https://doi.org/10.1016/j.isprsjprs.2018.11.020

Marker-free coregistration of UAV and backpack LiDAR point clouds in forested areas

Przemyslaw Polewski[a], Wei Yao[a], Lin Cao[b,*], Sha Gao[b]

[a]*Dept. of Land Surveying and Geo-Informatics, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*
[b]*Co-Innovation Center for Sustainable Forestry in Southern China, Nanjing Forestry University, 159 Longpan Road, Nanjing, Jiangsu, China 210037*

Abstract

Unmanned aerial vehicle Laser Scanning (ULS) and Backpack Laser Scanning (BLS) are two emerging mobile mapping technologies applicable for monitoring forested environments in unprecedented detail from complementary perspectives. Although ground-based backpack techniques provide detailed information about the forest understory and terrain, the measured point clouds based on SLAM techniques are stitched together gradually and normally expressed in a less-accurate arbitrary coordinate system. Conversely, ULS point clouds are acquired from above and usually georeferenced, yet the point density and penetrability near the ground may still suffer from dense overstory despite the low attitude operation. Coregistering the ground and aerial point clouds in the ULS coordinate system therefore provides a method for fusing understory and overstory information at single tree level without the time consuming procedure of applying ground control points. Since the ULS and BLS acquisition viewpoints differ greatly, standard coregistration methods requiring 3D point-level correspondences are likely to fail. This paper presents an object-level coregistration approach which instead operates on two sets of tree positions, with the goal of finding the optimal 3D transformation (consisting of rotation, translation and scaling) between the respective coordinate systems. The entire task is decomposed into separate problems of computing the common Z axis, estimating the scale, and 2D coregistration. In contrast to existing methods, our approach does not require additional information such as tree diameters or heights. We evaluated our method on real test plots involving diverse stem densities and tree species situated in forest farm of the eastern coastal region of Jiangsu, China. The tree positions for ground and aerial data were obtained respectively by cylinder fitting and tree segmentation. On 3 coniferous (dawn redwood) plots, 46-81% trees were matched with a distance below 50 cm, and mean position deviation of 27-36 cm. For 4 broadleaf (poplar) plots, no more than 50% trees were matched below a 1 m threshold and mean error of 54-67 cm, which can be attributed to the broadleaf trees' more irregular shape and lack of a well defined tree top. Moreover, we show that the introduction of scaling into the transform can increase the matched tree count by up to 20 percentage points and decrease the mean matched distance by up to 13% compared to a strictly rigid transform.

Keywords:
coregistration, unmanned aerial vehicle, backpack laser scanning, graph matching, precision forestry

1. Introduction

Airborne Laser Scanning (ALS) is an established remote sensing technique with many applications in forest monitoring tasks (Hyyppä et al., 2012). Although traditional ALS systems are mounted on piloted aircraft, recent advances in hardware have led to the emergence of unmanned aerial vehicle (UAV)-borne laser scanning (ULS). This new technique has already shown promise for several forest inventory applications (Dandois et al., 2015; Wallace et al., 2016; Sankey et al., 2017). Similar to its predecessor, ULS also possesses integrated global navigation satellite system (GNSS) hardware as well as an inertial measurement unit (IMU), which results in a georeferenced point cloud. This feature is useful for directly relating the acquired 3D data to actual geographic locations. On the other hand, despite the lower flying altitudes compared to ALS, in case of ULS the laser beam is still subject to attenuation when interacting with a dense tree canopy, resulting in insufficient point density near the ground. This may prove problematic for estimating some forest parameters of interest, including tree species diversity, vertical structure, biomass and the soils carrying capacity, which require detailed understory information (Korpela et al., 2012). To name a concrete example, the minimum point density near the ground required to successfully segment fallen dead trunks with a diameter at breast height (DBH) of about 30 cm is 20 points per m^2 (Polewski et al., 2015). If the target objects are smaller or exhibit finer geometric detail, an even higher point density is necessary. To alleviate this problem, techniques complementary to aerial view perspectives may be used. Backpack laser scanning (BLS) is an emerging technology which constitutes an ap-

*Corresponding author.
Email addresses:
przemyslaw.polewski@muvi-smart-tech.com (Przemyslaw Polewski), wei.hn.yao@polyu.edu.hk (Wei Yao), ginkgocao@gmail.com (Lin Cao), nfu_gaosha@126.com (Sha Gao)

pealing alternative to TLS in forested environments due to its relatively low cost, complementary performance and fast operational time. Note that the ground-based methods including BLS generally tend to provide the measured point cloud in a local coordinate system (CS), whereas in a common forest inventory scenario, a georeferenced CS is required (Kukko et al., 2017; Bauwens et al., 2016). In particular, BLS can be equipped with cheap embedded global positioning system (GPS) modules which are still insufficient due to their low precision and susceptibility to canopy attenuation in closed forested areas. Therefore, in order to transform the coordinate system, additional effort must be undertaken by providing artificial control points with a priori measured positions, or by using expensive external navigation devices (GPS, IMU). Moreover, for BLS based on the simultaneous localization and mapping (SLAM) technique it usually requires several identifiable feature objects in the scene, which are unfortunately scarcely available in forested areas. Consequently, one usually needs to set up artificial markers with known coordinates placed on the scanning route to provide additional support to strip adjustment/stitching and trajectory georeferencing ability (Rönnholm et al., 2016; Tang et al., 2015). In this case, the derived BLS point clouds could further suffer from scale variance and planimetric/vertical deviations. To enable accurate fusion with the georeferenced ULS point cloud, these errors need to be corrected through coregistration of the two datasets. Unfortunately, most generic methods for point cloud coregistration require correspondence on individual 3D point level to calculate a transformation between their respective coordinate systems (Theiler et al., 2014; Weinmann and Jutzi, 2015). Due to the strongly diverging acquisition perspectives, the scans of a forest plot obtained from BLS and ULS are not expected to overlap to a high degree. Also, the geometric characteristics of the acquired datasets are likely to be different. This work presents an alternative approach based on *semantic* overlap between the two datasets - i.e. the fact that they contain complementary information about the same objects (trees). We define a fully automated, marker-free method for coregistering BLS and ULS point clouds of forested areas, which utilizes relative tree positions to compute the optimal transformation (consisting of translation, rotation and scaling) between their CSs. The core idea is to decompose the coregistration problem into a series of independent, simpler tasks. First, the Z axis of the terrestrial dataset is aligned according to the dominant tree trunk direction, which yields two of three rotation angles. Next, the mutual distances between trees are used for constructing a similarity function to determine corresponding trees in the two datasets. High-probability correspondences are found by means of a graph matching procedure. Moreover, the graph matching score function is optimized with respect to the scale factor using univariate maximization methods, which leads to an approximation of the scale difference between the datasets. Finally, a combinatorial optimization metaheuristic is applied to find the subset of correspondences which leads to the optimal non-rigid transform (measured by matched tree count), consisting of the final rotation angle, 3D translation as well as a refinement of the scale factor. This method is an extension of our previous work (Polewski et al., 2016), which was originally introduced in the context of coregistering point clouds obtained by ALS as well as terrestrial photogrammetry. Aside from the application to new types of data, the methodological contributions of this paper are: (i) support for automatic scale calculation at the graph matching and transform calculation levels, (ii) generalization of the graph matching framework to handle combinations of similarity functions, and (iii) introduction of simulated annealing-based optimization for determining the optimal corresponding tree subset. We evaluate the performance of the proposed method through comprehensive experiments on 7 real plots from planted forests situated in the eastern coastal region of Jiangsu, China. Also, a comparison to the most relevant competing method is carried out. We investigate the influence of various parts of the CS transformation (scaling, 3D rotations) on the coregistration accuracy. Finally, we examine the sensitivity of our method to large scale variations.

The rest of this paper has the following structure: section 2 discusses related work and outlines the differences between our approach and competing methods. Section 3 is devoted to presenting the entire coregistration approach, including a method overview (see Fig. 1) and detailed descriptions of each processing stage. In section 4, the details of data acquisition, evaluation criteria and the conducted experiments are given, while the next section presents and discusses the experimental results. The principal conclusions are gathered in the final section.

2. Related work

Several studies regarding coregistration of point clouds obtained from differing perspectives have been reported in literature. Persad and Armenakis (2017) proposed an approach for the automatic coarse alignment of 3D point clouds acquired from various platforms. However, the method was based on 2D keypoint matching performed on height map images of the point clouds, which was designed primarily for scenes containing man-made objects. Some research effort has also been dedicated specifically to point clouds acquired in forested areas. St-Onge et al. (2015) consider semi-automatic fusion of aerial photogrammetric and ALS point clouds. Turning to the case of TLS/ALS coregistration, Hauglin et al. (2014) consider the case when an initial georeferenced position of the TLS scanner is available. They exhaustively check groups of 2 matched tree pairs with distance and tree size constraints to derive the optimal rotation and translation. This method requires estimates of tree height (from ALS) and DBH (from TLS) for approximating the tree size. In the work of Brede et al. (2017), the authors compared RIEGL VUX-1 ULS system with TLS for estimating forest canopy height and DBH under the studied forest conditions. They found out that ALS canopy height models were on average 11.5 cm

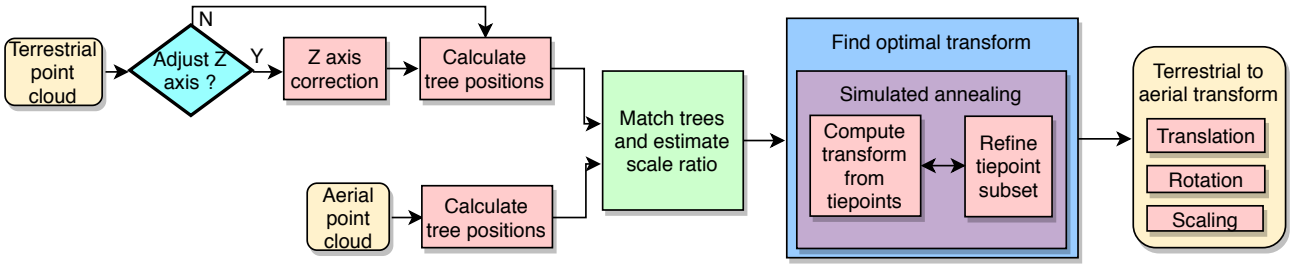


Figure 1: Overview of coregistration framework for a terrestrial and an aerial point cloud representing the same forested area.

higher, which means that TLS could not always detect the top of canopy. Moreover, 39 out of 58 extracted tree trunks could be found simultaneously in ALS and TLS for modeling stem diameter (DBH). It showed the complementary information provided by both data sources. Lindberg et al. (2012) also consider the TLS/ALS scenario, however it is assumed that coarse locations of the trees are available in both datasets (e.g. from GNSS), and the goal is to refine the positions by linking tree pairs with the smallest treetop distances. For this purpose, they create pairs of grayscale images corresponding to the stem positions and heights of the two tree sets. They assume that the Z axes of both coordinate systems are equal and focus on calculating the in-plane rotation angle and 2D translation through cross-correlation analysis of the grayscale images. Paris et al. (2017) utilize correlation images of the aerial and terrestrial canopy height models to calculate the optimal 2D rigid transform (2 translations, one rotation). However, this method may not be suitable for BLS data since in the general case, the scanned point cloud does not always contain canopy-level information. Moreover, this method requires initial parameter estimates derived from a coarse GNSS position. Finally, perhaps the most similar approach to ours is that of Kelbe et al. (2016), concerning the coregistration of two TLS point clouds of forested areas. Similar to our method, their strategy is based on analysis of relative tree positions in both datasets. These positions are derived from stems segmented in the 3D data. This method analyses pairs of tree position triplets to find the best rigid 3D transform aligning the two CSs. To limit the number of considered triplet pairs, a filtering step is applied based on the stem diameter (DBH). The remaining triplets are ranked based on their geometric similarity. One problematic aspect for applying this strategy to our case of BLS/ULS data is that no tree attributes common to both datasets are guaranteed to exist, except the tree locations themselves. In particular, neither tree height or DBH can be estimated from both BLS and ULS data, which renders the triplet filtering step ineffective and causes the geometric similarity ranking to boil down to an exhaustive search. This would be prohibitively time-consuming for even medium-sized datasets containing ca. 100 trees. In summary, to the best of our knowledge there is currently no approach applicable to our scenario having the following features: (i) not requiring any initial position/transformation parameter estimates, (ii) handling changes of scale, (iii) applicable to large-scale datasets, (iv) not requiring additional common tree attributes. Our work attempts to bridge this gap by providing a method fulfilling criteria (i)-(iv).

3. Methods

3.1. Overview

We present a general coregistration framework which covers various scenarios of terrestrial/aerial point cloud combinations. The framework distinguishes two types of input point clouds based on the availability of tree stem reflections. If stems are well preserved in the 3D data for most trees, the tree positions can be derived directly from stem modeling through 3D fitting of cylinders (Polewski et al., 2017) or lines (Amiri et al., 2017) depending on the geometric detail. We regard this step as an external subroutine to our framework, which accepts the point cloud as input and returns a list of 3D segments representing the fitted cylinder axes or stem lines. At this stage, depending on the input sensor type, a Z axis correction procedure may be applied (Sec. 3.4). For datasets originating from laser scanning (BLS,TLS), the coordinate Z axis is usually well-aligned with the Z axis of the georeferenced CS. However, in case of photogrammetric point clouds, the CS orientation is arbitrary and needs to be corrected. This is accomplished by aggregating the line segment orientations obtained from stem modeling and aligning the Z axis with the dominant direction. The tree positions are calculated as intersections of the stem lines with the digital terrain model (DTM). The second category of input point clouds does not contain sufficient tree stem information. In this case, single-tree segmentation is applied, and the tree positions are computed as highest points of each segmented tree cluster. Here, we assume that the local CS Z axis is aligned with the Z axis of the georeferenced CS, which holds for laser scanning systems (ALS, ULS). Once the tree positions are known and the Z axes are aligned, the remainder of the problem boils down to 2D coregistration. The challenging part of this task is finding pairs of corresponding trees in the two datasets. The sets of trees captured in each of the two datasets need not be equal, we only assume that a sufficient number (at least 3) of trees is present in both. In Section 3.5, we describe the part of the framework which is responsible for constructing a similarity measure between tree pairs based on the distributions

of inter-tree distances. The pairwise similarities induce a weighted bipartite graph, whose maximum-weight matching represents the set of high-probability tree correspondences (Sec. 3.6). Note that since the similarity measure is based on tree distances, it is not invariant under scaling. Therefore, the graph matching score function is parameterized by a rescaling factor applied to one of the point clouds. The extremum of this function constitutes an approximation of the true scale ratio between the coregistered point clouds. In the final part of the method (Sec.3.7), we seek a subset of the calculated correspondences which will yield the optimal coordinate transformation between the point clouds' respective CSs. This step is necessary because the graph matching procedure provides a match for every tree (from the smaller of the two datasets), but since only a subset of the trees is common to both point clouds, not all calculated correspondences are correct. The subset of optimal correspondences is found by applying the well-known simulated annealing (SA) metaheuristic in order to make better use of computational resources compared to exhaustive or random search. The typical workflow for coregistering a pair of terrestrial/aerial point clouds is shown in Fig. 1.

3.2. Notation

In the description of the proposed method, we utilize the following notation: T_A, T_B denote respectively the first and second set of trees to be coregistered, while $t_i^A \in T_A, t_j^B \in T_B$ indicate individual trees belonging to these sets. Datasets A and B contain respectively n_A, n_B trees. The 3D positions associated with the trees from each dataset are represented by P_A, P_B, where p_i^A is the position of the $i-$th tree from set A.

3.3. Tree stem modeling with cylinders

To fit cylinders to tree trunk points, we first perform DTM filtering of the point cloud, retaining a vertical slice of the data in a height interval $[h^c_{min}; h^c_{max}]$, such that the points lying within this interval are expected to belong mostly to tree stems. The specific values of h^c_{min}, h^c_{max} depend on the target area, in particular the height of the herbal/ground vegetation layer as well as the average tree heights. We then perform connected component segmentation on the filtered points, and retain components having at least n_{cc} members, resulting in candidate stem clusters. If the ground vegetation is low, usually a good separation of stem candidates can be achieved (see differently colored point clusters in Fig. 2), and the stem positions can be approximated by the centroids of the clusters' member points. In more complicated scenarios, dense and high shrub/herbal vegetation may cause the connected component segmentation to yield clusters which contain more than one stem. In this case, a cylinder fitting method which simultaneously locates multiple cylinders in its input data should be applied independently to each connected component, i. e. the method of Polewski et al. (2017). The stem positions and orientations can then be derived from the fitted cylinder axes.

3.4. Z axis adjustment

To find the most probable Z axis based on tree stem orientations, we make use of the fact that trees usually grow in a direction antiparallel to the vector of gravity, a phenomenon also known as gravitropism. After obtaining the list of stem lines from cylinder/line fitting, we gather the normalized line direction vectors v_i and compute their 3D spatial median $\hat{v}$:

$$\hat{v} = \arg\min_{v \in \mathbf{R}^3} \sum_i \|v - v_i\|_2 \tag{1}$$

The spatial, or geometric, median is one of several possible generalizations of the univariate median to multiple dimensions. It is defined as the vector having the minimal sum of distances to the set of input vectors. Here we utilize Euclidean distance as the metric. We chose the multivariate median over a mean as a measure of central tendency due to its robustness in the presence of outliers, which in our case may represent false positives from the stem modeling process or inaccurately fitted cylinders/stem lines. The algorithm of Kärkkäinen and Äyrämö (2005) is applied to ensure efficient computation of $\hat{v}$. The original point cloud is then rotated to align the vector $\hat{v}$ with the coordinate Z axis. A sample scene depicting the individual tree axes and the aggregate Z axis is shown in Fig. 2.

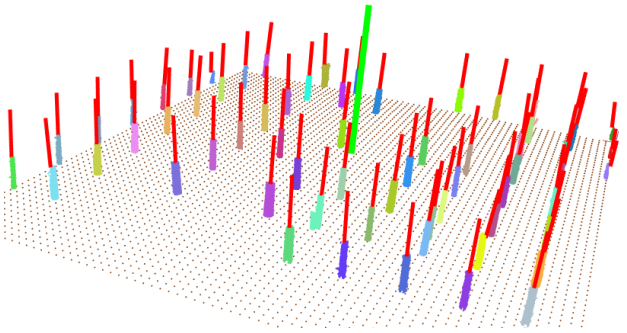


Figure 2: Aggregation of Z axis votes of individual trees (red vectors) using spatial median results in combined Z axis estimate (green vector). Point colors represent assigned clusters from connected component segmentation, shown over the DTM (brown surface). Length of aggregate axis (green vector) is 10 meters.

3.5. Similarity function

The proposed similarity function $S(t_i^A, t_j^B)$ is defined on a pair of trees belonging respectively to the first and second dataset. Each tree t_i is associated with a set of M abstract features $F_i^1, \ldots, F_i^M$. The aggregate similarity function is a convex combination of partial similarities $S^m(F_i^m, F_j^m)$, which are normalized on the interval $[0; 1]$ and weighted by the non-negative coefficients γ_m:

$$S(t_i^A, t_j^B) = \sum_{m=1}^{M} \gamma_m S^m(F_i^m, F_j^m), \sum_{k=1}^{M} \gamma_k = 1 \tag{2}$$

The relative weights γ_k should be set based on how well an attribute can discriminate between different trees in a dataset. Two types of features may be distinguished based

on whether they characterize 'absolute' or 'relative' traits. By 'absolute', we are referring to unary properties of single trees which have meaning regardless of other trees in the same dataset, e.g. tree height, stem diameter, species etc. Their advantage is that their similarity is easy to model through a univariate monotonically decreasing function, e.g. a Gaussian kernel with an appropriate bandwidth σ_m:

$$S^m(F_i^m, F_j^m) = \exp(-|F_i^m - F_j^m|^2/(2\sigma_m^2)) \quad (3)$$

It should be noted that matching features may not be available in the coregistered datasets, due to differing data acquisition perspectives. For example, it is hard to estimate the tree height from terrestrial point clouds, and the same is true for DBH estimation from aerial data. Even when common features are available, they may not be distinctive enough to provide any useful basis for a similarity function. A prime example of this would be a dataset covering a monoculture planted forest, where all trees have approximately the same height and diameter at any point in time. For these reasons, we augment the absolute features with the 'relative' category, where the feature is based only on mutual distances between trees in the dataset, but does not have any meaningful interpretation outside the local CS.

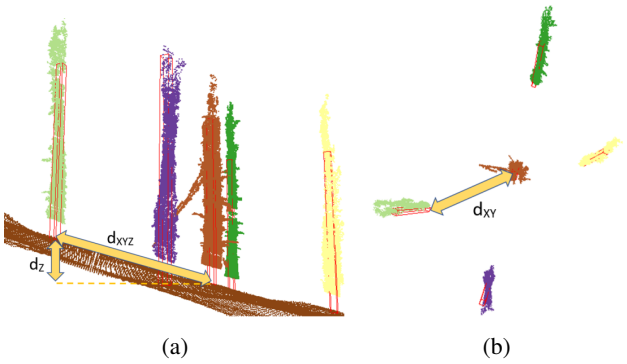
Relative features

We consider vector-valued features $F^k = [f_1^k, \ldots, f_n^k]$ which can be interpreted as descriptors of a tree's distance to other trees within the dataset. Specifically, recalling the notation from Sec. 3.2, the general form of the proposed relative features is:

$$F^k(t_i^X) = [d_k(p_i^X, p^X_{\pi^X(1)}), \ldots, d_k(p_i^X, p^X_{\pi^X(n_X)})] \quad (4)$$

In the above, X is a placeholder for either 'A' or 'B', while d_k represents a distance metric on the pair of 3D points, and $\pi^X(\cdot)$ is the permutation of the tree positions which sorts F^k according to ascending distance values. To ease the notational burden, we will write $F^k_{X,i} \equiv F^k(t_i^X)$. Some natural choices for d_k include planimetric Euclidean distance, 3D Euclidean distance, or Z deviation (see Fig. 3). Several features of the form given in Eq. 4 can be combined within the framework of Eq. 2, with different relative weights γ_i depending on the plot characteristics.

Note that the relative features are, in general, vectors of different length, since the numbers of tree positions derived from the first and second dataset may differ. Moreover, only a subset of trees will actually be common to both datasets, while the remainder will belong to only one of them. This calls for a special similarity measure which can handle such properties of the relative descriptors. We utilize an approach from computational biology for the optimal matching of DNA sequences, developed by Needleman and Wunsch (1970). The general algorithm computes the highest-similarity alignment between two sequences of elements, equipped with a pairwise similarity score $s(\cdot,\cdot)$. In our case, the aligned sequences are the descriptors $F^k_{A,i}, F^k_{B,j}$, and the elements are the distances d_k. The aggregate similarity $S^k_{i,j}$ between these two feature



(a) (b)

Figure 3: Tree positions marked with red boxes on DTM with different types of distances displayed. (a) side view - Z deviation and full Euclidean XYZ distance, (b) top view - XY (planimetric) distance.

descriptors, belonging to the i-th tree in the first dataset and the j-th tree in the second, is defined recursively as $S^k_{i,j} = T(n_A, n_B)$:

$$T(m,n) = \max \begin{cases} T(m-1,n-1) + s(F^k_{A,i}[m], F^k_{B,j}[n]) \\ T(m-1,n) + p_g \\ T(m,n-1) + p_g \end{cases} \quad (5)$$

The term $F^k_{A,i}[m]$ is the m-th element in the feature vector F_k describing the i-th tree in the first dataset. The function $s(\cdot,\cdot)$ represents a partial similarity score between two elements of the feature vector. Its purpose is to quantify how likely it is that the two compared values represent the same relative distance, taking into account measurement uncertainty and other errors. The recursive formula in Eq.5 represents the three options that the algorithm may choose from at every step of the recursion. To illustrate these choices, consider the position (m,n) of the aligned sequence pair. Assuming that the solution for the smaller subsequences $T(m-1,n-1), T(m-1,n), T(m,n-1)$ is known (through recursion), there are three ways that the optimal solution for the shorter subsequence may be combined with the current elements $F^k_{A,i}[m]$ and $F^k_{B,j}[n]$. First, these two elements may be considered as matched, in which case the new solution is formed by appending the current elements to the solution of $T(m-1,n-1)$. The second option is to assign a 'gap' to element m in the first sequence, which effectively means that this element does not have a similar enough counterpart in the second sequence. This may happen if the sets of tree positions detected in the two data sources are not equal, for example if additional trees were discovered from the terrestrial perspective, which could not be acquired from the aerial perspective due to occlusions. Finally, the third option is to assign a 'gap' to element n in the second sequence. The algorithm selects the locally optimal move (resulting in the maximum aggregate similarity for $T(m,n)$) at each step of the recursion. Based on the principle of dynamic programming and due to the structure of this combinatorial problem, it can be proved that the locally optimal moves lead to a globally optimal result for the entire sequences. In our setting, the

gap penalty p_g is set to zero, and the boundary conditions for the recursion are $T(0,0) = 0$. We propose to model the similarity score $s(\cdot,\cdot)$ using a Gaussian kernel of the form given in Eq. 3, truncated to 0 at $3\sigma_k$ to avoid wasteful computation of near-zero values. A suitable value for σ_k can be derived from the mean nearest-neighbor value of d_k over all pairs of trees in each dataset.

3.6. Correspondence matching and scale estimation

A complete weighted bipartite graph $G = (V, E)$ is constructed, such that the set of vertices V comprises two disjoint subsets $V_A, V_B \subset V, V_A \cap V_B = \emptyset$ corresponding respectively to trees detected in the first and second dataset. To ensure that the vertex counts of V_A, V_B are equal, dummy nodes are added to the smaller set. The edge set E consists of all pairs $(u, v)|u \in V_A, v \in V_B$. The weight w_{ij} associated with the edge connecting t_i^A with t_j^B is their similarity value $S(t_i^A, t_j^B)$ (Eq. 2), whereas the weights of any edges containing the dummy nodes are set to zero. Figure 4 visualizes the graph construction process.

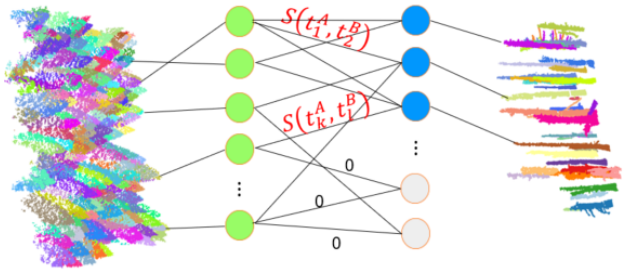


Figure 4: Bipartite graph on datasets A,B with weights obtained from pairwise similarity function S. White nodes represent auxiliary vertices to remove imbalance in tree counts between A and B.

We find the maximum weight matching $M \subseteq E$ in G with respect to the weights (w_{ij}) using the Kuhn-Munkres algorithm, such that the total similarity S_T of the matched vertices is maximized:

$$S_T = \sum_{(i,j)\in M} w_{ij} = \sum_{(i,j)\in M} S(t_i^A, t_j^B) \tag{6}$$

Scale search

Since the relative features describe distances between the tree positions, descriptors constructed from different datasets in general exhibit a high similarity only when the dataset scales are nearly equal. To account for potentially large scale differences, we model the total similarity (Eq. 6) as a continuous function of the second dataset's scale k:

$$S_T(k) = \sum_{(i,j)\in M} S(t_i^A, k \cdot t_j^B) \tag{7}$$

In the above, the term $k \cdot t_j^B$ refers to rescaling all the spatial attributes of tree t_j^B (e.g. height, diameter, position) by a factor of k. By maximizing S_T w.r.t. k, we obtain a rough estimate of dataset B's relative scale versus dataset A. Moreover, based on empirical evidence it seems that S_T has a single, dominant global optimum on a broad range of scale factors (e.g. 0.1 to 10, see Fig. 5), making it amenable to unimodal, univariate optimization methods. Our method of choice is golden section search, since it is derivative-free and in practice it is able to converge to the optimal scale factor in a limited number of iterations. Our approach of explicitly minimizing S_T is a relatively computationally cheap method of obtaining a relative scale estimate compared to a brute-force grid search for k over a potentially broad interval.

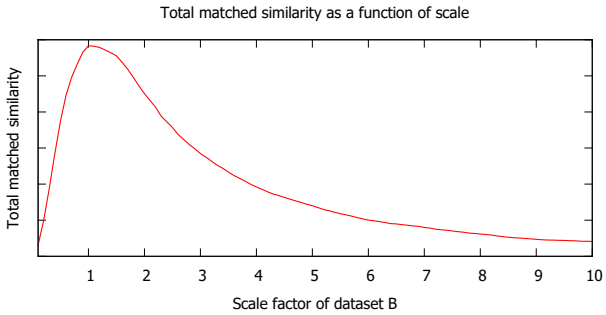


Figure 5: A typical profile of the total matched tree similarity when one of the datasets has its tree positions rescaled. Usually, a single strong local maximum can be observed near the relative scale of the two datasets.

3.7. Optimal coordinate transform search

The matching M^* which yields the highest total similarity is used as a basis for finding a subset $T_S \subset T_A \times T_B$ of tiepoints (or 'tie-trees') that will define the coordinate transformation between the datasets' CSs. First, we describe the adopted transformation type and a way of calculating it from sample points. Then, we describe the process of finding the optimal tiepoint subset.

3.7.1. Coordinate transform

We wish to find a coordinate transform T_{AB} from the CS of dataset A to that of dataset B. Since we assume that the Z axes have already been equalized, T_{AB} consists of a single 2D rotation by angle θ, a scaling factor k_s, and a 3D translation vector t_{xyz}:

$$\begin{pmatrix} x^B \\ y^B \\ z^B \end{pmatrix} = k_s \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^A \\ y^A \\ z^A \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{8}$$

Assuming that a sequence of n_c corresponding points from T_A, T_B is known, four of the five parameters of interest (k_s, θ, t_x, t_y) may be recovered by applying the method of Umeyama (1991) in two dimensions. Let μ^A, μ^B denote the planimetric centroids of the n_c tree positions respectively from T_A and T_B. Also, let Σ^{AB} indicate the (2D) cross-covariance matrix between the corresponding points, and let S be the identity matrix, whose last diagonal element is set to -1 if and only if $\det(\Sigma^{AB}) < 0$. Consider the singular value decomposition of Σ^{AB} as $\Sigma^{AB} = UDV^T$, where D is the diagonal matrix of singular values, and U, V represent matrices with columns corresponding respectively to the left and right singular vectors of Σ^{AB}.

Umeyama's main result connects the singular value decomposition of Σ^{AB} to the transform parameters yielding the minimal average least-squares distance on the matched point pairs after transformation:

$$\begin{aligned} R_\theta &= USV^T \\ t_{xy} &= \mu^B - k_s R_\theta \mu^A \\ \sigma_A &= \frac{1}{n_c}\sum_i \|p_i^A - \mu_A\|^2 \\ k_s &= \frac{1}{\sigma_A}\,\mathrm{tr}(DS) \end{aligned} \tag{9}$$

In the above, R_θ indicates the standard 2D rotation matrix by angle θ, from which the angle can be recovered. The only missing parameter, the vertical translation t_z, is found as the median signed Z deviation over the n_c corresponding tree positions.

3.7.2. Optimal tiepoint selection

Our goal is to find a coordinate transform which 'optimally' aligns the datasets T_A, T_B. This optimality may be defined in different ways. We propose to use a modified random sample consensus-style criterion which is a linear combination of two terms n_m and d_{avg}. The first term n_m denotes the number of matched tree pairs whose distance (after transformation into the common CS) does not exceed a threshold value d_{thr}. The second term is the average d_{avg} distance between the n_m matched pairs. They are combined with the proportion coefficient β to form the scoring criterion V for a transform T:

$$V(T) = n_m(T) + \beta d_{avg}(T) \tag{10}$$

Equipped with this criterion, we wish to solve a combinatorial problem of choosing n_c tiepoint pairs out of the matching M^*, such that the transform obtained by applying Umeyama's method (Eq. 9) maximizes Eq. 10 on the datasets T_A, T_B. It should be noted that by decoupling the correspondence calculation step from the tiepoint subset selection, the resulting combinatorial problem's complexity has been greatly reduced compared to the simultaneous variant. Using the big O notation, the latter has a solution space of size $O(\binom{n_A}{n_c}\binom{n_B}{n_c}n_c!)$, whereas in case of our subproblem it decreases to $O(\binom{\min(n_A,n_B)}{n_c})$. We follow the general scheme of the simulated annealing metaheuristic to maximize (10). The outline of the procedure is given in Algorithm 1. The function *optimalTransform*, called in lines 3 and 8 of the top-level function *OptimalTiepointsSA* (Algorithm 1), implements Umeyama's procedure (Eq. 9). Our algorithm uses standard meta-parameters concerning the cooling scheme: t_0, t_f indicate the initial and final temperatures, while ρ is the cooling factor. At each temperature level, n_{iter} iterations are executed. The function *neighbor* (called in line 7 of *OptimalTiepointsSA*) produces a neighboring solution out of an existing matching M^c by randomly swapping out *moveOrder* elements of M^c and replacing them with elements from M^* which were previously not included. The value of *moveOrder* is typically 1 or 2.

Algorithm 1 Optimal tiepoint search - simulated annealing

1: **function** OptimalTiepointsSA(T_A,T_B,M^*,n_c)
2: randomly initialize $M^c \subset M^*, |M^C| = n_c$
3: $T^c \leftarrow T^{opt} \leftarrow optimalTransform(T_A, T_B, M^c)$
4: $t \leftarrow t_0$
5: **while** $t \geq t_f$ **do**
6: **for** k=1..n_{iter} **do**
7: $M^{new} \leftarrow neighbor(M^c, \text{moveOrder})$
8: $T^{new} \leftarrow optimalTransform(T_A, T_B, M^{new})$
9: $\Delta = V(T^c) - V(T^{new})$
10: **if** $\Delta < 0$ or $\exp(-\Delta/t) \geq rand(0,1)$ **then**
11: $M^c \leftarrow M^{new}$
12: $T^c \leftarrow T^{new}$
13: **if** $V(T^c) > V(T^{opt})$ **then**
14: $T^{opt} \leftarrow T^c$
15: $t \leftarrow t \cdot \rho$
16: **return** T^{opt}

To ensure a quick execution of each move, we propose an incremental update scheme for the Umeyama algorithm, avoiding the most computationally expensive part of re-calculating the entire covariance matrix, even though only 1-2 data points have changed. This is based on the observation that both the means and σ_A as well as the cross-covariance matrix Σ^{AB} may be updated on-line by maintaining certain additional accumulation variables. Let $\mu^{A\prime}, \mu^{B\prime}, \Sigma^{AB\prime}, \sigma'_A$ denote the updated parameter values after tree positions p^A_{old}, p^B_{old} have been removed and replaced with p^A_{new}, p^B_{new}. First, it is straightforward to update the two means $\mu^{A\prime}, \mu^{B\prime}$ by subtracting the (appropriately scaled) removed point and adding the newly introduced vector:

$$\mu^{X\prime} \leftarrow \mu^X + \frac{p^X_{new} - p^X_{old}}{n_c} \tag{11}$$

Regarding σ'_A, we note that σ_A (Eq. 9) can be expressed as $\sigma_A = \frac{1}{n_c}\sum_{i=1}^{n_c}(p_i^A)^T p_i^A - (\mu^A)^T\mu^A$. It is then sufficient to maintain an accumulator variable $S^{AA} = \sum_{i=1}^{n_c}(p_i^A)^T p_i^A$, which can be updated on-line in constant time. The formula for updating σ'_A is given by:

$$\begin{aligned} S^{AA\prime} &\leftarrow S^{AA} + (p^A_{new})^T p^A_{new} - (p^A_{old})^T p^A_{old} \\ \sigma'_A &\leftarrow S^{AA\prime}/n_c - (\mu^{A\prime})^T\mu^{A\prime} \end{aligned} \tag{12}$$

Finally, a similar trick may be applied to $\Sigma^{AB\prime}$, using another accumulator variable $S^{AB} = \sum_{i=1}^{n_c} p_i^A (p_i^B)^T$, which leads to the on-line update formula:

$$\begin{aligned} S^{AB\prime} &\leftarrow S^{AB} + p^A_{new}(p^B_{new})^T - p^A_{old}(p^B_{old})^T \\ \Sigma^{AB\prime} &\leftarrow S^{AB\prime}/n_c - \mu^{A\prime}(\mu^{B\prime})^T \end{aligned} \tag{13}$$

The search for corresponding trees necessary to perform the evaluation in $V(\cdot)$ (Eq. 10) is accelerated by using a grid-like spatial index on the tree positions of T_B, which reduces the time complexity of evaluation to $O(n_A)$. The dominant component of evaluating V in terms of computing time is applying the transform to the tree positions in T_A.

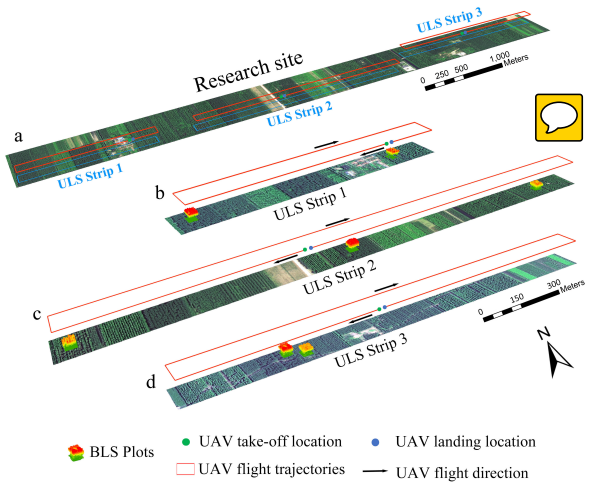


Figure 6: The orthophoto of the research site, ULS strips and BLS plots. (a) overview of the research site and the three ULS strips (swath width=100 m); (b) ULS strip one with flight trajectories and the included two BLS plots; (c) ULS strip two with flight trajectories and the included three BLS plots; (d) ULS strip three with flight trajectories and the included two BLS plots.

3.7.3. Fine-grained scale search

To account for the fact that the scale estimate obtained from the graph matching score (Sec. 3.6) is a rough estimate and may differ from the true scale by up to 15-20%, we execute the processing pipeline on a small grid of scale factors around the estimate. Typically, we found that the scale estimate is biased towards higher values, therefore we multiply it by factors of 0.8-1.0 with a step of Δ_s, rescale one of the datasets and run the graph matching and tiepoint search with the altered scales. We retain the result which obtains the highest score V (Eq. 10).

4. Experiments

4.1. Material

4.1.1. Research site and sample plots

The research site is located in the Yellow Sea coastal national forest park of southeast China (120° 49'9.80"E, 32° 52'23.27"N). The total area of the forest park is approximately 2240 ha (Cao et al., 2016). Approximately 80% of the forest park is covered by planted forests dominated by poplar (*Populus deltoids*), dawn redwood (*Metasequoia glyptostroboides*) and ginkgo (*Ginkgo biloba*). On the north side of the forest park, an array of forest stands (covering an area of approximately 6000 m × 600 m, elongated in east-west direction) was selected as the research site for this study. Within the study site, there are pure poplar and dawn redwood plantation stands grown at different developmental stages (Fig. 6). Seven square plots (30 m × 30 m) were established within the ULS strips. These plots were chosen according to a GIS-based forest inventory data (2017) covering a range of tree species, stand ages and forest structures (e.g. mean DBH and stem density) - see Table 1 for overview of plot properties. Note that the dawn redwood and poplar are respectively coniferous and broadleaf species, resulting in a partition of 3 vs. 4 plots of these two categories.

Plot	Tree species	Stand age [yr]	Mean DBH [cm]	Stem density [1/ha]
1	dawn redwood	39	29.1	422
2	dawn redwood	43	31.6	689
3	dawn redwood	34	27.1	744
4	poplar	21	33.3	242
5	poplar	22	34.6	411
6	poplar	23	38.2	344
7	poplar	13	24.7	611

Table 1: Summary of forest parameters measured within the field plots.

4.1.2. ULS data

A GreenValley (GreenValley, 2018) UAV LiDAR System was implemented in this study for acquiring strips of LiDAR point clouds of the forest stands. An eight-rotor UAV was used as the remote sensing platform in the system (Fig. 7a). The components of a LiDAR scanner (Velodyne Puck VLP-16), an IMU (Novatel SPAN-MEMS-IMU), a dual-frequency GPS (Novatel) receiver are mounted on the UAV platform. The UAV has a controlling system and an auto-pilot system, which allows the UAV to fly according to the pre-designed trajectories in approximately 25 minutes maximum flight time (Guo et al., 2017). The ground control station includes two laptops (using GreenValley LiAcquire software to control the UAV and monitor the flight and scanner parameters) and a Novatel GNSS ground base station. Two dual frequency antennae on the UAV provided real-time kinematic differencing positioning observations. The IMU and GPS data were further post-processed using Novatel Inertial Explorer software and the Novatel GNSS reference station information to improve the accuracy of UAV flight trajectories, and to calculate the final LiDAR point clouds in georeferenced mode (Guo et al., 2017). Within the research site, three strips of ULS point clouds (placed from west to east) were acquired during July 24-26th, 2017 under leaf-on conditions. The swath widths of the strips were approximately 100 m, and the lengths of the strips used in this study were approximately 1000 m, 2000 m and 1500 m respectively (Fig. 6). The UAV flight parameters and scanner properties can be found in Table 2.

4.1.3. BLS data

In this study, the GreenValley LiBackpack LiDAR system included the main components of a LiDAR scanner (Velodyne Puck VLP-16), a Position Orientation System

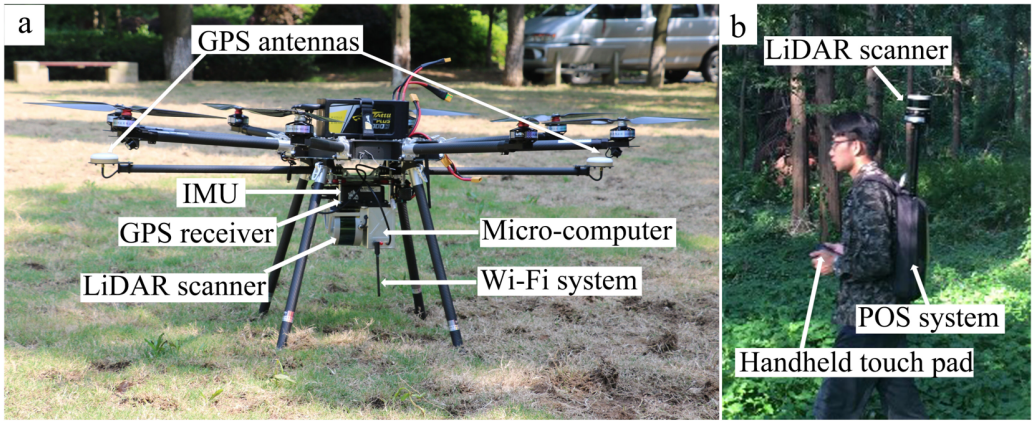


Figure 7: The UAV LiDAR system and Backpack LiDAR system used in this study. (a) the main components mounted on the UAV platform; (b) the main components of the BLS

Parameter name	Value
Flight altitude (m)	86
Flight velocity (m s^{-1})	3.6
Side overlap (%)	100
Wavelength (nm)	903
Beam divergence (mrad)	3
Footprint (cm)	26
Pulse repetition frequency (kHz)	21.7
Scan frequency (line s^{-1})	16
Maximum scan angle (°)	30
Point density (pts m^{-2})	84

Table 2: Flight parameters and scanner properties of the UAV LiDAR system data acquisition

(POS) and a handheld touch pad (Fig. 7b). The BLS point cloud registration was performed by the integration of LiDAR scans and POS information under a SLAM process. Within the strips, seven 30 m × 30 m square sample plots were scanned by the BLS. An "S" shape strip path was designed to collect the BLS point clouds for the plots (Fig. 8a). The characteristics of the BLS scanning system are given in Table 3.

4.2. Single tree segmentation

The GreenValley LiForest (LiForest, 2018) software was used to pre-process the ULS and BLS acquired point clouds. The ULS point clouds were de-noised by setting the filtering height-threshold and interactively removing the outlier points. The nearby strips of ULS data were co-registered by finding the matching points. A 0.5 m digital terrain model was created by first filtering the above-ground returns using an improved triangulated irregular network (TIN) densification filtering algorithm (Zhao et al., 2016), and then the remaining ground points were interpolated by the Inverse Distance Weighted

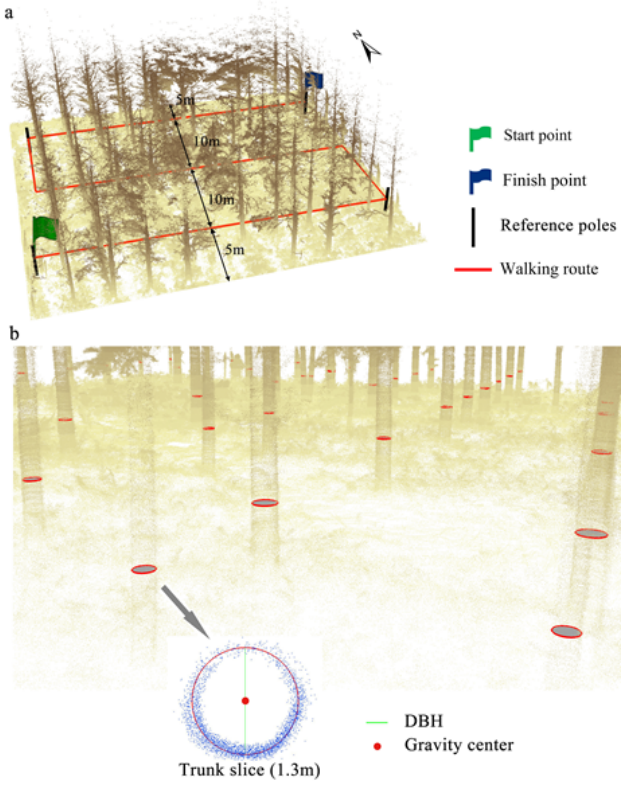


Figure 8: The path of BLS data acquisition within one sample plot (size: 30 by 30 m), which is shown as the co-registered point clouds using a SLAM process by integrating LiDAR scans and POS information (a), and the demonstrated results of a trunk slice at 1.3 m, the extracted gravity center from trunk slice and the calculated DBH (b).

(IDW) algorithm. The point cloud height (Z) was normalized by subtracting the ground surface height, and then the plots were extracted by the GPS-measured plot-corner coordinates. Two different individual tree detection (ITD) approaches were applied to both ULS and BLS point clouds respectively. For the ULS data, a point cloud based ITD approach proposed by (Li et al., 2012), which judges the relative spacing between trees by top-to-bottom region growing, was implemented to delineate the tree crowns and detect the tree tops. In contrast to the ULS data, where

Ground speed (m s^{-1})	1
Maximum range (m)	100
Horizontal Field-of-View (°)	360
Vertical Field-of-view (°)	15
Beam divergence (mrad)	3
Scan frequency (line s^{-1})	16
Total Weight (kg)	6
Point density (pts m^{-2})	5136
Min. footprint size (cm)	1.5
Max. footprint size (cm)	12.7

Table 3: Data acquisition parameters and scanner properties of BLS within the plots

mainly densely intertwined forest canopies were acquired, for BLS scanning the tree-trunks are usually isolated and therefore relatively easier to detect. In this study, an algorithm of Density-based spatial clustering of applications with noise (DBSCAN) was implemented for tree-trunk detection (Wu et al., 2013). A slice (vertical distance of 10 cm) at 1.3 m height was extracted for each plot and used as input for the DBSCAN algorithm (Tao et al., 2015). After the tree-trunk segmentation, the gravity centers of all points for each slice of the trunk were calculated, and the radius was determined as the average distance between the gravity center and each trunk point. The DBH was calculated by doubling the radius at the height of 1.3 m. It is possible that some trees were not fully scanned from all directions, leading to high variations in sample density on the trunk surface. This in turn might have distorted the gravity center calculation. Assuming that each trunk was scanned at least in an arc of 180°, based on simple simulations we estimate the error in the center computations at 50-60% of the trunk's radius, which translates to absolute values of up to 12 cm average deviation (see average DBH in Table 1). Table 4 summarizes the number of trees segmented in both ULS and BLS point clouds for each plot.

Plot	#ULS trees	#BLS trees	Type
1	65	36	coniferous
2	107	59	coniferous
3	97	64	coniferous
4	22	18	broadleaf
5	53	33	broadleaf
6	37	27	broadleaf
7	107	52	broadleaf

Table 4: Number of segmented trees in ULS and BLS data for all test plots

4.3. Evaluation criterion

The evaluation criterion for coregistration quality was the criterion V (Eq. 10) with the balance coefficient β set to 1. For coniferous-dominated plots, we assumed a matched distance threshold d_{thr} of 0.5 m, whereas for plots with an advantage of broadleaf trees, this was increased to 1.0 m due to the less precise tree positioning. This means that the most significant factor for evaluation was the number of matched trees with distance under d_{thr}, and in case of equality the average matched distance was decisive. For each experiment, we report the percentage (relative to smaller dataset) of matched trees having a distance below d_{thr}, as well as the average matched tree distance.

4.4. Coarse co-registration

To provide a semi-automatic reference for our automatic method, we performed coarse co-registration for both data sources was performed in the sense of georeferencing ULS and BLS point clouds respectively with the help of onboard GPS-IMU systems and GPS-measured controlled points set up in the forest stand. Particularly for BLS data of each plot, four reference poles were placed at the beginning and end of two outer strip paths (see Fig. 8a). The coordinates of reference poles (height of 2 m) were measured using the Real-time kinematic (RTK) positioning system (positioning accuracy not exceeding 5cm), which used the Trimble NetR9 GNSS as the base station and a Trimble R4 GNSS receiver as RTK rover. The collected BLS point clouds were transformed from relative coordinates to the geo-referenced coordinates by the GreenValley LiForest software using the coordinates (X,Y,Z) of reference pole tops. The accuracy of the coarse coregistration with respect to the evaluation criteria from the previous section are shown in Table 5. It should be noted that such a coarse co-registration is not at all required as a pre-processing step for our method (it does not influence the results), and it is included in the experiment only for comparison purposes.

Plot	# matched	% matched	Avg. distance [cm]
1	2	6	42
2	5	8	31
3	26	41	28
4	3	17	79
5	7	21	69
6	1	4	33
7	13	25	73

Table 5: Coarse coregistration results. Number and ratio of matched trees below 0.5 m (Plots 1-3) and 1.0 m (Plots 4-7) distance as well as mean matched distance are given.

4.5. Experimental setup

We performed a comprehensive evaluation of our proposed method over the 7 diverse test plots. The first group of experiments aims at comparing our approach with a baseline method adopted from Kelbe et al. (2016), whereas the second group deals with the sensitivity of the algorithm to big scale variations between the coregistered datasets.

	Baseline				graph match + SA			
	Scaled		Unscaled		Scaled		Unscaled	
Plot	% match	d_{avg}	% match	d_{avg}	% match	d_{avg}	% match	d_{avg}
1	75	26	72	27	**81**	28	72	27
2	39	31	41	32	44	33	**46**	36
3	48	31	45	31	**78**	28	52	34
4	50	55	44	67	**50**	54	44	66
5	33	57	33	63	30	53	**36**	62
6	41	63	**48**	62	41	75	48	66
7	44	65	44	72	**46**	58	44	67

Table 6: Coregistration quality of baseline method and our approach using 2D rotation, in scaled/unscaled configurations. Distances in cm. Best result per plot outlined in bold. The best result is determined as the one with the highest percentage of matched trees, and in case of equality the tie-breaking criterion is the lower average matched distance.

All experiments were performed on the same tree position sets calculated by tree crown/stem segmentation as described in Sec. 4.2. The planimetric positions were projected onto the respective DTMs to obtain the Z values. Since the aerial and terrestrial data did not possess any common attributes, and all plots were situated in flat terrain, only one relative feature F^1 was used as the basis for coregistration, using 2D Euclidean distance as its metric. In all experiments we used a tiepoint subset size $n_c = 10$.

4.5.1. Comparison with baseline and manual technique

We implemented the method of Kelbe et al. (2016), based on matching tree position triplets using geometric similarity. It should be noted that the original method was designed for coregistering pairs of TLS point clouds, and makes use of DBH estimates for filtering out many triplet pairs. In our setting, this is not possible since DBH estimates are unavailable for the ULS portion of the data. Therefore, a significantly higher number of tiepoint candidates must be examined than would normally be the case for TLS data. We executed several configurations of both the baseline and our method and compared the results. First, the case was considered where the coordinate transform included only a 2D rotation angle, as described in Eq. 8. Also, because the original implementation of the baseline method calculated a 3D rotation, we investigated the second configuration where the transformation includes 3 rotation angles. Our working hypothesis was that because the Z axes were already well aligned, the gain in coregistration quality would be marginal and not worth the additional computational effort. Moreover, for both the baseline and the proposed methods, the calculation of potential corresponding trees takes up a significant portion of the computation time. Therefore, we examined the situation where the geometry-based matching steps of both methods were replaced by random matching. For our method, this meant omitting the graph-based correspondence calculation and generating a completely random set of tree correspondences, after which point the simulated annealing algorithm was applied. For the baseline method, the correspondence ranking of triplet pairs based on their geometry was replaced by a random order. We hypothesized that the matching of tree positions based on their geometric properties is crucial to the coregistration quality. Finally, all three configurations were investigated with and without incorporating a scale factor into the transform, to verify if scaling can lead to a significant improvement even in cases of laser scanning data, where the dataset scales are expected to be approximately equal. In this experiment, since we already knew from coarse coregistration that the relative scale factor is approximately 1, we did not perform the rough scale estimation (Sec. 3.6). To ensure a fair comparison between our method and the baseline, we allocated the same number of computational resources to both, by allowing them to check the same number of corresponding tiepoint subsets as a basis for the CS transformation. Also, for the sake of completeness, we compare both automatic methods to the coarse coregistration technique (Sec. 4.4).

4.5.2. Sensitivity to scaling

In the second major group of experiments, we artificially rescaled each of the 7 test plots by factors of 0.1 to 10 and examined the performance of our method under conditions of large scale differences. We executed the coarse scale estimation as well as the line search for scale refinement (Sec. 3.7.3). The fine-grained scale factors were 0.8 to 1.0 with a step size $\Delta_s = 0.03$, for a total of 10 attempts per test plot and artificial scaling. The goal of these experiments was twofold: (i) to quantify the accuracy of the coarse scale estimate obtained from similarity-based graph matching, and (ii) to determine to what extent the coregistration quality deteriorates as the scale differences between the coregistered datasets increases.

5. Results and discussion

5.1. Comparison with baseline

Table 6 shows the coregistration quality of both the baseline and the proposed method for the case when the

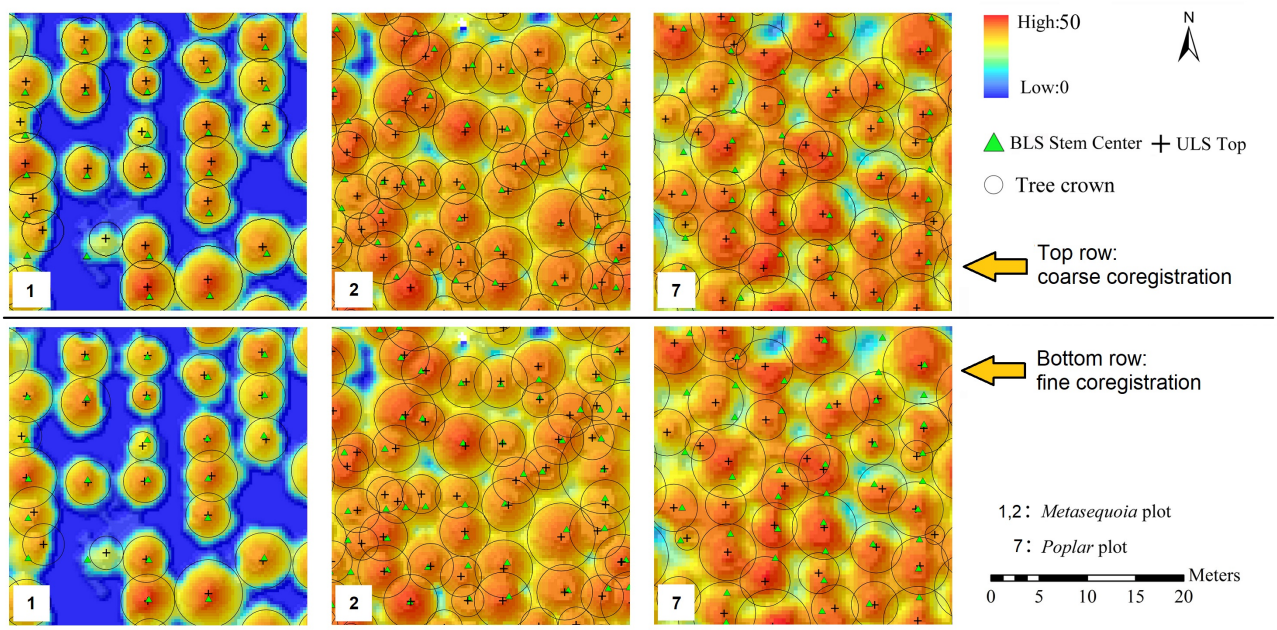


Figure 9: Transformed tree positions after coarse (top row) and fine (bottom row) coregistration procedure for plots 1,2 and 7 shown over false-color digital surface model. Plots 1 and 2 are dominated by conifers, whereas plot 7 consists of mostly broadleaf trees.

CS transform includes a 2D rotation. First, we compare both automatic methods to the coarse manual coregistration (Tab. 5). Apparently, the automatic methods yield significantly higher matched tree rates for all test areas, with increases between 37 - 75 percentage points (pp) on coniferous and 15 - 44 pp on broadleaf plots. This shows that accurate manual coregistration is difficult to achieve and the result may usually be improved upon by automatic methods. It is interesting to note that broadleaf plots have a significantly lower matched tree rate (average 0.45) compared to coniferous plots (average 0.68) although the matched threshold distance d_{thr} for the former is twice that of the latter (1 m vs. 0.5 m). This confirms that coregistering broadleaf plots is generally a harder task due to less precise tree localization. We think that this phenomenon can be attributed to the fact that coniferous trees generally possess a well-defined and distinctive tree top, which provides an estimate of the tree stem position and on the other hand is relatively easy to calculate from ULS data as the point having the highest Z coordinate value within the tree cluster. In contrast, the broadleaf trees utilized in this study were characterized by crowns which did not follow any rigorous shape pattern, therefore the highest points in the crowns were not as good estimates of the stem position as in the coniferous case. Visualizations of coarse (manual) and fine (automatic) coregistration for two plot pairs are shown in Fig. 9.

Turning to the issue of comparing the two automatic methods, the results indicate that for coniferous plots, our method achieves an advantage in the matched tree rate in case of both the scaled and unscaled transform, displaying gains of respectively 5-30 pp and 5-7 pp. On the other hand, on broadleaf plots the average matched tree rate difference is below 1 pp regardless of whether scaling is used. This could be associated with the worse quality of broadleaf tree localization due to the less regular crown shape, however we believe there is a more plausible explanation. The four plots where our method proved superior by the highest margin are precisely the largest 4 plots with respect to the total tree count (see Tab. 4). The number of configurations which the baseline (triplet-based) method considers is fixed and independent of the dataset size. Therefore, for plots with higher tree counts, a smaller percentage of the candidate triplet pairs can be examined, resulting in solutions farther away from the optimum. In this case, our method based on the simulated annealing metaheuristic seems to make better use of the (same) computational resources. Moreover, the baseline method relies on a similarity ranking of all candidate triplet pairs, the number of which is of the order of $n_A^3 \times n_B^3$, or more specifically $O(\binom{n_A}{3}\binom{n_B}{3})$. Even for our moderately-sized datasets, we observed that this step constituted the majority of the computational burden. For larger plots with a higher tree count, exhaustive enumeration of triplets could prove prohibitively computationally expensive. On the other hand, the results in Table 7 suggest that the triplet ordering step is essential to the method's success: for both the scaled and unscaled variations, the average matched tree rate on the 7 plots decreases by 14 pp. Based on these results, we conclude that the triplet enumeration-based method would not scale well to larger datasets in the terrestrial/aerial scenario, where additional common tree attributes for filtering tiepoint candidates are not available. It should be noted that our method is also severely affected when the tree correspondence step is missing (replaced by a random matching) - especially in the scaled case the performance plummets by up to 50 pp. However, we argue that for our method this is less of an issue, because in our case the

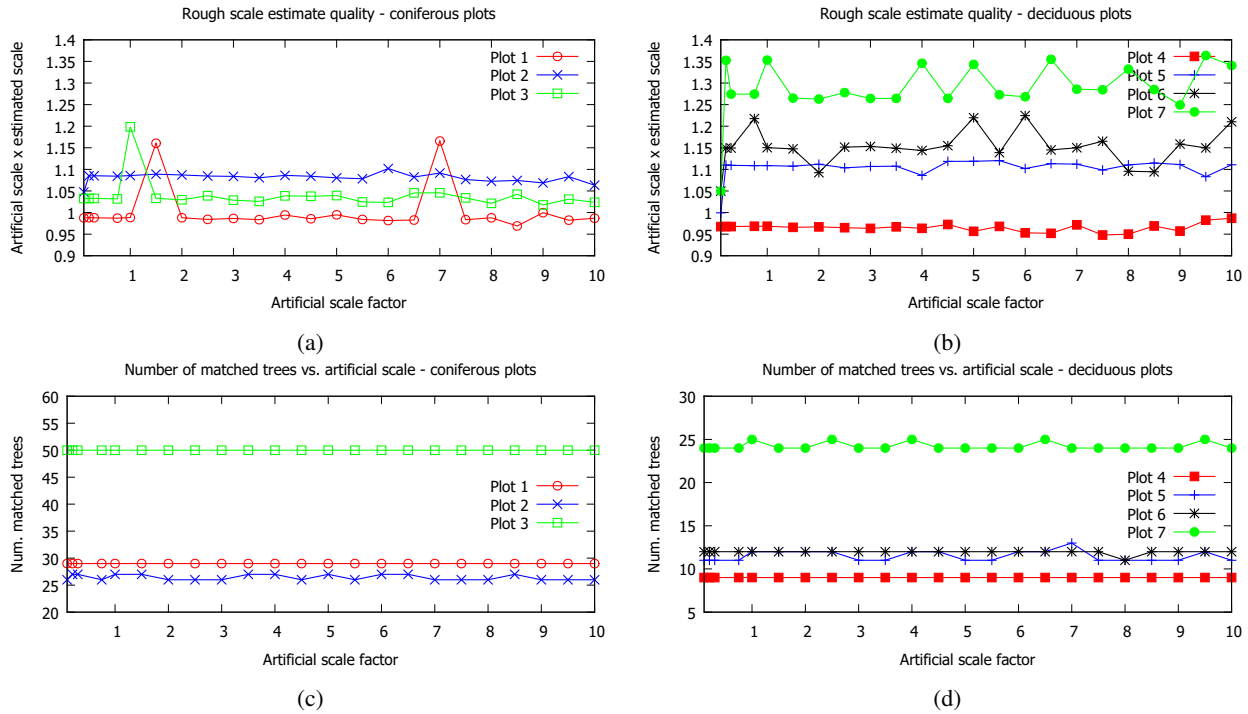


Figure 10: Top row: quality of coarse scale estimate from graph matching vs. scale factor - vertical axis: calculated scale × artificial scale (closer to 1 indicates higher quality); bottom row: dependence of coregistration quality (num. matched trees) on scale factor.

tree correspondence search based on graph matching has a computational complexity of $O(n_A^2 n_B^2 + (\max n_A, n_B)^3)$, which includes executing the Needleman-Wunsch algorithm for all pairs of trees followed by the Kuhn-Munkres algorithm. This function grows significantly slower than the complexity function of exhaustive triplet examination, and hence permits larger datasets before a prohibitive computational cost is reached.

In Table 8, the coregistration results are presented for the scenario where the CS transform was extended to include a 3D rotation instead of a single in-plane rotation angle. As expected, there was little change in the coregistration quality. For our method, 5 out of 7 plots retained their matched tree rates, for one plot the rate dropped by 4 pp and for another, it increased by 2 pp. Also for the baseline method, 5 plots had an unaltered matched rate, and for the remaining two plots the rate increased by 2 and 3 pp. Therefore we conclude that the incorporation of 3 rotation angles instead of a single in-plane rotation does not yield an improvement significant enough to warrant the higher computational cost resulting from operating on 3x3 instead of 2x2 matrices.

Finally, we consider the benefit of introducing a scale factor into the CS transform model. In 3 cases, the scale term led to a significant improvement of coregistration quality. For two coniferous plots, the gain in matched tree rate was 9 and 26 pp, while for the last broadleaf plot, there was a small gain (2 pp) in the matched rate, but more importantly the mean matched tree distance was reduced by 13 %. There were also cases where the scaled variation resulted in a deterioration of the result by 6-7 pp on two smaller broadleaf plots. We believe this could be attributed to an insufficient number of iterations in the simulated annealing algorithm, which indicates that the more complex transformation model incorporating scale needs more computational resources to find the optimal solution. The variations in coregistration quality could be explained by the specific conditions of performing SLAM/point cloud stitching while acquiring the BLS data for each test plot. Note that the optimal scales recovered by our algorithm showed small deviations from 1, not exceeding 3-4 %, which is consistent with an error level that could be introduced by SLAM.

5.2. *Sensitivity to scaling*

The top row of Fig. 10 presents the quality of the coarse scale estimate obtained from maximizing the graph match similarity (Sec. 3.6), as a function of the artificial scale factor applied to the data. Since initially the scales of the ULS and BLS data were approximately equal, a good estimate $\hat{k}_s$ of the scale would be near the inverse of the artificial scale factor $1/k_a$, such that $\hat{k}_s \cdot k_a \approx 1$. We utilize the deviation of this product from 1 as a measure of estimate error. Indeed, for both coniferous and broadleaf plots the product is distributed around 1. In case of the former, 62 % of the estimates have an error below 0.05, and 94 % below 0.09. As for the latter, the quality is somewhat lower, with 70 % estimates having an error not exceeding 0.16, and 91 % not exceeding 0.29. Nevertheless, this scale estimate is useful in both cases, considering that it can be obtained relatively cheaply compared to a grid search on the whole interval of potential scale factors. This is confirmed by the results of the fine grid search around the coarse scale estimates, shown in the bottom row of Fig. 10.

	Baseline (random)				random match + SA			
	Scaled		Unscaled		Scaled		Unscaled	
Plot	% match	d_{avg}	% match	d_{avg}	% match	d_{avg}	% match	d_{avg}
1	31	32	33	25	22	28	**67**	26
2	17	30	31	31	29	31	**41**	36
3	47	28	38	33	**48**	33	47	35
4	39	44	28	51	33	55	**44**	67
5	30	76	27	60	30	59	**30**	55
6	33	60	37	73	37	57	**44**	58
7	37	68	38	60	42	63	**46**	66

Table 7: Coregistration quality of baseline method and our approach using 2D rotation, in scaled/unscaled configurations, with the tree correspondence search/ranking replaced by random matching. Distances in cm. Best result per plot outlined in bold. The best result is determined as the one with the highest percentage of matched trees, and in case of equality the tie-breaking criterion is the lower average matched distance.

	Baseline (3D)				graph match + SA (3D)			
	Scaled		Unscaled		Scaled		Unscaled	
Plot	% match	d_{avg}	% match	d_{avg}	% match	d_{avg}	% match	d_{avg}
1	75	26	72	27	**81**	29	72	27
2	39	29	41	32	**44**	33	**44**	33
3	50	34	45	31	**78**	28	52	34
4	50	55	44	67	**50**	54	44	66
5	33	57	33	63	30	55	**33**	56
6	44	60	**48**	62	37	50	48	74
7	44	66	44	72	**48**	61	46	57

Table 8: Coregistration quality of baseline method and our approach using 3D rotation, in scaled/unscaled configurations. Distances in cm. Best result per plot outlined in bold. The best result is determined as the one with the highest percentage of matched trees, and in case of equality the tie-breaking criterion is the lower average matched distance.

For both coniferous and broadleaf plots, the coregistration quality remains remarkably stable regardless of the point cloud scaling. Three of the plots show no change in the number of successfully matched trees, while for the remaining 4 plots, the difference from the optimal number of matches (at scale 1) is no more than 1 tree. This shows that our proposed approach can handle even large differences in point cloud scales (1:10) quite well using moderate computational effort, since the fine-grained scale search involves only 10 outer iterations of the 'unscaled' processing pipeline.

6. Conclusions and outlook

This work presented a fully automatic approach for coregistering aerial and terrestrial point clouds of forested areas. Our method does not require any auxiliary attributes of trees except the tree positions themselves. The success of our approach depends on the quality of the tree positions recovered by the respective localization methods in terrestrial and aerial data. Some factors that may inhibit the calculation of tree positions are (i) dense, high ground vegetation and/or forest regenerations, occluding tree stems from the terrestrial perspective, and (ii) high density of trees in the plot, resulting in a thick canopy cover and complicating 3D tree segmentation from aerial/ULS data. The method was applied to 7 pairs of ULS/BLS plots with varying characteristics. Experiments showed that the automatic method can greatly improve the coregistration quality compared to manual techniques. Also, processing coniferous plots seems significantly easier than broadleaf plots, probably due to the irregular shape of broadleaf tree crowns lacking a distinct center position. This is in contrast to coniferous trees, which follow an approximate elliptic paraboloid model with a clear peak. Although both data types (ULS,BLS) are derived from laser scanning and their scales are similar, apparently the creation of the BLS point cloud through SLAM techniques may in some cases lead to small scale variations, which may be compensated

by introducing a scale term into the coregistering coordinate transform. Moreover, we found that the decoupling of the tree correspondence matching step and the optimal transform derivation can lead to superior results compared to an exhaustive-search style approach, which may be attributed to a reduction of the problem's solution space as well as more efficient use of computational resources by the applied simulated annealing metaheuristic. Tests on artificially scaled data indicated that the proposed framework can handle large differences in the relative scales of the coregistered datasets, which could prove useful for processing other data types lacking an absolute scale, e.g. photogrammetric point clouds. In the future, we would like to test our framework on a broad spectrum of point cloud types from different sources. Although this study was carried out on stands from planted forests and hence situated on a raster, we believe our method would be well suited to natural forests. In fact, the inherent randomness of tree positions in natural forests could improve the discriminative capabilities of the descriptors used for tree matching, because the variety in inter-tree distances would be greater compared to a regular structure (grid). As long as sufficiently many common tree positions can be recovered in both data sources, the coregistration method should be applicable to mixed as well as multi-layered stands. The precise boundary conditions (stand characteristics) beyond which coregistration is likely to fail are left to future investigations. An interesting future direction would be to develop more accurate approximations of the tree positions for aerial data, for example derived from paraboloid fitting. Particularly improving the localization of broadleaf trees based on their crowns could bridge the performance gap between coniferous and broadleaf plots.

Acknowledgment

The research data acquisition was supported by the National Key R&D Program of China (2017YFD0600904) and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

References

Amiri, N., Polewski, P., Yao, W., Krzystek, P., Skidmore, A.K., 2017. Detection of single tree stems in forested areas from high density als point clouds using 3d shape descriptors. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences IV-2/W4, 35–42. doi:10.5194/isprs-annals-IV-2-W4-35-2017.

Bauwens, S., Bartholomeus, H., Calders, K., Lejeune, P., 2016. Forest inventory with terrestrial lidar : a comparison of static and hand-held mobile laser scanning. Forests 7, 17. doi:10.3390/f7060127.

Brede, B., Lau, A., Bartholomeus, H., Kooistra, L., 2017. Comparing RIEGL RiCOPTER UAV LiDAR derived canopy height and DBH with terrestrial LiDAR. Sensors 17, 2371. doi:10.3390/s17102371.

Cao, L., Gao, S., Li, P., Yun, T., Shen, X., Ruan, H., 2016. Aboveground biomass estimation of individual trees in a coastal planted forest using full-waveform airborne laser scanning data. Remote Sensing 8, 729. doi:10.3390/rs8090729.

Dandois, J.P., Olano, M., Ellis, E.C., 2015. Optimal altitude, overlap, and weather conditions for computer vision uav estimates of forest structure. Remote Sensing 7, 13895–13920. doi:10.3390/rs71013895.

GreenValley, 2018. GreenValley International. `http://greenvalleyintl.com/`. Accessed: 2018-07-13.

Guo, Q., Su, Y., Hu, T., Zhao, X., Wu, F., Li, Y., Liu, J., Chen, L., Xu, G., Lin, G., Zheng, Y., Lin, Y., Mi, X., Fei, L., Wang, X., 2017. An integrated UAV-borne lidar system for 3D habitat mapping in three forest ecosystems across China. International Journal of Remote Sensing 38, 2954–2972. doi:10.1080/01431161.2017.1285083.

Hauglin, M., Lien, V., Naesset, E., Gobakken, T., 2014. Geo-referencing forest field plots by co-registration of terrestrial and airborne laser scanning data. International Journal of Remote Sensing 35, 3135–3149. doi:10.1080/01431161.2014.903440.

Hyyppä, J., Holopainen, M., Olsson, H., 2012. Laser scanning in forests. Remote Sensing 4, 2919–2922. doi:10.3390/rs4102919.

Kärkkäinen, T., Äyrämö, S., 2005. On computation of spatial median for robust data mining, in: Schilling, R., Haase, W., Periaux, J., Baier, H., Bugeda, G. (Eds.), Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems.

Kelbe, D., van Aardt, J., Romanczyk, P., van Leeuwen, M., Cawse-Nicholson, K., 2016. Marker-free registration of forest terrestrial laser scanner data pairs with embedded confidence metrics. IEEE Transactions on Geoscience and Remote Sensing 54, 4314–4330. doi:10.1109/TGRS.2016.2539219.

Korpela, I., Hovi, A., Morsdorf, F., 2012. Understory trees in airborne lidar data selective mapping due to transmission losses and echo-triggering mechanisms. Remote Sensing of Environment 119, 92 – 104. doi:https://doi.org/10.1016/j.rse.2011.12.011.

Kukko, A., Kaijaluoto, R., Kaartinen, H., Lehtola, V.V., Jaakkola, A., Hyypp, J., 2017. Graph slam correction for single scanner mls forest data under boreal forest canopy. ISPRS Journal of Photogrammetry and Remote Sensing 132, 199 – 209. doi:https://doi.org/10.1016/j.isprsjprs.2017.09.006.

Li, W., Guo, Q., Jakubowski, M.K., Kelly, M., 2012. A new method for segmenting individual trees from the lidar point cloud. Photogrammetric Engineering & Remote Sensing 78, 75–84. doi:10.14358/PERS.78.1.75.

LiForest, 2018. GreenValley International. `http://www.liforest.com/`. Accessed: 2018-09-28.

Lindberg, E., Holmgren, J., Olofsson, K., Olsson, H., 2012. Estimation of stem attributes using a combination of terrestrial and airborne laser scanning. European Journal of Forest Research 131, 1917–1931. doi:10.1007/s10342-012-0642-5.

Needleman, S.B., Wunsch, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48, 443 – 453.

Paris, C., Kelbe, D., van Aardt, J., Bruzzone, L., 2017. A novel automatic method for the fusion of als and tls lidar data for robust assessment of tree crown structure. IEEE Transactions on Geoscience and Remote Sensing 55, 3679–3693. doi:10.1109/TGRS.2017.2675963.

Persad, R.A., Armenakis, C., 2017. Automatic co-registration of 3d multi-sensor point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 130, 162 – 186. doi:https://doi.org/10.1016/j.isprsjprs.2017.05.014.

Polewski, P., Erickson, A., Yao, W., Coops, N., Krzystek, P., Stilla, U., 2016. Object-based coregistration of terrestrial photogrammetric and als point clouds in forested areas. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences III-3, 347–354.

Polewski, P., Yao, W., Heurich, M., Krzystek, P., Stilla, U., 2015. Detection of fallen trees in ALS point clouds using a Normalized Cut approach trained by simulation. ISPRS Journal of Photogrammetry and Remote Sensing 105, 252 – 271. doi:http://dx.doi.org/10.1016/j.isprsjprs.2015.01.010.

Polewski, P., Yao, W., Heurich, M., Krzystek, P., Stilla, U., 2017. A voting-based statistical cylinder detection framework applied to fallen tree mapping in terrestrial laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 129, 118 – 130. doi:https://doi.org/10.1016/j.isprsjprs.2017.04.023.

Rönnholm, P., Liang, X., Kukko, A., Jaakkola, A., Hyyppä, J., 2016. Quality analysis and correction of mobile backpack laser scanning data. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences III-1, 41–47. doi:10.5194/isprs-annals-III-1-41-2016.

Sankey, T., Donager, J., McVay, J., Sankey, J.B., 2017. Uav li-

dar and hyperspectral fusion for forest monitoring in the southwestern usa. Remote Sensing of Environment 195, 30 – 43. doi:https://doi.org/10.1016/j.rse.2017.04.007.

St-Onge, B., Audet, F.A., Bégin, J., 2015. Characterizing the height structure and composition of a boreal forest using an individual tree crown approach applied to photogrammetric point clouds. Forests 6, 3899. doi:10.3390/f6113899.

Tang, J., Chen, Y., Kukko, A., Kaartinen, H., Jaakkola, A., Khoramshahi, E., Hakala, T., Hyyppä, J., Holopainen, M., Hyyppä, H., 2015. Slam-aided stem mapping for forest inventory with small-footprint mobile lidar. Forests 6, 4588–4606. doi:10.3390/f6124390.

Tao, S., Wu, F., Guo, Q., Wang, Y., Li, W., Xue, B., Hu, X., Li, P., Tian, D., Li, C., Yao, H., Li, Y., Xu, G., Fang, J., 2015. Segmenting tree crowns from terrestrial and mobile lidar data by exploring ecological theories. ISPRS Journal of Photogrammetry and Remote Sensing 110, 66 – 76. doi:https://doi.org/10.1016/j.isprsjprs.2015.10.007.

Theiler, P.W., Wegner, J.D., Schindler, K., 2014. Keypoint-based 4-points congruent sets automated marker-less registration of laser scans. ISPRS Journal of Photogrammetry and Remote Sensing 96, 149. doi:http://dx.doi.org/10.1016/j.isprsjprs.2014.06.015.

Umeyama, S., 1991. Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 376–380. doi:10.1109/34.88573.

Wallace, L., Lucieer, A., Malenovsky, Z., Turner, D., Vopenka, P., 2016. Assessment of forest structure using two uav techniques: A comparison of airborne laser scanning and structure from motion (sfm) point clouds. Forests 7, 1–16. doi:10.3390/f7030062.

Weinmann, M., Jutzi, B., 2015. Geometric point quality assessment for the automated, markerless and robust registration of unordered TLS point clouds. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences II-3/W5, 89–96. doi:10.5194/isprsannals-II-3-W5-89-2015.

Wu, J., Cawse-Nicholson, K., van Aardt, J., 2013. 3d tree reconstruction from simulated small footprint waveform lidar. Photogrammetric Engineering & Remote Sensing 79, 1147–1157. doi:doi:10.14358/PERS.79.12.1147.

Zhao, X., Guo, Q., Su, Y., Xue, B., 2016. Improved progressive tin densification filtering algorithm for airborne lidar data in forested areas. ISPRS Journal of Photogrammetry and Remote Sensing 117, 79 – 91. doi:https://doi.org/10.1016/j.isprsjprs.2016.03.016.