3D Mapping by Multi-Sensor Fusion for Construction Cranes (建設用クレーンのためのマルチセンサ融合による3次元地図生成)

January, 2024

Doctor of Philosophy (Engineering)

Mahmood UL Hassan
（マフムード ウル ハッサン）

Toyohashi University of Technology

別紙３

PUBLICATION LIST
論　文　目　録

Date:
2024 年 01 月30 日

Name: Mahmood Ul Hassan
氏名　マフムード ウル ハッサン

List of Papers with Referee's Review
査読付学術論文

1. Mahmood Ul. Hassan, D. Das and J. Miura, "3D Mapping for a Large Crane Using Rotating 2D-Lidar and IMU Attached to the Crane Boom," in IEEE Access, doi: 10.1109/ACCESS.2023.3250248, vol. 11, pp. 21104-21116, 2023.
2. Mahmood Ul. Hassan, and J. Miura, "Sensor Pose Estimation and 3D Mapping for Crane Operations Using Sensors Attached to the Crane Boom," in IEEE Access, doi: 10.1109/ACCESS.2023.3307197, vol. 11, pp. 90298-90308, 2023.

List of Papers at International Conference with Referee's Review
査読付国際会議論文

1. Mahmood Ul Hassan and J. Miura, "Neural Network-Based Real-Time Odometry Using IMU for Crane System and Its Application to Large-Scale 3D Mapping," IEEE International Conference on Mechatronics and Automation (ICMA), doi: 10.1109/ICMA54519.2022.9856381, Guilin, Guangxi, China, pp. 1062-1068, 2022.

I confirm that all co-authors of the listed papers have agreed with the applicant to use them for the degree application.
申請者が上記の論文を学位申請に用いることについて, 共著者全員の同意を得ています。

Date:
2024 年　01　月　30　日

Name: Mahmood Ul Hassan
氏名　マフムード ウル ハッサン

Presentation at International Conference
国際会議発表　※

Mahmood Ul Hassan and J. Miura, "Neural Network-Based Real-Time Odometry Using IMU for Crane System and Its Application to Large-Scale 3D Mapping," IEEE International Conference on Mechatronics and Automation (ICMA), doi: 10.1109/ICMA54519.2022.9856381, Guilin, Guangxi, China, pp. 1062-1068, 2022.
.

Note 1: Sort by types of paper, and list in chronological order with reference numbers.
学術論文と国際会議論文に分け, 発表年次の古い順に番号を付して記入すること。

Note 2: Write names of all authors, title, title of journals, volume, pages (first and last), and year of publication.
著者名（全員記入, 申請者名に下線）, 発表論文名, 掲載誌名, 巻号, ページ(最初と最終), 発表年（西暦）の順に記入すること。

※Write a representative presentation which you did by yourselves for confirmation your English skill.
英語能力の確認のため、自身が行った代表的な国際会議発表を１件記入すること。

別紙４－１（課程博士（英文））

Date of Submission（month day, year）: 01, 30, 2024

Department of Computer Science & Engineering		Student ID Number	D209301
Applicant’s name	Mahmood Ul Hassan		

Supervisors	Jun Miura, Naoki Uchiyama

Abstract（Doctor）

Title of Thesis	3D Mapping by Multi-Sensor Fusion for Construction Cranes

Approx. 800 words

The creation of a 3D map holds significant importance for autonomous systems navigating in unknown environments. The application of 3D mapping spans various fields, including autonomous driving, service robotics, agriculture, augmented reality, and construction. The need for efficient 3D mapping methods is growing with the proliferation of robots and autonomous systems.

Despite extensive research on 3D mapping, particularly for ground vehicles and drones, limited attention has been given to 3D mapping for construction cranes. Current 3D mapping methods encounter limitations when applied to constructing a map for cranes, primarily due to the unique challenges and complexities associated with crane mapping. There is a strong need for innovative approaches specifically tailored for mapping in crane scenarios. This thesis proposes three novel methods for mapping for a construction crane using multi-sensor fusion.

First, we propose a complementary filter and crane structure-based real-time sensor pose estimation and 3D mapping method for construction cranes with an arbitrary motion of the sensor system (2D lidar and IMU) attached to the crane boom. A heavy lidar with a slowly rotating base is needed to make a large-scale map both vertically and horizontally for cranes. In the proposed method, we introduce a complementary filter with moving average filtering for lidar pose estimation, which is more robust to severe vibration than Kalman filter-based methods. As there are only a small amount of overlaps between 2D lidar scans, we propose a map correction method based on pose graph optimization with planar environmental constraints. We evaluate the proposed method in a simulation and a real environment and compare it with one of the state-of-the-art methods. The evaluation results reveal that the proposed method can accurately estimate the sensor poses, thereby generating a high-quality, large-scale 3D point cloud map.

Second, we introduce a method for neural network-based real-time pose estimation using an IMU (inertial measurement unit) and its application in large-scale 3D mapping using a slowly rotating 2-D LiDAR. In this method, a neural network consisting of a convolutional neural network (CNN) and long short-term memory (LSTM) is employed to estimate the change in pose. Firstly, online pre-filtering using a low-pass filter is implemented on the time windows of IMU measurements before feeding them as input to the neural network to estimate the change in position and rotation of the sensor. After that, the estimated sensor pose is used to register the scans of 2D-rotating LiDAR to build a large-scale 3D map. The proposed method is tested in a gazebo environment by attaching the sensors to a crane boom. In this study, we also investigate the impact of different time windows of IMU measurements on the accuracy of pose estimation by the neural network.

Third, we proposed a method for sensor pose estimation, as well as creating large-scale 3D maps, for construction cranes equipped with a sensor system consisting of a camera, 2D lidar, and IMU. To tackle the challenges posed by the crane boom's complex motion, we utilize an Extended Kalman filter (EKF) to improve the accuracy and reliability of sensor pose estimation. By combining pose estimates from Visual-Inertial Navigation System (VINS) with data from an additional IMU, we estimate the scale value of a monocular camera. This scale value, obtained from the EKF, is then integrated into the VINS algorithm to refine the previously estimated scale value. Slowly rotating 2D lidar is used to build a 3D map. Since there is limited overlap between 2D lidar scans, we leverage the estimated pose to align and construct a comprehensive 3D map. Additionally, we thoroughly evaluate the effectiveness of the latest VINS techniques, as well as the EKF-enhanced VINS approach, in the specific context of crane operations. Through comprehensive performance assessments conducted in both simulated and real environments, we compare the EKF-added VINS method with state-of-the-art VINS techniques. The evaluation results demonstrate that the EKF-added VINS method accurately estimates sensor poses, leading to the generation of high-quality, large-scale 3D point cloud maps for construction cranes.

別紙６

Date of Submission:
2024 年　01　月　30　日

CURRICULUM VITAE
履　歴　書

Name ふりがな 氏名	Mahmood UL Hassan マフムード ウル ハッサン

Date of Birth and (Age) 生年月日（年齢）	1994 年 01 月 01 日生 （ 30 歳） year month date age	Nationality 本籍地 (都道府県)	Pakistan

Current Address 現住所	Room 206, Tophill2, 20-2, Sokuten, Akebono, Toyohashi, Aichi, 441-8151.

Academic History
学　歴

Year, month 年　月	History 事　項
2007 - 04 ~ 2011 - 04	College Degree at Ever Shine Public College, Mehrabpur Sindh Pakistan.
2012 - 01 ~ 2015 - 12	Bachelor of Engineering at Institute of Industrial Electronics Engineering, NED University of Engineering and Technology, Karachi Pakistan.
2017 - 09 ~ 2020 - 03	M.S at Department of Instrument Science and Technology, Shanghai Jiao Tong University, Shanghai China.
2020 - 09 ~ 2023 - 09	Ph.D at Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi Japan.

Employment History
職　歴

Year, month 年　月	History 事　項
2016 - 02 ~ 2017 - 07	Automation & Control Engineer at Rays Control Engineering Services, Karachi Pakistan.
2023 - 10 ~ Present	Researcher at Active Intelligent Systems Laboratory, Toyohashi University of Technology Toyohashi Japan.

3D Mapping by Multi-sensor Fusion for Construction Cranes

Mahmood Ul Hassan

Department of Computer Science and Engineering
Toyohashi University of Technology

Doctor of Philosophy (Engineering)

March 2024

Acknowledgements

I am deeply indebted to Professor Jun Miura for his unwavering support and invaluable guidance throughout my doctoral journey. Under his supervision, I have not only expanded my knowledge but also honed my skills as a professional researcher in robotics. His mentorship has equipped me with the ability to approach problems comprehensively and devise systematic solutions effectively. Additionally, I am grateful for the opportunity he provided me to refine my expertise while working at Kobelco Construction Company, a distinguished organization in Japan.

I extend my heartfelt appreciation to my colleagues at the Active Intelligent System Laboratory for fostering a conducive academic environment. Special thanks to Ito Kaich-san for his support during my initial year at Toyohashi University of Technology.

I am also deeply grateful to the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) for their generous financial assistance through the Scholarship program. Without their support, pursuing my doctoral studies would have been exceedingly challenging.

Lastly, I would like to express my gratitude to my wife, parents, and family for their unwavering encouragement and assistance throughout this journey. Their steadfast support has been a constant source of strength and motivation.

Abstract

The creation of a 3D map holds significant importance for autonomous systems navigating in unknown environments. The application of 3D mapping spans various fields, including autonomous driving, service robotics, agriculture, augmented reality, and construction. The need for efficient 3D mapping methods is growing with the proliferation of robots and autonomous systems.

Despite extensive research on 3D mapping, particularly for ground vehicles and drones, limited attention has been given to 3D mapping for construction cranes. Current 3D mapping methods encounter limitations when applied to constructing a map for cranes, primarily due to the unique challenges and complexities associated with crane mapping. There is a strong need for innovative approaches specifically tailored for mapping in crane scenarios. This thesis proposes three novel methods for mapping for a construction crane using multi-sensor fusion.

First, we propose a complementary filter and crane structure-based real-time sensor pose estimation and 3D mapping method for construction cranes with an arbitrary motion of the sensor system (2D lidar and IMU) attached to the crane boom. A heavy lidar with a slowly rotating base is needed to make a large-scale map both vertically and horizontally for cranes. In the proposed method, we introduce a complementary filter with moving average filtering for lidar pose estimation, which is more robust to severe vibration than Kalman filter-based methods. As there are only a small amount of overlaps between 2D lidar scans, we propose a map correction method based on pose graph optimization with planar environmental constraints. We evaluate the proposed method in a simulation and a real environment and compare it with one of the state-of-the-art methods. The evaluation results reveal that the proposed method can accurately estimate the sensor poses, thereby generating a high-quality, large-scale 3D point cloud map.

Second, we introduce a method for neural network-based real-time pose estimation using an IMU (inertial measurement unit) and its application in large-scale 3D mapping using a slowly rotating 2-D LiDAR. In this method, a neural network consisting of a convolutional neural network (CNN) and long short-term memory (LSTM) is employed to estimate the change in pose. Firstly, online pre-filtering using a low-pass filter is implemented on the

time windows of IMU measurements before feeding them as input to the neural network to estimate the change in position and rotation of the sensor. After that, the estimated sensor pose is used to register the scans of 2D-rotating LiDAR to build a large-scale 3D map. The proposed method is tested in a gazebo environment by attaching the sensors to a crane boom. In this study, we also investigate the impact of different time windows of IMU measurements on the accuracy of pose estimation by the neural network.

Third, we proposed a method for sensor pose estimation, as well as creating large-scale 3D maps, for construction cranes equipped with a sensor system consisting of a camera, 2D lidar, and IMU. To tackle the challenges posed by the crane boom's complex motion, we utilize an Extended Kalman filter (EKF) to improve the accuracy and reliability of sensor pose estimation. By combining pose estimates from Visual-Inertial Navigation System (VINS) with data from an additional IMU, we estimate the scale value of a monocular camera. This scale value, obtained from the EKF, is then integrated into the VINS algorithm to refine the previously estimated scale value. Slowly rotating 2D lidar is used to build a 3D map. Since there is limited overlap between 2D lidar scans, we leverage the estimated pose to align and construct a comprehensive 3D map. Additionally, we thoroughly evaluate the effectiveness of the latest VINS techniques, as well as the EKF-enhanced VINS approach, in the specific context of crane operations. Through comprehensive performance assessments conducted in both simulated and real environments, we compare the EKF-added VINS method with state-of-the-art VINS techniques. The evaluation results demonstrate that the EKF-added VINS method accurately estimates sensor poses, leading to the generation of high-quality, large-scale 3D point cloud maps for construction cranes.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Research Background

Building a 3D map is vital for autonomous systems operating in unknown environments. 3D mapping is widely used for many domains, such as autonomous driving, service robotics, agriculture, augmented reality, and construction [20, 86, 29]. As robots and autonomous systems have recently become common, the demand for 3D mapping is increasing rapidly. Although numerous studies on 3D mapping have been done, especially for ground vehicles [88, 79, 68, 14] and drones [44, 55, 66], very few have been done for 3D mapping for construction cranes [92, 45].

3D mapping for a crane in construction sites faces specific challenges, as follows: First, construction sites are generally open-sky, feature-scarce environments. Second, we need to make a large-scale map, both vertically and horizontally. Third, sensors attached to a long crane boom face much vibration. Fourth, when the crane is in operation, sensors mounted to the crane boom confront large rotations and displacements in any direction. These challenges make it difficult to adopt existing 3D mapping techniques.

1.2 Limitations of the Existing Work

Despite extensive research on 3D mapping of environments, there are limitations in current methods when it comes to applying them to crane operations. This is primarily due to the specific challenges and complexities involved in mapping for a crane within dynamic construction sites. Visual SLAM methods are popular in 3D mapping [30, 76, 75] but tend to be weak under varying lighting conditions or feature-scarce environments. 3D lidar-based mapping is also popular [80, 56, 44, 13], as 3D scans provide rich structural information even under poor lighting conditions outdoors. However, the limited vertical field of the usual 3D lidars is not suitable for large-scale mapping for cranes.

By rotating a 2D lidar, we can get a very wide (e.g., spherical) virtual 3D scan, to which we can apply 3D lidar-based mapping methods that adopt scan matching-based relative pose estimation [90, 89, 6, 19, 55]. The lidar-IMU fusion approach is effective in compensating the lidar motion during the rotation [90, 89, 66, 87]. For large-scale crane mapping, we need to use a heavy, long-range 2D lidar, and the rotation speed has to be small enough for motion stability. As a result, the motion between scan timings becomes large, making it difficult to obtain a virtual 3D scan reliably. There are many methods proposed for 3D mapping using fast-rotating 2D-LiDAR for autonomous vehicles, such as [69, 90, 80, 70]. These approaches would not work for 3D mapping for the crane when sensors are attached to the boom of the crane because, in this case, the sensor faces rotation along all three axes, unlike the case of mapping for an autonomous vehicle, where the sensor faces rotation in only one axis.

Although it is possible to continuously estimate the lidar pose using an IMU, as in the case of lidar-IMU fusion, commonly-used Kalman filter-based methods [90, 89, 66, 87] are not robust enough for cranes with severe vibration and might cause significant map distortion. Another common approach to map distortion correction is pose graph optimization (PGO) [9, 69]. However, usual PGO methods, which rely on scan matching, are not appropriate for our case because we have little overlap between scans when generated by a slowly rotating 2D lidar.

1.3 Research Objectives

Although extensive research has been conducted in the domain of 3D mapping, little attention has been given to the specific challenges posed by large-scale mapping for cranes. The goal of this study is real-time 6 DOF sensor pose estimation and large-scale 3D mapping for construction cranes. A heavy lidar with a slowly rotating base is needed to make a large-scale map both vertically and horizontally for cranes. This sensor configuration and mapping conditions entail handling each 2D scan separately, making it difficult to adopt existing rotating 2D lidar-based methods that construct a virtual 3D scan from a set of 2D scans. Due to the inherent challenges associated with the 3D mapping of construction sites for a large crane, we use a slowly-rotating 2D lidar and an IMU as base components and attach them to the crane boom to build a large-scale 3D map for cranes. The goal is to estimate the lidar pose for each scan and use the estimated pose to build a large-scale map while the crane boom moves arbitrarily during operations.

1.4 Contributions

The main contribution of this research work is the design, implementation, and testing of the following three novel real-time sensor pose estimation and large-scale 3D mapping methods for large construction cranes. We validate the performance of each of the proposed methods by implementing them in simulation and real-world environments using a crane.

1.4.1 Complementary Filter and Crane Structure-based Real-time Sensor Pose Estimation and 3D Mapping

In this approach, we estimate the lidar pose using crane structural information and IMU-based complementary filter [78] with moving average filtering, which is more robust to severe vibration than Kalman filter-based methods [18, 52, 28, 94]. To further improve the map accuracy, we develop a new pose graph optimization method using planar environmental constraints that naturally exist in construction sites. The proposed method can construct accurate 3D maps even when some state-of-the-art methods [90, 89] fail.

1.4.2 IMU-based Neural Network Approach for Real-time Sensor Pose Estimation and 3D Mapping

In this method, a learning-based approach is used for real-time sensor pose estimation by implementing the neural network on IMU measurements, and estimated 6D pose information is used to register the 2D LiDAR scan to build a large-scale 3D map.

1.4.3 Multi-sensor Fusion-based Real-time Sensor Pose Estimation and 3D Mapping

The proposed approach involves integrating measurements from multiple sensors, including a camera, a rotating 2D-Lidar, and an IMU mounted on the crane boom, to create a comprehensive 3D map for a large-scale crane. To achieve accurate pose estimation, the method employs an Extended Kalman filter (EKF) that combines the initial rough pose estimation with data from the IMU's gyroscope, accelerometer, and barometer. This fusion of sensor measurements enables precise sensor pose estimation within the system.

1.5 Thesis Organization

The rest of the thesis is organized as follows: In Chapter 2, we review the literature on relevant research. Chapter 3 explains complementary filter and crane structure-based real-time sensor pose estimation and 3D mapping methods. Chapter 4 describes our IMU-based neural network approach for real-time sensor pose estimation and our 3D mapping method. In Chapter 5, we explained the multi-sensor fusion-based real-time sensor pose estimation and 3D mapping method. Finally, in Chapter 6, we conclude the thesis and discuss future work.

Chapter 2

Literature Review

The creation of a three-dimensional (3D) map is crucial for enabling the effective functioning of autonomous systems in unfamiliar environments. The applications of 3D mapping span a wide range of fields, including unmanned aerial vehicles[80], robotics [32], biomedical engineering [52], sports training, agriculture [4], augmented reality [74], and construction site [20, 86, 29]. With the increasing prevalence of robots and autonomous systems, there is a growing demand for 3D mapping. While extensive research has been conducted on 3D mapping for ground vehicles [88, 79, 68, 14] and drones [44, 55, 66], there is a noticeable gap in studies focusing on 3D mapping for construction cranes [92, 45].

Mapping construction sites for cranes presents unique challenges that need to be addressed. These challenges include the absence of distinctive features in open-sky construction environments, the requirement to create comprehensive maps encompassing both vertical and horizontal dimensions, the sensitivity of sensors attached to the crane boom to significant vibrations, and the substantial rotations and displacements experienced by sensors mounted on the crane boom during crane operations. Overcoming these formidable challenges poses difficulties in adopting existing 3D mapping techniques.

A lot of research have been done on the 3D mapping of ground vehicles. For autonomous driving of ground vehicles, the LiDAR sensor is generally fixed at the top of the vehicle such as [88, 79, 68, 14]. In that case, only the yaw angle of the LiDAR sensor changes, and the rotation along the other two axes of the LiDAR is negligible so the roll and pitch angles hardly change in the ground vehicle. Because of this, the ground vehicle faces relatively less distortion in the 3D mapping of the environment. Limited research work has been done on 3D mapping for other dynamic applications such as 3D mapping for cranes. For construction cranes, the sensor is mounted on the boom of the crane for a wide and clear view of the construction site. As a consequence when the crane performs any task, the boom can move the sensor in any random direction. Collecting consistent LiDAR data for 3D mapping of cranes pose significant challenges, beacsue the LiDAR is attached to the boom of crane that is in continuous motion and rotation along all its three axes during operation. In this case, when constructing a point cloud map, the LiDAR sensor's trajectory must be taken into account throughout the scan time, otherwise, the cloud's structure may be severely deformed and potentially unrecognizable [6]. This type of cloud distortion has a number of drawbacks, including deforming the different shapes of objects and making appropriate localization difficult [83]. The accuracy of the distance measurement employed in the object detection-based collision avoidance system has also been compromised by the data distortion[37].

2.1 Diverse Sensor Utilization for 3D Mapping in Crane Environments

Numerous studies have investigated sensor pose estimation and 3D mapping using a variety of sensors. In the following section, we will provide an overview of some well-known studies in this field, focusing on the utilization of different sensors and discussing their respective limitations when applied to 3D mapping in crane environments.

2.1.1 Vision-based Approach

Visual SLAM (vSLAM) [30, 76, 75] is a low-cost but effective way of 3D mapping. vSLAM is suitable for small- or moderate-sized scenes, such as indoor and traffic scenes, but not for large-sized and feature-scarce and varying-illumination environments like construction sites. In addition, when a camera faces a quick motion by, for example, a severe vibration of the crane boom, the acquired image could easily be blurred, making feature extraction difficult.

2.1.2 3D Lidar-based Approach

Many 3D lidar-based mapping methods have been developed and effectively utilized in applications such as autonomous driving [80, 56] and drone-based aerial mapping [44, 13]. Point clouds provided by 3D lidars are easily matched between frames and are suitable for 3D mapping for limited and continuous sensor motions. However, the limited vertical field of view of usual 3D lidars does not fit the large-scale mapping at construction sites.

2.1.3 2D Lidar-based Approach

The combination of a 2D lidar and a rotating base is a promising approach to developing a low-cost and wide-area range measurement system [90, 6, 89, 19, 55]. These systems usually rotate the lidar fast enough to construct a virtual 3D scan from a sequence of 2D scans; consecutive virtual 3D scans sufficiently overlap with each other to be used for estimating the sensor pose change. For example, in [6], the rotational speed is 30 rpm or $180^\circ/\text{sec}$. For large-scale crane mapping, however, we need to use a heavy, long-range 2D lidar, and the rotation speed has to be small enough for motion stability. This slow speed makes it difficult to construct a consistent virtual 3D scan, and the existing methods do not work.

2.1.4 Lidar-IMU Fusion-based Approach

IMU has commonly been used for estimating the pose of a sensor in motion. However, under severe vibration, gyroscope and accelerometer measurements exhibit unmanageable sensor drift caused by sensor bias and noise uncertainty [18]. Many lidar-IMU mapping methods, such as LOAM [90, 89] and its extensions [80, 69, 13, 93], employ the Kalman filter to cope with noise in IMU measurements. However, there is a high chance that Kalman filter-based linear state estimation will diverge under high vibration, as in the case of crane application. The complementary filter [78] is used for orientation estimation using an IMU and exhibits better estimation accuracy and robustness than the Kalman filter under high vibration [18, 52, 28, 94]. Combined with the structural information of a crane, the complementary filter can be used for sensor pose estimation.

2.2 Different Methods for 3D Mapping in Crane Environments

Several studies have delved into sensor pose estimation and 3D mapping through the application of diverse fusion techniques and methodologies. In the subsequent section, we aim to outline some prominent fusion techniques and methodologies, while also addressing their individual constraints when implemented for 3D mapping within crane environments.

2.2.1 3D Lidar Based Methods for Sensor Pose Estimation and 3D Mapping

LiDAR sensors are used for the purpose of 3D mapping in many application areas[38]. The 3D mapping method based on only the LiDAR sensor faces issues in an open-sky featureless environment such as LiDAR can't provide an accurate sensor rotation estimation in an open-sky featureless environment because there is no prominent pattern of point distribution in z-direction [2]. Moreover, if the laser scanning motion is relatively slower than the extrinsic LiDAR motion, then the scan data distortion is very obvious. To overcome this problem, some authors integrate other sensors such as IMU, GPS, encoders, camera, or compass with the LiDAR sensors [67, 9] to build a 3D map. These 3D mapping methods generally use two different approaches for LiDAR scan registration. One approach is scan matching based registration and another approach is feature extraction based registration. Scan matching based registration approach which generally uses Iterative Closest Point (ICP) methods [53, 31] can give good results if the LiDAR's extrinsic motion is very slow and the laser scan rate is high. In this case, the motion distortion within the scan is relatively small. In [53], Francois Pomerleau uses a standard ICP based method to match laser scan returns among different scans. A method that comprises two steps for scan data correction is proposed in [31]. In the first step an ICP based velocity measurement is computed and then in the second step distortions are compensated using the measured velocity. Registration of 3D point cloud data based on scan-to-scan matching such as ICP and modified versions of ICP are an iterative technique that takes a lot of processing time and is not practically suitable for real-time because of high computation time [65], [54]. Furthermore, in a dynamic environment, there is a big probability of wrong matching results due to changes in surrounding conditions [2]. The methods in which registration is based on feature extraction sequentially register the extracted planar and edge features to incrementally build a global map. These methods can be implemented for real-time applications, such as LOAM [90] which extract the features to build a map in real-time. However, firstly, most of these 3D mapping methods are designed for general autonomous vehicles and secondly, these methods are based on 3D LiDAR.

2.2.2 IMU-based Neural Network for Sensor Pose Estimation and 3D Mapping

The primary problem in IMU based traditional pose estimation approaches is precisely computing the linear acceleration produced by only sensor motion while simultaneously estimating the inherent error, bias, and gravity direction from acceleration measurements[84]. Recently learning-based approaches have been proven to be capable of dealing with the drift problem of IMU-based pose estimation approaches by directly estimating the device's spatial displacement using supervised learning [84, 1, 11, 10, 42]. Chen et al. [10] used an IMU-based inertial odometry neural network (IONet) to estimate velocity. To reduce drift, he divides the acceleration data into independent windows. This method is not designed to estimate the pose or odometry. Kim at al. [84] presented Extended IONet, a system that combines a 9-Axis IONet with Pose-Tuning Net to enhance trajectory tracking accuracy by compensating for the 6-Axis IONet's drift issue. In this method, the magnetometer is required and magnetic disturbance influences the attitude and heading accuracy. A neural network-based inertial navigation is proposed in [85, 72] to track the positions and orientations of a moving human using smartphone IMU measurements but this method is limited to 2D pose estimation and does not work for 3D pose estimation. João at al [71] proposes an end-to-end learning framework for 6-DOF pose estimation using IMU, but the approach's limitations are that it introduces a delay in real-time pose estimation due to the proposed method for inertial data feeding in neural networks, and its accuracy is affected if the IMU is subjected to a lot of vibration and noise.

2.2.3 Multi-sensor Fusion for Sensor Pose Estimation and 3D Mapping

The research conducted in [49] demonstrated that the integration of additional sensors to VINS using the Extended Kalman Filter (EKF) improves the accuracy of odometry estimation in robot systems. They also compare the accuracy of pose estimation using two IMUs versus a single IMU. However, during the experiment, it was observed that the second IMU encountered a failure after approximately 45% of the trajectory, limiting the evaluation of its impact on pose estimation accuracy. The authors of [5] proposed fusion algorithms using multiple IMUs to enhance pedestrian navigation performance. They found that the accuracy of pose estimation is directly related to the number of IMU sensors used. In [34], an experimental comparison of various Visual-Inertial Navigation System (VINS) algorithms in the underwater domain was conducted. The study revealed that while VINS-Mono [58] demonstrated excellent performance, but its scale estimation was consistently inaccurate due to the use of monocular vision. The limitation of VINS-MONO [58] regarding scale estimation is that VINS-MONO depends on an initial scale parameter estimated at the initialization step. This means that any errors in the initial scale estimation could propagate throughout the system. The results in [34] confirmed that incorporating IMU measurements significantly improved performance compared to pure Visual Odometry, extending the findings reported in [34]. The improved performance of IMU integration was observed across diverse underwater environments [34].

Typically, in estimating scale using VINS with a monocular camera, a combination of data from inertial and visual sensors is utilized during initialization. However, if the visual data is insufficient, unclear, or noisy, it can negatively affect the accuracy of scale estimation, consequently impacting the overall performance of the system and, ultimately, pose estimation. In our proposed approach, we introduced an Extended Kalman filter (EKF) to iteratively update the scale value, thereby improving the precision and dependability of sensor pose estimation, particularly in complex crane boom trajectories. This enhancement was achieved by integrating pose estimations from VINS with additional IMU data.

2.2.4 3D Lidar-based Mapping Correction

There are many different approaches are proposed in the literature for mapping correction such as in [9], GPS/INS sensors integrated with the LiDAR sensor to build a map, and data correction method is proposed for an autonomous vehicle using the position of the vehicle before and after every scan. Thus, the accuracy depends on the position measurement of the vehicle. In [17], a visual feature-based approach for correcting the geometric motion distortion of the LiDAR sensor is proposed. In which a frame-by-frame visual odometry estimation framework based on a pose interpolation scheme is proposed. A similar approach is proposed in [3] that uses a RANSAC-based algorithm in the visual odometry pipeline using LiDAR intensity data to construct a constant velocity model for data correction. However, both of the above approaches depend on visual features and dense point cloud data. The extraction of reliable visual features from LiDAR data is relatively difficult. Ji Zhang in [90] proposes a motion model that extracts and matches geometric features in cartesian space and his approach does not require dense point cloud data. For a moving car with a LiDAR sensor, Pierre Merriaux [48] proposes a data correction approach based on CAN bus data using a linear motion model. Most recently, Tobias Renzler [61] proposes motion distortion correction for scanning LiDAR measurement using odometry information. LeGO [69] modified the LOAM [90] for the UGV application by canceling out unreliable features and applying ground plane extraction and point cloud segmentation, which presents great stability in areas that contain noisy objects such as trees and grass. In our case, we use a completely unique approach for mapping correction, which is based on pose graph optimization technique with plane constraints to estimate the sensor's pose and then use the pose information to correct the distorted data.

Pose graph optimization (PGO) is a map correction method based on inter-pose relationships and can generate an accurate map, especially with loop closing information [43, 95, 36]. In 3D lidar-based PGO [9, 69], matching between actual or virtual 3D scans can provide relationships between nearby pose nodes. However, in our slowly-rotating 2D lidar case, we do not have enough overlapping between scans, and the usual PGO approach with only sensor pose nodes does not work. To deal with this problem, we need to introduce extra constraints [40, 39].

Chapter 3

Complementary Filter and Crane Structure-based Real-time Sensor Pose Estimation and 3D Mapping

3.1 Introduction

This chapter describes a novel method for large-scale 3D mapping for construction cranes with an arbitrary motion of the sensor system (2D lidar and IMU) attached to the crane boom. In the proposed method, we introduce a complementary filter with moving average filtering for lidar pose estimation, which is more robust to severe vibration than Kalman filter-based methods. As there are only a small amount of overlaps between 2D lidar scans, we propose a map correction method based on a pose graph optimization with planar environmental constraints. We evaluate the proposed method in a simulation and real environment and compare it with one of the state-of-the-art methods. The evaluation results reveal that the proposed method can accurately estimate the sensor poses, thereby generating a high-quality, large-scale 3D point cloud map.

Fig. 3.1 shows the overall block diagram of the proposed method. The sensor system is composed of a 2D lidar, a rotating base, and an IMU. The *odometry module* calculates the inertial odometry with the complementary filter. It then estimates the lidar pose with the angle of the rotating base and the crane's structural information. The *mapping module* receives lidar poses from the odometry module, transforms the lidar measurements from the lidar frame to the fixed world frame, and assembles them to construct a 3D point cloud map. The *mapping correction module* further improves the map by pose graph optimization with planar environmental constraints.

The contribution of this work is summarized as follows:

- We propose a novel approach to making a large-scale 3D map for construction cranes.
- We introduce a complementary filter with moving average filtering to accurately and robustly estimate the sensor pose combined with the structural information of the crane.
- We develop a novel map correction method using planar environmental constraints with the pose graph optimization scheme.
- We show the effectiveness of the proposed method in simulation and real experiments.

The rest of the chapter is organized as follows. Section 3.2 explains lidar pose estimation using a complementary filter. Section 3.3 describes the 3D mapping method, and Section 3.4 explains the mapping correction using the pose graph optimization with planar environmental constraints. Sections 3.6, 3.7 and 3.8 describe experimental results in a simulation, a small-sized model crane, and a real crane respectively. Section 3.9 concludes the chapter.

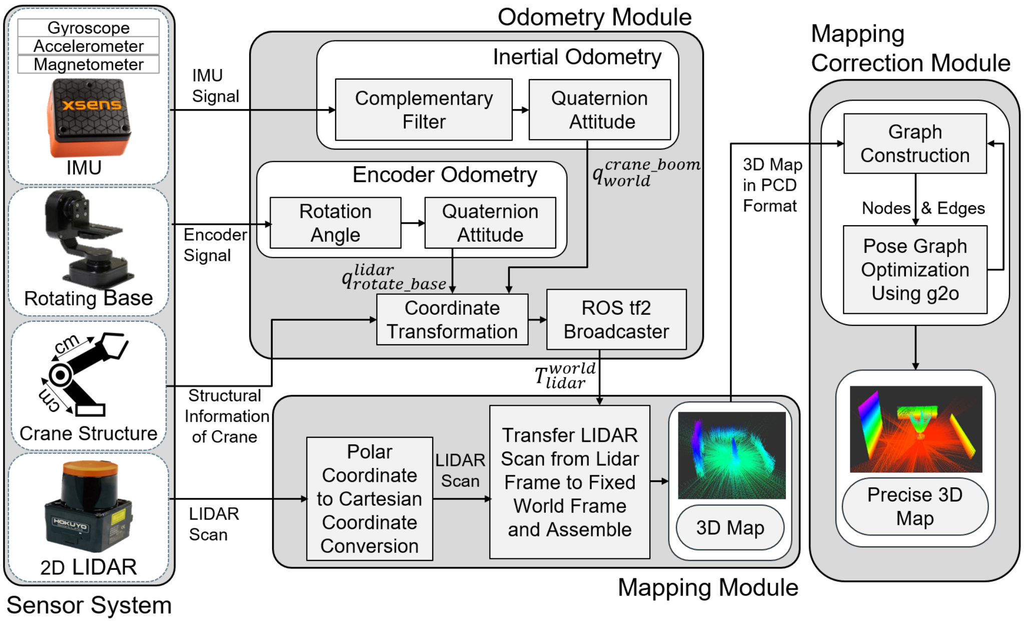


Fig. 3.1 Block diagram of proposed method showing all three modules: Inertial Odometry Module, Mapping Module, and Mapping Correction Module.

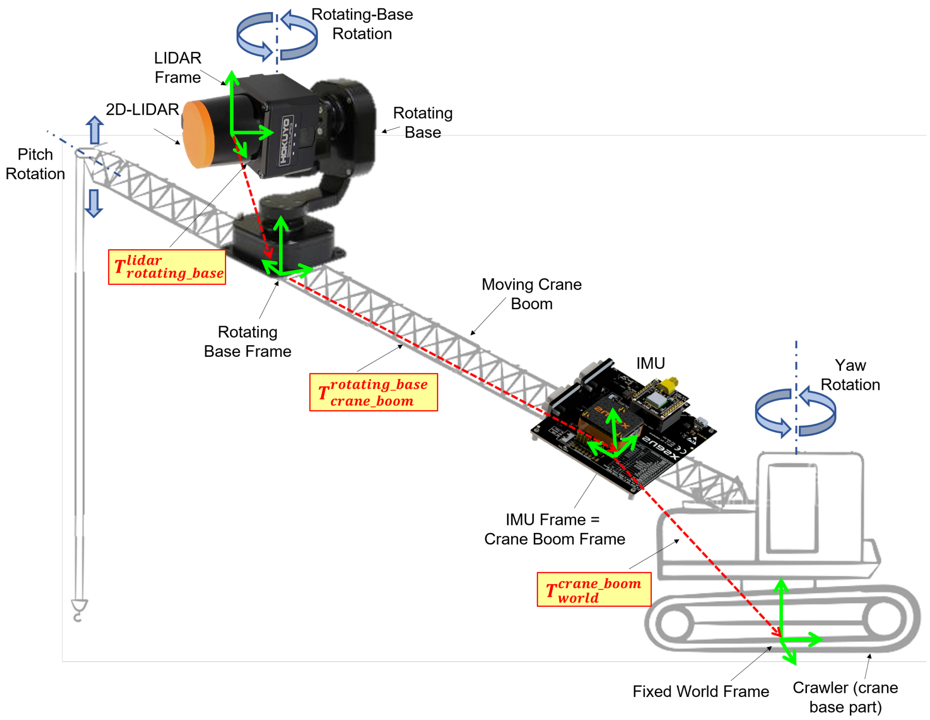

Rotating-Base Rotation
LIDAR Frame
2D-LIDAR
Rotating Base
Pitch Rotation
$T^{lidar}_{rotating\_base}$
Moving Crane Boom
Rotating Base Frame
$T^{rotating\_base}_{crane\_boom}$
IMU
Yaw Rotation
IMU Frame = Crane Boom Frame
$T^{crane\_boom}_{world}$
Fixed World Frame
Crawler (crane base part)

Fig. 3.2 Sensor system attached on crane boom and the relationship between different coordinate frames is also shown.

3.2 Odometry Module

The odometry estimates the lidar's position in real-time. As demonstrated in Fig. 3.2, due to the movement of crane boom, the sensor system attached to boom also moves and odometry module's objective is to calculate the pose of the lidar during crane boom motion. The change in pose comprises a translation and rotation of lidar. A novel odometry estimation method for the crane is proposed in which the rotation is estimated using a rotating base encoder and IMU, and translations are obtained using structural information of the crane. We use a quaternion to represent the rotation and a 3D vector to depict the translation. The details of coordinate transformation and estimation of rotation and translation are given below.

3.2.1 Coordinate Transformation

Fig. 3.2 shows that in our design there are four coordinate systems (shown in green) and three transformations between them (shown with red dotted arrows). To compute the transformation T^{world}_{lidar} from the lidar frame to the fixed world frame, we can chain the transformation between

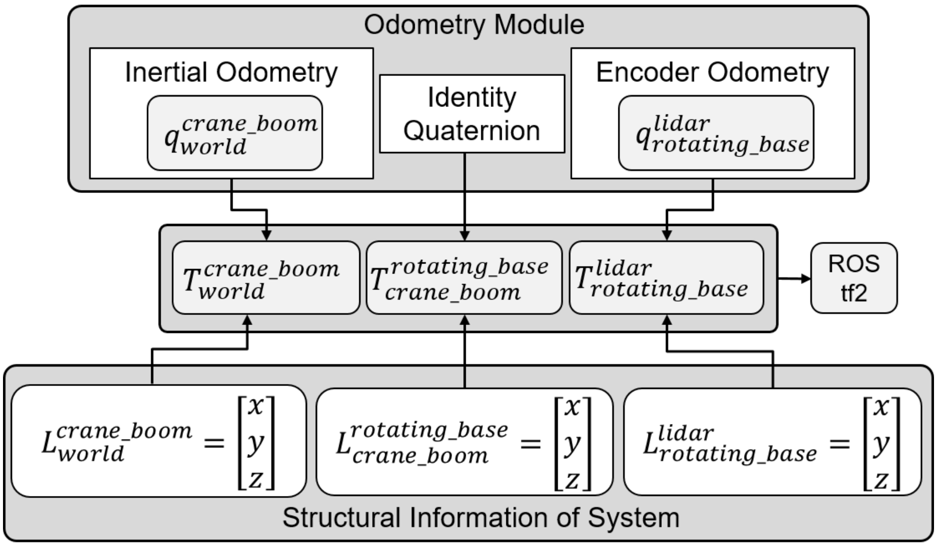

Odometry Module
Inertial Odometry
$q_{world}^{crane\_boom}$
Identity Quaternion
Encoder Odometry
$q_{rotating\_base}^{lidar}$
$T_{world}^{crane\_boom}$
$T_{crane\_boom}^{rotating\_base}$
$T_{rotating\_base}^{lidar}$
ROS tf2
$L_{world}^{crane\_boom} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$
$L_{crane\_boom}^{rotating\_base} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$
$L_{rotating\_base}^{lidar} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$
Structural Information of System

Fig. 3.3 Coordinate transformation of all frames in our system.

the coordinate frames, as given below:

$$T_{lidar}^{world} = T_{crane\_boom}^{world} \, T_{rotating\_base}^{crane\_boom} \, T_{lidar}^{rotating\_base}, \tag{3.1}$$

where $T_{crane\_boom}^{world}$ is the transformation between the crane boom frame and the fixed world frame, $T_{rotating\_base}^{crane\_boom}$ is the transformation from the crane boom to the rotating base frame, and $T_{lidar}^{rotating\_base}$ is the transformation between the rotating base frame to the lidar frame. The transformation of each coordinate system is calculated as follows (see Fig. 3.3):

- $T_{crane\_boom}^{world}$: The crane boom frame is attached to the crane boom at the point where the IMU is mounted. Thus the IMU frame and the crane boom frame are aligned with each other. Fig. 3.3 shows that the quaternion ($q_{crane\_boom}^{world}$) obtained from inertial odometry and a 3D vector ($L_{crane\_boom}^{world}$) representing the distance between the world frame and the crane boom frame along x, y, and $z-$axes is used for the transformation $T_{crane\_boom}^{world}$.

- $T_{rotating\_base}^{crane\_boom}$: The rotating base frame is placed at the bottom of the rotating base. It is a child frame of the crane boom frame. Fig. 3.3 shows that the rotation between the rotating base frame and crane boom frame is given by identity quaternion, because both frames are fixed on the crane boom, and no rotation occurs between them. The vector $L_{crane\_boom}^{rotating\_base}$ given the translation between the crane boom frame and rotating base frame.

- $T_{lidar}^{rotating\_base}$: The lidar frame (positioned at the lidar) is a child frame of the rotating base frame. The Fig. 3.3 shows that the quaternion $q_{rotating\_base}^{lidar}$ and vector $L_{rotating\_base}^{lidar}$ given the transformation between the rotating base and lidar frame.

3.2.2 Estimation of Rotation

The lidar attached to the crane boom faces two rotations: one is the rotation due to the rotating base and the other is the rotation due to the rotation of the crane boom (see Fig. 3.2). The rotating base's rotation angle is measured by its encoder and converted into a quaternion. This quaternion $q_{rotating\_base}^{lidar}$ represents the rotation of the lidar frame with respect to the rotating base frame, and this step is called the *encoder odometry* in Figs. 3.1 and 3.3.

The rotation of the crane boom frame with respect to the fixed world frame is measured by an IMU in quaternion form $q_{world}^{crane\_boom}$ using quaternion-based complementary filter [78]. This step is called *inertial odometry*. The complementary filter fuses the orientation estimated using a gyroscope with the orientation computed using an accelerometer and magnetometer. The rotation of the crane boom is first predicted using gyroscope measurements; then the roll and pitch of the boom are corrected using accelerometer data, and the yaw of the crane boom is corrected using magnetometer readings.

When an IMU is attached to the crane boom, accelerometer readings may be fluctuated abruptly due to the high vibrations of the crane boom. In addition, magnetometer readings may be distorted due to the magnetic field of the site. For the former problem, we apply a moving average filter to obtain the stabilized accelerometer reading a_k using:

$$a_k = \frac{1}{N} \sum_{i=k-N}^{k} \tilde{a}_k \tag{3.2}$$

where $\hat{a}_k$ is accelerometer reading at time k, and N is the window size of the smoothing. For the latter problem, we confirm if the magnetic distortion is small enough before fusing the magnetometer readings into the complimentary filter. The magnitude of the magnetic distortion is estimated by the total flux $||^b m||$ defined as:

$$||^b m|| = \sqrt{m_x^2 + m_y^2 + m_z^2} \tag{3.3}$$

where m_x, m_y, and m_z are magnetometer readings in the three axes. When there is no distortion, total flux is normalized to unity ($||^b m|| = 1$) [62]. We use this as a standard for identifying magnetic distortion. If the total flux is close to unity, we consider the magnetometer readings are reliable and there is no significant magnetic distortion. Next, we

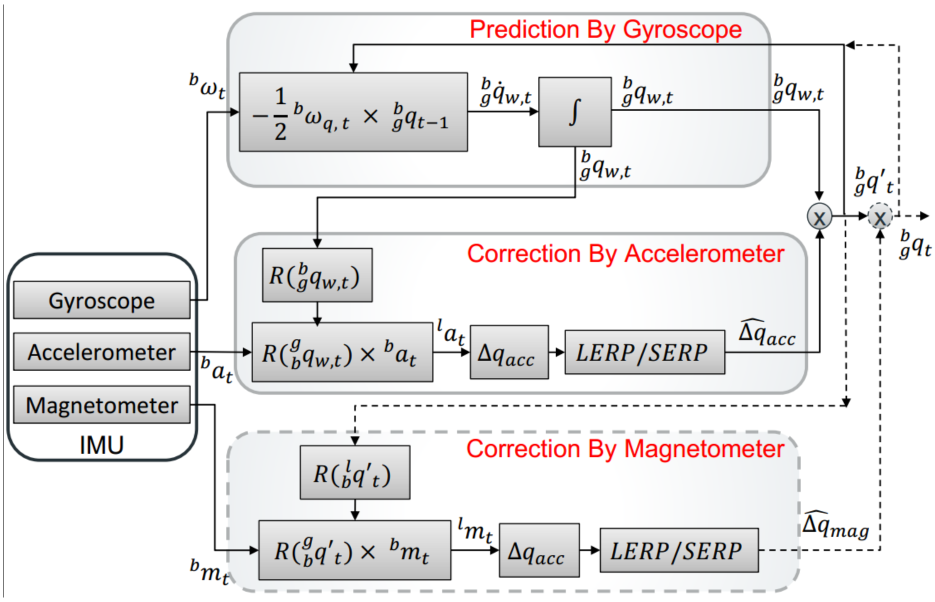


Fig. 3.4 Quaternion-based complementary filter.

will discuss our approach to computing the crane boom rotation using a quaternion-based complementary filter [78]. The block diagram of the computation of a quaternion-based complementary filter is shown in Fig. 3.4.

Crane boom rotation prediction by gyroscope The orientation between the crane boom frame (b) to fixed-world frame (w) in quaternion form $q_{world}^{crane\_boom}$ is estimated by using measurements of angular velocity obtained from the gyroscope. Consider the following for the sake of equation simplicity:

$$q_{world}^{crane\_boom} = {}^b_w q, \tag{3.4}$$

The angular velocity in quaternion form ${}^b\omega_{q,t_k}$ and quaternion derivative ${}^b_w\dot{q}_{w,t_k}$ at time t_k are related by the following equation:

$${}^b_w\dot{q}_{w,t_k} = -\frac{1}{2}\,{}^b\omega_{q,t_k} \otimes {}^b_w q_{t_{k-1}}, \tag{3.5}$$

where ${}^b_w q_{t_{k-1}}$ is previous estimate of quaternion. The matrix form of above equation is given below:

$${}^b_w\dot{q}_{w,t_k} = \Omega({}^b\omega_{t_k})\,{}^b_w q_{t_{k-1}}, \tag{3.6}$$

$$\Omega({}^b\omega_{t_k}) = \begin{bmatrix} 0 & {}^b\omega_{t_k}{}^T \\ {}^b\omega_{t_k} & -[{}^b\omega_{t_k}\times] \end{bmatrix}, \tag{3.7}$$

where ${}^{b}\omega_{t_k}\times$ is a cross-product matrix associated with angular velocity ${}^{b}\omega_{t_k}$ at time t_k. The orientation of crane boom frame (b) relative to the fixed-world frame (w) at time t_k, ${}^{b}_{w}q_{t_k}$ is computed by integration of the quaternion derivative and the sampling period Δt:

$$ {}^{b}_{w}q_{w,t_k} = {}^{b}_{w}q_{t_{k-1}} + {}^{b}_{w}\dot{q}_{w,t_k}\Delta t. \tag{3.8}$$

Crane boom roll and pitch correction by accelerometer The gravity vector measured by the accelerometer ${}^{b}a$ is transferred from the crane boom frame to the fixed-world frame by using the inverse predicted quaternion ${}^{w}_{b}q_{w}$ from Eq. 3.8 as given below

$$R({}^{w}_{b}q_{w})\,{}^{b}a = {}^{w}g_{p}. \tag{3.9}$$

The predicted gravity ${}^{w}g_{p}$ have small deviation from real gravity vector ${}^{w}g$; therefore Δq_{acc} which rotates ${}^{w}g_{p}$ into ${}^{w}g$ is computed as:

$$R(\Delta q_{acc})\,{}^{w}g = {}^{w}g_{p}. \tag{3.10}$$

If we write ${}^{w}g$ and ${}^{w}g_{p}$ in vector form then Eq. 3.10 become as

$$R(\Delta q_{acc})\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}g_x\\g_y\\g_z\end{bmatrix}. \tag{3.11}$$

By simplifying the Eq. 3.11 , we can get

$$\Delta q_{acc} = \begin{bmatrix}\sqrt{\frac{g_z+1}{2}} & -\frac{g_y}{\sqrt{2(g_z+1)}} & \frac{g_x}{\sqrt{2(g_z+1)}} & 0\end{bmatrix}^T, \tag{3.12}$$

where g_x, g_y and g_z are the x, y, and z components of acceleration measured by the accelerometer which are affected by high-frequency noise. Interpolation with identity quaternion q_I is used to minimize that accelerometer noise. Two different interpolation approaches are used based on the angle $\Delta q0_{acc}$ between q_I and Δq_{acc}. If $\Delta q0_{acc}$ is greater than a predefined threshold value ε ($\Delta q0_{acc} > \varepsilon$), linear interpolation(LERP) is used as given below as follows [78]:

$$\overline{\Delta q_{acc}} = (1-\alpha)q_I + \alpha\Delta q_{acc}, \tag{3.13}$$

where α is the gain that represents the cut-off frequency of the filter [16]. By normalizing the Eq. 3.13 we get

$$\widehat{\Delta q_{acc}} = \frac{\overline{\Delta q_{acc}}}{||\overline{\Delta q_{acc}}||}. \tag{3.14}$$

If $\Delta q_{0acc} < \varepsilon$, spherical linear interpolation (SLERP) is used as given below as follows [78]:

$$\widehat{\Delta q_{acc}} = \frac{sin([1-\alpha]\Omega)}{sin\Omega} q_I + \frac{sin(\alpha\Omega)}{sin\Omega} \Delta q_{acc}. \tag{3.15}$$

Finally, this filtered quaternion $\widehat{\Delta q_{acc}}$ is multiplied by the quaternion predicted by the gyroscope ${}^{b}_{w}q_{\omega}$ and it provides the correction in roll and pitch component as given in the below equation:

$${}^{b}_{w}q' = {}^{b}_{w}q_{\omega} \otimes \widehat{\Delta q_{acc}} \tag{3.16}$$

Crane boom yaw correction by magnetometer If the magnetic distortion detector does not detect any distortion, the magnetic field vector ${}^{b}m$ measured in the crane boom frame is transformed to the fixed-world frame using the inverse predicted quaternion ${}^{w}_{b}q'$ from Eq. 3.16 as follows:

$$R({}^{w}_{b}q')\ {}^{b}m = l. \tag{3.17}$$

where l represents the rotated magnetic field vector. The next step is to find the delta quaternion Δq_{mag}, which rotates the vector l into the vector that lies on the xz-semiplane, using the following equation:

$$R^T(\Delta q_{mag}) \begin{bmatrix} l_x \\ l_y \\ l_z \end{bmatrix} = \begin{bmatrix} \sqrt{l_x^2 + l_y^2} \\ 0 \\ l_z \end{bmatrix}, \tag{3.18}$$

where l_x, l_y, and l_z are x, y and z axes of l. As this delta quaternion performs a rotation only along z-axis so other axes in quaternion Δq_{mag} are set to zero as:

$$\Delta q_{mag} = \begin{bmatrix} \Delta q_{0mag} & 0 & 0 & \Delta q_{3mag} \end{bmatrix} \tag{3.19}$$

By solving the system of equation Eq. 3.18 by substituting Δq_{mag}, we will get following equation as follows [78]:

$$\Delta q_{mag} = \begin{bmatrix} \frac{\sqrt{\Gamma + l_x\sqrt{\Gamma}}}{\sqrt{2\Gamma}} & 0 & 0 & \frac{l_y}{\sqrt{2(\Gamma + l_x\sqrt{\Gamma})}} \end{bmatrix}. \tag{3.20}$$

where

$$\Gamma = l_x^2 + l_y^2 \tag{3.21}$$

To minimize the noise of the magnetometer, same LERF and SLERP is used as given in Eq. 3.13 and Eq. 3.15 and final quaternion is obtained as follows [78]:

$$ {}^{b}_{w}q = {}^{b}_{w}q' \otimes \widehat{\Delta q_{mag}} \tag{3.22}$$

This final quaternion represents the rotation of crane boom with respect to the fixed-world coordinate frame.

3.2.3 Translation Parameters

The structural information (dimension) of the crane system is used to specify the translations between coordinate frames. A 3D vector L_a^b provides the distance along x, y, and z axes between frames “a" and “b" and is used as a translation between them. Fig. 3.3 shows all translations between different frames in our system.

The tf2 broadcaster [21] of robot operating system (ROS) broadcasts the transformation of all coordinate systems. Coordinate transformation messages are broadcasted each time an update occurs about a specific transform of any frame, to keep track of the moving lidar frame.

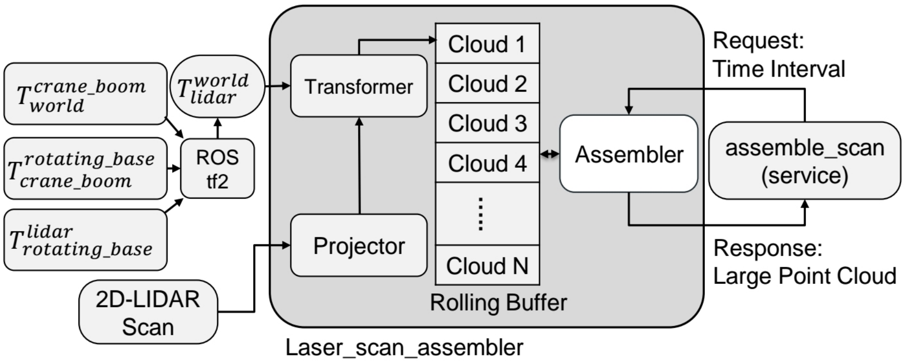


Fig. 3.5 Block diagram of mapping module.

3.3 Mapping Module

We build a 3D map during the sensor system’s motion using a laser-assembler [63], [82]. The laser-assembler assembles individual laser scan lines of 2D lidar into a composite 3D point cloud. Fig. 3.5 shows a block diagram of our mapping method. First the block “projector" converts the lidar scans from polar coordinate to Cartesian coordinate (XYZ), named as lidar frame. Because the lidar frame is in motion because of lidar motion, the next step is to

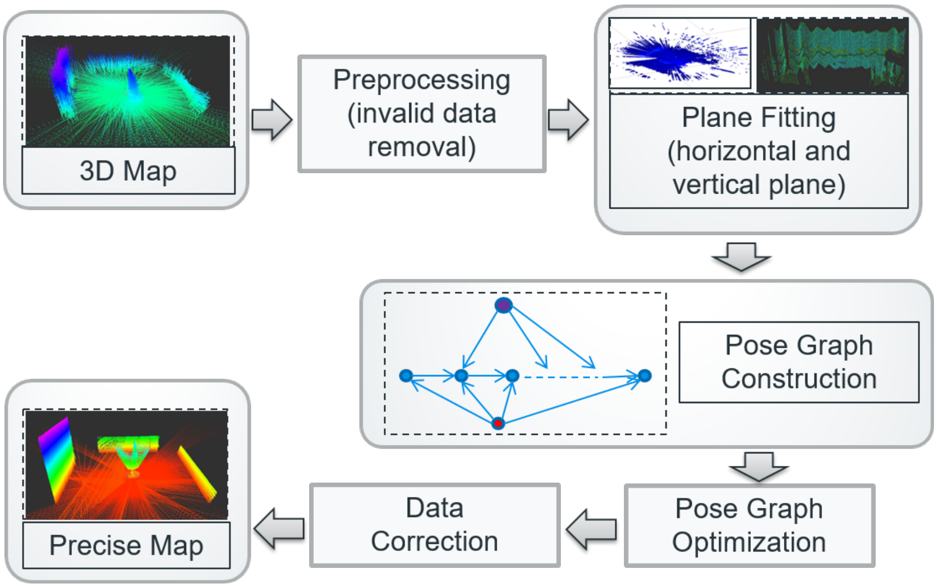

3D Map
Preprocessing (invalid data removal)
Plane Fitting (horizontal and vertical plane)
Pose Graph Construction
Pose Graph Optimization
Data Correction
Precise Map

Fig. 3.6 Block diagram of mapping correction module.

transfer the moving lidar frame to a fixed world frame to obtain a 3D view of the world. This coordinate transformation is an important step to create a 3D map using a moving 2D-lidar. The block “transformer" transforms the lidar measurements from the lidar frame to the fixed world frame as

$$P_{world} = T_{lidar}^{world} P_{lidar} \tag{3.23}$$

Then, the transferred lidar measurements are stored in a rolling buffer for a predetermined time. Whenever a request is sent for a 3D point cloud, the rolling buffer sends out large assembled transferred laser scans in Point Cloud (.pcd format).

3.4 Mapping Correction Module

Fig. 3.6 shows the proposed map correction approach. After the pre-processing step, we extracted two types of planes: ground planes and vertical wall planes. After extracting the planes, we constructed the pose graph using the sensor pose as internal nodes and plane constraints as external nodes. A general graph optimization (g2o) technique [43] was used to optimize the constructed pose graph. We used the sum of squared distances between plane to all points as the optimization criterion. The optimized sensor poses in the pose graph for each scan line were used as a transformation matrix to modify the distorted data and minimize the data distortion error.

3.4.1 Pre-processing and Plane Extraction

Extracting reliable planes from 3D point cloud data is an important task in our approach. Before extracting planes, the original point cloud data are pre-processed to filter out invalid data (NaN values). The NaN values in point cloud data represent the erroneous and too far points.The filtered data are used for extracting different planes. We use a RANSAC-based plane extraction routine of point cloud library (PCL). First, we extracted a ground plane by choosing a loosely defined threshold and subtracted the points belonging to the ground plane from the pre-processed point cloud data. We then extracted one or more vertical wall plane(s). The algorithm for Ground Plane Extraction and Wall Plane Extraction are given in (**Algorithm 1**) and (**Algorithm 2**).

The original point cloud data are divided into frames, known as scan lines. The sensor poses vary for each frame. We performed an index mapping that keeps track of index information of the original point cloud data, the filtered point cloud data, and the point cloud data of each extracted plane. Thus, the index mapping provides information about which frame belongs to which plane. This information is required for pose graph construction. The mapping information is also needed for scene reconstruction using the optimized sensor poses.

Algorithm 1 Ground Plane Extraction.

```
Ensure:
  Define ground plane extraction parameters:
  ground_angle_degree, ground_distance_threshold, min_ground_plane_size,
  ground_plane_cluster_Tolerance
  while size of remaining plane points < threshold do
    Extract the ground plane from the input cloud using parameters
    Perform clustering on the extracted ground plane
    Select the largest cluster plane from the ground plane clustering
    Remove the extracted input cloud
  end while
```

3.4.2 Pose Graph Construction

The pose graph presented in Fig. 3.7 is composed of internal and external nodes. The internal nodes (pose nodes; light green color nodes) are defined for each frame in the point cloud data. An edge, e_i between pose node $i-1$ and pose node i is represented by two parameters: relative pose initially set to be an identity matrix, and an information matrix. The information matrix values are chosen to be small values based on trial and error. The external nodes

Algorithm 2 Wall Plane Extraction.

```
Define wall plane extraction parameters:
wall_angle_degree, wall_distance_threshold, min_wall_plane_size,
wall_plane_cluster_Tolerance
Define clustering parameters:
min_wall_segment_cluster_size, max_wall_segment_cluster_size,
wall_segment_cluster_Tolerance
Perform clustering on the remaining points after removing ground plane
for each cluster do
    while size of remaining plane points in the cluster $<$ threshold do
        Extract a plane from the cluster
        Remove the extracted plane points from the cluster
    end while
end for
```

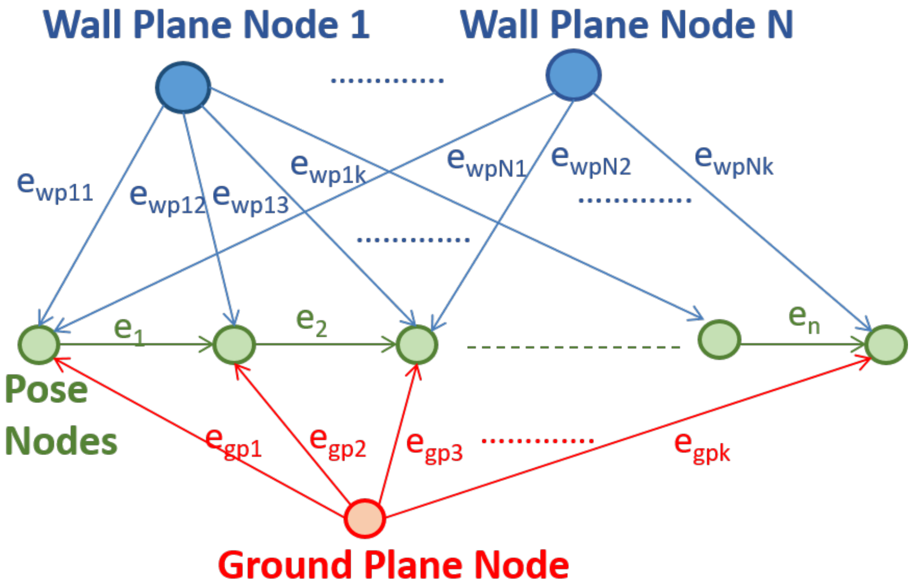


Fig. 3.7 Pose graph showing pose nodes, plane nodes and edges.

are defined by the different plane constraints. The ground plane constraint is added as an external node (red color nodes in Fig. 3.7); in this research, we assume that we have only one ground plane. The edge between the ground plane node and the i_{th} pose node is denoted as e_{gp_i}. One or more wall plane nodes are also added as external nodes (blue nodes in Fig. 3.7). The edge between the k_{th} wall plane node and the i_{th} pose node is denoted as $e_{wp_{ki}}$. Such an edge between a plane node and a pose node is added if the corresponding frame contains points on the plane and is represented by the information matrix and the measurement error. The information matrix values are chosen based on trial and error. The measurement error is calculated based on the points-to-plane distance. For a pair of a plane and a pose node, the measurement error is defined as:

$$E = \sum_{j=1}^{K} d_{\perp}{}_{j}^{2}, \tag{3.24}$$

where $d_{\perp j}$ is the perpendicular distance from point j to the plane and K is the total number of points of the pose node belonging to the plane.

3.4.3 Pose Graph Optimization and Map Correction

The general graph optimization (g2o) [43] is used to optimize the pose graph. In the optimization principle, we used the sum of squared errors calculated using Eq. (3.24). At the end of the optimization process, we obtained the sensor pose matrix for each sensor pose node. The sensor pose matrix is then used to transform the point cloud for the corresponding frame or scan line. Finally, we merge the transformed point cloud to obtain the corrected point cloud data.

3.4.4 Successive and Iterative Data Correction

We perform our data correction process successively and iteratively as shown in Fig. 3.8. Because in the beginning, the distortion error in the given point cloud data (PCD) is high, we first extract only the ground plane from the given PCD using a loosely defined threshold. After the first optimization and data correction process, distortion error is reduced, and we can reliably extract more wall planes successively. Each time we extract one more plane and add it to the pose graph, we again optimize the sensor pose and make data corrections.

When all wall plane constraints from the distorted 3D map are added to the pose graph, we calculate the average of points to plane distance to measure the desirable accuracy. If the average plane to points distance is below the specific threshold value, we use the sensor pose information as a final sensor pose to modify the point cloud data and generate final corrected map. Otherwise, we again extract all the planes with a reduced threshold value

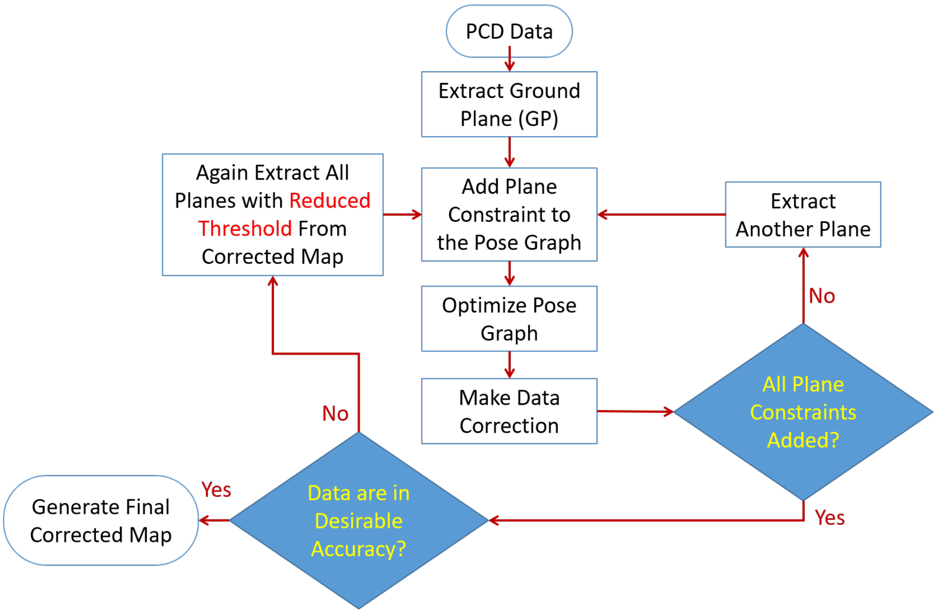

PCD Data
Extract Ground Plane (GP)
Add Plane Constraint to the Pose Graph
Extract Another Plane
Again Extract All Planes with Reduced Threshold From Corrected Map
Optimize Pose Graph
Make Data Correction
All Plane Constraints Added?
No
Yes
Data are in Desirable Accuracy?
No
Yes
Generate Final Corrected Map

Fig. 3.8 Successive and iterative approach of pose optimization.

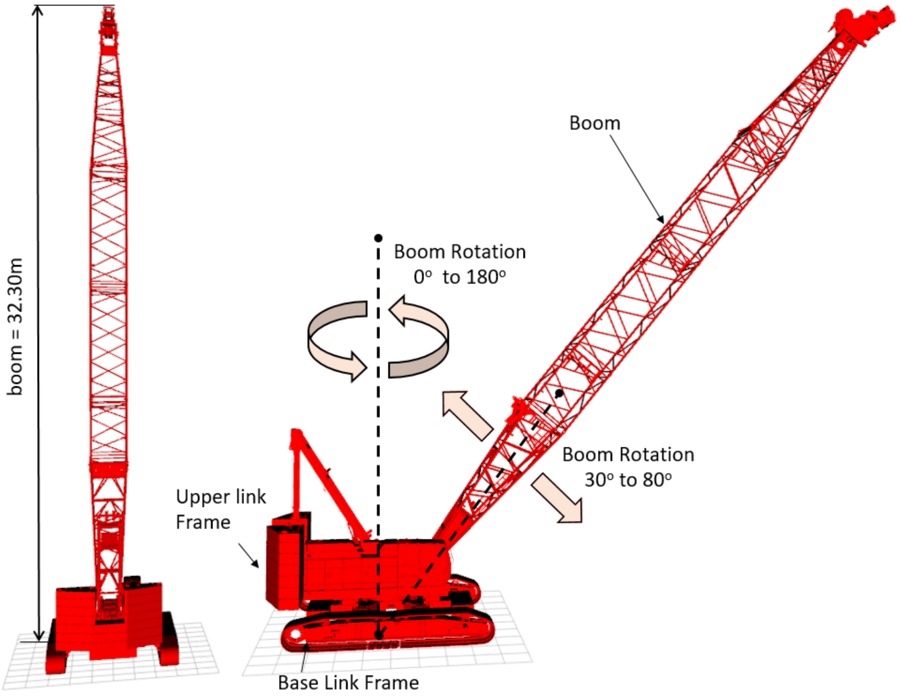


Fig. 3.9 Crane model designed in Gazebo simulation environment. The boom of crane has two types of rotations.

from the corrected data and repeat the optimization process until a desirable accuracy has been reached.

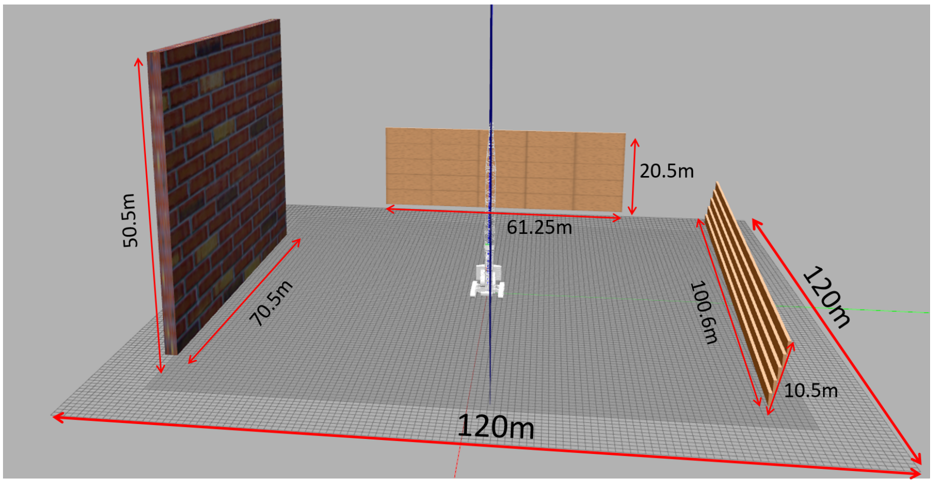


Fig. 3.10 The featureless environment consisting of only walls.

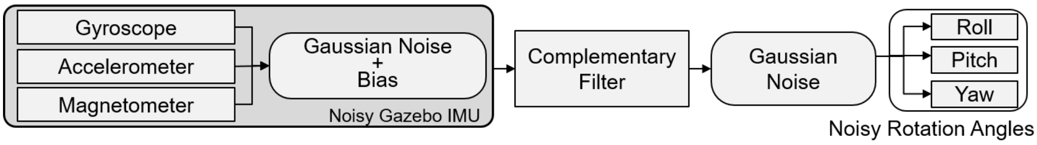


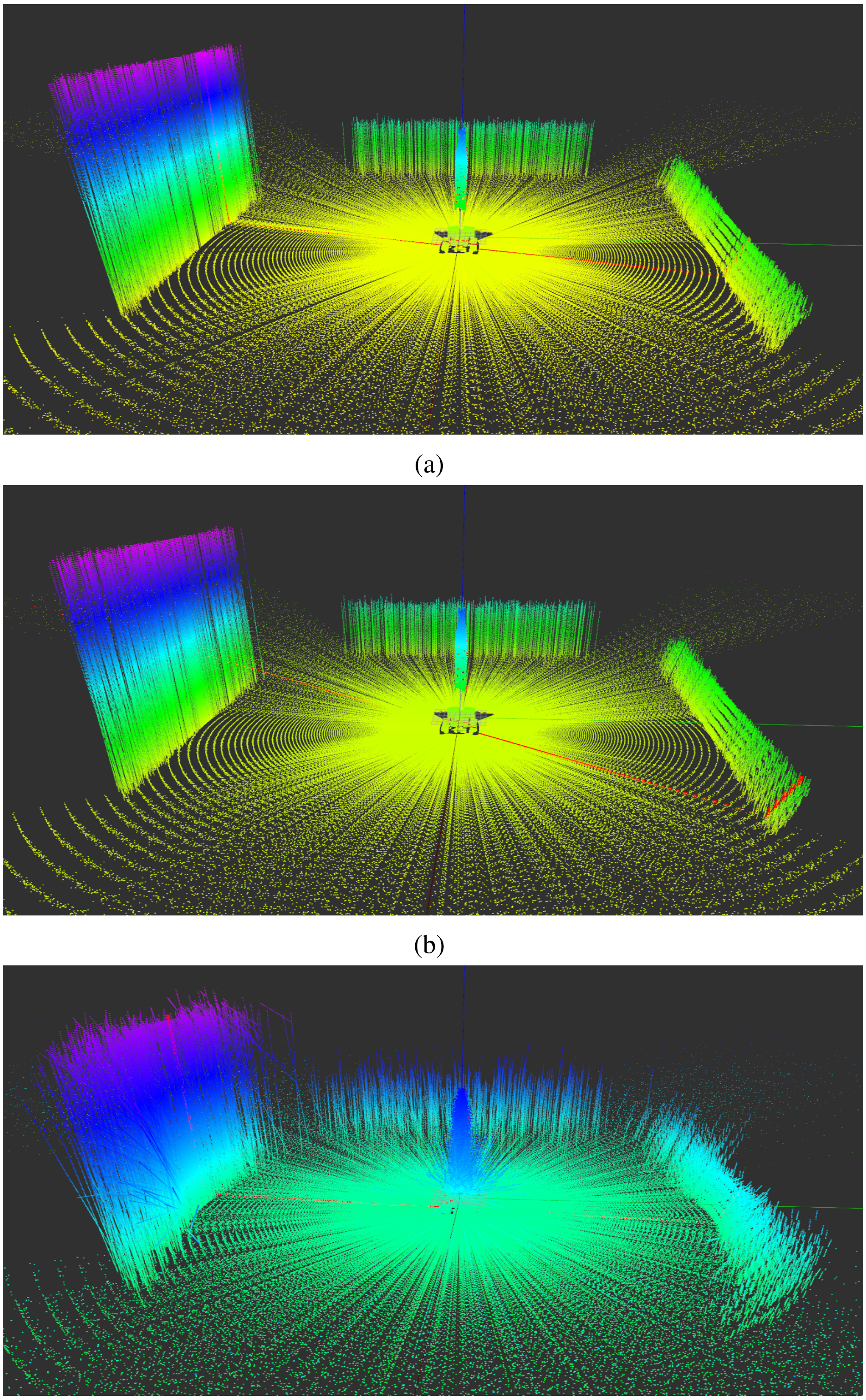
Fig. 3.11 IMU designed in Gazebo simulator by adding noise and bias.

Table 3.1 Different Level of Noise Added to IMU .

Noise Level	Noise in IMU design			Noise in final rotation	
	Gassusion Noise	Linear Bias	Angular Bias	Roll & Pitch (Degree)	Yaw (Degree)
level 1	0.01	0.01	0.01	0.5	1
level 2	0.01	0.01	0.01	1	2
level 3	0.05	0.05	0.05	1	2

3.5 Plane Extraction Results

Extraction of planes from a point cloud is executed across four distinct environments, each showcasing unique characteristics: Illustrated in Fig. 3.14 is the outcome of plane extraction within a simulated environment featuring simple walls. Fig. 3.15 displays the result of plane extraction in a simulated complex construction environment. Within the modeled indoor

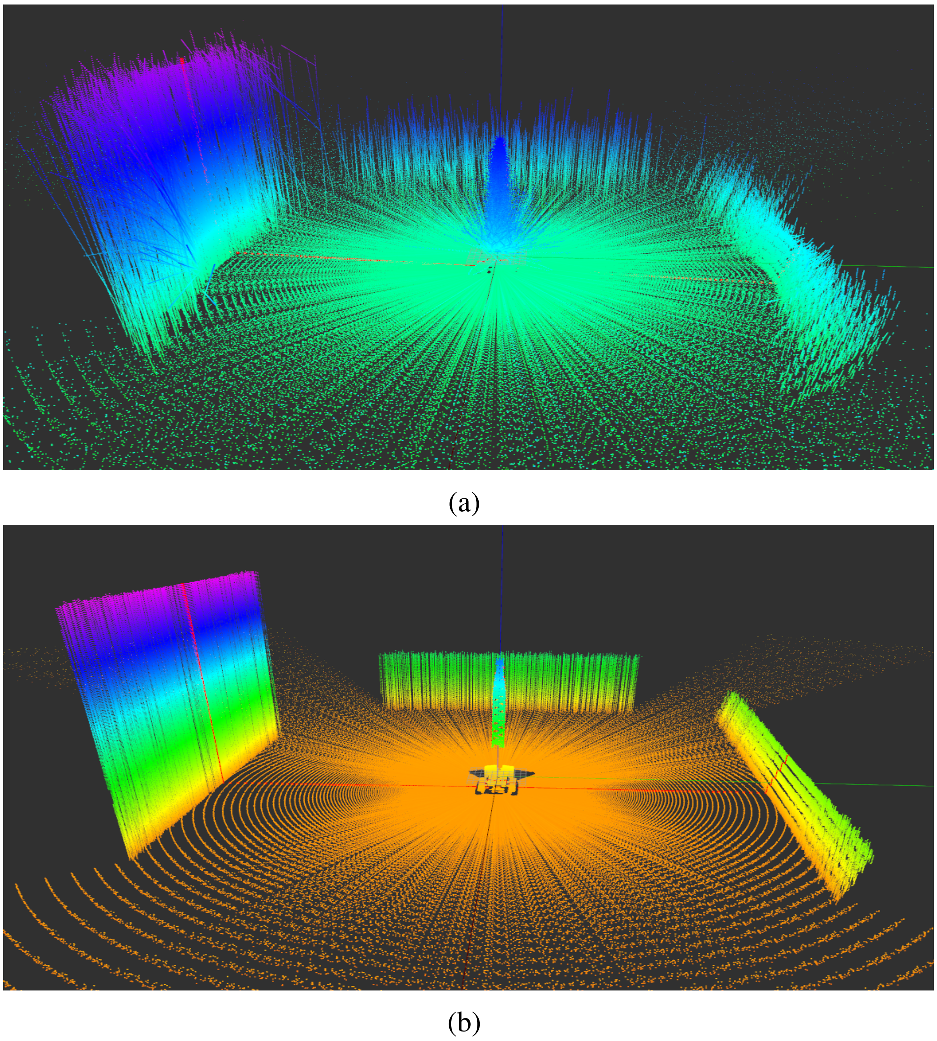


(a)

(b)

(c)

Fig. 3.12 3D point cloud map built using three different levels of IMU noise. The color indicates the height of each point. (a) 3D map built using IMU noise level 1. (b) 3D map built using IMU noise level 2. (c) 3D map built using IMU noise level 3. The map is distorted because of the high noise.



(a)

(b)

Fig. 3.13 Result of mapping correction method. The color represents the height of each point. (a) Before optimization the distorted point cloud. (b) After optimization the corrected point cloud.

crane environment, as depicted in Fig. 3.16, the extracted planes are exhibited. Fig. 3.17 portrays the plane extraction results within the outdoor Kobelco environment.

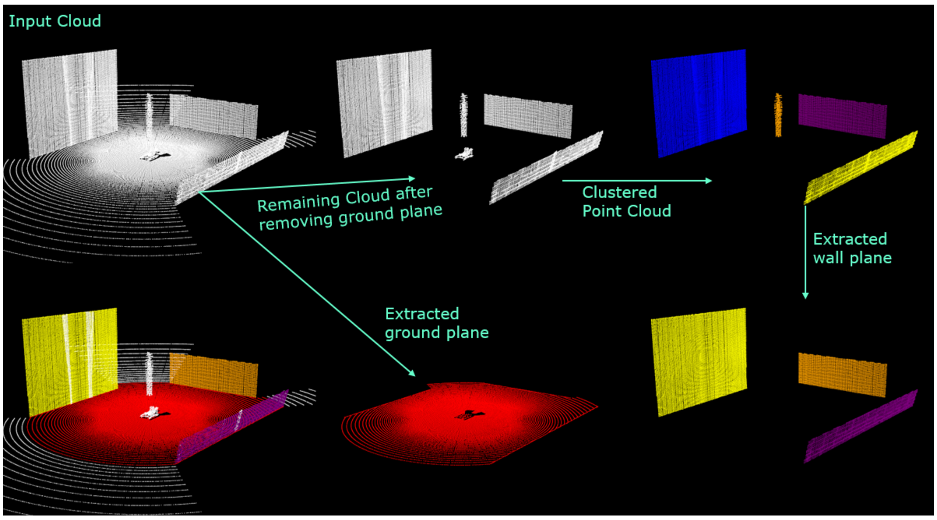


Fig. 3.14 Plane extraction results in a simulated environment with simple walls.

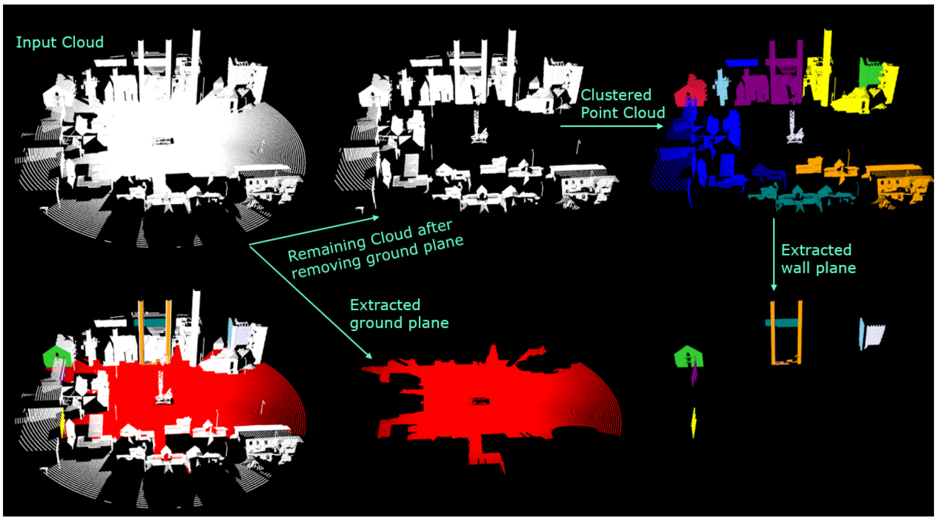


Fig. 3.15 plane extraction outcomes in a complex construction simulation environment.

The indices assigned to points on planes play a vital role in monitoring the 2D LiDAR scan positioned on the plane. These indices play a key role in establishing a direct correlation between the points within the plane and the respective data points in the 2D LiDAR scan. The objective of the plane extraction algorithm is not solely to recognize and extract planes within a point cloud but also to record the indices of all the 2D LiDAR scan points associated with each identified plane.

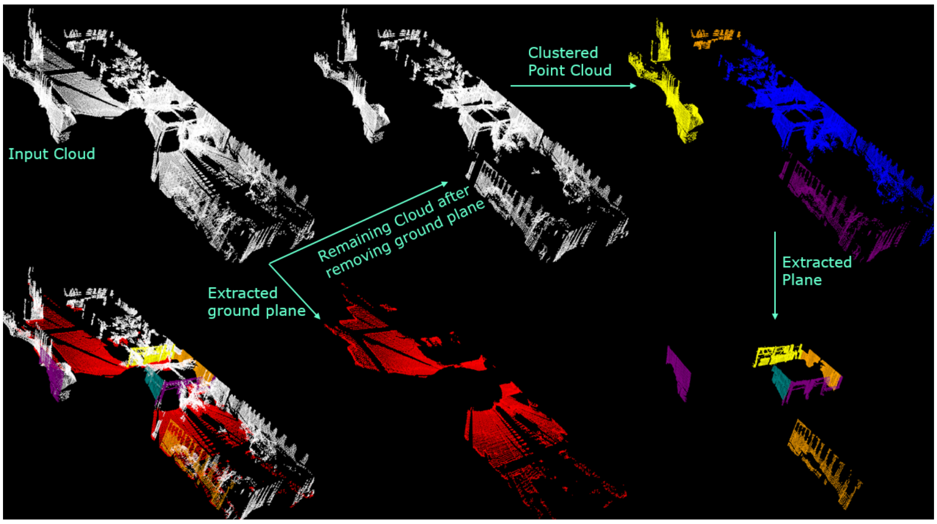


Fig. 3.16 Displaying the extracted planes within the modeled indoor crane environment.

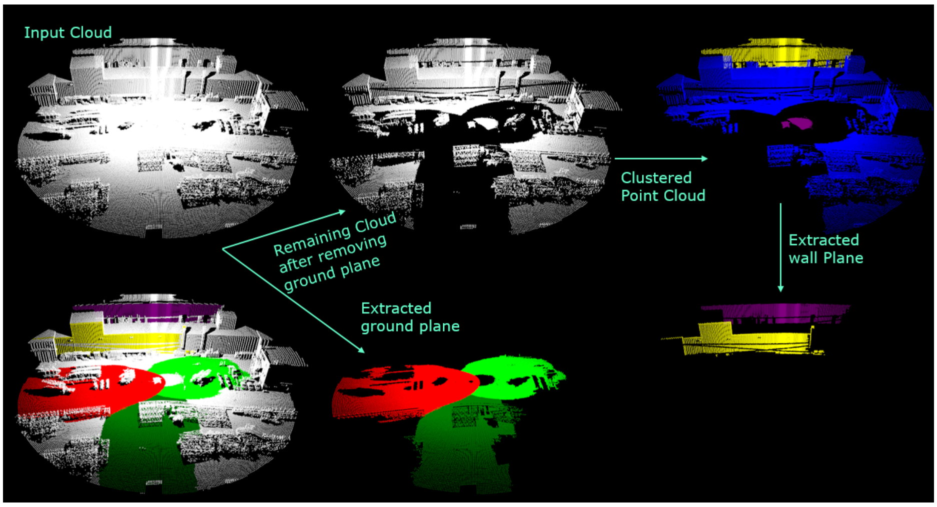


Fig. 3.17 Visual representation of plane extraction results within the outdoor Kobelco environment.

Fig. 3.18 illustrates the plane extraction process, showcasing the identification of planes within a simulated environment featuring simple walls. Additionally, it highlights all points from the 2D LiDAR scan line that belong to each plane. Moving forward, Fig. 3.19 provide a similar visualization, demonstrating the plane extraction process in the outdoor Kobelco environment. These figures emphasize the accurate identification and highlighting of all points associated with planes in the 2D LiDAR scan lines within the specified outdoor setting.

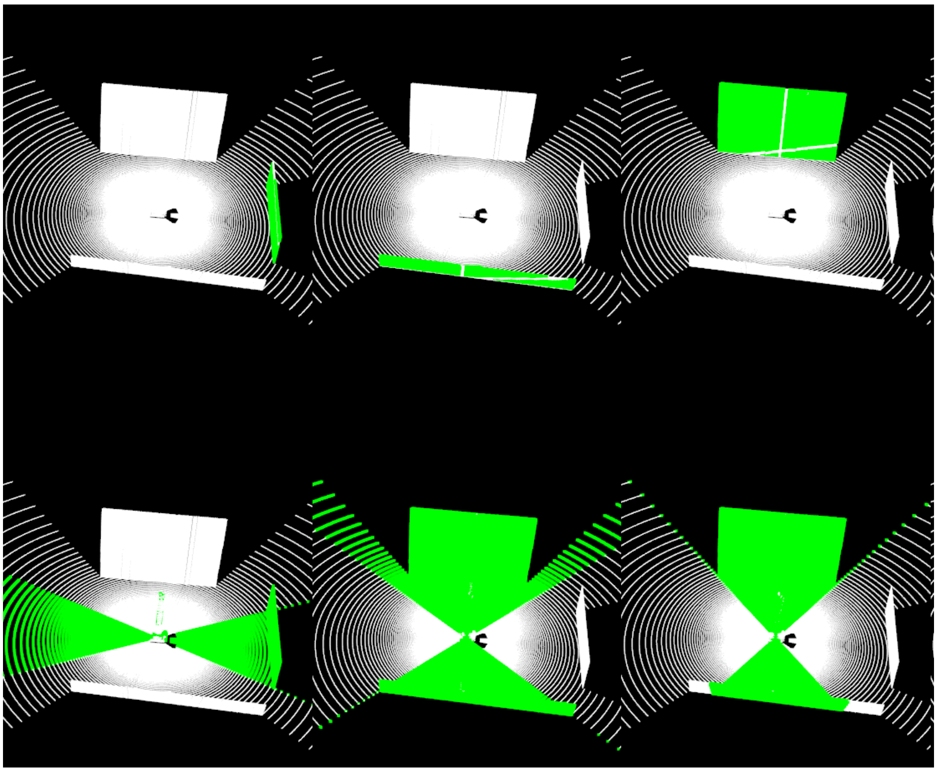


Fig. 3.18 Simulated Environment - Plane extraction and 2D LiDAR scan line points highlighting with simple walls.

3.6 Simulation Results

We first evaluated the proposed method in a simulation environment. For simulation, we used Gazebo simulator [25] and ROS environments. In Gazebo, a crane robotic model is designed as shown in Fig. 3.9. The size of the crane is shown in the Fig. 3.9. A crane boom can rotate along two axes: up-down and horizontal directions. ROS joint trajectory control is used to control both rotations. The sensor system comprising a 2D-lidar, a rotating base, and IMU is attached to the crane boom. We evaluated our proposed method in two different environments.

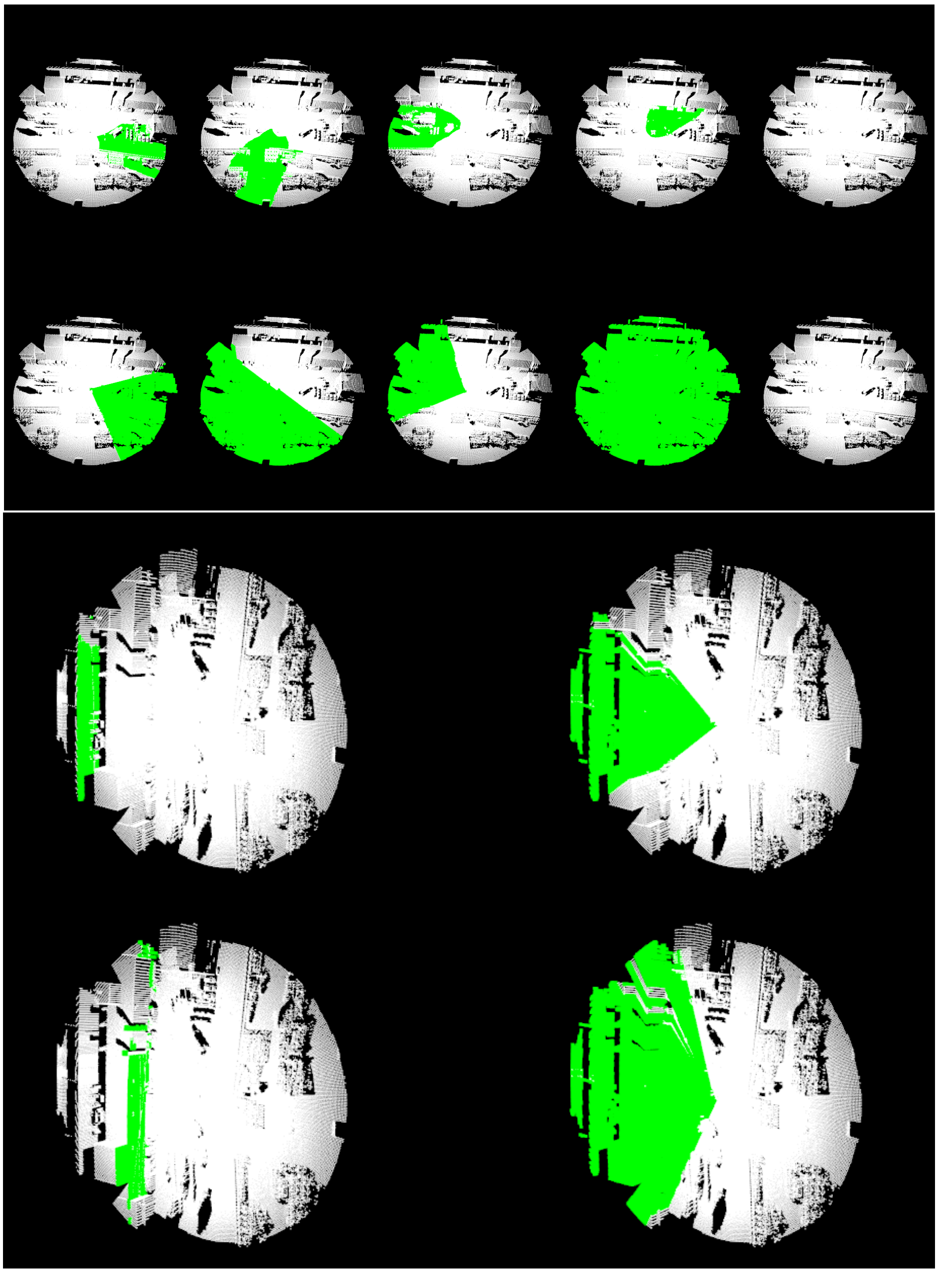
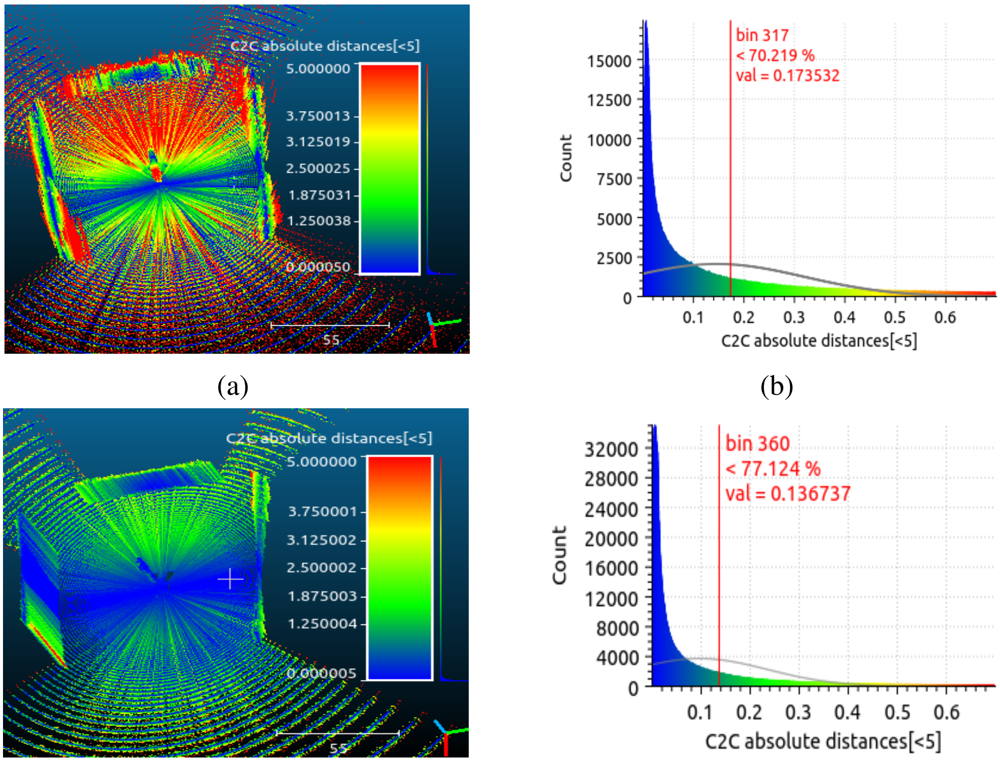


Fig. 3.19 Outdoor Kobelco - Plane extraction and 2D LiDAR scan line points highlighting in the outdoor environment.

The supplementary video of the results is available at https://youtu.be/UH8QB7AKUk4. The description of each environment and results are provided below.



(a) (b)

(c) (d)

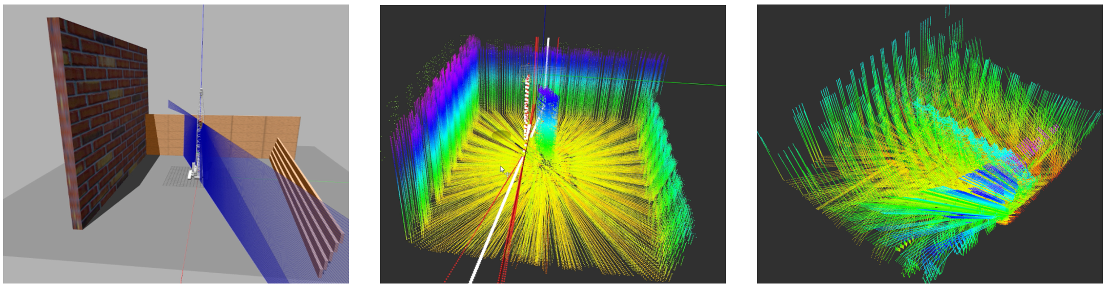
Fig. 3.20 Analysis the 3D mapping results in simulation environment-1 before and after mapping correction. The color represents the error (cloud-to-cloud distance between ground truth and point cloud obtained by the proposed method) (a) Point-to-point distances before mapping correction. (b) Distribution fitting of point-to-point distances before mapping correction. (c) Point-to-point distances after mapping correction. (d) Distribution fitting of point-to-point distances after mapping correction.

3.6.1 Simulation Environment-1

The first simulated environment is an open-sky featureless environment comprising simple three walls, as shown in Fig. 3.10. The space enclosed by all three walls is 120 m by 120 m. We scanned this space to build a 3D map of the environment using the proposed method. The crane having sensors on its boom is placed in the center of the space. The noise of the IMU affects the accuracy of the point cloud map. To evaluate the impact of noise on the point cloud map, different noise levels (Table 3.1) are added to the IMU, as shown in Fig. 3.11. We built 3D maps during an arbitrary motion of the crane boom. Fig. 3.12 shows the 3D

Table 3.2 Optimization performance after each iteration.

	Before Mapping Correction		After Mapping Correction					
			Iteration-1		Iteration-2		Iteration-4	
	Mean	GSSE	Mean	GSSE	Mean	GSSE	Mean	GSSE
GP	0.1613	437.16	0.02551	118.48	0.0152	41.44	0.0105	21.40
WP-1	0.1475	187.46	0.0555	81.64	0.0353	35.13	0.0123	10.95
WP-2	0.1491	153.84	0.0349	47.57	0.0241	21.24	0.0114	08.37

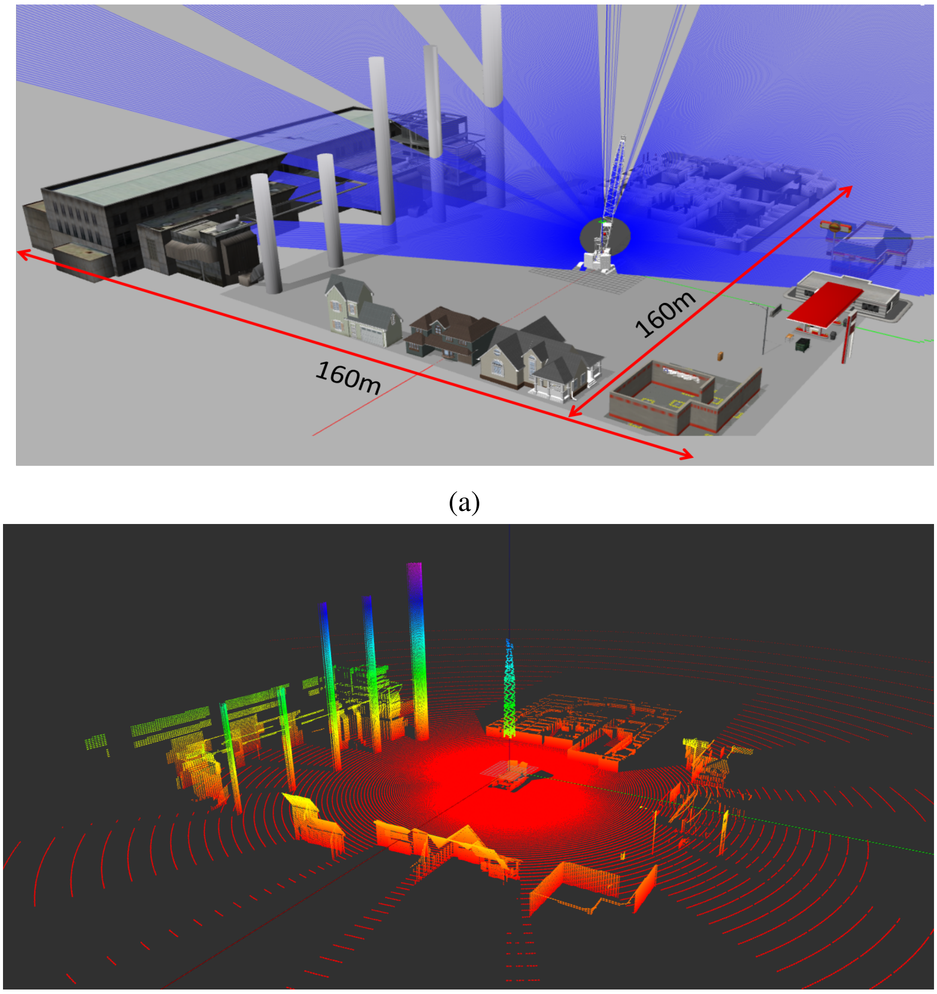


(a) (b) (c)

Fig. 3.21 The results of LOAM implemented in simulation environment-1. (a) Reduced area of simulation environment-1 by bringing walls closer to each other. (b) LOAM results during static crane boom. The color represents the height of each point. (c) LOAM results during moving crane boom. The color represents the height of each point.

point cloud maps built for different levels of noise. The color represents the height of each point. As the noise level increases (from level 1 to level 3), the distortion in the point cloud map also increases. For mapping correction in simulation, we used a threshold of 0.4 m and 0.01 m for the initial and the subsequent plane extraction, respectively. Fig. 3.13 compares the results before and after optimization-based mapping correction. The color represents the height of each point. The mapping correction reduces the effect of noise to build a corrected 3D map.

To evaluate the validity of our proposed method, the 3D map built by the proposed method is compared with the ground truth. The ground truth is obtained from the simulation model. The cloud-to-cloud distance (point-to-point distance) between both point cloud maps is calculated using cloud compare [15]. Fig.3.20 shows the result of the point-to-point distance before and after mapping correction using a color scale map. The blue color shows a smaller distance, while the red color represents a larger distance. From Fig. 3.20(a), before mapping correction, many points are in red and yellow indicating large errors. The error in the 3D point cloud map scatters randomly because it is due to random sensor noise. Fig. 3.20(b) shows the distribution fitting graph of point-to-point distances. After mapping correction, most points in Fig. 3.20(c) are blue and green, proving that the mapping correction approach was effective in reducing errors in the map caused by sensor noise. Thus the proposed method builds an accurate 3D map and minimizes errors. Fig. 3.20(d) shows the distribution fitting graph of point-to-point distances. It shows that the point cloud map created using the proposed method is close to the ground truth. Table 3.2 shows the mean of a plan to point distance for points belonging to the wall planes (WP-1 and WP-2) and a ground plane (GP) for iterations(1 to 4). It also shows the global sum of squared errors (GSSE) for these planes on each iteration (1 to 4). The GSSE is calculated as the sum of squared distances for all points belonging to the plane. The pose graph optimization-based data correction technique significantly reduces the GSSE for all planes in each iteration.

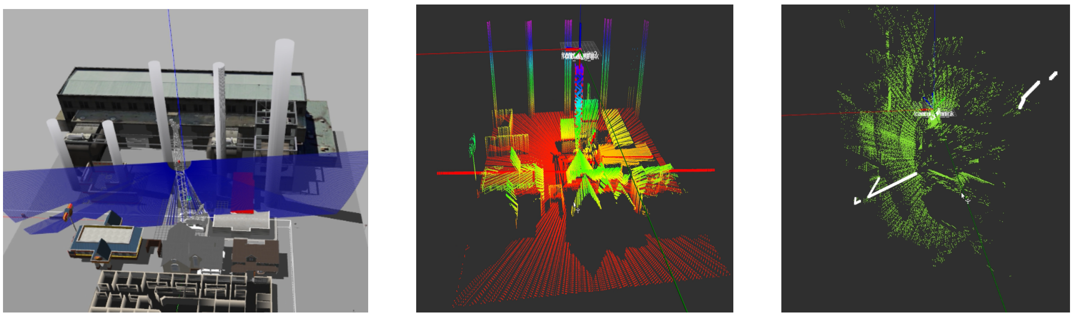


(a)

(b)

Fig. 3.22 Proposed method implemented in simulation environment-2. (a) Complex construction environment. (b) 3D point cloud map for complex construction environment. The color represents the height of each point.

We compare the proposed method with LOAM [90], which is a state-of-the-art 2D lidar and IMU-based mapping method. In order to implement LOAM on the crane system, the sensor system and rotating base parameters are configured in accordance with LOAM [90]. Because of LOAM's limited range for mapping, the area of simulation environment-1 (Fig. 3.10) is reduced by bringing walls closer to each other, as shown in Fig. 3.21(a). Fig. 3.21(b) and (c) show LOAM results for static and moving crane boom, respectively. The color represents the height of each point. We found that LOAM can create a 3D map when the crane boom is static but fails to build a 3D map when the crane boom is in motion. The reason for the failure of LOAM during crane boom motion is twofold. Firstly, the crane boom speed is higher than the rotating base speed, and the slowly-rotating lidar attached to the crane boom faces large changes in pose, and LOAM fails to get a consistent 3D point cloud and estimate such large pose changes. Secondly, the LOAM approach for estimating lidar orientation using IMU does not produce correct results. Due to heavy vibration in a crane's boom, the estimated orientation values fluctuate and sometimes diverge. This wrong orientation produces a distorted 3D map.

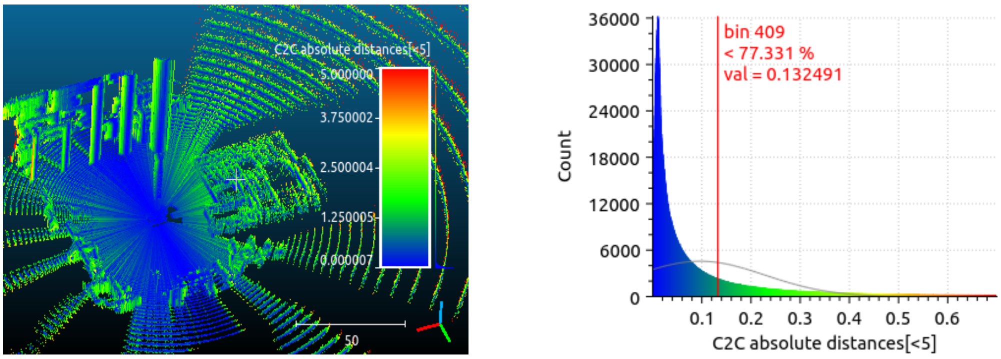


(a) (b) (c)

Fig. 3.23 LOAM method implemented in simulation environment-2. (a) Reduced area of simulation environment-2 by bringing building closer to each other (b)LOAM results during static crane boom. The color represents the height of each point. (c)LOAM results during moving crane boom.

3.6.2 Simulation Environment-2

The proposed method was also tested in another complex construction site environment shown in Fig. 3.22(a). The area of the construction site is 160 m by 160 m in size which is scanned to build the 3D map. Fig. 3.22(b) shows the 3D map after the mapping correction. In Fig. 3.22(b) the color represents the height of each point. Fig. 3.24(a) shows the point-to-point distance between the generated and the ground truth map, and Fig. 3.24(b) shows the distribution of point-to-point distances. The fact that the majority of the points in the 3D map



(a) (b)

Fig. 3.24 Analysis of simulation environment-2. The color represents the error (cloud-to-cloud distance between ground truth and point cloud obtained by the proposed method) (a) Point-to-point distances. (b) Distribution fitting of point-to-point distances.

are blue and some of them are green indicates that the distance between ground truth and the point cloud created by the proposed method is not the same all over the map. This error is due to the random noise in the sensor's measurement. However, the majority of green spots on the map, show that the map's overall error is very low, which proves that the proposed technique constructs an accurate 3D map. The distribution graph demonstrates that the cloud produced using the proposed framework is reliable and close to ground truth.

We also evaluate LOAM in this environment. We reduced the size of simulation environment-2 by moving the buildings closer to one another, as in the case of the first environment (see Fig. 3.23(a)). The results for a static and moving crane boom are shown in Fig. 3.23(b) and Fig. 3.23(c), respectively. The color represents the height of each point. We discovered that when the crane boom is static, LOAM results are better in this environment than in simulation environment-1 thanks to a sufficient number of features. However, LOAM fails to make a consistent map in the dynamic case.

Table 3.3 Optimization performance of mapping correction method.

	Before Map Correction			After Map Correction		
	Mean	SD	GSSE	Mean	SD	GSSE
GP	0.01494	0.01884	14.1373	0.01287	0.01809	12.8541
WP-1	0.03299	0.04431	55.3265	0.01787	0.02798	35.0048
WP-2	0.04463	0.04521	29.5326	0.04807	0.03850	28.6269

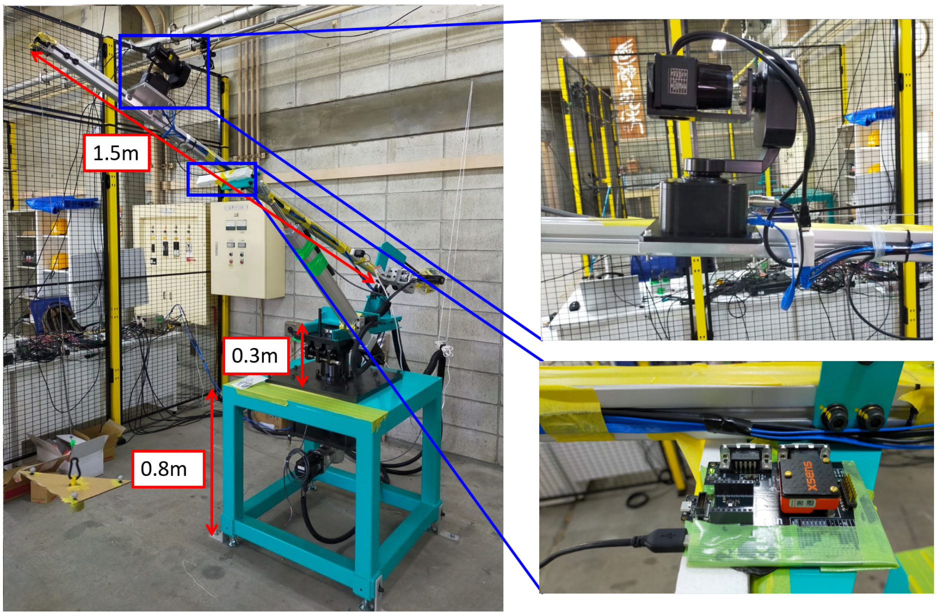


Fig. 3.25 Crane model and sensor system used for experiment are shown. The sensor system is attached to the crane boom.

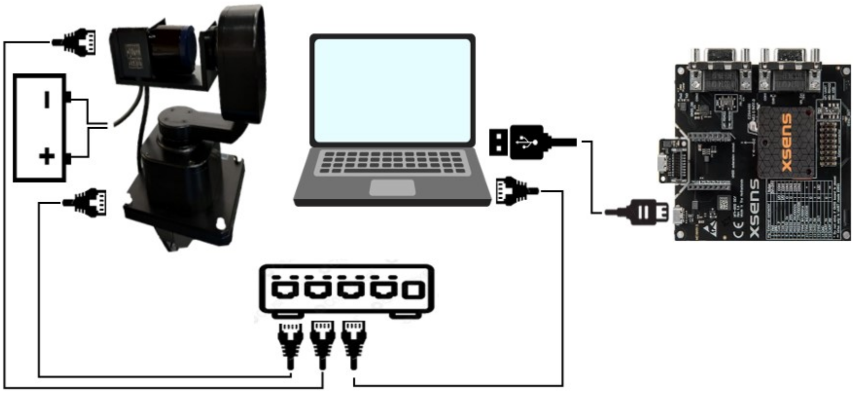


Fig. 3.26 Connection between sensors and PC.

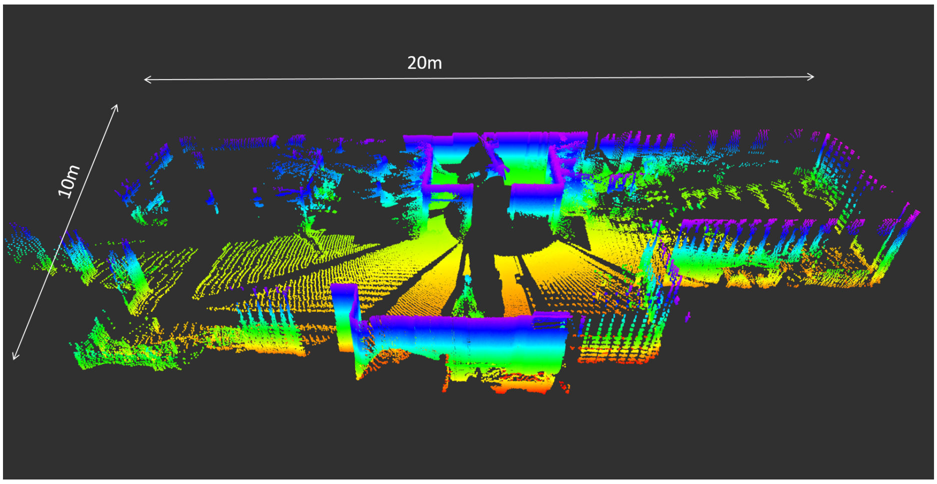
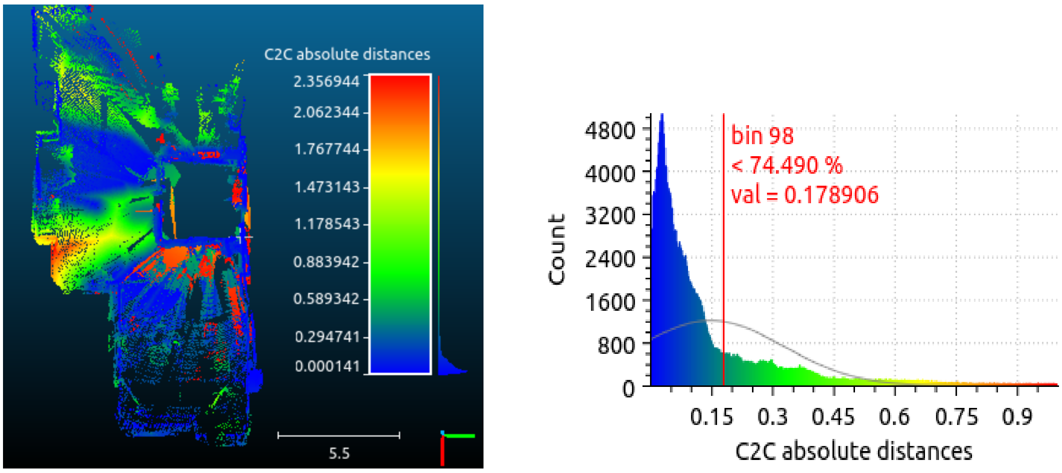


Fig. 3.27 3D point cloud map of environment built during experiment when crane boom is in continuous motion. The color represents the height of each point.



(a) (b)

Fig. 3.28 Result analysis of real world experiment. The color represents the error (cloud-to-cloud distance between ground truth and point cloud obtained by the proposed method) (a) point-to-point distance of two 3D maps (b) Distribution fitting of point-to-point distance.

3.7 Experiment Using Model Crane

The proposed method tested in real-world experiments on a model crane (construction machine) is shown in Fig. 3.25. The crane model was present in the indoor space which consists of a 20 m by 10 m area. In the experiment, we used a sensor system comprising a Hokuyo UST-20LX 2D lidar, an Orion Giken RHST-PA1L rotating base, and an XSENS MTI-630 IMU attached to the crane's boom. The connection between sensors and PC is shown in Fig. 3.26. It illustrates the interconnections, cables, and data flow pathways that facilitate the transfer of data from the sensors to the PC for further processing and analysis. For constructing a 3D map, the rotating base of the 2D lidar rotates at a speed of 6 deg/sec. For mapping correction, we employed a threshold of 0.6 m and 0.1 m for initial and final plane extraction, respectively. Fig. 3.27 shows the 3D point cloud map built when the crane's boom is in motion. In the figure the color represents the height of each point. As we can see in the figure, even when the sensor system is in continuous motion, we can obtain an accurate point cloud map.

The 3D map created using the proposed approach was compared to the ground truth map to evaluate the accuracy of our method. We use the map constructed while the lidar is static as a ground truth. Fig. 3.28(a) shows the point-to-point distance between two 3D maps using a color scale map showing the 0 to 2.5-m distance.The figure shows that most of the points in the point cloud map are blue, some are green, and there are very few red points. Red points in a few portions of the map show a large error. The reason for the large error is that the static lidar used for ground truth misses certain areas of the environment, whereas the boom-mounted lidar can scan a larger area when in motion. When we compute the distance between the ground truth and the generated point cloud, the points of those areas that are not available in the ground truth show a large error. The figure shows that the overall error is very low, so the proposed technique can create an accurate 3D map. Fig. 3.28(b) shows the distribution fitting plot of point-to-point distances between two clouds. The mean point-to-point distance is 0.1480 m, with a standard deviation of 0.17856, and 74.490% of points are less than 0.1789 m. Therefore, the point cloud map obtained using our method is close to the ground truth, with very little point-to-point distance between them.

We also used points to plane distances to evaluate the impact of our map-correcting approach. Table 3.3 displays the mean and standard deviation of plan-to-point distances for points belonging to the wall planes (WP-1 and WP-2) and a ground plane (GP). For these planes, it displays the global sum of squared errors (GSSE). The GSSE for all planes in each iteration is considerably reduced by the map correction.

To analyze the accuracy of the trajectory estimated using the proposed method, we computed the sensor system's trajectory using a motion capture system and treated it as a ground truth. In Fig. 3.29, the lidar rotation estimated using the proposed method is compared with the lidar rotation obtained using motion capture system. The figure shows that during arbitrary motion of the crane boom, the lidar attached to the boom faces rotation along three axes. The proposed method tracks the motion of the sensor system precisely.

We also performed another experiment by placing some boxes in the environment as shown in the Fig. 3.30 (a) and the 3D map is shown in Fig. 3.30 (b). The boxes placed in the environment and their corresponding 3D map are circled and marked with red arrows for emphasis.

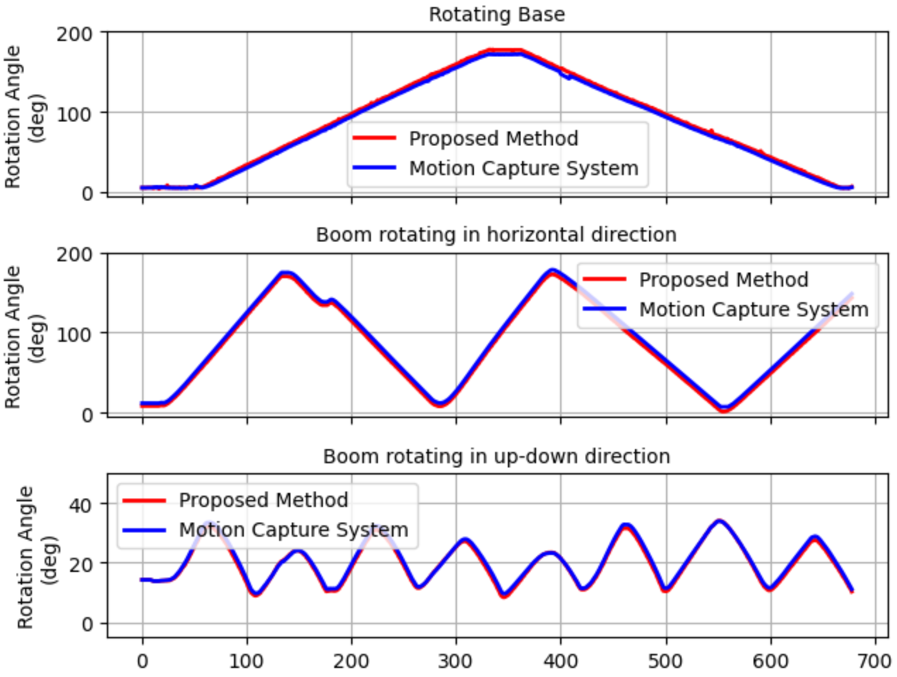


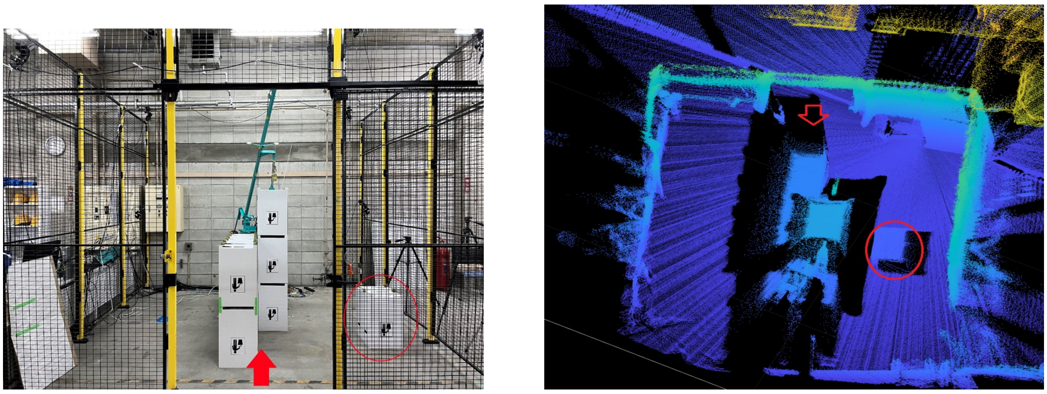
Fig. 3.29 Compare of oddomatry and optimization with motion capture system.

3.8 Experiment Using Real Crane

In this experiment, the proposed 3D mapping method is put to the test using a large crane. The detail of the experiment and result analysis is described as follows

(a) Sensors Setup

During the experiment, our setup involved utilizing a sensor system that consisted of a SICK LD-LRS 3611, 2D lidar, and an XSENS MTI-630 IMU. These sensors were securely



(a) (b)

Fig. 3.30 The boxes placed in the environment and their corresponding 3D map (a) Boxes placed in the environment (b) Corresponding 3D map of environment.

mounted onto the crane's boom for data collection and analysis as shown in Fig. 3.31. Fig. 3.32 presents the mechanical mounting of sensors on the crane's boom using specialized mounting plates. PC, and other devices (AC to DC converter, Ethernet switch, Power plug) which enables the transmission of data from the sensors to the PC are shown in Fig. 3.33.

(b) Experimental Setup

The large crane which is used the experiment and the location where the experiment was conducted is shown in Fig. 3.34, and Structural information of crane used in the proposed method for the 3D mapping is shown in Fig. 3.35.

(c) Data Acquisition Procedure and Mapping Result

We activate the sensors and record data while the crane boom is in motion. To test the robustness of our approach against variations in crane boom rotation speed, we conducted a series of experiments at different rotation speeds. Additionally, to analyze the effectiveness of the proposed method in handling different crane boom trajectories, we varied the trajectory during the experiments. The Table 3.4 presents the details of the crane boom rotation speed, boom pitch rotation, and boom yaw rotation cycles used in each experiment. The 3D mapping results of different experiments with varying crane boom yaw rotations at different speeds are presented in Fig.3.36 and Fig.3.37. Fig.3.36 shows the results of experiments conducted with the crane boom pitch angle set to approximately 40 degrees and Fig.3.37 shows the results when crane boom pitch angle was set at approximately 80 degrees.

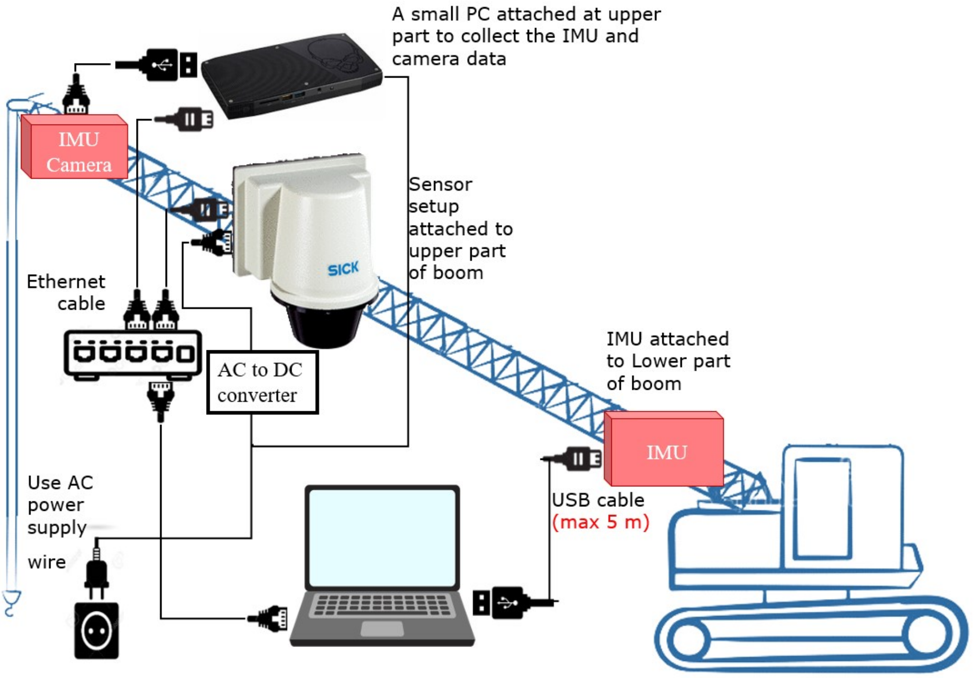

A small PC attached at upper part to collect the IMU and camera data
IMU
Camera
Sensor setup attached to upper part of boom
SICK
Ethernet cable
AC to DC converter
IMU attached to Lower part of boom
IMU
USB cable
(max 5 m)
Use AC power supply wire

Fig. 3.31 Sensors attached to crane.

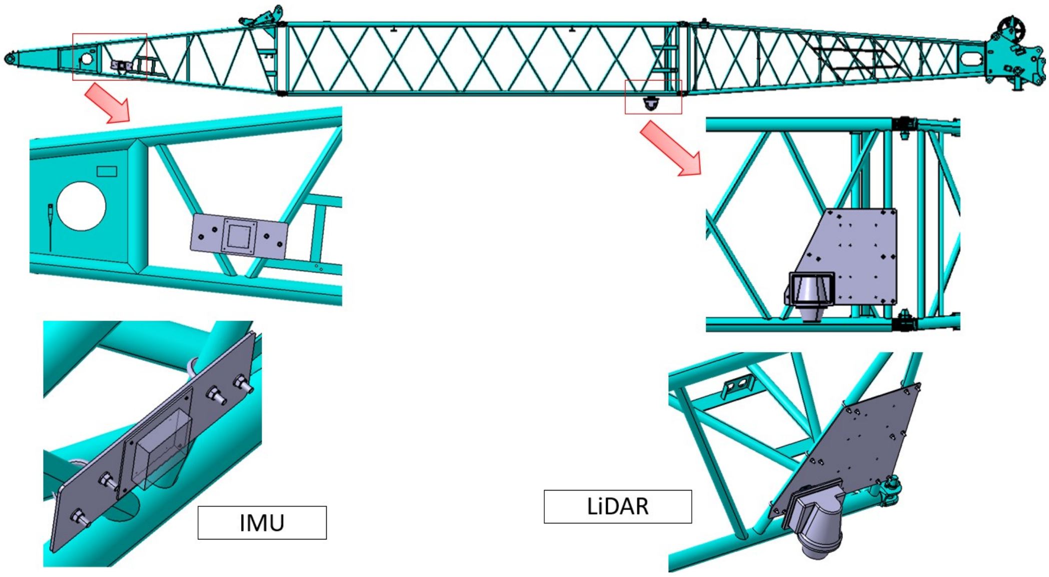


Fig. 3.32 Mechanical mounting of sensors on the crane's boom.

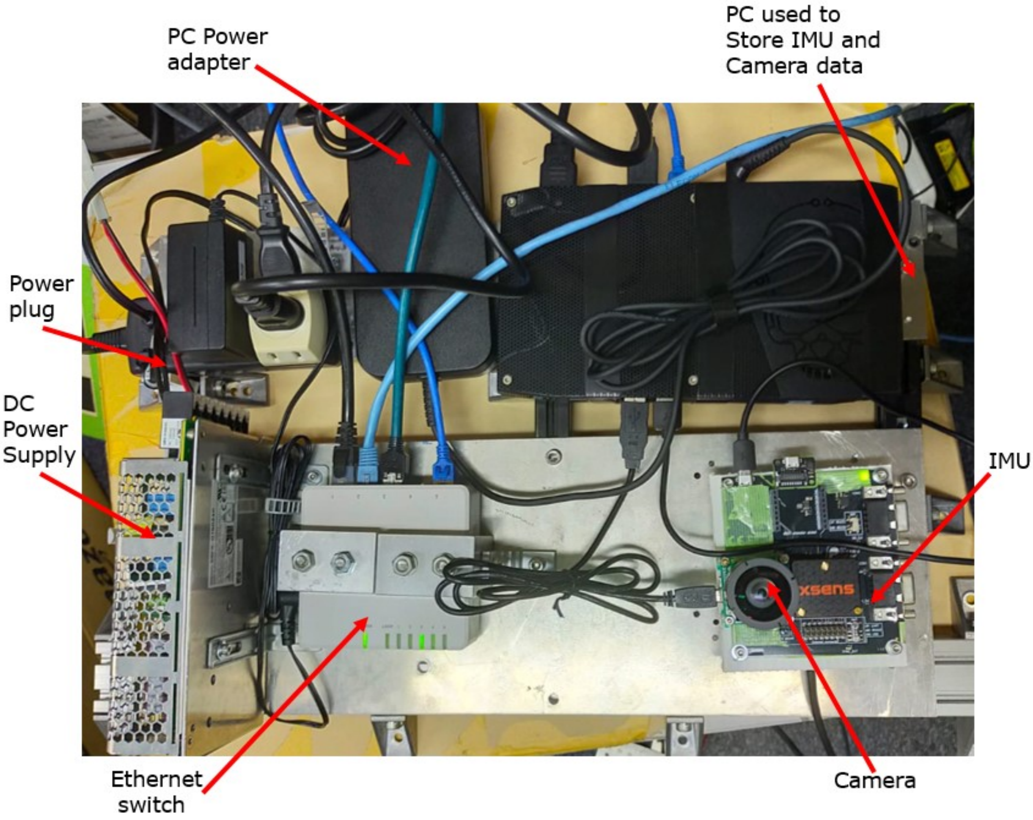


Fig. 3.33 Sensors attached to crane.

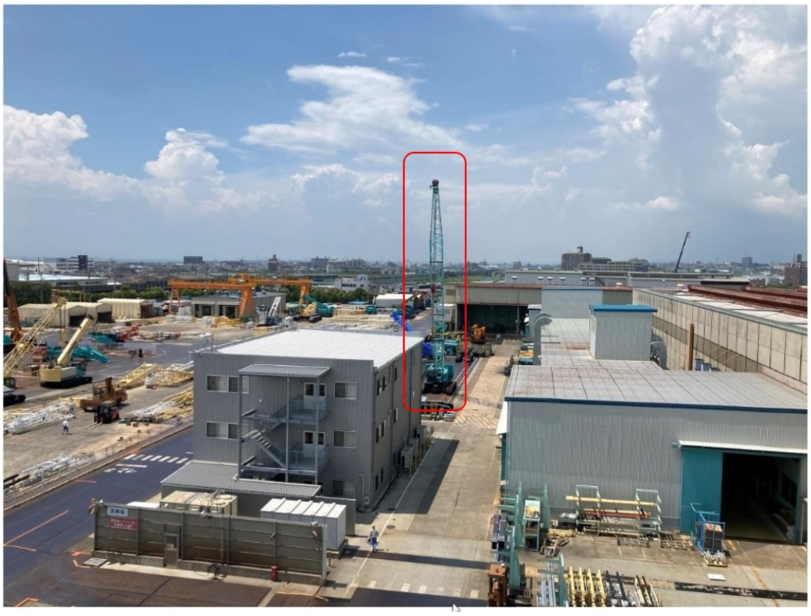
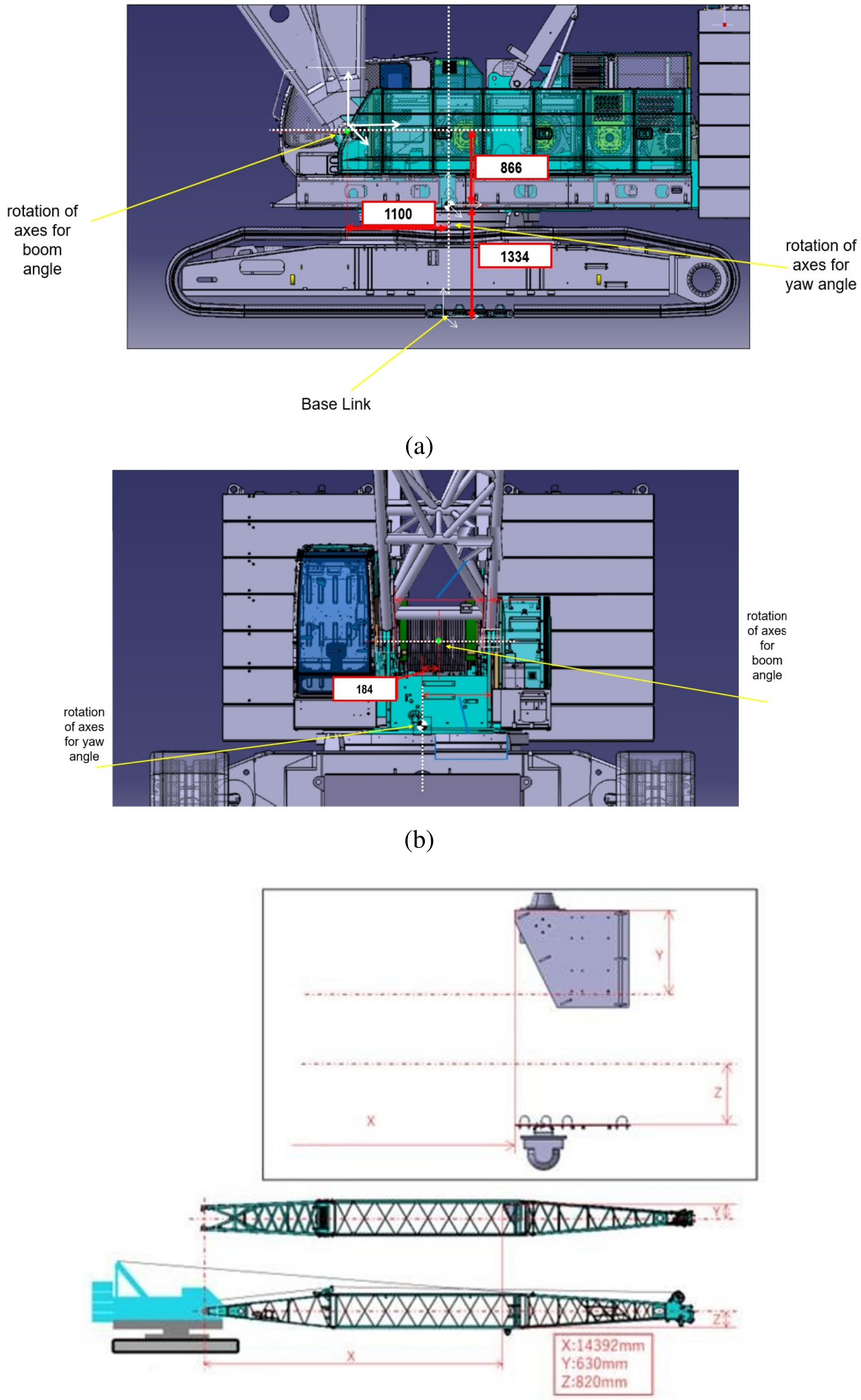


Fig. 3.34 The crane and the location where the experiment was conducted is shown.


rotation of axes for boom angle
866
1100
1334
rotation of axes for yaw angle
Base Link
rotation of axes for boom angle
184
rotation of axes for yaw angle
Y
Z
X
Y
Z
X
X:14392mm
Y:630mm
Z:820mm

(a)

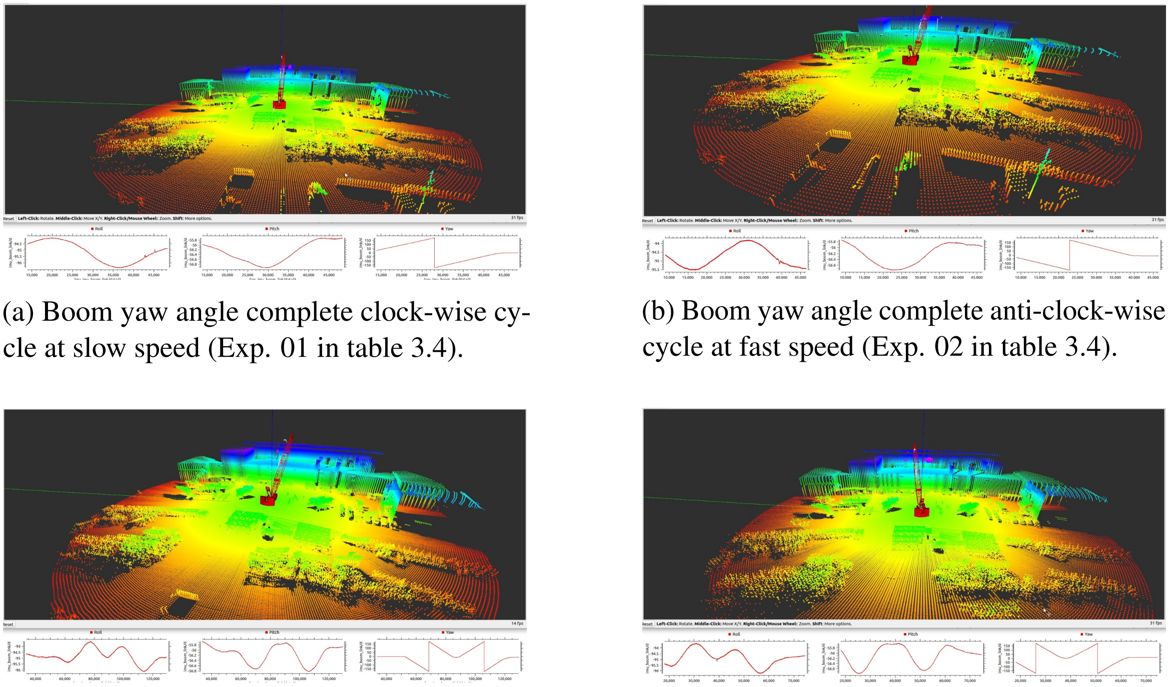
(b)

(c)

Fig. 3.35 Structural information of crane used in the proposed method for the 3D mapping.

Table 3.4 Details of Crane Boom Rotation Speed, Boom Pitch Rotation, and Boom Yaw Rotation Cycles.

Exp. No.	Crane Speed	Boom Position (Pitch Angle)	Boom Yaw Angle Rotation
1	Slow	Max. down (40 deg)	Complete clock-wise cycle
2	Fast	Max. down (40 deg)	Complete anti-clock-wise cycle
3	Fast	Max. down (40 deg)	2 Complete cycle (Clock-wise Anti-clock-wise)
4	Slow	Max. down (40 deg)	2 Complete cycle (Clock-wise Anti-clock-wise)
5	Slow	Max. up (80 deg)	Complete clock-wise cycle
6	Fast	Max. up (80 deg)	Complete anti-clock-wise cycle
7	Fast	Max. up (80 deg)	2 Complete cycle (Clock-wise Anti-clock-wise)
8	Slow	Max. up (80 deg)	2 Complete cycle (Clock-wise Anti-clock-wise)
9	Fast	Changing pitch angle	2 Complete cycle (Clock-wise Anti-clock-wise)
10	Slow	Changing pitch angle	2 Complete cycle (Clock-wise Anti-clock-wise)



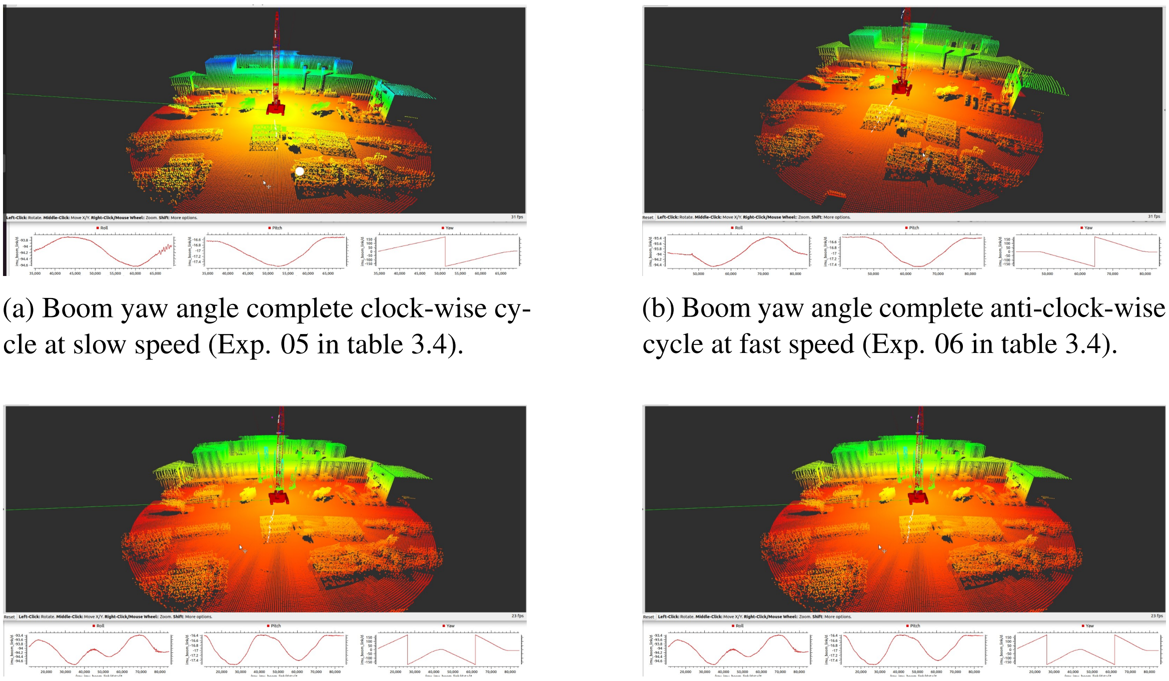
(a) Boom yaw angle complete clock-wise cycle at slow speed (Exp. 01 in table 3.4).

(b) Boom yaw angle complete anti-clock-wise cycle at fast speed (Exp. 02 in table 3.4).

(c) Boom yaw angle complete two cycle (clock-wise and anti-clock-wise) at fast speed (Exp. 03 in table 3.4).

(d) Boom yaw angle complete two cycle (clock-wise and anti-clock-wise) at slow speed (Exp. 04 in table 3.4).

Fig. 3.36 The 3D mapping results of experiments with varying crane boom yaw rotations at different speeds are presented. These experiments were conducted with the crane boom pitch angle set to 40 degrees.



(a) Boom yaw angle complete clock-wise cycle at slow speed (Exp. 05 in table 3.4).

(b) Boom yaw angle complete anti-clock-wise cycle at fast speed (Exp. 06 in table 3.4).

(c) Boom yaw angle complete two cycle (clock-wise and anti-clock-wise) at fast speed (Exp. 07 in table 3.4).

(d) Boom yaw angle complete two cycle (clock-wise and anti-clock-wise) at slow speed (Exp. 08 in table 3.4).

Fig. 3.37 3D mapping results of experiments involving diverse crane boom yaw rotations at various speeds. Throughout these experiments, the crane boom pitch angle was set at 80 degrees.

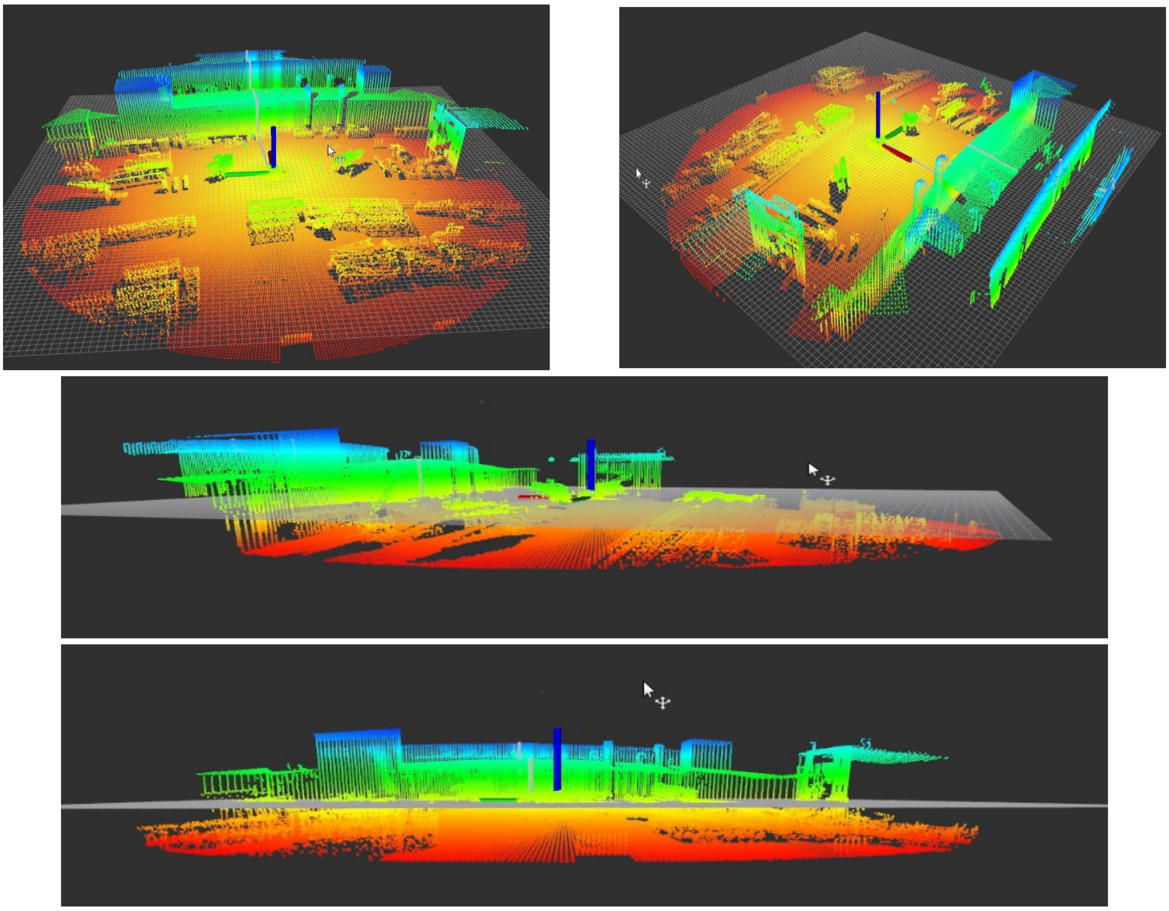


Fig. 3.38 3D Mapping results before optimization. The ground plane is curved, and the walls plane are bent.

(d) Results of Pose Graph Optimization

3D Mapping results before optimization is shown in Fig. 3.38. As we can see in the 3D map in Fig. 3.38, the ground plane is curved, and the walls are bent. It is due to accumulations of different types of errors, such as sensor noise, alignment errors, and inaccuracies in the structural information of the crane. To minimize these errors, we applied the proposed pose graph optimization using plane constraints. Different plane constraints used in pose graph optimization are shown in Fig.3.39, and 3D mapping results after optimization are shown in Fig.3.40, and we can see the 3D map is corrected and the ground plane and wall planes are straight now.

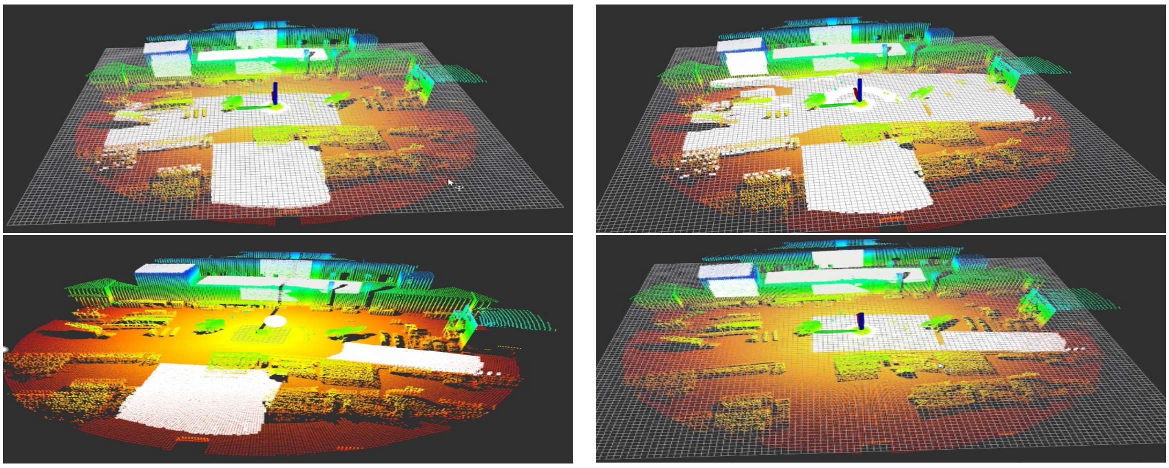


Fig. 3.39 Different plane constraints used in pose graph optimization are shown.

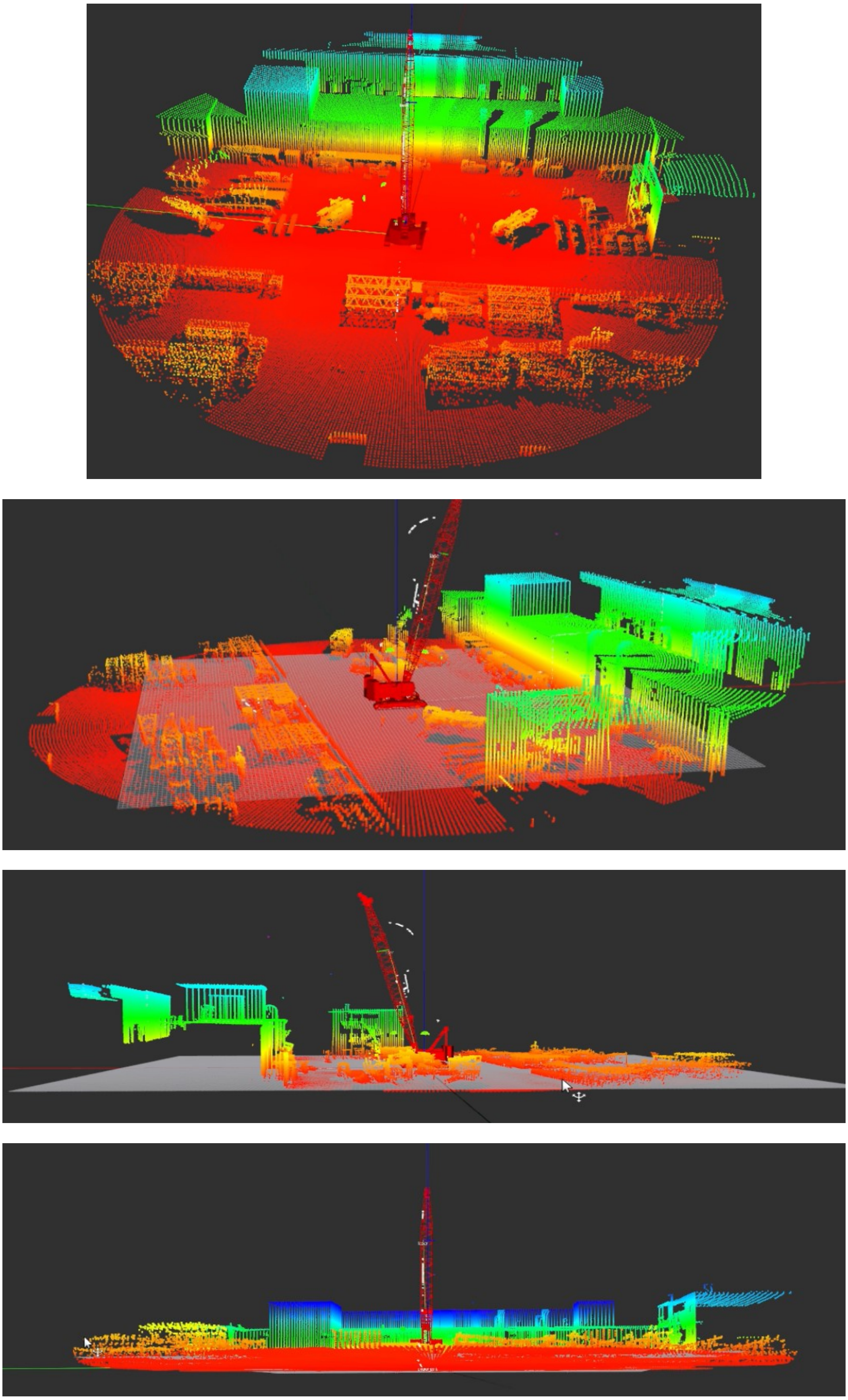


Fig. 3.40 3D mapping results after optimization. The 3D map is corrected and the ground plane and wall planes are straight.

3.9 Conclusion

This chapter presented a unique technique for large-scale 3D mapping for cranes using a slowly-rotating lidar and an IMU attached to the crane boom. The method can generate an accurate 3D map under arbitrary crane motion during lidar scanning. We use a complementary filter in series with moving average filtering, combined with the structural information of the crane, to estimate the sensor pose at each scan, even under the boom vibration. Using the estimated sensor poses, we convert a set of 2D scans into a 3D point cloud map. To further improve the map, we also developed a new pose graph optimization approach that extracts planar structures in the environment and introduces them as additional nodes in the pose graph. We evaluated the proposed method in simulation and real-world experiments. The experimental results show that our method can effectively estimate the sensor trajectory and build an accurate 3D point cloud map and outperforms one of the state-of-the-art methods. In the current implementation, pose estimation using IMU can run in real-time, while the map correction part takes a long time for large-scale mapping. Developing a more efficient map correction algorithm is future work.

Chapter 4

IMU-based Neural Network Approach for Real-time Sensor Pose Estimation and 3D Mapping

4.1 Introduction

This chapter introduces a method for neural network-based real-time pose estimation using an IMU (inertial measurement unit) and its application in large-scale 3D mapping using a slowly rotating 2-D LiDAR. In this method, a neural network consisting of a convolutional neural network (CNN) and long short-term memory (LSTM) is employed to estimate the change in pose. Firstly, online pre-filtering using a low-pass filter is implemented on the time windows of IMU measurements before feeding them as the input to the neural network to estimate the change in position and rotation of the sensor. After that, the estimated sensor pose is used to register the scans of 2D-rotating LiDAR to build a large-scale 3D map. The proposed method is tested in a gazebo environment by attaching the sensors to a crane boom. In this study, we also investigate the impact of different time windows of IMU measurements on the accuracy of pose estimation by the neural network.

In this chapter, first, we introduced online window-based low-pass filtering on IMU measurements before passing these measurements to neural network[71] to minimize the effect of sensor vibration and sensor noise and to get more accurate odometry results. Secondly, we modified the method of IMU data flow in the neural network proposed in[71] to get real-time odometry and implemented it for the 3D-mapping applications. We also analyze the effect of the window size of a sequence of IMU measurements for data flow in a

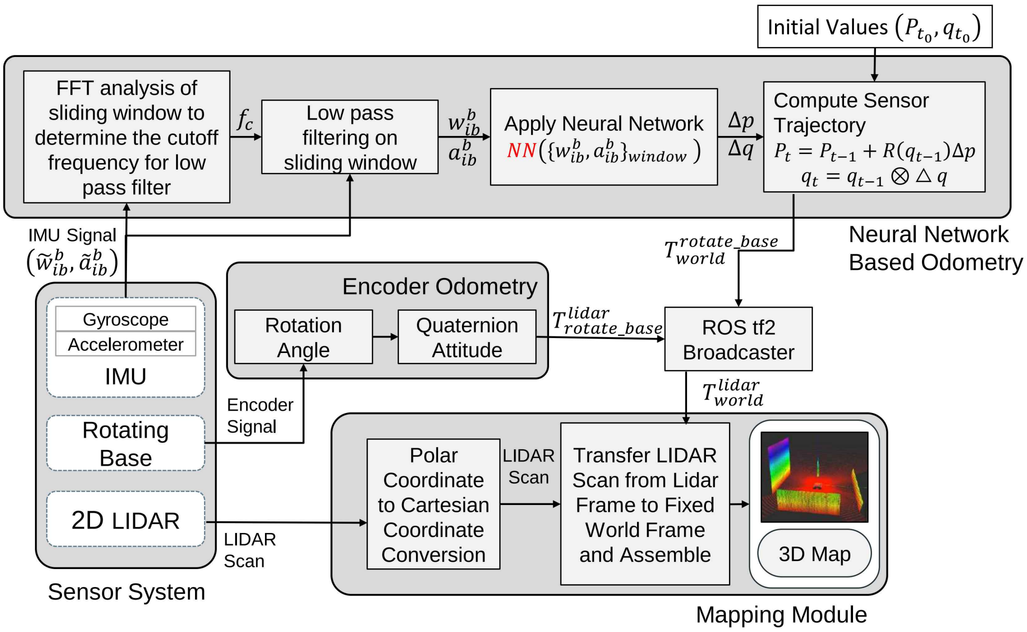

Initial Values (P_{t_0}, q_{t_0})
FFT analysis of sliding window to determine the cutoff frequency for low pass filter
f_c
Low pass filtering on sliding window
w_{ib}^b
a_{ib}^b
Apply Neural Network
$NN(\{w_{ib}^b, a_{ib}^b\}_{window})$
Δp
Δq
Compute Sensor Trajectory
$P_t = P_{t-1} + R(q_{t-1})\Delta p$
$q_t = q_{t-1} \otimes \Delta q$
Neural Network Based Odometry
IMU Signal
$(\tilde{w}_{ib}^b, \tilde{a}_{ib}^b)$
$T_{world}^{rotate\_base}$
Encoder Odometry
Rotation Angle
Quaternion Attitude
$T_{rotate\_base}^{lidar}$
ROS tf2 Broadcaster
T_{world}^{lidar}
Gyroscope
Accelerometer
IMU
Rotating Base
2D LIDAR
Sensor System
Encoder Signal
LIDAR Scan
Polar Coordinate to Cartesian Coordinate Conversion
LIDAR Scan
Transfer LIDAR Scan from Lidar Frame to Fixed World Frame and Assemble
3D Map
Mapping Module

Fig. 4.1 Block diagram of proposed method.

neural network on the accuracy of estimation. We trained different models with different size windows and tested those models to compare the accuracy of predicted odometry.

The main contributions of this work are

- We applied real-time filtering to IMU readings prior to using them in a neural network to minimize the noise and vibration impacts in tough environments.

- We analyze the effect of different time windows for IMU measurements on the accuracy of pose estimation.

- We build a large-scale real-time 3D mapping using slowly-rotating 2D-LiDAR.

4.2 Overview of proposed mapping framework

The proposed technique uses a sensor setup that includes 2D-LiDAR, IMU, and a rotating base to create a 3D map of the environment. The 2D-LiDAR is fixed on a rotating base and IMU is placed near a rotating base. This sensor system is attached to the boom of a crane. A boom of a crane moves during performing any task so our goal is to build a 3D map during that motion of the sensor system.

Fig. 4.1 shows the block diagram of proposed method. In proposed method a window consists of predefined number of sample of accelerometer $\tilde{a}_{ib}^b$ and gyroscope $\tilde{w}_{ib}^b$ is used for prefiltering. First FFT analysis is implemented on a sliding window to compute the cutoff frequency f_c for the low-pass filter. After that low-pass filter is applied to noisy IMU and cutoff frequency, the change in position Δp and rotation, Δq of rotating-base is estimated by applying neural network to the filtered window data. The pose of the rotating base in form of a position vector and quaternion at the current time t is estimated by transferring the Δp and Δq from the IMU body frame to the fixed-world frame using the following Eq.

$$P_t = P_{t-1} + R(q_{t-1}\Delta p), \tag{4.1}$$

$$q_t = q_{t-1} \otimes \Delta q, \tag{4.2}$$

where p_{t-1} and q_{t-1} is previous position and orientation and p_t and q_t is the current position and orientation. The $R(q)$ is the rotation matrix for q, and $\otimes$ is the Hamilton product. The encoder of the rotating base provides the angle of rotation of the LiDAR sensor with respect to the rotating base.

The rotating angle is converted to quaternion to get the transformation from the rotate-base frame to the LiDAR frame. The transformation of the LiDAR frame to a rotating-base frame and transformation of the rotate-base frame to a fixed-world frame is broadcasted by ROS-tf2. The LiDAR scan is first converted from polar coordinate to cartesian coordinate and then transferred to the fixed-world frame. Finally, the LiDAR scans in the fixed-world frame are assembled to gather to get a 3D map of the environment.

4.3 Real-Time 6-DOF Odometry Using Neural Network

For 6-DOF odometry estimation, we use the same network architecture as described in [71], but we modify the approach for IMU data flow in the network architecture. In [71], a window of 200 frames of 3-axis accelerometer a and gyroscope w is used as input, and the relative pose between the 95th frame and 105th frame of the window is used as output. It is mentioned in [71] that both past 100 and future 100 IMU measurements frames impact the regressed relative posture, but in reality, the future IMU measurements cannot affect the past relative pose; only past IMU measurements have an impact on future pose estimation. Another issue in the data flow approach given in [71] is that it cannot be used for real-time pose estimation. For example, for an IMU with a frequency of 100Hz, if we provide past 200 samples of IMU measurements from the current time t to $t-2$ as input to a neural network, we get the relative pose of the past time $t-1$. To solve this issue and obtain real-time pose from the network, we use two different approaches, as detailed in Section 4.3.3.

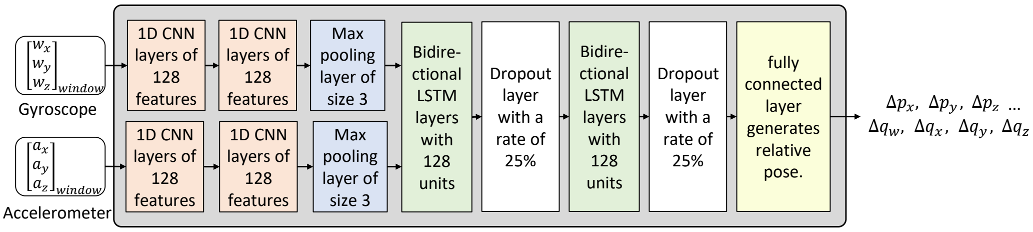

$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}_{window}$
Gyroscope
$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{window}$
Accelerometer
1D CNN layers of 128 features
1D CNN layers of 128 features
Max pooling layer of size 3
1D CNN layers of 128 features
1D CNN layers of 128 features
Max pooling layer of size 3
Bidire-ctional LSTM layers with 128 units
Dropout layer with a rate of 25%
Bidire-ctional LSTM layers with 128 units
Dropout layer with a rate of 25%
fully connected layer generates relative pose.
$\Delta p_x,\ \Delta p_y,\ \Delta p_z\ ...$
$\Delta q_w,\ \Delta q_x,\ \Delta q_y,\ \Delta q_z$

Fig. 4.2 Network architecture used in proposed method.

4.3.1 Low Pass Filter

In our case IMU sensor is attached to the boom of the crane and faces a lot of vibrations which affects the IMU measurements so to minimize the noise of IMU measurements, a low pass filter is designed as given in [73]. This step is data pre-processing that aims to provide a cleaner version of sensory input by removing errors caused by vibration, mechanical, electrical, and signal processing flaws. The window of samples of IMU measurements is passed through a low pass filter before using it as an input to the neural network. In lower pass filter signals with frequencies lower than a specific cutoff frequency are passed through, whereas signals with frequencies greater than the cutoff frequency are attenuated. Online FFT analysis is implemented on the IMU samples window to estimate the cutoff frequency for that window.

4.3.2 Network Architecture

As demonstrated in Fig. 4.2, the network is built on CNN combined with LSTM, which gives excellent performance for problems that require sequence processing [35]. The input is IMU data in a window of a pre-defined number of frames, each frame consists of 3-axis angular velocity and 3-axis acceleration. 1D convolutional layers with 128 features and a kernel size of 11, first process the gyroscope and accelerometer data separately. A max-pooling layer with a size of 3 is utilized after two convolutional layers. The output of these layers is concatenated and fed to 128-unit bidirectional LSTM layers. The network used a two-layer stacked LSTM model, in which a bidirectional LSTM produces a full sequence, which is then fed into a second bidirectional LSTM. To avoid overfitting, a dropout layer with a 25% rate is inserted after each LSTM layer. Finally, a fully connected layer provides an output of relative pose in form of position vector and quaternion.

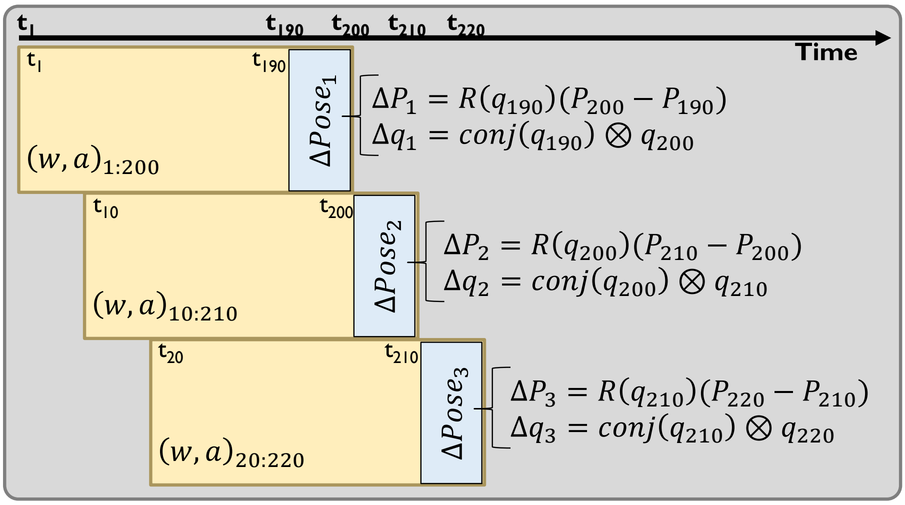

t_1
t_190
t_200
t_210
t_220
Time
t_1
t_190
(w, a)_1:200
ΔPose_1
ΔP_1 = R(q_190)(P_200 − P_190)
Δq_1 = conj(q_190) ⊗ q_200
t_10
t_200
(w, a)_10:210
ΔPose_2
ΔP_2 = R(q_200)(P_210 − P_200)
Δq_2 = conj(q_200) ⊗ q_210
t_20
t_210
(w, a)_20:220
ΔPose_3
ΔP_3 = R(q_210)(P_220 − P_210)
Δq_3 = conj(q_210) ⊗ q_220

Fig. 4.3 windows size of 200 samples used to feed data into the neural network.

4.3.3 IMU Data Flow in Neural Network

We employed two distinct approaches to generate windows of successive IMU measurements in order to feed data into the neural network. The first approach is based on windows consisting of repeated IMU measurements as shown in Fig. 4.3. We can see IMU measurements from t_{10} to t_{200} used in the first window are repeated in the second window. In this approach, a stride of 10 new IMU measurements is added to each new window. The relative pose to be regressed from this given window is the one that occurred between the newly added 10 frames of IMU measurement (frame number 190 and frame number 200 if a window size of 200 samples is used). In this case, we can get the current change in pose after every 10 frames.

In our second approach, IMU measurements are not repeated in windows. Every window use new IMU measurements as shown in Fig. 4.4. The relative pose between the first frame and the last frame of the window is the pose to be regressed from the given window. The drawback of this approach is that it introduces a delay in updating the relative pose, because IMU measurements are not repeated we have to wait for new IMU measurements for every new window. For training in this approach, a lot of IMU data is needed.

Both approaches, the windows based on repeated IMU measurements (Fig. 4.3) and windows do not use repeated IMU measurements (Fig. 4.4) are tested with different window sizes such as 26, 50, 200. Mostly in literature [71, 85, 12] a window size of 200 samples are used as input.

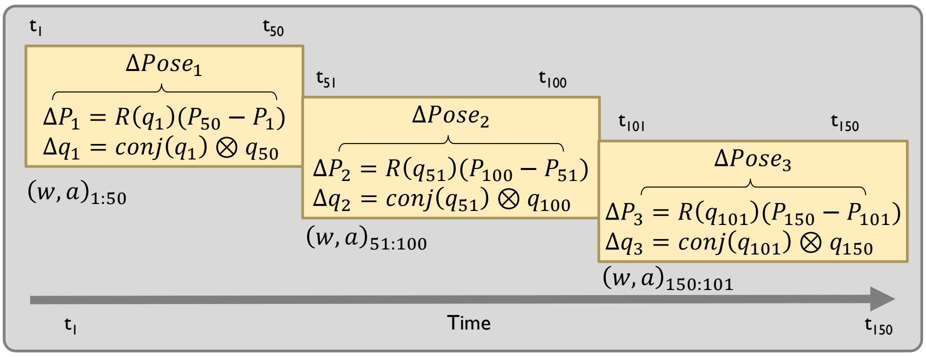


Fig. 4.4 windows size of 50 samples used to feed data into the neural network.

As seen in Fig. 4.3 and Fig. 4.4 the translation vector and quaternion based format is utilized for relative pose which is computed using pervious and current position (P_{t-1}, P_t) and orientations (q_{t-1}, q_t) associated with a specific IMU data window as follows:

$$\Delta P = R(q_{t-1})(P_t - P_{t-1}) \tag{4.3}$$

$$\Delta q = conj(q_{t-1}) \otimes q_t \tag{4.4}$$

4.3.4 Pose Distance Metric

In our case quaternions is used for 6-DOF pose representation so loss function is the geometric difference between the ground truth pose (p,q) and the predicted pose $(\hat{p},\hat{q})$ which is defined below as follows [71]

$$L_{PE} = \|\hat{P} - P\| \tag{4.5}$$

$$L_{QE} = 2.\|imag(\hat{q} - conj(q))\| \tag{4.6}$$

where L_{PE} and L_{QE} is loss function for position and quaternion. We used multi-task learning for the Metric Balancing as described in [71].

4.3.5 Sensor Data Collection For Training

To train the neural network, data of accelerometer and gyroscope of IMU, and data of change of position and rotation in terms of a quaternion are required. For data collection, a trajectory for the motion of the boom of the crane is designed. The trajectory consists of two rotations: one is yaw rotation and another is pitch rotation of crane boom as shown in Fig. 4.5. The algorithm for a trajectory is given in Alg. 1. ROS Joint Trajectory Action is used to control the trajectory of the boom of the crane in the Gazebo. Three-hour data is collected during the

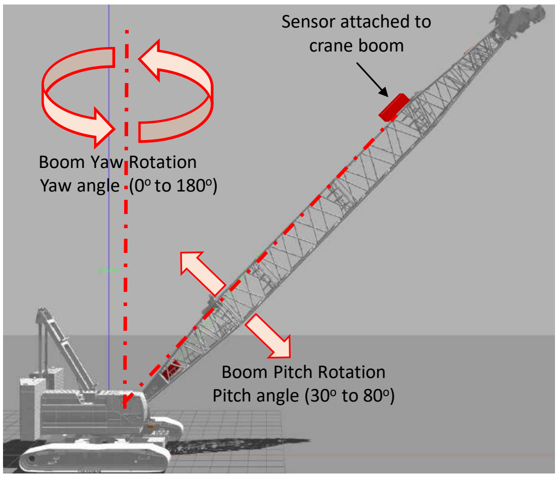


Fig. 4.5 Sensor system attached on boom of crane system and two different types of boom of rotations.

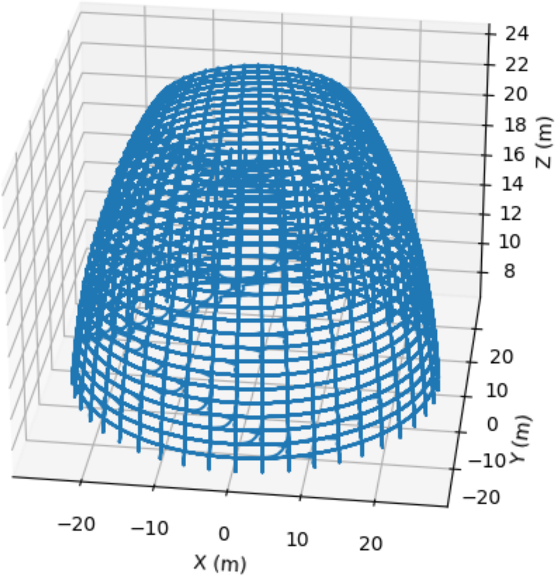


Fig. 4.6 Trajectory followed by sensor system.

crane boom following the trajectory. The trajectory followed by the sensor system attached on boom of crane is shown in Fig. 4.6. The horizontal circular lines in Fig. 4.6 are due to yaw rotation of boom and the vertical lines in trajectory (Fig. 4.6) is due to pitch rotation of boom. The radius of horizontal lines at bottom is higher than the radius of horizontal lines at top.

Algorithm 3 Trajectory of crane boom for data collection.

```
Ensure:
    Define initial value for yaw and pitch angle
    Define start point for yaw and pitch cycle
    Define the rotating speed of boom

    while $duration \leq defined\ time$ do

        $duration \leftarrow change\ in\ time$

        if $yaw\ angle \geq 180$ then
            Change the direction of yaw rotation
        else if $yaw\ angle \leq 0$ then
            Change the direction of yaw rotation
        else if $pitch\ angle \geq 80$ then
            Change the direction of pitch rotation
        else if $pitch\ angle \leq 30$ then
            Change the direction of pitch rotation
        end if

        if $yaw\ cycle = True$ then
            yaw angle = yaw angle + yaw increment
        else if $pitch\ cycle = True$ then
            pitch angle = pitch angle + pitch increment
        end if

        if yaw cycle is completed in both clockwise and anti clockwise direction then
            $yaw\ cycle \leftarrow False$
            $pitch\ cycle \leftarrow True$
            Change the start point of yaw cycle
        else if pitch cycle is completed in both clockwise and anti clockwise direction then
            $pitch\ cycle \leftarrow False$
            $yaw\ cycle \leftarrow True$
            Change the start point of pitch cycle
        end if
    end while
```

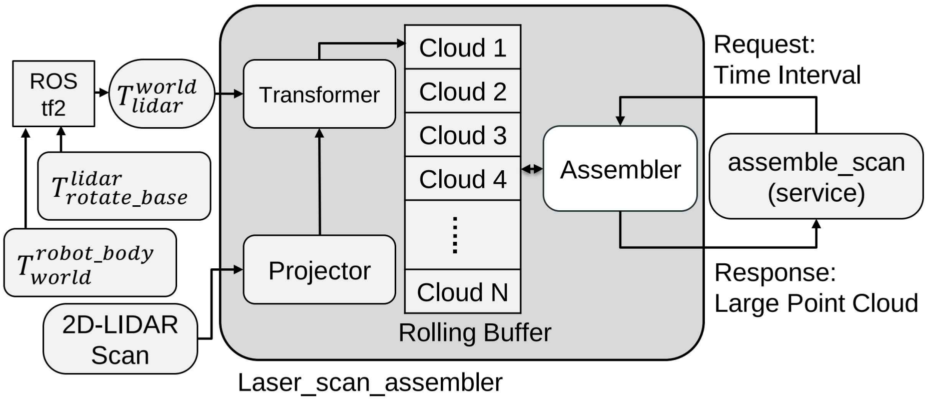


Fig. 4.7 Block diagram of mapping module.

4.4 Real-Time Mapping

We build a 3D map during motion of sensor system using laser-assembler [63], [82]. The block diagram of our mapping method is shown in Fig. 4.7. Firstly the block “projector" in Fig. 4.7 converts the LiDAR scans from polar coordinate to Cartesian coordinate (XYZ) we named it as a LiDAR frame. As the LiDAR coordinate frame is in motion because of LiDAR motion, the next step is to transfer the moving LiDAR coordinate frame to a fixed world frame in order to get a 3D view of the world. This coordinate transformation is an important step to create a 3D map by using a moving 2D-LiDAR. The block “transformer" in Fig. 4.7 transforms the LiDAR measurements from LiDAR frame to fixed world frame as

$$P_{world} = T_{lidar}^{world} P_{lidar} \tag{4.7}$$

After that, the transferred LiDAR measurements are stored in a rolling buffer for a predetermined time. Whenever a request is sent for a 3D point cloud, in response to the request the rolling buffer sends out a large assembled transferred laser scans in PointCloud format.

4.5 Simulation Setup and Results

The proposed method is verified by simulation in the Gazebo simulator. A crane model is designed as shown Fig. 4.5, and the sensor system comprises of 2D-LiDAR, rotating base, and IMU attached to the crane’s boom.

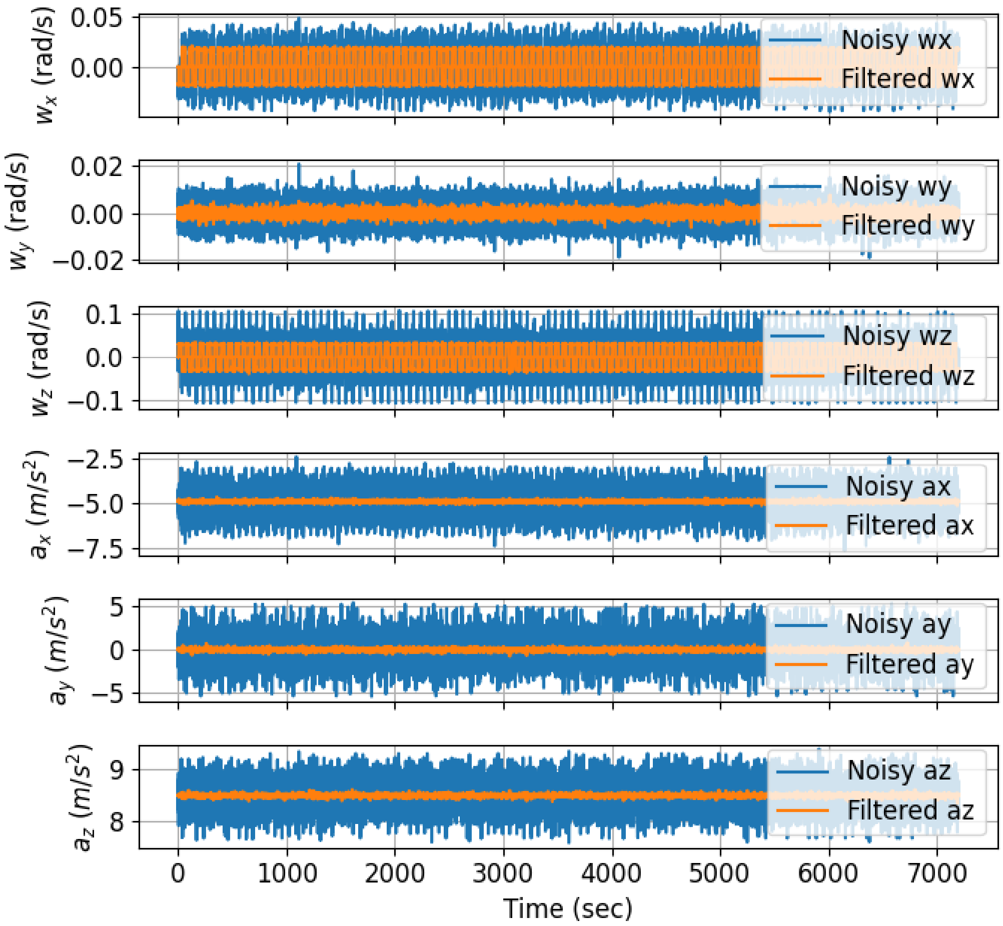


Fig. 4.8 Noisy imu data vs filtered imu data (wx, wy, wz represents the x,y,z axes of gyroscope and ax, ay, az represents the x,y,z axes of accelerometer).

4.5.1 Training the Neural Network

Before training the neural network, we implement the low pass filter on noisy IMU sensor data. The Fig. 4.8 shows filtered IMU data and raw IMU data and Fig. 4.9 visualize the zoomed view of a portion of Fig. 4.8. In Figs. 4.8 and 4.9, the w_x, w_y and w_z are x, y and z axes of IMU's gyroscope and the a_x, a_y and a_z are x, y and z axes of IMU's accelerometer. We can see from these figures that the filter effectively reduces the noise of IMU data. After filtering we train the neural network with filtered IMU data. We train the different models using both windows approaches as described in section 4.3.5 (windows based on repeated IMU measurements (Fig. 4.3) and windows do not use repeated IMU measurements (Fig. 4.4)). To check the effect of the selection of window size of IMU data on training we train different models with three different sliding window sizes of 26, 50, and 200 as described

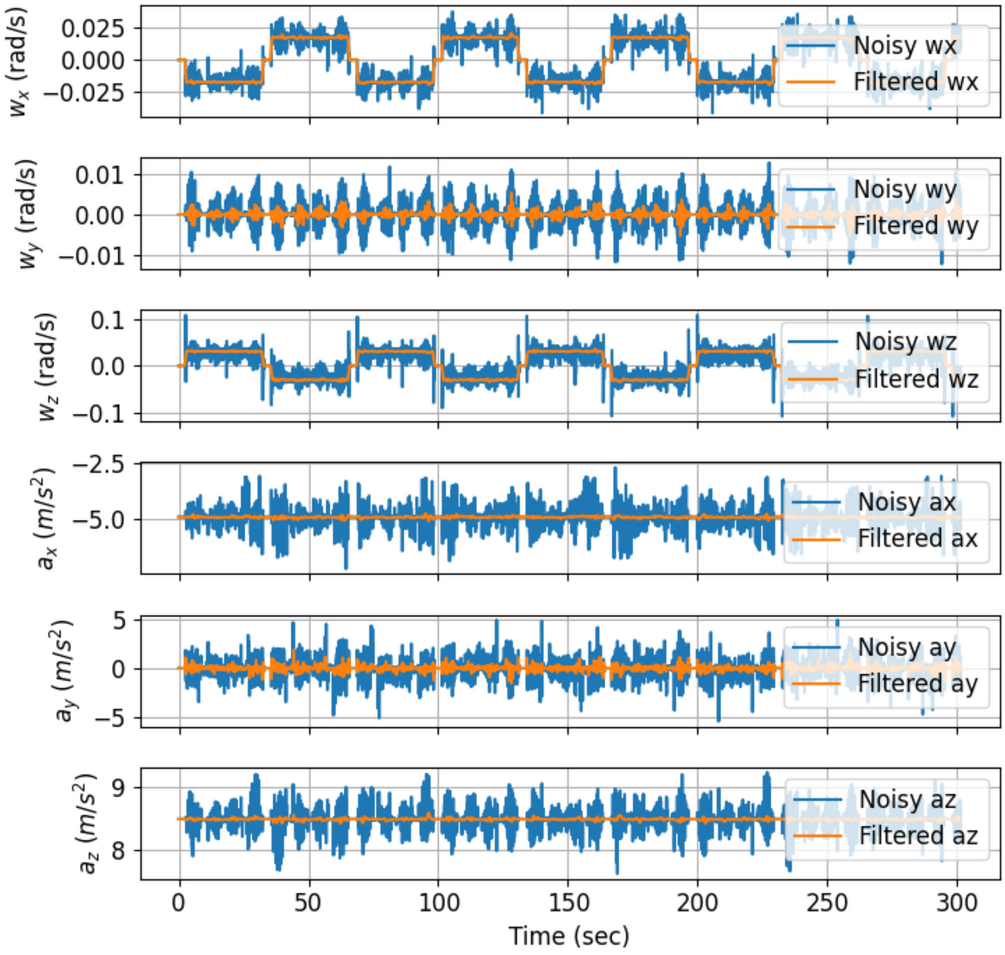


Fig. 4.9 Zoomed view of noisy imu data vs filtered imu data (wx, wy, wz represents the x,y,z axes of gyroscope and ax, ay, az represents the x,y,z axes of accelerometer).

in Table 4.1. The training and validation loss of each training model using windows based on repeated IMU measurements (Fig. 4.3) is shown in Fig. 4.10. The neural network is trained for 500 iterations and 10% of the training data is utilized as validation data. We can observe in Fig. 4.10 that changing the window size has just a little impact on the training and validation loss. The model with the lowest validation loss during training is chosen as the testing model. The best validation loss is found at epoch 200. The training and validation loss of each training model using windows that do not use repeated IMU measurements (Fig. 4.4) are shown in Fig. 4.11. In this case, the neural network is trained for 800 iterations and we can observe how the size of the windows affects the model loss. As we can notice in the Fig. 4.11 the training and validation losses for window size 26 are the minimum, whereas the

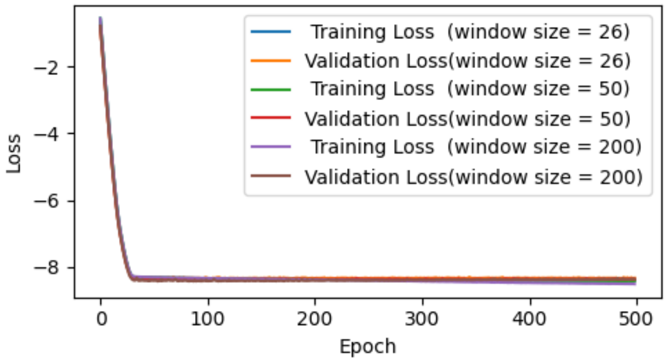


Fig. 4.10 Model loss using windows approach based on repeated IMU measurements (Fig. 4.3) for different window sizes.

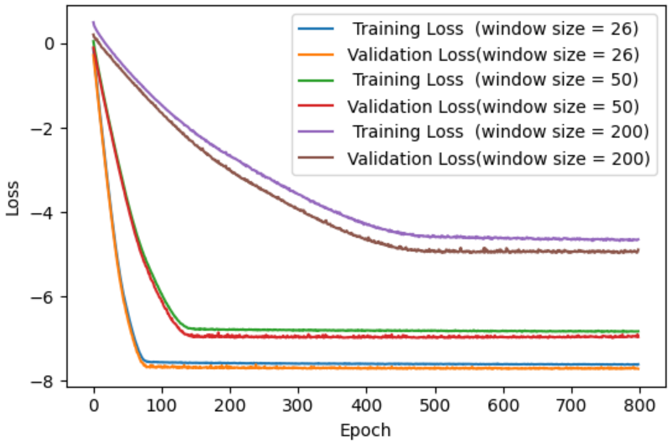


Fig. 4.11 Model loss using windows approach which do not use repeated IMU measurements (Fig. 4.4) for different window sizes.

losses for window size 200 are the highest. In this case, a shorter window makes the training of neural networks easier so losses are less for a shorter window.

4.5.2 Testing the Neural Network

To test the neural network models as given in Table 4.1, we collected data from an accelerometer, gyroscope, and ground truth of pose during the moving boom of the crane in the gazebo. The data is used to test all trained models to predict the change in pose. The difference between the predicted pose and the ground truth, for the windows approach which does not use repeated IMU measurements (Fig. 4.4) are shown in Fig. 4.12. The difference between predicted pose and ground truth for the windows approach based on repeated IMU measurements (Fig. 4.3) are shown in Fig. 4.13. The error for window size 26 is much higher

Table 4.1 Different Parameters of data used for training the different models .

Window size	RMSE for windows based on (Fig. 4.3)	RMSE for windows based on (Fig. 4.4)
26	3.8396	0.9069
26 (Filtered)	3.6278	0.7272
50	1.4250	0.2481
50 (Filtered)	1.2025	0.1374
200	0.6163	0.1322
200 (Filtered)	0.5123	0.1166

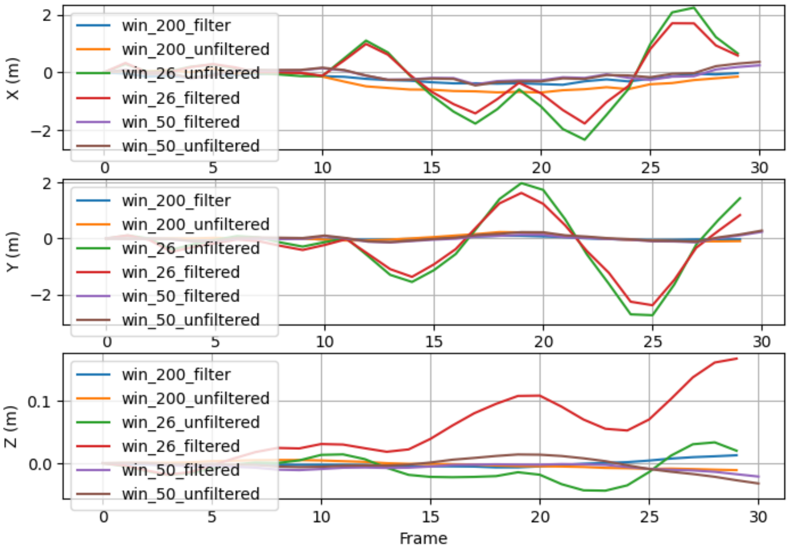


Fig. 4.12 Error of estimated trajectory of different window size using windows approach which do not use repeated IMU measurements (Fig. 4.4).

than the error for window sizes 50 and 200 for both windows approaches (repeated IMU measurements Fig. 4.12 and non-repeated IMU measurements Fig. 4.13), indicating that this size of the window does not contain enough features for training the neural network and producing a good estimate of the pose. When the errors of both windows techniques (Fig. 4.12 and Fig. 4.13) are compared, we can observe that the windows approach that does not employ repeated IMU measurements (Fig. 4.4) has less error than the window strategy based on repeated IMU measurements (Fig. 4.3). In Fig. 4.12 the maximum error is ±2.5m for x and y axes of trajectory while in Fig. 4.13 the maximum error is ±5m for x and y axes of trajectory. Because the window based on repeated IMU measurements uses the same IMU data during training, it has a higher probability of memorizing some patterns. In this case

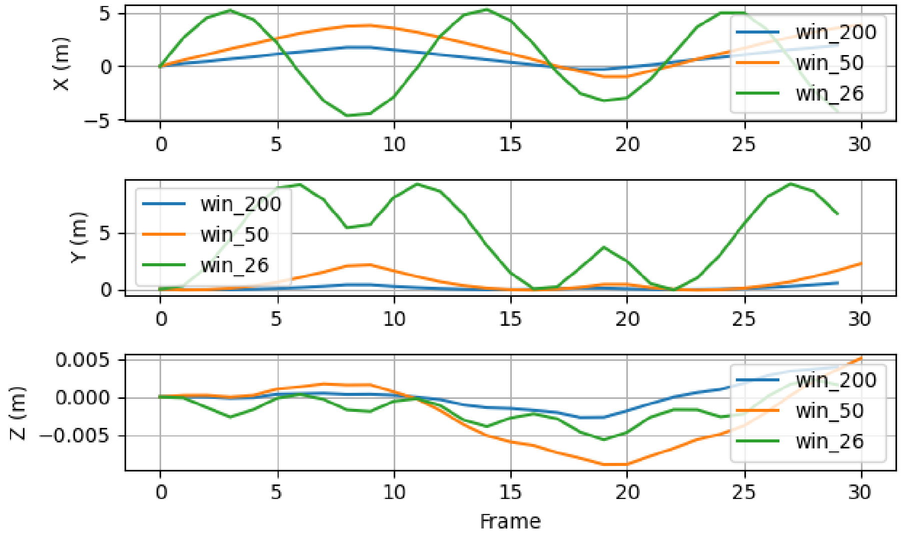
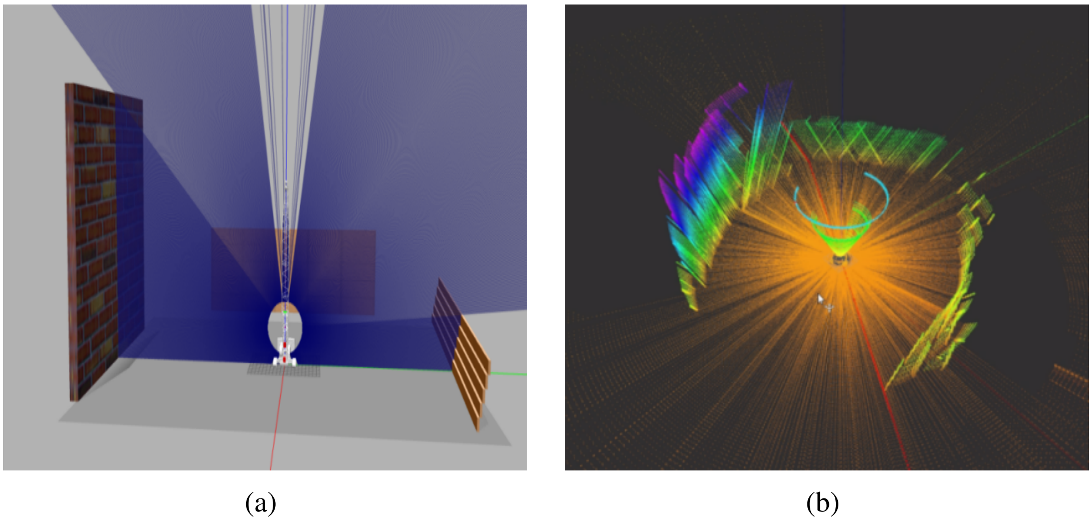


Fig. 4.13 Error of estimated trajectory of different window size using windows approach based on repeated IMU measurements (Fig. 4.3).

the accuracy of prediction is affected when unseen IMU data is used. The drawback of not repeating the IMU measurements-based window approach is that it causes a delay in updating the relative posture since more new IMU readings are required for each new window. The root means square error of the estimated trajectory using different windows approaches and sizes are shown in Table 4.1. As we can see in Table 4.1 the predicted trajectory error for window size 26 is highest because the length of the window is not enough for filtering, and this small window length has also not enough features which can be used for learning. On another side, window size 200 has the lowest trajectory error; that is why we selected window size 200 in our system.

4.5.3 3D Mapping Result

For mapping purpose, 6D pose is estimated using windows approach without repeating IMU measurements (Fig. 4.4). The estimated 6D trajectory is used to register the LiDAR scan to build a 3D point cloud map of an environment. The gazebo environment is shown in Fig. 4.14(a) and point cloud 3D map is shown Fig. 4.14(b). We can see the map is not very accurate because of the error in the estimated pose. These results can be improved if we get more accurate 6D pose estimation using the integration of learning-based inertial odometry and traditional integration-based method and by introducing the close loop during registering the LiDAR scan.



(a) (b)

Fig. 4.14 Mapping Results. (a) Gazebo Environment used for 3D mapping. (b) 3D map of environment.

4.6 Conclusion

This research presents a method for large-scale 3D mapping and neural network-based real-time odometry using an IMU (inertial measurement unit) and a slowly rotating 2-D LiDAR. In this technique, the window of IMU readings is pre-filtered using a low-pass filter before being sent as input to the neural network to estimate the change in position and rotation. A convolutional neural network (CNN) and LSTM make up the neural network (LSTM). To create a large-scale map, the predicted sensor pose is utilized to register the scans of a 2D-rotating LiDAR. The proposed approach is tested in a gazebo environment. The limitation of the proposed method is that 3D mapping is not very precise. Our future work includes firstly the integration of learning-based inertial odometry and traditional integration-based method to get more accurate 6D odometry and secondly minimizing the mapping error by introducing the close loop during registering the LiDAR scan.

Chapter 5

Multi-sensor Fusion-based Real-time Sensor Pose Estimation and 3D Mapping

5.1 Introduction

This chapter describes a method for sensor pose estimation, as well as creating large-scale 3D maps, for construction cranes equipped with a sensor system consisting of a camera, 2D lidar, and IMU. To tackle the challenges posed by the crane boom's complex motion, we utilize an Extended Kalman filter (EKF) to improve the accuracy and reliability of sensor pose estimation. By combining pose estimates from Visual-Inertial Navigation System (VINS) with data from an additional IMU, we estimate the scale value of a monocular camera. This scale value, obtained from the EKF, is then integrated into the VINS algorithm to refine the previously estimated scale value. Slowly rotating 2D lidar is used to build a 3D map. Since there is limited overlap between 2D lidar scans, we leverage the estimated pose to align and construct a comprehensive 3D map. Additionally, we thoroughly evaluate the effectiveness of the latest VINS techniques, as well as the EKF-enhanced VINS approach, in the specific context of crane operations. Through comprehensive performance assessments conducted in both simulated and real environments, we compare the EKF-added VINS method with state-of-the-art VINS techniques. The evaluation results demonstrate that the EKF-added VINS method accurately estimates sensor poses, leading to the generation of high-quality, large-scale 3D point cloud maps for construction cranes.

Generally, to estimate scalem, VINS based on a monocular camera uses the combined data from inertial and visual sensors at initialization stage. If the visual information is insufficient, ambiguous, or noisy, it may have an adverse effect on the scale estimation's

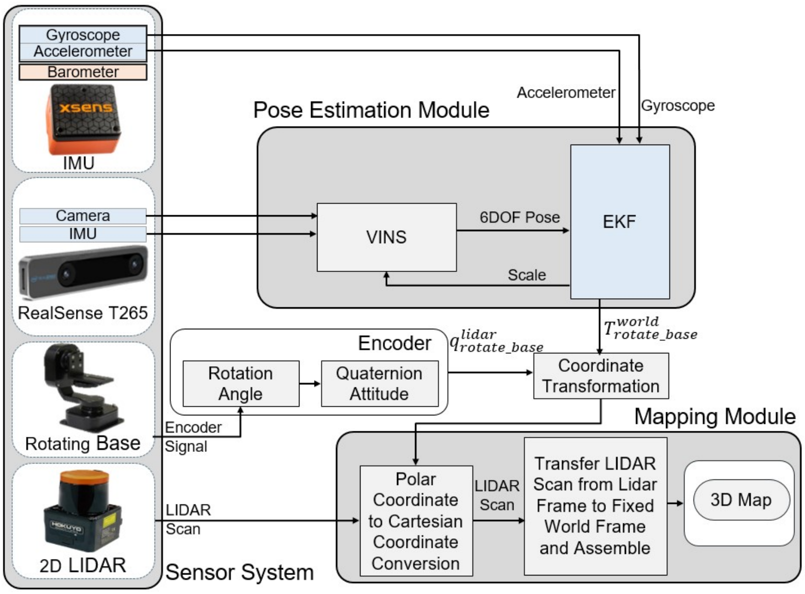


Fig. 5.1 Over all block diagram of proposed method.

accuracy, which will then have a bad impact on the system's overall performance and finally on pose estimation.

In our proposed method, we employed an Extended Kalman filter (EKF) to continuously update the scale value and to enhance the accuracy and reliability of sensor pose estimation in challenging crane boom trajectories. This was achieved by integrating the pose estimates from VINS with data from an additional IMU. The pose used in EKF for fusion is estimated using VINS and we implement the EKF-based approach on four different VINS methods: VINS-MONO [58], VINS-Fusion [60, 59, 57], Multi-state Constraint Kalman Filter (MSCKF) algorithm [50], Robocentric visual-inertial odometry (R-VIO) [33]. Furthermore, we evaluate the effectiveness of these four cutting-edge VINS techniques, as well as EKF added VINS techniques, in the specific context of a crane system. We assess the performance, suitability, and effectiveness of these methods, focusing specifically on their application in crane operations.

The scale value obtained from the EKF is then incorporated into the VINS algorithm to update the previously estimated scale value. This approach effectively addresses one of the limitations of VINS, which previously relied on an initial scale parameter estimated during the initialization step for a monocular camera. By integrating the EKF for continuous scale estimation, the VINS algorithm becomes more robust and accurate in its scale estimation process.

The main contributions of this study are as follows:

- EKF is used to continuously update the scale value, thereby improving the accuracy and reliability of sensor pose estimation in complex crane boom trajectories.
- Evaluate the effectiveness, suitability, and performance of state-of-the-art VINS and EKF-added VINS, with a specific focus on their applicability in crane operations.
- The proposed approach generates a more accurate 3D map for a crane by utilizing a rotating 2D-Lidar mounted on the crane boom, and the pose estimation obtained from the VINS and EKF added VINS techniques. The estimated pose is utilized to register the 2D lidar scanlines, enabling the construction of an accurate and comprehensive 3D map.

5.2 Overview of proposed Method

The proposed method is based on the integration of VINS based poses and additional IMU to estimate more accurate and robust pose. The estimated pose is used to create a large-scale crane map. As shown in block diagram of proposed method Fig. 5.1, the proposed method consists on two main modules: Pose estimation module, and mapping modul. In pose estimation module, 6 Dof pose is estimated using EKF which is used to fuse the pose estimated by VINS [58] and measurements from acceleromter, gyroscope and barometer of IMU. The mapping module receives lidar poses and 2D lidar scan and transforms 2D lidar measurements to world frame to construct a 3D point cloud map. In following sections we will explain each module in detail.

5.3 Overview of VINS

In this section, we provide a concise overview of VINS algorithms that were implemented on a crane to estimate the trajectory of the sensor. For a more detailed understanding, we recommend referring to the original papers. In this chapter, these algorithms were specifically evaluated for their application on a crane. The objective of this comparison is to assess the appropriateness of various VIO algorithms for sensor pose estimation and 3D map building in crane operations.

VINS-MONO

VINS-MONO [58], is a versatile monocular visual-inertial state estimator. It utilizes a robust initialization procedure and a nonlinear optimization-based approach that combines IMU measurements and feature observations. This results in accurate visual-inertial odometry. The integration of a loop detection module enables efficient relocalization, and a 4-DOF pose graph optimization ensures global consistency. Overall, VINS-MONO offers a reliable and adaptable solution for precise localization applications.

VINS-Fusion

VINS-Fusion [60, 59, 57], an optimization-based multi-sensor state estimator, demonstrates precise self-localization capabilities for various autonomous applications. Serving as an extension of VINS-Mono, VINS-Fusion supports a range of visual-inertial sensor combinations, including mono camera with IMU, stereo cameras with IMU, and even stereo cameras alone. Its key features encompass online spatial calibration (transformation between the camera and IMU), as well as online temporal calibration, which accounts for the time offset between the camera and IMU.

MSCKF

The MSCKF algorithm [50], originally developed as the Multi-state Constraint Kalman Filter, introduces a measurement model that captures the geometric constraints among camera poses observing a specific image feature. Unlike traditional approaches that require estimating the 3D feature position, the MSCKF eliminates this need by directly expressing the constraints. The extended Kalman filter backend incorporates this formulation of the MSCKF specifically for event-based camera inputs but has been modified to handle feature tracks from standard cameras as well.

R-VIO

R-VIO [33] is a lightweight and efficient visual-inertial navigation algorithm designed for 3D motion tracking by utilizing only a monocular camera and IMU. Unlike traditional world-centric algorithms that estimate absolute motion with respect to a fixed global frame, R-VIO focuses on estimating relative motion with higher accuracy with respect to a local frame. The algorithm then incrementally updates the global pose through a composition step, resulting in improved performance and precision.

To perform a thorough evaluation, we assessed the algorithms in various modes supported by VINS. This included analyzing their performance with monocular + IMU, stereo without IMU, and stereo + IMU configurations. By incorporating data from different sensors, we gained valuable insights into the impact of sensor fusion on the performance of VINS algorithms.

5.4 Error-state Extended Kalman filter (ES-EKF)

The EKF formulation and algorithm are well known for integrating diverse sensors in order to estimate the pose of the sensor [49, 81, 46, 47, 7]. Here, we focus on conveying important implementation details. Our objective is to accurately estimate the scale value for a monocular camera, the complete 3D pose (including all six degrees of freedom), and the velocity of a sensor system attached to a crane boom during crane operation.

Fig. 5.2 illustrates the configuration of the sensors setup along with its associated coordinate frames. The inertial sensor measures acceleration and rotational velocity along three axes in IMU body frame. On the other hand, VINS supplies the 3D position and attitude whih are referenced to a visual frame established at the initialization. The Error-state EKF is used to fuse inertial sensors measurements and pose estimated by VINS. This fusion process enables the determination of the scale value for monocular cameras and improves the accuracy and robustness of the pose estimation.

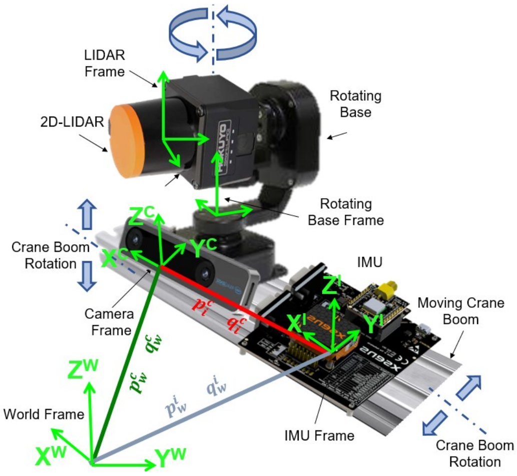


Fig. 5.2 The sensor setup and coordinate frame attaced to each sensor is shown. Transformation between frames is represented by a rotation q and a translation p. The transformation between IMU and camera frame have fixed values, which is highlighted in red.

5.4.1 Modeling Inertial Sensor

An inertial sensor commonly consists on accelerometer, gyroscope. Gyroscope measures the angular velocity $\widetilde{w}$ at each time instance t. However, its measurements are affected by a slowly changing bias b_w and noise n_w over time. As a result, the model representing the gyroscope measurements is formulated as follows:

$$w_t = \widetilde{w}_t - b_{w_t} - n_{w_t} \tag{5.1}$$

At time instance t, the accelerometer measures the specific force $\widetilde{a}_t$. However, its measurements are influenced by both bias b_a and noise n_a as given below:

$$a_t = \widetilde{a}_t - b_{a_t} - n_{a_t} \tag{5.2}$$

It is common assumption that the acceleration and gyroscope measurements noise follows a Gaussian distribution. The biases in acceleration and gyroscope are treated as random walk processes, where the derivatives of these biases are assumed to follow a Gaussian distribution as [41, 58]:

$$\dot{b}_{w_t} = n_{b_w} \qquad \dot{b}_{a_t} = n_{b_a} \tag{5.3}$$

The error state EKF offers several advantages over the vanilla EKF. Firstly, it exhibits superior performance due to the error state's closer approximation to linearity during evolution. Secondly, the error state formulation simplifies the handling of special quantities like 3D rotations, facilitating their integration within the EKF framework.The error state formulation in the Extended Kalman Filter (EKF) approach involves separating the state into a larger nominal state and a smaller error state. Next, we will discuss both of these briefly.

5.4.2 Nominal State

The nominal state represents the predicted states based on the motion model using IMU measurements. The nominal state vector is composed of the following elements:

$$x_{25\times1} = [p^i_w \quad v^i_w \quad q^i_w \quad b_w \quad b_a \quad \lambda \quad q^c_i]^T \tag{5.4}$$

where
$p^i_w = [p_x, p_y, p_z]^T$ is position along x, y, and z axes
$v^i_w = [v_x, v_y, v_z]^T$ is velocity along x, y, and z axes
$q^i_w = [q_w, q_x, q_y, q_z]^T$ is orientation in quaternion form along x, y, and z axes
$b_w = [b_{w_x}, b_{w_y}, b_{w_z}]^T$ is bias along x, y, and z axes of gyroscope
$b_a = [b_{a_x}, b_{a_y}, b_{a_z}]^T$ is bias along x, y, and z axes of accelerometer
$q^c_i = [q^c_{i_w}, q^c_{i_x}, q^c_{i_y}, q^c_{i_z}]^T$ is the rotation between the IMU to the camera frame
$p^c_i = [p^c_{i_x}, p^c_{i_y}, p^c_{i_z}]^T$ is the distance from the IMU to camera frame
λ is scale of monocular camera

The state is governed by the following set of differential equations based on continuous motion model using IMU measurements:

$$
\begin{aligned}
\dot{p}_i^w &= v_w^i \\
\dot{v}_i^w &= C_{q_w^i}^T(\widetilde{a} - b_a - n_a) - g \\
\dot{q}_w^i &= \frac{1}{2} q_w^i \otimes (\widetilde{\omega} - b_\omega - n_\omega)
\end{aligned}
\tag{5.5}
$$

$$
\dot{b}_\omega = n_{b_\omega} \quad \dot{b}_a = n_{b_a} \quad \dot{\lambda} = 0 \quad \dot{p}_c^i = 0 \quad \dot{q}_i^c = 0
$$

here g is the gravity vector in the world frame and $\otimes$ is a quaternion product operator. We make the assumption that the scale factor drifts at a very slow rate, hence $\dot{\lambda} = 0$. Since the IMU provides discrete measurements, Eq. 5.5 must be discretized by considering the sampling time interval Δt. As a result, the discrete-time motion model can be expressed through the following equations. For the simplicity, the equations presented below do not utilize subscripts or superscripts for coordinate frame notations.

$$
\begin{aligned}
p_k &= p_{k-1} + v_{k-1}.\Delta t + (C_{q_{k-1}}^T.a_k - g).\Delta t^2/2, \\
v_k &= v_{k-1} + (C_{q_{k-1}}^T.a_k - g).\Delta t, \\
q_k &= q_{k-1} \otimes q(\omega_k.\Delta t)
\end{aligned}
\tag{5.6}
$$

here, k and $k-1$ represent the indices for the current and previous time stamp.

5.4.3 Error State

The error state captures the accumulated modeling errors and process noise. We estimate this small error in the error state EKF and use it as a correction to the nominal state [7]. The error state vector is stated as

$$
\delta x = [\delta p_w^i \quad \delta v_w^i \quad \delta \theta_w^i \quad \delta b_w \quad \delta b_a \quad \delta \lambda \quad \delta p_i^c \quad \delta \theta_i^c]^T \tag{5.7}
$$

The error state kinematics model equation can be represented as follows:

$$
\begin{aligned}
\dot{\delta x} &= F \delta x + G n \\
P_{k+1} &= F P_k F^T + Q
\end{aligned}
\tag{5.8}
$$

where F is kinematic model that propagates the errors over time. n is noise vector and can be expressed as $n = \left[n_a^T, n_{ba}^T, n_\omega^T, n_{b\omega}^T \right]$. Q is system or process noise covariance matrix and can be represented as a $Q = \mathrm{diag}(\sigma_{n_a}^2, \sigma_{n_{ba}}^2, \sigma_{n_\omega}^2, \sigma_{n_{b\omega}}^2)$. P is state covariance matrix. The detailed explanation and derivation of F, G and Q can be found in [81, 27]. For F, G and Q, we use same approach as given in [81].

5.4.4 Measurement Model

The measurement model for the camera pose measurement, obtained from the VINS can be expressed as follows [46, 81]

$$z = \begin{bmatrix} p_w^c \\ q_w^c \end{bmatrix} = \begin{bmatrix} (p_w^i + C_{(q_w^i)}^T p_i^c)\lambda + n_p \\ q_i^c \otimes q_w^i \end{bmatrix} \tag{5.9}$$

The equation can be linearized as $z = H\delta x + n$ as given in [46, 81], where H represents the Jacobian matrix of the VINS pose measurement with respect to the error state. we update and correct our estimates using Extended Kalman Filter based procedure as:
Compute the residual

$$\delta z = z - \hat{z} \tag{5.10}$$

Estimate the Kalman gain

$$K = PH^T(HPH^T + R)^{-1} \tag{5.11}$$

Calculate the correction

$$\widehat{\delta x} = K\delta z \tag{5.12}$$

Update the state covariance

$$P = (I - KH)P(I - KH)^T + KRK^T \tag{5.13}$$

5.5 Mapping Module

To construct a dense 3D point cloud map, we have made some modifications to our previous approach proposed in [77]. The previous approach involved building the 3D map using structural information of the crane and rotation estimates from an IMU. However, in this modified approach, we utilize the pose estimates provided by VINS+EKF for building the 3D map. Our approach involves utilizing a 2D lidar sensor that is mounted on a rotating base, which, in turn, is attached to a crane boom. This configuration allows us to capture comprehensive spatial information and generate a detailed representation of large environment both horizontally and vertically.

During crane operations, the lidar faces motions which arise from two sources: the rotation of the rotating-base and the motion of the crane boom (see Fig. 5.2). Since the lidar is continuously in motion, the successive 2D lidar scan lines do not overlap with one

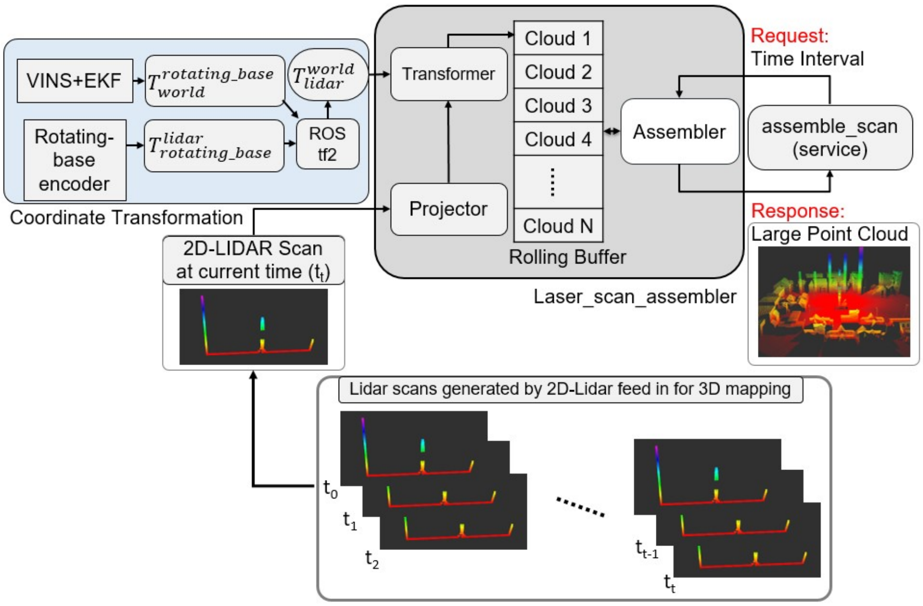


Fig. 5.3 Block diagram of 3D mapping method using 2D lidar by laser assembler.

another. Consequently, in order to register the lidar scans and construct a comprehensive 3D map, it becomes essential to accurately track the lidar pose. By continuously monitoring the lidar's position and orientation in space, we can align and integrate the individual scans into a coherent 3D representation. We track the lidar pose as: The rotational angle of the rotating-base is measured using its encoder. This transformation, denoted as $T_{rotating\_base}^{lidar}$, represents the lidar frame's rotation relative to the rotating base frame. The motion of the crane boom frame relative to the fixed world frame ($T_{world}^{rotating\_base}$) is measured using VINS+EKF. To calculate the transformation from the lidar frame to the fixed world frame (T_{world}^{lidar}), we establish a chain of transformations between the respective coordinate frames, as shown in the following equation.

$$T_{world}^{lidar} = T_{rotating\_base}^{lidar} T_{world}^{rotating\_base} \tag{5.14}$$

The tf2 broadcaster [21], a package of the Robot Operating System (ROS) is used in broadcasting the transformations of all coordinate systems. Whenever an update occurs regarding a specific transform of any frame, coordinate transformation messages are broadcasted by the tf2 broadcaster. This mechanism enables us to keep track of the motion of the lidar frame as it moves.

Once transformation of lidar frame to fixed world T_{world}^{lidar} is obtained, it is used in the laser-assembler [63], [82] to construct a 3D map during the lidar's motion, which combines individual laser scan lines obtained from a 2D lidar and creates a composite 3D point cloud. The mapping process is shown in Fig. 5.3 using a block diagram. For 3D mapping, the *projector* block converts the polar coordinate lidar scans measurements into Cartesian coordinates (XYZ), which we refer to as the lidar frame. Since the lidar frame is subject to motion, our next step involves transforming the moving lidar frame into a fixed world frame, enabling us to obtain a three-dimensional representation of the environment. This coordinate transformation is shown by *transformer* block, which by using transformation information (translation and rotation) of lidar frame obtained from VINS+EKF and rotating base converts the lidar measurements from the lidar frame to the fixed world frame. Subsequently, the transformed lidar measurements are stored in a rolling buffer for a predetermined duration. Whenever a request for a 3D point cloud is received, the rolling buffer retrieves and delivers large assembled transferred laser scans in the Point Cloud format.

5.6 Implementation of VINS

For implementation of VINS we needs IMU parameters such as noise and random walk, camera intrinsic parameters and Extrinsic parameter between IMU and Camera. To calibrate the IMU and estimate the noise and random walk, the ROS package tool imu_utils and allan_variance_ros [24, 8] was utilized. Data collection was performed over a duration of two hours while the IMU was kept stationary. VINS requires camera calibration parameters such as image width and height, camera distortion model, Intrinsic camera matrix and projection matrix which consists on focal lengths and principal point. VINS supports the pinhole model and the MEI model. A OpenCV camera calibration package based ros tool [51, 64] is used to provide these camera calibration parameters to VINS. Kalibr calibration toolbox [23] [22] is used for imu-camera joint calibration to estimate the Spatial and temporal calibration paramters between IMU and Camera. To achieve precise camera calibration, we used an 8x6 checkerboard with 108mm squares and moved the checkerboard within the camera frame to different positions: left, right, top, and bottom of the field of view. Additionally, adjust the position of the checkerboard by moving it towards or away from the camera while tilting it. The parameters of each package were manually adjusted, starting from their default values and fine-tuning them for improved performance. Any recommendations provided by the authors were taken into consideration during this parameter adjustment process.

5.7 Simulation Results

We first conducted an evaluation of proposed mapping method using VINS and VINS+EKF in a simulated environment using the Gazebo simulator [25] and ROS environments. In

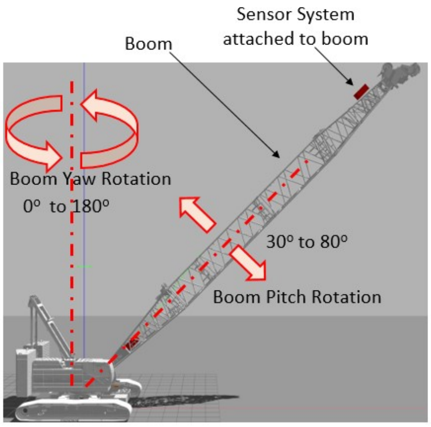


Fig. 5.4 The Gazebo simulation environment was used to create a crane model. The crane's boom has two different types of rotations.

the Gazebo simulation, we designed a robotic model of a crane as shown in Fig. 5.4. The crane's boom has two rotational axes: one for vertical movement and the other for horizontal movement. To control these rotations, we utilized ROS joint trajectory control. The sensor system, consisting of a monocular camera, 2D lidar, a rotating base, and two IMU, was attached to the crane's boom. We performed evaluations of our proposed method in a simulated environment that represents an open-sky complex construction site area as shown in Fig. 5.5(a). Fig. 5.5(b) shows the 3D map built by the proposed mapping method using VINS-MONO+EKF. We can see the proposed method created accurate and precise 3D mapping while the crane boom is moving in different directions.

In order to assess the accuracy of VINS and VINS+EKF, we compared the 3D map generated by using VINS and VINS+EKF with the ground truth obtained from a simulation model. The point-to-point distances between the two point cloud maps were calculated using cloud compare [15]. Figure 5.6 presents the results of this comparison for the 3D maps created using VINS-MONO and VINS-MONO+EKF, visualized with a color scale map. Blue indicates smaller distances, while red represents larger distances.

In Fig. 5.6(a) and Fig. 5.6(c), we can observe the maps generated by VINS-MONO and VINS-MONO+EKF, respectively. The VINS-MONO map has a higher number of points represented in green, while the VINS-MONO+EKF map predominantly contains points in blue. This discrepancy indicates that the VINS-MONO map has more errors compared to the VINS-MONO+EKF map. Fig. 5.6(b) and Fig. 5.6(d) display the distribution fitting graphs of the point-to-point distances for the VINS-MONO and VINS-MONO+EKF maps, respectively. Approximately 91.6% of the points in the VINS-MONO map and VINS-MONO+EKF map have distances below 0.199 m and 0.357 m, respectively. These findings demonstrate the effectiveness of the VINS-MONO+EKF approach in reducing errors. Moreover, the point cloud map generated using the VINS-MONO+EKF method closely resembles the ground truth.

5.8 Experimental Results Using Model Crane

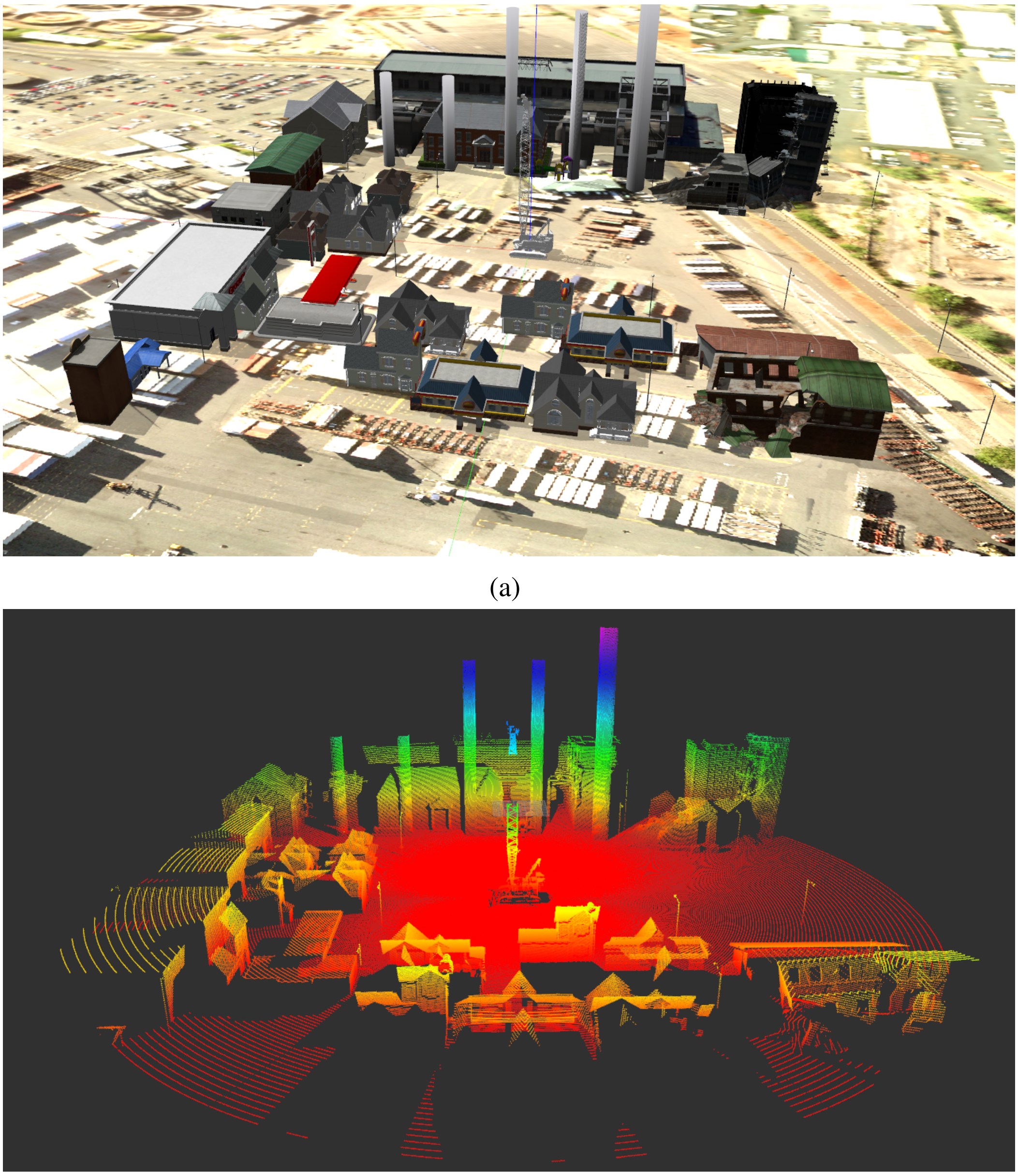
The effectiveness of our proposed method was further evaluated through real-world experiments conducted on a crane model, as illustrated in Figure 5.7. The crane model was situated in an indoor environment spanning a 20 m by 10 m area and motion capture system is installed in environment as shown in Fig. 5.8 to get ground truth of sensor's trajectory. To conduct the experiments, we utilized a sensor system comprising a realsense T265, Hokuyo UST-20LX 2D lidar, an Orion Giken RHST-PA1L rotating base, and an XSENS MTI-630 IMU mounted on the crane's boom. This sensor setup enabled us to capture extensive data for mapping the crane environment.

5.8.1 Trajectory evaluation

In order to analyze the accuracy and robustness of different VINS and VINS+EKF approaches, we conducted a series of experiments under different scenarios. These scenarios included varying crane boom rotation speeds as well as challenging crane boom trajectories. The objective was to assess the performance of the different approaches in these diverse scenarios and gain insights into their effectiveness.

To do quantitative evaluation we compared the estimated trajectory with ground truth obtained from a motion capture system. We utilized the sim3 trajectory alignment method described in [91, 26] to align the estimated trajectory with the ground truth. We then calculated the Root Mean Square Error (RMSE) and Relative Pose Error (RPE) using [91, 26] to quantify the position and orientation errors of the estimated trajectory over the aligned trajectory. Due to space limitations, we present three crane boom trajectories under three distinct scenarios. For each trajectory, three graphs are provided. The first graph illustrates the estimated trajectories of the different VINS and VINS+EKF methods in comparison to the ground truth. The second graph displays the RMSE at each timestep of the trajectory, offering insights into periods where estimation performance may be compromised. The third graph presents the RPE, which is computed for segments of the dataset and enables an examination of how localization solutions drift as the trajectory lengthens.

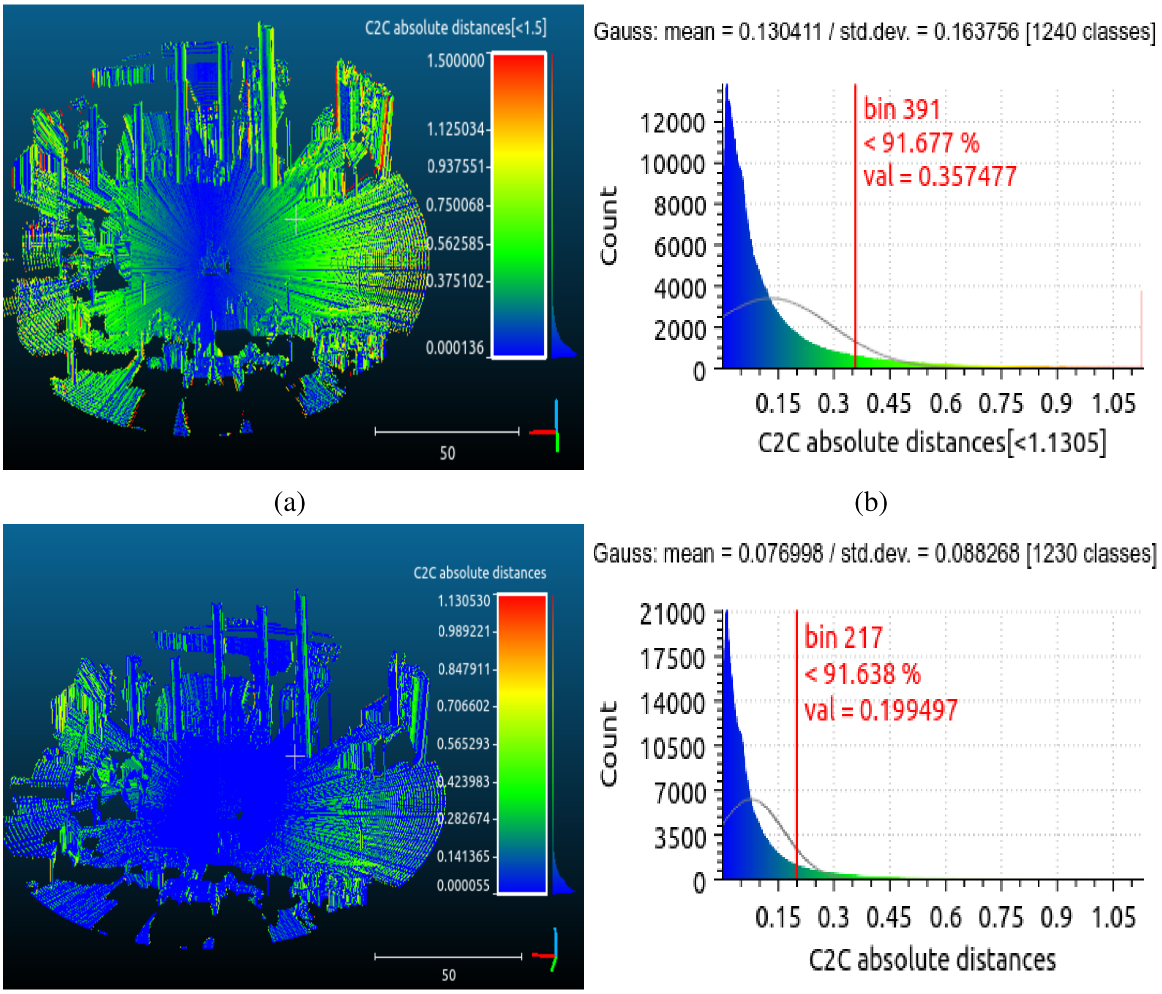
In the first case, the crane boom moves in an arbitrary motion. Fig. 5.9(a) shows the trajectory estimated using VINS and VINS+EKF methods. Fig. 5.9(b) and Fig. 5.9(c)



(a)

(b)

Fig. 5.5 3D map implemented in Gazebo simulation environment (a) Complex construction environment. (b) 3D point cloud map for complex construction environment. The varying colors in the map indicate the elevation or height of each individual point.



(a) (b)

(c) (d)

Fig. 5.6 Analysis of a 3D map in a simulation environment. The Point-to-Point Distances for the map generated by VINS-MONO are shown in (a), and the map generated by VINS-MONO+EKF is shown in (c). Color represents an error (cloud-to-cloud distance between ground truth and point cloud). The distribution fitting of point-to-point distances for the map generated by VINS-MONO is shown in (b) and the map generated by VINS-MONO+EKF is shown in (d).

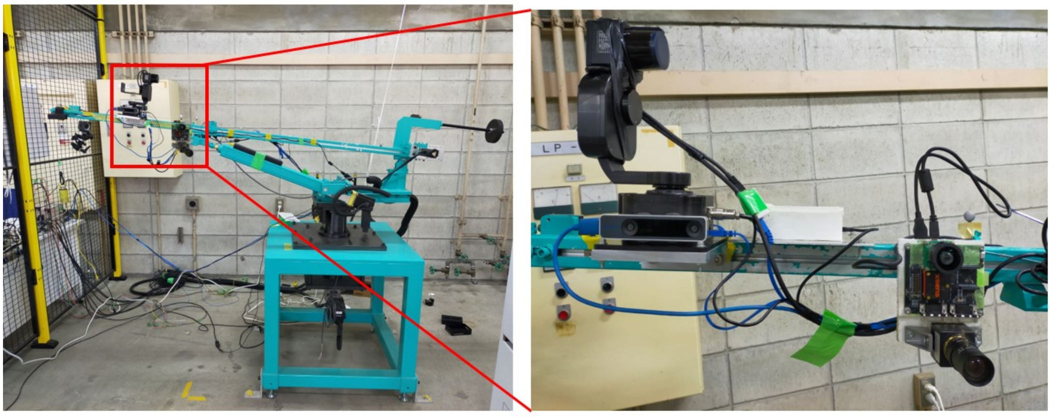


Fig. 5.7 The experiment showcases the crane model along with its accompanying sensor system. The sensor system is connected to the boom of the crane.

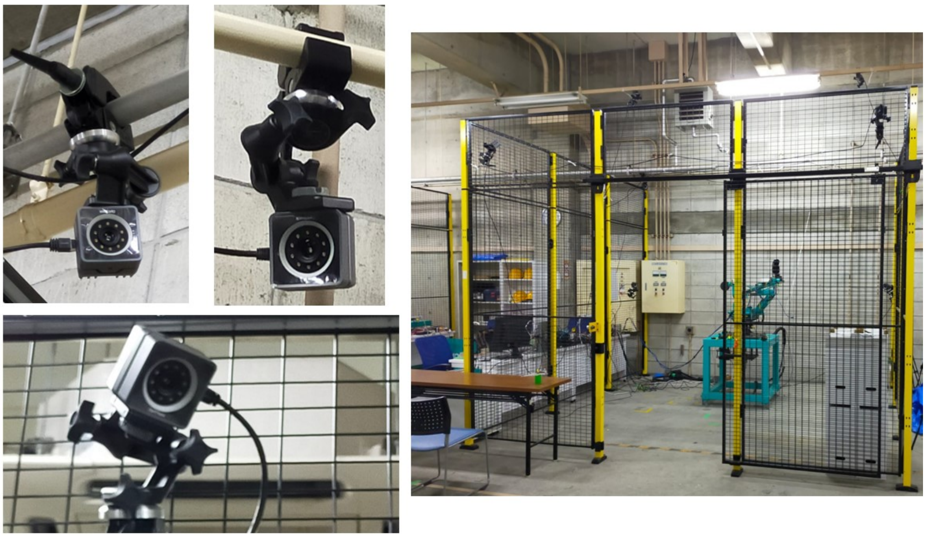
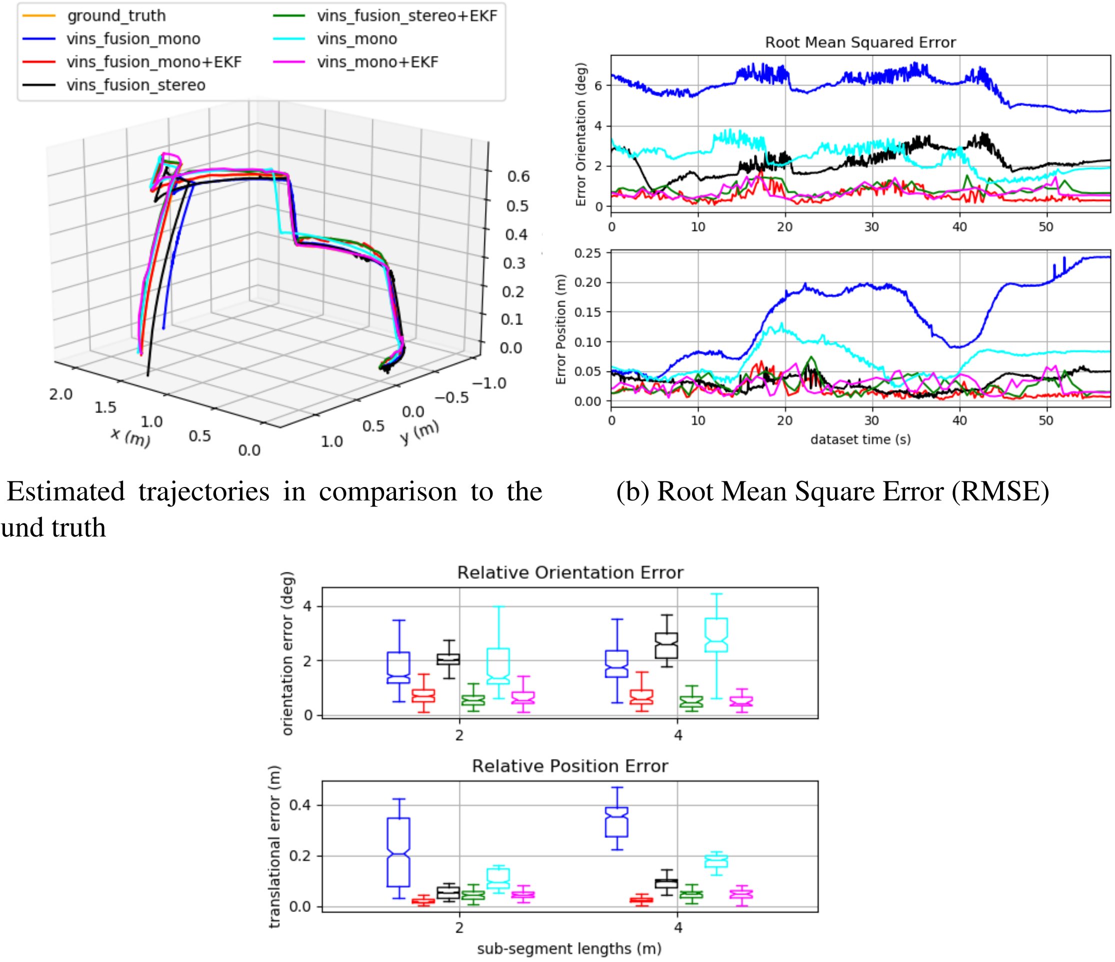


Fig. 5.8 Motion capture system attached to crane environment.

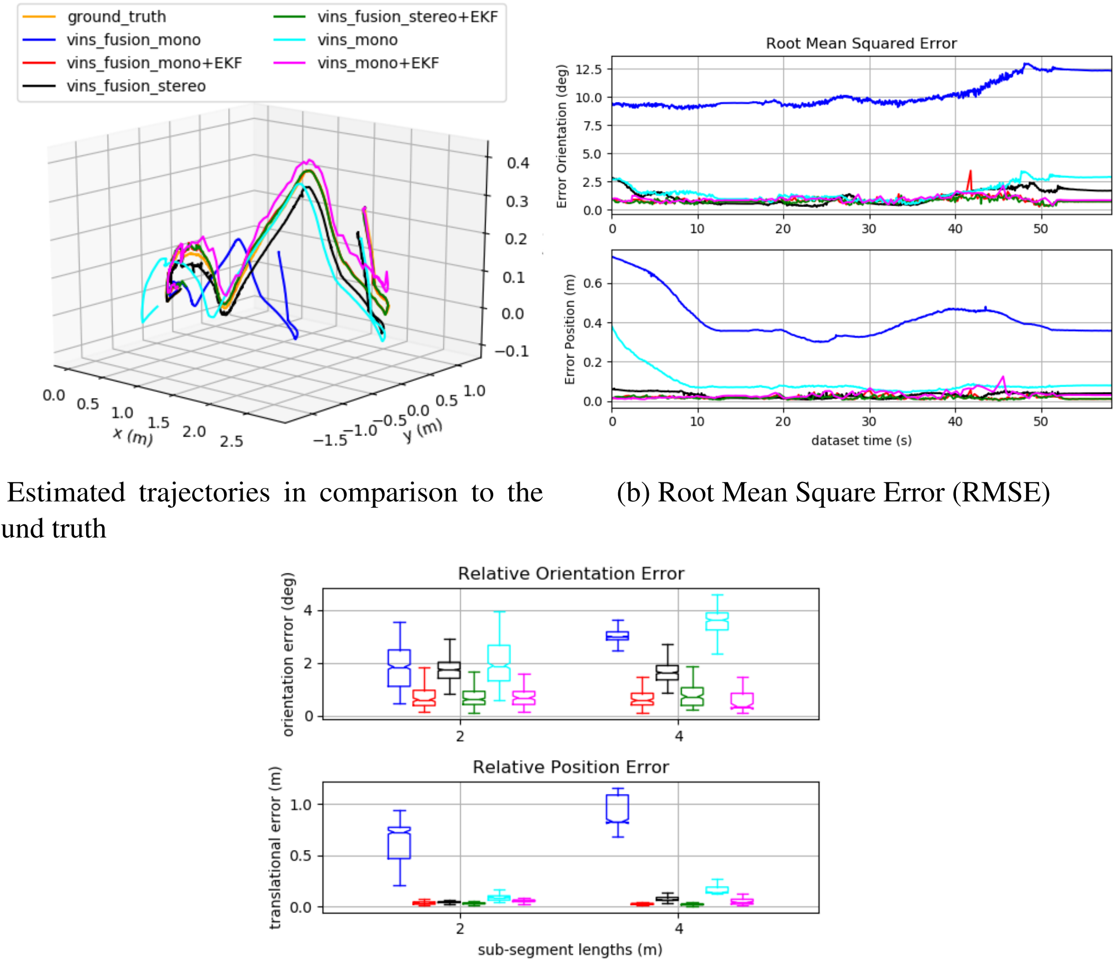


(a) Estimated trajectories in comparison to the ground truth

(b) Root Mean Square Error (RMSE)

(c) A boxplot summarizes error statistics with trajectory segments of lengths 2 and 4 m

Fig. 5.9 Trajectory evaluation for case 1 (crane boom moves in arbitrary motion).

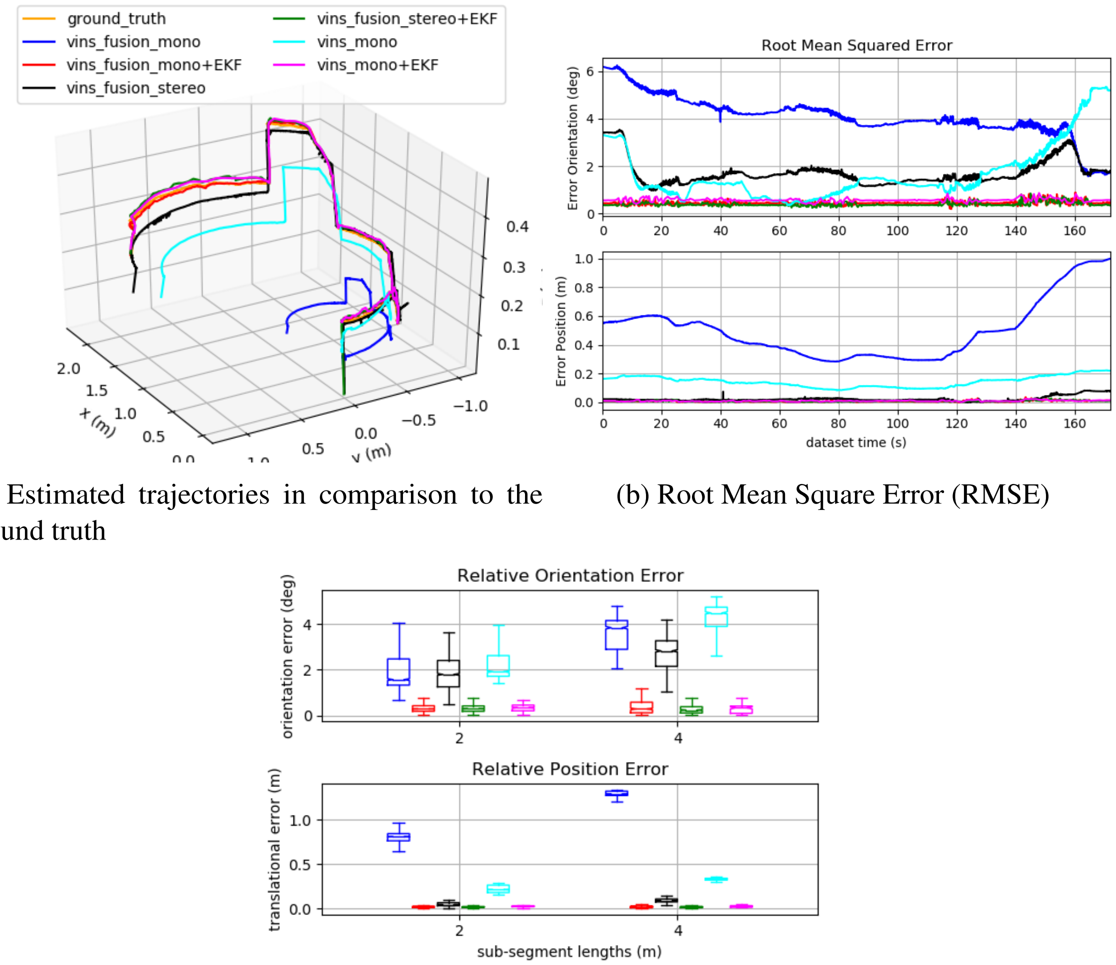


(a) Estimated trajectories in comparison to the ground truth

(b) Root Mean Square Error (RMSE)

(c) A boxplot summarizes error statistics with trajectory segments of lengths 2 and 4 m

Fig. 5.10 Trajectory evaluation for case 2. Crane boom started moving with a jerk.



(a) Estimated trajectories in comparison to the ground truth

(b) Root Mean Square Error (RMSE)

(c) A boxplot summarizes error statistics with trajectory segments of lengths 2 and 4 m

Fig. 5.11 Trajectory evaluation for case 3. Crane boom is subjected to large rotational changes.

Table 5.1 The results Absolute Pose Error (APE) is shown.

	trj_01 (deg/m)	**trj_02(deg/m)**	**trj_03(deg/m)**	**trj_04(deg/m)**	**trj_05(deg/m)**
vins_fusion_mono	6.326 / 0.432	5.847 / 0.161	8.940 / 0.828	2.091 / 0.150	0.181 / 0.789
vins_fusion_mono+EKF	0.855 / 0.019	0.556 / 0.018	3.646 / 0.818	1.910 / 0.110	0.701 / 0.610
vins_fusion_stereo	1.306 / 0.033	2.221 / 0.034	1.732 / 0.074	1.795 / 0.030	1.600 / 0.030
vins_fusion_stereo+EKF	1.194 / 0.010	2.012 / 0.023	0.767 / 0.009	0.393 / 0.009	0.408 / 0.024
vins_mono	1.762 / 0.102	2.421 / 0.072	1.816 / 0.143	2.091 / 0.150	2.147 / 0.095
vins_mono+EKF	0.957 / 0.034	0.742 / 0.030	0.571 / 0.026	0.578 / 0.011	0.492 / 0.033

present the RMSE and RPE, respectively. We observed that VINS-Fusion-Stereo+EKF had the lowest RMSE and RPE, while VINS-Fusion-Mono had the highest.

In the second scenario, the crane's abrupt movement (with a jerk) was studied. Fig. 5.11(a), Fig. 5.11(b) and Fig. 5.11(c) display the trajectory, RMSE, and RPE, respectively. We can observe that the trajectories of VINS-MONO and VINS-FUSION-MONO are significantly affected by jerk, resulting in higher errors. However, incorporating an additional IMU using EKF mitigated the impact of a jerk on the trajectory.

In the third case, the crane boom undergoes significant rotational changes. Fig. 5.10(a), Fig. 5.10(b), and Fig. 5.10(c) display the trajectory, RMSE, and RPE, respectively. VINS-FUSION-MONO and VINS-MONO faced inaccurate scaling, leading to larger pose errors. The EKF approach accurately addresses the scaling issue in both methods.

Based on the results of the Absolute Pose Error (APE) in Table 5.1, the VINS-MONO+EKF approach demonstrated precise and consistent performance, making it a robust choice for crane state estimation.

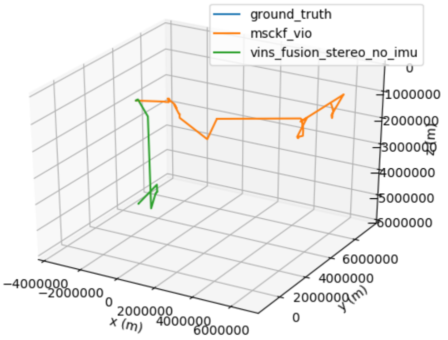


Fig. 5.12 Trajectory evaluation using MSCKF and Vins-fusion-stereo-no-imu.

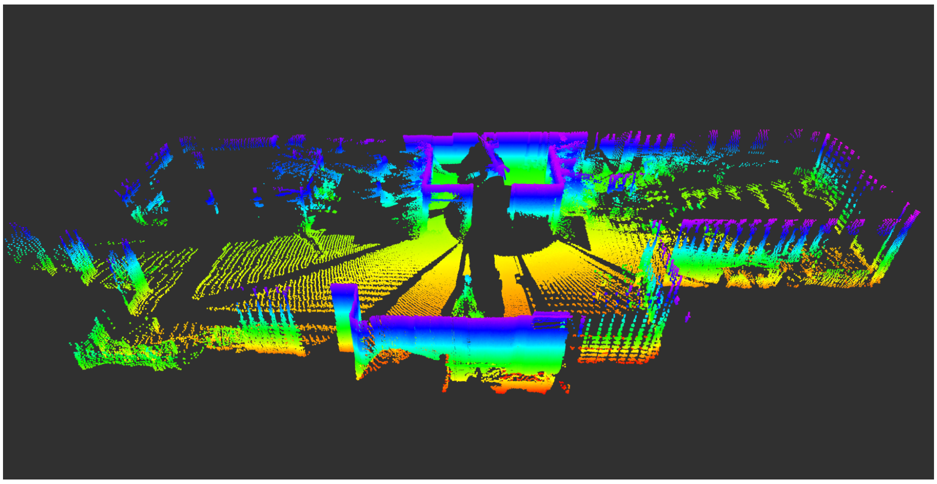
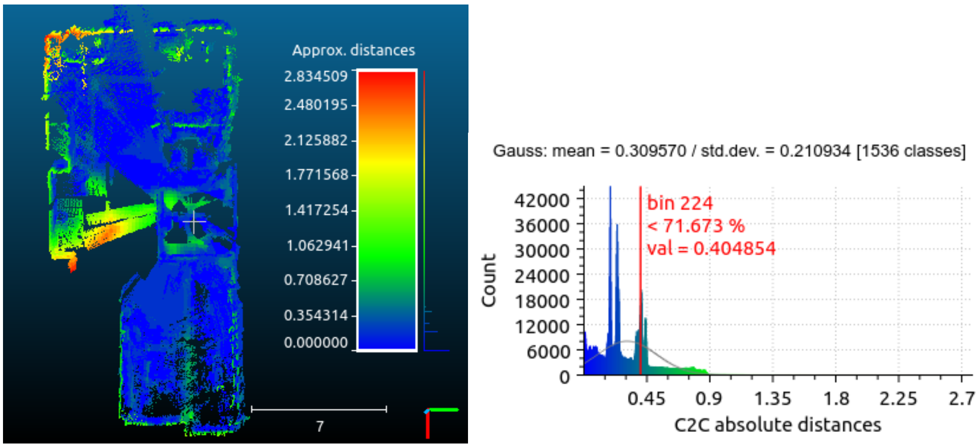


Fig. 5.13 3D map for crane environment.



(a) (b)

Fig. 5.14 Analysis of 3D map for model crane environment. (a) Point-to-Point Distances. Color represents error (cloud-to-cloud distance between ground truth and point cloud). (b) Distribution Fitting of Point-to-Point Distances. Color represents error in distribution fitting.

5.8.2 3D Mapping evaluation

In order to create a 3D map, we configured the rotating base of the 2D lidar to rotate at a constant speed of 6 degrees per second. Figure 5.13 illustrates the resulting 3D point cloud map when the crane's boom is in motion. Each point color represents its corresponding height. Notably, the figure demonstrates that even when the sensor system is continuously moving, we are able to generate a precise and accurate point cloud map.

We compared the 3D map generated using our proposed approach with a ground truth map to assess the accuracy of our method. The ground truth map was constructed by the trajectory obtained from a motion capture system. Figure 5.14(a) depicts the point-to-point distance between the two 3D maps using a color scale that represents distances ranging from 0 to 2.83 meters. The majority of points in the point cloud map are shown as blue, with some green points and only a few red points. This indicates a low overall error, suggesting that our proposed technique is capable of producing an accurate 3D map. Additionally, Figure 5.14(b) presents a distribution fitting plot of the point-to-point distances between the two point clouds. Approximately 71.6% of points have a distance of less than 0.4 meters. Consequently, the point cloud map obtained using our method closely aligns with the ground truth, exhibiting minimal point-to-point distances between them.

5.9 Experimental Results Using Real Crane

The experiment involves testing the proposed sensor pose estimation and 3D mapping method using a large size real crane. The sensor system, comprising a SICK LD-LRS 3611, 2D lidar, XSENS MTI-630 IMU, and realsensor T265 camera, is mounted securely on the crane's boom for data collection as shown in Fig. 5.15 and Fig. 5.16 The setup includes a small PC to collect the sensor data and transmit the data to main laptop. The experiment aims to assess the method's robustness against variations in crane boom rotation speed and different boom trajectories. Data is recorded while the crane is in motion.

Figure 5.17 showcases the trajectory estimated by the multi-sensor fusion-based approach. The 2D trajectory is depicted in Figure 5.17(a), while the corresponding 3D trajectory is presented in Figure 5.17(b). As shown in Fig. 5.17, the trajectory estimated by the multi-sensor fusion-based approach (VINS) exhibits gradual divergence due to cumulative errors over time. These errors stem from the dependence on unstable and inaccurate camera image features for pose estimation. In real-world scenarios, the presence of unpredictable sky features (such as clouds) and dynamic crane components, particularly the crane load ropes illustrated in Fig. 5.19, further contribute to pose estimation errors. Figure 5.19 presents the 3D map generated using the pose estimated through the VINS approach. Notably, pose

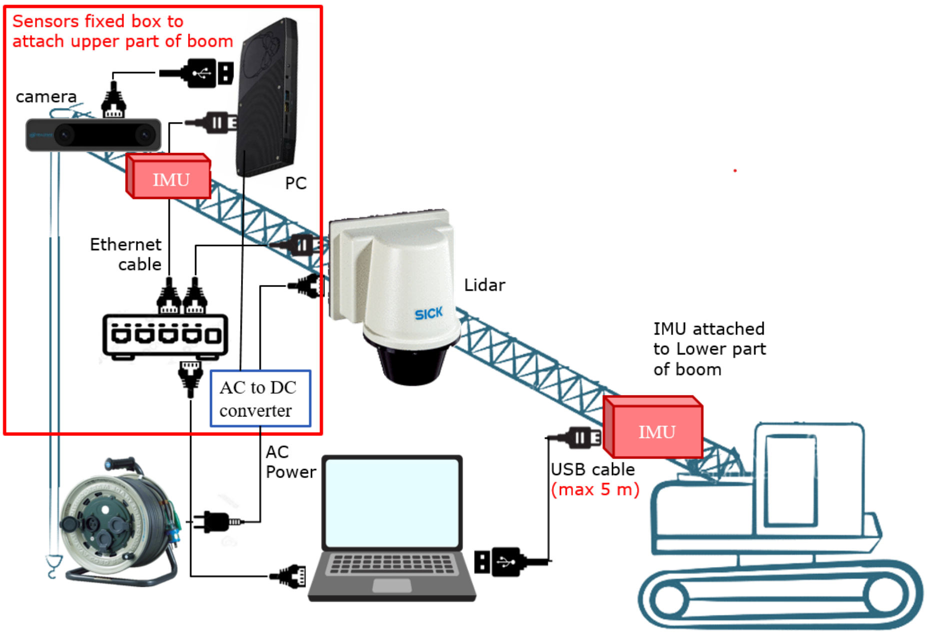
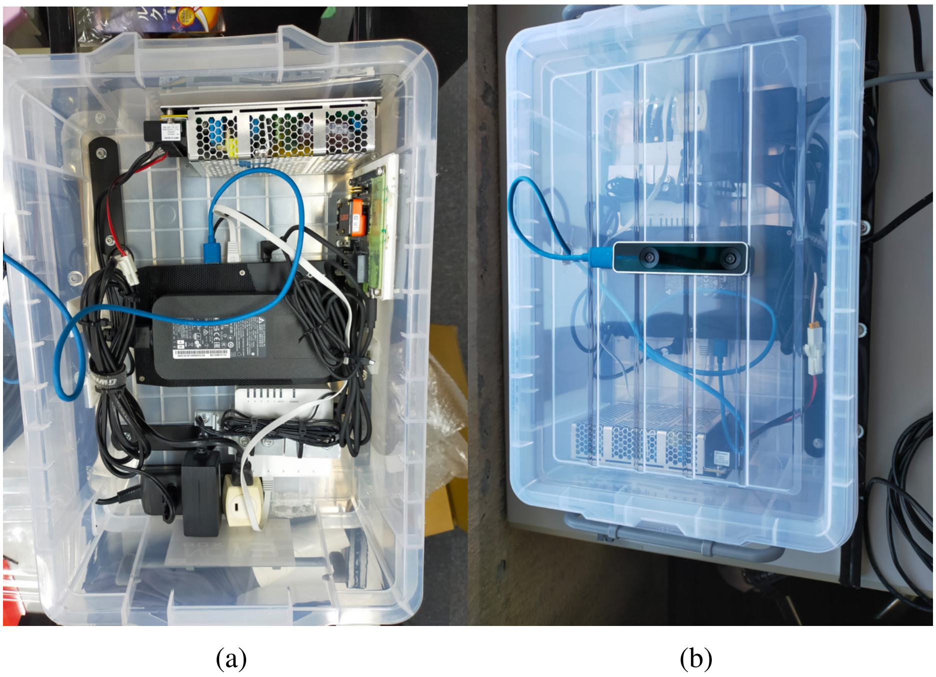


Fig. 5.15 Sensor System attahed to crane boom.



(a) (b)

Fig. 5.16 Sensor System fixed in a box attached to crane boom.

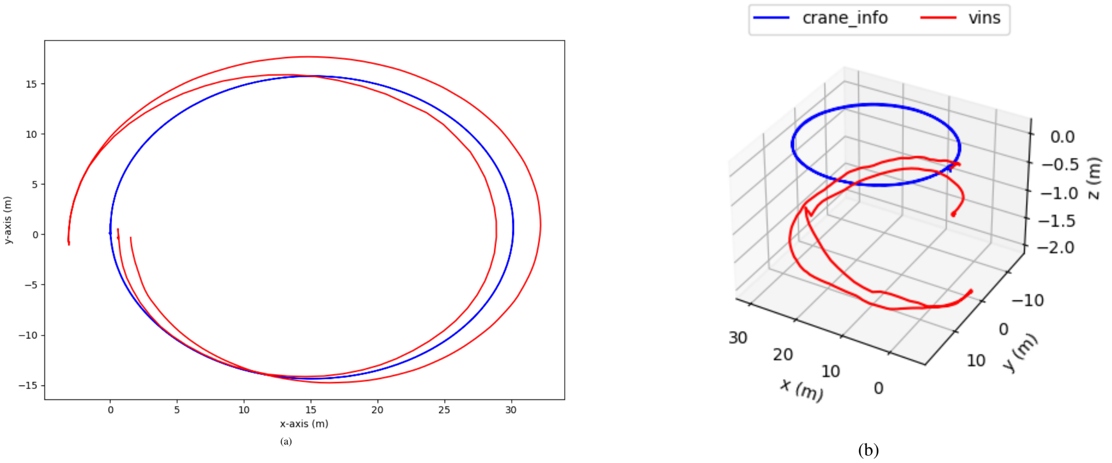


Fig. 5.17 Visualization of the estimated trajectory. (a) illustrates the 2D trajectory of the experiment, while (b) displays the corresponding 3D trajectory.

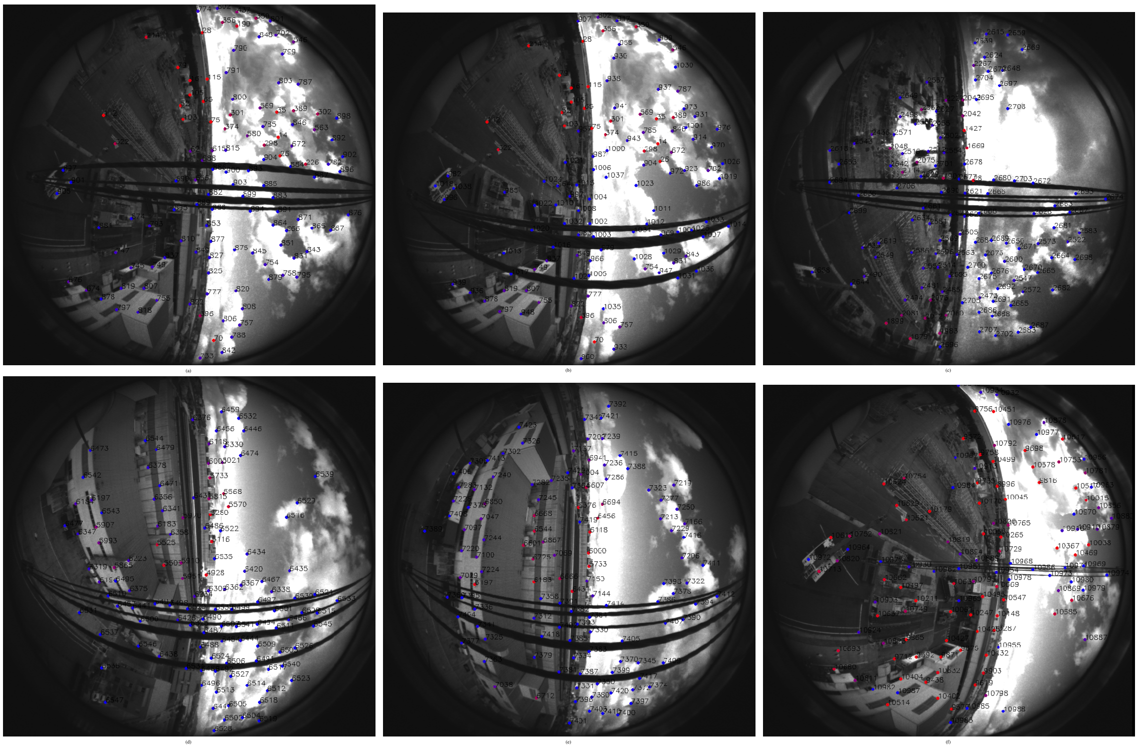
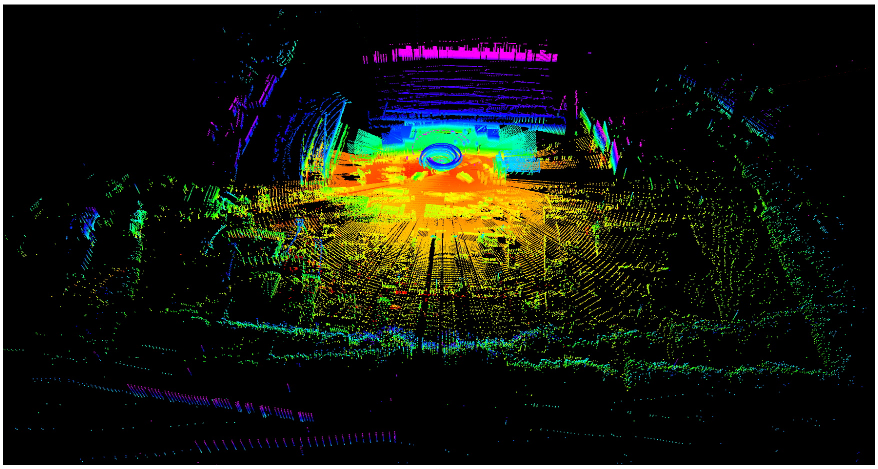
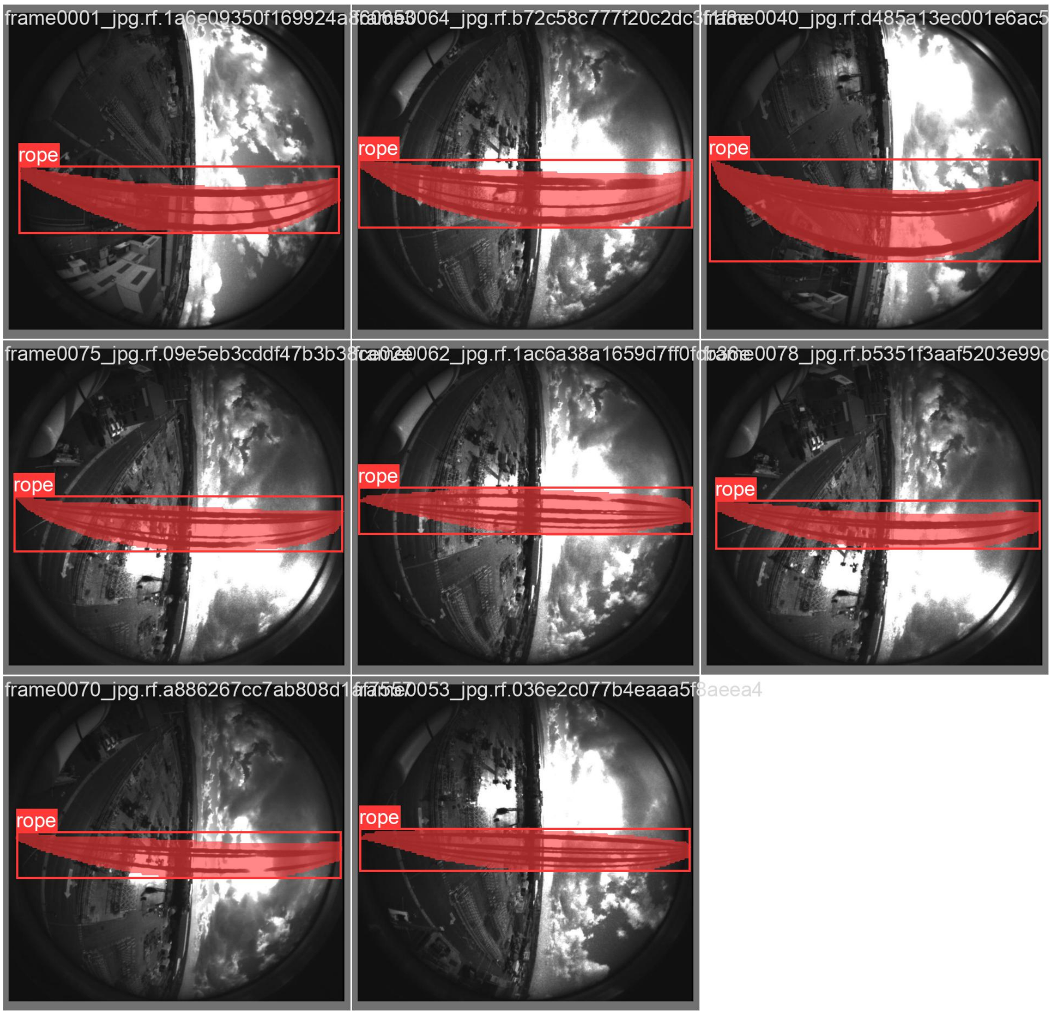


Fig. 5.18 Visualization of the presence of unpredictable sky features (such as clouds) and dynamic crane components, particularly the crane load ropes.



(a)

Fig. 5.19 3D map created using the pose estimated through the VINS-based approach.


frame0001_jpg.rf.1a6e09350f169924a
frame0064_jpg.rf.b72c58c777f20c2dc3
frame0040_jpg.rf.d485a13ec001e6ac5
frame0075_jpg.rf.09e5eb3cddf47b3b3
frame0062_jpg.rf.1ac6a38a1659d7ff0f
frame0078_jpg.rf.b5351f3aaf5203e99c
frame0070_jpg.rf.a886267cc7ab808d1
frame0053_jpg.rf.036e2c077b4eaaa5f8aeea4
rope
rope
rope
rope
rope
rope
rope
rope

(a)

Fig. 5.20 Employing the YOLOv8 model to detect the crane load rope and mask out the portion of the crane load rope.

errors significantly impact the accuracy of the 3D map, resulting in instances of double wall representation for a single wall and the duplication of certain buildings, as observed in Figure 5.19. To address this issue, we are currently employing the YOLOv8 model to detect the crane load rope. Subsequently, we mask out the portion of the crane load rope that contains unreliable features, as depicted in Fig. 5.20 By adopting this approach, we can disregard the problematic region and utilize only reliable features for pose estimation in VINS. Presently, our efforts are focused on seamlessly integrating the YOLOv8-based model into the VINS-based approach.

5.10 Conclusion

In this chapter a method for estimating sensor poses and generating extensive 3D maps for construction cranes is described. The study focuses on construction cranes equipped with a sensor system comprising a camera, 2D lidar, and IMU. To address the complexities arising from the crane boom's motion, an Extended Kalman filter (EKF) is employed to enhance the accuracy and reliability of sensor pose estimation. The proposed method involves combining pose estimates from the Visual-Inertial Navigation System (VINS) with data from an additional IMU to estimate the scale value of a monocular camera. This scale value, obtained from the EKF, is then integrated into the VINS algorithm to refine the previously estimated scale value. The construction of a 3D map is facilitated by employing a slowly rotating 2D lidar. Given the limited overlap between 2D lidar scans, the estimated pose is utilized to align and construct a comprehensive 3D map. The study also includes a comprehensive evaluation of the efficacy of the latest VINS techniques, as well as the EKF-enhanced VINS approach, within the context of crane operations. Extensive performance assessments are conducted in simulated and real environments, comparing the EKF-added VINS method against state-of-the-art VINS techniques. The evaluation results affirm the accurate estimation of sensor poses by the EKF-added VINS method, thereby enabling the generation of high-quality, large-scale 3D point cloud maps for construction cranes.

Our future work includes, firstly, further evaluation of the MSCKF and R-VIO methods in the current approach. These methods failed in our testing, but we plan to use precise IMU calibration parameters and camera intrinsic and extrinsic parameters to again evaluate these methods. Currently, the IMU calibration parameter is obtained using 6-hour static IMU data. However, using more static IMU data can provide more accurate calibration parameters. These methods will undergo further testing by adjusting the various parameters and initialization values. Secondly, in the current mapping method, the 2D lidar scan registration relies solely on estimated pose values. However, this approach leads to the accumulation of errors over time. In future work, these errors can be eliminated by performing scan matching between two point clouds generated by consecutive complete rotations of the lidar sensor.

Chapter 6

Comparative Analysis of Proposed Approaches for Real-Time Pose Estimation and 3D Mapping

6.1 Introduction

In this chapter, our central focus is the comprehensive comparison of accuracy and robustness among three proposed algorithms for real-time pose estimation: the Complementary Filter and Crane Structure-based Approach (elaborated in Chapter 3), IMU-based Neural Network Approach (explained in Chapter 4), and Multi-sensor Fusion-based Approach (explored in Chapter 5). To gauge the efficacy of these methods in real-world scenarios, we conducted evaluations using two distinct types of cranes. One scenario involved a model crane positioned in an indoor environment, while the other featured a real crane situated outdoors. This chapter unfolds a detailed exploration, aiming to discern the performances and effectiveness of these three methods in pose estimation. The subsequent sections delve into the intricate details of this comparative analysis.

6.2 Comparison Using Model Crane

To conduct a comprehensive comparison of all proposed methods using the model crane, we first attached all sensors employed in the proposed methods, including a RealSense T265 camera, Hokuyo UST-20LX 2D lidar, an Orion Giken RHST-PA1L rotating base, and an XSENS MTI-630 IMU, as depicted in Fig. 6.1. In order to compare the proposed methods, we obtained ground truth data from a motion capture system. Given the distinct reference

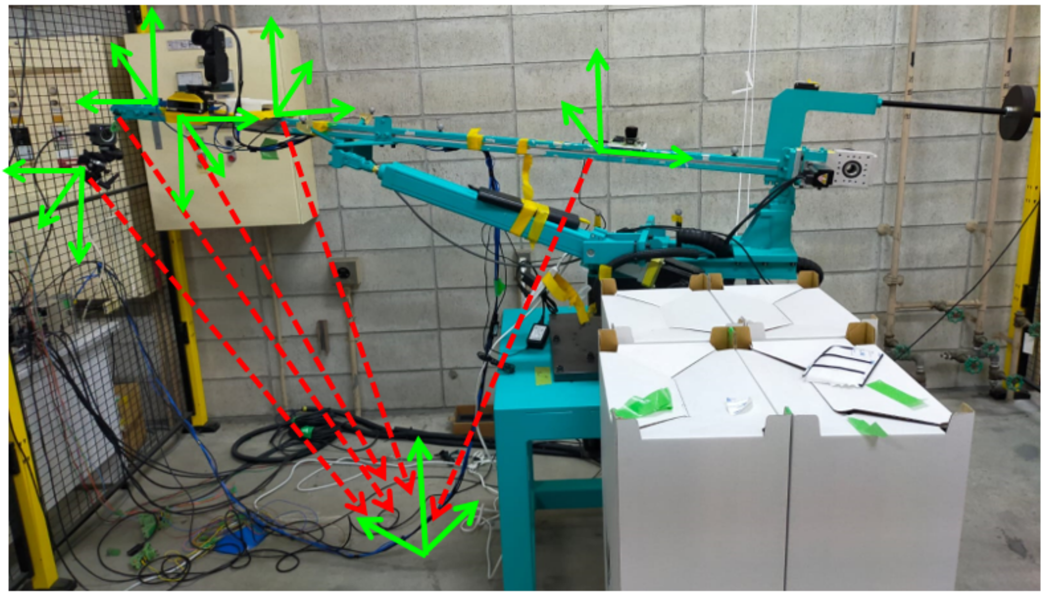
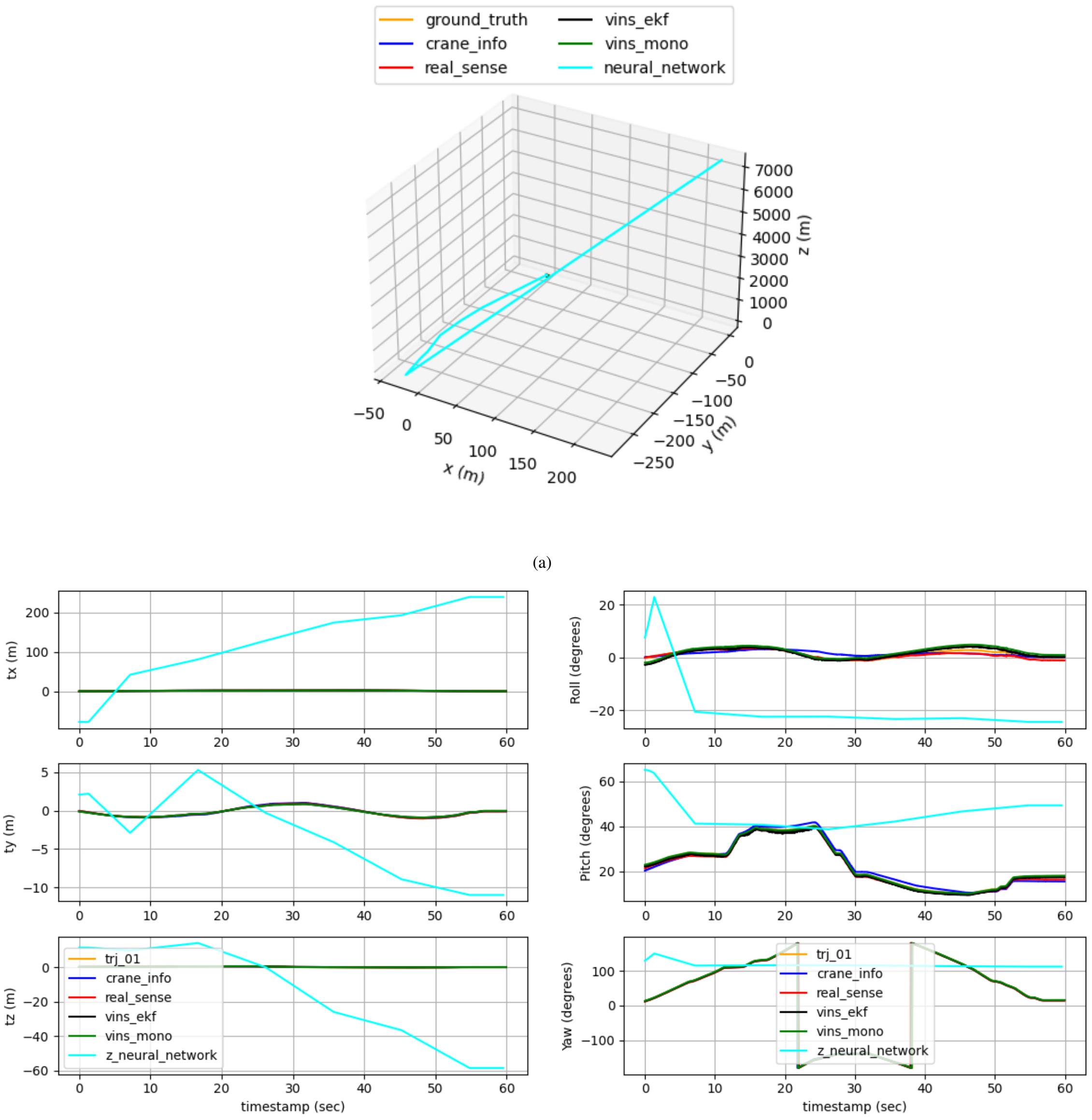


Fig. 6.1 Crane model along with its accompanying sensor system. The sensor system is connected to the boom of the crane.

frames for each proposed method and the ground truth, establishing a common frame of reference was essential. This common frame allowed us to compute the change in pose using each proposed method relative to a unified baseline. Initially, we roughly transferred all frames of reference to this common frame, as illustrated in Fig. 6.1 using red arrows. Subsequently, we employed the sim3 trajectory alignment method, as described in [91, 26], to align the estimated trajectory with the ground truth.

In order to assess the accuracy and robustness of the proposed approaches, a series of experiments were conducted under different scenarios. These scenarios encompassed varying crane boom rotation speeds and challenging crane boom trajectories. The objective was to evaluate the performance of the different approaches in these diverse scenarios and gain insights into their effectiveness. We calculated the Root Mean Square Error (RMSE) and Relative Pose Error (RPE) using [91, 26], to quantify the position and orientation errors of the estimated trajectory over the aligned trajectory. For each scenario, we generated graphs to visually represent the estimated trajectories produced by each proposed method in comparison to the ground truth. Additionally, accompanying graphs illustrate the RMSE at each time step of the trajectory, providing insights into periods where estimation performance may be compromised. Furthermore, the RPE graph offers a comprehensive view of error computation across segments of the dataset, enabling an examination of how localization solutions drift as the trajectory lengthens. This quantitative analysis, supported by visual

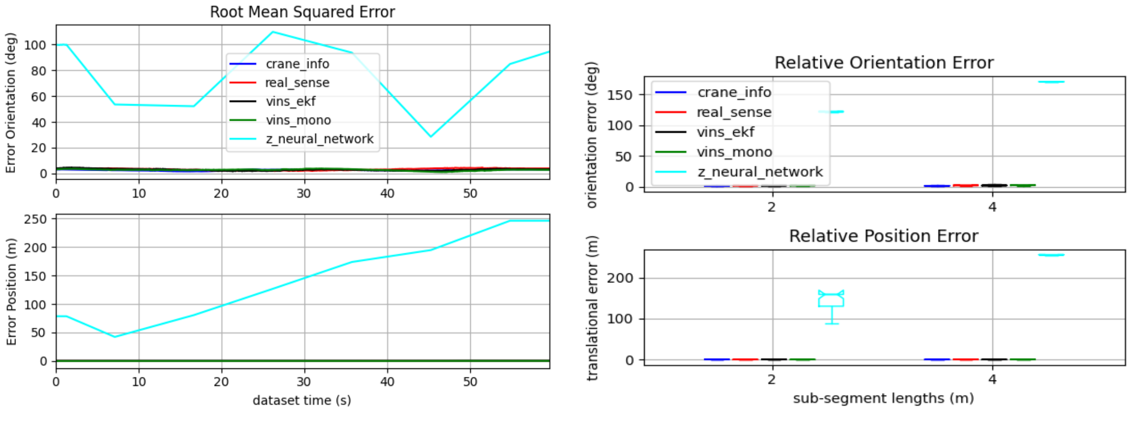


(a)

(b)

(c)

Fig. 6.2 illustrates the trajectory during the experiment in scenarios of Smooth and Uniform Crane Boom Motion (a) depicts the 3D trajectory of the experiment. (b) illustrates the changes in roll, pitch, and yaw over time, while (c) presents the variations in the x, y, and z components of translation with respect to time.



(a)

(b)

Fig. 6.3 shows the errors in trajectory estimation during the experiment in scenarios of Smooth and Uniform Crane Boom Motion. (a) shows the Root Mean Square Error (RMSE), and (b) presents the Relative Pose Error (RPE).

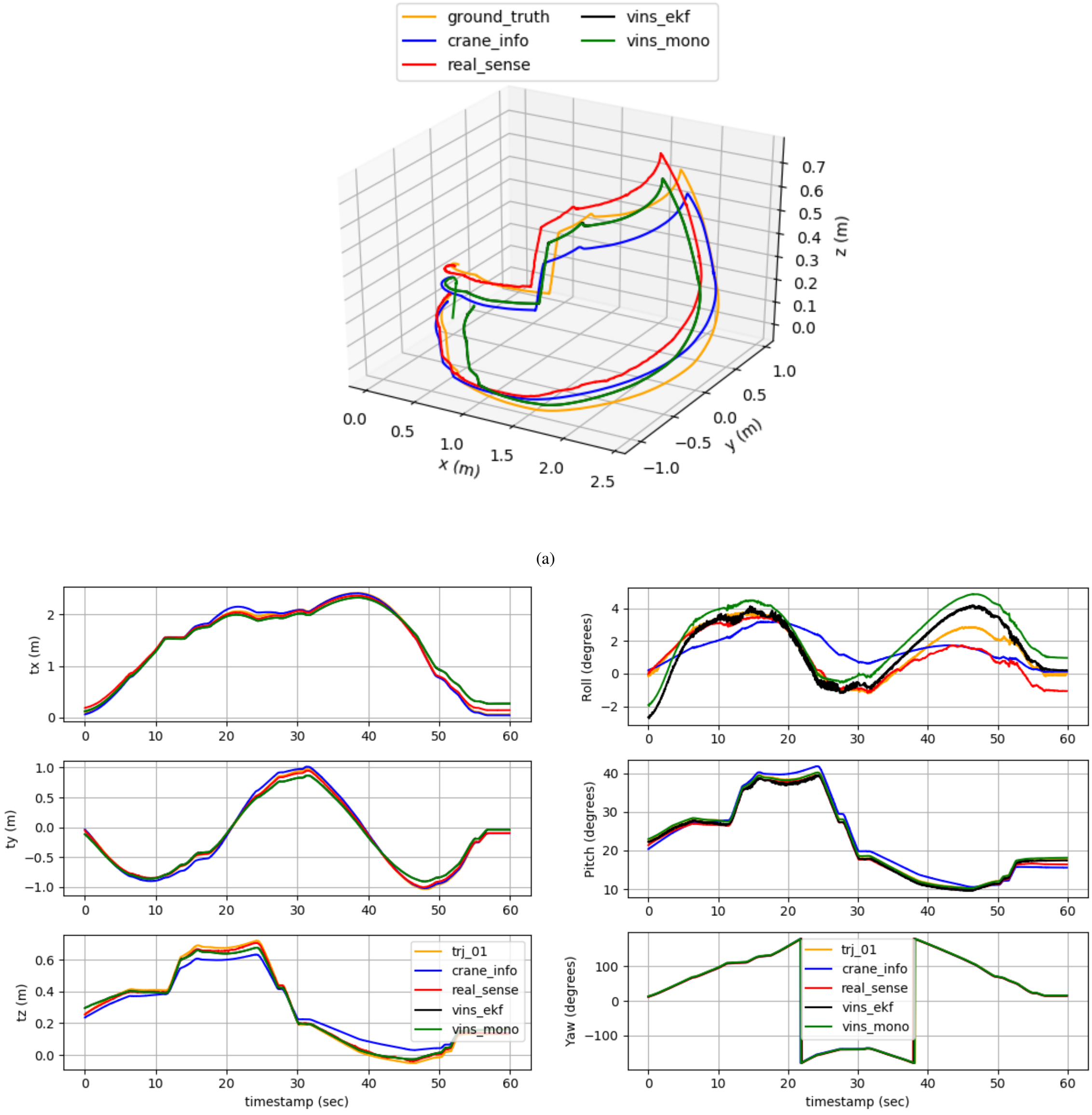
representations, contributes to a thorough understanding of the comparative performance of the proposed methods in real-world scenarios.

Due to space constraints, this section presents the comparison results for three distinct scenarios, each featuring a crane boom with a unique motion pattern, as explained below:

6.2.1 Scenario 1: Smooth and Uniform Crane Boom Motion

The 3D trajectory of this experiment is depicted in Fig. 6.2(a). The changes in the x, y, and z components of translation with respect to time are presented in Fig. 6.2(b), while the change in roll, pitch, and yaw over time are illustrated in Fig. 6.2(c). The RMSE and RPE are shown in Fig. 6.3(a) and Fig. 6.3(b) respectively. As illustrated in the Fig. 6.2 and Fig. 6.3, the IMU-based Neural Network Approach fails to provide accurate pose estimations in real-world scenarios, evident in the divergence of the z_neural_network curve over time. This limitation arises from training the neural network with simulated crane data, while real-world testing involves a different crane model than the one used in simulation. Performance evaluation underscores the shortcomings of the Neural Network Approach when applied to real-world data trained with simulated data.

Due to the failure of the IMU-based Neural Network Approach, it is excluded, and other methods are analyzed. Excluding the Neural Network Approach, the trajectory of the remaining proposed methods for this experiment is shown in Fig. 6.4(a), the change in the x, y, and z components of translation is shown in Fig. 6.4(b), as well as the changes in roll, pitch, and yaw over time is shown in Fig. 6.4(c). Similarly, excluding the Neural Network Approach, Fig. 6.5(a) and Fig. 6.5(b) displays the RMSE and RPE of the remaining proposed

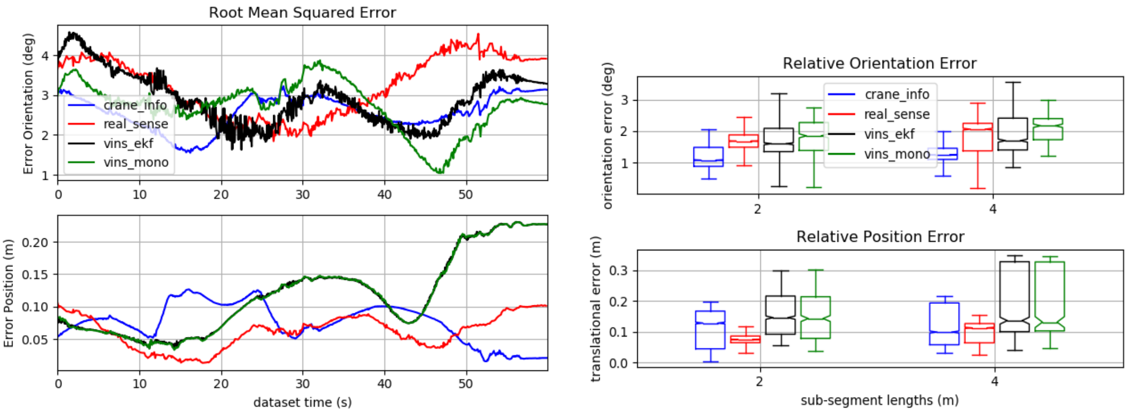


(a)

(b)

(c)

Fig. 6.4 illustrates the trajectory excluding the Neural Network Approach, during the experiment in scenarios of Smooth and Uniform Crane Boom Motion (a) depicts the 3D trajectory of the experiment. (b) illustrates the changes in roll, pitch, and yaw over time, while (c) presents the variations in the x, y, and z components of translation with respect to time.



(a)

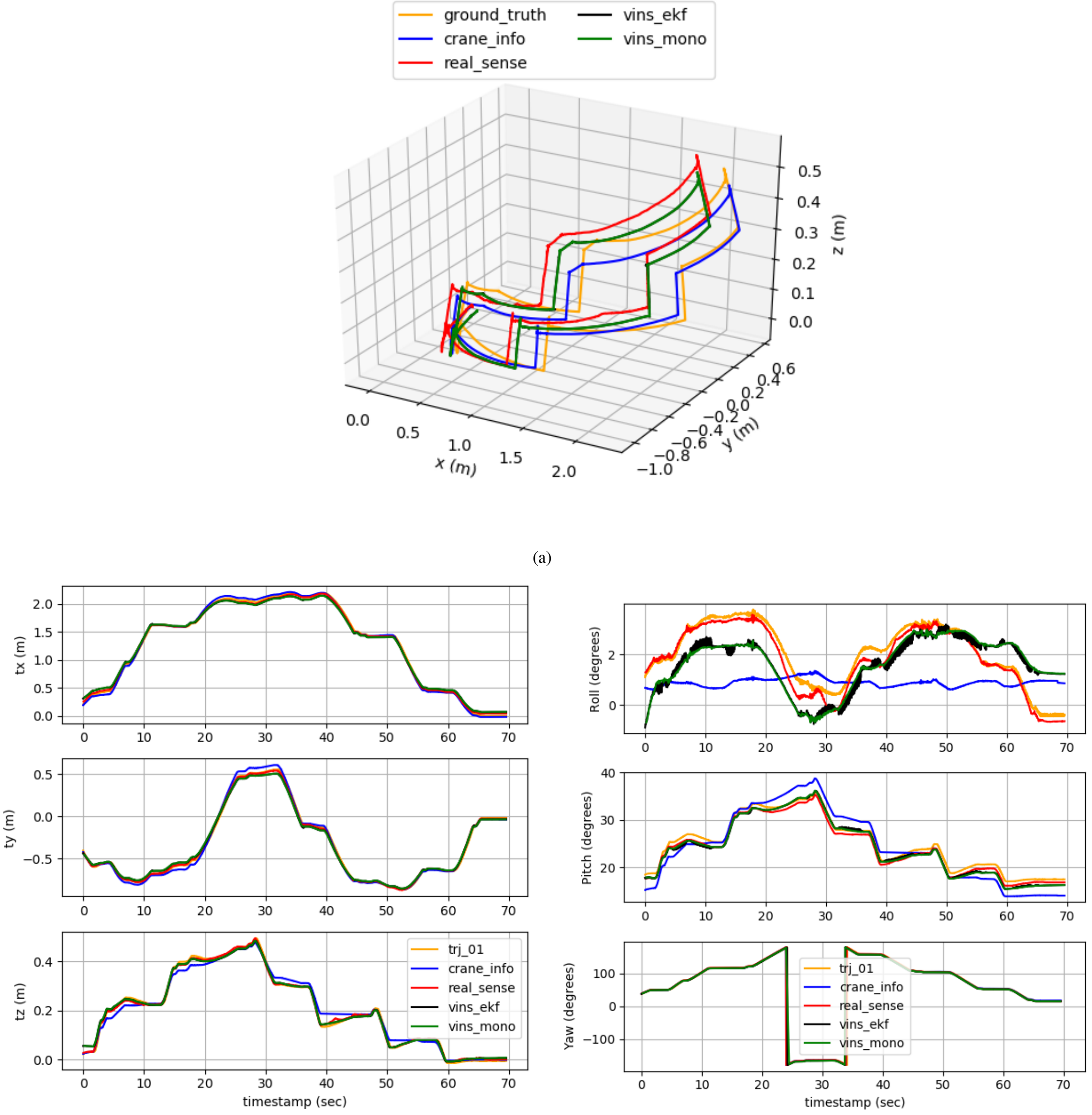
(b)

Fig. 6.5 shows the errors in trajectory estimation excluding the Neural Network Approach, during the experiment in scenarios of Smooth and Uniform Crane Boom Motion. (a) shows the Root Mean Square Error (RMSE), and (b) presents the Relative Pose Error (RPE).

methods receptively. In scenarios characterized by smooth, uniform, and continuous crane boom motion, the performance of Multi-sensor Fusion-based Approach methods (vins_mono and vins_ekf) is similar. Notably, the orientation estimation accuracy of the Crane Structure-based Approach (illustrated by the "crane_info curve") exhibits a slight superiority over all other methods. Similarly, the position estimation accuracy of the Realsense slightly outperforms that of other methods.

6.2.2 Scenario 2: Sudden or Abrupt Crane Boom Movements

The 3D trajectory of this experiment is represented in Fig. 6.6(a), and the changes in the x, y, and z components of translation over time are depicted in Fig. 6.6(b). Additionally, the changes in roll, pitch, and yaw over time are shown in Fig. 6.6(c). Correspondingly, the RMSE is portrayed in Fig. 6.7(a), and the RPE is shown in Fig. 6.7(b). In scenarios involving sudden or abrupt crane boom movements, the fusion of VINS and EKF (vins_ekf) showcases a marginally superior performance for rotation estimation when compared to the standalone VINS (vins_mono). This superiority arises from the incorporation of another IMU data within vins_ekf fusion systems, enhancing their capacity to adeptly manage rapid changes in the crane boom's motion dynamics. The Crane Structure-based Approach, as demonstrated by the crane_info curve, exhibits a slight superiority in orientation estimation accuracy over all other methods and the Realsense shows a marginally better position estimation accuracy compared to other methods.

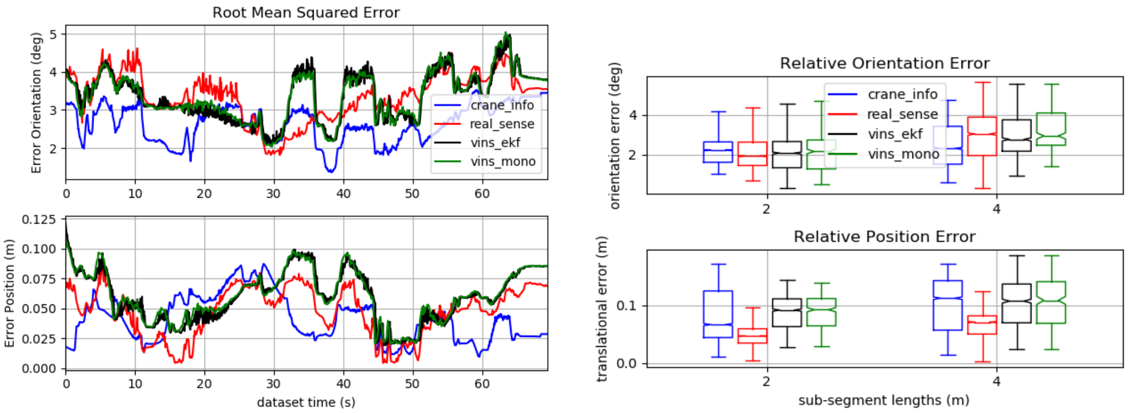


(a)

(b)

(c)

Fig. 6.6 illustrates the trajectory during the experiment in scenarios of Sudden or Abrupt Crane Boom Movements (a) depicts the 3D trajectory of the experiment. (b) illustrates the changes in roll, pitch, and yaw over time, while (c) presents the variations in the x, y, and z components of translation with respect to time.



(a)

(b)

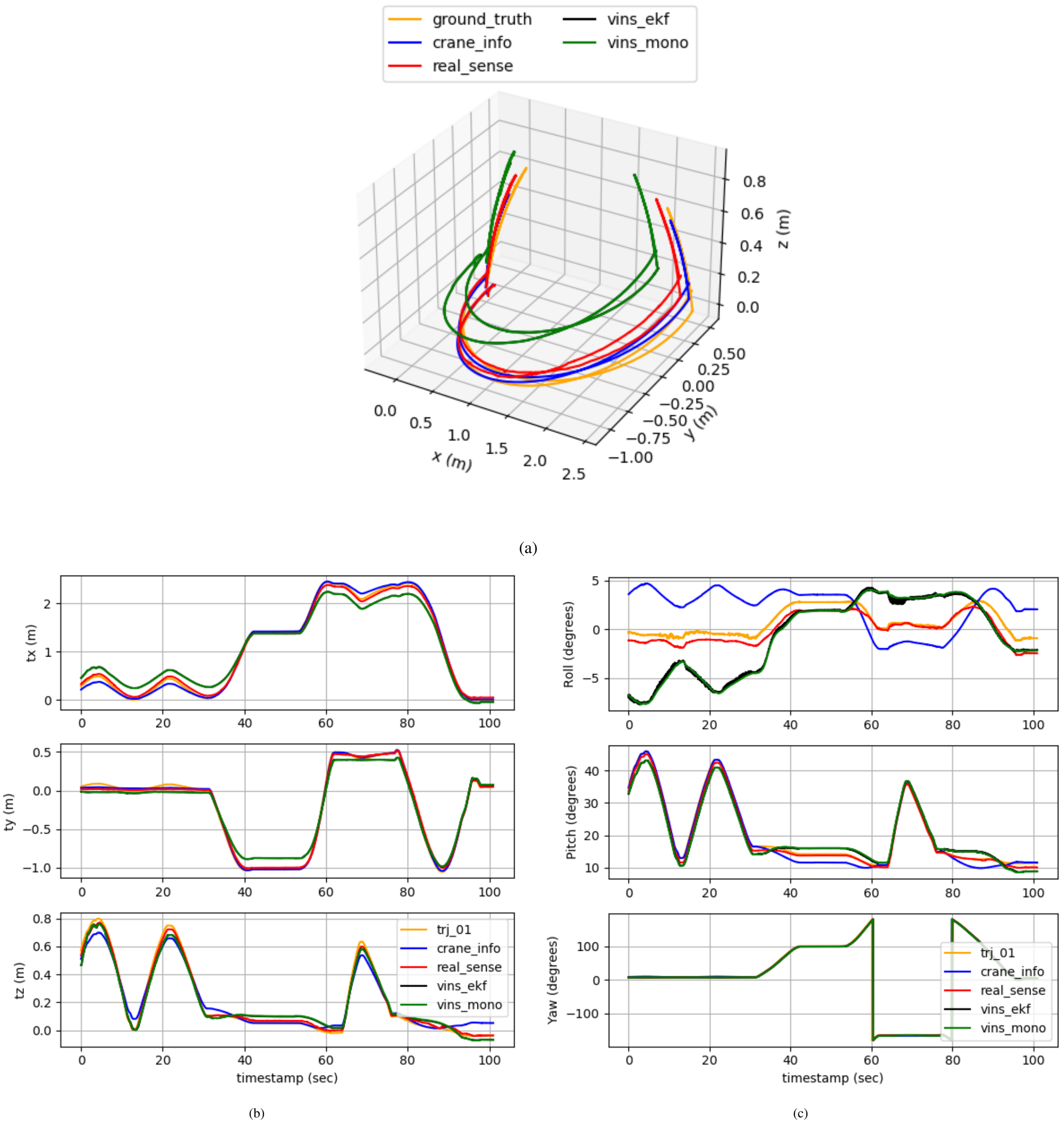
Fig. 6.7 shows the errors in trajectory estimation during the experiment in scenarios of Sudden or Abrupt Crane Boom Movements. (a) shows the Root Mean Square Error (RMSE), and (b) presents the Relative Pose Error (RPE).

6.2.3 Scenario 3: Crane Boom's Vertical Movement

The 3D trajectory of this experiment is illustrated in Fig. 6.8(a), while the change in the translation components (x, y, z) over time are presented in Fig. 6.8(b). Additionally, Fig. 6.8(c) displays the change in roll, pitch, and yaw throughout the experiment. The RMSE is visualized in Fig. 6.9(a), and Fig. 6.9(b) showcases the RPE. During the crane boom's vertical movement, VINS accuracy is compromised due to the absence of distinctive and reliable ceiling features essential for VINS as reference points. This limitation impedes its ability to effectively track the crane's movement, resulting in a noticeable compromise in performance during such scenarios. In contrast, Crane Structure-based Approach proves resilient against these challenges, operating independently of specific features. It consistently demonstrate accuracy regardless of the crane's vertical movement, showcasing their reliability even in situations where identifiable features are scarce or absent. The Crane Structure-based Approach, as demonstrated by the crane_info curve, exhibits a superiority in pose estimation accuracy over all other methods.

6.2.4 Collective Position and Rotation Errors

To comprehensively assess positional and rotational estimation errors across diverse trajectories and scenarios, the collective position and rotation errors are vividly illustrated in the accompanying Fig. 6.10. The orientation estimation accuracy of the Crane Structure-based Approach surpasses that of all other methods. While both the Realsense and Crane Structure-based Approach demonstrate similar accuracy in position estimation, it is noteworthy that the

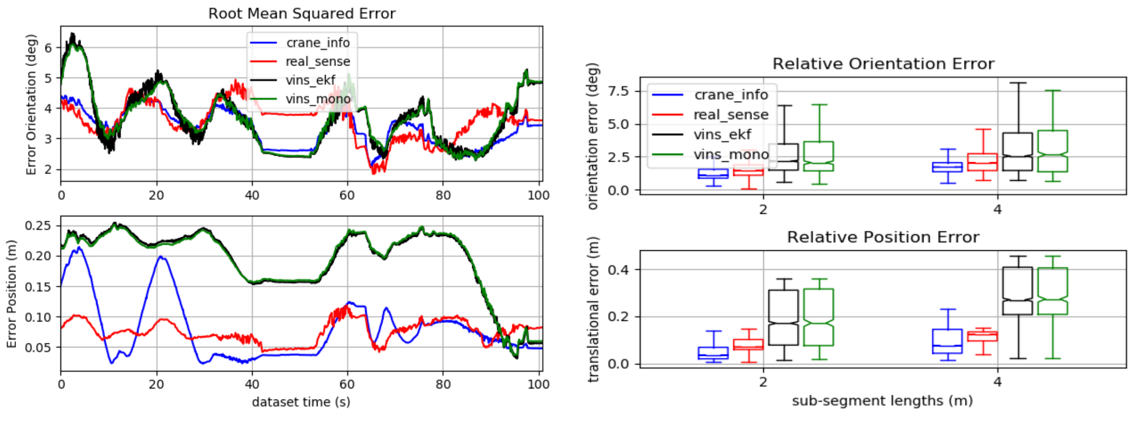


(a)

(b)

(c)

Fig. 6.8 illustrates the trajectory during the experiment in scenarios of Crane Boom's Vertical Movement (a) depicts the 3D trajectory of the experiment. (b) illustrates the changes in roll, pitch, and yaw over time, while (c) presents the variations in the x, y, and z components of translation with respect to time.



(a)

(b)

Fig. 6.9 shows the errors in trajectory estimation during the experiment in scenarios of Crane Boom's Vertical Movement. (a) shows the Root Mean Square Error (RMSE), and (b) presents the Relative Pose Error (RPE).

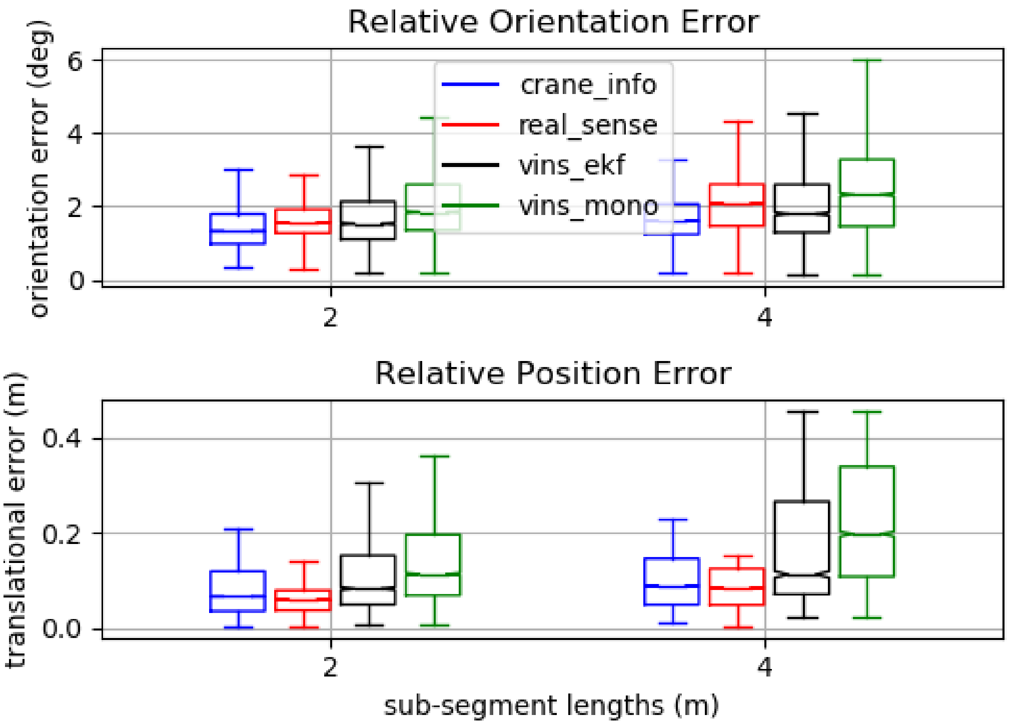


Fig. 6.10 shows the collective position and rotation errors in trajectory estimation of different approaches.

Crane Structure-based Approach tends to exhibit more scattered errors in position estimation compared to the Realsense.

6.3 Comparison Using Kobelco Real Crane

To conduct a comprehensive comparison of all proposed methods using the kobelco real crane, we first attached all sensors employed in the proposed methods, including a RealSense T265 camera, Sick LD-LRS LD-LRS3611, and two XSENS MTI-630 IMU, as depicted in Fig. 6.11.

In order to assess the accuracy and robustness of the proposed approaches, a series of experiments were conducted under different scenarios. These scenarios encompassed varying crane boom rotation speeds and challenging crane boom trajectories. The objective was to evaluate the performance of the different approaches in these diverse scenarios and gain insights into their effectiveness. Due to space constraints, this section presents the comparison results for three distinct scenarios, each featuring a crane boom with a unique motion pattern, as explained below:

6.3.1 Scenario 1: Simple Crane Boom Motion With Varying Speed

The crane operation follows these given steps, moving its boom along a specific path shown in the Fig. 6.12

(a) Set the crane boom angle to 50 degrees at crane boom home position.

(b) Perform an anti-clockwise complete cycle (360 degree yaw rotation) at a slow speed to return to the crane boom home position.

(c) Then perform a clockwise complete cycle (360 degree yaw rotation) at a fast speed to return to the crane boom home position.

(d) Finally, stop.

The trajectory estimated by the Crane Structure-based Approach (crane/info), the Multi-sensor Fusion-based Approach (vins), and the RealSense sensor, is presented in Fig. 6.13. Fig. 6.13(a) illustrates the temporal changes in the x, y, and z components of translation, while Fig. 6.13(b) depicts the change in roll, pitch, and yaw overtime. The 2D trajectory of the experiment is showcased in Fig. 6.13(c), and the corresponding 3D trajectory is presented in Fig. 6.13(d).

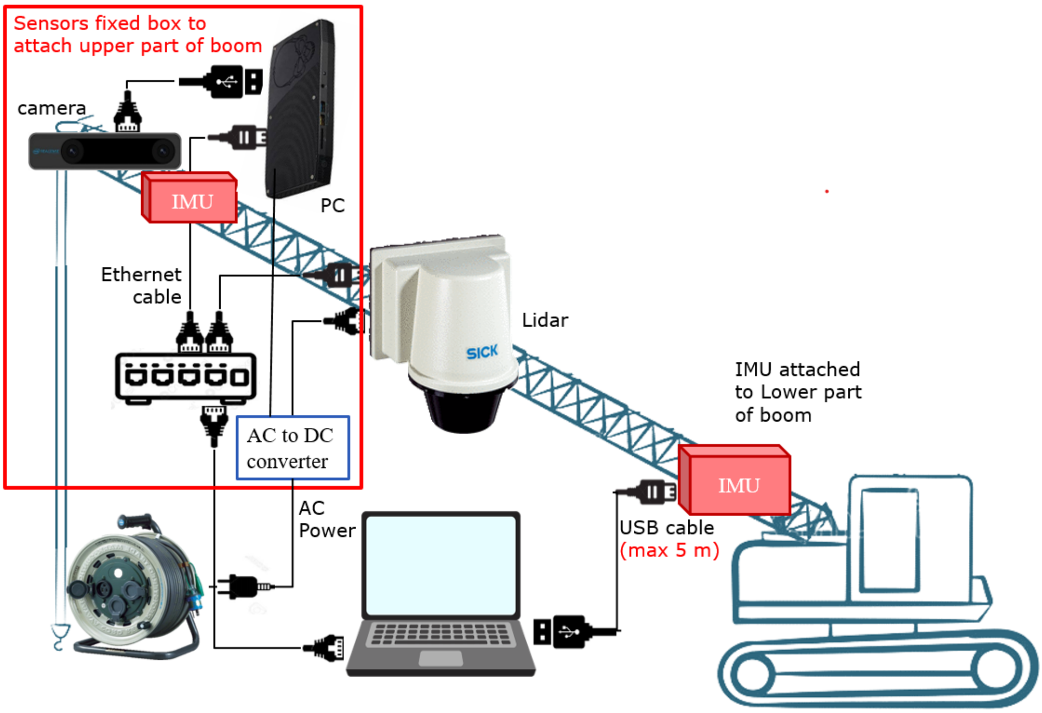


Fig. 6.11 Crane along with its accompanying sensor system. The sensor system is connected to the boom of the crane.

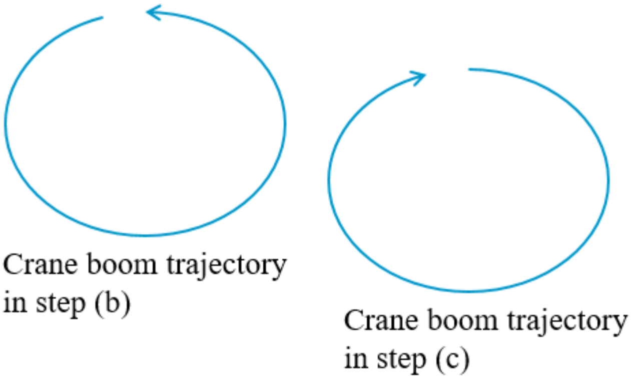
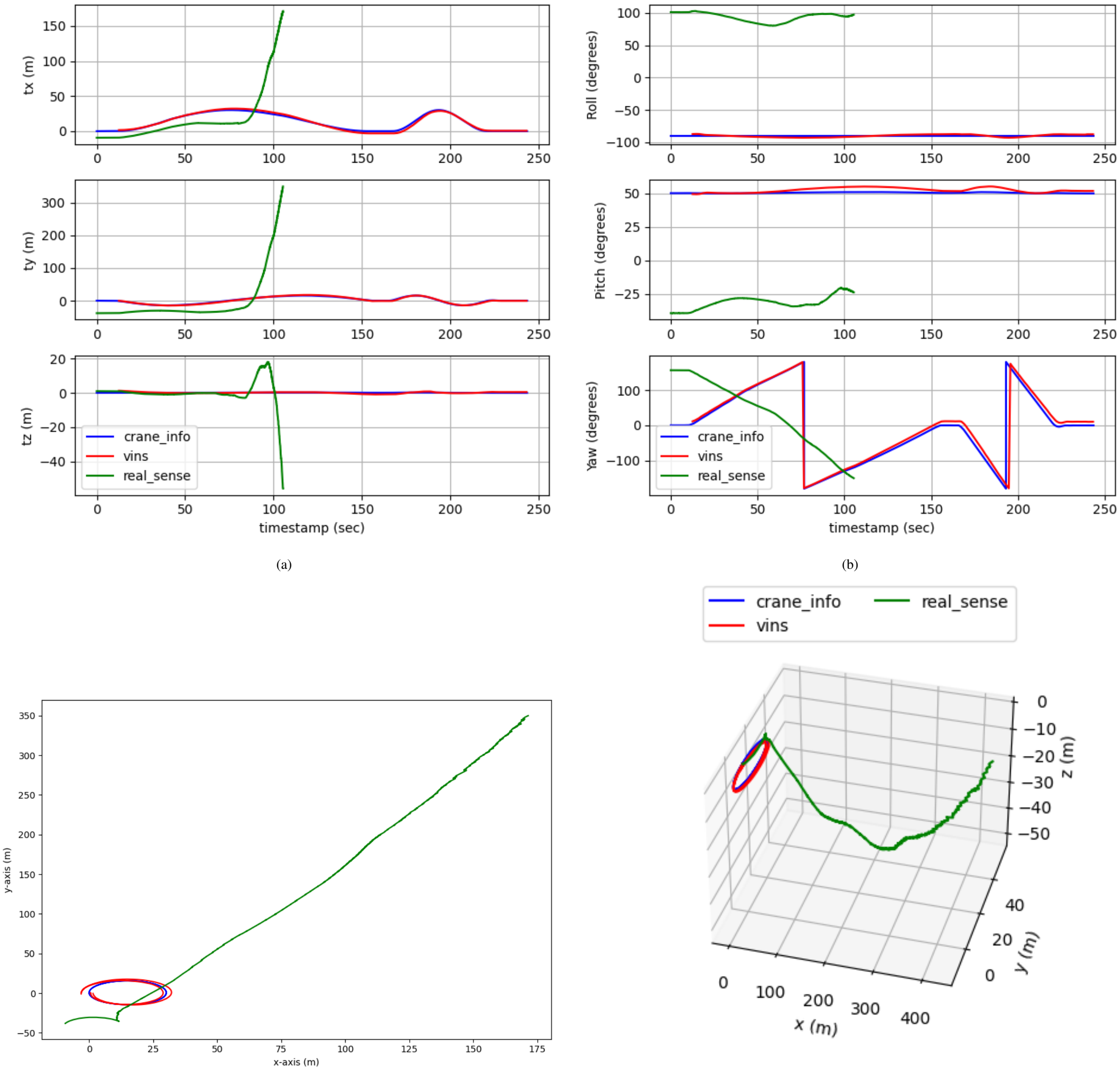

Crane boom trajectory in step (b)
Crane boom trajectory in step (c)

Fig. 6.12 During Scenario 1, tracjectory followed by sensor system attached to crane boom.



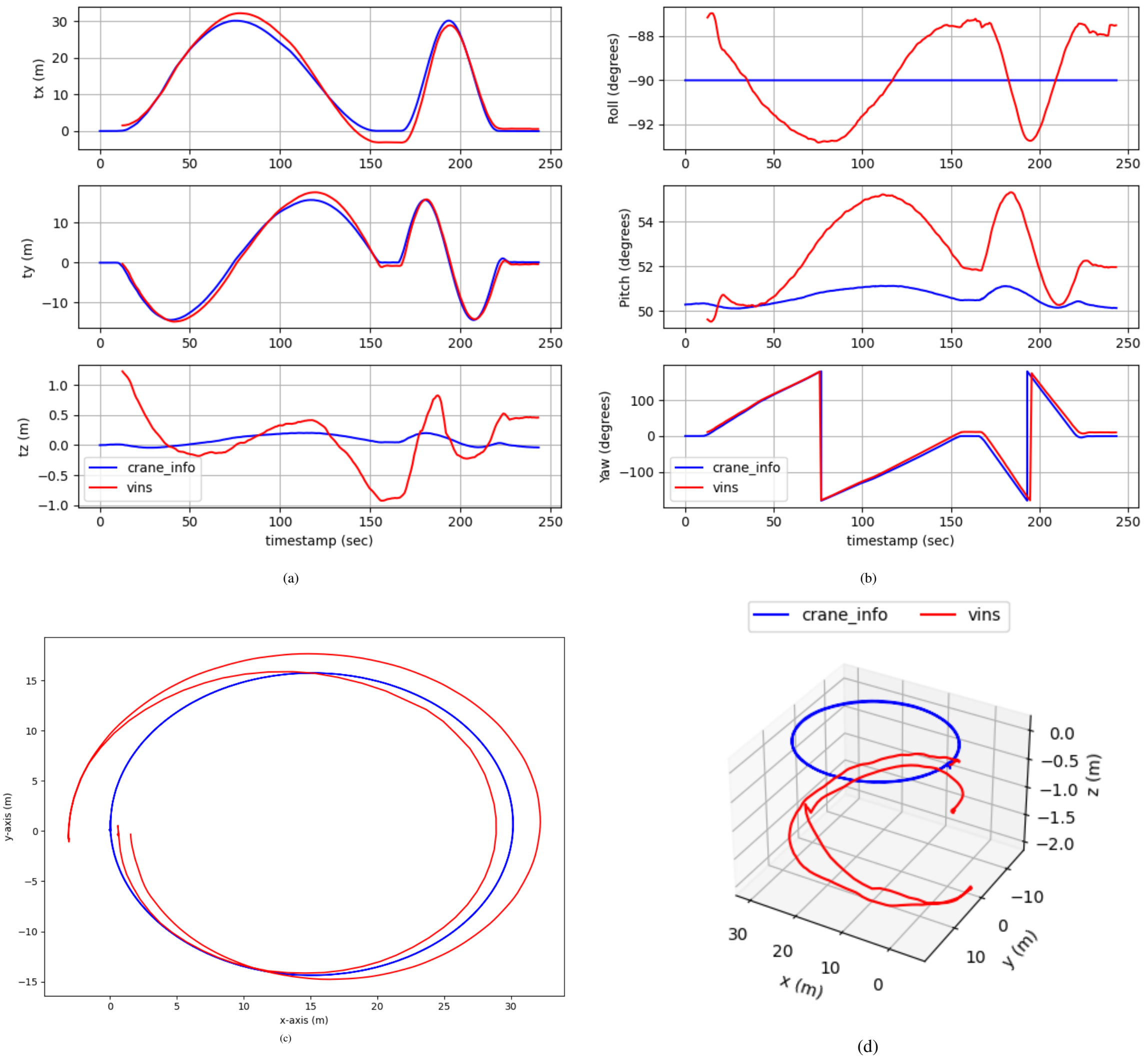
(a)

(b)

(c)

(d)

Fig. 6.13 Depiction of the trajectory as estimated by the proposed methods in scenario 1 is shown in (a), indicating change in the x, y, and z components of translation with respect to time. In (b), the alterations in roll, pitch, and yaw are illustrated. The 2D trajectory of the experiment is portrayed in (c), and (d) showcases the corresponding 3D trajectory.



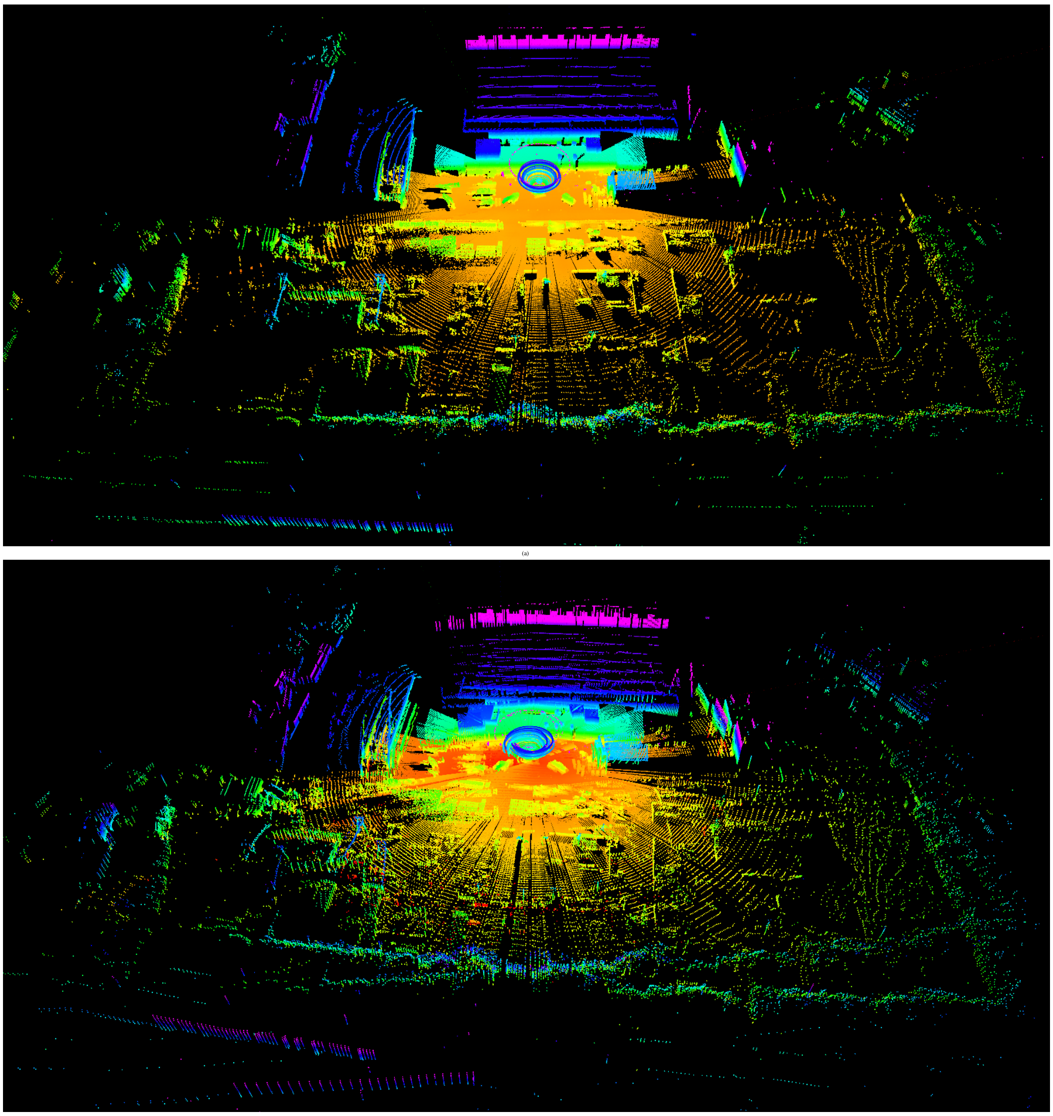
(a)

(b)

(c)

(d)

Fig. 6.14 Illustration of the trajectory estimated by the proposed methods in scenario 1 excluding the trajectory estimated by real-sense. Subfigure (a) displays the variations in the x, y, and z components of translation over time, while subfigure (b) illustrates the changes in roll, pitch, and yaw. Subfigure (c) depicts the 2D trajectory of the experiment, and subfigure (d) presents the 3D trajectory.



(a)

(b)

Fig. 6.15 3D map generated by using the pose estimation from the proposed method. Subfigure (a) shows the 3D map constructed using the pose estimated through the crane_info-based approach, while subfigure (b) illustrates the 3D map generated with the pose estimated via the VINS-based approach.

As depicted in Fig. 6.13, the trajectory derived from the RealSense sensor exhibits a noticeable divergence and ultimately fails to yield accurate results. Our analysis indicates that the RealSense-based trajectory performs well in indoor environments, as demonstrated in section 6.2. However, in outdoor environments with sensors positioned at significant heights above the ground and attached to a sizable crane boom, RealSense-based pose estimation faces challenges in achieving accurate precision. The trajectory, determined by the Crane Structure-based Approach (crane/info), and the Multi-sensor Fusion-based Approach (vins), excluding realsense, is displayed in Fig. 6.14. In Fig. 6.14(a), changes in the x, y, and z components of translation over time are showcased, while Fig. 6.14(b) illustrates alterations in roll, pitch, and yaw. The 2D experiment trajectory is presented in Fig. 6.14(c), and Fig. 6.14(d) exhibits the corresponding 3D trajectory.

As observed in Fig. 6.14, the trajectory estimated by the Crane Structure-based Approach (crane/info) aligns accurately with the trajectory developed following the provided instructions for crane operation during the experiment. However, the trajectory estimated by the multi-sensor fusion-based approach (vins) gradually diverges over time, attributed to error accumulation inherent in this approach. The Multi-sensor Fusion-based Approach relies on camera image features to estimate the pose, and the presence of unstable and unstatic features in the sky, including clouds, along with dynamic features introduced by the crane structure such as the rope, becomes the primary cause of errors in this method.

In Fig. 6.15(a), the 3D map constructed using the crane structure-based approach is displayed, showcasing its high accuracy and precision. Fig. 6.15(b) illustrates the 3D map generated using the pose estimated through the multi-sensor fusion-based approach (vins). The errors in pose estimation significantly impact the accuracy of the 3D map. Instances of double wall representation for a single wall and the duplication of certain buildings in the 3D map (as seen in Fig. 6.15(b)) result from pose errors in the vins approach.

6.3.2 Scenario 2: Crane Boom Motion With Varying Boom Angle During Cycle (4 cycle)

The crane adheres to these provided instructions, guiding its boom along a particular trajectory.

(a) Set the boom angle to 80 degrees at crane boom home position.

(b) Perform an anti-clockwise complete cycle (360 degree yaw rotation) with changing boom angle during the cycle at slow speed and return to the crane boom home position.

(c) Followed by a clockwise complete cycle (360 degree yaw rotation) with changing boom angle during the cycle at slow speed to return to crane boom home position.

(d) Repeat the procedure from (a) to (c) one time.

(e) Finally, stop.

Fig. 6.16 displays the trajectory estimated by the crane structure-based approach (crane/info), the Multi-sensor Fusion-based Approach (vins), and the RealSense sensor. Temporal changes in the x, y, and z components of translation are detailed in Fig. 6.16(a), while Fig. 6.16(b) visually represents changes in roll, pitch, and yaw. The 2D trajectory of the experiment is portrayed in Fig. 6.16(c), and the corresponding 3D trajectory is showcased in Fig. 6.16(d).

In Fig. 6.16, the trajectory estimated by the crane structure-based approach (crane/info) closely matches the trajectory derived from the provided crane operation instructions. However, the trajectory obtained through the multi-sensor fusion-based approach (vins) gradually deviates over time, attributed to accumulating errors inherent in this method. The reliance on camera image features for pose estimation in the Multi-sensor Fusion-based Approach, coupled with the presence of unstable atmospheric features like clouds and dynamic elements such as the crane's rope, emerges as the main source of errors in this approach.

In Fig. 6.17(a), the 3D map created using the crane structure-based approach is depicted, highlighting its notable accuracy and precision. Fig. 6.17(b) demonstrates the 3D map generated using the pose estimated through the multi-sensor fusion-based approach (vins). Pose estimation errors significantly affect the precision of the 3D map. The presence of double wall representations for a single wall and the duplication of specific buildings in the 3D map (as observed in Fig. 6.17(b)) is attributed to pose errors in the vins approach.

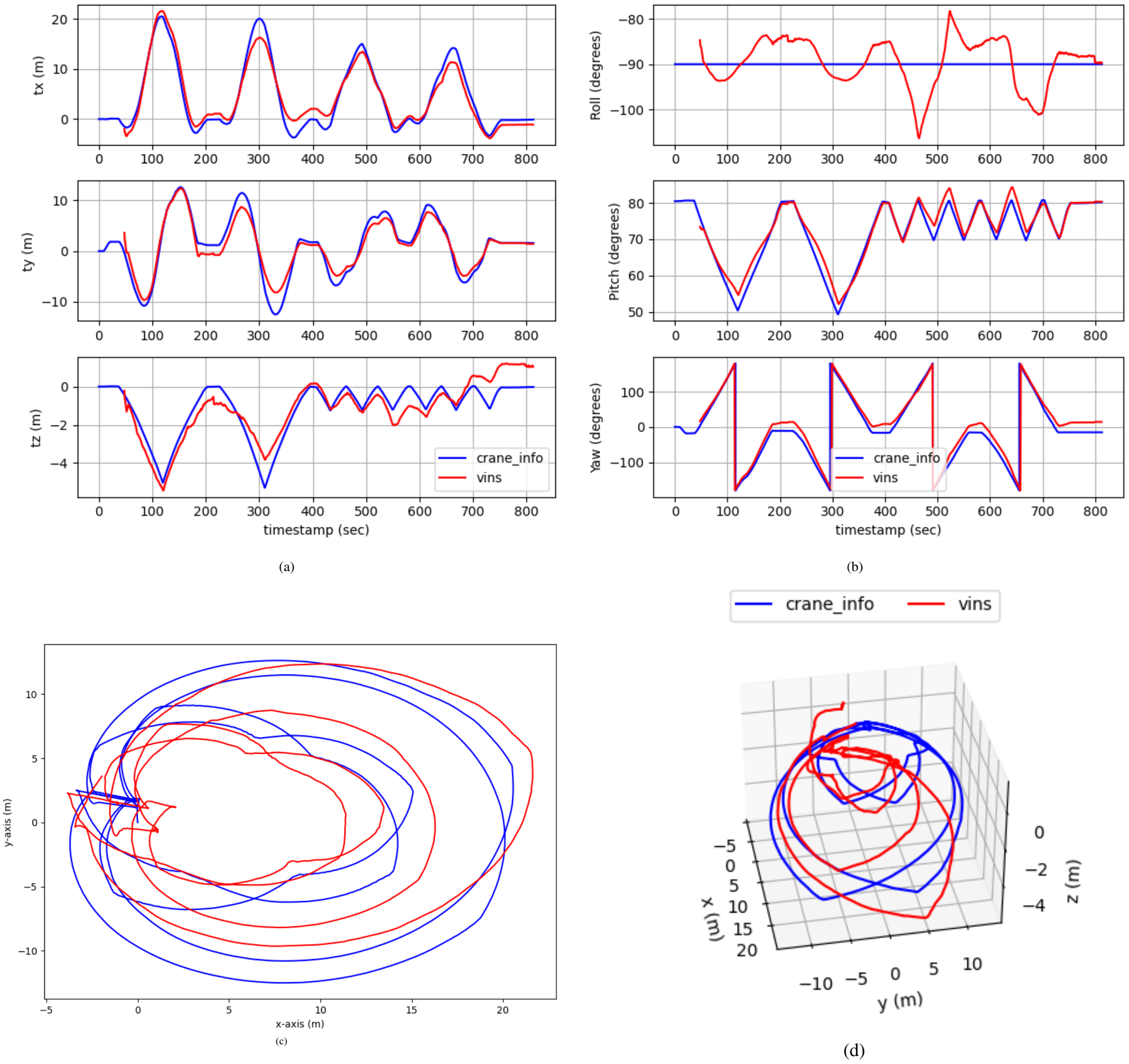
6.3.3 Scenario 3: Crane Boom Motion With Varying Speed And Boom Angle At The Start Of Cycle (6 Cycle)

The crane adheres to these provided instructions, guiding its boom along a specific trajectory illustrated in the Fig. 6.12

(a) Set the boom angle to 80 degrees at crane boom home position.

(b) Perform an anti-clockwise complete cycle (360 degree yaw rotation) at a slow speed and return to the crane boom home position.

(c) Then at crane boom home position, set the boom angle to 50 degrees.



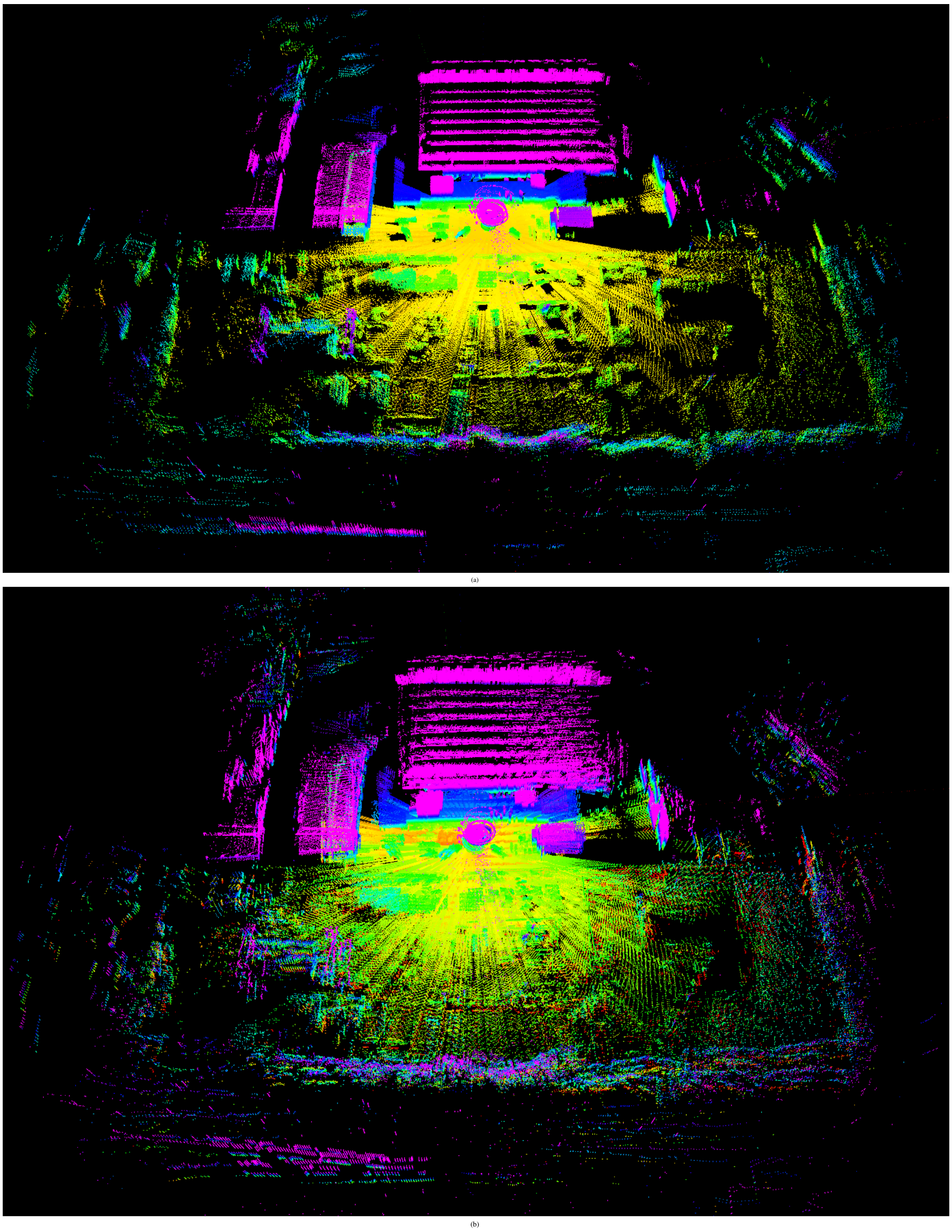
(a)

(b)

(c)

(d)

Fig. 6.16 Illustration of the trajectory estimated by the proposed methods in scenario 2. Subfigure (a) displays the variations in the x, y, and z components of translation over time, while subfigure (b) illustrates the changes in roll, pitch, and yaw. Subfigure (c) depicts the 2D trajectory of the experiment, and subfigure (d) presents the 3D trajectory.


(a)

(b)

Fig. 6.17 3D map is constructed utilizing the pose estimated by the proposed method . In subfigure (a), the 3D map is formed based on the pose estimated using the crane_info approach, while in subfigure (b), the pose estimated through the VINS approach is used to create the 3D map.

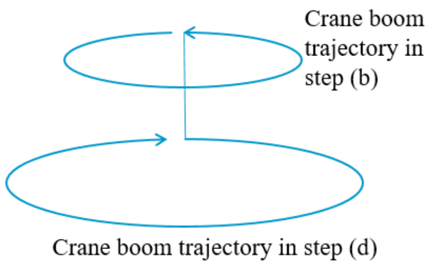

Crane boom trajectory in step (b)
Crane boom trajectory in step (d)

Fig. 6.18 During Scenario 3, trajectory followed by sensor system attached to crane boom.

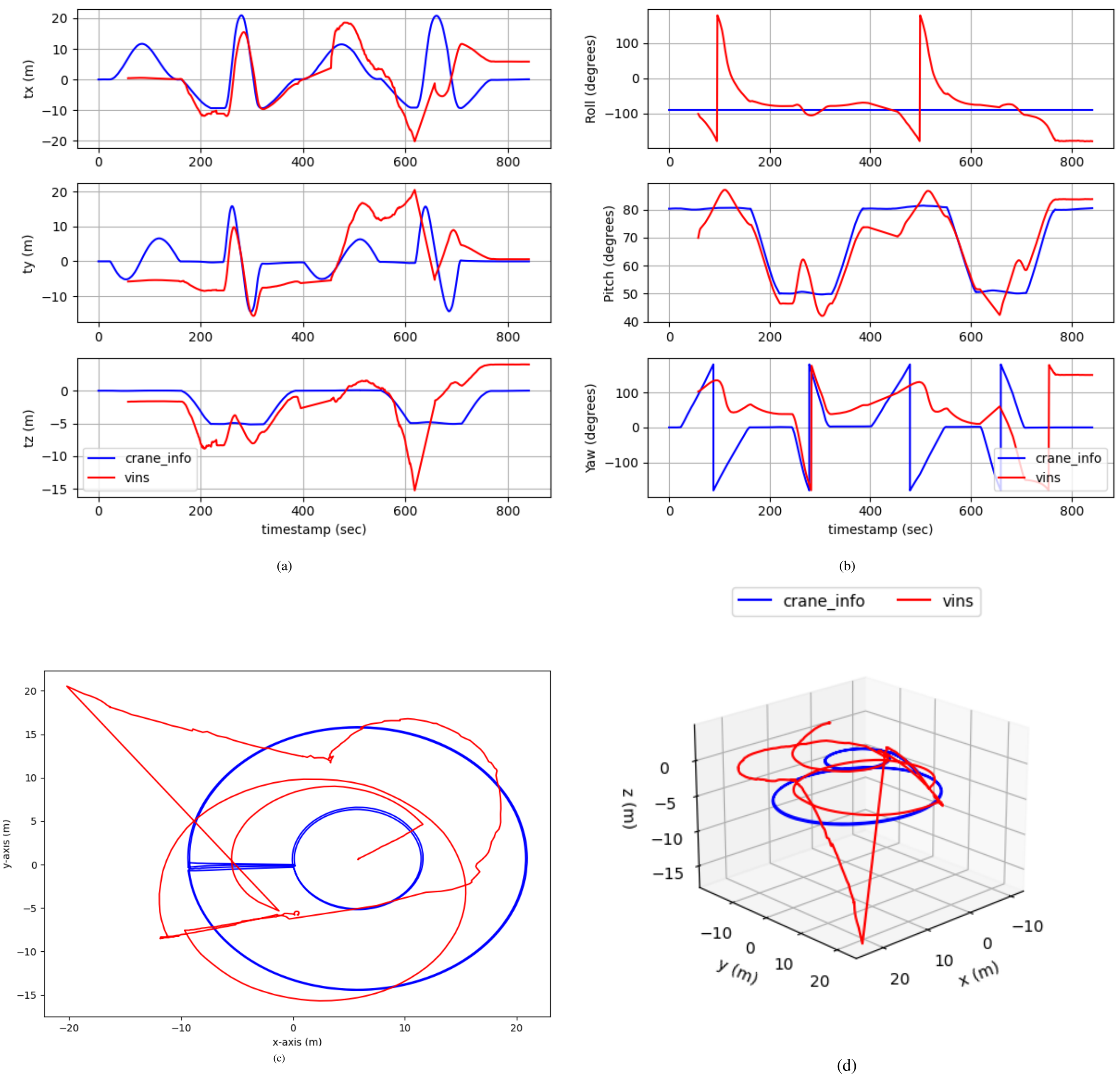
(d) Then perform a clockwise complete cycle (360 degree yaw rotation) at a fast speed and return to the crane boom home position.

(e) Repeat the procedure from (a) to (d) two times.

(f) Finally, stop after two cycles.

In Fig. 6.19, we observe the trajectory estimated by the Crane Structure-based Approach (crane/info), the Multi-sensor Fusion-based Approach (vins), and the RealSense sensor. The graph in Fig. 6.19(a) illustrates the changes in the x, y, and z components of translation with respect to time, while Fig. 6.19(b) provides a visual representation of the alterations in roll, pitch, and yaw. Additionally, Fig. 6.19(c) showcases the 2D trajectory of the experiment, and Fig. 6.19(d) presents the corresponding 3D trajectory.

In Fig. 6.19, the trajectory estimated by the crane structure-based approach (crane/info) is accurate, while the trajectory obtained through the multi-sensor fusion-based approach (vins) fails to accurately estimate the pose. The main cause of this failure is attributed to the presence of unstable atmospheric features like clouds and dynamic elements such as the crane's rope in this approach. In Fig. 6.20(a), the 3D map created using the crane structure-based approach is depicted, emphasizing its significant accuracy and precision. However, in Fig. 6.20(b), the multi-sensor fusion-based approach (vins) is unable to construct a 3D map successfully.



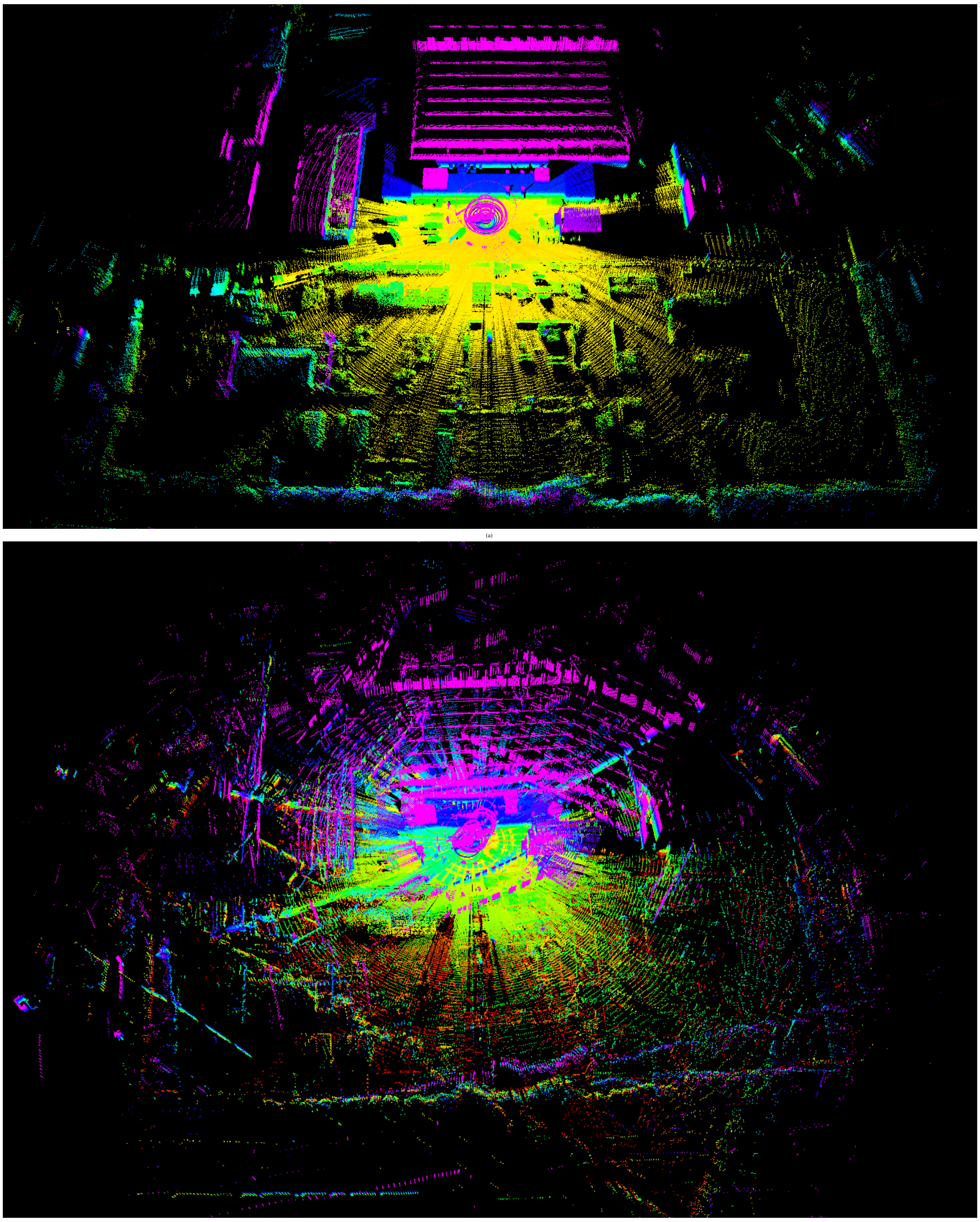
(a)

(b)

(c)

(d)

Fig. 6.19 The trajectory estimated by the proposed methods in scenario 3 is portrayed in (a), revealing variations in the x, y, and z components of translation over time. The changes in roll, pitch, and yaw are visually represented in (b). The experiment's 2D trajectory is depicted in (c), while (d) presents the 3D trajectory.



(a)

(b)

Fig. 6.20 The proposed method’s pose estimation is employed to create a 3D map. Subfigure (a) displays the 3D map generated with the pose estimated through the crane_info-based approach, and subfigure (b) exhibits the 3D map created using the pose estimated through the VINS-based approach.

6.4 Conclusion

The central focus of this chapter is to comprehensively compare the accuracy and robustness of three proposed algorithms for real-time pose estimation: the complementary filter and crane structure-based approach, IMU-based neural network approach, and multi-sensor fusion-based approach and real sensor-based pose estimation. Evaluations were carried out using two types of cranes in different scenarios—indoors with a model crane and outdoors with a real crane. The Crane Structure-based Approach yielded the most accurate pose estimation results in both environments. However, the real sensor-based approach was accurate indoors but unsuccessful outdoors, and the accuracy of vins pose estimation was influenced by the presence of features in images.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The creation of a 3D map is of great importance for autonomous systems navigating in unfamiliar environments. The application of 3D mapping spans various fields, including autonomous driving, service robotics, agriculture, augmented reality, and construction.

In this study, three novel methods for multisensor fusion-based crane mapping are proposed.

First, we propose a method for real-time sensor pose estimation and 3D mapping that is based on complementary filters and crane structures. In this technique, we use a complementary filter in series with moving average filtering, combined with the structural information of the crane, to estimate the sensor pose at each scan, even under the boom vibration. Using the estimated sensor poses, we convert a set of 2D scans into a 3D point cloud map. To further improve the map, we also developed a new pose graph optimization approach that extracts planar structures in the environment and introduces them as additional nodes in the pose graph. We evaluated the proposed method in simulation and real-world experiments. The experimental results show that our method can effectively estimate the sensor trajectory, build an accurate 3D point cloud map, and outperform one of the state-of-the-art methods.

Second, we present a method for large-scale 3D mapping and neural network-based real-time odometry using an IMU and a slowly rotating 2-D LiDAR. In this technique, the window of IMU readings is pre-filtered using a low-pass filter before being sent as input to the neural network to estimate the change in position and rotation. A convolutional neural network (CNN) and LSTM make up the neural network (LSTM). To create a large-scale map, the predicted sensor pose is utilized to register the scans of a 2D-rotating LiDAR. The proposed approach is tested in a gazebo environment.

Third, our approach combines multiple sensors, including a camera, a rotating 2D lidar, and an IMU on the crane boom, to create a real-time 3D map. Using an Extended Kalman Filter (EKF), we fuse sensor measurements to estimate accurate sensor poses. The method is evaluated in a simulation, demonstrating its effectiveness in precise sensor position estimation and large-scale mapping.

7.2 Future Work

Here we summarize some limitations of the proposed 3D mapping methods and discuss future research directions: In our complementary filter and crane structure-based mapping method, pose estimation using IMU can run in real time, while the pose-graph optimization-based map correction part takes a long time for large-scale mapping. Developing a more efficient map correction algorithm is future work.

The limitation of the IMU-based neural network approach for the mapping method is that 3D mapping is not very precise. Our future work includes, firstly, the integration of learning-based inertial odometry and traditional integration-based methods to get more accurate 6D odometry and, secondly, minimizing the mapping error by introducing the close loop during registration of the LiDAR scan.

The results of multi-sensor fusion-based mapping methods are preliminary. I am working on implementing the proposed method in a real-world environment using the crane.

References

[1] Mahdi Abolfazli Esfahani, Han Wang, Keyu Wu, and Shenghai Yuan. Aboldeepio: A novel deep inertial odometry network for autonomous vehicles. volume 21, pages 1941–1950, 2020. doi: 10.1109/TITS.2019.2909064.

[2] Mohammad Aldibaja, Noaki Suganuma, Reo Yanase, and Keisuke Yoneda. Reliable graph-slam framework to generate 2d lidar intensity maps for autonomous vehicles. In *2020 IEEE Veh. Technol. Conf. (VTC2020-Spring)*, pages 1–6, 2020.

[3] Sean Anderson and Timothy D. Barfoot. Ransac for motion-distorted 3d visual sensors. In *2013 IEEE/RSJ Int. Con. Intel. Robot. Syst.*, pages 2093–2099, 2013. doi: 10.1109/IROS.2013.6696649.

[4] Aguiar AS, Neves Dos Santos F, Sobreira H, Boaventura-Cunha J, and Sousa AJ. Localization and mapping on agriculture based on point-feature extraction and semiplanes segmentation from 3d lidar data. 2022. doi: 10.3389/frobt.2022.832165.

[5] Jared B. Bancroft and Gerard Lachapelle. Data fusion algorithms for multiple inertial measurement units. volume 11, pages 6771–6798, 2011. doi: 10.3390/s110706771. URL https://www.mdpi.com/1424-8220/11/7/6771.

[6] Michael Bosse and Robert Zlot. Continuous 3d scan-matching with a spinning 2d laser. In *2009 IEEE Int. Conf. Robot. Autom.*, pages 4312–4319, 2009.

[7] Igor Brigadnov, Aleksandr Lutonin, and Kseniia Bogdanova. Error state extended kalman filter localization for underground mining environments. volume 15, 2023. doi: 10.3390/sym15020344. URL https://www.mdpi.com/2073-8994/15/2/344.

[8] Russell Buchanan. allan_variance_ros - allan variance ros package. GitHub repository, Accessed 2023. URL https://github.com/ori-drs/allan_variance_ros.

[9] Jaemin Byun, K.-I Na, B.-S Seo, and Myungchan Roh. Drivable road detection with 3d point clouds based on the mrf for intelligent vehicle. volume 105, pages 49–60, 01 2015. doi: 10.1007/978-3-319-07488-7_4.

[10] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. volume abs/1802.02209, 2018.

[11] Changhao Chen, Peijun Zhao, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham, and Niki Trigoni. Deeplearning-based pedestrian inertial navigation: Methods, data set, and on-device inference. volume 21, pages 4431–4441, 2020. URL https://doi.org/10.1109/JIoT.648890710.1109/JIOT.2020.2966773.

[12] Changhao Chen, Xiaoxuan Lu, Johan Wahlstrom, A. Markham, and Niki Trigoni. Deep neural network based inertial odometry using low-cost inertial measurement units. volume 20, pages 1351–1364, 2021.

[13] Steven W. Chen, Guilherme V. Nardari, Elijah S. Lee, Chao Qu, Xu Liu, Roseli A. F. Romero, and Vijay Kumar. Sloam: Semantic lidar odometry and mapping for forest inventory, 2019. URL https://arxiv.org/abs/1912.12726.

[14] Travis J. Crayton and Benjamin Mason Meier. Autonomous vehicles: Developing a public health research agenda to frame the future of transportation policy. volume 6, pages 245–252, 2017. doi: https://doi.org/10.1016/j.jth.2017.04.004.

[15] Daniel. Cloudcompare-wiki: Distances computation, 11 2021. URL https://www.cloudcompare.org/doc/wiki/index.php?title=Distances_Computation.

[16] M. de Franceschi and D. Zardi. Evaluation of Cut-Off Frequency and Correction of Filter-Induced Phase Lag and Attenuation in Eddy Covariance Analysis of Turbulence Data. volume 108, pages 289–303, January 2003. doi: 10.1023/A:1024157310388.

[17] Hang Dong and Timothy D. Barfoot. Lighting-invariant visual odometry using lidar intensity imagery and pose interpolation. In *Field. Serv. Robot.: Results of the 8th Int. Conf.*, pages 327–342, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. ISBN 978-3-642-40686-7. doi: 10.1007/978-3-642-40686-7_22.

[18] Xin Dong, Ziyu Wang, Fangyuan Liu, Song Li, Fan Fei, Daochun Li, and Zhan Tu. Visual-inertial cross fusion: A fast and accurate state estimation framework for micro flapping wing rotors. volume 6, 2022. doi: 10.3390/drones6040090. URL https://www.mdpi.com/2504-446X/6/4/90.

[19] Zheng Fang, Shibo Zhao, and Shiguang Wen. A real-time and low-cost 3d slam system based on a continuously rotating 2d laser scanner. In *2017 IEEE 7th Annu. Int. Conf. CYBER Technol. Autom. Control, Intell. Syst.*, pages 454–459, 2017.

[20] Florent Feriol, Damien Vivet, and Yoko Watanabe. A review of environmental context detection for navigation based on multiple sensors. volume 20, 2020. doi: 10.3390/s20164532. URL https://www.mdpi.com/1424-8220/20/16/4532.

[21] Tully Foote. tf: The transform library. In *IEEE Int. Conf. Technol. Pract. Robot. Appl. (TePRA),*, Open-Source Software workshop, pages 1–6, April 2013. doi: 10.1109/TePRA.2013.6556373.

[22] Furgale, Rehder, and Siegwart. Kalibr: A generic multi-use toolbox for calibration of sensor systems, Accessed 2023. URL https://github.com/ethz-asl/kalibr.

[23] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286, 2013. doi: 10.1109/IROS.2013.6696514.

[24] gaowenliang. imu_utils - imu sensor utility library. GitHub repository, Accessed 2023. URL https://github.com/gaowenliang/imu_utils.

[25] Gazebo. Gazebo robot simulation made easy, 9 2013. URL http://gazebosim.org/.

[26] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. OpenVINS: A research platform for visual-inertial estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020. URL https://github.com/rpng/open_vins.

[27] Paul Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*. 03 2013.

[28] Pengfei Gui, Liqiong Tang, and Subhas Mukhopadhyay. Mems based imu for tilting measurement: Comparison of complementary and kalman filter based data fusion. In *2015 IEEE 10th Conf. Ind. Electron. Appl. (ICIEA)*, pages 2004–2009, 2015. doi: 10.1109/ICIEA.2015.7334442.

[29] M. Gunduz, U. Isikdag, and M. Basaraner. A review of recent research in indoor modelling and mapping. volume XLI-B4, pages 289–294, 2016. doi: 10.5194/isprs-archives-XLI-B4-289-2016. URL https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLI-B4/289/2016/.

[30] Dinh-Cuong Hoang, Todor Stoyanov, and Achim J. Lilienthal. High-quality instance-aware semantic 3d map using RGB-D camera. volume abs/1903.10782, 2019. URL http://arxiv.org/abs/1903.10782.

[31] Seungpyo Hong, Heedong Ko, and Jinwook Kim. Vicp: Velocity updating iterative closest point algorithm. In *2010 IEEE Int. Conf. Robot. Autom.*, pages 1893–1898, 2010. doi: 10.1109/ROBOT.2010.5509312.

[32] Satoshi Hoshino and Hideaki Yagi. Mobile robot localization using map based on cadastral data for autonomous navigation. volume 34, pages 111–120, 2022. doi: 10.20965/jrm.2022.p0111.

[33] Zheng Huai and Guoquan Huang. Robocentric visual-inertial odometry. volume 41, pages 667–689. SAGE Publications Sage UK: London, England, 2022.

[34] Bharat Joshi, Sharmin Rahman, Michail Kalaitzakis, Brennan Cain, James Johnson, Marios Xanthidis, Nare Karapetyan, Alan Hernandez, Alberto Quattrini Li, Nikolaos Vitzilaios, and Ioannis Rekleitis. Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7227–7233, 2019. doi: 10.1109/IROS40897.2019.8968049.

[35] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2342–2350, 2015.

[36] Anđela Jurić, Filip Kendeš, Ivan Marković, and Ivan Petrović. 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. doi: 10.23919/MIPRO52101.2021.9596721.

[37] Kiyosumi Kidono, Takeo Miyasaka, Akihiro Watanabe, Takashi Naito, and Jun Miura. Pedestrian recognition using high-definition lidar. In *2011 IEEE Intell. Veh. Symp. (IV)*, pages 405–410, 2011. doi: 10.1109/IVS.2011.5940433.

[38] Stefan Kohlbrecher, Oskar von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *2011 IEEE Int. Symp. Safety Security Rescue Robot.*, pages 155–160, 2011. doi: 10.1109/SSRR.2011.6106777.

[39] Kenji Koide, Jun Miura, and Emanuele Menegatti. A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement. volume 16, 2019. doi: 10.1177/1729881419841532. URL https://doi.org/10.1177/1729881419841532.

[40] Kenji Koide, Jun Miura, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. Interactive 3d graph slam for map correction. volume 6, pages 40–47, 2021. doi: 10.1109/LRA.2020.3028828.

[41] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. Using inertial sensors for position and orientation estimation. volume 11, pages 1–153. Now Publishers, 2017. doi: 10.1561/2000000094. URL https://doi.org/10.1561%2F2000000094.

[42] Yacouba KONE, Ni ZHU, Valérie Renaudin, and Miguel Ortiz. Machine Learning-Based Zero-Velocity Detection for Inertial Pedestrian Navigation. January 2020. doi: 10.1109/JSEN.2020.2999863.

[43] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *2011 IEEE Int. Conf. Robot. Autom.*, pages 3607–3613, 2011. doi: 10.1109/ICRA.2011.5979949.

[44] Zhiteng Li, Jiannan Zhao, Xiang Zhou, Shengxian Wei, Pei Li, and Feng Shuang. Rtsdm: A real-time semantic dense mapping system for uavs. volume 10, 2022. doi: 10.3390/machines10040285. URL https://www.mdpi.com/2075-1702/10/4/285.

[45] Xiaowei Luo, Fernanda Leite, and William O'Brien. Requirements for autonomous crane safety monitoring. 06 2011. doi: 10.1061/41182(416)41.

[46] Simon Lynen, Markus W. Achtelik, Stephan Weiss, Margarita Chli, and Roland Siegwart. A robust and modular multi-sensor fusion approach applied to mav navigation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3923–3929, 2013. doi: 10.1109/IROS.2013.6696917.

[47] Lovro Marković, Marin Kovač, Robert Milijas, Marko Car, and Stjepan Bogdan. Error state extended kalman filter multi-sensor fusion for unmanned aerial vehicle localization in gps and magnetometer denied indoor environments. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 184–190, 2022. doi: 10.1109/ICUAS54217.2022.9836124.

[48] Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. Lidar point clouds correction acquired from a moving car based on can-bus data. volume abs/1706.05886, 2017. URL http://arxiv.org/abs/1706.05886.

[49] Thomas Moore and Daniel Stouch. A generalized extended kalman filter implementation for the robot operating system. In Emanuele Menegatti, Nathan Michael, Karsten Berns, and Hiroaki Yamaguchi, editors, *Intelligent Autonomous Systems 13*, pages 335–348, Cham, 2016. Springer International Publishing. ISBN 978-3-319-08338-4.

[50] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *2007 IEEE International Conference on Robotics and Automation, ICRA'07*, Proceedings - IEEE International Conference on Robotics and Automation, pages 3565–3572, 2007. doi: 10.1109/ROBOT.2007.364024.

[51] OpenCV Documentation Contributors. OpenCV camera calibration and 3d reconstruction documentation. OpenCV Documentation, Accessed 2023. URL https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html.

[52] Gonzalo Perez Paina, David Gaydou, Javier Redolfi, Claudio Paz, and Luis Canali. Experimental comparison of kalman and complementary filter for attitude estimation. In *Argent. Symp. Technol. (ASTAt): Córdoba, Argentina*, 08 2011.

[53] François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing icp variants on real-world data sets. volume 34, pages 133–148, 04 2013. doi: 10.1007/s10514-013-9327-2.

[54] François Pomerleau, Francis Colas, and Roland Siegwart. A review of point cloud registration algorithms for mobile robotics. volume 4, pages 1–104, 2015. doi: 10.1561/2300000035. URL http://dx.doi.org/10.1561/2300000035.

[55] Tarik Pozderac, Jasmin Velagić, and Dinko Osmanković. 3d mapping based on fusion of 2d laser and imu data acquired by unmanned aerial vehicle. In *2019 6th Int. Conf. Cont. Decis. Inf. Technol. (CoDIT)*, pages 1533–1538, 2019. doi: 10.1109/CoDIT.2019.8820583.

[56] Tong Qin and Shaozu. A-loam - advanced implementation of loam, 2019. URL https://github.com/HKUST-Aerial-Robotics/A-LOAM.

[57] Tong Qin, Shaozu Cao, Jie Pan, Peiliang Li, and Shaojie Shen. Vins-fusion: An optimization-based multi-sensor state estimator. 2016. URL https://github.com/HKUST-Aerial-Robotics/VINS-Fusion.

[58] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. volume 34, pages 1004–1020, 2018. doi: 10.1109/TRO.2018.2853729.

[59] Tong Qin, Shaozu Cao, Jie Pan, and Shaojie Shen. A general optimization-based framework for global pose estimation with multiple sensors, 2019.

[60] Tong Qin, Jie Pan, Shaozu Cao, and Shaojie Shen. A general optimization-based framework for local odometry estimation with multiple sensors, 2019.

[61] Tobias Renzler, Michael Stolz, Markus Schratter, and Daniel Watzenig. Increased accuracy for fast moving lidars: Correction of distorted point clouds. In *2020 IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, pages 1–6, 2020. doi: 10.1109/I2MTC43012.2020.9128372.

[62] D. Roetenberg, H.J. Luinge, C.T.M. Baten, and P.H. Veltink. Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. volume 13, pages 395–405, 2005. doi: 10.1109/TNSRE.2005.847353.

[63] ROS. laser-assembler-0.3.0, 11 2011. URL http://library.isr.ist.utl.pt/docs/roswiki/laser_assembler(2d)0(2e)3(2e)0.html.

[64] ROS Wiki Contributors. Ros camera calibration wiki. ROS Wiki, Accessed 2023. URL http://wiki.ros.org/camera_calibration.

[65] Szymon M. Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proc. Third Int. Conf. 3-D Digit. Imaging. Model.*, pages 145–152, 2001.

[66] Nargess Sadeghzadeh-Nokhodberiz, Aydin Can, Rustam Stolkin, and Allahyar Montazeri. Dynamics-based modified fast simultaneous localization and mapping for unmanned aerial vehicles with joint inertial sensor bias and drift estimation. volume 9, pages 120247–120260, 2021. doi: 10.1109/ACCESS.2021.3106864.

[67] Sebastian Scherer, Joern Rehder, Supreeth Achar, Hugh Cover, Andrew Chambers, Stephen Nuske, and Sanjiv Singh. River mapping from a flying robot: State estimation, river detection, and obstacle mapping. volume 33, pages 189–214, 08 2012. doi: 10.1007/s10514-012-9293-0.

[68] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. volume 1, pages 187–210, 2018. doi: 10.1146/annurev-control-060117-105157.

[69] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. pages 4758–4765, 2018.

[70] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Rus Daniela. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pages 5135–5142. IEEE, 2020.

[71] João Paulo Silva do Monte Lima, Hideaki Uchiyama, and Rin-ichiro Taniguchi. End-to-end learning framework for imu-based 6-dof odometry. volume 19, 2019. doi: 10.3390/s19173777.

[72] Scott Sun, Dennis Melamed, and Kris Kitani. IDOL: inertial deep orientation-estimation and localization. volume abs/2102.04024, 2021.

[73] Li Tan and Jean Jiang. Chapter 6 - digital signal processing systems, basic filtering types, and digital filter realizations. In *Digital Signal Processing (Second Edition)*, pages 161–215. 2013. ISBN 978-0-12-415893-1. doi: https://doi.org/10.1016/B978-0-12-415893-1.00006-8.

[74] Baihui Tang and Sanxing Cao. A review of VSLAM technology applied in augmented reality. volume 782, mar 2020. doi: 10.1088/1757-899x/782/4/042014.

[75] Fulin Tang, Yihong Wu, Xiaohui Hou, and Haibin Ling. 3d mapping and 6d pose computation for real time augmented reality on cylindrical objects. volume 30, pages 2887–2899, 2020. doi: 10.1109/TCSVT.2019.2950449.

[76] Kyriaki Tychola, Ioannis Tsimperidis, and George Papakostas. On 3d reconstruction using rgb-d cameras. volume 2, pages 401–423, 08 2022. doi: 10.3390/digital2030022.

[77] Mahmood Ul Hassan, Dipankar Das, and Jun Miura. 3d mapping for a large crane using rotating 2d-lidar and imu attached to the crane boom. volume 11, pages 21104–21116, 2023. doi: 10.1109/ACCESS.2023.3250248.

[78] Roberto G. Valenti, Ivan Dryanovski, and Jizhong Xiao. Keeping a good attitude: A quaternion-based orientation filter for imus and margs. volume 15, pages 19302–19330, 2015. doi: 10.3390/s150819302. URL https://www.mdpi.com/1424-8220/15/8/19302.

[79] Jessica Van Brummelen, Marie O'Brien, Dominique Gruyer, and Homayoun Najjaran. Autonomous vehicle perception: The technology of today and tomorrow. volume 89, pages 384–406, 2018. doi: https://doi.org/10.1016/j.trc.2018.02.012.

[80] H. Wang, C. Wang, C. Chen, and L. Xie. F-loam : Fast lidar odometry and mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2021.

[81] Stephan Weiss and Roland Siegwart. Real-time metric state estimation for modular vision-inertial systems. pages 4531–4537, 06 2011. doi: 10.1109/ICRA.2011.5979982.

[82] wiki.ros. laser-assembler, 9 2013. URL http://wiki.ros.org/laser_assembler.

[83] Ryan W Wolcott and Ryan M Eustice. Robust lidar localization using multiresolution gaussian mixture maps for autonomous driving. volume 36, pages 292–319, 2017. doi: 10.1177/0278364917696568.

[84] Dong-Hoan Seo Won-Yeol Kim, Hong-Il Seo. Nine-axis imu-based extended inertial odometry neural network. volume 178, page 115075, 2021. doi: https://doi.org/10.1016/j.eswa.2021.115075.

[85] Hang Yan, Sachini Herath, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods. volume abs/1905.12853, 2019.

[86] Tao Yang, You Li, Cheng Zhao, Dexin Yao, Guanyin Chen, Li Sun, Tomas Krajnik, and Zhi Yan. 3d tof lidar in mobile robotics: A review. 2022. doi: 10.48550/ARXIV.2202.11025. URL https://arxiv.org/abs/2202.11025.

[87] Haoyang Ye, Yuying Chen, and Ming Liu. Tightly coupled 3d lidar inertial odometry and mapping. In *2019 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2019.

[88] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. volume 8, pages 58443–58469. Institute of Electrical and Electronics Engineers (IEEE), 3 2020. doi: 10.1109/access.2020.2983149.

[89] Ji Zhang and Singh. Low-drift and real-time lidar odometry and mapping. volume 41, pages 401–416, 2017.

[90] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Proceedings of Robotics: Science and Systems (RSS '14)*, pages 109–111. roboticsproceedings.org, July 2014. doi: https://doi.org/10.15607/rss.2014.x.007.

[91] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7244–7251, 2018. doi: 10.1109/IROS.2018.8593941.

[92] Keyuan Zhao, Qicai Zhou, Xiaolei Xiong, and Jiong Zhao. Active visual mapping system for digital operation environment of bridge crane. volume 93, 01 2022. doi: 10.1063/5.0067246.

[93] Zhihao Zhao, Wenquan Zhang, Jianfeng Gu, Junjie Yang, and Kai Huang. Lidar mapping optimization based on lightweight semantic segmentation. volume 4, pages 353–362, 2019. doi: 10.1109/TIV.2019.2919432.

[94] Åkerblom Svensson and J. Gullberg Carlsson. Analysis of comparative filter algorithm effect on an imu. 2021. URL http://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-54147.

[95] Krzysztof Ćwian, Michał R. Nowicki, Jan Wietrzykowski, and Piotr Skrzypczyński. Large-scale lidar slam with factor graph optimization on high-level geometric features. volume 21, 2021. doi: 10.3390/s21103445. URL https://www.mdpi.com/1424-8220/21/10/3445.

Appendix A

Publication List

- Mahmood Ul Hassan, D. Das and J. Miura, "3D Mapping for a Large Crane Using Rotating 2D-Lidar and IMU Attached to the Crane Boom," in IEEE Access, doi: 10.1109/ACCESS.2023.3250248, vol. 11, pp. 21104-21116, 2023.
- Mahmood Ul Hassan, and J. Miura, "Sensor Pose Estimation and 3D Mapping for Crane Operations Using Sensors Attached to the Crane Boom," in IEEE Access, doi: 10.1109/ACCESS.2023.3307197, vol. 11, pp. 90298-90308, 2023.
- Mahmood Ul Hassan and J. Miura, "Neural Network-Based Real-Time Odometry Using IMU for Crane System and Its Application to Large-Scale 3D Mapping," IEEE International Conference on Mechatronics and Automation (ICMA), Guilin, Guangxi, China, pp. 1062-1068, 2022.