


RESEARCH ARTICLE

Navigation of biped wall-climbing robots using BIM and ArUco markers

Shichao Gu^{1,2,3,4} , Ziyang Fu², Weinan Chen², Yisheng Guan², Hongmin Wu^{3,4}, Xuefeng Zhou^{3,4} and Haifei Zhu²

¹Guangzhou Maritime University, Guangzhou, China

²School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, China

³Institute of Intelligent Manufacturing, Guangdong Academy of Sciences, Guangzhou, China

⁴Guangdong Key Laboratory of Modern Control Technology, Guangzhou, China

Corresponding author: Haifei Zhu; Email: hfzhu@gdut.edu.cn

Received: 17 September 2024; **Revised:** 22 November 2024; **Accepted:** 27 November 2024

Keywords: biped wall-climbing robot; environment map; localization; climbing path correction; BIM; ArUco marker

Abstract

Biped wall-climbing robots (BWCRs) serve as viable alternatives to human workers for inspection and maintenance tasks within three-dimensional (3D) curtain wall environments. However, autonomous climbing in such environments presents significant challenges, particularly related to localization and navigation. This paper presents a pioneering navigation framework tailored for BWCRs to navigate through 3D curtain wall environments. The framework comprises three essential stages: Building Information Model (BIM)-based map extraction, 3D climbing path planning (based on our previous work), and path tracking. An algorithm is developed to extract a detailed 3D map from the BIM, including structural elements such as walls, frames, and ArUco markers. This generated map is input into a proposed path planner to compute a viable climbing motion. For path tracking during actual climbing, an ArUco marker-based global localization method is introduced to estimate the pose of the robot, enabling adjustments to the target foothold by comparing desired and actual poses. The conducted experiments validate the feasibility and efficacy of the proposed navigation framework and associated algorithms, aiming to enhance the autonomous climbing capability of BWCRs.

1. Introduction

Curtain walls are widely employed in modern architecture due to their integration of esthetic appeal, functionality, energy efficiency, and structural integrity. The high-rise tasks related to the curtain walls, such as cleaning, maintenance, and repair, currently depend primarily on manual labor, which is both time-consuming and labor-intensive, while also presenting significant safety risks. Climbing robots with flexible mobility and reliable attachment capabilities are considered ideal assistants or substitutes for performing these operations [1]. In recent decades, many wall-climbing robots have been developed, including wheeled, tracked, foot, and hybrid motion modes. The attachment methods include vacuum adsorption, magnetic adsorption, hook claw, and other biomimetic structures. In environments comprising frames and glass fillers, climbing robots must possess the capability to overcome obstacles and transition across distinct walls. A bipedal climbing robot, featuring a multi-degree-of-freedom series manipulator as its trunk and vacuum adsorption devices at both ends, is regarded as a preferred solution. This approach is exemplified by systems such as ROMA II [2], MATS [3], CROC [4], and W-Climbot [5, 6] (as shown in Figure 1). However, research to date has predominantly concentrated on developing prototypes, with limited focus on addressing challenges related to positioning and navigation, which are critical for enhancing autonomous climbing capabilities.

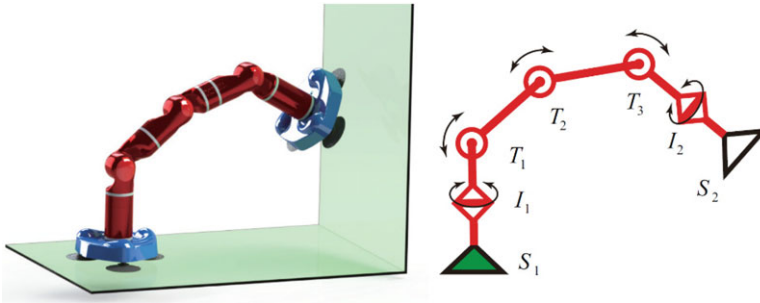


Figure 1. *W-Climbot built with our third generation of modules and its mechanism schematic diagram.*

To climb autonomously in an unknown wall scenario, the environment map must first be constructed, followed by the planning of the climbing path based on task requirements. During the execution phase, BWCRs track the planned path through environmental perception, self-localization, online path replanning, and autonomous attachment. Unlike the environments encountered by ground mobile robots, BWCRs operate within complex building environments comprising multiple planes arranged in 3D space with defined boundaries. Structural elements such as walls, columns, beams, and windows may serve as obstacles or attachment targets and require accurate perception and modeling. However, the uniform texture of many structural elements and the transparent nature of glass significantly degrade or even nullify the perception capabilities of most sensors, posing substantial challenges to environment map construction and robot localization. Additionally, the complex structure of these environments complicates path planning for BWCRs, which rely on discontinuous and discrete motion. Robots must identify a sequence of footholds from the starting point to the destination and plan a collision-free trajectory for the swinging end to transition between footholds. A hierarchical planning framework for global wall sequence, foothold selection, and single-step motion has been proposed in our prior work [6], forming the foundation of this study. Accurate long-term path tracking requires BWCRs to estimate their global position and orientation within the environment and correct motion deviations promptly. While ground mobile robots and UAVs need to estimate no more than four degrees of freedom (DoFs) for self-positioning, BWCRs must estimate six DoFs, in addition to addressing the complexities of curtain wall environment perception, which exacerbates the challenges of localization.

This paper introduces a navigation framework for BWCRs in large curtain wall buildings, utilizing BIM, ArUco markers, and visual sensors to achieve robot localization and climbing path deviation correction. To address the sensing challenges in 3D curtain wall environments, a BIM-based climbing map construction method for large-scale building environments is developed. The geometric information of structural elements is directly extracted from BIM to generate a map representation of the climbing environment. ArUco marker information is incorporated into the map, and onboard cameras are utilized to recognize markers, estimate pose, and derive the global pose of the robot's swinging end using map data association. Using localization information and the resulting planned path as input, a climbing path correction strategy for BWCRs is proposed. The foothold and single-step trajectory are adjusted based on path deviation feedback, ensuring that the robot continuously tracks the planned path to reach the target point. A bipedal wall-climbing robot navigation system integrating visual positioning, deviation correction, and autonomous adsorption has been developed. The feasibility and effectiveness of the proposed theory and algorithms have been validated through a series of simulations and experiments.

The remainder of this paper is organized as follows. State-of-the-art climbing mapping methods, localization, and navigation methods are reviewed in Section 2. The navigation problem of BWCRs is introduced in Section 3. The BIM-based map extraction method is proposed in Section 4. The resulted path-tracking method including the global localization and climbing path correction is presented in Section 5. In Section 6, the development of the biped wall-climbing robot system is introduced.

Experimental results verify the performances of the proposed methods and algorithms in Section 7. Finally, concluding statements and directions of future research are presented in Section 8.

2. Related works

Methods for BWCRs to construct maps can be categorized into two types. The first involves mapping the environment by collecting sensor data, with the sensors either integrated into the robot or fixed within the environment. The second type is the environment model-based method. This method constructs the map by extracting relevant geometric and semantic information from models or electronic maps, such as BIM and Computer-Aided Design (CAD) models of buildings or 2D/3D electronic maps of the environment.

A quadruped climbing robot [7], MRWALLSPECT IV, used a 2DOF long-range scanner, 5 short-range sensors, and a gyroscope to perceive the environment, then a grid model was obtained to rebuild a surface that most fit the real environment. An articulated wall-climbing robot acquired data and created the 3D surface map with both a laser range finder and a low-resolution camera [8]. An inchworm climbing robot adopted an RGB-D sensor to inspect areas inside steel bridge archways [9]. To obtain the 3D map, the end-effector transform calculated from a kinematic chain robot model was used to project rays into a partial map. To get the CAD model of truss-style structures, the RTAB-map algorithm was selected to build the environment model with the RGB-D sensors [10]. Then, a Pouring Segmentation Algorithm was presented to get the truss point-cloud model, and the Truss Parametric Expression Algorithm output the parametric expression of each frame. To build a representation of the environment, a 3D planar structure modeling and parameterization system were proposed, which included a lidar-IMU fusion method to map and parameter expression by extracting the vertexes of planes [11].

The above schemes are implemented by robots carrying sensors. In contrast, there are few research achievements in building maps by collecting environmental data from external sensors. Chen et al [12] developed an integrated approach for constructing a complete 3D map using three ground robots and one City-Climber robot, which were equipped with panoramic cameras and rotary laser range sensors. A deterministic method and a probabilistic method were proposed to solve the perspective three points camera pose estimation problem. A 3D map was constructed by integrating the four individual maps using the geometric information acquired by an intra-robot localization method. Similarly, Tavakoli et al [13] also proposed autonomous 3D mapping methods based on cooperation between one climbing robot with some fixed markers and ground robots equipped with wide angle Visual Graphics Array camera. This kind of map-building method is easily affected by the environmental occlusion problem and the flatness of the ground, and gradually degrades as the complexity of the environment increases.

Due to the noise of sensor data, the calibration error between sensors and robots, and the influence of mapping algorithm precision, the map obtained by sensors has a large error. Because BIM contains a large amount of geometric and semantic indoor information about buildings, and with the improvement of construction accuracy, it is becoming a trend to use BIM to build maps for robot positioning. Park et al [14] translated the BIM model environment into a navigation site map with representations of nodes and edges, which was utilized for path planning of the mobile robot. The effective information was extracted from BIM to reorganize entity and network models respectively [15]. Then, these two models were organically connected through the connections between elements in different models and established a hybrid map. Moura et al [16] converted BIM models in Industry Foundation Classes format to geometrical representations and resulted in a pose-graph structure that is appended to the serialized state representation of the target environment. Chen et al [17] proposed a multi-layer map generation method, including the lifecycle layer, reality layer, and UGV layer, to convert Operation and Maintenance BIM to the grid map for robot path planning.

Robot localization technology can be divided into relative localization and absolute localization. The former determines the current pose of the robot by measuring the distance and orientation of the robot relative to the initial position [18]. Inertial navigation and dead reckoning are two common methods in

relative localization. They mainly use sensors such as the inertial measurement unit (IMU) and encoder on the robot for motion measurement and estimation [19]. The positioning accuracy depends on the sensor accuracy. The wheel slip of the robot and integration error of the algorithms are the main factors affecting the positioning accuracy. Absolute localization is generally to obtain the pose of the robot in the global coordinate system by perceiving the fixed objects or features in the environment, including the beacon-based positioning method and SLAM [20]. Beacon-based positioning method needs to arrange beacons in advance in the environment. In order to obtain the current position information, robots need to carry sensors to establish contact with the beacon. The mainstream solutions include infrared [21], ultrasonic [22], WiFi [23], Bluetooth, and ultra-wideband positioning [24]. However, this kind of method cannot directly obtain the pose of the robot and has the problems of low positioning accuracy and weak penetration ability, so it is difficult to apply to situations such as wall occlusion. SLAM algorithm has been widely used in ground mobile robots [25]. However, due to the particularity of the climbing environment and motion form of BWCR, the existing SLAM methods are difficult to be directly applied to BWCRs.

To navigate within a constructed map, BWCRs must possess path-planning capabilities. In addition to planning a trajectory that connects the starting point and the target point, it is essential to determine the robot's pose using localization methods and correct it when deviations occur. The core requirement for point-to-point path planning in BWCRs is to identify an accessible climbing path, including a sequence of footholds and collision-free trajectories between adjacent footholds. A number of climbing path-planning methods have been proposed [26, 27]. Our proposed three-level path-planning algorithm serves as the technical foundation of this paper for generating climbing paths [6]. While this planning method plays an integral role in the overall navigation system, this paper primarily focuses on robot localization and navigation within glass curtain wall environments. The path planning associated with navigation is primarily focused on correcting the robot's motion to follow the planned climbing path. However, the literature has rarely addressed the issue of climbing path correction for BWCRs. Shih et al [28] proposed a path correction and obstacle avoidance method for a bipedal intelligent robot that utilizes ultrasonic and electronic compass sensors to detect environmental information. Similarly, 3DCLIMBER could correct various errors using a self-calibration algorithm based on accelerometers and infrared range sensors [29]. A correction algorithm for wall-climbing robots has been introduced to minimize errors by transforming them from Cartesian space to joint space using a Jacobian matrix [30]. To enable the wall-climbing robot to track the desired path, dead reckoning and a "Preview-follow" theory-based algorithm were employed, ensuring accurate and efficient movement [31].

3. Problem statement and our solutions

3.1. BWCR navigation problem

The BWCR navigation means that the biped wall-climbing robot can build a map of the curtain wall environment, plan a 3D climbing path according to the set destination, and track the resulted path until reaching the target.

A curtain wall environment can be represented as,

$$M = \{\boldsymbol{\varepsilon}_i \mid i = 1, 2, \dots, N_e\} \quad (1)$$

where $\boldsymbol{\varepsilon}_i$ is the structural elements such as wall planes, frames (i.e., beams and columns), and features detected for localization, and N_e denotes the number of the elements. The environment map can be obtained through SLAM using appropriate sensors or CAD models of buildings.

Figure 2 shows a scenario in which a BWCR navigates in a 3D curtain wall environment. After constructing the environment map, one feasibility climbing path is generated based on a hierarchical 3D climbing path planner [6], which consists of a sequence of footholds (gray triangles in Figure 2) connecting the start point and the destination, and single-step collision-free motion of each adjacent footholds (solid curves in Figure 2). The robot is placed in the starting position with an initial configuration and begins to climb along the planned path. Due to the influence of initial position deviation,

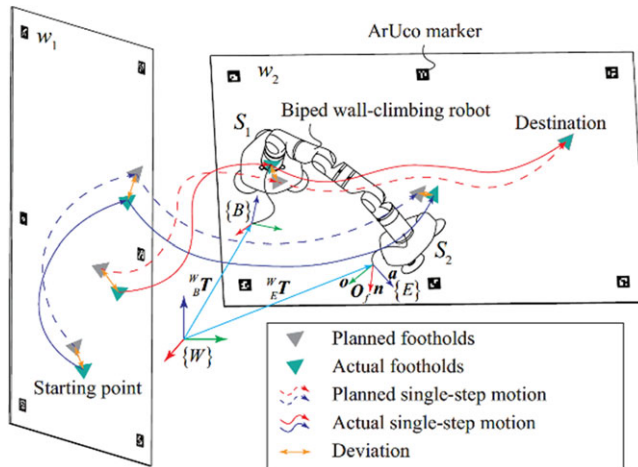


Figure 2. BWCRCs navigate in a curtain wall environment.

motion deviation, environmental geometric errors, and the elastic deformation of the suction cup at the base, the robot may deviate from the predetermined path. Therefore, BWCR needs to constantly estimate its global position and correct the error in time, while meeting the kinematic constraints and avoiding obstacles. The resulted path-tracking model can be represented as,

$$\min d = \|f_p - f_a\| \tag{2}$$

where f_p denotes the planned foothold through the proposed climbing path planner [6], and f_a is the actual adsorption point of BWCR, which can be obtained by minimizing the deviation d between f_p and f_a . Therefore, it is necessary to determine the pose of the swinging end using an appropriate sensing system and a corresponding method.

3.2. Proposed solution

Figure 3 depicts an overview of the proposed localization and navigation framework. To address the limitations of vision and other sensors in perceiving glass curtain walls, geometric information about environmental elements is extracted from the corresponding BIM to generate an environment map. Wall and frame information is used to plan a collision-free climbing route, and marker features are utilized for robot localization. A camera integrated into the suction module detects the ArUco marker, enabling the determination of the local pose of the swinging end within the marker coordinate frame. The global pose is then calculated based on the environment map. If deviations exist, appropriate motion correction strategies are employed to generate corrected footholds and plan collision-free single-step motions for various climbing states, such as moving on a single wall or transitioning between walls. Control parameters are sent to the robot controller to move the swinging end to the new foothold and ensure reliable adhesion to the wall surface. These operations are repeated until the robot reaches its destination.

4. BIM-based map extraction

4.1. Environment map and its mathematical description

Curtain walls are widely used in large modern architecture, such as airports, train stations, cultural venues, financial centers, and commercial hotels. To facilitate understanding, the environment can be simplified as a combination of walls and supporting frames. Thus, the map M can be represented in the

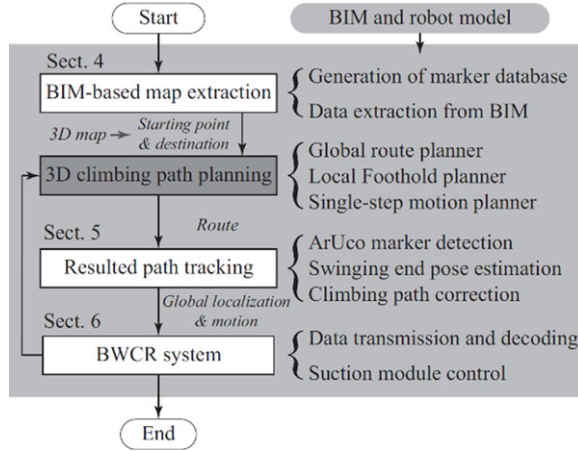


Figure 3. Overview of BWCR navigation method.

world coordinate frame $\{W\}$ as,

$$\begin{aligned}
 M = \{ & \{\omega_i \mid i = 1, 2, \dots, N_m, N_m \geq 1\}, \\
 & \{\alpha_j \mid j = 1, 2, \dots, N_a, N_a \geq 1\}, \\
 & \{\varsigma_k \mid k = 1, 2, \dots, N_s, N_s \geq 0\} \}
 \end{aligned} \tag{3}$$

where ω_i denotes the i -th wall, α_j is the j -th marker, and ς_k is the k -th frame. A flat wall can be represented as a convex polygon with vertices,

$$\omega_i = ({}^W\mathbf{v}_\gamma, \mathbf{n}_i) \tag{4}$$

where ${}^W\mathbf{v}_\gamma$ is the ordered vertices, and ${}^W\mathbf{v}_\gamma = \{{}^W\mathbf{P}_i \mid i = 1, 2, \dots, N, N > 3\}$. \mathbf{n}_i is the normal vector of ω_i , and indicates the surface of the wall allowed to climb on. Note that, a concave polygon can always be divided into a limited number of convex polygons.

The ArUco marker is selected for locating the BWCR due to its effectiveness, robustness against detection errors and occlusion, and capability to directly estimate the camera’s relative pose [32]. A mark on the wall ω_i can be expressed as,

$$\alpha_i = \{ID, i, {}^W\boldsymbol{\vartheta}_a, {}^W\mathbf{T}_j\} \tag{5}$$

where ID is the unique identifier of a marker, i represents the serial number of the wall where the marker is located, ${}^W\boldsymbol{\vartheta}_a = \{{}^W\mathbf{P}_i \mid i = 1, 2, 3, 4\}$ is the set of the four corners in a clockwise direction, while ${}^W\mathbf{T}_j$ denotes 4×4 homogeneous transformation matrix from the ArUco marker coordinate frame to the world coordinate frame, which constitutes the ArUco marker database.

For a BWCR, the walls are the passable area, while the frames and connectors are the obstacles, which can be described as a minimum envelope cylinder. Thus, a frame can be expressed with the following parameter equation

$$\varsigma_k = \{{}^W\mathbf{P}_s, {}^W\mathbf{P}_e, r\} \tag{6}$$

Where ${}^W\mathbf{P}_s, {}^W\mathbf{P}_e$ and denote the two endpoints and radius of the simplified cylinder respectively.

4.2. ArUco marker database

Figure 4 shows us the model of solving the transformation matrix ${}^W\mathbf{T}$. The coordinates of the four corners ${}^W\boldsymbol{\vartheta}_a$ of the ArUco marker can be extracted from the BIM model. Each point ${}^W\mathbf{P}_i$ can also be

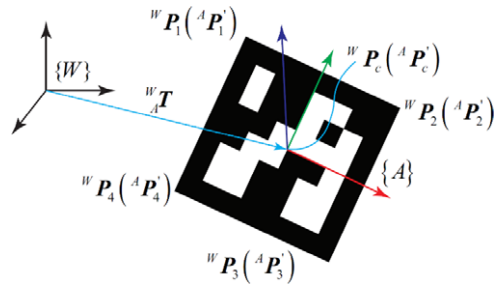


Figure 4. Calculation of transformation matrix from the marker to the world coordinate frame.

calculated through the corresponding corner ${}^A P'_i$ described in the marker coordination frame $\{A\}$. The transformation formula can be written as,

$$\begin{bmatrix} {}^w P_i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & t \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^A P'_i \\ 1 \end{bmatrix} \tag{7}$$

where \mathbf{R} and t are rotation component and translation component of ${}^w_A T$. To solve those two components, the iterative closest point (ICP) algorithm [33] can be used by converting the calculation process into the pose estimation problem with known feature point pairs.

First, calculating the centroids ${}^w P_c$ and ${}^A P'_c$ sets of corresponding points

$${}^w P_c = \frac{1}{n} \sum_{i=1}^n P_i, \quad {}^A P'_c = \frac{1}{n} \sum_{i=1}^n P'_i \tag{8}$$

Then, the centroid-removed coordinates of each point can be obtained

$${}^w Q_i = {}^w P_i - {}^w P_c, \quad {}^A Q_i = {}^A P'_i - {}^A P'_c \tag{9}$$

After that, a 3×3 matrix can be calculated by

$$\mathbf{Q} = \sum_{i=1}^n {}^w Q_i {}^A Q_i^T \tag{10}$$

The SVD method [34] is used to decompose the matrix \mathbf{Q} . The decomposed diagonal matrix is represented by \mathbf{U} and \mathbf{V} respectively, and the singular value is represented by σ

$$\mathbf{Q} = \mathbf{U} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \mathbf{V}^T \tag{11}$$

Finally, the rotation matrix component can be calculated as

$$\mathbf{R} = \mathbf{V} \times \mathbf{U} \tag{12}$$

And, the translation matrix component can be obtained

$$t = {}^w P - \mathbf{R} \cdot {}^A P'_i \tag{13}$$

After extracting the coordinates of the markers and the coordinate wall in the world coordinate frame, the ICP algorithm can be used to solve the coordinates of the marker in the wall, and then the mark can be arranged in the real environment.

Algorithm 1: Data extraction from BIM

Input: β BIM of the building.
Output: Environment map.
 $C \leftarrow \text{FILTERCOMPONENT}(\beta)$;
 $n \leftarrow \text{NUMBERCOMPONENT}(C)$;
for $i = 1$ **to** n **do**
 if C_i is curtain wall **then**
 $w \leftarrow \text{EXTRACTWALL}(C_i)$;
 ω .ADDWALL(w);
 else if c_i is marker **then**
 $a \leftarrow \text{EXTRACTMARKER}(C_i)$;
 α .ADDMARKER(a);
 else if c_i is frame **then**
 $f \leftarrow \text{EXTRACTFRAME}(C_i)$;
 ζ .ADDFRAME(f);
 else
 $f \leftarrow \text{CYLINDRICALENVELOPE}(C_i)$;
 ζ .ADDTRUSSMEMBER(f);
 end
end
M.ADDMAP(ω, α, ζ).

4.3. Data extraction from BIM

According to the International Organization for Standardization (ISO 19650-1:2018), BIM is defined as the utilization of a shared digital representation of a constructed asset to streamline design, construction, and operational processes, thereby providing a dependable foundation for decision-making. In the BIM modeling phase, designers can readily create standardized components, including curtain walls and frames. However, for distinctive elements like ArUco markers, customized component libraries can be developed and seamlessly incorporated into the design workflow. Nevertheless, generating the map for BWCR directly from BIM is not feasible. Instead, an extraction algorithm must be meticulously crafted to retrieve specific data from the 3D building model in BIM, enabling the construction of the curtain wall environment map.

The diagram of our proposed BIM data extraction method is shown in Algorithm 1. First, the various types of environment components are found through a filter. Then, all the components are traversed to extract the corresponding geometric data and ID information according to their internal structure. The sets of curtain walls, ArUco markers, and frames ζ can be obtained, respectively. Note that, other components are enveloped as cylinders through CYLINDRICALENVELOPE() and considered cylinder-type obstacles. Finally, all extracted data will be stored to build the curtain wall environment map M.

5. Resulted path tracking of BWCR

5.1. Global localization

During climbing in a curtain wall scenario, BWCR needs to obtain its own global position and posture information to evaluate whether it deviates from the planned route. Figure 5 illustrates the basic idea of the visual positioning scheme of BWCR based on ArUco markers. The swinging end of the robot moves along the planned single-step path to the next foothold. The onboard monocular camera installed at the suction module perceives the environment. Once the robot detects and obtains the pose of the markers ${}^A_C\mathbf{T}$, where $\{C\}$ and $\{A\}$ are the coordinate frames of the camera and the ArUco marker, we can calculate the global pose of the swinging end $\{E\}$ with the relationship between the coordinate frames,

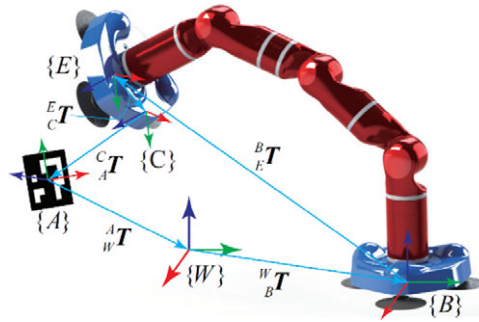


Figure 5. Global visual localization of BWCR.

$${}^W T_E = {}^W T_A {}^A T_C {}^C T_E \tag{14}$$

where ${}^W T_A$ is the transformation matrix from the marker coordinate frame to the world coordinate frame, which can be extracted from the environment map. ${}^C T_E$ denotes the transformation matrix from the swinging end of the robot to the monitoring camera, which can be obtained through eye-in-hand calibration. Thus, obtaining ${}^A T_C$ with an appropriate algorithm is the most critical step of the global localization of BWCR.

5.1.1. Camera pose estimation

To estimate the camera pose, the intrinsic and extrinsic parameters of the camera must be calibrated. The former represents a projective transformation from the 3D coordinates of the camera to the 2D image coordinates, while the latter defines a rigid transformation from the world coordinate system to the camera's coordinate system, i.e., ${}^A T_C$. Many calibration methods have been proposed to estimate the intrinsic parameters [35]. Extrinsic parameter estimation is typically performed after the intrinsic parameter calibration, where extrinsic parameters are determined by mapping 3D space points to their corresponding image points. This process is known as the perspective-n-point (PnP) problem, which can be solved using methods such as Direct Linear Transform, P3P, and EPnP [36], etc. The P3P method, which relies on three sets of corresponding feature points, can yield up to four solutions. Consequently, a fourth point is generally used for verification. Pixel points are first transformed into physical space coordinates, after which the Iterative Closest Point (ICP) method is applied to determine the camera pose.

5.1.2. ArUco marker detection algorithm

The ArUco marker detection process is outlined in Algorithm 2. First, the visual image I is collected by the monitoring camera on the swinging end, and transmitted to the host PC through the wireless network. Then, the operator `RGB2GRAY()` converts the RGB image into a grayscale image. After filtering the image with a Gauss filter `IMAGEFILTER()`, an adaptive threshold method `ADAPTIVESEGMENT()` segments the markers. Next, the contours of squares and convex hulls are detected using an approximation polygon algorithm `APPROXCONTOUR()`. If the contour perimeter exceeds the set threshold, the corresponding marker is eliminated. At this stage, all candidate markers are identified, and the front view of the image is generated using a perspective transformation algorithm `PERSPECTIVETRANSFORM()`. Afterward, the Otsu method `OTSUBINARYZATION()` is used for thresholding. The marker information can be extracted by matching it with the marker dictionary. In certain detection poses, the robot may detect multiple markers, including those behind the glass. To ensure accurate and reliable positioning, the marker closest to the image center is selected to calculate

Algorithm 2: ArUco marker detection

Input: I Image.
 M Environment map.
Output: ${}^W_B T$ Environment map.
 $I_g \leftarrow RGB2GRAY(I)$;
 $I_b \leftarrow IMAGEFILTER(I_g)$;
 $I_s \leftarrow ADAPTIVESEGMENT(I_b)$;
 $C_s \leftarrow APPROXCONTOUR(I_s)$;
 $\alpha_s \leftarrow FILTERMARKER(C_s)$;
 $\Phi_p \leftarrow PERSPECTIVETRANSFORM(I_s, \alpha_s)$;
 $\Phi_o \leftarrow OTSUBINARIZATION(\Phi_p)$;
 $S_{id} \leftarrow IDENTIFYMARKER(\Phi_o)$;
 $N_m \leftarrow NUMBER(S_{id})$
if $N_m > 1$ **then**
 $P(x_c, y_c) \leftarrow (width/2, height/2)$; // Image center coordinates
 for $i = 1$ **to** N_m **do**
 $(P(x_m, y_m)_i, ID_i) \leftarrow CENTERMARKER(S_{id}, i)$;
 //Marker center coordinates and its ID
 $(D_i, ID_i) \leftarrow DISTANCE(P(x_c, y_c), P(x_m, y_m)_i, ID_i)$;
 end
 $ID_m \leftarrow MINIMUMDIS(D, ID)$;
else
 $ID_m \leftarrow EXTRACTID(S_{id})$;
end
 ${}^W_B T \leftarrow POSEESTIMATION(M, \alpha_s, ID_m)$.

the global localization of the swinging end of the BWCR using the proposed pose estimation algorithm. Additionally, the system can leverage the pre-extracted BIM-based environment map to validate the detected markers. By comparing the detected markers with their expected positions on the map, the system ensures that only relevant markers are utilized for localization.

As the robot's tasks within wall environments are often dynamic, it cannot be ensured that markers remain within the camera's field of view throughout traversal. The marker detection actions can be intentionally increased during the robot movement. Even if the robot's pose deviates slightly from the planned path, executing detection movements significantly enhances the probability of locating markers for self-localization.

5.2. Climbing path correction of BWCR

The BWCR traverses the curtain wall environment following the planned path. Deviations in the initial pose and climbing motion may cause the BWCR to deviate from the planned path during its movement. As the number of climbing steps increases, foothold deviations accumulate, potentially causing the robot's landing point to exceed the wall's range, leading to a fall hazard. Thus, timely monitoring and correction of the robot's motion deviations are required to ensure the BWCR continuously follows the planned path.

5.2.1. Climbing path correction model

The model of climbing path correction is depicted in Figure 6. A foothold received from the planner can be expressed in the base coordinate frame,

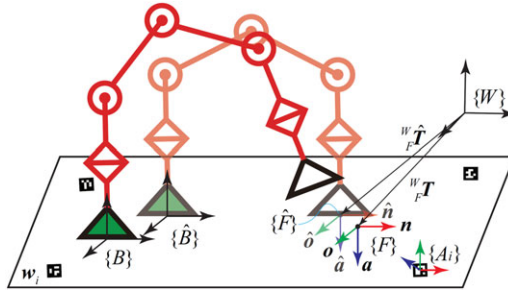


Figure 6. Model of climbing path correction.

$${}^w_F \hat{T} = \begin{bmatrix} \hat{R} & {}^w \hat{P}_f \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \hat{n} & \hat{o} & \hat{a} & {}^w \hat{P}_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

where the sign $\hat{\cdot}$ denotes the ideal foothold. To make the actual foothold ${}^w_F T$ as close to the ideal as possible, we have to minimize the distance of the attaching orientation and the attaching position, respectively.

Since the walls in this paper are plane, according to the geometric constraints of the robot’s adsorption, the attaching orientation of the robot is fixed at any position in the plane. Therefore, the corrected attaching orientation is consistent with the ideal,

$${}^w_F R = [\hat{n} \quad \hat{o} \quad \hat{a}] \tag{16}$$

For the position component, the attaching point must be on the wall and closest to the idea,

$$\underset{x,y,z}{\operatorname{argmin}} f(s, y, z) = \left\| {}^w \hat{P}_f - {}^w P_f \right\| \tag{17}$$

$$\text{s.t.} \begin{cases} {}^w P_f \in \omega_i \\ \exists \mathbf{q} = IK({}^B_F T), \mathbf{q} \in [\underline{\mathbf{q}}, \bar{\mathbf{q}}] \\ \mathfrak{R}(\mathbf{q}) \cap E_w = \emptyset \end{cases}$$

where denotes the joint angle vector of BWCR and $[\underline{\mathbf{q}}, \bar{\mathbf{q}}]$ are the joint limits, $IK(\cdot)$ represents the robot inverse kinematics, while \mathfrak{R} denotes the robot geometry calculated by its kinematics. The constraint $\mathfrak{R}(\mathbf{q}) \cap E_w = \emptyset$ means that the robot does not collide with the environment. Notably, ${}^B_F T$ signifies the actual foothold in the base coordinate frame and is obtained

$${}^B_F T = ({}^W T_E {}^E T_B)^{-1} {}^W T \tag{18}$$

where ${}^W T_E$ can be calculated by Equation (14), while ${}^B_E T$ can be read from the robot controller directly.

5.2.2. Deviation correction method

Figure 7 illustrates the proposed climbing path correction method. BWCR utilizes the onboard monitoring camera to detect the ArUco marker for global localization during each climbing step. If the marker cannot be detected, the robot initiates dead reckoning using forward kinematics for auxiliary positioning. Upon detecting a climbing motion deviation, the robot calculates the corrected foothold based on its current motion state and re-plans the swinging end’s trajectory to the new target. As vacuum adsorption necessitates precise alignment of all suction module suckers with the target surface to create an airtight chamber for vacuum generation, an autonomous pose detection and alignment method utilizing ultrasonic sensors is employed to ensure accurate wall adherence by the swinging end [37].

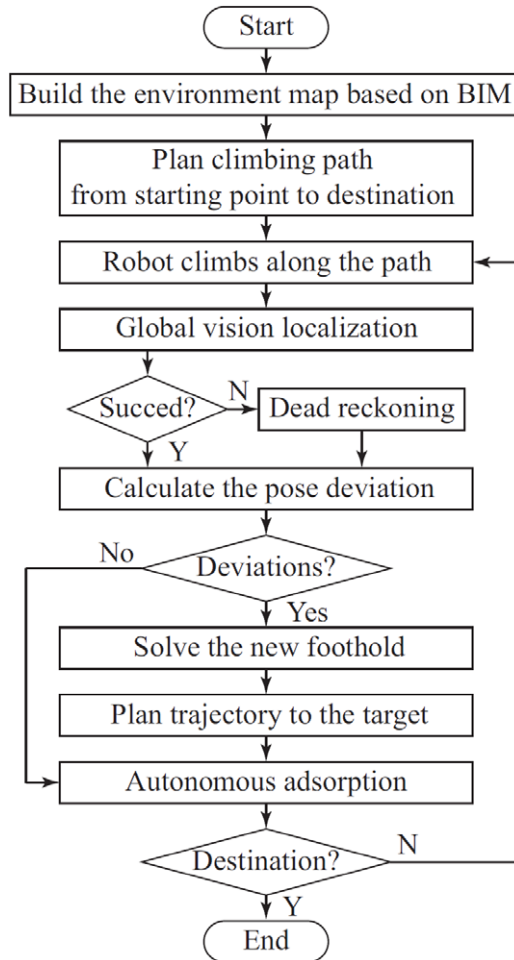


Figure 7. The proposed climbing path correction method.

Several strategies are defined to enable the robot to perform long-duration climbing navigation:

- Visual-based localization is the primary, supplemented by odometer positioning: The robot utilizes ArUco markers for visual localization, with the odometer activated simultaneously. If visual detection fails, the system automatically switches to dead reckoning to estimate the robot's global pose.
- Reliable adsorption must be ensured: Negative pressure adsorption imposes strict geometric constraints between the suction cups and the target surface. Therefore, the suction module's reliable adhesion must be maintained during climbing, particularly when transitioning between walls.
- Deviation correction within one step is the primary, supplemented by multi-step correction: If the motion deviation becomes excessive due to prolonged reliance on dead reckoning, the robot cannot correct the deviation in a single step and must invoke the path planner to compute multiple footholds for correction.

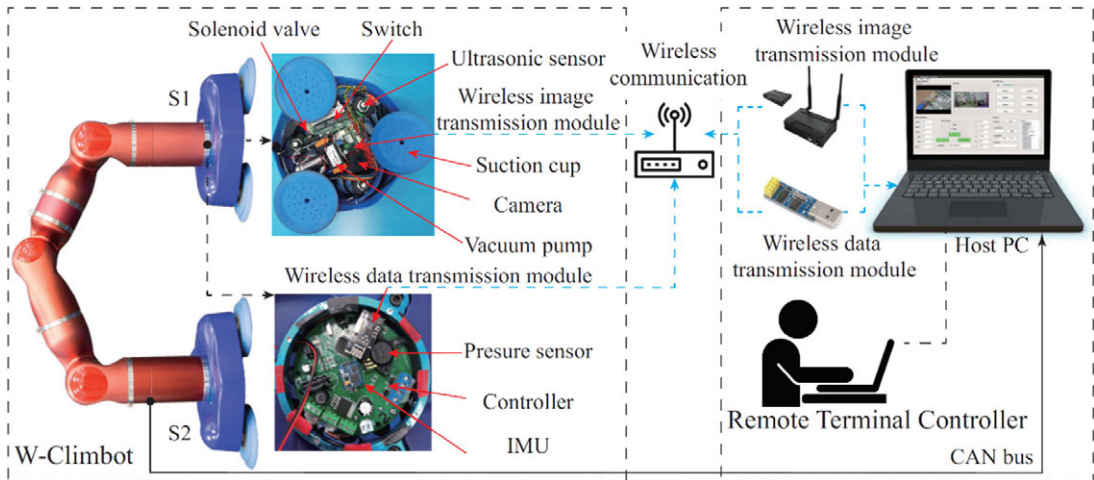


Figure 8. Architecture of the robotic control system.

6. Climbing robot system development

6.1. Robot system

The proposed algorithms were applied to localization and navigation within the self-developed W-Climbot, a modular biped climbing robot consisting of five joint modules and two suction modules [5, 6]. Its configuration is depicted in Figure 1. A host personal computer (PC) serves as the robot controller, communicating with the distributed servo drivers of joint modules via the CAN bus, as illustrated in Figure 8. The suction module uses STM32F103 as the primary control chip to collect the sensor data, including pressure sensor, ultrasonic sensor, IMU, and camera sensors, and to transmit control signals for the vacuum pump and solenoid valve via wireless data and image transmission modules.

The BIM-based environment mapping method, robot localization algorithm, and climbing path correction method were executed directly on the host PC. A hierarchical motion planning method was implemented to generate the climbing path, connecting the starting point and the destination within the wall environment. The single-step motion planner was deployed to re-plan the collision-free movement swinging the robot suction module from the current pose to a pre-suction point, which was generated by offsetting the corrected foothold in the wall's normal direction by a safe distance. The robot then detected the wall using the sensing system and aligned the suction module with the target wall.

6.2. Image wireless transmission and decoding

To enhance the climbing robot's flexibility and minimize the impact of wired connections on its movement, a wireless high-definition image transmission system was developed. The hardware system primarily includes monitoring cameras, switches, and wireless image transmission modules. Monitoring cameras mounted on the two suction modules capture images and encode them into video streams compliant with the Real-Time Streaming Protocol (RTSP). These streams are transmitted via the Internet Protocol network to a wireless image transmission transmitter, where they are modulated into data packets using orthogonal frequency division multiplexing (OFDM). The packets are then sent to a ground-based wireless image transmission receiver. After OFDM demodulation, the receiver restores the data packets into a video stream with the RTSP protocol and transmits it to the computer. On the remote terminal controller of the W-Climbot, OpenCV image processing operators convert the video frames from YUV420 encoding to RGB format. These RGB frames are then utilized for video playback, image analysis, and robot localization.

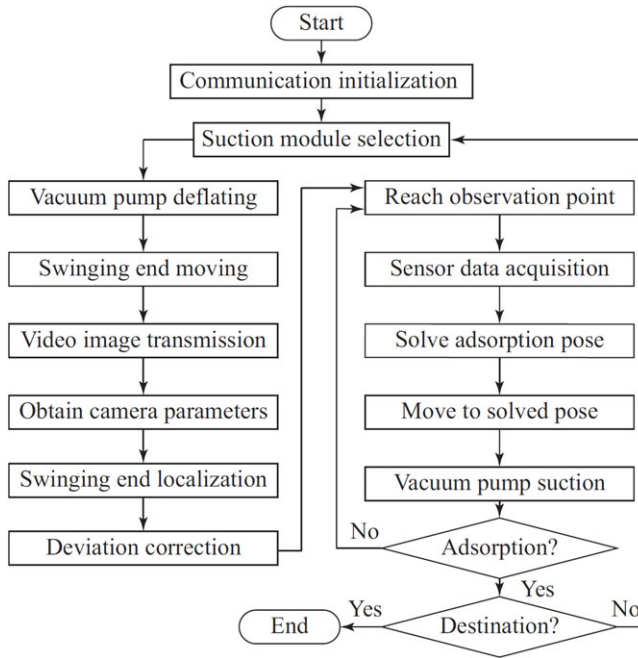


Figure 9. *The suction module control method.*

6.3. Suction module control system

The W-Climbot control system on the ground side was developed using C++ and Python on the Linux platform, with the user interface (UI) designed using QT. The suction module control system integrates functions such as autonomous adsorption, video stream decoding, remote video monitoring, environmental structured map import, and status display. The monitoring video is displayed in real-time on the UI and can be switched based on the active adsorption module.

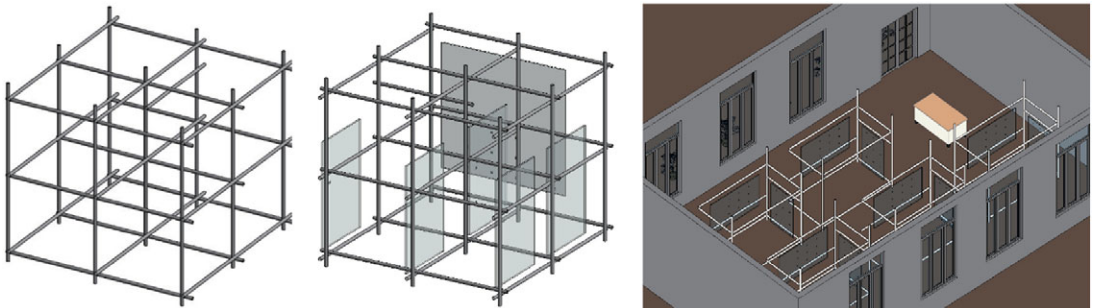
The control process of the suction module is shown in Figure 9. Initially, the robot control system executes initialization tasks, establishes communication with the suction modules, and loads the ArUco marker database. The appropriate suction module is then selected based on the generated climbing path. The alternate module serves as the swinging end, responsible for releasing pressure, disengaging from the wall surface, and transitioning to the next foothold. The UI plays the monitoring video captured by the camera in the corresponding suction module, while simultaneously loading the intrinsic calibration parameters and hand-eye calibration matrix of the camera. Upon detecting the ArUco marker, the swinging end autonomously localizes itself. After motion deviation correction, the swinging end reaches an intermediate observation point near each foothold, maintaining a safe distance from the curtain wall. The control system then collects data from various sensors in the suction module to compute deviations in autonomous adsorption, including the distance and angle between the suction module and the wall. Finally, the swinging end fine-tunes the angle deviation and moves parallel to the wall until the suction cup securely adheres to the surface. If the set pressure is not achieved, the suction module returns to the observation point and repeats the adsorption operation. The robot will continue transforming the suction end and performing the adsorption process until it reaches the destination.

7. Experiments

To verify the proposed localization and navigation approach and analysis, a series of experiments were conducted and presented in this section.

Table I. Results of data extraction based on BIM.

| Testing model | Number of frames | Number of markers | Number of glasses | Extraction rate | Time (ms) |
|---------------|------------------|-------------------|-------------------|-----------------|-----------|
| 1 | 18 | 0 | 0 | 100% | 7.09 |
| 2 | 28 | 0 | 0 | 100% | 8.46 |
| 3 | 252 | 0 | 0 | 100% | 33.43 |
| 4 | 28 | 1 | 1 | 100% | 47.94 |
| 5 | 1680 | 0 | 0 | 100% | 159.01 |
| 6 | 28 | 6 | 6 | 100% | 188.40 |
| 7 | 28 | 8 | 5 | 100% | 239.80 |
| 8 | 28 | 17 | 6 | 100% | 427.31 |
| 9 | 84 | 18 | 18 | 100% | 628.29 |
| 10 | 70 | 90 | 10 | 100% | 2200.46 |
| 11 | 0 | 100 | 100 | 100% | 4275.05 |

**Figure 10.** Some testing scenarios for environment element extraction.

7.1. Map extraction from BIM

In this paper, Autodesk Revit, a BIM authoring tool, was used as the 3D modeling and secondary development platform to develop a Revit plug-in for automatically extracting the map of the curtain wall environment.

A total of 11 Revit models, containing architectural elements such as frames, ArUco markers, and glasses, were selected for testing, with some examples shown in Figure 10. The ArUco markers used in this study are black-bordered squares with an internal binary matrix and a unique ID, each measuring 40 mm × 40 mm. These markers were positioned arbitrarily on the glass surfaces.

Table I presents the results of the data extraction. As shown, the extraction rates for frames, markers, and glass in all models are 100%. The time required for extraction increases progressively with the number of components. For instance, information extraction takes only 159.01 ms for the frame environment, which consists of 1680 elements. However, when the number of glass walls and markers reaches 100, the extraction time increases to 4275.05 ms. From the 10th data group, it is evident that the extraction time is primarily spent on the glass walls. Reducing the number of glass walls significantly decreases the time consumption. This is because the storage method for glass walls in Revit is complex. Moreover, a glass wall is essentially a cuboid made up of six surfaces. During extraction, the two largest adsorbable walls are selected first, followed by the selection of the side with the ArUco markers as the attachment surface.

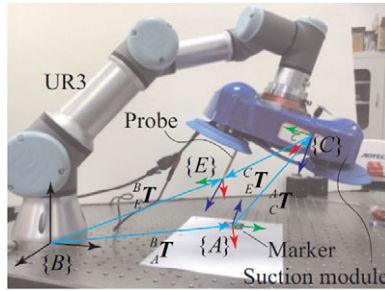


Figure 11. Experimental setup used to evaluate the location accurately.

7.2. Evaluation of positioning accuracy of BWCR

To verify the positioning accuracy and stability of global visual localization using ArUco markers, two sets of evaluation experiments were conducted, as shown in Figure 11. The UR3 robot, chosen as the experimental platform due to its superior motion precision and stability compared to W-Climbot, ensures precise and reliable results. A suction module, integrated with a manufactured probe, was mounted on the end of the UR3 robot to observe the ArUco marker placed on the workbench. After moving the robot’s Tool Center Point (TCP) to the end of the probe, the transformation matrix ${}^A T_C$ between the camera and the new TCP was obtained via hand-eye calibration. By touching the four corners of the ArUco marker with the probe tip and reading its coordinates from the robot controller, the pose of the marker in the robot’s base coordinates frame ${}^B T_A$ was determined. The suction module then moved to capture the relative pose of the camera using the proposed visual localization method. The positioning pose of the TCP was calculated by

$${}^B T_E = {}^B T_A {}^A T_C {}^C T_E \tag{19}$$

the positioning accuracy was evaluated by comparing it with the robot TCP pose ${}^B T_E$ read from the robot controller directly,

$$Err = {}^B T_E - {}^B T_E \tag{20}$$

To assess the stability of the positioning method, the suction module was placed in the same pose for multiple observations to examine the fluctuation in the localization data. The camera plane, positioned parallel to the marker, was placed at distances of 120 mm, 170 mm, and 220 mm with 10 repeated measurements for each. The results are presented in Table II. On average, the positional error of the TCP was within 2.2 mm, while the average orientation error was within 2 degrees. As the distance between the camera and the marker increases, the positioning error also increased. The maximum errors for both position and orientation occurred at the 220 mm, with values of 2.88 mm and 3.14 degrees, respectively. This increase in error is attributed to the decrease in image quality as the distance increases, which reduces positioning accuracy.

To evaluate the positioning accuracy, the suction module was placed at different positions within the UR3 workspace for global visual localization, and the positioning error distribution was analyzed. During the experiment, the robot TCP was moved to the geometry center of the suction module’s bottom plane. The center point of the marker was considered the center of the sphere, with the distance between the robot TCP and the marker serving as the sphere’s radius. Observation points were selected along the spherical surfaces to obtain positioning data, as illustrated in Figure 12. In this experiment, 22 groups of data were collected along the x-axis and y-axis on two spherical surfaces with radii of 120 mm and 170 mm, respectively, and the corresponding positioning errors were recorded, as shown in Figure 13. Overall, the positioning accuracy was minimally affected by spatial angle, with the position error at all observed points remaining below 3 mm, and the angle error within 3 degrees.

Table II. Results of localization stability test.

| Distance (mm) | Item | X (mm) | Y (mm) | Z (mm) | Rx (degree) | Ry (degree) | Rz (degree) |
|---------------|---------|--------|--------|--------|-------------|-------------|-------------|
| 120 | Maximum | 1.52 | -0.20 | 0.23 | -1.33 | 1.85 | -0.17 |
| | Mean | 1.49 | -0.20 | 0.16 | -1.30 | 1.81 | -0.16 |
| 170 | Maximum | 2.67 | 2.49 | -2.02 | -1.81 | 0.24 | -0.23 |
| | Mean | 1.28 | 2.11 | -1.50 | -1.53 | 0.24 | -0.09 |
| 220 | Maximum | 2.88 | 1.08 | -1.38 | -3.14 | 2.22 | -0.40 |
| | Mean | 1.40 | 0.72 | -0.62 | -1.71 | 1.90 | -0.07 |

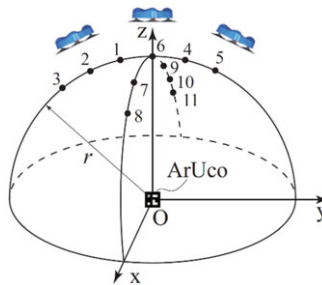


Figure 12. The accuracy evaluation of spatial multi-pose positioning.

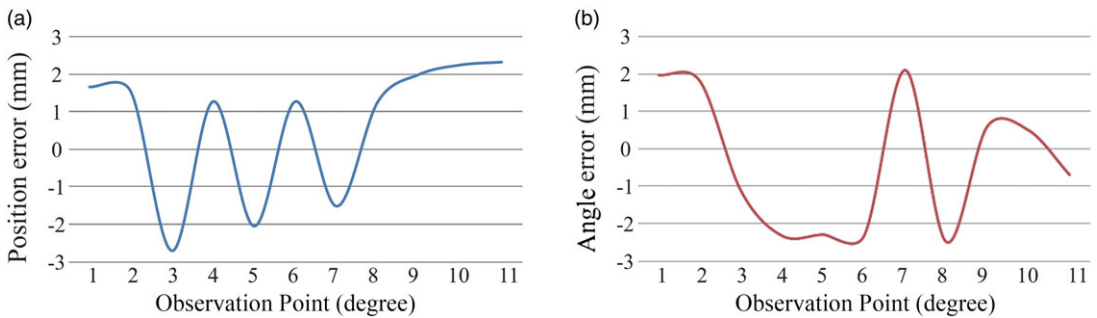


Figure 13. Evaluation results of positioning accuracy with a radius of 120 mm. (a) Position error. (b) Angle error.

7.3. Multi-step climbing

To evaluate the feasibility and effectiveness of the visual localization and path correction method based on ArUco markers and BIM, this subsection performed a comprehensive climbing comparison experiment involving W-Climbot climbing multiple steps on different wall surfaces.

Figure 14 depicts the experimental scenario. A 3D frame environment was constructed using poles of varying lengths connected by fasteners. Glass walls were affixed to the frame via surrounding connecting claws, and ArUco markers were applied to each wall surface in a grid pattern. The experimental environment was positioned within the working range (8 m × 7 m × 3 m) of the OptiTrack motion capture system, which offered a tracking accuracy of 0.1 mm. The motion capture system monitored the robot’s motion trajectory by tracking markers attached to both ends of the W-Climbot. Two sets of climbing experiments were conducted: one utilizing the proposed navigation scheme with visual localization and

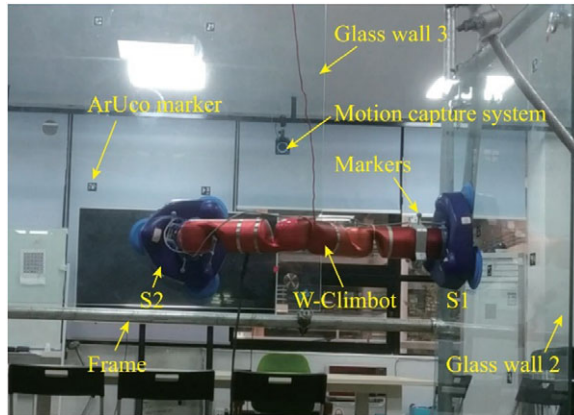


Figure 14. Multi-step climbing environment scenario.

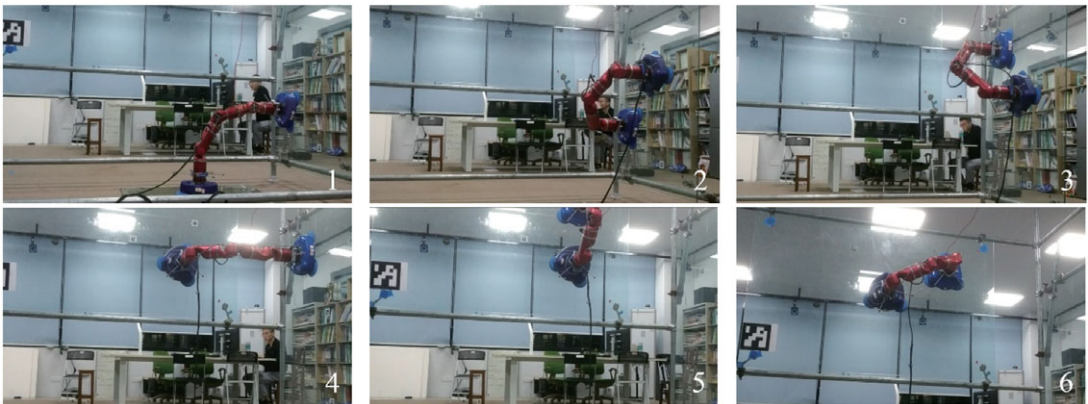


Figure 15. W-Climbot climbed multiple steps consecutively in a wall scenario.

path correction, and the other serving as a comparative test without localization. Both experiments followed the same climbing path, allowing verification of the scheme's effectiveness by comparing the actual trajectory with the theoretically planned trajectory and evaluating the deviations between the two sets of footholds. The robot's climbing cycle is depicted in Figure 15.

Figure 16 shows the experimental results. The deviations observed in the footholds of both groups on wall w_1 were attributed to initial placement errors, which were random and did not influence the experimental outcomes. As depicted in Figure 17, the deviation in the third foothold increased significantly, exceeding 120 mm. This was primarily due to the initial error and the transition between walls. In the visual localization experiment, the robot exhibited a larger deviation in the third step, which was attributed to the amplified initial error, indicating a noticeable error accumulation during the multi-step climbing process of BWCR. After incorporating visual localization, the robot's foothold error decreased markedly, with a deviation of 34 mm in the fourth step compared to 82.5 mm in the experiment without visual localization. For the final foothold in the multi-step climbing sequence, the deviation without visual localization exceeded 170 mm, significantly diverging from the theoretical trajectory reference. In contrast, the foothold deviation in the visual localization experiment remained around 50 mm and demonstrated a trend toward stabilization.

In summary, the robot's foothold error is significantly larger in the absence of visual localization and path correction, with the error accumulation effect becoming increasingly pronounced as the number of climbing steps increases. Conversely, the robot employing visual localization maintains a single-step

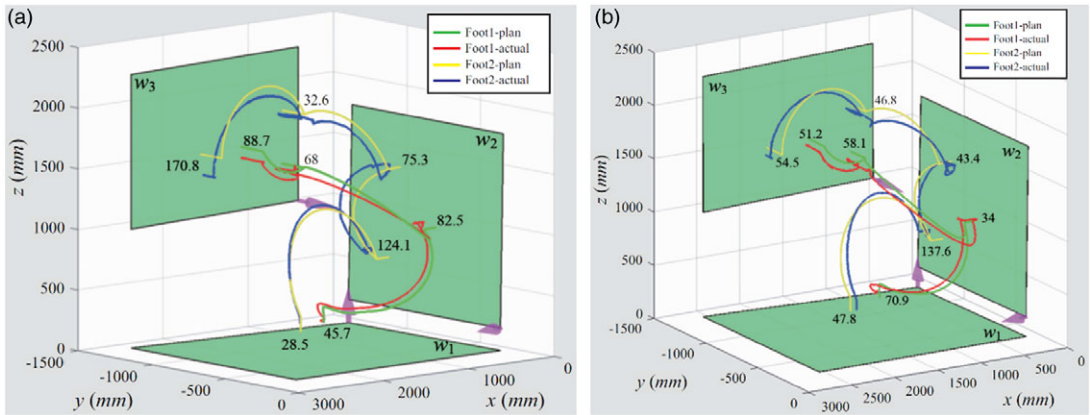


Figure 16. Comparison of the actual and the planned climbing motion. (a) Without visual localization and path correction. (b) With visual localization and path correction.

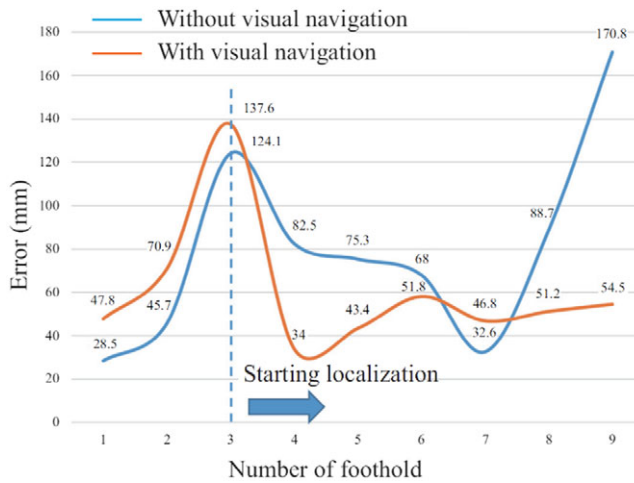


Figure 17. Foothold errors during multi-step climbing for the two group experiments.

foothold error within 58.1 mm. Furthermore, the average deviation across the robot’s final six steps is reduced by 44.39%, while the maximum deviation decreases by 65.98%.

8. Conclusion

BWCRs hold significant potential for executing versatile tasks on curtain wall buildings. Efficient environment mapping and navigation capabilities are essential to enable autonomous climbing in real-world scenarios. This study proposed a novel localization and navigation framework to enhance BWCR autonomy. Geometric information from the building’s BIM was extracted to construct an environment map, which facilitated climbing path planning. Geometric information from the building’s BIM was extracted to construct an environment map, which facilitated climbing path planning. When deviations occurred, path correction strategies were employed to compensate effectively. Experimental results confirmed the feasibility and effectiveness of the proposed framework and algorithms.

Future work will address the optimization of marker placement, as the current layout occasionally results in unobservable or redundant markers during climbing. Additionally, the integration of onboard

sensors such as LiDAR and depth cameras, along with advanced object recognition and simultaneous localization and mapping (SLAM) algorithms, will be explored to enable BWCRs to dynamically map their surroundings and achieve real-time localization.

Supplementary material. To view supplementary material for this article, please visit <https://doi.org/10.1017/S0263574724002170>

Author contributions. Shichao Gu organized the study, developed the methodology, and conducted the experiments. Ziyang Fu and Weinan Chen collaborated on the design of the localization and navigation algorithms and contributed to the experimental work. Hongmin Wu and Xuefeng Zhou provided research supervision and contributed to the analysis and interpretation of results. Yisheng Guan and Haifei Zhu conceptualized the methodology, supervised the research, and prepared the final manuscript.

Financial support. This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation (Grant No.2022A1515110750 and 2022A1515240013), the China Postdoctoral Science Foundation (Grant No.2023M730736), National Natural Science Foundation of China (Grant No. 62203126), GDAS' Project of Science and Technology Development (Grant No. 2022GDASZH-2022010108), and Talent Introduction Project of Guangdong Polytechnic Normal University (Grant No. 2023SDKYA013).

Competing interests. The authors declare no competing interests exist.

Ethical standards. None.

References

- [1] Y. Fang, S. Wang, Q. Bi, D. Cui and C. Yan, "Design and technical development of wall-climbing robots: A review," *J. Bionic Eng.* **19**(4), 1–25 (2022).
- [2] C. Balaguer, A. Gimenez and C. Abderrahim, "ROMA robots for inspection of steel-based infrastructures," *Ind. Robot Int. J.* **29**(3), 246–251 (2002).
- [3] C. Balaguer, A. Gimenez, A. J. Huete, A. M. Sabatini, M. Topping and G. Bolmsjo, "The mats robot: Service climbing robot for personal assistance," *IEEE Robot. Autom. Mag.* **13**(1), 51–58 (2006).
- [4] C.-h. J. Yang, G. Paul, P. Ward and D. Liu, "A Path Planning Approach Via Task-objective Pose Selection with Application to an Inchworm-inspired Climbing Robot," *In: 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE (2016), pp. 401–406.
- [5] Y. Guan, H. Zhu, W. Wu, X. Zhou, L. Jiang, C. Cai, L. Zhang and H. Zhang, "A modular biped wall-climbing robot with high mobility and manipulating function," *IEEE/ASME Trans. Mech.* **18**(6), 1787–1798 (2012).
- [6] H. Zhu, J. Lu, S. Gu, S. Wei and Y. Guan, "Planning three-dimensional collision-free optimized climbing path for biped wall-climbing robots," *IEEE/ASME Trans. Mech.* **26**(5), 2712–2723 (2020).
- [7] V. G. Loc, M. K. Ig, D. T. Tran, Y. K. Song, H. M. Kim, P. M. Hyung and H. R. Choi, "Sensing and Control of Quadruped Walking and Climbing Robot Over Complex Environment," *In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE (2008), pp. 3884–3889.
- [8] B. Howarth, J. Katupitiya, J. Guivant and A. Szewc. "Novel Robotic 3D Surface Mapping Using Range and Vision Fusion," *In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE (2010), 1539–1544.
- [9] G. Paul, P. Quin, A. W. K. To and D. Liu, "A Sliding Window Approach to Exploration for 3D Map Building Using a Biologically Inspired Bridge Inspection Robot," *In: 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, IEEE (2015), pp.1097–1102.
- [10] W. Chen, S. Gu, L. Zhu, H. Zhang, H. Zhu and Y. Guan, "Representation of truss-style structures for autonomous climbing of biped pole-climbing robots," *Robot. Auton. Syst.* **101**, 126–137 (2018).
- [11] G. Cai, W. Lin, S. Wei, S. Gu, H. Zhu and Y. Guan, "Representation of 3D Structure for Path Planning with Biped Wall-climbing Robots," *In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE (2019), pp. 2250–2255.
- [12] J. Xiao and Z. Zhu. *Cooperative Wall-climbing Robots in 3D Environments for Surveillance and Target Tracking* (CITY COLL OF THE CITY UNIV OF NEW YORK, (2009).Tech. Rep.
- [13] M. Tavakoli, G. Cabrita, R. Faria, L. Marques and A. T. de Almeida, "Cooperative multi-agent mapping of three-dimensional structures for pipeline inspection applications," *Int. J. Robot. Res.* **31**(12), 1489–1503 (2012).
- [14] J. Park, Y. K. Cho and D. Martinez, "A BIM and UWB integrated mobile robot navigation system for indoor position tracking applications," *J. Constr. Eng. Project Manage.* **6**(2), 30–39 (2016).
- [15] J. Liu, J. Luo, J. Hou, D. Wen, G. Feng and X. Zhang, "A bim based hybrid 3D indoor map model for indoor positioning and navigation," *ISPRS Int. J. Geo-Inf.* **9**(12), 747 (2020).
- [16] Q. Jiang, Y. Liu, Y. Yan, X. Mao, H. Xu and X. Jiang, "BIM-based 3-D multimodal reconstruction for substation equipment inspection images," *IEEE Trans. Instrum. Meas.* **73**, 1–14 (2024).

- [17] Z. Chen, K. Chen, C. Song, X. Zhang, J. C. Cheng and D. Li, “Global path planning based on bim and physics engine for ugv’s in indoor environments,” *Automat. Constr.* **139**, 104263 (2022).
- [18] O. Acar and C. F. YaŞar, “Autonomous climbing robot for tank inspection,” *Procedia Comput. Sci.* **158**, 376–381 (2019).
- [19] L. Shao, S. Yang, H. Liu and J. Li. “Research on Location Method of Pipe Climbing Robot based on Gyroscope,” *In: 2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, IEEE (2018), pp. 238–242.
- [20] X. Yu, W. Zheng and L. Ou, “CPR-SLAM: RGB-D SLAM in dynamic environment using sub-point cloud correlations,” *Robotica* **42**(7), 2367–2387 (2024).
- [21] T. I. Liao, S. S. Chen, C. C. Lien, H. C. Lin, M. F. Lu and W. P. Chen, “Development of a high- endurance cleaning robot with scanning-based path planning and path correction,” *Microsyst. Technol.* **27**(4), 1061–1074 (2021). doi: [10.1007/s00542-018-4048-2](https://doi.org/10.1007/s00542-018-4048-2).
- [22] Q. Lin, Z. An and L. Yang. “Rebooting Ultrasonic Positioning Systems for Ultrasound-incapable Smart devices,” *In: 25th Annual International Conference on Mobile Computing and Networking*, (2019) pp. 1–16.
- [23] Y. Wu, R. Chen, W. Li, Y. Yu, H. Zhou and K. Yan, “Indoor positioning based on walking-surveyed wifi fingerprint and corner reference trajectory-geomagnetic database,” *IEEE Sens. J.* **21**(17), 18964–18977 (2021).
- [24] B. Jang, H. Kim and J. wook Kim, “Survey of landmark-based indoor positioning technologies,” *Inform. Fusion* **89**(2023), 166–188 (2022).
- [25] J. Yin, D. Luo, F. Yan and Y. Zhuang, “A novel lidar-assisted monocular visual slam framework for mobile robots in outdoor environments,” *IEEE Trans. Instrum. Meas.* **71**, 1–11 (2022).
- [26] P. Quin, G. Paul, A. Alempijevic and D. Liu. “Exploring in 3D with a Climbing Robot: Selecting the Next Best Base Position on Arbitrarily-oriented Surfaces,” *In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2016), pp. 5770–5775.
- [27] S. Gu, H. Zhu, H. Li, Y. Guan and H. Zhang, “Optimal collision-free grip planning for biped climbing robots in complex truss environment,” *Appl. Sci.* **8**(12), 2533 (2018).
- [28] B. Y. Shih, C.-Y. Chen and W. C. Chou, “Retracted: Obstacle avoidance using a path correction method for autonomous control of a biped intelligent robot,” *J. Vib. Control* **17**(10), 1567–1573 (2011).
- [29] M. Tavakoli, L. Marques and A. T. de Almeida, “A low-cost approach for self-calibration of climbing robots,” *Robotica* **29**(1), 23–34 (2011).
- [30] L. Jiajun, S. Zhenguo and Q. C. W. Zhang, “Error correction algorithm for manipulator of wall climbing robot with both ends having magnetic ad- sorption,” *J. Tsinghua Univ. (Sci. Technol.)* **54**(2), 185–190 (2015).
- [31] W. Song, Z. Wang, T. Wang, D. Ji and S. Zhu, “A path tracking method of a wall-climbing robot towards autonomous inspection of steel box girder,” *Machines* **10**(4), 256 (2022).
- [32] F. J. Romero-Ramirez, R. Munoz-Salinas and R. M. Carnicer, “Speeded up detection of squared fiducial markers,” *Image Vision Comput.* **76**, 38–47 (2018).
- [33] P. J. Besl and N. D. McKay, “Method for Registration of 3-d Shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. **1611**. Spie, United States (1992) pp. 586–606.
- [34] F. Pomerleau, F. Colas and R. Siegwart, “A review of point cloud registration algorithms for mobile robotics,” *Found. Trends® Robot* **4**(1), 1–104 (2015).
- [35] J. Zhang, H. Yu, H. Deng, Z. Chai, M. Ma and X. Zhong, “A robust and rapid camera calibration method by one captured image,” *IEEE Trans. Instrum. Meas.* **68**(10), 4112–4121 (2018).
- [36] X. X. Lu, “A Review of Solutions for Perspective-n-Point Problem in Camera Pose Estimation,” *In: Journal of Physics: Conference Series*. vol. **1087** (IOP Publishing, 2018) pp. 052009.
- [37] H. Zhu, Y. Guan, W. Wu, L. Zhang, X. Zhou and H. Zhang, “Autonomous pose detection and alignment of suction modules of a biped wall-climbing robot,” *IEEE/ASME Trans. Mech.* **20**(2), 653–662 (2014).