Efficiency Improvement of SfM/MVS by Omni-directional Camera Network Estimation for Water-borne MMS

*Teruhiko Meguro[1], Naoto Kimura[1], Masafumi Nakagawa[1],
Takeshi Komori[2], Nobuaki Kubo[2], Etsuro Shimizu[2]
[1]Shibaura Institude of Technology
[2]Tokyo University of Marine Science and Technology
*ah20092@shibaura-it.ac.jp

Abstract **:** *In recent years, the development of real-space base data for urban digital twins has advanced, driven by the need for accurate and efficient spatial data in urban environments. Two primary methods are used to generate base data in urban river spaces: LiDAR (Light Detection and Ranging) and camera-based techniques. The LiDAR method is particularly advantageous due to its ability to directly measure distance data, making the generation of point clouds relatively easy. However, the point cloud generation process in mobile mapping systems using LiDAR is highly dependent on the accuracy of the external orientation parameters, requiring high-performance GNSS (Global Navigation Satellite System) and IMU (Inertial Measurement Unit) systems, making the equipment expensive and complex. In contrast, camera-based methods, primarily using Structure from Motion (SfM) and Multi-View Stereo (MVS), offer a more cost-effective alternative for point cloud generation, although they are slower than LiDAR. Despite the longer processing time, camera-based systems are inexpensive to build and deploy. Simultaneous localization and mapping using LiDAR (LiDAR-SLAM) combined with GNSS positioning has been shown to be effective for 3D measurements in urban river environments. However, in cases where high-frequency 3D measurements or anomaly detection in civil engineering structures are required, camera-based methods offer advantages over LiDAR, such as improved flexibility and adaptability. A major challenge in wide-area camera measurements with cameras is the increased processing time due to the enormous number of captured images required to generate point clouds. To address this issue, this study proposes a method to improve the efficiency of point cloud measurements in urban river environments. This includes the use of omnidirectional cameras to increase the acquisition efficiency and the development of a method to optimize the SfM/MVS processing. Specifically, the study explores the estimation of an omnidirectional camera network that takes into account the fixed baseline relationships between cameras and the improved multi-perspective capabilities achieved through round-trip image acquisition. These innovations are expected to significantly reduce processing time while maintaining the accuracy and reliability of the generated point clouds, offering a promising approach for efficient urban river space measurements.*

Measurement keywords: Camera network estimation, Multi-view stereo, Structure from Motion, Water-borne MMS

Introduction

In recent years, the development of real-world infrastructure data for urban digital twins of cities has been promoted. In Japan, there is PLATEAU, a project led by the Ministry of Land, Infrastructure, Transport and Tourism to develop and open-source 3D city models for the

entire country. This project promotes the 3D modeling of urban infrastructure such as buildings and highways. However, in PLATEAU, the development of 3D models for river structures such as river banks, water gates, and bridges has not yet been fully implemented, which is a prominent issue in terms of the accuracy and comprehensiveness of urban digital twins. The main methods for generating infrastructure data in urban river spaces are the method using LiDAR and the method using a camera. The method using LiDAR has the characteristic that it can easily generate point clouds because it can directly measure distance data. However, the point cloud generation method used in the mobile mapping system is highly dependent on the accuracy of the external orientation, so there is a problem that the performance required for GNSS/IMU is high and the equipment is extremely expensive. In the point cloud generation using a camera, SfM/MVS is applied. Although it is inferior to the method using LiDAR in terms of the time required to generate the point cloud, the measurement system can be constructed inexpensively. It has been confirmed that LiDAR-SLAM combined with GNSS positioning is effective for the 3D measurement of urban river spaces (Nakagawa et al., 2022). However, it is difficult to obtain accurate position information using GNSS in places where the sky is blocked, such as bridges and highways. In addition, in urban environments, the need for high-frequency 3D measurements in urban environments is increasing due to the deterioration of infrastructure and the increasing risk of disasters. There are other situations where multiple measurements are needed in a short period of time, such as detecting deformations in civil engineering structures and quickly assessing the damage situation during disasters. The camera-based method can be applied to such high-frequency measurements, and allows for rapid data updates, especially in large urban river areas. However, in wide-area measurements, the increase in processing time for point cloud generation processing time due to the enormous number of images acquired is a major issue. In addition, in urban environments, the need for high-frequency 3D measurements in urban environments is growing due to the deterioration of infrastructure and the increasing risk of disasters. For example, there are an increasing number of situations that require regular and frequent measurements are required, such as detecting anomalies in civil engineering structures and quickly assessing the damage situation during disasters. For such high-frequency 3D measurements and detection of anomalies in structures, camera-based methods have advantages over LiDAR-based methods. However, when measuring large areas, a major issue is the increased processing time required to generate point clouds due to, the enormous number of images that need to be captured. In this study, we focus on the SfM/MVS method and improve it to address these issues.

Methodology

The proposed method consists of omnidirectional camera images image preprocessing, mask image generation, feature point extraction, omnidirectional camera network estimation for image pair estimation, SfM/MVS processing using the image pair estimation results, back-projected point cloud generation and error evaluation using point clouds, as shown in Figure 1.

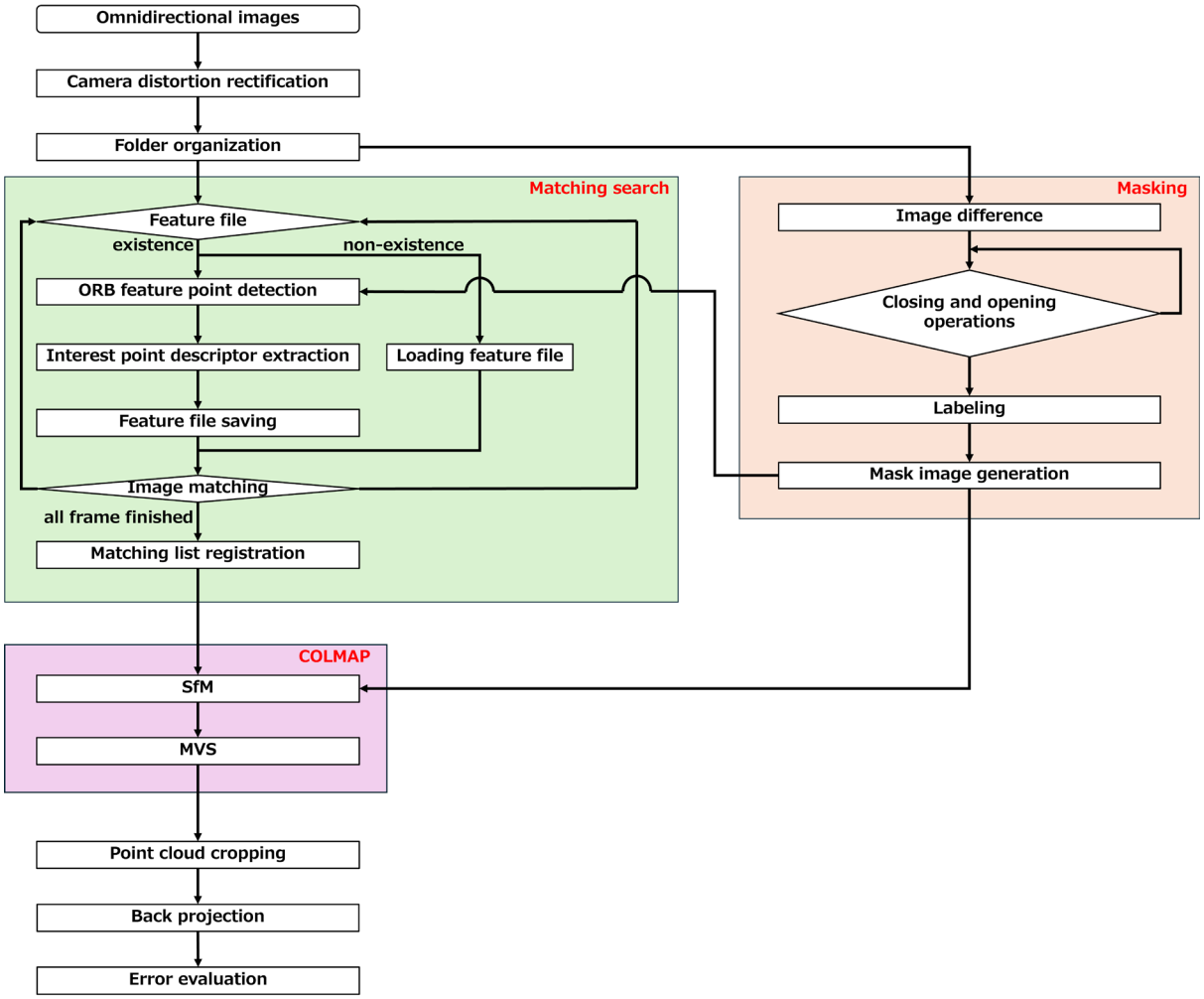

Figure 1: Proposed methodology.

As a comparison with the proposed method, an exhaustive search is applied in the matching search as conventional methods.

a. Conventional Methodology:

Image matching in SfM processing typically uses, a brute-force search to find corresponding images for each image, as shown in Figure 2.

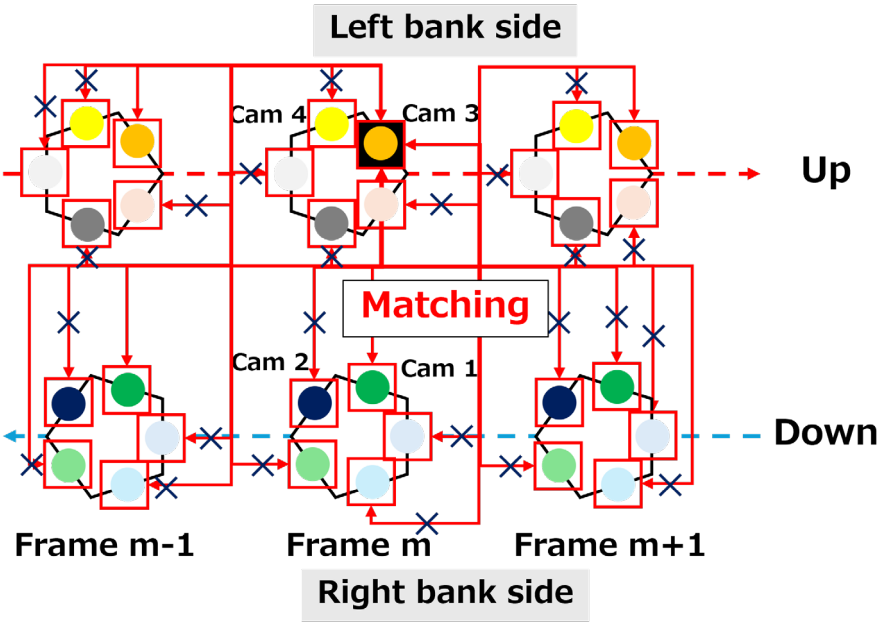

Figure 2: Brute force matching example.

However, this method requires long processing times because it searches for image pairs that cannot be matched due to the relative positions of the cameras, or the subject being photographed. In this study, we generate point clouds using brute force matching as a comparison to the proposed method.

b. Mask image generation for omnidirectional camera images:

In this study, we use an omnidirectional camera consisting of a total of six cameras. Although the omnidirectional camera has five side cameras and one top camera, the images from the five side cameras are used. The pre-processing of the omnidirectional images consists of cropping, distortion correction, and mask processing. The input images are organized based on each direction of the omnidirectional camera, as shown in Figure 3.

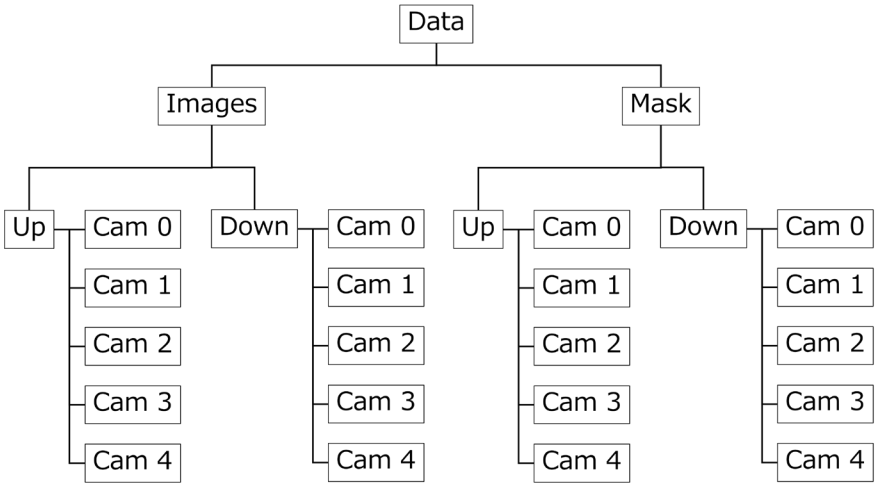

Figure 3: Folder structure.

The images used contain many areas that reflect the water surface, sky, people, and boats that serve as measurement platforms, which can lead to the detection of unnecessary feature points in the SfM processing, resulting in corresponding points (Figure 4).

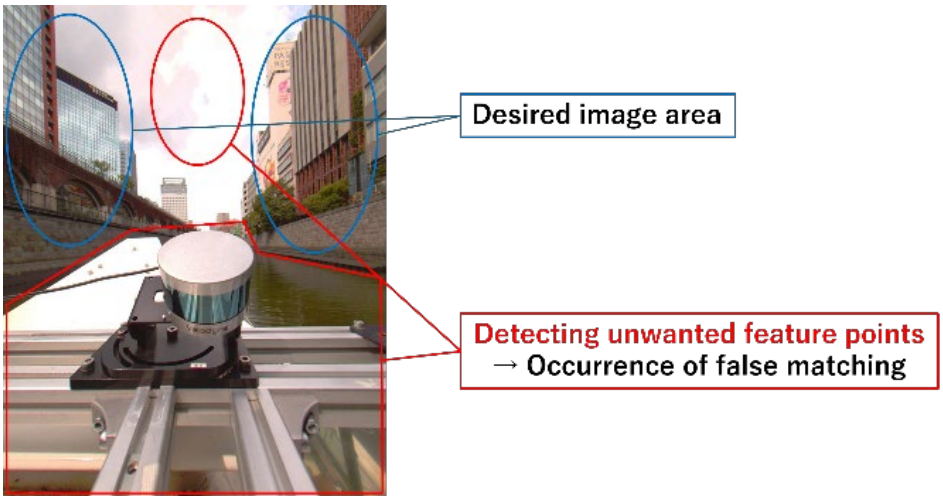

Figure 4: Image issues.

Therefore, we create a mask image for each camera image (Figure 5). By using this mask image, we can limit the area of feature point detection and reduce unnecessary image matching processing in areas where feature point matching should not be performed. The mask image is generated by performing image difference, opening, and closing, and annotation in this order. Image difference uses two different images captured by the same omnidirectional camera and detects areas in the image where the pixel difference is small by taking the difference between them. Next, an initial mask image is generated by setting a certain threshold for the difference image. However, the mask image at this stage is based on pixel-by-pixel differences, and the masked area is still sparse, with some areas being very small. For this reason, the opening and closing process is used to more roughly specify the mask area more roughly. Opening and closing is a process that repeatedly dilates and shrinks on a binary image. In opening, the mask area can be expanded by dilating the image. In closing, noise such as small mask regions present in the mask image can be removed by shrinking the image (Maxell Frontier Co., Ltd, 2021). This converts fine mask regions into coarse mask regions, which simplifies the image segmentation. However, there is a possibility that the mask image may have small areas of both white and black after processing. To solve this problem, labeling is performed. Labeling is a process of assigning labels to continuous white areas within a binary image. In this study, the number of pixels in all labeled white areas is calculated, and areas below a certain threshold are changed to black areas, and a similar process is performed on the black areas to ensure that there are no small

areas exist. The mask image generated by this series of processes is used for feature point detection and image matching.

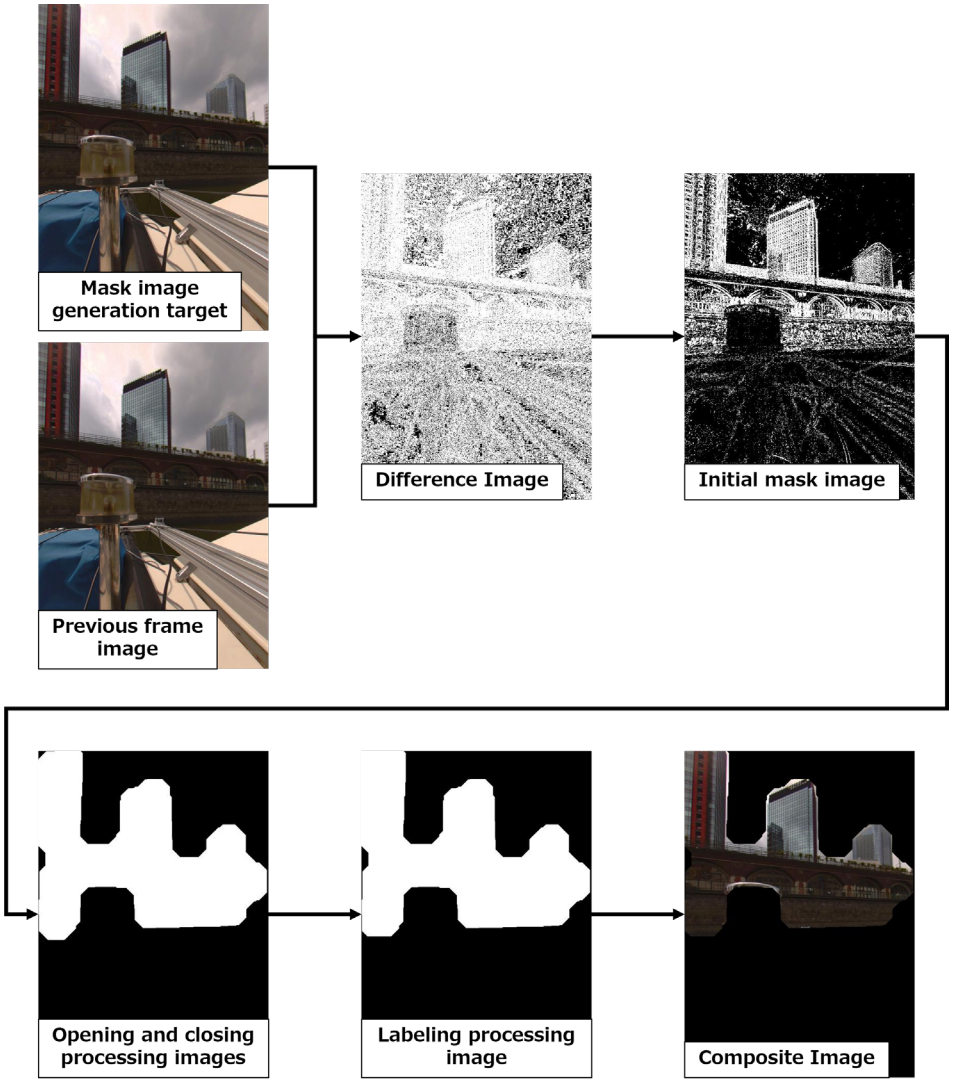

Figure 5: Masking.

c. Omnidirectional camera network estimation

When using an omnidirectional camera, the relative positional relationship between each camera is known. Consider the positions and optical axis directions of camera 1 and camera 2 in frame m as shown in Figure 6. If the positional relationship between the two images is on the same path and has the same frame number, there is a constraint. Assuming that the horizontal camera is in the n direction, the position coordinates p_{m1} of camera 1 in frame m are (x_{1m}, y_{1m}, z_{1m}), and the position coordinates p_{m2} of camera 2 in frame m are (x_{2m}, y_{2m}, z_{2m}). The optical axis vector $\overline{t_{m1}}$ of camera 1 is $(\alpha_{1m}, \beta_{1m}, \gamma_{1m})$, and the optical axis $\overline{t_{m2}}$ of camera 2 is $(\alpha_{2m}, \beta_{2m}, \gamma_{2m})$. Assuming that the cameras in the coordinate system in which the point cloud is generated are very small, the center coordinates of each camera

can be approximated to be the same, and the difference in the angle difference in the optical axis direction between each camera is known, the relationship between the position of each camera and the optical axis direction in the same frame can be expressed as follows.

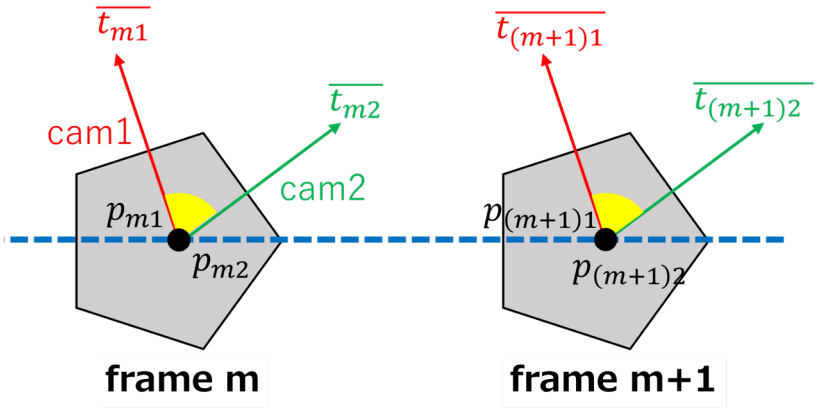

Figure 6: Camera network.

$$(x_{1m}, y_{1m}, z_{1m}) = (x_{2m}, y_{2m}, z_{2m}) \tag{1}$$

$$\begin{bmatrix} \alpha_{1m} \\ \beta_{1m} \\ 1 \end{bmatrix} = rot_\gamma \begin{bmatrix} \alpha_{2m} \\ \beta_{2m} \\ 1 \end{bmatrix} \tag{2}$$

$$rot_\gamma = \begin{bmatrix} \cos\left(\frac{2\pi}{n}\right) & -\sin\left(\frac{2\pi}{n}\right) & 0 \\ \sin\left(\frac{2\pi}{n}\right) & \cos\left(\frac{2\pi}{n}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Figure 7 shows an overview of the omnidirectional camera network estimation, which supports round-trip image acquisition.

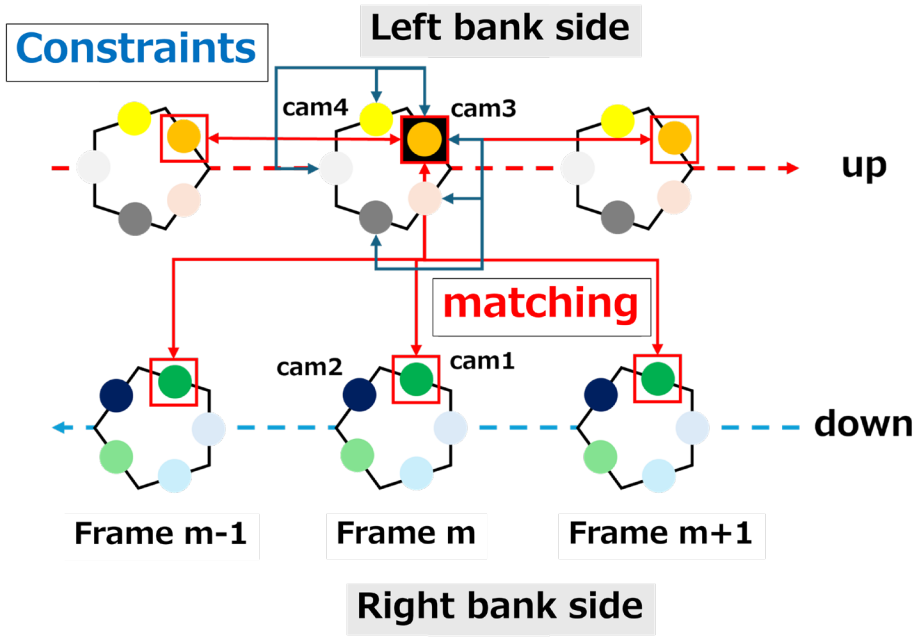

Figure 7: Camera network with round-trip images

With omnidirectional camera network estimation, the search area can be limited based on the relative positions of each camera. This prevents searching between cameras that do not match, such as when the camera directions are more than 180 degrees opposite. When focusing on two images, if there are paired images with a high matching number and strength from the shooting camera and nearby camera images, the paths of the paired images are saved in a text file as a matching list. In addition, the matching list is loaded into COLMAP, a free software for SfM, so that some of the folder paths are listed, as shown in Figure 8.

```
.
.
up/cam0/image0up_168035.jpg down/cam2/image2down_184005.jpg
up/cam0/image0up_168035.jpg down/cam3/image3down_184075.jpg
up/cam0/image0up_168040.jpg up/cam0/image0up_168045.jpg
up/cam0/image0up_168040.jpg down/cam2/image2down_184045.jpg
up/cam0/image0up_168040.jpg down/cam3/image3down_184080.jpg
up/cam0/image0up_168045.jpg up/cam0/image0up_168050.jpg
up/cam0/image0up_168045.jpg up/cam1/image1up_167960.jpg
up/cam0/image0up_168045.jpg up/cam4/image4up_167960.jpg
up/cam0/image0up_168045.jpg down/cam2/image2down_184005.jpg
up/cam0/image0up_168045.jpg down/cam3/image3down_184090.jpg
up/cam0/image0up_168050.jpg up/cam0/image0up_168055.jpg
up/cam0/image0up_168050.jpg up/cam1/image1up_167960.jpg
up/cam0/image0up_168050.jpg up/cam4/image4up_167965.jpg
.
.
```

Figure 8: Example of matching list.

d. Structure from Motion (SfM) and Multi-View Stereo (MVS)

SfM is a technology that simultaneously estimates 3D structure and camera pose from multiple 2D images. SfM is a method that simultaneously reconstructs the camera motion (Motion) and the 3D structure (Structure) of a scene by inputting multiple images from different viewpoints and tracking corresponding feature points between these images. The major advantage of SfM is that it can reconstruct 3D information from standard 2D images without the need for complex sensors or prior information. Therefore, SfM has a wide range of applications and has been used in topographic surveying by drones (Nex et al., 2014), digital archiving of cultural assets (Remondino et al., 2006), and construction of 3D spaces in AR/VR systems (Zhang et al., 2019).

Methods for extracting feature points used in SfM processing include Oriented fast and Rotated BRIEF (ORB) (Rublee et al., 2011) and Scale Invariant Feature Transform (SIFT) (Lowe, 2004). SIFT is often used in SfM processing because it has the advantage of being

independent of rotation and scale. However, it has the disadvantage that it takes time to compute the distance between feature vectors. Therefore, in this study, we use ORB, a method for expressing features as binary vectors. Feature points in the image excluding the mask image area are detected.

MVS is a technique for reconstructing a high-density scene from 2D images acquired from multiple viewpoints. MVS is known as a method to reconstruct a highly accurate 3D shape by complementing the camera positions and initial 3D point clouds obtained mainly by SfM. MVS is used in various computer vision applications because it accurately reconstructs the entire surface of a scene using the geometric relationship between corresponding viewpoints. The feature of MVS is that it can generate a high-density 3D point cloud by using additional image data for the low-density initial point cloud obtained by SfM. Therefore, MVS is widely used in fields that require detailed 3D models, such as 3D scanning, topographic surveying (Furukawa et al., 2010), urban modeling, and virtual reality content creation (Schöps et al., 2017).

SfM processing using the matching list and mask image, SfM processing using the conventional method, and MVS processing were performed using COLMAP, an open source SfM/MVS tool. This software was selected because it can detect feature points using mask images and has higher performance than other free SfM software. In COLMAP, the proposed method reads the mask image and matching list, and then generates a sparse point cloud through SIFT feature point detection, camera position estimation, and bundle adjustment. Then, based on the results, MVS processing was performed to generate a dense point cloud. On the other hand, the conventional method performed image matching by full search, and then generated a dense point cloud by MVS processing in the same way as the proposed method.

e. Back projection:

Back projection is a technique for reconstructing points in 3D space from their 2D coordinates in an image. It is the process of projecting a point on a 2D image plane back into 3D space using the intrinsic parameters of the camera. Specifically, an image coordinate $p = [u, v]^T$ is mapped to a corresponding 3D point $X_w = [x, y, z]^T$ using the intrinsic camera parameter matrix K. The intrinsic camera parameter matrix K is defined as follows:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Here, f_x and f_y represent the focal length of the camera, and c_x and c_y represent the optical center. In back projection, the three-dimensional camera coordinate $X_c = [x_c, y_c, z_c]^T$ can be calculated from the 2D image coordinate p using the following equation.

$$p = K \cdot X_c \quad (5)$$

The Z coordinate z_c represents the depth, and is usually a known or estimated value. Thus, for each 2D point in the image, the corresponding point in 3D space can be estimated. This method makes it possible to reconstruct a 3D point cloud with high accuracy from 2D images taken from multiple viewpoints. In addition, since the back projection is based on the correction of camera parameters, precise calculations can be performed that take into account the correction of lens distortion and the effects of focal length. In this study, the dense point cloud generated by the SfM/MVS processing was converted to homogeneous coordinates, and the back projection was performed using the Z coordinate value of the point cloud.

f. Error evaluation:

RMSE is an index to quantitatively evaluate the error between the actual observed value and the estimated value (Iwahori, 2015). In this study, the difference in coordinate values of the point cloud generated by back projection for the point cloud generated by SfM/MVS processing is calculated using the RMSE, and the accuracy is evaluated as the back projection error. The RMSE is defined by the following formula.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X_i^{true} - \left(R^T(X_i^{est} - t)\right)\right)^2} \quad (6)$$

Here, the true 3D point in the world coordinate system is X_i^{true}, the estimated point in the camera coordinate system is X_i^{est}, the total number of points is N, and the rotation matrix and the translation from the camera coordinate system to the world coordinate system are R^T, t respectively.

Experiment

On September 15, 2023, surface measurements were conducted from the battery-powered boat Raicho I. The boat was equipped with the first LiDAR (VLP-16, Velodyne), the second LiDAR (VLP-32C, Velodyne), CLAS receivers (AsteRx4, Septentrio), and an omnidirectional camera (Ladybug5, FLIR), as shown in Figure 9.

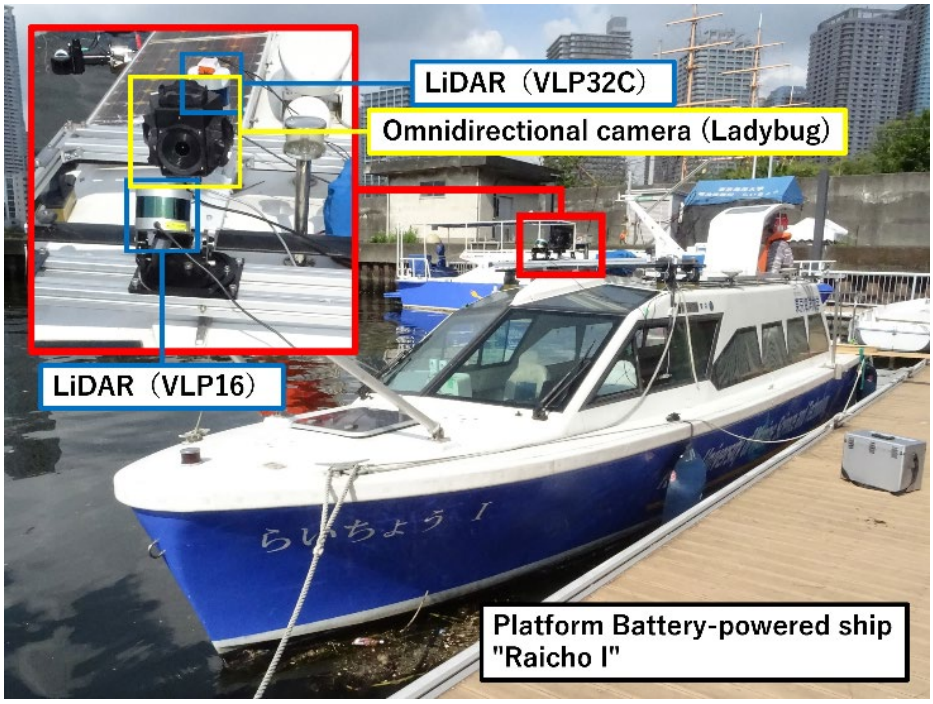

Figure 9: Water-borne MMS.

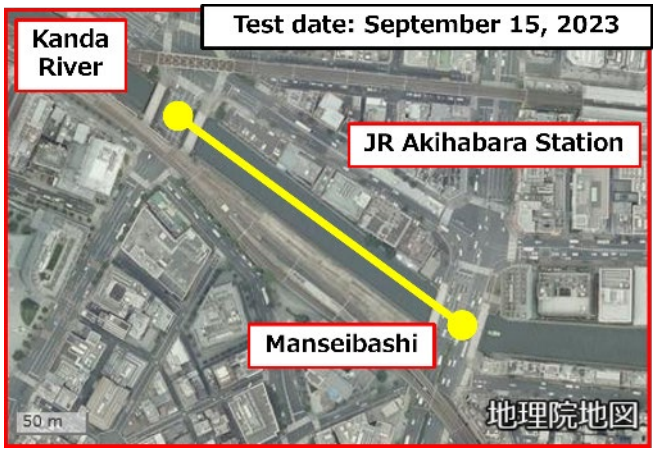

Figure 10: Experiment area.

A desktop PC was used to process the omnidirectional images (CPU: Intel(R) Core (TM) i7-11700, memory: 32GB, GPU: NVIDIA GeForce GT 1030). Of all the measurement data, 500 images near Manseibashi Bridge (Figure 10) were used for SfM/MVS processing using COLMAP, a free SfM software. In the experiment, data were collected on a round trip route

up and down the river. By masking the boat itself, the LiDAR installed on the boat, and the sky with a mask image, only the image areas to be converted into point clouds were left.

Results

The results of the image masking, image matching, and SfM/MVS processing, as well as an evaluation of each point cloud, are described below.

a. Masking images:

The total time to generate the mask images was 813 seconds, or 1.6 seconds per image.

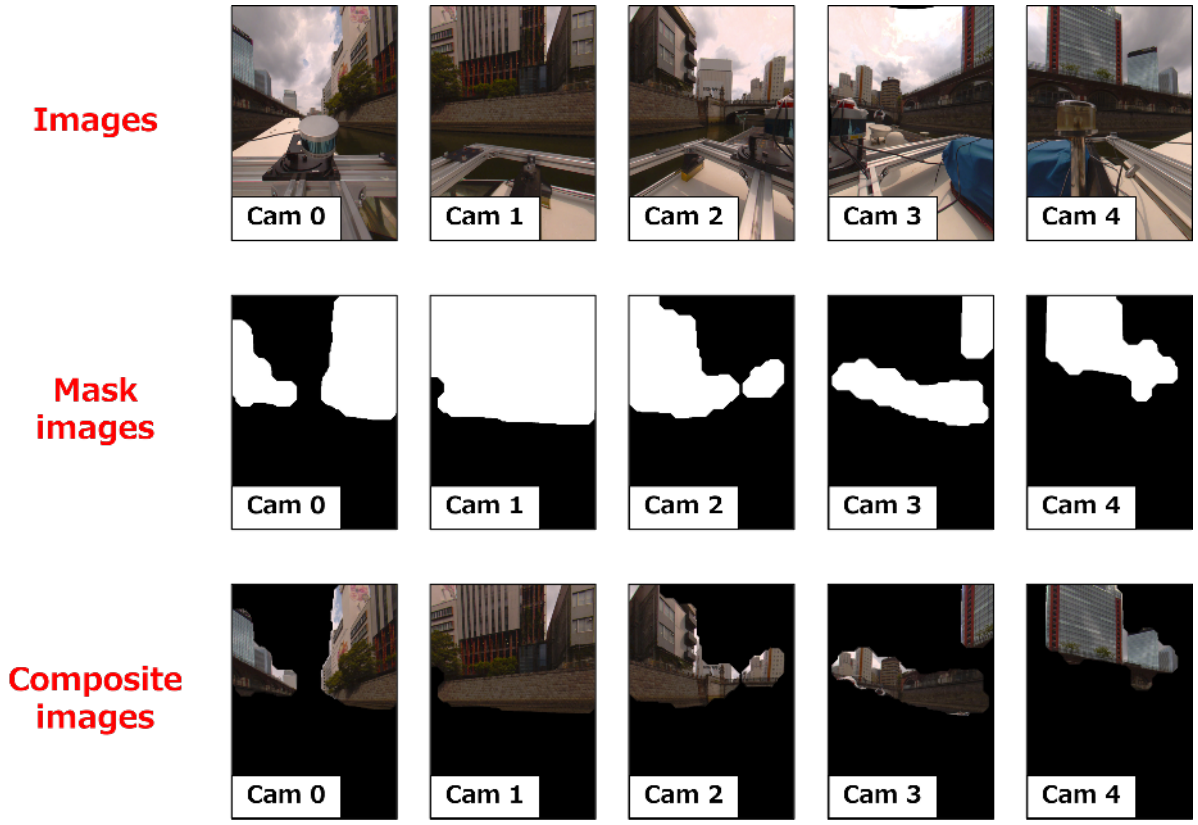

Figure 11: Example of mask processing results for omnidirectional camera images.

Figure 11 shows a composite image of the mask images from each camera and the original image in a given frame. While the overall masking of the sky, boats, and rivers was successful, there were problems with not masking small areas that were not needed to match some of the equipment installed on the boats. There was also a technical issue with masking being performed on areas that did not need to be masked, such as buildings and revetments that appeared dark in the image. This is thought to be due to the low brightness of the images, making it difficult to detect differences such as edges between the two images.

b. Image Matching Search and SfM Processing:

The image matching search and SfM processing times are shown in Table 1. In the proposed method, the frame difference for matching was calculated between each camera in the omnidirectional camera was calculated, and a small search area was set based on this information to perform the image matching search. In the image matching search, it took about 813 seconds to calculate the frame difference between images, which was longer than the matching search time, but the total image matching processing time was about 10 times that of brute force matching. On the other hand, in SfM processing using COLMAP, the conventional method used all 500 images to estimate the camera position, but the proposed method used only 222 images. This is thought to be because matching pairs could not be successfully searched for when loading the matching on COLMAP. Due to the difference in the number of images used, the number of point clouds generated by the proposed method was about 7 times smaller than that of the conventional method.

Table 1: Comparison of image matching search and SfM processing times.

			Conventional method	Proposed method
Mask image creation[s]			813.0	
SfM processing[s]	Matching search	Inter-image range search time		744.9
		Matching list creation time	10274.4	446.3
	COLMAP	Feature extraction	22.4	23.0
		Feature matching	333.4	3.5
		Resume reconstruction	1591.8	39.3
Total time of SfM processing[s]			12222	1257
The number of images			500	222
The number of points			45473	6779

c. MVS processing and point cloud evaluation:

Figure 12 shows the dense point clouds generated by MVS processing in two cases, the conventional method, and the proposed method, based on the camera position estimated by SfM processing. First, we focus on the shape of the point clouds and evaluate and compare the dense point clouds generated by the conventional method and the proposed method. The dense point cloud generated by the conventional method has a rough shape restoration overall. However, there is more noise around the river compared to the point cloud generated by the proposed method. In addition, in places such as river banks where the same texture continues for a long section, there is a problem that the generation location is generated on the opposite bank, as shown in Figure 13.

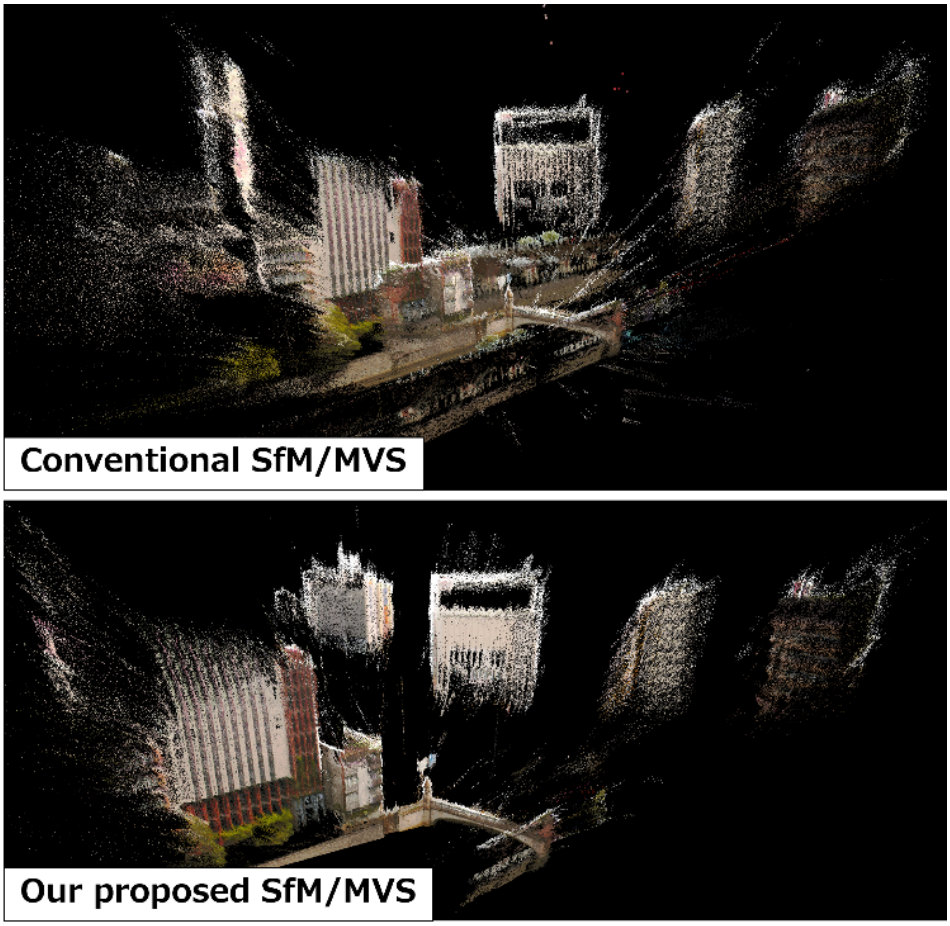

Figure 12: Dense point clouds using conventional and proposed methods.

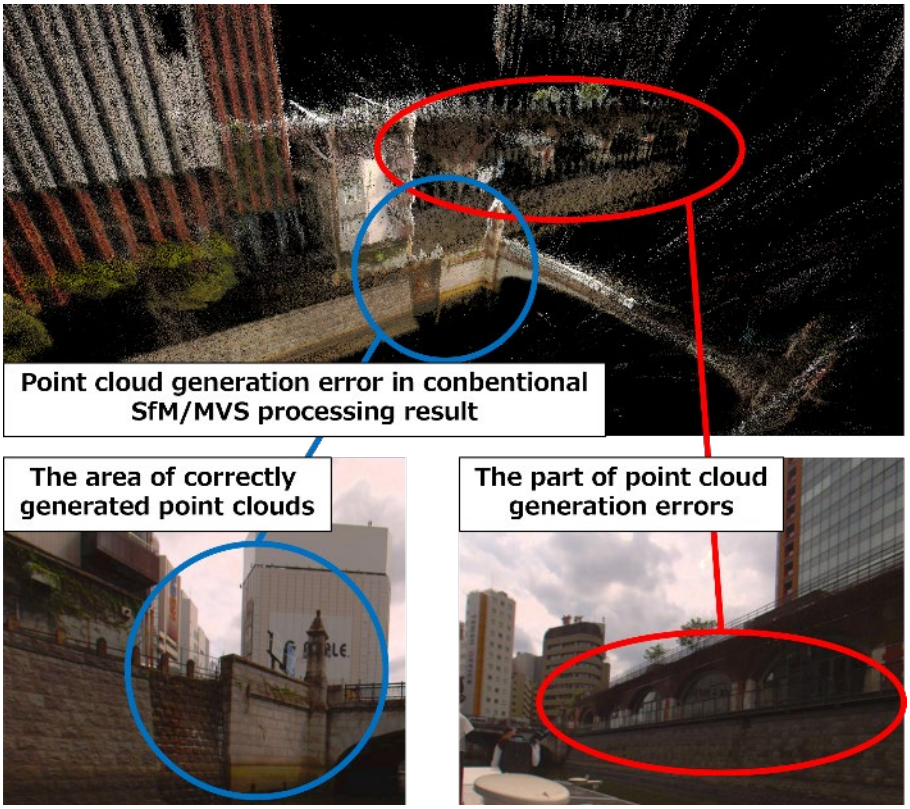

Figure 13: Wrong location for point cloud generation.

This is believed to be due to erroneous matching between pairs that are physically impossible to match because there is no constraint on the camera direction. The dense point cloud generated by the proposed method can generate buildings that are relatively far away (Figure 14) because unnecessary image matching processing can be removed in advance, and it was confirmed that the point cloud can be generated for the bank and surrounding buildings, and the shape of the structure can be fully captured. While the point cloud was generated at a level where the letters on the buildings could be read, it was confirmed that the plan shape of the interior of the buildings was not correctly generated, as shown in Figure 15.

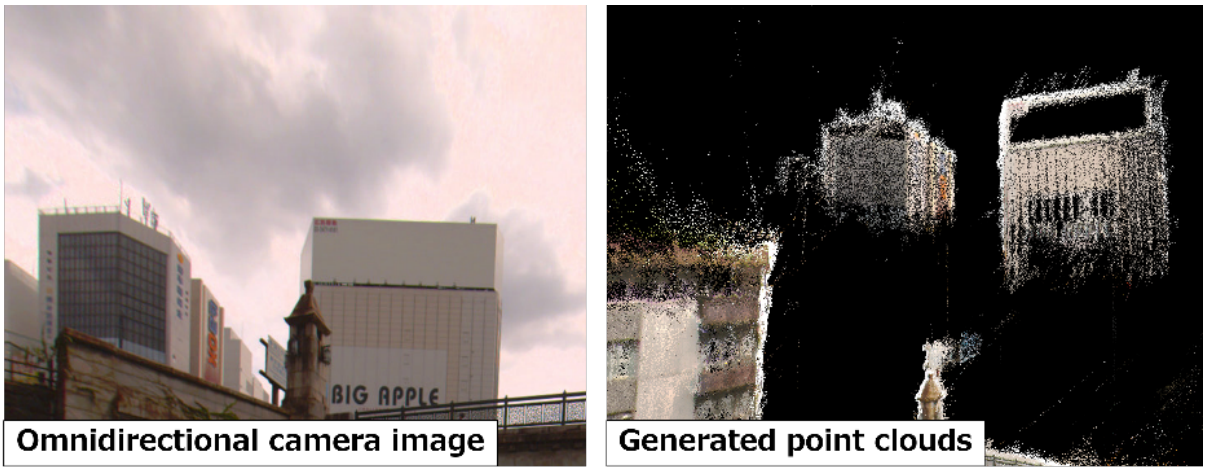

Figure 14: Point cloud generation of a building far from a river.

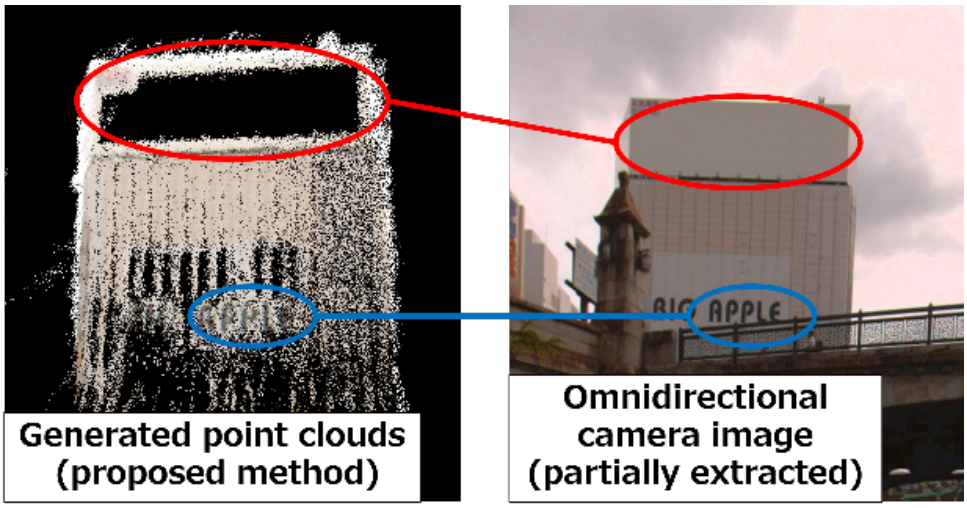

Figure 15: Building point cloud generation.

Next, we compare the results based on the processing time and RMSE. The total processing time and RMSE are shown in Table 2. In MVS processing, the conventional method took longer to process than the proposed method due to the difference in the number of matching images. When comparing the total processing time and the number of dense point clouds

generated, the proposed method was able to reduce the time by about three times compared to the conventional method. At the time of SfM processing, there was a difference of about seven times in the number of point clouds between the sparse point clouds of the conventional method and the sparse point clouds of the proposed method, but it was confirmed that the reduction could be limited to about 1.3 times with MVS processing. In the RMSE evaluation using the back-projected point cloud, we confirmed that the error was small in both cases.

Table 2: Total processing time and error evaluation.

	Conventional method	Proposed method
Mask image creation[s]	813	
SfM processing[s]	12222	1257
MVS processing[s]	40276	16359
Total time[s]	53311	18429
The number of points	2294850	1698971
RMSE evaluation[m]	0.0234	0.0266

Conclusion and Recommendation

In this study, we proposed a method to improve the processing efficiency of SfM/MVS using omnidirectional camera network estimation, which consists of efficient image matching by improving the multi-viewpoint relationship of each camera and improving the multi-viewpoint quality by shooting in both directions, using a group of omnidirectional camera images measured from a boat navigating an urban river, and compared and verified the SfM/MVS point clouds of the conventional method and the proposed method. The processing results confirmed that it is possible to establish an automatic mask generation method that is not limited by each camera and shooting location, and to improve the efficiency and accuracy of point cloud generation by using an omnidirectional camera network. Future challenges include improving the mask generation method to focus more on the image. The goal is to eliminate the small areas of sky that occurred in this study and to more accurately protect targets. In addition, there are improvements to the method and bridge inspection using a top-view camera. We aim to automatically search for image-matching search sections and automatically generate bridges based on the brightness and features between top-view camera images by using a top-view camera that was not used in this study.

References

Furukawa, Y., & Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(8), pp.1362-1376

Schöps, T., Schönberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., & Geiger, A. (2017). A multi-view stereo benchmark with high-resolution images and multi-camera videos. Conference on Computer Vision and Pattern Recognition (CVPR).

Maxell Frontier Co., Ltd. Binary Image Dilation and Erosion (2), 2021, from https://www.frontier.maxell.co.jp/blog/posts/22.html

Lowe, D. G., (2004). Distinctive Image Features from Scale-Invariant Keypoints. IJCV, Volume 60, pp.91-110

Remondino, F., & El-Hakim, S. (2006). Image-based 3D modelling: A review. The Photogrammetric Record, 21(115), pp.269-291

Hiroyuki Iwahori, (2015), New developments in 3D image sensing - from elemental technologies for real-time and high-precision sensing to industrial applications - (First edition, first printing).

Rublee, E., Rabaud, V., Konolige, K. and Bradski, G., (2011). ORB: An efficient alternative to SIFT or SURF. ICCV, pp.2564–2571

Zhang, Z., et al. (2019). Photogrammetry-based augmented reality for 3D reconstruction and virtual fitting. Computers & Graphics, 82, pp.70-80

Masafumi Nakagawa, Naoto Kimura, Tomohiro Ozeki, Nobuaki Kubo, Etsuro Shimizu, (2022). Seamless indoor/outdoor positioning using GNSS/SLAM in urban rivers. Japan Surveying Association, Applied Surveying Papers Volume 33, pp.37-46.

Nex, F., & Remondino, F. (2014). UAV for 3D mapping applications: A review. Applied Geomatics, 6(1), pp.1-15.