IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 January 2023, accepted 22 February 2023, date of publication 28 February 2023, date of current version 6 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3250248

APPLIED RESEARCH

3D Mapping for a Large Crane Using Rotating 2D-Lidar and IMU Attached to the Crane Boom

MAHMOOD UL HASSAN[1], DIPANKAR DAS[2], AND JUN MIURA[1], (Member, IEEE)
[1]Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan
[2]Department of Information and Communication Engineering, University of Rajshahi, Rajshahi 6205, Bangladesh

Corresponding author: Mahmood Ul Hassan (mahmood.ul.hassan.vo@tut.jp)

This work was supported in part by Kobelco Construction Machinery Company Ltd., Japan.

ABSTRACT This paper 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. 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. 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 a small-sized 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.

INDEX TERMS 3D mapping, 2D lidar, IMU, pose graph optimization, complementary filter, large crane.

I. INTRODUCTION

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 [1], [2], [3]. 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 [4], [5], [6], [7] and drones [8], [9], [10], very few have been done for 3D mapping for construction cranes [11], [12].

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

Visual SLAM methods are popular in 3D mapping [13], [14], [15] but tend to be weak under varying lighting conditions or feature-scarce environments. 3D lidar-based mapping is also popular [8], [16], [17], [18], as 3D scans provide rich structural information even under poor lighting conditions outdoors. However, a limited vertical field of 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 which adopt scan matching-based relative pose estimation [9], [19], [20], [21], [22]. The lidar-IMU fusion approach is effective in compensating the lidar motion during the rotation [10], [19], [20], [23]. 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

The associate editor coordinating the review of this manuscript and approving it for publication was Guilin Yang.

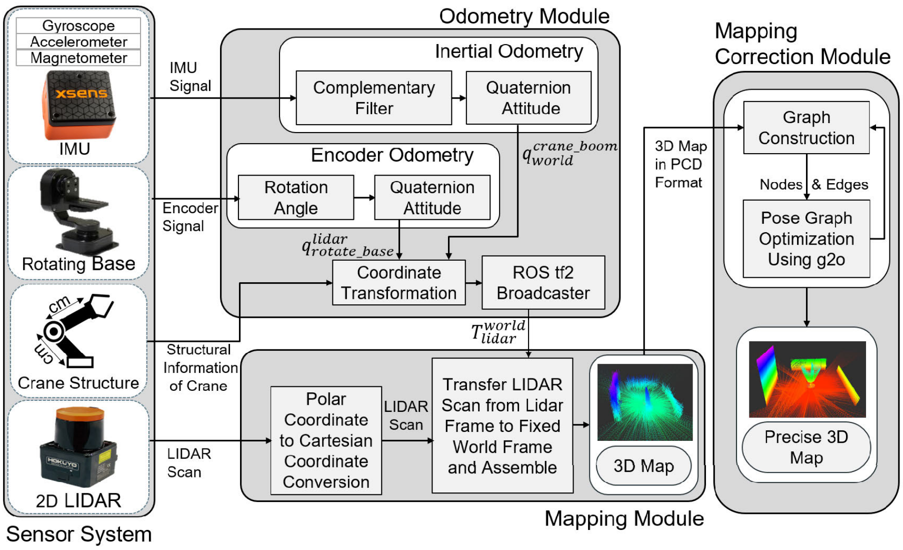

FIGURE 1. Block diagram of proposed method showing all three modules: Inertial odometry module, mapping module, and mapping correction module.

scan reliably. 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 [10], [19], [20], [23] 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) [24], [25]. However, usual PGO methods which rely on scan matching are not appropriate for our case because we have little overlaps between scans when generated by a slowly-rotating 2D lidar.

This paper describes a novel method for large-scale 3D mapping for construction cranes. Considering the challenges mentioned above, we use a slowly-rotating 2D lidar and an IMU as base components and attach them to the crane boom. As the crane boom moves arbitrarily during operations, we estimate the lidar pose for each scan using the IMU using a complementary filter [26] with moving average filtering, which is more robust to severe vibration than Kalman filter-based methods [27], [28], [29], [30]. 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 [19], [20] fail.

Fig. 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 paper is organized as follows. We analyze the relevant related work in Section II. Section III explains lidar pose estimation using a complementary filter. Section IV describes the 3D mapping method, and Section V explains the mapping correction using the pose graph optimization with planar environmental constraints. Sections VI and VII describe experimental results in a simulation and a small-sized real environment, respectively. Section VIII concludes the paper.

II. RELATED WORK

A. VISION-BASED MAPPING

Visual SLAM (vSLAM) [13], [14], [15] 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.

B. 3D LIDAR-BASED MAPPING

Many 3D lidar-based mapping methods have been developed and effectively utilized in applications such as autonomous driving [16], [17] and drone-based aerial mapping [8], [18]. 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.

C. 2D LIDAR-BASED 3D MAPPING

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 [9], [19], [20], [21], [22]. 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 [21], the rotational speed is 30 rpm or 180°/ 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.

D. LIDAR-IMU FUSION

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 [27]. Many lidar-IMU mapping methods, such as LOAM [19], [20] and its extensions [16], [18], [25], [31], 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 [26] is used for orientation estimation using an IMU and exhibits better estimation accuracy and robustness than the Kalman filter under high vibration [27], [28], [29], [30]. Combined with the structural information of a crane, the complementary filter can be used for sensor pose estimation.

E. 3D LIDAR-BASED MAPPING CORRECTION

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 [32], [33], [34]. In 3D lidar-based PGO [24], [25], 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 [35], [36].

III. ODOMETRY MODULE

The odometry estimates the lidar's position in real-time. As demonstrated in Fig. 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.

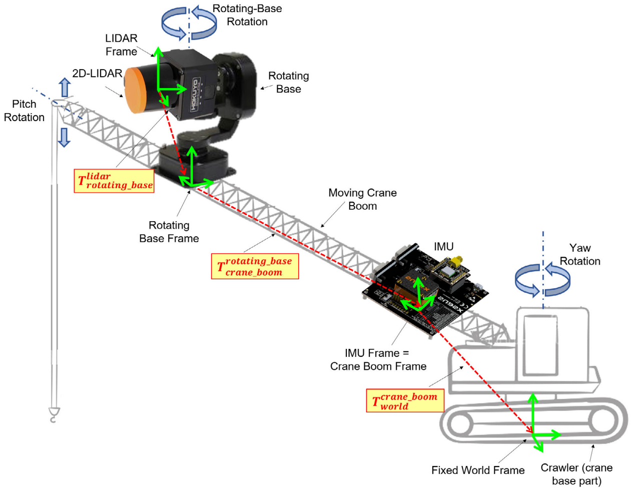

FIGURE 2. Sensor system attached on crane boom and the relationship between different coordinate frames is also shown.

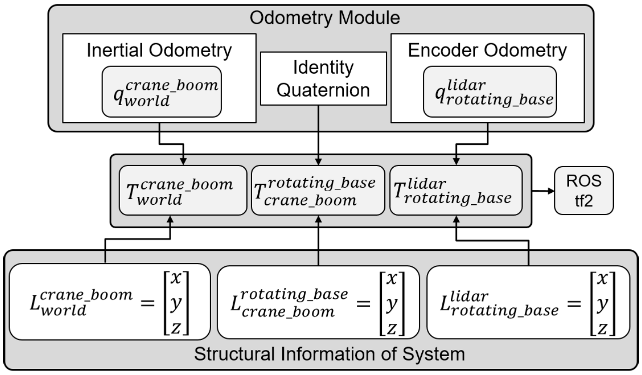
Odometry Module
Inertial Odometry
$q_{world}^{crane\_boom}$
Identity
Quaternion
Encoder Odometry
$q_{rotating\_base}^{lidar}$
$T_{world}^{crane\_boom}$
$T_{crane\_boom}^{rotating\_base}$
$T_{rotating\_base}^{lidar}$
ROS
tf2
$L_{world}^{crane\_boom} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$
$L_{crane\_boom}^{rotating\_base}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

FIGURE 3. Coordinate transformation of all frames in our system.

A. COORDINATE TRANSFORMATION

Fig. 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_{lidar}^{world} from the lidar frame to the fixed world frame, we can chain the transformation between the coordinate frames, as given below:

$$T_{lidar}^{world} = T_{crane\_boom}^{world} \; T_{rotating\_base}^{crane\_boom} \; T_{lidar}^{rotating\_base}, \qquad (1)$$

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):

- $T_{crane\_boom}^{world}$: 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 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 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 shows that the quaternion $q_{rotating\_base}^{lidar}$ and vector $L_{rotating\_base}^{lidar}$ given the transformation between the rotating base and lidar frame.

B. 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. 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. 1 and 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 [26]. 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 [34] to obtain the stabilized accelerometer reading a_k using:

$$a_k = \frac{1}{N} \sum_{i=k-N}^{k} \tilde{a}_k \tag{2}$$

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 [36]:

$$\|^b m\| = \sqrt{m_x^2 + m_y^2 + m_z^2} \tag{3}$$

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 ($\|^b m\| = 1$) [37]. 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 will discuss our approach to computing the crane boom rotation using a quaternion-based complementary filter [26].

1) 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:

$$q_{world}^{crane\_boom} = {}^b_w q, \tag{4}$$

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:

$${}^b_w\dot{q}_{w,t_k} = -\frac{1}{2} {}^b\omega_{q,t_k} \otimes {}^b_w q_{t_{k-1}}, \tag{5}$$

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

$${}^b_w\dot{q}_{w,t_k} = \Omega({}^b\omega_{t_k})\, {}^b_w q_{t_{k-1}}, \tag{6}$$

$$\Omega({}^b\omega_{t_k}) = \begin{bmatrix} 0 & {}^b\omega_{t_k}{}^T \\ {}^b\omega_{t_k} & -[{}^b\omega_{t_k} \times] \end{bmatrix}, \tag{7}$$

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:

$${}^b_w q_{w,t_k} = {}^b_w q_{t_{k-1}} + {}^b_w\dot{q}_{w,t_k} \Delta t. \tag{8}$$

2) 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. 8 as given below

$$R({}^w_b q_w)\, {}^b a = {}^w g_p. \tag{9}$$

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:

$$R(\Delta q_{acc})\,{}^{w}g = {}^{w}g_p. \tag{10}$$

If we write ${}^{w}g$ and ${}^{w}g_p$ in vector form then Eq. 10 become as

$$R(\Delta q_{acc})\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}g_x\\g_y\\g_z\end{bmatrix}. \tag{11}$$

By simplifying the Eq. 11, we can get

$$\Delta q_{acc} = \begin{bmatrix}\sqrt{\frac{g_z+1}{2}} & -\frac{g_y}{\sqrt{2(g_z+1)}} & \frac{g_x}{\sqrt{2(g_z+1)}} & 0\end{bmatrix}^T, \tag{12}$$

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 Δq_{0acc} between q_I and Δq_{acc}. If Δq_{0acc} is greater than a predefined threshold value ϵ ($\Delta q_{0acc} > \epsilon$), linear interpolation(LERP) is used as given below as follows [26]:

$$\overline{\Delta q_{acc}} = (1-\alpha)q_I + \alpha\Delta q_{acc}, \tag{13}$$

where α is the gain that represents the cut-off frequency of the filter [38]. By normalizing the Eq. 13 we get

$$\widehat{\Delta q_{acc}} = \frac{\overline{\Delta q_{acc}}}{||\overline{\Delta q_{acc}}||}. \tag{14}$$

If $\Delta q_{0acc} < \epsilon$, spherical linear interpolation (SLERP) is used as given below as follows [26]:

$$\widehat{\Delta q_{acc}} = \frac{sin([1-\alpha]\Omega)}{sin\Omega}q_I + \frac{sin(\alpha\Omega)}{sin\Omega}\Delta q_{acc}. \tag{15}$$

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:

$${}^{b}_{w}q' = {}^{b}_{w}q_{\omega} \otimes \widehat{\Delta q_{acc}} \tag{16}$$

3) 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. 16 as follows:

$$R({}^{w}_{b}q')\,{}^{b}m = l. \tag{17}$$

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:

$$R^T(\Delta q_{mag})\begin{bmatrix}l_x\\l_y\\l_z\end{bmatrix} = \begin{bmatrix}\sqrt{l_x^2+l_y^2}\\0\\l_z\end{bmatrix}, \tag{18}$$

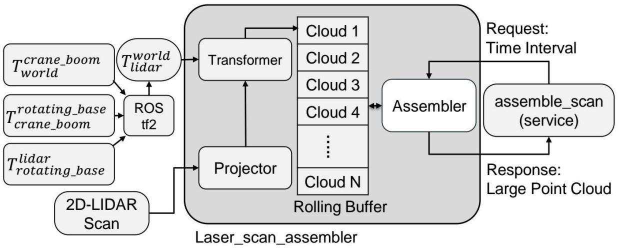

FIGURE 4. Block diagram of mapping module.

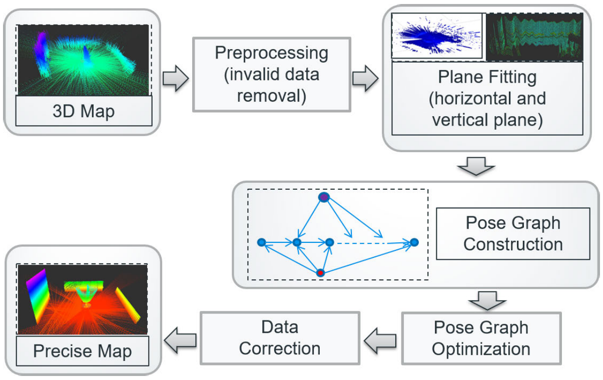
3D Map
Preprocessing (invalid data removal)
Plane Fitting (horizontal and vertical plane)
Pose Graph Construction
Pose Graph Optimization
Data Correction
Precise Map

FIGURE 5. Block diagram of mapping correction module.

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:

$$\Delta q_{mag} = \begin{bmatrix}\Delta q_{0mag} & 0 & 0 & \Delta q_{3mag}\end{bmatrix} \tag{19}$$

By solving the system of equation Eq. 18 by substituting Δq_{mag}, we will get following equation as follows [26]:

$$\Delta q_{mag} = \begin{bmatrix}\frac{\sqrt{\Gamma + l_x\sqrt{\Gamma}}}{\sqrt{2\Gamma}} & 0 & 0 & \frac{l_y}{\sqrt{2(\Gamma + l_x\sqrt{\Gamma})}}\end{bmatrix}. \tag{20}$$

where

$$\Gamma = l_x^2 + l_y^2 \tag{21}$$

To minimize the noise of the magnetometer, same LERF and SLERP is used as given in Eq. 13 and Eq. 15 and final quaternion is obtained as follows [26]:

$${}^{b}_{w}q = {}^{b}_{w}q' \otimes \widehat{\Delta q_{mag}} \tag{22}$$

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

C. TRANSLATION PARAMETERS

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

The tf2 broadcaster [39] 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.

IV. MAPPING MODULE

We build a 3D map during the sensor system's motion using a laser-assembler [40], [41]. The laser-assembler assembles individual laser scan lines of 2D lidar into a composite 3D point cloud. Fig. 4 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 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

$$P_{world} = T^{world}_{lidar} P_{lidar} \tag{23}$$

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).

V. MAPPING CORRECTION MODULE

Fig. 5 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 [32] 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.

A. 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 original point cloud data are divided into frames, known as scan lines. The sensor poses vary for each frame.

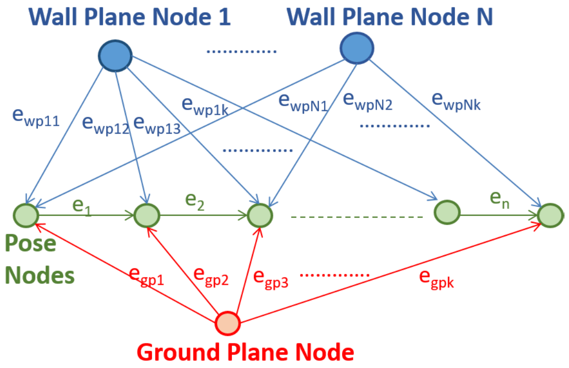

FIGURE 6. Pose graph showing pose nodes, plane nodes and edges.

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.

B. POSE GRAPH CONSTRUCTION

The pose graph presented in Fig. 6 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 are defined by the different plane constraints. The ground plane constraint is added as an external node (red color nodes in Fig. 6); 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. 6). 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:

$$E = \sum_{j=1}^{K} d_{\perp}{}_{j}^{2}, \tag{24}$$

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.

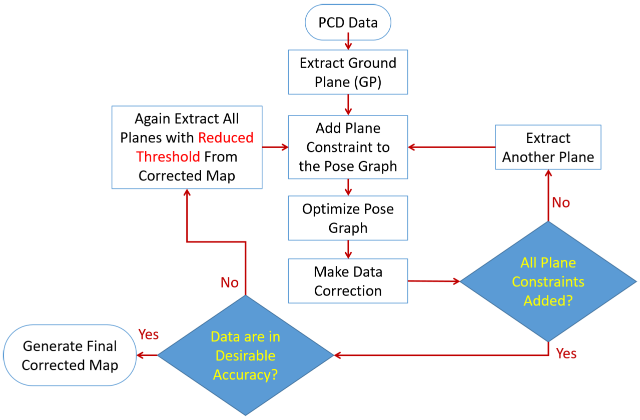
PCD Data
Extract Ground Plane (GP)
Add Plane Constraint to the Pose Graph
Again Extract All Planes with Reduced Threshold From Corrected Map
Extract Another Plane
Optimize Pose Graph
Make Data Correction
No
All Plane Constraints Added?
No
Data are in Desirable Accuracy?
Yes
Yes
Generate Final Corrected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

FIGURE 7. Successive and iterative approach of pose optimization.

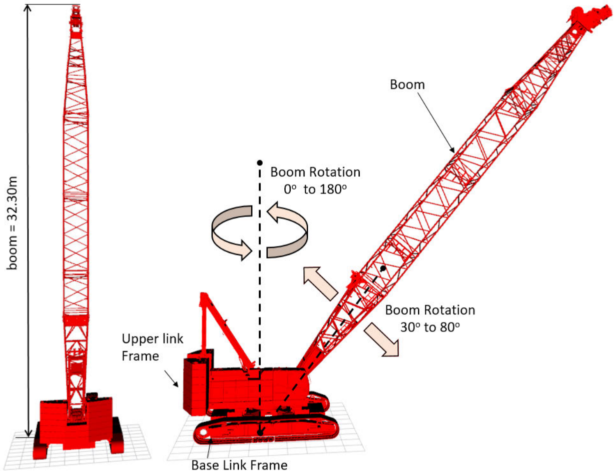

FIGURE 8. Crane model designed in Gazebo simulation environment. The boom of crane has two types of rotations.

C. POSE GRAPH OPTIMIZATION AND MAP CORRECTION

The general graph optimization (g2o) [32] is used to optimize the pose graph. In the optimization principle, we used the sum of squared errors calculated using Eq. (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.

D. SUCCESSIVE AND ITERATIVE DATA CORRECTION

We perform our data correction process successively and iteratively as shown in Fig. 7. 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 from the corrected data and repeat the optimization process until a desirable accuracy has been reached.

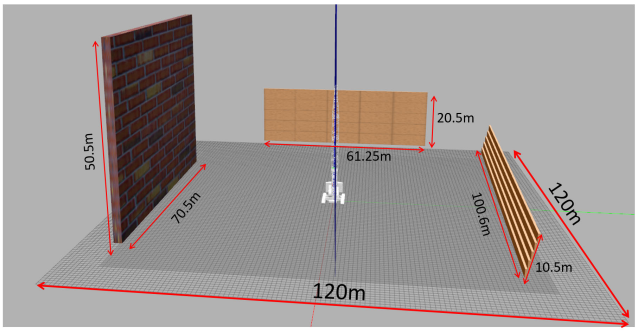

FIGURE 9. The featureless environment consisting of only walls.

TABLE 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

VI. SIMULATION RESULTS

We first evaluated the proposed method in a simulation environment. For simulation, we used Gazebo simulator [42] and ROS environments. In Gazebo, a crane robotic model is designed as shown in Fig. 8. The size of the crane is shown in the Fig. 8. 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. The supplementary video of the results is available at https://youtu.be/UH8QB7AKUk4. The description of each environment and results are provided below.

A. SIMULATION ENVIRONMENT-1

The first simulated environment is an open-sky featureless environment comprising simple three walls, as shown in Fig. 9. 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 1) are added to the IMU, as shown in Fig. 10. We built 3D maps during an arbitrary

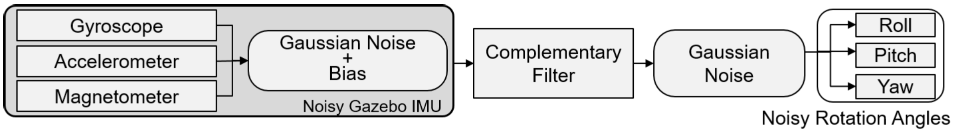

FIGURE 10. IMU designed in Gazebo simulator by adding noise and bias.

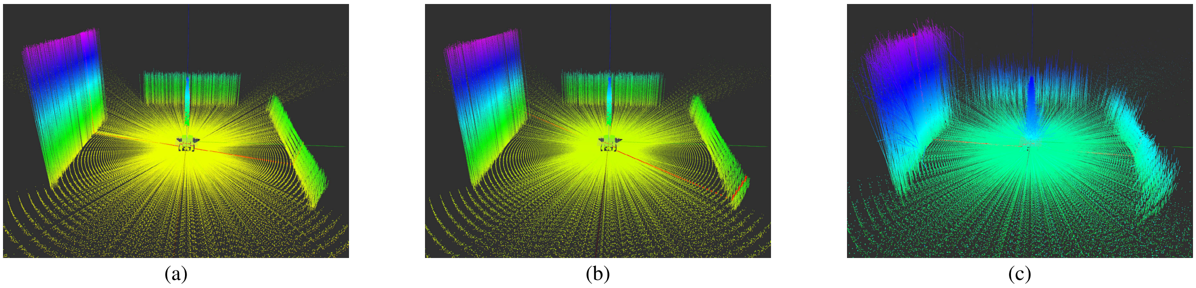

FIGURE 11. 3D point cloud map built using three different levels of IMU noise. The color indicates the height of each point. (a) 3D map built using IMU noise level 1. (b) 3D map built using IMU noise level 2. (c) 3D map built using IMU noise level 3. The map is distorted because of the high noise.

TABLE 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

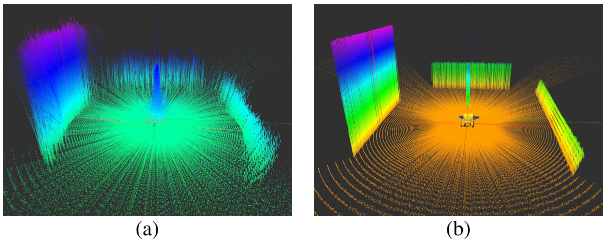

FIGURE 12. Result of mapping correction method. The color represents the height of each point. (a) Before optimization the distorted point cloud. (b) After optimization the corrected point cloud.

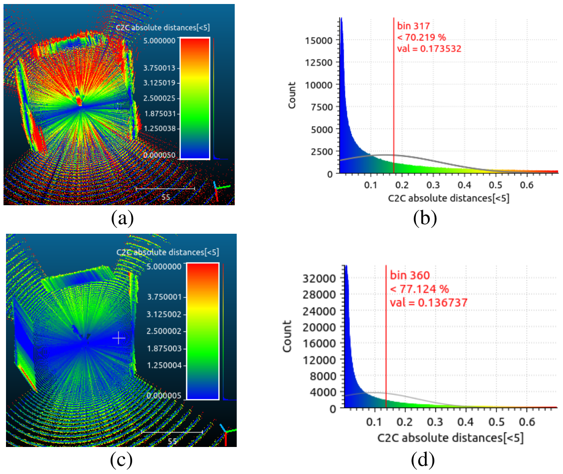

FIGURE 13. Analysis the 3D mapping results in simulation environment-1 before and after mapping correction. The color represents the error (cloud-to-cloud distance between ground truth and point cloud obtained by the proposed method) (a) Point-to-point distances before mapping correction. (b) Distribution 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.

motion of the crane boom. Fig. 11 shows the 3D 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. 11 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

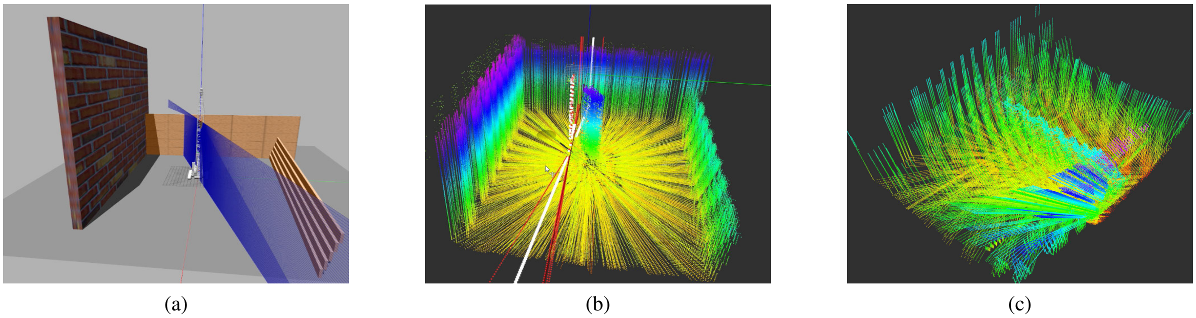

FIGURE 14. The results of LOAM implemented in simulation environment-1. (a) Reduced area of simulation environment-1 by bringing walls closer to each other. (b) LOAM results during static crane boom. The color represents the height of each point. (c) LOAM results during moving crane boom. The color represents the height of each point.

using cloud compare [43]. Fig.13 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. 13(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. 13(b) shows the distribution fitting graph of point-to-point distances. After mapping correction, most points in Fig. 13(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. 13(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 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.

We compare the proposed method with LOAM [19], 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 [19]. Because of LOAM's limited range for mapping, the area of simulation environment-1 (Fig. 9) is reduced by bringing walls closer to each other, as shown in Fig. 14(a). Fig. 14(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.

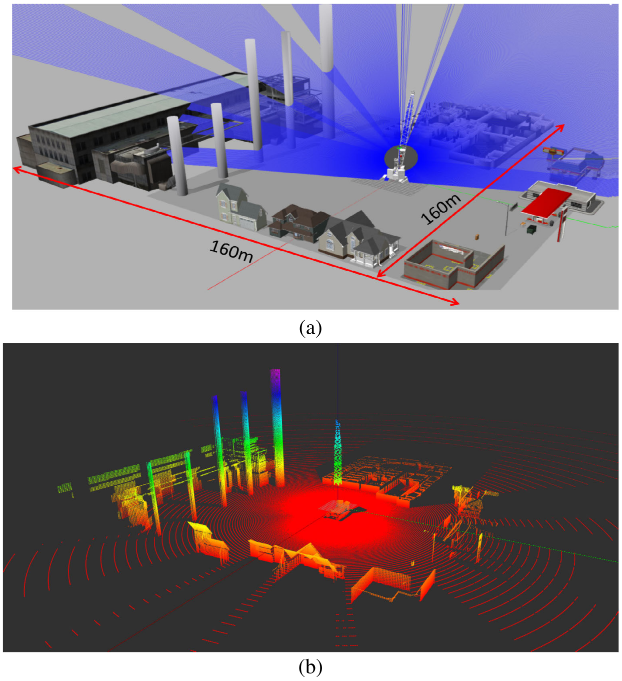

FIGURE 15. Proposed method implemented in simulation environment-2. (a) Complex construction environment. (b) 3D point cloud map for complex construction environment. The color represents the height of each point.

B. SIMULATION ENVIRONMENT-2

The proposed method was also tested in another complex construction site environment shown in Fig. 15(a). The area of the construction site is 160 m by 160 m in size which is scanned to build the 3D map. Fig. 15(b) shows the 3D map after the mapping correction. In Fig. 15(b) the color represents the height of each point. Fig. 17(a) shows the point-to-point distance between the generated and the ground

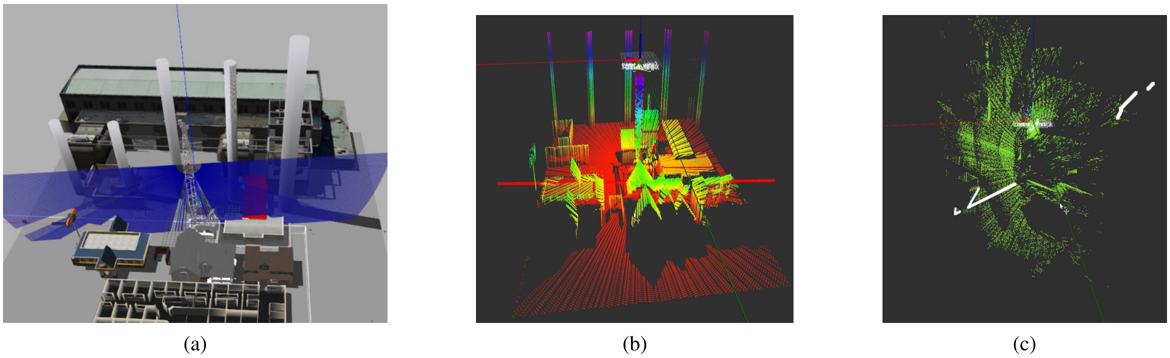

FIGURE 16. LOAM method implemented in simulation environment-2. (a) Reduced area of simulation environment-2 by bringing building closer to each other (b)LOAM results during static crane boom. The color represents the height of each point. (c)LOAM results during moving crane boom.

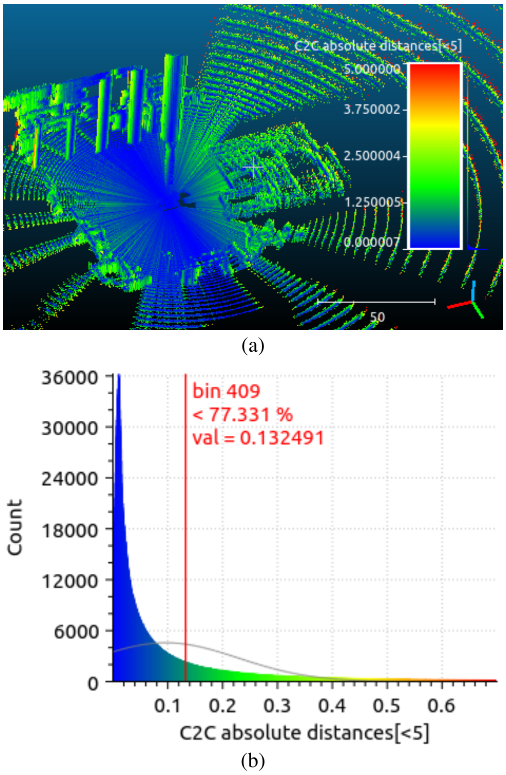

FIGURE 17. Analysis of simulation environment-2. The color represents the error (cloud-to-cloud distance between ground truth and point cloud obtained by the proposed method) (a) Point-to-point distances. (b) Distribution fitting of point-to-point distances.

truth map, and Fig. 17(b) shows the distribution of point-to-point distances. The fact that the majority of the points in the 3D map 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.

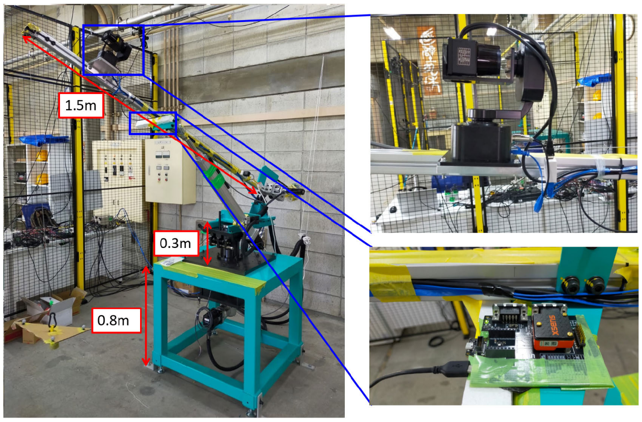

FIGURE 18. Crane model and sensor system used for experiment are shown. The sensor system is attached to the crane boom.

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. 16(a)). The results for a static and moving crane boom are shown in Fig. 16(b) and Fig. 16(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.

VII. EXPERIMENTAL RESULTS

The proposed method was also evaluated in real-world experiments on a crane model (construction machine) as shown in Fig. 18. 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. For constructing a 3D map, the rotating base of the 2D lidar rotates at

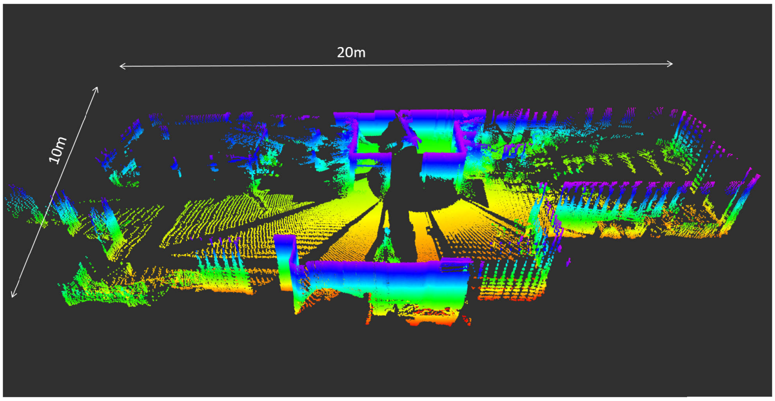

FIGURE 19. 3D point cloud map of environment built during experiment when crane boom is in continuous motion. The color represents the height of each point.

a 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. 19 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. 20(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. 20(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 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. 21, 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.

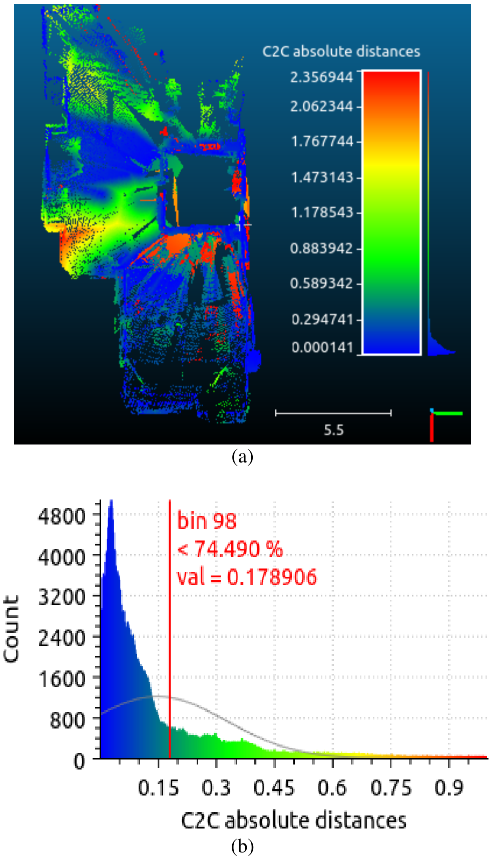

FIGURE 20. Result analysis of real world experiment. The color represents the error (cloud-to-cloud distance between ground truth and point cloud obtained by the proposed method) (a) point-to-point distance of two 3D maps (b) Distribution fitting of point-to-point distance.

TABLE 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

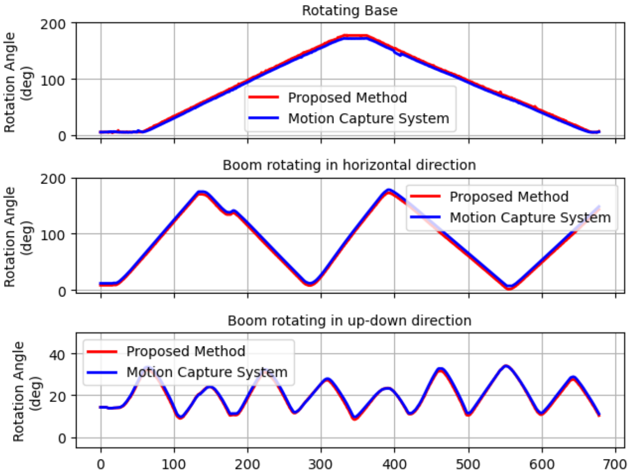

FIGURE 21. Compare of oddomatry and optimization with motion capture system.

In real-world experiments, due to the unavailability of a large crane on campus, we evaluated the proposed method using a small-scale crane model in a small space. However, the proposed method can be applied without any modification to the real-world environment using a large crane, as verified in the simulation environment for large-scale mapping.

VIII. CONCLUSION

This paper 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.

ACKNOWLEDGMENT

This work is a part of the collaboration with Kobelco Construction Machinery Company Ltd., Japan.

REFERENCES

[1] F. Feriol, D. Vivet, and Y. Watanabe, "A review of environmental context detection for navigation based on multiple sensors," *Sensors*, vol. 20, no. 16, p. 4532, Aug. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/16/4532

[2] T. Yang, Y. Li, C. Zhao, D. Yao, G. Chen, L. Sun, T. Krajnik, and Z. Yan, "3D ToF LiDAR in mobile robotics: A review," 2022, *arXiv:2202.11025*.

[3] M. Gunduz, U. Isikdag, and M. Basaraner, "A review of recent research in indoor modelling and mapping," *Int. Arch. Photo. Remote Sens. Spat. Inf. Sci.*, vol. 41, pp. 289–294, Jul. 2016. [Online]. Available: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLI-B4/289/2016/

[4] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[5] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 384–406, Apr. 2018.

[6] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 1, pp. 187–210, May 2018.

[7] T. J. Crayton and B. M. Meier, "Autonomous vehicles: Developing a public health research agenda to frame the future of transportation policy," *J. Transp. Health*, vol. 6, pp. 245–252, Sep. 2017.

[8] Z. Li, J. Zhao, X. Zhou, S. Wei, P. Li, and F. Shuang, "RTSDM: A real-time semantic dense mapping system for UAVs," *Machines*, vol. 10, no. 4, p. 285, Apr. 2022. [Online]. Available: https://www.mdpi.com/2075-1702/10/4/285

[9] T. Pozderac, J. Velagic, and D. Osmankovic, "3D mapping based on fusion of 2D laser and IMU data acquired by unmanned aerial vehicle," in *Proc. 6th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Apr. 2019, pp. 1533–1538.

[10] N. Sadeghzadeh-Nokhodberiz, A. Can, R. Stolkin, and A. Montazeri, "Dynamics-based modified fast simultaneous localization and mapping for unmanned aerial vehicles with joint inertial sensor bias and drift estimation," *IEEE Access*, vol. 9, pp. 120247–120260, 2021.

[11] K. Zhao, Q. Zhou, X. Xiong, and J. Zhao, "Active visual mapping system for digital operation environment of bridge crane," *Rev. Sci. Instrum.*, vol. 93, no. 1, Jan. 2022, Art. no. 015008.

[12] X. Luo, F. Leite, and W. O'Brien, "Requirements for autonomous crane safety monitoring," in *Proc. Comput. Civil Eng.*, Jun. 2011, pp. 331–338.

[13] D.-C. Hoang, T. Stoyanov, and A. J. Lilienthal, "High-quality instance-aware semantic 3D map using RGB-D camera," 2019, *arXiv:1903.10782*.

[14] K. Tychola, I. Tsimperidis, and G. Papakostas, "On 3D reconstruction using RGB-D cameras," *Digital*, vol. 2, pp. 401–423, Aug. 2022.

[15] F. Tang, Y. Wu, X. Hou, and H. Ling, "3D mapping and 6D pose computation for real time augmented reality on cylindrical objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 2887–2899, Sep. 2020.

[16] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-LOAM: Fast LiDAR odometry and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 4390–4396.

[17] T. Qin and Shaozu. (2019). *A-Loam—Advanced Implementation of Loam*. [Online]. Available: https://github.com/HKUST-Aerial-Robotics/A-LOAM

[18] S. W. Chen, G. V. Nardari, E. S. Lee, C. Qu, X. Liu, R. A. F. Romero, and V. Kumar, "SLOAM: Semantic LiDAR odometry and mapping for forest inventory," 2019, *arXiv:1912.12726*.

[19] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot., Sci. Syst. (RSS)*, Jul. 2014, pp. 109–111.

[20] J. Zhang and S. Singh, "Low-drift and real-time LiDAR odometry and mapping," *Auton. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[21] M. Bosse and R. Zlot, "Continuous 3D scan-matching with a spinning 2D laser," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 4312–4319.

[22] Z. Fang, S. Zhao, and S. Wen, "A real-time and low-cost 3D SLAM system based on a continuously rotating 2D laser scanner," in *Proc. IEEE 7th Annu. Int. Conf. CYBER Technol. Autom., Control, Intell. Syst. (CYBER)*, Jul. 2017, pp. 454–459.

[23] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D LiDAR inertial odometry and mapping," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3144–3150.

[24] J. Byun, K.-I. Na, B.-S. Seo, and M. Roh, "Drivable road detection with 3D point clouds based on the MRF for intelligent vehicle," *Spring. Tract. Adv. Robot.*, vol. 105, pp. 49–60, Jan. 2015.

[25] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.

[26] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs," *Sensors*, vol. 15, no. 8, pp. 19302–19330, 2015. [Online]. Available: https://www.mdpi.com/1424-8220/15/8/19302

[27] X. Dong, Z. Wang, F. Liu, S. Li, F. Fei, D. Li, and Z. Tu, "Visual-inertial cross fusion: A fast and accurate state estimation framework for micro flapping wing rotors," *Drones*, vol. 6, no. 4, p. 90, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/4/90

[28] G. P. Paina, D. Gaydou, J. Redolfi, C. Paz, and L. Canali, "Experimental comparison of Kalman and complementary filter for attitude estimation," in *Proc. Argent. Symp. Technol. (ASTAt)*, Córdoba, Argentina, Aug. 2011, pp. 1–11.

[29] P. Gui, L. Tang, and S. Mukhopadhyay, "MEMS based IMU for tilting measurement: Comparison of complementary and Kalman filter based data fusion," in *Proc. IEEE 10th Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2015, pp. 2004–2009.

[30] S. Vensson and J. G. Carlsson. (2021). *Analysis of Comparative Filter Algorithm Effect on an IMU*. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-54147

[31] Z. Zhao, W. Zhang, J. Gu, J. Yang, and K. Huang, "LiDAR mapping optimization based on lightweight semantic segmentation," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 3, pp. 353–362, Sep. 2019.

[32] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2O: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jan. 2011, pp. 3607–3613.

[33] K. Ćwian, M. R. Nowicki, J. Wietrzykowski, and P. Skrzypczyński, "Large-scale LiDAR SLAM with factor graph optimization on high-level geometric features," *Sensors*, vol. 21, no. 10, p. 3445, May 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/10/3445

[34] A. Juric, F. Kendes, I. Markovic, and I. Petrovic, "A comparison of graph optimization approaches for pose estimation in SLAM," in *Proc. 44th Int. Conv. Inf., Commun. Electron. Technol. (MIPRO)*, 2021, pp. 1113–1118.

[35] K. Koide, J. Miura, M. Yokozuka, S. Oishi, and A. Banno, "Interactive 3D graph SLAM for map correction," *IEEE Robot. Autom. Lett.*, vol. 6, no. 1, pp. 40–47, Jan. 2021.

[36] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LiDAR-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, 2019, Art. no. 1729881419841532, doi: 10.1177/1729881419841532.

[37] 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," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 3, pp. 395–405, Sep. 2005.

[38] 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," *Boundary-Layer Meteorol.*, vol. 108, no. 2, pp. 289–303, Aug. 2003.

[39] T. Foote, "Tf: The transform library," in *Proc. IEEE Conf. Technol. Pract. Robot Appl. (TePRA)*, Apr. 2013, pp. 1–6.

[40] ROS. (Nov. 2011). *Laser-Assembler-0.3.0*. [Online]. Available: http://library.isr.ist.utl.pt/docs/roswiki/laser_assembler(2d)0(2e)3(2e)0.html

[41] (Sep. 2013). *Laser-Assembler*. [Online]. Available: http://wiki.ros.org/laser_assembler

[42] Gazebo. (Sep. 2013). *Gazebo Robot Simulation Made Easy*. [Online]. Available: http://gazebosim.org/

[43] Daniel. (Nov. 2021). *Cloudcompare-Wiki: Distances Computation*. [Online]. Available: https://www.cloudcompare.org/doc/wiki/index.php?title=Distances_Computation

MAHMOOD UL HASSAN received the B.Eng. degree in industrial electronic engineering from the NED University of Engineering and Technology and the M.S. degree from the Department of Instrument Science and Engineering, Shanghai Jiao Tong University. He is currently pursuing the Ph.D. degree with the Active Intelligent Systems Laboratory, Toyohashi University of Technology, Toyohashi, Aichi, Japan.

DIPANKAR DAS received the B.Sc. and M.Sc. degrees in computer science and technology from the University of Rajshahi, Rajshahi, Bangladesh, in 1996 and 1997, respectively, and the Ph.D. degree in science and engineering from Saitama University, Saitama, Japan, in 2010. He worked with the Active Intelligent Systems Laboratory (AISL), Toyohashi University of Technology, Toyohashi, Aichi, Japan. He is currently with the Department of Information and Communication Engineering, University of Rajshahi. His research interests include object recognition and human–computer interaction.

JUN MIURA (Member, IEEE) received the B.Eng. degree in mechanical engineering and the M.Eng. and Dr.Eng. degrees in information engineering from The University of Tokyo, Tokyo, Japan, in 1984, 1986, and 1989, respectively. In 1989, he joined the Department of Computer-Controlled Mechanical Systems, Osaka University, Suita, Japan. From March 1994 to February 1995, he was a Visiting Scientist with the Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. Since April 2007, he has been a Professor with the Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi, Japan.