GNSS/LiDAR-SLAM with Depth Image-based Scan Matching for Waterborne Mobile Mapping

Masafumi Nakagawa[1], Nobuaki Kubo[2], Etsuro Shimizu[3]

[1]Shibaura Institute of Technology, Japan – mnaka@shibaura-it.ac.jp
[2]Tokyo University of Marine Science and Technology - nkubo@kaiyodai.ac.jp
[3]Tokyo University of Marine Science and Technology - shimizu@kaiyodai.ac.jp

Keywords: Streaming point cloud, Point Cloud Segmentation, Simultaneous Localization and Mapping, LiDAR, 3D River Mapping

Abstract

In this research, we propose a methodology to improve the performance of scan matching and point cloud segmentation for 3D mapping of urban river environments. We also focus on the integration of depth image-based scan matching and spatial segmentation using streaming LiDAR data embedded in GNSS/LiDAR-SLAM. Moreover, we conduct experiments using a waterborne mobile mapping system to verify that our methodology can improve the stability and scalability of point cloud processing and achieve high-speed processing even in measured environments that cause SLAM degeneration problems. In addition, we propose a fast object classification based on rule-based segmentation using streaming point clouds.

1. Introduction

In this paper, we focus on 3D urban river mapping for autonomous boat navigation with simultaneous localization and mapping (SLAM). SLAM enables the generation of point clouds and self-positioning data for various applications such as 3D mapping, virtual reality, augmented reality, mixed reality, UAV flight control, and autonomous vehicle control. Depending on the input data, SLAM can be classified by input data into visual SLAM (SLAM using streaming imagery), depth SLAM (SLAM using streaming depth imagery), and LiDAR-SLAM (SLAM using streaming LiDAR point clouds). In the field of surveying and construction, LiDAR has advantages such as a wider 3D measurement range and greater resistance to changes in illumination, thus, it is often used to measure large areas and objects in changing weather conditions. Moreover, LiDAR-SLAM is used to improve the efficiency of surveying in local areas using UAV-LiDAR, handheld LiDAR, and wearable LiDAR. In surveying and construction, SLAM facilitates real-time or post-processing point cloud acquisition for ICT construction and digital twinning of urban areas. For automated 3D mapping and modeling, 3D object recognition methodologies based on model fitting and machine learning are applied using dense point clouds acquired by SLAM processing. In these topics, scan matching and point cloud segmentation are key processes for tasks in 3D mapping, scan-to-BIM, and autonomous robot operations. Segmentation techniques are divided into image-based and 3D-based approaches, each with unique advantages and disadvantages.

SLAM can be categorized into three main approaches: Bayesian filter-based SLAM, graph-based SLAM, and scan matching-based SLAM. The first approach, SLAM with Bayesian filters, includes methodologies such as Rao-Blackwellized particle filter (RBPF)-SLAM (Murphy, 2000) and extended Kalman filter (EKF)-SLAM (Weingarten et al., 2005). Each methodology has technical issues such as poor scalability and limitations to 2D processing. The second approach, graph-based SLAM, optimizes networks of sensors and feature positions through non-real-time processing. Although graph-based SLAM can estimate accurate pose and point clouds, this approach is unsuitable for 3D urban river mapping because the edges of the graphs are too long for accurate pose estimation. The third approach, scan matching-based SLAM, is a real-time processing method that aligns point clouds through optimization computations. This approach can be further divided into methodologies that require initial values, such as the iterative closest point (ICP) algorithm (Chen et al., 1992) and the normal distribution transform (NDT) algorithm (Biber et al., 2003), and methodologies without initial values, such as the globally optimal ICP algorithm (Yang et al., 2016), which ensures global optimization without the need for an initial position. We focus on scan matching-based SLAM with the ICP or NDT because we use GNSS positioning data as reliable initial values in this study. The ICP algorithm is a methodology for point cloud matching that iteratively calculates the relative position and orientation to minimize the distance between the base and corresponding point clouds, as shown in Figure 1. However, when the quality of initial values is low, the ICP algorithm tends to remain in a local optimum. The NDT algorithm approximates the set of points within each grid using a normal distribution, which reduces the processing cost based on a grid-based approach.

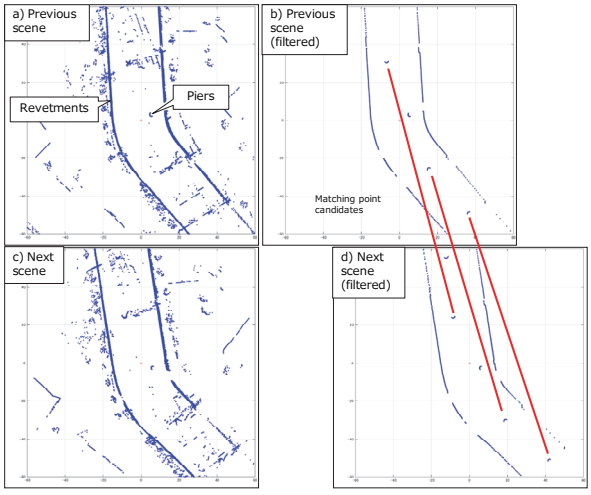

Figure 1. Example of ICP and LiDAR-SLAM

In this study, we apply the ICP algorithm because we focus on using GNSS data for precise initial values, eliminating parameter settings such as grid size optimization, and achieving higher accuracy in point cloud matching. However, conventional ICP algorithms have technical issues, including high computational cost depending on point cloud volume, sensitivity to spike noise in local optimization, and SLAM degeneration problem, as shown in Figure 2. To address these limitations, we improve the computational cost and robustness to spike noise and SLAM degeneration.

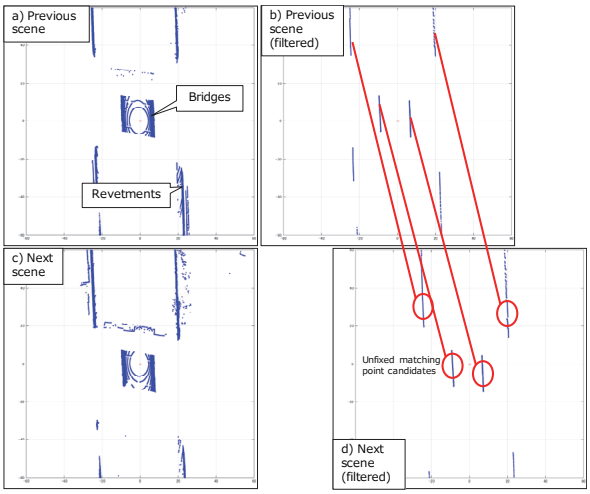

Figure 2. Example of SLAM degeneration

In this research, we propose a methodology to improve the performance of scan matching and point cloud segmentation for 3D mapping of urban river environments. We also focus on the integration of depth image-based scan matching and spatial segmentation using streaming LiDAR data embedded in GNSS/LiDAR-SLAM. Through experiments with a waterborne mobile mapping system (MMS), we verify that our proposed methodology can achieve the performance improvement of simultaneous mapping with point cloud acquisition even when measured environments that cause SLAM degeneration problem. Moreover, we propose a fast object classification of streaming point clouds based on rule-based segmentation.

2. Methodology

Our methodology consists of GNSS/LiDAR-SLAM, depth image-based scan matching for LiDAR-SLAM, and spatial segmentation using streaming LiDAR data, as shown in Figure 3.

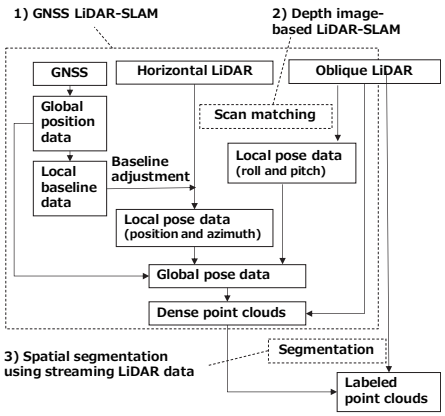

Figure 3. Methodology

The GNSS/LiDAR-SLAM is a combination of GNSS positioning and self-position estimation using SLAM with LiDAR. The GNSS/LiDAR-SLAM uses GNSS and LiDAR data for pose estimation and point cloud integration with GNSS/non-GNSS seamless positioning. Point clouds are acquired using horizontal and oblique LiDARs. For pose estimation, the horizontal LiDAR is used to estimate position and yaw angle, and the oblique LiDAR is used to estimate roll and pitch angle, based on depth image-based scan matching. The point clouds acquired by the oblique LiDAR are mainly used for point cloud integration with the estimated pose data. Moreover, labeled point clouds are generated by spatial segmentation using streaming oblique LiDAR data.

2.1 GNSS/LiDAR-SLAM

GNSS/LiDAR-SLAM is a processing that simultaneously achieves point cloud generation and seamless GNSS/non-GNSS positioning by integrating GNSS positioning with external orientation parameter estimation through LiDAR-SLAM (Nakagawa et al., 2023). The methodology includes LiDAR-SLAM based on ICP scan matching, the rectification of external orientation parameters using GNSS positioning results, and the integration of point clouds using the rectified external orientation parameters based on loosely coupled processing.

The LiDAR data for GNSS/LiDAR-SLAM is acquired using 3D multi-layer LiDAR, which is capable of acquiring point clouds with multiple cross-sectional information. Our proposed GNSS/LiDAR-SLAM which achieves seamless GNSS/non-GNSS positioning follows a loosely coupled processing integrating GNSS data as reference points and LiDAR-SLAM results as inertial data. In this framework, GNSS position data are used for data rectification of accumulated errors in self-position and attitude estimates derived from LiDAR-SLAM. The GNSS/LiDAR-SLAM eliminates the need for loop closure to adjust accumulated errors, thus, this methodology is suitable for measurements in rivers where loop closure is impractical.

Precise point positioning (PPP) using the centimeter-level augmentation service (CLAS) is applied as real-time kinematic PPP (RTK-PPP) (Mimura et al., 2020) for global position data acquisition in GNSS/LiDAR-SLAM. While PPP-RTK positioning with CLAS provides slightly lower accuracy compared to RTK-GNSS positioning, it is limited to use within Japan. However, the PPP-RTK methodology using CLAS offers advantages in terms of availability and convenience, as positioning is possible without reference station installation, continuous communication with reference stations, and the consideration of baseline length constraints to maintain positioning accuracy.

2.2 Depth image-based scan matching for LiDAR-SLAM

In LiDAR-SLAM, many conventional methods use a scan matching processing that matches paired scenes using ICP or NDT processing. Paired scenes are required to have sufficient geometric overlap, and point-to-point, point-to-line, and point-to-plane are applied in the scan matching processing. In the scan matching processing, initial values for the rotation matrix and translation vectors improve the stability of the processing. In LiDAR inertial odometry (Gao, et.al, 2024), IMU data are used as initial values for a tightly-coupled position and orientation estimation processing with scan distortion rectification processing and point-to-plane scan matching using multiple planes with different normal vectors, or scan-to-map. The GNSS/LiDAR-SLAM also uses GNSS positioning data acquired in a GNSS positioning environment as initial values

for the scan matching processing. Although both the LiDAR inertial odometry and GNSS/LiDAR-SLAM approaches use initial values to streamline the scan matching processing, the final accuracy and overall processing speed of the scan matching processing depends on the performance of the corresponding point finding processing. The ICP algorithm evaluates the distances between the base and corresponding point clouds. Even if the number of point clouds is large, a fast processing can be achieved if the nearest neighbor search is performed only on sample points. However, the ICP algorithm has a technical issue of high computational complexity of the nearest neighbor point search. The NDT algorithm approximates the distribution of points belonging to each voxel as a Gaussian distribution and evaluates the likelihood with the corresponding point cloud. The voxel size setting when converting point clouds to voxels has a significant impact on the processing performance. Based on these technical issues, we proposed a methodology to improve the accuracy and processing speed of the scan matching processing by omitting the neighborhood search processing and the cell size determination processing by depth image-based scan matching, which consists of depth image generation and subtraction, as shown in Figure 4.

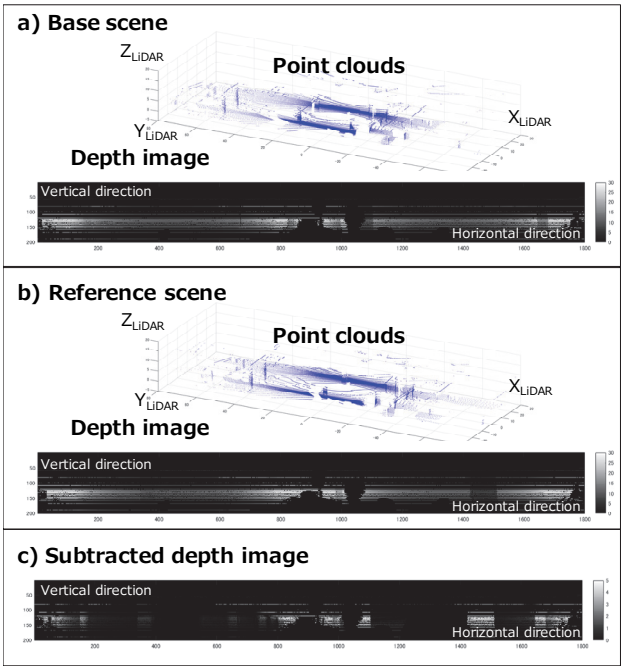

Figure 4. Depth subtraction in depth image-based LiDAR-SLAM

The processing flow of depth image-based scan matching for LiDAR-SLAM is shown in Figure 5. First, a base depth image is obtained by using the scan angle in the LiDAR coordinate system to project a set of points onto an image with the vertical scan angle as the vertical axis and the horizontal scan angle as the horizontal axis. The resolution of the image is determined by the scan angle resolution of the LiDAR, which solves the technical issue of determining the resolution of the voxel size in the NDT algorithm. Next, the corresponding depth image is generated by back-projecting the corresponding point clouds after rotation and translation to the base depth image. By applying an image subtraction between the base depth image and the corresponding depth image, the neighborhood point search problem becomes an image subtraction problem. This reduces the computational load by simplifying the 3D distance minimization to the 1D distance (depth) minimization between the base point clouds and the corresponding point clouds. This idea also eliminates the neighborhood point search processing, which is a technical issue of the ICP algorithm. Our methodology requires repeated generation of depth images due to the amount of searching for the rotation and translation parameters of the corresponding point clouds. Thus, the depth image generation process is a bottleneck step in our proposed methodology. However, a rapid rasterizer that geometrically simulates for point cloud self-occlusion (Nakagawa, et.al, 2014) is applied to speed up the overall processing.

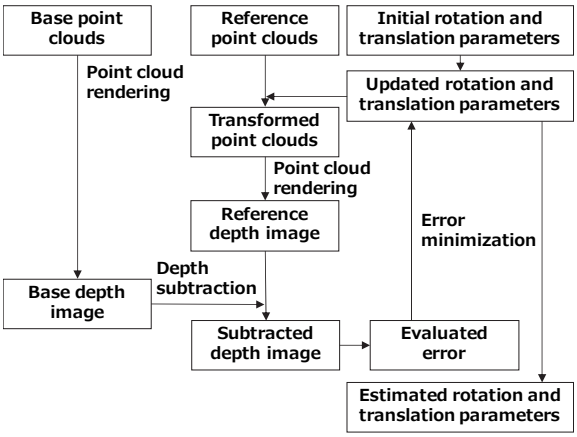

Figure 5. Depth image-based scan matching for LiDAR-SLAM

2.3 Spatial segmentation using streaming LiDAR data

Research on conventional point cloud segmentation can generally be divided into two main categories: image-based segmentation and segmentation in 3D space. For image-based approaches, a variety of methodologies have been proposed, such as PointSeg (Wang et al., 2018), RangeNet++ (Milioto et al., 2019), and SqueezeSegV3 (Xu et al., 2020). PointSeg is a semantic segmentation approach suitable for road object detection. It processes LiDAR point clouds that are projected into panoramic images using a convolutional neural network (CNN). The network is built on SqueezeNet (Iandola et al., 2017), which achieves a performance level comparable to AlexNet (Krizhevsky et al., 2012).

RangeNet++ operates on depth images, while SqueezeSegV3 is designed to exploit both camera images and point clouds by analysing similarities and differences in their features. These three methodologies share common application domains, notably autonomous driving and robotics. Their common strategy is to transform point clouds into range images, enabling real-time processing.

In contrast, segmentation techniques that operate directly in 3D space include Volumetric CNN (Qi et al., 2016a), PointNet (Qi et al., 2016b), and PointNet++ (Qi et al., 2017). Volumetric CNN converts point clouds into voxel grids and applies 3D convolution after voxelization. However, this approach deletes local geometric details and incurs high computational cost. PointNet addresses the unordered point clouds by combining global and local feature extraction. It also employs T-Net, a network that learns an affine transformation to ensure invariance to rotation and translation of the input point clouds. Despite these advantages, PointNet struggles to capture local neighborhood structures. Therefore, PointNet++ introduces a hierarchical learning strategy: it repeatedly samples and groups point sets, applying PointNet to each subset to extract local features. This hierarchical design enables more accurate segmentation of complex point clouds.

The image-based point cloud segmentation methodology can improve the processing cost and time. Thus, the image-based point cloud segmentation methodology is suitable for segmenting large streaming point clouds. However, the number of geometrical features captured in a single frame depends on the scanning range. Furthermore, it is not easy to segment objects that require observation. We focused on preprocessing with simple clustering and segmentation of point clouds, which tends to improve both the processing efficiency and prediction accuracy in deep learning for point clouds. Moreover, although many previous studies have addressed the segmentation of streaming point clouds in road environments, there are no examples of for their application to streaming point clouds acquired from boats on urban rivers due to few training data. Therefore, we aimed to develop a simple segmentation methodology for streaming point clouds used for preprocessing such as an automatic annotation labeler to improve the performance of deep learning with streaming point clouds across multiple frames to map a single geometrical feature. In contrast, the 3D spatial segmentation methodology of point clouds is suitable for extracting many features from multiple frames of streaming point clouds covering large objects. However, the processing cost is worse than that of image-based segmentation. Therefore, we focus on these advantages and disadvantages, and we propose a hybrid methodology of streaming point cloud segmentation, which combines a high-speed processing function of the image-based point cloud segmentation and the function of feature extraction from point clouds in the methodology of 3D spatial segmentation of point clouds.

This paper describes segmentation using oblique 3D-LiDAR. An overview of the spatial segmentation using streaming LiDAR data is shown in Figure 6. In the segmentation of streaming point clouds, revetment detection, height clearance filtering, and the point cloud labeling are applied to the point clouds obtained in each scene. The labeled point clouds are integrated into the labeled point clouds of the subsequent scenes.

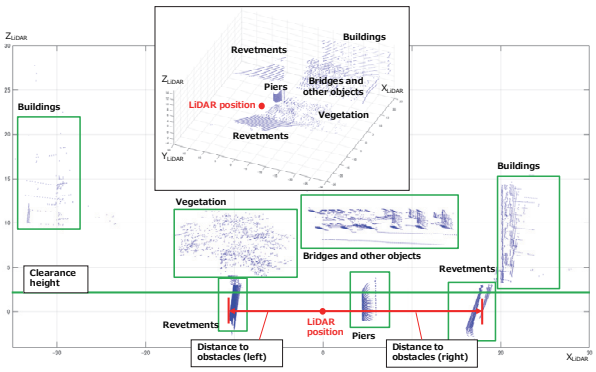

Figure 6. Spatial segmentation

Revetment detection first determines the initial horizontal position of the revetment. If the top of the revetment is higher than the LiDAR height, far points are extracted from the LiDAR on the horizontal section are extracted from each scan line. Next, revetment lines are detected using the extracted far points by line extraction with robust estimation such as least median squares. The detected revetment lines are used to extract objects, such as revetments, and obstacles. Revetments are extracted using buffered revetment lines, as shown in Figure 7. Moreover, normal vectors of point clouds are also estimated with range image-based segmentation using optical vectors from LiDAR to remove noise such as bridges and vegetation.

Then, obstacles are extracted from point clouds in or around the water area using clearance filtering. The obstacles are divided into two types of objects. The first obstacles are objects required for 2D boat navigation such as piers and other boats. The second obstacles are objects for 3D boat navigation such as girders of bridges and vegetation. The value of the clearance height depends on the size of the boat and tidal changes.

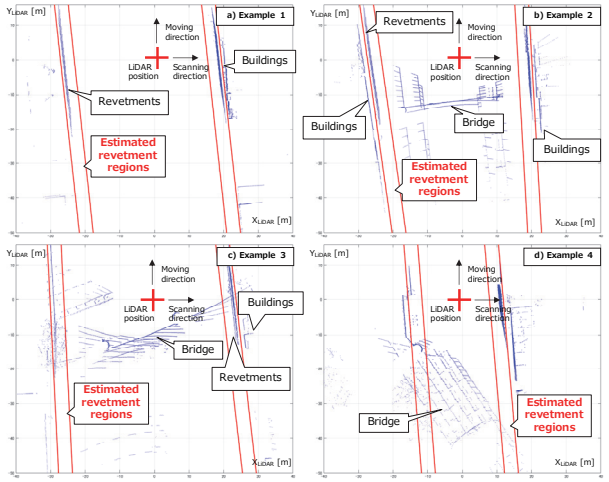

Figure 7. Examples of revetment line detection

3. Experiments

We selected the Kanda River and the Nihonbashi River as our experimental areas, as shown in Figures 8 and 9. The Kanda River is a moderate GNSS positioning environment consisting of open sky areas and non-GNSS positioning areas due to bridges. The Nihonashi River is a poor GNSS positioning environment because the almost all the sections are under the Metropolitan Expressway, as shown in Figure 10.

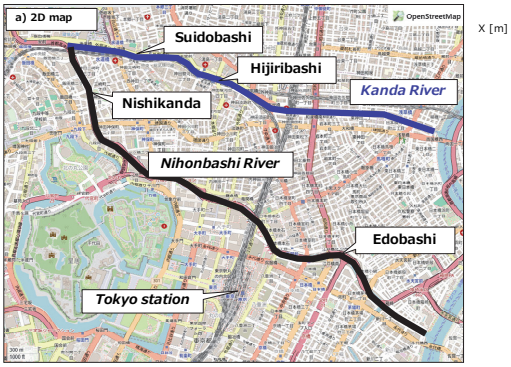

Figure 8. Experimental area

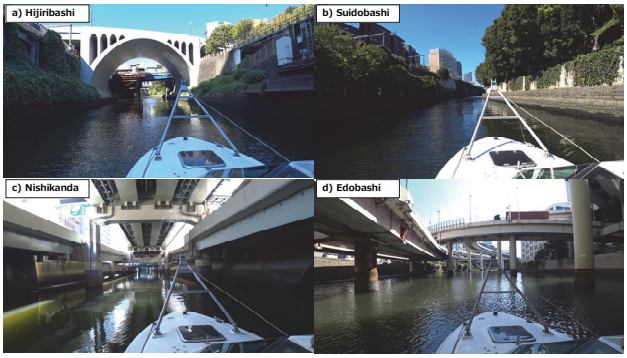

Figure 9. Photos of experimental areas

https://doi.org/10.5194/isprs-archives-XLVIII-4-W15-2025-121-2025

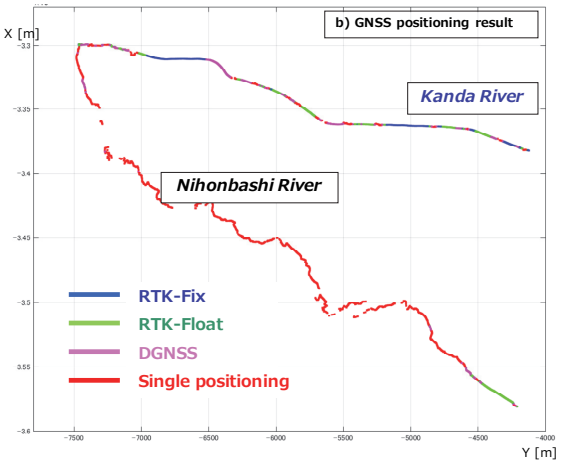

Figure 10. GNSS positioning result

We developed a waterborne MMS integrated on an autonomous battery-powered boat named Raicho I, as shown in Figure 11. For dense point cloud acquisition, and pose estimation, we used horizontal and oblique scanning LiDAR at a frequency of 10 Hz. Moreover, we used a GNSS antenna connected to a GNSS receiver for PPP based on real-time kinematic positioning with CLAS using quasi-zenith satellite systems.

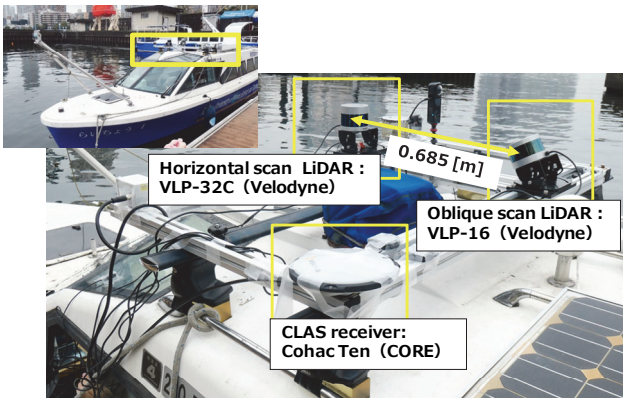

Figure 11. Waterborne MMS

We acquired two datasets, as shown in Table 1, at a navigation speed of 4 to 6 knots in a nearly windless environment when the tidal level in Tokyo Bay was low. The first dataset was mainly used to evaluate the performance of GNSS/non-GNSS seamless positioning and underwater mapping (Nakagawa et al., 2024). Both datasets were mainly used to evaluate scan matching in GNSS/LiDAR-SLAM processing. We describe our results in this paper using LiDAR data in both datasets. The number of streaming point clouds acquired is shown in Table 2.

Table 1. Datasets

Dataset ID	1		2	
Date and time	10:00-11:30, October 13, 2023		10:30-11:30, October 28, 2024	
Front LiDAR	Oblique scanning LiDAR	VLP-32C (Velodyne)	Horizontal scanning LiDAR	VLP-32C (Velodyne)
Rear LiDAR	Horizontal scanning LiDAR	VLP-32C (Velodyne)	Oblique scanning LiDAR	VLP-16 (Velodyne)
GNSS	CLAS (PPP-RTK) RTK-GNSS	AsteRx4 (CORE) ZED-F9P (u-blox)	CLAS (PPP-RTK)	Cohac Ten (CORE)
Sonar	Multi-beam scanning sonar	BV5000 (Teledyne BlueView)		

Table 2. Acquired LiDAR data

Dataset ID	1	2
The number of frames	38,501	33,001
The number of point clouds (horizontal scanning)	1.6 billion points	1.3 billion points
The number of point clouds (oblique scanning)	1.1 billion points	0.5 billion points
The number of point clouds (total)	2.7 billion points	1.8 billion points

4. Results and discussion

The processing time for GNSS/LiDAR-SLAM was 0.08139 [second/frame] (3399 seconds for dataset 1, and 2421 seconds for dataset 2. Processing environment: CPU: Intel Core Ultra 7 (3.80 GHz), RAM: 32 GBytes) for the generated point clouds (Figure 12). We confirmed that the processing time was higher than 10 Hz of LiDAR data acquisition. Moreover, average error of scan matching was approximately 0.02-0.03 m/frame. Thus, we confirmed that the scan matching accuracy reached the same value as the LiDAR range accuracy of 0.03 m (catalog value).

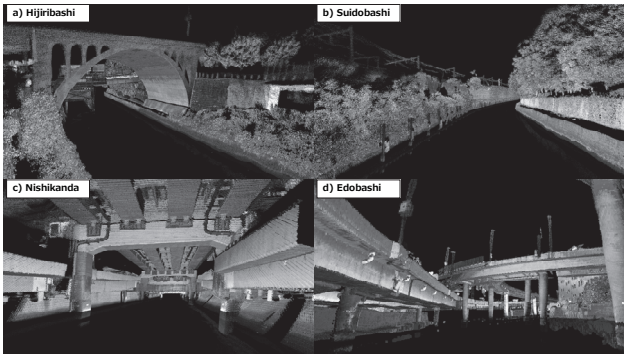

Figure 12. GNSS/LiDAR-SLAM processing results

Figures 13 shows the comparison result of the scan matching performance. In our comparative evaluation, the point-to-point based ICP matching was applied to the scan matching as the conventional LiDAR-SLAM. The conventional LiDAR-SLAM result shows that the pose estimation was failed due to the SLAM degeneration problem although the scan matching error was about 0.03 m. Figures 13 also show that the conventional LiDAR-SLAM estimated smooth poses, however the integrated point clouds were distorted. Thus, the estimated poses contained errors caused by the SLAM degeneration problem. In contrast, the result of our proposed methodology shows that the SLAM degeneration problem was avoided so that the shapes and boundaries of revetments, bridges, and buildings are clearly represented, as shown in the visualized result using all scan data and integrated point clouds.

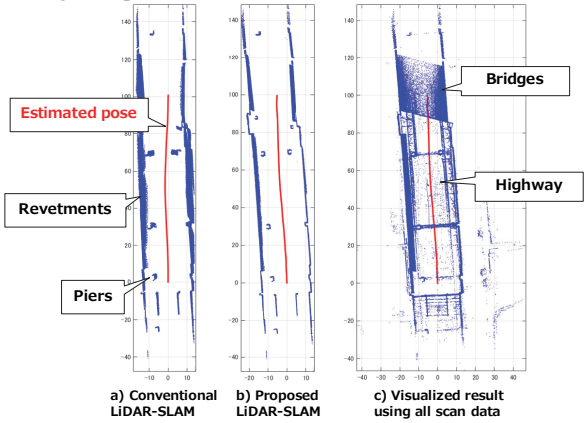

Figure 13. Pose estimation result

Figure 14 shows spatial segmentation results generated from streaming LiDAR point clouds. Distances to obstacles for revetment detection were automatically estimated, and the clearance height was temporarily set to 3 m above the water surface. Yellow points indicate the estimated revetments, and red points indicate the estimated obstacles for boat navigation, such as bridges, piers, and other objects. The processing time was 0.0206 [second/frame] (Processing environment: CPU: Intel Core Ultra 7 (3.80 GHz), RAM: 32GBytes).

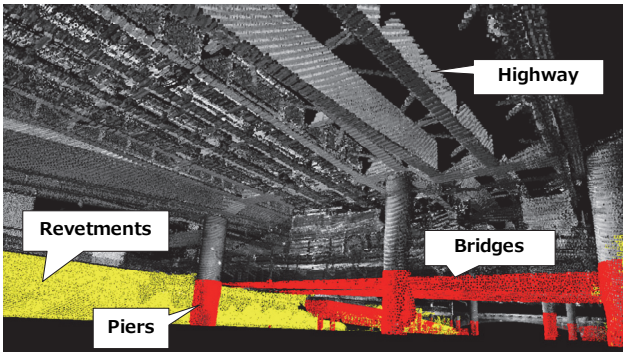

Figure 14. Spatial segmentation results

5. Conclusion

In this research, we proposed a methodology to enhance scan matching and point cloud segmentation for 3D mapping of urban rivers by integrating depth image-based scan matching with spatial segmentation using streaming LiDAR data embedded in GNSS/LiDAR-SLAM. Experiments with a waterborne MMS demonstrated improved stability, scalability, and processing speed even in measured environments that may cause SLAM degeneration problems. We also confirmed that our fast classification based on rule-based segmentation using streaming point clouds can achieve accurate classification comparable to conventional image-based and 3D-based point cloud segmentation approaches. Future work includes a higher level of point cloud segmentation for 3D modeling with level of detail for CityGML and 3D model design for real-time navigation of an autonomous boat.

References

Biber, P., Straßer, W., 2003: The Normal Distributions Transform: A New Approach to Laser Scan Matching. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol. 3, 2743–2748.

Chen, Y., Medioni, G., 1992: Object Modelling by Registration of Multiple Range Images. Image Vision, *Computing. Butterworth-Heinemann*, Vol. 10, Issue 3, 145-155.

Gao, Y., Zhao, L., 2024: VE-LIOM: A Versatile and Efficient LiDAR-Inertial Odometry and Mapping System, *Remote Sensing*, 16(15), 2772.

Iandola, N. F., Han, S., Moskewicz, W. M., Ashraf, K., Dally, J. Keutzer, W , K., 2017: SqueezeNet: AlexNet-level Accuracy with 50x Fewer Parameters and <0.5MB Model Size. *Proc. International Conference on Learning Representation*, 13 pages.

Krizhevsky, A., Sutskever, I., Hinton, E. G., 2012: ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 141, 1097-1105.

Milioto, A., Vizzo, I., Behley, J., Stachniss, C., 2019: RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4213-4220.

Mimura, D, Miyatake, K., Kubo, Y., Sugimoto, S., 2020: Positioning Accuracy of Single Frequency GNSS PPP by CLAS Comparing with MADOCA Products. *Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications*, 43-48.

Murphy, P. K., 2000: Bayesian Map Learning in Dynamic Environments. *Advances in Neural Information*, Processing Systems 12, 1015-1021.

Nakagawa, M., Yamamoto, T., Kataoka, K., Shiozaki, M., Ohhashi, T., 2014: A Point-based Rendering for 3D Polygon Extraction In Indoor Environment, *The 35th Asian Conference on Remote Sensing* 2014, 6 pages.

Nakagawa, M., Kimura, N., Komori, T., Kubo, N., Shimizu, E., 2023: Point Cloud Acquisition based on GNSS/SLAM with an Autonomous Boat in Urban River Environments, *Asian Conference on Remote Sensing 2023*, 10 pages.

Nakagawa, M., Kimura, N., Sadachika, N., Komori, T., Kubo, N., Shimizu, E., 2024: Streaming Point Cloud Segmentation of GNSS/SLAM-LiDAR and Multi-beam Scanning Sonar Data for Urban River Mapping, *Asian Conference on Remote Sensing 2024*, 19 pages.

Qi, R. C., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, J. L., 2016: Volumetric and Multi-View CNNs for Object Classification on 3D Data, *In Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE, 5648-5656.

Qi, R. C., Su, H., Mo, K., Guibas, J. L., 2016: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. Computer Vision and Pattern Recognition, arXiv:1612.00593, 19 pages.

Qi, R. C., Yi, L., Su, H., Guibas, J. L., 2017: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space., *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ArXiv:1706.02413, 5105-5114.

Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M., 2020: SqueezeSegV3: Spatially-adaptive Convolution for Efficient Point-cloud Segmentation, *European Conference on Computer Vision*, 21 pages.

Wang, Y., Shi, T., Yun, P., Tai, L., Liu, M., 2018: PointSeg: Real-Time Semantic Segmentation Based on 3D LiDAR Point Cloud. *Computer Vision and Pattern Recognition*, arXiv:1807.06288, 7 pages.

Weingarten, J., Siegwart, R., 2005: EKF-based 3D SLAM for Structured Environment Reconstruction. *Proc. IEEE/RSJ International Conference on Intelligent Robots and System*, 2089-2094.

Yang, J., Li, H., Campbell, D., Jia, Y., 2016: Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14 pages.