

Cross-source Registration of Point Clouds in Urban Scenes using Structured Features

Shu Peng¹, Minglei Li^{*1,3}, Junnan Zhang¹, Jiarui Tang²

¹ College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics,
211106 Nanjing, China - (ps0272, minglei_li, zhangjunnan)@nuaa.edu.cn;

² College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics,
211106 Nanjing, China - 2997794724@qq.com;

³ Key Laboratory of Radar Imaging and Microwave Photonics (Nanjing University of Aeronautics and Astronautics),
Ministry of Education, 211106 Nanjing, China;

Keywords: Cross-source Point cloud, Coarse registration, Structured Features, Urban Scene.

Abstract

Cross-source point cloud registration technology offers the potential to harness the complementary advantages of multiple data sources by registering and integrating point clouds from diverse origins. This paper proposes a cross-source point cloud coarse registration method based on structured features in urban scenes. Firstly, we extract adjacent plane intersection lines and vertical plane boundary lines from the vertical planes of the building point cloud. Subsequently, we construct triangles based on the intersection of vertical feature lines with the ground, and use geometric constraints and semantic information for triangle matching. Finally, quick validation and fine validation are sequentially employed to determine the optimal coarse registration transformation matrix. Our experimental results demonstrate that, in comparison to point feature-based and similar point cloud coarse registration methods, the proposed method exhibits superior average accuracy, efficiency, and robustness.

1. INTRODUCTION

The digitization of the urban landscape serves a critical role in spatial planning, heritage conservation, and smart city initiatives. Among the technologies enabling this digital transformation are Light Detection and Ranging (LiDAR) and photogrammetry, each contributing unique strengths to the field of 3D reconstruction. LiDAR point clouds provide dense and accurate data which is particularly robust to changes in lighting conditions, ensuring consistent performance regardless of the time of day or weather conditions. However, LiDAR systems typically lack the rich texture and color information that is crucial for visual analytics, and they can fall short in representing fine surface details compared to photogrammetric methods (Zhang and Lin, 2017).

Conversely, photogrammetry, which relies on passive acquisition of images, excels in capturing surface textures and color information necessary for creating photo-realistic 3D models. The high-resolution imagery utilized in photogrammetry can provide detailed information about material properties and aesthetics of the urban features. Despite these strengths, photogrammetric point clouds usually have deficiencies in terms of absolute accuracy, particularly in dense urban areas with high-rise structures and varying terrain, leading to challenges in scale and perspective.

The registration and fusion of LiDAR and photogrammetric point clouds address these individual limitations by combining the reliable geometric framework from the former with the detailed textural and color information of the latter, thus enabling a more comprehensive and enriched representation of urban scenes (Li et al., 2024). However, the process is non-trivial due to differences in data characteristics, formats, density, and acquisition geometry. The risks of misalignment and data redundancy are significant, and require sophisticated processing techniques.

This paper aims to explore the technical challenges involved in the registration and fusion of urban LiDAR and photogrammetric point clouds and proposes an effective methodological framework to combine the strengths of both technologies. Through our research, we aspire to enhance the quality of urban 3D models, offering richer and more precise spatial data to support decision-making in areas such as urban planning and smart city construction.

2. RELATED WORK

3D point cloud registration refers to the process of converting 3D point cloud data from different coordinate systems to the same coordinate system. The overall process can be divided into two steps: coarse point cloud registration and fine point cloud registration. Point cloud coarse registration is often performed without any prior knowledge, so coarse registration requires solving a larger range of coordinate transformation problems. Based on the types of feature elements used and the characteristics of point cloud registration methods, this article divides point cloud coarse registration methods into three categories: point based registration, geometric based registration, and learning based registration methods.

The point cloud fine registration is performed after the point cloud coarse registration, with the aim of optimizing the results of the point cloud coarse registration and obtaining more accurate registration results. In order to achieve high-precision registration, point cloud fine registration algorithms generally use iterative methods to continuously optimize the registration results. Many professionals at home and abroad are constantly optimizing algorithms, from Iterative Closest Point (ICP) (Besl and McKay, 1992) and 3D Normal Distribution Transform (3D-NDT) (Magnusson et al., 2007) to Granger and Pennec (Granger and Pennec, 2002) proposing an Expectation Maximization Iterative Closest Point (EM-ICP) algorithm that combines multi-scale and annealing schemes based on probability statistical

models, or Fitzgibbon(Fitzgibbon, 2003) improving the convergence speed and accuracy of ICP algorithm by introducing Levenberg Marquardt algorithm, these new algorithms are improving and promoting the development of point cloud registration.

Heterogeneous point cloud refers to the point cloud data obtained by different sensor systems, which has obvious differences in scale, resolution, color characteristics, overlap rate and noise level. Researchers have developed many methods for registering heterogeneous point clouds based on points, graph theory, and structural features. Although point based registration methods are difficult to search for point correspondences in heterogeneous point clouds, they have strong universality and can be applied to different types of point cloud scenes. Therefore, researchers have adapted and improved traditional point based registration methods for heterogeneous point cloud registration. Graph theory methods can effectively characterize complex structural relationships in point cloud data and have good robustness to noise and outliers. Therefore, graph theory based methods are an effective way to solve the problem of heterogeneous point cloud registration. However, the existing methods for registering heterogeneous point clouds are still not perfect and mature enough, and it is necessary to conduct more research on the registration of heterogeneous point clouds.

These technical methods should be referenced from FPF-HSAC (Rusu et al., 2009), K4PCS (Theiler et al., 2014), FGR (Zhou et al., 2016), and PLADE (Chen et al., 2019).

3. PROPOSED METHOD

3.1 Overview of the methodology

The implementation of the proposed coarse registration method comprises 4 fundamental steps: 1) extraction of vertical feature lines, 2) matching of structured features, and 3) transformation estimation and 4) parameter verification. Detailed instructions for each step will be expounded upon in the subsequent sections.

As point clouds in urban scenes often contain an excessive amount of raw data, voxel grid down-sampling is used, dividing the point cloud into uniform voxel units and replacing points with centroids to significantly reduce the data volume. Additionally, statistical filtering method is applied to eliminate noise from cross-source point cloud data, minimizing the impact of noise on feature extraction and registration processes. Subsequently, weighted principal component analysis is utilized to estimate the normal vectors of the point clouds. The registration method presented in this paper utilizes vertical feature lines from building point clouds and the maximum plane from ground point clouds to create structured features. Furthermore, we note that cross-source vegetation point clouds can substantially interfere with subsequent planar feature extraction and registration. Based on this, we employ the Cloth Simulation Filter (CSF) algorithm(Zhang et al., 2016) to separate ground point clouds and then apply the Euclidean clustering algorithm with normal vector angle constraints to differentiate between vegetation and building point clouds in urban scenes.

3.2 Extraction of vertical feature lines

In order to make the registration more robust, we consider increasing the number and types of features. Therefore,

this paper introduces two types of vertical feature lines. This method first utilizes the RANSAC algorithm(Schnabel et al., 2007) to accurately extract planar features from building point clouds. Subsequently, adjacent plane intersection lines and vertical plane boundary lines are extracted based on all vertical planes, as shown in figure 1. The following are the methods for extracting these two types of feature lines:

Adjacent plane intersection lines. Firstly, the intersection line of two non-parallel vertical planes is computed based on their parameters. If the distance between the current intersection line and the point clouds of the two vertical planes meets the specified distance constraint, the intersection line is considered valid.

Vertical plane boundary lines. This method first extracts boundary points from each vertical plane point cloud, and then fits lines in the boundary points using the RANSAC algorithm combined with the number of line point cloud, number and direction of lines.

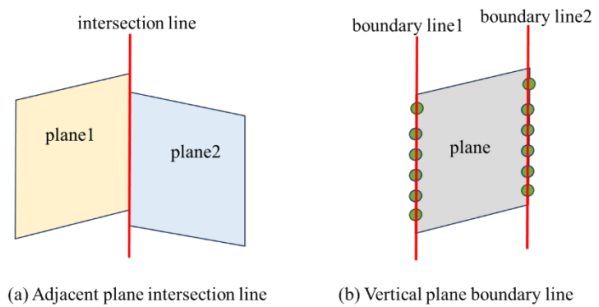


Figure 1. Extraction of Vertical Feature Lines.

3.3 Matching of structured features

3.3.1 Triangle construction In this paper, we obtain distinctive feature points by intersecting the ground plane with the extracted vertical feature lines. These feature points are utilized to construct triangles based on the spatial topological relationship between them. To achieve this, we adopt the triangle construction method proposed in GlobalMatch(Wang et al., 2023), which is based on the K-nearest neighbors. Specifically, we establish a KD tree for the feature points and retrieve only the most recent adjacent points for each feature point. In the triangle construction process, we consider a feature point, p_i , as one point of the triangle and select any two points from the adjacent points to form potential triangles. However, if the selected three points are collinear, the triangle condition is not satisfied and the triangle is disregarded. Figure 2 shows the triangle construction method with p_i as the search point when the number of domain points is 4.

After constructing the triangles, the order of the vertices (structure level key points) recorded in each triangle set is irregular, which undoubtedly increases the difficulty of subsequent matching. Based on this, this article takes the vertex corresponding to the longest side of the triangle as the first determined point in the set of triangle vertices, and then rearranges the order of the remaining two vertices in the triangle through geometric space constraints.

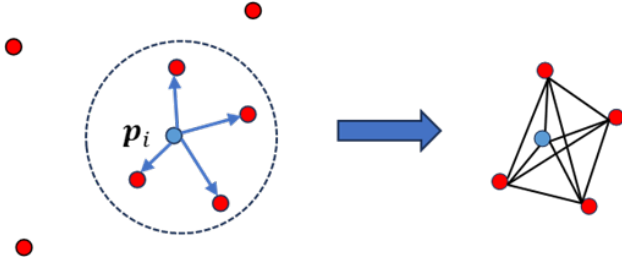


Figure 2. Triangle Construction Based on K-Nearest Neighbors.

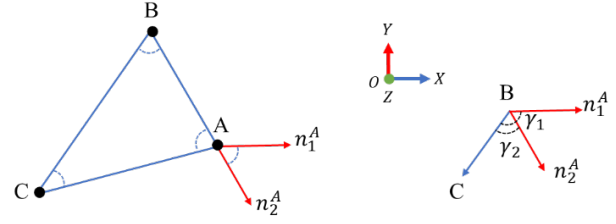


Figure 3. Construction of normal vector angle for plane projection.

3.3.2 Triangle matching We were inspired by reference (Yang et al., 2016) to introduce semantic information and geometric constraints in feature description and matching. Then, based on the characteristics of cross-source point clouds, we proposed a robust triangle matching method. Firstly, based on the semantic type of vertices and the direction of the supporting plane, vertex descriptors are constructed. Then, triangles are mapped to the feature space according to the size of their interior angles. Next, KD trees are used to match candidate corresponding triangles. Finally, the vertex descriptors of each vertex are compared in the corresponding triangle pairs to determine the correspondence between the triangles. Next, we will introduce the establishment of vertex descriptors and triangle matching.

Establishment of vertex feature descriptor Each triangle vertex is associated with specific attributes, comprising the point coordinate, $p_i = (x, y, z)$, semantic label, L_{Label} , and unit normal vectors of the support planes, $n_1 = (n_{x1}, n_{y1}, n_{z1})$ and $n_2 = (n_{x2}, n_{y2}, n_{z2})$, denoted as $\Gamma_i = p_i, L_{Label}, n_1, n_2$. The semantic labels encompass two categories: the intersection of adjacent vertical planes and the boundary lines of vertical planes. In the case of a vertex originating from the intersection of adjacent vertical planes, L_{Label} is set to 0, and both n_1 and n_2 are valid. Conversely, when a vertex arises from a vertical plane boundary line, L_{Label} is set to 1, signifying that the boundary line is determined by only one plane, rendering n_1 valid while n_2 is deemed invalid.

As the normal vector of the vertical plane is nearly parallel to the XOY plane, it is projected onto the XOY plane, resulting in a projected normal vector that shares the same horizontal coordinate system with the vector formed by the triangle vertices. This projected normal vector maintains the same minimum angle with the vector formed by the two vertices of the triangle both before and after the similarity transformation. Hence, this angle can serve as a condition for the corresponding vertices. We assume A is the intersection point between adjacent vertical planes and the ground, where Π_1^A and Π_2^A are two support planes. n_1^A and n_2^A are projection vectors of the plane normal vectors onto the XOY plane respectively. The angles γ_1 and γ_2 between the vector BC, and the vectors n_1^A and n_2^A are calculated respectively. The construction method of γ_1 and γ_2 is illustrated in Figure 3. The feature descriptors for each vertex are:

$$[L_{Label} = 0 \quad \beta_1 = \min(\gamma_1, \gamma_2) \quad \beta_2 = \max(\gamma_1, \gamma_2)] \quad (1)$$

When vertex A corresponds to the intersection of vertical boundary lines with only one valid plane, the descriptor is defined as follows:

$$[L_{Label} = 1 \quad \beta_1 = \gamma_1] \quad (2)$$

Matching of triangles The process of triangle matching method is as follows:

1. Initially, based on the three inner angles of the triangle, each triangle constructed from the source and target point clouds is mapped to the three-dimensional feature space, with one mapping point for each triangle Δp_i . The coordinate of Δp_i is denoted as $(\theta_{\Delta A}, \theta_{\Delta B}, \theta_{\Delta C})$, where $(\theta_{\Delta A}, \theta_{\Delta B}, \theta_{\Delta C})$ represent the inner angle values of the three vertices of a triangle, with the vertices arranged clockwise.

2. A KD tree for the triangle mapping points of the source point cloud is established, and radius neighborhood search is employed to locate the corresponding triangle in the target point cloud. Set the search radius to d^A . For the triangle mapping points in the target triangle Δp_i^t , the neighborhood points within d^A are $\{\Delta p_j^t, j = 1, \dots, K\}$. Traverse all triangle mapping points in the target triangle to obtain all candidate matching triangles. Subsequently, perform vertex consistency checks on each pair of candidate matching triangles to further confirm their correspondence.

3. For each initial corresponding triangle pair matched in step 2, the descriptors of each corresponding vertex in the triangle are compared to determine their correspondence. The matching method involves comparing each item in the vertex descriptors. Firstly, the label type, L, of the vertex is checked for consistency. Inconsistent labels result in matching failure, while consistent labels lead to the determination of the vertex type. If the vertex is of the type of the intersection of adjacent vertical planes, equation (1) is used to verify the consistency of β_1 and β_2 . If the difference between corresponding β_1 and β_2 is less than the angle threshold θ_{th}^A , the vertex is represented as corresponding. Similarly, for vertices of the vertical plane boundary line type, equation (2) is employed to verify the consistency of β_1 . If the difference of corresponding β_1 is less than the angle threshold θ_{th}^A , the vertex is represented as corresponding. Only when all three vertices of a triangle correspond, is it considered a corresponding triangle.

3.4 Estimation of transformation matrix

For a set of three-dimensional corresponding point sets $C = \{(p_i^s, p_i^t), i = 1, 2, \dots, N_c\}$, where p_i^s represents the source point cloud, p_i^t represents from the target point cloud, N_c is the number of corresponding points. The similarity transformation relationship between p_i^s and p_i^t can be expressed as:

$$p_i^t = sRp_i^s + t \quad (3)$$

where $s > 0$ is the scale factor, R is a 3×3 orthogonal rotation matrix, t is a 3×1 translation vector. Due to measurement errors between corresponding points, the objective function can be

defined as the sum of squared residuals of the distances between point pairs. The point cloud registration model based on point correspondence can be represented by the following equation:

$$(s^*, \mathbf{R}^*, \mathbf{t}^*) = \underset{s, \mathbf{R}, \mathbf{t}}{\operatorname{argmin}} \sum_{i=1}^{N_c} \|\mathbf{p}_i^t - (s\mathbf{R}\mathbf{p}_i^s + \mathbf{t})\|^2 \quad (4)$$

This paper calculates the transformation matrix based on three corresponding vertices in a matched triangle pair. In the calculation of the transformation matrix, we decompose the calculation of the scale factor, rotation matrix, and translation vector into three sub problems.

Estimation of scale factor. The transformation scale between point clouds is equal to the ratio of the distance

$$s^* = \frac{1}{3} \left(\frac{\|\mathbf{p}_1^t - \mathbf{p}_2^t\|}{\|\mathbf{p}_1^s - \mathbf{p}_2^s\|} + \frac{\|\mathbf{p}_2^t - \mathbf{p}_3^t\|}{\|\mathbf{p}_2^s - \mathbf{p}_3^s\|} + \frac{\|\mathbf{p}_1^t - \mathbf{p}_3^t\|}{\|\mathbf{p}_1^s - \mathbf{p}_3^s\|} \right) \quad (5)$$

between any two corresponding pairs of vertices.

Estimation of rotation matrix. We use Singular Value Decomposition (SVD) to calculate the rotation matrix $\mathbf{R}$.

Calculate the centroids of the source and target triangles:

$$\mu^s = \frac{1}{3} (\mathbf{p}_1^s + \mathbf{p}_2^s + \mathbf{p}_3^s), \mu^t = \frac{1}{3} (\mathbf{p}_1^t + \mathbf{p}_2^t + \mathbf{p}_3^t) \quad (6)$$

Subtract centroid from each vertex:

$$\tilde{\mathbf{p}}_i^s = \mathbf{p}_i^s - \mu^s, \tilde{\mathbf{p}}_i^t = \mathbf{p}_i^t - \mu^t \quad (7)$$

Calculate the covariance matrix $\mathbf{H}$:

$$\mathbf{H} = \sum_{i=1}^3 \tilde{\mathbf{p}}_i^s (\tilde{\mathbf{p}}_i^t)^T \quad (8)$$

Decompose $\mathbf{H}$ to obtain:

$$\mathbf{H} = \mathbf{U} \sum \mathbf{V}^T \quad (9)$$

$$\mathbf{R}^* = \mathbf{V} \mathbf{U}^T \quad (10)$$

Estimation of translation vector. We use the obtained s^* and $\mathbf{R}^*$ to calculate the translation vector $\mathbf{t}^*$:

$$\mathbf{t}^* = \mu^t - s^* \mathbf{R}^* \mu^s \quad (11)$$

The transformation matrix $\mathbf{T}$ can be represented as follows:

$$\mathbf{T} = [s^* \mathbf{R}^* \quad \mathbf{t}^* \mathbf{0}^T \quad 1] \quad (12)$$

After calculating the transformation matrix based on all matching triangles, obtain the set of transformation matrices $\Psi = \mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_n$.

3.5 Verification of the transformation matrix

To ensure the success of point cloud coarse registration, we propose a two-level verification method for evaluating transformation matrices. Firstly, a quick verification method for transformation matrices based on bidirectional KD trees is performed, followed by a fine verification method for transformation matrices based on octree.

(1) Quick Verification

During the quick verification stage, the confidence of the transformation matrix is assessed by counting the number of corresponding pairs between the transformed source and target keypoints. To mitigate one-to-many errors during correspondence searches, we employ a bidirectional KD tree approach based on Euclidean distance to identify corresponding pairs, subsequently ranking the transformation matrices by the number of matched pairs. Only the top N_{qf} matrices with the most corresponding pairs are selected for fine verification in subsequent stages.

(2) Fine Verification

The fine verification of the transformation matrix involves evaluating the global similarity between two sets of point clouds using the current transformation matrix. However, computing the global similarity necessitates processing the entire original point cloud, which can be challenging due to the large number of point clouds typically found in urban scenes, leading to potential efficiency issues. To address this, it is crucial to employ suitable methods for organizing large-scale point cloud data. Octree, a highly efficient data structure, offers superior time efficiency compared to KD tree in voxel search. Building upon this idea, this paper adopts the confidence calculation method based on octree, proposed in reference (Li et al., 2022), to estimate the similarity between the transformed source point cloud and the target point cloud. This approach effectively verifies the transformation matrix with high accuracy.

4. EXPERIMENTS

4.1 Dataset and evaluation metrics

(1) Dataset

In this paper, we employ a self-built cross-source point cloud dataset to validate the proposed cross-source point cloud registration method. The dataset is collected in a university campus. The dataset comprises three distinct sets of data, denoted as Campus pair1, Campus pair2, and Campus pair3. Each set comprises a collection of LiDAR point clouds and photogrammetric point clouds, obtained from different sources. The LiDAR point clouds are derived from two sources: one obtained through the Leica BLK 360 ground laser scanner and the other captured using the FAST-LIO2 (Xu et al., 2022) SLAM system. The image data was acquired using a DJI Mini3 drone, and the photogrammetric point clouds were generated using Context Capture, two different 3D reconstruction software.

Detailed information for each data group in the dataset is presented in Table 1, while the visualization of point clouds is illustrated in Figure 4. The dataset reveals significant disparities in data size, bounding box size, and coverage range among point clouds obtained through diverse LiDAR and photogrammetric techniques.

The accurate transformation matrix values between each LiDAR point cloud and photogrammetric point cloud group were obtained through meticulous manual point selection and registration optimization.

The algorithm developed in this study is implemented using C++ language and Visual Studio 2017. The implementation

scene	Data source	Data source	Point number	Bounding box size/m	Number of images/stations
Campus pair1	LiDAR	FAST-LIO2 SLAM	22757296	314.2×273.2×100.6	-
	Image	Pix4d Mapper	6674566	320.7×333.4×133.5	1538
Campus pair2	LiDAR	FAST-LIO2 SLAM	16686380	127.1×122.6×56.1	-
	Image	Context Capture	87947530	94.3×92.7×29.9	1575
Campus pair3	LiDAR	BLK360	133813207	153.5×152.6×38.9	12
	Image	Context Capture	85610231	58.5×54.4×13	1436

Table 1. Self built cross-source point cloud dataset

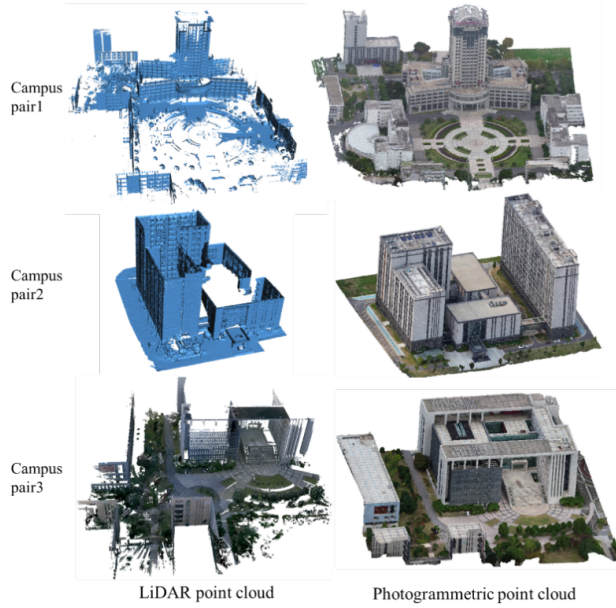


Figure 4. Self built cross-source point cloud dataset.

process involved the application of open-source libraries such as PCL and Ceres. The computer environment is Windows 11, with an Intel i5-11400h CPU and 16GB of memory.

(2) Evaluation

In this paper, the proposed cross-source point cloud registration algorithm is comprehensively evaluated using both qualitative and quantitative analyses. The qualitative evaluation involves visual inspection of the registered point clouds to assess the quality of the registration results. The quantitative evaluation focuses on two main aspects: registration accuracy and computational efficiency.

To assess registration accuracy, the algorithm is evaluated in terms of transformation matrix errors and the root mean square error (RMSE). The transformation matrix errors consist of scale error (E^s), rotation error (E^R), and translation error (E^t), which are calculated as shown in equations (5.1):

$$\begin{cases} E^s = |s^g - s^e| \\ E^R = \arccos\left(\frac{\text{trace}(\mathbf{R}^g(\mathbf{R}^e)^{-1}) - 1}{2}\right) \\ E^t = \|\mathbf{t}^g - \mathbf{t}^e\| \end{cases} \quad (13)$$

where $\text{trace}()$ denotes the trace of a matrix, and s^e , $\mathbf{R}^e$, and $\mathbf{t}^e$ represent the estimated scale, rotation matrix, and translation vector obtained from the proposed algorithm. Correspondingly,

s^g , $\mathbf{R}^g$, and $\mathbf{t}^g$ denote the ground truth scale, rotation matrix, and translation vector.

The calculation formula for RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} \|s^e \mathbf{R}^e \mathbf{p}_i + \mathbf{t}^e - (s^g \mathbf{R}^g \mathbf{p}_i + \mathbf{t}^g)\|_2^2} \quad (14)$$

where N_s represents the number of points in the source point cloud, $\mathbf{p}_i$ represents the position of i th point in the source point cloud.

4.2 Vertical feature line extraction

The extraction results of vertical feature lines and structural key points are shown in Figure 5, and the number of extracted features is shown in Table 2. According to the experimental results, the number of adjacent plane intersection lines extracted in the point cloud of Campus pair 1 is less than the number of plane boundary lines, while the extraction results in Campus pair 2 and Campus pair 3 are the opposite. Only 2 adjacent plane intersection lines were extracted from the photogrammetric point cloud of Campus pair 1, but 33 plane boundary lines were extracted. Therefore, by combining the intersection lines of adjacent vertical planes and the boundary lines of vertical planes, the types and quantities of features are increased, providing a sufficient number of structural level key points for the point cloud coarse registration method in this paper.

4.3 Experimental results

Figure 6 shows the results of our point clouds coarse registration method in three sets of data, with each set of data presenting results from four perspectives: front view, side view, top view, and local view. By observing the shape and contour features in each view of the experimental results, it can be seen that the landmark objects such as buildings and floors in the scene have been roughly aligned and there have been no major matching errors. From the color boundaries between point clouds, it can be seen that in areas with deviations, some point clouds have more prominent colors, indicating that some areas have not fully matched and there is a slight misalignment. Overall, our coarse registration method successfully achieved rough alignment of three sets of cross-source point cloud data.

We selected four representative point cloud coarse registration methods as comparison methods, namely FPF-HSAC(Rusu et al., 2009), K4PCS(Theiler et al., 2014), FGR(Zhou et al., 2016), and PLADE(Chen et al., 2019). Due to their inability to handle unknown scale issues in cross-source point cloud data, the scale of the source point cloud was manually restored using real scales before conducting point cloud

scene	Data source	Number of Vertical planes	Number of adjacent plane intersection lines	Number of Vertical plane boundary lines	Number of structural level key points
Campus pair1	LiDAR	35	14	31	45
	Image	21	2	33	35
Campus pair2	LiDAR	15	17	6	23
	Image	21	17	7	24
Campus pair3	LiDAR	23	19	13	32
	Image	25	24	8	32

Table 2. Results of Feature Extraction

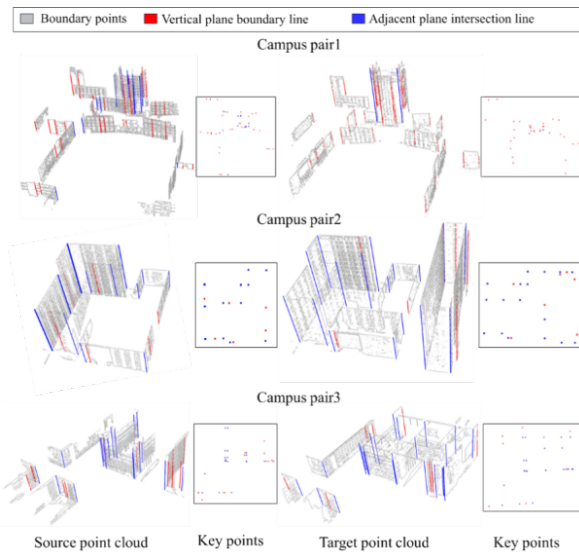


Figure 5. Extraction Results of Vertical Feature Lines and Structural Key Points.

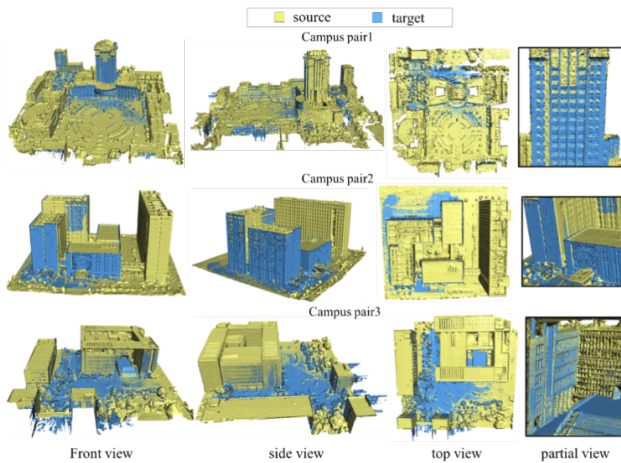


Figure 6. Coarse registration results from different perspectives.

registration experiments. The registration results of coarse registration and other methods in this paper are shown in Figure 7. The registration error and time are shown in Table 3.

From the experimental results, FPFHSAC, K4PCS, and FGR all failed to register under the three sets of data. There are two main reasons for the registration failure. Firstly, there are many similar smooth structures in the urban scene point cloud, and the extracted point features are highly similar. Secondly,

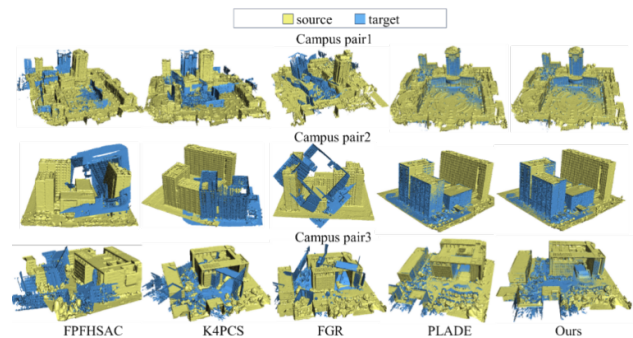


Figure 7. Results of point cloud coarse registration.

there are significant differences in the local geometric characteristics between cross-source point clouds, resulting in low reliability of the extracted point features. In these cases, matching the points correspondence is very difficult. In summary, traditional point based registration methods have poor robustness and are difficult to register cross-source point clouds in urban scenes with significant differences in density, noise levels, and other aspects.

PLADE and the coarse registration method proposed in this article are both based on structured features. PLADE successfully registered Campus pair1 and Campus pair2, and obtained good rotation parameters in Campus pair3. However, the translation error and root mean square error were greater than 3m, resulting in registration failure. Our coarse registration method successfully registered three sets of heterogeneous point cloud data, with an average error of scale error 0.0067, rotation error 0.29°, translation error 0.2002m, and root mean square error 0.4357m. The PLADE algorithm has better accuracy than the coarse registration method proposed in this paper in some scenarios, but it is not as good as the coarse registration method proposed in this paper in terms of average accuracy and robustness. In addition, our coarse registration method requires the least amount of time among all the compared methods. The superior average accuracy and robustness of our method stem from its reliance on a minimal number of structural-level key-points. By utilizing geometric constraints and semantic information for corresponding matching, it effectively enhances the algorithm’s robustness and efficiency.

4.4 Robustness experiment

The coarse registration stage of point cloud registration covers the most difficult part of the registration process, which requires the ability to handle issues such as different point densities, noise, and data loss. The robustness of coarse registration algorithms is crucial for achieving accurate registration. The

scene	metrics	FPFHSAC	K4PCS	FGR	PLADE	Ours
Campus pair1	E^s	Manual	Manual	Manual	Manual	0.0048
	$E^R(^{\circ})$	6.204	105.1	154.2	0.4177	0.4064
	$E^t(m)$	50.77	40.54	37.85	0.1561	0.1822
	$RMSE(m)$	52.52	160.1	193.4	0.581	0.6089
	Time(s)	152.3	49.2	371.5	168.5	5.5
Campus pair2	E^s	Manual	Manual	Manual	Manual	0.0091
	$E^R(^{\circ})$	157.6	22.51	148.5	0.189	0.2222
	$E^t(m)$	33.62	44.75	38.06	0.3279	0.3047
	$RMSE(m)$	87.49	47.01	100.4	0.4861	0.4531
	Time(s)	129.4	42.2	178.6	24.7	10.6
Campus pair3	E^s	Manual	Manual	Manual	Manual	0.0063
	$E^R(^{\circ})$	142.6	53.54	30.26	0.2969	0.2415
	$E^t(m)$	31.81	12.85	19.97	3.184	0.1138
	$RMSE(m)$	117.7	52.68	35.72	3.136	0.2452
	Time(s)	185.3	58.7	169.2	192.4	6.3

Table 3. Comparison of coarse registration methods

robustness of the coarse registration algorithm proposed in this paper is tested on a self built dataset from three dimensions: different data missing, different point densities, and different noise levels.

(1) Different data missing conditions

Multiple different perspectives were applied to crop the source point cloud in Campus Pair 3, resulting in a total of ten shapes. The cropped source point cloud was then registered with the target point cloud in Campus Pair 3. The cropped source point cloud and registration result are shown in Figure 8, where yellow represents the source point cloud and blue represents the target point cloud. The registration error is shown in Figure 9. According to the experimental results, all ten sets of data were successfully registered, achieving scale errors of less than 0.01, rotation errors of less than 0.35°, translation errors of less than 0.4m, and root mean square errors of less than 0.5m. The experimental results fully demonstrate the robustness of our coarse registration method to different data missing situations.

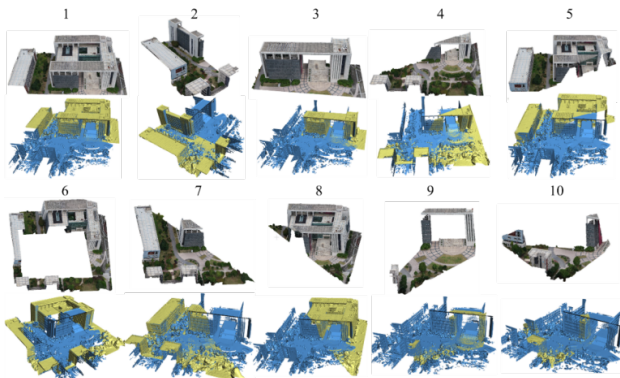


Figure 8. Registration results under different data missing conditions.

(2) Different point cloud densities

Sample the source point cloud points of the three sets of data to 80%, 60%, 40%, 20%, and 10% of the denoised point cloud points, respectively. The registration errors at different

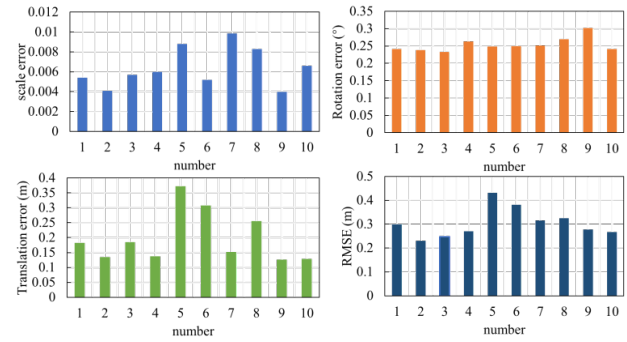


Figure 9. Registration results under different data missing conditions.

point densities are shown in Figure 10. The experimental results show that as the point cloud density decreases, the coarse registration error in this paper fluctuates with relatively small numerical changes, achieving scale errors of less than 0.014, rotation errors of less than 0.7°, translation errors of less than 0.5m, and root mean square errors of less than 0.9m. Even at 10% of the original point density, our coarse registration method can maintain good accuracy, indicating its strong robustness to changes in point density.

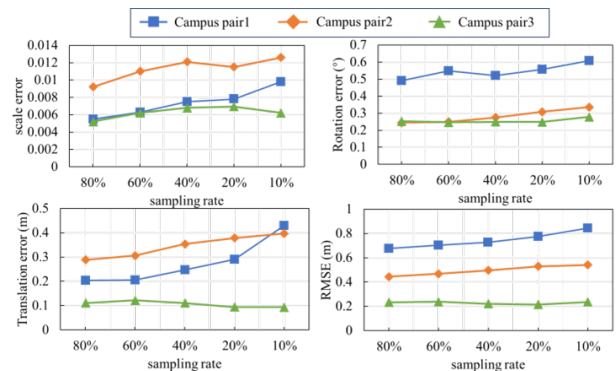


Figure 10. Registration errors at different point densities.

(3) Different noise levels

Add Gaussian noise with five different standard deviations to the source point clouds of three sets of data, where the standard deviations are 4cm, 8cm, 12cm, 16cm, and 20cm, respectively. Figure 11 shows the registration errors at different noise levels. From the experimental results, it can be seen that as the noise level increases, the registration error changes at an acceptable level, achieving scale errors less than 0.02, rotation errors less than 0.6° , translation errors less than 0.6m, and root mean square errors less than 0.9m. Therefore, our coarse registration method has strong robustness under different noise levels.

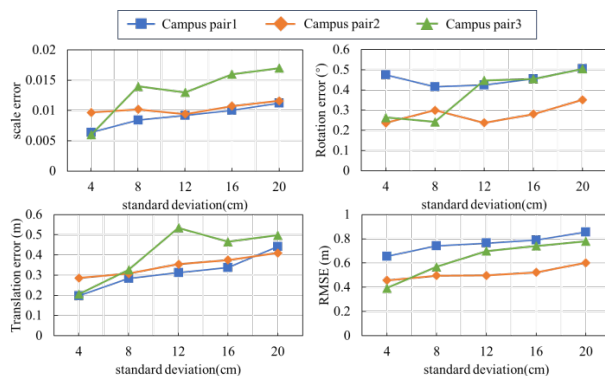


Figure 11. Registration errors at different noise levels.

5. CONCLUSIONS

We propose a cross-source point cloud coarse registration method based on structured features in urban scenes. Firstly, based on the vertical plane in the building point cloud, adjacent plane intersection lines and vertical plane boundary lines are extracted. The combination of these two feature lines enriches the number and types of vertical feature lines, providing sufficient feature support for subsequent key point generation and registration. Next, the intersection point between the vertical feature lines and the ground is used as a structural level key point, and triangles are constructed using the geometric spatial relationship between the key points. Subsequently, we establish triangle matching criteria based on geometric constraints and semantic information. Finally, we propose a two-level validation method to determine the optimal transformation matrix. The experimental results show that compared with point-feature-based and same types of point cloud coarse registration methods, using structured features as the fundamental elements for registration proves to be more robust and faster when processing point clouds acquired from different sensors.

References

Besl, P. J., McKay, N. D., 1992. Method for registration of 3-d shapes. *Sensor fusion IV: control paradigms and data structures*, 1611, Spie, 586–606.

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

Fitzgibbon, A. W., 2003. Robust registration of 2D and 3D point sets. *Image and vision computing*, 21(13-14), 1145–1153.

Granger, S., Pennec, X., 2002. Multi-scale em-icp: A fast and robust approach for surface registration. *Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part IV* 7, Springer, 418–432.

Li, J., Zhan, J., Zhou, T., Bento, V. A., Wang, Q., 2022. Point cloud registration and localization based on voxel plane features. *ISPRS Journal of Photogrammetry and Remote Sensing*, 188, 363–379.

Li, M., Peng, S., Nan, L., 2024. Hybrid geometry sets for global registration of cross-source geometric data. *International Journal of Applied Earth Observation and Geoinformation*, 128, 103733.

Magnusson, M., Lilienthal, A., Duckett, T., 2007. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10), 803–827.

Rusu, R. B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (fpfh) for 3d registration. *2009 IEEE international conference on robotics and automation*, IEEE, 3212–3217.

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient ransac for point-cloud shape detection. *Computer graphics forum*, 26number 2, Wiley Online Library, 214–226.

Theiler, P. W., Wegner, J. D., Schindler, K., 2014. Keypoint-based 4-points congruent sets—automated marker-less registration of laser scans. *ISPRS journal of photogrammetry and remote sensing*, 96, 149–163.

Wang, X., Yang, Z., Cheng, X., Stoter, J., Xu, W., Wu, Z., Nan, L., 2023. GlobalMatch: Registration of forest terrestrial point clouds by global matching of relative stem positions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197, 71–86.

Xu, W., Cai, Y., He, D., Lin, J., Zhang, F., 2022. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4), 2053–2073.

Yang, B., Dong, Z., Liang, F., Liu, Y., 2016. Automatic registration of large-scale urban scene point clouds based on semantic feature points. *ISPRS Journal of Photogrammetry and Remote Sensing*, 113, 43–58.

Zhang, J., Lin, X., 2017. Advances in fusion of optical imagery and LiDAR point cloud applied to photogrammetry and remote sensing. *International Journal of Image and Data Fusion*, 8(1), 1–31.

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

Zhou, Q.-Y., Park, J., Koltun, V., 2016. Fast global registration. *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II* 14, Springer, 766–782.