RGB-D SLAM FOR INDOOR MOBILE PLATFORM BASED ON HYBRID FEATURE FUSION AND WHEEL ODOMETER INTEGRATION

ZOU YAJING

PhD

The Hong Kong Polytechnic University

2022

The Hong Kong Polytechnic University

Department of Land Surveying and Geo-Informatics

RGB-D SLAM for Indoor Mobile Platform Based on Hybrid Feature Fusion and Wheel Odometer Integration

Zou Yajing

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
February 2022

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: ____________________

Name of Student: Zou Yajing

ABSTRACT

Self-tracking and scene reconstruction are crucial for mobile platform navigation in unknown indoor environments. Red-Green-Blue Depth (RGB-D) camera is an ideal choice of the onboard sensor on the mobile platform, in consideration of its small size, light weight, and cheap price. However, the performance of RGB-D simultaneous localization and mapping (SLAM) is degenerated due to two main problems: (a) the SLAM system is prone to lost tracking under low textured scenes, and (b) the accuracy is inadequate for mobile platform navigation because of the accumulating drift. This thesis aims to solve the first problem by hybrid feature fusion and the second one by wheel odometer integration, and therefore improves the continuity and accuracy of the RGB-D SLAM system on the mobile platform.

Firstly, a new RGB-D SLAM fusing point and line features is proposed. While previous line-based methods utilize either 3D-3D or 3D-2D line correspondences, the new method combines both and can exploit more line information. It is evaluated on Technical University of Munich (TUM) RGB-D datasets and real-world experiments. Experiment results show that the proposed method can yield better continuity than state-of-the-art (SOTA) methods. In addition, it can improve the localization accuracy of the method utilizing 3D line features by 22.5% and the mapping accuracy by 10.2%. The improvements over the method utilizing 2D line features are 25.8% and 14.7% in consideration of localization and mapping accuracies, respectively.

Secondly, a new RGB-D SLAM fusing point and plane features is proposed. While previous plane-based methods assign experimental weights to the plane features, the new method derives the analytical covariances by plane fitting and covariance propagation. Point and plane features are optimally combined to construct the cost function based on the derived covariances. Furthermore, a new representation form for plane features is developed based on the parallel and vertical relationships among planes. It encodes the structural regularity in indoor scenes and is further utilized by factor graph optimization. Experiments on the TUM RGB-D datasets prove that the proposed method yields better continuity than the feature point-based methods. In the lab room experiment, the proposed method can improve the localization accuracy by 23.6% using the analytical covariances, and enhance that by 27.6% using the new representation form. In the corridor experiment, the improvements of the mapping accuracies are 11.5% owing to the analytical covariances, and 8.8% using the new representation form.

Thirdly, a new localization and mapping method by tightly coupling the RGB-D camera and the wheel odometer is proposed. Previous methods assume the platform moves on a 100% flat floor which is unpractical and may lead to non-optimal estimation results. To avoid the disadvantage, the new method adopts a soft assumption that the platform moves with small perturbations due to uneven terrain and develops a two-step strategy to handle the perturbations: (a) firstly, the Mahalanobis distance test is applied to examine the motion assumption, and (b) secondly, the ground plane is detected to constrain the mobile platform. Moreover, the visual and wheel odometer constraints are tightly coupled in a new factor graph. The proposed method is evaluated by two real-world experiments in a lab room and a corridor, respectively. Compared with the previous loose-coupled method utilizing a hard planar motion assumption, it can improve the localization accuracy by 40.7% and the mapping accuracy by 33.8%.

Finally, based on the algorithms developed in this study, a comprehensive real-time RGB-D SLAM system is developed for mobile platform navigation. Point, line, and plane features are simultaneously fused in the comprehensive system. Hybrid features are combined with the wheel odometer under the soft planar motion assumption. In real-world experiments, compared with the feature point-based system, the proposed system can improve the localization accuracy by 70.1% and the mapping accuracy by 75.9%, combing the wheel odometer can improve these accuracies by 66.3% and 72.1%, fusing points, lines, and planes can improve them by 57.2% and 62.6%, fusing plane features can improve them by 53.8% and 55.6%, and the smallest improvements are 33.6% and 39.1% by fusing line features.

PUBLICATIONS ARISING FROM THE THESIS

During four years of work, three papers were published, and two papers are currently under revision or preparation. Publications associated with this research are listed as follow:

1. Zou, Y., Eldemiry, A., Li, Y., & Chen, W. (2020). Robust RGB-D SLAM using point and line features for low textured scene. *Sensors*, 20(17), 4984.

2. Chen, S., Wen, C. Y., Zou, Y., & Chen, W. (2020). Stereo visual inertial pose estimation based on feedforward-feedback loops. *arXiv* preprint arXiv:2007.02250.

3. Eldemiry A, Zou Y, Li Y, Wen C-Y, Chen W. Autonomous Exploration of Unknown Indoor Environments for High-Quality Mapping Using Feature-Based RGB-D SLAM. Sensors. 2022; 22(14):5117.

4. ZOU, Y., Eldemiry, A., Chen, S., Wen, C., Zhang, X., Li, Y.,& Chen, W. (2022). Robust RGB-D SLAM using point and plane features for low textured indoor scene. Under revision.

5. ZOU, Y., & Chen, W. (2022). Tightly coupling RGB-D camera and wheel odometer for ground vehicle navigation. Under preparation.

ACKNOWLEDGMENTS

First, I would like to express my gratitude to my supervisor, Professor Wu Chen. Thank him for giving me the opportunity to start the journey in PolyU, embrace different cultures and step into the area of SLAM. Without his guidance and help, I could not fulfil the difficult tasks during my PhD period. His inspiring ideas and excellent leadership have taught me a lot and will also help me in the future.

I would like to extend my thanks to the colleagues in the navigation lab. Especially, I would like to thank the members of our 3D modelling group, Amr El-Demiry, Yulin Hu, Yaxin Li, Shengjun Tang, and Walid Darwish. I learned a lot from the daily discussion and cooperation with them.

Acknowledgments must go to Dr. Xiang Gao, who is the writer of SLAM 14 lectures and guides me to the road of SLAM. The members in the Paopao Robot Group must receive my sincere thanks for delivering and introducing influential and inspiring works in the SLAM area. Thanks also go to my friend and colleague, Dr. Shengyang Chen. During the implementation of FLVIS, an open-source visual-inertial system derived from our cooperation, I learned a lot from his creative idea and earnest attitude.

I would also express my special thanks to my committee members. Thank Professor Man-sing Wong for reviewing my thesis and arranging my viva. Thank Professor Ruizhi Chen and Professor Xialing Yuan for their careful, helpful, and inspiring comments, which improve the quality of the thesis.

Finally, love goes to my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

The applications of autonomous mobile platforms have grown rapidly in recent decades, and various types of robots have been developed, such as Unmanned Aerial Vehicles (UAV), smart wheelchairs, and ground vehicles. For example, UAVs are employed for agriculture, traffic monitoring, and mine inspection, smart wheelchairs are designed for the safety, and mobility of disabled people, and ground vehicles are commonly applied in security, logistics, and autonomous driving (Siegwart et al., 2011). In general, the operation scenes of these mobile platforms in the above applications can be categorized as: (a) the outdoor open-sky scene with reliable Global Navigation Satellite System (GNSS) signal, and (b) the indoor scene with GNSS denied environments.

While the location can be provided by GNSS in outdoor scenes, alternative techniques are required for positioning in indoor scenes. Lidars and cameras are widely used on the mobile platform for self-localization and scene-perception in GNSS-denied environments. While Lidars outperform cameras in terms of the detection distance and ranging accuracy, cameras are also attractive in consideration of the low cost, high portability, and ability to provide additional colour information (Aulinas et al., 2008). The technique of tracking the mobile platform and reconstructing the surrounding scene based on cameras is called Visual SLAM (Fuentes-Pacheco et al., 2015). It can be further divided into three groups based on the utilized cameras: (a) monocular SLAM using a single camera; (b) stereo SLAM using stereo cameras; and (c) RGB-D SLAM using the RGB-D camera (Aulinas et al., 2008).

In these SLAM systems, depth images are utilized for dense scene reconstruction, which is essential for path planning of the mobile platform (Ling & Shen, 2019). In general, depth estimation is achieved by three traditional techniques:

(a) Stereo matching. It matches the pixels on two rectified images and then computes the disparity between these pixels for depth estimation (Bleyer et al., 2011; Chang & Chen, 2018).

(b) Structured Light (SL). It projects a known pattern onto the surface by a projector, captures a distorted image by an infra camera, and recovers the depth by image analysis (Bleyer et al., 2011; Chang & Chen, 2018).

(c) ToF (Time of Flight). It emits light onto the surface, measures the travel time, and then computes the depth using the incident angle and the travel distance of the light (Fuchs & Hirzinger, 2008).

Stereo matching is utilized for depth estimation using a single camera or stereo cameras. It requires a powerful Graphics Processing Unit (GPU) for real-time processing due to the high computation cost by pixel matching. On the other hand, SL or ToF technique is employed by the RGB-D camera, which consists of RGB and depth cameras within the small size, light weight, and cheap price (Zhang, 2012). Compared with a single camera or stereo cameras, an RGB-D camera can generate pixel-wise colour and depth images in real time on a commercial Central Processing Unit (CPU). Furthermore, the quality of the depth images from the RGB-D camera is much higher than those from stereo matching (Zhang, 2012).

Therefore, utilizing the RGB-D camera is a potential solution for rea-time indoor mobile platform navigation, in consideration of the size, weight, price, and computation cost. However, two major problems prevent it from general usage:

(a) Tracking failure in low textured scenes (Fu et al., 2019). Low textures are commonly generated by the white wall, occlusion, and illumination. Compared with manual operation, the mobile platform lacks the ability to understand the environment, and cannot avoid low textures by selecting a good camera view. Traditional methods rely on point features to estimate the position and orientation of the mobile platform, but these features are hard to detect and match under low textured scenes. Therefore, the mobile platform may fail to track itself continuously.

(b) Accumulating drift during mobile platform operation (Filipenko & Afanasyev, 2018). The tracking result of the mobile platform has a drift due to the following factors: (a) image noise; (b) incorrect feature detection and matching; (c) low-quality initial guess of the mobile platform pose. This drift is always accumulating, and the accuracy of the RGB-D SLAM system is reduced after long-term operation. This is unfavourable for the real-time operation of the mobile platform, which demands high-precision positions and orientations.

The system continuity in low textured scenes can be enhanced by fusing other features (i.e., lines, planes, and objects) (Bowman et al., 2017; Gomez-Ojeda et al., 2019; Taguchi et al., 2013). These high-level features still exist in low textured scenes. They can provide additional feature matches for pose estimation when point features are unavailable. An alternative method to improve the system continuity is to combine with photometric information, which is less affected by low textures (Engel et al., 2017; Engel et al., 2014). The system accuracy is commonly improved by backend optimization techniques, such as loop closing and bundle adjustment (Mur-Artal et al., 2015). The former reduces the tracking drift when the mobile platform re-visits an old place, while the latter uses joint optimization to correct the localization error in a local sliding window. Sensor fusion is also a feasible way for drift compensation (Qin et al., 2018; Wu et al., 2017). Inertial measurement unit (IMU) and wheel odometer are commonly equipped on the mobile platform. They can measure the relative motion of the platform and provide a good initial guess for RGB-D SLAM. Furthermore, the integrated measurements of IMU and wheel odometer are accurate in a short period, which can be coupled with visual measurements to further improve the system accuracy.

This thesis aims to (a) solve the first problem and then enhance the continuity of the RGB-D SLAM system for the mobile platform by exploiting the features from the RGB-D camera; and (b) solve the second problem and then improve the system accuracy by utilizing the motion measurements from the wheel odometer.

1.2 Objectives and contributions

To achieve the above aim, two main objectives are addressed in this thesis.

(a) Fusing hybrid features to avoid the tracking failure of the mobile platform.

While point features are plentiful in rich textured scenes, they are hard to detect and match in low textured scenes. On the other hand, high-level features(i.e., lines and planes) still exist and are predominant in such challenging scenes. They are more reliable than point features because: (a) they are less affected by image noise; and (b) depth outliers can be detected and removed during line or plane fitting. The onboard computer on the mobile platform can afford the computation cost of handling line and plane features by utilizing a multi-thread design. Therefore, fusing hybrid features (i.e., points, lines, and planes) is feasible to avoid tracking failure in low textured scenes.

(b) Combining the wheel odometer and the RGB-D camera to reduce the tracking drift.

Though backend techniques (i.e., loop closing and bundle adjustment) have been applied to reduce the accumulating drift of the RGB-D SLAM system (Mur-Artal et al., 2015), the system accuracy can be further enhanced by combining an additional sensor installed on the mobile platform (i.e., a wheel odometer). The wheel odometer can measure the relative motion of the mobile platform and predict an initial value for the RGB-D SLAM system. It helps the results of pose estimation avoid falling into local optima, which is a non-convex problem (Wang et al., 2017). In addition, the integrated measurements from the wheel odometer are accurate in a short period. They can provide additional constraints for mobile platform tracking and reduce the drift of the system using the graph optimization technique.

Pursuing the above objectives, a lot of studies have been carried out. The main contributions of the thesis research are summarized as follows:

(a) The first contribution is related to the fusion of the point and line features.

The existing methods exploit either 3D–3D or 3D–2D line correspondences for pose estimation (Cheng & Wang, 2018; Fu et al., 2019; Lu & Song, 2015; Y. Zhou et al., 2018). This thesis proposes a new method to combine both correspondences and utilizes more line information than the previous methods. If the depth measurements on the detected 2D lines are valid and can be used to fit a 3D line with small fitting errors, 3D-3D line correspondences are employed in the new method, which contains more depth information than 3D-2D line correspondences. In contrast, if the depth measurements are invalid or the line fitting errors are too large, 3D-2D line correspondences are utilized, which are neglected by those methods using only 3D–3D correspondences. In the new method, the camera pose is estimated by minimizing a new cost function, which consists of the point reprojection errors, and the 3D and 2D line reprojection errors. Compared with the previous methods, the new cost function utilizes more constraints from line features and can therefore yield higher localization accuracy.

The performance of the new method is evaluated on both TUM RGB-D datasets and real-world experiments and compared with SOTA methods. The results on public datasets show that it can yield higher continuity in low textured scenes than other point-based or line-based methods. In addition, it can improve the localization accuracy of the method using 3D-3D line correspondences by 22.5% and enhance its mapping accuracy by 10.2%. The improvements over the method utilizing 3D-2D line correspondences are 25.8% and 14.7% in consideration of the localization and mapping accuracies, respectively.

(b) The second contribution is related to the fusion of point and plane features.

While most of the existing plane-based methods just assign experimental weights to the plane features (Guo et al., 2019; Hsiao et al., 2017; Kaess, 2015; Taguchi et al., 2013), this thesis derives the variance-covariance matrix of the plane measurements in the spherical form based on plane fitting and covariance propagation. A new cost function is constructed for pose estimation, which fuses the point reprojection error and the plane transformation error based on the derived variance-covariance matrix instead of experimental weights.

Furthermore, the parallel and vertical relationships among the indoor structures are not exploited properly by the previous methods. Most of them assume that all the surfaces in the operation scenes are parallel to three main orthogonal directions, which is called Manhattan World (MW) assumption (Kim, Coltin, et al., 2018a, 2018b; Yanyan Li, Nikolas Brasch, et al., 2020a; Yanyan Li, Raza Yunus, et al., 2020; Zhou et al., 2016). Though the assumption is beneficial for exploiting structural regularity, it may lead to wrong tracking results when the assumption is not strictly satisfied. Less strict assumptions of the operation scene are utilized by some studies, such as the Atlanta World (Joo et al., 2019), the mixture of MW frames (Straub et al., 2014), and the Stata Center World (Kaneko & Ichinose, 2019). Different from the previous work, this thesis develops a new representation form for plane features, which can avoid the backward of the MW assumption and also preserve the structural regularity. It represents a new plane using the parallel and vertical relationships among the planes and MW axes. A new factor graph utilizing the new representation form is built for pose optimization, which can further enhance the localization and mapping accuracy of the RGB-D SLAM system.

A new method for fusing point and plane features is proposed based on the analytical covariances and the new representation form for plane features. It yields higher continuity than feature point-based methods in low textured scenes, and outperforms other SOTA methods in rich textured scenes. In the lab room experiment, the localization accuracy is improved by 23.6% using the analytical covariances, and enhanced by 27.6% using the new representation form. In the corridor experiment, the proposed method can improve the mapping accuracy by 11.5% owing to the analytical covariances, and by 8.8% using the new representation form.

(c) The third part is related to the combination of the RGB-D camera and the wheel odometer.

Few studies are conducted in this area (Labbé & Michaud, 2019; Ligocki & Jelínek, 2019; D. Yang et al., 2019). Some of them use a loose-coupled design and do not exploit the potential of the integrated measurements from the wheel odometer. In addition, they assume the mobile platform moves on a flat ground plane without perturbation, which is not practical and may lead to non-optimal estimation results. This thesis proposes a tight-coupled method combining the RGB-D camera and the wheel odometer. It develops a two-step strategy to handle the perturbation of the mobile platform on the floor: (a) firstly, the Mahalanobis distance test is employed to detect the perturbation; (b) secondly, the ground plane is detected, and its coefficients are used to constrain the mobile platform. A new factor graph is constructed in the proposed method, which consists of the visual and wheel odometer constraints and the constraints from the planar motion assumption. The associated errors and covariances of these constraints are derived and fused in the new factor graph. The potential of the integrated measurements of the wheel odometer is further exploited by joint optimization.

The proposed method is evaluated using the experiments in a lab room and a corridor. It can outperform other SOTA methods in consideration of both localization and mapping. Compared with the loose-coupled method using a hard planar motion assumption, the proposed method can improve the localization and mapping accuracy by 40.7% and 33.8%, respectively.

1.3 Structure of the dissertation

The main contents of the remaining chapters are summarized as follows.

Chapter 2 reviews the literature about mobile platforms, onboard sensors, scene representations, and SLAM algorithms. The mobile platforms are utilized to fulfill the navigation task, the sensors are equipped on the mobile platforms for localization and mapping, different types of maps are applied to represent the operation scenes for direct understanding, and finally, the SLAM algorithms are the core methodologies to localize the mobile platform and reconstruct the operation scene using the sensors equipped on the platform.

Chapter 3 proposes a new RGB-D SLAM system fusing point and line features for low textured scenes. It firstly investigates the representation types of the point and line features for projection and optimization, and then introduces the extraction and matching pipeline of these features. Both the 3D and 2D line reprojection errors are exploited with the point reprojection errors, and a new cost function is built based on these errors for pose optimization. Finally, extensive experiments are conducted to prove the superiority of the proposed system by comparing it with other SOTA methods.

Chapter 4 proposed s new RGB-D SLAM system fusing point and plane features. It first investigates the representation types for plane features and introduces a new representation form using the vertical and parallel relationships among planes and MW axes. Then it introduces the pipeline of extracting and matching plane features.

A new cost function is built by fusing the point reprojection errors and the plane transformation errors based on their analytical covariances. A new factor graph is constructed by utilizing the new representation type for plane features. Finally, extensive experiments are carried out to evaluate the performance of the proposed system and compare it with other SOTA methods.

Chapter 5 introduces a tight-coupled localization and mapping system for the mobile platform navigation by fusing the RGB-D camera and the wheel odometer under the planar motion assumption. It first investigates the measurement model for wheel odometer integration. Then it derives the errors and covariances of the wheel odometer constraints and the planar motion constraints. After that, a new factor graph is built by fusing the visual, wheel odometer, and planar motion constraints. Experiments are conducted in a lab room and in a corridor, respectively, to prove that the proposed system generates higher localization and mapping accuracy than other SOTA methods.

Chapter 6 proposes a comprehensive real-time RGB-D SLAM system by hybrid feature fusion and wheel odometer integration. The proposed system is based on the core techniques proposed in the last three chapters: (a) fusing point and line features; (b) fusing point and plane features; (c) fusing the RGB-D camera and the wheel odometer under the planar motion assumption. Experiments in the lab room and the corridor are carried out and show that the comprehensive system can outperform the systems presented in the last three chapters owing to both hybrid features fusion and wheel odometer integration.

Chapter 7 draws the conclusions based on the methodologies and experiments results from Chapter 3 to Chapter 6, and then offers recommendations for future research plans.

CHAPTER 2

RELATED WORKS

This chapter aims to deliver a detailed and selective review of previous research works, with the purpose of offering an insight into the background of the research studies endeavoured in this thesis. It focuses on reviewing the localization and mapping algorithms but also attempts to give a basic introduction about the mobile platforms, the onboard sensors, and the scene representations.

Mobile platforms are the main body to fulfill navigation or other high-level tasks (Rubio et al., 2019). They can be divided into the following types based on the locomotion system: (a) stationary; (b) water-based; (c) air-based; and (d) land-based. The most common mobile platform is land-based, which can be further classified into three sub-types: (a) wheeled (Chan et al., 2013); (b) legged (Wieber et al., 2016); and (c) tracked (Vu et al., 2008). This thesis mainly focuses on the wheeled mobile platform, which applies wheels for platform mobility. Compared with the legged and tracked mobile platforms using legs and threads respectively, wheels are cheap and easy to control on the flat ground.

2.1 Onboard sensors

Mobile platforms are equipped with sensors to locate themselves and percept the operation scenes, which is necessary for subsequent tasks, such as navigation, exploration, and grasping. In general, onboard sensors can be divided into two sub-types: (a) the internal sensors, which measure the self-motion of the mobile platform, such as the angular rate, accelerometers, and wheel speed; (b) the external sensors, which measures the operation scenes, such as the light intensity and the distance from the sensors to the scene surfaces (Coiffet & Chirouze, 2012). The internal sensors and external sensors can be integrated to avoid the disadvantages of individual sensors.

Several popular sensors for mobile platforms are listed in Table 2.1 (Coiffet & Chirouze, 2012). In general, internal sensors, such as the IMU and the wheel odometer, are equipped on the mobile platform for motion control. They are applied on both indoor and outdoor scenes and can provide high-frequency output. An IMU is commonly equipped on a UAV for motion control as its roll and pitch are observable by aligning to the gravity direction. The disadvantage is the accumulating drift error of dead reckoning. In addition, the poses from the internal sensors are relative to the local coordinate system when the sensors start to work. On the other hand, GNSS can provide absolute positions for the mobile platform without the accumulating drift. But the results are only reliable in outdoor and open-sky scenes. Lidar and camera can be utilized in both indoor and outdoor environments for pose estimation. Compared with the camera, the Lidar has a long detection range and high-precision point cloud. On the other hand, the camera is also attractive because of its high-density pixels, low cost, and high portability.

Table 2. 1 The advantages and disadvantages of several internal and external sensors.

Onboard Sensors		**Operation Scenes**	**Pros**	**Cons**
internal sensors	IMU	indoor/ outdoor	high-frequency output observable roll and pitch	accumulating drift of dead reckoning
	wheel odometer	indoor/ outdoor	high-frequency output cheap and accessible	accumulating drift of dead reckoning
external sensors	GNSS	outdoor	absolute position no accumulating drift	no signal in indoor scenes
	Lidar	indoor/ outdoor	long detection range high-precision point cloud	expensive big size
	camera	indoor/ outdoor	high-density pixels portability cheap	short detection range

Among various kinds of cameras, the RGB-D camera is the most suitable for indoor mobile platforms because it can provide high-quality depth images in real time on a commercial CPU. In terms of the principles to generate depth images, RGB-D cameras can be divided into two sub-types: (a) SL-based, such as Occipital Structure Sensor, Microsoft Kinect V1, and Asus Xtion PRO Live; (b) ToF-based, such as Microsoft Kinect V2, Intel Realsense L515, and Microsoft Azure Kinect (Albert et al., 2020; Diaz et al., 2015; Lourenço & Araujo, 2021). Their parameters are compared in Table 2.2. Some cameras may have multiple resolutions, frequencies, and fields of views (FOVs), and only the suitable parameters for camera localization are listed below.

Table 2. 2: Parameters of various types of RGB-D cameras.

RGB-D camera	**Principle**	**Depth Resolution**	**Frequency**	**FOV**	**Working Range**
Occipital Structure Sensor	SL	640×480	30/60	H: 58.0° V: 45.0°	0.4- 3.5m
Microsoft Kinect V1	SL	320×240	30	H: 58.5° V: 46.6°	1.2-3.5 m
Asus Xtion PRO Live	SL	640×480	60	H: 58.0° V: 45.0°	0.8-3.5 m
Microsoft Kinect V2	ToF	514×424	30	H: 84.1° V: 53.8°	0.5-4.5m
Intel Realsense L515	ToF	640×480	30	H: 70.0° V: 55.0°	0.25-9.0m
Microsoft Azure Kinect	ToF	640×576	30	H: 75.0° V: 65.0°	0.5-3.9m

2.2 Scene representations

The operation scene is reconstructed by the onboard sensors. For purposes of visualization, interaction, localization, mapping, or planning, the reconstructed scene must be understood by the onboard computer (Slabaugh et al., 2001). Five types of maps are utilized corresponding to different purposes: (a) feature point map (Klein & Murray, 2007); (b) point cloud map (Rusu & Cousins, 2011); (c) voxel map (Muglikar et al., 2020); (d) mesh map (Cignoni et al., 2011); and (e) surfel map (Andersen et al., 2010). In general, the voxel map is favoured by the navigation task of the mobile platform, point cloud and mesh are utilized for 3D reconstruction, the feature point map is applied in visual SLAM, and the surfel map is a mid-production for mesh generation. Their advantages and disadvantages are listed in Table 2.3 and discussed as follows.

Table 2. 3: The advantages and disadvantages of scene representations.

Scene representations	Pros	Cons
feature point map	lightweight	too sparse
point cloud map	dense	noisy; lack of relationship
voxel map	unified	implicit
mesh map	explicit; lightweight	unable to represent complicated scenes
surfel map	easy to update	implicit; high computation cost

2.2.1 Feature point map

A feature point map stores the feature points in the operation scene and builds their relationships with selected camera poses. The indexes of the camera poses, and the pixel locations associated with the feature point are also contained in the map. This structure enables fast data association for bundle adjustment and re-localization, and is popularly applied in feature point-based SLAM methods, such as PTAM, ORB-SLAM2, and VINS-mono (Klein & Murray, 2009; Mur-Artal & Tardós, 2017; Qin et al., 2018). However, the feature points in this map are too sparse and thus not useful for obstacle avoidance and path planning.

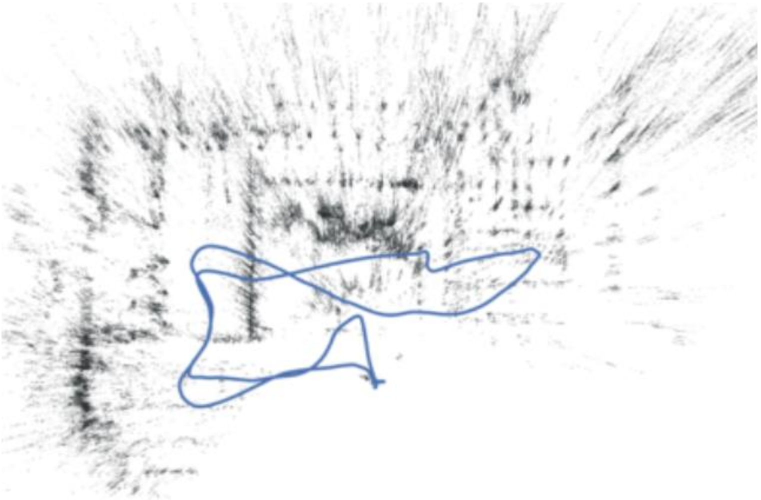


Figure 2. 1: Feature point map of Machine Hall in EuRoc MAV dataset (Burri et al., 2016) constructed by FLVIS (Chen et al., 2020).

2.2.2 Point cloud map

A point cloud map stores a set of points in the operation scene, and these points are unique and have no relationship with each other. The advantage of the point cloud map is its convenience and ease of generation. Point clouds can be easily produced by Lidar, RGB-D camera, and photogrammetry techniques, and then integrated to build a more complete map (Rusu & Cousins, 2011). Voxel grid filter is essential when redundant points are inserted into the map, because it can keep fewer points in the region defined by the voxel resolution (Han et al., 2017). With the purpose of real-time navigation based on a point cloud map, high-precision perception sensors are required to reduce the noise of the map, and the static operation scene is preferred because dynamic objects cannot be updated and removed quickly in the map (Gao et al., 2019; Whitty et al., 2010).

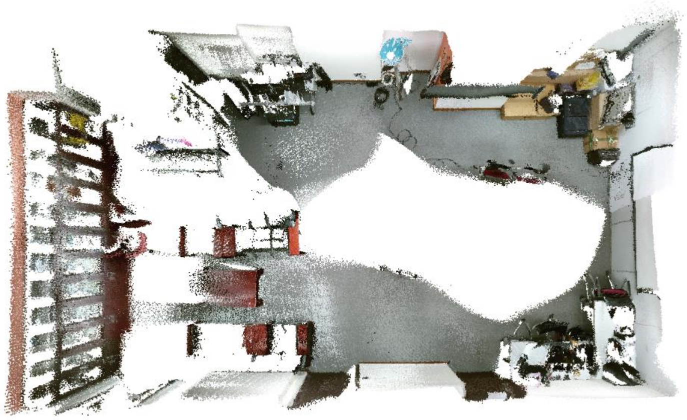


Figure 2. 2: Point cloud map of Navigation Lab in the Hong Kong Polytechnic University (PolyU).

2.2.3 Voxel map

A voxel or volumetric map divides the operation scene into voxels of equal size. These voxels are similar to the pixels in the 2D image plane. Each voxel can store its properties, such as colour, density, and species. Truncated Signed Distance Function (TSDF), Euclidean Signed Distance Function (ESDF), and occupancy maps are three popular voxel-based representation types. TSDF is popularly applied in direct SLAM methods (Dai et al., 2017; Newcombe, Izadi, et al., 2011). The signed distance to the nearest surface is maintained and updated in the voxel, and the surfaces of the objects and scenes can be easily extracted by the Marching Cubes algorithm (Lorensen & Cline, 1987). Occupancy and ESDF maps are gaining popularity in navigation applications. The former one stores the occupancy probability of each voxel, while the latter one stores the Euclidean distance to the nearest occupied voxel. A collision-free path can be simply generated from an occupancy map (Hornung et al., 2013). ESDF further enables fast obstacle avoidance because the distance of every voxel to the nearby obstacle is straightforward (Chen et al., 2021).

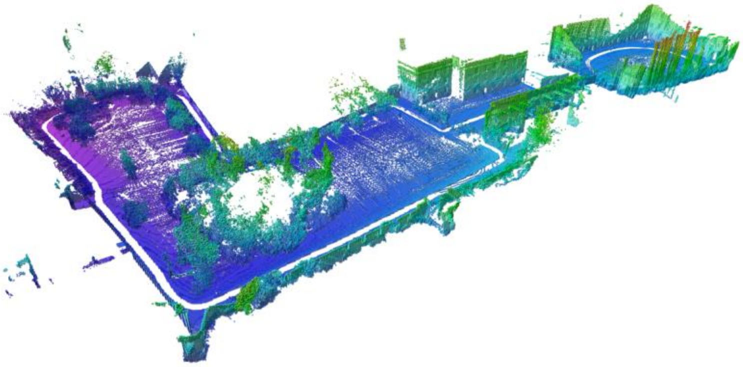


Figure 2. 3: An occupancy map using octomap (Hornung et al., 2013).

2.2.4 Mesh map

Mesh is a collection of the vertices, edges, and faces, where the edges are the connection of the vertices and the faces are a close set of the edges (Smith, 2006). Owing to topological relationships inside the mesh, it becomes a popular intermediary for texture mapping. Mesh generation is also a hot topic in the area of computer graphics and geometric modeling. The main backward is that it is difficult to use mesh to represent scenes with complicated topological relationships, such as overlapping, separation, and containment.

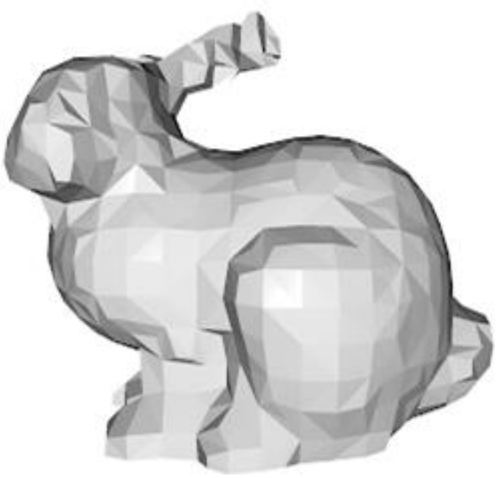


Figure 2. 4: A mesh model of a rabbit reconstructed by open3d (Q.-Y. Zhou et al., 2018).

2.2.5 Surfel map

Surfel is commonly used as a reference model for RGB-D frame alignment (Whelan et al., 2015; Whelan et al., 2016). Each surfel stores its position, normal, colour, weight, radius, initialization time, and last updated time. These properties are updated by weighted averaging when new depth images are inserted (Whelan et al., 2015). The noise of the depth is reduced by the averaging operation which is essential for producing a high-quality 3D mesh. In general, due to the computation cost by updating the surfel and the memory cost by storing the surfel, GPU is required for the real-time processing (Andersen et al., 2010).

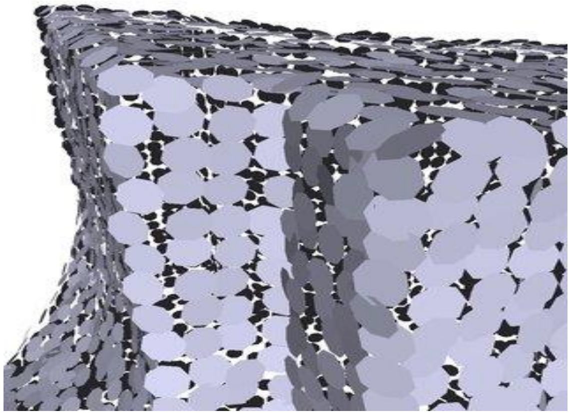


Figure 2. 5: A surfel-based reconstruction model(Andersen et al., 2010).

2.3 SLAM algorithms

Localization and mapping are two basic tasks for the mobile platform's navigation. In outdoor environments, the mobile platform is located by GNSS, and the map is constructed by ranging sensors. In indoor GNSS-denied scenes, the tasks of localization and mapping are simultaneously handled by the SLAM algorithms. This thesis focuses on using the RGB-D camera to track the mobile platform and reconstruct the operation scene. The attention of this section is paid to the SLAM algorithms based on the RGB-D camera. In addition, to show the origin and progress of the RGB-D SLAM algorithms, related research works based on monocular and stereo cameras are also reviewed.

In general, the SLAM algorithms can be divided into two categories based on their principles: (a) direct methods, which utilize all the photometric or geometric information from all the pixels; (b) feature-based methods, which exploits salient features(i.e., points, lines, and planes) (Aulinas et al., 2008). The advantages of the direct methods are twofold: (a) photometric or geometric information is less affected by low textures ; (b) direct methods provide a dense representation for mobile platform planning. The main backward is that high computation and memory resources are required, and these algorithms are usually implemented on a GPU. Feature-based methods are more lightweight and can run on a commercial CPU. Based on the exploited features, these methods can be further divided into: (a) point-based methods (Mur-Artal et al., 2015); (b) line-based methods (Cheng & Wang, 2018); and (c) plane-based methods (Ji et al., 2018). In addition, there are feature-based methods attempting to build the MW frame for pose estimation, which is classified as MW-based methods (Zhou et al., 2016). MW axes are constructed using the parallel and vertical relationships among the point, line, and plane features (Zhou et al., 2016). In general, in the line-based methods, plane-based methods, and MW-based methods, point features are also utilized to ensure tracking accuracy in rich textured scenes.

To sum up, this section aims to review the research works about: (a) direct methods; and (b) feature-based methods, which is further divided into sub-types: (a) point-based methods; (b) line-based methods; (c) plane-based methods; and (d) MW-based methods. Their advantages and disadvantages are discussed as follows.

2.3.1 Direct methods

This section first introduces the basic pipeline of the direct methods, then reviews the literature about monocular and RGB-D direct methods, and finally points out the drawback of these direct methods.

In general, the basic pipeline of the direct methods includes (a) registering the current frame to the previous frame or a global model; (b) computing the camera pose by minimizing the photometric or geometric errors from all the pixels; (c) iteratively executing (a) and (b) until the update of camera pose reaches a threshold; (d) updating the global model using the new camera pose and point cloud (Izadi et al., 2011).

The first complete direct monocular SLAM system is DTAM (Newcombe, Lovegrove, et al., 2011). The camera pose is computed by matching the current image with the synthetic view image from the 3D model. The cost function consists of the photometric errors from all the pixels. GPU is required for generating dense 3D models and registering the frames in DTAM. With the purpose of implementation on CPU, semi-dense direct methods and sparse direct methods are proposed to reduce the computation cost (Engel et al., 2017; Engel et al., 2014). A semi-dense map is used for scene representation in LSD-SLAM (Engel et al., 2014). Sparse points can be sampled from edges and weak intensity variations in DSO (Engel et al., 2017). Geometric camera calibration is integrated with photometric camera calibration to enhance the alignment quality.

Kinect-Fusion is a masterpiece for direct RGB-D methods (Newcombe, Izadi, et al., 2011). The current depth frame is aligned to a global volumetric model, and its pose is estimated by a coarse-to-fine Iterative Closest Point (ICP) algorithm. However, it is limited to small workspaces due to the high memory consumption. To lower the memory cost by map representation and updating, Whelan et al. (2012) present Kintinuous based on a shift volumetric map, and Nießner et al. (2013) maintain a lightweight map by combining the sparse volumetric grid and the voxel hashing.

The loop closure is also a hard task for direct methods, which commonly have the requirement to deliver consistent 3D models on the fly. Kähler et al. (2016) divided the scene into the submaps and associate them with their overlaps. Loop closure is detected by the randomized ferns (Glocker et al., 2013) and the poses of the submaps are refined by the global optimization. Then the individual submaps can be fused in real time. Elastic-Fusion develops a two-step approach to implement the loop closure (Whelan et al., 2015). Firstly, the local model-to-model optimization is applied to correct the local loop closure. Secondly, the randomized fern is applied to detect the global loop closure and the deformation graph is then refined after adding the loop closure constraint. Kerl et al. (2013) develop DVO-SLAM, which utilizes the photometric and depth geometry errors from all the pixels. It is much more efficient than other direct methods because it does not provide a dense voxel map. The loop closure is searched in a sphere around the keyframe position and selected based on the entropy ratio from the current keyframe to the loop candidate keyframe.

Because of the high computation cost by volume or surfel updating, most of the direct methods require GPU to perform in real time, which increases the surveying cost and limits their applications.

2.3.2 Feature-based methods

Feature-based methods are more efficient because they focus on only part of the salient features(i.e., points, lines, and planes). Therefore, they are preferred by the applications related to the mobile platform. Research works about point-based methods, line-based methods, plane-based methods, and MW-based methods are discussed in this section.

2.3.2.1 Point-based methods

Point-based methods detect and match the point features and then minimize the reprojection errors from the feature matches to compute the camera pose (Aulinas et al., 2008). This section first introduces the basic pipeline of these methods, and then summarizes the literature about monocular and RGB-D point-based methods, and finally, conclude the disadvantages of these methods for indoor mobile platform navigation.

In general, the basic pipeline of the feature point-based methods follows four steps:

(a) Feature extraction.

SIFT (Lowe, 1999) is the most popular handmade feature for Structure From Motion (SFM). However, its low speed is conflict with the requirement of the high-frequency output. With the purpose of the real-time implementation, low-cost features are favoured by the SLAM algorithms, such as FAST, Brisk, Harris, and ORB (Leutenegger et al., 2011; Rosten & Drummond, 2006; Rublee et al., 2011; Shi, 1994; Tareen & Saleem, 2018). Feature selection methods are studied, and different metrics are introduced to guide the feature selection, such as information gain, trace, and determinant (Davison, 2005; Kaess & Dellaert, 2009; Zhang et al., 2005; Zhao & Vela, 2018). Deep conventional neural networks are also explored for feature extraction with the development and availability of powerful hardware units (Di Febbo et al., 2018; Widya et al., 2018).

(b) Feature matching (or data association).

In general, point features are matched based on the descriptors (such as SIFT and ORB) or optical flow tracking (Horn & Schunck, 1981; Rublee et al., 2011; Tareen & Saleem, 2018). The outliers of feature matching can be removed by (a) ratio test; (b) cross-checking; and (c) geometry constraint.

(c) Pose estimation (or frame alignment).

The core of the SLAM algorithms is to compute the pose from valid feature matches. ICP and Perspective-n-Point (PnP) are two common algorithms for pose estimation in the point-based methods (Segal et al., 2009; Wu & Hu, 2006). The former depends on 3D-3D point correspondences, while the latter focuses on 3D-2D correspondences. Both algorithms attempt to minimize the geometric errors, while Liu et al. (2017) further combine the photometric errors to enhance the tracking accuracy of the RGB-D camera. In addition, the problem of pose estimation is often modelled as a factor graph in modern SLAM algorithms, and several optimization libraries are developed to provide fast and accurate solutions for this problem, such as g2o (Grisetti et al., 2011), ceres-solver (Agarwal & Mierle, 2012), iSAM2 (Kaess et al., 2012).

(d) Loop closure.

The SLAM algorithms should have the ability to detect and close the loop when the camera re-visits a place. The simplest way is to randomly match the current frame with the previous frames and then perform frame alignments. More effectively, loop candidates can be determined by Randomized Ferns (Glocker et al., 2014), Bag of Word (Gálvez-López & Tardos, 2012), and FABMAP (Cummins & Newman, 2008).

The first successful monocular SLAM system is Mono-SLAM (Davison et al., 2007). It constructs a probabilistic feature point map consisting of camera poses, feature points, and their estimation uncertainties. The map is updated by Extended Kalman Filter (EKF) using a constant-velocity motion model. PTAM is the next milestone (Klein & Murray, 2009). In PTAM, bundle adjustment is executed in real time in a SLAM system for the first time, owing to the multi-thread design and the sparse structure of the Hessian matrix.

An early RGB-D mapping system is developed by Henry et al. (2012), where FAST features and Calonder descriptors are applied to build the point feature matches. It uses the bag-of-word method to improve the speed of the loop closure detection and applies sparse bundle adjustment to improve the tracking accuracy. Engelhard et al. (2011) then introduce a hand-held RGB-D SLAM system for indoor mapping. The basic pipeline includes SURF feature extraction and matching, ICP for pose estimation, and pose graph optimization for trajectory refinement. Endres et al. (2013) extend this work comprehensively with more types of features and map representations. It provides SURF, SIFT, and ORB features and evaluates their accuracy, robustness, and runtime on TUM datasets (Sturm et al., 2012). The experiment results indicate ORB is the most suitable for the real-time application. Both point cloud and octree-based maps are provided for 3D reconstruction (Hornung et al., 2013). Mur-Artal and Tardós (2017) propose ORB-SLAM2 that can handle monocular, stereo and RGB-D frames. It is the first work composed of three threads: (a) camera tracking, (b) local mapping; and (c) loop closing. The comprehensive backend is constructed by bundle adjustments and loop closing, which can significantly lower the trajectory drift. Tang et al. (2018) introduce a hybrid SLAM system handling the 2D–2D, 3D–2D, and 3D–3D point pairs, in which the initial camera pose is determined by the ICP using 3D–3D point pairs and then refined using all the pairs. Dai et al. (2017) develop Bundle-Fusion which applies a sparse-to-dense approach for the global pose estimation. Coarse camera poses are obtained by matching sparse SIFT features and refined by combining dense photometric and geometric information. Real-time mapping is achieved based on surface reintegration on GPU.

Point features are essential for point-based methods. However, these features are hard to detect and match in low textured scenes, which may lead to the tracking failure of the mobile platform. On the other hand, line and plane features are still existing in such scenes, which can provide additional constraints for mobile platform tracking and enhance the continuity of the SLAM system.

2.3.2.2 Line-based methods

Line-based methods minimize the line reprojection errors from the line features for pose estimation. The basic pipeline is similar to that of point-based methods, but the method to detect and match line features is changed. This section introduces the progress of the line-based methods and shows their backward.

Lemaire and Lacroix (2007) propose a line-based monocular SLAM using EKF. The 3D line is represented by a Plücker coordinate and stored in a vector state together with the camera pose. Pumarola et al. (2017) build PL-SLAM upon ORB-SLAM, which is a monocular SLAM system. The 3D line is represented by the endpoints on the line, and the endpoint-to-line error is combined with the point reprojection error for pose estimation. Gomez-Ojeda et al. (2019) extend it to a stereo version and apply the line descriptors in the bag-of-words approach for the loop closure detection. He et al. (2018) develop a tightly coupled visual-inertial odometry fusing point and line features. The Plücker coordinate and orthonormal representation are applied to represent and update the 3D line in a sliding-window framework (Bartoli & Sturm, 2005). Lu and Song (2015) design a robust RGB-D odometry fusing both points and lines. It uses two endpoints to represent a 3D line. 3D points are sampled on the 3D line and used to build the 3D point-to-line errors. Fu et al. (2019) extend PL-SLAM to the RGB-D version. It also uses endpoints to parametrize the 3D line, and project the 3D line to the 2D line segment on an image. Zhou et al. (2018) present Canny-VO, which extracts Canny edge features and calculates the camera pose based on the 3D–2D edge alignment.

The inevitable backward of the line-based methods is the additional cost by the line features. In addition, these methods exploit either 3D–3D line correspondences or 3D–2D line correspondences. For the methods based on 3D lines, 2D line segments are neglected. On the other hand, for the methods based on 2D lines, the depth measurements on the lines are ignored. Therefore, part of the line information is not utilized by these methods, which may reduce the tracking and mapping quality. To avoid the disadvantage, this thesis proposes a new method combining both 3D-3D and 3D-2D line correspondences, which will be introduced in Chapter 3.

2.3.2.3 Plane-based methods

Plane-based methods pay attention to the plane transformation errors. The basic pipeline of these methods is similar to that of the point-based methods, but the methodologies to extract and match the plane features are different. This section reviews the significant research works in the plane-based methods and point out their disadvantages.

Kaess (2015) designs a 3D SLAM system using plane features and represents them by unit quaternions. However, there are not always enough planes to fully constrain the camera poses. Taguchi et al. (2013) propose a point-plane SLAM, which fuses point and plane features using a Random sample consensus (RANSAC) framework. Additional point features can provide more geometry constraints to help the pose estimation. Ma et al. (2016) develop CPA-SLAM, which performs a two-step approach to align the RGB-D frames: first frame-to-keyframe alignment and then frame-to-plane alignment. It applies planes for both scene representation and graph optimization. The disadvantage is that it requires GPU for real-time mapping. Hsiao et al. (2017) introduce a keyframe-based dense planar SLAM system. The initial camera pose is provided by a dense visual odometry (Kerl et al., 2013), and then used for plane association. It builds a factor graph to refine camera poses and plane features together, which consists of both visual odometry constraints and pose-to-plane constraints. Hsiao et al. (2018) extend this work to a planar-inertial system, which further exploits pre-integration IMU constraints and structural constraints between nearby planes. Zhang et al. (2019) develop a point-plane SLAM exploiting the supposed planes from the plane intersections. Contour points are used to robustly associate the planes and structural constraints are added to reduce the localization drift. Li et al. (2020) develop an improved plane fitting approach by minimizing the radial distances with a rigorous error model, which can fit planes from noisy depth images with high accuracy.

Weighting point and plane measurements is essential for accurate pose estimation. However, most of these plane-based methods just assign experimental weights for different kinds of feature measurements, which is non-optimal for pose estimation. To avoid this disadvantage, this thesis derives the covariance of the plane measurements by plane fitting and covariance propagation, and fuse the point and plane features based on their analytical covariances in Chapter 4.

2.3.2.4 MW-based methods

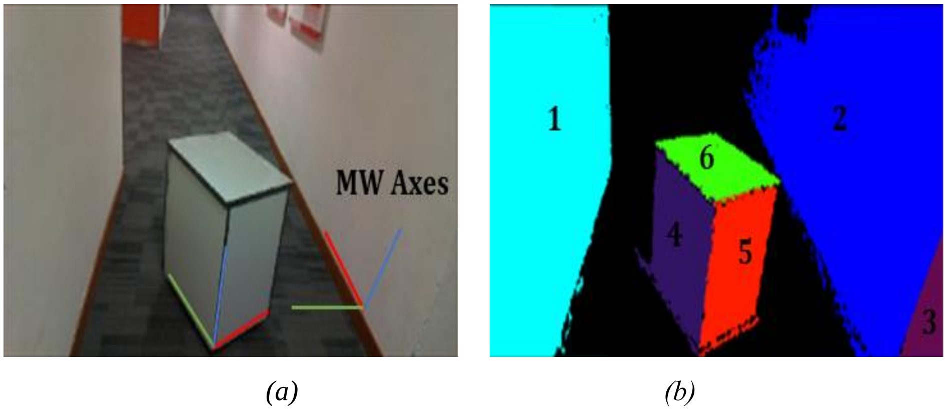
Besides the measurements from the point, line, and plane features, MW-based methods attempt to further constrain the camera pose by exploiting the MW assumption. It assumes that all the surfaces in the operation scenes are aligned with three orthogonal directions, which define the MW axes. This section introduces the basic pipeline of the MW-based methods, summarizes the important works, and finally points out the backward of these methods.

The pipeline of the MW-based methods is different from other feature-based methods. In general, the rotation and translation of the camera pose are decoupled and computed separately in the MW-based methods. In addition, MW axes are built from the parallel lines and main planes for the rotation estimation.

Zhou et al. (2016) propose a mean shift algorithm to track the plane directions for the rotation estimation, and the translation component is solved by three 1D density alignments. Kim et al. (2018b) combine the vanishing directions of lines and the normal vectors of planes to recover the drift-free rotation, and then compute the translation vector by minimizing the de-rotated reprojection errors. This work is then extended to a linear SLAM system based on Kalman Filter by Kim et al. (2018). The rotation matrix is computed using the structure regularity, and then the translation vector and the plane features are updated as a state vector in the EKF framework. Li et al. (2020a) propose Structure-SLAM, where camera rotation is estimated by the MW frame alignment using parallel lines and surface normals. Then the translation is computed from the reprojection errors of point and line features. It is extended to a more comprehensive framework by Li et al. (2020) which refines the translation estimation module by fusing point, line, and plane features.

The main disadvantage of these MW-based methods is that they may fail in operation scenes not satisfying the MW assumption strictly, as shown in Figure 2.6. For example, while Planes 1-3 and 6 extracted from walls can be aligned with the MW axes, Planes

4-5 from the cabinet are not parallel to any direction of the MW axes. Instead of using the MW assumption to exploit the structural regularity, this thesis proposes a new representation form for the plane features in Chapter 4, based on the parallel and vertical relationships among planes and MW axes.



(a) *(b)*

Figure 2. 6: An indoor scene does not satisfy the MW assumption strictly. (a) RGB capture of the scene; (b) plane segmentation result.

2.4 Sensor fusion

Though the continuity of the point-based methods is enhanced by fusing line and plane features, the tracking drift of the mobile platform is still a problem. In modern SLAM systems, backend optimization techniques are usually applied to reduce the tracking drift (Grisetti et al., 2011; Kaess et al., 2012). For a mobile platform equipped with additional internal sensors, sensor fusion is a commercial and effective solution to further reduce the drift and enhance the tracking accuracy (Campos et al., 2020; Leutenegger et al., 2015; Qin et al., 2018). Compared with the pipeline of Visual SLAM algorithms, the algorithms of sensor fusion need to additionally process the measurements from the internal sensors and fuse them with the feature measurements from the cameras.

This section firstly introduces the importance of the sensor fusion algorithms and their difference with Visual SLAM algorithms. Then it summarizes the research works about sensor fusion algorithms. Finally, the disadvantages of these algorithms are discussed. Specifically, the research works are divided into three parts: (a) fusing the camera and the IMU; (b) fusing the camera and the wheel odometer; (c) fusing the RGB-D camera and the wheel odometer.

The algorithms fusing the IMU and the camera are called visual-inertial SLAM, and can be divided into the filtering-based approaches (Li, 2014; Li & Mourikis, 2012, 2013; Zhang et al., 2020) and the optimization-based approaches (Campos et al., 2020; Leutenegger et al., 2015; Qin et al., 2018). MSCKF (Li & Mourikis, 2013) is an EKF-based approach, which builds a sliding window to store the state vector of the latest keyframe poses. Its computation complexity is linear in the number of features by excluding them in the state vector. Forster et al. (2016) introduce a pre-integration theory to iteratively compute the inertial constraints between the consecutive keyframes. The theory is then commonly applied in the SOTA visual-inertial systems (Campos et al., 2020; Leutenegger et al., 2015; Qin et al., 2018). Leutenegger et al. (2015) propose OKVIS, where visual and inertial constraints are processed simultaneously in a nonlinear optimization framework. ORB-SLAM3 and VINS-mono further implement the modules of loop closure and map reuse to improve the tracking accuracy (Campos et al., 2020; Qin et al., 2018).

Similar to the IMU, the wheel odometer can be fused with the camera to enhance the tracking accuracy of the mobile platform (Kang et al., 2019; Liu et al., 2019; Wu et al., 2017; Zheng & Liu, 2019; Zheng et al., 2018). Wu et al. (2017) derive the unobservability of the visual-inertial system on the wheeled platform and then incorporate the planar motion constraints to improve the tracking and mapping accuracy. They compare the deterministic and stochastic constraints of planar motion and show the benefits of the stochastic one, which is also utilized by other visual-wheeled localization systems (Liu et al., 2019; Quan et al., 2019; Zheng et al., 2018). Quan et al. (2019) design a novel odometer error by fusing the wheel encoder and gyroscope measurements, and combine this error with the point reprojection error using factor graph optimization. Liu et al. (2019) present a comprehensive positioning system by combining the IMU, the camera, and the wheel encoder. It calibrates the extrinsic parameters among these sensors in real time, which can help to improve the accuracy of the vehicle pose.

The above methods can provide accurate positions for the mobile platform, but they cannot reconstruct the dense surrounding scenes, which is essential for obstacle avoidance and path planning. Accurate localization and dense mapping can be simultaneously implemented on a CPU in real time by fusing the RGB-D camera and the wheel odometer. However, the related studies are still few (Labbé & Michaud, 2019; D. Yang et al., 2019). RTAB-Map is a comprehensive localization and mapping system which supports a variety of sensors(i.e., IMU, camera, Lidar, and wheel odometry) (Labbé & Michaud, 2019). It assumes the platform moves on the ground plane with no perturbations and uses a loose-coupled design to combine the RGB-D camera and the wheel odometer. Yang et al. (2019) present DRE-SLAM, which tightly fuses the RGB-D camera and wheel encoders in a factor graph. Dynamic objects are detected and removed to improve the localization accuracy in the dynamic scenes. However, it also employs a deterministic constraint for the planar motion assumption. The tracking and mapping accuracy can be reduced by the platform vibration on the uneven floor.

To sum up, the research works about fusing the wheel odometer and the RGB-D camera are still few. The main disadvantage of these methods is that they apply a strict assumption for the motion of the mobile platform, which is not practical and may lead to non-optimal estimation results. In Chapter 5, a soft planar motion assumption is applied and a two-stage strategy is proposed to examine the assumption. To further improve the accuracy of the RGB-D SLAM system, the constraints from the RGB-D camera, the wheel odometer, and the planar motion assumption are tightly fused in a factor graph.

CHAPTER 3

RGB-D SLAM FUSING POINT AND LINE FEATURES FOR LOW TEXTURED SCENES

As discussed in Section 2.3.2, point-based methods are efficient and can be implemented on a commercial CPU in real time, which makes it suitable for the application of the mobile platform. However, in low textured scenes, these methods cannot provide reliable constraints because few points are extracted and many of them are wrongly matched. Despite low texture, most indoor scenes contain abundant high-level geometry primitives(i.e., line features), which can be fused to aid mobile platform tracking. Lines have been widely applied in monocular and stereo SLAM systems (Gomez-Ojeda et al., 2019; He et al., 2018; Jeong & Lee, 2006; Lemaire & Lacroix, 2007; Yanyan Li, Nikolas Brasch, et al., 2020a; Pumarola et al., 2017; Zuo et al., 2017) but attracted less attention in the RGB-D research area (Fu et al., 2019; Lu & Song, 2015; Y. Zhou et al., 2018). Moreover, the existing line-based methods exploit either 3D line features or 2D line features and neglect part of the line information.

In this chapter, to improve the continuity of mobile platform tracking in low textured scenes, a new RGB-D SLAM system fusing point and line features are proposed. The main contributions are as follows:

(a) It exploits both 3D-3D and 3D-2D line correspondences and builds a new cost function by fusing the 3D and 2D line reprojection errors, which can utilize more line information than the previous line-based methods.

(b) A new factor graph is built based on the above reprojection errors. It exploits more line constraints than the previous line-based methods during bundle adjustment and therefore can improve the tracking quality.

(c) The proposed system is evaluated on a public dataset and two real-world experiments. It yields the same-level accuracy in rich textured scenes compared with SOTA methods, and generates high continuity and accuracy in low textured scenes. In a lab room experiment, owing to the fusion of 3D and 2D line features, the proposed system generates better localization results than the systems utilizing 3D or 2D line features, and can improve their localization accuracies by 22.5% and 25.8%. In a corridor experiment, the improvements of the mapping accuracy over the methods utilizing 3D or 2D line features are 10.2% and 14.7%, respectively.

The remaining content of this chapter is divided into five parts, which focus on the system overview, preliminaries, the full pipeline of the proposed system, the experiments and results, and the summary, respectively.

3.1 System overview

The proposed system is built upon the open-source visual-inertial system FLVIS (Chen et al., 2020), which develops a feedback/feedforward loop to fuse the data from IMU and stereo/RGB-D cameras. To work in low textured scenes with only an RGB-D camera, the function for IMU processing is disabled and specific support for the line features is added.

As shown in Figure 3.1, the proposed system has two parts: frontend and backend. A feature map is maintained to store the camera poses, points, and lines, which can be updated by both frontend and backend.

(a) Frontend: the frontend has one thread for pose tracking. Firstly, point and line features are detected in the current frame and matched with the previous frame. Secondly, the 3D information of the matched features in the world coordinate is searched in the feature map. Thirdly, a robust pose solver is built based on the point and line reprojection errors, and the wrong matches are deleted by the

Mahalanobis distance test. Fourthly, the camera pose is outputted, and the 3D model is expanded. Finally, the keyframe decision is made based on the relative motion and the matched features from the previous keyframe. The feature map will be updated if a new keyframe comes.

(b) Backend: the backend has two threads, local mapping and loop closing. In the local mapping thread, a novel factor graph is built to update the feature map when a new keyframe arrives. In the loop closing thread, firstly, the arrived keyframe is transferred to a word vector by the bag-of-word approach, and the loop candidate is detected by the word vector comparison. The loop candidate is then verified by a geometry test by a RANSAC PnP algorithm. Finally, the loop closure is corrected by pose graph optimization.

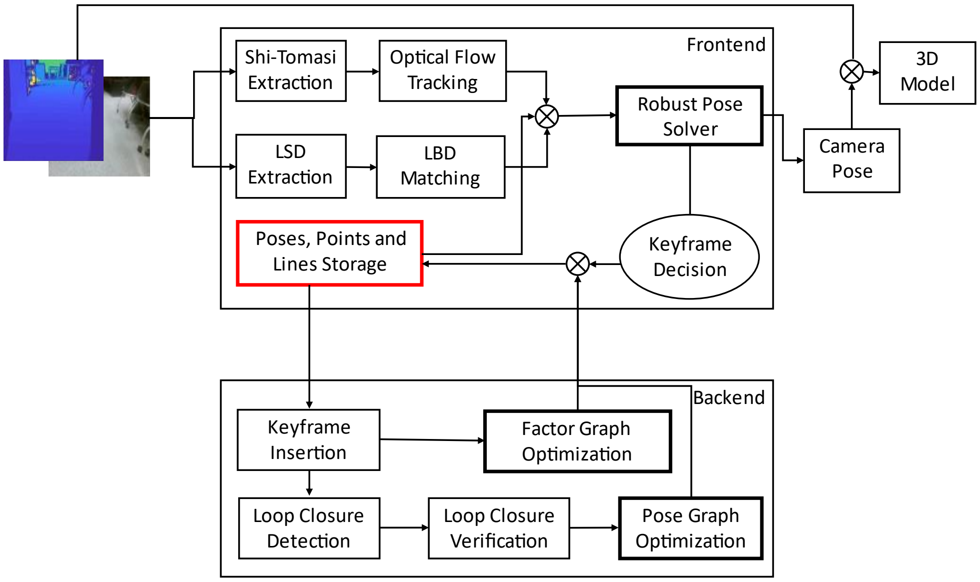


Figure 3. 1: Overview of the proposed system fusing point and line features.

3.2 Preliminaries

This section introduces the representation types of the camera poses, point and line features as shown in Figure 3.2. The motion of the mobile platform can be computed based on the camera pose and the extrinsic matrix between the camera and the mobile platform.

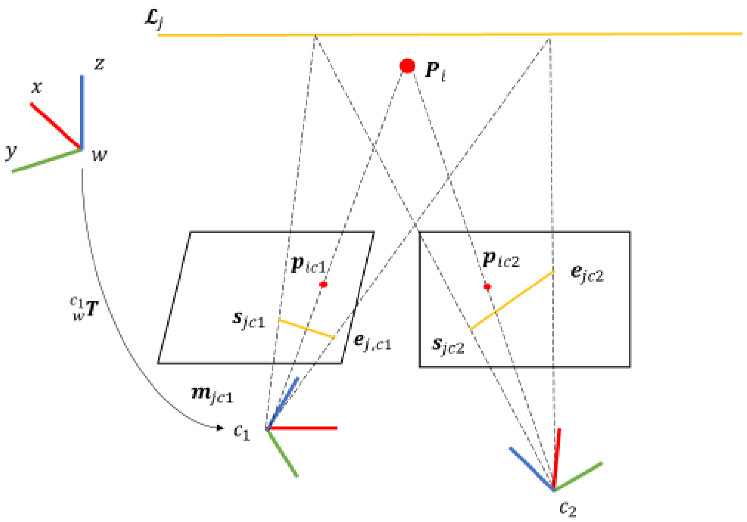


Figure 3. 2: An illustration of the RGB-D camera and point and line measurements.

3.2.1 Camera pose representation

This study assumes that all the depth measurements have been calibrated and registered to the RGB camera frame, so only the RGB camera frame is considered for the coordinate transformation. The world frame is defined as the initial frame of the RGB-D camera. The camera pose is defined as the 6-DoF motion between the world frame and the RGB-D frame, which consists of a 3-DoF translation and a 3-DoF rotation. The translation is simply represented by a 3×1 vector, while the rotation has various representation types (Barfoot, 2017): (a) rotation matrix; (b) rotation vector (Lie algebra); (c) unit quaternion; (d) Euler angles.

(a) The rotation matrix is a 3×3 orthogonal matrix whose determinant equals 1. The set of rotation matrices forms a Lie group, specifically,3D Special Orthogonal Group (SO3).

$$\mathrm{SO}(3) = \{\boldsymbol{R}_{3\times 3} | \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\} \tag{3.1}$$

(b) The rotation vector is the Lie algebra corresponding to SO(3). It consists of a rotation axis $\boldsymbol{n} = [n_1, n_2, n_3]^T$ and a rotation angle α, and can be transformed from a rotation matrix by

$$\boldsymbol{R} = \cos\alpha \boldsymbol{I} + (1 - \cos\alpha)\boldsymbol{n}\boldsymbol{n}^T + sin\alpha\boldsymbol{n}^{\wedge} \tag{3.2}$$

$$\boldsymbol{n}^{\wedge} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix} \tag{3.3}$$

where ^ indicates the transform from a vector to a skew-symmetric matrix.

(c) The unit quaternion can be transformed from the rotation vector by

$$\boldsymbol{q} = [q_1, q_2, q_3, q_4]^T = \left[\cos\frac{\alpha}{2}, n_1\sin\frac{\alpha}{2}, n_2\sin\frac{\alpha}{2}, n_3\sin\frac{\alpha}{2}\right]^T \tag{3.4}$$

(d) The Euler angles consist of three rotation angles around different axes.

Euler angles suffer from the problem of singularity, so it is not suitable for the interpolation and iteration steps in the SLAM algorithms. Both rotation matrix and unit quaternion are not compact. The former has 9 parameters, and the latter has 4 parameters, so additional constraints are required for 3-DoF rotation estimation. Lie algebra is adopted in this thesis for its compactness. For the representation of the full 6-DoF camera pose, transformation matrix $\boldsymbol{T}$ is constructed by a rotation matrix and a translation vector, whose set forms 3D Special Euclidean Group (SE(3)).

$$\mathrm{SE}(3) = \left\{\boldsymbol{T}_{4\times4} = \begin{bmatrix} \boldsymbol{R}_{3\times3} & \boldsymbol{t}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 1 \end{bmatrix} | \boldsymbol{R} \in \mathrm{SO}(3)\right\} \tag{3.5}$$

Its Lie algebra is utilized for the iterative motion estimation and is associated with the Lie group by the matrix exponentials and logarithms.

$$\boldsymbol{\xi}_{6\times1} = \log(\boldsymbol{T}_{4\times4})^{\vee}, \boldsymbol{T}_{4\times4} = \exp(\boldsymbol{\xi}_{6\times1}^{\wedge}) \tag{3.6}$$

3.2.2 Point representation

Two types of representations for the point feature have been generally utilized in SLAM algorithms: (a) its 3D position in the world frame; (b) its inverse depth on the first keyframe observing it. The second type can deal with large-depth scenes, but it involves the keyframe pose and is more complicated for transformation between different frames. The first type is selected in this thesis, which is more widely used. It is assumed that the 2D pixel measurement of a 3D point P_i is $\boldsymbol{p}_{ic} = [u_{ic}, v_{ic}]^T$ and its depth measurement is pd_{ic}. When $\boldsymbol{P}_i$ is observed by a new keyframe for the first time, its 3D position in the world frame can be recovered $\boldsymbol{P}_{iw} = [x_{iw}\,, y_{iw}\,, z_{iw}]^T$ and added to the feature map.

3.2.3 Line representation

Lines can be represented by various types: (a) two endpoints; (b) Cartesian coordinate; (c) Plücker coordinate; (d) orthonormal representation. Both the Plücker coordinate and orthonormal representation are utilized for line parameterization in this chapter. The Plücker coordinate is convenient for the line transformation and projection, while orthonormal representation is compact with four DoFs.

As shown in Figure 3.3, the Plücker coordinate consists of two 3D vectors $\boldsymbol{d}$ and $\boldsymbol{m}$. It can be initialized by two points on the 3D line:

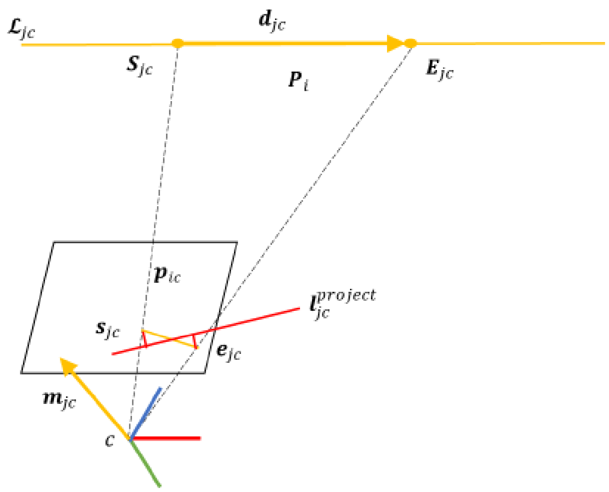


Figure 3. 3: Plücker coordinate of a straight line.

$$\boldsymbol{\mathcal{L}}_{jc} = \begin{bmatrix} \boldsymbol{E}_{jc} \times \boldsymbol{S}_{jc} \\ \boldsymbol{E}_{jc} - \boldsymbol{S}_{jc} \end{bmatrix} = \begin{bmatrix} \boldsymbol{m}_{jc} \\ \boldsymbol{d}_{jc} \end{bmatrix} \tag{3.7}$$

where $\boldsymbol{\mathcal{L}}_{jc}$ is the Plücker coordinate of the 3D line in the camera frame, $\boldsymbol{S}_{jc}$ and $\boldsymbol{E}_{jc}$ are two points on the line, $\boldsymbol{m}_{jc}$ is the normal of the plane constructed by the line and the frame origin, and $\boldsymbol{d}_{jc}$ is the line direction.

To avoid the overparameterization problem caused by the Plücker coordinate with six parameters, orthonormal representation $(\boldsymbol{U}, \boldsymbol{W}) \in \mathrm{SO}(3) \times \mathrm{SO}(2)$ is applied. The convention between the orthonormal representation and the Plücker coordinate is simply given below, and the detail can be referred to (Bartoli & Sturm, 2005; Zuo et al., 2017).

$$\boldsymbol{U} = \ \boldsymbol{R}(\boldsymbol{\varphi}) = \begin{bmatrix} \frac{\boldsymbol{m}}{||\boldsymbol{m}||} & \frac{\boldsymbol{d}}{||\boldsymbol{d}||} & \frac{\boldsymbol{m} \times \boldsymbol{d}}{||\boldsymbol{m} \times \boldsymbol{d}||} \end{bmatrix} \tag{3.8}$$

where $\boldsymbol{U}$ is a 3D rotation matrix and $\boldsymbol{\varphi} = \left[\varphi_x \, , \varphi_y \, , \varphi_z\right]^T$ is the rotation vector.

$$\boldsymbol{W} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \frac{1}{\sqrt{||\boldsymbol{m}||^2 + ||\boldsymbol{d}||^2}} \begin{bmatrix} ||\boldsymbol{m}|| & -||\boldsymbol{d}|| \\ ||\boldsymbol{d}|| & ||\boldsymbol{m}|| \end{bmatrix} \tag{3.9}$$

where $\boldsymbol{W}$ is the 2D rotation matrix and θ is the rotation angle.

$\boldsymbol{\psi} = [\boldsymbol{\varphi}^T, \theta]^T$ is used for the minimal representation during the sliding window bundle adjustment. The Plücker coordinate of the 3D line can be transferred from the optimized $\boldsymbol{\psi}$ by

$$\boldsymbol{\mathcal{L}} = [cos\theta\boldsymbol{u}_1^T \quad sin\theta\boldsymbol{u}_2^T] \tag{3.10}$$

where $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are the first and second columns of $\boldsymbol{U}$.

3.3 Fusing point and line features

The full pipeline of fusing the point and line features is introduced in this section, which consists of feature extraction and matching, robust pose solver, factor graph construction, and loop closure.

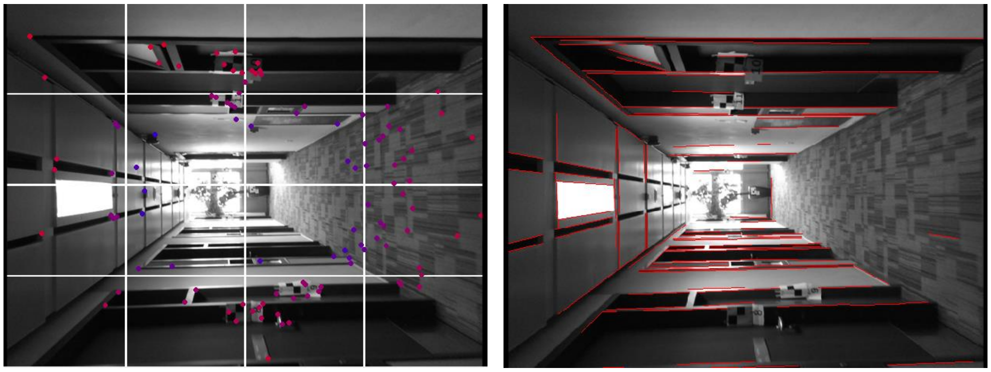
3.3.1 Extraction and matching of point and line features

Shi-Tomasi is used as the point feature extractor, which is improved based on the Harris corner (Shi, 1994). As shown in Figure 3.4(a), the image plane is divided into 16 regions and newly detected features are added to these regions based on the score of the Harris index (Harris & Stephens, 1988). The maximum number of features in every region is set as 30.

For the first frame, points are extracted from the colour image and their depths are recovered from the depth image. These points are then added to the feature map as landmarks. For the following frames, these points are tracked by the Lucas–Kanade optical flow (Lucas & Kanade, 1981), and new points will be re-extracted and selected from the 16 regions until the maximum number is reached.

A more uniform distribution of point features is achieved by dividing the image into smaller regions and controlling the number of features in these regions. The values of the region number and the feature number in the region are tuned, and it is argued that 16 regions with a maximum of 30 features are suitable for images with 640×480 resolution.

Line Segment Detector (LSD) is applied as the line feature extractor as shown in Figure 3.4(b), which can detect the line segments with high accuracy and fast speed (Von Gioi et al., 2012). The binary descriptor for the line segment is extracted using Line Band Descriptor (LBD), which is an efficient line descriptor with both appearance and geometry constraints (Zhang & Koch, 2013).



(a) *(b)*

Figure 3. 4: Point and line feature extractor. (a) Improved Shi-Tomasi extractor; (b) LSD.

The combination of LSD and LBD can be implemented on a CPU in real time, and therefore has been widely applied in the line-based SLAM methods (Fu et al., 2019; Gomez-Ojeda et al., 2019; Yanyan Li, Nikolas Brasch, et al., 2020b; Lu & Song, 2015; Pumarola et al., 2017). To further improve the speed, the number of the line features is controlled, and its maximum is settled as 100. Line features are selected based on the length and distribution and those with small lengths or near the boundary of the image are less likely to be selected.

To effectively remove the outlier matches of the line features, the appearance information from LBD and the geometry information from LSD are combined for line feature matching, and a comprehensive three-step method is detailed below:

(a) Cross-check. FLANN (Muja & Lowe, 2009) is applied twice to match the line descriptors. In the first matching, the descriptors from the previous frame are set as the query set, and those from the current frame are set as the train set. A matched descriptor DesT1 in the train set can be found for a descriptor DesQ1 in the query set. By contrast, in the second matching, the descriptors from the previous frame are set as the train set, and those from the current frame are set as the query set. Again, DesT2 and DesQ2 can be found. DesT1 and DesQ2 are from the previous frame, while DesT2 and DesQ1 are from the current frame. If DesT1 and DesQ2 are the same descriptors, then DesT2 and DesQ1 should also have the same descriptor indices. Otherwise, they will be removed as the wrong feature match.

(b) Ratio-test. It is assumed that DesT1 is the matched feature of DesQ1 after cross-check, which means DesT1 has the smallest distance from DesQ1 among the train set. The ratio between DesT1 and the second smallest distance should be smaller than 0.75. Otherwise, the feature match for DesT1 and DesQ1 is rejected.

(c) Geometry test. Based on the indexes of DesT1 and DesQ1, corresponding line segments are associated. LSD provides the orientation, length, and endpoints of the line segments. If the line segments have highly different orientations, lengths, or endpoints, the line match will be discarded and not used for the pose estimation.

For a new keyframe, if extracted line features are not matched to any line landmark in the feature map, their Plücker coordinates will be computed and then inserted into the feature map using Eq. (3.7).

3.3.2 Robust pose solver utilizing point and line features

In this part, an infinite impulse response (IIR) filter is firstly introduced for updating the point landmark in the feature map. The IIR filter is a recursive filter, which outputs the 3D position of the point landmark in the current camera frame based on the measured 3D position and the projected 3D position. As the depth measurements are utilized to compute the measured 3D position, they are neglected during the pose estimation and only the pixel measurements are used. The 2D point reprojection error, 2D and 3D line reprojection errors are then derived, respectively. Finally, they are combined to build a new cost function and detect the wrong feature matches based on the Mahalanobis distance test.

3.3.2.1 Infinite iImpulse rponse filter

If the tracked point feature has a reliable depth measurement, e.g., pd_{ic} is larger than 0.2 m and smaller than 6.0 m, its measured 3D position in the camera frame is derived by

$$\boldsymbol{P}_{ic}^{measure} = pd_{ic}\boldsymbol{K}^{-1}[u_{ic}, \quad v_{ic}, \quad 1]^T, \boldsymbol{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

where $\boldsymbol{K}$ is the intrinsic parameter matrix.

The 3D position in the camera frame can be also projected from that in the world frame using the transformation matrix

$$\boldsymbol{P}_{ic}^{project} = {}_{w}^{c}\boldsymbol{R}\boldsymbol{P}_{iw} + {}_{w}^{c}\boldsymbol{t} \tag{3.12}$$

The IIR filter is then applied to update the 3D position by

$$\boldsymbol{P}_{ic} = \lambda\boldsymbol{P}_{ic}^{project} + (1-\lambda)\,\boldsymbol{P}_{ic}^{measure} \tag{3.13}$$

where λ is the parameter of the IIR filter. The advantage of the IIR filter is that it utilizes all the measurements of the point landmark throughout the lifespan. The position error of the landmark will converge faster, and the negative effect of the depth outlier will be lowered. From the experience of tuning, the IIR filter works better if λ is set between 0.6 and 0.9. The value of λ indicates the confidence in the historical information in the feature map, while the value of $1-\lambda$ means the confidence of the quality of the feature extractor and the depth measurements.

3.3.2.2 2D point reprojection error

If point $\boldsymbol{P}_{iw}$ is tracked to the camera frame, and the pixel measurements on the image plane are $\boldsymbol{p}_{ic}$, 2D point reprojection error is derived as

$$\boldsymbol{r}_{ic}^{2p} = \boldsymbol{p}_{ic} - f\left(\boldsymbol{K}({}_{w}^{c}\boldsymbol{R}\boldsymbol{P}_{iw} + {}_{w}^{c}\boldsymbol{t})\right), f\left(\begin{bmatrix} a \\ b \\ c \end{bmatrix}\right) = \begin{bmatrix} a/c \\ b/c \end{bmatrix} \tag{3.14}$$

where f is a normalization function.

3.3.2.3 3D line reprojection error

Plücker coordinate $\boldsymbol{\mathcal{L}}_{jw}$ can be transformed from the world frame to the camera frame by

$$\boldsymbol{\mathcal{L}}_{jc}^{project} = \begin{bmatrix} \boldsymbol{m}_{jc}^{project} \\ \boldsymbol{d}_{jc}^{project} \end{bmatrix} \begin{bmatrix} {}_{w}^{c}\boldsymbol{R} & {}_{w}^{c}\boldsymbol{t}^{\wedge}\, {}_{w}^{c}\boldsymbol{R} \\ \boldsymbol{0} & {}_{w}^{c}\boldsymbol{R} \end{bmatrix} \boldsymbol{\mathcal{L}}_{jw} \tag{3.15}$$

If its associated 2D line segment is detected on the current frame, then the 2D pixels are sampled on the line and projected to the 3D space. If more than 70% of these points have reliable depth measurements, it is argued that the depth measurements along the 2D line segment are reliable. These points can be robustly fitted to the Cartesian coordinate and then transferred to the Plücker coordinate $\boldsymbol{\mathcal{L}}_{jc}$.

Because both $\boldsymbol{\mathcal{L}}_{jc}^{project}$ and $\boldsymbol{\mathcal{L}}_{jc}$ have six parameters that are over-parameterized, they are not used to build the 3D line reprojection error. Instead, they are transformed to orthonormal representation $\boldsymbol{\psi}_{jc}^{project}$ and $\boldsymbol{\psi}_{jc}$ for compact comparison by Eq. (3.8) and (3.9). The 3D line reprojection error is derived as

$$\boldsymbol{r}_{jc}^{3l} = \boldsymbol{\psi}_{jc} - \boldsymbol{\psi}_{jc}^{project} \tag{3.16}$$

3.3.3.4 2D line reprojection error

If the depth measurements along the matched 2D line segment are not reliable, the Plücker coordinate $\boldsymbol{\mathcal{L}}_{jc}^{project}$ are projected from the camera frame to the image plane by

$$\boldsymbol{l}_{jc}^{project} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = \boldsymbol{\mathcal{k}}\boldsymbol{m}_{jc}^{project} = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y c_x & -f_x c_y & f_x f_y \end{bmatrix} \boldsymbol{m}_{jc}^{project} \tag{3.17}$$

where $\boldsymbol{m}_{jc}^{project}$ is the plane normal of $\boldsymbol{\mathcal{L}}_{jc}^{project}$, $\boldsymbol{l}_{jc}^{project}$ is the 2D projected line and $\boldsymbol{\mathcal{k}}$ is the line projection matrix.

As shown in Figure 3.3, the 2D line reprojection error is defined as the distance from the endpoints to the projected line $\boldsymbol{l}_{jc}^{project}$

$$\boldsymbol{r}_{jc}^{2l} = \left[r_{jc}^{2s}, \quad r_{jc}^{2e}\right]^T = \left[\frac{u_{jc}^s l_1 + v_{jc}^s l_2 + l_3}{\sqrt{l_1^2 + l_2^2}}, \quad \frac{u_{jc}^e l_1 + v_{jc}^e l_2 + l_3}{\sqrt{l_1^2 + l_2^2}}\right]^T \tag{3.18}$$

where $\boldsymbol{s}_{jc} = [u_{jc}^s, \quad v_{jc}^s]^T$ and $\boldsymbol{e}_{jc} = [u_{jc}^e, \quad v_{jc}^e]^T$ are the endpoints of the detected line segment on the image plane.

3.3.3.5 Novel cost function

Eq. (3.14), (3.16), and (3.18) are combined to build a novel cost function below

$$\sum_i \rho\left(\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^2\right) + \sum_j \rho\left(\left\|\boldsymbol{r}_{jc}^{3l}\right\|_{\boldsymbol{\Sigma}_{jc}^{3l}}^2\right) + \sum_j \rho\left(\left\|\boldsymbol{r}_{jc}^{2l}\right\|_{\boldsymbol{\Sigma}_{jc}^{2l}}^2\right) \tag{3.19}$$

where ρ is the Huber function and $\boldsymbol{\Sigma}$ is the covariance matrix associated with the reprojection error. The covariance of $\boldsymbol{r}_{ic}^{2p}$ is a 2×2 identity matrix times the variance of pixel measurements

$$\boldsymbol{\Sigma}_{ic}^{2p} = \sigma_p^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.20}$$

where $\sigma_p^2 = 1/12$ (Proença & Gao, 2018). Eq. (3.18) indicates that $\boldsymbol{\Sigma}_{jc}^{2l}$ can be propagated from the covariance of the endpoint pixels $\boldsymbol{s}_{jc}$ and $\boldsymbol{e}_{jc}$.

$$\boldsymbol{\Sigma}_{jc}^{2l} = \frac{\partial \boldsymbol{r}_{jc}^{2l}}{\partial \boldsymbol{se}_{jc}} diag\left(\sigma_{u_{jc}^s}^2, \sigma_{v_{jc}^s}^2, \sigma_{u_{jc}^e}^2, \sigma_{v_{jc}^e}^2\right) \frac{\partial \boldsymbol{r}_{jc}^{2l}}{\partial \boldsymbol{se}_{jc}}^T \approx \begin{bmatrix} \sigma_{u_{jc}^s}^2 & 0 \\ 0 & \sigma_{u_{jc}^e}^2 \end{bmatrix} \tag{3.21}$$

$$\boldsymbol{se}_{jc} = \begin{bmatrix} \boldsymbol{s}_{jc} \\ \boldsymbol{e}_{jc} \end{bmatrix}, \frac{\partial \boldsymbol{r}_{jc}^{2l}}{\partial \boldsymbol{se}_{jc}} = \begin{bmatrix} \frac{l_1}{\sqrt{l_1^2+l_2^2}} & \frac{l_2}{\sqrt{l_1^2+l_2^2}} & 0 & 0 \\ 0 & 0 & \frac{l_1}{\sqrt{l_1^2+l_2^2}} & \frac{l_2}{\sqrt{l_1^2+l_2^2}} \end{bmatrix}^T \tag{3.22}$$

On the other hand, $\boldsymbol{\Sigma}_{jc}^{3l}$ is equal to the covariance of 3D line measurements $\boldsymbol{\psi}_{\boldsymbol{jc}}$, which can be propagated from $\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{L}}_{jc}}$ and $\frac{\partial \boldsymbol{r}_{jc}^{3l}}{\partial \boldsymbol{\mathcal{L}}_{jc}}$.

$$\boldsymbol{\Sigma}_{jc}^{3l} = \frac{\partial \boldsymbol{r}_{jc}^{3l}}{\partial \boldsymbol{\mathcal{L}}_{jc}} \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{L}}_{jc}} \frac{\partial \boldsymbol{r}_{jc}^{3l}}{\partial \boldsymbol{\mathcal{L}}_{jc}}^T \ , \ \frac{\partial \boldsymbol{r}_{jc}^{3l}}{\partial \boldsymbol{\mathcal{L}}_{jc}} = \frac{\partial \boldsymbol{\psi}_{jc}}{\partial \boldsymbol{\mathcal{L}}_{jc}} \tag{3.23}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{L}}_{jc}}$ and $\frac{\partial \boldsymbol{\psi}_{jc}}{\partial \boldsymbol{\mathcal{L}}_{jc}}$ are computed during line fitting (Bartoli & Sturm, 2005).

In general, point-based methods utilizes the PnP algorithm for pose estimation and the cost function is $\sum_i \rho\left(\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^2\right)$. The cost function for the line-based methods based on 3D-3D line correspondences is $\sum_i \rho\left(\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^2\right) + \sum_j \rho\left(\left\|\boldsymbol{r}_{jc}^{3l}\right\|_{\boldsymbol{\Sigma}_{jc}^{3l}}^2\right)$, and that for the methods based on 3D-2D line correspondences is $\sum_i \rho\left(\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^2\right) + \sum_j \rho\left(\left\|\boldsymbol{r}_{jc}^{2l}\right\|_{\boldsymbol{\Sigma}_{jc}^{2l}}^2\right)$. The novel cost function can improve the continuity of the RGB-D SLAM system in low textured scenes owing to fusing line features. Furthermore, compared with the methods using either 3D-3D or 3D-2D line correspondences, the novel cost function utilizes more constraints from line features and can generate better pose estimation results.

The iterative Gauss-Newton method implemented in g2o is applied to minimize the cost function and solve the camera pose ${}_{w}^{c}\boldsymbol{\xi}$ (Grisetti et al., 2011). The Chi-Square test is applied during the pose estimation process to remove the outlier feature matches and enhance the tracking quality. The details are as below:

(a) The initial camera pose is solved by the RANSAC PnP method before minimizing the function. The wrong matches among the 2D features are filtered out by RANSAC. If the relative motion between the previous frame and the initial camera pose exceeds a threshold, the initial guess from RANSAC PnP will be rejected and is calculated again using a constant-velocity motion model. The implementation of RANSAC PnP from OpenCV is directly used (Bradski & Kaehler, 2008). After 100 iterations, the point feature will be removed if its reprojection error is larger than three pixels. It is assumed that the camera motion during a short period follows a constant-velocity assumption, so the camera pose of the current frame can be predicted using the velocity and the camera pose of the previous frame.

(b) For all the line matches, their 2D line reprojection errors are calculated based on the initial camera pose and the 3D line landmarks in the feature map using Eq. (3.18). The line matches associated with large initial line reprojection errors are filtered out.

(c) The remaining feature matches are then sent to Eq. (3.19) for optimization. After every four iterations, wrong matches will be removed if they fail in the Mahalanobis distance test. Optimization is continued using the remaining matches.

$$\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^{2} < \chi_{\alpha,n_{2p}}, \left\|\boldsymbol{r}_{jc}^{3l}\right\|_{\boldsymbol{\Sigma}_{jc}^{3l}}^{2} < \chi_{\alpha,n_{2l}}, \left\|\boldsymbol{r}_{jc}^{2l}\right\|_{\boldsymbol{\Sigma}_{jc}^{2l}}^{2} < \chi_{\alpha,n_{3l}} \qquad (3.24)$$

where α the threshold of Chi-Square distribution, and $n_{2p} = 2$, $n_{3l} = 4$ and $n_{2l} = 2$ are the degrees of freedom associated with the reprojection error.

The analytical Jacobian matrices of line reprojection errors with respect to camera pose are derived in previous line-based methods (Bartoli & Sturm, 2005; Zhang et al., 2015). The proposed system uses the automatic differentiation in g2o to compute the Jacobian matrices (Grisetti et al., 2011).

3.3.3 Point-line factor graph construction

As shown in Figure 3.5, the frontend will publish a keyframe message if a new keyframe is determined by the relative motion from the previous keyframe. For example, if the relative translation exceeds 0.1 m, or the relative rotation angle exceeds 0.2 rad, it is argued that the camera has moved enough, and the feature map needs to be updated by a new keyframe.

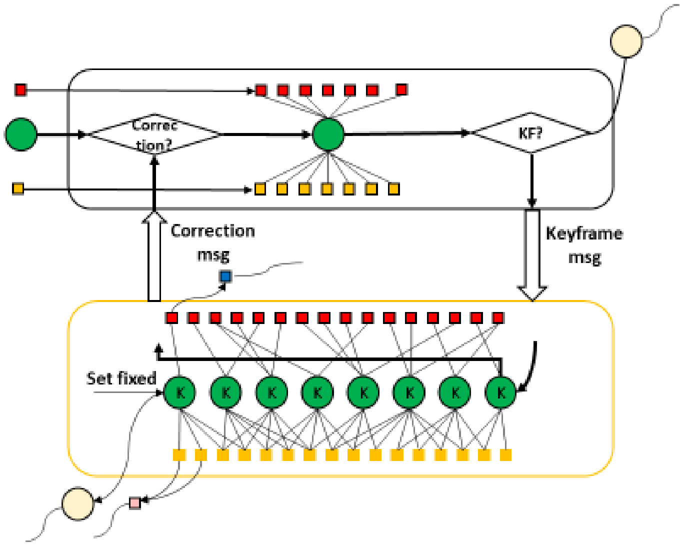


Figure 3. 5: Data communication between tracking thread and local mapping thread, and the novel factor graph based on point and line features.

The keyframe message contains the camera pose, and point and line associations attached to the current frame. If the keyframe message arrives in the local mapping thread, it will delete the oldest keyframe and add the new keyframe to the sliding-window framework to fix the keyframe number. As well, the features that are observed only by the oldest keyframe will be deleted accordingly.

A new factor graph in the downer part of Figure 3.5 is constructed using all the point and line reprojection errors in the sliding window. It consists of more constraints than previous line-based methods owing to the comprehensive 3D-3D and 3D-2D line correspondences, which is derived as

$$\sum_k \sum_i \rho\left(\left\|\boldsymbol{r}_{ik}^{2p}\right\|_{\boldsymbol{\Sigma}_{ik}^{2p}}^2\right) + \sum_k \sum_j \rho\left(\left\|\boldsymbol{r}_{jk}^{3l}\right\|_{\boldsymbol{\Sigma}_{jk}^{3l}}^2\right) + \sum_k \sum_j \rho\left(\left\|\boldsymbol{r}_{jk}^{2l}\right\|_{\boldsymbol{\Sigma}_{jk}^{2l}}^2\right) \qquad (3\text{-}25)$$

where *k*, *i*, and *j* are the indexes of the keyframe poses, points, and lines, respectively. The oldest keyframe pose is set fixed, and all the other keyframe poses and features (i.e., ${}_{w}^{k}\boldsymbol{T}$, $\boldsymbol{P}_{iw}$ and $\boldsymbol{\psi}_{jw}$) will be refined by the iterative Gauss-Newton method in g2o. Finally, the local mapping thread will publish a correction message to the frontend. The pose of the current frame and the related features will be updated accordingly.

3.3.4 Loop closing

The keyframe message is also sent to the loop closing thread, which has three parts, loop closure detection, loop closure verification, and pose graph optimization.

3.3.4.1 Loop closure detection

DboW2 is applied to detect loop candidates, which is a bag-of-word approach (Gálvez-López & Tardos, 2012). DboW2 has been widely applied for loop detection and shows the advantages in consideration of speed and accuracy (Mur-Artal & Tardós, 2017; Qin et al., 2018).

ORB descriptors are extracted from the new keyframe and transferred to a word vector by DboW2. The loop candidate is determined by comparing the similarity score between the word vector of the new keyframe and the previous keyframe. To ensure the detection speed, accuracy, and precision of the loop candidate keyframe, four conditions are set: (a) The difference between the indexes of the new keyframe and the candidate keyframe exceeds 100. Therefore, a keyframe close to the new keyframe will not be selected; (b) The candidate keyframe has the highest score; (c) The highest score should exceed 0.15. Otherwise, it is argued that the similarities between the two keyframes are insufficient. (d) The scores of continuous three keyframes before the candidate keyframe exceed 0.12. Thus, an isolated keyframe with the highest score will be rejected, which may be caused by perceptual aliasing.

3.3.4.2 Loop closure verification

Wrong loop closure may occur due to perceptual aliasing, especially when the surveying environment contains a similar texture. Therefore, the loop candidate should be verified by a geometry constraint. To build correspondences between the new keyframe and the candidate frame, FLANN is applied to associate their ORB descriptors. The wrong matches are ruled out by cross-checking and ratio-test first. Then the relative motion between the new keyframe and the loop candidate is calculated by RANSAC PnP. The loop candidate will be rejected if insufficient inlier matches are found, or the relative motion exceeds a threshold.

3.3.4.3 Pose graph optimization

Pose graph optimization will be performed to correct the loop closure if the loop candidate is verified by the geometry constraint. For the pose graph optimization, the vertexes are the keyframe poses, and the edges are the transformation matrices between the adjacent keyframes and those between the loop keyframes

$$\boldsymbol{r}^{m,n} = \log\left({}^{km}_{w}\boldsymbol{T}^{-1} * {}^{km}_{kn}\boldsymbol{T} * {}^{kn}_{w}\boldsymbol{T}\right)^{\vee} \tag{3-26}$$

where *km* and *kn* are the indexes of the two keyframes associated with the edges.

The cost function of pose graph optimization is built as

$$\sum\nolimits_{a} \rho\left(\|\boldsymbol{r}^{a,a+1}\|^{2}_{\boldsymbol{\Sigma}^{a,a+1}}\right) + \sum\nolimits_{l} \rho\left(\|\boldsymbol{r}^{l1,l2}\|^{2}_{\boldsymbol{\Sigma}^{l1,l2}}\right) \tag{3-27}$$

where $\boldsymbol{r}^{a,a+1}$ is the transformation error between the adjacent keyframes and $\boldsymbol{r}^{l1,l2}$ is the error between the loop keyframes. The loop closure is corrected by the iterative Gauss-Newton method in g2o, and the keyframe poses are refined accordingly.

3.4 Experiments and results

In this section, the performance of the proposed system is evaluated by TUM RGB-D datasets and real-world experiments and compared with SOTA methods (Sturm et al., 2012). All the experiments of the proposed system are carried out on a standard laptop (CPU: Core i5-5200U; RAM 8G).

3.4.1 Evaluation metrics

The absolute pose error (APE) is used to reflect the drift between the ground truth trajectory and the estimated trajectory. The root mean square error (RMSE) of APE is applied to evaluate the localization accuracy of the proposed system. The definitions of APE and RMSE are shown below

$$\mathbf{e}_i = trans\left({}_{gt}^{i}\boldsymbol{T}\right) - \; trans\left({}_{est}^{i}\boldsymbol{T}\,{}_{gt}^{est}\boldsymbol{T}\right) \qquad (3\text{-}28)$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\mathbf{e}_i^T\mathbf{e}_i}{n}} \qquad (3\text{-}29)$$

where ${}_{gt}^{i}\boldsymbol{T}$ is the ground truth for camera pose, ${}_{est}^{i}\boldsymbol{T}$ is the estimated pose from the SLAM system, and ${}_{gt}^{est}\boldsymbol{T}$ is the transformation between two trajectories by Umeyama alignment (Umeyama, 1991), *trans* represents the translation part of the pose.

The relative improvement of method A over method B is defined as

$$1 - \frac{\mathrm{RMSE}_A}{\mathrm{RMSE}_B} \qquad (3\text{-}30)$$

Similarly, to evaluate the mapping performance of the proposed system, the RMSE of the point-to-point distance (PTPD) is computed. PTPD is the distance between a point on the outputted 3D model and its corresponding point on the ground truth model after aligning two models by ICP.

3.4.2 TUM RGB-D datasets

TUM RGB-D datasets consist of sequences recorded with a Microsoft Kinect RGB-D camera in a variety of scenes. The frequency of the datasets is 30 frames per second (FPS), and the resolution is 640×480. The ground truth trajectory is recorded with a high-accuracy motion capture system with 100 Hz. Ten sequences are selected for trajectory evaluation. fr1_desk, fr1_floor, fr2_desk and fr3_long_office are common indoor scenes with texture and structure, fr3_nstr_tex_far and fr3_nstr_tex_near lack structure, fr3_str_ntex_far and fr3_str_ntex_near lack texture, and fr3_str_tex_far and fr3_str_tex_near contain highly discriminative texture.

The performance of the proposed system is compared with SOTA systems, i.e., ORB-SLAM2, DVO-SLAM, LSD-SLAM, DSO, PL-SLAM and Canny-VO (Engel et al., 2017; Engel et al., 2014; Kerl et al., 2013; Mur-Artal & Tardós, 2017; Pumarola et al., 2017; Y. Zhou et al., 2018). PL-SLAM (Pumarola et al., 2017) is evaluated using the implementation from https://github.com/HarborC/PL-SLAM, as its original code is not open-sourced. The scales of trajectories from LSD-SLAM, DSO, and PL-SLAM (Engel et al., 2017; Engel et al., 2014; Pumarola et al., 2017) are corrected by aligning to the ground truth trajectories. Table 3.1 shows the comparison results of APE RMSE, where “-” represents tracking failure. The smallest values are bolded and indicate the best accuracy.

Table 3. 1: Comparison of APE RMSE (cm) on TUM RGB-D datasets.

Sequence	**Length (m)**	**Proposed**	**ORB-SLAM2**	**DVO-SLAM**	**LSD-SLAM**	**DSO**	**PL-SLAM**	**Canny-VO**
fr1_desk	9.3	4.6	**2.1**	2.4	10.7	-	3.0	4.4
fr1_floor	12.6	3.2	6.1	10.2	38.1	5.5	3.0	**2.1**
fr2_desk	18.9	4.5	1.7	1.7	4.5	-	**1.4**	3.7
fr3_long_office	21.5	6.5	4.1	**3.5**	38.5	14.4	-	8.5
fr3_nstr_tex_far	4.3	7.0	5.6	2.8	18.3	4.8	-	**2.6**
fr3_nstr_tex_near	13.5	**3.3**	3.5	7.3	7.5	3.6	3.5	9.0
fr3_str_ntex_far	4.4	9.0	-	3.9	14.6	18.4	-	**3.1**
fr3_str_ntex_near	3.8	3.7	-	**2.1**	-	-	-	-
fr3_str_tex_far	5.9	1.3	1.3	3.9	8.0	7.9	**0.9**	1.3
fr3_str_tex_near	5.1	**1.2**	**1.4**	4.1	-	24.1	2.6	2.5

In Table 3.1, most of the systems can yield high accuracy (RMSE/Length < 1%) in most of the selected sequences. ORB-SLAM2 yields the best accuracy in 1 sequence out of 10. The proposed system, DVO-SLAM, and PL-SLAM yield the best accuracy in two sequences, and Canny-VO achieves the highest accuracy in three sequences.

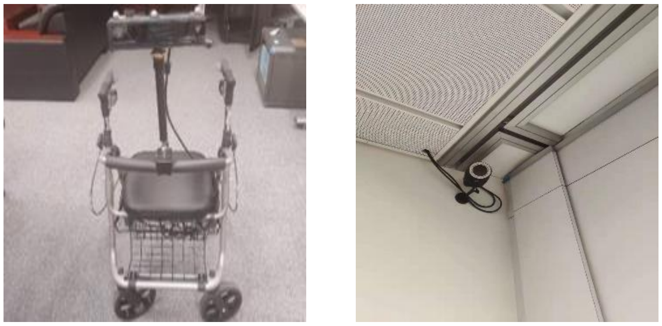
ORB-SLAM2 fails in fr3_str_ntex_far and fr3_str_ntex_near with low texture as it is highly dependent on the point features. The performance of LSD-SLAM will degrade if the camera is too close to walls or floors, i.e., in fr3_str_ntex_near and fr3_str_tex_near. DSO delivers much worse results than its original paper (Engel et al., 2017) because the original datasets prepare photometric calibration files while TUM RGB-D datasets do not. PL-SLAM is built based on the monocular version of ORB-SLAM2. It can outperform ORB-SLAM2 in some sequences owing to the fusion of the line features, but four sequences are not successfully tracked. Canny-VO fails in fr3_str_ntex_near due to the ambiguous structure.

The proposed system and DVO-SLAM success in all the sequences. DVO-SLAM is a direct SLAM system exploiting the photometric and depth information from all the pixels instead of partial point features, so it can avoid tracking failure in low textured scenes. On the other hand, the continuity of the proposed system is from the fusion of both 3D and 2D line features. Therefore, it is concluded that compared with SOTA works, the proposed system yields a similar level of accuracy and high continuity in a variety of scenes.

3.4.3 Lab room experiment

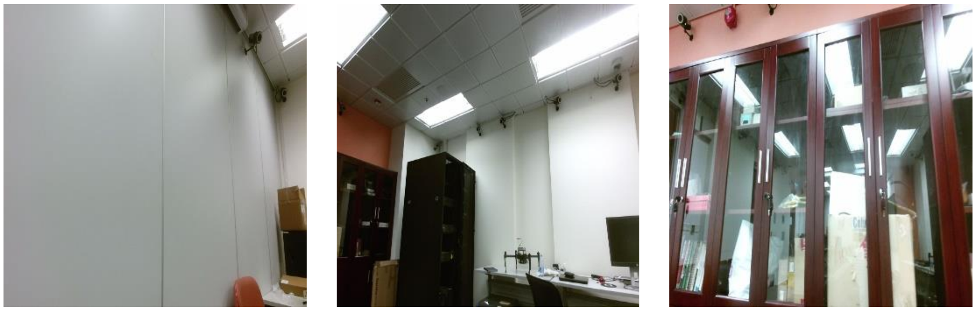
This experiment is conducted in a laboratory room with a Kinect V2 camera mounted on a wheelchair. Qualisys, the motion capture system, is installed in the room as shown in Figure 3.6, which provides the ground truth of the trajectory (Qualisys, 2008). The sequence collected by Kinect V2 covers some challenging scenes, i.e., low texture, illumination variation, and glass window as shown in

Figure 3.7. The main difficulty of the sequence is that the wheelchair will cross the low textured wall.



(a) *(b)*

Figure 3. 6: Experiment device. (a) Kinect v2 is mounted on a wheelchair. (b) Qualisys, the motion capture system.



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

Figure 3. 7: Challenging scenes. (a) Low texture; (b) Illumination variation; (c) Glass window.

To further verify the improvement of the localization accuracy by the fusion of comprehensive line features, five evaluations of the experiment sequence is presented in Table 3.2: (a) ORB-SLAM2; (b) point; (c) point + 3D line; (d) point + 2D line and (e) the proposed system. In Evaluations (b)–(d), some functions in the proposed system are disabled, so it runs with point feature, point + 3D line feature, and point + 2D line feature, respectively. The RMSE of the APE is used to evaluate the localization accuracy again. The smallest values are bolded and indicate the best accuracy.

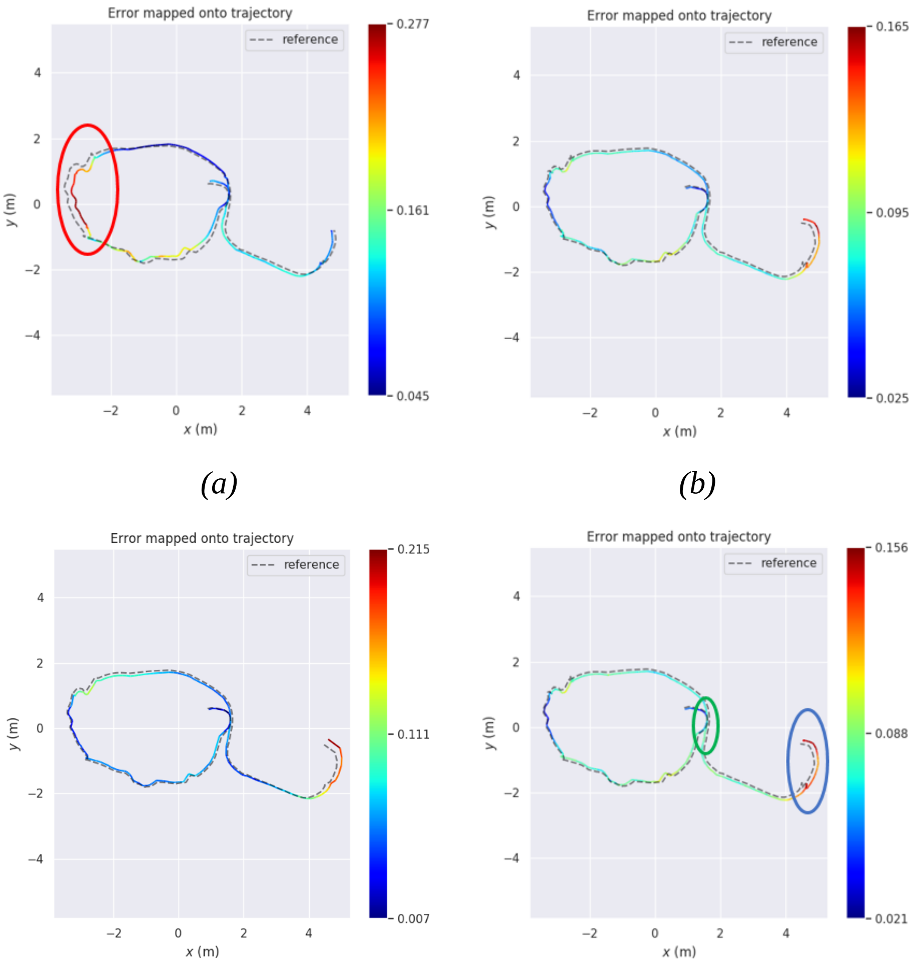
Table 3. 2: Comparison of APE RMSE (cm) on the room sequence.

Sequence	Length (m)	ORB-SLAM2	Point	Point + 3D Line	Point + 2D Line	Proposed
room	28.4	-	14.9	9.3	9.7	**7.2**

The results in Table 3.2 indicate the benefits of the proposed system. ORB-SLAM2 loses tracking when crossing the low textured wall, while other evaluations succeed. Considering the decrease of RMSE, the proposed system yields the best accuracy and can improve the accuracy of Evaluations (b)–(d) by 7.7 cm, 2.1 cm, and 2.5 cm, respectively, and the relative improvements are 51.6%, 22.5%, and 25.8%, respectively.

The trajectories of Evaluations (b)–(e) are shown in Figure 3.8. The red circle in Figure 3.8(a) indicates the partial trajectory crossing the low textured wall. The green circle in Figure 3.8(d) indicates the closed loop of the trajectory. The blue circle in Figure 3.8(d) indicates the end of the trajectory.

Due to lacking reliable and sufficient point features, Evaluation (b) has the highest trajectory error when crossing the wall, which is still significant after loop closing. For Evaluations (c)–(e), with the help of line features, the trajectory errors crossing the wall are much smaller. The values of the colour bar in Figure 3.8 also indicate the accuracy improvement of the proposed system.



(a) *(b)*

(c) *(d)*

Figure 3. 8: Comparison of estimated trajectories with ground truth on room sequence. (a) Point; (b) Point + 3D Line; (c) Point + 2D Line; (d) Proposed.

Figure 3.9 shows the reconstruction models from ORB-SLAM2 and the proposed system. The red circle in Figure 3.9(a) indicates the low textured wall where ORB-SLAM2 loses tracking. Therefore, only the partial trajectory is outputted, and the partial model is reconstructed by ORB-SLAM2. On the other hand, the proposed system can generate a complete 3D model owing to the line fusion.

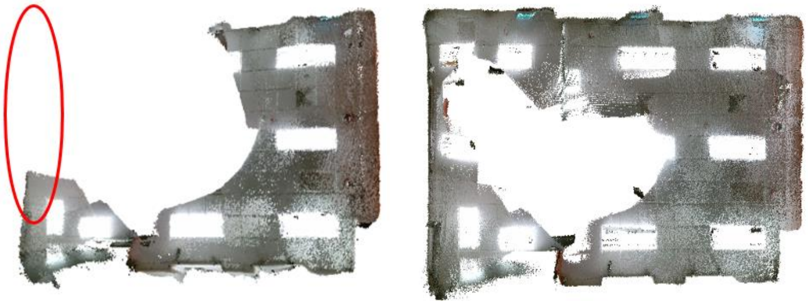

(a) (b)

Figure 3. 9: Reconstruction 3D models from simultaneous localization and mapping (SLAM) systems (a) ORB-SLAM2; (b) Proposed.

3.4.4 Corridor experiment

An experiment sequence is collected in a corridor with an iPad and a structure sensor. The ground truth of the corridor model is provided by NavVis M6, a high-precision Lidar-based indoor mapping system. Figure 3.10 shows the ground truth model.

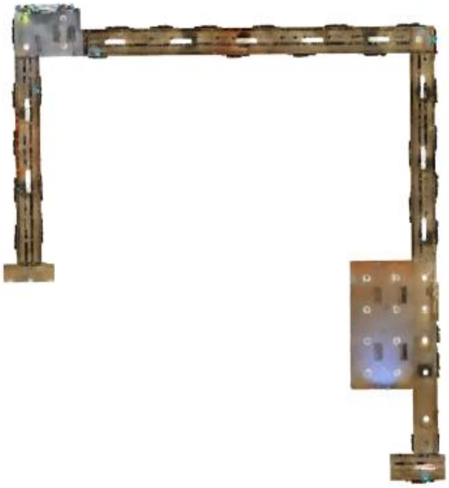


Figure 3. 10: Ground truth of the corridor provided by NavVis M6.

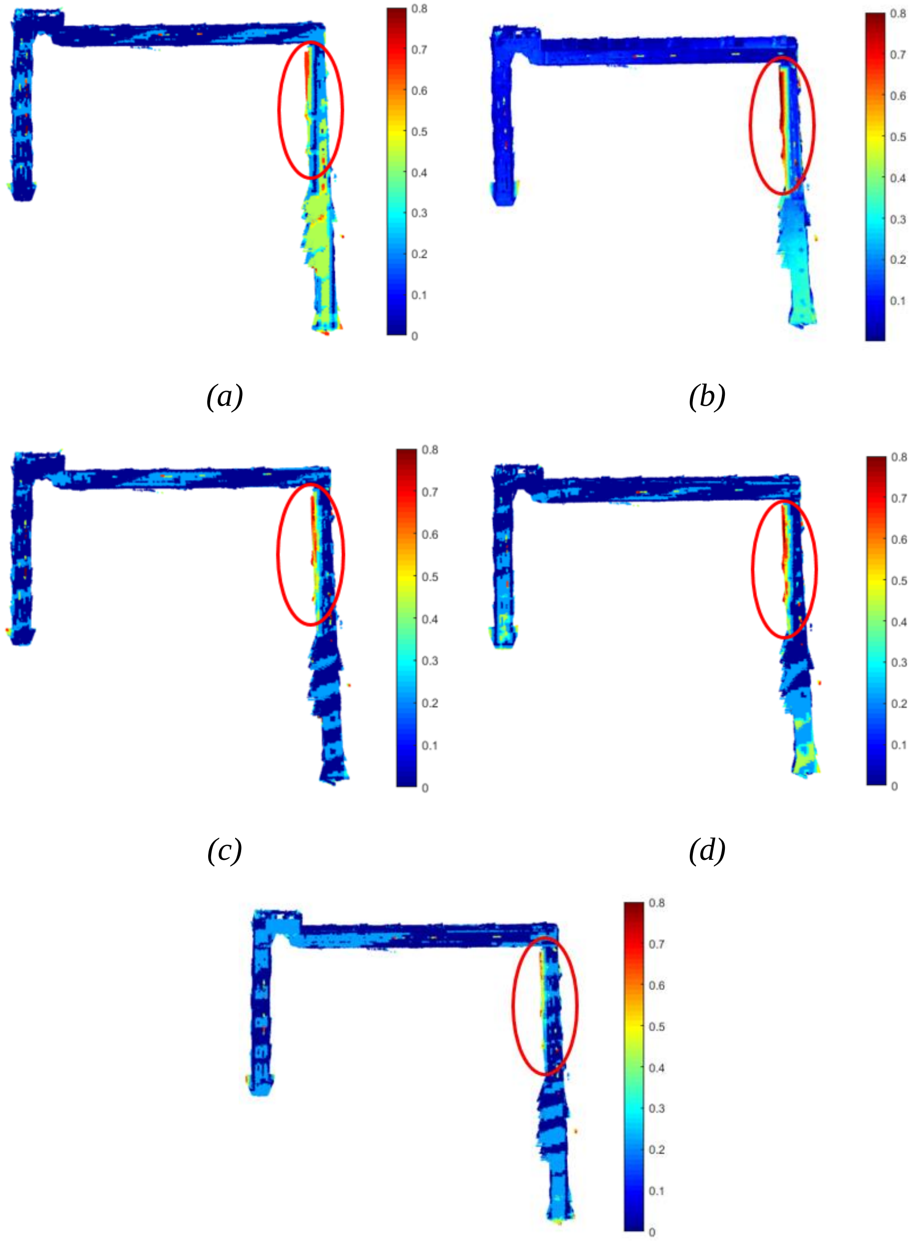
To further verify the improvement of mapping accuracy by the proposed system, five evaluations are presented in Table 3.3: (a) ORB-SLAM2; (b) point; (c) point + 3D line; (d) point + 2D line and (e) the proposed system. 3D corridor model is incrementally built based on the outputted camera pose and registered RGB-D frames. It is then compared with the ground truth. The reconstruction model and the ground truth are aligned by ICP and the RMSE of the PTPD are used to evaluate the reconstruction quality.

As shown in Table 3.3, the smallest value is indicated in bold. The proposed system yields the highest mapping quality among all the evaluations. It can improve the mapping accuracy of the point-based method by 28.9%, that of the method using 3D line features by 10.2%, and that of the method using 2D line features by 14.7%.

Table 3. 3: Comparison of RMSE (cm) of the point-to-point distance on the corridor sequence.

Sequence	Length (m)	ORB-SLAM2	Point	Point+3D Line	Point + 2D Line	Proposed
corridor	60.8	27.2	30.1	23.8	25.1	**21.4**

For a more intuitive comparison, the reconstruction models from five evaluations are shown in Figure 3.11. The colour of the model indicates the value of the point-to-point distance. The red circles indicate the biggest point-to-point distances after the second turning of the corridor. The five circles in Figure 3.11(a–e) are of the same size, and the areas with the biggest distances are selected to indicate the mapping quality of the five evaluations.



(a)

(b)

(c)

(d)

(e)

Figure 3. 11: Point-to-point distances between the reconstruction models and the ground truth. (a) ORB-SLAM; (b) Point; (c) Point + 3D Line; (d) Point + 2D Line; (e) Proposed.

Evaluation (e) has the smallest area, Evaluation (b) has the biggest area, while Evaluations (c)–(d) both improve the mapping accuracy and have smaller areas than

Evaluation (b). Though the area of Evaluation (a) from ORB-SLAM2 is the second smallest, its point-to-point distance at the end of the corridor is higher than the rest evaluations, which increases the value of the RMSE in Table 3. It is argued that fusing 3D or 2D line features can improve the mapping quality, and the combination of both 3D and 2D line features in the proposed system can yield the best mapping quality in the five evaluations.

3.4.5 Computation speed

The computation speed of the proposed system is then investigated using the sequence collected in the corridor. The time consumption by the loop closure module is not listed as it is highly dependent on the keyframe number. Similarly, the time cost of incremental mapping is also increasing with the frame number, so its processing time is not listed either. Table 3.4 shows the processing time of each part in the tracking thread and local mapping thread and compares the total time cost with ORB-SLAM2.

Table 3. 4: Processing time (ms) of each part of the proposed system.

Thread	Part	Proposed	ORB-SLAM2
Tracking	Optical Flow	7.5	
	Line Extraction and Matching	42.1	
	Robust Pose Solver	2.3	
	Shi-Tomasi Detection	7.6	
	IIR Filter	1.0	
	Total	60.4	32.1
Local Mapping	Factor Graph Optimization	115.3	186.3

Due to the time consumption by line extraction, the tracking frequency of the proposed system is only half of ORB-SLAM2. The proposed system sacrifices the computation speed for higher continuity in low textured scenes. The factor graph optimization of the proposed system is much faster than that of ORB-SLAM2 for two reasons:

(a) In the factor graph, the number of keyframes of the proposed system is lower than ORB-SLAM2. While the proposed system maintains a sliding window with 8 keyframes, ORB-SLAM2 builds a co-visibility map for every keyframe, where the connected keyframes can be more than 20.

(b) While ORB-SLAM2 needs to build a new optimizer using g2o for every keyframe, the proposed system maintains the same optimizer for every keyframe. Therefore, the optimizer of the proposed system can converge faster, and it also saves time to build the new optimizer.

The average number of correct feature matches is listed in Table 3.5. The fusion of 3D and 2D line features increases the number of measurements and contributes to the performance of the proposed system.

Table 3. 5: Average number of correct feature matches.

Types	**2D Point Feature Matches**	**2D Line Feature Matches**	**3D Line Feature Matches**
number	251.4	22.3	25.5

3.5 Summary

To improve the continuity of the RGB-D SLAM system in low textured scenes, a new method using point and line features is presented in this chapter. In summary,

(a) This chapter investigates the representation types for camera poses, point features and line features. Lie algebra is utilized for camera pose optimization, 3D position is adopted for point refinement, Plücker coordinate is selected for line projection, and orthonormal representation is used for line optimization.

(b) This chapter exploits both 3D-3D and 3D-2D line correspondences, and then builds a new cost function utilizing both 3D and 2D line reprojection errors, which can utilize more line constraints than the previous line-based methods.

(c) Experiment results of the TUM dataset show that the proposed system can achieve the same-level accuracy in rich textured scenes compared with SOTA methods and can improve their continuity in low textured scenes. The room experiment shows the improvements of the localization accuracy of the proposed system over the method using 3D line features and the method using 2D line features, which are 22.5% and 25.8, respectively. In the corridor experiment, the proposed system can improve the mapping accuracies of these two methods by 10.2% and 14.7%, respectively, owing to fusing 3D and 2D line features.

CHAPTER 4

RGB-D SLAM FUSING POINT AND PLANE FEATURES FOR LOW TEXTURED SCENES

As discussed in Section 2.3.2, point-based methods are efficient but may lose tracking in low textured scenes. High-level features (i.e., planes) are predominant in the indoor scenes and can be extracted from structures and objects (i.e., floor, wall, ceiling, desk, and cabinet). They are less affected by low textures and can provide more constraints for mobile platform tracking. Though various plane-based algorithms (Guo et al., 2019; Hosseinzadeh et al., 2017; Hsiao et al., 2017; Kaess, 2015; Taguchi et al., 2013; Yang et al., 2016; Zhang et al., 2019) have been proposed to fuse point and plane features, most of them just assign the experimental weights to plane measurements, which are non-optimal for pose estimation. Furthermore, the potential of structural regularity can be exploited from the plane features to enhance the tracking quality. By extracting MW axes from plane directions, MW-based methods can constrain the rotation of the mobile platform with high accuracy (Kim, Coltin, et al., 2018a, 2018b; Yanyan Li, Nikolas Brasch, et al., 2020a; Yanyan Li, Raza Yunus, et al., 2020; Zhou et al., 2016). However, they may fail in the operation scenes which are not strictly satisfying the MW assumption.

In this chapter, to improve the continuity of the RGB-D SLAM system and also avoid the disadvantages of the previous plane-based and MW-based methods, a new system fusing point and plane features is proposed. Its main contributions are as follows:

(a) The covariance of plane measurements in the spherical form is derived and a novel cost function is developed by fusing the point re-projection errors and the plane transformation errors based on the covariances. The new cost function can help to generate more accurate pose estimation results than the previous methods using the experimental weights.

(b) A new form for plane representation is developed, which utilizes the parallel and vertical relationships among planes and MW axes. It preserves the structural regularity in the operation scenes and does not rely on the MW assumption. To further improve the localization accuracy, a novel factor graph utilizing the new form is constructed to optimize the keyframe poses and the point and plane features.

(c) The proposed system is evaluated on TUM RGB-D datasets and shows superior performance compared with SOTA methods. In the lab room experiment, the proposed system can avoid tracking failure while the point-based system loses tracking due to a low textured wall. In addition, the proposed system can improve the localization accuracy by 23.6% using the analytical covariances, and enhance that by 27.6% using the new representation form. In the corridor experiment, the improvements of the mapping accuracies are 11.5% and 8.8%, respectively.

The remainder of this chapter is organized as follows. The system is overviewed in Section 4.1, the new representation form for plane features is introduced in Section 4.2, the methodology of fusing point and plane features is detailed in Section 4.3, the experiment results are compared in Section 4.4, and the conclusion is drawn in Section 4.5.

4.1 System overview

As shown in Figure 4.1, the proposed system is comprised of three threads running on the robot operation system (ROS) (Quigley et al., 2009): tracking, local mapping, and loop closing. It is built upon FLVIS (Chen et al., 2020) with additional support for plane feature processing. The tracking thread is called frontend and provides approximate estimation for camera pose. The combination of local mapping and loop closing threads is called backend and serves for pose refinement.

(a) Frontend: Firstly, point and plane features are detected from the current frame and matched with the previous frame. Secondly, the feature matches and the covariances of feature measurements are sent to the pose solver, which combines the pose-to-point and pose-to-plane constraints. Thirdly, the camera pose is outputted for the 3D modelling. Finally, if the current frame is determined as a new keyframe, its pose and the detected features will be added to the feature map. New planes will be inserted by the new representation form based on its parallel and vertical relationships with previous parent planes.

(b) Backend: In the local mapping thread, a new factor graph is built which represents the plane features using the new form and therefore encodes the parallel and vertical constraints in the operation scenes. Bundle adjustment is performed based on the new factor graph, which updates the keyframe poses, point and plane features simultaneously. The loop closing thread consists of three parts: loop closure detection, loop closure verification, and pose graph optimization. If a loop candidate is found by the bag-of-word approach (Gálvez-López & Tardos, 2012) and verified by the geometry check, pose graph optimization will be applied to correct the keyframe poses.

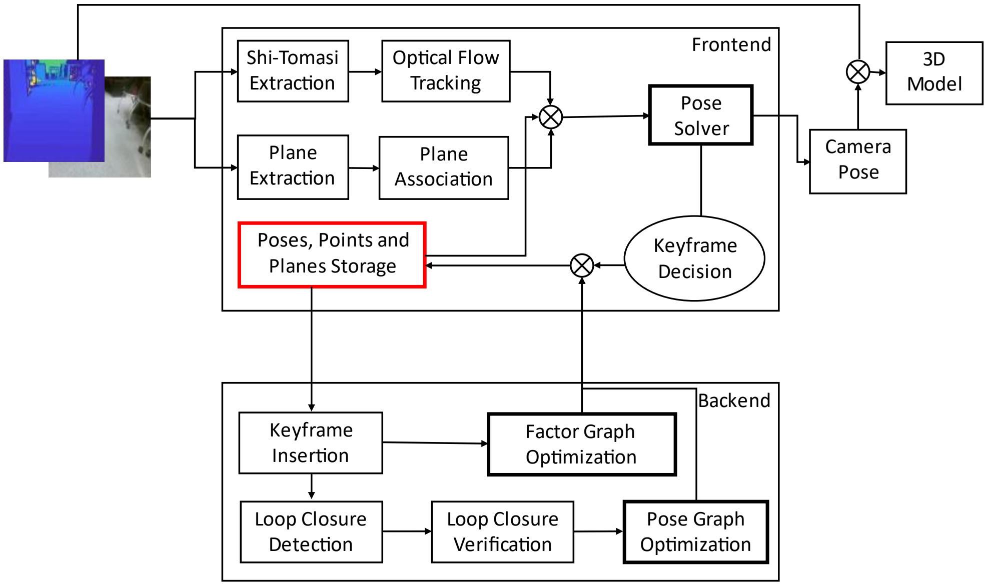


Figure 4. 1: Overview of the proposed system fusing point and plane features.

4.2 A new representation form for plane feature

The representation of the camera pose and the point feature is similar to Chapter 3 and detailed in Section 3.1. Four representation types for plane features are applied in the proposed system: (a) the Hessian form; (b) the spherical form (Yang & Huang, 2018); (c) the inverse depth form (Tang et al., 2011); and (d) the new form using parallel and vertical relationship among planes.

A plane *j* in the world frame $\{w\}$ can be represented by the Hessian form $\boldsymbol{\pi}_{jw} = [\boldsymbol{n}_{jw}^T, d_{jw}]^T$, where $\boldsymbol{n}_{jw} = [nx_{jw}, ny_{jw}, nz_{jw}]^T$ is the unit normal vector of the plane *j*, and d_{jw} is the distance from the origin to the plane. The point *i* on the plane *j* should satisfy

$$\boldsymbol{n}_{jw}^T \boldsymbol{P}_{iw} + d_{jw} = 0 \tag{4.1}$$

$\boldsymbol{\pi}_{jw}$ can be transformed to $\boldsymbol{\pi}_{jc}$ in the camera frame by

$$\boldsymbol{\pi}_{jc} = {}_{w}^{c}\boldsymbol{T}^{-T} \boldsymbol{\pi}_{jw} \tag{4.2}$$

The Hessian form is over-parametrized as it has four parameters while a 3D plane has three DoFs. If we consider the normal vector as a point on a unit sphere and transfer it to two angles, plane j can be represented by the spherical form with minimal parameters

$$\boldsymbol{\tau}_{jw} = q(\boldsymbol{\pi}_{jw}) = \left[\phi_{jw} = \arctan\frac{ny_{jw}}{nx_{jw}}, \quad \psi_{jw} = \arcsin nz_{jw}, \quad d_{jw}\right]^T \tag{4.3}$$

The inverse depth form $\boldsymbol{\mu}_{jw} = \boldsymbol{n}_{jw}/d_{jw}$ is also compact, and later used for plane fitting and covariance estimation. In the proposed system, the planes in the feature map are divided into parent planes and child planes by the new type. MW axes can be easily extracted using the new form. The pipeline of representing a new plane is shown in Figure 4.2.

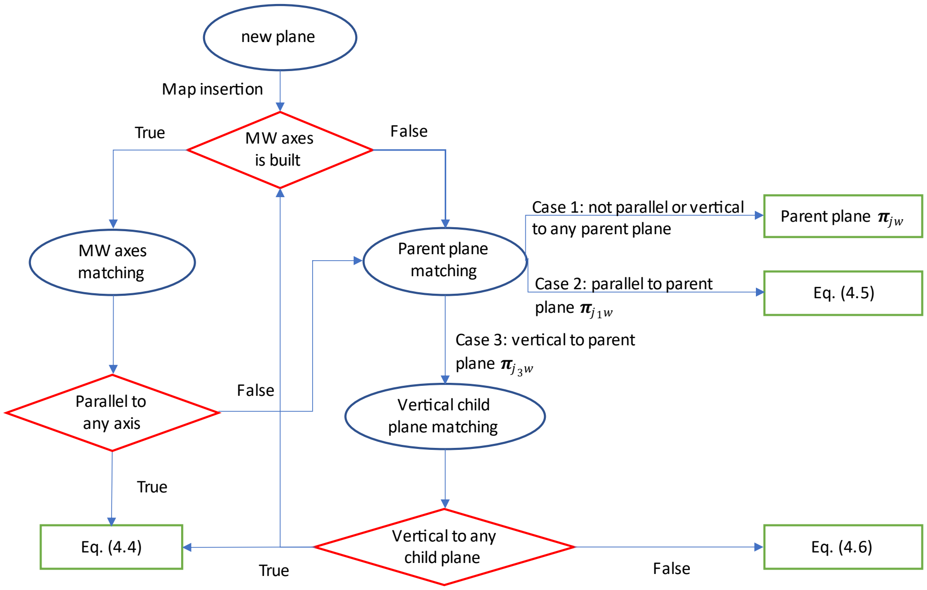

new plane
Map insertion
True
MW axes is built
False
MW axes matching
Parent plane matching
Case 1: not parallel or vertical to any parent plane
Parent plane $\boldsymbol{\pi}_{jw}$
Case 2: parallel to parent plane $\boldsymbol{\pi}_{j_1w}$
Eq. (4.5)
Case 3: vertical to parent plane $\boldsymbol{\pi}_{j_3w}$
False
Parallel to any axis
Vertical child plane matching
True
Eq. (4.4)
Vertical to any child plane
Eq. (4.6)
True
False

Figure 4. 2: Pipeline of representing a new plane with the new type.

(a) If the MW axes are built, the new plane is first matched with three MW axes $\mathrm{MW} = [\boldsymbol{r}_x, \boldsymbol{r}_y, \boldsymbol{r}_z]$. If it is parallel to any MW axis $\boldsymbol{r}_x$, the plane j is represented by the direction of the MW axis and a distance:

$$\boldsymbol{\pi}_{jw} = f_{MW}(\mathrm{MW}, d_{jw}) = [\boldsymbol{r}_x^T, d_{jw}]^T \tag{4.4}$$

(b) If MW is not built or the new plane is not parallel to any MW axis, the new plane will be matched with the parent planes in the feature map. If it is not parallel or vertical to any parent planes, it is marked as a new parent plane $\boldsymbol{\pi}_{jw}$.

(c) If it is parallel to a parent plane $\boldsymbol{\pi}_{j_1 w}$, we will represent it as a parallel child plane of $\boldsymbol{\pi}_{j_1 w}$.

$$\boldsymbol{\pi}_{jw} = f_p(\boldsymbol{\pi}_{j_1 w}, d_{jw}) = [\boldsymbol{n}_{j1w}^T, d_{jw}]^T \tag{4.5}$$

(d) If it is vertical to a parent plane $\boldsymbol{\pi}_{j_2 w}$, we will further match it with other child planes of $\boldsymbol{\pi}_{j_2 w}$.

(e) If it is not vertical to any other planes that vertical to the parent plane, we will represent it as a vertical child plane of $\boldsymbol{\pi}_{j_2 w}$

$$\boldsymbol{\pi}_{jw} = f_v(\boldsymbol{\pi}_{j_2 w}, \theta_{jw}, d_{jw}) = \begin{bmatrix} \boldsymbol{R}_{j2}^T \begin{bmatrix} 0 \\ \cos\theta_{jw} \\ \sin\theta_{jw} \end{bmatrix} \\ d_{jw} \end{bmatrix} \tag{4.6}$$

$$\boldsymbol{R}_{j_2} = \begin{bmatrix} \cos\phi_{j2w} & -\sin\phi_{j2w} & 0 \\ \sin\phi_{j2w} & \cos\phi_{j2w} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\psi_{j2w} & 0 & \sin\psi_{j2w} \\ 0 & 1 & 0 \\ -\sin\psi_{j2w} & 0 & \cos\psi_{j2w} \end{bmatrix} \tag{4.7}$$

where R_{j_2} is a rotation matrix that leads to the normal of the parent plane to $[1, 0, 0]^T$ and the normal of $\boldsymbol{\pi}_{jw}$ to $[0, \cos\theta_{jw}, \sin\theta_{jw}]^T$ as shown in Figure 4.3.

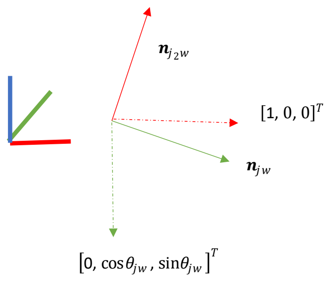


Figure 4. 3: Directions of parent plane $\boldsymbol{\pi}_{\boldsymbol{j_2 w}}$ *and child plane* $\boldsymbol{\pi}_{\boldsymbol{jw}}$ *are rotated by* $\boldsymbol{R}_{\boldsymbol{j_2}}$*.*

(f) If it is vertical to a vertical child plane $\boldsymbol{\pi}_{j_3 w}$, the MW axes can be built easily by the plane normal using singular value decomposition (SVD)

$$[\boldsymbol{r}_x \quad \boldsymbol{r}_y \quad \boldsymbol{r}_z] = \boldsymbol{U}\boldsymbol{V}^T \tag{4.8}$$

where $[\boldsymbol{U} \quad \boldsymbol{D} \quad \boldsymbol{V}] = SVD[\lambda_x \boldsymbol{n}_{j2w} \quad \lambda_y \boldsymbol{n}_{j3w} \quad \lambda_z \boldsymbol{n}_{jw}]$ and λ_x, λ_y, and λ_z are the numbers of the points on the corresponding planes.

In this chapter, the Hessian form is used for the plane transformation, and it is transferred to the Spherical form during the pose optimization. The inverse depth form is used to fit the plane and propagate to the plane covariance in the Spherical form. The new form is used for the factor graph construction. Compared with other representation types for the plane feature, the new form has two advantages: (a) it estimates fewer parameters; (b) it encodes the parallel and vertical relationships in the operation scenes.

4.3 Fusing point and plane features

The full pipeline of fusing point and plane features is introduced in this section, which consists of plane extraction and matching, robust pose solver, factor graph construction, and loop closure. Loop closure and point feature processing are handled similarly with Chapter 3, so the details are not presented again.

4.3.1 Plane extraction and matching

Point feature is extracted using the improved Shi-Tomasi detector, and matched by the optical flow tracking, as detailed in Chapter 3.4.1. The plane feature is extracted by a fast algorithm of agglomerative hierarchical clustering, which can be implemented on a CPU in real time (Feng et al., 2014). Firstly, an organized point cloud is extracted from the depth image and can be divided into non-overlapping groups. Then a graph is built whose nodes and edges are the group members and their neighbors. Finally, an agglomerative hierarchical clustering is performed to merge the nodes belonging to the same plane, as shown in Figure 4.4(b).

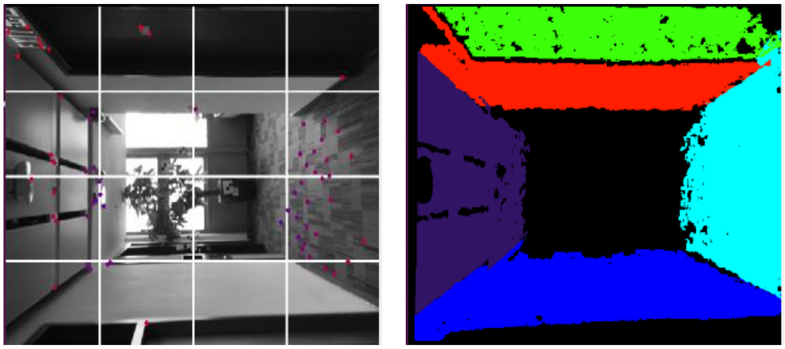
The plane feature is matched based on its Hessian form in the world frame. The initial pose of the current frame ${}_{w}^{c}\boldsymbol{T}$ is estimated by a consistent-velocity motion model, and the Hessian form of the plane feature in the current frame $\boldsymbol{\pi}_{jc}$ is fitted using the segmented points from Figure 4.3. Then the Hessian form of the plane j in the world frame is predicted by

$$\boldsymbol{\pi}_{jw}^{predict} = {}_{w}^{c}\boldsymbol{T}^{T}\boldsymbol{\pi}_{jc} \tag{4.9}$$

$\boldsymbol{\pi}_{jw}^{predict}$ is compared with the existing planes in the feature map. The associated plane is matched if it satisfies the following conditions:

$$\left\|\left(\boldsymbol{n}_{jw}^{predict}\right)^{T}\boldsymbol{n}_{jw}\right\|^{2} < n_{thre},\ \left\|d_{jw}^{predict} - d_{jw}\right\| < d_{thre} \tag{4.10}$$

where n_{thre} and d_{thre} are the thresholds for the plane association.



(a) *(b)*

Figure 4. 4: Point and plane features. (a) Improved Shi-Tomasi detector (Shi, 1994); (b) Fast plane extraction (Feng et al., 2014).

4.3.2 Robust pose solver utilizing point and plane features

In this part, the 2D point reprojection error and the 3D plane transformation error are combined to build a novel cost function based on their covariance. While the covariance of the 2D point reprojection error is similar to that in Chapter 3, the covariance propagation of the plane measurements using the spherical form is derived in this section.

4.3.2.1 Covariance propagation for the plane feature

It is assumed that the plane j is observed by the current frame, and its Hessian form in the camera frame and the world frame are π_{jc} and π_{jw}, respectively. The constraint between the camera pose and the plane feature can be represented by the 3D plane transformation error $\boldsymbol{r}_{jc}$ and its covariance $\boldsymbol{\Sigma}_{jc}$

$$\boldsymbol{r}_{jc} = q\left(\boldsymbol{\pi}_{jc}\right) - q\left({}_{w}^{c}\boldsymbol{T}^{-T}\boldsymbol{\pi}_{jw}\right) \tag{4.11}$$

$\boldsymbol{\Sigma}_{jc}$ can be computed by the forward propagation (Proença & Gao, 2018; Y. Yang et al., 2019) or using the experimental value (Kaess, 2015; Ma et al., 2016; Tang et al., 2011; Zhang et al., 2019). The main backward of the methods based on the forward propagation (Proença & Gao, 2018; Y. Yang et al., 2019) is that the Jacobians of the plane parameters with respect to points on the plane are required, which is time-consuming if the plane has a large number of points. Furthermore, the depth variances used in these methods (Proença & Gao, 2018; Y. Yang et al., 2019) are simply taken from the experimental function. More accurate depth variance can be obtained from plane fitting tests (Khoshelham & Elberink, 2012), but requires much more human labor. The method to derive the covariance of the plane feature in this chapter does not rely on the depth variance. Instead, it estimates the unbiased variance σ_{pf}^2 of the plane fitting errors, and then directly propagate it to compute $\boldsymbol{\Sigma}_{jc}$. It is more reasonable than using the experimental weights and more efficient that the methods based on the forwards propagation. The detail is introduced as below:

(a) Firstly, because Eq. (4.1) is nonlinear in terms of the Hessian form $\boldsymbol{\pi}_{jw} = \left[\boldsymbol{n}_{jw}^T, d_{jw}\right]^T$, which has an inside condition that $\left\|\boldsymbol{n}_{jw}^T\right\| = 1$, the inverse depth form is adopted $\boldsymbol{\mu}_{jw} = \boldsymbol{n}_{jw}/d_{jw} = (a, b, c)^T$ to rewrite Eq. (4.1).

$$\frac{nx_{jw}}{d_{jw}}x + \frac{ny_{jw}}{d_{jw}}y + \frac{nz_{jw}}{d_{jw}}z + 1 = 0 \tag{4.12}$$

(b) Secondly, if the plane *j* has N points, the matrix form of Eq. (4.12) can be written as

$$\boldsymbol{A}\boldsymbol{\mu}_{jw} + \boldsymbol{B} = 0 \tag{4.13}$$

where $\boldsymbol{A} = [\boldsymbol{P}_1 \quad \cdots \quad \boldsymbol{P}_N]^T$, and $\boldsymbol{B} = [(1 \quad \cdots \quad 1)]^T$. $\boldsymbol{P}_i$ represents the point on the plane.

(c) Thirdly, $\boldsymbol{\mu}_{jw}$ can be easily solved by

$$\boldsymbol{\mu}_{jw} = -(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{B} \tag{4.14}$$

(d) Fourthly, the covariance of $\boldsymbol{\mu}_{jw}$ is

$$\boldsymbol{\Sigma}_{\boldsymbol{\mu}_{jw}} = \sigma_{pf}^2(\boldsymbol{A}^T\boldsymbol{A})^{-1} \tag{4.15}$$

where $\sigma_{pf}^2 = \left(\boldsymbol{A}\boldsymbol{\mu}_{jw} + \boldsymbol{b}\ \right)^T\left(\boldsymbol{A}\boldsymbol{\mu}_{jw} + \boldsymbol{b}\right)/(N-3)$ is the variance of plane fitting errors.

(e) Finally, the covariance of $\boldsymbol{r}_{jc}$ is derived as

$$\boldsymbol{\Sigma}_{jc} = \boldsymbol{\Sigma}_{\boldsymbol{\tau}_{jw}} = J_{\boldsymbol{\tau},\boldsymbol{\mu}} \boldsymbol{\Sigma}_{\boldsymbol{\mu}_{jw}} J_{\boldsymbol{\tau},\boldsymbol{\mu}}^{T}, \boldsymbol{\tau}_{jw} = \left[\arctan\frac{\mathrm{b}}{\mathrm{a}}, \arcsin\frac{c}{\sqrt{a^2+b^2+c^2}}, \frac{1}{\sqrt{a^2+b^2+c^2}}\right]^{T} \quad (4.16)$$

$$J_{\boldsymbol{\tau},\boldsymbol{\mu}} = \frac{\delta_{\tau_{jw}}}{\delta_{\mu_{jw}}} = \begin{bmatrix} -\frac{b^2}{b^2+a^2} & \frac{a^2}{b^2+a^2} & 0 \\ \frac{ac}{(a^2+b^2+c^2)\sqrt{a^2+b^2}} & \frac{bc}{(a^2+b^2+c^2)\sqrt{a^2+b^2}} & \frac{\sqrt{a^2+b^2}}{a^2+b^2+c^2} \\ -\frac{a}{(a^2+b^2+c^2)^{\frac{3}{2}}} & -\frac{b}{(a^2+b^2+c^2)^{\frac{3}{2}}} & -\frac{c}{(a^2+b^2+c^2)^{\frac{3}{2}}} \end{bmatrix} \quad (4.17)$$

4.3.3.2 Novel cost function

Eq. (3.14), (3.20), (4.11), and (4.16) are combined to build a novel cost function below

$$\sum_i \rho\left(\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^{2}\right) + \sum_j \rho\left(\left\|\boldsymbol{r}_{jc}\right\|_{\boldsymbol{\Sigma}_{jc}}^{2}\right) \quad (4.18)$$

where *i* and *j* are the indexes of the point and plane feature matches of the current frame, respectively, ρ represents the Huber function, and $\boldsymbol{\Sigma}$ is the covariance matrix associated with the feature measurement. Compared with the cost function of the point-based method $\sum_i \rho\left(\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^{2}\right)$, Eq. (4.18) provides additional plane-based constraints which are reliable in low textured scenes. Furthermore, the covariances of the plane feature $\boldsymbol{\Sigma}_{jc}$ is derived by covariance propagation and plane fitting, which is more reasonable than the previous plane-based methods using the experimental weights.

The iterative Gauss-Newton method in g2o (Grisetti et al., 2011) is applied to minimize Eq. (4.18) and solve ${}_{w}^{c}\boldsymbol{\xi}$. The Mahalanobis distance test is also employed after every four iterations.

$$\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^{2} < \chi_{\alpha,2}, \left\|\boldsymbol{r}_{jc}\right\|_{\boldsymbol{\Sigma}_{jc}}^{2} < \chi_{\alpha,3} \quad (4.19)$$

where α is the threshold of Chi-square distribution, 2 and 3 are the DoFs of point and plane measurements, and $\|\boldsymbol{r}_{ic}\|_{\boldsymbol{\Sigma}_{ic}}^{2}$ and $\left\|\boldsymbol{r}_{jc}\right\|_{\boldsymbol{\Sigma}_{jc}}^{2}$ are the Mahalanobis distances.

The feature matches with large Mahalanobis distance will be marked as the outliers and excluded in the next iteration for robustness.

4.3.3 Point-plane factor graph construction

As shown in Figure 4.5, the tracking thread will publish a new keyframe message if its relative motion to the previous keyframe exceeds a threshold or insufficient feature matches are found. The new keyframe and its features are contained in the keyframe message and sent to the local mapping thread for further refinement.

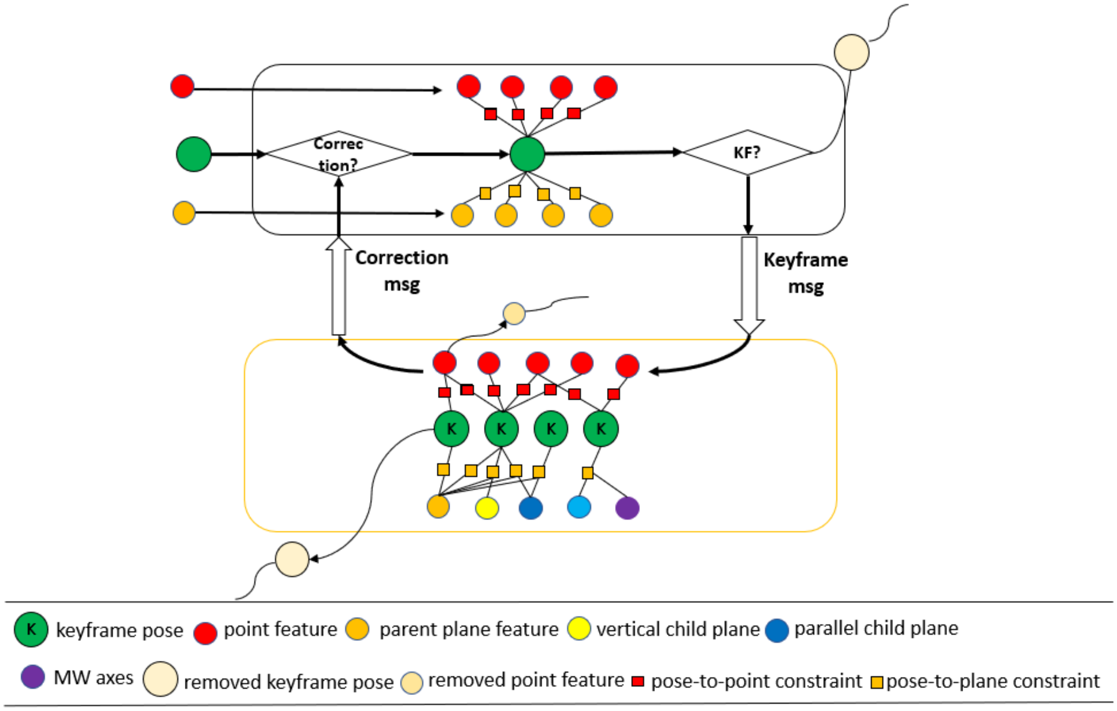


Figure 4. 5: Data communication between tracking and local mapping threads and the novel factor graph based on point and plane features.

In the tracking thread, the plane features are simply represented by the spherical form because they are considered fixed during pose optimization. However, in the local mapping thread, the plane features are adjusted together with the keyframe poses and the point features during bundle adjustment. A novel factor graph is constructed in Figure 4.5, where the plane features are inserted and represented by the new form. The cost function for the novel factor graph is built by fusing Eq. (4.4-4.6), (3.14), (3.20), (4.11), and (4.16)

$$\sum_k \sum_i \rho\left(\left\|\boldsymbol{r}_{ik}^{2p}\right\|_{\boldsymbol{\Sigma}_{ik}^{2p}}^{2}\right) + \sum_k \sum_j \rho\left(\left\|\boldsymbol{r}_{jk}\right\|_{\boldsymbol{\Sigma}_{jk}}^{2}\right) \tag{4.20}$$

where *k*, *i*, and *j* are the indexes of the keyframe, point, and plane, respectively. The plane transformation error is represented by $\boldsymbol{r}_{jk}$ and computed by Eq. (4.11), and the plane feature $\boldsymbol{\pi}_{jw}$ in Eq. (4.11) is represented by the new form using Eq. (4.4-4.6). The iterative Gauss-Newton method in g2o is again applied to solve Eq. (4.20). Then the local mapping thread will publish a correction message to update the camera poses, point, and plane features in the tracking thread.

In the novel factor graph, the pose-to-plane constraints are similarly built according to Eq. (4.11) and (4.16) when the plane feature *j* is observed by the keyframe k. However, the factors connected to these constraints are different based on Eq. (4.4-4.6) which apply the new form for plane representation:

(a) If *j* is a parent plane, then the constraint connects the keyframe *k* and the parent plane *j*.

(b) If *j* is a parallel child plane of the parent plane j_1, and represented by Eq. (4-5), then the constraint connects the keyframe *k*, the parent plane $j_{1,}$ and the child plane *j*.

(c) If *j* is a vertical child plane of the parent plane j_2, and represented by Eq. (4-6), then the constraint connects the keyframe *k*, the parent plane j_2 and the child plane *j*.

(d) If *j* is parallel to any MW axis and represented by Eq. (4-4), then the constraint connects the keyframe *k*, the plane *j*, and the MW axes. The MW axes are fixed during bundle adjustment.

Using the new form, the parent planes and child planes are connected in the new factor graph, which encodes parallel and vertical relationships among planes. The connections between the plane features and the MW axes also help to exploit more structural regularity. As shown in Figure 2.6, Plane 4-5 are not parallel to any of the MW axes and break the MW assumption. In the MW-based methods, MW axes are estimated from the surface normal vectors of all the points. Then the rotation matrix is obtained by aligning the MW axes of the current frame to that of the first frame. Because of the undesirable points on Plane 4-5, the accuracy of rotation estimation may be reduced. However, in the proposed system which adopts the new form for the plane representation, Planes 4-5 are represented by a vertical child plane using Eq. (4.6), while Planes 1-3 and 6 are represented by a plane parallel to the MW axis using Eq. (4.4). Therefore, the new form can exploit the structural regularity reasonably whether the scene satisfies the MW assumptions.

4.3.4 Loop closing

The pipeline of the loop closing module is similar to Section 3.3.4. The keyframe message is also received by the loop closing thread. The descriptors extracted on the new keyframe are encoded as a word vector by DBoW2. The loop candidate is detected by comparing the similarity scores between the word vectors of the keyframes. The geometry check is performed to verify the loop candidate. The relative motion between the new keyframe and the loop candidate frame is then computed, and the loop candidate will be accepted if: (a) the relative motion is small; and (b) sufficient feature matches are found. Finally, pose graph optimization is applied to correct the loop closure based on Eq. (3.26) and (3.27).

4.4 Experiments and results

In this section, experiments are conducted using TUM RGB-D dataset and self-collected datasets. The performance of the proposed system is evaluated and compared with other SOTA SLAM systems. Similar to Section 3.4.1, the RMSE of the APE and PTPD are adopted as evaluation matrices. The same room and corridor in Section 3.4.3 and 3.4.4 are selected as the operation scenes. All the experiments in this section run on a laptop with i5-5200U CPU and 8G RAM.

4.4.1 TUM RGB-D datasets

TUM RGB-D dataset is the standard dataset for evaluating the performance of SLAM systems. Six sequences with structural regularity are selected, including fr3_str_ntex_far, fr3_str_ntex_near, fr3_str_tex_far, fr3_str_tex_near, fr3_cabinet and fr3_large_cabinet.

The example images in the first column of Figure 4.6 indicate that fr3_str_ntex_far, fr3_str_ntex_near, fr3_cabinet, and fr3_large_cabinet contain low textures. The trajectories from the proposed system are compared with the ground truth in the second column of Figure 4.6. The small differences between the compared trajectories demonstrate the high accuracy of camera localization. The outputted 3D models are shown in the third column of Figure 4.6.

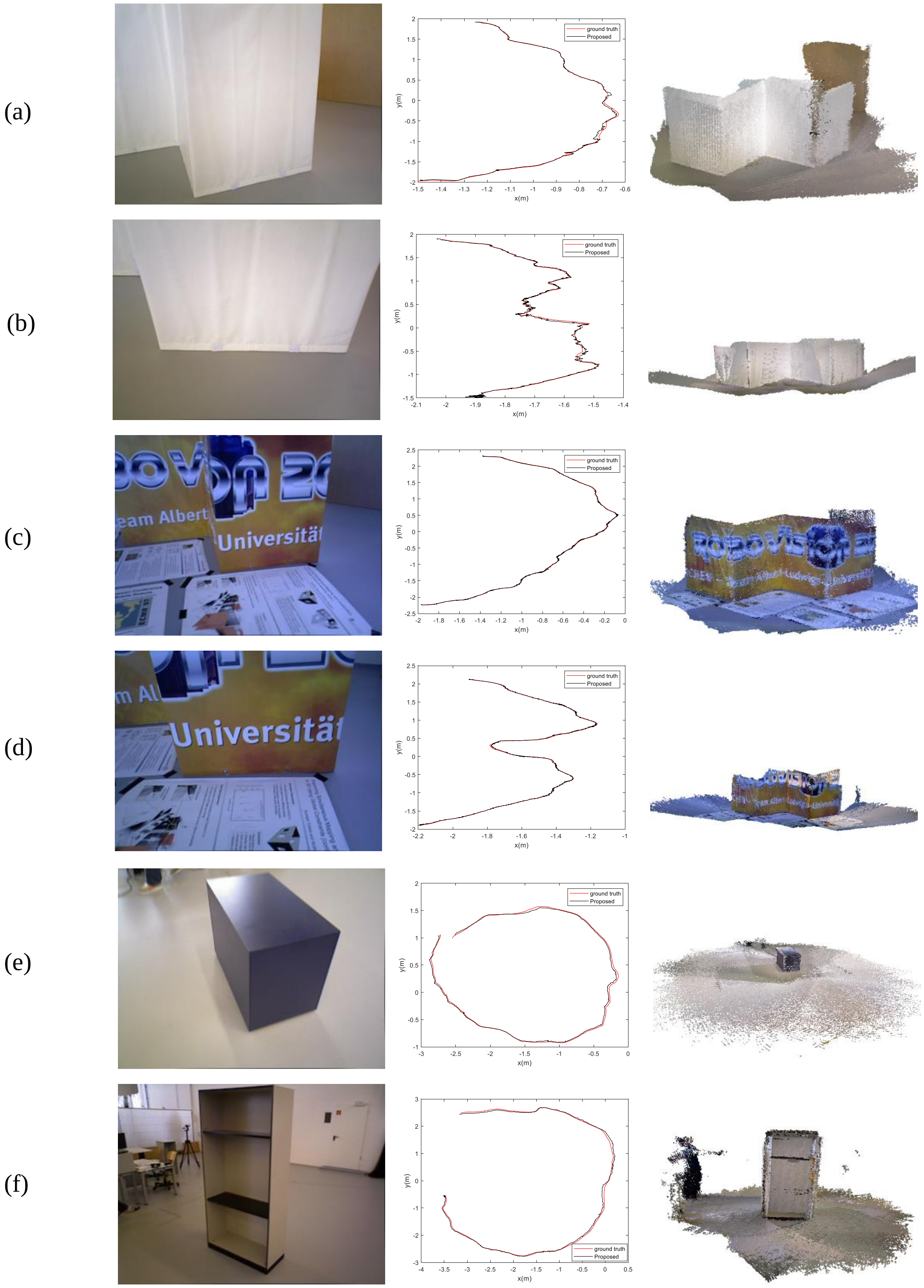


Figure 4. 6: Trajectories by the proposed system in six sequences:(a) fr3_str_ntex_far; (b) fr3_str_ntex_near; (c) fr3_str_tex_far; (d) fr3_str_tex_near; (e) fr3_cabinet; (f) fr3_large_cabinet.

The performance of the proposed system is then compared with other SOTA systems, i.e., ORB-SLAM2, DVO, LPVO, L-SLAM, and PS-SLAM (Kerl et al., 2013; Kim, Coltin, & Jin Kim, 2018; Kim, Coltin, et al., 2018b; Mur-Artal & Tardós, 2017; Zhang et al., 2019). The comparison results are shown in Table 1, where "-" represents the tracking failure and the bolded value means the best accuracy.

Table 4. 1: Comparison of APE RMSE (cm) on TUM RGB-D dataset.

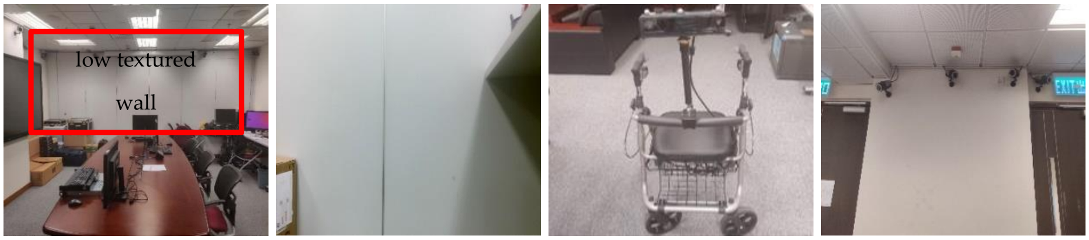
Sequence	Length (m)	Proposed	ORB-SLAM2	DVO-SLAM	LPVO	L-SLAM	PS-SLAM
fr3_str_ntex_far	4.4	**1.6**	-	3.9	7.5	14.1	2.0
fr3_str_ntex_near	3.8	1.5	-	2.1	8.0	6.6	**1.3**
fr3_str_tex_far	5.9	**1.1**	1.3	3.9	17.4	21.2	**1.1**
fr3_str_tex_near	5.1	1.1	1.5	4.1	11.5	15.6	**1.0**
fr3_cabinet	11.2	**4.1**	11.7	69.0	52.0	29.1	6.7
fr3_large_cabinet	20.7	**3.6**	-	97.9	27.9	14.0	7.9

ORB-SLAM2 is a famous feature point-based system. Due to the lack of point features, it fails in three low textured sequences: fr3_str_ntex_far, fr3_str_ntex_near, and fr3_large_cabinet. DVO-SLAM is a direct method that does not rely on the point features, so it can perform well in fr3_str_ntex_far and fr3_str_ntex_near despite of low textures. However, in fr3_cabinet and fr3_large_cabinet, its accuracy degrades due to insufficient gradient for the photometric error when the camera observes the floor. LPVO and L-SLAM are two MW-based methods. It can also avoid the tracking failure problem owing to the line features and the surface normal, which are utilized to estimate the MW frame and recover the rotation matrix. However, as indicated by the example images and the reconstructed models in Figure 4.6(e) and (f), fr3_cabinet and fr3_large_cabinet do not strictly satisfy the MW assumption which can lower the performance of the MW-based methods. PS-SLAM and the proposed system are plane-based methods. They do not strictly rely on the MW assumption, and the plane features can be fused to improve the continuity of the SLAM systems in low textured scenes.

Among all the above methods, PS-SLAM achieves the highest accuracy in three sequences owing to the additional constraints from the supposed planes, and the proposed system yields the best accuracy in four sequences owing to (a) the new form for plane representation and (b) the analytical covariance for the plane measurements in the spherical form.

4.4.3 Lab room experiment

The operation scene and device are shown in Figure 4.7. A Microsoft Kinect V2 camera is mounted on a wheelchair to capture the RGB-D sequence in a laboratory room. Qualisys motion capture system (Qualisys, 2006) is installed in the room to provide ground truth trajectories. As shown in Figure 4.7(b), the operation scene contains a low textured wall which is challenging for the feature point-based methods.



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

Figure 4. 7: Room experiment setup. (a) Operation scene; (b) Low textured wall; (c) Surveying device; (d) Qualisys motion capture system.

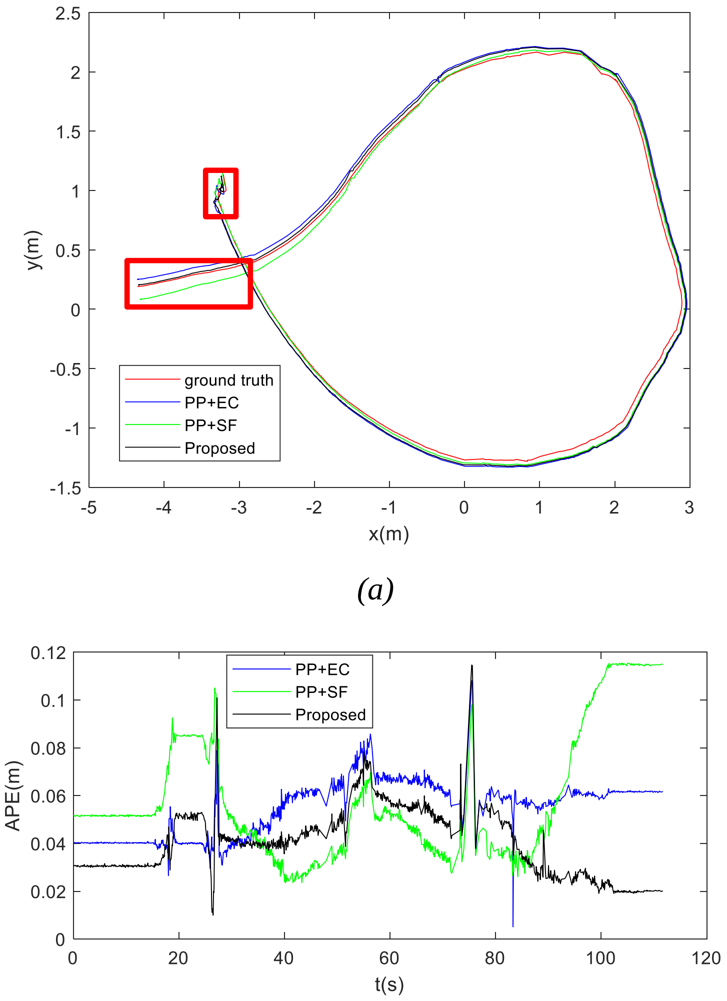
To show the help of the plane features in low textured scenes, we present a modified version of the proposed system with only point features and abbreviate it as PF. To demonstrate the advantage of the novel cost function with the analytical covariance for the plane measurement, we present a modified version without the covariance propagation (Zhang et al., 2019), and is abbreviated as PP+EC. To confirm the effectiveness of the novel factor graph using the new form for plane representation, we present a modified version without the new form, which simply uses the spherical form during bundle adjustment and is abbreviated as PP+SF.

The comparison results between the modified versions and the proposed system are shown in Table 4.2. PF loses tracking in front of the low textured wall due to lacking sufficient point feature matches. The proposed system improves the localization accuracy of PP+EC by 1.3 cm and 23.6%, which indicates the benefit of the analytical covariance. Furthermore, the proposed system outperforms PP+SF by 27.6% because the new form exploits more structural regularity.

Table 4. 2: Comparison of APE RMSE (cm) in room experiment.

Sequence	**Length (m)**	**PF**	**PP+EC**	**PP+SF**	**Proposed**
room	19.0	-	5.5	5.8	**4.2**

Figure 4.8(a) compares the trajectories from various methods with the ground truth. After the alignment, the trajectories from these methods are very consistent except for the beginning and the end. In the red box, it is indicated that the trajectory from the proposed system is the most consistent with the ground truth. Figure 4.8(b) shows the APEs of these methods. During short periods, Qualisys fails to track the reflective balls on the Kinect V2 due to occlusion, therefore the ground truth trajectory is not complete which results in large gradients of the APE curve in Figure 4.8(b). In the middle part of the sequence, the APE curve from the proposed system is close to those from PP+EC and PP+SF. But at the beginning and end, the proposed system generates smaller APE than them. To sum up, the proposed system is the most accurate among the four methods, which further confirms the effect of the analytical covariance and the new form for the plane representation.



(a)

(b)

Figure 4. 8: (a) Trajectories and (b) APE curves of PP+EC, PP+SF, and the proposed system.

4.4.4 Corridor experiment

An RGB-D sequence is captured in a 60-meter corridor using the same device in Figure 4.7(c). The ground truth of the corridor model is provided by NavVis M6 (NavVis M6, 2018). The example image in Figure 4.9 shows that the corridor contains a long and low textured wall

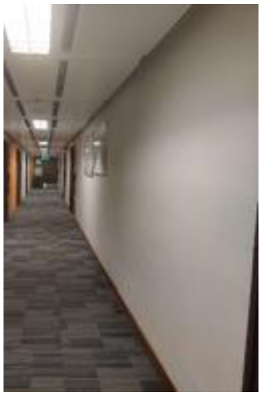


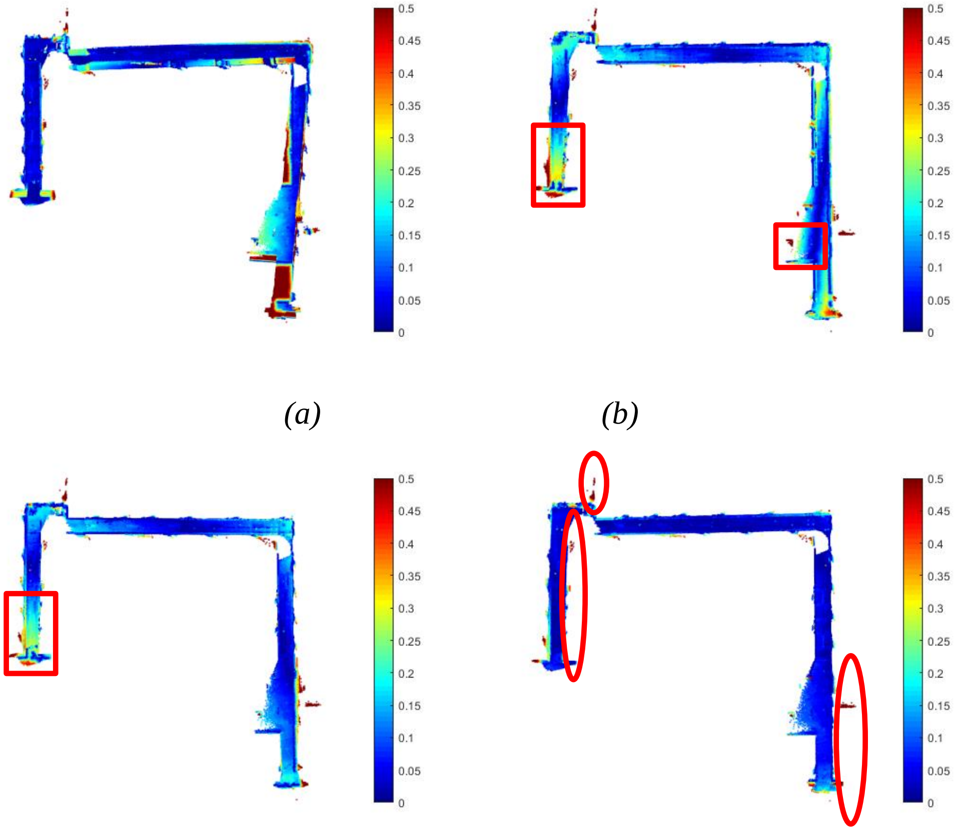
Figure 4. 9: Long and low textured wall in the operation scene.

The outputted model from the proposed system and the ground truth model are aligned by the ICP and then PTPD RMSE is computed for evaluating mapping quality. Table 4.3 shows the comparison results between the proposed system and the modified versions. The mapping quality of PF is low because point features are not reliable when the wheelchair passes the low textured walls. The difference between PP+EC and the proposed system is 2.5 cm, which reflects the advantage of using the analytical covariance. Moreover, the accuracy of the proposed system is better than PP+SF owing to the structural regularity exploited by the new form. The relative improvements by using the analytical covariance and the new representation form are 8.8% and 11.5%

Table 4. 3: Comparison of PTPD RMSE (cm) in corridor experiment.

Sequence	**Length (m)**	**PF**	**PP+EC**	**PP+SF**	**Proposed**
corridor	60.8	29.4	16.4	15.9	**14.5**

The outputted models from four methods are shown in Figure 4.10. The colour on the models represents the value of the PTPD. For better visualization, the maximum value on the colour bar is set as 0.5 m. The red circles in Figure 4.10(d) indicate outliers due to the glass windows. As indicated by the colour map in Figure 4.10(a), the model from PF has a large drift because the performance of point features degrades due to the low textured wall. The models from PP+EC and PP+SF are more accurate as shown in Figure 4.10(b) and (c) owing to the fusion of the plane features. The red boxes in Figure 4.10(b) and (c) indicate the bad-quality area from PP+EC and PP+SF, which are not observed in Figure 4.10(d). The comparison further proves the benefits of using the analytical covariance for the plane measurements and also the effects of using the new form for the plane representation.



(a) *(b)*

(c) *(d)*

Figure 4. 10: Point-to-point distances between the outputted models and the ground truth model. (a) PF; (b) PP+EC; (c) PP+SF; (d) Proposed.

4.4.5 Computation speed

The computation speed of the proposed system is then investigated using the sequence collected in the corridor. The time consumption by loop closure and incremental mapping is not listed as it is highly dependent on the keyframe number. Table 4.4 shows the processing time of each part in the tracking thread and local mapping thread and compares the total time cost with ORB-SLAM2.

Table 4. 4: Processing time (ms) of each part of the proposed system.

Thread	Part	Proposed	ORB-SLAM2
Tracking	Optical Flow	8.1	
	Plane Extraction and Matching	36.5	
	Robust Pose Solver	2.5	
	Shi-Tomasi Detection	9.0	
	IIR Filter	1.0	
	Total	57.1	40.6
Local Mapping	Factor Graph Optimization	142.9	235.7

Unlike Section 3.4.4, the image resolution from Kinect V2 is 960×540 instead of 640×480 provided by the iPad and structure sensor. The time cost for the point feature extraction is improved by the larger resolution. The processing speed of the proposed system is not comparable to ORB-SLAM2 due to time-consuming plane extraction, but it can still output localization and mapping results at 15 HZ. Furthermore, higher continuity and accuracy are achieved by the fusion of the plane features.

The average number of the correct feature matches is listed in Table 4.5. Though the plane features are much fewer than the point features, they are less affected by low textures and preserve more structure regularity.

Table 4. 5: Average number of correct feature matches.

Types	2D Point Feature Matches	3D Plane Feature Matches
number	242.8	2.9

4.5 Summary

As to perform successfully and completely in low textured scenes, a new RGB-D system using point and plane features is proposed in this chapter. In summary,

(a) This chapter derives the analytical covariance of the plane measurement in the spherical form by plane fitting and covariance propagation, which can overcome the disadvantage of the experimental weights used in the previous plane-based methods. The point reprojection errors and the plane transformation errors are combined based on the derived covariances, to build a new cost function for high-precision pose estimation.

(b) This chapter investigates the representation forms for plane features and develops a new form based on the parallel and vertical relationships among planes and MW axes. The new representation form can encode the structural regularity and does not rely on the MW assumption. A novel factor graph is constructed utilizing the new form to further refine the keyframe poses, point, and plane features.

(c) The experiment results of TUM RGB-D datasets prove that in low textured scenes, the proposed system yields higher continuity than the feature point-based method, and in rich textured scenes, the proposed system can generate higher accuracy than SOTA methods. The room and corridor experiments show that the proposed system can improve the localization and mapping accuracy, owing to the derived covariance and the new form for plane representation.

CHAPTER 5

TIGHTLY COUPLING RGB-D CAMERA AND WHEEL ODOMETER FOR GROUND VEHICLE NAVIGATION

As discussed in Section 2.4, the drift of the point-based methods is affecting the tracking quality of the mobile platform. Beyond the backend optimization techniques, fusing internal sensors is an effective method to reduce the drift. Though there are many visual-inertial systems (Li, 2014; Li & Mourikis, 2012, 2013; Zhang et al., 2020) and visual-odometric systems (Kang et al., 2019; Liu et al., 2019; Wu et al., 2017; Zheng & Liu, 2019; Zheng et al., 2018), the research works on fusing the wheel odometer and the RGB-D camera are still few (Labbé & Michaud, 2019; D. Yang et al., 2019). In addition, these works assume the mobile platform moves on a ground plane with no perturbation, which is not practical and may lead to non-optimal estimation results.

In this chapter, to reduce the tracking drift and avoid the backward of the previous methods, a tight-coupled system fusing the RGB-D camera and the wheel odometer is proposed. It softly assumes that a ground vehicle is equipped with an RGB-D camera and a wheel odometer with a rigid extrinsic transformation, and the vehicle moves on the ground plane with perturbations due to uneven terrain or platform vibration. To handle the perturbations on the ground plane, a two-stage strategy is employed to examine the planar motion assumption: (a) the Mahalanobis distance test in the frontend and (b) the ground plane estimation in the backend. The main contributions are listed as follows:

(a) It proposes a two-step strategy to examine the planar motion assumption. Compared with the previous methods using a hard planar motion assumption, the strategy can detect the perturbation of the mobile platform and further reduce the localization drift.

(b) It develops a novel factor graph consisting of all the constraints from the RGB-D camera, the wheel odometer, and the planar motion assumption. The associated covariances are also derived. This factor graph utilizes more information than the previous methods because of the derived covariances and the soft planar motion constraints.

(c) The proposed system is evaluated on the real-world datasets collected from both a small room scenario and a large corridor scenario. In the lab room experiment, compared with RTAB-map (Labbé & Michaud, 2019), which fuses the RGB-D camera and the wheel odometer loosely under a hard planar motion assumption, the proposed system can improve the localization accuracy from 8.1 cm to 4.8 cm (40.7%). In the corridor experiments, the improvement of the mapping accuracy is from 13.6 cm to 9.0 cm (33.8%).

The remainder of this chapter is designed as follows. Section 5.1 overviews the system, Section 5.2 introduces the notations, Section 5.3 presents the methodology, Section 5.4 shows the experiment results and Section 5.5 gives the conclusions.

5.1 System overview

The proposed system is built upon the open-sourced benchmark ORB-SLAM2 (Mur-Artal & Tardós, 2017). The RGB-D processing module is modified, and relative functions are added to support the wheel odometer and planar motion constraints. As shown in Figure 5.1, the proposed system consists of two sections running on ROS, namely frontend, and backend. A two-step strategy is utilized in the proposed system to handle the perturbation of the ground plane.

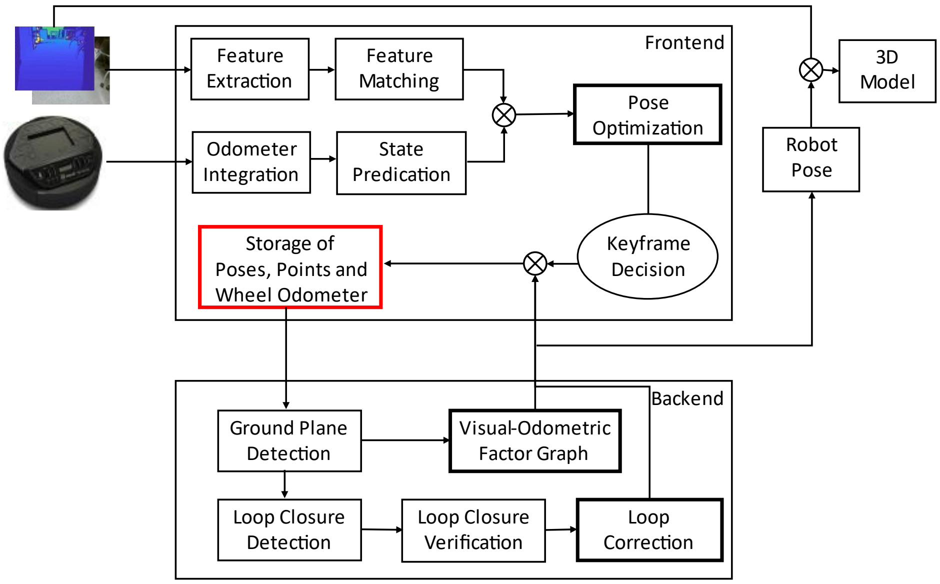


Figure 5. 1: Overview of the proposed system fusing RGB-D camera and wheel odometry.

(a) Frontend: RGB-D images and wheel odometer measurements are simultaneously fed into the frontend, and combined for the vehicle motion estimation. Firstly, ORB features are detected from the current RGB image and matched with the previous feature points in the map storage. Secondly, an initial guess of the vehicle pose is computed by the wheel odometer integration and its covariance is also propagated. Thirdly, a robust pose solver is built by jointly optimizing the visual, wheel odometer, and planar motion constraints. If a non-planar motion is detected by the Mahalanobis distance test, the planar motion constraint will be removed in the next iteration of the solver. Fourthly, the 3D occupancy map and the 2D grid map are constructed using the outputted vehicle pose and the down-sampled point cloud.

(b) Backend: Firstly, if the current frame is determined as a new keyframe, its associated feature matches, and integrated wheel odometer measurement are fed to the map storage. Secondly, a ground plane is detected and used to refine the planar motion constraint. Thirdly, a new factor graph is built by adding all the visual, wheel odometer, and planar motion constraints in a sliding window. Its optimization results are sent to the map storage to update the vehicle poses and map points. Fourthly, a loop closure candidate is searched by a bag-of-word approach and verified by a geometry check. Pose graph optimization is then performed to adjust the vehicle poses from the candidate frame to the current frame.

(c) Two-step strategy: The first step is the Mahalanobis distance test in the frontend. Large vibration is detected and removed in the pose solver. The second step is the ground plane detection in the backend. If a ground plane is estimated, its coefficients will be used to replace the experimental weight of the planar motion constraint. The ground plane detection is implemented in the backend instead of the frontend, which saves the computation cost. Furthermore, as the ground plane is not always observable, e.g., when the RGB-D camera is in front of an object, it is useful to employ the Mahalanobis distance test for the first step. The benefits of the two-step strategy are threefold: (a) the uneven terrain or the bumping can lead to the vibration of the ground vehicle, which can be detected by the Mahalanobis distance test; (b) the ground plane coefficients can further constrain the motion of the ground vehicle; (c) the planar motion constraints can be added to the new factor graph for the vehicle motion refinement.

5.2 Preliminaries

Figure 5.2 presents a ground vehicle equipped with an RGB-D camera and a wheel odometer. It is assumed that the wheel odometer is equipped in the center of the vehicle, and the vehicle frame is coincident with the wheel odometer frame. $\{w\}$ is employed to represent the world frame, $\{v_k\}$ and $\{c_k\}$ to denote the vehicle frame and the camera frame at instant k. The extrinsic transformation between the vehicle frame and the camera frame $\boldsymbol{T}_c^v$ are first calibrated by Zuñiga-Noël et al. (2019) and then refined by a full bundle adjustment. Feature point P is observed by the camera at consecutive instants.

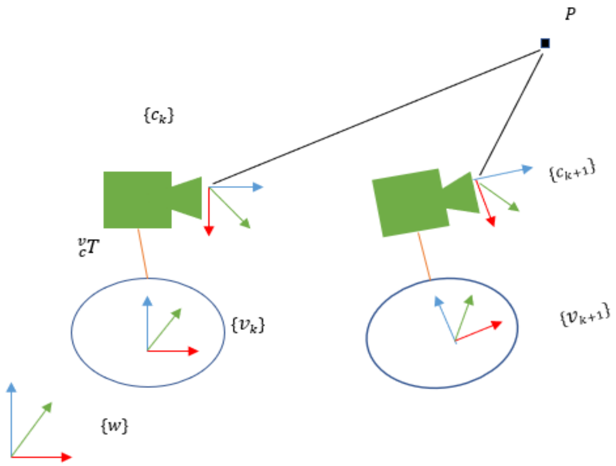


Figure 5. 2: Notationsof camera and vehicle frames.

The vehicle pose is defined as the transformation from $\{w\}$ to $\{v\}$, and the camera pose is that from $\{w\}$ to $\{c\}$. They are represented by ${}^{v}_{w}\boldsymbol{T}$ and ${}^{c}_{w}\boldsymbol{T}$ with six DoFs.

$$ {}^{v}_{w}\boldsymbol{T}_{4\times4} = \begin{bmatrix} {}^{v}_{w}\boldsymbol{R}_{3\times3} & {}^{v}_{w}\boldsymbol{t}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 1 \end{bmatrix} \in \mathrm{SE}(3),\ {}^{v}_{w}\boldsymbol{T} = \boldsymbol{T}^{v}_{c}\,{}^{c}_{w}\boldsymbol{T} \qquad (5.1) $$

5.3 Tightly coupling RGB-D camera and wheel odometer

The detail of tightly coupling RGB-D camera and wheel odometer is presented in Figure 5.1. It is further divided into five parts: (a) construction of visual, wheel odometer and planar motion constraints; (b) robust pose optimization; (c) ground plane detection; (d) construction of new factor graph; and (e) loop closure.

5.3.1 Visual, wheel odometer, and planar motion constraints

5.3.1.1 RGB-D image processing

Instead of detecting improved Shi-Tomasi features, the proposed system relies on modified ORB features in ORB-SLAM2 owing to its uniformed distribution and its invariance to rotation (Mur-Artal & Tardós, 2017). To speed up the feature matching procedure, the map points are projected to the image plane using the initial guess from the wheel odometer integration. Following conditions are set to ensure the correct matches: (a) the projected feature from the map point is closed to the detected feature; (b) the difference between their descriptors is small; (c) the detected feature and the map point are both projected back to the camera frame, and the difference between their 3-D positions in the camera frame is small.

After feature matching, point $\boldsymbol{P}_{iw}$ is tracked to the camera frame, and the pixel measurements on the image plane are $\boldsymbol{p}_{ic}$. $\boldsymbol{P}_{ic}$ is projected to the current frame using the estimated vehicle pose ${}_{w}^{v}\boldsymbol{T}$ and the extrinsic transformation ${}_{v}^{c}\boldsymbol{T}$

$$\boldsymbol{P}_{ic} = \left({}_{v}^{c}\boldsymbol{T}\,{}_{w}^{v}\boldsymbol{T}\begin{bmatrix}\boldsymbol{P}_{iw}\\ 1\end{bmatrix}\right)_{1:3} \tag{5.2}$$

where $(\quad)_{1:3}$ represents the first three elements of the homogeneous coordinate. The 2D point reprojection error is derived as

$$\boldsymbol{r}_{ic}^{2p} = \boldsymbol{p}_{ic} - f(\boldsymbol{K}\boldsymbol{P}_{ic}), f\left(\begin{bmatrix}a\\ b\\ c\end{bmatrix}\right) = \begin{bmatrix}a/c\\ b/c\end{bmatrix} \tag{5.3}$$

The covariance of $\boldsymbol{r}_{ic}^{2p}$ is similarly determined by Eq. (3.20).

5.3.1.2 Wheel odometer processing

To derive a general formulation for the wheel odometer integration, we assume that the wheel odometer provides a translation vector $\boldsymbol{\upsilon}$ and a rotation angle ω between the consecutive instants *t* and *t*+1, and its noise follows a normal distribution.

$${}_{t}^{t+1}\boldsymbol{\gamma}_{3\times1} = [{}_{t}^{t+1}\boldsymbol{\upsilon}^{T}, {}_{t}^{t+1}\omega]^{T} \tag{5.4}$$

$${}_{t}^{t+1}\boldsymbol{\delta}_{\boldsymbol{\gamma}_{3\times1}} = \left[{}_{t}^{t+1}\boldsymbol{\delta}_{\boldsymbol{\upsilon}}{}^{T}, {}_{t}^{t+1}\boldsymbol{\delta}_{\omega}\right]^{T} \sim N\left(0, {}_{t}^{t+1}\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}\right) \tag{5.5}$$

where ${}_{t}^{t+1}\boldsymbol{\gamma}_{3\times1}$ is the Lie algebra of SE(2) and $\boldsymbol{\delta}$ represents the noise. The state of ${}_{0}^{t+1}\boldsymbol{\gamma}$ can be propagated by

$$ {}^{t+1}_{0}\overline{\boldsymbol{\gamma}} = \left({}^{t+1}_{t}\boldsymbol{\gamma} - {}^{t+1}_{t}\boldsymbol{\delta}_{\gamma}\right) \oplus \left({}^{t}_{0}\boldsymbol{\gamma} - {}^{t}_{0}\boldsymbol{\delta}_{\gamma}\right) = \begin{bmatrix} {}^{t}_{0}\boldsymbol{v} + \varphi({}^{t}_{0}\omega - {}^{t}_{0}\boldsymbol{\delta}_{\omega})({}^{t+1}_{t}\boldsymbol{v} - {}^{t+1}_{t}\boldsymbol{\delta}_{v}) \\ {}^{t}_{0}\omega - {}^{t}_{0}\boldsymbol{\delta}_{\omega} + {}^{t+1}_{t}\omega - {}^{t+1}_{t}\boldsymbol{\delta}_{\omega} \end{bmatrix} \tag{5.6} $$

$$ \varphi({}^{t}_{0}\omega) = \begin{bmatrix} \cos{}^{t}_{0}\omega & -\sin{}^{t}_{0}\omega \\ \sin{}^{t}_{0}\omega & \cos{}^{t}_{0}\omega \end{bmatrix} \tag{5.7} $$

where $\oplus$ is the addition function for Lie algebra, and φ is the function to convert rotation angle to a rotation matrix.

For predicting the initial guess of the vehicle pose, the wheel odometer measurements between the previous keyframe and the current keyframe are integrated.

$$ {}^{vc}_{vk}\overline{\omega} = \textstyle\sum_{t=vk}^{vc-1}({}^{t+1}_{t}\omega - {}^{t+1}_{t}\boldsymbol{\delta}_{\omega}) = \sum_{t=wk}^{vc-1}{}^{t+1}_{t}\omega - \sum_{t=vk}^{vc-1}{}^{t+1}_{t}\boldsymbol{\delta}_{\omega} = {}^{vc}_{vk}\omega - {}^{vc}_{vk}\boldsymbol{\delta}_{\omega} \tag{5.8} $$

$$ \begin{aligned} {}^{vc}_{vk}\overline{\boldsymbol{v}} &= \sum_{t=vk}^{vc-1} \varphi({}^{t}_{0}\omega - {}^{t}_{0}\boldsymbol{\delta}_{\omega})({}^{t+1}_{t}\boldsymbol{v} - {}^{t+1}_{t}\boldsymbol{\delta}_{v}) \\ &\approx \sum_{t=vk}^{vc-1} \varphi({}^{t}_{0}\omega)\left(\boldsymbol{I} - \begin{bmatrix} 0 & -{}^{t}_{0}\boldsymbol{\delta}_{\omega} \\ {}^{t}_{0}\boldsymbol{\delta}_{\omega} & 0 \end{bmatrix}\right)({}^{t+1}_{t}\boldsymbol{v} - {}^{t+1}_{t}\boldsymbol{\delta}_{v}) \\ &= \sum_{t=vk}^{vc-1} \varphi({}^{t}_{0}\omega)\,{}^{t+1}_{t}\boldsymbol{\phi} - \sum_{t=wk}^{vc-1} \varphi({}^{t}_{0}\omega)\begin{bmatrix} 0 & -{}^{t}_{0}\boldsymbol{\delta}_{\omega} \\ {}^{t}_{0}\boldsymbol{\delta}_{\omega} & 0 \end{bmatrix}{}^{t+1}_{t}\boldsymbol{v} \\ &\quad - \sum_{t=vk}^{vc-1} \varphi({}^{t}_{0}\omega)\,{}^{t+1}_{t}\boldsymbol{\delta}_{v} + \sum_{t=wk}^{vc-1} \varphi({}^{t}_{0}\omega)\begin{bmatrix} 0 & -{}^{t}_{0}\boldsymbol{\delta}_{\omega} \\ {}^{t}_{0}\boldsymbol{\delta}_{\omega} & 0 \end{bmatrix}{}^{t+1}_{t}\boldsymbol{\delta}_{v} \\ &\approx {}^{vc}_{vk}\boldsymbol{v} - {}^{vc}_{vk}\boldsymbol{\delta}_{v} \end{aligned} \tag{5.9} $$

where ${}^{vc}_{vk}\omega$ and ${}^{vc}_{vk}\boldsymbol{v}$ are the integrated measurements for the rotation and the translation, and ${}^{vc}_{vk}\boldsymbol{\delta}_{\omega}$ and ${}^{vc}_{vk}\boldsymbol{\delta}_{v}$ are the associated noises. Specifically,

$$ {}^{vc}_{vk}\omega = \textstyle\sum_{t=vk}^{vc-1}{}^{t+1}_{t}\omega\,,\ {}^{vc}_{vk}\boldsymbol{v} = \sum_{t=vk}^{vc-1}\varphi({}^{t}_{0}\omega)\,{}^{t+1}_{t}\boldsymbol{v} \tag{5.10} $$

$$ {}^{vc}_{vk}\boldsymbol{\delta}_{\omega} = \textstyle\sum_{t=vk}^{vc-1}{}^{t+1}_{t}\boldsymbol{\delta}_{\omega} \tag{5.11} $$

$$ {}^{vc}_{vk}\boldsymbol{\delta}_{v} \approx \textstyle\sum_{t=vk}^{vc-1}\varphi({}^{t}_{0}\omega)\left(\begin{bmatrix} 0 & -{}^{t}_{0}\boldsymbol{\delta}_{\omega} \\ {}^{t}_{0}\boldsymbol{\delta}_{\omega} & 0 \end{bmatrix}{}^{t+1}_{t}\boldsymbol{\phi}\boldsymbol{v} + {}^{t+1}_{t}\boldsymbol{\delta}_{v}\right) \tag{5.12} $$

Its iterative form is derived for noise propagation.

$$\begin{aligned} \begin{bmatrix} {}^{t+1}_{vk}\boldsymbol{\delta}_{\boldsymbol{v}} \\ {}^{t+1}_{vk}\boldsymbol{\delta}_{\omega} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{I} & \varphi({}^{t}_{0}\omega)\begin{bmatrix} v_y \\ -v_x \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{t}_{vk}\boldsymbol{\delta}_{\boldsymbol{v}} \\ {}^{t}_{vk}\boldsymbol{\delta}_{\omega} \end{bmatrix} + \begin{bmatrix} \varphi({}^{t}_{0}\omega) & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} {}^{t+1}_{t}\boldsymbol{\delta}_{\boldsymbol{v}} \\ {}^{t+1}_{t}\boldsymbol{\delta}_{\omega} \end{bmatrix} \\ &= \boldsymbol{A}\begin{bmatrix} {}^{t}_{vk}\boldsymbol{\delta}_{\boldsymbol{v}} \\ {}^{t}_{vk}\boldsymbol{\delta}_{\omega} \end{bmatrix} + \boldsymbol{B}\begin{bmatrix} {}^{t+1}_{t}\boldsymbol{\delta}_{\boldsymbol{v}} \\ {}^{t+1}_{t}\boldsymbol{\delta}_{\omega} \end{bmatrix} \end{aligned} \tag{5.13}$$

where v_x and v_y are the first and second elements of ${}^{t+1}_{t}\boldsymbol{v}$, respectively. The covariance of the integrated wheel odometer measurements can be propagated by

$${}^{t+1}_{vk}\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} = \boldsymbol{A}{}_{vk}^{t}\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}\boldsymbol{A}^{T} + \boldsymbol{B}{}^{t+1}_{t}\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}\boldsymbol{B}^{T} \tag{5.14}$$

The initial guess of the vehicle pose ${}^{vc}_{w}\boldsymbol{\gamma}$ can be predicted using ${}^{vk}_{w}\boldsymbol{\gamma}$ and ${}^{vc}_{vk}\boldsymbol{\gamma}$. The state ${}^{vc}_{w}\boldsymbol{\gamma}$ and its covariance ${}^{vc}_{w}\Sigma_{\boldsymbol{\gamma}}$ can be also predicted using Eq. (5.13) and (5.14). In the 3D space, with the assumption of the planar motion, ${}^{vc}_{w}\boldsymbol{\gamma}$ is extended to a 3D transformation matrix ${}^{vc}_{w}\widetilde{\boldsymbol{T}}$ by

$${}^{vc}_{w}\widetilde{\boldsymbol{T}} = \begin{bmatrix} \cos{}^{vc}_{w}\omega & -\sin{}^{vc}_{w}\omega & 0 & {}^{vc}_{w}v_x \\ \sin{}^{vc}_{w}\omega & \cos{}^{vc}_{w}\omega & 0 & {}^{vc}_{w}v_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.15}$$

The initial guess ${}^{vc}_{w}\widetilde{\boldsymbol{T}}$ can be an alternative if insufficient features are detected in the operation scene. Furthermore, it is also a prior constraint for pose optimization. The error of the wheel odometer constraint is modelled as

$$\boldsymbol{r}_{c}^{wo} = \log\left({}^{vc}_{w}\widetilde{\boldsymbol{T}}^{-1}\,{}^{v}_{w}\boldsymbol{T}\right)^{\vee}_{1,2,6} \tag{5.16}$$

where the first and second elements of $\log\left({}^{vc}_{w}\widetilde{\boldsymbol{T}}^{-1}\,{}^{v}_{w}\boldsymbol{T}\right)^{\vee}$ correspond to the translation vector of ${}^{vc}_{w}\boldsymbol{\gamma}$, and the sixth element corresponds to the rotation of ${}^{vc}_{w}\boldsymbol{\gamma}$, respectively. The covariance of $\boldsymbol{r}_{c}^{wo}$ is equal to ${}^{vc}_{w}\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}$.

5.3.1.3 Planar motion assumption

The planar motion assumption indicates that the roll, pitch, and translation on the *z*-axis of vehicle motion should be close to zero with small variances. Therefore, we set the measurements for planar motion constraint as

$$\boldsymbol{\kappa} = [0,0,0]^{T} \tag{5.17}$$

The covariances are set experimentally by

$$\boldsymbol{\Sigma}_c^{\boldsymbol{\kappa}} = diag\left(\sigma_{rx}^2, \sigma_{ry}^2, \sigma_z^2\right) \tag{5.18}$$

The error of planar motion constraint is constructed as

$$\boldsymbol{r}_c^{\boldsymbol{\kappa}} = \boldsymbol{\kappa} - \log({}_{w}^{v}\boldsymbol{T})^{\vee}{}_{3,4,5} \tag{5.19}$$

where the fourth and fifth elements of $\log({}_{w}^{v}\boldsymbol{T})^{\vee}$ correspond to the rotation on the *x*-axis and *y*-axis, and the third element corresponds to the translation on the *z*-axis.

5.3.2 Robust pose optimization

The vehicle pose ${}_{w}^{v}\boldsymbol{\xi}$ can be optimized using the matched features, the prior states from the wheel odometer integration, and the planar motion assumption. The cost function is constructed as

$$\sum_i \rho\left(\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^2\right) + \rho\left(\|\boldsymbol{r}_c^{wo}\|_{{}_{w}^{vc}\boldsymbol{\Sigma}_{\gamma}}^2\right) + \rho\left(\|\boldsymbol{r}_c^{\boldsymbol{\kappa}}\|_{\boldsymbol{\Sigma}_c^{\boldsymbol{\kappa}}}^2\right) \tag{5.20}$$

where *i* is the index of the matched features, ρ is the Huber function, and $\boldsymbol{\Sigma}$ is the associated covariance. Compared with the cost function of the point-based methods $\sum_i \rho\left(\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^2\right)$, the new cost function utilizes additional constraints from the wheel odometer and the planar motion assumption. Instead of assuming a 100% flat ground plane, the planar motion constraint $\rho\left(\|\boldsymbol{r}_c^{\boldsymbol{\kappa}}\|_{\boldsymbol{\Sigma}_c^{\boldsymbol{\kappa}}}^2\right)$ is derived based on a soft assumption that the ground plane is flat but with perturbations, which is more practical and can lead to optimal estimation results (Labbé & Michaud, 2019).

Eq. (5.20) is iteratively solved using the Gauss-Newton algorithm implemented in g2o (Grisetti et al., 2011). The Mahalanobis distance test is employed after every four iterations.

$$\left\|\boldsymbol{r}_{ic}^{2p}\right\|_{\boldsymbol{\Sigma}_{ic}^{2p}}^{2} < \chi_{\alpha,2}, \|\boldsymbol{r}_{c}^{wo}\|_{{}_{w}^{vc}\Sigma_{\boldsymbol{\gamma}}}^{2} < \chi_{\alpha,3}, \|{}_{\boldsymbol{\kappa}}^{c}\boldsymbol{r}\|_{\Sigma_{c}^{\boldsymbol{\kappa}}}^{2} < \chi_{\alpha,3} \qquad (5.21)$$

where α is the threshold of Chi-square distribution, 3 is the DoF of ${}_{\boldsymbol{\kappa}}^{c}\boldsymbol{r}$, and $\|{}_{\boldsymbol{\kappa}}^{c}\boldsymbol{r}\|_{\Sigma_{\boldsymbol{\kappa}}^{c}}^{2}$ is its Mahalanobis distance. If $\|{}_{\boldsymbol{\kappa}}^{c}\boldsymbol{r}\|_{\Sigma_{c}^{\boldsymbol{\kappa}}}^{2}$ is above the threshold $\chi_{\alpha,3}$, a non-planar motion is marked that the ground vehicle is moving with large perturbations. Then the planar motion constraint will be excluded in the next iteration.

After pose estimation, a new keyframe will be selected if: (a) the relative motion between the current frame and the previous keyframe is above a threshold, or (b) the number of the correct feature matches are insufficient. The unmatched features on the new keyframe are then projected to the world frame$\{w\}$ using the vehicle pose ${}_{w}^{v}\boldsymbol{T}$ and the extrinsic transformation ${}_{c}^{v}\boldsymbol{T}$.

The map storage is then updated by the following factors: (a) the new keyframe and the new map points; (b) the data associations between the detected features and the map points; (c) the integrated wheel odometer measurement from the previous keyframe to the new keyframe; (d) the non-planar motion flag which is marked by the Mahalanobis distance test.

5.3.3 Ground plane detection

The Mahalanobis distance test may fail to detect the vibration of the ground vehicle if the experimental covariance of planar motion constraint $\boldsymbol{\Sigma}_{c}^{\boldsymbol{\kappa}}$ is not tuned well. Ground plane detection is the second step for insurance. The estimated coefficients of the ground plane are used to refine the planar motion constraint.

Similar to Section 4.3.1, a fast algorithm of agglomerative hierarchical clustering (Feng et al., 2014) is employed for the plane segmentation on the new keyframe, as shown in Figure 5.3. The clustered points on the segmented planes are robustly fitted to compute their coefficients in the Hessian form $\boldsymbol{\pi}_{jk} = [\boldsymbol{n}_{jk}^{T}, d_{jk}]^{T}$. These planes are projected to the world frame $\{w\}$ by

$$\boldsymbol{\pi}_{jw} = ({}^{c}_{v}\boldsymbol{T}\,{}^{v}_{w}\boldsymbol{T})^{T}\boldsymbol{\pi}_{jk} \tag{5.22}$$

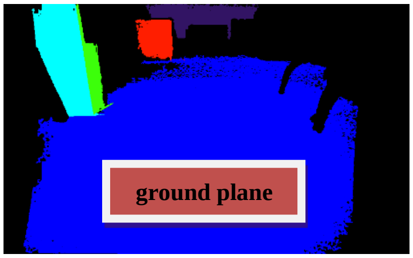


Figure 5. 3: Ground plane detection.

Four conditions are set to determine an accurate ground plane: (a) the included angle between $\boldsymbol{\pi}_{jw}$ and the z-axis is small; (b) d_{jw} is close to zero; (c) the plane j contains sufficient points; (d) the average of the plane fitting errors is small.

If an accurate ground plane is found on the new keyframe, its Hessian form on the camera frame $\boldsymbol{\pi}_{k}^{g} = \boldsymbol{\pi}_{jk}$ is applied to refine the planar motion constraint.

$$\boldsymbol{\tau}_{k}^{g} = s(\boldsymbol{\pi}_{k}^{g}) = \left[\arctan\frac{ny_g}{nz_g}, -\arcsin nx_g, d_g\right]^{T} \tag{5.23}$$

where $\boldsymbol{\pi}_{k}^{g} = [nx_g, ny_g, nz_g, d_g]^{T}$ and $\boldsymbol{\tau}_{k}^{g} = [rx_g, ry_g, d_g]^{T}$. Noticed that, $s(\,)$ is different from $q(\,)$ in Eq. (4.3). They both convert the unit normal to two rotation angles, which can build a rotation matrix like Eq. (4.7). While the rotation matrix from Eq. (5.22) can bring the normal to $[0, 0, 1]^{T}$ (the normal of ground plane), the matrix from Eq. (4.3) brings the normal to $[1, 0, 0]^{T}$. The rotation matrix of $\boldsymbol{\tau}_{k}^{g}$ is derived below and then extended to a full transformation matrix $\boldsymbol{T}_{k}^{g}$.

$$\boldsymbol{R}_k^g = \begin{bmatrix} \cos ry_g & 0 & \sin ry_g \\ 0 & 1 & 0 \\ -\sin ry_g & 0 & \cos ry_g \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos rx_g & -\sin rx_g \\ 0 & \sin rx_g & \cos rx_g \end{bmatrix} \tag{5.24}$$

$$\boldsymbol{T}_k^g = \begin{bmatrix} \boldsymbol{R}_k^g & \boldsymbol{t}_k^g \\ \boldsymbol{0} & 1 \end{bmatrix}, \boldsymbol{t}_k^g = \left[0,0,d_g\right]^T \tag{5.25}$$

The planar motion constraint is re-written by

$$\boldsymbol{r}_k^g = \log\left(\boldsymbol{T}_k^g {}_v^c\boldsymbol{T} {}_w^{vk}\boldsymbol{T}\right)^{\vee}{}_{3,4,5} \tag{5.26}$$

where ${}_w^{vk}\boldsymbol{T}$ is the vehicle pose of the keyframe k, ${}_v^c\boldsymbol{T}$ is the extrinsic calibration matrix between camera and vehicle frame, ${}_v^c\boldsymbol{T} {}_w^{vk}\boldsymbol{T}$ is the camera pose, and $\boldsymbol{T}_k^g$ aims to project the camera pose to the ground plane. The third, fourth, and fifth elements of $\log\left(\boldsymbol{T}_k^g {}_v^c\boldsymbol{T} {}_w^{vk}\boldsymbol{T}\right)^{\vee}$ should be close to zero, and indicates that the camera pose is transformed to the planar motion by $\boldsymbol{T}_k^g$. The covariance of re-written planar motion constraint is $\boldsymbol{\Sigma}_k^g$, which can be similarly derived by Eq. (4.16) and (4.17)

$$\boldsymbol{\Sigma}_k^g = J_{\boldsymbol{g},\boldsymbol{\mu}} \boldsymbol{\Sigma}_{\boldsymbol{\mu}_{jw}} J_{\boldsymbol{g},\boldsymbol{\mu}}^T, \boldsymbol{\tau}_k^g = \left[\arctan\frac{\mathrm{b}}{\mathrm{c}}, -\arcsin\frac{a}{\sqrt{a^2+b^2+c^2}}, \frac{1}{\sqrt{a^2+b^2+c^2}}\right]^T \tag{5.27}$$

$$J_{\boldsymbol{g},\boldsymbol{\mu}} = \frac{\delta_{\boldsymbol{\tau}_k^g}}{\delta_{\boldsymbol{\mu}_{jw}}} = \begin{bmatrix} 0 & \frac{c^2}{b^2+c^2} & -\frac{b^2}{b^2+c^2} \\ -\frac{\sqrt{b^2+c^2}}{a^2+b^2+c^2} & -\frac{ab}{(a^2+b^2+c^2)\sqrt{b^2+c^2}} & -\frac{ac}{(a^2+b^2+c^2)\sqrt{b^2+c^2}} \\ -\frac{a}{(a^2+b^2+c^2)^{\frac{3}{2}}} & -\frac{b}{(a^2+b^2+c^2)^{\frac{3}{2}}} & -\frac{c}{(a^2+b^2+c^2)^{\frac{3}{2}}} \end{bmatrix} \tag{5.28}$$

5.3.4 Visual-odometric factor graph construction

In Figure 5.4, after the ground plane detection, a new factor graph is constructed using all the visual, wheel odometer, and planar motion constraints in a sliding window. The vehicle poses outside the sliding window are set as fixed, and the map points are also fixed if they are observed by an early keyframe outside the window.

In addition, we assume that the extrinsic transformation $\boldsymbol{T}_c^v$ is fixed, so the camera poses are simply transformed from the vehicle poses, and no need to be optimized in the graph.

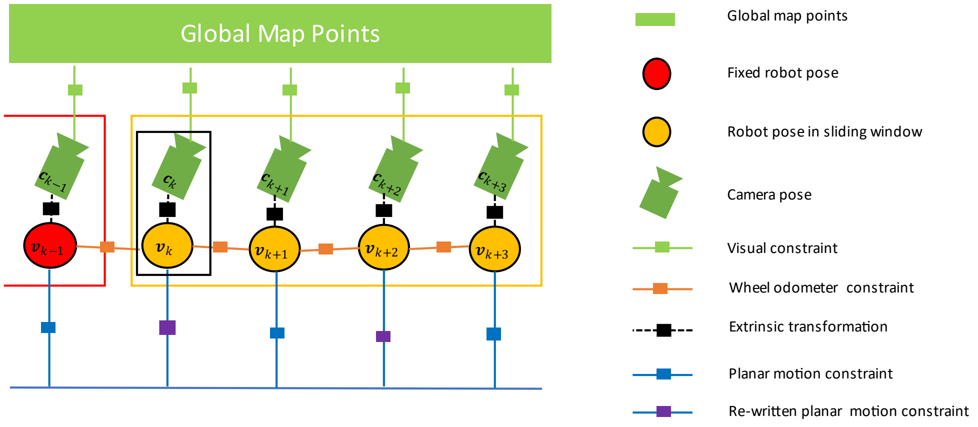


Figure 5. 4: Factor graph with all the visual, wheel, and planar motion constraints.

The wheel odometer constraint in Figure 5.4 is very different from the prior constraint in Section 4.2. Supposed that ${}^{vk+1}_{vk}\boldsymbol{\gamma}$ is the integrated wheel odometer measurement between the keyframes k and k+1, it is extended to a 3D transformation matrix ${}^{vk+1}_{vk}\widetilde{\boldsymbol{T}}$ similarly by Eq. (5.15). The error of the wheel odometer constraint is constructed by

$$\boldsymbol{r}_k^{wo} = \log\left({}^{vk+1}_{vk}\widetilde{\boldsymbol{T}}^{-1}\,{}^{vk+1}_{w}\boldsymbol{T}\,{}^{vk}_{w}\boldsymbol{T}^{-1}\right)^{\vee}_{1,2,6} \tag{5.29}$$

where ${}^{vk}_{w}\boldsymbol{T}$ and ${}^{vk+1}_{w}T$ are the vehicle poses to be estimated in the factor graph. The covariance for the wheel odometer integration $\boldsymbol{\Sigma}_{{}^{vk+1}_{vk}\boldsymbol{\gamma}}$ is propagated by Eq. (5.14).

A novel cost function is built below with all the constraints in the sliding window

$$\textstyle\sum_k \sum_i \rho\left(\left\|\boldsymbol{r}_{ik}^{2p}\right\|^2_{\boldsymbol{\Sigma}_{ik}^{2p}}\right) + \sum_k \|\boldsymbol{r}_k^{wo}\|^2_{\boldsymbol{\Sigma}_{{}^{vk+1}_{vk}\boldsymbol{\gamma}}} + \sum_k \|\boldsymbol{r}_k^{\boldsymbol{\kappa}}\|^2_{\boldsymbol{\Sigma}_c^{\boldsymbol{\kappa}}} + \sum_k \left\|\boldsymbol{r}_k^{g}\right\|^2_{\boldsymbol{\Sigma}_k^{g}} \tag{5.30}$$

where k and i are the indexes of the keyframes and feature points, ρ is the Huber function and $\boldsymbol{\Sigma}$ is the covariance matrix. The new cost function tightly fuses the constraints from the RGB-D camera, the wheel odometer, and the planar motion assumption, and utilizes more information than the point-based methods and previous methods fusing the wheel odometer and the RGB-D camera (Labbé & Michaud, 2019; D. Yang et al., 2019). In addition, it can work on uneven floors owing to the re-written planar motion constraints. Gauss-Newton method in g2o is applied to solve Eq. (5.30) again, which can provide the refined vehicle poses ${}^{vk}_{w}\boldsymbol{\xi}$ and the map points $\boldsymbol{P}_{iw}$. The planar motion constraint is an optional factor based on the two-step strategy:

(a) If the Mahalanobis distance test proves that the vehicle performs a non-planar motion, and an accurate ground plane is not found, neither $\boldsymbol{r}_k^{\boldsymbol{\kappa}}$ nor $\boldsymbol{r}_k^{g}$ is applied in Eq. (5.30).

(b) If the planar motion is marked by the Mahalanobis distance test, but an accurate ground plane is not found, $\boldsymbol{r}_k^{\boldsymbol{\kappa}}$ is utilized in Eq. (5.30).

(c) If an accurate ground plane is detected, then $\boldsymbol{r}_k^{\boldsymbol{\kappa}}$ is removed in Eq. (5.30) and $\boldsymbol{r}_k^{g}$ is applied to re-write the planar motion constraint.

5.3.5 Loop closing

Firstly, loop candidate is found by a bag-of-word approach and checked by geometry test, similar to Section 3.3.4. Pose graph optimization is then performed to adjust the vehicle poses. Unlike Section 3.3.4, integrated wheel odometer measurements are also fused to the pose graph, which can help to reduce the drift of vehicle pose. More detail can be referred to Section 3.3.4.

5.4 Experiments and results

Experiments are carried out using a Turtlebot2 ground vehicle in a lab room and a corridor scenario. As shown in Figure 5.5, the ground vehicle is equipped with a forward-looking Kinect V2 camera outputting RGB-D images and a Kobuki base providing wheel odometer measurements. Though 3-D LiDAR Velodyne VLP-16 is installed on the top of the vehicle, it is not used in the experiments. The onboard computer is an Intel NUC with i7-6770HQ CORE and 16 GB RAM. The power for Intel NUC and Kinect V2 is supplied by a GBTIGER power bank. Reflective balls are placed on the vehicle as markers for Qualisys, the motion capture system. In addition, an indoor mobile mapping system NavVis M6 is used to generate the ground truth for 3-D scene reconstruction, which is shown in Figure 3.10.

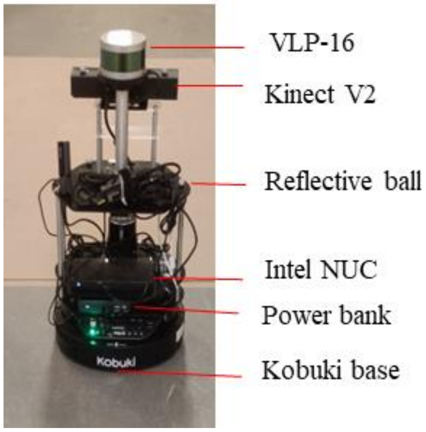
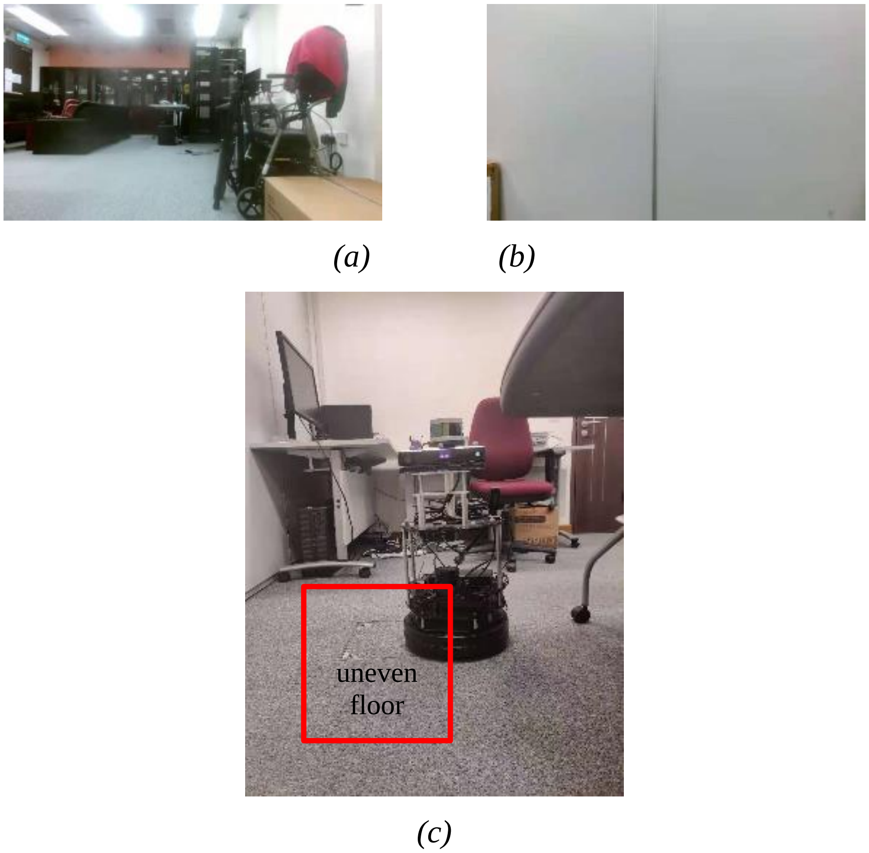


Figure 5. 5: Turtlebot 2, the ground vehicle.

The proposed system is compared with the wheel odometry, and two SOTA systems, RTAB-map (Labbé & Michaud, 2019) and ORB-SLAM2 (Mur-Artal & Tardós, 2017). The former one is a comprehensive system supporting a variety of sensors. This chapter uses the module of fusing the wheel odometer and the RGB-D camera for comparison. It loosely couples both sensors under a hard assumption that the robot is moving on a flat floor without perturbation. ORB-SLAM2 is also a comprehensive system supporting monocular, stereo, and RGB-D cameras. This chapter uses the module of RGB-D SLAM for comparison. In the room experiment, the APE between the ground truth and the trajectory from the proposed system is computed by aligning the latter to the former using Umeyama alignment (Umeyama, 1991). Then the RMSE of APE is employed to evaluate its localization accuracy. In the corridor experiment, the 3-D model reconstructed by the proposed system is aligned to the ground truth by the ICP. Then the RMSE of the PTPD is calculated to evaluate its mapping accuracy.

5.4.1 Lab room experiment

The lab room experiment covers several challenging instants for vehicle tracking. For example, two sampled images collected by RGB-D camera are shown in Figure 5.6(a) with black sofas and cabinets, glass windows, and illumination variation, and in Figure 5.6(b) with a low textured wall. Figure 5.6(c) is collected from the third-person view using a camera from a cell phone, and indicates the vibration of the ground vehicle due to the uneven floor in the clustered room.



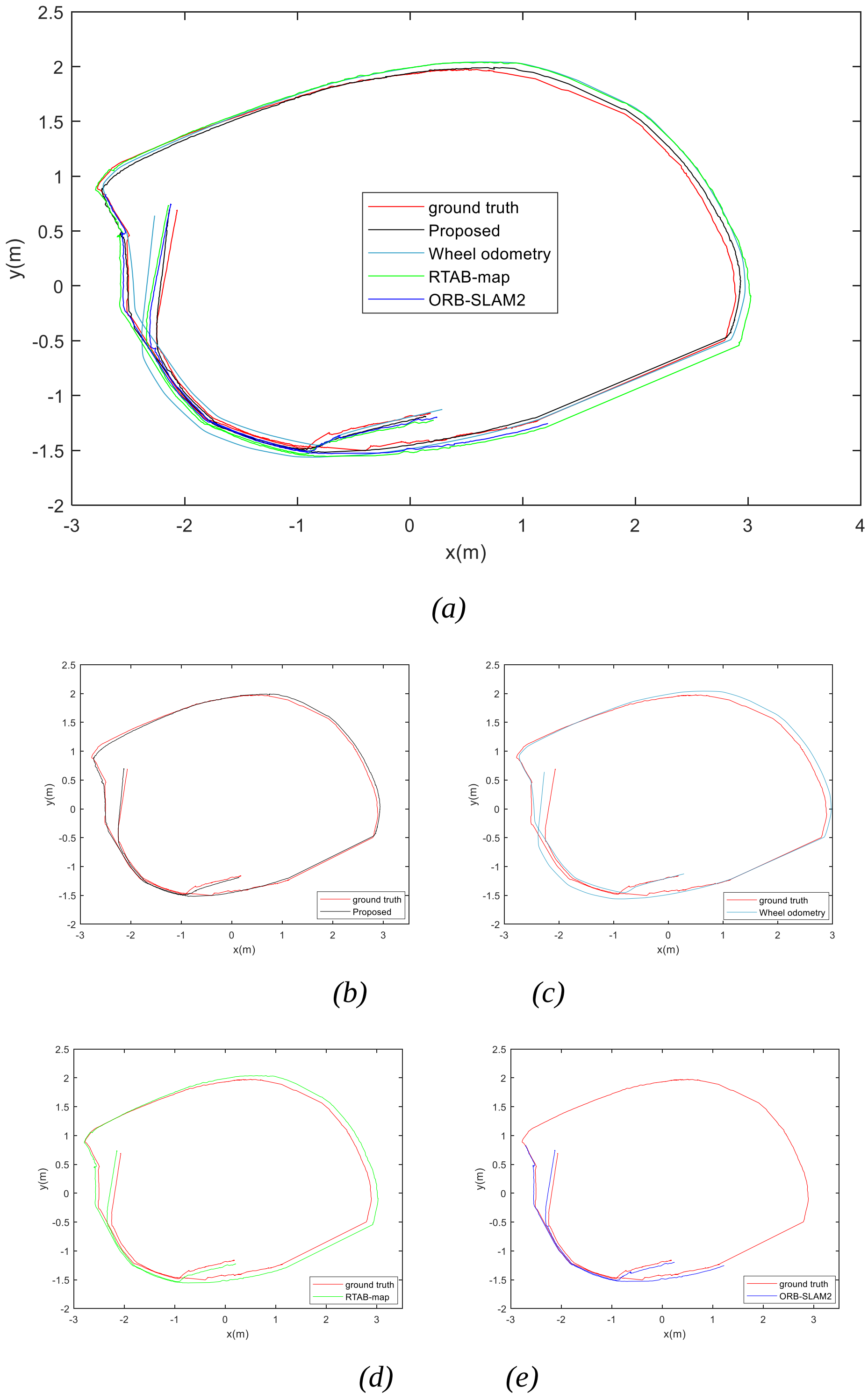
(a) *(b)*

(c)

Figure 5. 6: Challenging scenes in the room. (a) Black sofas and cabinets, glass windows and illumination variation; (b) the low textured wall; (c) the uneven floor.

The vehicle trajectories are estimated using various algorithms, including the proposed system, wheel odometry, RTAB-map, and ORB-SLAM2. They are aligned to the ground truth and compared in Figure 5.7(a). In addition, separate comparisons are presented in Figure 5.7(b)-(e). Above all, the most accurate trajectory is generated from the proposed system, which benefits from the two-stage strategy and the tight-coupling design. Though the trajectory of ORB-SLAM2 is aligned well with the ground truth, it loses tracking soon due to the low textured wall in Figure 5.6(b). Followed is RTAB-map, which assumes a 100% flat floor and loosely fuses the RGB-D camera and the wheel odometry. The worst trajectory is provided by the wheel odometry. Compared with RTAB-map, the proposed system can improve the localization performance by 3.3 cm and 40.7%.

In Table 5.1, the trajectory length is about 20.103 m, and the localization accuracy is defined as the ratio of APE RMSE to the length. The RMSEs of the *x*-axis, *y*-axis, *z*-axis, and the translation of the estimated trajectories are also presented. Above all, the best accuracy is marked as bolded and yielded by the proposed system, which is followed by ORB-SLAM2, RTAB-map, and wheel odometer. Because ORB-SLAM2 has no constraint for the planar motion assumption, its RMSE of the *z*-axis is not comparable to others. Both the wheel odometer and RTAB-map employ a hard planar motion constraint, which fixes the *z*-axis translation to zero. However, due to uneven floor, the ground vehicle may deviate on the ground plane and enlarge the localization error. The proposed system applies a two-stage approach for the planar motion constraint, and its motion in the *z*-axis is the most consistent with the ground truth.



(a)

(b) *(c)*

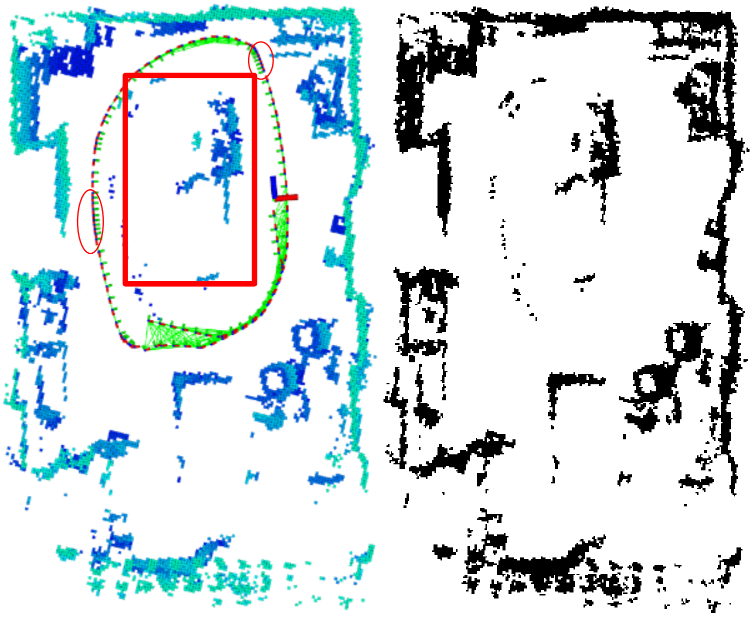
(d) *(e)*

Figure 5. 7: Vehicle trajectories in the room experiment. (a) Trajectories estimated by various algorithms are aligned to the ground truth; (b) Separate comparison of the proposed system; (c) wheel odometry; (c) RTAB-map; (d) ORB-SLAM2.

Table 5. 1: Comparison of APE RMSE (cm) in lab room experiment.

Room (20.103 m)	**Proposed**	**Wheel odometry**	**RTAB-map**	**ORB-SLAM2**
x-axis	**3.9**	10.9	6.2	5.1
y-axis	**2.8**	5.4	5.2	4.5
z-axis	**0.2**	0.6	0.6	0.8
APE RMSE	**4.8**	12.2	8.1	6.9
accuracy	**0.239%**	0.607%	0.403%	0.343%

Figure 5.8(a) presents the 3-D octomap, which is built by ray-casting using the camera poses and the point clouds. The voxel size of the octomap is set to 0.05 m. The red circles in Figure 5.8(a) indicate that dense keyframes are inserted due to insufficient feature matches occurring in Figure 5.6(a) and (b). The 3-D voxels in the octomap are projected to the ground plane as a 2-D grid map in Figure 5.8(b). A big meeting table is placed in the middle of the room and marked in a red box. There are lots of objects placed beside the walls, and they are projected to the ground together with the walls, so the boundary of the room is thick in the grid map.

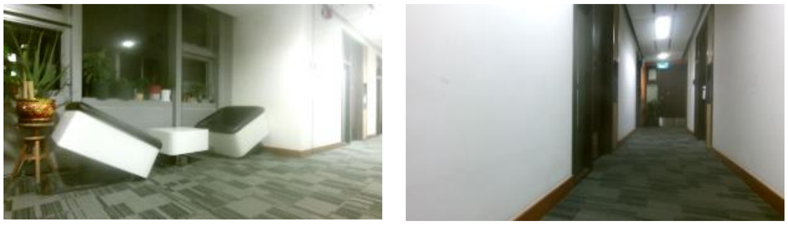


(a) *(b)*

Figure 5. 8: 3-D octomap (a) and 2-D grid map (b) in the lab room.

5.4.2 Corridor experiment

The corridor experiment also covers some challenging scenes, e.g., glass windows and low textured walls as shown in Figure 5.9, which can affect the continuity of SLAM systems. However, as the ground vehicle has a good viewing angle, all the algorithms can successfully process the corridor sequence and generate full trajectories as shown in Figure 5.10. As to align these trajectories in the same coordinate system, the camera trajectory from ORB-SLAM2 is transformed to the vehicle trajectory using ${}^{c}_{v}\boldsymbol{T}$, while other algorithms directly output vehicle pose. Then the trajectory from the proposed system is used as a reference for alignment. The green trajectory from RTAB-map is the most consistent with the reference. Followed are the trajectories from ORB-SLAM2 and wheel odometry.



(a) *(b)*

Figure 5. 9: Challenging scenes in the corridor. (a) glass windows; (b) low textured walls.

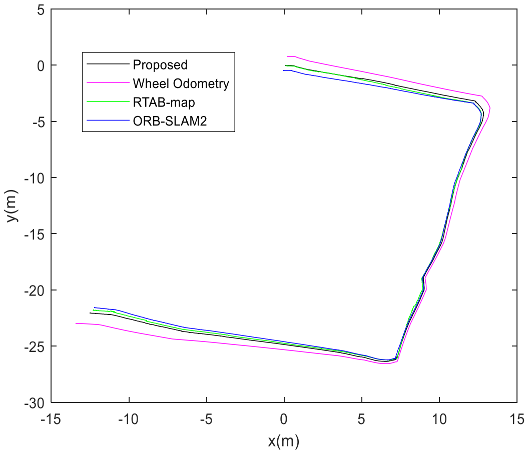
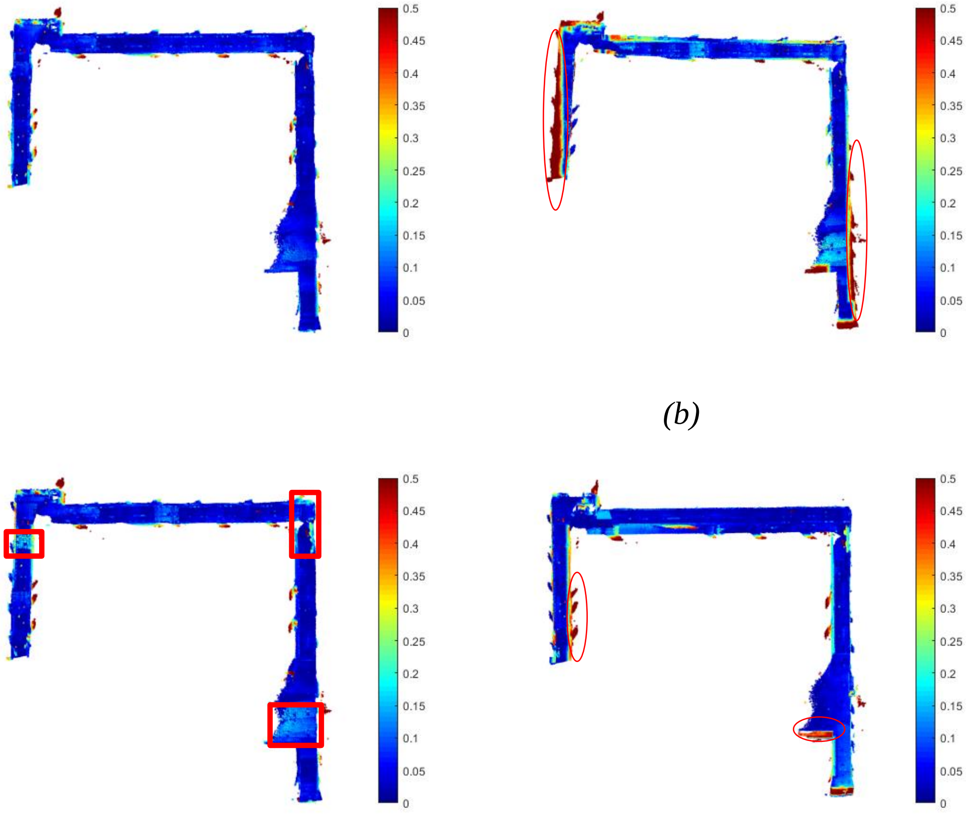


Figure 5. 10: Vehicle trajectories in the corridor experiment.

3-D models of the corridor are built incrementally by projecting the point cloud of RGB-D frames to the world frame. As to evaluate the mapping accuracy of various algorithms, 3-D models are compared with the ground truth in Figure 3.10 and the comparison result is shown in Figure 5.11. In Figure 5.11(a), there are some outlier points close to the glass windows, which are resulted from glass transmission. In Figure 5.11(b) and (d), red circles are employed to mark the points with large errors due to the tracking drift. To compare the mapping accuracy of the proposed system and RTAB-map, red boxes are placed in the area with large errors in Figure 5.11(c).



(b)

(c) (d)

Figure 5. 11: 3-D models generated by various algorithms. (a) the proposed system; (b) the wheel odometry; (c) RTAB-map; (4) ORB-SLAM2.

In Table 5.2, the length of the model is about 60.792 m, and the mapping accuracy is defined as the ratio of the PTPD RMSE to the length. RMSEs of the distances in three dimensions x-axis, y-axis, z-axis are also presented. The proposed system yields the best accuracy (9.0 cm and 0.148%) owing to the tight coupling of the RGB-D camera, wheel odometer, and planar motion constraints. Followed is RTAB-map (13.6 cm and 0.224%), which benefits from the loose coupling of RGB-D camera and wheel odometer. Compared with RTAB-map, the proposed system can improve the mapping accuracy by 4.6 cm and 33.8%.

Table 5. 2: Comparison of PTPD RMSE (cm) in corridor experiment

Corridor (60.792 m)	**Proposed**	**Wheel odometer**	**RTAB-map**	**ORB-SLAM2**
x-axis	**6.4**	16.6	7.4	15.9
y-axis	**5.5**	28.3	10.5	6.4
z-axis	**3.1**	4.9	4.3	8.7
PTPD RMSE	**9.0**	33.2	13.6	19.2
accuracy	**0.148%**	0.546%	0.224%	0.316%

The 3-D octomap and the 2-D grid map of the corridor are shown in Figure 5.12, which can be used for path planning and obstacle avoidance of the ground vehicle.

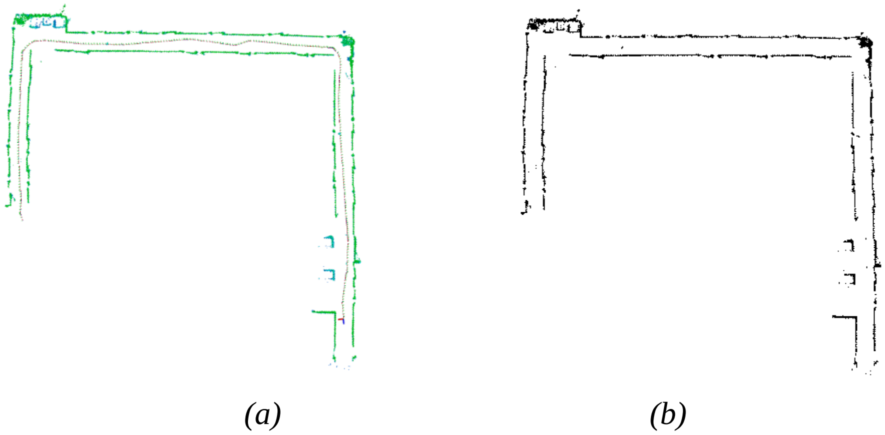


Figure 5. 12: 3-D octomap (a) and 2-D grid map (b) in the corridor.

5.4.3 Computation speed

As to investigate the computation speed, the room sequence is processed using the proposed system, and the processing time of each part is listed in Table 5.3. The time consumption by the loop closure is not listed as it is highly dependent on the keyframe number. While ORB-SLAM2 maintains a co-visibility map for local bundle adjustment, the sliding window used in the proposed system is lighter and contains fewer keyframes and map points, which reduces the computation cost. On average, the proposed system can output the vehicle pose and update the scene map within 52.5 (39.6+12.9) ms, which is sufficient for ground vehicle navigation.

Table 5. 3: Processing time (ms) of each part of the proposed system.

Thread	Part	Proposed	ORB-SLAM2
Tracking	ORB Extraction	25.1	
	ORB Matching	6.3	
	Wheel Odometer Integration	1.0	
	Planar Motion Assumption	1.0	
	Robust Pose Optimization	6.2	
	Total	39.6	41.2
Local Mapping	Factor Graph Optimization	55.8	235.7
Octomap	Map Update	12.9	

5.5 Summary

This chapter proposes a localization and mapping system for ground vehicle navigation, which fuses the RGB-D camera and the wheel odometer under a soft planar motion assumption. In summary,

(a) This chapter assumes the vehicle moves on the ground floor with perturbations. To deal with the perturbations, a two-stage strategy is proposed, which uses the Mahalanobis distance test in the frontend and the ground plane detection in the backend. Large vibration of the ground vehicle is detected and removed during pose optimization. The coefficients of the ground plane are also useful for constraining the vehicle motion. This strategy can work properly on the uneven floor and avoid the disadvantage of the methods based on the hard planar motion assumption.

(b) To further correct the drift of the vehicle tracking, a novel factor graph is constructed in this chapter, which tightly couples the visual, wheel odometer, and planar motion constraints. The associated covariances of these constraints are derived respectively. Compared with the previous methods fusing the wheel odometer and the RGB-D camera, the new factor graph exploits more constraints based on the results of ground plane detection. Therefore, it can help to reduce the accumulating drift and improve the accuracy of mobile platform tracking.

(c) The proposed system is evaluated and compared with other algorithms using self-collected datasets, which shows its superior performance in consideration of vehicle localization and scene reconstruction. Compared with RTAB-map, a loose-coupled system under hard motion assumption, the proposed system can improve the localization accuracy by 40.7% and the mapping accuracy by 33.8%.

CHAPTER 6

RGB-D SLAM BY HYBRID FEATURE FUSION AND WHEEL ODOMDTER INTEGRATION

6.1 Introduction and system overview

Three SLAM systems are presented respectively in the last three chapters. Chapter 3 focuses on fusing point and line features and extends previous works by combining both the 3D and 2D line reprojection errors. Chapter 4 pays attention to the fusion of point and plane features, and improves previous research from two aspects: (a) a new representation form for the plane features; (b) the analytical covariance of the plane measurement in the spherical form. Chapter 5 tightly couples RGB-D camera, wheel odometer under the planar motion assumption in a new factor graph, and proposes a two-step strategy to handle the perturbation of the ground plane.

In this chapter, benefiting from the research points highlighted in the last three chapters, a comprehensive SLAM system is presented to further improve the tracking continuity and accuracy, which tightly couples RGB-D camera and wheel odometer, and fuse point, line, and plane features. To achieve real-time processing, hybrid features are extracted and matched in three different threads in the proposed system. As shown in Figure 6.1, it has mainly two parts, frontend, and backend.

(a) Frontend. Firstly, the wheel odometer measurements are integrated to predict the initial pose of the mobile platform. Secondly, point, line, and plane features are simultaneously detected by ORB (Rublee et al., 2011), LSD (Von Gioi et al., 2012), and fast plane extraction algorithm (Feng et al., 2014). Thirdly, ORB and LSD features are matched by descriptor comparison, and the outlier matches are removed by ratio test and cross-check. Plane features are associated with previous planes by computing their coefficients in the world frame. Specifically, the initial guess of the mobile platform is used to aid the matching of hybrid features. Fourthly, a comprehensive and robust pose solver is built, which consists of pose-to-point, pose-to-line, and pose-to-plane constraints and the prior constraints from wheel odometer integration and planar motion assumption. Specifically, outlier measurements are detected and removed by the Mahalanobis distance test. Finally, if the current frame is selected as a keyframe, the map storage will be updated by the vehicle pose, the feature matches, the integrated wheel odometer measurements, and the planar motion flag. The point cloud of the current frame is also integrated to build a global 3D model for mobile platform navigation.

(b) Backend. The backend aims to compensate the vehicle drift in the frontend, and it consists of two threads for local mapping and loop closing, respectively. Firstly, in the local mapping thread, a new visual-odometric factor graph is constructed, which contains the wheel odometer constraint, the planar motion constraint, and the visual constraints from the hybrid features. The vehicle poses and hybrid features are simultaneously refined by the new factor graph. In the loop closing thread, the vehicle pose drift will be further reduced by pose graph optimization if the loop closure is detected by the bag-of-word approach and verified by the geometry check.

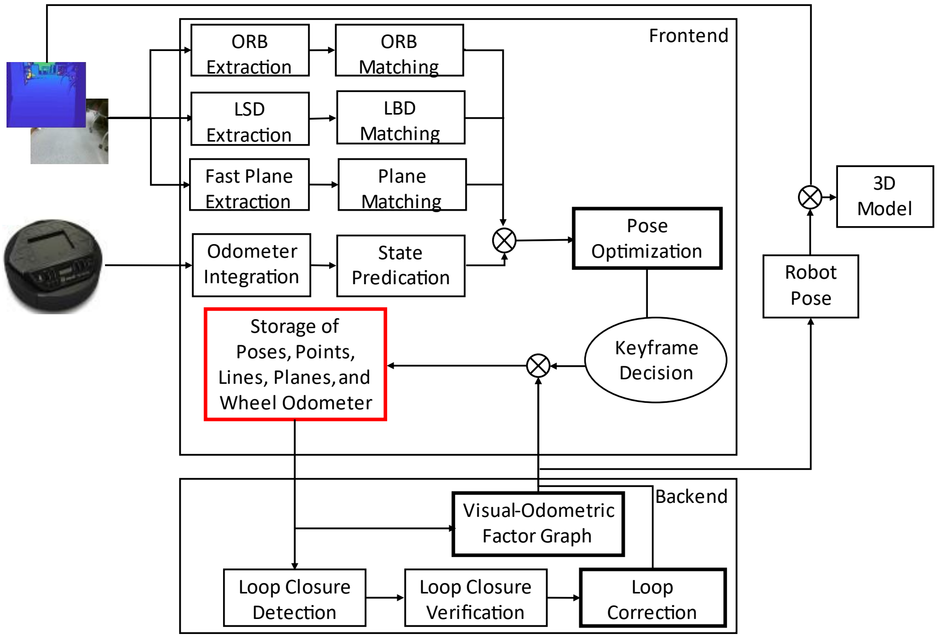


Figure 6. 1: Pipeline of the comprehensive SLAM system.

The remaining content of this chapter is divided into three parts, which focus on the full pipeline of the comprehensive system, the experiments and results, and the summary, respectively.

6.2 Hybrid feature fusion and wheel odometer integration

The full pipeline of the proposed system is presented in this section. The first step is to construct the constraints from the hybrid features, wheel odometer, and planar motion assumption. The second step is robust pose optimization with all the constraints. The third and fourth steps are factor graph optimization and loop closing, respectively.

6.2.1 Constraints from hybrid features, wheel odometer, and planar motion

In this section, the construction of wheel odometer, planar motion, and pose-to-point constraints is the same as Section 5.3.1. Pose-to-line and pose-to-plane constraints are built similarly with Section 3.3.1 and 4.3.1.

Firstly, the vehicle pose is propagated by wheel odometer integration and the prior constraint of the vehicle pose is modelled by Eq. (5.16). Secondly, the planar motion constraint is constructed by Eq. (5.19). Thirdly, the point features are detected by and matched by ORB algorithms (Rublee et al., 2011). Pose-to-point feature constraint is constructed based on Eq. (5.3) using the correct point matches. Fourthly, the line features are extracted by LSD (Von Gioi et al., 2012) matched by LBD, and selected by the three-step method proposed in Section 3.4.1. The 2D pose-to-line constraint is built using Eq. (3.14) if depth measurements on the detected line segments are not reliable. Otherwise, the 3D pose-to-line constraint is constructed by Eq. (3.16). Thirdly, the plane features are detected by a fast plane extraction algorithm (Feng et al., 2014), and matched based on their coefficients in the world frame. Eq. (4.11) is utilized to develop pose-to-plane constraints.

6.2.2 Robust pose optimization

The vehicle pose ${}^{v}_{w}\boldsymbol{\xi}$ can be robustly estimated using the guess from wheel odometer integration, the planar motion assumption, and hybrid feature measurements. The comprehensive cost function is designed below

$$\sum_i \rho\left(\left\|\boldsymbol{r}_{ic}^{2p}\right\|^2_{\boldsymbol{\Sigma}_{ic}^{2p}}\right) + \sum_j \rho\left(\left\|\boldsymbol{r}_{jc}^{3l}\right\|^2_{\boldsymbol{\Sigma}_{jc}^{3l}}\right) + \sum_j \rho\left(\left\|\boldsymbol{r}_{jc}^{2l}\right\|^2_{\boldsymbol{\Sigma}_{jc}^{2l}}\right) + \sum_m \rho\left(\|\boldsymbol{r}_{mc}\|^2_{\boldsymbol{\Sigma}_{mc}}\right) + \rho\left(\|\boldsymbol{r}_{c}^{wo}\|^2_{{}^{vc}_{w}\boldsymbol{\Sigma}_{\gamma}}\right) + \rho\left(\|\boldsymbol{r}_{c}^{\boldsymbol{\kappa}}\|^2_{\boldsymbol{\Sigma}_{c}^{\boldsymbol{\kappa}}}\right) \quad (6.1)$$

where i, j, and m are the indexes of the point, line, and plane features, ρ is the Huber function, and $\boldsymbol{\Sigma}$ is the associated covariance. Owing to the wheel odometer integration and the hybrid feature fusion, the comprehensive cost function utilizes more constraints than the cost functions of the proposed systems in the last three chapters. These constraints are beneficial for high-precision pose estimation. $\left\|\boldsymbol{r}_{ik}^{2p}\right\|^2_{\boldsymbol{\Sigma}_{ik}^{2p}}$ represents a pose-to-point constraint. $\left\|\boldsymbol{r}_{jk}^{2l}\right\|^2_{\boldsymbol{\Sigma}_{jk}^{2l}}$ and $\left\|\boldsymbol{r}_{jk}^{3l}\right\|^2_{\boldsymbol{\Sigma}_{jk}^{3l}}$ defines 2D and 3D pose-to-line constraints, respectively, $\|\boldsymbol{r}_{mc}\|^2_{\boldsymbol{\Sigma}_{mc}}$is the pose-to plane constraint by Eq. (4.11) with a different symbol. $\|\boldsymbol{r}_{c}^{wo}\|^2_{{}^{vc}_{w}\boldsymbol{\Sigma}_{\gamma}}$ and $\|\boldsymbol{r}_{c}^{\boldsymbol{\kappa}}\|^2_{\boldsymbol{\Sigma}_{c}^{\boldsymbol{\kappa}}}$ are the prior constraints from the wheel odometer and the planar motion assumption, and derived from Eq. (5.16) and (5.19), respectively. The comprehensive cost function exploits constraints from hybrid features, the wheel odometer prediction, and the planar motion assumption, which can help to generate more accurate pose estimation results than individual modules presented in the last three chapters.

The cost function can be iteratively minimized using the Gauss-Newton algorithm implemented in g2o (Grisetti et al., 2011), and the outlier measurements are detected and removed by the Mahalanobis distance test after every four iterations. A new keyframe is determined using the same criteria in the last three chapters: (a) the relative motion between the current frame and the previous keyframe; (b) the number of the correct feature matches. The unmatched point, line, and plane features of the new keyframe are inserted into the map storage together with the keyframe pose, the integrated wheel odometer measurement, and the non-planar motion flag.

6.2.3 Comprehensive factor graph construction

As shown in Figure 6.2, a comprehensive visual-odometric factor graph is built based on the visual constraints from the point, line, and plane features, and the wheel odometer and the planar motion constraints. The vehicle poses and hybrid features outside the window are considered fixed. Specifically, the 2D and 3D pose-to-line constraints are inserted in the factor graph based on Eq. (3.14) and (3.16). Four types of point-to-plane constraints are designed based on the new representation form of plane features by Eq. (4.4-4.6), and the detail is presented in Section 4.3.3.

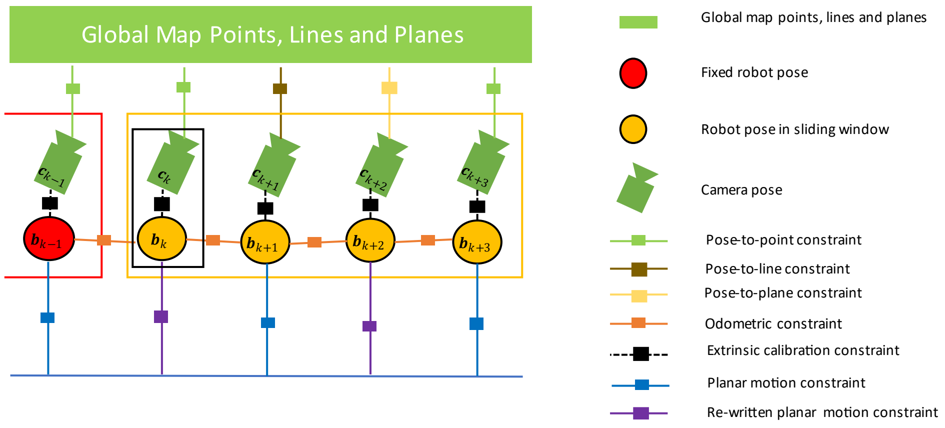


Figure 6. 2: Comprehensive factor graph with the visual, wheel and planar motion constraints.

A novel cost function is built below with all the constraints in the sliding window

$$\sum_k \sum_i \rho\left(\left\|\boldsymbol{r}_{ik}^{2p}\right\|_{\boldsymbol{\Sigma}_{ik}^{2p}}^2\right) + \sum_k \sum_j \rho\left(\left\|\boldsymbol{r}_{jk}^{2l}\right\|_{\boldsymbol{\Sigma}_{jk}^{2l}}^2\right) + \sum_k \sum_j \rho\left(\left\|\boldsymbol{r}_{jk}^{3l}\right\|_{\boldsymbol{\Sigma}_{jk}^{3l}}^2\right) +$$

$$\sum_k \sum_m \rho\left(\|\boldsymbol{r}_{mk}\|_{\boldsymbol{\Sigma}_{mk}}^2\right) + \sum_k \|\boldsymbol{r}_k^{wo}\|_{\boldsymbol{\Sigma}_{{}^{vk+1}_{vk}\boldsymbol{\gamma}}}^2 + \sum_k \|\boldsymbol{r}_k^{\boldsymbol{\kappa}}\|_{\boldsymbol{\Sigma}_C^{\boldsymbol{\kappa}}}^2 + \sum_k \left\|\boldsymbol{r}_k^g\right\|_{\boldsymbol{\Sigma}_k^g}^2 \tag{6.2}$$

where *k* is the index of the keyframe poses, *i* , *j*, and *m* are the indexes of the point, line, and plane features, respectively. The factor graph consists of comprehensive constraints from the hybrid features, the wheel odometer integration, and the planar motion assumption, which are more beneficial for pose estimation than the factor graphs in the last three chapters. $\|\boldsymbol{r}_{mk}\|_{\boldsymbol{\Sigma}_{mk}}^2$ is constructed based on the new representation form for the plane features. The wheel odometer constraint is derived from Eq. (5.29). A two-step strategy is employed to select the planar motion constraint optionally, which is detailed in Section 5.3.4. The cost function of Eq. (6.2) is again solved by the Gauss-Newton method in g2o, where the vehicle poses ${}_w^{vk}\boldsymbol{\xi}$ and the map features are refined iteratively.

6.2.4 Loop closing

The pipeline of the loop closing is the same as Section 3.3.4. In general, the similarity score between two frames is low in low textured scenes, and loop closure is only triggered in textured scenes, where the point feature matches are sufficient for loop closure verification. Therefore, the relative motion between the loop candidate and the current frame is computed using point features only, though hybrid features are detected.

6.3 Experiments and results

The proposed system is mainly built upon the navigation system proposed in Chapter 5. The processing modules of line and plane features in Chapters 3 and 4 are added to construct the pose-to-line and pose-to-plane constraints, respectively. All the experiments are conducted based on a Turtlebot2 ground vehicle, as shown in Figure 5.6. The ground truth of the vehicle trajectory is generated by Qualisys motion capture system in Figure 3.6, and that of the 3D scene reconstruction is obtained using NavVis M6 mobile mapping system in Figure 3.10.

In the last three chapters, three SLAM systems are proposed and then compared with SOTA systems by sufficient experiments and thorough analysis. In this chapter, a comprehensive RGB-D SLAM system by hybrid feature fusion and wheel odometer integration is presented. It is then compared with the previous systems to prove the advantages of sensor fusion and feature fusion. Similar to the last three chapters, RMSEs of the APE and the PTPD are employed to evaluate the localization accuracy and mapping accuracy, respectively. Furthermore, two additional experiments are carried out in two rooms with high rich and low textures, respectively. The ground vehicle starts and ends at the same position. The closing errors is defined as the difference between the estimated start and end points from SLAM systems, and is utilized to evaluate the continuity and accuracy of SLAM systems.

6.3.1 Lab room experiment

We collect a new dataset in the lab room using the Turtlebot2 ground vehicle. It also contains a challenging scene as shown in Figure 6.3. The vehicle trajectories are estimated by six methods: (a) point-based SLAM (abbreviated as P); (b) point + line SLAM proposed in Chapter 3 (abbreviated as PL); (c) point + plane SLAM proposed in Chapter 4 (abbreviated as PP); (d) point + line + plane SLAM (abbreviated as PLP); (e) tightly coupling of RGB-D camera and wheel odometer, proposed in Chapter 5 (abbreviated as WP); (f) comprehensive RGB-D SLAM by wheel odometer integration and hybrid feature fusion (abbreviated as WPLP).

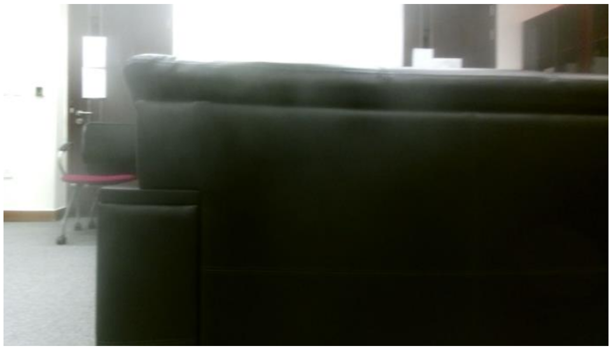
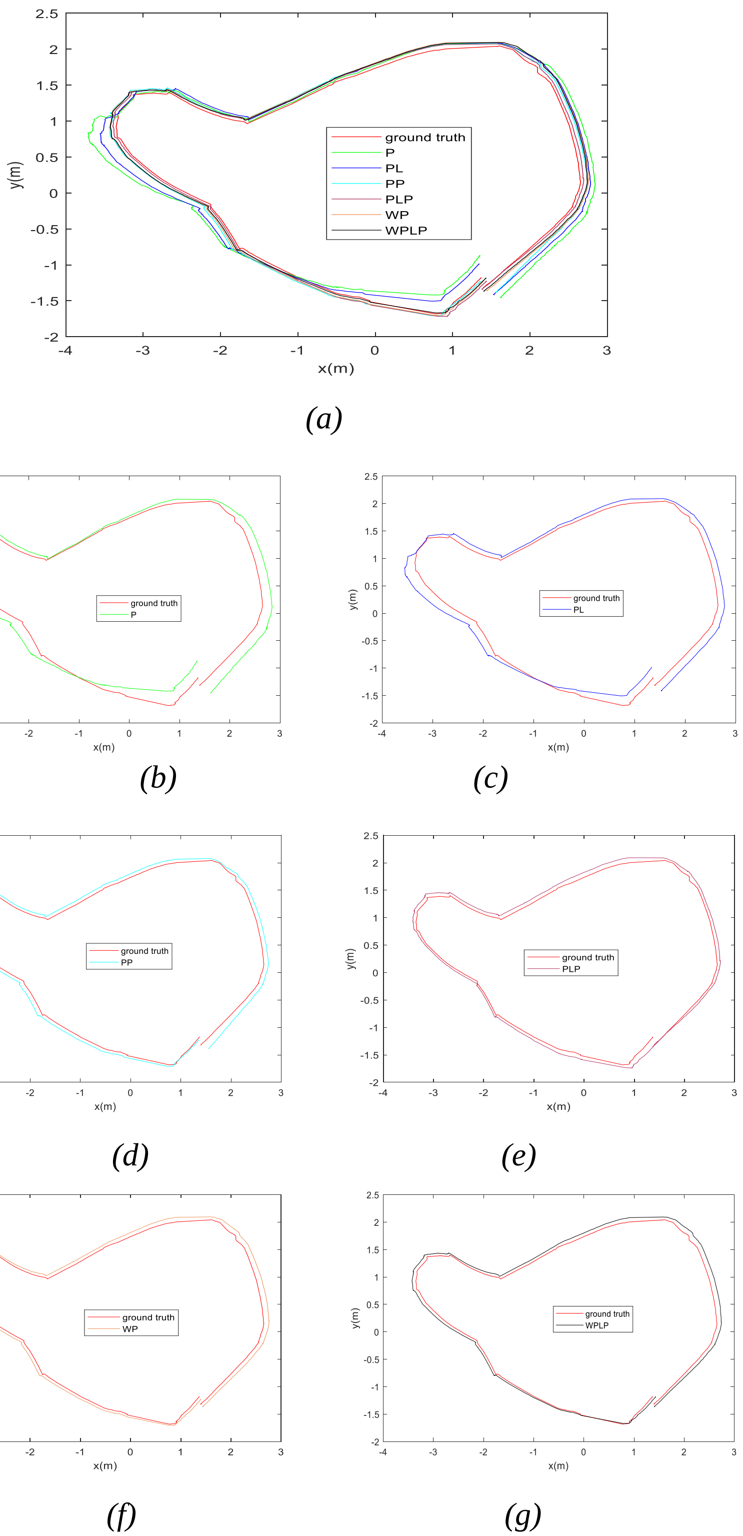


Figure 6. 3: A challenging scene where the camera view is blocked by a black sofa.

These trajectories are aligned to the ground truth and compared in Figure 6.4(a). In addition, separate comparisons are presented in Figure 6.4(b)-(g). A large drift is observed in Figure 6.4(b) and marked by a red circle, which indicates the performance degeneracy of point-based SLAM in the low textured scene in Figure 6.3. Such drift can corrupt the localization continuity and result in tracking failure of the ground vehicle. It can be compensated by fusing line and plane features in Figure 6.4(c)-(e). Specifically, the improvement by fusing the plane feature is more significant than fusing the line features. Figure 6.4(f) indicates that the prior constraints from the wheel odometer and the planar motion assumption can also reduce the drift. With the aid of hybrid features, the accuracy of vehicle pose is further improved in Figure 6.4(g).



(a)

(b)

(c)

(d)

(e)

(f)

(g)

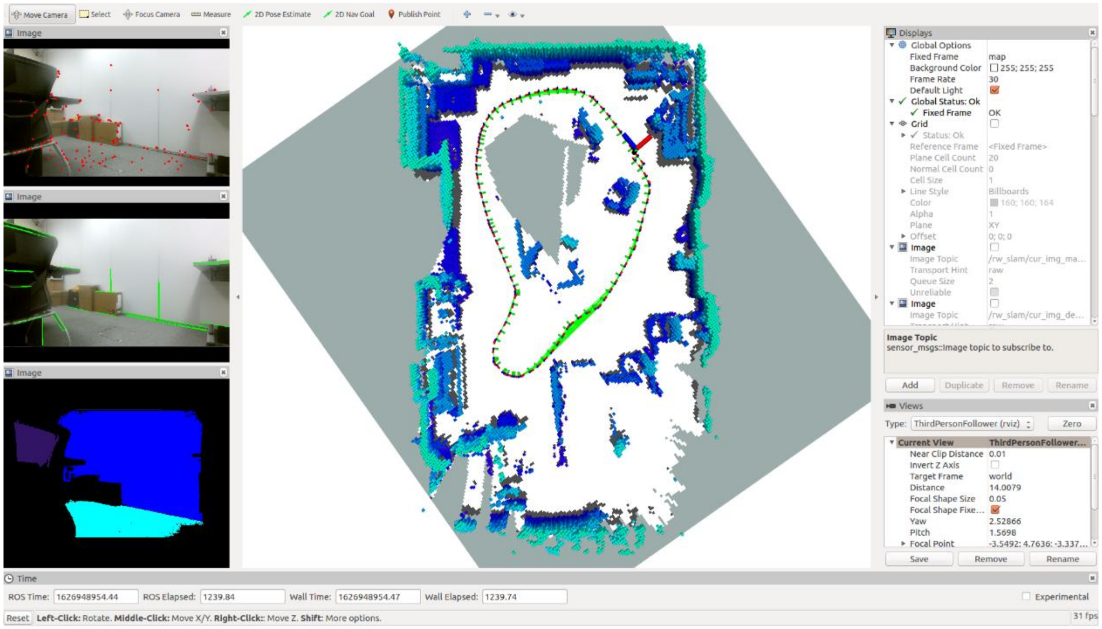
Figure 6. 4: Vehicle trajectories in the room experiment. (a) Trajectories estimated by six methods are aligned to the ground truth; (b) Separate comparison of P; (c) PL; (c) PP; (d) PLP; (f) WP; (g) WPLP.

In Table 6.1, the trajectory length is about 18.405 m, and the RMSEs and localization accuracies of six methods are also presented. The accuracy of the point-based SLAM is promoted by the fusion of line features, plane features, and wheel odometer, respectively. Above all, the best accuracy of 0.337% and the lowest RMSE of 6.2 cm are generated by the comprehensive RGB-D SLAM by wheel odometer integration and hybrid feature fusion. It takes advantage from two aspects: (a) the prior constraints from the wheel odometer and the planar motion assumption, which can help to select correct feature matches and guide the optimization of the vehicle pose; (b) robust and accurate line and plane features, which can reduce the drift when point feature is insufficient or with low quality. Compared with the feature point-based system, the comprehensive system can improve the localization accuracy by 70.1%, fusing the wheel odometer can improve it by 66.3%, the fusion of the points, lines, and planes can improve it by 57.2%, combining plane features can improve it by 53.8%, and adding line features can improve it by 33.6%.

Table 6. 1: Comparison of APE RMSE (cm) in lab room experiment.

Room (18.405 m)	**P**	**PL**	**PP**	**PLP**	**WP**	**WPLP**
RMSE	20.8	13.8	9.6	8.9	7.0	**6.2**
accuracy	1.130%	0.750%	0.522%	0.484%	0.380%	**0.337%**

Figure 6.5 presents a snapshot of the comprehensive RGB-D SLAM system. Three images in the left part of Figure 6.5 show the extraction and matching results of point, line, and plane features, respectively. The vehicle poses of the keyframes, the 3D octomap and the 2D grid map are depicted in the right part. Specifically, the octomap is built by ray-casting and the grid map is generated by projecting the octomap to the ground plane.



Global Options
Fixed Frame map
Background Color □ 255; 255; 255
Frame Rate 30
Default Light ☑
Global Status: Ok
✓ Fixed Frame OK
Grid □
✓ Status: Ok
Reference Frame <Fixed Frame>
Plane Cell Count 20
Normal Cell Count 0
Cell Size 1
Line Style Billboards
Color 160; 160; 164
Alpha 1
Plane XY
Offset 0; 0; 0
Image □
Image Topic /rw_slam/cur_img_ma…
Transport Hint raw
Queue Size 2
Unreliable □
Image □
Image Topic /rw_slam/cur_img_de…

Image Topic
sensor_msgs::Image topic to subscribe to.

Type: ThirdPersonFollower (rviz)

Current View ThirdPersonFollower…
Near Clip Distance 0.01
Invert Z Axis □
Target Frame world
Distance 14.0079
Focal Shape Size 0.05
Focal Shape Fixe… ☑
Yaw 2.52866
Pitch 1.5698
Focal Point -3.5492; 4.7636; -3.337…

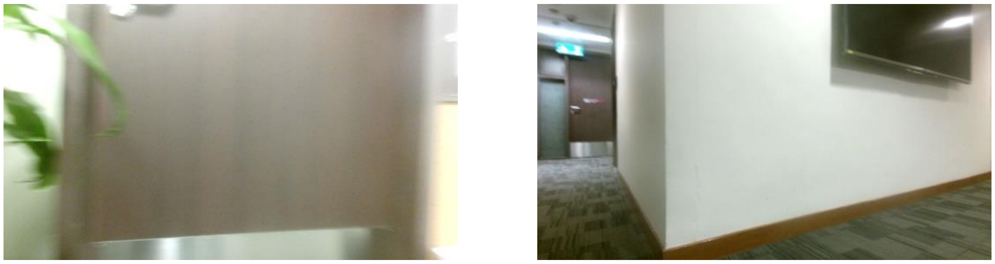
ROS Time: 1626948954.44 ROS Elapsed: 1239.84 Wall Time: 1626948954.47 Wall Elapsed: 1239.74 □ Experimental

Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Move Z. Shift: More options. 31 fps

Figure 6. 5: Snapshot of the comprehensive RGB-D SLAM system in the lab room experiment.

6.3.2 Corridor experiment

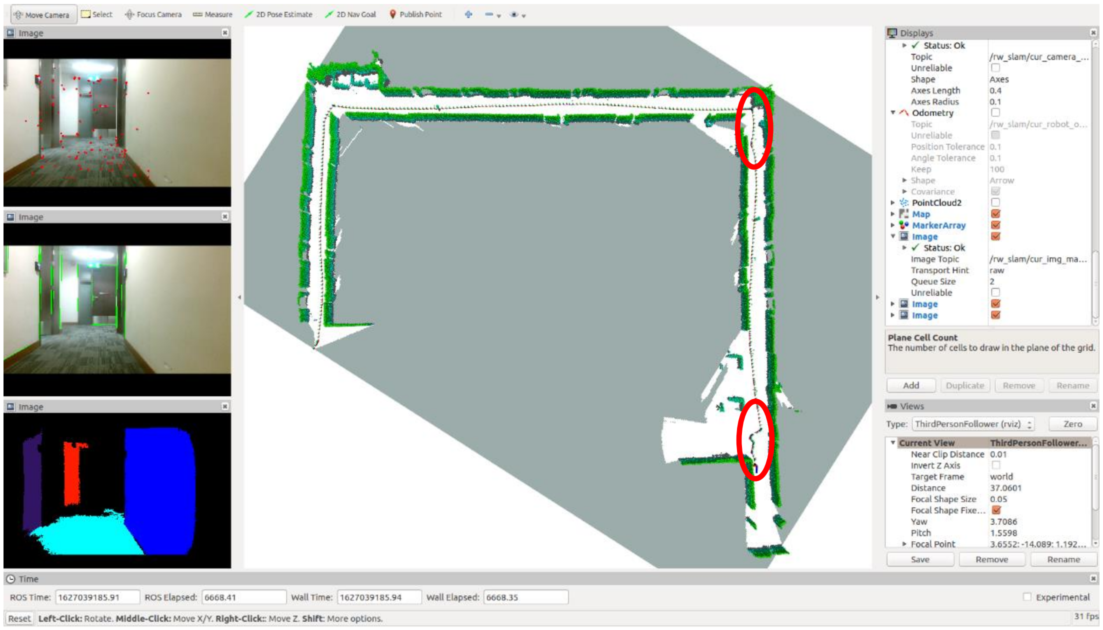
We collect a new dataset in the corridor, where the Turtlebot2 ground vehicle performs fast rotations in the turnings of the corridor. Two example images captured in the turnings are shown in Figure 6.6, which shows motion blur and low texture, respectively.



(a) (b)

Figure 6. 6: Example image captured in the first (a) and second (b) turnings.

Figure 6.7 is the snapshot of the comprehensive system in the corridor experiment, and the fast motion parts are marked by red circles.



✓ Status: Ok	
Topic	/rw_slam/cur_camera_...
Unreliable	□
Shape	Axes
Axes Length	0.4
Axes Radius	0.1
Odometry	□
Topic	/rw_slam/cur_robot_o...
Unreliable	□
Position Tolerance	0.1
Angle Tolerance	0.1
Keep	100
Shape	Arrow
Covariance	☑
PointCloud2	□
Map	☑
MarkerArray	☑
Image	☑
✓ Status: Ok	
Image Topic	/rw_slam/cur_img_ma...
Transport Hint	raw
Queue Size	2
Unreliable	□
Image	☑
Image	☑

Plane Cell Count
The number of cells to draw in the plane of the grid.

Type: ThirdPersonFollower (rviz)

Current View	ThirdPersonFollower...
Near Clip Distance	0.01
Invert Z Axis	□
Target Frame	world
Distance	37.0601
Focal Shape Size	0.05
Focal Shape Fixe...	☑
Yaw	3.7086
Pitch	1.5598
Focal Point	3.6552; -14.089; 1.192...

ROS Time: 1627039185.91 ROS Elapsed: 6668.41 Wall Time: 1627039185.94 Wall Elapsed: 6668.35 □ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click::** Move Z. **Shift:** More options. 31 fps

Figure 6. 7: Snapshot of the comprehensive RGB-D SLAM system in the corridor experiment.

The trajectories of the ground vehicle are computed by six methods and compared in Figure 6.8. We argue that with the fusion of the wheel odometer and hybrid features, the comprehensive system generates the most accurate results, which can be utilized as a reference for trajectory comparison. The tracking accuracy of the point-based SLAM is the worst, and its trajectory deviates the most from the reference, especially in the turning parts with fast motion. With the fusion of the line and plane features, the trajectories of PL, PP, and PLP are more consistent with the reference. Specifically, the improvement by the plane feature is more essential than that by the line feature. By tightly coupling the RGB-D camera and the wheel odometer, the accuracy of WP is also promoted, which is the most consistent with the reference.

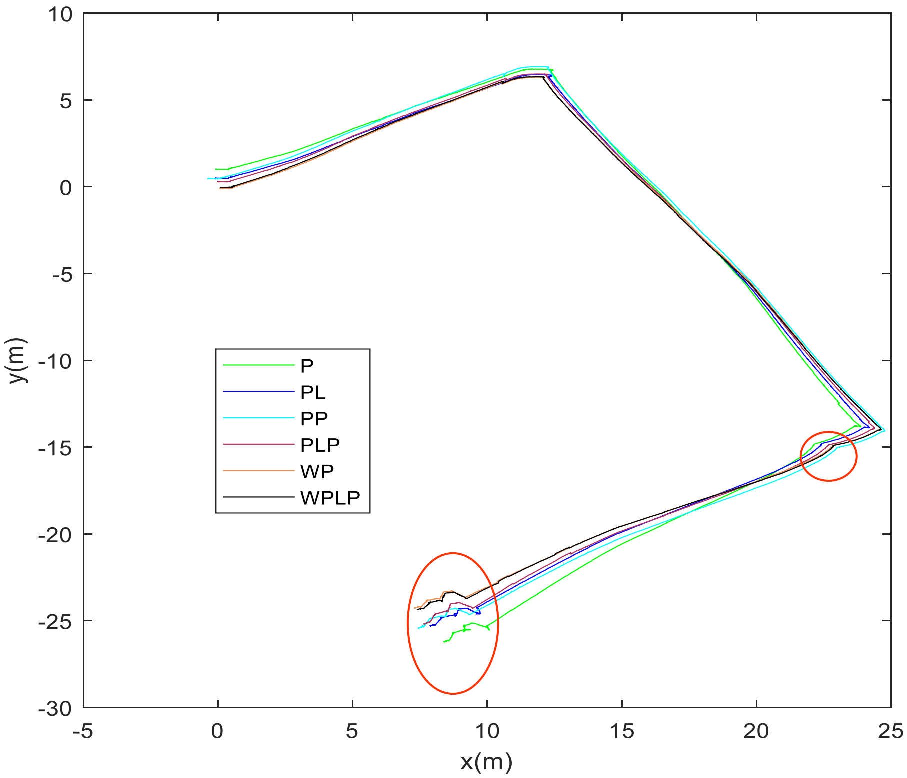
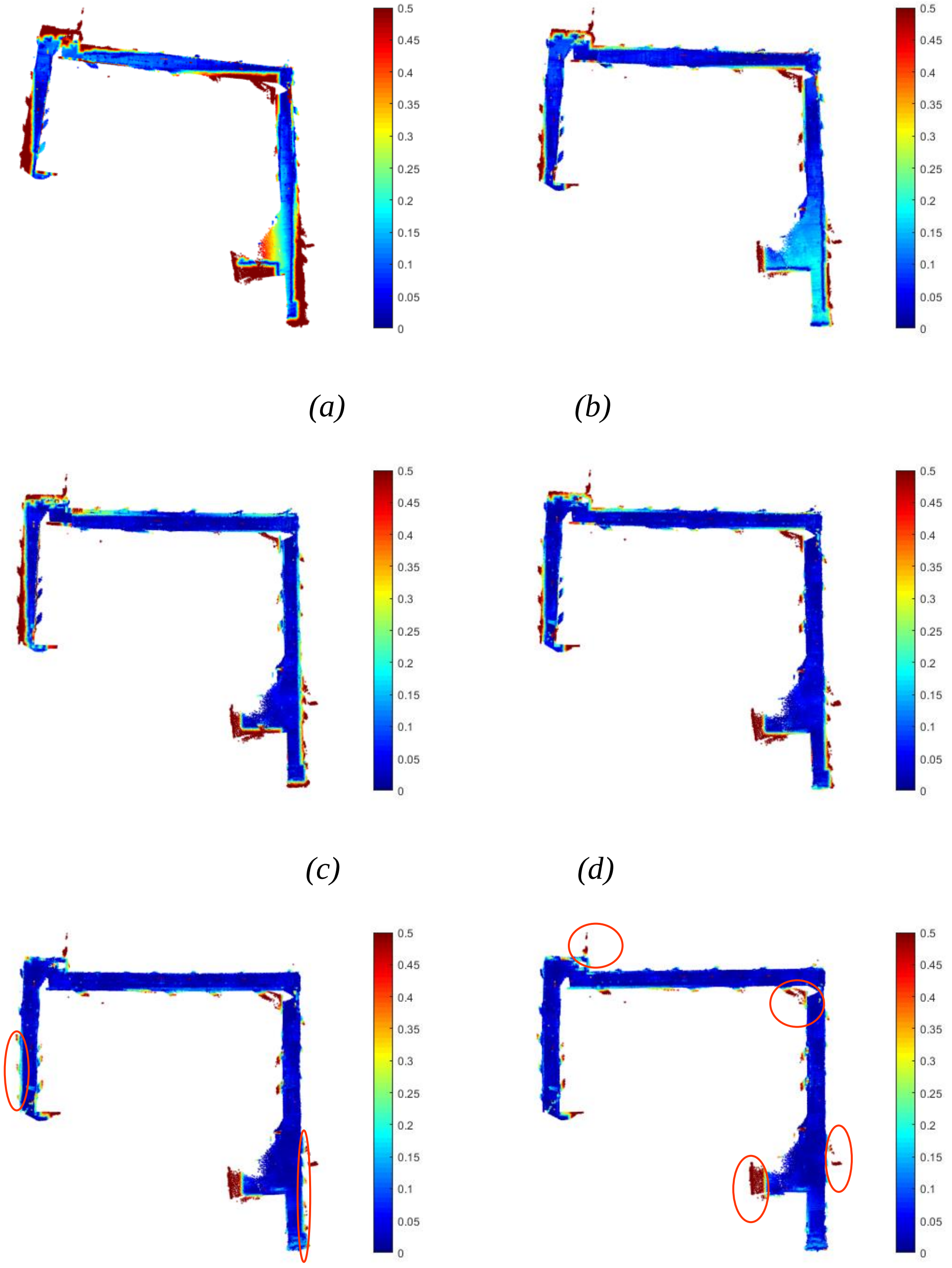


Figure 6. 8. Vehicle trajectories in the corridor experiment.

3-D models of the corridor are built incrementally by projecting the point cloud of RGB-D frames to the world frame. Their accuracies are evaluated by comparing with the ground truth model in Figure 3.10, and shown in Figure 6.9. The red circles in Figure 6.9(f) indicate the outliers due to opened door, glass transmission and unobservable area in ground truth model, which are observed in all the 3D models. In Figure 6.9(a), the model of point-based SLAM shows a large drift compared with the ground truth mode. This drift is reduced by the fusion of the line and plane features in Figure 6.9(b-d) significantly. With the aid of the wheel odometer and the planar motion assumption, WP can generate a high-quality 3D model, but the outlier points are still observed and marked by red circles in Figure 6.9(e). By further fusing hybrid features, these outliers disappear in Figure 6.9(f), which can prove the benefit of the comprehensive RGB-D SLAM system.



(a) (b)

(c) (d)

(e) (f)

Figure 6. 9: 3-D models generated by six methods. (a) P; (b) PL; (c) PP; (d) PLP; (e) WP; (f) WPLP.

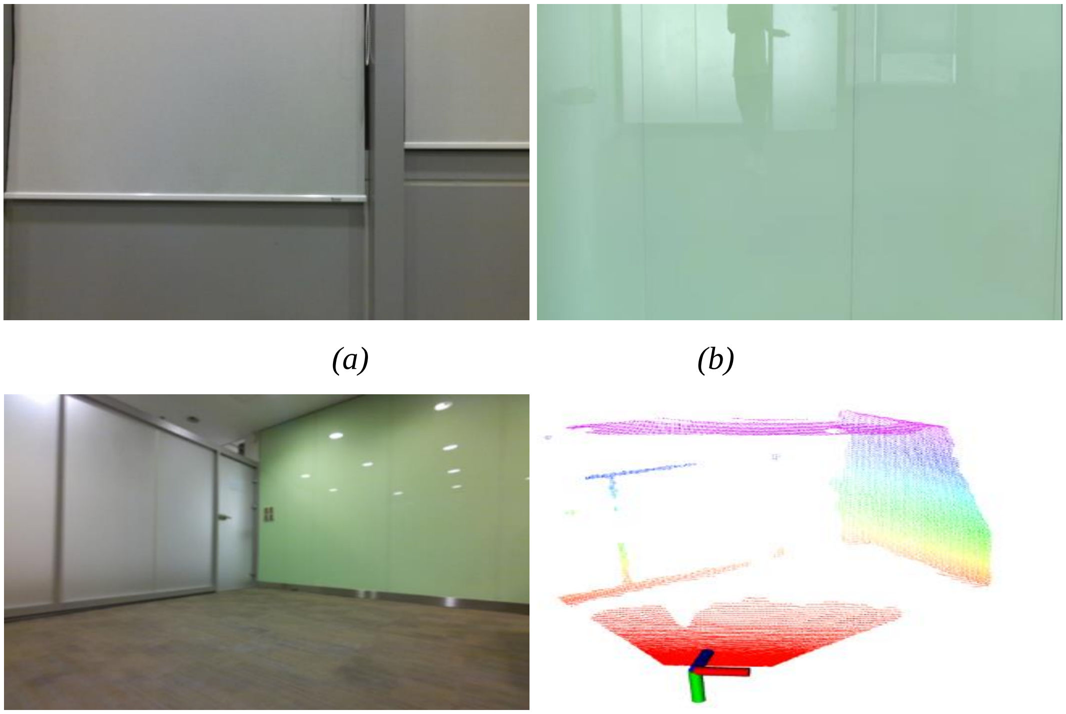
Table 6.2 lists the PTPD RMSE of the 3D models by the six methods, and their accuracies are defined as the ratio of PTPD RMSE to the length. The accuracy of the comprehensive system (RMSE: 8.3 cm, and RMSE/length: 0.136%) is the highest owing to sensor fusion and hybrid feature fusion. The line features, plane features, and the wheel odometer can all improve the performance of the feature point-based method, and the biggest improvement is resulted by wheel odometer fusion (from 34.5 cm to 9.6 cm), which is followed by the plane fusion (from 34.5 cm to 15.3 cm) and then the line fusion (from 34.5 cm to 21.0 cm). Compared with the feature point-based system, the highest improvement of the mapping accuracy is 75.9% by the comprehensive system, followed by 72.1% from the system fusing the wheel odometer and the RGB-D camera, fusing point, line, and plane features can improve the accuracy by 62.6%, and the improvements by fusing plane and line features are 55.6% and 39.1%, respectively.

Table 6. 2: Comparison of PTPD RMSE (cm) in corridor experiment.

Corridor (60.792 m)	**P**	**PL**	**PP**	**PLP**	**WP**	**WPLP**
PTPD RMSE	34.5	21.0	15.3	12.9	9.6	**8.3**
accuracy	0.567%	0.345%	0.251%	0.212%	0.157%	**0.136%**

6.3.3 Highly low textured room

We collect a new dataset in a highly low textured room. As shown in Figure 6.10, it covers white curtains, walls, and glass window with invalid depth. It is difficult to detect sufficient point features in this room. In Figure 6.11, the trajectory and 3D octomap of the comprehensive SLAM system in the highly low textured room are presented. The ground vehicle starts and ends at the same position. The loop closing modules of SLAM systems are disabled for the evaluation of the closing errors.



(a) *(b)*

(c) *(d)*

Figure 6. 10: Highly low textured room. (a) whitle curtain; (b) wall; (c) galss window; (d) invalid depth.

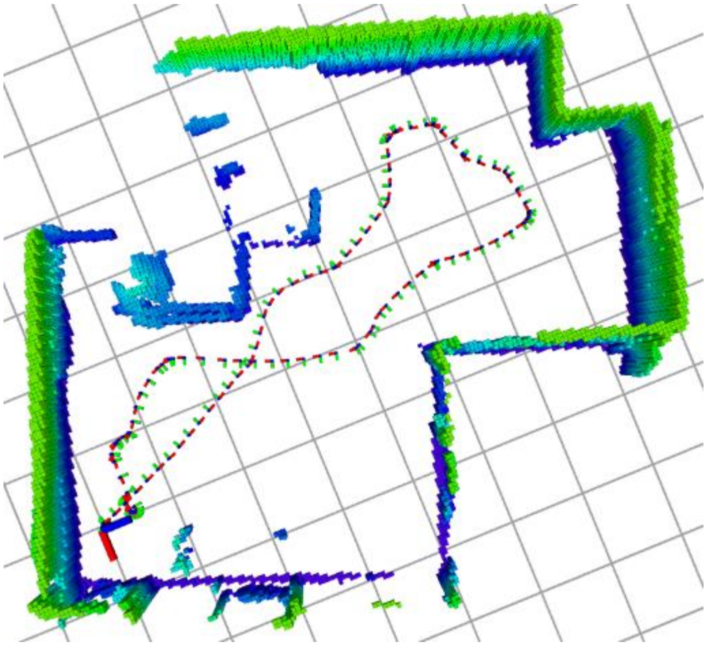


Figure 6. 11: 3D model and trajectory in the highly low textured room.

In Table 6.3, P and PL fail while others run successfully. Among them, the comprehensive method generates the smallest closing error, owing to hybrid feature fusion and wheel odometer integration. Its relative improvements over PP, PLP and WP are 79.0%, 67.3% and 39.3%, respectively. The trajectories of various SLAM systems are also shown in Figure 6.12. As shown in the zoomed figure, the smallest drift is achieved by the comprehensive SLAM system.

Table 6. 3: Comparison of closing errors(cm) in the highly low textured room.

Low texture (18. 325 m)	**P**	**PL**	**PP**	**PLP**	**WP**	**WPLP**
x-axis	-	-	3.8	2.9	0.1	**0.0**
y-axis	-	-	6.8	3.7	2.8	**1.7**
z-axis	-	-	2.3	2.2	0.0	**0.0**
closing error	-	-	8.1	5.2	2.8	**1.7**

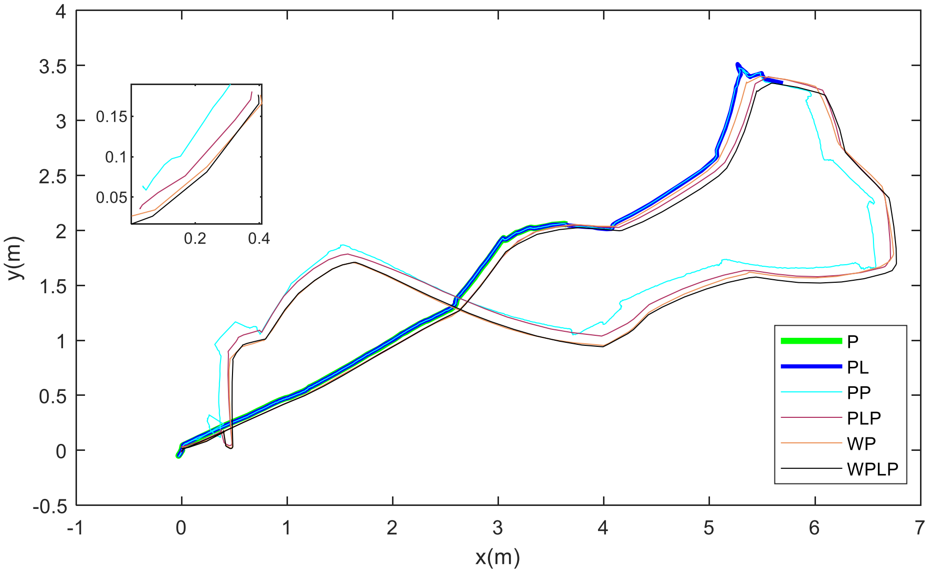
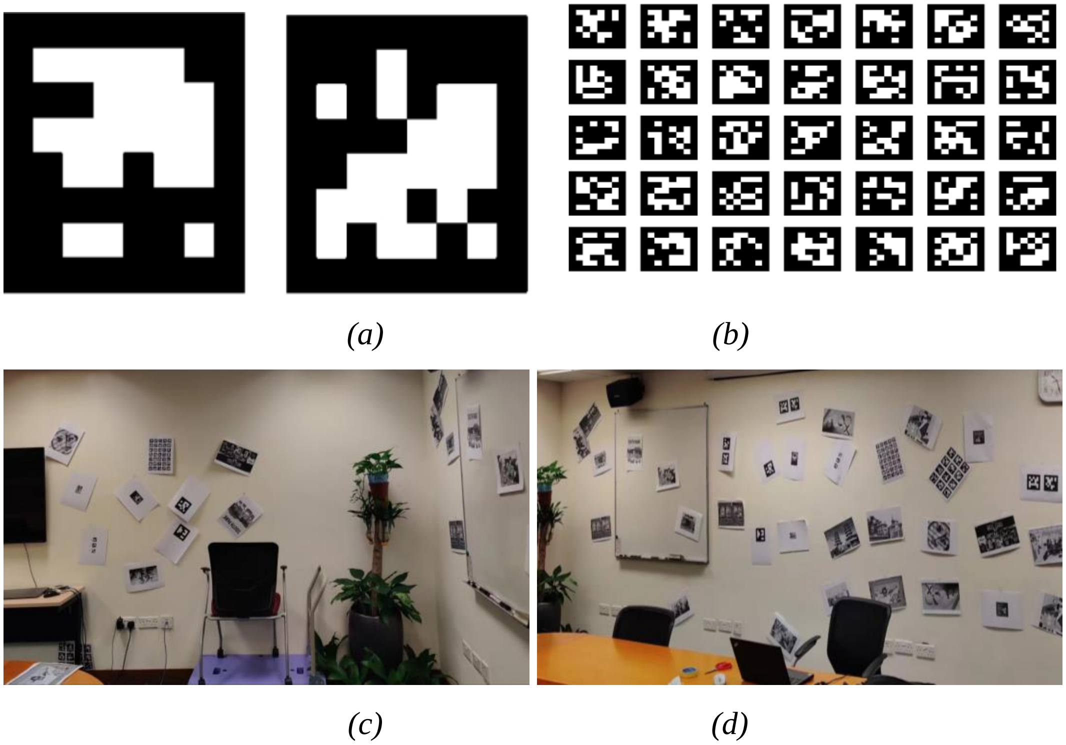


Figure 6. 12: Closing trajectories in the highly low textured room.

6.3.4 Highly rich textured room

In the last experiment, the ground vehicle moves in a highly rich textured room, where textured images are printed and sticked on the walls (Olson, 2011), as shown in Figure 6. 13. Figure 6.14 presents the trajectory and 3D octomap of the comprehensive method, where the start and end positions of the ground vehicle are the same.



(a) (b)

(c) (d)

Figure 6. 13: Textured images (a) and (b) on the walls (c) and (d).

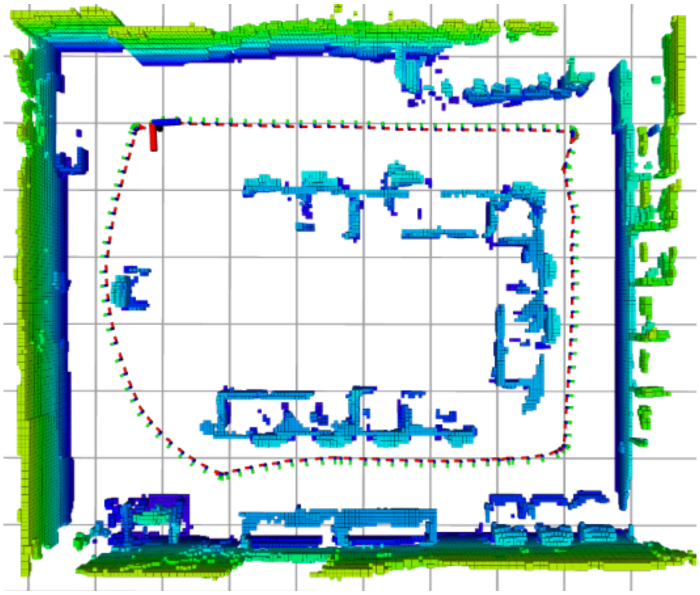


Figure 6. 14: 3D model and trajectory in the highly rich textured room.

In Table 6.4, all the methods run successfully in the high rich textured room because of sufficient point features. Among them, the smallest closing error is generated by the comprehensive method, and its relative improvement over others is at least 18.5%. Owing to hybrid feature fusion and wheel odometer integration, it can improve P by 86.4%, PL by 84.0%, PP by 74.1%, PP by 51.1%, and WP by 18.5%. Figure 6.15 presents the trajectories of six methods in the highly rich textured room. The zoomed figure also indicates the smallest accumulating drift by the comprehensive method.

Table 6. 4: Comparison of closing errors(cm) in the highly rich textured room.

Low texture (18. 325 m)	**P**	**PL**	**PP**	**PLP**	**WP**	**WPLP**
x-axis	12.9	5.1	-6.8	3.1	**0.7**	1.0
y-axis	2.9	10.1	-3.9	2.0	2.6	**2.0**
z-axis	9.4	6.5	3.3	2.6	**0.0**	**0.0**
closing error	16.2	13.0	8.5	4.5	2.7	**2.2**

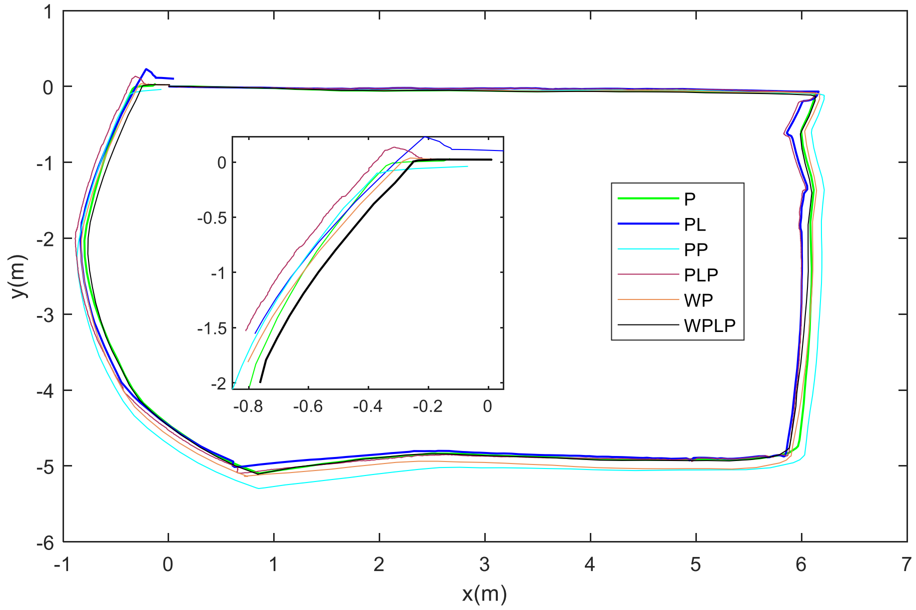


Figure 6. 15: Closing trajectories in the highly rich textured room.

6.3.5 Computation speed

We process the corridor dataset using the comprehensive system and list the processing time of each part in Table 6.5. The time cost by loop closing is not listed as it is highly dependent on the keyframe number, which is continuously growing. It should be noticed that the point, line, and plane features are extracted and matched on three independent threads, to speed up the pipeline and reduce the computation time.

Table 6. 5: Processing time (ms) of each part of the comprehensive system.

Thread	Part	Time
Tracking	Wheel Odometer Integration	1.0
	Planar Motion Assumption	1.0
	ORB Extraction LSD Extraction Fast Plane Extraction	62.3
	ORB Matching LBD Matching Plane Association	18.6
	Robust Pose Optimization	9.8
	Total	92.7
Local Mapping	Factor Graph Optimization	186.1
Octomap	Map Update	13.6

Though the system applies a multi-thread design, the computation burden still increases for processing the hybrid features. The average time cost of hybrid feature extraction is 62.3 ms, and that of hybrid feature matching is 18.6 ms. On the other hand, the extraction and matching times of the point features are only 24.1 ms and 7.5 ms, respectively. On average, the comprehensive system can complete the localization and mapping task within 106.3 ms (92.7+13.6). It is sufficient for a low-speed ground vehicle to perform path planning and obstacle avoidance. If the onboard computer has lower performance, both the processing functions of the line and plane features can be disabled to improve the computation speed and save the computation memory.

In addition, the average number of hybrid feature matches is shown in Table 6.4. Though the number of the line and plane feature matches is lower than that of the point feature matches, they are less affected by low textures and can provide additional and reliable constraints in low textured scenes.

Table 6. 6: Average number of hybrid feature matches.

Types	**Point Feature Matches**	**Line Feature Matches**	**Plane Feature Matches**
number	225.3	37.1	2.9

6.4 Summary

Based on the contributions listed in the last three chapters, this chapter presents a comprehensive RGB-D SLAM system with the aid of the wheel odometer and hybrid features, for ground vehicle navigation. In summary,

(a) It builds a comprehensive cost function, consisting of the constraints from hybrid features, wheel odometer, and planar motion assumption. It relies on more constraints and can generate higher localization accuracy than the previous methods in the last three chapters.

(b) It fuses point, line and plane features, combines wheel odometer constraints under the planar motion assumption, and then builds a comprehensive factor graph. Both 3D and 2D line features are utilized. The new representation type of the plane feature is also employed and the covariance of the plane measurements in the spherical form is computed. A two-stage strategy is used to handle the perturbations of the mobile platform. All the constraints are tightly coupled in the factor graph, which can further improve the accuracy of vehicle tracking and scene reconstruction.

(c) The comprehensive system is evaluated and compared with the modified versions using self-collected datasets in a lab room and a long corridor, which proves its superior performance of tracking and mapping. In the lab room experiment, the comprehensive system generates the highest localization accuracy 6.2 cm, which is followed by the system fusing the wheel odometer and the RGB-D camera (APE RMSE: 7.0 cm), the system fusing hybrid features (APE RMSE: 8.9 cm), the system fusing point and plane features (APE RMSE: 9.6 cm), the system fusing point and line features (APE RMSE: 13.8 m) and the point-based system (APE RMSE: 20.8 cm). In the corridor experiment, the highest mapping accuracy 8.3 cm is generated by the comprehensive system, which is followed by the system fusing the wheel odometer and the RGB-D camera (PTPD RMSE: 9.6 m), the system fusing hybrid features (PTPD RMSE: 12.9 m), the system fusing point and plane features (PTPD RMSE: 15.3 cm), the system fusing point and line features (PTPD RMSE: 21.0 cm) and the point-based system (PTPD RMSE: 32.5 cm). In the high low textured room, the point-based system and the system fusing point and line features fail, while the comprehensive method generates the smallest accumulating drift (closing error: 1.7 cm), which is followed by the system fusing the wheel odometer and the RGB-D camera (closing error: 2.8 cm), the system fusing hybrid features (closing error: 5.2 cm), and the system fusing point and plane features (closing error: 8.1 cm). In the highly rich textured room, all the methods run successfully. The smallest closing error is generated by the comprehensive method (2.2 cm), followed by the system fusing the wheel odometer and the RGB-D camera (2.7 cm), the system fusing hybrid features (4.5 cm), the system fusing point and plane features (8.5 cm), the system fusing point and line features (13.0 cm) and the point-based system (16.2 cm).

(d) The computation burdens increase due to the hybrid feature extraction and matching, though a multi-thread design is applied. Compared with the point-based system, the time cost of feature extraction increases from 24.1 ms to 62.3 ms, and that of feature matching increases from 7.5 ms to 18.6 cm. Above all, the proposed system can run a commercial CPU within about 10 FPS, which is sufficient for indoor low-speed applications.

CHAPTER 7

CONCLUSIONS AND FUTURE WORKS

7.1 Conclusions

RGB-D SLAM can localize the mobile platform and reconstruct the surrounding scenes simultaneously in unknown indoor environments. Compared with Lidars, RGB-D cameras are small in size, light in weight, and cheap in price. While monocular and stereo cameras require GPU for real-time processing, RGB-D cameras have the ability of real-time dense mapping on a commercial CPU. Therefore, utilizing RGB-D cameras is an attractive choice for mobile platform navigation in indoor scenes.

However, the continuity of RGB-D SLAM system is reduced under low textured scenes. While humans can avoid low textures during manual operation, the mobile platform lacks the ability to understand the environments and may fail to select a camera view with rich textures. Thus, RGB-D SLAM system may lose tracking in low textured scenes where few point features are detected and matched. Furthermore, the accuracy of RGB-D SLAM system is lowered because of the accumulating drift during mobile platform operation. The result of mobile platform tracking may fall into a local optimum if the initial guess is not accurate. The tracking accuracy is also reduced by images noises and incorrect feature matches.

Targeting on improving the continuity and accuracy of RGB-D SLAM for real-time indoor mobile platforms, this study is conducted with the following two main objectives:

(a) Fusing hybrid features to avoid tracking failure under low textured scenes.

(b) Combining the wheel odometer and the RGB-D camera to reduce the tracking drift.

Four major tasks were pursued and carried out in this thesis. The experiment results and discussions lead to the following specific conclusions:

(a) **A new RGB-D SLAM system fusing point and line features has been proposed.** The previous line-based methods either exploit 3D-3D or 3D-2D line correspondences and neglect part of line information. The new system combines both correspondences and develops a new cost function consisting of both 3D and 2D line reprojection errors. The new cost function can generate higher continuity and accuracy owing to more constraints utilized from line features.

In the experiments using TUM RGB-D datasets, the proposed system outperforms other SOTA methods in consideration of continuity and yields the same-level accuracy. In the lab room experiment, the proposed system can improve the localization accuracy of the method using 3D line features by 22.5%, and improve that of the method using 2D features by 25.8%. In the corridor experiments, the improvements of the mapping accuracy over the methods using 3D or 2D line features are 10.2% and 14.7, respectively.

(b) **A new RGB-D SLAM system fusing point and plane features has been proposed.** The previous plane-based methods use experimental weights for plane measurements, which are non-optimal for pose estimation. The new system derives the covariances of the plane features by plane fitting and covariance propagation. Point reprojection errors and plane transformation errors are combined optimally based on the derived covariances. In addition, the new system develops a new representation form for plane features based on the parallel and vertical relationships among planes and MW axes. The new representation form encodes the structural regularity and is employed in the factor graph optimization to further improve the tracking performance.

In the experiments using TUM RGB-D datasets, the proposed system generates higher localization accuracy than SOTA methods and achieves higher continuity than the feature point-based method. With the derived covariances of plane features, the proposed system can improve the localization and mapping accuracies of the method using experimental weights by 23.6% and 11.5%, respectively. The improvements of the proposed system owing to the new representation form are 27.6% and 8.8%, respectively.

(c) **A new localization and mapping system tightly coupling the RGB-D camera and the wheel odometer has been proposed.** The research works about fusing the RGB-D camera and the wheel odometer are few. They employ a hard assumption that the mobile platform moves on the ground plane without perturbations, which is not practical and non-optimal. Unlike them, the new system assumes the platform moves with perturbations on the floor and develops a two-stage strategy to handle the perturbations. In the frontend, the Mahalanobis distance test is used to detect large perturbations. In the backend, the coefficients of the ground plane are computed to constrain the mobile platform. To further improve the tracking accuracy, the new system proposes a tight-coupled factor graph, which consists of all the visual, wheel odometer, and planar motion constraints. Compared with the previous methods, it pays attention to the platform perturbations and exploits more motion constraints from the ground plane.

In real-world experiments, the proposed system is compared with the wheel odometer, the visual odometer, and a loose-coupled method using a hard planar motion assumption. It can outperform the other methods owing to the tight-coupled design and the two-stage strategy. Specifically, compared with the loose-coupled method using a hard planar motion assumption, it can improve the localization and mapping accuracies by 40.7% and 33.8%, respectively.

(d) **A comprehensive real-time RGB-D SLAM system by hybrid feature fusion and wheel odometer integration has been proposed.** In the comprehensive system, point, line, and plane features are simultaneously exploited from the RGB-D images, and they are then combined with the wheel odometer under the planar motion assumption.

The proposed system is evaluated by real-world experiments, and compared with other SLAM systems. By comparing with the feature point-based system, the proposed system can improve the localization accuracy by 70.1% and the mapping accuracy by 75.9%, utilizing the wheel odometer can improve the accuracies by 66.3% and 72.1%, hybrid feature fusion can improve them by 57.2% and 62.6%, fusing plane features can improve them by 53.8% and 55.6%, and the smallest improvements are 33.6% and 39.1% by fusing line features.

7.2 Future works

Here, the research interest is briefly stated:

(a) Robust tracking in dynamic scenes (Cui & Ma, 2019; Henein et al., 2020; Xiao et al., 2019; Yu et al., 2018). This thesis assumes that the mobile platform is moving in a static scene, so the features extracted from the dynamic objects may corrupt the quality of mobile platform tracking. To robustly estimate the pose of the mobile platform in dynamic scenes, dynamic objects should be identified and removed for feature extraction and matching. Furthermore, the point cloud belonging to the dynamic objects should not be inserted into the global map for navigation, and only the long-term static objects should remain.

(b) Seamless indoor and outdoor positioning system (Basiri et al., 2016; Cao et al., 2021; Chu et al., 2012; Li et al., 2019). This thesis assumes the mobile platform moves in indoor environments. The navigation task in large-scale outdoor environments cannot be fulfilled by the proposed system because it has no global reference. To seamlessly locate the mobile platform in indoor and outdoor environments, GNSS receivers should be equipped, which can provide reliable global positioning results in open-sky scenes. When the performance of GNSS degrades in indoor or clustered scenes, V-SLAM can be applied to compute the relative motion of the mobile platform.

REFERENCES

Agarwal, S., & Mierle, K. (2012). Ceres solver: Tutorial & reference. *Google Inc*, *2*(72), 8.

Albert, J. A., Owolabi, V., Gebel, A., Brahms, C. M., Granacher, U., & Arnrich, B. (2020). Evaluation of the pose tracking performance of the azure kinect and kinect v2 for gait analysis in comparison with a gold standard: A pilot study. *Sensors*, *20*(18), 5104.

Andersen, V., Aanæs, H., & Bærentzen, J. A. (2010). Surfel Based Geometry Reconstruction. *TPCG*, *10*, 39-44.

Aulinas, J., Petillot, Y., Salvi, J., & Lladó, X. (2008). The SLAM problem: a survey. *Artificial Intelligence Research and Development*, 363-371.

Barfoot, T. D. (2017). *State estimation for robotics*. Cambridge University Press.

Bartoli, A., & Sturm, P. (2005). Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer vision and image understanding*, *100*(3), 416-441.

Basiri, A., Amirian, P., Winstanley, A., Marsh, S., Moore, T., & Gales, G. (2016). Seamless pedestrian positioning and navigation using landmarks. *The Journal of Navigation*, *69*(1), 24-40.

Bleyer, M., Rhemann, C., & Rother, C. (2011). PatchMatch Stereo-Stereo Matching with Slanted Support Windows. Bmvc.

Bowman, S. L., Atanasov, N., Daniilidis, K., & Pappas, G. J. (2017). Probabilistic data association for semantic slam. 2017 IEEE international conference on robotics and automation (ICRA).

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc..

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., . . . Siegwart, R. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, *35*(10), 1157-1163.

Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., & Tardós, J. D. (2020). ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *arXiv preprint arXiv:2007.11898*.

Cao, S., Lu, X., & Shen, S. (2021). GVINS: Tightly Coupled GNSS-Visual-Inertial for Smooth and Consistent State Estimation. *arXiv e-prints*, arXiv: 2103.07899.

Chan, R. P. M., Stol, K. A., & Halkyard, C. R. (2013). Review of modelling and control of two-wheeled robots. *Annual reviews in control*, *37*(1), 89-103.

Chang, J.-R., & Chen, Y.-S. (2018). Pyramid stereo matching network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Chen, S., Chen, H., Chang, C.-W., & Wen, C.-Y. (2021). Multilayer Mapping Kit for Autonomous UAV Navigation. *IEEE Access, 9,* 31493-31503.

Chen, S., Wen, C.-Y., Zou, Y., & Chen, W. (2020). Stereo visual inertial pose estimation based on feedforward-feedback loops. *arXiv preprint arXiv:2007.02250*.

Cheng, Z., & Wang, G. (2018). Real-time rgb-d slam with points and lines. 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC).

Chu, T., Guo, N., Backén, S., & Akos, D. (2012). Monocular camera/IMU/GNSS integration for ground vehicle navigation in challenging GNSS environments. *Sensors*, *12*(3), 3162-3185.

Cignoni, P., Ranzuglia, G., Callieri, M., Corsini, M., Ganovelli, F., Pietroni, N., & Tarini, M. (2011). MeshLab.

Coiffet, P., & Chirouze, M. (2012). *An introduction to robot technology*. Springer Science & Business Media.

Cui, L., & Ma, C. (2019). SOF-SLAM: A semantic visual SLAM for dynamic environments. *IEEE Access*, *7*, 166528-166539.

Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, *27*(6), 647-665.

Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., & Theobalt, C. (2017). Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, *36*(4), 1.

Davison, A. J. (2005). Active search for real-time vision. Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1.

Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*, *29*(6), 1052-1067.

Di Febbo, P., Dal Mutto, C., Tieu, K., & Mattoccia, S. (2018). Kcnn: Extremely-efficient hardware keypoint detection with a compact convolutional neural network. Proceedings of the IEEE conference on computer vision and pattern recognition workshops.

Diaz, M. G., Tombari, F., Rodriguez-Gonzalvez, P., & Gonzalez-Aguilera, D. (2015). Analysis and evaluation between the first and the second generation of RGB-D sensors. *IEEE Sensors Journal*, *15*(11), 6507-6516.

Endres, F., Hess, J., Sturm, J., Cremers, D., & Burgard, W. (2013). 3-D mapping with an RGB-D camera. *IEEE transactions on robotics*, *30*(1), 177-187.

Engel, J., Koltun, V., & Cremers, D. (2017). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(3), 611-625.

Engel, J., Schöps, T., & Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. European conference on computer vision.

Engelhard, N., Endres, F., Hess, J., Sturm, J., & Burgard, W. (2011). Real-time 3D visual SLAM with a hand-held RGB-D camera. Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden.

Feng, C., Taguchi, Y., & Kamat, V. R. (2014). Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. 2014 IEEE International Conference on Robotics and Automation (ICRA).

Filipenko, M., & Afanasyev, I. (2018). Comparison of various slam systems for mobile robot in an indoor environment. 2018 International Conference on Intelligent Systems (IS).

Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2016). On-Manifold Preintegration for Real-Time Visual--Inertial Odometry. *IEEE Transactions on robotics*, *33*(1), 1-21.

Fu, Q., Yu, H., Lai, L., Wang, J., Peng, X., Sun, W., & Sun, M. (2019). A robust RGB-D SLAM system with points and lines for low texture indoor environments. *IEEE Sensors Journal*, *19*(21), 9908-9920.

Fuchs, S., & Hirzinger, G. (2008). Extrinsic and depth calibration of ToF-cameras. 2008 IEEE Conference on Computer Vision and Pattern Recognition.

Gálvez-López, D., & Tardos, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on robotics*, *28*(5), 1188-1197.

Gao, F., Wu, W., Gao, W., & Shen, S. (2019). Flying on point clouds: Online trajectory generation and autonomous navigation for quadrotors in cluttered environments. *Journal of Field Robotics*, *36*(4), 710-733.

Glocker, B., Izadi, S., Shotton, J., & Criminisi, A. (2013). Real-time RGB-D camera relocalization. 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR).

Glocker, B., Shotton, J., Criminisi, A., & Izadi, S. (2014). Real-time RGB-D camera relocalization via randomized ferns for keyframe encoding. *IEEE transactions on visualization and computer graphics*, *21*(5), 571-583.

Gomez-Ojeda, R., Moreno, F.-A., Zuniga-Noël, D., Scaramuzza, D., & Gonzalez-Jimenez, J. (2019). PL-SLAM: A stereo SLAM system through the combination of points and line segments. *IEEE Transactions on robotics*, *35*(3), 734-746.

Grisetti, G., Kümmerle, R., Strasdat, H., & Konolige, K. (2011). g2o: A general framework for (hyper) graph optimization. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China.

Guo, R., Peng, K., Fan, W., Zhai, Y., & Liu, Y. (2019). RGB-D SLAM Using Point–Plane Constraints for Indoor Environments. *Sensors*, *19*(12), 2721.

Han, X.-F., Jin, J. S., Wang, M.-J., Jiang, W., Gao, L., & Xiao, L. (2017). A review of algorithms for filtering the 3D point cloud. *Signal Processing: Image Communication, 57*, 103-112.

Harris, C. G., & Stephens, M. (1988). A combined corner and edge detector. Alvey vision conference.

He, Y., Zhao, J., Guo, Y., He, W., & Yuan, K. (2018). Pl-vio: Tightly-coupled monocular visual–inertial odometry using point and line features. *Sensors*, *18*(4), 1159.

Henein, M., Zhang, J., Mahony, R., & Ila, V. (2020). Dynamic SLAM: The need for speed. 2020 IEEE International Conference on Robotics and Automation (ICRA).

Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2012). RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International journal of robotics research*, *31*(5), 647-663.

Horn, B. K., & Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*, *17*(1-3), 185-203.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous robots*, *34*(3), 189-206.

Hosseinzadeh, M., Latif, Y., & Reid, I. (2017). Sparse Point-plane Slam. Australasian Conference on Robotics and Automation.

Hsiao, M., Westman, E., & Kaess, M. (2018). Dense planar-inertial slam with structural constraints. 2018 IEEE International Conference on Robotics and Automation (ICRA).

Hsiao, M., Westman, E., Zhang, G., & Kaess, M. (2017). Keyframe-based dense planar SLAM. 2017 IEEE International Conference on Robotics and Automation (ICRA).

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., . . . Davison, A. (2011). KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. Proceedings of the 24th annual ACM symposium on User interface software and technology.

Jeong, W. Y., & Lee, K. M. (2006). Visual SLAM with line and corner features. 2006 IEEE/RSJ international conference on intelligent robots and systems.

Ji, P., Zeng, M., & Liu, X. (2018). Geometric primitives based rgb-d slam for low-texture environment. Proceedings of the 31st International Conference on Computer Animation and Social Agents,

Joo, K., Oh, T.-H., Kweon, I. S., & Bazin, J.-C. (2019). Globally optimal inlier set maximization for atlanta world understanding. *IEEE transactions on pattern analysis and machine intelligence*, *42*(10), 2656-2669.

Kaess, M. (2015). Simultaneous localization and mapping with infinite planes. 2015 IEEE International Conference on Robotics and Automation (ICRA),

Kaess, M., & Dellaert, F. (2009). Covariance recovery from a square root information matrix for data association. *Robotics and Autonomous Systems*, *57*(12), 1198-1210.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, *31*(2), 216-235.

Kähler, O., Prisacariu, V. A., & Murray, D. W. (2016). Real-time large-scale dense 3D reconstruction with loop closure. European Conference on Computer Vision.

Kaneko, A. M., & Ichinose, R. (2019). Stata Center Frame: A Novel World Assumption for Self-Localization. 2019 4th International Conference on Robotics and Automation Engineering (ICRAE).

Kang, R., Xiong, L., Xu, M., Zhao, J., & Zhang, P. (2019). VINS-Vehicle: A Tightly-Coupled Vehicle Dynamics Extension to Visual-Inertial State Estimator. 2019 IEEE Intelligent Transportation Systems Conference (ITSC).

Kerl, C., Sturm, J., & Cremers, D. (2013). Dense visual SLAM for RGB-D cameras. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Khoshelham, K., & Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, *12*(2), 1437-1454.

Kim, P., Coltin, B., & Jin Kim, H. (2018). Linear RGB-D SLAM for planar environments. Proceedings of the European Conference on Computer Vision (ECCV),

Kim, P., Coltin, B., & Kim, H. J. (2018a). Linear RGB-D SLAM for planar environments. Proceedings of the European Conference on Computer Vision (ECCV).

Kim, P., Coltin, B., & Kim, H. J. (2018b). Low-drift visual odometry in structured environments by decoupling rotational and translational motion. 2018 IEEE international conference on Robotics and automation (ICRA).

Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. 2007 6th IEEE and ACM international symposium on mixed and augmented reality.

Klein, G., & Murray, D. (2009). Parallel tracking and mapping on a camera phone. 2009 8th IEEE International Symposium on Mixed and Augmented Reality,

Labbé, M., & Michaud, F. (2019). RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, *36*(2), 416-446.

Lemaire, T., & Lacroix, S. (2007). Monocular-vision based SLAM using line segments. Proceedings 2007 IEEE International Conference on Robotics and Automation.

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. 2011 International conference on computer vision.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2015). Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, *34*(3), 314-334.

Li, M. (2014). *Visual-inertial odometry on resource-constrained systems*. UC Riverside.

Li, M., & Mourikis, A. I. (2012). Improving the accuracy of EKF-based visual-inertial odometry. 2012 IEEE International Conference on Robotics and Automation.

Li, M., & Mourikis, A. I. (2013). High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research*, *32*(6), 690-711.

Li, T., Zhang, H., Gao, Z., Niu, X., & El-Sheimy, N. (2019). Tight fusion of a monocular camera, MEMS-IMU, and single-frequency multi-GNSS RTK for precise navigation in GNSS-challenged environments. *Remote Sensing*, *11*(6), 610.

Li, Y., Brasch, N., Wang, Y., Navab, N., & Tombari, F. (2020a). Structure-SLAM: Low-Drift Monocular SLAM in Indoor Environments. *IEEE Robotics and Automation Letters*, *5*(4), 6583-6590.

Li, Y., Brasch, N., Wang, Y., Navab, N., & Tombari, F. (2020b). Structure-SLAM: Low-Drift Monocular SLAM in Indoor Environments. *arXiv preprint arXiv:2008.01963*.

Li, Y., Li, W., Darwish, W., Tang, S., Hu, Y., & Chen, W. (2020). Improving Plane Fitting Accuracy with Rigorous Error Models of Structured Light-Based RGB-D Sensors. *Remote Sensing*, *12*(2), 320.

Li, Y., Yunus, R., Brasch, N., Navab, N., & Tombari, F. (2020). RGB-D SLAM with Structural Regularities. *arXiv preprint arXiv:2010.07997*.

Ligocki, A., & Jelínek, A. (2019). Fusing the RGBD SLAM with Wheel Odometry. *IFAC-PapersOnLine*, *52*(27), 7-12.

Ling, Y., & Shen, S. (2019). Real-time dense mapping for online processing and navigation. *Journal of Field Robotics*, *36*(5), 1004-1036.

Liu, H., Li, C., Chen, G., Zhang, G., Kaess, M., & Bao, H. (2017). Robust keyframe-based dense SLAM with an RGB-D camera. *arXiv preprint arXiv:1711.05166*.

Liu, J., Gao, W., & Hu, Z. (2019). Visual-inertial odometry tightly coupled with wheel encoder adopting robust initialization and online extrinsic calibration.

2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics*, *21*(4), 163-169.

Lourenço, F., & Araujo, H. (2021). Intel RealSense SR305, D415 and L515: Experimental Evaluation and Comparison of Depth Estimation. VISIGRAPP (4: VISAPP).

Lowe, D. G. (1999). Object recognition from local scale-invariant features. Proceedings of the seventh IEEE international conference on computer vision.

Lu, Y., & Song, D. (2015). Robust RGB-D odometry using point and line features. Proceedings of the IEEE International Conference on Computer Vision.

Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision.

Ma, L., Kerl, C., Stückler, J., & Cremers, D. (2016). CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM. 2016 IEEE International Conference on Robotics and Automation (ICRA).

Muglikar, M., Zhang, Z., & Scaramuzza, D. (2020). Voxel map for visual slam. 2020 IEEE International Conference on Robotics and Automation (ICRA).

Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP*, *2*(331-340), 2.

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on robotics*, *31*(5), 1147-1163.

Mur-Artal, R., & Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, *33*(5), 1255-1262.

NavVis M6. (2018). https://www.navvis.com/m6

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., . . . Fitzgibbon, A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. 2011 10th IEEE international symposium on mixed and augmented reality.

Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. 2011 international conference on computer vision.

Nießner, M., Zollhöfer, M., Izadi, S., & Stamminger, M. (2013). Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, *32*(6), 1-11.

Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. 2011 IEEE international conference on robotics and automation.

Proença, P. F., & Gao, Y. (2018). Probabilistic RGB-D odometry based on points, lines and planes under depth uncertainty. *Robotics and Autonomous Systems*, *104*, 25-39.

Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A., & Moreno-Noguer, F. (2017). PL-SLAM: Real-time monocular visual SLAM with points and lines. 2017 IEEE international conference on robotics and automation (ICRA).

Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on robotics*, *34*(4), 1004-1020.

Qualisys, A. (2006). Qualisys track manager user manual.

Qualisys, A. J. U. h. w. q. s., accessed on. (2008). Qualisys motion capture systems. 04-04.

Quan, M., Piao, S., Tan, M., & Huang, S.-S. (2019). Tightly-coupled Monocular Visual-odometric SLAM using Wheels and a MEMS Gyroscope. *IEEE Access*, *7*, 97374-97389.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., . . . Ng, A. Y. (2009). ROS: an open-source Robot Operating System. ICRA workshop on open source software.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. European conference on computer vision.

Rubio, F., Valero, F., & Llopis-Albert, C. (2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, *16*(2), 1729881419839596.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. 2011 International conference on computer vision.

Rusu, R. B., & Cousins, S. (2011). 3d is here: Point cloud library (pcl). 2011 IEEE international conference on robotics and automation.

Segal, A., Haehnel, D., & Thrun, S. (2009). Generalized-icp. Robotics: science and systems.

Shi, J. (1994). Good features to track. 1994 Proceedings of IEEE conference on computer vision and pattern recognition.

Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT press.

Slabaugh, G., Schafer, R., Malzbender, T., & Culbertson, B. (2001). A survey of methods for volumetric scene reconstruction from photographs. Volume Graphics 2001.

Smith, C. (2006). On vertex-vertex systems and their use in geometric and biological modelling.

Straub, J., Rosman, G., Freifeld, O., Leonard, J. J., & Fisher, J. W. (2014). A mixture of manhattan frames: Beyond the manhattan world. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Sturm, J., Burgard, W., & Cremers, D. (2012). Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark. Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS).

Taguchi, Y., Jian, Y.-D., Ramalingam, S., & Feng, C. (2013). Point-plane SLAM for hand-held 3D sensors. 2013 IEEE international conference on robotics and automation, Tang, S., Chen, W., Wang, W., Li, X., Darwish, W., Li, W., . . . Guo, R. (2018). Geometric integration of hybrid correspondences for RGB-D unidirectional tracking. *Sensors*, *18*(5), 1385.

Tang, T. J. J., Lui, W. L. D., & Li, W. H. (2011). A lightweight approach to 6-dof plane-based egomotion estimation using inverse depth. Australasian Conference on Robotics and Automation.

Tareen, S. A. K., & Saleem, Z. (2018). A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. 2018 International conference on computing, mathematics and engineering technologies (iCoMET),

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *13*(04), 376-380.

Von Gioi, R. G., Jakubowicz, J., Morel, J.-M., & Randall, G. (2012). LSD: a line segment detector. *Image Processing On Line*, *2*, 35-55.

Vu, Q.-H., Kim, B.-S., & Song, J.-B. (2008). Autonomous stair climbing algorithm for a small four-tracked robot. 2008 International Conference on Control, Automation and Systems.

Wang, J., Shi, Z., & Zhong, Y. (2017). Visual SLAM incorporating wheel odometer for indoor robots. 2017 36th Chinese Control Conference (CCC).

Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., & McDonald, J. (2012). Kintinuous: Spatially extended kinectfusion.

Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., & Davison, A. (2015). ElasticFusion: Dense SLAM without a pose graph.

Whelan, T., Salas-Moreno, R. F., Glocker, B., Davison, A. J., & Leutenegger, S. (2016). ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research*, *35*(14), 1697-1716.

Whitty, M., Cossell, S., Dang, K. S., Guivant, J., & Katupitiya, J. (2010). Autonomous navigation using a real-time 3d point cloud. 2010 Australasian Conference on Robotics and Automation.

Widya, A. R., Torii, A., & Okutomi, M. (2018). Structure from motion using dense CNN features with keypoint relocalization. *IPSJ Transactions on Computer Vision and Applications*, *10*(1), 1-7.

Wieber, P.-B., Tedrake, R., & Kuindersma, S. (2016). Modeling and control of legged robots. In *Springer handbook of robotics* (pp. 1203-1234). Springer.

Wu, K. J., Guo, C. X., Georgiou, G., & Roumeliotis, S. I. (2017). Vins on wheels. 2017 IEEE International Conference on Robotics and Automation (ICRA).

Wu, Y., & Hu, Z. (2006). PnP problem revisited. *Journal of Mathematical Imaging and Vision*, *24*(1), 131-141.

Xiao, L., Wang, J., Qiu, X., Rong, Z., & Zou, X. (2019). Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robotics and Autonomous Systems, 117*, 1-16.

Yang, D., Bi, S., Wang, W., Yuan, C., Qi, X., & Cai, Y. (2019). DRE-SLAM: dynamic RGB-D encoder SLAM for a differential-drive robot. *Remote Sensing*, *11*(4), 380.

Yang, S., Song, Y., Kaess, M., & Scherer, S. (2016). Pop-up slam: Semantic monocular plane slam for low-texture environments. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Yang, Y., Geneva, P., Zuo, X., Eckenhoff, K., Liu, Y., & Huang, G. (2019). Tightly-coupled aided inertial navigation with point and plane features. 2019 International Conference on Robotics and Automation (ICRA).

Yang, Y., & Huang, G. (2018). Aided inertial navigation with geometric features: Observability analysis. 2018 IEEE International Conference on Robotics and Automation (ICRA).

Yu, C., Liu, Z., Liu, X.-J., Xie, F., Yang, Y., Wei, Q., & Fei, Q. (2018). DS-SLAM: A semantic visual SLAM towards dynamic environments. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Zhang, G., Lee, J. H., Lim, J., & Suh, I. H. (2015). Building a 3-D line-based map using stereo SLAM. *IEEE transactions on robotics*, *31*(6), 1364-1377.

Zhang, L., & Koch, R. (2013). An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency.

Journal of Visual Communication and Image Representation, *24*(7), 794-805.

Zhang, S., Xie, L., & Adams, M. D. (2005). Entropy based feature selection scheme for real time simultaneous localization and map building. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Zhang, X., Wang, W., Qi, X., Liao, Z., & Wei, R. (2019). Point-plane slam using supposed planes for indoor environments. *Sensors*, *19*(17), 3795.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, *19*(2), 4-10.

Zhang, Z., Dong, P., Wang, J., & Sun, Y. (2020). Improving S-MSCKF with variational Bayesian adaptive nonlinear filter. *IEEE Sensors Journal*, *20*(16), 9437-9448.

Zhao, Y., & Vela, P. A. (2018). Good feature selection for least squares pose optimization in VO/VSLAM. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Zheng, F., & Liu, Y.-H. (2019). Visual-Odometric Localization and Mapping for Ground Vehicles Using SE (2)-XYZ Constraints. 2019 International Conference on Robotics and Automation (ICRA).

Zheng, F., Tang, H., & Liu, Y.-H. (2018). Odometry-vision-based ground vehicle motion estimation with se (2)-constrained se (3) poses. *IEEE transactions on cybernetics*, *49*(7), 2652-2663.

Zhou, Q.-Y., Park, J., & Koltun, V. (2018). Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*.

Zhou, Y., Kneip, L., Rodriguez, C., & Li, H. (2016). Divide and conquer: Efficient density-based tracking of 3D sensors in Manhattan worlds. Asian Conference on Computer Vision.

Zhou, Y., Li, H., & Kneip, L. (2018). Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d–2-d edge alignment. *IEEE Transactions on robotics*, *35*(1), 184-199.

Zuñiga-Noël, D., Ruiz-Sarmiento, J.-R., Gomez-Ojeda, R., & Gonzalez-Jimenez, J. (2019). Automatic multi-sensor extrinsic calibration for mobile robots. *IEEE Robotics and Automation Letters*, *4*(3), 2862-2869.

Zuo, X., Xie, X., Liu, Y., & Huang, G. (2017). Robust visual SLAM with point and line features. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).