FAST AND LIGHTWEIGHT LOOP CLOSURE DETECTION IN LIDAR-BASED SIMULTANEOUS LOCALIZATION AND MAPPING

HAODONG XIANG

PhD

The Hong Kong Polytechnic University

2022

The Hong Kong Polytechnic University
Department of Land Surveying and Geo-Informatics

Fast and Lightweight Loop Closure Detection in LiDAR-based Simultaneous Localization and Mapping

Haodong Xiang

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

CERTIFICATE OF ORIGINALITY

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

Signature: _____

Name of Student: Haodong Xiang

Abstract

Over the last decade, complex urban environments raised higher demands on geographic information data collection. Traditional data collection methods gradually fail to meet the growing efficiency, completeness, flexibility, and safety requirements. The advent of mobile mapping systems (MMS) filled these gaps but it has also brought new challenges to data processing. The processing of mobile measurement data requires automated, accurate, and efficient algorithms, which have been the hottest research topics in many relevant fields.

In the thesis, loop closure detection (LCD), as one of the core problems of Simultaneous localization and mapping (SLAM) will be studied in depth. Focusing on LCD in indoor environments, a fast and compact algorithm is proposed utilizing comprehensive descriptors extraction and machine learning. Besides, a novel double-deck loop candidate verification strategy is proposed to validate loop candidates and reject false positives.

As for outdoor large-scale environments, point clouds do not exhibit significant structural and regular geometric characteristics. Thus, the deep learning model is utilized to mine advanced and high-dimensional features. A very deep and lightweight neural network DeLightLCD is proposed to enable efficient LCD. The framework contains two key modules: a feature extraction module and a feature difference module. Depth-wise separable convolution (DSC) and batch normalization (BN) are utilized to ensure that the network is lightweight and trainable.

In practical use, the generalization and flexibility performance of LCD algorithms are affected by sensor changes and indoor-outdoor environmental changes. Thus, DeLightLCD++ is proposed to address these problems. The improvement of DeLightLCD++ is threefold. (1) A novel data presentation method encoding measurement distance and azimuth angle information is used to reduce the effects of sensor changes. (2) The architecture of the network is also adjusted to ensure that the algorithm is rotation invariant. (3) A loop candidate fast search method is used to suppress the computation cost and time cost increase due to ultra-long measurement distance.

After loop closures are detected, the results will be utilized for the pose optimization to eliminate accumulative errors in LiDAR odometry (LO). An enhanced graph optimization strategy based on LCD results is utilized in this thesis. Besides, three types of loops in graphs, detected loop closures, pseudo loop closures, and enhanced loop closures are introduced. Then, experiments are conducted to study factors affecting trajectory optimization performance. Finally, some guidance is given on fieldwork and data processing of the mobile mapping backpack system.

The proposed methods are evaluated on open-source datasets and in-house datasets. The in-house datasets are captured by a self-designed mobile mapping backpack system. The backpack is equipped with two multi-line laser scanners. Results show that the LCD algorithms are superior to state-of-the-art algorithms in precision, time efficiency, generalization performance, and flexibility. The optimization method could effectively improve the LiDAR odometry results and enable a consistent map result.

In sum, this thesis focuses on LCD and optimization for LiDAR-SLAM. The three LCD algorithms presented in the thesis aim to solve LCD problems in indoor and outdoor large-scale scenes. The experiments exhibit the effectiveness and superior performance of the proposed algorithms. The work presented can be implemented in LiDAR-SLAM for surveying and mapping. Furthermore, it could be used for autonomous driving, high-definition maps, and urban 3D modeling.

Publications Arising from the Thesis

1. Haodong Xiang, Wenzhong Shi, Wenzheng Fan, Pengxin Chen, Sheng Bao, and Mingyan Nie, "FastLCD: A fast and compact loop closure detection approach using 3D point cloud for indoor mobile mapping", in *International Journal of Applied Earth Observation and Geoinformation (JAG)*, pp. 102430, 2021.

2. Haodong Xiang, Xiaosheng Zhu, Wenzhong Shi, Pengxin Chen, Sheng Bao, and Yue Yu, "DeLightLCD: A Deep and Lightweight Network for Loop Closure Detection in LiDAR SLAM", in *IEEE Sensors Journal (SenJ)*, 2022, doi: 10.1109/JSEN.2022.3206506.

3. Wenzheng Fan, Wenzhong Shi, Haodong Xiang, and Ke Ding, "A Novel Method for Plane Extraction from Low-Resolution Inhomogeneous Point Clouds and its Application to a Customized Low-Cost Mobile Mapping System", in *Remote Sensing*, pp. 2789, 2019.

4. Pengxin Chen, Wenzhong Shi, Wenzheng Fan, Haodong Xiang, and Sheng Bao, "RectMatch: A novel scan matching method using the rectangle-flattening representation for mobile LiDAR systems", in *ISPRS Journal of Photogrammetry and Remote Sensing (ISPRS)*, pp. 191-208, 2021.

5. Sheng Bao, Wenzhong Shi, Pengxin Chen, Haodong Xiang, and Yue Yu, "A Sys-

tematic Mapping Framework for Backpack Mobile Mapping System in Common Monotonous Environments", in *Measurement*, pp. 111243, 2022.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Wenzhong, John Shi, a respectable, responsible, and resourceful scholar who has given me numerous important suggestions and invaluable guidance during the period of my thesis writing.

All our team members, especially Dr. Wenzheng Fan, Dr. Pengfei Chen, Dr. Zhicheng Shi, Dr. Anshu Zhang, Dr. Min Zhang, Dr. Zhewei Liu, Dr. Zhipeng Luo, Dr. Yangjie Sun, Dr. Yue Yu, Mr. Yepeng Yao, Mr. Pengxin Chen, Mr. Sheng Bao, Mr. Muyang Wang, Mr. Xiaosheng Zhu, Mr. Mingyan Nie, Mr. Daping Yang, Ms. Xiaolin Zhou, Mr. Shanxiong Chen, Mr. Zhao Zhan, Mr. Chao Zhang, Mr. Fanxin Zeng, and Mr. Chengzhuo Tong have also provided me a profusion of help and suggestions for my research and thesis writing.

I would like to extend my sincere gratitude to all the teachers who have taught me or given me advice on my research or daily life. They have helped me a lot when I was confronted with confusion or difficulties. I would also thank the non-academic staff for their support and assistance.

I am also deeply grateful to my lovely friends for their cordial help and academic inspiration. They spare no efforts to give me respect and support when I was writing my thesis, especially Dr. Yi Jian, Dr. Yiran Wang, Dr. Junbiao Su, Mr. Keru Lu, Mr. Jiaming Zhu, and Mr. Hao Fu. I would also thank Chen Jin, Jianan He, Zhaowen Wu, Sida, Derek, Brian, Joe, and Mutu. Thank you very much for your emotional

support and encouragement. There are so many people that have supported me. I could not list all your names here, but thank you all for every moment we shared and the support you gave me. I am deeply moved by the pure friendship. It will be a beautiful and cherishing memory in my lifetime.

I would like to express my thanks to my beloved parents for their love and support without expectation of return. For so many years, they have always been supporting me and respecting me. Their love and care are the greatest fortunes of my life.

Finally, I would like to thank myself. All the struggle and pain in these years finally paid off. Thank you for not giving up.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Background

The Government published the Smart City Blueprint for Hong Kong (Blueprint 2.0) in 2017, setting out six smart areas, including "Smart Mobility", "Smart Living", "Smart Environment", "Smart People", "Smart Government" and "Smart Economy". Behind them, many fields both in academia and industry should develop efficient and practical technologies to support the construction and improvement of the Smart City scheme in Hong Kong. The vital and core technologies of the Smart City scheme include urban planning, big data, the internet of things, artificial intelligence, etc. However, the most fundamental technology is surveying and mapping cities efficiently and precisely. Academic and industrial practitioners have made great hardware development and algorithm research efforts to make urban 3D data acquisition more efficient, complete, and precise. Then, mobile mapping was created to overcome the shortcomings of the traditional terrestrial data collection system with low efficiency and high labor cost.

The last decade saw great progress in data capture sensors, which drove the development of various data capture systems based on multiple platforms, especially mobile

mapping systems. Mobile mapping system has become one of the most important and widely used data capture methods for urban 3D geographic information data collection. The data capture sensors include cameras, Light Detection and Ranging (LiDAR), Radar, the Inertial Navigation System (INS), Global Navigation Satellite System (GNSS), etc. The sensors capture texture and color information, ranging measurement information, positioning, and orientation information. The multiple geometry and texture information enables 3D model reconstruction of urban environments with precise geometry information and photo-realistic information. Besides, various platforms from ground-based to aerial-based allow the data capture of complete urban scenes. The common data capture platforms include vehicles, Unmanned Aerial Vehicles (UAV), backpacks, trolleys, vessels, and robots. The aerial-based platforms, like Unmanned Aerial Vehicle (UAV), could capture data from rooftops or high places, compensating for the blind spots of the ground-based data collection system.

Currently, the two most important data capture sensors are LiDAR and cameras. In recent years, the cameras' frame rate and pixel resolution have been greatly improved, which lay a solid foundation for algorithms based on visual data. Vision sensors have become the most widely used type of sensor due to their low cost. However, there are many inherent defects of visual sensors. (1) Most cameras, as passive sensors, are more sensitive to illumination condition changes and seasonal changes; (2) Observing depth is difficult for cameras. Although some types of cameras could obtain ranging distance information, the distance of observing depth is quite limited; (3) In some regions with weak texture information, it is difficult to extract valid features from the visual data, which brings challenges to subsequent algorithm development. Although these defects are greatly compensated with the emergence of new algorithms, the application of pure cameras is still insufficient to meet the demand. LiDAR sensors, also generally called laser scanners, are measuring devices that measure the distance from a sensor to a target by emitting a laser beam to illuminate the target. LiDARs

have many advantages, like measurement of distance information directly, high range measurement accuracy, a wide range of detection, being less affected by illumination condition changes and seasonal changes, and resistance to electromagnetic interference. The sensors usually include 2D laser scanners and 3D laser scanners. In recent years, as LiDAR sensors have been upgraded from 2D Lidar to 3D Lidar, and the price has declined, LiDAR sensors are widely used in mobile mapping systems and play an increasingly significant role.

After data collection, some technologies will be utilized to process the data. Simultaneous Localization and Mapping (SLAM) plays a significant role in mobile mapping data processing. SLAM is a computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. Before SLAM was proposed, localization and mapping were always considered to be two separate problems(Smith, Self, & Cheeseman, 1988). Since SLAM was first proposed by Leonard and Durrant-Whyte (J. J. Leonard & Durrant-Whyte, 1991), which has always been considered by researchers to be a major problem in the field of mobile robots. The solution of SLAM will make it possible for mobile agents to move in an environment without prior knowledge, which is of great significance. SLAM is especially significant for indoor or other GNSS-denied scenes mobile mapping.

When graph-based optimization is adopted in SLAM computation, the SLAM process can be roughly divided into two steps: front-end odometry and back-end optimization. Loop closure detection (LCD) plays a core role in back-end optimization. LCD is to detect whether the agent revisits the places. LCD is a problem of data association in SLAM. It aligns two data that are in the same place but discontinuous in time. LCD provides control information that could be treated as redundant observations for back-end optimization.

Currently, LCD and SLAM have gained outstanding progress in the past decades. Especially in recent years, with the development of LiDAR sensors and computer

science, the research on LCD and SLAM has been a hot research direction in the field of computer science, surveying and mapping, robotics, electronics, etc. Tradition approaches are based on handcrafted features, such as key points, feature lines, and planar features. When deep learning is used in this field, semantic features and high-dimensional features are learned to address the problems. From the perspective of the application, indoor small-scale scenes and large-scale outdoor scenes require different feature types and detection strategies to guarantee the results' precision and time efficiency. Specifically, the human-made indoor scenes always contain rich geometry features, like planar features and line features, while the outdoor scenes may not have sufficient and regular geometry features, especially in natural environments. Thus, the challenges for LCD in indoor scenes are feature integration and fusion, while the problems for LCD in outdoor environments are feature extraction. Besides, time efficiency and computation complexity are also key problems for LCD and SLAM. The method should be fast even conducted in real-time to save time cost, and lightweight to save computation resources.

1.2 Research Scope and Problem Statement

In the thesis, loop closure detection in SLAM using 3D point cloud data will be researched in depth. Loop closure detection means identifying whether the place has been revisited. The problem is simplified to check whether a pair of data is similar enough in many algorithms. If two LiDAR scans are highly similar, they are recognized to be collected in the same scenes. Then, the two scans construct a loop. LCD is a problem of data association [56]. Many researchers contribute to finding solutions from distinct views, like data retrieval [74] or pattern recognition [163]. In the field of place recognition and computer vision, deep learning is popular and widely used [148, 100, 26].

The problem statement of LCD could be represented as follows. A LiDAR scan query

denoted as $\{(P_1, P_2, \ldots, P_I)|P \in \mathbb{R}^{N\times 3}\}$, where P_I means the point cloud scan in the query. A pair of LiDAR scans P_i and P_j will be input into an LCD algorithm module for detection, while the output is a binary classification to identify whether they are a loop closure.

1.3 Current Research Problems

- Because of the huge point volume in every point cloud scan, the dimensional reduction is needed to reduce the computation cost and time cost. Feature extraction is a conventional method for dimensional reduction, but how to define the features which can describe the environment comprehensively and precisely, and how to extract features effectively and efficiently from the inhomogeneous point cloud data are urgent problems to be solved.

- For indoor scenes, current research always utilize the rich geometry features, while other information also may play a significant role in this task. The challenges are how to fuse and integrate them for LCD in a highly efficient time level.

- For outdoor scenes, the computation cost and data amount will increase sharply with the measuring distance and measuring time increasing, Thus, fast and lightweight algorithms should be researched to conduct highly efficient or even real-time LCD. Besides, in large-scale environments, sparse point cloud density, long measurement distances, indistinctive geometry features, fast-moving measurement platforms, and a large number of moving objects all bring new challenges to LCD algorithms. Thus, robust LCD approaches need to be researched.

- Point cloud data is 3-dimensional and unordered. Using point clouds as input directly for the deep learning model will suffer from permutation invariance

problems. Some researchers transform 3D point clouds into 2D image space. However, the transformation methods are worth studying to ensure low data loss and efficient computation.

- Currently, many algorithms could only process specific point clouds collected by one kind or one type of LiDAR sensor. Thus, a general and robust LCD framework is worth studying regardless of sensor types and environment scale changes.

1.4 Research Objectives

- Objective 1: To propose a fast LCD algorithm for indoor environments. Comprehensive and multi-modality features of point clouds are extracted and analyzed to enable a fast and compact LCD method.

- Objective 2: To propose a lightweight LCD algorithm for large-scale environments. The deep learning technique will be used to extract advanced and high-dimensional features from point cloud data.

- Objective 3: To propose a flexible deep-learning-based LCD algorithm regardless of sensor changes and environmental changes. Besides, a fast loop candidates search strategy is also needed to ensure time efficiency.

- Objective 4: To propose an enhanced graph optimization method based on LCD in SLAM which is used to suppress the accumulative error in LiDAR odometry. Factors of loops affecting optimization performance are worthy research topics to give some guidance on fieldwork measurement and data processing.

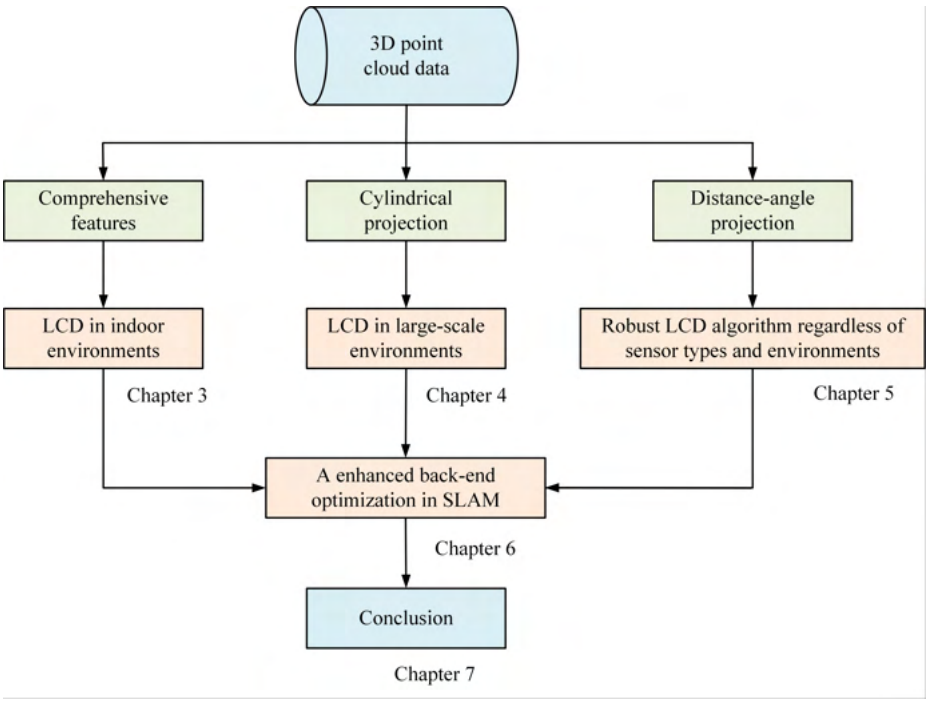


Figure 1.1: Outline of the thesis

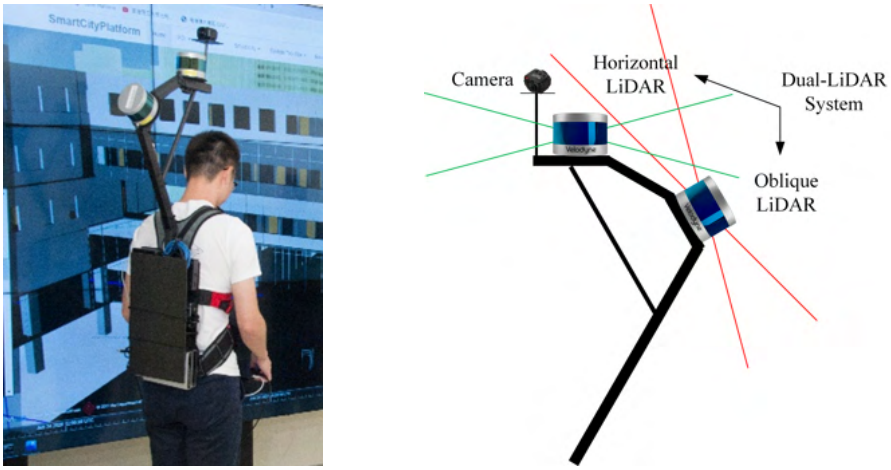
1.5 Thesis Outline

The thesis is organized as follows, as shown in Fig. 1.1. In *Chapter* 3, comprehensive descriptors are proposed encoding discriminative multi-modality features to describe each scan of point clouds. A machine learning model is used to learn from the descriptors to perform fast and compact LCD (objective 1). A very deep and lightweight neural network DeLightLCD is proposed in *Chapter* 4 to enable real-time loop closure detection in large-scale environments (objective 2). An improved method DeLightLCD++ is proposed in *Chapter* 5 based on DeLightLCD which is more flexible to diverse point cloud input and robust to the environment changes, whether in small-scale indoor environments or large-scale outdoor scenes. Besides, an efficient loop candidates search strategy is designed to suppress the time cost (objective 3). In *Chapter* 6, an enhanced graph optimization based on loop closure detection is used to reduce the cumulative error of LiDAR odometry in SLAM. In addition, a study of

factors of loops affecting optimization performance is also conducted (objective 4).

1.6 Research Basis

To verify the proposed framework of loop closure detection and back-end optimization, a lightweight 3D mobile mapping backpack named *SpaceScanX* is designed for data collection, as shown in Fig. 1.2. The mobile mapping backpack is equipped with LiDAR, an omnidirectional camera, an IMU, and GNSS receivers.



(a) (b)

Figure 1.2: The self-designed mobile mapping backpack system

The 3D mobile mapping backpack has the following advantages:

- Lightweight, only about 9 kg;
- Seamless indoor-outdoor highly efficient data capturing;
- Flexible data capture in some challenging environments, like indoor, underground, city canyon, mine caves, or other GNSS-denied environments;
- Abundant data including 3D point clouds, panoramic images, IMU data, and GNSS data;

- Accurate system calibration and time synchronization;
- Facilitating with efficient SLAM solutions.

Chapter 2

Literature Review

2.1 3D Mapping Technologies: A Review

The last decades have witnessed the substantial development of data acquisition systems: satellites, drones, Unmanned Aerial Vehicles(UAV), vehicles, vessels, trolleys, backpacks, and robots. These autonomous and intelligent data acquisition systems greatly facilitate the surveying and mapping industry which are also expanded and implemented in many fields of work for 3D modeling, urban planning, building, electricity industry, public infrastructure management, autonomous driving, etc. [134]. Especially, driven by the development of the data acquisition sensors, problems of urban 3D modeling, like building information models (BIM) have become hot topics in academia and industry. Urban 3D modeling aims at establishing a 2.5D or 3D digital representation of urban areas and the objects, such as roads, buildings, vegetation, indoor environments, and other man-made structures. There are three major techniques utilized in this field: (1) conventional geodetic mapping, (2) 2D image photogrammetry, and (3) 3D measurements, such as laser scanning. Thus, the development of the urban 3D modeling depends on the development of the data acquisition sensor [151].

The tremendous growth of these versatile mobile mapping systems is expected to be promoted by (1) the abundance of new data acquisition sensors and mobile mapping platforms, (2) the increasingly low price, small volume, and lightweight of sensors, (3) the development in many relevant industries of communication, computer, machinery, electronic, and geographic information, (4) the constant advancement of the fundamental technologies from the robotics and autonomous positioning and navigation communities [73].

Compared with conventional data acquisition methods, data collection by mobile mapping methods determines a higher efficiency. Thus, the mobile mapping approach greatly increases the frequency of data updates. Besides, for the data capture in dangerous environments, the unmanned mobile mapping method improves safety and saves manpower. However, mobile mapping also brings more complex problems of data processing.

In this chapter, data capture sensors will be summarized and compared first. Some common mobile mapping systems including UAV-based mobile mapping, vehicle-based mobile mapping, and backpack-based mobile mapping systems will be briefly reviewed. Then, as the core technology of mobile mapping, simultaneous localization and mapping (SLAM) will be introduced and reviewed in detail. Especially, loop closure detection which is one of the fundamental parts of SLAM will also be researched.

2.2 Data Capture Sensors

With the development of sensor technology in the last decade, the types of sensors have become more abundant, and their accuracy, reliability, and availability have greatly improved. As for the sensors installed on mobile mapping platforms, they can be generally grouped into three categories: (1) laser scanners for capturing point

cloud data of the surrounding environments directly, (2) visual sensors for obtaining visual data with rich texture information, (3) positioning and navigation sensors for providing positioning information or geo-reference information. Generally, every single type of sensor has its unique strengths and inherent weaknesses. A reliable and mature mobile mapping system usually needs to fuse the data acquired by different types of sensors. In this section, some common sensors for mobile mapping systems will be summarized.

2.2.1 Laser Scanners

With increasing applications relying on geospatial information, laser scanning has become one of the most popular surveying and mapping methods. Laser scanning systems also called light detection and ranging systems (LiDAR), use light pulses to collect information from surroundings. The systems capture a huge amount of 3D coordinates (also known as points), which are combined to generate point clouds.

Laser scanners generally could be divided into airborne laser scanning (ALS), terrestrial laser scanning (TLS), and mobile laser scanning (MLS). Due to our focus on ground-based mobile mapping platforms, TLS and MLS will be introduced in detail. Both the two types use similar measurement principle of light pulses. However, there are some fundamental differences between these two laser scanning systems.

2.2.1.1 Terrestrial Laser Scanning

TLS uses ground-based remote sensing systems, which are now being widely used for topographic mapping applications. TLS sensors are usually mounted on static tripods. The measurement mode of TLS is station-by-station. After the scanning in an area finishing, the system needs to be moved to another position. TLS is also a subtype of MLS. TLS could also be installed on mobile land-based platforms. Then, it could be used for mobile scanning in large areas.

According to the measuring techniques adopted for data capture, terrestrial laser scanners are classified into those that employ pulses ranging or time-of-flight (TOF) measuring principle and those that utilize the phase measuring technique [50]. Besides, in [126], terrestrial laser scanners are also divided into panoramic-type 3D scanners, hybrid laser scanners, and camera-type 3D laser scanners. Based on the platforms where laser scanners are installed, terrestrial laser scanners are generally grouped into static terrestrial laser scanners and dynamic laser scanners. In the study of [130], according to the range or distance, the static terrestrial laser scanners also can be divided into three types:

- Short-range laser scanners are limited up to 150 m, some are also limited to 30-60 m. They usually utilize the phase measuring principle for distance measurement. The limitation of the short measurement range significantly restricts the application of these laser scanners. Therefore, short-range laser scanners are commonly used in indoor environments within buildings and urban area outdoor environments with high buildings. However, the defects in the range of these sensors are compensated by the high accuracy that they achieve in distance measurement, generally at a few millimeters level.

- Medium-range laser scanners typically measure 150-450 m. Due to the measurement range being farther than the short-range laser scanner, accordingly, its measurement accuracy has been reduced. Medium-range laser scanners almost utilize the TOF technique for distance measurement.

- Long-range laser scanners can cover very long distances, up to several kilometers. Such a long measurement distance also inevitably brings a decrease in measurement accuracy, but the current level of accuracy is still acceptable. Long-range laser scanners are usually used in large-scale environments, like mine surveying and mapping. This type of laser scanner also adopts the pulse ranging technique which allows much longer distances.

(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

(m) (n) (o) (p)

Figure 2.1: Some common TLS sensors. (a) Zoller + Fröhlich (Z+F) IMAGER 5006EX 3D laser scanner, (b) Z+F IMAGER 5016 3D laser scanner, (c) Faro Focus^M series laser scanner, (d) Faro Focus^S series laser scanner, (e) Surphaser IR_100HS laser scanner, (f) Surphaser 10_HS laser scanner, (g) Leica BLK360 Imaging laser scanner, (h) Leica RTC360 3D laser scanner, (i) Leica ScanStation P50, (j) TOPCON GLS-2200 series 3D laser scanner, (k) Teledyne Optech Polaris laser scanning system, (l) Teledyne Optech TLS-M3 laser scanning system, (m) Trimble X7 3D laser scanning system, (n) Trimble TX8 3D laser scanner, (o) RIEGL VZ400i laser scanner, and RIEGL VZ6000 laser scanner.

Table 2.1: Maximum range and accuracy comparison of some common TLSs

Products	Maximum range	Range accuracy
Z+F IMAGER 5006EX	79 m	0.7 mm
Z+F IMAGER 5016	360 m	0.25 mm
Faro Focus^{M}70	70 m	±3 mm
Faro Focus^{S}70	70 m	±1 mm
Faro Focus^{S}150	150 m	±1 mm
Faro Focus^{S}350	350 m	±1 mm
Surphaser IR_100HS	90 m	0.16 mm @10 m
Surphaser 10_HS	180 m	0.25 mm @15 m
Leica BLK360	60 m	4 mm @10 m / 7 mm @20 m
Leica RTC360	130 m	1.0 mm + 10 ppm
Leica ScanStation P50	¿1 km	0.4mm @10 m / 0.5 mm @50 m
TOPCON GLS-2200-short	130 m	3.1mm @1-90 m
TOPCON GLS-2200-middle	350 m	3.1mm @1-110 m
TOPCON GLS-2200-long	500 m	3.1mm @1-150 m
Teledyne Optech POLARIS	¿2 km / 750 m / 250 m	5 mm @100 m
Teledyne Optech TLS-M3	¿2 km / 750 m / 250 m	5 mm @100 m
Trimble X7	80 m	¡2.5 mm @30 m
Trimble TX8	340 m	¡2 mm @2-120 m
RIEGL VZ-400i	800 m	5 mm
RIEGL VZ-2000i	2500 m	5 mm
RIEGL VZ-6000	>6 km	5 mm

As shown in Tab. 2.1 and Fig. 2.1, some common static TLS sensors are summarized. Sensors with a short measurement range are advanced in range accuracy. Among these sensors, Leica, Teledyne Optech, and Riegl produce high-precision long-range laser scanners, while Faro is more common for its short-range and medium-range laser scanners.

As for mobile terrestrial laser scanning, a terrestrial laser scanner is mounted on a mobile platform, as shown in Fig. 2.2. Common mobile platforms include vehicles,

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

Figure 2.2: Mobile TLSs. (a) RIEGL VMR with RIEGL VZ-400i laser scanner, (b) Faro TLS laser scanner mounted on Boston Dynamics Spot Robot platform, (c) Leica BLK360 installed on a UAV platform, and (d) Leica BLK360 installed on a handheld platform.

trolleys, UAVs, or robots. Mobile terrestrial laser scanning measurement not only maintains the high accuracy on the measurement range of fixed station laser scanners but also improves efficiency and broadens the scope of laser scanning work. In some areas that are dangerous or difficult for humans to access, mobile TLS is significant and effective to be used. Thus, in recent years, many manufacturers have dedicated a lot of effort to producing easy-to-use, reliable, cost-effective mobile TLS systems.

2.2.1.2 Mobile Laser Scanning

Mobile Laser Scanning (MLS) uses laser scanning sensors mounted on mobile platforms. These MLS sensors can be mounted on land-based vehicles such as trolleys, cars, airborne vehicles such as UAVs and helicopters, boats, backpacks, robots, or handheld platforms.

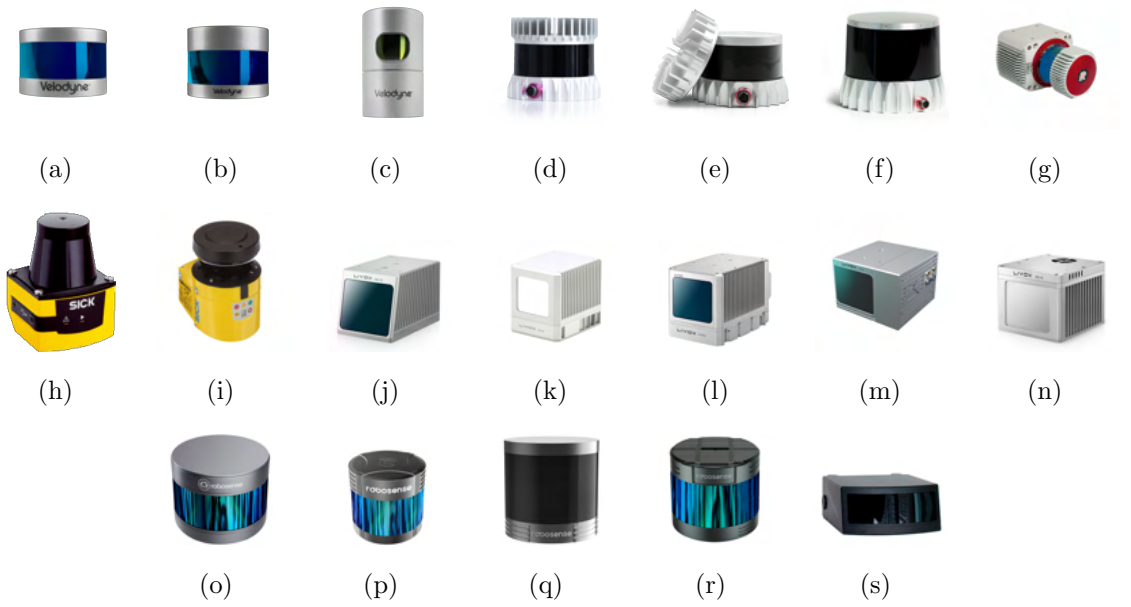


Figure 2.3: Some common MLSs. (a) Velodyne Puck LiDAR, (b) Velodyne Ultra Puck LiDAR, (c) Velodyne HDL-32E LiDAR, (d) Ouster OS0 LiDAR sensor, (e) Ouster OS1 LiDAR sensor, (f) Ouster OS2 LiDAR sensor, (g) RIEGL VUX-1HA laser scanner, (h) SICK TIM781S-2174104 laser scanner, (i) SICK S30B-3011BA laser scanner, (j) Livox Mid-70 laser scanner, (k) Livox Mid-40 laser scanner, (l) Livox Horizon laser scanner, (m) Livox Avia laser scanner, (n) Livox Tele-15 laser scanner, (o) Robosense RS-LiDAR-16 LiDAR, (p) Robosense RS-LiDAR-32 LiDAR, (q) Robosense RS-Helios LiDAR, (r) Robosense Ruby LiDAR, and (s) Robosense RS-LiDAR-M1 LiDAR.

Table 2.2: Maximum range and accuracy comparison of some common MLSs

Products	Channels	Maximum range	Range accuracy	Weight
Velodyne Puck	16	100 m	±3 cm	830 g
Velodyne Ultra Puck	32	200 m	±3 cm	925 g
Velodyne HDL-32E	32	100 m	±2 cm	1000 g
Velodyne Alpha Prime	128	245 m	±3 cm	3500 g
Ouster OS0	32/64/128	50 m	$\pm 1.5 - 5$ cm	447 g
Ouster OS1	32/64/128	120 m	$\pm 0.7 - 5$ cm	447 g
Ouster OS3	32/64/128	240 m	$\pm 2.5 - 8$ cm	1100 g
RIEGL VUX-1HA^{2}2	1	475 m	±5 mm	3500 g
SICK TIM781S-2174104	1	25 m	±60 mm	250 g
SICK S30B-3011BA	1	30 m	\	1200 g
Livox Mid-70	\	¿260 m	±2 cm	580 g
Livox Mid-40	\	¿260 m	±2 cm	760 g
Livox Horizon	\	¿260 m	±2 cm	1100 g
Livox Avia	\	450 m	±2 cm	498 g
Livox Tele-15	\	1000 m	±2 cm	1500 g
Robosense RS-LiDAR-16	16	150 m	±2 cm	870 g
Robosense RS-LiDAR-32	32	200 m	±3 cm	1130 g
Robosense RS-Helios	32	150 m	±3 cm	1000 g
Robosense Ruby	128	250 m	±3 cm	3750 g
Robosense RS-LiDAR-M1	128	200 m	±5 cm	730 g

Compared with MLS methods, traditional measurements are time-consuming, subject to traffic, pedestrian, and road conditions, and even require isolation of the measurement area from traffic. Besides, traditional measurements require many on-site workers. Whereas, MLS is outstanding of its high time efficiency and could save project budgets for on-site operation. Due to the high time efficiency, MLS can quickly analyze environmental conditions for emergency response, and can also be used for omnidirectional data collection for street view, such as Google Maps.

As shown in Fig. 2.3, some important MLS systems are summarized. Their channels, maximum range, range accuracy, and weights are compared in Tab. 2.2. All the mobile laser scanners can be divided into two categories, non-repetitive scanning pattern, and repetitive scanning pattern, according to the scanning theory. Livox series laser scanners are all non-repetitive laser scanners. The point cloud density will increase with resting time. The repetitive laser scanners are more common, usually generating multiple scan lines, like 16, 32, 64, or 128 channels. Mobile laser scanners are more lightweight than terrestrial laser scanners, while the range accuracy is worse than TLS sensors.

2.2.1.3 Comparison Between TLS and MLS

With laser scanning as one of the most important and commonly used methods of environmental data collection, surveying teams must decide whether to use TLS or MLS systems. This decision affects the cost of the project, schedule, measurement time, quality, and accuracy of the data. The defects and advantages of these two measurement methods are summarized as follows:

- MLS are much more efficient than TLS. Thus, for large-scale environments measurement, MLS can complete data collection tasks very quickly. However, data captured by MLS are less accurate than those from TLS.
- MLS is more suitable for measuring in areas with limited accessibility. MLS sys-

tems can measure the areas that are unsafe or inaccessible for workers. Sensors mounted on the UAV platforms could capture data from restricted locations.

- Generally, static TLS systems can collect more accurate, detailed, and high-density point clouds. The static TLS sensor remains completely still during the scanning of a station, which results in the low risk of data outliers. The static TLS sensor can also be moved to another station for measuring the environment from different angles and locations. Thus, the static TLS system could capture more accurate and detailed information from environments.

- It may take longer to process data of static TLS than that of MLS. Denser point clouds and richer details bring larger file sizes. More time will be needed for data processing if the files are larger. While MLS data processing, especially point cloud registration, is also time-consuming, static TLS data processing will take more time.

- Data storage should also be considered for static TLS system. Some supporting software could provide cloud storage functions for large datasets. If not, users need to equip a large hard disk for data storage.

2.2.2 Visual Sensors

Laser scanners provide high range accuracy, while visual data is rich in texture. The two types of data have their irreplaceable advantages. Compared with point clouds, visual data are sensitive to illumination and season changes which may lead to critical degradation under certain circumstances. In previous decades, driven by the reduced prices and widespread applications of laser scanners, increasing work began to focus on LiDAR-based SLAM and LCD. However, visual SLAM is also a hot spot and a difficult area in both academia and industry. As shown in Tab. 2.3, the characteristics of LiDAR sensors and visual sensors are listed. Due to their respective unique

advantages, sensor fusion is a reasonable and feasible strategy to reinforce strengths and suppress weaknesses. Thus, many measurement systems integrate laser scanners, visual sensors together to achieve robust performance and versatile information.

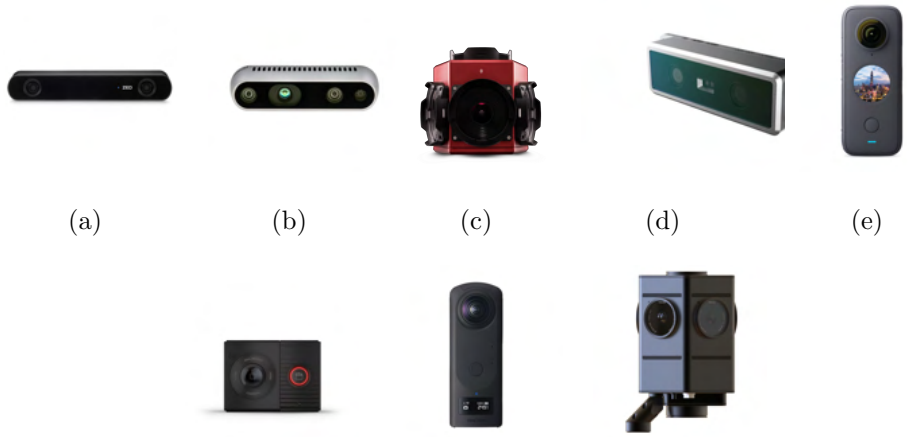
Table 2.3: Comparison between LiDAR sensors and Visual sensors

	LiDAR Sensors	Visual Sensors
Advantages	1. high range accuracy 2. insensitive to illumination changes 3. excellent distance tracking performance	1. rich texture information 2. low cost 3. high resolution
Disadvantages	1. relatively expensive 2. low anti-interference capability from weather	1. sensitive to illumination changes 2. fail in feature extraction in some areas

Table 2.4: Some common visual sensors and resolution comparison

Products	Number of lenses	Panoramic	Maximum Resolution	Weights
ZED 2 stereo camera	2	No	4416×1242	124 g
Intel Depth Camera D435	2	No	1920×1080	\
Teledyne FLIR Ladybug 5+	6	Yes	2048×2464	3 kg
MYNT EYE P Depth camera	1	No	2560×720	184 g
Insta 360 One X2	2	Yes	2560×1440	149 g
Garmin Dash Cam Tandem	2	Yes	2560×1440	65.4 g
Ricoh Theta Z1 camera	2	Yes	6720×3360	182 g
Teche TE720 pro	7	Yes	4608×3456 for each	1.4 kg

Visual sensors are widely used in data acquisition, which results in their rapid development. There is a wide variety of cameras, including monocular cameras, stereo cameras, depth cameras (RGB-D cameras), panoramic cameras, event cameras, etc. Versatile types of visual sensors are briefly shown in Fig. 2.4. Visual sensors also can generate point cloud data. Depth cameras can measure depth information directly, while stereo cameras can calculate range information. Besides, structure-from-motion



(a) (b) (c) (d) (e)

(f) (g) (h)

Figure 2.4: Some common visual sensors. (a) ZED 2 stereo camera, (b) Intel Depth Camera D435, (c) Teledyne FLIR Ladybug 5+ camera, (d) MYNT EYE P Depth camera, (e) Insta 360 One X2 camera, (f) Garmin Dash Cam Tandem Dual-lens camera, (g) Ricoh Theta Z1 camera, and (h) Teche TE720 pro panoramic camera.

(SfM) or visual SLAM also could generate point cloud data. However, the point clouds generated by these visual techniques are still different from the point clouds captured by laser scanners. Generally, the measurement range is much shorter than that of laser scanners. The focal length of cameras also affects the performance of point clouds generated.

2.2.3 Positioning and Navigation Sensors

Currently, there are many positioning technologies used in mobile mapping applications. Some prevalent devices and technologies include Global Navigation Satellite Systems (GNSS), Inertial Measurement Unit (IMU), WiFi, Bluetooth, Ultra Wide Band (UWB), ZigBee, and Radio Frequency Identification (RFID), which are the most commonly used positioning and orientation technologies. Besides, SLAM has been adopted as one of the most popular software solutions to generate trajectories

and orientations.

GNSS refers to all satellite navigation systems in general, including global and regional navigation systems and their augmentation systems, such as Global Positioning System (GPS), Globalnaya Navigatsionnaya Sputnikovaya Sistema (GLONASS), BeiDou Navigation Satellite System (BDS), and Galileo; augmentation systems, such as Wide Area Augmentation System (WAAS), European Geostationary Navigation Overlay Service (EGNOS), and MTSAT Satellite Augmentation System (MSAS).

GNSS receivers are devices that can provide absolute position in all-weather, all-day, all-range in the global coordinate system, but they also have the disadvantages of poor positioning accuracy and stability. They are easily affected by the environment and weather. It is responsible for receiving the satellite signals to solve the position information. When working in an urban area with dense buildings, the GNSS signal is not reliable enough, i.e. multi-path noise and occlusion by surrounding tall buildings. As a data processing method based on high-frequency measurements and second-order integration, GNSS suffers from drift accumulation with distance increasing. However, in indoor environments, GNSS receivers cannot receive any signal for position information calculation.

IMU devices consisting of accelerators and gyroscopes are used to calculate motion and attitude changes. Regardless of the operating conditions and environments, IMUs

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

Figure 2.5: Some common positioning and navigation sensors. (a) NovAtel GPS-704-X antennas, (b) Hemisphere HA32 UAV GNSS Antenna, (c) Trimble R10 Integrated GNSS System, and (d) Emlid Reach M2 RTK GNSS modules.

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

Figure 2.6: Some MEMs IMU systems. (a) EnOcean STM300 IMU, (b) HoneyWell HGuide i300 MEMs IMU, (c) LORD 3DM-GX5-25 Attitude Heading Reference System, and (d) Xsens MTi-100 IMU.

can output the information of motion state change at high frequency. However, IMU will also suffer from the error of drift accumulation and noise. The drift error increases sharply with time. Thus, suppressing the drift error from accumulation is one of the key issues for every IMU-based mobile mapping system. Generally, IMU could provide an initial value for LiDAR or visual odometry.

2.3 Mobile Mapping System

Due to the indisputable value of the 3D point cloud, many data acquisition systems have been developed over the last decades. The fixed mapping system has the advantage of high accuracy but the disadvantages are also obvious, low data capture efficiency, labor-intensive, and inflexible. The laser scanning systems mounted on mobile platforms seek to balance the costs, times, accuracy, and efficiency. Thus, the Mobile Mapping Systems (MMS) have been a valuable alternative to geospatial data acquisition. The application of MMS is widely used in several fields of industry and academia: urban environment, 3D modeling, surveying and mapping, cultural heritage, environmental monitoring, autonomous driving, robotics, surveillance and security, road management, construction site monitoring, etc. [142]

Due to the mobile platforms, MMS could collect data in dynamic environments and

some environments that are dangerous or not easy to access. Currently, for different requirements and different environments characteristics, the platforms utilized for MMS are various including UAVs, airplanes, vessels, vehicles, trolleys, backpacks, and helmets. The sensors could be installed on MMS including GNSS, INS, visual sensors, LiDAR, Radar, and other remote sensing sensors. Generally, in outdoor large-scale environments, MMS enables the rapid and accurate calculation of continuous 3D position, velocity, and attitude by combining GNSS information with INS. It should be indicated that all GNSS position and INS attitude data should be time synchronized with the mapping sensors for direct and precise mapping. However, in indoor environments or other GNSS-denied environments, no GNSS signal will be recorded. Thus, some positioning algorithms are utilized to calculate the position and attitude information by point clouds, images, or other remote sensing data. Simultaneous Localization and Mapping (SLAM) is one of the most important technologies in this field and will be reviewed in detail in Sec. 2.4.

MMS has the outstanding advantages of costs, times, and flexibility, while it also raises new challenges of data processing, dynamic object effects, environmental degradation, accuracy decrease, trajectory drift, and a large amount of data. Thus, recent trends in MMS have led to a proliferation of studies both in industry and academia. In this section, we will review some important and common MMS of UVAs, vehicles, trolleys, and backpacks.

Table 2.5: Comparison of some common MMSs on various platforms

Platforms	Products	LiDAR	Camera	IMU	GNSS
UAV	Riegl VUX-SYS system	√	√	√	√
	Leica Aibot SX	√	√	√	√
	LiAir V70 UAV 3D Mapping System	√	√	√	√
	DJI PHANTOM 4 RTK	×	√	√	√
	CHC Navigation P580 system	√	√	√	√
	CHC Navigation BB4 system	√	√	√	√
Vehicle	Leica Pegasus:Swift	√	√	√	√
	Leica Pegasus:Two Ultimate	√	√	√	√
	RIEGL VMQ-1HA Mapping System	√	√	√	√
	RIEGL VMX-2HA Mobile Mapping System	√	√	√	√
	RIEGL VMY-1 Mapping System	√	√	√	√
	RIEGL VMY-2 Mapping System	√	√	√	√
Trolley	Trimble GEDO CE 2 Rail Measuring System	√	×	√	√
	Viametris IMS3D Trolley	√	√	√	√
	NavVis M6	√	√	√	×
Backpack	Leica Pegasus:Backpack	√	√	√	√
	GreenValley LiBackpack C50	√	√	√	×
	GreenValley LiBackpack DGC50	√	√	√	√
	NavVis VLX	√	√	√	×
	Viametris BMS3D-HD Backpack	√	√	√	√
Handheld	GeoSALM ZEB Go	√	√	√	×
	GeoSALM ZEB Revo RT	√	√	√	×
	GeoSALM ZEB Horizon	√	√	√	×

Figure 2.7: Some common MMSs on various platforms. (a) Riegl RiCOPTER with VUX-SYS, (b) Leica Aibot SX, (c) LiAir V70 UAV, (d) DJI PHANTOM 4 RTK Drone, (e) CHC Navigation P 580 UAV, and (f) CHC Navigation BB4 UAV, (g) Leica Pegasus:Swift, (h) Leica Pegasus:Two Ultimate, (i)RIEGL VMQ-1HA Mapping System, (j) RIEGL VMX-2HA Dual Scanner Mobile Mapping System, (k) RIEGL VMY-1 Mapping System, (l) RIEGL VMY-2 Mapping System, (m) Trimble GEDO CE 2 Rail Measuring System, (n)Viametris IMS3D Trolley,(o) NavVis M6, (p) Leica Pegasus:Backpack, (q) GreenValley LiBackpack C50, (r)GreenValley LiBackpack DGC50, (s) NavVis VLX, (t) Viametris BMS3D-HD Backpack, (u) GeoSALM ZEB Go, (v)GeoSALM ZEB Revo RT, (w)GeoSALM ZEB Horizon.

The UAV-based MMS allows the aerial data capture from a height of tens of meters or even thousands of meters. UAV mobile measurement data can be combined with other measurement technologies, such as GNSS, backpack-based MMS data, and vehicle-based MMS data. The fusion and unification of measurement data from these different platforms can provide a more complete measurement of the urban environment. UAVs can quickly survey a site from the air, enabling the completion of projects at a lower cost and with little personnel. It is more suitable for power line inspection, forestry survey, mining survey, and urban environment survey. The UAV MMS can be equipped with many different sensors, such as laser scanners, RGB cameras, thermal infrared cameras, etc. Due to its inherent advantage of being able to measure data from the air, it can accomplish both large-scale topographic data acquisition and provide data support for fine modeling. The UAV MMS is currently limited by their battery capacity and load capacity, making them unable to work for a long time.

The vehicle-based MMS is the most commonly used data collection method for urban measurements, including high definition maps (HD Map) for autonomous driving, 3D modeling of cities, etc. Vehicle-MMSs could only be used in outdoor urban or large-scale environments due to the environmental requirements of vehicle driving. Equipped with GNSS, IMU, LiDAR, cameras, and radars, vehicle-MMSs could always collect abundant, high-precision, high-density, and large-volume data.

Unlike MMSs on UAV and vehicle platforms, which are suitable for large-scale outdoor scenes, trolleys, backpacks, and handheld MMSs are dedicated to small-scale scenes, especially for indoor, underground, and other GNSS-denied environments. However, they are still some differences in their measurement applications. The trolley-MMS can only run in a flat indoor environment and cannot be used in staircases, construction sites, and other rough-ground scenarios. Backpack-MMS can measure the environment that surveyors can access. The handheld platform is more suitable for small-scale scenes, especially for the detailed data collection of pipes, ceilings, corners,

etc. Besides, robots are also widely used in measurement nowadays, but there are still no mature and widely used industrial solutions. Thus, robot-based MMSs are one of the hot topics both in academia and industry.

2.4 SLAM

In the years when SLAM was just proposed, researchers mainly focused on how to apply probability methods to the field of robot technology. In the subsequent development, the solutions of SLAM tend to be diverse. In this section, the SLAM methods will be reviewed according to the basic theory, SLAM based on filtering theory, and SLAM based on graph theory. In the last decades, deep learning boomed in many research fields. In SLAM applications, deep learning also was utilized and achieved significant results. Thus, SLAM based on deep learning will also be reviewed especially.

2.4.1 SLAM based on Filtering Theory

This kind of method is to consider the solution of the SLAM problem as an estimation process. In this research field, some famous research achievements include introducing Kalman Filter (KF) model, Extended Kalman Filter (EKF) model, the Particle Filter (PF)model, and the Maximum Likelihood Estimation (MLE) into SLAM. The idea of SLAM was introduced in [87], which used EKF to solve the SLAM problem. The work was inspired by the work done in [137]. The method uses a probabilistic approach to limit the influence of errors on the accuracy of the generated maps [110]. Since then, this has become the basic processing flow for SLAM problems, and EKF has gradually become the most commonly used method in newly proposed SLAM algorithms.

EKF is one of the earliest theoretical methods applied in SLAM. Many successful implementations of this approach have been reported in indoors [17], outdoors [58],

sub-sea [88] and air-borne [75] applications. The EKF assumes that the noise conforms to a Gaussian distribution, which leads to the assumption that the environmental characteristics are unique and the EKF algorithm is prone to divergence when this assumption is not true. Moreover, EKF uses the covariance matrix between the robot and the environmental eigenvalues. To eliminate the cumulative error, the matrix is processed every time it is estimated and corrected, which makes the computational complexity reach a high-level [116], and seriously restricts the method in large-scale environment applications [18].

Another solution is based on Bayes Filter theory, which can represent any probability density function. However, the original Bayes Filter SLAM is difficult to use in practical applications, due to the huge computation amount. To reduce the computation complexity, finding a functional representation for probability distributions is a useful method. The Sum of Gaussian (SOG) method provides such a representation. Durrant-Whyte et al applied SOG to represent the environmental landmark feature model in the full Bayesian algorithm [45]. This method has the advantage of being computationally tractable and indeed can be implemented with many of the same rules that are employed in Kalman filtering. FastSLAM is a solution approaching the SLAM problem from a Bayesian point of view, which will recursively estimate the full posterior distribution over robot pose and landmark locations [107].

Particle filtering or Monte-Carlo methods are another SLAM method based on Bayes Filter theory, which aims to provide a complete representation of the joint posterior probability using a large set of sample points, also called particles. These points provide a faithful state approximation to the true shape of the full distributions employed. State propagation and observation models are also represented in the form of a sampled distribution. Particle filters can perform system state estimation in a recursive form. Particle filtering is to obtain the proposed distribution of states from the known prior information, and then extract some of the particles from the proposed distribution and perform iterative operations. In this process, the weight of each par-

ticle is obtained, and the particle state and weight will be updated continuously. The approximate posterior probability distribution of the system state will be obtained according to the weight. In the current open-source algorithms, the GMapping algorithm is an improvement and implementation of the particle filter algorithm [57]. The GMapping algorithm can compute an accurate proposal distribution, taking into account not only the movement of the robot but also the most recent observation, which will drastically decrease the uncertainty about the robot's pose in the prediction step of the filter.

Cadena et al regarded the year 2004 as the watershed of SLAM algorithm research [14]. They called the previous two decades the classical age. The classical age saw the introduction of the main probabilistic formulations for SLAM, including approaches based on extended Kalman filters (EKF), Rao-Blackwellized particle filters, and maximum likelihood estimation. Moreover, it also delineated the basic problems of efficiency and robust data association. The subsequent period is the algorithmic-analysis age (2004-2018). The algorithmic analysis period witnessed the research of basic problems of SLAM, including observability, convergence, and consistency. In this period, the significance of sparsity towards efficient SLAM solvers was also analyzed. Besides, some far-reaching open-source SLAM frameworks were developed.

2.4.2 SLAM based on Optimization Theory

Lu and Milios proposed a new SLAM algorithm based on graph optimization. They used the pose graph to represent the SLAM problem. The nodes in the pose graph represented the poses of the robot at different times. The edges in the pose map represented the pose constraints between different nodes. Although the algorithm computationally efficiency is relatively low when building large maps, many researchers have discovered the prospects of this research direction and have invested a lot of energy into it. Graph-based SLAM methods are post-processing methods, also called

full SLAM methods.

With the continuous development of graph optimization, graph-based SLAM algorithm research has become the mainstream trend of SLAM algorithms currently, because graph optimization not only has advantages in accuracy and efficiency in practical applications but also has an elegant framework and can stand the test of practice. Konolige et al [80] explored the sparsity of graph-based two-dimensional SLAM problems, then an efficient solution of graph-based SLAM problems has been proposed. PTAM (Parallel Tracking And Mapping) is the first SLAM method to separate tracking and mapping as two threads [76]. It is a monocular vision SLAM algorithm based on keyframes. The main steps of PTAM include FAST corner detection [124], map initialization, tracking localization, keyframes selection, relocalization, bundle adjustment, etc. ORB-SLAM is another famous feature-based monocular SLAM system that can operate in real-time, in small and large indoor and outdoor environments [109]. The system is robust to severe motion clutter, allows wide baseline loop closing and relocalization, and includes fully automatic initialization. Building on excellent algorithms of recent years, ORB-SALM can be seen as an extension of PTAM, which adds another thread called loop closing to the original tracking and mapping threads of PTAM. Its performance is better than PTAM in most instances. Besides, it also integrates covisible graphs, relocalization, and loop closing based on the DBoW2 library. However, in the ORB-SLAM program, the FAST features are replaced by ORB features. Until now, ORB-SLAM has been keeping updating and optimizing, and new research results are achieved.

As for LiDAR-based SLAM, there were tremendous achievements in the last decade. LOAM [171] is a milestone of LiDAR-SLAM. It achieves practical LiDAR odometry with both low-drift and low-computational complexity. It does not need high accuracy ranging or inertial measurements. It builds a general odometry pipeline for subsequent LiDAR-SLAM frameworks. SA-LOAM [91] integrates semantics in odometry and loop closure detection with LOAM. Loam_ livox [97] is a LOAM framework

using point clouds data generated by non-repetitive scanning. Edge features and planar features are used for LiDAR odometry. F-LOAM [152] extracts edge and planar features from each scan and is registered to a local edge map and a local plane map separately, where the local smoothness is also adopted for iterative pose optimization. Lego-LOAM [131] is a lightweight system. It also uses the planar and edge features to solve the 6-DoF transformation parameter by a proposed two-step Levenberg-Marquardt optimization method. T-LOAM [174] utilizes a hierarchical feature-based LiDAR-only odometry that performs precise pose estimates by extracting four peculiar features: edge features, sphere features, planar features, and ground features. HDL Graph SLAM [78] proposes a general LiDAR-SLAM pipeline: front-end LiDAR odometry and back-end loop closure detection and graph-based pose optimization. SUMA++ [25] uses a efficient surfel-based mapping method [10] and exploits 3D point clouds by integrating semantic information. The semantic information is extracted by RangeNet++ [106]. LIO-SAM [132] is a tightly-coupled LIO using smoothing and mapping. It uses points and planar features to perform odometry. A factor graph is used for multi-sensor fusion and global optimization. Then, the visual sensor is also integrated to propose an updated LVI-SAM [133]. FAST-LIO [166] fuses LiDAR feature points with IMU information by a tightly-coupled iterated extended Kalman filter with an optimized Kalman gain. Then, two aspects of optimization including registering raw points to the maps and maintaining a map by an incremental k-d tree data structure (ikd-Tree), are conducted to FAST-LIO2 [165]. LiTAMIN [168] proposes an optimized ICP method stabilized with normalization of the cost function by the Frobenius norm and a regularized covariance matrix. The cost function is further optimized in LiTAMIN2 [169] by introducing symmetric KL-divergence that reflects the difference between two probabilistic distributions. The current LiDAR-SLAM generally contains two main modules front-end and back-end. The front-end always contains data preprocessing, point cloud filtering, and LO, while the back-end always contains LCD, pose optimization, and mapping. According to the aforementioned LiDAR-SLAM review, features adopted in LiDAR odometry

generally are corner points, edge features, and planar features.

SLAM based on optimization is the current mainstream direction of SLAM research. This method divides SLAM problems into two steps, front-end data alignment, and back-end optimization. Loop closure detection is one of the key issues that remain unresolved for SLAM based on optimization. Besides, there are still some problems unresolved [96]:

1. Efficiency problem. When the SLAM method based on graph optimization was just proposed, the method based on nonlinear least squares is adopted. However, because it does not consider the sparse structure in the SLAM problem, and even directly solves the problem by using the matrix inversion method, the solution efficiency is very low. Based on the relaxation and stochastic gradient descent method, the solution efficiency is improved to some extent, but it does not fully utilize the advantages of the nonlinear least-squares problem. For example, the stochastic gradient descent method only uses the first-order property of the function, and the convergence is very slow when the optimal solution is approached. In the case where the number of iterations is limited, the accuracy of the result is affected. The method recently proposed based on nonlinear least squares not only makes full use of the sparse structure in SLAM [81] but also draws important research results from sparse linear algebra. This greatly improves the solution efficiency of the problem, and the scale of the problem can be greatly improved, representing the current high level of the field.

2. Robustness problem. The robustness of the solution method is mainly considered from two aspects, one is the dependence on the initial value, and the other is the adaptability to the error loop closing information. (1) Robustness to initial values. Since the odometer (including the wheel odometer and other methods relying on observation information for self-motion estimation) information may have a large cumulative error, there is a large deviation between the initial value and the true value of the resulting pose sequence. Therefore, reducing the dependence on the initial value is important to enhance the convergence domain of the method. There are two solu-

tions to this problem: one is to improve the global search ability of the optimization method itself, and the other is to quickly obtain an acceptable initial value by other methods. Carlone et al. [16] proposed a linear approximation of SLAM based on graph optimization and gave an analytical solution. The method has no dependence on the initial value, and the result can be used as the initial value of the nonlinear least-squares method. However, this method is currently only available for 2D SLAM. (2) Robustness to error loop closure information. Conventional graph optimization methods usually assume that the graph has the correct topology [113]. If the map introduces improper positive loop closure, it may lead to erroneous convergence results. This is because the least-squares optimization method itself is not robust to outliers. To ensure that the map obtained during convergence is correct, strict constraints can be imposed on the ring closure, so that the error rate of detection is low enough and the effect is reduced by the kernel function method.

3. Scalability problem. The SLAM method based on graph optimization takes the pose of the robot as the node. Usually, the longer the trajectory of the robot, the more pose nodes that need to be processed, which is not conducive to the expansion of the method. When the robot is walking in a fixed-size environment, the number of nodes in the graph should be related to the size of the environment and not to the length of the motion trajectory. To make the SLAM method have good scalability, the key is to effectively control the graph nodes. The most direct way to reduce the number of nodes in the graph is to limit the distance between nodes. Only if the distance between nodes exceeds a certain threshold, the node can be added to the graph [79].

2.4.3 SLAM based on Deep Learning

Motivated by the success of deep learning applied in many fields, many researchers contribute to using deep learning in SLAM components. Generally, according to

whether it is based on deep learning, we divided SLAM frameworks into learning-based SLAM and geometry-based SLAM, also called traditional SLAM frameworks. There have been many successful approaches in visual-SLAM, including feature extraction, odometry, loop closure detection or relocalization, and semantic SLAM. These aspects are also hot topics in LiDAR-based SLAM. Although learning-based SLAM approaches still do not significantly outperform traditional SLAM solutions, they provide new solutions to SLAM problems and are proven to be more robust than traditional SLAM. Thus, learning-based SLAM methods are well-worthy to research.

Instead of manually estimating the geometry of the environments, learning-based approaches automatically learn the features, correspondences, and relationships between sources and targets for visual odometry (VO) and LiDAR odometry (LO). We will briefly review some significant learning-based VO methods, followed by learning-based LO.

An end-to-end SLAM framework DeepVO [154] is dedicated to monocular visual odometry problems by using deep Recurrent Convolutional Neural Networks (RCNNs) [42]. Features are extracted from a pre-trained FlowNet [43] and then forwarded to LSTM. VINet [32] is an on-manifold sequence-to-sequence algorithm of visual-inertial odometry (VIO). The proposed learning-based VIO eliminates the need for tedious synchronization of the camera and IMU and the need for calibration between the IMU and camera, which are two fundamental problems for traditional SLAM solutions. VidLoc [31] is a recurrent model performing 6-DoF pose estimation of video-clips. LS-VO [34] jointly estimates a low dimensional representation of dense optical flow manifold based on Auto-Encoder (AE) and meanwhile computes the camera ego-motion estimation by a standard convolutional network. Chen et al [20] propose a generic framework to learn selective sensor fusion which can be visualized and interpreted enabling more robust and accurate ego-motion estimation. UnDeepVO [93] is an unsupervised deep learning scheme that enables the estimate of the 6-DoF pose of a monocular camera and the depth. GANVO [3] is a unsupervised

learning algorithm that estimate the pose information and depth information of the environments by unsupervised deep convolutional Generative Adversarial Networks (GANs) [55]. DL_Hybrid [9] can extract effective key points from each frame even in extreme scenes, and it has good performance even in extreme moving conditions. In the proposed framework, two deep learning neural networks are designed to extract feature maps between image frame pairs. One named DenseFlowNetwork is dedicated to estimating the dense optical flow map, and another named DenseDepthNetwork is proposed to extract the dense depth map per frame. To sum up, learning-based VO or VIO systems are realized by training in a supervised or self-supervised manner to end-to-end estimate the pose. However, although the ability of networks can be improved by increasing the number of training data sets and optimizing the network structure, insufficient generalization ability and insufficient accuracy problems are inevitable.

As for learning-based LO, [112] proposed a two-stream CNN architecture for frame-to-frame point cloud odometry. The proposed method transforms high-dimensional point cloud data to a depth image that could be fed into a CNN to perform motion estimation directly. This method circumvents the defects of high computational constraints associated with traditional scan matching. Different from existing LO pipelines that go through individually designed feature selection, feature matching, and pose estimation, networks trained in an end-to-end manner are new research trends in this field. DeepPCO [155] is a dual-branch scheme to infer 3D translation and orientation separately, which is trained in an end-to-end fashion. LO-Net [92] is a real-time LiDAR odometry estimation framework, which is also trained in an end-to-end manner. A scan-to-map module is proposed to improve the odometry accuracy by utilizing the geometric and semantic features. DeepLO [27] is a geometry-aware deep LiDAR odometry framework that is trainable via both supervised and unsupervised manners. The unsupervised LO approach proposed in [28] introduces the uncertainty-aware loss with geometric confidence to enable the reliability of the

proposed pipeline. DMLO [95] enforces geometry constraints in the framework and decomposes the 6-DoF pose estimation into two parts. A learning-based matching network is designed to extract high confidence correspondences from successive LiDAR scans. Then, rigid transformation estimation is performed by Singular Value Decomposition (SVD). SLOAM [24] introduces an end-to-end pipeline for tree diameter estimation based on semantic segmentation and LiDAR odometry and mapping in forest scenes. A synthetic network based on deep learning is proposed for achieving an integrated navigation performance of LIO [138]. The networks proposed in [149] could perform fast, real-time and precise estimation of translation. 3DRegNet [114] is a deep neural network including two sub-blocks: classification block of the point correspondences into inliers/outliers, and regression block of the motion parameters. A transformation estimation approach using SVD is also used as an alternative to deep neural network registration. PointNetLK [5] uses PointNet [19] as an image function. It combines PointNet and Lucas & Kanade algorithm into a single recurrent deep neural network. Odometry is the core problem of SLAM, which has been researched for many years. Deep learning networks always are more robust than traditional geometry-based methods. However, the precision and computation cost still need to be optimized.

2.5 Loop Closure Detection

LCD is to check whether the robot revisits the same scene. It is a significant method to control or even eliminate the cumulative error. The loop closing problem is one of the most important problems in SLAM which remains unresolved. In SLAM based on graph optimization methods, there are two types of edges in a graph, one kind of edge is obtained from LiDAR odometry, the other kind of edges is to provide control information, which is generally obtained from loop closure detection.

In recent years, SLAM solutions based on graph optimization gradually become the

mainstream of research. As the key problem in back-end optimization of SLAM, loop closure detection also attracts much research interest. Loop closing is vital for cumulative error reduction or even elimination. However, if the loop information is wrong, it means a disaster to the global map, which may result in the inconsistency of the map. Therefore, LCD methods should achieve high accuracy and recall rate as much as possible. The conventional way to detect loops is to match the current scan with all the scans collected before, but it will waste much time and cannot achieve a real-time level. With the measuring time and measuring distance increasing, the computation amount will increase exponentially. Thus, to reduce the number of matches, some researchers will not match the current scan with all the scans collected before but select some scans to match the current scan. However, this method cannot make sure that the frames which will form a loop can be selected. To achieve high accuracy and high recall rate, many researchers develop algorithms to detect loops. According to basic theory, LCD algorithms can be divided into two categories: pose probability estimation and scene appearance matching.

2.5.1 LCD based on Pose Probability Estimation

Approaches based on pose probability estimation are to detect loops according to the location reckoning of the measuring equipment. This approach implements in the process of map construction and mobile navigation synchronously.

Haris [8] constructs the 3D features of the environment based on laser ranging data and visual data fusion, then the probability of whether the two frames of data are collected from the same environment are estimated based on these accurate features. Data fusion is achieved by validating 3D structure assumptions formed according to 2D range scans of the environment, through the exploitation of visual information. Jochen [139] proposes a heuristic LCD method based on the 3D point cloud. The pose of the scanner is corrected timely by loop information and constructs a sparse

SLAM map without iterative optimization, then the LCD process can be conducted in real-time. Gong [108] conducts EKF-SLAM to realize localization and mapping in real-time based on laser range data. And a new description method that is similar to the raster map is proposed for the local map. Then LCD is conducted based on the local map description. Many research results are achieved based on this approach, however, it is a passive LCD method, which is inflexible and shows high complexity in model construction and calculation. Therefore, many researchers choose scene appearance matching for LCD.

2.5.2 LCD based on Scene Appearance Matching

LCD in monocular visual SLAM can be divided into three categories according to the data association method. Williams [159] summarized and analyzed the three strategies from two aspects of theories and experiment results.

- Map to map matching strategy. Appearances and relative positions are considered to find correspondences between two submaps. Clemente [33] proposed a new method based on the variable scale geometric compatibility branch-and-bound (GCBB) algorithm to detect loops. A new visual map matching algorithm stitches these maps together and can detect large loops automatically, taking into account the unobservability of scale intrinsic to pure monocular SLAM.

- Image to image strategy. Cummins [35, 36] used visual appearance to describe the environment and to loop closing by identifying the visual appearance of the area which has been explored before. Bag-of-Word (BoW) model was used to describe every image. And a method that is similar to text retrieval methods is adopted to match search in visual word space.

- Image to map strategy. Williams [158] proposed an LCD method based on

relocalization technique. The relocalization model is constructed based on the similarity of feature points between images and sub-maps, then the pose of the camera relative to the environment will be get according to RANSAC (Random Sample Consensus) algorithm and three-point pose calculation algorithm.

In addition, Liu [99] adopted feature matching to detect loops directly to avoid perceptual ambiguity and constructs the KD tree according to the features in every frame of images to realize real-time loop closure detection. Labbé [83] proposed a graph-based global LCD approach. This approach can realize online detection and correct mapping result in real-time. In recent years, with machine learning and deep learning methods attracting much research interest, those methods are applied in many research areas and show better performance. Some researchers also adopt these thoughts into loop closure detection. In [56] the loop closure detection problems are treated as a classification problem. Geometric features and range histograms will be adopted to describe the external environment, and training samples will be generated based on the features to train an AdaBoost classifier. Then, the classifier will have the ability to distinguish whether two scans of the point cloud are collected from the same scene. Hao et al [120] proposed a new loop closure detection algorithm that combines both camera and LiDAR sensors. In the image matching session, an improved learning-based descriptors generator with triplets and an adaptive max-pooling layer will be adopted to generate more robust and accurate descriptors. To improve the accuracy of loop detection, the sliding window will be used first to find the exact matching position of the scan in the corresponding sub-maps, and then image matching to remove errors and increase the optimization conditions. Experimental results show that the method is helpful to improve the accuracy of the SLAM algorithm. In the application field, DBoW2 [51] is a very famous library for loop closure detection. DBoW2 received a great deal of attention, because of its good performance in ORB-SLAM2. It is a model for visual place recognition based on bags of words obtained from FAST keypoints [125] and BRIEF descriptors [15].

Chapter 3

FastLCD: A Fast and Compact Loop Closure Detection Approach

This chapter proposes a fast and compact loop closure detection method based on comprehensive descriptors and machine learning using 3D point clouds for indoor LiDAR mobile mapping. Comprehensive descriptors proposed in this chapter encode discriminative multi-modality features to describe each scan of point clouds. The specific values of descriptors of point cloud scan pairs are fed into a machine learning model. We leverage the pre-trained learning model as a classifier to distinguish whether a pair of laser scans is a loop candidate. Then, to ensure the results' precision, a novel double-deck loop candidate verification strategy is used to reject false positives. The algorithm is evaluated on datasets of some typical indoor environments. Compared with some state-of-the-art loop closure detection algorithms, the proposed FastLCD algorithm demonstrates superior performance in precision and recall rate. Moreover, the method proposed also exhibits high time efficiency, excellent generalization performance, and insensitivity to threshold changes.

3.1 Introduction

Surveying robots are being increasingly used for mobile mapping and model reconstruction, especially in environments that lack the reliable signals of the global navigation satellite system (GNSS) or other localization methods, such as indoor environments, city canyons, and underground scenes. In such environments, sensor-based localization methods are adopted, such as scan matching and visual odometry. However, these methods cause a drift in position because of the generation of cumulative errors. With increasing measurement distance, the drift also grows sharply [147]. Loop closure detection is a key step in SLAM to restrict these cumulative errors. It can be defined as a data association problem that aims to determine whether a place has been previously visited. It can reduce the pose estimation uncertainty and map inconsistency [160], to construct an accurate and consistent map for model reconstruction [135].

So far, several loop closure detection algorithms using 2D point cloud data [64, 65, 179] or visual data [4, 82, 83, 105] have been proposed. However, a 2D laser scanner only captures environmental information on a plane in each frame, which does not contain sufficient information [90]. When robots are running on a rough floor, the 2D scans might appear markedly different despite only slight changes in the position. Thus, in such cases, loop closure detection based on 2D scans is not reliable. Visual data are also widely used as they contain adequate information about environments with much lower costs. However, visual sensors are sensitive to illumination conditions [22, 85], especially in indoor environments. Recently, due to the declining costs, 3D laser scanners have been widely applied in many fields. They capture more information than 2D scanners do and work stably under illumination changes, even in dark environments.

In this chapter, the loop closure detection problem is treated as a classification problem to identify whether two scans are captured from the same environment. We

propose a loop closure detection method in indoor environments based on comprehensive descriptors and machine learning. Considering multi-modality information in 3D point clouds, the comprehensive descriptor is proposed to describe each point cloud scan. The descriptor ratios are fed into a supervised learning model to detect loop closures. Loop results provide control information for back-end optimization in SLAM [144], enhancing the reliability of the adjustment network and improving the map's overall accuracy. False loop closures will have disastrous effects on SLAM results. Thus, a double-deck verification strategy is used to reject false detection to ensure the algorithm's precision.

The main contributions of this chapter are:

- The FastLCD algorithm leverages multi-modality features to transform a point cloud scan to a global comprehensive descriptor. The multi-modality features are extracted from each single raw 3D LiDAR scan without any transformation or projection. Theory analysis and experiment results demonstrate the comprehensive descriptor is discriminative to location and external environments.
- A highly-efficient supervised learning model is used as a classifier to identify loop closure candidates without prior poses. The model can also provide an estimate of the reliability of detection results. The detection results comply with Gaussian Mixture Model (GMM), which indicates the method proposed has great separability and insensitivity to threshold changes.
- A double-deck loop verification strategy comprising cross-validation and post-verification is implemented to reject false positives to ensure precision.

The rest of this chapter is organized as follows. In Sec. 3.2, a brief review of loop closure detection methods is summarized and classified. Their problems and limitations are also stated in this section. In Sec. 3.3, the FastLCD method based on comprehensive descriptors and machine learning are introduced in detail. The discrimination

of multi-modality features is also demonstrated. Section 3.4 shows the experimental results of our method and comparison algorithms in indoor environments. Sec. 3.5 discusses the performance and limitations of the FastLCD algorithm. In Section 6, conclusions are stated, and directions of future work are presented.

3.2 Related Works

Loop closure detection is a critical component towards addressing the problem of SLAM. This section briefly summarizes the previous work related to loop closure detection using point clouds. Loop closure detection algorithms can be classified into four categories according to the features adopted: (1) based on local features, (2) based on handcrafted global descriptors, (3) based on planes, objects, or semantic information, (4) based on deep learning.

Many of the traditional approaches are based on local features, such as key points. Steder et al. [140] proposed a place recognition method based on point features extracted from 3D range data. The obtained points of interest were applied to extract features and score candidate transformations. Then, a threshold was applied to validate the candidates. In [141], normal aligned radial features were applied using bag-of-words models. The fast point feature histograms (FPFHs) proposed in [128] optimized the traditional point feature histograms (PFHs) [129]. The computation of FPFH was based on the combination of geometry relations between the key points and neighbors. The FPFHs not only retained most of the descriptive power of a PFH but also could be computed online for real-time application.

Regarding methods based on handcrafted global descriptors, substantial achievements have been gained. Magnusson et al. [103] exploited the surface representation of a normal distribution transform to create feature histograms. Here, a point cloud scan was split into several overlapping grids, and their linear, planar, and spherical properties

were computed and compressed into a shape histogram. In addition, expectation-maximization was used to fit a gamma mixture model to output similarity measures for automatically determining the threshold for loop closure detection. Röhling et al. [123] proposed a fast histogram computed from the distances between the point and robot. A discrete Wasserstein metric was used to compare the two histograms, and loop closures were detected using an appropriate distance threshold. Considering structural information, a non-histogram-based global descriptor from 3D LiDAR scans, called scan context [74] was proposed. This approach directly recorded the 3D structure of a space that was invariant to LiDAR viewpoint changes. Scan context is also expanded to intensity scan context [153], considering intensity information. Granström et al. [56] used two types of global features, geometry features, and range histograms. AdaBoost is used to learn a classifier from these features. The approach proposed in [178] comprised local speeded-up robust features (SURFs) and global spatial features for the place recognition task. M2DP [62] was a global descriptor produced by projecting a 3D point cloud to multiple 2D planes and computation of the signatures of the cloud on these planes. LiDAR Iris [156] is a binary signature image representation. Place recognition is implemented by calculating the Hamming distance as similarities of two corresponding binary signature images.

Besides artificially designed local and global features, some algorithms based on advanced features, like planes, objects, or semantic information are proposed. Dude et al. [44] proposed 3D segmentation methods and realized place recognition through segment matching and geometry verification. Cupec et al. [37] proposed an indoor place recognition approach based on matching planar surface segments and straight edges in-depth images obtained from RGB-D images. Luo et al. [102] proposed a scene recognition algorithm based on object descriptors, including the oriented, unique, and repeatable-clustered viewpoint feature histogram descriptor [2] and an ensemble of shape functions descriptor [161], which were extracted from the submaps segmented from the RGB-D range data. Furthermore, a distance metric was learned

[38], to increase the precision of place recognition under environmental changes.

Each type of feature has inherent disadvantages. Local features generally lack descriptive power and suffer from ambiguity and environment changes, while global descriptors always face problems of view-dependent and invariance. Algorithms based on planes, objects, or semantic information rely on the performance of these advanced feature extraction. Thus, multi-modalities integration is an effective approach to remedy the defects of a single feature. The mining of point cloud features aims at describing environments more comprehensively and discriminatively, which found a basis for our FastLCD method, a feasible and reliable loop closure detection algorithm. Algorithms also need to balance performance in terms of accuracy and efficiency. In addition, the majority of the existing methods rely on appropriate threshold setting, which needs to be adjusted on new datasets, while the proposed FastLCD could use a uniform threshold ignoring dataset changes.

3.3 Methodology

The input of this algorithm is raw point clouds without any transformation and projection. Multi-modality features are extracted directly from raw 3D LiDAR scans. Then, they are concatenated into a discriminative global comprehensive descriptor, by which the computational and storage cost will reduce significantly. The descriptor ratio is calculated from a pair of comprehensive descriptors. The descriptor ratio will be used to be checked by the pre-trained machine learning model without any prior pose information to obtain loop candidates. Then, a double-deck verification strategy comprising cross-validation and post-verification is implemented to ensure the final results' precision. The effective and efficient loop closure detection results will greatly enhance the localization and mapping tasks in the application of robotics and self-driving. The algorithm architecture is shown in Fig. 3.1.

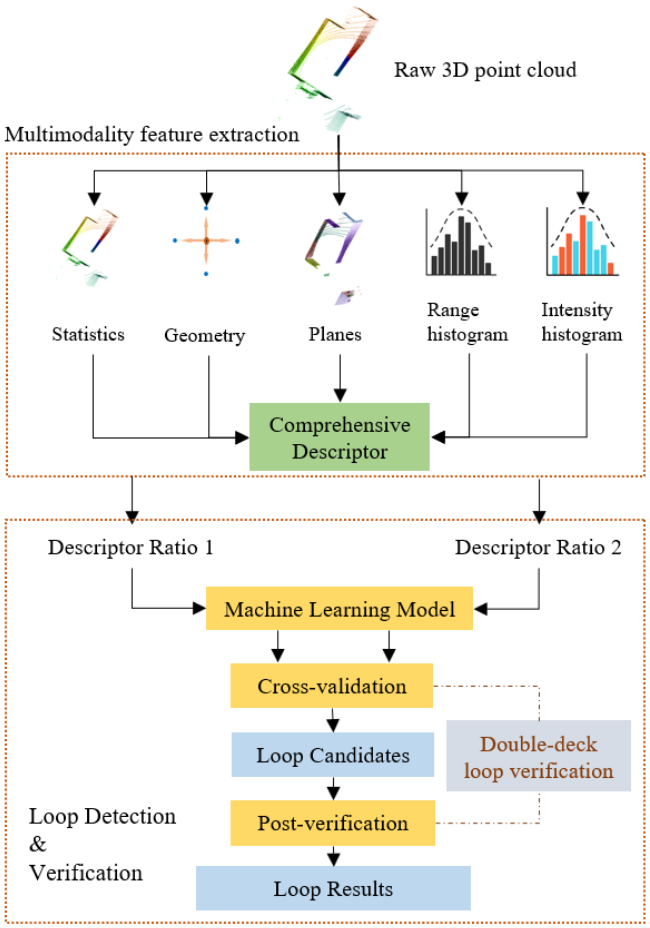


Figure 3.1: Flowchart of proposed FastLCD algorithm based on a comprehensive descriptor and machine learning. The workflow includes two main modules: multi-modality feature extraction, loop detection, and verification.

3.3.1 Multi-modality Feature Extraction

The proposed FastLCD algorithm leverages discriminative global comprehensive descriptors encoded by multi-modality features, which are extracted from each single 3D LiDAR scan, including statistics, geometry, planes, range histogram, and intensity histogram. The multi-modality features are all invariant to rotation. Therefore, the FastLCD algorithm is also rotation-invariant.

Notations: Given a point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$, i is the point ID ($i \in [1, N]$) and (X_i, Y_i, Z_i) are the point coordinates. f_m^{id} is defined as the features, with id and m denoting the scan ID and the feature ID, respectively.

3.3.1.1 Statistics

The statistical features are computed using the nominal range distance, point coordinates, and point number. They can reflect the point distribution, which represents the surrounding environment intuitively.

The statistics include mean value of X, Y, and Z respectively. $(\bar{X}, \bar{Y}, \bar{Z})$, mean measuring distance $(\bar{R})$, maximum and minimum of distances (R_{max}, R_{min}), standard deviation (σ_R), coordinate of mass center $(\tilde{X}, \tilde{Y}, \tilde{Z})$, average distance between each point and mass center $(\bar{\tilde{R}})$, skewness (S_R), and kurtosis (γ_R).

3.3.1.2 Geometry Features

Geometry features (F_g) describe the contextual information of each point on one scan line. Features of each point are computed concerning the adjacent points. Geometry can describe local information of the point cloud.

(1) Sum and standard deviation of distances between adjacent point $(D_{i,i+1})$ on the same scan line.

(2) Sum and standard deviation of curvatures. A is defined as the area of a triangle formed by three points p_{i-1}, p_i, and $p_(i+1)$. The distances among the three points are $D_{i,i+1}$, $D_{i-1,i}$, and $D_{i-1,i+1}$. The curvature (C_i) at p_i is computed as

$$C_i = \frac{4A}{D_{i,i+1}D_{i-1,i}D_{i-1,i+1}}, \tag{3.1}$$

in which

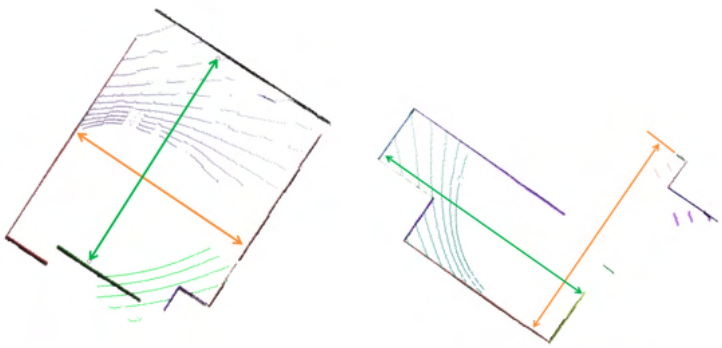
$$A = \sqrt{s(s - D_{i,i+1})(s - D_{i-1,i})(s - D_{i-1,i+1})} \tag{3.2}$$

$$s = \frac{D_{i,i+1} D_{i-1,i} D_{i-1,i+1}}{2} \tag{3.3}$$

(3) Mean value and standard deviation of range ratios on one LiDAR scan. Range ratio is defined as the specific value of ranging distances of adjacent point pairs.

(4) Mean value and standard deviation of range differences. Range difference is defined as the difference value of ranging distances between adjacent point pairs.

3.3.1.3 Planar Features



(a) (b)

Figure 3.2: Planar features in indoor environments. The green arrow indicates the maximum distance between two parallel planes, while in the vertical direction of those two planes, the maximum distance is denoted by orange arrows.

Planar surfaces are the most common geometric structure in man-made environments, such as ground, walls, ceilings, and furniture. These plane features can reflect the environment's structural information and complexity, as shown in Fig. 3.2. We define some structural features based on these plane features. The parallel planes in an indoor environment always refer to ground, ceiling, or walls. Thus, the features can be designed as (1) the number of plane features in a point cloud scan, (2) the

maximum distance between two parallel planes P_1 and P_2, (3) The maximum distance between two parallel planes, which are vertical to P_1 and P_2, (4) structure index: the ratio of two maximum distances computed in (2) and (3). The ratio can reflect the shape of the indoor space. Generally, the man-made environment is four-sided. The ratio ranges from 0 to 1. If it is close to 0, the environment is long and narrow, such as corridors and tunnels. Meanwhile, if it is close to 1, the environment is more like a square. In our approach, the plane feature extraction algorithm is proposed by [46].

3.3.1.4 Range Histogram

Each LiDAR scan is represented as an unstructured and uneven distributed 3D point cloud and always associated with a location. With the measuring distance defining the range between each point and the sensor's center, the range histogram describes the distribution of the points in a scan and reflects the environment's size and complexity.

Assuming a bucket count b and a value range $R \in [R_{min}, R_{max}]$, we can divide$R$ into sub-intervals of size.

$$\Delta = \frac{1}{b}(R_{max} - R_{min}) \tag{3.4}$$

Each point falls in a corresponding bucket according to the value of R.

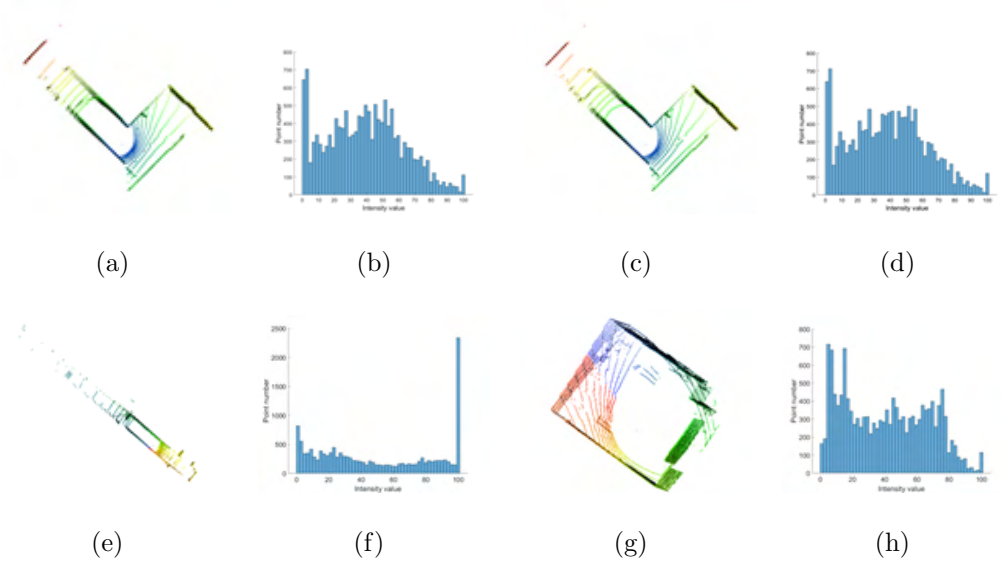
$$(R_{min} + k \cdot \Delta) < R < [R_{min} + (k+1) \cdot \Delta] \tag{3.5}$$

Then, the histogram for a point cloud scan $\mathbf{P}$ can be written as:

$$H_b = (h_0, h_1, \cdots, h_{b-1}), \quad h_k = count(p_k)/N, \tag{3.6}$$

in which $count(p_k)$ refers to the point number in bucket k. Theoretically, the number of measurement points in each LiDAR scan is relatively constant. While in practice, the number will show slight difference due to specific reflection in some surface types, or the slight vibrations of rotating devices. The normalization of point count ensures that the range histograms remain comparable under these unexpected conditions.

3.3.1.5 Laser Intensity Histogram



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

(e) (f) (g) (h)

Figure 3.3: Intensity histograms of different environments. (a) and (c) are at different locations in the same corridor, (e) is a scan of a long corridor, (g) is captured in an office room. (b), (d), (f) and (h) are the intensity histogram of (a), (c), (e) and (g) respectively. (b) and (d) shows the similar trend and peakness, while they are different from those in (f) and (h).

Similar to the range histograms, the laser intensity of the points in a scan can also be counted as a laser intensity histogram. In Fig. 3.3, if the two point clouds are captured in the same environments, the intensity histograms are also similar. By contrast, the laser intensity histograms of different environments vary on trends and peaks.

The histogram reflects the measuring distance and object attributes. It should be indicated that the intensity value of different laser scanners is defined differently. Thus, normalization of intensity value should be implemented for each point. The calculation method of intensity histogram is similar to range histogram.

3.3.2 Feature Discriminative and Rotation Invariant Analysis

Loop closure detection is to find whether the location has been revisited. To ensure the algorithm's detection performance, the features should be discriminative. Besides, rotation-invariant is also significant for loop closure detection. The loop closure should be detected if the robot revisits the same place wherever the laser scanner is facing.

- Feature discriminative analysis: statistics, geometry features, and range histograms are computed according to nominal distances and adjacent points on each scan line. The three modalities are all sensitive to locations and environments. A slight location perturbation of the laser scanner will cause changes in the ranging distances and scan lines. Even in the same environment, ranging histograms will be different when the sensor's location changes, which indicates the location discriminative characteristic. The plane features describing the shape and complexity of the scenes will differ with the scene changes. The intensity information is affected by the system and objects. The incident angle and materials of objects matter much on intensity values. The incident angles are influenced by the sensor's relative locations in the environment. Besides, the materials of objects might be different in diverse environments. Thus, the intensity histogram feature is also discriminative to locations and environments.

- Rotation invariance analysis: the statistics, geometry features, plane features, range histograms, and laser intensity histogram are all statistical-based feature quantities. No matter how much the laser scanner rotates, if the robot revisits the same place, the features will be similar.

3.3.3 Loop Closure Detection

After the discriminative global comprehensive descriptor $F^{(id1,id2)}$ is extracted, the descriptor ratios are computed by concatenating the specific values of multi-modalities. Then, the descriptor ratios are organized as samples to be fed into machine learning models. Methods of calculating descriptor ratios vary for different modalities. For statistics (f_s), geometry features (f_g) and plane features (f_p), the element-wise specific values are computed as:

$$f^{id1,di2} = \begin{cases} f_m^{id1} / f_m^{id2}, \\ (f_m^{id1} / f_m^{id2})^{-1} \end{cases} \tag{3.7}$$

Each pair of LiDAR scans generate two samples due to the two calculation methods in (3.7). It should be indicated that a minimum value needs to be added to the denominator in the case of NaN value. If two LiDAR scans are captured from the same environment, the values of f_m^{id1} and f_m^{id2} will be very similar, and the value of $f^{(id1,id2)}$ is close to 1.

As for range histogram and intensity histogram, correlation coefficient c_r and c_i is computed. Then, $F^{(id1,id2)}$ is computed by concatenating the specific values of each element.

$$F^{id1,id2} = f_z \oplus f_g \oplus f_p \oplus c_r \oplus c_i \tag{3.8}$$

For model training, a descriptor ratio and a binary label are combined as a training sample $\{y, F_m^{(id1,id2)}\}$. If y is 0, the scan pair is not a loop closure, whereas the value of 1 means that the scan pair is a loop closure. Then, training samples are fed into the supervised learning model.

$$y = \begin{cases} 1 & \text{positive} \\ 0 & \text{negative} \end{cases} \tag{3.9}$$

A supervised learning model will learn from training samples to identify whether a scan pair is a loop candidate or not, meanwhile, the machine learning model could also

provide a posterior probability of being detected as loops to estimate the reliability of the detection results.

3.3.4 Double-deck Loop Verification

Precision is the dominant indicator for evaluating loop closure detection results, due to wrong loop conditions might ruin the global map. Thus, to ensure precision and reject false positives, a novel double-deck loop verification strategy will be implemented. The loop verification contains two parts: cross-validation and post-verification.

Cross-validation: due to the two calculation methods of descriptor ratios in Eq.**??**, each scan pair generate two samples. Then, if one of the two samples is identified as negative, this pair of laser scans will be rejected.

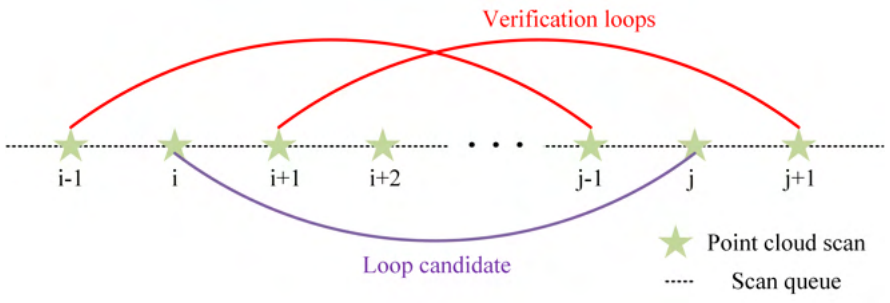


Figure 3.4: Schematic of post-verification.

Post-verification: if a laser scan pair is identified as a loop candidate, it will be verified according to time consistency and geometry consistency. As shown in Fig. 3.4, in an appropriate time buffer, scan pairs are combined. Then, all these scan pairs are detected by the machine learning model to check whether they are positive or not. If they are positive, the scan i and j are verified as a loop closure.

3.4 Experimental Results

3.4.1 Data

We train and evaluate FastLCD algorithm on the in-house datasets and Mimap in SLAM 00 dataset [157, 150], as shown in Fig. 3.5.

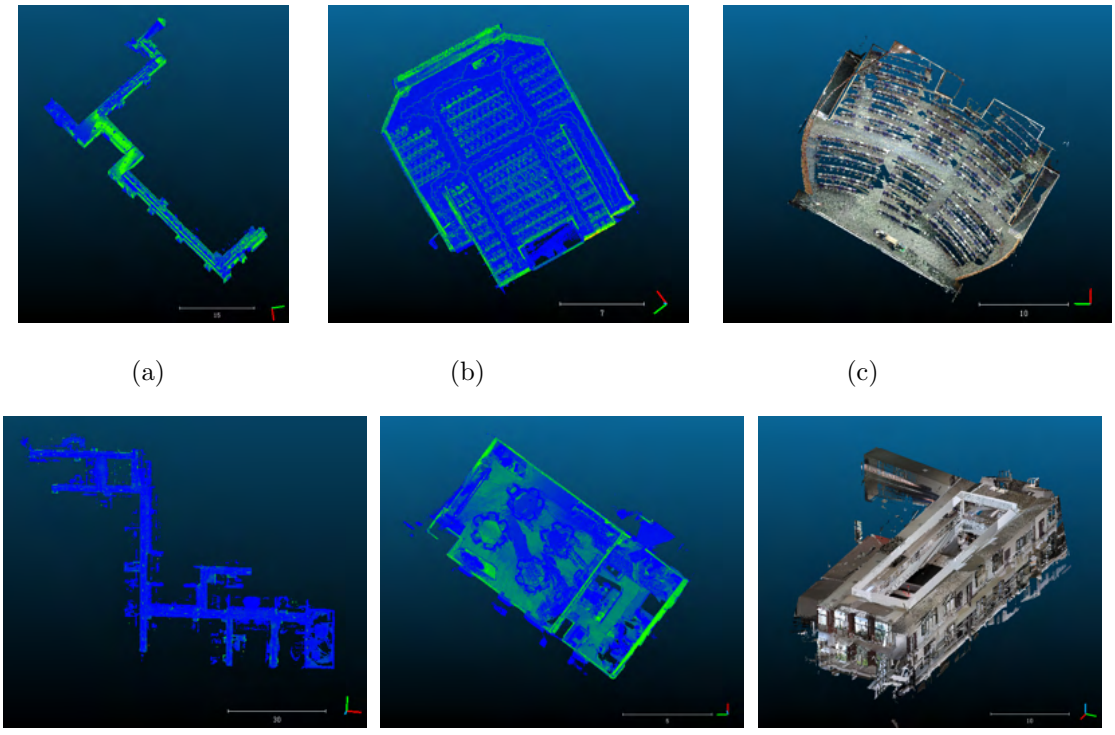
In-house datasets: (a) The corridor A dataset is captured in a long and narrow corridor with some corners; (b) The small lecture room is an irregular lecture theatre with approximately 200 seats; (c) The large lecture room is a large irregular lecture theatre with approximately 400 seats; (d) The corridor B dataset has long and narrow corridors, corners, and an open small podium; (e) The office room dataset is captured in a square office room with some desks, chairs, computers, and laboratory equipment, which is much smaller than the two lecture rooms. To validate the learning model's generalization performance, the supervised machine learning model is only trained by the corridor A dataset, then tested on the other four datasets. These datasets are the most typical scenes in indoor environments. Most indoor environments are a combination of these scenes.

The in-house datasets are captured on the Hong Kong Polytechnic University campus using a backpack mobile mapping system [46]. The laser scanner mounted on the backpack mobile platform is Velodyne's Puck LiDAR sensor.

Mimap in slam 00 datasets: the dataset is collected in a two-floor building scene, including data of individual rooms, non-enclosed loop corridors, and stairs. The point cloud scans are captured by a Velodyne Ultra puck scanner. The open-source dataset download link is: *https://www2.isprs.org/commissions/comm1/wg6/isprs-benchmark-on-multisensory-indoor-mapping-and-positioning/*

3.4.2 Supervised Model Selection

In this section, we compare the impact of different machine learning models on the algorithm's performance. Four popular machine learning models are adopted to con-



(a) (b) (c)

(d) (e) (f)

Figure 3.5: Experimental datasets. (a) corridor A dataset, (b) small lecture room dataset, (c) large lecture room dataset, (d) corridor B dataset, (e) office room dataset, (f) Mimap in slam 00 dataset.

duct the comparative experiments, including AdaBoost[49], random forest (RF)[12], support vector machine (SVM)[117], and artificial neural network (ANN)[127]. It should be specified that a backpropagation neural network (BPNN), one of the ANN models, will be used. The results are shown in Fig. 3.6 and Tab. 3.1.

3.4.3 Ablation Studies

3.4.3.1 Feature Elements

The feature ablation study results are demonstrated in Tab. 3.2. Ablation of plane features does not make noticeable impacts on results' precision but results in a slight

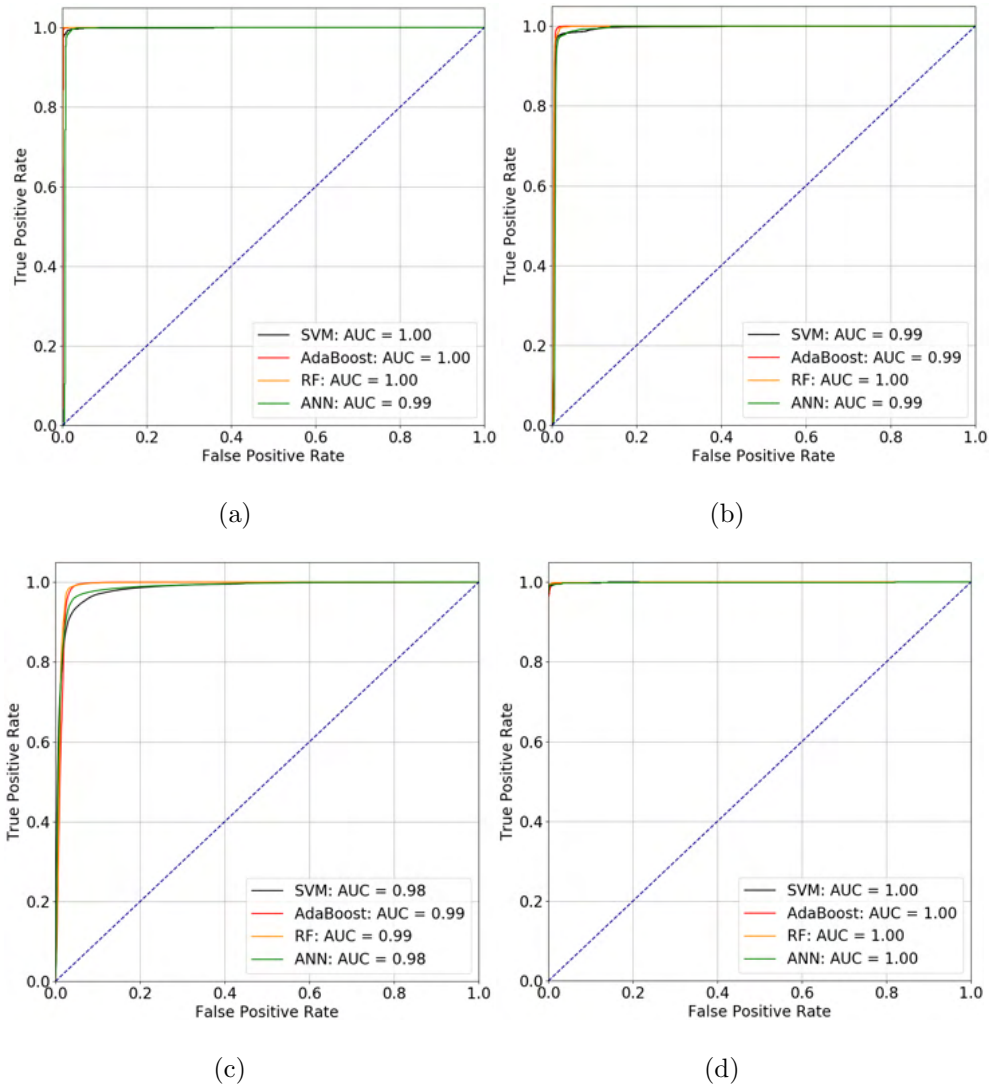


Figure 3.6: ROC curves of different learning models on the in-house datasets. (a) small lecture room dataset, (b) large lecture room dataset, (c) corridor B dataset, (d) office room dataset.

reduction of recall rate. It should be emphasized that range histogram ablation ruined the algorithm's performance. The other four types of features all influenced more on recall rate, while precision remains stable without obvious loss. Thus, the five types of feature elements affect the precision and recall rate of results to varying degrees.

Besides feature importance analysis, feature selection also relies on correlation analy-

Table 3.1: F1-scores and AUCs of machine learning models on in-house datasets

Algorithms	Small lecture room		Large lecture room		Corridor B		Office room	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC
SVM[117]	0.98	**1.00**	0.94	0.99	0.83	0.98	0.99	**1.00**
AdaBoost[49]	0.98	**1.00**	0.98	0.99	0.79	**0.99**	0.99	**1.00**
RF[12]	**0.99**	**1.00**	**0.99**	**1.00**	0.92	**0.99**	**1.00**	**1.00**
ANN[127]	0.77	0.99	0.74	0.99	**0.94**	0.98	0.96	**1.00**

Table 3.2: Result of feature ablation experiments

Feature ablation	Precision	Recall	F1
Complete	**0.98**	0.87	0.92
Statistics ablation	**0.98**	0.55	0.70
Geometry features ablation	**0.98**	0.77	0.86
Plane features ablation	**0.98**	0.86	0.92
Range histogram ablation	0.82	0.06	0.12
Intensity histogram ablation	0.97	0.85	0.91
After feature selection	**0.98**	**0.92**	**0.95**

sis results. In this experiment, the chi-square test will be used to perform correlation analysis. After feature selection, the updated descriptor shows the best performance with the precision, recall rate, and F1-score achieving 0.98, 0.92, and 0.95, respectively. Thus, the FastLCD adopts the five types of features ultimately.

3.4.3.2 Double-deck Verification Ablation Study

In this section, we will study the impact of the double-deck verification step. The ablation experiment results are demonstrated in the Tab. 3.3, in which the precision and F1-score are compared. We can see the precision of the results increases to

varying degrees after adding verification steps. However, due to the discriminative descriptor and excellent supervised learning model, even the double-deck verification step is ablated, the precision and F1-score remain relatively stable without much loss. After all, the verification step indeed increases the precision of the results more or less, which aims to make the algorithm more robust.

Table 3.3: Double-deck verification ablation experiment results

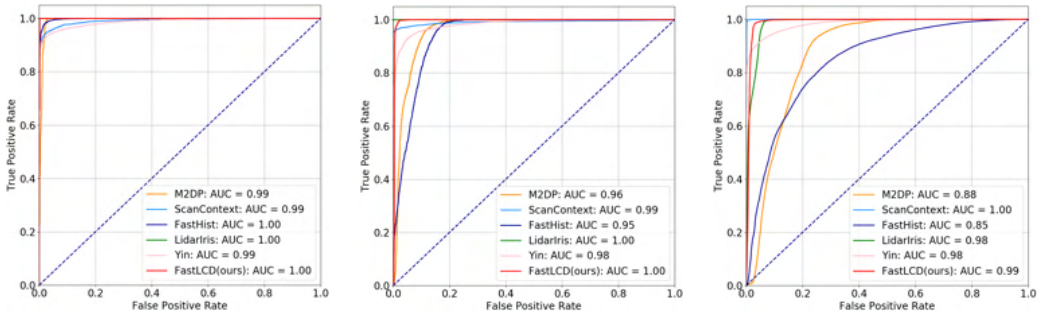
Dataset	Without double-deck verification		Double-deck verification	
	F1	Precision	F1	Precision
Small lecture room	0.99	0.98	1.00	1.00
Large lecture room	0.96	0.94	0.99	0.99
Corridor B	0.95	0.98	0.95	0.98
Office room	0.99	1.00	1.00	1.00

3.4.4 Loop Closure Detection Results

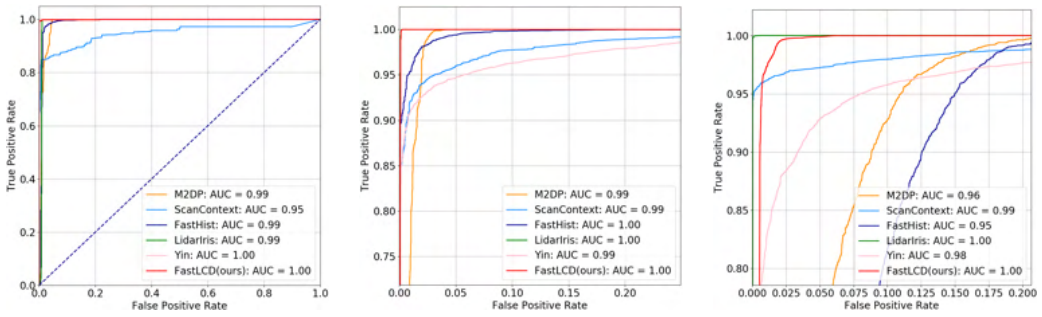
3.4.4.1 FastLCD Results

In-house datasets: we compare our FastLCD algorithm with some state-of-the-art methods on the in-house indoor datasets. Comparison results are shown in Fig. 3.7 and Tab. 3.4, where M2DP [62], FastHistogram [123], ScanContext [74],LiDAR Iris [156], Yin [167] are adopted. We can find that the comparison algorithms are difficult to achieve stable performance on all four datasets. FastLCD algorithm over-performs the state-of-the-art methods on the four datasets as the AUCs are almost equal to 1. The specific F1 scores and AUCs are demonstrated in Tab. 3.4. Though on corridor B dataset, the F1-score and AUC both rank second with 0.95 and 0.99 respectively, on the other three datasets, FastLCD all obtains superior performance. Because the experiments are trained only on the corridor A dataset, the FastLCD's superior results on the four datasets indicate the great generalization ability.

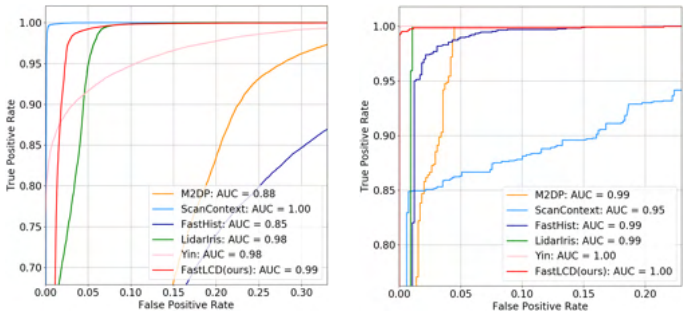
Mimap in slam 00 datasets: compared with the five algorithms in mimap in slam 00 dataset, FastLCD still outperforms stably, with 0.94 F1-score and 1.00 AUC. The Yin method also shows the same great performance as our method. However, ScanContext and LiDAR Iris are not suitable for this scene, only getting 0.56 and 0.53 F1-score respectively, which is almost useless for indoor mapping. FastLCD and Yin method both need training samples. The same training samples are used, while the training time of FastLCD is significantly shorter than that of the Yin method. The other four methods all define a distance to measure the differentiation between a pair of point clouds. According to the results in Tab. 3.4 and Tab. 3.5, FastLCD shows superior performance to the five methods on in-house datasets and mimap slam 00 datasets.



(a) Small lecture room datasets (b) Large lecture room dataset (c) Corridor B dataset



(d) Office room datset (e) Small lecture room dataset (f) Large lecture room dataset



(g) Corridor B dataset (h) Office room datset

Figure 3.7: ROC curves of FastLCD and state-of-the-art algorithms in the four in-house datasets. (a), (b), (c), and (d) are ROC curves of FastLCD and state-of-the-art algorithms in the four in-house datasets respectively. The pictures (e), (f), (g), (h) are the zoom in parts of (a), (b), (c), and (d).

Table 3.4: F1-scores and AUCs of FastLCD and state-of-the-art algorithms on in-house datasets

Algorithms	Small lecture room		Large lecture room		Corridor B		Office room	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC
M2DP[62]	0.75	0.99	0.82	0.96	0.79	0.88	0.85	0.99
ScanContext[74]	0.75	0.99	0.87	0.99	0.86	**1.00**	0.83	0.95
FastHistogram[123]	0.95	**1.00**	0.86	0.95	0.76	0.85	0.95	0.99
LiDAR Iris[156]	0.90	**1.00**	0.95	**1.00**	**0.96**	0.98	0.99	0.99
Yin[167]	0.90	0.99	0.91	0.98	0.66	0.98	0.92	1.00
FastLCD	**0.99**	**1.00**	**0.99**	**1.00**	0.95	0.99	**1.00**	**1.00**

Table 3.5: F1-scores and AUCs of FastLCD and state-of-the-art algorithms on Mimap 00 datasets

Algorithms	F1	AUC
M2DP[62]	0.91	0.97
ScanContext[74]	0.56	0.65
FastHistogram[123]	0.81	0.90
LiDAR Iris[156]	0.53	0.64
Yin[167]	**0.94**	**1.00**
FastLCD	**0.94**	**1.00**

M2DP, FastHistogram, ScanContext, and LiDAR Iris are four popular approaches based on advanced handcraft features, while the Yin method uses a siamese CNN-based network, in which it is also trained by corridor A dataset. M2DP processes 3D point clouds by projecting them onto 2D planes, which will lose some 3D information. FastHistogram and Yin methods only use ranging distance histograms as features and set thresholds experimentally. However, FastHistogram uses ranging distance histograms directly, while Yin learns deep features by a siamese CNN-based

network using histograms as input. ScanContext considers the geometry characteristics and transforms the 3D point cloud into a 2D feature image. Similarly, LiDAR Iris also generates 2D binary feature maps based on geometry information. All these methods consider part of the characteristics of the 3D point cloud with information loss. Our method integrates geometry, ranging distance and intensity information to a discriminative global comprehensive descriptor, by which our algorithm exhibits superior and stable performance on different datasets. The thresholds of the experiments are all set accordingly. We can find the comparison algorithms' performances all rely on the appropriate threshold setting, which brings in extra work and makes the algorithm not robust. Besides, overdependence on thresholds would also weaken the generalization performance of the methods. By contrast, our algorithm shows stable results using a uniform threshold, which will be analyzed in the Section 3.4.4.2.

3.4.4.2 Separability and Threshold Sensitivity Studies

If AUC equals 1, it is confirmed that there is an appropriate threshold making the model a perfect classifier. However, the defect of the comparison methods is that the thresholds may differ on different datasets. Due to the comparison methods all depending on appropriate thresholds, much effort is needed to adjust the parameters and thresholds for a relatively good result. If the algorithm is sensitive to the threshold, the threshold needs to be updated accordingly when a new dataset comes. Thus, this section is to study the threshold sensitivity of our algorithm.

As shown in Fig. 3.8, the bar graphs depict the posterior probability distribution of being recognized as loop closures. The X-axis is the posterior probability, while the Y-axis is the number of laser scan pairs to be detected. The color of the bars refers to the ground truth. The orange bars are the correct loops, while the blue bars are not loop closures.

Loop closure detection could be treated as a binary classification problem. We can

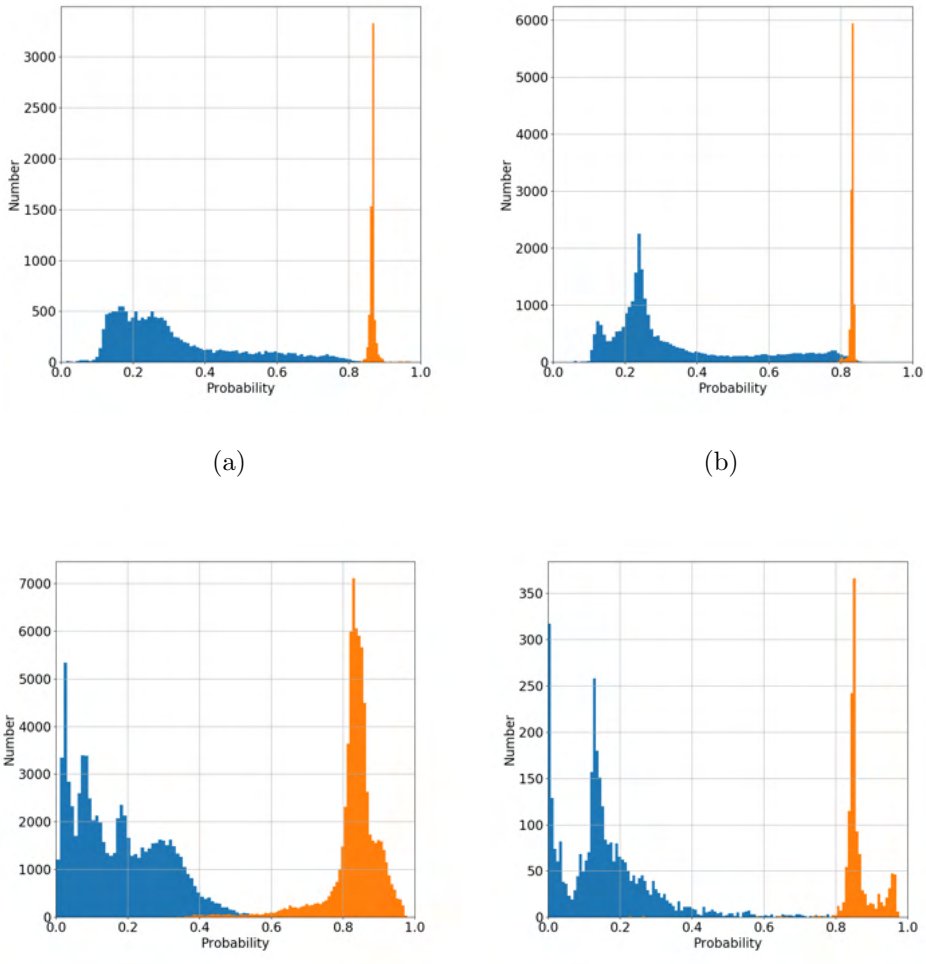


Figure 3.8: The posterior probability distribution of being recognized as loops on the in-house datasets. (a), (b). (c) and (d) are the bar graph of probability distribution on small lecture room, large lecture room, corridor B, and office room, respectively.

find that the probability distribution complies with Gaussian Mixture Model (GMM). It is obvious that the probability distributions of correct loops on the four experiments are all concentrated in $[0.8, 1]$, while the probability of negative samples is concentrated in $[0, 0.4]$. The probability distribution shows a clear trend of the U-shape canyon. It proves the superior separability and insensitivity to the threshold of the proposed algorithm. Low threshold sensitivity makes it easy to determine the threshold. According to Fig. 3.8, we can find on the four datasets, a uniform threshold 0.8 for posterior probability could be adopted.

According to the probability distribution, we can also analyze the reliability of the results. Due to the true positive probability's concentrated distribution in $[0.8, 1]$, which demonstrates that the results are highly reliable. Moreover, the posterior probability also can be used as the estimation of the loop closure detection results' reliability. Our method decides threshold by posterior probability, while the aforementioned state-of-the-art algorithms all use distance metrics. The posterior probability will be more stable than distance metrics on different datasets.

3.4.5 Time Efficiency

This experiment is conducted to compute the time efficiency of the proposed FastLCD approach. The algorithm includes three key steps: comprehensive descriptor extraction, supervised learning model training, and loop detection and verification. Model training could be conducted offline. Thus, the time efficiency experiment will focus on comprehensive descriptor extraction and loop detection and verification. We test the time efficiency on a system equipped with an Intel i7-7700 CPU with 3.6 GHz running Windows 10 x64 operating system. It should be emphasized that the code has not been optimized by time acceleration technologies, such as Compute Unified Device Architecture (CUDA) or multi-threading. The results are shown in Tab. 3.6.

As shown in Tab. 3.6, the time cost of feature extraction is stable on different datasets.

Table 3.6: Descriptor extraction time cost of FastLCD on in-house datasets

Datasets	Data amount (scan)	Time cost in total (s)	Time cost (ms/scan)
Small lecture room	9,545	790	82
Large lecture room	16,538	1,474	89
Corridor B	19,801	1,784	90
Office room	1,948	166	85

About 85 ms is needed to process each LiDAR scan, in which around 15,000 points are stored. While in Tab. 3.7, compared with feature extraction, loop detection is much more time-saving. Only about 1 ms is needed for each laser scan pair detection, which includes loop candidate determined by supervised learning model and double-deck verification. To check the time cost of the double-deck verification, we remove this step. Then, we can find the time efficiency increase sharply to 0.03 ms per detection. Compared with the state-of-the-art methods, our algorithm is the most time-efficient, with less than 100ms to detect a loop.

Time efficiency comparison experiments are also performed. The results are shown in Tab. 3.8. The experiments use open-source code of the comparison algorithms on the aforementioned equipment. High time efficiency is a decisive factor for the feasibility of the loop closure detection algorithm. All the algorithms show fast and efficient characteristics, while FastLCD costs the least time among all comparison algorithms.

Overall, considering the computation cost and practical feasibility, it is not necessary to conduct loop closure detection when every single LiDAR scan is captured. Therefore, our FastLCD algorithm can realize real-time loop closure detection in SLAM, if the data capture frequency of the scanner is set appropriately.

Table 3.7: Loop detection and verification time cost of FastLCD on in-house datasets

Datasets	Data amount (laser pair)	With verification or not	Time cost in total (s)	Time cost (ms/detection)
Small lecture room	22,104	With	25.26	1.14
		Without	0.68	0.03
Large lecture room	38,246	With	41.12	1.08
		Without	1.19	0.03
Corridor B	148,536	With	176.00	1.18
		Without	4.59	0.03
Office room	4,576	With	5.60	1.22
		Without	0.14	0.03

Table 3.8: Time cost (s) comparison of FastLCD and the state-of-the-art methods on in-house datasets

Algorithms	Small lecture room	Large lecture room	Corridor B	Office room
M2DP[62]	1097.68	1984.56	2574.13	214.28
ScanContext[74]	998.23	1730.00	2071.34	202.59
FastHistogram[123]	1587.20	2756.33	3298.16	306.66
LiDAR Iris[156]	1408.33	2439.55	2970.15	351.67
Yin[167]	1345.40	2234.17	2813.60	252.00
FastLCD	**815.26**	**1515.12**	**1960.00**	**171.60**

3.5 Discussion

According to the experiment results of the proposed FastLCD algorithm, some characteristics and limitations are summarized.

- Precision, generalization performance, and time efficiency. The proposed method

shows superior performance to some state-of-the-art algorithms, being more accurate, reliable, and robust. The AUCs of the FastLCD algorithm on the different datasets are all close to 1 indicating that it has excellent generalization performance. As for time cost, feature extraction and loop detection are both show high time efficiency. If the data capture frequency of the scanner is set appropriately, FastLCD can realize real-time loop closure detection in SLAM.

- Feature importance. The five types of features have varying degrees of influence. Range histogram plays the dominant role, while plane features impact a little. Statistics, intensity histogram, and geometry features matter much on recall rate. According to feature importance analysis and correlation analysis, we select some significant feature elements into the comprehensive descriptors.

- Separability and threshold sensitivity. The posterior probability distribution of being recognized as loops almost perfectly complies with GMM, which indicates the separability of FastLCD is excellent. The probability distributions of being detected as loops on different datasets all concentrate in the same and narrow range. It shows FastLCD is insensitive to the threshold. Thus, the loop results detected are reliable and robust.

- The proposed algorithm could detect loops in those typical indoor scenes. Moreover, because the discriminative features are learned, the method is not limited to the LCD in regular scenes. The algorithm could also be used in scenes with not flat ground or walls.

- Limitations. The FastLCD algorithm is designed for indoor environments. If the algorithm is extended to outdoor environments and large-scale scenes, some optimization and experiments should be conducted. Besides, the computation cost will grow sharply with the measuring distance and time increasing. Thus, some computation cost optimization and acceleration technologies could be adopted to make it more feasible and robust for SLAM.

3.6 Conclusion

This chapter proposes a fast and compact loop closure detection method FastLCD based on comprehensive descriptors and machine learning. to achieve reliable and precise results using 3D point clouds for indoor LiDAR mobile mapping. FastLCD algorithm extracts multi-modality features from each single 3D LiDAR scan without any transformation and projection, to map a LiDAR scan into a discriminative comprehensive descriptor. A machine learning model with a double-deck loop verification strategy is used not only to identify loop closures without prior poses but also to provide estimates of the reliability of detection results. Experiments show the algorithm can detect loop results reliably and precisely. The algorithm also shows great performance on separability, threshold insensitivity, and generalization. Besides, the high time efficiency makes it possible to realize real-time loop closure detection for SLAM in indoor mapping. In the future, as the proposed approach is only designed in indoor environments for LiDAR mobile mapping, it can be extended to outdoor and large-scale environments. Furthermore, a deep learning model could be used to uncover some hidden features.

Chapter 4

DeLightLCD: A Deep and Lightweight Network for LCD in LiDAR SLAM

Loop closure detection is a critical yet still open technique to enhance the performance of Simultaneous Localization and Mapping (SLAM). In this chapter, a very deep and lightweight neural network DeLightLCD is proposed to enable real-time loop closure detection in large-scale environments. The raw 3D point clouds are mapped into 2D depth image spaces as the input of the network. The architecture of the network contains two key modules: a feature extraction module and a feature difference module. A very deep but lightweight feature extraction network is designed to extract high-dimensional and discriminative features. Depth-wise separable convolution and batch normalization are utilized to ensure the network is lightweight and trainable. The feature difference module enhanced by the dual attention technique generates feature difference maps to identify the difference between pairs of LiDAR scans. The proposed algorithm has been tested on the KITTI odometry datasets and Ford campus datasets. The experimental results demonstrate that the proposed algorithm

outperforms the existing state-of-the-art methods. Although the model was trained only on the KITTI dataset, it also demonstrated superior performance on the Ford campus dataset. In particular, the proposed algorithm is much more lightweight than the state-of-the-art methods.

4.1 Introduction

In recent years, methods concerning the loop closure detection (LCD) tasks have been extensively examined from the robotics and autonomous driving viewpoints within the scope of SLAM applications. LCD is used to check whether the place has been previously explored, i.e. to distinguish whether a pair of LiDAR scans are captured in the same environment. Thus, LCD is also recognized as a problem of place recognition or instance retrieval, as shown in Fig. 4.1. KITTI odometry datasets [54, 53] are used in the figure. By providing control information for back-end optimization, LCD plays an important role in addressing the issue of eliminating cumulative errors in SLAM.

Currently, solutions towards the LCD problem focus more on whether the two scans of point clouds have sufficiently high similarity to enable a judgment based on the distance and a predefined threshold. The deep learning model always plays a role as a feature encoder to obtain a global descriptor. Distance or metric learning will then normally be used to measure the similarity between a pair of laser scans. Instead, proposed here, is an end-to-end deep neural network for LCD to directly give a binary result.

In [145], researchers argued that the most straightforward way to improve the performance of deep neural networks is to increase their size, including depth and breadth. Increasing the depth of the network is more feasible and cost-effective than increasing the breadth. Nevertheless, deeper networks also bring along such typical problems as excessive parameter count, gradient vanishing, and overfitting. Proposed in this

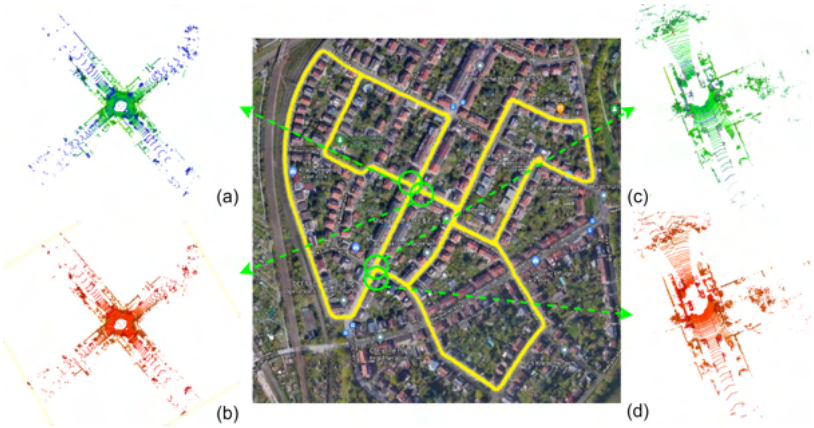


Figure 4.1: Sequence point clouds diagram (*trajectory: KITTI odometry 00*). Point clouds shown in (a) and (b) are loop closures because they are captured in the same environment, as do those in (c) and (d), whereas the point clouds in (a) and (c) are not loop closures. Neither are those in (b) and (d).

chapter, to extract high-dimensional and more abstract features is a feature extraction module with a deeper network than the state-of-the-art networks. Depth-wise separable convolution (DSC) and batch normalization (BN) are utilized to suppress those problems caused by the deeper network. The contributions of this chapter are:

- The proposal of an super lightweight neural network for LiDAR-based LCD without using prior pose information. The proposed DeLightLCD offers a performance that provides both highly efficient and reliable loop closure detection.
- The proposed network structure is deep and fully exploits the geometry information of the 3D point clouds. Only leveraging (x, y, z) as the input, the network not only achieves superior LCD performance comparing with the state-of-the-art methods but also is much more lightweight in parameter amounts.
- The proposed feature extraction module integrates with DSC and BN to ensure that the network is deep and lightweight while aiming at extracting discriminative point cloud features.

- The feature difference module with a dual attention mechanism is proposed to generate the feature difference map to measure the difference between a pair of LiDAR scans, to determine if they are loop closures.

- The experimental results based on the KITTI [54] odometry benchmark and the Ford campus dataset [115] show that the proposed method outperforms the state-of-the-art LiDAR-based LCD methods.

4.2 Related Works

Unlike traditional approaches that use artificially designed features and metrics, deep learning-based approaches deduce, or learn features and metrics from training data. PointNetVLAD [148] combined PointNet [19] to learn about local features and NetVLAD [6] architecture to build global descriptors from 3D point clouds. Since then, many methods have followed the structure of PointNetVLAD, learning local features, encoding local features into global features, and measuring similarities between global descriptors. The architecture has also become the standard workflow for subsequent deep learning-based methods, such as SeqLPD [100], SOE-Net [162], LPD-Net [101], and PCAN [173], etc. OverlapNet [26] transferred the raw 3D point clouds into 2D images and adopted a Siamese Network as the backbone and defined overlap to compute the yaw angle. It is worth mentioning that some cutting-edge technologies, such as the attention mechanism [162] and the transformer [176] have become widely used in LCD.

Current LiDAR loop closure detection approaches based on deep learning always adopt relatively shallow networks for feature extraction. In image-based networks, increasing network depth is a straightforward and effective way to achieve superior

performance, such as ResNet [61] and VGG16 [136]. Thus, this study intended to design a very deep neural network to extract highly-dimensional and advanced features. Then, we proposed a feature difference network enhanced by the dual attention technique to identify the degree of similarity between the LiDAR scans. The proposed DeLightLCD approach can detect loop closures without prior pose knowledge and predefined thresholds, which guarantee the robustness of the delivered method.

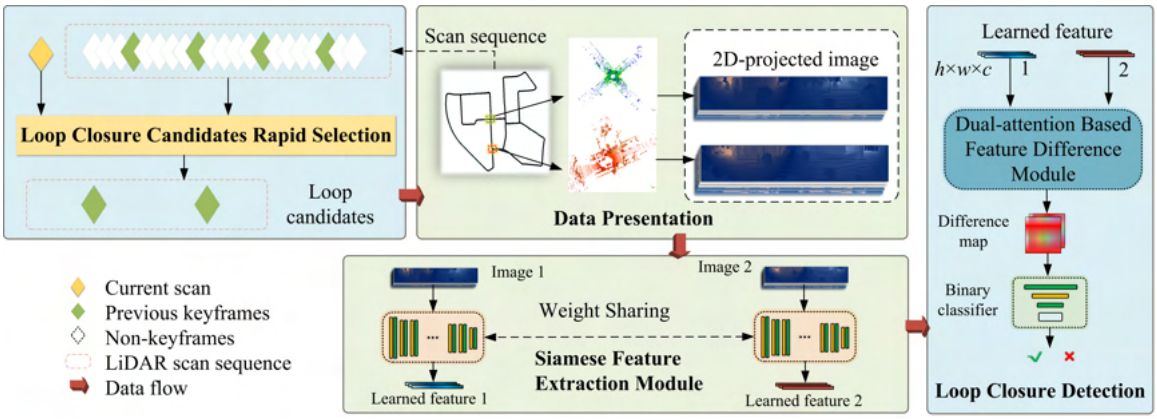


Figure 4.2: Pipeline overview of the proposed DeLightLCD approach

4.3 Methodology

This chapter addresses LCD as a point cloud retrieval problem. The proposed DeLightLCD method uses an end-to-end network to detect loop closures without prior pose information. The raw point clouds are transformed into 2D depth images and encoded into feature spaces. The architecture of DeLightLCD is shown in Fig. 4.2. DeLightLCD consists of two core modules: the feature extraction module and the feature difference module. In the feature extraction module, DSC and BN are utilized to ensure that the proposed very deep network is lightweight and computationally efficient. The dual attention technique composed of channel-wise attention and point-wise attention enables a more outstanding and obvious feature difference.

The efficient and accurate LCD approach greatly facilitates localization and mapping tasks in robotics and automatic drive applications.

4.3.1 Data Representation

Cyclic projections of LiDAR scans are used as input, which is often used to improve computational efficiency [11]. Many algorithms utilize the original 3D point cloud data directly as the input to the network [173, 148], while the projected 2D images can bypass the problem of permutation invariance. We project the 3D point cloud $\mathcal{P} \in \mathbb{R}^{n_k \times 3}$ into 2D image plane $\mathcal{V} \in \mathbb{R}^{H \times W}$ [26] regarding cyclic coordinates using the projection function $\phi(\mathcal{P}_k)$: $\mathbb{R}^3 \mapsto \mathbb{R}^2$, where n_k, h, and w are the number of points, the height and the width of resulting images. The laser scan $\mathcal{P}_k$ is the keyframe at the time step $k \in \mathbb{Z}^+$ in the point cloud query. Each point cloud $\mathcal{P}_k = (x, y, z)$ is transformed to cyclic coordinates (u, v), according to:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2}[1 - \arctan(y, x)\pi^{-1}]W \\ [1 - (\arcsin(zr_k^{-1}) + |\mathrm{f_{down}}|)\mathrm{f}^{-1})]H \end{pmatrix}, \tag{4.1}$$

where $r_k = \|\mathcal{P}_k\|_2$ is the range, $\mathrm{f} = \mathrm{f_{up}} + |\mathrm{f_{down}}|$ is the vertical field-of-view of the laser scanner. h represents the number of scan lines, whereas the w direction contains information on each single scan line. Thus, h and w are set as 64 and 900.

4.3.2 Feature Extraction Module

Proposed in this study is a deep and lightweight feature extraction network with shared weights to learn high-dimensional and discriminative features from each LiDAR scan. The point coordinates (x, y, z) are leveraged only from each LiDAR scan to generate projected maps $\mathcal{C}$.

The architecture of the feature extraction network is shown in Tab. 4.1. The network designed for poing cloud feature extraction is a convolutional neural network (CNN) composed of 17 convolutional layers with batch normalization and 3 average pooling

layers. Inspired by the studies in [145], increasing network depth and breadth is a straightforward and effective way to improve network performance. Solid evidence is yielded in [60] that increasing depth is a prior condition for improving accuracy. If the network depth is increased reasonably, more complex, abstract, and high-dimensional features are likely to be learned. Thus, to be designed is a feature extraction network with 17 convolutional layers. The network will thus be deeper than the state-of-the-art LCD networks. However, there are trade-offs regarding depth, width, and

Table 4.1: Layers of feature extraction module network architecture

	Operator	Filters	Size	Output Shape
Vertical Encoder	DSC+BN	8	(3,13)	(62,888,8)
	DSC+BN	8	(3,13)	(60,876,8)
	AP	-	(2,2)	(30,438,8)
	DSC+BN	8	(3,13)	(28,426,8)
	DSC+BN	8	(3,13)	(26,414,8)
	AP	-	(2,1)	(13,414,8)
	DSC+BN	16	(3,13)	(11,402,16)
	DSC+BN	16	(2,13)	(10,390,16)
	AP	-	(2,1)	(5,390,16)
	DSC+BN	16	(3,11)	(3,380,16)
	DSC+BN	16	(3,11)	(1,370,16)
Horizontal Encoder	DSC+BN	32	(1,11)	(1,360,32)
	DSC+BN	32	(1,11)	(1,350,32)
	DSC+BN	32	(1,9)	(1,342,32)
	DSC+BN	32	(1,9)	(1,334,32)
	DSC+BN	32	(1,9)	(1,326,32)
	DSC+BN	32	(1,9)	(1,318,32)
	DSC+BN	32	(1,7)	(1,312,32)
	DSC+BN	32	(1,7)	(1,306,32)
	DSC+BN	32	(1,7)	(1,300,32)

filter sizes. When the increased depth becomes saturated, the model performance may show no improvement and even degradation. Increasing depth introduces new

problems, such as an excessive parameter count and gradient vanishing. Any of those particular problems may lead to non-convergence and the network untrainable. In the proposed feature extraction module, DSC and BN have been adopted to restrain these problems. DSC is a mechanism that has been widely recognized because it can significantly reduce the number of parameters, especially after the success of the lightweight neural networks Xception [30] and MobileNets [66]. This mechanism is based on an assumption that each channel is highly auto-correlative across space, while different channels may not be highly correlated with the other [29]. In the proposed approach, only (x, y, z) information is leveraged to enable the learning of features. Point clouds may exhibit different distribution characteristics in the x, y, and z spaces. A DSC consists of a depth-wise convolution and a point-wise convolution performed subsequently. The former refers to a convolution performed separately on each channel, while the latter refers to a 1×1 convolution over channel space. They play different roles in extracting new features: depth-wise convolution is used for obtaining spatial correlations whereas point-wise convolution could capture channel-wise correlations [59]. Thus, the DSC is not only more efficient than a standard convolution which maps channel correlations and spatial correlations simultaneously [30], but also enables the collection of features from multiple domains [59].

In addition, when networks go deeper, gradient propagation through a deep stack of layers is more difficult, hence making the related network untrainable. Originally, BN was proposed to reduce internal covariate shift, defined as "the change in the distribution of network activations due to the change in network parameters during training" [69]. This can normalize input data even allowing for means and variance changes over time, during the training stage. BN can also both reduce the dependence on gradients and benefit the gradient flow. Thus, BN is utilized to control the problems brought by the increasing depth of the network.

The feature extraction network contains two feature encoders in two directions: a vertical encoder and a horizontal encoder. The vertical encoder aims at extracting

features from among different scan lines, while the horizontal encoder focuses on the information on each scan line. Average pooling (AP) is used to reduce dimensions in the height direction. The vertical encoder includes 4 groups, two DSC+BN convolutional layers in all the four groups, and AP in the first three groups. In the horizontal encoder, the three groups are all composed of DSC+BN convolutional layers. The dimension change from input image map $\mathcal{V}$ to output features $\mathcal{F}$ is $\mathbb{R}^{H\times W\times C} \mapsto \mathbb{R}^{h\times w\times c}$. In the proposed network, h, w, and c are set as 1, 300, and 32, respectively.

4.3.3 Feature Difference Module

Currently, most LiDAR-based deep learning LCD algorithms extract the global descriptor from each LiDAR scan, generated by local features extracted in advance. A distance between two global descriptors will then be computed, and predefined threshold or metric learning will be used [148]. The proposed feature difference module includes a feature difference network to identify loop closures by building feature difference maps $\mathcal{M}$. A feature difference map integrated with dual attention is fed into a fully connected network to detect loop closures. The process of a feature difference map calculation is shown in Fig. 4.3.

This module contains three key parts: feature broadcasting, feature difference, and dual attention. The difference broadcasting features $\mathcal{F}_d$ are generated from $\mathcal{F} \in \mathbb{R}^{h\times w\times c} \mapsto \mathcal{F}_d \in \mathbb{R}^{w\times w\times c}$. It should be noted that h, w and c, are set as 1, 300, and 32, respectively.

Dual attention contains point-wise attention A_p and channel-wise attention A_c. Point-wise attention is designed to encode significant inter-spatial relationships into feature difference maps, while channel-wise attention expresses the importance of inter-channel relationships. Inspiration generated by the work of [67], led in this study, to the design of a lightweight network to calculate dual attention. Channel-wise attention is performed by global average pooling G_c and fully connected layers on each

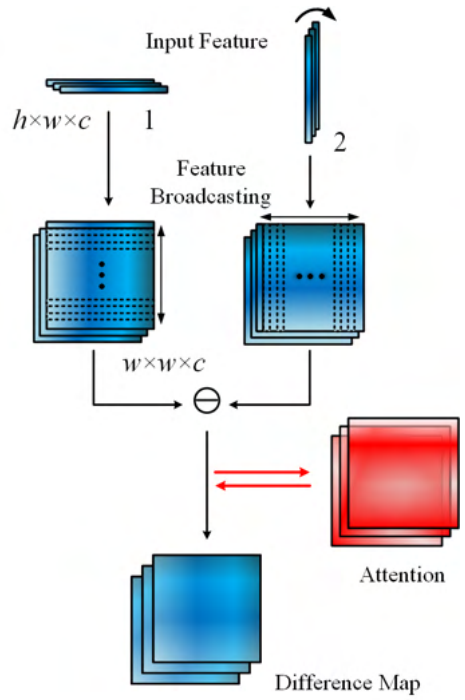


Figure 4.3: The architecture of feature difference module

channel across space, whereas point-wise attention is conducted by global average pooling G_p and fully connected layers on every point across the depth.

$$G_p(i,j) = \frac{1}{c}\sum_{u=1}^{c}\mathcal{F}_d(i,j,u), \tag{4.2}$$

$$G_c(u) = \frac{1}{w \times w}\sum_{i=1}^{w}\sum_{j=1}^{w}\mathcal{F}_d(i,j,u), \tag{4.3}$$

where $G_p(i,j)$ refer to global average pooling in the space domain (i,j), while $G_c(u)$ denotes global average pooling in the channel domain u.

$$A_p(i,j) = sigmoid(\sigma(G_p(i,j))), \tag{4.4}$$

$$A_c(u) = sigmoid(\sigma(G_c(u))), \tag{4.5}$$

where $sigmiod(\cdot)$ refers to the sigmoid function, and $\sigma(\cdot)$ denotes the fully connected layers. The feature difference map $\mathcal{M} \in \mathbb{R}^{w\times w\times c}$ is then generated. The map not

only contains differential information extracted from a pair of LiDAR scans, but also the weights of each point and each channel.

$$\mathcal{M} = \mathcal{F}_d \odot \delta(A_c \otimes A_p), \tag{4.6}$$

where $\odot$ and $\otimes$ refers to element-wise multiplication and Kronecker product, respectively. δ denotes the reshaping function to transform the tensor into $\mathbb{R}^{w \times w \times c}$ space.

Based on the dual attention technique, the difference map could adaptively recalibrate point-wise and channel-wise feature responses by explicitly building the weight matrix across the space domain and the channel domain.

A simple convolutional neural network is used as a binary classifier to detect whether the LiDAR scan pair constructs a loop. The network contains a 3-layer fully connected network and a sigmoid layer. The network architecture is shown in Fig. 4.4. The input of the network is a difference map generated by the feature difference module with the dimension of $300 \times 300 \times 32$. The probabilities output by this network indicates whether the LiDAR pair is a loop.

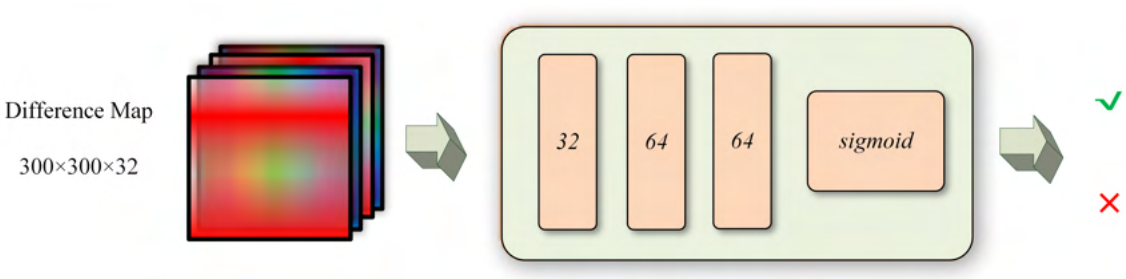


Figure 4.4: The network architecture of the binary classifier

4.3.4 Loss Functions

The developed DeLightLCD network was trained end-to-end to detect loop closures from 3D LiDAR scans using a binary cross entropy loss function $\mathcal{L}_{(\mathcal{P}_1,\mathcal{P}_2)}$, where

$(\mathcal{P}_1, \mathcal{P}_2)$ is a pair of LiDAR scans.

$$\mathcal{L}_{(\mathcal{P}_1,\mathcal{P}_2)} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p(y_i)) + (1-y_i)\log(1-p(y_i)), \tag{4.7}$$

where N is the number of training samples, and y_i binary denotes the ground truth. $p(y_i)$ refers to the probability that the pair of LiDAR scans $(\mathcal{P}_1, \mathcal{P}_2)$ form a loop closure.

The training process of DeLightLCD is presented in *Algorithm* 1.

4.4 Experiments

4.4.1 Datasets

The KITTI odometry benchmark [54, 53] and Ford campus datasets [115] were used to evaluate the proposed DeLightLCD approach. The two datasets both were collected from large-scale outdoor environments.

- KITTI odometry data were captured by a Velodyne HDL-64E laser scanner. The KITTI sequence 02 was used to validate the algorithm, while other sequences were used for training. If the distance between a pair of LiDAR scans is less than 2m, they were considered as positive samples, whereas negative samples are represented by distances greater than 10m. The rigorous training sample thresholds ensured the correctness of training samples. The threshold was set following the work in [148]. The open-source dataset download link is: *https://www.cvlibs.net/datasets/kitti/*

- Tests were also made using the Ford campus 02 datasets. The data were also captured by Velodyne 64-channel laser scanners in outdoor large-scale environments. It should be emphasized that our network was never trained on

Algorithm 1 DeLightLCD Training Workflow

Input: a pair of laser scans $\mathcal{P}_m$, $\mathcal{P}_n$

Output: a trained model

Spherical projection: Equation (4.1):

$\mathcal{C}_m$: size$(N, 64, 900, 3) \leftarrow \mathcal{P}_m$;

$\mathcal{C}_n$: size$(N, 64, 900, 3) \leftarrow \mathcal{P}_n$;

for epoch in enumerate maximum epoch **do**

Siamese feature extraction module:

$\mathcal{F}_m$: size$(N, 1, 300, 32) \leftarrow \mathcal{C}_m$;

$\mathcal{F}_n$: size$(N, 1, 300, 32) \leftarrow \mathcal{C}_n$;

Feature difference module:

$\mathcal{M}$: size$(N, 300, 300, 32) \leftarrow \mathcal{F}_m, \mathcal{F}_n$;

Binary classifier module:

Predict result: $p(y_i) \leftarrow \mathcal{M}$;

Compute loss: Equation (4.7);

Backpropagation;

Update the network parameters;

end for

return a trained model;

the Ford campus dataset. Thus, the generalization performance of the proposed method can be evaluated. The open-source dataset download link is: *http://robots.engin.umich.edu/SoftwareData/Ford*

In the experiments, the results are evaluated by three indices: precision, AUC of PR-curve (AUC), and F1-score (F1). F1 is a comprehensive evaluation index calculated by precision and recall rate. Each LCD algorithm aims at achieving higher precision and recall rate synchronously, although the two indices are often in an inverse relationship. For LCD tasks, precision plays a primary role because if the wrong loop results are adopted, a mapping disaster may result. Thus, an attempt was made to seek a trade-off between the two indices while maintaining the highest precision. In addition, The experimental setup is presented in Tab. 4.2.

Table 4.2: Parameter configuration of experimental setup

Parameters	Description	Configuration
h	Height of 2D images	64
w	Width of 2D images	900
σ_p	Threshold of positive samples	2 m
σ_n	Threshold of negative samples	10 m
e	Epochs	20
l	Learning rate	0.0001
f	Factor of learning rate reduced	0.1
d	Decay of every epoch	1e-6
m	Momentum	0.9

4.4.2 Loop Closure Detection

The proposed DeLightLCD approach was compared with the three state-of-the-art methods: PointNetVLAD [148], LPD-Net [101], and OverlapNet [26]. These three

LiDAR-based LCD algorithms are well-established in the field of place recognition. PointNetVLAD combined PointNet and NetVLAD with a fully connected network to extract global features. Metric learning with the novel lazy triplet and lazy quadruplet losses is utilized to learn discriminative features. Similarly, LPD-Net also used PointNet as the backbone for feature extraction and metric learning with lazy quadruplet loss. OverlapNet algorithm utilizes various information and defines an overlap estimation between a pair of point cloud data. The input to both PointNetVLAD and LPD-Net are raw point clouds, while those for OverlapNet are 2D depth images containing multiple cues: coordinates, normal, measuring distances, intensity, and semantic information. For the proposed DeLightLCD algorithm, raw point clouds were mapped into 2D depth images by cyclic projection.

Table 4.3: Comparison with state-of-the-art methods LCD results

Dataset	Algorithm	Precision	AUC	F1
KITTI	PointNetVLAD[148]	0.81	0.83	0.79
	LPD-Net[101]	0.83	0.83	0.85
	OverlapNet[26]	0.94	0.86	0.87
	DeLightLCD (ours)	0.96	0.99	0.90
Ford campus	PointNetVLAD[148]	0.67	0.64	0.72
	LPD-Net[101]	0.74	0.77	0.77
	OverlapNet[26]	0.85	0.88	0.83
	DeLightLCD (ours)	0.88	0.99	0.93

The results of the comparative experiments are demonstrated in Tab. 4.3. The LCD results on KITTI 02 sequence is depicted in Fig. 4.5. The same training and test samples were used on the both datasets. It should be emphasized that the model was only trained on the KITTI datasets. It was found that DeLightLCD can effectively

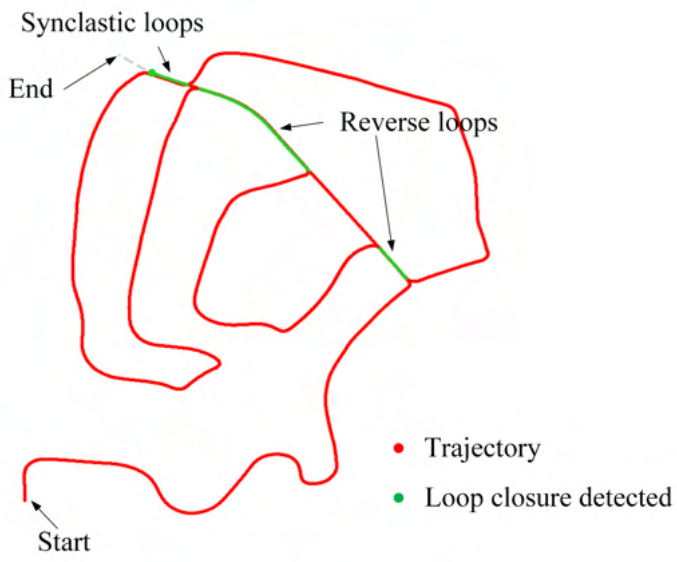


Figure 4.5: The LCD results of KITTI 02 dataset

detect loop closures surpassing the performance of state-of-the-art methods based on the two datasets. The approach used achieves a very high AUC value of 0.99 in both cases. DeLightLCD also surpasses the precision and F1-score of other algorithms.

4.4.3 Impact of Network Depth

According to the previous study [145], increasing the depth of the network is a more feasible and cost-effective way of improving the performance of deep neural networks. Therefore, a very deep neural network DeLightLCD was designed. In this section, the experimental results of feature extraction networks with different depths are compared. According to the architecture of the proposed feature extraction network in Tab. 4.1, the layers can be grouped into DSC+BN or DSC+BN+AP. This network depth ablation study was conducted by cutting off groups in the network. It should be indicated that the experiments were evaluated on the KITTI sequence 02 datasets. The experimental results are shown in Fig.4.6. It was found that all three indices increase sharply from 6 layers to 8 layers, with AUC, precision, and F1 growing from

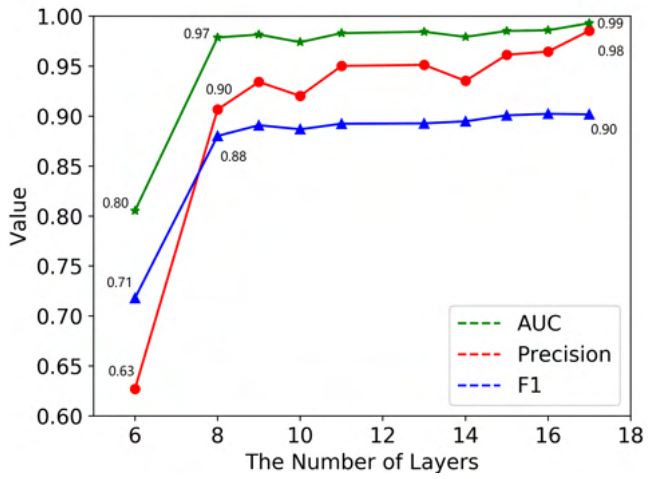


Figure 4.6: The impact of the number of layers on the results

0.81, 0.63, and 0.72 to 0.98, 0.91, and 0.88, respectively. AUC and F1 will then show relative stability but still slight growth trends at around 0.98 and 0.90 respectively, whereas precision keeps growing to 0.96 on the 17-layer network.

To explore the reasons, the convolutional layers were found to be responsible for advanced features extraction in the proposed feature extraction network. When employing deeper networks, the algorithms could extract higher dimensional and more complex features for LCD tasks, while the shallow network performs weakly in this respect. The end-to-end network is trained to extract discriminative features with strong descriptive power, which forms the basic premise for the LCD task.

However, the network cannot be deepened indefinitely. Various issues are significantly highlighted as the network depth increases. The major problems include an excessive parameter count and gradient vanishing. The problems may be mitigated and ameliorated utilizing DSC and BN techniques, but as the network depth continues to deepen, the techniques will lose efficacy. According to results shown in Fig.4.6, the proposed 17-layer network can obtain satisfactory LCD results. Continuing to deepen the network does not lead to a significant optimization of the results. The

state-of-the-art methods, PointNetVLAD, LPD-Net, and OverlapNet have 5, 9, and 11 layers, respectively, in the feature extraction network. The 17-layer feature extraction network of the proposed DeLightLCD is deeper, but yet much more lightweight. The comparison of parameter count is shown in Sec. 4.4.5.

4.4.4 Ablation Study of the Dual Attention Technique

Table 4.4: Ablation study of the dual attention technique

Dataset	Algorithm	Precision	AUC	F1
KITTI	With	0.96	0.99	0.90
	Without	0.95	0.99	0.90
Ford campus	With	0.88	0.99	0.93
	Without	0.69	0.97	0.80

In this section, the effect of the dual attention technique in feature difference layers is investigated. In DeLightLCD, the dual attention module calculates point-wise attention and channel-wise attention. It is argued that the channel-wise attention could explicitly model the interdependence between channels, and that point-wise attention could identify interdependence in the space domain. This strategy may improve the robustness of the algorithm.

The comparison results are shown in Tab. 4.4. For both the KITTI dataset and the Ford campus dataset, the results of the complete DeLightLCD are superior to those of the methods where there is no dual attention mechanism. The dual attention module does have different influences on the two datasets. On the KITTI dataset, the precision of results reduces just slightly after module ablation, while the three indices all reduce obviously for the Ford campus dataset. The precision is even reduced from 0.88 to 0.69. Thus, the dual attention module plays a significant role in DeLightLCD.

4.4.5 Parameters and Computation Cost

Table 4.5: Comparison of parameter count and time cost

Algorithm	Parameters	Time cost (ms/detection)
PointNetVLAD[148]	2 M	24.88
LPD-Net[101]	2 M	22.94
OverlapNet[26]	921 K	8.88
DeLightLCD (ours)	89 K	1.69

In this section, the parameter and time cost of the proposed DeLightLCD are compared with the three state-of-the-art methods. The time efficiency was tested on a computer equipped with the Intel Xeon E5-2660 v4 CPU and two NVIDIA Tesla P100 16GB GPUs. AAs shown in Tab.4.5, It is found that the parameters vary greatly among these methods. For PointNetVLAD and LPD-net, the numbers are similar, at 2 M parameters [101]. The proposed DeLightLCD is much more lightweight, involving only 89 K parameters. The number of parameters with PointNetVLAD and LPD-net is about 20 times that of DeLightLCD, therefore. Fewer parameters lead to a more lightweight model and a faster calculation speed. The time penalty for DeLightLCD is only 1.69 ms for each detection, which is enough to meet real-time needs.

4.5 Limitations

The DeLightLCD achieves precise and highly efficient result. However, in the application of the DeLightLCD method, some practical problems still exist. Although DeLightLCD achieves a very high time-efficiency level, when measurement distance keeps increasing in large-scale environments, the time cost will also increase exponentially. Thus, the LCD should be facilitated with a loop candidate fast search strategy

to control the time cost. Besides, limited by the projection method, DeLightLCD is dedicated to detecting loops using multi-line laser scanners and faces the problems of data loss to some extent.

4.6 Conclusions

Proposed is an end-to-end LCD approach, DelightLCD, which can perform real-time and reliable LCD in large-scale environments. The proposed approach consists of a very deep and lightweight feature extraction module and a feature difference module. The feature extraction module aims at extracting discriminative and high-dimensional features. The feature difference module generates feature difference maps using the dual attention technique. The approach was evaluated on the open-source datasets and outperforms state-of-the-art LCD methods. It can conduct LCD without prior pose knowledge and predefined thresholds. It should be emphasized also, that the proposed DeLightLCD is superior on parameter count and real-time detection speed. However, the input to DeLightLCD consists of 2D depth images. A network using raw point clouds as input will be the target of future research.

Chapter 5

DeLightLCD++: An Improved and Flexible LCD Network for LiDAR SLAM

5.1 Introduction

In applications of DeLightLCD, some practical problems are discovered. (1) The 3D point cloud projection method faces data loss problems inevitably. Due to the dimension of 2D projected images being set as 64×900, it addresses the multi-line laser scanners with 64 channels. If the scan lines change, the dimension of the data must change accordingly. Then, the pre-trained model will lose effect, due to the dimension of input data is not the same. Thus, we need to optimize the data representation method to make the algorithm more flexible. (2) When DeLightLCD is performed in a large-scale environment, only keyframes extracted by the front-end of SLAM will be detected. However, with measurement distance increasing, the time cost will also increase exponentially. Although the time cost of every single detection is low, it will also bring unacceptable computational and time costs when the measurement

distance keeps increasing.

To address the aforementioned problems, we propose an optimized approach named DeLightLCD++. A novel data representation method is utilized to make the model invariant to data changes. Practical loop candidates fast search method is adopted, enabling the high time efficiency of the method. Besides, a weighted and hierarchy back-end graph optimization method is proposed to control the drift error of front-end odometry. The main contributions of this chapter are:

- The proposal of an improved data representation method for 3D point cloud projection to 2D images. The proposed data representation utilizes measurement distance and angle information to encode the 3D point clouds to 2D images. The representation method is flexible and invariant to sensors and environment changes.

- The optimized feature extraction module aggregate the information in the horizontal direction, which makes the feature extracted is invariant to rotation. Thus, the feature different module will be simplified without feature broadcasting.

- A practical and efficient loop candidate fast search strategy is utilized to control the time cost in case the measurement distance keeps increasing.

- The experimental results based on the KITTI [54] odometry benchmark, the Ford campus dataset [115], and the in-house datasets show that the proposed method outperforms the state-of-the-art LiDAR-based LCD and back-end optimization methods.

5.2 Related Works

In Sec. 3.2 and 4.2, point cloud data are represented by feature descriptors and spherical projecting to 2D image spaces. Generally, the advantages of data representation of raw point cloud data are threefold: dimension reduction, feature mining, and regularization of unordered point clouds. Besides, using raw point cloud data as the input of the deep learning model brings problems of high computation cost and permutation invariance. In this section, a brief review of data presentation for deep learning models of point clouds will be given. All the representation ideas aim at transforming unordered point cloud data into ordered data.

Volumetric representation. The volumetric representation methods transform the point cloud to 3D grids/voxels, where the size of each grid/voxel is fixed. The points or features filled in each grid/voxel are designed artificially or learned by deep learning models. PointGrid [84], VoxelNet [175], SEGCloud [146] and VoxNet [104] voxelized point cloud data as the input of neural networks. However, the problem of volumetric representation is information loss. The small voxel size ensures the high spatial resolution, while it also brings high computation costs. A balance between the spatial resolution and computation cost should be reached for volumetric representation.

Tree representation. To address the unbalanced problems of volumetric representation, adapted resolution methods that utilize tree-based data structures. The methods proposed in [170, 77] use kd-tree structure. The octree structure is adopted in [122, 52, 86]. Tree-based methods divide the point cloud into a series of unbalanced trees based on point densities.

2D image representation. 3D point clouds could also be transformed into the 2D space by projection. Common projection methods include spherical projection [26, 11] and multi-views projection [143, 98]. Based on the image representation, the permutation invariance problem is circumvented. The problem of 2D image presentation is information loss and pointwise identification.

Graph representation. Point clouds also can be represented as graphs in the spatial domain or spectral domain. Graph-convolution methods proposed in [23, 63] transform point clouds into spatial domain. Spectral-domain graph-convolution methods use Laplacian Spectrum for spectral filtering on graphs [13, 39].

Point cloud representation. Point cloud representation methods use 3D point clouds directly. The pioneering works is PointNet [19]. Since than, many point cloud representation methods are developed motivated by PointNet, like PointNet++ [119], PointCNN [94], and [68, 7]. Point cloud representation suffers from the problem of permutation invariance and high computation cost. However, compared with other methods, direct point cloud representation has no information loss.

In this chapter, a novel distance-angel representation is proposed to project 3D point clouds to 2D image space. The proposed data representation utilizes measurement distance and angle information to encode the 3D point clouds. The method is flexible and invariant to scan lines and environment changes, which enables DeLightLCD++ flexible in either indoor small-scale environments or outdoor large-scale environments.

5.3 Methodology

In this chapter, an improved DeLightLCD approach, i.e. DeLightLCD++ is proposed to addresses the aforementioned problem. The proposed DeLightLCD++ method optimizes the data presentation method and utilizes a novel projection method from 3D point clouds to 2D images. The new data representation method reduces data loss because it considers all points in each LiDAR scan, unlike in DeLightLCD where only some points are counted. All points in each LiDAR scan are voxelized according to measurement distance and angle information. A new strategy to realize rotation invariance of loop closure detection is proposed in the feature extraction module. The information in the horizontal direction which is related to the rotation is encoded

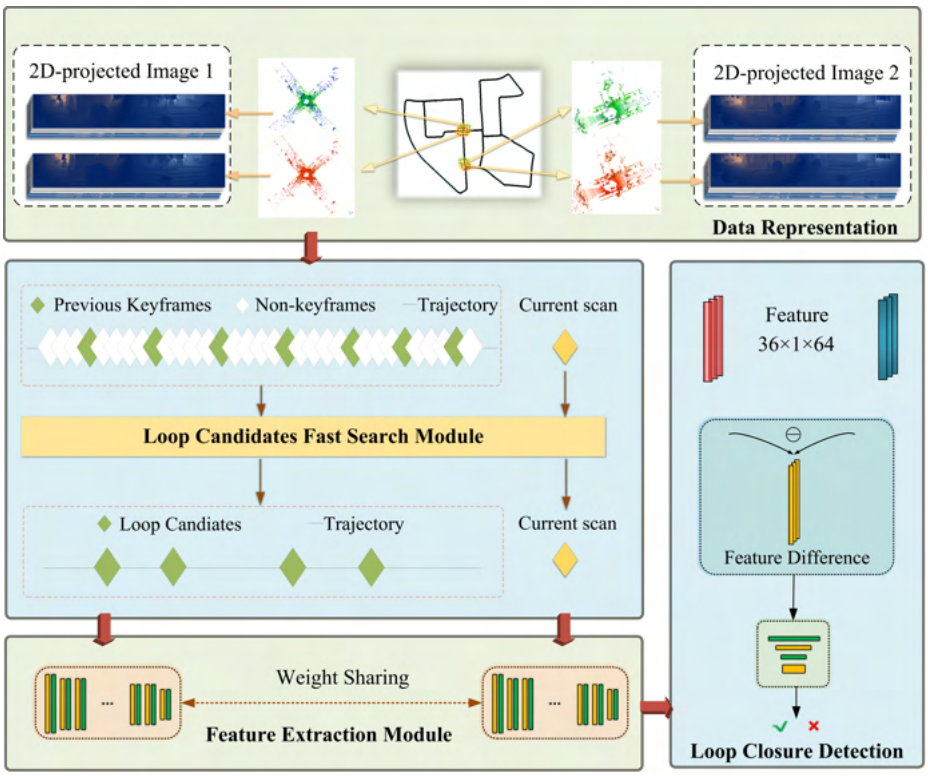


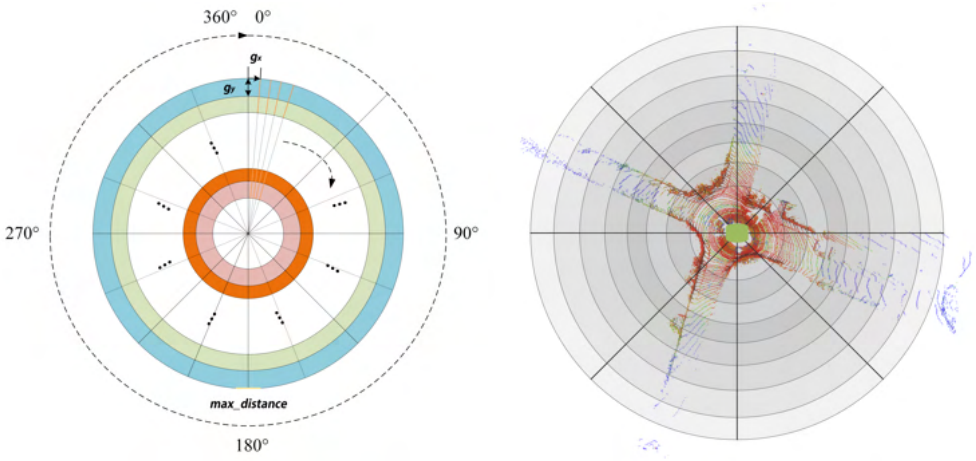
Figure 5.1: Schematic diagram of the proposed DeLightLCD++ algorithm.

into 1-dimension. Then, in the feature difference module, feature broadcasting is not necessary to be conducted. Besides, the loop candidate fast search strategy is proposed to control the time cost, in case it increases exponentially when the measurement continues for a long time. After loop closure detection, the results should be used for pose drift error elimination. A weighted and hierarchy graph optimization strategy is proposed. The framework of DeLightLCD++ is shown in Fig. 5.1.

5.3.1 Data Representation

Projecting raw 3D point clouds to 2D image space has multiple obvious advantages, including reducing data dimensionality, reducing data volume, improving computation time efficiency, and saving computation costs. Besides, transforming 3D point

clouds to 2D images facilitates the direct use of a 2D convolutional deep learning model. If 3D point clouds are input into the deep learning model directly, an inevitable problem should be considered-permutation invariance[173, 148]. The image-based 2D convolutional deep learning model can circumvent that problem. In this chapter, we propose a novel 3D point cloud projection method based on the nominal measurement distance and azimuth angle.



(a) (b)

Figure 5.2: Schematic diagram of the proposed data presentation method.

We project the 3D point cloud $P \in \mathbb{R}^{n_k \times 3}$ into 2D image plane $M \in \mathbb{R}^{H \times W \times 6}$ regarding the nominal measurement distance and azimuth angle. The key idea of this projection method is inspired by Scan Context [74]. First, we divide a 3D LiDAR scan into a 2D coordinate system. The horizontal axis is $[0, 2\pi]$, while the vertical axis is nominal measurement distance from 0 m to maximum ranging distance $[0, D_{max}]$m. The selection maximum distance D_{max} considers a balance of four aspects: the scale of outdoor environments, the measurement distance of laser scanner, the pixel size of projected images, and the sparsity of 2D projected data. In this chapter, we just use the points within 100 m of the sensor. We also set the rows H and columns W for projected 2D images to be 200 and 360. Thus, the pixel sizes of azimuth g_x and ranging distance g_y is 0.5 m and $\pi/180$. Every point in a LiDAR scan will be

projected into 2D spaces to obtain 2D coordinates (u, v).

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \lfloor \sqrt{x^2 + y^2 + z^2}/g_x \rfloor \\ \lfloor \arctan(y, x) \cdot 180/(\pi \cdot g_y) \rfloor \end{pmatrix}, \tag{5.1}$$

where $\lfloor \rfloor$ refers to the round-down function. Points will fall in the same pixel by the 2D coordinates calculation in Eq. 5.1. Then we obtain an image with 200 rows and 360 columns. Then, information with six dimensions will be filled into each pixel. Different from selecting the maximum z-coordinates point in Scan Context, we provide six dimensions of each LiDAR scan in every pixel, including the point number, ranging distance, and mean values of x, y, z, intensity, respectively. The ablation study and final decision of input channel selection is presented in Sec. 5.4.3. Utilizing mean values of x, y, z, intensity, and distance aims at suppressing the outliers and ranging noises of the coordinates. The projected images of six channels are shown in Fig. 5.3.

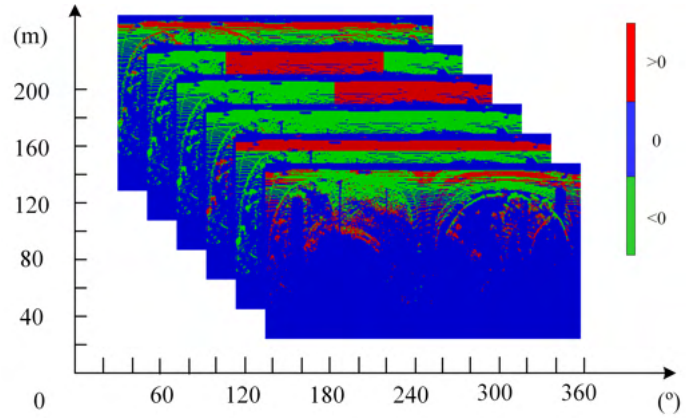


Figure 5.3: Data presentation results of a LiDAR scan.

Similar to Overlapnet [26], DeLightLCD used cyclic projection approach. However, this method is sensitive to the scanner changes. Its row number and column number were set as the number of scan lines and the point number in each scan line. When a new laser scanner was used, the parameters should be adjusted accordingly. Then, the LCD model would become invalid, due to the change of input data dimensions.

In addition, due to a large number of points, Overlapnet only used 900 points in each scan line, which would result in massive data loss. The projection method used in this chapter overcomes those shortcomings. The methods will not be affected by the sensor change and environmental changes. We only need to adjust the pixel size to keep the image size constant at 200×360. Besides, all points are projected and computed to get the six-dimension information. Thus, the method proposed also suppresses data loss to a certain extent.

5.3.2 Feature Extraction Module

Proposed in this study is a deep and lightweight feature extraction network with shared weights to learn high-dimensional and discriminative features from each LiDAR scan. The information of six dimensions including point number in each pixel, ranging distance, and mean values of x, y, z, intensity are leveraged from each LiDAR scan to generate projected maps M.

The architecture of the feature extraction network is shown in Table 5.1. The network is a CNN which contains 15 convolutional layers with batch normalization and 3 average pooling layers. Inspired by the studies in [145], increasing network depth and breadth is a straightforward and effective way to improve network performance. Solid evidence is yielded in [60] that increasing depth is a prior condition for improving accuracy. If the network depth is increased reasonably, more complex, abstract, and high-dimensional features are likely to be learned. Thus, to be designed is a feature extraction network with 15 convolutional layers. The network will thus be deeper than the state-of-the-art LCD networks.

Similar to the DeLightLCD network in Sec. 4, DSC and BN have been adopted to restrain excessive parameter count and gradient vanishing problems, which may lead to non-convergence and the network untrainable. The experiments in Sec. 4.4.3 exhibit the impact of network depth on LCD performance. However, if the network

Table 5.1: Layers of feature extraction network architecture

Operator	Filters	Size	Output Shape
DSC+BN	8	(9,13)	(192,348,8)
DSC+BN	8	(9,13)	(184,336,8)
AP	-	(1,2)	(184,168,8)
DSC+BN	16	(9,11)	(176,158,16)
DSC+BN	16	(9,11)	(168,148,16)
AP	-	(2,2)	(84,74,16)
DSC+BN	32	(7,7)	(78,68,32)
DSC+BN	32	(7,7)	(72,62,32)
AP	-	(1,2)	(72,31,32)
DSC+BN	32	(5,7)	(68,25,32)
DSC+BN	32	(5,7)	(64,19,32)
DSC+BN	64	(5,5)	(60,15,64)
DSC+BN	64	(5,5)	(56,11,64)
DSC+BN	64	(5,3)	(52,9,64)
DSC+BN	64	(5,3)	(48,7,64)
DSC+BN	64	(5,3)	(44,5,64)
DSC+BN	64	(5,3)	(40,3,64)
DSC+BN	64	(5,3)	(36,1,64)

is too deep, BN may also lose effect. Thus, we design this 15-layer feature extraction network, after balancing the LCD performance and computation cost resulting from layers increase.

The feature extraction network utilizes different sizes of kernels in H and W directions. In the vertical H direction, the network aims at extracting features among different scan lines, while the horizontal direction focuses on the information on each scan line. Average pooling (AP) is used to reduce dimensions. The network contains 15 convolutional layers and 3 average pooling layers. Each convolutional layer is composed of a DSC layer and a BN layer. The detailed network information and output

tensor shapes of every layer are shown in Tab. 5.1. The dimension change from input image map M to output features F is $\mathbb{R}^{H\times W\times C} \mapsto \mathbb{R}^{h\times w\times c}$. In the proposed network, h, w, and c are set as 36, 1, and 64, respectively.

It should be specified that the dimensions of horizontal direction which stand for the information in every single scan line are encoded to 1. The information in the horizontal direction is related to rotation. This strategy enables the rotation invariance of the proposed feature extraction network.

5.3.3 Feature Difference Module

Different from the dual-attention-based feature difference module in DeLightLCD, DeLightLCD++ will remove the feature broadcasting step since it has solved the rotation invariance problem in the feature extraction stage. The feature difference module proposed in DeLightLCD++ mainly includes a dual-attention network. A feature vector integrated with dual attention will be fed into a fully connected network to detect loop closures. The process of a feature difference calculation is shown in Fig. 5.4.

The features F_d ($F \in \mathbb{R}^{h\times w\times c}$). It should be noted that h, w and c, are set as 36, 1, and 64, respectively.

Dual attention contains point-wise attention A_p and channel-wise attention A_c. The calculation principle of dual attention has been introduced in Sec. 4.3.3. Channel-wise attention is performed by global average pooling G_c and fully connected layers on each channel across space, whereas point-wise attention is conducted by global average pooling G_p and fully connected layers on every point across the depth.

$$G_p(i,j) = \frac{1}{c}\sum_{u=1}^{c} F_d(i,j,u), \tag{5.2}$$

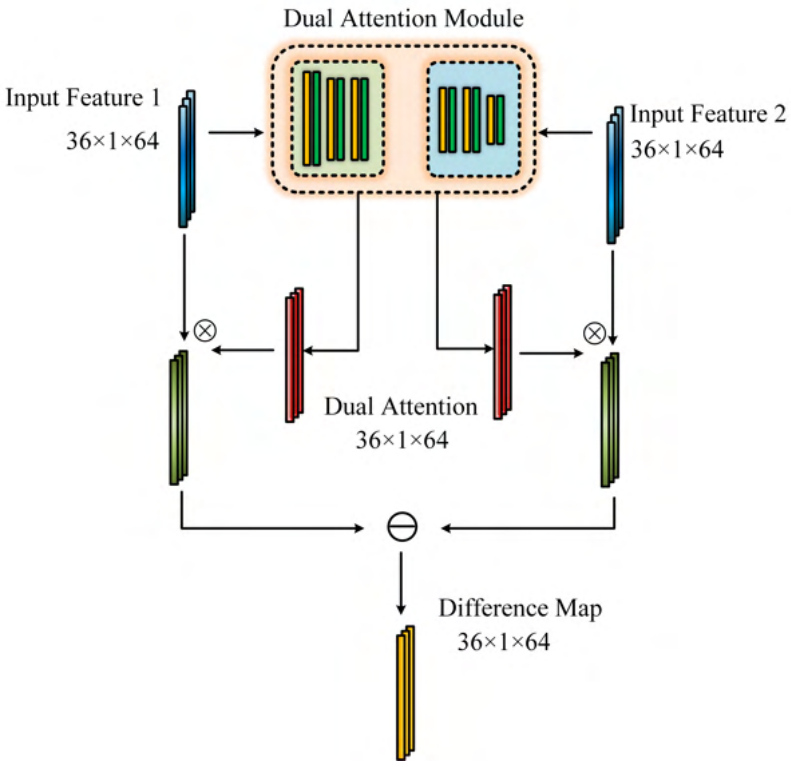


Figure 5.4: The architecture of feature difference module

$$G_c(u) = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} F_d(i, j, u), \tag{5.3}$$

where $G_p(i, j)$ refer to global average pooling in the space domain (i, j), while $G_c(u)$ denotes global average pooling in the channel domain u.

$$A_p(i, j) = sigmoid(\sigma(G_p(i, j))), \tag{5.4}$$

$$A_c(u) = sigmoid(\sigma(G_c(u))), \tag{5.5}$$

where $sigmiod(\cdot)$ refers to the sigmoid function, and $\sigma(\cdot)$ denotes the fully connected layers. The feature difference map $C \in \mathbb{R}^{h \times w \times c}$ is then generated. The map not only contains differential information extracted from a pair of LiDAR scans, but also the weights of each point and each channel.

$$C = F_d \odot \delta(A_c \otimes A_p), \tag{5.6}$$

where $\odot$ and $\otimes$ refers to element-wise multiplication and Kronecker product, respectively. δ denotes the reshaping function to transform the tensor into $\mathbb{R}^{h\times w\times c}$ space.

Based on the dual attention technique, the feature vectors could adaptively recalibrate point-wise and channel-wise feature responses by explicitly building the weight matrix across the space domain and the channel domain.

Different from what is utilized in Sec. 5.4, a simple fully connected neural network is used as a binary classifier to detect whether the LiDAR scan pair constructs a loop. The network contains a 3-layer fully connected network and a sigmoid layer. The network architecture is shown in Fig. 5.5. The input of the network is a difference map generated by the feature difference module with the dimension of $36 \times 1 \times 64$. The probabilities output by this network indicates whether the LiDAR pair is a loop closure.

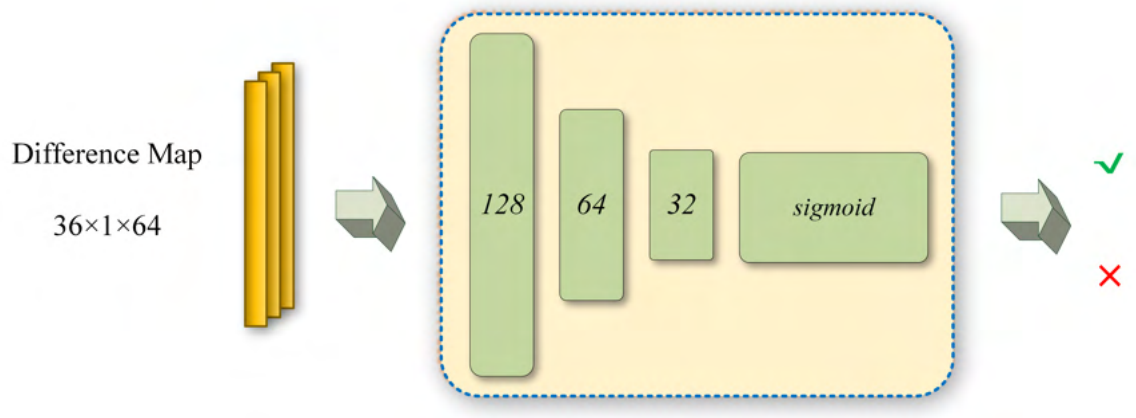


Figure 5.5: The network architecture of the binary classifier network

5.3.4 Loop closure detection

The DeLightLCD++ network was trained end-to-end to detect loop closures from 3D LiDAR scans using a binary cross entropy loss function $\mathcal{L}_{(\mathcal{P}_1,\mathcal{P}_2)}$, where $(\mathcal{P}_1, \mathcal{P}_2)$ is a

pair of LiDAR scans.

$$\mathcal{L}_{(\mathcal{P}_1,\mathcal{P}_2)} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p(y_i)) + (1-y_i)\log(1-p(y_i))], \tag{5.7}$$

where N is the number of training samples, and y_i binary denotes the ground truth. $p(y_i)$ refers to the probability that the pair of LiDAR scans $(\mathcal{P}_1, \mathcal{P}_2)$ form a loop closure. However, the number of training samples for LCD task are always unbalanced which means that the negative samples are much more than the positive samples. Therefore, the classifier is tend to predict a pair of laser scans to the negative which will receive a small loss [172]. The binary cross entropy loss (5.7) will become (5.8):

$$\mathcal{L}_{(\mathcal{P}_1,\mathcal{P}_2)} \approx -\frac{1}{N}\sum_{i=1}^{N}[(1-y_i)\log(1-p(y_i))], \tag{5.8}$$

The the loss could be optimized by weighting it, the weighted loss is formulated as:

$$\mathcal{L}_{(\mathcal{P}_1,\mathcal{P}_2)} = -\frac{1}{N}\sum_{i=1}^{N}[\alpha_{\text{-}} y_i \log(p(y_i)) + \alpha_{+}(1-y_i)\log(1-p(y_i))], \tag{5.9}$$

where $\alpha_{\text{-}}$ and α_{+} are the ratios of the number of negative samples to the total and the ratio of the number of positive samples to the total, respectively. The weights ranges in $[0, 1]$ and $\alpha_{\text{-}} + \alpha_{+} = 1$.

The training process of DeLightLCD++ is presented in *Algorithm* 2.

5.3.5 Loop Candidate Fast Search Strategy

To ensure the LCD efficiency in SLAM, we design the proposed network DeLightLCD. However, with measurement distance increase, the number of LiDAR scan pairs needed to be detected also increases inevitably. Thus, a searching strategy should be utilized to facilitate the high time efficiency of LCD. Generally, the main task for LCD includes pairwise similarity scoring and loop candidates search [74]. In this

Algorithm 2 DeLightLCD Training Workflow

Input: a pair of laser scans $\mathcal{P}_m$, $\mathcal{P}_n$

Output: a trained model

Spherical projection: Equation (5.1):

$\mathcal{C}_m$: size$(N, 200, 360, 6) \leftarrow \mathcal{P}_m$;

$\mathcal{C}_n$: size$(N, 200, 360, 6) \leftarrow \mathcal{P}_n$;

for epoch in enumerate maximum epoch **do**

Siamese feature extraction module:

$\mathcal{F}_m$: size$(N, 36, 1, 64) \leftarrow \mathcal{C}_m$;

$\mathcal{F}_n$: size$(N, 36, 1, 64) \leftarrow \mathcal{C}_n$;

Feature difference module:

$\mathcal{M}$: size$(N, 36, 1, 64) \leftarrow \mathcal{F}_m, \mathcal{F}_n$;

Binary classifier module:

Predict result: $p(y_i) \leftarrow \mathcal{M}$;

Compute loss: Equation (5.9);

Backpropagation;

Update the network parameters;

end for

return a trained model;

chapter, we propose a hierarchical loop closure detection strategy including *Top-N* loop candidate search and loop closure detection to save time cost.

A simple and coarse-grained coding method is used, which is similar to the ring key generation method proposed in [74]. According to the measuring distance of each point, a LiDAR scan is divided into several rings. This measuring distance encoding method ignores the problem of sensor orientation. Thus, it is rotation invariant.

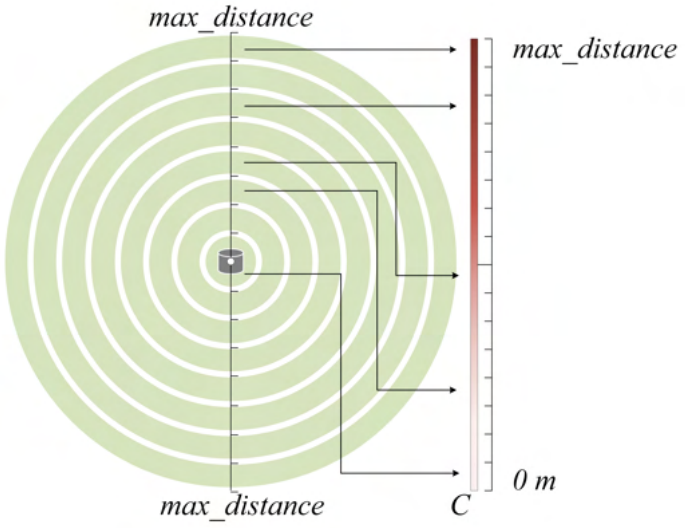


Figure 5.6: The data encoder method for loop closure candidate fast search

As shown in Fig. 5.6, a LiDAR scan is represented as a group of ring features. The features is denoted as C and calculated by Eq. 5.10 and Eq. 5.11. The number of the points falling into the same rings will be counted as the ring features.

$$C = \{\varphi_1, \varphi_2, \ldots, \varphi_K\}, \tag{5.10}$$

$$\varphi_n = ||p_n||_0, \{n \in \mathbb{N} | n \leq K\} \tag{5.11}$$

where φ_n is the ring code, K is the number of rings, and p_n refer to a collection of point falling in n^{th} ring.

The ring features coarsely encode the point distribution of a LiDAR scan. They can be used for loop candidates fast searching. A distance is calculated between the ring feature of the current LiDAR scan and the ring features of the previous scans. The distances are sorted and selected *Top-N* LiDAR pairs as loop candidates. The parameter N are set according to the motion velocity, environment complexity, measurement trajectory, and measurement time. In this chapter, the Euclidean distance is used to search loop candidates.

$$\{C^*\} = \underset{top-N}{\operatorname{argmin}} D(C_i, C_i), \tag{5.12}$$

where $\{C^*\}$ means the selected *Top-N* LiDAR pairs as loop candidates.

The loop candidates will be input into the deep learning model to be determined, by which the calculation cost will reduce sharply. In real circumstances, the majority of the LiDAR pairs are not loops, which could be removed by a fast search strategy. The longer the measurement distance, the more significant the performance of this method to control detection time.

Then, the detection workflow of DeLightLCD++ is presented in *Algorithm* 3.

5.4 Experiments

5.4.1 Datasets

The KITTI odometry datasets, Ford campus datasets, and Mimap datasets were used to evaluate the proposed DeLightLCD approach. The introduction of the three datasets is listed below. It should be emphasized that our network was only trained on KITTI 00 and 08 sequence datasets and never trained on Ford campus datasets and Mimap 00 datasets. Thus, the generalization performance of the proposed method can be evaluated.

Algorithm 3 DeLightLCD++ Prediction Workflow

Input: a sequence of existing keyframe scans $\{\mathcal{P}_0, \cdots, \mathcal{P}_m\}$, the current laser scan $\mathcal{P}_{m+1}$

Output: Loop closure scans of $\mathcal{P}_{m+1}$

Ranging distance histogram $\mathcal{H}_{m+1} \leftarrow \mathcal{P}_{m+1}$;

for $\mathcal{P}_i$ in $\{\mathcal{P}_0, \cdots, \mathcal{P}_m\}$ **do**

Ring features $\mathcal{H}_i \leftarrow \mathcal{P}_i$;

Distance $D_{(\mathcal{H}_{m+1}, \mathcal{H}_i)} \leftarrow \mathcal{H}_{m+1}, \mathcal{H}_i$;

if $D_{(\mathcal{H}_{m+1}, \mathcal{H}_i)} > \tau$ **then**

$\mathcal{D}$ *push back* $D_{(\mathcal{H}_{m+1}, \mathcal{H}_i)}$;

end if

end for

K-D tree construction $\leftarrow \mathcal{D}$;

Loop closure candidates: $\mathcal{P}^* = \underset{\text{Top-K}}{\text{argmax}} \mathcal{D}$;

for enumerate loop closure candidates in $\mathcal{P}^*$ **do**

Spherical projection: Equation (5.1):

$\mathcal{C}_{m+1}$: size$(N, 200, 360, 6) \leftarrow \mathcal{P}_{m+1}$;

$\mathcal{C}_i$: size$(N, 200, 360, 6) \leftarrow \mathcal{P}_i^*$;

Siamese feature extraction module:

$\mathcal{F}_{m+1}$: size$(N, 36, 1, 64)$; $\leftarrow \mathcal{C}_{m+1}$

$\mathcal{F}_i$: size$(N, 36, 1, 64) \leftarrow \mathcal{C}_i$;

Feature difference module:

$\mathcal{M}$: size$(N, 36, 1, 64) \leftarrow \mathcal{F}_{m+1}, \mathcal{F}_i$;

Binary classifier module:

Predict result: $p(y_i) \leftarrow \mathcal{M}$;

if $p(y_i) > 0.5$ **then**

Loop closures: $\hat{\mathcal{P}}^*$ *push back* $\mathcal{P}_i^*$;

end if

end for

return Loop closure scans of $\mathcal{P}_{m+1}$: $\hat{\mathcal{P}}^*$;

- KITTI odometry data were captured by a Velodyne HDL-64E laser scanner. The KITTI sequence 02 dataset was used to evaluate the algorithm, while sequence 00 and sequence 08 were used for training. If the distance between a pair of LiDAR scans is less than 2m, they were considered as positive samples, whereas negative samples are represented by distances greater than 10m. The rigorous training sample thresholds ensured the correctness of training samples. The threshold was set following the work in [148]. The open-source dataset download link is: *https://www.cvlibs.net/datasets/kitti/*

- Tests were also made using the Ford campus 02 datasets. The data were also captured by Velodyne 64-channel laser scanners in outdoor large-scale environments. The open-source dataset download link is: *http://robots.engin.umich.edu/SoftwareData/Fo*

- Mimap 00 dataset is collected in a two-floor building scene, including data of individual rooms, non-enclosed loop corridors, and stairs. The point cloud scans are captured by a Velodyne Ultra puck, a 32-channel laser scanner. The open-source dataset download link is: *https://www2.isprs.org/commissions/comm1/wg6/isprs-benchmark-on-multisensory-indoor-mapping-and-positioning/*

In the experiments, the results are evaluated by three indices: precision, AUC of PR-curve (AUC), and F1-score (F1). F1 is comprehensive evaluation indices calculated by precision and recall rate, while AUC is the area under PR-curve. The precision is evaluated on the same possibility threshold for each algorithm. Each LCD algorithm aims at achieving higher precision and recall rate synchronously, although the two indices are often in an inverse relationship. For LCD tasks, precision plays a primary role because if the wrong loop results are adopted, a mapping disaster may result. Thus, an attempt was made to seek a trade-off between the two indices while maintaining the highest precision. In addition, the experimental setup is represented in Tab. 5.2.

Table 5.2: Parameter configuration of experimental setup

Parameters	Description	Configuration
h	Height of 2D images	200
w	Width of 2D images	360
σ_p	Threshold of positive samples	2 m
σ_n	Threshold of negative samples	10 m
e	Epochs	50
l	Learning rate	0.001
f	Factor of learning rate reduced	0.1
d	Decay of every epoch	1e-6
m	Momentum	0.9

5.4.2 Loop Closure Detection

DeLightLCD++ is first evaluated on outdoor large-scale environments. The results are presented in Tab. 5.3. In outdoor large-scale environments, DeLightLCD++ was compared with PointNetVLAD [148], LPD-Net [101], OverlapNet [26], Scan Context [74], and DeLightLCD proposed in Sec. 4. DeLightLCD++ shows superior performance on KITTI 02 datasets, while the results on Ford 02 dataset is second only to DeLightLCD. That exhibits DeLightLCD++ is also effective and accurate. Although the generalization performance is slightly weaker than DeLightLCD in outdoor environment, DeLightLCD++ has another irreplaceable advantage: indoor-outdoor LCD. OverlapNet and DeLightLCD could only be used to process the data captured from limited LiDAR types, while DeLightLCD++ ignores sensor type changes and environment scale changes.

The indoor LCD experiments are evaluated on Mimap 00 dataset. DeLightLCD++ was compared with some popular algorithms, M2DP [62], FastHistogram [123],LiDAR Iris [156], and FastLCD proposed in Sec. 3. FastLCD is trained and dedicated in

Table 5.3: Comparison Experimental Results on Outdoor Datasets

	Methods	Precision	AUC	F1-score
KITTI 02	PointNetVLAD[148]	0.81	0.83	0.79
	LPD-net[101]	0.83	0.83	0.85
	OverlapNet[26]	0.94	0.86	0.87
	Scan Context[74]	0.92	0.96	0.89
	DeLightLCD[164]	0.96	0.99	0.90
	DeLightLCD++(ours)	0.98	0.99	0.95
Ford 02 campus dataset	PointNetVLAD[148]	0.67	0.64	0.72
	LPD-net[101]	0.74	0.77	0.77
	OverlapNet[26]	0.85	0.88	0.83
	Scan Context[74]	0.78	0.50	0.46
	DeLightLCD[164]	0.88	0.99	0.93
	DeLightLCD++(ours)	0.96	0.92	0.90

solving LCD in indoor environments. Thus, it outperforms other algorithms, while DeLightLCD++ is second only to FastLCD. It should be emphasized that although DeLightLCD++ is never trained on any indoor datasets, it can also obtain comparable results as FastLCD. The results on indoor environments exhibit that DeLightLCD++ has great generalization ability and flexibility in indoor-outdoor seamless LCD.

5.4.3 Ablation Study

The ablation study of DeLightLCD++ is performed on three aspects, the dual-attention mechanism, the input channels, and the loop candidate fast search.

As shown in Tab. 5.5, the input channels are studied. The data representation method introduced in Sec. 5.3.1 contain six input channels: X coordinates (X), Y coordinates (Y), X coordinates (Z), intensity (I), the point number in each grid (N),

Table 5.4: Comparison Experimental Results on Indoor Datasets

Methods	AUC	F1-score
M2DP[62]	0.97	0.91
FastHistogram[123]	0.90	0.81
LiDAR Iris[156]	0.64	0.53
FastLCD[163]	1.00	0.94
DeLightLCD++(ours)	0.91	0.90

and mean value of distances in each grid (D). Because the Ford 02 data does not contain intensity information, the intensity-relevant results are not presented. The 6-channel input obtains the best performance. It is significant to find that XYZ gets a comparable result as 6-channel input, while $XYZN$ and $XYZND$ are failed. Then, we could infer that the input information that plays a decisive role is the XYZ coordinate value. Adding the input information of N and D makes the model learning chaotic and could not give correct prediction results. The reason may be that intensity information is not reliable. It is affected by many factors, like the object materials, the incidence angle, the shooting angle, and the measurement distance. Thus, the values of intensity may differ even the information is captured at the same place. So, the relationships learned from intensity maybe not consistent with features learned from coordinates. Besides, the distance D is correlated with the coordinate information (X, Y, Z). The addition of D could not improve the performance. Thus, we select 5 input channels combination of $XYZIN$ as the final version.

Ablation study *w.r.t.* dual-attention mechanism including point-wise attention (P) and channel-wise attention (C) is presented in Tab. 5.6. Dual-attention obtains the best performance. We could find that channel-wise attention alone is useless for LCD, while point-wise attention is significant. However, DeLightLCD++ without dual-attention also get a acceptable result on KITTI 02 result but almost fails on Ford 02 dataset which means that DeLightLCD++ without dual-attention is weak in

Table 5.5: Ablation study *w.r.t.* input channels

						KITTI02		Ford02	
X	*Y*	*Z*	*I*	*N*	*D*	AUC	F1	AUC	F1
✓	✓	✓	×	×	×	0.99	0.92	0.92	0.90
✓	✓	✓	✓	×	×	0.97	0.88	-	-
✓	✓	✓	×	✓	×	0.22	0.16	0.24	0.15
✓	✓	✓	×	×	✓	0.40	0.51	0.35	0.48
✓	✓	✓	✓	✓	×	0.99	0.95	-	-
✓	✓	✓	✓	×	✓	0.99	0.94	-	-
✓	✓	✓	×	✓	✓	0.20	0.23	0.18	0.17
✓	✓	✓	✓	✓	✓	0.99	0.95	-	-

generalization ability.

Table 5.6: Ablation study *w.r.t.* dual-attention

		KITTI02		Ford02	
P	*C*	AUC	F1	AUC	F1
×	×	0.74	0.83	0.51	0.68
×	✓	0.59	0.46	0.54	0.53
✓	×	0.86	0.82	0.74	0.66
✓	✓	0.99	0.95	0.92	0.90

5.4.4 Time Efficiency and Parameters

The time efficiency and parameter amount were compared with some deep-learning-based LCD algorithms. The time efficiency was tested on a computer equipped with the Intel Xeon E5-2660 v4 CPU and two NVIDIA Tesla P100 16GB GPUs. Among these algorithms, DeLightLCD++ also shows great time efficiency, only about 5 ms

for each detection, slightly higher than DeLightLCD. The binary classifier used in DeLightLCD is a CNN network with a DSC mechanism which has a relatively low parameter amount, while DeLightLCD++ uses an FC network with more parameters than DeLightLCD. Thus, the time cost is slightly higher than DeLightLCD. However, when the measurement distance is very long, the time cost increase of DeLightLCD is not controlled, while loop candidate fast search in DeLightLCD++ suppresses the time cost to an acceptable level. In sum, DeLightLCD++ shows superior performance in time efficiency.

Table 5.7: Comparison of parameter count and time cost

Algorithm	Parameters	Time cost (ms/detection)
PointNetVLAD[148]	2 M	24.88
LPD-Net[101]	2 M	22.94
OverlapNet[26]	921 K	8.88
DeLightLCD[164]	89 K	1.69
DeLightLCD++ (ours)	185 K	5.32

As shown in Fig. 5.7, as the number of laser point cloud frames increases, there is a varying increase in time cost. Although the single detection time of the presented algorithms is all acceptable, a lack of efficient detection strategy makes the time cost shows an exponential growth for PointNetVLAD, LPD-NET, OverlapNet, DeLightLCD, and DeLightLCD++(no loop candidates fast search). The time cost curve of DeLightLCD++ nearly presents a linear growth with the increase of laser scans, while the other 5 curves shows an exponential trend. Thus, DeLightLCD++ is more practical due to its superior time efficiency, especially when the number of laser scans is very large.

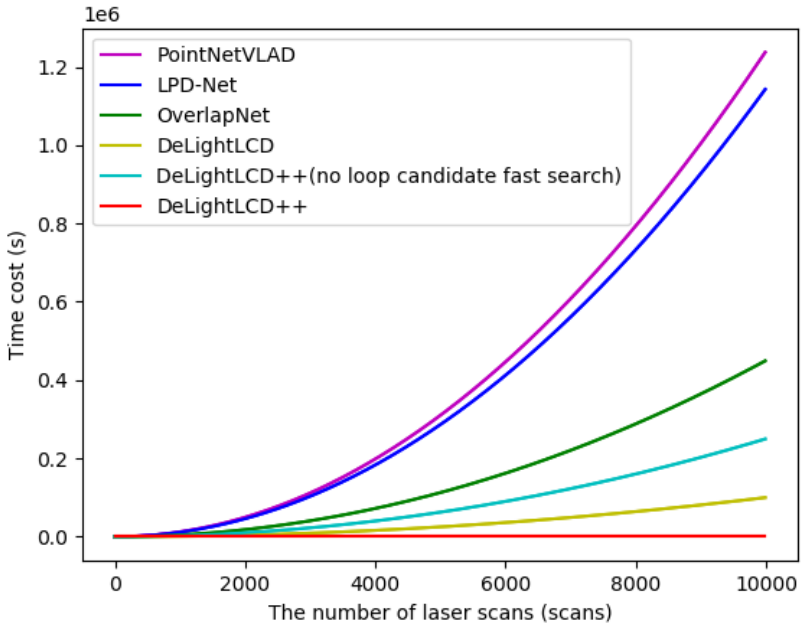


Figure 5.7: The time cost with the number of laser scans increasing.

5.5 Discussion and Limitations

According to the experiment results of the proposed DeLightLCD++ algorithm, the advantages are threefold.

- Due to the distance-angle data representation method, DeLightLCD++ just uses the measurement distance and azimuth angle which circumnavigates the usage of the number of scan lines, angle resolution, and field of view. Thus, the data representation is not limited by sensor type changes and environment scale changes.
- The data representation is calculated by all points in each LiDAR scan. Compared with DeLightLCD, it suppresses data loss.
- With measurement distance increasing, loop candidate fast search controls the time cost of LCD.

However, there are still some limitations. The parameter amount could be reduced to a lower level. Limited by the computer hardware performance, the sample number is limited. Thus, more training samples, such as indoor data samples, could be used to improve the LCD performance in indoor environments.

5.6 Conclusion

In this section, an improved LCD algorithm DeLightLCD++ is proposed to solve the problems of DeLightLCD. The main problems of DeLightLCD includes time cost in large-scale scenes and data loss. The improvement is threefold. A novel data representation method encoding measurement distance and the azimuth angle is used to circumnavigate the sensor type limitation. A new feature extraction network is designed to ensure rotation invariance. Besides, a loop candidate fast search method is proposed to control the time cost of LCD. The strategy can not only suppress the time cost but also reject false loop closures in advance. Experimental results on three open-source datasets demonstrate the great performance and flexibility of DeLightLCD++. In the future, more training samples including outdoor data and indoor data should be used to train a more precise and robust model.

Chapter 6

An Enhanced Graph Optimization in SLAM based on LCD

6.1 Introduction

A pair of loop-closed LiDAR scans will be associated and registered after LCD for back-end optimization. According to the theory of graph optimization, an edge will be added to the graph as a redundant observation. This facilitates the SLAM pipeline for eliminating cumulative errors. In this chapter, we will apply the LCD results to optimization the precision of the LiDAR odometry in SLAM using an enhanced graph optimization method. The weight is generated according to the fitness score of two LiDAR scans registration. The main contributions of the chapter are:

- The LCD results are utilized to eliminate the cumulative drift error and build a consistent and accurate map by the enhanced graph optimization. The enhanced method is evaluated qualitatively and quantitatively on open-source datasets and in-house datasets.
- The impact of loop closures will be investigated, including the scale parameter

of the weight, the precision of the loop closures, and the types of the loop closure edges. Quantitative experiments will be conducted to study the impacts of some factors of loop closures to map building.

- Some discussions, analysis, and guidance of error distribution and fieldwork measurement solutions for mobile mapping backpack will be given.

6.2 Related Works

Based on the previous work represented in *Chapter* 3, 4, and 5, the loop closures are detected. Then, the results should be used in the SLAM pipeline. In this chapter, an enhanced graph-based back-end optimization approach is used by a weighting method. Qualitative experiments are conducted to show the optimization performance, while quantitative experiments are conducted to exhibit the impact of weights and loop numbers. Finally, some advice will be given for measurement implementations of fieldwork.

Currently, SLAM solutions based on graph and back-end optimization become mainstream in the field. Some popular graph optimization techniques like Ceres [1], g^2o [81], and GTSAM [40], etc. In this section, these common graph optimization libraries will be introduced.

Ceres solver [1] is an open-source C++ library. It is developed to address the nonlinear least-squares problems with bounds constraints. It is widely used for pose estimation in SLAM and some general unconstrained optimization problems. The library is easy to use, flexible, mature, and performant library. Some fundamental trust-region methods algorithms are integrated like Levenberg-Marquardt and Powell's Dogleg. As one of the most mature and popular optimization libraries, Ceres has been used in many applications, especially for pose estimation or odometry in SLAM [89, 121].

g^2o [81] is an open-source C++ framework for optimizing nonlinear error functions

that can be defined as graphs. It has the advantage of being easily extensible, efficient, and applicable to a wide range of problems, especially in SLAM and bundle adjustment (BA) problems. It has many advantages: (1) g^2o provides comparable performance to implementations of state-of-the-art approaches with a high degree of versatility and scalability. (2) efficient computation is performed by exploiting sparse connectivity, the special structure of graphs, and the characteristics of modern processors. (3) the framework integrates three different pose graph optimization algorithms: Gauss-Newton, Levenberg-Marquard, and Powell's Dogleg. Since it has these advantages, some popular SLAM solutions utilize g^2o for back-end optimization, like ORB-SLAM [109], SVO [48], HDL graph SLAM [78].

GTSAM [40] is another state-of-the-art C++ open-source library. It uses the factor graph as the basic theory to model complex problems. It also integrates incremental smoothing and mapping methods, iSAM [72] and iSAM2 [71] to provide an efficient solution to the SLAM problem. It enables sensor fusion for robotics and computer vision applications. It also integrates Gauss-Newton, Levenberg-Marquard, and Powell's Dogleg optimization algorithms. GTSAM is known for its high efficiency. Therefore, it is popular in many online or real-time SLAM solutions, like LIO-SAM [132], LVI-SAM [133] and a variant of SVO [47].

The three all can provide efficient and effective pose estimation for SLAM. According to the study in [41, 70], the three pose estimation solutions all could perform significant results, while there still are some minor differences. Based on the quantitative experiments on many benchmarks, g2o performs relatively slightly best on precision among the three methods, while GTSAM is time-efficient.

In this chapter, an enhanced graph optimization strategy is proposed based on g^2o. Because the odometry drift error will grow with the measurement distance increasing, we assume that if the registration precision of a pair of loop LiDAR scans is higher, the more it will optimize the whole graph. Then, we could trust the loop edges with better registration performance and set the heavy weight to them. Thus, we

use an enhanced graph constructed by a weighted edge for optimization. After a loop closure between two LiDAR scans x_i and x_j is detected, point cloud registration will be performed to obtain the transformation matrix between the two point clouds. Then, an edge z_{ij} will be added into the graph connecting the two vertices x_i and x_j. The edge will yield a heavier weight if the fitness score of point cloud registration is lower.

6.3 Graph Optimization in SLAM Pipeline

Graph optimization can be treated as a nonlinear least squares problem, which is generally transferred to form a linear system around the current state [81]. A graph is consist of edges and vertices. An edge connects two vertices. The vertex i represents the state parameters x_i in a graph. Each x_i is a generic parameter block. An edge between vertex i and j represents an ordered constraint , represented as $\mathrm{z}_{i,j}$. In graph-based optimization, the objective function is represented as

$$\mathrm{F(x)} = \sum_{<i,j>\in \mathrm{C}} \mathrm{F}_{i,j} \tag{6.1}$$

$$\mathrm{F}_{i,j} = \mathrm{e}(\mathrm{x}_i, \mathrm{x}_j, \mathrm{z}_{i,j})^{\top} \Omega_{i,j} \mathrm{e}(\mathrm{x}_i, \mathrm{x}_j, \mathrm{z}_{i,j}), \tag{6.2}$$

$$\mathrm{x}^* = \mathrm{argmin} \mathrm{F(x)} \tag{6.3}$$

$\mathrm{e}(\mathrm{x}_i, \mathrm{x}_j, \mathrm{z}_{i,j})$ represents the error function that expresses how well the state parameters x_i and x_j fit the constraint $\mathrm{z}_{i,j}$. The error function will be 0 when x_i and x_j perfectly fit the constraint. $\Omega_{i,j}$ is information matrix of the constraint $\mathrm{z}_{i,j}$. Our aim is to calculate the parameter blocks x to find the minimum of the objective function F(x).

Fig. 6.1 depicts the architecture of a graph and parameters in it. Many solutions have been proposed, like *Gauss-Newton* (GN) method or *Levenberg-Marquardt* (LM)

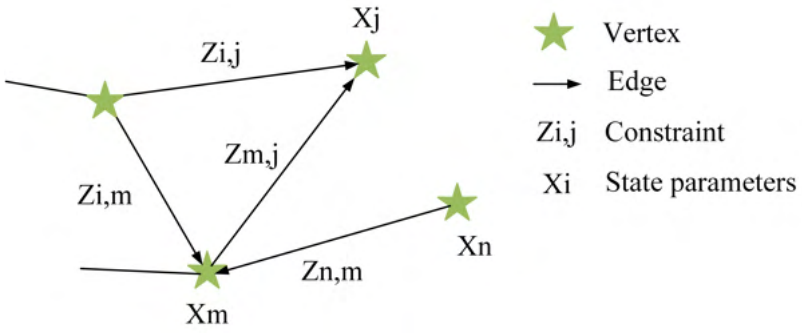


Figure 6.1: The architecture of a graph

method. In the graph optimization framework *g2o*, it provides both GN and LM algorithms to solve the problem. The characteristic of sparse connectivity of the graph is exploited to simplify the solution and improve the computation efficiency.

6.4 Optimization based on LCD

In the LiDAR-SLAM graph optimization, edges in a graph have two sources: pose relationship computed by point cloud registration in front-end odometry and the pose relationship between two LiDAR scans detected as a loop closure. The latter can be regarded as a redundant observation in the adjustment problem. Then, we call the edges computed from LiDAR odometry as odometry edge, while the edges computed from LCD result as loop closure edges. As the measurement distance increases, no matter how accurate the point cloud registration algorithm is, it will suffer from the problem of error accumulation. Cumulative drift errors can make the SLAM build results inconsistent with reality. Thus, the pose relationship between a pair of loop scans will be added to the graph to eliminate the drift error.

In the proposed enhanced graph optimization method, weights will be discussed according to the edge types. We set weights to edges to improve the performance of graph optimization. The weights are calculated from the registration fitness score

then used in the information matrix.

The weights calculation will be discussed according to two edge types. For odometry edges, the frame-by-frame LiDAR odometry utilizes a point cloud registration algorithm. The error of frame-by-frame registration is not very large. Thus, for simplicity, we set the information matrix of the odometry matrix as identity matrix, which means each odometry edge plays the same importance level in graph optimization.

As for loop closure edges, the constraint $\mathrm{z}_{i,j}$ is computed from point cloud registration. The fitness score f of point cloud registration refers to mean squared error (MSE). A pair of loop scans S and T is registered. The valid correspondence point is denoted as $\{(s_i, t_i)|s_i \in S, t_i \in T\}$. The fitness score is calculated as:

$$f = \frac{\sum_n D_{(\hat{s_i}, t_i)}}{n}, \tag{6.4}$$

where, n refers to the number of valid corresponding points, and $\hat{s_i}$ denotes the point s_i in source point cloud transformed to the target point cloud space. Thus, we could find that the lower the fitness score is, the better the registration performance between the pair of the loop LiDAR scans.

Then, the weight of loop closure edges W is calculated in Eq. 6.5, in which φ and α both are constant-coefficient to make the equation workable. α is set to make $\alpha f < 1$, while φ is a scale parameter, a multiplier coefficient to scale the weights. Besides, a threshold T_f needs to be set for the fitness score to reject the edge constraints with large error.

$$\mathrm{W} = -\frac{\varphi}{\ln(1 - \alpha f)}, \;\; s.t. \;\; \alpha f < 1 \tag{6.5}$$

The weight function W is a monotonic decreasing function when $\{f \in [0, \mathrm{T}_f]|\alpha f < 1\}$. The graph of the proposed weight function is shown in Fig. 6.2. The range of value f is $(0, 1)$. When a pair of loop scans have a small fitness score, which means they are registered well together. Then, the edge between will get a heavy weight. Here, we assume that the loop closure edges with low fitness scores should get heavier

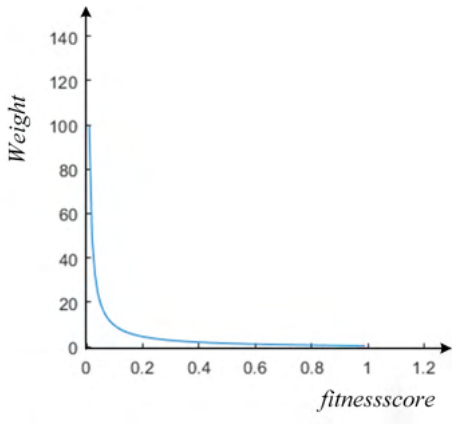


Figure 6.2: The graph of the proposed weight function

weights than those of odometry edges to control drift error better. By contrast, if a loop closure edge gets a high fitness score, the light weight will be determined. Thus, we design a weight calculation function shown in Eq. 6.5. The function maps $\alpha f \in [0, 1] \mapsto \mathrm{W} \in [0, +\infty]$.

$$\mathrm{M} = \mathrm{W} \cdot I_6 \tag{6.6}$$

Then, the information matrix M is computed as Eq. 6.6. 6-dimensional identity matrix means the 6-DoF parameters in the same parameter block share the same weights.

6.5 Experiments

In this part, experiments are demonstrated to exhibit the effectiveness of the optimization method. Then, comparative experiments are conducted to analyze the impacts of some factors on the optimization results. Datasets used in this chapter are KITTI benchmark 00 datasets [54, 53] and in-house dataset. The introduction of the experimental dataset is in Sec. 5.4.1.

(a) (b)

Figure 6.3: Optimization results on various datasets (a) KITTI 00 dataset, (b) in-house dataset.

6.5.1 Qualitative Experiments

As shown in Fig. 6.3, in KITTI 00 dataset and in-house indoor dataset, the effectiveness of the proposed optimization method is obvious. The optimized trajectory is more consistent with the ground truth compared with the trajectory before optimization. The trajectory of LO before optimization is computed by the algorithm in [21].

6.5.2 Quantitative Experiments

According to the results reported in Sec. 6.5.1, it is shown the effectiveness of LCD and graph optimization to the whole trajectory results. Furthermore, we try to study the impact of some factors on optimization performance. According to the definition of LCD, we could broaden the scope that if two LiDAR scans are similar enough, the two scans are determined as a loop. Based on this explanation, the loop closures are

divided into three types, detected loops, pseudo loops, and enhanced loops.

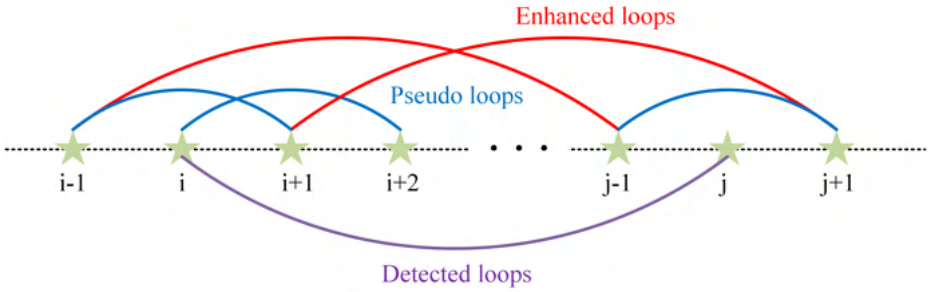


Figure 6.4: Schematic diagram of three types of loops: detected loops, pseudo loops, and enhanced loops

As shown in Fig. 6.4, i and j are ID of scans . They are connected as a loop detected by LCD algorithms. There is a long time interval between them. The sensor has experienced a scene transition during this time interval. $(i-1)$ and $(i+1)$ are two adjacent scans of scan i and j. Due to the high data capture frequency, the sensor will capture many LiDAR scans in a short time interval. Scans $(i-1)$ and $(i+1)$ are also highly-similar but not adjacent. The loops between the two scans are pseudo loops. If we add pseudo loops to the graph, a dense graph will be controlled by these pseudo loops. In this section, we will study the impact of adding pseudo loops to the graph. Besides, according to time consistency and geometry consistency, if a pair of scans are detected as a loop closure, the adjacent scans must be loop closures too. Because of the high data capture frequency of LiDAR sensors, many scans will be collected at a very close moving distance. Thus, in this section, we will also research whether the optimization performance will be better if we add enhanced loops to the graph. The comparative experimental results are shown in Tab. 6.1, 6.2, 6.3, 6.4, and Fig. 6.5. For evaluation, average translation error (ATE) against ground truth pose information is used [111].

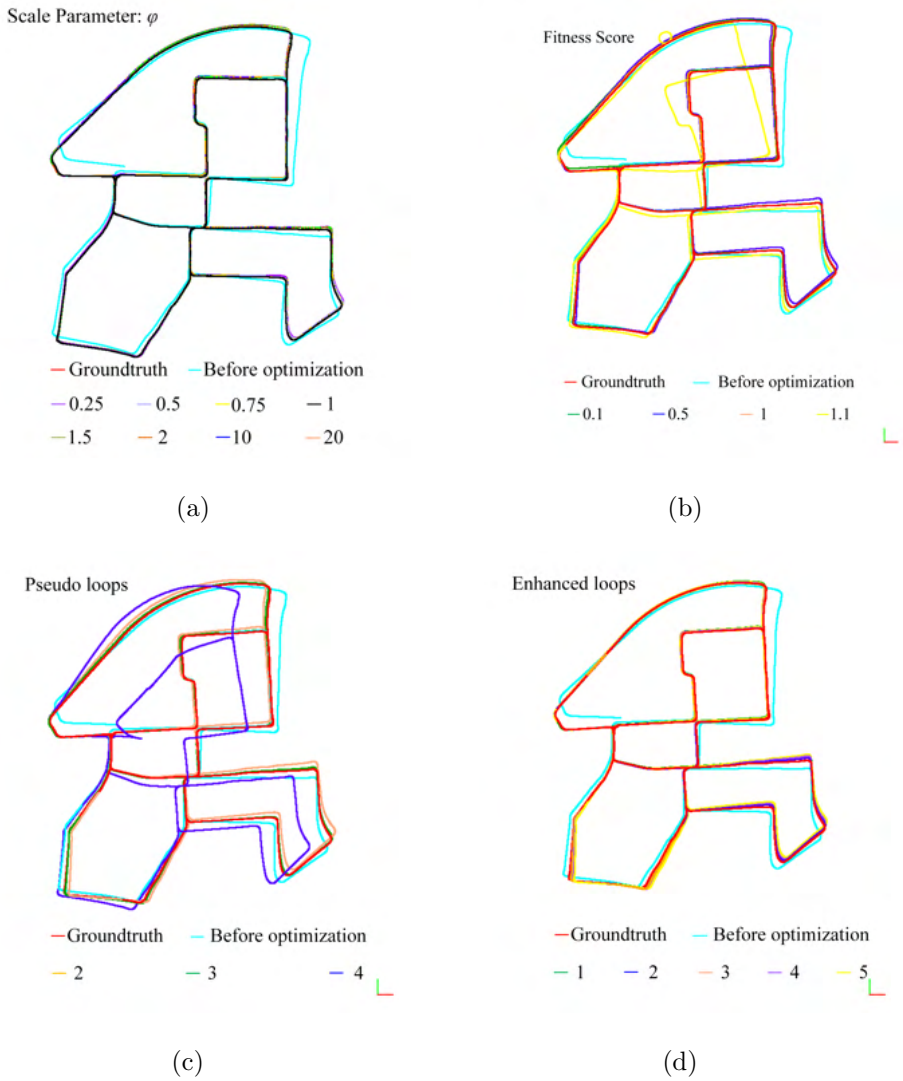


Figure 6.5: The impact of four factors on optimization performance. (a) Impact of weight parameter: φ, (b) impact of the precision of loop closure edges, (c) impact of adding pseudo loops, (d) Impact of adding enhanced loops.

Table 6.1: Impact of scale parameter of weight: φ

	Weight	Fitness Score	Loop Closure Number	Enhanced Loop Buffer	Pseudo Loop Gap	ATE(cm)
Before	-	-	-	-	-	14.98
After	-	1	1132	-	-	8.25(↓ 44.93%)
	0.25	1	1132	-	-	8.20(↓ 45.26%)
	0.5	1	1132	-	-	8.08(↓ 46.06%)
	0.75	1	1132	-	-	8.04(↓ 46.33%)
	1	1	1132	-	-	**8.03(↓ 46.40%)**
	1.5	1	1132	-	-	8.04(↓ 46.33%)
	2	1	1132	-	-	8.05(↓ 46.26%)
	5	1	1132	-	-	8.12(↓ 45.79%)
	10	1	1132	-	-	8.17(↓ 45.46%)
	20	1	1132	-	-	8.22(↓ 45.13%)

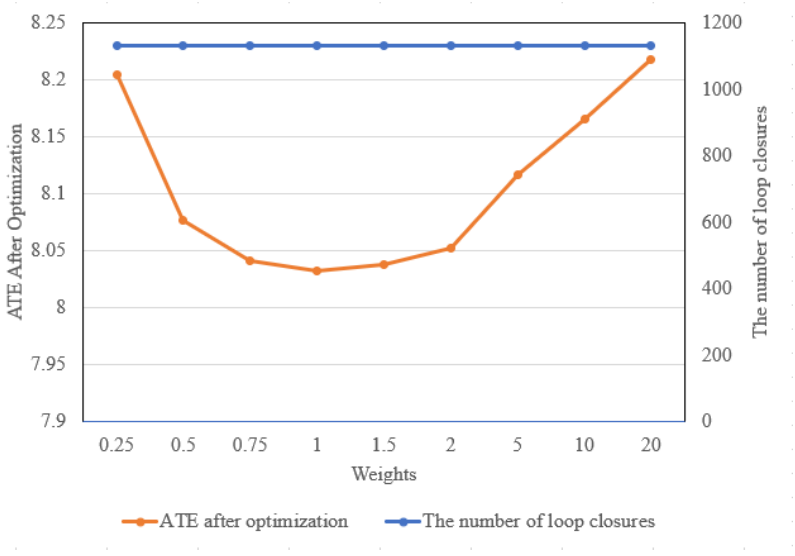


Figure 6.6: Experimental results of scale parameter changing

In the experiment of research on the impact of the scale parameter on optimization performance, different values of the scale parameter φ are set. The scale parameter determines the value of the weight. The fitness score threshold is set as 1 so that the number of loop closures remains the same, while no pseudo edges and enhanced edges are added to the graph. According to the result reported in Tab. 6.1, a heavy weight of loop closure edge will be set if φ is set a large value. We could find that the scale parameter affects the results not very significantly, while when the scale parameter is 1, the optimization results are comparably best. The weights bigger or smaller than 1 all result in slightly worse performance.

Table 6.2: Impact of the precision of loop closure edges

	Weight	Fitness Score	Loop Closure Number	Enhanced Loop Buffer	Pseudo Loop Gap	ATE(cm)
Before	-	-	-	-	-	14.98
	1	0.1	16	-	-	9.67(↓ 35.45%)
	1	0.2	179	-	-	8.64(↓ 42.32%)
	1	0.3	356	-	-	9.18(↓ 38.72%)
	1	0.4	544	-	-	10.75(↓ 28.24%)
	1	0.5	703	-	-	10.04(↓ 32.98%)
	1	0.6	824	-	-	9.42(↓ 37.12%)
	1	0.7	908	-	-	8.97(↓ 40.12%)
	1	0.8	1005	-	-	8.71(↓ 41.86%)
After	1	0.9	1078	-	-	8.54(↓ 42.99%)
	1	1.0	1132	-	-	**8.03(**↓ 46.40%**)**
	1	1.1	1174	-	-	14.66(↓ 2.14%)
	1	1.2	1225	-	-	19.64(↑ 31.11%)
	1	1.4	1302	-	-	31.8(↑ 112.28%)
	1	1.6	1342	-	-	22.13(↑ 47.73%)
	1	1.8	1398	-	-	24.39(↑ 62.82%)
	1	2.0	1437	-	-	30.15(↑ 101.27%)

In the following experiment, we try to adopt loop closure edges with different fitness

score to explore the effect of loop closure edge precision on optimization results. It should be indicated that if the registration between a pair of loop-closed laser scans reach a low fitness score, the precision of the loop closure edge will be higher and reliable. In this experiments, no pseudo loops and enhanced loops are added to the graph and the scale parameter φ is set as 1. According to the results in Tab. 6.2, we could find that optimization reaches the best performance when the threshold is 1, while the results are almost ruined at threshold 1.4. When a rigorous threshold is set smaller than 1, the edge constraints added must be reliable and precise. However, ATE results are worse because fewer loops are added and the control information is insufficient. The more rigorous threshold is selected, the fewer detected loops could be adopted. Thus, a balance should be reached between the loop precision threshold and loop numbers.

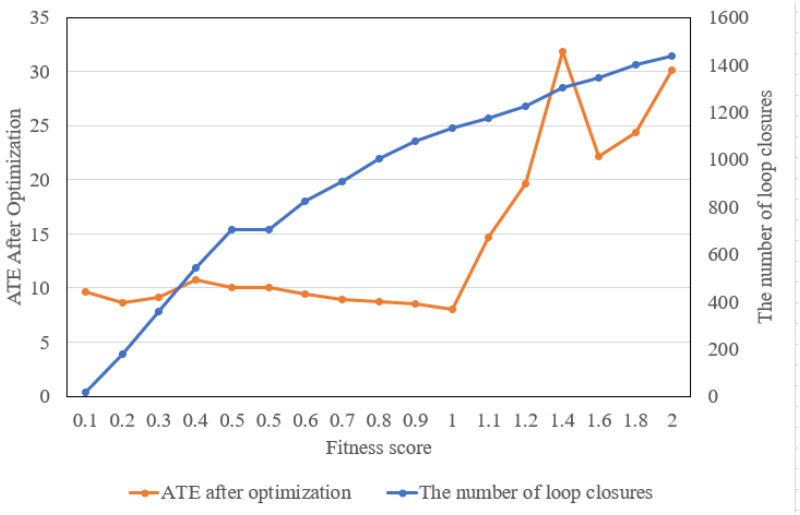


Figure 6.7: Experimental results of different loop precision

The following experimental design is originally intended if adding many loop closure edges to the graph, even though they are pseudo loops, a dense and strongly controlled graph will be generated. A study is conducted to explore whether it will improve the optimization performance. The parameter of pseudo loop gap means that a certain number of frames separate two point clouds, which are aligned as a pseudo loop

closure. The scan pairs that meet the fitness score threshold requirements are added to the graph as pseudo loop closure edges. In this experiment, the value of scale parameter and fitness score are set as 1 and 1, respectively. According to Tab. 6.3, when the gap is set as 2 and 3, the results are better than ATE before optimized, while ATE grows sharply when the gap number increases. The reason may be that if the gap is large, the number of pseudo edges reduced and the effective control information is insufficient, so that the graph optimization is ruined.

Table 6.3: Impact of adding pseudo loops

	Weight	Fitness Score	Loop Closure Number	Enhanced Loop Buffer	Pseudo Loop Gap	ATE(cm)
Before	-	-	-	-	-	14.98
After	1	1	4966	-	2	8.27(↓ 44.79%)
	1	1	4333	-	3	**7.75(↓ 48.26%)**
	1	1	3669	-	4	51.75(↑ 245.46%)
	1	1	3068	-	5	69.03(↑ 360.81%)
	1	1	2636	-	6	48.01(↑ 220.49%)

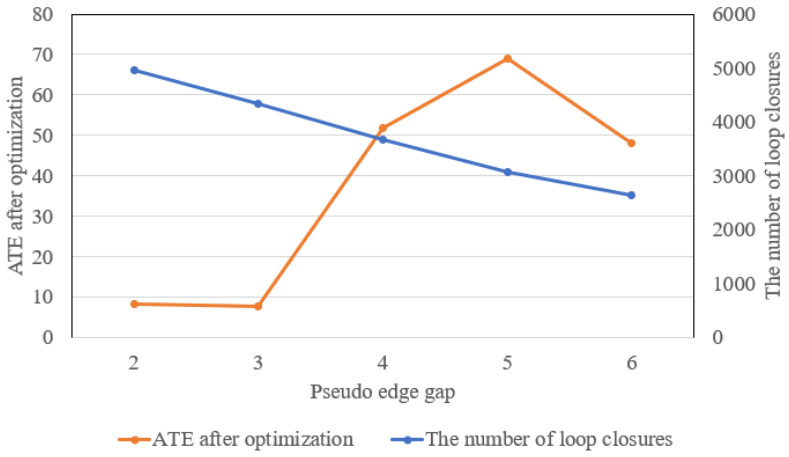


Figure 6.8: Experimental results of different pseudo edge gap

Table 6.4: Impact of adding enhanced loops

	Weight	Fitness Score	Loop Closure Number	Enhanced Loop Buffer	Pseudo Loop Gap	ATE(cm)
Before	-	-	-	-	-	14.98
	1	1	3201	1	-	8.01(↓ 46.53%)
	1	1	5099	2	-	**4.85(↓ 67.62%)**
After	1	1	6818	3	-	4.93(↓ 67.09%)
	1	1	8353	4	-	5.80(↓ 61.28%)
	1	1	9714	5	-	6.17(↓ 58.81%)

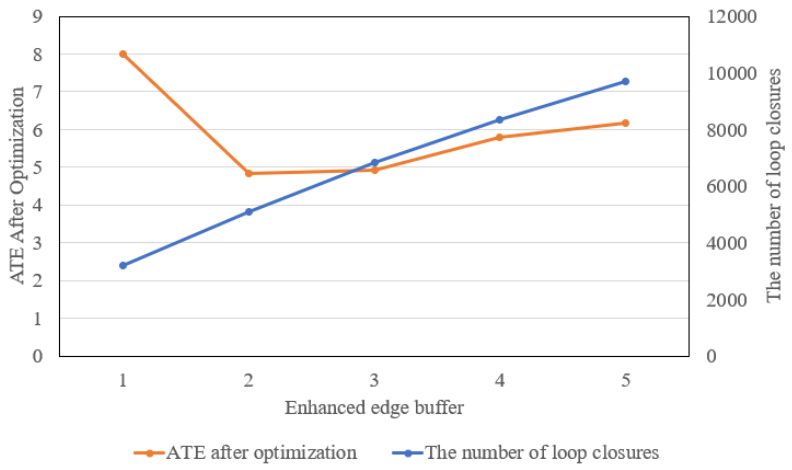


Figure 6.9: Experimental results of different enhanced edge buffer

As for enhanced loops, they are determined by time consistency and geometry consistency. The enhanced loop buffer in Tab. 6.4 means that in the range of a certain number of scans adjacent to the detected loop scans, the scans are selected and registered to obtain the pose relationships. The enhanced loops which meet the fitness score threshold requirements are adopted and added to the graph.In this experiment, the value of scale parameter and fitness score are set as 1 and 1, respectively. We could find that the results are improved significantly compared with the results of weight, precision, and pseudo loops experiments. However, the number of loop closure edges

increases sharply. Thus, some enhanced loop closure selection strategies could be used to control the number of enhanced loop closure edges to save computation cost. When the enhanced loop buffer is set as 1, 3201 loop closure edges are added into the graph, while when the pseudo loop gap is set as 5, the similar number of loop closure edges, 3068, are added. However, the results are different. The enhanced loop closures facilitates the graph to reach a precision of 8.01 cm, while the pseudo loop closure get a 69.03 cm result. Thus, enhanced loop closures could provide stronger control and more significant optimization than pseudo loop closures.

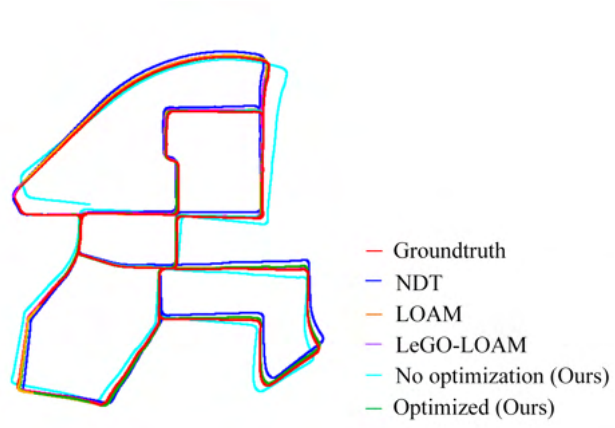


Figure 6.10: Trajectory results of the proposed algorithm and some state-of-the-art algorithms

Table 6.5: Experimental results compared with SOTA algorithms on the KITTI dataset

	NDT[176]	LOAM[171]	LeGO-LOAM[131]	No optimization	Optimized (Ours)
ATE(cm)	9.87	**2.32**	10.68	14.98	4.85

The optimization method is also compared with some state-of-the-art odometry and mapping algorithms on KITTI 00 dataset. The trajectory results are shown in Fig. 6.10 and the ATE results are listed in Tab. 6.5. LOAM [171] as a milestone in LiDAR-based odometry and mapping algorithm achieves the best results, while our

optimization method also get a comparable ATE result.

6.6 Limitations and Discussions

The discussion is summarized as below:

- Main sources of error of LiDAR-SLAM data collection and processing include observation errors of sensors, the motion error of mobile platforms, the point cloud registration error, false loops introduced by LCD, optimization errors, and the errors brought by other point cloud processing steps, like down-sampling, ground segmentation, and normal computation.

- Generally speaking, the point cloud registration algorithms and the odometry strategy play the dominant role in SLAM localization accuracy. As for the quality of map results, localization accuracy and point cloud quality are two significant factors. The point cloud quality includes the precision and accuracy of the laser measurement, the noise of the laser point, the stability, and the homogeneity of the laser scanning, etc.

- Affected by the measurement distance, those errors would accumulate and ruin the localization and mapping results. Before optimization, the error accumulates to a maximum level when the ending point is reached.

- The proposed weighted strategy is better than identify-weight, which means that we set slightly higher weights to loop closure edges will enhance the optimization ability of the graph.

- The precision of the loop closure edges is also significant. A reliable and accurate point cloud scan registration method should be used to obtain the pose transformation information.

- The pseudo loop closures and enhanced loop closures both could enhance the optimization performance significantly, while the costs should be balanced because the number of loop closure edges will grows sharply to make the graph dense and redundant. Thus, the pseudo loop gap and the enhanced loop buffer need to be adjusted according to the hardware computation power.

According to the optimization experimental results in multiple scenarios, some findings of graph optimization strategy based on LCD could be summarized:

- If a measurement environment is globally controlled by loop closures, which means the starting point and ending point are at the same place. The proposed graph optimization based on LCD would be effective. The middle part of the trajectory will obtain the largest error.

- If a measurement environment is partially controlled by loop closures, which means the ending point doesn't go back to the starting point. The suspended part of the trajectory is not be optimized. Thus, the ending point returning to the starting point is suggested to provide globally control information for mapping.

- Loops formed by reciprocating measurements on the same path are not able to provide effective information for optimization. Not only does it not even optimize the cumulative error, but it makes the point cloud thickness larger.

- Optimization based on LCD could reduce or even eliminate cumulative error when effective, sufficient, and precise loops are used. However, if the false loops are introduced or there are suspended trajectories with a loop, optimization performance will be limited or even lose effects.

Although the optimization methods is effective, limitations still exist. As shown in Fig. 6.11, some problems arise in the in-house outdoor dataset. The shown trajectory

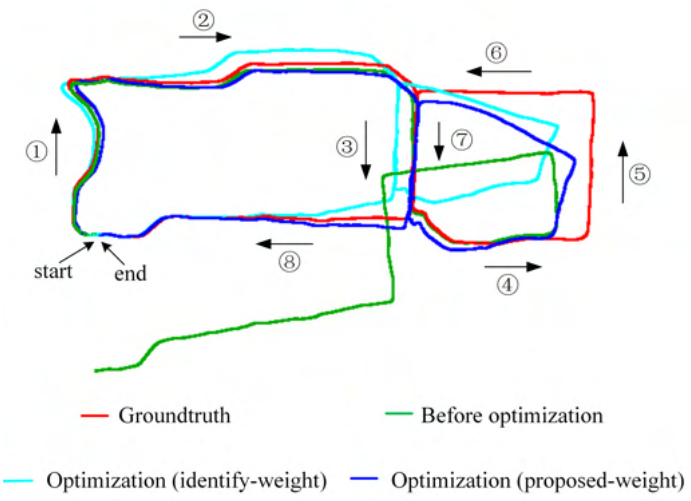


Figure 6.11: The limitation of LO optimization based on LCD (*in-house outdoor datasets*).

travels in the order of the marked numbers. The trajectory drifts severely, especially in the trajectory ④→⑤→⑥→⑦→ end, because the drift error is cumulative and increases sharply with the distance growing. Even the loops are used, the pose in trajectory ⑤ could not be corrected effectively. Thus, the pure LiDAR solutions, using LiDAR odometry, LCD and graph optimization may not provide robust and reliable localization information. Other positioning and orientation information provided by GNSS or INS then could be utilized to enhance the performance. GNSS could provide positioning information directly in outdoor open area, while INS could provide high-frequency orientation information and initialization values for odometry.

Thus, based on the experiments on factors of optimization performance and limitation discussions, some guidance about mobile mapping backpack fieldwork measurement and data processing could be presented.

- Loop is essential for cumulative error elimination. In fieldwork measurement, it is highly advisable to build loop closures. Especially, when the measurement scenario contains multiple independent sub-scenes, at least one loop closure

needs to be constructed in each sub-scenes. The trajectory shown lemniscate shape is recommended.

- When planning the measurement path, the ending point of the measurement trajectory is better to return to the starting point. This provides global optimization to the optimization.

- A correct loop closure could facilitate the LiDAR odometry and mapping, while a wrong loop or edge constraint with low precision might ruin the whole trajectory and map. Thus, the correctness and the precision of the LCD algorithm is more important than the recall rate of LCD algorithms.

- The precision of loop scans registration has a significant impact on the results. Thus, an effective and precise point cloud registration algorithm should be adopted.

- Pure LiDAR-SLAM framework is less precise and robust than SLAM based on multiple sensors integration. IMU and GNSS should be fused into the solution to provide sustained positioning and attitude information.

- The proposed LCD algorithms detect the LiDAR scans with high similarity thus the detect results suffer from highly similar and repetitive environments. Some strategies should be taken, such as two scans of a loop candidate should be close in space distance and far in a time interval. The space distance could be obtained from front-end odometry, while time interval could be computed by timestamps of each LiDAR scan.

6.7 Conclusions

In this chapter, an enhanced graph optimization approach is utilized to reduce the cumulative error of LiDAR odometry in SLAM. The weight is calculated based on the

assumption that the loop closure edges with higher registration precision will improve optimization performance more. Thus, they will obtain heavy weights. Qualitative experiments are performed on open-source datasets and in-house datasets to exhibit the effectiveness of the LCD results and optimization. Besides, Loop closure edges are divided into three types, detected loops, pseudo loops, and enhanced loops. A study of factors of loops affecting optimization performance is investigated. The factors include the values of weights, the precision of loop closure edges, the number of pseudo loops, and enhanced loops. Conclusions are reached according to quantitative comparison experiments on open-source datasets. Then, some guidance on fieldwork measurement and data processing is given. In the future, more strategies will be researched to make the optimization more robust and reliable.

Chapter 7

Conclusion

7.1 Contributions

To summarize, this thesis focuses on fast and lightweight LCD and optimization for LiDAR-SLAM. The contributions of this thesis are:

- A fast and compact LCD method is proposed based on comprehensive descriptors and machine learning. Comprehensive descriptors are encoded by discriminative multi-modality features to describe each laser scan, including statistics, geometry, planar features, intensity, and ranging distance. RF model is used as a binary classifier to detect loop closure candidates. Then, a novel double-deck loop candidate verification strategy is used to reject false positives. This method is dedicated to solving LCD in indoor or human-made structure scenes.

- As for the outdoor large-scale environments, the point cloud does not exhibit significant structural and regular geometric characteristics. Thus, a very deep and super lightweight neural network DeLightLCD is proposed to enable highly efficient loop closure detection in large-scale environments. Depth-wise separable convolution (DSC) and batch normalization (BN) are utilized to ensure

that the network is lightweight and trainable.

- DeLightLCD++ is an improved LCD algorithm to address some practical problems of sensor alteration and environmental changes. A novel data presentation method is used to reduce data loss and elimination the effects of environmental changes. The architecture of the network is also adjusted to ensure that the algorithm is rotation invariant. Besides, a loop candidate fast search method is used to suppress the computation cost increase for ultra-long measurement distance.

- After loop closures are detected, the loop closures will be utilized for pose optimization. An enhanced graph optimization strategy is used. We introduced three types of loop closures: detected loops, pseudo loops, and enhanced loops. Then, factors affecting optimization performance are studied, including, weights, the precision of loop closure edges, the number of pseudo loop closures, and the number of enhanced loop closures. Finally, some guidance is given on fieldwork and data processing of the mobile mapping backpack system.

- A mobile mapping backpack equipped with two multi-line laser scanners was designed to collect point cloud data and test the performance of the proposed methods.

7.2 Discussions

The discussions of every proposed algorithm have been provided in each chapter. In this section, we would like to discuss LCD and SLAM research in a broader view.

- Both LiDAR sensors and depth cameras (RGBD-cameras) could obtain ranging distance information and generate point cloud data. There are many differences in measurement theory between the two sensors, thus the applicable scenes,

data processing algorithms, and data results are all different. Depth cameras are usually limited by natural light conditions and measurement distance, while LiDAR sensors are not affected by illumination changes and have relatively long measurement distances. As for the data processing, LiDAR-based SLAM results usually have better precision than visual-based SLAM. Depth cameras could capture texture information with real color, while LiDAR data only collected laser point data with intensity.

- Visual sensors could capture abundant texture information. The sufficient information of image data could compensate for the defects of sparse point cloud data and no texture information collected. However, image data could not obtain precise ranging information like LiDAR data. Thus, visual sensors could be fused with LiDAR sensors for an improved LCD approach. More broadly, some other tasks based on point clouds, like object detection, semantic segmentation, and instance segmentation, could be facilitated with the image data.

- As for the information adopted to solve LCD and SLAM tasks using point cloud data, some high-level features, such as semantic information, object labels, and spatial relationships, could be used to facilitate LCD and SLAM problems. Factually, some researchers have tried in this research direction[177, 118]. However, the efficiency of semantic segmentation or object detection should be considered.

7.3 Limitations and Open Problems

Admittedly, the research and analysis still have limitations. The limitations and future works are listed below:

- For FastLCD, it is limited in indoor environments due to the features used. Thus, some novel and environmental-insensitive features should be researched for extending the method to outdoor and large-scale scenes.

- To address the problems of FastLCD, DeLightLCD is dedicated to LCD in outdoor large-scale scenes. However, the data presentation of DeLightLCD limits the input data format. Thus, the generalization performance on other LiDAR sensors is restricted. A new data presentation method should be studied to reduce data loss and ignore sensor changes.

- Although both FastLCD and DeLightLCD are highly time-efficient, with measurement distance increasing, the time cost will grow sharply. They ignore integrating an efficient search strategy.

- DeLightLCD++ is robust and insensitive to sensor changes and environmental changes. However, the parameter amount grows resulting in time cost increases.

7.4 Future Works

- Although the LCD algorithm should be independent without prior pose knowledge, LCD, as the subsequent step after LO, could use the features extracted in LO. Then, the computation efficiency might be further improved.

- Some high dimensional and advanced features designed artificially or learned by deep learning models will be further studied and applied in LCD and SLAM. Especially, semantic information, as common information in urban environments will be researched to improve the algorithm performance.

- The enhanced graph optimization strategy shows effective performance. However, the convergence depends on the performance of front-end odometry and the precision of loop closure edges. A poor front-end odometry might need a large number of iterations. Thus, low-drift, precise and robust LiDAR odometry algorithms will be researched in the next stage.

- The pure LiDAR-SLAM solution with LiDAR odometry, LCD, and optimization may not be very robust to different scenes. Multiple sensor fusion would enhance the performance and make the solution reliable and robust. Thus, a SLAM solution integrated with LiDAR, visual sensors, GNSS, IMU, and other technologies would be researched in the future.

References

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[2] Aitor Aldoma, Federico Tombari, Radu Bogdan Rusu, and Markus Vincze. Our-cvfh – oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation. In Axel Pinz, Thomas Pock, Horst Bischof, and Franz Leberl, editors, *Pattern Recognition*, pages 113–122, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[3] Yasin Almalioglu, Muhamad Risqi U. Saputra, Pedro P. B. de Gusm?o, Andrew Markham, and Niki Trigoni. Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5474–5480, 2019.

[4] Adrien Angeli, David Filliat, St?phane Doncieux, and Jean-Arcady Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, 24(5):1027–1037, 2008.

[5] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust amp; efficient point cloud registration using pointnet. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7156–7165, 2019.

[6] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016.

[7] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *CoRR*, abs/1803.10091, 2018.

[8] Haris Baltzakis, Antonis Argyros, and Panos Trahanias. Fusion of laser and visual data for robot motion planning and collision avoidance. *Machine Vision and Applications*, 15(2):92–100, 2003.

[9] Xicheng Ban, Hongjian Wang, Tao Chen, Ying Wang, and Yao Xiao. Monocular visual odometry based on depth and optical flow using deep learning. *IEEE Transactions on Instrumentation and Measurement*, 70:1–19, 2021.

[10] Jens Behley and Cyrill Stachniss. Efficient surfel-based slam using 3d laser range data in urban environments. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[11] Igor Bogoslavskyi and Cyrill Stachniss. Fast range image-based segmentation of sparse 3d laser scans for online operation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 163–169, 2016.

[12] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[13] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[14] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[15] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 778–792, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[16] Luca Carlone, Rosario Aragues, José A Castellanos, and Basilio Bona. A linear approximation for graph-based simultaneous localization and mapping. In *Robotics: Science and Systems*, volume 7, pages 41–48, 2012.

[17] J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardos. The spmap: a probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–952, 1999.

[18] José A. Castellanos, José Neira, and Juan D. Tardós. Limits to the consistency of ekf-based slam. *IFAC Proceedings Volumes*, 37(8):716–721, 2004. IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 5-7 July 2004.

[19] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.

[20] Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. Selective sensor fusion for neural visual-inertial odometry. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10534–10543, 2019.

[21] Pengxin Chen, Wenzhong Shi, Wenzheng Fan, Haodong Xiang, and Sheng Bao. Rectmatch: A novel scan matching method using the rectangle-flattening representation for mobile lidar systems. *ISPRS Journal of Photogrammetry and Remote Sensing*, 180:191–208, 2021.

[22] Shilang Chen, Junjun Wu, Yanran Wang, Lin Zhou, Qinghua Lu, and Yunzhi Zhang. Robust loop-closure detection with a learned illumination invariant representation for robot vslam. In *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 342–347, 2019.

[23] Siheng Chen, Chaojing Duan, Yaoqing Yang, Duanshun Li, Chen Feng, and Dong Tian. Deep unsupervised learning of 3d point clouds via graph topology inference and filtering. *IEEE Transactions on Image Processing*, 29:3183–3198, 2020.

[24] Steven W. Chen, Guilherme V. Nardari, Elijah S. Lee, Chao Qu, Xu Liu, Roseli Ap. Francelin Romero, and Vijay Kumar. Sloam: Semantic lidar odometry and mapping for forest inventory. *IEEE Robotics and Automation Letters*, 5(2):612–619, 2020.

[25] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss. SuMa++: Efficient LiDAR-based Semantic SLAM. In *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[26] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Jens Behley, and Cyrill Stachniss. Overlapnet: a siamese network for computing lidar scan similarity with applications to loop closing and localization. *Autonomous Robots*, pages 1573–7527, 2021.

[27] Younggun Cho, Giseop Kim, and Ayoung Kim. Deeplo: Geometry-aware deep lidar odometry. *CoRR*, abs/1902.10562, 2019.

[28] Younggun Cho, Giseop Kim, and Ayoung Kim. Unsupervised geometry-aware deep lidar odometry. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2145–2152, 2020.

[29] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2017.

[30] Fran?ois Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017.

[31] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2652–2660, 2017.

[32] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), 02 2017.

[33] Laura A Clemente, Andrew J Davison, Ian D Reid, José Neira, and Juan D Tardós. Mapping large loops with a single hand-held camera. In *Robotics: Science and Systems*, volume 2, 2007.

[34] Gabriele Costante and Thomas Alessandro Ciarfuglia. Ls-vo: Learning dense optical subspace for robust visual odometry estimation. *IEEE Robotics and Automation Letters*, 3(3):1735–1742, 2018.

[35] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[36] Mark Cummins and Paul Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.

[37] Robert Cupec, Damir Filko, and Emmanuel Karlo Nyarko. Place recognition based on planar surfaces using multiple rgb-d images taken from the same position. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–8, 2019.

[38] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 209–216, New York, NY, USA, 2007. Association for Computing Machinery.

[39] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[40] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.

[41] ML Doaa, A Mohammed, Megeed Salem, H Ramadan, and Mohamed I Roushdy. Comparison of optimization techniques for 3d graph-based slam. *Recent Advances in Information Science*, 2013.

[42] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2017.

[43] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip H?usser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015.

[44] Renaud Dubé, Daniel Dugas, Elena Stumm, Juan Nieto, Roland Siegwart, and Cesar Cadena. Segmatch: Segment based place recognition in 3d point clouds.

In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5266–5272, 2017.

[45] Hugh Durrant-Whyte, Somajyoti Majumder, Sebastian Thrun, Marc de Battista, and Steve Scheding. A bayesian algorithm for simultaneous localisation and map building. In Raymond Austin Jarvis and Alexander Zelinsky, editors, *Robotics Research*, pages 49–60, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[46] Wenzheng Fan, Wenzhong Shi, Haodong Xiang, and Ke Ding. A novel method for plane extraction from low-resolution inhomogeneous point clouds and its application to a customized low-cost mobile mapping system. *Remote Sensing*, 11(23), 2019.

[47] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.

[48] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[49] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[50] C. Fröhlich and Markus Mettenleiter. Terrestrial laser scanning-new perspectives in 3d surveying. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36:7–13, 01 2004.

[51] Dorian Galvez-López and Juan D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[52] Diogo C. Garcia, Tiago A. Fonseca, Renan U. Ferreira, and Ricardo L. de Queiroz. Geometry coding for dynamic voxelized point clouds using octrees and multiple contexts. *IEEE Transactions on Image Processing*, 29:313–322, 2020.

[53] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[54] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[55] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[56] Karl Granstrm̈ and Thomas B. Schön. Learning to close the loop from 3d point clouds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2089–2095, 2010.

[57] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.

[58] J.E. Guivant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.

[59] Yunhui Guo, Yandong Li, Liqiang Wang, and Tajana Rosing. Depthwise convolution is all you need for learning multiple visual domains. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8368–8375, 07 2019.

[60] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5353–5360, 2015.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[62] Li He, Xiaolong Wang, and Hong Zhang. M2dp: A novel 3d point cloud descriptor and its application in loop closure detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 231–237, 2016.

[63] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[64] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278, 2016.

[65] Marian Himstedt, Jan Frost, Sven Hellbach, Hans-Joachim B?hme, and Erik Maehle. Large scale place recognition in 2d lidar scans using geometrical landmark relations. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5030–5035, 2014.

[66] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[67] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(8):2011?2023, August 2020.

[68] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[69] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07 2015. PMLR.

[70] An?ela Juri?, Filip Kende?, Ivan Markovi?, and Ivan Petrovi? A comparison of graph optimization approaches for pose estimation in slam. In *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 1113–1118, 2021.

[71] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *2011 IEEE International Conference on Robotics and Automation*, pages 3281–3288, 2011.

[72] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. isam: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[73] I Kalisperakis, T Mandilaras, A El Saer, P Stamatopoulou, C Stentoumis, S Bourou, and L Grammatikopoulos. A modular mobile mapping platform for complex indoor and outdoor environments. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:243–250, 2020.

[74] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809, 2018.

[75] Jong-Hyuk Kim and S. Sukkarieh. Airborne simultaneous localisation and map building. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 1, pages 406–411 vol.1, 2003.

[76] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

[77] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[78] Kenji Koide, Jun Miura, and Emanuele Menegatti. A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, 16(2):1729881419841532, 2019.

[79] Kurt Konolige and Motilal Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

[80] Kurt Konolige, Giorgio Grisetti, Rainer Kümmerle, Wolfram Burgard, Benson Limketkai, and Regis Vincent. Efficient sparse pose adjustment for 2d mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 22–29, 2010.

[81] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G¡sup¿2¡/sup¿o: A general framework for graph optimization.

In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.

[82] Mathieu Labbé and Francois Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3):734–745, 2013.

[83] Mathieu Labbé and Francois Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666, 2014.

[84] Truc Le and Ye Duan. Pointgrid: A deep network for 3d shape understanding. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9204–9214, 2018.

[85] Seongwon Lee, HyungGi Jo, Hae Min Cho, and Euntai Kim. Visual loop closure detection over illumination change. In *2019 16th International Conference on Ubiquitous Robots (UR)*, pages 77–80, 2019.

[86] Huan Lei, Naveed Akhtar, and Ajmal Mian. Octree guided cnn with spherical kernels for 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[87] J Leonard. Mobile robot lacalization by tracking geometric beacons. *IEEE Trans. Robot. Autom.*, 7(3):89–97, 1991.

[88] John L Leonard, Robert N Carpenter, and Hans Jacob S Feder. Stochastic mapping using forward look sonar. *Robotica*, 19(5):467–480, 2001.

[89] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual?inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[90] Jiaxin Li, Huangying Zhan, Ben M. Chen, Ian Reid, and Gim Hee Lee. Deep learning for 2d scan matching and loop closure. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 763–768, 2017.

[91] Lin Li, Xin Kong, Xiangrui Zhao, Wanlong Li, Feng Wen, Hongbo Zhang, and Yong Liu. Sa-loam: Semantic-aided lidar slam with loop closure. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7627–7634, 2021.

[92] Qing Li, Shaoyang Chen, Cheng Wang, Xin Li, Chenglu Wen, Ming Cheng, and Jonathan Li. Lo-net: Deep real-time lidar odometry. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8465–8474, 2019.

[93] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7291, 2018.

[94] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[95] Zhichao Li and Naiyan Wang. Dmlo: Deep matching lidar odometry. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6010–6017, 2020.

[96] Mingjie Liang, Huaqing Min, and Ronghua Luo. Graph-based slam: a survey. *Robot*, 35(4):500–512, 2013.

[97] Jiarong Lin and Fu Zhang. Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3126–3131, 2020.

[98] Qi Liu, Hui Yuan, Honglei Su, Hao Liu, Yu Wang, Huan Yang, and Junhui Hou. Pqa-net: Deep no reference point cloud quality assessment via multi-view projection. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(12):4645–4660, 2021.

[99] Yang Liu and Hong Zhang. Indexing visual features: Real-time loop closure detection using a tree structure. In *2012 IEEE International Conference on Robotics and Automation*, pages 3613–3618, 2012.

[100] Zhe Liu, Chuanzhe Suo, Shunbo Zhou, Fan Xu, Huanshu Wei, Wen Chen, Hesheng Wang, Xinwu Liang, and Yun-Hui Liu. Seqlpd: Sequence matching enhanced loop-closure detection based on large-scale point cloud description for self-driving vehicles. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1218–1223, 2019.

[101] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yunhui Liu. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2831–2840, 2019.

[102] Ren C. Luo, Vincent W.S. Ee, and Chung-Kai Hsieh. 3d point cloud based indoor mobile robot in 6-dof pose localization using fast scene recognition and alignment approach. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 470–475, 2016.

[103] Martin Magnusson, Henrik Andreasson, Andreas Nüchter, and Achim J Lilienthal. Automatic appearance-based loop detection from three-dimensional laser

data using the normal distributions transform. *Journal of Field Robotics*, 26(11-12):892–914, 2009.

[104] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.

[105] Azam Rafique Memon, Hesheng Wang, and Abid Hussain. Loop closure detection using supervised and unsupervised deep neural networks for monocular slam systems. *Robotics and Autonomous Systems*, 126:103470, 2020.

[106] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220, 2019.

[107] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. volume 593598, page 593?598, 2002.

[108] Gong Bo Moon, Sebum Chun, Moon-Beom Hur, and Gyu-In Jee. A robust indoor positioning system using two-stage ekf slam for first responders in an emergency environment. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pages 707–711, 2013.

[109] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[110] Megan R Naminski. An analysis of simultaneous localization and mapping (slam) algorithms. *Mathematics, Statistics, and Computer Science Honors Projects Paper*, 2013.

[111] Thien-Minh Nguyen, Shenghai Yuan, Muqing Cao, Lyu Yang, Thien Hoang Nguyen, and Lihua Xie. Miliom: Tightly coupled multi-input lidar-inertia odometry and mapping. *IEEE Robotics and Automation Letters*, 6(3):5573–5580, 2021.

[112] Austin Nicolai, Ryan Skeele, Christopher Eriksen, and Geoffrey A Hollinger. Deep learning for laser based odometry estimation. In *RSS workshop Limits and Potentials of Deep Learning in Robotics*, volume 184, page 1, 2016.

[113] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2262–2269, 2006.

[114] G. Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C. Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[115] Gaurav Pandey, James R McBride, and Ryan M Eustice. Ford campus vision and lidar data set. *The International Journal of Robotics Research*, 30(13):1543–1552, 2011.

[116] L.M. Paz, P. Jensfelt, J.D. Tardos, and J. Neira. Ekf slam updates in o(n) with divide and conquer slam. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1657–1663, 2007.

[117] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft, April 1998.

[118] Georgi Pramatarov, Daniele De Martini, Matthew Gadd, and Paul Newman. Boxgraph: Semantic place recognition and pose estimation from 3d lidar, 2022.

[119] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[120] Hao Qin, May Huang, Jian Cao, and Xing Zhang. Loop closure detection in slam by combining visual cnn features and submaps. In *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, pages 426–430, 2018.

[121] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[122] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[123] Timo Röhling, Jennifer Mack, and Dirk Schulz. A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 736–741, 2015.

[124] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[125] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 430–443, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[126] Staiger Rudolf. Terrestrial laser scanning technology, systems and applications. In *2003 Second FIG Regional Conference*, 2003.

[127] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[128] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009.

[129] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Learning informative point classes for the acquisition of object model maps. In *2008 10th International Conference on Control, Automation, Robotics and Vision*, pages 643–650, 2008.

[130] Jie Shan and Charles K Toth. *Topographic laser ranging and scanning: principles and processing*. CRC press, 2018.

[131] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765, 2018.

[132] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142, 2020.

[133] Tixiao Shan, Brendan Englot, Carlo Ratti, and Daniela Rus. Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5692–5698, 2021.

[134] Weijing Shi, Mohamed Baker Alawieh, Xin Li, and Huafeng Yu. Algorithm and hardware implementation for visual perception system in autonomous vehicle: A survey. *Integration*, 59:148–156, 2017.

[135] Wenzhong Shi, Wael Ahmed, Na Li, Wenzheng Fan, Haodong Xiang, and Muyang Wang. Semantic geometric modelling of unstructured indoor point cloud. *ISPRS International Journal of Geo-Information*, 8(1), 2019.

[136] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[137] Randall C Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56–68, 1986.

[138] Hyunjin Son, Byungjin Lee, and Sangkyung Sung. Synthetic deep neural network design for lidar-inertial odometry based on cnn and lstm. *International Journal of Control, Automation and Systems*, 19(8):2859–2868, 2021.

[139] Jochen Sprickerhof, Andreas Nüchter, Kai Lingemann, and Joachim Hertzberg. A heuristic loop closing technique for large-scale 6d slam. *Automatika*, 52(3):199–222, 2011.

[140] Bastian Steder, Giorgio Grisetti, and Wolfram Burgard. Robust place recognition for 3d range data based on point features. In *2010 IEEE International Conference on Robotics and Automation*, pages 1400–1405, 2010.

[141] Bastian Steder, Michael Ruhnke, Slawomir Grzonka, and Wolfram Burgard. Place recognition in 3d scans using a combination of bag of words and point feature based relative pose estimation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1249–1255, 2011.

[142] Francesco Di Stefano, Stefano Chiappini, Alban Gorreja, Mattia Balestra, and Roberto Pierdicca. Mobile 3d scan lidar: a literature review. *Geomatics, Natural Hazards and Risk*, 12(1):2387–2429, 2021.

[143] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 12 2015.

[144] Niko Sünderhauf. *Robust optimization for simultaneous localization and mapping.* PhD thesis, Technischen Universitat Chemnitz, 2012.

[145] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[146] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 International Conference on 3D Vision (3DV)*, pages 537–547, 2017.

[147] Sebastian Thrun et al. Robotic mapping: A survey. 2002.

[148] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2018.

[149] Martin Velas, Michal Spanel, Michal Hradis, and Adam Herout. Cnn for imu assisted odometry estimation using velodyne lidar. In *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 71–77, 2018.

[150] Cheng Wang, Shiwei Hou, Chenglu Wen, Zheng Gong, Qing Li, Xiaotian Sun, and Jonathan Li. Semantic line framework-based indoor building modeling

using backpacked laser scanning point cloud. *ISPRS Journal of Photogrammetry and Remote Sensing*, 143:150–166, 2018.

[151] Cheng Wang, Chenglu Wen, Yudi Dai, Shangshu Yu, and Minghao Liu. Urban 3d modeling with mobile laser scanning: a review. *Virtual Reality & Intelligent Hardware*, 2(3):175–212, 2020. 3D Visual Processing and Reconstruction Special Issue.

[152] Han Wang, Chen Wang, Chun-Lin Chen, and Lihua Xie. F-loam : Fast lidar odometry and mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4390–4396, 2021.

[153] Han Wang, Chen Wang, and Lihua Xie. Intensity scan context: Coding intensity and geometry relations for loop closure detection. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2095–2101, 2020.

[154] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050, 2017.

[155] Wei Wang, Muhamad Risqi U. Saputra, Peijun Zhao, Pedro Gusmao, Bo Yang, Changhao Chen, Andrew Markham, and Niki Trigoni. Deeppco: End-to-end point cloud odometry through deep parallel neural network. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3248–3254, 2019.

[156] Ying Wang, Zezhou Sun, Cheng-Zhong Xu, Sanjay E. Sarma, Jian Yang, and Hui Kong. Lidar iris for loop-closure detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5769–5775, 2020.

[157] Chenglu Wen, Yudi Dai, Yan Xia, Yuhan Lian, Jinbin Tan, Cheng Wang, and Jonathan Li. Toward efficient 3-d colored mapping in gps-/gnss-denied environments. *IEEE Geoscience and Remote Sensing Letters*, 17(1):147–151, 2020.

[158] Brian Williams, Mark Cummins, Jose Neira, Paul Newman, Ian Reid, and Juan Tardos. An image-to-map loop closing method for monocular slam. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2053–2059, 2008.

[159] Brian Williams, Mark Cummins, José Neira, Paul Newman, Ian Reid, and Juan Tardós. A comparison of loop closing techniques in monocular slam. *Robotics and Autonomous Systems*, 57(12):1188–1197, 2009. Inside Data Association.

[160] Brian Williams, Georg Klein, and Ian Reid. Automatic relocalization and loop closing for real-time monocular slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1699–1712, 2011.

[161] Walter Wohlkinger and Markus Vincze. Ensemble of shape functions for 3d object classification. In *2011 IEEE International Conference on Robotics and Biomimetics*, pages 2987–2992, 2011.

[162] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. Soe-net: A self-attention and orientation encoding network for point cloud based place recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11343–11352, 2021.

[163] Haodong Xiang, Wenzhong Shi, Wenzheng Fan, Pengxin Chen, Sheng Bao, and Mingyan Nie. Fastlcd: A fast and compact loop closure detection approach using 3d point cloud for indoor mobile mapping. *International Journal of Applied Earth Observation and Geoinformation*, 102:102430, 2021.

[164] Haodong Xiang, Xiaosheng Zhu, Wenzhong Shi, Wenzheng Fan, Pengxin Chen, and Sheng Bao. Delightlcd: A deep and lightweight network for loop closure detection in lidar slam. *IEEE Sensors Journal*, pages 1–1, 2022.

[165] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. FAST-LIO2: fast direct lidar-inertial odometry. *CoRR*, abs/2107.06829, 2021.

[166] Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.

[167] Huan Yin, Xiaqing Ding, Li Tang, Yue Wang, and Rong Xiong. Efficient 3d lidar based loop closing using deep neural network. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 481–486, 2017.

[168] Masashi Yokozuka, Kenji Koide, Shuji Oishi, and Atsuhiko Banno. Litamin: Lidar-based tracking and mapping by stabilized icp for geometry approximation with normal distributions. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5143–5150, 2020.

[169] Masashi Yokozuka, Kenji Koide, Shuji Oishi, and Atsuhiko Banno. Litamin2: Ultra light lidar-based SLAM using geometric approximation applied with kl-divergence. *CoRR*, abs/2103.00784, 2021.

[170] Wei Zeng and Theo Gevers. 3dcontextnet: K-d tree guided hierarchical learning of point clouds using local and global contextual cues. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[171] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, 2014.

[172] Min Zhang and Wenzhong Shi. A feature difference convolutional neural network-based change detection method. *IEEE Transactions on Geoscience and Remote Sensing*, 58(10):7232–7246, 2020.

[173] Wenxiao Zhang and Chunxia Xiao. Pcan: 3d attention map learning using contextual information for point cloud based retrieval. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12428–12437, 2019.

[174] Pengwei Zhou, Xuexun Guo, Xiaofei Pei, and Ci Chen. T-loam: Truncated least squares lidar-only odometry and mapping in real time. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.

[175] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[176] Zhicheng Zhou, Cheng Zhao, Daniel Adolfsson, Songzhi Su, Yang Gao, Tom Duckett, and Li Sun. Ndt-transformer: Large-scale 3d point cloud localisation using the normal distribution transform representation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5654–5660, 2021.

[177] Yachen Zhu, Yanyang Ma, Long Chen, Cong Liu, Maosheng Ye, and Lingxi Li. Gosmatch: Graph-of-semantics matching for detecting loop closures in 3d lidar data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5151–5157, 2020.

[178] Yan Zhuang, Nan Jiang, Huosheng Hu, and Fei Yan. 3-d-laser-based scene measurement and place recognition for mobile robots in dynamic indoor environments. *IEEE Transactions on Instrumentation and Measurement*, 62(2):438–450, 2013.

[179] Robert Zlot and Michael Bosse. Place recognition using keypoint similarities in 2d lidar maps. In Oussama Khatib, Vijay Kumar, and George J. Pappas, editors, *Experimental Robotics*, pages 363–372, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.