Received: 3 November 2024 | Accepted: 11 March 2025

DOI: 10.1111/mice.13469

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

# Complete-coverage path planning for surface inspection of cable-stayed bridge tower based on building information models and climbing robots

**Zhe Xia**[1] | **Jiangpeng Shu**[1,2] | **Wei Ding**[1] | **Yifan Gao**[1] | **Yuanfeng Duan**[1] | **Carl James Debono**[3] | **Vijay Prakash**[3] | **Dylan Seychell**[4] | **Ruben Paul Borg**[5]

[1]College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China

[2]Innovation Center of Yangtze River Delta, Zhejiang University, Jiaxing, China

[3]Department of Communications and Computer Engineering, University of Malta, Msida, Malta

[4]Department of Artificial Intelligence, University of Malta, Msida, Malta

[5]Faculty for the Built Environment, University of Malta, Msida, Malta

**Correspondence**
Jiangpeng Shu, Zhejiang University, Anzhong Building, 886 Yuhangtang Road, Hangzhou 310058, China.
Email: jpeshu866@outlook.com

**Funding information**
National Key R&D Program of China, Grant/Award Number: 2023YFE0115000; National Natural Science Foundation of China, Grant/Award Number: W2412092; Key R&D Program of Ningbo, Grant/Award Number: 2024H013; SINO-MALTA Fund 2023 Call (Science and Technology Cooperation)

## Abstract

Climbing robots present transformative potential for automated structural inspections, yet their deployment remains limited by the reliance on manual control due to the absence of effective environment perception and path-planning solutions. The critical bottleneck lies in the difficulty of generating accurate planning maps solely through onboard sensors due to the challenge of capturing open, large-scale, and irregular environments (e.g., cable-stayed bridge towers). This study proposes a building information modeling (BIM)-based complete-coverage path planning (BCCPP) framework, leveraging BIM to enable autonomous robotic inspection. The framework constructs accurate grid maps through BIM data, addressing the map-perception problem for robots in open, large-scale, and irregular environment while refining the boustrophedon-A* algorithm with multi-heuristic optimization, which reduces path repetition and improves energy efficiency. Field and simulated experiments on a cable-stayed bridge tower show the BCCPP achieves 93.5% coverage with 9.1% repetition, and planned paths were executable within a 0.2 m tolerance and collisions avoided. This work bridges BIM, climbing robot, and path planning, offering a scalable solution for intelligent infrastructure inspection.

# 1 | INTRODUCTION

## 1.1 | Background

Cable-stayed bridges are critical components of modern transportation infrastructure, with their towering structures serving as essential load-bearing elements (X. Zhou & Zhang, 2019). However, these bridge towers, often constructed from reinforced concrete, are susceptible to gradual structural deterioration due to environmental exposure, dynamic loads, and material aging (Alonso Medina & León González, 2022). In order to prevent the hazards of structural deterioration, periodic condition assessment inspection or long-term health monitoring are recognized as the effective solutions (Javadinasab Hormozabad et al., 2021) because it provides essential information in real time about the structure state, which is imperative to ensure structural integrity, safety, and cost-effective maintenance (Pan et al., 2023; Rossi & Bournas, 2023). Yet, structural health monitoring or inspection technique for cable-stayed bridge towers, which serve as an essential component of large-scale aquatic infrastructures, remains an underdeveloped research area as highlighted in recent studies (Pezeshki et al., 2023). Traditional manual inspections, conducted by personnel using elevated platforms or ropes, are not only labor-intensive and time-consuming but also pose significant safety risks, particularly given the height and scale of these structures (Cao, 2023).

To address the challenges associated with bridge-tower surface inspection, climbing robots have been increasingly promoted for automated inspection (Tian et al., 2022). As early as a decade ago, climbing robots have been able to move across vertical surfaces, collect high-resolution imagery, and preliminary detect defects such as cracks or corrosion (Leibbrandt et al., 2012). Although the technology of climbing robots matured a decade ago and continues to advance at the same time, research has remained at the stage of manual remote control due to the absence of an effective informational interactions with the target engineering projects (Liu et al., 2013). In previous research in the field of robotics, the interactive means between climbing robots and engineering projects information primarily relied on sensors (e.g., light detection and ranging [LiDAR], RGB-D cameras), with few considering the problem of sparse data collection by sensors in open large-scale environments (e.g., bridge tower; Li, 2022; Zhu et al., 2024).

Besides, as pointed out by Dogru and Marques (2022), the automated application of climbing robots requires complete-coverage path planning (CCPP) to ensure no critical areas are overlooked while minimizing energy consumption and operational time. At present, CCPP algorithms are primarily designed for rectangular or indoor environments, as they assume prior knowledge of the environment or depend on real-time sensor data in their researches (Chea et al., 2024; Muñoz et al., 2021; Ren & Jebelli, 2024). For example, Muñoz et al. (2021) pregenerated cluttered urban environments for testing drone coverage planning. Chea et al. (2024) and Ren and Jebelli (2024) used RGB-D camera and LiDAR, respectively, to acquire environmental information for instructing robotic collaboration and navigation in construction mission. However, neither method is applicable to bridge towers, due to the fact that open, large-scale, and irregular features of the bridge tower environment result in sparse environmental information acquired by the on-board sensors. This ultimately leads to the inability to generate accurate planning maps for robots.

As the level of information technology increases in the field of civil engineering, building information modeling (BIM) offers a transformative solution by providing detailed 3D geometric and semantic data of structures (L. Ding et al., 2020). Integrating BIM with robotic systems enables pre-planned, optimized paths derived from digital models, bypassing the limitations of sensor-based mapping (Xiao et al., 2025). Recent studies have demonstrated BIM's potential in robotic applications, such as assembly and U-based inspections, yet its application to climbing robots for bridge tower inspection remains unexplored (Y. Gao et al., 2023).

In summary, the key to solving the scientific problem is to establish an accurate mathematical relationship between real environment, map, and climbing robot in spatial dimension and to develop a complete coverage planning algorithm for task performance in the temporal dimension.

## 1.2 | Review of BIM and robotics

BIM serves as a cornerstone for managing complex geometries, structural health data, and maintenance workflows in the context of large-scale infrastructure maintenance (J. Chen et al., 2023). Recent advancements highlight its growing role in optimizing inspection and repair processes for aging infrastructure, particularly for durable structures like large-scale bridges, where manual methods are impractical (Hattori et al., 2024). While BIM's potential in building maintenance is well-documented, its integration with robotic systems for automated inspections is emerging as a transformative research field (Y. Gao et al., 2025).

Current research on the integration of BIM and robotics for mainly involves three aspects, which are localization, task planning, and path planning (Odugu et al., 2024). In localization, BIM serves as a critical data source for

providing spatial and semantic information, enabling precise positioning and environmental awareness for robotic systems. For example, Kim et al. (2021) utilized BIM for a robotic interior painting task simulation, augmenting the BIM data with additional construction details. The enhanced BIM data include information such as schedules, contract details, and other relevant construction details. In task planning, BIM facilitates the organization and execution of complex tasks by offering detailed geometric and operational insights into the environment. For example, Y. Gao et al. (2022) developed a robotic assembly framework for COVID-19 hospitalization facilities based on BIM. A collision-free task planning approach was used to assemble the components by utilizing their coordinates in BIM. In path planning, robots can obtain the required path-planning information from an existing building information model to construct planning maps. Follini et al. (2020) leveraged BIM to supply semantic knowledge, thereby improving the robot's navigation capabilities during maintenance tasks. Huang et al. (2023) proposed a systematic path-planning approach for unmanned aerial vehicles to inspect building exterior surfaces, supported by BIM. Z. Chen et al. (2024) designed an improved coverage path-planning system based on BIM for indoor robots, which includes mapping of multi-rectangular areas and integrated path planning of all areas. Their approach applies to rectangular planes but ignores the condition in irregular planes.

The integration of BIM and robotics has shown immense promise in revolutionizing building maintenance and inspection processes, particularly in terms of increasing the accuracy, efficiency, and automation of robot control in complex environments. However, despite these advancements, the application of BIM-integrated robotic systems in large-scale engineering inspections, such as bridge towers, remains scarce. This represents a significant gap in current research and practice, as the unique challenges posed by large scales, open conditions, and irregular geometries of such structures demand innovative solutions. Addressing this technological void could unlock new possibilities for automated, data-driven inspection and maintenance.

### 1.3 | Review of CCPP related to climbing robot and BIM

CCPP is a specialized technique in robotics for planning a continuous path that passes through all accessible points within a defined zone, which can be further categorized into two types: online and offline approaches (L. Gao et al., 2022). Online approaches assume that no prior or limited knowledge of the workspace hosted and require onboard sensors to construct maps in real time, which allows robots to react to the environment at any time (Muthugala et al., 2022). However, it is challenging to acquire bridge-tower surface information solely using sensors due to the fact that open, large-scale, and irregular features of the bridge tower environment. Therefore, the online approaches are not applicable in the case. In contrast, offline approaches require map fully modeling, which allows running various optimization algorithms for global planning to achieve optimal or near-optimal solutions (Dogru & Marques, 2022). This provides the offline approach with the advantages of high coverage rate, low repetition rate, small path length, and strong rationality of the path (Yi et al., 2023). To ensure complete coverage, CCPP developed a variety of coverage schemas. For example, one classical coverage schema is spanning tree coverage (STC; G. -Q. Gao & Xin, 2019), which covers an area by using a spanning tree to generate a spiral path. Another classical coverage pattern is wavefront (WF; Zidane & Ibrahim, 2018), which uses a distance from goal to start-based gradient to traverse the region. Comparing to STC and WF, boustrophedon motion distinguishes itself through its characteristic "S" shaped patterns (Choset, 2000). This motion strategy is fundamentally enhanced by the integration of the A* (A-star) algorithm, which serves as an essential pathfinding method that efficiently computes optimal paths between two points through cost function minimization. Specifically, the A* algorithm provides the computational foundation for determining optimal transitions between coverage swaths. Building upon this synergistic combination, the BA* (boustrophedon A*) algorithm was subsequently developed as an advanced hybrid approach that systematically integrates the A* framework with boustrophedon motion patterns (Viet et al., 2013). This integration enables more efficient and complete area coverage, explaining why the BA* algorithm has gained widespread adoption in coverage planning applications across various domains. With the development of learning methods in recent years, researchers have begun to consider using reinforcement learning (RL) in CCPP (Mannan et al., 2023). Comparing to other learning methods, RL distinguishes itself by establishing system mapping from the environment to the behavior, which means RL does not rely on supervision but learns from its own experience. For example, Kyaw et al. (2020) solved CCPP problems in large complex environments based on a new approach combining traveling salesman problem with the deep RL. Jonnarth et al. (2024) used an end-to-end RL approach based on a continuous state and action space to address online CCPP problems in unknown environments.

Although CCPP has been going on for a long time, existing studies always focused on refining coverage patterns through assuming known environment and ignored

methods to acquire an environmental map. However, acquiring planning maps is a prerequisite for path planning and intelligent detection. For open and large-scale environments such as bridge towers, there is bound to be the limitation of not being able to acquire planning maps from the surface of bridge towers.

In the context of climbing robots, the application of advanced planning algorithms remains relatively underexplored (Jang et al., 2020; S. T. Nguyen et al., 2020). For example, Jang et al. (2020) applied a ring-type climbing robot enabling automated detection of cracks in a high-rise bridge pier. S. T. Nguyen et al. (2020) designed an adaptable tank-like robot for visual and in-depth fatigue crack inspection of steel structures. But path planning is not involved in their work. While some studies have demonstrated the feasibility of integrating BIM with path planning to climbing robotic systems, these methods are only applicable to specific scenarios or missions. Ma et al. (2024) proposed a BIM-based indoor path planning method designed for wall-climbing robots, enabling them to achieve point-to-point traversal across different planes. However, this approach is constrained by its applicability to indoor environments with flat geometries and does not address the requirement for full coverage, which is often essential in inspection tasks. Gu et al. (2025) introduce a novel navigation framework specifically designed for biped climbing robots to operate within 3D curtain wall environments. A marker-based global localization approach is proposed to estimate the robot's pose, allowing for precise adjustments to the target foothold by comparing the desired and actual poses. However, their solution is limited in its applicability to regular glass curtain walls, which may restrict its use in more complex or irregular architectural structures.

Despite significant advances in CCPP and climbing robots, the integration of CCPP algorithms with BIM for large-scale engineering inspections, such as those required for bridge towers, remains a largely unexplored area. The unique challenges posed by irregular geometries, large-scale surface areas, and complex environmental conditions further complicate the development of effective solutions.

### 1.4 | Scope of the paper

To address the limitations mentioned in Sections 1.1–1.3, this study proposed a BIM-based CCPP (BCCPP) approach that can address the aforementioned technical challenges of robotic inspection for bridge towers. The proposed approach comprises the following key components:

1. A BIM-driven map generation algorithm that establishes a spatial mathematical relationship between the coordinates of the map-grid centroids and the irregular boundaries of the bridge-tower surface, considering the size and centroid of the climbing robot.
2. A heuristic-optimized path-generation algorithm that plans a coverage path based on a grid map for robotic inspection of bridge-tower surfaces.
3. A robotic task platform that controls the climbing robot to complete inspection tasks.

## 2 | SYSTEM OVERVIEW OF THE PROPOSED APPROACH

This study focuses on enabling autonomous damage inspection of cable-stayed bridge-tower surfaces using a climbing robot equipped with an RGB camera. While existing coverage path planning methods assume prior knowledge of the environment, they often fail to address the critical challenge of acquiring accurate planning maps for large-scale, irregular structures like bridge towers. To overcome this, we propose a BIM-robot operating system (ROS)-integrated framework that directly translates the geometric and spatial data from BIM into executable robotic inspection paths.

BIM design software (e.g., Rhino) manages informative databases for their projects, where application programming interfaces (API; e.g., Grasshopper) can applied to coupling with and structuring database so that end-users can acquire access to data and extract the required model elements (Jun Wu et al., 2023). The ROS framework provides a development basis in which developers can integrate numerous subroutines, called ROS nodes, into an application to interact with the robot actions functionally. This study used Rhino with Grasshopper API and ROS to develop a robotic prototype for inspecting cable-stayed bridge-tower surfaces.

As illustrated in Figure 1, the framework operates through three tightly coupled modules:

BIM-driven map generation (Section 4.1): Rhino–Grasshopper processes BIM data (e.g., tower surfaces, cable locations) to generate obstacle-aware grid maps, resolving irregular surface modeling challenges.

Heuristic-optimized path generation (Section 4.2): A refined BA* algorithm plans complete-coverage paths on the grid map, minimizing redundancy while ensuring collision avoidance.

Robotic task platform (Section 5): The planned coverage path is converted into ROS navigation instructions, enabling the robot to traverse the tower surface and collect inspection data.

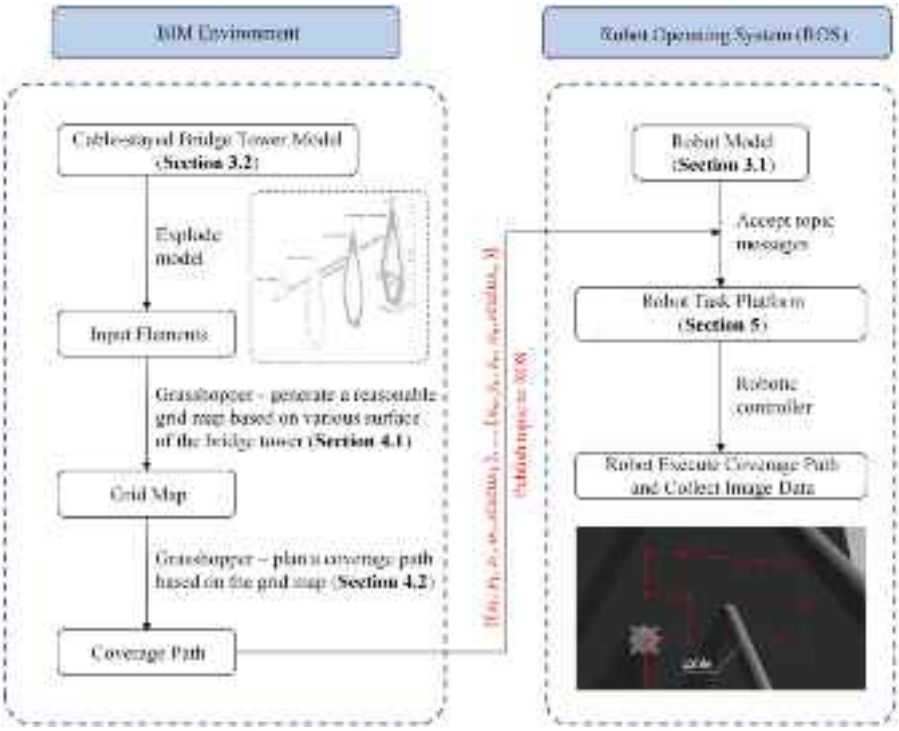


**FIGURE 1** Overview of the architecture of the proposed approach. BIM, building information modeling.

Critical inputs to this framework include:

- Robotic configuration (Section 3.1): Key robotic parameters such as robot size and camera scope directly impact algorithms and robot simulation.
- Inspected bridge tower (Section 3.2): BIM-derived geometric details (e.g., 180 m height, cable positions).

## 3 | ROBOTIC CONFIGURATION AND INSPECTED OBJECTS

To apply the proposed approach to an engineering project, the algorithm was developed considering the following aspects that directly or indirectly impact planning: robotic configuration (climbing robot) and inspected objects (cable-stayed bridge tower).

### 3.1 | Robotic configuration

To test the proposed approach, a climbing robot was selected and reconstructed in a simplified format using SolidWorks (3D modeling software for industrial robotics) as shown in Figure 2. The robot comprised three parts: a pan-tilt-zoom RGB camera, a robot base, and drive wheels. Each part was designed for a specific application. This section explains several factors related to the RGB camera and climbing robot directly or indirectly considered in path planning.

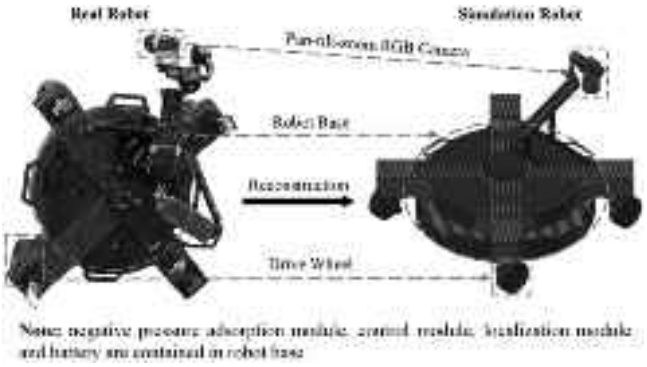


**FIGURE 2** Simulation of the climbing robot.

#### 3.1.1 | Climbing robot size

The robot size is determined by the size of the negative-pressure adsorption module contained in the base, a vortex suction unit with a maximum diameter of 360 mm (Zhao et al., 2018). In addition, considering the widths of the drive wheels on both sides, the overall maximum width of the robot was approximately 440 mm (Figure 3).

#### 3.1.2 | Camera scope

The horizontal width of the camera capture ($l_H$) should be considered during inspection to adjust it to the size of each inspected area ($G_s$). The left panel of Figure 3 illustrates a

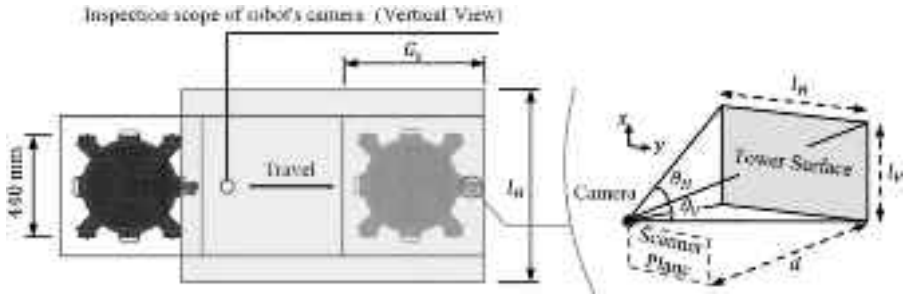


**FIGURE 3** Relationships between climbing robot inspection width ($l_H$), inspected area width ($G_s$), and climbing robot size. "$\theta_H$" represents the horizontal scanning angle, "$\theta_V$" represents the vertical scanning angle, and "$d$" represents the distance from the camera to the surface.

typical scenario and the right panel shows the calculation of $l_H$, which is defined in Equation (1).

### 3.1.3 | Functional modules

Several significant functional modules were integrated into the robot base: negative-pressure adsorption, control, and localization modules and a battery. The negative-pressure adsorption module provides sufficient suction force for the robot. The control module converts the instructions into robotic motion. The localization module considers the transmission of the local odometry and global position information. The battery guarantees sufficient endurance for the robotic inspection of the tower surface.

### 3.1.4 | Drive wheel

The robot adopts an all-wheel-drive driving mode, enabling it to gain the required movement and steering speeds by leveraging the suction force while climbing a facade (Q. Zhou & Li, 2018).

### 3.1.5 | Robot constraints

The climbing robot is designed to operate on a single plane and cannot autonomously transition from one plane to another, necessitating manual repositioning to accommodate different surfaces. The duration of a single operation is dependent on the battery capacity. Alternatively, a lightweight cable can be connected to the power climbing robot directly. And the climbing robot is programmed to return to a designated location when the battery is in a low state.

## 3.2 | Inspected bridge tower

To demonstrate the applicability of the proposed approach to typical bridge tower inspection problems, the Zhoushan Zhoudai Bridge, a cross-sea cable-stayed main passage bridge tower in Zhejiang, China, was selected as the study object (Figure 4b). As shown in Figure 4c, the cable-stayed bridge tower comprises the main body, pier, cables, and access platforms, with a 180 m height, 7 m width longitudinal to the bridge, and 47.2 m maximum width transverse to the bridge. The climbing robot inspects the front, back, and sides of the bridge tower.

Rhino allows users to construct 3D models with points, curves, and surfaces, enabling full utilization of the geometric information of the models (Aftab et al., 2023). Based on the engineering design drawing (Figure 4a) of the Zhoushan Zhoudai Bridge, a bridge-tower model was developed using the software at an equal scale. Figure 4c shows the modeled bridge-tower structure.

The model, derived from the engineering design drawings, adheres to the level of detail (LOD) specifications and is classified as LOD 300. At this level, the model incorporates precise geometric information, enabling accurate representation of the shapes, positions, and quantities of the components. It is important to note that the model primarily focuses on the shape, position, and count of the components and does not include detailed material properties, as such information is not required for path planning. Besides, the modeling is also based on the following prerequisites:

**First**, the bridge-tower surface is plane or continuous, with no concave or convex areas through which the climbing robot cannot travel.

**Second**, cables are the main structural obstacles on the bridge-tower surface.

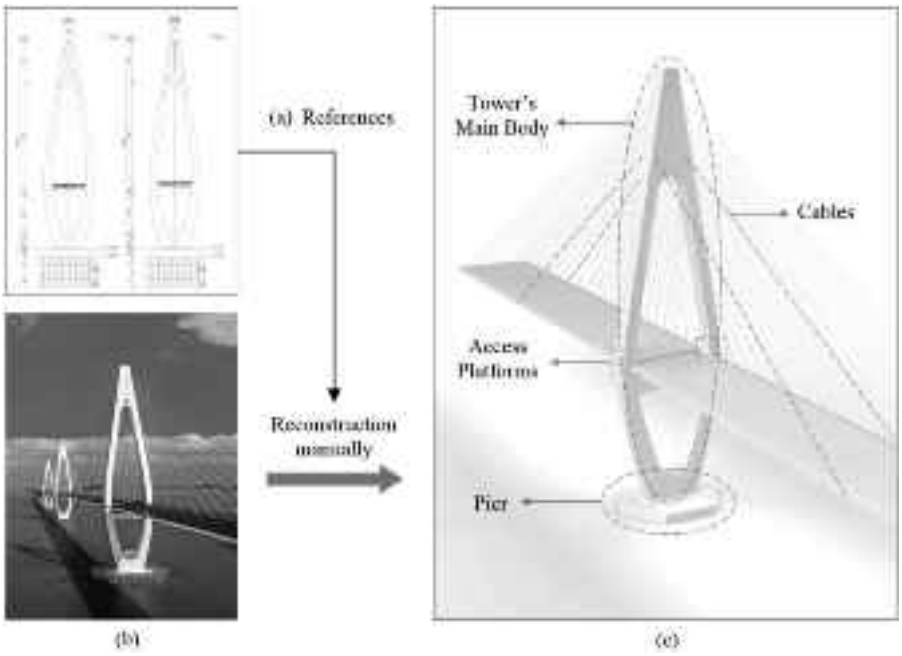


**FIGURE 4** Reconstructing the inspected bridge tower in the software: (a) engineering drawings of the tower, (b) inspected bridge tower in Zhejiang, China, and (c) composition of the modeled tower structure.

To effectively utilize the information from the model, the model was preprocessed in the software before being transmitted into the algorithmic framework as input elements, explained as follows:

**First**, Disassemble the bridge-tower model on several separate surfaces, defined as workspaces for climbing robots. (Input 0 in Figure 5.)
**Second**, Extract the surface edge to line and offset it inward by a certain distance $d_f$. (Input 1 in Figure 5.)
**Third**, Extract obstacle points on the surface, including hawser points. (Input 2 in Figure 5.)

## 4 | BCCPP ALGORITHM

This study proposes the BCCPP algorithm for inspecting cable-stayed bridge towers using a climbing robot. The climbing robot travels according to a generated path to inspect the surface and collect image data.

Figure 6 presents the composition and data streams of the developed BCCPP algorithm. The proposed algorithm comprises two primary layers: map generation (Section 4.1) and path generation (Section 4.2). The map-generation layer generates an adequately dense grid map (Output 0 in Figure 6) and provides the centroids of all grids (Output 1 in Figure 6), as well as the quantities of grids in the horizontal and vertical directions (Output 2 in Figure 6). Based on these data, the path-generation layer solves the problem of CCPP on a bridge-tower surface and outputs path data (Output 3 in Figure 6).

### 4.1 | BIM-driven map generation

As shown in Figure 6, the BCCPP algorithm structure can be subdivided into four blocks: source, map generation, path generation, and data output. This section focuses on the first two blocks: source and map generation.

The source block of the algorithm inputs the required model geometry information into Grasshopper as shown in Figure 5. In Input 0, the extracted surface of the bridge tower serves as the actual workspace for the climbing robot. This surface element is subsequently used to generate grids in constituting a collection of grids (grid $\in S_g$). In addition, the centroid of each grid is extracted and formed into a collection of centroids (centroid $\in P_c$). In Input 1, the extracted inwardly shifted surface edge is defined as the permitted workspace on the surface ($X_p$) for the center of the climbing robot. In Input 2, the extracted hawser points are considered as the centers of the obstacles. Subsequently, according to the cable size and the capturing range of the device, circular no-access zones around the cables ($X_{obs}$) are created to prevent collisions. Except for Input [0:2], Input 3 ($G_s$) is also imported as an original parameter to the map-generation block, used to directly determine the map-grid size. This parameter should be proportional to the capture range of the device and sufficiently small to achieve the targeted re-coverage ratio. The inspection

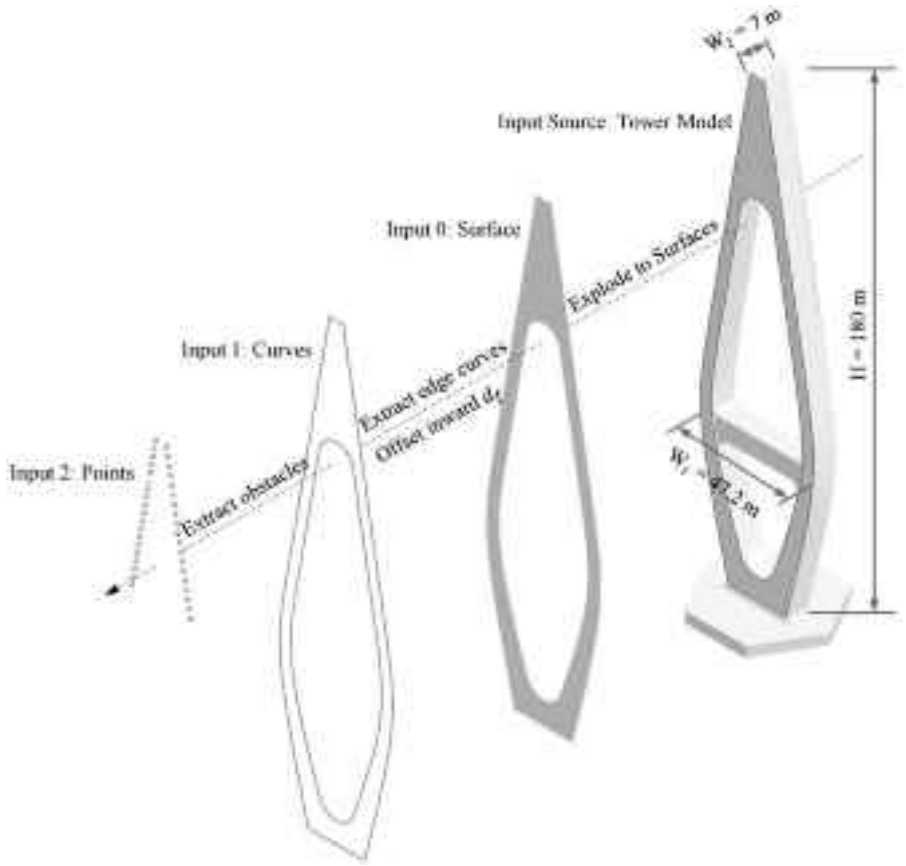


**FIGURE 5** Exploded view of the input cable-stayed bridge-tower model.

scale is primarily determined by the horizontal width ($l_H$) of the camera capture. As shown in Figure 3, the captured horizontal width ($l_H$) can be calculated using Equation (1).

$$l_H = 2dtan\,(\theta_H/2) \tag{1}$$

Considering the deviation during robotic travel and the risk of missed inspection, repeat inspection is introduced as an effective method. Presetting an appropriate re-coverage ratio based on the actual condition of an engineering project is crucial. The re-coverage ratio ($\eta$) shown in Figure 7 is expressed in Equation (2).

$$\eta = S_O/(S_N + S_O) \tag{2}$$

The map-grid size ($G_s$) depends on $l_H$ and $\eta$, which can be calculated using Equation (3).

$$G_s = l_H/(\eta + 1) \tag{3}$$

The climbing robot is assumed to be a rigid object when it travels over the tower surface. For an irregular plane, such as the front surface of the bridge tower, if robotic motion is considered as original motion in planning, the robot tends to travel beyond the actual workspace plane ($orgin \notin X_p$) when it travels over the edge of the surface or an obstacle. To solve this problem, a safe edge is introduced in the developed algorithm, obtained by offsetting the actual edge inwardly by a specific distance ($d_f$) and creating a circle of a specific radius ($r_{obs}$) centered on the obstacle points. Figure 8 shows an example of robotic traveling; depending on whether the centroid of a grid is inside the edge, grids are divided into accessible areas and inaccessible areas. The climbing robot travels in accessible areas to implement the inspection task. To ensure inspection of the inaccessible area, several parameter constraints should be declared in advance according to Equation (4).

$$\begin{cases} 1.\ \theta_V \geq \theta_H, \\ 2.\ \eta \geq 0.5 + d_f/l_H, \\ 3.\ \ d_f \geq\ d_{robot}/2, \\ 4.\ r_{obs} \geq (d_{cable} + d_{robot})/2 \end{cases} \tag{4}$$

where "$\theta_V$" and "$\theta_H$" represent the vertical and horizontal scanning angles, respectively; and $d_{robot}$ and $d_{cable}$ represent the maximum diameters of the climbing robot and cables, respectively.

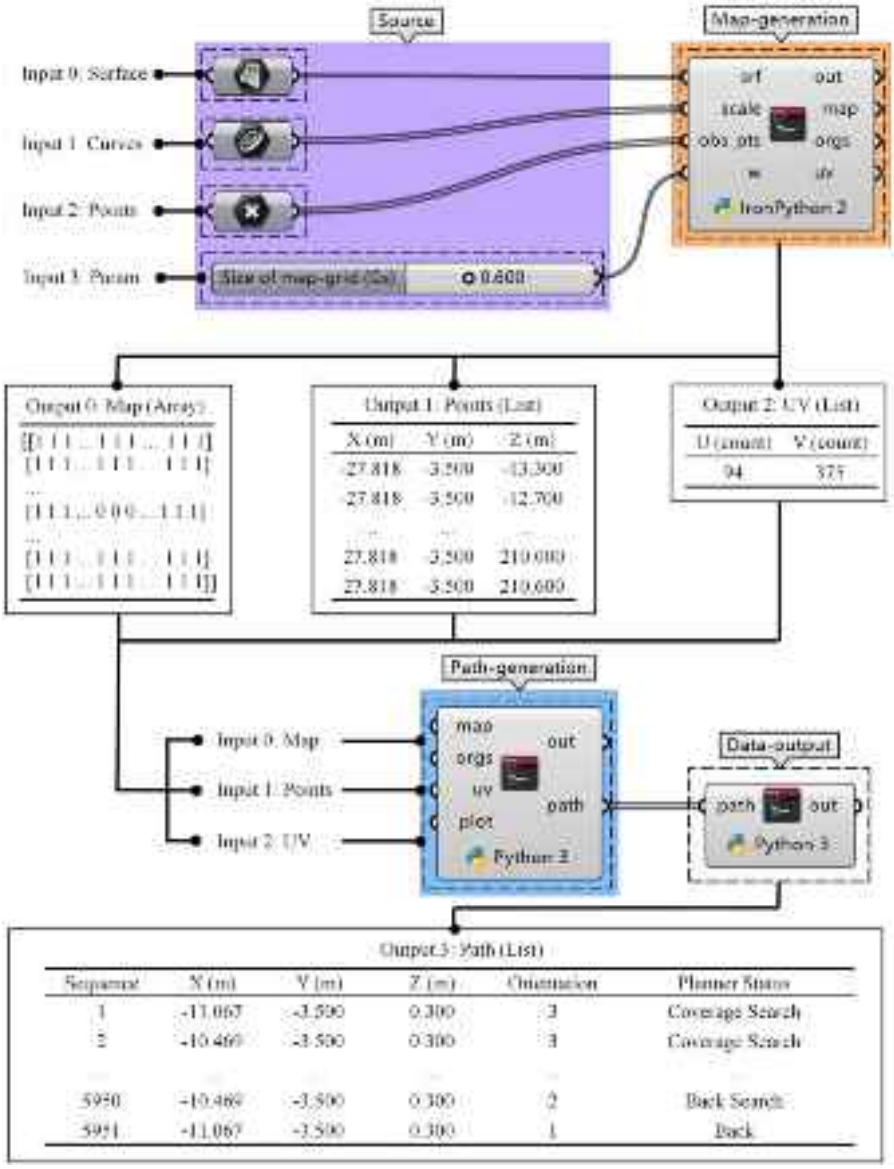


**FIGURE 6** Architecture of the building information modeling (BIM)-based complete coverage path-planning (BCCPP) algorithm.

The map-generation block utilizes BIM's high-fidelity geometric data from the source block to resolve the limitations of sensor-based mapping in irregular environments. Unlike sensor-derived point clouds, BIM provides exact spatial coordinates of the bridge tower's surfaces, edges, and obstacles (e.g., cables), enabling reasonable grid division even on non-rectangular planes. This geometric accuracy ensures that the generated grid map aligns with the actual structure, avoiding coverage gaps caused by sensor noise or incomplete scans. For example, the inward offset $d_f$ (Equation 4) is calculated using BIM-derived surface boundaries, guaranteeing that robot trajectory remains within safe margins while inspecting complex geometries.

The outputs of the map-generation block shown in Figure 6 function as follows: Output 0 (Map) exports the grid map as an array, where "0" and "1" indicate accessible and inaccessible grids, respectively; Output 1 (Points) exports centroid coordinates of various grids ($(x_n, y_n, z_n) \in P_c$); and Output 2 ($UV$) exports grid quantities in the horizontal and vertical directions ($[u, v]$).

A series of Grasshopper-integrated methods are called in the map-generation block, labeled by gh. The details of the block are presented in Algorithm 1 and explained below.

**First**, the function gh.Untrim($s$) restores the irregular plane to a rectangle. The number of grids $UV$ is rounded to the calculated value by dividing the function gh.Dimensions($s$) by the map-grid size $G_s$.

**Second**, according to $UV$, the surface is divided into a collection of grids $S_g$ using the function gh.Isotrim($UV$). In addition, a shape-corresponding initialized array $G_m$ is created based on $UV$.

**ALGORITHM 1** Map generation

**Input**: Surface $s$; Surface boundaries $L$; Obstacle Coordinates $P_{obs}$; Grid Size $G_s$
**Output**: Grid Map $G_m$; Grid Centroid Coordinates $P_c$; Map Dimension $UV$
Rectangle $rec \leftarrow$ gh. Untrim($s$)
$UV \leftarrow$ gh. Dimensions($s$)/$G_s$
Grids $S_g \leftarrow$ gh. Isotrim($s$, $UV$)
$G_m \leftarrow$ array[$UV.u$][$UV.v$] $\leftarrow 0$
$P_c \leftarrow$ gh. DeconstuctRectangle($S_g$). centroid
Circles $X_{obs} \leftarrow$ gh. Circle($P_{obs}, r_{obs}$)
**for** $i \leftarrow 0$ to len($S_g$) **do**
  **if not** gh. PointInCurves($P_c[i], L.outline$) **or** gh. PointInCurves($P_c[i], L.inline$) **then**
    $G_m[i \bmod v][i/v] \leftarrow 1$
  **end**
  **if** $X_{obs}$ **then**
    **if** gh. PointinCurves($P_c[i], X_{obs}$) **then**
      $G_m[i \bmod v][i/v] \leftarrow 1$
    **end**
  **end**
**end**

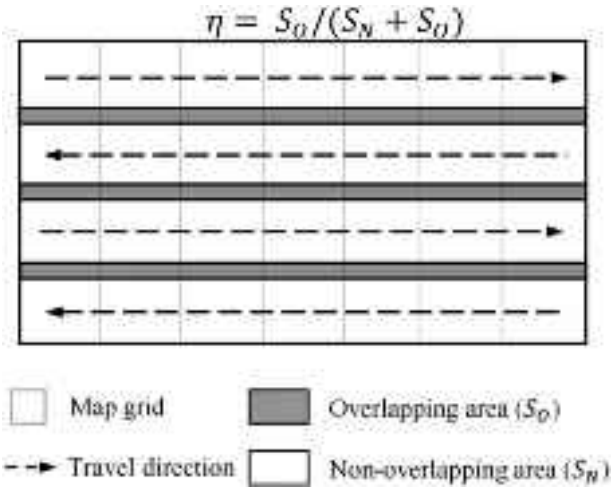


**FIGURE 7** Architecture of the BCCPP algorithm.

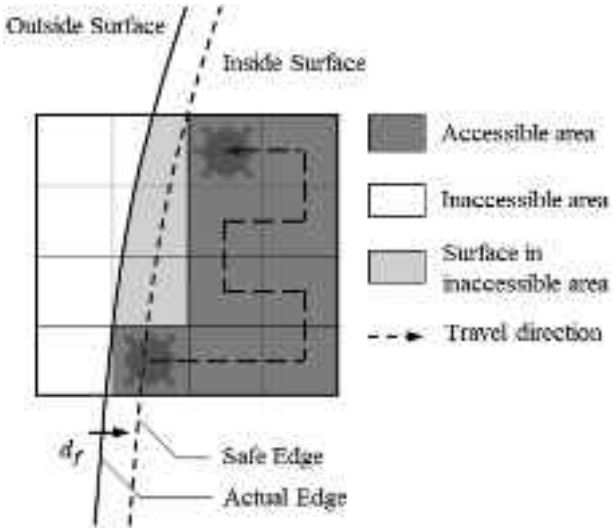


**FIGURE 8** Considering surface in grid outside.

**Third**, grid centroids are extracted to $P_c$ using the function gh.DeconstuctRectangle($S_g$).

**Fourth**, obstacle areas $X_{obs}$ are generated by the function gh.Circle($P_{obs}$, $r_{obs}$), where $r_{obs}$ can be considered by (4) in Equation (4).

**Fifth**, the combination of the functions gh.PointInCurves ($P_c[i]$, $L$) judges whether the grid centroid is outside the surface. If True, "1" is assigned to the corresponding position of this centroid in the map array.

**Sixth**, if there are obstacle inputs, the function gh.PointinCurves($P_c$[i], $X_{obs}$) determines whether the grid centroid is inside the potential collision areas. If True, "1" is assigned to the corresponding position of this centroid in the map array.

**Finally**, the end loop returns the array $G_m$, collection of centroids $P_c$, and number of grids $UV$.

**Overall**, the map-generation block first processes the original surface into a collection of grids $S_g$, establishes obstacle areas $X_{obs}$, and then checks whether the centroid of grid $P_c$[i] is within the permitted collision-free space $X_{free}$ by iterating over $S_g$, which determines the final map array. Figure 9 shows the results of applying map generation to the front surface of the tower to generate a grid map. The figure shows that the front surface of the

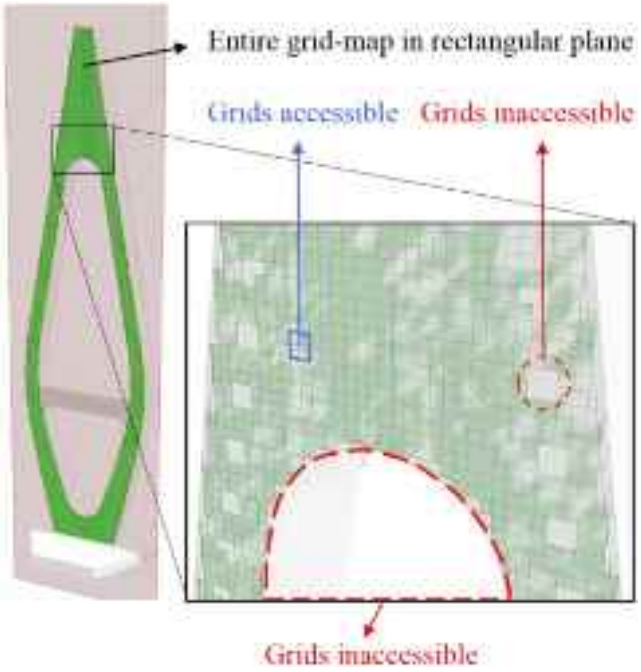


FIGURE 9 Applying map generation to the front surface of the tower.

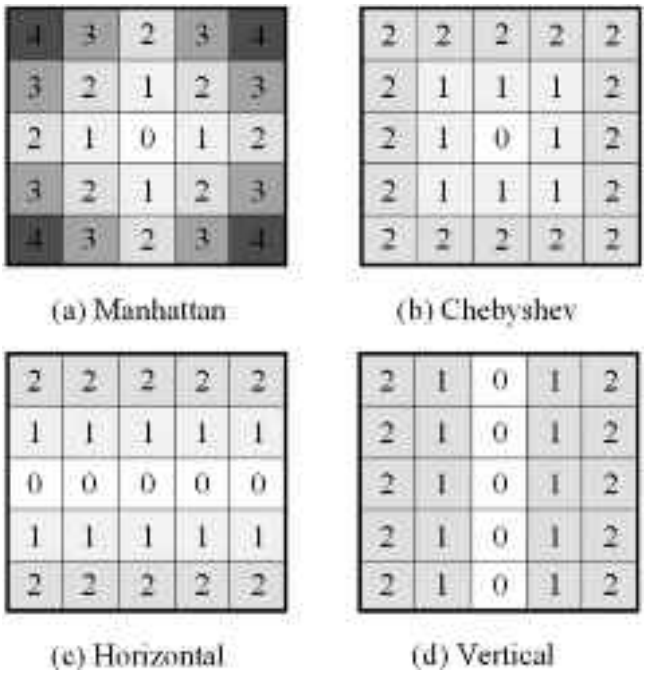


FIGURE 10 Applied heuristics in refined boustrophedon-A* (BA*), number in blocks represents the cost.

bridge tower is divided into a grid map in a specific size based on a rectangular plane. The grids are categorized into the following categories: grids accessible (symbol "0") and grids inaccessible (symbol "1"), used for the subsequent path-generation block.

## 4.2 | Heuristic-optimized path generation

This section focuses on the last two blocks: path generation and data output. The path-generation block of the algorithm involves planning a coverage path based on bridge surfaces and exporting the path data to the data-output block. Output 3 (Path) of the data-output block shown in Figure 6 exports the path data as a list, comprising the sequence, position ($[x, y, z, o]$), and planner status, to describe a robotic waypoint.

Considering the requirement of the climbing robot to find a coverage path to inspect the entire surface with minimal effort based on the outputs from the map-generation block, we refined the classical coverage planning algorithm (BA*) by introducing multiple heuristics for meritocracy in the path-generation block. Heuristics play a key role in decision-making and cost calculation during path planning. The refined BA* algorithm integrates four domain-specific heuristics (Figure 10) tailored to climbing robot dynamics and bridge-tower geometries:

- **Manhattan**, prioritizing grid-aligned motion (horizontal or vertical) to minimize traversal steps in structured environments.
- **Chebyshev**, optimizing diagonal movements across grids, reducing path length in obstacle-sparse regions.
- **Horizontal**, favoring lateral exploration to maximize horizontal coverage efficiency, critical for wide bridge surfaces.
- **Vertical**, penalizing frequent vertical adhesion changes to conserve energy on tall structures.

Traditional single-heuristic approaches (e.g., Manhattan-only) often fail to address the competing demands of coverage completeness, energy efficiency, and obstacle avoidance. For example, while Manhattan minimizes step count, it may increase adhesion transitions on vertical surfaces, raising energy costs. Conversely, Vertical reduces motor strain but risks incomplete coverage. By dynamically selecting these heuristics, such as prioritizing Horizontal on expansive planes and Manhattan near obstacles, the algorithm achieves context-aware optimization.

The finite-state machine (FSM) governs transitions between planning modes to ensure optimal path generation under varying conditions. By modeling planner status as a combination of three planning modes (Coverage Search, A*Unvisited Search, and A*Back Search) and four evaluation modes (Complete, Not Complete, Back,

and Not Back), the FSM enforces context-aware switching logic. For example, when the Coverage Search mode encounters an obstacle or a search dead region, the FSM transitions to A*Unvisited Search to prioritize unexplored areas. Conversely, if battery levels fall below a threshold during exploration or coverage mission is completed, the A*Back Search mode is activated to plan a return path while minimizing adhesion changes. State transitions are optimized using real-time heuristic evaluations, ensuring that the planner avoids local minima and maintains global coverage progress. This structured approach guarantees that the robot dynamically balances exploration and resource constraints without manual intervention.

The FSM, a tool for modeling object behavior in response to external events (H. D. Nguyen et al., 2023), is employed here to manage the planner's status. Details of the path-generation block are presented in Algorithm 2 and explained as follows:

**First**, the planner initializes the start $p_i$ using the function Init() and plans from the start using the function Coverage_search() in the planner status of Coverage Search.

**Second**, if the planner is trapped, it switches the planner status to an A*unvisited search to find an unvisited grid using the function A*_univisited_search().

**Third**, if the planner achieves the planning task, it switches the planner status to A*Back Search to find a path back to the start using the function A*_back_search().

**Finally**, the function Match_path() links the planning results to the actual coordinates of the waypoints and returns the data in Coverage Path *CP*.

Different heuristics and initial orientations significantly influence the cost of the planned paths. Thus, the algorithm selects cost-optimal solutions for the four heuristics for the four initial orientations. Finally, the data-output block of the algorithm exports the path data externally and publishes it to the ROS as a topic intended to instruct robotic motion.

## 5 | ROBOTIC TASK PLATFORM

This study developed a robotic task platform to comprehensively implement the BCCPP algorithm from the BIM platform to the ROS platform and test its effectiveness in an engineering perspective. This section describes the architecture of the robotic task platform and its subcomponents.

The robotic task platform was developed based on the ROS framework. Four new nodes were developed and integrated with the existing ROS nodes in the system architecture (Figure 11). These nodes and their interactions within the simulation are detailed as substeps in the following three steps: the nodes labeled in the description are all shown in Figure 11.

Step 1/3 creates a simulation environment comprising a cable-stayed bridge tower and climbing robot attached to the facade (Figure 12). Gazebo is used in the simulation, which handles the physics of the environment and provides a graphical representation of the world (Farley et al., 2022). First, a Gazebo (N1) simulation is initiated in ROS and loads the simulation environment. The world file includes building information from the exported SDF files. In this method, any object model, except the robot, can be inserted as an SDF file into the simulation environment. In addition, a Gazebo-based physical plugin was also developed and inserted into the world file to adjust the physical relationship between the climbing robot and the upright surfaces of the bridge tower to which the robot was attached. The robot file was exported in URDF format through Gazebo standard spawning processing. A robot spawned in a simulation environment is linked to various nodes for specific applications, such as motion planning, location, and visualization.

After the Gazebo's physical environment is started, Step 2/3 inputs the path waypoints information and generates detailed robot actions by communicating with various nodes. Coverage_taskpoint_manager (N8) is a key node that provides task-point information of the coverage path through which the robot is required to pass and coordinates the task process based on robotic execution. The node reads the CSV file, P-Path, which contains the planned coverage path information comprising the sequence, position, and planner status (Output 3 in Figure 6). Subsequently, the information is published point-by-point to node move_base (N4) according to the real-time position feedback from the simulation environment and localization node. Node move_base (N4) aggregates messages from other nodes and instructs the motion of the climbing robot. Nodes robot_state_publisher (N6) and joint_state_publisher (N5) provide feedback on the real-time poses of the robotic links to node move_base (N4). The adaptive Monte Carlo localization (amcl) node (N2) addresses localization of the robot. Node map_server (N3) provides a boundless blank map for navigation. Node odom_transformer (N9) converts the coordinate system from the ground plane to the bridge-tower surface for the amcl and move_base nodes.

When the robotic task platform is online, Step 3/3 records the executed path of the climbing robot, collecting image data from the inspected surface and visualizing the task points and robotic execution path. The detailed executed path of the robot are generated in simulation environment. Compared to Gazebo, Rviz specializes in more abundant visualization of robot motions (Toan

**ALGORITHM 2** Path generation

**Input**: Grid Map $G_m$; Grid Centroid Coordinates $P_c$; Map Dimension $UV$
**Output**: Coverage Path $CP$
Heuristics $Heur \leftarrow [V, H, C, M]$; Orientations $O \leftarrow [0, 1, 2, 3]$
Paths $PS \leftarrow \emptyset$; Planner Status $S \leftarrow$ Coverage Search
Back Judge $BJ \leftarrow$ True; Search Judge $SJ \leftarrow$ False
**for** $h$ in $Heur$ **do**
  **for** $o$ in $O$ **do**
    Path $P \leftarrow \emptyset$; Inital Position $p_i \leftarrow$ Init$(h, o, G_m)$
    **while** $BJ$ **do**
      $SJ \leftarrow$ False
      **if** $S ==$ Coverage Search **then**
        $result \leftarrow$ Coverage_search$(h, p_i, G_m)$; $P$.extend$(result$.path$)$
        **if** $result$.status **then**
          $S \leftarrow$ Complete
        **else**
          $S \leftarrow$ A* Unvisited Search; $SJ \leftarrow$ True
        **end**
      **elif** $S ==$ A* Unvisited Search **then**
        $result \leftarrow$ A*_unvisited_search$(h, G_m)$; $P$.extend$(result$.path$)$
        **if** $result$.status **then**
          $S \leftarrow$ Coverage Search; $SJ \leftarrow$ True
        **else**
          $S \leftarrow$ Not Complete
        **end**
      **end**
      **if** $SJ ==$ False **then**
        $S \leftarrow$ A* Back Search
        $result \leftarrow$ A*_back_search$(h, G_m)$; $P$.extend$(result$.path$)$
        **if** $result$.status **then**
          $S \leftarrow$ Back; $BJ \leftarrow$ False
        **else**
          $S \leftarrow$ Not Back
        **end**
      **end**
    **end**
    $PS$.append$(P)$
  **end**
**end**
$PS.sortby(P$.step, $P$.cost$)$
Coverage Path $CP \leftarrow$ Match_path$(PS[0], P_c, UV)$

et al., 2023). All task points are marked in Rviz, allowing the developer to intuitively observe the progress of task-point execution and the relationship between the robotic real-time execution path and task points. The path_recorder (N10) node generates an overall record of the executed robotic path in a CSV file, E-Path, comprising the time-stamped global coordinates of the robot. The imagecollector (N11) node collects the image data of the inspected surface at a specific frequency.

## 6 | RESULTS

Experiments were performed to evaluate the developed prototype in terms of reasonableness of the planning logic,

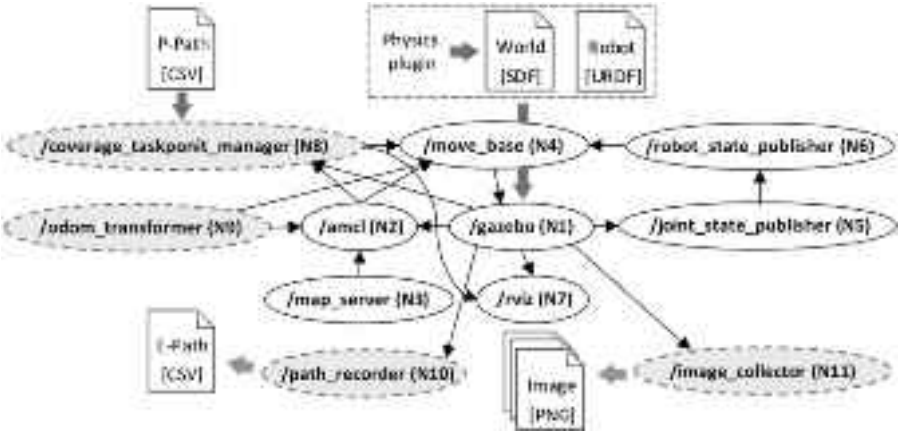


**FIGURE 11** Simplified node graph of the robot operating system environment. (Nodes developed for this research are shown with a grey dashed ellipse.)

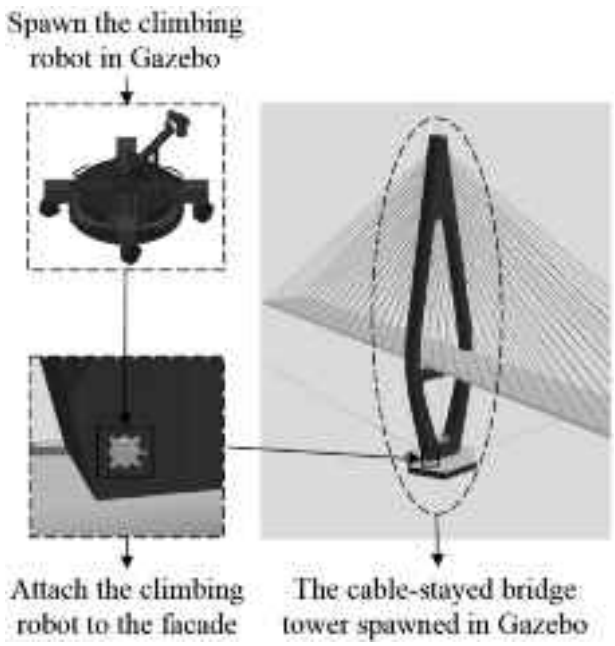


**FIGURE 12** Creating the simulation environment in Gazebo to test the proposed approach.

coverage, repetition and turning ratio, and executability of the planned path both in simulation and field.

## 6.1 | Reasonableness of planning logic

The cross-sectional shapes of cable-stayed bridge towers can broadly be categorized into H-type, A-type, and Diamond-type. To evaluate the generalizability of the proposed algorithm, totally three cases of bridge tower are introduced into the study. As shown in Figure 13I, except for the focused study target in Case 1, which represents Diamond-type, Cases 2 and 3 represent H-type and A-type, respectively.

**TABLE 1** Waypoint sequence, positions, and planner status planned on the side surface of the tower.

| Seq. | Positions | Status |
|---|---|---|
| 1 | (11.497, −3.208, 0.292, 3) | Coverage |
| 2 | (11.497, −2.625, 0.292, 3) | Coverage |
| 3 | (11.497, −2.041, 0.292, 3) | Coverage |
| 4 | (11.497, −1.458, 0.292, 3) | Coverage |
| 5 | (11.497, −0.875, 0.292, 3) | Coverage |
| … | … | … |
| 3659 | (2.992, 2.625, 179.354, 3) | Coverage |
| 3660 | (2.992, 3.209, 179.354, 3) | Complete |
| 3661 | (3.094, 3.209, 178.764, 2) | A*back |
| 3662 | (3.196, 3.209, 178.175, 2) | A*back |
| 3663 | (3.297, 3.209, 177.585, 2) | A*back |
| … | … | … |
| 3971 | (11.788, −2.041, 1.453, 1) | A*back |
| 3972 | (11.643, −2.041, 0.873, 2) | A*back |
| 3973 | (11.497, −2.041, 0.292, 2) | A*back |
| 3974 | (11.497, −2.625, 0.292, 1) | A*back |
| 3975 | (11.497, −3.208, 0.292, 1) | Back |

These cable-stayed bridge towers were constructed in BIM software to test the BCCPP algorithm. Take Case 1 as an example, the tower is symmetrical from front to back and side to side; thus, the front and side surfaces were used for the test results. Tables 1 and 2 present the path-planning results for the side and front surfaces of the tower in Case 1, respectively. The results demonstrated that the BCCPP algorithm could create a series of reasonable waypoints and travel orientations for robotics to scan the bridge-tower surface. The position of each waypoint comprised $x$, $y$, $z$, $o$, representing the coordinates and orientations of the robot (0, 1, 2, and 3 in $o$ represent up,

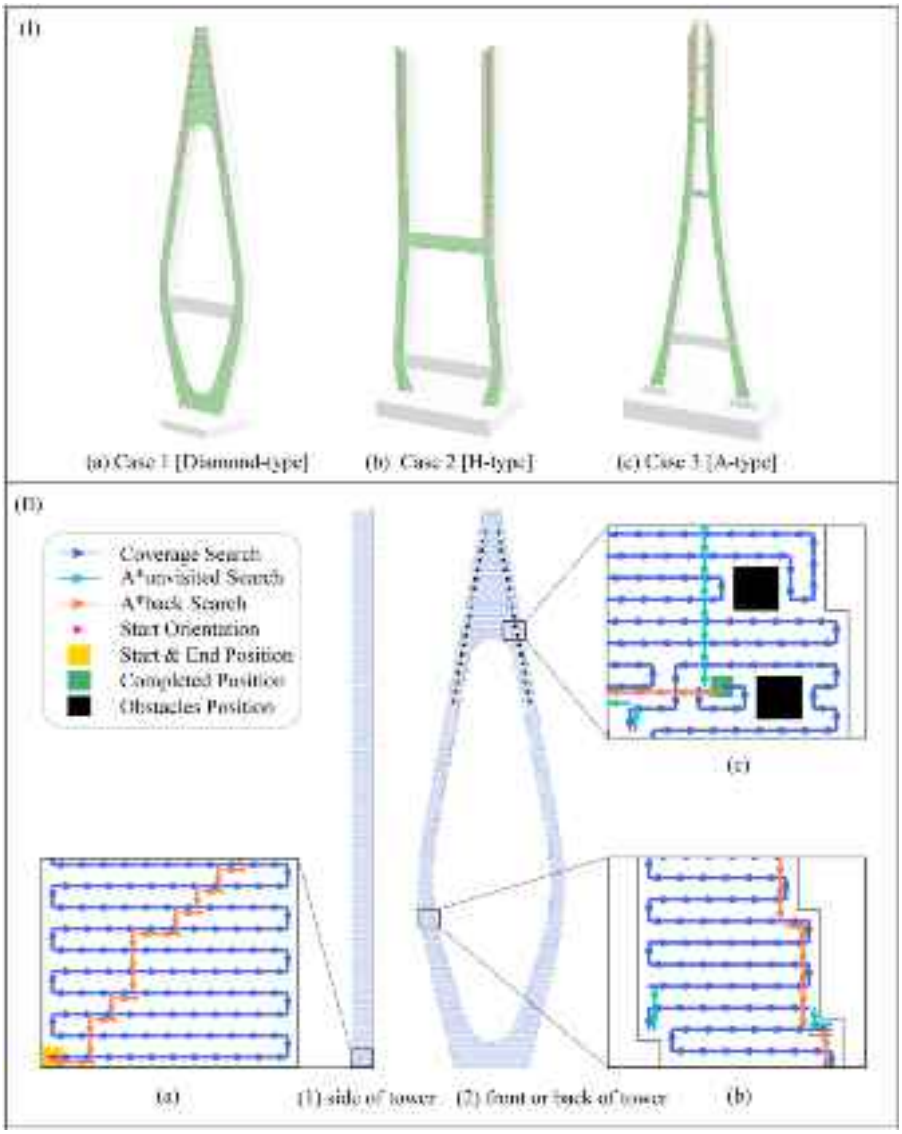


**FIGURE 13** Path-planning results: (I) Representative cases of cable-stayed bridge towers and (II) applying the proposed BCCPP algorithm to the BIM of the bridge tower in Case 1.

left, down, and right). The planning starts with a Coverage Search (Positions 1–5, Table 1; Positions 1–5, Table 2) and searches for unvisited areas with A*unvisited Search when the robot is caught in localized dilemmas (Positions 232–233, Table 2). Finally, the planning ends with A*Back Search finding a way to the start (Positions 3661–3974, Table 1; Positions 5609–5892, Table 2). Note that all projection distances between neighboring waypoints were approximately equal to 0.6 m, implying a reasonable coverage density. The orientation of each position indicates the direction in which the robot moves from the current to the next coordinate.

Figure 13II shows the planned paths. Each arrow points from the center of the previous grid to the next grid. The yellow areas represent the start and end positions of the robot. The robot starts from the yellow area and returns to the yellow area after completing all tasks. The green area represents the position at which the robot completes the inspection. The black areas represent the obstacle positions (cable locations). The blue arrow indicates that the robot uses the CCPP algorithm for path planning. The blue-green arrow indicates that the robot uses the A* algorithm to search for an unvisited position. The orange arrow indicates that the robot uses the A* algorithm to search for the end position. As shown in Figure 13II-a, the planned path in Coverage Search follows an "S" shape, indicating a high-efficiency coverage scanning. In addition, planning alternates between Coverage Search and A*Unvisited Search until the entire coverage is completed. This implies that the robot can apply A*Unvisited Search to continue working when caught in localized dilemmas in Coverage Search. This is illustrated in Figure 13II-b.

**TABLE 2** Waypoint sequence, positions, and planner status planned on the front surface of the tower.

| Seq. | Positions | Status |
|---|---|---|
| 1 | (−11.067, −3.500, 0.300, 3) | Coverage |
| 2 | (−10.469, −3.500, 0.300, 3) | Coverage |
| 3 | (−9.871, −3.500, 0.300, 3) | Coverage |
| 4 | (−9.272, −3.500, 0.300, 3) | Coverage |
| 5 | (−8.674, −3.500, 0.300, 3) | Coverage |
| … | … | … |
| 232 | (−11.665, −3.500, 2.700, 2) | A*unvisited |
| 233 | (−11.665, −3.500, 3.300, 0) | A*unvisited |
| 234 | (−11.665, −3.500, 3.900, 0) | Coverage |
| 235 | (−11.067, −3.500, 3.900, 3) | Coverage |
| 236 | (−10.469, −3.500, 3.900, 3) | Coverage |
| … | … | … |
| 5607 | (6.880, −3.500, 138.300, 0) | Coverage |
| 5608 | (6.281, −3.500, 138.300, 1) | Complete |
| 5609 | (5.683, −3.500, 138.300, 1) | A*Back |
| 5610 | (5.085, −3.500, 138.300, 1) | A*Back |
| 5611 | (4.487, −3.500, 138.300, 1) | A*Back |
| … | … | … |
| 5889 | (−10.469, −3.500, 2.100, 2) | A*Back |
| 5890 | (−10.469, −3.500, 1.500, 2) | A*Back |
| 5891 | (−10.469, −3.500, 0.900, 2) | A*Back |
| 5892 | (−10.469, −3.500, 0.300, 2) | A*Back |
| 5893 | (−11.067, −3.500, 0.300, 1) | Back |

Figure 13II-c shows the planning results for the obstacles (cables). The figure shows that the planned path in Coverage Search avoids potential collision areas reasonably and maintains an "S" shape, indicating that the planning could satisfactorily avoid obstacles. In conclusion, planning using the proposed algorithm provides reasonable internal operating logic.

## 6.2 | Evaluation of planning performance

To comparatively evaluate the planning performance of the proposed algorithm, STC, WF, BA*(boustrophedon + A*), RLA* (RL + A*), DRLA*(deep RL + A*) were selected as the initial benchmarks in tests for validation. The first three are the most representative classical CCPP algorithms, while the latter two are the most representative CCPP algorithms based on learning in recent years (Galceran & Carreras, 2013; Jie Wu et al., 2024). It is worth noting that algorithms based on learning are combined with A* here. Because it was found that new algorithms were not stable enough in the complex planning environment of this study and often fell into standstill, A* is introduced here to enhance the robustness of its planning.

In this study, the BCCPP algorithm refined the BA* algorithm by introducing a cost-optimization strategy based on multiple heuristics. To quantitatively evaluate the performance of the BCCPP algorithm, six metrics were designed based on the objective evaluation benchmark of coverage paths (Tan et al., 2021). Detailed definitions of the evaluation system are provided in Table 3.

Comparative experiments were conducted in the front surface of the selected representative cases since complexity of the front surface could highlight the performance of the planning algorithms. Figure 14 depicts various coverage paths by selected algorithms in Case 1. As can be seen, WF and refined BA* were the algorithms that generated well logical coverage paths ("S" type). The evaluation test results of each algorithm in each case are summarized in Table 4, where refined BA* emerged as the top-performing algorithm.

The superior performance of the proposed BCCPP algorithm stems from its dynamic heuristic strategy and FSM. Unlike conventional BA*, which relies on a single heuristic (e.g., Manhattan), BCCPP dynamically selects optimal heuristics. For instance, applying dynamic heuristic to BA* reduced total rotation by 11% on average, compared to BA* in single Manhattan in this case study (Table 4). This adaptability ensures efficient coverage while minimizing redundant movements. Additionally, the FSM enables seamless transitions between coverage modes (e.g., Coverage Search to A*Unvisited Search), preventing localized deadlocks and ensuring complete inspection. The integration of BIM further enhances robustness by predefining obstacle geometries (e.g., cables) and surface boundaries, eliminating reliance on error-prone sensor-based mapping. Experimental results validate that BCCPP achieves only 9.6% repetition and 113.6 min estimated executed time in average, which outperforms BA* (16.0% repetition and 124.4 min time), WF (12.6% repetition and 124.0 min time), and RLA* (40.5% repetition and 188.8 min time).

## 6.3 | Simulation experiment

To validate the executability of the path generated by BCCPP, the robot-executed paths in three cases were simulated in Gazebo under simulated conditions. The deviations between the robot-executed and planned robot paths originate from three factors: (1) limitations of localization algorithms in resolving positional ambiguities within feature-sparse environments (e.g., the expansive open spaces of bridge towers); (2) cumulative odometer drift from wheel slippage and numerical integration errors, modeled with a noise covariance matrix ( $\sigma_x^2 = 0.01\ m^2$, $\sigma_y^2 = 0.01\ m^2$, $\sigma_\theta^2 = 0.001\ \text{rad}^2$) and (3) potential geometric mismatches in BIM. In this study, the

**TABLE 3** Evaluation metrics of coverage path planning algorithm.

| Metric | Formula | Explanation |
|---|---|---|
| Path length | $\int d_{p(t)}dt$ | Total length of the path. $p(t)$: Robot's position at time $t$ |
| Rotation rads | $\int d_{o(t)}dt$ | Total rotation rads of the path $o(t)$: Robot's orientation at time $t$ |
| Coverage ratio | $S_{cov}/S_{total}$ | Percentage of coverage area. $S_{cov}$: Area of coverage regions $S_{total}$: Area of regions requiring coverage |
| Repetition ratio | $S_{rep}/S_{cov}$ | Percentage of the repetition coverage area $S_{rep}$: Area of repetition coverage regions |
| Turn ratio | $C_{sw}/C_{total}$ | Percentage of the turns. $C_{sw}$: Count of turns by robot $C_{total}$: Count of total steps by robot |
| Estimated executed time | $\frac{\int d_{o(t)}dt}{\bar{v}} + \frac{\int d_{p(t)}dt}{\bar{\omega}}$ | Estimated total time of the path to be executed $\bar{v}$: Average linear velocity of robot motion, est. 1 m/s $\bar{\omega}$: Average angular velocity of robot rotation, est. $\frac{\pi}{6}$ $rad/s$ |

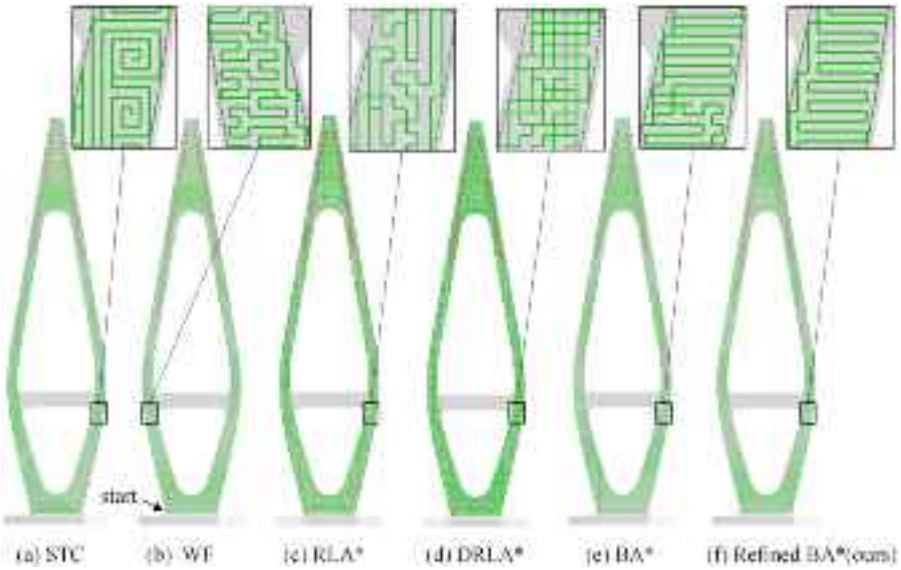


**FIGURE 14** Coverage process of various CCPP algorithms on front surface of the bridge tower in Case 1. RLA*, reinforcement learning + A*; STC, spanning tree coverage; WF, wavefront.

BIM applied in the simulation was rigorously aligned with the planning reference, eliminating geometric mismatches. Consequently, the dominant deviations arose from odometry drift and localization challenges in large-scale, structurally sparse environments. To ensure path executability, two criteria were enforced: (1) the executed path remained within a reasonable tolerance threshold of the planned path, and (2) all motions between consecutive waypoints adhered to workspace boundaries and collision-free constraints configured from the BIM.

Figure 15I shows two examples of the planned path in Case 1 executed by the climbing robot: paths near the edge of the bridge tower and around an obstacle (cable). Figure 15II lists the differences between the robot-executed path and planned path for the examples in Figure 15I. Notably, relative to the planned path, the maximum offsets of the executed path in the $x, z$ direction were within the maximum permissible tolerance of 0.2 m, and the executed path was not beyond the tower surface or in the potential collision area. Thus, the paths in these examples were executable.

All planned paths by BCCPP were tested in the robotic simulation experiments. The maximum speed and acceleration of the climbing robot were set to 1 m/s and 1.8 m/s$^2$. And the maximum angular speed and acceleration of the climbing robot were set to 2 rad/s and 6 rad/s$^2$. The result presents that, the planned paths for front and side surfaces of each bridge tower were executed successfully. And corresponding path length, rotation, and actual execution time were recorded in Table 5. These results affirm the executability of the planned path, even with deviations between the executed and planned paths.

**TABLE 4** Performance of various complete-coverage path planning (CCPP) algorithms. The best metrics are highlighted with bold.

| Case no. | Methods | Path length (m) | Rotation (rad) | Coverage ratio (%) | Repetition ratio (%) | Turn ratio (%) | Estimated executed time (min) |
|---|---|---|---|---|---|---|---|
| Case 1 (Diamond-type) | STC | 6488.4 | 3295.53 | 93.5% | 96.4% | **17.6%** | 213.0 |
| | WF | 3542.3 | 3267.26 | 93.5% | 9.2% | 31.1% | 156.3 |
| | RLA* | 4527.6 | 4283.56 | 93.5% | 41.0% | 34.6% | 211.9 |
| | DRLA* | 5888.4 | 10,741.11 | 93.5% | 78.0% | 64.1% | 440.0 |
| | BA* | 3713.4 | 2935.03 | 93.5% | 14.4% | 28.7% | 155.5 |
| | **Refined BA* (BCCPP)** | **3535.8** | **2748.11** | **93.5%** | **9.1%** | 28.0% | **146.0** |
| Case 2 (H-type) | STC | 4188.3 | **1551.95** | 97.8% | 96.5% | **12.4%** | 118.0 |
| | WF | 2381.4 | 2199.12 | 97.8% | 13.2% | 33.9% | 109.7 |
| | RLA* | 4586.4 | 3026.92 | 97.8% | 34.6% | 24.1% | 172.8 |
| | DRLA* | 5327.4 | 9380.80 | 97.8% | 75.6% | 62.1% | 387.4 |
| | BA* | 2431.8 | 2208.54 | 97.8% | 15.4% | 32.8% | 111.8 |
| | **Refined BA* (BCCPP)** | **2262.0** | 1814.27 | **97.8%** | **8.0%** | 28.1% | **95.5** |
| Case 3 (A-type) | STC | 3619.2 | 2093.41 | 92.6% | 95.0% | **18.5%** | 127.0 |
| | WF | 2105.4 | 2232.10 | 92.6% | 15.5% | 38.0% | 106.1 |
| | RLA* | 4702.2 | 3249.98 | 92.6% | 46.0% | 25.1% | 181.8 |
| | DRLA* | 5251.8 | 9302.26 | 92.6% | 72.8% | 62.6% | 383.6 |
| | BA* | 2161.8 | 2191.26 | 92.6% | 18.2% | 36.3% | 105.8 |
| | **Refined BA* (BCCPP)** | **2041.2** | **2051.46** | **92.6%** | **11.8%** | 35.8% | **99.3** |
| Average | STC | 4765.3 | 2299.64 | 94.6% | 96.0% | **16.2%** | 152.7 |
| | WF | 2676.4 | 2566.16 | 94.6% | 12.6% | 34.3% | 124.0 |
| | RLA* | 4605.4 | 3520.15 | 94.6% | 40.5% | 27.9% | 188.8 |
| | DRLA* | 5489.2 | 9809.06 | 94.6% | 75.5% | 62.9% | 403.7 |
| | BA* | 2769.0 | 2444.94 | 94.6% | 16.0% | 32.6% | 124.4 |
| | **Refined BA* (BCCPP)** | **2653.2** | **2204.61** | **94.6%** | **9.6%** | 30.6% | **113.6** |

Abbreviations: BA*, boustrophedon-A*; BCCPP, building information modeling-based CCPP; DRLA*, deep reinforcement learning + A*; RLA*, reinforcement learning + A*; STC, spanning tree coverage; WF, wavefront.
BA*

**TABLE 5** Actual execution time of BCCPP-planned paths for each surface in simulation experiments.

| Metrics\surface | Case 1—front | Case 1—side | Case 2—front | Case 2—side | Case 3—front | Case 3—side |
|---|---|---|---|---|---|---|
| Path length (m) | 3535.8 | 2385.0 | 2262.0 | 1792.2 | 2653.2 | 2217.0 |
| Rotation (rad) | 2748.11 | 958.18 | 1814.27 | 741.41 | 2051.46 | 827.80 |
| Actual execution time (min) | 109.3 | 57.9 | 69.2 | 44.5 | 82.7 | 55.6 |

Abbreviation: BCCPP, building information modeling-based CCPP.

## 6.4 | Field experiment

To investigate a climbing robot can execute planned task point messages and record image data in a real scenario, practical experiments were conducted targeting the whole front and side surface of Zhoushan Zhoudai Bridge tower (prototype bridge tower of Case 1).

The climbing robot achieves power supply through a direct a direct lightweight cable, while notably maintaining stable operational performance despite significant wind disturbances. After initial position calibration, the algorithm-generated path was transmitted to the climbing automated robot for execution, with photographs recorded at each assigned task-point. Subsequently, we

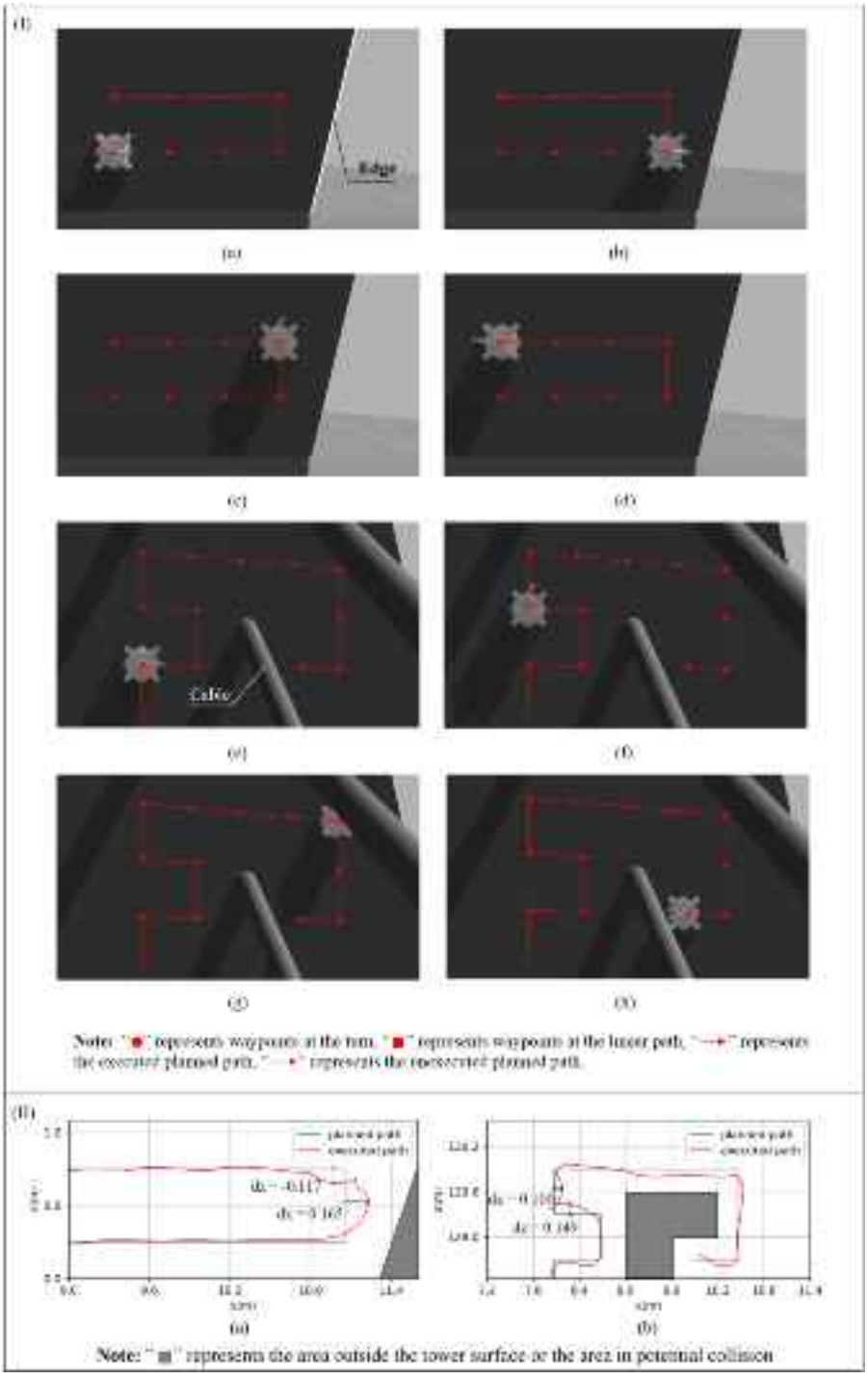


**FIGURE 15** Examples and results: (I) Robot executed planned paths in a simulated inspection environment. (a–d) close to the edge of the bridge tower, (e–h) around the diagonal cable of the bridge tower. (II) Maximum difference between the executed and planned paths for the examples presented in (I).

performed surface damage segmentation for the recorded photographs.

Figure 16I presents photographs captured from different areas in the front and side surfaces of the bridge tower, of which (a–c) were captured in the front surface, while (d–f) were captured in the side surface. This proves that the climbing robot equipped with an RGB camera under algorithmic control was able to maintain normal operation and acquire high-definition images in the presence of significant wind disturbances and the humid environment of the ocean, while going with no apparent deviation in its execution path. Figure 16II shows a clip of the field experiment, in which the climbing robot is performing an assigned task clip. The starting point position was (0, 0, 0, 0) in the base

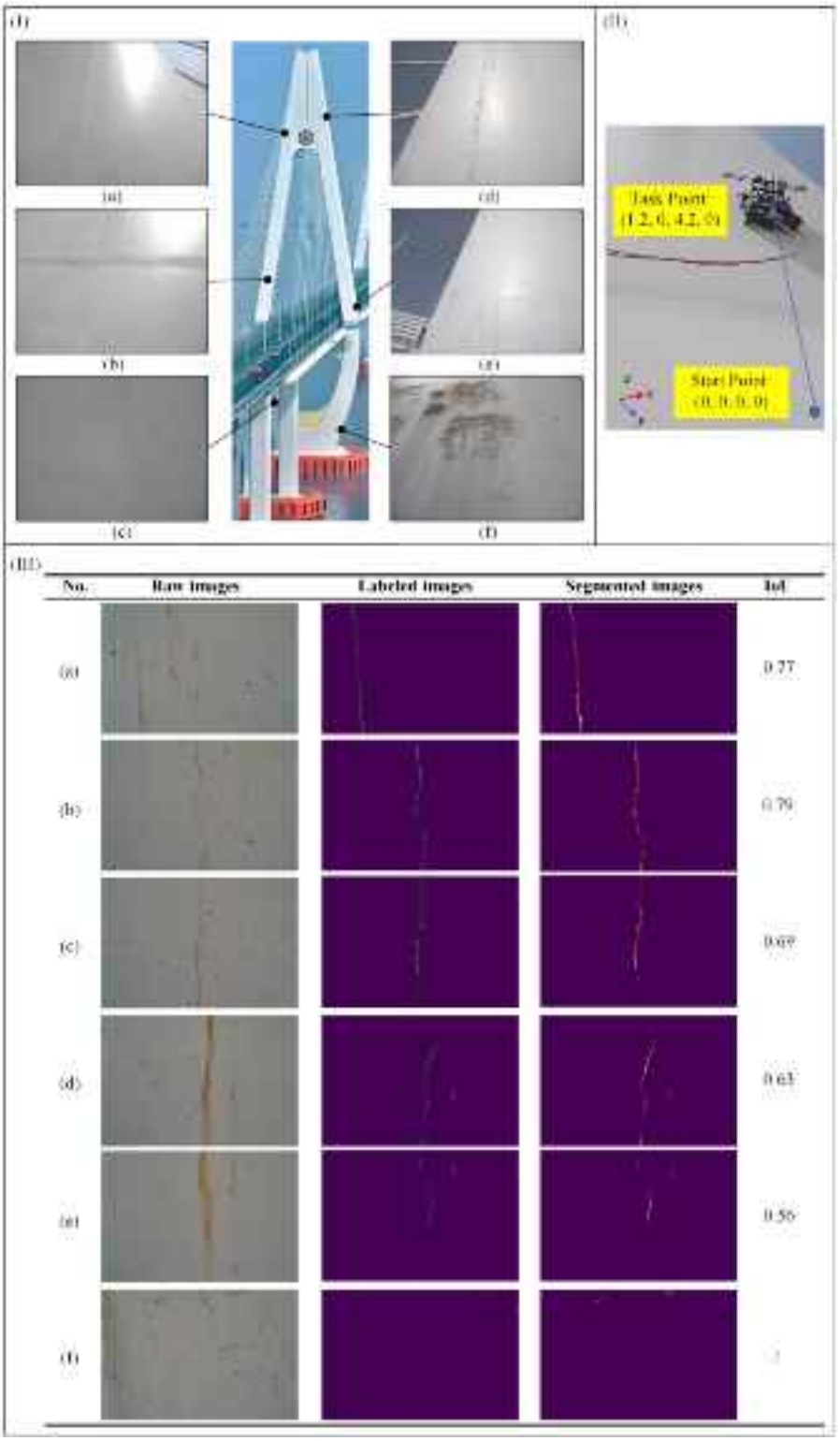


**FIGURE 16** Field Experiment: (I) Photographs captured from different areas in front and side surfaces of the bridge tower. (a–c) in front surface, (d–f) in side surface. (II) One clip of the field experiment. (III) IoU of images captured during clip of the field experiment, labeled images and segmented images.

coordinate system of the climbing robot. The goal of the climbing robot is to reach the assigned halfway task-point position of (1.2, 0, 4.2, 0). Raw images in Figure 16III show a selection of photographs recorded throughout the clip of the field experiment. From (a) to (f), the corresponding point positions are (0.6, 0, 4.2, 1); (0, 0, 3.6, 0); (0, 0, 3.0, 0); (0, 0, 1.8, 0); (0, 0, 1.2, 0); (0, 0, 0.6, 0). The reinforced-concrete damage-detection algorithm from our previous work (W. Ding et al., 2024) was employed in this pilot trial to process raw images. Figure 16III evaluates the processed result in terms of IoU using labeled images and segmented images. With IoU in segmented images (a-b) exhibiting values higher than 0.75, but values between 0.55 and 0.7 in images (c–e) and 0 in image (f), the algorithm is proved acceptable prediction in unstained areas, but still affected by the discontinuity of damage and staining.

# 7 | DISCUSSION AND CONCLUSION

## 7.1 | Discussion on planning result

In graphic level, the efficiency and logic of coverage path benefit from reasonable organization of boustrophedon and A* by FSM. In the quantitative level, BCCPP outperforms the five representative algorithms, STC, WF, BA*, RLA*, and DRLA*. In contrast to WF and BA*, BCCPP is refined from BA* and uses multiple heuristics for meritocracy instead of a single heuristic. Its advantage emerges in selecting a more appropriate heuristic for planning local map. As a result, the cost of path could be further reduced by an appropriate heuristic. STC underperforms, compared to WF and BA*, because its spanning tree pattern determines a high repetition ratio, resulting in inefficiency within complex environments. The learning-based algorithms, compared in this study, were trained based on general-purpose scenarios. Despite their suboptimal performance in the complex planning environments of this study, their performance remains fundamentally contingent upon intrinsic factors including network architectures, training hyperparameters, dataset characteristics, and applicable scenarios. In other words, the learning-based algorithms still hold a significant potential for improvement.

## 7.2 | Discussion on executive result

The executed path was within tolerance (0.2 m) and did not extend beyond the workspace or collide with obstacles. This indicated that the executability of the planned path was guaranteed, even with deviations between executed and planned paths.

## 7.3 | Discussion on field experiment

The proposed planning approach was validated in whole front and side surface of the real-world bridge tower. The experimental result shows that the climbing robot, under the control of the algorithm, is able to execute the coverage path and acquire high-definition data normally, even though the real bridge tower environment is significantly wind-disturbed and humid.

This study proposed a BIM-based complete-coverage path-planning (BCCPP) approach for the robotic inspection of cable-stayed bridge tower surfaces. This approach comprises three components: map generation, path generation, and a robotic task platform. The first two components function sequentially to generate a coverage path. The robotic task platform is to instruct robot executing path and collecting data. In addition, preliminary pilot experimental investigations on real climbing robots were conducted to validate the proposed approach. The major conclusions of this study are summarized as follows:

- **First**, BIM-driven map generation: This module resolves the challenges of mapping irregular bridge-tower surfaces successfully by leveraging BIM's geometric precision. Experimental result shows that the map coverage rate reaches 94.6% on average in three typical cases.
- **Second**, Heuristic-optimized path generation: In comparative experiments, the improved BA* algorithm achieves optimal efficiency, completing the inspection of the bridge tower's front surface (the most representative surface) in an estimated executed time of only 113.6 min, a 9.5% reduction in estimated executed time, compared to the baseline BA* algorithm in average. Key metrics included a 11% reduction in rotation and a 6.4% reduction in repetition ratio, ensuring energy-efficient trajectories.
- **Third**, robotic execution: The robotic task platform validates the planned paths without collisions and adherence to structural boundaries. Simulation experiments confirm a maximum positional tolerance of 0.2 m, critical for safe operation near obstacles (e.g., cables). And field experiment further demonstrated the feasibility of robotic platforms in field bridge tower environments.
- **Overall**, this study fills the knowledge gap in structural surface inspection by providing an approach for generating a complete-coverage path for robotic inspection of cable-stayed bridge-tower surfaces from the BIM to ROS platforms.

However, this study still has limitations. The approach proposed in this study is only applicable in cases where the BIM model is close to reality. Auxiliary procedures are performed to update and correct the model in real-time. In future works, we will explore the use of a LiDAR-equipped climbing robot for scanning an inspected surface to allow simultaneous updating of the building information model with the actual structure.

## ACKNOWLEDGMENTS

This work was partially supported by the National Key R&D Program of China under Grant Number (2023YFE0115000), National Natural Science Foundation of China (W2412092), Key R&D Program of Ningbo (2024H013). The project DiHICS received funding from the Malta Council for Science and Technology (MCST) and the Ministry for Science and Technology of the People's Republic of China (MOST), through the SINO-MALTA Fund 2023 Call (Science and Technology Cooperation).

## REFERENCES

Aftab, I., Kapitány, K., & Lovas, T. (2023). Automating scan to BIM operations using Grasshopper. *Periodica Polytechnica Civil Engineering*, *67*, 1187–1192.

Cao, M.-T. (2023). Drone-assisted segmentation of tile peeling on building façades using a deep learning model. *Journal of Building Engineering*, *80*, 108063.

Chea, C. P., Bai, Y., & Zhou, Z. (2024). Design and development of robotic collaborative system for automated construction of reciprocal frame structures. *Computer-Aided Civil and Infrastructure Engineering*, *39*, 1550–1569.

Chen, J., Lu, W., & Lou, J. (2023). Automatic concrete defect detection and reconstruction by aligning aerial images onto semantic-rich building information model. *Computer-Aided Civil and Infrastructure Engineering*, *38*, 1079–1098.

Chen, Z., Wang, H., Chen, K., Song, C., Zhang, X., Wang, B., & Cheng, J. C. P. (2024). Improved coverage path planning for indoor robots based on BIM and robotic configurations. *Automation in Construction*, *158*, 105160.

Choset, H. (2000). Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots*, *9*, 247–253.

Ding, L., Jiang, W., Zhou, Y., Zhou, C., & Liu, S. (2020). BIM-based task-level planning for robotic brick assembly through image-based 3D modeling. *Advanced Engineering Informatics*, *43*, 100993.

Ding, W., Shu, J., Debono, C. J., Prakash, V., Seychell, D., & Borg, R. P. (2024). Quantitative assessment of cracks in concrete structures using active-learning-integrated transformer and unmanned robotic platform. *Automation in Construction*, *168*, 105829.

Dogru, S., & Marques, L. (2022). ECO-CPP: Energy constrained online coverage path planning. *Robotics and Autonomous Systems*, *157*, 104242.

Farley, A., Wang, J., & Marshall, J. A. (2022). How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion. *Simulation Modelling Practice and Theory*, *120*, 102629.

Follini, C., Magnago, V., Freitag, K., Terzer, M., Marcher, C., Riedl, M., Giusti, A., & Matt, D. T. (2020). BIM-integrated collaborative robotics for application in building construction and maintenance. *Robotics*, *10*, 2.

Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, *61*, 1258–1276.

Gao, G.-Q., & Xin, B. (2019). A-STC: Auction-based spanning tree coverage algorithm formotion planning of cooperative robots. *Frontiers of Information Technology & Electronic Engineering*, *20*, 18–31.

Gao, L., Lv, W., Yan, X., & Han, Y. (2022). Complete coverage path planning algorithm based on energy compensation and obstacle vectorization. *Expert Systems with Applications*, *203*, 117495.

Gao, Y., Meng, J., Shu, J., & Liu, Y. (2022). BIM-based task and motion planning prototype for robotic assembly of COVID-19 hospitalisation light weight structures. *Automation in Construction*, *140*, 104370.

Gao, Y., Shu, J., Xia, Z., & Luo, Y. (2025). From muscular to dexterous: A systematic review to understand the robotic taxonomy in construction and effectiveness. *Journal of Field Robotics*, *42*, 180–205.

Gao, Y., Shu, J., Xiao, W., & Jin, Z. (2023). Polyhedron-bounded collision checks for robotic assembly of structural components. *Automation in Construction*, *152*, 104904.

Gu, S., Fu, Z., Chen, W., Guan, Y., Wu, H., Zhou, X., & Zhu, H. (2025). Navigation of biped wall-climbing robots using BIM and ArUco markers. *Robotica*. Advance online publication. https://doi.org/10.1017/S0263574724002170

Hattori, K., Oki, K., Sugita, A., Sugiyama, T., & Chun, P. (2024). Deep learning-based corrosion inspection of long-span bridges with BIM integration. *Heliyon*, *10*(15), e35308.

Huang, X., Liu, Y., Huang, L., Stikbakke, S., & Onstein, E. (2023). BIM-supported drone path planning for building exterior surface inspection. *Computers in Industry*, *153*, 104019.

Jang, K., An, Y.-K., Kim, B., & Cho, S. (2020). Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot. *Computer-Aided Civil and Infrastructure Engineering*, *36*, 14–29.

Javadinasab Hormozabad, S., Gutierrez Soto, M., & Adeli, H. (2021). Integrating structural control, health monitoring, and energy harvesting for smart cities. *Expert Systems*, *38*, 12845.

Jonnarth, A., Zhao, J., & Felsberg, M. (2024). *Learning coverage paths in unknown environments with deep reinforcement learning*. arXiv.

Kim, S., Peavy, M., Huang, P.-C., & Kim, K. (2021). Development of BIM-integrated construction robot task planning and simulation system. *Automation in Construction*, *127*, 103720.

Kyaw, P. T., Paing, A., Thu, T. T., Mohan, R. E., Vu Le, A., & Veerajagadheswar, P. (2020). Coverage path planning for decomposition reconfigurable grid-maps using deep reinforcement learning based travelling salesman problem. *IEEE Access*, *8*, 225945–225956.

Leibbrandt, A., Caprari, G., Angst, U., Siegwart, R. Y., Flatt, R. J., & Elsener, B. (2012). Climbing robot for corrosion monitoring of reinforced concrete structures. *2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI)*, Zurich, Switzerland (pp. 10–15).

Li, X. (2022). Research on cooperative control technology and application of wall-climbing robot and UAV based on multi-sensing fusion. *Journal of Applied Science and Engineering*, *26*, 651–662.

Liu, Y., Dai, Q., & Liu, Q. (2013). Adhesion-adaptive control of a novel bridge-climbing robot. *2013 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, Nanjing, China (pp. 102–107).

Ma, L., Chen, Q., Hartmann, T., & Zeng, C. (2024). *BIM-supported indoor path planning for wall-climbing robots*. http://dx.doi.org/10.2139/ssrn.5012303

Mannan, A., Obaidat, M. S., Mahmood, K., Ahmad, A., & Ahmad, R. (2023). Classical versus reinforcement learning algorithms for unmanned aerial vehicle network communication and coverage path planning: A systematic literature review. *International Journal of Communication Systems*, *36*(5), e5423.

Alonso Medina, P., & León González, J. (2022). Reinforced concrete long-term deterioration prediction for the implementation of a Bridge Management System. *Materials Today: Proceedings*, *58*, 1265–1271.

Muñoz, J., López, B., Quevedo, F., Monje, C. A., Garrido, S., & Moreno, L. E. (2021). Multi UAV coverage path planning in urban environments. *Sensors*, *21*(21), 7365.

Muthugala, M. A. V. J., Samarakoon, S. M. B. P., & Elara, M. R. (2022). Toward energy-efficient online Complete Coverage Path Planning of a ship hull maintenance robot based on Glasius Bio-inspired Neural Network. *Expert Systems with Applications*, *187*, 115940.

Nguyen, H. D., Choi, M., & Han, K. (2023). Risk-informed decision-making and control strategies for autonomous vehicles in emergency situations. *Accident Analysis & Prevention*, *193*, 107305.

Nguyen, S. T., Pham, A. Q., Motley, C., & La, H. M. (2020). A practical climbing robot for steel bridge inspection. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France (pp. 9322–9328).

Odugu, O., Ghafari, F., Shourangiz, E., Khan, M. T., & Wang, C. (2024). Building Information Model (BIM) and robotic systems integration for construction: A comprehensive workflow analysis and future perspectives. In H. Degen & S. Ntoa (Eds.), *Lecture notes in computer science: Vol. 14736. Artificial intelligence in HCI* (pp. 272–282). Cham: Springer.

Pan, X., Yang, T. Y., Xiao, Y., Yao, H., & Adeli, H. (2023). Vision-based real-time structural vibration measurement through deep-learning-based detection and tracking methods. *Engineering Structures*, *281*, 115676.

Pezeshki, H., Adeli, H., Pavlou, D., & Siriwardane, S. C. (2023). State of the art in structural health monitoring of offshore and marine structures. *Proceedings of the Institution of Civil Engineers—Maritime Engineering*, *176*, 89–108.

Ren, T., & Jebelli, H. (2024). Efficient 3D robotic mapping and navigation method in complex construction environments. *Computer-Aided Civil and Infrastructure Engineering*. Advance online publication. https://doi.org/10.1111/mice.13353

Rossi, M., & Bournas, D. (2023). Structural health monitoring and management of cultural heritage structures: A state-of-the-art review. *Applied Sciences*, *13*, 6450.

Tan, C. S., Mohd-Mokhtar, R., & Arshad, M. R. (2021). A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. *IEEE Access*, *9*, 119310–119342.

Tian, Y., Chen, C., Sagoe-Crentsil, K., Zhang, J., & Duan, W. (2022). Intelligent robotic systems for structural health monitoring: Applications and future trends. *Automation in Construction*, *139*, 104273.

Toan, N. V., Hoang, M. D., Khoi, P. B., & Yi, S.-Y. (2023). The human-following strategy for mobile robots in mixed environments. *Robotics and Autonomous Systems*, *160*, 104317.

Viet, H. H., Dang, V.-H., Laskar, M. N. U., & Chung, T. (2013). BA*: An online complete coverage algorithm for cleaning robots. *Applied Intelligence*, *39*, 217–235.

Wu, J., Cheng, L., Chu, S., & Song, Y. (2024). An autonomous coverage path planning algorithm for maritime search and rescue of persons-in-water based on deep reinforcement learning. *Ocean Engineering*, *291*, 116403.

Wu, J., Wang, X., Huang, L., Wang, Z., Wan, D., & Li, P. (2023). Parameterized site selection approach of park entrance based on crowd simulation and design requirement. *Applied Sciences*, *13*, 6280.

Xiao, Y., Yang, T. Y., & Xie, F. (2025). Autonomous construction framework for crane control with enhanced soft actor–critic algorithm and real-time progress monitoring. *Computer-Aided Civil and Infrastructure Engineering*, Advance online publication. https://doi.org/10.1111/mice.13427

Yi, L., Wan, A. Y. S., Le, A. V., Hayat, A. A., Tang, Q. R., & Mohan, R. E. (2023). Complete coverage path planning for reconfigurable omni-directional mobile robots with varying width using GBNN(n). *Expert Systems with Applications*, *228*, 120349.

Zhao, J., Li, X., & Bai, J. (2018). Experimental study of vortex suction unit-based wall-climbing robot on walls with various surface conditions. *Proceedings of the Institution of Mechanical Engineers*, *232*, 3977–3991.

Zhou, Q., & Li, X. (2018). Experimental investigation on climbing robot using rotation-flow adsorption unit. *Robotics and Autonomous Systems*, *105*, 112–120.

Zhou, X., & Zhang, X. (2019). Thoughts on the development of bridge technology in China. *Engineering*, *5*(6), 1120–1130.

Zhu, H., Ye, W., Ye, P., Hou, Y., Chen, W., & Guan, Y. (2024). Structured light vision-based automatic gripping of pipes in biped climbing robots. *IEEE Sensors Journal*, *24*(14), 22156–22167.

Zidane, I. M., & Ibrahim, K. (2018). Wavefront and A-star algorithms for mobile robot path planning. *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017 (AISI 2017)*, Cairo, Egypt (pp. 69–80).

**How to cite this article:** Xia, Z., Shu, J., Ding, W., Gao, Y., Duan, Y., Debono, C. J., Prakash, V., Seychell, D., & Borg, R. P. (2025). Complete-coverage path planning for surface inspection of cable-stayed bridge tower based on building information models and climbing robots. *Computer-Aided Civil and Infrastructure Engineering*, *40*, 4149–4171. https://doi.org/10.1111/mice.13469