Journal of Global Positioning Systems (*2022*)
Vol. 18, No. 1: 70-83
DOI: 10.5081/jgps.18.1.70

Bio-inspired Map Construction based on Brain Navigation Mechanism for Indoor Robots

Yixuan Long, Hao Wang, Fang Ye, Yibing Li, Qian Sun

College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001

✉ **Corresponding author**: Qian Sun, qsun@hrbeu.edu.cn

Abstract: Mapping is critical for an autonomous robot performing tasks in an unknown environment, which provides the environment information for task planning. Inspired by the presence of cells in the mammals' brain that help mammals rapidly cognize the surroundings, considering visual ambiguity that may be happened indoors, an orientation-independent boundary cell model based on the boundary vector cells in the brain is proposed to tackle the obstacle information in the environment, and it is fused into a metric-topological map to represent the structural information which increases the functionality of the map. The simulation results show that the expression of boundaries or obstacles in the environment can be obtained through the firing rate of boundary cells, which enhances the information content of the map. Meanwhile, the algorithm can build a consistent representation of the environment with sensor noise and achieves a root mean square error of 11.42cm in a 16m×17m indoor environment, effectively calibrating the sensor drift error, and ensuring the accuracy of the map.

Keywords: bio-inspired; mapping; metric-topological map; indoor robots

1 Introduction

In recent years, robots have been widely used in a variety of industries to execute tasks autonomously. Simultaneous localization and mapping (SLAM) is a critical technology for robots working in an unknown environment, which allows the robot to map the environment while running and then utilize the built map as the prior information for navigation. Thus, constructing a suitable map is significant considering the intention of the tasks and the working environment. As humans and many animals have the nature to explore and navigate in new environments, researchers are inspired to imitate this mechanism in robotics to enhance their mapping and cognitive ability.

The spatial cognition mechanism in the mammal's brain has been researched by neuroscientists for decades. Early in 1971, O'Keefe and Dostrovsky found that some cells in the rodent's hippocampus are active only when the rat visit a specific location, which serves as an internal cognitive map, named place cells [1]. Based on this discovery, more spatial cells are founded. In the entorhinal cortex, a kind of neuron is observed activating at multiple specific locations which form a hexagonal grid in space [2]. These grid cells integrate the rat's self-motion information and provide the path integration function of brain navigation, while head direction cells respond to the movement of the rat's head facing, improving an animal's ability to solve spatial problems [3]. With the help of the directional information, the boundary vector cells which produce a high firing rate when the rat reaches the boundary of the test environment can generate direction-independent activity [4]. Through the coordinated activity of neural circuits between these cells, the positioning system in the brain is formed.

Current research on bio-inspired map construction methods can be divided into two groups: one group uses the place cells to code the whole environment which means that each site corresponds to a specific neuron. Arleo proposed a computational model based on head direction cells and place cells to achieve navigation in a two-dimensional small-scale environment [5]. Tian used competitive Hebbian learning to select grid cell subsets to generate place cell population activities, thereby obtaining a robot exploration map [6]. Instead of using the place cells only, Zhou proposed a new place cell representation under multi-information perception generated by a weighted fusion of visual-related and grid cells [7]. Tang established a cognitive map based on the functioning of the hippocampus and the entorhinal cortex [8]. Although these works have high biology fidelity, the complexity and required resources could increase greatly when working in larger environments, while the other group builds the map using a hybrid map model based on the episodic memory mechanism, which can be utilized in a large area. Milford proposed a metric-topological experience map that records the activity of cells in a representative place as an experience so that the map is more extensible [9]–[11]. In this system, because visual information is the only source of allothetic cues, many improvements to the visual system have been done to increase the reliability of the map when facing visual ambiguity. The FAB-MAP is introduced to deal with the changes over time [12]. In [13], an appearance-based frequency-tuned model is proposed, which converts visual input into saliency maps to reduce the influence of light. To avoid perceptual aliasing, a dynamic growing self-organizing map based on direction and feature parameters is introduced and gains a good performance in the office environment [14]. There are many other sources of information used by animals during navigation, such as Zeng proposed that local view information and motion information were introduced into HD-by-velocity cells and GirD-by-velocity cells, and the two types of cells were connected and the relocation was achieved according to attractor dynamics [15]. Based on the self-centered confidence map, Gupta generates the current positioning through multi-scale confidence superposition [16]. In addition, literature [17]-[18] points out that distance information extracted from tactile sense and audio can also be used for localization. However, there are a few types of research on using structural information. In BatSLAM, visual templates are replaced by sonar fingerprints [19]. But in their experiment, using only one type of source still could suffer from ambiguity. For example, the lidar-based method can hardly distinguish repetitive environments with similar structures [20]. In ViTa-SLAM, the local tactile information is saved as a histogram template and a slope distribution array in the node [21], but it is hard to extract the global structure from the map. In this paper, a novel cognitive map is proposed to increase the feature redundancy and functionality in the indoor visual ambiguous environment. An ego-allocentric modulation method is proposed to transform the egocentric lidar information into the allocentric boundary information so that the lidar information can be fused into the map in a more biological way to provide an absolute reference for the localization process.

The rest of this paper is organized as follows. The computational model of navigation cells and the problem formulation are both introduced in Section 2. The whole mapping system including the ego-allocentric modulation and the cognitive map-building process is described in Section 3 and Section 4. The experimental results are demonstrated in Section 5. Finally, the conclusion is presented in Section 6.

2 Preliminaries

2.1 Pose cells model

Since the activity of place cells and head direction cells indicate the rodent's pose in the real world, a pose cell network is proposed to represent the belief of the agent's current location and orientation [22]. To simulate the activity of the

above two cells, the pose cell network uses a three-dimensional continuous attractor neural network (CANN) which is a recurrent dynamical network that is presented to model the space-related cells in neuroscience[23]. The CANN consists of a group of attractors and the connections between them, while the connections' weights are fixed and can be both excitatory and inhibitory. The activation of each attractor is continuous ranging from zero to one, which indicates the firing rate of the pose cell as well as the belief of the robot's pose in this paper, and it is operated by the self-motion cues, and the activation changes from other connected navigation cells.

The three dimensions of the pose cell network represent the plane coordinates (*x*, *y*) and the robot head facing θ, respectively. Due to the attractor dynamics of CANN, the activity in the pose cell network will converge to a single cluster, which is named the activity packet or energy packet. Thus, the attractor dynamics of the pose cell network can be represented as

$$\Delta P_{x',y',\theta'} = \sum_{i=0}^{s_{xy}} \sum_{j=0}^{s_{xy}} \sum_{k=0}^{s_{th}} P_{i,j,k} \varepsilon_{a,b,c} - \varphi + I_{ext} \tag{1}$$

where $P_{i,j,k}$ and $\Delta P_{x',y',\theta'}$ denote the activity level and the activation change of the corresponding cell, while the subscripts indicate the cell's location in the network. s_{xy} and s_{th} represent the side length of the CANN. $\varepsilon_{a,b,c}$ is the fixed connection weight matrix between pose cells which activates neighboring neurons and suppresses the distant cells, while constant φ serves as the global inhibition in the network. Because the pose cell also connects with other navigation cells, I_{ext} is introduced to represent the external excitatory input. As the internal connection weight is related to cells' relative distance, the matrix is given by

$$\begin{aligned} \varepsilon_{a,b,c} = & \exp\left(-\frac{(a^2+b^2)}{k_p^{exc}}\right)\exp\left(-\frac{c^2}{k_d^{exc}}\right) \\ & -\exp\left(-\frac{(a^2+b^2)}{k_p^{inh}}\right)\exp\left(-\frac{c^2}{k_d^{inh}}\right) \end{aligned} \tag{2}$$

where k_p and k_d are the variance constants, The inhibitory variance k^{inh} is larger than the excitatory variance k^{exc}, forming the Mexican hat function. And the indices *a*, *b* and *c* are the distances between cells in different dimensions.

The activity packet in the network can be shifted by self-motion cues during path integration which is analog to the dead reckoning in a robotic system. If the self-motion signal only has the translation velocity, the activity packet would move within the *x*-*y* plane; If the angular velocity is not zero, the packet would shift along the θ axis. The magnitude of shift depends on the translation velocity *v* and angular velocity ω, which is calculated as follows:

$$\delta x_o = \lfloor k_x v \cos\theta \rfloor, \delta x_f = k_x v \cos\theta - \delta x_o \tag{3}$$

$$\delta y_o = \lfloor k_y v \sin\theta \rfloor, \delta y_f = k_y v \sin\theta - \delta y_o \tag{4}$$

$$\delta\theta_o = \lfloor k_\theta \omega \rfloor, \delta\theta_f = k_\theta \omega - \delta\theta_o \tag{5}$$

Here, $\lfloor x \rfloor$ is a floor function. k_x, k_y, and k_θ are the path integral constants. Thus, δx_o and δx_f represent the integral and fractional part of the shift size along x-axes, respectively. The path integration process in the pose cell network can be expressed as follow:

$$\Delta P_{x',y',\theta'} = \sum_{i=\delta x_o}^{\delta x_o+1} \sum_{j=\delta y_o}^{\delta y_o+1} \sum_{k=\delta\theta_o}^{\delta\theta_o+1} \alpha_{i,j,k} P_{(x'+i)(y'+j)(\theta'+k)} \tag{6}$$

$$\alpha_{i,j,k} = g\left(\delta x_f, i-\delta x_o\right) g\left(\delta y_f, i-\delta y_o\right) g\left(\delta\theta_f, i-\delta\theta_o\right) \tag{7}$$

$$g\left(a,b\right) = \begin{cases} 1-a, & b=0 \\ a, & b=1 \end{cases} \tag{8}$$

$\alpha_{i,j,k}$ is a residual component that is spread over a $2\times2\times2$ cube to quantify the effect of the grid.

2.2 Boundary vector cells model

The firing rate of boundary vector cells is independent of the head direction, which means the receptive field of the cell is a specific distance and allocentric direction (such as south or east) in the world frame. Thus, the computational model of the i-th boundary vector cell's firing rate is as follows :

$$\delta f_i = g(r,\theta)\delta\theta_i \tag{9}$$

$$g_i(r,\theta) \propto \frac{\exp[-(r-d_i)^2/2\sigma_{rad}^2(d_i)]}{\sqrt{2\pi\sigma_{rad}^2(d_i)}} \times \frac{\exp[-(\theta-\phi_i)^2/2\sigma_{ang}^2]}{\sqrt{2\pi\sigma_{ang}^2}} \tag{10}$$

Here, (d_i,ϕ_i) is the receptive field of the cell, while (r,θ) denotes the location of boundary section in the polar coordinate, supposing the rodent is in the origin of the coordinate system. σ_{rad} and σ_{ang} are constant variances of distance and direction respectively, and δ is the impulse function. Fig.1 shows the relationship between the boundary vector cell's receptive field and the firing rate.

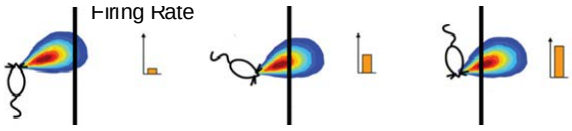


Fig. 1 The boundary vector cell model[4]

2.3 Local view cells model

The local view cells model is an array of cells that record the visual information during experiments. The cell's firing rate is related to the vision system which converts the visual input into a visual template and compares it with existing templates. The comparison results, including the absolute difference between input and database as well as the shift angle, activate the local view cell according to the index of templates.

2.4 Problem formulation

In mammals' brains, the activity patterns of those navigation cells can generate episodic memory and are served as a cognitive map. In this paper, we refer to the experience map that is used in RatSLAM[9] to construct a metric-topological map that can be utilized for robots. The map can be described as a tuple:

$$\mathrm{EM} = \langle \boldsymbol{E}, \boldsymbol{L} \rangle \tag{2}$$

where $\boldsymbol{E} = \{e_1,\ldots,e_n\}$ is a set of vertices and $\boldsymbol{L} = \{l_1,\ldots,l_n\}$ is a set of edges which denote the relative positional relationship between linked vertices. Each vertex e_i is an experience during running, which can then be defined as:

$$e_i = \{P^i, V^i, B^i, \mathbf{p}^i\} \tag{3}$$

where P^i, V^i, B^i are the state of pose cells, local view cells, and the boundary vector cells, respectively. $\mathbf{p}^i$ is the location of the experience node in the world coordinate system.

3 System Framework

To enhance the functionality and enrich the information of the map, this paper proposes a framework to combine the visual and lidar cues into the experience map. As shown in Fig.2, the framework consists of three threads for tackling different types of inputs. During the mapping process, the wheel odometry and onboard sensors record the idiothetic and environmental information and send them to the corresponding thread. Since the visual cues are transferred as images, the region of interest which is assigned manually is clipped

from the image and then downsize to calculate similarities and the offsets of head direction with existing templates using Sum of Absolute Differences (SAD). A local view cell is activated when there is a difference lower than the threshold, which means the robot sees a familiar appearance, while a new local view cell can be generated if the current appearance has never been seen. Meanwhile, the pose cell network is driven by the movement of the agent by path integration which has been described in section 2. As the lidar information is generated in a self-centered coordinate system, we propose an egocentric boundary cell model which is analog to the parietal window boundary coding (PWb) cells[25] found in the mammal's brain to tackle it. The egocentric boundary cells' firing rate is then transformed into the allocentric pattern through the ego-allocentric modulation utilizing the orientation information in the pose cell network.

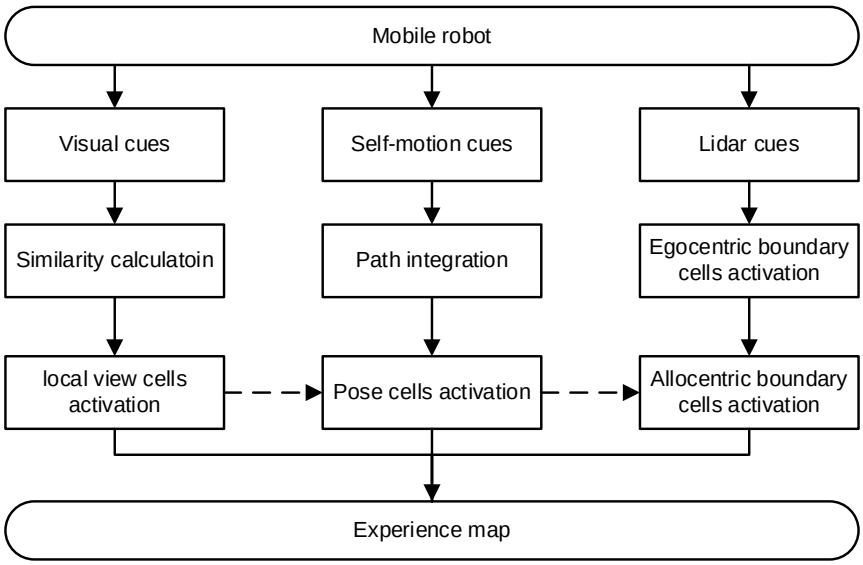


Fig. 2 The framework of the proposed mapping algorithm (The dotted line indicates the transfer of orientation information)

4 Methodology

4.1 Egocentric boundary cells model and ego-allocentric modulation

During operation, the data received from lidar at time t is an array of distances to obstacles $\boldsymbol{l}_t = \{l_{t1}, \ldots, l_{tn}\}$ while each l_{ti} corresponding to a fixed angle which takes the agent's orientation as the positive of the x-axis. Therefore, the structural information provided by LiDAR is self-centered. The egocentric boundary cells model is constructed to process it and every cell has its receptive field (ρ, θ) which responds to the boundary section within it. According to the firing pattern of boundary cells, the computational model of a cell's firing rate is described as follows [24]:

$$r_e^k = \frac{1}{\rho}\exp\left(-\left(\frac{\theta_k - \theta_b}{\sigma_s}\right)^2\right)\exp\left(-\left(\frac{\rho_k - \rho_b}{\sigma_\rho}\right)^2\right) \quad (4)$$

where (ρ_b, θ_b) is the coordinate of a boundary section in the robot coordinate system, while (ρ_k, θ_k) is the receptive field of the k-th egocentric boundary cell [25]. The firing rate of a cell is proportional to the distance and orientation of the obstacle.

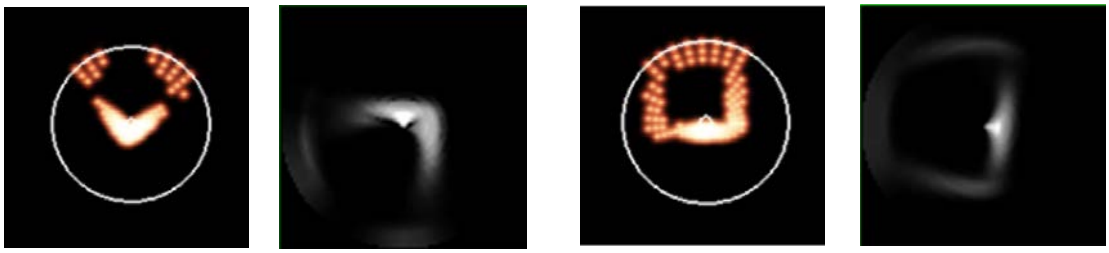
Inspired by the gain-field circuit in the retrosplenial cortex [26], head direction cells provide a transformation between egocentric and allocentric representation, which is similar to the

transformation matrix used in traditional SLAM. Instead of directly providing the representation in the world coordinate system, the information is converted into the allocentric frame. When the robot is in the mapping process, the odometer information is integrated by the pose cells. Thus, the location of the centroid of the activity package in the theta-axis denotes the integral result of all historical orientations, which can be considered as the current robot's head direction. To simulate the function of the gain-field circuit, we propose a model to realize the ego-allocentric modulation with the help of the head direction information in the pose cell network θ_{HD}. The firing rate of the allocentric boundary cell is as follows:

$$r_a^j = \sum_{i=0}^{N_{oc}} r_e^i \delta\left(\theta_i + \theta_{HD} - \theta_j\right) \tag{5}$$

where θ_i and θ_j are the angular receptive field of the i-th egocentric and the j-th allocentric cells, respectively. $\delta(x)$ is the impulse function.

Fig.3 shows the results of ego-allocentric modulation. According to the firing rate of head-oriented cells, the conversion from egocentric to allocentric firing rate is realized.



egocentric allocentric egocentric allocentric

(a) (b)

Fig. 3 Results of ego-allocentric modulation

(a) The robot is in a corner and faces northwest. (b) The robot moves towards the north.

4.2 Experience map building

In the mapping process, every vertex of the map contains the spatial information that the robot experienced at this location. Therefore, to save resources, an experienced node should be created when the perceived information has been largely changed, compared to the current node. In this paper, we use an experience similarity score S to evaluate whether a node needs to be generated. Since the appearance in some places, like a long corridors, it's inadequate to estimate the change using only visual cues. Meanwhile, when there is no node in the system that can sufficiently describe the current experience, it means either the robot is distant from the existed nodes or the appearance is changed. Thus, the states of pose cells and local view cells are applied to determine the score. The experience similarity score is calculated as follows:

$$S = \mu_V S_V + \mu_P S_P \tag{6}$$

where S_V and S_P denotes the similarity between the current states of pose cells and local view cells and the previous activity patterns preserved in the current experience node while μ_V and μ_P are the weight of corresponding cells. Considering the distance influence, the measurement of similarity of pose cells' state is defined as follows:

$$S_P = \begin{cases} 0, & d \geq 2 \\ 2 - d, & d < 2 \end{cases} \tag{7}$$

$$d = \frac{\sqrt{\left(x_{pc}^{'} - x_i^{'}\right)^2 + \left(y_{pc}^{'} - y_i^{'}\right)^2 + \left(\theta_{pc}^{'} - \theta_i^{'}\right)^2}}{r_c} \tag{8}$$

where are the coordinates of the centroid of the

largest activity packet in the pose cell network, $\left(x_i^{'}, y_i^{'}, \theta_i^{'}\right)$ is the pose cell's location which is connected with the currently activated experience node e_i, and r_c is the distance constant.

When the system receives external visual cues, the corresponding local view cells are activated, and the scoring metric of the corresponding local view cells is

$$S_V = \begin{cases} 0, & V^i \notin \boldsymbol{V}^{curr} \\ 1, & V^i \in \boldsymbol{V}^{curr} \end{cases} \tag{9}$$

where $\boldsymbol{V}^{curr}$ is a set of currently activated local view cells, and V^i is the cell that is linked to the current experience node. Thus, if the experience similarity score is less than the threshold $S_{\max}$, a new experience e_j will be created and linked to the previously activated experience node.

$$\boldsymbol{e}_j = \left\{ \mathbf{p}^i + \Delta\mathbf{p}^{ij}, P^{curr}, V_{max}^{curr}, \boldsymbol{B}^{curr} \right\} \tag{10}$$

where P^{curr} is the index of the cell with the largest activity level in the pose cell population, and V_{max}^{curr} is index of the most active local view cell. Meanwhile, $\boldsymbol{B}^{curr}$ denotes the population activity pattern of allocentric boundary cells, which indicates that the firing rates of all cells are recorded. The firing rates are originally stored in a matrix, but to demonstrate the environmental structure intuitively, we transfer the data format as an image. In this population snapshot, cells are arranged in the polar coordinates based on their receptive fields. $\Delta\mathbf{p}^{ij}$ is a vector to describe the movement between the newly created and the previous experience nodes, which also means the length of the edge l_{ij} in the experience map.

$$\Delta\mathbf{p}^{ij} = \mathbf{p}^j - \mathbf{p}^i = \begin{pmatrix} x_j \\ y_j \\ \theta_j \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \\ \theta_i \end{pmatrix} = \begin{pmatrix} \Delta x_{ij} \\ \Delta y_{ij} \\ \Delta\theta_{ij} \end{pmatrix} \tag{11}$$

If the robot revisits a place along an exit path, information in the edge is updated by averaging the odometry data:

$$\Delta\mathbf{p}_{new}^{ij} = A \cdot \Delta\mathbf{p}_{old}^{ij} - B \cdot \Delta\mathbf{p}_{curr}^{ij} \tag{12}$$

$$\boldsymbol{A} = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & \Delta s\cos\Delta\theta & -\Delta s\sin\Delta\theta \\ 0 & \Delta s\sin\Delta\theta & \Delta s\cos\Delta\theta \end{pmatrix} \boldsymbol{B} = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{22}$$

$$\Delta\theta = \frac{1}{2}\left[\tan^{-1}\left(\frac{\Delta y_{curr}^{ij}}{\Delta x_{curr}^{ij}} \right) - \tan^{-1}\left(\frac{\Delta y_{old}^{ij}}{\Delta x_{old}^{ij}} \right) \right] \tag{13}$$

$$\Delta s = \left(d_{curr}^{ij} + d_{old}^{ij} \right) \Big/ \left(2d_{old}^{ij} \right) \tag{14}$$

4.3 Map correction

In the experience map, each experience node not only receives links from other experiences but also sends out links to one or more nodes. Therefore, the correction of the pose of each node needs to integrate the information of all the connected experiences. Thus, the implementation of correction is as follows

$$\Delta\mathbf{p}^i = \alpha \left[\sum_{j=1}^{N_f} \left(\mathbf{p}^j - \mathbf{p}^i - \Delta\mathbf{p}^{ij} \right) + \sum_{k=1}^{N_t} \left(\mathbf{p}^k - \mathbf{p}^i - \Delta\mathbf{p}^{ki} \right) \right] \tag{15}$$

where α is the correction rate constant, N_f is the number of links from experience $\boldsymbol{e}_i$ to other experiences, and N_t is the number of links from other experiences to experience $\boldsymbol{e}_i$. The correction of the map is iteratively applied during the operation, making the position of nodes in the map gradually approach a distribution that minimizes the average error of the trajectory. As a result, links between nodes also need to be updated to align with the change in orientation

$$\Delta \mathbf{p}_{new}^{ij} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\Delta\theta_i & -\sin\Delta\theta_i \\ 0 & \sin\Delta\theta_i & \cos\Delta\theta_i \end{pmatrix} \cdot \Delta \mathbf{p}_{curr}^{ij} \quad (16)$$

5 Experiments

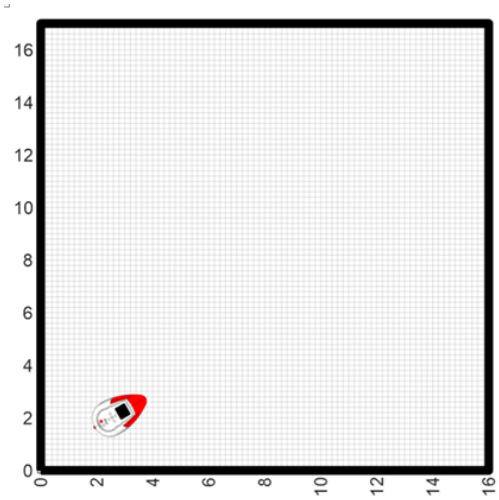
5.1 Experimental settings and criteria

The dataset was generated using MATLAB, simulating a static two-dimensional indoor environment of 16m×17m. The output data of the camera, LiDAR, and odometry sensors are simulated. The frequency of all sensors is 10Hz. Camera data cannot be obtained directly in the simulated two-dimensional environment. In this paper, the environment is divided into 0.2m×0.2m grids, and the robot obtains the output of different visual templates in different grids. The motion of the robot follows the mode of walking against the wall, and its velocity follows the distribution: $N(1,0.05)$ (m/s). When the robot runs in a straight line along the wall, the velocity follows the distribution: $N(1,0.01)$ (rad/s). When the robot reach the corner, the velocity follows the distribution: $0.0982N(1,0.01)$ (rad/s). The LiDAR scanning rate is 400 points per turn, the angle increment is fixed to 0.9°, and the additive noise on distance measurement follows the distribution:

$$noise_{\text{laser}} \sim \begin{cases} N(0,0.005), d \le 1.5 \\ N(0,0.015), d > 1.5 \end{cases} \quad (27)$$

During the operation, the robot moves along the same path, and the sensor data is released almost at the same time in the experiment, without considering the asynchronous problem.

The generated test environment and data format are shown in Fig.4.




```
path:       mapping.bag
version:    2.0
duration:   3:37s (217s)
start:      Jun 25 2021 20:34:16.50 (1645792456.50)
end:        Jun 25 2021 20:37:53.78 (1645792673.78)
size:       15.2 MB
messages:   8692
compression: none [20/20 chunks]
types:      nav_msgs/Odometry        [cd5e73d190d741a2f92e81eda573aca7]
            ratslam_ros/EgoCues      [595417e279b603926045805e4631ff91]
            ratslam_ros/ViewTemplates [b4b1d645c31606cb3d41fe361c493814]
            rosgraph_msgs/Log        [acffd30cd6b6de30f120938c17c593fb]
            std_msgs/Int32           [da5909fbe378aeaf85e547e830cc1bb7]
topics:     /egocues     2171 msgs   : ratslam_ros/EgoCues
            /odom        2171 msgs   : nav_msgs/Odometry
            /rosout         4 msgs   : rosgraph_msgs/Log
            /rosout_agg     4 msgs   : rosgraph_msgs/Log
            /seq_num     2171 msgs   : std_msgs/Int32
            /vtid        2171 msgs   : ratslam_ros/ViewTemplates
```

Fig. 4 The test environment: (left) the visualization of the experimental environment; (right) the information of the dataset (the topics of egocues, odom, and vtid submitter to the LiDAR, odometry and camera outputs)

Since the coordinates of the built map usually is not consistent with the given ground truth coordinate system, a transformation matrix $\boldsymbol{S}$ is implied to align them. Therefore, the absolute trajectory error (ATE) for the i-th frame is defined as follows:

$$\boldsymbol{F}_i = \boldsymbol{Q}_i^{-1}\boldsymbol{S}\boldsymbol{P}_i \quad (28)$$

where $\boldsymbol{P}_i$ represents the pose of the robot of the

i-th frame estimated by the algorithm while $\boldsymbol{Q}_i$ represents the ground truth pose of the robot. However, when evaluating the performance of an algorithm, the root means square error (RMSE) is usually used to deal with ATE:

$$\mathrm{RMSE}\left(\boldsymbol{F}_{1:n},\Delta\right)=\left(\frac{1}{m}\sum_{i=1}^{m}\left\|\mathrm{trans}\left(\boldsymbol{F}_i\right)\right\|^2\right)^{\frac{1}{2}} \tag{29}$$

where Δ represents the time interval and trans($\boldsymbol{F}_i$) represents the translational part of the absolute trajectory error.

5.2 Results and discussion

Fig.5 shows the comparison of the experience map built by the proposed method, the result of path integration in the pose cell network, and the dead reckoning trajectory based on the raw odometry information. The dead reckoning is suffered from sensor noise which leads to a large drift. Thus, it cannot form a consistent trajectory, and the inconsistency will affect robot localization and task performance. Since the path integration of self-motion information is conducted in pose cells which receive the calibration cues from local view cells, it can restore the real trajectory, but there are sharp changes in some positions, as shown in Fig.5 A, when a closed loop is detected. Among them, the experience map achieves the best performance in consistency, which effectively correct the drift caused by sensor noise. As the rotational movement leads to a fast change in neurons' states, more experience nodes are created when the angular velocity increases. In Fig.5 B, the density of the experience node in the up-left corner is much higher than in other places.

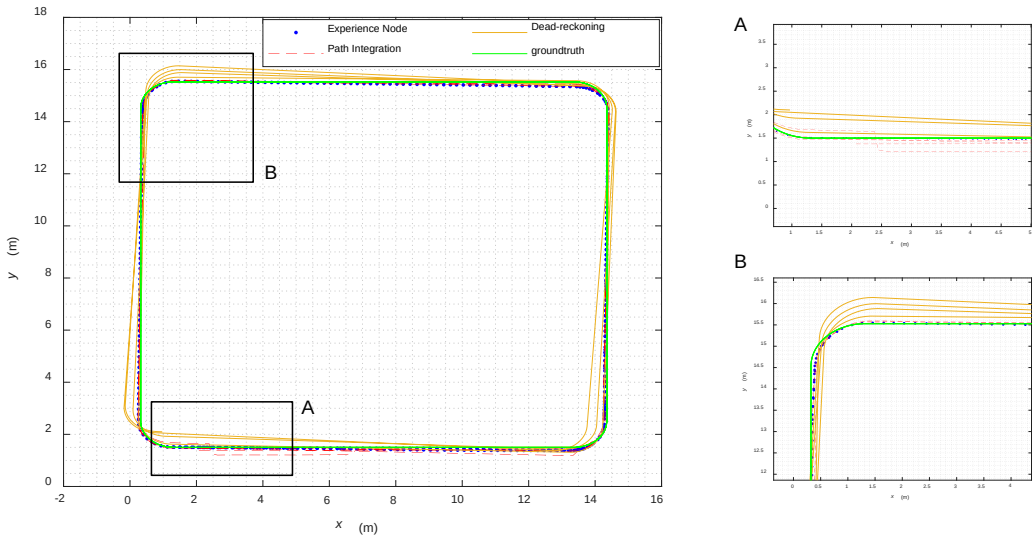


Fig. 5 Comparison of trajectory

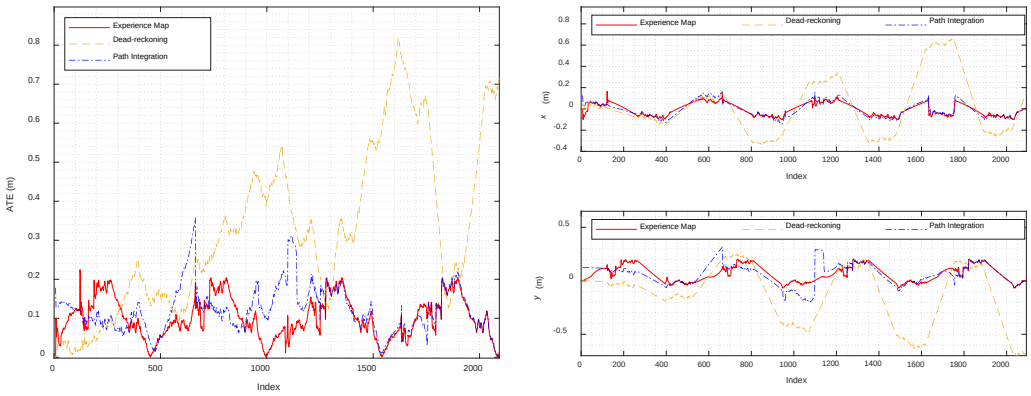


Fig. 6 Comparison of absolute trajectory error; The index corresponds to the frame number of the camera. Since the frequency is fixed, the index is linearly related to the running time

Fig.6 shows the comparison of the absolute trajectory errors of the three algorithms to construct the map. In the left column, as the cumulative displacement of time increases, the dead reckoning error also accumulates. While the path integration result has the largest error at first, with the robot's repetitive visits, the performance is close to the experience map which realizes the loop-closure and calibrates the trajectory. Therefore, the error of the proposed method shows a periodic change and converges to a size of about 0.1m. Meanwhile, in the right column, the errors of the trajectories are calculated from the x and y directions, respectively. The errors of dead reckoning share the same pattern as before, while the errors of the path integration and the experience map fluctuate around a value. Additionally, in the y-axis direction, the path integration result shows two large jumps in the interval of (600, 800) and (1000, 2000), which are due to the loop correction illustrated in Fig.5 B, while the amplitude of the experience map is relatively stable.

Table 1 shows the statistical data of absolute trajectory error of experience map, path integration, dead reckoning, and ground truth. It can be seen that the statistical results of the proposed method are better than the others in all three indicators. The span of the data set used in the manuscript is about 4 minutes. If the method runs for a longer period, in the case of repeating the same trajectory, the odometer drift error, will generate a new experience node. When the existing experience node is matched again, the displacement error position introduced by the new experience node can be alleviated by the graph relaxation algorithm. This is due to loop-closure testing will be detected from the current experience node movement is triggered after a certain distance, by adjusting the parameters match experience node density and test environment after repeated several times track experience node has been very intense, under the condition of not introducing additional noise, the algorithm will no longer cause a new experience, so repeating the same motion can guarantee the same trajectory. Therefore, the proposed method will still have a good ability for map construction for a long time.

Table 1 Comparison of absolute trajectory error

	RMSE(m)	Mean(m)	Median(m)
Dead-reckoning	0.4067	0.3372	0.3073
Path integration	0.1340	0.1201	0.1201
Experience map	0.1142	0.1015	0.0935

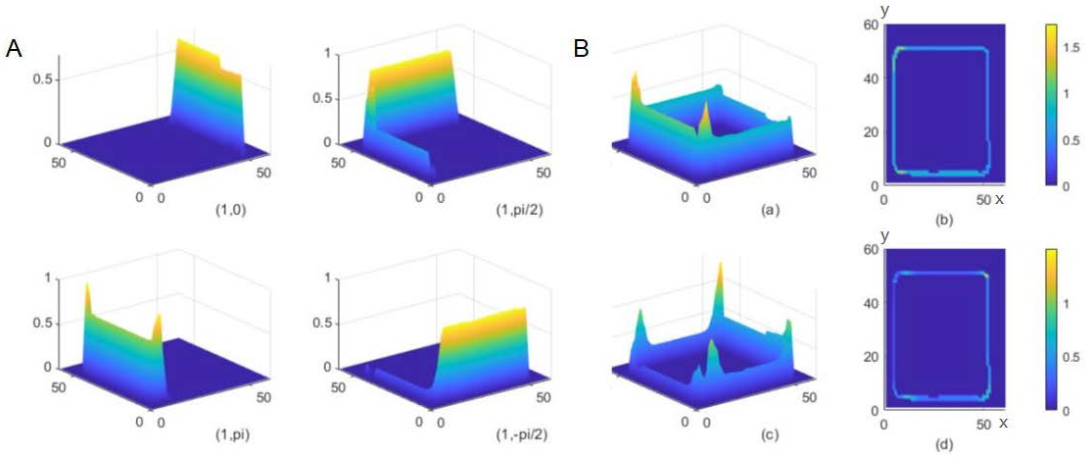


Fig. 7 The firing rate map of allocentric boundary cells in the mapping process

Fig.7 shows the firing rate map of allocentric boundary cells in the environment during the entire robot operation process. In Fig. 7, the x-y plane is corresponding to the scaled coordinate system of the experience map and the z-axis is the firing rate of the cell with

$$x_{firingmap} = x_{experience} * 0.3 \,(\mathrm{m}) \tag{30}$$

$$y_{firingmap} = y_{experience} * 0.3 + 5 \,(\mathrm{m}) \tag{31}$$

Its left part shows the firing rate maps of cells with different receptive fields and the right column is the summation of cells' firing rates. The upper one combines four cells whose receptive fields are: $(1,0)$, $(1,\pi/2)$, $(1,\pi)$, and $(1,-\pi/2)$, while its lower part demonstrates the sum of cells with their receptive fields of $(4,0)$、$(4,\pi/2)$、$(4,\pi)$ and $(4,-\pi/2)$. The centers of the receptive fields of the corresponding cells shown in Fig.7 A are (1, 0), (1, π/2), (1, π) and (1, -π/2), respectively. For a single orientation-independent boundary cell, it is activated only when there is a barrier in the receptive field. Because the robot is set to move along the wall, only one cell in this figure is activated in most cases. And since the shape of the receptive field is a Gaussian distribution, the firing rate is possible to be nonzero in the corner when the robot is close to the wall in this test. In Fig.7B, the upper part shows the population firing rate map by superposing the four figures shown in Fig.7A, while the lower part is similar to it but the radial receptive fields of the cell are changed to be 4. Considering that the distance between the trajectory of the robot and the boundary is ranging from 0.2m to 2m, the cells with a radial receptive field of 1m will obtain higher firing rates, so spatial relationships between detected obstacles and the robot could be indicated from the activation of cells. Thus, according to these two population firing rate maps, we can see the probability of the barrier's existence at 1 or 4 meters away from the robot's current location. In the experiment, multiple boundary cells with different receptive fields are used, so a more accurate understanding and description of the environment structure could be obtained, which enhances the information on the map.

6 Conclusions

In this paper, aiming at the insufficient features in the existing map when working in a visually ambiguous environment, an allocentric boundary cells model is proposed in this paper, which is integrated into the experience map. Allocentric boundary cells generate the orientation-independent structural information through the head direction modulation, and their population activities are fused into the experience map which improves the

functionality of the map. Simulation shows that the proposed model can effectively reflect the boundary structure in the environment, and indicates the obstacle information on the map. At the same time, the algorithm can build a consistent representation of the environment with sensor noise and achieves a root mean square error of 11.42 cm in an indoor environment, which effectively corrects the drift and ensures the accuracy of the map. In the future, it is considered to build the connection between pose cells and every single boundary cell, and explore a sparser expression that makes it possible to spontaneously recover the firing rate pattern in the boundary cells population when activating the experience node, without storing a large number of connection weights. Meanwhile, it is intended to apply the experience map to path planning, so that the robot can judge the traversability of the path according to the boundary cell's state.

7 References

[1] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat," Brain Research, vol. 34, no. 1, pp. 171–175, Nov. 1971.

[2] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," Nature, vol. 436, no. 7052, pp. 801–806, Aug. 2005.

[3] J. Taube, R. Muller, and J. Ranck, "Head-direction cells recorded from the postsubiculum in freely moving rats. II. Effects of environmental manipulations," J. Neurosci., vol. 10, no. 2, pp. 436–447, Feb. 1990.

[4] C. Lever, S. Burton, A. Jeewajee, J. O'Keefe, and N. Burgess, "Boundary Vector Cells in the Subiculum of the Hippocampal Formation," Journal of Neuroscience, vol. 29, no. 31, pp. 9771–9777, Aug. 2009.

[5] A. Arleo and W. Gerstner, "Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity," Biological cybernetics, vol. 83, no. 3, pp. 287–299, 2000.

[6] M. Yuan, B. Tian, V. A. Shim, H. Tang, and H. Li, "An Entorhinal-Hippocampal Model for Simultaneous Cognitive Map Building," p. 7.

[7] Y. Zhou and D. Wu, Spatial Representation and Location Estimation Model Based on Place Cell, Journal of Shanghai Jiao Tong University, no. 04 vo 52, pp. 488–494, 2018.

[8] H. Tang, W. Huang, A. Narayanamoorthy, and R. Yan, "Cognitive memory and mapping in a brain-like system for robotic navigation," Neural Networks, vol. 87, pp. 27–37, 2017.

[9] D. Ball, S. Heath, J. Wiles, G. Wyeth, P. Corke, and M. Milford, "OpenRatSLAM: an open source brain-based SLAM system," Autonomous Robots, vol. 34, no. 3, pp. 149–176, 2013.

[10] M. Milford, A. Jacobson, Z. Chen, and G. Wyeth, "RatSLAM: Using Models of Rodent Hippocampus for Robot Navigation and Beyond," in Robotics Research, Isrr, 2016, vol. 114, pp. 467–485.

[11] M. J. Milford and G. F. Wyeth, "Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System," Ieee Transactions on Robotics, vol. 24, no. 5, pp. 1038–1053, 2008.

[12] A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth, "FAB-MAP + RatSLAM: Appearance-based SLAM for multiple times of day," in 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, May 2010, pp. 3507–3512.

[13] S. Yu, J. Wu, H. Xu, R. Sun, and L. Sun, "Robustness Improvement of Visual Templates Matching Based on Frequency-Tuned Model in RatSLAM," Front. Neurorobot., vol. 14, p. 568091, Sep. 2020.

[14] T. Xu, Y. Ling, and M. Chen, A bio-inspired algorithm integrated with DGSOM neural network. CAAI Transactions on Intelligent Systems, vol. 12, no. 03, pp. 405–412, 2017.

[15] Zeng T, Si B. Cognitive Mapping Based on Conjunctive Representations of Space and Movement[J]. Frontiers in Neurorobotics, 2017, 11: 61.

[16] Gupta S, Davidson J, Levine S, et al. Cognitive

mapping and planning for visual navigation[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2616-2625.

[17] Salman, M., & Pearson, M.J. (2018). Whisker-RatSLAM Applied to 6D Object Identification and Spatial Localisation. *Living Machines*.

[18] Chen, M., & Hu, W. (2019). Research on BatSLAM Algorithm for UAV Based on Audio Perceptual Hash Closed-Loop Detection. *Int. J. Pattern Recognit. Artif. Intell., 33*, 1959002:1-1959002:19.

[19] J. Steckel and H. Peremans, “BatSLAM: Simultaneous Localization and Mapping Using Biomimetic Sonar,” Plos One, vol. 8, no. 1, p. e54076, 2013.

[20]Z. Wu, Y. Yue, M. Wen, J. Zhang, J. Yi and D. Wang, "Infrastructure-Free Hierarchical Mobile Robot Global Localization in Repetitive Environments," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-12, 2021.

[21] O. Struckmeier, K. Tiwari, M. Salman, M. J. Pearson, and V. Kyrki, “ViTa-SLAM: A Bio-inspired Visuo-Tactile SLAM for Navigation while Interacting with Aliased Environments,” in 2019 IEEE International Conference on Cyborg and Bionic Systems (CBS), 2019, pp. 97–103.

[22] M. J. Milford, G. F. Wyeth, and D. Prasser, “RatSLAM: A hippocampal model for simultaneous localization and mapping,” in 2004 Ieee International Conference on Robotics and Automation, Vols 1- 5, Proceedings, 2004, pp. 403–408.

[23] A. Samsonovich and B. Mcnaughton, “Path Integration and Cognitive Mapping in a Continuous Attractor Neural Network Model,” The Journal of neuroscience : the official journal of the Society for Neuroscience, vol. 17, pp. 5900–20, Sep. 1997.

[24] Bicanski A, Burgess N. A neural-level model of spatial memory and imagery[J]. eLife, 2018, 7: e33752.

[25] A. Bicanski and N. Burgess, “A neural-level model of spatial memory and imagery,” eLife, vol. 7, p. e33752, Sep. 2018.

[26] A. Bicanski and N. Burgess, “Environmental Anchoring of Head Direction in a Computational Model of Retrosplenial Cortex,” J Neurosci, vol. 36, no. 46, pp. 11601–11618, Nov. 2016.

Authors

Yixuan Long has obtained her Master's Degree in College of Information and Communication Engineering, Harbin Engineering University. Her research interest is Simultaneous Localization and Mapping.

Hao Wang is pursuing his Master's Degree in College of Information and Communication Engineering, Harbin Engineering University. His research interest is Simultaneous Localization and Mapping.

LI Yibing received his Ph.D. Degree from Harbin Engineering University in 2003. He is a professor in the School of Information Engineering, Harbin Engineering University. His research interests include communication signal processing, radio navigation and positioning.

YE Fang received her Ph.D. Degree from Harbin Engineering University in 2006. She is currently working in the College of Information and Communication Engineering, Harbin Engineering University. Her research interests include cognitive confrontation and intelligent decision-making.

Qian Sun received his Ph.D. Degree from the School of Automation, Harbin Engineering University in 2016, and is currently a tutor of Master's Degree at the School of Information and Communication Engineering, Harbin Engineering University. His research interest is focused on collaborative localization.