IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 August 2023, accepted 17 August 2023, date of publication 21 August 2023, date of current version 29 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3307197

APPLIED RESEARCH

Sensor Pose Estimation and 3D Mapping for Crane Operations Using Sensors Attached to the Crane Boom

MAHMOOD UL HASSAN AND JUN MIURA, (Member, IEEE)
Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan

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

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

ABSTRACT This paper 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.

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

I. INTRODUCTION

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 autonomous driving, service robotics, agriculture, augmented reality, and construction [1], [2], [3]. 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 [4], [5], [6], [7] and drones [8], [9], [10], there is a noticeable gap in studies focusing on 3D mapping for construction cranes [11], [12]. 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.

The research conducted in [13] demonstrated that the integration of additional sensors using the Extended Kalman Filter (EKF) improves the accuracy of odometry estimation in robot systems. This paper also aims to 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 [14] proposed fusion algorithms using multiple IMUs to enhance pedestrian navigation performance. They found that the accuracy of

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

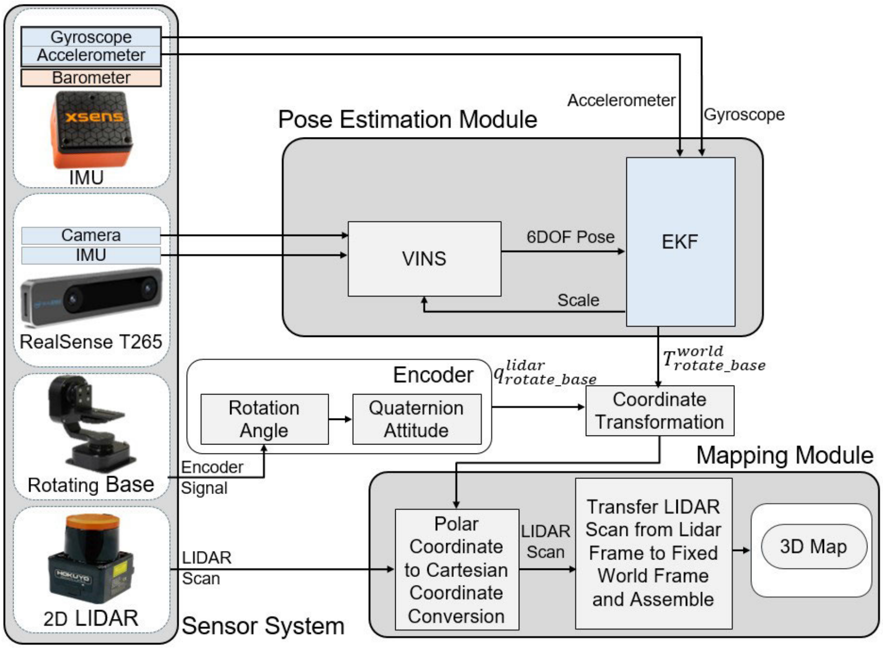


FIGURE 1. Over all block diagram of proposed method.

pose estimation is directly related to the number of IMU sensors used. In [15], an experimental comparison of various Visual-Inertial Navigation System (VINS) algorithms in the underwater domain was conducted. The study revealed that while VINS-Mono [16] demonstrated excellent performance, but its scale estimation was consistently inaccurate due to the use of monocular vision. The limitation of VINS-MONO [16] 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 [15] confirmed that incorporating IMU measurements significantly improved performance compared to pure Visual Odometry, extending the findings reported in [15]. The improved performance of IMU integration was observed across diverse underwater environments [15].

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 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 [16], VINS-Fusion [17], [18], [19], Multi-state Constraint Kalman Filter (MSCKF) algorithm [20], Robocentric visual-inertial odometry (R-VIO) [21]. 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 scan-lines, enabling the construction of an accurate and comprehensive 3D map.

II. 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 in Fig. 1, the proposed method consists on two main modules: Pose estimation module, and mapping module. In pose estimation module, 6 Dof pose is estimated using EKF which is used to fuse the pose estimated by VINS [16] and measurements from acceleromter, and gyroscope 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.

III. 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 paper, 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.

A. VINS-MONO

VINS-MONO [16], 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.

B. VINS-FUSION

VINS-Fusion [17], [18], [19], an optimization-based multisensor 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.

C. MSCKF

The MSCKF algorithm [20], 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.

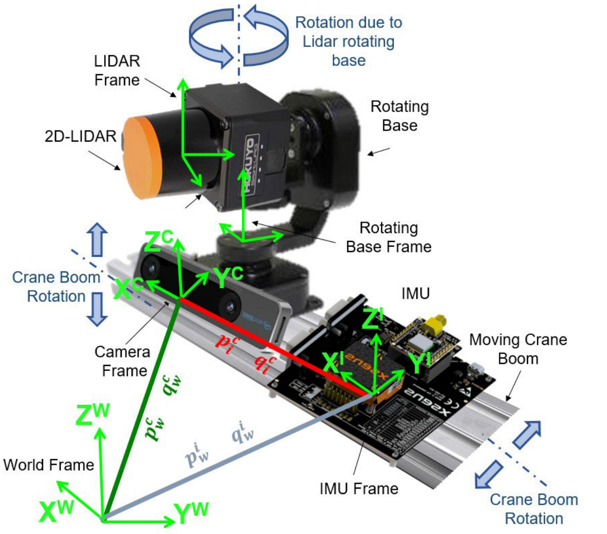


FIGURE 2. The sensor setup and coordinate frame attaced to each sensor is shown. Transformation between frames is represented by a rotation *q* and a translation *p*. The transformation between IMU and camera frame have fixed values, which is highlighted in red.

D. R-VIO

R-VIO [21] 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.

IV. 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 [13], [22], [23], [24], [25]. 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. 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

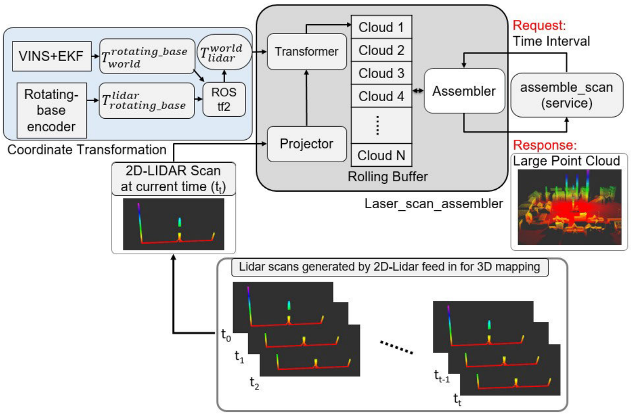


FIGURE 3. Block diagram of 3D mapping method using 2D lidar by laser assembler.

determination of the scale value for monocular cameras and improves the accuracy and robustness of the pose estimation.

A. 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:

$$w_t = \widetilde{w}_t - b_{w_t} - n_{w_t} \tag{1}$$

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:

$$a_t = \widetilde{a}_t - b_{a_t} - n_{a_t} \tag{2}$$

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 [16] and [26]:

$$\dot{b}_{w_t} = n_{b_w} \quad \dot{b}_{a_t} = n_{b_a} \tag{3}$$

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.

B. 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:

$$x_{25\times 1} = [p_w^i \quad v_w^i \quad \text{Â}q_w^i \quad b_w \quad b_a \quad \lambda\text{Â} \quad \text{Â}q_i^c]^T \tag{4}$$

where

$p_w^i = [p_x, p_y, p_z]^T$ is position along x, y, and z axes

$v_w^i = [v_x, v_y, v_z]^T$ is velocity along x, y, and z axes

$q_w^i = [q_w, q_x, q_y, q_z]^T$ 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

$q_i^c = [q_{i_w}^c, q_{i_x}^c, q_{i_y}^c, q_{i_z}^c]^T$ is the rotation between the IMU to the camera frame

$p_i^c = [p_{i_x}^c, p_{i_y}^c, p_{i_z}^c]^T$ 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:

$$\begin{aligned}
\dot{p}_i^w &= v_w^i \\
\dot{v}_i^w &= C_{q_w^i}^T(\widetilde{a} - b_a - n_a) - g \\
\dot{q}_w^i &= \frac{1}{2} q_w^i \otimes (\widetilde{\omega} - b_\omega - n_\omega) \\
\dot{b}_\omega &= n_{b_\omega} \quad \dot{b}_a = n_{b_a} \quad \dot{\lambda} = 0 \quad \dot{p}_c^i = 0 \quad \dot{q}_i^c = 0
\end{aligned} \tag{5}$$

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 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.

$$\begin{aligned}
p_k &= p_{k-1} + v_{k-1}.\Delta t + (C_{q_{k-1}}^T.a_k - g).\Delta t^2/2, \\
v_k &= v_{k-1} + (C_{q_{k-1}}^T.a_k - g).\Delta t, \\
q_k &= q_{k-1} \otimes q(\omega_k.\Delta t)
\end{aligned} \tag{6}$$

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

C. ERROR STATE

The error state captures the accumulated modeling errors and process noise. We estimate this small error in the error state EKF andÂ use it as a correction to the nominal state [25]. The error state vector is stated as

$$\delta x = [\delta p_w^i \quad \delta v_w^i \quad \text{Â}\delta\theta_w^i \quad \delta b_w \quad \delta b_a \quad \delta\lambda\text{Â} \quad \delta p_i^c \quad \text{Â}\delta\theta_i^c]^T \tag{7}$$

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

$$\begin{aligned}
\dot{\delta x} &= F\delta x + Gn \\
P_{k+1} &= FP_kF^T + Q
\end{aligned} \tag{8}$$

where F is kinematic model that propagates the errors over time. n is noise vector and can be expressed as

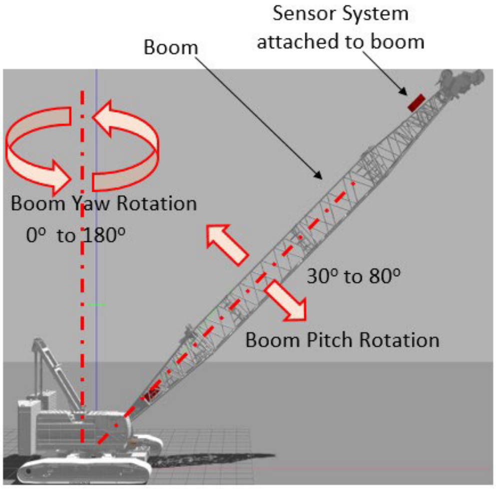
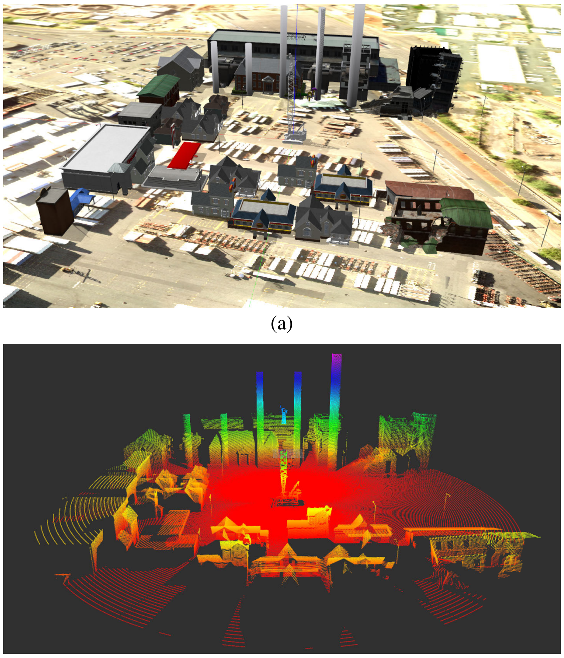


FIGURE 4. **The Gazebo simulation environment was used to create a crane model. The crane's boom has two different types of rotations.**

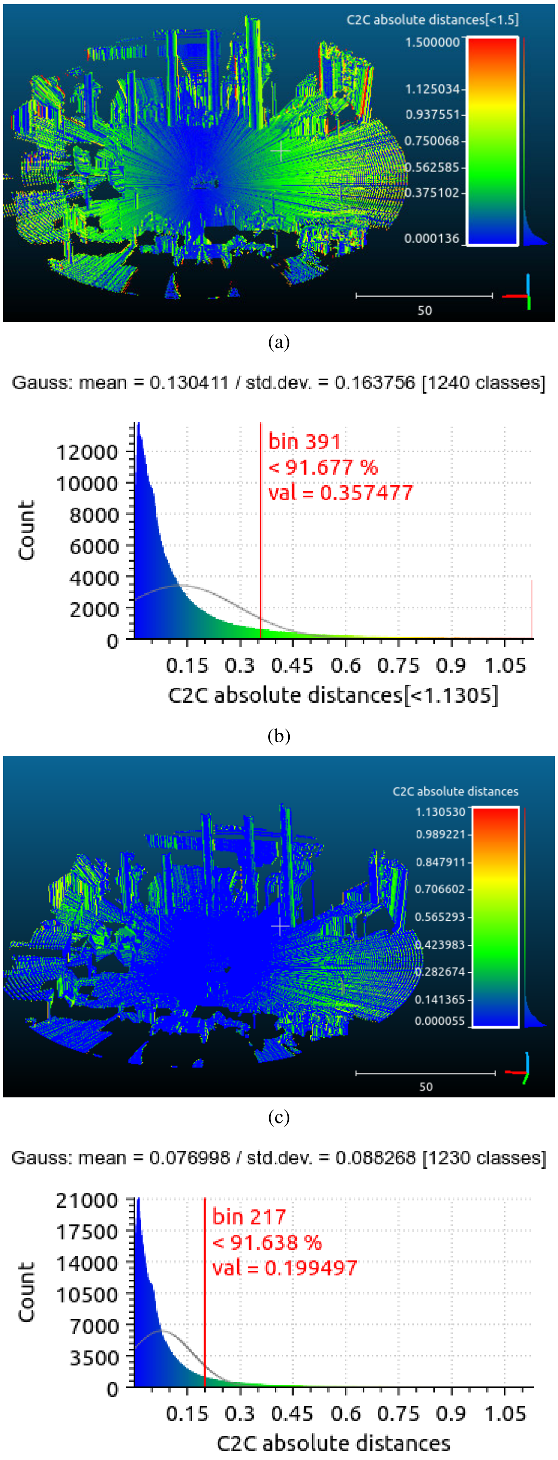


(a)

(b)

FIGURE 5. **3D map implemented in Gazebo simulation environment (a) Complex construction environment. (b) 3D point cloud map of complex construction environment. The varying colors in the map indicate the elevation or height of each individual point.**

$n = [n_a^T, n_{ba}^T, n_\omega^T, n_{b\omega}^T]$. Q is system or process noise covariance matrix and can be represented as a $Q = \text{diag}(\sigma_{n_a}^2, \sigma_{n_{ba}}^2, \sigma_{n_\omega}^2, \sigma_{n_{b\omega}}^2)$. P is state covariance matrix. The detailed explanation and derivation of F, G and Q can be found in [22] and [27]. For F, G and Q, we use same approach as given in [22].



(a)

(b)

(c)

(d)

FIGURE 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).**

D. MEASUREMENT MODEL

The measurement model for the camera pose measurement, obtained from the VINS can be expressed as follows [22], [23]

$$z = \begin{bmatrix} p_w^c \\ q_w^c \end{bmatrix} = \begin{bmatrix} (p_w^i + C_{(q_w^i)}^T p_i^c)\lambda + n_p \\ q_i^c \otimes q_w^i \end{bmatrix} \tag{9}$$

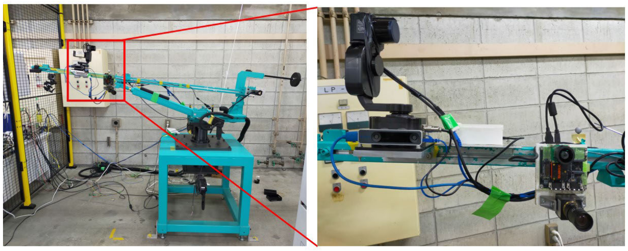


FIGURE 7. The experiment showcases the crane model along with its accompanying sensor system. The sensor system is connected to the boom of the crane.

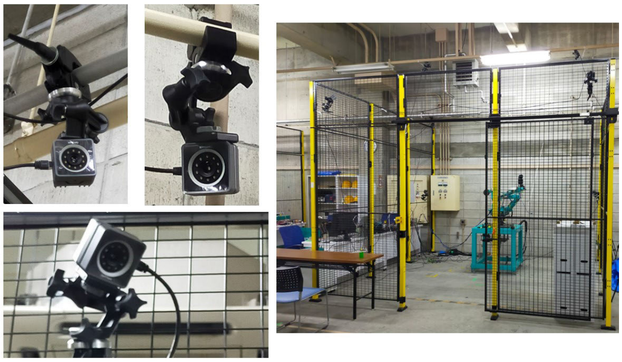


FIGURE 8. Motion capture system attached to crane environment.

The equation can be linearized as $z = H\delta x + n$ as given in [22] and [23], 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

$$\delta z = z - \hat{z} \tag{10}$$

Estimate the Kalman gain

$$K = PH^T(HPH^T + R)^{-1} \tag{11}$$

Calculate the correction

$$\widehat{\delta x} = K\delta z \tag{12}$$

Update the state covariance

$$P = (I - KH)P(I - KH)^T + KRK^T \tag{13}$$

V. MAPPING MODULE

To construct a dense 3D point cloud map, we have made some modifications to our previous approach proposed in [28]. 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. 2). Since the lidar is continuously in motion, the successive 2D lidar scan lines do not overlap with one 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 $T^{lidar}_{rotating\_base}$, 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 ($T^{rotating\_base}_{world}$) is measured using VINS+EKF. To calculate the transformation from the lidar frame to the fixed world frame (T^{lidar}_{world}), we establish a chain of transformations between the respective coordinate frames, as shown in the following equation.

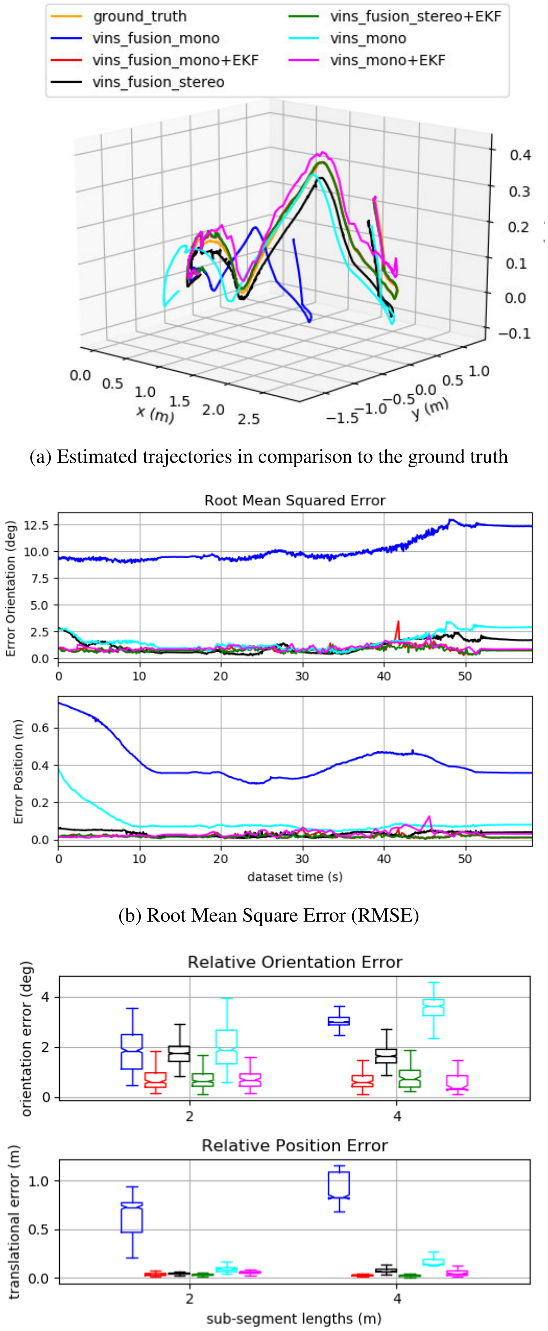
$$T^{lidar}_{world} = T^{lidar}_{rotating\_base} T^{rotating\_base}_{world} \tag{14}$$

The tf2 broadcaster [29], 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^{lidar}_{world} is obtained, it is used in the laser-assembler [30], [31] 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. 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.

VI. 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

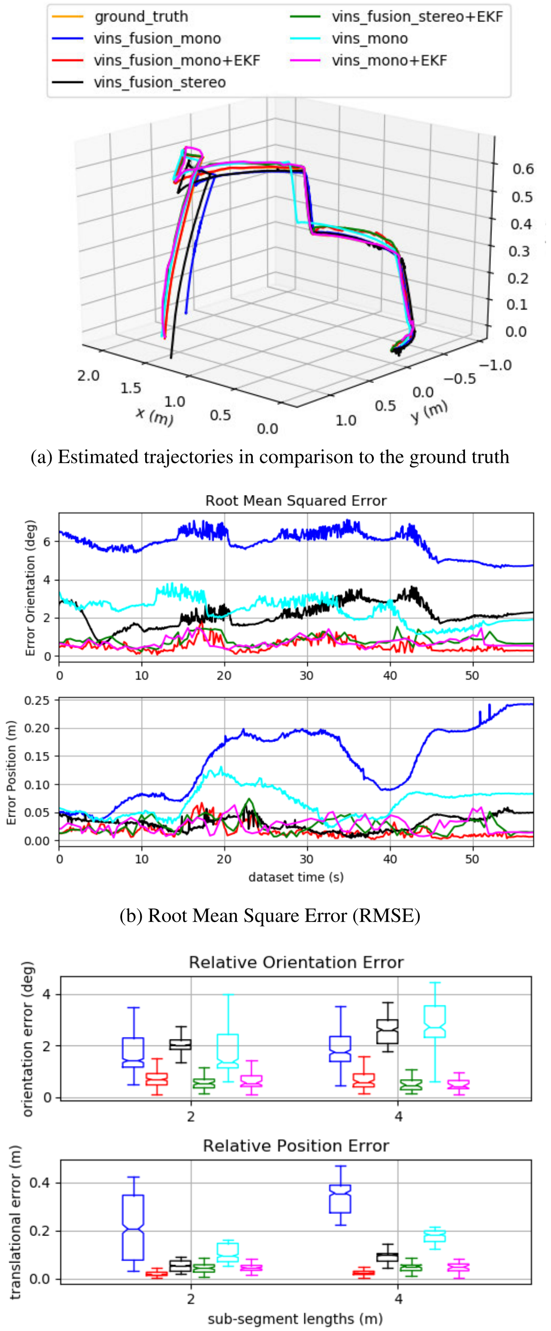


(a) Estimated trajectories in comparison to the ground truth

(b) Root Mean Square Error (RMSE)

(c) A boxplot summarizes the Relative Pose Error statistics (RPE) with trajectory segments of lengths 2 and 4 m

FIGURE 9. Trajectory evaluation for case 1. (Crane boom started moving with a jerk).



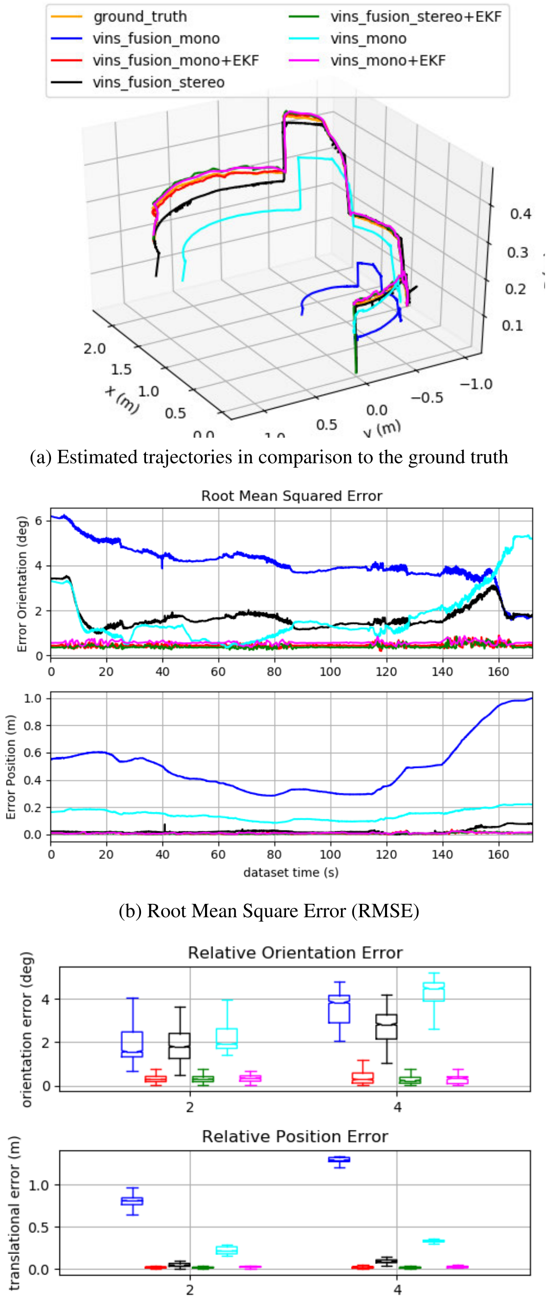
(a) Estimated trajectories in comparison to the ground truth

(b) Root Mean Square Error (RMSE)

(c) A boxplot summarizes Relative Pose Error (RPE) statistics with trajectory segments of lengths 2 and 4 m

FIGURE 10. Trajectory evaluation for case 2. (Crane boom moves in arbitrary motion).

package tool imu_utils and allan_variance_ros [32], [33] 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 [34], [35] is used to provide these camera calibration parameters to VINS. Kalibr calibration toolbox [36], [37] 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 8 × 6 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



(a) Estimated trajectories in comparison to the ground truth

(b) Root Mean Square Error (RMSE)

(c) A boxplot summarizes Relative Pose Error (RPE) statistics with trajectory segments of lengths 2 and 4 m

FIGURE 11. Trajectory evaluation for case 3. (Crane boom is subjected to large rotational changes).

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.

VII. SIMULATION RESULTS

We first conducted an evaluation of proposed mapping method using VINS and VINS+EKF in a simulated environment using the Gazebo simulator [38] and ROS environments. In the Gazebo simulation, we designed a robotic model of a crane as shown in Fig. 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(a). Fig. 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 [39]. Fig. 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. 6(a) and Fig. 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. 6(b) and Fig. 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.

VIII. EXPERIMENTAL RESULTS

The effectiveness of our proposed method was further evaluated through real-world experiments conducted on a crane model, as illustrated in Fig. 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. 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.

A. TRAJECTORY EVALUATION

In order to analyze the accuracy and robustness of different VINS and VINS+EKF approaches, we conducted a series

TABLE 1. Comparison of VINS methods in terms of Absolute Pose Error (APE). APE represents the error between the ground truth and the estimated pose, which is computed at each timestep and subsequently averaged.

Method	Case 1 (deg / m)	Case 2 (deg / m)	Case 3 (deg / m)
vins_fusion_mono	10.326 / 0.432	5.847 / 0.161	4.183 / 0.539
vins_fusion_mono+EKF	0.855 / 0.019	**0.556 / 0.018**	0.455 / 0.009
vins_fusion_stereo	1.306 / 0.033	2.221 / 0.034	1.795 / 0.030
vins_fusion_stereo+EKF	**0.773 / 0.017**	0.837 / 0.028	**0.393 / 0.009**
vins_mono	1.762 / 0.102	2.421 / 0.072	2.091 / 0.150
vins_mono+EKF	0.957 / 0.034	0.742 / 0.030	0.578 / 0.011

TABLE 2. Comparison of VINS methods in terms of Relative Pose Error with trajectory segments of lengths 2 and 4 meters.

Method	Case 1		Case 2		Case 3	
	2m (deg / m)	**4m** (deg / m)	**2m** (deg / m)	**4m** (deg / m)	**2m** (deg / m)	**4m** (deg / m)
vins_fusion_mono	1.797 / 0.641	2.875 / 0.918	1.701 / 0.209	1.878 / 0.343	1.903 / 0.798	3.608 / 1.280
vins_fusion_mono+EKF	0.768 / 0.039	0.697 / 0.028	0.741 / **0.022**	0.657 / **0.025**	**0.327** / 0.020	0.357 / 0.019
vins_fusion_stereo	1.744 / 0.044	1.687 / 0.079	2.133 / 0.053	2.575 / 0.091	1.881 / 0.048	2.800 / 0.091
vins_fusion_stereo+EKF	**0.693 / 0.035**	0.765 / **0.024**	**0.571** / 0.044	0.516 / 0.048	0.332 / **0.019**	**0.268 / 0.018**
vins_mono	1.979 / 0.115	3.480 / 0.177	1.765 / 0.107	2.747 / 0.176	2.215 / 0.221	4.257 / 0.330
vins_mono+EKF	0.739 / 0.064	**0.630** / 0.053	0.689 / 0.046	**0.507** / 0.044	0.330 / 0.026	0.297 / 0.025

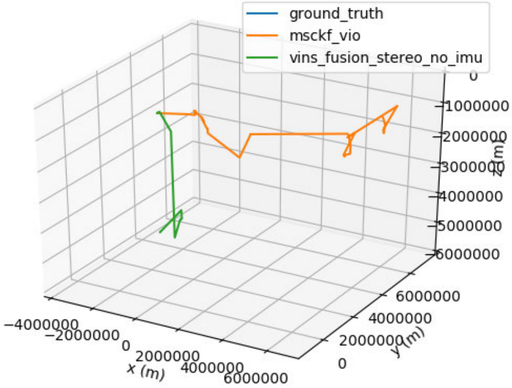


FIGURE 12. Trajectory evaluation using MSCKF and Vins-fusion-stereo-no-imu.

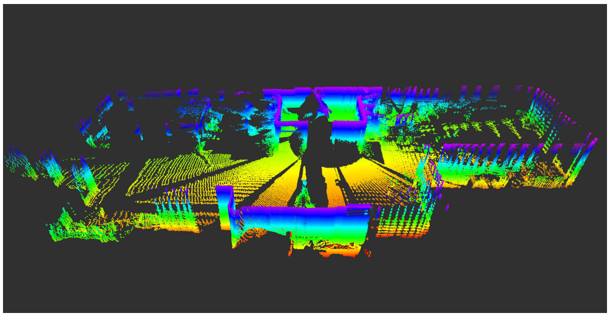
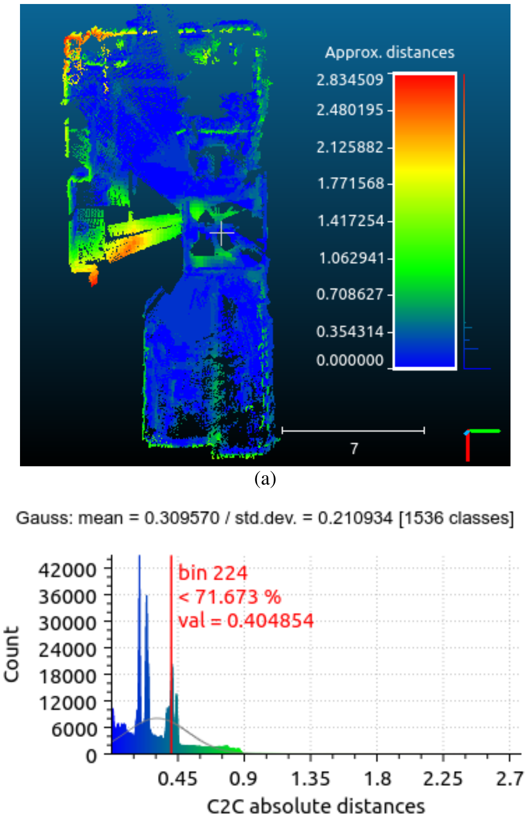


FIGURE 13. 3D map for crane environment.



(a)

(b)

FIGURE 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.

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 [40] and [41] to align the estimated trajectory with the ground truth. We then calculated the Root Mean Square Error (RMSE), Relative Pose Error (RPE) and Absolute Pose Error (APE) using [40], [41] 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's abrupt movement (with a jerk) was studied. Fig. 9(a), Fig. 9(b) and Fig. 9(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 second case, the crane boom moves in an arbitrary motion. Fig. 10(a) shows the trajectory estimated using VINS and VINS+EKF methods. Fig. 10(b) and Fig. 10(c) present the RMSE and RPE, respectively. We observed that VINS-Fusion-Mono+EKF had the lowest RMSE, while VINS-Fusion-Mono had the highest.

In the third case, the crane boom undergoes significant rotational changes. Fig. 11(a), Fig. 11(b), and Fig. 11(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.

Table 1 and Table 2 present the APE and RPE values, respectively, for the trajectory estimated using VIO algorithms across all three cases. The algorithm with the highest performance is emphasized in bold, and the algorithms with the next best performance are indicated with underlines. Based on these results, VINS-Fusion-Stero+EKF, VINS-Fusion-Mono+EKF and VINS-MONO+EKF approaches demonstrated precise and consistent performance, making it a robust choice for crane state estimation.

B. 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. Fig 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. Fig. 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, Fig. 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.

IX. CONCLUSION

In this paper 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.

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 & mapping," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 289–294, Jun. 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, no. 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.

[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. J. O'Brien, "Requirements for autonomous crane safety monitoring," in *Computing in Civil Engineering*. American Society of Civil Engineers (ASCE), 2011, pp. 331–338. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/41182%28416%2941

[13] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," in *Intelligent Autonomous Systems 13*, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds. Cham, Switzerland: Springer, 2016, pp. 335–348.

[14] J. B. Bancroft and G. Lachapelle, "Data fusion algorithms for multiple inertial measurement units," *Sensors*, vol. 11, no. 7, pp. 6771–6798, Jun. 2011. [Online]. Available: https://www.mdpi.com/1424-8220/11/7/6771

[15] B. Joshi, S. Rahman, M. Kalaitzakis, B. Cain, J. Johnson, M. Xanthidis, N. Karapetyan, A. Hernandez, A. Q. Li, N. Vitzilaios, and I. Rekleitis, "Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 7227–7233.

[16] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[17] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019, *arXiv:1901.03638*.

[18] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," 2019, *arXiv:1901.03642*.

[19] T. Qin, S. Cao, J. Pan, P. Li, and S. Shen. (2016). *VINS-Fusion: An Optimization-Based Multi-Sensor State Estimator*. [Online]. Available: https://github.com/HKUST-Aerial-Robotics/VINS-Fusion

[20] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 3565–3572.

[21] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," *Int. J. Robot. Res.*, vol. 41, no. 7, pp. 667–689, 2022.

[22] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4531–4537.

[23] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 3923–3929.

[24] L. Markovic, M. Kovac, R. Milijas, M. Car, and S. Bogdan, "Error state extended Kalman filter multi-sensor fusion for unmanned aerial vehicle localization in GPS and magnetometer denied indoor environments," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2022, pp. 184–190.

[25] I. Brigadnov, A. Lutonin, and K. Bogdanova, "Error state extended Kalman filter localization for underground mining environments," *Symmetry*, vol. 15, no. 2, p. 344, Jan. 2023. [Online]. Available: https://www.mdpi.com/2073-8994/15/2/344

[26] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Found. Trends Signal Process.*, vol. 11, nos. 1–2, pp. 1–153, 2017.

[27] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Norwood, MA, USA: Artech House, Mar. 2013.

[28] M. U. Hassan, D. Das, and J. Miura, "3D mapping for a large crane using rotating 2D-LiDAR and IMU attached to the crane boom," *IEEE Access*, vol. 11, pp. 21104–21116, 2023.

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

[30] 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

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

[32] Gaowenliang. *IMU_Utils—IMU Sensor Utility Library*. GitHub repository. Accessed: 2023. [Online]. Available: https://github.com/gaowenliang/imu_utils

[33] R. Buchanan. *Allan_Variance_Ros—Allan Variance ROS Package*. Accessed: 2023. GitHub Repository. [Online]. Available: https://github.com/ori-drs/allan_variance_ros

[34] OpenCV Documentation Contributors. *OpenCV Camera Calibration and 3D Reconstruction Documentation*. OpenCV Documentation. Accessed: 2023. [Online]. Available: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

[35] ROS Wiki Contributors. *ROS Camera Calibration Wiki*. ROS Wiki. Accessed: 2023. [Online]. Available: http://wiki.ros.org/camera_calibration

[36] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Mar. 2013, pp. 1280–1286.

[37] R. Furgale. *Kalibr: A Generic Multi-Use Toolbox for Calibration of Sensor Systems*. Accessed: 2023. [Online]. Available: https://github.com/ethz-asl/kalibr

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

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

[40] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 7244–7251.

[41] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4666–4672. [Online]. Available: https://github.com/rpng/open_vins

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.

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.