ORIGINAL ARTICLE

Mapping and localization for autonomous ship using LiDAR SLAM on the sea

Ryohei Sawada[1,2] · Koichi Hirata[1]

Abstract

In this study, the accuracy of estimation of the ship's position and the heading angle using simultaneous localization and mapping (SLAM) with a light detection and ranging (LiDAR) at sea was verified. In general, the ship's position and the heading angle are obtained using a global navigation satellite system (GNSS) such as global positioning system (GPS) positioning and a GPS compass. The quasi-zenith satellite system (QZSS) has also emerged in Japan with the centimeter-level positioning augmentation system (CLAS), but it does not always provide the best positioning accuracy, even at sea. In addition, while position information and the heading angle of ship are important for control of maritime autonomous surface ships (MASS), they are conventionally dependent on GNSS-based sensor systems. In considering sensor redundancy for safety of MASS, this study performed SLAM using the LiDAR to create a point cloud map of the coast. The point cloud map was compared with open map data, and it was confirmed that a highly accurate map was obtained. This point cloud map can be used for autonomous navigation such as automatic berthing. In order to evaluate the accuracy of estimating the ship's position and the heading angle using LiDAR SLAM, we used the experimental ship to simultaneously take measurements with the marine GPS compass and the QZSS receiver for comparison, in addition to the IMU and the LiDAR, and compared the measurement accuracy by the three sensors. As a result, the position and the heading angle were estimated with higher accuracy using LiDAR SLAM at $\mu = 1.4$ m ($\sigma = 1.2$ m) for position estimation than the GPS with the QZSS data in the RTK-Fix condition as a reference.

Keywords Automatic berthing · LiDAR SLAM · Localization · Mapping · GNSS

1 Introduction

Estimation of ship's pose including the position and the heading angle is important for automatic control of a ship. For example, in automatic berthing control in [1], the distance from the floating pier at the time of berthing was set at 5 [m] for 4.38 [m] width of the ship hull. The positional accuracy required for control here is less than that, and for this purpose, positional information accuracy of approximately 1 [m] is required. Since the GPS conventionally used for ship operation cannot meet this accuracy, the authors used a receiver compatible with the quasi-zenith satellite

system (QZSS) to measure the ship's position ([2]). The QZSS receiver used in this study is capable of using CLAS (Centimeter Level Positioning Augmentation System, [3]), which is capable of centimeter-level positioning. However, a common problem with GNSS (Global Navigation Satellite System) is that it does not always provide the best accuracy depending on reception conditions and satellite constellations. Figure 1 shows the GNSS quality of the QZSS receiver during navigation in the test area of the experimental ship. As this figure shows, the QZSS receiver does not always obtain a fix solution during navigation, so the positioning accuracy is not always in the centimeter-level, even in open seas. The ship's heading angle is also based on a GPS compass, which uses two GPS antennas to measure the ship's attitude by GPS positioning. The current means of estimating ship's position and attitude relies strongly on GNSS. In building automated systems of ships, redundancy of sensors is important to guarantee the reliability of the system, and alternative means to GNSS should be considered.

✉ Ryohei Sawada
  sawada-r@m.mpat.go.jp

1 National Maritime Research Institute, 6-38-1, Shinkawa, Mitaka, Tokyo 181-0004, Japan

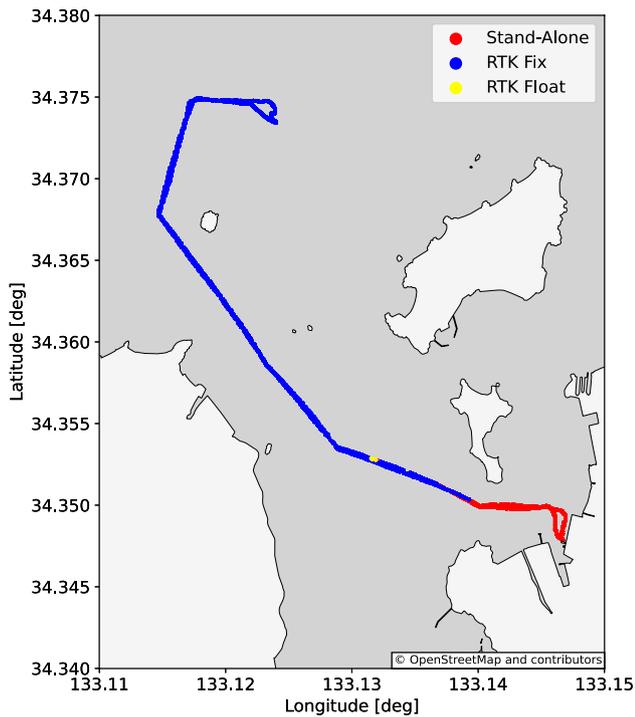2 Osaka University, 2-1 Yamadaoka, Suita, Osaka, Japan

Fig. 1 An example of measured GNSS quality data of the QZSS receiver around the test sea area

Recently, SLAM (Simultaneous Localization and Mapping) has been widely used in the fields of autonomous vehicles ([4, 5]) and mobile robotics ([6–9]). SLAM uses sensors such as cameras, image sensors, and LiDARs (Light Detection and Ranging) to measure environmental information, which is then used to simultaneously create a map of the surrounding environment and estimate self-position. For self-position estimation, the horizontal position and orientation of a mobile object are usually estimated. Thus, it may be used as an alternative to GNSS-based methods such as GPS positioning and a GPS compass ([10]). There are, however, few studies of SLAM at sea using a ship designed for use in MASS (Maritime Autonomous Surface Ship).

This study examines the availability of LiDAR-based self-position estimation and mapping technology as a means of positioning and estimating the heading angles other than GNSS, especially for use during berthing, when high accuracy of localization is required. Specifically, we applied the LiDAR-based SLAM algorithm on the sea and verified the performance of point cloud mapping, positioning, and heading angle estimation at the coast around a pier.

This paper is organized as follows: in the next section, we discuss previous research relevant to the present study, followed by an explanation of method including SLAM algorithms, sensors, software and measurement of data. In Sect. 4, we show results of mapping and localization using LiDAR SLAM and evaluate them by comparing them with

GPS/QZSS data and open map data. We discuss the practical use of SLAM on an autonomous ship in Sect. 5. Finally, we conclude the paper.

2 Related work

LiDAR and SLAM using LiDAR (LiDAR SLAM) are widely used for mobile robotics and autonomous driving of cars. On the other hand, there are few examples of LiDAR SLAM in ships due to several difficulties.

SLAM at sea has been conducted by Ødven [11]. In this example, 2D SLAM, Hector SLAM ([12]), and two 3D SLAM algorithms, LOAM (LiDAR Odometry and Mapping, [13]) and BLAM (Berkeley Localization and Mapping, [14]), were applied to localization and mapping in a port using Velodyne VLP-16, which is 16 line LiDAR. This study shows the estimation error of the position and the heading angle when compared to RTK-GPS (Real-Time Kinematic GPS) positioning and the GPS compass heading angle. In the study, neither method has been able to produce a point cloud map with sufficient accuracy for navigation. As a result, Ødven reported that neither SLAM algorithm was sufficient to serve as a complete solution for operation of the autonomous ferry.

In another study, Shan et al. [15] used measurements taken during a three-hour cruise of Amsterdam's canals as a dataset for validation of LIO-SAM. The canals are surrounded by the shore on both sides of the navigation channel, and the measurement distance was small (the canal's width is approximately 20–100 [m]). The present paper assumed the typical case of berthing at sea, and SLAM was performed by measuring the shore from offshore. In their study, Shen et al. point out that SLAM in canals is a difficult problem for several reasons. Indeed, despite the least vertical displacement and smaller maximum turning velocity than other land-based measured data sets, the SLAM success rate is the lowest, and three of the four SLAM algorithms compared fail to produce meaningful results. One reason for the difficulty of SLAM on canals is that there are significantly fewer planar feature points on the water due to the absence of the ground, and the lack of planar feature points makes vertical point cloud integration difficult. In this study, the matching error of SLAM was reduced by adding a constraint to the coordinate transformation in the scan-matching calculation, which transforms the point cloud scanned with LiDAR into the coordinates of the point cloud map.

Zhou et al. [16] proposed S4-SLAM algorithm for multi-scene environments including ground and water-surface. They showed results without GPS and the inertial navigation system (INS) for harbor dataset using S4-SLAM algorithm. As reported in [15], LOAM fails on some harbor data here as well in this study. In their research, however, only

the trajectory plots and final drift error are shown, and no evaluation of the heading angle estimation is made. The loop closure of S4-SLAM probably did not work because the trajectories of the ship was not looped in any of the harbor datasets. The loop closure refers to the post-processing that is performed during SLAM to distort the entire map being created. The result was a drift of 0.56 to 0.93%, which was expressed as a percentage of deviation relative to the total displacement distance of the ship in the paper, for a few kilometers of measurements.

Based on the above previous research, we will conduct LiDAR and IMU (Inertial Measurement Unit) measurements in the test area from the *Innoshima marina* in Japan, where the experimental ship is berthed, to the area near the high-speed boat terminal in the north to verify the availability of LiDAR SLAM at sea.

3 Method

3.1 Algorithms of SLAM

SLAM refers to a technology that simultaneously estimates the self-position of a mobile object and maps its environment. The process of SLAM can be divided into two main categories. The first is the front-end part that sequentially processes the data measured by the sensors such as a LiDAR. The other is the back-end part, which focuses on optimization of the pose graph.

In this study, two algorithms were used to verify SLAM performance. One is a simple algorithm, which consists of a front-end part of NDT (Normal Distributions Transform [17]) scan matching using PCL (Point Cloud Library, [18]) and a back-end part of graph-based SLAM using graph optimization with g^{2}o (General Graph Optimization, [19]). NDT is one of the methods for matching two point clouds. In NDT, the reference point cloud is divided into voxels, the distribution of points in the voxels is approximated by a normal distribution, and matching is performed with the newly scanned point cloud data. The matching involves minimizing the following evaluation function:

$$E(\boldsymbol{p}) = \sum_{i=0}^{N-1} \exp \frac{-(\boldsymbol{x}_i' - \boldsymbol{q}_i)^t \sum_i^{-1} (\boldsymbol{x}_i' - \boldsymbol{q}_i)}{2}, \tag{1}$$

where $\boldsymbol{p} = (t_x, t_y, \theta)^t$ is a coordinate transformation parameters, $\boldsymbol{x}_i$ is i-th input scan point cloud data, $\boldsymbol{x}_i'$ is the transformed scan point cloud data of $\boldsymbol{x}_i$ using the parameter $\boldsymbol{p}$, $\sum_i$ and $\boldsymbol{q}_i$ are the covariance matrix and mean coordinates of the transformed scan point cloud data $\boldsymbol{x}_i$, and N is the number of points. The advantage of NDT is that it is generally less computationally intensive than ICP (Iterative Closest Point, [20]), which evaluates the distance between all points,

because the amount of data can be reduced by dividing the reference point clouds into voxels. The back-end graph optimization is mainly used to apply the loop closure removing map distortions. The loop closure corrects errors accumulated due to sensor noise and sequential scan matching. The loop closure detects a loop when the measurement target (in this case, a ship) circles around in an environment and returns to its original position during the SLAM run, and corrects the entire map so that the measurement results point to the same location.

The other algorithm used in this study is LIO-SAM (tightly-coupled LiDAR Inertial Odometry via Smoothing and Mapping, [15]), which use an IMU to remove distortion of maps. LIO-SAM pre-integrates IMU measurements to motion estimation to de-skew point clouds and provides an initial guess for LiDAR odometry optimization. LIO-SAM uses code of LeGO-LOAM ([21]) and adopts scan-matching algorithm by [22]. LIO-SAM framework formulates the problem using factor graph ([23]) and pre-integrates IMU measurements as the IMU pre-integration factor ([24, 25]). The ship state x can be defined as

$$x = [R^T, p^T, v^T, b^T], \tag{2}$$

where $R \in SO(3)$ is the rotation matrix, $p \in \mathbb{R}^3$ is the position vector, v is the velocity, and b is the bias of the IMU measurement. The transformation $T \in SE(3)$ from the world frame W to the body frame B is defined as $T = [R|p]$.

The IMU measurements of angular velocity and acceleration are defined as follows:

$$\hat{\omega}_t = \omega_t + b_t^\omega + n_t^\omega, \tag{3}$$

$$\hat{a}_t = R_t^{BW}(a_t - g) + b_t^\omega + n_t^\omega, \tag{4}$$

where $\hat{\omega}_t$ and $\hat{a}_t$ are the raw IMU measurements in the coordinates B as time t. b_t and n_t are the bias and white noise. R_t^{BW} is the rotation matrix from W to B. g is the constant gravitational acceleration vector in W. The pre-integration of IMU measurements proposed in [24] is applied using the relative body motion between two timesteps as follows:

$$\Delta v_{ij} = R_i^T(v_j - v_i - g\Delta t_{ij}) \tag{5}$$

$$\Delta p_{ij} = R_i^T(p_j - p_i - v_i\Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2) \tag{6}$$

$$\Delta R_{ij} = R_i^T R_j, \tag{7}$$

where v, p and R are the velocity, position and rotation of the motion of the ship. This IMU pre-integration factor will be incorporated into the factor graph of LIO-SAM. Keyframes of LiDAR scans are associated with the transformations and

merged into a voxel map M_i. A voxel map consists of the LiDAR keyframes including the transformed edge and planar features $'F_i^e$ and $'F_i^p$. The distance between the edge and planar features extracted from the scan point cloud can be calculated by the following equation:

$$d_{e_k} = \frac{\left|\left(p_{i+1,k}^e - p_{i,u}^e\right) \times \left(p_{i+1,k}^e - p_{i,v}^e\right)\right|}{\left|p_{i,u}^e - p_{i,v}^e\right|} \tag{8}$$

$$d_{p_k} = \frac{\left|\left(p_{i+1,k}^p - p_{i,u}^p\right) \cdot \left(p_{i,u}^p - p_{i,v}^p\right) \times \left(p_{i,u}^p - p_{i,w}^p\right)\right|}{\left|\left(p_{i,u}^p - p_{i,v}^p\right) \times \left(p_{i,u}^p - p_{i,w}^p\right)\right|}, \tag{9}$$

where k, u, v, and w are the feature indices. $p_{i+1,k}^e$ is an edge feature in $'F_{i+1}^e$ in W corresponding edge line consists of the points $p_{i,u}^e$ and $p_{i,v}^e$. For planar feature $p_{i+1,k}^p$ in $'F_{i+1}^p$, $p_{i,u}^p$, $p_{i,v}^p$ and $p_{i,w}^p$ are the points from the corresponding planar patch in M_i^p. The optimal transformation T_{i+1} is obtained by minimizing as follows:

$$\min_{T_{i+1}} \left\{ \sum_{p_{i+1,k}^e \in 'F_{i+1}^e} d_{e_k} + \sum_{p_{i+1,k}^p \in 'F_{i+1}^p} d_{p_k} \right\}. \tag{10}$$

Using the transformation T_i, we can obtain the ship state as follows:

$$x_{i+1} = \Delta T_{i,i+1} x_i, \tag{11}$$

where the relative transformation $\Delta T_{i,i+1} = T_i^\top T_{i+1}$. We refer the reader to the description from [15] for the detailed algorithm of LIO-SAM. We set the noise and bias parameters for acceleration and gyro of LIO-SAM to 10 times those of the original to adjust to the IMU used in this study. Several parameters were adjusted and combinations were considered in running the SLAM. In particular, the tolerance settings of coordinates transformation of point clouds had a significant impact, especially in suppressing the drift during matching point clouds as described in the next section.

3.2 SLAM for few horizontal planar features

SLAM at sea is considered difficult because the water depth near the shore is often shallow, and the measurement distance from the shore tends to be long, reducing the number of LiDAR scan points that can be measured. Furthermore, SLAM at sea has the fatal feature that point clouds of the ground surface and sea surface can be hardly acquired, and thus horizontal planar features cannot be sufficiently obtained. This is not merely a matter of having fewer point clouds available. Even when a sufficient scanned point cloud
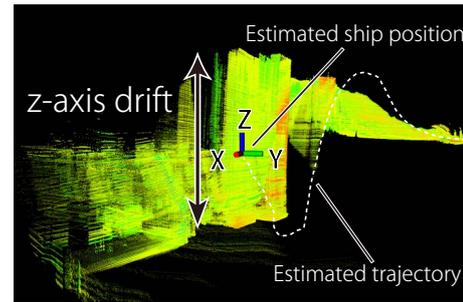

Fig. 2 Z-axis drift of scan matching on the sea
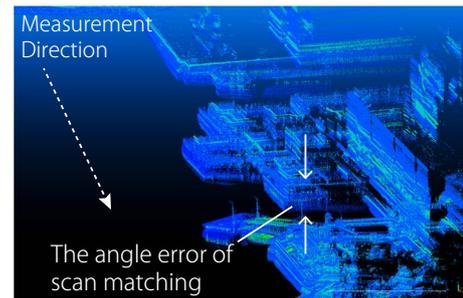

Fig. 3 The error in angular transformation of scan matching in the roll direction of the ship

is available, while coordinate transformations of the horizontal position and yaw angle can be calculated, the lack of horizontal planar features results in matching errors in the vertical and roll and pitch directions. Figures 2 and 3 show typical examples of scan-matching failures in SLAM at sea. Figure 2 shows a point cloud matching error in the vertical z-axis direction. Because of the map matching failure, the estimated trajectory of the ship also took an improbable trajectory in the vertical direction. In Fig. 3, a matching error occurs in the roll direction relative to the measurement direction, resulting in a point cloud map that appeared to be doubly overlapped. This error can also occur in the pitch direction. Figure 4 illustrates the causes of these scanning errors: in examples (b) and (c) of Fig. 4, the heave displacement and roll angle of the ship cannot be determined because horizontal planar features cannot be obtained. The LIO-SAM used in this study utilizes IMU observations to optimize LiDAR odometry, which removes distortion of scanned point cloud and provides an initial guess of scan matching. The IMU measurements thus only affect the initial value of the optimization and do not fundamentally solve the problem of matching failures described in Fig. 4 as a result of the optimization.

In this study, the following coordinate transformation constraints were introduced in the SLAM calculation to suppress matching errors due to missing horizontal planar
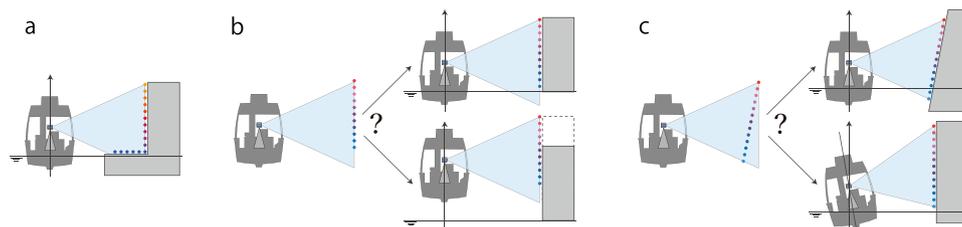
Fig. 4 Examples of SLAM failures at Sea. **a** is an example where horizontal planar features are obtained with LiDAR scan, in which case the position and angle of the ship (LiDAR) in the vertical plane can be determined. **b** and **c** are examples where horizontal planar features aren't obtained. In **b**, the heave displacement of the ship (LiDAR) cannot be determined because the true height of the object is not known from the acquired point clouds. In **c**, it is not possible to determine whether the object being measured is tilted or the ship (LiDAR) is tilted because it is not known how much the obtained point cloud is tilted in the space-fixed coordinates

Fig. 5 Experimental ship "Shinpo" used for measurement data

Fig. 6 Sensor configuration of LiDAR and IMU

Fig. 7 Sensor configuration of GNSS (GPS and QZSS)

features at sea: (1) for the z-axis coordinate transformation T_z, $T_z < 1.0$ [m]; (2) for coordinate transformations of roll angle T_{roll} and pitch angle T_{pitch}, T_{roll}, $T_{pitch} < 10$ [deg]. The parameters of constraints were determined by trial and error for the measurement data. We tested these constraints with smaller values than above, which did not affect the accuracy of the results. Furthermore, the point cloud of 30 [m] around the LiDAR was excluded so as not to include the point cloud of the own ship and ship wake that would be noise during the scan matching of the point cloud.

3.3 Sensors, software, measurement

We used the experimental ship "Shinpo" to collect data for this study (Fig. 5). The LiDAR sensor used in this study was a Velodyne VLP-32C LiDAR and installed on deck using a tripod. The IMU used to correct the LiDAR values was a sensor by WitMotion with a 9-axis sensor (3-axis acceleration + 3-axis gyro + 3-axis magnetometer). The output frequency of the 9-axis sensor was set to 50 Hz. To measure the motion of the LiDAR body using the 9-axis sensor, a PLA (Polylactic Acid) resin mount was made using an FDM (Fused Deposition Modeling) 3D printer instead of
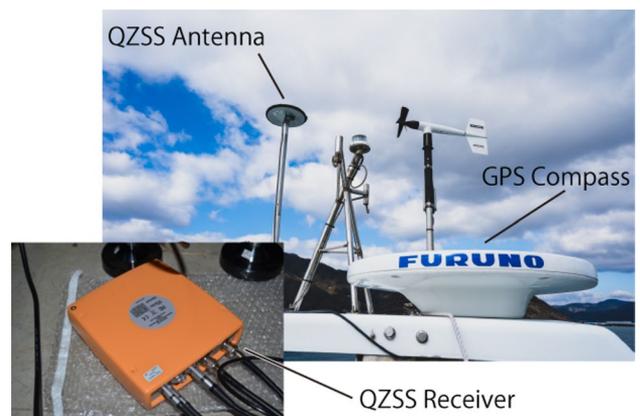
using a metal mount, considering the effect on the IMU's geomagnetic sensor. The mount was successfully fixed to the LiDAR. Figure 6 shows the installation of the sensor in the experiment. To evaluate the results of LiDAR SLAM, we prepared the marine GPS compass and the QZSS receiver for this study as shown in Fig. 7. The marine GPS was installed on the roof of the cockpit to provide GPS positioning and heading. The QZSS receiver was connected to two antennas, one near the GPS compass and one near the bow of the experimental ship, to enable measuring the ship's position
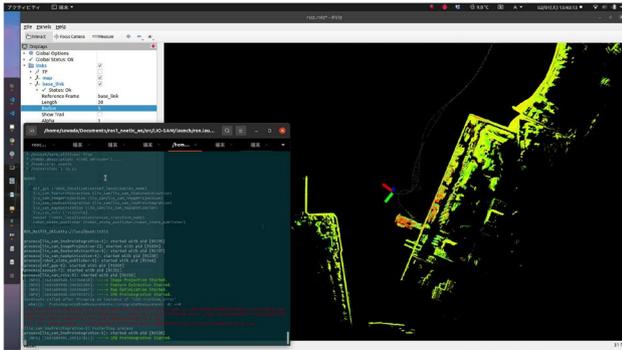
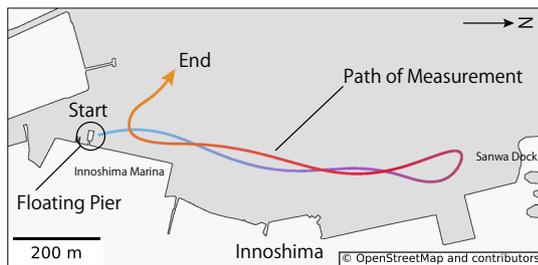**Fig. 8** Screenshot of LiDAR SLAM (LIO-SAM)

Fig. 9 Location and trajectory of measurement

and heading angle. LiDAR and IMU data were recorded as rosbag, a ROS (the Robot Operating System) data log format, for use in SLAM. GPS and QZSS measurement data were also logged using the onboard PLC (Programmable Logic Controller) system ([1]).

SLAM algorithms in this paper were run on ROS, which is widely used to develop robotics. The NDT-based SLAM was run on ROS 2 Galactic. LIO-SAM was run on ROS 1 Noetic. Both were run on the laptop with Ubuntu 20.04 LTS and equipped with the Core i7-9750 H processor and 16 GB RAM. Figure 8 shows a screenshot of LIO-SAM running on the laptop. The screen outputs the computation logs for scan

matching and graph optimization, and the current map creation is shown on the Rviz visualization software. In this case, we manually sent commands to output the created map and the results of self-position estimation for analysis. The map was created along a route starting from the floating pier to the beach north of the floating pier where the dock is located and back as shown in Fig. 9. The average speed over ground of the ship during the measurement was 4.2 [kt].

4 Results

4.1 Mapping

SLAM with the NDT-based algorithm and LIO-SAM was performed on the data measured using the experimental ship. The point cloud map created in this study is shown in Figs. 10 and 11. For comparison, the point cloud map is superimposed on the aerial image derived from Google Earth. The two cases SLAM performed in Figs. 10 and 11 are different measurement data, one without the IMU and one with the IMU, and the measurement paths are the almost same as those shown in Fig. 9. In the NDT-based algorithm without the IMU, the map distortion increases after leaving the floating pier where the Innoshima Marina is located at the starting point of measurement. In this example, the loop closure to remove the distortion of the map was also not performed because the map creation failed in the middle of the process due to the significant distortion. The result of SLAM using LIO-SAM is shown in Fig. 11. In this example, the loop closure worked many times during the SLAM computation, indicating that the map distortion was appropriately removed. A closer look reveals a slight distortion as the distance from the floating pier, the starting point of the mapping, increases compared to the aerial image. Although the two algorithms cannot be easily compared because they use different cartographic methods other than with and without the IMU, the LIO-SAM produced more accurate maps.

Fig. 10 Point cloud map generated using the NDT-based algorithm without an IMU. The background aerial photograph is referred to Google Earth

Fig. 11 Point cloud map generated using the LIO-SAM algorithm. The background aerial photograph is referred to Google Earth

In the case of automatic ships, maps are used to monitor the ship's position and course on a monitoring screen. For standard vessels, the electronic navigational chart (ENC) such as the ECDIS (Electronic Chart Display and Information System) and radar (radio detecting and ranging) are used for navigation For automatic berthing control navigation, we have developed a GUI for automatic berthing control, converting map data created with OpenStreetMap into a 3D map in a previous study. Here, a point cloud map created by LIO-SAM is superimposed on this 3D map, and they are compared. Figure 12 shows a screenshot of the automatic berthing GUI, and below it is a magnified image of the area around the floating pier. As can be seen from the screenshot, overall, the point cloud and the 3D map are generally consistent. However, when looking at the point cloud around the floating pier, the location is out of alignment with the map. Also, the position of the girder for bringing the ship ashore, which protrudes to the left of the floating pier, is also different. These mapped shapes are out of alignment not only with OpenStreetMap but also with other map data, such as Google Maps and maps made by GSI (Geospatial Information Authority of Japan). In both cases, the floating pier is not on the map in the first place. Electronic navigational charts are not open data, making it difficult to use them for autonomous ships tasks such as obstacle avoidance and path planning, for example. Open map data are easy to use for path calculation and monitoring GUI, but has accuracy problems; if LiDAR can be used to create highly accurate maps, applications such as path planning that takes into account port geometry, as in [26], can be expected.

In the next section, we compare the ship's position and the heading angle estimated by LIO-SAM with data measured using GPS and QZSS.

4.2 Localization

This section compares the self-position estimation of LiDAR SLAM (LIO-SAM) and positioning performance of GPS and QZSS. The trajectories of the measurement is shown
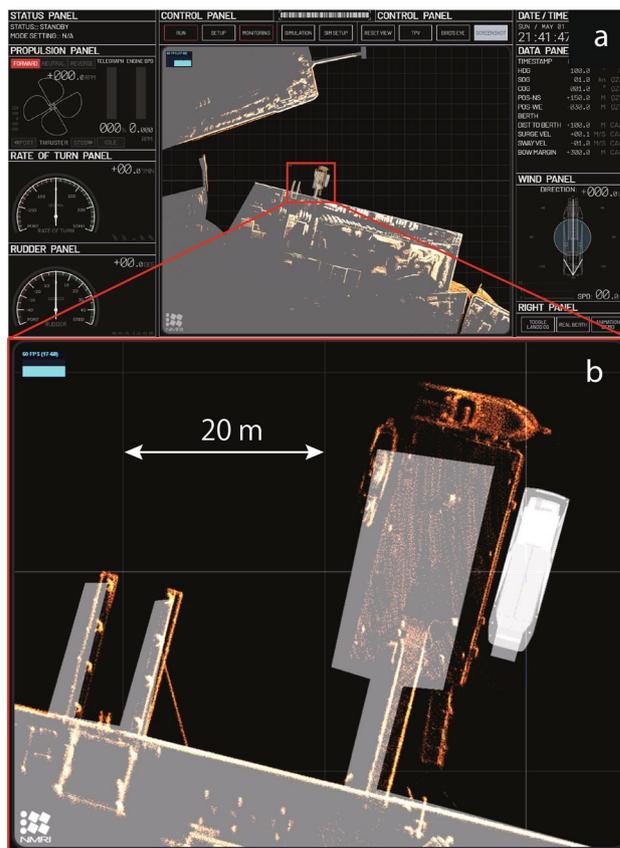

Fig. 12 a The point cloud map created using LIO-SAM superimposed on the 3D map data in the automatic berthing control GUI. **b** Enlarged image of the area around the floating pier. The original map data of 3D map is referred to OpenStreetMap

in Fig. 13. The QZSS positioning information during the measurement was always state of a fix solution of RTK (Real-Time Kinematic) positioning. The position shown in the figure represents the ship's midship position, which was calculated from the onboard locations of the LiDAR and GPS/QZSS antennas. The ship's heading angle information in the following figures was obtained from the localization

Fig. 13 Comparison of trajectories acquired by SLAM, QZSS and GPS. The point cloud map in this figure was downsampled for plotting
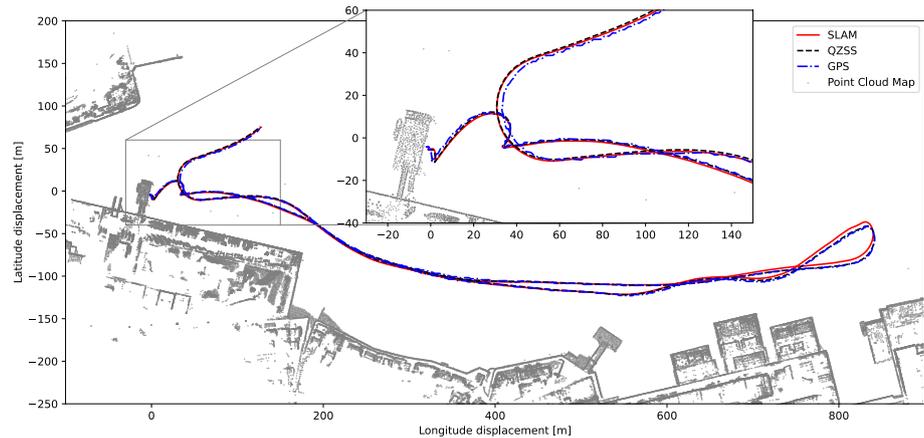
results for SLAM, the GPS compass for GPS, and the QZSS compass for QZSS, respectively. Consequently, it should be noted that the accuracy of the heading angle data from these sensors affected the error in the midship position estimation. Within the latitudinal displacement of 600 [m] or less, the trajectories obtained by the three methods agree well. However, an enlarged view of the area near the pier in Fig. 13 shows some sections where the GPS trajectories deviates significantly from the other two. In contrast, trajectories of SLAM and QZSS appear to be almost identical. In addition, the overall trajectory using GPS is not smooth, while SLAM and QZSS trajectories are smooth.

Next, the time series of latitude and longitude displacement and the heading angle estimated or measured with SLAM, GPS, and QZSS are shown in Fig 14. The heading angle is shown in a range of ± 180 degrees with north as zero degrees, east as positive, and west as negative, for ease of reading the graph. In Fig. 14, it can be seen that the values obtained by the three methods are generally consistent in most sections. However, Fig. 14 reveals that SLAM had a more significant error than GPS/QZSS, in particular for the heading angle at the beginning of the measurement. This error in the heading angle estimation is assumed to be due to incomplete mapping just after the start of SLAM computation.

Since a simple comparison of the time series makes it difficult to understand the error of each estimate, Fig. 15 shows the difference based on the value from QZSS with RTK fix. This figure clearly shows the difference between GPS and SLAM against QZSS. The difference in the displacement estimated using SLAM seen in Fig. 15 is larger in the longitude direction than the difference in the latitude direction. The results of GPS is noisier than SLAM and QZSS, and the positioning results are more varied. Looking at the displacement in the latitudinal direction, it can also be read that GPS errors occur in the opposite direction of movement and this indicates that the GPS positioning results are delayed. Regarding the heading angle, it is surprising that SLAM was
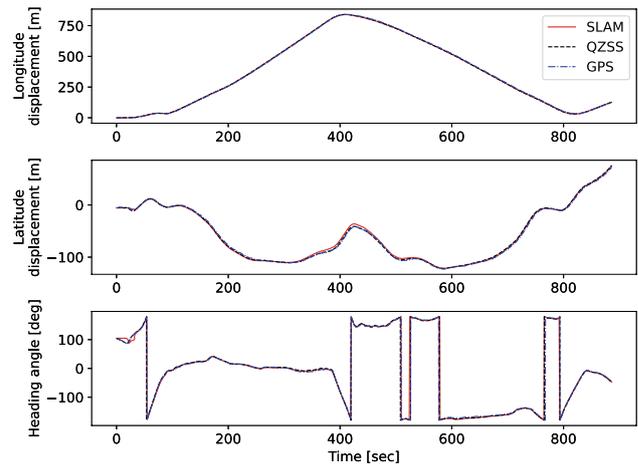

Fig. 14 Time series of the displacement estimated using SLAM (LIO-SAM) and GPS/QZSS positioning and the heading angle estimated using SLAM(LIO-SAM) and the GPS/QZSS compass
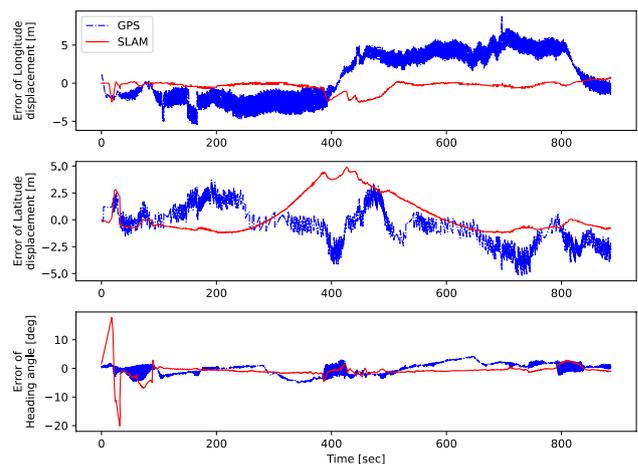

Fig. 15 Error of the displacement using SLAM (LIO-SAM) and GPS positioning versus QZSS positioning, and error of the heading angle estimated using SLAM (LIO-SAM) and the GPS compass versus the QZSS compass

able to output more stable estimates to the values output by marine GPS compass. In particular, during the three turning during the measurement, both SLAM and GPS had noise in the heading angle data, but SLAM estimated with the same level of accuracy as the marine GPS compass. However, the deviation at the beginning of the measurement is up to about 20 degrees, and it is recommended not to use SLAM until estimation of the heading angle is stabilized.

To further compare the accuracy of position estimation, Fig. 16 compares the errors of GPS and SLAM results based on the QZSS positioning data with the distance from the origin of the coordinates as the horizontal axis. The fact that the position error of SLAM is constant with respect to the distance from the origin on the way there and on the way back of the measurement indicates that the self-position estimation with SLAM itself is stable. In the section within 600 [m] of the origin, the error of the SLAM position compared with the QZSS position is mostly within the range of 1.3 [m]. As for GPS, when the trajectories were superimposed in Fig. 13, the GPS and QZSS trajectories seemed to agree well overall, but in fact, the position data acquired at each time point contained a large error, with the mean and variance of the error being 3.0 [m] and 1.5 [m] for the entire interval. Compared to GPS location data, SLAM-based self-position estimation tends to produce smaller variations in position data.

Finally, we will look at the distribution of errors for each data set. As usual, we will compare the GPS and SLAM results with respect to the QZSS data. Figures 17, 18, 19, and 20 show the density distribution of error of the longitude displacement, the latitude displacement, the position, and the heading angle for SLAM and GPS compared with QZSS, respectively. First, for latitude displacement, the distribution of GPS errors clearly shows two peaks, suggesting that the delay in estimation is the cause; compared to GPS,
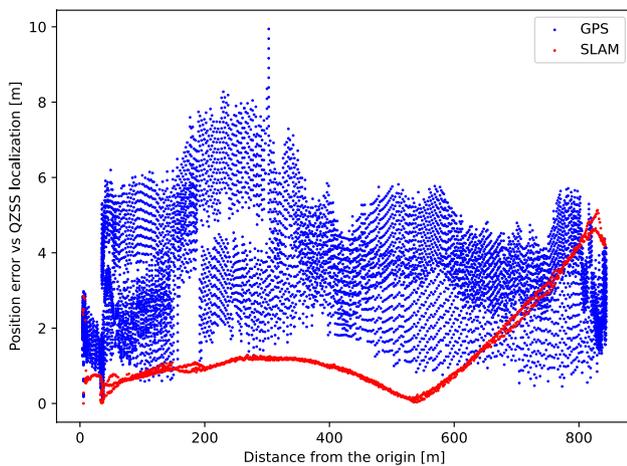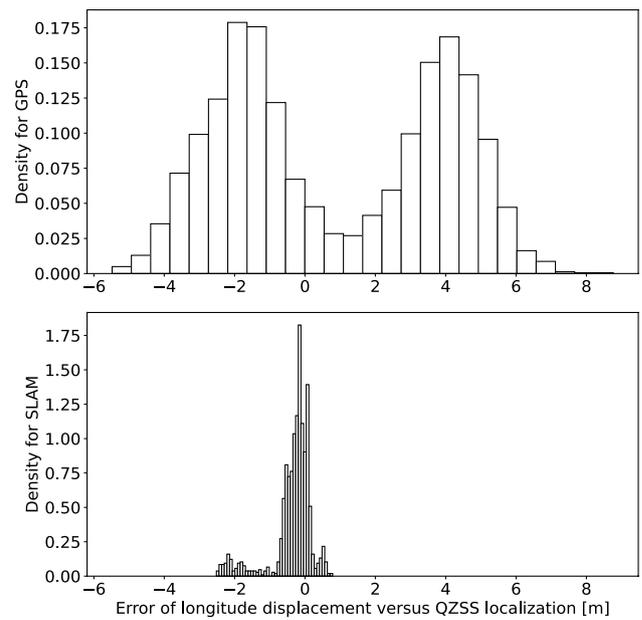

Fig. 17 Probability density of longitude displacement error of SLAM and GPS versus QZSS

the variance of SLAM latitude displacement is very small. The distribution of longitude displacement has a single peak thanks to the path of the measurement in this case. In contrast, error of SLAM distribution is hemmed on the positive side due to errors caused by map distortion. Comparing the two errors in terms of position, SLAM is affected by the error in longitude displacement, but the distribution is closer
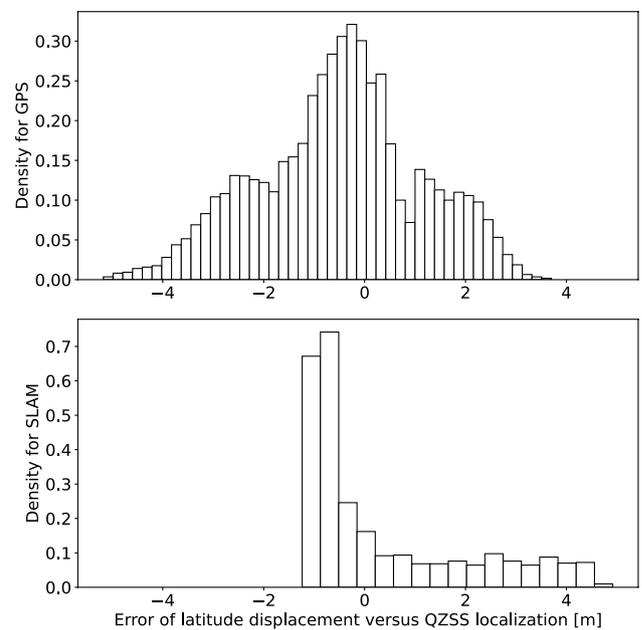

Fig. 16 Position error of SLAM and GPS vs QZSS

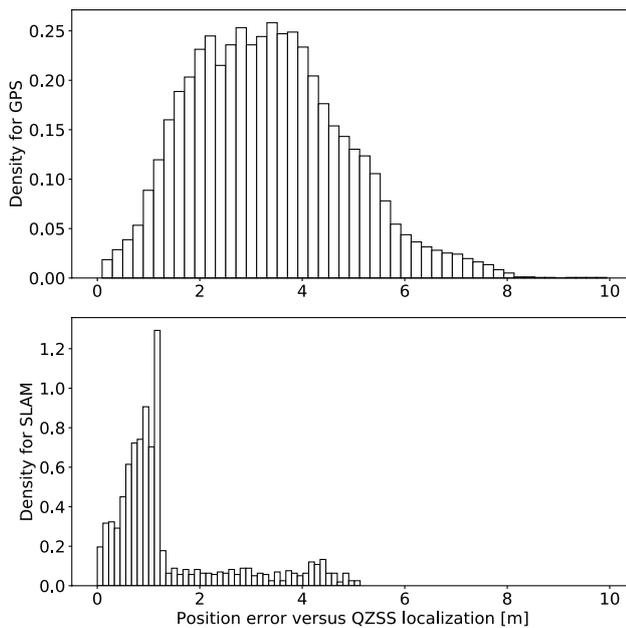**Fig. 18** Probability density of latitude displacement error of SLAM and GPS versus QZSS

Fig. 19 Probability density of position error of SLAM and GPS versus QZSS
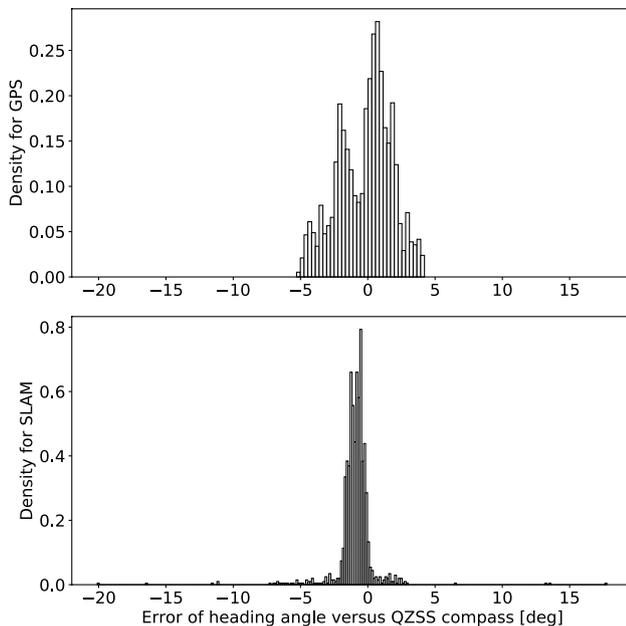

Fig. 20 Probability density of error of heading angle estimated using LIO-SAM and the GPS compass versus the QZSS compass

to the origin than that of GPS. For heading angle, the error distribution of GPS is more spread out than that of SLAM, indicating that SLAM is more accurate. The distribution of SLAM shows a slight negative bias. Table 1 summarizes the statistics of error of GPS and SLAM compered with QZSS for the longitude/latitude displacement, position, and heading angle. The values in bold represent the better one

Table 1 Statistics of error of GPS and SLAM compared with QZSS

Subject	Average μ	Standard deviation σ
Displacement		
Longitude (GPS) [m]	0.90	3.1
Longitude (SLAM) [m]	**− 0.34**	**0.59**
Latitude (GPS) [m]	− 0.50	**1.6**
Latitude (SLAM) [m]	**0.32**	1.67
Position (GPS) [m]	3.3	1.5
Position (SLAM) [m]	**1.4**	**1.2**
Heading angle (GPS) [deg]	**− 0.19**	2.0
Heading angle (SLAM) [deg]	− 0.90	**1.5**

Bold numbers indicate that the corresponding method is superior to the other

of the GPS and SLAM results. In more categories, SLAM estimates were more accurate than GPS. In actual berthing control, accuracy of position data near the pier is critical. Hence, SLAM results were also analyzed for data within 600 [m] of the pier. The average position error within 600 [m] of the pier was 0.79 [m] with the standard deviation of 0.36 [m]. For latitudinal displacement, the error averaged − 0.63 [m] with the standard deviation of 0.51 [m] within 600 [m] of the pier. These results indicate that LiDAR SLAM can satisfy the accuracy required for the automatic berthing control described in the introduction of the paper.

5 Discussion

This section provides a discussion of the application of SLAM to the actual operation of autonomous ships. In the previous section, we noted that LiDAR estimates are stable because of the small changes in estimates during the round-trip of measurements. In the actual positioning system using QZSS for automatic berthing control, the ship's position is measured while the ship is moored, and the position during automatic control is calculated using this as the zero point [1]. For LiDAR SLAM, the reliability of position measurement around the pier can be improved by setting the zero point before the start of control.

Considering the practical application of SLAM to automatic ships, performance under various weather conditions is important. Carballo et al. [27] tested LiDAR, but not SLAM, measurements of stationary targets under conditions of fog, rain, and intense light (simulating sunlight) using the Japan Automobile Research Institute's (JARI) weather experiment facility. In this study, for example, in a situation simulating fog, all LiDARs used in the experiment produced toroidal-shaped noise due to reflections from the fog and caused a reduction in reflection intensity. They also report

that in a rain situation, LiDAR sensors detect sprinkler water as a rain column. The effect of these noises of LiDAR under challenging weather conditions on the SLAM should also be discussed. For example, using only a certain range of point clouds for noise, or using only data with a certain intensity of reflection, could be a way to reduce SLAM errors under these weather conditions.

Some SLAM algorithms, including the ones used in this study, perform loop closure to correct distortion of the map. This is to correct the gap in localization that occurs when a ship is regarded as having returned to its original position after having moved some distance from the position it was in during the measurement. Loop closure is important in removing map distortion, but on the other hand, it is detrimental to control using outputs of localization because it causes the estimated self-position to jump. SLAM is therefore not used for real-time localization in actual autonomous ship applications, but rather the scan matching, which matches LiDAR-measured point clouds with pre-made maps in real time, is used for automatic control. In this paper, we have done the mapping and localization using SLAM algorithm for measured LiDAR scan. The real-time localization using scan matching and its application to automatic control will be our future work.

6 Conclusion

In this study, we performed SLAM using LiDAR at sea on the experimental ship with a view to applying it to autonomous ships.

We examined two algorithms for SLAM mapping, one using the IMU and the other not, and compared the maps created using these algorithms with the aerial image and the map of open data. The IMU mount was created with a 3D printer for stable measurements using the IMU and the LiDAR, and measurements with the QZSS receiver equipped with two antennas and the marine GPS compass were taken using the experimental ship for comparison.

The NDT-based algorithm failed to produce a map, but LIO-SAM, which integrates IMU measurements on mapping, was able to generate a map. Compared to open maps, the geometry of the floating piers and other features differed from actual conditions, indicating the importance of point cloud maps using LiDAR SLAM.

The accuracy of localization by SLAM was compared and evaluated using the marine GPS compass and the QZSS receiver. Although there were some estimation errors due to map distortion and incomplete maps at the start of the measurement, the positions and the heading angle estimated by LiDAR SLAM were both acceptable. In particular, within 600 [m] of the floating pier, LiDAR SLAM was able to estimate the position and the heading angle with the accuracy required for automatic berthing control.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Sawada R, Hirata K, Kitagawa Y, Saito E, Ueno M, Tanizawa K, Fukuto J (2021) Path following algorithm application to automatic berthing control. J Mar Sci Technol 26:541
2. Office Cabinet (2022) Quasi-zenith satellite system performance standard (PS-QZSS-003). Tech Rep
3. Office Cabinet (2022) Quasi-zenith satellite system interface specification centimeter level augmentation service (IS-QZSS-L6-004). Tech Rep
4. Kato S, Tokunaga S, Maruyama Y, Maeda S, Hirabayashi M, Kitsukawa Y, Monrroy A, Ando T, Fujii Y, Azumi T (2018) Autoware on board: enabling autonomous vehicles with embedded systems. in 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), pp. 287–296
5. Elhousni M, Huang X (2020) A survey on 3D lidar localization for autonomous vehicles. in 2020 IEEE Intelligent Vehicles Symposium (IV), pp. 1879–1884
6. Filipenko M, Afanasyev I (2018) Comparison of various slam systems for mobile robot in an indoor environment. In: 2018 International conference on Intelligent Systems (IS), pp 400–407
7. Dang T, Tranzatto M., S. Khattak, F. Mascarich, K. Alexis, M. Hutter (2020) Graph-based subterranean exploration path planning using aerial and legged robots. J Field Robot 37(8):1363 (wiley Online Library)
8. M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann, V. Reijgwart, P.D. Petris, H. Nguyen, N. Khedekar, C. Papachristos, L. Ott, R. Siegwart, M. Hutter, K. Alexis, Autonomous teamed exploration of subterranean environments using legged and aerial robots. in 2022 IEEE International Conference on Robotics and Automation (ICRA) (IEEE, Philadelphia (PA), USA, 2022)
9. Miki T, Wellhausen L, Grandia R, Jenelten F, Homberger T, Hutter M (2022) Elevation mapping for locomotion and navigation using GPU. https://doi.org/10.48550/ARXIV.2204.12876. https://arxiv.org/abs/2204.12876
10. Grisetti G, Kümmerle R, Stachniss C, Burgard W (2010) A tutorial on graph-based slam. IEEE Intell Transp Syst Mag 2(4):31
11. Ødven MS (2019) Lidar-based SLAM for autonomous ferry. Master's thesis, Norwegian University of Science and Technology
12. Kohlbrecher S, Meyer J, von Stryk O, Klingauf U (2011) A flexible and scalable slam system with full 3d motion estimation. in Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR) (IEEE)
13. J. Zhang, S. Singh (2014) LOAM: lidar odometry and mapping in real-time. In: Fox D, Kavraki LE, Kurniawati H (eds) Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014
14. Nelson E (2016) Berkeley localization and mapping. https://github.com/erik-nelson/blam. Accessed 29 June 2023

15. T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, D. Rus, LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2020), pp. 5135–5142

16. Zhou B, He Y, Qian K, Ma X, Li X (2021) S4-SLAM: A real-time 3D lidar SLAM system for ground/watersurface multi-scene outdoor applications. Auton Robot 45(1):77

17. P. Biber, W. Strasser, The normal distributions transform: a new approach to laser scan matching. in Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), vol. 3 (2003), vol. 3, pp. 2743–2748 vol.3

18. Rusu RB, Cousins S (2011) 3D is here: Point Cloud Library (PCL). in IEEE International Conference on Robotics and Automation (ICRA) (Shanghai, China, 2011)

19. R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, g^2o: A general framework for graph optimization. in 2011 IEEE International Conference on Robotics and Automation (2011), pp. 3607–3613

20. Besl P, McKay ND (1992) A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell 14(2):239

21. Shan T, Englot B (2018) LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE, 2018), pp. 4758–4765

22. Zhang J, Singh S (2017) Low-drift and real-time lidar odometry and mapping. Auton Robot 41(2):401

23. Dellaert F, Kaess M (2017) Factor Graphs for Robot Perception (Now Publishers Inc.)

24. Forster C, Carlone L, Dellaert F, Scaramuzza D (2015) IMU pre-integration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In: Robotics: science and systems XI

25. Dellaert F (2012) Factor graphs and GSTAM: a hands-on introduction. GT-RIM-CP&R-2012-002

26. Sawada R, Hirata K (2021) Path planning for automatic berthing control with waterway geometry. In: Proceedings of Japan society naval architects and ocean engineers, pp. 37–40 (in Japanese)

27. A. Carballo, J. Lambert, A. Monrroy, D. Wong, P. Narksri, Y. Kitsukawa, E. Takeuchi, S. Kato, K. Takeda, LIBRE: The multiple 3D lidar dataset. in 2020 IEEE Intelligent Vehicles Symposium (IV) (2020), pp. 1094–1101