# Pose Correction for SfM-based River Mapping using the Fixed Baseline and Optical Axis of an Omnidirectional Camera as Constraints

Teruhiko M. [1*], Tetsu Y.[1], Nobuaki K.[2], Etsuro S.[2], Masafumi N.[1]

[1]Shibaura Institute of Technology, 3-7-5, Toyosu, Koto-ku, Tokyo 135-8548, Japan

[2]Tokyo University of Marine Science and Technology, 2-1-6 Etchujima, Koto-ku, Tokyo 135-8533, Japan

*ah20092@shibaura-it.ac.jp

***Abstract*:** *In recent years, the acquisition of dense point clouds has been promoted as on approach to create 3D urban models. However, sufficient progress has not been made in preparing data for urban river environments. There are two main methodologies for 3D data acquisition in the urban river environments: laser scanning and image-based 3D measurement. Laser scanning directly measures distances and acquires dense point clouds. However, the accuracy of point cloud acquisition depends heavily on exterior orientation estimation, which requires expensive GNSS/IMU systems. On the other hand, image-based point cloud generation primarily uses structure from motion (SfM) and multi-view stereo (MVS). Although SfM/MVS typically requires more time than laser scanning to generate point clouds, it enables the development of more affordable measurement systems. However, matching large-scale image datasets requires significant computational resources. In particular, boat-based measurements require the linear trajectories along long urban riverbanks. However, long measurement paths tend to result in accumulated matching errors and distorted point clouds. To address this issue, we propose a methodology that improves the reliability of SfM-based pose estimation using omnidirectional images captured from a boat. First, we generate masks that exclude the water surface and occlusions. This enhances, the reliability of image matching. Then, constraints are introduced into the SfM process by leveraging the fixed baseline and optical axis alignment of the omnidirectional camera. These constraints stabilize the selection of image pairs and the pose estimation by guiding the image matching process with prior knowledge of the camera arrangement. Finally, point clouds are generated using SfM/MVS, with reduced matching errors along the measurement trajectory. We also compared these results with those from LiDAR-SLAM and conventional methodologies.*

*Keywords: structure from motion, multiview stereo, waterborne MMS, omnidirectional camera, pose estimation correction*

## Introduction

In recent years, the acquisition and mapping of high-accuracy, high-density 3D point cloud data has been actively promoted worldwide to construct urban digital twins. In Japan, the Ministry of Land, Infrastructure, Transport and Tourism (MLIT) has led the PLATEAU

project [1], which develops and publicly releases nationwide 3D city models. The project has advanced the modeling of urban infrastructure such as buildings and highways. However, models of urban river structures, including revetments, water gates, and bridges, remain inadequate. This hinders the accuracy and comprehensiveness of urban digital twins. The two main methodologies for acquiring 3D data in urban river spaces are Light Detection and Ranging (LiDAR)-based and image-based measurements. Although LiDAR can directly measure distances and acquire dense point clouds, the accuracy of point cloud generation depends heavily on the exterior orientation estimation performance using expensive GNSS/IMU systems. In contrast, image-based methodologies, such as structure from motion (SfM) and multi-view stereo (MVS), take more time to generate point clouds. However, these methodologies can be implemented with relatively inexpensive equipment. These methodologies are well-suited to the high-frequency measurements required by applications such as detecting structural deformations and quickly assessing damage during disasters. Nevertheless, processing large-scale image datasets requires significant computational resources. In wide-area measurements, accumulated matching errors can cause geometric distortions in the point clouds. Furthermore, urban river environments present unique challenges. Waterborne surveys are necessary for acquiring data beneath bridges and along entire revetments. However, these surveys present several challenges, such as restrictions on navigation routes and speeds, difficulty installing multiple cameras, and positioning in non-GNSS environments. To address these issues, this study proposes an error correction methodology that uses the attitude information from both the omnidirectional camera and the boat. The methodology applies constraints based on the fixed baseline between the cameras and the optical axis direction. Unnecessary regions in the omnidirectional camera images are masked. Next, matching constraints and corrections are applied based on the position and orientation of each camera. Then, point clouds are then generated through SfM/MVS processing based on the corrected data. Finally, we evaluate the proposed methodology by comparing its results with those obtained with LiDAR-SLAM and conventional SfM/MVS methodologies.

**Related Works**

**a. SfM/MVS:**

SfM is the simultaneous processing of 3D scene reconstruction and poses (positions and orientations) estimation of cameras using 2D images captured from multiple viewpoints. SfM detects and matches feature points across overlapping images to reconstruct both

camera trajectories and point clouds representing the geometry of features. A significant advantage of SfM is its ability to generate 3D information from images without requiring prior knowledge. MVS is a processing for dense point cloud generation using SfM results. SfM/MVS is used for many applications, such as aerial photogrammetry using UAVs, 3D urban modeling, disaster monitoring, infrastructure maintenance, digital documentation of cultural heritage, and spatial modeling in augmented and virtual reality systems [2][3][4][5].

**b. COLMAP:**

This study employed SfM/MVS using COLMAP [6]. COLMAP is an open-source and high-performance SfM/MVS framework that supports a complete pipeline including scale-invariant feature transform (SIFT) feature extraction, image matching, camera pose estimation, bundle adjustment, and point cloud generation. COLMAP also allows for the use of custom inputs, such as mask images and external matching lists, offering both flexibility and accuracy. In the proposed methodology, mask images and a pre-generated matching list are prepared as inputs to COLMAP. Next, the pipeline estimates camera poses and sparse point clouds. Conventional SfM/MVS performs brute-force image matching without masks or geometric constraints. However, technical issues exist, such as long processing time and numerous matching errors. In contrast, our proposed methodology focuses on the improvement of processing speed and image matching accuracy for trajectory estimation and dense point cloud generation.

**c. Brute-force Search for Image Matches:**

A brute-force search for image matches identifies pairs of matching images among all of the captured images. This approach enables straightforward, image matching, even when the relative positions of the cameras are unknown. Thus, the brute-force search is advantageous for complex or irregular image acquisition when there are no temporal or geometric relationships between the images. However, as the total number of images increases, the number of image pairs to be matched grows exponentially, reaching $N(N-1)/2$ for $N$ images. This increases the computational cost of large-scale 3D reconstruction tasks of urban environments, for which hundreds or thousands of images typically used. Nevertheless, few overlapping images typically exist in a brute-force search for image matches. Therefore, we propose a methodology that improves computational efficiency by rejecting image pairs from a brute-force search for image matches based on the geometric

relationships of an omnidirectional camera. We compare our proposed methodology with the conventional SfM/MVS approach that uses a brute-force search for image matches.

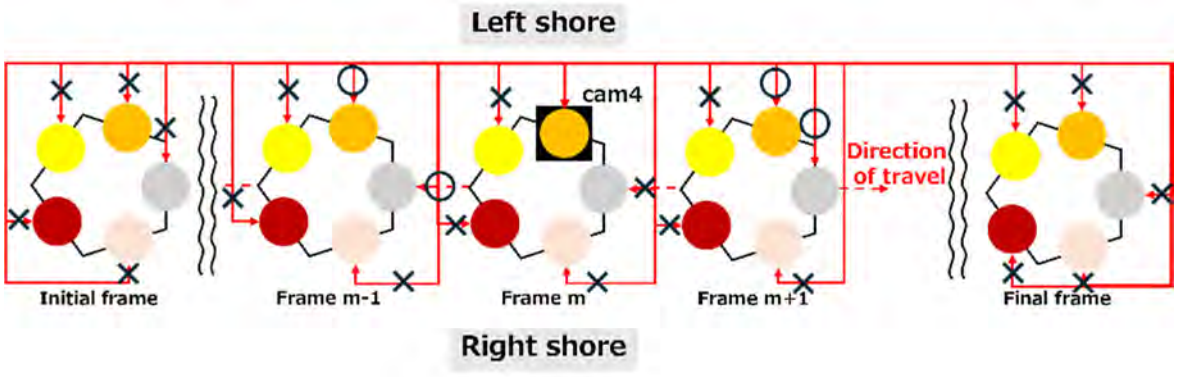


Figure 1: Brute-force Search for Image Matches.

**d. LiDAR-SLAM:**

Simultaneous localization and mapping using LiDAR (LiDAR-SLAM) is a technique that performs self-localization and map reconstruction simultaneously using LiDAR point clouds [7]. Because it is unaffected by lighting conditions or the presence of texture, LiDAR-SLAM produces more stable 3D reconstruction than vision-based methodologies. However, the performance of LiDAR-SLAM strongly depends on the surrounding geometric structure. In environments with continuously repeated features, such as river revetments, it may suffer from reduced accuracy in both localization and mapping. Moreover, scan matching is computationally intensive and can be sensitive to measurement noise. Additionally, multiple LiDAR units are often required to ensure sufficient viewpoint coverage, and the high cost of these sensors presents practical challenges to deployment.

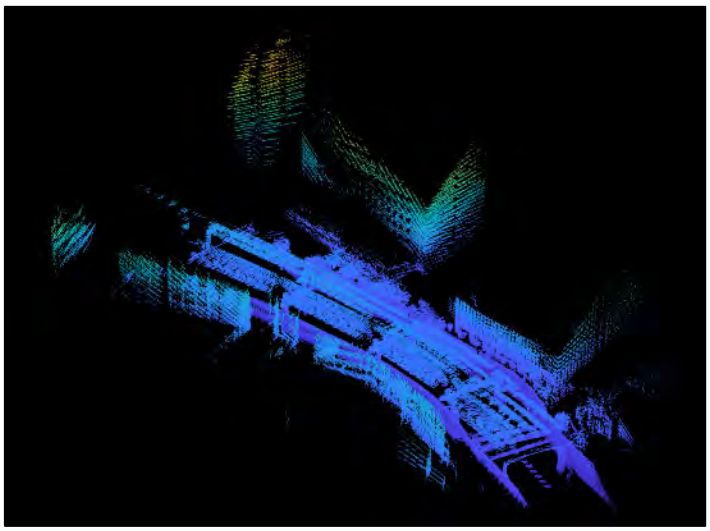

Figure 2: A Example of LiDAR-SLAM Point Clouds.

## Methodology

The proposed methodology consists of three main components: preprocessing and masking of omnidirectional camera images, creation of a matching list based on the omnidirectional camera network, and SfM/MVS processing, as shown in Figure 3.

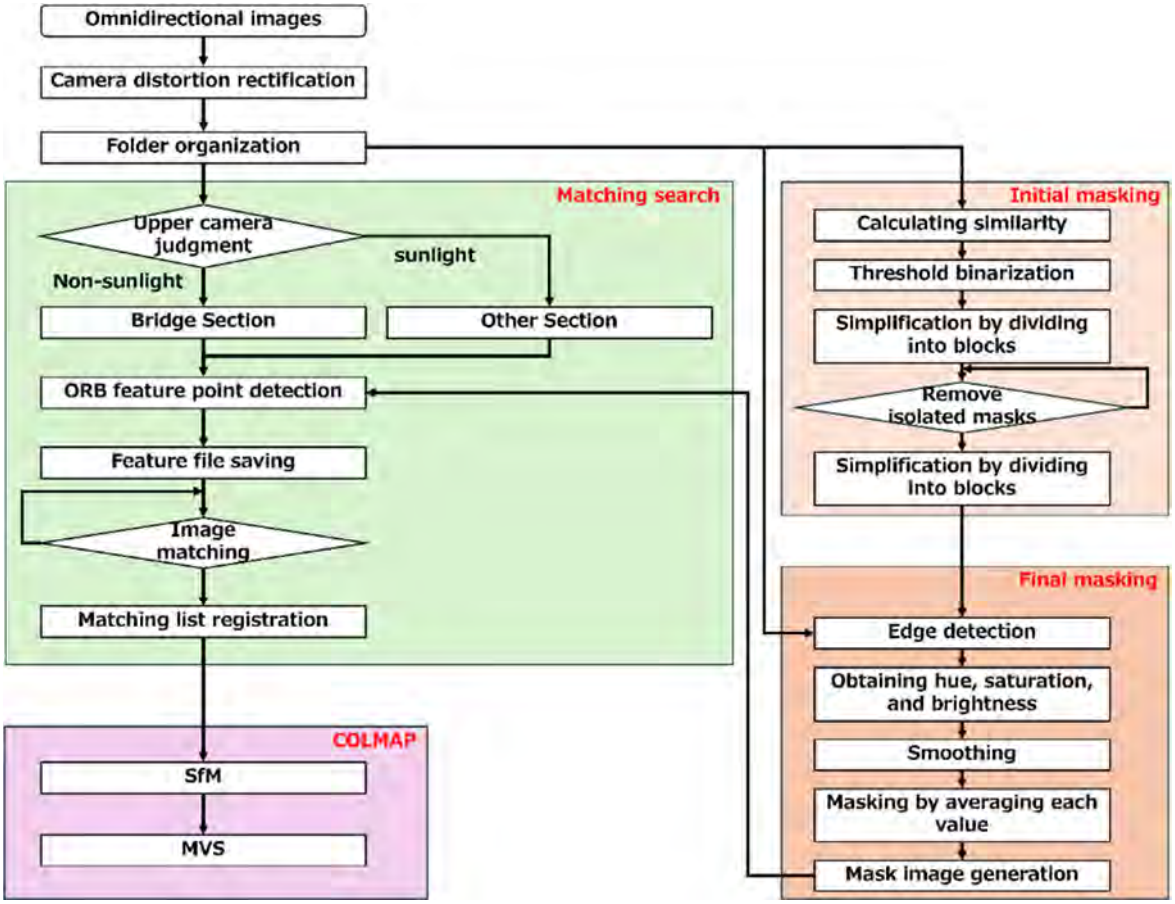


Figure 3: Proposed Methodology.

### a. Mask Generation for Omnidirectional Camera Images:

In this study, a masking process is applied as part of image preprocessing before SfM processing to improve the accuracy of 3D reconstruction in urban river environments. In particular, the process aims to remove useless regions for SfM from images, such as the sky, water surfaces, and the measurement platform. After the rejection of the regions, unnecessary feature matching can be reduced to improve point cloud accuracy. The images used in this study are captured using an omnidirectional camera, which consists of five horizontal wide-angle cameras and an upward-facing camera. The images from the five side cameras are mainly used. These images often contain not only river structures but also large areas of sky, water surfaces, and the boat itself. All of these elements may result

mismatching in SfM processing. The masking procedure consists of the following two stages, as shown in Figure 4.

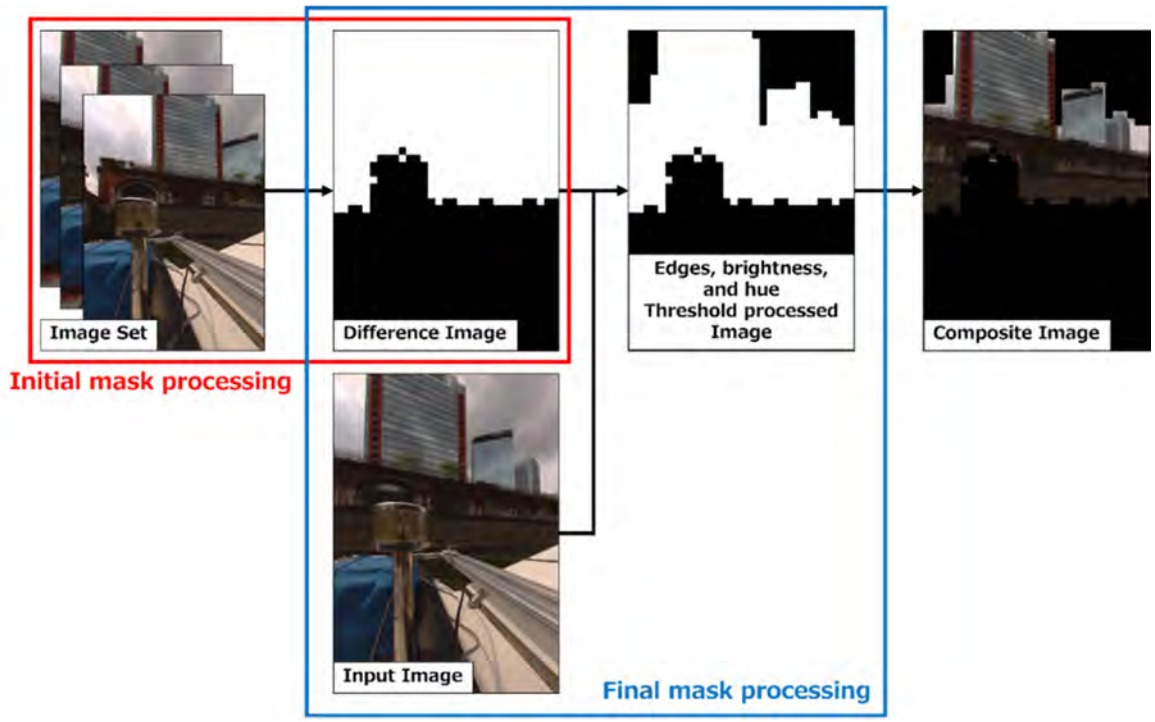


Figure 4: Masking Methodology.

First, the initial mask generation distinguishes between static and dynamic regions and exclude areas unsuitable for feature extraction. Two images, typically the first and the last frames, captured at different times for each camera are compared using the structural similarity index (SSIM) [8]. Regions with minimal change, such as static background regions, are extracted by calculating the SSIM difference between the two images. Pixels with high similarity are considered outside the region of interest for feature detection and are masked accordingly. Since the initial mask is based on pixel-level similarity, the resulting mask may be sparse and fragmented. To address this issue, the image is divided into blocks of a fixed size. Blocks with a ratio of masked pixels below a certain threshold are excluded, thereby smoothing out local inconsistencies. In addition, a labeling process removes small isolated regions, such as noise, while retaining larger, contiguous regions. Then, a neighborhood-based filtering step is applied. If three or more adjacent blocks are marked as "1" (i.e., masked), the central block is also set to "0" to eliminate edge noise. These steps generate a refined initial mask image that serves as the basis for the next stage. During the final stage of mask refinement, semantic information is added to the initial mask

to more precisely remove meaningless regions such as the sky and water surfaces. First, structural features of the image are extracted using Sobel edge detection [9]. Based on the detected edge positions, the top of the image is assumed to be the sky and is masked accordingly. This assumption is based on the fact that the upper areas of omnidirectional images often contain sky. Next, the RGB image is converted to the HSV color space. Pixels within the typical range of sky colors, such as hue, saturation, and brightness, are extracted as a blue sky mask. This enables the detection of sky regions based on both structural features and color information. Then, the two types of sky masks (edge and HSV-based) are combined and merged with the initial mask to generate the final mask. Additionally, regions with extremely high brightness, or, overexposed regions, are identified through grayscale conversion and thresholding. The regions are excluded as part of the refinement process. Finally, the image is divided into blocks once more, and a pixel ratio check is performed to smooth the mask and ensure consistency across the image. This sequence of processes generates the final binary mask image, which strictly limits the regions for feature point detection in the SfM pipeline. In summary, the masking method proposed in this study effectively eliminates irrelevant regions from the large quantity of images obtained from omnidirectional cameras. The proposed methodology contributes to efficient and highly accurate 3D reconstruction by optimizing the feature detection process in SfM.

**b. Dynamic Matching Range Control for Multi-View Images in SfM-Based Reconstruction:**

This study proposes an efficient and accurate methodology for feature point matching in large-scale image datasets acquired from multiple viewpoints, as shown in Figure 5. The core of the methodology is local matching between adjacent frames within the same camera. For inter-camera matching, the search is automatically limited to a limited range before and after the reference frame based on the frame offset corresponding to the greatest number of matches found during an initial search. This strategy reduces redundant matching attempts while balancing computational efficiency and matching accuracy. In addition, for images captured by the upward-facing camera, the brightness distribution across pixels is analyzed to identify bridge regions. Based on this analysis, the images are classified as captured under bridges or in open environments, as shown in Figure 6. Separate matching lists are then generated for each group to account for variations in lighting conditions such as shadows and illumination changes caused by overhead structures. This classification enables the optimization of matching processes specific to the structure, suppressing false

correspondences and improving the geometric accuracy of the point cloud generation. The proposed methodology dynamically controls the matching search space to enable efficient, high-precision 3D reconstruction through SfM/MVS processing.

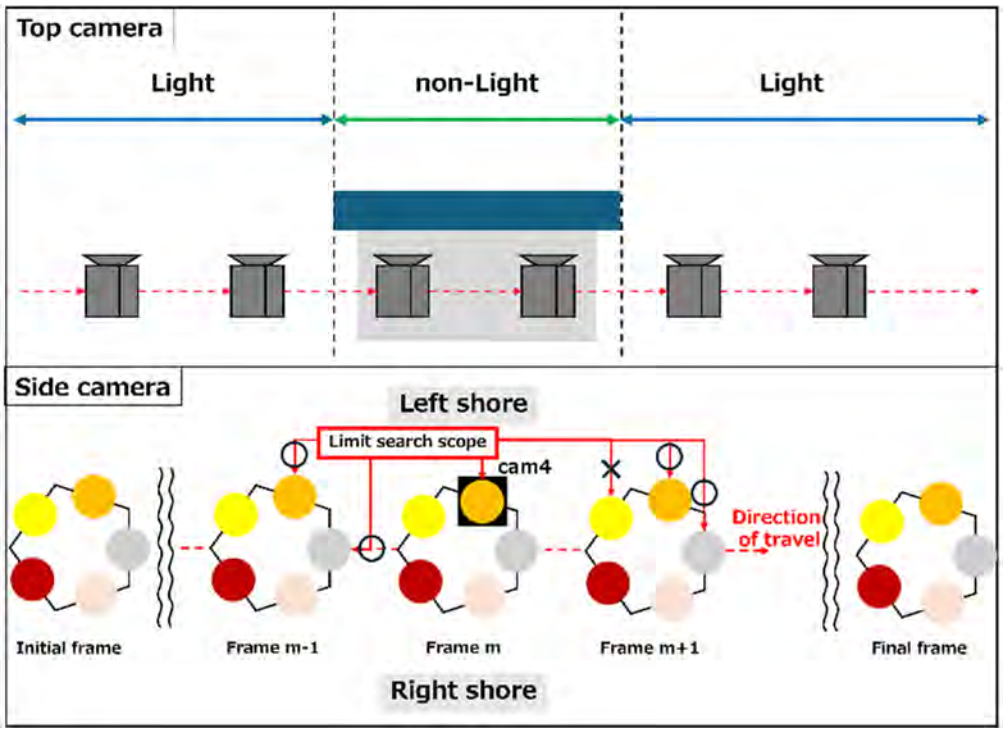


Figure 5: Proposed Methodology.

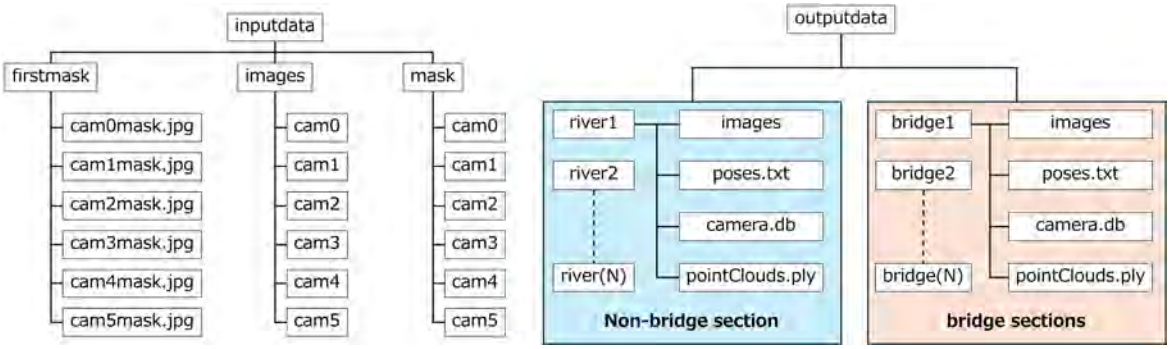


Figure 6: Folder Structures.

**Experiment**

On September 15, 2023, we acquired images and LiDAR data using a waterborne mobile mapping system (MMS) on the battery-powered boat Raicho I. The waterborne MMS was equipped with various sensors, including LiDARs (VLP-16 and VLP-32C, Velodyne), a CLAS GNSS receiver (AsteRx4, Septentrio), an inertial measurement unit (MTi-G-710, Xsens), and an omnidirectional camera (Ladybug, Teledyne FLIR) (Figure 7). Image processing of the omnidirectional camera data was performed on a desktop PC (CPU:

Intel(R) Core (TM) i7-11700, RAM: 32GB, GPU: NVIDIA GeForce GT 1030). We selected 4,020 omnidirectional images from the collected dataset for the SfM/MVS processing. These images, which include including bridges and open sky areas, were captured between Manseibashi Bridge and Shohei Bridge (near JR Akihabara Station) (Figure 8). We performed all SfM/MVS processing was performed using the open-source software COLMAP. Image masking was applied to focus the reconstruction on the relevant elements of the scene and reject the sky regions, the boat, and sensor equipment such as LiDAR. This process isolated the image regions that were converted into point clouds. For error evaluation, we used conventional SfM/MVS point clouds acquired with waterborne MMS. We also used LiDAR-SLAM data using two types of LiDAR (a horizontally scanning LiDAR and an oblique scanning LiDAR) mounted on the waterborne MMS as reference data to compare the point clouds generated by the proposed methodology.

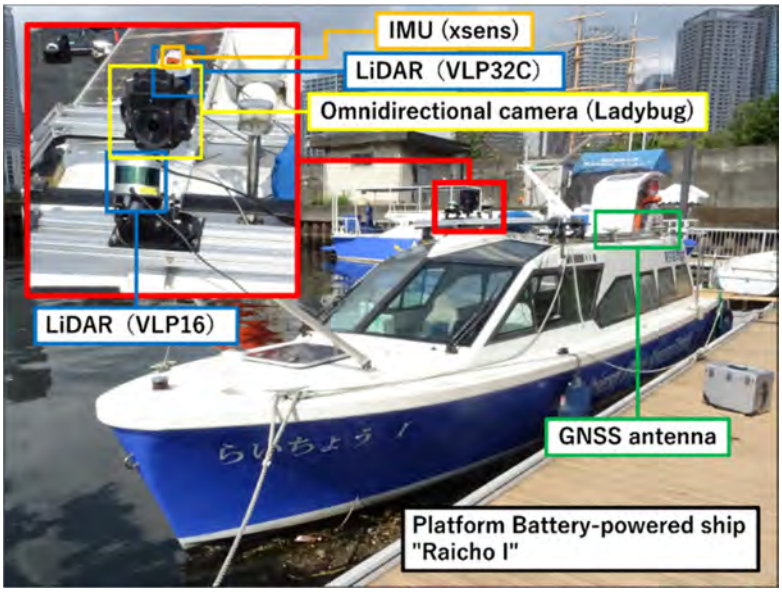


Figure 7: Waterborne MMS.

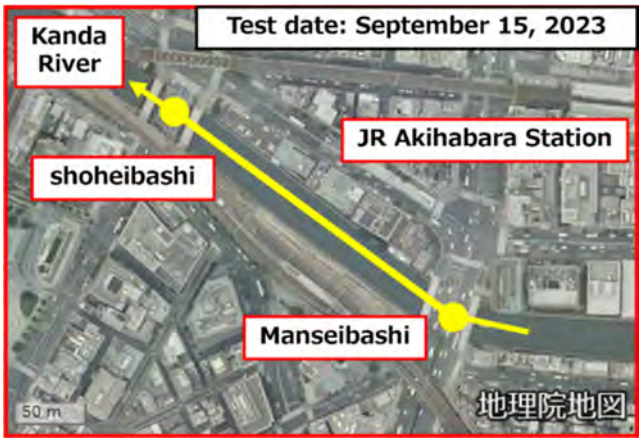


Figure 8: Experiment Section.

**Result**

The results of image masking, image-based point cloud generation, and LiDAR-SLAM based point cloud generation are presented below.

**a. Exhaustive Matching:**

Figure 9 shows the results of generating the point clouds using exhaustive matching-based MVS. Table 1 shows the point cloud generation time and the number of points. Camera positions estimated by SfM and GNSS position data were used to determine the scale, and the root mean square error (RMSE) was calculated using reprojection. The resulting point cloud contained 39,142,809 points, and the total processing time was approximately 214.1 hours.

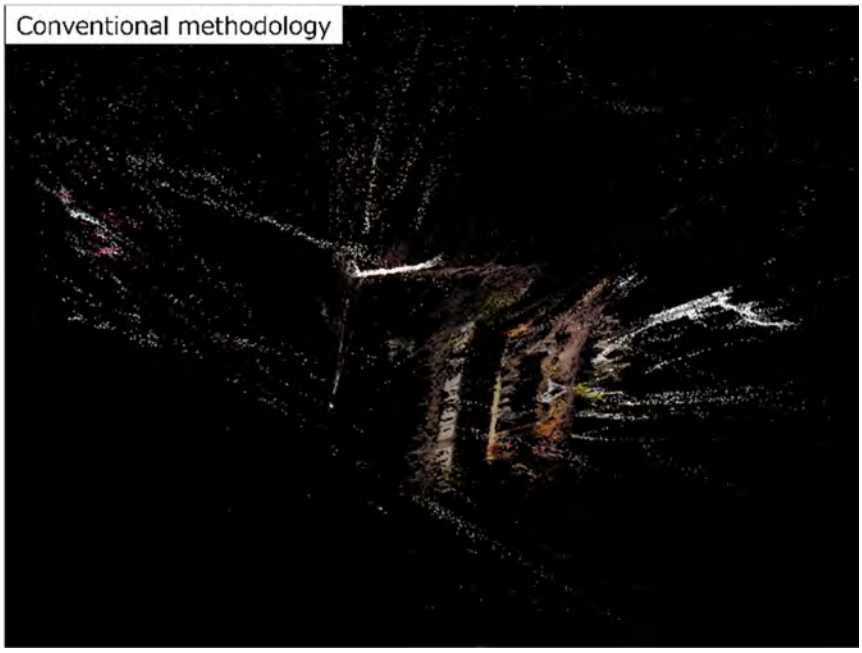


Figure 9: Generated SfM/MVS Point Clouds.

Table 1: Processing Time and Accuracy (Conventional Methodology).

| | Conventional methodology |
|---|---|
| Final mask processing [s] | 804.0 |
| SfM [s] | 400777.0 |
| MVS [s] | 369288.6 |
| Overall processing [s] | 770869.6 |
| The number of input images | 4020 |
| The number of used images | 3981 |
| The number of point clouds after MVS | 39142809 |
| RMSE [m] | 2.168 |

**b. Proposed Methodology:**

The results of point cloud generation using our proposed method are shown below.

**[1] Masking:**

Mask processing took approximately 2.6 seconds for the initial mask and 759.8 seconds for the final mask, for a total of 762.4 seconds, or approximately 0.18 seconds per image. Figure 10 shows the mask generation results for each camera for an arbitrary frame. Table 2 shows the accuracy of mask image generation for a total of 300 randomly selected images (50 from each camera). To create true images for comparison, unnecessary regions for the matching process were visually removed using image editing software, and the proportion of masked regions was compared with that of the images generated by the proposed methodology. Manual mask generation takes 1 to 2 minutes per image. It was confirmed that the generation time could be reduced to approximately 1/450. The mask image accuracy for all images was approximately 91%.


Figure 10: Masking Results.

Table 2: Accuracy of Mask Generation.

| | cam0 | cam1 | cam2 | cam3 | cam4 | cam5 | all |
|---|---|---|---|---|---|---|---|
| Accuracy [%] | 94 | 95 | 92 | 87 | 95 | 86 | 91 |

**[2] Bridge Sections:**

Two locations were identified in the area between Manseibashi Bridge and Shoheibashi Bridge. The MVS point cloud generation results for each location are shown in Figure 11. The processing time results are summarized in Table 3. Bridge 1 (Manseibashi Bridge) had a total of 329,578 points in the point cloud and took approximately 1.6 hours to process. Bridge 2 (Shoheibashi Bridge) had a total of 480,113 points in the point cloud and took approximately 2.8 hours to process.

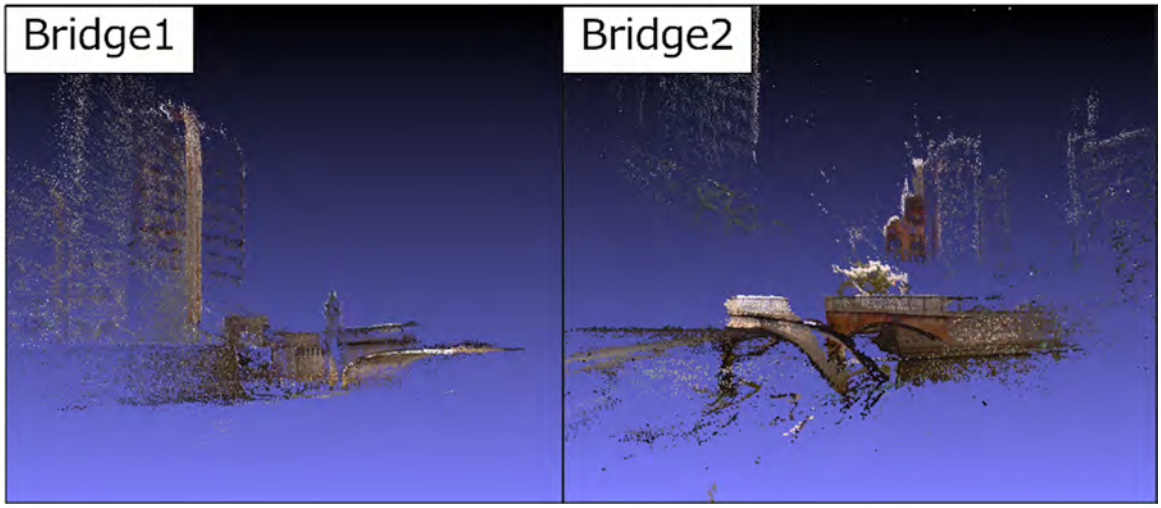


Figure 11: Generated SfM/MVS Point Clouds (Bridges).

Table 3: Processing Time and Accuracy.

| | Bridge1 | Bridge2 |
|---|---|---|
| Final mask processing [s] | 120.0 | 126.0 |
| Feature point extraction [s] | 308.8 | 324.3 |
| Matching search [s] | 1194.9 | 1724.4 |
| SfM [s] | 544.2 | 396.6 |
| MVS [s] | 3597.5 | 7349.2 |
| Overall processing [s] | 5765.5 | 9920.4 |
| The number of input images | 600 | 630 |
| The number of used images | 62 | 104 |
| The number of points in the MVS point clouds | 329578 | 480113 |
| RMSE [m] | 0.045 | 0.007 |

**[3] Other Sections:**

A total of three locations were detected, excluding the bridge section. The results of the MVS point cloud generation for each location are shown in Figure 12. The processing time and the number of points for each section are summarized in Table 4. For River 1, which had 210,732 points, the total processing time was approximately 1.3 hours. For River 2, which had 3,818,276 points, the total processing time was approximately 17.6 hours. For River 3, which had 1,372,927 points, the total processing time was approximately 3.8 hours.


Figure 12: Generated SfM/MVS Point Clouds (Other Sections).

Table 4: Processing Time and Accuracy.

| | River1 | River2 | River3 |
|---|---|---|---|
| Final mask processing [s] | 32.4 | 488.4 | 37.2 |
| Feature point extraction [s] | 83.4 | 1256.9 | 95.7 |
| Matching search [s] | 427.8 | 7829.2 | 950.2 |
| SfM [s] | 158.2 | 3388.3 | 494.8 |
| MVS [s] | 3812.9 | 50373.8 | 12174.8 |
| Overall processing [s] | 4514.7 | 63336.7 | 13752.7 |
| The number of input images | 162 | 2442 | 186 |
| The number of used images | 54 | 562 | 155 |
| The number of points in the MVS point clouds | 210732 | 3818276 | 1372927 |
| RMSE [m] | 1.319 | 0.023 | 0.316 |

**Discussion**

**a. Exhaustive Matching and Proposed Methodology:**

Figure 13 shows the results of point cloud generation and camera position estimation using conventional and proposed SfM processing. For this comparison, the point cloud sets for Bridge 1, Bridge 2, River 2, and River 3 were combined and the camera positions were

selected from the database based on accuracy. Table 5 compares the processing time and accuracy of the conventional and proposed methodologies. Compared to the proposed method, SfM processing reconstructed only part of the structure and inaccurately estimated the camera position. Although MVS processing generated many point clouds, their positions were not accurately reconstructed. This is likely due to inaccurate estimation of the camera position in SfM (see Figure 14), and insufficient matching in areas with large brightness differences, such as bridge sections. Furthermore, the matching process took approximately 8 times longer than the proposed methodology, because it matched all images. It was also likely that incorrect matching occurred between distant but similar features. This resulted in point clouds being reconstructed in locations where they should not exist.

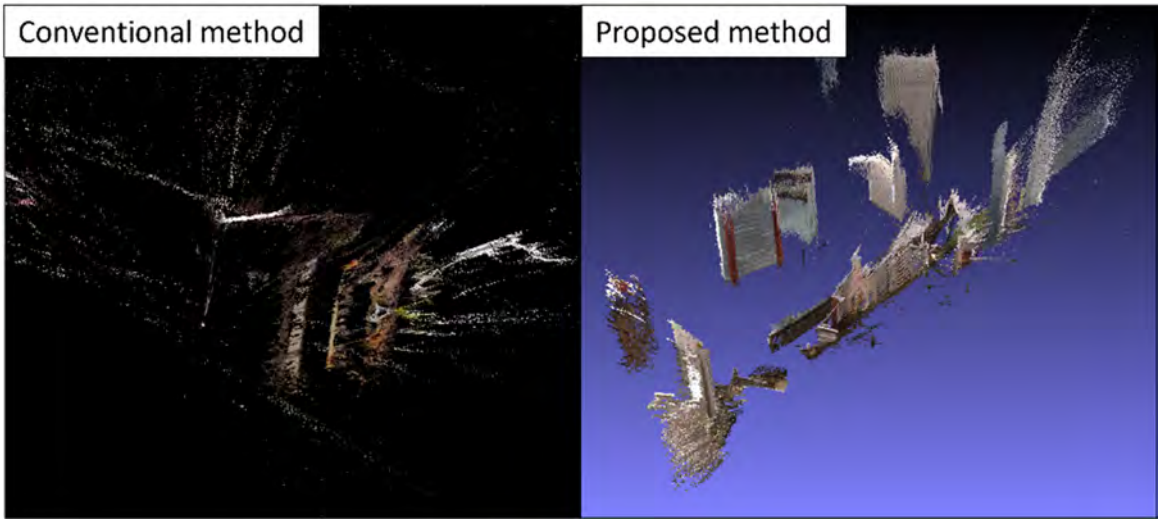


Figure 13: Comparison of SfM/MVS Point Clouds.

Table 5: Comparison of Processing Time and Accuracy.

| | Conventional method | Proposed method |
|---|---|---|
| Final mask processing [s] | 804.0 | 804.0 |
| SfM [s] | 400777.0 | 19177.7 |
| MVS [s] | 369288.6 | 77308.4 |
| Overall processing [s] | 770869.6 | 97290.1 |
| The number of input images | 4020 | 4020 |
| The number of used images | 3981 | 937 |
| The number of point clouds after MVS | 39142809 | 5936986 |
| RMSE [m] | 2.168 | 0.396 |

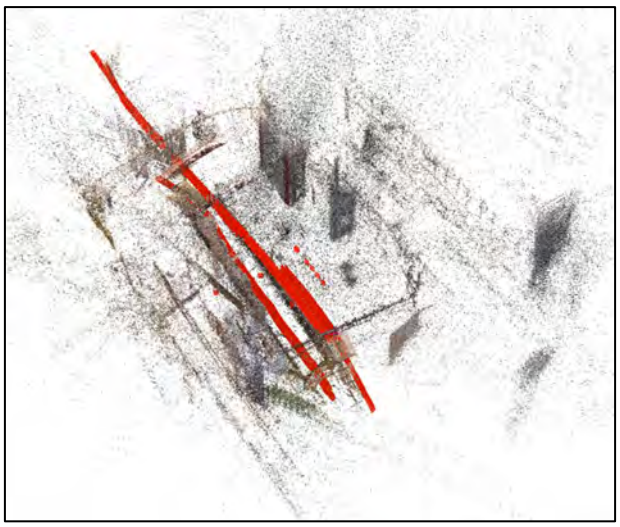

Figure 14: Estimated Camera Positions and Sparse Point Clouds after SfM Processing.

Figure 15 shows a comparison between the 3D point cloud generated by our proposed methodology and an aerial photograph. The methodology generally reproduced the scene for areas near the boat-mounted sensors accurately, such as revetments and bridges. However, some discrepancies were observed between the point clouds for distant buildings.

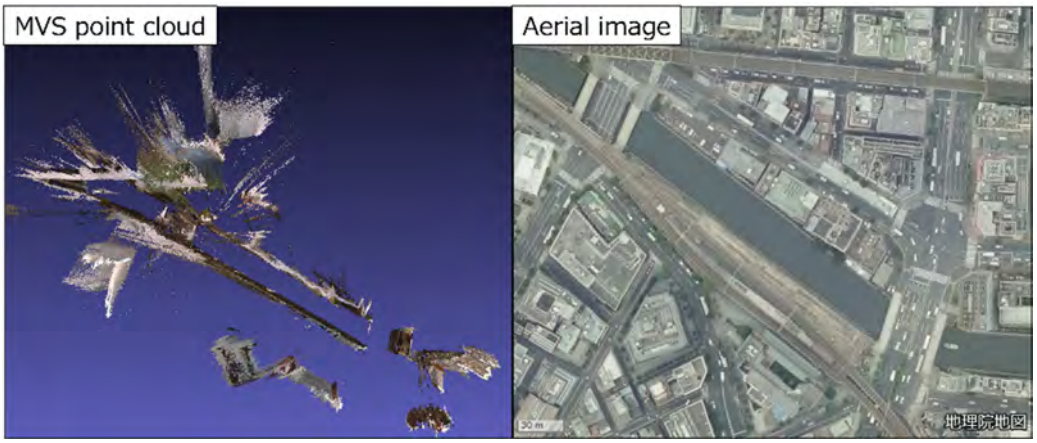


Figure 15: SfM/MVS Point Clouds and Aerial Image of the Same Area.

**b. Proposed Methodology:**

**[1] Masking:**

The mask images from cam1 and cam4 had an accuracy rate of approximately 95%, whereas those from cam3 and cam5 were in the 80% range. This is thought to be because some regions appearing unnatural after cropping and the inaccurate identification of light and dark regions. In particular, the pedestrian and bright cloud regions were remained in the images of cam3 and cam5, as shown in Figure 16.

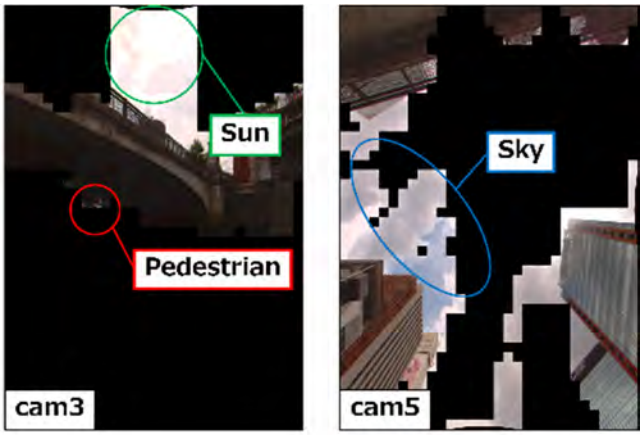


Figure 16 : Example of a Poorly Masked Image.

**[2] Bridge Section:**

Although the reconstruction of the area under part of Bridge 1 (Figure 17) was successful, it was unsuccessful at most other locations. This was primarily because images from the upper cameras were not selected during the SfM/MVS processing. This is likely because the images taken from close range were out of focus or had an inappropriate shutter speed, resulting in blurry images. However, point cloud generation was confirmed in relatively open areas.

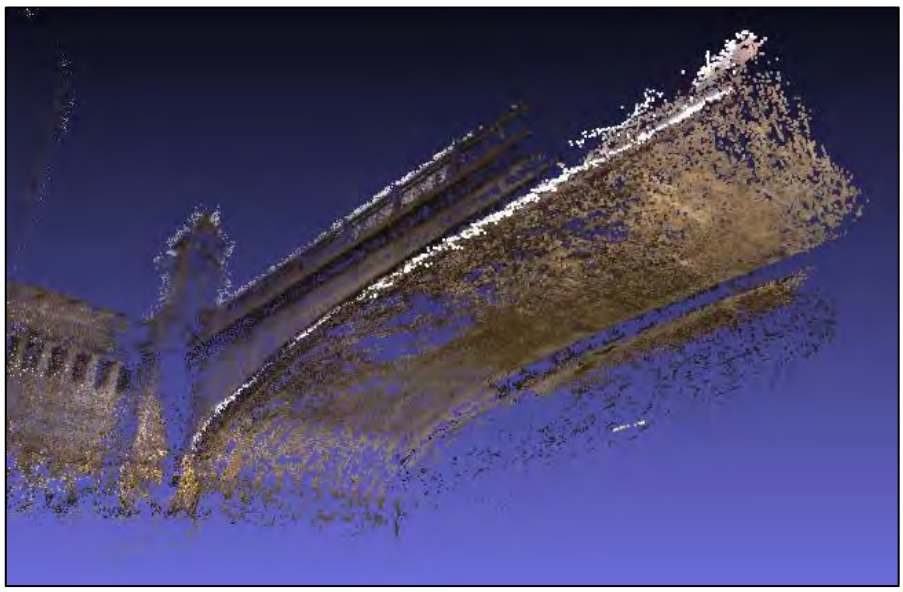

Figure 17: Example of Point Cloud Reconstruction under a Bridge.

**[3] Other Sections:**

At the revetment sites, the point cloud generation was successful because there were many image features, as shown in Figure 18 (river 2). However, although we used images from

the overhead camera as supplementary data, some of these images were not used in the SfM processing. Using these images may have introduced additional noise. In River 1, the point cloud generation was unsuccessful. Although some building structures could be identified as shown in Figure 19, this was likely due to the small number of images used in the SfM/MVS processing and the inaccurate estimation of camera position.

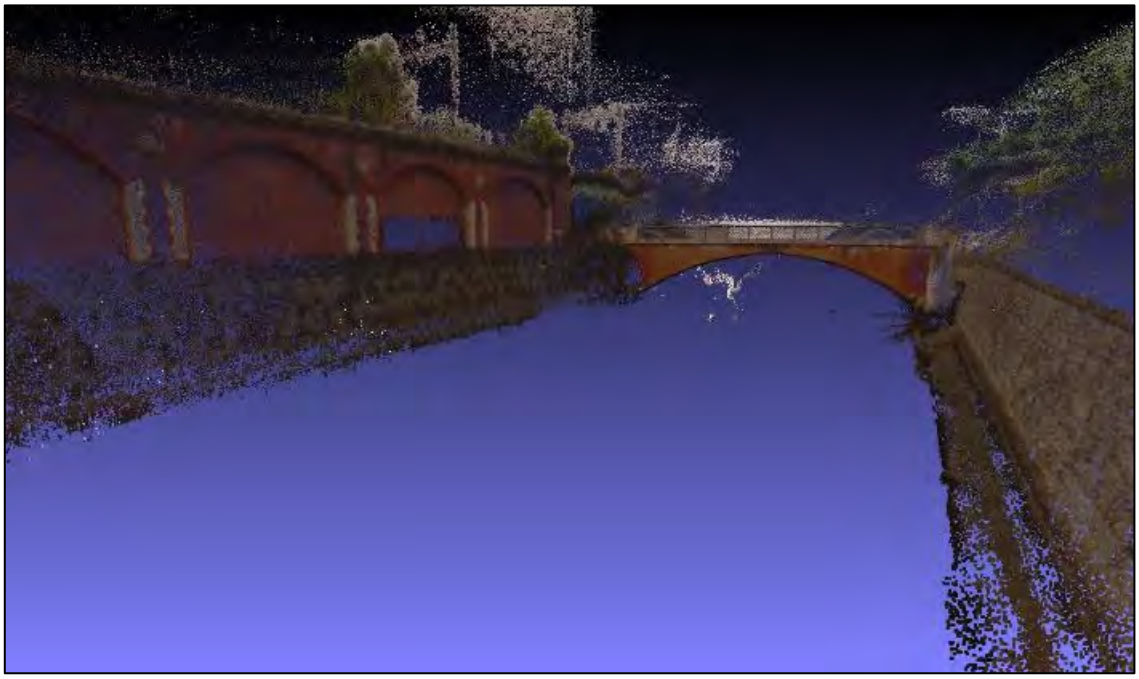

Figure 18: Examples of Point Cloud Generation in River Environments.

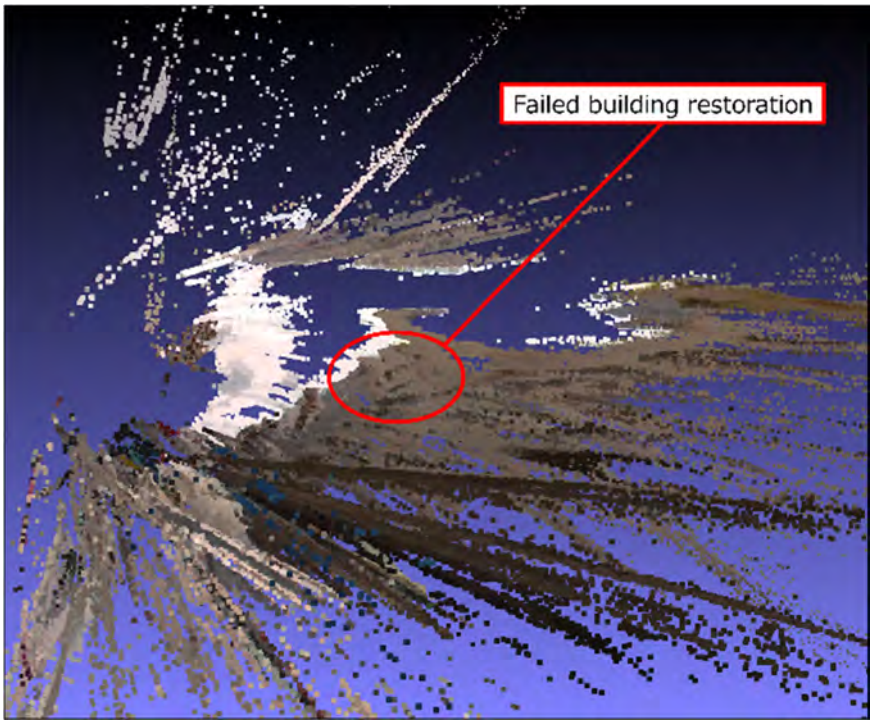


Figure 19: Examples of Failed Building Reconstruction.

## Conclusion and Recommendation

This study aimed to improve the accuracy and efficiency of SfM/MVS processing by implementing image masking for omnidirectional images, object classification using an overhead camera, and omnidirectional camera constraints. The results showed that this approach reduced the processing time required for point cloud generation and improved the quality of point cloud generation compared to conventional methodology. Future work will focus on the development of a more robust point cloud generation that integrates LiDAR point clouds and images.

## References

[1] Ministry of Land, Infrastructure, Transport and Tourism. (n.d.). Project PLATEAU. Retrieved August 22, 2025, from https://www.mlit.go.jp/plateau/

[2] Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (TOG)*, 25(3), 835–846.

[3] Schönberger, J. L., & Frahm, J.-M. (2016). Structure-from-Motion Revisited. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4104–4113.

[4] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. *IEEE Computer Vision and Pattern Recognition (CVPR)*, 519–528.

[5] Furukawa, Y., & Ponce, J. (2010). Accurate, Dense, and Robust Multi-View Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8), 1362–1376.

[6] Schönberger, J. L., & Frahm, J. M. (2016). Structure-from-Motion Revisited. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4104–4113.

[7] Hess, Wolfgang, Damon Kohler, Holger Rapp, and Daniel Andor. "Real-Time Loop Closure in 2D LIDAR SLAM." 2016 IEEE International Conference on Robotics and Automation (ICRA). 2016.

[8] Zhou, W., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity." IEEE Transactions on Image Processing. Vol. 13, Issue 4, April 2004, pp. 600–612.

[9] Parker, James R., Algorithms for Image Processing and Computer Vision, New York, John Wiley & Sons, Inc., 1997, 23-29.