remote sensing

MDPI

Article

Automatic Point Cloud Registration for Large Outdoor Scenes Using a Priori Semantic Information

Jian Li [1], Shuowen Huang [1,*], Hao Cui [1], Yurong Ma [2] and Xiaolong Chen [3]

[1] School of the Geo-Science & Technology, Zhengzhou University, Zhengzhou 450001, China; jianli@zzu.edu.cn (J.L.); cuihao@zzu.edu.cn (H.C.)
[2] University Library, Zhengzhou University, Zhengzhou 450001, China; mayurong@zzu.edu.cn
[3] School of Water Conservancy Science & Engineering, Zhengzhou University, Zhengzhou 450001, China; chenxiaolong@gs.zzu.edu.cn
* Correspondence: huangshuowen@gs.zzu.edu.cn; Tel.: +86-136-1920-7750

Abstract: As an important and fundamental step in 3D reconstruction, point cloud registration aims to find rigid transformations that register two point sets. The major challenge in point cloud registration techniques is finding correct correspondences in the scenes that may contain many repetitive structures and noise. This paper is primarily concerned with improving registration using a priori semantic information in the search for correspondences. In particular, we present a new point cloud registration pipeline for large, outdoor scenes that takes advantage of semantic segmentation. Our method consisted of extracting semantic segments from point clouds using an efficient deep neural network, then detecting the key points of the point cloud and using a feature descriptor to get the initial correspondence set, and, finally, applying a Random Sample Consensus (RANSAC) strategy to estimate the transformations that align segments with the same labels. Instead of using all points to estimate a global alignment, our method aligned two point clouds using transformations calculated by each segment with the highest inlier ratio. We evaluated our method on the publicly available Whu-TLS registration data set. These experiments demonstrate how a priori semantic information improves registration in terms of precision and speed.

Keywords: terrestrial laser scanning; point cloud registration; deep learning; semantic segmentation; feature extraction

Citation: Li, J.; Huang, S.; Cui, H.; Ma, Y.; Chen, X. Automatic Point Cloud Registration for Large Outdoor Scenes Using a Priori Semantic Information. *Remote Sens.* **2021**, *13*, 3474. https://doi.org/10.3390/rs13173474

Academic Editor: Sander Oude Elberink

Received: 13 August 2021
Accepted: 31 August 2021
Published: 1 September 2021
Corrected: 18 May 2022

1. Introduction

Point clouds are important data structures that represent the three-dimensional, real world. Because point clouds have no topological structure and are also easy to store and transmit, they are widely used in 3D reconstruction, autonomous driving, intelligent robots, and many other applications [1–3]. However, the point cloud for an object is usually obtained using two or more scans from different reference frames because of the limitation of the geometric shape of the measured object and the scanning angle. Therefore, it is necessary to align all scans to obtain one point cloud for a complete scene in the common point cloud reference system, which is called point cloud registration.

Point cloud registration of large, outdoor scenes faces many challenges, such as the registration of symmetrical objects in large scenes, incomplete data, noise, artefacts caused by temporary or moving objects, and cross-source point clouds captured by different types of sensors. To address these challenges, many studies have proposed different methods for registering point clouds in large, outdoor scenes [4–7]. The classic registration pipeline usually extracts the feature elements of the point cloud, then uses the designed feature descriptor to establish the correspondence between the points through a nearest neighbor search, and, finally, the registration result is produced. This method is widely used in engineering and has achieved good results. However, it is computationally expensive, and the cost significantly increases as the number of points increases. Moreover, the handcrafted feature descriptor is usually based on low-level features such as curvature, normal vector, color, and reflection intensity. As a result, it may generate incorrect correspondence, which affects registration accuracy when a large-scene point cloud contains many repetitive and symmetrical structures. Recently, with the development of deep learning in point cloud semantic segmentation [8], a point cloud's high-level semantic information can be quickly obtained. Inspired by this development, we considered incorporating high-level semantic features into point cloud registration to solve the problems in the classic registration pipeline.

In this work, we present a complete point cloud registration pipeline for large, outdoor scenes using semantic segmentation. Our method used an efficient deep neural network to perform semantic segmentation on two point clouds that needed to be registered. For each segment of the source point cloud, we detected the key points of the point cloud and used a feature descriptor to generate the correspondence set in the respective segment of the target point cloud. Using the alignment of each segment, our algorithm extracted the transformation that can best align the initial point cloud. We performed tests on different scenes from Whu-TLS [9] to verify the effectiveness of the algorithm. The algorithm's parameter settings will also be discussed.

The rest of paper is organized as follows. Section 2 reviews some relevant works on point cloud registration and semantic segmentation. Section 3 describes the detail of the segmentation model and registration algorithm. Experiments and results of the proposed method are presented in Section 4, followed by the conclusions in Section 5.

2. Related Work

2.1. Point Cloud Registration

In engineering, a target is usually used for point cloud registration [10]. During registration, three or more targets are placed in the common area between scanning stations. After scanning the target area from different stations, the fixed targets are scanned accurately at each station and then used for registration. This method achieves a high registration accuracy in engineering applications, but it is time consuming and labor intensive and requires the scanning target to have an overlap and obvious geometric characteristics. To overcome these problems, many automatic registration methods that do not require manual intervention were proposed, and we will introduce them in the following content.

Iterative Closest Point (ICP) algorithm is a classic automatic algorithm used to solve the problem of point cloud registration [11]. It establishes the relationship between point pairs through the Euclidean distance, uses the least square method to minimize the objective loss function, and obtains the rotation parameters and translation matrix. Although the ICP algorithm is simple and practical, it is sensitive to noise and requires a good initial pose between the two point clouds; otherwise, it easily falls into a local optimal solution. To solve these problems, many scholars have improved the ICP algorithm, focusing primarily on the selection of matching points [12,13], the calculation of the initial value of the registration [14], and the design and optimization of the objective function [15,16], among other aspects.

Another type of point cloud registration method is based on feature element matching. In this method, the key points [17], line [18], surface [19], or other elements from the scanned point cloud are first extracted. Then, key elements are matched using their feature descriptions to calculate transformation parameters. This type of method does not need to provide an initial value, but it cannot achieve good results when the target point cloud data are missing.

Methods based on mathematical statistics, such as Normal Distribution Transform (NDT) [20] and Gaussian Mixture Models (GMM) [21], describe the point cloud's distribution characteristics by establishing a probability model so that after registration, the probability distribution between the two point clouds is the most similar. This registration method has strong robustness to noise and a lack of point cloud data, but the precise mathematical description of complex point clouds is too complicated and it also easily falls into the local optimal solution in the absence of an initial solution.

Deep learning methods for point cloud registration have recently developed rapidly. Some methods, such as 3DMatch [22], PPFNet [23], and Fully Convolutional Geometric Features (FCGF) [24], use deep neural networks to train point cloud feature descriptors. Because deep neural networks are capable of powerful feature extraction when trained on large data sets, they have excellent performance in point cloud registration tasks. There are also some end-to-end methods, such as PointNetLK [25], AlignNet-3D [26], and Deep Closest Point (DCP) [27]. These methods can directly output the transformation matrix for two point clouds that need to be registered even if they do not explicitly calculate the correspondence between the points. However, while these methods are simple and efficient, they do not consider the point cloud's local neighborhood information. Therefore, they are usually not used for large-scene registration tasks.

2.2. Point Cloud Semantic Segmentation

Efficiently obtaining accurate semantic information is an important part of our registration pipeline. Therefore, it is necessary to discuss some of the progress that has been made in the field of point cloud semantic segmentation. Traditionally, many point cloud semantic segmentation methods project 3D point clouds into 2D images and use more mature image segmentation technology to obtain results [7]. However, these methods are easily affected by image resolution and viewing angle selection. PointNet [28] is the first network to directly work on irregular point clouds, and it learns per-point features using shared multilayer perceptron (MLP) and global features using symmetrical pooling functions. It is a lightweight network and has been widely employed in many fields that utilize point clouds, but it lacks consideration of a point cloud's local information. To solve this problem, there are many methods for improving point cloud registration using point convolution or graphs. PointConv [29] uses a shared multilayer perceptron network to learn continuous weight functions for neighborhood points to define point convolution. KPConv [30] uses the kernel point as a reference and calculates the weight of these kernel points to update each point so that the point cloud's neighborhood information can be extracted. DGCNN [31] constructs a graph in the feature space, defined as EdgeConv, and dynamically updates that graph in each layer. However, using a convolution operation increases the memory and reduces the efficiency of the algorithm. To make semantic segmentation more efficient, RandLa-Net [32] uses random point sampling and a local feature aggregation module to reduce memory usage and computational complexity while maintaining high precision.

3. Methodology

As shown in Figure 1, our registration pipeline for large, outdoor scenes consisted primarily of these steps:

1. For the source point clouds $\boldsymbol{M}$ and the template point clouds $\boldsymbol{N}$, we first downsampled it by voxel filter, and voxel size was set to 0.05 m. Then we used statistical analysis filters to eliminate outliers in the point cloud. The number of neighborhood points analyzed for each point was set to 30, and the multiple of the standard deviation was set to 1. After that, a deep neural network was used to predict semantic labels for the input cloud.
2. The point cloud was divided into different subsets based on semantic labels. For the subsets that had the same labels, we extracted key points using intrinsic shape signatures (ISS). Additionally, for each key point, the hand-crafted feature Fast Point Feature Histograms (FPFH) [7] was calculated to get the initial correspondence set.
3. The random sample consensus (RANSAC) strategy was used to reject incorrect correspondence and calculate the transformation matrix between subsets in the source and template point clouds.

4. For each transformation matrix, we applied it to the source point cloud and chose the transformation matrix that had the highest inlier ratio as the final result.

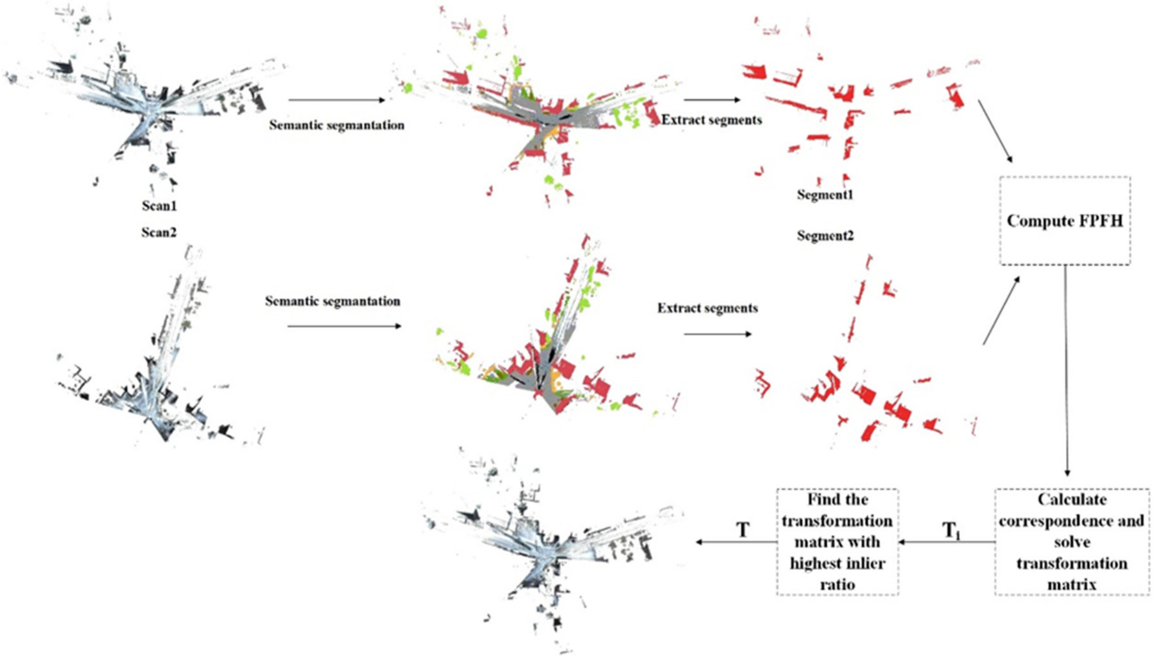


Figure 1. The over information flow of our method.

A detailed description of each step is as follows.

3.1. Semantic Information Extraction—Randla-Net

To quickly and effectively obtain semantic information for the next steps, we employed RandLA-Net, a lightweight and low-memory network structure that can directly process large-scale 3D point clouds. RandLA-Net first uses random sampling to process large-scale point clouds and then designs a local feature aggregation module to capture local structures. Its structure is similar to the classic point cloud segmentation encoder–decoder network structure. As shown in Figure 2, for encode, the input is a point cloud with a size of $\boldsymbol{N} \times \boldsymbol{d}_{\text{in}}$, where N is the number of points and $\boldsymbol{d}_{\text{in}}$ is the feature dimension of each input point, which may contain information such as coordinates, colors, normal, etc. In each encoding layer, the size of the point cloud is reduced by random sampling and per-point feature dimensions is increased by local feature aggregation module. For each layer in the decoder, the point feature set is upsampled through a nearest-neighbor interpolation. Next, the upsampled features are concatenated with the intermediate feature produced by encoding layers through skip connections, after which a shared MLP is applied to the concatenated feature. Finally, the semantic label of each point is obtained through shared fully connected layers.

Next, we will give a brief description of the key local feature aggregation module (LFA) in the network, including local spatial encoding, attentive pooling and the dilated residual block.

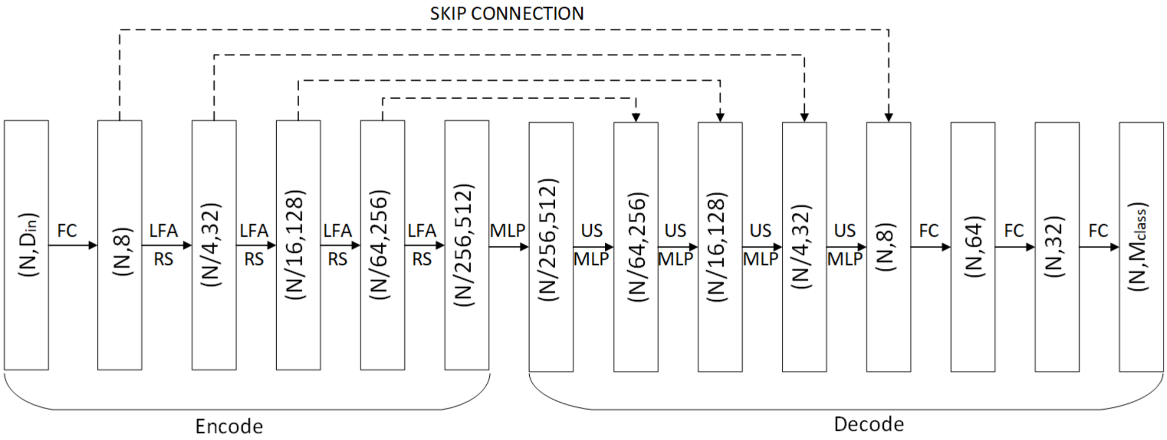


Figure 2. The structure of RandLA-Net.

1. Local spatial encoding: First, for each point, the nearest neighbor search algorithm is used to find the nearest neighborhood points in Euclidean space. Then, the neighborhood points are encoded by concatenating the three-dimensional coordinates of the center point, the three-dimensional coordinates of the neighboring point's relative coordinates, and the Euclidean distance, calculated as follows:

$$\mathbf{r}_i^k = MLP\left(p_i \oplus p_i^k \oplus \left(p_i - p_i^k\right) \oplus \|p_i - p_i^k\|\right) \tag{1}$$

$$\hat{\mathbf{f}}_i^k = \mathrm{f}_i^k \oplus \mathbf{r}_i^k \tag{2}$$

where $\oplus$ is the concatenation operation, $\mathbf{r}_i^k$ represents the features after aggregation, and $\hat{\mathbf{f}}_i^k$ represents new features in the neighborhood after concatenation.

2. Attentive pooling: This module is used to aggregate feature sets of neighborhood points. Unlike traditional algorithms, which usually use pooling to achieve hard integration of the feature set of neighborhood points, attentive pooling applies an attention mechanism to automatically learn and aggregate useful information in the feature set. It is defined as follows:

$$\widetilde{\mathbf{f}}_i = \sum_{k=1}^{K}\left(\hat{\mathbf{f}}_i^k \cdot \mathbf{s}_i^k\right) \tag{3}$$

where $\hat{\mathbf{f}}_i^k$ is features of each point in the neighborhood and $\mathbf{s}_i^k$ is the attention score learned using shared MLP.

3. Residual block: In simple terms, this module connects skip connections with multiple local spatial encoding and attentive pooling to form a dilated residual block so that the network can obtain a larger receptive field when the point cloud is continuously downsampled.

We adopted the network's original architecture for training and testing. When the network finished training on the outdoor data set, NPM3D, we used that trained network to obtain predicted labels for the Whu-TLS data set and used them to test our methods.

3.2. *Point Cloud Registration with Semantic Information*

Given two sets of points, $\boldsymbol{M}$ and $\boldsymbol{N}$ in arbitrary initial positions, let $\boldsymbol{M} = \{\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_{k_1}\}$ and let $\boldsymbol{N} = \{\boldsymbol{N}_1, \boldsymbol{N}_2, \ldots, \boldsymbol{N}_{k_1}\}$ be the semantic segments obtained after segmenting $\boldsymbol{M}$ and $\boldsymbol{N}$, respectively. For each $\boldsymbol{M}_i$ and $\boldsymbol{N}_i$ with the same labels, we used a voxel filter for downsampling and extracted ISS key points. Then, the FPFH feature descriptor was utilized to compute and match key points. Two points were matched if their FPFH features were one of the five nearest neighbors to each other. When the correspondences were established, we used a RANSAC-based strategy to calculate rotation and translation parameter, as follows:

1. A subset was randomly selected from the set of matched feature pairs.
2. Singular value decomposition (SVD) method was applied to calculate the rotation and translation matrix. First, we defined $\overline{\boldsymbol{M}_i}$ and $\overline{\boldsymbol{N}_i}$ as centroids of $\boldsymbol{M}_i$ and $\boldsymbol{N}_i$, which are two point sets to be registered. The cross-covariance matrix $\boldsymbol{H}$ is calculated by:

$$\boldsymbol{H} = \sum_{i=1}^{N} \left(\boldsymbol{M}_i - \overline{\boldsymbol{M}_i}\right)\left(\boldsymbol{N}_i - \overline{\boldsymbol{N}_i}\right)^T \tag{4}$$

Then, we used SVD to decompose $\boldsymbol{H}$ to $\boldsymbol{U}$, $\boldsymbol{V}$:

$$[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] = \mathrm{SVD}(\boldsymbol{H}) \tag{5}$$

Subsequently, we extracted the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ by Equations (6) and (7):

$$\boldsymbol{R} = \boldsymbol{V}\boldsymbol{U}^T \tag{6}$$

$$\boldsymbol{t} = -\boldsymbol{R} \cdot \overline{\boldsymbol{N}_i} + \overline{\boldsymbol{M}_i} \tag{7}$$

3. The matching results were verified to ensure they can make the most of the feature points' overlap. If the accuracy met the requirement or reached the maximum number of iterations, the registration result was output; otherwise, it returned to (1).

We obtained a transformation matrix set $\boldsymbol{T} = \left\{\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_{k_1}\right\}$ after the last step. To register two point clouds, we computed parameters by the following object function:

$$\hat{\boldsymbol{T}} = \mathrm{argmax}_{\boldsymbol{T}_i}(|g(\boldsymbol{T}_i(\boldsymbol{M}), \boldsymbol{N})|) \tag{8}$$

where $\boldsymbol{T}_i$ is the transformation matrix calculated by $\boldsymbol{M}_i$ and $\boldsymbol{N}_i$, which have the same semantic labels, and $g(.)$ is a function that calculates the inlier ratio after source point cloud transformation and it is defined as follows:

$$g(\boldsymbol{T}) = \frac{\sum_{i=1}^{D} \mathrm{I}(\|\boldsymbol{R}\boldsymbol{M} + \boldsymbol{t} - \boldsymbol{N}\| \leq \epsilon)}{S} \tag{9}$$

where I(.) is the indicator function, which equals 1 if the input is true or 0 otherwise, ϵ is the inlier threshold, and S is the number of points in the source point cloud.

To maximize the object function, we applied each $\boldsymbol{T}_i$ to the source point cloud and chose a $\boldsymbol{T}_i$ that aligned the largest numbers of inliers so that two point clouds could be registered with the highest probability.

4. Experiments and Results

4.1. Data Sets and Configuration

For semantic segmentation, we trained the semantic segmentation model on the NPM3D [33] data set, which is a large-scale, outdoor, point cloud segmentation benchmark. This data set was generated by a mobile laser system that accurately scanned two different cities in France (Paris and Lille). It was labelled into nine categories: ground, building, pole, bollard, trash can, barrier, pedestrian, car, and natural (vegetation). The specific distribution of each category is shown in Table 1. The data set was split into three stations for training, one station for validation, and four stations without ground truth for testing in our experiments. For the registration task, we used the Whu-TLS data set for the algorithm test. This data set was a large-scale point cloud registration benchmark that consisted of 11 different environments, such as subway station, mountain, forest, campus, etc. The Whu-TLS data set also provided ground-truth transformations to verify the accuracy of the registration results.

Table 1. NPM3D dataset category distribution table. The training and validation dataset are a 1% random sample.

Class	Train Set (1%)	Valid Set (1%)	Test Set
Ground	470,186	181,180	18,118,080
Building	240,665	89,278	6,741,262
Pole	4591	1905	155,999
Bollard	376	277	14,915
Trash can	2680	271	33,860
Barrier	16,603	23,723	310,146
Pedestrain	337	1396	26,554
Car	29,259	13,073	949,635
Natural	43,319	62,300	2,781,644
Total	808,006	373,403	29,132,095

We compared our method to Four Points Congruent Sets (4PCS) [34], Fast Global Registration (FGR) [35], and PointNetLK. For 4PCS, we set the delta to 0.8 and the number of samples to 1000. PointNetLK was implemented by the authors' release code with a pretrained model. We set a 5-m radius to estimate the normal and an 8-m radius to calculate FPFH for both FGR and our methods. The mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) between the ground truth transformation and the predicted transformation were used for accuracy evaluation (angular measurements are in units of degrees). All experiments were implemented on an Intel Xeon W-2145@ 3.70 GHz, 64 G RAM, GPU NVIDIA TITAN RTX 24G workstation.

4.2. RandLA-Net or KPConv

To choose a model to obtain predicted labels for each point in the registration data set more precisely and quickly, we compared the performance of RandLA-Net and KPConv, which are current popular semantic segmentation networks. Their performance on the NPM3D data set is presented in Table 2 (the data are from the official website of the benchmark data set). From Table 2, we can see that KPConv outperformed RandLA-Net in the mean Intersection-over-Union(mIoU) metric. However, RandLA-Net achieved better results for five categories, including building, bollard, barrier, car, and natural. In other categories, the accuracy of KPconv was better than RandLA-Net. We inferred from this that RandLA-Net may have a better semantic segmentation effect on large-object point cloud because the LFA module increased the receptive field of the network, while KPConv had a better effect on small-object point cloud.

Table 2. Intersection over union (IoU) metric of RandLA-Net and KPConv for the different classes of the Semantic3D dataset. mIoU refers to the mean of IoU of each class (%).

Methods	mIoU	Ground	Building	Pole	Bollard	Trash Can	Barrier	Pedestrian	Car	Natural
RandLA-Net	78.5	99.5	97.0	71.0	86.7	50.5	65.5	49.1	95.3	91.7
KPConv	82.0	99.5	94.0	71.3	83.1	78.7	47.7	78.2	94.4	91.4

Table 3 presents the effectiveness of the two networks over two scans of campus scenes in the Whu-TLS registration data set. The runtime of RandLA-Net was greatly reduced when compared with KPConv in the downsampled point cloud. From Figure 3, we can see that RandLA-Net mostly correctly classified building, ground, and natural point clouds. But for KPConv, some building point clouds were misclassified as natural. Perhaps this was due to our downsampling of the original point cloud and the difference between the types of features in the campus scene and the training data set. Therefore, for the sake of accuracy and efficiency, we finally chose RandLA-Net as our semantic segmentation network.

Table 3. Runtime of the two networks in the point cloud to be registered.

Methods	Scan	Point Numbers	Time(s)
RandLA-Net	Station 1	79,137	4.22
	Station 2	136,443	7.04
KPConv	Station 1	79,137	94.11
	Station 2	136,443	125.52

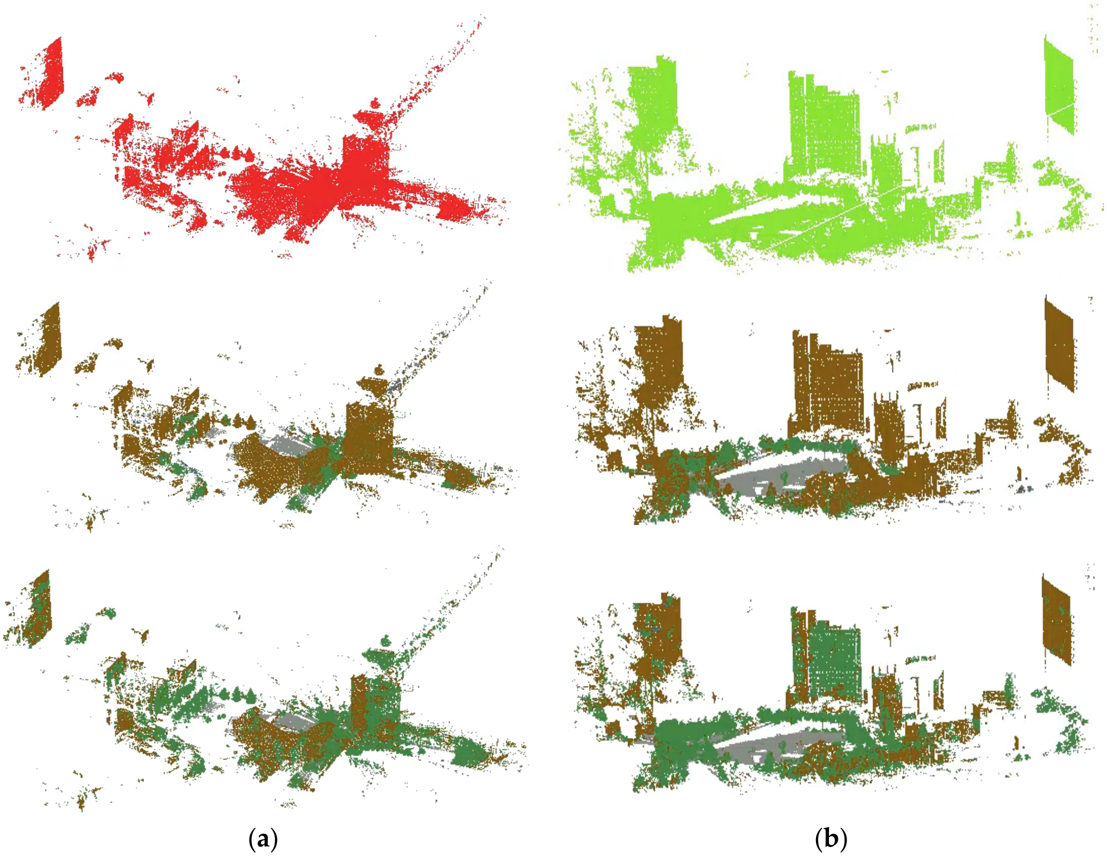


Figure 3. Semantic segmentation visualization of the source and target point clouds. (**a**) Source point cloud and its segmentation result; the former is RandLA-Net, and the latter is KPConv.; (**b**) Target point cloud and its segmentation result. Green represents natural points, gray represents ground points, and brown represents building points.

4.3. The Effect of Different Classes

As our method was based on semantic segmentation, we explored which class was suitable for registration in our method. Table 4 presents the registration results calculated for each class in our experiment. We chose three classes that had enough points to calculate FPFH and correspondences. It can be seen that the ground and natural points could not be registered well, and their predicted transformation was very different from the ground truth. We inferred that this was because the ground points had no obvious geometric structure and the natural points contained too many noise points. Therefore, few correct correspondences were generated and the accurate registration transformation was difficult to calculate. In contrast, building points contained obvious structural features and less noise, so the methods produced better results for these categories.

Table 4. Registration error and time for different classes.

Class	R/MSE	R/RMSE	R/MAE	t/MSE	t/RMSE	t/MAE	Point Number	Time(s)
Ground	12,750.873	112.919	65.915	2519.015	50.189	31.179	8579	0.29
Natural	10,132.264	100.659	71.129	338.403	18.395	16.367	23,278	4.19
Building	0.238	0.489	0.418	0.134	0.366	0.346	48,419	19.86

4.4. The Effect of Different Max Iterations

We compared our method, fusing semantic segmentation with RANSAC, with the traditional RANSAC method in regard to the maximum number of iterations. As shown in Table 5 and Figure 4, the errors descended as the maximum number of iterations increased in both methods. This is because, for the RANSAC method, as the number of iterations increases, more accurate and reliable correspondences can always be found to obtain better registration results, but it also requires more time consumption. Our method achieved good registration results when the maximum number of iterations was small. Additionally, the efficiency was also significantly improved. This improvement was due to the addition of a semantic segmentation module, which filtered out a large number of incorrect correspondences. For example, two points with different semantic labels could not be the correspondence. It made the search for correct correspondences between different stations faster and more accurate. Moreover, the traditional RANSAC method needed more iterations to get a better result because there were points with similar FPFH features in different classes, causing us to get some wrong matches.

Table 5. Registration error and time of different max iterations.

Iterations				Ours							RANSAC			
	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	MRSE(t)	MAE(t)	Time	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	MRSE(t)	MAE(t)	Time
1000	8.250	2.872	1.739	8.973	2.995	2.914	3.27	214.223	14.636	9.882	935.62	30.587	25.259	7.55
5000	0.199	0.446	0.374	1.511	1.229	1.056	6.31	0.878	0.936	0.885	2.598	1.611	1.092	18.66
10,000	0.238	0.489	0.418	0.134	0.366	0.346	25.11	1.774	1.332	0.814	1.027	1.014	0.698	62.49
20,000	0.238	0.489	0.418	0.134	0.366	0.346	50.23	1.770	1.330	0.813	1.022	1.011	0.697	125.31

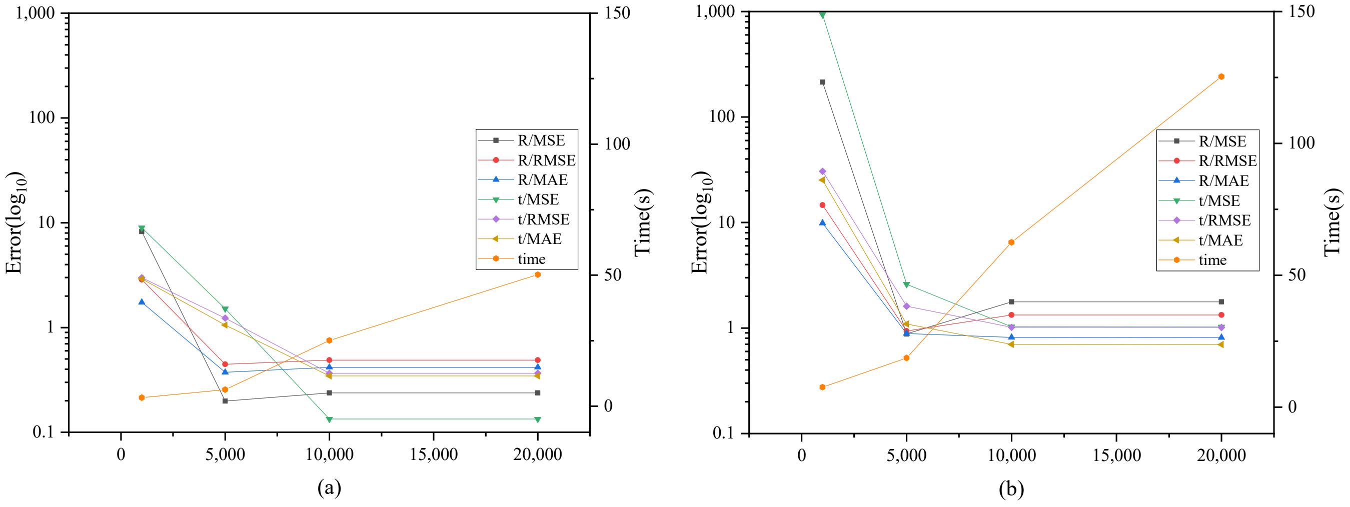


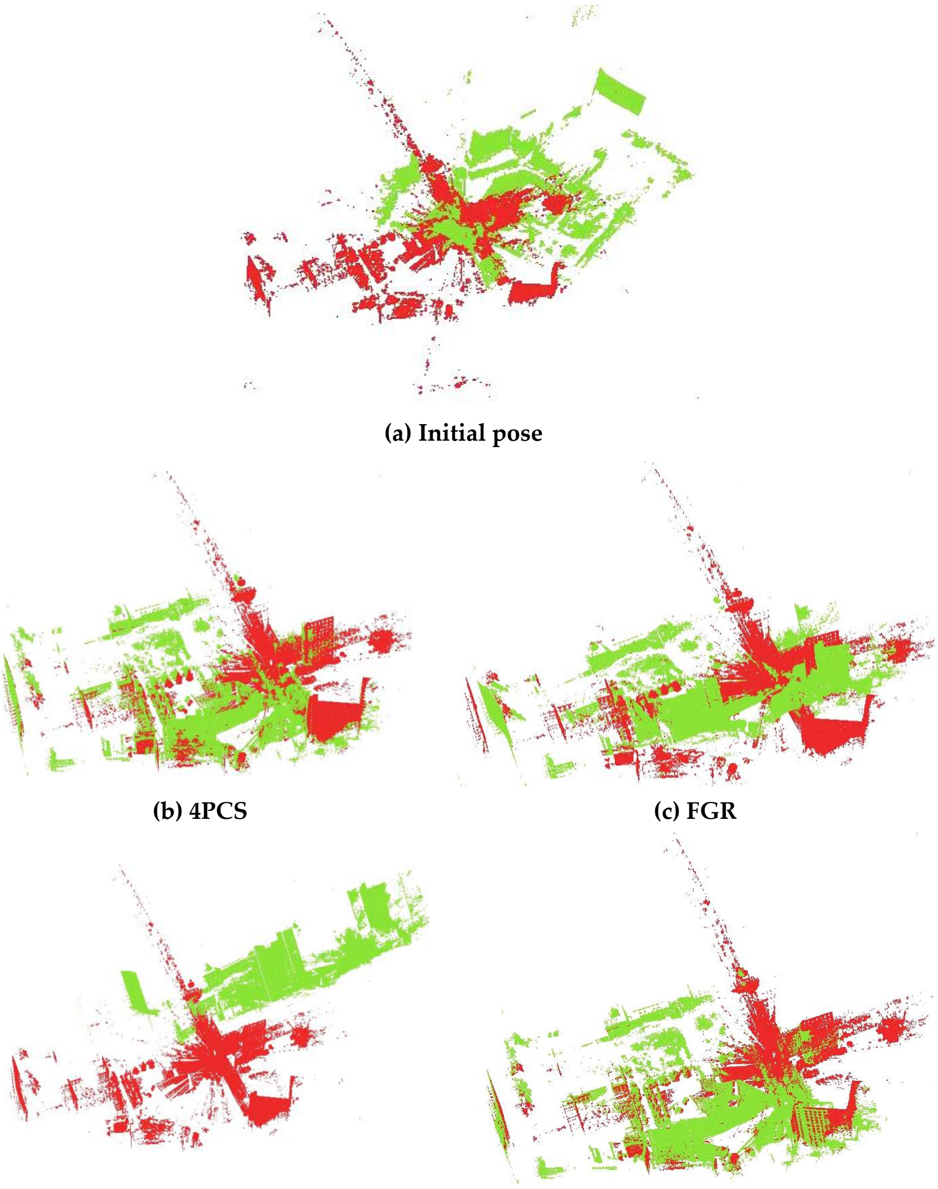
Figure 4. Each line shows registration error and time cost with respect to the maximum number of iterations, (**a**) Our method, (**b**) Traditional RANSAC method.

4.5. Comparison with Different Methods

We compared our method with 4PCS, FGR, and PointNetLK for two point clouds' registration in different scenes. Table 6 quantitatively shows the registration error and total time for different methods when registering campus scenes, which contain many artefacts. And Figure 5 shows the registration result of each algorithm for campus scenes. It can be seen that (1) 4PCS took the most time and obtained a good registration result, but it sometimes exhibited a poor registration effect or direct deviation in our experiment because it randomly sampled the points every time. (2) FGR roughly aligned two point clouds with initial poses that were far away but still needed to be further refined. (3) PointNetLK is an end-to-end network in deep learning with extremely high registration efficiency for small objects, but it is not directly applicable to point cloud registration in large scenes with complex and asymmetric structures. (4) Our method produced better results for point cloud alignment in terms of accuracy and efficiency, although the two point clouds had artefact noise points. The a priori semantic information was used to avoid incorrect classes from affecting the correspondence search.

Table 6. Registration error and time of different methods for campus scenes.

Methods	R/MSE	R/RMSE	R/MAE	t/MSE	t/RMSE	t/MAE	Time(s)
4PCS	1.039	1.019	0.887	8.019	2.832	2.373	58.59
FGR	116.181	10.778	10.621	412.365	20.307	16.199	49.09
PointNetLK	8857.101	94.112	64.473	995.206	99.575	87.587	5.86
OURS	0.238	0.489	0.418	0.134	0.366	0.346	39.11



(a) Initial pose

(b) 4PCS

(c) FGR

(d) PointNetLK

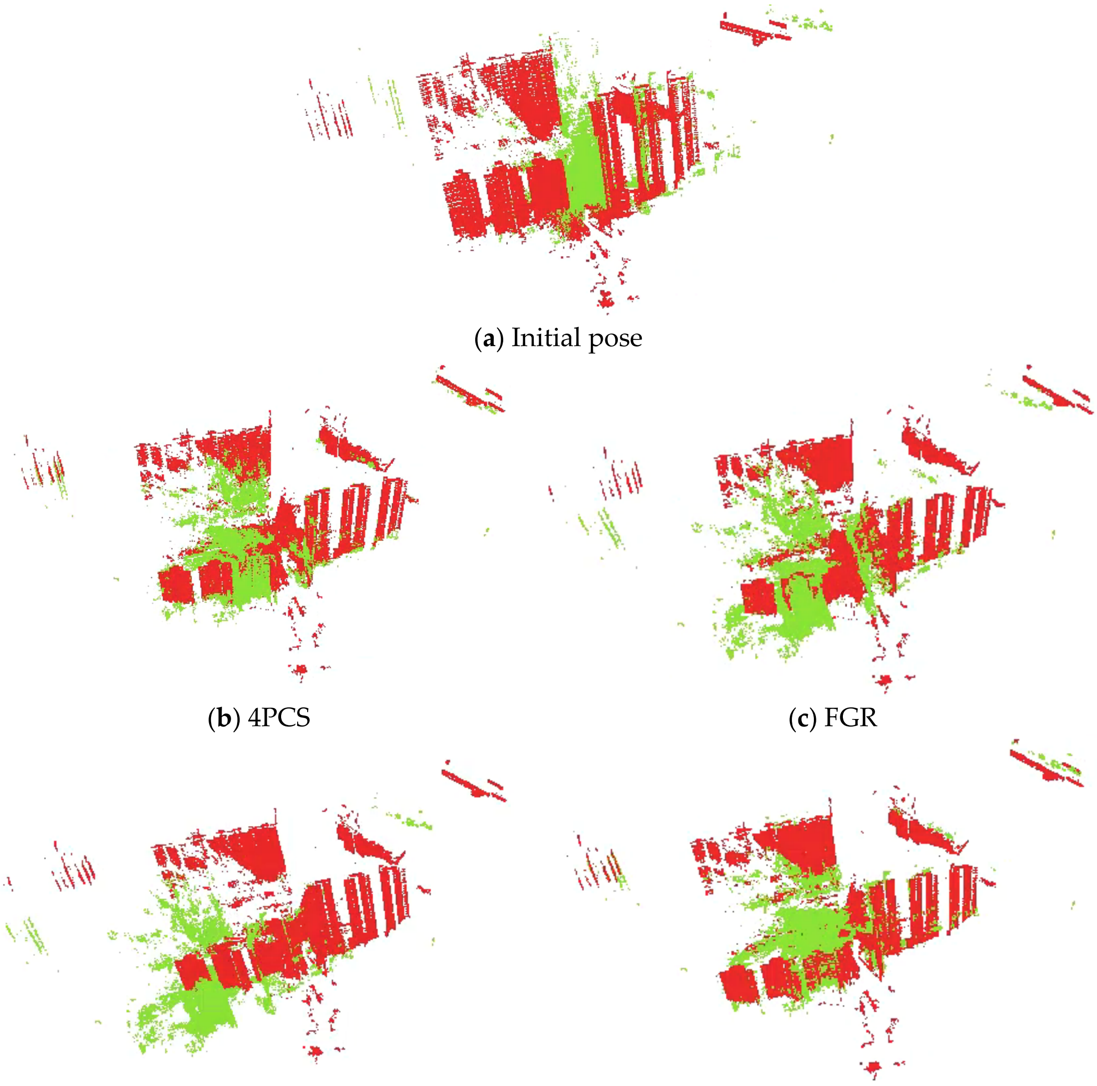
(e) OURS

Figure 5. Registration results of each algorithm for campus scenes.

The residence scene in the Whu-TLS data set contained many repetitive structures and homogeneous architectural layouts. We compared the accuracy and efficiency of the four methods in this scene. It can be seen from Table 7 and Figure 6 that 4PCS and our method obtained worse results than for campus scenes because of the ambiguity caused by repetitive structures. As the model was trained on a simulation data set whose transformation was set manually (the source point cloud and the target point cloud were symmetrical), PointNetLK obtained a better result than the campus scene.

Table 7. Registration error and time of different methods for residence scenes.

Methods	R/MSE	R/RMSE	R/MAE	t/MSE	t/RMSE	t/MAE	Time(s)
4PCS	3.895	1.973	1.351	84.742	9.205	6.881	7.51
FGR	53.082	7.285	6.841	54.676	7.394	6.863	82.52
PointNetLK	23.627	4.861	4.108	118.068	10.866	9.116	8.12
OURS	2.878	1.696	1.197	3.418	1.849	1.735	80.23


(**a**) Initial pose

(**b**) 4PCS

(**c**) FGR

(**d**) PointNetLK

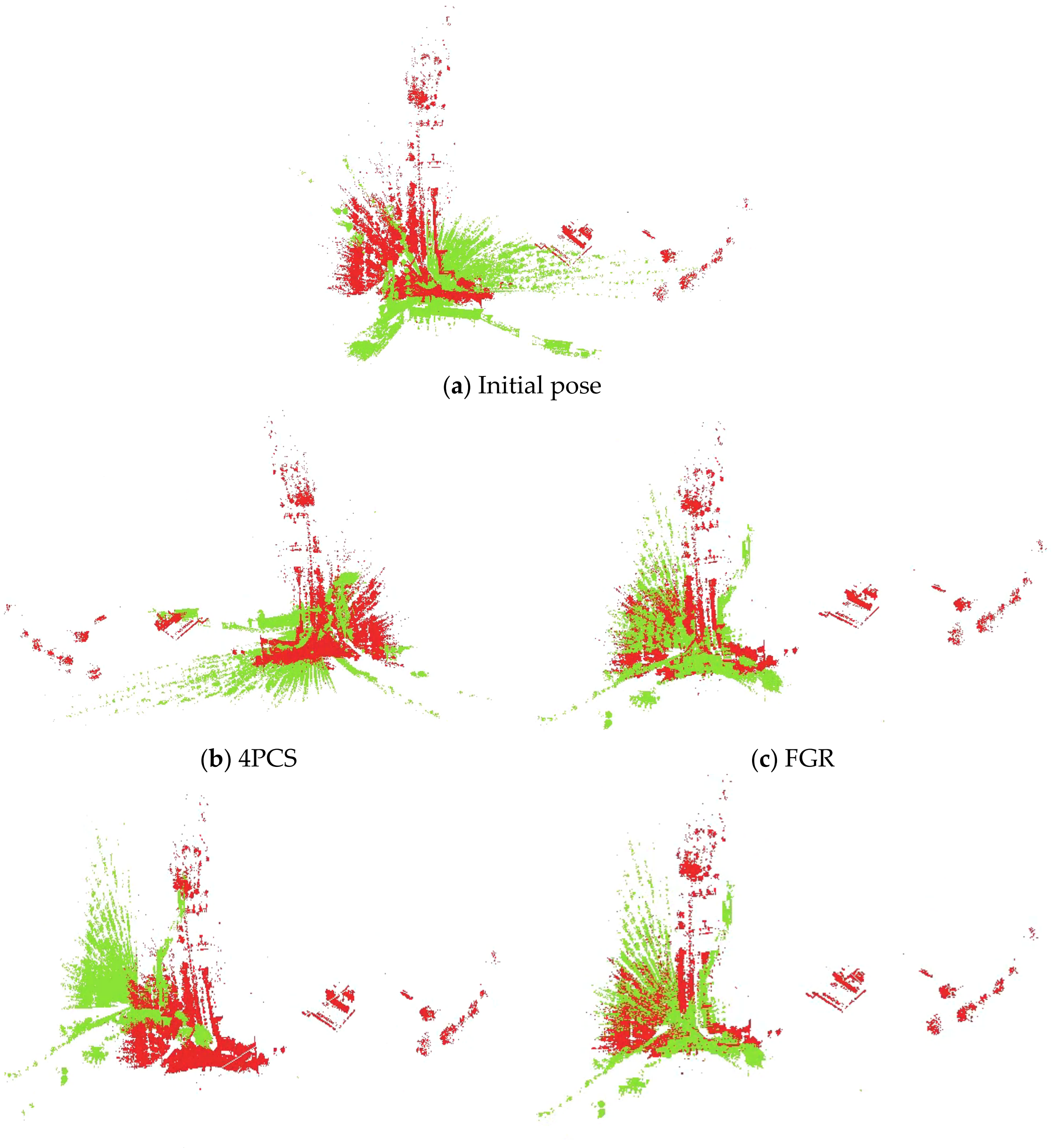
(**e**) OURS

Figure 6. Registration results of each algorithm for residence scenes.

Table 8 and Figure 7 show that 4PCS and PointNetLK both failed to register due to noise and semi-environments in the park scene. Although FGR and our method both used FPFH to generate a set of correspondences, our method was more robust for large, outdoor scenes, which may contain noise, artefacts, and complex structures. The addition of semantic information can reduce the probability of incorrect correspondence and achieve a better registration result.

Table 8. Registration error and time of different methods for park scenes.

Methods	R/MSE	R/RMSE	R/MAE	t/MSE	t/RMSE	t/MAE	Time(s)
4PCS	10,978.821	104.779	71.407	548.138	23.412	19.646	5.28
FGR	16.617	4.076	3.749	2.286	1.512	1.349	130.63
PointNetLK	3364.173	58.001	36.161	522.466	22.857	18.831	10.36
OURS	0.292	0.540	0.381	0.025	0.016	0.143	30.25



(**a**) Initial pose

(**b**) 4PCS

(**c**) FGR

(**d**) PointNetLK

(**e**) OURS

Figure 7. Registration results of each algorithm for park scenes.

5. Conclusions

In this work, we presented a new pipeline for large, outdoor scenes' point cloud registration. Unlike traditional RANSAC-based methods, we first performed semantic segmentation on the point cloud and calculated the geometric transformation for each segment of two point clouds that had the same semantic labels. We aimed to find the transformation that had the highest inlier ratio so that the point cloud could be registered to the greatest extent.

Our method was proposed to use semantic segmentation to improve the accuracy and efficiency of point cloud registration in large, outdoor scenes. Because the outdoor scene point cloud contains many noise points and the volume is huge, the traditional registration method based on low-level feature descriptors like FPFH usually takes a lot of time to obtain unsatisfactory results. Sometimes, points in different categories may have the same feature descriptor, which will generate wrong correspondence. This may reduce the accuracy of the registration or take more time to eliminate wrong matches. However, it can be solved by using a priori semantic information during the registration process.

We tested our method on the Whu-TLS registration data set. The results of the experiments showed that our method produced results with better quality and run time than the other methods for different scenes. However, it is worth mentioning that the registration results obtained from our methods were highly dependent on the semantic segmentation step. If the result of semantic segmentation is bad, it will directly affect our registration step because the data will be missing points of the same label. Additionally, more detailed and faster semantic segmentation may be able to further improve our method. Therefore, our future works will focus on the improvement of the semantic segmentation step.

Author Contributions: J.L. and S.H. proposed the original idea; S.H. and H.C. conceived and designed the experiments; Y.M. and X.C. performed the experiments, analyzed the data, and wrote the manuscript. All the authors discussed the results and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by research grants from China Postdoctoral Science Foundation (NO.2019M662534) and National natural science foundation (NO.42001405).

Data Availability Statement: Publicly available data sets were analyzed in this study. This data can be found here: https://npm3d.fr; http://3s.whu.edu.cn/ybs/en/benchmark, accessed on 12 August 2021.

Acknowledgments: The Whu-TLS data set was provided by the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. The authors would like to thank them for providing the point cloud data for experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705. [CrossRef]
2. Chen, S.; Liu, B.; Feng, C.; Vallespi-Gonzalez, C.; Wellington, C. 3D Point Cloud Processing and Learning for Autonomous Driving: Impacting Map Creation, Localization, and Perception. *IEEE Signal. Proc. Mag.* **2020**, *38*, 68–86. [CrossRef]
3. Bian, Y.; Liu, X.; Wang, M.; Liu, H.; Fang, S.; Yu, L. Quantification Method for the Uncertainty of Matching Point Distribution on 3D Reconstruction. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 187. [CrossRef]
4. Kuçak, R.A.; Erol, S.; Erol, B. An Experimental Study of a New Keypoint Matching Algorithm for Automatic Point Cloud Registration. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 204. [CrossRef]
5. Xiong, B.; Jiang, W.; Li, D.; Qi, M. Voxel Grid-Based Fast Registration of Terrestrial Point Cloud. *Remote Sens.* **2021**, *13*, 1905. [CrossRef]
6. Yang, B.; Dong, Z.; Liang, F.; Liu, Y. Automatic registration of large-scale urban scene point clouds based on semantic feature points. *ISPRS J. Photogramm.* **2016**, *113*, 43–58. [CrossRef]

7. Rusu, R.B.; Blodow, N.; Beetz, M. Fast Point Feature Histograms (FPFH) for 3D registration. In Proceedings of the IEEE International Conference on Robotics & Automation, Kobe, Japan, 12–17 May 2009; pp. 3212–3217. [CrossRef]
8. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [CrossRef] [PubMed]
9. Dong, Z.; Liang, F.; Yang, B.; Xu, Y.; Zang, Y.; Li, J.; Wang, Y.; Dai, W.; Fan, H.; Hyyppä, J.; et al. Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. *ISPRS J. Photogramm.* **2020**, *163*, 327–342. [CrossRef]
10. Cheng, L.; Tong, L.; Li, M.; Liu, Y. Semi-Automatic Registration of Airborne and Terrestrial Laser Scanning Data Using Building Corner Matching with Boundaries as Reliability Check. *Remote Sens.* **2013**, *5*, 6260–6283. [CrossRef]
11. Besl, P.J.; McKay, N.D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
12. Greenspan, M.; Yurick, M. Approximate k-d tree search for efficient ICP. In Proceedings of the International Conference on 3-d Digital Imaging & Modeling, Banff, AB, Canada, 27 October 2003; pp. 442–448. [CrossRef]
13. Weik, S. Registration of 3-D partial surface models using luminance and depth information. In Proceedings of the International Conference on 3-d Digital Imaging & Modeling, Ottawa, ON, Canada, 12–15 May 1997; pp. 93–100. [CrossRef]
14. Campbell, R.J.; Flynn, P.J. A Survey Of Free-Form Object Representation and Recognition Techniques. *Comput. Vis. Image Underst.* **2001**, *81*, 166–210. [CrossRef]
15. Yang, J.; Li, H.; Jia, Y. Go-ICP: Solving 3D Registration Efficiently and Globally Optimally. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 1457–1464. [CrossRef]
16. Segal, A.; Hhnel, D.; Thrun, S. Generalized-ICP. In *Robotics: Science and Systems V*; University of Washington: Seattle, DC, USA, 2009. [CrossRef]
17. Rusu, R.B.; Blodow, N.; Marton, Z.C.; Beetz, M. Aligning Point Cloud Views using Persistent Feature Histograms. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3384–3391. [CrossRef]
18. Li, J.; Zhong, R.; Hu, Q.; Ai, M. Feature-Based Laser Scan Matching and Its Application for Indoor Mapping. *Sensors* **2016**, *16*, 1265. [CrossRef]
19. Serafin, J.; Olson, E.; Grisetti, G. Fast and robust 3D feature extraction from sparse point clouds. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems, Daejeon, South, 9–14 October 2016; pp. 4105–4112. [CrossRef]
20. Takeuchi, E.; Tsubouchi, T. A 3-D Scan Matching using Improved 3-D Normal Distributions Transform for Mobile Robotic Mapping. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 3068–3073. [CrossRef]
21. Jian, B.; Vemuri, B.C. Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1633–1645. [CrossRef] [PubMed]
22. Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; Funkhouser, T. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1802–1811. [CrossRef]
23. Deng, H.; Birdal, T.; Ilic, S. Ppfnet: Global context aware local features for robust 3d point matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 195–205. [CrossRef]
24. Choy, C.; Park, J.; Koltun, V. Fully convolutional geometric features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 8958–8966. [CrossRef]
25. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7156–7165. [CrossRef]
26. Groß, J.; Ošep, A.; Leibe, B. Alignnet-3d: Fast point cloud registration of partially observed objects. In Proceedings of the 2019 International Conference on 3D Vision, Quebec City, QC, Canada, 16–19 September 2019; pp. 623–632. [CrossRef]
27. Wang, Y.; Solomon, J.M. Deep closest point: Learning representations for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 3523–3532. [CrossRef]
28. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660. [CrossRef]
29. Wu, W.; Qi, Z.; Fuxin, L. Pointconv: Deep convolutional networks on 3d point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9621–9630. [CrossRef]
30. Thomas, H.; Qi, C.R.; Deschaud, J.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6411–6420. [CrossRef]
31. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph. (Tog)* **2019**, *38*, 1–12. [CrossRef]
32. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11105–11114. [CrossRef]

33. Roynard, X.; Deschaud, J.; Goulette, F. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Int. J. Robot. Res.* **2018**, *37*, 545–557. [CrossRef]
34. Aiger, D.; Mitra, N.J.; Cohen-Or, D. 4-Points Congruent Sets for Robust Pairwise Surface Registration. In Proceedings of the 35th International Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 11–15 August 2008; Volume 27. [CrossRef]
35. Zhou, Q.Y.; Park, J.; Koltun, V. Fast Global Registration. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 766–782. [CrossRef]