the points adjacent to the cluster but not part of it or any other cluster. No other limitations are given when we collect the neighbors of a cluster. The neighboring points can be used for various purposes that have their own limitations. Further filtration is carried out later.

For the created clusters it is still possible that multiple seed points are present in one single tree, even though a local maximum-decreasing step was described in Section 4.2. This might occur when a tree consists of multiple local peaks that surround local *valleys* (Chen et al., 2006). If two (or more) clusters cover the same tree then they should be merged. In order to determine whether a valley between seed points is empty space between two trees, or a local valley in a single tree, the ratio r of the tree heights and the depth of the valley has to be calculated. For this calculation, we chose the seed point of the shorter tree, so if $r > 1.0$, then the valley defines separate trees, otherwise it is a local valley in one tree. The possible cluster merge cases are illustrated in Fig. 6:

(A) depicts the case where two tall trees are very close. The valley between them is marking that the seed points belong to different trees.
(B) illustrates the case when a tall and a shorter tree are close. This case does not require merging either.
(C) [(A)]shows that when there is a valley inside a (taller) tree, merging is required.
(D) depicts when two shorter trees are close. This case does not require merging either.

Equipped with the collection of seed points and the depth of the valley, the following algorithm is described for the construction of the cluster map:

1. Define max_h and max_v.
2. Construct a cluster for each seed point. Let the set of all clusters be C.
3. Filter the neighbors of every cluster: calculate the horizontal $d_h(p, s)$ and vertical $d_v(p, s)$ distance of a point p to the seed point s. If $d_h(p, s) \leq max_h$ and $d_v(p, s) \leq max_v$ and p is not a nodata value, then add the point to the *filtered neighbor set* of the cluster.
4. Take all cluster pairs (c_i, c_j) $(i < j)$ and the height of their seed points $z(s_i)$ and $z(s_j)$. Construct the intersection of their filtered neighbor sets.
5. Take each point p in the intersection (if there is any), and calculate the sum height difference d_z for $z(s_i)$ and $z(s_j)$ and the height of the current point, $z(p)$.

$$d_z = z(s_i) + z(s_j) - 2 \times z(p) \tag{1}$$

6. Calculate the ratio r of d_z against $z(s_i)$ and $z(s_j)$.

$$r = \frac{d_z}{min(z(s_i), z(s_j))} \tag{2}$$

If $r < 1.0$ and neither of c_i and c_j are to be merged yet, then list c_i and c_j to be merged.
7. Merge the listed cluster pairs.
8. Expand the clusters by the previously determined neighbors. Do not add points twice that form intersections, nor add points to clusters that no longer exist.
9. Repeat from step 3 until there are no changes made to the cluster map.

At the end of the algorithm, a cluster map is constructed which covers one tree by one cluster. The results of tree crown segmentation are illustrated in Fig. 7.

This step is followed by the removal of small clusters that contain too few points to cover a tree and are most likely the result of DTMs containing tall objects other than trees such as lamp posts, or reflected points on buildings. As a result of the removal step, the number of clusters is significantly decreased.

4.7. Morphological filtering

Morphological image processing (Gonzalez and Woods, 2006; Efford, 2000) is an image-manipulation method that is suitable for modifying the shape of an image. In our research, it relies on the existence of pixels, therefore it is specifically applicable for binary image processing.

Morphological opening was performed on the previously constructed clusters iteratively three times. In the erosion operation of the opening, a point was removed from the cluster if less than 6 of its neighboring points were in the cluster. In the dilation operation, a point was added to the cluster if any of its neighbors were already in it.[5]

The results of morphological opening are illustrated in Fig. 8.

[5] These constants (6 and 0) may vary according to the average canopy radius of the examined area.

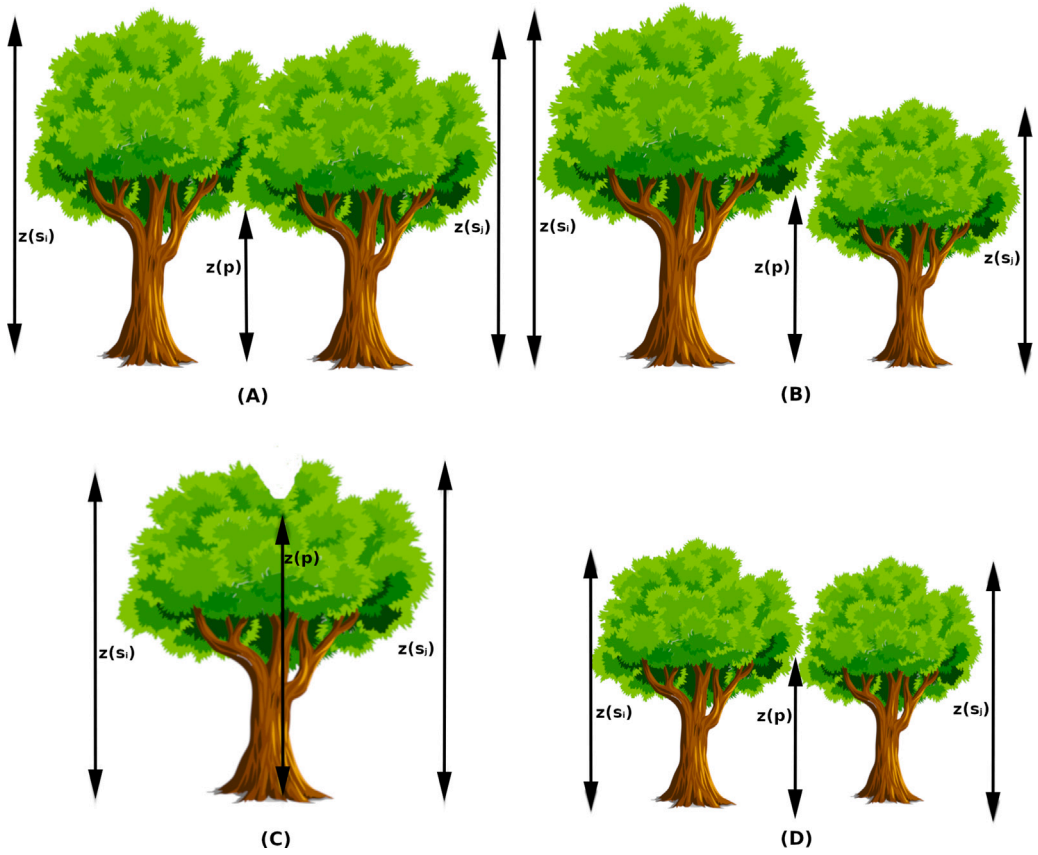

Fig. 6. The four types of possible cluster merges.

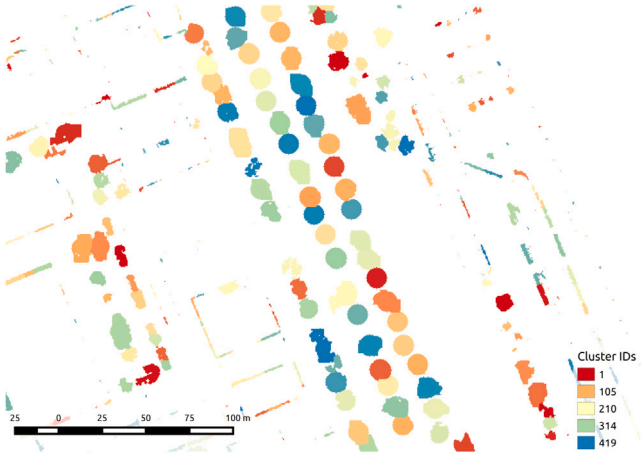

Fig. 7. Tree crown segmentation performed on AHN-2.

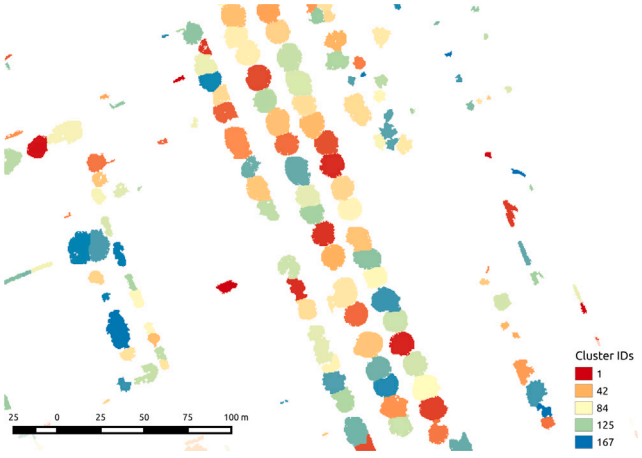

Fig. 8. Morphological opening performed on AHN-2.

4.8. Cluster pairing

While the irregularity of the original 3D data points are smoothed with the DEM construction, it still results in clusters covering the same tree not being located at the exact same place even in the DEM. Therefore we must pair up the clusters in the two epochs and also determine which trees lack a pair. If the algorithm does not find a pair for a given cluster then it is presumable that if it is present in the first point cloud then the tree it covers was cut between the epochs, and if it is present in the second one, it was planted some time after the first data acquisition.

Two methods were implemented and tested for pairing as described in the following subsections.

4.8.1. Centroid distance

Let $Epoch_1$ and $Epoch_2$ denote the two processed epochs and n_1 and n_2 be the number of clusters in the examined two cluster maps respectively. Pairing up clusters according to their distance of centroids can be done in a linear $\theta(n_1 * n_2)$ asymptotic complexity.[6] This method allows pairing one $Epoch_2$ cluster to multiple others in $Epoch_1$,

[6] Assuming that $n_1 = n_2 = n$ as a simplification it will be $\theta(n^2)$..

thus losing partial injectivity and distorting results. This problem was handled as follows.

1. Define the maximum horizontal distance max_{hc} of any two clusters. Let C_1 be the cluster set of $Epoch_1$ and C_2 be that of $Epoch_2$. Let S be the set of pairs.
2. Take every c_i $(i \in 1..|C_1|)$ that is not already paired and search for the nearest c_j $(j \in 1..|C_2|)$ by calculating the distance of their centroids where $d_h(c_i, c_j) \leq max_{hc}$ and insert the pair into S. Note, that this can result in c_j paired to multiple clusters in C_1.
3. Take each pair from S where $c_j = c_k$ $(k \in 1..|C_2|)$, search for the pair with minimal distance and erase the others from S. This step results in previously paired clusters from C_1 getting unpaired. For this reason, a new iteration is needed to search for a pair for lone clusters.
4. Repeat from step 2 until no new pairs are found. Hence, the pair set is partially injective.

4.8.2. Hausdorff-distance

The Hausdorff-distance (Rockafellar and Wets, 2009) of two point sets is a *maximin function*: the maximum distance of the cluster to the nearest point of the other cluster. Formally, for sets A and B the Hausdorff-distance can be defined as:

$$h(A, B) = max_{a \in A}(min_{b \in B}(d(a, b))) \tag{3}$$

The naive distance calculation and comparison require very high computational cost due to the high number of points in a cluster. In addition to n_1 and n_2 representing the cluster counts as defined in Section 4.8.1, let m_1 be the average number of points in an $Epoch_1$ cluster and m_2 be that in $Epoch_2$. Since the calculation requires the distance of every point in an $Epoch_1$ cluster to be calculated to every point in an $Epoch_2$ cluster and this calculation has to be done for each cluster in both cluster maps, the asymptotic bound for the calculation of Hausdorff-distance is $\theta(n_1 * n_2 * m_1 * m_2)$.[7] In order to decrease the consequent long runtime and high CPU time consumption, we applied the *early break* and the *random sampling* optimizations defined by Taha and Hanbury (2015) and refined by Zhang et al. (2017). These optimizations make the Hausdorff distance calculation more effective: in a theoretic best-case scenario[8] it reaches the same $\theta(k * l)$ execution time like the centroid-based approach. We also introduced a horizontal threshold value Th_h as the greatest possible distance between two clusters based on their centroid distance (as defined in Section 4.8.1) to further boost the performance.

Pairing by the Hausdorff-distance is not partially injective either, which problem was handled in a similar manner as for the centroid distance. The results of cluster pairing are illustrated in Fig. 9.

4.9. Difference of tree heights

Height differences of individually paired trees and average height difference of an area can be calculated from the data acquired in the previous steps.

Let C_1 be the cluster set of $Epoch_1$ and C_2 be that of $Epoch_2$. Let $z(peak(c_i))$ be the maximum height of a cluster c_i, where $c_i \in C_1$. Let $z(peak(c_j))$ be the maximum height of a cluster c_j, where $c_j \in C_2$. Let $\delta_h(c_i, c_j)$ be the height difference of this cluster pair which is calculated as follows:

$$\delta_h(c_i, c_j) = z(peak(c_j)) - z(peak(c_i)) \tag{4}$$

If $\delta_h > 0$, then the tree has grown since the first scan, and if $\delta_h < 0$, then the tree has been cut back by some natural or human force. $\delta_h = 0$ is highly unlikely even if a tree has already reached its height limit since the scanning is inconsistent and points do not usually fall to the same exact coordinates in different inspections.

Height difference for individual trees of the sample area are visualized by Fig. 10.

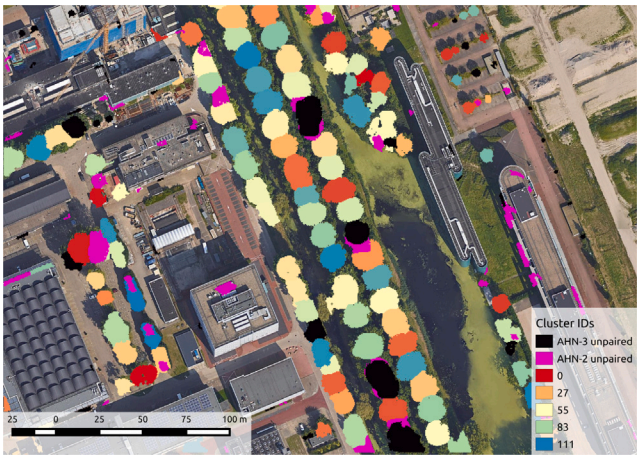

Fig. 9. Detected paired and unpaired clusters.

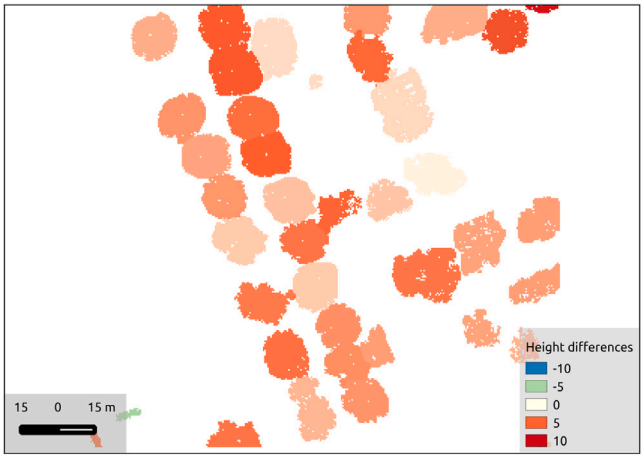

Fig. 10. Height differences of paired clusters.

4.10. Difference between tree volumes

Once each and every one of the trees in the examined area are located, we are able to calculate their individual and aggregate canopy volume. This calculation can only be done in seasons when there are actual leaves on the trees since they build up the canopy and provide the volume.

Calculating the volume of a tree is a difficult task if done very accurately since the clusters that represent tree canopies can be considered completely irregular polygons. It would be a very high-demanding task to calculate the exact volume of a cluster regarding computational costs and execution time, so we took advantage of the raster grid instead. Let V_{c_i} be the volume of cluster c_i, $z(p_n)$ be the height of a point and $|c_i|$ be the total number of points in the cluster. The computation of V_{c_i} is described by Eq. (5).

$$V_{c_i} = \sum_{n=1}^{|c_i|} z(p_n) * 0.5^2 \tag{5}$$

As mentioned in Section 3, the real distance between two grid points is $0.5m$ which is why the multiplication is done by 0.5^2. After that, the

[7] Assuming that $n_1 = n_2 = n$ and $m_1 = m_2 = m$ as a simplification, the computational cost can be approximated as $\theta(n^2 * m^2)$..

[8] In a worst-case scenario the execution time will not improve at all.

Table 1
Basic data and runtime information of the sample territories.

Name	Area	Pairing method	Runtime
Single street	0.103 km^2	centroid Hausdorff	4.09 s 26.29 s
Amsterdam city center	1.937 km^2	centroid Hausdorff	16 min 08 s 20 min 07 s
TU Delft Campus	2.143 km^2	centroid Hausdorff	20 min 30 s 30 min 17 s
Delft city center	8.640 km^2	centroid Hausdorff	9 h 04 min 10 h 51 min
Tallinn city center	1.0 km^2	centroid Hausdorff	6 min 32 s 11 min 58 s

total volume of the trees in the former epoch is extracted from that of the latter epoch, and similar conclusions can be drawn from the difference as in the case of tree height changes.

5. Results and discussion

We tested the algorithm on four territories of different sizes from the AHN dataset and one territory from the Estonian Elevation Dataset. Table 1 contains the name and area of each territory along with performance information for both pairing methods. The preprocessing steps (starting from 4.1, finished at 4.7) were carried out concurrently for the two epochs, while the further tasks (from 4.8 to 4.10) were executed sequentially without any parallelization applied on a single CPU core. Increase of runtime is non-linear with the growth of territory size due to the quadratic computational cost mentioned in Section 4.8.1.

Table 2 contains the results of the tree segmentation (number of detected clusters) and the pairing carried out by different pairing methods for each sample territory. The centroid distance pairing method turned out to be a better solution in a general scenario with respect to both computational complexity and – contradicting our initial assumptions – the accuracy of the cluster pairing. Hausdorff distance was proved to be a better choice in case of concave polygons, such as buildings (Liu et al., 2019), since it provides the distance of actual points in a cluster rather than a fictive centroid, which might lie outside of the edges of a concave polygon. Our research showed that in the case of trees, where the segmented objects are nearly always convex polygons, the simpler centroid distance can be a better approach in cluster pairing, as the centroids give a good representing point for the middle of trees. Increasing the horizontal threshold Th_h of the cluster pairing could improve the results of the Hausdorff distance based pairing with the possible side-effect of mispairing clusters in other cases.

Removed trees were only present among the $Epoch_1$ clusters, while *New trees* were only found in the $Epoch_2$ clusters. These values are unrealistically high compared to the number of detected cluster pairs. Through manual evaluation we deduced that building facades, irregular rooftops and larger statues can be misdetected as trees by our algorithm, as also observable on Fig. 9.

Table 3 contains the aggregated volume and total volume difference of both epochs for each sample territory. The individual results should be considered an approximation of the reality, since the volumes were calculated from the dimensions of the bounding box of each tree. However, since this distortion of data is present in both epochs, the difference of the two aggregated values is an accurate indicator of the overall change in biomass.

Individual height differences of trees are visualized by Fig. 10 and Fig. 12 for the *Single Street* and the *TU Delft Campus* sample territories respectively. For the larger *Delft city center* territory, an interactive online visualization was created.[9] Through this interface the users can view the raw, tree level output of the algorithm and fetch the exact altimetry difference of a marked location. A screenshot of the web application is depicted in Fig. 11.

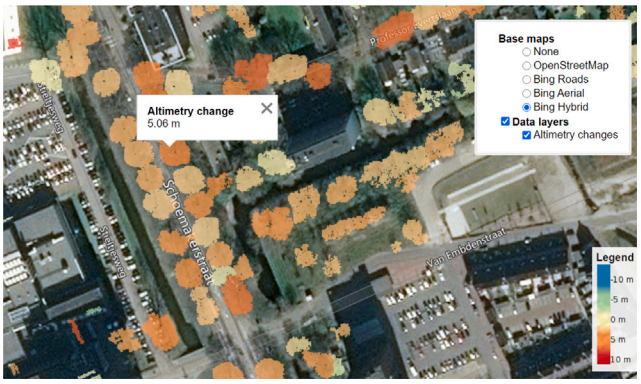

Fig. 11. Interactive tool for detailed, vegetation level altimetry change analysis for the *Delft city center* sample territory.

5.1. Validation

To assess the quality of our method, a validation of the results have been performed using the city-wide *Trees* and *Main tree structure* datasets from the *Amsterdam open geodata portal*[10] as reference.

- The *Trees* dataset contains various information about the trees that are managed by the municipality of Amsterdam, including their location and optionally the year of planting, the height and the radius of their crown. This dataset was useful to calculate a proper estimation of true positive and false negative detections.
- The *Main tree structure* dataset describes the main road network of Amsterdam where the trees are managed by the municipality and local regulations of planting, replacing, rootable space, allowed trunk circumference, etc. apply. This dataset was useful to filter out trees in private properties, so the proper rate of false positives could also be calculated.

The validation was evaluated by comparing the coordinates of detected trees in the *Amsterdam city center* sample territory for the AHN-3 dataset with the coordinates of the registered trees in the reference *Trees* dataset. We consider only those trees that (*i*) lie within the 20 meter buffer of the road network[11] in the *Main tree structure* dataset; and (*ii*) were planted before 2015, since the AHN-3 data acquisition was completed in that year for Amsterdam. To compensate for the inaccuracy of positioning between the input and the reference datasets, we allowed 3 meters of tolerance between the segmented clusters' centers and the registered coordinates in the reference dataset.

For the examined territory our proposed algorithm detected 1738 individual trees. Among the 1411 trees contained in the reference dataset, 1129 of them (80.01%) were successfully matched by our algorithm. There were 282 false negative (19.99%) and 609 false positive (35.04%) detections. An interactive online visualization of the output of the validation was also created.[12] Through manual examination of the results we concluded that the most common reasons for misdetection of trees were the following:

- Their crown was too small horizontally and was removed after the segmentation phase of the algorithm.

[9] Available at http://gis.inf.elte.hu/ahn/ahn_veg_delft_wms.html.

[10] https://maps.amsterdam.nl/open_geodata/

[11] The *Main tree structure* dataset describes the road network as a vector layer of linestrings.

[12] Available at http://gis.inf.elte.hu/ahn/ahn_veg_ams_ver.html.

Table 2
Results of tree segmentation and different pairing methods at the sample territories.

Sample territory	Epoch 1 clusters	Epoch 2 clusters	Pairing method	Tree pairs	Removed trees	New trees
Single street	168	173	centroid Hausdorff	112 106	56 62	61 67
Amsterdam city center	3 865	3 585	centroid Hausdorff	2 157 2 170	1 708 1 695	1 428 1 415
TU Delft Campus	3 834	4 128	centroid Hausdorff	2 115 1 922	1 719 1 912	2 013 2 206
Delft city center	17 375	17 634	centroid Hausdorff	9 878 9 137	7 497 8 238	7 756 8 497
Tallinn city center	2 109	2 082	centroid Hausdorff	1 431 1 343	678 766	651 739

Fig. 12. Height differences of paired clusters in the *TU Delft Campus* sample territory.

- Their height was too small, and only a few pixels reached the 1.5 meter threshold in the DEM. Hence they were considered small clusters and removed.
- The distance between the detected cluster center and the reference coordinate of the tree was over 3 m. This usually occurred for larger trees as the manually recorded reference position is sometimes on the boundary of the tree crown (or even outside of it).

Table 3
Results of volume calculation for different epochs at the sample territories.

Sample territory	Epoch 1 volume	Epoch 2 volume	Difference
Single street	80 751 m^3	128 616 m^3	47 866 m^3
Amsterdam city center	607 223 m^3	716 067 m^3	108 845 m^3
TU Delft Campus	1 582 340 m^3	2 481 330 m^3	898 989 m^3
Delft city center	5 669 190 m^3	8 755 340 m^3	3 086 150 m^3
Tallinn city center	1 487 250 m^3	1 686 700 m^3	199 453 m^3

5.2. Comparison with other methods

We compared the results of our algorithm with other existing methods we previously discussed in Section 2 to assess the effectiveness of the proposed method. Table 4 summarizes the measured differences. We investigated the extraction, matching, commission, and omission rate defined as Eq. (6) and (9), in accordance with Yang et al. (2020).

$$\text{Extraction rate} = \frac{N_{ext}}{N_{ref}} \cdot 100\% \tag{6}$$

$$\text{Matching rate} = \frac{N_{match}}{N_{ref}} \cdot 100\% \tag{7}$$

Table 4
Comparison of results provided by previous tree segmentation methods and our algorithm.

ID	Method	Extraction rate	Matching rate	Commission rate	Omission rate
1	Local maximum + Filtering	51	45	9	59
2	Local maximum + Region growing	57	43	20	61
3	Local maximum + Multiscale CHM	101	46	61	57
4	Local maximum + Watershed	86	49	49	55
5	Segmentation + Clustering	139	53	95	51
6	Local maximum with 3 × 3 kernel	154	54	113	51
7	Local maximum with 5 × 5 kernel	52	41	16	63
8	Watershed + 3D spatial distribution	107.8	67.6	37.3	32.5
9	Our proposed algorithm	123.2	80.0	35.0	20.0

$$\text{Commission rate} = \frac{N_{com}}{N_{ext}} \cdot 100\% \quad (8)$$

$$\text{Omission rate} = \frac{N_{om}}{N_{ref}} \cdot 100\% \quad (9)$$

N_{ext}, N_{match}, N_{com}, N_{om}, and N_{ref} are the cardinalities of extracted, matching, false positive (commission), false negative (omission), and reference trees, respectively.

The experimental results of *Methods #1-#7*, i.e. local maximum + filtering (Monnet et al., 2010), local maximum + region growing (Dalponte et al., 2014), local maximum + multiscale CHM (Eysn et al., 2015), local maximum + watershed (Lindberg et al., 2014), segmentation + clustering (Sambugaro et al., 2013), local maximum with 3 × 3 kernel, and local maximum with 5 × 5 kernel (Eysn et al., 2012) were previously benchmarked by Eysn et al. (2015). *Method #8*, i.e. watershed + 3D spatial distribution (Yang et al., 2020) is a state-of-the-art method.

The experimental results show that our proposed algorithm significantly outperforms the previous methods regarding the matching rate and omission rate of tree segmentation, while still producing an acceptable extraction rate and commission rate, comparable to other state of the art methods. Considering the robustness in execution time, evaluation area and the independence of tree species, the proposed method has multiple advantages.

6. Conclusions

The goal of our research was to develop an automatized, robust algorithm for the change detection of trees based on LiDAR data in large urban areas. The algorithm works with preprocessed DEM files instead of raw point clouds in order to decrease computational complexity. Two cluster maps are produced of the DEMs through a processing pipeline. Afterwards, the corresponding clusters of each cluster map are paired, producing a new map. The paired cluster map is suitable for change detection analysis.

Two nationwide datasets covering the Netherlands and Estonia were selected for testing our method. The sample territories cover urban environments with plenty of vegetation.

The results and their validation have shown that the algorithm fulfills the initial expectations for detecting trees in an urban area. However, artificial objects (typically building facades) can distort the results with false positive cases , which increase the number of removed and new trees in the first place. Cluster construction and pairing has provided representative data of the quantifiable changes (i.e. tree presence, height, and volume).

Future work includes increasing the accuracy of tree segmentation and cluster pairing, primarily focusing on reducing false positive detections. From a performance point of view, the algorithm could be improved by parallel data processing in the algorithm steps. The concurrent processing of non-overlapping areas (e.g. AHN tiles) would enable large-scale execution of the algorithm in a HPC environment. Both pairing methods can be further accelerated by creating spatial indexes for the cluster maps, e.g. a *quadtree* or *R-tree*.

Computer code availability

The prototype implementation of the algorithm was carried out in standard C++11 as part of the *PointCloudTools* geospatial framework.

Source code is available at https://github.com/mcserep/PointCloudTools and released under the BSD-3 license. The project was tested to build and run on Windows 10 and Ubuntu Linux 18.04/20.04 LTS.

CRediT authorship contribution statement

Anett Fekete: Methodology, Investigation, Software, Writing – original draft, Visualization. **Mate Cserep:** Conceptualization, Supervision, Software, Validation, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Datasets used in Section 5 can be found at http://dx.doi.org/10.17632/9thyzzwd5d.2, an open-source online data repository hosted at Mendeley Data (Fekete and Cserép, 2021).

References

Antonarakis, A., Richards, K.S., Brasington, J., 2008. Object-based land cover classification using airborne LiDAR. Remote Sens. Environ. 112 (6), 2988–2998. http://dx.doi.org/10.1016/j.rse.2008.02.004.

Bellakaout, A., Cherkaoui, M., Ettarid, M., Touzani, A., 2016. Automatic 3D extraction of buildings, vegetation and roads from lidar data. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 41 (B3), 173–180. http://dx.doi.org/10.5194/isprsarchives-XLI-B3-173-2016.

Butkiewicz, T., Chang, R., Wartell, Z., Ribarsky, W., 2008. Visual analysis and semantic exploration of urban LIDAR change detection. Comput. Graph. Forum 27 (3), 903–910. http://dx.doi.org/10.1111/j.1467-8659.2008.01223.x.

Chen, Q., Baldocchi, D., Gong, P., Kelly, M., 2006. Isolating individual trees in a savanna woodland using small footprint lidar data. Photogramm. Eng. Remote Sens. 72 (8), 923–932. http://dx.doi.org/10.14358/PERS.72.8.923.

Dalponte, M., Frizzera, L., Gianelle, D., 2014. Estimation of forest attributes at single tree level using hyperspectral and ALS data. In: 2014 ForestSAT Conference.

dos Santos, R., Galo, M., Carrilho, A., Pessoa, G., de Oliveira, R., 2020. Automatic building change detection using multi-temporal airborne lidar data. In: 2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference. LAGIRS, IEEE, pp. 54–59. http://dx.doi.org/10.1109/LAGIRS48042.2020.9165628.

Efford, N., 2000. Morphological image processing. In: Digital Image Processing: A Practical Introduction using Java, first ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, pp. 271–297.

Eysn, L., Hollaus, M., Lindberg, E., Berger, F., Monnet, J.M., Dalponte, M., Kobal, M., Pellegrini, M., Lingua, E., Mongus, D., et al., 2015. A benchmark of lidar-based single tree detection methods using heterogeneous forest data from the alpine space. Forests 6 (5), 1721–1747.

Eysn, L., Hollaus, M., Schadauer, K., Pfeifer, N., 2012. Forest delineation based on airborne LIDAR data. Remote Sens. 4 (3), 762–783.

Fekete, A., Cserép, M., 2021. Tree segmentation and change detection of Large Urban Areas based on airborne LiDAR: Datasets and supplementary materials. http://dx.doi.org/10.17632/9thyzzwd5d.2, Mendeley Data, V2.

Gonzalez, R.C., Woods, R.E., 2006. Morphological image processing. In: Digital Image Processing, 3rd Edition Prentice-Hall, Inc., Upper Saddle River, NJ, USA, pp. 649–710.

Gurney, C.M., Townshend, J.R., et al., 1983. The use of contextual information in the classification of remotely sensed data. Photogramm. Eng. Remote Sens. 49 (1), 55–64.

Hamraz, H., Contreras, M.A., Zhang, J., 2017. A scalable approach for tree segmentation within small-footprint airborne lidar data. Comput. Geosci. 102, 139–147.

Hyyppä, J., Hyyppä, H., Leckie, D., Gougeon, F., Yu, X., Maltamo, M., 2008. Review of methods of small-footprint airborne laser scanning for extracting forest inventory data in boreal forests. Int. J. Remote Sens. 29 (5), 1339–1366.

Hyyppä, J., Kelle, O., Lehikoinen, M., Inkinen, M., 2001. A segmentation-based method to retrieve stem volume estimates from 3-D tree height models produced by laser scanners. IEEE Trans. Geosci. Remote Sens. 39 (5), 969–975. http://dx.doi.org/10.1109/36.921414.

Jakubowski, M.K., Li, W., Guo, Q., Kelly, M., 2013. Delineating individual trees from LiDAR data: A comparison of vector-and raster-based segmentation approaches. Remote Sens. 5 (9), 4163–4186.

Kaartinen, H., Hyyppä, J., Yu, X., Vastaranta, M., Hyyppä, H., Kukko, A., Holopainen, M., Heipke, C., Hirschmugl, M., Morsdorf, F., et al., 2012. An international comparison of individual tree detection and extraction using airborne laser scanning. Remote Sens. 4 (4), 950–974.

Kaasalainen, S., Krooks, A., Liski, J., Raumonen, P., Kaartinen, H., Kaasalainen, M., Puttonen, E., Anttila, K., Mäkipää, R., 2014. Change detection of tree biomass with terrestrial laser scanning and quantitative structure modelling. Remote Sens. 6 (5), 3906–3922. http://dx.doi.org/10.3390/rs6053906.

Lindberg, E., Eysn, L., Hollaus, M., Holmgren, J., Pfeifer, N., 2014. Delineation of tree crowns and tree species classification from full-waveform airborne laser scanning data using 3-D ellipsoidal clustering. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 7 (7), 3174–3181.

Lindenbergh, R., Pietrzyk, P., 2015. Change detection and deformation analysis using static and mobile laser scanning. Appl. Geomat. 7 (2), 65–74. http://dx.doi.org/10.1007/s12518-014-0151-y.

Liu, K., Ma, H., Zhang, L., Cai, Z., Ma, H., 2019. Strip adjustment of airborne LiDAR data in urban scenes using planar features by the minimum Hausdorff distance. Sensors 19 (23), 5131.

Meyer, V., Saatchi, S., Chave, J., Dalling, J., Bohlman, S., Fricker, G., Robinson, C., Neumann, M., Hubbell, S., 2013. Detecting tropical forest biomass dynamics from repeated airborne lidar measurements. Biogeosciences 10 (8), 5421–5438. http://dx.doi.org/10.5194/bg-10-5421-2013.

Monnet, J.M., Mermin, E., Chanussot, J., Berger, F., 2010. Tree top detection using local maxima filtering: A parameter sensitivity analysis. In: 10th International Conference on LiDAR Applications for Assessing Forest Ecosystems (Silvilaser 2010), pp. 9.

Mukherjee, S., Joshi, P.K., Mukherjee, S., Ghosh, A., Garg, R., Mukhopadhyay, A., 2013. Evaluation of vertical accuracy of open source digital elevation model (DEM). Int. J. Appl. Earth Obs. Geoinf. 21, 205–217.

PDOK, 2013. In: van der Zon, N. (Ed.), Kwaliteitsdocument AHN2. Technical Report 1.3 final, Dutch National Spatial Data Infrastructure, URL: http://www.ahn.nl/.

PDOK, 2015. Besteksvoorwaarden Inwinning Landsdekkende Dataset AHN2014-2019. Technical Report 2.0 final, Dutch National Spatial Data Infrastructure, URL: http://www.ahn.nl/.

Raumonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., Lewis, P., 2013. Fast automatic precision tree models from terrestrial laser scanner data. Remote Sens. 5 (2), 491–520. http://dx.doi.org/10.3390/rs5020491.

Rockafellar, R.T., Wets, R.J.B., 2009. Set convergence. In: Variational Analysis, Vol. 317. Springer Science & Business Media, pp. 108–147. http://dx.doi.org/10.1007/978-3-642-02431-3.

Sambugaro, M., Colpi, C., Marzano, R., Pellegrini, M., Pirotti, F., Lingua, E., et al., Utilizzo del telerilevamento per l'analisi della biodiversità strutturale: Il caso studio della riserva forestale di clöise (Asiago, VI). In: Proceedings of the 17th Conferenza Nazionale ASITA, Riva Del Garda, Italy, pp. 5–7.

San, B.T., Suzen, M., 2005. Digital elevation model (DEM) generation and accuracy assessment from ASTER stereo data. Int. J. Remote Sens. 26 (22), 5013–5027.

Shaker, A., Yan, W.Y., LaRocque, P.E., 2019. Automatic land-water classification using multispectral airborne LiDAR data for near-shore and river environments. ISPRS J. Photogramm. Remote Sens. 152, 94–108. http://dx.doi.org/10.1016/j.isprsjprs.2019.04.005.

Shepard, D., 1968. A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 23rd ACM National Conference. ACM '68, ACM, New York, NY, USA, pp. 517–524. http://dx.doi.org/10.1145/800186.810616, URL: http://doi.acm.org/10.1145/800186.810616.

Song, J.H., Han, S.H., Yu, K.Y., Kim, Y.I., 2002. Assessing the possibility of land-cover classification using lidar intensity data. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 34 (3/B), 259–262.

Suárez, J.C., Ontiveros, C., Smith, S., Snape, S., 2005. Use of airborne LiDAR and aerial photography in the estimation of individual tree heights in forestry. Comput. Geosci. 31 (2), 253–262.

Sun, Y., Huang, J., Ao, Z., Lao, D., Xin, Q., 2019. Deep learning approaches for the mapping of tree species diversity in a tropical wetland using airborne lidar and high-spatial-resolution remote sensing images. Forests 10 (11), 1047. http://dx.doi.org/10.3390/f10111047.

Swart, L., 2010. How the up-to-date height model of the netherlands (AHN) became a massive point data cloud. NCG KNAW 17, 17–32.

Taha, A.A., Hanbury, A., 2015. An efficient algorithm for calculating the exact Hausdorff distance. IEEE Trans. Pattern Anal. Mach. Intell. 37 (11), 2153–2163.

Vosselman, G., Maas, H.G. (Eds.), 2010. Airborne and Terrestrial Laser Scanning. CRC Press, United Kingdom.

Yang, J., Kang, Z., Cheng, S., Yang, Z., Akwensi, P.H., 2020. An individual tree segmentation method based on watershed algorithm and three-dimensional spatial distribution analysis from airborne LiDAR point clouds. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13, 1055–1067.

Yu, X., Hyyppä, J., Kukko, A., Maltamo, M., Kaartinen, H., 2006. Change detection techniques for canopy height growth measurements using airborne laser scanner data. Photogramm. Eng. Remote Sens. 72 (12), 1339–1348. http://dx.doi.org/10.14358/PERS.72.12.1339.

Zhang, D., He, F., Han, S., Zou, L., Wu, Y., Chen, Y., 2017. An efficient approach to directly compute the exact Hausdorff distance for 3D point sets. Integr. Comput.-Aided Eng. 24 (3), 261–277.

Zhang, W., Montgomery, D.R., 1994. Digital elevation model grid size, landscape representation, and hydrologic simulations. Water Resour. Res. 30 (4), 1019–1028.