Semi-Direct Visual SLAM for Stereo Cameras: System Design and Validation

by

Boyu Gao

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Faculty of Engineering and Applied Science

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

November 2021

THESIS EXAMINATION INFORMATION

Submitted by: **Boyu Gao**

Master of Applied Science in **Electrical and Computer Engineering**

Thesis title: Semi-Direct Visual SLAM for Stereo Cameras: System Design and Validation

An oral defense of this thesis took place on November 16th, 2021 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Khalid Elgazzar
Research Supervisor	Dr. Jing Ren
Research Co-supervisor	Dr. Haoxiang Lang
Examining Committee Member	Dr. Akramul Azim
Thesis Examiner	Dr. Xianke Lin – Ontario Tech University

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

Simultaneous Localization and Mapping (SLAM) requires an autonomous vehicle in an unknown environment to learn about the environment, generates a map and localize itself at the same time. To solve this problem, various types of sensors are equipped to gather information. Nowadays, with the development of computer vision, Visual-SLAM (VSLAM) that relies on cameras becomes a major topic. Specifically, stereo cameras can provide additional depth information along with regular RGB information.

In this thesis, state-of-the-art VSLAM methodologies are reviewed and evaluated over standard vision benchmarks. Then a novel semi-direct VSLAM system for stereo cameras is proposed. It utilizes direct image alignment for camera pose estimation, and indirect methods to optimize poses and landmarks. The system maintains a sparse point cloud map and allows loop closing and relocalization when tracking is lost. Further experiments validate that it can achieve competitive accuracy with higher efficiency comparing to other VSLAM methods.

Keywords: simultaneous localization and mapping (SLAM); stereo vision; direct image alignment; bundle adjustment (BA)

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

_____ Boyu Gao

STATEMENT OF CONTRIBUTIONS

Part of this thesis has been published as:

B. Gao, H. Lang and J. Ren, "Stereo Visual SLAM for Autonomous Vehicles: A Review," 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020, pp. 1316-1322, doi: 10.1109/SMC42975.2020.9283161.

The writing of the manuscripts, data collection and research conducted in this work and the mentioned publication(s) was completed by the author. Co-authors reviewed and provided technical support when required.

ACKNOWLEDGEMENTS

I would like to express my appreciation to all who helped me during the writing of this thesis. First and foremost, it is my great honor to perform my master's research works under the supervision of Dr. Jing Ren and Dr. Haoxiang Lang, who have provided me with guidance, expertise, and support in every stage of my Master's studies. Their enlightening instructions, impressive kindness and patience help me exceed my self-limitations and make this thesis better and better. Secondly, I wish to acknowledge the support from the staff in the Faculty of Engineering and Applied Science of Ontario Tech University and my colleagues in the GRASP Lab. In addition, I shall extend my thanks to those talented and splendid researchers who have developed the foundation of this field and accomplished essential achievements in academic or application fields, no matter whether their works are cited in this paper. Special thanks to Dr. Xiang Gao and his publication "14 Lectures on Visual SLAM: From Theory to Practice". Last but not least, sincerely grateful to my parents and friends for their support and everything.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 3

CHAPTER 5

LIST OF FIGURES

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

LIST OF ABBREVIATIONS AND SYMBOLS

6DoF	Six Degrees of Freedom
ADAS	Advanced Driver Assistance System
ADS	Automated Driving System
AI	Artificial Intelligence
APE	Absolute Pose Error
BoW	Bag-of-Words
FPS	Frame Per Second
NHTSA	National Highway Traffic Safety Administration
RMSE	Root Mean Square Error
RoI	Region of Interest
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
UAV	Unmanned Aerial Vehicle
VSLAM	Visual SLAM
ToF	Time-of-Flight
VO	Visual Odometry
w.r.t	with respect to

NOMENCLATURES

$x = (u, v, d)$	Feature of image at coordinates (u, v) with its descriptor d.
$x_s = (u_l, u_r, v)$	A stereo feature represented by three coordinates, where (u_l, u_r) are the horizontal coordinates on the left and right images, v is the shared vertical coordinate.
$x_m = (u_l, v_l)$	A monocular feature represented by its two coordinates on the left image.
z	The depth triangulated from the left-right disparity and the intrinsic of the stereo camera.
$T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$	Transform matrix of camera pose, where R is Rotation Matrix and t is transformation vector that describe the 6DoF camera motion.
$F\ and\ K$	Frame and Keyframe.
P_w	A landmark position in the world coordinate system.
n	Viewing direction of a map point.
θ	Weight of the edge in covisibility graph and essential graph.
$\pi_{(.)}$	3D-2D projection function.
(f_x, f_y)	Stereo camera intrinsic: focal length.
(c_x, c_y)	Stereo camera intrinsic: principal point.
b	Stereo camera intrinsic: length of baseline.
$\mathcal{p}_x$	A small patch centered at feature x.
$I_{\mathcal{p}_x}$	Intensity values of the small patch centered at feature x.
$\mathcal{X}_{(.)}$	A set of features.
ρ	The robust Huber cost function.

Chapter 1. Introduction

1.1 Background

In the field of robotics and automotive industry, researchers and developers have been making an effort to equip computers with the ability to understand the world with their own "observations" so that robots or autonomous vehicles are able to interact with the surroundings without human interference or prior-knowledge-based programming. It is a major branch of applications of Artificial Intelligence (AI) in engineering. The goal of this field is to liberate human labour from fundamental working scenarios, or take the place of human in dangerous or costing tasks, such as rescuing and extreme environment exploring.

In terms of the autonomous vehicle, referring to the National Highway Traffic Safety Administration (NHTSA) of America, there are six levels of vehicle automation [1]:

Level 0 – The human driver does all the driving.

Level 1 – An Advanced Driver Assistance System (ADAS) on the vehicle can sometimes assist the human driver with either steering or braking/accelerating, but not both simultaneously.

Level 2 – ADAS on the vehicle can itself actually control both steering and braking/accelerating simultaneously under some circumstances. The human driver must continue to pay full attention ("monitor the driving environment") at all times and perform the rest of the driving tasks.

Level 3 – An Automated Driving System (ADS) on the vehicle can itself perform all aspects of the driving task under some circumstances. In those circumstances, the human driver must be ready to take back control at any time when the ADS requests the human driver to do so. In all other circumstances, the human driver performs the driving task.

Level 4 – An ADS on the vehicle can itself perform all driving tasks and monitor the driving environment – essentially, do all the driving – in certain circumstances. The human need not pay attention in those circumstances.

Level 5 – An ADS on the vehicle can do all the driving in all circumstances. The human occupants are just passengers and need never be involved in driving.

Obviously, level 0 - level 2 vehicles are not really "autonomous", only working to some extent as an assistant of the driver. Vehicles within these levels are able to control the movements but cannot "decide" the way of driving. In this thesis, "autonomous vehicles" refers to vehicles ranging from level 3 to level 5, which drive on their own, relying on equipped sensor inputs and computational outputs, with human on-board only participating as supervisors or even passengers.

The driving force behind the development of autonomous vehicles lies in their superiority in safety and efficiency. The main leads of most traffic crashes are human mis operations or negligence. Autonomous vehicles have the potential to reduce injuries and save lives, as well as erase the direct or indirect economic loss (A NHTSA study showed motor vehicle crashes in 2010 cost US$242 billion in economic activity, including US$57.6 billion in lost workplace productivity, and

US$594 billion due to loss of life and decreased quality of life due to injuries [1]). Based on the newly released TomTom Traffic Index 2019 [2], the congestion level of Toronto is 33% (ranked 80th of 416 cities around the world) , meaning 33% of daily commute time is wasted because of traffic delay. Autonomous vehicles have superiority in cooperation compared to human drivers to maintain smooth traffic flow and reduce traffic congestion, therefore more efficient.

One of the most fundamental functionalities of autonomous vehicles is Simultaneous Localization and Mapping (SLAM), introduced in the next section.

1.2 Simultaneous Localization and Mapping (SLAM)

In a real-world case of driving to another unfamiliar place (defined as "destination"), firstly, the current location of the vehicle is required to determine relative positions of the vehicle and the destination. The second step is to acquire a global map (no matter a paper one or a digital one) containing the two positions, then a path leading to the destination can be planed, making it easy for a navigation system to guide the driver to the destination. This process can be summarized as a repetition of "self-localization along the path within the map" and "moving along the path (current position update)". However, what if there is no global map containing the required information nor any available navigation system? After determining self location and the relative positions, the driver is supposed to learn about the local area to generate a local map, and choose a possible path leading to the destination, then move along the path to a new location. Repeat the steps of "self-localization", "local mapping", "path planning based on relative positions and the local map", "current position update", until reach the destination or ensure it is unreachable.

The second process discussed above, where the "driver" is to learn about an unknown environment and acquire the position information for further applications (e.g., navigation in the above scenario), is the basic idea of SLAM. Level 5 autonomous vehicles are supposed to work in any environment without prior knowledge, and SLAM is one of the most fundamental functionalities that help them explore the unknown environment.

SLAM stands for Simultaneous Localization and Mapping [3, 4]. As the name indicates, it has two main tasks: localization and mapping. The purpose of mapping is to generate a mathematical model of the environment based on the information acquired by means of sensing techniques. A basic map usually contains spatial structure of the environment, locations of surrounding obstacles and topological relationship between different spots. Localization determines the self-position of the autonomous vehicle inside the generated map. Performing localization and mapping simultaneously and continuously, an autonomous vehicle perceives the environment and understands the relative spatial relations between the environment and itself. This is the foundation of autonomation, upon which an autonomous vehicle is able to navigate itself to any destination position within or beyond the mapped area.

2D grid map (see Figure 1.1) and 3D point cloud (see Figure 1.2) are two typical forms of map used in SLAM systems. 2D grid map is simple and contains limited information of environment. It divides the "ground" into small girds and shows the "occupancy state" of each grid. Black grids represent obstacles in the environment and white grids are free space accessible to the autonomous vehicle. 2D grid maps are usually generated with sensors that operate in a limited plane in space, such as

2D lidar and radio signals. 3D point cloud is a set of 3D points corresponding to real-world obstacles, which can be directly generated by RGB-D cameras (e.g., the Kinect) and 3D lidar, or triangulated from stereo cameras. According to the density of 3D points generated, a point cloud is classified into “dense”, “semi-dense” or “sparse”. The more 3D points generated, the heavier it becomes and the richer information it contains. Figure 1.2 (a) is a sparse 3D point cloud of real-world outdoor road scenes and (b) is a reconstruction model of a real-world office scenario generated from a dense point cloud.

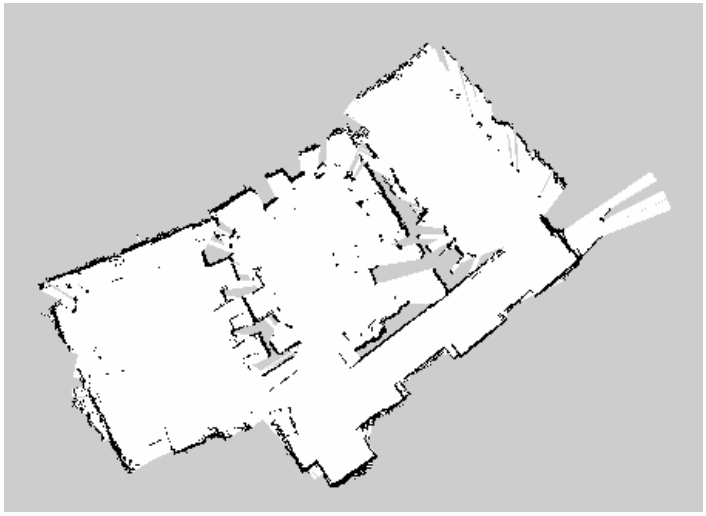
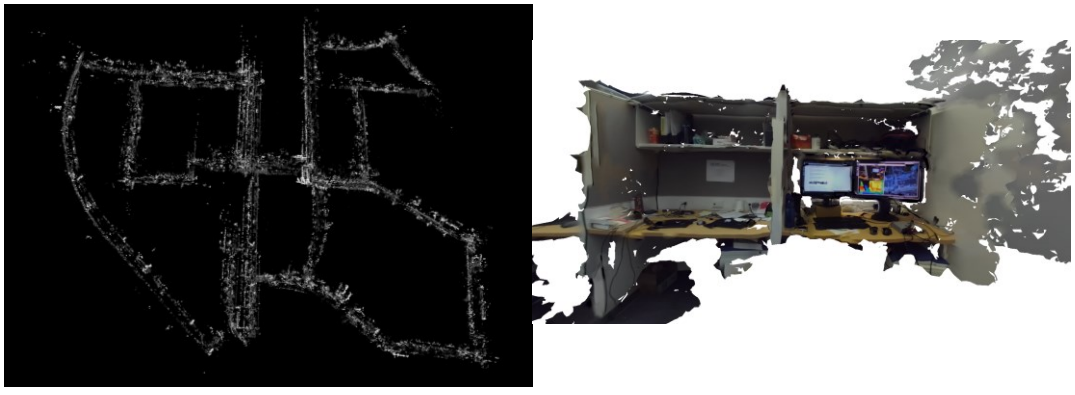


Figure 1.1: 2D grid map.



(a) sparse road scenes (b) dense reconstruction of office scenario

Figure 1.2: 3D point cloud.

A SLAM system usually utilizes same sensor for both localization and mapping. Localization has proceeding importance since mapping usually works with localization information to generate maps, which means localization accuracy directly affects the quality of mapping. Therefore, further discussions are mainly from the perspective of localization.

1.3 Visual SLAM (VSLAM)

In the state-of-the-art field of autonomous vehicles, position information can be acquired through various practical and well-developed modern techniques, such as the global positioning system (GPS), 4G/5G cellular systems, ultra-wideband (UWB), WLAN, wireless sensor network (WSN), Bluetooth, ultrasound, lidar, etc.[5-14] and various SLAM solutions are developed using different sensors, such as Gmapping [15], HectorSLAM [16], tinySLAM [17] and many other mature solutions [18-21].

However, challenges and limitations still exist in terms of accuracy, cost, complexity, security and scalability. GPS and cellular signals are severely degraded in indoor environment, dense urban areas, and scenarios with shadowing effects. Other radio signals have the limitations in propagation attenuation, sensitivity to noise, least number requirement of base stations, similar carrier frequency interference, and energy consumption. Thus, researchers and developers in this field are still seeking better solutions.

Compared to other sensor modalities, camera is cheap with low power consumption, and provides detailed environment information in multiple dimensions. Camera

mimics the sense of human, which means better comprehension and recognition of the environment. It is a crucial module of a SLAM system to detect and close loops, and recover localization when tracking is lost. Images captured by cameras are utilized in recognition because each frame in the image sequence can be easily described distinguishably from each other. Therefore, Visual SLAM (VSLAM) has now become the mainstream of research[22-26].

1.4 Sensor Setup for VSLAM

One monocular camera is the minimum sensor setup for a VSLAM system, but monocular VSLAM has a severe drawback, lack of depth information, since an image is a projection from 3D real world to a 2D image plane, eliminating the distance between camera and objects. One defect is scale ambiguity, real size of the object cannot be speculated from the image without depth information. And this may result in scale drift between monocular VSLAM estimations and the real world. At the meantime, depth information is necessary for both localization (pose estimation) and mapping (3D structure reconstruction). Thus, to acquire depth information, at least two different image frames are required to get their disparities, shift of objectives between the frames. The closer an object is, the more shift it appears between frames. Depth information are then calculated with the disparities and known camera calibration parameters. This causes extra specialized initialization steps in monocular VSLAM, which may also influence accuracy. Other algorithms inferring depth information from monocular images, such as [27, 28], address high demand of computational resources and GPU parallelism for calculations or machine learning, which are not suitable for an autonomous vehicle either.

To acquire depth information, one way is to use an RGB-D camera (e.g., Kinect), adding an infrared laser scanner to measures the distance. Yet, infrared laser scanner has limitations in effective range and anti-interference ability to natural lights, thus RGB-D camera mainly works in indoor environment.

Another way is to use a stereo camera, adding another camera horizontally with fixed baseline. Depth information is triangulated from the disparities between left and right views, just like human eyes. Since depth measurement does not rely on any other sensor, and the measurement range only depends on camera resolution and baseline length, stereo cameras can be used both indoor and outdoor.

1.5 Motivations

SLAM is the foundation of autonomous vehicles. Nowadays, VSLAM applications have already been in practice in many fields.

One is self-driving car. Tesla, Apple, Google, Uber and many other companies have developed their own self-driving cars along various technique routines. Tesla mainly relies on visual solutions. A Tesla car is equipped with eight surround cameras providing 360 degrees of visibility around the car at up to 250 meters of range, and twelve updated ultrasonic sensors complementing vision. According to the official videos and reports, most Tesla cars achieve level 2 of automation, and level 3 under experimental conditions. Self-driving car is believed to become a main way of transport in the future.

Another field is about delivery robots. Companies such as Starship, Kiwi, TeleRetail, Amazon and FedEx have deployed their autonomous robots within some campus or

local areas to deliver foods or products. In-person interactions are limited to a low level, which is extremely beneficial under the current situation of COVID-19. Some of the robots use cameras to perform VSLAM and distinguish obstacles or pedestrians along the road.

Autonomous Unmanned Aerial Vehicle (UAV) is also a potential field that can expand the scope of human perception. Nowadays, most civilian UAVs (e.g., DJI) use cameras for obstacle sensing and visual servoing to assist human control. They can be further developed for VSLAM without adding extra sensors. Besides, UAVs have limited carrying capacity and battery endurance. It is better to amount as few sensor modalities as possible. Camera is a best choice in this case.

1.6 Objectives

The primary objective of this thesis is to propose a novel Stereo VSLAM system that exclusively utilizes a stereo camera to perform accurate localization and mapping simultaneously in real time, with low consumption of both energy and computational resources. It should also apply standard input and output format to be more adaptable.

The system is expected to meet the following specifications:

- Accuracy

  It is important to provide accurate localization and mapping information. The method is supposed to achieve competitive accuracy with respect to other state-of-the-art Stereo VSLAM methods.

- Efficiency

The system is designed to work in real-time, which requires the capability of rapid processing of the input images and pose estimation.

- Low Consumption

The system is expected to be operated on either a large autonomous car or a small unmanned robot, even a drone with limited carriage. Thus, firstly, it should not perform heavy computational tasks (such as GPU computing and machine learning), since the autonomous platform may not have excess computational resources. At the meantime, this also means the system should be energy saving for longer endurance.

- Adaptability

The system should be robust enough to be competent in both indoor and outdoor scenarios. Besides, the output of the system should be standardized as other existing VSLAM methods so that it can be easily adapted to various further applications.

1.7 Contributions

Main contributions:

1) Proposed a novel semi-direct VSLAM system for stereo cameras.

2) Validated system design over standard benchmarks in terms of accuracy and efficiency.

Minor contributions:

3) Provided a general introduction SLAM and VSLAM.

4) Provided a detailed literature review of VSLAM methodologies.

5) Evaluated five state-of-the-art VSLAM systems for stereo cameras with standard benchmarks in terms of accuracy and efficiency.

1.8 Content Structure

Chapter 2 provides a detailed literature review of VSLAM methodologies. The classic framework of a VSLAM system is illustrated and five state-of-the-art stereo VSLAM systems of two different categories are reviewed.

Chapter 3 evaluates the five Stereo VSLAM systems on several well-known standard vision benchmarks, and discusses their performance, superiorities, and drawbacks. A discussion on how to deal with the deficiencies while retaining the merits of state-of-the-art VSLAM methods is also carried out in this Chapter.

Chapter 4 presents detailed design of the proposed Stereo VSLAM system.

Chapter 5 presents validation of the proposed Stereo VSLAM system as well as the performance comparison to the state-of-the-art Stereo VSLAM systems.

Chapter 6 concludes the thesis with a discussion on the pros and cons of the proposed Stereo VSLAM system and its future development potentials.

Chapter 2. Literature Review

2.1 Classic VSLAM System Framework

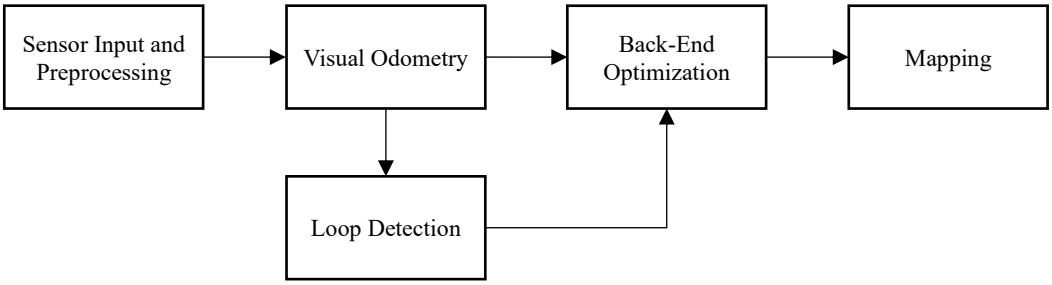


Figure 2.1: Classic VSLAM system framework.

The classic VSLAM system framework is illustrated in Figure 2.1. There are five main blocks that are the foundations of all VSLAM systems.

- Sensor Input and Preprocessing

This is the input block that separates the VSLAM system and the real world. Various types of data from all the equipped sensors are gathered and preprocessed, then dropped. The VSLAM system merely works with the output of this block in order to get invariant to sensor diversification.

Besides cameras as main input, sensors such as depth sensor and Inertial Measurement Unit (IMU) may also be utilized. The input images (or image pairs from a stereo camera) are rectified and synchronized with other sensor inputs then preprocessed into required format and sent to Visual Odometry and Loop Detection blocks.

- Visual Odometry (VO)

This key block is responsible for estimating vehicle trajectories and generating local maps according to information from the input block. Since VO only calculates the motion between each two adjacent frames, the noise from input and inaccurate estimation would lead to inevitable accumulating drift, and the final map and trajectory estimation would become unreliable. Back-End Optimization and Loop Detection blocks are designed to solve this problem.

- Loop Detection

By comparing the similarity between current frame and input block to historical frames inside current global map, this block judges whether the vehicle is visiting a location for a second time, called a loop. A detection of a loop would be sent to Back-End Optimization block to erase the accumulating drift, called loop closure.

- Back-End Optimization

This block estimates spatial uncertainty of current trajectory estimations and current global map under the influence of system noise, then applies particle or non-linear filters to minimize the uncertainty. In most cases, the motion (i.e., camera pose) and structure (i.e., landmarks) are jointly optimized for most accurate estimation.

- Mapping

Stitch map points generated from input frames according to trajectory estimation to generate required global map. Maps can be categorized into metric map and topological map. A matric map uses landmarks to precisely represent object locations

of the environment, whereas a topological map is a graph consisting of nodes and edges, which merely indicates the connectivity between locations.

2.2 Indirect and Direct VSLAM

VSLAM systems can be categorized as direct VSLAM [29] and indirect VSLAM. Direct VSLAM systems directly exploit illumination information and photo consistency of RoIs (region of interest) as system input for the following steps. Pose estimations are obtained by minimizing the photometric error between corresponding pixels. Indirect VSLAM systems, on the other hand, firstly extract features of the image as the system input. Then the features are described and matched for pose estimation by minimizing the reprojection error. Direct systems avoid the reliance on typical types of features or texture of the environment, and have no prior step of image processing and feature association, but is sensitive to illumination conditions. In contrast, indirect systems benefit from the features that they can be applied for robust tracking and loop-closure. Also, joint optimization of feature-based motion and structure is an established non-linear least-squares problem, known as Bundle Adjustment (BA) [30], which can be addressed by many existing solvers efficiently[31-34].

In this thesis, five methods are selected as the representatives of the state-of-the-art VSLAM methodologies. ProSLAM [35], ORB-SLAM [36, 37] and OpenVSLAM [38] are indirect and LSD-SLAM [39, 40] and DSO [41, 42] are direct. These methods are reviewed in the following sections and evaluated with various benchmarks in the next Chapter.

2.3 ProSLAM

ProSLAM (Programmers SLAM) is a simple indirect stereo-camera-only VSLAM method, designed as a highly modular system that can be easily understood and implemented. The method is suggested to execute in a single thread to avoid multi-thread synchronizing complexity and preserve memory. The absence of BA and other further simplifications lead to substantially low computational requirements.

ProSLAM has a straightforward non parallel pipeline with four cycling core module sequences:

- Triangulation

  Extracts, describes, and matches features from input image pairs, and produces 3D points. A structure named as a frame is used to store the output 3D points and corresponding feature descriptors.

- Incremental Motion Estimation

  Estimate the pose of the current frame from a pair of subsequent frames.

- Map Management

  Commonly image points tracked along multiple subsequent frames are grouped as landmarks. Then landmarks in the nearby regions of the current frame along the trajectory are grouped as a point cloud with each point having multiple descriptors called a local map. Finally, pose-graph [43] is used to arrange the local maps.

- Relocalization

Compares the current local map against other past generated local maps. The Hamming binary search tree library HBST [44] is utilized for a descriptor to descriptor similarity search. Upon a successful search, classic ICP [45] achieves an alignment. High inlier count and low average error transforms are applied to the pose-graph to update the entire map.

2.4 ORB-SLAM and ORB-SLAM2

ORB-SLAM2 is an open-source complete indirect VSLAM method for monocular, stereo, and RGB-D cameras, including map reuse, loop closing and relocalization capabilities expanded from the previous work ORB-SLAM. The method shows adaptability for both small indoor scenarios and large-scale outdoor environments. It achieves accuracy improvement by using stereo feature classification and Levenberg–Marquardt BA optimization implemented in g2o [34]. A lightweight localization mode is also integrated in this method to efficiently reuse the map.

ORB-SLAM2 has three main parallel threads: tracking, local mapping and loop closing, followed by a full BA thread as global optimization. In the tracking thread, the system first extracts ORB features [46] of input images. ORB is a robust and efficient binary feature descriptor that presents good invariance to image rotation, scale and brightness as well as fast real-time processing speed. All other system operations are based on the extracted features and without using the original images, making the system independent of the type of camera. As for stereo cameras, ORB features are extracted in the rectified stereo image pairs. Secondly, matched features

of the image pairs are selected as stereo features and classified into close or far depending on whether their associated depth is less than 40 times the stereo camera baseline. Close features can be triangulated to estimate information of depth, scale, translation and rotation, while far features provide accurate rotation information. In this way, camera poses are estimated and then optimized using motion-only BA. A place recognition module based on DBoW2 [47] is applied in case of tracking lost due to occlusions or abrupt movements or loop detection. After camera pose estimation, the system generates a local visible map using covisibility graph of keyframes and match the features of the current keyframe with the local map points to optimize camera pose again. The last step of the tracking thread is to decide if the current keyframe should be inserted. The keyframe must meet the conditions with at least 50 features and less than 90% similarity to keyframes sharing points with a current frame inside the local map. Meanwhile, it is only inserted when the map is not in process of global relocalization or local optimization.

If a new keyframe is inserted, the local mapping thread searches the new features corresponding to features inside the covisibility graph (local map) to triangulate a new local map. Exigent points and redundant keyframes are culled based on tracking information and local BA optimization is performed in this thread for a high-quality reconstruction. If the new keyframe is inserted while the local mapping is still performing local BA for a previous keyframe, it will stop the previous optimization step and process the new keyframe as soon as possible.

The loop closing thread starts with every new keyframe. If a loop is detected, the system computes the drift accumulated in the loop, aligns both sides of the loop and

fuse duplicated features, and finally, performs a pose-graph optimization [48] based on rigid body transformations to achieve global consistency. The optimization is performed over the Essential Graph, a sparser subgraph of the covisibility graph that retains all the keyframes with less edges.

The three threads above operate parallelly. After the pose-graph optimization in the loop closing thread, a full BA optimization is launched in a separate thread because it might be costly. If the loop closing thread detects a new loop while the optimization is running, the optimization is aborted and will be relaunched after the loop closing operations. When the full BA finishes with no loop occurred, the system merges the optimization output with the latest state of the map. Newly inserted keyframes and points while the optimization was running are also transformed according to the correction transformation of the updated part.

2.5 OpenVSLAM

Shinya Sumikura et al. proposed a high usable and extensible visual SLAM method, OpenVSLAM. It is appropriately designed as open-source callable libraries from third-party programs. OpenVSLAM is compatible with various types of cameras, even with fisheye and equirectangular cameras. Besides, the built maps can be stored and loaded for future localization applications with interfaces that are provided for convenience. Furthermore, users can easily check the results on a cross-platform viewer running on web browsers.

Similar to the ORB-SLAM, OpenVSLAM also applies ORB as a feature descriptor and BA method in g2o for optimization. The system consists of three modules:

tracking, mapping and global optimization. The tracking module estimates a camera pose for an input image frame via features matching and pose optimization and decides whether to insert it as a new keyframe to other modules. The mapping triangulates new 3D points from inserted keyframes to create and extend the map and perform local BA optimization for windowed map. The global optimization module is responsible for loop detection, pose-graph optimization, and global BA.

One of the notable advantages compared to other indirect VSLAM methods is that, OpenVSLAM accepts input images captured with perspective, fisheye and equirectangular cameras and pre-process the images to extract features, especially for the capability with omnidirectional image, which implies a significant benefit for tracking and mapping.

2.6 LSD-SLAM and Stereo LSD-SLAM

Stereo LSD-SLAM is a large-scale direct SLAM method for stereo cameras. In contrast to the indirect methods, photo consistency of high-contrast pixels, including corners, edges and high texture areas, are exploited to align images, letting a rich set of pixels contributes to depth estimation and mapping. The authors also proposed an approach to enforce illumination invariance, making it capable of handling aggressive brightness change between frames.

The preceding monocular method LSD-SLAM has three main steps. At first, a reference keyframe is selected in the map, and the camera motion is then estimated, referring to the keyframe. A new keyframe is generated if the current camera capture of little similarity to the existing keyframes. Furthermore, the system applies

temporal stereo to estimate depth in the current reference keyframe according to the tracked motion. Finally, the poses of the keyframes are made globally consistent by mutual direct image alignment and pose graph optimization.

In Stereo LSD-SLAM, temporal stereo is combined with static stereo in the fixed-baseline stereo camera setup where scale is determined. It allows depth estimation beyond the fixed baseline and fixed direction. Illumination invariance is achieved using a modified cost function similar to the normalized cross-correlation (NCC) that is invariant to affine lighting changes.

2.7 DSO and Stereo DSO

DSO stands for Direct Sparse Odometry. Published by the same research group after LSD-SLAM Stereo LSD-SLAM and DSO, it is also a large-scale direct VSLAM method for stereo cameras that combined static stereo with multi-view stereo. It differs from stereo LSD-SLAM that an active sliding window is used to jointly optimize all the model parameters, including the camera parameters and the affine brightness parameters of all keyframes as well as the depth information of all selected pixels. Stereo DSO performs better and faster in large-scale environments because of the joint optimization inside active windows rather than the global optimization. In addition, the metric 3D reconstruction by Stereo DSO is more precise than their previous direct methods.

Chapter 3. Benchmarks of Current VSLAM Systems

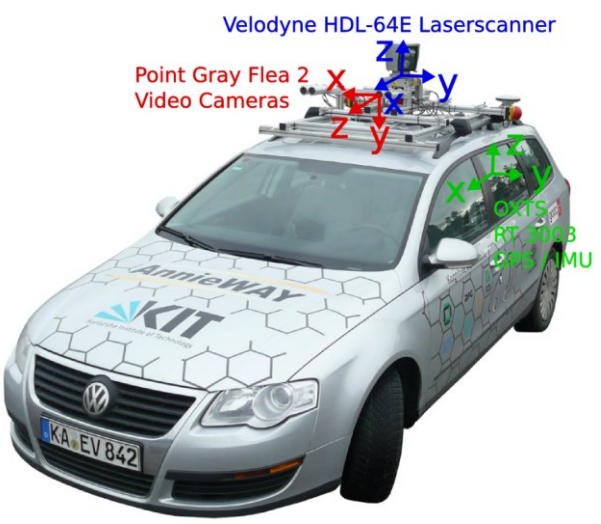
3.1 KITTI and EuRoC Dataset

3.1.1 KITTI Dataset

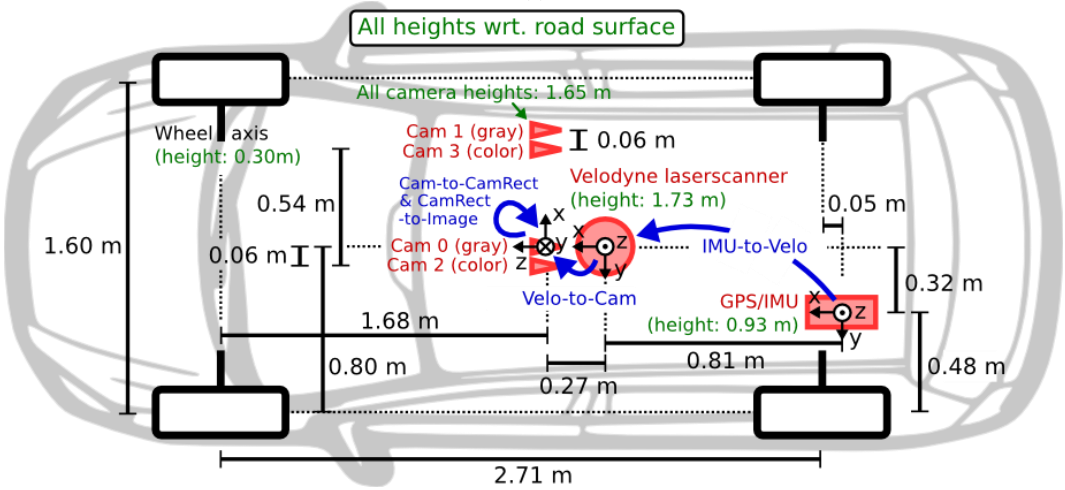
The KITTI Vision Benchmark Suite [49, 50] is a project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago. Real-world driving data ranging from rural highways to inner-city road scenes are captured with a variety of sensor modalities mounted on a car-based data-collecting platform for use in mobile robotics and autonomous driving research. It provides calibrated, synchronized, and timestamped data sequences including high-resolution color and grayscale stereo camera images, 3D laser-scan point clouds, high-precision GPS measurements and 6D IMU accelerations, as well as object labels in the form of 3D tracklets. The data-collecting platform and its sensor setup are illustrated in Figure 3.1 and Table 3.1. There are 11 training sequences (00-10) published with ground-truth motion trajectories from a combined GPS/IMU system and 11 testing sequences (11-21) without ground-truth for their online benchmarks for stereo, optical flow, object detection and other tasks.

TABLE 3.1:
DATA-COLLECTING PLATFORM SENSOR SETUP OF KITTI

Data	Sensor Type	Rate	Specifications
Stereo Images (grayscale)	PointGray FL2-14S3MC (x2)	10 Hz	1.4 Megapixels, 1/2” Sony ICX267 CCD, global shutter, ∼0.54 m baseline
Stereo Images (color)	PointGray FL2-14S3C-C (x2)	10 Hz	1.4 Megapixels, 1/2” Sony ICX267 CCD, global shutter, ∼0.54 m baseline
3D Point Cloud	Velodyne HDL-64E	10 Hz	64 beams, 0.09 degree angular resolution, 2 cm distance accuracy, collecting ∼1.3 million points/second, field of view: 360◦ horizontal, 26.8◦ vertical, range: 120 m
IMU and Ground-Truth	OXTS RT3003 + GPS	100 Hz	6 axis, L1/L2 RTK, resolution: 0.02 m / 0.1◦



(a)



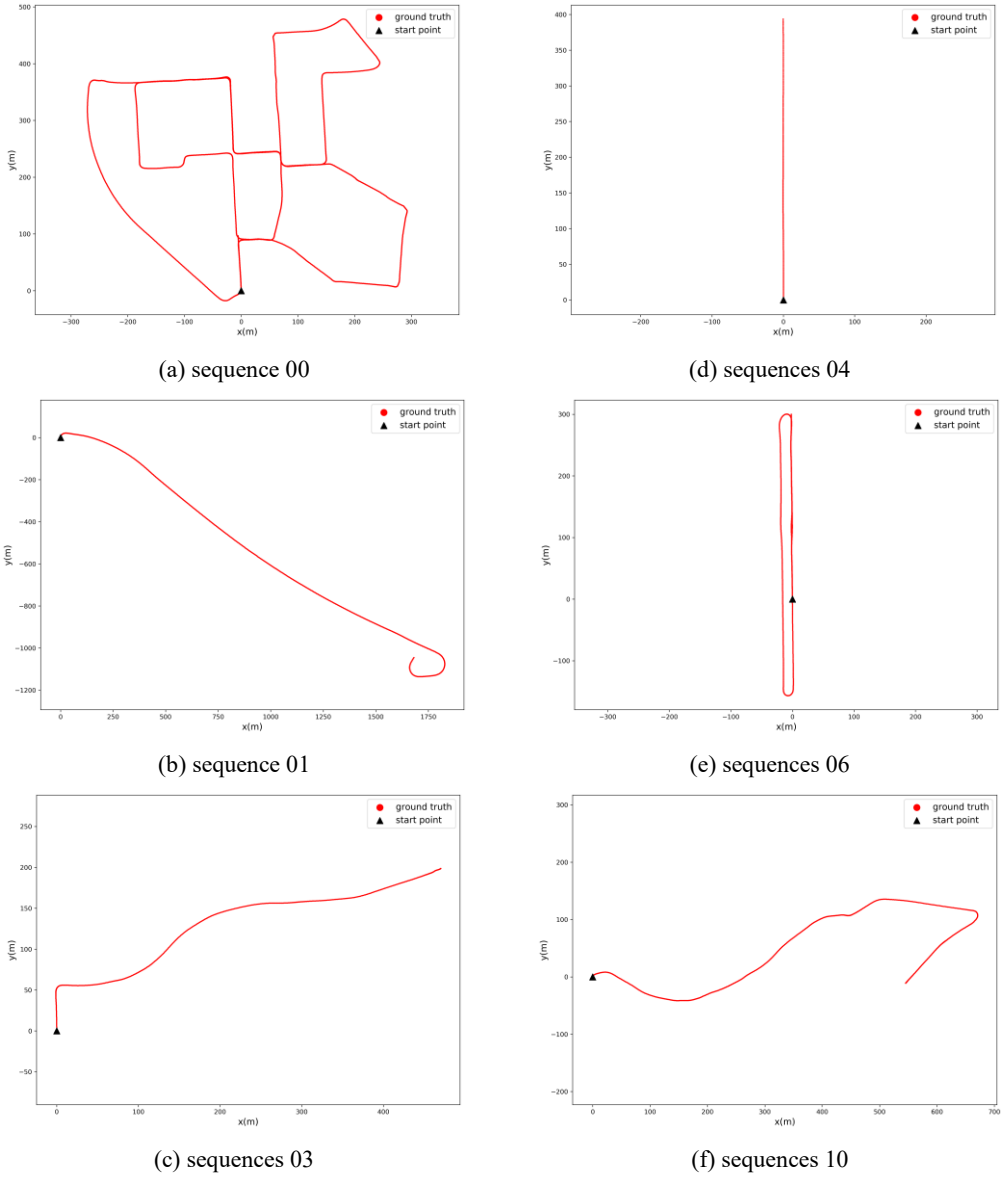
(b)

Figure 3.1: KITTI data-collecting platform (a) and sensor setup (b) [26].

In this section, the KITTI training sequences 00, 01, 03, 04, 06, 10 (sequence 00 and 06 contain loops) are selected as outdoor benchmarks of to evaluate the state-of-the-art VSLAM systems. The rectified stereo grayscale image sequences are used as input of VSLAM systems, and the provided ground-truth camera poses are used to evaluate their estimated trajectories of camera motion. Figure 3.2 and Table 3.2 offer an overview of selected KITTI sequences.

TABLE 3.2:
OVERVIEW OF SELECTED KITTI SEQUENCES

Sequence	**00**	**01**	**03**	**04**	**06**	**10**
Road Type	city	highway	city	city	city	residential
Contain Loops	yes	no	no	no	yes	no
Number of Image Frames	4541	1101	801	271	1101	1201
Total Path Length (m)	3724	2453	560	393	1232	919
Max Driving Speed (km/h)	46	96	31	56	51	51



(a) sequence 00

(d) sequences 04

(b) sequence 01

(e) sequences 06

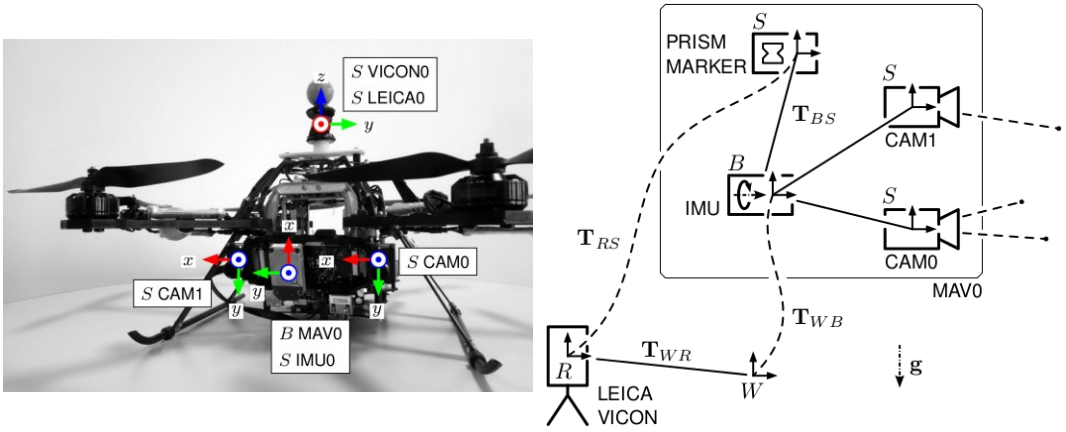
(c) sequences 03

(f) sequences 10

Figure 3.2: Path of selected KITTI sequences.

3.1.2 EuRoC Dataset

The EuRoC dataset [51] uses a Micro Aerial Vehicle (MAV) to collect 11 visual-inertial data sequences in two different indoor situations. The first batch of the dataset (MH01-MH05) are real-flight synchronized grayscale stereo images and IMU measurements collected in an industrial Machine Hall, with millimeter accurate ground-truth positions from a laser tracking system. The rest of the dataset (V101-V103, V201-V203) are collected in two Vicon Rooms equipped with a 6D motion capture system. In addition of synchronized grayscale stereo images, IMU measurements and 6D ground-truth poses, detailed 3D laser-scan point clouds are also contained. The data-collecting platform and its sensor setup are illustrated in Figure 3.3 and Table 3.3. All data sequences are leveled to easy, medium, or difficult according to flying speed, motion blur and illumination conditions. The EuRoC dataset assists researchers in both fields of VSLAM and 3D environment reconstruction.



(a) (b)

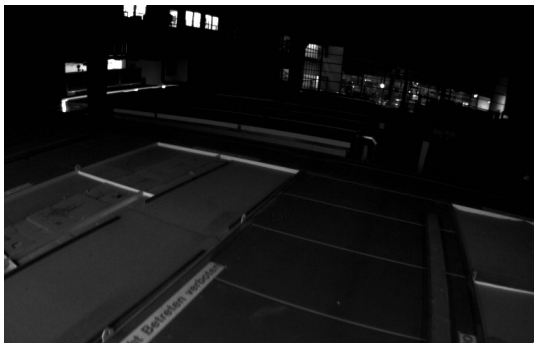
Figure 3.3: EuRoC data-collecting platform (a) and sensor setup (b) [28].

TABLE 3.3:
DATA-COLLECTING PLATFORM SENSOR SETUP OF EUROC

Data	**Sensor Type**	**Rate**	**Specifications**
Stereo Images (grayscale)	MT9V034 (x2)	20 Hz	WVGA, global shutter
3D Point Cloud	Leica MS50	20 Hz	Accuracy: ~1mm
IMU	ADIS16448	200 Hz	MEMS, intr. calibrated
Ground-Truth	Leica MS50	20 Hz	Accuracy: ~1mm
	Vicon	100 Hz	6D

In this section, the EuRoC sequences MH_01_easy (good texture, bright scene), MH_05_difficult (fast motion, dark scene) and V1_01_easy (slow motion, bright scene) are selected as indoor benchmarks of current VSLAM methods, as shown in Figure 3.4. The rectified stereo grayscale image sequences are used as input of VSLAM systems, and the provided ground-truth camera poses are used to evaluate their estimated trajectories of camera motion.

(a) MH01_easy (good texture, bright scene)



(b) MH05_difficult (fast motion, dark scene)

(c) V101_easy (slow motion, bright scene)

Figure 3.4: Selected EuRoC sequences.

3.2 Experiment Environment Setup

The available open-source codes of ORB-SLAM2, ProSLAM and OpenVSLAM are compiled and deployed on an Intel Core i5-7400 desktop with 8 GB RAM, and executed on selected benchmarks of KITTI and EuRoC. The rectified stereo grayscale image sequences of the selected KITTI and EuRoC sequences are used as input, and the output trajectory estimations of the VSLAM systems are evaluated corresponding to the provided ground-truth poses. Each benchmark is executed for five times to show their average performance in accuracy and efficiency considering the uncertainty of multi-thread.

The results of Stereo LSD-SLAM on the KITTI sequences are cited from [40] (executed on an Intel i7-4900MQ 2.8 GHz CPU), and the estimated poses of Stereo DSO on the KITTI sequences are cited from [42] (available on their official website). For the EuRoC sequences, only three indirect methods are executed and evaluated.

3.3 Analysis based on KITTI Benchmarks

3.3.1 Accuracy Evaluation

The average relative translation RMSE t_{rel} (%) and rotation RMSE r_{rel} (deg/100m) are calculated for accuracy comparison. Table 3.4 shows the accuracy comparison results (lower the better). And the best results are shown as bold numbers.

TABLE 3.4:
ACCURACY RESULTS ON SELECTED KITTI SEQUENCES

	ORB-SLAM2		ProSLAM		OpenVSLAM		St. LSD (official)		St. DSO (official)	
Seq.	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
00	0.7010	**0.25**	0.8174	0.30	0.7160	0.27	**0.63**	0.26	0.9606	0.30
01	**1.4288**	0.33	1.7687	0.30	3.9887	0.92	2.36	0.36	1.5402	**0.10**
03	0.7383	0.20	**0.6837**	0.30	0.7348	**0.17**	1.01	0.28	0.9967	**0.17**
04	0.4706	0.19	0.4981	0.13	0.4690	**0.11**	**0.38**	0.31	0.8945	0.17
06	**0.4983**	**0.15**	0.6698	0.18	0.5244	0.17	0.71	0.18	0.7898	0.23
10	0.6118	0.28	0.6642	0.21	0.6344	0.31	0.72	0.33	**0.4799**	**0.19**
mean	**0.7415**	0.23	0.8503	0.24	1.1779	0.33	0.97	0.29	0.9436	**0.19**

The mean translation and rotation RMSEs indicate that, ORB-SLAM2 outperforms other methods in translation estimations (0.7415% mean translation RMSE), while Stereo DSO has relatively better performance in rotation estimation (0.19 deg/100m mean rotation RMSE). To summarize the observation, direct methods show advantages in rotation estimation using illumination information, while indirect methods reinforce translation estimation by tracking features but weak in rotations.

The error values of sequence 01 in Table 3.4 show significant divergence compared to other sequences with larger RMSEs compared to other sequences. Figure 3.5 shows detailed pose estimations (a) and RMSE trends with respect to path length (b) and driving speed (c) of sequence 01. It can be seen from Figure 3.5 that, ProSLAM and OpenVSLAM deviate the ground truth in an early stage. With the increasement of driving speed, translational RMSEs of ProSLAM and OpenVSLAM raise significantly. Compared to other sequences, sequence 01 is the only highway and high-speed driving sequence with max driving speed at 96 km/h. The higher driving speed leads to higher differentiation between adjacent frames and larger estimation

errors, especially for the indirect methods with fewer matched and tracked features. Most images in this sequence are low-texture, with few buildings or trees along the high way, which also reduces the number of valuable features. Approaching the end of sequence 01, there is a high-speed sharp turn, which is also challenging to feature-based methods. ORB-SLAM2 achieves the best performance because of its feature classification strategy. Far features can be stably tracked for rotation estimation, while close features contribute to more accurate translation estimation. Direct methods are more suitable for this type of scenario, especially for rotation estimation. The official experimental data of Stereo LSD-SLAM and Stereo DSO always show lower or similar RMSEs than other methods in Figure 3.5 (b) and (c).

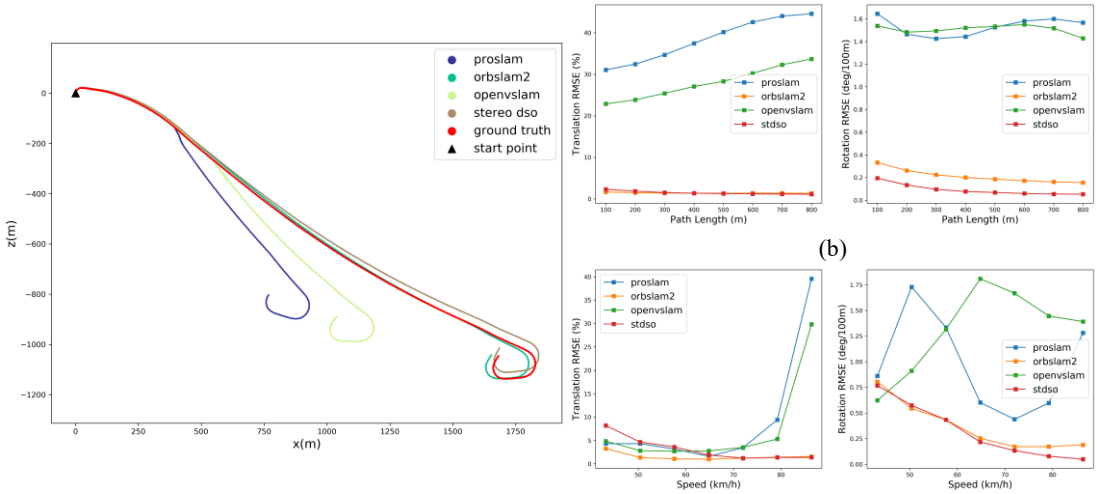


Figure 3.5: Qualitative and quantitative results of KITTI sequence 01. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

Qualitative and quantitative results of other KITTI sequences are also provided in Figure 3.6-3.10 at the end of this review. Sequence 00 is the most representative sequence of KITTI (see Figure 3.6) driving on inner-city roads with several loops. Indirect systems perform better than Stereo DSO. On sequences 03, 04 and 10

(shown in Figure 3.7, 3.8, 3.9, respectively), all indirect methods achieve comparable performances. These sequences are simple with driving speed lower than 60 km/h. Enough valuable features can be stably extracted and tracked. Sequence 10 is longer. Indirect systems accumulate error and get worse results than Stereo DSO. When it comes to sequence 06, also a simple low-speed driving sequence but with one loop (see Figure 3.10), ORB-SLAM2 achieves higher accuracy because its loop-detection module is triggered to erase the accumulated error, then a global optimization is also performed for better results. On the other hand, Stereo LSD-SLAM and Stereo DSO show worse translation estimations than the indirect methods on these sequences. The possible reason is that, these driving sequences are on roads inside cities and residential areas, where more valuable information can be acquired from features inside the captured images.

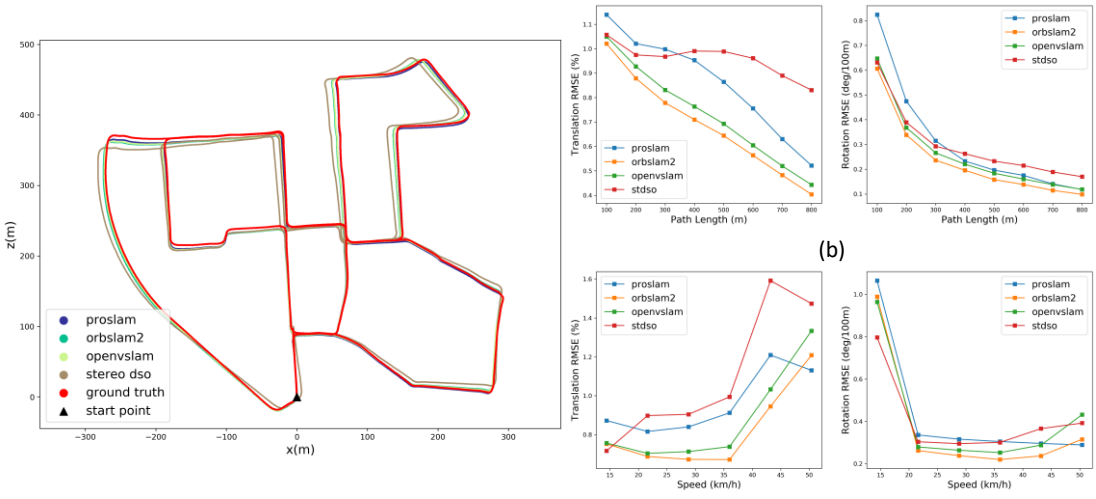


Figure 3.6: Qualitative and quantitative results of KITTI sequence 00. Ground truth (red) and estimated

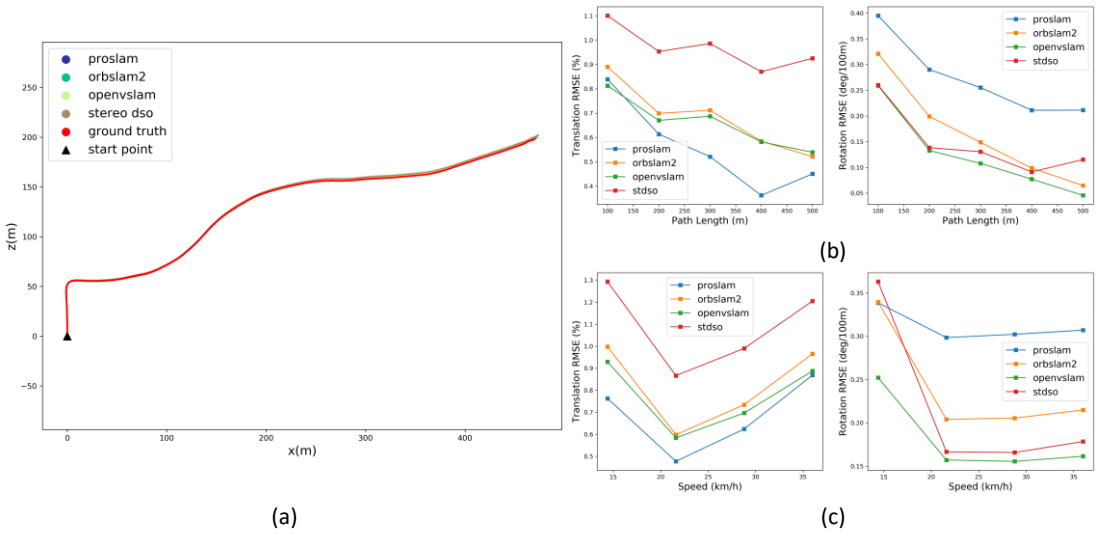


Figure 3.7: Qualitative and quantitative results of KITTI sequence 03. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

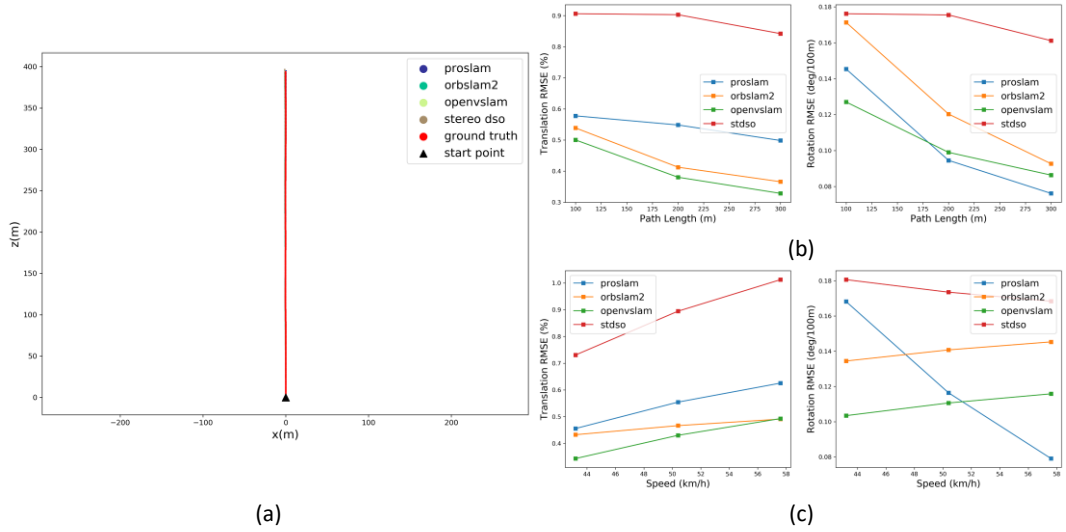


Figure 3.8: Qualitative and quantitative results of KITTI sequence 04. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

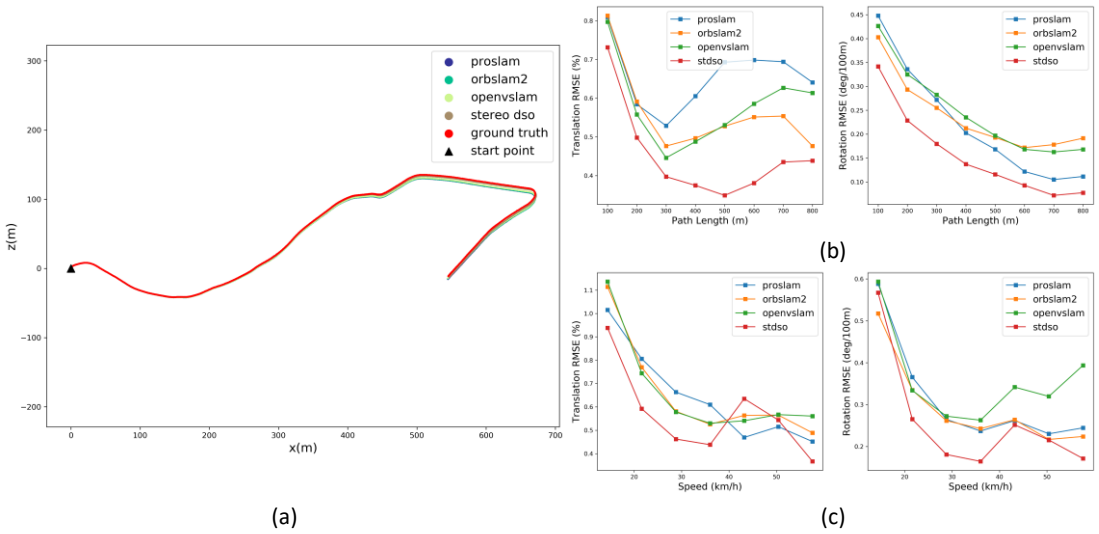


Figure 3.9: Qualitative and quantitative results of KITTI sequence 10. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

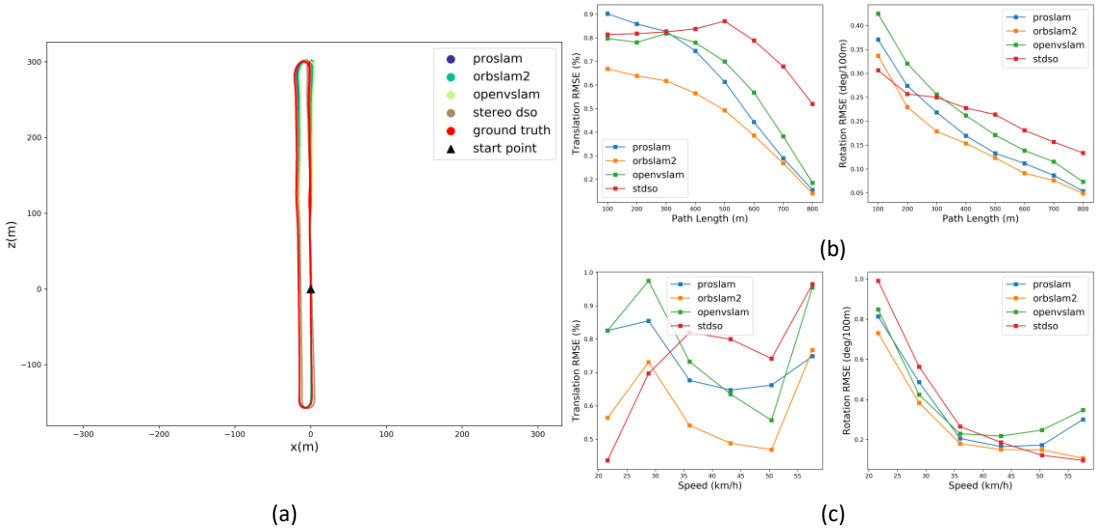


Figure 3.10: Qualitative and quantitative results of KITTI sequence 06. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

3.3.2 Efficiency Evaluation

Efficiency evaluation is carried out upon the available indirect methods by calculating the mean processing time in milliseconds per frame (lower the better), and hence get the frame rate (FPS, higher the better). The average results of three complete executions are stored in Table 3.5.

Table 3.5:
EFFICIENCY RESULTS ON SELECTED KITTI SEQUENCES

Seq.	Frames	ProSLAM		OpenVSLAM		ORB-SLAM2	
		time	*rate*	*time*	*rate*	*time*	*rate*
00	4541	31.3584	31.89	66.1369	15.12	91.0221	10.99
01	1101	16.0399	62.34	62.4548	16.01	110.5358	9.05
03	801	22.5245	44.40	68.3038	14.64	77.8148	12.85
04	271	16.5963	60.25	63.6313	15.72	81.7149	12.24
06	1101	17.5473	56.99	62.1988	16.08	92.6912	10.79
10	1201	22.7482	43.96	63.4279	15.77	73.4403	13.62
mean		21.1358	49.97	64.3589	15.56	87.8699	11.59

Obviously, ProSLAM takes the shortest mean processing time on every selected KITTI sequence. This is because ProSLAM only works on trajectory estimation with input image sequences one after another in a single thread, with no delay for synchronization to the input timestamps, nor real-time display of camera movement and generated map, as performed in other methods. ProSLAM has no further optimization, which also contributes to efficiency.

Taking input frame delay (around 10 ms per frame) and real-time display into consideration, the frame rates of ORB-SLAM2 and OpenVSLAM drop below 20 FPS. The two indirect methods both utilize ORB features for localization and mapping with similar system frameworks, while OpenVSLAM does not apply

motion-only BA to optimize pose estimation within the tracking module as in ORB-SLAM2. This makes OpenVSLAM averagely 23.5110 ms faster, with similar performance in most sequences. However, when it comes to a challenging scenario like Sequence 01, OpenVSLAM shows less robustness than ORB-SLAM2 (see discussions in the previous section).

There is always a trade-off between accuracy and efficiency. Methods like ProSLAM are more suitable for the scenario where computational resources and endurance are of primary consideration with simple driving conditions.

3.4 Analysis based on EuRoC Benchmarks

3.4.1 Accuracy Evaluation

Absolute Pose Error (APE) in meter is used for evaluation of trajectory estimations, lower the better. Umeyama's SE(3) transformation [52] is applied to align the estimated and ground-truth trajectories. Table 3.6 shows the mean and median APEs of the available indirect methods on selected EuRoC sequences.

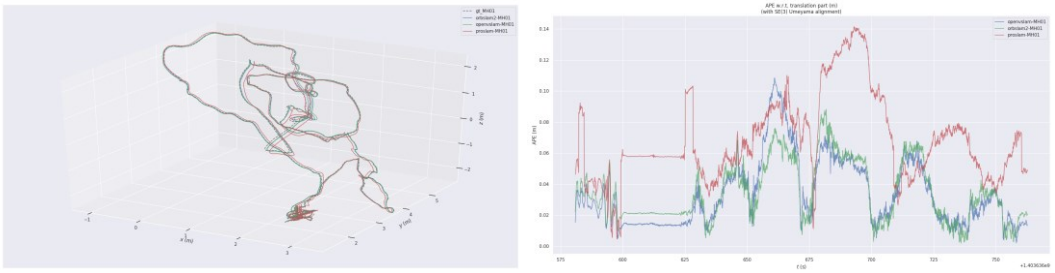
TABLE 3.6:
ACCURACY RESULTS ON SELECTED EUROC SEQUENCES

	ProSLAM		**OpenVSLAM**		**ORB-SLAM2**	
Seq.	***APE (mean)***	***APE (median)***	***APE (mean)***	***APE (median)***	***APE (mean)***	***APE (median)***
MH01	0.067832	0.058281	0.033546	0.026819	0.034592	0. 030058
MH05	0.123825	0.138871	0.050280	0.039510	0.051630	0. 042251
V101	0.131515	0.127050	0.082421	0.078223	0.082617	0.077822
mean	0.107724	0.108067	0.055416	0.048184	0.05628	0.077822

ProSLAM again shows limited performance on each sequence, with about twice APEs than other methods. This is reasonable as discussed previously that ProSLAM

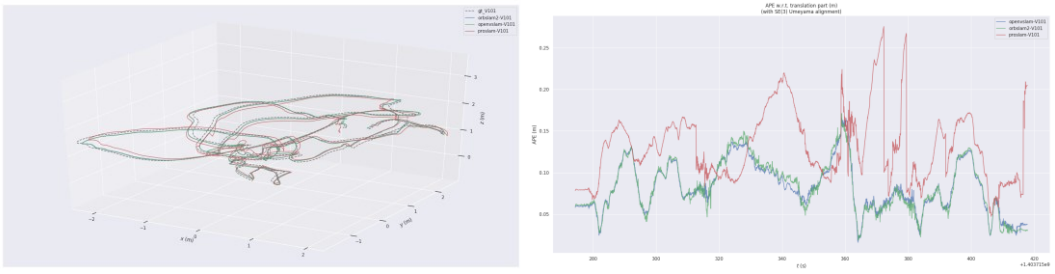
has the simplest structure and minimum optimization. ORB-SLAM2 and OpenVSLAM perform equivalently since they both apply ORB features and BA optimizations.

Figure 3.11-3.13 are the qualitative and quantitative results of the selected EuRoC sequences MH01, V101 and MH05, respectively. It can be observed from the comparison between the trajectory estimations and the ground-truth trajectories that, all the methods keep positive tracking to the real motion in all three sequences. This is because adequate features can be extracted and tracked in indoor scenarios, especially for the ORB feature extractor and descriptor.



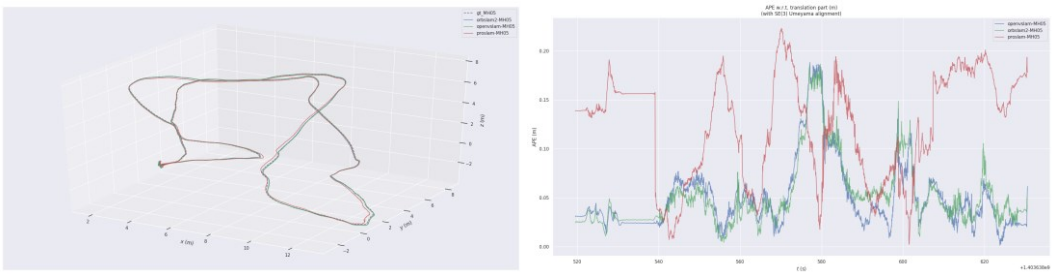
(a) (b)

Figure 3.11: Qualitative and quantitative results of EuRoC sequence MH01. Ground truth (dash line) and estimated trajectories (other colors) (a) and APE with SE(3) Umeyama alignment (b).



(a) (b)

Figure 3.12: Qualitative and quantitative results of EuRoC sequence V101. Ground truth (dash line) and estimated trajectories (other colors) (a) and APE with SE(3) Umeyama alignment (b).



(a) (b)

Figure 3.13: Qualitative and quantitative results of EuRoC sequence MH05. Ground truth (dash line) and estimated trajectories (other colors) (a) and APE with SE(3) Umeyama alignment (b).

MH05 is a sequence of difficult level because the MAV moves fast in dark scenes. Telling form Figure 3.13 (b), ORB feature extractor shows better invariance to illumination conditions. Another issue is that, compared to KITTI sequence 01, although the platform also moves at a high speed, the cameras in EuRoC sequences are working at a higher frame rate (20 Hz in EuRoC sequences to 10 Hz in KITTI sequences), which reduces the differation between frames so that feature tracking can be more stable.

3.4.2 Efficiency Evaluation

TABLE 3.7:
EFFICIENCY RESULTS ON SELECTED EUROC SEQUENCES

Seq.	Frames	ProSLAM		OpenVSLAM		ORB-SLAM2	
		time	*rate*	*rate*	*rate*	*time*	*rate*
MH01	3682	56.102	17.825	116.090	8.614	200.350	4.991
MH05	2273	45.554	21.952	110.252	9.070	175.398	5.701
V101	2912	44.461	22.492	116.777	8.563	136.183	7.343
mean		48.706	20.756	114.373	8.749	170.644	6.012

Efficiency evaluation is carried out upon the available indirect methods by calculating the mean processing time in milliseconds per frame (lower the better),

and hence get the frame rate (FPS, higher the better). The average results of three complete executions are stored in Table 3.6.

All three methods use longer mean processing time compared to KITTI sequences. This is mainly because the camera in EuRoC is mounted on an UAV and moves in 3D space, which has six degrees of freedom. Similar to the results of the selected KITTI sequences, ProSLAM takes the shortest mean processing time (regardless of input delay) for each sequence, and ORB-SLAM2 gets the lowest frame rates. In these cases, OpenVSLAM achieves slightly higher processing speed than ORB-SLAM2 as well as equivalent accuracy.

3.5 Discussions

The above experimental results indicate that, indirect VSLAM systems generally achieve higher accuracy yet with lower efficiency. In each frame, features of image are extracted and their descriptors are calculated. The system than matches the features according to their descriptors and get the relationship between features in different frames corresponding to same real-world 3D landmarks. This results in more accurate pose estimation, higher tracking and mapping quality, and also benefits the loop closure and optimization modules such as BA. The feature classification strategy of ORB-SLAM2 also improves robustness in low-texture environments. However, the performance of indirect methods relies on the quality of features. Feature processing is also time-consuming, which may become a limitation when low latency is enforced or the processor is less powerful. Also, ORB-SLAM2 optimizes each camera pose at least twice in each frame, which improves accuracy but costs longer processing time at the same time.

Direct methods, on the other hand, do not extract features or descriptors. Thus, they can theoretically more efficient and can be executed with low-texture images. However, the accuracy performance of the direct methods is not guaranteed, especially when the illumination condition varies remarkably. Another shortage is that, these methods are visual odometry, which only focus on localization and do not cooperate with mapping. They only optimize motion without structure, and have no loop detection. This also limits their performance.

Efforts have been made to deal with the deficiencies while retain the merits.

Kuang, Wang et al. [53] proposed an improved method based on ORB-SLAM that applies a quasi-physical sampling algorithm based on BING features[54] combined with depth information to reduce the scale of the features to be extracted, and optimizes the matching strategy using an improved KD-Tree[55], which consequently reduce the processing time.

Zhetao Zhang and Wanggen Wan proposed DOVO[56], a mixed visual odometry based on direct method and ORB feature. They set a threshold K for the number of ORB features computed from an image frame, to determine either using feature-based pose estimation or direct tracking method.

Forster, Zhang et al.[57] came up with the idea of semi-direct VO (SVO) that uses direct methods to track and triangulate pixels characterized by high image gradients, but relies on proven indirect methods for joint optimization of structure and motion. SVO achieves exceptionally fast processing speed, requiring only 2.5 milliseconds

to estimate the pose of a frame on a standard laptop computer, while achieving comparable accuracy with respect to the state of the art on benchmark datasets.

Our proposed system is based on the idea of semi-direct. Detailed system design is explained in the next Chapter.

Chapter 4. System Design

4.1 Inspirations

In the previous chapters, we have evaluated five state-of-the-art Stereo VSLAM systems, and concluded that, indirect VSLAM methods generally outperform direct methods in accuracy, among which ORB-SLAM2 achieves the highest accuracy in most benchmarks. Nevertheless, the limitation exists in efficiency since ORB-SLAM2 extracts features and calculates descriptors in each frame, which is computationally intensive. Direct methods, on the other hand, exploit illumination information for pose estimation and perform better in terms of efficiency, yet the accuracy and robustness may not be guaranteed and there is still a lack of mature solutions on joint optimization of motion (camera poses) and structure (landmarks) for direct methods.

SVO [57] is designed with the idea that combine the advantage of direct methods in pose estimation efficiency, with feature-based bundle adjustment, joint optimization of motion and structure utilized by indirect methods, and achieves faster processing speed along with competitive accuracy in tracking and localization.

Inspired by the idea of semi-direct, we propose a semi-direct stereo VSLAM system based on the integration of direct tracking method and feature-based optimization techniques. It utilizes direct image alignment to estimate camera pose, and further perform feature-based optimizations on camera poses and 3D landmarks. As a complete VSLAM system, it also maintains a 3D point cloud map and allows loop closing and relocalization after tracking lost.

4.2 System Overview

A general workflow of our system is shown in Figure 4.1.

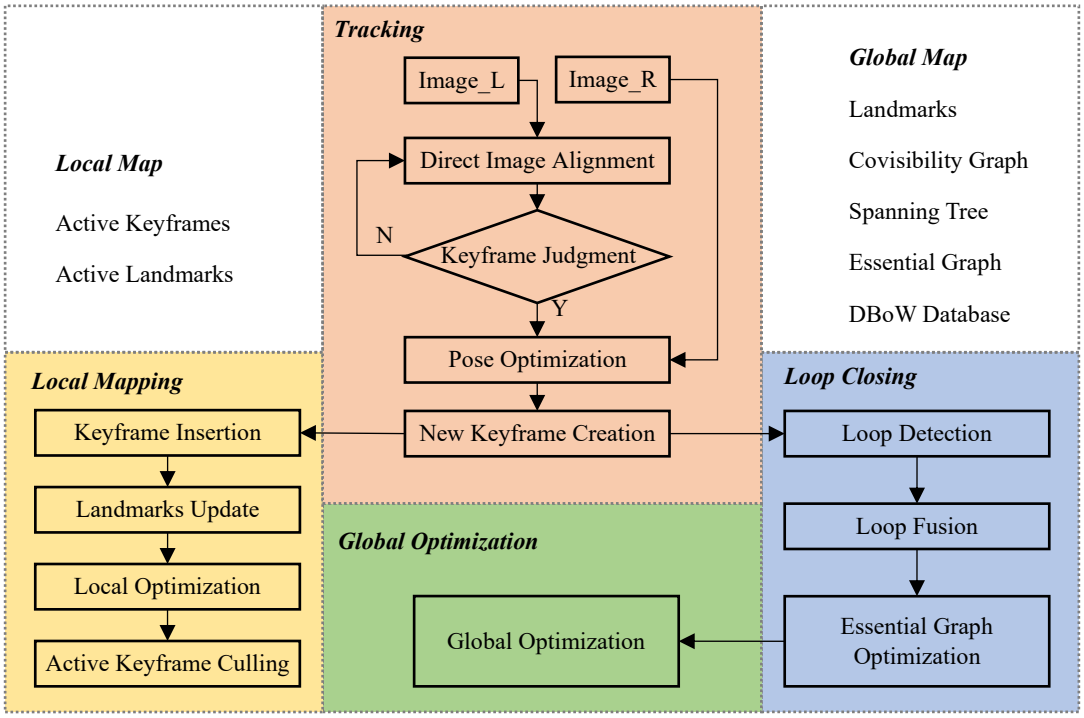


Figure 4.1: System workflow with four parallel threads:
tracking (red), local mapping (yellow), loop closing (blue) and global optimization (green).

The system has four main parallel threads: tracking, local mapping, loop closing and global optimization. Tracking estimates camera poses and decides whether to insert a new keyframe. A new keyframe is sent to local mapping to generate map and loop closing to check if a loop occurred. Global optimization is only triggered after a successful loop closing.

The main difference of our system is that, we combine direct methods and indirect methods in tracking and optimization modules, and extend VO into VSLAM.

- Tracking

Estimates camera pose of every frame by constant motion model and direct image alignment. If the tracking is lost (e.g., due to occlusions or abrupt movements), the place recognition module is used to perform a global relocalization. At last, it judges whether the current frame should be inserted as a new keyframe. If a new keyframe is needed, features of both left and right images are extracted features of both left and right images are extracted and matched. 3D landmarks, also called map points, are triangulated from the matched features. In this way, correspondences are established between the features, landmarks and the keyframe. Then feature-based pose estimation is performed to increase accuracy. Once a new keyframe is created, local mapping and loop closing are triggered.

- Local mapping

Maintains and manages keyframes and landmarks. New keyframe and landmarks are inserted and updated, then local BA [30] is performed to jointly optimize camera poses of local keyframes and 3D coordinates of local landmarks. The covisibility graph, spanning tree and essential graph are also maintained and updated in this thread with inserted/culled keyframes and landmarks. A Bag-of-Words (BoW) database is also updated with each new keyframes for further place recognition (Section 4.3-H).

- Loop Closing

By comparing BoW of the new keyframe with all the historical keyframes in the covisibility graph, the system determines whether the current pose coincides with a previous spot (loop detection and decision). If a loop is detected, both sides of the loop are aligned to eliminate the accumulated drift (loop closure) and trigger a global optimization.

- Global Optimization

In global optimization, all camera poses and landmarks (except the initial one) are optimized with global BA over the essential graph. This is the most time-costly module and runs in a separate thread. Global BA is only launched after a loop is closed or a global relocalization after tracking lost. If global BA is triggered again while a previous global BA is in progress, the former execution would be suspended immediately.

4.3 Notions

A. Feature (x)

A feature is a certain keypoint at coordinates (u, v) of an image with a descriptor d calculated from its neighboring pixels that describes certain properties of the feature to identify it. For example, corners and edges with high image gradient in an image are remarkable features that contains structure information corresponding to real-world 3D points.

SIFT [58], SURF [59], A-KAZE[60] and LDB [61] are several popular feature extractor and descriptor in the field of computer vision and image processing. However, they are either time consuming or inferior in rotation, which limit their performance in real-time VSLAM. Thus, we use ORB [46], oriented multi-scale FAST corners associated with 256-bit rotated BRIEF descriptors. According to the author, it takes only ~15.3ms to generate (extract and describe) ~1000 ORB features while SIFT and SURF need ~ 5228.7 ms and ~217.3 ms respectively. ORB features s also robust to rotation and scale, and present a good invariance to camera auto-gain and auto-exposure.

B. Feature Classification

Having features of stereo images, the next step is to match features in the left image to those in the right image by calculating the Hamming distance between two descriptors. This can be done efficiently with undistorted and rectified stereo images since the matched features always lies in the same horizontal epipolar line. As illustrated in [37], we classify all the features into stereo/mono features according to matching results.

A stereo feature is a pair of matched features of the stereo images, which means the two features correspond to the same real-world 3D point. It is represented by three coordinates $x_s = (u_l, u_r, v)$, where (u_l, u_r) are the horizontal coordinates on the left and right images, v is the shared vertical coordinate. The depth Z of the 3D point can be easily triangulated with known image coordinates and stereo camera intrinsic. A stereo feature is further classified as close/far upon the associated depth less/more than $40b$, being b the length of baseline of the stereo camera, as suggested in [62].

Close features contribute to accurate estimation of scare, translation, and rotation information, while far features contribute more to rotation estimation.

On the contrary, a mono feature $x_m = (u_l, v_l)$ is a feature on the left image without a match on the right image. These features only contribute to rotation and translation estimation in multiple frames.

C. Frame (F) and Keyframe (K)

The frame is the basic unit of the system. At each frame, stereo input images are processed and camera pose T is estimated using the direct method in the tracking module (See Section 4.5). Then the frame is checked with specific conditions to be decided as a keyframe.

When a new keyframe is inserted, the local mapping thread and loop closing thread are triggered and more detailed information in the keyframe is extracted to update all related system modules. The system always attempts to insert keyframes as fast as possible, and redundant keyframes are removed at last by a culling policy, specified in Section 4.6.

D. Camera Pose (T)

The goal of localization in VSLAM is to estimate accurate camera pose presented as transform matrix $T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \in SE(3)$ [63], where $R \in SO(3)$ is rotation matrix and $t \in \mathbb{R}^3$ is transformation vector that describe the 6DoF camera motion.

E. Landmark

The map generated by the system is consists of keyframes and landmarks. A landmark is "observed" by several keyframes, while a keyframe is "observing" several landmarks.

The 3D coordinates of a landmark in the world coordinate are presented as $P_w = \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$. A real-world 3D landmark can be projected to a matched feature on the left image with the following equation:

$$x_{(.)} = \pi_{(.)}(RP + t) \tag{1}$$

where $\pi_{(.)}$ is the 3D-2D projection function in the camera coordinate system, π_s for stereo features x_s and π_m for mono features x_m:

$$x_s = \pi_s\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}\right) = \begin{bmatrix} f_x \frac{X}{Z} + c_x \\ f_x \frac{X - b}{Z} + c_x \\ f_y \frac{Y}{Z} + c_y \end{bmatrix}, x_m = \pi_m\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}\right) = \begin{bmatrix} f_x \frac{X}{Z} + c_x \\ f_y \frac{Y}{Z} + c_y \end{bmatrix} \tag{2}$$

where (f_x, f_y) is the focal length, (c_x, c_y) is the principal point, and b is the length of the baseline of the stereo camera. The back-projection function from a feature to a 3D landmark is presented as $P = \pi^{-1}(x)$.

F. Covisibility Graph, Spanning Tree and Essential Graph

Covisibility Graph [64] is an undirected weighted graph where each node is a keyframe and an edge between nodes shows the covisibility relationship between the

two keyframes with the weight θ of the edge indicating the number of common observed landmarks ($\theta \geq 15$).

Spanning Tree is a connected subgraph of the covisibility graph with minimal number of edges. With the initial keyframe as the root node, a new keyframe is always connected to the node sharing the most landmarks.

The number of edges in a covisibility graph is predictably large, which is costly to pose graph optimization [65]. Thus, a simplified Essential Graph is constructed with all the nodes (keyframes) in the covisibility graph but only representing edges retained that maximally preserve the information in a smaller scale, achieving efficient and qualified optimization. It contains the spanning tree and the subset of edges from the covisibility graph with high covisibility ($\theta \geq 100$).

G. Bundle Adjustment (BA)

The estimation of camera pose (motion) and landmark (structure) comes with unavoidable error from input noise, mathematical system error, accumulate drift, etc. Thus, optimization should be applied for better results. One mature way is to jointly optimize both motion and structure.

Stereo features in a new keyframe can be matched to existing landmarks with descriptors. If those matched landmarks are projected into the new keyframe with initial estimation of camera pose using Eq. (1) in the previous section E, there is a reprojection error between the real position and projected position of the features. Minimizing the reprojection error is the basic idea of bundle adjustment [30] and hance optimize both camera pose and landmarks.

Our system applies a motion-only BA to optimize the camera pose of a new keyframe, a local BA to optimize a local window of keyframes and landmarks in the local mapping thread (Section 4.6) and a global-BA after loop closing.

H. BoW Place Recognition

Visual Vocabulary is a dictionary of discrete visual words representing various feature descriptors extracted from a general image set. An image can be converted in to a vector of visual BoW over a specific visual vocabulary for dimension reduction, and the similarity of different images is easily quantified by comparing the BoWs over the same vocabulary, which is an efficient way of place recognition for loop detection, loop decision, and global relocalization.

DBoW2 [47] is used to achieve a bag-of-words place recognition. The system maintains a database of invert indexes for each visual word in the vocabulary, indicating in which keyframes the visual word has been seen, making it efficient to query the database. Because of the visual overlap between keyframes, no keyframe can get a best score with high divergency to the others when querying the database. The original DBoW2 deal with the problem by adding up the score of keyframes that are close in time, which has the limitation of not including keyframes viewing the same place but inserted at a different time. Instead, all connected keyframes in the covisibility graph are grouped and keyframes scoring top 25% are returned.

DBoW2 is also used for efficient computation of correspondence between ORB sets, such as feature to feature or feature to land mark. It applies a brute force matching

constrained only to those features belonging to the same node in the vocabulary tree at a certain level (the second out of six in our method), as described in [47].

4.4 System Initialization and Keyframe Pre-Processing

The first frame is also selected as the first keyframe for system initialization, being the first node of covisibility graph, spanning tree and essential graph.

For each frame selected as a keyframe, a proper number of ORB keypoints are extracted from both left and right stereo input images according to the image resolution and quality. We prefer ~1000 keypoints for low-resolution images as in the EuRoC dataset [51] and ~2000 for large images of the KITTI dataset [49]. The system divides the image into grids and tries to extract at least 5 keypoints per grid for a homogeneous distribution, then corresponding ORB descriptors are computed on the retained keypoints. Then we can get stereo features as illustrated in Section 4.3-B, and triangulate corresponding 3D landmarks to create an initial map.

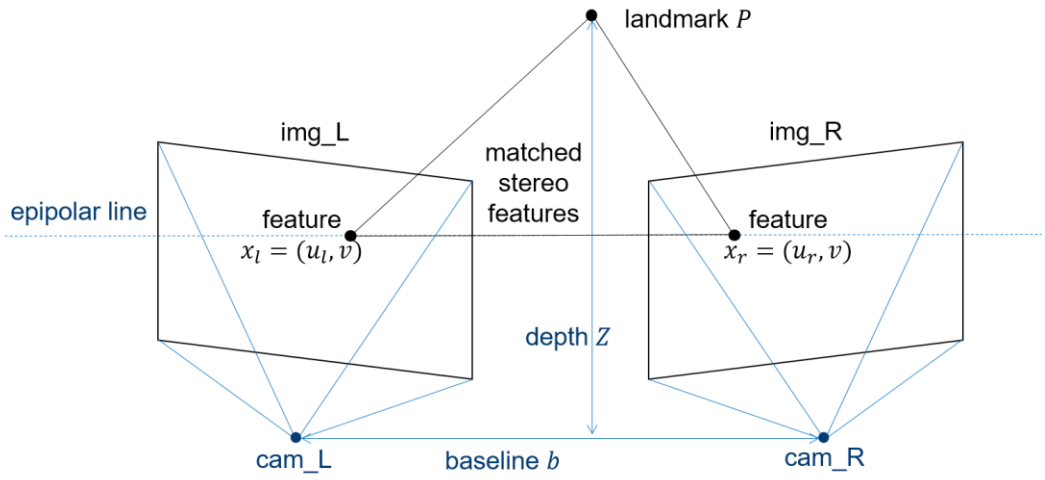


Figure 4.2: Triangulate landmarks from stereo features.

4.5 Tracking

4.5.1 Direct Image Alignment

Assuming the illumination conditions of the input frames remain constant, pixels observing the same landmarks in different frames are supposed to have the same intensity values, which means the photometric error should be zero. While in real world, this idle situation cannot be guaranteed. Thus, the objective of direct image alignment is to minimize the photometric error of pixels observing the same landmarks.

For a new stereo input frame, an initial camera pose estimation is acquired from a constant motion model. Then the system applies direct image alignment to refine the camera pose estimation. Active landmarks P_x of a reference keyframe are projected in to the new frame and the optimization energy function is:

$$\{R,t\} \underset{R,t}{\operatorname{argmin}} \sum_{x\in\mathcal{p}} \| I(\pi(RP_x + t)) - I(\pi(P_x)) \|_{\Sigma}^2 \tag{3}$$

where $I(x)$ is the intensity value at pixel in the frame. To improve robustness, our method aggregates the photometric cost in a small patch $\mathcal{p}$ centered at feature x assuming the neighboring pixels has the same depth to the feature. The solution of this optimization function is the estimation of pose transform matrix $T_{cr} = \{R,t\}$ from the reference keyframe to the current frame. Then the current camera pose is acquired according to the pose of the reference keyframe.

This optimization for direct image alignment, and BA in other modules, are solved with the standard iterative non-linear least squares algorithms such as Levenberg-Marquardt implemented in g2o [34].

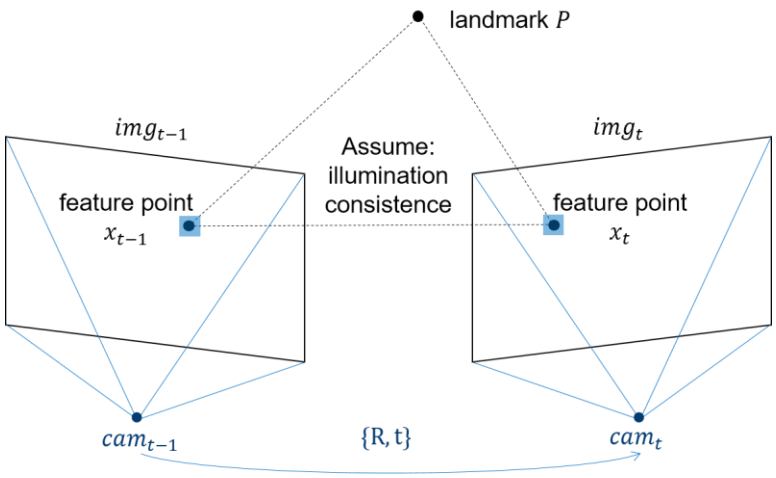


Figure 4.3: Direct image alignment, minimizing photometric error.

4.5.2 Keyframe Judgment

The last step is to judge if the current frame meets the requirements of a keyframe:

(1) Local mapping thread is free or enough frames (depends on input framerate) have passed from last keyframe insertion.

(2) Enough frames have passed from the last global relocalization (loop closure and lost relocalization).

(3) Current frame has a sufficient illumination change compared to the latest keyframe in the active window.

(4) Current frame has a sufficient scene change (indicated by the mean squared optical flow) compared to the latest keyframe in the active window.

If a keyframe is inserted when the local mapping is busy, a signal is sent to suspend local BA to insert keyframes as fast as possible, because that makes the tracking more robust to challenging camera movements, typically rotations.

4.5.3 Pose Optimization

When a frame is decided to be a new keyframe, the system extracts feature from both left and right images and compute their descriptors to match them and get stereo features. These stereo features are triangulated to be landmarks and searched for matches within landmarks of a reference keyframe. Then a local keyframe set is created with those (1) observing same landmarks to the new keyframe or (2) adjacent neighbors in the covisibility graph to the keyframes in (1). All landmarks of each keyframe in the local keyframe set become a local map, and more matched are searched between landmarks of the new keyframe and the local map.

Now the new keyframe has an initial pose estimation $\{R, t\}$ from direct image alignment, and an initial set of correspondences between stereo features X and landmarks P_X. Projecting P_X from world coordinates to the camera plane using camera pose $\{R, t\}$ produces a reprojection error between projected coordinates and the coordinates of matching features X. By minimizing the reprojection error, camera pose is optimized. This is the basic idea of Motion-only BA:

$$\{R, t\} = \underset{R,t}{\operatorname{argmin}} \sum_{x \in X} \rho \| x - \pi(RP_x + t) \|_{\Sigma}^2 \tag{4}$$

where ρ is the robust Huber cost function.

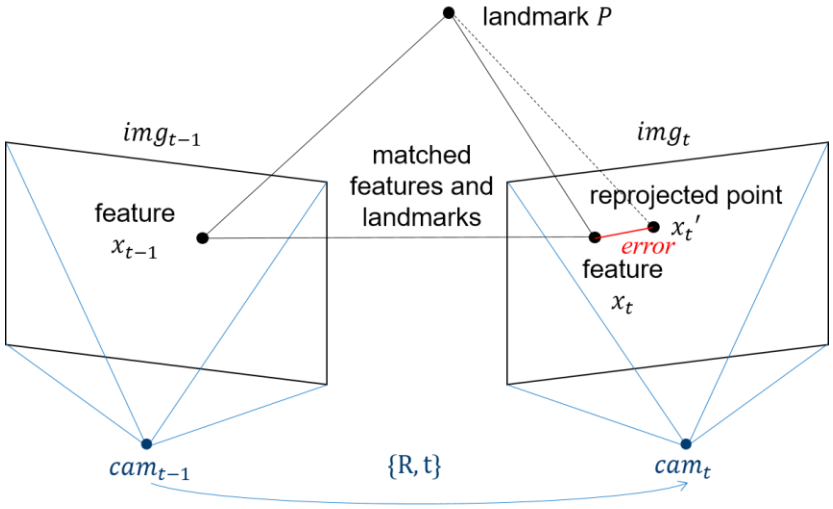


Figure 4.4: Pose optimization, minimizing reprojection error.

4.5.4 Global Relocalization

If the tracking is lost, the current frame is converted into a new keyframe and use BoW to query the recognition database for similar keyframes for global relocalization. The correspondences with ORB descriptors of features in the current keyframe and those associated to the landmarks in each candidate keyframe are calculated as described in Section 4.3-H, then alternatively RANSAC iterations are performed to each keyframe using the PnP algorithm [66] for camera pose estimation. A pose with enough inliers is optimized and a guided search is performed for more matches with the landmarks in the candidate keyframes. Finally, the camera pose is again optimized, and if supported with enough inliers, tracking is recovered.

4.6 Local Mapping

4.6.1 Keyframe Insertion

When a new keyframe is created, the local mapping thread and loop closing thread are triggered. The new keyframe is inserted as a new node in to the covisibility

graph updating edges connecting to other nodes with weight of shard landmarks' amount, and in to the spanning tree linked to the node with the most common landmarks. Then the BoW of the new keyframe is computed and stored in the recognition database for global relocalization.

4.6.2 Landmarks Update

After every new keyframe inserted, landmarks inserted within the past three keyframes are tested to ensure they are trackable and not wrongly triangulated with two conditions: 1) A landmark must be observable to at least three keyframes. 2) The tracking must find the landmark in more than the 25% of the frames in which it is predicted to be visible. A landmark is culled if any of the conditions is not satisfied during the test.

New landmarks are then created by triangulating features from connected keyframes in the covisibility graph. This should be easy since the depth is known for the stereo features. The unmatched features are matched to other features and landmarks in other keyframes using the same correspondence strategy as in Section 4.3-H.

4.6.3 Local BA

Local BA optimizes a set of connected keyframes in the covisibility graph $\mathcal{K}_L$ and all landmarks $\mathcal{P}_\mathrm{L}$ observed by those keyframes. All other keyframes $\mathcal{K}_F$, not in $\mathcal{K}_L$, observing points in $\mathcal{P}_\mathrm{L}$ contribute to the cost function but remain fixed in the optimization. Defining $\mathcal{X}_k$ as the set of matches between landmarks in $\mathcal{P}_\mathrm{L}$ and features in a keyframe k, the optimization problem is:

$$\{P^i, R_l, t_l \mid i \in \mathcal{P}_L, l \in \mathcal{K}_L\} = \underset{P^i, R_l, t_l}{\operatorname{argmin}} \sum_{k \in \mathcal{K}_L \cup \mathcal{K}_F} \sum_{j \in \mathcal{X}_k} \rho(E_{kj}) \quad (5)$$

$$\text{with } E_{kj} = \|x^j - \pi(R_k P^j + t_k)\|_\Sigma^2.$$

4.6.4 Local Keyframes Culling

As the fast insertion of keyframes, the scale of the map and the covisibility graph grows, and the complexity of BA increases as well. The system controls the scale of the keyframes by culling the keyframes, whose 90% landmarks have been seen in other at least three keyframes. This policy was inspired by the one proposed in the work of Tan et al. [67], where keyframes were discarded after a process of change detection.

4.7 Loop Closing

4.7.1 Loop Detection

Firstly, the minor-neighbors to K^i, those keyframes with edge weight $\theta \geq 30$ in the covisibility graph, but not directly connected in the spanning tree, are selected. Their similarity scores to K^i over the BoW vectors are calculated with the lowest score marked as s_{min}. Then, by querying all historical keyframes in the recognition database, keyframes with score lower than s_{min} are discarded. Within the remaining set of keyframes, every three keyframes that are consistent (connected in the covisibility graph), is considered as a loop candidate to K^i.

Loop decision is quite similar to global relocalization. Firstly, correspondences between ORB associated to landmarks in K^i and the loop candidate keyframes are

computed as in Section 4.3-H, and these 3D-3D correspondences for each loop candidate keyframe K^l are used to calculate a rigid body transformation $T_{il} \in SE(3)$. A transformation with enough inliers is optimized and a guided search is performed for more matches with the landmarks in the candidate keyframe. Finally, the transformation is again optimized, and if supported with enough inliers, K^l and T_{il} is determined for loop fusion.

4.7.2 Loop Fusion

The duplicated landmarks in the loop are fused and new edges are added to K^l according to K^i in the covisibility graph. The keyframe pose T^i is fused with T_{il} and the transformation is also cascaded to all the neighbors in the covisibility graph to align both sides of the loop. All landmarks observed by K^l-centered keyframes in the covisibility graph are projected into K^i for more landmark correspondences, and the edges of all related keyframes are updated as well.

4.7.3 Essential Graph Optimization

Finally, pose graph optimization distributes the loop closing error along the essential graph, and each landmark is transformed according to the fusion of a representing keyframe observing it.

4.8 Global Optimization

Global optimization is triggered after loop closing to optimize the fused camera poses and map. It applies global-BA, using the same equation as local-BA with extended scale of all keyframes and landmarks in the essential graph.

Global optimization is very time-consuming so it is launched alone in a new thread with a low priority. It will be blocked from running if a new loop closing is performed, then run again with newest essential graph.

After a global-BA, the correction is also propagated to the newly inserted keyframes and their landmarks during the period of global-BA.

Chapter 5. System Validation

5.1 Experiment Environment Setup

As introduced in Chapter 3, six selected sequences from the KITTI dataset are used as input to evaluate the performances of our proposed system. The rectified stereo grayscale image sequences are used as input, and the output camera pose estimations are evaluated with the provided ground-truth poses. All experiments are carried out on an Intel Core i5-7400 desktop with 8 GB RAM. Each sequence is executed for five times to show their average performance in accuracy and efficiency considering the uncertainty of multi-thread. Results of ORB-SLAM2 [37] (acquired from self executions, refer to Chapter 3 for more information) and Stereo DSO [41, 42] (pose estimations are cited from the official website).

Our experiments are performed as follows. Firstly, we implement a direct image alignment method to acquire initial estimations of camera poses. Then we apply feature-based bundle adjustment to optimize the estimations and get final results. The results are illustrated in the following sections.

5.2 Accuracy Validation

The average relative translation RMSE t_{rel} (%) and rotation RMSE r_{rel} (deg/100m) are calculated for accuracy evaluation on selected KITTI sequences. Table 5.1 shows the accuracy results (lower the better).

TABLE 5.1:
ACCURACY RESULTS ON SELECTED KITTI SEQUENCES

Seq.	ORB-SLAM2		St. DSO (official)		Direct Estimation		After BA	
	t_{rel}	t_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
00	0.7010	0.25	0.9606	0.30	2.2358	0.26	2.2323	0.29
01	1.4288	0.33	1.5402	0.10	2.8207	0.14	2.8105	0.42
03	0.7383	0.20	0.9967	0.17	4.3532	0.16	0.7576	0.19
04	0.4706	0.19	0.8945	0.17	1.6318	0.15	0.4940	0.15
06	0.4983	0.15	0.7898	0.23	2.7962	0.44	1.5693	0.48
10	0.6118	0.28	0.4799	0.19	0.8221	0.18	0.6385	0.32
mean	0.7415	0.23	0.9436	0.19	2.4433	0.22	1.417	0.31

The results show that, the initial estimations from direct image alignment are generated with large error. In most cases, feature-based optimizations can erase the error and significantly improve the accuracy.

In simple sequences 03, 04 and 10, our proposed method achieves similar accuracy to ORB-SLAM2. Although the initial estimation from direct image alignment is bad, since the camera moves slow and smooth without sharp turns, enough features can be tracked to improve the accuracy. Qualitative and quantitative results are shown in Figure 5.1 to Figure 5.3.

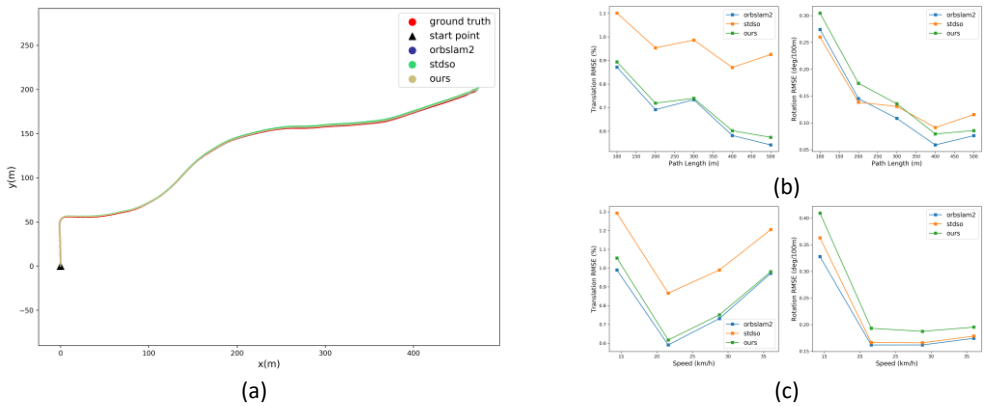


Figure 5.1: Qualitative and quantitative results of KITTI sequence 03. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

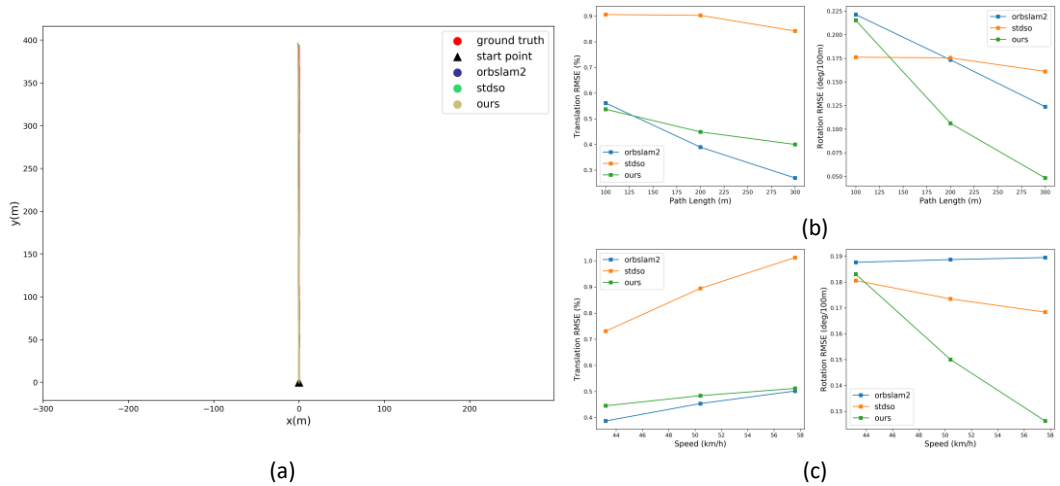


Figure 5.2: Qualitative and quantitative results of KITTI sequence 04. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

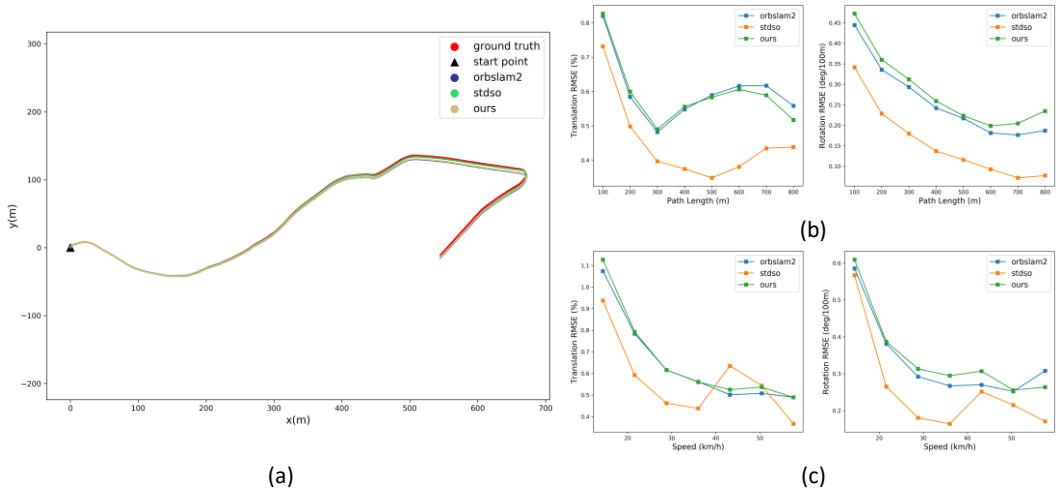


Figure 5.3: Qualitative and quantitative results of KITTI sequence 10. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

As for sequences 00 and 06, in some of the executions, tracking is lost because of the bad initial estimation. Tracking lost usually happens when the vehicle is making a sharp turn at intersections in sequence 00 or a U-turn in sequence 06. Usually, the relocalization module is able to recover tracking when loop occurs. However, bundle adjustment is not performed while tracking is lost. In sequence 06, tracking is recovered soon after the lost and only a small proportion of frames are lost. Bundle

adjustment is able to optimize the trajectory to some extent, yet still worse compared to ORB-SLAM2 and Stereo DSO. When it comes to sequence 00, once tracking is lost, it takes a long time before the vehicle returns to a mapped spot. Thus, bundle adjustment contributes zero to trajectory optimization, which means our method is failed in this case. Similar problems also exist in sequences 01, where bundle adjustment cannot work properly, even decreases the accuracy in rotation estimations.

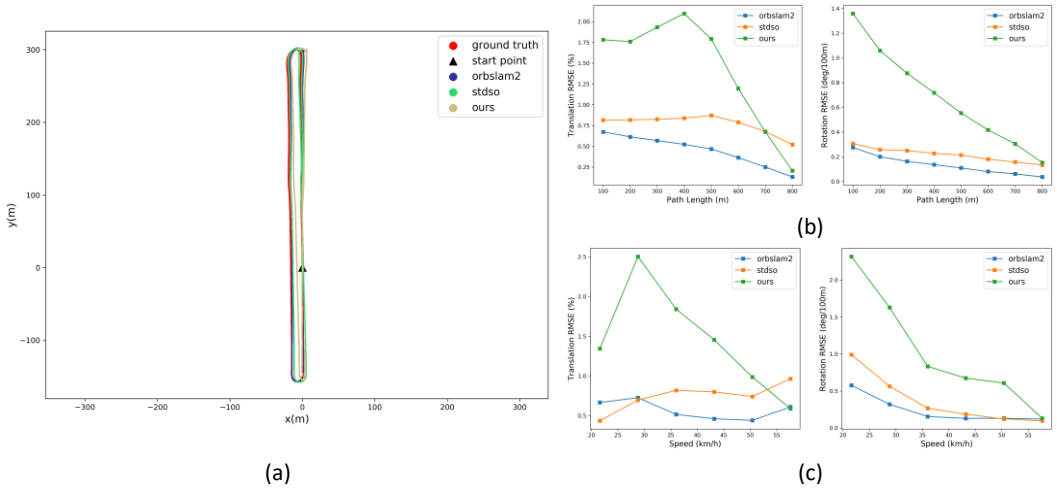


Figure 5.4: Qualitative and quantitative results of KITTI sequence 06. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

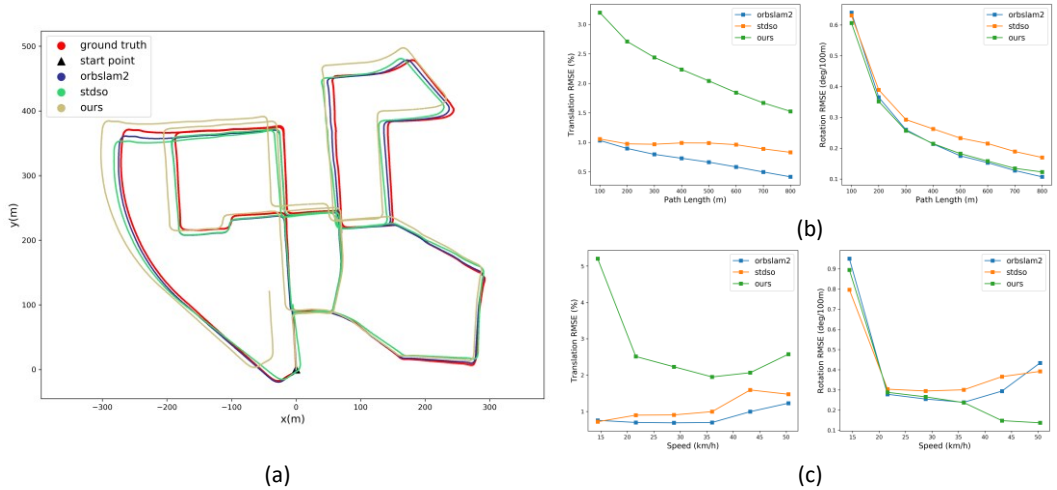


Figure 5.5: Qualitative and quantitative results of KITTI sequence 00. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

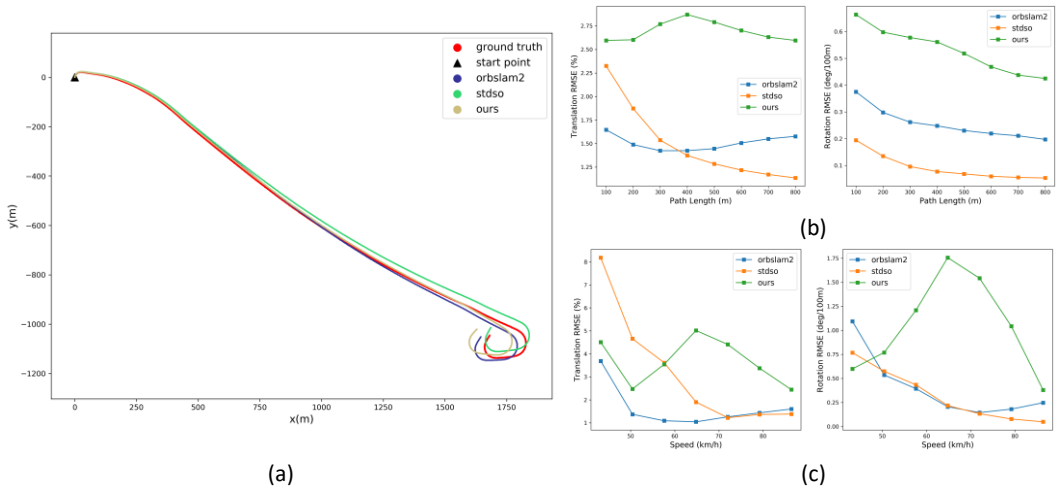


Figure 5.6: Qualitative and quantitative results of KITTI sequence 10. Ground truth (red) and estimated trajectories (other colors) (a). Average translation and rotation RMSEs over 100m to 800m intervals, with respect to path length (b) and driving speed (c).

5.3 Efficiency Validation

Table 5.2 shows the mean processing time in milliseconds per frame (lower the better) and the corresponding frame rate (FPS, higher the better) for each selected sequence.

TABLE 5.2:
EFFICIENCY RESULTS ON SELECTED KITTI SEQUENCES

Seq.	Frames	ORB-SLAM2		Ours	
		time	*rate*	*time*	*rate*
00	4541	91.0221	10.99	82.3385	12.14
01	1101	110.5358	9.05	98.9097	10.11
03	801	77.8148	12.85	68.8645	14.52
04	271	81.7149	12.24	67.0347	14.91
06	1101	92.6912	10.79	82.9231	12.05
10	1201	73.4403	13.62	61.5898	16.23
mean		87.8699	11.59	76.9434	13.33

The results validate that our proposed method use shorter processing time than ORB-SLAM2.

Chapter 6. Conclusion and Future Work

6.1 Conclusion

In this thesis, we firstly introduced the background of autonomous driving technologies and Vision-based Simultaneous Localization and Mapping (VSLAM). Several state-of-the-art direct and indirect VSLAM systems, specifically for stereo cameras, were reviewed with standard vision benchmarks. ORB-SLAM2 representing the indirect systems shows superiority in terms of estimation accuracy yet the high computational costs make it inferior in efficiency compared to the direct VSLAM systems.

This brought us the inspiration to integrate the advantages of both direct VSLAM and indirect VSLAM, utilizing the direct method for pose estimation between frames and indirect feature-based bundle adjustment for joint-optimization of motion and structure, to achieve competitive accuracy, with higher efficiency saving the time of feature processing.

We proposed a semi-direct VSLAM system for stereo camera. It utilizes direct image alignment for motion estimation, and ORB feature-based bundle adjustment for joint-optimization of motion and structure. The system maintains a sparse point cloud map and allows loop closing and relocalization when tracking is lost.

Finally, we validated our system over selected KITTI benchmarks in terms of accuracy and efficiency. Results prove that our system is able to achieve competitive accuracy to ORB-SLAM2 in some situations with higher efficiency.

However, our proposed system is not perfect. When initial pose estimations from direct image alignment is too bad, feature based optimization may not be able to fuse the error and thus cannot provide accurate results. In the worst cases, it may lead to negative optimization.

6.2 Future Work

There are two primary aspects of future improvements: accuracy and efficiency from both direct and indirect methods. In addition, usability and applicability should also be enhanced.

- Design and implement a more accurate direct pose estimation model.
- Improve feature-based optimizations in both accuracy and efficiency.
- Adapt the system to ROS [68] for other autonomous platforms and applications.
- Integrate AI into the system, such as machine learning techniques.
- If appliable, design and generate novel datasets for both outdoor and indoor scenarios and carry out more real-world and real-time experiments.

REFERENCES

[1] NHTSA. (2017). *Automated Vehicles for Safety* [Online]. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety.

[2] TomTom, "Traffic Index 2019," 2020.

[3] H. Taheri and Z. C. Xia, "SLAM; definition and evolution," *Engineering Applications of Artificial Intelligence,* vol. 97, p. 104032, 2021.

[4] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on robotics and automation,* vol. 17, no. 3, pp. 229-241, 2001.

[5] A. Yassin *et al.*, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 2, pp. 1327-1346, 2016.

[6] A. Alarifi *et al.*, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors,* vol. 16, no. 5, p. 707, 2016.

[7] S. Halder and A. Ghosal, "A survey on mobility-assisted localization techniques in wireless sensor networks," *Journal of Network and Computer Applications,* vol. 60, pp. 82-94, 2016.

[8] C. Li *et al.*, "A review on recent progress of portable short-range noncontact microwave radar systems," *IEEE Transactions on Microwave Theory and Techniques,* vol. 65, no. 5, pp. 1692-1706, 2017.

[9] Y. Liu, W. Liu, and X. Luo, "Survey on the Indoor Localization Technique of Wi-Fi Access Points," *International Journal of Digital Crime and Forensics (IJDCF),* vol. 10, no. 3, pp. 27-42, 2018.

[10] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 4, pp. 3607-3644, 2018.

[11] J. Xiao, Z. Zhou, Y. Yi, and L. M. Ni, "A survey on wireless indoor localization from the device perspective," *ACM Computing Surveys (CSUR),* vol. 49, no. 2, pp. 1-31, 2016.

[12] Y. Zhuang *et al.*, "A survey of positioning systems using visible LED lights," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 3, pp. 1963-1988, 2018.

[13] A. F. G. G. Ferreira, D. M. A. Fernandes, A. P. Catarino, and J. L. Monteiro, "Localization and positioning systems for emergency responders: A survey," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 4, pp. 2836-2870, 2017.

[14] B. Jang and H. Kim, "Indoor positioning technologies without offline fingerprinting map: A survey," *IEEE Communications Surveys & Tutorials,* vol. 21, no. 1, pp. 508-525, 2018.

[15] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics,* vol. 23, no. 1, pp. 34-46, 2007.

[16] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *2011 IEEE international symposium on safety, security, and rescue robotics*, 2011: IEEE, pp. 155-160.

[17] B. Steux and O. El Hamzaoui, "tinySLAM: A SLAM algorithm in less than 200 lines C-language program," in *2010 11th International Conference on Control Automation Robotics & Vision*, 2010: IEEE, pp. 1975-1979.

[18] A. Eliazar and R. Parr, "DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks," in *IJCAI*, 2003, vol. 3: Citeseer, pp. 1135-1142.

[19] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," *Aaai/iaai,* vol. 593598, 2002.

[20] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *IJCAI*, 2003, vol. 3, pp. 1151-1156.

[21] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, 2003, vol. 1: IEEE, pp. 206-211.

[22] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics,* vol. 32, no. 6, pp. 1309-1332, 2016.

[23] G. Younes, D. Asmar, and E. Shammas, "A survey on non-filter-based monocular visual SLAM systems," *arXiv preprint arXiv:1607.00470,* vol. 413, p. 414, 2016.

[24] B. Gao, H. Lang, and J. Ren, "Stereo Visual SLAM for Autonomous Vehicles: A Review," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020: IEEE, pp. 1316-1322.

[25] D. Scaramuzza and F. Fraundorfer, "Visual Odometry: Part I - The First 30 Years and Fundamentals," *IEEE robotics & automation magazine,* vol. 18, no. 4, pp. 80-92, 2011.

[26] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part II - Matching, Robustness, Optimization, and Applications," *IEEE Robotics & Automation Magazine,* vol. 19, no. 2, pp. 78-90, 2012.

[27] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014: IEEE, pp. 2609-2616.

[28] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8977-8986.

[29] M. Irani and P. Anandan, *All About Direct Methods*. 1999, pp. 267-277.

[30] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*, 1999: Springer, pp. 298-372.

[31] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research,* vol. 25, no. 12, pp. 1181-1203, 2006.

[32] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research,* vol. 31, no. 2, pp. 216-235, 2012.

[33] S. A. a. K. M. a. Others. *Ceres Solver*.

[34] G. Grisetti, R. Kümmerle, H. Strasdat, and K. Konolige, "g2o: a general framework for (hyper) graph optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, 2011, pp. 9-13.

[35] D. Schlegel, M. Colosi, and G. Grisetti, "ProSLAM: Graph SLAM from a Programmer's Perspective," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3833-3840.

[36] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics,* vol. 31, no. 5, pp. 1147-1163, 2015.

[37] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics,* vol. 33, no. 5, pp. 1255-1262, 2017.

[38] S. Sumikura, M. Shibuya, and K. Sakurada, "OpenVSLAM: A Versatile Visual SLAM Framework," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2292-2295.

[39] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," Cham, 2014: Springer International Publishing, pp. 834-849.

[40] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1935-1942.

[41] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Trans Pattern Anal Mach Intell,* vol. 40, no. 3, pp. 611-625, Mar 2018.

[42] R. Wang, M. Schwörer, and D. Cremers, "Stereo DSO: Large-Scale Direct Sparse Visual Odometry with Stereo Cameras," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3923-3931.

[43] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine,* vol. 2, no. 4, pp. 31-43, 2010.

[44] D. Schlegel and G. Grisetti, "Visual localization and loop closing using decision trees and binary features," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016: IEEE, pp. 4616-4623.

[45] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor fusion IV: control paradigms and data structures*, 1992, vol. 1611: International Society for Optics and Photonics, pp. 586-606.

[46] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, 2011, pp. 2564-2571.

[47] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics,* vol. 28, no. 5, pp. 1188-1197, 2012.

[48] H. Strasdat, J. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM," *Robotics: Science and Systems VI,* vol. 2, no. 3, p. 7, 2010.

[49] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research,* vol. 32, no. 11, pp. 1231-1237, 2013.

[50] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354-3361.

[51] M. Burri *et al.*, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research,* vol. 35, no. 10, pp. 1157-1163, 2016.

[52] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 13, no. 4, pp. 376-380, 1991.

[53] H. Kuang, X. Wang, X. Liu, X. Ma, and R. Li, "An Improved Robot's Localization and Mapping Method Based on ORB-SLAM," in *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, 2017, vol. 2, pp. 400-403.

[54] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3286-3293.

[55] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM,* vol. 18, no. 9, pp. 509-517, 1975.

[56] Z. Zhang and W. Wan, "Dovo: Mixed visual odometry based on direct method and orb feature," in *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, 2018: IEEE, pp. 344-348.

[57] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics,* vol. 33, no. 2, pp. 249-265, 2016.

[58] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60, no. 2, pp. 91-110, 2004.

[59] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, 2006: Springer, pp. 404-417.

[60] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell,* vol. 34, no. 7, pp. 1281-1298, 2011.

[61] X. Yang and K.-T. Cheng, "LDB: An ultra-fast feature for scalable augmented reality on mobile devices," in *2012 IEEE international symposium on mixed and augmented reality (ISMAR)*, 2012: IEEE, pp. 49-57.

[62] L. M. Paz, P. Piniés, J. D. Tardós, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *IEEE transactions on robotics,* vol. 24, no. 5, pp. 946-957 %@ 1552-3098, 2008.

[63] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An invitation to 3-d vision: from images to geometric models*. Springer Science & Business Media, 2012.

[64] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual SLAM," in *2011 International Conference on Computer Vision*, 2011, pp. 2352-2359.

[65] R. Mur-Artal and J. D. Tardós, "Fast relocalisation and loop closing in keyframe-based SLAM," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014: IEEE, pp. 846-853.

[66] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision,* vol. 81, no. 2, p. 155, 2009.

[67] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013: IEEE, pp. 209-218.

[68] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, 2009, vol. 3, no. 3.2: Kobe, Japan, p. 5.