Cognitive Neurodynamics (2021) 15:91–101
https://doi.org/10.1007/s11571-020-09621-6

RESEARCH ARTICLE

A brain-inspired compact cognitive mapping system

Taiping Zeng[1,2] · Bailu Si[3]

Received: 19 June 2020 / Revised: 7 July 2020 / Accepted: 20 July 2020 / Published online: 30 July 2020

Abstract
In many simultaneous localization and mapping (SLAM) systems, the map of the environment grows over time as the robot explores the environment. The ever-growing map prevents long-term mapping, especially in large-scale environments. In this paper, we develop a compact cognitive mapping approach inspired by neurobiological experiments. Mimicking the firing activities of neighborhood cells, neighborhood fields determined by movement information, i.e. translation and rotation, are modeled to describe one of the distinct segments of the explored environment. The vertices with low neighborhood field activities are avoided to be added into the cognitive map. The optimization of the cognitive map is formulated as a robust non-linear least squares problem constrained by the transitions between vertices, and is numerically solved efficiently. According to the cognitive decision-making of place familiarity, loop closure edges are clustered depending on time intervals, and then batch global optimization of the cognitive map is performed to satisfy the combined constraint of the whole cluster. After the loop closure process, scene integration is performed, in which revisited vertices are removed subsequently to further reduce the size of the cognitive map. The compact cognitive mapping approach is tested on a monocular visual SLAM system in a naturalistic maze for a biomimetic animated robot. Our results demonstrate that the proposed method largely restricts the growth of the size of the cognitive map over time, and meanwhile, the compact cognitive map correctly represents the overall layout of the environment. The compact cognitive mapping method is well suitable for the representation of large-scale environments to achieve long-term robot navigation.

Keywords SLAM · Compact cognitive map · Long-term mapping · Neighborhood cells · Neighborhood fields

Introduction

Spatial cognition endows mammals with impressive long-term navigation capabilities, such as homing (Mittelstaedt and Mittelstaedt 1980), or migrating between seasonal habitats (Naidoo et al. 2016). Mammals surpass robots with better by far navigation performance in large-scale dynamic environments. It is believed that mammals can learn spatial information from the surrounding environment to form an internal map-like representation in the brain, namely cognitive map, to help mammals navigate in the complex environments (Tolman 1948).

Neurobiological studies have discovered that the hippocampal-entorhinal neural circuit plays a key role in spatial cognition. Neurons in these areas increase and decrease in electrical activities to encode various allocentric information of the animal while navigating in environment (McNaughton et al. 2006; Moser et al. 2008, 2015). These neurons, including place cells (O'Keefe and Dostrovsky 1971), grid cells (Hafting et al. 2005), head direction cells (Taube et al. 1990), speed cells (Kropff et al. 2015), boundary cells (Lever et al. 2009), and etc., provide detailed representations of self and environments. In the behavioral level, coarser and more abstract information contributes to spatial navigation. Humans often reason about space in high-level concepts, i.e.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11571-020-09621-6) contains supplementary material, which is available to authorized users.

✉ Bailu Si
bailusi@bnu.edu.cn

Taiping Zeng
zengtaiping@fudan.edu.cn

1 Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai, China

2 Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence (Fudan University), Ministry of Education, Shanghai, China

3 School of Systems Science, Beijing Normal University, Beijing 100875, China

topographical orientation. A typical example goes like: turn left at the supermarket, down the road, and turn right at the convenience store. In recent years, neurons, called neighborhood cells, are found in the perirhinal cortex, providing strong support for topographical orientation (Bos et al. 2017). Neighborhood cells fire persistently in the large area of the environment, and encode navigation behavior at larger scales. Neighborhood cells, therefore, help the brain differentiates distinct segments of the environment, a function beyond detailed representations provided by place cells and grid cells.

In the field of robotics, mapping the environment is a prerequisite for autonomous mobile robots to carry out tasks like transportation, delivery, search and rescue, especially in indoor or underground environments where global positional information, such as GPS, is not easily available. Algorithms were developed to construct a topological graph mapping the structure of the environment. Based on topological graphs, robots are able to plan trajectories and navigate to goal locations. While, in most of the existing graph-based methods, the complexity of maps grows quickly with the length of the robot's trajectory (Kretzschmar and Stachniss 2012). As new vertices and edges are constantly added to the map, the demand for computational time and memory footprint grows over time, preventing the applications in long-term mapping. So, approaches to control the size of the map, i.e. compact map, are key to practical robotic applications that involve continuous exploration in environments (Cadena et al. 2016).

The mechanisms of mammalian spatial cognition show a great potentiality to inspire novel algorithms to help to improve the navigation ability of mobile robots. In this study, following our previous model (Zeng and Si 2017), which represent the allocentric position and orientation within the environment with grid cells, and head direction cells, we focus now on how compact cognitive map efficiently encodes the surrounding environments and sparsely store information when a mobile robot explores a new environment.

To develop compact cognitive map representations that enable long-term navigation, we set the basis for several neurobiological findings. First, the novelty or discontinuity in the incoming sensory streams increases the firing activities of CA1 neurons in the Hippocampus (Larkin et al. 2014). This novelty signal is important for neighborhood cells to group information in chunks and only encode behaviorally relevant regions in the environment. Second, mammals rethink where they are, as manifested by the awake replay of neural activities during navigation (Carr et al. 2011). Replay can be initiated by sensory inputs. The reactivation of cell assemblies enables map learning and reorganization of the cognitive map. Due to the fact that awake replay is triggered only sparsely, the optimization of a cognitive map could be performed for each loop closure cluster, in which new sensory information should be integrated into the existing cognitive map. Third, place cells and neighborhood cells show robust firing activities. When familiar places are revisited, the same population of cells is activated again. There is no need to recruit extra cells for the encoding of familiar places. In a compact cognitive map, scenes from familiar places should be integrated without adding new vertices.

Inspired by neural mechanisms of spatial cognition, we develop in this paper a pragmatic compact cognitive mapping solution to control the growth of the size of the cognitive map. Compared with other approaches (Ball et al. 2013; Zeng and Si 2017; Zeng et al. 2020), the main contribution of this paper is four-fold.

First, we introduce a concept of neighborhood fields to segment the explored environment, mimicking the high-level representations of neighborhood cells. The sizes of neighborhood fields are dependent on movement information, and in turn, determine whether new vertices and edges are added to the cognitive map or not. Since the neighborhood field is a mechanism to sparsify the cognitive map, there is no need to partition the environment to reduce the number of vertices in the map. This pragmatic approach allows us to gently trade accuracy for computational efficiency including computational time and memory footprint, yet still, keep the topology information and the fidelity of representation of the environment.

Second, we formulate the optimization of the cognitive map as a constrained robust non-linear least squares problem. The objective is to minimize the inconsistency based on the local geometrical relationships experienced during exploration. The optimization is triggered once for each loop closure cluster, in which a familiar place is revisited and odometry error could be minimized most efficiently given the new information of previously experienced places.

Third, during loop closure, the multiple correspondences between currently experiencing places and previously experienced places are clustered into one edge. The clustering criteria are based on time information. The clustering of overlapping vertices and edges into one connection reduces the redundancy of the cognitive map. In addition, during loop closure, redundant vertices and edges are removed to ensure no extra neural coding is added to represent the familiar places.

Fourth, we develop a monocular visual SLAM system to evaluate the proposed compact cognitive mapping algorithm. The mapping performance of this monocular visual SLAM system is demonstrated on an iRat rodent-sized robot platform in a naturalistic maze (iRat 2011 Australia dataset) (Ball et al. 2013). Experimental results showed that the size of the map is well controlled by our approach over time. The proposed algorithm is suitable for compact cognitive mapping to support long-term robot navigation.

The rest of this paper is organized as follows. We describe our approach in "Method" section. "Experimental results" section presents the experimental results. We discuss results and future works in "Discussion" section. A brief conclusion is given in "Conclusion" section.

Related work

In the context of the SLAM problem, many effective approaches have been proposed to solve robot mapping. Lu and Milios (Lu and Milios 1997) first introduced global map optimization using a pose graph. The graph-based approach models the poses of the robot as vertices, and spatial constraints between poses as edges in a graph. In this standard graph-based approach, the size of the map expands quickly, while the robot explores new areas. As a result, the demand for storage and computational resources increase rapidly. In the worst case, the standard graph-based approach has quadratic growth of memory usage with the number of variables when direct linear solvers are used. There are great efforts to improve the efficiency of graph-based mapping algorithms. The sparsity structure of the matrix in the normal equations is used to enable the fast linear online solvers. Many SLAM libraries, such as g2o (Kümmerle et al. 2011), GTSAM, Ceres (Agarwal et al. 2012), are available to solve this problem with tens of thousands of variables in just a few seconds. However, even using iterative linear solvers, memory consumption grows linearly with the numbers of variables. Revisiting the same place many times makes this situation more complicated. As more vertices and edges are continuously added to the same spatial area, this approach becomes less efficient. For now, there are few works to solve the question of how to store the map for long-term exploration (Cadena et al. 2016). Therefore, it is valuable to achieve a long-term mapping solution that can control, or at least reduce the growth of the size of the map.

One of the most important ways to reduce the complexity of the map is the vertex and edge sparsification, which trades the accuracy of the map for memory and computational efficiency. Information-based compact pose SLAM algorithm (Ila et al. 2010) was proposed in an information-theoretic manner to reject redundant vertices and add informative measurements to the map. An information-based criterion (Kretzschmar and Stachniss 2012) was introduced to determine which laser scan should be marginalized in pose global optimization, which retains the sparsity for laser-based 2D pose graphs. The generic linear constraint factors (Carlevaris-Bianco and Eustice 2013) and the nonlinear graph sparsification (Mazuran et al. 2016) were proposed to achieve a sparse blanket based on the Markov blanket of a marginalized vertex.

Another class of method was proposed focusing on addressing temporal scalability of standard pose graph (Johannsson et al. 2013). This method avoids adding redundant vertices and edges before the global optimization of the graph. Demonstrated on a binocular visual SLAM system in indoor environments, this method is an efficient solution for medium-scale environments, such as buildings.

A map pruning method was proposed in a biologically inspired monocular visual SLAM system, RatSLAM (Milford and Wyeth 2010). The pruning method removes experiences in map regions where the density of experiences exceeds a threshold. When revisiting a familiar view, the current location of the robot is assigned to the vertex corresponding to the familiar view, instead of adding duplicate experience vertices to the map. During loop closure, the experience map is optimized by iterative map relaxation after each loop closure edge is added to the map, and may results in low optimization efficiency since there are often dozens of similar loop closure edges that could be established in one loop closure event.

Method

In this section, we develop a brain-inspired compact cognitive mapping approach to control the size of the cognitive map over the exploration time. The proposed compact mapping algorithm includes several major computational blocks that work together. First, the global optimization of the cognitive map is formulated as a solution of non-linear least squares problems. A sparse solver is applied to solve the normal equations with high-performance Ceres Solver (Agarwal et al. 2012). Second, when the robot comes across novel image views, inspired by neighborhood cells, neighborhood fields are used to drive the recruitment of new vertices and edges. Third, according to the cognitive decision-making of place familiarity, the time intervals between loop closure edges are applied to cluster loop closures edges, and further, the global optimization is carried out to satisfy the combined constraint of the whole cluster. After non-linear least-squares optimization, short edges of the length below a threshold are merged to filter noises from movement and measurement. Finally, when the robot revisits familiar visual scenes, redundant vertices corresponding to the same location are merged. The familiar scenes are integrated. The overview of the brain-inspired compact cognitive mapping algorithm is shown in Algorithm 1.

Algorithm 1: Overview of the Brain-Inspired Compact Cognitive Mapping Algorithm

Robot move around;
if $g(d,\theta) > \delta$ **then**
    Add a vertex;
    if *Visual template matched* **then**
        Add a loop closure edge;
        Cluster loop closure edges;
        if $(t_{interval} > T_{interval}) || (t_{end} - t_{start} > T_{total})$ **then**
            Batch global optimization of cognitive map;
            Scene Integration;
            Removing Short Edges;
            $t_{start} \leftarrow t_{current}$;
            $t_{end} \leftarrow t_{current}$;
            $t_{interval} \leftarrow 0$;
        end
        else
            $t_{end} \leftarrow t_{current}$;
            $t_{interval} \leftarrow t_{current} - t_{current-1}$;
        end
    end
    else
        Add a sequential edge;
    end
end

Formulation of cognitive map optimization

The optimization of the cognitive map is formulated as a constrained robust non-linear least squares problem (Agarwal et al. 2012). Vertices represent locations in the environment. Edges are used to model spatial constraints between vertices. Sequential edges arise from odometry measurements while the robot traverses the vertices. And loop closure edges arise from visual template matching, which is a process to mimic visual recognition of familiar places (Zhao et al. 2019). Formally, optimization of the cognitive map is to find a solution to

$$\min_{\mathbf{e}} \quad \frac{1}{2}\sum_{i,j} \rho_i\left(\left\|f_i(e_i, e_j, e_{ij})\right\|^2\right), \tag{1}$$

where $\mathbf{e}$ is the set including all vertices $e_i, 1 \le i \le N$. N is the number of vertices in the cognitive map. $e_i = (x_i, y_i, \theta_i)$ is the vertex state, i.e. spatial coordinates and orientation of the robot when it passes the place. The objective function is minimized given the constraints defined by edges e_{ij}. $e_{ij} = (x_{ij}, y_{ij}, \theta_{ij})$ describes the transition from vertex e_i to e_j, obtained from movement information or visual template matching. $\rho_i\left(\left\|f_i(e_i, e_j, e_{ij})\right\|^2\right)$ is a residual block, where $f_i(\cdot)$ is a cost function. ρ_i is a loss function, which can be applied to decrease the influence of outliers on the global optimization of non-linear least squares problems. Here, Huber Loss is used, which can be defined as

$$\rho(s) = \begin{cases} s & quads \le 1 \\ 2\sqrt{s} - 1 & s > 1 \end{cases}. \tag{2}$$

More specifically, cost function $f_i(\cdot)$ for a pair of vertices e_i and e_j connected by an edge e_{ij} is computed by

$$\begin{aligned} f_i(e_i, e_j, e_{ij}) &= [\, e_j - e_i - e_{ij} \,] = \begin{bmatrix} x_j - x_i - x_{ij} \\ y_j - y_i - y_{ij} \\ \theta_j - \theta_i - \theta_{ij} \end{bmatrix} \\ &= \begin{bmatrix} x_j - x_i - d_{ij} \cdot \cos(\theta_i + \theta_{ij} + \theta_{relative}) \\ y_j - y_i - d_{ij} \cdot \sin(\theta_i + \theta_{ij} + \theta_{relative}) \\ \theta_j - \theta_i - \theta_{ij} \end{bmatrix}, \\ \text{s.t.} \quad & -\pi \le \theta_i < \pi, \quad -\pi \le \theta_j < \pi, \end{aligned} \tag{3}$$

where d_{ij} is the distance between e_i and e_j. $\theta_{relative}$ is the relative angle between matched visual templates and current visual scenes (Ball et al. 2013). The values of θ_i and θ_j are limited to the range $[-\pi, \pi)$.

The constrained optimization problem is solved by Ceres Solver (Agarwal et al. 2012). It is an open-source C++ library for modeling and solving large complicated optimization problems, especially for Non-linear Least Squares problems. During mapping, once a loop closure is detected, Ceres Solver is called to compute a solution to Eq. (1) given the geometric information brought by the new clustered edge in "Clustering loop closure edges" section.

Adding sparse vertices through neighborhood fields

To support topographical orientation, mammals are able to represent environments at a high level. In this paper, neighborhood cells (Bos et al. 2017) are modeled to differentiate distinct segments of the environment. The firing fields of neighborhood cells, called neighborhood fields, are introduced to describe distinct segments of the environment. Although in the real situation, the neural representation of a distinct segment may be influenced by visual marks, odor, sound, movement, etc., here, in the cognitive map model, we only consider movement information to determine whether the neighborhood field is strong enough to form a distinct segment. The movement information of mammals includes translation d and rotation θ. Translation d stands for the distance in meters traversed by the robot between the tentative vertex and the previous vertices. Rotation θ stands for the absolute amount of rotation of the robot's head direction in radians between the two vertices. The neighborhood field is defined by

$$g(d, \theta) = (1 + \alpha \cdot d)(1 + \beta \cdot \theta), \tag{4}$$

where, α is the weight for translation d, and β is the weight for rotation θ. $g(d, \theta)$ is high either the translation or the rotation is large, therefore it is an indicator of novelty. If the robot makes large enough movement, $g(d, \theta)$ comparing with the previous vertex is novel enough to create a new neighborhood field, i.e. $g(d, \theta) > \delta$, where δ is the threshold, and a new vertex will be added to the cognitive map. In other words, translation d and rotation θ jointly decide whether new information is added to the cognitive map. If the translation d is large enough to provide novel information, a new vertex is also added to the cognitive map, regardless of rotation θ. When the robot makes a turn, new views should be remembered, even if the translation is small.

We further illustrate in details the sparsification technique using the concept of neighborhood field. Compared with the standard approach (Ball et al. 2013; Zeng and Si 2017), if the neighborhood field $g(d, \theta)$ does not meet the threshold to create a new vertex, the tentative vertex e_{i+1} is removed, and edges $e_{i,i+1}$ and $e_{i+1,i+2}$ are merged into $e_{i,i+2}$ (Fig. 1). The threshold is an inhibition mechanism. The neighborhood field of the tentative vertex e_{i+1} is not strong enough to overcome the inhibition from the previous vertex e_i, and therefore is not added to the cognitive map. If many consecutive vertices do no overcome the inhibition, i.e. neighborhood field activities below threshold, they are removed and their corresponding edges are merged into one, which is shown in Fig. 2.

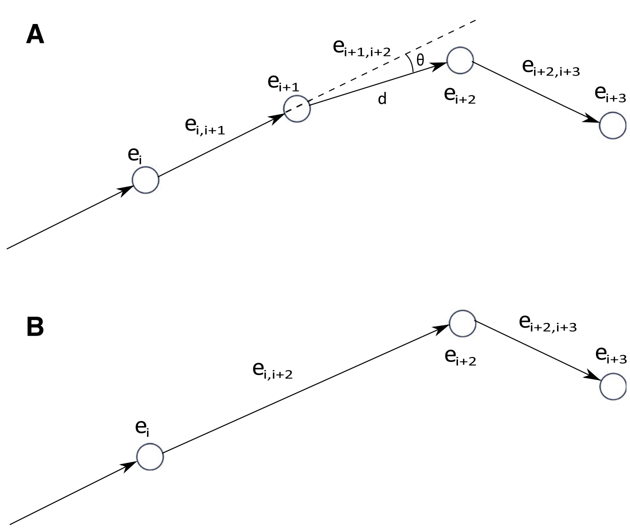

Fig. 1 Adding sparse sequential vertices and edges using neighborhood field. **a** Standard cognitive map with new sequential vertices and edges; **b** Compact cognitive map after removing vertices whose neighborhood fields are weak

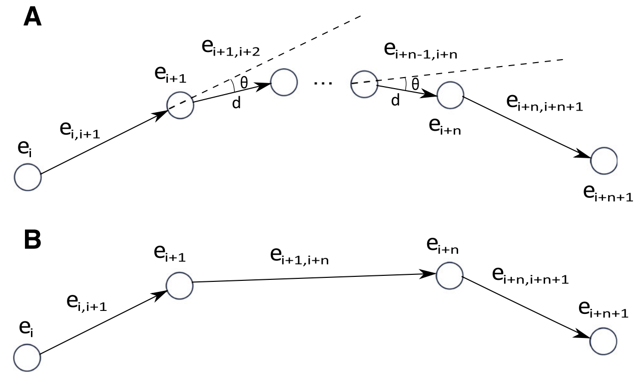

Fig. 2 Adding a number of consecutive sequential vertices. **a** shows standard cognitive map; **b** shows compact cognitive map

Clustering loop closure edges

In the hippocampus, "time cells" encode successive moments during an empty temporal gap between key events (MacDonald et al. 2011; Eichenbaum 2014). Inspired by time cells, time intervals are introduced to help search matched visual scenes for clustering. According to the perceptual decision-making process, whether the animal is located in a previously visited place is decided by consecutive familiar scene views in their memory, therefore loop closure edges are clustered into one edge. After enclosing the clustered edge into the cognitive map, optimization of the cognitive map is performed once, so that there is no need to perform optimization for every loop closure edge.

An incremental procedure is used to group loop closure edges into a cluster based on timestamps. When the first loop closure edge is created, a cluster is initialized and the start time (t_{start}) of the current cluster is the timestamp of the first loop closure edge. If the time interval between the current loop closure edge (t_{end}) and the previous loop closure edge (t_{end-1}) is smaller than a threshold of time interval ($T_{interval}$), the current loop closure edge would be added into the current cluster. Otherwise, a new cluster would be created. If the total time between the current loop closure edge (t_{end}) and the first loop closure edge (t_{start}) is greater than a threshold of total time (T_{total}), the current loop closure edge would be added into a new cluster. An illustration of clustering is shown in Fig. 3. Cluster 1 is a normal situation in which time interval and total time are both greater than the threshold of time interval ($T_{interval}$) and total time (T_{total}) relative to the next loop closure edge, which is segmented into the next cluster. Total time longer than threshold leads to the division of consecutive loop closure edges into cluster 2 and cluster 3. Since time interval is longer than the threshold, cluster 4 and cluster 5 are created.

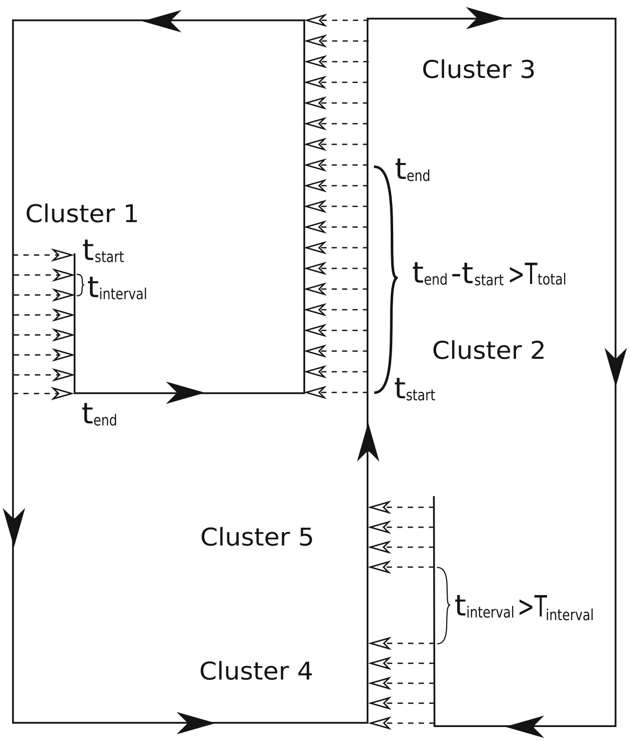

Fig. 3 Clustering loop closure edges. According to the thresholds of the time interval and total time, loop closure edges are clustered. The solid line shows the trajectory of the robot, on which the arrows are the movement directions. The parallel dashed arrows are the loop closure edges from new vertices to old vertices

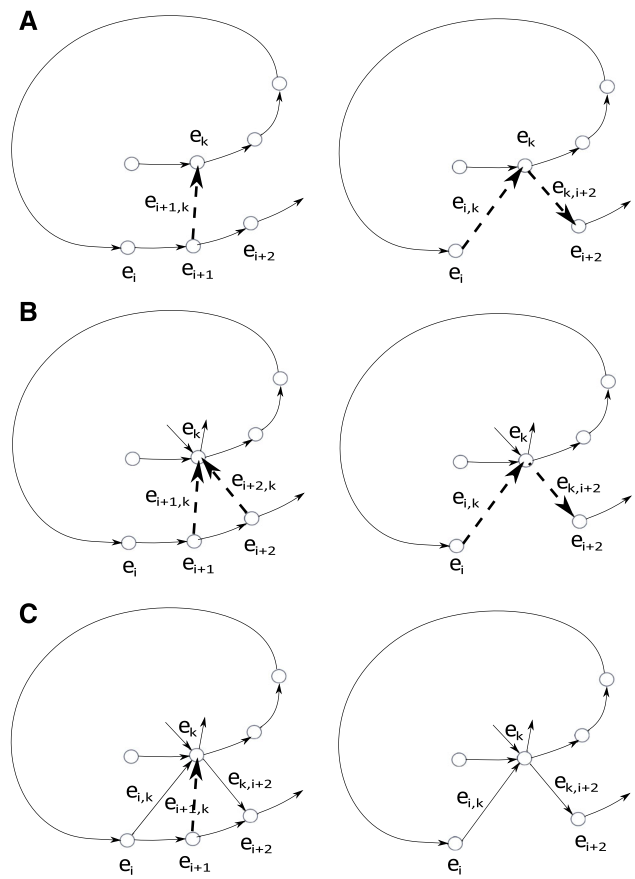

Fig. 4 Scene Integration. Standard cognitive map is on the left. Compact cognitive map is on the right. **a–c** show three different cases we remove revisited vertices. Dashed thick arrows stand for the newly added loop closure edges

Scene integration

When mammals revisit familiar environments, the current scene would be integrated with previous old scenes stored in the memory. For the robot, to achieve long-term mapping, one critical issue is to control the size of the cognitive map to be bounded by the size of the explored environment and be independent of the exploration time. Equivalently, the size of the map does not grow unless the unknown environment is explored. Therefore, when the robot revisits familiar image views, redundant vertices should not be added into, and visual scenes should be integrated consistently into the old scenes. Scene integration is performed after the optimization of the cognitive map, different from the method that avoids adding redundant vertices to begin with Johannsson et al. (2013).

We illustrate this reduction approach in Fig. 4. Dashed thick arrows stand for newly added loop closure edges for the current moment. In Fig. 4a, new vertex $e_{i,i+1}$ is found to be the same location as vertex e_k, therefore it is removed, and the edges $e_{i,i+1}$, $e_{i+1,i+2}$ and $e_{i+1,k}$ are merged into two edges $e_{i,k}$ and $e_{k,i+2}$. Situation shown in Fig. 4b often happens, where multiple vertices connect to the same vertex. After scene integration, the multiple vertices are reduced to one vertex, and the edges are merged accordingly. As for Fig. 4c, the redundant vertex $e_{i,i+1}$ is interfused with vertex e_k, and its incoming and outgoing edges are removed.

Removing short edges

When the robot moves around, due to motion noises and measurement errors, redundant vertices might be created. These vertices represent the same information. One of these vertices will be kept, and the others are removed. The transitions between these vertices are small relative to the noises and errors, and convey low topological information. After loop closures, the extra constraints given by these vertices need to be removed. Figure 5 shows how these additional vertices and edges are removed according to the threshold of edge length.

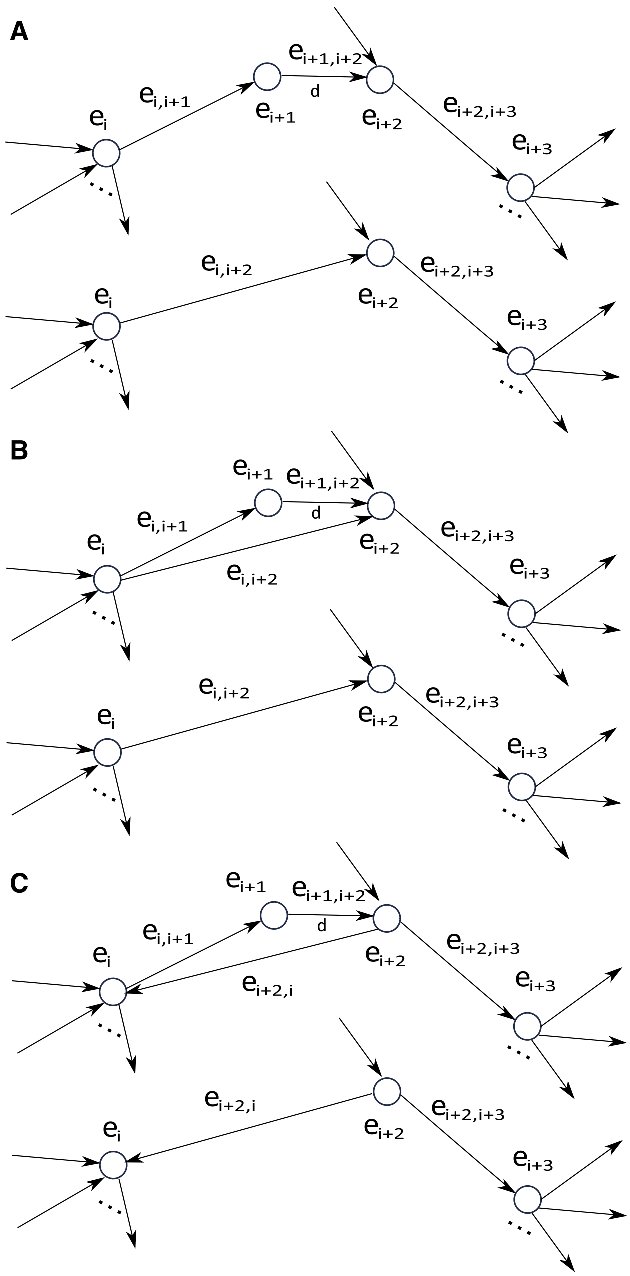

Fig. 5 Removing short edges. Standard cognitive map is at the top. Compact cognitive map is at the bottom. **a** Edge $e_{i+1,i+2}$ length is smaller than threshold in (**a**). Vertex e_{i+1} is removed. Edge $e_{i,i+1}$ and edge $e_{i+1,i+2}$ are merged into $e_{i,i+2}$. **b**, **c** There exist multiple constraints between vertex e_{i+1} and e_{i+2}. We remove edges $e_{i,i+1}$ and $e_{i+1,i+2}$, and vertex e_{i+1}

Visual SLAM system

Our compact cognitive mapping approach is demonstrated on a brain-inspired visual SLAM system. The presented visual SLAM system is improved from our previous work described in Zeng et al. (2020). Here, we employ the above mentioned compact cognitive mapping technique to replace the graph relaxation algorithm for the experience map (Ball et al. 2013).

Experimental results

In this section, we demonstrate our compact cognitive mapping technique on iRat Australia dataset (Ball et al. 2010), a publicly available open-source dataset. Intelligent Rat animate technology, iRat, is a small mobile robot developed to study navigation and embodied cognition for robotic and neuroscience researchers. iRat has a similar size and shape like a rat. Images of the iRat Australia dataset are obtained by a web camera on the iRat robot. The iRat ROS bag provides camera images, odometry messages, and overhead images.

We compare our approach with the method in which no vertices and edges are discarded. The latter is referred to as the standard method and the corresponding cognitive map is referred to as the standard cognitive map. Compact cognitive mapping processes with varying sparsity levels are shown in video s1, s2, s3 in the supplementary materials.

Cognitive map

To better show the ability of our approach to achieving vertex and edge sparsification, we mainly consider the influence of two steps in the algorithms: adding sparse sequential vertices and edges through neighborhood fields, and scene integration. Apart from reducing the number of vertices and edges, clustering loop closure edges also reduce the frequency of executing batch global optimization. Batch global optimization is parallelized based on OpenMP and ensures that the optimization problem can be quickly solved. Since the frequency of executing batch global optimization is not high, the cognitive mapping process is able to run in real-time on a PC, even for large-scale environments. As for the step of removing short edges, it is applied mainly to reduce the possible noises.

In order to qualitatively compare the compact cognitive maps with the actual environment, the overhead image of the explored environment is shown in Fig. 6a, together with the cognitive maps constructed by different methods (Fig. 6b–d). The standard cognitive map created by the visual SLAM system is shown in Fig. 6b. Green circles are the vertices of the cognitive map. Blue thin lines are edges between connected vertices. There are 3911 vertices and 5184 edges entailed in the standard cognitive map. The standard mapping process is shown in video s1 in supplementary materials. The process of clustering loop closure edges and batch global optimization can be clearly seen from the video s1 in the supplementary materials.

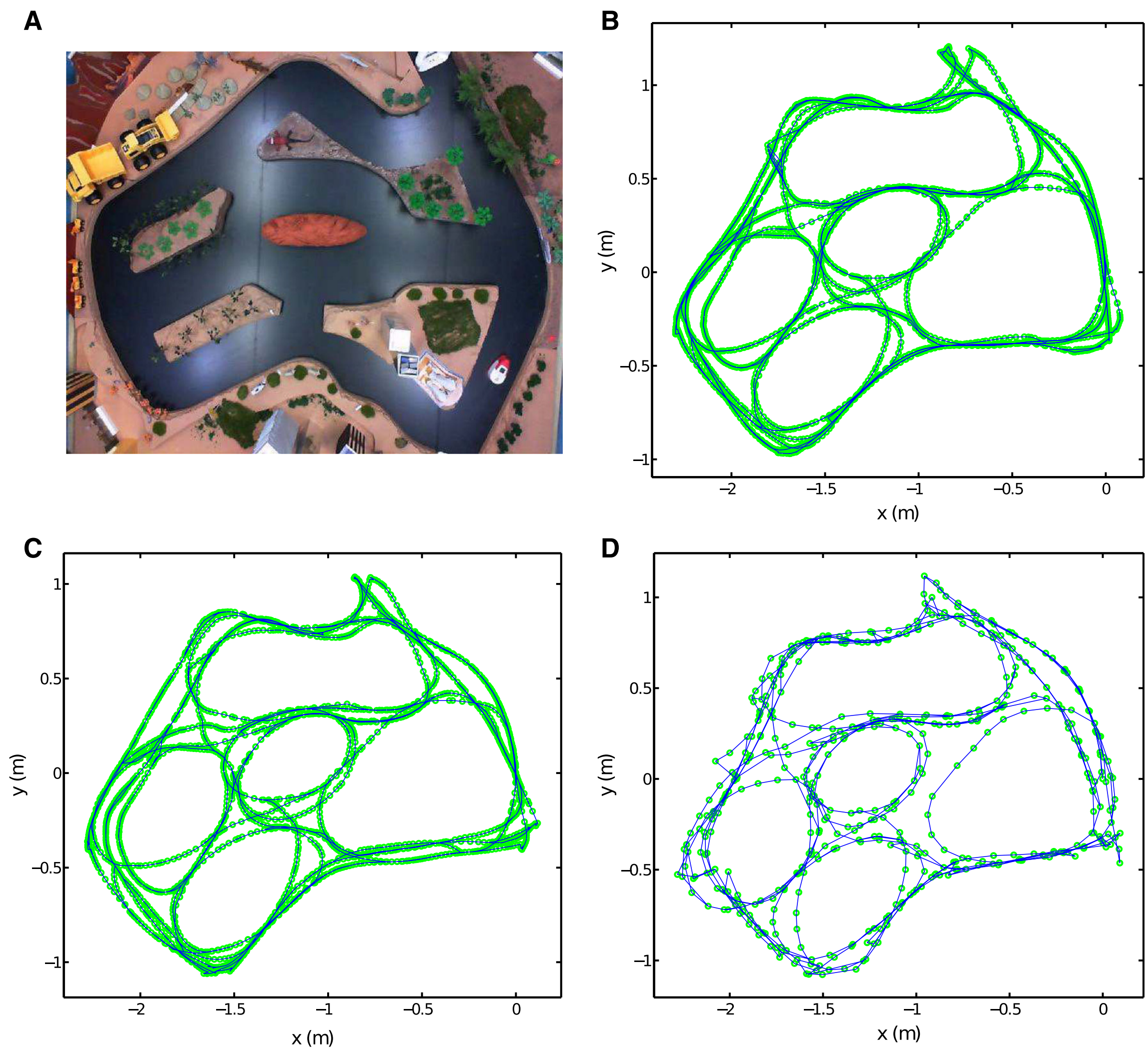

Fig. 6 Cognitive Map. Green dots are vertices of the cognitive map. Blue thin lines are edges between connected vertices. **a** The overhead view of the explored environment. **b** shows the standard cognitive map with 3911 vertices and 5184 edges. **c** shows the compact cognitive map with 2191 vertices and 2152 edges by the step, scene integration. **d** shows the compact cognitive map with 497 vertices and 602 edges by steps: adding sparse vertices and edges, and scene integration. (Color figure online)

When scene integration kicks in cognitive mapping, the total number of vertices in the compact cognitive map reduces from 3911 to 2191, edges from 5182 to 2152. Although the number of vertices in the compact cognitive map (Fig. 6c) is reduced nearly twice compared with the number of vertices in the standard cognitive map (Fig. 6b), the compact cognitive map resembles the standard cognitive map almost identically, as can be seen by naked eyes. Due to the integration of scenes of the same places, the redundant verities are substantially reduced while keeping the correct topology of the environment. The mapping process corresponding to Fig. 6c is shown in video s2 in the supplementary materials.

In addition to scene integration, when the step of sparse vertices through the neighborhood field is adopted, the cognitive mapping process witnesses a further reduction in the number of vertices and edges. Compared with the standard cognitive map, the number of vertices is reduced from 3911 to 497, edges from 5182 to 602 (Fig. 6d), a reduction factor about eight. Here, neighborhood fields are employed to confine the distance between two vertices and restrict the angle of rotation. The same weights for translation $\alpha = 10.0$ and rotation $\beta = 10.0$ are used to describe movement information. The threshold of neighborhood fields is tuned to $\delta = 3.746$ to determine whether a new vertex is added to the cognitive map. Although part of the compact cognitive map in Fig. 6d looks less smooth as the cognitive maps in Fig. 6b, c, it correctly represents all loop closures and intersections of the environment. The compact cognitive map built is therefore consistent with the standard cognitive map.

All in all, as can be seen in Fig. 6, compared with the overhead view of the explored environment in Fig. 6a, except small rotation offset, the compact cognitive maps can successfully represent the overall layout of the explored environment (Fig. 6c, d), and gain big improvement of efficiency than standard cognitive map (Fig. 6b).

Size of cognitive map

Depicted in Fig. 7, the number of vertices and edges in the cognitive map grows as a function of exploration time. For the standard cognitive map, since no vertices and edges are discarded, it has the highest rate of growth (dark lines with star markers $*$ in Fig. 7a, b). The number of vertices and edges increases to 3911 and 5182 in the standard cognitive map, respectively.

Adding the step of scene integration, the number of vertices and edges is reduced to 2191 and 2152 respectively, which is described by a blue line with circular markers (o). The number of vertices and edges increases with a slower slop, as similarly reported in Ball et al. (2013).

Red lines with diamond markers (◇) show the number of vertices and edges of the final compact cognitive map. Controlled by two steps, adding sparse vertices and edges, and scene integration, the size of the cognitive map climbs up in time with the slowest rate among the three methods, only reaching to 497 vertices and 602 edges.

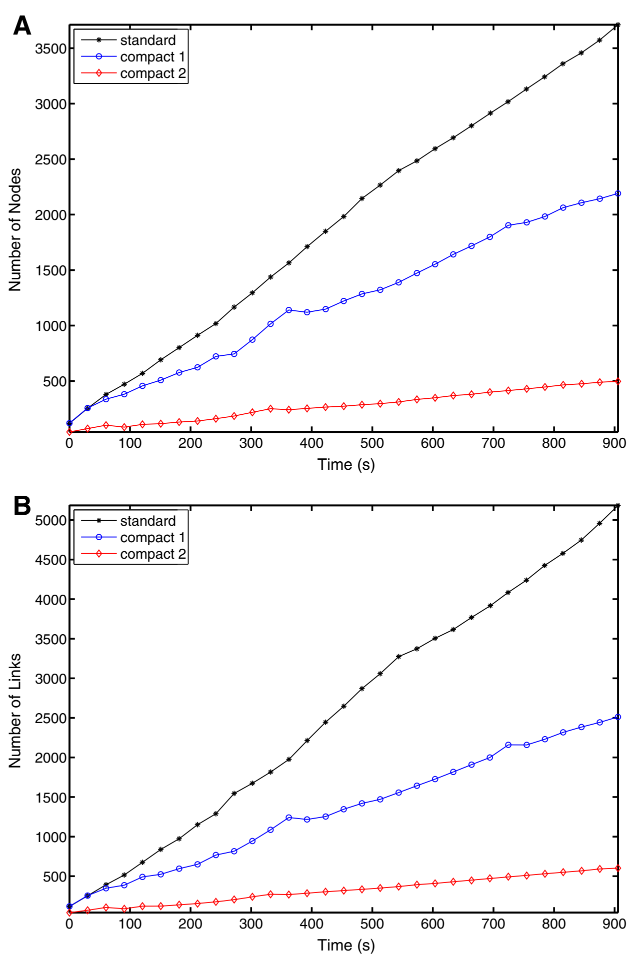

Fig. 7 Size of cognitive map when exploring the environment. Legends standard, compact 1, and compact 2 correspond to cognitive maps in Fig. 6a–c, respectively. **a** The number of vertices. **b** The number of edges

Clustering of loop closure edges

As loop closure edges are created in the experience map, they are incrementally assigned to an existing cluster when they satisfy the requirements shown in Fig. 3, or to a new cluster. Every cluster grows dynamically, which allows to group loop closure edges with a similar trajectory together. In our experiment on iRat Australia dataset, we consider the threshold of total time T_{total} for a cluster to be less than 100 seconds, and the threshold for loop cluster edge time interval $T_{interval}$ to be longer than 2 seconds for a new cluster. Examples of clustering loop closures are shown in Fig. 8. The sizes of the cluster are different from each other, which depends on the trajectory of the robot. For

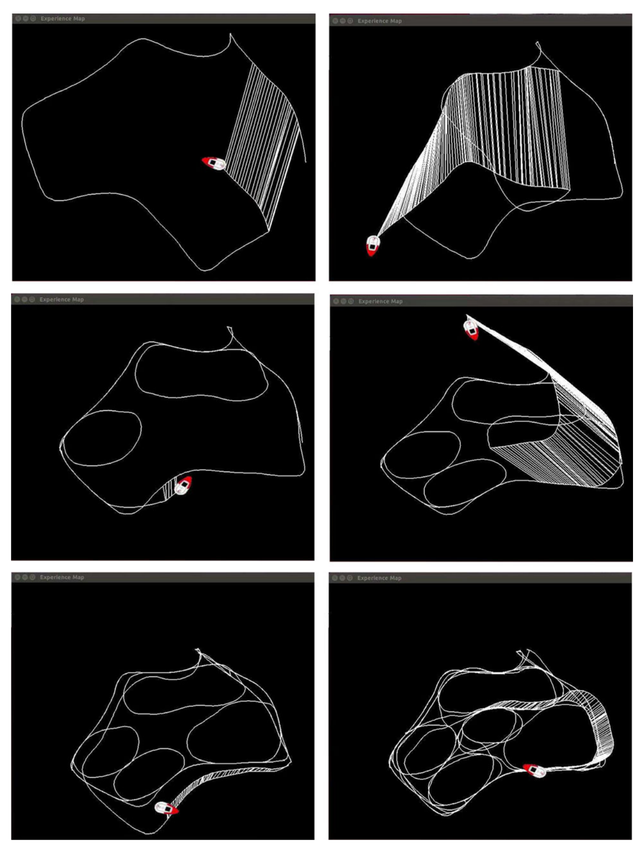

Fig. 8 Clustering of loop closure edges. Six loop closure clusters of variaous cluster sizes are selected throughout the mapping process

each cluster, only once batch global optimization is required to ensure computational efficiency.

Discussion

In this study, we proposed a brain-inspired compact cognitive mapping solution to control the growth of the size of the cognitive map. We implemented our solution in a monocular visual SLAM system, and demonstrated that our cognitive mapping system could successfully build a compact cognitive map and correctly represent the overall layout of the environment (see video s1, s2, and s3 in the supplementary materials).

Our approach allows us to gently trade accuracy for computational efficiency. Neighborhood fields, which are inspired by neighborhood cells, can be employed to adjust the sparsity of the cognitive map, as can be seen in Fig. 6b, c. Although the size of compact cognitive maps can be as small as one-eighth of that of the standard cognitive map (Fig. 7), the compact cognitive maps correctly represent the overall layout of the environment (Fig. 6b, c v.s. a). Also, compared with the result of OpenRatSLAM on the iRat dataset, the size of vertices in our compact cognitive map is smaller than one-fifth of the size of vertices (497 v.s. about 2800) in Ball et al. (2013).

In our approach, we first formulate the global optimization of the cognitive map as a non-linear least-squares problem. Compared with graph relaxation (Ball et al. 2013), a fast sparse solver is applied to solve the non-linear least squares problem with high efficiency using Ceres Solver. Second, clustering loop closure edges are used to group loop closure edges with similar trajectory together. The solution to the global optimization problem is computed only once for each cluster, which greatly improves operational efficiency. Parallel processing with multicores based OpenMP boosts further the computational speed of global optimization. Third, inspired by neurobiological experiments, a method based on neighborhood fields to sparsify sequential vertices and edges, is proposed to perform the compact cognitive mapping. As the neighborhood fields are determined by movement information including translation and rotation, along a straight road, the neighborhood field increases slowly, meaning that less information is needed to remember. Whereas, when making a turn, such as at a crossroad, the neighborhood field increases rapidly, indicating that more information should be retained. Finally, revisiting the same place should not engender multiple spatial representations, but visual scenes from multiple visits should be integrated into the same vertex for robust maintenance of the map. Scene integration would not increase the size of the cognitive map, unless the unknown environment is explored, and is effective to conserve the size of the cognitive map when exploration is not expanded.

Compared with the pose graph compressed method based on information-theoretic measure (Ila et al. 2010; Kretzschmar and Stachniss 2012), our approach is not theoretically derived, but pragmatic and efficient for building cognitive maps. Temporally scalable stereo-vision-based SLAM by Johannsson and Leonard (Johannsson et al. 2013) is also relevant to our approach. Their method avoids adding redundant vertices right from the beginning to reduce the pose graph, and then iSAM is used to perform map state estimation. In our approach, we cluster loop closure edges, after optimization, then revisited vertices are removed. And we formulate a nonlinear least-squares problem to solve the global optimization using a general fast non-linear least-square solver, i.e. Ceres Solver. A brain-inspired monocular visual SLAM decreases the size of the experience map by removing experiences to maintain one experience per grid square density by partitioning the environment (Milford and Wyeth 2010). We achieve the sparsification of the cognitive map by introducing the concept of the neighborhood fields. The movement information with topographical orientation is applied to sparsify sequential vertices and edges in the cognitive map. Besides, we also cluster loop closures edges and perform batch optimization with parallel computing to ensure real-time performance.

Several aspects of our study are worthy of further exploration in order to be deployed on autonomous robots in large-scale environment. First, our algorithm needs to be tested for longer time in larger environments. Second, although the loss function in map optimization reduces the influence of outliers, it can not entirely guarantee the quality of map estimation during the global optimization of the cognitive map. Loop closure constraints should be modelled to reject incorrect loop closures. More supplicated methods for the representation of visual features could be used to support robust place recognition and loop closure.Third, objects are detected by proximity sensors, and simple reflexive rules are used to avoid obstacles in this study at the moment. In the future, active obstacle avoidance algorithm should be integrated into the system, to achieve flexible navigation in more complex environments such as city streets with moving objects like pedestrians and vehicles.

Conclusion

In short, we proposed a brain-inspired compact cognitive mapping system. Inspired by neurobiological experiments, the concept of neighborhood field and scene integration are applied to achieve sparsification of the cognitive map, without adding redundant vertices and edges into the cognitive map. Imitating the way that mammals control the size of the cognitive map, it is possible to develop practical algorithms to store the map during the long-term operation of the robot in complex, large-scale, and dynamic environments.

Acknowledgements The authors would like to thank the support from the National Key Research and Development Program of China (No. 2016YFC0801808).

References

Agarwal S, Mierle K, Others (2012) Ceres solver. http://ceres-solver.org

Ball D, Heath S, Wyeth G, Wiles J (2010) IRat: Intelligent rat animat technology. In: Proceedings of the 2010 Australasian conference on robotics and automation, pp 1–3

Ball D, Heath S, Wiles J, Wyeth G, Corke P, Milford M (2013) OpenRatSLAM: an open source brain-based SLAM system. Autonomous Robots 34(3):149–176

Bos JJ, Vinck M, van Mourik-Donga LA, Jackson JC, Witter MP, Pennartz CM (2017) Perirhinal firing patterns are sustained across large spatial segments of the task environment. Nat Commun 8:15602

Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, Reid I, Leonard JJ (2016) Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. IEEE Trans Robot 32(6):1309–1332

Carlevaris-Bianco N, Eustice RM (2013) Generic factor-based node marginalization and edge sparsification for pose-graph slam. In: 2013 IEEE international conference on robotics and automation (ICRA), IEEE, pp 5748–5755

Carr MF, Jadhav SP, Frank LM (2011) Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. Nat Neurosci 14(2):147–153

Eichenbaum H (2014) Time cells in the hippocampus: a new dimension for mapping memories. Nat Rev Neurosci 15(11):732–744

Hafting T, Fyhn M, Molden S, Moser MB, Moser EI (2005) Microstructure of a spatial map in the entorhinal cortex. Nature 436(7052):801–806

Ila V, Porta JM, Andrade-Cetto J (2010) Information-based compact pose slam. IEEE Trans Robot 26(1):78–93

Johannsson H, Kaess M, Fallon M, Leonard JJ (2013) Temporally scalable visual slam using a reduced pose graph. In: 2013 IEEE international conference on robotics and automation (ICRA), IEEE, pp 54–61

Kretzschmar H, Stachniss C (2012) Information-theoretic compression of pose graphs for laser-based slam. Int J Robot Res 31(11):1219–1230

Kropff E, Carmichael JE, Moser MB, Moser EI (2015) Speed cells in the medial entorhinal cortex. Nature 523(7561):419–424

Kümmerle R, Grisetti G, Strasdat H, Konolige K, Burgard W (2011) g 2 o: A general framework for graph optimization. In: 2011 IEEE international conference on robotics and automation (ICRA), IEEE, pp 3607–3613

Larkin MC, Lykken C, Tye LD, Wickelgren JG, Frank LM (2014) Hippocampal output area ca1 broadcasts a generalized novelty signal during an object-place recognition task. Hippocampus 24(7):773–783

Lever C, Burton S, Jeewajee A, O'Keefe J, Burgess N (2009) Boundary vector cells in the subiculum of the hippocampal formation. J Neurosci 29(31):9771–9777

Lu F, Milios E (1997) Globally consistent range scan alignment for environment mapping. Auton Robot 4(4):333–349

MacDonald CJ, Lepage KQ, Eden UT, Eichenbaum H (2011) Hippocampal "time cells" bridge the gap in memory for discontiguous events. Neuron 71(4):737–749

Mazuran M, Burgard W, Tipaldi GD (2016) Nonlinear factor recovery for long-term slam. Int J Robot Res 35(1–3):50–72

McNaughton BL, Battaglia FP, Jensen O, Moser EI, Moser MB (2006) Path integration and the neural basis of the 'cognitive map'. Nat Rev Neurosci 7(8):663–678

Milford M, Wyeth G (2010) Persistent navigation and mapping using a biologically inspired slam system. Int J Robot Res 29(9):1131–1153

Mittelstaedt ML, Mittelstaedt H (1980) Homing by path integration in a mammal. Naturwissenschaften 67(11):566–567

Moser EI, Kropff E, Moser MB (2008) Place cells, grid cells, and the brain's spatial representation system. Ann Rev Neurosci 31:69–89

Moser MB, Rowland DC, Moser EI (2015) Place cells, grid cells, and memory. Cold Spring Harbor Perspect Biol 7(2):a021808

Naidoo R, Chase MJ, Beytell P, Du Preez P, Landen K, Stuart-Hill G, Taylor R (2016) A newly discovered wildlife migration in Namibia and Botswana is the longest in Africa. Oryx 50(1):138–146

O'Keefe J, Dostrovsky J (1971) The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. Brain research 34(1):171–175

Taube JS, Muller RU, Ranck JB (1990) Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. J Neurosci 10(2):420–435

Tolman EC (1948) Cognitive maps in rats and men. Psychol Rev 55(4):189

Zeng T, Si B (2017) Cognitive mapping based on conjunctive representations of space and movement. Front Neurorobot 11:61

Zeng T, Tang F, Ji D, Si B (2020) Neurobayesslam: neurobiologically inspired bayesian integration of multisensory information for robot navigation. Neural Netw Off J Int Neural Netw Soc 126:21–35

Zhao D, Si B, Tang F (2019) Unsupervised feature learning for visual place recognition in changing environments. In: Proceedings of the 2019 international joint conference on neural networks