electronics

MDPI

Article

A Monocular Visual Localization Algorithm for Large-Scale Indoor Environments through Matching a Prior Semantic Map

Tianyi Lu, Yafei Liu, Yuan Yang, Huiqing Wang and Xiaoguo Zhang *

School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China
* Correspondence: xgzhang@seu.edu.cn

Abstract: It is challenging for a visual SLAM system to keep long-term precise and robust localization ability in a large-scale indoor environment since there is a low probability of the occurrence of loop closure. Aiming to solve this problem, we propose a monocular visual localization algorithm for large-scale indoor environments through matching a prior semantic map. In the approach, the line features of certain semantic objects observed by the monocular camera are extracted in real time. A cost function is proposed to represent the difference between the observed objects and the matched semantic objects in the preexisting semantic map. After that, a bundle adjustment model integrating the semantic object matching difference is given to optimize the pose of the camera and the real-time environment map. Finally, test cases are designed to evaluate the performance of our approach, in which the line features with semantic information are extracted in advance to build the semantic map for matching in real time. The test results show that the positioning accuracy of our method is improved in large-scale indoor navigation.

Keywords: large-scale indoor localization; visual-SLAM; semantic map; bundle adjustment

Citation: Lu, T.; Liu, Y.; Yang, Y.; Wang, H.; Zhang, X. A Monocular Visual Localization Algorithm for Large-Scale Indoor Environments through Matching a Prior Semantic Map. *Electronics* **2022**, *11*, 3396. https://doi.org/10.3390/electronics11203396

Academic Editor: Hamid Reza Karimi

Received: 30 August 2022
Accepted: 17 October 2022
Published: 20 October 2022

1. Introduction

Visual SLAM localization plays an increasingly important role in indoor navigation [1]. However, it generally relies on loop closures to remove accumulated errors. For large-scale indoor scenarios such as airports, shopping malls, and museums, it is challenging for a visual SLAM system to maintain long-term precise and robust localization ability since there is a low probability of the occurrence of loop closures. Figure 1 shows a typical framework of a visual SLAM system.

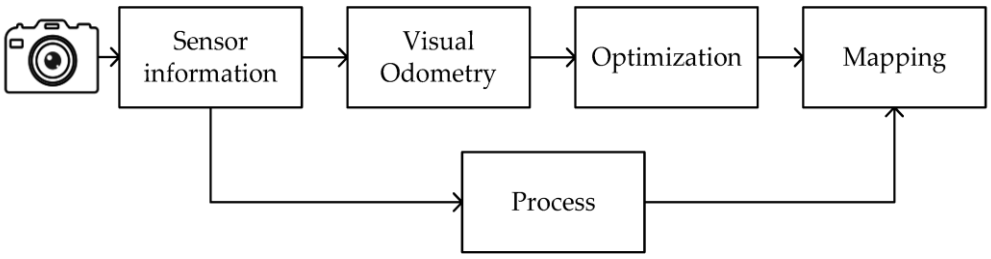
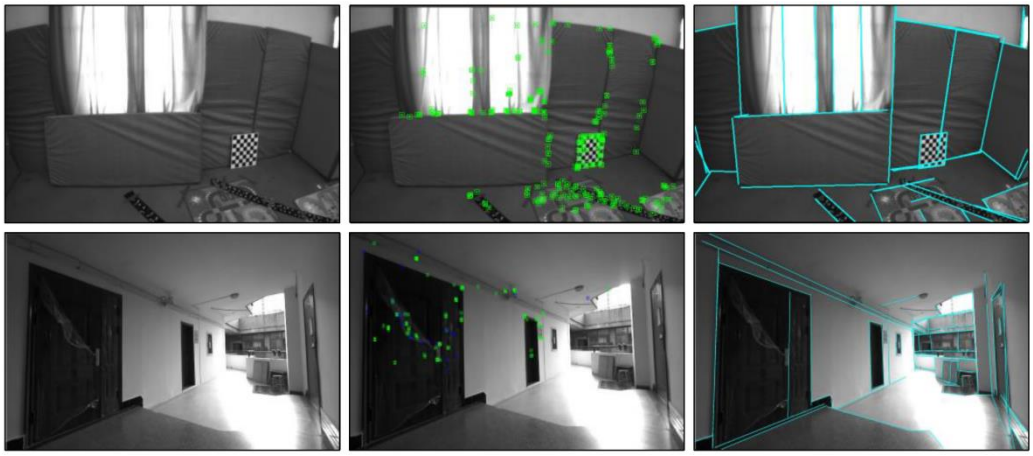


Figure 1. A typical framework of a visual SLAM system.

As shown in Figure 2, line features are more robust and more powerful to represent the environment than point features in the same scene [2,3]. Various approaches have been proposed to provide better indoor localization performance by using line features in the bundle adjustment model in SLAM systems for localization in a low-texture environment [4–7]. Gomez-Ojeda et al. designed a SLAM algorithm that involves line features in the optimization model to enable the whole system to achieve a more stable estimation of positional information in relatively low-texture scenarios [4]. Pumarola and colleagues

added line feature representations to the SLAM mechanism and proposed a new method for approximating the initialized map based on the corresponding line features [5]. Considering the occlusion and disconnection of line features, Gomez-Ojeda et al. compared the direction and length of line features and eliminated outliers [6]. He et al. proposed a tight-coupled monocular visual–inertial odometer system by integrating point and line features [7]. By using line features, or integrating point and line features, the aforementioned approaches generally are able to keep a relatively sufficient localization ability in a low-texture environment. However, since the SLAM algorithm itself is theoretically an integral computation model, it generally depends on loop closure to remove the accumulated errors. In large-scale indoor navigation scenarios, the error of the SLAM system generally will become increasingly bigger if there is no loop closure or an absolute position injection for a long time.



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

Figure 2. Results of the extracted point and line feature: (**a**) the original images; (**b**) the point feature detection results; (**c**) the line feature detection results.

Therefore, it is an urgent challenge to realize long-term precise and robust large-scale indoor navigation when there is a low probability of the occurrence of loop closure [8,9]. In the field of vehicle navigation, numerous map-matching approaches were proposed to remove accumulated vehicle positioning errors when the GNSS signal is blocked [10,11], which can generally provide persistent positioning ability even without any GNSS service for 10 min or more. Map matching, assuming there are similarities between the vehicle's track and the shape of the road network, is able to remove the error in the vertical direction of the road at any time and the bi-direction error generally when the vehicle makes a turn. Based on a similar idea, continuous and high-precision localization can be achieved in large-scale indoor scenes without loop closure by matching the SLAM reconstructed environment and the existing semantic map model data (such as BIM models) in real time.

Aiming at the problem, we propose a monocular visual localization method for providing long-term precise and robust localization ability in large-scale indoor scenarios. Compared with previous methods, a prior semantic map which can fully utilize the environmental information is used to solve the problem that error accumulation cannot be effectively eliminated in large-scale indoor environments because of the lack of loop closure. In this approach, the line features of certain semantic objects observed by the monocular camera are extracted in real time. A cost function is designed for indicating the difference between the observed objects and the matched semantic objects in the preexisting semantic map. After that, a bundle adjustment model integrating the object matching difference information is proposed to optimize the pose of the camera and the real-time environment map. Finally, a test design is given to evaluate the performance of the proposed approach, in which the line features with semantic information are extracted in advance to build the semantic map for matching in real time. The test results show that the positioning accuracy

of our method is improved in large-scale indoor localization. The main contributions are as follows:

1. A method for maintaining high-precision persistent localization capability through matching a prior semantic map with line features is proposed. In this method, semantic objects are identified in the environment in real time and the key line features corresponding to the objects are extracted. Subsequently, the semantic objects identified in real time are matched with the line features of the corresponding objects in the prior map, effectively associating each key frame with the previous semantic map.
2. A bundle adjustment model integrating the semantic object matching information using line features is proposed to achieve higher localization accuracy and robustness performance in large scenes.
3. To verify the performance of the proposed approach, we designed test cases and propose a method to build the prior semantic map for the real-time object matching. The test results show that our method reduces the drift of monocular vision in a large-scale and low-texture environment and effectively improves the accuracy of localization.

The remainder of this paper is organized into four parts. Firstly, Section 2 reviews the literature and related works. Then, the specific principles and implementation details of the algorithm are given in Section 3. Section 4 introduces the test design and the pre-processing of the prior semantic map, and analyzes the comparison of experimental results with classical and state-of-the-art methods. Finally, the conclusions and potential future research directions are summarized in Section 5.

2. Related Work

SLAM systems can be divided into two categories: filter-based [12–14] and factor graph model-based [15–17]. Now, the factor-graph-based optimization approach has become mainstream. Regarding factor graph optimization models, the most frequently used methods are based on point features [18,19], point-line fusion [20,21], or the simultaneous use of point-line surface features based on the Manhattan assumption [22,23]. Theoretically, the factor graph optimization methods can work in two scenarios: one is a local optimization using co-visual constraints of wayfinding points [24,25], and the other is a global optimization using the loop closure constraint [26,27]. Although local optimization can improve localization accuracy, global optimization based on loop closure is indispensable to achieving continuous high-precision localization over a longer period of time.

If there is a low loop closure probability in large-scale indoor scenarios such as an airport or a shopping mall, the accuracy of the V-SLAM system could be severely downgraded. Existing studies usually try to reduce the positioning error by combining inertial devices such as IMU or wireless positioning methods such as WIFI/UWB [28,29], but the low-cost IMU has the same accumulated error in terms of parameter drift, and the WIFI/UWB also could fail to solve this problem even though additional nodes and sensors are used.

Recently, researchers have tried to use high-precision prior maps to aid real-time visual localization. Such methods, basically based on point cloud matching, can be divided into two main categories: (1) image information-based matching [30–32] and (2) geometry information-based matching [33,34].

2.1. Image Information-Based Matching

In terms of matching based on image information, Pascoe et al. focused on a monocular camera localization algorithm performed in a textured three-dimensional prior mesh [28]. In the approach, a synthetic image generated from the previous best synthetic image was matched with the real-time camera image to optimize the current pose. Neubert et al. integrated the 3D distance information on the map with the current visual image on the robot camera by synthesizing the depth image into Monte Carlo localization to track a given target trajectory [31]. Wolcott saved the point cloud map as a raster image, discarding the height information, and measured the vehicle captured on the static 3D point cloud map for overall image alignment to obtain the positional information [32]. However, given the raw

depth information of the 3D map, this kind of localization approach requires additional information, such as laser intensity [31].

2.2. Geometry Information-Based Matching

Caselitz et al. gave a visual odometry system based on the local bundle adjustment to reconstruct a sparse set of 3D points from image features and built a positional optimization model using the spatial location correlation of two kinds of point clouds [31]. Since the algorithm relies only on matched geometry, it is insensitive to the luminance variation of ambient light. The accuracy of the algorithm is highly correlated with the precision of the reconstructed sparse points, and it is difficult to avoid the large errors of deconstructed sparse points in long-term localization. Kim et al. matched the depth in the stereo parallax map with the 3D LiDAR map by using a binocular camera; in the method, the six-degree-of-freedom (DOF) camera pose was estimated by minimizing the depth residuals [34]. It is noted that this paper proposed the concept of depth residuals and used depth residuals instead of photometric errors to achieve sufficient localization results. The accuracy of the above algorithms is related to the accuracy of the reconstructed point clouds, which could be affected by the lighting or viewpoint. In a large scene, the increase in cumulative error of the reconstruction results will cause lower accuracy of map matching.

To avoid the negative effect of viewpoint and lighting conditions, Gawel et al. performed semi-dense environment reconstruction using ORB-SLAM2 and used structural information to perform 3D feature matching [35]. This algorithm requires only point cloud data from different light sources as input, independent of specific visual features. Zou et al. developed a low-cost stereo visual–inertial localization system providing bounded-error 3D navigation using LiDAR maps [36]. In this method, the registrations of visual semi-dense reconstruction and LiDAR maps are used to update the multi-state constraint Kalman filter (MSCKF) to correct the accumulated errors.

Overall, the above studies ignored the case of no loop closure correction in large scenarios. In this paper, a monocular visual localization algorithm for large-scale indoor environments through matching a prior semantic map is proposed. In the approach, the line features of certain semantic objects observed by the monocular camera are extracted in real time. A cost function is proposed for the evaluation of the difference between the observed objects and the matched semantic objects in the preexisting semantic map, and a bundle adjustment method is proposed to optimize the pose of the camera and the real-time environment map based on the given cost function. Finally, we give a test design to evaluate the performance of the proposed approach, in which the line features with semantic information are extracted in advance to build the semantic map for matching in real time. The results show that the positioning accuracy is improved in large-range indoor localization and the drift error of monocular vision sensors within a limited amount of computation is effectively reduced.

3. Algorithmic Approach

3.1. Overall Framework

To improve the positioning accuracy in the low-texture environment without loop closure, a novel positioning algorithm is proposed, which matches the line features in certain types of those semantic objects recognized in real time with those in the prior semantic map. The structure of the proposed indoor localization algorithm is shown in Figure 3.

In the real-time visual tracking thread, the environmental semantic information is considered to constrain the camera poses for the purpose of correcting drift and avoiding positional jumps. In the process of visual localization, the visibility of the gate frame lines provided by the prior map is identified, and the 2D gate frame lines in the image are detected in real time and matched with the gate frame line features within the field of view in the map, so as to achieve the adjustment of the visually estimated poses.

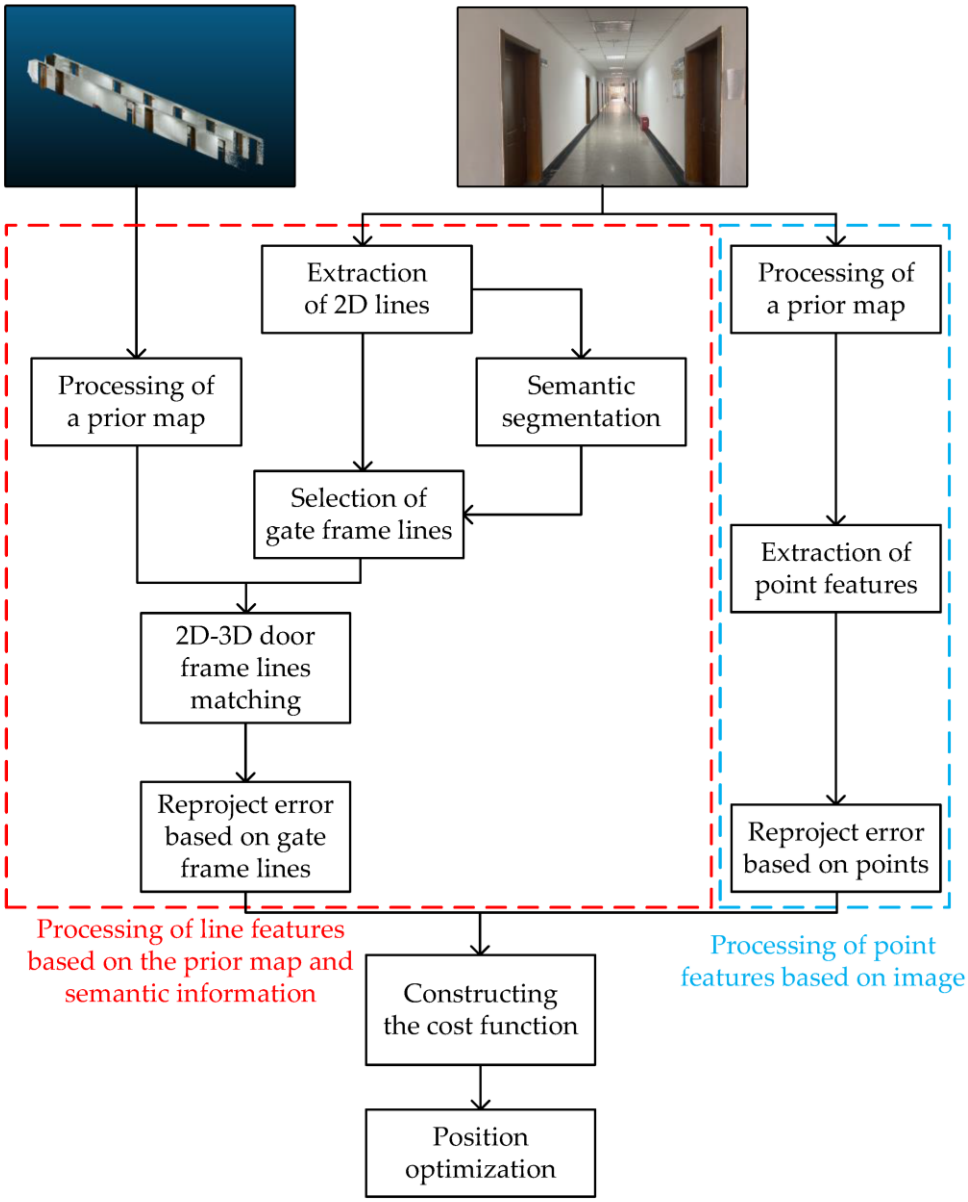


Figure 3. Block diagram of the proposed indoor location algorithm.

3.2. Extraction of 2D Gate Frame Lines in Real Time

3.2.1. Extraction of 2D Line

In our approach, we optimize the camera pose using the similarity information of line features in those semantic objects detected in real time and the features stored in the matched semantic objects in the prior map. The first step of our SLAM algorithm is to reconstruct the line features in the detected semantic objects, which means we should first extract lines in captured images. To simplify the representation of our algorithm, we take the gate object as the example of semantic object matching in the paper.

Due to the interference of texture noise signals, a large number of fragmented line segments could be generated during the process. Therefore, it is critical to extract line segments with high consistency with the length of the gate frame lines as much as possible. The M-LSD algorithm is able to extract line segments with better integrity [37] compared to the LSD algorithm and the Edline algorithm [38,39]. As shown in Figure 4c, this algorithm has adequate consistency with the geometric 3D structure (gate frame lines) and is robust to texture noise. Therefore, in this paper, the M-LSD algorithm is used to extract line features of those semantic objects.

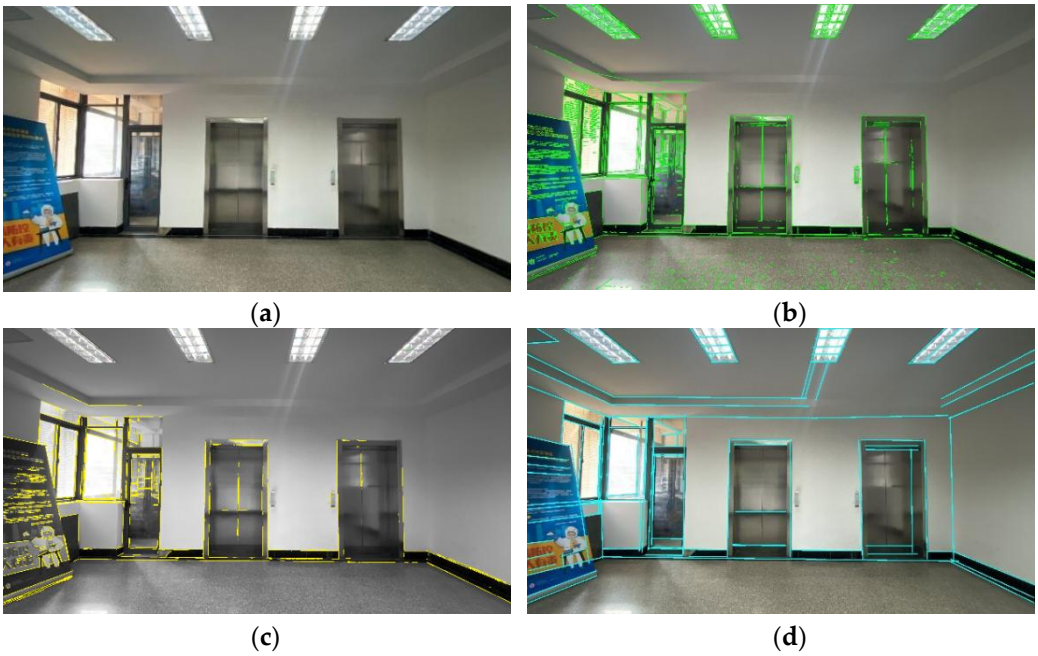


Figure 4. Results of different two-dimensional line segment detection methods: (**a**) the original image; (**b**) LSD; (**c**) ED-line; (**d**) M-LSD.

3.2.2. Selection of Gate Frame Lines

The semantic detection thread is added to the SLAM system, and 2D lines are extracted to select the gate frame lines in a single frame. In addition, in order to improve the speed of processing, two threads process the RGB images simultaneously to jointly complete the screening of the gate frame lines.

In our research, RGB images are semantically segmented using SegNet networks [40]. As shown in Figure 5, if an extracted line segment is within the color block corresponding to a gate, the corresponding semantic category label is given, and the line segment whose label category is 'gate' is reserved. The details of the process are as follows:

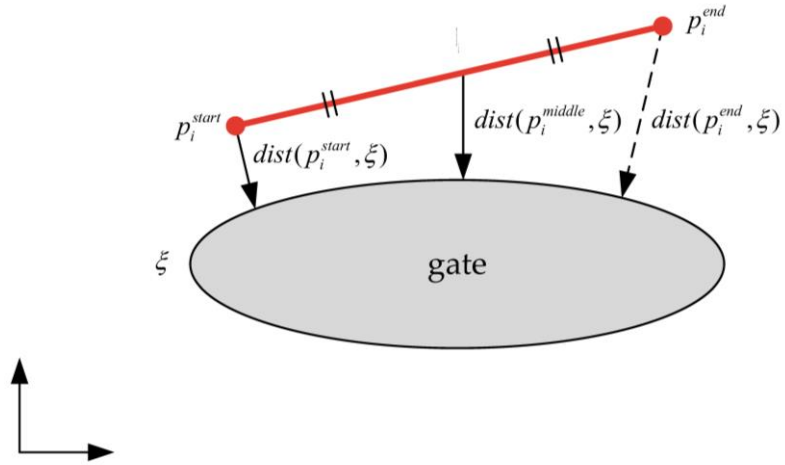


Figure 5. Merging of similar line segments.

1. Firstly, roughly remove the line segments that are too far from the boundary of the semantic category. Regarding the roughly identified line segments $V_c = \{l_1, l_2, \cdots, l_i\}$, the straight lines close to the semantic boundary curve ξ are retained.
2. The distances $\mathrm{dist}(p, \xi)$ from the two endpoints and the midpoint of the line segments to the semantic boundary ξ are calculated. Then, the maximum value of these three distances is discarded, and the smaller two values are summed and denoted as Σ_i.

When Σ_i is smaller than the threshold Σ_d, the line segment is considered to belong to the searched two-dimensional gate frame line segments V_d.

$$V_d = \{V_c : \forall l \in V_c, \Sigma_i < \Sigma_d\} \quad (1)$$

3. Finally, the similar line segments in V_d are merged and optimized. As shown in Figure 6, if the overlapping part O of the line segments in the *x*-axis or *y*-axis direction is larger than the threshold O_d, the angle α and β between the line segment and the coordinate axis are used as the judgment factors. If the value of $|\alpha - \beta|$ is less than the threshold γ_d, the similar line segments are considered to be merged. The endpoints with the top two maximum distances are denoted as the new line segment $V = \{l_1, l_2, \cdots, l_n\}$.

$$V = \{V_d : \forall l \in V_d, O < O_d, |\alpha - \beta| < \gamma_d\} \quad (2)$$

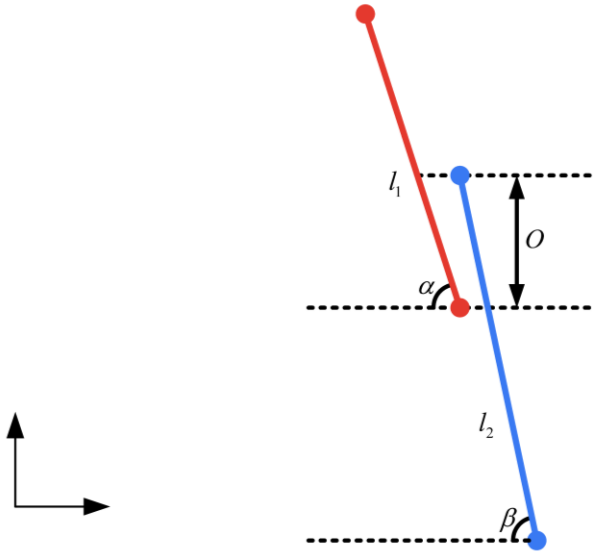


Figure 6. Merging of similar line segments.

3.3. The Model for Matching Existing Prior Semantic Map

To match the line features of the semantic objects with the corresponding line features in the prior map, two approaches could be used: (1) mapping the 3D lines to the image, then matching lines in 2D; (2) reconstructing 3D lines of those identified semantic objects in real time, and matching 3D lines in the world coordinate system. In this paper, we use the first approach. For the matching of 2D gate frame lines with the prior maps in a single frame image, in the case of a large scene, there is a problem of a large number of gate frame lines obtained by offline processing in the prior maps. This can seriously affect the computing efficiency and the computation speed. Therefore, the visibility of endpoints is used to select the 3D gate frame lines and remove the line segments which are not in the field of view in this frame. The endpoints X^{start} and X^{end} of the gate frame lines are extracted. Three types of situations are handled with different strategies:

1. If X^{start} and X^{end} are both within the field of view, this gate frame line is also considered to be within the field.
2. One of X^{start} or X^{end} is within the field of view. The point within the field is retained. If X^{middle} is within the field of view, the line segment $L = \left\{X^{start}, X^{end}\right\}$ is kept. If X^{middle} is not within the field of view, the process is repeated for the newly generated line segment until the length of the line segment is less than the threshold l_0.
3. If X^{start} and X^{end} are not in the field of view, then this gate frame line is discarded.

After analyzing the visibility of the 3D gate frame lines, 2D–3D gate lines will be matched. For the gate frame line $L = \left\{X^{start}, X^{end}\right\}$, its two endpoints are projected into the image plane, and the projected endpoints are denoted as $l_n = \{p_n^{start}, p_n^{end}\}$. The angle θ, the difference of length Δl, and the distance d between l_n and the two-dimensional line segment $l_c = \{p_c^{start}, p_c^{end}\}$ are computed.

The angle between l_c and l_n can be obtained using θ:

$$\theta = \arccos\left(\boldsymbol{\nu}_n \cdot \boldsymbol{\nu}_c\right) \tag{3}$$

$$\boldsymbol{\nu} = \frac{\left(p^{end} - p^{start}\right)}{\| p^{end} - p^{start} \|} \tag{4}$$

where ν_c and ν_n represent the normalized vectors of l_c and l_n, respectively, and the difference of the length can be described using the following equation:

$$\Delta l = |l_c - l_n| = \left| \sqrt{\left(x_c^{end} - x_c^{start}\right)^2 + \left(y_c^{end} - y_c^{start}\right)^2} - \sqrt{\left(x_n^{end} - x_n^{start}\right)^2 + \left(y_n^{end} - y_n^{start}\right)^2} \right| \tag{5}$$

where $l = \sqrt{\left(x^{end} - x^{start}\right)^2 + \left(y^{end} - y^{start}\right)^2}$.

Let the expression of the line where the two-dimensional line segment is located be equal to $Ax + By + C = 0$. The distance d between two line segments is defined as the distance between the two intersections of the perpendicular bisector of a 2D line segment, as shown in Figure 7.

$$d = \frac{\left|A \times \left(x_n^{end} + x_n^{start}\right) + B\left(y_n^{end} + y_n^{start}\right) + 2C\right|}{2 \times \sqrt{A^2 + B^2}} \tag{6}$$

where $A = y_c^{end} - y_c^{start}$; $B = x_c^{start} - x_c^{end}$; $C = x_c^{start} \times y_c^{end} - x_c^{end} \times y_c^{start}$.

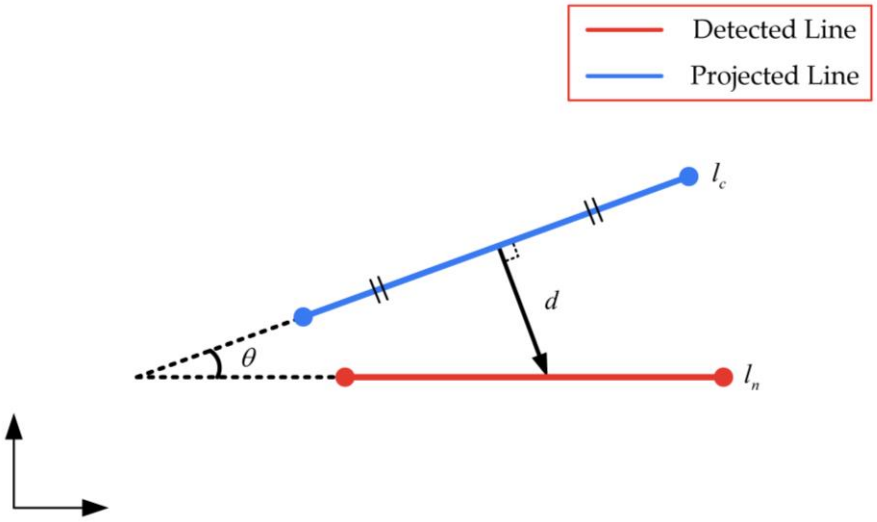


Figure 7. Distance between two line segments.

To complete the matching of 2D line segments with the prior map, all 3D gate frame lines within the field of view are traversed until matching conditions $\theta < \theta_0$, $\Delta l < \Delta l_0$, and $d < d_0$ are met.

3.4. Bundle Adjustment Integrating Linear Semantic Objects

The incremental maps given by the visual odometer could be inaccurate over long time periods due to unavoidable cumulative errors. To address this problem, an optimization model for matching the real-time map and a prior map is proposed in this section. The

error term of the matched gate frame lines in Section 3.3 is added to the bundle adjustment model. The motion between two consecutive frames is subsequently estimated iteratively to improve the positioning accuracy of the system. The details of this algorithm are as follows.

As shown in Figure 8, considering the constraint of the gate frame line in the prior map, a new error term $r_l\left(z_{L_i}^{c_k}, \chi\right)$ is defined as the sum of the distances from the two endpoints of the gate frame line after projection to the 2D plane to the line where the 2D line segment is located; then, for a single image frame, the residual of the i-th spatial line observed in the k-th camera frame could be written as:

$$r_l\left(z_{L_i}^{c_k}, \chi\right) = \sum_{i=1}^{N} \frac{\left|D_i \cdot Ke^{\xi} \cdot L_i\right|}{\sqrt{A_i{}^2 + B_i{}^2}} = \sum_{i=1}^{N} \frac{\left|A_i x_{n_i}^{end} + B_i y_{n_i}^{end} + C\right| + \left|A_i \times x_{n_i}^{start} + B_i y_{n_i}^{start} + C\right|}{\sqrt{A_i{}^2 + B_i{}^2}} \tag{7}$$

$$D_i = [A_i\ B_i\ C_i] = \left[\left(y_{c_i}^{end} - y_{c_i}^{start}\right)\left(x_{c_i}^{start} - x_{c_i}^{end}\right)\left(x_{c_i}^{start} \times y_{c_i}^{end} - x_{c_i}^{end} \times y_{c_i}^{start}\right)\right] \tag{8}$$

$$L_i = \left[X_{n_i}^{start}\ X_{n_i}^{end}\right] \tag{9}$$

where N is the number of line features in a single image frame, L_i contains the coordinates of the two endpoints of the three-dimensional gate frame line, and K is the number of previous frames in the sliding window. Then, the cost function corresponding to the final line feature part can be expressed as:

$$\begin{aligned} \sum_{(i,k)\in L} \| r_l\left(z_{L_i}^{c_k}, \chi\right) \|_{\Sigma_{L_i}^{c_k}}^2 &= \sum_{k=0}^{K} \sum_{i=1}^{N} \frac{\|D_i^k \cdot Ke^{\xi} \cdot L_i^k\|^2}{\sqrt{A_i^{k2} + B_i^{k2}}} \\ &= \sum_{k=0}^{K} \sum_{i=1}^{N} \| \frac{\left|A_i^k x_{n_i^k}^{end} + B_i^j y_{n_i^k}^{k\,end} + C\right| + \left|A_i^k \times x_{n_i^k}^{start} + B_i^k y_{n_i^k}^{j\,start} + C\right|}{\sqrt{A_i^{k2} + B_i^{k2}}} \|^2 \end{aligned} \tag{10}$$

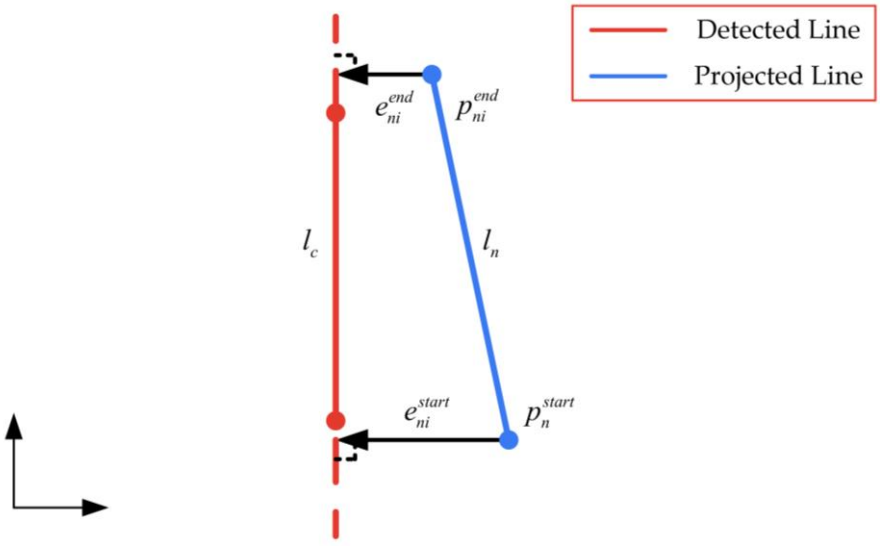


Figure 8. Error of line segment.

Ultimately, under the constraint of the prior semantic map, the cost function summing all error terms in the form of least squares can be written as:

$$\begin{aligned} \min_{\chi}\{\sum \| r_p - J_p\chi \|_{\Sigma_p}^2 + \sum_{k\in B} \| r_b\left(z_{b_k b_{k+1}}, \chi\right) \|_{\Sigma_{b_k b_{k+1}}}^2 \\ + \sum_{(j,k)\in F} \| r_f\left(z_{L_i}^{c_k}, \chi\right) \|_{\Sigma_{f_j}^{c_k}}^2 + \sum_{(i,k)\in L} \| r_l\left(z_{L_i}^{c_k}, \chi\right) \|_{\Sigma_{L_i}^{c_k}}^2 \} \end{aligned} \tag{11}$$

where the first term is the IMU residual item between adjacent frames in the sliding window, the second term is the marginalization residual item, and the third term is the in-camera visual reprojection residual of feature points in the sliding window. To iteratively optimize the camera's poses, Ceres Solver is used to minimize the cost function [41].

4. Experiment

To verify the performance of our proposed algorithm, comparative experiments in sequence images obtained from a monocular camera are conducted. Considering that there is no ground-truth trajectory for the captured sequence images, the trajectories of the laser SLAM algorithm with higher accuracy are used as the ground-truth data. The evaluation metrics are the absolute position error (absolute translation error, ATE) and the closure error of the trajectory, where the ATE is expressed as:

$$E_i = Q_i^{-1} S P_i \tag{12}$$

where $P_i \in SE(3), i = 1, \cdots, n$ is the estimated trajectory, $Q_i \in SE(3), i = 1, \cdots, n$ is the ground-truth trajectory (reference trajectory), and S is the rigid body transformation corresponding to the least square solution of the estimated trajectory $P_{1:m}$ mapped to the ground-truth trajectory (reference trajectory) $Q_{1:m}$. In most cases, the root mean square error (RMSE) of the ATE at each moment is mainly used as a criterion; the expression of RMSE is as follows:

$$RMSE(E_{i:n}) = \left(\frac{1}{m} \sum_{i=1}^{m} \| \text{ trans}(E_i) \|^2 \right)^{\frac{1}{2}} \tag{13}$$

The test environment for this experiment is an indoor corridor environment, and the ZED 2i camera is used as the data acquisition device to capture the test sequence with a resolution of $1280 * 720$. Before shooting, the ZED 2i left camera is calibrated with IMU, and the results are shown in Tables 1 and 2.

Table 1. Internal parameters of ZED 2i's left camera.

Parameter	f_x	f_y	c_x	c_y
Value	534.53	534.60	637.52	346.08
Parameter	k_1	k_2	p_1	p_2
Value	−1.24	2.08	0.18	0.00

Table 2. Internal parameters of IMU.

Parameter	gyr_n	gyr_w	acc_n	acc_w
Value	$1.67262942 \times 10^{-3}$	$3.85294351 \times 10^{-6}$	$1.89324403 \times 10^{-3}$	$4.03542685 \times 10^{-5}$

The testing platform is a desktop with Intel Xeon Cold 5115 CPU and a Nvidia GeForce GTX 2080Ti GPU.

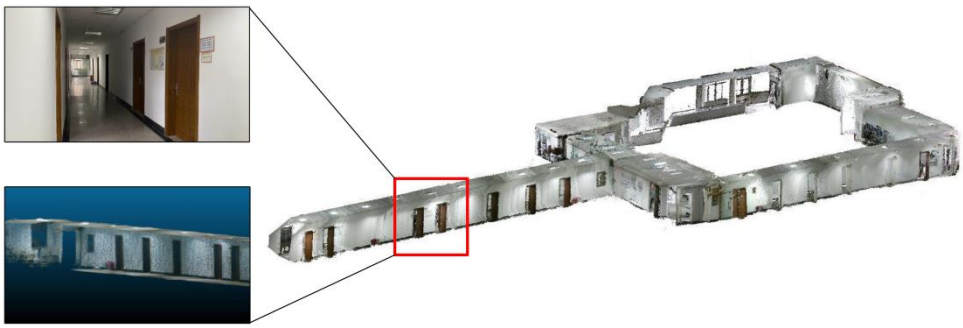
4.1. Acquisition of the Prior Semantic Map

Considering that there is no high-precision building information model (BIM) for existing indoor scenes, we propose a method for building the prior map using a reconstructed dense point cloud map. The dense point cloud is computed using ORB-SLAM2 [18], and the RGB images and depth maps of the scene are captured by moving in the corridor environment of the experiment with a handheld Kinect v2.0 camera. Subsequently, the captured information is used to construct a dataset as a prior map, and the dense point cloud information is stored in a ply format file after running the ORB-SLAM2 algorithm. Figure 9 shows our equipment for the data collection. To further evaluate our method under various situations, the experiments are divided into two parts. The first part is

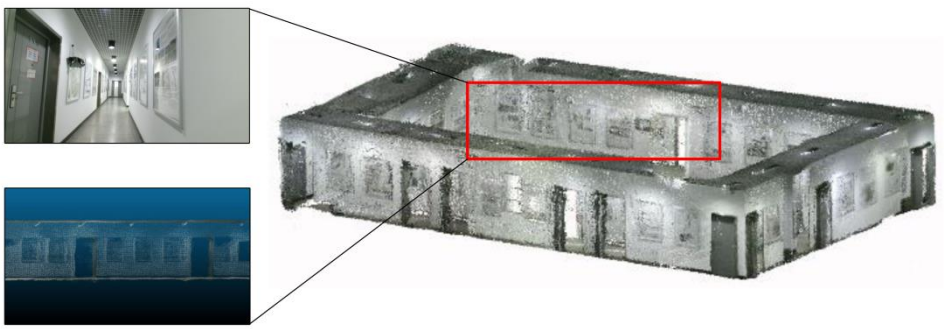
conducted under a structured corridor without any closed loop, as shown in Figure 10a, while the second part of the experiment is arranged to be executed in a circular corridor, as shown in Figure 10b.

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

Figure 9. Equipment used in the experiment: (**a**) ZED 2i camera; (**b**) Kinect v2.0 camera; (**c**) flat calibration board.



(**a**)



(**b**)

Figure 10. Prior maps obtained after reconstruction: (**a**) the unclosed loop corridor where experiment 1 was conducted; (**b**) the loop corridor where experiment 2 was conducted.

The point cloud map processing is divided into two steps: (1) a semantic segmentation algorithm is used to retain the local point cloud map with 'gate' labels, and (2) a 3D line segment detection algorithm is used to extract the gate frame lines required for subsequent matching.

To build the prior semantic map, a deep learning network named RandLA-Net is used to recognize environmental semantic information [42]. This network is an efficient semantic segmentation model for large-scale point clouds, open-sourced by Hu Q et al. in 2020, which can effectively reduce the computational cost and improve the operation speed of the system compared with PointNet++, PointCNN, and KPConv [43–45].

After the processing of RandLA-Net, the point clouds labeled as 'gate' are saved as a site point cloud map. The 3D lines in the point cloud labeled as 'gate' are extracted and filtered according to the length threshold to obtain the gate frame lines.

For the extraction of 3D lines, it is difficult for the planar-based extraction method to determine the boundaries of planes, and this method may produce unexpected lines on non-planar surfaces when the data become complex. The sharp feature-based extraction method is less robust because the regions with sharp features and noisy regions have similar high surface gradients.

Therefore, we use an image-based 3D line segment detection method to extract gate frame lines, and the details of the Algorithm 1 are as follows:

Algorithm 1: Image-based 3D line segment detection method

Input: -The local point cloud map with 'gate' label
Output: -The endpoints of each line p_s, p_e
1. Find the neighborhood IPi of each data point based on KNN
2. Estimation of the normals of adjacent surfaces nP using principal component analysis;
3. Extraction of local area R_i according to the area growth method;
4. **for** $i < num++i$
5. Calculate the normal deviation normalDev;
6. **if** $normalDev < thNormal$
7. **continue**;
8. **end if**
9. Calculate the orthogonal distance $dOrtho$;
10. **if** $dOrtho > thOrtho$
11. **continue**;
12. **end if**
13. Calculate parallel distance $dPara$;
14. **if** $dPara > thRadius$
15. **continue**;
16. **end if**
17. **end for**
18. Similar regions are merged to obtain a 3D planar group Π;
19. Projection to the plane Π_i from the point P_Π belonging to this plane, then transform into a binarized image
20. Extract the above binary image contour and use RANSAC to extract the contour line segment l_j;
21. **for** $j = 0\ j < numj++$
22. Calculate the pixel coordinates (u_i,v_i) of each point, corresponding to the two-dimensional plane coordinates (x_i,y_i);
23. Calculate each two-dimensional point (x_i,y_i), corresponding to the three-dimensional point P_i
24. **if** $j = 0\ ||\ j = num - 1$
25. Output the coordinates of endpoint
26. **end if**
27. **end for**

Figure 11 shows the processing with a section of the corridor as an example.

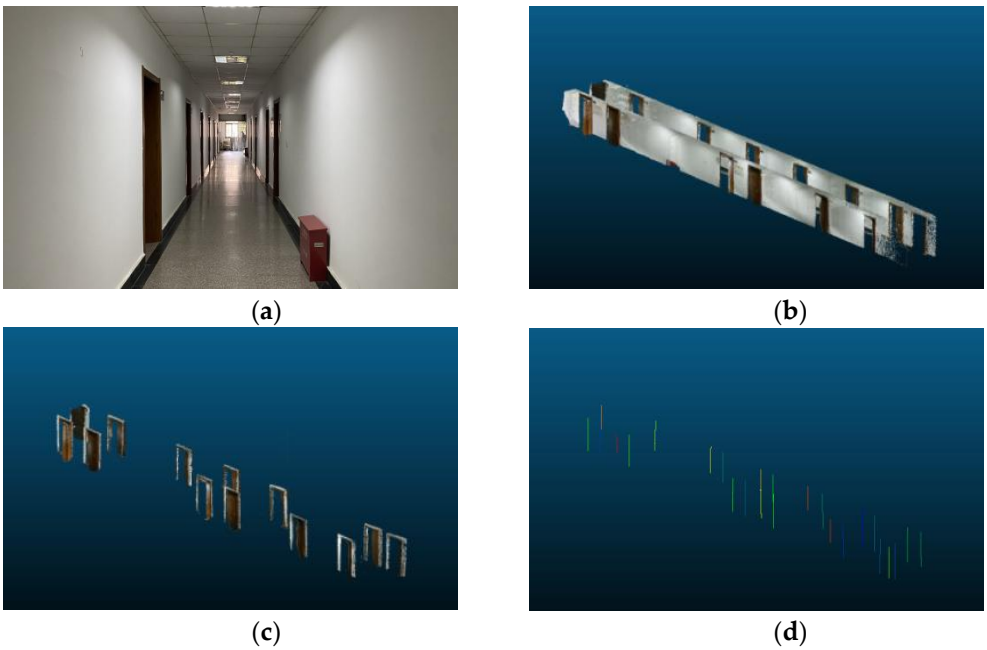


Figure 11. The point cloud map processing: (**a**) the original image; (**b**) the point cloud map of the corridor (partial); (**c**) after processing using RandLA-Net; (**d**) gate frame lines.

4.2. Absolute Position Error for Unclosed Trajectory

The LiDAR algorithm with high accuracy is chosen as the comparison reference in the experiments, and the reference trajectory is generated by LeGO-LOAM [46]. The ZED 2i camera is fixed to the same position with the LIDAR Robosense RS-LiDAR-16 to collect data, which ensures that the true motion trajectory is identical. Figure 12 shows the RS-LiDAR-16 we used during the experiment. To simulate the scenario with no closed loop, we do not pass by the same location during the recording of the dataset.

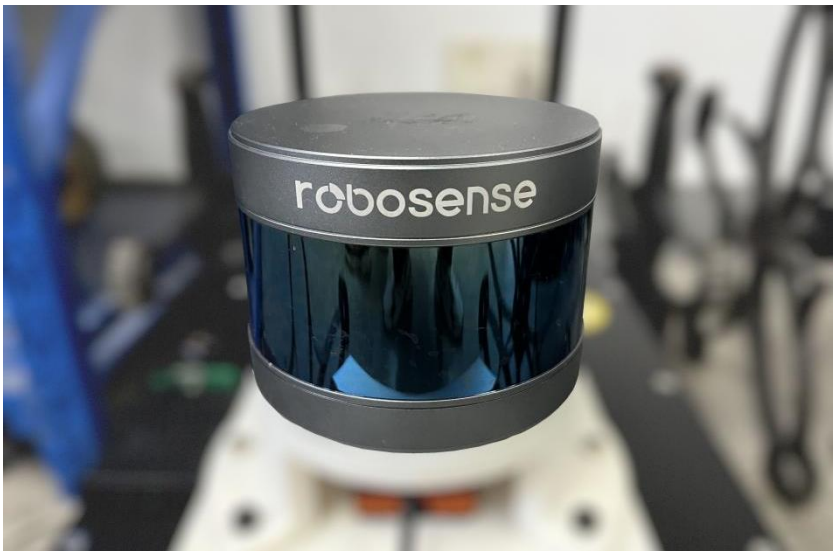
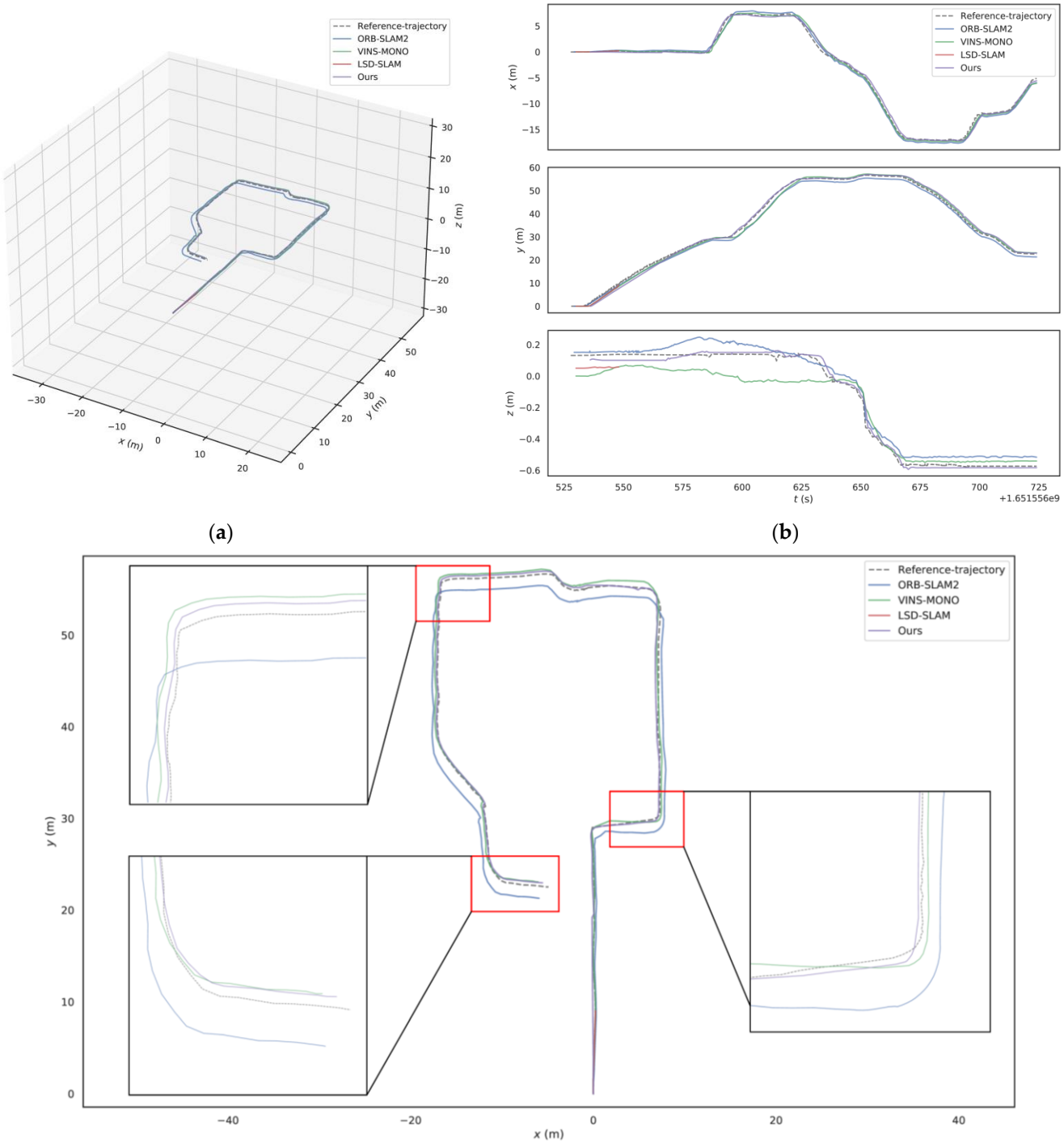


Figure 12. RS-LiDAR-16 used in our test.

The trajectories using the different algorithms and the accuracy on the XYZ axis are shown in Figure 13, where the dashed lines represent the reference trajectories. The ATEs using different algorithms are shown in Table 3, which contains the maximum value (MAX), minimum value (MIN), and root mean square error (RMSE).



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

(**c**)

Figure 13. Results of the unclosed trajectory experiment: (**a**) the trajectories of the proposed method (purple) and each comparison group; (**b**) the accuracy of the three coordinate axes of each method; (**c**) the projection of the trajectory of each algorithm to the xy-plane.

Table 3. Comparison of ATE among algorithms. Bold numbers represent the best performances.

ATE	ORB-SLAM2	VINS-MONO	LSD-SLAM	Ours
MAX	2.569	2.629	Lost	**1.072**
MIN	0.173	0.363		**0.027**
RMSE	0.369	0.559		**0.282**

From Table 3, it can be found that our method has adequate performance in the test environment, while LSD-SLAM does not perform as well [47], with tracking lost at about 10 m from the starting point in corners where the scene changes quickly. Our algorithm has the smallest ATE in the experiment, which effectively improves the positioning accuracy. The ATE RMSE of our method is 0.282, while those of ORB-SLAM2 and VINS-MONO are 0.369 and 0.559, respectively. Furthermore, as shown in Figure 13c, the trajectory of our method is closer to the reference trajectory at several corners. Thus, the performance of the proposed method is better than that of comparison algorithms when scenes change rapidly.

4.3. Closure Error of Circular Trajectory

The experiments were conducted inside a circular building. Firstly, a landmark is placed at the starting point, and then the camera-loaded device is made to follow the corridor and return to the landmark after passing through four corners to obtain a rectangular trajectory. Because the ground truth is closed, the positioning accuracy can be expressed by the distance between the starting point and the endpoint. The smaller this distance is, the smaller the error is. For more realistic results, the loop closing technique is not used to correct the error during the experiment.

To verify the effect of path length on the closure error of different algorithms, experiments with different numbers of loops are conducted. It is noted that the short length of the trajectory will result in an insignificant closure error gap. On the other hand, a long length of the trajectory allows a sufficient contrast effect, but introduces the confusion of the chart. In the experimental validation, the numbers of path laps are determined as one and three, and the length of the path is about 53 and 161 m, respectively. As shown in Table 4, when the number of path laps is one, the closure errors of ORB-SLAM2 and VINS-MONO [17] are measured to be 1.752 m and 0.972 m, respectively. The closure error of our method is 0.389 m, accounting for 0.726% of the length, which is better than the other two algorithms. When the number of path laps is three, the closure error of our method increases from 0.726% (0.389 m) to 1.615% (2.615 m), while the closure errors of ORB-SLAM2 and VINS-MONO increase from 3.252% and 1.832% to 4.364% and 3.240%. Figure 14 shows the estimated trajectories of the camera. It is clear that our algorithm is the least affected by the increase in trajectory length compared to other algorithms in large-scale and low-texture environments.

Table 4. Comparison of the closure errors of algorithms. Bold numbers represent the best performances.

Number of Laps	Algorithm	Path Length (m)	Closure Error (m)	Error (%)
1	ORB-SLAM2	53.875	1.752	3.252
	VINS-MONO	53.045	0.972	1.832
	Ours	53.592	**0.389**	**0.726**
3	ORB-SLAM2	160.004	6.983	4.364
	VINS-MONO	162.688	5.271	3.240
	Ours	161.919	**2.615**	**1.615**

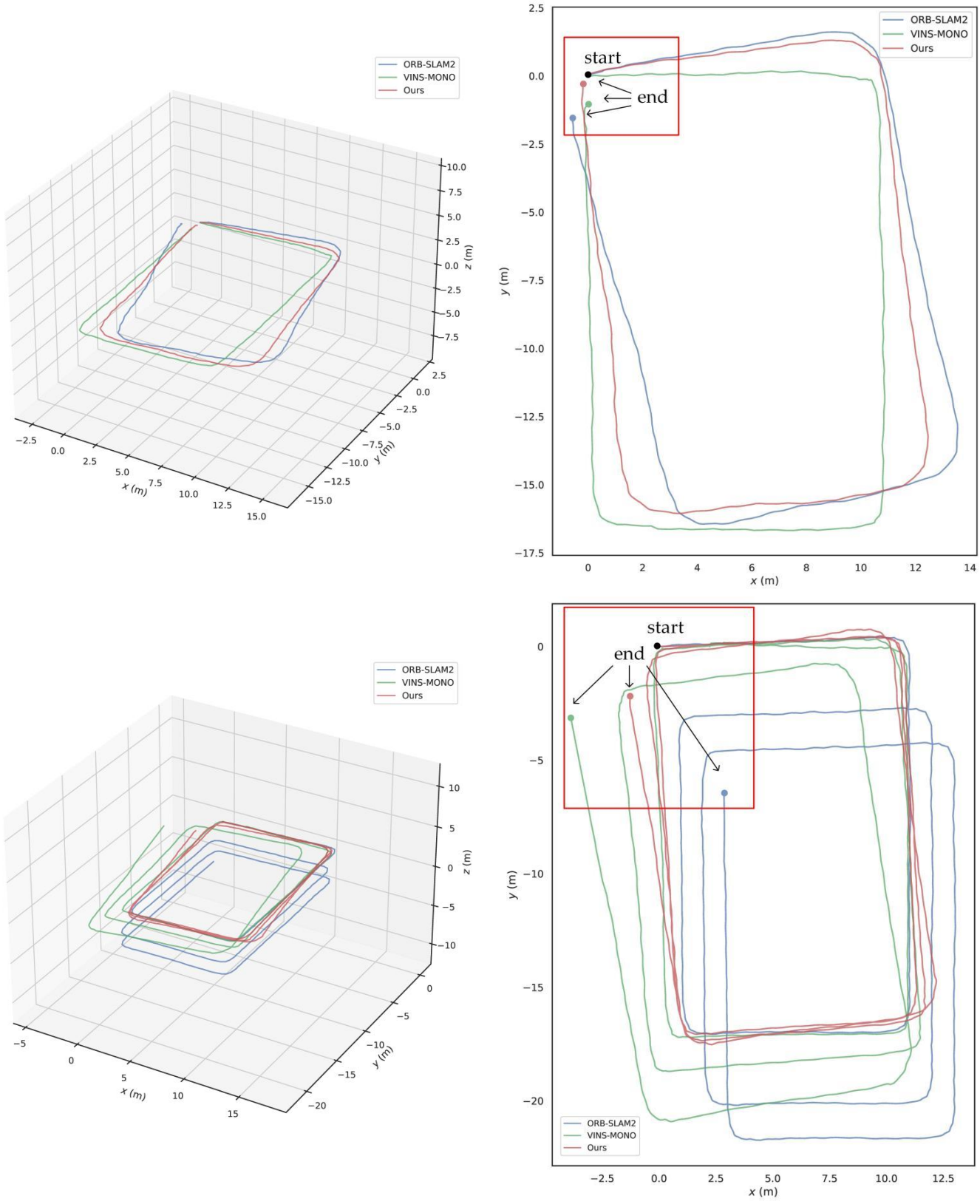


Figure 14. Results of the circular trajectory experiment: (**a**) trajectories of this method (red) and each comparison group; (**b**) comparison of the starting and ending points of the three trajectories—the distance between this method and the starting point is the smallest.

5. Discussion

A monocular visual localization method combining object matching information between a real-time map, a prior semantic map, and line features is presented in this paper, which guarantees continuous and stable localization ability in large-scale environments. Traditional visual SLAM localization methods have difficulty in solving the problem of localization accumulation error in large-scale indoor environments. Although SLAM systems using both point and line features are more powerful to represent the environment, they still suffer from large accumulated errors when working for long periods of time. Aiming to solve this problem, we designed a matching algorithm that combines visual information with existing semantic map model data, and we propose a bundle adjustment model integrating point features and semantic object matching information. The method can effectively reduce the cumulative error without using loop closure detection and improve the localization capability and robustness in large-scale indoor environments.

In the experiments, the ATE of the unclosed trajectory and the closure error of the circular trajectory are used to evaluate and validate the overall performance of the method, and our algorithm is compared with several classical SLAM algorithms. The experimental results for Section 4.3 of the unclosed trajectory are listed in Table 3. Our method presents better localization ability under the semantic object matching constraint and shows sufficient robustness in corners where the scene changes quickly. In Section 4.3, we tested the system performance of closure error, which is a method frequently used to test the performance of inertial navigation systems. As shown by Table 4, our method has the smallest error in both sets of experiments. Moreover, the closure error of our method increases the least after the trajectory length increases. With the two different experiments, it can be confirmed that our method possesses higher localization accuracy.

In this paper, although only the 'gate' semantic objects are used to show the feasibility of our method, theoretically, it is a versatile approach that can integrate more semantic objects into the framework. It should be pointed out that different objects show different degrees of uncertainty; for example, some semantic objects could be occluded or moved in a chaotic or stochastic environment. Thus, a sophisticated SLAM model using matching information of more object types, which we plan to execute in the near future, should carefully consider such uncertainty information; otherwise, worse results could be derived.

Limited by the COVID-19 pandemic in China, and due to the lack of ground-truth data of the test environment, we used laser data as the 'ground-truth' environment data for testing our method. In the future, we wish to test our algorithm in a more complex environment with more kinds of spatial features and with moving objects, such as an airport or a large shopping mall. With this approach, it is possible to provide a continuously precise and robust localization ability in large-scale indoor navigation scenarios even just with a camera and a building information model (BIM) through real-time object matching.

6. Conclusions

In this paper, we propose a monocular visual SLAM approach for large-scale indoor environments by matching a prior map. In this approach, before real-time localization, a prior map consisting of lines of certain semantic objects is built. When running the real-time localization, the lines of a certain type of semantic object are extracted using the M-LSD method. A cost function is proposed to describe the difference between the lines detected in real time and the matched lines of the semantic objects in the prior map. A bundle adjustment model considering the aforementioned cost function is given to optimize the camera pose in real time. We designed a method to evaluate the performance of our approach and a method to build the prior semantic map by using an RGB-D camera. The test results show that our approach can effectively remove the accumulated error in the large-scale indoor visual localization process and provide precise and robust localization ability even without loop closure.

Our future research will be focused on integrating more semantic objects into the theoretical framework, fully considering their different degrees of uncertainty with a CAD, BIM, or another style of semantic map.

Author Contributions: Conceptualization, X.Z., H.W. and Y.Y.; methodology, X.Z. and T.L.; software, X.Z. and T.L.; validation, T.L. and Y.L.; writing—original draft preparation, X.Z. and T.L.; writing—review and editing, X.Z., T.L., Y.Y., H.W. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Technologies R&D Program of China, grant number 2020YFD1100201 and the National Natural Science Foundation of China, grant Number 62073078.

Acknowledgments: We appreciate Deyu Shen, Wenjun Du, and Wenqi Jiang for their suggestions and assistance during the research and the writing process of the paper.

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

References

1. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–108. [CrossRef]
2. Jia, Y.-B. Plücker coordinates for lines in the space. In *Problem Solver Techniques for Applied Computer Science, Com-S-477/577 Course Handout*; Iowa State University: Ames, IA, USA, 2020. Available online: http://web.cs.iastate.edu/~{}cs577/handouts/plucker-coordinates.pdf (accessed on 1 January 2022).
3. Yang, Y.; Geneva, P.; Eckenhoff, K.; Huang, G. Visual-inertial navigation with point and line features. In Proceedings of the 2019 IEEE International Workshop on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019.
4. Gomez-Ojeda, R.; Briales, J.; Gonzalez-Jimenez, J. PL-SVO: Semi-direct monocular visual odometry by combining points and line segments. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4211–4216.
5. Pumarola, A.; Vakhitov, A.; Agudo, A.; Sanfeliu, A.; Moreno-Noguer, F. PL-SLAM: Real-time monocular visual SLAM with points and lines. In Proceedings of the 2017 IEEE international conference on robotics and automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4503–4508.
6. Gomez-Ojeda, R.; Moreno, F.-A.; Zuniga-Noël, D.; Scaramuzza, D.; Gonzalez-Jimenez, J. PL-SLAM: A stereo SLAM system through the combination of points and line segments. *IEEE Trans. Robot.* **2019**, *35*, 734–746. [CrossRef]
7. He, Y.; Zhao, J.; Guo, Y.; He, W.; Yuan, K. Pl-vio: Tightly-coupled monocular visual–inertial odometry using point and line features. *Sensors* **2018**, *18*, 1159. [CrossRef]
8. Hoshi, M.; Hara, Y.; Nakamura, S. Graph-based SLAM using architectural floor plans without loop closure. *Adv. Robot.* **2022**, *36*, 715–723. [CrossRef]
9. Bellavia, F.; Fanfani, M.; Pazzaglia, F.; Colombo, C. *Robust Selective Stereo SLAM without Loop Closure and Bundle Adjustment*; Springer: Berlin/Heidelberg, Germany, 2013.
10. Zhang, X.; Wang, Q.; Wan, D. Map matching in road crossings of urban canyons based on road traverses and linear heading-change model. *IEEE Trans. Instrum. Meas.* **2007**, *56*, 2795–2803. [CrossRef]
11. Hashemi, M.; Karimi, H.A. A critical review of real-time map-matching algorithms: Current issues and future directions. *Comput. Environ. Urban Syst.* **2014**, *48*, 153–165. [CrossRef]
12. Mourikis, A.I.; Roumeliotis, S.I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3565–3572.
13. Sun, K.; Mohta, K.; Pfrommer, B.; Watterson, M.; Liu, S.; Mulgaonkar, Y.; Taylor, C.J.; Kumar, V. Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robot. Autom. Lett.* **2018**, *3*, 965–972. [CrossRef]
14. Zhang, Z.; Liu, S.; Tsai, G.; Hu, H.; Chu, C.-C.; Zheng, F. Pirvs: An advanced visual-inertial slam system with flexible sensor fusion and hardware co-design. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3826–3832.
15. Concha, A.; Loianno, G.; Kumar, V.; Civera, J. Visual-inertial direct SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1331–1338.
16. Tateno, K.; Tombari, F.; Laina, I.; Navab, N. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6243–6252.
17. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
18. 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]

19. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
20. Rong, H.; Gao, Y.; Guan, L.; Ramirez-Serrano, A.; Xu, X.; Zhu, Y. Point-Line Visual Stereo SLAM Using EDlines and PL-BoW. *Remote Sens.* **2021**, *13*, 3591. [CrossRef]
21. Lee, J.; Park, S.-Y. PLF-VINS: Real-time monocular visual-inertial SLAM with point-line fusion and parallel-line fusion. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7033–7040. [CrossRef]
22. Zou, D.; Wu, Y.; Pei, L.; Ling, H.; Yu, W. StructVIO: Visual-inertial odometry with structural regularity of man-made environments. *IEEE Trans. Robot.* **2019**, *35*, 999–1013. [CrossRef]
23. Xu, B.; Wang, P.; He, Y.; Chen, Y.; Chen, Y.; Zhou, M. Leveraging structural information to improve point line visual-inertial odometry. *IEEE Robot. Autom. Lett.* **2022**, *7*, 3483–3490. [CrossRef]
24. Maity, S.; Saha, A.; Bhowmick, B. Edge slam: Edge points based monocular visual slam. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 2408–2417.
25. Li, X.; Ling, H. Hybrid camera pose estimation with online partitioning for slam. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1453–1460. [CrossRef]
26. Chen, S.; Zhou, B.; Jiang, C.; Xue, W.; Li, Q. A LiDAR/Visual SLAM Backend with Loop Closure Detection and Graph Optimization. *Remote Sens.* **2021**, *13*, 2720. [CrossRef]
27. Chen, H.; Hu, W.; Yang, K.; Bai, J.; Wang, K. Panoramic annular SLAM with loop closure and global optimization. *Appl. Opt.* **2021**, *60*, 6264–6274. [CrossRef]
28. Motlagh, H.D.K.; Lotfi, F.; Taghirad, H.D.; Germi, S.B. Position Estimation for Drones based on Visual SLAM and IMU in GPS-denied Environment. In Proceedings of the IEEE 2019 7th International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 20–21 November 2019; pp. 120–124.
29. Hashemifar, Z.S.; Adhivarahan, C.; Balakrishnan, A.; Dantu, K. Augmenting visual SLAM with Wi-Fi sensing for indoor applications. *Auton. Robot.* **2019**, *43*, 2245–2260. [CrossRef]
30. Pascoe, G.; Maddern, W.; Stewart, A.D.; Newman, P. FARLAP: Fast robust localisation using appearance priors. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015.
31. Neubert, P.; Schubert, S.; Protzel, P. Sampling-based methods for visual navigation in 3D maps by synthesizing depth images. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2492–2498.
32. 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.
33. 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.
34. Kim, Y.; Jeong, J.; Kim, A. Stereo camera localization in 3d lidar maps. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–9.
35. Gawel, A.; Cieslewski, T.; Dubé, R.; Bosse, M.; Siegwart, R.; Nieto, J. Structure-based vision-laser matching. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 182–188.
36. Zuo, X.; Geneva, P.; Yang, Y.; Ye, W.; Liu, Y.; Huang, G. Visual-inertial localization with prior lidar map constraints. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3394–3401. [CrossRef]
37. Gu, G.; Ko, B.; Go, S.; Lee, S.-H.; Lee, J.; Shin, M. Towards real-time and light-weight line segment detection. *arXiv* **2021**, arXiv:2106.00186. [CrossRef]
38. Von Gioi, R.G.; Jakubowicz, J.; Morel, J.-M.; Randall, G. LSD: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *32*, 722–732. [CrossRef]
39. Akinlar, C.; Topal, C. EDLines: A real-time line segment detector with a false detection control. *Pattern Recognit. Lett.* **2011**, *32*, 1633–1642. [CrossRef]
40. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
41. Agarwal, S.; Mierle, K. Ceres Solver. 2012. Available online: http://ceres-solver.org (accessed on 23 September 2021).
42. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, DC, USA, 16–18 June 2020; pp. 11108–11117.
43. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. 2017. Available online: http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space (accessed on 7 June 2018).
44. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*; Neural Information Processing Systems Foundation, Inc.: La Jolla, CA, USA, 2018; Volume 31.
45. Thomas, H.; Qi, C.R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6411–6420.

46. Shan, T.; Englot, B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765.
47. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In Proceedings of the 13th European Conference of Computer Vision, Zürich, Switzerland, 6–12 September 2014; pp. 834–849.