applied sciences

MDPI

Article

Real-Time RGB-D Simultaneous Localization and Mapping Guided by Terrestrial LiDAR Point Cloud for Indoor 3-D Reconstruction and Camera Pose Estimation

Xujie Kang [1,2], Jing Li [1,*], Xiangtao Fan [1] and Wenhui Wan [3]

[1] Key Laboratory of Digital Earth Science, Aerospace Information Research Institute (AIR), Chinese Academy of Sciences, Beijing 100094, China

[2] University of Chinese Academy of Sciences, Beijing 100049, China

[3] State Key Laboratory of Remote Sensing Science, Aerospace Information Research Institute (AIR), Chinese Academy of Sciences, No. 20A, Datun Road, Chaoyang District, Beijing 100101, China

* Correspondence: lijing@radi.ac.cn

Received: 14 July 2019; Accepted: 6 August 2019; Published: 9 August 2019

check for updates

Featured Application: Robotics, Augmented Reality and self-driving.

Abstract: In recent years, low-cost and lightweight RGB and depth (RGB-D) sensors, such as Microsoft Kinect, have made available rich image and depth data, making them very popular in the field of simultaneous localization and mapping (SLAM), which has been increasingly used in robotics, self-driving vehicles, and augmented reality. The RGB-D SLAM constructs 3D environmental models of natural landscapes while simultaneously estimating camera poses. However, in highly variable illumination and motion blur environments, long-distance tracking can result in large cumulative errors and scale shifts. To address this problem in actual applications, in this study, we propose a novel multithreaded RGB-D SLAM framework that incorporates a highly accurate prior terrestrial Light Detection and Ranging (LiDAR) point cloud, which can mitigate cumulative errors and improve the system's robustness in large-scale and challenging scenarios. First, we employed deep learning to achieve system automatic initialization and motion recovery when tracking is lost. Next, we used terrestrial LiDAR point cloud to obtain prior data of the landscape, and then we applied the point-to-surface inductively coupled plasma (ICP) iterative algorithm to realize accurate camera pose control from the previously obtained LiDAR point cloud data, and finally expanded its control range in the local map construction. Furthermore, an innovative double window segment-based map optimization method is proposed to ensure consistency, better real-time performance, and high accuracy of map construction. The proposed method was tested for long-distance tracking and closed-loop in two different large indoor scenarios. The experimental results indicated that the standard deviation of the 3D map construction is 10 cm in a mapping distance of 100 m, compared with the LiDAR ground truth. Further, the relative cumulative error of the camera in closed-loop experiments is 0.09%, which is twice less than that of the typical SLAM algorithm (3.4%). Therefore, the proposed method was demonstrated to be more robust than the ORB-SLAM2 algorithm in complex indoor environments.

Keywords: SLAM; LiDAR; RGB-D camera; deep learning

1. Introduction

Simultaneous localization and mapping (SLAM) has been rapidly developing, and it is being widely used in the simultaneous estimation of the pose of a moving platform and the building of a map of the surrounding environment [1–3]. Consequently, SLAM has become a key prerequisite for unmanned systems to operate autonomously in unknown environments. In known environments [4,5], the localization part of SLAM can provide a six-degree-of-freedom sensor pose estimation, and thus plays an essential role in positioning and navigation applications such as navigation and planetary exploration missions, especially in environments where the global navigation satellite system (GNSS) is unavailable. Moreover, recent developments in SLAM have been driven by advances in sensor technology (e.g., RGB and depth (RGB-D) cameras and mobile Light Detection and Ranging (LiDAR) scanners) [6–8], as well as by the potential large market opportunities in emerging applications such as mobile augmented reality, self-driving vehicles, unmanned aerial vehicles, and home robots. SLAM has been extensively researched over the past 30 years by the robotics and computer vision community, and thus a comprehensive review on this topic is out of the scope of this paper; instead, we focus on the most relevant literature applicable to this research. First, excellently robust and quick feature extraction techniques such as SIFT [9], SURF [10], BRISK [11], and ORB [12] that are to be used in the SLAM framework are developed. PTAM [13] incorporated parallel computing in small-scale Augmented Reality (AR) tracking and mapping to improve efficiency, and was the first multithreaded approach to separate localization and mapping into two different threads. To further improve the accuracy and robustness, many SLAM algorithms utilize loop closure detection using the bag-of-words model [14]. Once a loop closure is successfully identified, the cumulative tracking error can be significantly reduced. In recent years, in indoor environments without GPS coverage, visual SLAM has become an important method for self-localization and environmental sensing [15]. Visual SLAM uses a monocular camera [16,17], stereo cameras [18,19], and RGB-D cameras [20] to construct various forms of SLAM systems.

With people increasingly spending most time indoors, the necessity for indoor positioning and mapping of large scenes has been increasing. However, without prior data, the cumulative error in both mapping and camera pose increases with the tracking distance. This leads to low mapping precision, and thus new technologies that rely on high-precision indoor positioning and mapping, such as mobile augmented reality, self-driving vehicles, path planning, and obstacle avoidance in service robots, are almost rendered unusable. To address this problem, this study is focused on eliminating the cumulative errors by fully utilizing the information acquired by the RGB and depth (RGB-D) sensors by incorporating high-precision prior LiDAR data. In addition, a multithreaded segment-based map double window optimization strategy is adopted to construct a consistent and accurate map in real-time and deep learning is utilized in indoor complex environments. Therefore, this research finally improves the robustness and flexibility of the RGB-D SLAM algorithm.

2. Related Works

With lightweight and low-cost RGB-D sensors, such as Microsoft Kinetic, becoming increasingly popular, RGB-D SLAM has attracted significant attention from researchers. For example, a pose graph optimization framework is proposed to realize real-time and highly accurate SLAM [21,22]. In addition, the conventional bundle adjustment framework is extended to incorporate the RGB-D data with inertial measurement unit to optimize mapping and pose estimation in an integrated manner [23–25]. The inductively coupled plasma (ICP) method is used to estimate the motion between continuous frames containing dense point cloud, and is effectively applied to RGB-D SLAM [26,27]. it is clear that ORB SLAM2 [28] is one of the best SLAM methods which includes most state-of-the art techniques developed in recent years, and supports single, stereo, and RGB-D cameras. Mishima, M presents a framework for incremental 3D cuboid modeling combined with RGB-D SLAM [29]. To improve the positioning and mapping accuracy and the robustness of the system in various scenarios, various RGB-D SLAM methods are proposed correspondingly. To eliminate the influence of dynamic objects, Bescos, B presents a dynamic scene SLAM [30] which improves the dynamic object detection using multiview geometry and deep learning, along with background inpainting. Because the line features are more stable than point features, many researchers use line features to estimate camera poses. Sun, Q. matches the plane features between two successive RGB-D scans and the scans are used to estimate the camera pose [31] only if they can provide sufficient constraints. Zhou, Y provides Canny-VO [32], which efficiently tracks all Canny edge features extracted from the images, to estimate the camera pose. A real-time RGB-D SLAM with points and lines [33] is proposed. To reduce cumulative errors, a vertex-to-edge weighted closed-form algorithm [34] is used to reduce the camera drift error for dense RGB-D SLAM, which results in significantly low trajectory error than several state-of-the-art methods; in addition, this research has received great attention for the improving real-time performance. The back-end optimization problem is decoupled into linear components (feature position) and nonlinear components (camera poses) [35], and as a result complexity is significantly reduced without compromising accuracy; in addition, this algorithm can achieve globally consistent pose estimation in real-time via CPU computing. Thus, it is clear that, after various improvements, all of the above-mentioned RGB-D SLAM algorithms generally achieve better results in indoor environments involving small scenes. In environments involving large scenes, if prior data is not obtained, large cumulative errors are often caused by complex environments, such as by changes in illumination, motion blur, and feature mismatches. Although the global drift error can be reduced by loop closure, no loop is available in many scenarios, and computation of large loops is time-consuming. To address this problem, we intend to utilize the priori LiDAR data as the control information to guide the RGB-D SLAM operation in large indoor environments; the priori data can neglect the cumulative error problem irrespective of the tracking distance. Recently, multisource data fusion SLAM has been developed for complex indoor environments, which combines the LiDAR data and vision data; this technology mainly involves real-time fusion and postprocessing fusion. A lot of research has been conducted on real-time data acquisition fusion, by which the data acquired by the laser sensor and the camera are directly fused online for positioning and mapping. A real-time method is presented, namely V-LOAM [36], which combines visual odometry in high frequency and LiDAR odometry in low frequency to provide localization and mapping without global map optimization. Sarvrood, Y.B combines stereo visual odometry, LiDAR odometry, and reduced inertial measurement unit (IMU) to construct a reliable and relatively accurate odometry-only system [37]. Although both the above-mentioned methods function well, because the LiDAR data is usually obtained in real time without preprocessing, large cumulative errors may still exist after long-distance tracking. For postprocessed fusion, the LiDAR data can be utilized to obtain a priori map to enhance visual localization; however, little research exists on this technique. Fast and accurate feature landmark-based camera parameter estimation is achieved by adopting tentative camera parameter estimation and considering the local 3-D structure of a landscape using the dense depth information obtained by a laser range sensor [38]; however, only positioning, and not map construction, is obtained by this method. A mobile platform tracks the pose of a monocular camera with respect to a given 3-D LiDAR map, achieving excellent results under changing photometric appearance of the environment [39], because it relies only on matching geometry; however, the initialization time is long. The normalized mutual information between real camera measurements and synthetic views is utilized to the maximum extent, based on a 3D prior ground-map generated by a survey vehicle equipped with 3D LiDAR scanners to localize a single monocular camera [40]. This method does not fully utilize the scene information of the LiDAR data map, but utilizes only that of the ground part; therefore, it has reduced robustness in special environments. A UAV platform is utilized to obtain prior LiDAR data and a stereo camera is used to map large-scale outdoor scenes without automatic initialization and real-time performance [41]. To address the real-time map optimization problem, a multithreaded segment-based map double window optimization strategy is adopted in the method proposed in this paper, to ensure consistency and accuracy of map construction, as well as high real-time performance. The algorithm used for initial registration in the above-mentioned methods is time consuming or requires manual operation, which greatly limits its application, whereas the proposed method uses the deep learning neural network to perform automatic initialization and motion recovery, which improves the flexibility and robustness of the algorithm in difficult scenarios. In recent years, deep learning has shown great advancements and very high performance in various computer research fields because of their similar cognitive characteristics as humans. The combination of deep learning and geometric SLAM has been receiving increasing attention from researchers. Deep learning is used for closed-loop detection [42,43], and the results indicate that it is more suitable than conventional methods based on visual low-level features. Researches have been focused not only on addressing the conventional SLAM problems, geometric mapping, and localization, but also on semantic SLAM, which has gradually become a trend in the field of SLAM. The object detection module, realized by the deep-learning method and localization module, is integrated seamlessly into the RGB-D SLAM framework to build semantic maps with object-level entities [44]. A robust semantic visual SLAM named DS-SLAM [45], aimed at dynamic environments, is proposed. The absolute trajectory accuracy of this method can be improved by one order of magnitude compared with that of the ORB-SLAM2.

A review of literature shows that visual and RGB-D SLAM generally use the natural landscape to construct a 3-D environment and simultaneously estimate the camera pose; however, under highly variable illumination conditions and low-textured environments, without priori data, errors accumulate in long distance tracking, causing tracking failures, which may result in kidnapped-robot problem. This limitation often hinders the practicality of SLAM technology in real-life applications. Moreover, many SLAM applications such as the Google self-driving car, which is based on high definition map and indoor navigation using priori map, do not require a moving platform to navigate in unknown environments. Therefore, this research aims at proposing a novel RGB-D SLAM approach that, along with deep learning, utilizes a highly accurate priori LiDAR point cloud data as guidance for indoor 3-D scene construction and camera pose estimation. The contributions of this research can be summarized as follows.

(1) Using an effective combination of priori LiDAR data and deep learning, an approach for SLAM system automatic initialization and motion recovery in complex indoor environments, is provided.
(2) Priori LiDAR data of the indoor scene, along with the multithreaded segment-based map double window optimization strategy, is used to eliminate cumulative error in camera pose and map construction, as well as improve the real-time performance.
(3) A feasible solution is provided for automatic matching of 3-D LiDAR data and 2-D image data from the RGB-D sensor in large indoor scenes, and fusing the 2-D and 3-D data to construct an accurate environmental model of the scene, thereby compensating the deficiency of single data source.
(4) To ensure accuracy, high real-time performance, and robustness, priori LiDAR data and deep learning are successfully integrated into ORB-SLAM2, creating a complete and novel SLAM framework.

3. Materials and Methods

The proposed method, built on the ORB-SLAM2 framework, uses the terrestrial LiDAR point cloud data with high accuracy and precision to guide the RGB-D SLAM in real-time to construct large-scale and accurate 3-D textured models and obtain accurate camera poses using a multithreaded framework, so that the accumulation error of the RGB-D SLAM during long-distance tracking is eliminated. Automatic system initialization and motion recovery under difficult scenarios, real-time performance, accurate priori LiDAR data, drift-error-free performance, dynamic object culling, and accurate and consistent global map are the six key characteristics of the proposed algorithm. The proposed approach primarily and innovatively integrates the existing methods to yield perfect results while ensuring the system operates in real-time with high precision and remains robust under challenging environments.

Figure 1 illustrates the proposed method in a detailed flowchart, which is composed of five parts: (1) acquisition and preprocessing of LiDAR data from the experimental site, (2) accurate automatic initialization of the RGB-D SLAM algorithm in the LiDAR data coordinate system framework, (3) accurate camera pose determination using the ICP algorithm with priori LiDAR data as ground truth control, (4) accurate construction of 3-D indoor map by multithreaded segment-based map optimization using LiDAR corrected camera pose from the previous step, and (5) improving the algorithm's robustness under challenging environments, using the priori LiDAR data and a combination of deep learning and geometric SLAM. The proposed algorithm has four important threads: pose tracking, map construction, loop closing, and LiDAR data optimization. Furthermore, the LiDAR data optimization thread includes two key functions, which are camera pose estimation with LiDAR ground control information and multithreaded segment-based map optimization.

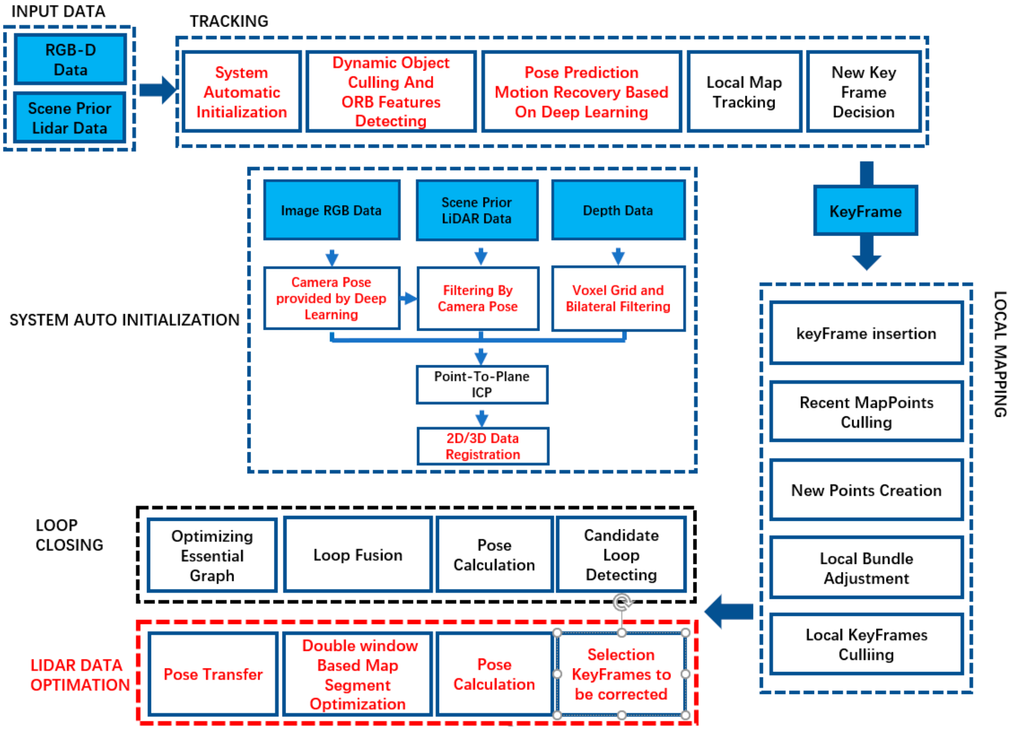


Figure 1. Flowchart of the proposed Light Detection and Ranging (LiDAR)-guided RGB and depth (RGB-D) simultaneous localization and mapping (SLAM).

3.1. LiDAR Data Acquisition and Preprocessing

We used a 3-D ground-based LiDAR scanner RIEGL VZ-1000 (RIEGL Laser Measurement Systems GmbH, Horn, Austria, shown in Figure 2), capable of providing a range of 1400 m, measurement accuracy of 5 mm, measurement rate of 300,000 points/second, and field of view of 100° × 360°. Single scans are registered together to form a single model, as shown in Figure 3, using the RIEGL LiDAR processing software. Table 1 lists the specifications of the LiDAR scanner RIEGL VZ-1000.

Figure 2. RIEGL VZ-1000 LiDAR scanner.

Table 1. Theoretical accuracy of RIEGL VZ-1000.

Field of View	Accuracy (mm)	Range (m)	Acquisition Rate (Measurement/s)
100° × 360°	5	1400	122,000

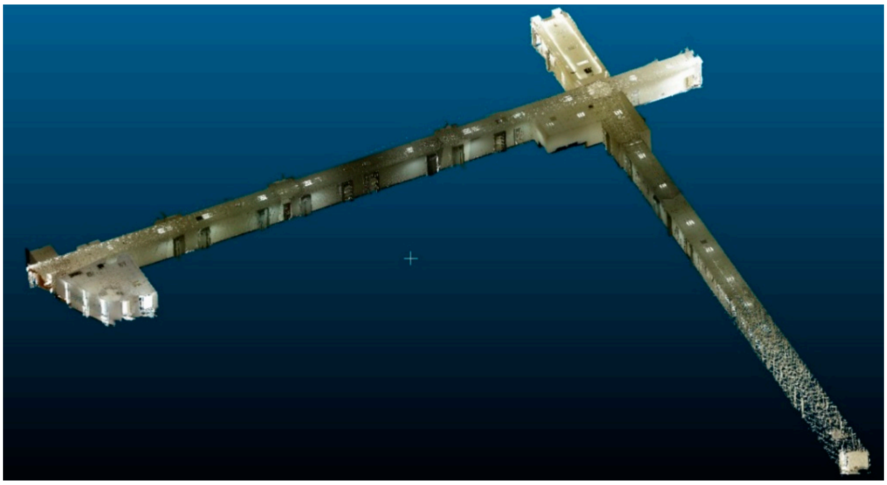


Figure 3. Registered LiDAR point cloud from multiple scans.

Because the volume of data output of the 3D LiDAR scanner is huge, the collected data must be preprocessed to ensure efficient and good real-time operation of the proposed algorithm. For this purpose, downsampling and KD-tree data structure are applied to the LiDAR point cloud data. This study uses a voxel with a resolution of 5 cm to downsample the point cloud, while retaining the geometrical information of the point cloud data with minimum data volume, as shown in Figure 4. Further, the KD-tree structure, used as the data structure of point cloud storage management, greatly improves the retrieval rate of cloud data in large-scale indoor scenes.

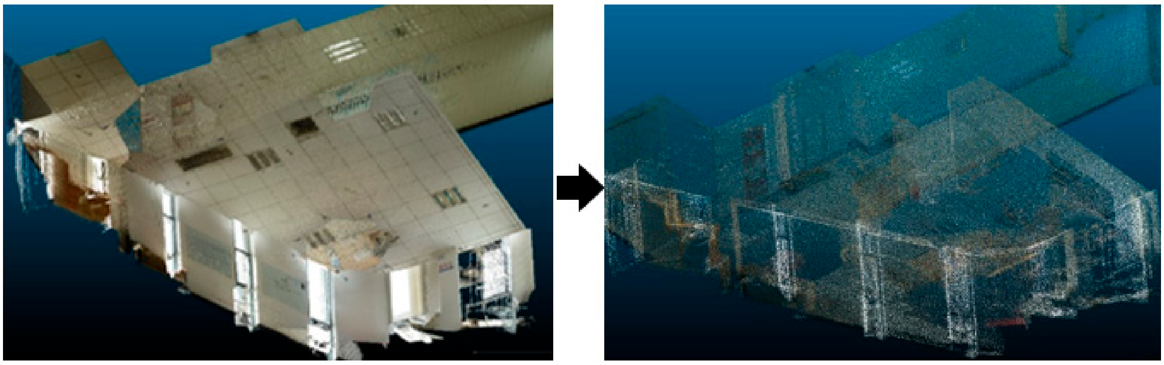


Figure 4. LiDAR point cloud data voxel grid downsampling.

3.2. Automatic and Accurate System Initialization

System initialization is a prerequisite for the proposed algorithm. Initially if the system cannot match the LiDAR data accurately under its coordinate framework, the LiDAR data cannot be used as ground truth for subsequent optimization. In this study, deep learning (PoseNet) is used to automatically align the priori LiDAR point cloud data with the RGB-D point cloud in a single coordinate system framework. PoseNet neural network was transformed from GoogLeNet, which is a 22-layer convolutional network with six “inception modules” and two additional intermediate classifiers. PoseNet replaces all three softmax classifiers in GoogLeNet with affine regressors. The softmax layers were removed and each final fully connected layer was modified to output a pose vector of 7-dimensions representing position (3) and orientation (4). Based on our previous work [46], first, we manually aligned the priori LiDAR data with RGB-D point cloud, obtained consecutive video frames labeled in camera pose produced by the relocalization function in the RGB-D SLAM method for the corresponding scene, trained the CNN network using consecutive video frames, and produced an average positional error of about 1 m and average angular error of 4°. Then the initial coarse camera pose was achieved from deep learning with errors (i.e., misalignment between RGB-D and the corresponding LiDAR point cloud set), as indicated clearly in Figure 5b. A more accurate and refined initial camera pose was obtained using the nearest neighbor iterative algorithm initialized by pose from deep learning, as shown in Figure 5c. Figure 6 demonstrates the final initialization result with accurately matched RGB-D and LiDAR scenes.

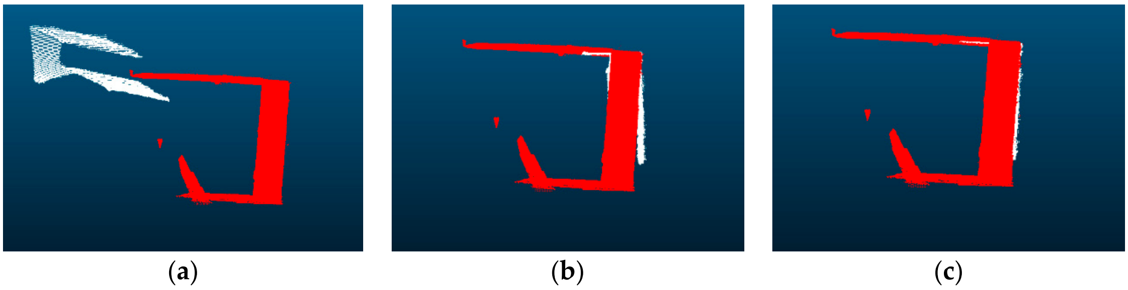


Figure 5. LiDAR point cloud is shown in red and RGB-D point cloud in white. (**a**) Unmatched RGB-D point cloud and LiDAR point cloud, (**b**) matched RGB-D point cloud and LiDAR point cloud using deep learning, and (**c**) matched RGB-D point cloud and LiDAR point cloud using inductively coupled plasma (ICP) with initial value from deep learning.

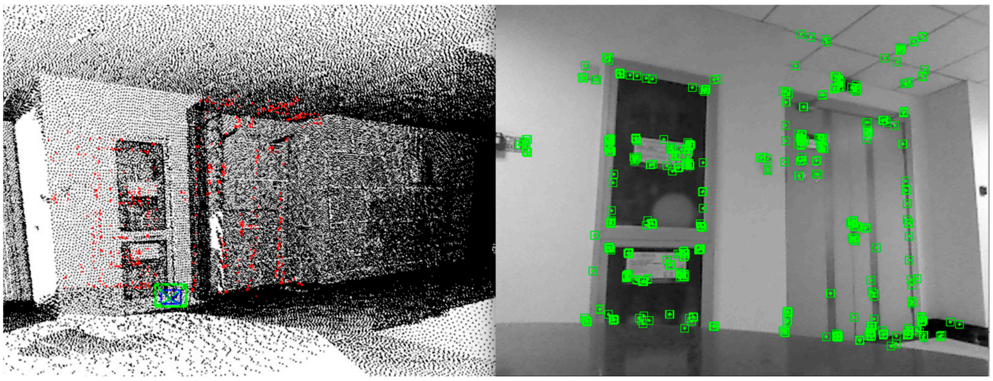


Figure 6. Successful initialization by matching the RGB-D and LiDAR scenes by ICP with initial value from deep learning.

3.3. Accurate Camera Pose Determination Using ICP with LIDAR and RGB-D Point Cloud Data

After the proposed algorithm is successfully initialized, the priori LiDAR point cloud data is aligned to the RGD-SLAM. Now the objective of the proposed algorithm is to obtain accurate camera pose ground control information from the priori LiDAR data, and control the operation and optimization of the RGB-D SLAM in LiDAR data optimization thread. Figure 7 illustrates the framework of the LiDAR guided camera pose estimation algorithm, which comprises three steps: (1) selection of the keyframes to be corrected and generation of corresponding RGB-D point cloud scene with dynamic object culling and denoising of the depth image data, (2) data acquisition of local LiDAR point cloud based on initial pose produced by the visual tracking thread, and (3) using point-to-plane ICP algorithm to obtain and refine accurate pose of the keyframes and transfer the poses to the local map through the keyframe covisibility graph. The following subsections describe each step in detail.

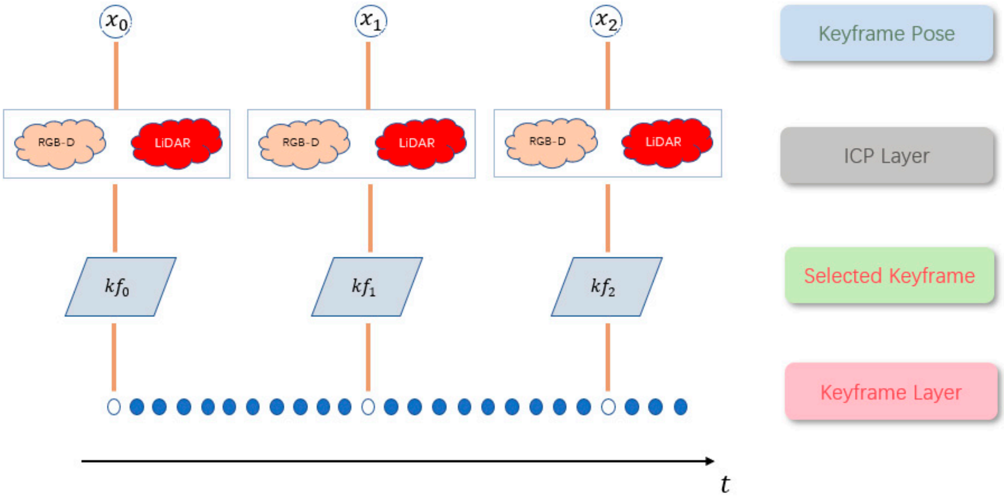


Figure 7. Process of accurate pose determination from LiDAR data.

3.3.1. Selection of Keyframes to Be Corrected and the Generation of the Corresponding RGB-D Point Cloud Scene

In a typical visual SLAM method, the error in camera pose estimation usually accumulates over time. Although it is unnecessary to correct the cumulative error of each frame with the LiDAR data, the accumulation error must be maintained sufficiently low within the initial range required by ICP convergence to the global optimal. To ensure both drift-error-free and real-time performances, LiDAR data are used to eliminate accumulation error for every 40 keyframes. A 40-keyframe interval is chosen because a balance between successful convergence of the ICP algorithm and real-time performance is maintained by experiments. For instance, if an interval lower than 20 keyframes is applied, the LiDAR data optimization thread falls behind the tracking thread and results in delayed camera pose correction of the current frame. An interval higher than 60 keyframes produces higher error accumulation and results in failure of ICP convergence to the global optimal, because the camera pose provided by the tracking thread is used as the initial value of the ICP algorithm.

Furthermore, because of the complexity of the test environment, the selected keyframes for correction are further filtered according to the field operational condition of the proposed algorithm. For instance, in the case of fast rotation of camera, the cumulative error grows accordingly, because the tracking image becomes blurred and feature mismatch rate increases. Therefore, the selection criterion for the uncorrected keyframe is "whether the change in positive direction of the camera before and after the two adjacent keyframes exceeds a certain threshold." If the threshold is not exceeded, the camera moves forward smoothly; however, if the threshold is exceeded, the camera rotates too fast and must be corrected. The threshold was set as 45° in our research.

On the other hand, dynamic objects in the scene may cause error in pose tracking, because they interfere with the acquisition of accurate pose control information from LiDAR data. To address this problem, the optical flow method is used to detect and remove dynamic objects in RGB and depth images in real time, as shown in Figure 8. Because variable illumination and infrared reflection characteristics of the surface material of the object may produce noise in the depth data, a bilateral filter denoising method, which combines the spatial proximity and similarity of the pixel values, is adopted to preserve detail and remove noise from the depth image data.

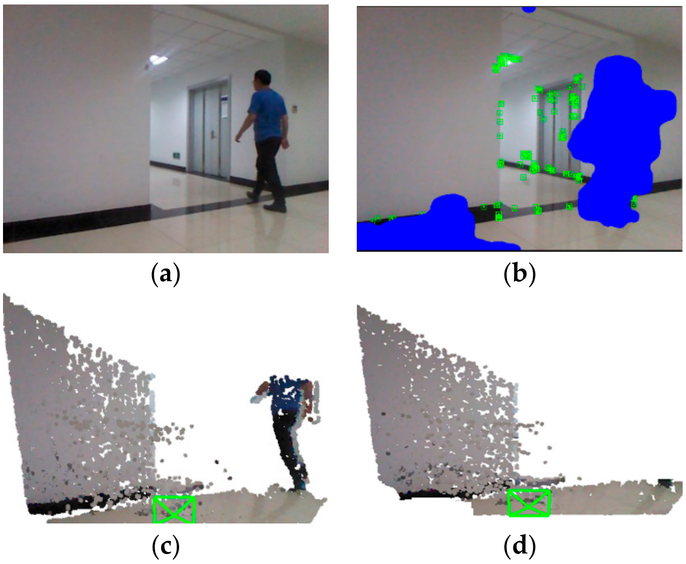

(**a**) (**b**) (**c**) (**d**)

Figure 8. Dynamic object culling based on optical flow method. (**a**) Original RGB image, (**b**) dynamic object culling in RGB image, (**c**) corresponding point clouds generated by original depth data, and (**d**) dynamic object culling in depth data.

After selecting the keyframe, culling the dynamic objects, and performing noise removal, the RGB-D point cloud data is generated based on the depth information of the RGB-D camera and the rough pose of the tracking thread. Because the volume of RGB-D point cloud data is huge, which affects the efficiency of the ICP iterative algorithm, the voxel grid accordingly downsamples the point cloud to a resolution of 5 cm to acquire RGB-D sparse point cloud for matching with LiDAR point cloud.

According to the rough pose provided by the visual tracking of the keyframe, the LiDAR point cloud data covering the complete experimental site is filtered to obtain the local LiDAR point cloud data, ensuring a high degree of overlapping with the RGB-D point cloud data at the current keyframe for ICP iteration. The degree of overlapping between the acquired local LiDAR data and the RGB point cloud depends on the cumulative error of the keyframe camera pose.

3.3.2. Accurate Pose Determination

Owing to complexity of the indoor scene, obtaining a precise camera pose from the LiDAR data is challenging. Generally, an indoor environment satisfies the Manhattan world hypothesis that is mainly composed of cubical objects, which have abundant geometric structured information in LiDAR data. Therefore, the point-to-plane ICP iterative algorithm initialized by the pose of the visual tracking thread achieves fast and robust convergence to the global optimal, and can obtain precise camera pose control information, although the point-to-plane ICP iterative algorithm is not innovative. Equation (1) illustrates the point-to-plane ICP distance error function:

$$p_i \in P,\ q_i \in Q$$

$$\begin{aligned} \mathrm{E}(\alpha,\beta,\gamma,t_x,t_y,t_z)^k &= \sum_{i}^{|D|}\sum_{j}^{|M|} w_{ij} d_s^2(T^k p_i, S_j^k) \\ &= \sum_{i}^{|D|}\sum_{j}^{|M|} w_{ij}\left(\frac{\|p_i(x)*A_j+p_i(y)*B_j+p_i(z)*C_j+D_j\|}{\sqrt{{A_j}^2+{B_j}^2+{C_j}^2}}\right) \end{aligned} \tag{1}$$

where,

P is the RGB-D point cloud data generated by the corrected keyframe (Figure 9); Q is the local LiDAR point cloud data; $A_j, B_j, C_j,$ and D_j are the plane parameters of the tangent plane ${S_j}^k$; w_{ij} is the weight of the distance from the point p_i to the plane ${S_j}^k$ (usually set to one); $\alpha, \beta,$ and γ are rotation angles; and t_x, t_y, and t_z are translation parameters. The registration algorithm finds a final transformation T which minimizes the above-mentioned error function $E(\alpha, \beta, \gamma, t_x, t_y, t_z)^k$ by iteratively using the least squares method; the iterative termination condition is that the number of iterations is greater than the maximum number of iterations, or that the registration error defined in Equation (2) is less than a given threshold.

$$\sigma = \frac{\|E(\alpha, \beta, \gamma, t_x, t_y, t_z)^k - E(\alpha, \beta, \gamma, t_x, t_y, t_z)^{k-1}\|}{N} \leq \epsilon_e \ (\epsilon_e > 0) \tag{2}$$

where ϵ_e is the error threshold and N is the number of used points p_i.

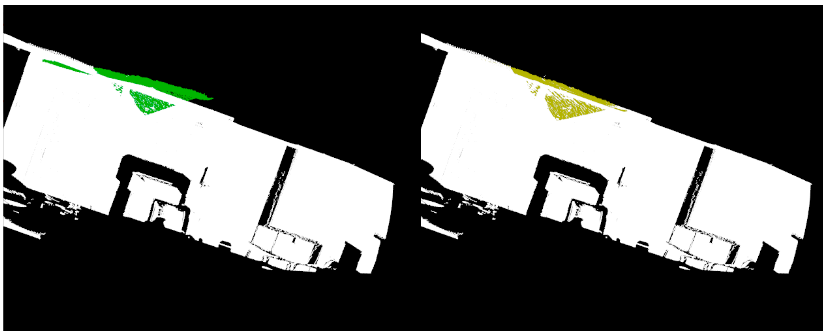


Figure 9. RGB-D point cloud generated by keyframes and LiDAR point cloud data ICP (point to plane) registration.

Using the above-mentioned ICP algorithm, the camera pose is corrected and the accurate pose is obtained. However, because the effect of single LiDAR corrected keyframe camera pose in the map is small, the accurate pose needs to be propagated to the local map using keyframe covisibility graph to expand the control range of the accurate pose. The pose transfer to the local map is described in Equation (3).

$$\begin{cases} K_f = \left\{k_f^i \vee k_f^i \cap k_f! = 0, k_f^i \in Map\right\} \\ T_{c_i c} = T_{c_i m} T_{mc} \\ T_{c_i w} = T_{c_i c} T_{cw} \\ p_{iw}^j = T_{wc_i} p_{ic}^j \end{cases} \tag{3}$$

where, K_f is a keyframe set is shared with the corrected keyframe k_f, $T_{c_i c}$ is the relative pose between the common view keyframe and the corrected keyframe, $T_{c_i w}$ is the corrected pose of the common view keyframe, p_{iw}^j is the corrected 3D coordinates of the j-th map point observed by the common view keyframe, and p_{ic}^j is the coordinate of the map point in the keyframe camera coordinate system. After the above steps are performed, the accuracy of the local map around the corrected keyframe is improved, and the control range of the accurate pose information in the map is expanded.

3.4. Sparse Maps and Semi-Dense Map Construction

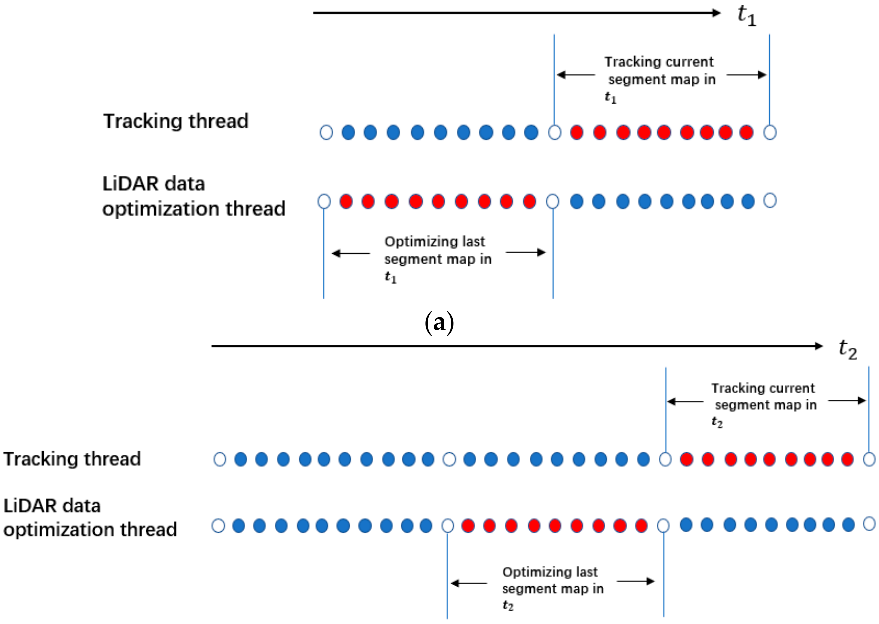
Although the map model of the scene is available as the LiDAR data, this data often contains only the intensity information, and not the texture information. In addition, the data of some detailed scenes are missed because of occlusion problems during scanning. Therefore, the proposed algorithm constructs high-precision maps with texture information under constraint of the priori LiDAR data, improves the content and use of the LiDAR data map, and patches data holes.

The proposed method is based on the ORB-SLAM2 algorithm framework, which has a sparse map construction thread. The sparse map consists of keyframes and corresponding 3-D ORB feature points. The 3-D ORB feature points are created by projection based on keyframe camera pose and ORB feature point depth data. Then the final keyframes and 3-D ORB feature points in the local map are created by applying special selection strategies to candidates in the local mapping thread. Local optimization and global optimization for the sparse map are applied in ORB-SLAM2 without external constraints; thus, errors in both camera pose and sparse map are cumulated after long distance tracking. The proposed method further optimizes the sparse map constructed by ORB-SLAM2 using the precise camera pose constraints provided by the LiDAR data to eliminate the cumulative error. Because the optimization operation is performed once for every 40 keyframes, the optimization frequency is high. Therefore, to achieve a good real-time performance, and high accuracy and consistency of sparse map optimization, the multithreaded segment-based map double window optimization strategy is adopted. After optimization, precise camera poses for all keyframes were obtained by combining the depth image data of keyframes to recover the dense point cloud model of the scene. Then we constructed a semi-dense map based on keyframes, which fuses the 2-D rich texture information from image data and the high-precision 3-D geometrical information from the terrestrial LiDAR point cloud data.

3.5. Segment-Based Sparse Map Optimization Using Double Window Approach

In previous sections, though locally optimized poses were obtained, the map did not achieve the global optimal. Therefore, to achieve good real-time performance, as well as a globally optimal and consistent map, we adopted a multithreaded segment-based map double window optimization strategy that considered the first and last keyframes of the current map segment corrected by the ICP algorithm with LiDAR data as the true values and then applied the double window approach to optimize this map segment.

In multithread segment-based map optimization, as shown in Figure 10, the visual tracking thread operates on current map segment whereas the LiDAR data optimization thread operates on the previous map segment. When the visual tracking thread starts tracking the next map segment from t_1 to t_2, the LiDAR data optimization thread simultaneously completes optimizing the previous map segment and starts optimizing the next segment; therefore, real-time optimization and tracking performance are ensured.


t_1
Tracking current segment map in t_1
Tracking thread
LiDAR data optimization thread
Optimizing last segment map in t_1
t_2
Tracking current segment map in t_2
Tracking thread
LiDAR data optimization thread
Optimizing last segment map in t_2

(**a**)

(**b**)

Figure 10. Tracking threads and LiDAR data optimization threads operate on different segments of the map; small dots in red represent keyframes currently activated by the corresponding thread. (**a**) keyframes currently activated by the corresponding thread in t_1, (**b**) keyframes currently activated by the corresponding thread in t_2.

In the double window approach, as shown in Figure 11, the optimization window is divided into a front window and a back window. The front window constraint ensures the accuracy of the new map and the back window constraint ensures consistency between front and rear map segment. The front window comprises 40 keyframes in the current segment and uses the pose-point constraint to control the precise construction of this local map segment. The back window consists of an optimized set of keyframes that observe the map points in the front window. The back window constraints include the pose–pose constraints between two keyframes with common visible map points in the back and front window, as well as the pose-point constraints between the keyframes in the back window and the map points in the front window. These two constraints help achieve global consistency of the map. To improve the optimization efficiency and memory consumption, the map points that were optimized multiple times and the keyframes in the back window that are far from the front window are removed from the current optimization. This research considers the map points that have been optimized more than 6 times to be sufficiently accurate, and removes them from current optimization; therefore, a set of front window map point optimization variable $V_{p-front}$ is obtained. The set of front window keyframe optimization variable is then $V_{kf-front}$. Keyframes with distance more than 4 m were removed. The set of the back window keyframe optimization variable is $V_{kf-back}$.

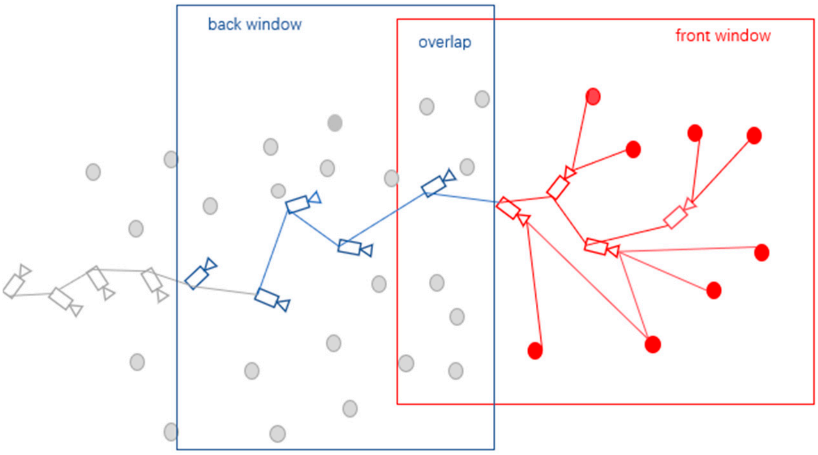


Figure 11. Front window and back window.

Error Equation Formation

We constructed the total error equation in the current segment, which is given by

$$\begin{aligned} E = E_1 + E_2 + E_3 &= \sum_{i=1}^{m} E(kf_i) + \sum_{k=1}^{a} E(kf_k) + \sum_{k=1}^{a}\sum_{i=1}^{m} E(T_{ki}) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} \|z_{ij} - h(\xi_i, p_{ij})\|^2 + \sum_{k=1}^{a}\sum_{j=1}^{n} \|z_{kj} - h(\xi_k, p_{kj})\|^2 \\ &+ \sum_{k=1}^{a}\sum_{i=1}^{m} v_{ki}{}^T \Lambda_{T_{ki}} v_{ki} \end{aligned} \tag{4}$$

$$p_{ij} \in V_{p-front}$$

$$p_{kj} \in V_{p-front}$$

$$kf_k \in V_{kf-back}$$

$$kf_i \in V_{kf-front}$$

where, $E(kf_i)$ is the error equation of pose-point constraint in the front window; $E(kf_k)$, $E(T_{ki})$ are the error equations of pose-map point constraint and pose–pose constraint in back window, respectively.

The Jacobian matrix (J), Hessian matrix (H), and gradient vector (g) can be obtained as follows

$$\begin{cases} H = \sum\limits_{i,j} {J_{ij}}^T J_{ij} + \sum\limits_{k,j} {J_{kj}}^T J_{kj} + \sum\limits_{k,i} {J_{T_{ki}}}^T J_{T_{ki}} + \lambda I \\ g = H\Delta x \end{cases}$$

By the above steps, the optimization model for double window is established, where λ is the positive definite factor in the LM iterative optimization algorithm.

3.6. Combination of Deep Learning CNN with Geometric SLAM

The deep learning CNN position system (PoseNet) consists of the GoogleNet classification deep convnet and pose regression convnet, which are based on high-level semantic features and output a pose vector P, given by the camera position x and orientation represented by quaternion q. The deep learning CNN position system is resistant to multimodal, motion blur, variable illumination, and less-texture scene. Therefore, the proposed algorithm, which utilizes ICP iterative algorithm initialized by the coarse pose from the deep learning to obtain precise camera pose, can recover camera motion during tracking failure, and greatly improve algorithm's robustness in difficult environments.

4. Results

An indoor scene generally consists of two geometric structures: rectangular block and curved plane structures. To evaluate the effectiveness of the proposed algorithm in different indoor scenarios, we used the RGB-D camera (kinect v1) to capture image data and depth data in two typical indoor scenes: the third floor of Block A consisting of planar rectangular block structures and closed round hall consisting of curved plane structures located in the New Technology Park of Aerospace Information Research Institute, Academy of Sciences. Because the published data set does not have the terrestrial LiDAR data of the scene, we conducted the experiment with two data sets collected from each experimental site, long-distance tracking data set, and closed-loop data set. The RIEGL VZ-1000 terrestrial LiDAR scanner was used to collect the LiDAR data. The resolution of the point cloud data had an accuracy of the order of millimeter.

Before performing the experiment, we used the OpenCV camera calibration module to conduct the camera calibration process. The calibration parameters include lens distortion coefficients and the internal parameters, as seen in Table 2.

Table 2. Calibration results of RGB-D camera.

Camera	f_x	f_y	c_x	c_y	k_1	k_2	k_3	p_1	p_2
Kinect v1	514.994	513.758	321.045	244.587	0.0110	−0.0521	0021	0028	−0.0064

4.1. Experiment in Corridor

4.1.1. Long Distance Tracking

We designed this experiment to verify if the proposed method can eliminate the cumulative error of RGB-D SLAM in long distance tracking in corridor scene with poor texture information. The experiments were performed in a travel distance of 100 m using a laptop (Dell precision7530 GPU work station)—12 Intel Core i7-8750H CPU, 2.20 GHz frequency, 16 GB RAM memory, 6G NVIDIA corporation GP104GLM. Figure 12a,b shows the sparse maps constructed by the two methods.

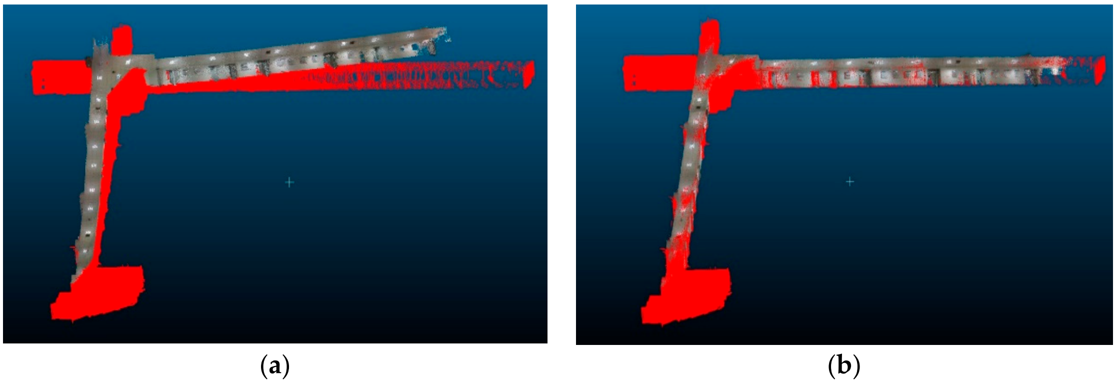


Figure 12. Comparison of models constructed by two methods, with LiDAR data rendered in red. (**a**) Model constructed by ORB-SLAM2 and (**b**) model constructed by the proposed method.

To quantitatively assess the accuracy of the model, Table 3 has been provided, which shows the matching error between the ground truth LiDAR data and the sparse map generated by the two methods. As seen in Table 3, the mean error of our method is 0.08 m, whereas that of ORB-SLAM2 is 1.18 m.

Table 3. Statistical results of matching error between sparse maps generated by the two methods and the ground truth LiDAR data.

Method	Max (m)	Mean (m)	Std. Dev. (m)
ORB-SLAM2	4.5549	1.1882	0.9833
Proposed Method	2.0370	0.0834	0.0992

4.1.2. Loop Closure Experiment

We conducted a loop closure experiment in a travel distance of 100 m. Figure 13a,b clearly show the maps and the camera trajectory generated by the proposed method and ORB-SLAM2. It is observed that, in Figure 13a, the map has been wrongly divided into two. As seen from statistical results of matching error between the models generated by the two methods in loop closure experiment and the ground truth LiDAR data in Table 4, the mean error of our method is 0.1722 m, whereas that of ORB-SLAM2 is 0.7337 m.

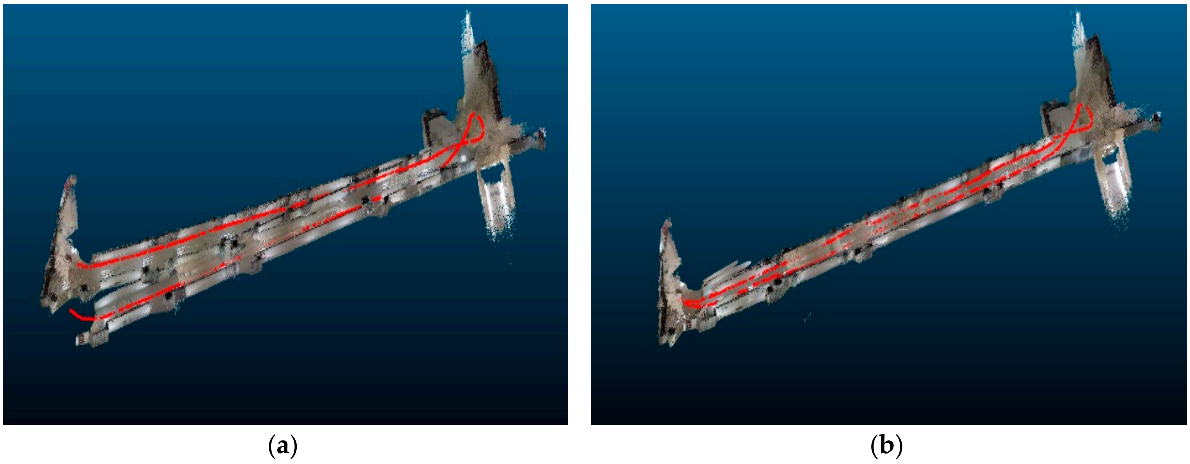


Figure 13. Camera trajectory of the two methods in loop closure experiment. (**a**) Camera trajectory of ORB-SLAM2 and (**b**) camera trajectory of the proposed method.

Table 4. Statistical results of matching error between the models generated by the two methods in loop closure experiment and the ground truth LiDAR data.

Method	Max (m)	Mean (m)	Std. Dev. (m)
ORB-SLAM2	3.5282	0.7337	0.5743
Proposed Method	2.0370	0.1722	0.2233

Figure 14 illustrates a comparison of the camera trajectories produced by the two methods in the loop closure experiment; the trajectory in red was generated by the proposed method, and that in white was produced by ORB-SLAM2. When the map segment optimization approach was applied, the continuity of the trajectory was affected, as shown in Figure 13. Poor continuity occurs when the camera's cumulative error is corrected by the LiDAR data. The quantitative statistical results are provided in Table 5. It can be observed from Table 5 that the total lengths of the trajectories of the same data set produced by two different methods are different because of the difference in the cumulative error in the camera trajectory. It is evident from this result that the proposed method has a closure error of 0.0977 m and an error percentage of 0.09%, whereas ORB-SLAM2 has a much larger closure error of 3.4247 m and an error percentage of 3.46%.

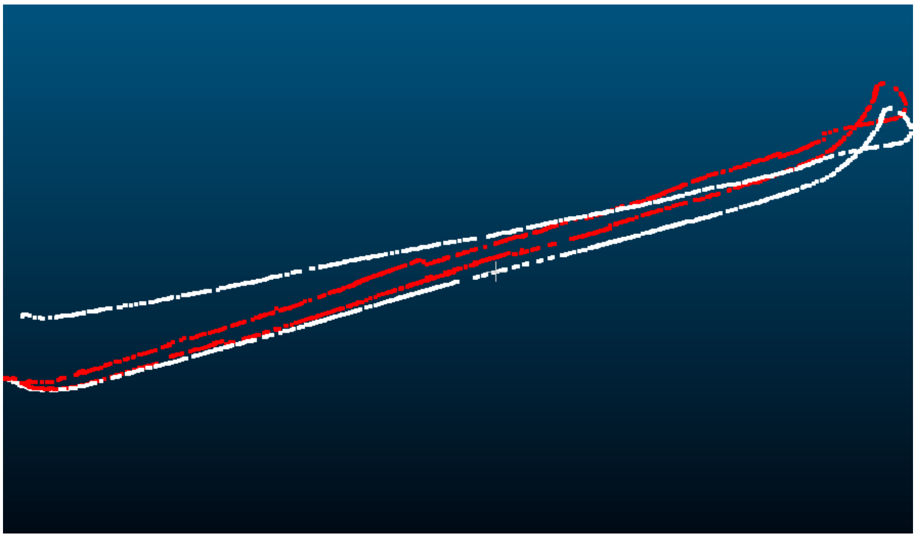


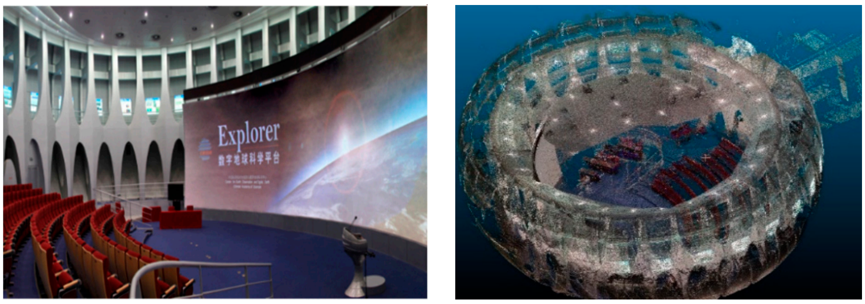
Figure 14. Camera trajectories produced by ORB-SLAM2 and the proposed method are shown in white and red, respectively.

Table 5. Camera trajectory error results for ORB-SLAM2 and the proposed method.

Method	Total Length (m)	Closure Error (m)	Percentage Error (m)
ORB-SLAM2	99.039	3.4247	3.46%
Proposed Method	106.552	0.0977	0.09%

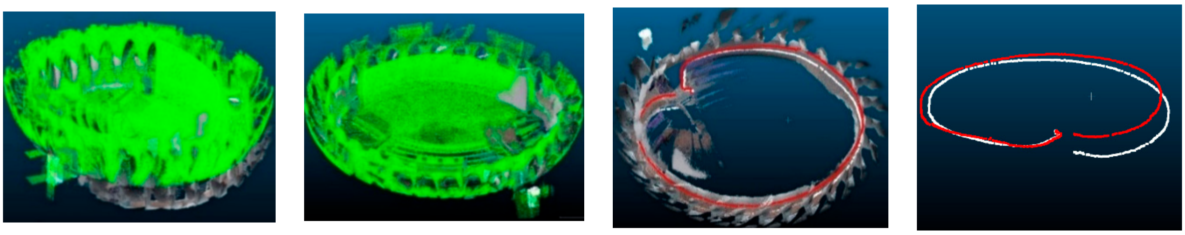
4.2. Experiment in Round Hall

Another long-distance tracking experiment was designed in a closed round hall, which had two floors and a circumference of 85 m, as seen in Figure 15. Compared with the corridor scene, this scene was more complex where the distance from the ceiling to the ground was about 11 m; the interior structure consisted of a curved plane, and a lot of empty space existed in the middle of the hall. Therefore, obtaining accurate camera pose control information from prior LiDAR data is more difficult. The hall supported controlling the lighting conditions, thus testing the robustness of the approach with variable illumination was possible. Figure 16a,b shows the sparse map constructed by the two methods, and Figure 16c,d shows the camera trajectory. Clearly, the camera trajectory of the proposed method (shown in red) is generally consistent with the actual scene, whereas that of ORB-SLAM2 (shown in white) has a large misalignment. Table 6 shows the matching error between the model and LiDAR data. The mean error of the proposed method is 0.2142 m, whereas that of ORB-SLAM2 is 0.6406 m. The results of two typical indoor scenes illustrate that the proposed method can eliminate the cumulative error in long distance tracking.



(**a**) (**b**)

Figure 15. (**a**) Closed round hall consisting of a curved plane structure and (**b**) LiDAR point cloud data in closed round hall obtained by voxel grid downsampling.



(**a**) (**b**) (**c**) (**d**)

Figure 16. (**a**) Registration between models constructed by ORB-SLAM2 and LiDAR point cloud data model rendered in green. (**b**) Registration between models constructed by the proposed method and LiDAR point cloud data. (**c**) Sparse map constructed by the proposed method with camera trajectory overlay. (**d**) Comparison of the camera trajectories: camera trajectories produced by ORB-SLAM2 and the proposed method are shown in white and red, respectively.

Table 6. Statistical results of matching error between the sparse maps generated by the two methods and the ground truth LiDAR data in closed round hall.

Method	Total Length (m)	Max (m)	Mean (m)	Std Dev (m)
ORB-SLAM2	90.5698	2.8368	0.6406	0.6214
Proposed Method	94.8059	1.7830	0.2142	0.2046

4.3. Deep Learning CNN Training and System Robustness Test

By employing the proposed algorithm by manually selecting point registration in LiDAR data and the RGB image, an accurate sparse map, which combines high-precision 3-D geometrical information with rich 2-D texture information, can be constructed. The precise camera pose of multiview image with centimeter-level accuracy can be obtained by the relocalization function based on bag-of-words model. Therefore, the proposed method can easily achieve high-quality training sample dataset for the deep learning position system, which can reduce the cost of labor. We trained the deep learning position network with 7813 RGB images with pose labels. The training data set basically included all corners of the closed round hall, which covered an area of 796.1216 square meters.

As seen from Figures 17 and 18, and Table 7, deep learning position network training with high-quality and full data set produces less error in the case of large indoor space; the average position error was 0.77 m and angular error was ~4°. With these degrees of errors, the acquired camera pose can be sufficiently used as an initial value of the ICP iteration algorithm for obtaining accurate camera pose from the LiDAR data.

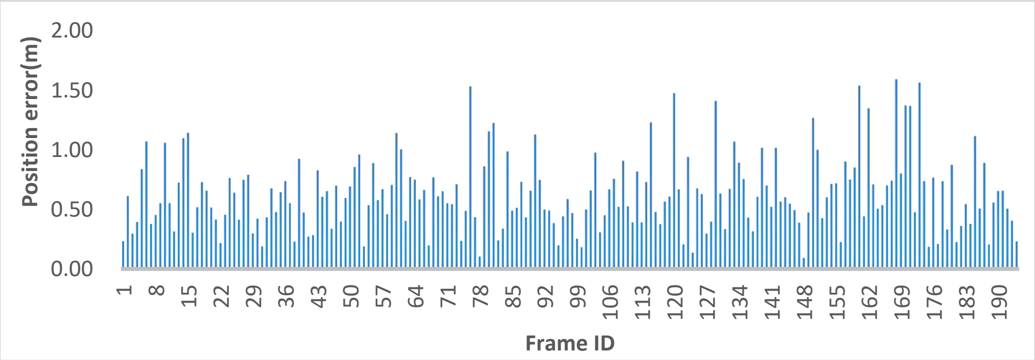


Figure 17. Position error of deep learning position network (PoseNet) in closed round hall.

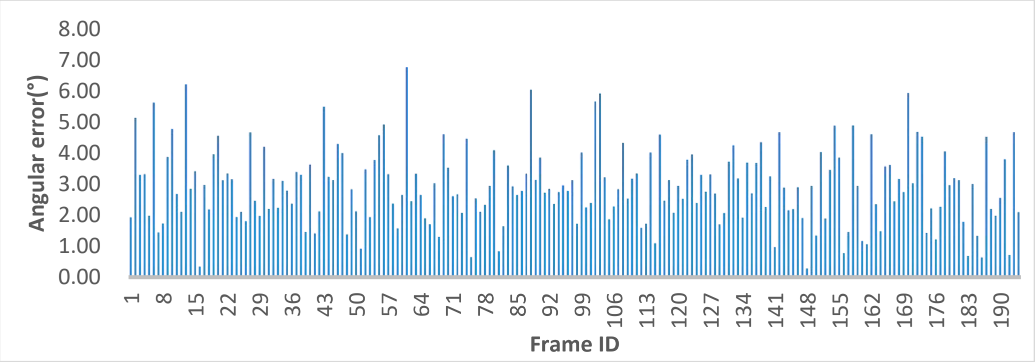


Figure 18. Angular error of deep learning position network (PoseNet) in closed round hall.

Table 7. Deep learning position results in closed round hall.

Error Type	Size (m^2)	Max	Min	Median	Average
Position error (m)	796.1216	1.5307	0.0909	0.7238	0.7700
Angular error (°)	796.1216	6.7675	0.2712	2.6541	4.0697

In the closed round hall scene, we changed the illumination suddenly and moved the camera quickly to initiate system track failure. The relocalization based on the bag-of-words model in ORB-SLAM2 was unusable in this condition, whereas deep learning based on high-level semantic feature recovered its motion successfully. The experimental results show that the constructed map overlaps with the LiDAR data well after restoration of system motion, as shown in Figure 19a,b and Table 8.

Table 8. Statistical results of matching error between constructed points after motion recovery with the LiDAR data in low-light and motion blur scenarios.

Scenes	Max (m)	Mean (m)	Std Dev(m)
low-light	0.6114	0.0909	0.0498
motion blur	0.4323	0.1225	0.1168

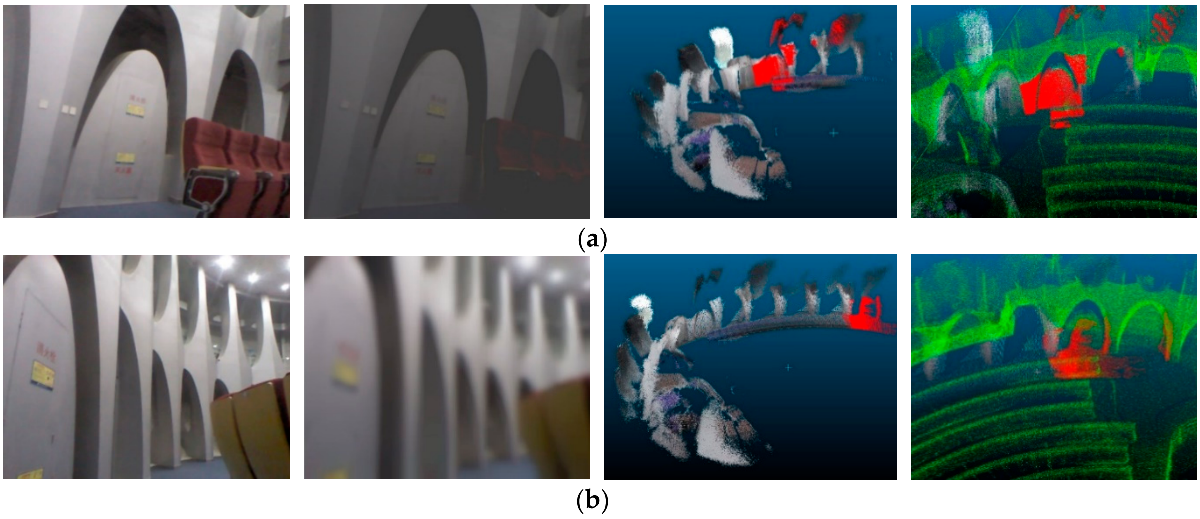


Figure 19. (**a**) Constructed points (red) overlap with the LiDAR data (green) after motion recovery in low-light scenario and (**b**) constructed points (red) overlap with the LiDAR data (green) after motion recovery in motion blur scenario.

5. Discussion

5.1. Definition of Keyframe

Before providing a detailed discussion, we would like to elaborate the definition of keyframe. The image insertion frequency is very high during the operation of SLAM, which leads to a rapid increase in information redundancy. However, the accuracy of the redundant information is low, and does not improve; further, it requires more computing resources. Therefore, the SLAM algorithm uses a specific selection strategy to select part of the image frames as the keyframe for location and mapping. The purpose of this strategy is to obtain the best balance between information redundancy and computing resource loss without loss of accuracy, to satisfy the requirements. Generally, as shown in Table 9, the visual tracking thread takes ~550 ms to select a new keyframe, and ~30 ms to receive a frame from the camera sensor.

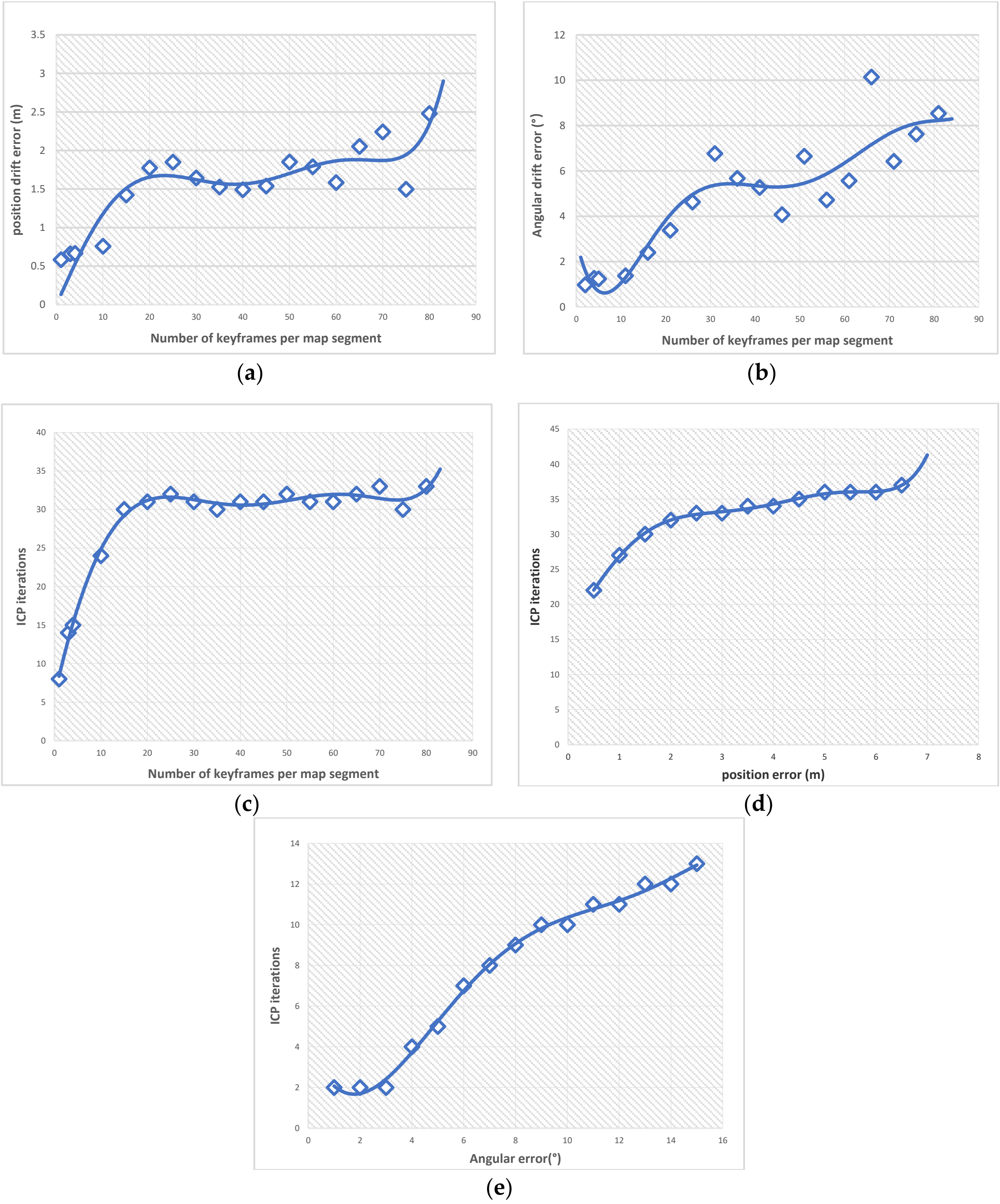
Table 9. Computation time for creating one keyframe in visual tracking thread.

Operation	Medium (ms)	Mean (ms)	Std. Dev. (ms)
Receiving a frame	33.33	33.33	33.33
Creation of keyframe	552.90	553.18	157.33

5.2. Keyframe-Based LiDAR Optimization in a Multithreaded Framework

In terms of accuracy, the ICP iterative algorithm can meet the global optimum although the geometrical information of the scene is not rich; for instance, when the scene has only one or several planes of cylindrical geometry and the pose of the ICP iterative calculation has a good constraint on the direction of the camera. Therefore, the geometric information of the scene can be used to constrain the visual SLAM. The local LiDAR corrected pose is propagated into local map construction, which can then provide an appropriate initial value for the map segmentation iterative optimization algorithm, and make the map optimization quickly reach the global optimum.

After conducting a large number of experiments under normal tracking in different scenes, we obtained the correlation between the number of keyframes per map segment and the absolute cumulative error obtained by registration with LiDAR data, the ICP iterations, and correlation between the ICP iterations and the absolute cumulative error. As seen from Figure 20a,b, as the number of keyframes increases, the cumulative errors increase. When the number of keyframes per map segment is within 80 frames, the position error is generally below 2.5 m, angular error is below 10°, and the ICP iterative algorithm can quickly converge to the global optimal value with at most 35 in the number of iterations, as shown in Figure 20e. However, as the cumulative error increases, the locally acquired LiDAR data may not overlap with the point cloud generated by the depth image, resulting in the ICP algorithm acquiring an inaccurate camera pose.



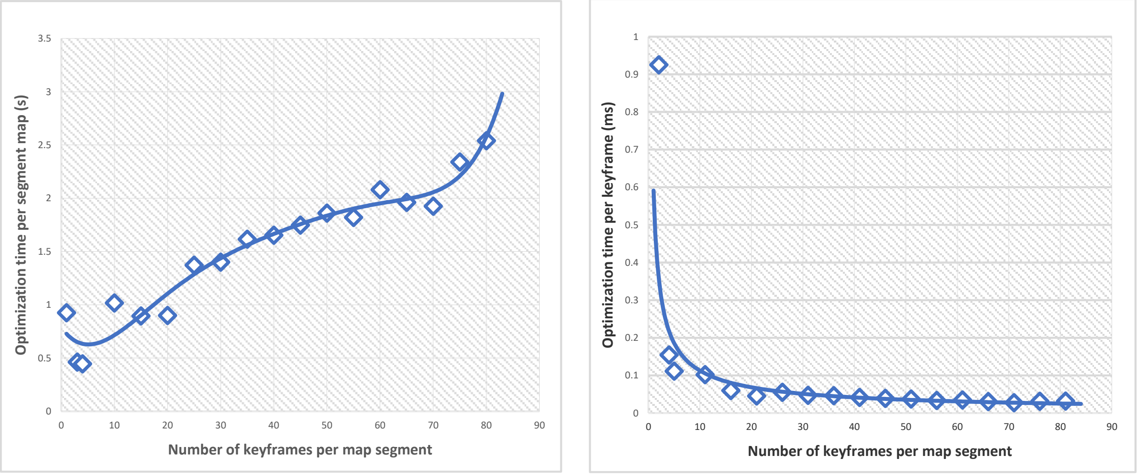
(**a**) (**b**)

(**c**) (**d**)

(**e**)

Figure 20. Correlation between the number of keyframes per map segment and cumulative error as well as LiDAR data optimization time. (**a**) Correlation between position drift error and number of keyframes per map segment, (**b**) correlation between angular drift error and number of keyframes per map segment, (**c**) correlation between ICP iterations and number of keyframes per map segment, and (**d**) correlation between ICP iterations and position error; (**e**) correlation between ICP iterations and angular error.

In terms of real-time performance, the voxel downsampling of a certain resolution of the LiDAR data of a large scene greatly reduces the amount of data used. Furthermore, LiDAR data retrieval for large scenes requires high computation time. In this study, the pose obtained by camera tracking is used as the prerequisite for LiDAR data retrieval, and the LiDAR data is organized by the KD-tree method, which greatly reduces the computation time. In addition, the point cloud data collected by RGB-D is downsampled with a certain resolution to reduce the number of point clouds participating in the ICP iteration. The adopted map point culling strategy of the front window and the keyframe culling strategy of the back window eliminate the redundant optimization variables, and the calculation time of the Jacoby matrix and the Hessian matrix is thus reduced. The multithread segmentation optimization is applied to avoid global optimization of the overall map, ensuring the optimization time is within a certain threshold, and ensuring the optimized results are shared by each thread in real time. As seen from Figure 21a,b, the average optimization time per keyframe is reduced with the increase in the number of keyframes per map segment. When the number of keyframes is less than 20 frames, the optimization time is above 75 ms. Therefore, from the real-time perspective of the algorithm operation, the number of keyframes per map segment should be set to at least 20 frames. After a comprehensive study of real-time operation and accuracy, we set the number of keyframes per map segment to 40 frames. Table 10 shows the time spent by LiDAR data optimization thread for every 40 keyframes; each operation in the table was performed once in every 40 keyframes. It is clear that the average time spent on each keyframe is very small. As seen from Table 11, the time spent by every keyframe in optimization operation is only ~40.20 ms. It must be noted that the tracking thread took about 550 ms to create a new keyframe, as shown in Table 9. These results clearly demonstrate the effective real-time performance of the proposed method.



(**a**) (**b**)

Figure 21. (**a**) Correlation between the overall optimization time of the map segment and the number of keyframes per map segment and (**b**) correlation between the average optimization time of each keyframe and the number of keyframes per map segment.

Table 10. Average computation time of each operation for every map segment (40 keyframes) in LiDAR data optimization thread.

	Operation	Medium (ms)	Mean (ms)	Std. Dev. (ms)
Multithreaded segment-based optimization for every 40 keyframes	Correction keyframe selection	0.143	0.208	0.02
	RGB-D points creation	100.05	100.05	80.5
	LiDAR points acquisition	121.90	121.90	98.8
	ICP operation	961.03	935.76	82.69
	Local propagation	99.43	94.19	49.10
	Segment-based optimization	335.55	274.09	88.76
	Pose transfer and Map fusion	79.20	81.86	50.04

Table 11. Average optimization time per keyframe in LiDAR correction thread.

Operation	Medium (ms)	Mean (ms)	Std. Dev. (ms)
LiDAR optimization	44.92	40.20	11.26

In terms of consistency, the map segmentation optimization had a 25% overlap with the previous segment of the map, whereas the keyframe poses at both ends of the map remained fixed as true value constraints. Using the double window optimization strategy, the front window uses the map point-pose constraint to construct an accurate map. The back window adopts the map point-position constraint and the pose–pose constraint to ensure front and rear map segment consistency.

In terms of practicability, from the above experiments and analyses of real-time operation, accuracy, and consistency, it has been demonstrated that the proposed algorithm can accurately map and locate most indoor scenes in real-time, which is beneficial in large indoor scenes such as airports and shopping malls. With advancements in sensor technology, cost-effective and portable depth cameras will become popular. In addition to positioning and mapping of the surrounding environment in real-time, users carrying depth cameras can avail more services and experiences, such as augmented reality online games, path planning, and intelligent interaction with the environment, based on precise map and camera pose. Although the high-precision LiDAR data for large indoor scene are difficult to obtain, the data once obtained can be used permanently after subsequent processing; in addition, all users in the environment can share this data for high-precision positioning and mapping. Even in case the scene changes for any reason, the LiDAR data of the scene can be automatically updated using the high-precision model produced by the proposed algorithm, eliminating the need for rescanning.

6. Conclusions

A novel RGB-D SLAM approach that uses priori LiDAR point cloud data as the guidance for indoor 3-D scene construction and navigation was presented in this paper. There were three important innovations in this method: first, the high-precision geometrical control information from the LiDAR data of the indoor scene was used to eliminate the cumulative error of the RGB-D SLAM system in large-scale indoor operation; second, a combination of deep learning and geometric SLAM system enabled automatic initialization and motion recovery, which improved the robustness of the system in challenging environments; and, third, the multithreaded map segment optimization method using double window was adopted to ensure the algorithm runs in real-time with high precision under the guidance of LiDAR data and constructed large-scale accurate and consistent dense map, which can patch the LiDAR data holes. The standard deviation of the 3D map construction was approximately 10 cm in a travel distance of 100 m compared with the LiDAR ground truth, and the relative cumulative error of the camera in closed-loop experiments was 0.09%, which was twice less than that of the typical SLAM algorithm (3.4%). Thus, it is clear that the positioning accuracy and map construction accuracy of the proposed system are significantly higher than those of the most advanced SLAM systems that do not utilize priori data. Furthermore, the fusion of the 2-D rich texture image data from RGB-D camera and the high-precision 3-D terrestrial LiDAR point cloud data was performed automatically; it compensated the shortcomings of the LiDAR data, which has high precision but insufficient texture information. Moreover, the proposed method can effectively construct dense indoor maps of large indoor scenes under the unified coordinate system of multiple data sources, for scene recognition and understanding, robot path planning and navigation, artificial intelligence (e.g., robot and 3D environment interaction), augmented reality, and virtual game environment development.

In future, we will build semantic maps of large scenes based on the constraints of terrestrial LiDAR data. Accurate poses obtained from terrestrial LiDAR data assist accurate semantic segmentation of single-frame images and enhance the consistency of semantic segmentation of multiview image frames so that robots can interact with the environment better.

Author Contributions: Conceptualization: J.L.; methodology: J.L. and X.K.; software: X.K., J.L., and W.W.; validation: X.K.; formal analysis, X.K.; investigation, X.K. and J.L.; resources, J.L. and X.F.; data curation, J.L. and X.K.; writing—J.L. and X.K.; writing—review and editing, J.L.; visualization, X.K.; supervision, J.L. and X.F.; project administration, J.L. and X.F.; funding acquisition, J.L. and X.F.

Funding: This work was funded and supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA19080101). This work was also partially supported by Key Deployment Program of Chinese Academy of Sciences (No. KFZD-SW-317), the National Key Research and Development Program of China (No. 2016YFB0502102), and the National Natural Science Foundation of China (No. 41101436).

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

References

1. Rodriguez-Losada, D.; Segundo, P.S.; Matia, F.; Galan, R.; Jiménez, A.; Pedraza, L. Dual of the factored solution to the simultaneous localization and mapping problem. *IFAC Proc. Vol.* **2007**, *40*, 542–547.
2. Durrant-whyte, H.F.; Bailey, T. Simultaneous Localization and Mapping. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110.
3. Fuentes-Pacheco, J.; Ruiz-Ascencio, J.; Rendón-Mancha, J.M. *Visual Simultaneous Localization and Mapping: A Survey*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2015; pp. 55–81.
4. Dailey, M.N.; Parnichkun, M. Simultaneous Localization and Mapping with Stereo Vision. In Proceedings of the 2006 9th International Conference on Control, Automation, Robotics and Vision, Singapore, 5–8 December 2006; pp. 1–6.
5. Dissanayake, M.W.M.G.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* **2001**, *17*, 229–241. [CrossRef]
6. Hu, G.; Huang, S.; Zhao, L.; Alempijevic, A.; Dissanayake, G. A robust RGB-D SLAM algorithm. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems, Vilamoura, Portugal, 7–12 October 2012.
7. Kerl, C.; Sturm, J.; Cremers, D. Dense visual SLAM for RGB-D cameras. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 2100–2106.
8. Baglietto, M.; Sgorbissa, A.; Verda, D.; Zaccaria, R. Human navigation and mapping with a 6DOF IMU and a laser scanner. *Robot. Auton. Syst.* **2011**, *59*, 1060–1069. [CrossRef]
9. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
10. Bay, H.; Ess, A.; Tuytelaars, T.; Gool, L.V. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359.
11. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary Robust invariant scalable keypoints. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011.
12. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G.R. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011.
13. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 1–10.
14. Galvez-Lopez, D.; Tardos, J.D. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Trans. Robot.* **2012**, *28*, 1188–1197. [CrossRef]
15. Celik, K.; Chung, S.J.; Clausman, M.; Somani, A.K. Monocular Vision SLAM for Indoor Aerial Vehicles. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems, St. Louis, MO, USA, 10–15 October 2009.
16. Davison, A.J.; Reid, I.D.; Molton, N.D.; Olivier, S. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [CrossRef]
17. Kai, W.; Kaichang, D.; Xun, S.; Wenhui, W.; Zhaoqin, L. Enhanced monocular visual odometry integrated with laser distance meter for astronaut navigation. *Sensors* **2014**, *14*, 4981–5003.
18. Zou, D.; Tan, P. CoSLAM: Collaborative visual SLAM in dynamic environments. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 354–366. [CrossRef]

19. Moratuwage, D.; Wang, D.; Rao, A.; Senarathne, N.; Han, W. RFS collaborative multivehicle SLAM: SLAM in dynamic high-clutter environments. *IEEE Robot. Autom. Mag.* **2014**, *21*, 53–59. [CrossRef]
20. Liu, T.; Zhang, X.; Wei, Z.; Yuan, Z. A robust fusion method for RGB-D SLAM. In Proceedings of the 2013 Chinese Automation Congress, Changsha, China, 7–8 November 2013; pp. 474–481.
21. Klüssendorff, J.H.; Hartmann, J.; Forouher, D.; Maehle, E. Graph-based visual SLAM and visual odometry using an RGB-D camera. In Proceedings of the 9th International Workshop on Robot Motion and Control, Kuslin, Poland, 3–5 July 2013; pp. 288–293.
22. Chen, H.; Lin, C. RGB-D sensor based real-time 6DoF-SLAM. In Proceedings of the 2014 International Conference on Advanced Robotics and Intelligent Systems (ARIS), Taiwan, China, 6–8 June 2014; pp. 61–65.
23. Chow, J.C.K.; Lichti, D.D.; Hol, J.D.; Bellusci, G.; Luinge, H. IMU and Multiple RGB-D Camera Fusion for Assisting Indoor Stop-and-Go 3D Terrestrial Laser Scanning. *Robotics* **2014**, *3*, 247–280. [CrossRef]
24. Deilamsalehy, H.; Havens, T.C. Sensor fused three-dimensional localization using IMU, camera and LiDAR. In Proceedings of the 2016 IEEE SENSORS, Orlando, FL, USA, 30 October–3 November 2016; pp. 1–3.
25. Qayyum, U.; Ahsan, Q.; Mahmood, Z. IMU aided RGB-D SLAM. In Proceedings of the 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 10–14 January 2017; pp. 337–341.
26. Kim, D.H.; Kim, J.H. *Image-Based ICP Algorithm for Visual Odometry Using a RGB-D Sensor in a Dynamic Environment*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 423–430.
27. Steinbrücker, F.; Sturm, J.; Cremers, D. Real-time visual odometry from dense RGB-D images. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Barcelona, Spain, 6–13 November 2011; pp. 719–722.
28. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
29. Mishima, M.; Uchiyama, H.; Thomas, D.; Taniguchi, R.I.; Roberto, R.; Lima, J.P.; Teichrieb, V. RGB-D SLAM Based Incremental Cuboid Modeling. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
30. Bescos, B.; Fácil, J.M.; Civera, J.; Neira, J. DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. *IEEE Robot. Autom. Lett.* **2018**, *3*, 4076–4083. [CrossRef]
31. Sun, Q.; Yuan, J.; Zhang, X.; Sun, F. RGB-D SLAM in Indoor Environments with STING-Based Plane Feature Extraction. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 1071–1082. [CrossRef]
32. Zhou, Y.; Li, H.; Kneip, L. Canny-VO: Visual Odometry with RGB-D Cameras Based on Geometric 3-D–2-D Edge Alignment. *IEEE Trans. Robot.* **2019**, *35*, 184–199. [CrossRef]
33. Cheng, Z.; Wang, G. Real-Time RGB-D SLAM with Points and Lines. In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 25–27 May 2018; pp. 119–122.
34. Tang, S.; Li, Y.; Yuan, Z.; Li, X.; Guo, R.; Zhang, Y.; Wang, W. A Vertex-to-Edge Weighted Closed-Form Method for Dense RGB-D Indoor SLAM. *IEEE Access* **2019**, *7*, 32019–32029. [CrossRef]
35. Han, L.; Xu, L.; Bobkov, D.; Steinbach, E.; Fang, L. Real-Time Global Registration for Globally Consistent RGB-D SLAM. *IEEE Trans. Robot.* **2019**, *35*, 498–508. [CrossRef]
36. Ji, Z.; Singh, S. Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015.
37. Sarvrood, Y.B.; Hosseinyalamdary, S.; Yang, G. Visual-LiDAR odometry aided by reduced IMU. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 3. [CrossRef]
38. Taketomi, T.; Sato, T.; Yokoya, N. Real-time and accurate extrinsic camera parameter estimation using feature landmark database for augmented reality. *Comput. Graph.* **2011**, *35*, 768–777. [CrossRef]
39. Caselitz, T.; Steder, B.; Ruhnke, M.; Burgard, W. Monocular camera localization in 3D LiDAR maps. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 1926–1931.
40. Wolcott, R.W.; Eustice, R.M. Visual localization within LIDAR maps for automated urban driving. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 176–183.

41. Gee, T.; James, J.; Mark, W.V.D.; Delmas, P.; Gimel'farb, G. Lidar guided stereo simultaneous localization and mapping (SLAM) for UAV outdoor 3-D scene reconstruction. In Proceedings of the 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ), Palmerston North, New Zealand, 21–22 November 2016; pp. 1–6.
42. Xia, Y.; Li, J.; Qi, L.; Yu, H.; Dong, J. An Evaluation of Deep Learning in Loop Closure Detection for Visual SLAM. In Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 21–23 June 2017; pp. 85–91.
43. Zhang, X.; Su, Y.; Zhu, X. Loop closure detection for visual SLAM systems using convolutional neural network. In Proceedings of the 2017 23rd International Conference on Automation and Computing (ICAC), Huddersfield, UK, 7–8 September 2017; pp. 1–6.
44. Zhang, L.; Wei, L.; Shen, P.; Wei, W.; Zhu, G.; Song, J. Semantic SLAM Based on Object Detection and Improved Octomap. *IEEE Access* **2018**, *6*, 75545–75559. [CrossRef]
45. Yu, C.; Liu, Z.; Liu, X.; Xie, F.; Yang, Y.; Wei, Q.; Fei, Q. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1168–1174.
46. Li, J.; Wang, C.; Kang, X.; Zhao, Q. Camera localization for augmented reality and indoor positioning: A vision-based 3D feature database approach. *Int. J. Digit. Earth* **2019**. [CrossRef]