https://doi.org/10.3311/PPci.38854 | 447
Creative Commons Attribution

Periodica Polytechnica Civil Engineering, 69(2), pp. 447–460, 2025

Image Acquisition of Critical Bridge Components Using Vision-guided Autonomous Vehicle

Linlong Meng[1], Yuxuan Li[1,2], Shijie Zhou[1], Liangjing Yang[3], Yuki Nishimura[4], Yasutaka Narazaki[1*]

[1] Department of Civil and Environmental Engineering, Zhejiang University/University of Illinois Urbana-Champaign Institute, Zhejiang University, 718 East Haizhou Rd., 314400 Haining, Zhejiang, China
[2] Department of Civil Engineering, College of Civil Engineering and Architechture, Zhejiang Univiersity, 866 Yuhangtang Rd., 310058 Hangzhou, Zhejiang, China
[3] Department of Mechanical Engineering, Zhejiang University/University of Illinois Urbana-Champaign Institute, Zhejiang University, 718 East Haizhou Rd., 314400 Haining, Zhejiang, China
[4] School of Mechanical Engineering, Hangzhou Dianzi University, Xiasha Higher Education Zone, 310018 Hangzhou, Zhejiang, China
[*] Corresponding author, e-mail: narazaki@intl.zju.edu.cn

Received: 22 November 2024, Accepted: 03 January 2025, Published online: 21 January 2025

Abstract

This research proposes a vision-guided autonomous navigation framework for unmanned vehicles performing image acquisition for bridge inspection. The proposed framework integrates visual SLAM with RGB-D image input with semantic segmentation to detect and localize critical structural components like columns. The detected components are converted to the parametric map to generate navigation goals for image collection. The proposed approach is first validated in the synthetic bridge inspection environment using an unmanned ground vehicle. The feasibility of the framework is further studied by the laboratory-scale prototyping and validation using TurtleBot3 equipped with Jetson TX2 onboard computer. In the simulation environment, the proposed framework can achieve autonomous navigation to up to 6 columns and acquisition of image data with 90% success rate for 3 columns. Furthermore, the performance evaluation in the real-world environment shows that the developed hardware-software prototype can navigate and collect image data of up to 2 columns, with more than 60% success rate navigating to the first column. The results indicate the significant potential of achieving autonomous navigation and image acquisition with limited onboard computational resources, contributing to the enhanced efficiency and reliability of bridge management.

Keywords

bridge inspection, autonomous structural inspection, unmanned vehicle, semantic segmentation, autonomous navigation planning

1 Introduction

Bridges play a critical role in road and rail transportation, and the health of bridge structures is the basis for ensuring the safety of transportation. By the end of 2022, the total number of highway and railway bridges in China has exceeded 1.2 million [1]. Many of those bridges experience structural degradation as their duration of service increases [2]. If such degradation is not identified and managed appropriately, bridge safety may be compromised, potentially resulting in fatalities and injuries [3].

In recent years, computer vision (CV)-based structural inspection and monitoring approaches have been investigated as an alternative to traditional approaches that require dense instrumentations (e.g., [4]). For example, visual recognition algorithms today can extract structural component types and structural damage from the images of critical parts of the structures [5–7]. Successful results from those applications push the need for higher levels of automation, where image data collection in the bridge inspection environment is performed by mobile robots with visual recognition capabilities.

This research proposes a vision-based autonomous navigation planning approach that can be deployed on small, unmanned vehicles to acquire images in bridge inspection environments. The proposed approach first combines visual simultaneous localization and mapping (visual SLAM) and semantic segmentation of bridge components to obtain parsed sparse point cloud map of the critical structural components (e.g., columns). Then, bounding boxes representing those critical structural components are fitted to the parsed point cloud data. The bounding boxes are used to determine the navigation goals for collecting images from the desired distance to the target surface. This research investigates the feasibility of the proposed approach through hardware implementation using

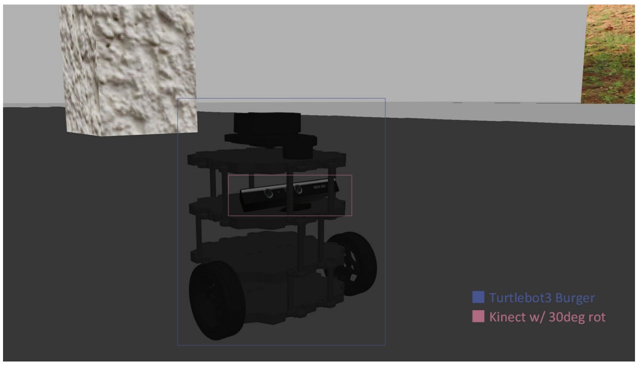
Cite this article as: Meng, L., Li, Y., Zhou, S., Yang, L., Nishimura, Y., Narazaki, Y. "Image Acquisition of Critical Bridge Components Using Vision-guided Autonomous Vehicle", Periodica Polytechnica Civil Engineering, 69(2), pp. 447–460, 2025. https://doi.org/10.3311/PPci.38854

TurtleBot3 Waffle Pi unmanned ground vehicle (UGV) [8] and a one-third-scale laboratory specimen of a reinforced concrete (RC) railway viaduct [9]. The results highlight the potential of collecting image data autonomously and effectively in the actual bridge inspection scenarios.

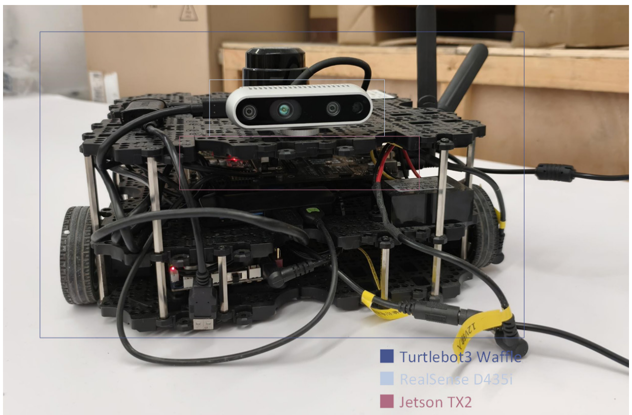
This next section discusses the related work, followed by the detailed description of the proposed autonomous navigation planning framework. Then, feasibility, potential, and challenges of the proposed framework is discussed by developing prototypes (Fig. 1) in synthetic and laboratory environments (Fig. 2).

2 Related works

Computer vision-based structural inspection has been investigated actively to reduce the workload and subjectivity of manual visual inspection [7, 10]. Deep learning-based algorithms, such as convolutional neural networks (CNN) and vision transformers (ViT), are known to be effective in automatically extracting information about structural conditions from images taken appropriately [5, 11, 12]. However, manual image collection process for large and complex civil structures requires significant labor and potentially raises safety concerns.



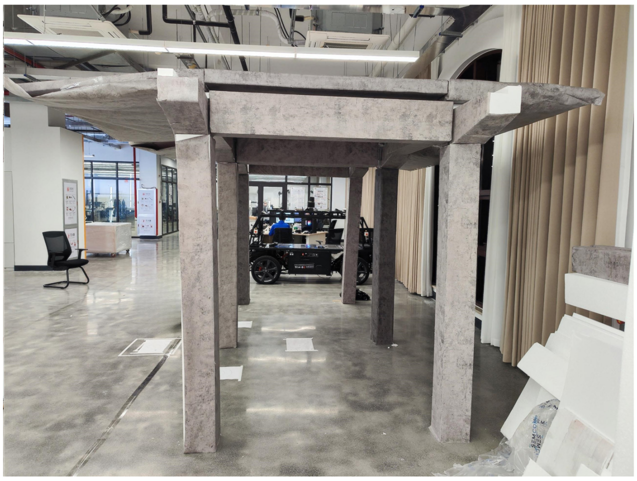
(a)



(b)

Fig. 1 Robot configuration: (a) Configuration in simulation environment, (b) Configuration in laboratory environment

(a)



(b)

Fig. 2 Laboratory environment with a specimen of a high-speed railway viaduct: (a) Front view, (b) Side view

The use of mobile robots, including UGVs and Unmanned Aerial Vehicles (UAVs), can address some of the shortcomings of manual image collection process. Unmanned vehicles can be operated remotely to collect high-quality images of civil structures [13–15]. Image data collected by UAVs can be used for critical component recognition [6, 16, 17], damage detection [5, 18, 19] etc. Existing research implies that the efficiency and reliability of the bridge inspection process would be improved further if the autonomous navigation and image collection by unmanned vehicles are realized.

GNSS positioning and satellite maps are used frequently to automate the UAV-based collection. Morgenthal et al. [20] proposed bridge inspection framework, in which a UAV collects images automatically, following predefined flight trajectories derived from a 3D bridge model using the Global Navigation Satellite System (GNSS). Lin et al. [21] proposed a satellite map-based bridge inspection mission planner which generates 3D waypoints according to the selected map area and data collection requirements. However, mobile robot positioning and navigation in GNSS-denied areas (e.g., under bridges) should be based on onboard sensors, such as cameras and LiDAR.

Mobile robot navigation paths for data collection can be planned without explicit use of the GNSS by leveraging accurate pre-built maps or Building Information Modeling (BIM) models. Bolourian and Hammad [22] presented a 3D path planning method for UAV-assisted bridge inspection using a LiDAR scanner. The method optimizes flight time while ensuring maximum visibility of potential defect locations. Prieto et al. [23] proposed a BIM-based framework for data collection in the indoor environment using multiple robots. Asadi et al. [24] investigated a navigation method for data collection in an indoor environment using a combination of UGV and UAV. In this approach, a UGV explores the environment to create a 2D map, based on which the UAV and UGV perform data collection cooperatively. These navigation path planning methods require complete pre-built models or maps, making the methods potentially inefficient and uncertain for complex bridge inspection environments.

Another approach plans mobile robot navigation paths on-site based on the robot's own sensor inputs, without fully relying on pre-built models or maps. Car et al. [25] proposed a semi-autonomous UAV navigation approach for the wind turbine blade inspection. The method performs plane detection and relative distance/heading adjustments to maintain the constant position relative to the blade. Xu et al. [26] incorporated CNN into real-time power line detection and 3-D point set construction using UAVs. In this approach, the UAV is guided to focus its camera on the center of the tower and to traverse a path aligned with the direction of the power lines. Those studies presented successful preliminary results toward the automation of inspections for relatively simple structures, such as wind turbine blades and power line cables; however, direct extensions of those approaches to more complex bridge structures are not straightforward.

Research about fully autonomous navigation planning (online and onsite planning based on robot's own sensors) for bridge inspection is relatively sparse in the field. Peng et al. [27] developed an autonomous UAV platform for post-disaster bridge inspection. Their approach uses an onboard 3D lidar sensor to map the environment and to detect structural damage. Narazaki et al. [28] proposed a vision-based autonomous navigation approach for post-earthquake inspection of railway viaduct columns using UAVs. This approach can plan UAV navigation paths without relying on known models or maps. However, those investigations are limited to conceptual and theoretical development. Further investigations are required to realize the autonomous bridge inspection based on real-world online navigation for image acquisition, requiring

- efficient interfacing of different hardware and software components,
- real-time data processing with limited onboard computational resources,
- dealing with lower perception and robot localization accuracies in the real-world compared to the simulation environments.

This research aims at developing an autonomous unmanned vehicle navigation framework and its laboratory-scale prototype for image acquisition of critical bridge components. The proposed framework combines image semantic segmentation and sparse point cloud map to detect critical structural components (e.g., columns), and computes navigation paths for collecting images from desired distance and view angles. The efficiency and feasibility of the proposed framework is investigated using TurtleBot3 Waffle Pi with NVIDIA Jetson TX2 [29]. To address the issues of computational efficiency, this research optimizes the combination of visual SLAM algorithm, image semantic segmentation network architecture, and column shape fitting algorithm based on the performance evaluation on the selected hardware. To reduce the performance gap between simulation and real-world environments, the semantic segmentation network is trained by an unsupervised domain adaptation (UDA) approach [30, 31] with the rich synthetic dataset [9]. All those data processing steps are interfaced by Robot Operating System (ROS) [32]. Based on the parsed sparse cloud map obtained from SLAM and semantic segmentation (sufficient image overlaps and correct camera pose estimation are identified to be critical therein), the proposed framework can execute the autonomous navigation for image data acquisition. This research contributes to the realization of the fully autonomous "inspection robot" by the wholistic development and feasibility study.

3 An autonomous unmanned vehicle navigation framework for bridge inspection

3.1 Framework overview

An overview of the proposed framework for autonomous unmanned vehicle navigation for the image acquisition in unknown (without pre-built maps or models) bridge inspection environments is shown in Fig. 3. In this framework, waypoints are determined based on the online processing of RGB-D image stream from an onboard camera to collect close-up images or videos of critical bridge components (e.g., columns). The framework consists of three modules interacting with each other: visual SLAM, bridge component recognition, and target generation and navigation. The image frame is first processed by visual SLAM algorithm to update the robot pose and the sparse point cloud map of the environment. At the same time, a semantic segmentation model is applied to obtain the bridge component label map. The target generation and navigation module accesses the visual SLAM and bridge component recognition results to parse the sparse point cloud map into different bridge component types. When the number of points for the critical component class reaches the threshold, the point cloud for that component class is clustered into component instances. These instances are converted into a parameterized component map, which is used to determine the waypoints for collecting images of those components from desirable distance and view angles. This map and the associated waypoints are updated continuously, until all the components are detected and inspected. The detail of each module is described in the following sections.

3.2 Robot Operating System

Robot Operating System (ROS) [32] is an open-source software development platform that allows programs with different functions (referred to as "nodes") to be interconnected via standardized communication protocols. For example, this research runs visual SLAM, semantic segmentation, and navigation target generation as nodes in the ROS framework, ensuring that those algorithms can coordinate effectively to perform the autonomous bridge inspection task.

3.3 Visual SLAM module

The visual SLAM algorithm estimates the camera pose and updates the point cloud map simultaneously by processing RGB or RGB-D image stream. This research evaluates the following three SLAM algorithms, OV^2SLAM [33], ORB-SLAM3 [34] with RGB image input, and ORB-SLAM3 with RGB-D image input.

OV^2SLAM [33] is a SLAM algorithm that combines optical flow with feature extraction. The algorithm achieves efficient tracking and localization by adopting the fast optical flow algorithm, while performing feature matching only on keyframes. ORB-SLAM3 [34] is a SLAM algorithm that relies on ORB (Oriented FAST and Rotated BRIEF) [35] features and utilizes co-visible views for backend local bundle adjustment. Extracting ORB features from every frame results in a relatively dense point cloud map at the expense of increased processing overhead. ORB-SLAM3 also supports map merging and loop detection, improving the tracking and mapping results. ORB-SLAM3 can be applied to RGBD image data, allowing the localization and mapping results to reflect the actual scale. Use of RGBD image data can also improve the accuracy of triangulation-based depth reconstruction during small movements.

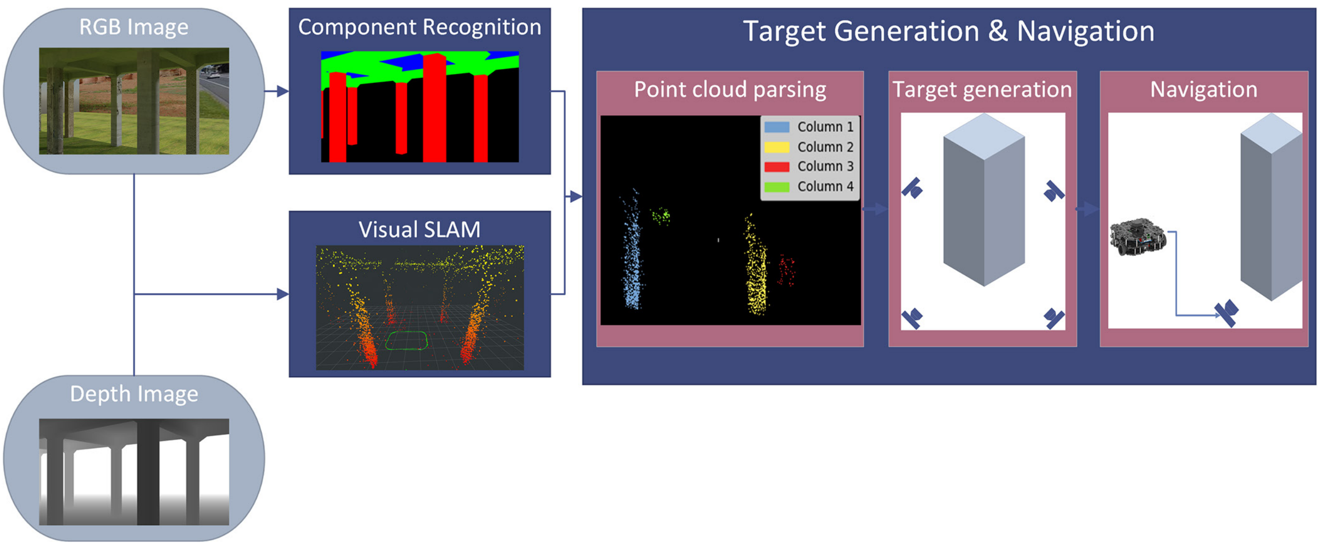


Fig. 3 Overview of the proposed UGV-based autonomous navigation and image acquisition system under bridge inspection task (Dark blue blocks are the main modules of our approach)

When the SLAM algorithm processes the new image frame, the image point coordinates corresponding to the 3D points are computed using the estimated camera projection matrix. These image coordinates are published as ROS messages, allowing other modules to access the information in real-time.

3.4 Bridge component recognition module

This module performs semantic segmentation to identify critical structural components that need to be inspected. The proposed approach first performs 2D semantic segmentation of each frame of the image stream, and merges the results into point cloud to obtain parsed map of the environment. Compared to semantic segmentation methods that operate directly on point cloud data, e.g., PointNet++ [36], this image semantic segmentation-based approach has the following advantages:

1. semantic segmentation of a new incoming image frame tends to be computationally more efficient than performing point cloud semantic segmentation of the entire scene,
2. point cloud map creation and parsing can be performed online, without waiting for the robot to finish exploring the environment, and
3. images tend to provide richer contextual information compared to the sparse and incomplete point cloud map obtained by SLAM algorithms.

This research trains the semantic segmentation algorithm by a UDA approach, termed DAFormer [30, 31], with a large-scale synthetic dataset, termed Tokaido Dataset [9]. The DAFormer framework combines multiple UDA techniques, such as self-training, rare-class sampling, and learning rate warmup, to obtain high-performance semantic segmentation networks for the target domain (real-world images). The training is based on annotated data in the source domain (synthetic data) and unannotated data in the target domain. The DAFormer facilitates the transition from preliminary algorithm development in the synthetic environment to the real-world implementations.

This research adopts DeepLab V2 segmentation model [37] to realize near real-time processing. This research compares two backbones, ResNet-101 and ResNet-50 [38], to identify the network architecture that can balance the accuracy and computational efficiency (Section 4.3). The recognition results are published as ROS messages.

3.5 Target generation and navigation module

The target generation and navigation module parses point cloud map, determines navigation goals for bridge component inspection, and controls the unmanned vehicle toward the navigation goals. This module works in real-time, interacting with other modules that compute the robot pose, point cloud map, and semantic segmentation results.

3.5.1 Point cloud parsing

This step segments and filters the point cloud map to extract the parts corresponding to the selected critical structural component class (columns in this research). First, semantic label predictions at the image key point locations are read and stored in a hash table containing the point ID p, the number of times those points are classified into the selected critical structural component class N^p_{cl}, and the number of times those points are observed N^p. When the new frame is processed, the counts N^p_{cl} and N^p for every point p in the current view are updated, and the label for the selected critical structural component class is assigned if $N^p_{cl} \geq t_{cl}$ and $N^p \geq t$, where the t and t_{cl} are thresholds ($t = 6$ and $t_{cl} = 10$ in this research). This process enables online incremental parsing of sparse point cloud maps.

After extracting parts of the point cloud map corresponding to the critical structural component class, the point cloud is further segmented into component instances using the Ordering Points to Identify Cluster (OPTICS) method [39]. This method can identify clusters of different densities without the need for specifying the number of clusters. This method is suitable for this research, because:

1. the point cloud map is continuously updated and expanded, changing the number of components in the map, and
2. the point cloud map created by the SLAM algorithm typically exhibits non-uniform point densities (e.g., point density is higher for nearby components, while regions between two components have very sparse point distributions).

Instead of the number of clusters, the OPTICS method uses the core distance (r_0, 0.05 m in this research) and reachability distance (r_1, 0.25 m in this research) to perform clustering. This point cloud parsing and clustering process is implemented as a callback function in ROS framework, which is invoked every time the message is published from the recognition module.

3.5.2 Navigation goal generation

The segmented point cloud for critical structural components is used to generate navigation goals and guide the robot toward the goals. This research determines camera viewpoints using a parametric map containing tight bounding boxes of component instances.

Parametric map

The first step of navigation goal generation is to obtain tight bounding boxes of the parsed point cloud for component instances. Bounding box detection begins with the estimation of the transformation between the global coordinate system and the coordinate system of each bounding box. For each point p in the cluster, the neighborhood $\mathcal{N}_p = \{p_i | i = 1,2,\ldots,n\}$ is obtained by either k-nearest neighbors (kNN) or spherical neighborhood within radius r as follows:

$$\mathcal{N}_p = \begin{cases} \{p_i | r_i < r\}, & N_c < N_s \\ \mathrm{kNN}(p), & N_c \geq N_s \end{cases} \tag{1}$$

where kNN(p) stands for the k-nearest-neighbors of point p, ($k = 15$). N_c is the number of points in the selected cluster, N_s is the threshold for using the spherical search method, which is set to 200. After obtaining the neighborhood of the point p, the centroid $\bar{p}$ and the covariance matrix Σ_p are calculated. The point normal vector is then calculated by the principal component analysis. After obtaining the normal vectors of the point cloud, a Euclidean distance-based clustering method is applied, and the average normal vector of the largest cluster is selected as the principal normal vector of the component. This vector is regarded as the normal of bounding box.

The normal vector is used to transform the cluster, and the axis-aligned bounding box (AABB) is fitted to the transformed cluster. The tight (oriented) bounding box representing the column can then be obtained by transforming the AABB back to the original coordinate system. Those tight bounding boxes constitute the parametric map; the map stores the centers of the bounding boxes, the projections of the centers of the 4 side faces of the bounding box onto the horizontal plane, and the normal vectors of the vertical faces.

The parametric map is updated continuously, following the updates in point cloud parsing results. If significant changes in the length, width (>10% change in size), or the normal vector (>15 degrees in rotation) are observed, the parametric map is updated to incorporate those changes. If the number of points in the new bounding box exceeds the threshold, that bounding box is added to the parametric map. When the bounding box does not fully enclose the component because of the incompleteness of the cluster, the lengths of the short and long sides are temporarily assumed to be the same, until the actual aspect ratio of the bounding box is greater than 2:1, at which point the shorter dimension is used. This process realizes the incremental creation and continuous updating of the parametric map of the environment.

Navigation goal generation

The navigation goal is generated based on the parametric map. During the initialization stage, the robot is commanded to move forward until it finds the first critical component. Afterwards, navigation goals are computed by adding an offset $\boldsymbol{n} \cdot x_d$ to the geometric center of the nearest surface of the current target component $\boldsymbol{x}_c$:

$$\boldsymbol{x}_t = \boldsymbol{x}_c + \boldsymbol{n} \cdot x_d \ . \tag{2}$$

After the initial navigation goal is generated, the robot starts the inspection mode. In this mode, the robot continuously checks if the bounding box is updated or not; if the bounding box is updated, the navigation goal is also updated. Otherwise, the navigation goal is fixed until the robot arrives at the goal.

When the robot arrives at the navigation goal, it checks whether the data for the current component has been fully collected. When the collection is complete, the robot queries the map for the unvisited bounding box with the highest number of points as the next target. If the collection is not complete, the next navigation goal is set to continue to inspect the current component.

3.5.3 Navigation

This research applies ROS navigation [40] to let the unmanned vehicle follow the navigation plans. The path planning in ROS navigation consists of a global planner and a local planner. This research uses the SLAM point cloud map as input to the ROS navigation, and treats the components in the environment as non-static obstacles. By this implementation, the robot can effectively avoid obstacles through local path planning as the vehicle approaches them.

4 Experiments

4.1 Overview of experiments

This research investigates the feasibility of the proposed approach in both synthetic and laboratory environments.

In the synthetic environment, TurtleBot3 Burger equipped with a simulated Kinect depth camera [8] is used as a hardware platform (Fig. 1 (a)). The resolution, frame rate, and field of view (FOV) settings for both RGB and depth images are set to 640 × 480, 30 fps, and 60 degrees horizontally, respectively. For the laboratory experiments, a Jetson TX2 onboard computer and an Intel RealSense D435i RGB-D camera [41] are mounted on the TurtleBot3 Waffle Pi UGV [8] (Fig. 1 (b)). The image resolution and frame rate are 640 × 480 and 15 fps, respectively. During the laboratory experiments, all the data processing is performed by the onboard computer, as opposed to the synthetic experiments where the richer computational resources from a desktop computer are leveraged.

In this research, RC railway viaducts are used as target structures for investigating autonomous bridge inspection tasks. The synthetic environment is set up by importing the model of an RC railway viaduct [9], while the laboratory environment was made by constructing a one-third-scale specimen of a high-speed railway viaduct (Fig. 2). The dimensions of the laboratory specimen are 5.00 m, 3.63 m, and 2.33 m in longitudinal, transversal, and vertical directions, respectively. The aluminum frame was first made as the core, and plastic foam components are attached around the core to realize the target shapes. Finally, wall papers resembling the textures of the concrete material are attached to the foam components to achieve the visual realism.

4.2 SLAM algorithm selection

This section discusses the selection of a visual SLAM algorithm for the proposed framework. This research compares the ORB-SLAM3 algorithm with RGB and RGB-D image inputs, as well as OV^{2}SLAM algorithm with RGB image input. The camera trajectory estimation results for a benchmark synthetic experiment are shown in Table 1 and Fig. 4. The corresponding mapping result is shown in Table 2 and Fig. 5. This research selects ORB-SLAM3 with RGB-D image input based on the high trajectory estimation accuracy (absolute trajectory error, ATE), superior map quality, and comparable relative pose error (RPE).

Table 1 Displacement ATE and RPE for different configurations

Configuration	Mean ATE (m)	RMSE ATE (m)	Mean RPE (deg)	RMSE RPE (deg)
ORB-SLAM3 RGB	0.026	0.039	0.150	0.261
ORB-SLAM3 RGB-D	0.023	0.026	0.163	0.271
OV^{2}SLAM	0.036	0.056	0.132	0.252

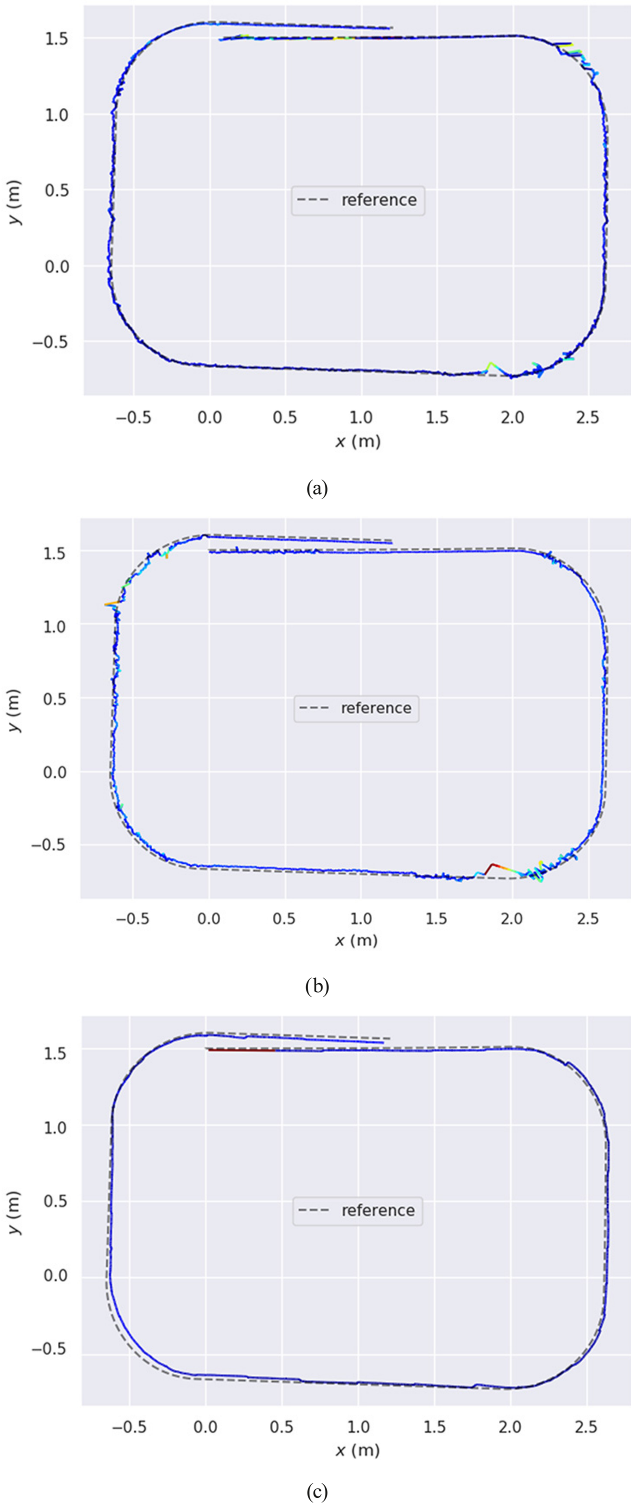
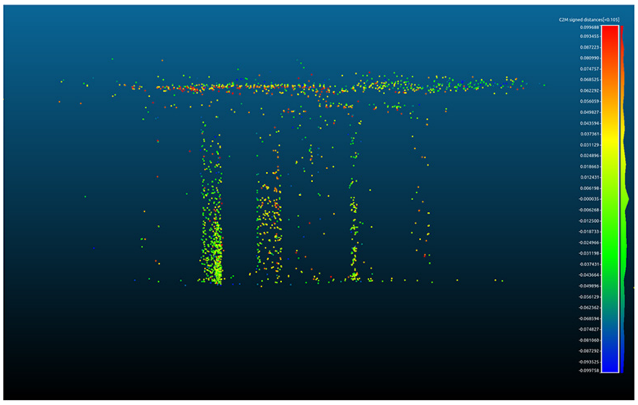


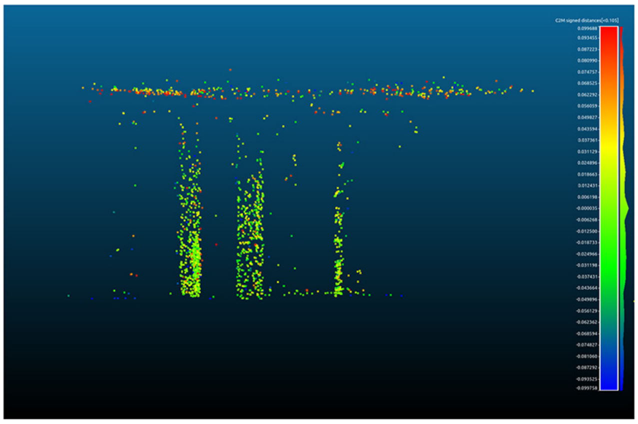
Fig. 4 Estimated trajectories with rotation error labeled: (a) ORB-SLAM3 RGB, (b) ORB-SLAM3 RGB-D, (c) OV^{2}SLAM

Table 2 Cloud/mesh distance result for different configuration

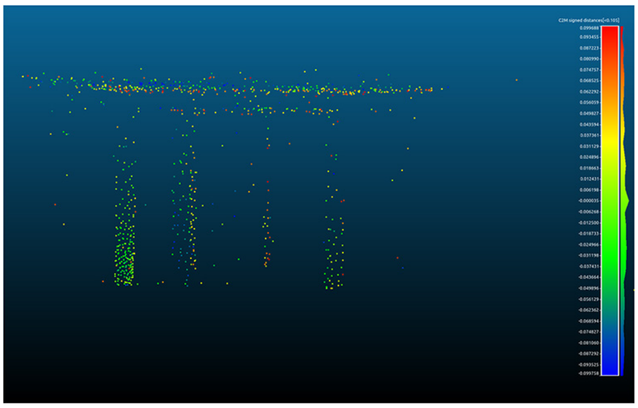
Configuration	Mean distance (m)	Standard deviation (m)	Valid points	Validation rate
ORB-SLAM3 RGB	0.035	0.044	2127	65.05%
ORB-SLAM3 RGB-D	0.027	0.037	2197	76.42%
OV²SLAM	0.043	0.050	950	55.17%



(a)



(b)



(c)

Fig. 5 Cloud/mesh distance in different configurations: (a) ORB-SLAM3 RGB, (b) ORB-SLAM3 RGB-D, (c) OV²SLAM

4.3 Semantic segmentation network aelection

This section discusses the selection of a semantic segmentation network architecture that balances accuracy and computational efficiency. Two backbone architectures, ResNet-101 and ResNet-50, are combined with the DeepLab V2 [37] segmentation head, and those networks are trained by the DAFormer framework [30]. During testing, all the computations are performed using the Jetson TX2 onboard computer. Table 3 shows the accuracy of bridge component segmentation with different backbones measured by pixel accuracy. Table 4 shows the mean processing time per frame. The results show that the inference time with ResNet-101 backbone is significantly more than that with ResNet-50 backbone. Although accuracy is slightly better for ResNet-101 backbone, this research selects ResNet-50, considering the balance between speed and inference accuracy evaluated using the Jetson TX2.

4.4 Synthetic experiment results

This section discusses the results of the autonomous UGV navigation for image data acquisition in the synthetic bridge inspection environment. The robot is first placed at a fixed position under the bridge. During the initialization, the robot moves forward until the first column is detected. Then, the robot initiates the inspection mode, collecting images of the column autonomously.

This research evaluates the autonomous navigation results using success rates. The success in this evaluation

Table 3 Pixel accuracy of DeepLab V2 with ResNet-101 and ResNet-50

Class	ResNet-101 Acc. (%)	ResNet-50 Acc. (%)
Non-bridge	82.49	81.50
Slab	76.32	74.78
Beam	96.22	92.33
Column	95.80	90.19
Non-structure components	64.85	83.27
Rail	0.14	0.11
Sleeper	0.00	0.00

Table 4 Processing speed of DeepLab V2 with different backbones on Jetson TX2

Type	Mean processing time (ms)	Mean time w/ SLAM (ms)
Deeplab V2 w/ ResNet-101	786.4	908.1
Deeplab V2 w/ ResNet-50	523.6	601.3

is defined using two tolerance levels (50 cm and 25 cm in the synthetic experiments). In each UGV navigation attempt and for each column of the bridge, the number of navigation goals visited successfully (defined by the two tolerance levels, Fig. 6) is counted. Because each column has four navigation goals corresponding to the four faces, this count takes the value of 0, 1, 2, 3, or 4 (0%, 25%, 50%, 75%, and 100%). The navigation attempt is repeated five times, and the average count and average success rate for each column is calculated. Finally, those column-wise success rates are sorted from most successful columns to least successful columns (1 column, 2 columns, …). The resulting success rates are presented in Table 5, and the example trajectories are showcased in Fig. 7.

4.5 Laboratory experiment results

This section discusses the autonomous navigation capabilities of the UGV for acquiring image data in a laboratory environment of an RC railway viaduct. The success rates defined for synthetic experiments are used in this section, with the low and high tolerances set to 12.5 cm and 25 cm, respectively, considering that the bridge model is

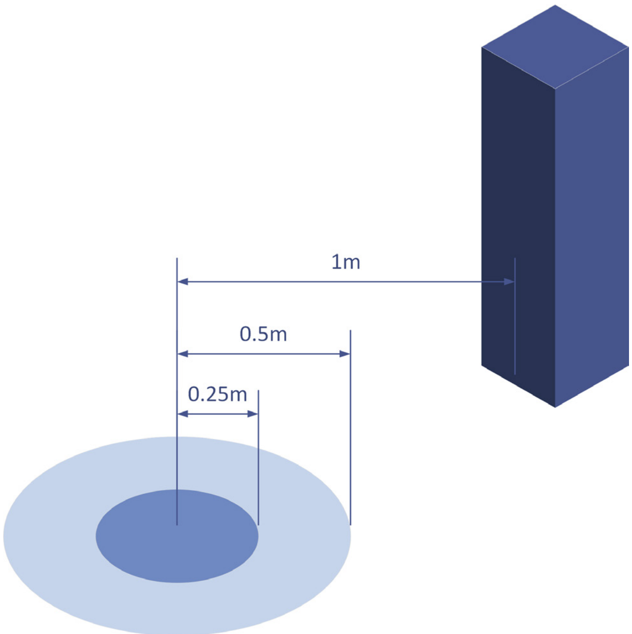


Fig. 6 Success region (Darker blue area indicates the low tolerance success region while lighter blue area indicates the high tolerance one)

one-third scale. The success rates are presented in Table 6. Example image collection results with the trajectory read from the odometry are shown in Fig. 8.

5 Discussion

The results from the synthetic and laboratory experiments demonstrate the significant potential of the proposed autonomous navigation framework. The output of the proposed framework is close-up images of critical structural components (columns in this research). Such images can be processed by image-based damage detection algorithms; for example, Prasanna et al. [42] showed a successful example of processing close-up images of structural components to identify structural damage. This research about autonomous navigation planning will provide an enhanced level of automation and accuracy for the structural condition assessment by providing ideal image data for assessment efficiently and autonomously. On the other hand, challenges that need to be explored have also been identified, which are discussed in this section.

5.1 Simulation analysis

In the synthetic environment, the UGV can accomplish the goal of autonomous navigation to acquire images at relatively high success rates. On the other hand, the following types of failures have been observed:

1. When the columns have relatively uniform surface textures, SLAM point cloud is sparse, ending up with the missing column detections (columns that are consistently not visited in Fig. 7).
2. False-positive detections of columns cause the robot to perform inspections not at the column locations. False positive detections are caused by the error of the semantic segmentation algorithm or the clustering process.
3. A gap between the success rates under the high and low tolerances exists, indicating the misalignment between the robot's observation and the target surface. The cause of this inaccuracy could be explained by the inaccuracy in the size or orientation of the estimated bonding boxes representing columns.

Table 5 Success rate in percentage of different counts of succeed navigated columns under simulation environment (The robot can navigate up to 6 columns)

Criteria	1 column success rate	2 columns success rate	3 columns success rate	4 columns success rate	5 columns success rate	6 columns success rate
Low tolerance	70.00%	65.00%	55.00%	30.00%	20.00%	5.00%
High tolerance	100.00%	100.00%	90.00%	65.00%	45.00%	20.00%

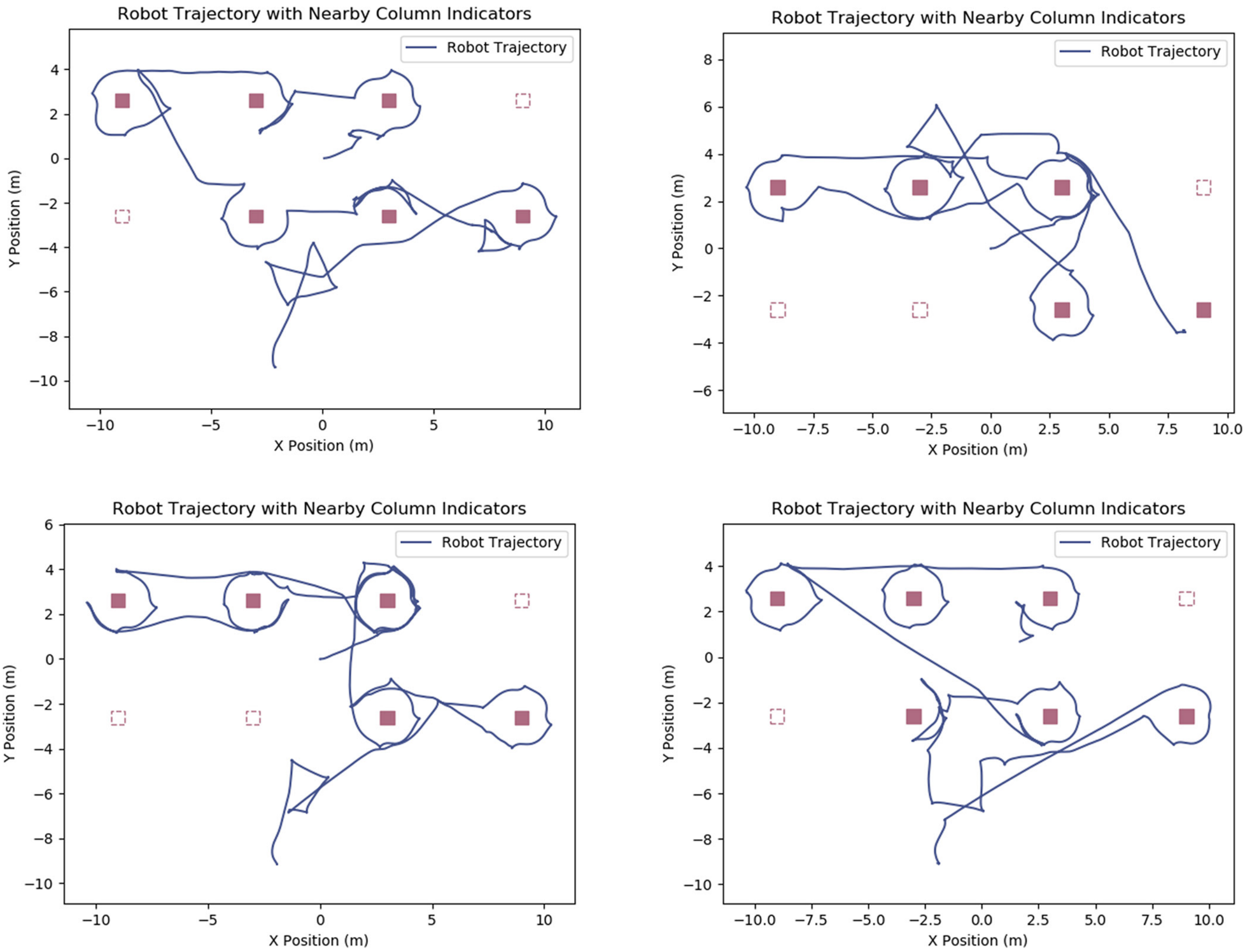


Fig. 7 Sample trajectories of autonomous navigation in simulation environment (Red solid square shows the successfully navigated columns, including partially and fully navigated; red dashed square shows the undetected columns)

Table 6 Success rate in percentage of different counts of succeed navigated columns under laboratory environment (The robot can only navigate up to 2 columns)

Criteria	1 column success rate (%)	2 columns success rate (%)
Low tolerance	25.00%	0.00%
High tolerance	64.29%	3.57%

4. Sometimes, when using Dynamic Window Approach (DWA) local planner in ROS navigation, robot oscillation and deadlock occurs when the navigation point is far from the current position.

5.2 Laboratory analysis

The performance in the laboratory environment is lower than that in the synthetic environment, which is explained in detail in the following:

1. The UGV platform is equipped with a Jetson TX2 computer, which poses constraints on the available computational resources. At a resolution of 640 × 480, the Jetson TX2 can only support the frame rate of 15 fps, and it requires a longer time for backend processing when adding new keypoints. This time delay can cause tracking loss, particularly during rotation.
2. Compared to the synthetic environment, the laboratory environment has significantly more complex background. As a result, most features appear in the background, while feature points on the bridge columns are relatively scarce. The lack of keypoints on the columns leads to the difficulty of detecting the column shapes during the navigation.
3. Image semantic segmentation is less accurate in the laboratory environment than in the synthetic environments, because of the existence of many patterns that confuses the algorithm. Such inaccuracy, combined with the rich features of the background, can cause the spurious detection of new "columns", which guide the robot to perform inspection tasks not at column locations.

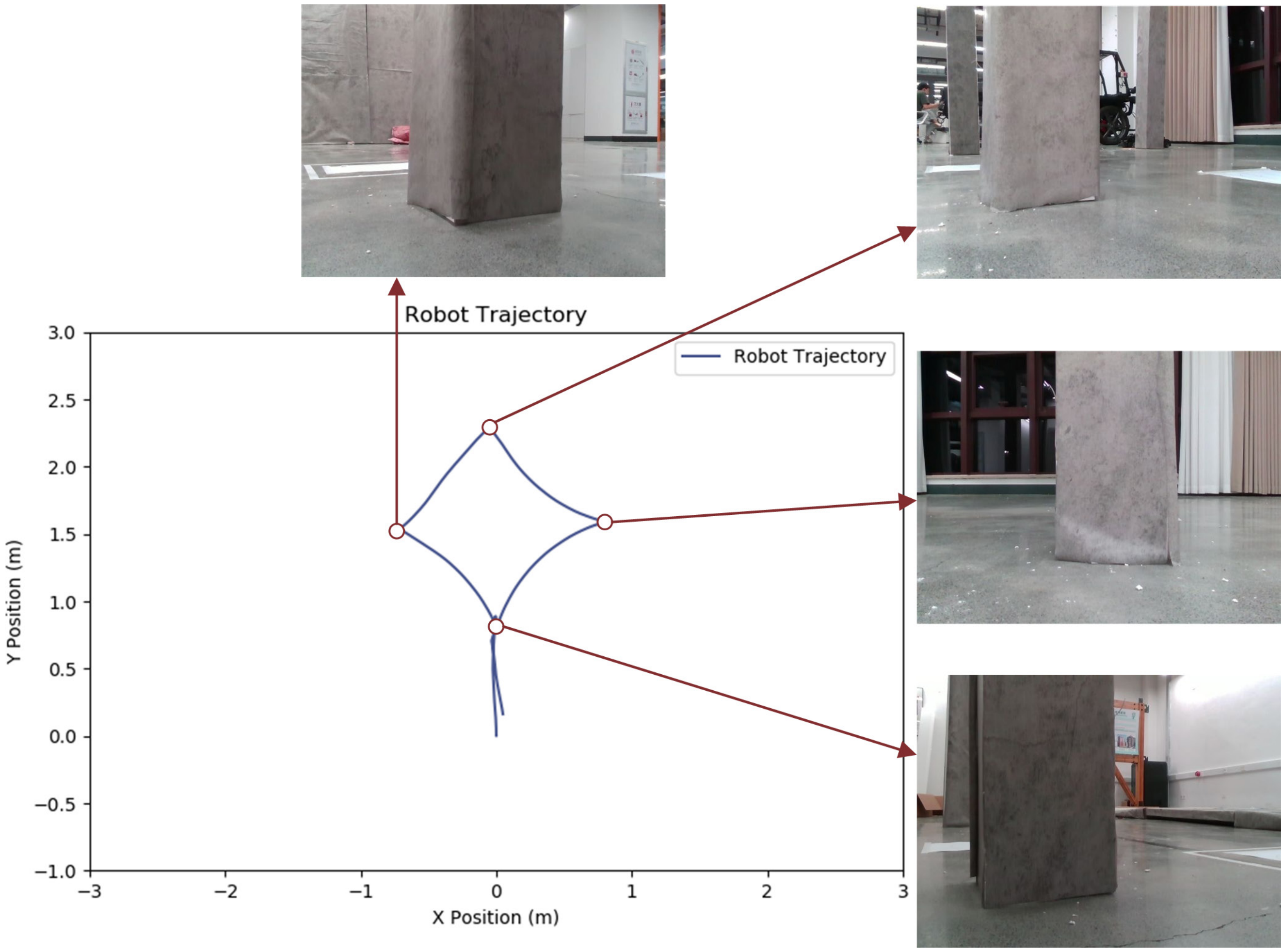


Fig. 8 Sample collected images during the navigation for a column in the laboratory

5.3 Future extensions

To address the challenges identified in this section, further investigations to improve the hardware, software, and their combinations are needed in the future. This includes:

1. better combinations of unmanned vehicles (e.g., mobile robots that can carry larger payloads),
2. better combinations of onboard computers, sensors (e.g., LiDAR sensor), and SLAM/visual recognition algorithms (e.g., direct point cloud processing),
3. improvement of semantic segmentation or clustering algorithms, taking into account the computational efficiency for the selected hardware, and
4. improvement of navigation module.

Optimized combinations of those technical components will be derived from through evaluations in synthetic and laboratory environments, further enhancing the potential and feasibility of the proposed autonomous bridge inspection approach.

6 Conclusions

This research developed an autonomous UGV navigation approach to acquire image data of critical bridge components. The proposed approach does not rely on pre-built bridge models or maps; instead, the UGV navigates based on the online and onboard processing of images, i.e., by recognizing components from the RGB images captured by the onboard camera, and planning and executing navigation based on the recognition results. Firstly, the visual SLAM module processes the RGB-D image streams to estimate the current robot position and the surrounding point cloud. Concurrently, the component recognition module performs semantic segmentation on the images. Subsequently, the target generation and navigation module extracts the points of interest corresponding to the structural components. The extracted points are clustered into component instances, based on which the parametric map is updated, and navigation targets are established. The proposed approach was validated in both synthetic and laboratory bridge inspection environments, demonstrating its significant potential for autonomous and effective bridge monitoring in practical applications.

There is a room for optimization in terms of performance and efficiency. Firstly, the study relies on accurate and efficient segmentation results; more efficient and accurate semantic segmentation models can be employed, and additional sensor information (such as depth and point cloud data) can be incorporated for more precise semantic segmentation. Secondly, multi-sensor fusion localization and mapping methods can be utilized; incorporating sensor information such as LiDAR and IMU can improve localization accuracy and map quality. Incorporating map-building techniques based on more effective geometric description can also improve the versatility and efficiency of the navigation system. Finally, the system can be extended to UAVs or quadrupedal robots for more viable autonomous navigation in actual scenarios of unstructured environment, and even for coordinated multi-robot navigation.

Acknowledgement

The authors acknowledge the financial support from the National Science Foundation of China (Grant No. 5236116565) and Shanghai Research Institute of Building Sciences Group Co., Ltd. (Grant No. KY10000038.20240011).

References

[1] Ministry of Transport of the People's Republic of China "2022年交通运输行业发展统计公报" (2022 Statistical Bulletin on the Development of the Transportation Industry), Ministry of Transport of the People's Republic of China, China, 000019713O04/2023-00049, 2022. [online] Available at: https://xxgk.mot.gov.cn/2020/jigou/zhghs/202306/t20230615_3847023.html [Accessed: 21 November 2024] (in Chinese)

[2] Sun, H., Li, H. "土木工程结构健康监测研究进展" (State-of-the-Art Review of the Structural Health Monitoring in Civil Engineering), Journal of Disaster Prevention and Mitigation Engineering, 23(3), pp. 92–98, 2003. [online] Available at: https://www.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFD&dbname=CJFD2003&filename=DZXK200303015&uniplatform=OVERSEA&v=ua2JvhRij9A514-LQ8mtbxrocNKhWdad_z34Maf01Io6EbnwsA-R69Q3Nc_6O46O [Accessed: 21 November 2024] (in Chinese)

[3] National Railway Administration of the People's Republic of China ""11·1" 天津南环临港铁路桥梁坍塌铁路交通较大事故调查情况公告" (Investigation Report of "11·1" Tianjin Nanhuan Lingang Railway Bridge Collapse Major Railway Traffic Accident), [online] Available at: http://www.nra.gov.cn/zzjg/jgj/bjdc/ggbj/202104/t20210409_337999.shtml [Accessed: 18 November 2024] (in Chinese)

[4] Zheng, W., Jianxin, G., Chengyu, H., Fengkang, Q. "Research on Pavement Crack Detection and Recognition Algorithm", Academic Journal of Computing & Information Science, 6(1), pp. 83–98, 2023. https://doi.org/10.25236/ajcis.2023.060113

[5] Hoskere, V., Narazaki, Y., Hoang, T. A., Spencer Jr., B. F. "MaDnet: Multi-task Semantic Segmentation of Multiple Types of Structural Materials and Damage in Images of Civil Infrastructure", Journal of Civil Structural Health Monitoring, 10(5), pp. 757–773, 2020. https://doi.org/10.1007/s13349-020-00409-0

[6] Narazaki, Y., Hoskere, V., Hoang, T. A., Fujino, Y., Sakurai, A., Spencer Jr., B. F. "Vision-based Automated Bridge Component Recognition with High-level Scene Consistency", Computer-Aided Civil and Infrastructure Engineering, 35(5), pp. 465–482, 2020. https://doi.org/10.1111/mice.12505

[7] Luo, K., Kong, X., Zhang, J., Hu, J., Li, J., Tang, H. "Computer Vision-based Bridge Inspection and Monitoring: A Review", Sensors, 23(18), 7863, 2023. https://doi.org/10.3390/s23187863

[8] ROBOTIS e-Manual "Turtlebot3", [online] Available at: https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/ [Accessed: 18 November 2024]

[9] Narazaki, Y., Hoskere, V., Yoshida, K., Spencer, B. F., Fujino, Y. "Synthetic Environments for Vision-Based Structural Condition Assessment of Japanese High-speed Railway Viaducts", Mechanical Systems and Signal Processing, 160, 107850, 2021. https://doi.org/10.1016/j.ymssp.2021.107850

[10] Hussein, A. B., Abdi, M. "Review Paper on Non-Destructive Testing and Their Accuracies to Measure the Mechanical Properties of Concrete", International Journal of Engineering Applied Sciences and Technology, 5(12), pp. 1–9, 2021. https://doi.org/10.33564/IJEAST.2021.v05i12.001

[11] Dais, D., Bal, İ. E., Smyrou, E., Sarhosis, V. "Automatic Crack Classification and Segmentation on Masonry Surfaces Using Convolutional Neural Networks and Transfer Learning", Automation in Construction, 125, 103606, 2021. https://doi.org/10.1016/j.autcon.2021.103606

[12] Gao, Y., Yang, J., Qian, H., Mosalam, K. M. "Multiattribute Multitask Transformer Framework for Vision-based Structural Health Monitoring", Computer-Aided Civil and Infrastructure Engineering, 38(17), pp. 2358–2377, 2023. https://doi.org/10.1111/mice.13067

[13] Gupta, N., Gupta, A. "Condition Assessment of the Structural Elements of a Reinforced Concrete Structure Using Non-destructive Techniques", IOP Conference Series: Materials Science and Engineering, 1116, 012164, 2021. https://doi.org/10.1088/1757-899X/1116/1/012164

[14] Spenser Jr., B. F., Hoskere, V., Narazaki, Y. "Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring", Engineering, 5(2), pp. 199–222, 2019. https://doi.org/10.1016/j.eng.2018.11.030

[15] Nishimura, Y., Mochiyama, H., Yamaguchi, T. "Propeller-Type Wall-Climbing Robot for Visual and Hammering Inspection of Concrete Surfaces", IEEE Access, 12, pp. 70963–70972, 2024. https://doi.org/10.1109/ACCESS.2024.3402218

[16] Kim, H., Narazaki, Y., Spencer Jr., B. F. "Automated Bridge Component Recognition Using Close-Range Images from Unmanned Aerial Vehicles", Engineering Structures, 274, 115184, 2023. https://doi.org/10.1016/j.engstruct.2022.115184

[17] Kim, H., Yoon, J., Sim, S.-H. "Automated Bridge Component Recognition from Point Clouds Using Deep Learning", Structural Control and Health Monitoring, 27(9), e2591, 2020. https://doi.org/10.1002/stc.2591

[18] Bianchi, E. L., Sakib, N., Woolsey, C., Hebdon, M. "Bridge Inspection Component Registration for Damage Evolution", Structural Health Monitoring, 22(1), pp. 472–495, 2023. https://doi.org/10.1177/14759217221083647

[19] Hajializadeh, D. "Deep Learning-based Indirect Bridge Damage Identification System", Structural Health Monitoring, 22(2), pp. 897–912, 2023. https://doi.org/10.1177/14759217221087147

[20] Morgenthal, G., Hallermann, N., Kersten, J., Taraben, J., Debus, P., Helmrich, M., Rodehorst, V. "Framework for Automated UAS-based Structural Condition Assessment of Bridges", Automation in Construction, 97, pp. 77–95, 2019. https://doi.org/10.1016/j.autcon.2018.10.006

[21] Lin, J. J., Ibrahim, A., Sarwade, S., Golparvar-Fard, M. "Bridge Inspection with Aerial Robots: Automating the Entire Pipeline of Visual Data Capture, 3D Mapping, Defect Detection, Analysis, and Reporting", Journal of Computing in Civil Engineering, 35(2), 04020064, 2021. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000954

[22] Bolourian, N., Hammad, A. "LiDAR-equipped UAV Path Planning Considering Potential Locations of Defects for Bridge Inspection", Automation in Construction, 117, 103250, 2020. https://doi.org/10.1016/j.autcon.2020.103250

[23] Prieto, S. A., Giakoumidis, N., de Soto, B. G. "AutoCIS: An Automated Construction Inspection System for Quality Inspection of Buildings", In: Proceedings of the 38th International Symposium on Automation and Robotics in Construction (ISARC), Dubai, United Arab Emirates, 2021, pp. 669–676. ISBN 978-952-69524-1-3 https://doi.org/10.22260/ISARC2021/0090

[24] Asadi, K., Suresh, A. K., Ender, A., Gotad, S., Maniyar, S., Anand, S., Noghabaei, M., Han, K., Lobaton, E., Wu, T. "An Integrated UGV-UAV System for Construction Site Data Collection", Automation in Construction, 112, 103068, 2020. https://doi.org/10.1016/j.autcon.2019.103068

[25] Car, M., Markovic, L., Ivanovic, A., Orsag, M., Bogdan, S. "Autonomous Wind-turbine Blade Inspection Using LiDAR-Equipped Unmanned Aerial Vehicle", IEEE Access, 8, pp. 131380–131387, 2020. https://doi.org/10.1109/ACCESS.2020.3009738

[26] Xu, C., Li, Q., Zhou, Q., Zhang, S., Yu, D., Ma, Y. "Power Line-Guided Automatic Electric Transmission Line Inspection System", IEEE Transactions on Instrumentation and Measurement, 71, 3512118, 2022. https://doi.org/10.1109/TIM.2022.3169555

[27] Peng, X., Su, G., Sengupta, R. "An autonomous Unmanned Aerial Vehicle exploration platform with a Hierarchical Control Method for Post-disaster Infrastructures", IET Cyber-Systems and Robotics, 6(1), e12107, 2024. https://doi.org/10.1049/csy2.12107

[28] Narazaki, Y., Hoskere, V., Chowdhary, G., Spencer Jr, B. F. "Vision-Based Navigation Planning for Autonomous Post-Earthquake Inspection of Reinforced Concrete Railway Viaducts Using Unmanned Aerial Vehicles", Automation in Construction, 137, 104214, 2022. https://doi.org/10.1016/j.autcon.2022.104214

[29] NVIDIA "NVIDIA Jetson TX2", [online] Available at: https://www.nvidia.cn/autonomous-machines/embedded-systems/jetson-tx2/ [Accessed: 18 November 2024]

[30] Hoyer, L., Dai, D., Van Gool, L. "DAFormer: Improving Network Architectures and Training Strategies for Domain-adaptive Semantic Segmentation", In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 9924–9935. ISBN 978-1-6654-6947-0 https://doi.org/10.1109/CVPR52688.2022.00969

[31] Narazaki, Y., Pang, W., Wang, G., Chai, W. "Unsupervised Domain Adaptation Approach for Vision-based Semantic Understanding of Bridge Inspection Scenes Without Manual Annotations", Journal of Bridge Engineering, 29(2), 04023118, 2024. https://doi.org/10.1061/JBENF2.BEENG-6490

[32] Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R. Ng, A. "ROS: An Open-source Robot Operating System", In: ICRA Workshop on Open Source Software, Kobe, Japan, 2009, 3(3.2), 5.

[33] Ferrera, M., Eudes, A., Moras, J., Sanfourche, M., Le Besnerais, G. "OV^{2}SLAM: A Fully Online and Versatile Visual SLAM for Real-time Applications", IEEE Robotics and Automation Letters, 6(2), pp. 1399–1406, 2021. https://doi.org/10.1109/LRA.2021.3058069

[34] Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M. M., Tardós, J. D. "ORB-SLAM3: An Accurate Open-source Library for Visual, Visual–inertial, and Multimap SlAM", IEEE Transactions on Robotics, 37(6), pp. 1874–1890, 2021. https://doi.org/10.1109/TRO.2021.3075644

[35] Rublee, E., Rabaud, V., Konolige, K., Bradski, G. "ORB: An Efficient Alternative to SIFT or SURF", In: 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564–2571. ISBN 978-1-4577-1101-5 https://doi.org/10.1109/ICCV.2011.6126544

[36] Qi, C. R., Yi, L., Su, H., Guibas, L. J. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space", In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 2017, pp. 5105–5114. ISBN 9781510860964 https://doi.org/10.5555/3295222.3295263

[37] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy K., Yuille, A. L. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs", IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), pp. 834–848, 2018. https://doi.org/10.1109/TPAMI.2017.2699184

[38] He, K., Zhang, X., Ren, S., Sun, J. "Deep Residual Learning for Image Recognition", In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778. ISBN 978-1-4673-8852-8
https://doi.org/10.1109/CVPR.2016.90

[39] Ankerst, M., Breunig, M. M., Kriegel, H.-P., Sander, J. "OPTICS: Ordering Points to Identify the Clustering Structure", ACM Sigmod Record, 28(2), pp. 49–60, 1999.
https://doi.org/10.1145/304181.304187

[40] Marder-Eppstein, E., Berger, E., Foote, T., Gerkey, B., Konolige, K. "The Office Marathon: Robust Navigation in an Indoor Office Environment", In: 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 2010, pp. 300–307. ISBN 978-1-4244-5038-1
https://doi.org/10.1109/ROBOT.2010.5509725

[41] Intel "Intel® RealSense™ Depth Camera D435i", [online] Available at: https://www.intelrealsense.com/depth-camera-d435i [Accessed: 18 November 2024]

[42] Prasanna, P., Dana, K. J., Gucunski, N., Basily, B. B., La, H. M., Lim, R. S., Parvardeh, H. "Automated crack detection on concrete bridges", IEEE Transactions on Automation Science and Engineering, 13(2), pp. 591–599, 2016.
https://doi.org/10.1109/TASE.2014.2354314