Real-time 3D Mapping of Construction Sites Using ORB SLAM and Stereo Cameras

R. Ishiguro*[1], J. Susaki[2] and Y. Ishii[3]

[1]Graduate School of Engineering, Kyoto University, Kyotodaigaku Katsura, Nishikyoku, Kyoto 615-8540, Japan. Email:<ishiguro.ryunosuke.62w@st.kyoto-u.ac.jp>

[2]Professor, Graduate School of Engineering, Kyoto University, Kyotodaigaku Katsura, Nishikyoku, Kyoto 615-8540, Japan. Email:<susaki.junichi.3r@kyoto-u.ac.jp>

[3]Assistant Professor, Graduate School of Engineering, Kyoto University, Kyotodaigaku Katsura, Nishikyoku, Kyoto 615-8540, Japan. Email:<ishii.yoshie.4k@kyoto-u.ac.jp>

*Corresponding author: R. Ishiguro, Email: <ishiguro.ryunosuke.62w@st.kyoto-u.ac.jp>

Received: November 23, 2024; Accepted: January 14, 2025; Published: January 27, 2025

ABSTRACT

In this study, we developed a method to create a dense three-dimensional (3D) map in real time using a stereo camera mounted on a drone and oriented features from accelerated segment test (FAST) and rotated binary robust independent elementary features (BRIEF) simultaneous localization and mapping (ORB SLAM), which simultaneously estimates self-location and generates sparse 3D point clouds. Sparse point clouds from ORB SLAM are insufficient for automating crane operations, necessitating conversion to dense point clouds. Traditional multi-view stereo (MVS) methods are unsuitable for real-time processing due to their computational demands. Our method addresses this by generating dense point clouds from stereo cameras, integrating them using self-estimation data, and filtering out outliers. Using simulation data representing construction sites, including buildings and cranes, we evaluated approximately 4,500 video frames. The process took 545.1 seconds and accurately captured site details such as building textures and object shapes. Future work will focus on developing algorithms to update only changed objects in the map, enabling dynamic representation of construction sites.

Keywords: Photogrammetry, ORB SLAM, Computer vision, 3D mapping, ROS, Three-dimensional map

1. INTRODUCTION

Recently, the number of crane operators at construction sites has been decreasing owing to the aging of workers, declining rate of young people entering the workforce, and reduction in working hours because of work-style reforms. According to the report "Current Status and Issues Surrounding the Construction Industry," published by the Ministry of Land, Infrastructure, Transport, and Tourism, the number of construction companies at the end of fiscal year 2021 was approximately 480,000, a decrease of approximately 21% from the peak at the end of fiscal year 1999 (Ministry of Land, Infrastructure,

https://doi.org/10.30217/AJG.202501_25(1).0003

Transport and Tourism, 2023). Additionally, the average number of construction workers in 2022 was 4.79 million, a decrease of approximately 30% from the average in 1997.

As highlighted above, labor shortages in the construction industry have become a significant societal issue. One potential solution to this problem is the automation of crane operations, which could help alleviate the strain on the workforce and maintain productivity. For instance, automating the transport of suspended loads can improve efficiency and reduce reliance on experienced crane operators. However, to enable automated crane operations, obtaining an accurate three-dimensional (3D) representation of the construction site is essential. In our crane automation concept, we aim to create a fresh 3D map of the construction site in approximately 5–10 min each morning before work begins. By generating a map at the start of each day, we can capture the latest state of the site environment, thereby providing a reliable basis for automation. This initial map then serves as a foundation, enabling us to update it in real time as construction progresses and conditions change throughout the day. Our study focuses specifically on the process of daily initial map generation and targets cost-effectiveness, speed, and accuracy, using only image data.

To develop the initial map, we employed a 3D map derived from a monocular camera affixed to the end of the crane boom in accordance with methodologies established in prior research (Kobayashi et al., 2023). This study involved capturing the surrounding environment within approximately 10 min through the rotation of the boom before the initiation of operational activities. However, several challenges hinder its practical application. The foremost issue pertains to the physical limitations of the camera mounted at the end of the crane hook, which restrict the range of the generated 3D map. In addition, the scale remains indeterminate when using a monocular camera. These challenges are unacceptable, given the objective of automating crane operations. Therefore, we propose a method to address these issues. Next, we provide an overview of the proposed approach.

To address these challenges, a stereo camera was attached to the drone. This mounting strategy enabled the mitigation of the physical limitations imposed by attaching a camera to the crane hook. Furthermore, the implementation of a stereo camera effectively resolves the scale ambiguity characteristics of monocular cameras. Reconstruction in 3D using drones and image data has been widely studied and offers solutions across various fields such as construction, archaeology, agriculture, and environmental monitoring. For instance, archaeological site mapping uses structure from motion (SfM) (Tomashi et al., 1992; Snavely et al., 2006), in which drone images are processed to reconstruct detailed 3D models (Barratt, 2021). Additionally, the use of drones for photogrammetry has proven to be effective in monitoring construction progress (Loyola et al., 2016) and forestry management for 3D modeling (Honkavaara et al., 2012). However, all these studies reported considerable computational times for processing, which can delay immediate application in dynamic environments.

To achieve a more efficient real-time mapping process, oriented features from accelerated segment test (FAST) and rotated binary robust independent elementary features (BRIEF) simultaneous localization and mapping (ORB SLAM) (Campos et al., 2021) was adopted as the core component of our methodology. ORB SLAM provides a powerful framework for simultaneous localization and mapping, enabling real-time estimation of the position of the drone while generating a sparse 3D point cloud of the surroundings. This approach allows construction of an initial 3D map with minimal delay, which is crucial for the dynamic conditions of construction sites. However, because the point cloud generated by ORB SLAM is sparse and insufficient for crane automation, we developed an additional process to convert the sparse point cloud into a denser representation. Our proposed method leverages the position and orientation data provided by ORB SLAM to integrate 3D point clouds generated by a stereo camera, thereby achieving a denser and more accurate representation in real time. This approach addresses the initial challenge related to physical constraints and overcomes the limitations of time-intensive methods, such as multi-view stereo (MVS), ensuring that real-time performance is maintained.

2. METHODOLOGY

2.1 Overview

This study presents a comprehensive methodology for generating a 3D map of a construction site by integrating a unity-based simulation environment with a Robot Operating System (ROS) framework designed specifically for image processing. Figure 1 illustrates the overall process flow. The simulated environment, meticulously constructed using Unity (Unity Technologies, 2024), accurately replicates a realistic construction site complete with buildings and vehicles, as shown in Figure 2. Within this virtual setting, a drone equipped with a stereo camera captures images from the left and right sides of the site, which were subsequently used to reconstruct a 3D map. Image processing occurs within the ROS framework, which initiates a mapping workflow by estimating the position and orientation of the stereo camera using the ORB-SLAM algorithm. This algorithm processes the stereo images obtained from a drone to provide precise self-localization data. Once the position of the camera is established, a disparity image is generated by analyzing the left and right image pairs captured by the stereo camera. This disparity image is then converted into a 3D point cloud for each frame. The 3D point clouds generated at various time intervals are integrated using the camera position and orientation data to ensure spatial consistency throughout the sequence. However, a noise reduction process is implemented because of the inherent noise present in point clouds arising from factors such as unintended distortions, pixelation, blurring, or color shifts caused by sensor limitations or errors in image processing. Specifically, the k-nearest neighbor (k-NN) method filters out erroneous points and enhances the overall quality of the 3D map. The proposed workflow demonstrates seamless integration of simulation and real-time image processing, facilitating the development and evaluation of 3D mapping techniques within a controlled environment.

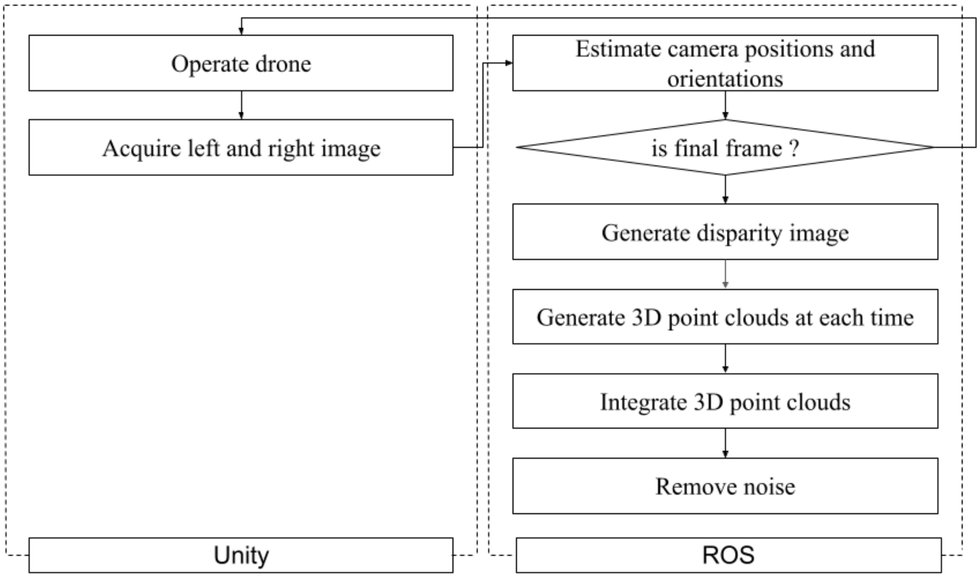
Operate drone
Acquire left and right image
Estimate camera positions and orientations
is final frame ?
Generate disparity image
Generate 3D point clouds at each time
Integrate 3D point clouds
Remove noise
Unity
ROS

Figure 1: Overview of the study method.

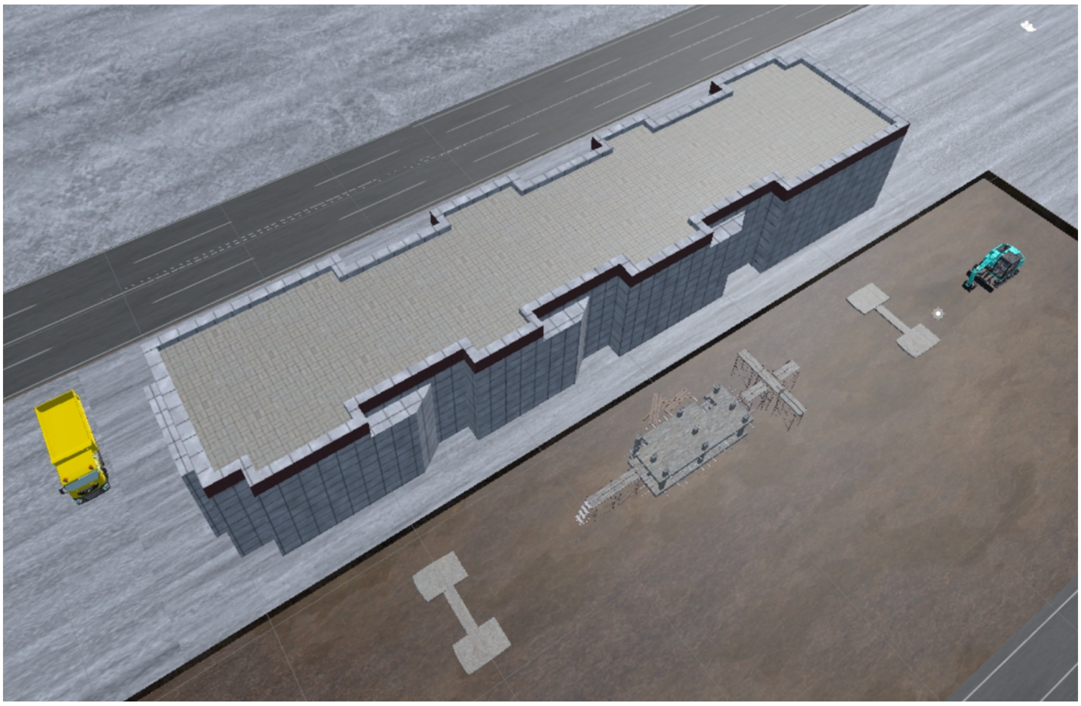

Figure 2: Image of the simulator environment used for the study.

2.2 ORB SLAM

ORB-SLAM systems are extensively employed across various domains, including robotics, autonomous driving, and augmented reality (AR). This is a visual SLAM system capable of real-time operation that supports monocular stereo and red, green, blue, depth (RGB-D) cameras. For the purposes of this study, a stereo camera was selected as the sensor for self-localization and mapping, which uses the 3D information of the environment. ORB SLAM adopts a feature-based approach that involves detecting keypoints using the FAST algorithm and describing these features using BRIEF, thereby enabling efficient and robust tracking and map generation. Furthermore, ORB SLAM incorporates functionalities for loop closure detection and relocalization. Loop closure detection facilitates the identification of instances when the system re-enters a previously mapped area, thereby rectifying accumulated drift errors and enhancing the overall map accuracy. Relocalization refers to the capability of the system to recover from tracking failures by recognizing features from a previously mapped environment and reestablishing its spatial position within the map. These functionalities contribute significantly to maintaining high accuracy even during prolonged operational periods. ORB SLAM was selected for this study because it provides accurate self-localization and map generation in visual SLAM using stereo cameras. It is distinguished by its proficiency in real-time processing and its stable performance in dynamic environments, which align harmoniously with the objectives of this study.

2.3 Generate disparity images and 3D point clouds

Disparity refers to the variation in the image coordinate system when capturing the vertices of a feature from the left and right cameras. This quantifies the positional difference of the same object between images obtained from different viewpoints, resulting in a disparity value for each pixel in the left and right images. Figure 3 illustrates the concept of disparity images. This diagram shows the feature point P through two cameras positioned on the left and right sides. It is assumed that both camera coordinate systems have no rotation around their respective axes (i.e., 0 degrees), with the X-axis of the absolute coordinate system aligned with the line segment (O_1, O_2). The point P is represented as (X_p, Y_p, h) in the absolute coordinate system. At the same time, it is designated as (u_L, v_L) and (u_R, v_R) in the image coordinate systems of the left and right cameras, respectively. Disparity d_P is defined as in Equation (1):

$$d_P = u_L - u_R \tag{1}$$

The distance H can be calculated using Equation (2), where the distance between the plane containing the principal points of the camera and the feature is H, the focal length is f, and the baseline length B:

$$H = \frac{Bf}{d_p} \tag{2}$$

Disparity is inversely proportional to the distance from an object. A larger disparity indicates proximity to an object, whereas a smaller disparity signifies greater distance. In this investigation, the disparity images at each point are combined to extract depth information for the entire construction site. A stereo camera generates these disparate images. The parallel alignment of the optical axes in stereo cameras helps identify the corresponding points between the left and right images, which increases the efficiency of subsequent calculations. In a stereo camera setup, the parallel optical axes ensure that the corresponding points shift primarily in the horizontal direction within the image. This limits the search for the corresponding points to a horizontal range, thus accelerating the computation. In contrast, using a monocular camera to estimate the disparity from images captured from multiple viewpoints requires capturing images from various angles. This necessitates an additional preprocessing step to correct and align them as if the optical axes were parallel. This rectification involves geometric corrections across the images, leading to higher computational costs compared with a stereo camera setup. Consequently, stereo cameras avoid this rectification process, thereby simplifying the identification of corresponding points.

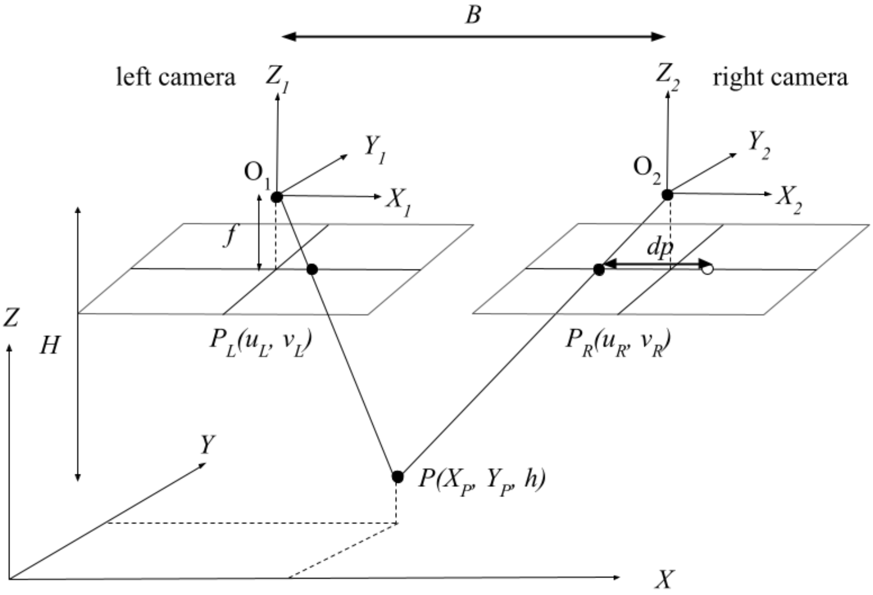

Figure 3: Illustration of disparity image generation using stereo cameras.

Smooth and continuous disparity maps are created while reducing the effect of local noise, thereby achieving high accuracy and in-depth estimation for various scenes. Furthermore, Semi-Global Matching (SGM) (Hirschmüller, 2008) has lower computational demands than full global optimization, making it suitable for real-time processing.

Once the disparity map is generated, the 3D position of feature point P in the camera coordinate system can be calculated. Given a feature point P with its coordinates in the camera coordinate system as (X_P^C, Y_P^C, Z_P^C) and its corresponding image coordinates as (u, v) in the image coordinate system, then (X_P^C, Y_P^C, Z_P^C) can be derived using the intrinsic parameters of the camera as shown in Equation (3). Here, f_x and f_y are the focal lengths along the x and y axes, respectively, and c_x and c_y are the distances from the origin of the image coordinate system to the principal point.

$$\begin{pmatrix} X_P^C / Z_P^C \\ Y_P^C / Z_P^C \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{3}$$

This transformation allows for direct calculation of the depth from the disparity values of each pixel. This process is applied to every pixel in the image, resulting in a 3D point cloud that captures the depth information of the entire scene. The stereo camera setup eliminates the need for image rectification, which is essential in monocular camera systems. This enhances the efficiency of the depth estimation process.

2.4 Integrate 3D point clouds

The 3D point clouds obtained in the previous section are integrated across all the time instances. To achieve this goal, the coordinate system must be unified within a single reference frame. For convenience, the camera position in the first keyframe is set as the origin of the world coordinate system. Figure 4 illustrates the relationship between the positions and orientations of the two cameras at different timestamps. The transformation between two coordinates systems is given in Equation (4). The camera coordinate system of Camera 2 can be transformed into that of Camera 1 using the rotation matrix R and translation vector T. Here, R and T are are derived from the pose estimation provided by ORB SLAM. Given a 3D point P $\left(X_P^{C_2}, Y_P^{C_2}, Z_P^{C_2}\right)$ represented in the coordinate system of Camera 2, the corresponding point in the Camera 1 coordinate system can be expressed as

$$\begin{pmatrix} X_P^{C_1} \\ Y_P^{C_1} \\ Z_P^{C_1} \end{pmatrix} = \boldsymbol{R} \begin{pmatrix} X_P^{C_2} \\ Y_P^{C_2} \\ Z_P^{C_2} \end{pmatrix} + \boldsymbol{T} \tag{4}$$

When integrating 3D point clouds across different keyframes, the overlapping regions between the newly acquired and previously integrated point clouds are handled by taking the average. This transformation is performed on the 3D point clouds obtained for all keyframes, and then integrated into a single coordinate system.

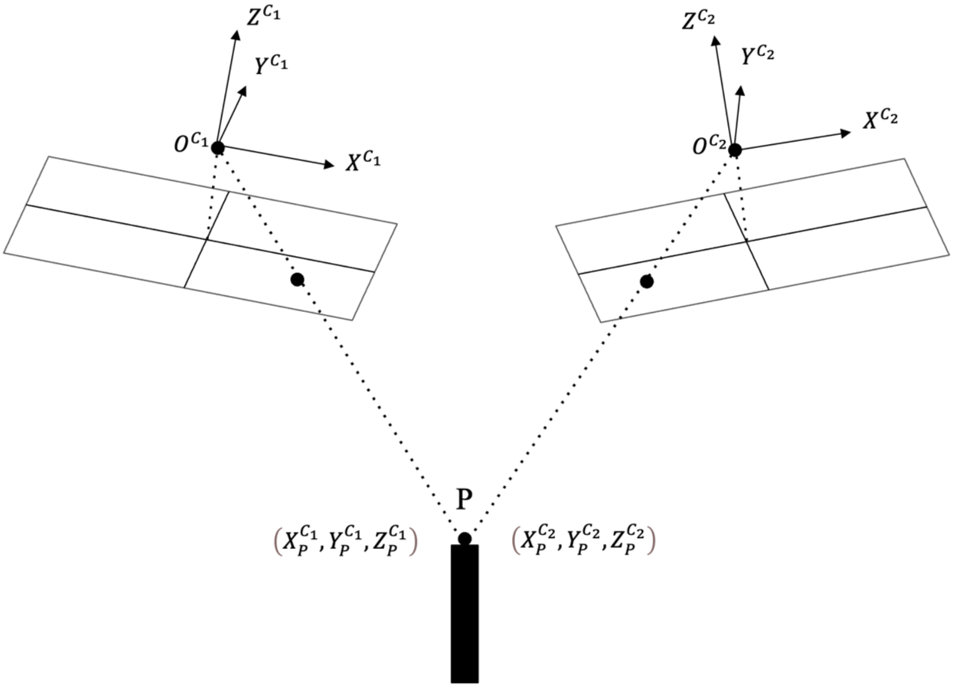
Z^{C_1}
Y^{C_1}
O^{C_1}
X^{C_1}
Z^{C_2}
Y^{C_2}
X^{C_2}
O^{C_2}
P
$\left(X_P^{C_1}, Y_P^{C_1}, Z_P^{C_1}\right)$
$\left(X_P^{C_2}, Y_P^{C_2}, Z_P^{C_2}\right)$

Figure 4: Illustration of the relationship of two cameras.

2.5 Remove noise

The generated 3D point clouds contain noise because of factors such as imperfect image acquisition, errors in disparity estimation, and slight misalignments during point cloud integration. The k-NN method (Altman, 1992) is employed to mitigate this noise. This method is effective for noise removal because it evaluates the local density of points around a target point, allowing for the identification of outliers that deviate significantly from the expected distribution of neighbors. Such outliers typically occur due to measurement errors, sensor limitations, or integration inaccuracies, which result in low-density regions compared to the surrounding points. By defining a threshold based on the average density of neighbors, the algorithm ensures that outliers can be robustly excluded. This approach significantly enhances both the accuracy and usability of 3D point clouds.

The k-NN method is employed to analyze the density of points in the vicinity of each point and to identify outliers that are considered as noise. For a given point Pi, we determine the set of neighboring points $\mathcal{N}(\boldsymbol{P_i})$ based on the Euclidean distance. The Euclidean distances $d(\boldsymbol{P_i}, \boldsymbol{P_j})$ are defined as follows in Equation (5):

$$d(\boldsymbol{P_i}, \boldsymbol{P_j}) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (5)$$

where $\boldsymbol{P_i} = (x_i, y_i, z_i)$ and $\boldsymbol{P_j} = (x_j, y_j, z_j)$ represent the points in a 3D space. Next, the local density around point Pi is evaluated by averaging the distances to neighboring points in $\mathcal{N}(\boldsymbol{P_i})$. The average distance $\overline{d_\iota}$ is calculated as follows in Equation (6):

$$\overline{d_\iota} = \frac{1}{k} \sum_{\boldsymbol{P_j} \in \mathcal{N}(\boldsymbol{P_i})} d(\boldsymbol{P_i}, \boldsymbol{P_j}) \quad (6)$$

If this average distance (d_i)¯ exceeds a predefined threshold τ, the point P_i is classified as noise and removed from the point cloud:

If $\overline{d_\iota} > \tau$, then $\boldsymbol{P_i}$ is classified as noise.

In addition to using the k-NN method for noise removal, we performed downsampling as part of the noise reduction process. After identifying and eliminating noisy points based on local density evaluation, the remaining point cloud is downsampled using a voxel grid approach. This method groups points within a predefined voxel size into a single representative point, thereby effectively reducing the number of points while preserving the overall structure and details of the scene. By doing so, it is easier to update the 3D point clouds.

3. RESULTS AND DISCUSSION

3.1 Simulator environment

In this study, we used a simulation environment developed using Unity. Figure 2 shows an image captured using the developed simulator. The simulated environment included buildings, construction materials, and vehicles replicated at the construction site. A stereo camera was mounted vertically downward on a drone, which can be controlled using the directional keys of the Joy-Con for flight. The drone captured images while orbiting the construction site. The captured left and right images were transmitted from Unity to the ROS. Once the image pairs were sent to the ROS, ORB SLAM was initiated for processing.

3.2 Computational environment

The computational environment used in this study is summarized in Table 1. This setup was used to run ORB SLAM and the associated 3D mapping processes.

Table 1: Summary of the computation environment.

Specification	Detail
CPU	Intel(R) Core (TM) i7-9700
Memory	32 GB
Swap Memory	2 GB
Operating System	Ubuntu 20.04.1 LTS
Kernel Version	5.15.0-107-generic

3.3 Drone flight path

The precision of the 3D mapping was significantly enhanced by optimizing the flight trajectory of the drone. This trajectory incorporates loop closures, which are critical for mitigating the accumulated drift errors within the ORB SLAM framework. By devising a route that enables a return to previously surveyed locations, the system can rectify the positional inaccuracies that may arise during the mapping process. Moreover, the drone maintains a constant altitude, ensuring that the distance between the camera and the ground or targeted objects remains uniform. This stability contributes to the dependable detection of feature points, thereby improving the accuracy of both the disparity images and the resultant 3D point clouds.

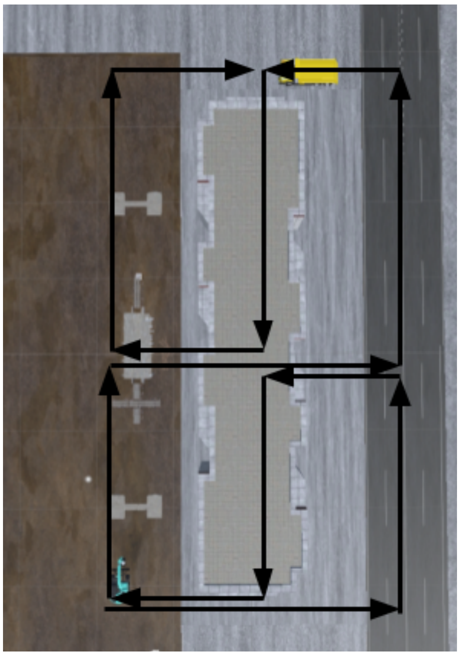

Figure 5: Optimized drone flight path for enhanced 3D mapping accuracy with loop closure considerations.

Figure 5 illustrates the optimized drone flight trajectory, which was specifically designed to enhance the accuracy of 3D mapping. This trajectory is characterized by intentional loop closures, which are vital for minimizing drift errors in the ORB SLAM methodology. The planned route enables the drone to revisit previously mapped areas, facilitating the rectification of positional inaccuracies identified during the initial mapping phase. Additionally, as illustrated by the arrows in the accompanying diagram, maintaining a stable altitude of the drone ensures a consistent distance between the camera and ground or targeted objects. This condition promotes the reliable detection of feature points, ultimately enhancing the accuracy of the disparity images and the resulting 3D point clouds.

3.4 Data

The stereo camera setup was created by aligning two ideal cameras without lens distortion and was provided by Unity by default. The cameras were set to a baseline of 0.3 meters. Table 2 lists the details of the camera settings used in this study.

Table 2: Setting of the stereo camera.

Setting	Value
Focal length [px]	20.78461
Vertical viewing angle [°]	60
Sensor size [mm]	(32, 24)
Number of pixels [px]	(3840,2880)
Frame rate [fps]	30

3.5 Results

Figure 6 illustrates the trajectory of the estimated camera position via ORB SLAM compared with the actual camera trajectory.

Table 3 presents the trajectory of the camera position estimated using ORB SLAM compared with the actual camera trajectory evaluated using the absolute pose error (APE). APE is a metric used to quantitatively assess the error between the actual and estimated camera positions. Let x_i^{gt} represent the coordinates of the actual camera position at the i frame and x_i^{est} represent the estimated camera coordinates. The APE was calculated using the following equation:

$$APE = \left| \boldsymbol{x}_i^{gt} - \boldsymbol{x}_i^{est} \right| \qquad (7)$$

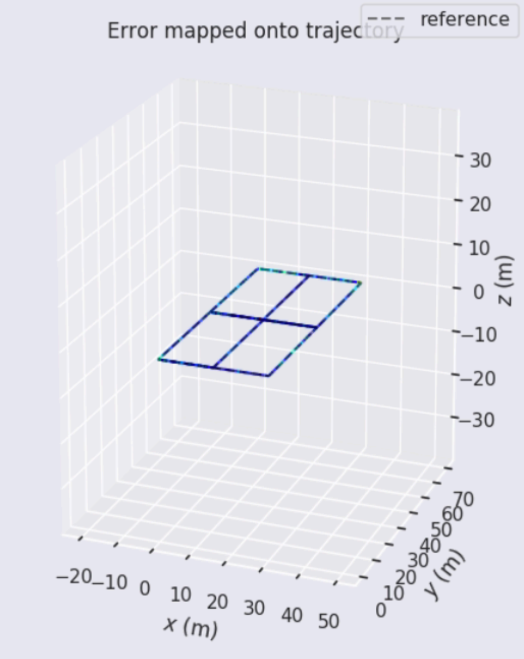

Figure 6: Figure comparing the camera trajectories in a 3D coordinate system. The dotted line represents the trajectory estimated by ORB SLAM, whereas the blue line represents the actual trajectory.

Table3: Summary of the absolute position error between the trajectory of the camera position estimated using ORB SLAM and the actual trajectory of the camera position.

APE	Value [m]
Max value	0.298
Average value	0.030
Median value	0.020
RMSE	0.033

As shown in Equation (7), APE provides a quantitative measure of the difference between the actual and estimated positions, enabling an assessment of the accuracy of the ORB SLAM-based trajectory estimation.

The average APE for a construction site measuring approximately 60 m long and 50 m wide was 0.030 m. Significant errors were observed in the z-direction.

Figure 7 illustrates the 3D point cloud prior to noise removal. Figure 7 shows the integrated 3D point cloud obtained after iterative ORB SLAM-based self-localization and 3D point cloud generation using disparity images. To facilitate an accuracy comparison, Figure 7 was captured from the same viewpoint as the reference 3D model shown in Figure 2. Although the overall features of the model are discernible compared with the 3D model in Figure 2, some noise remains, indicating that the results are not entirely accurate.

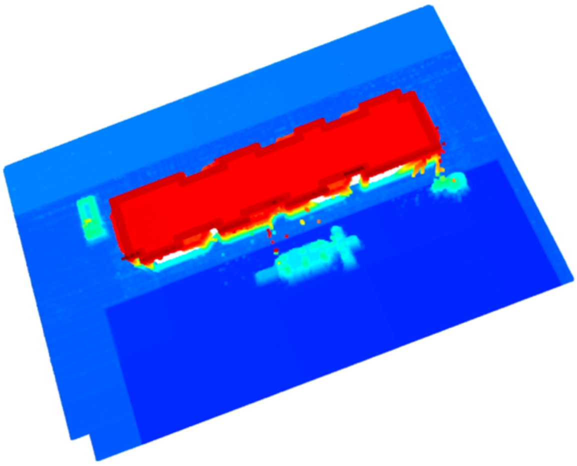

Figure 7: Final 3D point cloud of the simulator.

Finally, we present the results of noise removal using the k-NN method, which effectively cleaned the generated 3D point cloud of the entire construction site. The results are presented in Figure 8. The uneven surfaces of the building and overall shape and scale of the boxes are accurately represented. The process took 545.1 s, meeting the time requirement for generating the initial map, which was typically expected to be within 10 min.

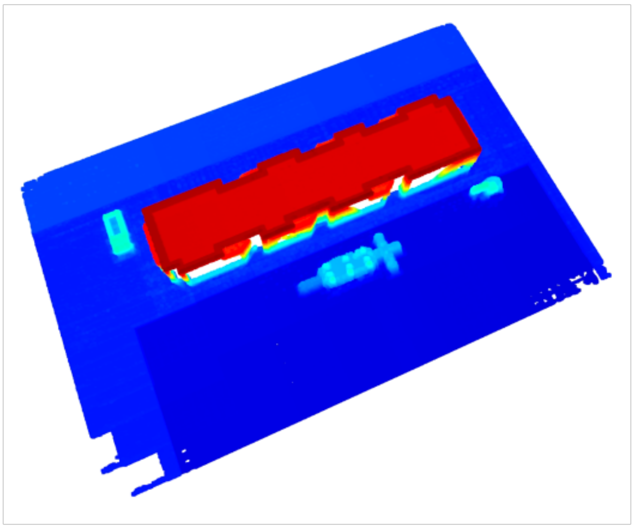

Figure 8: 3D point cloud with noise removed.

4. CONCLUSION AND RECOMMENDATION

In this study, we developed and evaluated a real-time 3D mapping methodology tailored to dynamic environments at construction sites. This methodology employs a stereo camera affixed to a drone and integrates the ORB SLAM technology. The primary objective is to address the limitations of current methodologies in generating precise and dense 3D point clouds for applications such as automated crane operation, where prompt performance and adaptability to continuously evolving environments are essential.

Recent advancements in 3D mapping techniques for construction automation rely predominantly on monocular cameras. However, these methods exhibit several inherent limitations. Monocular cameras cannot determine scale, which complicates the acquisition of precise distance information. In current methodologies, the camera is often mounted on the crane hook, which results in challenges such as reduced map accuracy owing to vibrations from the hook and a constrained field of view resulting from the fixed position of the camera. These physical impediments render these approaches unsuitable for scenarios that require high-density and real-time data acquisition. To address these challenges, our method employs a stereo camera to determine the scale accurately alongside ORB SLAM for real-time self-localization and mapping.

The proposed methodology comprises three primary stages. Initially, dense point clouds are generated by calculating disparity images obtained from a stereo camera affixed to the drone. This stereo camera captures depth information in real time, thereby addressing the limitations inherent in traditional monocular configurations. Subsequently, ORB SLAM is employed for self-localization by utilizing feature-based techniques to accurately estimate the position and orientation of the camera. These estimates facilitate the temporal integration of point clouds, culminating in a comprehensive 3D model of the environment. Finally, noise removal is executed using the k-NN method, which filters outliers by assessing the density of adjacent points. This process ensures that the final 3D map is dense and devoid of prevalent noise issues associated with image acquisition errors, disparities in estimation, and integration misalignments.

We tested the effectiveness of the proposed approach in a simulated construction environment created using Unity. This environment features detailed scene elements, such as buildings, vehicles, and construction materials, that closely resemble a real construction site. A stereo camera with a 0.3-meter baseline was mounted on a drone, which captured images while orbiting the site. The captured frames were processed in the ROS to generate disparity maps, estimate the pose of the drone using ORB SLAM, and integrate the resulting 3D point clouds.

The experimental results indicate that the proposed methodology can generate a highly accurate and dense 3D representation of a construction site in real time. The resulting map effectively captures the irregular surfaces of buildings, accurately reflects the scale and geometry of objects such as boxes and construction equipment and demonstrates minimal deviation in pose estimation when compared to the ground truth. An analysis of the APE confirmed that the average deviation remained within acceptable limits for practical applications, with a root mean square error (RMSE) of 0.033 m throughout the entire trajectory. Furthermore, noise filtering using the k-NN method significantly enhanced the clarity of the point cloud, culminating in a map that closely coincides with the actual structure while effectively eliminating extraneous points.

In the future, we will focus on enhancing the adaptability of the system to dynamic environments. Our objective was to develop algorithms that selectively update only the regions of a 3D map where alterations transpire, thereby facilitating efficient real-time updates at continually evolving construction sites. Furthermore, we intend to broaden our validation efforts from existing simulation environments to real-world contexts to further augment the system's practicality.

REFERENCES

Altman, N. S. 1992. "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression." The American Statistician 46 (3): 175–185.

Barratt, R. P. “Speculating the Past: 3D Reconstruction in Archaeology.” In Virtual Heritage: A Guide, edited by Erik Malcolm Champion, 13–24. Ubiquity Press, 2021.

Campos, C., R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós. 2021. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM." IEEE Transactions on Robotics 37 (6): 1874–1890.

Hirschmüller, Heiko. 2008. "Stereo Processing by Semiglobal Matching and Mutual Information." IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2): 328–341.

Honkavaara, E., J. Kaivosoja, J. Mäkynen, I. Pellikka, L. Pesonen, H. Saari, H. Salo, T. Hakala, L. Marklelin, and T. Rosnell. 2012. "Hyperspectral Reflectance Signatures and Point Clouds for Precision Agriculture by Light Weight UAV Imaging System." In XXII ISPRS Congress 2012: Technical Commission VII, edited by M. Shortis, W. Wagner, and J. Hyyppä, 353–58.

Jacob-Loyola, Nicolás, Felipe Muñoz-La Rivera, Rodrigo F. Herrera, and Edison Atencio. 2021. "Unmanned Aerial Vehicles (UAVs) for Physical Progress Monitoring of Construction" Sensors 21, no. 12: 4227.

Kobayashi, T., Susaki, J., Shigemori, H., Yoneda, T., Ososinski, M. 2023. “High Speed 3D-Mapping around Crane from Video Images of Monocular Camera.” Japanese Journal of JSCE, vol. 79, no.22.

Ministry of Land, Infrastructure, Transport and Tourism. 2023. “Current Status and Issues

Surrounding the Construction Industry." Accessed January 20, 2024. https://www.mlit.go.jp/policy/shingikai/content/001610913.pdf.

Snavely, Noah, Seitz, Steven M. and Szeliski, Richard. 2006. "Photo tourism: Exploring photo collections in 3D." ACM Transactions on Graphics, 835-846.

Tomasi, Carlo, and Takeo Kanade. 1992. "Shape and Motion from Image Streams under Orthography: A Factorization Method." International Journal of Computer Vision 9 (2): 137–154.

Unity Technologies. "Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine." Accessed April 15, 2024. https://unity.com/