

Article

# Simultaneous Localization and Mapping Methods for Snake-like Robots Based on Gait Adjustment

Chaoquan Tang, Zhipeng Zhang, Meng Sun, Menggang Li , Hongwei Tang and Deen Bai \* 

Jiangsu Key Laboratory of Mine Mechanical and Electrical Equipment, School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou 221116, China; tangchaoquan@cumt.edu.cn (C.T.); ts24050084a31@cumt.edu.cn (Z.Z.); dundsm@163.com (M.S.); sallylmg@cumt.edu.cn (M.L.); thw@cumt.edu.cn (H.T.)

\* Correspondence: baideen@cumt.edu.cn

**Abstract:** Snake robots require autonomous localization and mapping capabilities for field applications. However, the characteristics of their motion, such as large turning angles and fast rotation speeds, can lead to issues like drift or even failure in positioning and map building. In response to this situation, this paper starts from the gait motion characteristics of the snake robot itself, proposing an improved gait motion method and a tightly coupled method based on IMU and visual information to solve the problem of poor algorithm convergence caused by head-shaking in snake robot SLAM. Firstly, the adaptability of several typical gaits of the snake robot to SLAM methods was evaluated. Secondly, the serpentine gait was selected as the object of gait improvement, and a head stability control method for the snake robot was proposed, thereby reducing the interference of the snake robot's motion on the sensors. Thirdly, a visual-inertial tightly coupled SLAM method for the snake robot's serpentine gait and Arc-Rolling gait was proposed, and the method was verified to enhance the robustness of the visual SLAM algorithm and improve the positioning and mapping accuracy of the snake robot. Finally, experiments proved that the methods proposed in this paper can effectively improve the accuracy of positioning and map building for snake robots.

**Keywords:** snake robot; SLAM; typical gait; stability control; IMU



**Citation:** Tang, C.; Zhang, Z.; Sun, M.; Li, M.; Tang, H.; Bai, D. Simultaneous Localization and Mapping Methods for Snake-like Robots Based on Gait Adjustment. *Biomimetics* **2024**, *9*, 710. <https://doi.org/10.3390/biomimetics9110710>

Academic Editors: Changsheng Wu and Stanislav N. Gorb

Received: 3 September 2024

Revised: 14 October 2024

Accepted: 14 November 2024

Published: 19 November 2024



**Copyright:** © 2024 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

Currently, robots have a wide range of applications in many fields, such as security, search and rescue, and transportation. Operating in narrow and confined spaces is one of the practical requirements for robots in these fields, requiring specialized capabilities such as compact size and strong environmental adaptability [1]. Among all types of robots, serpentine robots have a thin cross-section and an elongated flexible body, which helps the body to move forward in very narrow spaces with minimal interference to the surrounding environment, making it highly advantageous in the detection of complex and narrow environments, such as post-disaster rescue, underground exploration, and so on [2]. However, in GPS denial environments, snake robots are unable to obtain their position and environmental information, making it difficult for them to realize autonomous navigation and thus difficult to discover the purpose of performing tasks autonomously [3]. SLAM technology addresses the problem of recovering the relative spatial relationship between the observer itself and its surroundings, giving a description of the surrounding environment. With this foundation, the snake robots can carry out path planning, detect obstacles encountered promptly, and avoid obstacles encountered in time [4,5].

SLAM is an important technique to estimate the state of the target and its surroundings [6–8]. Currently, snake robots face many challenges in SLAM applications, such as large hardware size, high energy consumption, algorithmic complexity, and high computational effort [9]. Although LiDAR and ultrasonic sensors can provide environmental depth information to help navigation and localization, they usually have insufficient depth

resolution and lack color information, which limits their application in target detection and recognition [10]. In contrast, vision cameras can capture rich scene details and have the advantages of small size, low energy consumption, and cost-effectiveness, making them ideal for SLAM solutions for snake robots [11].

However, the snake robot itself has a variety of motion forms, such as serpentine, rolling, side-shifting, side-rolling, lifting up, overcoming obstacles, etc. During the rapid movement and rotation of the snake robot, there are motion characteristics such as violent shaking, large corners, and fast rotation speed that occur as a result of feature-point tracking failure, leading to poor localization and mapping accuracy, poor robustness, and even failure [12,13]. In a previous work, Gong. C et al. [14] used a modal decomposition method to keep the head of a snake robot in the same direction during rolling gait. This method ensures that the head camera of the snake robot works properly during rolling gait. H. Yamada et al. [15] stabilized the head of the snake robot by controlling the movement of its neck. However, when the method is converted from a continuous model to a discrete model, it causes the head position to move back and forth, which affects the control of the robot. Z. Bing [16] compensated for the head offset angle of the snake robot by calculating the derivative of the snake curve concerning time. Wu et al. [17] introduced virtual joints as the desired directions and analyzed the virtual joints in the HNC (Head Navigational Control) about the first two head joint angle relationships. However, this lacks validation of the head navigation effect, and it is difficult to adjust the parameters using a cyclic inhibition model. G. Qiao et al. [18] implemented partial control by proposing to divide the snake robot into different parts such as the head, neck, and torso. The torso is used to propel the snake robot forward, and the neck and head are used to control the direction of the snake robot. However, this method requires multiple modules to be treated as necks, thus increasing the number of joints and making it difficult to apply to curvature control-based CPG models. Wu et al. [19] used virtual joints based on a CPG network to achieve head stabilization. Although this scheme can effectively ensure the head stabilization of the snake robot, it lacks the feedback of head position information and is not efficient.

The low viewing angle of snake robots makes them susceptible to obstacles, which leads to visual loss and trajectory offset. Kazunori Ohno et al. [20] implemented snake robot trajectory estimation and SLAM reconstruction using an SR-2 RGBD camera and a gyroscope, which has a large yaw angle error, large size, and high power consumption. Liu et al. [21] proposed a bi-directional long short-term memory (BiLSTM) neural network for a snake robot trajectory prediction method and visual localization system, which solves the problem of the visual tracking loss of the snake robot. Kim et al. [22] proposed an error-based adaptive robust RBF-NN-BSC for effective image data reading of the snake robot head system while the snake robot is traveling. Li et al. [23] proposed a system based on the ORB-SLAM3 for stabilized RGB-D SLAM to improve localization accuracy in dynamic environments. Wang et al. [24] develop Dym-SLAM for reconstructing 4D dynamic scenes using rigid moving objects. Despite their contributions, existing methods often face challenges related to generalization, scene awareness, or accurate segmentation.

To solve the above problems, this paper starts from the typical gait motion characteristics of a snake robot, analyzes the gait characteristics of a snake robot and the adaptability with SLAM, and based on this, puts forward a scheme to improve the visual SLAM localization accuracy of the snake robot and the effect of constructing a map: by controlling the joint angle of the motion of the snake robot to improve the form of the snake robot's movement and enhance the stability of the snake robot in the process of movement and thus improve the SLAM effect. On the other hand, from the point of view of the accuracy and robustness of the SLAM algorithm itself, the multi-sensor fusion method is used to enhance the robustness of the visual SLAM algorithm.

The overall structure of this paper is as follows: Section 2 analyzes the adaptability of each gait of the snake robot to the SLAM method. Section 3 proposes the head stability algorithm of the snake robot to reduce the amplitude of sensor shaking based on serpentine gait. Section 4 details a vision-inertial SLAM approach for snake robot motion swaying.

Finally, Section 5 presents the snake robot platform for the related experiments and validates the algorithms proposed.

## 2. Adaptability Study of SLAM for Typical Gaits of Snake-like Robots

Gait generation methods are used to determine the form of motion of the snake robot by controlling the change in joint angles, while the control function method is currently the main method for controlling the gait generation of snake robots, which is characterized by model independence and flexible adjustability. The parameterized gaits are all based on sinusoidal waves in two mutually perpendicular planes: a horizontal plane parallel to the ground and a vertical plane perpendicular to the ground and parallel to the snake's longitudinal axis. When the wave perpendicular to the ground is sent, only the even-numbered joints are involved in the motion and the odd-numbered joints remain stationary; similarly, only the odd-numbered modules are involved in the motion of the transverse wave sent through the robot parallel to the ground [25]. Each joint angle is modeled as a sinusoidal function with a phase shift by describing the gait through joint angles. Then, the angle of the  $n$ th joint at the moment  $t$  can be expressed as follows:

$$\alpha(n, t) = \begin{cases} \beta_{odd} + A_{odd} \sin(\theta_{odd}), n = odd \\ \beta_{even} + A_{even} \sin(\theta_{even} + \delta), n = even \end{cases} \quad (1)$$

$$\theta = \frac{d\theta}{dn}n + \frac{d\theta}{dt}t \quad (2)$$

where  $\beta$  is the offset,  $A$  is the amplitude,  $\theta$  is the frequency,  $\delta$  is the phase shift,  $A_{even}$  and  $A_{odd}$  is the amplitude term,  $d\theta/dn$  determines the shape of the serpentine robot, and  $d\theta/dt$  is the magnitude of the frequency.

Based on the above model, the motion adaptations of four typical gaits, namely, serpentine gait, rolling gait, pedal gait, and sidewinding gait, are analyzed sequentially for ORB-SLAM.

The meandering gait, as the most common two-dimensional locomotion for snake-like robots, is suitable for movement on flat surfaces, propagating body waves from the head to the tail, which is the most widespread mode of movement exhibited by biological snakes. The rolling gait is unique to snake-like robots, including two-dimensional arc rolling within a plane and three-dimensional helix rolling in space, which is not a natural movement for biological snakes. In the pedal gait, the robot makes contact with the ground at only a few points to generate thrust while lifting areas with high curvature to reduce friction. The sidewinding gait combines transverse and longitudinal waves, allowing the robot to both flip and side-shift perpendicular to the body direction.

The form of snake robot motion under serpentine gait is sinusoidal, so the size of the sinusoidal wave amplitude will be an important factor affecting the adaptability of this gait with ORB-SLAM, while the increase in speed will make the snake robot's left and right swing frequency speed up. Therefore, the idea of studying the adaptability of testing the serpentine gait with ORB-SLAM is to change the wave amplitude of the snake robot's motion and the speed of the snake robot's motion under each wave amplitude. Based on the analysis of the error between the estimated value and the true value of the motion, the effect of the size of the wave amplitude and the speed of the motion on the SLAM adaptability of the snake robot is judged.

The snake robot model for the adaptive analysis in this section consists of 35 joints with a length of 89.95 mm, and the robot model's joints are connected orthogonally to each other. Since amplitude  $A$  is related to  $\alpha$ , the change in parameter  $A$  is replaced by changing  $\alpha$ . When the parameters of the fitting curve are five joint lengths and  $\alpha$  is  $0.1\pi$ , the wave amplitude is small during the motion of the snake robot, and the maximum pendulum amplitude is 0.173 m (5.50%). The head stability of the snake robot in this state is better, the feature extraction is continuous, and there is no tracking loss. When  $\alpha = 0.3\pi$ , the swing amplitude of the snake robot increases significantly, with a maximum swing amplitude of

0.416 m (13.2%) and the SLAM method estimates the trajectory to be cluttered during the simulation, then the SLAM method is assumed to be invalid. When  $\alpha = 0.4\pi$ ,  $0.5\pi$ , even if the fitting curve parameter is set to a joint length, it is still impossible to build a map, and the maximum swing is 0.56 m (17.8%) when  $\alpha = 0.4\pi$ , and 0.62 m (19.7%) when  $\alpha = 0.5\pi$ . As the parameter increases, the waveforms of the motion become bigger and bigger, which leads to less or even no feature extraction, and the visual front-end cannot obtain data correlation through feature matching, which leads to the failure of camera position solving.

Under the arc-rolling gait, the snake robot rotates around its axis. Therefore, the idea of testing the localization effect of the snake robot under this gait is to change the size of the fitting curve parameters while keeping other parameters unchanged. Through the simulation analysis, the arc-rolling gait movement speed is lower, the image captured by the camera is clearer, at this time the feature extraction is more stable, and there is no loss of keyframes; pure rotary motion lacks translation-related data excitation, the motion matrix degrees of freedom remain unchanged, while the motion itself decreases in the degrees of freedom, which makes the extra degrees of freedom susceptible to the influence of noise; with the increase in speed, the image captured by the camera appears to be dragged, which is a good idea. With the increase in speed, the image captured by the camera suffers from the phenomenon of dragging, leading to a reduction in the number of features extracted, and the system suffers from the problem of lost trajectory tracking due to insufficient features, which in turn leads to the failure of the ORB-SLAM scheme.

Due to the characteristics of the pedal gait, the snake robot sacrifices stability characteristics while pursuing high motion efficiency. In the vision SLAM method with ORB features as the front-end, the pedal gait causes the camera attached to the head of the snake robot to shake up and down as the snake robot jerks up and down drastically during the motion process, which leads to the failure of the extraction of the ORB feature points and the matching of the failure of the feature. The ORB-SLAM scheme fails in this gait.

In comparison with the arc-rolling gait, the rotation of the sidewinding gait is a rotation around the support point with the surface at different moments rather than a rotation around its axis. The head joints of the snake robot cannot operate smoothly in this gait because the snake robot undergoes a large displacement at the same time as the rotation. The camera shoots the ground for a long time in this gait, and the ORB feature extraction fails due to the lack of texture information and the high speed. So the ORB-SLAM scheme fails in this gait.

From the above analysis, it can be concluded that the pedal gait and the sidewinding gait are poorly adapted to SLAM due to high-frequency jerks and are not suitable for performing the tasks of localization and map building. Therefore, the sidewinding gait and the arc-rolling gait are selected as SLAM research objects. To improve the SLAM adaptability of the snake robot gait, several methods are proposed to address the problems of the large swings of the head joints and the too fast pure rotational speed in the movement process.

### 3. Research on Visual SLAM Method Based on Head Stability Control of Snake-like Robots

For the snake robot's motion gait on the horizontal plane, which refers to the robot's movement within the two-dimensional space defined by the ground surface, the serpentine gait makes head stability control a possibility because of the smooth gait motion characteristics. Therefore, the serpentine gait is selected as the research object to improve head stability during its motion, which in turn improves the accuracy and stability of the visual SLAM during the serpentine gait.

#### 3.1. Modeling of Head Stability Control Based on Link Angle Correction

The snake robot consists of three parts: the "body", "neck", and "head". It is assumed that when the snake robot carries out the serpentine movement, the "body" carries out the undulating movement. During the process, the "head" and "neck" parts are influenced by the fluctuation and thus perform the same undulating movement as the "body". Therefore,

the key to the control principle is to find a way to control the “neck” to eliminate the effect of the “body” fluctuation and stabilize the “head”. Figure 1 shows the model of the snake robot, where the X-axis represents its direction of movement and the Y-axis represents the left side of the movement.

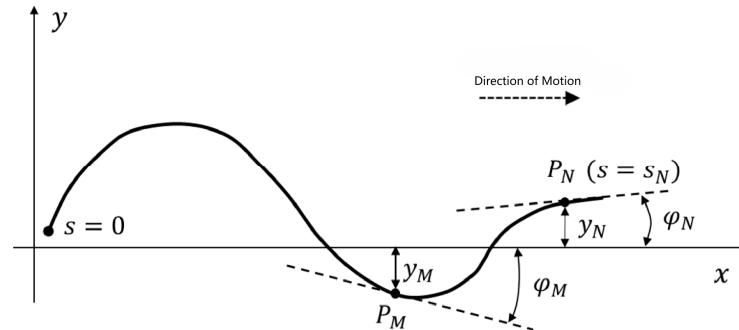


Figure 1. Principle of head stability control.

Where  $P_M$  is the point of demarcation between the “cervical region” and the “neck”, and  $P_N$  is the boundary point between the “head” and the “neck”.

To realize the stability control of the motion direction and position of the snake robot that is equivalent to satisfy  $\dot{\varphi}_N(t) = 0$  and  $\dot{y}_N(t) = 0$ . At this moment, the head of the snake robot is kept in a neutral position, located on the x-axis, namely, the forward direction of the snake robot.

### 3.2. Head Direction Stability Control

The head direction stabilization control method ensures that the head of the snake robot points in one direction as much as possible, rather than constantly and drastically changing direction as the snake swings. Assuming that the motion of point  $P_M$  of the serpentine robot is as shown in Figure 2.  $P_M$  is moving around a curve of radius of  $1/k(s, t)$ , and we can obtain the following:

$$\dot{\varphi}_M(t) = k(s_M, t) \cdot v \tag{3}$$

where  $\dot{\varphi}_M(t)$  is the angle between the X-axis and the tangent line at  $P_M$ ,  $s_M$  is the arc length at  $P_M$ ,  $k(s, t)$  is the curvature at  $s = s, t = t$ , and  $v$  is the velocity at moment  $t$ .

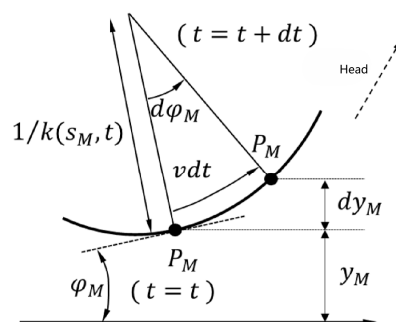


Figure 2. Angular differential variation.

It is assumed during the motion of the snake robot that the snake robot will not slide on the y-axis during its motion, which means that there will be no rotational motion due to sliding. Since the parameters on the right side of the equation are controlled by the motion of the snake robot, we obtain the following:

$$\varphi_N(t) - \varphi_M(t) = \varphi(s_N, t) - \varphi(s_M, t) = \int_{s_M}^{s_N} k(s, t) ds \tag{4}$$

Differentiating Equation (4) and shifting the terms yields Equation (5).

$$\dot{\varphi}_N(t) = \dot{\varphi}_M(t) + \int_{s_M}^{s_N} k(s, t) ds \tag{5}$$

To satisfy head direction stability, namely, make  $\dot{\varphi}_N(t) = 0$ ,  $\dot{\varphi}_M(t)$  can be obtained as follows:

$$\dot{\varphi}_M(t) = - \int_{s_M}^{s_N} k(s, t) ds \tag{6}$$

Using Equations (3) and (6), then Equation (7) can be expressed as follows:

$$\int_{s_M}^{s_N} k(s, t) ds = -k(s_M, t) \cdot v \tag{7}$$

Equation (7) is the control theory for the head stability of the snake robot in the serpentine gait described in this chapter. From Equation (4), when fitting the head direction of the snake robot to the motion direction at the beginning of the motion,  $k(s, t)$  should also satisfy the equation as follows:

$$\int_{s_M}^{s_N} k(s, t) ds = -\dot{\varphi}_M(t) \tag{8}$$

Therefore, for simplicity of calculation, it is assumed that when  $s_M < s < s_N$ , the curvature is a constant, namely  $k(s, t) = c_M(t)$ . The “neck” is a circular arc in this case. At any moment, the arc of the circle and the body part of the sine wave articulation can point the direction of the head of the snake robot to the direction of its movement. According to the curvature of the arc, to solve the formula, the ratio of the angle of the center of the circle to the length of the arc and combined with Equation (7) can be seen when Equation (9) is valid and is then a constant. The  $\dot{c}_M(t)$  is expressed as follows:

$$\dot{c}_M(t) = - \frac{k(s_M, t)}{l_M} t \tag{9}$$

Thus, the curvature of the snake curve in serpentine gait is obtained as a segmented function:

$$k(s, t) = \begin{cases} -ab \sin(bs + \sigma(t)), & s < s_M \\ -\frac{\dot{\varphi}_M(t)}{l_N}, & s_M < s < s_N \end{cases} \tag{10}$$

where  $b = 2\pi/L$ . Finally, the mathematical formulation for controlling the head stability of the snake robot in the serpentine gait is obtained by integration as follows:

$$\varphi(s, t) = \begin{cases} \alpha \cos(bs + \sigma(t)), & s < s_M \\ (1 - \frac{s-s_M}{l_M})\alpha \cos(bs + \sigma(t)), & s_M < s < s_N \end{cases} \tag{11}$$

where  $\sigma(t)$  is the function of time.

From Equation (9), the joint position of the  $i$ th joint can be expressed as follows:

$$Angle = \varphi((2i) \cdot l_0) - \varphi((2i - 2) \cdot L_0) \tag{12}$$

### 3.3. Head Position Stability Control

The head position stabilization control method can build on the algorithm in the previous section to minimize the sensor wobble caused by the snake robot’s head swinging from side to side during its serpentine motion. As shown in Figure 2, point  $P_M$  of the

“neck” moves with speed  $v$  in the direction of the tangent point  $M$ . The case of satisfying the assumptions made in the previous section,  $\dot{y}_M(t)$  can be expressed as follows:

$$\dot{y}_M(t) = -\sin \varphi_M(t) \cdot v \tag{13}$$

Substitute Equation (6) into Equation (13).

$$y_N(t) = y_M(t) + \int_{s_M}^{s_N} \sin(k(s,t)ds + \varphi_M(t))ds \tag{14}$$

Derivate Equation (14) and substitute Equation (6) into Equation (13).

$$\int_{s_M}^{s_N} \sin(\int_s^{s_H} k(s,t)ds)ds = -\sin(\int_{s_M}^{s_N} k(s,t)ds) \cdot v \tag{15}$$

From the above derivation, it can be seen that any curvature satisfying Equation (15) can stabilize the position and direction of the head of the snake robot, but the complexity of Equation (15) leads to the problem of complicated and difficult curvature calculation. In the previous subsection, to simplify the calculation, it is assumed that the “neck” of the snake robot is a circular arc. So it is assumed that the snake robot consists of two circular arcs with openings at both ends in the opposite direction, and Equation (15) is linearized. Thus, when  $s > s_M$ , the curvature expression is divided into two parts:

$$k(s,t) = \begin{cases} c_M(t) + f_M(t), s_M < s < s_M + l_N/2 \\ c_M(t) - f_M(t), s > s_M + l_N/2 \end{cases} \tag{16}$$

Substitute Equation (15) into Equation (16). The differential expression for  $f(t)$  can be expressed as follows:

$$\dot{f}_M(t) = \frac{4c_M(t)}{l_N} - 2\frac{k(s_M,t)}{l_N} \tag{17}$$

Integrating Equation (17) yields an expression for the curvature. To facilitate the computation of the curvature, in this paper we use the discretization method to solve  $f(t)$ .

$$f_M(t_{k+1}) = f_M(t_k) + \dot{f}_M(t_k) \cdot \delta t \tag{18}$$

Finally, the curvature of the head direction and position stability of the snake robot in the serpentine gait is obtained into three segments as in Equation (19).

$$k(s,t) = \begin{cases} -\alpha b \sin(bs + \sigma(t)), s < s_M \\ c_M(t) + f_M(t), s_M < s < s_M + l_N/2 \\ c_M(t) - f_M(t), s > s_M + l_N/2 \end{cases} \tag{19}$$

Integrate over the curvature.  $\varphi(s,t)$  can be expressed as follows:

$$\varphi(s,t) = \begin{cases} \alpha \cos(bs + \sigma(t)), s < s_M \\ (1 - \frac{s-s_M}{l_N})\alpha \cos(bs + \sigma(t)) + f_M(t)(s - s_M), s_M < s < s_M + l_N/2 \\ (1 - \frac{s-s_M}{l_N})\alpha \cos(bs + \sigma(t)) + f_M(t)(s - s_M - l_N/2), s > s_M + l_N/2 \end{cases} \tag{20}$$

Similarly, the joint position of the  $i$ th joint can be obtained as follows:

$$Angle = \varphi((2i) \cdot l_0) - \varphi((2i - 2) \cdot l_0) \tag{21}$$

#### 4. Research on Visual-Inertial Tightly Coupled SLAM Method for Snake-like Robots

From the analysis in Section 2, it can be seen that the snake robot’s serpentine gait and arc-rolling gait suffer from poor head stability due to large head joint swing and large SLAM drift due to too fast rotation speed, respectively. In this chapter, we will improve the



adaptability of the snake robot’s typical gait with SLAM from the perspective of improving the robustness of the algorithm itself. Inertial Measurement Units (IMU) complement the vision sensors well, so this chapter will incorporate new sensor-IMUs within the vision SLAM framework to act as a constraint on the vision estimation [26,27]. Firstly, the vision-inertial SLAM is used as an observation; then, the kinematic models of the snake robot’s serpentine gait and arc-rolling gait are utilized as the motion models, and finally, the IESKF algorithm is utilized for estimating the final results, thus improving the robustness of the algorithm.

#### 4.1. Visual–Inertial-Based Observation Models

When the camera is only the sensor of the SLAM system, the surrounding waypoints will only constrain the camera, and at this time, it is only necessary to construct the reprojection error as a least-squares problem to find the optimal solution. However, a trajectory will also be generated based on the IMU data as well as the equations of motion as the carrier is in motion when an IMU is added. Since both the camera and the IMU are solidly attached to a carrier, there is a connection between the trajectory information estimated by each of the two sensors, and this connection can be determined by the external reference between the camera and the IMU.

The sampling frequency of the IMU is higher than that of the camera, and the bit position estimated by the camera is usually the bit position between the keyframes, so the IMU observations between two frames need to be accumulated. When constructing the optimization problem of the vision-inertial system, the visual information, along with the IMU zero-bias, velocity, and gravity direction, need to be used as the optimization vectors, and while iteratively updating these vectors, the information of the key frames’ position, velocity, and so on, will continuously be performed as integration operations, which wastes the computational resources, and therefore, the pre-integration of the IMU is needed, i.e., the terms that are invariant at each iteration of the optimization are extracted, and the PVQ is obtained for each cycle. Position, velocity, and attitude increments are measured against other estimates of PVQ increments obtained by non-IMU means, and then the residuals of the PVQ increments are obtained. This residual is then used to construct a cost function to iteratively update and optimize the PVQ for each node.

The measured values of the gyroscope and accelerometer are first constructed as follows:

$$\begin{aligned} \omega_M(t) &= \omega_b(t) + b_g(t) + n_g \\ a_M(t) &= a_b(t) + b_a(t) + R_w^{b_t} g^w + n_a \end{aligned} \tag{22}$$

where  $\omega_m(t)$  is the angular velocity measurement of the gyroscope at moment  $t$ ,  $\omega_b(t)$  is the true value of the gyroscope angular velocity at moment  $t$ ,  $b_g(t)$  is the random zero bias of the gyroscope at moment  $t$ ,  $n_g$  is the gyroscopic white noise,  $a_m(t)$  is the measured acceleration of the gyroscope at moment  $t$ ,  $a_b(t)$  is the true value of the acceleration of the gyroscope at moment  $t$ ,  $b_a(t)$  is the random zero bias of the accelerometer at moment  $t$ ,  $R_w^{b_t}$  is the rotation matrix for conversion from the world coordinate system to the IMU coordinate system,  $g^w$  is the acceleration in the world coordinate system, and  $n_a$  is the accelerometer white noise.

Integrate the PVQ of frame  $k$ , and the PVQ of frame  $(k + 1)$  can be expressed as follows:

$$\begin{aligned} p_{b_{k+1}}^w &= p_{b_k}^w + v_{b_k}^w \Delta t_k + \iint_{t \in [t_k, t_{k+1}]} \left\{ R_{b_t}^w [a_m(t) - b_a(t) - n_a] - g^w \right\} dt^2 \\ v_{b_{k+1}}^w &= v_{b_k}^w + \int_{t_k}^{t_{k+1}} \left\{ R_{b_t}^w [a_m(t) - b_a(t) - n_a] - g^w \right\} dt \\ q_{b_{k+1}}^w &= q_{b_k}^w \otimes \int_{t_k}^{t_{k+1}} \frac{1}{2} \Omega(\omega_m(t) - b_g(t) - n_g) q_t^{b_k} dt \end{aligned} \tag{23}$$



where  $w$  is the world coordinate system,  $b_k$  and  $b_{k+1}$  are the different image sampling moments,  $\otimes$  denotes the operations between rotating quaternions, and  $q_t^{b_k}$  denotes the transformation matrix of the position from moment  $t$  to moment  $b_k$ .

$$\Omega(\omega) = \begin{bmatrix} -[\omega]_{\times} & \omega \\ -\omega^T & 0 \end{bmatrix}, [\omega]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (24)$$

Perform the left-multiplication of Equation (24).

$$\begin{aligned} R_w^{b_k} p_{b_{k+1}}^w &= R_w^{b_k} (p_{b_k}^w + v_{b_k}^w \Delta t_k - \frac{1}{2} g^w \Delta t_k^2) + \alpha_{b_{k+1}}^{b_k} \\ R_w^{b_k} v_{b_{k+1}}^w &= R_w^{b_k} (v_{b_k}^w - g^w \Delta t_k) + \beta_{b_{k+1}}^{b_k} \\ q_w^{b_k} \otimes q_{b_{k+1}}^w &= \gamma_{b_{k+1}}^{b_k} \end{aligned} \quad (25)$$

where  $\alpha_{b_{k+1}}^{b_k}$ ,  $\beta_{b_{k+1}}^{b_k}$ , and  $\gamma_{b_{k+1}}^{b_k}$  are pre-integral, and the expression can be expressed as follows:

$$\begin{aligned} \alpha_{b_{k+1}}^{b_k} &= \iint_{t \in [t_k, t_{k+1}]} \{ R_{b_l}^{b_k} [a_m(t) - b_a(t) - n_a] \} dt^2 \\ \beta_{b_{k+1}}^{b_k} &= \int_{t_k}^{t_{k+1}} \{ R_{b_l}^{b_k} [a_m(t) - b_a(t) - n_a] \} dt \\ \gamma_{b_{k+1}}^{b_k} &= \int_{t_k}^{t_{k+1}} \frac{1}{2} \Omega(\omega(t) - b_g(t) - n_g) \gamma_t^{b_k} dt \end{aligned} \quad (26)$$

The IMU pre-integration error term between image sampling from moment  $b_k$  to moment  $b_{k+1}$  is  $r_B(\hat{z}_{b_{k+1}}^{b_k}, \chi)$

$$r_B(\hat{z}_{b_{k+1}}^{b_k}, \chi) = \begin{bmatrix} \delta \alpha_{b_{k+1}}^{b_k} \\ \delta \beta_{b_{k+1}}^{b_k} \\ \delta \gamma_{b_{k+1}}^{b_k} \\ \delta b_a \\ \delta b_w \end{bmatrix} = \begin{bmatrix} R_w^{b_k} (p_{\delta_{k+1}}^w - p_{\delta_k}^w - v_{\delta_k}^w \Delta t_k + \frac{1}{2} g^w \Delta t_k^2) - \alpha_{b_{k+1}}^{b_k} \\ R_w^{b_k} (v_{b_{k+1}}^w - v_{b_k}^w - g^w \Delta t_k) - \beta_{b_{k+1}}^{b_k} \\ 2[q_w^{b_k} \otimes q_{b_{k+1}}^w (\gamma_{b_{k+1}}^{b_k})^{-1}]xyz \\ b_{a_{k+1}} - b_{a_k} \\ b_{g_{k+1}} - b_{g_k} \end{bmatrix} \quad (27)$$

where  $\hat{z}_{b_{k+1}}^{b_k} = (\alpha_{b_{k+1}}^{b_k}, \beta_{b_{k+1}}^{b_k}, \gamma_{b_{k+1}}^{b_k})$  is the IMU integral term, and  $\chi$  is the state variables to be optimized.

#### 4.2. Snake Robot Motion Estimation

Suppose for the serpentine gait, the snake robot is equivalently composed of  $N$  connecting rods with length of  $2l$  and uniform mass distribution  $m$  and connected at the head and tail. The center of mass inertia coordinates of the whole snake robot are expressed as follows:

$$P = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \frac{1}{Nm} \sum_{i=1}^n mx_i \\ \frac{1}{Nm} \sum_{i=1}^n my_i \end{bmatrix} = \frac{1}{N} \begin{bmatrix} e^T X \\ e^T Y \end{bmatrix} \quad (28)$$

The serpentine robot linkage joint constraints in matrix form can be expressed as follows:

$$\begin{aligned} DX + lA \cos \theta &= 0 \\ DY + lA \sin \theta &= 0 \end{aligned} \quad (29)$$

where  $D$  is subtraction of matrices,  $X = [x_1, \dots, x_n]^T$ ,  $Y = [y_1, \dots, y_n]^T$ , and  $\theta = [\theta_1, \dots, \theta_n]^T$ . Combining Equations (28) and (29) yields Equation (30)

$$\begin{aligned} TX &= \begin{bmatrix} -lA \cos \theta \\ p_x \end{bmatrix} \\ TY &= \begin{bmatrix} -lA \sin \theta \\ p_y \end{bmatrix} \end{aligned} \tag{30}$$

where  $T = \begin{bmatrix} D \\ \frac{1}{N}e^T \end{bmatrix} \in R^{N \times N}$ , then  $T^{-1} = \begin{bmatrix} D^T(DD^T)^{-1} \\ e \end{bmatrix}^T$ .

The linear velocity of the connecting rod can be expressed as follows:

$$\begin{aligned} \dot{X} &= lK^T S_\theta \dot{\theta} + e \dot{p}_x \\ \dot{Y} &= -lK^T C_\theta \dot{\theta} + e \dot{p}_y \end{aligned} \tag{31}$$

where  $S_\theta = \text{diag}(\sin \theta) \in R^{N \times N}$ ,  $C_\theta = \text{diag}(\cos \theta) \in R^{N \times N}$ .

According to the motion characteristics of the arc-rolling gait, the motion of the snake robot along the forward direction is divided into two parts; one is the rolling of the joints in contact with the ground, and its velocity can be expressed as follows:

$$v_{wheel} = \omega \cdot r \tag{32}$$

where  $r$  is the radius of the robot module. The other is the forward swing of the elevated joint, the motion of which can be expressed in polar coordinates with the value of:

$$\begin{cases} r_l(t) = l \cdot \sin(\frac{1}{2}\alpha(t)) = l \cdot \sin(\frac{A}{2} \cos(\omega t)) \\ \theta(t) = \omega t \end{cases} \tag{33}$$

where  $l$  is the length of the robot joint. Then, the oscillating velocity of the joint in the x-axis and z-axis can be expressed as follows:

$$\begin{cases} v_{swing}^x = \frac{dx}{dt} = \frac{d}{dt}(r_l(t) \cdot \cos \theta(t)) \\ v_{swing}^z = \frac{dz}{dt} = \frac{d}{dt}(r_l(t) \cdot \sin \theta(t)) \end{cases} \tag{34}$$

The velocity of motion of the joint along the target direction is summarized as follows:

$$v(t) = \begin{cases} v_{wheel} + v_{swing}^x, t \in (0, T/4] \\ v_{wheel}, t \in (T/4, T/2] \end{cases} \tag{35}$$

Then, the explicit expression for the motion velocity of the robot's joint points can be obtained as follows:

$$\begin{aligned} v(t) &= \omega \cdot r + \omega \cdot l \cdot [\frac{A}{2} \cdot \cos(\frac{A}{2} \cos(\omega t)) \cdot (-\sin(\omega t)) \cdot \cos(\omega t) \\ &\quad + \sin(\frac{A}{2} \cos(\omega t)) \cdot (-\sin(\omega t))] \end{aligned} \tag{36}$$

#### 4.3. State Fusion Estimation for Snake Robots Based on IESKF

First, we define the state vector of the snake robot, which can be expressed as follows:

$$x = [p_x, p_y, \theta, v, \omega]^T \tag{37}$$

where  $p_x$  and  $p_y$  is the coordinates of the robot's position in a two-dimensional plane,  $\theta$  is the angle of the robot's forward direction for the global coordinate system,  $v$  is the linear velocity, and  $\omega$  is the angular velocity.

Considering the motion characteristics of the snake robot, the state transfer model can be expressed as:

$$x_{k+1} = f(x_k, u_k, \Delta t) + w_k \tag{38}$$

where  $f$  is the state transition function,  $u_k$  is the control input,  $\Delta t$  is the time interval, and  $w_k$  is the process noise.

The observation model combines data from vision and IMU and can be expressed as follows:

$$z_k = h(x_k) + v_k \tag{39}$$

where  $z_k$  is the observation vector,  $h$  is the observation function, and  $v_k$  is the observational noise.

The IESKF algorithm consists of two main steps: prediction and calibration. In the prediction step, the state at the next moment is predicted based on the current state estimation and control inputs:

$$\begin{aligned} \hat{x}_{k|k-1} &= f(\hat{x}_{k-1|k-1}, u_{k-1}, \Delta t) \\ P_{k|k-1} &= FP_{k-1|k-1}F^T + Q \end{aligned} \tag{40}$$

In the calibration step, the state estimates are updated using the observed data:

$$\begin{aligned} y_k &= z_k - h(\hat{x}_{k|k-1}) \\ S_k &= H_k P_{k|k-1} H_k^T + R \\ K_k &= P_{k|k-1} H_k^T S_k^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k y_k \\ P_{k|k} &= (I - K_k H_k) P_{k|k-1} \end{aligned} \tag{41}$$

where  $F$  is the state transfer Jacobi matrix,  $H_k$  is the observation of the Jacobi matrix,  $Q$  is the process noise covariance matrix,  $R$  is the observation noise covariance matrix,  $K_k$  is the Kalman gain,  $y_k$  is the residual vector, and  $I$  is the identity matrix.

### 5. Simultaneous and Experimental Research on SLAM for Snake-like Robots

#### 5.1. Simulation Verification

In this paper, we use CoppeliaSim software (V4.0.0) as the robot simulation platform and directly use CoppeliaSim to build the snake robot model. The snake robot model built in this paper consists of 35 joints with a length of 89.95 mm, and the robot model's joints are connected orthogonally to each other.

The serpentine motion simulation experiments were conducted by taking  $\alpha$  as  $0.3\pi$ ,  $0.5\pi$ , and  $0.7\pi$ , respectively, and the comparison results of the no algorithm, stability algorithm, and fusion algorithm serpentine motion SLAM simulations are shown in Figure 3:

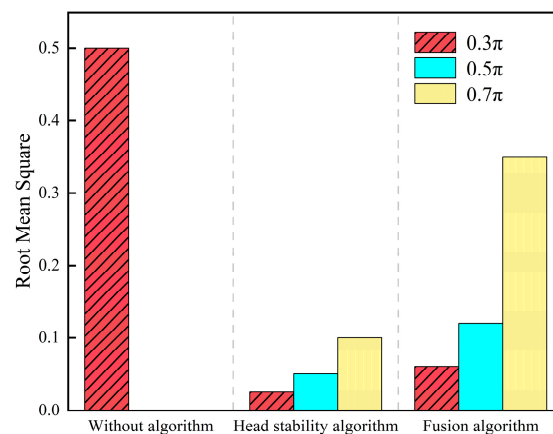
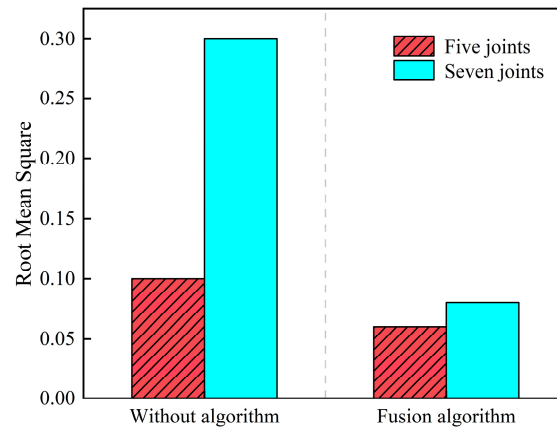


Figure 3. Comparison of SLAM simulation under the serpentine gait.

The arc-rolling motion simulation experiment is divided into two groups; in the case of keeping other parameters unchanged, the first group is set to five snake robot joint lengths, and the second group is set to seven joint lengths. The simulation comparison results are shown in Figure 4.

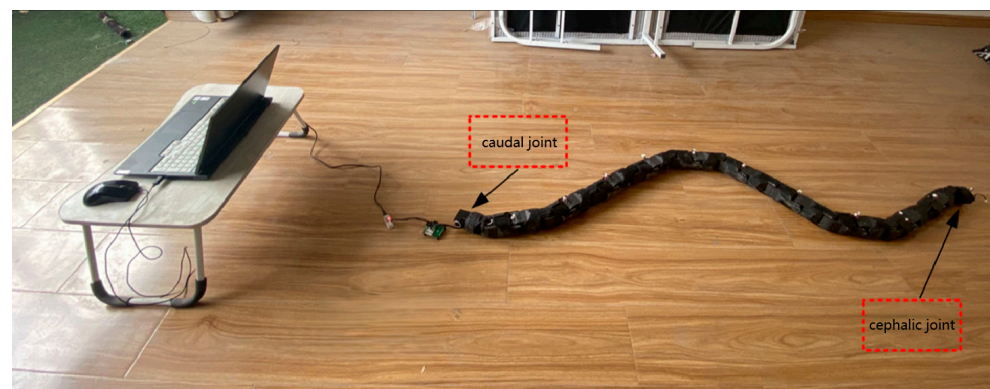


**Figure 4.** Comparison of SLAM simulation under the arc-rolling gait.

## 5.2. Experiment Verification

### 5.2.1. Experimental Platform Introduction

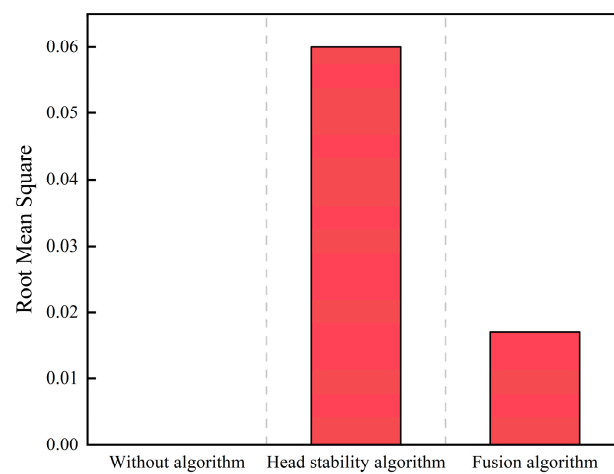
The snake robot control system is co-built with an orthogonal snake robot as the ontological structure, an ROS open source robot system as the overall architecture, and two-level distribution of the upper and lower computers. The whole system consists of an orthogonal snake robot body, a perception module, and other hardware components. The orthogonal snake robot is alternately composed of orthogonal joints and pitch joints, with a total of 24 joints, a joint length of 96 mm, and a diameter of 64 mm. Each joint is covered with rubber to increase friction, and the prototype of the snake robot is shown in Figure 5. The sensing module mainly integrates the depth camera, core module, and battery pack. The high-resolution depth camera is used to collect image information and provide raw data for SLAM; the core module serves as the core of the perception module and provides arithmetic support for the realization of the SLAM scheme; the battery pack supplies power to the head module. Among them, the Intel RealSense l515 camera is selected as the vision sensor in this paper, which has a very small size and weight and can capture high-quality color images. The NVIDIA Jetson Xavier NX core board is used as the embedded computing platform, and the Y-C7 is used as the industrial-grade carrier board that is used in conjunction with the NVIDIA Jetson Xavier NX core module, which can capture high-quality color images. The Y-C7 is an industrial-grade carrier board for use with the NVIDIA Jetson Xavier NX core module, featuring low cost and small size.



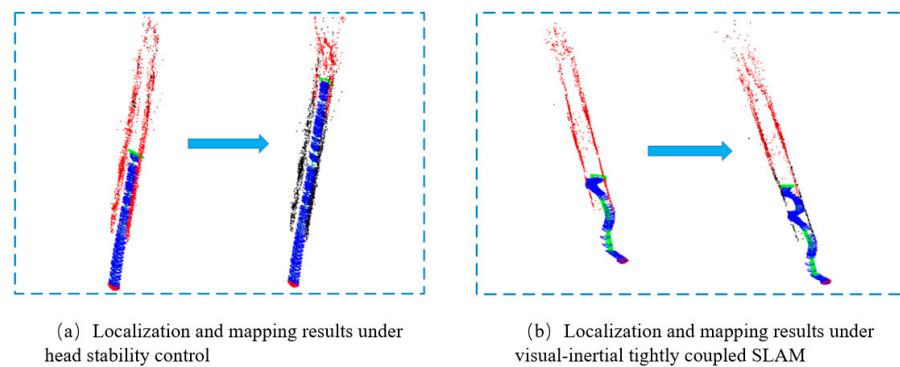
**Figure 5.** The experimental system of the snake robot.

### 5.2.2. Experiments on Simultaneous Localization and Mapping for Snake-like Robots in the Serpentine Gait

The experiments on simultaneous localization and mapping of the snake robot under the serpentine gait are divided into three groups of cases, namely, the condition without head stability control, the condition with head stability control added, and the vision-inertia tight coupling method. The amplitude parameter  $\alpha = 0.5\pi$  is set for all three groups, and the simulation analysis shows that the vision algorithm fails under this amplitude parameter. The first group of experiments can be compared with the second and third groups of experiments: the first and the second groups control the effect of the head stability control method on the improvement of the SLAM accuracy of the snake robot; the first and the third groups control the effect of the tightly coupled vision-inertia SLAM method on the improvement of the SLAM accuracy of the snake robot and the robustness of the algorithm itself. The results of the experimental comparison are shown in Figures 6 and 7:



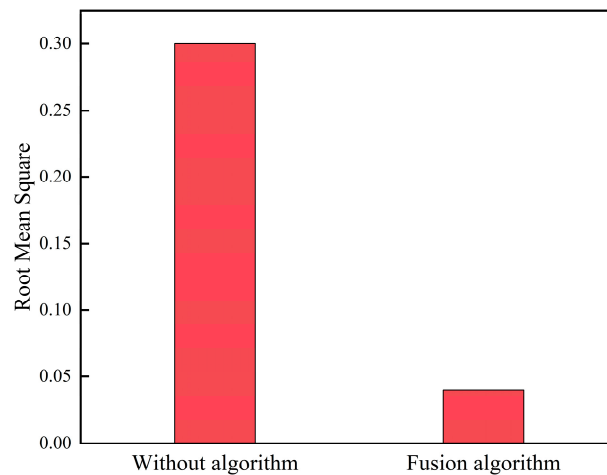
**Figure 6.** Comparison of localization and mapping under the serpentine gait.



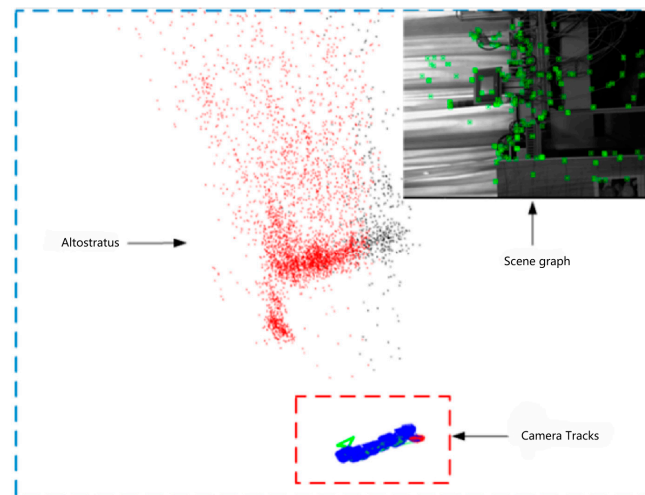
**Figure 7.** Comparison of localization and mapping under the two methods.

### 5.2.3. Experiments on Simultaneous Localization and Mapping for Snake-like Robots in the Arc-Rolling Gait

The analysis indicates that when the fitting curve parameter is 7 joint lengths, the estimated trajectory of the snake robot under the arc-rolling gait has a large drift, so this set of experiments set the fitting curve parameter of the snake robot to 7 joint lengths. The experiment is divided into two groups; in the case of keeping other parameters unchanged, the first group is set to without algorithm, and the second group is set to fusion algorithm. The experimental results are shown in Figures 8 and 9:



**Figure 8.** Comparison of localization and mapping under the arc-rolling gait.



**Figure 9.** Localization and mapping results under the arc-rolling gait.

The trajectory values show that in the arc-rolling gait, the snake robot moved 1.85 m in the x-axis direction and rotated around the axis in the y-axis direction. The mean root mean square (RMS) of the mean difference in this set of experiments is 0.04, which indicates that the snake robot has high accuracy in localization and map building under this method.

## 6. Conclusions

In this paper, we propose a SLAM method based on the head stabilization control method and the fusion of kinematic estimation of the snake robot to address the problem of the easy failure of localization and mapping caused by the large head joint angle and fast rotation speed during the motion of the snake robot, which can effectively improve the accuracy of the localization and mapping of the snake robot. Through simulation and physical experimentation, the head stability control method, under the undulating gait and with a waveform amplitude of  $0.7\pi$ , achieved an average root mean square (RMS) error of 0.058 in localization and map building, significantly outperforming the SLAM scheme that fails under an amplitude of  $0.3\pi$  without head stability control. The visual-inertial tightly coupled SLAM method demonstrated average root mean square (RMS) errors of 0.1 for the undulating gait and 0.08 for the arc-rolling gait, both below 0.1, indicating that this approach effectively enhances the robustness of the algorithm. The SLAM algorithm based on the head stability method can reduce the root-mean-square (RMS) of the error from 0.3 to 0.05 when the parameter  $\alpha = 0.5\pi$ . The simulation and experimental validation show that the proposed method can effectively improve the accuracy of the localization and mapping



method of the snake robot and broaden the scope of the localization and mapping method under the serpentine gait. However, this paper only investigates the SLAM adaptability of the snake robot's motion gait in a two-dimensional plane and the application scenarios are relatively simple. The validation of the snake robot's three-dimensional gait and more complex scenarios will be carried out at a later stage.

**Author Contributions:** Methodology, C.T.; validation, M.L. and H.T.; supervision, D.B.; writing—original draft, M.S.; writing—review and editing, Z.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the National Nature Science Foundation of China under Grant 62073328, 62233011 and 52005496, the Jiangsu Province Key Research and Development Program under Grant BE2023008, and the China Postdoctoral Science Foundation under Grant 2022M723392.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

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

## References

- Zhou, X.; Zhang, Y.; Qiu, Z.; Shan, Z.; Cai, S.; Bao, G. Locomotion control of a Rigid-Soft coupled snake robot in multiple environments. *Biomim. Intell. Robot.* **2024**, *4*, 100148. [[CrossRef](#)]
- Liu, J.; Tong, Y.; Liu, J. Review of Snake Robots in Constrained Environments. *Robot. Auton. Syst.* **2021**, *141*, 103785. [[CrossRef](#)]
- Borenstein, J.; Borrell, A. The OmniTread OT-4 serpentine robot. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 1766–1767.
- Liljebäck, P.; Pettersen, K.Y.; Stavdahl, Ø. A snake robot with a contact force measurement system for Obstacle-Aided loco-motion. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 683–690.
- Wright, C.; Buchan, A.; Brown, B.; Geist, J.; Schwerin, M.; Rollinson, D.; Tesch, M.; Choset, H. Design and architecture of the unified modular snake robot. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 4347–4354.
- Makhubela, J.K.; Zuva, T.; Agunbiade, O.Y. A review on vision simultaneous localization and mapping (VSLAM). In Proceedings of the 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC), Mon Tresor, Mauritius, 6–7 December 2018; pp. 1–5.
- Kohlbrecher, S.; von Stryk, O.; Meyer, J.; Klingauf, U. A flexible and scalable SLAM system with full 3D motion estimation. In Proceedings of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 1–5 November 2011; pp. 155–160.
- Khan, S.; Wollherr, D.; Buss, M. Modeling laser intensities for simultaneous localization and mapping. *IEEE Robot. Autom. Lett.* **2016**, *1*, 692–699. [[CrossRef](#)]
- Chavan, P.; Murugan, M.; Unnikannan, E.V.V.; Singh, A.; Phadatar, P. Modular snake robot with mapping and navigation: Urban search and rescue (USAR) robot. In Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Pune, India, 26–27 February 2015; pp. 537–541.
- Ponte, H.; Queenan, M.; Gong, C.; Mertz, C.; Travers, M.; Enner, F.; Hebert, M.; Choset, H. Visual sensing for developing autonomous behavior in snake robots. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 2779–2784.
- Nivethika, S.D.; Nivethetha, T.; Priyadharshini, P.; Nithyasri, V.T.; SenthilPandian, M.; Sivaprasad, R. Design and development of pipe inspection snake locomotion robot. In Proceedings of the 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 8–9 December 2022; pp. 1–5.
- Yang, W.; Bajenov, A.; Shen, Y. Improving Low-Cost Inertial-Measurement-Unit (IMU)-Based motion tracking accuracy for a biomorphic Hyper-Redundant snake robot. *Robot. Biomim.* **2017**, *4*, 16. [[CrossRef](#)] [[PubMed](#)]
- Billah, M.M.; Khan, M.R. Smart inertial Sensor-Based navigation system for flexible snake robot. In Proceedings of the 2014 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Kuala Lumpur, Malaysia, 25 November 2014; pp. 1–5.
- Gong, C.; Travers, M.; Astley, H.C.; Goldman, D.I.; Choset, H. Limbless locomotors that turn in place. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 3747–3754.
- Yamada, H.; Mori, M.; Hirose, S. Stabilization of the head of an undulating Snake-like robot. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October 2007–2 November 2007; pp. 3566–3571.

16. Bing, Z.; Cheng, L.; Huang, K.; Jiang, Z.; Chen, G.; Röhrbein, F.; Knoll, A. Towards autonomous locomotion: Slithering gait design of a Snake-like robot for target observation and tracking. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2698–2703.
17. Wu, X.; Ma, S. Neurally controlled steering for Collision-Free behavior of a snake robot. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 2443–2449. [[CrossRef](#)]
18. Qiao, G.; Song, G.; Zhang, Y.; Zhang, J.; Li, Y. Head stabilization control for Snake-like robots during lateral undulating locomotion. In Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), Bali, Indonesia, 5–10 December 2014; pp. 392–397.
19. Nor, N.M.; Ma, S. CPG-Based locomotion control of a Snake-like robot for obstacle avoidance. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 347–352.
20. Ohno, K.; Nomura, T.; Tadokoro, S. Real-Time robot trajectory estimation and 3D map construction using 3D camera. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; 2006; pp. 5279–5285.
21. Liu, X.; Wei, W.; Li, Y.; Gao, Y.; Xiao, Z.; Lin, G. Trajectory prediction and visual localization of snake robot based on BiLSTM neural network. *Appl. Intell.* **2023**, *53*, 27790–27807. [[CrossRef](#)]
22. Kim, S.-J.; Jin, M.; Suh, J.-H. A study on the design of Error-Based adaptive robust RBF neural network Back-Stepping controller for 2-DOF snake robot's head. *IEEE Access* **2023**, *11*, 23146–23156. [[CrossRef](#)]
23. Li, X.; Wu, H.; Chen, Z. Dynamic objects recognizing and masking for RGB-D SLAM. In Proceedings of the 2021 4th International Conference on Intelligent Autonomous Systems (ICoIAS), Wuhan, China, 14–16 May 2021; pp. 169–174.
24. Wang, C.; Luo, B.; Zhang, Y.; Zhao, Q.; Yin, L.; Wang, W.; Su, X.; Wang, Y.; Li, C. DymSLAM: 4D dynamic scene reconstruction based on geometrical motion segmentation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 550–557. [[CrossRef](#)]
25. Tesch, M.; Lipkin, K.; Brown, I.; Hatton, R.; Peck, A.; Rembisz, J.; Choset, H. Parameterized and scripted gaits for modular snake robots. *Adv. Robot.* **2009**, *23*, 1131–1158. [[CrossRef](#)]
26. Servières, M.; Renaudin, V.; Dupuis, A.; Antigny, N. Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking. *J. Sens.* **2021**, *1*, 2054828. [[CrossRef](#)]
27. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *11*, 1874–1890. [[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.