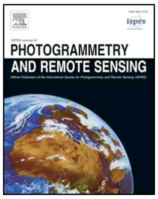
ISPRS Journal of Photogrammetry and Remote Sensing 163 (2020) 214–230

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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



SLAM-aided forest plot mapping combining terrestrial and mobile laser scanning

Jie Shao[a,b], Wuming Zhang[c,d,*], Nicolas Mellado[b], Nan Wang[e], Shuangna Jin[a], Shangshu Cai[a], Lei Luo[f], Thibault Lejemble[b], Guangjian Yan[a]

[a] *State Key Laboratory of Remote Sensing Science, Beijing Engineering Research Center for Global Land Remote Sensing Products, Institute of Remote Sensing Science and Engineering, Faculty of Geographical Science, Beijing Normal University, Beijing 100875, China*
[b] *IRIT, CNRS, University of Toulouse, Toulouse 31062, France*
[c] *School of Geospatial Engineering and Science, Sun Yat-Sen University, Zhuhai 519082, China*
[d] *Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai), Guangdong, China*
[e] *School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China*
[f] *Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China*

ARTICLE INFO

Keywords:
Forest mapping
LiDAR
SLAM
Single-scan TLS
MLS

ABSTRACT

Precise structural information collected from plots is significant in the management of and decision-making regarding forest resources. Currently, laser scanning is widely used in forestry inventories to acquire three-dimensional (3D) structural information. There are three main data-acquisition modes in ground-based forest measurements: single-scan terrestrial laser scanning (TLS), multi-scan TLS and multi-single-scan TLS. Nevertheless, each of these modes causes specific difficulties for forest measurements. Due to occlusion effects, the single-scan TLS mode provides scans for only one side of the tree. The multi-scan TLS mode overcomes occlusion problems, however, at the cost of longer acquisition times, more human labor and more effort in data preprocessing. The multi-single-scan TLS mode decreases the workload and occlusion effects but lacks the complete 3D reconstruction of forests. These problems in TLS methods are largely avoided with mobile laser scanning (MLS); however, the geometrical peculiarity of forests (e.g., similarity between tree shapes, placements, and occlusion) complicates the motion estimation and reduces mapping accuracy.

Therefore, this paper proposes a novel method combining single-scan TLS and MLS for forest 3D data acquisition. We use single-scan TLS data as a reference, onto which we register MLS point clouds, so they fill in the omission of the single-scan TLS data. To register MLS point clouds on the reference, we extract virtual feature points that are sampling the centerlines of tree stems and propose a new optimization-based registration framework. In contrast to previous MLS-based studies, the proposed method sufficiently exploits the natural geometric characteristics of trees. We demonstrate the effectiveness, robustness, and accuracy of the proposed method on three datasets, from which we extract structural information. The experimental results show that the omission of tree stem data caused by one scan can be compensated for by the MLS data, and the time of the field measurement is much less than that of the multi-scan TLS mode. In addition, single-scan TLS data provide strong global constraints for MLS-based forest mapping, which allows low mapping errors to be achieved, e.g., less than 2.0 cm mean errors in both the horizontal and vertical directions.

1. Introduction

Precise measurements of forest structures are crucial for the management of and decision-making regarding forest resources, studies of ecosystem processes and biodiversity, and so on (Spies, 1998). However, accurate forestry measurements are not straightforward because of the complexity of forests. Some conventional, simple tools (e.g., calipers and clinometers) have been used in forest field measurements, but these measurement methods are widely recognized as time consuming, laborious and expensive. Terrestrial laser scanning (TLS), also known as ground-based light detection and ranging (LiDAR), has been suggested to be a practical option to quickly provide accurate and nondestructive estimations of forest biophysical metrics (Latifi et al., 2015; Stovall et al., 2017; Wilkers et al., 2017). Compared to

[*] Corresponding author at: School of Geospatial Engineering and Science, Sun Yat-Sen University, Zhuhai 519082, China.
E-mail address: zhangwm25@mail.sysu.edu.cn (W. Zhang).

https://doi.org/10.1016/j.isprsjprs.2020.03.008
Received 15 May 2019; Received in revised form 6 March 2020; Accepted 11 March 2020

conventional manual measurements, it has also shown a higher work efficiency in forestry inventories (Murphy et al., 2010; Luo et al., 2019). Three-dimensional (3D) data acquisition is an essential prerequisite and is the key to digital forest measurements. Three data-acquisition modes have been reported in TLS-based field measurements: single-scan, multi-scan and multi-single-scan (Liang and Hyyppä, 2013).

In single-scan mode, the scanner is placed at a single point in the forest sample plot, which allows the acquisition of only one side of the visible trees. The single-scan mode has a simpler data-acquisition setting and faster measuring speed than those of the other two modes, e.g., it can typically measure a plot within 20 min (Liang et al., 2016). However, occlusion effects from the forest objects (e.g., trees, branches, and ground vegetation) in the direction of the laser beams cause a low detection rate in forest measurements, e.g., from 0% to 46% depending on the plot size and the forest type (Maas et al., 2008; Lovell et al., 2011; Liang et al., 2012; Trochta et al., 2013; Mengesha et al., 2015; Wan et al., 2019). Consequently, multiple scans are often necessary to observe all the trees in the sample plot. In multi-scan mode, the scanner observes a sample plot from multiple positions, which gives it the potential to detect all trees and provides full coverage of a stem surface. This mode is considered the most accurate for forest mapping. Unfortunately, it requires more time and laborious field measurements and more data preprocessing. Depending on the plot size and forest type, the multi-scan mode typically takes one to ten hours to measure a forest sample plot, which includes determining the location and distribution of the scanners and reference targets and performing multiple scans. For example, TLS-based field measurements usually take at least one hour in a forest sample plot of 30 m × 30 m in size and approximately ten hours in a plot of 100 m × 100 m in size. In addition, the cost of the manual or semiautomated registration of multiple scans limits its practicality. For instance, placing reflective targets in forest environments is complex because most methods evenly distribute targets in positions that can be seen from multiple viewpoints and require additional user interactions to identify undetected targets, which are difficult to extract automatically by commercial software. In multi-single-scan mode, several scans are implemented in a sample plot, but artificial targets and data-level registration are not required. The multi-single-scan mode performs processing of each scan independently and automatically detects the individual trees from each individual scan to map the forest sample plot. This mode offers compensation for the problem of the occlusion effects in the single-scan mode and decreases the workload of field measurement compared to that of the multi-scan mode.

In principle, the registration of multiple scans can be performed at different levels, including the data-level (point-level), feature-level and decision-level. The data-level (or point-level) registration of multiple scans transforms several point clouds into a common coordinate system, and the fused data are used for the interpretation of the forest sample plot. In general, the multi-scan mode performs the process of data-level registration. At the feature level, some features (e.g., diameter at breast height and tree height) are extracted from each individual scan and merged for estimating tree attributes. At the decision level, the individual scans are processed independently, and the extracted tree attributes from each scan are combined for the interpretation of the plot. In the three acquisition modes, the multi-single-scan mode generally performs the process of merging at the feature and/or decision levels. In forest inventories, the multi-single-scan mode can improve the estimation accuracy of tree attributes compared with that of the single-scan mode and obtain a similar estimation result with that of the multi-scan mode (Liang and Hyyppä, 2013). However, for the precise and complete 3D reconstruction of the forest environment, data-level registration is more necessary than merging at the feature and decision levels.

Recently, mobile laser scanning (MLS) has gained attention in forest plot mapping because of the advantage of immensely faster data collection in comparison to those of TLS modes (Liang et al., 2014), e.g., MLS can measure small sample plots within a few minutes. Therefore, we propose a simultaneous localization and mapping (SLAM) method combining a single-scan TLS point cloud and MLS point clouds for forest environments. The method addresses the global consistency problem and maintains the accuracy of mapping without the GNSS-IMU system, even in the case of trajectory discontinuity and without loop closures. To solve the occlusion and object similarities problem, we propose combining virtual feature points that represent the tree stem center and real, evenly distributed feature points in forest plot mapping, which allows the mapping of forest point clouds with low overlap and prevents errors caused by inaccurate corresponding pairs in forest environments. Following the introduction section, Section 2 summarizes the related work. The key steps of the proposed method are elaborated in Section 3. Section 4 introduces the materials and the performance of the proposed method on field measurements and then evaluates the advancements of the proposed method, after which discussions are presented and conclusions are drawn.

2. Related work

The main limitation of MLS appears during the mapping step, where each MLS point cloud is mapped on the point clouds acquired in the previous time steps. Most of the existing MLS-based mapping techniques are based on global navigation satellite system/inertial measurement unit (GNSS-IMU)-based techniques. In that setup, the GNSS maintains the global position accuracy, and the IMU provides attitude information for the orientation of the laser scanner. In contrast to the road and urban contexts, the occlusion of trees often weakens or blocks the GNSS signal and prevents forest mapping. In such cases, the location of the MLS point cloud needs to be estimated during the mapping step, which leads to the so-called SLAM problem (Dissanayake et al, 2001).

Common SLAM techniques involve filter-based and graph-based methods. In the filter-based method, the extended Kalman filter (EKF) (e.g., Hector SLAM) (Kohlbrecher et al., 2011) and particle filters (PF) (e.g., G-mapping) (Grisetti et al., 2007) are the common filters for SLAM technology. The related EKF and PF methods rely on strong assumptions about the robot motion model and the sensor noise and generally only consider the motion relationship between adjacent data. When the assumptions are violated or loop closure is executed, the filter-based methods will be difficult to address. In addition, as the scenario expands, the filter-based method will increase the memory consumption and computation. The graph-based method is popular in the SLAM community, as it solves both the position and mapping problems by combining poses of the scanner and constraint relationships between these poses. For example, Karto-SLAM (Konolige et al., 2010) and Cartographer (Hess, et al., 2016) calculate the poses of the scanner at different times and execute loop closure detection to construct a pose graph and then eliminate the cumulative error by optimizing the pose graph. Because only pose optimization is considered, these methods can achieve low computational resource consumption and even real-time optimization. However, this kind of method has difficulty obtaining highly accurate positioning and mapping results, which makes it difficult to meet the requirements of high-precision forest measurements. In contrast, another graph-based method, bundle adjustment (BA), is widely used for solving the SLAM problem (Mouragnon et al., 2009). The BA method simultaneously optimizes the features and poses of the scanner using nonlinear optimization, which strongly relies on the matched features and can obtain highly accurate mapping results. For instance, the LOAM method (Zhang and Singh, 2014) selects the line and plane features on object surfaces to estimate the motion of a scanner and obtains highly accurate mapping results in indoor and urban scenarios that consist of stable and distinct features. However, due to the complexity and similarity of the object in forests, reliable features are difficult to extract from the object surface, and inaccurate corresponding pairs can make scan matching fall into a local optimum. Moreover, another challenge in SLAM is to avoid error accumulation during data acquisition, i.e., by considering global optimization

(Grisetti et al., 2010). Several methods maintain global positioning accuracy by performing multiple loop-closure detections (Mur-Artal, et al., 2015), generally increasing the complexity of the algorithm (Labbé and Michaud, 2014). Other studies achieve global optimization using prior information. For example, Kukko et al. (2017) used graph-based SLAM to correct the GNSS-IMU trajectory for position drift, and in turn, the initial trajectory obtained from the GNSS-IMU system was regarded as prior information and provided the constraints for the graph SLAM. The method not only achieved forest mapping but also improved the absolute positioning accuracy; however, the GNSS signal loss caused by the occlusions of trees may affect the performance in practice. In addition, aerial images (Kümmerle et al., 2011; Javanmardi et al., 2017) and sketch maps (Shah and Campbell, 2013; Behzadian et al., 2015; Mielle et al., 2018) are usually used to generate constraints for SLAM-based mapping, and the related methods show the reliability of prior information in indoor and urban scenarios (Wang et al., 2018). However, aerial images have difficulty providing constraints for below-canopy forest mapping due to canopy occlusions; in addition, methods based on sketch maps have difficulty meeting mapping requirements for high-precision forest measurements. Consequently, the complex and irregular forest environments pose problems to the existing SLAM methods.

A key step of most SLAM systems is the registration step, where pairs of input scans are matched and aligned. This process is known as point cloud registration, a topic of studies for decades with a wide range of approaches proposed in the general setting (Mitra et al. 2004, Rusu et al. 2008, Mellado et al. 2014, Pomerleau et al. 2015). In this work, we focus on forest mapping and restrict our review to this application case. The first type of approach uses artificial markers placed in the scene, e.g., reflective tape, retroreflective spheres, and reflectors (Henning and Radtke, 2006; Hilker et al., 2012; Zhang et al., 2016a). These markers serve as precise and unambiguous tie points for achieving highly accurate point cloud registration. Nevertheless, the placement of artificial markers is generally time consuming and labor intensive in forest environments, and related studies commonly focus on several dense point clouds (e.g., TLS data). Extracting markers from sparse MLS data is, however, more difficult due to the size of markers and the effects of occlusion and might also be impractical for thousands of MLS point clouds. In contrast, marker-free registration methods aim at registering the scans without markers. The first category of approaches detects geometric features that play the role of digital markers. For example, Kelbe et al. (2016) regarded stem-terrain intersection points as matching features and generated tie point triplets for registration of TLS data. Polewski et al. (2019) used the tree positions to achieve marker-free registration of point clouds. A second category of approaches uses descriptor-free registration methods; see, for instance, the study by Theiler et al., (2014a) on the use of congruent set registration for, among other application cases, forest registration. The aforementioned methods commonly focus on coarse alignment and require a fine registration step in postprocessing. The iterative closest point (ICP) (Besl and Mckay, 1992) is currently the standard approach for local registration. It starts by computing correspondences between pairs of point clouds and then minimizes the distance between those corresponding pairs. Normal distribution transform (NDT) (Magnusson et al., 2007) is also often considered. In the NDT method, a point cloud is represented by local normal distributions that are subdivided into a grid of cells; then, the search of the closest normal distribution is used to replace iterating over a whole point cloud. The two methods generally need a certain point cloud overlap to maintain the registration accuracy and are suitable for scenarios with strong anti-density interference abilities, e.g., indoor and urban. However, forest scenarios exhibit strong auto-similarities (e.g., tree stems are often very similar), which may confuse these methods. When the distance between scan locations is large, inaccurate corresponding pairs are easily generated by the methods and cause inaccurate registration results, especially in the registration of tree stems (Fig. 1). Therefore, the selection of features brings challenges for scan matching in forest environments.

3. Methods

3.1. Overview of the method

The goal of this paper is to realize forest plot mapping by combining single-scan TLS data and MLS data. In this paper, we solve the problem in two steps: LiDAR odometry and global optimization (see Fig. 2).

(1) Odometry: estimate the motion of each frame MLS point cloud relative to the single-scan TLS data:

i. Feature extraction: extract feature points from each frame MLS point cloud. We distinguish between *virtual* features, which sample the reconstructed tree stem centerlines, and *real* features, which evenly sample the point cloud. In this paper, virtual features are points that are not part of the input point clouds and thus cannot be directly extracted from them, and the points that can be directly extracted from the point cloud are defined as real features. A layered clustering method is proposed for virtual feature extraction, and the *difference of Gaussian* (DoG) method is used to extract real features.

ii. Feature correspondence: *virtual* and *real* feature points are matched between the reference single-scan TLS data and each MLS point cloud by a nearest neighbor search.

iii. Motion estimation: the pose of each MLS point cloud is optimized with respect to the single-scan TLS point cloud by nonlinearly minimizing the corresponding feature point distances.

(2) Global optimization: simultaneously optimize the poses of all the MLS point clouds. The coordinate system of the reference single-scan TLS point cloud can be considered a common coordinate system and defines a global constraint for estimating the trajectory of the MLS device. We use an incremental map to optimize the poses of all the MLS point clouds, of which the incremental map is built by the continuous MLS point clouds and the reference.

3.2. LiDAR odometry

LiDAR odometry is used to estimate the motion of the MLS system. In this paper, it is used to calculate the transformation between the MLS point cloud and the reference. To reduce the cost of field measurement and extract accurate corresponding pairs, we propose combining two types of features, virtual features and real features, for LiDAR odometry. Before solving the MLS-based SLAM problem in a forest, we first set certain conditions:

- The single-scan TLS point cloud is set as a reference, and its coordinate system is regarded as the common coordinate system $\{W\}$. The coordinate system of each MLS point cloud is set to a local coordinate system $\{L\}$. $\{W\}$ and $\{L\}$ follow the right hand rule.
- Let M_n and Γ_{TLS} be the MLS point cloud at the time of sweep n and the TLS point cloud, respectively.
- Coarse alignment of the first frame MLS point cloud and the reference is manually performed.

The NDT algorithm only considers the probability distribution of points, so it does not take much time to search for matching features and has a certain stability; thus, the algorithm is used to provide the initial transformation for the following LiDAR odometry. Let $\widetilde{M}_n^w$ be the reprojected point cloud based on the NDT.

3.2.1. Feature extraction

The NDT algorithm can obtain an initial transformation between the MLS point cloud and the reference, but there are obvious deviations in tree stem registration, especially in the horizontal direction (see Fig. 1). Therefore, a constraint that combines the virtual features and real

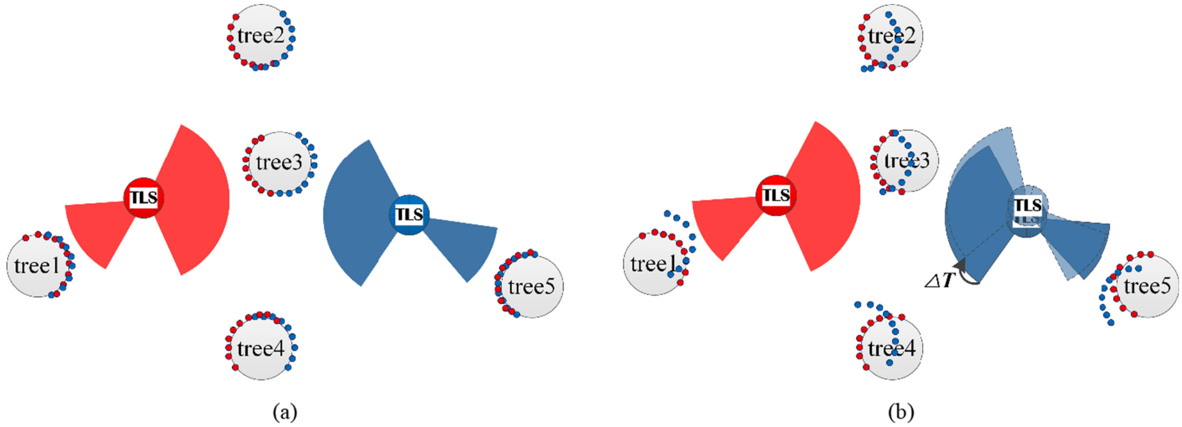


Fig. 1. The existing methods generally fail in forest environments because of the inaccurate corresponding pairs. The overlapping rates in tree 1 and tree 5 are high; the overlapping rates in tree 2, tree 3, and tree 4 are low, especially in tree 3. Result (b) will replace result (a) when applying the conventional methods of point cloud registration.

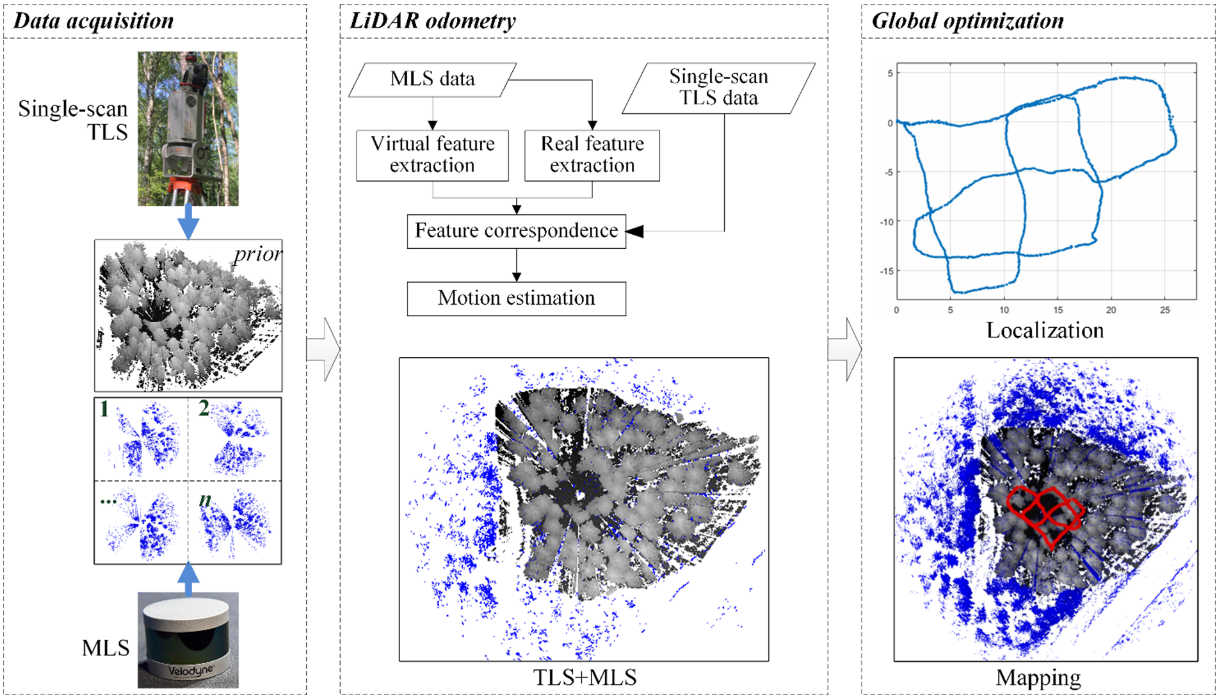


Fig. 2. Flowchart.

features is proposed to reduce registration error.

(1) Extraction of virtual features

In forestry inventories, in practice, we generally assume that the cross sections of tree stems are approximately circular (Polewski et al., 2017) and that the shortest distance from the geometric center of the tree stem cross section to the surface of the tree stem equals the radius of the circle. It is obvious that these assumptions provide horizontal constraints. Therefore, we use the geometric centers of tree stems as horizontal constraints for the registration of the MLS point cloud and the single-scan TLS data. Because the centers of stems cannot be directly extracted from point clouds, we defined them as virtual features.

Specifically, we extract virtual features based on the layered clustering method. First, we divide each MLS point cloud into multiple subsets according to the vertical angular resolution. Then, the continuous and adjacent points in each subset are clustered based on the connected-component labeling algorithm (Zhang et al., 2019). If the distance between the two farthest points in a cluster is greater than the maximum diameter at breast height (DBH) or less than the minimum DBH in the plot, then the cluster will be removed. Finally, the circle fit based on the least squares method is used to detect the centers from these retained clusters. Furthermore, the centers that are continuously distributed in the vertical direction are used as virtual features (see Fig. 3). Let $\mathscr{V}_n^L$ be the set of virtual features at the time of sweep n ($\mathscr{V}_n^L \subset \mathrm{M}_n$) and $\tilde{\mathscr{V}}_n^W$ be the reprojected point sets based on the initial transformation. In addition, let $r_{\mathscr{V}_n}$ be the set of radii corresponding to the features.

(2) Extraction of real features

Although virtual features can reduce the deviations in the horizontal direction, it is difficult to provide a constraint in the vertical direction because the tree stems are generally parallel in the vertical direction. However, those features that are evenly distributed in the point cloud can provide an overall registration constraint. In this paper, we use the

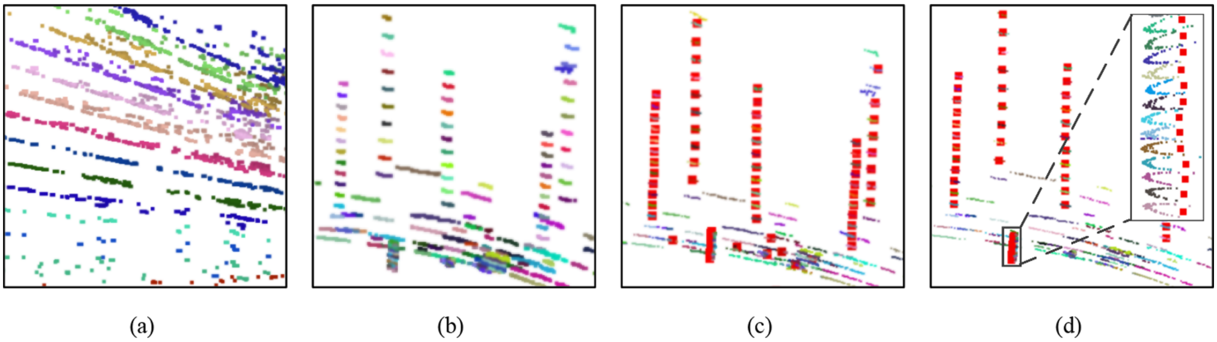


Fig. 3. Extraction of virtual features. (a) Layered processing; different colors represent subsets on different layers. (b) Retaining cluster points. (c) Fitting the circle whose radius is less than half of the maximum DBH (red points) and detecting its center. (d) Determining virtual features (larger red points). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

difference of Gaussian (DoG) algorithm to extract real features.

The DoG algorithm is a feature enhancement and extraction algorithm in digital image processing. The major advantages of features extracted by the DoG algorithm are their invariance to scaling, rotation, and translation. Many high-contrast edges in two dimensions are on object silhouettes and have depth discontinuities, leading to unstable features across viewpoints. 3D detection avoids such unstable feature points that highly contrast with their spatial neighbors, so we directly extract the real features in 3D. The DoG-based feature extraction computes a Gaussian response for each point in each blurred level and then subtracts the responses of adjacent blurred levels at each point to obtain DoG responses (Theiler et al., 2014b). The blurred level is determined by the Gaussian responses of normal vectors at each point. When the normal vector of each point is calculated by taking into account different search scales, different blurred levels will be obtained. For the DoG responses, points with local maxima or minima in the response space are extracted as real features. In practice, to reduce the impact of noise or outliers, we calculate the spatial distance between each point and its neighboring points. If the distance is large, the point will be tagged as noise or outlier and removed. Finally, let $\mathscr{R}_n^L$ be the set of real features at the time of sweep n ($\mathscr{R}_n^L \subset \mathrm{M}_n$) and $\widetilde{\mathscr{R}}_n^W$ be the reprojected point sets based on the initial transformation. The result is shown in Fig. 4.

3.2.2. *Feature correspondence*

Feature correspondence is used to search for the corresponding features of the virtual feature and real feature from the MLS and the single-scan TLS data and to estimate the motion of the MLS system in the common coordinate system. Therefore, the nearest neighbor algorithm is used to search for corresponding features.

(1) Virtual feature correspondence

Theoretically, the distance from the center of the cross section of the tree stem to the nearest point on the surface of the tree stem is approximately equal to the radius of the cross section, so the point-to-point distance can easily establish a more accurate relationship between the MLS point cloud and the reference. If a distance between a virtual feature and its nearest point in the reference data is approximately equal to the corresponding radius of the virtual feature, then we set the virtual feature point as a keypoint for motion estimation and regard its nearest point in the reference data as its corresponding feature. Let $\widetilde{X}^W_{(\mathscr{V}_n,i)}$ be a keypoint ($\widetilde{X}^W_{(\mathscr{V}_n,i)} \in \widetilde{\mathscr{V}}_n^W$) and $\widetilde{X}_{(\Gamma,j)}$ be the corresponding point ($\widetilde{X}_{(\Gamma,j)} \in \Gamma_{TLS}$); then, the point-to-point distance $d_{(\mathscr{V}_n,i)}$ can be computed by

$$d_{(\mathscr{V}_n,i)} = |\widetilde{X}^W_{(\mathscr{V}_n,i)} - \widetilde{X}_{(\Gamma,j)}| \tag{1}$$

for each virtual keypoint. When $d_{(\mathscr{V}_n,i)}$ approximates the corresponding radius, the relationship between the keypoint and the corresponding point will be more accurate.

(2) Real feature correspondence

The point-to-plane metric is usually solved using standard nonlinear least squares methods, and the error metric converges much faster (Low, 2004). Therefore, in this paper, the planar patch is found to be the corresponding feature of a real feature. If a real feature has three nearest points in the reference data, which are within a certain

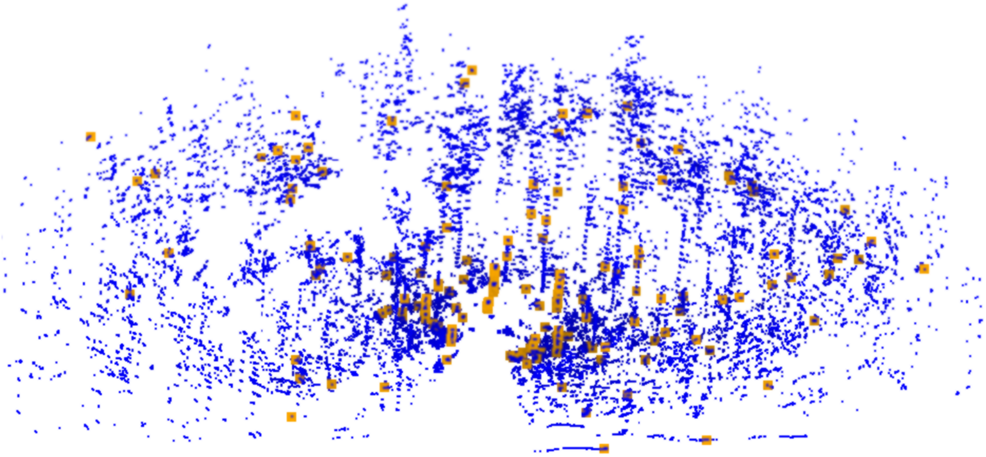


Fig. 4. Extraction of real features. Blue points are the MLS data. Orange points are real feature points, and 180 real feature points were extracted from each MLS point cloud. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

neighborhood of the real feature point and are not on the same line, then a planar patch consisting of the three points is regarded as a corresponding feature of the real feature, and the real feature point will be used as a keypoint for estimating motion in the MLS system. Let $\widetilde{X}^W_{(\mathscr{R}_n,i)}$ be a keypoint in $\widetilde{\mathscr{R}}^W_n$ ($\widetilde{X}^W_{(\mathscr{R}_n,i)} \in \widetilde{\mathscr{R}}^W_n$), and let the corresponding plane be set to $\{\widetilde{X}_{(\Gamma,a)}, \widetilde{X}_{(\Gamma,b)}, \widetilde{X}_{(\Gamma,c)}\} \subset \Gamma_{TLS}$. The distance $d_{(\mathscr{R}_n,i)}$ between point and plane can be computed by

$$d_{(\mathscr{R}_n,i)} = \frac{|\boldsymbol{X_R}\overrightarrow{\boldsymbol{X}}_{(\boldsymbol{\Gamma},\boldsymbol{a})} \cdot \vec{\boldsymbol{n}}|}{|\vec{\boldsymbol{n}}|} = \frac{|(\widetilde{X}^W_{(\mathscr{R}_n,i)} - \widetilde{X}_{(\Gamma,a)}) \cdot ((\widetilde{X}_{(\Gamma,a)} - \widetilde{X}_{(\Gamma,b)}) \times (\widetilde{X}_{(\Gamma,a)} - \widetilde{X}_{(\Gamma,c)}))|}{|(\widetilde{X}_{(\Gamma,a)} - \widetilde{X}_{(\Gamma,b)}) \times (\widetilde{X}_{(\Gamma,a)} - \widetilde{X}_{(\Gamma,c)})|} \tag{2}$$

where $\vec{\boldsymbol{n}}$ is the normal vector of the plane. For each real keypoint, when $d_{(\mathscr{R}_n,i)}$ tends toward zero, the relationship of the corresponding pair will be more stable.

3.2.3. Motion estimation

This step is to achieve registration using the corresponding pairs built by the virtual features and the real features. Let T^W_n be the transformation vector between the MLS point cloud $\widetilde{M}^w_n$ and the reference Γ_{TLS}, where T^W_n contains rigid motion in 6 degrees of freedom (DOF), i.e.,

$$\boldsymbol{T}^W_n = [\omega, \varphi, \kappa, t_x, t_y, t_z]$$

where ω, φ and κ are rotations around the x-, y-, and z-axes of $\{W\}$, respectively, following the right hand rule, and t_x, t_y and t_z are translations of $\{W\}$ along the x-, y- and z-axes, respectively. Let us assume that $\widetilde{X}^W_{(n,i)}$ are the feature points from $\widetilde{\mathscr{V}}^W_n$ and $\widetilde{\mathscr{R}}^W_n$, where the features are extracted from the MLS point cloud, and $X^W_{(n,i)}$ are the transformation results of $\widetilde{X}^W_{(n,i)}$. To estimate the accurate motion of the MLS system, a rigid transformation relationship between $X^W_{(n,i)}$ and $\widetilde{X}^W_{(n,i)}$ can be established:

$$X^W_{(n,i)} = \boldsymbol{R}\widetilde{X}^W_{(n,i)} + \boldsymbol{T}^W_n(1{:}\ 3) \tag{3}$$

where $\boldsymbol{R}$ is the rotation matrix ($\boldsymbol{R} \in \mathbb{R}^{3\times3}$). In this paper, we consider the y-axis as the principal axis and calculate $\boldsymbol{R}$ by rotating around the y-x-z axis.

$$\boldsymbol{R} = R_y(\varphi)R_x(\omega)R_z(\kappa) = \begin{bmatrix} \cos\varphi\cos\kappa - \sin\varphi\sin\omega\sin\kappa & -\cos\varphi\sin\kappa - \sin\varphi\sin\omega\cos\kappa & -\sin\varphi\cos\omega \\ \cos\omega\sin\kappa & \cos\omega\cos\kappa & -\sin\omega \\ \sin\varphi\cos\kappa + \cos\varphi\sin\omega\sin\kappa & -\sin\varphi\sin\kappa + \cos\varphi\sin\omega\cos\kappa & \cos\varphi\cos\omega \end{bmatrix}$$

From Eq. (1), we can derive a geometric relationship between each virtual keypoint in the MLS data and the corresponding point in the reference:

$$\mathbf{f}_{\mathscr{V}}(\widetilde{X}^W_{(n,i)}) = d_{\mathscr{V}} - r_{(\mathscr{V}_n,i)},\ r_{(\mathscr{V}_n,i)} \in r_{\mathscr{V}_n},\ \widetilde{X}^W_{(n,i)} \in \widetilde{\mathscr{V}}^W_n \tag{4}$$

where $r_{(\mathscr{V}_n,i)}$ is the corresponding radius of virtual keypoint $\widetilde{X}^W_{(n,i)}$.

Similarly, from Eq. (2), we can derive a geometric relationship between each real keypoint in the MLS data and the corresponding planar patch in the reference:

$$\mathbf{f}_{\mathscr{R}}(\widetilde{X}^W_{(n,i)}) = d_{\mathscr{R}},\ \widetilde{X}^W_{(n,i)} \in \widetilde{\mathscr{R}}^W_n \tag{5}$$

Combining Eq. (4) and Eq. (5), a nonlinear function about $\boldsymbol{T}^W_n$ can be established:

$$\mathbf{f}(\boldsymbol{T}^W_n) = d = \sum \mathbf{f}_{\mathscr{V}}(\widetilde{X}^W_{(n,i)}) + \sum \mathbf{f}_{\mathscr{R}}(\widetilde{X}^W_{(n,i)}) \to 0 \tag{6}$$

where each row of $\mathbf{f}$ corresponds to a keypoint and d represents the distance between the keypoint and its corresponding feature. Finally, we can solve Eq. (6) through nonlinear iterations by minimizing the error e toward zero with the Levenberg-Marquardt (L-M) method:

$$e = \underset{e}{arg\min}\frac{1}{2}\sum_{i=1}^{N}\|d_i - 0\|^2 = \underset{e}{arg\min}\frac{1}{2}\mathbf{f}(\boldsymbol{T}^W_n)^T\mathbf{f}(\boldsymbol{T}^W_n) \tag{7}$$

First, we linearize Eq. (7) with the first-order approximation of a Taylor expansion:

$$\mathbf{f}(\boldsymbol{T}^W_n) = \mathbf{f}(\widehat{\boldsymbol{T}}^W_n + \Delta\boldsymbol{T}) = \mathbf{f}(\widehat{\boldsymbol{T}}^W_n) + \boldsymbol{J}\Delta\boldsymbol{T} \tag{8}$$

where $\widehat{\boldsymbol{T}}^W_n$ is the initial motion in 6-DOF and $\Delta\boldsymbol{T}$ is the correction of the initial motion. In this paper, 6-DOF T^W_n, i.e., the rotations (ω, φ, κ) and the translations (t_x, t_y, t_z), are regarded as the unknowns, and $\boldsymbol{J}$ is the Jacobian matrix of $\mathbf{f}(\cdot)$ and can be calculated by combining Eq. (3) and Eq. (6):

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial\mathbf{f}}{\partial\omega} & \frac{\partial\mathbf{f}}{\partial\varphi} & \frac{\partial\mathbf{f}}{\partial\kappa} & \frac{\partial\mathbf{f}}{\partial t_x} & \frac{\partial\mathbf{f}}{\partial t_y} & \frac{\partial\mathbf{f}}{\partial t_z} \end{bmatrix} \tag{9}$$

By combining Eq. (3) and Eq. (6), we derive every element of matrix $\boldsymbol{J}$. Taking a virtual feature point (x, y, z) as an example to describe the derivation of ω,

$$\frac{\partial\mathbf{f}}{\partial\omega} = \frac{\partial\mathbf{f}_{\mathscr{V}}}{\partial\mathbf{X}^W_{(n,i)}} \cdot \frac{\partial\mathbf{X}^W_{(n,i)}}{\partial\omega} = \frac{\partial\mathbf{f}_{\mathscr{V}}}{\partial x} \cdot \frac{\partial x}{\partial\omega} + \frac{\partial\mathbf{f}_{\mathscr{V}}}{\partial y} \cdot \frac{\partial y}{\partial\omega} + \frac{\partial\mathbf{f}_{\mathscr{V}}}{\partial z} \cdot \frac{\partial z}{\partial\omega} =$$

$$\frac{x - x_\Gamma}{d_v} \cdot \left(\frac{\partial\boldsymbol{R}_{11}}{\partial\omega} \cdot x + \frac{\partial\boldsymbol{R}_{12}}{\partial\omega} \cdot y + \frac{\partial\boldsymbol{R}_{13}}{\partial\omega} \cdot z\right) + \frac{y - y_\Gamma}{d_v} \cdot \left(\frac{\partial\boldsymbol{R}_{21}}{\partial\omega} \cdot x + \frac{\partial\boldsymbol{R}_{22}}{\partial\omega} \cdot y + \frac{\partial\boldsymbol{R}_{23}}{\partial\omega} \cdot z\right) +$$

$$\frac{z - z_\Gamma}{d_v} \cdot \left(\frac{\partial\boldsymbol{R}_{31}}{\partial\omega} \cdot x + \frac{\partial\boldsymbol{R}_{32}}{\partial\omega} \cdot y + \frac{\partial\boldsymbol{R}_{33}}{\partial\omega} \cdot z\right)$$

where (x_Γ, y_Γ, z_Γ) is the closest neighboring point of (x, y, z) in the single-scan TLS point cloud, d_v is the distance between (x, y, z) and (x_Γ, y_Γ, z_Γ), and $\boldsymbol{R}_{ij}$ represents the element in rotation matrix $\boldsymbol{R}$. Similarly, we can derive all the other elements of matrix $\boldsymbol{J}$. For the virtual feature, other elements of matrix $\boldsymbol{J}$ are shown in Appendix A, and for the real feature, the elements of matrix $\boldsymbol{J}$ are shown in Appendix B. In addition, according to the virtual and real features, we can construct the Jacobian matrix $\boldsymbol{J}$ with a size of $M \times 6$, where M denotes the sum of the number of virtual features and the number of real features. By deriving Eq. (7) with 0, then combining Eq. (8) and Eq. (9), the correction $\Delta\boldsymbol{T}$ can be solved by

$$\Delta\boldsymbol{T} = (\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J} + \lambda\boldsymbol{I})^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{d} \tag{10}$$

where λ is the damping factor determined by the L-M method. Then, fine motion can be calculated by

$$\boldsymbol{T}^W_n = \widehat{\boldsymbol{T}}^W_n + \Delta\boldsymbol{T} \tag{11}$$

where the MLS point cloud can be transformed into the common coordinate system based on $\boldsymbol{T}^W_n$.

3.3. Global optimization

Global optimization mitigates the accumulative error and transforms all point clouds into a common coordinate system (Shao et al., 2019). The challenge of global optimization is solving the spatial inconsistency problem (Liang et al., 2018a). Therefore, we adopt a method based on a global map that does not consider loop closure and the adjustment of all of the data to address the global optimization of SLAM. In this paper, the single-scan TLS point cloud is used as a reference, and a global map combining the reference and incremental MLS point clouds provides a constraint for forest mapping. The detailed schematic is shown in Fig. 5.

Let M_k (k = 1, 2, 3 … n-1, n) represent an MLS point cloud at the time of sweep k, where forest mapping starts from point cloud M_1. For M_1, we set prior information Γ_{TLS} as the reference and optimize the transformed M_1 using the L-M method. Then, we can obtain the motion $\boldsymbol{T}^w_1$ of MLS data at the time of sweep 1 and transform M_1 into the common coordinate system $\{W\}$. Simultaneously, we build the global map $G_{\Gamma+M_1}$ by combining Γ_{TLS} and M_1 and reset $G_{\Gamma+M_1}$ as the global

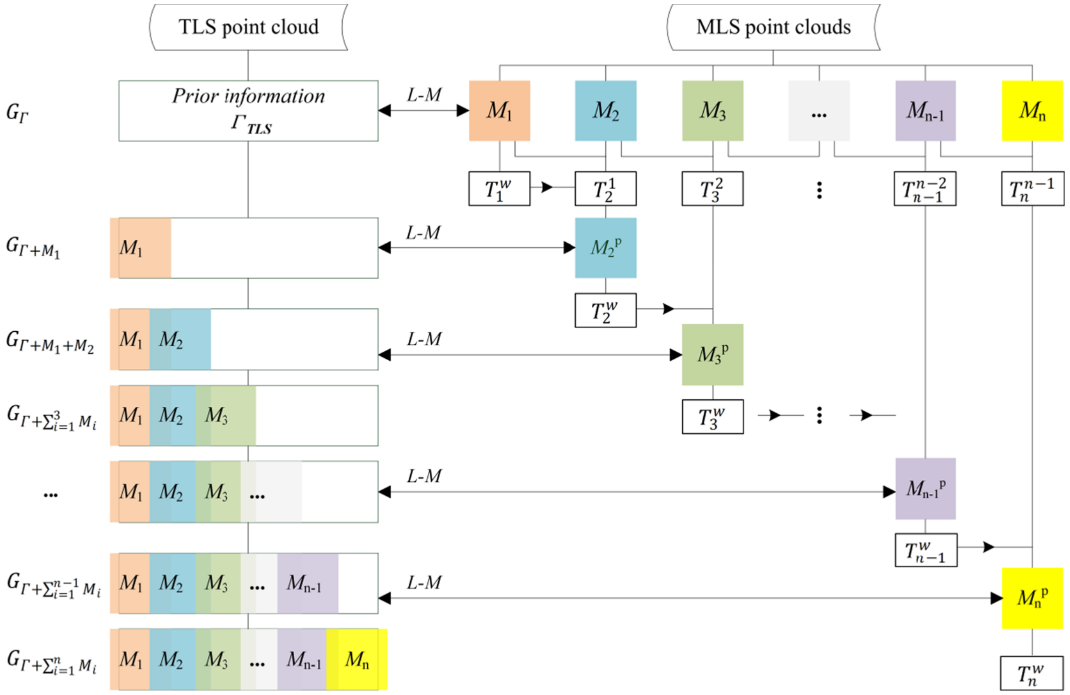


Fig. 5. Schematic of global optimization. $\{M_1, M_2, M_3 \cdots M_n\}$ with different colors represent the MLS point clouds at different times. T_i^w represents the motion of MLS data at the time of sweep i, and M_i^p represents an initial transformed MLS point cloud. $G_{\Gamma+\sum_{i=1}^{n} M_i}$ represents the global incremental map.

constraint.

For the MLS point cloud M_2 at the time of sweep 2, we calculate the transformation $\boldsymbol{T}_2^1$ between M_1 and M_2 using 3D-NDT; then, an initial transformation $\widehat{\boldsymbol{T}}_2^W$ between M_2 and the global map $G_{\Gamma+M_1}$ can be calculated based on $\boldsymbol{T}_1^w$ and $\boldsymbol{T}_2^1$. Furthermore, an initial transformed point cloud M_2^p from the local coordinate system of M_2 to the common coordinate system $\{W\}$ can be obtained based on $\widehat{\boldsymbol{T}}_2^W$. Because of the propagation of error, we optimize the initial transformation $\widehat{\boldsymbol{T}}_2^W$ and the transformed point cloud M_2^p based on the global map $G_{\Gamma+M_1}$ using the L-M method. In the process, the corresponding points of virtual keypoints and the corresponding patches of real keypoints are extracted from the global map $G_{\Gamma+M_1}$. According to the L-M method, we can obtain an optimized transformation $\boldsymbol{T}_2^w$ and a point cloud M_2^w. Simultaneously, we rebuild a new global map $G_{\Gamma+M_1+M_2}$ by combining $G_{\Gamma+M_1}$ and M_2^w. Similar to the MLS point cloud M_2, we can estimate the motion $\boldsymbol{T}_i^w$ of the subsequent MLS point clouds and transform each MLS point cloud M_i into a global map and rebuild a new cumulative global map $G_{\Gamma+\sum_{i=1}^{n} M_i}$. We locate each MLS point cloud in the common coordinate system $\{W\}$ and simultaneously map the environment by combining the MLS data and the single-scan TLS data.

In practice, some MLS point clouds affected by motion distortion and measurement error reduce the accuracy of forest mapping. Therefore, we detect the key MLS point cloud by determining the relationship between the virtual keypoints and their corresponding points (Eqs. (1) and (4)). If the difference between the points in the corresponding pair is less than the set threshold, we add the current point cloud to the global map. In this paper, the threshold is set as 6 cm, which is twice the measurement precision (i.e., 3 cm).

4. Results and discussion

4.1. Study area and data collection

The study area, located in Saihanba National Forest Park in Hebei Province in northern China, is dominated by coniferous trees. The study area includes dozens of square sample plots of size 25–30 m, and the tree species is larch. For this study, we selected three datasets from two sample plots: Plot A and Plot B (Fig. 6). In Plot A, we acquired two datasets of point clouds at different periods: Dataset A1 and Dataset A2. In addition, one dataset was acquired in Plot B, i.e., Dataset B. The parameters of the two plots are summarized in Table 1.

The MLS data were captured by cross-moving around the forest plots using the Velodyne VLP-16 laser scanning system. Its angular resolution is 2° in the vertical direction, and the scan frequency was set to 10 Hz. We obtained 1918 MLS point clouds in Dataset A1, 1395 point clouds in Dataset A2, and 1045 point clouds in Dataset B. The TLS data were captured using the Riegl VZ-1000 laser scanning system, and the scan angular resolution was 0.03° in both the horizontal and vertical directions. Table 2 gives the parameters of the two scanners.

The TLS system was placed in the middle of the forest plot to acquire point clouds with a full field-of-view scan; the multi-scan TLS mode was also used to verify and compare methods. In addition, the tree position is represented by extracting the center of the cross section of the point cloud above ground level and is used to evaluate the accuracy of the forest mapping results.

4.2. Evaluation of feature extraction

The virtual feature is represented by the center of the tree stem cross section, so we used the radius that corresponds to the center to assess the virtual feature. Specifically, the deviation between the radius from the MLS data and its corresponding radius from multi-scan TLS data was calculated and used for the evaluation of the extracted virtual features. Fig. 7 shows the results of feature extraction from one frame MLS point cloud.

To ensure the reliability and correctness of the virtual features, strong constraints are used for feature extraction. For example, a virtual feature point is discarded if its radius is significantly different from the other feature points on the same tree. Therefore, although this method was unable to detect all trees in the MLS data, the virtual features from the extracted trees with dense points could be correctly extracted (Fig. 7 (a)). Fig. 7 (b) shows that the radius deviations of the virtual features are small, e.g., the radius deviations of almost all virtual features stay within −0.03 m to 0.03 m. In addition, the mean absolute deviation

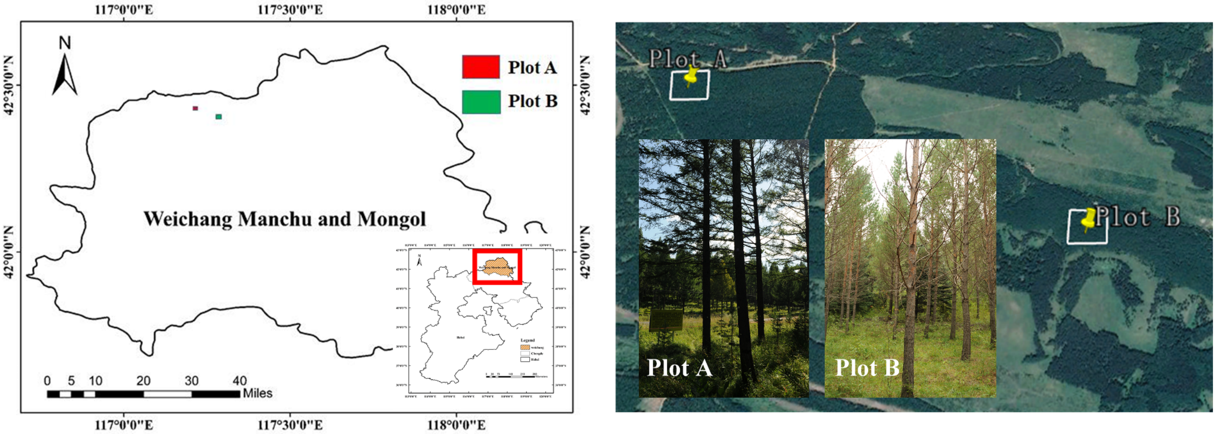


Fig. 6. Study area.

(mean) and the root mean square error (RMSE) values are less than 0.02 m, which indicates the reliability of the virtual feature points. Meanwhile, as iterative optimization, some real features with large errors were discarded (Fig. 7 (c)), and accurate correspondences were generated (Fig. 7 (d)). Overall, the radius deviations of most virtual feature points are small, and the accurate corresponding planes of real features are obtained, so the extracted features suggest the potential for accurate scan matching.

4.3. Evaluation of corresponding pairs

The corresponding pairs were composed of the feature points from each frame MLS point cloud and their nearest neighbor points from the single-scan TLS point cloud, of which the distance deviation between the pairs was regarded as a constraint for motion estimation of the MLS system. Therefore, we used Eq. (1) and Eq. (2) to quantitatively evaluate the performance of the corresponding pairs, of which the deviation between the value of Eq. (1) and the corresponding radius and the value of Eq. (2) were regarded as evaluation indices for evaluating the corresponding pairs of the real features and the virtual features, respectively (Fig. 8).

Because the cross section of the tree stem is not strictly circular, the distance between the center of the circle and its closest point on the stem is generally less than the corresponding radius. Therefore, due to the error of an initial transformation, the distances between the virtual feature points and their closest points on the tree stems of reference are usually less than their radius, and the deviations between the distances and the corresponding radius are large (see Fig. 8 (a)). As the number of iterations during the optimization increases, the corresponding pairs will change, and the deviations will decrease, in which the virtual feature points with large deviations are removed (see Fig. 8 (b)). For example, the deviations of most virtual features vary between −0.1 m and −0.03 m, and the mean and RMSE values are 0.057 m and 0.064 m, respectively (Fig. 8 (a)), which indicate inaccurate scan matching in the horizontal direction. From Fig. 8 (c), the distances of most real features vary between 0.0 m and 0.14 m, and the mean and RMSE values are 0.069 m and 0.091 m, respectively. Apparently, with an initial pose of the MLS system, the corresponding pairs derived from the MLS and single-scan TLS data have difficulty achieving accurate estimation of MLS motion. Therefore, nonlinear optimization is used to update the corresponding pairs. In theory, if accurate corresponding pairs can be generated at the end of the iterations, the deviation value will be small and tend to zero. For example, the deviations of most virtual features vary between −0.03 m and 0 m, and the mean and RMSE values drop to 0.020 m and 0.024 m, respectively (see Fig. 8 (b)), of which several factors, such as the inaccurate features and the non-circular cross sections of the tree stems, led to large deviations in the results. For the real features, the distances mainly vary between 0.0 m and 0.05 m, and the mean and RMSE values drop to 0.024 m and 0.03 m, respectively (see Fig. 8 (d)). In addition, because some inaccurate corresponding pairs were discarded in the process of nonlinear optimization, the total number of features decreased. In general, the evaluation results showed certain reliability of the corresponding pairs, especially the virtual feature points and their corresponding points, which also indicated a possibility for achieving accurate motion estimation of the MLS system.

Table 2
Parameters of the laser scanners used.

Scanner	MLS (Velodyne VLP-16)	TLS (Riegl VZ-1000)
Channels	16	1
Range (max.)	~100 m	~1400 m
Measurement rate (max.)	600,000 points/sec	122,000 points/sec
Field of view	360° × 30°	360° × 100°
(horizontal × vertical)	(+15°/−15°)	(+60°/−40°)
Measurement precision	± 3 cm	± 0.5 cm

4.4. Forest mapping and positioning accuracy

In the proposed method, two results can be obtained: localization and mapping. The localization is represented by the trajectory of the MLS system, and the mapping is represented by the reconstruction of the forest plots. To evaluate the effectiveness and robustness of the

Table 1
Parameters of the three datasets in the two plots.

Dataset	Stem density (stems/ha)	Tree height (m)				DBH (m)			
		Max	Min	Mean	SD	Max	Min	Mean	SD
A1	302	21.563	18.631	19.852	0.823	0.310	0.243	0.271	0.019
A2	302	21.602	18.714	20.097	0.733	0.349	0.258	0.291	0.028
B	415	21.012	18.431	19.674	0.801	0.302	0.248	0.279	0.018

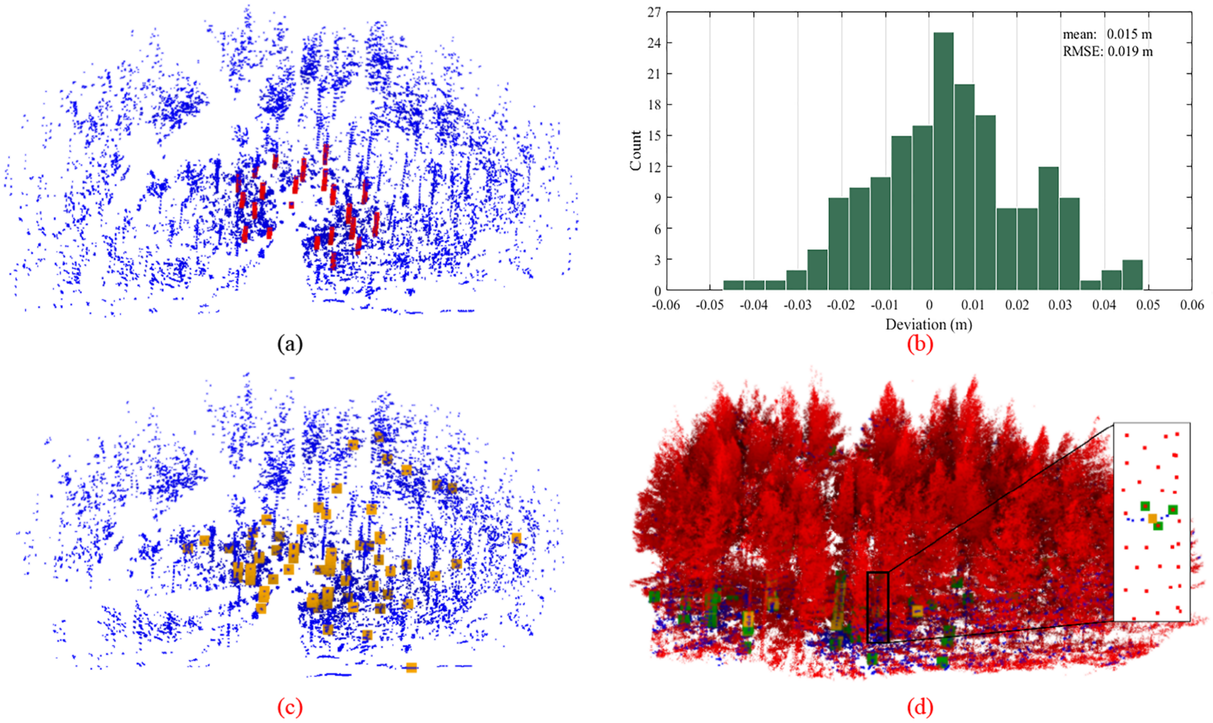


Fig. 7. Feature extraction results (blue points are the MLS data). (a) The result of virtual feature extraction (red points are the virtual feature points). (b) The distribution of the radius deviation; the x-axis represents the deviations, and the y-axis represents the number of virtual features in different ranges. (c) Shows the real features used for motion estimation (orange points are the real features). (d) shows the result of scan matching (red points are the single-scan TLS data, and green points are the correspondences of real features). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

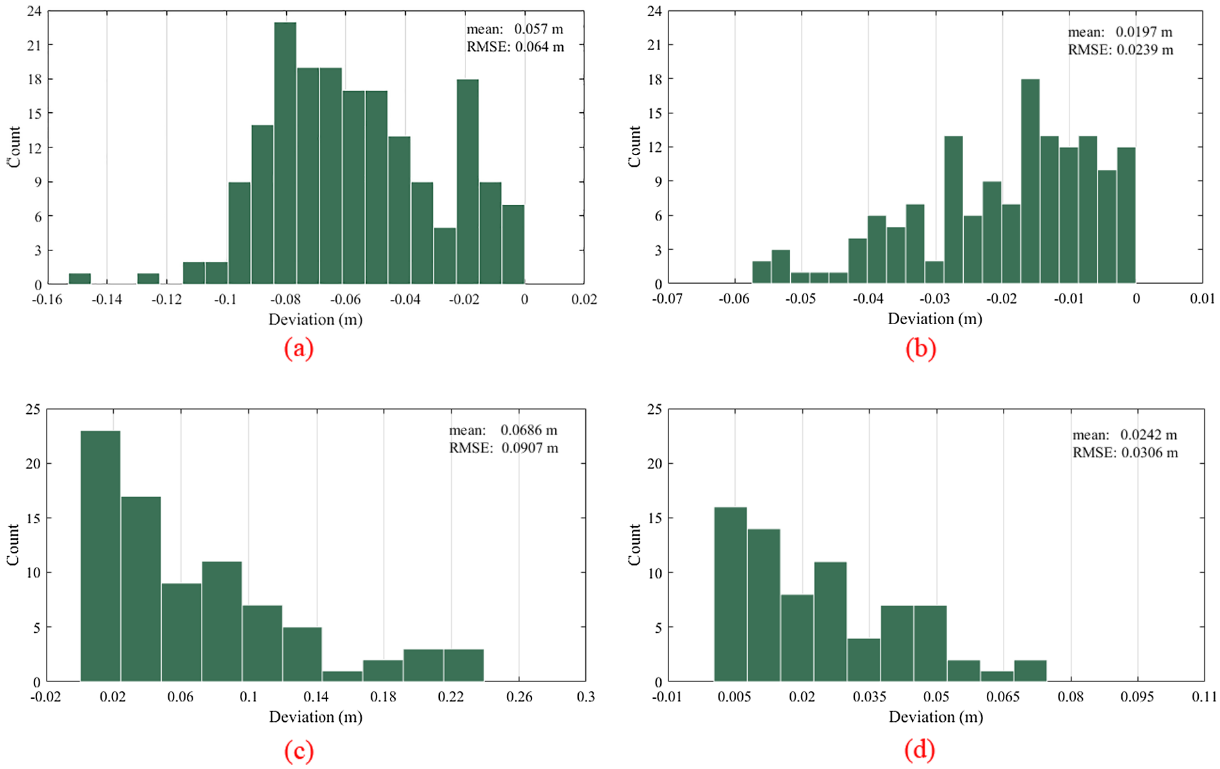


Fig. 8. Evaluation of the corresponding pairs. The x-axis denotes the deviations, and the y-axis denotes the count of features in different deviation ranges. (a) and (b) Represent the relation between the virtual features in the MLS data and their corresponding features in the single-scan TLS data before and after nonlinear optimization, respectively. (c) and (d) Represent the distance between the real features in the MLS data and their corresponding features in the single-TLS data before and after nonlinear optimization, respectively.

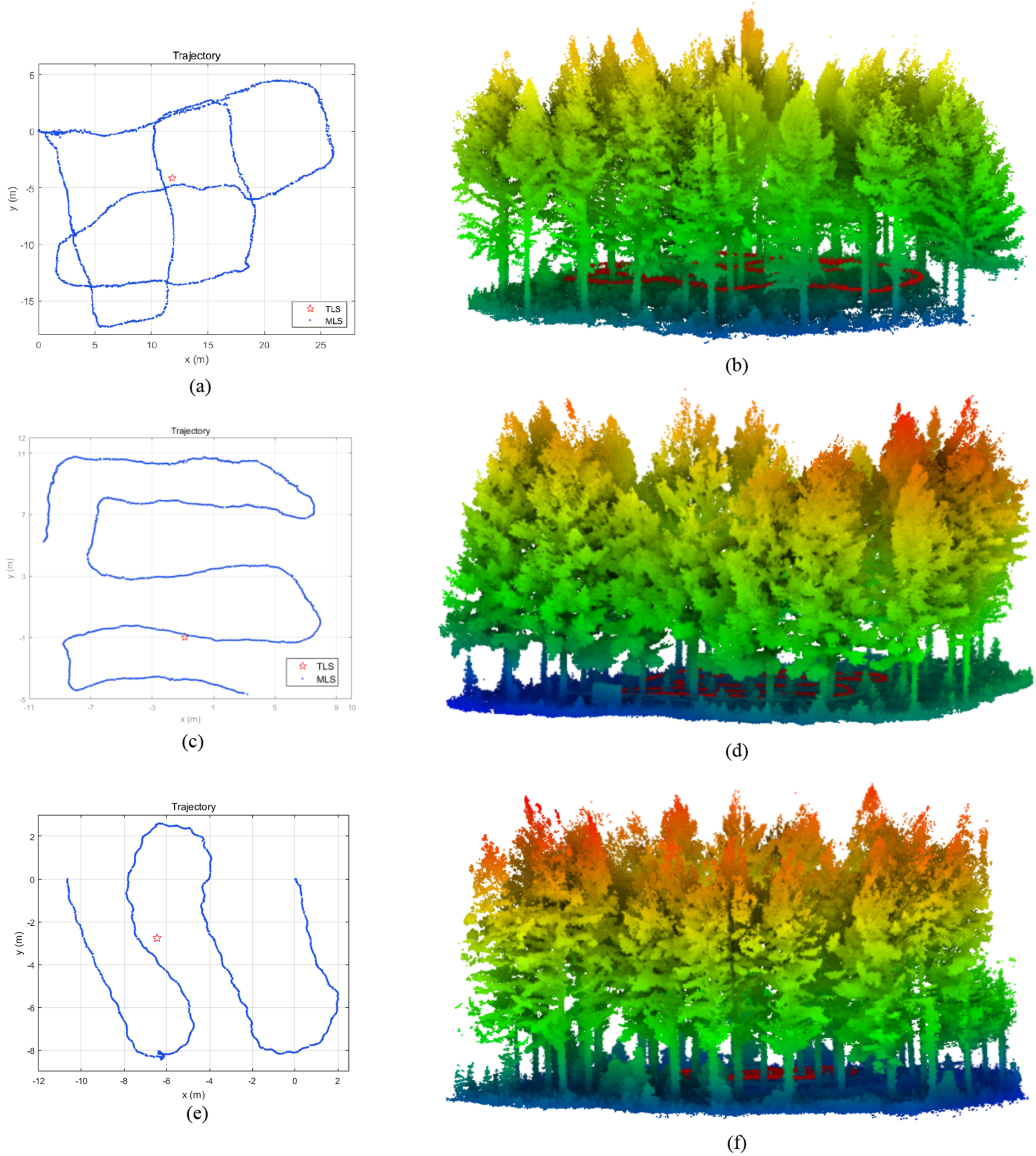


Fig. 9. Forest mapping results. (a) and (b) represent the mapping of Dataset A1, (c) and (d) represent the mapping of Dataset A2, and (e) and (f) represent the mapping of Dataset B. (a), (c), and (e) Represent the trajectories of the MLS system in the three datasets (blue points are the positions of the MLS system, and red stars represent the scan locations of the single-scan TLS). (b), (d), and (f) show the reconstruction results of the three datasets. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

proposed method in forests, we tested two forest plots in which the movements of the MLS system had different trajectories (see Fig. 9).

The results in Fig. 9 (a), (c), and (e) are coincident with the practical movements of the MLS system. Fig. 9 (b), (d), and (f) show that the forest plots can be reconstructed well, and the distributions and shapes of individual trees reconstructed by the proposed method are clear and identifiable. In addition, the proposed method is implemented on a computer with an Intel® Core™ i7-3520 M CPU @ 2.90 GHz and with 8.00 GB RAM, and forest plot mapping is achieved by offline processing, in which the runtimes of the proposed method are approximately 100 min, 70 min, and 50 min for the three datasets, i.e., approximately three seconds for one frame MLS point cloud.

In the reference, the locations of trees were available for evaluation of planimetric accuracy, so to quantitatively evaluate the performance

Table 3
Tree position accuracy. The ‘Trees’ column gives the number N_T of the detected trees in each dataset. Tree position deviation is calculated as the planimetric distance between the detected tree and its reference in multi-scan TLS data.

Dataset	Trees N_T	Tree position deviations (m)		
		Mean	RMSE	Max
A1	16	0.013	0.017	0.031
A2	14	0.018	0.020	0.029
B	14	0.012	0.014	0.023

of the proposed method, the locations of trees obtained from the proposed method were compared to those from the multi-scan TLS data. First, each forest mapping result was manually registered with its corresponding multi-scan TLS data. Then, 16 trees, 14 trees, and 14 trees were extracted from the three datasets to evaluate the mapping results. The accuracy results of the tree position are summarized in Table 3.

In the three datasets, both the mean and RMSE values varied between 0.01 m and 0.02 m. The results in Table 3 reveal the small tree position deviation values, which indicated accurate forest plot mapping results. Combining the results in Fig. 9, the location and orientation of the optimized data do not drift from their correct values during global optimization, e.g., the trajectory of the MLS system can be closed without loop-closure detection in Dataset A1 (see Fig. 9 (a)), and the open trajectories in Dataset A2 and Dataset B can still be accurately recovered by the proposed method (Fig. 9 (b) and (c)). Therefore, the reference, i.e., single-scan TLS data, can provide an effective consistency constraint for MLS-based forest plot mapping, of which these virtual features retain the location accuracies of trees in the horizontal direction.

In addition to the evaluation of planimetric accuracy, we compared some feature points from MLS data with their corresponding points in multi-scan TLS data to reflect the mapping accuracy in the vertical direction. The tree branch position deviations in the vertical direction were calculated for the evaluation of mapping accuracy in this paper. In practice, 15 feature points on the branches were evenly selected from each of the three datasets to evaluate the vertical accuracy. The accuracy results are shown in Table 4.

As seen in Table 4, the deviations were at the centimeter level in the three datasets, of which the mean absolute deviations varied between 0.015 m and 0.02 m, the RMSE values were approximately 0.02 m, and the maximum deviations were 0.04 m or less. Because of the constraints from the ground and canopy, especially the ground, the NDT algorithm could maintain the forest mapping accuracy in the vertical direction; the real features further optimized the vertical accuracy using evenly distributed points in forests, and the proposed optimization framework provided certain global consistency constraints for MLS-based forest plot mapping. In forest measurements, the requirement of accuracy in the vertical direction is generally lower than that in the horizontal direction (e.g., tree height measurement), which is at the centimeter or decimeter level. Thus, the results in Table 4 indicate highly accurate mapping results in the vertical direction. Overall, the results from Fig. 9, Table 3, and Table 4, suggest the high reliability and robustness of the proposed method.

Table 4
Vertical accuracy. The ‘Points’ column gives the number N_P of the selected sample points on branches of each dataset. The deviation is calculated as the vertical distance between the sample point and its reference in multi-scan TLS data.

Dataset	Points N_P	Vertical deviations (m)		
		Mean	RMSE	Max
A1	15	0.019	0.021	0.040
A2	15	0.016	0.019	0.036
B	15	0.017	0.019	0.037

4.5. Data completeness and performance

The completeness of the structural information is an important basis on which to select a data-acquisition method. The paper combined the single-scan TLS and MLS data to reconstruct the forest plots so that the MLS data could offer compensation for the omission of single-scan TLS data. Therefore, we analyzed and compared the data completeness on the plot scale by the proposed method with the single-scan TLS data. The results in Dataset A1 are shown in Fig. 10.

As shown in Fig. 10 (a), due to the occlusion effects from the other objects, there are some regions that cannot be scanned by the single-scan TLS at the forest plot scale, such as the sector regions labeled by the green triangles. According to the proposed method, these omitted regions can be filled by the MLS point clouds (Fig. 10 (b)). Thus, the proposed method can obtain more complete structural information of the forest. In addition, ground-based LiDAR systems are mainly used for data acquisition below the canopy, of which the tree stem is one of the significant tree-level attributes that is being widely studied. Therefore, we compared the proposed method with the single-scan TLS method and multi-scan TLS method from the completeness of individual tree stem data in Dataset A1. Fig. 11 shows a comparison of the three data-acquisition methods.

From the side view and the cross section of the stem in Fig. 11 (a), only one side of the tree stem can be scanned by the single-scan TLS method because the laser cannot penetrate the tree stem. Fig. 11 (b) shows that the omission of tree stems caused by one scan (see Fig. 11 (a)) can be compensated for by the MLS data of the proposed method. Fig. 11 (c) shows that the proposed method and the multi-scan TLS method acquired a complete tree stem structure. Nevertheless, the structural information above the canopy acquired by the multi-scan TLS method is more complete than that of the proposed method. Furthermore, due to the limitation of the field of view, a single MLS scanner can only achieve stem mapping using the proposed method, and the canopy structural information is generally limited. Consequently, the tree attributes related to tree height are difficult to estimate accurately. In contrast, the tree attributes related to the DBH can be obtained because of the complete stem. Therefore, to evaluate the performance of the MLS data, we calculated the DBH and stem curve values.

The DBH is an important structural parameter in forestry inventories and can be used to analyze tree growth. Therefore, we calculated the DBH values in the three datasets and compared the results with those from multi-scan TLS data. The DBH value is determined by extracting a cross section of the stem that falls between 1.2 m and 1.4 m above ground level. Thus, we first filtered the ground and nonground points (Zhang et al., 2016b) and extracted points that represented the tree stem at breast height from the nonground points and then used the least square method to fit a circle. To evaluate the accuracy of the DBH fitted by the proposed method, the DBH fitted by multi-scan TLS data is regarded as the reference and the DBH deviation is calculated (Table 5).

In the results of Table 5, the mean absolute deviations and the RMSE values were approximately 0.01 m, and the maximum deviations were approximately 0.02 m. In general, the overall accuracies could reach more than 90%. In addition, the mean bias was 0.001 m in Dataset A1. In Dataset A2 and Dataset B, the average DBH values from the proposed method were less than those from multi-scan TLS data, and the bias values were approximately −0.01 m. Overall, accurate DBH values can be obtained from the mapping results of the proposed method.

In forest measurements, the stem curve is usually used to describe the shape of tree stems and consists of stem diameters from specific tree heights. Therefore, we compared the stem curve from the MLS data to its corresponding stem curve from the multi-scan TLS data, of which the stem curve is represented by several diameters at different heights, and the corresponding diameters from the MLS and TLS data were at the

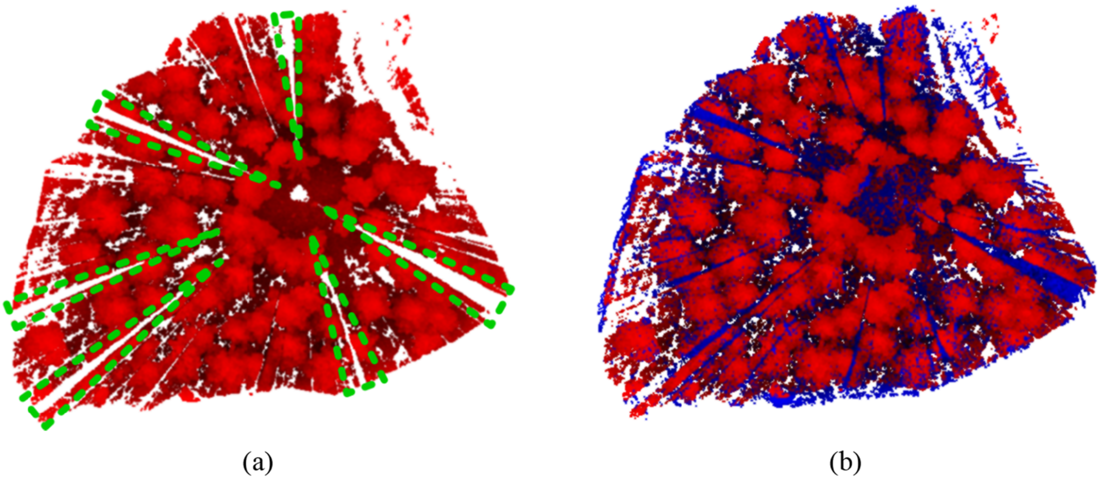


Fig. 10. Comparison between the single-scan TLS point cloud and the result of the proposed method. (a) represents the single-scan TLS point cloud (red points), and green triangles show the omitted regions. (b) represents the mapping result of the proposed method (red points are the single-scan TLS point cloud and blue points are the MLS data). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

same heights. The accuracies, including the mean absolute deviation, RMSE, and the maximum deviation, are summarized in Table 6. The mean absolute deviations and the RMSE values were at the centimeter level, approximately 0.01 m, and the maximum deviations were less than 0.03 m. The results indicated that the overall correctness of the stem curve could reach 90% in the three datasets, which suggests certain effectiveness of the mapping results from the proposed method.

4.6. Comparison of data-acquisition methods

Compared to the TLS system, the major advantage of the MLS system is the rapid acquisition of point clouds. In addition, one of the purposes of this paper is to supplement single-scan TLS data with MLS data. Thus, to evaluate the performance of the proposed method, we compared it with the single-scan TLS and multi-scan TLS modes from the time of the field measurement and the completeness of individual tree stems in Dataset A1 (Table 7).

At the time of the field measurement, the single-scan method requires the least amount of time of the three methods. To ensure an adequate precision and detection rate, we took approximately 20 min

Table 5
The accuracy of DBH. The ‘Trees’ column gives the number N_T of the extracted trees in each dataset. DBH deviation is the difference between the DBH from the proposed method and the DBH from multi-scan TLS data.

Dataset	Trees N_T	DBH deviations (m)			
		Mean	RMSE	Max	Bias
A1	16	0.009	0.010	0.021	0.001
A2	14	0.011	0.013	0.023	−0.008
B	14	0.010	0.011	0.019	−0.009

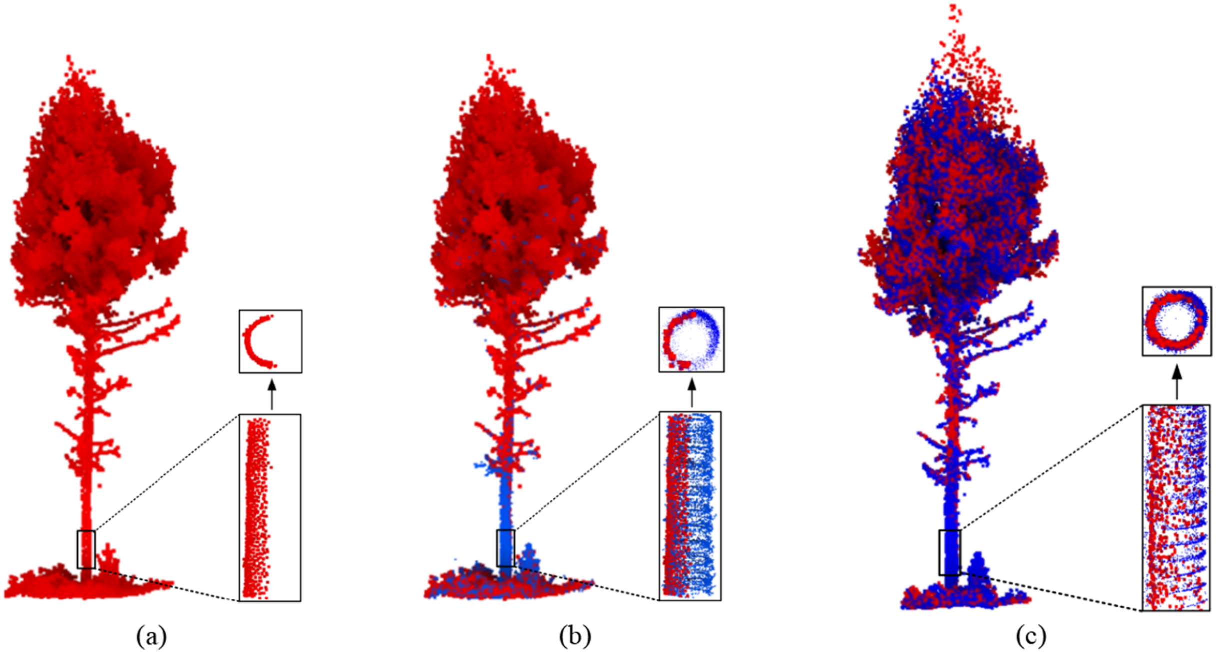


Fig. 11. Comparison of individual tree mapping. (a) Tree mapping by the single-scan mode. (b) Comparison between the single-scan TLS mode (red points) and the proposed method (blue points). (c) Comparison between the multi-scan TLS mode (red points) and the proposed method (blue points). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
The accuracy of the stem curve. The 'Stems' column gives the number N_S of the detected stems in each dataset. The stem curve is calculated by averaging the diameters at different heights.

Dataset	Stems N_S	Stem curve deviations (m)		
		Mean	RMSE	Max
A1	16	0.011	0.012	0.027
A2	14	0.010	0.011	0.020
B	14	0.010	0.011	0.019

Table 7
The completeness of individual stem data and the time of the field measurement in Dataset A1. The "Criteria" column lists two evaluation criteria, and "%" represents the point cloud coverage rate.

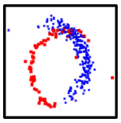
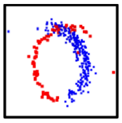
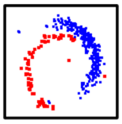
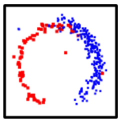
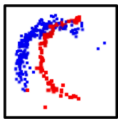
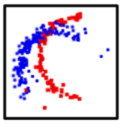
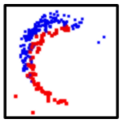
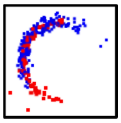
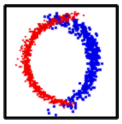
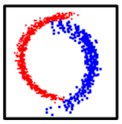
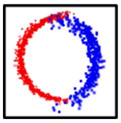
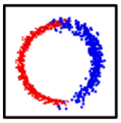
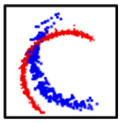
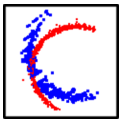
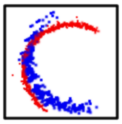
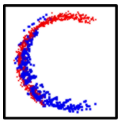
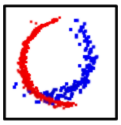
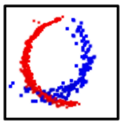
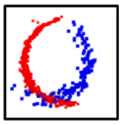
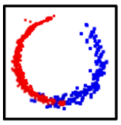
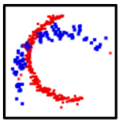
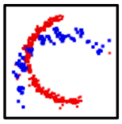
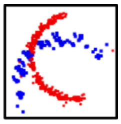
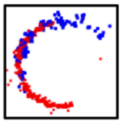
Criteria	Data-acquisition methods		
	Single-scan TLS	Multi-scan TLS	The proposed method
Time of the field measurement (min)	≈**20 min**	≈210 min	≈25 min
Completeness of individual tree stems (%)	< 50%	≈**100%**	≈**100%**

for the field measurement: selection of scan position and set up of the scanner took 5–10 min, and a full field-of-view scan in fine-scan mode took approximately 10 min. For the proposed method, the single-scan TLS and MLS devices were combined for data acquisition. In addition to single-scan TLS, the MLS took approximately 5 min to acquire point clouds in a forest plot, including planning trajectory and scanning. In total, the proposed method took approximately 25 min for plot A. In practice, the TLS system of the proposed method can be directly placed at the middle of the forest plot and acquires a point cloud with a coarse-scan mode, and we do not consider the detection rate and the precision of trees in the placement of the TLS system, which only takes 5–10 min to set up and scan the plot. In theory, therefore, we only need approximately 15 min to scan plot A. On the other hand, the multi-scan TLS method used five scan positions to scan plot A with fine-scan mode and placed some reflective targets in the plot for point cloud registration. In complex forests, these placed reflective targets need to be scanned from different scan positions so that the set of these reflective targets generally takes at least one hour. In each scan position, TLS performed horizontal and vertical scans and needed to find and extract reflective targets, which took approximately half an hour. As a result, the method took approximately 210 min to field-measure one forest plot. Therefore, single-scan TLS and the proposed method are more efficient (see Table 7).

In practice, although the single-scan TLS method offers low detection to the complete forest scenario, the method is capable of achieving stem detection with approximately 75% completeness and 90% correctness in easy forest stands (Liang et al., 2018b). Nevertheless, due to the occlusions from a stem and its nearby stems, the single-scan TLS mode only scanned one side of the stem, and the mean completeness of individual stems was less than 50%. The multi-scan TLS method and the proposed method acquired more complete tree stems from multiple perspectives in plots, and their mean scan completeness of individual tree stems were approximately 100%.

In conclusion, according to the evaluation criteria, the three data-acquisition methods all have their own advantages. When the amount of time needed for field measurements is of interest, the single-scan TLS method and the proposed method are the best choices. If the completeness of the stem is needed, multi-scan TLS and the proposed method have a slight advantage over single-scan TLS. Nevertheless, if

Table 8
Cross sections of four tree stems in registration results by different methods (red points are from the TLS point cloud; blue points are from the MLS point cloud). Each line represents tree stem data.

Methods	NDT (Magnusson et al., 2007)	LOAM (Zhang and Singh, 2014)	ICP (Besl and Mckay, 1992)	The proposed method
Low-overlap in Dataset A1				
High-overlap in Dataset A1				
Low-overlap in Dataset A2				
High-overlap in Dataset A2				
Low-overlap in Dataset B				
High-overlap in Dataset B				

the completeness of tree stems and the time needed for field measurements are all considered together, the proposed method may be the best choice.

4.7. Comparison of motion estimation

To evaluate the performance and effectiveness of the proposed virtual feature, tree stem position is used to evaluate the horizontal accuracy of motion estimation. Furthermore, we compared the proposed method with three methods: the NDT method, the ICP method, and the LOAM method. Specifically, we visualized the cross sections of four tree stems in the low-overlap and high-overlap regions of the three datasets (see Table 8).

From Table 8, distinct deviations were observed in the results from the NDT, LOAM, and ICP methods, in which the deviations in the NDT and LOAM methods are similar and more obvious, and the stem points in the MLS data transformed by the two methods intersect with the reference points, especially the stems in the high-overlap regions of the three datasets. In contrast, the results of the proposed method showed that the cross sections of the tree stems in the low-overlap regions are approximately circular, and the tree stems in the MLS point cloud transformed by the proposed method in the high-overlap regions agree with the TLS point cloud. Furthermore, to quantitatively evaluate registration accuracy in the horizontal direction, we detected several easily identifiable stems to calculate the position deviations and compared the performances of the four methods. The results are summarized in Table 9.

The results show large deviations in the results from the NDT, LOAM, and ICP methods. In practice, the three methods consider the points on different sides of one tree stem to be on the same surface. The NDT and ICP methods are collectively called dense point registration methods because they consider all points in the registration. The NDT algorithm considers the distribution of all points in the overlap area, and the ICP algorithm achieves the registration based on the minimization of the distances between matching points, which are easily influenced by those points on different sides of the stems, so there are large errors in their results. The deviations in the NDT and LOAM methods were approximately equal in the three datasets (Table 9) and corresponded to the visual performances in Table 8. The LOAM method is a sparse feature method that combines line and plane features that are extracted from the object surface for motion estimation, and it generally requires a fine initial transformation between point clouds. Although only a few features are required by the method, the inaccurate corresponding pairs in the low-overlap region make the method easily fall into a local minimum. Because the initial transformation was provided by the NDT method in this study, the results of the LOAM and NDT methods were similar. The results of the proposed method show a vast overall improvement in the three datasets. The mean absolute stem position deviations between the correspondences decrease to approximately 0.015 m (in Dataset A1 and Dataset B) and even millimeters (in Dataset A2). In addition, the RMSE values and the maximum deviations drop below 0.02 m and 0.03 m, respectively.

Table 9
Stem position accuracy of different methods.

Dataset	Method	Trees N_T	Stem position deviations (m)		
			Mean	RMSE	Max
A1	NDT (Magnusson et al., 2007)	11	0.071	0.075	0.100
	LOAM (Zhang and Singh, 2014)	11	0.064	0.070	0.104
	ICP (Besl and Mckay, 1992)	11	0.060	0.065	0.102
	The proposed method	11	**0.013**	**0.014**	**0.021**
A2	NDT (Magnusson et al., 2007)	11	0.048	0.053	0.080
	LOAM (Zhang and Singh, 2014)	11	0.049	0.054	0.096
	ICP (Besl and Mckay, 1992)	11	0.033	0.034	0.055
	The proposed method	11	**0.008**	**0.010**	**0.026**
B	NDT (Magnusson et al., 2007)	10	0.041	0.044	0.057
	LOAM (Zhang and Singh, 2014)	10	0.050	0.053	0.065
	ICP (Besl and Mckay, 1992)	10	0.057	0.060	0.082
	The proposed method	10	**0.016**	**0.018**	**0.028**

Table 10
Vertical accuracy of different methods.

Dataset	Method	Points N_P	Vertical deviations (m)		
			Mean	RMSE	Max
A1	NDT (Magnusson et al., 2007)	15	0.022	0.027	0.044
	LOAM (Zhang and Singh, 2014)	15	**0.018**	0.024	0.049
	ICP (Besl and Mckay, 1992)	15	0.021	0.025	0.042
	The proposed method	15	0.018	**0.020**	**0.037**
A2	NDT (Magnusson et al., 2007)	15	0.015	0.017	**0.029**
	LOAM (Zhang and Singh, 2014)	15	0.021	0.024	0.049
	ICP (Besl and Mckay, 1992)	15	0.014	0.016	0.029
	The proposed method	15	**0.012**	**0.015**	0.035
B	NDT (Magnusson et al., 2007)	15	0.014	0.015	0.027
	LOAM (Zhang and Singh, 2014)	15	0.013	0.014	0.020
	ICP (Besl and Mckay, 1992)	15	0.013	0.014	0.022
	The proposed method	15	**0.011**	**0.012**	**0.018**

Table 10 shows the mapping accuracy of the proposed method and the NDT, ICP, and LOAM methods in the vertical direction. We selected some stable feature points from the ground and branches in the MLS point cloud and their corresponding points in the TLS data, and then the vertical distance deviations between the corresponding pairs are calculated. From Table 10, the deviations obtained by the four methods are small and approximate in the three datasets, of which the mean distance deviations and the RMSE values varied between 0.01 m and 0.03 m, and the maximum deviations were less than 0.05 m. Because the ground provides strong constraints, the NDT and ICP methods, which achieve registration based on the whole points, obtain accurate registration results in the vertical direction. The proposed method and the LOAM method achieve registration based on some features, and their initial transformations between the MLS data and the single-scan TLS data are provided by the NDT method in this paper, so after nonlinear optimization, the vertical deviations from the two methods are close to the results from the NDT method. Overall, the proposed method performs well in the three datasets.

The above results of the proposed method are more reliable than the results of the other three methods. Due to weak constraints, the registration errors of the NDT, ICP, and LOAM methods are mainly in the horizontal direction. The proposed method is highly dependent on virtual features and real features, especially virtual features. Thus, in the process of motion estimation, the virtual features effectively decrease the errors in tree stem locations that exist in the other three methods by providing a constraint between the center and the surface of the tree stem; the real features retain the overall accuracy of the LiDAR odometry.

5. Conclusions

LiDAR-based forest mapping is a significant method for obtaining precise forestry inventories. To achieve complete and fast forest mapping, this paper proposed a novel method combining the single-scan TLS and MLS systems for forest measurements. Meanwhile, comprehensive experiments were performed to verify the feasibility and effectiveness of the proposed method, and a good result was obtained. In the data acquisition phase, the proposed method acquired more complete data than the single-scan TLS method, and the efficiency is higher than that of the multi-scan TLS method. In the forest mapping phase, the single-scan TLS point cloud provided a strong global consistency constraint for the MLS-based SLAM technique and maintained accuracies without a GNSS-IMU system. Moreover, in the case without loop-closure detection, the proposed method still achieved accurate forest mapping. In addition, a new point cloud registration method for combining the virtual features and the real features specific to forest environments was proposed by the paper. The method effectively solved the inaccurate registration problem caused by insufficient overlap and inaccurate corresponding pairs, for which the virtual features played an important role in reducing the horizontal errors. Compared to other classic methods, the registration results of the proposed method were also more accurate.

As an aerial platform, the UAV-LiDAR system (ULS) can obtain structural information below the canopy that approximates that obtained by the TLS system. Moreover, the ULS is more efficient than the TLS system, and the information from above the canopy is more complete. As a result, the ULS is increasingly being used more widely in recent years. In the future, therefore, to achieve fast and complete data acquisition in large-scale forest plots, the ULS and MLS will be combined for forest mapping based on the proposed method.

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.

Acknowledgements

This work was supported by the National Natural Science Foundation of China, grant nos. 41671414 and 41971380. This work was also supported by Guangxi Natural Science Fund for Innovation Research Team (grant no. 2019JJF50001) and the Open Fund of State Key Laboratory of Remote Sensing Science (grant no.OFSLRSS201920).

Appendix A

$$\frac{\partial \mathbf{f}}{\partial \varphi} = \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial \boldsymbol{X}^{W}_{(n,i)}}\cdot\frac{\partial \boldsymbol{X}^{W}_{(n,i)}}{\partial \varphi} = \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial x}\cdot\frac{\partial x}{\partial \varphi} + \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial y}\cdot\frac{\partial y}{\partial \varphi} + \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial z}\cdot\frac{\partial z}{\partial \varphi} =$$

$$\frac{x - x_{\Gamma}}{d_v}\cdot\left(\frac{\partial \boldsymbol{R}_{11}}{\partial \varphi}\cdot x + \frac{\partial \boldsymbol{R}_{12}}{\partial \varphi}\cdot y + \frac{\partial \boldsymbol{R}_{13}}{\partial \varphi}\cdot z\right) + \frac{y - y_{\Gamma}}{d_v}\cdot\left(\frac{\partial \boldsymbol{R}_{21}}{\partial \varphi}\cdot x + \frac{\partial \boldsymbol{R}_{22}}{\partial \varphi}\cdot y + \frac{\partial \boldsymbol{R}_{23}}{\partial \varphi}\cdot z\right) +$$

$$\frac{z - z_{\Gamma}}{d_v}\cdot\left(\frac{\partial \boldsymbol{R}_{31}}{\partial \varphi}\cdot x + \frac{\partial \boldsymbol{R}_{32}}{\partial \varphi}\cdot y + \frac{\partial \boldsymbol{R}_{33}}{\partial \varphi}\cdot z\right)$$

$$\frac{\partial \mathbf{f}}{\partial \kappa} = \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial \boldsymbol{X}^{W}_{(n,i)}}\cdot\frac{\partial \boldsymbol{X}^{W}_{(n,i)}}{\partial \kappa} = \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial x}\cdot\frac{\partial x}{\partial \kappa} + \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial y}\cdot\frac{\partial y}{\partial \kappa} + \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial z}\cdot\frac{\partial z}{\partial \kappa} =$$

$$\frac{x - x_{\Gamma}}{d_v}\cdot\left(\frac{\partial \boldsymbol{R}_{11}}{\partial \kappa}\cdot x + \frac{\partial \boldsymbol{R}_{12}}{\partial \kappa}\cdot y + \frac{\partial \boldsymbol{R}_{13}}{\partial \kappa}\cdot z\right) + \frac{y - y_{\Gamma}}{d_v}\cdot\left(\frac{\partial \boldsymbol{R}_{21}}{\partial \kappa}\cdot x + \frac{\partial \boldsymbol{R}_{22}}{\partial \kappa}\cdot y + \frac{\partial \boldsymbol{R}_{23}}{\partial \kappa}\cdot z\right) +$$

$$\frac{z - z_{\Gamma}}{d_v}\cdot\left(\frac{\partial \boldsymbol{R}_{31}}{\partial \kappa}\cdot x + \frac{\partial \boldsymbol{R}_{32}}{\partial \kappa}\cdot y + \frac{\partial \boldsymbol{R}_{33}}{\partial \kappa}\cdot z\right)$$

$$\frac{\partial \mathbf{f}}{\partial t_x} = \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial \boldsymbol{X}^{W}_{(n,i)}}\cdot\frac{\partial \boldsymbol{X}^{W}_{(n,i)}}{\partial t_x} = \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial x}\cdot\frac{\partial x}{\partial t_x} + \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial y}\cdot\frac{\partial y}{\partial t_x} + \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial z}\cdot\frac{\partial z}{\partial t_x} = \frac{x - x_{\Gamma}}{d_v}$$

$$\frac{\partial \mathbf{f}}{\partial t_y} = \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial \boldsymbol{X}^{W}_{(n,i)}}\cdot\frac{\partial \boldsymbol{X}^{W}_{(n,i)}}{\partial t_y} = \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial x}\cdot\frac{\partial x}{\partial t_y} + \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial y}\cdot\frac{\partial y}{\partial t_y} + \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial z}\cdot\frac{\partial z}{\partial t_y} = \frac{y - y_{\Gamma}}{d_v}$$

$$\frac{\partial \mathbf{f}}{\partial t_z} = \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial \boldsymbol{X}^{W}_{(n,i)}}\cdot\frac{\partial \boldsymbol{X}^{W}_{(n,i)}}{\partial t_z} = \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial x}\cdot\frac{\partial x}{\partial t_z} + \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial y}\cdot\frac{\partial y}{\partial t_z} + \frac{\partial \mathbf{f}_{\mathscr{V}}}{\partial z}\cdot\frac{\partial z}{\partial t_z} = \frac{z - z_{\Gamma}}{d_v}$$

Appendix B

In Eq. (2), let a real feature point $\widetilde{X}^{W}_{(\mathscr{R}_n,i)}$ be set to (x, y, z), and let its corresponding plane be set to

$\{\widetilde{X}_{(\Gamma,a)}, \widetilde{X}_{(\Gamma,b)}, \widetilde{X}_{(\Gamma,c)}\} = \{(x_a, y_a, z_a), (x_b, y_b, z_b), (x_c, y_c, z_c)\}$

In addition, let $(p_a, p_b, p_c) = (\widetilde{X}_{(\Gamma,a)} - \widetilde{X}_{(\Gamma,b)}) \times (\widetilde{X}_{(\Gamma,a)} - \widetilde{X}_{(\Gamma,c)})$, and

$$p_a = (y_a - y_b)(z_a - z_c) - (z_a - z_b)(y_a - y_c)$$

$$p_b = (z_a - z_b)(x_a - x_c) - (x_a - x_b)(z_a - z_c)$$

$$p_c = (x_a - x_b)(y_a - y_c) - (y_a - y_b)(x_a - x_c)$$

Thus, Eq. (2) can be changed to

$$d_{(\mathscr{R}_n,i)} = \frac{|\overrightarrow{\boldsymbol{X_R X}_{(\Gamma,a)}}\cdot\vec{\boldsymbol{n}}|}{|\vec{\boldsymbol{n}}|} = \frac{|(x - x_a, y - y_a, z - z_a)\cdot(p_a, p_b, p_c)|}{|(p_a, p_b, p_c)|} = \frac{p_a(x - x_a) + p_b(y - y_a) + p_b(z - z_a)}{p_s} \tag{B.1}$$

where $p_s = \sqrt{{p_a}^2 + {p_b}^2 + {p_c}^2}$. Then, for the real feature, the elements of matrix $\boldsymbol{J}$ can be established by,

$$\frac{\partial \mathbf{f}}{\partial \omega} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial \mathbf{X}_{(n,i)}^{W}} \cdot \frac{\partial \mathbf{X}_{(n,i)}^{W}}{\partial \omega} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial x} \cdot \frac{\partial x}{\partial \omega} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial y} \cdot \frac{\partial y}{\partial \omega} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial z} \cdot \frac{\partial z}{\partial \omega} =$$

$$\frac{p_a}{p_s} \cdot \left(\frac{\partial \boldsymbol{R}_{11}}{\partial \omega} \cdot x + \frac{\partial \boldsymbol{R}_{12}}{\partial \omega} \cdot y + \frac{\partial \boldsymbol{R}_{13}}{\partial \omega} \cdot z \right) + \frac{p_b}{p_s} \cdot \left(\frac{\partial \boldsymbol{R}_{21}}{\partial \omega} \cdot x + \frac{\partial \boldsymbol{R}_{22}}{\partial \omega} \cdot y + \frac{\partial \boldsymbol{R}_{23}}{\partial \omega} \cdot z \right) +$$

$$\frac{p_c}{p_s} \cdot \left(\frac{\partial \boldsymbol{R}_{31}}{\partial \omega} \cdot x + \frac{\partial \boldsymbol{R}_{32}}{\partial \omega} \cdot y + \frac{\partial \boldsymbol{R}_{33}}{\partial \omega} \cdot z \right)$$

$$\frac{\partial \mathbf{f}}{\partial \varphi} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial \mathbf{X}_{(n,i)}^{W}} \cdot \frac{\partial \mathbf{X}_{(n,i)}^{W}}{\partial \varphi} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial x} \cdot \frac{\partial x}{\partial \varphi} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial y} \cdot \frac{\partial y}{\partial \varphi} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial z} \cdot \frac{\partial z}{\partial \varphi} =$$

$$\frac{p_a}{p_s} \cdot \left(\frac{\partial \boldsymbol{R}_{11}}{\partial \varphi} \cdot x + \frac{\partial \boldsymbol{R}_{12}}{\partial \varphi} \cdot y + \frac{\partial \boldsymbol{R}_{13}}{\partial \varphi} \cdot z \right) + \frac{p_b}{p_s} \cdot \left(\frac{\partial \boldsymbol{R}_{21}}{\partial \varphi} \cdot x + \frac{\partial \boldsymbol{R}_{22}}{\partial \varphi} \cdot y + \frac{\partial \boldsymbol{R}_{23}}{\partial \varphi} \cdot z \right) +$$

$$\frac{p_c}{p_s} \cdot \left(\frac{\partial \boldsymbol{R}_{31}}{\partial \varphi} \cdot x + \frac{\partial \boldsymbol{R}_{32}}{\partial \varphi} \cdot y + \frac{\partial \boldsymbol{R}_{33}}{\partial \varphi} \cdot z \right)$$

$$\frac{\partial \mathbf{f}}{\partial \kappa} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial \mathbf{X}_{(n,i)}^{W}} \cdot \frac{\partial \mathbf{X}_{(n,i)}^{W}}{\partial \kappa} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial x} \cdot \frac{\partial x}{\partial \kappa} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial y} \cdot \frac{\partial y}{\partial \kappa} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial z} \cdot \frac{\partial z}{\partial \kappa} =$$

$$\frac{p_a}{p_s} \cdot \left(\frac{\partial \boldsymbol{R}_{11}}{\partial \kappa} \cdot x + \frac{\partial \boldsymbol{R}_{12}}{\partial \kappa} \cdot y + \frac{\partial \boldsymbol{R}_{13}}{\partial \kappa} \cdot z \right) + \frac{p_b}{p_s} \cdot \left(\frac{\partial \boldsymbol{R}_{21}}{\partial \kappa} \cdot x + \frac{\partial \boldsymbol{R}_{22}}{\partial \kappa} \cdot y + \frac{\partial \boldsymbol{R}_{23}}{\partial \kappa} \cdot z \right) +$$

$$\frac{p_c}{p_s} \cdot \left(\frac{\partial \boldsymbol{R}_{31}}{\partial \kappa} \cdot x + \frac{\partial \boldsymbol{R}_{32}}{\partial \kappa} \cdot y + \frac{\partial \boldsymbol{R}_{33}}{\partial \kappa} \cdot z \right)$$

$$\frac{\partial \mathbf{f}}{\partial t_x} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial \mathbf{X}_{(n,i)}^{W}} \cdot \frac{\partial \mathbf{X}_{(n,i)}^{W}}{\partial t_x} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial x} \cdot \frac{\partial x}{\partial t_x} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial y} \cdot \frac{\partial y}{\partial t_x} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial z} \cdot \frac{\partial z}{\partial t_x} = \frac{p_a}{p_s}$$

$$\frac{\partial \mathbf{f}}{\partial t_y} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial \mathbf{X}_{(n,i)}^{W}} \cdot \frac{\partial \mathbf{X}_{(n,i)}^{W}}{\partial t_y} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial x} \cdot \frac{\partial x}{\partial t_y} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial y} \cdot \frac{\partial y}{\partial t_y} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial z} \cdot \frac{\partial z}{\partial t_y} = \frac{p_b}{p_s}$$

$$\frac{\partial \mathbf{f}}{\partial t_z} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial \mathbf{X}_{(n,i)}^{W}} \cdot \frac{\partial \mathbf{X}_{(n,i)}^{W}}{\partial t_z} = \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial x} \cdot \frac{\partial x}{\partial t_z} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial y} \cdot \frac{\partial y}{\partial t_z} + \frac{\partial \mathbf{f}_{\mathscr{R}}}{\partial z} \cdot \frac{\partial z}{\partial t_z} = \frac{p_c}{p_s}$$

References

Behzadian, B., Agarwal, P., Burgard, W., Tipaldi, G.D., 2015. Monte Carlo localization in hand-drawn maps. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Besl, P.J., Mckay, N.D., 1992. A method for registration of 3-D shapes. IEEE T. Pattern Anal. 14, 239–256.

Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., Csorba, M., 2001. A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans. Robot. Autom. 17 (3), 229–241.

Grisetti, G., Kummerle, R., Stachniss, C., Burgard, W., 2010. A tutorial on graph-based SLAM. IEEE Intell. Transp. Syst. Mag. 2, 31–43.

Grisetti, G., Stachniss, C., Burgard, W., 2007. Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Trans. Robot. 23, 34–46.

Hess, W., Kohler, D., Rapp, H., Andor, D., 2016. Real-time loop closure in 2D LiDAR SLAM. IEEE International Conference on Robotics and Automation (ICRA).

Henning, J.G., Radtke, P.J., 2006. Detailed stem measurements of standing trees from ground-based scanning lidar. Forest Sci. 52 (1), 67–80.

Hilker, T., Coops, N.C., Culvenor, D.D., Newnham, G., Wulder, M.A., Bater, C.W., Siggins, A., 2012. A simple technique for co-registration of terrestrial LiDAR observations for forestry applications. Remote Sens. Lett. 3, 239–247.

Javanmardi, M., Javanmardi, E., Gu, Y., Kamijo, S., 2017. Towards high-definition 3D urban mapping: road feature-based registration of mobile mapping systems and aerial imagery. Remote Sens. 9, 975.

Kelbe, D., Aardt, J.V., Romanczyk, P., Leeuwen, M.V., Cawse-Nicholson, K., 2016. Marker-free registration of forest terrestrial laser scanner data pairs with embedded confidence metrics. IEEE Trans. Geosci. Remote Sens. 54, 4314–4330.

Kohlbrecher, S., Von Stryk, O., Meyer, J., Klingauf, U., 2011. A flexible and scalable SLAM system with full 3D motion estimation. IEEE International Symposium on Safety, Security, and Rescue Robotics.

Konolige, K., Grisetti, G., Kümmerle, R., Limketkai, B., Vincent, R., 2010. Efficient sparse pose adjustment for 2D mapping. Proc. of Int. Conf on Intelligent Robots and Systems (IROS).

Kukko, A., Kaijaluoto, R., Kaartinen, H., Lehtola, V.V., Jaakkola, A., Hyyppä, J., 2017. Graph SLAM correction for single scanner MLS forest data under boreal forest canopy. ISPRS J. Photogramm. Remote Sens. 132, 199–209.

Kümmerle, R., Steder, B., Dornhege, C., Kleiner, A., Grisetti, G., Burgard, W., 2011. Large scale graph-based SLAM using aerial images as prior information. Autonomous Robots 30, 25–39.

Labbé, M., Michaud, F., 2014. Online global loop closure detection for large-scale multi-session graph-based SLAM. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems.

Latifi, H., Fassnacht, F.E., Müller, J., Tharani, A., Dech, S., Heurich, M., 2015. Forest inventories by LiDAR data: A comparison of single tree segmentation and metric-based methods for inventories of a heterogeneous temperate forest. Int. J. Appl. Earth Observ. Geoinform. 42, 162–174.

Liang, X., Hyyppä, J., 2013. Automatic stem mapping by merging several terrestrial laser scans at the feature and decision levels. Sensors 13, 1614–1634.

Liang, X., Hyyppä, J., Kaartinen, H., Lehtomäki, M., Pyörälä, J., Pfeifer, N., Holopainen, M., Brolly, G., Francesco, P., Hackenberg, J., Huang, H., Jo, H., Katoh, M., Liu, L., Mokroš, M., Morel, J., Olofsson, K., Poveda-Lopez, J., Trochta, J., Wang, D., 2018a. International benchmarking of terrestrial laser scanning approaches for forest inventories. ISPRS J. Photogramm. Remote Sens. 144, 137–179.

Liang, X., Hyyppä, J., Kukko, A., Kaartinen, H., Jaakkola, A., Yu, X., 2014. The use of a mobile laser scanning system for mapping large forest plots. IEEE Geosci. Remote Sens. Lett. 11, 1504–1508.

Liang, X., Kankare, V., Hyyppä, J., Wang, Y., Kukko, A., Haggrén, H., Yu, X., Kaartinen, H., Jaakkola, A., Guan, F., Holopainen, M., Vastaranta, M., 2016. Terrestrial laser scanning in forest inventories. ISPRS J. Photogramm. Remote Sens. 115, 63–77.

Liang, X., Kukko, A., Hyyppä, J., Lehtomäki, M., Pyörälä, J., Yu, X., Kaartinen, H., Jaakkola, A., Wang, Y., 2018b. In-suit measurements from mobile platforms: an emerging approach to address the old challenges associated with forest inventories. ISPRS J. Photogramm. Remote Sens. 143, 97–107.

Liang, X., Litkey, P., Hyyppä, J., Kaartinen, H., Vastaranta, M., Holopainen, M., 2012. Automatic stem mapping using single-scan terrestrial laser scanning. IEEE Trans. Geosci. Remote Sens. 50, 661–670.

Lovell, J.L., Jupp, D.L.B., Newnham, G.J., Culvenor, D.S., 2011. Measuring tree stem diameters using intensity profiles from ground-based scanning lidar from a fixed viewpoint. ISPRS J. Photogramm. Remote Sens. 66, 46–55.

Low, K., 2004. Linear least-squares optimization for point-to-plane ICP surface registration. Chapel Hill, University of North Carolina 4(10).

Luo, L., Wang, X., Guo, H., Lasaponara, R., Zong, X., Masini, N., Wang, G., Shi, P., Khatteli, H., Chen, F., Tariq, S., Shao, J., Bachagha, N., Yang, R., Yao, Y., 2019. Airborne and spaceborne remote sensing for archaeological and cultural heritage application: a review of the century (1907–2017). Remote Sens. Environ. 232, 111280.

Maas, H.G., Bienert, A., Scheller, S., Keane, E., 2008. Automatic forest inventory parameter determination from terrestrial laser scanner data. Int. J. Remote Sens. 29, 1579–1593.

Magnusson, M., Lilienthal, A., Duckett, T., 2007. Scan registration for autonomous mining vehicles using 3D-NDT. J. Field Robot. 24, 803–827.

Mellado, N., Aiger, D., Mitra, N.J., 2014. August. Super 4pcs fast global pointcloud registration via smart indexing. Comput. Graphics Forum 33 (5), 205–215.

Mengesha, T., Hawkins, M., Nieuwenhuis, M., 2015. Validation of terrestrial laser scanning data using conventional forest inventory methods. Eur. J. For. Res. 134, 211–222.

Mielle, M., Magnusson, M., Lilienthal, A.J., 2018. A method to segment maps from different modalities using free space layout MAORIS: MAp Of RIpples Segmentation. Proc. of the IEEE International Conference on Robotics and Automation (ICRA).

Mitra, N.J., Gelfand, N., Pottmann, H., Guibas, L., 2004. July. Registration of point cloud data from a geometric optimization perspective. In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, pp. 22–31.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P., 2009. Generic and real-time structure from motion using local bundle adjustment. Image Vis. Comput. 27 (8), 1178–1193.

Mur-Artal, R., Montiel, J.M.M., Tardós, J.D., 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE T. ROBOT. 31, 1147–1163.

Murphy, G.E., Acuna, M.A., Dumbrell, T., 2010. Tree value and log product yield determination in radiate pine (Pinus radiate) plantations in Australia: comparisons of terrestrial laser scanning with a forest inventory system and manual measurements. Can. J. For. Res. 40, 2223–2233.

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 J. Photogramm. Remote Sens. 129, 118–130.

Polewski, P., Yao, W., Cao, L., Gao, S., 2019. Marker-free coregistration of UAV and backpack LiDAR point clouds in forested areas. ISPRS J. Photogramm. Remote Sens. 147, 307–318.

Pomerleau, F., Colas, F., Siegwart, R., 2015. A review of point cloud registration algorithms for mobile robotics. Foundations and Trends®. Robotics 4 (1), 1–104.

Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M., 2008. September. Aligning point cloud views using persistent feature histograms. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3384–3391.

Shah, D.C., Campbell, M.E., 2013. A qualitative path planner for robot navigation using human-provided maps. Int. J. Robot. Res. 32, 1517–1535.

Shao, J., Zhang, W., Mellado, N., Grussenmeyer, P., Li, R., Chen, Y., Wan, P., Zhang, X., Cai, S., 2019. Automated markerless registration of point clouds from TLS and structured light scanner for heritage documentation. J. Cult. Herit. 35, 16–24.

Spies, T.A., 1998. Forest structure: a key to the ecosystem. Northwest Sci. 72, 34–39.

Stovall, A.E.L., Vorster, A.G., Anderson, R.S., Evangelista, P.H., Shugart, H.H., 2017. Non-destructive aboveground biomass estimation of coniferous trees using terrestrial LiDAR. Remote Sens. Environ. 200, 31–42.

Theiler, P.W., Wegner, J.D., Schindler, K., 2014a. Fast registration of laser scans with 4-point congruent sets-what works and what doesn't. ISPRS Ann. Photogram., Remote Sens. Spatial Inform. Sci. 2 (3), 149.

Theiler, P.W., Wegner, J.D., Schindler, K., 2014b. Keypoint-based 4-points congruent sets – automated marker-less registration of laser scans. ISPRS J. Photogramm. Remote Sens. 96, 149–163.

Trochta, J., Kral, K., Janik, D., Adam, D., 2013. Arrangement of terrestrial laser scanner positions for area-wide stem mapping of natural forests. Can. J. For. Res. 43, 355–363.

Wan, P., Wang, T., Zhang, W., Liang, X., Skidmore, A.K., Yan, G., 2019. Quantification of occlusions influencing the tree stem curve retrieving from single-scan terrestrial laser scanning data. Forest Ecosyst. 6 (1).

Wang, C., Hou, S., Wen, C., Gong, Z., Li, Q., Sun, X., Li, J., 2018. Semantic line framework-based indoor building modeling using backpacked laser scanning point cloud. ISPRS J. Photogramm. Remote Sens. 143, 150–166.

Wilkers, P., Lau, A., Disney, M., Calders, K., Burt, A., Gonzalez de Tanago, J., Bartholomeus, H., Brede, B., Herold, M., 2017. Data acquisition considerations for terrestrial laser scanning of forest plots. Remote Sens. Environ. 196, 140–153.

Zhang, J., Singh, S., 2014. LOAM: lidar odometry and mapping in real-time. Robotics: Science and Systems Conference.

Zhang, W., Chen, Y., Wang, H., Chen, M., Wang, X., Yan, G., 2016a. Efficient registration of terrestrial LiDAR scans using a coarse-to-fine strategy for forestry applications. Agric. For. Meteorol. 225, 8–23.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016b. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. Remote Sens. 8, 501.

Zhang, W., Wan, P., Wang, T., Cai, S., Chen, Y., Jin, X., Yan, G., 2019. A novel approach for the detection of standing tree stems from plot-level terrestrial laser scanning data. Remote Sens. 11 (2), 211.