TU Delft

Delft University of Technology

GlobalMatch

Registration of forest terrestrial point clouds by global matching of relative stem positions

Wang, Xufei; Yang, Zexin; Cheng, Xiaojun; Stoter, Jantien; Xu, Wenbing; Wu, Zhenlun; Nan, Liangliang

DOI
10.1016/j.isprsjprs.2023.01.013

Publication date
2023

Document Version
Final published version

Published in
ISPRS Journal of Photogrammetry and Remote Sensing

Citation (APA)
Wang, X., Yang, Z., Cheng, X., Stoter, J., Xu, W., Wu, Z., & Nan, L. (2023). GlobalMatch: Registration of forest terrestrial point clouds by global matching of relative stem positions. *ISPRS Journal of Photogrammetry and Remote Sensing*, *197*, 71-86. https://doi.org/10.1016/j.isprsjprs.2023.01.013

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

ISPRS Journal of Photogrammetry and Remote Sensing 197 (2023) 71–86

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

GlobalMatch: Registration of forest terrestrial point clouds by global matching of relative stem positions

Xufei Wang [a], Zexin Yang [a,b,*], Xiaojun Cheng [a,1], Jantien Stoter [b], Wenbing Xu [c], Zhenlun Wu [d], Liangliang Nan [b]

[a] College of Surveying and Geo-Informatics, Tongji University, Shanghai, 200092, China
[b] 3D Geoinformation Research Group, Delft University of Technology, 2628 BL, Delft, The Netherlands
[c] College of Environment and Resources, Zhejiang A&F University, Hangzhou, 311300, Zhejiang, China
[d] Big Data Development Administration of Yichun, Yichun, 336000, Jiangxi, China

ARTICLE INFO

Keywords:
Point cloud
Registration
Forest
Laser scanning
Dataset

ABSTRACT

Registering point clouds of forest environments is an essential prerequisite for LiDAR applications in precision forestry. State-of-the-art methods for forest point cloud registration require the extraction of individual tree attributes, and they have an efficiency bottleneck when dealing with point clouds of real-world forests with dense trees. We propose an automatic, robust, and efficient method for the registration of forest point clouds. Our approach first locates tree stems from raw point clouds and then matches the stems based on their relative spatial relationship to determine the registration transformation. The algorithm requires no extra individual tree attributes and has quadratic complexity to the number of trees in the environment, allowing it to align point clouds of large forest environments. Extensive experiments on forest terrestrial point clouds have revealed that our method inherits the effectiveness and robustness of the stem-based registration strategy while exceedingly increasing its efficiency. Besides, we introduce a new benchmark dataset that complements the very few existing open datasets for the development and evaluation of registration methods for forest point clouds. The source code of our method and the dataset are available at https://github.com/zexinyang/GlobalMatch.

1. Introduction

The light detection and ranging (LiDAR) technology, especially terrestrial laser scanning (TLS), has brought forest inventories to a brand new 3D era (Calders et al., 2020; Disney, 2019; Liang et al., 2016). The registration of TLS scans collected from different viewpoints is a fundamental and necessary precondition for subsequent forestry applications. However, the registration of forest point clouds remains a challenging task due to the mutual occlusion of forest structures. The most common and reliable practice is marker-based registration, which requires manually placing and transporting markers and is thus expensive, laborious, and time-consuming. Approaches dedicated to urban scenes rely on either local (Cai et al., 2019; Huang et al., 2022) or global (Chen et al., 2019) descriptors, which are effective for well-structured urban environments. However, due to the complexity, irregularity, and instability of forest environments, these methods become fragile and ineffective when applied to forest point clouds.

Given the fact that tree stems are the most stable structure in the forest environment, existing forest point cloud registration methods rely on the extracted tree stems and other attributes for the registration. These methods share a similar two-stage workflow: *stem mapping* followed by *stem matching* (Henning and Radtke, 2006, 2008; Liang and Hyyppä, 2013; Liu et al., 2017; Kelbe et al., 2016a,b; Tremblay and Béland, 2018; Dai et al., 2020; Ge et al., 2021; Polewski et al., 2019; Guan et al., 2019; Hauglin et al., 2014; Hyyppä et al., 2021). The stem mapping stage locates tree stem positions and extracts necessary individual tree attributes, e.g., diameter at breast height (DBH) and tree height. The second stage establishes correspondences of tree stems across a pair of point clouds by matching their positions and attributes, which is typically conducted in an iterative trial-and-error manner (see Fig. 1(a)). Although this stem-based strategy has been continuously improved in the past decades, significant limitations remain as follows.

(1) Inefficiency in stem mapping. Most methods (Liang and Hyyppä, 2013; Kelbe et al., 2016a,b; Liu et al., 2017; Tremblay and Béland,

* Corresponding author at: College of Surveying and Geo-Informatics, Tongji University, Shanghai, 200092, China.
E-mail addresses: tjwangxufei@tongji.edu.cn (X. Wang), zexinyang@tongji.edu.cn (Z. Yang), j.e.stoter@tudelft.nl (J. Stoter), liangliang.nan@tudelft.nl (L. Nan).

[1] He unfortunately passed away before the submission of this paper.

https://doi.org/10.1016/j.isprsjprs.2023.01.013
Received 28 May 2022; Received in revised form 7 December 2022; Accepted 20 January 2023
Available online 1 February 2023
0924-2716/© 2023 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.

2018; Polewski et al., 2019; Guan et al., 2019; Hyyppä et al., 2021) adopt an existing algorithm originally developed for tree detection or parameter estimation to identify tree stems. However, the existing stem detection algorithms (Othmani et al., 2011; Liang et al., 2011; Kelbe et al., 2015; Xia et al., 2015; Yang et al., 2016a; Polewski et al., 2017; Ye et al., 2020; Hyyppä et al., 2020) can be time-consuming since they introduce computationally intensive steps (e.g., individual tree segmentation, leaf and wood separation, and subsection stem modeling) to guarantee the completeness of detection and the high accuracy of extracted tree attributes.

(2) Ambiguities and inefficiency in stem matching. Most existing stem matching methods rely on both tree locations and other attributes (e.g., DBH, tree height, or tree crown) (Liu et al., 2017; Kelbe et al., 2016a,b; Tremblay and Béland, 2018; Dai et al., 2020; Ge et al., 2021; Hauglin et al., 2014; Dai et al., 2019), and thus they are sensitive to ambiguities in the extracted tree attributes. For example, it is common that multiple trees in a plantation forest have similar DBH and height values. Besides, the overwhelming computational burden is a fatal flaw of most stem-based matching algorithms. Specifically, their running time grows dramatically with the increasing number of trees due to the exhaustive nature of their verification based on the iterative trial-and-error alignment framework, which results in their inability to handle forest data containing a large number of trees, not to mention the demand for highly accurate tree attributes.

(3) Insufficient data for evaluation. Though improvements have been reported, the actual progress is vague to the research community because experiments are conducted on different closed datasets. To enable reliable evaluation and comparison of forest point cloud registration methods, open-access benchmark datasets with a large data volume and diverse forestry scenarios are urgently needed but unfortunately lacking.

In this work, we aim to resolve the above challenges to achieve efficient and robust registration of forest point clouds. Since exploiting individual tree attributes slows down stem mapping and makes stem matching sensitive to complex forest environments, we build an efficient attribute-free registration method named *GlobalMatch* for forest point clouds. The proposed approach relies on tree positions only, avoiding the computationally expensive and unstable tree attribute derivation process. It consists of a stem mapping algorithm that can efficiently locate tree stems in raw scans and a simple yet significantly efficient stem matching algorithm that needs no extra tree attributes but only stem positions. Moreover, most existing forest registration methods follow an iterative trial-and-error procedure (Henning and Radtke, 2006, 2008; Liang and Hyyppä, 2013; Kelbe et al., 2016a,b; Liu et al., 2017; Tremblay and Béland, 2018; Dai et al., 2020; Guan et al., 2020; Ge et al., 2021; Hauglin et al., 2014; Dai et al., 2019; Polewski et al., 2019; Hyyppä et al., 2021) and have a complexity between cubic and exponential with regard to the number of trees (Liang and Hyyppä, 2013; Kelbe et al., 2016a; Tremblay and Béland, 2018; Polewski et al., 2019) or keypoints (Guan et al., 2020; Dai et al., 2020). By contrast, our work falls in the inlier-grouping framework and has quadratic time complexity, so it can be applied to point clouds of large forest environments or those with dense tree instances. The comparison of these two underlying registration frameworks is illustrated in Fig. 1 and detailed in Section 2.2.

Extensive experiments show that our approach inherits effectiveness and robustness and meanwhile exceedingly increases the efficiency of the stem-based registration scheme. For comprehensive evaluation and valid comparison of registration algorithms, we also introduce a new benchmark dataset named *Tongji-Trees*, to complement the very rare publicly available data for marker-free registration of TLS scans of forest areas. In summary, our main contributions include:

(1) A fast stem mapping method that can robustly extract stem positions from raw forest scans.
(2) An efficient, robust, and deterministic stem matching algorithm that does not require iteratively extracting the correspondences and estimating the registration transformation. Our algorithm has quadratic time complexity, making it suitable for registering point clouds of large forestry areas.
(3) A benchmark dataset for evaluating forest point cloud registration methods. The new dataset consists of twenty point clouds collected from four forest plots with diversity concerning understory vegetation and the density, distributions, and species of trees.

2. Related work

In light of the extensive literature on point cloud registration (Maiseli et al., 2017; Dong et al., 2020), in this section, we mainly review the works that are most relevant to ours, namely, stem mapping, stem-based registration, and benchmark datasets.

2.1. Stem mapping

Existing stem mapping solutions mainly fall into two categories: range image-based methods (Haala et al., 2004; Forsman and Halme, 2005; Forsman et al., 2012) and unordered point-based methods (Othmani et al., 2011; Liang et al., 2011; Kelbe et al., 2015; Xia et al., 2015; Yang et al., 2016a; Polewski et al., 2017; Ye et al., 2020; Shao et al., 2020). The former category of methods, taking advantage of the ordered image structure, can efficiently extract tree stems by grouping the pixels of the range image based on local properties. However, range images are not always available, which hampers their use. The latter category of methods slices the point cloud vertically and identifies tree stems by fitting 2D circles or 3D cylinders. A series of contiguous circles or cylinders are then connected to form a complete tree stem. Typically, complex shape fitting (e.g., ellipse Ye et al., 2020 or tapered cylinder Kelbe et al., 2015), skeleton optimization (Othmani et al., 2011), individual tree segmentation (Yang et al., 2016a; Guan et al., 2019), supervised point classification (Polewski et al., 2017), and growth direction constraints (Xia et al., 2015; Ye et al., 2020) are used to improve detection integrity and mapping accuracy. However, these computationally intensive steps harm the efficiency of stem mapping.

We want to point out that existing stem mapping techniques pursue the completeness of detection results or the accuracy of calculated tree attributes using computationally expensive steps, which may not be necessary. In registration, not every single tree contributes to the registration. A good registration result can be guaranteed as long as parts of trees in the overlapping area have been identified. It is, therefore, valuable to design a fast stem mapping approach for forest point cloud registration. In this work, we introduce a fast stem mapping method to locate stems from raw scans. As our registration method does not require accurate tree attributes, our stem mapping does not have computationally expensive steps for complete stem structure recovery.

2.2. Stem-based registration

Marker-free forest registration techniques generally use tree stems as basic primitives. They broadly fall into two registration frameworks: iterative trial-and-error (Fig. 1(a)) and inlier-grouping (Fig. 1(b)).

2.2.1. Iterative trial-and-error based methods

In these methods (Henning and Radtke, 2006, 2008; Liang and Hyyppä, 2013; Kelbe et al., 2016a,b; Liu et al., 2017; Tremblay and Béland, 2018; Dai et al., 2020; Guan et al., 2020; Ge et al., 2021; Hauglin et al., 2014; Dai et al., 2019; Polewski et al., 2019; Hyyppä et al., 2021), finding the optimal transformation is achieved through an iterative process. First, a minimum sample set (MSS) is chosen to calculate a transformation. Using this transformation, source stems are then transformed to the coordinate system of target stems. Finally, correspondences are extracted by matching close trees with similar tree

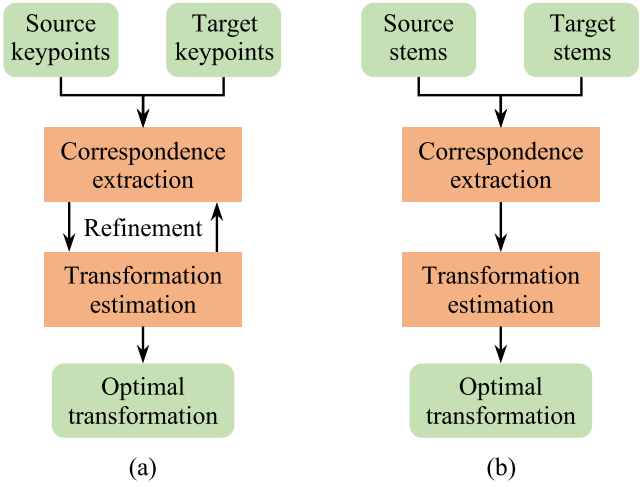

Fig. 1. Comparison between the existing forest point cloud registration framework and ours. (a) The iterative trial-and-error framework. Given two sets of keypoints, correspondences and transformations are estimated and refined iteratively until the optimal transformation is found. (b) Our inlier-grouping framework. With two sets of stem positions extracted from forest scans, our method first establishes correspondences by grouping all inliers, and then it computes the optimal transformation using the correspondences. Compared to the methods that follow the iterative trial-and-error framework, our method establishes correspondences and computes the transformation only once.

attributes and employed to re-estimate the transformation. The above three steps are iteratively conducted until the optimal transformation (i.e., the one with the largest number of corresponding trees) is found. One well-known paradigm of this type of method is the random sample consensus (RANSAC) algorithm (Fischler and Bolles, 1981; Kelbe et al., 2015; Tremblay and Béland, 2018).

These methods are robust but computationally expensive since, in each iteration, all the stem positions are transformed and used to determine the inliers. Attempting all possible MSSs is thus not computationally affordable. For example, restricted by the heavy computational burden of examining all possible stem matches, Liang and Hyyppä (2013) only manage to achieve 2D horizontal alignment of forest TLS scans and remain the vertical translation unsolved. On the other hand, sampling MSS randomly (i.e., RANSAC) does not perform well at a high outlier ratio (which is common in forest environments) since its running time increases exponentially with the outlier ratio (Cai et al., 2019; Kelbe et al., 2015). To enable the practical registration of point clouds of large forest environments, current studies have proposed three strategies to address the computational inefficiency.

The first strategy is to discard unlikely MSSs. Specifically, the similarities derived from relative stem positions (Hauglin et al., 2014; Kelbe et al., 2016a,b; Liu et al., 2017; Tremblay and Béland, 2018; Hyyppä et al., 2021) and additional tree attributes (e.g., DBH Hauglin et al., 2014; Kelbe et al., 2016a,b; Liu et al., 2017; Tremblay and Béland, 2018 and tree height Hauglin et al., 2014; Liu et al., 2017) are used to eliminate false MSSs and sort the remaining ones. These methods work well for forests of small or medium sizes. However, most of them remain inefficient when dealing with large-scale forests. For example, the memory usage and execution time of Kelbe et al. (2016a) can reach prohibitively large when the forest scene has more than 50 trees, while its efficient version (Tremblay and Béland, 2018) also takes more than an hour for registering scans containing more than 100 trees. Moreover, they can become impractical in plantation forests due to the ambiguities of tree attributes. Using only the locations of trees as input, the 2D coarse registration method presented by Hyyppä et al. (2021) can efficiently co-register multiplatform LiDAR data in large-scale forests. However, the algorithm is designed to solve a 2D registration problem and only works with well-leveled point clouds (i.e., the Z axis of the point cloud points upward).

The second strategy is to reduce the number of keypoints. Recent studies introduce novel keypoints, e.g., visual occlusion points identified from shaded areas (Guan et al., 2020) or mode points extracted from tree crowns (Dai et al., 2019, 2020). Compared to stems, these local keypoints typically have a smaller number. They are used to register forest scans (Dai et al., 2019; Guan et al., 2020) or to guide the stem matching (Dai et al., 2020) to bypass the computational burden of matching numerous stems. Notwithstanding the enhanced efficiency, these local keypoints require high-quality input point clouds that are still challenging to acquire for complex forest environments (Ge et al., 2021). Furthermore, similar to stem positions, the increase in the number of keypoints also results in a dramatic increase in computation time (Guan et al., 2020).

The third strategy is to use advanced optimization techniques. Ge et al. (2021) register multiview forest point clouds by adopting the 4-points congruent sets algorithm (Aiger et al., 2008; Ge, 2017) that reduces the number of trials required to establish a reliable registration. Polewski et al. (2019) apply simulated annealing metaheuristic (van Laarhoven and Aarts, 1987) to determine the optimal corresponding trees between multiplatform LiDAR point clouds. Dai et al. (2019) consider the alignment as a maximum likelihood estimation problem and employ the coherent point drift algorithm (Myronenko and Song, 2010) to fuse forest airborne and terrestrial point clouds.

2.2.2. *Inlier-grouping based methods*

This category of methods seeks the optimal transformation in a one-shot manner. Specifically, all corresponding trees (i.e., inliers) are first extracted and then employed to determine the optimal transformation. On the one hand, rather than verifying the transformation and refining correspondences in each iteration as in the iterative trial-and-error framework, inlier-grouping methods extract correspondences and estimate the transformation only once and are thus generally more efficient (Zhao et al., 2021). On the other hand, as the transformation is estimated only once, the resulting transformation may not be accurate if the correspondences are unreliable (Gressin et al., 2013; Dai et al., 2019). Therefore, the key to these methods is to extract true correspondences (i.e., inliers) and eliminate false ones (i.e., outliers) Cai et al., 2019.

Though the inlier-grouping framework has been widely adopted in registration techniques targeting man-made objects (Chen and Bhanu, 2007; Albarelli et al., 2010; Aldoma et al., 2012) or urban scenes (Yang et al., 2016b; Cai et al., 2019), it is rarely used in forest point cloud registration. To the best of our knowledge, the co-registration approach for multiplatform forest LiDAR data proposed by Guan et al. (2019) is the only one that falls into this framework. For each stem position extracted from raw scans via individual tree segmentation, the horizontal coordinates and neighboring stem positions are used to construct a 2D triangulated irregular network (TIN). Then, stem correspondences are extracted by matching 2D TINs (where the matching quality is measured by the number of similar triangles within the TINs) and employed to determine the registration transformation. Their method can reach a satisfactory accuracy (i.e., less than 20 cm) in co-registering multiplatform LiDAR scans, while their running time has not been reported.

In this work, we propose a stem matching method that follows the inlier-grouping framework. We generalize the urban matching approach of Yang et al. (2016b) to forest environments, and we tailor three main improvements for robust and efficient forest point cloud registration. First, we utilize only geometric coordinates because the semantic features defined in their method are not available in forest environments. Second, by constructing local triangles that encode relative stem positions, we reduce the algorithmic complexity concerning the number of trees from sextic to quadratic. Last, unlike their greedy grouping algorithm that directly adopts the line-to-line geometric consistency of Chen and Bhanu (2007), we propose a graph-to-graph consistency measure that guarantees the robustness in triangle-based correspondence grouping.

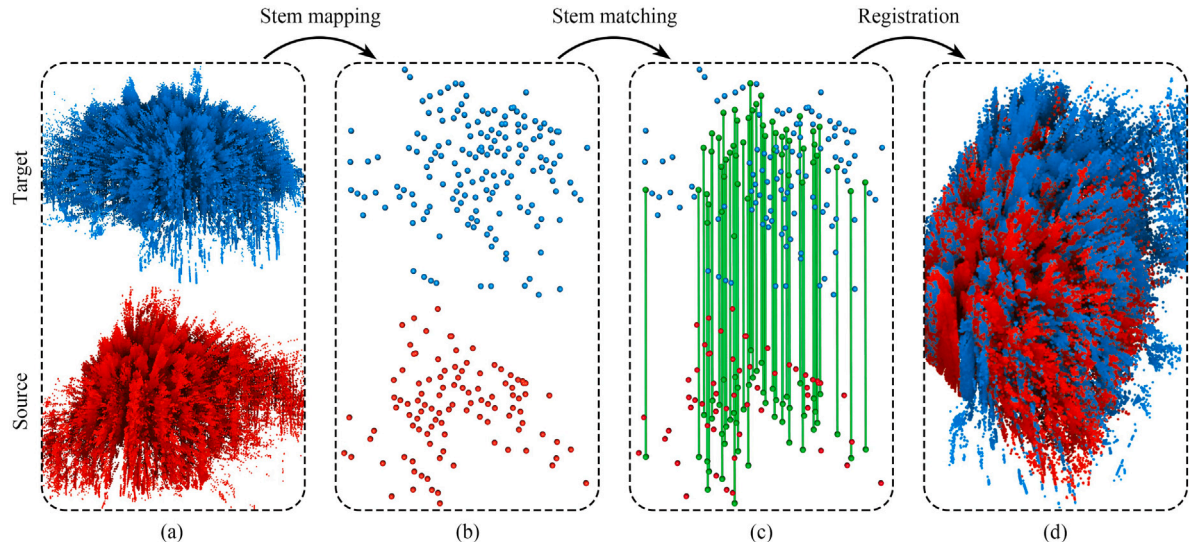

Fig. 2. The pipeline of the proposed method. (a) Input point clouds. (b) Stem positions. (c) Stem matches. (d) Registration result. Given a pair of forest TLS point clouds as input (a), the stem positions of trees are first extracted from each point cloud (b). Then, correspondences between stem positions are established by exploiting relative distances of the stem positions (c). Finally, registration is achieved by applying the transformation computed from the stem correspondences (d).

2.3. Benchmark datasets

Existing benchmark datasets for point cloud registration typically target man-made objects, indoor scenes, or urban environments (Mian et al., 2006; Theiler et al., 2015; Sanchez et al., 2017; Chen et al., 2019; Dong et al., 2020). Very few datasets are publicly available for the evaluation of registration methods for forest point clouds. To the best of our knowledge, WHU-FGI (Dong et al., 2020) and ETH-Trees (Theiler et al., 2015) are the only two datasets for this purpose, which contain only five and six scans of a single forest plot, respectively. The other FGI forest dataset (Liang et al., 2018) is intended for assessing tree attribute extraction algorithms, which cannot be used for evaluating registration methods due to the inaccessibility of the registration ground truth. In this work, we release Tongji-Trees, a new benchmark dataset consisting of twenty scans for the comprehensive evaluation of marker-free registration methods for TLS scans of forest areas.

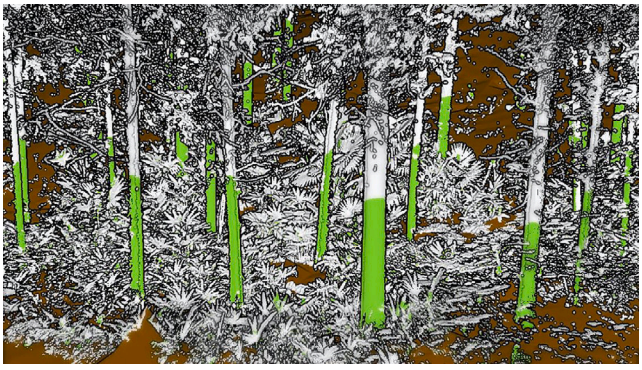

Fig. 3. An example of stem extraction results (a close-up view). The reconstructed DTM is visualized in brown and the extracted tree stems are in green. In this point cloud, the number of points is 30,574,461, while the number of extracted tree stems is only 116. Our method uses the extracted tree stems for registration.

3. Methodology

Our method takes as input a pair of forest TLS scans and outputs a rigid transformation to register the two scans. It consists of three main stages illustrated in Fig. 2: stem mapping, stem matching, and registration, which are detailed as follows.

3.1. Stem mapping

This stage aims to identify stem positions, i.e., the intersections of stem axes and the ground surface, from a raw TLS scan. This is achieved by first extracting stems, followed by intersecting the stem axes and the ground surface.

To avoid interference from unstable structures such as forest canopy and understory vegetation, we identify the stem points of trees before the extraction of individual stems. To filter out forest canopies, we create a digital terrain model (DTM) based on the input point cloud (Zhang et al., 2016) and mark the points within 0.2–3 m from the DTM as the understory layer. Before excluding understory shrubs, we downsample the data of the understory layer for efficiency reasons. We exploit a voxel grid structure with a resolution of 1 cm, and for each voxel, only the point that is closest to its centroid is kept. Considering that tree stems typically have a vertical orientation, we identify tree stems by looking into the pointwise verticality (Demantké et al., 2012; Weinmann et al., 2013) defined as

$$v = 1 - |n_z|, \tag{1}$$

where n_z represents the third component of the normal vector of a point. A value of 1 indicates that the local surface of a point is perfectly vertically oriented, while a zero value indicates a perfectly horizontal local geometry. In this work, we use a fixed radius of 10 cm for querying the neighbors of a point using the method described in Behley et al. (2015) to estimate its normal using principal component analysis (Sanchez et al., 2020). We mark the points whose verticality is higher than a threshold γ as stem points. Because most real-world trees are near vertical, we empirically set γ to 0.9 in all our experiments. With this threshold, stem points are identified as those whose normal deviates within 5.8 degrees from the horizontal direction. A smaller γ is recommended if the majority of trees in the data are inclined. By extracting the stem points, we also significantly reduce the amount of data to be further processed. Fig. 3 shows an example of the stem extraction results, in which the number of points in the extracted stems is only about 1/80 the size of the initial input.

With the stem points identified from the previous step, we extract individual tree stems by detecting cylinders using the RANSAC

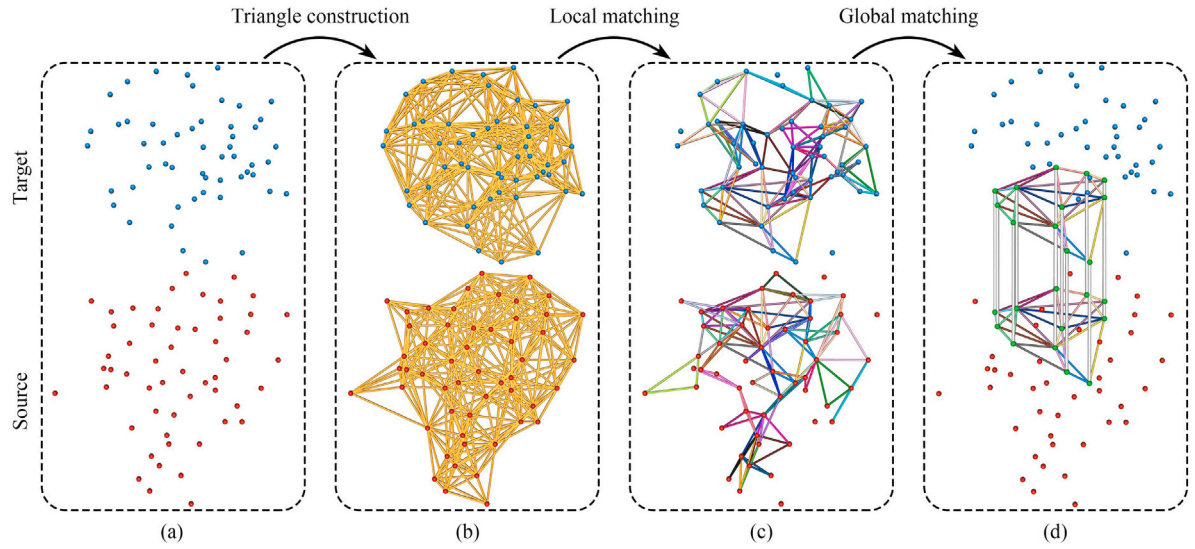

Fig. 4. Stem matching. (a) Stem positions. (b) Triangles encoding the local relative relationship of stem positions. (c) Locally matched triangle pairs. Each pair is assigned a random color for the visualization purpose. (d) Globally matched triangle pairs and stem matches. In this example, 12 stem matches (denoted by the white lines in (d)) are extracted from 53 target stem positions (denoted by the blue dots in (a)) and 53 source stem positions (denoted by the red dots in (a)). The numbers of initial triangles for the target and source scans are both 523 (b). The numbers of locally and globally matched triangle pairs are 43 (c) and 13 (d), respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

algorithm (Fischler and Bolles, 1981). Since detecting multiple cylinders by sequentially applying RANSAC is inefficient and sub-optimal (e.g., a previous wrong detection usually harms the subsequent detection) (Pham et al., 2016), we first employ the Euclidean clustering algorithm (Rusu et al., 2009b) to separate the stem points into small groups each containing a single stem (or in very rare cases few adjacent stems for trees with multi-branch structures near the ground surface). Then for the points of each stem, we apply RANSAC to detect a cylinder. It is worth noting that a few incorrectly detected stems will not affect the subsequent registration because they do not have correspondences in the other point cloud and are thus filtered out in the stem matching stage. Finally, we obtain a set of stem positions $\mathcal{L} = \{\mathbf{l}_i, 1 \le i \le N_l\}$ by intersecting the axes of the cylinders with the DTM model, where N_l denotes the number of stem positions.

3.2. Stem matching

The goal of this stage is to establish the correspondences between the previously extracted stem positions of the two input point clouds. Our stem matching method differs from most existing methods in the following two aspects: (1) it requires only the information of stem positions while existing methods (Liu et al., 2017; Kelbe et al., 2016a,b; Hauglin et al., 2014; Tremblay and Béland, 2018; Dai et al., 2020; Ge et al., 2021) also require additional tree attributes, such as DBH, tree height, or tree crown; (2) our stem matching is a deterministic non-iterative process while existing methods (Henning and Radtke, 2006, 2008; Liang and Hyyppä, 2013; Kelbe et al., 2016a,b; Liu et al., 2017; Tremblay and Béland, 2018; Dai et al., 2020; Guan et al., 2020; Ge et al., 2021; Hauglin et al., 2014; Dai et al., 2019; Polewski et al., 2019; Hyyppä et al., 2021) follow an iterative trial-and-error procedure that is less efficient.

Our stem matching consists of three steps as shown in Fig. 4. We first construct a set of local triangles for each scan from its stem positions to encode their relative spatial relationship. These triangles are then matched in the subsequent local matching and global matching steps to obtain the correspondences between the stem positions of the two input scans. In the following, we elaborate on each step of our stem matching algorithm and derive its overall time complexity.

3.2.1. Triangle construction

RANSAC-based methods (Kelbe et al., 2016a,b; Tremblay and Béland, 2018; Dai et al., 2020) commonly use an exhaustive approach to create triangles from the stem positions, i.e., any triplet of stem positions forms a triangle. Thus, the number of triangle pairs to be verified using this exhaustive approach is

$$\binom{N_l^\star}{3} \cdot \binom{N_l'}{3},$$

where $N_l^\star$ and N_l' denote the number of stems in the source and target scans, respectively. With this exhaustive approach, the number of triangles to be matched increases explosively with the number of stems extracted from the input point clouds. We observe that only certain triangles whose vertices have correspondences in the other point cloud contribute to stem matching and it is sufficient to build triangles by only connecting the adjacent stem positions. To this end, we construct only local triangles by connecting every stem position with its K-nearest stem positions. This simple strategy significantly reduces the number of triangle pairs to be verified from $(N_l^\star)^3 \cdot (N_l')^3$ to $N_l^\star \cdot N_l'$. Specifically, with an increasing number of detected stem positions in a scan, the number of triangles created by the exhaustive approach increases cubically, i.e.,

$$N_e = \frac{1}{6}N_l^3 - \frac{1}{2}N_l^2 + \frac{1}{3}N_l, \tag{2}$$

while that of our method increases linearly, i.e.,

$$N_K = \frac{K^2 - K}{2}N_l, \tag{3}$$

where N_l is the number of stem positions, and N_e and N_K denote the number of triangles constructed by the exhaustive approach and our method, respectively. In our implementation, we set K to 20, which ensures sufficient local triangles to be constructed to establish the correspondences between the two sets of stem positions. Fig. 5 depicts how the number of triangles increases for both methods. Our triangle construction step has quasilinear complexity of $\mathcal{O}\left(N_l^\star \log N_l^\star + N_l' \log N_l'\right)$.

With the set of triangles $\mathcal{T}^\star = \{t_i^\star, 1 \le i \le N_t^\star\}$ constructed from the source scan and the set of triangles $\mathcal{T}' = \{t_i', 1 \le i \le N_t'\}$ constructed from the target scan, where $N_t^\star$ and N_t' denote the number of triangles in the source and target scans respectively, we establish the correspondences between the source and target triangles in two steps: a local matching step and a global matching step.

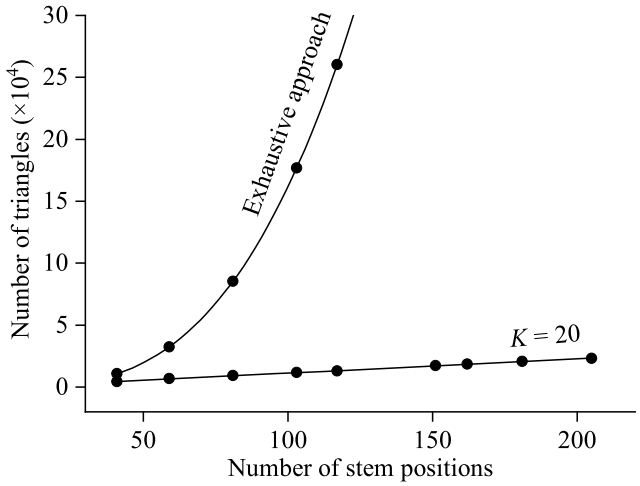

Fig. 5. The relationship between the number of constructed triangles and the number of stem positions. Curves for the exhaustive approach and our method (with $K = 20$) are plotted using stem positions extracted from the proposed Tongji-Trees dataset. For our method, the number of triangles increases linearly with respect to the number of stem positions, while the increase for the exhaustive approach is cubic.

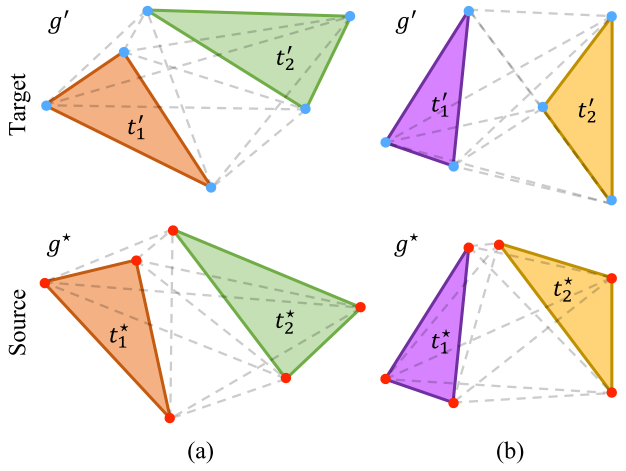

Fig. 6. Two examples illustrating the global dissimilarity between two locally matched triangle pairs $u_1 = \langle t_1^\star, t_1' \rangle$ and $u_2 = \langle t_2^\star, t_2' \rangle$. (a) u_1 and u_2 form a consensus set, i.e., $D_{global}(u_1, u_2) < \epsilon$. (b) u_2 does not agree with u_1 (i.e., $D_{global}(u_1, u_2) > \epsilon$) and thus they cannot form a consensus set.

3.2.2. Local matching

In this step, we establish triangle-wise correspondences between the two sets of triangles. This can be easily achieved by comparing the edge lengths of two triangles from the two sets. For efficiency reasons, we first sort the vertices of each triangle such that the one opposite to the longest edge comes first and the triangle is in counter-clockwise orientation. The following lemma lays the foundation for our triangle-wise local matching.

Lemma 1. *If three stem positions in the source scan $\mathbf{l}_i^\star, \mathbf{l}_j^\star, \mathbf{l}_k^\star \in \mathcal{L}^\star$ have correspondences in the target scan $\mathbf{l}_i', \mathbf{l}_j', \mathbf{l}_k' \in \mathcal{L}'$, the source triangle $\triangle\, l_i^\star l_j^\star l_k^\star \in \mathcal{T}^\star$ is congruent with the target triangle $\triangle\, l_i' l_j' l_k' \in \mathcal{T}'$.*

As the contrapositive of Lemma 1, the following lemma is equally true:

Lemma 2. *Given three pairs of stem positions $\langle \mathbf{l}_i^\star, \mathbf{l}_i' \rangle$, $\langle \mathbf{l}_j^\star, \mathbf{l}_j' \rangle$, $\langle \mathbf{l}_k^\star, \mathbf{l}_k' \rangle$, if the source triangle $\triangle\, l_i^\star l_j^\star l_k^\star \in \mathcal{T}^\star$ is not congruent with the target triangle $\triangle\, l_i' l_j' l_k' \in \mathcal{T}'$, then at least one pair of the stem positions does not match.*

By verifying any pair of source and target triangles, we accumulate potentially matched triangle pairs (Lemma 1) and filter out false matches (Lemma 2). In our work, we define the local dissimilarity between two triangles t_1 and t_2 as

$$D_{local}(t_1, t_2) = \sum_{i=1}^{3} \left| \varepsilon_i^\star - \varepsilon_i' \right|, \forall \left| \varepsilon_i^\star - \varepsilon_i' \right| < \epsilon, \tag{4}$$

where $\varepsilon_i^\star$ and ε_i' denote respectively the length of the corresponding edges in the two triangles. ϵ is the maximum allowed difference in edge length for two triangles to be matched, which is empirically set to 5 cm in all our experiments. Note that if the difference in edge length of any pair of edges is greater than ϵ, the two triangles are not considered matched. For each triangle t_j' in the target scan, we find its potential corresponding triangle $t_i^\star$ in the source scan as the one that has the minimum dissimilarity value, resulting in a locally matched triangle pair $u = \langle t_i^\star, t_j' \rangle$. The local matching step has quadratic complexity of $\mathcal{O}\left(N_t^\star \cdot N_t'\right)$, where $N_t^\star$ and N_t' are the number of triangles constructed in the source and target scans, respectively.

After local matching, a considerable portion of triangles for which local matches could not be found will be deleted. Since the locally matched triangle pairs may contain outliers, as seen in Fig. 4(c), a global matching step is necessary because two triangles being congruent is only a necessary (but not sufficient) condition for the stem positions to be correspondent.

3.2.3. Global matching

This step aims to establish the correspondences between the two sets of stem positions encoded in the source triangles and target triangles respectively. In theory, this can be achieved by matching two undirected fully connected graphs whose vertices are the stem positions of the source and target scans, respectively. We approach the graph matching by looking for the largest consensus set of the two graphs.

Unlike the local matching step that compares individual triangles, our global matching step uses triangle pairs as primitives. Before explaining our global matching, we first define a metric called global dissimilarity to measure the consistency between two pairs of locally matched triangles. Given two locally matched triangle pairs $u_1 = \langle t_1^\star, t_1' \rangle$ and $u_2 = \langle t_2^\star, t_2' \rangle$ (see Fig. 6), we build two undirected fully connected graphs, i.e., $g^\star$ from the vertices of $t_1^\star$ and $t_2^\star$, and g' from the vertices of t_1' and t_2'. Then the global dissimilarity between the two locally matched triangle pairs u_1 and u_2 is defined as

$$D_{global}(u_1, u_2) = \max \left| \mathcal{E}_i^\star - \mathcal{E}_i' \right|, \tag{5}$$

where $\mathcal{E}_i^\star$ and $\mathcal{E}_i'$ denote the lengths of the corresponding edges respectively in $g^\star$ and g'. Since $t_1^\star$ and t_1', $t_2^\star$ and t_2' are respective congruent triangle pairs, it is sufficient to test only the edges connecting vertices of different triangles (i.e., the edges denoted by the dashed lines in Fig. 6). D_{global} measures how much adding a new pair of locally matched triangles disagrees with an existing pair. If a locally matched triangle pair is consistent with any existing pair, we call it a globally matched pair, just to differentiate it from the locally matched pairs. Note that Yang et al. (2016b) compare only the relative positions of the centroids of two triangles, which has ambiguities. See Fig. 6(b) for an example, where the distance between the centroids of t_1' and t_2' is identical to that of $t_1^\star$ and $t_2^\star$ but the two pairs of triangles do not match.

We obtain the optimal correspondences between the two sets of stem positions by finding the largest consensus set. For each locally matched triangle pair u_i, our method grows a consensus set by accumulating globally matched triangle pairs u_j that agree with u_i (i.e., $D_{global}(u_i, u_j) < \epsilon$, where ϵ is the same threshold used in the local matching step). Fig. 7 illustrates the growth of a consensus set of globally matched triangle pairs. Each consensus set grows independently and the one that has accumulated the most globally matched triangle pairs gives us the optimal correspondences between the two sets of stem

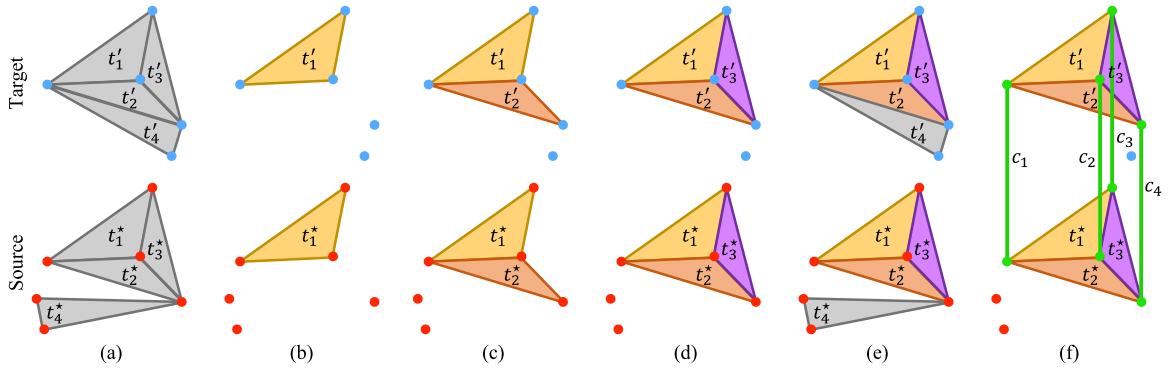

Fig. 7. An example illustrating the growth of a consensus set. Given an initial set of locally matched triangle pairs $\mathcal{U} = \{u_1, u_2, u_3, u_4\}$, $u_i = \langle t_i^\star, t_i' \rangle$ (a), our algorithm starts accumulating globally matched triangle pairs from an initial pair $\langle t_1^\star, t_1' \rangle$ (b). In each iteration, our method tries to find one more globally matched triangle pair and adds it to the growing set ((c) and (d)). The growing process stops when no more triangle pair that is consistent with the initial pair can be found. In this example, the triangle pair u_4 does not agree with the initial pair u_1 due to $D_{global}(u_1, u_4) > \epsilon$ (e). Finally, a set of globally matched triangle pairs $\mathcal{U}^\dagger = \{u_1, u_2, u_3\}$ is obtained, giving us the correspondences of the stem positions $C = \{c_1, c_2, c_3, c_4\}$ (f). The correspondences are denoted by the green lines connecting the stem positions of the two scans.

positions. The process for finding the largest consensus set is detailed in Algorithm 1.

The global matching step has quadratic complexity of $\mathcal{O}\left(N_u \cdot (N_u - 1)\right)$, where N_u represents the number of locally matched triangle pairs. It is worth noting that the growth of a consensus set can be immediately stopped when a sufficient number of globally matched triangle pairs have been accumulated. However, the growing process only involves comparing edge lengths and it is quite efficient to find the largest consensus set using an exhaustive search because of the small number of triangle pairs that remained after the local matching step. Such an exhaustive search guarantees the optimal consensus set is always used for the registration, ensuring robustness in registration. Note that our global matching step is carried out over the already matched local triangle pairs, and it has quadratic complexity (measured against the number of locally matched triangle pairs). Existing methods, such as Kelbe et al. (2016a,b), Tremblay and Béland (2018), Dai et al. (2020), perform an exhaustive search on the initial sets of stem positions, which has sextic time complexity (measured against the number of stem positions). Thus, the complexity of our global matching is several orders of magnitude smaller. For example, for the scan pairs shown in Fig. 4, our global matching does 2756 times triangle matching test, while the methods (Kelbe et al., 2016a,b; Tremblay and Béland, 2018) will have to evaluate 548 777 476 times.

3.2.4. Overall time complexity

Before deriving the overall time complexity of the proposed stem matching algorithm, notations for quantities are first recalled for readability. N_l, N_t, and N_u are the number of detected stem positions, constructed triangles, and locally matched triangle pairs, respectively. The superscript $\star$ represents the source scan, while the superscript $'$ denotes the target scan.

The overall time complexity of our method can be derived as follows. We can consider the majority of input stem positions are inliers, i.e., $N_l^\star \approx N_l' \approx N_l$. In this case, every triangle can find a local match in the other scan. Thus, the number of locally matched triangle pairs $N_u \approx N_t^\star \approx N_t' \approx N_t$, which leads to the worst-case time complexity. Because of the linear relationship between N_t and N_l (see Eq. (3)), the overall complexity can be given by $\mathcal{O}\left(2 \cdot N_l \log N_l + N_l \cdot N_l + N_l \cdot (N_l - 1)\right)$, which can be further simplified to $\mathcal{O}\left(N_l^2\right)$. Therefore, our method has quadratic time complexity.

3.3. Registration

From the correspondences of stem positions, a rigid transformation can be computed to register the two input scans. As there is no scale change among TLS scans, this transformation has 6 degrees of freedom

Algorithm 1 Global matching

Input: a set of locally matched triangle pairs $\mathcal{U}$
Output: correspondences of stem positions C

```
Initialization: C ← ∅; each consensus set U_i^† ← ∅
for each u_i ∈ U do
    insert u_i into U_i^†
    for each u_j ∈ U, i ≠ j do
        if D_global(u_i, u_j) < ε then
            insert u_j into U_i^†
        end if
    end for
end for
U_max^† ← max({U_i^†})
for each pair of corresponding vertices c_i ∈ U_max^† do
    insert c_i into C
end for
```

(DoFs) and can be written as a 4×4 matrix with the following structure

$$\mathbf{T} = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array}\right] = \left[\begin{array}{ccc|c} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \tag{6}$$

where $\mathbf{R}$ is the 3-DoF rotation and $\mathbf{t}$ is the 3-DoF translation. More information about computing a rigid transformation from a set of correspondences can be found in Sorkine-Hornung and Rabinovich (2017) and Besl and McKay (1992).

Using the above computed transformation, we can register the two input point clouds into the same coordinate system. Considering that scanners nowadays typically produce well-leveled point clouds, the complexity of the registration problem for such data can be reduced to 4-DoF, i.e., a translation (3 DoFs) and a rotation around the Z axis (1 DoF), and the final registration transformation has the following structure,

$$\dot{\mathbf{T}} = \left[\begin{array}{c|c} \mathbf{R}(\phi) & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array}\right] = \left[\begin{array}{ccc|c} \cos\phi & -\sin\phi & 0 & t_x \\ \sin\phi & \cos\phi & 0 & t_y \\ 0 & 0 & 1 & t_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \tag{7}$$

where $\mathbf{R}(\phi)$ defines the 1-DoF rotation about the azimuth ϕ. In this case, the 4-DoF registration can be achieved by the alignment of the two scans in the 2D horizontal direction (3 DoFs) and a 1D vertical translation (1 DoF). Specifically, with the x and y coordinates of the corresponding stem positions, we estimate the 2D horizontal

registration parameters ϕ, t_x, and t_y by solving a least-squares problem (Sorkine-Hornung and Rabinovich, 2017). Using the z coordinates of the corresponding stem positions $\langle z_i^\star, z_i' \rangle$, we calculate the vertical translation $t_z = \frac{1}{N_c}\sum_{i=1}^{N_c}\left(z_i' - z_i^\star\right)$, where N_c is the number of correspondences of stem positions.

Our method allows both 6-DoF registration and 4-DoF registration. When aligning forest TLS scans that are well leveled, the 4-DoF solution is preferred since it takes advantage of the scanners and may achieve better coarse registration accuracy. We provide an experimental comparison between our 6-DoF and 4-DoF registration strategies in Section 5.4.3. For the remaining experiments, we employed the 4-DoF solution.

After the coarse registration of the two scans, a fine registration algorithm can be employed to further improve the registration accuracy. Although there are fine registration algorithms (e.g., Shao et al., 2020) developed for forest scenarios, we use the iterative closest points (ICP) algorithm (Besl and McKay, 1992) because of its simplicity and effectiveness.

3.4. Implementation details

For efficiency, we parallelize computationally intensive parts of the proposed algorithm. Specifically, the following parts of our algorithm are parallelized:

- neighborhood query, verticality calculation, cylinder detection, and stem position computation in the stem mapping stage;
- local matching and global matching in the stem matching stage.

4. Benchmark dataset

4.1. Study areas

The Tongji-Trees dataset contains TLS scans of four plantation forest plots, which were collected in Shanghai, China (31.15° N, 121.12° E). The plot size varies from Plots #1 to #4, with a cover area of 65 × 50, 90 × 50, 80 × 40, and 65 × 55 m^2, respectively. Metasequoia trees are irregularly distributed in Plots #1 and #2, with an average height of 19 m and 23 m and a density of 1036 trees/ha and 1210 trees/ha, respectively. As for Plots #3 and #4, Liriodendron chinense trees with an average height of 7 m and Sapindus mukorossi trees with an average height of 10 m were planted in rows with spacing of 5 m and 3 m, respectively. Compared to Plots #1 and #2, Plots #3 and #4 are sparser, with a density of 433 trees/ha and 982 trees/ha, respectively. As shown in the top row of Fig. 8, the surface of Plot #1 is covered with wild grass. Plot #2 is partially overgrown with breast-high shrubs. Both Plots #3 and #4 have bare but humped surfaces. The mutual occlusion of complex shrubs in Plot #2 makes it more challenging for registration than in the other plots. Table 1 summarizes the information of all the plots in the Tongji-Trees dataset.

4.2. Data acquisition

By exploiting the "multi-scan" data acquisition approach (Liang et al., 2016), each of the four plots was scanned at five positions, i.e., one at the center and the other four away around the center, forming five scans for each plot and twenty scans for the whole dataset. The bottom row of Fig. 8 illustrates the scanner positions for each plot in the Tongji-Trees dataset. During field surveys, we used a Z+F 5010C terrestrial laser scanner to capture the point cloud data. The scanner has a measurement accuracy of ±3 mm at 50 m, a measurement range of 0.3–180 m, and a view angle of 360° × 320°.

4.3. Ground truth

Before scanning, we strictly leveled the scanner and set up spherical reference markers that could help us to produce correspondences for deriving the ground-truth registration transformation. Each marker was mounted on a tripod or placed stably on the ground (rather than fixed on tree stems, which usually causes occlusions). Each raw scan contains about 30 million points (see Table 1), with each point represented as a 7D vector (xyz coordinates, rgb colors, and an intensity value). Although our registration method relies on 3D coordinates only, the provided attributes may benefit other methods or studies. We identified the markers and served them as correspondences to compute the ground-truth transformation matrix for every possible pair of scans. The ground-truth transformation matrices have an average registration accuracy of 2.8 mm and thus can be used to evaluate the performance of registration methods.

5. Results and discussion

5.1. Experimental setup

Our algorithm is implemented in C++ using Point Cloud Library (Rusu and Cousins, 2011) and OpenMP (Dagum and Menon, 1998). All experiments were carried out on a laptop with an Intel Core 2.60 GHz i7 CPU (8 threads) and 32 GB RAM.

5.1.1. Test datasets

We have tested the proposed method on our Tongji-Trees dataset, the WHU-FGI dataset (Dong et al., 2020), and the ETH-Trees dataset (Theiler et al., 2015). The WHU-FGI dataset contains five scans captured from a 32 × 32 m^2 plot of a natural forest dominated by Scots pines, located at Evo, Finland (61.19° N, 25.11° E). The ETH-Trees dataset consists of six scans acquired in a forest with a large amount of underwood.

5.1.2. Evaluation metrics

To provide a comprehensive evaluation, we evaluated separately the results of coarse registration and fine registration (i.e., using the ICP algorithm Besl and McKay, 1992 as a post-process). We have recorded both matrix-based errors and pointwise errors. Furthermore, we have calculated the success rate of coarse registration methods.

- Matrix-based errors, i.e., the rotation error e_R and the translation error e_t between the estimated transformation parameters $\mathbf{R}, \mathbf{t}$ and the corresponding ground-truth values $\tilde{\mathbf{R}}, \tilde{\mathbf{t}}$:

$$e_R = \arccos\left(\frac{\mathrm{tr}(\tilde{\mathbf{R}}\mathbf{R}^\intercal) - 1}{2}\right), \tag{8}$$

$$e_t = \left\|\mathbf{t} - \tilde{\mathbf{t}}\right\|, \tag{9}$$

where e_R measures the difference between the two rotation matrices $\mathbf{R}$ and $\tilde{\mathbf{R}}$, and e_t quantifies the magnitude of the difference

Table 1
Statistics on the Tongji-Trees benchmark dataset.

Plot ID	Main tree species	Plot size (m^2)	Tree height (m)	Tree density (trees/ha)	Number of points of each scan				
					S1	S2	S3	S4	S5
#1	Metasequoia	65 × 50	19	1036	33,540,068	33,469,682	13,553,220	34,488,581	33,652,969
#2	Metasequoia	90 × 50	23	1210	33,661,875	33,891,812	30,574,461	34,193,448	33,197,039
#3	Liriodendron chinense	80 × 40	7	433	35,510,140	35,169,411	33,680,749	33,445,603	37,076,208
#4	Sapindus mukorossi	65 × 55	10	982	37,189,070	35,274,954	37,113,902	35,598,180	35,783,768

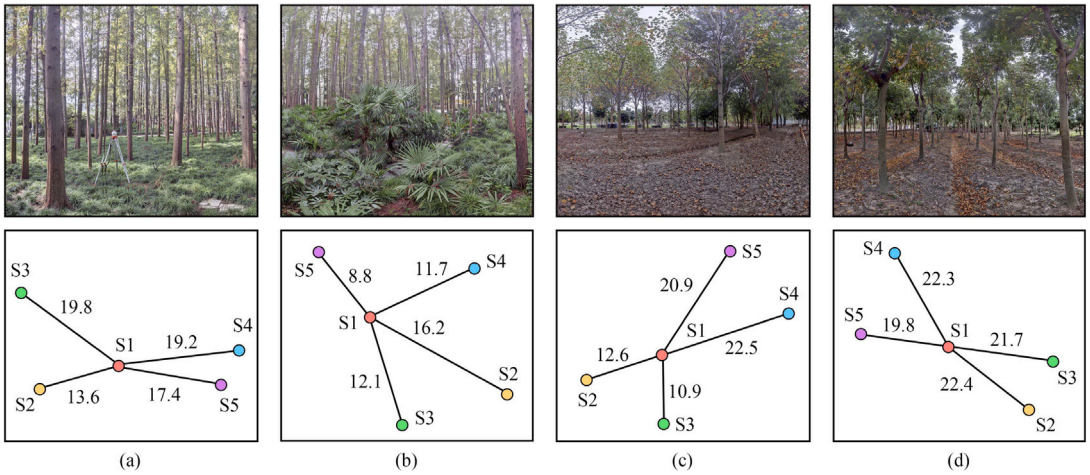

Fig. 8. The four plots of the Tongji-Trees dataset. (a) Plot #1. (b) Plot #2. (c) Plot #3. (d) Plot #4. The top row shows a representative view and the bottom row shows the five scanning positions (S1–S5) for each plot. The numbers indicate the distances (in meters) between the related scanning positions.

Table 2
Quantitative evaluation of stem mapping results on the three test datasets, i.e., Tongji-Trees (Plots #1–#4), WHU-FGI (Dong et al., 2020), ETH-Trees (Theiler et al., 2015).

Metric	Results of stem mapping (%)						
	Plot #1	Plot #2	Plot #3	Plot #4	WHU-FGI	ETH-Trees	Average
Precision	96.0	97.6	96.1	92.9	97.1	90.8	95.1
Recall	96.8	96.2	94.6	95.5	95.8	93.4	95.4
F1-score	96.4	96.9	95.4	94.2	96.4	92.1	95.2

of the two translation vectors $\mathbf{t}$ and $\tilde{\mathbf{t}}$. It is worth noting that the matrix-based metric is not straightforward for comparing two registration methods because a method may perform better in one measure but worse in the other (see Plot #1 in Table 4). In this case, it is not obvious to determine which method is more effective.

- Pointwise error, proposed by Chen et al. (2019). Given a source scan $\mathcal{P}^\star = \{\mathbf{p}_i^\star, 1 \le i \le N_p^\star\}$ containing $N_p^\star$ points, the pointwise error is defined as

$$e_p = \frac{1}{N_p^\star} \sum_{i=1}^{N_p^\star} \left\| \mathbf{R}\mathbf{p}_i^\star + \mathbf{t} - \left(\tilde{\mathbf{R}}\mathbf{p}_i^\star + \tilde{\mathbf{t}}\right) \right\|. \tag{10}$$

  Compared to the matrix-based errors, the pointwise error is more intuitive in comparing the performance of different registration methods.
- Success rate. A coarse registration is considered successful if the registered point cloud is sufficiently close to the ground truth, i.e., with a pointwise error less than a given threshold. This threshold was set to 50 cm for our test datasets following the evaluation method described in Chen et al. (2019). The success rate of coarse registration is measured by the number of successful registrations divided by the total number of scan pairs.

Matrix-based and pointwise errors are determined primarily by the quality of point clouds and the consistency of keypoints extracted from them. These metrics are only meaningful if the algorithm finds the correct correspondences. A failed registration will result in random and extremely large error values (> meters). Furthermore, successful coarse results can be optimized via the ICP algorithm to achieve similarly satisfactory accuracy (see Table 4). However, when a coarse registration fails, the fine-registration step becomes infeasible, resulting in complete failure in pairwise registration. From this point of view, we consider the success rate as the main metric for evaluating coarse registration.

5.2. Results

5.2.1. Stem mapping results

We quantitatively evaluated the effectiveness of our stem mapping method. Using the manually identified stem positions as ground truth, we calculated the precision, recall, and F1-score of our automatic detection results. As can be seen from Table 2, our stem mapping approach achieved 95.1%, 95.4%, and 95.2% in terms of average precision, recall, and F1-score, respectively on all six test plots.

5.2.2. Registration results

Following the previous studies (Liang and Hyyppä, 2013; Liu et al., 2017; Tremblay and Béland, 2018; Guan et al., 2020; Dai et al., 2020), we aligned scans at the corners of each plot to the one at the center. This is a common practice for registering point clouds of LiDAR-based forest inventory.

Fig. 9 demonstrates the final registration results of our proposed method on the three test datasets. As can be seen from the close-up view of the registration result of each plot, our method properly registered all TLS scans from diverse forest environments. The test datasets contain both sparse forests (with a tree density below 500 trees/ha, as shown in Fig. 9(c) and (e)) and dense forests (with a tree density around 1000 trees/ha, as shown in Fig. 9(a), (b), and (d)). Since our method has quadratic complexity to the number of trees, it can register the scans of both sparse and dense forests. Compared to plots covered in wild grass (see Fig. 9(a)) and plots having bare ground (see Fig. 9(c) and (d)), Plot #2 is partially overgrown with breast-high shrubs (see Fig. 9(b)), whereas the WHU-FGI and ETH-Trees datasets contain a large amount of underwood. Despite the complex understory structures, our method still extracted valid stem positions and managed to register these point clouds. Our method is also robust to the distribution of trees. It correctly aligned scans containing scattered trees (Fig. 9(a), (b), (e), and (f)) and scans consisting of trees planted in rows (Fig. 9(c) and (d)). Besides, our registration method succeeded on different tree species, i.e., Metasequoia (Fig. 9(a) and (b)), Liriodendron chinense (Fig. 9(c)), Sapindus mukorossi (Fig. 9(d)), and Scots pines (Fig. 9(e)).

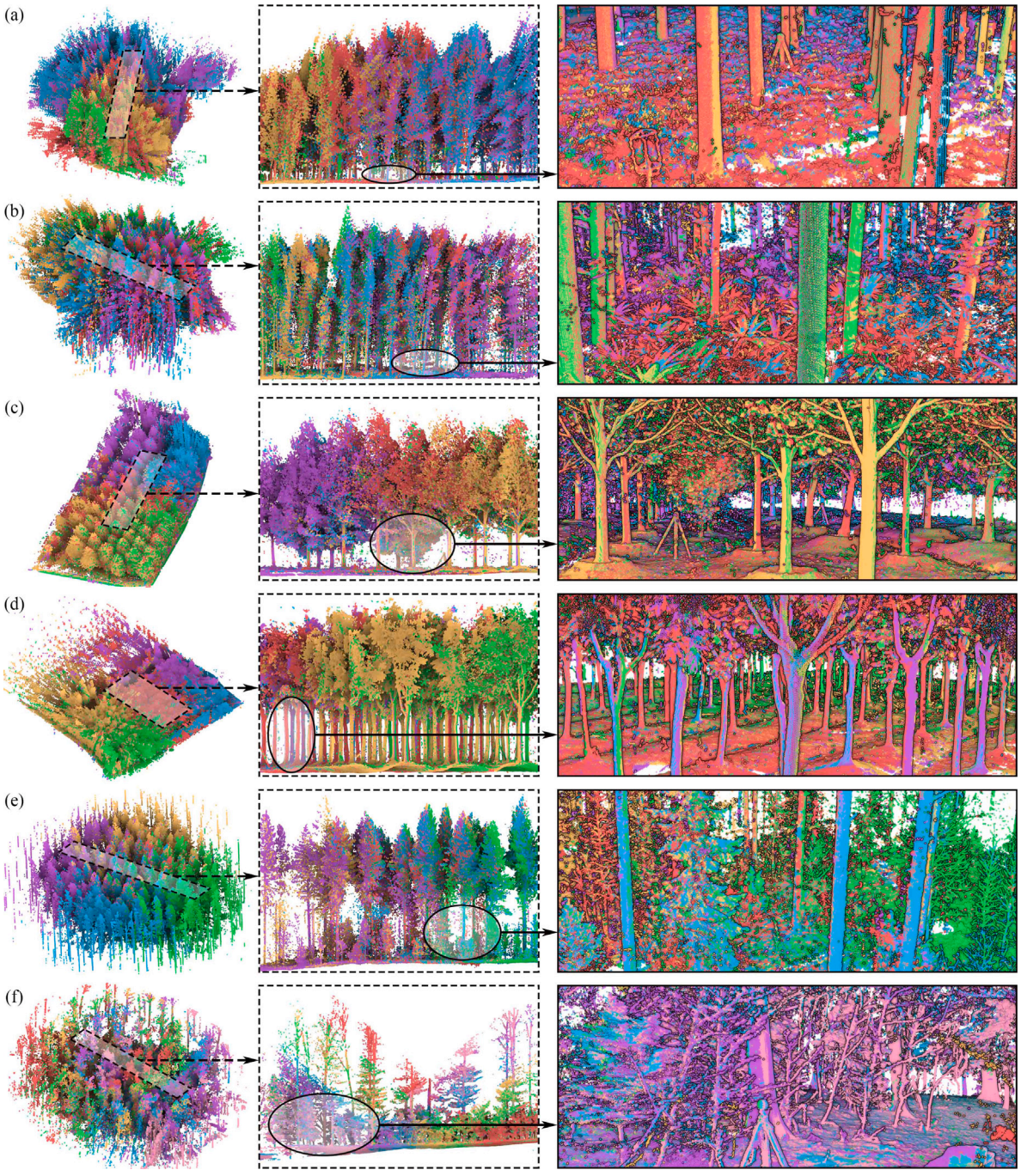

Fig. 9. Registration results on the three test datasets. (a) to (d) Plots #1–#4 of the Tongji-Trees dataset, respectively. (e) WHU-FGI (Dong et al., 2020). (f) ETH-Trees (Theiler et al., 2015). From left to right: an overview, a cross-section, and a close-up view of the registered point clouds of each plot. Scans S1–S5 in each plot are visualized in red, yellow, green, blue, and purple colors, respectively, and the S6 of ETH-Trees is in pink. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3 reports a quantitative evaluation of our final registration results. The average pointwise error of all test plots is 1 cm, showing the great potential of the proposed approach to substitute the costly marker-based methods.

5.3. Comparison

We have compared our method with three existing coarse registration algorithms: TOA (Kelbe et al., 2016a), Fast-TOA (Tremblay and Béland, 2018), and FMP+BnB (Cai et al., 2019).

- TOA and Fast-TOA are both tree-oriented approaches and are often used as the baseline methods for evaluating forest point cloud registration methods. Fast-TOA is an efficient variant of TOA. Using tree positions with DBH values as keypoints, they extract correspondences using the RANSAC strategy coupled with geometric and attribute constraints derived from the keypoints. The correspondences are used to determine a least-squares fit for the rigid transformation.
- FMP+BnB is one of the state-of-the-art methods for pairwise registration of general TLS scans (Ge and Hu, 2020; Li et al., 2021). Intrinsic-shape-signatures keypoints (Zhong, 2009) are extracted from the two point clouds, described using fast point feature

Table 3
Accuracy of the proposed method on the three test datasets, i.e., Tongji-Trees (Plots #1–#4), WHU-FGI (Dong et al., 2020), ETH-Trees (Theiler et al., 2015). e_p denotes the pointwise error, while e_R and e_t represent the two components (i.e., rotation and translation errors) of matrix-based errors, respectively.

Plot ID	Forest environment		Fine registration errors		
	Tree locations	Understory	e_p (cm)	e_R (mrad)	e_t (cm)
Plot #1	Dense, scattered	Wild grass	0.7	0.5	0.7
Plot #2	Dense, scattered	High shrub	1.3	1.2	1.0
Plot #3	Sparse, regular	Bare ground	0.9	0.3	0.8
Plot #4	Dense, regular	Bare ground	1.0	0.3	1.0
WHU-FGI	Sparse, scattered	Underwood	1.6	0.9	1.4
ETH-Trees	Sparse, scattered	Underwood	0.8	0.9	0.6
Average			1.0	0.7	0.9

histograms (Rusu et al., 2009a), and then matched to generate initial correspondences. Fast match pruning (FMP) is used to prune the correspondence set, followed by fast branch-and-bound (BnB) on the remaining correspondences to determine the optimal registration parameters.

Note that TOA and Fast-TOA are not end-to-end solutions: they cannot directly consume raw point clouds but require stem maps (i.e., stem positions and DBHs) as input. For a fair comparison, we prepared stem maps for every scan as has been done in the Fast-TOA paper (Tremblay and Béland, 2018). Moreover, TOA was not able to produce any results for scans containing a large number (e.g., greater than 70) of trees. Fast-TOA also becomes quite inefficient when handling point clouds consisting of more than 100 trees. For example, it took over an hour to register two scans containing only 117 and 109 trees, respectively. Hence, we had to reduce the number of trees in every stem map to approximately 50 (similar to the number of trees in the experimental data used in the original paper Tremblay and Béland, 2018) to be able to conduct the comparison. To guarantee sufficient overlap between each pair of simplified stem maps, we manually selected these 50 stems to include all corresponding ones.

5.3.1. Effectiveness

Following the common practice for plot-scale forest measurement, we used all coarse registration methods to register side scans of each plot to the central ones. Since all scan pairs overlap sufficiently, our approach and the three competing methods were able to register them all. All coarse registration results were further refined using the ICP algorithm (Besl and McKay, 1992). Both the coarse and fine registration errors are reported in Table 4.

Our method outperforms the others in terms of coarse registration accuracy on three of the six test plots and achieves the best overall performance with a minimum average pointwise error of 5.9 cm. We can also see that the ICP step significantly improved the coarse registration results, and it eliminated the accuracy gap between coarse registration methods. This indicates that, when evaluating a coarse registration method, we should prioritize its efficiency and success rate over accuracy.

5.3.2. Efficiency

The running times of our method are reported in Fig. 10. The average running times of the stem mapping and stem matching stages are 14.1 s and 0.6 s, respectively. The stem mapping stage takes most of the time, while the stem matching stage is relatively fast, taking less than three seconds even when handling scans containing a large number of trees (more than 200 trees/scan).

Fig. 11 reports the running times of our proposed method and the competing methods. We can see that our approach is much faster than the two stem-based methods, i.e., TOA and Fast-TOA, even though their preparation times of stem maps were not counted and their input was already simplified. Notably, TOA (relying heavily on DBH values)

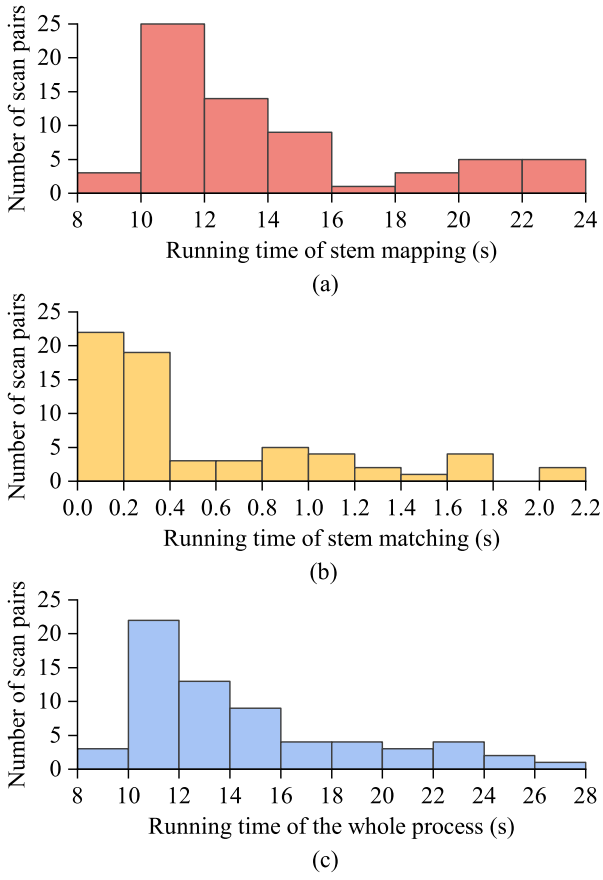

Fig. 10. Histograms of running times of the proposed method. (a) The stem mapping stage. (b) The stem matching stage. (c) The whole coarse registration process.

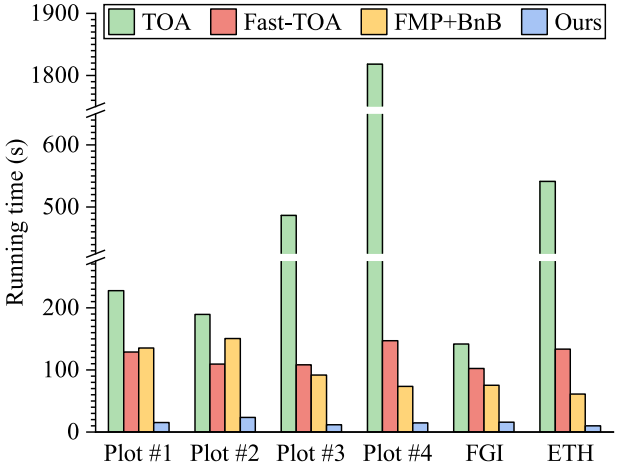

Fig. 11. Comparison of running times. Note that the running times of TOA (Kelbe et al., 2016a) and Fast-TOA (Tremblay and Béland, 2018) contain only their stem matching part that performs on simplified stem maps (~50 trees/scan), and the times for the creation of their input (stem maps) were not included. For FMP+BnB (Cai et al., 2019), the running time includes both keypoint detection and matching, and similarly for our method, the running time includes both stem mapping and stem matching.

took unacceptably long times on Plot #3, Plot #4, and ETH-Trees because trees in these plots have similar attributes. Our method tackles the inefficiency problem of the stem-based strategy by significantly reducing the complexity of stem matching, which allows it to handle scans containing a large number of trees and leads to consistently high efficiency. Compared to the local-descriptor-based approach FMP+BnB

Table 4
Accuracy comparison with three competing coarse registration algorithms, i.e., TOA (Kelbe et al., 2016a), Fast-TOA (Tremblay and Béland, 2018), and FMP+BnB (Cai et al., 2019). Every method successfully registered all scan pairs from the side scans of each plot to the central ones. All coarse registration results were further optimized by using the ICP algorithm (Besl and McKay, 1992) as a fine registration step. e_p denotes the pointwise error, while e_R and e_t represent the two components (i.e., rotation and translation errors) of matrix-based errors, respectively.

Plot ID	Method	Coarse registration errors			Fine registration errors		
		e_p (cm)	e_R (mrad)	e_t (cm)	e_p (cm)	e_R (mrad)	e_t (cm)
Plot #1	TOA	11.0	10.9	8.7	0.7	0.5	0.7
	Fast-TOA	9.4	9.2	6.7	0.7	0.5	0.7
	FMP+BnB	9.1	3.2	8.8	0.7	0.6	0.7
	Ours	**6.2**	0.7	6.3	0.7	0.5	0.7
Plot #2	TOA	27.8	22.4	21.4	1.2	1.2	0.9
	Fast-TOA	27.3	24.7	19.7	1.5	1.2	1.2
	FMP+BnB	6.6	2.5	6.3	1.1	1.2	0.9
	Ours	**6.2**	1.2	6.0	1.3	1.2	1.0
Plot #3	TOA	1.4	1.5	1.3	0.7	0.3	0.8
	Fast-TOA	**1.3**	1.3	1.2	0.7	0.3	0.7
	FMP+BnB	3.5	1.1	3.5	1.1	0.4	1.0
	Ours	3.6	0.5	3.6	0.9	0.3	0.8
Plot #4	TOA	3.9	3.6	3.8	0.9	0.3	0.9
	Fast-TOA	3.7	3.6	3.5	0.9	0.3	0.9
	FMP+BnB	6.3	1.8	6.3	1.1	0.4	1.1
	Ours	**3.1**	0.3	3.1	1.0	0.3	1.0
WHU-FGI	TOA	7.6	7.6	5.5	1.7	1.0	1.5
	Fast-TOA	**6.1**	7.5	3.7	1.7	1.0	1.5
	FMP+BnB	7.3	3.6	6.9	1.4	0.8	1.2
	Ours	7.6	0.6	7.6	1.6	0.9	1.4
ETH-Trees	TOA	20.7	22.8	18.2	0.8	0.9	0.6
	Fast-TOA	22.2	20.0	20.0	0.8	0.9	0.7
	FMP+BnB	**5.0**	3.6	4.5	0.8	0.9	0.6
	Ours	8.7	3.4	8.6	0.8	0.9	0.6
Average	TOA	12.1	11.4	9.8	1.0	0.7	0.9
	Fast-TOA	11.7	11.0	9.1	1.0	0.7	1.0
	FMP+BnB	6.3	2.6	6.1	1.0	0.7	0.9
	Ours	**5.9**	1.1	5.8	1.0	0.7	0.9

Table 5
Success rates of all coarse registration methods.

Method	Success rate of coarse registration (%)						
	Plot #1	Plot #2	Plot #3	Plot #4	FGI	ETH	Average
TOA	100	90	100	100	100	67	91
Fast-TOA	100	80	100	100	100	87	94
FMP+BnB	50	80	90	60	70	87	74
Ours	100	100	100	100	100	100	**100**

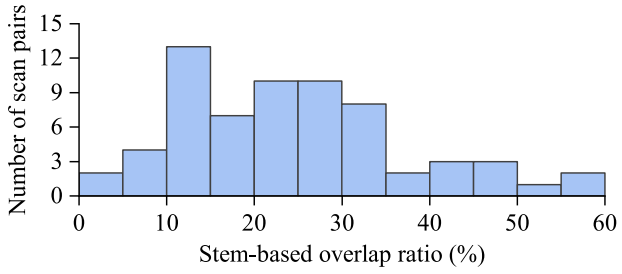

Fig. 12. Histogram of stem-based overlap ratios on all scan pairs of the three test datasets.

that intends to accelerate the coarse registration of general TLS scans, the running time of our method is only 1/6 of it on average.

5.3.3. Robustness

We have also tried to exhaustively register all scan pairs of each plot to understand the robustness of our method concerning overlap ratio. The three datasets form a total of 65 scan pairs. Some of these scan pairs have a small overlap (see Fig. 12), which usually fails marker-free registration methods. By providing simplified stem maps (~ 50 trees/scan), TOA registered 59 scan pairs while Fast-TOA succeed on 61. FMP+BnB failed in aligning almost a quarter of scan pairs (with only 48 successful) due to the limited overlap of the input scan pairs. In contrast, our method managed to register all 65 raw scan pairs. We have recorded the success rate of all competing methods on the coarse registration of each plot. The results are reported in Table 5, from which we can see that the forest-oriented registration methods (i.e., TOA, Fast-TOA, and ours) are more robust than the general local descriptor-based methods (i.e., FMP+BnB). Among these methods, our method has the highest successful registration rate (i.e., no failure cases).

In addition to the registration errors reported in Table 4, we also recorded the pointwise errors for the registration of all other scan pairs in each plot of the three test datasets. Our method achieves the best performance on 47.5% (19 out of 40) scan pairs. Intuitive visualization of the comparison results is demonstrated in Fig. 13.

5.4. Ablation study

5.4.1. Stem mapping with/without the verticality constraint

In the stem mapping stage, we apply a verticality constraint to filter out understory points, which speeds up the subsequent stem modeling process. We have conducted an ablation study to validate the effects of this constraint. Specifically, we ran our stem mapping algorithm with and without the verticality constraint and evaluated the results based on the ground truth (i.e., manually identified stem positions). Table 6 reports the results. We can see that the verticality constraint accelerated the stem mapping process. On the other hand, even without the verticality constraint, our method still correctly registered all scan pairs. The results indicate that our approach is not limited to vertical trees.

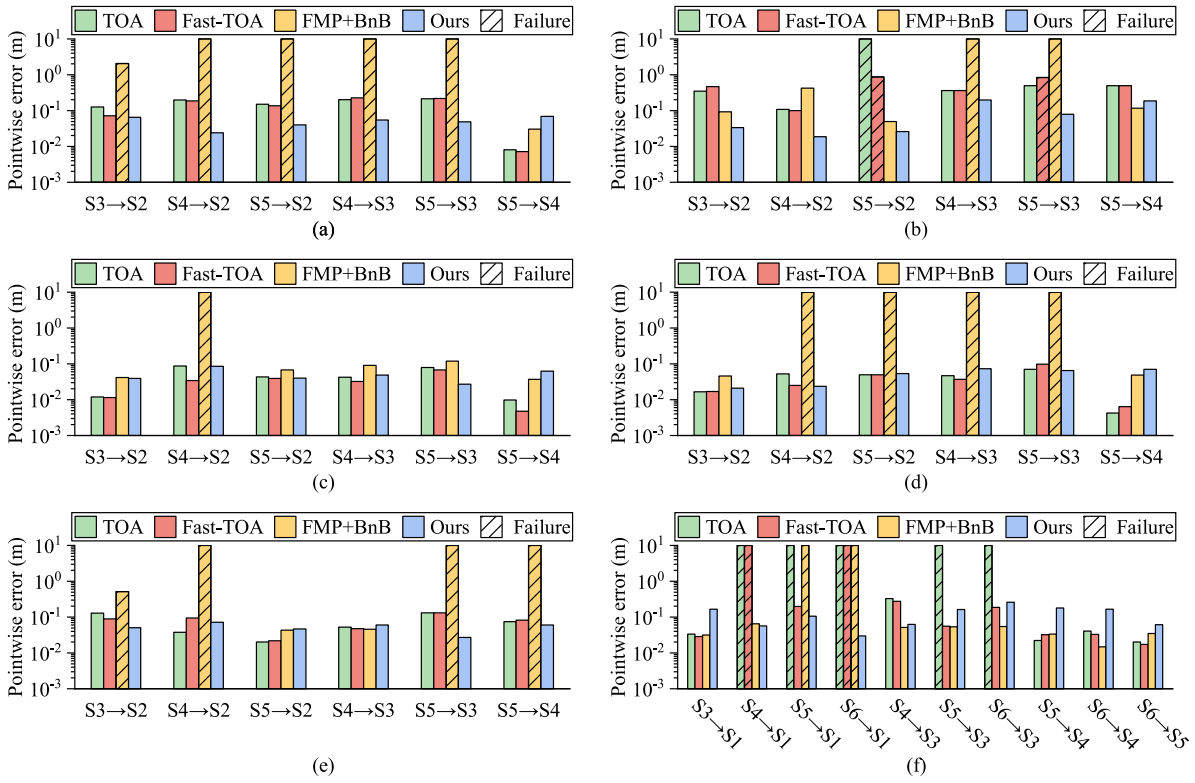

Fig. 13. Pointwise errors (in logarithmic scale) for the registration of all other scan pairs in each plot of the three test datasets (in addition to the registration accuracy reported in Table 4). (a) to (d) Plots #1–#4 of the Tongji-Trees dataset, respectively. (e) WHU-FGI (Dong et al., 2020). (f) ETH-Trees (Theiler et al., 2015). For the visualization purpose, the error value for any failed registration with a pointwise error exceeding 10 m is truncated to 10 m.

Table 6
Comparison of stem mapping with and without the verticality constraint. e_p denotes the pointwise error of coarse registration results.

Plot ID	Verticality	Running time (s)	Success rate (%)	e_p (coarse) (cm)
Plot #1	✓	7.9	100	6.2
	✗	11.8	100	5.6
Plot #2	✓	12.2	100	6.2
	✗	94.1	100	13.4
Plot #3	✓	5.8	100	3.6
	✗	91.0	100	4.1
Plot #4	✓	6.5	100	3.1
	✗	38.4	100	2.6
WHU-FGI	✓	7.7	100	7.6
	✗	102.4	100	8.1
ETH-Trees	✓	6.1	100	8.7
	✗	85.6	100	25.1

Table 7
Accuracy comparison of stem matching algorithms implemented in TOA (Kelbe et al., 2016a), Fast-TOA (Tremblay and Béland, 2018), and ours. Using simplified stem maps (SSM) as input, all matching methods successfully aligned all scan pairs from the side scans of each plot to the central ones. e_p denotes the pointwise error, while e_R and e_t represent the two components (i.e., rotation and translation errors) of matrix-based errors, respectively.

Plot ID	Method	Coarse registration errors		
		e_p (cm)	e_R (mrad)	e_t (cm)
Plot #1	TOA	11.0	10.9	8.7
	Fast-TOA	9.4	9.2	6.7
	SSM + ours	**4.4**	0.7	4.5
Plot #2	TOA	27.8	22.4	21.4
	Fast-TOA	27.3	24.7	19.7
	SSM + ours	**20.5**	2.0	20.3
Plot #3	TOA	1.4	1.5	1.3
	Fast-TOA	**1.3**	1.3	1.2
	SSM + ours	3.2	0.9	3.2
Plot #4	TOA	3.9	3.6	3.8
	Fast-TOA	3.7	3.6	3.5
	SSM + ours	**1.0**	0.3	1.0
WHU-FGI	TOA	7.6	7.6	5.5
	Fast-TOA	6.4	7.5	3.7
	SSM + ours	**5.0**	1.0	5.0
ETH-Trees	TOA	20.7	22.8	18.2
	Fast-TOA	22.2	20.0	20.0
	SSM + ours	**6.5**	3.5	6.0
Average	TOA	12.1	11.5	9.8
	Fast-TOA	11.7	11.1	9.1
	SSM + ours	**6.8**	1.4	6.6

5.4.2. Comparison with stem matching methods

We performed an ablation study to investigate the efficacy of our matching approach. Specifically, we registered side scans of each plot to the central ones by utilizing the tree positions of simplified stem maps (SSM) used by TOA/Fast-TOA algorithms as the input for our matching approach. Table 7 reports the results. Our matching approach produced the most accurate results on five out of six test plots, showing the best overall performance with an average pointwise error of 6.8 cm.

5.4.3. 6-DoF registration versus 4-DoF registration

As has been explained in Section 3.3, our method allows both 6-DoF registration and 4-DoF registration for well-leveled TLS forest scans, and for such data, the 4-DoF solution is preferred. To understand the effectiveness of both the 6-DoF registration and the 4-DoF registration strategies, we compared them using the three test datasets. This comparison has revealed that the coarse registration results from both the 6-DoF and the 4-DoF strategies are sufficiently accurate for the subsequent ICP fine registration to converge. We further computed the horizontal and vertical deviations of the pointwise errors and depicted

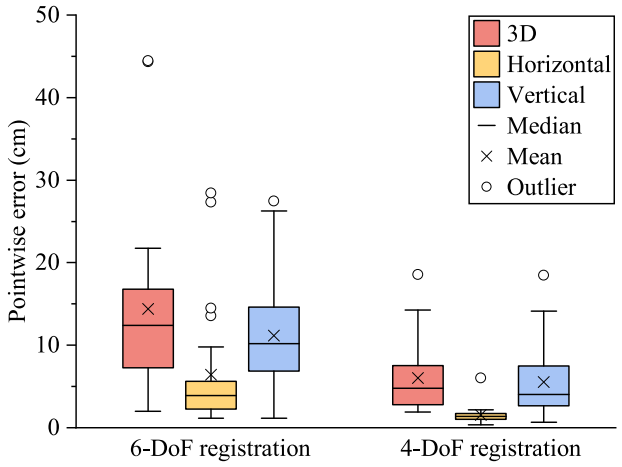

Fig. 14. Box plot showing the horizontal and vertical deviations of pointwise registration errors, for both the 6-DoF registration and the 4-DoF registration. "3D" denotes the pointwise error, while "Horizontal" and "Vertical" represent the horizontal and vertical components of the pointwise error, respectively. "Mean" and "Median" are the respective average and middle values of all pointwise errors. "Outlier" is defined as an error value located outside the whiskers of the box plot.

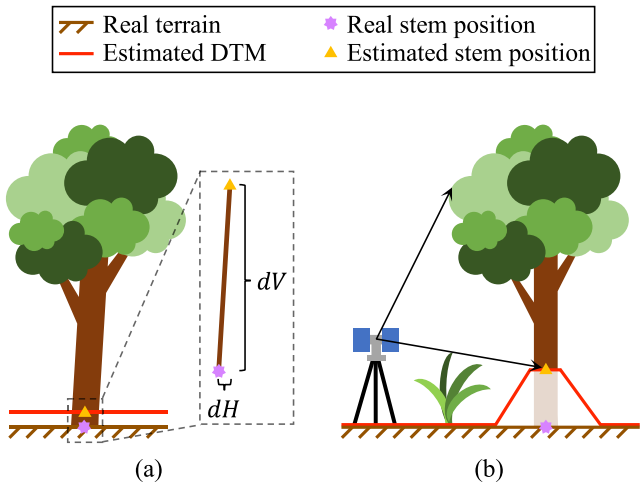

Fig. 15. Inaccuracy in the extracted stem positions. (a) The vertical component of stem positions is sensitive to the change of DTM. (b) Insufficient ground data (due to occlusions) results in a large error in the z coordinate of the stem position.

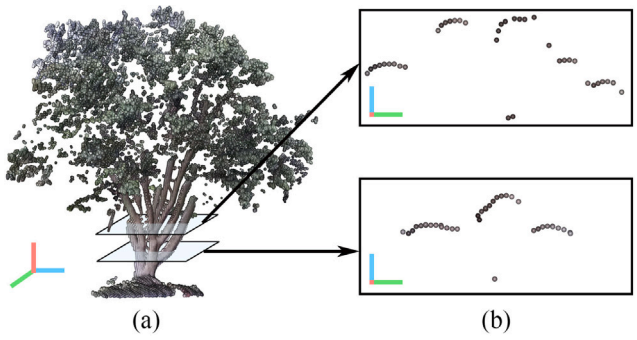

Fig. 16. A tree whose stem cannot be modeled using a cylinder. (a) Point cloud. (b) Two horizontal cross-sections of the point cloud.

them in a box plot shown in Fig. 14. From this figure, we can see that the 4-DoF registration results are more accurate than those from the 6-DoF registration, and the 4-DoF registration strategy is remarkably accurate in the horizontal direction (with a mean horizontal pointwise error of only 1.5 cm). The 4-DoF strategy is more accurate because it takes the advantage of well-leveled scans.

It is also worth noting that the registration errors in the vertical direction are greater than in the horizontal direction. This is a common limitation of the stem-based methods (Liu et al., 2017; Kelbe et al., 2016a; Tremblay and Béland, 2018; Dai et al., 2020; Ge et al., 2021) because stem positions are more accurate in the horizontal direction than those in the vertical direction. In other words, the z coordinates of the extracted stem positions are more sensitive to the change of DTM, compared with their x and y coordinates, as illustrated in Fig. 15(a). The registration results depicted as outliers in Fig. 14 were obtained for the scans in Plot #2 and ETH-Trees where the ground near stems was severely occluded by understory vegetation, leading to large errors in the z coordinates of the stem positions. This can be explained from the illustration shown in Fig. 15(b). Such a test indicated that by decomposing the registration problem into two sub-problems, i.e., horizontal alignment and vertical alignment, the 4-DoF strategy can take advantage of the accurate horizontal coordinates of stem positions.

5.5. Limitation

Our stem mapping algorithm exploits cylinder primitives to extract tree stems, which may not handle tree stems with complex structures, e.g., tree stems consisting of multiple branches from the bottom (see Fig. 16). Since the large proportion of trees in the tested forest types has a single main stem and the complex stems will be treated as outliers and filtered out in the global matching step, we have not encountered such issues in our experiments. However, our method will fail if the majority of the trees have such complex structures.

6. Conclusion

We have introduced an efficient and robust method to align TLS point clouds captured in forest environments. Our method accelerates the current stem-based registration strategy in both stages by proposing a fast stem mapping method as well as a robust and efficient stem matching algorithm requiring only the extracted stem positions. Compared to existing forest-oriented registration techniques, our method performs on par or better regarding effectiveness and robustness while outperforming them significantly in terms of efficiency. We have also released a new benchmark dataset to the community to enable reliable evaluation and comparison of registration algorithms for forest point clouds.

Our method uses object-level primitives (i.e., tree stems) for registration, which has shown excellent performance. In future work, we would like to integrate this approach into frameworks for registration of multiview data, co-registration of multiplatform data, and fusion of spatial–temporal data.

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.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (No. 41974213). Zexin Yang was supported by the China Scholarship Council.

References

Aiger, D., Mitra, N.J., Cohen-Or, D., 2008. 4-points congruent sets for robust pairwise surface registration. ACM Trans. Graph. 1–10.

Albarelli, A., Rodola, E., Torsello, A., 2010. A game-theoretic approach to fine surface registration without initial motion estimation. In: 2010 IEEE Conf. Comput. Vis. Pattern Recognit., CVPR. IEEE, pp. 430–437.

Aldoma, A., Tombari, F., Di Stefano, L., Vincze, M., 2012. A global hypotheses verification method for 3d object recognition. In: Eur. Conf. Comput. Vis., ECCV. Springer, pp. 511–524.

Behley, J., Steinhage, V., Cremers, A.B., 2015. Efficient radius neighbor search in three-dimensional point clouds. In: 2015 IEEE Int. Conf. Rob. Autom., ICRA. IEEE, pp. 3625–3630.

Besl, P.J., McKay, N.D., 1992. A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. 14 (2), 239–256.

Cai, Z., Chin, T.-J., Bustos, A.P., Schindler, K., 2019. Practical optimal registration of terrestrial LiDAR scan pairs. ISPRS J. Photogramm. Remote Sens. 147, 118–131.

Calders, K., Adams, J., Armston, J., Bartholomeus, H., Bauwens, S., Bentley, L.P., Chave, J., Danson, F.M., Demol, M., Disney, M., et al., 2020. Terrestrial laser scanning in forest ecology: Expanding the horizon. Remote Sens. Environ. 251, 112102.

Chen, H., Bhanu, B., 2007. 3D free-form object recognition in range images using local surface patches. Pattern Recognit. Lett. 28 (10), 1252–1262.

Chen, S., Nan, L., Xia, R., Zhao, J., Wonka, P., 2019. PLADE: A plane-based descriptor for point cloud registration with small overlap. IEEE Trans. Geosci. Remote Sens. 58 (4), 2530–2540.

Dagum, L., Menon, R., 1998. Openmp: an industry standard API for shared-memory programming. IEEE Comput. Sci. Eng. 5 (1), 46–55.

Dai, W., Yang, B., Liang, X., Dong, Z., Huang, R., Wang, Y., Li, W., 2019. Automated fusion of forest airborne and terrestrial point clouds through canopy density analysis. ISPRS J. Photogramm. Remote Sens. 156, 94–107.

Dai, W., Yang, B., Liang, X., Dong, Z., Huang, R., Wang, Y., Pyörälä, J., Kukko, A., 2020. Fast registration of forest terrestrial laser scans using key points detected from crowns and stems. Int. J. Digit. Earth 13 (12), 1585–1603.

Demantké, J., Vallet, B., Paparoditis, N., 2012. Streamed vertical rectangle detection in terrestrial laser scans for facade database production. In: ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci, vol. 1. pp. 99–104.

Disney, M., 2019. Terrestrial LiDAR: a three-dimensional revolution in how we look at trees. New Phytol. 222 (4), 1736–1741.

Dong, Z., Liang, F., Yang, B., Xu, Y., Zang, Y., Li, J., Wang, Y., Dai, W., Fan, H., Hyyppä, J., et al., 2020. Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. ISPRS J. Photogramm. Remote Sens. 163, 327–342.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381–395.

Forsman, M., Börlin, N., Holmgren, J., 2012. Estimation of tree stem attributes using terrestrial photogrammetry. In: Int. Arch. Photogramm., Remote Sens. Spat. Inf. Sci. - ISPRS Arch.. Copernicus Gesellschaft, pp. B5–261.

Forsman, P., Halme, A., 2005. 3-D mapping of natural environments with trees by means of mobile perception. IEEE Trans. Robot. 21 (3), 482–490.

Ge, X., 2017. Automatic markerless registration of point clouds with semantic-keypoint-based 4-points congruent sets. ISPRS J. Photogramm. Remote Sens. 130, 344–357.

Ge, X., Hu, H., 2020. Object-based incremental registration of terrestrial point clouds in an urban environment. ISPRS J. Photogramm. Remote Sens. 161, 218–232.

Ge, X., Zhu, Q., Huang, L., Li, S., Li, S., 2021. Global registration of multiview unordered forest point clouds guided by common subgraphs. IEEE Trans. Geosci. Remote Sens..

Gressin, A., Mallet, C., Demantké, J., David, N., 2013. Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. ISPRS J. Photogramm. Remote Sens. 79, 240–251.

Guan, H., Su, Y., Hu, T., Wang, R., Ma, Q., Yang, Q., Sun, X., Li, Y., Jin, S., Zhang, J., et al., 2019. A novel framework to automatically fuse multiplatform LiDAR data in forest environments based on tree locations. IEEE Trans. Geosci. Remote Sens. 58 (3), 2165–2177.

Guan, H., Su, Y., Sun, X., Xu, G., Li, W., Ma, Q., Wu, X., Wu, J., Liu, L., Guo, Q., 2020. A marker-free method for registering multi-scan terrestrial laser scanning data in forest environments. ISPRS J. Photogramm. Remote Sens. 166, 82–94.

Haala, N., Reulke, R., Thies, M., Aschoff, T., 2004. Combination of terrestrial laser scanning with high resolution panoramic images for investigations in forest applications and tree species recognition. In: Proceedings of the ISPRS Working Group V/1, PART 5/W16.

Hauglin, M., Lien, V., Næsset, E., Gobakken, T., 2014. Geo-referencing forest field plots by co-registration of terrestrial and airborne laser scanning data. Int. J. Remote Sens. 35 (9), 3135–3149.

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

Henning, J.G., Radtke, P.J., 2008. Multiview range-image registration for forested scenes using explicitly-matched tie points estimated from natural surfaces. ISPRS J. Photogramm. Remote Sens. 63 (1), 68–83.

Huang, R., Yao, W., Xu, Y., Ye, Z., Stilla, U., 2022. Pairwise point cloud registration using graph matching and rotation-invariant features. IEEE Geosci. Remote Sens. Lett. 19, 1–5.

Hyyppä, E., Hyyppä, J., Hakala, T., Kukko, A., Wulder, M.A., White, J.C., Pyörälä, J., Yu, X., Wang, Y., Virtanen, J.-P., et al., 2020. Under-canopy UAV laser scanning for accurate forest field measurements. ISPRS J. Photogramm. Remote Sens. 164, 41–60.

Hyyppä, E., Muhojoki, J., Yu, X., Kukko, A., Kaartinen, H., Hyyppä, J., 2021. Efficient coarse registration method using translation-and rotation-invariant local descriptors towards fully automated forest inventory. ISPRS Open J. Photogramm. Remote Sens. 2, 100007.

Kelbe, D., Van Aardt, J., Romanczyk, P., Van Leeuwen, M., Cawse-Nicholson, K., 2015. Single-scan stem reconstruction using low-resolution terrestrial laser scanner data. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 8 (7), 3414–3427.

Kelbe, D., Van Aardt, J., Romanczyk, P., Van Leeuwen, M., Cawse-Nicholson, K., 2016a. Marker-free registration of forest terrestrial laser scanner data pairs with embedded confidence metrics. IEEE Trans. Geosci. Remote Sens. 54 (7), 4314–4330.

Kelbe, D., Van Aardt, J., Romanczyk, P., Van Leeuwen, M., Cawse-Nicholson, K., 2016b. Multiview marker-free registration of forest terrestrial laser scanner data with embedded confidence metrics. IEEE Trans. Geosci. Remote Sens. 55 (2), 729–741.

van Laarhoven, P.J.M., Aarts, E.H.L., 1987. Simulated annealing. In: Simulated Annealing: Theory and Applications. Springer, pp. 7–15.

Li, Z., Zhang, X., Tan, J., Liu, H., 2021. Pairwise coarse registration of indoor point clouds using 2D line features. ISPRS Int. J. Geo-Inf. 10 (1), 26.

Liang, X., Hyyppä, J., 2013. Automatic stem mapping by merging several terrestrial laser scans at the feature and decision levels. Sensors 13 (2), 1614–1634.

Liang, X., Hyyppä, J., Kaartinen, H., Lehtomäki, M., Pyörälä, J., Pfeifer, N., Holopainen, M., Brolly, G., Francesco, P., Hackenberg, J., et al., 2018. International benchmarking of terrestrial laser scanning approaches for forest inventories. ISPRS J. Photogramm. Remote Sens. 144, 137–179.

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

Liang, X., Litkey, P., Hyyppä, J., Kaartinen, H., Vastaranta, M., Holopainen, M., 2011. Automatic stem mapping using single-scan terrestrial laser scanning. IEEE Trans. Geosci. Remote Sens. 50 (2), 661–670.

Liu, J., Liang, X., Hyyppä, J., Yu, X., Lehtomäki, M., Pyörälä, J., Zhu, L., Wang, Y., Chen, R., 2017. Automated matching of multiple terrestrial laser scans for stem mapping without the use of artificial references. Int. J. Appl. Earth Obs. Geoinf. 56, 13–23.

Maiseli, B., Gu, Y., Gao, H., 2017. Recent developments and trends in point set registration methods. J. Vis. Commun. Image Represent. 46, 95–106.

Mian, A.S., Bennamoun, M., Owens, R., 2006. Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE Trans. Pattern Anal. Mach. Intell. 28 (10), 1584–1601.

Myronenko, A., Song, X., 2010. Point set registration: Coherent point drift. IEEE Trans. Pattern Anal. Mach. Intell. 32 (12), 2262–2275.

Othmani, A., Piboule, A., Krebs, M., Stolz, C., Voon, L.L.Y., 2011. Towards automated and operational forest inventories with T-lidar. In: 11th Int. Conf. LiDAR Appl. Assess. for. Ecosyst. (SilviLaser 2011).

Pham, T.T., Eich, M., Reid, I., Wyeth, G., 2016. Geometrically consistent plane extraction for dense indoor 3D maps segmentation. In: 2016 IEEE Int. Conf. Intell. Rob. Syst., IROS. IEEE, pp. 4199–4204.

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.

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.

Rusu, R.B., Blodow, N., Beetz, M., 2009a. Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation. IEEE, pp. 3212–3217.

Rusu, R.B., Cousins, S., 2011. 3D is here: Point cloud library (pcl). In: 2011 IEEE Int. Conf. Rob. Autom.. ICRA, IEEE, pp. 1–4.

Rusu, R.B., Meeussen, W., Chitta, S., Beetz, M., 2009b. Laser-based perception for door and handle identification. In: 2009 Int. Conf. Adv. Rob., ICAR. IEEE, pp. 1–8.

Sanchez, J., Denis, F., Checchin, P., Dupont, F., Trassoudaine, L., 2017. Global registration of 3D LiDAR point clouds based on scene features: Application to structured environments. Remote Sens. 9 (10), 1014.

Sanchez, J., Denis, F., Coeurjolly, D., Dupont, F., Trassoudaine, L., Checchin, P., 2020. Robust normal vector estimation in 3D point clouds through iterative principal component analysis. ISPRS J. Photogramm. Remote Sens. 163, 18–35.

Shao, J., Zhang, W., Mellado, N., Wang, N., Jin, S., Cai, S., Luo, L., Lejemble, T., Yan, G., 2020. SLAM-aided forest plot mapping combining terrestrial and mobile laser scanning. ISPRS J. Photogramm. Remote Sens. 163, 214–230.

Sorkine-Hornung, O., Rabinovich, M., 2017. Least-squares rigid motion using svd. Computing 1 (1), 1–5.

Theiler, P.W., Wegner, J.D., Schindler, K., 2015. Globally consistent registration of terrestrial laser scans via graph optimization. ISPRS J. Photogramm. Remote Sens. 109, 126–138.

Tremblay, J.-F., Béland, M., 2018. Towards operational marker-free registration of terrestrial lidar data in forests. ISPRS J. Photogramm. Remote Sens. 146, 430–435.

Weinmann, M., Jutzi, B., Mallet, C., 2013. Feature relevance assessment for the semantic interpretation of 3D point cloud data. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 5 (W2), 1.

Xia, S., Wang, C., Pan, F., Xi, X., Zeng, H., Liu, H., 2015. Detecting stems in dense and homogeneous forest using single-scan TLS. Forests 6 (11), 3923–3945.

Yang, B., Dai, W., Dong, Z., Liu, Y., 2016a. Automatic forest mapping at individual tree levels from terrestrial laser scanning point clouds with a hierarchical minimum cut method. Remote Sens. 8 (5), 372.

Yang, B., Dong, Z., Liang, F., Liu, Y., 2016b. Automatic registration of large-scale urban scene point clouds based on semantic feature points. ISPRS J. Photogramm. Remote Sens. 113, 43–58.

Ye, W., Qian, C., Tang, J., Liu, H., Fan, X., Liang, X., Zhang, H., 2020. Improved 3D stem mapping method and elliptic hypothesis-based dbh estimation from terrestrial laser scanning data. Remote Sens. 12 (3), 352.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. Remote Sens. 8 (6), 501.

Zhao, B., Chen, X., Le, X., Xi, J., Jia, Z., 2021. A comprehensive performance evaluation of 3-D transformation estimation techniques in point cloud registration. IEEE Trans. Instrum. Meas. 70, 1–14.

Zhong, Y., 2009. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. IEEE, pp. 689–696.