Visualization of Multiple Spatial Data using Game Engine for Route Navigation Planning of Autonomous Boats in Urban Rivers

Makoto Ishiwata[1*], Nozomi Sadachika[1], Naoto Kimura[1], Nobuaki Kubo[2], Etsuro Shimizu[2] and Masafumi Nakagawa[1]

[1]Shibaura Institute of Technology, Japan

[2]Tokyo University of Marine Science and Technology, Japan

ah21040@shibaura-it.ac.jp

Abstract : *The objective of this study is to develop a 3D map that supports safer ship navigation to utilize urban river areas for disaster prevention, material transportation, and Mobility as a Service (MaaS). Although open 3D urban data have been developed under the Project PLATEAU of the Ministry of Land, Infrastructure, Transport and Tourism, data on river areas have not yet been sufficiently developed. Therefore, we investigated the 3D data acquisition of river areas using a boat. We also verified the feasibility of multiple spatial data visualization using a game engine for route navigation planning of autonomous boats in urban rivers. First, we acquired point clouds using a waterborne MMS consisting of RTK-GNSS, SLAM-LiDAR, and multi-beam sonar mounted on a battery-powered boat "Raicho I" on an urban river in Tokyo. Second, we processed open 3D city data (PLATEAU data). Third, we integrated all the acquired and processed data, such as land point clouds, underwater point clouds, and PLATEAU data using a game engine (Unreal Engine 5.2.1, Epic Games). As a result, we confirmed that our approach can reconstruct river spaces. However, parts of the underwater point clouds under bridges were distorted due to the non-GNSS positioning environment. We also confirmed that the Unreal Engine requires high storage-memory capacity and graphics performance. As applications of the research results, we proposed use cases using a game engine, such as avoidance due to tidal changes, collision avoidance with underwater obstacles, and collision avoidance with land structures. Our future work includes a boat behavior estimation using a physical simulator.*

Keywords: *3D Mapping, 3D Visualization, Game Engine, LiDAR, Mobile Mapping System*

Introduction

Many rivers in urban areas are navigable by boats, and they are expected to be used in a variety of applications in the future, such as disaster evacuation routes, transportation of goods, and the realization of Mobility as a Service (MaaS) that combines water and land transportation. However, these applications face challenges, such as restrictions on passable routes due to narrow river widths, height restrictions, and changes in water depth due to tidal levels, as well as invisible obstacles below the water surface that can hinder the safe navigation of boats. Therefore, the development of 3D maps for boats is necessary. In recent years, the Ministry of Land, Infrastructure, Transport and Tourism (MLIT) has been working on Project PLATEAU, a project to make 3D data for the entire city open data. It is expected that this will enable the city to be reproduced in digital space, and various simulations will be carried out in digital space. However, for urban rivers, only models that take into account high tides and the like have been developed, and 3D data for urban river spaces have not yet been developed. Therefore, to create 3D maps for boats, a ship MMS was developed in previous research (Nakagawa et al., 2023) to obtain point clouds of river spaces. In this study, we verified the use of a game engine in the integrated visualization of multiple types of 3D data for boats, to utilize point clouds acquired in river spaces for the safety of navigation in urban rivers.

Methodology

The proposed method is shown in Figure 1. In this study, land point clouds and underwater point clouds acquired by a waterborne MMS are integrated with PLATEAU data and visualized on a game engine. Unnecessary points are removed from the integrated point clouds using segmentation. Several plug-ins are used to import the data into the game engine.

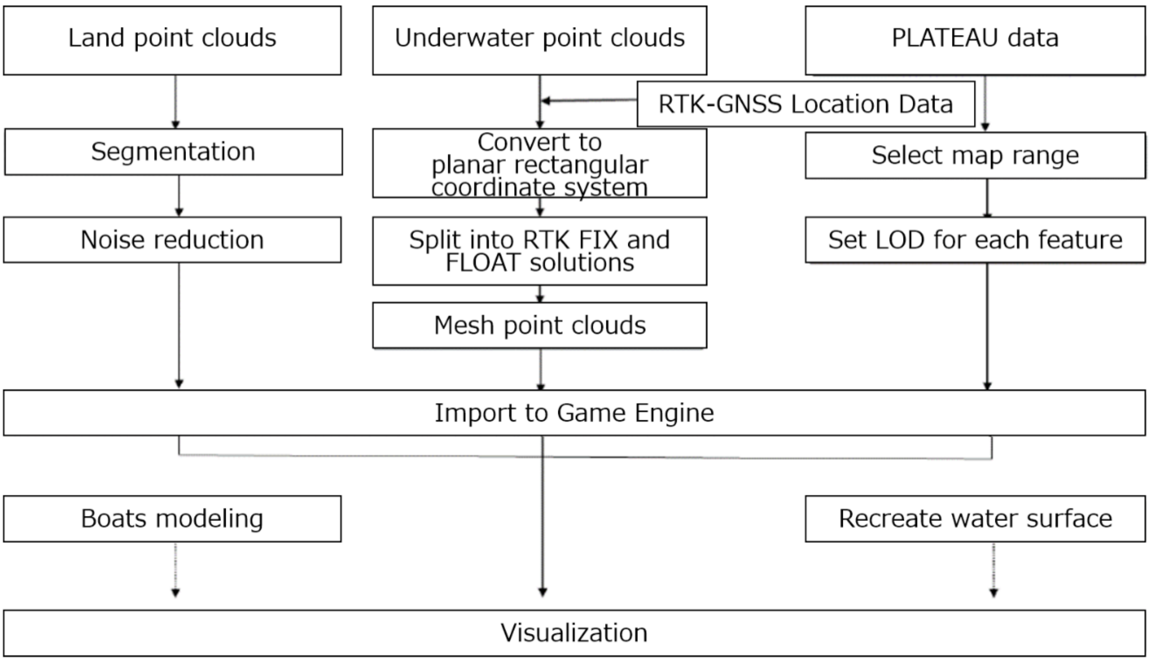

Figure 1: Proposed processing flow.

Meshing a point cloud is the process of converting point cloud data into a 3D mesh that represents smooth and complex curved shapes. A 3D mesh is easier to visually understand a 3D scene than point cloud data, thus, it is commonly used in 3D modeling and simulation (Kanai, 2018). The algorithm includes calculating the normal vectors of each point, generating a mesh taking into account the neighboring points and normal vectors, smoothing the generated mesh, and outputting it. In this study, the normal vectors of the point clouds are adjusted before the meshing process.

We selected Unreal Engine (Epic Games) as the game engine provided. We also selected the latest Unreal Engine 5.3. Unreal Engine, which has detailed graphics, The Unreal Engine is widely used for the development of PC games and smartphone games. We also used an open-source toolkit (PLATEAU SDK for Unreal) to import PLATEAU data into the Unreal Engine. The PLATEAU SDK for Unreal provides the following features.

a) City model import function
   3D city model data conforming to PLATEAU's standard product specifications can be imported into the Unreal Engine.
b) City model customization function
   The appearance of 3D city models can be adjusted with LOD and features.
c) Attribute information capture function
   All attribute information in CityGML can be retrieved using an API for Blueprint and C++ users.
d) City model export function
   The imported 3D city model can be exported in 3D file format.

The river in urban areas is generally shallow, thus, point clouds of the riverbed cannot be acquired by echo sounding. Therefore, riverbed point clouds are interpolated using the boundaries of revetments extracted from underwater point clouds. Next, the normal vectors of the point are adjusted, and mesh processing is applied. Finally, the processing result is saved in a file format compatible with the game engine (Figure 2).

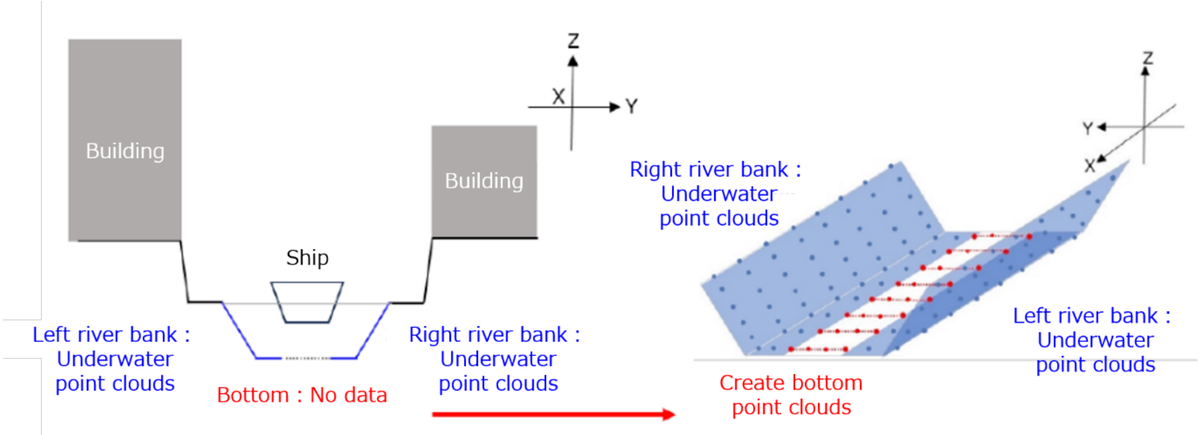

Figure 2: Image of water bottom point cloud creation.

Experiment

The Kanda River was selected as the experimental section, as shown in Figure 3. The measurements started at Asakusabashi and turned back at Suidobashi. The experiments were conducted in the morning at low tide on September 15 and September 29, as shown in Table 4.

The Kanda River is a first-class river that flows through the center of Tokyo. Although navigation is generally possible, aerial views are often obstructed by the construction of expressways, railways, and bridges across the river. High-rise office buildings and commercial buildings line the banks, and construction of coastal structures can narrow the river’s width by placing scaffolding in the river. This environment is believed to cause some degradation of the satellite signal reception environment. In addition, the water of the Kanda River is turbid, making it impossible to confirm the situation underwater. Because the water is shallow and the river is narrow, there is a risk of accidents such as running aground. In addition, since the bridge is not high enough, there is a risk of collision when large boats pass or during high tide.

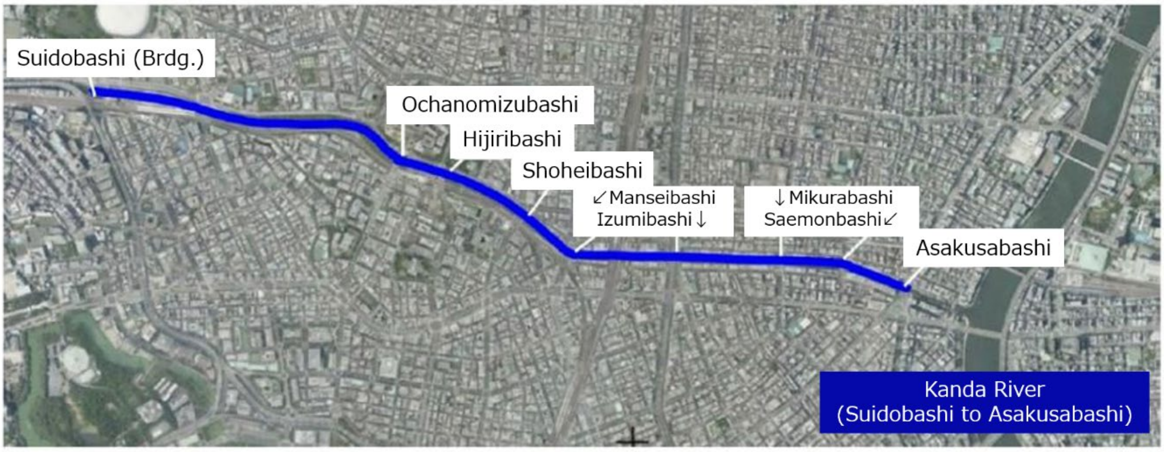

Figure 3: Target section of the experiment.

Table 4: Experiment date.

Date	Departure Time	Arrival Time	Measurement Time
2023/9/15	9:50 AM	13:00 PM	3h10m
2023/9/29	9:30 AM	13:30 PM	4h00m

Horizontal and diagonal scanning LiDARs (VLP-16 and VLP-32, Velodyne), RTK-GNSS antenna (Septentrio) and receiver (AsteRx4, Core), and multi-beam sonar (BV5000, Teledyne BlueView) were mounted on the battery-powered boat "Raicho I" (Shomasa, 2012), as shown in Figure 5. Moreover, a SLAM-LiDAR (Hovermap ST-X, Emesent) was used for additional point cloud acquisition. The measurements were performed with two round trips at a speed of three knots.

Figure 5: Experimental equipment for point cloud acquisition.

The processing flow with the Unreal Engine is described as follows.

a) PLATEAU data input

A 3D city model is created using the PLATEAU SDK for Unreal plugin to retrieve PLATEAU data from the server, set the offset value, and select the reference coordinate system, map area, and LOD.

b) Point cloud input

Point cloud files are imported into the Content Browser and placed in the viewport by dragging and dropping. The placement and size of the actors can be changed manually. A material is created in the Content Browser and the color is set. Next, the color is selected using the SpecularColor3 function in the blueprint. The material is then dragged and dropped from the Content Browser onto the actor in the viewport.

c) Water surface reconstruction

The water surface is reconstructed using the Water plugin. When reconstructing a river, the River Water Body is selected. When creating a long-distance river, the river width, tide level, and angle are adjusted using the created points.

d) Boat modeling

A 3D boat model is copied from a 3D Web library site, or a 3D boat model is created using a 3D modeling software product. The 3D model is imported into the Content Browser to create a blueprint class. Buoyancy and arrows are added to the boat model. Buoyancy is a feature used to simulate the behavior of underwater objects. The arrows are visual elements used to indicate direction. The added components are used to reconstruct the boat with the weight of the boat and the position where the buoyancy acts (waterline).

Results

A) GNSS positioning

The RTK-GNSS positioning data were visualized on a map, as shown in Figure 6. The red lines show RTK-FIX solutions, and the blue lines show other positioning statuses. We confirmed that RTK-FIX solutions were obtained at Kanda River. However, we also confirmed that measurements were intermittently unstable due to the FLOAT solution and single positioning under bridges. In the section between the Hijiri-bashi bridge and the Ochanomizu-bashi bridge, the GNSS positioning environment was poor due to construction work on the riverbank. Similarly, in the section between the Ochanomizu-bashi bridge and

Suido-bashi bridge, the GNSS environment was also poor due to the influence of high-rise buildings.

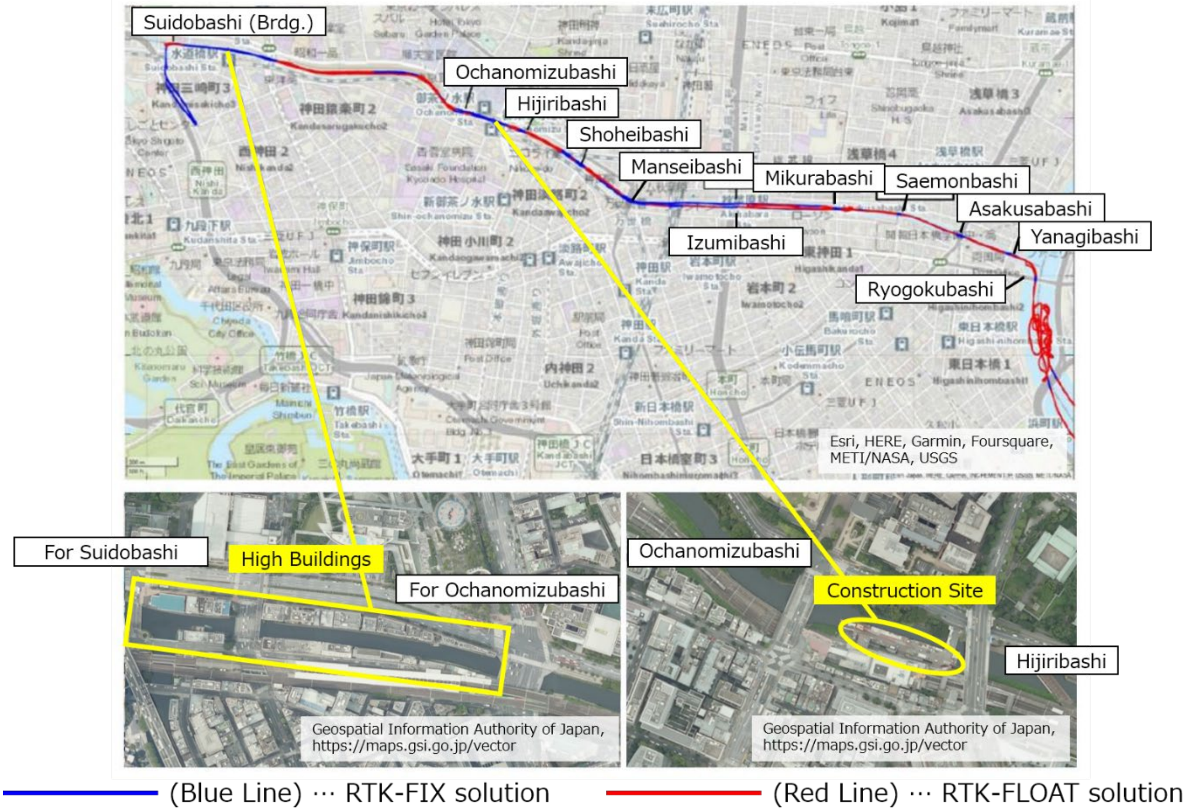

Figure 6: RTK-GNSS positioning result.

The mesh processing results of the underwater point clouds after the interpolation are shown in Figure 7.

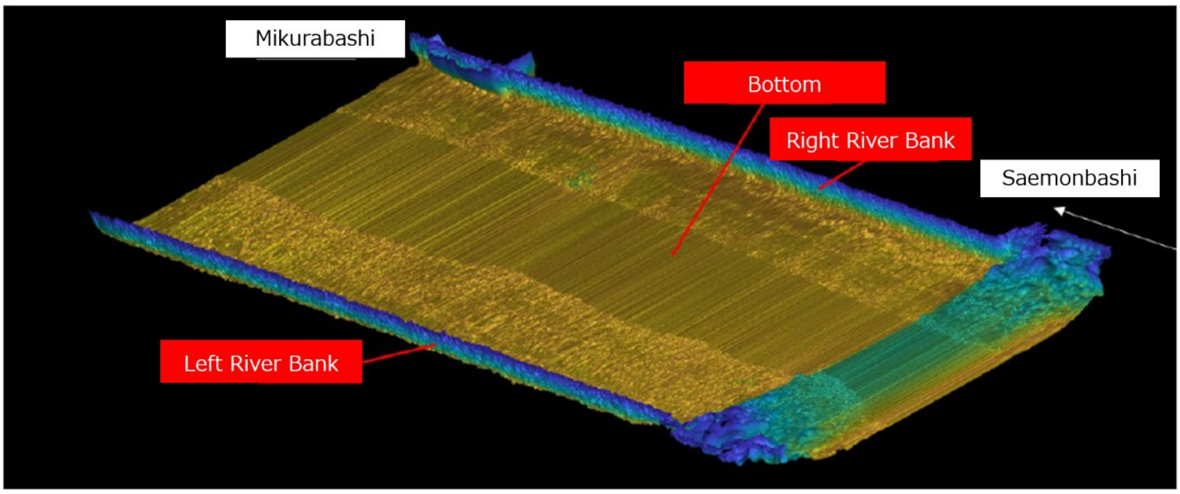

Figure 7: Meshing process.

Figure 8 shows the segmentation result applied to the land point clouds after unnecessary point rejection.

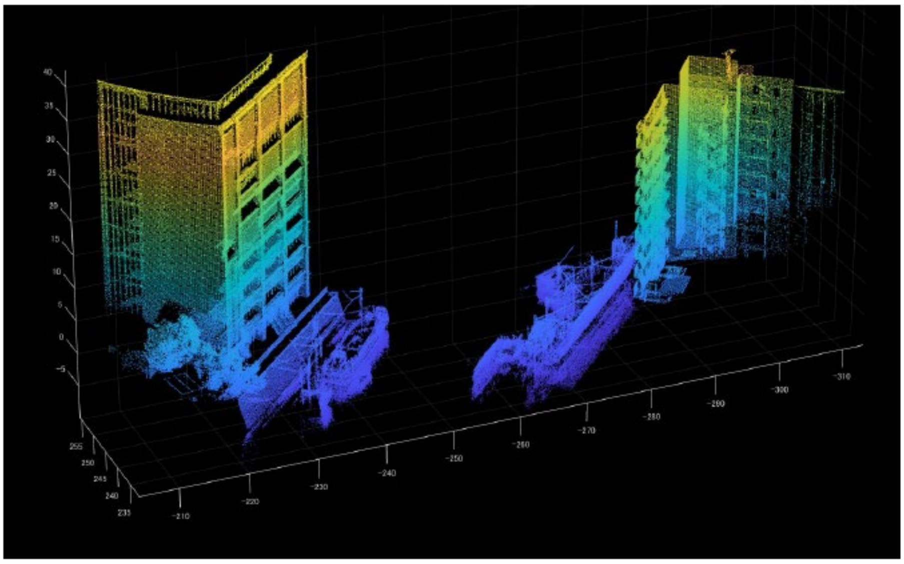

Figure 8: Land point clouds.

In the proposed method, the noise-removed land point cloud data, the meshed underwater point cloud data, and the PLATEAU data (CityGML/LOD1) were integrated into the Unreal Engine to reconstruct a moving boat, buildings, revetments, and the water surfaces along the Kanda-gawa River, as shown in Figure 9.

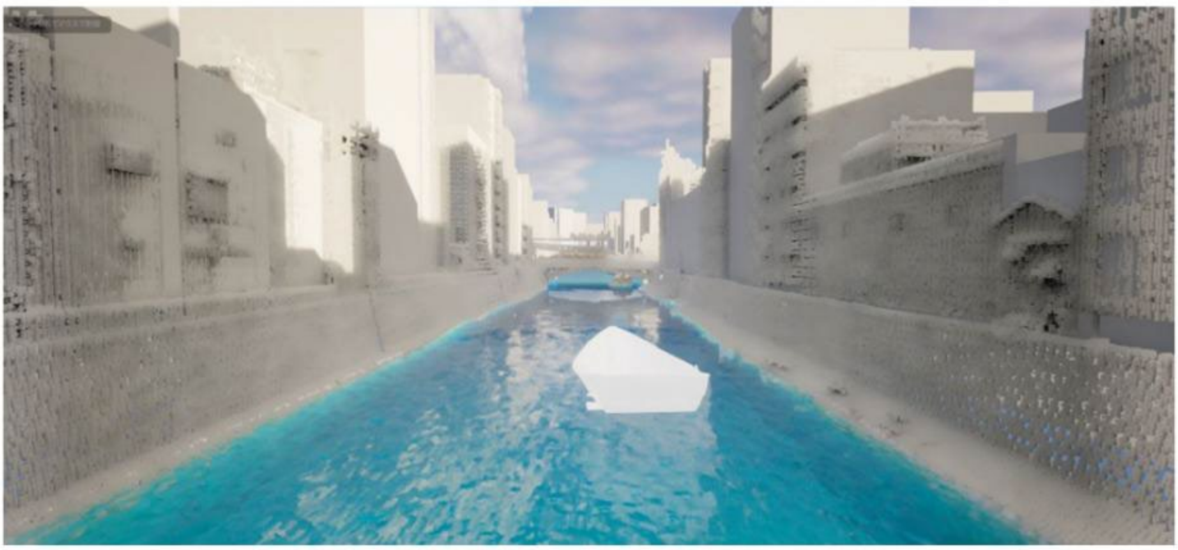

Figure 9: Processing results (between Saemonbashi and Mikurabashi).

Details of revetments and riverbeds were visualized using the multi-beam sonar data acquired with the RTK-FIX solution, as shown in Figure 10. On the other hand, fine mesh models around bridges and high-rise buildings could not be reconstructed due to the poor GNSS environment. Based on these technical issues, data acquisition methods that consider water depth for data acquisition and mesh modeling of areas in poor GNSS environments are our future work.

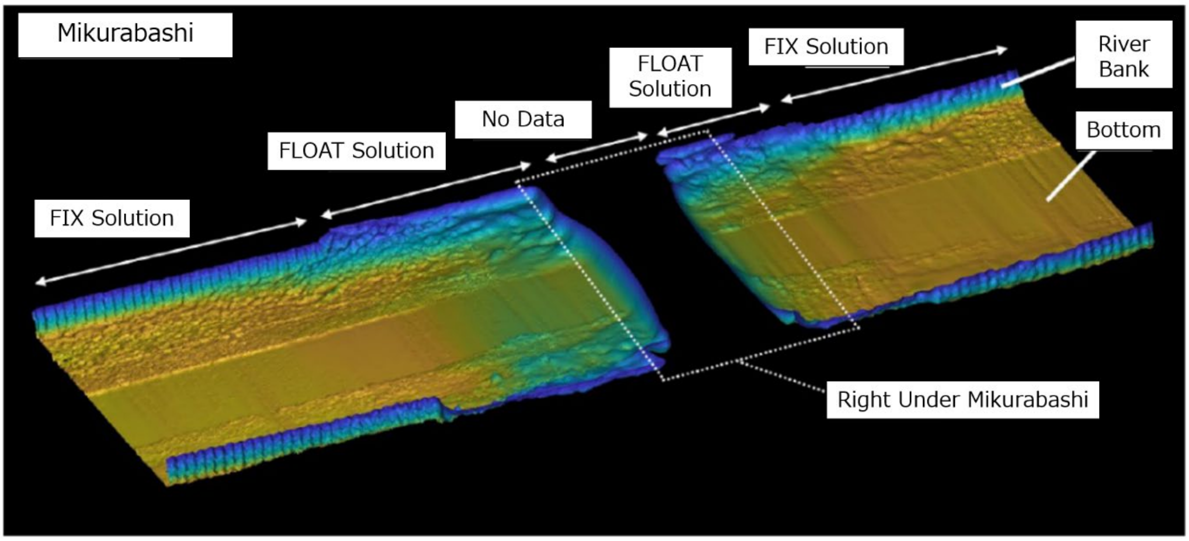

Figure 10: Scan matching output results for each measurement area.

B) Game engine

In this study, we used the PLATEAU SDK for Unreal, LiDAR Point Cloud Support, and the Water plugin. However, while only Unreal Engine 5.2.1 supports PLATEAU SDK for Unreal, Lidar Point Cloud Support was Unreal Engine 4. Since we were using Unreal Engine 5.2.1, we imported the land point cloud data every time we started the application. We confirmed that it is not easy to combine different plugins on one game engine. We also confirmed that the reproducibility of the visualization in Unreal Engine is low due to the limitations of using datasets with different coordinate systems. In addition, Unreal Engine requires a large amount of RAM, more than 20GB, and 512 GB of storage to achieve smooth manual work. Unreal Engine supports many file formats, and the 3D data collected in this study was easily imported without any problems. We confirmed that the point size and color of the land point cloud data can be changed in Unreal Engine. In contrast, we also confirmed that multiple colors and point cloud editing were not covered, thus, point cloud editing is required before the point cloud input to the Unreal Engine. Moreover, since the coordinate system was not reflected in the acoustic bathymetry data, it was necessary to manually position the objects to match different coordinate systems of 3D data, such as land point clouds, underwater point clouds, and PLATEAU data, which should be manually registered in the Unreal Engine. The background of this research was the question of how to deal with water depth changes due to tidal changes. Using the Water plug-in of UE5.2.1, a realistic water surface could be reconstructed with the tide level, current speed, and river width. In this study, although the tide level was manually adjusted, it is expected that the reconstruction of the tide level will be automatically estimated using the tidal level fluctuation data.

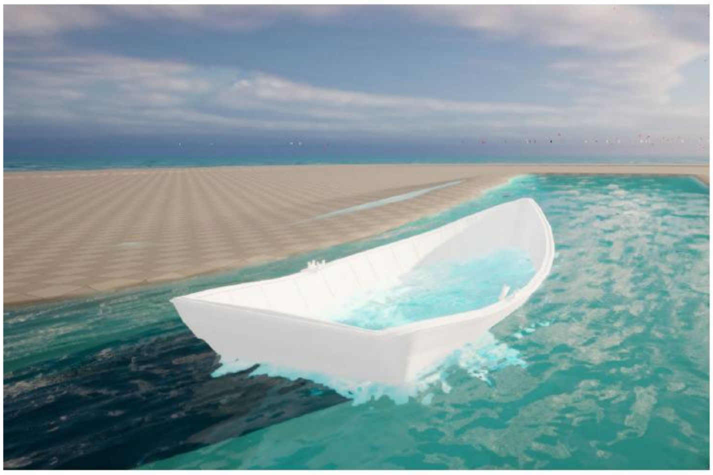

Figure 11: Water entering the boat.

C) Autonomous boat applications

The game engine can reconstruct the navigation of a boat, and the physical simulation can be processed. Moreover, the water surface can have a function to simulate the active tidal level and flow speed. Therefore, we propose three two specific ways to use the research results to improve the safety of navigation in urban rivers, which is the purpose of this research.

a) Avoidance of grounding of boats when passing each other (Figure 12):
   Prediction and visualization of grounding and collisions can suggest better routes and steering methods in advance when passing each other in narrow urban rivers. Navigable areas can be detected using a game engine, even if the underwater topography is invisible from a boat.

Figure 12: Avoiding grounding when passing each other.

b) Avoiding collisions with underwater obstacles due to tidal changes (Figure 13):
The game engine can predict collisions with underwater obstacles using information such as changing tidal level and water depth, which are necessary for safe navigation. The results of our experiments showed many obstacles and uneven topography in the underwater. It is expected that a game engine can be used to avoid damage to boats when the tidal level is low.

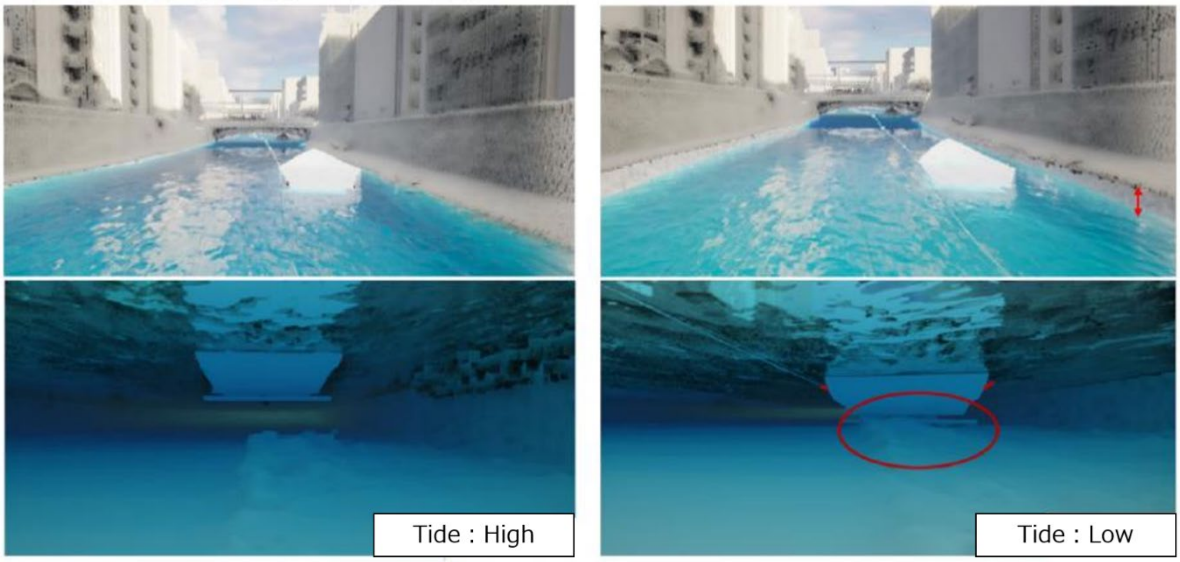

Figure 13: Collision with underwater obstacles due to tidal changes.

c) Avoiding collisions with overhead structures due to tidal changes (Figure 14):
The game engine can suggest better navigation to avoid collision with overhead structures during tidal changes in advance. The tidal level and boat model can be freely changed, thus, the structure gauge can be calculated in advance or in real-time. The structure gauge estimation can be also used for boats not only in peacetime during high tidal level, but also during disasters to improve the safety of navigation.

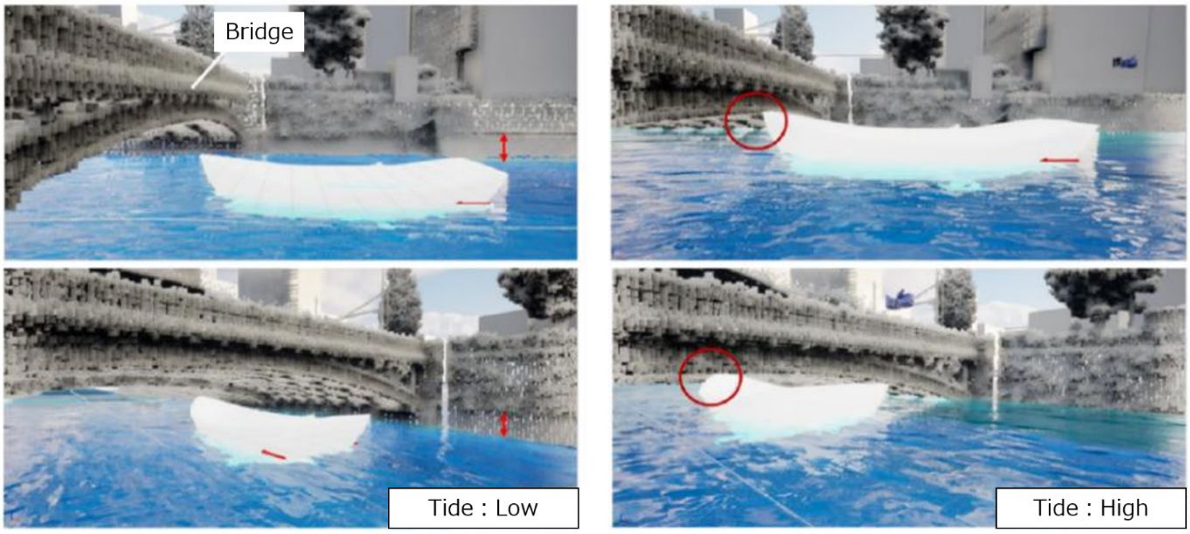

Figure 14: Avoiding collisions with overhead structures due to tidal changes.

Conclusion and Recommendation

In this study, we investigated the use of a game engine in the integrated visualization of multiple types of 3D data for safer boat navigation in urban rivers. We confirmed that the acquisition of underwater point clouds using a multi-beam sonar with mesh modeling can visualize urban rivers with low transparency and unknown riverbeds. We also confirmed that land point clouds using LiDAR can visualize the details of the urban river space, including bridges and revetments, for which the data organization is insufficient in existing open 3D urban models such as PLATEAU data is inadequate. We also confirmed that the game engine supports many file formats to import different 3D data. However, we found technical issues with visualization using the game engine, such as the overlay reproducibility of multiple 3D data with different coordinate systems, compatibility between engine versions, and plug-in integration. Although many challenges remain for practical use, we have confirmed that game engines are useful for civil engineering because of the ease of data integration, ease of use, and the free content provision.

References

Hiroyuki Shomasa, (2012). Development of rapid charging battery-propelled ship “Raicho I”. *Journal of the JIME Vol.47, No.1, pp9-16, 2012.*

Masafumi Nakagawa, Naoto Kimura, Takeshi Komori, Nobuaki Kubo, Etsuro Shimizu, (2023). Point Cloud Acquisition based on GNSS/SLAM with an Autonomous Boat in Urban River Environments. *44th Asian Conference on Remote Sensing, 30th October-4th November 2023, Taipei, Taiwan.*

Ministry of Land, Infrastructure, Transport and Tourism: Using PLATEAU SDK [2/2] | Using PLATEAU SDK for Unreal, from https://www.mlit.go.jp/plateau/learning/tpc17-2/

Takashi Kanai, (2018). Mesh Processing, *Journal of the Japan Society for Precision Engineering Vol.84, No.4, pp323-327.*