*machines*

MDPI

*Article*

# Window Shape Estimation for Glass Façade-Cleaning Robot

**Takuma Nemoto [1,*], Shunsuke Nansai [2], Shohei Iizuka [3], Masami Iwase [1,3] and Hiroshi Itoh [2]**

[1] Department of Robotics and Mechatronics, School of Science and Technology for Future Life, Tokyo Denki University, 5 Senju Asahi-cho, Adachi-ku, Tokyo 120-8551, Japan
[2] Department of Advanced Machinery Engineering, School of Engineering, Tokyo Denki University, 5 Senju Asahi-cho, Adachi-ku, Tokyo 120-8551, Japan
[3] Robotics and Mechatronics, Graduate School of Science and Technology for Future Life, Tokyo Denki University, 5 Senju Asahi-cho, Adachi-ku, Tokyo 120-8551, Japan
* Correspondence: nemoto@ctrl.fr.dendai.ac.jp

**Abstract:** This paper presents an approach to the estimation of a window shape for increasing the adaptability of glass façade-cleaning robots to different buildings. For this approach, a window scanning robot equipped with a 2D laser range scanner installed perpendicularly to a window surface is developed for the testbed, and a method for the window shape estimation is proposed, which consists of the robot's pose estimation with an extended Kalman filter (EKF) and the loop closure based on the robot's pose estimated. The effectiveness of the proposed approach is demonstrated through an experiment that is carried out on a window placed on a floor. The experimental results show that the window scanning robot can acquire a window shape, moving on a window surface, and the proposed approach is effective in increasing the accuracy of the window shape estimation.

**Keywords:** façade-cleaning robot; window shape estimation; extended Kalman filter (EKF); loop closure

check for updates

**Citation:** Nemoto, T.; Nansai, S.; Iizuka, S.; Iwase, M.; Itoh, H. Window Shape Estimation for Glass Façade-Cleaning Robot. *Machines* **2023**, *11*, 175. https://doi.org/10.3390/machines11020175

Academic Editors: Peter Odry, Akos Odry and Jan Awrejcewicz

Received: 27 December 2022
Revised: 22 January 2023
Accepted: 24 January 2023
Published: 27 January 2023

**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Increasing skyscrapers built with cutting-edge construction technologies and processes demand the involvement of robots in their maintenance. Such modern skyscrapers often have glass façades, which are maintained and cleaned by labor using gondolas and tethers in high places. The maintenance and cleaning of skyscrapers' glass façades by labor thus have the potential for causing serious accidents. The out-of-control gondola due to the strong wind at the Shanghai World Financial Center [1] and the gondola suspended at a height of 240 meters at the World Trade Center in New York City [2] are examples of accidents. The application of robots in the maintenance and cleaning of skyscrapers' glass façades can minimize the risk of such accidents.

Many research works on façade-cleaning robots have been reported [3,4]. The façade-cleaning robots are categorized into two broad types based on the movement mechanism: robots utilizing equipment installed on buildings, such as a crane, winch, gondola, and guide rails, and ones with no need of such equipment.

Concerning robots utilizing the equipment installed on buildings, Elkmann et al. developed an automatic façade-cleaning robot, SIRIUSc, for the Fraunhofer headquarters building in Munich, Germany [5–7]. SIRIUSc can move on and clean a building façade, using two pairs of linear modules, called the advanced sliding frame mechanism, and a rooftop gantry. S. Lee et al. suggested a built-in guide-type multi-robot concept and proposed a motion planning algorithm for façade glass cleaning with the robots [8]. Moon and Shin et al. developed a building façade maintenance robot (BFMR) based on built-in guide rails and its cleaning tool [9–11]. The BFMR consists of horizontal and vertical units and moves horizontally and vertically along the built-in guide rails. While moving horizontally, the BFMR cleans a façade with the cleaning tool which sprays and suctions water. Y. S. Lee et al. proposed an integrated control system for a built-in guided robot, which is divided

into three stages: the preparation stage, cleaning stage, and return stage [12]. C. Lee et al. suggested a three-modular obstacle-climbing robot for façade cleaning, which is composed of a main platform, three modular climbing units, and a winch mechanism set on the top of a building [13]. The robot can clean a building façade with a window-cleaning system installed on the middle module, overcoming obstacles with the climbing units and the winch mechanism. Yoo et al. introduced an unmanned façade-cleaning robot equipped on a gondola, which consists of a two-degrees-of-freedom (DOF) robotic manipulator and a cleaning device [14]. The performance of the robot was tested on the 63 Building in the Republic of Korea. For the gondola-type cleaning robot, Hong et al. designed a cleaning module, applying a passive linkage suspension mechanism and tri-star wheels to overcome step-shaped obstacles [15], and Park et al. designed a 3-DOF manipulator for a cleaning module to compensate for the horizontal disturbance of a gondola [16]. Furthermore, Chae et al. proposed the improved design of the gondola-type cleaning robot, which includes the modularized robot design, a passive obstacle-overcoming mechanism with tri-star wheels and a compliant manipulator, and a position sensing device for the measurement and compensation of the lateral disturbance [17]. Although the performance of the aforementioned robots was demonstrated through experiments, they have the limitation of operational buildings because of their movement mechanism requiring the installed equipment.

For robots with no need of the installed equipment, Zhu, Sun, and Tso developed a climbing robot for glass-wall cleaning and presented motion planning and visual sensing methods for the climbing robot [18–20]. The robot can adhere to a window with suction cups and move with a translational mechanism, on which the motion planning and visual sensing enable the robot to track a desired path and measure its position and orientation relative to a window frame and the locations of dirty spots. Zhang et al. proposed a series of autonomous pneumatic climbing robots named sky cleaners for glass-wall cleaning [21]. One of the climbing robots, sky cleaner 3, was built for the glass walls of the Shanghai science and technology museum [22–25]. The sky cleaner 3 can adhere to glass walls with vacuum suckers and move with cylinders, for which the authors designed an intelligent control system based on a programmable logic controller (PLC) and proposed a method of the segment and variable bang-bang controller. Furthermore, the authors proposed three nonlinear control strategies, the fuzzy PID, segmental proportional control, and segmental variable bang-bang controller, for the sky cleaner 1, 2, and 3 [26]. Zhang et al. also developed a climbing robot for cleaning the spherical surface of the national grand theater in China and designed an intelligent control system based on the CAN bus [27]. Seo and T. Kim et al. designed a wall-climbing robotic platform, ROPE RIDE, for cleaning walls of buildings, and its cleaning unit [28,29]. ROPE RIDE is built on a rope ascender-based locomotion mechanism combined with triangular tracks to climb up walls and overcome obstacles and two propeller thrusters to contact walls. For ROPE RIDE, the authors presented a position-based adaptive impedance control (PAIC) to maintain a constant contact force between a cleaning unit and a wall [30]. Tun et al. developed a glass façade-cleaning robot, vSlider, which has passive suction cups driven by self-locking lead screws to adhere to a glass façade [31]. The robot can perform façade cleaning with the mechanism, reducing the power consumption. Vega-Heredia et al. presented a modular façade-cleaning robot called Mantis and a method of multi-sensor orientation tracking for the robot [32,33]. Mantis can overcome window frames, detecting them with an inductive sensor. Chae et al. designed a novel rope-driven wall-cleaning robot, Edelstro-M2 [34]. Edelstro-M2 can move vertically and horizontally with a dual rope-climbing mechanism and parallel kinematics and can be operated by just fixing two pieces of rope on roof anchors. The robots with no need for the installed equipment can be applied to any building, compared to that utilizing the installed equipment. However, the robots tend to be used to clean the façades of buildings with a conventional appearance.

To improve the adaptability of façade-cleaning robots to façades of various types of building architecture, we have proposed a concept of nested reconfigurable robots for façade cleaning. The concept is aimed at achieving autonomous façade cleaning according

to window shapes, employing multiple modular multilegged robots capable of reconfiguring their morphology based on window shapes and letting the robots cooperate. Based on this concept, Nansai et al. suggested two types of glass façade-cleaning robots [35,36]. One is a modular robot assembled by a linear actuator, and another is a modular biped robot. For the modular biped robot, they proposed a foot location algorithm for glass façade cleaning [37]. In the previous works, however, the approach that the robots obtain environmental information and own states was not considered.

The task of façade-cleaning robots to perceive their surrounding environments and own states, which is of little or no consideration in the related works and our previous ones, is required to work autonomously in unknown environments and hence contributes to increasing their adaptability to façades of various types of building architecture. The task is similar to the simultaneous localization and mapping (SLAM) for autonomous driving [38,39]. In the SLAM for autonomous driving, the positions of a car and a map of its surrounding environment are estimated by observing its surroundings with external sensors, such as light detection and ranging (LiDAR) sensors and cameras, and performing feature matching based on observed data. In the exploration for autonomous façade cleaning, however, it is difficult to observe the environment surrounding a façade-cleaning robot and carry out feature matching as with the SLAM for autonomous driving because a façade-cleaning robot needs to observe window frames as their surroundings that have little or no rise from a window surface and have fewer features. Hence, we need to devise a method to obtain environmental information, especially window shapes, and robot states suitable for façade cleaning, which is a situation having difficulties in observing environments and performing feature matching.

In this paper, based on the concept of nested reconfigurable robots, we discuss a method for façade-cleaning robots to estimate a window shape as an approach to obtaining environmental information on a glass façade of a building in order to increase the adaptability of façade-cleaning robots to façades of various types of building architecture. To this end, we assume the following situations, focusing on the window shape estimation.

- A glass façade-cleaning robot moves on a window surface with a rectangular frame.
- The robot needs to estimate the window shape it is on with its own external sensor.

According to the assumptions, we require the robot to estimate not only a window shape but also its location on the window surface.

To achieve the window shape estimation, we develop a window scanning robot having a 2D laser range scanner installed perpendicularly to a window surface and an estimation method based on the robot's pose. The window scanning robot can obtain its odometry data and measure the relative distance between the robot and a window frame, moving on the window surface. The robot's pose is obtained by applying the extended Kalman filter (EKF) [40,41] with its odometry data, which is adopted due to the simplicity of a model of the window scanning robot. Based on the robot's pose estimated, the pose graph of the robot is constructed, and the window shape it is on is formed by arranging points obtained by the 2D laser range scanner, according to the pose graph and relative distances between the robot and the window frame. To improve the accuracy of the window shape, a loop closure [42] is performed when the robot returns to the start position, i.e., the loop of the pose graph is closed. The effectiveness of the proposed method is verified through the experiment in which the window scanning robot scans a frame of a window placed on the ground.

This paper is organized as follows. Section 2 refers to the related works concerning the window shape estimation. Section 3 presents the concept of nested reconfigurable robots for façade cleaning. Section 4 introduces the window scanning robot developed. Section 5 describes the method of the window shape estimation, which consists of the pose estimation with a robot model and the EKF and the loop closure based on the robot's pose, including the loop detection and pose adjustment. Section 6 devotes the experiment and its results to demonstrating the effectiveness of the proposed approach. Section 7 finally presents the concluding remarks and future work.

## 2. Related Work

This paper focuses on estimating a window shape to improve the adaptability of façade-cleaning robots. Research with respect to the window shape estimation was conducted from the perspective of façade cleaning and others.

In terms of façade cleaning, D. Y. Kim et al. proposed two approaches to detecting windows with a gondola-type robot equipped with a visual camera [43]. The authors utilized connect-component labeling and a histogram in each approach to extract a window from façade images. Furthermore, the authors improved the approach using a histogram to detect a tilted window [44].

The other perspective is the reconstruction of building models. Pu and Vosselman described an approach to extract windows from point clouds acquired by terrestrial laser scanning to reconstruct building models for virtual tourism, urban planning, and navigation systems [45]. To extract windows, the approach groups laser points in planar segments and detects walls, doors, and extrusions, applying feature constraints. Then, windows are detected through two strategies, depending on whether a window is covered with curtains or not. Pu and Vosselman also presented an automatic method for the reconstruction of building façade models from terrestrial laser scanning data, including window extraction [46]. The method provides polyhedron building models, utilizing knowledge about the features' sizes, positions, orientations, and topology to recognize features in a point cloud. Wang et al. presented an approach to window and façade detection with LiDAR data collected from a moving vehicle [47]. The proposed method combines bottom-up and top-down strategies to extract façade planes, and windows are detected by performing potential window point detection and window localization. Zolanvari et al. introduced a slicing method to quickly detect free-form openings and the overall boundaries from building façades with LiDAR point clouds [48]. In the method, each façade is roughly segmented by a RANSAC-based algorithm and sliced horizontally or vertically. In the slicing step, windows are detected, and then window boundaries are created.

Although the aforementioned works achieved the acquisition of window shapes by observing windows from outside with external sensors, our approach estimates window shapes by observing window frames with an external sensor mounted on a façade-cleaning robot on window surfaces. This is involved because of the need for additional equipment for observing windows from outside, which reduces the adaptability of façade-cleaning robots due to the limitation of installing the equipment in high-rise buildings. Our approach has the following contributions:

(1) A testbed for observing a window frame, called a window scanning robot, is presented: The window scanning robot having a 2D laser range scanner installed perpendicularly to a window surface is developed on the basis of a concept of nested reconfigurable robots for façade cleaning, detailed in the next section. This allows robots to observe window frames with little or no rising on the window surface they work on and to independently perform cleaning and exploration tasks. The window scanning robot offers an idea to acquire environmental data on a glass façade of a building for façade cleaning.

(2) A method for façade-cleaning robots to estimate a window shape is proposed: The window shape estimation is achieved by arranging points obtained by an external sensor and performing the loop closure based on the robot's pose estimated by the EKF. This is due to the environment on a window that has fewer features required for incorporating feature matching in a pose estimation, such as SLAM [38,39]. The method to obtain window shapes on the window surface a robot is on has not been presented to the knowledge of the authors.

(3) The validities of the window scanning robot and the window shape estimation method are demonstrated: Focusing on demonstrating the effectiveness of the ideas of window scanning and window shape estimation, we experiment with the window scanning robot developed on a window placed on the ground. The experimental results show that the robot can acquire the window shape by scanning the window frame, and

the proposed method is effective for estimating the shape of the window the robot works on.

## 3. Concept of Nested Reconfigurable Robots for Façade Cleaning

This paper discusses a method for window shape estimation based on a concept of nested reconfigurable robots for façade cleaning. The concept of nested reconfigurable robots for façade cleaning aims to develop robots that can be applied to autonomous façade cleaning on buildings with various types of architecture, such as a rounded glass surface and a spherical surface, shown in [4,27], improving the adaptability of façade-cleaning robots. In such various types of building architecture, while façade-cleaning robots work on flat glass panels connected by frames—called windows in this paper—especially in the case that window frames rise from window surfaces, they need to clean windows according to the frame shapes. The concept achieves autonomous façade cleaning according to window shapes, employing multiple modular multilegged robots capable of reconfiguring their morphology based on window shapes, which is executed by transforming their own modules and/or connecting with each other, and letting the robots cooperate (Figure 1). In the concept, the modular multilegged reconfigurable robots carry out tasks for façade cleaning, such as cleaning glass surfaces, the exploration of windows, and moving between windows through overcoming frames on each window.

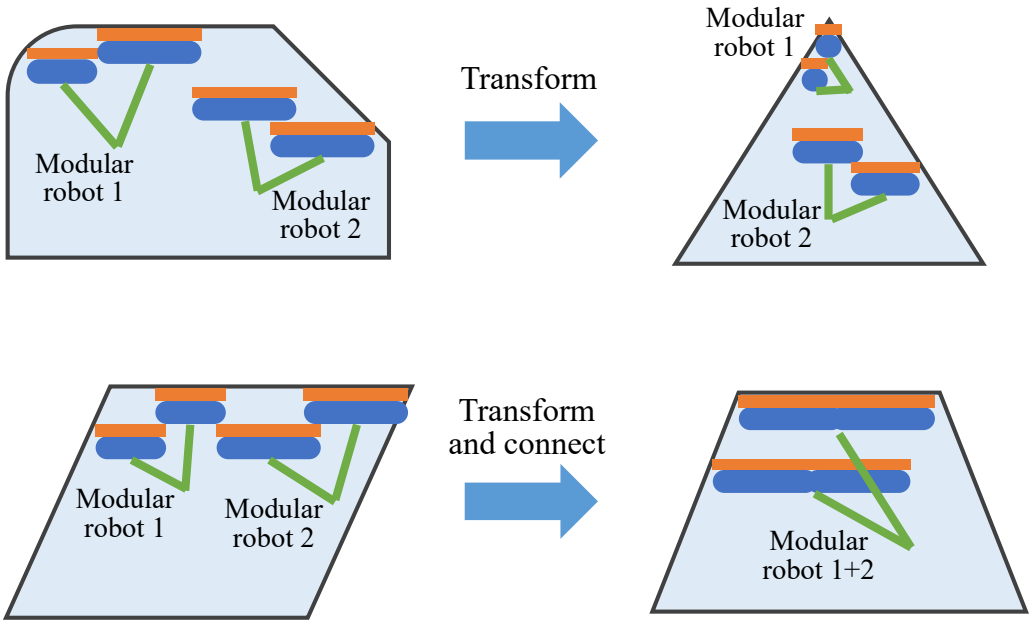



**Figure 1.** Concept of nested reconfigurable robots for façade cleaning. The concept employs multiple modular multilegged robots capable of reconfiguring their morphology based on window shapes. The modular multilegged reconfigurable robots can transform their own modules and connect with each other.

To achieve autonomous façade cleaning with the modular multilegged reconfigurable robot team, the window shape estimation is performed on each window by scanning window frames with one cleaning robot having an external sensor or a dedicated window scanning robot in the team. In the window shape estimation, a robot scanning a frame first searches for a part of the frame of the window the robot works on with an external sensor, turning at any initial point on the window the robot came in (Figure 2a). Once the robot detects a part of the window frame, the robot gets close to the window frame and starts moving along the frame, measuring the relative distance between the robot and the frame and the robot's pose (Figure 2b). After going around the same trajectory along the frame, the robot estimates the window shape based on the measured data (Figure 2c). Performing the above way on each window produces all the shapes of the windows in a façade.

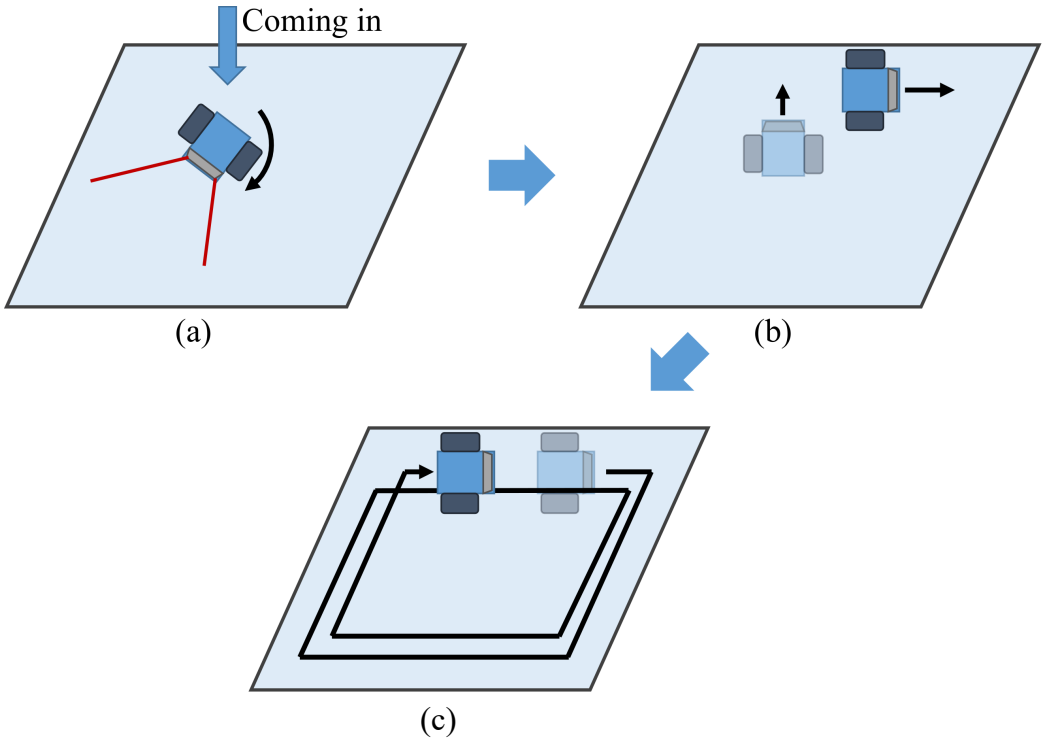



**Figure 2.** Exploration for the window shape estimation. (**a**) The scanning robot searches for a part of the window frame with an external sensor, turning at an initial point on the window the robot came in. (**b**) The robot gets close to the window frame and starts moving along the frame. (**c**) The robot goes around the same trajectory along the frame.

## 4. Window Scanning Robot

We consider that a façade-cleaning robot estimates a shape of a window it is on with data acquired by its own external sensor. In this paper, we focus on demonstrating the effectiveness of a method of the window shape estimation. Hence, we develop a robot to obtain window data with an external sensor, considering no façade-climbing mechanism, and validate a method of the window shape estimation with the developed robot through an experiment on a window placed on the ground.

The window scanning robot developed is shown in Figure 3, whose size is 189.7 $\times$ 138.7 $\times$ 191.9 mm and weight is 1.26 kg. The robot is equipped with a 2D laser range scanner installed perpendicularly to the ground to observe a window frame with little or no rising from the window surface because it is difficult to observe such a window frame with a 2D scanner installed horizontally. The robot can thus measure the relative distance between the robot and a window frame, moving on a window surface.

The system architecture of the window scanning robot is shown in Figure 4, which is developed on the basis of the architecture of TurtleBot3, the standard robot platform for the robot operating system (ROS). This system employs Raspberry Pi 3 Model B+ to activate the ROS, to which OpenCR, the open-source control module for ROS, and the 2D laser range scanner RPLIDAR A2M8 are connected via USB. OpenCR controls the motors Dynamixel XM430-W210 connected via the TTL communication interface, receiving a velocity command, and acquires data, such as the inertial measurement unit (IMU) and odometry, from sensors installed on the board. RPLIDAR A2M8 is controlled by Raspberry Pi 3 to obtain distance data from the robot to a window frame.

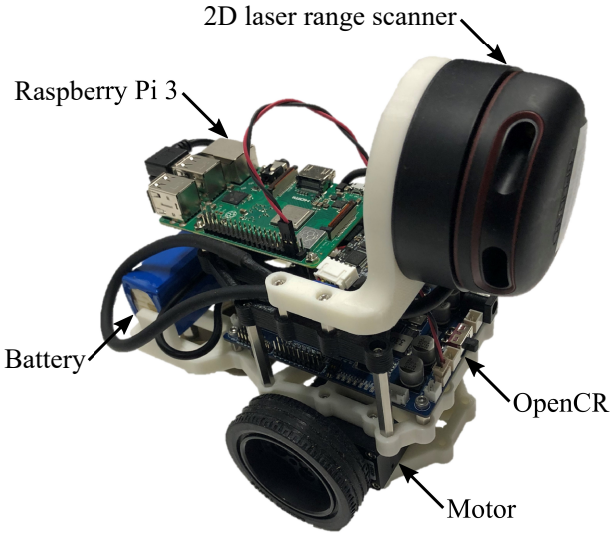



**Figure 3.** Window scanning robot. The robot's size is 189.7 × 138.7 × 191.9 mm, and its weight is 1.26 kg. The robot can obtain location information of window frames using a 2D laser range scanner installed perpendicularly to the ground.

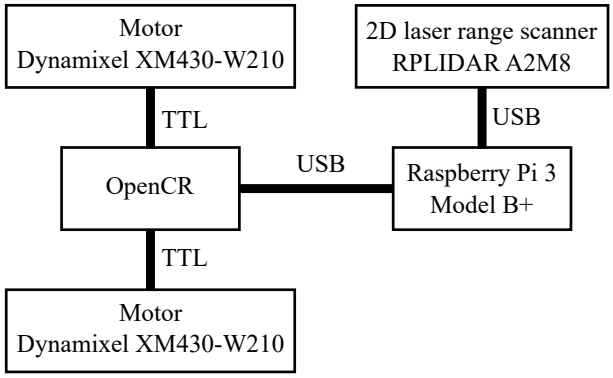



**Figure 4.** System architecture of the window scanning robot. The system is developed on the basis of the architecture of TurtleBot3 and consists of Raspberry Pi 3 Model B+, OpenCR, the RPLIDAR A2M8 2D laser range scanner, two Dynamixel XM430-W210 motors.

As shown in Figure 5, the 11.1 V 1800 mAh Li-Po battery is used for the power supply to OpenCR. From OpenCR, 5 V 4 A power is supplied to Raspberry Pi 3.




**Figure 5.** Power supply of the window scanning robot. The robot has an 11.1 V 1800 mAh Li-Po battery for the power supply to OpenCR. From OpenCR, 5 V 4 V power is supplied to Raspberry Pi 3.

## 5. Window Shape Estimation

In this paper, the window scanning robot cannot capture the overall shape of a window at one time because the robot scans the frame of a window on its surface. Thus, the window shape estimation is accomplished by measuring the relative distance between the robot and the window frame with the 2D laser range scanner along the frame and arranging the points obtained by the scanner according to the robot's pose.

The flowchart of the window shape estimation is shown in Figure 6. In the estimation, the robot's pose is estimated by the EKF [40,41] with the odometry data of the robot and is used to construct the pose graph of the robot. Based on the pose graph, the positions of the points obtained by the 2D scanner are recorded. In the loop closure [42], once it is

detected that the robot reaches the end point of the loop of the robot's trajectory, the pose adjustment is carried out to increase the accuracy of the window shape estimation.

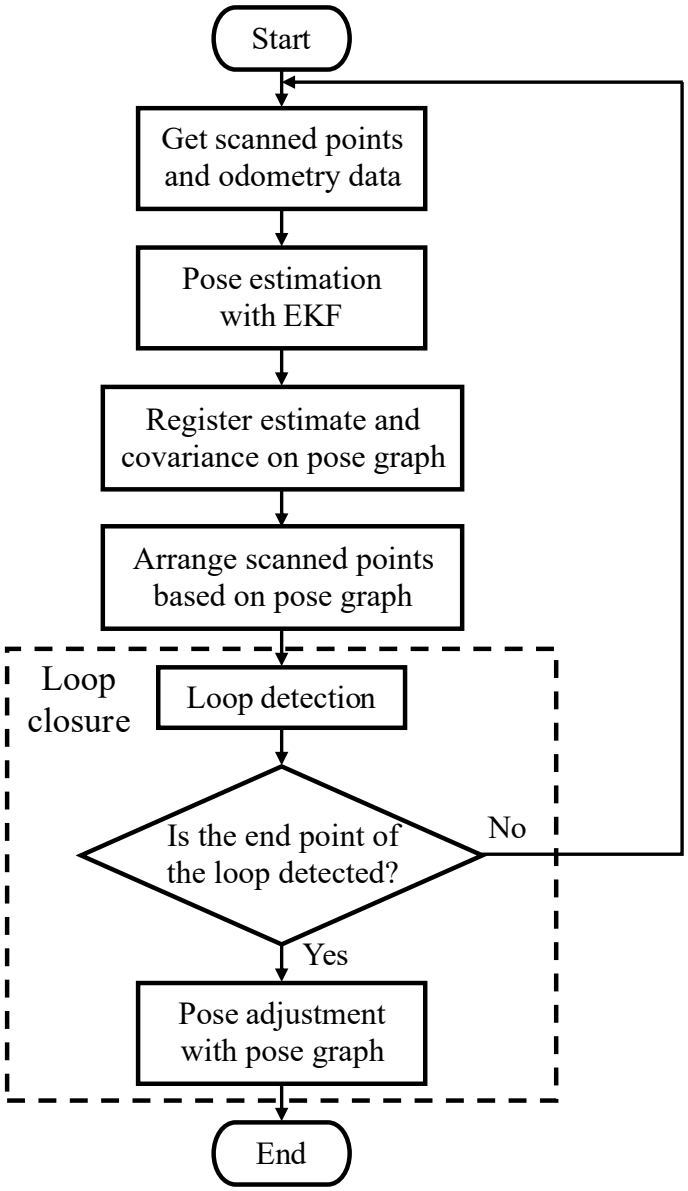



**Figure 6.** Flowchart of the window shape estimation. In the estimation, the robot's pose is estimated by the EKF with odometry data, and the estimate and the covariance of the robot's pose are registered on the pose graph of the robot. Based on the pose graph, the positions of the points obtained by the 2D scanner are recorded. In the loop closure, the end point of the loop of the robot's trajectory is detected, and the pose adjustment is carried out.

The variables and parameters are summarized in Tables 1 and 2.

**Table 1.** Variable descriptions.

| Symbol | Description |
|---|---|
| $x$ | Robot position in $X$ direction |
| $y$ | Robot position in $Y$ direction |
| $\theta$ | Robot heading angle |
| $v$ | Translational velocity input |
| $\omega$ | Rotational velocity input |
| $t$ | Timestep |
| $\boldsymbol{x}$ | Robot's pose: $\boldsymbol{x} = [x, y, \theta]^\mathsf{T}$ |
| $\boldsymbol{y}$ | Robot's observation: $\boldsymbol{y} = [x, y, \theta]^\mathsf{T}$ |
| $\boldsymbol{u}$ | Robot's input: $\boldsymbol{u} = [v, \omega]^\mathsf{T}$ |
| $\boldsymbol{d}$ | Relative robot's pose: $\boldsymbol{d} = [x', y', \theta']$ |
| $\boldsymbol{v}$ | Input noise: $\boldsymbol{v} \in \mathbb{R}^2$ |
| $\boldsymbol{w}$ | Observation noise: $\boldsymbol{w} \in \mathbb{R}^3$ |
| $\boldsymbol{Q}$ | Covariance of input noise: $\boldsymbol{Q} \in \mathbb{R}^{2\times2}$ |
| $\boldsymbol{R}$ | Covariance of observation noise: $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ |
| $\boldsymbol{P}$ | Covariance of robot's pose: $\boldsymbol{P} \in \mathbb{R}^{3\times3}$ |
| $\boldsymbol{\Sigma}$ | Covariance of relative robot's pose: $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$ |
| $\mathcal{X}$ | Set of robot's poses: $\boldsymbol{x} \in \mathcal{X}$ |
| $\mathcal{D}$ | Set of relative robot's poses: $\boldsymbol{d} \in \mathcal{D}$ |
| $\mathcal{G}$ | Set representing pose graph: $\mathcal{G} = (\mathcal{X}, \mathcal{D})$ |

**Table 2.** Parameter descriptions.

| Symbol | Description |
|---|---|
| $\Delta t$ | Travel time |
| $T$ | Time of the end of robot movement |
| $s$ | Time stamp at the start point of a loop |
| $e$ | Time stamp at the end point of a loop |
| $d_{\text{thr}}$ | Traveling distance threshold |
| $V_{\text{thr}}$ | Evaluation value threshold |
| $\boldsymbol{r}$ | Relative pose between start and end points: $\boldsymbol{r} \in \mathbb{R}^3$ |
| $\boldsymbol{S}$ | Covariance of relative pose between start and end points: $\boldsymbol{S} \in \mathbb{R}^{3\times3}$ |
| $\boldsymbol{K}$ | Covariance for the initial pose settlement: $\boldsymbol{K} \in \mathbb{R}^{3\times3}$ |
| $\boldsymbol{W}$ | Weight matrix: $\boldsymbol{W} \in \mathbb{R}^{3\times3}$ |
| $\mathcal{L}$ | Set of pairs of the time stamps $s$ and $e$: $(s, e) \in \mathcal{L}$ |

### *5.1. Pose Estimation*

The pose estimation of the window scanning robot is carried out, based on a model of the window scanning robot. Due to the simplicity of the robot model, the EKF [40,41] is employed to obtain the estimate and covariance of the robot's pose and construct the pose graph of the robot.

#### 5.1.1. Model of the Window Scanning Robot

The model of the window scanning robot used by the EKF is shown in Figure 7. In this model, $O$–$XY$ is the world coordinate, and $O_c$–$X_cY_c$ is the fixed coordinate to the robot, where $O_c$ is located on the center of the wheel shaft and $Y_c$ is along the shaft.

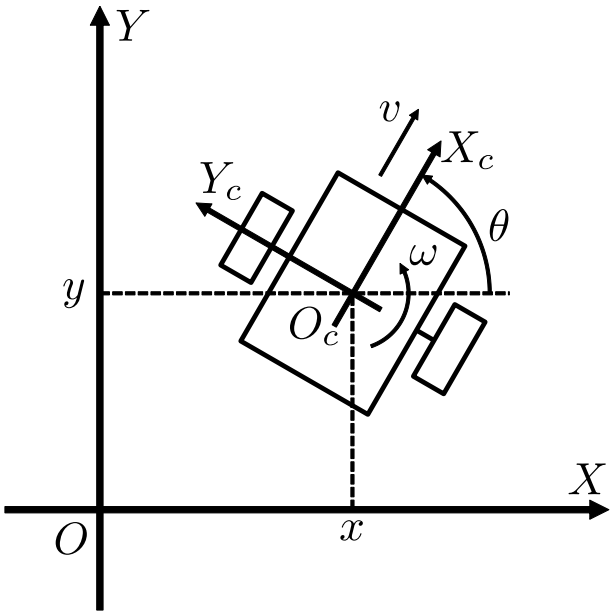



**Figure 7.** Model of the window scanning robot. $O$–$XY$ is the world coordinate, and $O_c$–$X_cY_c$ is the fixed coordinate to the robot, where $O_c$ is located on the center of the wheel shaft and $Y_c$ is along the haft.

In the model, the motion of the window scanning robot is represented as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix}, \tag{1}$$

where $(x, y)$ and $\theta$ are the position and the heading angle of the robot, and $v$ and $\omega$ are the translational and rotational velocities, which are the input of the robot. Thus, let $\boldsymbol{x}_t = [x_t, y_t, \theta_t]^\mathsf{T}$ as the robot's pose at timestep $t$ and $\boldsymbol{u}_t = [v_t, \omega_t]^\mathsf{T}$ as the input, the robot's pose after travel time $\Delta t$ is given as follows:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + v_t\Delta t\cos\theta_t \\ y_t + v_t\Delta t\sin\theta_t \\ \theta_t + \omega_t\Delta t \end{bmatrix}. \tag{2}$$

Based on (2), the following state and observation equations are established for the EKF:

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \begin{bmatrix} x_t + v_t\Delta t\cos\theta_t \\ y_t + v_t\Delta t\sin\theta_t \\ \theta_t + \omega_t\Delta t \end{bmatrix}, \tag{3}$$

$$\bar{\boldsymbol{u}}_t = \boldsymbol{u}_t + \boldsymbol{v}_t, \tag{4}$$

$$\boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_t) + \boldsymbol{w}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \boldsymbol{w}_t, \tag{5}$$

where $\boldsymbol{v}_t \in \mathbb{R}^2$ and $\boldsymbol{w}_t \in \mathbb{R}^3$ are the noise vectors for input and observation, respectively, and $\boldsymbol{v}_t \sim N(0, \boldsymbol{Q}_t)$ and $\boldsymbol{w}_t \sim N(0, \boldsymbol{R}_t)$ with the covariance of noise $\boldsymbol{Q}_t \in \mathbb{R}^{2\times2}$ and $\boldsymbol{R}_t \in \mathbb{R}^{3\times3}$. In the equations, $\bar{\boldsymbol{u}}_t$ denotes the input with the noise, and $\boldsymbol{y}_t$ is the observation values, which is the odometry data in the window scanning robot.

#### 5.1.2. Extended Kalman Filter

With the state and observation equations, we can obtain the estimate $\hat{\boldsymbol{x}}$ and covariance $\boldsymbol{P}$ of the robot's pose, according to the following algorithm of the EKF [40,41]:

Prediction step: The prior estimate $\hat{\boldsymbol{x}}_t^-$ and covariance $\boldsymbol{P}_t^-$ are calculated by applying the estimate $\hat{\boldsymbol{x}}_{t-1}$, input $\boldsymbol{u}_{t-1}$, and covariance $\boldsymbol{P}_{t-1}$ in previous timestep as follows:

$$\hat{\boldsymbol{x}}_t^- = \boldsymbol{f}(\hat{\boldsymbol{x}}_{t-1}, \boldsymbol{u}_{t-1}), \tag{6}$$

$$\boldsymbol{P}_t^- = \boldsymbol{A}_{t-1}\boldsymbol{P}_{t-1}\boldsymbol{A}_{t-1}^\mathsf{T} + \boldsymbol{B}_{t-1}\boldsymbol{Q}_{t-1}\boldsymbol{B}_{t-1}^\mathsf{T}, \tag{7}$$

where the matrices $\boldsymbol{A}_{t-1}$ and $\boldsymbol{B}_{t-1}$ are given by linearizing $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$:

$$\boldsymbol{A}_{t-1} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{t-1}, \boldsymbol{u}=\boldsymbol{u}_{t-1}} = \begin{bmatrix} 1 & 0 & -v_{t-1}\Delta t \sin\theta_{t-1} \\ 0 & 1 & v_{t-1}\Delta t \cos\theta_{t-1} \\ 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

$$\boldsymbol{B}_{t-1} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{t-1}, \boldsymbol{u}=\boldsymbol{u}_{t-1}} = \begin{bmatrix} \Delta t \cos\theta_{t-1} & 0 \\ \Delta t \sin\theta_{t-1} & 0 \\ 0 & \Delta t \end{bmatrix}. \tag{9}$$

Update step: The posterior estimate $\hat{\boldsymbol{x}}_t$ and covariance $\boldsymbol{P}_t$ are obtained by updating $\hat{\boldsymbol{x}}_t^-$ and $\boldsymbol{P}_t^-$ calculated in the prediction step with the observation values $\boldsymbol{y}_t$ as follows:

$$\hat{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_t^- + \boldsymbol{G}_t(\boldsymbol{y}_t - \boldsymbol{h}(\hat{\boldsymbol{x}}_t)), \tag{10}$$

$$\boldsymbol{P}_t = (\boldsymbol{I} - \boldsymbol{G}_t\boldsymbol{C}_t)\boldsymbol{P}_t^-, \tag{11}$$

where $\boldsymbol{G}_t$ is the Kalman gain calculated as

$$\boldsymbol{G}_t = \boldsymbol{P}_t^-\boldsymbol{C}_t^\mathsf{T}\left(\boldsymbol{C}_t\boldsymbol{P}_t^-\boldsymbol{C}_t^\mathsf{T} + \boldsymbol{R}_t\right)^{-1}. \tag{12}$$

In (12), the matrix $\boldsymbol{C}_t$ is given by linearizing $\boldsymbol{h}(\boldsymbol{x})$:

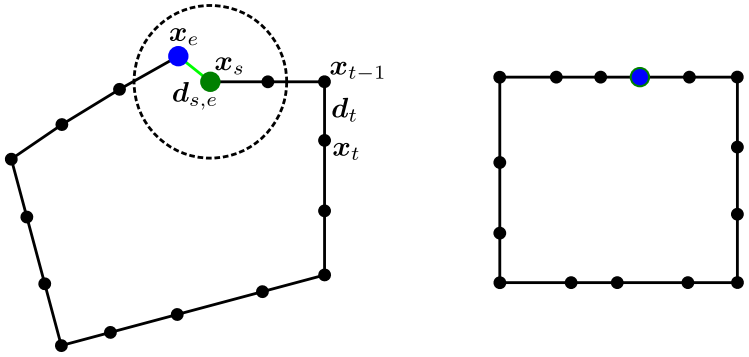
$$\boldsymbol{C}_t = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_t^-} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{13}$$

By repeating the prediction and update steps in every timestep, the estimate and covariance of the robot's pose are obtained to construct the pose graph of the robot.

### *5.2. Loop Closure*

The loop closure [42] is carried out to reduce accumulated error on the EKF and obtain an accurate shape of window frames. Upon the setup of the window scanning robot, it is difficult to perform the loop closure based on scan matching because its 2D laser range scanner is installed perpendicularly to the ground. In this paper, the loop closure hence exploits the result of the pose estimation.

To carry out the loop closure, the pose graph $\mathcal{G}_T = (\mathcal{X}_T, \mathcal{D}_T)$ of the window scanning robot moving by the time $T$, as shown in Figure 8, is applied. In this pose graph, the vertices $\mathcal{X}_T$ and the edges $\mathcal{D}_T$ represent the robot's poses $\boldsymbol{x}_t$ in the world coordinate and the relative poses $\boldsymbol{d}_t = [x_t', y_t', \theta_t']$ between $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$ in the fixed coordinate, respectively, that is, $\mathcal{X}_T = \{\boldsymbol{x}_t | t = 0, 1, \ldots, T\}$ and $\mathcal{D}_T = \{\boldsymbol{d}_t | t = 0, 1, \ldots, T\}$. The pose graph allows us to execute the loop closure through loop detection and pose adjustment.




(a) Pose graph (b) Adjusted pose graph

**Figure 8.** Pose graph. (**a**) The pose graph has the robot's pose $\boldsymbol{x}_t$ in the world coordinate as the vertex and the relative pose $\boldsymbol{d}_t = [x'_t, y'_t, \theta'_t]$ between $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$ in the fixed coordinate as the edge. (**b**) On the pose graph, the loop closure is carried out by matching the end point $\boldsymbol{x}_e$ of the loop to the start point $\boldsymbol{x}_s$.

#### 5.2.1. Loop Detection

The loop detection, whose process is summarized in Algorithm 1, is carried out to obtain the set $\mathcal{L}$ of a pair of the time stamps $(s, e)$ of the start and end points $\boldsymbol{x}_s$ and $\boldsymbol{x}_e$ of a loop, based on the pose estimation with the EKF in this paper. On the assumption that the window scanning robot moves around the same trajectory along the window frame on its surface, i.e., at the end of a loop the robot comes back to its start, the loop detection is executed with the following algorithm:

Step 1: Let $n = 1$, the time stamp $s_n$ of the start point $\boldsymbol{x}_{s_n}$ of a loop is set.

Step 2: If the traveling distance $d_{\text{total}} = \sum_{i \in [s_n, t]} \sqrt{(x'_i)^2 + (y'_i)^2}$ is larger than a distance threshold $d_{\text{thr}}$, the evaluation value $V_t$ of $\boldsymbol{x}_t$ is calculated as follows:

$$V_t = (\boldsymbol{x}_t - \boldsymbol{x}_{s_n})^\mathsf{T} \boldsymbol{W} (\boldsymbol{x}_t - \boldsymbol{x}_{s_n}), \tag{14}$$

where $\boldsymbol{W}$ is a weight matrix.

Step 3: If $V_{t-1} < V_{t-2}$, $V_{t-1} < V_t$, and $V_{t-1} < V_{\text{thr}}$, then the timestep $t-1$ of $\boldsymbol{x}_{t-1}$ is set as the time stamp $e_n$ of the end point $\boldsymbol{x}_{e_n}$ of the loop, where $V_{\text{thr}}$ is a value threshold.

Step 4: Once a pair of the time stamps $s_n$ and $e_n$ is given in the one loop, $n$ is updated as $n = n + 1$ and the timestep $t$ of $\boldsymbol{x}_t$ is set as the time stamp $s_n$ of the start point of a new loop.

Steps 2, 3, and 4 are repeated until all the loops are detected.

#### 5.2.2. Pose Adjustment

In the pose adjustment, the accurate pose graph is generated by reducing the accumulated error, as shown in Figure 8. The pose adjustment is executed by minimizing the following function with $\boldsymbol{x}$:

$$\begin{aligned} J \;=\; & (\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^\mathsf{T} \boldsymbol{K}^{-1} (\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0) \\ & + \sum_{t \in [1,T]} (\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t) - \boldsymbol{d}_t)^\mathsf{T} \boldsymbol{\Sigma}_t^{-1} (\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t) - \boldsymbol{d}_t) \\ & + \sum_{(s,e) \in L} \left(\boldsymbol{g}'(\boldsymbol{x}_s, \boldsymbol{x}_e) - \boldsymbol{r}_{s,e}\right)^\mathsf{T} \boldsymbol{S}_{s,e}^{-1} \left(\boldsymbol{g}'(\boldsymbol{x}_s, \boldsymbol{x}_e) - \boldsymbol{r}_{s,e}\right), \end{aligned} \tag{15}$$

where $\boldsymbol{r}_{s,e}$ is the relative pose between the start and end points, $\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t)$ and $\boldsymbol{g}'(\boldsymbol{x}_s, \boldsymbol{x}_e)$ are the functions to calculate a relative pose from $\boldsymbol{x}$ given as

$$\boldsymbol{g}(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t) = \begin{bmatrix} \cos\theta_{t-1} & \sin\theta_{t-1} & 0 \\ -\sin\theta_{t-1} & \cos\theta_{t-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t - x_{t-1} \\ y_t - y_{t-1} \\ \theta_t - \theta_{t-1} \end{bmatrix}, \tag{16}$$

$$\boldsymbol{g}'(\boldsymbol{x}_s, \boldsymbol{x}_e) = \begin{bmatrix} x_e - x_s \\ y_e - y_s \\ \theta_e - \theta_s \end{bmatrix}, \tag{17}$$

$\boldsymbol{K}$ is covariance to settle the initial pose $\boldsymbol{x}_0$ to the initial estimate $\hat{\boldsymbol{x}}_0$, and $\boldsymbol{\Sigma}_t$ and $\boldsymbol{S}_{s,e}$ are covariance of the relative poses $\boldsymbol{d}_t$ and $\boldsymbol{r}_{s,e}$, respectively. In this paper, $\boldsymbol{d}_t$ and $\boldsymbol{\Sigma}_t$ are obtained from the pose estimate $\hat{\boldsymbol{x}}$ and the covariance $\boldsymbol{P}$ through the EKF as follows:

$$\boldsymbol{d}_t = \boldsymbol{g}(\hat{\boldsymbol{x}}_{t-1}, \hat{\boldsymbol{x}}_t), \tag{18}$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{J}_d^{-1} \left( \boldsymbol{P}_t - \boldsymbol{J}_x \boldsymbol{P}_{t-1} \boldsymbol{J}_x^\mathsf{T} \right) \left( \boldsymbol{J}_d^\mathsf{T} \right)^{-1}, \tag{19}$$

$$\boldsymbol{J}_x = \begin{bmatrix} 1 & 0 & -x'_t \sin\theta_{t-1} - y'_t \cos\theta_{t-1} \\ 0 & 1 & x'_t \cos\theta_{t-1} - y'_t \sin\theta_{t-1} \\ 0 & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{J}_d = \begin{bmatrix} \cos\theta_{t-1} & -\sin\theta_{t-1} & 0 \\ \sin\theta_{t-1} & \cos\theta_{t-1} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and $\boldsymbol{r}_{s,e}$, $\boldsymbol{S}_{s,e}$, and $\boldsymbol{K}$ are adjustable parameters.

**Algorithm 1** Loop detection.

**Require:** $\mathcal{G}_T$, $\boldsymbol{W}$, $d_{\text{thr}}$, $V_{\text{thr}}$
**Ensure:** $\mathcal{L}$
  $\mathcal{L} = \{\}$
  $n = 1$
  $s_n = 0$
  **for** $t = 0, \ldots, T$ **do**
    $d_{\text{total}} = \sum_{i \in [s_n, t]} \sqrt{(x'_i)^2 + (y'_i)^2}$
    **if** $t \geq 2$ **and** $d_{\text{total}} > d_{\text{thr}}$ **then**
      $V_{t-2} = (\boldsymbol{x}_{t-2} - \boldsymbol{x}_{s_n})^\mathsf{T} \boldsymbol{W} (\boldsymbol{x}_{t-2} - \boldsymbol{x}_{s_n})$
      $V_{t-1} = (\boldsymbol{x}_{t-1} - \boldsymbol{x}_{s_n})^\mathsf{T} \boldsymbol{W} (\boldsymbol{x}_{t-1} - \boldsymbol{x}_{s_n})$
      $V_t = (\boldsymbol{x}_t - \boldsymbol{x}_{s_n})^\mathsf{T} \boldsymbol{W} (\boldsymbol{x}_t - \boldsymbol{x}_{s_n})$
      **if** $V_{t-1} < V_{t-2}$ **and** $V_{t-1} < V_t$ **and** $V_{t-1} < V_{\text{thr}}$ **then**
        $e_n = t - 1$
        $\mathcal{L} = \mathcal{L} \cup \{(s_n, e_n)\}$
        $n = n + 1$
        $s_n = t$
      **end if**
    **end if**
  **end for**

## 6. Experiment

We demonstrate the effectiveness of the proposed approach through the experiment. In the experiment, the window scanning robot moves around the same trajectory along the frame of the rectangle window, whose size is 900 × 1800 mm, on its surface placed on the floor (Figure 9), thereby measuring the distance between the robot and its frame with the 2D laser range scanner installed perpendicularly to its surface. Using this experiment data, we carry out the pose estimation of the robot with the EKF offline and generate the

robot's pose graph which stores the pose estimate and covariance with thinning out to reduce the computational load. On the pose graph, all the pairs of the start and end points in a loop are acquired by the loop detection and then the pose adjustment is executed for the loop closure.

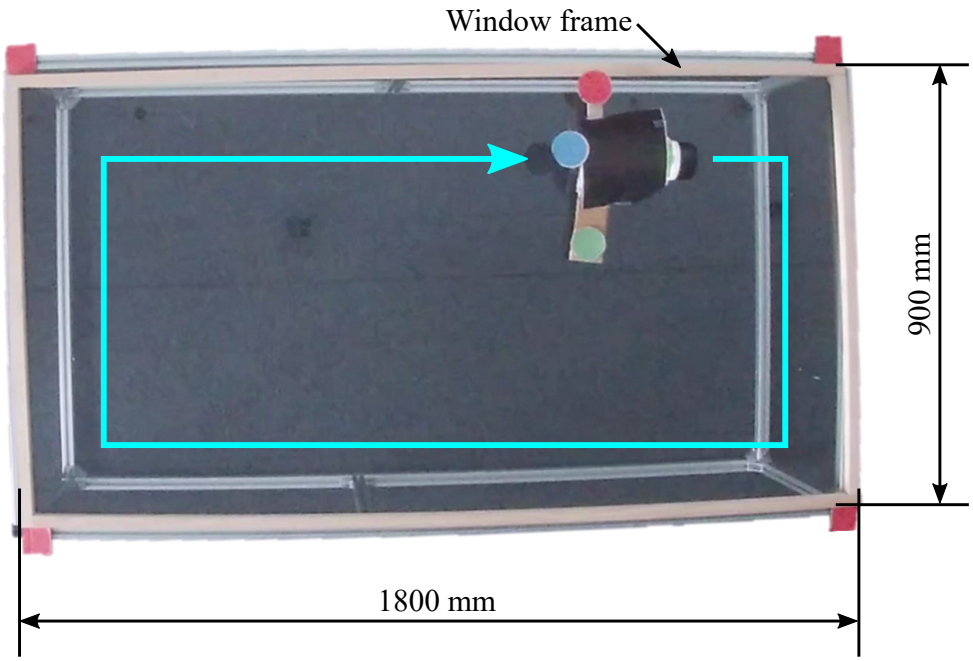



**Figure 9.** Experimental setup. In the experiments, the window scanning robot moves around the same trajectory along the frame of the rectangle window on its surface placed on the floor. Its size is 900 × 1800 mm.

### *6.1. ROS-Based Experimental System*

To carry out the experiment, the ROS-based system shown in Figure 10 was implemented. In this implementation, the *keyop_sensing_robot* node for the manual control of the window scanning robot is activated in a laptop PC to send */velocity_command*. This command is processed by *sensing_robot_node* in Raspberry Pi 3 installed on the robot to generate motor input. *sensing_robot_node* concurrently provides *imu_data* and *odometry_data*. In the Raspberry Pi 3, *rplidar_node* is also activated to control the 2D laser range scanner and provides *scan_data*. These data are stored by *server_node* in the laptop PC to be applied for the evaluation of the proposed approach.

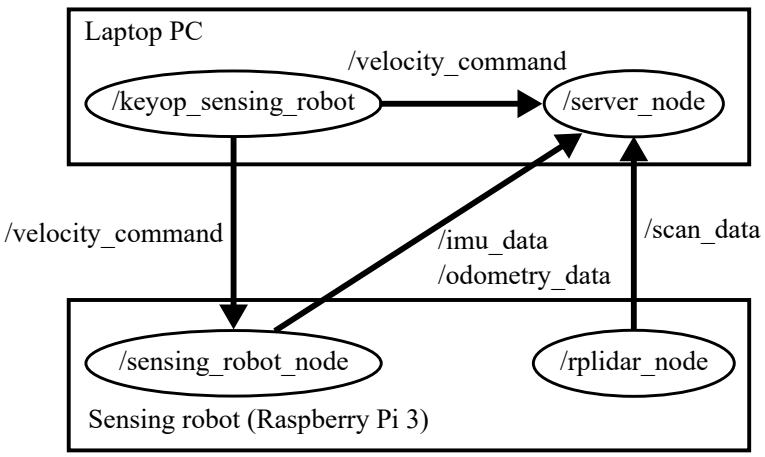



**Figure 10.** ROS-based implementation. The implementation consists of four ROS nodes: *keyop_sensing_robot*, *server_node*, *sensing_robot_node*, and *rplidar_node*.

*6.2. Variable and Parameter Settings*

For the pose estimation with the EKF, we set the initial estimate and covariance of the robot's pose

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0, \;\; \boldsymbol{P}_0 = \mathrm{diag}(1.00 \times 10^3, 1.00 \times 10^3, 1.00 \times 10^3),$$

and the covariance of noise

$$\begin{aligned} \boldsymbol{Q} &= \mathrm{diag}(1.00 \times 10^{-3}, 1.00 \times 10^{-3}), \\ \boldsymbol{R} &= \mathrm{diag}(1.00 \times 10^3, 1.00 \times 10^3, 1.00 \times 10^3). \end{aligned}$$

For the loop detection and the pose adjustment, we set the following parameters:

$$\begin{aligned} d_{\mathrm{thr}} &= 1.00, \\ V_{\mathrm{thr}} &= 5.00 \times 10^{-2}, \\ \boldsymbol{W} &= \mathrm{diag}(3.00, 3.00, 5.00 \times 10^{-1}), \\ \boldsymbol{r}_{s,e} &= \left[0.00, 0.00, 2.62 \times 10^{-1}\right]^{\mathsf{T}}, \\ \boldsymbol{S}_{s,e} &= \mathrm{diag}(10.0, 10.0, 10.0), \\ \boldsymbol{K} &= \mathrm{diag}(1.00 \times 10^6, 1.00 \times 10^6, 1.00 \times 10^6). \end{aligned}$$

*6.3. Experimental Results*

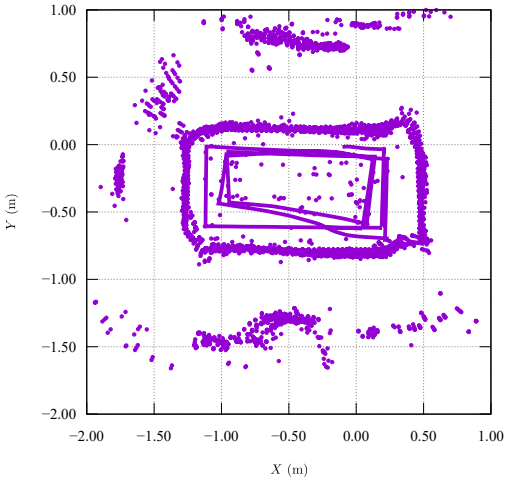
The results of the pose estimation and the loop closure are shown in Figures 11–14. Figure 11 shows the robot's trajectory and the scanned shape of the window frame obtained by the (a) measurement, (b) pose estimation, and (c) loop closure, respectively. This figure indicates the robot's trajectories with lines and the positions of objects scanned by the 2D laser range scanner with dots, where we removed the dots indicating the positions of objects that are not located at the level of the window frame, such as a floor, walls, and ceiling. Figures 12–14 show the time-series data of the robot's position and heading angle.

Figure 11a indicates that the window scanning robot was able to obtain the shape of the rectangular window frame, which is described outside the robot's trajectory by the gathered scan dots along the trajectory, and the size of the scanned window is broadly correct. The dots forming the window shape can be distinguished from dots indicating other objects, although the window shape is slightly twisted. This result argues that the shape of a window frame can be measured by scanning a window frame perpendicularly to the window surface with a 2D laser range scanner and this task is viable by a robot on the scanned window.

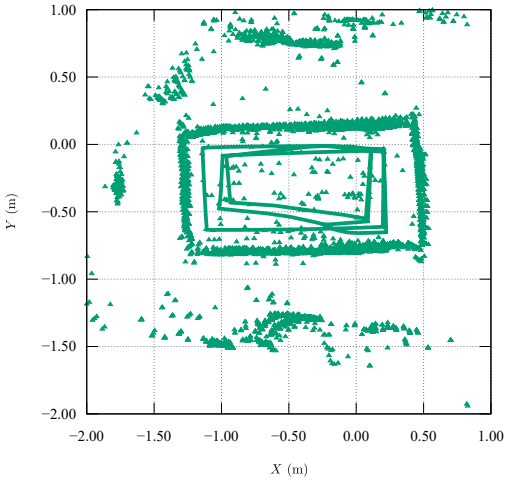
Figure 11b declares that the pose estimation with the EKF reduces the error of the robot's trajectory and thereby improves the accuracy of the window shape estimation. The window shape based on the estimates of the robot's pose is less twisted than that based on the measurement data. This result presents that the pose estimation with the proposed model and the EKF contributes to increasing the feasibility of the proposed window shape estimation with a 2D laser range scanner.

Figure 11c shows that the loop closure based on the pose graph utilizing the estimates influences the estimated shape of the window frame. This estimated window shape is rarely different from that in Figure 11b. However, the isolated dots decrease in Figure 11c. This result indicates that the loop closure is effective for the window shape estimation but also implies the need to improve the proposed method.

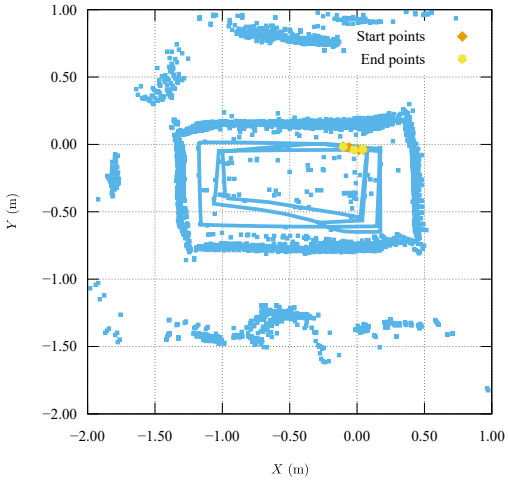
In the time scale, Figures 12 and 13 show that the loop closure adjusts the robot's positions in the horizontal direction during translational movement. By contrast, it hardly affected the heading angle of the robot, as shown in Figure 14.




(a) Measure




(b) Estimate




(c) Pose graph

**Figure 11.** Robot's trajectory and the scanned shape of the window frame. They are obtained by (**a**) measurement, (**b**) pose estimation, and (**c**) loop closure, respectively. They indicate the robot's trajectories with lines and the positions of objects scanned by the 2D laser range scanner with dots. The shape of the rectangular window frame is described outside the robot's trajectory by the gathered scan dots along the trajectory.

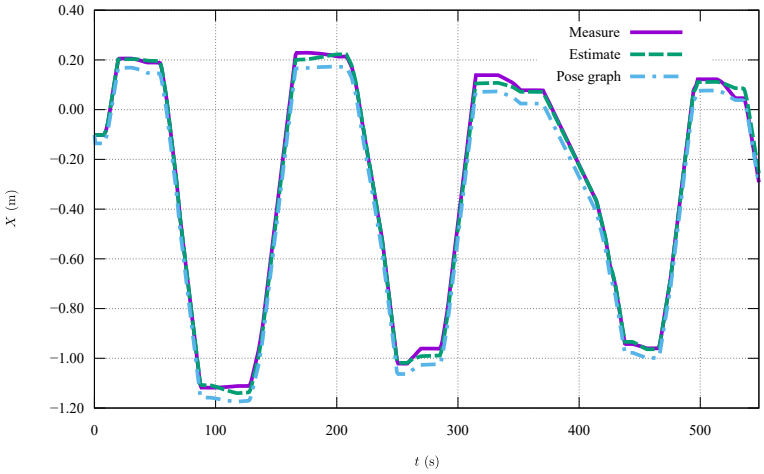



**Figure 12.** *X*-direction position of the robot. The robot's positions in the horizontal direction during translational movement are adjusted by the loop closure.

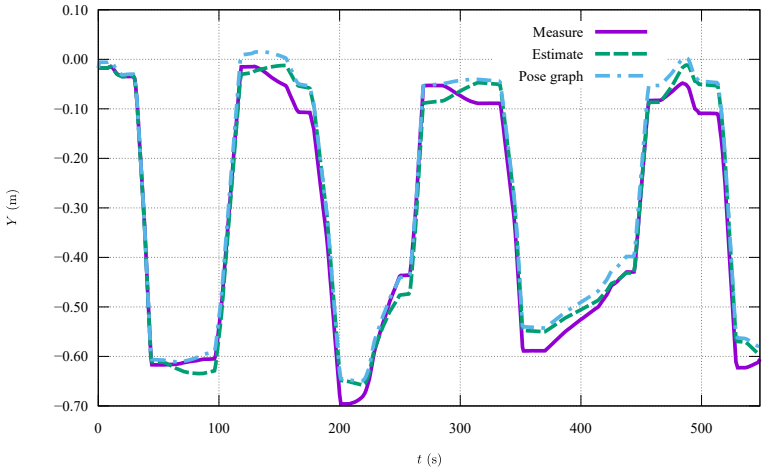



**Figure 13.** $Y$-direction position of the robot. The robot's positions in the horizontal direction during translational movement are adjusted by the loop closure.

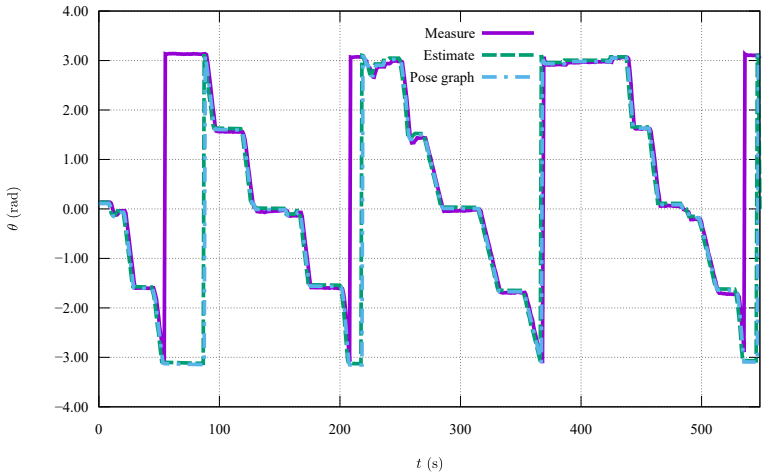



**Figure 14.** Heading angle of the robot. The heading angle of the robot is hardly affected by the loop closure.

## 7. Conclusions

In this paper, we have presented an approach to estimating the shape of a window so that a façade-cleaning robot can acquire information about the window it works on. To this end, we developed a robot equipped with a 2D laser range scanner installed perpendicularly to the ground to observe a window frame and proposed a method of window shape estimation with the developed robot, consisting of pose estimation and a loop closure. For this method, the pose estimation with the EKF was employed and the loop closure algorithm based on the estimate of the robot's pose was devised. This loop closure was accomplished by defining the start and the end point of a loop in the loop detection and modifying a pose graph of the robot in the pose adjustment. To demonstrate the effectiveness of the proposed approach, the experiment with the window scanning robot developed was carried out on a window placed on the floor. The experimental results have shown that the robot can acquire the window shape by scanning the window frame with the 2D laser range scanner installed perpendicularly to the window, and the proposed approach is effective for estimating a window shape it works on. In future work, we would improve the proposed approach to increase the accuracy of the window shape estimation. To this end, we will apply a different filter to the pose estimation, such as a fuzzy-based Kalman filter [49], and refine the method of the loop closure to increase the applicability to different shapes of windows. Moreover, we will carry out experiments using windows with rough glass surfaces to verify the influence of the geometry of window surfaces on the proposed approach.

**Author Contributions:** Conceptualization, T.N. and S.N.; methodology, T.N., S.N. and S.I.; software, T.N.; validation, T.N. and S.N.; formal analysis, T.N.; investigation, T.N. and S.N.; resources, S.N., M.I. and H.I.; data curation, T.N. and S.N.; writing—original draft preparation, T.N.; writing—review and editing, T.N., S.N., S.I., M.I. and H.I.; visualization, T.N.; supervision, S.N., M.I. and H.I.; project administration, S.N., M.I. and H.I.; funding acquisition, S.N. and H.I. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Research Institute for Science and Technology of Tokyo Denki University, grant number Q18T-06 and 19T-08/Japan. The APC was funded by Q18T-06 and 19T-08/Japan.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** The authors would like to thank S. Sasaki and M. Sasahira for assistance with the experiments.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| EKF | Extended Kalman filter |
| ROS | Robot operating system |
| IMU | Inertial measurement unit |

## References

1. BBC. Shanghai window cleaning cradle swings out of control. *BBC News*, 3 April 2015.
2. BBC. Window washers rescued from high up world trade center. *BBC News*, 12 November 2014.
3. Elkmann, N.; Hortig, J.; Fritzsche, M. Cleaning automation. In *Springer Handbook of Automation*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1253–1264. [CrossRef]
4. Seo, T.; Jeon, Y.; Park, C.; Kim, J. Survey on glass and façade-cleaning robots: Climbing mechanisms, cleaning methods, and applications. *Int. J. Precis. Eng.-Manuf.-Green Technol.* **2019**, *6*, 367–376. [CrossRef]
5. Elkmann, N.; Felsch, T.; Sack, M.; Saenz, J.; Hortig, J. Innovative service robot systems for facade cleaning of difficult-to-access areas. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002; Volume 1, pp. 756–762. [CrossRef]
6. Elkmann, N.; Kunst, D.; Krueger, T.; Lucke, M.; Böhme, T.; Felsch, T.; Stürze, T. SIRIUSc—Façade cleaning robot for a high-rise building in Munich, Germany. In *Climbing and Walking Robots*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1033–1040. [CrossRef]
7. Elkmann, N.; Lucke, M.; Krüger, T.; Kunst, D.; Stürze, T.; Hortig, J. Kinematics, sensors and control of the fully automated façade-cleaning robot SIRIUSc for the Fraunhofer headquarters building, Munich. *Ind. Robot Int. J.* **2008**. [CrossRef]
8. Lee, S.; Kang, M.S.; Han, C.S. Sensor based motion planning and estimation of highrise building façade maintenance robot. *Int. J. Precis. Eng. Manuf.* **2012**, *13*, 2127–2134. [CrossRef]
9. Moon, S.M.; Hong, D.; Kim, S.W.; Park, S. Building wall maintenance robot based on built-in guide rail. In Proceedings of the 2012 IEEE International Conference on Industrial Technology, Athens, Greece, 19–21 March 2012; pp. 498–503. [CrossRef]
10. Shin, C.; Moon, S.; Kwon, J.; Huh, J.; Hong, D. Force control of cleaning tool system for building wall maintenance robot on built-in guide rail. In Proceedings of the International Symposium on Automation and Robotics in Construction, Sydney, Australia, 9–11 July 2014; Volume 31, p. 1. [CrossRef]
11. Moon, S.M.; Shin, C.Y.; Huh, J.; Oh, K.W.; Hong, D. Window cleaning system with water circulation for building façade maintenance robot and its efficiency analysis. *Int. J. Precis. Eng.-Manuf.-Green Technol.* **2015**, *2*, 65–72. [CrossRef]
12. Lee, Y.S.; Kim, S.H.; Gil, M.S.; Lee, S.H.; Kang, M.S.; Jang, S.H.; Yu, B.H.; Ryu, B.G.; Hong, D.; Han, C.S. The study on the integrated control system for curtain wall building façade cleaning robot. *Autom. Constr.* **2018**, *94*, 39–46. [CrossRef]
13. Lee, C.; Chu, B. Three-modular obstacle-climbing robot for cleaning windows on building exterior walls. *Int. J. Precis. Eng. Manuf.* **2019**, *20*, 1371–1380. [CrossRef]
14. Yoo, S.; Joo, I.; Hong, J.; Park, C.; Kim, J.; Kim, H.S.; Seo, T. Unmanned high-rise façade cleaning robot implemented on a gondola: Field test on 000-building in Korea. *IEEE Access* **2019**, *7*, 30174–30184. [CrossRef]
15. Hong, J.; Park, G.; Lee, J.; Kim, J.; Kim, H.S.; Seo, T. Performance comparison of adaptive mechanisms of cleaning module to overcome step-shaped obstacles on façades. *IEEE Access* **2019**, *7*, 159879–159887. [CrossRef]
16. Park, G.; Hong, J.; Yoo, S.; Kim, H.S.; Seo, T. Design of a 3-DOF parallel manipulator to compensate for disturbances in facade cleaning. *IEEE Access* **2020**, *8*, 9015–9022. [CrossRef]

17. Chae, H.; Park, G.; Lee, J.; Kim, K.; Kim, T.; Kim, H.S.; Seo, T. Façade cleaning robot with manipulating and sensing devices equipped on a gondola. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 1719–1727. [CrossRef]
18. Zhu, J.; Sun, D.; Tso, S.K. Application of a service climbing robot with motion planning and visual sensing. *J. Robot. Syst.* **2003**, *20*, 189–199. [CrossRef]
19. Sun, D.; Zhu, J.; Lai, C.; Tso, S. A visual sensing application to a climbing cleaning robot on the glass surface. *Mechatronics* **2004**, *14*, 1089–1104. [CrossRef]
20. Sun, D.; Zhu, J.; Tso, S.K. A Climbing Robot for Cleaning Glass Surface with Motion Planning and Visual Sensing. In *Climbing and Walking Robots: Towards New Applications*; IntechOpen: London, UK, 2007. [CrossRef]
21. Zhang, H.; Zhang, J.; Zong, G. Requirements of glass cleaning and development of climbing robot systems. In Proceedings of the 2004 International Conference on Intelligent Mechatronics and Automation, Chengdu, China, 26–31 August 2004; pp. 101–106. [CrossRef]
22. Zhang, H.; Zhang, J.; Zong, G. Realization of a service climbing robot for glass-wall cleaning. In Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics, Shenyang, China, 22–26 August 2004; pp. 395–400. [CrossRef]
23. Zhang, H.; Zhang, J.; Liu, R.; Zong, G. A novel approach to pneumatic position servo control of a glass wall cleaning robot. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 1, pp. 467–472. [CrossRef]
24. Zhang, H.; Zhang, J.; Zong, G. Effective pneumatic scheme and control strategy of a climbing robot for class wall cleaning on high-rise buildings. *Int. J. Adv. Robot. Syst.* **2006**, *3*, 28. [CrossRef]
25. Zhang, H.; Zhang, J.; Zong, G.; Wang, W.; Liu, R. Sky cleaner 3: A real pneumatic climbing robot for glass-wall cleaning. *IEEE Robot. Autom. Mag.* **2006**, *13*, 32–41. [CrossRef]
26. Zhang, H.; Zhang, J.; Zong, G. Effective nonlinear control algorithms for a series of pneumatic climbing robots. In Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics, Kunming, China, 17–20 December 2006; pp. 994–999. [CrossRef]
27. Zhang, H.; Zhang, J.; Liu, R.; Zong, G. Realization of a service robot for cleaning spherical surfaces. *Int. J. Adv. Robot. Syst.* **2005**, *2*, 7. [CrossRef]
28. Seo, K.; Cho, S.; Kim, T.; Kim, H.S.; Kim, J. Design and stability analysis of a novel wall-climbing robotic platform (ROPE RIDE). *Mech. Mach. Theory* **2013**, *70*, 189–208. [CrossRef]
29. Kim, T.Y.; Kim, J.H.; Seo, K.C.; Kim, H.M.; Lee, G.U.; Kim, J.W.; Kim, H.S. Design and control of a cleaning unit for a novel wall-climbing robot. *Appl. Mech. Mater.* **2014**, *541*, 1092–1096. [CrossRef]
30. Kim, T.; Seo, K.; Kim, J.; Kim, H.S. Adaptive impedance control of a cleaning unit for a novel wall-climbing mobile robotic platform (ROPE RIDE). In Proceedings of the 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Besacon, France, 8–11 July 2014; pp. 994–999. [CrossRef]
31. Tun, T.T.; Elara, M.R.; Kalimuthu, M.; Vengadesh, A. Glass facade cleaning robot with passive suction cups and self-locking trapezoidal lead screw drive. *Autom. Constr.* **2018**, *96*, 180–188. [CrossRef]
32. Vega-Heredia, M.; Mohan, R.E.; Wen, T.Y.; Siti'Aisyah, J.; Vengadesh, A.; Ghanta, S.; Vinu, S. Design and modelling of a modular window cleaning robot. *Autom. Constr.* **2019**, *103*, 268–278. [CrossRef]
33. Vega-Heredia, M.; Muhammad, I.; Ghanta, S.; Ayyalusami, V.; Aisyah, S.; Elara, M.R. Multi-sensor orientation tracking for a façade-cleaning robot. *Sensors* **2020**, *20*, 1483. [CrossRef]
34. Chae, H.; Moon, Y.; Lee, K.; Park, S.; Kim, H.S.; Seo, T. A Tethered Façade Cleaning Robot Based on a Dual Rope Windlass Climbing Mechanism: Design and Experiments. *IEEE/ASME Trans. Mechatron.* **2022**. [CrossRef]
35. Nansai, S.; Elara, M.R.; Tun, T.T.; Veerajagadheswar, P.; Pathmakumar, T. A novel nested reconfigurable approach for a glass façade cleaning robot. *Inventions* **2017**, *2*, 18. [CrossRef]
36. Nansai, S.; Onodera, K.; Veerajagadheswar, P.; Rajesh Elara, M.; Iwase, M. Design and experiment of a novel façade cleaning robot with a biped mechanism. *Appl. Sci.* **2018**, *8*, 2398. [CrossRef]
37. Nansai, S.; Itoh, H. Foot location algorithm considering geometric constraints of façade cleaning. *J. Adv. Simul. Sci. Eng.* **2019**, *6*, 177–188. [CrossRef]
38. Singandhupe, A.; La, H.M. A review of slam techniques and security in autonomous driving. In Proceedings of the 2019 third IEEE international conference on robotic computing (IRC), Naples, Italy, 25–27 February 2019; pp. 602–607. [CrossRef]
39. Cheng, J.; Zhang, L.; Chen, Q.; Hu, X.; Cai, J. A review of visual SLAM methods for autonomous driving vehicles. *Eng. Appl. Artif. Intell.* **2022**, *114*, 104992. [CrossRef]
40. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005; ISBN 0-262-20162-3/978-0-262-20162-9.
41. Adachi, S.; Maruta, I. *Fundamentals of Kalman Filter*; Tokyo Denki University Press: Tokyo, Japan, 2012. (In Japanese)
42. Tomono, M. *Simultaneous Localization and Mapping*; Ohmsha: Tokyo, Japan, 2018. (In Japanese)
43. Kim, D.Y.; Yoon, J.; Sun, H.; Park, C.W. Window detection for gondola robot using a visual camera. In Proceedings of the 2012 IEEE International Conference on Automation Science and Engineering (CASE), Seoul, Republic of Korea, 20–24 August 2012; pp. 998–1003. [CrossRef]
44. Kim, D.Y.; Yoon, J.; Cha, D.H.; Park, C.W. Tilted Window Detection for Gondolatyped Facade Robot. *Int. J. Control Theory Comput. Model. (IJCTCM)* **2013**, *3*, 1–10. [CrossRef]

45. Pu, S.; Vosselman, G. Extracting windows from terrestrial laser scanning. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2007**, *36*, 12–14.
46. Pu, S.; Vosselman, G. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 575–584. [CrossRef]
47. Wang, R.; Bach, J.; Ferrie, F.P. Window detection from mobile LiDAR data. In Proceedings of the 2011 IEEE Workshop on Applications of Computer Vision (WACV), Kona, HI, USA, 5–7 January 2011; pp. 58–65. [CrossRef]
48. Zolanvari, S.I.; Laefer, D.F. Slicing Method for curved façade and window extraction from point clouds. *ISPRS J. Photogramm. Remote Sens.* **2016**, *119*, 334–346. [CrossRef]
49. Qasem, S.N.; Ahmadian, A.; Mohammadzadeh, A.; Rathinasamy, S.; Pahlevanzadeh, B. A type-3 logic fuzzy system: Optimized by a correntropy based Kalman filter with adaptive fuzzy kernel size. *Inf. Sci.* **2021**, *572*, 424–443. [CrossRef]


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.