Automatic Registration of Point Clouds by Combining Local Shape Descriptor and G4PCS Algorithm

Wuyong Tao, Jingbin Liu, Dong Xu, and Yanyang Xiao

***Abstract*—Registration is usually the first step for the usage of point cloud data. Registration of point clouds is crucial in numerous practical applications. In order to quickly and accurately align point clouds, we combine the local shape descriptor and generalized four-point congruent set (G4PCS) algorithm. A descriptor with high computation efficiency is chosen to establish the correspondences. Then, the constraints of the G4PCS algorithm are applied and three angular constraints are proposed. These constraints are divided into three groups according to their computation complexity. The three groups of constraints are used to accelerate the searching process of four correct correspondences. The experiments are performed to compare our method with several existing methods. The experimental results show that our registration method has the best registration precision and computation efficiency. The introduction of the local shape descriptor can enhance both registration precision and computation efficiency. The proposed three angular constraints can also further boost the search process.**

***Index Terms*—Constraints, four-point congruent set (4PCS), local shape descriptor (LSD), point cloud registration.**

I. Introduction

In MANY applications, such as scene reconstruction [1], [2], object recognition [3], [4], and place recognition [5], [6], point cloud registration plays a central role. In order to obtain the entire point cloud, the point clouds scanned from different stations should be transformed into a common coordinate system. To do this, an efficient and automatic registration method is crucial. Point cloud registration contains two steps: coarse and fine registration. The fine registration is usually performed by the iterative closest point (ICP) algorithm [7] and its variations, but these algorithms need a good initial pose to avoid getting trapped into locally minimum solution. Our work focuses on the coarse registration, aiming at providing a good initial pose to the fine registration.

At present, a large amount of registration methods have been proposed. These methods can be broadly categorized into three classes: point-based [8], [9], line/plane-based [10], [11], [12], and special object-based methods [13], [14]. The line/plane-based methods extract line or plane features from the point clouds. Then, these line or plane features are used to estimate the transformation parameters. Therefore, these methods require that the scanned scenes have a plenty of line or plane features, which limit the usage of these methods. The special object-based methods employ special objects (e.g., octagonal lamp pole) to perform the alignment of point clouds. Due to the special objects, this kind of methods is only suitable for the scenes with the special objects. The point-based methods are universal and suitable for different scenes. The local shape descriptor (LSD) based methods [9] and the four-point congruent set (4PCS) family [15] are the most popular point-based methods. The former first calculates the LSDs of the keypoints. Then, the point-to-point correspondences are established by the descriptor similarity. The latter utilizes the rule of intersection ratios to find 4-point base set. Both of the two kinds of methods have their own advantages.

The LSD has been widely studied, and a large amount of LSDs are proposed. A good LSD should have high descriptiveness, strong robustness, and low computation complexity. According to whether a local reference frame (LRF) is applied or not, these descriptors are divided into two categories: LRF-based and LRF-independent descriptor. The LRF-based descriptor first constructs an LRF on the local surface around a keypoint. Then, the geometric and spatial information are encoded with respect to the LRF. The LRF has at least two merits. The first is to give the descriptor rotation invariance. The second is to provide a manner to fully encode the spatial information. Examples contain rotational projection statistics (RoPS) [16], binary shape context (BSC) [3], triplet local coordinate images [4], rotational contour signatures (RCS) [17], local voxelized structure (LoVS) [18], and so forth. The LRF-independent descriptor encodes the local neighborhood information with respect to the normal vector of the keypoint. The spatial information is ignored or not fully encoded. Therefore, this kind of descriptors generally suffers from low descriptiveness. The LRF-independent descriptors include spin images (SI) [19], fast point feature histograms (FPFH) [20], local feature statistics histograms (LFSH) [21], divisional local

Manuscript received 1 May 2023; revised 5 June 2023 and 22 June 2023; accepted 4 July 2023. Date of publication 7 July 2023; date of current version 19 July 2023. This work was supported in part by the Spatial Cognition Augmented High-Usability and High-Precision Smartphone Indoor Positioning under Grant 41874031 and in part by the National Natural Science Foundation of China under Grant 62102174. *(Corresponding author: Wuyong Tao.)*

Wuyong Tao and Yanyang Xiao are with the School of Mathematics and Computer Sciences, Nanchang University, Nanchang 430031, China (e-mail: wuyong_tao@whu.edu.cn; xiaoyanyang@ncu.edu.cn).

Jingbin Liu and Dong Xu are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China, and also with the Department of Remote Sensing and Photogrammetry, Finnish Geospatial Research Institute, 02430 Masala, Finland (e-mail: jingbin.liu@whu.edu.cn; dongxu_@whu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3293409

feature statistics (DLFS) [22], and so on. Both the LRF-based and LRF-independent descriptors have been successfully used for point cloud registration. Among these descriptors, the BSC and LoVS are two binary descriptors, which consume less memory. Particularly, the LoVS has very good computation efficiency. Also, its descriptor matching performance is pretty good [18].

The 4PCS algorithm is first proposed by Aiger et al. [15]. Given four approximately coplanar points, the algorithm uses the rule of intersection ratios to search all possible 4-point bases. Due to brute-force search, the algorithm is very time-consuming. In order to increase the computation efficiency, many improvements have been made to the 4PCS algorithm, such as super 4PCS [23], generalized 4PCS (G4PCS) [24], keypoint-based 4PCS (K-4PCS) [25], and semantic keypoint-based 4PCS (SK-4PCS) [26]. These algorithms enhance the precision and computation efficiency of the 4PCS algorithm from different aspects. Nevertheless, they are still computationally expensive, as will be shown in the experiments.

The 4PCS family searches all possible 4-point bases from the target point cloud for randomly selected four points from the source point cloud. The number of the possible 4-point bases from the target point cloud is huge. Such search is eyeless, and thus leads to much computation time for verifying all possible 4-point bases. For the reason, the LSD is first introduced to establish point-to-point correspondences. Thus, once four correspondences are selected, the corresponding 4-point base from the target point cloud is fixed and its number is 1. We do not need to verify so many 4-point bases. However, many of the correspondences established by the LSD are incorrect. The rules of the G4PCS algorithm can help to reject incorrect correspondences and estimate accurate transformation under such cases. Hence, the LSD and G4PCS algorithm are combined in this article. Although the scholars have combined the 4PCS algorithm with the keypoint detectors (e.g., K-4PCS and SK-4PCS algorithms), no one explores to take the LSD and G4PCS algorithm into combination. The existing methods either only use the LSD or only use the 4PCS rules. Here, we choose a computationally efficient descriptor to establish correspondences. Then, some improvements are made to the G4PCS algorithm. Three groups of constraints are developed to search correct *four-correspondence bases instead of 4-point bases*. As a consequence, a computationally cheap and precise registration method is proposed in this article. This means that the registration method aggregates the advantages of both the LSD and G4PCS algorithm.

II. Related Work

Our work mainly involves the LSD and 4PCS algorithm. Therefore, the LSD is first reviewed in this section. Then, the development process of the 4PCS algorithm is presented.

A. Local Shape Descriptor

The LRF-independent descriptors are first reviewed. The SI [19] is the most frequently mentioned descriptor. It calculates two parameters for each neighbor. Then, the two parameters are divided to form 2-D grids. The number of the neighbors in a grid is employed to compute the grid value. The THRIFT descriptor [27] computes a weighted histogram by using the normal deviations of the neighboring points and keypoint. Rusu et al. [28] used the normal deviations of any two points in the local surface to encode the geometric information. The point feature histograms (PFH) is presented, but the descriptor is time-consuming. In order to improve the computation efficiency, Rusu et al. [20] proposed the FPFH descriptor by using a simplified PFH. In [21], the LFSH descriptor is proposed. This descriptor assembles three attributes, i.e., normal deviation, signed distance, and point density, so as to encode more geometric information. The DLFS descriptor [22] utilizes a local height and three angle attributes to generate four histograms, which are concatenated to form the final histogram. These descriptors discard spatial information, so they generally suffer from low descriptiveness.

Regarding the LRF-based descriptor, both the spatial and geometric information are encoded. Snapshots [29] is an early proposed descriptor. The descriptor partitions the *xy* plane of the LRF into 2-D grids. The feature values of the grids are calculated according to the projection distances of the neighboring points. Because the local surface information is only encoded from one view, much information is lost. Tombari et al. [30] proposed the signature of histograms of orientations descriptor. An LRF building method was developed first. The method removed the sign ambiguity of the three axes. Thus, the calculated LRF is unique. In addition, the distances between the neighboring points and the keypoint were applied to calculate the weights of the neighboring points. This is useful to enhance the robustness of the LRF to occlusion and clutter. The local neighborhood was then divided into some subspaces with respect to the LRF. The normal deviations were applied to calculate a histogram for each subspace. Finally, all the histograms were integrated to obtain the feature vector. A drawback of the descriptor is that it is sensitive to point density variation. The unique shape context (USC) [31] is an improvement to the 3-D shape context [32] by inserting an unique LRF. This descriptor has rather high dimensionality, leading to high computation complexity. The triple SI (TriSI) [33] applies the multiple-view mechanism so as to encode more information. It calculates three SIs from three views. The RoPS [16] also uses the multiple-view mechanism to make it include more information, but the descriptor achieves multiple views by rotating the local patch. For each rotated point cloud, three distribution matrices are generated by projecting it on the three planes of the LRF. The statistics (four central moments and a Shannon entropy) of all the distribution matrices form the feature histogram. The TriSI and RoPS cannot be applied for raw point cloud data because the LRF building method of the two descriptors is designed for triangle mesh data. In addition, all of the USC, TriSI, and RoPS use point distribution information to calculate the feature vector. Consequently, they are sensitive to point density variation. Similar to the TriSI descriptor, the local depth images are calculated from three views in the triple orthogonal local depth images (TOLDI) descriptor [34]. The pixel values of the three images constitute the feature vector. Subsequently, Tao et al. [8] considered that some information

is still lost in the TOLDI descriptor due to self-occlusion. The quintuple local coordinate images descriptor was developed. It calculates five local coordinate images from five views, but its dimensionality is high. The RCS [17] also rotates the local point cloud to obtain multiple views. The 2-D contour information is applied to compute the feature values. However, the descriptor is susceptible to noise and point density variation. Unlike the other descriptors, the signature of geometric centroids (SGC) [35] and LoVS [18] employs a bounding cubical volume instead of a spherical volume to enclose the local point cloud. The bounding cubical volume is then voxelized. For the SGC, the centroid and point number are applied to compute the voxel value. It is susceptible to occlusion and clutter. For the LoVS, if the grid is occupied by points, its value is 1 and 0, otherwise. It is a naturally binary descriptor without binarization. Ghorbani et al. [36] extended the 2-D DAISY descriptor [37] to tackle 3-D point cloud data. A 3-D local DAISY-style descriptor was proposed. A comprehensive evaluation about nine LSDs is presented in [38].

B. 4PCS Algorithm

The 4PCS algorithm [15] has been developed for a long time. Many efforts have been made to improve its performance. The main drawback of the algorithm is its high computation complexity. Therefore, most of the scholars work hard to enhance its computation efficiency. Mellado et al. [23] proposed the super 4PCS algorithm. The algorithm applied an efficient data structure to find all point pairs, making the algorithm run in optimal linear time. Mohamad et al. [24] removed the constraint that the points in a 4-point base should be coplanar. The G4PCS algorithm was developed. Because the number of nonplanar bases is significant smaller, the registration efficiency is improved. Subsequently, Mohamad et al. [39] combined the super 4PCS and G4PCS algorithms. The super G4PCS algorithm was proposed. The registration efficiency was further enhanced. Theiler et al. [25] proposed the K-4PCS algorithm, which first extracted keypoints from the point clouds by 3-D difference-of-Gaussian keypoint detector or 3-D Harris keypoint detector. These extracted keypoints were input into the 4PCS algorithm. Because the number of the keypoints is significantly less than the number of the points in a point cloud, the computation efficiency has a large improvement. Ge [26] also extracted keypoints first. The difference is that these keypoints are vertexes of lines, vertexes of arcs, intersection points of two lines, and centers of circles. Hence, they are called as semantic keypoint, and the algorithm is termed as SK-4PCS. These semantic keypoints are the input of the G4PCS algorithm. However, the algorithm is only suitable for some special scenes due to the usage of the semantic information. Li et al. [40] extracted the boundary points as the input of the super 4PCS algorithm. These algorithms still need to traverse all the keypoint pairs to search for congruent 4-point base set, so they are still pretty time-consuming.

III. Point Cloud Registration by Combining LSD and G4PCS Algorithm

Our registration method combines the LSD and G4PCS algorithm. First, the LSD is used to establish correspondences. Then, we make some improvements to the G4PCS algorithm. Three groups of constraints are developed to reject 4-correspondence bases that contain false correspondences. The detailed computation process is shown in Fig. 1.

Our method includes two stages. In the first stage, the keypoints are first extracted from the source and target point clouds. Here, we choose the 3-D Harris keypoint detector because it is easy to implement and has high computation efficiency, but the other detectors can also be applied. An example of the extracted keypoints is shown in Fig. 2. The corner points are detected as the keypoints.

Then, for each keypoint, an LSD is calculated. In this article, the LoVS descriptor is applied. The reasons for choosing the descriptor are as follows. 1) The descriptor has high computation efficiency. Thus, the computation burden that is caused by the calculation of the LSD will not increase so much. 2) The descriptor is highly descriptive and robust to many nuisances (include noise, point density variation, and partial overlap), as reported in [38]. Thus, more correct correspondences can be established. This makes that we have bigger possibility of selecting correct correspondences in the process of the random selection. 3) The descriptor is a binary one. This makes the algorithm consumes less memory. In addition, the descriptor similarity is calculated using the Hamming distance, decreasing the time for descriptor matching. Finally, the correspondences are established by the nearest neighbor similarity rate (NNSR) [16]. In NNSR, for a descriptor from source keypoint, two closest descriptors from target keypoints are searched and two Hamming distances are obtained. If the ratio between the two Hamming distances is smaller than a threshold, the source keypoint associated with the descriptor and the target keypoint associated with its closest descriptor are treated as a correspondence. In the second stage, four correspondences (i.e., *a 4-correspondence base*) are randomly selected. Three groups of constraints are applied to judge whether the four correspondences contain incorrect ones or not. If the four correspondences satisfy the three groups of constraints, they are used to estimate the rigid-body transformation. The details of the LoVS descriptor and three groups of constraints will be described in Sections III-A and III-B, respectively.

A. LoVS Descriptor

The LoVS descriptor is different from the other binary ones, such as BSC and RCS. Most of the binary descriptors are obtained by binarizing the real-valued descriptors, while the LoVS is a naturally binary descriptor. The LoVS descriptor first builds an LRF on the local neighborhood around the keypoint by the method proposed in [34]. The local neighborhood is cropped by a sphere with radius r around a keypoint. A local cubic volume is used to enclose the local neighborhood and is then divided into uniform subspaces, i.e., voxelization.

As shown in Fig. 3, for each voxel, we need to judge whether it contains some neighboring points or not. The neighboring points are the points in the local neighborhood. If the voxel is occupied by some neighboring points, its value is 1 and 0, otherwise

$$l(v_i) = \begin{cases} 1, & \text{if } \left|\boldsymbol{Q}_{v_i}\right| > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

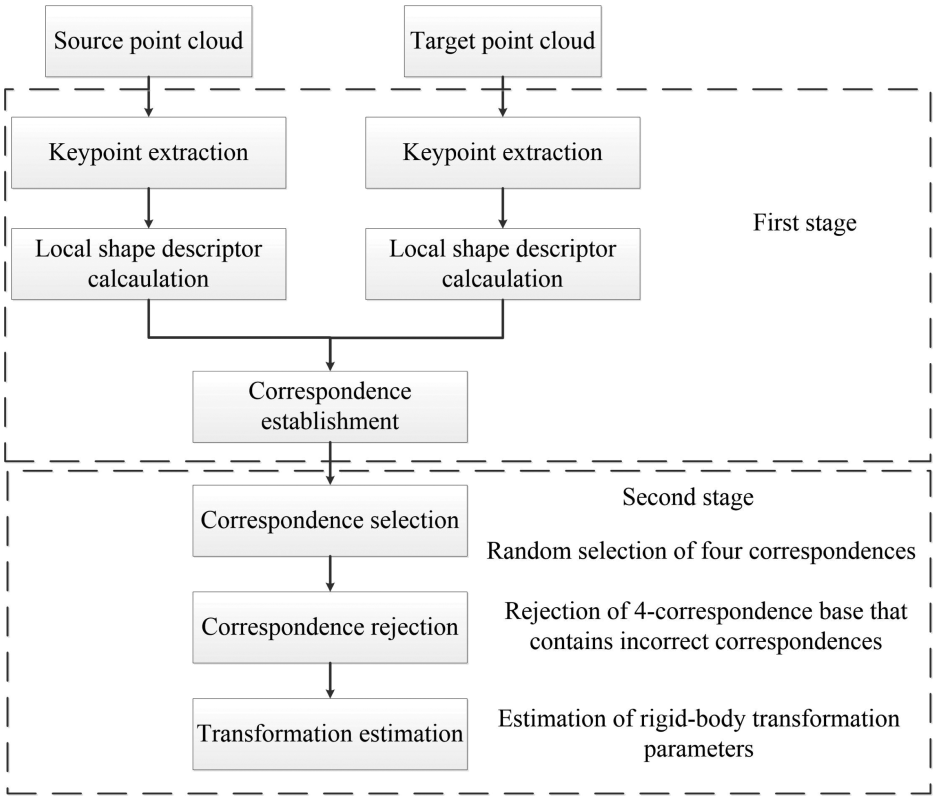


Fig. 1. Computation process of the proposed registration method.

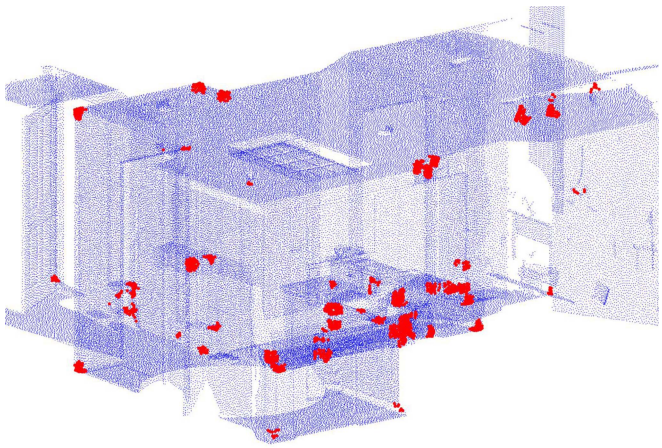


Fig. 2. Keypoints extracted by 3-D Harris keypoint detector. The red points are the keypoints.

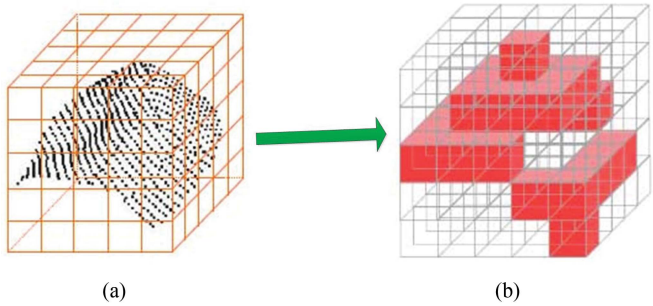


Fig. 3. Voxel value calculation. The values of the red voxels are 1 and the values of the others are 0. (a) Voxelization of local neighborhood. (b) Values of voxels.

where $\boldsymbol{Q}_{v_i}$ is the point set in a voxel v_i, and $|\cdot|$ denotes the cardinality of a point set. All the voxel values are concatenated to form the final feature vector, i.e., the LoVS descriptor

$$\boldsymbol{f}_{\text{LoVS}} = \begin{bmatrix} l(v_1) & l(v_2) & \cdots & l(v_{N_v}) \end{bmatrix} \tag{2}$$

where $N_v = m^3$ is the number of the voxels, and m is the number of the voxels along each edge of the local cubic volume.

Two parameters of the descriptor are m and r (r is the local neighborhood radius). By experiments, r is set as $15pr$ (pr denotes the point cloud resolution, i.e., the average value of the distances between all points and their nearest neighboring points) and m is set as 9. Therefore, the number of voxels is $9 \times 9 \times 9 = 729$, which means that the dimensionality of the descriptor is 729 and its storage is 729 bit. For more details, readers can refer to [18]. Although the LoVS descriptor has high descriptiveness, it is impossible to ensure that all the correspondences established by the descriptor are correct. Therefore, we need to find out the correct correspondences in the second stage.

B. Correspondence Rejection Based on Three Groups of Constraints

The 4PCS algorithm is proposed for the registration of partially overlapping point clouds. Its goal is to find corresponding

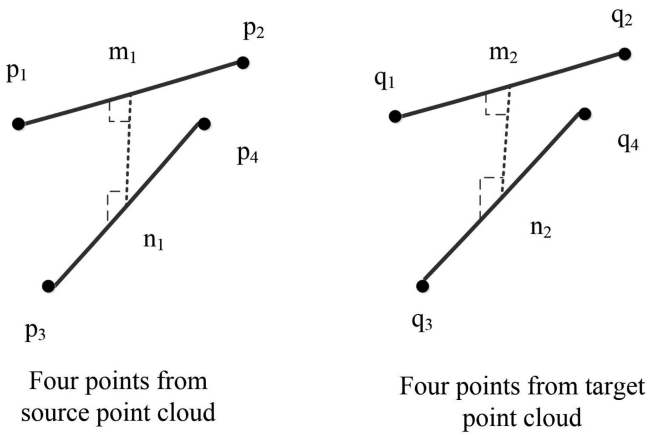


Fig. 4. Rule of intersection ratios. The dash line is the connecting line between two intersection points of two spatial lines.

four points to compute the transformation by the rule of intersection ratios. Mohamad et al. [24] proposed the G4PCS algorithm, which is an extension to the 4PCS algorithm. The randomly selected four points are no longer necessary to be coplanar.

Fig. 4 shows the rule of intersection ratios. $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4)$ is a 4-point base from source point cloud and $(\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3, \boldsymbol{q}_4)$ is a 4-point base from target point cloud. $\boldsymbol{m}_1$ and $\boldsymbol{n}_1$ are the intersection points of the lines $\boldsymbol{p}_1\boldsymbol{p}_2$ and $\boldsymbol{p}_3\boldsymbol{p}_4$ in the 3-D space. That is the distance between $\boldsymbol{m}_1$ and $\boldsymbol{n}_1$ is equal to the shortest distance between $\boldsymbol{p}_1\boldsymbol{p}_2$ and $\boldsymbol{p}_3\boldsymbol{p}_4$. $\boldsymbol{m}_1\boldsymbol{n}_1$ is orthogonal to both $\boldsymbol{p}_1\boldsymbol{p}_2$ and $\boldsymbol{p}_3\boldsymbol{p}_4$. Similarly, $\boldsymbol{m}_2$ and $\boldsymbol{n}_2$ are the intersection points of the lines $\boldsymbol{q}_1\boldsymbol{q}_2$ and $\boldsymbol{q}_3\boldsymbol{q}_4$ in the 3-D space. Then, the four intersection ratios can be calculated as

$$\lambda_1 = \frac{||\boldsymbol{p}_1 - \boldsymbol{m}_1||}{||\boldsymbol{p}_1 - \boldsymbol{p}_2||}, \quad \lambda_2 = \frac{||\boldsymbol{p}_3 - \boldsymbol{n}_1||}{||\boldsymbol{p}_3 - \boldsymbol{p}_4||}$$
$$\lambda_3 = \frac{||\boldsymbol{q}_1 - \boldsymbol{m}_2||}{||\boldsymbol{q}_1 - \boldsymbol{q}_2||}, \quad \lambda_4 = \frac{||\boldsymbol{q}_3 - \boldsymbol{n}_2||}{||\boldsymbol{q}_3 - \boldsymbol{q}_4||} \tag{3}$$

where $||\cdot||$ denotes the Euclidean distance. These intersection ratios are rotation invariant. Therefore, the following constraints are generated:

$$\text{abs}(\lambda_1 - \lambda_3) < \varepsilon_1, \quad \text{abs}\,(\lambda_2 - \lambda_4) < \varepsilon_1 \tag{4}$$

where $\text{abs}(\cdot)$ denotes the absolute value. Under the case of rigid transformation, the additional constraints can be added to boost the searching process. Aiger et al. [15] proposed two constraints based on the length of the diagonals in a 4-point base. Subsequently, Theiler et al. [25] developed four constraints based on the side length of a 4-point base. The six constraints are as follows:

$$\begin{aligned}
&\text{abs}(\|\boldsymbol{p}_1 - \boldsymbol{p}_2\| - \|\boldsymbol{q}_1 - \boldsymbol{q}_2\|) < \varepsilon_2 \\
&\text{abs}(\|\boldsymbol{p}_3 - \boldsymbol{p}_4\| - \|\boldsymbol{q}_3 - \boldsymbol{q}_4\|) < \varepsilon_2 \\
&\text{abs}(\|\boldsymbol{p}_1 - \boldsymbol{p}_3\| - \|\boldsymbol{q}_1 - \boldsymbol{q}_3\|) < \varepsilon_3 \\
&\text{abs}(\|\boldsymbol{p}_1 - \boldsymbol{p}_4\| - \|\boldsymbol{q}_1 - \boldsymbol{q}_4\|) < \varepsilon_3 \\
&\text{abs}(\|\boldsymbol{p}_2 - \boldsymbol{p}_3\| - \|\boldsymbol{q}_2 - \boldsymbol{q}_3\|) < \varepsilon_3 \\
&\text{abs}(\|\boldsymbol{p}_2 - \boldsymbol{p}_4\| - \|\boldsymbol{q}_2 - \boldsymbol{q}_4\|) < \varepsilon_3.
\end{aligned} \tag{5}$$

In addition, the distance between two lines in a 4-point base can also be used to formulate a constraint as follows:

$$\text{abs}(\|\boldsymbol{m}_1 - \boldsymbol{n}_1\| - \|\boldsymbol{m}_2 - \boldsymbol{n}_2\|) < \varepsilon_3. \tag{6}$$

In our method, the correspondences [i.e., $(\boldsymbol{p}_i, \boldsymbol{q}_i)$] have been established by the LoVS descriptor, i.e., the corresponding relationship between the source keypoints and target keypoints has been established, so $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4)$ are four source keypoints and $(\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3, \boldsymbol{q}_4)$ are their corresponding four target keypoints. Hence, $(\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3, \boldsymbol{q}_4)$ are not obtained by brute-force search. When four correspondences are selected, the four source keypoints and their corresponding four target keypoints are selected.

In order to further accelerate the searching process, three angular constraints are developed in this article. We know that the rotation between a source keypoint and a target keypoint is equal to that between source and target point clouds. The LRFs of two keypoints can be used to calculate the rotation between them. We can compute four angle differences according to the LRFs of the four correspondences

$$\begin{aligned}
\Delta\theta_1 &= a\cos\left(\frac{\text{trace}\left(\boldsymbol{V}_{p_1}\boldsymbol{V}_{q_1}^{-1}\right) - 1}{2}\right)\frac{180}{\pi} \\
\Delta\theta_2 &= a\cos\left(\frac{\text{trace}\left(\boldsymbol{V}_{p_2}\boldsymbol{V}_{q_2}^{-1}\right) - 1}{2}\right)\frac{180}{\pi} \\
\Delta\theta_3 &= a\cos\left(\frac{\text{trace}\left(\boldsymbol{V}_{p_3}\boldsymbol{V}_{q_3}^{-1}\right) - 1}{2}\right)\frac{180}{\pi} \\
\Delta\theta_4 &= a\cos\left(\frac{\text{trace}\left(\boldsymbol{V}_{p_4}\boldsymbol{V}_{q_4}^{-1}\right) - 1}{2}\right)\frac{180}{\pi}
\end{aligned} \tag{7}$$

where $a\cos(\cdot)$ is the inverse cosine function, $\text{trace}(\cdot)$ is the trace of a matrix, $\boldsymbol{V}_{p_1}, \boldsymbol{V}_{p_2}, \boldsymbol{V}_{p_3}, \boldsymbol{V}_{p_4}$ are the LRFs of $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4$, and $\boldsymbol{V}_{q_1}, \boldsymbol{V}_{q_2}, \boldsymbol{V}_{q_3}, \boldsymbol{V}_{q_4}$ are the LRFs of $\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3, \boldsymbol{q}_4$. An LRF is a local coordinate system, which is represented by a 3×3 orthogonal matrix. The column vectors of the matrix are the x-axis, y-axis, and z-axis of the coordinate system. Noting that the LRFs of the keypoints have been calculated when computing the descriptors. Here, we just need to reuse them. If all of the four correspondences are correct, the rotations of the four pairs of keypoints are equal to each other and equal to the rotation between source and target point clouds. The rotations can be measured by the angle differences in (7). This means that the four angle differences are equal to each other. Thus, the three angular constraints are defined as

$$\begin{aligned}
&\text{abs}(\Delta\theta_1 - \Delta\theta_2) < \delta \\
&\text{abs}(\Delta\theta_2 - \Delta\theta_3) < \delta \\
&\text{abs}(\Delta\theta_3 - \Delta\theta_4) < \delta.
\end{aligned} \tag{8}$$

The three angular constraints can help reject more incorrect 4-correspondence bases, and thus, the searching process is accelerated.

According to the computation complexity of these constraints, three groups of constraints are developed. The first group of constraints is composed of those defined by (5). These constraints are easiest to calculate. The second group of constraints is composed of those defined by (4) and (6). The angular constraints constitute the third group of constraints, i.e., the constraints defined by (8).

The second stage of our registration algorithm is given in Algorithm 1. The correspondence set $\boldsymbol{C}$ is established by the LoVS descriptor. At each iteration, four correspondences are randomly selected. If the three groups of constraints are successively satisfied, the rigid transformation is calculated. Here, the singular value decomposition (SVD) [41] is applied to calculate the rigid transformation. The following matrices are constructed:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \\ \boldsymbol{p}_4 \end{bmatrix}, \quad \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \boldsymbol{q}_3 \\ \boldsymbol{q}_4 \end{bmatrix}. \tag{9}$$

Then, the covariance matrix is calculated as

$$\boldsymbol{H} = \left[\boldsymbol{A} - \mathbf{ones}(4,1)\bar{\boldsymbol{A}}\right]^T \left[\boldsymbol{Y} - \mathbf{ones}(4,1)\bar{\boldsymbol{Y}}\right] \tag{10}$$

where $\bar{\boldsymbol{A}}$ is the mean vector of $\boldsymbol{A}$, $\bar{\boldsymbol{Y}}$ is the mean vector of $\boldsymbol{Y}$, and $\mathrm{ones}(4,1)$ is a 4×1 vector whose elements are all 1. Performing SVD on matrix $\boldsymbol{H}$

$$\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \mathrm{svd}(\boldsymbol{H}). \tag{11}$$

Thus, the rotation matrix is

$$\boldsymbol{R} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^T \tag{12}$$

where $\boldsymbol{D} = \mathbf{diag}(\mathbf{1}, \mathbf{1}, \mathbf{det}(\boldsymbol{U}\boldsymbol{V}^T))$, $\mathrm{diag}(\cdot)$ denotes a diagonal matrix, and $\mathbf{det}(\cdot)$ denotes the determinant of a matrix. The translation vector is

$$\boldsymbol{t} = \bar{\boldsymbol{Y}} - \bar{\boldsymbol{A}}\boldsymbol{R}^T. \tag{13}$$

Once one group of constraints is not satisfied, next four correspondences are randomly selected. In our algorithm, the parameters $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are set as $3pr$, and the parameter δ is set as $10°$. The parameter setting makes our algorithm has both high registration precision and good computation efficiency. These parameters are determined by many trials. When computing the number of the inliers, only the downsampled source point cloud is applied. We perform uniform sampling on the source point cloud. The sampling interval is set as $7pr$. By this, the number of the points is largely reduced. The downsampled source point cloud is transformed according to the calculated transformation. If the distance between one transformed point and its nearest point in the target point cloud is smaller than $3pr$, the transformed point is considered as an inlier. The number of the iterations L should be set as a big value because most of the correspondences are incorrect. Also, we find that a small value of L will lead to low registration precision. In spite of this, the computation efficiency of our algorithm is very high, because the computation is not performed at most of the iterations due to the three groups of constraints.

Algorithm 1:

Input: The established correspondence set $\boldsymbol{C} = \{\boldsymbol{c}_1 \quad \boldsymbol{c}_2 \quad \cdots \quad \boldsymbol{c}_n\}$
Output: Best transformation
$\boldsymbol{c}_i = (\boldsymbol{p}_i, \boldsymbol{q}_i)$ is one correspondence
L is the number of the iterations
Set the initial number of inliers $\boldsymbol{n}_0$ as zero
For i = 1 to L **do**
  Randomly select four correspondences $\boldsymbol{c}_a, \boldsymbol{c}_b, \boldsymbol{c}_c, \boldsymbol{c}_d$.
  If the length of the diagonals and side length fulfill the first group of constraints.
    Compute the intersection points and intersection ratios.
    If the intersection points and intersection ratios fulfill the second group of constraints.
      Compute the four angle difference.
      If the angle difference fulfill the third group of constraints.
        Compute the rigid transformation using the four correspondences
        Compute the number of the inliers $\boldsymbol{n}$
        If $\boldsymbol{n} > \boldsymbol{n}_0$
          The current rigid transformation is the best one
          $\boldsymbol{n}_0 = \boldsymbol{n}$
        End if
      End if
    End if
  End if
End for

IV. Experiments and Analysis

In this section, the experiments are performed to evaluate our method. The K-4PCS algorithm, which is better than most of the other methods of the 4PCS family, is used to compare with our method. Specially, the K-4PCS algorithm is based on the 3-D Harris keypoint detector because our registration method also applies this keypoint detector. In addition, the multithreading strategy is not applied in the K-4PCS algorithm for a fair comparison. The difference between the two methods is that our method introduces the LSD and develops the three angular constraints. In order to illustrate the effect of the three angular constraints, we remove them from our method, and the method without angular constraints is denoted as "ours-wa." Another method used for comparison is "LoVS+1-point RANSAC." The method uses the LoVS descriptor to establish the correspondences. Then, the 1-point random sample consensus (RANSAC) algorithm [42] is applied to calculate the rigid transformation. We compare the four methods in terms of registration precision and computation efficiency. The registration precision is measured by the rotation error and translation error, which are, respectively, calculated as

$$\mathrm{error}_R = a\mathbf{cos}\left(\frac{\mathrm{trace}\left(\boldsymbol{R}_{\mathrm{true}}\boldsymbol{R}^{-1}\right)}{2}\right)\frac{180}{\pi} \tag{14}$$

$$\mathrm{error}_t = \|\boldsymbol{t}_{\mathrm{true}} - \boldsymbol{t}\| \tag{15}$$

TABLE I
INFORMATION OF THE POINT CLOUDS FROM THE FOUR SCENES

Scene	*pr* (m)	Degree of overlap	Numbers of extracted keypoints
Boardroom	0.0256	0.8076	1283 versus 1095
Apartment	0.0224	0.6941	1321 versus 1303
Lobby	0.0338	0.6582	2850 versus 3644
City	0.1231	0.4432	11953 versus 5214

where $\boldsymbol{R}_{\text{true}}$ is the true rotation, $\boldsymbol{R}$ is the estimated rotation, $\boldsymbol{t}_{\text{true}}$ is the true translation, and $\boldsymbol{t}$ is the estimated translation. The true transformation is obtained by manual coarse registration and the ICP [7] fine registration. The proposed registration method randomly selects four correspondences, which leads to different registration results in different implementation. Due to the randomness, we run the four methods for 50 times. The mean values of the rotation errors and translation errors are computed. As for the computation efficiency, the computation time spent on the two stages is respectively presented. The two stages of our method are as shown in Fig. 1. For the K-4PCS algorithm, the first stage is the process of extracting keypoints, and the second stage is the process of calculating the transformation. All the experiments are performed in MATLAB by a laptop with AMD Ryzen 9 5900HX processor and 32.0 GB RAM.

A. Experimental Data

An indoor dataset [43] and an outdoor dataset [44] are applied to perform the experiments. The indoor dataset is obtained by scanning five scenes using a FARO Focus 3D X330 HDR scanner. For each scene, multiple point clouds are obtained. We choose three pairs of point clouds from the three scenes (boardroom, lobby, and apartment). The outdoor dataset is the point clouds of the city scene, which is obtained by a Leica C10 laser scanner. A pair of point clouds is chosen from the scene. Actually, the LSD is more suitable for the registration of model point clouds because such point clouds have plenty of geometric information. By contrast, the point clouds of buildings are generally composed of planes and, therefore, lack geometric information. Hence, using LSD to align the point clouds of buildings is more challenging. Considering that the point density is too high, we perform simplification on the point clouds. The information of the point cloud data after simplification is summarized in Table I. The numbers of the keypoints extracted by 3-D Harris keypoint detector are also listed in Table I. The two values separated by "versus" represent the numbers of the source keypoints and target keypoints, respectively.

B. Performance on Indoor Dataset

The iteration number of our method and ours-wa is set as 30 000 for the boardroom and lobby scenes, and 3 000 000 for the apartment scene. For the apartment scene, the overlapping part is mainly composed of plane structures. Therefore, only a small number of keypoints are extracted from the overlapping part. This also leads to a small number of correct correspondences being established by the LSD. Hence, the iteration number is set as such a big value so that the good registration performance can be achieved. The iteration number of the K-4PCS is set as 300 and that of the LoVS+1-point RANSAC is set as 500 for all the three scenes. The registered point clouds obtained by all methods look similar when the registration is successful, so only the registration results of our method are shown in Fig. 5. In each graph, the raw point clouds are on the left and the registered point clouds are on the right. As we can see from Fig. 5, the two point clouds of each scene are well aligned together. The registration results are enough to provide good initial poses to the fine registration.

The rotation errors and translation errors for the 50 coarse registrations using the four methods are shown in Fig. 6. In each graph, the rotation errors are on the left and the translation errors are on the right. It can be seen that the K-4PCS algorithm has the poorest registration precision. The rotation errors and translation errors are sometimes extremely large. The second poorest method is LoVS+1-point RANSAC. The 1-point RANSAC uses the LRFs of the keypoints to calculate the rotation matrix, so only one correspondence is enough to calculate the rigid transformation. Compared to the RANSAC algorithm [45], the 1-point RANSAC needs much less iterations. However, due to the errors of the LRFs, the rotation matrix and translation vector calculated by the 1-point RANSAC also have relative big error. Our method and the ours-wa have similar registration precision. Our method uses the LSD to establish point-to-point correspondences. Once a correct correspondence is selected, the source keypoint will have an accurate corresponding target keypoint. Therefore, the registration can always get high precision. By the contrast, the corresponding 4-point base found by the K-4PCS algorithm in the target point cloud is approximate, so the registration precision is poorer. Due to the introduction of the LSD, the registration precision obtains a large improvement. Specially, on the apartment scene, the registration precision of the K-4PCS is very poor. This is because the keypoints in the overlapping part are very few, the K-4PCS algorithm has difficulty to find corresponding 4-point base from the target point cloud. Therefore, the method always gets large rotation errors and translation errors.

The computation time spent on the two stages for the four methods is shown in Fig. 7. We can see that the computation time of the K-4PCS algorithm is prohibitive. The time spent on the first stage is about 10 s, which is very small. However, the time spent on the second stage is sometimes as high as 20 000 s on the lobby scene. The lobby scene has more complicated structure, so more keypoints are extracted. Therefore, more 4-point bases are found. It leads to a large amount of time spent on the second stage. For our method, the time of the first stage is about 35 s on the boardroom and apartment scenes, and about 85 s on the lobby scene. For the lobby scene, more LSDs need to be calculated due to more keypoints. Therefore, more computation time is spent on the first stage compared to the boardroom and apartment scenes, but the increase of time is not huge. This is mainly because we choose an LSD with low computation complexity. The computation time of the second stage is smaller than 10 s on the boardroom and lobby scenes, and 60 s on the apartment scene. The iteration number of the apartment scene is larger, so more time is spent on the second stage. However, it is still affordable.

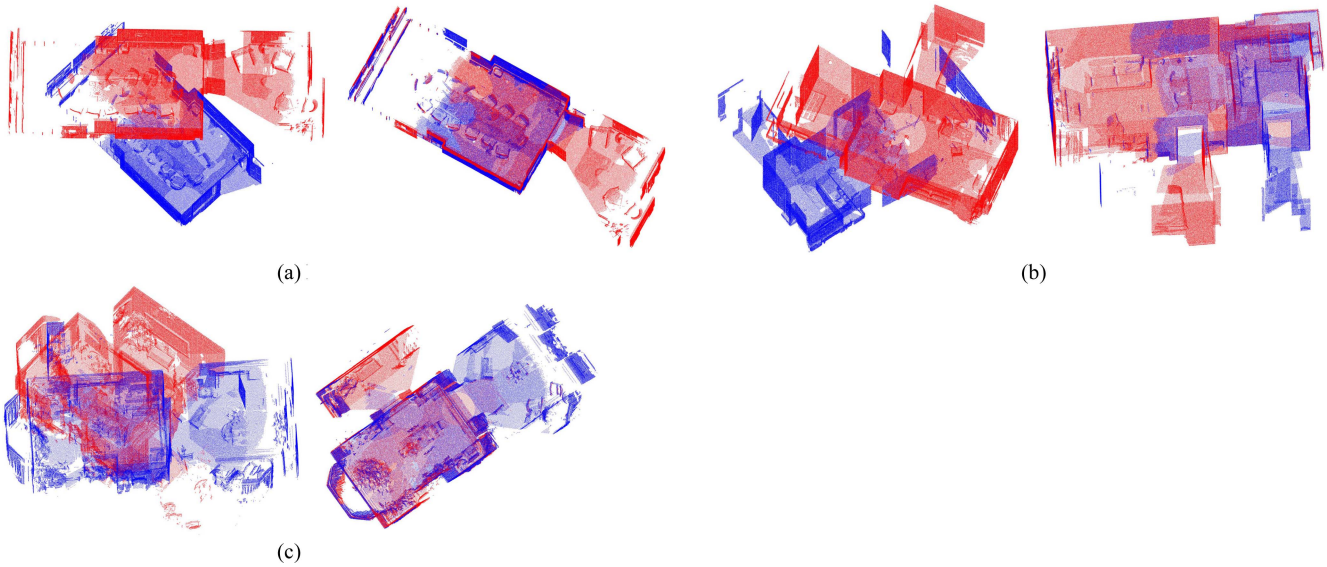


Fig. 5. Registration results on the three scenes. In each part, the initial point clouds are on the left and the registered point clouds are on the right. (a) Boardroom. (b) Apartment. (c) Lobby.

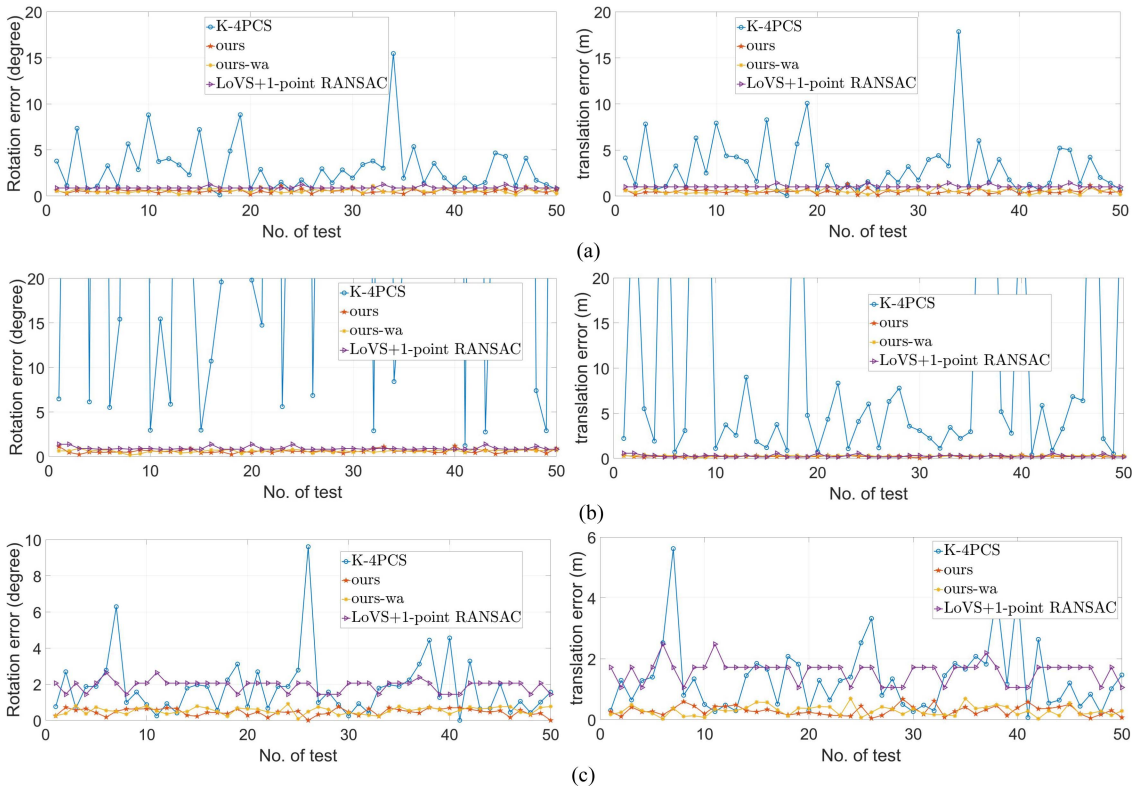


Fig. 6. Registration precision for 50 coarse registrations using the four methods on the three scenes. (a) Boardroom. (b) Apartment. (c) Lobby.

Although our method needs more iteration number, the total computation time is very small. This is because the computation of most iterations is not performed due to the three groups of constraints. However, in the K-4PCS algorithm, much time is spent on finding the 4PCS at each iteration. Our method costs more time on the first stage because of the introduction of the LSD, but the computation burden does not increase very much. This is because we choose a descriptor (i.e., LoVS), which is simple to compute, to establish the correspondences. The descriptor has not only the high computation efficiency but also the good matching performance. Although the first stage spends more time, the time spent on the second stage is rather small.

We can also see that the ours-wa obviously costs more time on the second stage compared to our method. This is because the

Fig. 7. Computation time for 50 coarse registrations using the four methods on the three scenes. (a) Boardroom. (b) Apartment. (c) Lobby.

ours-wa removes the three angular constraints. As we can see, the three angular constraints are useful to boost the search processing of the correct correspondences. The three angular constraints can help reject more incorrect 4-correspondence bases, so our method has better computation efficiency in comparison with the ours-wa. Particularly, when the required iteration number is very big, the effect of the three angular constraints is more obvious (e.g., the apartment scene). The LoVS+1-point RANSAC also spends more time on the second stage compared to our method. Although the method needs less iteration number, all the iterations are performed. Therefore, the method needs more time.

According to the calculated rotation errors and translation errors in Fig. 6, we calculate their mean values. The mean computation time of the two stages is also computed. All of them are listed in Table II.

TABLE II
REGISTRATION RESULTS FOR THE THREE SCENES

Method	$\overline{error}_R$ (°)	$\overline{error}_t$ (m)	$\bar{t}_1$ (sec)	$\bar{t}_2$ (sec)
Boardroom scene				
K-4PCS	3.0389	3.1995	10.0456	1283.0255
Our method	0.5222	0.5153	32.0233	4.7293
Ours-wa	0.5562	0.5513	31.7673	8.7998
LoVS+1-point RANSAC	0.9026	1.0472	31.7644	58.1043
Apartment scene				
K-4PCS	69.8638	11.4284	12.6543	1451.5358
Our method	0.5891	0.2063	33.3106	52.3795
Ours-wa	0.6104	0.2668	33.4846	94.2602
LoVS+1-point RANSAC	0.912	0.2574	33.7734	75.4875
Lobby scene				
K-4PCS	1.8233	1.3586	10.5137	12715.1328
Our method	0.467	0.2946	85.5806	7.8753
Ours-wa	0.5421	0.2891	88.9659	27.4557
LoVS+1-point RANSAC	1.9171	1.5725	84.8556	56.5856

$\overline{error}_R$ and $\overline{error}_t$ are the mean values of the rotation errors and translation errors. $\bar{t}_1$ and $\bar{t}_2$ are the mean computation time spent on the first stage and second stage.

From Table II, we can see that the mean values of the rotation errors and translation errors for our method are smaller than those of the K-4PCS algorithm by a large margin. The total mean computation time of our method is also obviously smaller. Therefore, the improvement of all aspects is significant. We can conclude that introducing the LSD can enhance both the registration precision and computation efficiency. By combining the LSD and G4PCS algorithm, we obtain a point cloud registration method with good registration precision and high computation efficiency. Admittedly, the increase of the iteration number can improve the registration precision for the two methods. However, the computation time will also increase. Especially for the K-4PCS algorithm, the increase of the computation time is very huge. This is the reason why we do not increase the iteration number of the K-4PCS algorithm on the apartment scene. In addition, extracting more keypoints can also improve the registration precision of the K-4PCS algorithm. On the lobby scene, because of more keypoints, the K-4PCS algorithm also gets relatively high registration precision. This is because more keypoints increase the possibility of finding more accurate 4-point base from the target point cloud.

The mean rotation errors and mean translation error of the ours-wa are bigger than those of our method, so the registration precision of the ours-wa is poorer. The mean computation time spent on the first stage is similar to our method, while the mean computation time spent on the second stage is more. This indicates that the proposed three angular constraints are effective. The LoVS+1-point RANSAC also has poorer registration precision and computation efficiency in comparison with our method. This method only uses one correspondence to calculate the rigid transformation, so the constraints, which are always the relationship between two correspondences, cannot be inserted into the method for skipping some iterations. Hence, the LoVS+1-point RANSAC costs much more time on the second stage.

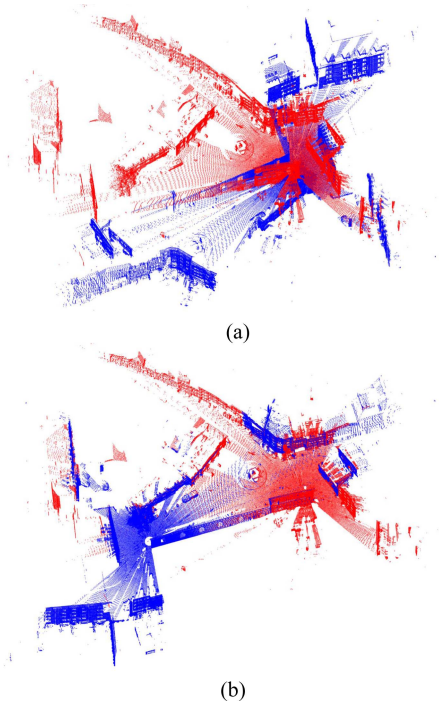


Fig. 8. Registration result of the city scene. (a) Initial point clouds. (b) Registered point clouds.

C. Performance on Outdoor Dataset

On the outdoor dataset, although more keypoints are extracted, the proportion of correct correspondences is small because of the low degree of overlap. Therefore, the iteration number of our method and ours-wa is set as 6 000 000. The iteration number of the K-4PCS and LoVS+1-point RANSAC is still respectively set as 300 and 500. The registered point clouds obtained by all methods look similar when the registration is successful, so only the registration result of our method is shown in Fig. 8. It can be seen from Fig. 8(b) that the two point clouds have been well registered.

The rotation errors and translation errors of the 50 tests are shown in Fig. 9. We can see that the K-4PCS algorithm gets big rotation errors and translation errors. Because the keypoints in the overlapped area are few due to the low degree of overlap, the K-4PCS algorithm is difficult to find accurate corresponding 4-point base. The LoVS+1-point RANSAC obtains the second poorest registration precision. In this pair of point clouds, the noise is serious so that the calculated LRFs have bigger errors. Therefore, the LoVS+1-point RANSAC gets relative big rotation errors and translation errors. Our method and the ours-wa have the best registration precision. Because the LSD is introduced to establish the correspondences, our method can always find accurate corresponding keypoints.

The comparison of the computation time is presented in Fig. 10. As we can see, the K-4PCS algorithm spends very few time (about 10 s) on the first stage but much time on the second stage. Sometimes, the total computation time is as high as 20 000 s. In comparison with the indoor dataset, our method spends more time on the first stage. This is because more

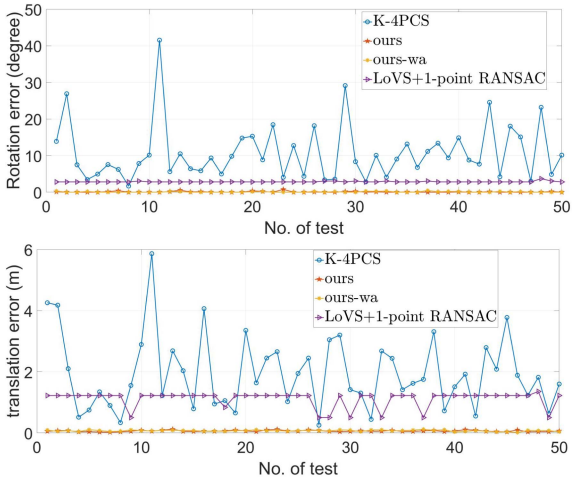


Fig. 9. Registration precision for 50 coarse registrations using the four methods on the city scene.

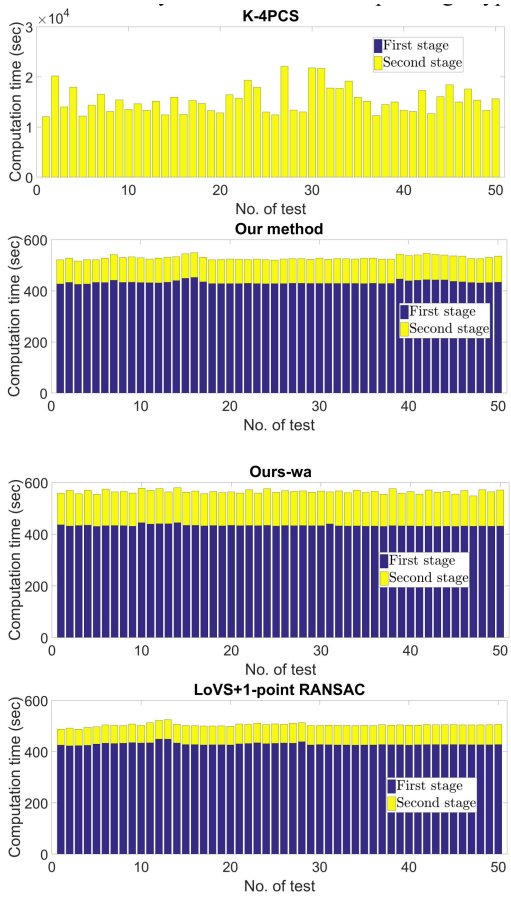


Fig. 10. Computation time for 50 coarse registrations using the four methods on the city scene.

keypoints are extracted from the point clouds. The time spent on the second stage is still very little. The total computation is significantly smaller than that of the K-4PCS. Due to the introduction of the LSD, our method does not need to traverse all the point pairs to find the corresponding target keypoints. The ours-wa also spends more time on the second stage compared to our method, which indicates that the proposed three angular

TABLE III
REGISTRATION RESULTS FOR THE CITY SCENE

Method	$\overline{error}_R$ (°)	$\overline{error}_t$ (m)	$\bar{t}_1$ (s)	$\bar{t}_2$ (s)
K-4PCS	10.7821	1.9416	14.831	15411.2077
Our method	0.0793	0.0649	432.1526	96.4455
Ours-wa	0.0758	0.0734	433.4224	131.3329
LoVS+1-point RANSAC	2.8433	1.1184	429.494	74.5976

$\overline{error}_R$ and $\overline{error}_t$ are the mean values of the rotation errors and translation errors. $\bar{t}_1$ and $\bar{t}_2$ are the mean computation time spent on the first stage and second stage.

constraints can further improve the searching process. More incorrect 4-correspondence bases are rejected by introducing the three angular constraints in our method. On the dataset, the LoVS+1-point RANSAC spends less total time. For any case, we can set a large value for the iteration number of our method because the computation time is still small. The iteration number does not affect the computation time very much in our method. From this point of view, our method is insensitive to iteration number. For the K-4PCS and 1-point RANSAC, if the iteration number increases, the computation time will also significantly increase.

According to the results in Fig. 9, the mean rotation error and mean translation error are calculated and listed in Table III, and the mean computation time of the two stages is also given. The K-4PCS algorithm has the largest mean rotation error and mean translation error, followed by the LoVS+1-point RANSAC. Our method and the ours-wa get similar mean rotation error and mean translation error. The K-4PCS algorithm spends a little time on the first stage. Due to the calculation of the LSD, the other three methods spend more time on the first stage. However, the computation time of the second stage of the other three methods is much less than that of the K-4PCS algorithm. Although our method requires a large amount of iterations, the computation time is rather few. At the aspects of the registration precision and computation efficiency, our method is obviously superior to the K-4PCS algorithm. Therefore, the introduction of the LSD is useful to improve both of the registration precision and computation efficiency.

V. CONCLUSION AND OUTLOOK

A registration method, which combined the LSD and G4PCS algorithm, had been presented in this article. The LoVS descriptor was selected to establish the correspondences. Then, three angular constraints were proposed. Both the constraints of the G4PCS algorithm and the three angular constraints were organized into three groups. Based on the three groups of constraints, Algorithm 1 was developed.

The experiments had been performed on the two datasets, including an indoor dataset and an outdoor dataset. The experimental results indicate that our method is able to get high registration precision and good computation efficiency. Our method introduces the LSD to establish the correspondences. Therefore, in the process of the random sampling, once the selected correspondences are correct, the good registration precision can be obtained because the corresponding keypoints are relatively accurate. However, the K-4PCS algorithm usually find

approximate 4-point base. In addition, also because of the LSD, our method does not need to traverse all the keypoint pairs for finding the 4-point base set. The verification of all the 4-point bases is very time-consuming. Therefore, the LSD can help to improve the registration precision and computation efficiency. Also, our method is not sensitive to the number of keypoints, thresholds of the constrains, and iteration number.

The drawback of our method is that the iteration number cannot be determined in advance. The descriptor matching performance depends on the quality of the data, so the proportion of the correct correspondences is unknown. Hence, it is hard to determine the iteration number. However, although the iteration number is set as a big value, the computation time is still small.

REFERENCES

[1] Y. Reiji, D. Hiroaki, K. Satoshi, H. Ryohei, O. Kazuo, and I. Tatsuya, "Automatic registration of MLS point clouds and SfM meshes of urban area," *Geo-Spatial Inf. Sci.*, vol. 19, no. 3, pp. 171–181, 2016, doi: 10.1080/10095020.2016.1212517.

[2] H. Hamidreza, E. Jan, B. Dorit, and N. Andreas, "A study of projections for key point based registration of panoramic terrestrial 3D laser scan," *Geospatial Inf. Sci.*, vol. 18, no. 1, pp. 11–31, 2015, doi: 10.1080/10095020.2015.1017913.

[3] Z. Dong, B. Yang, Y. Liu, F. Liang, B. Li, and Y. Zang, "A novel binary shape context for 3D local surface description," *ISPRS J. Photogrammetry Remote Sens.*, vol. 130, pp. 431–452, 2017, doi: 10.1016/j.isprsjprs.2017.06.012.

[4] W. Tao, X. Hua, K. Yu, X. Chen, and B. Zhao, "A pipeline for 3-D object recognition based on local shape description in cluttered scenes," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 801–816, Jan. 2021, doi: 10.1109/TGRS.2020.2998683.

[5] D. Xu, J. Liu, J. Hyyppa, Y. Liang, and W. Tao, "A heterogeneous 3D map-based place recognition solution using virtual LiDAR and a polar grid height coding image descriptor," *ISPRS J. Photogrammetry Remote Sens.*, vol. 183, pp. 1–18, 2022, doi: 10.1016/j.isprsjprs.2021.10.020.

[6] D. Xu, J. Liu, Y. Liang, X. Lv, and J. Hyyppa, "A LiDAR-based single-shot global localization solution using a cross-section shape context descriptor," *ISPRS J. Photogrammetry Remote Sens.*, vol. 189, pp. 272–288, 2022, doi: 10.1016/j.isprsjprs.2022.05.005.

[7] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992, doi: 10.1109/34.121791.

[8] W. Tao, X. Hua, R. Wang, and D. Xu, "Quintuple local coordinate images for local shape description," *Photogrammetric Eng. Remote Sens.*, vol. 86, no. 2, pp. 121–132, 2020, doi: 10.14358/PERS.86.2.121.

[9] J. Yang, Y. Xiao, and Z. Cao, "Aligning 2.5 D scene fragments with distinctive local geometric features and voting-based correspondences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 714–729, Mar. 2019, doi: 10.1109/TCSVT.2018.2813083.

[10] W. Tao, X. Hua, Z. Chen, and P. Tian, "Fast and automatic registration of terrestrial point clouds using 2D line features," *Remote Sens.*, vol. 12, no. 8, pp. 1283–1299, 2020, doi: 10.3390/rs12081283.

[11] S. M. Hasheminasab, T. Zhou, and A. Habib, "Linear Feature-based image/LiDAR integration for a stockpile monitoring and reporting technology," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 2605–2623, 2023, doi: 10.1109/JSTARS.2023.3250392.

[12] P. Wei, L. Yan, H. Xie, and M. Huang, "Automatic coarse registration of point clouds using plane contour shape descriptor and topological graph voting," *Automat. Construction*, vol. 134, 2022, Art. no. 104055, doi: 10.1016/j.autcon.2021.104055.

[13] T. Rabbani, S. Dijkman, F. van den Heuvel, and G. Vosselman, "An integrated approach for modelling and global registration of point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 61, pp. 355–370, 2007, doi: 10.1016/j.isprsjprs.2006.09.006.

[14] T. O. Chan, D. D. Lichti, D. Belton, and H. L. Nguyen, "Automatic point cloud registration using a single octagonal lamp pole," *Photogrammetric Eng. Remote Sens.*, vol. 82, pp. 257–269, 2016, doi: 10.14358/PERS.82.4.257.

[15] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," *Assoc. Comput. Mach. Trans. Graph.*, vol. 27, pp. 1–10, 2008, doi: 10.1145/1360612.1360684.

[16] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," *Int. J. Comput. Vis.*, vol. 105, pp. 63–86, 2013, doi: 10.1007/s11263-013-0627-y.

[17] J. Yang, Q. Zhang, K. Xian, Y. Xiao, and Z. Cao, "Rotational contour signatures for both real-valued and binary feature representations of 3D local shape," *Comput. Vis. Image Understanding*, vol. 160, pp. 133–147, 2017, doi: 10.1016/j.cviu.2017.02.004.

[18] S. Quan, J. Ma, F. Hu, B. Fang, and T. Ma, "Local voxelized structure for 3D binary feature representation and robust registration of point clouds from low-cost sensors," *Inf. Sci.*, vol. 444, pp. 153–171, 2018, doi: 10.1016/j.ins.2018.02.070.

[19] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999, doi: 10.1109/34.765655.

[20] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 3212–3217, doi: 10.1109/ROBOT.2009.5152473.

[21] J. Yang, Z. Cao, and Q. Zhang, "A fast and robust local descriptor for 3D point cloud registration," *Inf. Sci.*, vol. 346/347, pp. 163–179, 2016, doi: 10.1016/j.ins.2016.01.095.

[22] B. Zhao and J. Xi, "Efficient and accurate 3D modeling based on a novel local feature descriptor," *Inf. Sci.*, vol. 512, pp. 295–314, 2020, doi: 10.1016/j.ins.2019.04.020.

[23] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4PCS fast global point cloud registration via smart indexing," *Comput. Graph. Forum*, vol. 33, no. 5, pp. 205–215, 2014, doi: 10.1111/cgf.12446.

[24] M. Mohamad, D. Rappaport, and M. Greenspan, "Generalized 4-points congruent sets for 3D registration," in *Proc. 2nd Int. Conf. 3D Vis.*, 2014, pp. 83–90, doi: 10.1109/3DV.2014.21.

[25] P. W. Theiler, J. D. Wegner, and K. Schindler, "Keypoint-based 4-points congruent sets-automated marker-less registration of laser scans," *ISPRS J. Photogrammetry Remote Sens.*, vol. 96, pp. 149–163, 2014, doi: 10.1016/j.isprsjprs.2014.06.015.

[26] X. Ge, "Automatic markerless registration of point clouds with semantic-keypoint-based 4-points congruent sets," *ISPRS J. Photogrammetry Remote Sens.*, vol. 130, pp. 344–357, 2017, doi: 10.1016/j.isprsjprs.2017.06.011.

[27] A. Flint, A. Dick, and A. V. D. Hengel, "Local 3D structure recognition in range images," *Inst. Eng. Technol. Comput. Vis.*, vol. 2, no. 4, pp. 208–217, 2008, doi: 10.1049/iet-cvi:20080037.

[28] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Persistent point feature histograms for 3D point clouds," in *Proc. 10th Int. Conf. Intell. Auton. Syst.*, 2008, pp. 119–128, doi: 10.3233/978-1-58003-887-8-119.

[29] S. Malassiotis and M. G. Strintzis, "Snapshots: A novel local surface descriptor and matching algorithm for robust 3D surface alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1285–1290, Jul. 2007, doi: 10.1109/TPAMI.2007.1060.

[30] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. Eur. Conf. Comput. Vis.*, 2010, vol. 6313, pp. 356–369, doi: 10.1007/978-3-642-15558-1_26.

[31] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3D data description," in *Proc. Assoc. Comput. Mach. Workshop 3D Object Retrieval*, 2010, pp. 57–62, doi: 10.1145/1877808.1877821.

[32] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. 8th Eur. Conf. Comput. Vis.*, 2004, pp. 224–237, doi: 10.1007/978-3-540-24672-5_18.

[33] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu, "A novel local surface feature for 3D object recognition under clutter and occlusion," *Inf. Sci.*, vol. 293, pp. 196–213, 2015, doi: 10.1016/j.ins.2014.09.015.

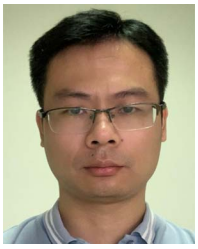
[34] J. Yang, Q. Zhang, Y. Xiao, and Z. Cao, "TOLDI: An effective and robust approach for 3D local shape description," *Pattern Recognit.*, vol. 65, pp. 175–187, 2017, doi: 10.1016/j.patcog.2016.11.019.

[35] K. Tang, P. Song, and X. Chen, "3D object recognition in cluttered scenes with robust shape description and correspondence selection," *IEEE Access*, vol. 5, pp. 1833–1845, 2017, doi: 10.1109/ACCESS.2017.2658681.

[36] F. Ghorbani, H. Ebadi, A. Sedaghat, and N. Pfeifer, "A novel 3-D local DAISY-style descriptor to reduce the effect of point displacement error in point cloud registration," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 2254–2273, 2022, doi: 10.1109/JSTARS.2022.3151699.

[37] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010, doi: 10.1109/TPAMI.2009.77.

[38] J. Yang, S. Quan, P. Wang, and Y. Zhang, "Evaluating local geometric feature representations for 3D rigid data matching," *IEEE Trans. Image Process.*, vol. 29, pp. 2522–2535, 2020, doi: 10.1109/TIP.2019.2959236.
[39] M. Mohamad, M. T. Ahmed, D. Rappaport, and M. Greenspan, "Super generalized 4PCS for 3D registration," in *Proc. Int. Conf. 3D Vis.*, 2015, pp. 598–606, doi: 10.1109/3DV.2015.74.
[40] S. Li, R. Lu, J. Liu, and L. Guo, "Super edge 4-Points congruent sets-based point cloud global registration," *Remote Sens.*, vol. 13, pp. 3210–3245, 2021, doi: 10.3390/rs13163210.
[41] Y. A. Felus and R. C. Burtch, "On symmetrical three-dimensional datum conversion," *GPS Sol.*, vol. 13, pp. 65–74, Jan. 2009, doi: 10.1007/s10291-008-0100-5.
[42] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "An integrated framework for 3-D modeling, object detection, and pose estimation from point-clouds," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 3, pp. 683–693, Mar. 2015, doi: 10.1109/TIM.2014.2358131.
[43] J. Park, Q. Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 143–152, doi: 10.1109/ICCV.2017.25.
[44] B. Zeisl, K. Koeser, and M. Pollefeys, "Automatic registration of RGB-D scans via salient directions," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2805–2815, doi: 10.1109/ICCV.2013.349.
[45] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. Assoc. Comput. Mach.*, vol. 24, no. 6, pp. 381–395, 1981, doi: 10.1145/358669.358692.



Wuyong Tao received the master's degree in surveying and mapping from the East China University of Technology, Nanchang, China, in 2015, and the Ph.D. degree in geodesy from Wuhan University, Wuhan, China, in 2020.

He was a visiting Ph.D. student with the University of Calgary, Calgary, AB, Canada, from 2019 to 2020. He is currently a Lecturer with Nanchang University, Nanchang, China. He has published more than 30 research papers. He has been a Reviewer for *Measurement and Control, Photonics, Journal of Intelligent and Fuzzy Systems, Forests, Sensors*, and *Remote Sensing*. His research interests include point cloud registration, 3D object recognition, and deep learning.

Jingbin Liu received the Ph.D. degree in surveying and mapping from Wuhan University, Wuhan, China, in 2008.

He was a Specialist Research Scientist with Finnish Geospatial Research Institute (FGI, formerly known as Finnish Geodetic Institute) from 2008 to 2016. He is a Professor in Positioning and Navigation with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. His research interest areas include indoor and outdoor positioning, smartphone navigation, indoor mobile mapping, and GNSS/INS/SLAM integration technology.

Dong Xu received the B.S. degree in surveying and mapping engineering in 2017 from the China University of Mining and Technology, Xuzhou, China, and the M.S. degree in surveying and mapping engineering in 2019 from Wuhan University, Wuhan, China, where he is currently working toward the doctoral degree.

His research interests include laser scanning and data processing.

Yanyang Xiao received the B.Eng. degree in mechanical engineering from Hunan University, Changsha, China, in 2012, and the Ph.D. degree in computer science from Xiamen University, Xiamen, China, in 2020.

He is currently an Assistant Professor with the School of Mathematics and Computer Sciences, Nanchang University, Nanchang, China. His research interests include computer graphics and point cloud processing.