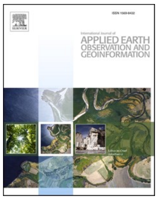
Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

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



Hybrid geometry sets for global registration of cross-source geometric data

Minglei Li [a,*], Shu Peng [a], Liangliang Nan [b]

[a] *College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China*
[b] *Urban Data Science, Delft University of Technology, Delft 2628BL, Netherlands*

ARTICLE INFO

Keywords:
Cross-source geometric data
Global registration
Hybrid geometry sets
Scale restoration

ABSTRACT

We propose a concept of hybrid geometry sets for registering cross-source geometric data. Specifically, our method focuses on the coarse registration of geometric data obtained from laser scanning and photogrammetric reconstruction. Due to different characteristics (e.g., variations in noise levels, density, and scales), achieving accurate registration between these data becomes a challenging task. The proposed method uses geometric structures to construct hybrid geometry sets, and the geometric relations between the elements of a hybrid geometry set are encoded in a hybrid feature space. This enables effective and efficient similarity query and correspondence establishment between the hybrid geometry sets. The proposed global registration method works in three steps. Firstly, a set of hybrid geometry sets is constructed using extracted planes and intersection lines. Then the features of the hybrid geometry sets are computed to encode the relative pose and topological relationships between the extracted planes and intersection lines, and their correspondences between the two inputs are established by querying hybrid geometry sets with similar features. Finally, the global registration parameters are calculated using the correspondences, and the registration result is further refined through continuous optimization. The robustness of the method has been evaluated using different real-world cross-source geometric data of urban scenes. Extensive comparisons with state-of-the-art algorithms have also demonstrated its effectiveness.

1. Introduction

Photogrammetry and computer vision techniques have enabled the acquisition of 3D information about the real world in various formats. There are scenarios where different acquisition methods may be used in combination. For instance, when scanning complex urban environments, data obtained from LiDAR scanners and photogrammetric reconstruction techniques can complement each other in terms of scene coverage and geometric details (Brenner, 2005; El-Hakim et al., 2007; Li et al., 2019; Liu et al., 2023). In these scenarios, the registration of cross-source geometric data becomes crucial for various applications.

The registration of unitary-source geometric data (e.g., LiDAR point clouds) has been extensively studied (Dong et al., 2020). A current consensus is that the coarse alignment is a crucial yet unsolved problem, where the data are presented in arbitrary unknown relative orientations. To accomplish successful registration, state-of-the-art global registration methods require the extraction of salient features from different data sources. However, it remains an open problem to define reliable features that can be simultaneously extracted and related between different data sources (Ghamisi et al., 2019; Saiti and Theoharis, 2020). In this work, we address the coarse registration of such cross-source geometric data, i.e., laser scans and photogrammetric data in the presentation of dense point clouds for urban scenes with plenty of plane structures.

Fig. 1 shows a pair of cross-source models acquired by a LiDAR scanner (Leica BLK360 mini) and the photogrammetric reconstruction method (Furukawa and Hernández, 2015). The geometric data acquired from LiDAR scanners and photogrammetric technologies exhibit varying levels of detail and accuracy, introducing additional complexities in registration and data fusion (Baltsavias, 1999). The global registration of cross-source geometric data faces the following challenges: 1) Pure image-based 3D reconstruction is known only up to scale. Determining the actual physical scale of 3D reconstruction can be challenging in practice, as it often involves tedious manual specifying known lengths of physical objects. 2) The density within a LiDAR point cloud varies significantly with the distances of the objects, amplifying the inherent density variations between cross-source geometric data. 3) The data loss between cross-source geometric data varies depending on the views, and LiDAR point clouds may suffer from severe data loss and inaccuracy in

* Corresponding author.
E-mail addresses: minglei_li@nuaa.edu.cn (M. Li), ps0272@nuaa.edu.cn (S. Peng).

https://doi.org/10.1016/j.jag.2024.103733
Received 20 November 2023; Received in revised form 5 February 2024; Accepted 22 February 2024
Available online 29 February 2024

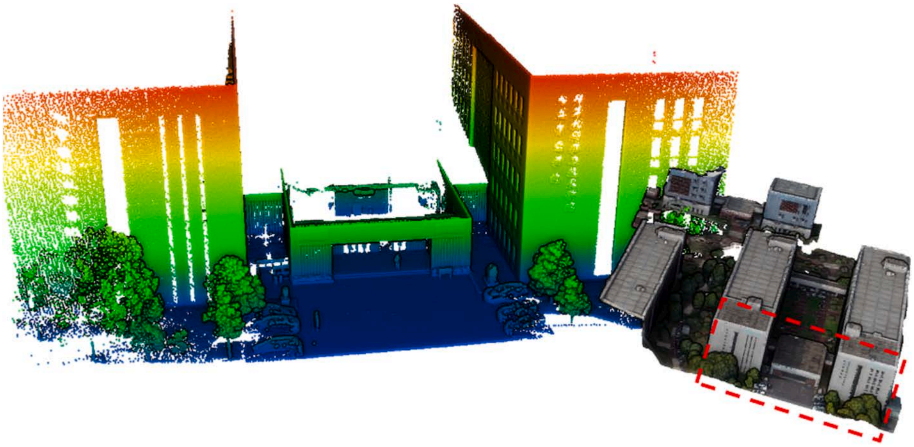


Fig. 1. An example of cross-source geometric data acquired by a LiDAR scanner (left) and the photogrammetric reconstruction method (right). The red rectangle reveals the limited overlapping ratio of the two input sources. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

areas with water or glasses, while image-based reconstructions typically miss texture-less regions. 4) A scan from modern devices can quickly accumulate millions of 3D points, posing a critical scalability requirement for registration methods.

Although the local structures captured in cross-source geometric data might have different appearances and are prone to failing the matching process, such data still encompasses many planar structures (e.g., ground planes, roofs, and facades) that convey the abstract characteristics of the scene, along with their intersection lines encoding the topology of the planar structures. To this end, we propose an automatic and efficient global registration method for cross-source geometric data by introducing **Hybrid Geometry Set** (HGS), which is constructed based on planes and their intersection lines extracted from the data. The features of HGSs are computed based on some geometric angles and line segment proportions, and they convert the initial cross-source geometric data into a unified feature space. In summary, the main contributions of this work are threefold:

- We introduce a concept of HGS that encodes spatial relationships between 3D planar structures and intersection lines. The proposed HGSs establish corresponding relationships without being affected by varying data quality. The geometric elements of HGSs are extracted from cross-source geometric data, forming descriptive feature descriptors. Feature correspondences can be established between cross-source models based on these feature descriptors.
- We propose a mathematical formulation for estimating the scale of the image-based geometric data in addition to recovering the rotation and translation parameters. This global registration accurately aligns the target data with the reference data, providing a reliable initialization for fine registration.
- We introduce a new cross-source dataset that contains LiDAR point clouds and the corresponding photogrammetric point clouds. The dataset includes ground truth transformations and can be used for developing and evaluating global registration methods for cross-source geometric data.

2. Related work

In most cases, the optimal registration can be achieved in two key stages: a global coarse registration stage followed by a local refinement stage. The global coarse registration stage aims to establish an approximate alignment between the target and the reference data, while the local refinement stage further refines the alignment using techniques like Iterative Closest Point (ICP) or its variants (Bae and Lichti, 2004; Rusinkiewicz, 2019; Zhang et al., 2022). Since our work focuses on the global coarse registration of cross-source geometric data, we mainly review the related works in coarse registration and cross-source registration methods.

2.1. Global coarse registration

Pointwise matching methods. The traditional registration methods establish correspondences between two models by extracting and matching key point features, such as FPFH (Rusu et al., 2009), USC (Tombari et al., 2010), and 3D-SIFT (Rister et al., 2017), etc. As the pointwise-matching algorithms might generate some false correspondences, outlier filtering techniques are needed to suppress false matches. For example, the FPFHSAC method (Holz et al., 2015) registers point clouds using FPFH features and a RANSAC process (Fischler and Bolles, 1981) for the robust estimation of transformation parameters. The fast global registration method (FGR) (Zhou et al., 2016) uses FPFH features to generate the initial correspondence set, and then they estimate a set of poses for multi-way registration problems using a synchronized global optimization approach.

Motivated by the advancements in deep learning, some networks are specifically trained to extract the features for 3D matching, such as Deep Closest Point algorithm (Wang and Solomon, 2019), PointNetLK (Aoki et al., 2019), JoKDNet (Wang et al., 2021), and Feature-Metric Registration (Huang et al., 2020). Recent point feature-based or deep learning-based approaches are typically prone to failure due to challenges in extracting discriminative point features from cross-source geometric data, as they have large differences in terms of point densities, resolutions, and noise levels.

Structure querying methods. Matching methods based on 3D structural elements have become increasingly attractive as they are more prominent and stable across different data sources. The 4PCS method (Aiger et al., 2008) is a classic method that associates data based on the 4-point coplanar-set structure. Some variants of 4PCS have been proposed to improve the efficiency and scalability (Theiler et al., 2014; Mellado, et al., 2014; Xu et al., 2019). In urban environments, semantic patterns are useful for establishing correspondences, such as the predefined windows and doors (Thapa et al., 2009; Yan et al., 2016). Yang et al. (2016) present a method using the intersection points of vertical lines and the ground as the semantic feature points. Then, the 3-point geometrical constraints are used to match these semantic feature points. These methods become fragile to establish accurate correspondences when parts of the patterns are missing. Chen et al. (2019) introduce a PLADE algorithm which utilizes a structure-level feature descriptor encoding the relationship between plane/line primitives. Wang et al. (2023) encode the shapes and orientations of extracted local

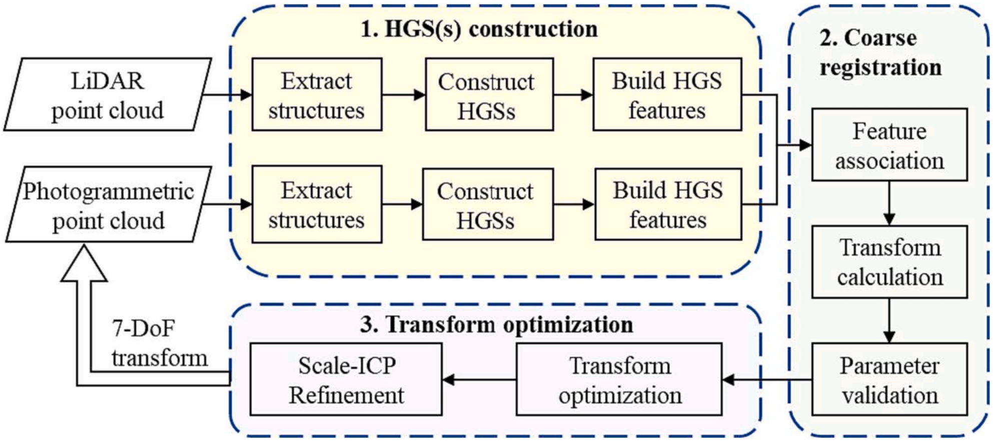


Fig. 2. The workflow of the proposed approach. We first extract structural primitives, from which HGSs are constructed and their features are computed. Subsequently, the coarse registration is obtained by matching the feature vectors of the two point clouds. Finally, the Scale-ICP refinement step further improves the registration transformation.

patches in the feature descriptors, and a transformation can thus be generated from a single correspondence, which leads to a linear complexity. Then, they use a rotation-guided RANSAC algorithm to find the best transformation.

2.2. Cross-source registration

The registration of cross-source geometric data is a special category, where the data pairs could be obtained from different data sources, such as optical imagery, LiDAR scanner, or photogrammetric reconstruction (Mura et al., 2015; Parmehr et al., 2016). In contrast to the unitary-source registration, it requires finding a global aligning transform over the 7 degrees of freedom (DoF) space, composed of a scale, a translation, and a rotation. Even though some registration methods are supposed to recover scale and rotation angles, such as the Scale-ICP method (Zinßer et al., 2005; Ying et al., 2009) and graph-based method (Huang et al., 2017), the large variability of cross-source geometric datasets poses great challenges for these methods.

To restore the relative scale, Mellado et al., (2015) present a descriptor based on Growing Least Squares in logarithmic scale-space. Huang et al. (2021) present a coarse registration method via 3D Fourier transformation. Li and Lafarge (2021) introduce an accumulation space of planar shapes and use a histogram-matching strategy to determine global rotation parameters. This method works only when two datasets having nearly identical observation regions. In Peng et al., (2014), the registration of LiDAR points and photogrammetric points of street views are studied. Under the assumption that the street ground is horizontal, the method first matches the ground of the two data, reducing the degrees of freedom to be solved. Some other works demonstrate cross-source registration as additional experiments within the design of registration methods for unitary-source registration (Xu et al., 2019; Mellado et al., 2014), while the non-uniform scale and varying density of cross-source geometric data have not been sufficiently analysed.

In this work, we observe that middle- and high-level structural features are less susceptible to noise and point density, making them relatively stable and prominent. Inspired by those structure querying methods (Aiger et al., 2008; Chen et al., 2019; Li and Lafarge, 2021), we propose an enhanced feature descriptor that also allows scale estimation and reduces the complexity of the descriptor. Our strategy takes advantage of data abstraction by exploiting the similarities of structural features between cross-source models rather than relying on local point-level features. Specifically, we employ geometric structures such as planes and intersection lines to establish correspondences between cross-source data. These structural features are integrated into the proposed HGSs, and the feature vectors of HGSs are correlated to determine the global registration parameters. After the coarse alignment, the registration result is further refined by a local iterative optimization step that also improves the estimated scale.

3. Proposed method

The algorithm takes as input a pair of point clouds generated from LiDAR scanning and photogrammetric reconstruction, respectively. In our study, the LiDAR point cloud serves as the reference data $\boldsymbol{P}^{\mathrm{r}} = \{\boldsymbol{p}_i^{\mathrm{r}}\}$, which remains fixed throughout the transformation process, and the photogrammetric point cloud is considered as the target data $\boldsymbol{P}^{\mathrm{t}} = \{\boldsymbol{p}_i^{\mathrm{t}}\}$. The transformation relationship can be written as:

$$\boldsymbol{p}_i^{\mathrm{r}} = s\boldsymbol{R}\boldsymbol{p}_i^{\mathrm{t}} + \boldsymbol{t} \tag{1}$$

where $s \in \mathbb{R}$ is the scale factor. $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix, and $\boldsymbol{t} \in \mathbb{R}^3$ is the translation vector that will be applied to the target point cloud. This transformation problem involves 7 degrees of freedom (7 DoF).

The proposed approach has a workflow consisting of three main steps, illustrated in Fig. 2:

- HGSs construction. We first extract structural primitives (i.e., planes and their intersection lines) from the two input point clouds. These primitives are then utilized to construct HGSs, with each set characterized by a specifically designed feature vector.
- Coarse registration. We associate the HGSs by matching their respective feature vectors between two data sources. These matching pairs lead to multiple coarse transform candidates. Then, redundant candidates are carefully eliminated in the transform parameter space through a clustering process.
- Transform optimization. An optimal coarse transform is determined with the highest matching scores among all transform candidates, and then the optimal transform is updated with all supporting matching pairs. Finally, a Scale-ICP refinement method is used to further improve the transform parameters.

3.1. HGSs construction

As the angle measures and ratios of corresponding line segments remain unchanged during similarity transformations, we explore using the structural primitives to design the structural features. The proposed HGSs are hybrid geometric primitive sets composed of planes and lines. The intersection angles and segments of primitives can be effectively

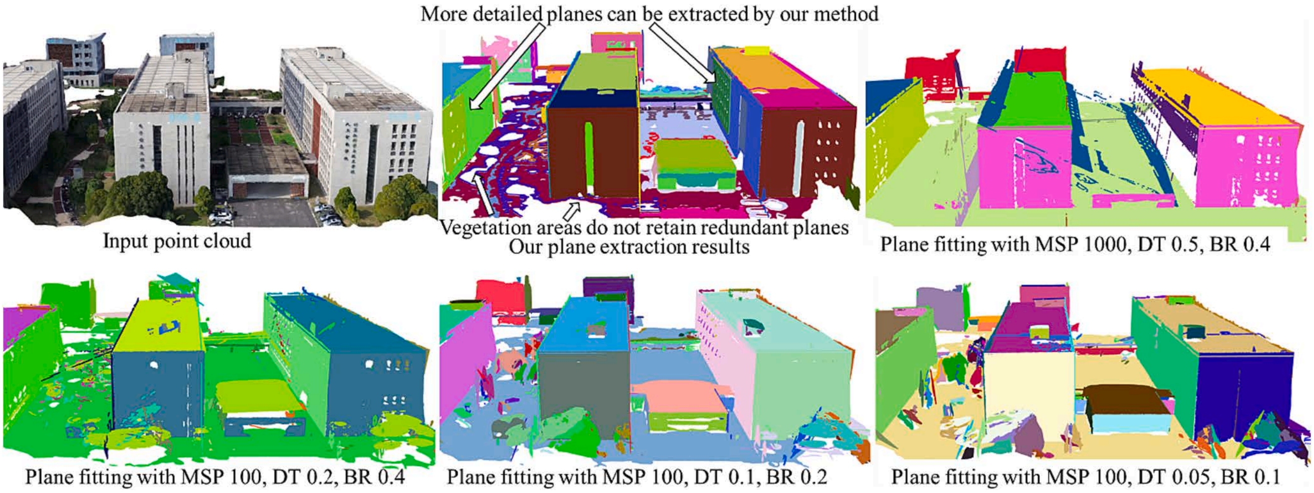

More detailed planes can be extracted by our method
Vegetation areas do not retain redundant planes
Input point cloud
Our plane extraction results
Plane fitting with MSP 1000, DT 0.5, BR 0.4
Plane fitting with MSP 100, DT 0.2, BR 0.4
Plane fitting with MSP 100, DT 0.1, BR 0.2
Plane fitting with MSP 100, DT 0.05, BR 0.1

Fig. 3. Comparison between our method and RANSAC-based fitting method with different parameters (MSP: Minimal support points, DT: distance threshold, BR: bitmap resolution).

employed as the foundation for constructing descriptors.

(1) Structural primitive extraction

We extract planes from both point clouds, which provide a structural representation of the scene. Then, the intersection lines of each pair of nearby planes are computed, as these lines convey the connections between the planes. To facilitate reliable plane extraction, we introduce an automatic method that is dedicated to handling point clouds of arbitrary scales based on the growth of super-voxels. The method consists of the following three main steps:

a. An octree structure is constructed on the point cloud, where the leaf nodes at a predetermined depth serve as the initial cluster seeds. These seeds, guided by the octree structure, have a uniform distribution and provide a rough representation of the overall shape of the input data.
b. A region-growing method is employed to generate over-segmentation results, which are represented by a set of super-voxels with internal structural consistency. During this step, points are progressively grouped into the group of adjacent seeds that exhibit closer distance and normal directions. After each round of growth, the position and normal of each seed are updated to the mean values of points in the same group. The growth and update steps are iterated until the seed positions remain unchanged as the convergence has been achieved, resulting in the formation of numerous super-voxels.
c. From the over-segmented voxels, we detect the coplanar shapes by merging the neighbour voxels whose normal vectors have an angle less than a threshold (a 10-degree threshold is chosen). We use least square fitting to calculate the plane parameters of the merged patches. The output primitives are clusters each associated with a supporting plane, i.e., the best plane to its inlier points.

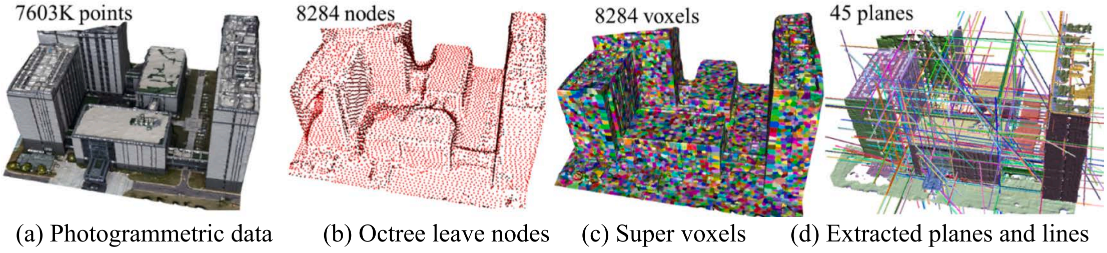
In fact, some other plane extraction methods (Schnabel et al., 2007; Li and Lafarge, 2021) are also optional, as long as enough reliable plane elements can be found. However, by leveraging the inherent consistency within voxels, our method simplifies the overall process, requiring fewer parameter tuning procedures than traditional plane fitting algorithms. In addition, by constraining the angle between the normal vectors of voxels, we can eliminate redundant planes fitted by points on vegetation. Fig. 3 shows a visual comparison between our method and other methods in plane extraction, and we can observe that the traditional plane fitting method requires complex parameter tuning.

The voxel-based method takes advantage of the initial over-segmentation to accumulate planar segments, which is robust against noise and outliers presented in the data. Additionally, by ensuring the uniformity of the initial seeds, the resulting voxels generated from the octree structure have similar sizes, as shown in Fig. 4 (c). After the planes have been extracted, we compute the lines by intersecting pairs of nearby planes. An example of the extracted planes and intersection lines is demonstrated in Fig. 4 (d). Although any pair of non-parallel planes can result in an intersection line, we only keep the lines located within the enlarged bounding box (i.e., actual bounding box size multiplied by a factor of 1.2) of the point cloud.

Now, a set of lines $\{l_1, l_2, \cdots, l_m\}$ is derived from planes, and each line is associated with its two supporting planes, i.e., l_i (Π_{i1}, Π_{i2}). In the next step, the HGS feature vectors will be computed based on these planes and the intersection lines.

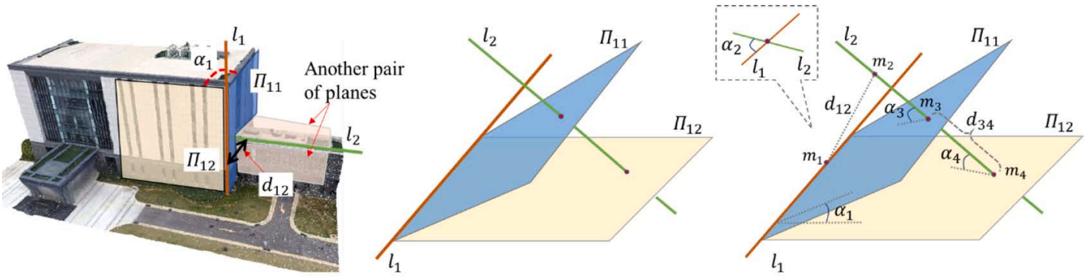
(2) HGSs construction

As shown in Fig. 5(a), a single HGS is a set of geometric primitives



(a) Photogrammetric data (b) Octree leave nodes (c) Super voxels (d) Extracted planes and lines

Fig. 4. An example of the extracted geometric primitives.



(a) Geometric primitives of a HGS (b) A single HGS (c) Feature elements defined on the HGS

Fig. 5. An illustration of the proposed HGS structure and its feature elements.

including two nearby non-parallel planes (i.e. Π_{11} and Π_{12}) and their intersection line l_1, as well as another intersection line l_2 derived from another pair of non-parallel planes. Fig. 5(b) is an illustration of a single HGS, where the two intersection lines are not parallel, represented by l_1 and l_2, respectively. The accompanying planes of l_1 are denoted by Π_{11} and Π_{12}. Compared to geometric sets such as 4PCS (Aiger et al., 2008) and K4PCS (Theiler et al., 2014), the HGS-based method has a more intuitive meaning in extracting structures.

Intuitively, a number of HGSs can be found for a point cloud. Then, HGSs can be matched between cross-source models through geometric angles and ratios of segment lengths without being affected by the different physical scales of the point clouds.

(3) HGS features

Based on the HGS structure (as illustrated in Fig. 5(c)), we define the HGS feature to encode the spatial relationships of the structural primitives in an HGS, which is a 5-dimensional feature vector, i.e.,

$$f^5 = [\lambda \bullet r, \alpha_1, \alpha_2, \alpha_3, \alpha_4]^T \tag{2}$$

Here the vector elements are defined as

$$\begin{aligned} r &= d_{12}/d_{34} \\ \alpha_1 &= \angle(\Pi_{11}, \Pi_{12}) \\ \alpha_2 &= \angle(l_1, l_2) \qquad . \\ \alpha_3 &= \angle(l_2, \Pi_{11}) \\ \alpha_4 &= \angle(l_2, \Pi_{12}) \end{aligned} \tag{3}$$

Here $\angle(*, *)$ denotes the angle between two primitives. d_{12} is the length between l_1 and l_2, and d_{34} is the length between two intersection points m_3 and m_4. λ is a regularization parameter that balances the unit between angle and distance ratio when querying matching pairs. The HGS feature not only provides an effective representation of the input data for matching but also enhances the robustness of matching.

3.2. Coarse registration

(1) Feature association

We use the L_2 norm to calculate the distance C between two HGS feature vectors, i.e.,

$$C = \|f_i^{\mathrm{r}} - f_j^{\mathrm{t}}\|_2 \tag{4}$$

where f_i^{r} represents the i-th HGS feature vector in the reference point cloud, and f_j^{t} represents the j-th feature vector in the target point cloud (see Eq. (2)). Only the HGS pairs with small distance (i.e., C) values can be used for identifying potential correspondences between the two point clouds, we filter out all HGS feature pairs with a distance value greater than a predefined threshold that is empirically set to 0.04 in our experiments.

(2) Transform calculation

Now, we have obtained a set of potential matching HGS pairs, and each pair of HGSs can lead to a 7-DoF transform candidate. In the following, we explain how to calculate the scaling factor, rotation matrix, and translation vector from a matching pair of HGSs.

Scaling factor estimation. Given a matching HGS pair, the scaling factor $\hat{s}$ is directly estimated by calculating the ratio of the distances between the corresponding intersection lines, i.e.,

$$\hat{s} = d_{12}^{\mathrm{r}}/d_{12}^{\mathrm{t}} \tag{5}$$

where d_{12}^{r} and d_{12}^{t} are the distances between the pairs of lines in the reference and the target HGSs (see Fig. 5 (c)).

Rotation matrix estimation. From Fig. 5 (a), it can be observed that once two planes and two lines are selected for an HGS, the spatial relationship between the lines and planes is determined. After two HGSs are matched, the directions of two lines can be determined. Then, we use the unit direction vectors of two pairs of matched intersection lines to estimate the rotation matrix. Since the rotation matrix $\hat{\boldsymbol{R}}$ converting a target unit vector $\boldsymbol{n}^{\mathrm{t}}$ to the reference vector $\boldsymbol{n}^{\mathrm{r}}$ satisfies

$$\boldsymbol{n}^{\mathrm{r}} = \hat{\boldsymbol{R}}\,\boldsymbol{n}^{\mathrm{t}} \tag{6}$$

we use the covariance matrix of the corresponding direction vectors to recover the rotation matrix $\hat{\boldsymbol{R}}$. The covariance matrix can be calculated as

$$\boldsymbol{H} = \sum_{i=1,2} \boldsymbol{n}_{l_i^{\mathrm{t}}} \boldsymbol{n}_{l_i^{\mathrm{r}}}^{\mathrm{T}} \tag{7}$$

where $\boldsymbol{n}_{l_i^{\mathrm{t}}}$ and $\boldsymbol{n}_{l_i^{\mathrm{r}}}$ are the unit direction vectors of l_i^{t} and l_i^{r}, respectively. $\hat{\boldsymbol{R}}$ is estimated by applying the singular value decomposition (SVD) to $\boldsymbol{H}$. As $\boldsymbol{H} = \boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}}$, we get $\hat{\boldsymbol{R}} = \boldsymbol{V}\boldsymbol{U}^{\mathrm{T}}$.

Translation vector estimation. After obtaining the scaling factor and the rotation matrix, the estimated translation vector $\hat{\boldsymbol{t}}$ can be simply calculated using a pair of point correspondences

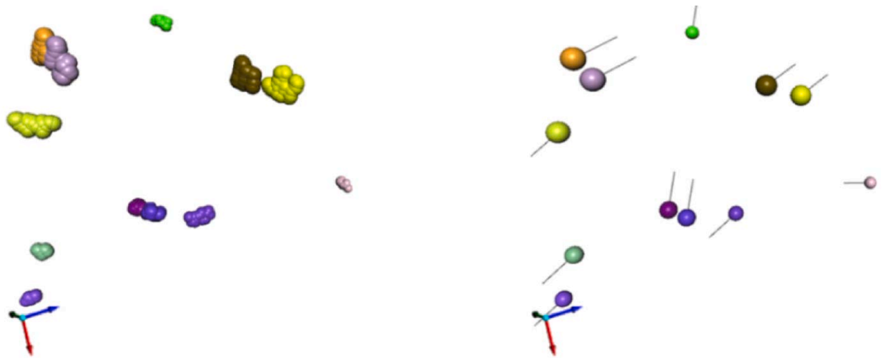
$$\hat{\boldsymbol{t}} = \boldsymbol{p}_{\mathrm{mid}}^{\mathrm{r}} - \hat{s}\hat{\boldsymbol{R}}\boldsymbol{p}_{\mathrm{mid}}^{\mathrm{t}} \tag{8}$$

where $\boldsymbol{p}_{\mathrm{mid}}$ denotes the midpoint of the shortest line segment between l_1 and l_2.

(3) Parameter validation

From the previous process, a large number of candidate transformation matrices have been obtained from the potential HGS correspondences, which contain a significant number of redundant matchings as well as false matchings. The transform candidates, calculated from all matching pairs, can be represented by a set of points in the 7D parameter space. In the following, we introduce a validation procedure that eliminates the redundant transforms.

In the parameter space, we cluster similar transformation matrices



(a) Translation vector groups.

(b) The kept cluster centres.

Fig. 6. Clustering of the candidate transformation parameters in the 3D translation space. Each point in (a) represents a translation vector. Similar transformation parameters have been grouped together, and only their centres have been retained. Subfigure (b) displays these cluster centres, with short segments indicating the direction of a unit direction $[1,0,0]^{\mathrm{T}}$ for visualization.

into groups according to their similarities. The clustering process in the parameter space is operated in two sub-spaces, i.e., a 3D translation space after scaling and a 3D rotation space. The similarities of the translation vectors are defined as the Euclidean distances in the 3D translation space, and the similarities of the rotation matrices are defined as the angular differences between two rotated unit vectors. A simple Euclidean clustering method (Rusu and Cousins, 2011) is used for the clustering of these parameter points in the two 3D spaces. The clustering criterion for grouping translation vectors is based on a distance threshold, i.e., $d_{clust} = 0.001 \bullet D$, where D denotes the diagonal length of the bounding box of the LiDAR point cloud. Then, only the cluster centres of parameter groups are kept as they reduce the variances of different potential transforms.

Fig. 6 (a) depicts the grouping state of the translation vectors, where the translation vectors in different groups are represented by different colour points. In Fig. 6 (b), each point is a kept cluster centre, whose 3D coordinates correspond to the translation vector, and the short segment indicates a direction that is the result of applying the rotation matrix $\widehat{\boldsymbol{R}}$ to a unit direction $[1,0,0]^{\mathrm{T}}$.

Then, each kept transform (i.e., a cluster centre in the parameter space) is applied on the extracted planes to find matching pairs between two data sources based on the coplanar criteria: 1) The angle between the normal vectors of the target plane and the reference plane is less than 5 degrees, and 2) the distance $d_i^{\mathrm{t,r}}$ between the target plane Π_i^{t} and the reference plane Π_i^{r} is less than 0.5 m. Here, $d_i^{\mathrm{t,r}}$ is defined as

$$d_i^{\mathrm{t,r}} = \frac{d\left(p_{\mathrm{c}}^{\mathrm{r}}, \Pi_i^{\mathrm{t}}\right) + d\left(p_{\mathrm{c}}^{\mathrm{t}}, \Pi_i^{\mathrm{r}}\right)}{2} \tag{9}$$

where $p_{\mathrm{c}}^{\mathrm{r}}$ and $p_{\mathrm{c}}^{\mathrm{t}}$ are the centroids of supporting points of Π_i^{r} and Π_i^{t}, respectively. $d(*,*)$ represents the distance between two geometric elements.

Once the matched planes are obtained, the transform candidate is evaluated using a score metric

$$S_{\mathrm{match}} = m + w_p{*}\frac{1}{\sum_{i=1}^{m}(\Delta D_i + \Delta\theta_i)} \tag{10}$$

where m is the number of matched plane pairs. In the second term, w_p is a weighting factor, and ΔD_i represents the distance between the i-th pair of matched planes, while $\Delta\theta_i$ denotes the angle between their normal vectors. It is worth noting that ΔD_i and $\Delta\theta_i$ are very small values, thus we add them up without considering their physical units. Besides, w_p is set to a very small value (0.0001 in our implementation) such that the second term is always less than 1. This way, m dominates the value of S_{match}, while the second term will play a decisive role only when two matrices have the same number of matched planes. Following the computation of scores for all transform candidates, we filter out those with the lowest 80 % S_{match} values, and the remaining transformation matrices are delivered to the next step to determine the optimal transformation.

3.3. *Transform optimization and optimal transform determination*

In this step, we further optimize the candidate transforms with confidence values, ultimately selecting an optimal transform among all the refined candidates. As the normal directions of planes are invariant to changes in scaling and translation, the optimization of transformation parameters is achieved by optimizing the rotation matrix $\widehat{\boldsymbol{R}}'$ followed by the scale factor $\widehat{s}'$ and translation vector $\widehat{\boldsymbol{t}}'$.

Given n pairs of corresponding planes that conform to the same transform, we solve the following optimization problem to refine the rotation matrix $\widehat{\boldsymbol{R}}'$,

$$\widehat{\boldsymbol{R}}' = \underset{\boldsymbol{R}}{\operatorname{argmin}} \sum_{i=1}^{n} \|\boldsymbol{n}_{\Pi_i^{\mathrm{r}}} - \boldsymbol{R}\boldsymbol{n}_{\Pi_i^{\mathrm{t}}}\|^2 \tag{11}$$

where $\boldsymbol{n}_{\Pi_i^*}$ denotes the normal vector of plane Π_i^*. $\widehat{\boldsymbol{R}}'$ can be solved with the SVD method by constructing a covariance matrix similar to Eq. (7).

Given a point $\boldsymbol{p}_i^{\mathrm{r}}$ lying on a plane $\Pi^{\mathrm{r}} : [\boldsymbol{n}_{\Pi^{\mathrm{r}}}, d_{\Pi^{\mathrm{r}}}]$, where $d_{\Pi^{\mathrm{r}}}$ denotes the distance between the origin of the coordinate system and the plane, and $\boldsymbol{n}_{\Pi^{\mathrm{r}}}$ and $d_{\Pi^{\mathrm{r}}}$ satisfy $(\boldsymbol{n}_{\Pi^{\mathrm{r}}})^{\mathrm{T}}\boldsymbol{p}_i^{\mathrm{r}} = d_{\Pi^{\mathrm{r}}}$. Based on Eq. (6) and Eq. (8), we can derive the following equation,

$$\left(\boldsymbol{n}_{\Pi_i^{\mathrm{r}}}\right)^{\mathrm{T}}\widehat{\boldsymbol{t}}' = d_{\Pi_i^{\mathrm{r}}} - \widehat{s}' d_{\Pi_i^{\mathrm{t}}} \tag{12}$$

Thus, the scaling factor and the translation vector can be jointly refined by solving the following optimization problem:

$$\widehat{s}', \widehat{\boldsymbol{t}}' = \underset{s,\boldsymbol{t}}{\operatorname{argmin}} \sum_{i=1}^{n} \|\left(\boldsymbol{n}_{\Pi_i^{\mathrm{r}}}\right)^{\mathrm{T}}\widehat{\boldsymbol{t}}' - d_{\Pi_i^{\mathrm{r}}} + \widehat{s}' d_{\Pi_i^{\mathrm{t}}}\|^2 \tag{13}$$

where n is the number of matched planes.

After parameter optimization, the resulting transformation matrix is expressed as:

$$\boldsymbol{T} = \begin{bmatrix} \widehat{s}'\widehat{\boldsymbol{R}}' & \widehat{\boldsymbol{t}}' \\ 0^{\mathrm{T}} & 1 \end{bmatrix} \tag{14}$$

Since the transformation matrix only considers the corresponding relationship of planes, it cannot be guaranteed to be globally optimal. Therefore, in the step of accurate verification, we use a confidence score that not only considers plane correspondences but also point correspondences. The confidence score S'_{match} is calculated by the following:

Table 1
Statistics on the test datasets.

Datasets	Scene name / Overlap ratio	Acquisition technique/device	Points (Million)	Valid plane num.	Line num.	HGS num.
Campus	Campus Pair1	BLK360	15.69	14	52	1750
	(~30 %)	MVS	8.37	19	119	4969
	Campus Pair2	BLK360	19.76	22	159	14,652
	(~30 %)	MVS	19.22	25	193	11,720
	Campus Pair3	BLK360	31.96	22	144	10,932
	(~40 %)	MVS	7.60	17	102	3114
	Campus Pair4	AVIA SLAM	3.71	33	409	121,484
	(~50 %)	MVS	7.97	25	240	21,422
	Campus Pair5	AVIA SLAM	2.66	26	261	50,408
	(>60 %)	MVS	5.80	33	407	65,700
	Campus Pair6	AVIA SLAM	2.65	20	113	6464
	(>60 %)	MVS	6.57	21	151	6671
DTU	DTU set016	Structured light	3.50	22	169	20,992
	(>80 %)	MVS	2.55	19	137	6625
	DTU set025	Structured light	3.47	19	134	10,756
	(>70 %)	MVS	2.68	15	86	1493
Barn	Barn	LiDAR	12.74	27	241	40,458
	(>70 %)	MVS	10.62	18	118	4581

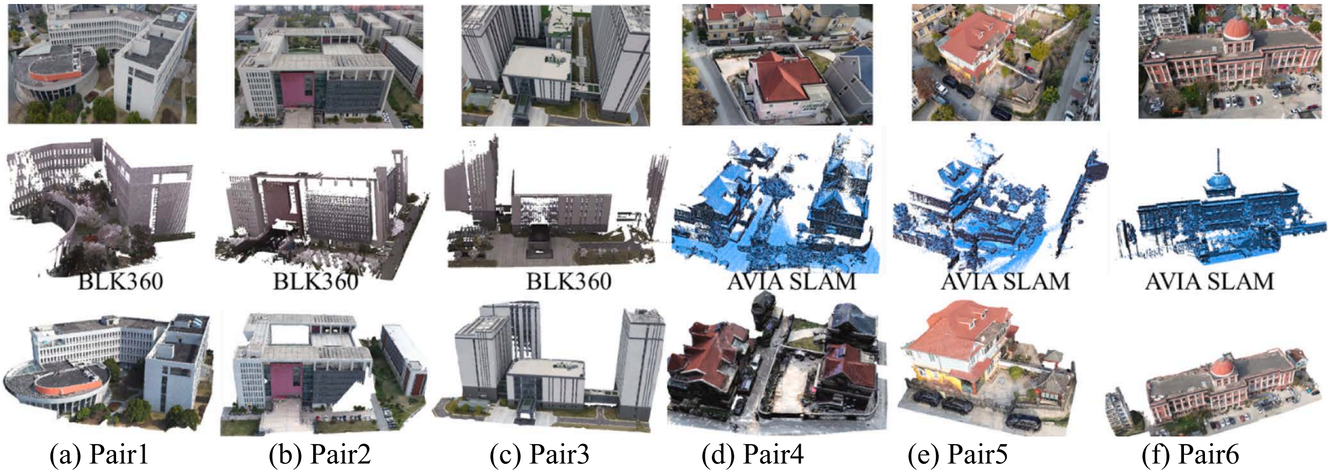


Fig. 7. Point clouds from our Campus dataset. Top: images. Middle: LiDAR point clouds (acquired by BLK360 and DJI AVIA-based Fast-lio2 SLAM algorithm (Xu et al., 2022)). Bottom: photogrammetric point clouds.

$$S'_{\text{match}} = w_{\text{planes}} \bullet R_{\text{planes}} + w_{\text{points}} \bullet R_{\text{points}} \tag{15}$$

where R_{planes} and R_{points} denote the ratios of matched planes and matched points, respectively, i.e.,

$$R_{\text{planes}} = \frac{S_{\text{planes}}}{\text{total plane number}}, \tag{16}$$

$$R_{\text{points}} = \frac{number\ of\ matched\ points}{\text{total point number}}, \tag{17}$$

and w_{planes} and w_{points} are the weights for matched planes and points, respectively. S_{planes} has the same form as S_{match} in Eq. (10).

We calculate the S'_{match} of each optimized transformation matrix and select the transformation with the highest score as the final coarse transformation. After the coarse registration, the Scale-ICP algorithm (Zinßer et al., 2005) is used to further refine the registration.

4. Experiments

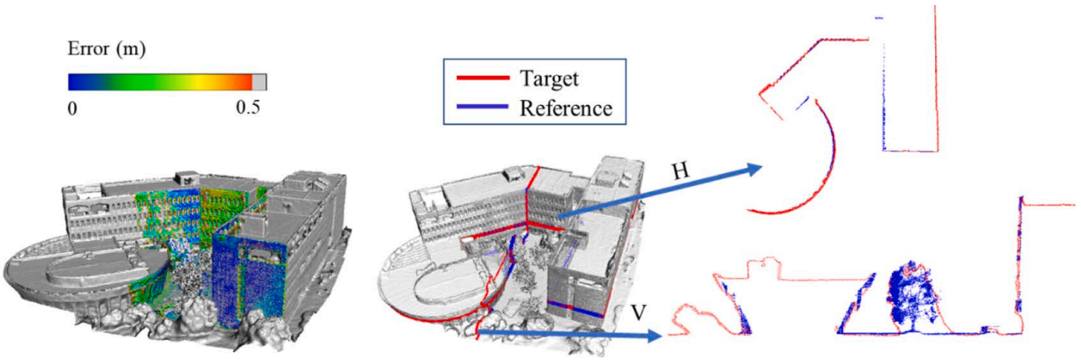
4.1. Experiment setup

The proposed method is implemented in C++ based on the PCL library (Rusu and Cousins, 2011). The optimization problem given in Eq. (13) is solved by the Gauss-Newton solver provided in the Ceres library (Agarwal and Mierle, 2022). All experiments are carried out on a laptop with an Intel Core i5-11400 h CPU and 16 GB RAM.

We tested our approach on 3 datasets, including our own dataset “Campus” as well as two public datasets, namely DTU (Aanæs et al., 2016) and Barn (Knapitsch et al., 2017). Table 1 provides an overview of these datasets. Each scene consists of a pair of point clouds obtained from different scanning methods. The overlapping ratio is estimated based on the coverage area of the reference data over the target data.

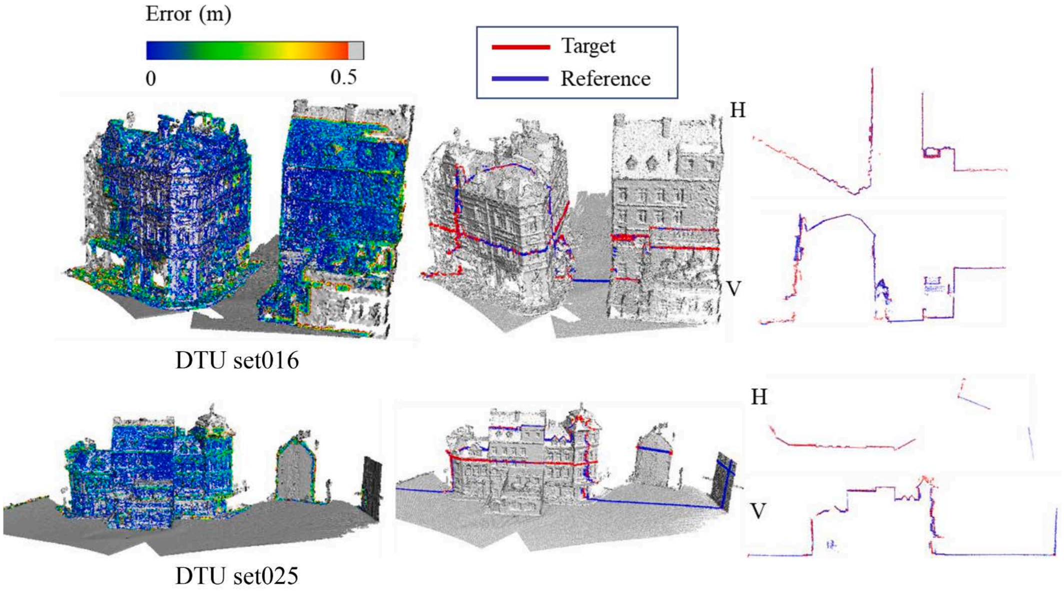
a) **Campus dataset:** To collect the LiDAR point clouds, we used two distinct types of commercial scanners, a static scanner Leica BLK360 and a mobile DJI AVIA sensor with the Fast-lio2 SLAM algorithm (Xu et al., 2022). The photogrammetric point clouds are generated from aerial images by the Autodesk ReCap software (Autodesk software ReCap Pro, 2022). The aerial images are captured by a digital camera mounted on an unmanned aerial vehicle (UAV) from an altitude of about 50 ~ 120 m. The ground truth registration parameters were obtained by manual registration. Fig. 7 demonstrates the point clouds from the Campus dataset.

b) **DTU dataset** (Aanæs et al., 2016)**:** Our experiments used the point clouds of two scenes from this dataset: DTU set016 and DTU set025. The point clouds of these scenes were reconstructed by structured light techniques and photogrammetric point clouds generated from



(a) Distance errors (b) Snapshots of horizontal/vertical cross-sections

Fig. 8. Registration results of Campus-Pair1.



(a) Distance errors (b) Snapshots of horizontal/vertical cross-sections

Fig. 9. Registration results of two scenes from the DTU dataset.

the DTU Robot Image datasets (Aanæs et al., 2016). Notably, the original data of set016 and set025 exhibit precise registration. To evaluate the performance of the registration algorithm, we deliberately applied random scale, rotation, and translation transformations to the original photogrammetric point clouds.

c) **Barn dataset** (Knapitsch et al., 2017)**:** The Barn dataset has a scene captured by a LiDAR point cloud and a photogrammetric point cloud.

In this work, we evaluate the registration results based on two commonly used metrics as follows.

- **Root Mean Square Error (RMSE)**. RMSE is a measure of the average distance between two structures. We use the mathematical notations $(\widehat{s}' \quad \widehat{\boldsymbol{R}}' \quad \widehat{\boldsymbol{t}}')$ that are consistent with Eq. (14) to present the transformation parameters, i.e.,

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N} \| (\widehat{s}'\widehat{\boldsymbol{R}}'\mathbf{p}_i + \widehat{t}') - (s^{\mathrm{truth}}\boldsymbol{R}^{\mathrm{truth}}\mathbf{p}_i + \boldsymbol{t}^{\mathrm{truth}}) \|_2^2} \tag{18}$$

where N is the number of points in the target point cloud. $\boldsymbol{R}^{\mathrm{truth}}$, $\boldsymbol{t}^{\mathrm{truth}}$, and s^{truth} are the ground-truth values of the parameters.

- **Parameter accuracy**. It measures the similarity between the calculated transformation parameters and the ground-truth parameters. The parameter accuracy includes errors e_s, $e_{\boldsymbol{R}}$, and e_t in scale, rotation, and translation, i.e.,

$$\begin{aligned} e_s &= \left| \widehat{s}' - s^{\mathrm{truth}} \right| \\ e_{\boldsymbol{R}} &= \arccos\left(\frac{\mathrm{tr}\left(\boldsymbol{R}^{\mathrm{truth}}\left(\widehat{\boldsymbol{R}}'\right)^{\mathrm{T}}\right) - 1}{2} \right) \\ e_t &= \| \widehat{\boldsymbol{t}}' - \boldsymbol{t}^{\mathrm{truth}} \|_2 \end{aligned} \tag{19}$$

where $\mathrm{tr}(\bullet)$ represents the trace of a matrix. $|\bullet|$ denotes the absolute value, and $\|\bullet\|_2$ is the L_2 norm.

4.2. Experimental results

We show the registration results of Campus-Pair1 in Fig. 8. This scene has many irregularly shaped plants on the ground, and non-planar points occupy a large proportion of the scene. To intuitively reveal the registration quality, we superimpose the colour-coded point clouds on the grayscale models, as shown in Fig. 8 (a), where the colour coding represents the error distribution. The efficacy of our matching method is further demonstrated by several cross-sections of the registered point clouds in Fig. 8 (b), where the red dots and blue dots correspond to the target points and the reference points, respectively. Similarly, the registration results on the DTU dataset are shown in Fig. 9. These results

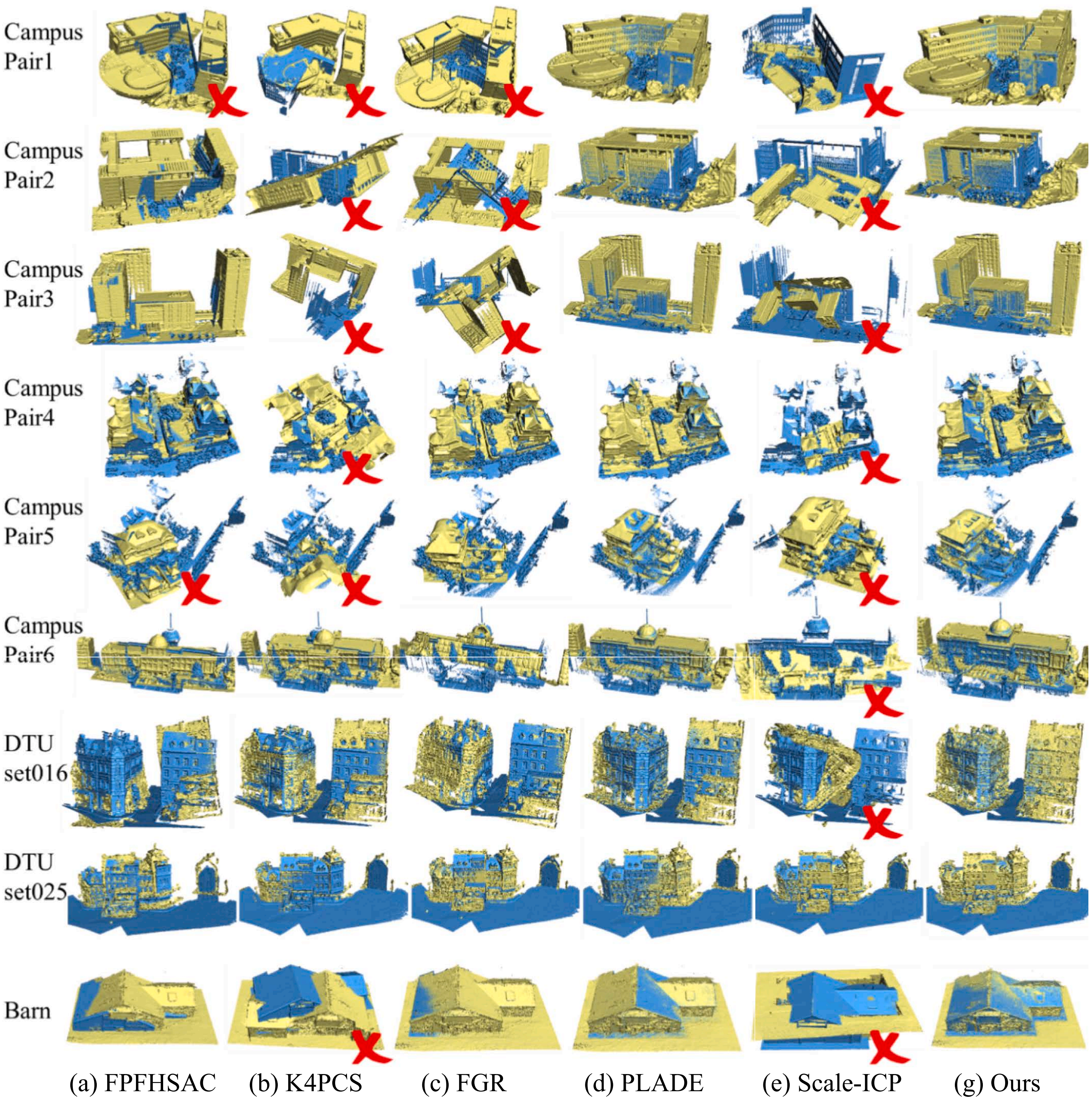


Fig. 10. Visual comparison between the coarse registration results from different methods: FPFHSAC (Holz et al., 2015), K4PCS (Theiler et al., 2014), FGR (Zhou et al., 2016), PLADE (Chen et al., 2019), and Scale-ICP (Ying et al., 2009). The red crosses indicate that the corresponding coarse registration has led to a failed ICP-based fine registration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

demonstrate the robustness of our proposed approach across diverse scenarios.

4.3. Comparisons

We compared the proposed approach with a few widely used methods, including FPFHSAC (Holz et al., 2015), K4PCS (Theiler et al., 2014), FGR (Zhou et al., 2016), PLADE (Chen et al., 2019), and Scale-ICP (Ying et al., 2009). Among these methods, only the Scale-ICP and our method can recover the unknown scales in addition to other registration parameters. To compare with other methods that assume known scales, we provide the competing methods with the ground-truth scale parameters. The results of the comparative visualization without ICP refinement are shown in Fig. 10, and the statistics are reported in Table 2.

In Fig. 10, we demonstrate the visual comparison of the coarse registration results from different methods. A coarse registration is considered successful if it has led to convergence in the subsequent ICP-based fine registration step. We have observed that even with the ground-truth scale parameters, methods like FPFHSAC (Holz et al., 2015), K4PCS (Theiler et al., 2014), and FGR (Zhou et al., 2016) only succeeded when the input point clouds exhibited high overlap. The K4PCS method relies on feature points to construct the 4-point congruent sets, hence it tends to miss sufficient supporting

Table 2
Statistics on the comparative results on different datasets using different methods: FPFHSAC (Holz et al., 2015), K4PCS (Theiler et al., 2014), FGR (Zhou et al., 2016), PLADE (Chen et al., 2019), and Scale-ICP (Ying et al., 2009).

Datasets	Overlap ratio	Metrics	FPFHSAC	K4PCS	FGR	PLADE	Scale-ICP	Ours
Campus	Campus Pair1	e_s	Manual	Manual	Manual	Manual	4.36	**0.02**
	(∼30 %)	$e_R(^\circ)$	24.88	177.20	58.70	**0.28**	71.32	0.87
		$e_t(m)$	12.38	117.70	51.89	**0.17**	49.01	0.56
		RMSE(m)	15.20	66.40	33.92	**0.22**	34.18	0.58
	Campus Pair2	e_s	Manual	Manual	Manual	Manual	4.62	**0.01**
	(∼30 %)	$e_R(^\circ)$	17.66	164.70	153.20	0.43	170.10	**0.40**
		$e_t(m)$	56.84	185.20	83.36	0.65	83.24	**0.23**
		RMSE(m)	33.32	99.60	108.10	**0.33**	87.14	0.76
	Campus Pair3	e_s	Manual	Manual	Manual	Manual	4.40	**0.01**
	(∼40 %)	$e_R(^\circ)$	7.82	173.50	175.80	0.60	175.00	**0.10**
		$e_t(m)$	8.04	160.10	170.10	0.79	63.13	**0.22**
		RMSE(m)	6.38	61.17	64.67	0.43	63.22	**0.17**
	Campus Pair4	e_s	Manual	Manual	Manual	Manual	1.41	**0.01**
	(∼50 %)	$e_R(^\circ)$	6.51	123.90	3.26	1.69	166.25	**0.70**
		$e_t(m)$	6.87	24.01	2.93	0.94	23.63	**0.42**
		RMSE(m)	4.11	29.09	1.81	0.49	26.99	**0.24**
	Campus Pair5	e_s	Manual	Manual	Manual	Manual	1.63	**0.06**
	(>60 %)	$e_R(^\circ)$	34.33	150.10	23.86	**1.68**	149.20	1.90
		$e_t(m)$	13.37	26.52	7.97	**0.64**	49.70	0.93
		RMSE(m)	8.75	20.48	4.62	**0.27**	24.69	0.41
	Campus Pair6	e_s	Manual	Manual	Manual	Manual	1.37	**0.07**
	(>60 %)	$e_R(^\circ)$	9.94	10.28	178.10	**0.60**	175.30	1.29
		$e_t(m)$	7.03	14.94	91.50	**0.45**	93.23	0.93
		RMSE(m)	13.55	8.83	63.45	**0.61**	56.89	1.04
DTU	DTU set016	e_s	Manual	Manual	Manual	Manual	0.97	**0.07**
	(>80 %)	$e_R(^\circ)$	11.98	3.84	6.18	0.28	111.80	**0.21**
		$e_t(m)$	16.72	4.48	6.52	**0.37**	44.81	0.48
		RMSE(m)	4.27	0.95	1.66	0.21	15.20	**0.15**
	DTU set025	e_s	Manual	Manual	Manual	Manual	**0.02**	**0.02**
	(>70 %)	$e_R(^\circ)$	5.13	1.69	2.80	1.16	1.98	**0.41**
		$e_t(m)$	7.16	1.57	3.34	1.39	1.52	**0.43**
		RMSE(m)	2.00	1.14	0.99	0.36	0.96	**0.20**
Barn	Barn	e_s	Manual	Manual	Manual	Manual	2.08	**0.01**
	(>70 %)	$e_R(^\circ)$	5.17	168.70	1.41	**0.18**	176.80	**0.18**
		$e_t(m)$	1.94	4.45	0.33	0.11	2.54	**0.01**
		RMSE(m)	1.96	16.24	0.36	0.11	13.30	**0.03**

Table 3
Statistics on the running times (in seconds) on different experiment cases using different methods including FPFHSAC (Holz et al., 2015), K4PCS (Theiler et al., 2014), FGR (Zhou et al., 2016), PLADE (Chen et al., 2019), and Scale-ICP (Ying et al., 2009).

	Campus Pair1	Campus Pair2	Campus Pair3	Campus Pair4	Campus Pair5	Campus Pair6	DTU set016	DTU set025	Barn
FPFHSAC	49.8	339.6	94.7	28.9	38.2	144.7	55.1	115.0	139.3
K4PCS	41.8	28.7	120.9	52.0	71.9	118.3	129.4	117.1	33.3
FGR	73.2	65.7	42.0	36.1	63.5	39.9	39.8	99.9	31.2
PLADE	13.6	23.8	17.5	11.6	12.1	13.4	22.6	9.2	11.4
Scale-ICP	334.8	871.1	476.3	179.7	125.9	106.3	194.2	346.1	63.6
Ours	5.8	8.6	11.6	8.8	9.2	5.2	7.7	6.8	7.3

correspondences when there are not enough extracted feature points due to the limited overlap between the two datasets. The FGR method failed when the relative orientations of the two input point clouds were far apart from each other. As for the Scale-ICP method (Ying et al., 2009), it converged to successful coarse registrations only in cases where substantial overlap exists. It is worth noting that although both PLADE (Chen et al., 2019) and our method have succeeded in the registration of all the point cloud pairs, our method outperforms the PLADE method in most cases based on evaluation metrics (as reported in Table 2). Our method has two advantages. First, our method has an important measure to estimate the scale factor of the cross-source point clouds, while PLADE assumes given scale parameters (or it requires an additional process to restore the scale parameter). Second, it reduces the elements in the feature vectors to reduce the correlation of the feature variables, thus it has better efficiency, as analysed in the following section.

4.4. Efficiency analysis

Table 3 presents the runtime analysis of our proposed approach along with several competing methods. Notably, FPFHSAC (Holz et al., 2015), FGR (Zhou et al., 2016), and Scale-ICP (Ying et al., 2009) operate based on pointwise correspondences. These methods become computationally expensive when the number of points increases. In contrast, K4PCS (Theiler et al., 2014), PLADE (Chen et al., 2019), and our method employ the structure querying strategies and thus demonstrate a clear advantage in terms of efficiency. As shown in the last column in Table 1, the number of HGSs is much smaller than that of points. Although the computational complexity of the feature matching is $O(n^2)$, the number of querying items is significantly reduced using the constructed HGSs, resulting in improved computational efficiency.

To evaluate the robustness of the proposed method against small overlap ratios, we tested our method on the Barn dataset that provides 9 small LiDAR scans of a scene, as shown in Fig. 11. In this test, the

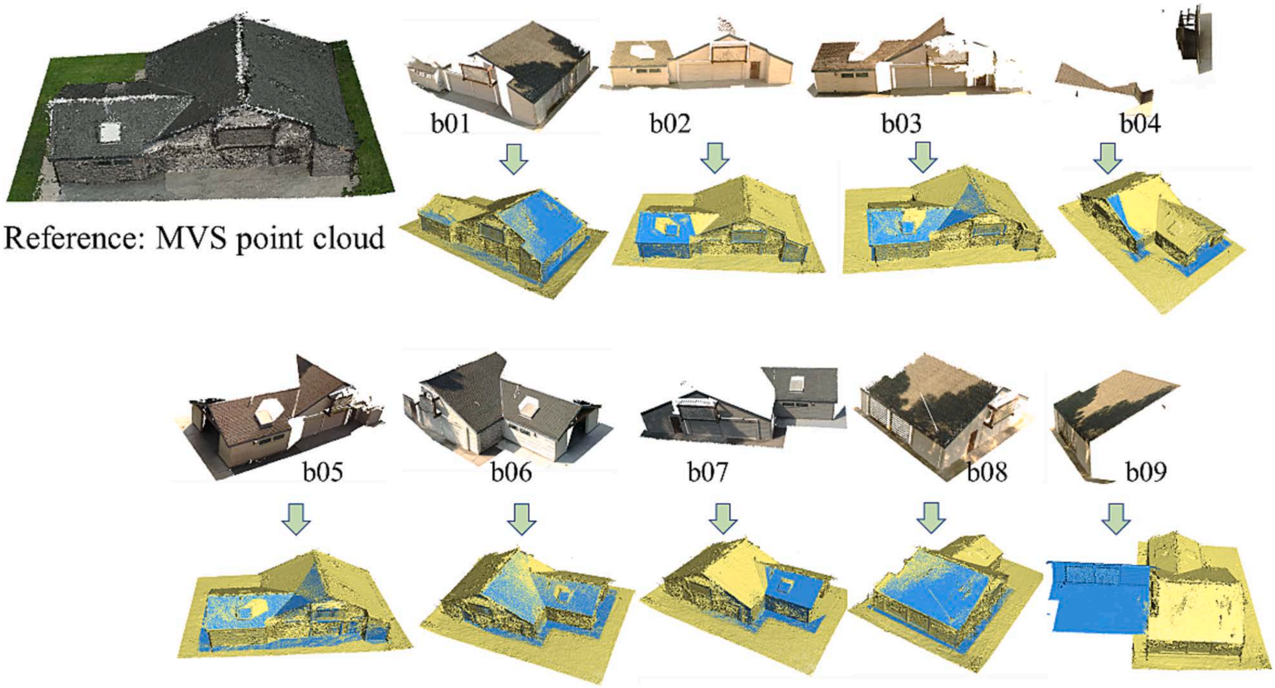


Fig. 11. The evaluation of our method in terms of robustness against limited overlapping ratios. This test was carried out on the Barn dataset (Knapitsch et al., 2017) that provides 9 small LiDAR scans of the scene with varying overlapping ratios to the reference MVS point cloud, ranging from 20% to 50%.

Table 4
Statistics on the number of matched planes and the corresponding registration errors.

	b01	**b02**	**b03**	**b04**	**b05**	**b06**	**b07**	**b08**	**Min**	**Max**	**Average**
Plane num.	8	6	6	6	9	7	6	5	5	9	6.7
E_s	0.03	0.14	0.19	0.02	0.01	0.01	0.03	0.03	0.01	0.19	0.06
$E_R(^\circ)$	0.37	0.43	0.14	0.25	0.22	0.13	0.89	0.22	0.13	0.89	0.33
$E_t(m)$	0.10	0.22	0.26	0.04	0.03	0.02	0.05	0.11	0.02	0.26	0.10
RMSE(m)	0.12	0.35	0.48	0.07	0.04	0.03	0.14	0.12	0.03	0.48	0.17

reference data is the MVS point cloud, and the target data are the 9 independent laser scans. The overlap ratios between the target data and the reference data range from 20 % to 50 %. This test has revealed that the proposed method can generally achieve effective registration with an overlap greater than 40 %, and it can even succeed in some cases with a low overlap of up to 25 %. It is worth noting that the registration of the b09 data failed since only 3 pairs of corresponding planes were captured. Such a limitation arose due to that our proposed method requires sufficient pairs of corresponding planes to generate a reasonably large number of HGSs. In Table 4, we provide the number of matched planes after the application of optimal transformations, along with the registration errors while matching b01-b08 with the MVS point cloud. From both the visual and quantitative results, we can conclude that our method has good robustness and efficacy in the registration of cross-source point clouds of challenging man-made structures.

Our approach, nevertheless, has limitations despite the advantages demonstrated previously. The method assumes that a scene is dominated by piecewise planar structures. This assumption prohibits its applicability to scenes that primarily consist of curved surfaces.

5. Conclusions

Combining LiDAR scanning and photogrammetric reconstruction technologies can provide complete and accurate 3D representations of complex objects or scenes. The complementary nature of these techniques allows for improved scene coverage, geometric details, and visual appearance in the final reconstructed models. The proposed approach in this work focuses on the registration of LiDAR scans and photogrammetric point clouds. We introduce a structural geometric set called HGS, which captures structural-level features of scenes, enabling automatic registration without prior knowledge of the scale parameters and improving robustness against varying point densities. Our work has shown that using high-level structural information facilitates the computation of correspondence queries, leading to higher accuracy and efficiency in the matching stage. This strategy also enables the recovery of transformation parameters when the point clouds exhibit different overlap ratios. Extensive experiments on a set of challenging scenes have validated the effectiveness and robustness of the proposed approach for the global coarse registration of LiDAR scans and photogrammetric point clouds. In future work, we plan to incorporate independently detected local structural patterns to better handle scenarios with limited overlap.

CRediT authorship contribution statement

Minglei Li: Writing – original draft, Visualization, Resources, Project administration, Methodology, Conceptualization. **Shu Peng:** Visualization, Software, Methodology. **Liangliang Nan:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

The authors would like to thank all data providers for sharing their data. This work was supported in part by the National Natural Science Foundation of China under Grant No. 42271343.

References

Aanæs, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B., 2016. Large-scale data for multiple-view stereopsis. Int. J. Comput. Vision 120, 153–168. https://doi.org/10.1007/s11263-016-0902-9.

Agarwal, S., Mierle, K., and The Ceres Solver Team, 2022, https://github.com/ceres-solver/ceres-solver.

Aiger, D., Mitra, N.J., Cohen-Or, D., 2008. 4-points congruent sets for robust pairwise surface registration. ACM Trans. Graphics 27 (3), 1–10. https://doi.org/10.1145/1399504.1360684.

Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S., 2019. PointNetLK: robust & efficient point cloud registration using PointNet. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7163–7172. https://doi.org/10.1109/CVPR.2019.00733.

Autodesk ReCap Pro. 2022. https://www.autodesk.co.uk/products/recap.

Bae, K.H., Lichti, D.D., 2008. A method for automated registration of unorganised point clouds. ISPRS J. Photogramm. Remote Sens. 63 (1), 36–54. https://doi.org/10.1016/j.isprsjprs.2007.05.012.

Baltsavias, E.P., 1999. A comparison between photogrammetry and laser scanning. ISPRS J. Photogramm. Remote Sens. 54 (2–3), 83–94. https://doi.org/10.1016/S0924-2716(99)00014-3.

Brenner, C., 2005. Building reconstruction from images and laser scanning. Int. J. Appl. Earth Obs. Geoinf. 6 (3–4), 187–198. https://doi.org/10.1016/j.jag.2004.10.006.

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. https://doi.org/10.1109/TGRS.2019.2952086.

Dong, Z., Liang, F., Yang, B., Xu, Y., Zang, Y., Li, J., Wang, Y., Dai, W., Fan, H., Hyyppäb, J., Stilla, U., 2020. Registration of large-scale terrestrial laser scanner point clouds: a review and benchmark. ISPRS J. Photogramm. Remote Sens. 163, 327–342. https://doi.org/10.1016/j.isprsjprs.2020.03.013.

El-Hakim, S., Gonzo, L., Voltolini, F., Girardi, S., Rizzi, A., Remondino, F., Whiting, E., 2007. Detailed 3D modelling of castles. Int. J. Archit. Comput. 5 (2), 199–220. https://doi.org/10.1260/1478-0771.5.2.200.

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. https://doi.org/10.1145/358669.358692.

Furukawa, Y., Hernández, C., 2015. Multi-view stereo: a tutorial. Found. Trends Comput. Graph. vis. 9 (1–2), 1–148. https://doi.org/10.1561/0600000052.

Ghamisi, P., Rasti, B., Yokoya, N., Wang, Q., Hofle, B., Bruzzone, L., Bovolo, F., Chi, M., Anders, K., Gloaguen, R., Atkinson, P.M., Benediktsson, J.A., 2019. Multisource and multitemporal data fusion in remote sensing: a comprehensive review of the state of the art. IEEE Geosci. Remote Sens. Mag. 7 (1), 6–39. https://doi.org/10.1109/MGRS.2018.2890023.

Holz, D., Ichim, A.E., Tombari, F., Rusu, R.B., Behnke, S., 2015. Registration with the point cloud library: a modular framework for aligning in 3D. IEEE Robot. Autom. Mag. 22 (4), 110–124. https://doi.org/10.1109/MRA.2015.2432331.

Huang, R., Xu, Y., Yao, W., Hoegner, L., Stilla, U., 2021. Robust global registration of point clouds by closed-form solution in the frequency domain. ISPRS J. Photogram. Rem. Sens. 171, 310–329. https://doi.org/10.1016/j.isprsjprs.2020.11.014.

Huang, X., Zhang, J., Fan, L., Wu, Q., Yuan, C., 2017. A systematic approach for cross-source point cloud registration by preserving macro and micro structures. IEEE Trans. Image Process. 26 (7), 3261–3276. https://doi.org/10.1109/TIP.2017.2695888.

Huang, X., Mei, G., Zhang, J., 2020. Feature-metric registration: a fast semi-supervised approach for robust point cloud registration without correspondences. 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V., 2017. Tanks and temples: benchmarking large-scale scene reconstruction. ACM Trans. Graphics 36 (4), 1–13. https://doi.org/10.1145/3072959.3073599.

Li, M., Lafarge, F., 2021. Planar Shape Based Registration for Multi-modal Geometry. In: BMVC 2021.

Li, M., Rottensteiner, F., Heipke, C., 2019. Modelling of buildings from aerial LiDAR point clouds using TINs and label maps. ISPRS J. Photogramm. Remote Sens. 154, 127–138. https://doi.org/10.1016/j.isprsjprs.2019.06.003.

Liu, W., Zang, Y., Xiong, Z., Bian, X., Wen, C., Lu, X., Wang, C., Junior, J.M., Gonçalves, W.N., Li, J., 2023. 3D building model generation from MLS point cloud and 3D mesh using multi-source data fusion. Int. J. Appl. Earth Obs. Geoinf. 116, 103171 https://doi.org/10.1016/j.jag.2022.103171.

Mellado, N., Aiger, D., Mitra, N.J., 2014. Super 4PCS fast global pointcloud registration via smart indexing. Comput. Graph. Forum 33 (5), 205–215. https://doi.org/10.1111/cgf.12446.

Mellado, N., Dellepiane, M., Scopigno, R., 2015. Relative scale estimation and 3D registration of multi-modal geometry using growing least squares. IEEE Trans. vis. Comput. Graph. 22 (9), 2160–2173. https://doi.org/10.1109/TVCG.2015.2505287.

Mura, M.D., Prasad, S., Pacifici, F., Gamba, P., Chanussot, J., Benediktsson, J.A., 2015. Challenges and opportunities of multimodality and data fusion in remote sensing. Proc. IEEE 22nd European Signal Processing Conference 103 (9), 1585–1601. https://doi.org/10.1109/JPROC.2015.2462751.

Parmehr, E.G., Fraser, C.S., Zhang, C., 2016. Automatic parameter selection for intensity-based registration of imagery to LiDAR data. IEEE Trans. Geosci. Remote Sens. 54 (12), 7032–7043. https://doi.org/10.1109/TGRS.2016.2594294.

Peng, F., Wu, Q., Fan, L., Zhang, J., You, Y., Lu, J., Yang, J.Y., 2014. Street view cross-sourced point cloud matching and registration. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 2026–2030. https://doi.org/10.1109/ICIP.2014.7025406.

Rister, B., Horowitz, M.A., Rubin, D.L., 2017. Volumetric image registration from invariant keypoints. IEEE Trans. Image Process. 26 (10), 4900–4910. https://doi.org/10.1109/TIP.2017.2722689.

Rusinkiewicz, S., 2019. A symmetric objective function for ICP. ACM Trans. Graphics 38 (4), 1–7. https://doi.org/10.1145/3306346.3323037.

Rusu, R. B., Cousins, S., 2011. 3D is here: Point cloud library (PCL), In: Proc. ICRA, May 2011, pp. 1-4. doi:10.1109/ICRA.2011.5980567.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE Int. Conf. Robot. Autom., pp. 3212-3217. doi:10.1109/ROBOT.2009.5152473.

Saiti, E., Theoharis, T., 2020. An application independent review of multimodal 3d registration methods. Comput. Graph. 91, 153–178. https://doi.org/10.1016/j.cag.2020.07.012.

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for point-cloud shape detection. Comput. Graph. Forum 26 (2), 214–226. https://doi.org/10.1111/j.1467-8659.2007.01016.x.

Thapa, A., Pu, S., Gerke, M., 2009. Semantic feature based registration of terrestrial point clouds. Proc. Int. Soc. Photogramm. Remote Sens. 230–235.

Theiler, P.W., Wegner, J.D., Schindler, K., 2014. Keypoint-based 4-points congruent sets-automated marker-less registration of laser scans. ISPRS J. Photogram. Rem. Sens. 96, 149–163. https://doi.org/10.1016/j.isprsjprs.2014.06.015.

Tombari, F., Salti, S., Di Stefano, L., 2010. Unique shape context for 3D data description. In: Proc. ACM Work. 3D Object Retr, pp. 57–62. https://doi.org/10.1145/1877808.1877821.

Wang, H., Liu, Y., Hu, Q., Wang, B., Chen, J., Dong, Z., Guo, Y., Wang, W., Yang, B., 2023. RoReg: pairwise point cloud registration with oriented descriptors and local rotations. IEEE TPAM I, 2023. https://doi.org/10.1109/TPAMI.2023.3244951.

Wang, Y., Solomon, J., 2019. Deep closest point: learning representations for point cloud registration. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3523–3532. https://doi.org/10.1109/ICCV.2019.00362.

Wang, Y., Yang, B., Chen, Y., Liang, F., Dong, Z., 2021. JoKDNet: a joint keypoint detection and description network for large-scale outdoor TLS point clouds registration. Int. J. Appl. Earth Obs. Geoinf. 104, 102534 https://doi.org/10.1016/j.jag.2021.102534.

Xu, Y., Boerner, R., Yao, W., Hoegner, L., Stilla, U., 2019. Pairwise coarse registration of point clouds in urban scenes using voxel-based 4-planes congruent sets. ISPRS J. Photogram. Rem. Sens. 151, 106–123. https://doi.org/10.1016/j.isprsjprs.2019.02.015.

Xu, W., Cai, Y., He, D., Lin, J., Zhang, F., 2022. Fast-lio2: fast direct lidar-inertial odometry. IEEE Trans. Robot. 38 (4), 2053–2073. https://doi.org/10.1109/TRO.2022.3141876.

Yan, F., Nan, L., Wonka, P., 2016. Block assembly for global registration of building scans. ACM Trans. Graphics 35 (6), 1–11. https://doi.org/10.1145/2980179.2980241.

Yang, B., Dong, Z., Liang, F., Liu, Y., 2016. Automatic registration of large-scale urban scene point clouds based on semantic feature points. ISPRS J. Photogram. Rem. Sens. 113, 43–58. https://doi.org/10.1016/j.isprsjprs.2015.12.005.

Ying, S., Peng, J., Du, S., Qiao, H., 2009. A scale stretch method based on ICP for 3D data registration. IEEE Trans. Autom. Sci. Eng. 6 (3), 559–565. https://doi.org/10.1109/TASE.2009.2021337.

Zhang, J., Yao, Y., Deng, B., 2022. Fast and robust iterative closest point. IEEE Trans. Pattern Anal. Mach. Intell. 44 (7), 3450–3466. https://doi.org/10.1109/TPAMI.2021.3054619.

Zhou, Q., Park, P., Koltun, V., 2016. Fast global registration. In: 2016 European Conference on Computer Vision (ECCV). https://doi.org/10.1007/978-3-319-46475-6_47.

Zinßer, T., Schmidt, J., Niemann, H., 2005. Point set registration with integrated scale estimation. In: International Conference on Pattern Recognition and Image Processing (PRIP 2005), pp. 116-119.