



Article

LFVB-BioSLAM: A Bionic SLAM System with a Light-Weight LiDAR Front End and a Bio-Inspired Visual Back End

Ruilan Gao ¹, Zeyu Wan ¹, Sitong Guo ¹, Changjian Jiang ¹ and Yu Zhang ^{1,2,*}

¹ State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China; gaoruilan@zju.edu.cn (R.G.); zeyuwan@zju.edu.cn (Z.W.); guositong@zju.edu.cn (S.G.); changjianjiang@zju.edu.cn (C.J.)

² Key Laboratory of Collaborative Sensing and Autonomous Unmanned Systems of Zhejiang Province, Hangzhou 310027, China

* Correspondence: zhangyu80@zju.edu.cn

Abstract: Simultaneous localization and mapping (SLAM) is one of the crucial techniques applied in autonomous robot navigation. The majority of present popular SLAM algorithms are built within probabilistic optimization frameworks, achieving high accuracy performance at the expense of high power consumption and latency. In contrast to robots, animals are born with the capability to efficiently and robustly navigate in nature, and bionic SLAM algorithms have received increasing attention recently. Current bionic SLAM algorithms, including RatSLAM, with relatively low accuracy and robustness, tend to fail in certain challenging environments. In order to design a bionic SLAM system with a novel framework and relatively high practicality, and to facilitate the development of bionic SLAM research, in this paper we present LFVB-BioSLAM, a bionic SLAM system with a light-weight LiDAR-based front end and a bio-inspired vision-based back end. We adopt a range flow-based LiDAR odometry as the front end of the SLAM system, providing the odometry estimation for the back end, and we propose a biologically-inspired back end processing algorithm based on the monocular RGB camera, performing loop closure detection and path integration. Our method is verified through real-world experiments, and the results show that LFVB-BioSLAM outperforms RatSLAM, a vision-based bionic SLAM algorithm, and RF2O, a laser-based horizontal planar odometry algorithm, in terms of accuracy and robustness.

Keywords: simultaneous localization and mapping (SLAM); bionic robotics; loop closure detection; path integration



Citation: Gao, R.; Wan, Z.; Guo, S.; Jiang, C.; Zhang, Y. LFVB-BioSLAM: A Bionic SLAM System with a Light-Weight LiDAR Front End and a Bio-Inspired Visual Back End. *Biomimetics* **2023**, *8*, 410. <https://doi.org/10.3390/biomimetics8050410>

Academic Editors: Mingguo Zhao and Biao Hu

Received: 14 July 2023

Revised: 17 August 2023

Accepted: 1 September 2023

Published: 5 September 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

The last 3 decades have witnessed remarkable progress in the research area of autonomous robot navigation. Simultaneous localization and mapping (SLAM), as one of the fundamental techniques utilized by robots to perform navigation tasks, refers to the process of constructing the map of the unknown environment that the robot is exploring and estimating the pose of the robot within it simultaneously [1–3]. As illustrated in the survey [4], the architecture of a SLAM system can be divided into two major parts: the front end and the back end. The front end receives sensor data and converts it into models suitable for robot state estimation, while the back end conducts inference on the abstracted data generated by the front end.

Currently, most popular SLAM algorithms typically conduct the back-end inference under probabilistic filtering or nonlinear optimization framework [5]. The sensor modalities they utilize to obtain environmental information include LiDARs [6], radars [7], cameras (e.g., standard RGB cameras [8], RGB-D cameras [9], infrared cameras [10] and event-based cameras [11]), and inertial measurement units (IMUs). Moreover, multi-sensor fusion approaches have attracted growing attention.

The LiDAR and radar enhance the reliability of spatial distance information by actively transmitting and receiving signals (laser beams and radio waves), thereby improving the accuracy of the SLAM algorithm at the expense of higher power consumption. LiDAR-based SLAM algorithms mainly include LOAM [12], LIO-SAM [6], and Fast-LIO2 [13].

SLAM algorithms based on standard RGB cameras use the data association between multiple frames of images to accomplish robot ego-motion estimation and environment mapping. Representative visual SLAM (vSLAM) algorithms include ORB-SLAM [8,14,15], DSO [16], and VINS-Mono [17]. The robustness and accuracy of vSLAM are constrained by environmental lighting conditions, making it challenging to handle scenarios with motion blur and high dynamic range (HDR) lighting conditions.

The event camera, as a novel type of biologically-inspired visual sensor modality, generates asynchronous pulse event outputs only when significant illumination changes are detected at each independent pixel. Consequently, event-based SLAM algorithms, such as EVO [18], ESVO [11], and Ultimate-SLAM [19], offer advantages including low power consumption, low latency, high dynamic response speed, and resistance to motion blur [20]. However, there is still a long way to go for the community to advance the development of event-based SLAM.

As described by Cadena et al. in their survey [4], the development of SLAM has gone through three main periods: the classical age (1986–2004), the algorithmic-analysis age (2004–2015), and the robust-perception age (since 2016). Despite the considerable development of SLAM research over the last 30 years, the robustness of SLAM algorithms still falls short when dealing with challenging environments, significantly limiting their applications in practice. Moreover, traditional SLAM techniques heavily rely on sufficient computing resources and power supply, further restricting their applicability.

In contrast to robots, animals possess the innate capability to navigate efficiently and robustly in natural environments by leveraging a range of sensory cues. Consequently, there has been a growing interest in developing SLAM algorithms that draw inspiration from the neural mechanisms employed by animals during navigation. By emulating the neural activities underlying animals' navigation, these bionic SLAM algorithms strive to enhance the performance and adaptability of robotic systems in challenging real-world scenarios.

RatSLAM [21,22] is a bio-inspired SLAM algorithm that derives insights from the hippocampal model of rodents and abstracts the concept of pose cells to represent the position and orientation of a mobile robot in a 2D environment. The RatSLAM system can be deployed in real robots and has successfully accomplished autonomous localization and mapping tasks in indoor office environments [23] and outdoor suburban areas [24].

Following RatSLAM, a series of relevant studies have been completed. Inspired by the three-dimensional navigation neural representation mechanisms found in birds and bats, Yu et al. proposed a four-degree-of-freedom NeuroSLAM [25] system as an extension of RatSLAM for mapping 3D environments. In order to address RatSLAM's sensitivity to perceptual aliasing caused by the low-dimensional sensory template matching, the LatentSLAM [26] system proposes an unsupervised representation learning method generating low-dimensional latent state descriptors and improves the robustness of the SLAM system by combining multiple sensors.

Spiking neural networks (SNNs) offer a suitable solution for feature extraction and descriptor representation for loop closure detection in SLAM. Safa et al. fused the data from an event camera and a radar, which is preprocessed by an SNN with continual spike-timing-dependent plasticity (STDP) learning, with the output used as feature descriptor encoding for loop closure detection and map correction in the SLAM architecture [27]. SNNs can also be applied in template matching and image-based place recognition. Hussaini et al. proposed a high-performance dual-layer SNN model with a novel weighted assignment scheme for visual place recognition (VPR) tasks, which can provide robust, energy-efficient, and low-latency robot localization [28].

SLAM algorithms running on neuromorphic hardware have been developed recently. Tang et al. proposed a brain-inspired SNN architecture that solves the uni-dimensional

SLAM problem by introducing spike-based reference frame transformation, visual likelihood computation, and Bayesian inference [29]. Integrated to Intel's Loihi processor, it performs well in terms of accuracy and energy-efficiency. Kreiser et al. introduced an SNN-based one-dimensional path integration architecture and implemented it on the neuromorphic hardware ROLLS [30]. Subsequently, Kreiser et al. extended this work to achieve two-dimensional path integration and map formation using an SNN architecture that comprises spiking neurons and plastic synapses, demonstrating the feasibility of the neuromorphic realization of SLAM [31].

In this paper, we propose LFVB-BioSLAM, a first-of-its-kind bionic SLAM system with a light-weight LiDAR-based front end and a bio-inspired vision-based back end. A range flow-based LiDAR odometry algorithm is adopted as the front end of the SLAM system, providing essential odometry estimation information for the back end, and a bio-inspired back-end processing algorithm based on the monocular RGB camera is proposed, performing loop closure detection and path integration. Experiments on the platform of a ground mobile robot are carried out to validate the feasibility of the proposed algorithm. The experimental results show that LFVB-BioSLAM outperforms RatSLAM [21,22], a vision-based bionic SLAM algorithm, and RF2O [32], a laser-based horizontal planar odometry algorithm, in terms of accuracy and robustness.

The main contributions of this paper are outlined below:

- We employ a lightweight range flow-based LiDAR odometry as the front end of our SLAM system, which quickly generates horizontal planar odometry output using the 3D LiDAR point cloud input.
- Our SLAM system incorporates a bio-inspired visual loop closure detection and path integration algorithm as the back end, which utilizes the odometry estimation from the front end, along with image input, to generate the robot's pose and construct the environmental map.
- We propose a novel bionic SLAM system called LFVB-BioSLAM, which combines the aforementioned front and back end components. Through real-world experiments, we validate the feasibility of LFVB-BioSLAM and demonstrate its superior performance in terms of accuracy and robustness.

2. Materials and Methods

We herein describe the proposed LFVB-BioSLAM, the system architecture of which is illustrated in Figure 1, in detail as follows. Section 2.1 introduces the light-weight range flow-based LiDAR odometry algorithm, which is the front end of the LFVB-BioSLAM system. Section 2.2 describes the bio-inspired visual loop closure detection and path integration algorithm, which constitutes the back end of LFVB-BioSLAM.

2.1. Front End: A Light-Weight Range Flow-Based LiDAR Odometry Algorithm

This section introduces the LiDAR odometry algorithm, which generates horizontal planar odometry output, i.e., the linear and angular velocity of the robot, from the 3D LiDAR point cloud input.

First, the required horizontal single-ring point cloud is extracted from the 3D LiDAR point cloud input. Next, a range flow constraint equation is established, with the horizontal planar velocities of the robot as variables. Finally, a robust cost function is applied to minimize the geometric constraints derived from the constraint equation, thereby generating the odometry estimation output.

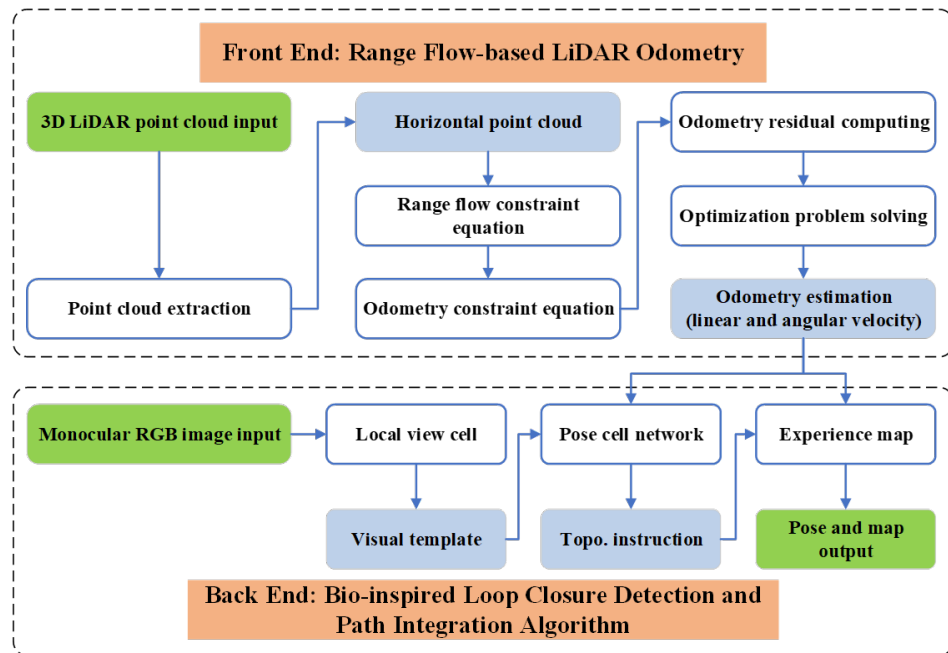
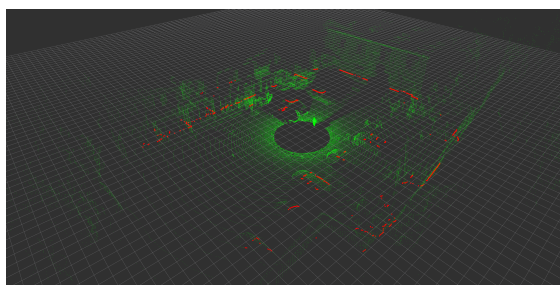


Figure 1. The LFVB-BioSLAM system architecture. The front-end range flow-based LiDAR odometry takes the 3D LiDAR point cloud as input, calculating the horizontal planar odometry estimation. The back-end bio-inspired visual loop closure detection and path integration algorithm takes both the monocular RGB image and the odometry estimation as input, outputting the pose of the robot and the map of the environment.

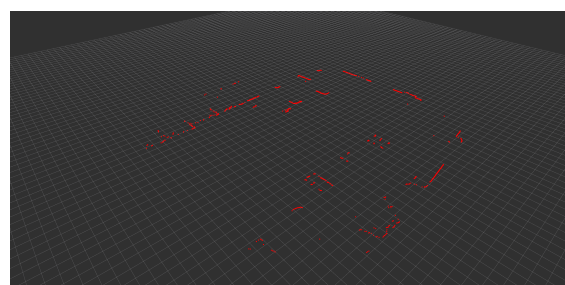
2.1.1. Horizontal Single-Ring Point Cloud Extraction

In order to use the results of the 3D LiDAR odometry (LO) algorithm as the benchmark for validating the designed algorithm, we adopt an Ouster 3D LiDAR in our sensor system. Within the input 3D multi-ring LiDAR point cloud (as shown in Figure 2a), the extraction of the required horizontal single-ring point cloud (as shown in Figure 2b) is performed, aiming to achieve a similar effect to that obtained by using a horizontal laser scanner.

Additionally, since our ground mobile robot platform equipped with the sensor system lacks corresponding planning and control algorithms and requires remote control by an operator, points generated by the robot itself and the operator are removed during the extraction process.



(a) 3D multi-ring LiDAR point cloud.



(b) Extracted horizontal single-ring point cloud.

Figure 2. An example of horizontal single-ring point cloud extraction.

Let P denote the multi-ring LiDAR point cloud in one scan and L denote the horizontal single-ring point cloud. Let P_i represent the point cloud on the i th ring, i.e., $P_i = \{p_{i,1}, p_{i,2}, \dots, p_{i,n_i}\}$, where n_i is the number of points on the i th ring and $p_{i,j}$ denotes the j th point on the i th ring. The required horizontal single-ring point cloud, L , can be expressed as

$$L = \bigcup_{i=1}^m L_i = \bigcup_{i=1}^m \{p_{i,j} \mid -h_0 \leq h(p_{i,j}) \leq h_0, d(p_{i,j}) \geq d_0\} \tag{1}$$

where m is the number of rings, and L_i is horizontal single-ring point cloud extracted from the i th ring, i.e., the points on the i th ring with their heights between $-h_0$ and h_0 and their distances from the LiDAR no less than d_0 . h_0 and d_0 are positive thresholds for height and distance, respectively.

2.1.2. Range Flow Constraint Equation

The following horizontal planar LO algorithm adopted in this paper draws inspiration from RF2O [32], a fast and precise laser-based horizontal planar odometry algorithm that estimates the 2D odometry based on consecutive range scans from a laser scanner.

At time t , the position of a scanned point P in $\{\mathbf{L}\}$, the local reference frame of the LiDAR, is determined by its polar coordinates $P(r, \theta)$, with $r \in (0, +\infty)$, and $\theta \in \left[-\frac{Fov}{2}, \frac{Fov}{2}\right]$, where Fov is the field of view of the horizontal planar single-ring point cloud. The scan range of point P can be expressed as $R(t, \theta) = r$.

Assuming the scan range function R is differentiable, we apply a first-order Taylor expansion to the scan range of point with the polar angle $\theta + \delta\theta$ in the subsequent scan:

$$R(t + \delta t, \theta + \delta\theta) = R(t, \theta) + \frac{\partial R}{\partial t}(t, \theta) \cdot \delta t + \frac{\partial R}{\partial \theta}(t, \theta) \cdot \delta\theta + o(\delta t, \delta\theta) \tag{2}$$

where δt denotes the time interval between two adjacent scans and $\delta\theta$ denotes the change in polar angle of the considered point. By ignoring the higher-order infinitesimal term $o(\delta t, \delta\theta)$ and dividing both sides of the equation by δt , the scan gradients can be related to the changes in range and polar angle within the time period $[t, t + \delta t]$:

$$\frac{\delta R}{\delta t} = R_t + R_\theta \frac{\delta\theta}{\delta t} \tag{3}$$

where $R_t = \frac{\partial R}{\partial t}(t, \theta)$ and $R_\theta = \frac{\partial R}{\partial \theta}(t, \theta)$ represent the scan gradients for t and θ , respectively.

When the time interval δt between two adjacent scans is sufficiently small, the changes in scan range and polar angle within the time period $[t, t + \delta t]$ divided by δt can be approximated by their corresponding derivatives, respectively, i.e., $\frac{\delta R}{\delta t} = \dot{r}$, $\frac{\delta\theta}{\delta t} = \dot{\theta}$. Thus, we have

$$\dot{r} = R_t + R_\theta \dot{\theta} \tag{4}$$

which is the range flow constraint equation [33].

Next, Equation (4) will be transformed into the constraint equation for the robot’s planar motion velocity (linear and angular velocity). First, the polar representation will be converted to its equal Cartesian representation. Let the Cartesian coordinates of point P in the local reference frame of the LiDAR be (x, y) , and assuming that the LiDAR is stationary, the velocity of point P in the x and y directions are denoted by \dot{x} and \dot{y} , respectively, as illustrated in Figure 3. Therefore, we have

$$\begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\frac{\sin \theta}{r} & \frac{\cos \theta}{r} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{5}$$

Considering the assumption of a static and rigid environment, the change in observation is in fact caused by the ego-motion of the LiDAR (both translational and rotational motion). Let the velocity of the LiDAR in the x and y directions be denoted as v_x and v_y , respectively, and the angular velocity be denoted as ω . Then, we have

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -1 & 0 & y \\ 0 & -1 & -x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{6}$$

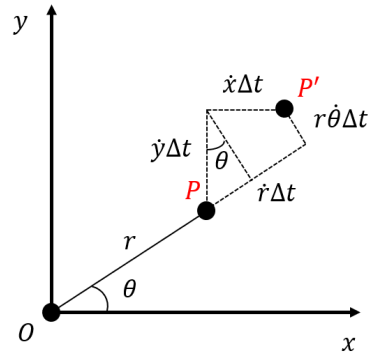


Figure 3. The vertical view of the local reference frame of the LiDAR. Assuming the LiDAR is stationary, the change in observation is caused by the motion from point P to point P' after a time interval δt .

By substituting the velocity transformation (6) into the polar-Cartesian transformation (5) and substituting \dot{r} and $\dot{\theta}$ into the range flow constraint Equation (4), we have

$$\left(\cos \theta + \frac{R_\theta \sin \theta}{r} \right) \cdot v_x + \left(\sin \theta - \frac{R_\theta \cos \theta}{r} \right) \cdot v_y - R_\theta \cdot \omega + R_t = 0 \tag{7}$$

which is the range flow constraint equation, with the LiDAR’s planar motion velocities (linear velocity v_x, v_y , and angular velocity ω) being the variables.

2.1.3. Estimation of Odometry

It is evident that in Equation (7), three sets of linearly independent constraints are theoretically sufficient to estimate the robot’s motion velocity. However, in practice, this approach is not feasible due to scan noise, linear approximation errors, and interference from moving objects in dynamic environments. These factors prevent Equation (7) from being an exact equality in typical situations. Therefore, we formulate the estimation of odometry as an optimization problem.

Let the robot’s planar motion velocity be denoted as $\mathbf{v} = [v_x, v_y, \omega]^T$. For each observed point in the scan and a given \mathbf{v} to be optimized, a range-flow residual can be constructed as

$$\psi(\mathbf{v}) = \left(\cos \theta + \frac{R_\theta \sin \theta}{r} \right) \cdot v_x + \left(\sin \theta - \frac{R_\theta \cos \theta}{r} \right) \cdot v_y - R_\theta \cdot \omega + R_t \tag{8}$$

The optimal estimate \mathbf{v}^* can be obtained by minimizing the sum of robust cost functions of all range-flow residuals:

$$\mathbf{v}^* = \arg \min_{\mathbf{v}} \sum_{i=1}^N F(\psi_i(\mathbf{v})) \tag{9}$$

with N denoting the number of scanned points, and the robust cost function $F(\psi)$ defined as

$$F(\psi) = \begin{cases} \frac{\psi^2}{k^2} \left(1 - \frac{\psi^2}{2k^2} \right), & \text{if } |\psi| < k, \\ \frac{1}{2}, & \text{if } |\psi| \geq k. \end{cases} \tag{10}$$

where k is a tunable positive parameter for decreasing the effects of outliers.

The optimization problem is then solved using the iteratively reweighted least squares (IRLS) algorithm. In each iteration, it re-computes the residuals and updates the weights until convergence is achieved.

2.2. Back End: A Bio-Inspired Visual Loop Closure Detection and Path Integration Algorithm

This section introduces the bio-inspired visual loop closure detection and path integration algorithm based on a monocular RGB camera, building upon the light-weight front-end odometry estimation algorithm discussed in Section 2.1.

First, the monocular RGB image is passed to the local view cell module, where the visual template corresponding to that specific scene is generated. Then, the odometry estimation and the visual template are jointly passed to the pose cell network module, which outputs topological graph instructions (creating new nodes, creating new edges, or setting nodes). Finally, the odometry estimation and the topological graph instructions are jointly passed to the experience map module, where map construction, graph relaxation, and path integration are performed, outputting the robot’s pose and the constructed environmental map.

The bio-inspired back-end processing algorithm implemented in this paper draws inspiration from RatSLAM [21], emulating the neural processes in the hippocampus of mammalian brains during navigation. It utilizes image data from a monocular RGB camera and combines it with the odometry estimation from the front-end LiDAR odometry algorithm to perform loop closure detection and path integration, outputting the robot’s pose and the constructed environmental map.

2.2.1. Local View Cell

Each local view cell is associated with a unique visual scene in the environment, which is intended to determine whether the current scene in the current view is a new scene or a previously seen one by comparing the input RGB image. In practice, we first preprocess the monocular RGB image, converting it into a corresponding visual template, based on which the determination of whether the current visual scene is new or previously seen is conducted. The image preprocessing procedure is illustrated in Figure 4.

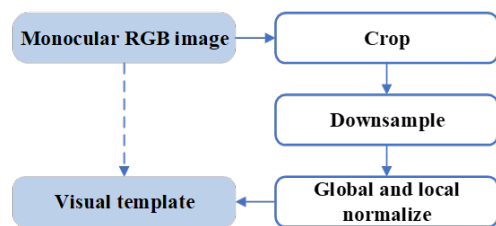


Figure 4. The image preprocessing procedure. First, we crop the image. By removing areas such as skies and roads, we focus on the regions of interest that are rich in environmental features, which can better represent the visual scene of the environment. Next, we downsample the cropped image, reducing its size while preserving important visual information of the environment. Last, we perform both global and local normalization on the downsampled image to minimize the effect of the illumination variation and ensure consistency of the visual template.

After the preprocessing steps, the local view cell compares the visual template Vt_{n+1} representing the current scene with all the historical visual templates $(Vt_1, Vt_2, \dots, Vt_n)$. The similarity comparison is based on the sum of absolute differences (SAD) between the current visual template and each historical visual template. For example, the SAD between visual templates Vt_p and Vt_q , denoted as $S_{p,q}$, is computed as

$$S_{p,q} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} |p_{i,j} - q_{i,j}| \tag{11}$$

where $p_{i,j}$ and $q_{i,j}$ are the pixel values of Vt_p and Vt_q respectively, with H and W being the height and width of the visual templates, respectively, which can be further illustrated in Figure 5.

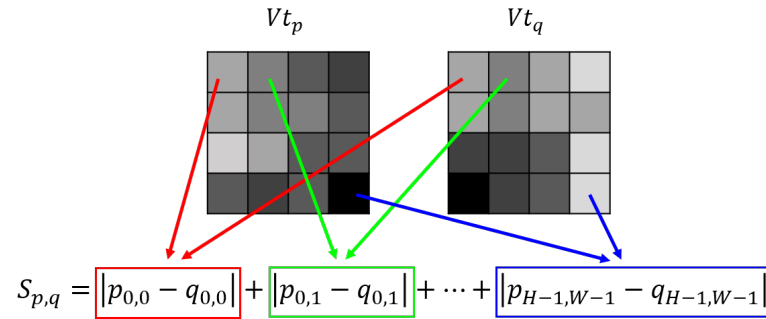


Figure 5. Computation of similarity between two visual templates Vt_p and Vt_q .

If the minimum SAD among all the historical visual templates is below a predetermined threshold, then the current visual template is supposed to represent the same visual scene as the corresponding historical visual template. Otherwise, if all SAD values are above the threshold, then the current visual template is added to the historical visual templates as a representation of the current new visual scene.

2.2.2. Pose Cell Network

The pose cell network draws inspiration from cells related to navigation (e.g., grid cells [34], place cells [35] and head direction cells [36]) found in the hippocampus of mammalian brains. It is built in the form of a continuous attractor network (CAN), as illustrated in Figure 6.

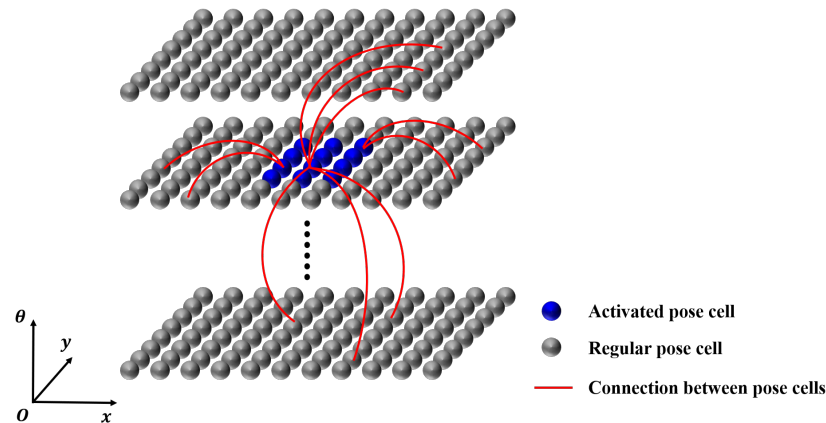


Figure 6. The pose cell network. Individual pose cells are interconnected with either excitatory or inhibitory connections. This network represents the three dimensions of a ground mobile robot’s pose, i.e., x , y , and θ .

The pose cell network has artificial dynamics designed to guide its behavior. When the network reaches a stable state, a cluster of activated pose cells forms a single energy packet, as shown by the dark blue region in Figure 6. The centroid of the energy packet represents the optimal estimate of the robot’s current pose within the network. The dynamics mechanism is implemented through the local excitation and global inhibition of the continuous attractor network, achieved by weighted connections with three-dimensional Gaussian distribution excitation coefficients $\epsilon_{a,b,c}^{exc}$ and inhibition coefficients $\epsilon_{a,b,c}^{inh}$, respectively, which are defined as

$$\epsilon_{a,b,c}^{exc} = e^{-\frac{a^2+b^2}{k_p^{exc}}} \cdot e^{-\frac{c^2}{k_\theta^{exc}}} \tag{12}$$

$$\epsilon_{a,b,c}^{inh} = -e^{-\frac{a^2+b^2}{k_p^{inh}}} \cdot e^{-\frac{c^2}{k_o^{inh}}} \tag{13}$$

where k_p and k_o are constants related to the spatial distribution variance of the pose cell network, and the subscripts “p” and “o” represent “position” and “orientation”, respectively.

Consequently, guided by the internal dynamics of the pose cell network, the changes in pose cell energies resulting from local excitation and global inhibition mechanisms are given by

$$\delta P_{x,y,\theta}^{exc} = \sum_{i,j,k} \left(P_{i,j,k} \cdot \epsilon_{a,b,c}^{exc} \right) \tag{14}$$

$$\delta P_{x,y,\theta}^{inh} = \sum_{i,j,k} \left(P_{i,j,k} \cdot \epsilon_{a,b,c}^{inh} \right) - \phi^{inh} \tag{15}$$

$$P_{x,y,\theta} \leftarrow P_{x,y,\theta} + \delta P_{x,y,\theta}^{exc} + \delta P_{x,y,\theta}^{inh} \tag{16}$$

where ϕ^{inh} is an additional global inhibition constant used to suppress the energies of other pose cell clusters with relatively high energies.

The pose cell network takes two types of inputs:

1. Visual templates from the local view cell module;
2. Odometry estimation from the front-end LiDAR odometry algorithm.

For the visual template input, there are two kinds of operations depending on whether the visual template is new or a historical one. If it is a new visual template, then it will be associated with the centroid of the current energy packet in the pose cell network. If it is a historical one, then the corresponding energy will be injected into the pose cells previously associated with that visual template through the excitatory connection coefficient γ^{exc} :

$$\delta P_{x,y,\theta} = \alpha \cdot \sum_j \left(\gamma_{j,x,y,\theta}^{exc} V_j \right) \tag{17}$$

where the constant coefficient α determines the degree of influence of visual information input on robot pose estimation. P and V represent the energy values of pose cells and local view cells, respectively.

For the odometry estimation input, we first perform the aforementioned local excitation and global inhibition steps, injecting and then removing a certain amount of energy around each active pose cell. Then, we normalize the energy of the whole pose cell network, maintaining its stability. Next, path integration is implemented using odometry information from the front end. Energy is transferred within the pose cells, facilitating the movement of the energy packet, the centroid of which in the current pose cell network is identified as the optimal estimate of the robot’s current pose within the network.

After these steps, the pose cell network outputs topological graph instructions and passes them to the experience map module. The topological graph instructions include:

1. Creating a new node (along with creating an edge from the previous node to the new node);
2. Creating an edge between two existing nodes;
3. Setting the current pose as an existing node.

2.2.3. Experience Map

The experience map represents the map in the form of a graph, combining information from the pose cell network and local view cells to uniquely estimate the robot’s pose. The node in the experience map is defined as

$$e_i = \{P_i, V_i, \mathbf{p}_i\} \tag{18}$$

where P_i and V_i represent the energy values of the pose cells and local view cells, respectively, and \mathbf{p}_i represents the pose in the experience map space. And the edge in the experience map is defined as

$$l_{ij} = \{\delta\mathbf{p}_{ij}, \delta t_{ij}\} \tag{19}$$

where $\delta\mathbf{p}_{ij}$ and δt_{ij} represent the relative differences in pose and time between two experience nodes e_i and e_j .

The experience map is updated based on the topological graph instructions given by the pose cell network, which include creating nodes, creating edges, or setting nodes.

Additionally, a graph relaxation algorithm is applied to reduce the drift of the odometry estimation, where the pose \mathbf{p}_i is modified as

$$\delta\mathbf{p}_i = \zeta \cdot \left(\sum_j (\mathbf{p}_j - \mathbf{p}_i - \delta\mathbf{p}_{ij}) + \sum_k (\mathbf{p}_k - \mathbf{p}_i - \delta\mathbf{p}_{ki}) \right) \tag{20}$$

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \delta\mathbf{p}_i \tag{21}$$

where ζ is the relaxation factor, e_j represents any experience node that e_i points to, and e_k represents any experience node that points to e_i , whose connection relationships are illustrated in Figure 7.

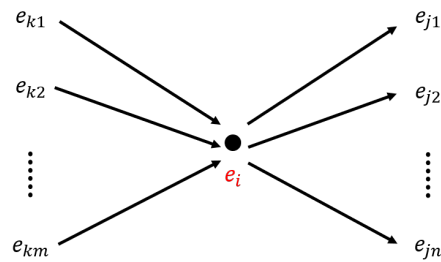


Figure 7. Connection relationships of the experience map. $e_{j1}, e_{j2}, \dots, e_{jn}$ are experience nodes that e_i points to, and $e_{k1}, e_{k2}, \dots, e_{km}$ are experience nodes that point to e_i .

3. Results

In this section, we validate the proposed LFVB-BioSLAM through the following real-world experiments using evaluation metrics that are designed to assess its localization and mapping performance.

3.1. Experimental Setup

The ground mobile robot platform equipped with the sensor system is illustrated in Figure 8.

The experimental field was chosen as a courtyard with a length of approximately 50 m and a width of approximately 20 m. Two groups of experiments (denoted as Exp. 1 and Exp. 2) were conducted, in each of which the ground mobile robot was remotely operated along a corresponding predetermined route (denoted as Route 1 and Route 2, respectively) to collect data from the sensors. The bird’s-eye view of the experimental field along with the starting and ending positions of the two routes are shown in Figure 9.

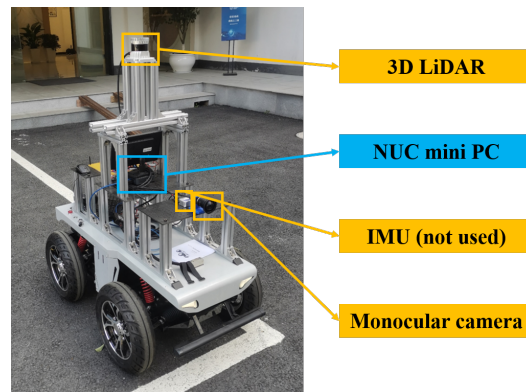


Figure 8. The ground mobile robot platform for data collection. The sensor system consists of a 3D LiDAR, an inertial measurement unit (IMU, not used in this implementation), and a monocular camera. The platform is equipped with an NUC mini PC running the necessary sensor drivers and collecting the datasets.



Figure 9. The bird's-eye view of the experimental field with the starting and ending positions of the two routes corresponding to Exp. 1 and Exp. 2, respectively.

3.2. Experimental Results

We adopt DLO (direct LiDAR odometry) [37], which is among the state-of-the-art (SOTA) algorithms in the field of LiDAR odometry and LiDAR SLAM, to obtain the relatively accurate robot odometry results using the 3D LiDAR point cloud data. The results of DLO serve as a benchmark against which the performance of the following three algorithms are evaluated:

1. RatSLAM [21], a classical bio-inspired visual SLAM, with a similar bio-inspired back-end processing mechanism to our LFVB-BioSLAM;
2. RF2O [32], a range flow-based horizontal planar laser odometry, with a similar processing mechanism to the front-end odometry estimation algorithm in our LFVB-BioSLAM;
3. LFVB-BioSLAM proposed by us, with a LiDAR-based front end and a vision-based bio-inspired back end.

The best parameter configurations for different algorithms are listed in Table 1. The poses and trajectories of the robot, as well as the constructed environment maps generated by these algorithms, are shown using RViz in Figure 10. Moreover, the open-source EVO [38] toolkit is used for the further visualization of robot trajectories obtained by each

algorithm, as shown in Figure 11, providing an intuitive understanding of the accuracy performance of each algorithm.

Table 1. Parameter configurations.

Algorithm	Parameter	Meaning	Value
RF2O & LFVB-BioSLAM	l_{ctf}	coarse-to-fine levels	5
	i_{irls}	IRLS iterations	5
	\mathbf{m}	Gaussian mask	(0.0625, 0.25, 0.375, 0.25, and 0.0625)
RatSLAM & LFVB-BioSLAM	y_m	max y for image cropping	1000
	th_{vt}	visual template matching threshold	0.03
	x_{vt}	x size of visual template	60
	y_{vt}	y size of visual template	10
	st_{vt}	visual template matching step	1
	n_{vt}	visual template normalization factor	0.4
	x_{pc}	x size of pose cell	2
	s_{pc}	side length of (x, y) plane of pose cell network	100
	l_{exp}	experience map graph relaxation loops per iteration	20

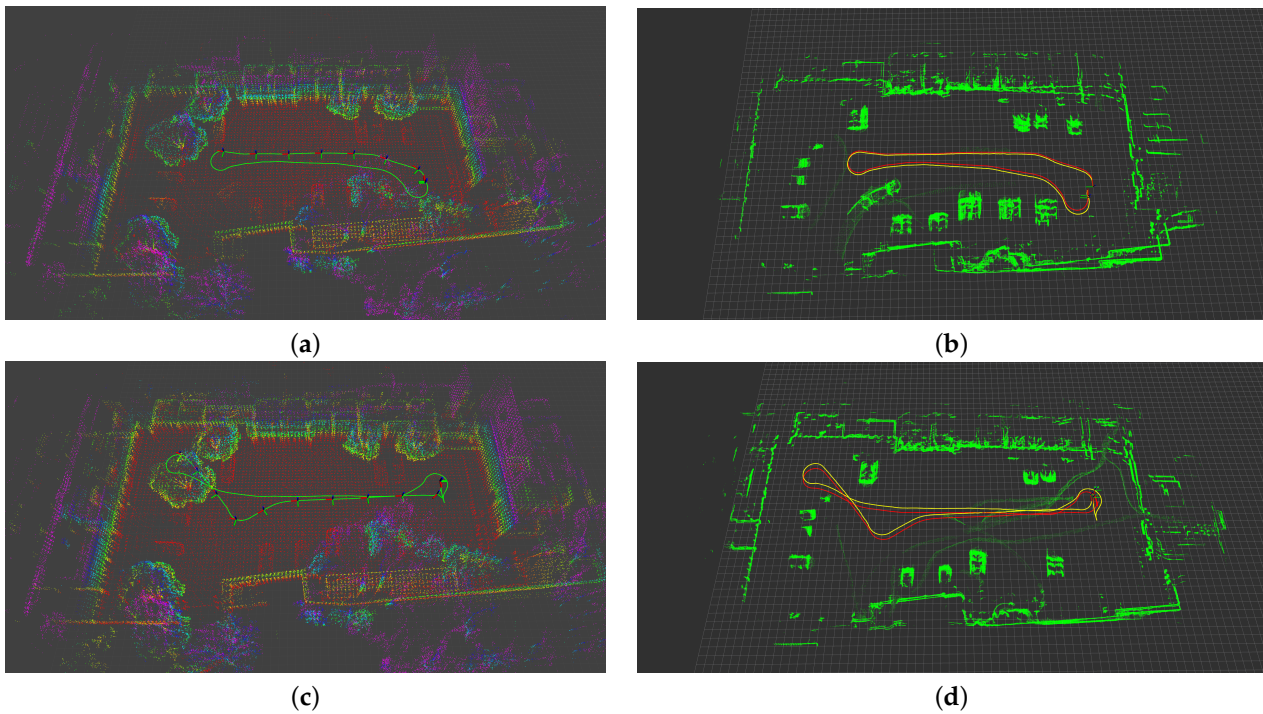
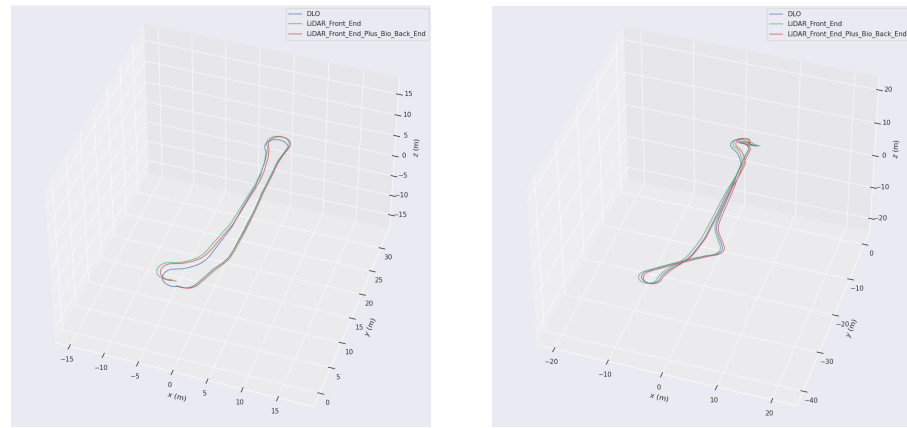


Figure 10. Results of robot poses and trajectories and environment maps generated by each algorithm. (a) Exp. 1, DLO. (b) Exp. 1, RF2O (in yellow) and LFVB-BioSLAM (in red). (c) Exp. 2, DLO. (d) Exp. 2, RF2O (in yellow) and LFVB-BioSLAM (in red).

It is worth mentioning that RatSLAM fails to produce reliable results in both Exp. 1 and Exp. 2 as it exhibits significant drift, resulting in localization failure. Therefore, the results obtained from the RatSLAM algorithm are not depicted in these figures.



(a) Exp. 1, visualization of trajectories. (b) Exp. 2, visualization of trajectories.

Figure 11. Visualization of robot trajectories generated by each algorithm in Exp. 1 and Exp. 2. The result of DLO, RF2O, and LFVB-BioSLAM are denoted as *DLO*, *LiDAR_Front_End*, and *LiDAR_Front_End_Plus_Bio_Back_End*, respectively.

In order to quantitatively evaluate the accuracy performance of the aforementioned algorithms, the translational absolute pose error (Trans. APE) and rotational absolute pose error (Rot. APE) are selected as the evaluation metrics. The maximum error (max), mean error (mean), and root mean square error (RMSE) are calculated to compare the errors between the results of each algorithm and the benchmark values obtained from DLO. The comparison results of Translational APE and Rotational APE are illustrated in Table 2 and Table 3, respectively, which can also be visualized in Figure 12.

Table 2. The comparison results of Translational APEs.

Experiment	Trajectory Length [m]	Algorithm	Trans. APE [m]		
			Max	Mean	RMSE
Exp. 1	74.76	RatSLAM *	/	/	/
		RF2O	1.3116	0.5484	0.6366
		LFVB-BioSLAM	1.0280	0.5347	0.5835
Exp. 2	101.76	RatSLAM *	/	/	/
		RF2O	1.8066	0.3913	0.4838
		LFVB-BioSLAM	0.5480	0.2967	0.3136

* Failure due to significant localization drift.

Table 3. The comparison results of Rotational APEs.

Experiment Number	Trajectory Length [m]	Algorithm	Rot. APE [°]		
			Max	Mean	RMSE
Exp. 1	74.76	RatSLAM *	/	/	/
		RF2O	8.3100	1.6831	1.8798
		LFVB-BioSLAM	2.8556	1.6128	1.7079
Exp. 2	101.76	RatSLAM *	/	/	/
		RF2O	4.9346	1.3323	1.6260
		LFVB-BioSLAM	2.8120	1.0148	1.1193

* Failure due to significant localization drift.

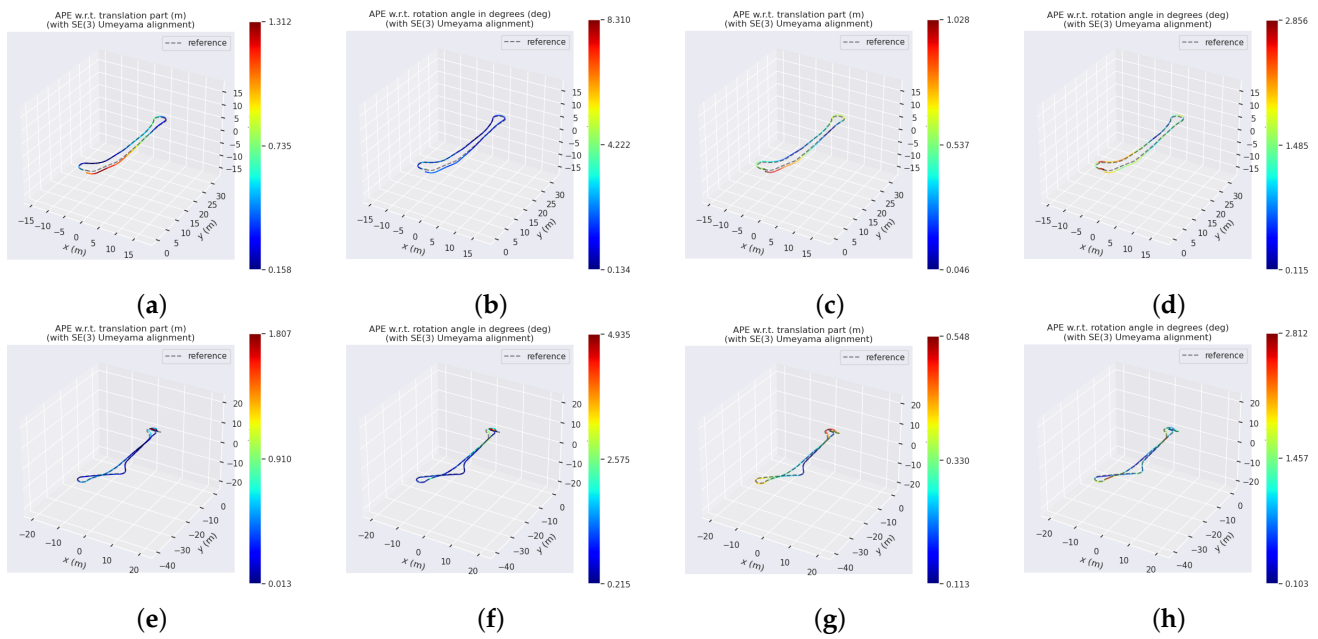


Figure 12. Visualization of the Translational and Rotational APEs of RF2O and LFVB-BioSLAM using the results of DLO as benchmark values for reference. (a) Exp. 1, Trans. APE of RF2O. (b) Exp. 1, Rot. APE of RF2O. (c) Exp. 1, Trans. APE of LFVB-BioSLAM. (d) Exp. 1, Rot. APE of LFVB-BioSLAM. (e) Exp. 2, Trans. APE of RF2O. (f) Exp. 2, Rot. APE of RF2O. (g) Exp. 2, Trans. APE of LFVB-BioSLAM. (h) Exp. 2, Rot. APE of LFVB-BioSLAM.

4. Discussion

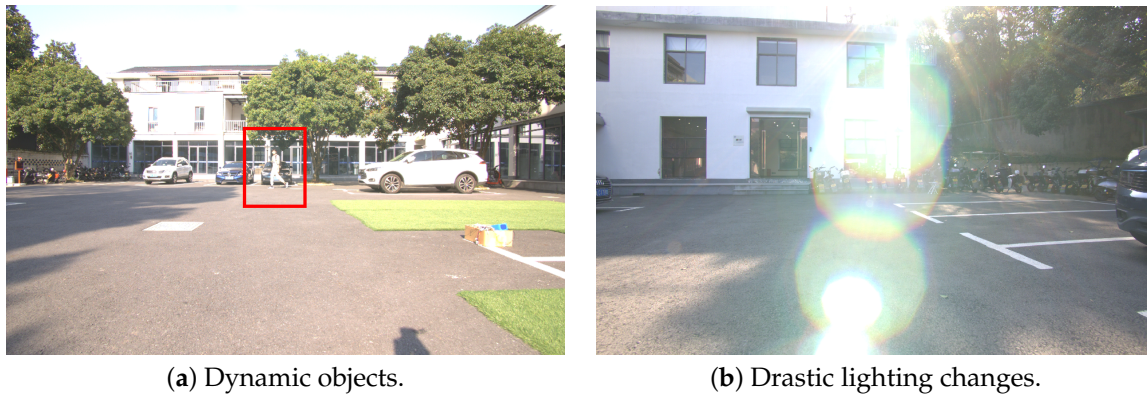
In the experiments presented above, the performances of RatSLAM, RF2O, and the proposed LFVB-BioSLAM are compared against each other using DLO as the benchmark algorithm. Compared with the other two algorithms, the proposed LFVB-BioSLAM has significant advantages in terms of accuracy and robustness. The qualitative comparison results of computational complexity, accuracy, and robustness can be summarized in Table 4.

Table 4. The qualitative comparison results of each algorithm.

Algorithm	Computational Complexity	Accuracy	Robustness
RatSLAM	+	+	+
RF2O	++	++	++
LFVB-BioSLAM	+++	+++	+++

The greater the number of +, the higher the corresponding index.

In both groups of experiments, RatSLAM failed due to significant localization drift, being incapable of performing localization and mapping tasks. The main reason is that RatSLAM uses a monocular camera as its single sensor, which fails to meet the practical navigation and localization requirements in certain experimental scenarios that are challenging for vision-based SLAM algorithms. For example, dynamic objects and drastic lighting changes in the environment, as shown in Figure 13, may lead to significant errors in the relatively simple front-end visual odometry and visual template matching steps of RatSLAM. These errors are difficult to correct in subsequent modules such as the pose cell network and the experience map, ultimately resulting in its failure with poor accuracy and robustness.



(a) Dynamic objects.

(b) Drastic lighting changes.

Figure 13. Scenarios in our experiments that are challenging for vision-based SLAM algorithms.

The proposed LFVB-BioSLAM, with the inclusion of the bio-inspired back end for loop closure detection and path integration, significantly outperforms RF2O in accuracy. The major function of the RF2O-like light-weight LiDAR odometry, which is the front end of LFVB-BioSLAM, is to provide fast and rough odometry estimation with low computational resource consumption, which will be further optimized by the bio-inspired back end. LFVB-BioSLAM also demonstrates good performance in the face of dynamic or partially degraded scenes with strong robustness.

5. Conclusions and Future Work

In this paper, we present LFVB-BioSLAM, a first-of-its-kind SLAM system consisting of a light-weight LiDAR-based front end and a bio-inspired vision-based back end, which, to the best of our knowledge, has not been previously proposed. The experimental results demonstrate that LFVB-BioSLAM outperforms RatSLAM, a vision-based bionic SLAM algorithm, and RF2O, a laser-based horizontal planar odometry algorithm, in terms of accuracy and robustness, validating the feasibility of the proposed bionic SLAM architecture.

In summary, the proposed bionic SLAM system, featuring an innovative framework, demonstrates commendable performance and practicality. This, to a considerable extent, facilitates the development of bionic SLAM research and lays a solid foundation for the future evolution of bionic SLAM systems tailored for fully autonomous robot navigation tasks.

For future work, more research on bionic SLAM remains to be done. On the one hand, from the perspective of bionics principles, bionic SLAM algorithms hold immense potential. By drawing inspiration from the latest theories about the animal navigation mechanisms in neural science, such as navigation cell collaboration mechanisms, it might be possible to construct a novel SLAM framework. The implementation of such a framework could potentially lead to breakthroughs in theoretical and technological advancements. On the other hand, neuromorphic hardware exhibits promising advantages, including asynchronous computing and event-based communication, along with ultra-low energy consumption, high parallelism, and efficiency. The application of neuromorphic hardware in bionic SLAM systems can provide assurance for achieving high real-time and energy-efficient performance, so the exploration of bionic SLAM algorithms based on neuromorphic hardware also carries significant research value.

Author Contributions: Conceptualization, R.G., Z.W. and S.G.; methodology, R.G. and S.G.; software, R.G.; validation, R.G. and S.G.; formal analysis, R.G.; investigation, R.G.; resources, R.G. and Z.W.; data curation, R.G., Z.W. and C.J.; writing—original draft preparation, R.G.; writing—review and editing, R.G., Z.W. and Y.Z.; visualization, R.G.; supervision, Y.Z.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by STI 2030-Major Projects 2021ZD0201403, in part by NSFC 62088101 Autonomous Intelligent Unmanned Systems.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

SLAM	Simultaneous localization and mapping
IMU	Inertial measurement unit
HDR	High dynamic range
SNN	Spiking neural network
STD	Spike-timing-dependent plasticity
VPR	Visual place recognition
LO	LiDAR odometry
SAD	Sum of absolute differences
CAN	Continuous attractor network
SOTA	State-of-the-art
RMSE	Root mean square error

References

1. Thrun, S. Probabilistic robotics. *Commun. Acm* **2002**, *45*, 52–57. [[CrossRef](#)]
2. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [[CrossRef](#)] [[PubMed](#)]
3. Zhuang, G.; Bing, Z.; Huang, Y.; Huang, K.; Knoll, A. A Biologically-Inspired Simultaneous Localization and Mapping System Based on LiDAR Sensor. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 13136–13142.
4. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [[CrossRef](#)]
5. Saputra, M.R.U.; Markham, A.; Trigoni, N. Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–36. [[CrossRef](#)]
6. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In Proceedings of the 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2020; pp. 5135–5142.
7. Doer, C.; Trommer, G.F. Radar visual inertial odometry and radar thermal inertial odometry: Robust navigation even in challenging visual conditions. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 331–338.
8. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [[CrossRef](#)]
9. Dai, W.; Zhang, Y.; Li, P.; Fang, Z.; Scherer, S. Rgb-d slam in dynamic environments using point correlations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 373–389. [[CrossRef](#)]
10. Saputra, M.R.U.; de Gusmao, P.P.; Lu, C.X.; Almalioglu, Y.; Rosa, S.; Chen, C.; Wahlström, J.; Wang, W.; Markham, A.; Trigoni, N. Deeptio: A deep thermal-inertial odometry with visual hallucination. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1672–1679. [[CrossRef](#)]
11. Zhou, Y.; Gallego, G.; Shen, S. Event-based stereo visual odometry. *IEEE Trans. Robot.* **2021**, *37*, 1433–1450. [[CrossRef](#)]
12. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. *Robot. Sci. Syst.* **2014**, *2*, 1–9.
13. Xu, W.; Cai, Y.; He, D.; Lin, J.; Zhang, F. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Trans. Robot.* **2022**, *38*, 2053–2073. [[CrossRef](#)]
14. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
15. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [[CrossRef](#)]
16. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 611–625. [[CrossRef](#)]
17. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [[CrossRef](#)]
18. Rebecq, H.; Horstschäfer, T.; Gallego, G.; Scaramuzza, D. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robot. Autom. Lett.* **2016**, *2*, 593–600. [[CrossRef](#)]
19. Vidal, A.R.; Rebecq, H.; Horstschäfer, T.; Scaramuzza, D. Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. *IEEE Robot. Autom. Lett.* **2018**, *3*, 994–1001. [[CrossRef](#)]

20. Huang, K.; Zhang, S.; Zhang, J.; Tao, D. Event-based Simultaneous Localization and Mapping: A Comprehensive Survey. *arXiv* **2023**, arXiv:2304.09793.
21. Milford, M.J.; Wyeth, G.F.; Prasser, D. RatSLAM: A hippocampal model for simultaneous localization and mapping. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'04. 2004, New Orleans, LA, USA, 26 April–1 May 2004; Volume 1, pp. 403–408.
22. Ball, D.; Heath, S.; Wiles, J.; Wyeth, G.; Corke, P.; Milford, M. OpenRatSLAM: An open source brain-based SLAM system. *Auton. Robot.* **2013**, *34*, 149–176. [[CrossRef](#)]
23. Milford, M.; Wyeth, G. Persistent navigation and mapping using a biologically inspired SLAM system. *Int. J. Robot. Res.* **2010**, *29*, 1131–1153. [[CrossRef](#)]
24. Milford, M.J.; Wyeth, G.F. Mapping a suburb with a single camera using a biologically inspired SLAM system. *IEEE Trans. Robot.* **2008**, *24*, 1038–1053. [[CrossRef](#)]
25. Yu, F.; Shang, J.; Hu, Y.; Milford, M. NeuroSLAM: A brain-inspired SLAM system for 3D environments. *Biol. Cybern.* **2019**, *113*, 515–545. [[CrossRef](#)] [[PubMed](#)]
26. Çatal, O.; Jansen, W.; Verbelen, T.; Dhoedt, B.; Steckel, J. LatentSLAM: Unsupervised multi-sensor representation learning for localization and mapping. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 6739–6745.
27. Safa, A.; Verbelen, T.; Ocket, I.; Bourdoux, A.; Sahli, H.; Catthoor, F.; Gielen, G. Fusing Event-based Camera and Radar for SLAM Using Spiking Neural Networks with Continual STDP Learning. *arXiv* **2022**, arXiv:2210.04236.
28. Hussaini, S.; Milford, M.; Fischer, T. Spiking neural networks for visual place recognition via weighted neuronal assignments. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4094–4101. [[CrossRef](#)]
29. Tang, G.; Shah, A.; Michmizos, K.P. Spiking neural network on neuromorphic hardware for energy-efficient unidimensional slam. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4176–4181.
30. Kreiser, R.; Cartiglia, M.; Martel, J.N.; Conradt, J.; Sandamirskaya, Y. A neuromorphic approach to path integration: A head-direction spiking neural network with vision-driven reset. In Proceedings of the 2018 IEEE international symposium on circuits and systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5.
31. Kreiser, R.; Renner, A.; Sandamirskaya, Y.; Pienroj, P. Pose estimation and map formation with spiking neural networks: Towards neuromorphic slam. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 2159–2166.
32. Jaimez, M.; Monroy, J.G.; Gonzalez-Jimenez, J. Planar odometry from a radial laser scanner. A range flow-based approach. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 4479–4485.
33. Spies, H.; Jähne, B.; Barron, J.L. Range flow estimation. *Comput. Vis. Image Underst.* **2002**, *85*, 209–231. [[CrossRef](#)]
34. Hafting, T.; Fyhn, M.; Molden, S.; Moser, M.B.; Moser, E.I. Microstructure of a spatial map in the entorhinal cortex. *Nature* **2005**, *436*, 801–806. [[CrossRef](#)] [[PubMed](#)]
35. O'Keefe, J.; Dostrovsky, J. The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* **1971**, *34*, 171–175. [[CrossRef](#)]
36. Taube, J.S.; Muller, R.U.; Ranck, J.B. Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *J. Neurosci.* **1990**, *10*, 420–435. [[CrossRef](#)]
37. Chen, K.; Lopez, B.T.; Agha-mohammadi, A.-a.; Mehta, A. Direct lidar odometry: Fast localization with dense point clouds. *IEEE Robot. Autom. Lett.* **2022**, *7*, 2000–2007. [[CrossRef](#)]
38. Grupp, M. evo: Python Package for the Evaluation of Odometry and SLAM. 2017. Available online: <https://github.com/MichaelGrupp/evo> (accessed on 1 April 2023).

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.