



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**Place Recognition and Localization for
Multi-Robot SLAM**

Liu Xiangyu

SCHOOL OF ELECTRICAL AND ELECTRONIC

ENGINEERING

MASTER OF SCIENCE IN COMPUTER CONTROL AND

AUTOMATION

2019

Contents

Abstract	iii
Acknowledgement	iv
Lists of Figures	vi
Lists of Tables	vii
1 Introduction	1
1.1 Background	1
1.2 Motivation and Objectives	2
1.3 Major contribution of the Dissertation	3
1.4 Organisation of the Dissertation	3
2 Literature Review	5
2.1 Visual SLAM	5
2.1.1 Introduction	5
2.1.2 ORB-SLAM	6
2.1.3 Multi-robot SLAM	8
2.1.4 CORB-SLAM	9
2.2 Illumination Variance	11
2.2.1 Appearance Change From Illumination	11
2.2.2 Formulation	11
2.2.3 Application of Localization	13

3 Approach	15
3.1 Quantitative Trajectory Evaluation Method	15
3.2 CORBSLAM with Illumination Variance	18
4 Test and Experiments	20
4.1 Datasets	20
4.1.1 KITTI Visual Odometry Dataset	20
4.1.2 Oxford RobotCar Dataset	21
4.1.3 NTU Dataset	24
4.2 Evaluation of CORBSLAM	26
4.2.1 KITTI Datasets	26
4.2.2 NTU Datasets	31
4.3 Evaluation under different illumination	35
5 Discussion	39
5.1 Results of multi ground robots cluster	39
5.2 Failure Reasons and Drawbacks of Illumination Variance Method	40
6 Conclusions and Future Work	44
6.1 Conclusions	44
6.2 Future Work	44
Appendix: Example Code	50

Abstract

SLAM (Simultaneous Localization and Mapping) is an important component of robot systems, enabling robot systems to exploit and map an unknown environment, while estimating its own position. Single robot SLAM techniques based on monocular or stereo cameras, color or depth images and segmentation- or feature-based algorithms, have been considerably developed with many successful solutions proposed, during years of research. Compared to single robot SLAM, multi robot SLAM have several advantage e.g. robustness and quicker exploration. However, the development of multi robot SLAM is much slower, because of the difficulties such as determination of relative poses of robots and communication, etc. CORB-SLAM as a multi robot SLAM system based on ORB-SLAM2, inheriting advantages of ORB-SLAM2 like low-cost sensors and low computation cost, has not been evaluated in a quantitative method to assess its performance. For multi robot SLAM, the adaptability in life-long scenarios is another important research topic, since multi robots may run in different dates and seasons. This work quantitatively evaluates CORB-SLAM, and experiments to combine it with illumination variance algorithm to improve its performance under different illumination and season.

Keywords: Multi-robot SLAM, CORB-SLAM, Quantitative Trajectory Evaluation, Life-long SLAM.

Acknowledgement

This dissertation is finished under the guidance of PhD students e.g. Zhang Jun, Peng Guohao, Yue Yufeng and Professor Wang Danwei, in ST Engineering-NTU ROBOTICS ADVANCE LABORATORY in NTU. I sincerely thank them and all other members in the lab for their patient and careful help during the implementation, experiment and compilation of the dissertation. Without their help on providing technical suggestions and possible solutions to the research objectives, and help during the collection of the related datasets, this dissertation project would not go so smoothly within two short semesters.

List of Figures

1.1	A demonstration of input and mapping results of SLAM systems	2
2.1	ORB-SLAM system overview.	7
2.2	The structure of CORB-SLAM system.	9
2.3	The flowchart of Map Fusing module.	10
2.4	Appearance changes caused by different lighting conditions. Pictures are selected from St Lucia dataset recording on 10/09/2009 8:45 am and 2:10 pm	12
2.5	Illumination invariance result images in St Lucia dataset. It shows how this approach suppress the effects caused by sun	13
2.6	A graphic demonstration about how ABLE works.	13
2.7	Block-flow diagram of the combined stereo localisation approach.	14
3.1	Mapping results of CORB-SLAM in [1].	16
3.2	Illustration of relative error.	18
3.3	Keypoints extracted from rgb images and illumination variance images.	19
4.1	Example image in KITTI Visual Odometry 2012 dataset.	21
4.2	Example image in robotcar dataset.	22
4.3	The robotcar platform and sensor location diagram.	23
4.4	Comparison of images captured in the same location in different seasons in RobotCar dataset.	24
4.5	Ground truth information of each rosbags in NTU Dataset.	24
4.6	Overview picture of NTU Husky platform.	25
4.7	Example images of NTU dataset.	25

4.8	Ground truth trajectory of partial and complete sequences of KITTI Datasets	27
4.9	Quantitative evaluation results of CORB-SLAM client mapping the entire KITTI Sequence 00.	27
4.10	Quantitative evaluation results of CORB-SLAM client mapping KITTI partial Seq.0.	28
4.11	Quantitative evaluation results of CORB-SLAM client mapping KITTI partial Seq.1.	28
4.12	Mapping results of the entire sequence without partial sequence.	29
4.13	Mapping results of Seq.0 and Seq.1, and the map fusion results of KITTI Datasets.	30
4.14	Quantitative evaluation results of fused map of KITTI Datasets.	30
4.15	Ground truth trajectory of partial and complete bags of NTU Datasets.	32
4.16	Mapping results of Bag.0 and Bag.1 and the map fusion result of server.	33
4.17	Quantitative evaluation results of mapping Bag.0 of NTU Datasets.	33
4.18	Quantitative evaluation results of mapping Bag.1 of NTU Datasets.	34
4.19	Quantitative evaluation results of fused map of NTU Datasets.	34
4.20	Image sequence with overexposed frames in RobotCar dataset.	35
4.23	Map fusion results of Seq.0 and Seq.1 without ground truth in Oxford RobotCar Datasets.	36
4.21	Ground truth individual and overall trajectories of Seq.0 and Seq.1 in Oxford RobotCar Datasets.	37
4.22	Mapping results of Seq.0 and Seq.1 and the map fusion result of server in Oxford RobotCar Datasets.	38
5.2	ORB Keypoint matching results of raw and illumination variance images in Oxford RobotCar Datasets.	41
5.1	Generated illumination variance images of example raw images in Oxford RobotCar Datasets.	42

List of Tables

4.1	Information of datasets used	20
4.2	Main characteristics of the rosbags in NTU Dataset used in the experiment.	24
4.3	Quantitative results of mapping unseparated Sequence 00.	26
4.4	Quantitative results of mapping Seq.0.	28
4.5	Quantitative results of mapping Seq.1.	29
4.6	Quantitative results of map fusion evaluation on KITTI partial sequences.	29
4.7	Quantitative results of mapping evaluation on Bag.0 NTU Datasets.	31
4.8	Quantitative results of mapping evaluation on Bag.1 NTU Datasets.	32
4.9	Quantitative results of map fusion evaluation on NTU Datasets.	34
4.10	Partial datasets selected in RobotCar dataset.	36

Chapter 1

Introduction

1.1 Background

SLAM(Simultaneous localization and mapping) is a key component in mobile autonomous systems. It describes the ability of a vehicle, once placed in an unknown environment, to explore and map that environment, while at the same time estimating its own position in the environment, using only its onboard sensing capabilities.

SLAM systems can be accomplished by both single or multiple robots. Multiple-robot SLAM or MRSLAM, offer several advantages compared to there single robot counterpart, for example:

- Robustness to single robot failure,
- Quicker exploration of environments in time critical SaR(Search and Rescue) mission.

However, Adapting SLAM technology to multiple-robot scenario brings some new changes as identified by Saeedi et al [2]:

- Relative Poses of Robots. The map created by each robot in its own reference frame in multiple-robot SLAM is called the local map. It is a difficult task to merge all the local client maps created by each robot to create a globally consistent map of the environment, as the required transformation or alignment matrices that relate these maps to each other are generally unknown.



(a) Input image and extracted keypoints.



(b) Mapping result.

Figure 1.1: A demonstration of input and mapping results of SLAM systems

- **Closing Loops.** Loop closure is defined as identifying a previously observed but not very recent place. Solving this problem for a multiple robots cluster requires taking advantage of all information resources from each client. In multi-robot SLAM, different events can trigger loop closure, such as direct or indirect robots encounter, when robots see the same area of features worldwide.
- **Communications.** Multi-robot SLAM requires a medium to be highly available for data sharing between robots. It is possible to exchange information between robots via communication channels. The quality of the channels of communication depends on the environment. Communication, for example, is a challenging issue for a robots team in underwater environments, where the environment imposes limitations on data rate and bandwidth.

Because of the limitation of the difficulties mentioned above, the development of multiple-robot SLAM is much slower than single-robot SLAM. Finding a solution to these problem will push the adaption of SLAM technology to a new level.

1.2 Motivation and Objectives

Recently, some solutions for multi-robot SLAM systems have been proposed. One of them is CORB-SLAM proposed in [1]. As its clients based on ORB-SLAM, it inherits major advantages of ORB-SLAM e.g. relatively low computational cost and low cost of sensors. But in its initial work, the accuracy of its map fusion results are

not evaluated in a quantitative evaluation method, while only a rough demonstration of its mapping results is given. Therefore, quantitative evaluation of CORB-SLAM with mapping results of single-robot ORB-SLAM system and CORB-SLAM clients given for comparison can help understand the performance of this multi-robot slam algorithm and assess the feasibility of its potential applications.

Another critical requirement of applications of multi-robot slam algorithms is its ability to fuse sub maps under different illumination conditions and season, which is named as life-long application circumstances. Illumination variance method proposed in [3] with advantages of easy implement and low computational cost, may be able to help CORB-SLAM to deal with illumination and season changes. Therefore related experiments are undertaken in this work in order to find a effective way to combine CORB-SLAM with illumination variance.

1.3 Major contribution of the Dissertation

1. Evaluation of CORB-SLAM on several datasets with quantitative trajectory evaluation results provided.
2. Experiment of combination of CORB-SLAM with illumination variance algorithm to test whether illumination variance method is suitable to enhance ability of CORB-SLAM to deal with illumination changes.

1.4 Organisation of the Dissertation

This dissertation is organised into several chapters:

1. Chapter 2 briefly outlines the development of visual SLAM technique. Firstly, the classic structure of visual SLAM system is introduced, and the critical algorithms involved are elaborated, so as the classifications of visual SLAM systems. Then two main algorithms involved in this work, ORB-SLAM and CORB-SLAM, are demonstrated. This chapter also explores prior work in shade dealing algorithms required to implement life-long SLAM.

2. Chapter 3 explains the methodology used in this dissertation to evaluate map fusion performance of CORB-SLAM, and how to combine illumination variance method with CORB-SLAM system to test if illumination variance can be utilized to enhance the ability of CORB-SLAM to deal with illumination changes.
3. Chapter 4 shows the results of
 - (a) the evaluation of CORB-SLAM on selected 2 datasets including (i) KITTI Visual Odometry Dataset [4], (ii) NTU Dataset [5],
 - (b) the evaluation of CORB-SLAM combined with illumination variance on Oxford RobotCar Dataset [6].
4. Chapter 5 analysis the results demonstrated in chapter 4 in detail, discussing the improvement and the disadvantages.
5. Chapter 6 concludes the work done in this dissertation, and comments on some limitation and drawbacks of algorithms used in this work, which future work need focus on.

Chapter 2

Literature Review

2.1 Visual SLAM

2.1.1 Introduction

Simultaneous Localizaiton and Mapping (SLAM) is a technique for obtaining an unknown environment's 3D structure and sensor movement in the environment. Following years of development, SLAM-based application has become widespread, such as 3D modeling based on computer vision, visualization based on augmented reality(AR) and self-driving cars.

In early SLAM algorithms, there exit many different modalities of sensors integrated in SLAM systems, such as rotary encoders, light detection and ranging radar (LiDAR), inertial sensors, GPS and cameras. In recent years, SLAM using cameras only, specifically referred to as visual SLAM (vSLAM), has been actively discussed because the sensor configuration is simple, low-cost, and contains abundant information. But meanwhile this technique also brings more difficulties than others using integrated sensors [7].

vSLAM algorithms have proposed widely in the field of computer vision, robotics and AR. The low requirement on the modalities of sensors, requiring cameras only, is the major advantage of vSLAM technique, so that it is very suitable for low-cost unmanned vehicles, robots with limited load capacity and power supply like drones, or mobile devices such as camera-mounted tablets or smart phones.

However, the difficulties brought by vSLAM can not be ignored. Instead of obtaining depth and location information directly from LiDAR, GPS or depth camera in integrated SLAM systems, vSLAM technique needs to compute all these information from color or gray images, which reduces stability and accuracy for several estimation steps involved in this process. Also obviously the computational cost are significantly higher. Therefore, the problem of how to improve the performance and reduce computational cost of vSLAM has always been widely concerned.

2.1.2 ORB-SLAM

One of the state-of-the-art vSLAM solutions for single-robot systems is ORB-SLAM, initially proposed in [8], and upgraded to a second version in [9].

ORB-SLAM is a monocular SLAM system based on features that maps in real time small or large, indoor or outdoor environments. The new vision has been upgraded to ORB-SLAM2 in [9] and can process monocular, stereo and RGB-D inputs. ORB-SLAM is based on the main ideas of PTAM, the place recognition algorithm of Gálvez-López and Tardós in the proposed work in [8], ORB-SLAM is based on the main ideas of PTAM, the place recognition algorithm of Gálvez-López and Tardós [10], the use of covisibility information for large-scale operation [12], [13] and the scale-aware loop closure of Strasdat et. Al Scale [?]. As a novel monocular SLAM system, the main contributions of ORB-SLAM are:

1. In all tasks, the same features are used: tracking, mapping, closing of loops and relocation. Making the system more efficient, reliable and simple with the same features. And using ORB features enables GPU-free real-time performance, with good invariance to changes in lighting and viewpoint.
2. In large environments, real-time performance. Due to the advantage of using a covisibility graph, the tracking and mapping modules are focused in a local visible area, independent of the global map.
3. Loop closing in real time. Optimizing a pose graph called the Essential Graph is adapted to perform the closing performance of the real-time loop. The Essential

Graph is built from links to loop closures, strong edges from the covisibility graph, and a system maintained spanning tree.

4. Real time camera relocation with exceptional viewpoint and illumination invariance. This improves the reuse of maps and also allows recovery from failure tracking.
5. A new robust and automatic initialization procedure based on model selection that creates an initial map of planar and non-planar scenes.
6. A survival of the most appropriate approach to keyframe selection and map point, generous in spawning but very restrictive in culling. This policy enhances service life and improves the robustness of tracking because redundant key frames are discarded.

ORB-SLAM system, see on Figure 2.1, incorporates three threads that run in parallel: tracking, local mapping and loop closing.

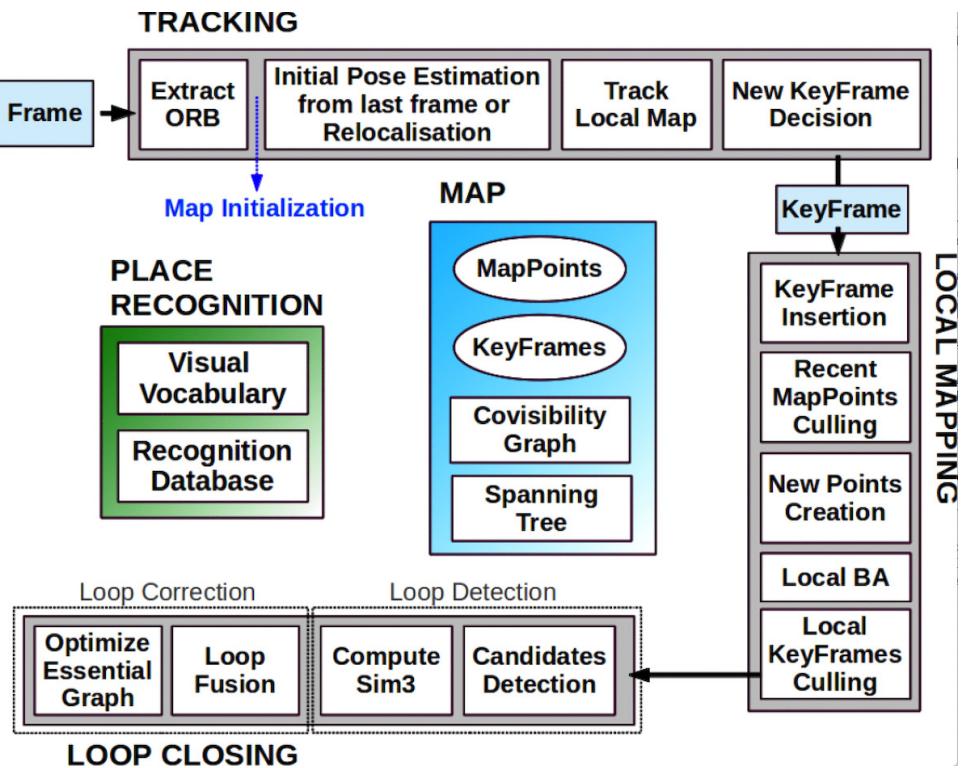


Figure 2.1: ORB-SLAM system overview.

2.1.3 Multi-robot SLAM

Multi-robot SLAM (MR-SLAM) uses several collaborative robots to map an unknown environment as an extension of single-robot SLAM systems. However, the extension can not be easily implemented as each client maps their surroundings on their own local frame of reference. Therefore, in order to merge the sub map of several clients into a consistent global map, the map fusion server needs to be able to connect and find the transformation between different local client reference frames. The environment can be mapped significantly faster using MR-SLAM technique on a cluster of multi ground or hybrid robots and the uncertainty in the system can be reduced due to data redundancy. Furthermore, if separate robots are equipped with heterogeneous sensors, the fused global map may contain much more feature information than a single robot map. Another obvious advantage is the robustness of one or more robots to fail, particularly when data is decentralized.

As concluded in [14], besides the advantages above, MR-SLAM systems also face problems and choices in the following aspects:

1. Centralization or Decentralization. Determining where data processing takes place is one of the main questions building MR-SLAM systems. Data is processed in centralized approaches on a central server, responsible for the information fusion of all robots and for distributing global map information to clients to improve their ability to self-locate [1, 15, 16]. On the other hand, decentralized MR-SLAM systems usually have several interconnected agents, managed by a coordinating structure that allows systems to easily and efficiently deal with increasing numbers of agents [17, 18].

2. Communication and information sharing. Decentralized data processing and consistent global map sharing have high requirements for communication bandwidth, latency and coverage area. Not only the connection characteristics e.g. connection architecture and protocol, but also the transmitting period and content etc. need to be selected carefully when designing MR-SLAM software systems.

2.1.4 CORB-SLAM

Proposed by F.Li et al. in [1], CORB-SLAM is centralized visual MR-SLAM systems based on ORB-SLAM2. As presented in Figure 2.2, the system of CORB-SLAM consists of robot-end clients running ORB-SLAM2 modified to transmit map information via ROS and a central server responsible to fuse maps and resend global maps.

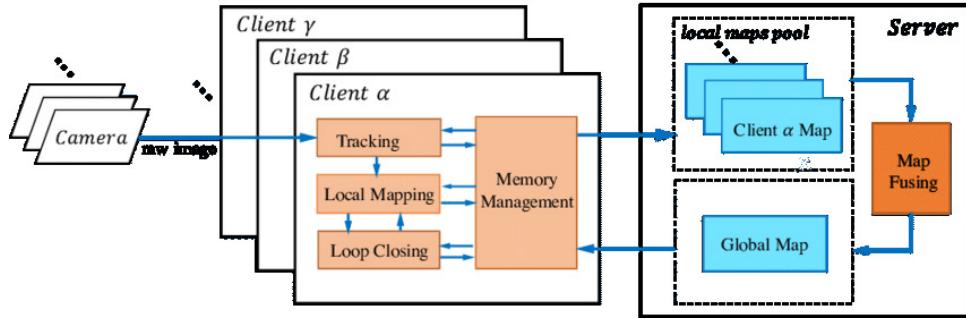


Figure 2.2: The structure of CORB-SLAM system.

CORB-SLAM Client

The robot-end client of CORB-SLAM is an ORB-SLAM client extended to have the functionality to communicate with the server, transmitting the local map information, with all functions and modules in original ORB-SLAM as listed in Chapter 2.1.2 reserved.

Map Fusion in the Server

The server's map fusing module receives and fuses the clients' local maps to achieve an optimized global map. Figure ?? shows the map fusion algorithm, which includes three main parts: global map initialization, map overlap detection, and local map fusion.

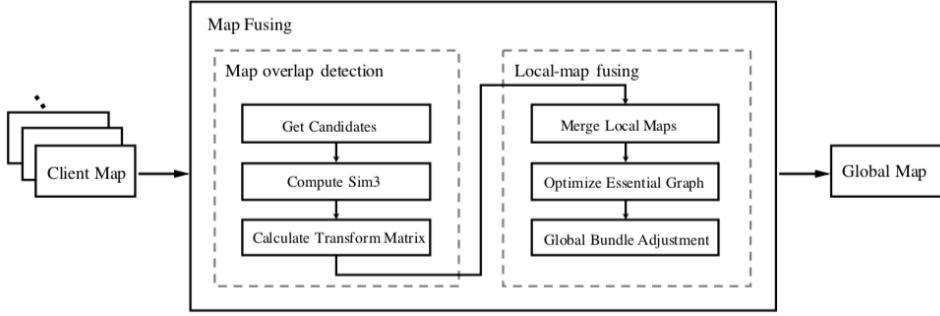


Figure 2.3: The flowchart of Map Fusing module.

1. Initializing global map

As the server system starts initially, the global map is set empty. As clients keep mapping and creating their local maps, the server receive data of local maps. After a first local map reaches the server, it will set as the initial global, based on which the global reference frame is decided.

2. Map overlap detection

To calculate relations between local maps, the server firstly detects the overlaps among local maps and then computes the transform matrices using Perspective-n-Point method.

The map overlap detection module follows the same solution of ORB-SLAM2, which firstly extract keypoints from input images using ORB features, then compute similarities between images by distance between image descriptors based on Bag of Binary Words (DBoW3) approach. Then, after map overlaps are detected, RANSAC iterations are adopted to calculate the 7 degrees-of-freedom (DoF) transform matrix between the current keyframe and all rough candidates detected. With enough inliers, a similar candidate, K_α is optimized. If it is supported by enough inliers, the overlap with K_α is accepted with the pose of K_α , T_{wc} calculate at the same time during optimization.

3. Local-map fusing

Local maps are merged into the consistent global after an acceptable transform matrix has been obtained. Using Bundle Adjustment (BA), a global 7 DoF optimization is then performed to suppress map errors caused by various SLAM clients.

After all process finished, the fused global map is resent to clients, and each client continues its own work to detect loop closure and correct loops on the received global map. If the global map is updated by computation on clients, the changes will be uploaded to server to update the map, and further transmitted to other clients by the server.

2.2 Illumination Variance

2.2.1 Appearance Change From Illumination

Facing appearance changes is an ongoing challenge for vision systems concerned with locating in known environments. Changes in appearance can result from several sources, for example, (i) different lighting conditions, (ii) varying weather conditions, and (iii) dynamic objects such as pedestrians or vehicles.

In previous work by Colin McManus et al., they showed how to leverage knowledge of previous 3D structure to suppress distracting objects for improved pose estimation in busy urban environments [19], and how to cope with long-term variation in appearance caused by changing weather conditions [20]. They proposed a new approach to addressing problem in [3] called the Illumination Variance Approach.

Appearance change caused by different lighting conditions in (i) is illustrated in Figure 2.4 with pictures selected from St Lucia dataset [21]. Compared to approaches proposed in [19] and [20], illumination variance approach is not model-based, requiring less computational cost. And in most of applications of vSLAM, appearance changes caused by (i) are a more common problem than (ii, iii). Therefore, how to combine illumination variance approach with multi-robot SLAM algorithms, to improve the performance of place recognition in changing illumination conditions, is the major objective focused on in this work.

2.2.2 Formulation

Illumination variance approach proposed in [3], is a simple method based on only one equation computing illumination variant images. The basic idea of this approach is to



(a) pic1.



(b) pic2.

Figure 2.4: Appearance changes caused by different lighting conditions. Pictures are selected from St Lucia dataset recording on 10/09/2009 8:45 am and 2:10 pm

map color images to an illumination invariant color space, where illumination change caused by different lighting condition like shade can be suppressed. The mapping equation is presented in Equation 2.1.

$$I = \log(R) - \alpha \log(G) - (1 - \alpha) \log(B) \quad (2.1)$$

Where, R, G, B are the raw image color channels and I is the resulting invariant image illumination. As shown in ??, α is a parameter that depends on each color channel's peak spectral responses ($\lambda_R, \lambda_G, \lambda_B$), normally available in camera specifications.

$$\frac{1}{\lambda_R} = \frac{\alpha}{\lambda_G} + \frac{1 - \alpha}{\lambda_B} \quad (2.2)$$

Therefore, considering the peak spectral responses, α can be easily calculated as exposed in Equation 2.3.

$$\alpha = \frac{\left(\frac{\lambda_B}{\lambda_G} - \frac{\lambda_B}{\lambda_R} \right)}{\left(1 - \frac{\lambda_B}{\lambda_R} \right)} \quad (2.3)$$

The influence of applying the illumination invariant transformation is showed in Figure 2.5.



(a) pic1.

(b) pic2.

Figure 2.5: Illumination invariance result images in St Lucia dataset. It shows how this approach suppress the effects caused by sun

2.2.3 Application of Localization

An open source toolbox called OpenABLE is implemented in [22] for lifelong visual localization. The proposed implementation in [22] uses the philosophy of the topological location recognition approach called ABLE introduced in [23–25] using illumination variant images for relocation.

A graphic representation about how the methodology proposed by OpenABLE is showed in Figure 2.6.

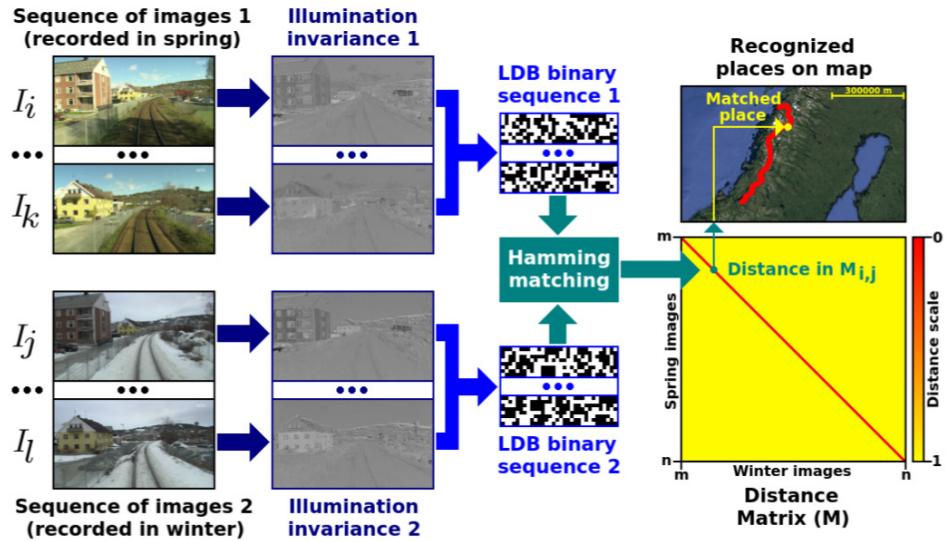


Figure 2.6: A graphic demonstration about how ABLE works.

The limitation of illumination variance approach is the transformation process produces resultant images with low resolution because all pixel values are turned into log values. These low-resolution resultant images still can be used as the input images of

visual topological localization where high resolution images are actually not needed. But in the mapping task, illumination variant images are too blurry to estimate camera motion and then reconstruct the map. Therefore, to improve the mapping performance in changing illumination conditions, rgb images and illumination variant images are both needed to perform relocalization and mapping, as the block-flow proposed in [26] presented in 2.7.

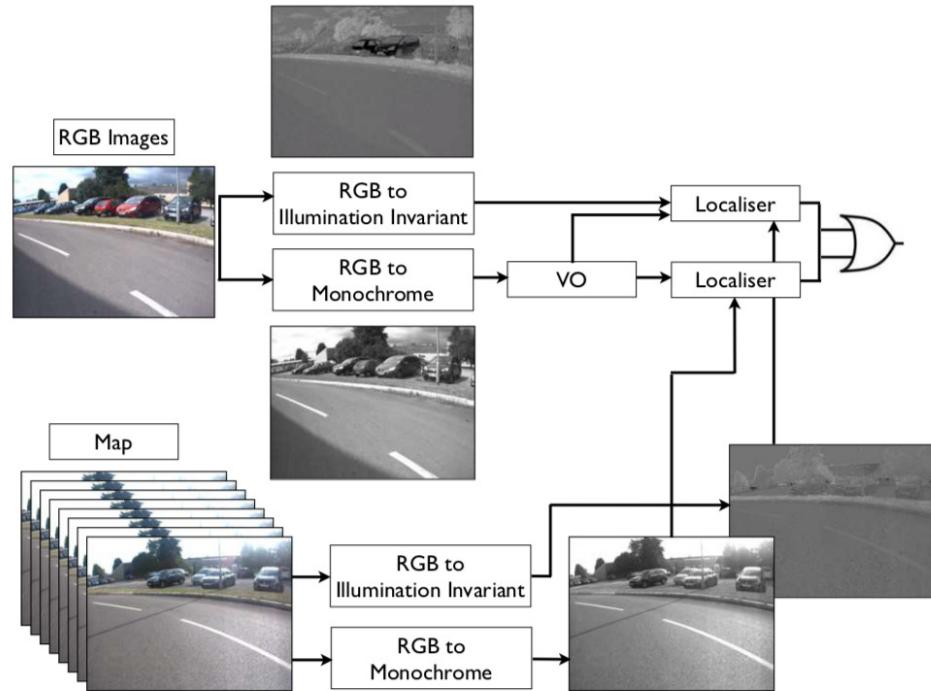


Figure 2.7: Block-flow diagram of the combined stereo localisation approach.

In the framework presented in 2.5, there is a second localizer making use of illumination invariant images in parallel with the main localization system. In [3], although the metric relative poses calculated from illumination variant images tends to be more noisy, the integrated localizer are less likely to fail if the scene appearance change is due to sunlight intensity direction or spectrum variation.

Chapter 3

Approach

3.1 Quantitative Trajectory Evaluation Method

To evaluate the mapping performance, the proposed method in [27] is modified to multi robot case, and employed in this work.

The previous work of CORB-SLAM in [1] only provides a rough overview of the mapping result of the multi robot system, as seen in Figure 3.1.

In [3] and [22], the results of the illumination variance localization are also only briefly introduced, with no quantitative results given.

In this paper, mapping results of CORB-SLAM and CORB-SLAM integrated with illumination variance are evaluated following the quantitative trajectory evaluation method proposed in [27]. Quantitative evaluation results of each datasets are demonstrated in several figures and tables including contents as follows, and see Section 4.2.1 as an example:

1. Ground truth trajectories of each partial sequence and the complete dataset for reference, e.g. Figure 4.8.
2. Mapping results of each client and fused map in server end, compared with ground truth trajectories, e.g. Figure 4.13.
3. Four charts of quantitative results, e.g. Figure 4.14, including
 - 1). Chart of relative translation error in meter, e.g. Figure 4.14(a).

- 2). Chart of relative translation error in percent, e.g. Figure 4.14(b).
- 3). Chart of relative yaw error in degree, e.g. Figure 4.14(c).
- 4). Chart of scale error in percent, e.g. Figure ??.
4. A table presenting numeric results of charts in 3, e.g. Table 4.6.
5. Charts of quantitative results of mapping each partial sequences in each client, in the same format of 3, e.g. Figure 4.10.
6. A table presenting numeric results of charts in 5, e.g. Table 4.4.
7. (if applicable) The mapping results of CORB-SLAM mapping the entire sequence without partial sequences for reference, e.g. Figure 4.12.
8. (if applicable) Charts of quantitative results of mapping the entire sequence for reference, in the same format of 3, e.g. Figure 4.9.
9. (if applicable) A table presenting numeric results of charts in 8, e.g. Table 4.3.

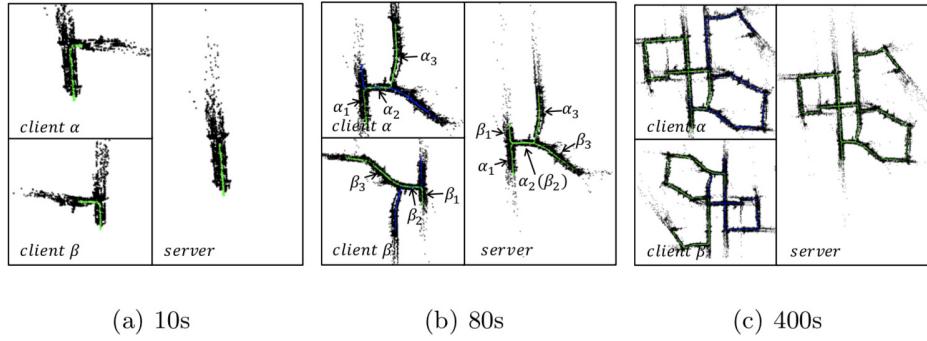


Figure 3.1: Mapping results of CORB-SLAM in [1].

The main task of map fusion module of multi robot SLAM server is to find relative relations including rotation and transformation matrices between client maps, based on which client maps are fused into a consist global map. Therefore, inaccurate rotation and transformation result in an inaccurate fused global map where all client maps but the one set to be the initial global map are stitched dramatically offsetting the ground

truth trajectories, which therefore causes serious translation and yaw error. Therefore in charts and tables of quantitative results, three types of relative errors are selected:

1. Relative Translation Error.
2. Relative Translation Error in percent.
3. Relative Yaw Error.

Relative error computation follows the basic idea stated in [27] that estimation quality can be measured by computing relative relationships between states at different times, as there is no global reference in VO systems, including global position and yaw.

Formally, given the ground truth trajectory \mathbf{X} and the estimated trajectory $\hat{\mathbf{X}}$:

$$\mathbf{X} = \{\mathbf{x}_i\} = \{R_i, p_i, v_i\}, \quad i = 1, \dots, t. \quad (3.1)$$

where, $p_i \in \Re^3$ is the position, $R_i \in SO(3)$ is the rotation matrix and $v_i \in \Re^3$ is the velocity of the system.

the following steps should be taken to explain and calculate relative errors:

Step 1. Given the ground truth positions p_i and estimated positions \hat{p}_i , the estimated trajectory can be aligned to the aligned estimated trajectory $\hat{\mathbf{X}}'$ by finding a similarity transformation $S' = \{s', R', t'\}$ that minimize:

$$S' = \arg \min_{S=\{s,R,t\}} \sum_{i=0}^{N-1} \|p_i - sR\hat{p}_i - t\|^2. \quad (3.2)$$

Then,

$$\hat{R}'_i = R'\hat{R}'_i, \quad \hat{p}'_i = s'R'\hat{p}'_i + t', \quad \hat{v}'_i = s'R'\hat{v}_i. \quad (3.3)$$

Step 2. A set of K pairs of states, each of which defines a sub trajectory, need to be selected from $\hat{\mathbf{X}}$ by a criteria e.g. distance or duration traveled:

$$\boldsymbol{\xi} = \{\mathbf{d}_k\}_{k=0}^{K-1}, \quad \mathbf{d}_k = \{\hat{\mathbf{x}}_s, \hat{\mathbf{x}}_e\}, \quad e > s \quad (3.4)$$

Step 3. Then for each \mathbf{d}_k , a relative error $\delta\mathbf{d}_k$ is calculated by:

$$\begin{aligned}
\delta \mathbf{d}_k &= \{\delta \phi, \delta p_k, \delta v_k\}, \\
\delta \phi_k &= \angle \delta R = \angle R_e(\hat{R}'_e)^\top, \\
\delta p_k &= \|p_e - \delta R_k \hat{p}'_e\|_2, \\
\delta v_k &= \|v_e - \delta R_k \hat{v}'_e\|_2.
\end{aligned} \tag{3.5}$$

Step 4. Collecting the relative errors in Equation 3.5 for all pairs in ξ gives:

$$\begin{aligned}
RE_{rot} &= \{\delta \phi_k\}_{k=1}^{K-1}, \\
RE_{pos} &= \{\delta p_k\}_{k=0}^{K-1}, \\
RE_{vel} &= \{\delta v_k\}_{k=0}^{K-1},
\end{aligned} \tag{3.6}$$

which is illustrated in Figure 3.2. relative translation errors in meter and percent can be extracted from RE_{pos} , and relative yaw errors from RE_{rot} .

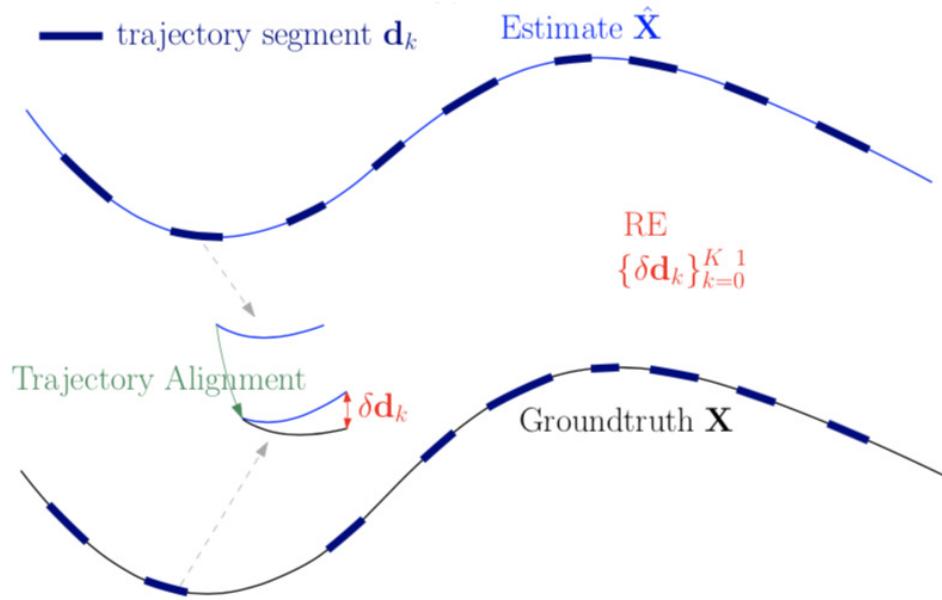
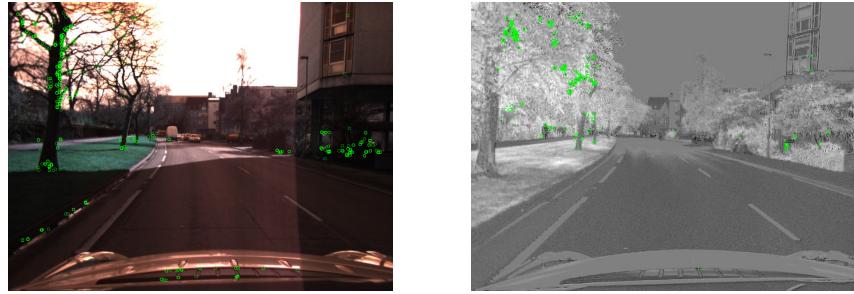


Figure 3.2: Illustration of relative error.

3.2 CORBSLAM with Illumination Variance

To experiment whether illumination variance method is possible to be utilized to enhance the ability of CORB-SLAM to map in different illumination conditions and seasons, it is combined into the map fusion modules of CORB-SLAM server in this work.

In the client end, the following modifications are made:



(a) Keypoints extracted from an rgb image. (b) Keypoints extracted from an illumination variance image.

Figure 3.3: Keypoints extracted from rgb images and illumination variance images.

1. A new thread running in parallel is added to process the input frame to transform into illumination variance images, and extract ORB keypoints in produced images (named as *II keypoints* in this paper), as illustrated in Figure 3.3.
2. II keypoint data is added into each frame as new member variables. And then following the CORB-SLAM methodology, only the integrated keyframes are transmitted to the server, which are serialized and packed by boost serialization library, and transmitted through ROS service.

Besides the above changes, the following modification are made in the server end:

1. A new keydataset containing II keypoint information.
2. A new illumination variance localizer running in parallel with the rgb localizer. When processing the input keyframe, a new localizer thread is started if the rgb localizer returns no result. The results from rgb and illumination variance localizer are added together in this work.
3. Since extracted II keypoint positions are not accurate as rgb keypoints, transform matrix is still trying to calculate from rgb keypoints where overlapping is detected.

Chapter 4

Test and Experiments

4.1 Datasets

Evaluation are performed in several datasets including KITTI Visual Odometry 2012 dataset [4], Oxford RobotCar dataset [6] and NTU dataset collected in NTU. The listing of used datasets is shown in Table 4.1

4.1.1 KITTI Visual Odometry Dataset

KITTI Visual Odometry 2012 is part of the [4, 28] KITTI Vision Benchmark Suite. KITTI data sets are captured by driving around a city of medium size, in rural areas and on highways. The recording platform is equipped with two stereo camera systems of high resolution capturing color and gray images, a Velodyne HDL-64E LIDAR and an OXTS RT 3003 localization system combining GPS, GLONASS, IMU and RTK correction signals.

KITTI Visual Odometry Evaluation 2012 provides 11 sequences with ground truth trajectories for training, and another 11 sequences without ground truth for evaluation.

Table 4.1: Information of datasets used

Datasets	Settings	Approx Scale	Diversity
KITTI	rural area	< 1 hour	one city, one weather condition, daytime
Oxford	city	214 hours	one city, multiple weather conditions, daytime
NTU	campus	< 1 hour	one campus (NTU), one weather condition

Example images are shown in Figure 4.1.



Figure 4.1: Example image in KITTI Visual Odometry 2012 dataset.

4.1.2 Oxford RobotCar Dataset

Oxford RobotCar Dataset is presented by Will Maddern et al. in [6], as a challenging dataset for autonomous driving. Collected over the period of May 2014 to December 2015, this datasets recorded images from 6 cameras mounted Nissan LEAF, along with LIDAR, GPS and INS ground truth. Images were recorded under different weather and illumination condition from 9:00 to 16:00 on average, from May to December. Example images is shown in Figure 4.2.



Figure 4.2: Example image in robotcar dataset.

The RobotCar platform is a Nissan LEAF equipped with sensors as following [6]:

1. Stereo Camera: Bumblebee XB3 trinocular stereo camera $\times 1$, 1/3" Sony ICX445 CCD, $1280 \times 960 \times 3$, 16Hz, 3.88mm lens, 66° HFoV, 12/24cm baseline, global shutter.
2. Monocular Camera: Grasshopper2 $\times 3$, 2/3" ICX285 CCD, 1024×1024 , 11.1Hz, 2.67mm fisheye lens, 180° , global shutter.
3. 2D LIDAR SICK LMS-151 2D LIDAR $\times 2$, 270° FoV, 50Hz, 50m range, 0.5° resolution.
4. 3D LIDAR: SICK LD-MRS 3D LIDAR $\times 1$, 85° HFoC, 3.2° VFoV, 4 panes, 12.5Hz, 50m range, 0.125° resolution.
5. GPS/INS Module: NovAtel SPAN-CPT ALIGN inertial and GPS navigation system $\times 1$, 6 axis, 50Hz, GPS/GLONASS, dual antenna.

The RobotCar platform and the sensor locations are demonstrated in Figure 4.3.

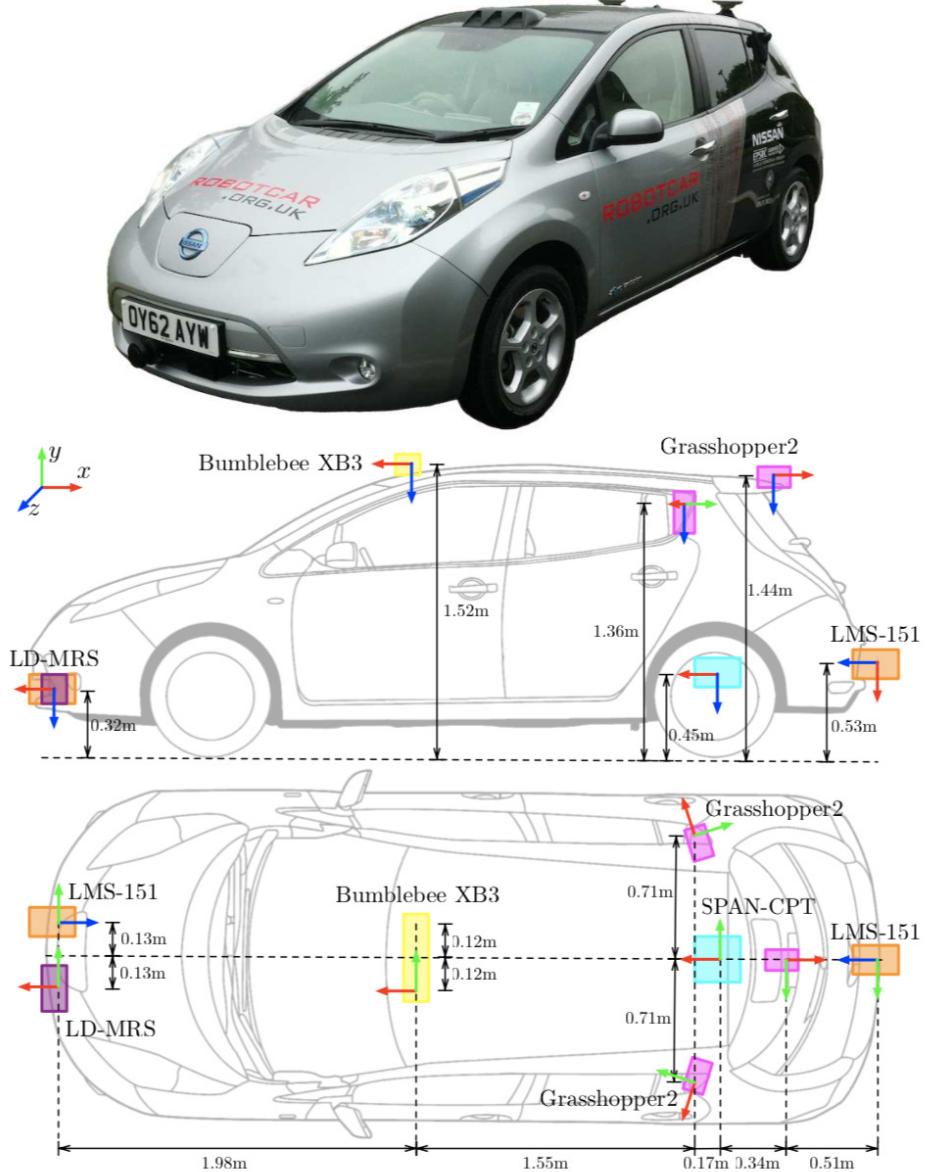
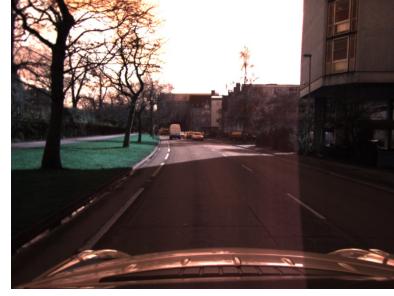


Figure 4.3: The robotcar platform and sensor location diagram.

RobotCar dataset is especially suitable to evaluate life-long SLAM systems, since it contains images taken in different hours of daytime under different illumination conditions, and in different seasons. The comparison between image in different illumination conditions and different seasons is shown in Figure 4.4.

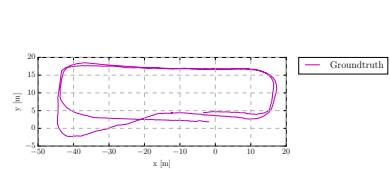


(a) Image captured in 14:49 07/14/2014.

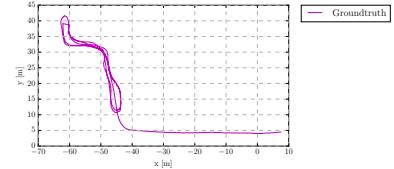


(b) Image captured in 12:32 02/24/2015.

Figure 4.4: Comparison of images captured in the same location in different seasons in RobotCar dataset.



(a) Ground truth trajectory of Bag.0.



(b) Ground truth trajectory of Bag.1.

Figure 4.5: Ground truth information of each rosbags in NTU Dataset.

4.1.3 NTU Dataset

Our NTU dataset [5] is collected by multi ground robots consisting of two husky UGV platforms, recording driving around the carpark in front of School of EEE building.

Our UGV platform is a HUSKY Clearpath robot, equipped with a ZED stereo camera $\times 1$, 672×376 , 87° HFoV, 56° VFoV. The picture of the platform and example images are shown in Figure 4.6 and 4.7.

The dataset provides 4 rosbag files. 3 of them are recorded by UGV, while the other one is recorded by UAV. The basic information of 4 rosbags are listed in Table 4.2. And the ground truth trajectories are shown in Figure 4.5.

Table 4.2: Main characteristics of the rosbags in NTU Dataset used in the experiment.

Bag No.	Data(M/D/Y)	Platform	Height(m)	Dep. Angle
0	10/27/2018	UGV	$\approx 0.7m$	0°
1	10/27/2018	UGV	$\approx 0.7m$	0°



Figure 4.6: Overview picture of NTU Husky platform.

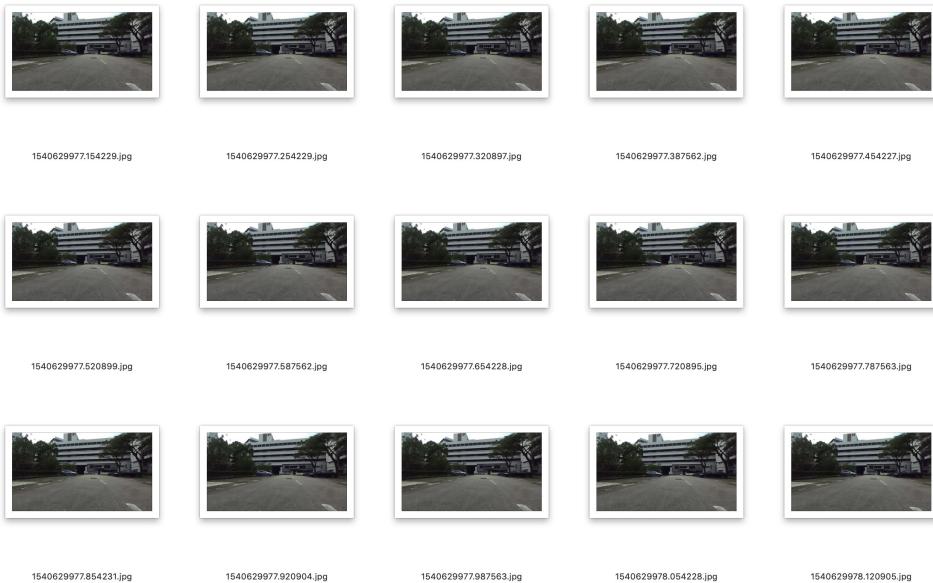


Figure 4.7: Example images of NTU dataset.

Table 4.3: Quantitative results of mapping unseparated Sequence 00.

Distance(m) ¹	Rel. Trans.(m) ²	Rel. Trans.(%) ³	Rel. Yaw(deg) ⁴
371	229.69	61.91	0.37
742	260.10	35.05	0.37
1113	260.20	23.38	0.31
1485	240.93	16.22	0.40
1856	255.16	13.74	0.32

¹ Distance in meter traveled before each time of statistics.

² Mean relative translation error in meter.

³ Mean relative translation error in percent.

⁴ Mean relative yaw error in degree.

4.2 Evaluation of CORBSLAM

4.2.1 KITTI Datasets

In order to evaluate CORB-SLAM system, sequence 00 is utilized and separated into two sub sequences with proper length of overlap. The following separating method is employed: We assume the time period of a KITTI sequence if $Seq.0[0, t]$. Then the sequence is separated into two sub sequences $Seq.01[0, \frac{t}{2} + \delta t]$ and $Seq.02[\frac{t}{2}, t]$ as the assumed input of two client robots.

Therefore, in this case, Sequence 00 containing $f = 4541$ frames and covering a total distance of $s = 1856m$ is separated into two partial sequences: $Seq.0[0, \frac{2}{3}f]$ and $Seq.1[\frac{1}{3}f, f]$, both containing $\frac{2}{3}f \approx 3027$ frames and covering distances of $\frac{2}{3}s \approx 1237m$ (a rough estimate since distances between each pair of frames are not equal).

The ground truth information of Seq.0, Seq.1 and the complete ground truth trajectory of Sequence 00 are shown for reference in Figure 4.8. And Figure 4.13 demonstrates mapping results of each partial sequence and the map fusion results of the server. Four charts in Figure 4.14 contains quantitative evaluation results, with corresponding numeric results shown in Table 4.9.

Clients' quantitative mapping results of each partial sequence are provided in Figure 4.10, 4.11 and Table 4.4, 4.5. And because the two partial sequences are extracted from Sequence 00, so the completed mapping results of CORB-SLAM client on Sequence 00 can be provided as a comparison by Figure 4.9, 4.12 and Table 4.3 in the same format as above. Results are further discussed in Section 5.1.

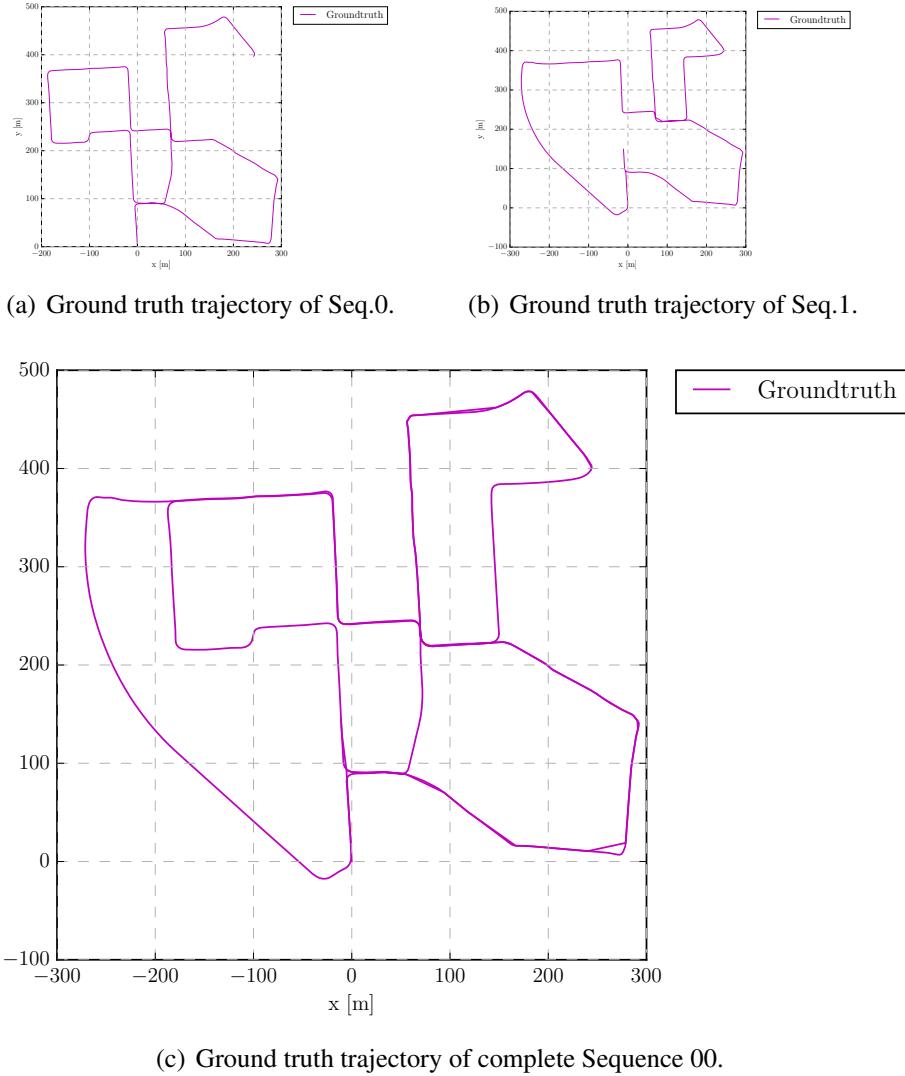


Figure 4.8: Ground truth trajectory of partial and complete sequences of KITTI Datasets.

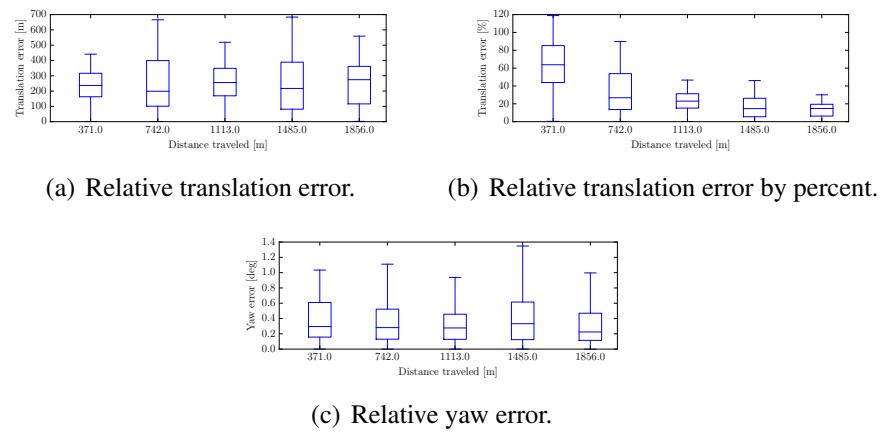


Figure 4.9: Quantitative evaluation results of CORB-SLAM client mapping the entire KITTI Sequence 00.

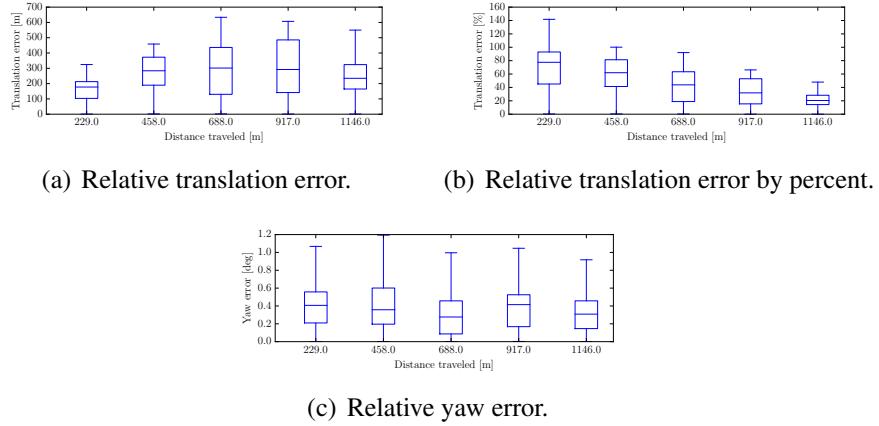


Figure 4.10: Quantitative evaluation results of CORB-SLAM client mapping KITTI partial Seq.0.

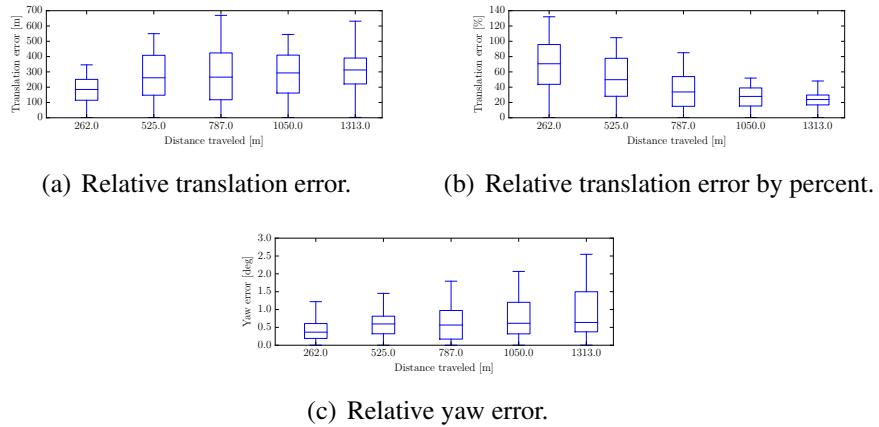


Figure 4.11: Quantitative evaluation results of CORB-SLAM client mapping KITTI partial Seq.1.

Table 4.4: Quantitative results of mapping Seq.0.

Distance(m) ¹	Rel. Trans.(m) ²	Rel. Trans.(%) ³	Rel. Yaw(deg) ⁴
229	161.01	70.31	0.40
458	267.43	58.38	0.42
688	299.07	43.47	0.31
917	306.37	33.41	0.39
1146	256.09	22.35	0.34

¹ Distance in meter traveled before each time of statistics.

² Mean relative translation error in meter.

³ Mean relative translation error in percent.

⁴ Mean relative yaw error in degree.

Table 4.5: Quantitative results of mapping Seq.1.

Distance(m) ¹	Rel. Trans.(m) ²	Rel. Trans.(%) ³	Rel. Yaw(deg) ⁴
262	229.41	114.28	0.40
525	412.58	78.59	0.57
787	386.45	49.10	0.61
1050	390.23	37.16	0.77
1313	464.24	35.36	0.89

¹ Distance in meter traveled before each time of statistics.

² Mean relative translation error in meter.

³ Mean relative translation error in percent.

⁴ Mean relative yaw error in degree.

Table 4.6: Quantitative results of map fusion evaluation on KITTI partial sequences.

Distance(m) ¹	Rel. Trans.(m) ²	Rel. Trans.(%) ³	Rel. Yaw(deg) ⁴
526	283.40	53.88	0.46
1053	276.16	26.23	0.51
1580	153.74	9.73	0.48
2106	284.95	13.53	0.57
2633	219.66	8.34	0.54

¹ Distance in meter traveled before each time of statistics.

² Mean relative translation error in meter.

³ Mean relative translation error in percent.

⁴ Mean relative yaw error in degree.

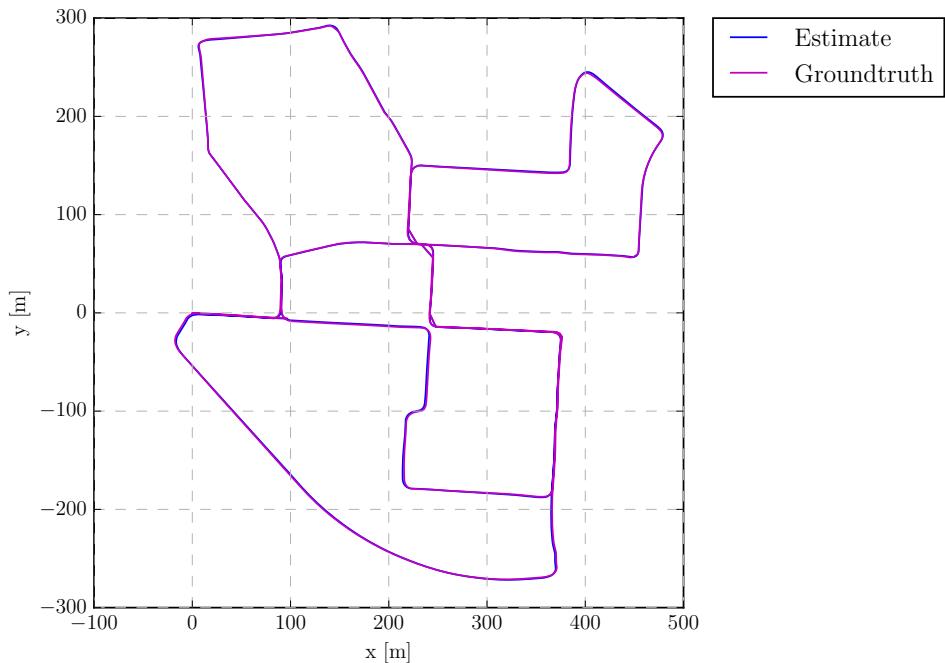
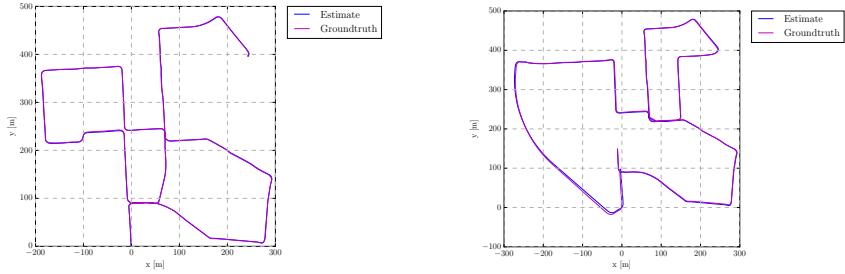
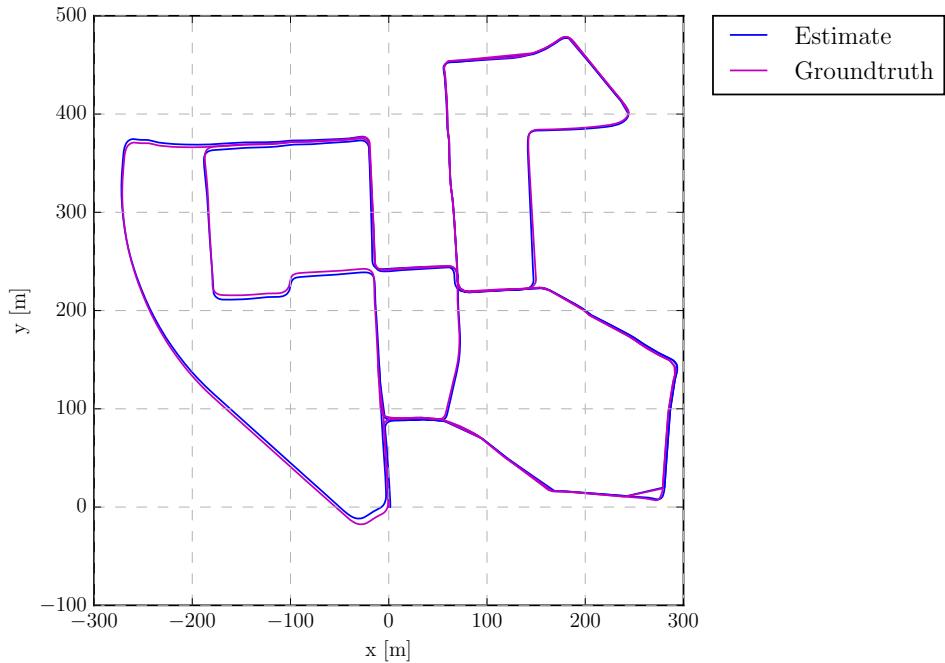


Figure 4.12: Mapping results of the entire sequence without partial sequence.



(a) Mapping result of Seq.0 compared with ground truth.
(b) Mapping result of Seq.1 compared with ground truth.



(c) Map Fusion results of Seq.0 and Seq.1 compared with ground truth.

Figure 4.13: Mapping results of Seq.0 and Seq.1, and the map fusion results of KITTI Datasets.

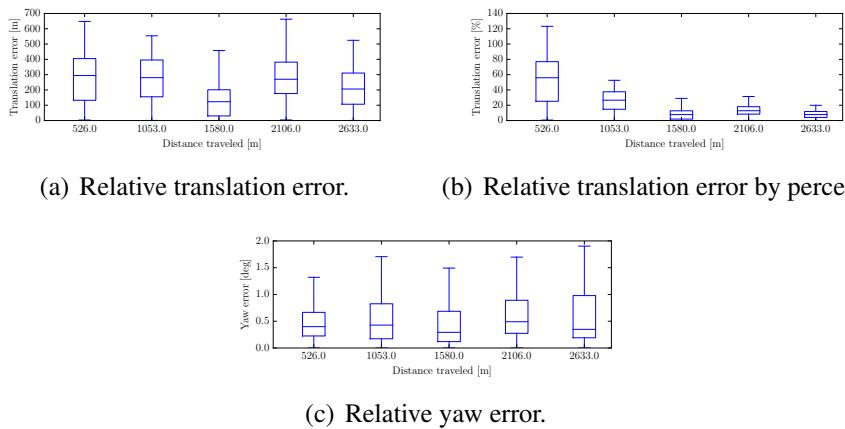


Figure 4.14: Quantitative evaluation results of fused map of KITTI Datasets.

Table 4.7: Quantitative results of mapping evaluation on Bag.0 NTU Datasets.

Distance(m) ¹	Rel. Trans.(m) ²	Rel. Trans.(%) ³	Rel. Yaw(deg) ⁴
29	28.75	99.13	45.55
58	38.97	67.19	34.89
88	51.96	59.05	43.74
117	37.88	32.38	36.31
147	9.29	6.32	10.71

¹ Distance in meter traveled before each time of statistics.

² Mean relative translation error in meter.

³ Mean relative translation error in percent.

⁴ Mean relative yaw error in degree.

4.2.2 NTU Datasets

An obvious drawback of the evaluation on KITTI dataset is the images which are overlapped by two clients are exactly identical because they are extracted from the same sequence. Therefore, the results are expected to be much better than real-world applications in which case it is impossible the images recorded by different clients can be identical.

In order to get more reliable and convincing evaluation results of CORB-SLAM system, another evaluation on multi ground robots is performed utilizing NTU Datasets. Bag.0 and Bag.1 described in Section 4.1.3 are selected in this test. These two bags recorded by two UGVs, have different starting and ending location, with limited overlapping, which is much more similar to the case of real-world applications. Clients' mapping results and map fusion results in the server end compared to ground truth trajectories are demonstrated in Figure 4.16. And ground truth information is provided in Figure 4.15 for reference. Quantitative results of the fused global map are represented in Figure 4.19 and Table 4.9. Quantitative results of each client are provided in Figure 4.17, 4.18 and Table 4.7, 4.8 in the same format as above.

Results are further discussed in Section 5.1.

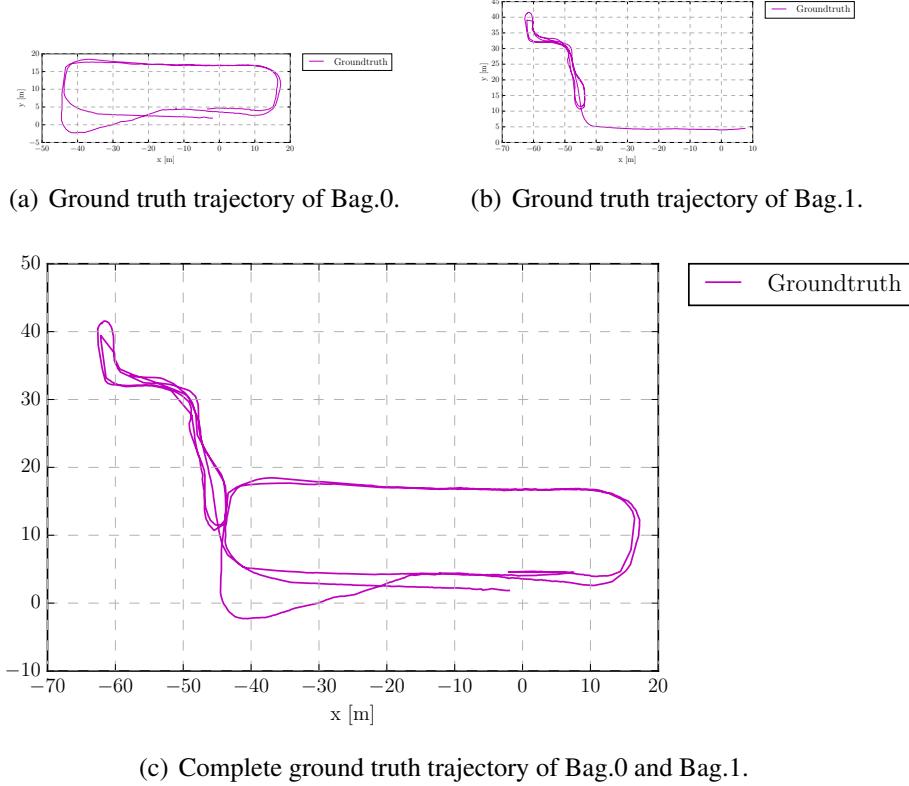


Figure 4.15: Ground truth trajectory of partial and complete bags of NTU Datasets.

Table 4.8: Quantitative results of mapping evaluation on Bag.1 NTU Datasets.

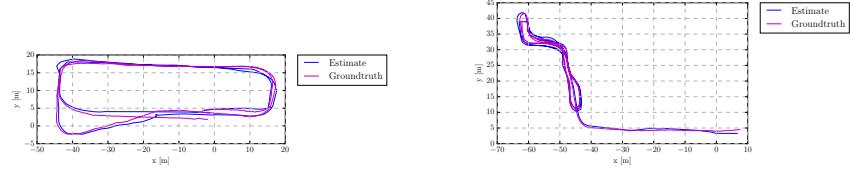
Distance(m) ¹	Rel. Trans.(m) ²	Rel. Trans.(%) ³	Rel. Yaw(deg) ⁴
25	24.79	99.17	71.03
51	37.20	72.94	88.44
76	29.90	39.34	49.64
102	34.40	33.73	67.14
127	38.99	30.70	91.89

¹ Distance in meter traveled before each time of statistics.

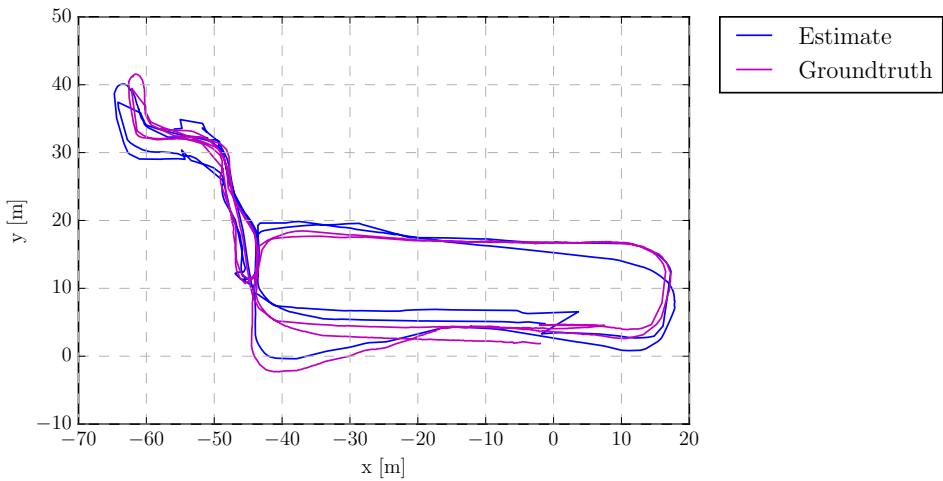
² Mean relative translation error in meter.

³ Mean relative translation error in percent.

⁴ Mean relative yaw error in degree.



(a) Mapping result of Bag.0 compared with ground truth.
(b) Mapping result of Bag.1 compared with ground truth.



(c) Map Fusion results in server end of Bag.0 and Bag.1.

Figure 4.16: Mapping results of Bag.0 and Bag.1 and the map fusion result of server.

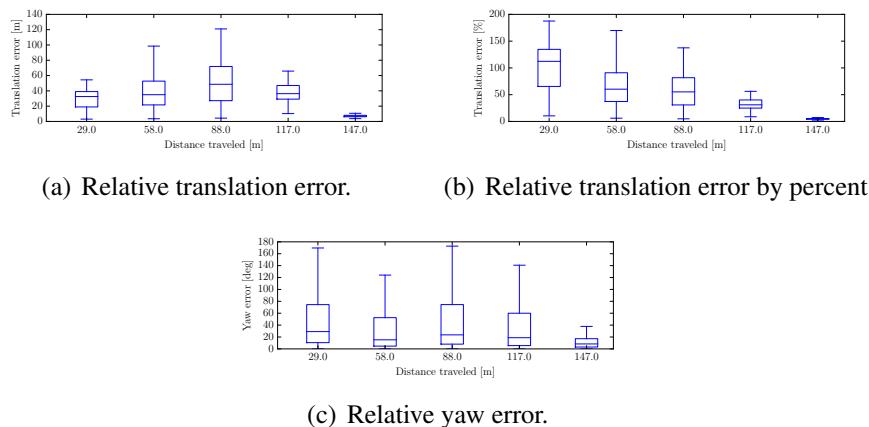


Figure 4.17: Quantitative evaluation results of mapping Bag.0 of NTU Datasets.

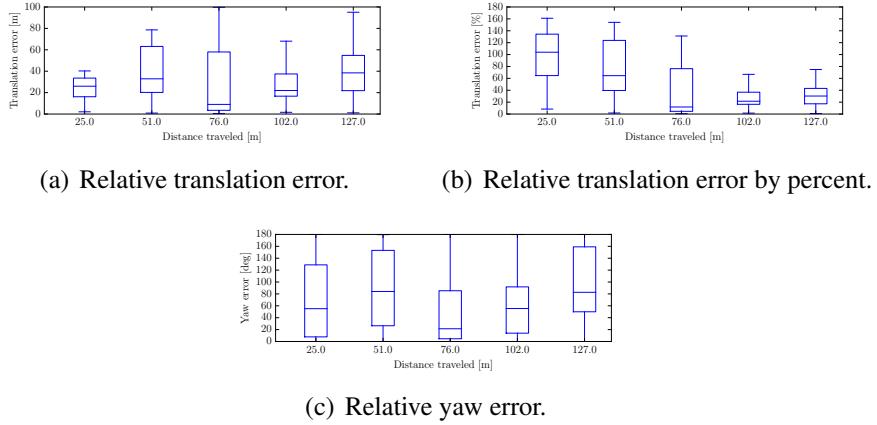


Figure 4.18: Quantitative evaluation results of mapping Bag.1 of NTU Datasets.

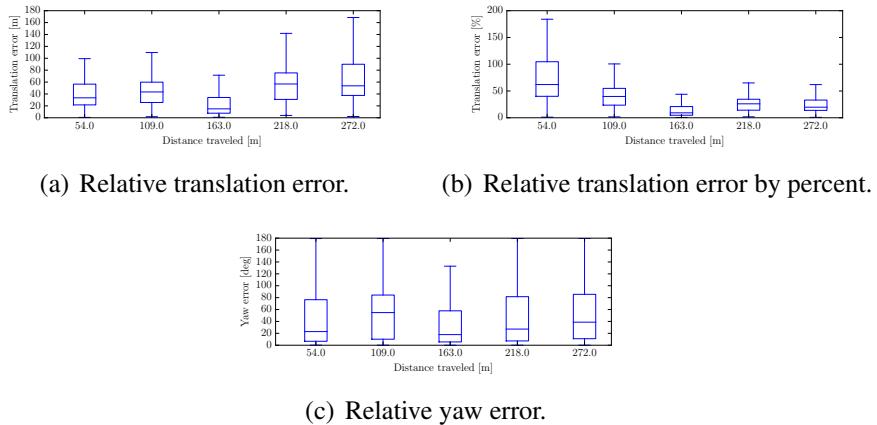


Figure 4.19: Quantitative evaluation results of fused map of NTU Datasets.

Table 4.9: Quantitative results of map fusion evaluation on NTU Datasets.

Distance(m) ¹	Rel. Trans.(m) ²	Rel. Trans.(%) ³	Rel. Yaw(deg) ⁴
54	39.38	72.93	44.39
109	44.88	41.18	57.01
163	26.09	16.00	32.12
218	54.99	25.23	46.11
272	64.66	23.77	54.65

¹ Distance in meter traveled before each time of statistics.

² Mean relative translation error in meter.

³ Mean relative translation error in percent.

⁴ Mean relative yaw error in degree.

4.3 Evaluation under different illumination

Oxford RobotCar Datasets

CORB-SLAM system integrated with illumination variance is firstly evaluated on the selected sequences of Oxford RobotCar Datasets, and then the mapping results are compared with the ground truth trajectories, with quantitative evaluation results calculated.

Two partial sequences are selected according the following principles:

1. Exclude the overexposed photo, which will cause tracking lost in ORBSLAM system. Because the dataset was collected in real-world outdoor street environment, there are some frames with overexposure e.g. Figure 4.20, which cannot be process by vSLAM. Therefore, in this work, this dataset is intercepeted into two sub sequences excluding overexposed images.



Figure 4.20: Image sequence with overexposed frames in RobotCar dataset.

2. Avoid partial sequences where traffic congestion occurred. Because RobotCar Datasets are recorded in different hours during daytime, there are congestion starting at approximately 15:00.

Table 4.10: Partial datasets selected in RobotCar dataset.

Seq. No.	Data(M/D/Y)	Time	Weather	Timestamps
0	07/14/2014	14:49	summer overcast	1405349847738682 to 1405350059147905
1	02/24/2014	12:32	winter overcast	1417794166325288 to 1417794407042717

3. Select partial sequence containing images with overlapping under different illumination conditions and in different season, as shown in Figure 4.4.

According to the above selection principles, the two sub sequences selected are listed in Table 4.10. The ground truth GPS/INS trajectories of two sub sequences and the combined overall trajectories are shown in Figure 4.21. The estimate trajectories and the fused map of two partial sequences are shown in Figure 4.22. Results of Oxford RobotCar Datasets are further discussed in Section 5.2.

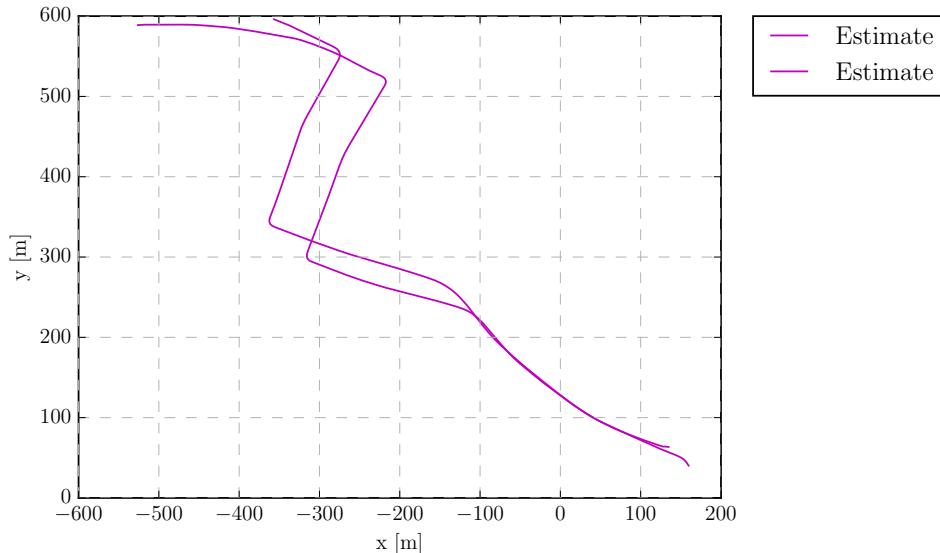
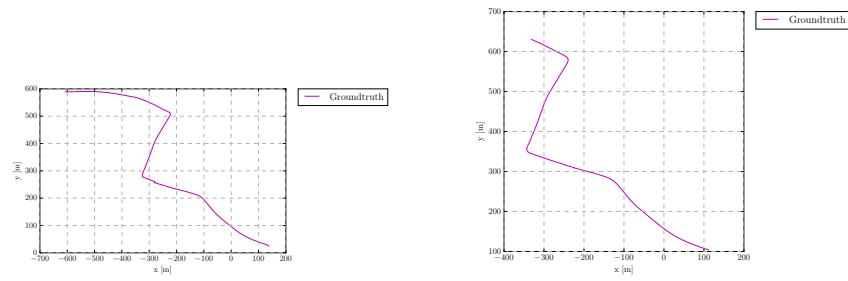
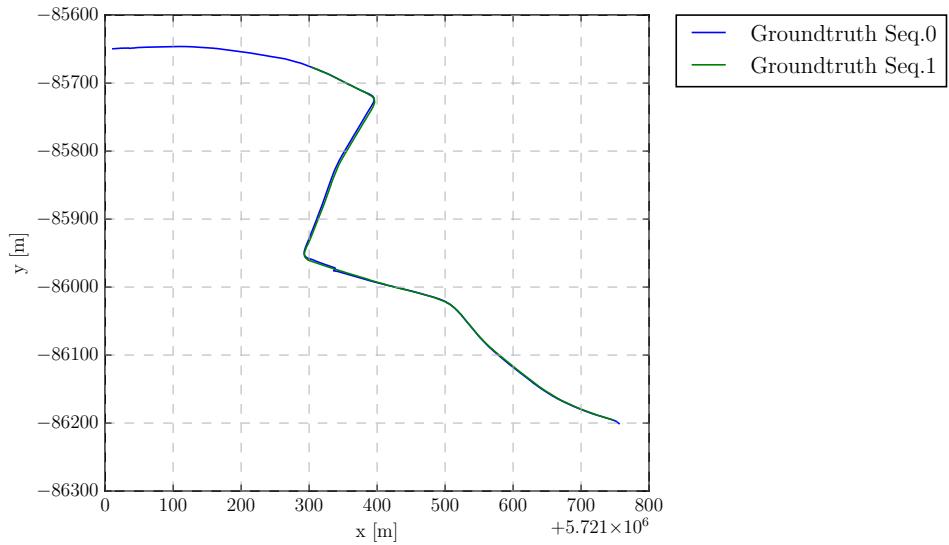


Figure 4.23: Map fusion results of Seq.0 and Seq.1 without ground truth in Oxford RobotCar Datasets.

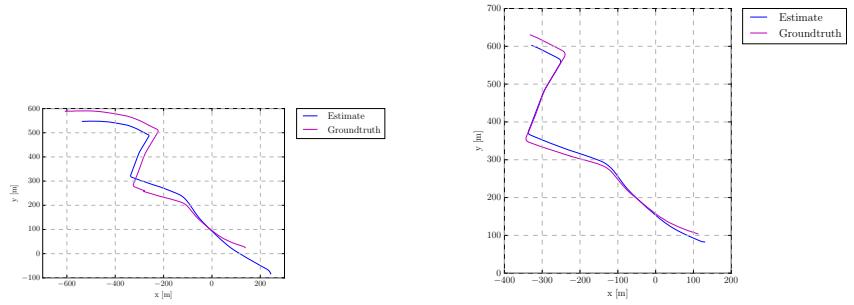


(a) Ground truth trajectory of Seq.0. (b) Ground truth trajectory of Seq.0.



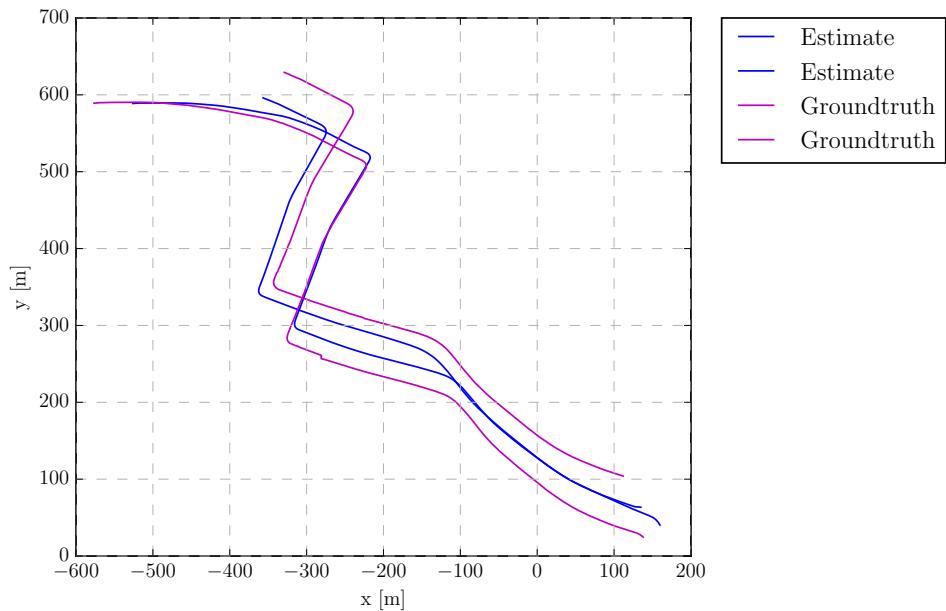
(c) Overall ground truth trajectory of Seq.0 and Seq.1.

Figure 4.21: Ground truth individual and overall trajectories of Seq.0 and Seq.1 in Oxford RobotCar Datasets.



(a) Mapping result of Seq.0.

(b) Mapping result of Seq.1.



(c) Map Fusion results in server end of Seq.0 and Seq.1.

Figure 4.22: Mapping results of Seq.0 and Seq.1 and the map fusion result of server in Oxford RobotCar Datasets.

Chapter 5

Discussion

5.1 Results of multi ground robots cluster

Seen graphically according to Figure 4.13, the mapping results of CORB-SLAM clients is able to match the ground truth trajectory fairly well enough. And seeing the numeric analysis in Table 4.6, the completed map fused by the server has 8.34% mean relative translation error and 0.54° mean relative yaw error with all distance traveled.

Comparing the fused map with the mapping results on the original unseparated sequence, according to the differences between Table 4.6 and 4.3, map fusion in CORB-SLAM server slight reduces the accuracy on relative yaw and scaling. However from the table, the relative translation accuracy is numerically increased, of which a critical reason is the distance traveled by each client is inevitable shortened because the sequence is divided into two parts, which means translation errors on each client are accumulated during shorter distances. Therefore, it is not concluded so far that map fusion module in server can increase accuracy of relative translation.

Evaluation on KITTI Dataset is an ideal circumstance, with all overlapped images identical. To evaluate under a more general real-world circumstance to get more universal and convincing results, another evaluation is performed utilizing NTU Dataset.

According to the comparison of quantitative map fusion result (in Figure 4.19 and Table 4.9) and mapping results of single bags (in Figure 4.17, 4.18 and Table 4.7, 4.8), compared to 6.32% relative translation error of client of Bag.0, and 30.70% of Bag.1, the global fused map in server has an error of 23.77%, which can be considered

as an acceptable result, considering Bag.1 has a complicated trajectory consisting both parts of indoor and outdoor environment causing client map has a relatively higher translation error.

5.2 Failure Reasons and Drawbacks of Illumination Variance Method

According to the ground truth information of the selected partial sequences of Oxford RobotCar Datasets, the correct fused estimate trajectories of clients should be coincident with very little offset seen as Figure 4.21. However as shown in Figure 4.23, integrating CORB-SLAM system with illumination variance failed to enhance the ability to deal with illumination and season changes in Oxford RobotCar Datasets.

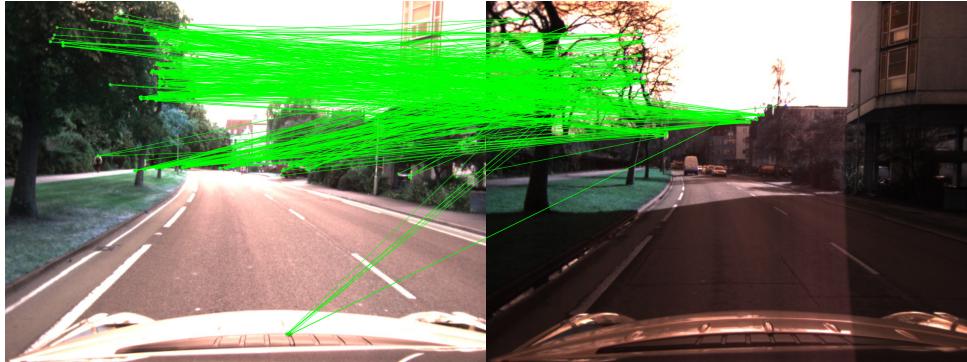
As seen in Figure 4.23, illumination variance method introduce many incorrect matches of keypoints while it is expected to allow ORB matcher to find more correct keypoint pairs that hard to match in raw images due to illumination changes.

By analyzing the basic formula to compute illumination variance, Equation 2.1 which is repeated below as Equation 5.1 for reference, and the result images in Figure 5.1, the following reasons and drawbacks of illumination variance method can be concluded to explain the failure of it to help CORB-SLAM deal with images under different illumination conditions and seasons:

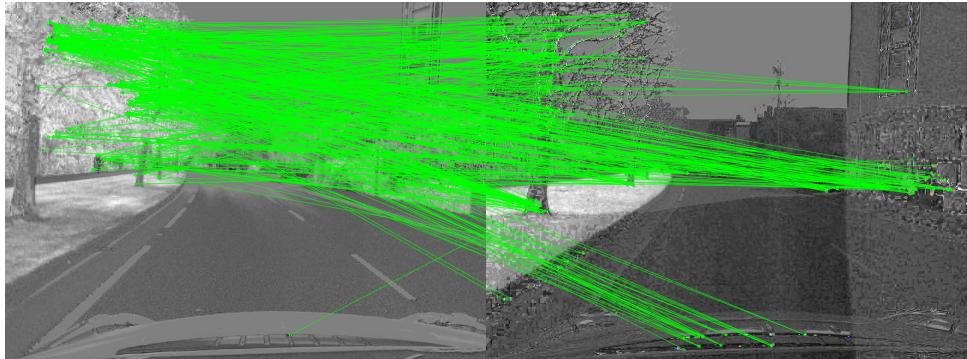
$$I = \log(R) - \alpha \log(G) - (1 - \alpha) \log(B) \quad (5.1)$$

1. Loss of resolution. According to the computation shown in Equation 5.1, there are two steps in this method to calculate the value of illumination variance: for each pixel, 1) logarithm of r,g, b channel values, 2) take their weighted difference as the result illumination variance value. Both steps cause serious loss of resolution, with the result image very blurry as seen in Figure 5.1.
2. Requirement of high brightness and contrast of input color images. In Figure 5.1, compared to Figure 5.1(a), Figure 5.1(b) has a higher resolution. This is because with the loss of resolution during the illumination variance computation, images

with higher brightness and contrast can remain more details in the result images. But the background of application of this method is in life-long localization and mapping system, running under significant lighting and season conditions, which means it is an unreasonable request to demand the input images always bright and sharp.



(a) Corresponding matched keypoint pairs of raw images Figure 4.4(a) and 4.4(b).



(b) Corresponding matched keypoint pairs of illumination variance images Figure 5.1(a) and 5.1(b).

Figure 5.2: ORB Keypoint matching results of raw and illumination variance images in Oxford RobotCar Datasets.

However, in both the original work [3] and the related implement in [22–25], the application of this method only focus on localization, which means the main function is designed to detect loop closures in images taken in different illumination and seasons, but not suitable for mapping. In all above papers, evaluation results are given only by localization results.

A critical difference of localization and mapping tasks is high resolution images are not necessary in localization, while in mapping task they are. Cited by [23], [29] explains that performing localization using a sequence of images rather than single



(a) Corresponding illumination variance image of Figure 4.4(a).



(b) Corresponding illumination variance image of Figure 4.4(b).

Figure 5.1: Generated illumination variance images of example raw images in Oxford RobotCar Datasets.

image removes the requirement that the image matching scheme be able to reliably calculate a single global image match. However, without the functionality of calculating matches between single images, sequence-based image matching algorithms have significant drawbacks that cannot calculate the transformation between images.

And in the case of CORB-SLAM, the image matching algorithm combining ORB keypoint and Bag of Words, is able to calculate matching and transformation between individual images, but fit not well with illumination variance method.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

By analyzing results of the experiments in this work, we can conclude:

1. On systems consisting of multi ground robots, CORB-SLAM provides promising map fusion results, without bringing in obvious relative translation and yaw error.
2. Illumination variance method proposed for dealing with illumination changes, does not cooperate with CORB-SLAM effectively, because of its characteristic of basic equations to generate low resolution images, while ORB Keypoint extraction, DBoW2 bag of words and keypoint-based mapping algorithms are all feature-based which highly rely on high resolution of input images.

6.2 Future Work

As a multi-robot SLAM system which fulfills its cooperative functionality by modifications based on ORB-SLAM2, it inherits the advantages of ORB-SLAM2 system, and meanwhile its disadvantage. Despite of the outstanding map fusion performance of CORB-SLAM system, there are several new drawbacks brought in by introduction of map fusion server that remain to improve in future work.

1. In server end, filtering conditions of (i) loop closure detection, (ii) keypoint inliers in iterations of the Perspective-n-Point solver (PnP Solver), are set significantly

strictly in order to achieve high accuracy in computation of translation matrix between client maps, but as a result, the application of the system are also limited to circumstances where images of clients should have high similarity, with only very slight illumination and season change allowed, so as the point of view difference, which means the application on multi hybrid robot systems is also seriously limited.

2. In the current version of CORB-SLAM, after client maps are successfully fused to global map, the new global map will be resent to each client, which is necessary to enable each client robot to run if they go offline. But in this way, this operation will dramatically increase the memory usage, especially when the server is running simultaneously with a client on a single robot. During the experiment in this work, when processing some datasets with high resolution of raw image, the memory usage of the server may reach up to 3 gigabytes, which is obviously a extra burden for any robot running as a client and the server at the same time.

Bibliography

- [1] Fu Li, Shaowu Yang, Xiaodong Yi, and Xuejun Yang. Corb-slam: a collaborative visual slam system for multiple robots. In Eai International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2017.
- [2] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. Journal of Field Robotics, 33(1):3–46, 2016.
- [3] Will Maddern, Alex Stewart, Colin McManus, Ben Upcroft, Winston Churchill, and Paul Newman. Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, volume 2, page 3, 2014.
- [4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
- [5] Jun Zhang, Prarinya Siritanawan, Yufeng Yue, Chule Yang, Mingxing Wen, and Danwei Wang. A two-step method for extrinsic calibration between a sparse 3d lidar and a thermal camera. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), pages 1039–1044. IEEE, 2018.
- [6] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. The International Journal of Robotics Research, 36(1):3–15, 2017.

- [7] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual slam algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16, 2017.
- [8] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [9] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgbd cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.
- [10] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.
- [11] Hauke Strasdat, J Montiel, and Andrew J Davison. Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, 2(3):7, 2010.
- [12] Hauke Strasdat, Andrew J Davison, JM Martínez Montiel, and Kurt Konolige. Double window optimisation for constant time visual slam. In *2011 International Conference on Computer Vision*, pages 2352–2359. IEEE, 2011.
- [13] Christopher Mei, Gabe Sibley, and Paul Newman. Closing loops without places. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3738–3744. IEEE, 2010.
- [14] Mahmoud A. Abdulgalil, Mahmoud M. Nasr, Mohamed H. Elalfy, Alaa Khamis, and Fakhri Karray. Multi-robot slam: An overview and quantitative evaluation of mrslam framework for mr-slam. In Jong-Hwan Kim, Hyun Myung, Junmo Kim, Weiliang Xu, Eric T Matson, Jin-Woo Jung, and Han-Lim Choi, editors, *Robot Intelligence Technology and Applications 5*, pages 165–183, Cham, 2019. Springer International Publishing.
- [15] Christian Forster, Simon Lynen, Laurent Kneip, and Davide Scaramuzza. Collaborative monocular slam with multiple micro aerial vehicles. In *2013 IEEE/RSJ*

International Conference on Intelligent Robots and Systems, pages 3962–3970. IEEE, 2013.

- [16] Zhao Li, Aftab Ahmed Chandio, Ryad Chellali, et al. Laser only feature based multi robot slam. In 2012 12th International Conference on Control Automation Robotics & Vision (ICARCV), pages 1012–1017. IEEE, 2012.
- [17] Guillaume Bresson, Romuald Aufrère, and Roland Chapuis. Consistent multi-robot decentralized slam with unknown initial positions. In Proceedings of the 16th International Conference on Information Fusion, pages 372–379. IEEE, 2013.
- [18] Dieter Fox, Jonathan Ko, Kurt Konolige, Benson Limketkai, Dirk Schulz, and Benjamin Stewart. Distributed multirobot exploration and mapping. Proceedings of the IEEE, 94(7):1325–1339, 2006.
- [19] Colin McManus, Winston Churchill, Ashley Napier, Ben Davis, and Paul Newman. Distraction suppression for vision-based pose estimation at city scales. In 2013 IEEE International Conference on Robotics and Automation, pages 3762–3769. IEEE, 2013.
- [20] Winston Churchill and Paul Newman. Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In 2012 IEEE International Conference on Robotics and Automation, pages 4525–4532. IEEE, 2012.
- [21] Arren J Glover, William P Maddern, Michael J Milford, and Gordon F Wyeth. Fab-map+ ratslam: Appearance-based slam for multiple times of day. In 2010 IEEE international conference on robotics and automation, pages 3507–3512. IEEE, 2010.
- [22] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Openable: An open-source toolbox for application in life-long visual localization of autonomous vehicles. In Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on, pages 965–970. IEEE, 2016.

- [23] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, J Javier Yebes, and Sergio Gámez. Bidirectional loop closure detection on panoramas for visual navigation. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 1378–1383. IEEE, 2014.
- [24] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, J Javier Yebes, and Sebastián Bronte. Fast and effective visual place recognition using binary codes and disparity information. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3089–3094. IEEE, 2014.
- [25] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Towards life-long visual localization using an efficient matching of binary sequences from images. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6328–6335. IEEE, 2015.
- [26] Colin McManus, Winston Churchill, Will Maddern, Alexander D Stewart, and Paul Newman. Shady dealings: Robust, long-term visual localisation using illumination invariance. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 901–906. IEEE, 2014.
- [27] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7244–7251. IEEE, 2018.
- [28] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [29] Michael Milford. Visual route recognition with a handful of bits. *Proc. 2012 Robotics: Science and Systems VIII*, pages 297–304, 2012.

Appendix: Example Code

Map Fuse To Global Map

```
1 bool MapFusion::mapFuseToGlobalMap( ServerMap *sMap ) {
2     // if global map is null, this submap is inserted into the global
3     // map and does not do transform
4     {
5         std::unique_lock<mutex> lock( nullGlobalMapMutex );
6         if( ifNullGlobalMap ) {
7             cv::Mat Tnorm = cv::Mat::eye(4,4, CV_32F);
8             std::unique_lock<mutex> lock( mSubMapUpdatedMutex );
9             insertServerMapToGlobleMap( sMap, Tnorm );
10            ifSubToGlobalMap[(*sMap).pCacher->pClientId] = true;
11            subMapTransM[(*sMap).pCacher->pClientId] = Tnorm;
12            pubToClient->transMs[ (*sMap).pCacher->pClientId ] = Tnorm;
13            sMap->clear();
14            ifNullGlobalMap = false;
15
16            cout <<"Global Map is not null!\n";
17            return true;
18        }
19    }
20
21    bool flag = false;
22    std::vector<KeyFrame*> allKeyFramesInMapy = sMap->pMap->
23        GetAllKeyFrames();
24
25    bool bOK = false;
26    std::vector<KeyFrame *> candidateKFs;
```

```

26     KeyFrame * currentKF ;
27
28     for( int mit = 0; mit < (int)allKeyFramesInMapy . size () ; mit++ ) {
29
30         KeyFrame * tKF = allKeyFramesInMapy [ mit ];
31
32         cv :: Mat oldTwc = tKF->GetPoseInverse () ;
33         cv :: Mat oldTcw = tKF->GetPose () ;
34         cv :: Mat newTcw = cv :: Mat :: eye ( 4 , 4 , newTcw . type () );
35
36         candidateKFs . clear () ;
37         currentKF = tKF ;
38
39         bOK = detectKeyFrameInServerMap ( globalMap , tKF , newTcw ,
40                                         candidateKFs ) ;
41
42         if ( bOK ) {
43
44             mpGBA->setCurrentKeyFrame ( currentKF ) ;
45
46             mpGBA->setCandidates ( candidateKFs ) ;
47
48             if ( mpGBA->ComputeSim3 () ) {
49
50                 ROS_INFO("Detect In serverMap[%d], from keyframe id[%d]" ,
51                         (int)sMap->pCacher->pClientId , (int)tKF->mnId ) ;
52                 // if this keyframe is deleted in the serverMapx
53
54                 cv :: Mat To2n = oldTwc * newTcw ;
55                 cv :: Mat Tnorm = cv :: Mat :: eye ( 4 , 4 , newTcw . type () );
56
57                 flag = true ;
58
59                 subMapTransM [ (*sMap) . pCacher->pClientId ] = To2n ;
60                 pubToClient->transMs [ (*sMap) . pCacher->pClientId ] = To2n ;
61                 ifSubToGlobalMap [ (*sMap) . pCacher->pClientId ] = true ;
62                 insertServerMapToGlobleMap ( sMap , To2n ) ;

```

```

62         sMap->clear();
63     }
64
65     mpGBA->CorrectLoop();
66
67     time_t end_t = clock();
68
69     cout << "mapfuse " << globalMap->pMap->KeyFramesInMap() << " "
70     << allKeyFramesInMapy.size() << " " << (double)(end_t - start_t)
71     /(double)CLOCKS_PER_SEC << endl;
72
73     break;
74
75 }
76
77 }
78
79 if( flag ){
80     resentGlobalMapToClient();
81 }
82 return flag;
83 }
```

Illumination Variance Conversion

```

1 Mat illumination_conversion(Mat image){
2     vector<Mat> channels(3);
3     split(image, channels);
4
5     Mat imageB,imageG,imageR;
6     Mat imageI = Mat(Size(image.cols,image.rows),CV_32FC1);
7     Mat imageI8U = Mat(Size(image.cols,image.rows),CV_8UC1);
8
9     channels[0].convertTo(imageB,CV_32FC1, 1.0/255.0, 0);
10    channels[1].convertTo(imageG,CV_32FC1, 1.0/255.0, 0);
11    channels[2].convertTo(imageR,CV_32FC1, 1.0/255.0, 0);
```

```

12
13     float valueG , valueB , valueR ;
14     for ( int i = 0; i < imageI .rows ; i ++){
15         for ( int j = 0; j < imageI .cols ; j ++){
16
17             if (imageG .at<float>(i ,j ) != 0)
18                 valueG=log (imageG .at<float>(i ,j )) ;
19             else
20                 valueG=0;
21
22             if (imageB .at<float>(i ,j ) != 0)
23                 valueB=alpha*log (imageB .at<float>(i ,j )) ;
24             else
25                 valueB=0;
26
27             if (imageR .at<float>(i ,j ) != 0)
28                 valueR=(1-alpha)*log (imageR .at<float>(i ,j )) ;
29             else
30                 valueR=0;
31
32             imageI .at<float>(i ,j ) = 0.5 + valueG - valueB - valueR ;
33
34             if (imageI .at<float>(i ,j )<0) imageI .at<float>(i ,j ) = 1;
35
36         }
37
38     }
39
40     imageI .convertTo (imageI8U , CV_8UC1 , 255.0 , 0);
41
42     return imageI8U;
43 }
```