



Article

A Robust Rigid Registration Framework of 3D Indoor Scene Point Clouds Based on RGB-D Information

Saishang Zhong^{1,2} , Mingqiang Guo^{3,4,5} , Ruina Lv⁴, Jianguo Chen^{1,2}, Zhong Xie^{3,4} and Zheng Liu^{3,4,*}

¹ State Key Laboratory of Geological Processes and Mineral Resources, China University of Geosciences, Wuhan 430074, China; saishang@cug.edu.cn (S.Z.); jgchen@cug.edu.cn (J.C.)

² School of Earth Resources, China University of Geosciences, Wuhan 430074, China

³ National Engineering Research Center of Geographic Information System, China University of Geosciences, Wuhan 430074, China; guomingqiang@cug.edu.cn (M.G.); xiezhong@cug.edu.cn (Z.X.)

⁴ School of Geography and Information Engineering, China University of Geosciences, Wuhan 430074, China; ruinalv@cug.edu.cn

⁵ Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources, Shenzhen 518034, China

* Correspondence: liuzheng@cug.edu.cn

Abstract: Rigid registration of 3D indoor scenes is a fundamental yet vital task in various fields that include remote sensing (e.g., 3D reconstruction of indoor scenes), photogrammetry measurement, geometry modeling, etc. Nevertheless, state-of-the-art registration approaches still have defects when dealing with low-quality indoor scene point clouds derived from consumer-grade RGB-D sensors. The major challenge is accurately extracting correspondences between a pair of low-quality point clouds when they contain considerable noise, outliers, or weak texture features. To solve the problem, we present a point cloud registration framework in view of RGB-D information. First, we propose a point normal filter for effectively removing noise and simultaneously maintaining sharp geometric features and smooth transition regions. Second, we design a correspondence extraction scheme based on a novel descriptor encoding textural and geometry information, which can robustly establish dense correspondences between a pair of low-quality point clouds. Finally, we propose a point-to-plane registration technology via a nonconvex regularizer, which can further diminish the influence of those false correspondences and produce an exact rigid transformation between a pair of point clouds. Compared to existing state-of-the-art techniques, intensive experimental results demonstrate that our registration framework is excellent visually and numerically, especially for dealing with low-quality indoor scenes.

Keywords: rigid registration; RGB-D information; correspondence; point normal filter; nonconvex optimization



Citation: Zhong, S.; Guo, M.; Lv, R.; Chen, J.; Xie, Z.; Liu, Z. A Robust Rigid Registration Framework of 3D Indoor Scene Point Clouds Based on RGB-D Information. *Remote Sens.* **2021**, *13*, 4755. <https://doi.org/10.3390/rs13234755>

Academic Editors: Maarten Bassier, Florent Poux, Shayan Nikoohemat and Maarten Vergauwen

Received: 23 September 2021

Accepted: 19 November 2021

Published: 24 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, 3D point clouds of indoor scenes have been regarded as the most appropriate data source for generating a building information model (BIM), which has become a crucial tool for constructions and architecture professionals [1]. In the past, indoor scene point clouds were usually acquired by static terrestrial laser scanners (TLS). Although TLS have a high scanning precision, they are expensive. As a result, the high price greatly restricted the application and popularization of TLS. Fortunately, the recent prevalence of consumer-grade RGB-D cameras (e.g., Microsoft Kinect, Asus Xtion, etc.) has allowed ordinary people to easily capture 3D indoor scene point clouds from the real world by scanning and reconstruction processes [2]. During the reconstruction process, as a fundamental task, 3D point clouds registration aims at registering individual scans in a unified coordinate system for producing a complete 3D point cloud of the target indoor scene [1,3]. Typically, the rigid registration of point clouds consists in finding correspondence points and minimizing the sum of residuals over all correspondence points to

estimate the rigid transformation information, which consists of a 3×3 matrix representing the rotation information and a 3×1 vector denoting the translation information. It is more challenging to register point clouds acquired using RGB-D cameras. In the rest of this paper, we refer to point clouds acquired using RGB-D cameras as RGB-D point clouds. The main reason is that this type of point clouds is inevitably corrupted by comparatively large noise and outliers, due to many factors including sensor errors, occlusions, etc. Thus, robustly registering RGB-D point clouds has been a crucial problem in geometry modeling, computer vision, and so on.

Researchers have extensively studied rigid registration of 3D point clouds and proposed many remarkable methods. For example, Besl and McKay [4] have introduced the popular iterative closest point (abbreviated as ICP) scheme. This method iteratively estimates correspondences and computes transformation information. Due to its simplicity, researchers have applied the ICP algorithm for developing 3D reconstruction systems, such as the famous Kinect-Fusion system [5,6]. Nevertheless, the main drawbacks of the ICP algorithm are its slow convergence and sensitivity to noise. Moreover, it is also computationally expensive. Researchers have also designed many registration methods in terms of correspondence extraction and point clouds alignment to address these issues. To extract correspondences accurately, researchers first introduced the methods based on point features including the fast point feature histograms (abbreviated as FPFH) feature [7] and integral volume descriptor [8]. Then, to enhance robustness of the above methods, more approaches were proposed by using high level information, such as color information [9–14], planar structure [15,16], and hybrid structure [17], etc. For point clouds alignment, many remarkable methods have been presented in terms of efficiency and robustness. For efficiency, Bylow et al. [18] have investigated the point-to-point metric and the point-to-plane metric. The Anderson acceleration strategy has been adopted to improve the convergence rate of the ICP algorithm [19,20]. For robustness against noise and outliers, researchers have developed numerous advanced methods based on many techniques, including the least trimmed squares [21], ℓ_p sparsity optimization [22], nonconvex optimization [20,23], maximum correntropy criterion (MCC) [24], branch-and-bound scheme [25]. Recently, deep-learning-based registration methods [26–30] have demonstrated promising results. However, their performance is limited by the completeness of training data sets.

However, the aforementioned methods still fail to register RGB-D point clouds with considerable noise and weak texture. Motivated by this observation, we present a new rigid registration framework to effectively deal with RGB-D point clouds. The key idea is to fully utilize texture and geometry information computed from RGB-D images to build correspondences between point clouds accurately. Specifically, the proposed framework consists of three consequent stages, i.e., point normal estimation, correspondence point extraction, and point clouds alignment. In the first stage, we introduce a variational normal estimation method by coupling the total variation model with a second-order operator, which can effectively remove the noise of input point clouds and simultaneously maintain sharp geometric features and smooth transition regions. In the second stage, we design a correspondence extraction method with the help of the RGB information, which can robustly extract corresponding points from noisy point clouds. Finally, we utilize a fast optimization-based method to compute rigid transformation matrices for aligning pairs of point clouds. The experiments on multiple open-source RGB-D datasets demonstrate the superiority of our method, especially its robustness against low-quality point cloud data.

Specifically, this paper has the following main contributions:

- We present a point normal estimation method by coupling total variation with second-order variation. The method is capable of effectively removing noise while keeping sharp geometric features and smooth transition regions simultaneously.
- We present a robust correspondence points extraction method, based on a descriptor (TexGeo) encoding both texture and geometry information. With the help of the TexGeo descriptor, the proposed method is robust when handling low-quality point clouds.

- We design a point-to-plane registration method based on a nonconvex regularizer. The method can automatically ignore the influence of those false correspondences and produce an exact rigid transformation between a pair of noisy point clouds.
- We verify the robustness of our approach on a variety of low-quality RGB-D point clouds. Intensive experiments demonstrate that our approach outperforms the selected state-of-the-art methods visually and numerically.

2. Related Work

As a fundamental problem for many geometric modeling applications, the rigid registration of 3D point clouds has drawn great attention in the past decades. There is a lot of existing point cloud registration work in the existing literature. For a more comprehensive review of rigid 3D point cloud registration, readers are referred to [31]. Although most of the existing methods are remarkable, a discussion of the full literature is beyond the scope of this study. Thus, we only focus on rigid point cloud registration and mainly review techniques closely related to our work.

Point normal estimation. As an important signal indicating the direction field of the scanned surface, the point normal field has been widely applied for constructing 3D point descriptors, such as the FPFH [7]. Note that the 3D point descriptor is fundamental to correspondence extraction. However, it is challenging to robustly estimate point normals, since the captured point clouds are inevitably corrupted by noise and outliers. To address this issue, researchers have proposed many valuable methods. Here, we only review remarkable ones related to our work. Avron et al. [32] have applied ℓ_1 regularization to recover the point normal field. In order to preserve sharp edges and corners, Sun et al. [33] have derived a sparsity-based method that uses ℓ_0 minimization for effectively processing point clouds whose underlying surfaces are piecewise constant. These two methods can keep sharp geometric features while removing noise effectively. However, both of them inevitably suffer from serious staircase artifacts in smooth transition regions [34–40]. For alleviating these artifacts, Liu et al. [41] have recently introduced a point cloud denoising framework, which presents an anisotropic second-order regularizer to remove noise and preserve sharp geometric features as well as smooth transition regions.

Correspondence extraction. Correspondence extraction consists in matching points to determine a coarse alignment. Existing methods are designed based on either point features or structure information to construct descriptors. Gelfand et al. [8] have identified features and extracted correspondences using a novel integral volume descriptor. Similar to [7], Zhou et al. [23] have utilized the FPFH descriptor to match points efficiently. In order to register RGB-D point clouds, the authors of [9,14] have applied texture information for extracting correspondences. Though the above methods can extract correspondences effectively, they are easily disturbed by large noise. Different from the above methods, the methods based on structure information need to construct some meaningful structures. For example, Aiger et al. [15] have matched points by comparing approximately congruent coplanar four-point sets selected from a pair of point clouds. Their approach, called 4PCS, can robustly register point clouds without any assumption about their initial poses. However, it costs a lot of time when handling large-scale point clouds, because it performs the RANSAC random iteration process [42,43]. As a result, to improve the efficiency of 4PCS, Mellado et al. [44] have derived Super4PCS. Moreover, by using structural information including planes and lines as well as their interrelationship, Chen et al. [16] have matched points for pairs of point clouds whose overlap ratios are small. Zhang et al. [17] have introduced a registration framework that computes correspondences by using middle-level structural features.

Point clouds alignment. Point clouds alignment estimates the rigid transformation for registering a pair of point clouds given the extracted correspondences. To this end, the ICP algorithm iteratively minimizes the sum of the ℓ_2 distance (i.e., point-to-point distance or point-to-plane distance) between correspondence points [4,18]. Though the ICP algorithm is simple, it is not only sensitive to noise and outliers, but also computation-

ally expensive. To overcome these limitations, Chetverikov et al. [21] have introduced a trimmed ICP algorithm, which can robustly register incomplete point clouds with noise. By utilizing a branch-and-bound scheme, Yang et al. [25] have presented the Go-ICP, which is a global algorithm for point cloud registration. Bouaziz et al. [22] have presented the sparse ICP algorithm that formulates the registration problem as an ℓ_p minimization problem. Though their method can limit the effect of noise and outliers on the aligned results by adjusting the value of parameters p , it is time-consuming to solve the nonconvex optimization problem. To conquer the issue, Mavridis et al. [45] have improved the sparse ICP for more efficiently solving the nonconvex optimization problem. Wu et al. [24] have eliminated the interference of outliers and noise by using the maximum correntropy criterion. Zhou et al. [23] have introduced a robust global approach by utilizing a scaled Geman–McClure function, which can automatically reduce wrong correspondences. To improve the convergence of the ICP algorithm, Rusinkiewicz [46] have presented a symmetric objective function. Furthermore, to speed up the ICP algorithm, Pavlov et al. [19] have proposed AA-ICP, a novel modification of the ICP algorithm based on Anderson acceleration, which substantially reduces the number of iterations with a negligible cost. However, when the ground-truth rotation is close to a gimbal lock [47], the AA-ICP method cannot produce the desired result. To alleviate this issue, Zhang et al. [20] have recently proposed a fast and robust variant of the ICP algorithm using Welsch’s function. Moreover, an Anderson-accelerated majorization-minimization algorithm has been proposed to solve their problem.

As a result, although the existing point clouds registration algorithms have good performance when processing point clouds corrupted by small-scale noise, their performance are significantly degraded when point clouds are corrupted by comparatively large noise. This situation becomes even worse when processing RGB-D point clouds. Thus, this paper proposes a robust registration framework for 3D indoor scene point clouds based on RGB-D information to address this issue.

3. Methodology

Figure 1 shows the pipeline of our method. In the first step, given a pair of point clouds **A**, **B**, we estimate point normals to provide the smoothed normal field, which can represent the orientation of the underlying surface (see Section 3.1). Note that we do not change point positions at this step. In the second step, called correspondence point extraction (Section 3.2), we first compute a novel TexGeo descriptor for each point, based on the filtered point normals and the texture information of RGB images, and then produces the point matching results (i.e., correspondence points) of the pair of point clouds. In the third step (i.e., point clouds alignment presented in Section 3.3), based on the correspondence points obtained in step two, we compute the final rigid transformation information, i.e., **R**, **t**.

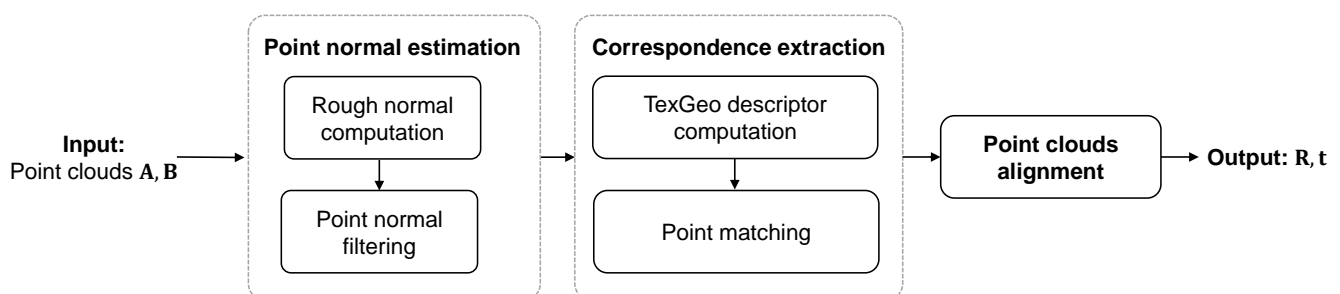


Figure 1. The pipeline of our method consisting of point normal estimation, correspondence extraction, and point clouds alignment.

Derived from RGB-D cameras, the RGB-D information consists of pairs of RGB and depth images, which capture the textural and geometry information of the scanned scene. To our knowledge, the RGB-D information has been widely used in many fields, such as 3D

reconstruction [9,48], simultaneous localization and mapping (abbreviated as SLAM) [49,50], and so on. Specifically, we assume that all the RGB and depth images are well registered. Moreover, since we produce point clouds using depth images with a pin-hole model, each depth image and its corresponding point cloud have the same size. Thus, we assume the size of each depth image to be $M \times N$, and denote each point cloud as $\{\mathbf{p}_{i,j}\}_{i=1,j=1}^{M,N}$.

3.1. Point Normal Estimation

Point normals are frequently used to construct 3D point descriptors, e.g., the FPFH feature [7] and the Harris feature [51], depicting the geometric characters of local surface regions. However, even with high-fidelity devices, real-scanned point clouds are inevitably corrupted by noise. Moreover, there are even some bumps on RGB-D point clouds [52]. Thus, it is necessary to remove the noise of point normals to construct 3D point descriptors that can accurately reflect the geometric characteristics of scanned surfaces. To this end, we first compute rough point normals, and then filter them by using a novel point normal filter. Details are elaborated as follows.

Rough point normal computation. We can easily compute rough point normals from the corresponding depth image. Formally, for each point $\mathbf{p}_{i,j}$, we compute its rough normal, $\hat{\mathbf{N}}_{i,j}$, as

$$\hat{\mathbf{N}}_{i,j} = \Phi((\mathbf{p}_{i+1,j} - \mathbf{p}_{i,j}) \times (\mathbf{p}_{i,j+1} - \mathbf{p}_{i,j})), \quad (1)$$

where $\Phi(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|}$ is the normalized function. Due to the effects of both intrinsic factors (i.e., device errors) and extrinsic factors (e.g., ambient light), the points generated from the depth image tend to be noisy. Therefore, the rough normals given by (1) are inevitably corrupted by noise.

Point normal filtering. To filter out the noise of the rough point normals, we present a sparsity-inspired global optimization method, whose goal is to find the smoothed point normals that best fit the input rough normals and the given sparsity constraints. We first design a point normal filtering model by using the total variation model and a second-order operator. Then, we present an iterative algorithm to minimize the point normal filtering model.

First of all, we briefly give the definition of a novel second-order operator defined on point clouds for measuring the second-order variations of local regions. Figure 2 shows the auxiliary geometric elements for constructing the second-order operator. For a point $\mathbf{p}_{i,j}$, we define its anticlockwise ordered 1-ring neighborhood as $\Omega(i,j) = \{\mathbf{p}_{i-1,j}, \mathbf{p}_{i+1,j}, \mathbf{p}_{i,j-1}, \mathbf{p}_{i,j+1}\}$. Let $e(i,j) = \{e = (\mathbf{p}_{i,j}, \mathbf{p}) : \mathbf{p} \in \Omega(i,j)\}$ be the set of edges connecting to $\mathbf{p}_{i,j}$. Then, we denote all the edges of a point cloud as $E = \{e(i,j)\}_{i=1,j=1}^{M,N}$. Let l be a line connecting $\mathbf{p}_{i,j}$ with a midpoint (e.g., m_l) of two consecutive neighbors (see Figure 2b). Let $L = \{\mathcal{L}(i,j)\}_{i,j}^{M,N} = \{l = (\mathbf{p}_{i,j}, m_k)\}_{i,j}^{M,N}$ be all such lines of a point cloud. Based on the above auxiliary elements, the operator can be defined as:

$$(D^2 \mathbf{u})_l = (w_j + w_k) \mathbf{u}_i - w_j \mathbf{u}_j - w_k \mathbf{u}_k, \quad (2)$$

where \mathbf{u} is the signal field defined on the point cloud, and w_j, w_k are two positive weighting parameters. If both w_j, w_k are set to 1, the operator (2) degenerates into an isotropic second-order operator. Recently, using the operator (2), Liu et al. [41] proposed a feature-preserving point denoising framework, which can remove small-scale noise and simultaneously keep sharp geometric features as well as nonlinear smooth transition regions. However, it is hard to keep geometric features in this framework when the noise level is high. Motivated by this situation, we present a point normal filtering model by combining the total variation with the second-order variation, which is formulated as:

$$\min_{\mathbf{N} \in \mathbb{C}} \left\{ \alpha \sum_{e \in E} \text{len}(e) w_e \left\| (D^1 \mathbf{N})_e \right\| + \beta \sum_{l \in L} \text{len}(l) \left\| (D^2 \mathbf{N})_l \right\| + \frac{1}{2} \sum_{i,j} \text{disk}(p_{i,j}) \left\| \mathbf{N}_{i,j} - \hat{\mathbf{N}}_{i,j} \right\|^2 \right\}, \quad (3)$$

where $len(e), len(l)$ are the length of edge e and line l , respectively. Moreover, $disk(p_{i,j})$ represents the area of a circle of radius r centered at $p_{i,j}$. Note that r is the average length of the edges contained in $e(i, j)$, and L is the set of auxiliary geometric lines of the input point cloud, $C = \{\mathbf{N}_i : \|\mathbf{N}_i\| = 1\}$; α, β are parameters for balancing the terms in (3).

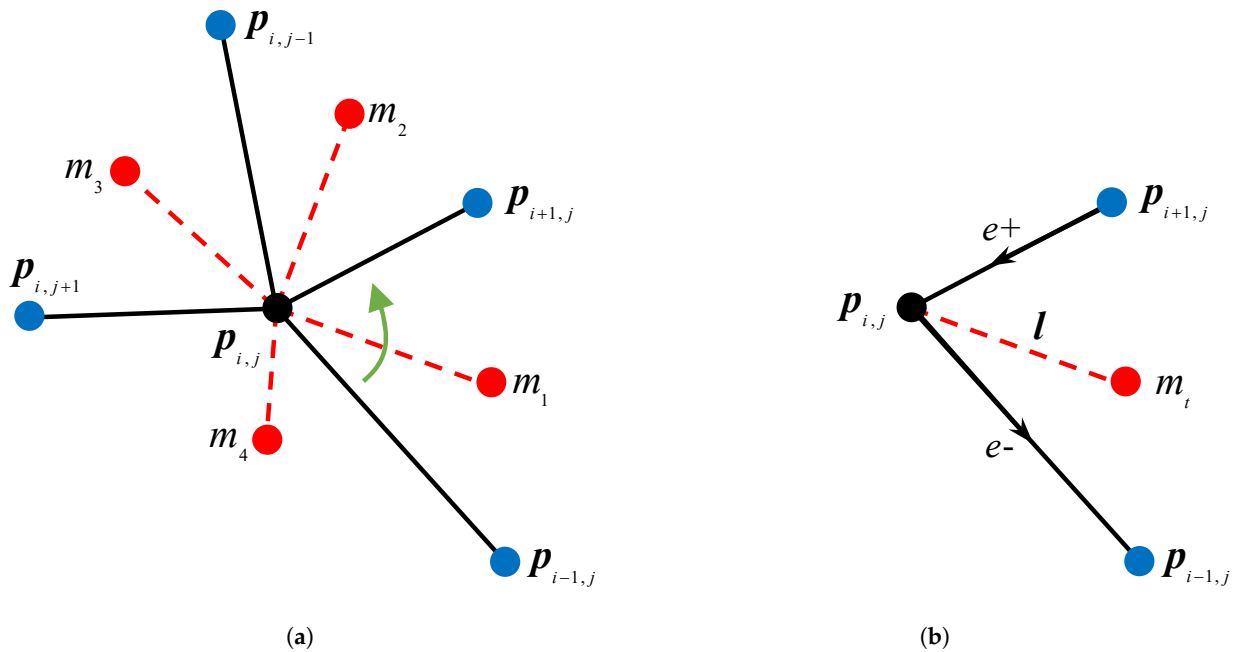


Figure 2. (a) Auxiliary geometric elements of point $p_{i,j}$, including neighbor points (colored in blue) and the midpoints of two neighbor points (colored in red), and auxiliary edges (colored in red). (b) Demonstration of second-order variation defined over the line l .

To obtain filtered point normals, we need to solve the minimization problem (3). However, since the the problem (3) has non-differentiability and nonlinear constraints, we cannot directly solve (3). Inspired by studies [34,35,41], we propose an iterative algorithm based on an augmented Lagrangian method. First, we define two auxiliary variables \mathbf{X}, \mathbf{Y} , and rewrite the minimization problem (3) as:

$$\min_{\mathbf{N}, \mathbf{X}, \mathbf{Y}} \left\{ \alpha \sum_{e \in E} len(e) w_e \|\mathbf{X}_e\| + \beta \sum_{l \in L} len(l) \|\mathbf{Y}_l\| + \frac{1}{2} \sum_i disk(\mathbf{p}_i) \|\mathbf{N}_i - \hat{\mathbf{N}}_i\|^2 + \psi(\mathbf{N}) \right\} \quad (4)$$

s.t. $\mathbf{X} = D^1 \mathbf{N}, \mathbf{Y} = D^2 \mathbf{N}$,

where $\psi(\mathbf{N}) = 0$, if $\mathbf{N} \in C$; otherwise $\psi(\mathbf{N}) = +\infty$.

To minimize the above problem, we first define the augmented Lagrangian function

$$\begin{aligned} L(\mathbf{N}, \mathbf{X}, \mathbf{Y}; \mu, \eta) = & \alpha \sum_{e \in E} len(e) \|\mathbf{X}_e\| + \beta \sum_{l \in L} len(l) \|\mathbf{Y}_l\| + \frac{1}{2} \sum_i disk(\mathbf{p}_i) \|\mathbf{N}_i - \hat{\mathbf{N}}_i\|^2 + \psi(\mathbf{N}) \\ & + \sum_{e \in E} len(e) w_e [\mu_e \cdot (\mathbf{X}_e - (D^1 \mathbf{N})_e)] + \frac{t_e}{2} \sum_{e \in E} len(e) w_e \|\mathbf{X}_e - (D^1 \mathbf{N})_e\|^2 \\ & + \sum_{l \in L} len(l) [\eta_l \cdot (\mathbf{Y}_l - (D^2 \mathbf{N})_l)] + \frac{t_l}{2} \sum_{l \in L} len(l) \|\mathbf{Y}_l - (D^2 \mathbf{N})_l\|^2, \end{aligned} \quad (5)$$

where $\mu = \{\mu_e\}, \eta = \{\eta_l\}$ are two Lagrangian multipliers and t_e, t_l are two positive penalty coefficients. Note that minimizing problem (3) is equivalent to minimizing function (5). The solving procedure of (5) mainly consists of three subproblems, i.e., the \mathbf{N} -subproblem, the \mathbf{X} -subproblem, and the \mathbf{Y} -subproblem.

(1) **N**-subproblem:

$$\min_{\mathbf{N}} \left\{ \frac{1}{2} \sum_i \text{disk}(\mathbf{p}_i) \|\mathbf{N}_i - \hat{\mathbf{N}}_i\|^2 + \frac{t_e}{2} \sum_{e \in E} \text{len}(e) w_e \|(D^1 \mathbf{N})_e - (\mathbf{X}_e + \frac{\mu_e}{t_e})\|^2 + \frac{t_l}{2} \sum_{l \in L} \text{len}(l) \|(D^2 \mathbf{N})_l - (\mathbf{Y}_l + \frac{\eta_l}{t_l})\|^2 + \psi(\mathbf{N}) \right\}. \quad (6)$$

This problem is quadratic, if we ignore the non-quadratic constraint term $\psi(\mathbf{N})$. Thus, we first obtain the solution of the quadratic problem, and then project the solution onto a unit sphere.

(2) **X**-subproblem:

$$\min_{\mathbf{X}} \alpha \sum_{e \in E} \text{len}(e) \|\mathbf{X}_e\| + \frac{t_e}{2} \sum_{e \in E} \text{len}(e) w_e \left\| \mathbf{X}_e - \left((D^1 \mathbf{N})_e - \frac{\mu_e}{t_e} \right) \right\|^2. \quad (7)$$

Obviously, we can solve each \mathbf{X}_e independently. Specifically, for each \mathbf{X}_e , we only solve the problem

$$\min_{\mathbf{X}_e} \alpha \|\mathbf{X}_e\| + \frac{t_e}{2} \left\| \mathbf{X}_e - \left((D^1 \mathbf{N})_e - \frac{\mu_e}{t_e} \right) \right\|^2.$$

Given the definition of $\text{shrink}(\mathbf{v}, \mathbf{w}) = \max\left(0, 1 - \frac{1}{\mathbf{v} \|\mathbf{w}\|}\right) \mathbf{w}$, the solution of this problem can be formulated as:

$$\mathbf{X}_e = \text{shrink}\left(t_e, (D^1 \mathbf{N})_e - \frac{\mu_e}{t_e}\right). \quad (8)$$

(3) **Y**-subproblem:

$$\min_{\mathbf{Y}} \beta \sum_{l \in L} \text{len}(l) \|\mathbf{Y}_l\| + \frac{t_l}{2} \sum_{l \in L} \text{len}(l) \left\| \mathbf{Y}_l - \left((D^2 \mathbf{N})_l - \frac{\eta_l}{t_l} \right) \right\|^2. \quad (9)$$

Similarity, the **Y**-subproblem can also be spatially decomposed. In other words, for each \mathbf{Y}_l , we solve the problem:

$$\min_{\mathbf{Y}_l} \beta \|\mathbf{Y}_l\| + \frac{t_l}{2} \left\| \mathbf{Y}_l - \left((D^2 \mathbf{N})_l - \frac{\eta_l}{t_l} \right) \right\|^2,$$

whose solution is as:

$$\mathbf{Y}_l = \text{shrink}\left(t_l, (D^2 \mathbf{N})_l - \frac{\eta_l}{t_l}\right). \quad (10)$$

Last, we show the overall procedure for solving problem (3) in Algorithm 1. As we can see, the proposed approach iteratively minimizes the above three subproblems and updates two Lagrangian multipliers.

Effectiveness of our point normal estimation. To test the effectiveness of our point normal filter, we ran it on indoor point clouds, and compared it to state-of-the-art methods including RIMLS [53], MRPCA [54], and L0P [33]. Figure 3 shows the comparison results. Note that, for better visualization, we adopt the strategy presented in [41] for updating point positions after filtering point normals. Apparently, all these tested methods can effectively filter out noise. However, RIMLS blurs sharp geometric features (e.g., edges and corners), though it recovers smooth transition regions well (see Figure 3c). On the contrary, MRPCA and L0 can preserve sharp geometric features. Nevertheless, L0 inevitably causes staircase artifacts in smooth transition regions (see Figure 3d), and MRPCA tends to over-sharpen smooth features (see Figure 3c). Unlike the above methods, our normal filtering method can simultaneously keep sharp geometric features and smooth transition regions

(see Figure 3e). The comparisons in Figure 3 demonstrate that, our approach outperforms the other ones in handling indoor point clouds, especially those containing sharp features and smooth regions.

Algorithm 1: The iterative algorithm for minimizing problem (3).

Input: Rough point normals $\hat{\mathbf{N}}, \mathbf{N}^{-1} = 0, \mu = 0, \eta = 0, k = 0, \epsilon = 1.0 \times 10^{-6}$.

Output: Filtered point normals \mathbf{N} .

repeat

Minimize \mathbf{N} -subproblem:

Given $(\mu, \eta, \mathbf{X}, \mathbf{Y})$, compute \mathbf{N} according to (6);

Normalize \mathbf{N} .

Minimize \mathbf{X} -subproblem:

Given $(\mu, \eta, \mathbf{N}, \mathbf{Y})$, compute \mathbf{X} according to (8).

Minimize \mathbf{Y} -subproblem:

Given $(\mu, \eta, \mathbf{N}, \mathbf{X})$, compute \mathbf{Y} according to (10).

Update Lagrange multipliers μ, η :

$$\mu^{k+1} = \mu^k + t_e(\mu^k - (D^1 \mathbf{N})^k),$$

$$\eta^{k+1} = \eta^k + t_l(\eta^k - (D^2 \mathbf{N})^k).$$

until $\sum_i \text{disk}(\mathbf{p}_i) \|\mathbf{N}_i^k - \mathbf{N}_i^{k-1}\|^2 < \epsilon$ or $k \geq 100$;

Return \mathbf{N}^k .

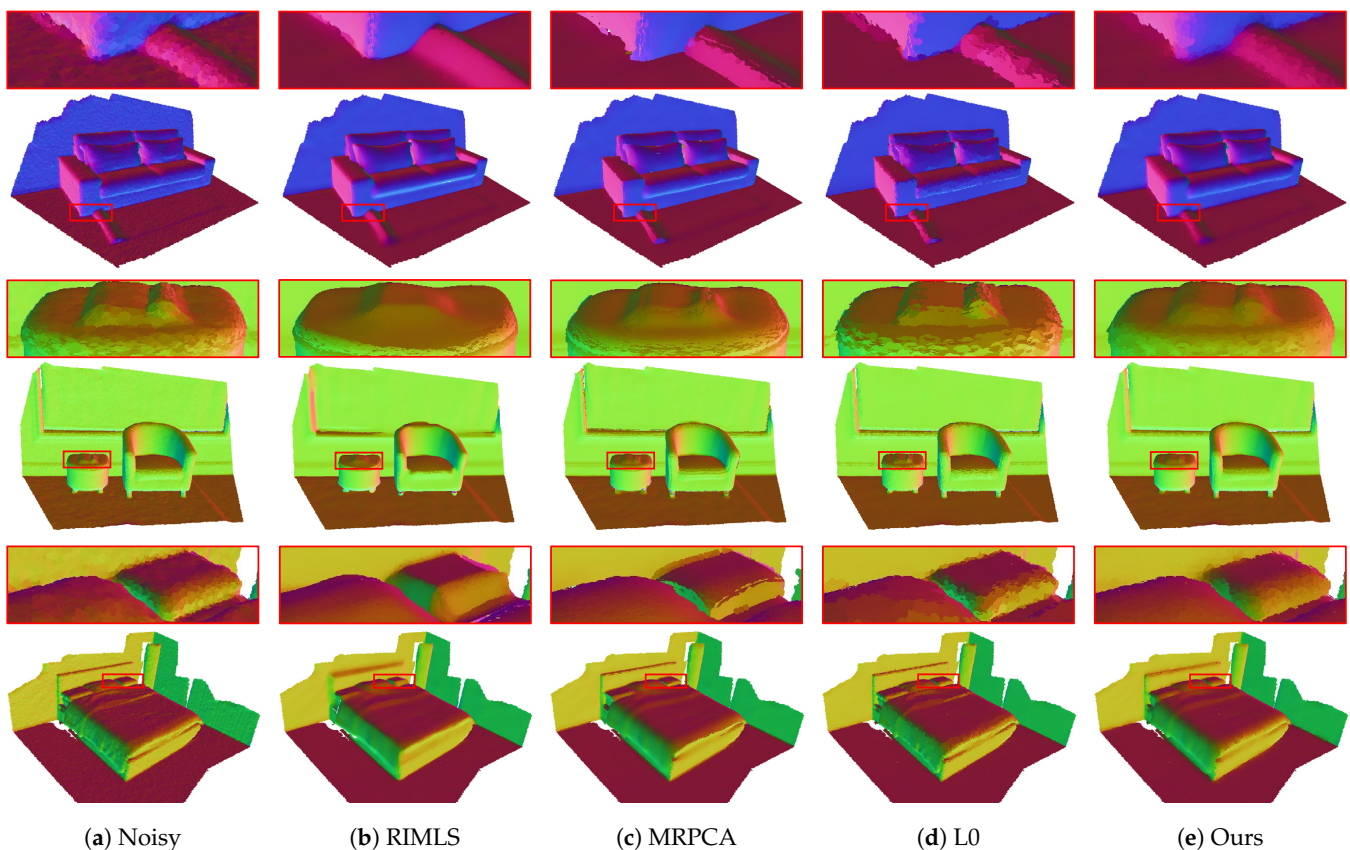


Figure 3. Denoising results of several indoor scene point clouds.

3.2. Correspondence Extraction

As an ingredient for point cloud registration, correspondence point extraction aims at estimating exactly the matching points between a pair of point clouds. In recent years, researchers have developed many remarkable methods to extract correspondence points [7,9,14,16,17]. Nevertheless, since existing methods usually only utilize the ge-

ometry information of the input clouds, they have ambiguity problems when handling point clouds containing noise or weak texture regions. To solve this issue, we design a robust correspondence extraction method using both texture and geometry information. Specifically, our method has two stages, i.e., 3D point descriptors computation followed by point matching. Details of each stage are given as follows.

First, we design a new 3D point descriptor, named TexGeo, based on both texture and geometry information. Formally, given one point $\mathbf{p}_{i,j}$, we can write this descriptor as

$$f(\mathbf{p}_{i,j}) = (cf_{2d}(\mathbf{p}_{i,j}), (1-c)f_{3d}(\mathbf{p}_{i,j})), \quad (11)$$

where $f_{2d}(\mathbf{p}_{i,j})$ is the textural feature term, $f_{3d}(\mathbf{p}_{i,j})$ is the geometric feature term, and $0 \leq c \leq 1$ is the weighting parameter. Here, for efficiency, we adopt the SURF descriptor [55] to extract the textural features and the FPFH descriptor [7] to extract the geometric features. Note that the textural and geometric features are both normalized. In particular, when $c = 0$, the TexGeo descriptor degenerates into the FPFH descriptor; when $c = 1$, the TexGeo descriptor degenerates into the SURF descriptor.

With the TexGeo descriptor, we can robustly extract correspondences between point clouds with repetitive structures, textureless regions, or large holes. Specifically, when the corresponding RGB images are textureless, we can set $c = 0$ to only use geometric features to extract correspondences; when point clouds have repetitive structures, we can set $c = 1$ to only use texture features to extract correspondences. In other cases, we empirically set $c = 0.5$, which means that both texture and geometric features contribute equally to the following point matching process. Moreover, since FPFH descriptors are computed using the filtered point normals, the proposed TexGeo descriptor can accurately describe local regions when point clouds are corrupted by noise.

To compute correspondences of a pair of point clouds, we need first to compute a TexGeo descriptor for each point, and then match all the points in feature space. Here, assume \mathbf{P} to be a point cloud, and $\Gamma(\mathbf{P}) = \{f(\mathbf{p}_{i,j}) : \forall \mathbf{p}_{i,j} \in \mathbf{P}\}$ to be all the computed TexGeo descriptor vectors of \mathbf{P} . In practice, for the input point clouds $\mathbf{P}_a, \mathbf{P}_b$, we can easily compute their descriptor vectors $\Gamma(\mathbf{P}_a)$ and $\Gamma(\mathbf{P}_b)$, respectively. Then, we build two k -D trees, T_a, T_b , by using descriptor vectors $\Gamma(\mathbf{P}_a)$ and $\Gamma(\mathbf{P}_b)$, respectively. Finally, we match all points by the following cross-validation strategy: (i) for each point $\mathbf{p} \in \mathbf{P}_a$, we compute its candidate matching point \mathbf{p}^1 by searching a point from tree T_b , whose descriptor vector is closest to the descriptor vector $f(\mathbf{p})$. Note that we measure the differences between descriptor vectors by using the Euclidean distance; (ii) for the point \mathbf{p}^1 , we also compute its candidate point \mathbf{p}^2 by searching a point from tree T_a in a similar way; (iii) if the point \mathbf{p}^2 and the point \mathbf{p} are the same, \mathbf{p} and \mathbf{p}^1 are matched. After computing all the matched points, we can obtain the correspondence points of the input point clouds.

3.3. Point Clouds Alignment

Let \mathbf{P} and \mathbf{Q} be two point clouds. Here, we aim to compute the rigid transformation information to align these two point clouds. To this end, ICP-like methods iteratively perform the following process: (1) for each point in \mathbf{Q} , compute one correspondence (corresponding point) by finding the closest point from \mathbf{P} ; (2) compute an intermediate transformation by minimizing the energy:

$$E(\mathbf{T}) = \sum_{(\mathbf{p}_i, \mathbf{q}_i)} \|\mathbf{p}_i - \mathbf{T}\mathbf{q}_i\|^2, \quad (12)$$

where $(\mathbf{p}_i, \mathbf{q}_i)$ are two points of the i -th correspondence, which is weighted by $w_i \geq 0$. If the energy converges, the algorithm ends. (3) transform \mathbf{Q} using the above \mathbf{T} , and return to step 1. Given a good enough initial state, we can compute an accurate rigid transformation by ICP. However, since the ICP algorithm needs to update correspondences iteratively, its performance would be greatly reduced when handling large-scale point clouds. Moreover, if the input point clouds are noisy, the ICP algorithm easily converges to a local minimum.

To overcome these limitations, Zhou et al. [23] used a robust penalty function to rewrite energy (12) as:

$$E(\mathbf{T}) = \sum_{(\mathbf{p}_i, \mathbf{q}_i)} g(\|\mathbf{p}_i - \mathbf{T}\mathbf{q}_i\|). \quad (13)$$

where $g(x) = \frac{\mu x^2}{\mu + x^2}$ is the scaled Geman–McClure function, and μ is a positive parameter. Because the Geman–McClure function can automatically validate and prune correspondences, it does not need to recompute correspondence during the optimization. Compared to traditional ICP-like methods, benefiting from the robust penalty function, the method in [23] is robust against noise. However, since the point clouds captured by RGB-D sensors are usually corrupted by large noise, this method may converge to a local minimum. To resolve this problem, instead of using point-to-point distance, we use point-to-plane distance to reformulate the energy (13) as

$$E(\mathbf{T}) = \sum_{(\mathbf{p}_i, \mathbf{q}_i)} g(\|(\mathbf{p}_i - \mathbf{T}\mathbf{q}_i) \cdot \mathbf{N}_{\mathbf{p}_i}\|), \quad (14)$$

where $\mathbf{N}_{\mathbf{p}_i}$ is the filtered normal of point \mathbf{p}_i .

Due to the nonlinearity, it is difficult to minimize energy (14) directly. Inspired by [23], we also present an iterative algorithm by using the method introduced in [56]. First, we introduce the set $\mathbb{L} = \{l_{\mathbf{p}, \mathbf{q}}\}$ as a line process over all the computed correspondence points. The problem (14) can be reformulated as

$$E(\mathbf{T}, \mathbb{L}) = \sum_{\mathbf{p}_i, \mathbf{q}_i} l_{\mathbf{p}_i, \mathbf{q}_i} \|(\mathbf{p}_i - \mathbf{T}\mathbf{q}_i) \cdot \mathbf{N}_{\mathbf{p}_i}\|^2 + \sum_{\mathbf{p}, \mathbf{q}} \psi(l_{\mathbf{p}, \mathbf{q}}), \quad (15)$$

where $\psi(l_{\mathbf{p}, \mathbf{q}}) = \mu(\sqrt{l_{\mathbf{p}, \mathbf{q}}} - 1)^2$. The procedure of minimizing energy (15) can be separated into two subproblems, i.e., the \mathbf{T} - and \mathbb{L} -subproblem. The optimization fixes \mathbb{L} when optimizing \mathbf{T} , and vice versa. We can minimize each subproblem as follows:

(i) \mathbf{T} -subproblem:

$$\min_{\mathbf{T}} \sum_{\mathbf{p}_i, \mathbf{q}_i} l_{\mathbf{p}_i, \mathbf{q}_i} \|(\mathbf{p}_i - \mathbf{T}\mathbf{q}_i) \cdot \mathbf{N}_{\mathbf{p}_i}\|^2. \quad (16)$$

To solve this problem, we first linearize \mathbf{T} as a vector $\zeta = (\alpha, \beta, \gamma, x, y, z)$. As a result, \mathbf{T} can be approximated as a function of ζ :

$$\mathbf{T} \approx \begin{pmatrix} 1 & -\gamma & \beta & x \\ \gamma & 1 & -\alpha & y \\ -\beta & \alpha & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{T}_i, \quad (17)$$

where \mathbf{T}_i is the rigid transformation information used in the previous step, and \mathbf{T}_0 is initialized as \mathbf{I} . Then, the energy function in Equation (16) can be regarded as a least-squares function on ζ , which can be solved as:

$$\mathbf{J}_r^\top \mathbf{J}_r \zeta = -\mathbf{J}_r^\top \mathbf{r}, \quad (18)$$

where \mathbf{r} is a residual vector and \mathbf{J}_r is the Jacobian matrix of \mathbf{r} . After that, the current rigid transformation information \mathbf{T} can be calculated using Equation (17).

(ii) \mathbb{L} -subproblem:

$$\min_{\mathbb{L}} \sum_{\mathbf{p}_i, \mathbf{q}_i} l_{\mathbf{p}_i, \mathbf{q}_i} \|(\mathbf{p}_i - \mathbf{T}\mathbf{q}_i) \cdot \mathbf{N}_{\mathbf{p}_i}\|^2 + \sum_{\mathbf{p}, \mathbf{q}} \psi(l_{\mathbf{p}, \mathbf{q}}). \quad (19)$$

This problem is decomposable spatially. Specifically, for each pair of corresponding points $(\mathbf{p}_i, \mathbf{q}_i)$, we need to solve:

$$\min_{l_{\mathbf{p}_i, \mathbf{q}_i}} l_{\mathbf{p}_i, \mathbf{q}_i} \|(\mathbf{p}_i - \mathbf{T}\mathbf{q}_i) \cdot \mathbf{N}_{\mathbf{p}_i}\| + \psi(l_{\mathbf{p}_i, \mathbf{q}_i}),$$

which has a closed form solution:

$$l_{\mathbf{p}_i, \mathbf{q}_i} = \left(\frac{\mu}{\mu + \|(\mathbf{p}_i - \mathbf{T}\mathbf{q}_i) \cdot \mathbf{N}_{\mathbf{p}_i}\|^2} \right)^2. \quad (20)$$

Similar to [23], to produce good results, we initialize $\mu = \mathcal{D}^2$, where \mathcal{D} is the diameter of the largest surface. During the optimization procedure, we also decrease μ in each iteration until $\mu = \delta^2$, where δ is the threshold for pruning correspondences. We outline the whole algorithm in Algorithm 2.

Algorithm 2: Robust rigid transformation computation.

Input: Point clouds (\mathbf{P}, \mathbf{Q}) , precomputed correspondences \mathcal{C} .

Output: Rigid transformation matrix \mathbf{T} that registers \mathbf{Q} to \mathbf{P} .

$\mathbf{T} = \mathbf{I}, \mu = \mathcal{D}^2$.

repeat

for $(\mathbf{p}_i, \mathbf{q}_i) \in \mathcal{C}$ **do**

 Compute $l_{\mathbf{p}_i, \mathbf{q}_i}$ by (20).

 Update \mathbf{J}_r and \mathbf{r} of energy (15).

end

 Solve Equation (18) and calculate \mathbf{T} using (17).

 Every four iterations, $\mu = \mu \times 0.5$;

until Converged or $\mu < \delta^2$;

4. Experimental Results

We tested the effectiveness of our approach on real-scanned indoor scene point clouds, which were generated from RGB-D images. Note that the RGB-D images used in this paper were derived from open-source datasets [49,57]. Moreover, the size of these images was 640×480 uniformly. The example RGB images of the tested point clouds are listed in Figure 4. Some of the tested point clouds are corrupted by zero-mean Gaussian noise with standard deviation proportional to the σ times the diagonal length of the minimum bounding box of the point cloud. We also give visual and numerical comparisons of our point cloud registration method with existing approaches including PCL [7], S4PCS [44], GICP [58], GICPT [58], and FGR [11]. For our method, we implemented it using C++. For the other compared methods, we used the source codes kindly provided by their authors.

4.1. Qualitative Comparison

We qualitatively compared our point cloud registration method to the other methods on indoor scene point clouds. During this process, we generated visually appealing results by carefully tuning their parameters. Note that we performed the comparison on indoor scene point clouds corrupted by synthetic noise followed by point clouds contaminated by real noise.



Figure 4. The example RGB images of six indoor scene point clouds named Lr1, Lr2, Lr3, Of1, Of2, and Teddy, respectively.

Figure 5 shows a comparison of a pair of indoor scene point clouds Lr1, which have a small rotation and translation between them. Note that these two point clouds are corrupted by a relatively small noise. As we can see, the results of GICP and GICPT have a comparatively large translation between the two point clouds (Figure 5c,d). The reason is that these two methods cannot accurately extract correspondences in the presence of noise. Moreover, although methods PCL, S4PCS, and FGR compute point features to construct the correspondences, their results still have small translations between the two point clouds (Figure 5b,e,f). The reason is that these methods produce some wrong correspondences between two input point clouds. On the contrary, our method generates the visually best registration result; see Figure 5g.

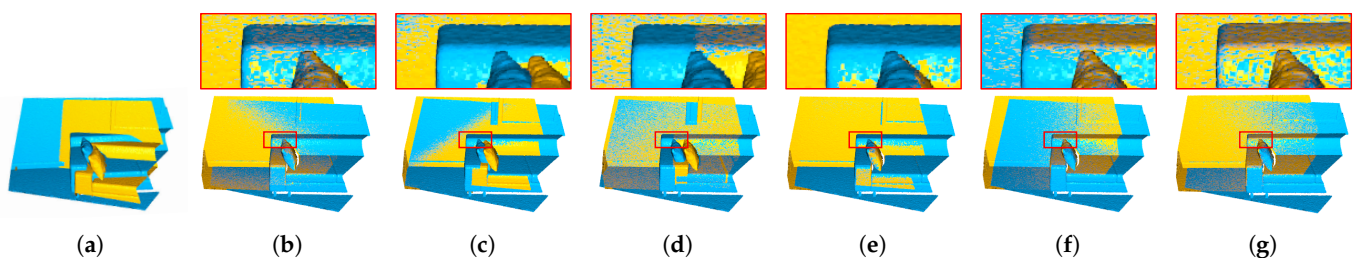


Figure 5. Registration results of Lr1 ($\sigma = 0.2\%d$) which have small rotation and translation between them. (a) Input. (b) PCL. (c) GICP. (d) GICPT. (e) S4PCS. (f) FGR. (g) Our approach.

Figure 6 demonstrates the registration results of indoor scene point clouds Lr2 with a large translation and rotation. As can be seen, all the tested methods can produce the right rotation information. However, due to the noise interference, the results of GICP and GICPT have a relatively large translation between the two point clouds; see Figure 6c,d. Apart from that, from Figure 6b,e,f,g we can see that PCL, S4PCS, FGR, and our approach generate visually good registration results. However, the numerical comparison in Table 1 shows that the registration error of our method is the smallest one. Thus, our method outperforms the others.

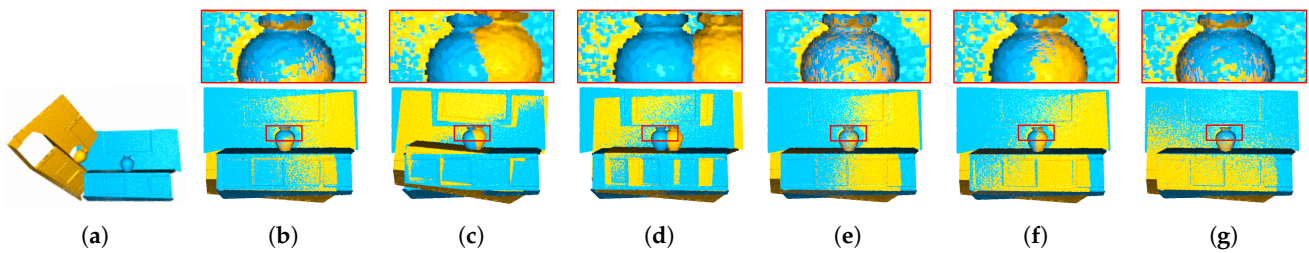


Figure 6. Registration results of Lr2 ($\sigma = 0.4\%d$) with both comparatively large translation and rotation between them. (a) Input. (b) PCL. (c) GICP. (d) GICPT. (e) S4PCS. (f) FGR. (g) Our approach.

Figure 7 shows the comparisons on a pair of indoor scene point clouds Lr3, which has large textureless regions. As can be seen, PCL, GICP, GICPT, and S4PCS do not generate satisfactory results (Figure 7b–e). Besides, FGR produces a better registration result due to its robustness against noise. However, the result of FGR still has a small translation between the two point clouds (Figure 7f). Compared to them, our approach can align two point clouds more exactly (Figure 7g).

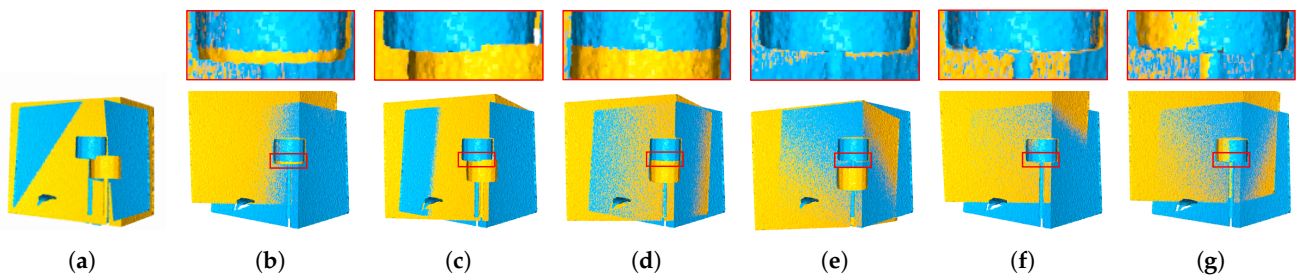


Figure 7. Registration results of indoor scene point clouds Lr3 ($\sigma = 0.4\%d$) with rich geometric features and large textureless regions. (a) Input. (b) PCL. (c) GICP. (d) GICPT. (e) S4PCS. (f) FGR. (g) Our approach.

Figure 8 shows comparison results on indoor scene point clouds Of1, which has a large translation and a large rotation between them. As can be seen, GICP and GICPT fail to obtain the right rotation information (Figure 8c,d). PCL can get the right rotation information, but its result has a relatively large translation between the two point clouds (Figure 8b). Similarly, although both S4PCS and FGR can produce the right rotation information, their results still have a small translation between the two point clouds (Figure 8e,f). In contrast, from Figure 8g, we can see that our method not only generates the right rotation information, but also effectively alleviates the translation between the input point clouds.

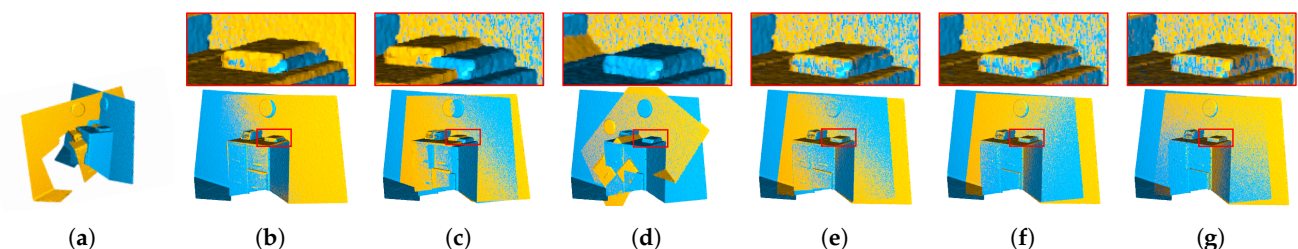


Figure 8. Registration results of indoor scene point clouds Of1 ($\sigma = 0.2\%d$) with both large translation and rotation between two input point clouds. (a) Input. (b) PCL. (c) GICP. (d) GICPT. (e) S4PCS. (f) FGR. (g) Our approach.

Figure 9 shows the registration results of indoor scene point clouds Of2 consisting of local regions from which it is difficult to extract geometric features. In other words, this indoor scene lacks geometric features. As we can see, GICPT generates the wrong translation and rotation information (Figure 9c). Except GICPT, the other methods register

input point clouds well (Figure 9b,e,f,g). Moreover, from the comparison presented in the next subsection, we can find that our method is the best one with the least registration error.

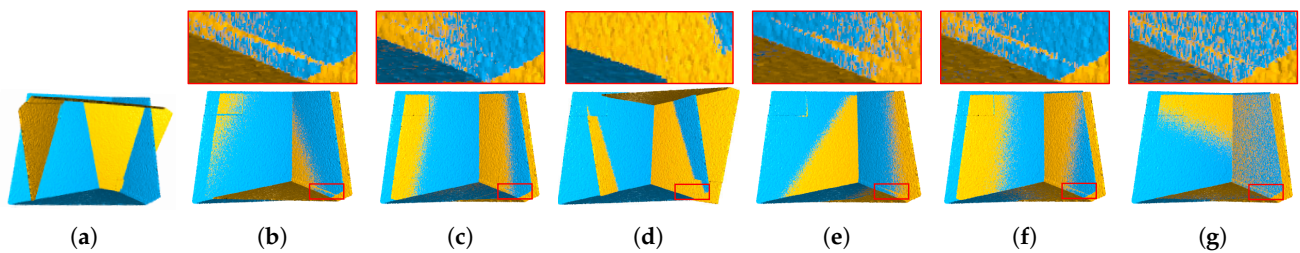


Figure 9. Registration results of indoor scene point clouds Of2 ($\sigma = 0.4\% \bar{d}$) with few geometric features and rich textures. (a) Input. (b) PCL. (c) GICP. (d) GICPT. (e) S4PCS. (f) FGR. (g) Our approach.

Figure 10 shows registration results of indoor scene point clouds Teddy, containing a comparatively large real noise. Note that these point clouds also contain holes, because of the scene occlusion problem. As we can see, due to the noise interference, the results of GICP and S4PCS have relatively large translations between the corresponding transformed point cloud and target point cloud (Figure 10c,e). Moreover, PCL, GICPT, FGR, and our approach can effectively align the input point clouds (Figure 10b,d,f,g). However, the numerical comparison listed in Table 1 shows that our method has the smallest registration error.

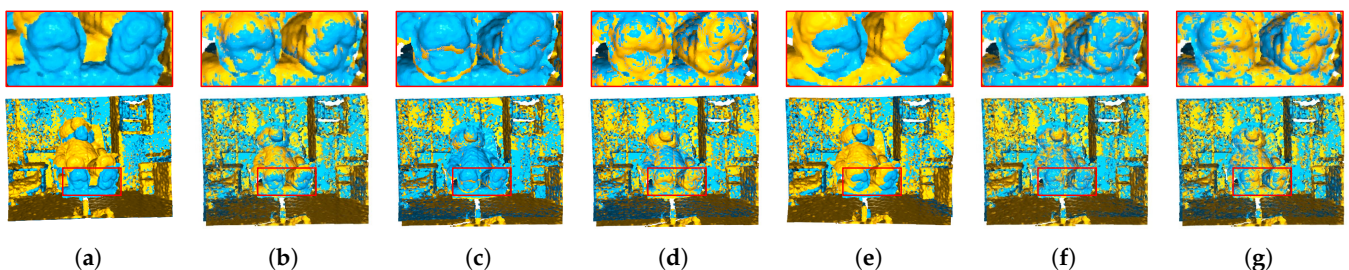


Figure 10. Registration results of point clouds Teddy, which are corrupted by real noise. (a) Input. (b) PCL. (c) GICP. (d) GICPT. (e) S4PCS. (f) FGR. (g) Our approach.

Figure 11 compares the registration results produced by our framework and the recent method abbreviated as SymICP, which is presented in [46]. As can be seen, both methods can produce visually good registration results (Figure 11b,c). However, from the zoomed views, we can find that our method achieves a higher registration accuracy than SymICP.

As a result, given the fact that indoor scene point clouds tend to be noisy, contain large textureless regions, or contain regions from which it is difficult to extract geometric features, it is challenging for the compared state-of-the-art methods to achieve desirable registration results from these low-quality indoor scene point clouds. One of the key reasons is that these tested methods cannot accurately estimate correspondences between indoor scene point clouds. Compared to the other methods, our method based on filtered point normals is robust against noise. Moreover, by using the TexGeo descriptor that combines both geometry and texture information, our method can effectively process point clouds with holes or large textureless regions. Therefore, our method outperforms the other compared methods when processing low-quality indoor scene point clouds.

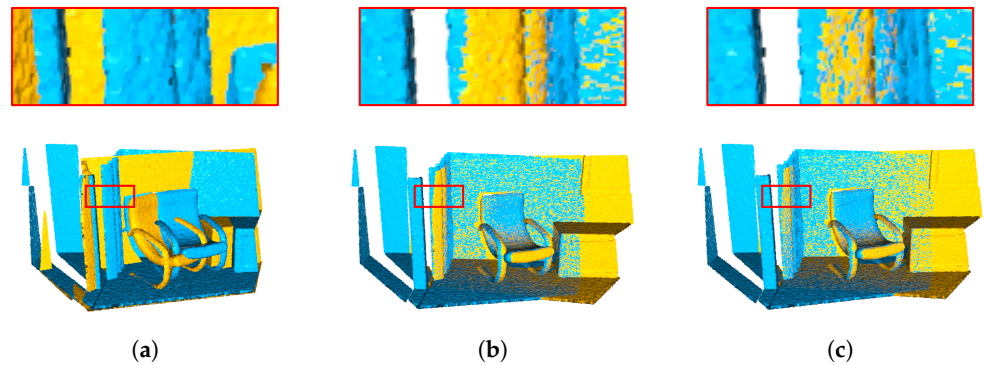


Figure 11. Comparison of registration results produced by SymICP and our approach. (a) input; (b) registration result produced by SymICP; (c) registration result produced by our approach.

Table 1. Quantitative evaluation results of the tested point cloud registration methods.

Point Clouds	RMSE ($\times 10^{-3}$)					
	PCL	GICP	GICPT	S4PCS	FGR	Our Approach
Lr1	2.800	4.791	2.658	4.275	2.614	2.526
Lr2	3.067	4.747	3.691	3.566	2.917	2.613
Lr3	4.485	5.509	3.352	3.492	3.614	3.202
Of1	3.006	2.715	3.049	2.218	2.070	1.869
Of2	5.013	4.584	5.547	5.043	4.283	3.935
Teddy	5.893	6.356	5.767	6.545	5.710	5.661

4.2. Quantitative Comparison

The above qualitative comparisons show that our method is capable of producing visually better registration results when compared to the other selected methods. Furthermore, to objectively assess our method, we further quantitatively compared it to the other methods on point clouds corrupted by synthetic noise. To achieve this, we used an error metric E_{reg} , which measures the average point-to-plane distance of all pairs of corresponding points. Specifically, assume the number of correspondences is M and the obtained rigid transformation matrix is \mathbf{T} , the metric E_{reg} is formulated as:

$$E_{reg} = \frac{1}{M} \sum_{i=1}^M \|(\mathbf{p}_i - \mathbf{T}\mathbf{q}_i) \cdot \mathbf{N}_{\mathbf{p}_i}\|^2 \quad (21)$$

where $\mathbf{p}_i, \mathbf{q}_i$ are two corresponding points. We computed E_{reg} for the examples shown in Figures 5–10, and listed the errors in Table 1. Obviously, our approach has the smallest errors for all the examples, which demonstrate that our approach quantitatively outperforms the other compared ones.

4.3. Ablation Study

To clearly show the contribution of each stage of the proposed framework, we further conducted a series of controlled experiments that tested the specific decision in each stage of the proposed framework. These studies were also performed on the above-mentioned data sets.

We first conducted an ablation study on the point normal estimation. Figure 12 compares the registration results produced using noisy point normals and estimated point normals. As can be seen, the registration result produced using estimated point normals is much better than the registration result produced using noisy point normals (Figure 12b,c). This suggests that the point normal estimation is essential in the proposed framework.

Then, we conducted an ablation study on the correspondence extraction. In this study, we tested our method using two 3D point descriptors including the FPFH descriptor and the proposed TexGeo descriptor. Figure 13 compares the registration results produced using these descriptors. As can be seen, our framework cannot produce the desired result when replacing the proposed TexGeo descriptor by the FPFH descriptor (Figure 13b,c). This means the proposed correspondence extraction is important to achieve a robust registration of indoor scene point clouds.

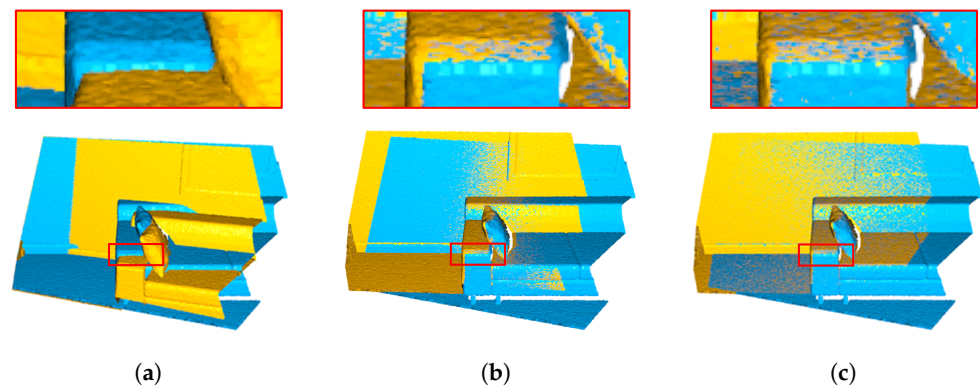


Figure 12. Ablation of point normal estimation. From left to right: (a) input; (b) registration result produced using noisy point normals; (c) registration result produced using estimated point normals.

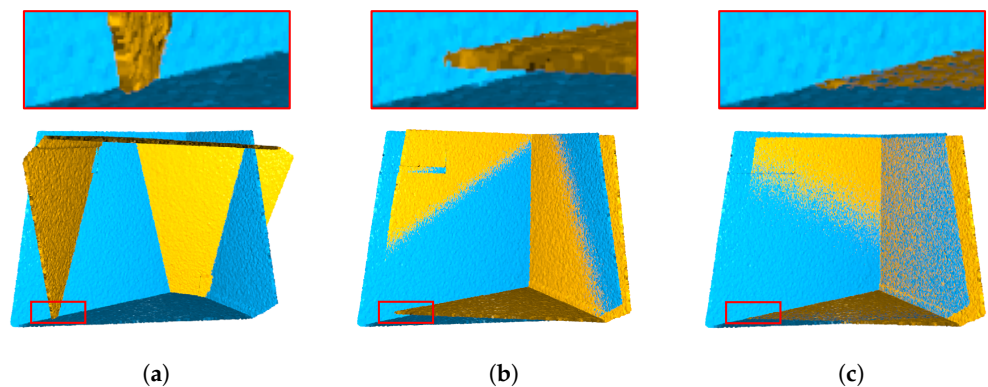


Figure 13. Ablation of correspondence extraction. (a) input; (b) registration result produced using the FPFH descriptor; (c) registration result produced using the proposed TexGeo descriptor.

Finally, we conducted an ablation study on the transformation computation scheme. In this study, we tested our framework using the ICP scheme and our computation method. Figure 14 shows the registration results. As can be seen, given the extracted correspondences, the ICP algorithm cannot produce the exact rigid transformation matrix for aligning input point clouds (Figure 14b). On the contrary, our computation method is capable of generating the accurate rigid transformation matrix (Figure 14c). The reason is that our computation method is robust against those remaining false correspondences. Thus, the proposed transformation computation scheme is important to the framework.

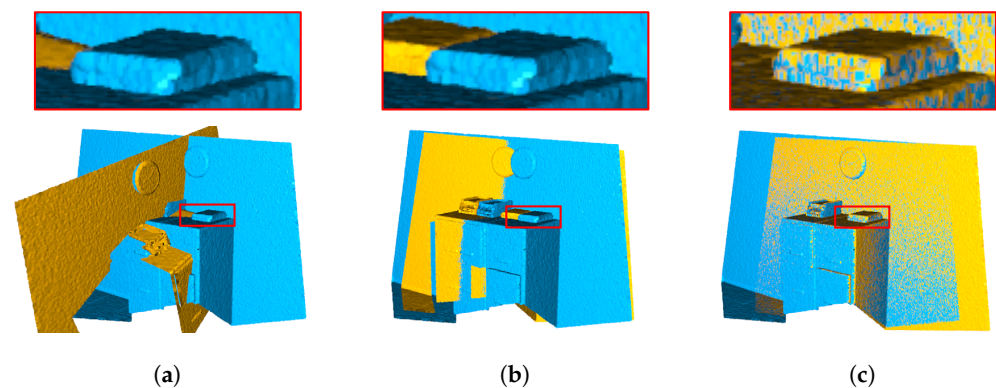


Figure 14. Ablation of rigid transformation computation scheme. (a) input; (b) registration result produced using the ICP algorithm; (c) registration result produced using our approach.

5. Discussion

In this section, we further discussed the main features of the proposed method based on its performances in the above experiments.

(1) 3D indoor scene point clouds scanned by RGB-D cameras are of low quality. To this end, existing point cloud registration approaches are usually designed based on the RANSAC scheme or some robust estimators. From the experiments, we found that an accurate point normal field can be used to further improve the registration accuracy. The reason is that accurate point normals are not only useful for constructing point descriptors, but also help the rigid transformation computation.

(2) The distinctiveness of 3D point features computed using common geometry descriptors is low in indoor scenes. The 2D features extracted from corresponding RGB images can improve the distinctiveness of the 3D points. From the experiments, we found that, with the help of the proposed descriptor, our method still has a good performance when point clouds contain large-scale noise or weak texture features. This is because both geometry and texture information can be complementary to each other.

(3) The rigid transformation computation using a nonconvex regularizer is important for accurate point cloud registration because it can further prune some false correspondences remaining from the first two stages. Moreover, the point-to-plane distance metric can enhance the registration accuracy.

(4) The proposed registration framework is flexible. First, the method used in each stage can be replaced by more advanced approaches to pursue a higher accuracy of indoor scene point cloud registration. Second, the framework can be adapted to process point clouds from other sources, e.g., laser-scanned point clouds.

6. Conclusions

This work presented a robust registration framework of indoor scene point cloud using RGB-D information, which consisted of three stages. Specifically, we first introduced a point normal filtering model combining the total variation with a second-order variation, which can effectively filter out noise while simultaneously maintaining sharp geometric features and nonlinear smooth transition regions. Then, we designed a novel point feature descriptor (TexGeo), encoding both texture and geometry information. By using this descriptor, we were able to robustly establish correspondences between a pair of point clouds, even when dealing with large noise or encountering local regions that are difficult to extract geometric features. Finally, we utilized a global optimization method to efficiently compute rigid transformations between those pairs. Intensive registration results verified that our method outperformed the state-of-the-art methods visually and numerically. In addition, our method is robust against noise, textureless regions, and the regions from which it is difficult to extract geometric features.

There are still some open problems. For example, many parameters in our method need to be tuned manually, which may be tedious. Moreover, when dealing with large-scale scene point clouds, our method seems to be time-consuming. The strategy of speeding up our method can be thoroughly explored in the future.

Author Contributions: Methodology, S.Z. and Z.L.; software, R.L.; validation, R.L.; investigation, S.Z.; writing—original draft preparation, S.Z.; writing—review and editing, Z.L. and M.G.; visualization, R.L.; supervision, Z.L. and Z.X.; project administration, S.Z. and Z.L.; funding acquisition, M.G., J.C. and Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Nos. U1711267, 41671400, 41971356 and 41701446), National Key R&D Program of China (Nos. 2017YCF0601500 and 2017YFC0601504), and the Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources (KF-2020-05-011).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in [59,60].

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

Abbreviations

The following abbreviations are used in this manuscript:

$\mathcal{L}(i, j)$	The auxiliary line connecting point $\mathbf{p}_{i,j}$ with some midpoint
$len(\cdot)$	The length of (\cdot)
$disk(\cdot)$	The area of (\cdot)
D^1	The first-order operator
D^2	The second-order operator
RIMLS	Robust implicit moving least squares
MRPCA	Moving robust principal components analysis
L0P	Denoising point sets via ℓ_0 minimization
PCL	A point cloud library implementation of Rusu et al. [7]
S4PCS	Super 4pcs fast global point cloud registration via smart indexing
GICP	Go-ICP: a globally optimal solution to 3D ICP point-set registration
GICPT	A trimming variant of GICP
FGR	Fast global registration
SymICP	A symmetric objective function for ICP

References

1. Pavan, N.L.; dos Santos, D.R.; Khoshelham, K. Global registration of terrestrial laser scanner point clouds using plane-to-plane correspondences. *Remote Sens.* **2020**, *12*, 1127. [\[CrossRef\]](#)
2. Cui, Y.; Chang, W.; Nöll, T.; Stricker, D. KinectAvatar: Fully automatic body capture using a single kinect. In Proceedings of the Asian Conference on Computer Vision, Daejeon, Korea, 5–9 November 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 133–147.
3. Globally consistent registration of terrestrial laser scans via graph optimization. *ISPRS J. Photogramm. Remote Sens.* **2015**, *109*, 126–138. [\[CrossRef\]](#)
4. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In Proceedings of the Sensor Fusion IV: Control Paradigms and data Structures, Boston, MA, USA, 14–15 November 1992; International Society for Optics and Photonics: Bellingham, WA, USA, 1992; Volume 1611, pp. 586–606.
5. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011; pp. 127–136.
6. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
7. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3212–3217.

8. Gelfand, N.; Mitra, N.J.; Guibas, L.J.; Pottmann, H. Robust Global Registration. In Proceedings of the Third Eurographics Symposium on Geometry Processing, Vienna, Austria, 4–6 July 2005.
9. Takimoto, R.Y.; Tsuzuki, M.d.S.G.; Vogelaar, R.; de Castro Martins, T.; Sato, A.K.; Iwao, Y.; Gotoh, T.; Kagei, S. 3D reconstruction and multiple point cloud registration using a low precision RGB-D sensor. *Mechatronics* **2016**, *35*, 11–22. [[CrossRef](#)]
10. Endres, F.; Hess, J.; Engelhard, N.; Sturm, J.; Cremers, D.; Burgard, W. An evaluation of the RGB-D SLAM system. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 1691–1696.
11. Henry, P.; Krainin, M.; Herbst, E.; Ren, X.; Fox, D. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Experimental Robotics*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 477–491.
12. Guo, M.; Wu, L.; Huang, Y.; Chen, X. An efficient internet map tiles rendering approach on high resolution devices. *J. Spat. Sci.* **2021**, 1–19. [[CrossRef](#)]
13. Wan, T.; Du, S.; Cui, W.; Yang, Y.; Li, C. Robust Rigid Registration Algorithm Based on Correntropy and Bi-Directional Distance. *IEEE Access* **2020**, *8*, 22225–22234. [[CrossRef](#)]
14. Wan, T.; Du, S.; Cui, W.; Yao, R.; Ge, Y.; Li, C.; Gao, Y.; Zheng, N. RGB-D Point Cloud Registration Based on Salient Object Detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–13. [[CrossRef](#)] [[PubMed](#)]
15. Aiger, D.; Mitra, N.J.; Cohen-Or, D. 4-points congruent sets for robust pairwise surface registration. *ACM Trans. Graphics* **2008**, *27*, 1–10. [[CrossRef](#)]
16. Chen, S.; Nan, L.; Xia, R.; Zhao, J.; Wonka, P. PLADE: A plane-based descriptor for point cloud registration with small overlap. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 2530–2540. [[CrossRef](#)]
17. Zhang, L.; Guo, J.; Cheng, Z.; Xiao, J.; Zhang, X. Efficient Pairwise 3-D Registration of Urban Scenes via Hybrid Structural Descriptors. *IEEE Trans. Geosci. Remote. Sens.* **2021**, 1–17. [[CrossRef](#)]
18. Bylow, E.; Sturm, J.; Kerl, C.; Kahl, F.; Cremers, D. Real-time camera tracking and 3D reconstruction using signed distance functions. In *Robotics: Science and systems (RSS)*; Robotics: Science and Systems: Berlin, Germany, 2013; Volume 2, p. 2.
19. Pavlov, A.L.; Ovchinnikov, G.W.; Derbyshev, D.Y.; Tsetserukou, D.; Oseledets, I.V. AA-ICP: Iterative closest point with Anderson acceleration. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3407–3412.
20. Zhang, J.; Yao, Y.; Deng, B. Fast and Robust Iterative Closest Point. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, 1. [[CrossRef](#)] [[PubMed](#)]
21. Chetverikov, D.; Stepanov, D.; Krsek, P. Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm. *Image Vis. Comput.* **2005**, *23*, 299–309. [[CrossRef](#)]
22. Bouaziz, S.; Tagliasacchi, A.; Pauly, M. Sparse iterative closest point. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2013; Volume 32; pp. 113–123.
23. Zhou, Q.Y.; Park, J.; Koltun, V. Fast global registration. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 766–782.
24. Wu, Z.; Chen, H.; Du, S.; Fu, M.; Zhou, N.; Zheng, N. Correntropy based scale ICP algorithm for robust point set registration. *Pattern Recognit.* **2019**, *93*, 14–24. [[CrossRef](#)]
25. Yang, J.; Li, H.; Jia, Y. Go-icp: Solving 3d registration efficiently and globally optimally. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1457–1464.
26. Feng, W.; Zhang, J.; Cai, H.; Xu, H.; Hou, J.; Bao, H. Recurrent Multi-view Alignment Network for Unsupervised Surface Registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 10297–10307.
27. Li, X.; Pontes, J.K.; Lucey, S. PointNetLK Revisited. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 12763–12772.
28. 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, 27–28 October 2019; pp. 3523–3532.
29. Guo, M.; Liu, H.; Xu, Y.; Huang, Y. Building extraction based on U-Net with an attention block and multiple losses. *Remote Sens.* **2020**, *12*, 1400. [[CrossRef](#)]
30. Guo, M.; Yu, Z.; Xu, Y.; Huang, Y.; Li, C. ME-Net: A Deep Convolutional Neural Network for Extracting Mangrove Using Sentinel-2A Data. *Remote Sens.* **2021**, *13*, 1292. [[CrossRef](#)]
31. Tam, G.K.; Cheng, Z.Q.; Lai, Y.K.; Langbein, F.C.; Liu, Y.; Marshall, D.; Martin, R.R.; Sun, X.F.; Rosin, P.L. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graph.* **2012**, *19*, 1199–1217. [[CrossRef](#)]
32. Avron, H.; Sharf, A.; Greif, C.; Cohen-Or, D. ℓ_1 -sparse reconstruction of sharp point set surfaces. *ACM Trans. Graph. TOG* **2010**, *29*, 1–12. [[CrossRef](#)]
33. Sun, Y.; Schaefer, S.; Wang, W. Denoising point sets via L0 minimization. *Comput. Aided Geom. Des.* **2015**, *35–36*, 2–15. [[CrossRef](#)]
34. Zhong, S.; Xie, Z.; Wang, W.; Liu, Z.; Liu, L. Mesh denoising via total variation and weighted Laplacian regularizations. *Comput. Animat. Virtual Worlds* **2018**, *29*, e1827. [[CrossRef](#)]
35. Zhong, S.; Xie, Z.; Liu, J.; Liu, Z. Robust Mesh Denoising via Triple Sparsity. *Sensors* **2019**, *19*, 1001. [[CrossRef](#)]
36. Liu, Z.; Zhong, S.; Xie, Z.; Wang, W. A novel anisotropic second order regularization for mesh denoising. *Comput. Aided Geom. Des.* **2019**, *71*, 190–201. [[CrossRef](#)]

37. Liu, Z.; Wang, W.; Zhong, S.; Zeng, B.; Liu, J.; Wang, W. Mesh Denoising via a Novel Mumford–Shah Framework. *Comput.-Aided Des.* **2020**, *126*, 102858. [[CrossRef](#)]
38. Guo, M.; Song, Z.; Han, C.; Zhong, S.; Lv, R.; Liu, Z. Mesh denoising via adaptive consistent neighborhood. *Sensors* **2021**, *21*, 412. [[CrossRef](#)] [[PubMed](#)]
39. Zhong, S.; Song, Z.; Liu, Z.; Xie, Z.; Chen, J.; Liu, L.; Chen, R. Shape-aware Mesh Normal Filtering. *Comput.-Aided Des.* **2021**, *140*, 103088. [[CrossRef](#)]
40. Liu, Z.; Li, Y.; Wang, W.; Liu, L.; Chen, R. Mesh Total Generalized Variation for Denoising. *IEEE Trans. Vis. Comput. Graph.* **2021**, *1*. [[CrossRef](#)] [[PubMed](#)]
41. Liu, Z.; Xiao, X.; Zhong, S.; Wang, W.; Li, Y.; Zhang, L.; Xie, Z. A feature-preserving framework for point cloud denoising. *Comput.-Aided Des.* **2020**, *127*, 102857. [[CrossRef](#)]
42. Raguram, R.; Frahm, J.M.; Pollefeys, M. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 500–513.
43. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
44. Mellado, N.; Aiger, D.; Mitra, N.J. Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2014; Volume 33; pp. 205–215.
45. Mavridis, P.; Andreadis, A.; Papaioannou, G. Efficient Sparse Icp. *Comput. Aided Geom. Des.* **2015**, *35*, 16–26. [[CrossRef](#)]
46. Rusinkiewicz, S. A symmetric objective function for ICP. *ACM Trans. Graph. TOG* **2019**, *38*, 1–7. [[CrossRef](#)]
47. Diebel, J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix* **2006**, *58*, 1–35.
48. Pan, H.; Guan, T.; Luo, Y.; Duan, L.; Tian, Y.; Yi, L.; Zhao, Y.; Yu, J. Dense 3D reconstruction combining depth and RGB information. *Neurocomputing* **2016**, *175*, 644–651. [[CrossRef](#)]
49. Handa, A.; Whelan, T.; McDonald, J.; Davison, A.J. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 1524–1531.
50. Dai, W.; Zhang, Y.; Li, P.; Fang, Z.; Scherer, S. RGB-D SLAM in Dynamic Environments Using Point Correlations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *1*. [[CrossRef](#)]
51. Sipiran, I.; Bustos, B. Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes. *Vis. Comput.* **2011**, *27*, 963–976. [[CrossRef](#)]
52. Wang, P.S.; Liu, Y.; Tong, X. Mesh Denoising via Cascaded Normal Regression. *ACM Trans. Graph.* **2016**, *35*, 232:1–232:12. [[CrossRef](#)]
53. Öztireli, A.C.; Guennebaud, G.; Gross, M. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2009; Volume 28; pp. 493–501.
54. Mattei, E.; Castrodad, A. Point cloud denoising via moving RPCA. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2017; Volume 36; pp. 123–137.
55. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
56. Black, M.J.; Rangarajan, A. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vis.* **1996**, *19*, 57–91. [[CrossRef](#)]
57. Park, J.; Zhou, Q.Y.; Koltun, V. Colored Point Cloud Registration Revisited. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 143–152.
58. Yang, J.; Li, H.; Campbell, D.; Jia, Y. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 2241–2254. [[CrossRef](#)] [[PubMed](#)]
59. Handa, A.; Whelan, T.; McDonald, J.B.; Davison, A.J. 2014. The ICL-NUIM Dataset. Available online: <http://www.doc.ic.ac.uk/~ahanda/VaFRIC/iclnuim.html> (accessed on 23 September 2021).
60. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. 2012. The TUM Dataset. Available online: <https://vision.in.tum.de/data/datasets/rgbd-dataset/download> (accessed on 23 September 2021).