Efficient Pairwise 3D Registration of Urban Scenes Via Hybrid Structural Descriptors

Long Zhang, Jianwei Guo, Zhanglin Cheng, Jun Xiao, Xiaopeng Zhang

***Abstract*—Automatic registration of point clouds captured by terrestrial laser scanning (TLS) plays an important role in many fields including remote sensing (*e.g.,* transportation management, 3D reconstruction in large-scale urban areas and environment monitoring), computer vision, virtual reality and robotics, among others. However, noise, outliers, non-uniform point density and small overlaps are inevitable when collecting multiple views of data, which poses great challenges to 3D registration of point clouds. Since conventional registration methods aim to find point correspondences and estimate transformation parameters directly in the original point space, the traditional way to address these difficulties is to introduce many restrictions during the scanning process (*e.g.,* more scanning and careful selection of scanning positions), thus making the data acquisition more difficult. In this paper, we present a novel 3D registration framework that performs in a "middle-level structural space" and is capable of robustly and efficiently reconstructing urban, semi-urban and indoor scenes, despite disturbances introduced in the scanning process. The new structural space is constructed by extracting multiple types of middle-level geometric primitives (planes, spheres, cylinders, and cones) from the 3D point cloud. We design a robust method to find effective primitive combinations corresponding to the 6D poses of the raw point clouds and then construct hybrid-structure-based descriptors. By matching descriptors and computing rotation and translation parameters, successful registration is achieved. Note that the whole process of our method is performed in the structural space, which has the advantages of capturing geometric structures (the relationship between primitives) and semantic features (primitive types and parameters) in larger fields. Experiments show that our method achieves state-of-the-art performance in several point cloud registration benchmark datasets at different scales and even obtains good registration results for data without overlapping areas.**

I. Introduction

With the rapid development of 3D scanning technologies in recent decades, it has become more convenient for users to quickly obtain 3D point clouds representing individual objects, indoor scenes, or large-scale urban and semi-urban scenes. As it contains the most concise and accurate 3D information about the real world, 3D point clouds are widely used in many fields including remote sensing (*e.g.,* transportation management, 3D reconstruction in large-scale urban areas and environment monitoring), computer graphics, computer vision, virtual reality, and robotics, etc. However, limited to capturing reflections, an individual scan cannot cover the whole surface of the target scene. To obtain data that represent a complete scene, multi-view scans are required to obtain a collection of unoriented point clouds with full coverage. To unite the raw data, it is necessary to align unoriented point clouds to a common coordinate system to reconstruct the real scene. There has been a great deal of research on point cloud registration [1], [2], predominantly using the *Iterative Closest Point* (ICP) algorithm [3], [4] and its many variants [5]–[9].

Fig. 1. Point cloud registration of an urban scene from Semantic3D dataset [10] in middle-level structural space, where the target and source point clouds are depicted in blue and yellow, respectively.

Previous works on point cloud registration generally have two steps: they search for a set of corresponding point pairs between two point clouds and then estimate the transformation based on the point pairs. A corresponding point is usually found by searching for the closest point or point with a similar local feature. Since these two steps are performed in the original point space, the correctness and accuracy of the registration result are highly dependent on the quality of point clouds in regard to several factors, such as noise level, outlier ratio, non-uniform point density, data completeness degree, and the scale of the overlapping areas between the source and target point clouds. Generally, it is difficult, if not impossible, to find correct point correspondences for two point clouds that have small overlapping areas or no overlapping area. In addition, when the outlier ratio, non-uniform point density, and missing data degree are higher, the incorrectness in

Long Zhang, Jun Xiao are with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China. Jun Xiao is the corresponding author (Xiaojun@ucas.ac.cn).

Jianwei Guo and Xiaopeng Zhang are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China.

Zhanglin Cheng is with Shenzhen Key Laboratory of Visual Computing and Analytics (VisuCA), Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China.

the point distribution increases drastically and thus disturbs the accuracy of feature point extraction. To address these difficulties, some plane-based approaches are presented [11]–[14], but since they are all based on the assumption that the scene should contain sufficient plane primitives, the adaptation scenarios are still limited. Furthermore, although there may exist overlapping planes in point clouds without overlapping areas, due to each plane contains only 2 degrees of freedom, these methods need to match at least 3 pairs of appropriate planes to achieve registration, which is still difficult to achieve for data that has small overlapping areas or no overlapping areas. As a result, point cloud registration remains a challenging problem, especially when two point clouds have no overlapping areas.

In contrast with previous point cloud registration methods that perform registration in the original point space, we propose a new 3D registration framework that extracts features, determines correspondence and computes transformation in a constructed feature space, as shown in the pipeline of Fig. 2. We extended the commonly used middle-level primitive shape (plane) to include more curved surfaces (*e.g.,* spheres, cylinders, and cones) and then proposed a method of constructing a descriptor that has a bijective relationship with the 6D pose of the raw point cloud and captures the geometric and semantic structures. Since these common primitive shapes exist in most urban and semi-urban scenes, it is reasonable for us to assume that the point clouds contain these primitive shapes that represent important structures, *e.g.,* roofs, facades, domes, and columns, etc. Moreover, the usage of hybrid primitive shapes improves the registration robustness on point clouds that lack enough planes or contain many similar local structures. Last but not least, point clouds without overlapping areas may still have overlapping primitives and can be registered by matching the overlapping primitives. By matching the descriptors, the point cloud registration is cast as the alignment of primitive features using middle-level semantics (*e.g.,* unit normal vectors of all planes, distances from the origin of coordinates to the planes, cylinder axis, cylinder radius, cone axis, and sphere center).

To emphasize the difference from the original "point space", we describe our method as registration that works on the "middle-level structural space". Operating in the middle-level structural space enables our framework to overcome the existence of noise, outliers, large holes, and small overlaps between point clouds since primitive shapes can be robustly extracted using an efficient RANSAC algorithm [15]. It should be noted that although there is another kind of scene abstraction that represents high-level semantics (*e.g.,* object types and part types), it has weak generalization potential and is not applicable to registration tasks. Experiments show that our method achieves state-of-the-art registration performance in several benchmark datasets of different scales, including real-world scans of urban scenes (an example is shown in Fig. 1), and even obtains good registration results for scans that do not have overlapping area. (see Fig. 9).

In summary, the main contributions of this work include the following:

- A novel generalized hybrid structural descriptor composed of multiple 3D primitive shapes. This new descriptor has a bijective relationship with the 6D pose of the raw point cloud and is able to capture the geometric and semantic structures;
- A new estimation method of transformation between two descriptors based on their contained middle-level semantics;
- A new and powerful framework for registering raw point clouds in middle-level structural space using our specially designed descriptors, which allows registration with noise, outliers, small overlapping areas and no overlapping area.

II. Related work

Geometric registration methods can be roughly divided into two categories: coarse registration and fine registration. The former computes an initial alignment of two point clouds, while the latter aims to further improve the accuracy of the initial estimate. Fine registration methods are usually based on an iterative optimization strategy, *e.g.,* the (ICP) algorithm [3], [4] and its variants [5]–[8] improve the registration accuracy by iteratively estimating the closest corresponding point pairs and optimizing the minimum distance between them. Our method falls into the coarse registration category and aims at pairwise registration using local features, which is different from registering multiple point clouds simultaneously [16], [17] and global feature-based methods [18], [19]. Therefore, we present a brief overview of the various techniques related to coarse registration, as well as some important feature-correspondence searching strategies since they are crucial to registration.

A. Point feature-based registration

Algorithms in this category usually follow a same procedure. First, key points are detected in the point cloud by using key point detectors, such as intrinsic shape signature [20], 3D Harris [21] and local curvature maximum, etc. These key points express significant geometric features within the neighborhood of the point cloud. Next, a local shape descriptor is constructed to encode the geometric information around the key points [22]. Johnson and Andrew E proposed the Spin Image [23] to measure the similarity of the scene and the model, but it is sensitive to data resolution changes and noise. Rusu *et al.* [24] proposed a fast point feature histogram (FPFH) which has the characteristics of fast and strong discrimination ability. They also proposed an algorithm (called SAC-IA) for the online computation of FPFH features for real-time applications. Tombari *et al.* [25] proposed the signature of histograms of orientations (SHOT), which directly encodes surface normal vectors at different locations in space, and adds robustness of histogram statistics to the geometric distribution information. Although SHOT is relatively descriptive, it is still sensitive to point density changes. Guo *et al.* [26] presented the RoPS which is robust to noise, but its main drawback is

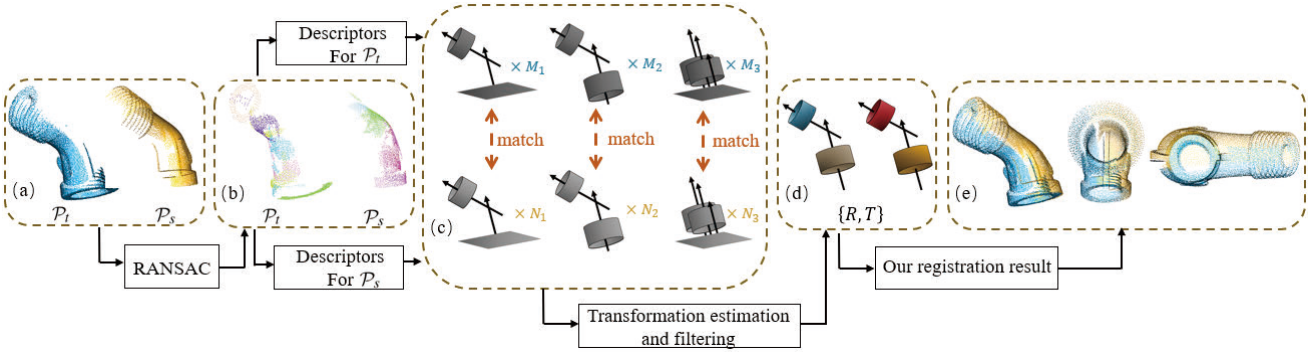

Fig. 2. Illustration of the pipeline of our registration framework using two view-opposite scans of a individual object. (a) input point clouds $\mathcal{P}_t$ and $\mathcal{P}_s$ which are scanned from opposite angles and almost have no overlapping area; (b) primitive extraction using efficient Ransac [15]; (c) descriptor construction and matching, where M and N are the number of descriptors for each category; (d) the optimal registration and the corresponding descriptors pair; (e) 3 perspectives of our registration result.

that the data with uneven distribution of points is poorly descriptive and the calculation is time-consuming. Frome *et al.* [27] extended the 2D shape context features to the 3D domain and proposed 3DSC. This descriptor for the first time shows the strong differentiation of the property of local depth. Using FPFH features, Zhou *et al.* [28] presented an optimization algorithm for fast global registration with partially overlapping 3D surfaces. Although these methods have achieved good results, they still require a large overlap between the point clouds.

B. Primitive feature-based registration

Compared with the point-based feature, the primitive-based feature covers a larger range, so it can perform a higher-level feature abstraction of the point cloud. There exist a vast number of approaches that focus on primitive shape extraction [29]. Schnabel *et al.* [15] proposed the efficient RANSAC algorithm, which automatically decomposes the point cloud into point sets associated with fitted basic primitive shapes. Che *et al.* [30] proposed a fast segmentation method for Terrestrial Laser Scanning (TLS) data in which the effects of edge-points on the normal estimation and region growing process were eliminated. To utilize the advantages of primitive shapes, primitive feature-based registration methods are proposed by which the robustness of feature recognition and feature matching is greatly improved. Habib *et al.* [31] and Al-Durgham *et al.* [32] proposed frameworks for point cloud registration using 3D straight lines. Yang and Zang [33] extended the use of straight lines to spatial curves. Based on a prior knowledge that man-made scenes usually contain many planes, researchers have proposed various feature construction and matching methods for the plane sets extracted from the point cloud. Dold *et al.* [34] used image information to improve registration methods based on planar matching. Xiao *et al.* [11] proposed an algorithm that uses planes for registration. After computing the plane area, a combination of heuristic search and pruning is used to find the optimal solution using the weighted least squares. Chuang *et al.* [35] proposed a multi-feature registration scheme that utilizes point, line, and plane features to achieve the registrations of multiple scans obtained from the same or different light detection and ranging (LiDAR) systems. Based on the deep convolutional neural network, Shi *et al.* [12] proposed a novel RGB-D block descriptor for detecting coplanar planes in SLAM reconstruction. Hattab *et al.* [36] proposed a RANSAC-based registration method that focuses on CAD point clouds using surface primitive group. Since the number of primitives selected in each round is fixed to 3, it cannot estimate the most accurate transformation by extracting effective primitive combinations. Xu *et al.* [13] proposed to use planar triples based on voxelization to construct descriptors and combine the RANSAC strategy for feature matching. On this basis, V4PCS [37] is then proposed, which further accelerates the efficiency of the algorithm by establishing voxel-based 4-planes congruent sets. Recently, Chen *et al.* [14] proposed the PLADE method, which aims to use plane-/line-based descriptor for establishing structure-level correspondences between point clouds. Plane-based features are more descriptive than local descriptors, and they are more robust to point cloud resolution changes and the existence of noise. However, the common drawback of these methods is that they are suitable for scenes that contain a large number of planar structures, thus they are difficult to register scenes dominated by curved surfaces.

C. Feature correspondence searching strategies

Aiming at increasing the efficiency and accuracy of pairwise registration, many widely-used efficient feature correspondence searching strategies are presented including geometric hashing [38] and RANSAC [39], [40]. In addition, some methods have also been proposed to speed up the establishment of key points correspondences. Aiger *et al.* [41] proposed to utilize the affine invariant ratio by constructing the 4-points congruent sets (4PCS), which increase the registration efficiency a lot. Mellado *et al.* [42] extended the original 4PCS algorithm and use an intelligent indexing strategy to achieve fast extraction of point pairs, reducing the computational complexity.

D. Learning-based registration

In recent years, some methods for constructing features based on deep neural networks have been proposed to replace hand-crafted descriptors. For the registration of indoor scenes, Zeng *et al.* [43] proposed the 3DMatch that establishes the correspondence between local 3D data by learning the descriptors of the local space blocks. This method can also be extended to different tasks and scales. Gojcic *et al.* [44] proposed a new workflow, called 3DSmoothNet, to match 3D point clouds based on a voxelized smoothed density value (SDV) representation. They use a Siamese deep learning architecture with fully convolutional layers to learn a compact local descriptor. 3DSmoothNet achieves state-of-the-art performance on 3DMatch benchmark. For single object registration, Aoki *et al.* [45] combined the classic LK image registration algorithm [46] and PointNet [47] into a single trainable neural network. This unified network has the advantages of good generalization ability for various shapes and high computational efficiency. Wang *et al.* [48] proposed an end-to-end method, referred to as Deep Closest Point (DCP), consisting of three parts: a point cloud embedding network, an attention-based module combined with a pointer generation layer to approximate combinatorial matching, and a differentiable singular value decomposition layer to extract the final rigid transformation. DCP achieves good results in the registration data set built on ModelNet40 [49]. These learning-based methods can achieve high accuracy and efficiency on their applicable data set. But they must be pre-trained on the appropriate dataset, and they are sensitive to noise and have poor generalization ability.

III. Overview

In this paper, we propose to construct robust hybrid-structure-based descriptors for efficiently registering point sets that have small overlap areas or no overlap area. The input to our method is a pair of point clouds named source $\mathcal{P}_s$ and target $\mathcal{P}_t$. Our goal is to estimate a 3×3 rotation matrix R and a 3×1 translation vector T that transform $\mathcal{P}_s$ to $\mathcal{P}_t$ to achieve rigid registration.

Fig. 2 illustrates the overall process of our algorithm using a simple case, which comprises four main steps. First, we extract the basic primitive shapes in $\mathcal{P}_s$ and $\mathcal{P}_t$ and construct the atomic structure sets $\{\mathcal{A}_k^s\}_{k=1}^4$ and $\{\mathcal{A}_k^t\}_{k=1}^4$, which abstract the main structure of the geometric primitives. Second, we build descriptors $\mathcal{D}_s$ and $\mathcal{D}_t$ by considering the geometric relationships between paired atomic structures. Next, we match the descriptors and obtain transformation hypotheses $\{R_m, T_m\}_{m=1}^N$ (N is the number of matched descriptors). Finally, we evaluate each estimated transformation, and a verification operation is performed to identify the transformation that achieves optimal registration, as shown in Fig. 2 (d) and (e).

IV. Methodology

A. Atomic structures

Primitive shape extraction. Taking into account the robustness to noise, efficiency and the ability to deal with a large number of points, we chose to use the efficient RANSAC algorithm [15] to decompose the entire area of each point cloud into subsets associated with the fitted primitive shapes (planes, cylinders, cones and spheres) and a set of unclaimed points. To further improve the robustness of primitive extractions, we adopt three simple strategies. First, the extractions are mainly focused on the major representative primitives that cover large areas with large point numbers. Second, the threshold δ on the maximum distance between a point and a primitive is set to be small (9 cm in urban scenes and 4 cm in indoor scenes by default). Third, the efficient RANSAC algorithm is implemented iteratively (1 round by default) in the unclaimed point set with δ increases by 4 cm in urban scenes and 2 cm in indoor scenes if the number of unclaimed points is higher than 70% of the total points. Since we extract major representative primitives robustly, our method can distinguish the surface of most structures in urban scenes or indoor scenes, *e.g.,* building facades, roofs, domes, cylinders, tabletops, floors, etc. Fig. 3 shows the extracted geometric primitives for the target point cloud in Fig. 1.

Fig. 3. Primitive shape extraction result for the target point cloud in Fig. 1, where each primitive is randomly color-encoded.

Atomic structure construction. After primitive shapes extraction, we discard the remaining unclaimed point set because it does not represent any of the four primitive shapes. We construct a set of *atomic structures* based on the extracted primitives, which is a mapping from point space to the middle-level structural space. The *atomic structures* abstract the main structure of the geometric primitives and will be used as basic elements to construct descriptors and enable them to correspond to unique coordinate systems in the raw point cloud space. As shown in Fig. 4 (b), we define 4 types of *atomic structures*:

- a plane $\mathcal{A}_1 = (\mathbf{p}, \mathbf{n})$ with the normal $\mathbf{n}$ and an arbitrary point $\mathbf{p}$;
- a sphere $\mathcal{A}_2 = (\mathbf{p}^*)$, where $\mathbf{p}^*$ represents the center of this sphere;
- a line $\mathcal{A}_3 = (l)$, which can be the intersection of two planes or the axis of a cylinder;

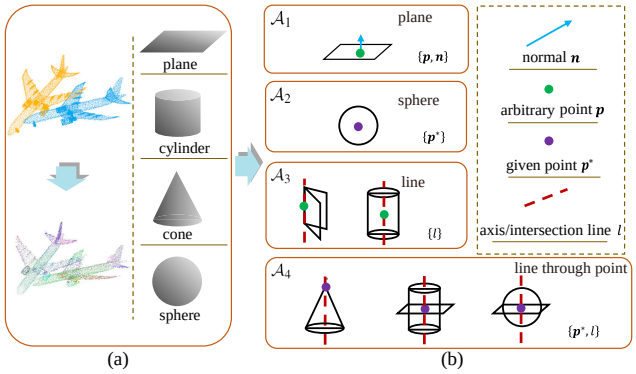

Fig. 4. Illustration of our defined *atomic structures*: (a) primitive shapes extraction; (b) 2D illustration of *atomic structures*. The auxiliary lines and points are explained in upper right.

- a line passes through a given point $\mathcal{A}_4 = (\mathbf{p}^*, l)$. There are 3 specific cases: the axis and apex of a cone, the axis of a cylinder and the intersection point with the plane perpendicular to it, and a spherical center and the normal of a plane intersecting it.

In the above definitions, $\mathbf{p}$ is an arbitrary point on the surface primitive (*e.g.,* a point on a plane) or on the element of one primitive (*e.g.,* a point on the axis of a cylinder). $\mathbf{p}^*$ is called a given point, *e.g.,* $\mathbf{p}^*$ can be a sphere center, a cone apex, or the intersection point between a line and a plane. When a pair of *atomic structures* are in a one-to-one correspondence, their given points should also match. The difference between $\mathcal{A}_3$ and $\mathcal{A}_4$ is the given point along the straight line, which allows recovery of two additional degree of freedom. This introduces a difference in the combination of *atomic structures*.

B. Hybrid-structure-based descriptors construction

For each view of the raw point clouds, we have obtained a candidate set of *atomic structures*. We now propose the construction of hybrid-structure-based descriptors by considering the combination of *atomic structures*. To ensure validity, a descriptor should have concise ingredients, be informative, and correspond to the raw point cloud coordinate system, which means it not only needs to abstract geometric structure information but also contains the minimum number of primitives to keep a bijective relationship with the 6D pose of the raw point cloud.

We first define three rules to avoid invalid combinations of the *atomic structures*:

- $\mathcal{R}1$: for *atomic structures* a_1 and a_2, if $a_1 \subseteq a_2$ or $a_2 \subseteq a_1$ (*e.g.,* a_1 is a line and a_2 is a plane that contains a_1), it is clear that this combination only provides information about one *atomic structure* which is not enough for registration. Therefore, this combination is not allowed;
- $\mathcal{R}2$: for *atomic structures* $a_1 \in \mathcal{A}_3$, $a_2 \in \mathcal{A}_4$ and $a_3 \in \mathcal{A}_1$, if $a_1 \perp a_3$ or $a_2 \perp a_3$, we can only recover four degrees of freedom in the vertical direction of plane a_3, but we cannot obtain the rotation angles around the axis that is parallel to a_3. We discard this combination;
- $\mathcal{R}3$: for *atomic structures* $a_1 \in \mathcal{A}_3$ and $a_2 \in \mathcal{A}_3$, a_1 is not allowed to be parallel to a_2. In this case, although we can recover two candidate rotations (one is in the same direction as the correct solution and the other is in the opposite direction), the translation between the two structures cannot be obtained.

Based on the above rules, we construct 4 kinds of descriptors $\mathcal{D} = \{\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \mathbf{D}_4\}$, where each descriptor $D \in \mathbf{D}_i$ is a multi-dimensional vector. As shown in Fig. 5, the descriptors are defined as follows:

1) $\mathbf{D}_1$ describes the geometric relationship between two unparallel lines $l_i \in \mathcal{A}_3$ and $l_j \in \mathcal{A}_3$, and each $D \in \mathbf{D}_1$ is defined as $D = (\mathcal{L}_1, min(r(l_i), r(l_j)), max(r(l_i), r(l_j)), angle(l_i, l_j), dist(l_i, l_j))$, where $\mathcal{L}$ is the label of the descriptor to distinguish the combination type of *atomic structures* that construct this descriptor, $r(l_i)$ and $r(l_j)$ refer to the radii of their *atomic structure*, $angle(l_i, l_j)$ refers to the acute angle between l_i and l_j, and $dist(*, *)$ denotes the shortest distance between two *atomic structures* (*e.g.,* line to line or line to point).
2) $\mathbf{D}_2$ describes the relationship between a line $l_i \in \mathcal{A}_3$ and a given point $\mathbf{p}_j^* \in \mathcal{A}_2$, and each $D \in \mathbf{D}_2$ is defined as $D = (\mathcal{L}_2, r(l_i), r(p_j), dist(l_i, \mathbf{p}_j^*))$.
3) $\mathbf{D}_3$ describes the relationship between a line $l_i \in \mathcal{A}_4$ and another line $l_j \in \mathcal{A}_3$, and each $D \in \mathbf{D}_3$ is defined as $D = (\mathcal{L}_3, min(r(l_i), r(l_j)), max(r(l_i), r(l_j)), r(\mathbf{p}_i^*), apx(\mathbf{p}_i^*), angle(l_i, l_j), dist(l_i, l_j))$, where $apx(\mathbf{p}_i^*)$ denotes the angle of the cone at apex $\mathbf{p}_i^*$.
4) $\mathbf{D}_4$ describes the relationship between a line $l_i \in \mathcal{A}_3$ and a plane $P_j \in \mathcal{A}_1$, and each $D \in \mathbf{D}_4$ is defined as $D = (\mathcal{L}_4, r(l_i), angle(l_i, P_j))$.

It should be mentioned that even for one descriptor type, the *atomic structures* can be quite different, leading to different vector elements. For example, in a descriptor $D \in \mathbf{D}_3$, l_j can represent the axis of a cone or a cylinder. If it is a cone, it is clear that the radius does not exist at the apex point $\mathbf{p}_i^*$. In this situation, we set $r(\mathbf{p}_i^*)$ as a relatively large number (*e.g.,* $10 * e^6$). If l_i represents a cylinder, the value of $apx(\mathbf{p}_i^*)$ is set to a relatively large number. As a result, the distance between the above two descriptors with different structures is large.

It is important to prove the bijective relationship between descriptor D and the raw point cloud coordinate system $C_{\mathcal{P}}$, which is the core of our registration algorithm. Obviously, each descriptor D is uniquely defined by $C_{\mathcal{P}}$. The key is to prove that $C_{\mathcal{P}}$ can be represented by D in a unique way. First, a unique local coordinate system can be recovered from each descriptor D following a predefined construction strategy. We extract two vectors and a given point $(\mathbf{v}_i, \mathbf{v}_j, \mathbf{p}^*)$ in each D, such as, for $D \in \mathbf{D}_1$, $\mathbf{v}_i$ is the unit vector of line l_i, $\mathbf{v}_j$ is the unit vector of line l_j, and $\mathbf{p}^*$ is the midpoint of the shortest distance between l_i and l_j, etc. Then, we obtain the $\mathbf{X}$-axis and $\mathbf{Z}$-axis that share the same direction with $\mathbf{v}_i$ and $\mathbf{v}_i \times \mathbf{v}_j$, respectively. Then,

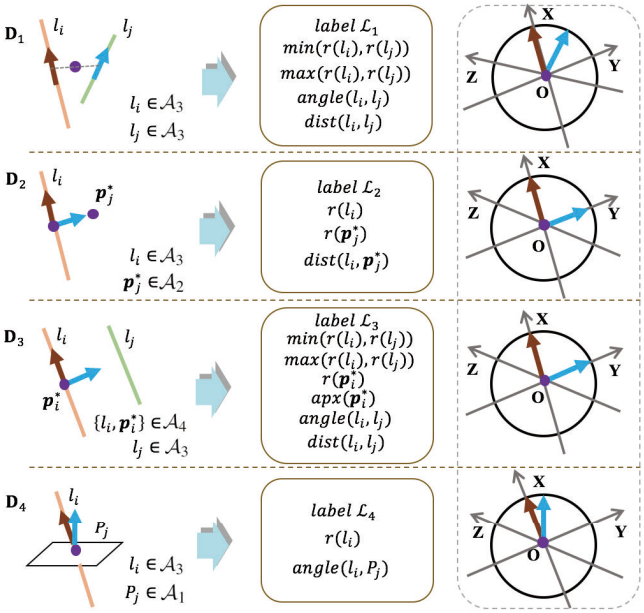

Fig. 5. Illustration of 4 kinds of descriptors. Left: *atomic structures* used for constructing descriptors. Middle: multi-dimensional vectors of the descriptors. Right: sectional view of the local coordinate systems defined by the descriptors.

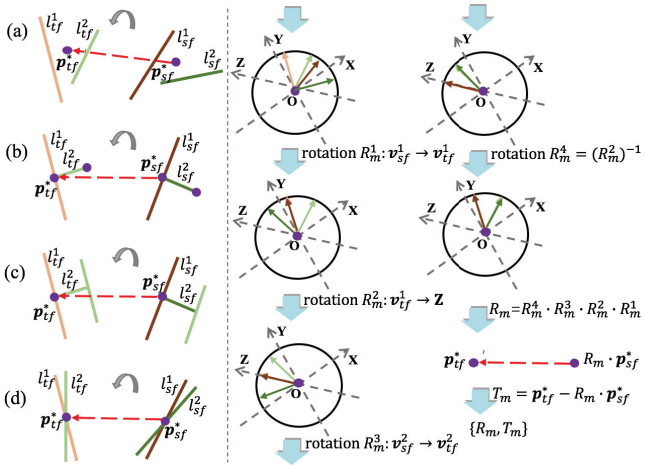

Fig. 6. Illustration of our registration process. Left: feature extraction from 4 kinds of matched descriptors. Right: registration process based on extracted features.

we obtain the $\mathbf{Y}$-axis direction as $\mathbf{v}_i \times \mathbf{v}_j \times \mathbf{v}_i$. Finally, we translate the origin of the local coordinate system to the given point $\mathbf{p}^*$ in D. According to the knowledge of coordinate transformation, $C_{\mathcal{P}}$ can be represented by the local coordinate system uniquely, which means that each descriptor D has an injective relationship with its raw point cloud coordinate system $C_{\mathcal{P}}$. Thus, the descriptor has a bijective relationship with its raw point cloud coordinate system ($D \sim C_{\mathcal{P}}$).

C. Registration

In the registration phase, we aim to examine the correspondence between descriptor D_i extracted in $\mathcal{P}_t$ and D_j extracted in $\mathcal{P}_s$. Since $D_i \sim C_{\mathcal{P}_t}$ and $D_j \sim C_{\mathcal{P}_s}$, by transforming D_j to coincide with D_i, the rigid transformation from $C_{\mathcal{P}_s}$ to $C_{\mathcal{P}_t}$ can be computed according to the invariant property of the equivalence relation.

Descriptors matching. We use the L_2 Euclidean norm as the similarity metric between two descriptors: $S(D_i, D_j) = ||D_i - D_j||_2$. Obviously, the closer the Euclidean distance is to 0, the higher the similarity of the matching pair. Note that if two descriptors have different labels, they cannot be matched by directly setting $S(D_i, D_j) = +\infty$. Although embedding the construction and matching of descriptors into the RANSAC framework can improve the matching efficiency, to ensure the registration result optimal, we still enumerate all the effective descriptors and evaluate their correspondences. First, we organize the descriptors in $\mathcal{P}_t$ into a hash table where the hash keys are their labels and the values encode the descriptors' multi-dimensional vectors. Then, we use a KD-tree to search the most similar descriptor $D_i^t \in \mathcal{D}_t$ for $D_j^s \in \mathcal{D}_s$ by looking that descriptors up in the hash table when D_j^s is constructed. Finally, we use an additional constraint to ensure the quality of descriptor correspondences. For a candidate matching pair (D_i^t, D_j^s), we conduct a reciprocity test, which means that when D_j^s is the most similar descriptor to D_i^t and vice versa, they can be regarded as successfully matched.

Transformation estimation. Fig. 6 (left) shows 4 kinds of matching configurations, which correspond to the descriptor types in Fig. 5. To compute the transformation between any pair of best-matched descriptors (D_i^t and D_j^s), we first need to recover two feature lines and a feature point from each descriptor, which are denoted as $(l_{tf}^1, l_{tf}^2, \mathbf{p}_{tf}^*)$ and $(l_{sf}^1, l_{sf}^2, \mathbf{p}_{sf}^*)$. In Fig. 6 (left), we also display how we compute the feature lines and feature points. Using the matching cases in Fig. 6 (a) as an example, the two feature lines are the original lines in the descriptor, and the feature point is the center point of the shortest vector between two lines. Other matching cases are handled similarly by making a perpendicular line and computing the intersection point.

After obtaining the feature lines and feature points, we compute the transformation between the coordinates recovered from matched descriptors by making $\{l_{tf}^1, l_{tf}^2, \mathbf{p}_{tf}^*\}$ and $\{l_{sf}^1, l_{sf}^2, \mathbf{p}_{sf}^*\}$ coincide as much as possible. Assuming that the current descriptor matching is correct, the final transformation calculation locates $\mathcal{P}_s$ and $\mathcal{P}_t$ in the same coordinate system as recovered from their descriptors, although their coordinate values are still based on the coordinate system of $\mathcal{P}_t$. On the right side of Fig. 6, we illustrate the registration process performed on a unit sphere, where we use unit vectors ($\mathbf{v}$) to represent the feature lines l with the same colors as these lines (in some combinations l corresponds to v in two opposite directions). Specifically, we first compute the rotation matrix R_m^1 that rotates $\mathbf{v}_{sf}^1$ to $\mathbf{v}_{tf}^1$, then apply rotation R_m^1 to $\mathbf{v}_{sf}^1$ and $\mathbf{v}_{sf}^2$, so now $\mathbf{v}_{sf}^1$ and $\mathbf{v}_{tf}^1$ coincide. Since the rotation applied by the rotation matrix is based on the Euler angle, the directly computed matrix of rotation $\mathbf{v}_{sf}^2$ to $\mathbf{v}_{tf}^2$ is likely to destroy the current coincidence. Therefore, we first make

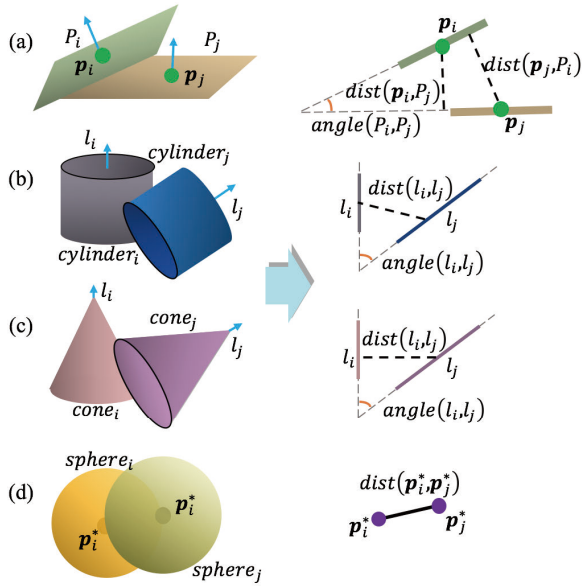

Fig. 7. Illustration of the evaluation between the registered primitives.

$\mathbf{v}_{tf}^1$ and $\mathbf{v}_{sf}^1$ coincide with the $\mathbf{Z}$ to avoid rotation around the coordinate axis. We obtain the rotation matrix as R_m^2 and we apply it to $\mathbf{v}_{tf}^2$ and $\mathbf{v}_{sf}^2$. We then compute the rotation matrix R_m^3 from $\mathbf{v}_{sf}^2$ to $\mathbf{v}_{tf}^2$ and finally obtain the rotation matrix R_m as follows:

$$R_m = R_m^4 \cdot R_m^3 \cdot R_m^2 \cdot R_m^1, \tag{1}$$

where R_m^4 is the inverse matrix of R_m^2. We give the translation vector T_m as:

$$T_m = \mathbf{p}_{tf}^* - R_m \cdot \mathbf{p}_{sf}^*. \tag{2}$$

D. Optimal registration identification

We estimate the transformation parameters between the target $\mathcal{P}_t$ and source $\mathcal{P}_s$ based on the similarity of the geometric structure. However, due to the existence of similar details in the scene, transformations derived from similar geometric structures do not always guarantee registration correctness. Additionally, among a set of correct registration results, we hope to identify the optimal one. Since our structural space contains planes and curved surfaces, the correctness of the coincidence of curved surfaces is difficult to misjudge. For example, it is easy to check whether a sphere and other primitives coincide. For each pair of primitives $(\mathcal{T}_i, \mathcal{T}_j)$, we evaluate their coincidence degree according to the following formula:

$$score(\mathcal{T}_i, \mathcal{T}_j) = \begin{cases} ang\left(\mathcal{T}_i, \mathcal{T}_j\right) + 0.5 \cdot \left(dist\left(\mathcal{T}_i, \mathcal{T}_j\right) + dist\left(\mathcal{T}_j, \mathcal{T}_i\right)\right), \\ \qquad\qquad if\ \mathcal{T}\ is\ plane; \\ ang\left(\mathcal{T}_i, \mathcal{T}_j\right) + 0.5 \cdot \left(dist\left(\mathcal{T}_i, \mathcal{T}_j\right) + \left|r\left(\mathcal{T}_i\right) - r\left(\mathcal{T}_j\right)\right|\right), \\ \qquad\qquad if\ \mathcal{T}\ is\ cylinder; \\ ang\left(\mathcal{T}_i, \mathcal{T}_j\right) + 0.5 \cdot \left(dist\left(\mathcal{T}_i, \mathcal{T}_j\right) + \left|apx\left(\mathcal{T}_i\right) - apx\left(\mathcal{T}_j\right)\right|\right), \\ \qquad\qquad if\ \mathcal{T}\ is\ cone; \\ dist\left(\mathcal{T}_i, \mathcal{T}_j\right) + \left|r\left(\mathcal{T}_i\right) - r\left(\mathcal{T}_j\right)\right|, \\ \qquad\qquad if\ \mathcal{T}\ is\ sphere, \end{cases} \tag{3}$$

where $ang\left(*,*\right)$ denotes the angle between two primitives (see Fig. 7), $dist\left(*,*\right)$ denotes the shortest distance between two primitives (point-to-plane distance in planes, axis-to-axis distance in cylinders and cones, and center-to-center distance in spheres), $r\left(*\right)$ denotes the radius of the primitive (cylinder radius or sphere radius), and $apx\left(*\right)$ refers to the angle of the cone. Only when $score(\mathcal{T}_i, \mathcal{T}_j)$ is less than the threshold ζ (which is 0.08 in planes, 0.09 in cylinders, 0.09 in cones, and 0.1 in spheres), $\mathcal{T}_i$ and $\mathcal{T}_j$ are considered coincident. For each pair of primitives verified to be coincident, we sum their scores to obtain S_m and count the number of all coincidence pairs as N_m. We give the evaluation of the transformation $\mathbb{S}_m$ as follows:

$$\mathbb{S}_m = \alpha \cdot N_m + \frac{1}{S_m}, \tag{4}$$

where we set α to be large enough to ensure that N_m becomes the most important criterion for measuring registration. By computing $\mathbb{S}_m$ for each transformation $\{R_m, T_m\}$, the transformation with the largest $\mathbb{S}_m$ can be considered the optimal transformation, which not only obtains the most coincident primitives but also makes the coincidence the most compact. Since we consider both primitive descriptors (in the previous stage) and registration quality to evaluate each transformation, even if primitives with similar descriptors are wrongly matched, other primitives from two point clouds can not find matches or have very small coincidence degrees, which makes the registration result is of low quality and can be easily rejected.

V. Experimental Results

A. Experimental Setup

In this section, our approach is first tested on several point clouds of large-scale urban/semi-urban scenes. Then we provide experiments in multiple registration scenarios including point clouds of indoor scenes and single objects to demonstrate that our method outperforms state-of-the-art registration approaches. Our algorithm is implemented in C++ and relies on the CGAL Library [1] to detect primitive shapes from the input point clouds. The shown results are obtained on a desktop computer equipped with an Intel Core i7-3770 Processor with 3.4 GHz and 16GB RAM.

Datasets. We select many pairs of point clouds for our performance evaluation and comparisons from six public-domain datasets. We first test our algorithm on the Robotic 3D scan repository [50], Semantic3D dataset [10] and Whu-TLS-dataset [16], both of which contain a lot of outdoor scenes with planar and curved surfaces. Then, we compare to a hierarchical method HMMR [16] on their presented Whu-TLS-dataset. Besides, we compare to the plane-based descriptor on RESSO [14] which contains real-world scans of indoor and urban scenes with a small overlap. We also compare with previous methods on data without overlapping areas. Finally, to test the generalization ability of our method on complex indoor scenes, we conduct

[1] www.cgal.org

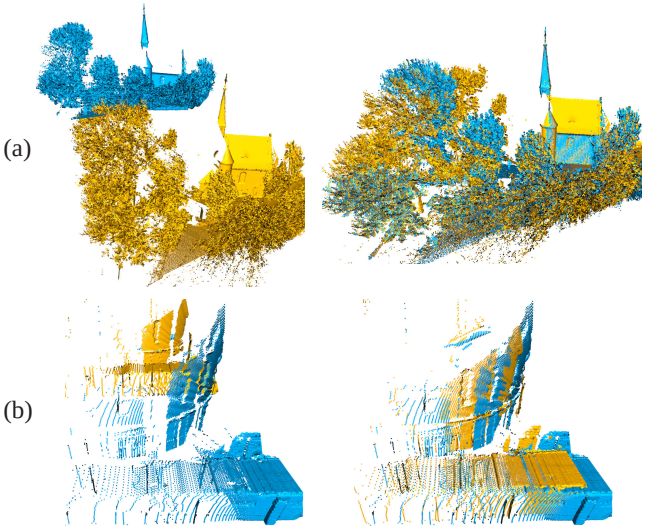

Fig. 8. Registration results of semi-urban and urban scenes. **Left:** Input two point clouds. **Right:** Our registration result.

another comparison on several difficult examples selected from an RGB-D indoor dataset [44]. In order to avoid the complicated parameter adjustment work that may be caused, we re-scale the coordinate of every two-frame point cloud into a bounding box with length size 2 and inverse map them in the end. To show the experimental results more clearly, we may have appropriately cropped the pictures.

Evaluation metric. To determine the registration accuracy, we measure the deviation between the predicted values and the ground truth values by calculating the *root mean squared error* (RMSE) and *mean absolute error* (MAE) for the translations and the Euler angles of the rotations. Obviously, the closer the RMSE and MAE are to 0, the more accurate the predicted values.

B. Evaluation

Evaluation on pairwise registrations. Fig. 1 and Fig. 8 display the registration results of the proposed method on three pairs of point clouds on urban scenes. Our method constructs descriptors based on the relationships between primitive shapes. The rigid transformation of $\mathcal{P}_s$ is recovered through descriptor matching. In Fig. 1, we show our registration results on a large scanned urban scene based on the matching of three planes, and in Fig. 8, we use different configurations of the matching primitive descriptors for registration: (a) two planes and a cylindrical surface and (b) one plane and a cylindrical surface. From these results, we observe that our approach successfully registered these pairs of point clouds.

We also conduct experiments on examples that do not have overlapping areas. As illustrated in Fig. 2, the input source and target point clouds are generated by scanning a pipe from two opposite views, and there is almost no overlapping area between them. Previous methods whose descriptors are based on key points, lines, or planes cannot resolve this registration problem due to the absence of corresponding coordinates and effective plane pairs. By contrast, our method builds descriptors and finds the correspondence of a cylindrical surface pair in a middle-level structural space. As a result, we can match the descriptors between primitives in each point cloud (as shown in Fig. 2) for registration. In addition, as shown in Fig. 9 (a), we use a pair of architectural point clouds derived from the Semantic3D dataset with little overlap as input and successfully register them, as shown in Fig. 9 (b). We have further trimmed the input point cloud of Fig. 9 (a), which not only reduces the number of potential primitives (points reduced by 50%) but also ensures that they do not have any overlapping parts in the real scene. The result of successful registration is shown in Fig. 9 (c). In Fig. 9 (d), we show that our method can perform accurate registration in a scene that contains only one plane and one cylindrical surface.

Evaluation on successive pairwise registrations. Although our method focuses on pairwise registration, we conduct another experiment to evaluate our overall performance for successive pairwise registrations. Specifically, among the urban scans of Whu-TLS-dataset [16], we select all the 10 scans of the "campus" scene whose ground truth dimension is (940.616 m $\times$ 978.801 m $\times$ 218.893 m). Then, the pairwise registrations perform in three sequences, which are "scan $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$", "scan $7 \rightarrow 6 \rightarrow 5 \rightarrow 4$" and "scan $10 \rightarrow 9 \rightarrow 8 \rightarrow 4$", respectively. By the end of successive registrations, all point clouds are aligned onto the reference point cloud of scan 4 and realize the registration of a complete "campus" scene. As shown in Fig. 10 (a) and (c), the 10 point clouds individually cover subareas of the complete scene and are correctly registered by our method. In addition, Fig. 10 (b) and Fig. 10 (d) show that the building boundaries and building facades are both aligned accurately, which further verifies the correctness of the successive registration results. Table I reports the rotation and translation errors of all registrations, where the related registration sequences are shown in the third column. These experimental results demonstrate that our proposed method performs well in registering successive point clouds, with the average rotation error (RMSE) and translation error (RMSE) are 0.1562° and 0.1689 m, both of which are quite small with respect to the dimensions of the real scene. There are mainly two reasons for the error accumulation can be limited: first, our method is able to register pairwise point clouds accurately, which generates little error; second, the orientation offset caused by rotation error generated in each pairwise registration is different, so is with the translation offset.

Evaluation on the feature observations. For urban, semi-urban and indoor scenes, since most surfaces that cover large areas are far from each other (usually more than 0.5 m in outdoor scenes and 0.2 m in indoor scenes), *e.g.,* the distance between two building facades or the distance between a desk and a floor, our method can easily distinguish the major representative primitives. However, when the positions and angles of similar primitives are too close, they may be hard to be separated. As shown in Fig. 11 (a), the facades of two buildings in the black box are nearly coplanar and are not separated. In such cases,

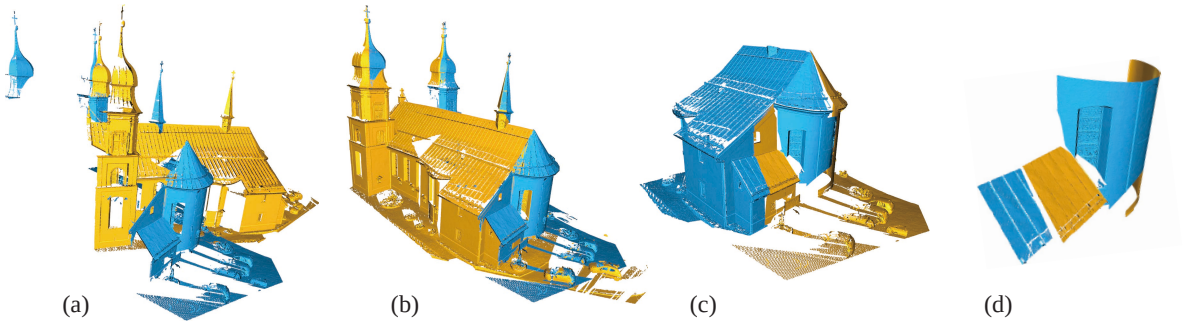

Fig. 9. Our algorithm takes as input a pair of point clouds with a very small overlap (a), which is particularly challenging for 3D registration. We propose to find correspondences between primitives and successfully register such data in a middle-level structural space (b). Our algorithm also works when data completeness is only 50% and does not contain any common parts of the real-scene (c). Our algorithm even achieves registration when the real-world scene contains only two primitives (d).

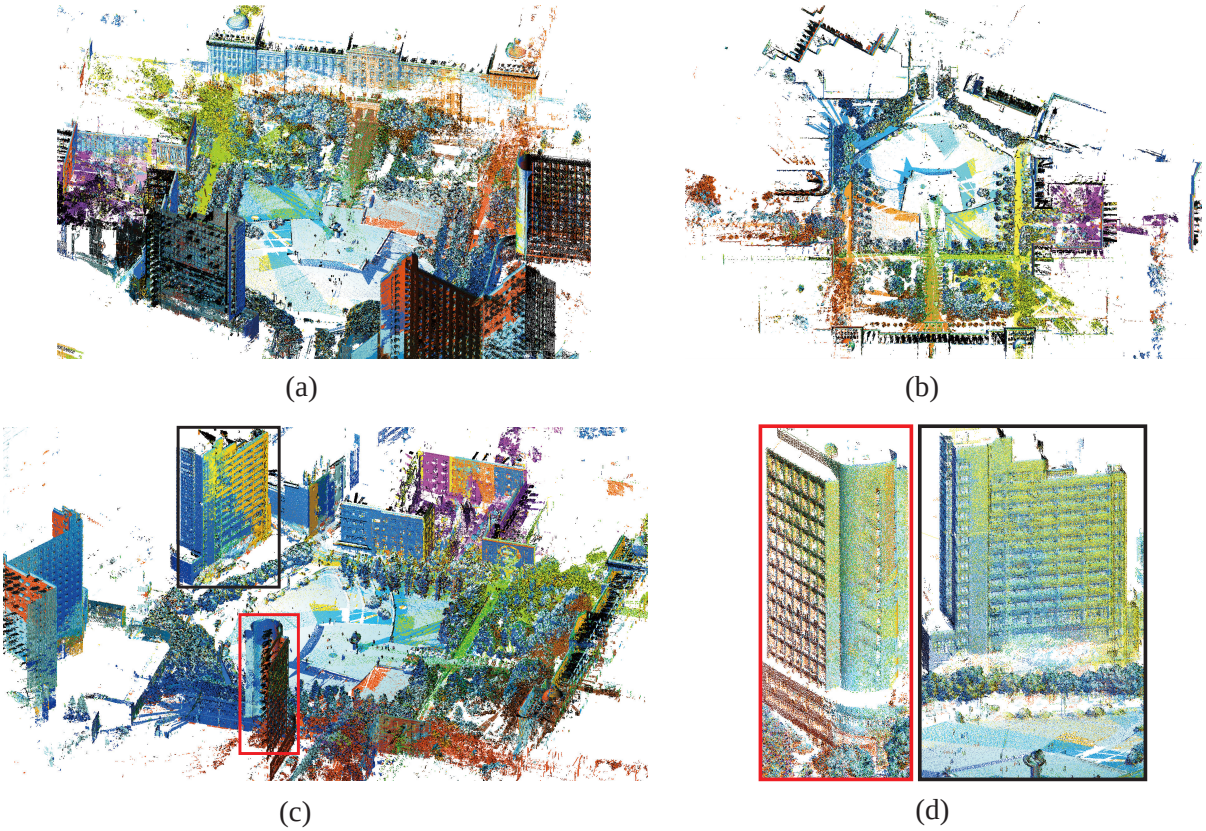

Fig. 10. Successive registration result of the 10 scans from the "campus" sub-set of Whu-TLS-dataset [16], where points from different point clouds are rendered with different colors. (a) and (c) are two side views; (b) is the plain view; (d) shows two zoom-in views of (c).

because our method uses the least number of primitives to represent each local structure, even if some of the extracted primitives are not accurate, we can still construct and match effective combinations composed of other accurate primitives. Therefore, as shown in Fig. 11 (b), our method registers point clouds by matching primitive combinations located in the orange boxes.

Evaluation on noises. In order to evaluate the impact of noises on registration performance, we added Gaussian noise increasingly to the point clouds from the "campus" subset of the Whu-TLS-dataset [16], where the standard deviation σ is set to 2.5, 5, 7.5, 10, 12.5, 15, 17.5 and 20 cm, respectively. As shown in Fig. 12 (a), although the number of extracted primitives in target and source point clouds fluctuates along with the increasing noise, we can still register point clouds by matching about 11 pairs of major representative primitives. Moreover, as shown in Fig. 12 (b), our method registers the eight pairs of noisy point clouds correctly, where the average rotation and translation errors are 0.166° and 0.2578 m. Comparing with the dimensions of the real scene (638.029 m $\times$ 833.059 m $\times$ 177.237 m) and the point spacing (0.1002 m), the errors are small and can satisfy the requirement of coarse registration. Fig. 12 (c) displays the registration result at noise level $\sigma = 20$ cm and locates the primitive combinations that are matched in orange boxes. These experimental results demonstrate

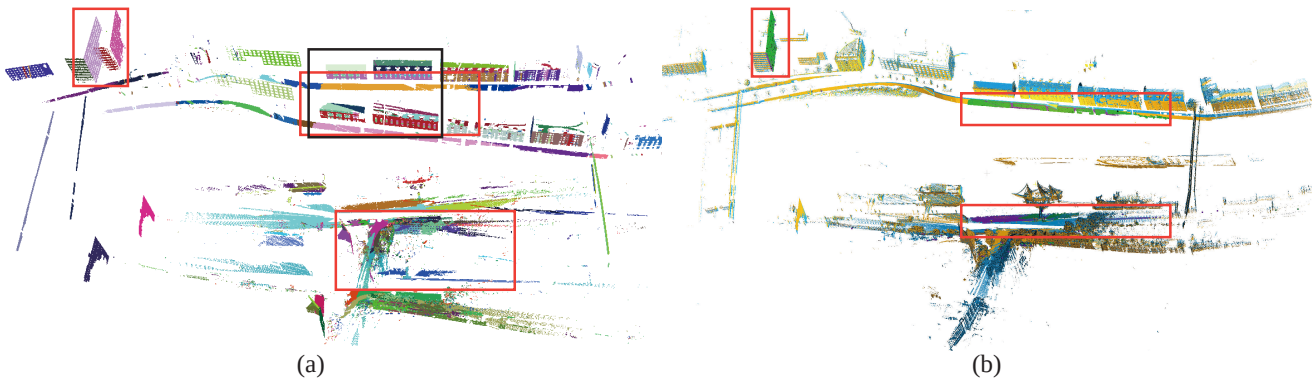

Fig. 11. Registration result on Robotic 3D scan repository [50]: (a) the primitive extraction result; (b) the matching result of three pairs of primitives.

TABLE I
QUANTITATIVE ANALYSIS OF THE SUCCESSIVE REGISTRATION RESULTS.

Data	Avg. point spacing(m)	Registration Sequence	Rotation err.(deg) RMSE	Rotation err.(deg) MAE	Translation err.(m) RMSE	Translation err.(m) MAE
Scan 1	0.1153	Scan 1-2	0.0782	0.0738	0.0701	0.0695
		Scan 1-2-3	0.0198	0.0181	0.1772	0.1402
		Scan 1-2-3-4	0.1228	0.1095	0.193	0.1762
Scan 2	0.1144	Scan 2-3	0.0721	0.0612	0.167	0.1491
		Scan 2-3-4	0.1502	0.1186	0.1932	0.1465
Scan 3	0.1165	Scan 3-4	0.1329	0.1129	0.023	0.0223
Scan 5	0.1075	Scan 5-4	0.0575	0.0433	0.056	0.0485
Scan 6	0.0824	Scan 6-5	0.1507	0.1304	0.2443	0.2133
		Scan 6-5-4	0.1964	0.1747	0.3006	0.2871
Scan 7	0.0597	Scan 7-6	0.3487	0.2887	0.067	0.0644
		Scan 7-6-5	0.3375	0.3281	0.2887	0.2381
		Scan 7-6-5-4	0.3331	0.3211	0.349	0.3428
Scan 8	0.0939	Scan 8-4	0.2051	0.1758	0.0831	0.062
Scan 9	0.1136	Scan 9-8	0.0303	0.0275	0.0662	0.0652
		Scan 9-8-4	0.1899	0.1356	0.3039	0.2393
Scan 10	0.1199	Scan10-9	0.0686	0.0452	0.0526	0.0457
		Scan 10-9-8	0.0604	0.0493	0.1274	0.1267
		Scan 10-9-8-4	0.2582	0.1688	0.2772	0.2316

that our method is robust to the noises and can register point clouds correctly, as long as the major representative primitives are not contaminated severely and a sufficient number of them can be extracted.

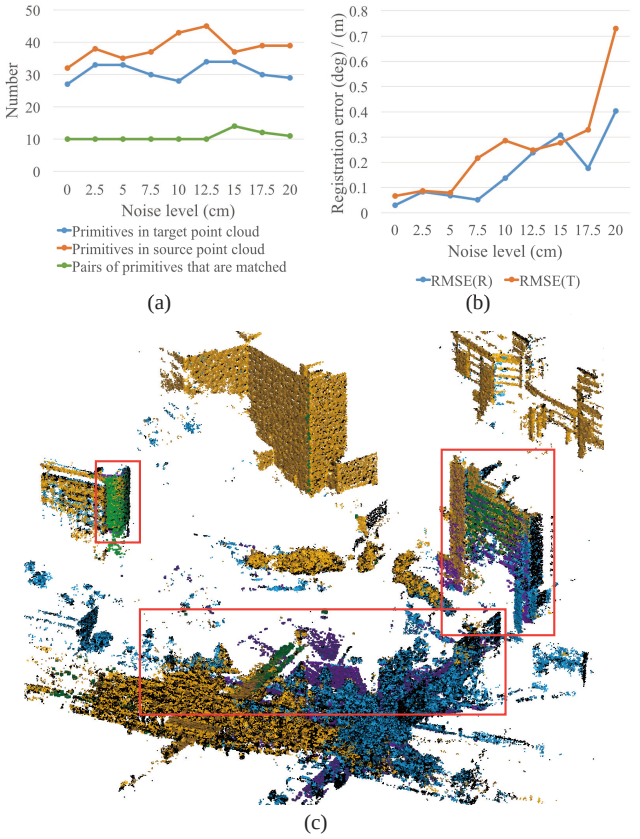

Fig. 12. Registration results on point clouds with increasing noise level: (a) the numbers of the extracted primitives and the primitives that are matched in a pair of point clouds obtained from the Whu-TLS-dataset [16]; (b) the rotation and translation error at each noise level; (c) the visualization of registration result at noise level $\sigma = 20$ cm.

C. Comparisons

We now compare our method against various registration competitors, including three classic approaches based on feature points (SAC-IA [24], Super4PCS [42], FGR [28]), a Plane-based registration method (PLADE [14]), and two recent deep learning approaches (3DSmoothnet [44], DCP [48]). These methods provide a plethora of comparison to other techniques and establish themselves as state-of-the-art methods. Besides, we use the same voxel size when computing the FPFH feature for SAC-IA [24] and FGR [28] approaches.

Comparison on point clouds of urban scenes. Fig. 13 visualizes the registration results of different methods on three pairs of real-scans in which the ground truth dimensions are 794.385 m $\times$ 808.98 m $\times$ 616.179 m, 880.754 m $\times$ 710.86 m $\times$ 318.738 m, and 195.506 m $\times$ 214.653 m $\times$ 61.9268 m, respectively. Table II gives a quantitative analysis of the registration results in Fig. 13. From the comparison, it can be seen that the 3 registration methods based on feature points have similar prediction accuracy for rotation. However, although we constructed the same FPFH features for SAC-IA and FGR, the translation accuracy of FGR on the scene in Fig. 13 (b) is significantly higher than SAC-IA. At the same time, Super4PCS shows obvious translation errors in all experimental cases due to incorrect

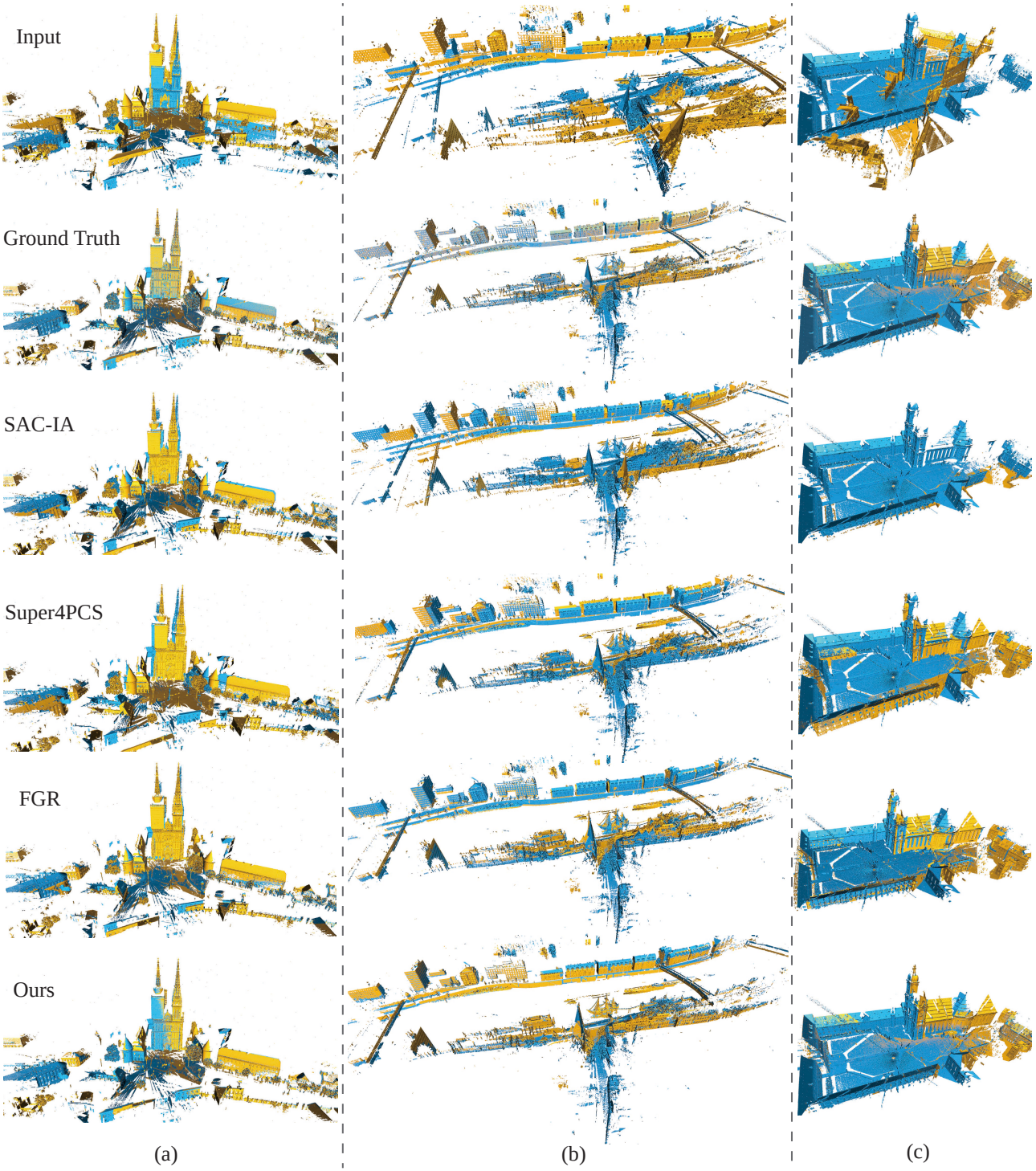

Fig. 13. Comparison with previous methods on three pairs of point clouds of large-scale urban scenes. The data in (a) and (b) are from Robotic 3D scan repository [50], while the data in (c) is from Semantic3D dataset [10].

matching of feature points. In urban scenes, there are many similar local features that are very close, *e.g.,* different doors or windows in the same building; therefore, even if the rotation error of Super4PCS is relatively small, its translation error may still be large. By contrast, our method can perceive a larger range of geometric structures, so it is not sensitive to locally similar features.

From the RMSE and MAE calculated for rotation and translation in Table II, it can be seen that the registration result obtained by our method is closest to the ground truth. Moreover, the average RMSEs for rotation and translation errors are 0.3523° and 0.2366 m, which are small comparing with the dimensions of the real scenes and only twice the average point spacing (0.1131 m). The experiment results further verify our proposed method performs well in registering the TLS point clouds for urban

TABLE II
QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON POINT CLOUDS OF URBAN SCENES.

Data	Avg. point spacing(m)	Method	Rotation err.(deg) RMSE	Rotation err.(deg) MAE	Translation err.(m) RMSE	Translation err.(m) MAE	$Time$ (s)
Fig. 13 (a)	0.1293	SAC-IA	1.5477	1.3717	3.3706	2.3568	125.775
		Super 4PCS	1.9096	1.1732	8.3882	7.0623	288.287
		FGR	2.4825	2.1026	1.3482	1.0782	57.6015
		Ours	**0.1808**	**0.1161**	**0.1591**	**0.1584**	**19.518**
Fig. 13 (b)	0.1273	SAC-IA	2.0659	1.7483	12.215	10.1572	110.663
		Super 4PCS	1.5577	1.3342	5.5534	4.4959	264.709
		FGR	1.6755	1.3719	1.9085	1.6097	39.0356
		Ours	**0.6721**	**0.5049**	**0.4852**	**0.4399**	**17.917**
Fig. 13 (c)	0.0828	SAC-IA	1.2137	1.0813	0.9366	0.8392	133.017
		Super 4PCS	4.9007	3.549	2.062	1.5555	322.872
		FGR	1.9462	1.4587	0.8228	0.613	61.0737
		Ours	**0.204**	**0.1442**	**0.0654**	**0.0561**	**28.811**

TABLE III
QUANTITATIVE COMPARISON OF OUR METHOD WITH HMMR [16].

Data	Avg. point Spacing(m)	Method	Rotation err.(deg) RMSE	Rotation err.(deg) MAE	Translation err.(m) RMSE	Translation err.(m) MAE
Fig. 14 (a)	0.0207	HMMR	0.0106	0.0071	**0.121**	**0.0841**
		Ours	**0.0091**	**0.0065**	0.1441	0.1361
Fig. 14 (b)	0.0548	HMMR	**0.0043**	**0.003**	**0.0043**	**0.004**
		Ours	1.6094	1.4761	0.394	0.3128
Fig. 14 (c)	0.0625	HMMR	×	×	×	×
		Ours	**1.3974**	**1.0748**	**0.4088**	**0.3372**

TABLE IV
QUANTITATIVE COMPARISON OF OUR METHOD WITH PLADE [14].

Data	Avg. point spacing(m)	Method	Rotation err.(deg) RMSE	Rotation err.(deg) MAE	Translation err.(m) RMSE	Translation err.(m) MAE
Fig. 15 (a)	0.0062	PLADE	1.1661	0.9738	**0.08**	**0.0697**
		Ours	**0.1815**	**0.1774**	0.1518	0.1414
Fig. 15 (b)	0.0086	PLADE	0.4361	0.3713	**0.0838**	**0.0658**
		Ours	**0.376**	**0.3483**	0.112	0.0933
Fig. 15 (c)	0.1198	PLADE	0.2716	0.2072	**0.1094**	**0.0936**
		Ours	**0.1815**	**0.1774**	0.1565	0.1338

scenes, especially considering that our method belongs to the coarse registration category. It can also be seen that since Super4PCS relies on searching for the corresponding 4-point-sets in $\mathcal{P}_s$ and $\mathcal{P}_t$, the corresponding relationships produced by this method have nothing to do with the similarity of local features but are related to the point numbers. Due to the significant local features in these examples, the efficiency of Super4PCS is slower than that of the other methods. For local feature-based methods, although the number of points participating in the FGR and SAC-IA operation is under the same downsample scale (in the FPFH-computing stage), FGR directly predicts the transformation parameters between the corresponding point pairs through optimization of an objective function, and it has a faster speed than SAC-IA. By contrast, our method registers the raw point clouds in the primitive space, thus the efficiency depends on the performance of the primitive shape extraction step, descriptor construction and matching step, and the transformation estimation step. In terms of the primitive shape extraction step, its efficiency is also related to the point number; however, it is still much faster than point-based registrations. For the other two steps, the efficiency depends on the primitive number and the number of matched descriptor pairs which are much less than the original point number and the feature-point number, respectively. Meanwhile, in the entire processing chain of our method, the primitive extraction step is the most computationally expensive part, and the time consumptions in Fig. 13 (a) to (c) are 19.327 s, 17.712 s and 28.623 s, respectively. In comparison with SAC-IA, Super4PCS and FGR, our method is the most efficient.

Comparison to HMMR on the Whu-TLS-dataset. We now compare our method with a hierarchical merging-based registration method (HMMR) on the Whu-TLS-dataset [16]. Note that the HMMR method contains a fine registration step which is not utilized in our method. In Fig. 14, we visualize the comparison results, where the ground truth dimensions of scenes (a), (b) and (c) are 388.616 m × 155.318 m × 10.5235 m, 255.592 m × 262.477 m × 93.394 m, and 899.628 m × 898.404 m × 57.5262 m, respectively. For the "subway station" scene in Fig. 14 (a), our method successfully registers the raw point clouds and achieves better rotation estimation performance than HMMR, which is also verified in Table III. Because we use primitives to register the point clouds, our method has obvious advantages in the urban scenarios; however, our method can also perform registrations on some nature scenes by extracting and matching the approximately accurate primitives, see the "mountain" scene shown in Fig. 14 (b). From Table III, we can see that since HMMR utilizes binary shape context descriptors, they are more capable of dealing with nature areas and obtains higher accuracy. However, HMMR fails in the registration of the difficult "railway" scene (see Fig. 14 (c)) because it contains too many similar local features. By contrast, we can still register the input by matching primitives despite relatively large errors occurring in both rotation and translation.

Comparison to PLADE on the RESSO dataset. We now compare our method with the state-of-the-art plane-based registration method, PLADE, on the 3Dmatch and RESSO [14] datasets. Fig. 15 and Table IV report the comparison results. The ground truth dimensions of scenes in Fig. 15 (a), (b) and (c) are 3.298 m × 2.072 m × 2.19 m, 12.2759 m × 14.5238 m × 5.9677 m, and 338.641 m × 242.85 m × 153.613 m, respectively. In Fig. 15 (a), our method recovers the transformation parameters using the descriptor based only on planes, while in Fig. 15 (b) and (c), the best matching descriptors constructed by our method consist of planes (1 in (b) and 2 in (c)) and cylinders (1 in (b) and (c)). From the results, we see that our method can achieve similar performance to PLADE on point clouds with planar primitives. However, in addition to plane-based shapes, our method can also handle point cloud registration for shapes and scenes that are composed of curved surfaces.

Comparison on point clouds without overlapping areas.

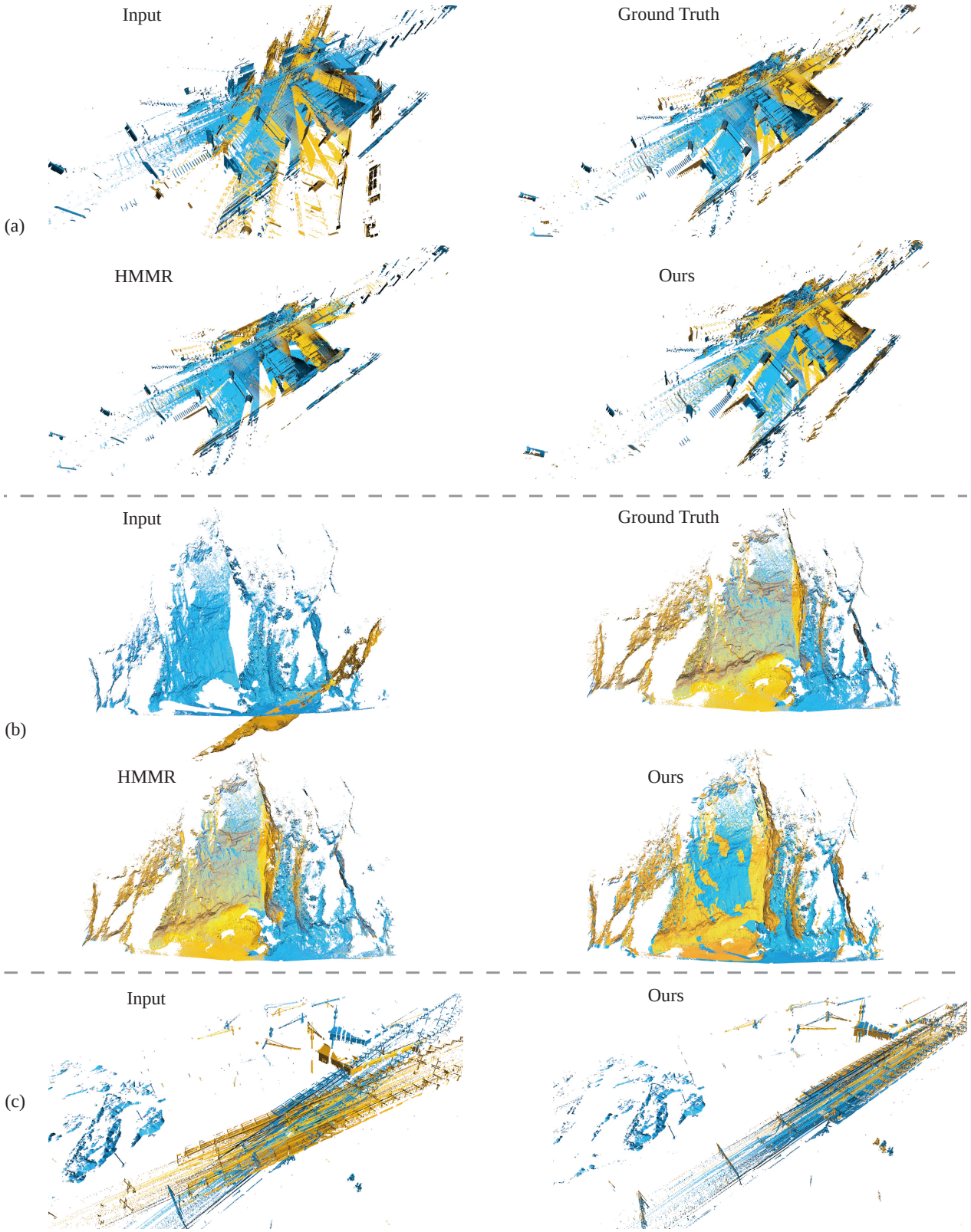

Fig. 14. Comparison with HMMR [16] on large-scale outdoor scans in the Whu-TLS-dataset.

In Fig. 16, we make a further comparison on point clouds without any overlapping area. We obtain point clouds by scanning objects from two opposite views or crop the existing views so that there do not exist overlapping areas between each pair of point clouds. We also downsample source $\mathcal{P}_s$ and target $\mathcal{P}_t$ in Fig. 16 (a) and (b) with different scales to ensure that they have different average point spacing, *e.g.,* 0.0124 m in $\mathcal{P}_s$ and 0.0208 m in $\mathcal{P}_t$ for Fig. 16

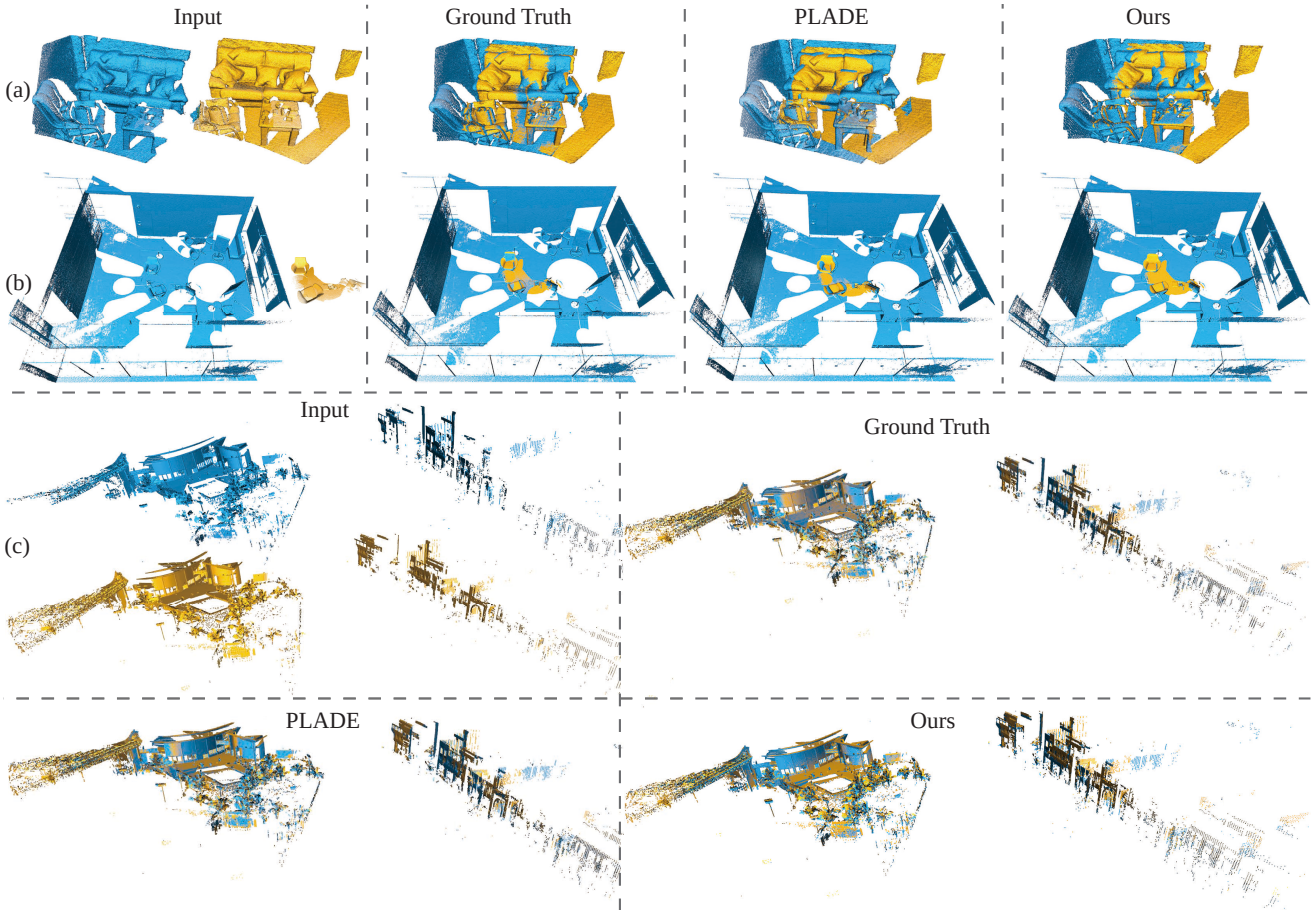

Fig. 15. Comparison with PLADE [14] on different pairs of point clouds in the 3DMatch dataset (a) and RESSO dataset ((b) and (c)).

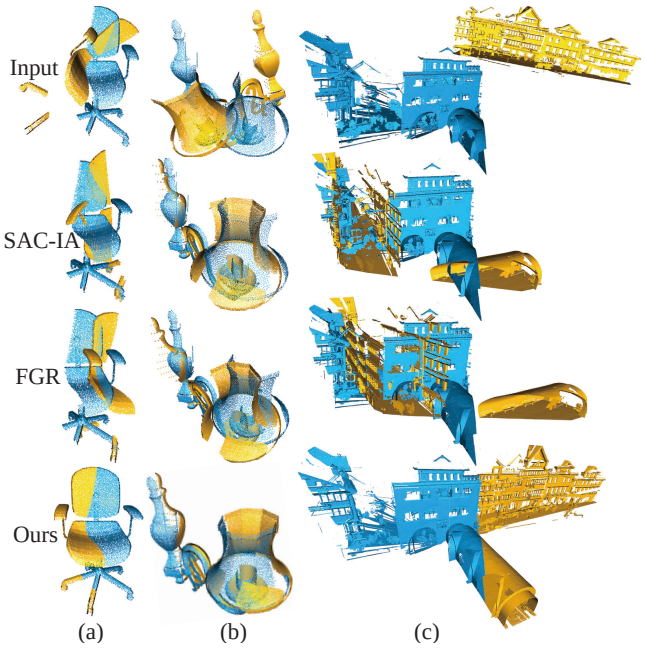

Fig. 16. Comparison with previous methods for registering point clouds with different point densities and without any overlapping area.

(b). As shown in Fig. 16, SAC-IA and FGR perform poorly due to the absence of enough point correspondences. Since our method is based on the abstraction of the primitives in the structural space, we can successfully register $\mathcal{P}_s$ and $\mathcal{P}_t$ with high accuracy by matching the overlapping primitives.

Comparison on 3DMatch dataset. We now compare registrations of indoor scenes on the 3DMatch dataset. In addition to SAC-IA [24] and FGR [28], we also compare to a deep learning method 3DSmoothNet [44], which achieves state-of-the-art performance on the 3DMatch dataset.

Fig. 17 and Table V report the qualitative and quantitative comparison results, where the ground truth dimensions are 2.994 m $\times$ 1.242 m $\times$ 2.178 m, 2.628 m $\times$ 2.4 m $\times$ 2.514 m, 2.67 m $\times$ 3.5936 m $\times$ 4.3673 m, 4.0586 m $\times$ 2.4444 m $\times$ 2.4879 m. In this experiment, due to incorrect point pair correspondence, FGR failed in all cases, while SAC-IA only achieved the near-accurate result shown in Fig. 17 (a) and still had considerable RMSE and MAE errors. 3DSmoothNet has successfully registered point clouds in Fig. 17 (a) and (b), but the quantitative statistics show that our results have lower RMSE and MAE. The difficulty of the cases in Fig. 17 (c) and (d) is that the overlapping area between the two-frame point clouds is too small and has too many repetitive local features. The comparison shows that only our method has achieved successful registration results thanks to the best matching between primitives in the structural space. Although we notice a relatively large error in our result on the example in Fig. 17 (c), our approach

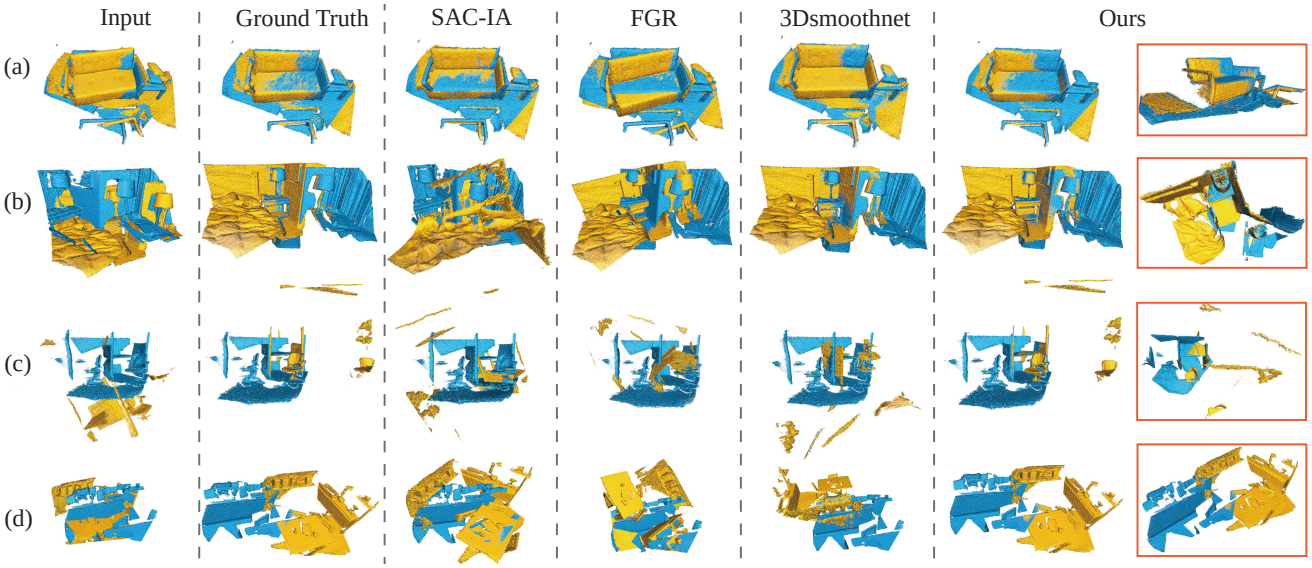

Fig. 17. Comparison with previous methods on four pairs of indoor point clouds in 3DMatch dataset.

TABLE V
QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON 3DMATCH DATASET. [44]

Data	Avg. point spacing(m)	Method	Rotation err.(deg)		Translation err.(m)		$Time$
			RMSE	MAE	RMSE	MAE	(s)
Fig. 17 (a)	0.0062	SAC-IA	1.7688	1.564	0.0885	0.0757	82.637
		FGR	8.0091	6.2351	0.3366	0.2551	**4.1936**
		3DSmoothNet	0.4504	0.4374	0.0182	0.0166	56.1282
		Ours	**0.4165**	**0.4079**	**0.0097**	**0.0086**	4.5423
Fig. 17 (b)	0.0062	SAC-IA	98.5408	89.8151	1.9669	1.8997	173.525
		FGR	5.8391	5.4922	0.0834	0.083	**6.3354**
		3DSmoothNet	1.0537	0.9413	0.0325	0.0272	57.1807
		Ours	**0.9353**	**0.9123**	**0.0207**	**0.0154**	6.609
Fig. 17 (c)	0.0062	SAC-IA	93.4877	93.2583	3.3444	3.0574	86.014
		FGR	114.573	114.269	2.1334	2.0335	**5.5208**
		3DSmoothNet	127.282	106.317	1.8102	1.806	55.1595
		Ours	**6.16**	**5.0516**	**0.088**	**0.0798**	12.2228
Fig. 17 (d)	0.0062	SAC-IA	26.5245	21.6016	1.3629	1.2677	179.873
		FGR	56.0427	49.4405	1.7233	1.6502	**8.1258**
		3DSmoothNet	116.658	103.406	3.3875	3.03	58.1974
		Ours	**0.967**	**0.7684**	**0.0318**	**0.0251**	13.981

still outperforms other methods, especially for point clouds with small overlapping areas.

From the perspective of algorithm efficiency, FGR is the fastest. Since the deep learning method 3DSmoothNet is a two-stage method and its feature point acquisition speed in the first stage is not fast, the efficiency is slow. From Table V, we can see that as the number of points increases (*e.g.,* from (a) to (b), and (c) to (d)), the efficiency of FGR decreases slightly, while the efficiency of SAC-IA is greatly reduced. By contrast, the primitive shape extraction step is less sensitive than the point-based registrations to the increasing point number, and its time consumption related to Fig. 17 (a) to (d) is 4.5183 s, 6.264 s, 12.0928 s and 13.537 s, respectively. In addition, since the efficiency of our core registration steps only depends on the primitive number and the number of descriptor pairs that are matched, our method is still fast enough in registering indoor scenes.

D. Discussion

To further analyze the characteristics of the proposed hybrid structural descriptors, we provide a comprehensive discussion of the overall performance of our method in experiments shown in Fig. 2, Fig. 8, Fig. 9 and Figs. 12-17. Similar to other registration methods, the robustness of our descriptors mainly relies on the percentage of overlapping primitives and the effectiveness of primitive configurations. Note that the percentage of overlapping primitives is not rigorously related to the percentage of the overlapping areas. Even though the overlapping areas do not exist, the percentage of overlapping primitives may still be surprisingly large, *e.g.,* examples shown in Fig. 9 (d) and Fig. 16 (a), (b). In general, as the percentage of overlapping primitives increases, there are more useful primitive features, which then make our method generates more effective descriptors and prone to obtaining higher registration quality. In contrast, when the percentage of overlapping primitives reduces, there are more invalid descriptors constructed from the non-overlapping primitives, which makes it harder to obtain high-quality registration. However, since the hybrid structural descriptors accurately encode the local structures, the valid descriptor pairs can always be chosen under a strict constraint in the matching process and make correct registrations achieved, *e.g.,* examples shown in Fig. 12 and Fig. 17 (b), (c), (d).

We also make quantitative analysis about the registration performances in terms of different primitive configurations, except for the cases on the repeated scene in Fig. 12 and the hard case in Fig. 17 (c). It should also be noted that for the scene shown in Fig. 9, we only statistic the registration (d) as its representative. Table VI reports the average RMSEs of rotation and translation errors for registrations that matching different primitive configurations, where *pln*,

TABLE VI
QUANTITATIVE ANALYSIS OF THE REGISTRATION PERFORMANCE IN TERMS OF THE PRIMITIVE CONFIGURATION OF DESCRIPTORS.

Type	ID	Atomic structure	Primitive configuration	Num.	Avg. RMSE of rotation(deg.)	Avg. RMSE of translation(m)	Example
$\mathbf{D}_1$	1	$\mathcal{A}_3, \mathcal{A}_3$	2 *pln*, 2 *pln*	7	0.3576	0.1478	Fig. 17 (a)
	2	$\mathcal{A}_3, \mathcal{A}_3$	2 *pln*, 1 *cyl*	7	0.1197	0.0815	Fig. 16 (c)
	3	$\mathcal{A}_3, \mathcal{A}_3$	1 *cyl*, 1 *cyl*	1	0.0016	0.0035	Fig. 2 (e)
$\mathbf{D}_2$	4	$\mathcal{A}_3, \mathcal{A}_2$	2 *pln*, 1 *sph*	1	1.6094	0.394	Fig. 14 (b)
$\mathbf{D}_3$	5	$\mathcal{A}_4, \mathcal{A}_3$	1 *con*, 1 *cyl*	1	0.0852	0.0016	Fig. 16 (b)
	6	$\mathcal{A}_4, \mathcal{A}_3$	1 *con*, 2 *pln*	1	0.0207	0.0154	Fig. 17 (b)
	7	$\mathcal{A}_4, \mathcal{A}_3$	1 *pln*, 2 *cyl*	2	0.095	0.1516	Fig. 13 (a)
$\mathbf{D}_4$	8	$\mathcal{A}_3, \mathcal{A}_1$	1 *cyl*, 1 *pln*	2	0.291	0.065	Fig. 9 (d)
	9	$\mathcal{A}_3, \mathcal{A}_1$	2 *pln*, 1 *pln*	5	0.4359	0.1672	Fig. 17 (d)

cyl, *con* and *sph* stand for the plane, cylinder, cone and sphere, respectively. By comparison, plane and cylinder primitives are used most commonly in our experiments. At the same time, cone and sphere primitives are also utilized to register point clouds that lack enough accurate plane or cylinder primitives, *e.g.,* examples shown in Fig. 17 (b) and Fig. 14 (b). In terms of accuracy performance, the deviations of registrations by matching different primitive configurations are small. However, when ignoring configuration 4 whose primitives are approximately extracted in a nature scene, registrations by matching two plane-based descriptors, configurations 1 and 9, obtain the comprehensive worst performance with the largest rotation and translation errors. There are two main reasons for the phenomenon. On one hand, for plane-based descriptor pairs with the same composition, we only estimate the transformation parameters once. Therefore, since only one plane normal and one intersection line or two intersection lines contribute to the rotation estimation, the probability of estimating rotations by matching the 2 direction vector pairs that have the highest similar degrees in 3- or 4-plane-based configurations is only $1/C_3^2 = 1/3$ or $1/C_4^2 = 1/6$, respectively. Although estimating the rotations between every plane-based descriptor pairs with the same composition can reduce the rotation error, the computation consumption also increases by 3 or 6 times, which is not affordable when there are too many planes. On the other hand, in order to acquire the translation parameters, it's essential to compute the intersection point of three planes or the midpoint of two intersection lines in each 3- or 4-plane-based configuration, which may accumulate more error compared with the directly computing the intersection point or midpoint of two primitive elements. In conclusion, the overall experimental results demonstrate that the proposed hybrid structural descriptors encode structural features with high accuracy, as well as making our method performs well in registering point clouds of urban/semi-urban scenes, indoor scenes and individual objects, which can satisfy the requirements of downstream applications in remote sensing and computer vision fields.

TABLE VII
QUANTITATIVE ANALYSIS OF THE REGISTRATION PERFORMANCE IN TERMS OF USING ONLY PLANE PRIMITIVES AND MULTI TYPES OF PRIMITIVES.

Data	Using only plane primitives		Using multi types of primitives	
	RMSE(R) (deg)	RMSE(T) (m)	RMSE(R) (deg)	RMSE(T) (m)
Fig. 10 "Scan 1-2"	0.1905	0.2954	**0.0782**	**0.0701**
Fig. 10 "Scan 3-4"	0.1916	0.2589	**0.1329**	**0.023**
Fig. 10 "Scan 9-8"	0.1931	0.2839	**0.0303**	**0.0662**
Fig. 10 "Scan 10-9"	0.1599	0.231	**0.0686**	**0.0526**
Fig. 13 (a)	0.2245	0.2885	**0.1808**	**0.1591**

E. Limitations

We successfully perform point cloud registration by matching geometric primitives under relation constraints in a structural space. Since the construction of the hybrid-structure-based descriptors requires the point cloud to be stereoscopic, our method cannot handle the registration of point clouds with a shape that is close to a plane. In addition, according to the registration errors that statistic in Table VII, although the registration performances of using multi types of primitives are good, it has large uncertainty when only planes are used or only planes are available in the scene. Moreover, our method aims at registrations on urban/semi-urban and indoor scenes, for complicated natural scenes that are hard to extract the approximately accurate primitives (*e.g.,* humans, forests), it remains a challenge for us to successfully constructing our structure-based descriptors, thus the proposed method often fails to return correct solutions.

VI. CONCLUSION AND FUTURE WORK

We have presented a new approach for point cloud registration. Our method transforms the point clouds into the middle-level structural spaces, achieves accurate registration by matching the structure-based descriptors that capture the relationships between geometric primitives. Experiments prove that our method is robust to data that contain noise, partial points, small overlapping areas, and even none overlapping areas. We also demonstrated the advantages of our approach by comparing to the state-of-the-art methods on previous benchmark datasets.

Although our method performs registration based on the middle-level geometric structures, it still lacks an understanding of high-level semantics. This causes our method to have larger registration errors when it encounters the point clouds are full of extremely similar geometric structures. In future work, we would like to incorporate high-level semantic information into our method to improve the accuracy and efficiency of feature matching. Besides, we would also like to use graph optimizations to further improve the robustness of hybrid structural features and estimate the transformation parameters simultaneously in multiple scans by using a non-linear optimization method.

ACKNOWLEDGMENTS

We thank anonymous reviewers for their valuable comments. This work is partially funded by the National Key R&D Program of China (2018YFB2100602), Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA23090304), the National Natural Science Foundation of China (61802406, U2003109, 61972388), the Key Research Program of Frontier Sciences CAS (QYZDY-SSW-SYS004), Shenzhen Basic Research Program (JCYJ20180507182222355), the Youth Innovation Promotion Association of the Chinese Academy of Sciences (Y201935), and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin, "Registration of 3d point clouds and meshes: A survey from rigid to nonrigid," *IEEE Trans. on Vis. and Comp. Graphics*, vol. 19, no. 7, pp. 1199–1217, 2012.

[2] B. Maiseli, Y. Gu, and H. Gao, "Recent developments and trends in point set registration methods," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 95–106, 2017.

[3] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[4] Y. Chen and G. G. Medioni, "Object modeling by registration of multiple range images." *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, 1992.

[5] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: application to breast mr images," *IEEE transactions on medical imaging*, vol. 18, no. 8, pp. 712–721, 1999.

[6] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *International Conference on 3-D Digital Imaging and Modeling*. IEEE, 2001, pp. 145–152.

[7] H. Li, R. W. Sumner, and M. Pauly, "Global correspondence optimization for non-rigid registration of depth scans," in *Computer Graphics Forum*, vol. 27, no. 5. Wiley Online Library, 2008, pp. 1421–1430.

[8] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[9] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Computer graphics forum*, vol. 32, no. 5. Wiley Online Library, 2013, pp. 113–123.

[10] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "SEMANTIC3D.NET: A new large-scale point cloud classification benchmark," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1-W1, 2017, pp. 91–98.

[11] J. Xiao, B. Adler, and H. Zhang, "3d point cloud registration based on planar surfaces," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2012, pp. 40–45.

[12] Y. Shi, K. Xu, M. Niessner, S. Rusinkiewicz, and T. Funkhouser, "Planematch: Patch coplanarity prediction for robust rgb-d reconstruction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 750–766.

[13] Y. Xu, R. Boerner, W. Yao, L. Hoegner, and U. Stilla, "Automated coarse registration of point clouds in 3d urban scenes using voxel based plane constraint." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2017.

[14] S. Chen, L. Nan, R. Xia, J. Zhao, and P. Wonka, "Plade: A plane-based descriptor for point cloud registration with small overlap," *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[15] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," in *Computer Graphics Forum*, vol. 26, no. 2. Wiley Online Library, 2007, pp. 214–226.

[16] Z. Dong, B. Yang, F. Liang, R. Huang, and S. Scherer, "Hierarchical registration of unordered tls point clouds based on binary shape context descriptor," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 144, pp. 61–79, 2018.

[17] P. W. Theiler, J. D. Wegner, and K. Schindler, "Globally consistent registration of terrestrial laser scans via graph optimization," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 109, pp. 126–138, 2015.

[18] R. Huang, Y. Xu, L. Hoegner, and U. Stilla, "Temporal comparison of construction sites using photogrammetric point cloud sequences and robust phase correlation," *Automation in Construction*, vol. 117, p. 103247, 2020.

[19] R. Huang, Y. Xu, L. Hoegner, and U. Stilla, "Efficient estimation of 3d shifts between point clouds using low-frequency components of phase correlation," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 227–234, 2020.

[20] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *International Conference on Computer Vision Workshops*. IEEE, 2009, pp. 689–696.

[21] I. Sipiran and B. Bustos, "Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes," *The Visual Computer*, vol. 27, no. 11, p. 963, 2011.

[22] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3d local feature descriptors," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 66–89, 2016.

[23] A. E. Johnson, "Spin-images: a representation for 3-d surface matching," 1997.

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

[25] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 356–369.

[26] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3d local surface description and object recognition," *Int. Journal of Computer Vision*, vol. 105, no. 1, pp. 63–86, 2013.

[27] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *European Conference on Computer Vision (ECCV)*. Springer, 2004, pp. 224–237.

[28] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 766–782.

[29] A. Kaiser, J. A. Ybanez Zepeda, and T. Boubekeur, "A survey of simple geometric primitives detection methods for captured 3d data," in *Computer Graphics Forum*, vol. 38, no. 1, 2019, pp. 167–196.

[30] E. Che and M. J. Olsen, "Multi-scan segmentation of terrestrial laser scanning data based on normal variation analysis," *ISPRS journal of photogrammetry and remote sensing*, vol. 143, pp. 233–248, 2018.

[31] A. Habib, M. Ghanma, M. Morgan, and R. Al-Ruzouq, "Photogrammetric and lidar data registration using linear features," *Photogrammetric Engineering & Remote Sensing*, vol. 71, no. 6, pp. 699–707, 2005.

[32] M. Al-Durgham and A. Habib, "A framework for the registration and segmentation of heterogeneous lidar data," *Photogrammetric Engineering & Remote Sensing*, vol. 79, no. 2, pp. 135–145, 2013.

[33] B. Yang and Y. Zang, "Automated registration of dense terrestrial laser-scanning point clouds using curves," *ISPRS journal of photogrammetry and remote sensing*, vol. 95, pp. 109–121, 2014.

[34] C. Dold and C. Brenner, "Registration of terrestrial laser scanning data using planar patches and image data," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS*, vol. 36, pp. 78–83, 2006.

[35] T.-Y. Chuang and J.-J. Jaw, "Multi-feature registration of point clouds," *Remote Sensing*, vol. 9, no. 3, p. 281, 2017.

[36] A. Hattab and G. Taubin, "3d rigid registration of cad point-clouds," in *2018 International Conference on Computing Sciences and Engineering (ICCSE)*. IEEE, 2018, pp. 1–6.

[37] Y. Xu, R. Boerner, W. Yao, L. Hoegner, and U. Stilla, "Pairwise coarse registration of point clouds in urban scenes using voxel-based 4-planes congruent sets," *ISPRS journal of photogrammetry and remote sensing*, vol. 151, pp. 106–123, 2019.

[38] H. J. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," *IEEE computational science and engineering*, vol. 4, no. 4, pp. 10–21, 1997.

[39] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[40] C.-S. Chen, Y.-P. Hung, and J.-B. Cheng, "Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1229–1234, 1999.
[41] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," in *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 2008, pp. 1–10.
[42] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," in *Computer Graphics Forum*, vol. 33, no. 5. Wiley Online Library, 2014, pp. 205–215.
[43] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1802–1811.
[44] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5545–5554.
[45] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7163–7172.
[46] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.
[47] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.
[48] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 3523–3532.
[49] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1912–1920.
[50] A. Nüchter and K. Lingemann, "Robotic 3d scan repository," *Jacobs University Bremen gGmbH and University of Osnabrück*, 2011.

Long Zhang is working toward the Ph.D. degree in the School of Artificial Intelligence at University of Chinese Academy of Sciences, Beijing. He obtained his bachelor degree from Southwest University in 2014. His research interests include computer graphics, geometry processing and 3D reconstruction.

Jianwei Guo is an associate professor in National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA). He received his Ph.D. degree in computer science from CASIA in 2016, and bachelor degree from Shandong University in 2011. His research interests include 3D vision, computer graphics and geometry learning.

Zhanglin Cheng is a professor in the Shenzhen Key Laboratory of Visual Computing and Analytics (VisuCA), Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. He received the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences in 2008. His research interests include computer graphics and visualization.

Jun Xiao is a professor in University of Chinese Academy of Sciences, Beijing. He obtained his Ph.D. degree in communication and information system from the Graduate University of Chinese Academy of Sciences in 2008. His research interests include computer graphics, computer vision, image processing and 3D reconstruction.

Xiaopeng Zhang is a Professor in National Laboratory of Pattern Recognition at Institute of Automation, Chinese Academic of Sciences (CAS). He received his Ph.D. degree in Computer Science from Institute of Software, CAS in 1999. He received the National Scientific and Technological Progress Prize (second class) in 2004. His main research interests include computer graphics, computer vision and image processing.