

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

Ishiguro, R.^{1*}, Susaki, J.², and Ishii, Y.³

¹ Student, Graduate School of Engineering, Kyoto University, Japan

² Professor, Graduate School of Engineering, Kyoto University, Japan

³ Professor, Graduate School of Management, Kyoto University, Japan

⁴ Assistant Professor, Graduate School of Engineering, Kyoto University, Japan

[*ishiguro.ryunosuke.62w@st.kyoto-u.ac.jp](mailto:ishiguro.ryunosuke.62w@st.kyoto-u.ac.jp) (*Corresponding author's email only)

Abstract *In this paper, we developed a method to create a 3D map in real time using a stereo camera attached to a drone and ORB SLAM. ORB SLAM is a technology that simultaneously estimates the self-location and generates a 3D point cloud of the surrounding environment in real time. The 3D point cloud generated by ORB SLAM is sparse and cannot be used to automate crane operation. Therefore, it is necessary to convert the sparse point cloud into a dense point cloud. MVS is generally used to convert a sparse point cloud into a dense point cloud, but MVS often requires a huge amount of time for calculation and is not applicable to this case where real-time processing is required. This method consists of three processes. The first is a process to generate a dense 3D point cloud using a stereo camera each time. The second is a process to complement the self-estimation of the camera position and orientation and integrate the point clouds. The third is a process to filter the integrated point cloud. Outliers are removed from the point cloud generated by the first two processes. In this method, data obtained from a drone moving on a simulator was used. The simulator includes buildings, cranes, trucks, etc. to simulate an actual construction site. Evaluations on a video of approximately 4,500 frames show that the proposed method achieves the following results in real time and with high accuracy. It takes 25 seconds to create the 3D point cloud, processing approximately 180 images per second. The final generated 3D point cloud also correctly represents the unevenness of the side of the building at the construction site, as well as the overall shape and scale of the box. It is necessary to develop a system that can accurately represent the ever-changing environment of a construction site. In the future, we will work on developing an algorithm that selectively updates only objects whose position has changed on the 3D map.*

Keywords: photogrammetry, ORB SLAM, computer vision, three-dimensional mapping, ROS

Introduction

In recent construction sites, the number of crane operators has been decreasing due to the aging of workers, the declining rate of young people entering the workforce, and the reduction in working hours because of work style reforms are problems. 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 about 21% from the peak at the end of fiscal year 1999. Additionally, the average number of

construction workers in 2022 was 4.79 million, a decrease of about 30% from the average in 1997. One solution to this problem is the automation of crane operation.

For the automation of cranes, it is important to generate a precise 3D map of the environment of a construction site, which changes every moment in real-time. This process involves initially creating a map within 5-10 minutes before the construction begins, followed by updating this map approximately once per second during operations. In this study, we focus on creating an initial map of construction site. For the creation of the initial map, a 3D map generated from a monocular camera attached to the end of the crane boom was used, as in previous research by Kobayashi et al. This research involves grasping the surrounding environment in about 10 minutes by rotating the boom before the crane begins work. However, there are several problems in putting this to practical use. The first is the physical constraint of the camera being attached to the end of the crane hook, which limits the range of the 3D map generated. The second is that the scale cannot be determined when using a monocular camera. These problems are unacceptable considering the goal of automating crane operation. Therefore, we propose a method that solves these problems. Next, we will briefly explain our approach.

To address these issues above, we attached a stereo camera to a drone. By mounting the stereo camera on a drone, we were able to overcome the physical constraints imposed by attaching the camera to the crane hook. Additionally, using a stereo camera solves the issue of scale ambiguity that arises when using a monocular camera. To create the initial map, our approach involves flying the drone around the construction site while capturing images of the surroundings. These images are then processed to generate a comprehensive 3D map of the environment. The method involves estimating the self-location using ORB SLAM, creating a disparity image using a stereo camera attached to a drone, and integrating them. ORB SLAM is a technology that simultaneously performs self-location estimation and generates a 3D point cloud of the surrounding environment in real time. The 3D point cloud generated by ORB SLAM is sparse and cannot be used to automate crane operation. Therefore, it is necessary to convert the sparse point cloud into a dense point cloud. A typical technology for converting a sparse point cloud into a dense point cloud is MVS (Multi-View Stereo). However, MVS calculations often take a huge amount of time, and it cannot be applied to this case, which requires real-time processing. Therefore, we propose a method that uses the position and orientation information of the

camera estimated by ORB SLAM to integrate the 3D point cloud generated using a stereo camera at each time. The stereo camera attached to the drone solves the first problem mentioned above, which is the problem of physical constraints. Self-location estimation and the creation and integration of a 3D point cloud using ORB SLAM solves the second problem, which is the problem of real-time performance.

Literature Review

Kobayashi et al., a 3D mapping method utilizing a monocular camera mounted at the tip of a crane boom was developed. This approach leverages the rotational motion of the crane to scan the surrounding environment over a period of approximately 10 minutes before operations commence, creating an initial 3D map. The camera continuously captures images as the boom rotates, and these images are subsequently integrated to map the environment. The system estimates disparity information from the captured images and integrates this data into a 3D point cloud. A key aspect of this process is the alignment and adjustment of scale and reference planes to reconcile images with varying disparity values. As a result, dense 3D point clouds are generated in quasi-real time, demonstrating improvements in both accuracy and speed compared to conventional methods. However, this method has several limitations: the generated map is restricted by the physical constraints of the camera's attachment point, which is insufficient for the rapidly changing conditions of active construction sites. Additionally, there is the issue that a monocular camera cannot determine the scale because it lacks depth information, making it impossible to accurately estimate the real-world size and distance of objects.

To overcome these limitations, more advanced 3D reconstruction techniques are necessary. Recent developments have introduced promising solutions in both passive and active methods. Passive methods include technologies such as COLMAP, Neural Radiance Fields (NeRF), and Gaussian Splatting. COLMAP combines Structure from Motion (SfM) and Multi-View Stereo (MVS) to generate detailed 3D models from images; however, its computational intensity makes it unsuitable for real-time applications. NeRF utilizes neural networks to model the radiance field of a scene from a sparse set of images, enabling highly detailed 3D reconstructions. Despite its potential, NeRF's high computational requirements limit its applicability in real-time scenarios. Gaussian Splatting offers a faster approach to producing high-quality 3D models, particularly in dynamic scenes, but still faces challenges in real-time processing.

Active methods, on the other hand, include technologies such as LiDAR (Light Detection and Ranging), Structured Light, and Time-of-Flight (ToF) cameras, which emit signals and analyze their reflections to directly measure distances. LiDAR is particularly effective in generating dense 3D point clouds over large areas quickly, making it ideal for real-time applications in dynamic environments. Structured Light and ToF cameras also offer precise 3D mapping capabilities, with ToF cameras being particularly suited for capturing depth information across entire scenes in real-time. However, these active methods typically require expensive equipment and systems, making them costly to implement and operate, which is a significant drawback.

While each of these technologies has its strengths, they also present specific challenges in terms of real-time processing and operational flexibility. Consequently, ORB SLAM (Oriented FAST and Rotated BRIEF SLAM) has emerged as a leading solution for real-time simultaneous localization and mapping. ORB SLAM utilizes monocular, stereo, or RGB-D cameras to achieve visual SLAM with high accuracy and real-time performance. This approach proposes leveraging the advantages of ORB SLAM by utilizing stereo images captured by a drone-mounted camera to generate high-precision, real-time 3D maps of dynamic construction sites, thereby addressing the critical demands of automated crane operations.

Despite significant advancements in 3D mapping technologies, achieving high-density, real-time 3D mapping for dynamic construction environments remains challenging. Building on previous research, this study integrates stereo vision with ORB SLAM to generate efficient 3D point clouds. Additionally, we enhance this approach with noise filtering and real-time processing optimizations. Unlike earlier methods that relied solely on monocular cameras or quasi-real-time processing, our approach leverages a stereo camera mounted on a drone in conjunction with real-time SLAM. This combination allows us to generate and integrate dense point clouds, effectively addressing the physical constraints and processing delays that have hindered prior efforts.

Methodology

a. Overview:

This research introduces a method for generating a 3D map of a construction site using a combination of a Unity-based simulation environment and a ROS (Robot Operating System) framework for image processing. Figure 1 illustrates the overall process flow. The simulated environment, built in Unity, replicates a realistic construction site complete with buildings and vehicles, as depicted in Figure 2. Within this virtual environment, a drone equipped with a stereo camera is operated to capture left and right images of the site, which are subsequently utilized to reconstruct a 3D map. The image processing is handled in ROS, where the mapping workflow begins with estimating the stereo camera's position and orientation using ORB SLAM. This algorithm processes the stereo camera images captured by the drone, providing accurate self-localization information. Once the camera's pose is determined, a disparity image is generated by analyzing the stereo camera's left and right image pairs. This disparity image is then converted into a 3D point cloud for each frame. The generated 3D point clouds at different time steps are integrated using the camera's position and orientation data, ensuring spatial consistency across the sequence. However, due to inherent noise in the point clouds caused by factors such as unintended distortions like pixelation, blurring, or color shifts resulting from sensor limitations or errors in image processing, a noise reduction process is applied. Specifically, the k-nearest neighbor (k-NN) method is employed to filter out erroneous points and enhance the overall quality of the 3D map. The proposed workflow demonstrates a seamless integration of simulation and real-time image processing, allowing for the development and testing of 3D mapping techniques in a controlled environment.

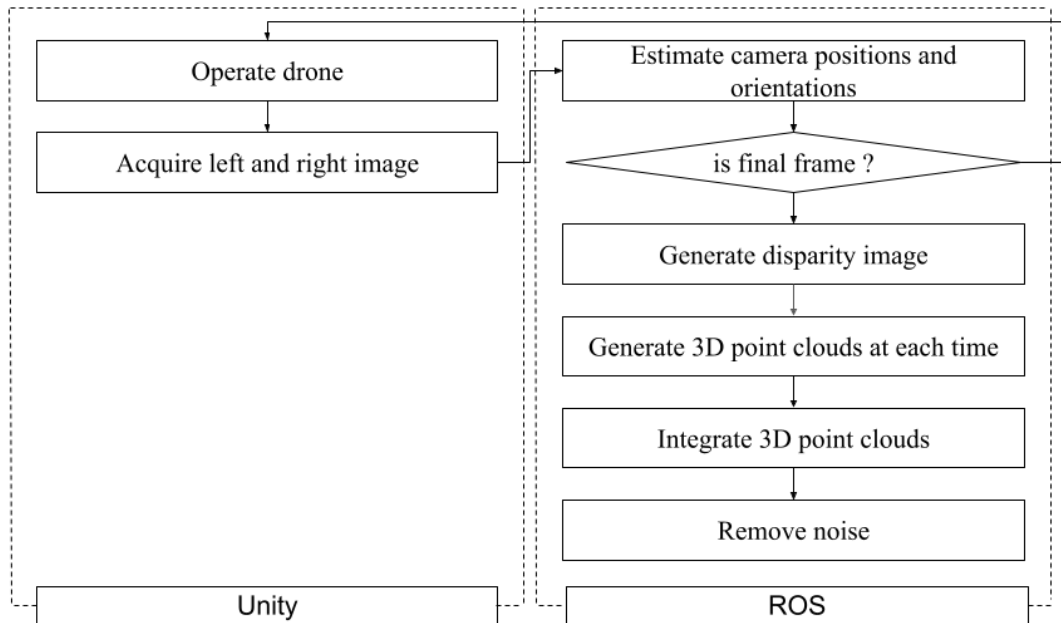


Figure 1: Overview of the method of this study

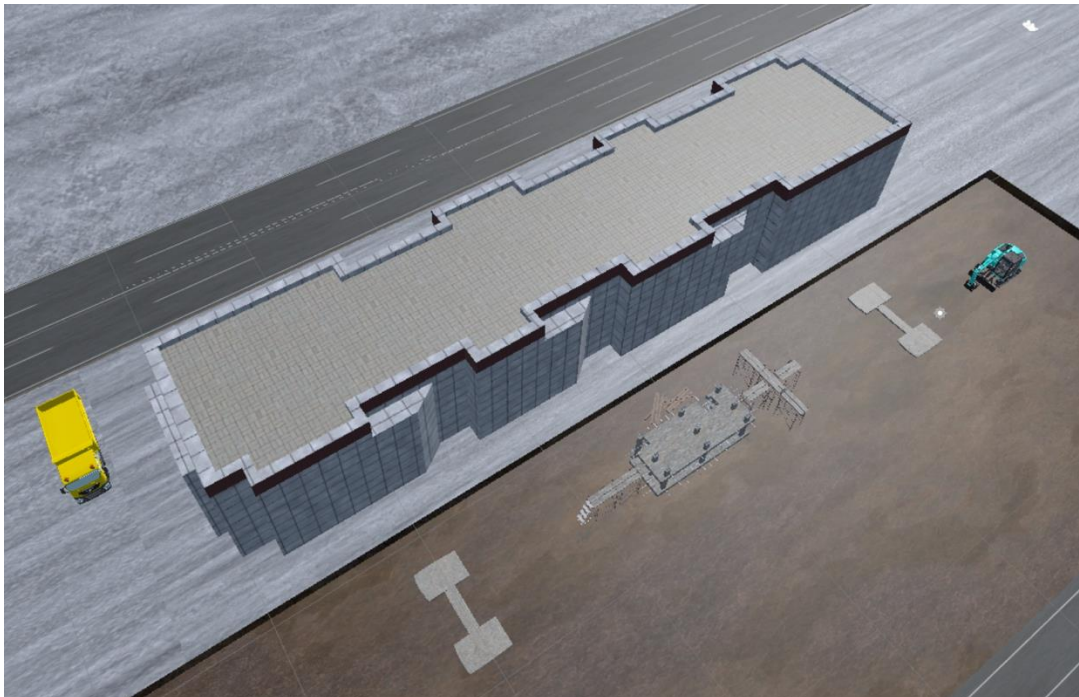


Figure 2: The image of the simulator environment used for the research.

b. ORB SLAM:

This research utilizes ORB-SLAM (Oriented FAST and Rotated BRIEF SLAM) is used as the SLAM (Simultaneous Localization and Mapping) technology. SLAM is a technique that allows a moving agent to estimate its position and simultaneously build a map of an

unknown environment. It is widely utilized in fields such as robotics, autonomous driving, and augmented reality (AR). ORB-SLAM is a visual SLAM system capable of real-time operation and supports monocular, stereo, and RGB-D cameras. In this research, a stereo camera is employed as the sensor to perform self-localization and mapping by leveraging the 3D information of the environment. ORB-SLAM uses a feature-based approach, detecting key points with the FAST algorithm and describing features using BRIEF, enabling efficient and robust tracking and map generation. Additionally, ORB-SLAM includes loop closure detection and relocalization functionalities. Loop closure detection identifies when the system revisits a previously mapped area, allowing it to correct accumulated drift errors and improve overall map accuracy. Relocalization, on the other hand, is the system's ability to recover from tracking failures by recognizing previously mapped features and re-establishing its position within the map. These features ensure high accuracy even during long-term operation. The reason ORB-SLAM was chosen for this research is due to its capability to provide accurate self-localization and map generation in visual SLAM using stereo cameras. ORB-SLAM excels in real-time processing and delivers stable performance in dynamic environments, aligning well with the objectives of this study.

c. Generate disparity image and 3D point clouds:

Disparity refers to the displacement in the image coordinate system when capturing the vertices of a given feature from left and right cameras. It quantifies the positional difference of the same object between images taken 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. In this figure, a feature point P is observed from two cameras positioned on the left and right. Both camera coordinate systems are assumed to have no rotational angle around their respective axes (i.e., 0° rotation), and the X-axis of the absolute coordinate system is aligned with the line segment $\overline{O_1O_2}$. The point P is represented as (X_p, Y_p, h) in the absolute coordinate system, while it is represented as (u_L, v_L) and (u_R, v_R) in the image coordinate systems of the left and right cameras, respectively. The disparity dp is defined as follows:

$$dp = u_L - u_R$$

Given that the distance between the plane containing the camera's principal points and the feature is H , the focal length is f , and the baseline length is B , the distance H can be calculated as follows:

$$H = \frac{Bf}{dp}$$

Disparity is inversely proportional to the distance to an object; the larger the disparity, the closer the object is, while a smaller disparity indicates a farther object. In this study, disparity images at each point are aggregated to obtain depth information for the entire construction site. A stereo camera is employed to generate the disparity images. Because the optical axes of the stereo cameras are parallel, it simplifies the search for corresponding points between the left and right images, making the calculations more efficient. In a stereo camera setup, the parallel optical axes ensure that corresponding points shift predominantly in the horizontal direction within the image. This allows the search for corresponding points to be constrained to a horizontal range, thereby speeding up the computation. In contrast, when using a monocular camera to estimate disparity from images captured at multiple viewpoints, the images are taken from different angles, requiring an additional preprocessing step to rectify and align them as if the optical axes were parallel. This rectification involves geometric corrections across the images, leading to additional computational costs compared to a stereo camera setup. As a result, stereo cameras inherently avoid the need for this rectification process, making the search for corresponding points more straightforward.

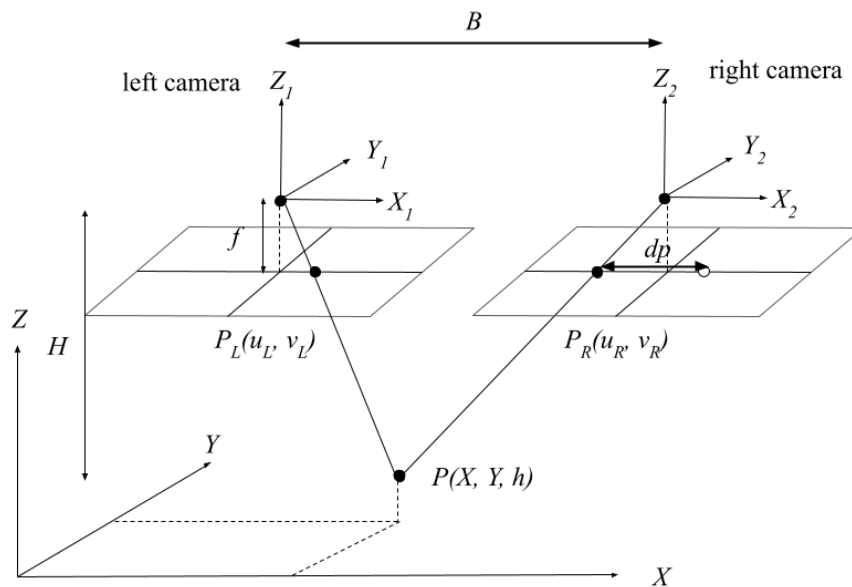


Figure 3: Illustration of Disparity Image Generation Using Stereo Cameras.

Semi-Global Matching (SGM) is employed for generating disparity images. SGM is a disparity estimation method in stereo vision that optimizes the disparity map by accumulating costs from multiple directions. Specifically, it computes the disparity cost

from eight different directions and integrates them to determine the disparity value with the minimum energy for each pixel. This approach allows for the creation of smooth and continuous disparity maps while mitigating the influence of local noise, thereby achieving high accuracy in depth estimation for scenes. Additionally, SGM is computationally less demanding than full global optimization, making it suitable for real-time processing.

Once the disparity map is generated, the 3D position of a 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 camera's intrinsic parameters. 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.

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

From this transformation, Z_P^C , which represents the depth, can be directly calculated from the disparity values for each pixel. This process is repeated for every pixel in the image, creating a 3D point cloud that represents the depth information of the entire scene. The stereo camera setup avoids the need for image rectification that would otherwise be required in a monocular camera system, thus making the depth estimation process more efficient.

d. Integrate 3D point clouds:

The 3D point clouds obtained in previous section are integrated across all time instances. To achieve this, the coordinate systems must be unified under a single reference frame. For convenience, the position of the camera 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 two cameras at different timestamps. The camera coordinate system of Camera 2 can be transformed into that of Camera 1 using a rotation matrix R and a translation vector T . Here, R and T are derived from the pose estimation provided by ORB-SLAM. Given a 3D point P $(X_P^{C2}, Y_P^{C2}, Z_P^{C2})$ represented in the coordinate system of

Camera 2, the corresponding point in Camera 1's coordinate system can be expressed as follows:

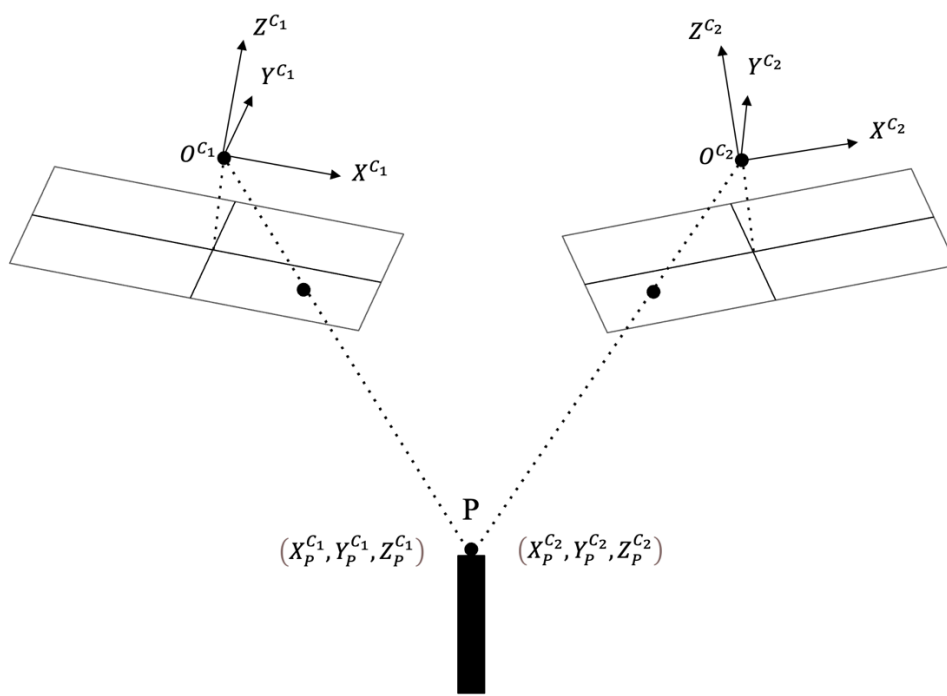
$$\begin{pmatrix} X_p^{C_1} \\ Y_p^{C_1} \\ Z_p^{C_1} \end{pmatrix} = \mathbf{R} \begin{pmatrix} X_p^{C_2} \\ Y_p^{C_2} \\ Z_p^{C_2} \end{pmatrix} + \mathbf{T}$$


Figure 4: The image of the simulator environment used for the research.

When integrating the 3D point clouds across different keyframes, overlapping regions between the newly acquired point cloud and the 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.

e. Remove noise:

The generated 3D point clouds contain noise due to factors such as imperfect image acquisition, errors in disparity estimation, and slight misalignments during point cloud integration. To mitigate this noise, we employ the k-NN method. The k-NN algorithm analyzes the density of neighboring points around each point and removes outliers that exceed a predefined threshold. This approach significantly enhances both the accuracy and usability of the 3D point clouds.

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

$$d(\mathbf{P}_i, \mathbf{P}_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

where $\mathbf{P}_i = (x_i, y_i, z_i)$ and $\mathbf{P}_j = (x_j, y_j, z_j)$ represent points in a three-dimensional space. Next, the local density around point \mathbf{P}_i is evaluated by averaging the distances to the neighboring points in $\mathcal{N}(\mathbf{P}_i)$. The average distance \bar{d}_i is calculated as follows:

$$\bar{d}_i = \frac{1}{k} \sum_{\mathbf{P}_j \in \mathcal{N}(\mathbf{P}_i)} d(\mathbf{P}_i, \mathbf{P}_j)$$

If this average distance \bar{d}_i , exceeds a predefined threshold τ , the point \mathbf{P}_i is classified as noise and removed from the point cloud:

If $\bar{d}_i > \tau$, then \mathbf{P}_i is classified as noise.

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

Results and Discussion

a. Simulator environment:

In this study, we utilized a simulator environment developed with Unity. Figure 2 shows an image captured within the developed simulator. The simulated environment includes buildings, construction materials, and vehicles that replicate an actual construction site. A stereo camera is mounted vertically downward on a drone, which can be controlled using the directional keys of a Joy-Con for flight. The drone captures images while orbiting the construction site. The captured left and right images are transmitted from Unity to ROS. Once the image pairs are sent to ROS, ORB SLAM is initiated for processing.

b. Computational Environment

The computation 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	31GB
Swap Memory	2GB
Operating System	Ubuntu 20.04.1 LTS
Kernel Version	5.15.0-107-generic

c. Drone Flight Path

The accuracy of 3D mapping is further enhanced by optimizing the drone's flight path. Specifically, the flight path is designed to include loop closures, which are essential for reducing accumulated drift errors in ORB SLAM. This is achieved by planning a flight route that returns to previously visited locations, allowing the system to correct any positional errors that may have occurred during the mapping process. Additionally, the drone maintains a constant altitude during flight, ensuring that the distance between the camera and the ground or target objects remains uniform. This consistent distance aids in the stable detection of feature points, thereby improving the accuracy of both the disparity images and the resulting 3D point clouds. Figure 5 illustrates the optimized drone flight path designed to enhance the accuracy of 3D mapping. As shown in the figure, the flight path includes deliberate loop closures, which are crucial for minimizing accumulated drift errors in ORB SLAM. The planned route ensures that the drone revisits previously mapped areas, allowing the system to correct any positional errors that may have occurred during the initial mapping process. Furthermore, the drone maintains a consistent altitude throughout the flight, as indicated by the arrows in the diagram. This constant distance between the camera and the ground or target objects facilitates the stable detection of feature points, ultimately improving the accuracy of the disparity images and the resulting 3D point clouds.

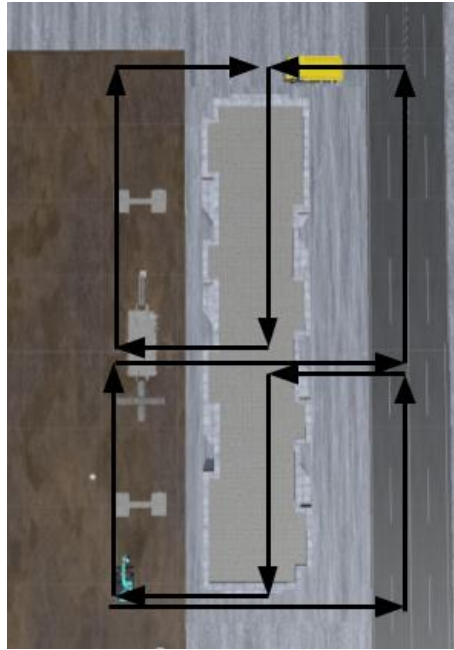


Figure 5: Optimized Drone Flight Path for Enhanced 3D Mapping Accuracy with Loop Closure Considerations.

d. Data:

The stereo camera setup is realized by aligning two ideal cameras, without lens distortion, provided by Unity by default, with a baseline of 0.3 meters. Table 2 details 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 [px]	(32, 24)
Number of pixels [px]	(640, 320)
Frame rate [fps]	30

e. Results:

Figure 6 illustrates the trajectory of the camera's estimated position via ORB SLAM compared to the actual camera trajectory.

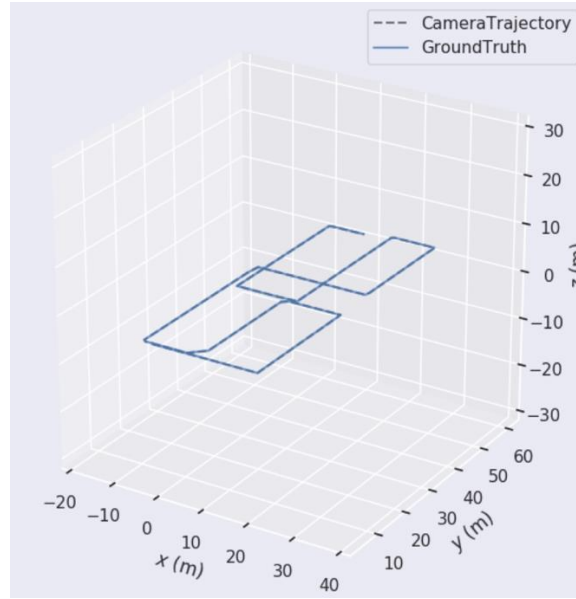


Figure 6: The figure compares the camera trajectories in a 3D coordinate system. The dotted line represents the camera trajectory estimated by ORB SLAM, while the blue line represents the actual camera trajectory.

Table 3 presents the trajectory of the camera's estimated position via ORB SLAM compared to the actual camera trajectory, evaluated using Absolute Pose Error (APE). APE is a metric used to quantitatively assess the error between the actual camera position and the estimated position. Let x_i^{gt} represent the coordinates of the actual camera position at the i frame, x_i^{est} represent the estimated camera coordinates, and N denote the total number of frames in the sequence. APE is then calculated using the following equation:

$$APE = \sum_{i=1}^N |x_i^{gt} - x_i^{est}|$$

Table 3: This table summarizes 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

The average APE was 0.030 meters for a construction site area measuring approximately 60 meters in length and 50 meters in width. Notably, significant errors were observed in the z-axis direction.

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

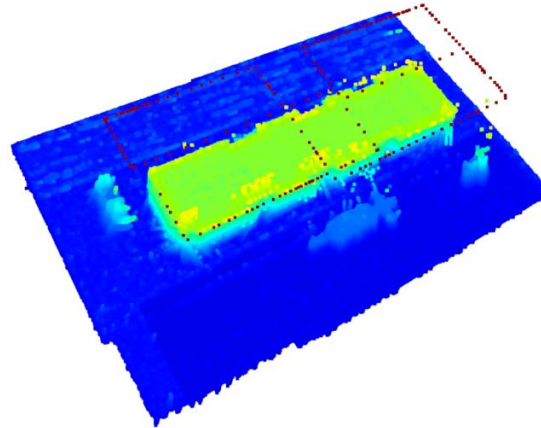


Figure 7: The final 3D point cloud of the simulator.

Finally, we present the results of noise removal. Using the k-NN method, noise was removed from the generated 3D point cloud of the entire construction site. The results are shown in Figure 8. The uneven surfaces of the buildings and the overall shape and scale of the boxes were accurately represented. The entire processing took 211 seconds, which meets the time requirement for generating the initial map, typically expected to be within 5 to 10 minutes.

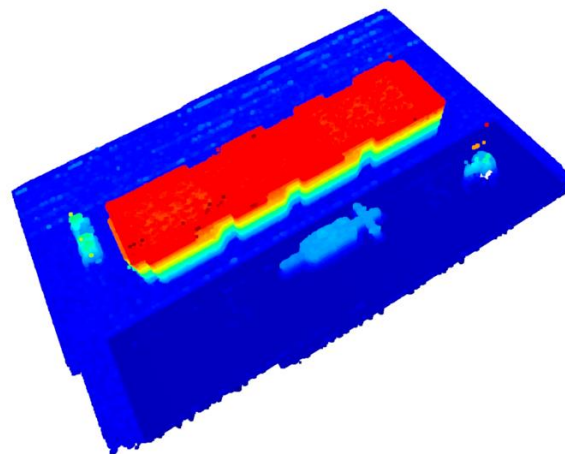


Figure 8: 3D point cloud with noise removed.

Conclusion and Recommendation

In this study, we developed and evaluated a real-time 3D mapping method tailored for dynamic construction site environments using a stereo camera mounted on a drone in combination with ORB SLAM. The primary objective was to address the limitations of existing methods in generating accurate and dense 3D point clouds for applications like automated crane operation, where real-time performance and adaptability to ever-changing environments are critical.

Recent 3D mapping methods for construction automation primarily rely on monocular cameras and have several significant limitations. Monocular cameras cannot determine scale, making it challenging to obtain accurate distance information. Additionally, in existing methods, the camera is often mounted on the crane's hook, leading to issues such as reduced map accuracy due to hook vibrations and limited field of view due to the fixed camera position. These physical constraints make such approaches unsuitable for scenarios that require high-density and real-time data. To address these challenges, our approach utilizes a stereo camera to accurately determine scale and employs ORB SLAM for real-time self-localization and mapping.

Our proposed method operates in three main stages. First, dense point clouds are generated by calculating disparity images using a stereo camera mounted on a drone. The stereo camera allows for capturing depth information in real-time, addressing the limitations posed by traditional monocular setups. Next, ORB SLAM is employed for self-localization, using feature-based methods to estimate the camera's position and orientation with high precision. These estimates are used to integrate the point clouds over time, effectively building a comprehensive 3D model of the environment. Finally, noise removal is performed using the k-NN method, which filters out outliers by analyzing the density of surrounding points. This ensures that the final 3D map is not only dense but also free from the common noise issues that arise from image acquisition errors, disparity estimation inaccuracies, and integration misalignments.

The effectiveness of our approach was tested using a simulated construction environment developed in Unity. The environment includes detailed elements such as buildings, vehicles, and construction materials, closely mimicking a real construction site. A stereo camera with a baseline of 0.3 meters was mounted on a drone, which was controlled to

capture images while orbiting the site. The captured frames were processed in ROS to generate disparity maps, estimate the drone's pose using ORB SLAM, and integrate the resulting 3D point clouds.

Our experimental results demonstrate that the proposed method can generate a highly accurate and dense 3D map of a construction site in real time. The generated map successfully captures the uneven surfaces of buildings, correctly represents the scale and geometry of objects such as boxes and construction equipment and shows minimal deviation in pose estimation when compared with the ground truth. The Absolute Pose Error (APE) analysis confirmed that the average deviation remained within acceptable limits for practical applications, with a root mean square error (RMSE) of 0.033 meters over the entire trajectory. The noise filtering using k-NN further enhanced the clarity of the point cloud, producing a map that closely aligns with the real-world structure while removing spurious points.

Looking ahead, we will focus on further enhancing the adaptability of the system to dynamic environments. We aim to develop algorithms that can selectively update only the regions of the 3D map where changes have occurred, enabling efficient real-time updates in continuously evolving construction sites. Additionally, we plan to extend our validation from the current simulation environment to real-world scenarios to further improve the system's practicality.

References

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis., ECCV 2020, vol. 12346, 2020.

C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Mon-tiel and J. D. Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM, IEEE Transactions on Robotics, vol. 37, No. 6, pp. 1874-1890, 2021.

D. G. Lowe: Object recognition from local scale-invariant features, Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150-1157, 1999.

J.L. Schönberger and J.-M. Frahm: Structure-from-motion revisited, IEEE Conf. Computer Vision and Pattern Recognition, pp. 4104–4113, 2016.

Kobayashi, T. (2021). Creation of a real-scale 3D ortho map around a crane from videos considering construction sites. Master's thesis, Department of Urban Management, Graduate School of Engineering, Kyoto University, pp. 1874-1890.

Ministry of Land, Infrastructure, Transport and Tourism: Current status and issues surrounding the construction industry,

<https://www.mlit.go.jp/policy/shingikai/content/001610913.pdf>, (viewed 2024.1.20).

OpenCV team: OpenCV - Open Computer Vision Library, <<https://opencv.org/>> , (viewed 2024.2.5)

P. F. Alcantarilla, A. Bartoli, and A. J. Davison: KAZE features, Computer Vision – ECCV 2012, vol. 7577, pp. 214–227, 2012.

Unity Technologies: Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine, <<https://unity.com/>>, (viewed 2024.4.15).