Journal of Intelligent & Robotic Systems
https://doi.org/10.1007/s10846-023-01816-3

REGULAR PAPER

A Multisession SLAM Approach for RatSLAM

Matheus Menezes[1] · Mauro Muñoz[1] · Edison Pignaton de Freitas[2] · Sen Cheng[3] · Areolino de Almeida Neto[1] · Paulo Ribeiro[1] · Alexandre Oliveira[1,4]

Received: 4 April 2022 / Accepted: 12 January 2023

Abstract

To successfully perform autonomous navigation, mobile agents must solve the Simultaneous Localization and Mapping (SLAM) problem. However, acquiring the map in a single SLAM session may not be possible, thus the map may be incrementally built over multiple sessions. Two solutions could be considered to solve the multisession SLAM problem: (i) the robot must localize itself in the previously stored map before the new session starts; (ii) it can start a new map and merge it with the map from the previous sessions. To date, only scenario (i) has been addressed by RatSLAM, an algorithm inspired by the navigation system in rodent brains. Therefore, this work proposes a multisession solution that solves both scenarios. A new mechanism merges the data from the RatSLAM structures of the current mapping session with those previously stored if there are connections between these paths. This approach was tested in four different scenarios, from virtual controlled environments to real-world environments with two, three, and five sessions. The robot started in an unfamiliar location for each mapping session, but it also works if the agent starts in a known place, scenario (ii) and (i), respectively. For all experiments, the entire map was consistently obtained. Furthermore, the proposed approach updates and enhances the previous session's map in real-world environments. Therefore, the proposed approach may be a multiple SLAM session solution for the RatSLAM algorithm.

Keywords Mobile robotics · RatSLAM · Multisession SLAM

1 Introduction

Autonomous navigation is one of the fundamental problems in mobile robotics [1], in which autonomous robots travel in unknown static or dynamic environments trying to reach a given objective [2]. While static environments present constant displacement of an object over time, dynamic ones may change their spatial features along with objects in the environment, such as vehicles, people, or other agents, which may move around. In both environments, a relevant and central issue for mobile robot navigation is to acquire a spatial map of the environment while simultaneously localizing the robot on the map, which is known as the Simultaneous Localization and Mapping (SLAM) problem [3, 4].

Traditional solutions in the literature, such as extended Kalman filter SLAM (EKF-SLAM) [3], FastSLAM [5] and Rao-Blackwellized particle filters [6], solve the SLAM problem using an engineering approach.

SLAM algorithms have been applied to several problems in real-world scenarios, such as: search and rescue [7–9], autonomous exploration of hazardous areas [10], area surveillance [11], among others. These are common problems that involve activities that can be hostile or dangerous for human-based interventions.

This study is mainly interested in the SLAM approach inspired by the localization and navigation system in the mammalian brain [12–15], which includes specialized neurons [16]. For instance, place cells in the hippocampus activate when the animal is located in a circumscribed region of space [17]. Grid cells in the entorhinal cortex activate when the animal's location coincides with a vertex of a hexagonal grid overlaid on the environment [18]. Head direction cells in several brain regions activate when the animals' head is rotated in a distinct direction [19].

Computational models of the neural process underlying these spatial representation cells have inspired RatSLAM [20–22]. Visual sensing, robot movement, and pose cells estimate the robot's position and orientation while exploring the environment, similar to a conjunctive grid and head direction cells. The visual representation and odometry input

✉ Alexandre Oliveira
amunizdeoliveira@ucc.ie

Extended author information available on the last page of the article

Published online: 17 July 2023

activate specific neural network regions that implement a three-dimensional Continuous Attractor Network (CAN). The combination of sensing and CAN activity generates an *experience*.

RatSLAM has been successfully used to solve indoor and outdoor SLAM tasks [13, 21–26]. However, RatSLAM requires improvements to deal with multisession SLAM problems, i.e. a robot maps along several sessions. Multisession is required when the robot cannot map the entire environment in one session, e.g., in large environments, or when the robot is shut down and later restarts mapping. It can also be applied when tracking fails caused by occlusion in visual SLAM. In multisession SLAM approaches, the robot can start at a random position in the environment when a new session begins. This new position may not be stored on the previously created map. This problem has been defined as the kidnapped robot [2, 27, 28].

According to [28], two solutions for this problem can be considered: (i) the robot localizes itself on the previously created map before the new session starts, or (ii) the robot starts mapping the new location with its reference coordinates - it may merge this new map with the previous ones to generate a unified map. This work focuses mainly on the latter.

One of the first works in which RatSLAM performs long-term large-scale autonomous mapping was developed by Milford and Wyeth [25]. In that work, the robot worked on delivery tasks for two weeks in two physically separate environments. The agent can be shut down and placed in either environment without notification. When switching from one environment to another for the first time, the agent attempts to locate itself on the previous map, and when unsuccessful, RatSLAM creates a new experience map for this new location. The new map shares the same RatSLAM structure but it is topologically separate from the first built map, i.e., it keeps multiple local maps in the same RatSLAM structure. However, in case of a single connection between them, the odometer error propagation will deform the map due to false metric representation. Therefore, strictly speaking, it does not solve the multisession scenario stated in (ii).

In recent work, a new navigation system has been developed based on the relationship between the hippocampus and episodic memory for mobile agents to perform multistep tasks [13]. The approach adopted RatSLAM as the hippocampus's spatial navigation mechanism and episodic memory to recover the steps of a given task. Moreover, the work proposed an improvement that enables the robot to localize itself globally in previously familiar areas. In their global localization module, the robot can be shut down and restarted in a random position of the environment, i.e., the kidnapped robot problem. Once the robot restarts and the global localization module is activated, RatSLAM uses the robot's visual input to check whether the new location has been saved in the stored map. As long as the robot does not localize itself, RatSLAM does not generate new experiences, as it usually does when fed a new visual input. Instead, it compares the visual information with the ones from the stored map. This solution is similar to scenario (i) mentioned above. Therefore, the approach proposed in [13] only works when the robot restarts inside the previously mapped area. Thus, the robot cannot explore distinct regions in multiple sessions in the global localization mode since no new experiences will be created in this new environment, and so it does not solve the multisession scenario (ii).

Although not fully encompassed by RatSLAM, multisession mapping has been addressed in many non-biological SLAM frameworks [28–36]. In particular, the multisession scenario (ii) is solved by visual/visual-inertial SLAM algorithms such as ORB-SLAM 3 [30] and SLAMM [31]. Both solutions propose starting a new map when the tracking is lost. This new map grows with new inputs, and additional operations continuously compare the current input of the "active" map with those from different stored maps. If new input information belongs to the "active" map, the algorithm performs loop closures; if it belongs to a different map, then both maps will be merged into a single one, which becomes the currently "active" map. This merging operation applies an alignment transformation between the two matched inputs and converts the information from the current map to the stored one.

The previous solutions either solve multisession using an engineering fashion for non-biological SLAM [28–36], or can only partially solve it for a neuro-inspired SLAM as mentioned above [13, 25]. However, to fully deal with multisession RatSLAM, a solution must consider both RatSLAM structures and their intrinsic relation. For example, to correctly reactivate a RatSLAM experience on the map, both visual input and CAN related to that experience must be correctly activated on RatSLAM. This issue presents a challenge in building multisession solutions for RatSLAM, specifically for its network. This paper proposes a new solution to merge the RatSLAM Pose Cell Network, keeping the experience map consistent and performing loop closures correctly after merging of RatSLAM structures, which was not explored in other works found in the literature.

This paper proposes merging multisession maps for a bioinspired SLAM, RatSLAM, improving the flexibility of mapping and allowing robots to build maps incrementally with RatSLAM. This method generates a separate *partial map* of the environment, which is built with the new experiences in a new session. Once an equivalence between a new and a previously stored experience is found in a *loaded map*, a *merge* mechanism combines these maps (partial and loaded maps), adjusting internal structures and yielding a new partial map, ready to continue the mapping in a new session.

If the new session starts at a known location on the loaded map, scenario (i), the agent activates the equivalent experience on the loaded map, and the mapping proceeds from this point. Similarly to ORB-SLAM 3 and SLAMM, this work establishes the matching of RatSLAM's Local Visual Cells as bases to find an alignment transformation among RatSLAM structures. In addition to the solution for (i), the proposed approach also solves (ii) once it merges the RatSLAM structures and ensures both the correct metric representation and the inner relations after the merge operation, which was not possible in other works found in the literature.

This paper is organized as follows: Section 2 presents the theoretical basis of the RatSLAM algorithm. In Section 3, the methodology of this work is depicted. Next, Section 4 presents the experimental setup, that is, the scenarios, robots, parameters, and other experimental details. In Section 5, the results are presented and discussed. Finally, the conclusions of this work and suggestions for future work are presented in Section 6.

2 RatSLAM foundations

RatSLAM was developed in 2004 by Milford, Wyeth, and Prasser [20] for general real-world examples of localization and mapping on mobile robots using a vision system as its main input sensor. Figure 1 shows the updated RatSLAM architecture [21]. It is composed of the following modules:

1. The **Robot Vision System** acquires the images and sends them to the other modules.
2. The module **Self Motion Cues** estimates the translational and angular velocities of robot odometry. This information is sent to the Pose Cell Network and Experience Map. Moreover, odometry information can be computed using images from the Robot Vision System using visual odometry (Fig. 1).
3. The **Pose Cells Network** is a three-dimensional Continuous Attractor Network (CAN) of units connected by excitatory and inhibitory connections. Each cell of the Pose Cell Network represents the robot's position (x, y) and orientation θ.
4. The **Local View Cells** maintain a list of scenes seen that are called templates. A new template is created when the scene sent by the Robot Vision System is new. A link between it and the activity in the Pose Cells Network is learned when a new template is created.
5. The **Experience Map** is a graphical map representing the environment built and maintained by the information from the Pose Cells Network, Local View cells, and Self Motion Cues modules.

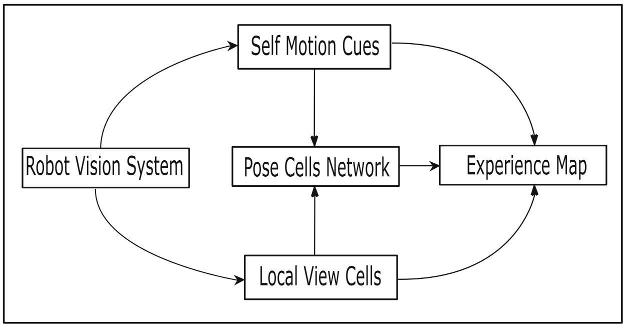

Fig. 1 **RatSLAM architecture.** The Local View Cells are activated by the Robot Vision System. The Local View Cells create and activate templates depending on whether the scene seen by the robot is new or not. The Self Motion Cues send the odometry information of the robot to the Pose Cells Network. Since the Robot Vision system can capture the odometer through visual odometry, an arrow is placed between these modules. In the Pose Cells Network, each cell represents the pose of the robot. Excitatory and inhibitory connections link these cells. The Experience Map represents the information learned about the environment

The following subsections describe the details of the Pose Cells Network, Local View Cells, Experience Map, and the role of the RatSLAM parameters in its function.

2.1 Pose Cells Network

The Pose Cells Network (PCN), denoted by P, is a CAN arranged in a three-dimensional structure (Fig. 2) and represents the position (x', y') and orientation θ of the robot. Excitatory and inhibitory connections link the CAN units. In addition, the connections among units wrap across all six faces of the PCN (e.g., red arrows in Fig. 2). With these wrapped connections, the network can work beyond its fixed size, i.e., it can represent environments with an area larger than the areas encoded by the PCN.

Furthermore, local excitatory and global inhibitory connectivity among cells provides inputs to these units and changes their activity. Over time, this dynamics allows the CAN to form a cluster of activated cells, known as *energy packet* or *activity packet* [21] (Fig. 2, blue cubes). In addition, the center of the energy packet (Fig. 2, darker blue cubes) is the best estimate of the robot's pose in the environment.

Local excitatory connections increase the activity of neighboring cells, whereas global inhibitory connections eliminate the activity of small clusters in the PCN. The excitatory and inhibitory connectivity are described by the distribution ε [37]:

$$\varepsilon_{a,b,c} = e^{-(a^2+b^2)/k_p^{exc}} e^{-c^2/k_d^{exc}} - e^{-(a^2+b^2)/k_p^{inh}} e^{-c^2/k_d^{inh}} \quad (1)$$

where k_p and k_d are the variance constants for place and direction, respectively [21]. The parameters a, b, and c represent the distances between two cells' coordinates, considering the network's periodic boundary conditions. The

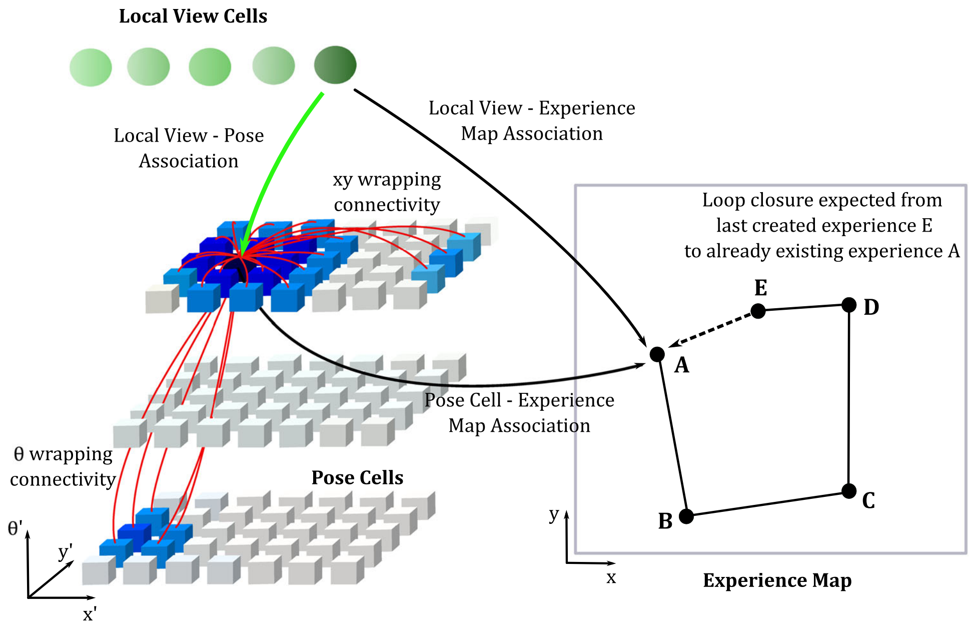
Local View Cells
Local View - Pose Association
Local View - Experience Map Association
xy wrapping connectivity
Loop closure expected from last created experience E to already existing experience A
E
D
A
Pose Cell - Experience Map Association
θ wrapping connectivity
Pose Cells
θ'
y'
x'
y
x
B
C
Experience Map

Fig. 2 **Associations among Local View Cells (LVCs), Pose Cells Network (PCN), and Experience Map (EM).** The LVCs inject energy into PCN, shown as cubes. The activated cells of the PCN are displayed as blue cubes, and the most activated cell of this packet (the darkest blue) is the best estimate of the robot's pose in the environment. The nodes on the EM are called "experiences" in RatSLAM. They compose a metric-topological map, where the edges represent the topological relation between experiences

distance between two cells with coordinates x', y', and θ and i, j, k , respectively, is given by:

$$a = (x' - i)(\mathrm{mod}\ n_{x'}),$$
$$b = (y' - j)(\mathrm{mod}\ n_{y'}),$$
$$c = (\theta' - k)(\mathrm{mod}\ n_{\theta'}), \qquad (2)$$

where *mod* is the modulo operator. The parameters $n_{x'}$, $n_{y'}$ and $n_{\theta'}$ indicate the network size in terms of the number of cells along each of the X', Y' and Θ' dimensions. The change of activity in a cell is given by [37]:

$$\Delta P_{x',y',\theta'} = \sum_{i=0}^{n_{x'}-1} \sum_{j=0}^{n_{y'}-1} \sum_{k=0}^{n_{\theta'}-1} P_{i,j,k}\varepsilon_{a,b,c} - \varphi \qquad (3)$$

where φ is the global inhibition. The final step in the network update limits the activation levels in P to non-negative values and normalizes the total activation to one [37].

In addition, the direction in which the energy package changes is driven by odometry information, which represents the robot's movement, and energy injection by the templates stored in LVC, which might move the activity packet to a different location in the PCN [21].

2.2 Local View Cells - LVC

The Local View Cells, V, form an array of *templates*, denoted by V_i. Each template represents a distinct visual scene of the environment captured by the Robot Vision System. When a template is created, a short learning excitatory link β is established between it and the center of the dominant activity packet in the PCN [21]. This β link associates the distinct view from that location with the robot's estimated pose encoded by the PCN. The link is given by [37]:

$$\beta^{t+1}_{i,x',y',\theta'} = \max(\beta^{t}_{i,x',y',\theta'}, \lambda V_i P_{x',y',\theta'}) \qquad (4)$$

where λ is the learning rate, and the variable i denotes which template V_i is activated. The x', y', and θ' are the coordinates of the center of the dominant energy package in the PCN. Note that $\beta^{t+1}_{i,x',y',\theta'}$ only creates a new link with value $\lambda V_i P_{x',y',\theta'}$ if a previous association was not already learnt, i.e. $\beta^{t}_{i,x',y',\theta'}$ does not exist, yet.

Moreover, when a loop closure occurs, i.e., the robot returns to a location it has already mapped, a consecutive sequence of previously stored templates are activated in the correct order. When the template V_i is activated, it injects energy in the PCN at (x', y', θ') coordinates via its learnt link (4) [21]:

$$\Delta P_{x',y',\theta'} = \delta \sum_{i} \beta_{i,x',y',\theta'} V_i \qquad (5)$$

where δ is the constant, determining the influence of visual features on the estimated robot's pose [21]. When the PCN receives a constant injection of activity, this changes the dominant energy package and, consequently, the re-localization of the robot.

2.3 Experience Map - EM

The Experience Map (EM) is a graph combining pose cells and templates information to estimate robot poses in a two-dimensional map. A node in the EM is defined as a 3-tuple [21]:

$$e_i = \{P^i, V^i, \mathbf{p}^i\} \tag{6}$$

where P^i and V^i are the activity states in the PCN and LVC, respectively, at the time the experience node is created, and $\mathbf{p}^i$ is the robot pose in the experience map space. Furthermore, a new experience node is created when the states P^i and V^i do not closely match the state of any existing experience.

A link $l_{i,j}$ is created and saved when the robot moves from a previous experience e_i to the new experience e_j [21]:

$$l_{i,j} = \{\Delta\mathbf{p}^{i,j}, \Delta t^{i,j}\} \tag{7}$$

where $\Delta\mathbf{p}^{i,j}$ is the relative odometry pose between the two experience nodes, and $\Delta t^{i,j}$ is the time taken by the robot to move between the nodes. The temporal information can be used to perform the path planning from a specific experience to the desired goal. As exemplified in [21], Dijkstra's algorithm can be used to find the shortest path between two nodes.

As long as there is no loop closure, the EM is based on the robot's odometry. Loop closure activates the robot's relocalization in the map and distributes the odometry error throughout the graph using a relaxation algorithm, changing experience poses. Changes in experience locations are obtained as follows [21]:

$$\Delta\mathbf{p}^i = \alpha\left(\sum_{j=1}^{N_f}(\mathbf{p}^j - \mathbf{p}^i - \Delta\mathbf{p}^{i,j}) + \sum_{k=1}^{N_t}(\mathbf{p}^k - \mathbf{p}^i - \Delta\mathbf{p}^{k,i})\right) \tag{8}$$

where α is a correction rate constant set to 0.5, N_f is the number of links from the experience node e_i to other experiences, and N_t is the number of links from the other experiences to the experience e_i.

2.4 Role of RatSLAM Parameters

The RatSLAM has parameters responsible for the algorithm's appropriate mapping performance. The parameters of *Local View* are responsible for template operations, such as comparison, size dimension, and when new templates need to be created. The *Pose Cells* parameters influence the dynamics and dimensions of the network, e.g., values for local excitation, global inhibition, and energy injection when a scene is revisited. Finally, *Experience Map* parameters define the number of executions of the graph relaxation algorithm (8).

In addition, The Self Motion Cues module (Fig. 1) provides odometry information computed from the Robot Vision System. In this case, Visual Odometry parameters are set to correctly determine the robot's translational and rotational velocities. However, the Visual Odometry parameters can be ignored if this information comes from a separate source, such as wheel encoders.

3 Methodology

At the end of each session of a multisession mapping, the robot generates a map of the environment using RatSLAM and stores it. In the next mapping session(s), this map is loaded by the robot (*loaded map*). The robot is assumed to receive no information about its position on the loaded map. In the new session, the robot builds a new map called *partial map*. If a common path that links the partial and the loaded maps is found, they are merged into a single RatSLAM structure, i.e., the merging results in a single LVC, PCN, and EM to represent the environment. This process is described in detail below.

It is assumed that the partial map has more than one experience before the merge procedure, i.e., the agent starts the new session in a nonmapped area of the environment. If the robot starts at a known location of the loaded map, the algorithm could activate the equivalent experience on the loaded map, and the mapping proceeds from this experience.

The merge process is triggered if the agent creates a *template* in the partial map that matches a previously stored template in the loaded map (Fig. 3), i.e., the agent entered a location that is represented by both maps. To compare templates, the same process that compares the templates in RatSLAM is employed, i.e., a new template of the partial map is compared to all templates of the loaded map. When a match occurs, both the *Loaded Map* and the *Partial Map* consist of RatSLAM's structures as shown in Fig. 2, where V, P, E, lm and pm stand for Local View Cell, Pose Cell Network, Experience Map, loaded map and partial map, respectively. The new template of the partial map matches a template from the loaded map $V^{\mathrm{pm}}_{n_{vp}} = V^{\mathrm{lm}}_u$ (Fig. 3), where n_{vp} is the number of templates of V^{pm}. In addition, the views $V^{\mathrm{pm}}_{n_{vp}}$ and V^{lm}_u are linked to the center of the activity packet, displayed as the green cubes, P^{pm}_z and P^{lm}_u, respectively. However, even though the templates represent the same place, their activity packets may activate different coordinates (x', y', θ') in the respective PCN, P^{pm} and P^{lm}. Similarly, for the Experience Map, the templates and activity packets are allocated to experiences e^{pm}_z and e^{lm}_u (Fig. 3, green nodes), where these experiences may have different poses coordinates.

A relevant information for merging also relies on the prior position before encountering the matching view in the partial

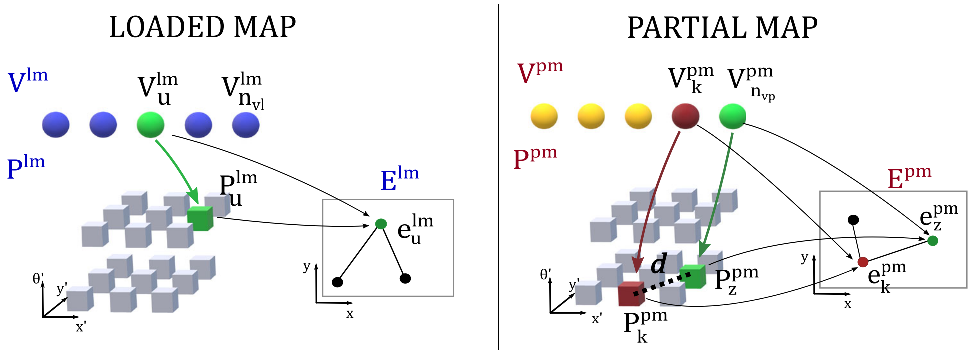

Fig. 3 **Condition for the merge procedure between two RatSLAM structures.** The left structure shows Loaded Map, i.e., the RatSLAM structure that has been previously created and saved. The right structure is the Partial Map, which is the current RatSLAM structure being used in the mapping. The relation among LVC, PCN, and EM is displayed in each structure. The condition to merge both structures is to find a similar local view cell in both Loaded and Partial Maps, shown in the green spheres $V_u^{lm} = V_{n_{vp}}^{pm}$. Note that these view cells are linked with activity packets P_u^{lm} and P_z^{pm} and experiences e_u^{lm} and e_z^{pm} in their respective structures. Furthermore, in the Partial Map, the relation between a previously learned local view, V_k^{pm} (red), its associated activity packet P_k^{pm} and experience e_z^{pm} is displayed. The activity packets P_z^{pm} and P_k^{pm} are in a distance **d** and there is a topological link between the experiences e_k^{pm} and e_z^{pm}

map, which corresponds to the location in the previous time step $k = n_{vp} - 1$. Note that the linked centers of the activity packets P_z^{pm} and P_k^{pm} are spatially separated by a distance d. Likewise, their associated experiences e_z^{pm} and e_k^{pm} have link information (7) that encodes the distance information between their poses.

At merging, the structures of the partial RatSLAM map and their relations are inserted in the loaded map in four operations (Fig. 4): i) the LVC are merged (Fig. 4a), ii) the matching view in the partial map $V_{n_{vp}}^{pm}$ is linked to a new PCN activity packet location, which corresponds to a shift (Fig. 4b), iii) the associations between all LVC and PCN are shifted by the same amount (Fig. 4c), and iv) the EM are merged (Fig. 4d). In the following subsections, these operations are described in more detail. Finally, at the end of the merge procedure, a single RatSLAM structure is obtained (Fig. 4e) and used in the remainder of the current mapping session.

3.1 Merging Local View Cells

The merge of LVC aims to join all the templates from the loaded and the partial map into a single LVC structure. The templates of the partial map are concatenated into the LVC of the loaded map, except for the last acquired template because it is already present in the loaded structure:

$$V^{\mathrm{lm}} = [V^{\mathrm{lm},1}, ..., V^{\mathrm{lm},n_{vl}}, \mathbf{V}^{\mathrm{pm},\mathbf{1}}, ..., \mathbf{V}^{\mathrm{pm},\mathbf{n_{vp}-1}}] \tag{9}$$

where n_{vl} is the number of templates of V^{lm}.

In the last stage of the LVC merge, the template that represents the actual robot's view scene V_u^{lm} is activated in the V^{lm} (Fig. 4, green sphere).

3.2 Pose Cells Network Activation

After merging the LVC, the activated template is V_u^{lm}, which needs to be associated with the units in the PCN at right. This injection of activity should occur at the coordinates of P_u^{lm} because it is associated with V_u^{lm}. Before the merge procedure, the last activated packet in the partial map was P_z^{pm} (Fig. 4B, light green cube). Thus, a change of activity from P_z^{pm} to P_u^{lm} is required and can be seen as a shift of activity in the P_u^{lm} (Fig. 4B, green arrow). The difference of coordinates between P_z^{pm} and P_u^{lm} is defined as:

$$\begin{aligned} \Delta x' &= x'_p + x'_l; \\ \Delta y' &= y'_p + y'_l; \\ \Delta \theta' &= \theta'_p + \theta'_l. \end{aligned} \tag{10}$$

where (x'_p, y'_p, θ'_p) and (x'_l, y'_l, θ'_l) are the coordinates of activity packets P_z^{pm} and P_u^{lm}, respectively.

3.3 Shifting Association between LVC and PCN

All associations between LVC and PCN in the partial map must be updated to be consistent with the associations in the loaded map. As an example, we consider the penultimate view in the partial map. P_k^{pm} must be shifted to keep the previous distance d to P_u^{lm} (Fig. 4C, red arrow). The

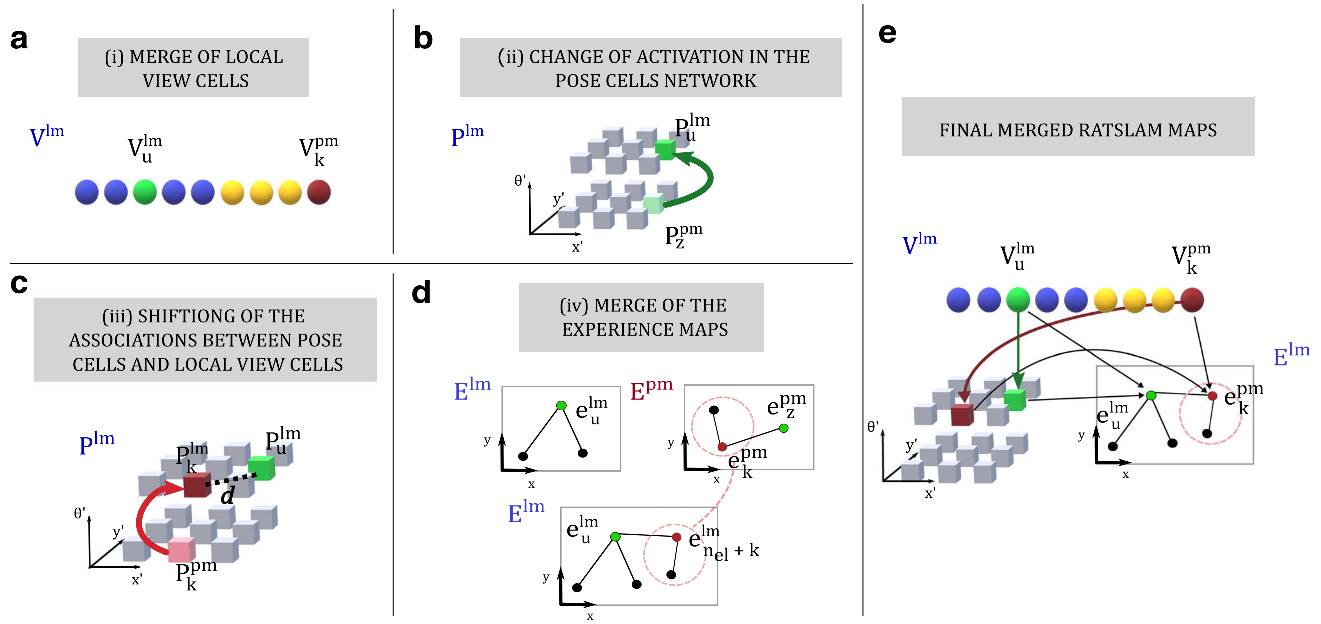

Fig. 4 **Merging partial and loaded RatSLAM maps.** The merge procedure is divided into four operations: (i) merge LVC; (ii) change activation in the PCN; (iii) shift the associations between LVC and PCN; and (iv) merge the EM. These are illustrated in the first four panels. **a**) All templates of the V^{pm} (Fig. 3) are inserted in the LVC structure V^{lm}, except the last template of the partial map (green circle of V^{pm} in Fig. 3), because this template is equivalent to V_u^{lm}, which becomes the current active template at the end of step (i). **b**) The activity packet is associated with the active template at the end of (i) P_u^{lm} (Loaded Map in Fig. 3). In addition, before the merge procedure, the activity packet encoding the robot's pose in the partial map was P_z^{pm}. However, after the merge, the similar energy packet that encodes the robot's pose in the PCN is P_u^{lm}. This change of activation can be seen as a shift of coordinates in the Pose Cell P^{lm}, and it is represented by the green arrow from P_z^{pm} to P_u^{lm}. **c**) The coordinates of the association between LVC and PCN of the partial map are changed when they are inserted into the loaded map. As illustrated, the energy packet of the partial map, P_k^{pm}, is shifted (red arrow) to keep the same distance **d** (dashed line) to the active energy packet P_u^{lm} as it had with P_z^{pm} in the partial map (Fig. 3). Note that the shift applied to P_k^{pm} is the same computed in (ii). **d**) The experiences from the partial map's ME are inserted into the EM of the loaded map, as shown by the dotted red circles. A function that transforms the coordinates of the e_z^{pm} (green node) in the E^{lm} to the coordinates of it equivalent experience e_u^{lm} (green node) in E^{lm} is computed, and the shift is applied to all experiences of E^{pm}, except e_z^{pm}. A final link between e_k^{pm} and e_u^{lm} is introduced to connect the inserted experiences from the E^{pm} to E^{lm}. **e**) The final RatSLAM structure after the merge

transformation function that shifts the activity packet $P_k^{\rm pm}$ to $P_k^{\rm lm}$ is defined as $f()$:

$$
\begin{aligned}
&f(x'_u, y'_u, \theta'_u) = (x'_s, y'_s, \theta'_s);\\
&x'_s = (x'_u + \Delta x')(\mathrm{mod}\ n_{x'});\\
&y'_s = (y'_u + \Delta y')(\mathrm{mod}\ n_{y'});\\
&\theta'_s = (\theta'_u + \Delta\theta')(\mathrm{mod}\ n_{\theta'}).
\end{aligned}
\tag{11}
$$

where (x'_u, y'_u, θ'_u) are the coordinates of $P_k^{\rm pm}$. The (x'_s, y'_s, θ'_s) are the shifted coordinates of $P_k^{\rm lm}$ in the loaded map.

Once this shifted operation is carried out, the excitatory links $\beta^{\rm lm}$ must be updated as follows:

$$
\beta^{\rm lm}_{n_{vl}+i, f(x',y',\theta')} = \beta^{\rm pm}_{i,x',y',\theta'}, i = \{1, ..., n_{vp} - 1\}. \tag{12}
$$

where (x', y', θ') are the coordinates of the energy packets from $P^{\rm pm}$.

3.4 Merging Experience Maps

Before the experiences of $E^{\rm pm}$ can be merged into $E^{\rm lm}$, the experiences in the partial map have to be map consistent with those in the loaded map. Before the merge, the experience $e_z^{\rm pm}$ represented the robot's actual pose and it is linked with $e_k^{\rm pm}$ in $E^{\rm pm}$, as shown in Fig. 3. $e_z^{\rm pm}$ is equivalent to $e_u^{\rm lm}$ in the the loaded map. As previously mentioned, $e_z^{\rm pm}$ and $e_k^{\rm pm}$ encodes the last and penultimate robot's poses in the partial map, respectively. The topological-metric relation between $e_k^{\rm pm}$ and $e_z^{\rm pm}$ must be kept between $e_k^{\rm pm}$ and $e_u^{\rm lm}$ after the merge process by applying the same transformation on $e_z^{\rm pm}$ and $e_k^{\rm pm}$. $e_k^{\rm pm}$ after the transformation is denoted as $e_{n_{el}+k}^{\rm lm}$ (see notation in (16)) and it is shown in the merged EM $E^{\rm lm}$ in Fig. 4D where the dashed red line indicates the transformation.

This transformation is defined as the combination of translational and rotational operation in two dimensions. Let $\mathbf{p}_l = [x_l, y_l, \theta_l]^T$ and $\mathbf{p}_p = [x_p, y_p, \theta_p]^T$ be the poses of the equivalent experiences e_u^{lm} and e_z^{pm}, respectively. The function $\mathbf{t}$ transforms the pose $\mathbf{p}_p$ to the pose $\mathbf{p}_l$, i.e. $\mathbf{t}(x_p, y_p, \theta_p) = x_l, y_l, \theta_l$, and is defined as follows:

$$\mathbf{t}(x_p, y_p, \theta_p) = [H(x_p, y_p), \theta_p + \Delta\theta]^T. \tag{13}$$

where $\Delta\theta = \theta_l - \theta_p$ and $H(x_p, y_p)$ is defined by:

$$H(x_p, y_p) = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} x_p - \Delta x \\ y_p - \Delta y \end{bmatrix}. \tag{14}$$

where $\Delta x = x_l - x_p$ and $\Delta y = y_l - y_p$. Once function $\mathbf{t}$ is defined, it must be applied on the poses of E^{pm} experiences so they could be inserted on the merged E^{lm}, which is carried out through operation T, defined below. In addition, operation T also changes the information of the new experiences in E^{lm} with their correspondent template and energy packets according to operations **(i)** and **(iii)** loaded map's structures V^{lm} and P^{lm}.

$$\begin{aligned} e^{\mathrm{lm}}_{n_{el}+i} &= T(e_i^{\mathrm{pm}}), i = 1, ..., n_{ep} - 1; \\ &= \{V^{\mathrm{lm}, i+n_{vl}}, P^{\mathrm{lm}, i+n_{vl}}_{f(x', y', \theta')}, \mathbf{t}(\mathbf{p}^{\mathrm{pm}, i})\}. \end{aligned} \tag{15}$$

where e^{pm} is an experience from E^{lm}. The n_{el} and n_{ep} are the number of experiences in E^{lm} and E^{pm} maps, respectively. It is important to mention that e_z^{pm} is not added in E^{lm} because it is already equivalent to e_u^{lm}. Then, the new experiences are inserted on the merged E^{lm}.

$$E^{\mathrm{lm}} = [e_1^{\mathrm{lm}}, ..., e_{n_{el}}^{\mathrm{lm}}, \mathbf{e}^{\mathrm{pm}}_{\mathbf{n_{el}+1}}, ..., \mathbf{e}^{\mathrm{pm}}_{\mathbf{n_{el}+n_{ep}-1}}] \tag{16}$$

The last step **(iv)** is to connect, through links, the added nodes on the E^{lm} as follows:

$$l_{i+n_{el}, j+n_{el}} = \{\Delta\mathbf{p}^{(i+n_{el})(j+n_{el})}, \Delta t^{i,j}\} \tag{17}$$

$i = 1, ..., n_{ep} - 2$ and $j = 2, ..., n_{ep} - 2$, which is similar to (7). As e_z^{pm} is not inserted its equivalent, e_u^{lm}, has to be connected to $e_k^{\mathrm{pm}*}$ through (17) with $i + n_{el}$ being k and $j + n_{el}$ is u.

3.5 Merge Algorithm

The merge algorithm is summarized in the Algorithm 1, highlighting the equations that make up each step. Complexity analysis can consider the match and the merge routines separately. Considering that the merge process is triggered from a matching between a template in the partial map of size n_{vp} and a template of size n_{vl}, stored in some of the previous sessions κ, the computational complexity can be given by:

- Matching routine: depending on the search implementation, it could be $\kappa \times O(1)$, $\kappa \times O(log(n_{vl}))$ or $\kappa \times O(n_{vl})$;
- MergingLocalViewCells(line 2): $O(n_{vp})$;
- ComputeDeltasPCN(line 3): $O(1)$;
- P^{lm} updating (line 4): $O(1)$;
- ShifitingAssociationLvcPcn(line 5): $O(n_{vp})$;
- ComputeTransformation (line 6): $O(1)$;
- MergeExperiencesMaps (line 7): $O(n_{ep})$.

Algorithm 1 Merge of RatSLAM structures.

```
1: procedure MERGE($\mathrm{V^{lm}, V^{pm}, P^{lm}, P^{pm}, E^{lm}, E^{pm}}$)
2:     $V^{\mathrm{lm}} \leftarrow$ MergingLocalViewCells($V^{\mathrm{lm}}, V^{\mathrm{pm}}$)    ▷ Eq. 9
3:     $\Delta x', \Delta y', \Delta\theta' \leftarrow$ ComputeDeltasPCN($P_u^{\mathrm{lm}}, P_z^{\mathrm{pm}}$)    ▷ Eq. 10
4:     $P^{\mathrm{lm}}_{current} = P_u^{\mathrm{lm}}$    ▷ PCN activation
5:     $P^{\mathrm{lm}} \leftarrow$ ShifitingAssociationLvcPcn($\beta^{\mathrm{pm}}_{x',y',\theta'}, f(x', y', \theta')$)    ▷ Eq. 12
6:     $\mathbf{t}() \leftarrow$ ComputeTransformation($e_u^{lm}, e_k^{pm}$)    ▷ Eq. 13
7:     $E^{\mathrm{lm}} \leftarrow$ MergeExperiencesMaps($E^{\mathrm{pm}}, \mathbf{t}()$)    ▷ Eq. 15,16,17
8: end procedure
```

In summary, the complexity of the merge algorithm does not exceed the linear behavior concerning the **Local View Cells** (in both partial and loaded maps) and **Experience Map** sizes.

4 Experimental Setup

Four different environments are explored in this study. For each environment, datasets of video streams or image frames have been captured during tours performed by real and virtual robots. Briefly, the four environment datasets are: i) videos generated by a virtual robot from an ellipse-shaped tour, named *Virtual Tour* dataset; ii) videos generated by a real robotic platform during a tour inside a research lab called *Lab Tour* dataset; iii) frames extracted from the *"iRat"* Australian dataset; and iv) frames extracted from the *The New College Vision and Laser dataset*. The latter two were used to validate the OpenRatSLAM implementation [21]. Each environment is detailed in the following subsections.

To evaluate the proposed multisession approach, single-session and multisession maps are compared using Iterative Nearest Points (ICP) [38]. The ICP solves the registration problem by finding a transformation matrix that approximates the two maps as closely as possible in iterated executions. As a criterion for the evaluation of the transformation matrix, the ICP computes the *root mean square errors* (RMSE) to the corresponding node distances in both maps. The ICP stops the iterated executions when either the RMSE is lower than a defined threshold or the algorithm executes the maximum number of iterations. Therefore, by returning the RMSE over the distances of the correspondent nodes, the ICP provides a single value that evaluates the overall trajectory of the multisession and single-session maps, with RMSE equal to 0 being a perfect match between them. This article uses the Libicp algorithm [39].

The RatSLAM algorithm requires a specific set of parameter values for each environment [21, 24]. These parameters are required in all the main structures, i.e. LVC, PCN, and EM. The parameters values for each environment are displayed in Table 1. In both the *"iRat"* and New College datasets, odometry information is obtained from the robot's wheel encoders, so the Visual Odometry parameters were not used.

The multisession RatSLAM source code was implemented using Python 3.6 and is available at https://zenodo.org/badge/latestdoi/568248424.

4.1 *Virtual Tour* Experiment

In the *Virtual Tour* experiment, the robot takes an ellipse-shaped tour in a virtual environment (Fig. 5). Both the environment and the robot were modeled in a simulation framework developed to study biomimetic models of rodent behavior in spatial navigation learning tasks [40].

Table 1 Parameter of RatSLAM's to each tested environment

name	value			
	Virtual Tour	Lab Tour	iRat	New College
# Visual Odometry				
vtrans_image_x_min	0	80	–	–
vtrans_image_x_max	256	560	–	–
vtrans_image_y_min	0	240	–	–
vtrans_image_y_max	64	360	–	–
vtrans_scaling	10.0	10	–	–
vtrans_max	0.1	0.1	–	–
vrot_image_x_min	0	80	–	–
vrot_image_x_max	256	560	–	–
vrot_image_y_min	0	240	–	–
vrot_image_y_max	64	360	–	–
camera_fov_deg	360	50	–	–
camera_hz	1	1	–	–
# Local View module				
vt_panoramic	0	0	0	1
vt_match_threshold	0.05	0.04	0.035	0.059
vt_shift_match	5	4	4	4
vt_step_match	2	1	1	10
vt_normalisation	0.5	0.5	0.5	0.5
vt_active_decay	1.5	1.0	1.0	1.5
template_x_size	128	80	50	60
template_y_size	32	60	30	10
# Pose Cell module				
pc_vt_restore	0.05	0.05	0.05	0.05
pc_dim_xy	80	80	11	80
pc_vt_inject_energy, δ	0.06	0.5	0.6	0.06
pc_dim_th	35	36	36	35
exp_delta_pc_threshold	4.0	1	2.0	4.0
pc_cell_x_size	1	1	0.015	1.0
# Experience Map module				
exp_loops	250	250	20	20
exp_initial_em_deg	140	90	140	140

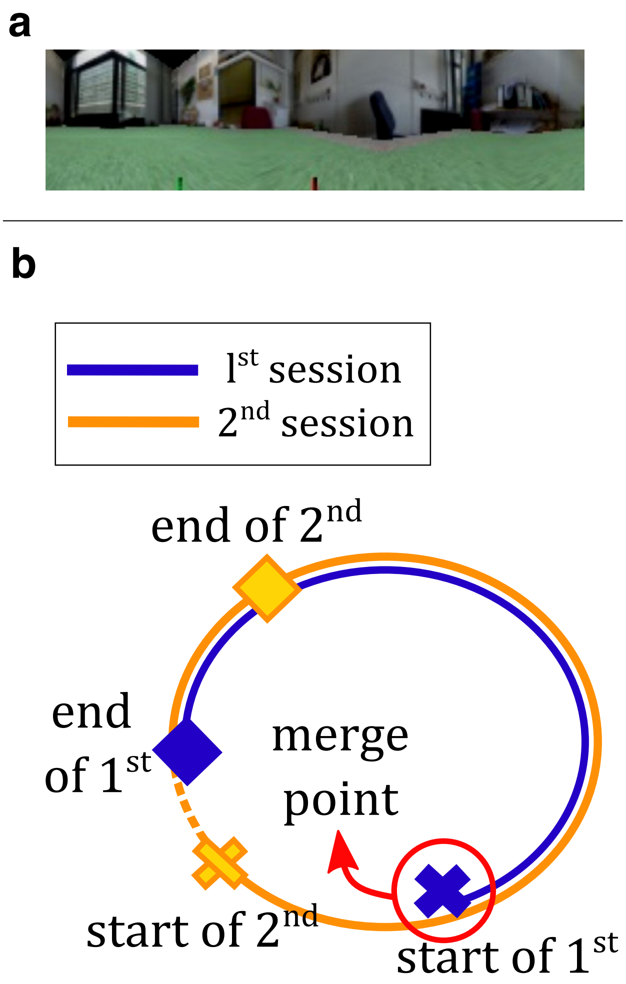

Fig. 5 **Virtual Tour environment setup.** The experiment is divided into two mapping sessions. **a**) Sample frame captured by the robot. **b**) Ellipse path travelled by the agent. In the first session, the agent maps the path of the 3/4 turn represented by the blue line, starting and ending at the blue cross and blue diamond, respectively-the map generated by the robot in the first session loaded in the second session. The robot travels almost three laps (yellow lines) in the video stream used for the second session. The agent starts and ends mapping at the yellow cross and diamond symbols, respectively. In addition, the *partial* map of the second session is the path that starts at the yellow cross and ends before the start of the blue cross. The merge between the partial map of the second session and the loaded map occurs when the robot reaches the start point of the first session (blue cross) in the second session (red circle). The merge procedure results in a single RatSLAM structure that holds both the loaded and partial map. Furthermore, after the merge, there is a non-mapped section of the path (dashed yellow line). The robot maps this area when it moves from the blue diamond to the yellow cross

This experiment is divided into two mapping sessions. First, the robot performs a 3/4 lap through the environment (Fig. 5**b**, blue line). When the robot reaches the end of the first session path (blue diamond), the experience map is saved. In the second session, the virtual agent loads this map. Since it is positioned at a novel location (Fig. 5**b**, yellow cross), the agent starts mapping with a new experience map (partial map). The robot performs almost three complete laps (yellow line) and eventually travels on the same path as it did in the first session.

The merge between the partial map in the second session and the loaded map of the first session occurs when the agent first encounters a view that was stored in the loaded map (Fig. 5b, red circle). After the merge, the virtual robot continues the mapping, using the merged RatSLAM structure, until the end of the second session.

Note that the video frames that generated the map in the first session are part of the video frames used in the second session. Specifically, the video of the first session is embedded in the video used in the second session. Thus, when the agent passes the merge point in the second session, the following frames are the same ones collected in the first session. Therefore, no new experiences should be created by RatSLAM when the agent moves through the path that was covered by the agent in the first session (blue path).

Moreover, after merging, there is a section of path between the end of the first and the start of the second sessions (dashed yellow line between the blue diamond and yellow cross). Such a session has not been mapped in either session. The agent i n to close the loop of the ellipse path fully.

4.2 *Lab Tour* Experiment

In the *Lab Tour* experiment, a research laboratory (Fig. 6**a**) is mapped by a real robot platform, called RoboDeck[1] (Fig. 6**b**). The robot platform is equipped with a monocular camera with 640×480 resolution. Since only the robot's camera was used to create the video streams used in this experiment, the odometry information was extracted by visual odometry. The RoboDeck obtained the video streams of this experiment in a manually driven tour of the room. The robot trajectory is a rectangular path, resembling a figure eight (Fig. 6**c**).

In the first session, the robot performs two counterclockwise laps along the small rectangle (Fig. 6**c**, blue line). The robot starts and ends at the blue cross and diamond locations, respectively. In the second session, the robot performs almost two counterclockwise laps around the large rectangle (Fig. 6**c**, yellow line), starting and ending at the locations marked by the yellow cross and diamond, respectively. The merge between the maps is expected where the two paths first meet (Fig. 6**c**, red circle). After the merge, a final loop closure is expected when the robot returns to the start point of the second session (yellow cross).

4.3 *iRat* Experiment

The *iRat 2011 Australia dataset* was used to validate the multisession approach with more than two mapping sessions,

[1] http://www.xbot.com.br/

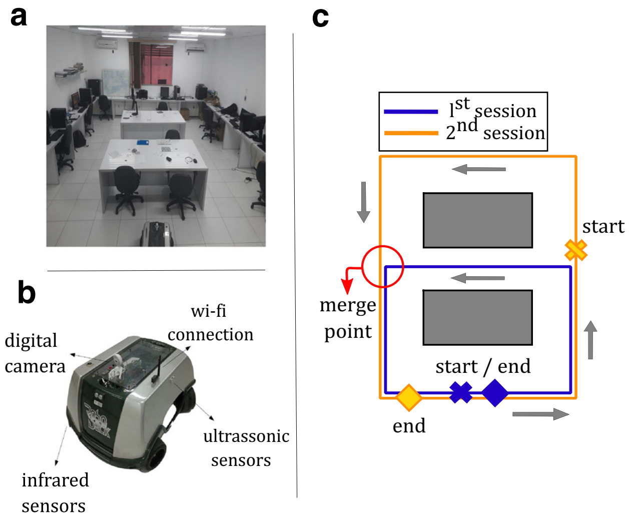

Fig. 6 **Lab Tour experiment setup. a**) Physical laboratory environment. **b**) The robotic platform RoboDeck. **c**) The paths travelled by the robot in the two mapping sessions. In the first session, two laps are performed, starting and ending at the blue cross and blue diamond, respectively. The gray arrows show the direction of movement. In the second session, the robots start at the yellow cross, perform almost two complete laps, and end at the yellow diamond. The red circle marks the merge point for the partial and loaded map. Additionally, a final loop closure is performed when the robot reaches the start point of the second session at the start of the second turn

focusing on consistency in more than two sessions. The dataset was obtained while a small mobile robot called *iRat*, similar in size and shape to a large rodent [21], explored an outdoor road tour (Fig. 7**a**). As main sensors, *iRat* was equipped with an overhead camera and dead reckoning sensors to provide images and odometry data.

The complete dataset composes a video of approximately 16 minutes long [21], during which the robot explores the environment moving without any pattern between the roads. However, only part of the dataset frames were used in the experiment, corresponding to five mapping sessions. The first four sessions are internal laps of the environment. The last session corresponds to the external lap (Fig. 7**b**) and therefore merges the four internal maps into a single one when it travels through their common areas.

As mentioned, the map from one mapping session is transferred to the next as the loaded map. Once the partial map session overlaps with the loaded map, they can be merged. This process is then repeated in the next mapping session. Figure 7**c** illustrates the merge points between the fifth and first sessions (red circles), and the fifth and second sessions, respectively.

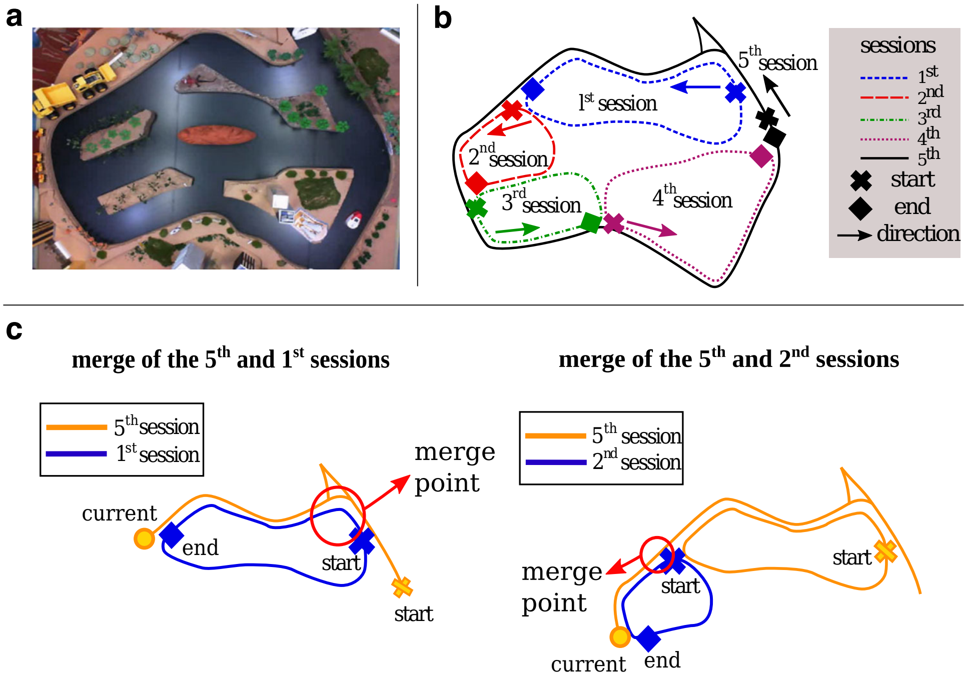

Fig. 7 **The *iRat* experiment setup. a**) Top view of the *iRat* environment. This experiment is performed in five mapping sessions. **b**) The experiment is composed of 5 mapping sessions. In each session, the robot performs a complete lap over internal and external paths, which should ensure a loop closure by the RatSLAM algorithm. The arrows give the direction of movement. **c**) The left panel depicts the merge of the maps from the fifth (partial map, yellow) and the first (loaded map, blue) mapping sessions. Once it encounters a previously mapped location (red circle), the two maps are merged and become the updated partial map. The agent continues mapping until finding a correspondent match in the second session (right panel). This process continues until the end of the fifth session

4.4 New College Experiment

The New College Vision and Laser is a full-scale long-term dataset collected from a robot completing several loops outdoors around the New College campus in Oxford [41]. The data includes 360° image (Fig. 8**a**), odometry, and laser scan information. Similarly to the *iRat* dataset, the New College has been used to validate the openRatSLAM implementation [21]. Due to its complexity, this experiment aims to validate the multisession approach with full-scale long-term data.

The experiment is divided into three mapping sessions. The first two sessions covered the entire environment in two different areas (Fig. 8**b**, blue and green paths). The third session aims to merge the two loaded maps into a single one (yellow path). In addition, in this session, considerable amounts of new visual information are added to the mapping. These new inputs will test the approach's capability to maintain consistent mapping for a long period after the merge operations.

In the first session, the robot completes three laps inside the ellipse shape in the clockwise direction (upper blue arrow). Subsequently, the agent leaves towards the intermediate area between the first and second sessions and returns to the ellipse area, covering its fourth lap, but in a counterclockwise direction (bottom blue arrow). In the second session, the agent travels through the different regions of the environment in a total of two clockwise turns.

The third mapping starts at a known position within the map of the first session (Fig. 8, yellow cross). The robot travels through the entire environment in counterclockwise laps. In particular, the robot's last counterclockwise lap within the area corresponding to the second session adds novel information for the mapping.

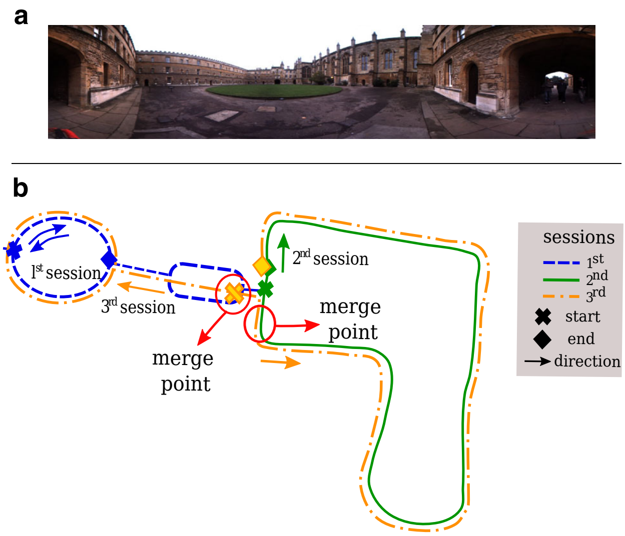

Fig. 8 **The New College Oxford experiment setup. a**) A sample frame of the dataset. **b**) The experiment is composed of 3 mapping sessions. In the first two sessions, the robot completes laps over distinct areas. The third (partial map) session merges both (loaded) maps on the merge points (red circle) and adds substantial new input for the RatSLAM. The arrows give the direction of movement

5 Results

In this section, the results of the *Virtual Tour*, *Lab Tour*, *iRat*, and New College experiments are presented. To compare the multisession solution with the standard RatSLAM process, the EMs for both multiple mapping and single mapping sessions are shown, respectively. Note that after the merge procedure, the resulting RatSLAM structure (LVC, PCN, and EM) is similar to the standard RatSLAM one.

5.1 *Virtual Tour* Results

5.1.1 *Virtual Tour* Experience Map

For comparison, the EM for the single RatSLAM mapping is displayed in Fig. 9**a**. The link between the start and endpoint of the EM shows that the loop is closed in the map. The result of the multisession mapping is presented in different stages to show the evolution of the EM, starting at the end of the first session (Fig. 9**b**). Figure 9**c** depicts the second session with the partial map (yellow), alongside the loaded map (blue), at the moment when the agent found an experience on its EM that matched with an experience in the loaded map, thus triggering the merge process. After the merge procedure, the experiences of the partial map were transformed (translation and rotation) and joined into the loaded map (Fig. 9**d**). Once the agent completed a full lap in the second session, the EMs of the first and second sessions formed a closed loop (Fig. 9**e**). As expected, the number of EM nodes in the single and multiple sessions is the same, meaning that no new experiences were created after merging the maps. Therefore, the merge operations made the creation of new experiences unnecessary. Neither new templates nor new activity packets on PCN were added to the EM.

Figure 9**f** displays the final merged EM of both the first and second mapping sessions. As displayed, a path correction is performed by RatSLAM over the EM. This correction on the merged EM shows that after operation iv), the nodes are linked in such a manner that they influence each other when the relaxation algorithm distributes the odometry error throughout the graph. Therefore, this path correction demonstrates that the RatSLAM works as expected after the merge procedure.

Finally, the ICP comparison of both single and multisession maps shows strong similarities between the paths after the transformation of the merged map to fit into the single-session map (Fig. 10). RMSE = 0.00773327.

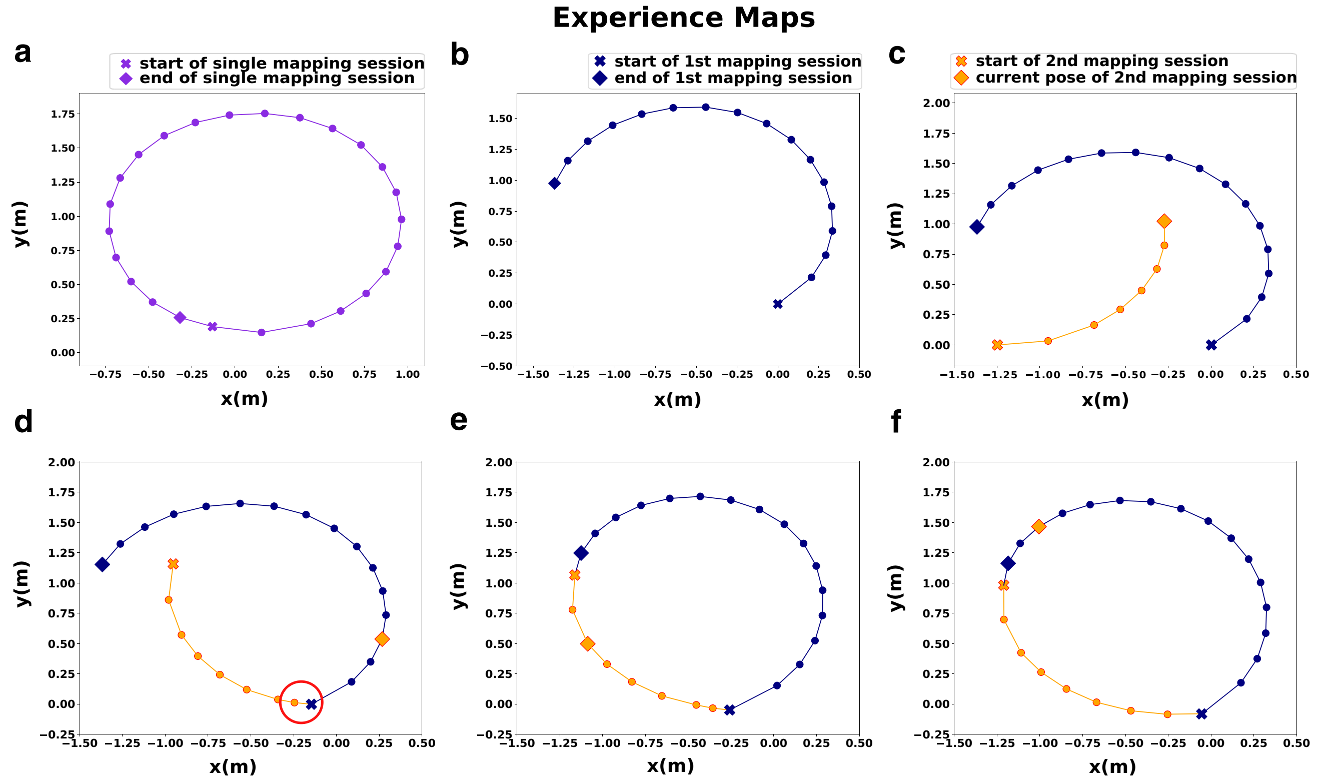

Fig. 9 **Experience Maps (EM) of the Virtual Tour experiment. a**) The path is generated in a single mapping session after the agent performs two turns and a half. **b**) EM of the first mapping session. **c**) EM in the second session when the agent finds a corresponding experience in the loaded map, and the merge condition is met. **d**) EM after merging the loaded and partial maps. The red circle represents the merge point. **e**) EM after the loop closure triggered when the agent completed a full lap. Note that new experiences are not generated when the agent travels in a previous path (blue dots and lines), i.e., no yellow experiences are plotted over the blue map. **f**) Path correction performed by RatSLAM after the loop closure in **e**

5.1.2 *Virtual Tour* Pose Cells Network

In this part, the behavior of the PCN is analyzed during the merge process. The Virtual Tour PCN was chosen because it has a simplified behavior due to the path mapped by the agent, i.e., the PCN is fully activated in the Θ axis (robot orientation) because of the ellipse-shaped laps. This behavior can be seen after the end of a single mapping session (Fig. 11**a**). It is worth mentioning that the second and third laps done by the virtual agent have the same frames and speed information as the first one. Hence, the PCN units from the first lap are reactivated in these next ones. Therefore, we define *start* and *end* for the first and last energy packets created in the single mapping session.

The PCN for the multisession is partially stored in the loaded and partial map structures. As the loaded map has not completed a full lap, its PCN is partially filled on the Θ axis (Fig. 11**b**). Likewise, the partial map also has only part of its PCN activated. Nevertheless, its last activation (blue) corresponds to the *start LM* activation of the loaded map (Fig. 11**c**). Through operation (iii) in the merge, the activities

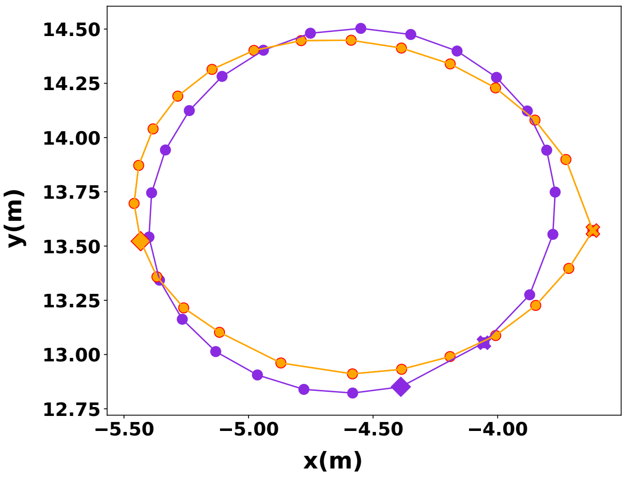

Fig. 10 **ICP comparison for *Virtual Tour* experiment.** The final merged experience map (EM) from the multiple session (orange) closely matched that from the single mapping session (purple). RMSE = 0.00773327

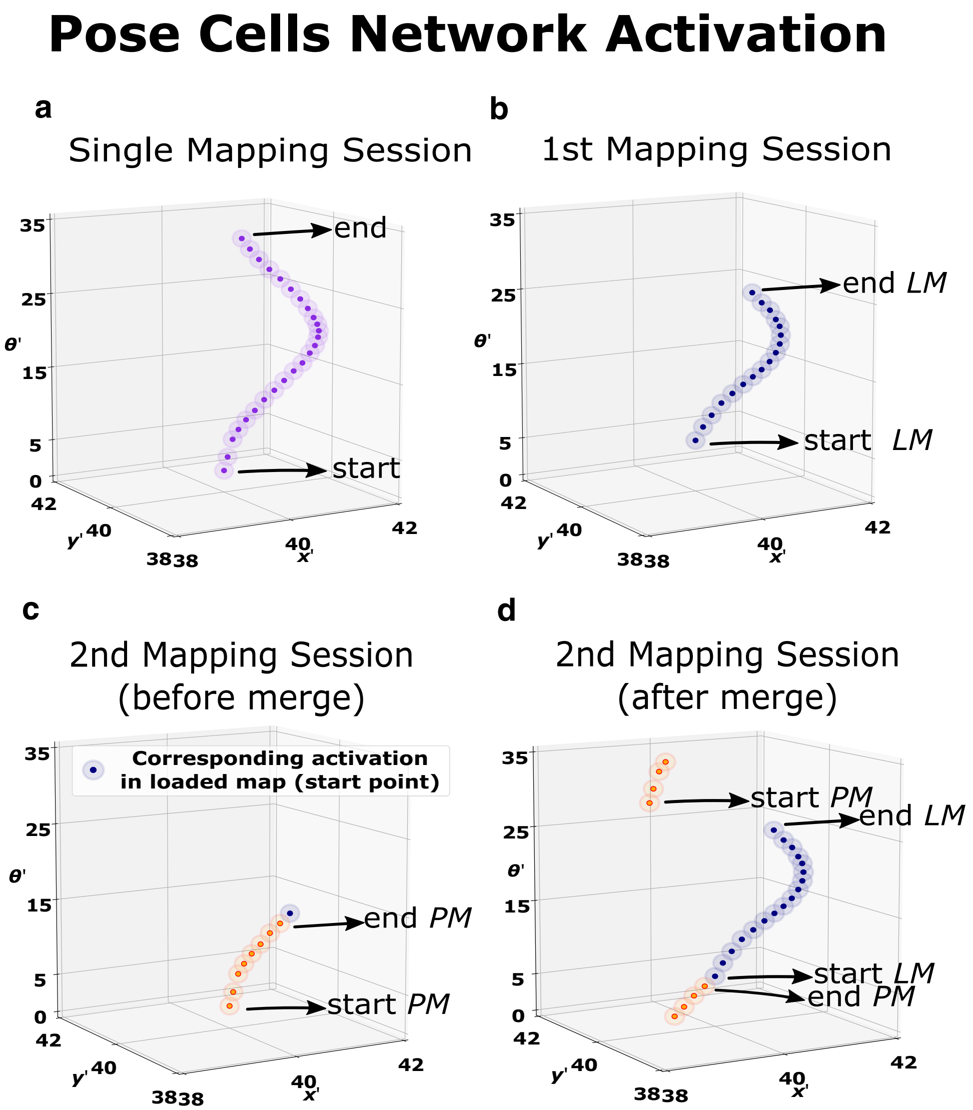

Fig. 11 **Pose Cells Network (PCN) activation in the Virtual Tour experiment. a**) PCN activation for the single mapping session. The Θ' axis activity represents the robot orientation change over the ellipse-shaped lap. Since the path is a closed loop, the Θ' axis is completely activated. The first and last energy packets created during the map are represented by the *start* and *end* dots. **b**) PCN activity of the loaded map in the first mapping session. **c**) PCN activity of the partial map in the second session before the merge procedure. The last activity packet created in **c** (blue) is equivalent to *start LM* in **b**. **d**) PCN activation after the merge procedure. Once the blue activity in **c** must match the *start LM* through operation ii), the energy packets of the partial map PCN are shifted (operation iii). Part of partial map activity packets is moved to the top of the network once they reach the network boundary on the bottom face. Like the single mapping session, the multisession PCN is activated completely in the Θ' axis, corresponding to its closed-loop EM

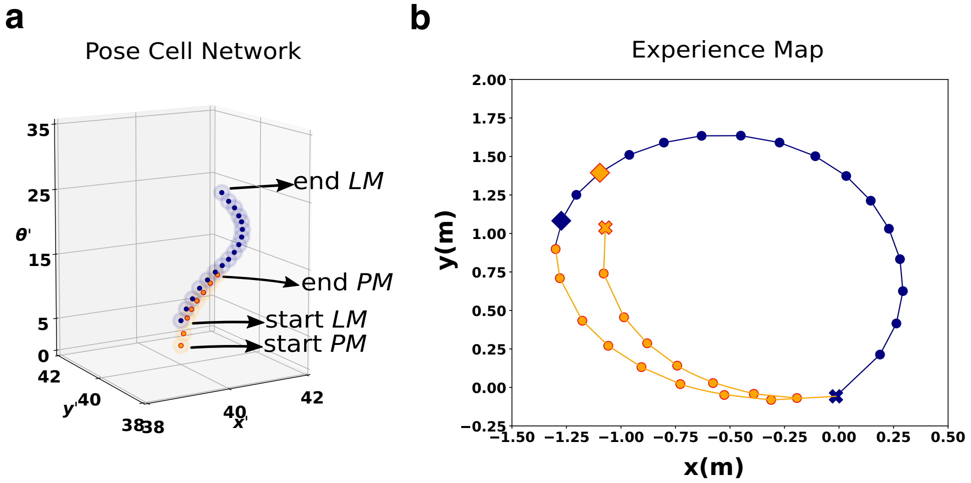

Fig. 12 **No shifting association between PCN and LVC.** Without the correct operations over the PCN, experiences on the Experience Map are incorrectly created instead of being reactivated

of the partial map are shifted to match the blue activity packet with the *start LM* in the loaded map. Note that part of the partial map PCN activity packets is shifted to the network's top face after reaching the bottom face's boundary.

Similar to the single SLAM session, the final merged map of the *Virtual Tour* is a closed-loop ellipse. Consistently, the final PCN is activated through the complete Θ axis after the merge (Fig. 11**c**). These results show that our method changes both EM and the PCN coherently, and their final results resemble the ones from a single mapping session.

Finally, to demonstrate the influence and impact of the PCN operation on the merge process, multisession mapping was carried out without shifting the PCN activities of the partial map (operation iii). As expected, new experiences were created due to the incorrect association between templates and PCN activity (Fig. 12). Therefore, as demonstrated, the merge operations on the PCN are necessary for RatSLAM to perform correctly.

5.2 *Lab Tour* Results

Compared to the *Virtual Tour*, the *Lab Tour* experiment is more complex in two regards. It was performed in a physical environment using a robot in a more complex environment's topology. Due to the robustness of RatSLAM, the EM for the single mapping session nevertheless correctly represents the environment's topology (Fig. 13**a**). So, does the multisession mapping, which we discuss step-by-step? After the first session, the EM only reflects one of the loops in the environment (Fig. 13**b**) since the agent only had experiences in the smaller loop. Up until the merge, conditions are met (Fig. 13**c**), the loaded and partial maps are not aligned, and their relative positions are random. This changes when the two maps are merged (Fig. 13**d**). Note that the nodes of the partial map had been concatenated at the merge point (red circle).

Figure 13**e** displays a path correction of the start and end nodes (blue cross and diamond symbols) of the first session

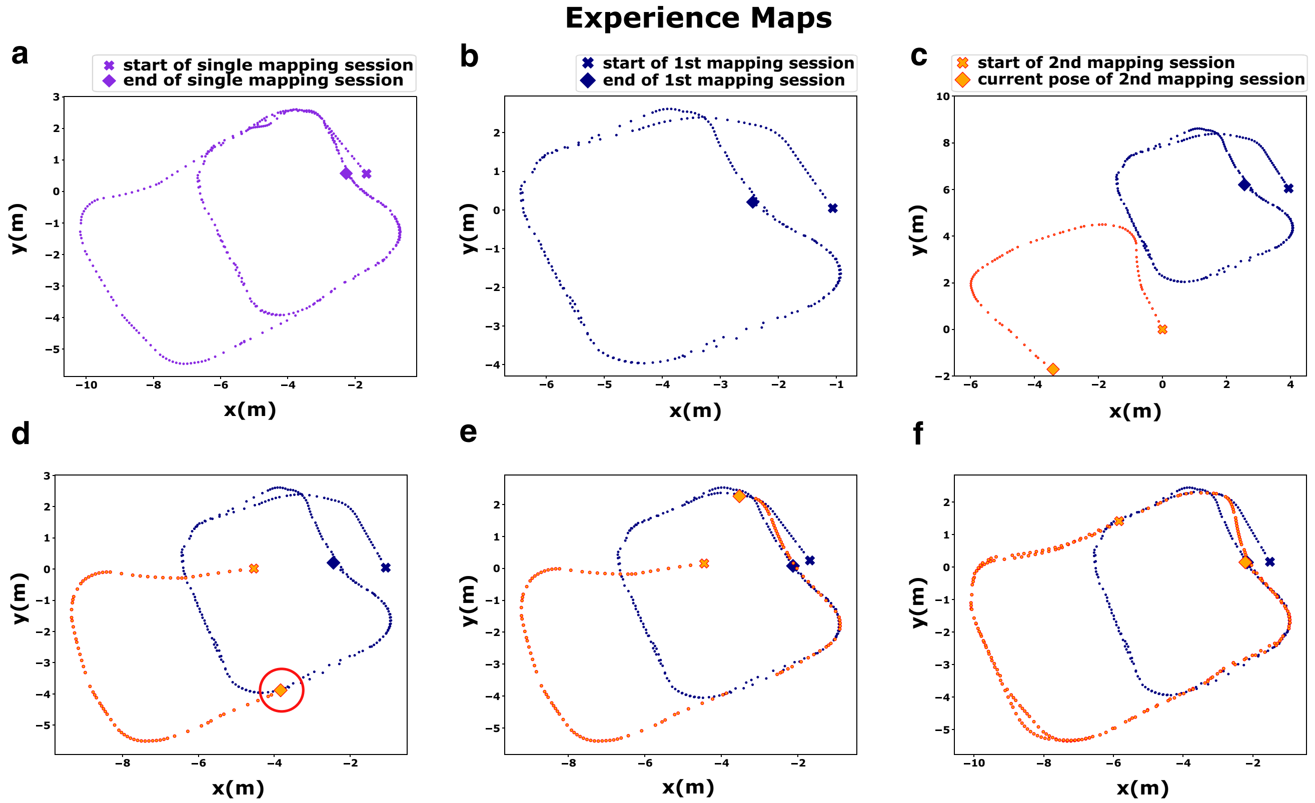

Fig. 13 **Experience maps (EM) of the Lab Tour experiment. a**) EM of the single mapping session. The cross and diamond, respectively, represent the start and end of this map. **b**) Partial map (blue) produced in the first session (see Fig. 6**b**). **c**) The loaded map (blue) and partial map (yellow) were in the second session when the conditions for the merge procedure were met. **d**) The two EMs maps merged in a single RatSLAM structure. **e**) Path correction in the second session after the merge procedure, i.e., the loaded map start and endpoint distance have changed in the second session. **f**) Final map produced at the end of the second session. Additionally, there was a loop closure after the robot completed the second session's larger lap

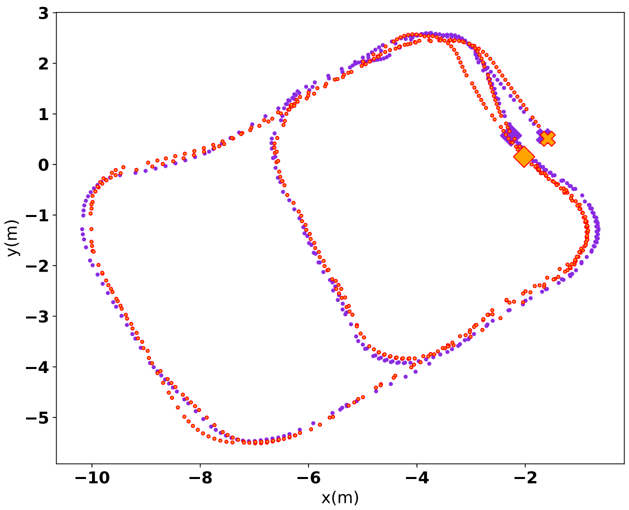

Fig. 14 **ICP comparison for *Lab Tour* experiment.** The final merged experience map (EM) from the multiple session (orange) closely matched that from the single mapping session (purple). RMSE = 0.00883269

map. Path correction over an already mapped place reinforces the expected execution of RatSLAM after the merge procedure since both maps are linked in one single structure. Thus, all the nodes should be affected if changes are needed in the EM after the merge. Besides that, this result shows that the merge approach could improve the mapping in multisession.

The final EM at the end of the second session exhibits the results of the second loop closure when the robot completes a full lap in the second session path Fig. 13**f**. This final map is similar to the EM generated in the single session (Fig. 14). RMSE = 0.00883269.

5.3 *iRat* Results

The *iRat* experiment is challenging since it requires merging maps in five sessions. Nevertheless, single-session RatSLAM performs very well, generating an accurate EM (Fig. 15**a**). Note that the robot continuously moves through the environment in a single mapping session. This continuous movement influences the behavior of RatSLAM's relaxation algorithm (8) since the path correction depends on the connection among the nodes.

The results of the multisession mapping show that the fifth session seamlessly connected the loaded EMs of the previous sessions. (Fig. 15**b-f**). Unlike single-session mapping, multisession maps are not generated through continuous robot

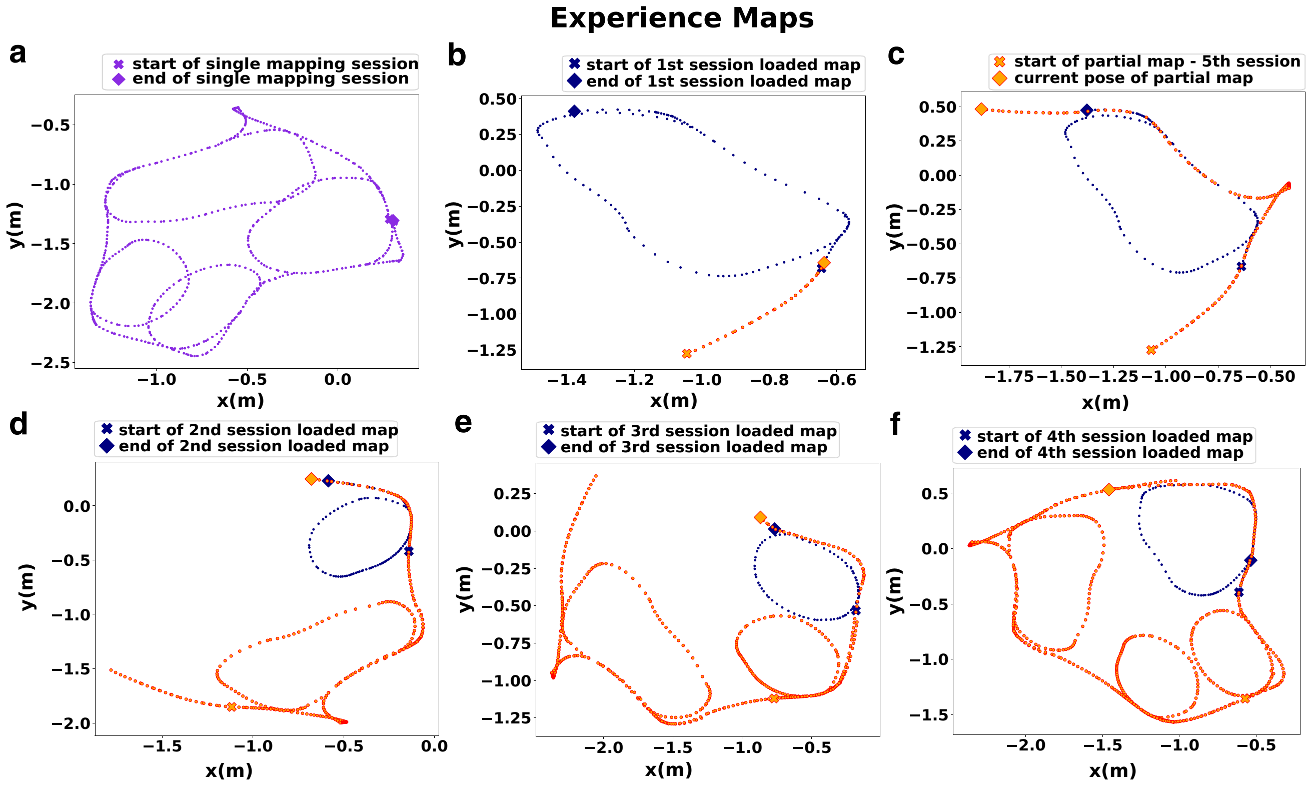

Fig. 15 **Experience maps (EM) of the *iRat* experiment. a**) EM of the single mapping session. The start and end points of the map are depicted as the cross and diamond symbols, respectively. **b**–**f**) The Experience Map produced in the fifth mapping session (see Fig. 7**b**). After merging the partial map with a loaded map (blue), the resulting map becomes the partial map (yellow)

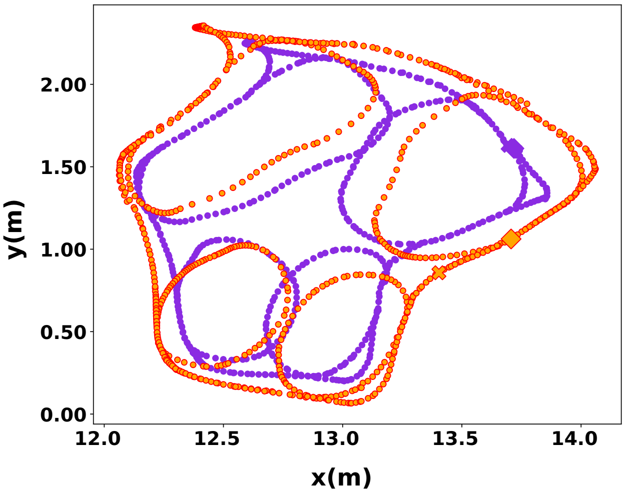

Fig. 16 **ICP comparison of EM for *iRat* experiment.** Despite a noted scale difference, the final merged experience map (EM) from the multiple session (orange) closely matched that from the single mapping session (purple), RMSE = 0.0126932

movements, that is, each session generates its map independently. In addition, the mapping of the fifth session covers only the external area of the environment. Therefore, the existing links between the loaded and partial maps are their regions corresponding to this external path. Without linking the internal nodes with the external ones, no path corrections can be made on the final map. This may have influenced the high distances shown in the internal maps compared to single-session EM (Fig. 16). Nevertheless, the ICP comparison between the final multisession EM with the single-session EM shows clear visual similarities between the internal paths and the overall map. (Fig. 16). RMSE = 0.0126932.

5.4 New College Results

The New College experiment is the biggest challenge since it involves mapping an even more complex environment, and the multisession version requires the merging of large maps. The single-session EM closely matches the physical structure of the environment (Fig. 17**a**). Similarly, the first and second sessions show similar maps for the different areas (Fig. 17**b-c**).

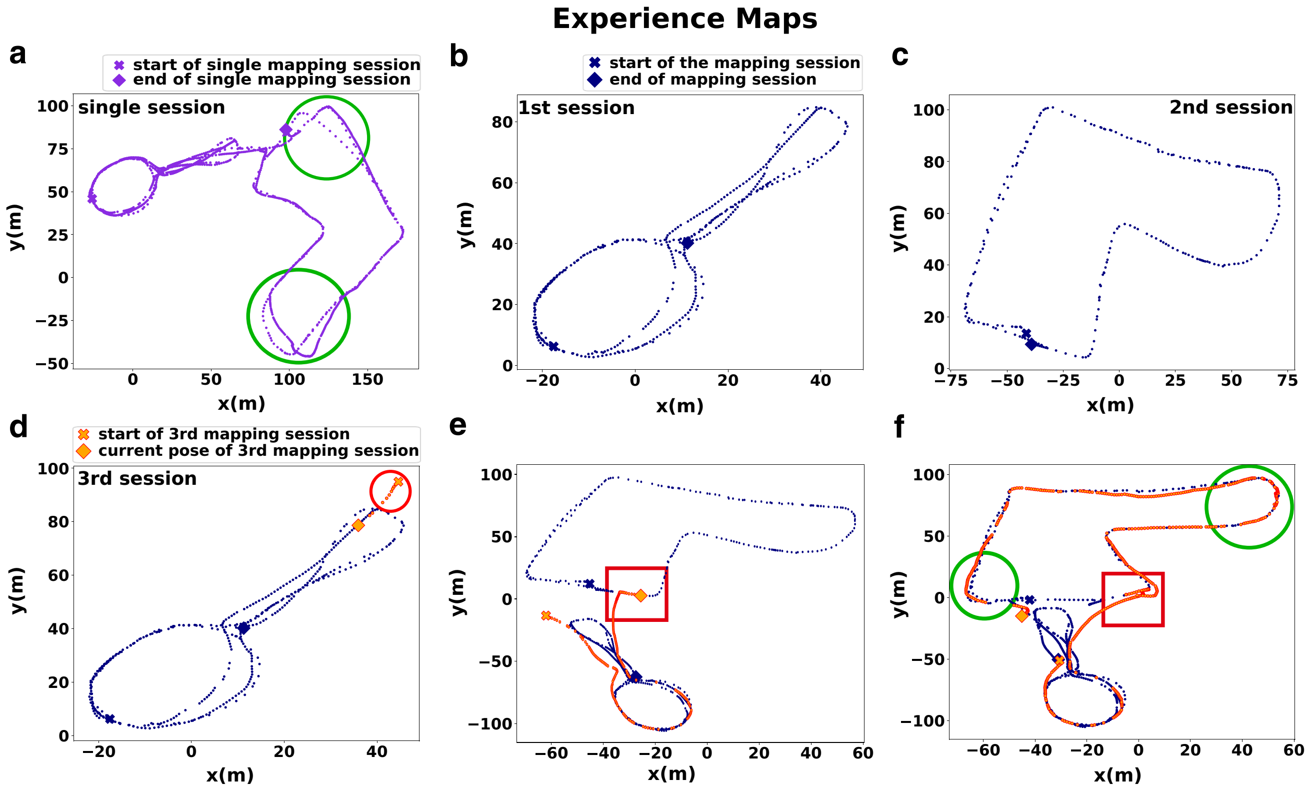

Fig. 17 **Experience maps (EM) of the *Oxford New College* experiment. a**) EM of the single mapping session. The start and end points of the map are depicted as the cross and diamond symbols, respectively. **b**–**c**) The Experience Map produced in the first–second mapping sessions. **d**– **f**) The Experience Map produced in the third mapping session. The red circle shows the partial map merged into the first session loaded map. Red squares depict path correction after the merge within the second session EM. Green circles show loop closures comparisons between multisession and single-session EMs

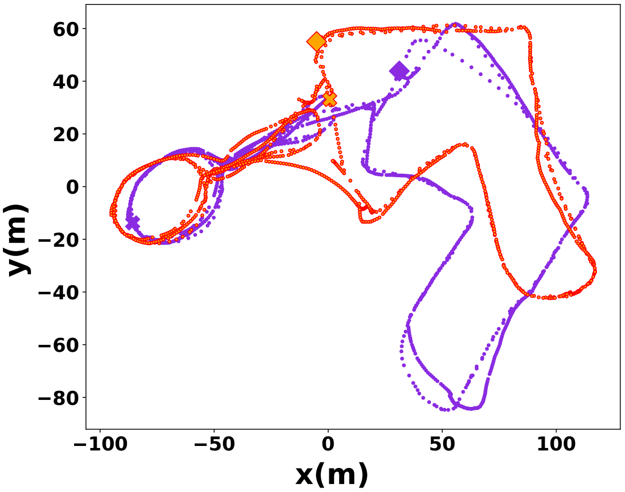

Fig. 18 ICP comparison of EM for *Oxford New College* experiment. Despite the visual difference between the final merged and single-session mapping EM (purple), both resemble to correct environment structure. RMSE = 85.4687

The partial map of the third session is quickly merged since it starts in a known place inside the first session loaded map (Fig. 17**d**, red circle). After completing its trajectory in the first session area, the resulting map is merged into the second session loaded map at its expected location (Fig. 17**e**, red circle).

As can be observed, in continuing the mapping, the RatSLAM algorithm applied an inconsistent path correction at the second merge point, which affected the final map (red squares, Fig. 17**e**–**f**). This inconsistency led to a clear difference between single-session and multisession maps when compared by ICP (Fig. 18, RMSE = 85.4687). However, despite this inconsistent correction, the final multisession map presents a sufficient similarity of structures that compose the environment path (see Fig. 8). Notably, the multisession map correctly closed loops where the single-session mapping failed to perform it. (green circles, Fig. 17**a**,**f**).

6 Conclusion

This work presents a novel solution to the multisession SLAM problem for the RatSLAM algorithm. Specifically, the proposed solution considers the multisession mapping scenario where the robot is not localized within the previously generated map, and newly mapped areas have to be merged with previously mapped ones into a single coherent map of the environment.

The merging of maps is obtained from complete structures (LVC, PCN, and EM) acquired and stored in previous mapping sessions. The proposed approach merges the map under construction with previously stored maps when both maps share a template, adjusting all structures spatially and producing a new partial map. After merging, the mapping session continues, following the standard algorithm, and can perform new merging procedures.

Four different environments were explored in this work: the *Virtual Tour* experiment, in which a virtual robot performed an ellipse path; the *Lab Tour* experiment, where a robotic platform called RoboDeck mapped a research laboratory; the *iRat* dataset used to validate the openRatSLAM implementation [21]; and the challenging New College dataset [41]. The *Virtual Tour* and *Lab Tour* experiments were divided into two mapping sessions, while the New College and *iRat* experiments were separated into three and five mapping sessions, respectively.

The multisession maps obtained were comparable to single-mapping maps in several experiments. For all experiments, the final EMs were similar to the EMs of the single mapping session. On the other hand, path deviation increased with the number of mapping sessions and in long-term and large-scale mapping scenarios, as shown by the EMs of iRat and the New College experiment, respectively. In the former, the difference between them may be due to the absence of connection between the nodes of the internal and external laps. In the latter case, the path deviation occurred after an inconsistent path correction after the merge point. Regardless of path distortions, the final maps of both resemble the single-session map, which shows a potential application for the merge approach in more than two sessions.

Furthermore, the experimental results showed: a) path corrections performed in EM; b) no new experiences created in the *Virtual Tour* experiment; c) the PCN activity packages of *Virtual Tour* in the single mapping session are similar to the PCN of the multisession; d) EM matching in real-world *Lab Tour*, and realistic *iRat* experiments; e) loop closure correctly performed when compared to single-session mapping in the *New College* dataset. Therefore, the present work shows that the proposed merge mechanism addresses the multisession solutions (i) and (ii), and can be used as a solution for the RatSLAM algorithm.

In future work, a deeper investigation of the results presented on the New College experiment should be conducted. With this, improvements in the merge mechanism could be expected for large-size long-term environments. Moreover, the overall merge mechanism could be adapted to compare the current under-construction map with multiple previously stored maps.

Acknowledgements Our research group acknowledges financial support from FAPEMA/COOPI COOPI-05109/18, CAPES/BRAZIL Finance Code 001 and CNPq/BRAZIL (Projects 309505/2020-8 and 420109/2018-8). Sen Cheng was supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) - project number 316803389 - SFB 1280, A14. Edison Pignaton de Freitas also thanks to the Ruhr-Universität Bochum - Research School PLUS, for the Visiting International Professor (VIP) grant.

Author Contributions Matheus: Method conception/design, implementation and tests. Wrote original draft, revised and edited the manuscript. Mauro: Method conception/design. Revised and edited the manuscript. Edison: Revised and edited the manuscript. Sen Cheng: Revised and edited the manuscript. Areolino: Revised and edited the manuscript. Paulo: Wrote original draft, revised and edited the manuscript. Alexandre: Wrote original draft, revised and edited the manuscript.

Funding Financial support from FAPEMA/COOPI COOPI-05109/18, CAPES/BRAZIL (Finance Code 001), CNPq projects 309505/2020-8 and 420109/2018-8, and German Research Foundation (DFG) - project number 316803389 - SFB 1280, A14. Edison Pignaton de Freitas also thanks to the Ruhr-Universität Bochum - Research School PLUS, for the Visiting International Professor (VIP) grant. Alexandre Oliveira also thanks to the Science Foundation Ireland (SFI) grant SFI/16/RC/3918 (CONFIRM) and Marie Sklodowska-Curie agreement 847.577 co-funded by the European Regional Development Fund.

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Code Availability The code that supports the findings of this study is available at https://zenodo.org/badge/latestdoi/568248424.

Declaration

Conflicts of interest The authors have no conflicts of interest.

References

1. Pandey, A., Pandey, S., Parhi, D.: Mobile robot navigation and obstacle avoidance techniques: a review. Int. Rob. Auto. J. **2**(3), 105 (2017). https://doi.org/10.15406/iratj.2017.02.00023
2. Labbé, M., Michaud, F.: Online global loop closure detection for large-scale multi-session graph-based slam. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2661–2666 (2014). https://doi.org/10.1109/IROS.2014.6942926
3. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part i. IEEE Robot. Autom. Mag. **13**(2), 99–110 (2006). https://doi.org/10.1109/MRA.2006.1638022
4. Stachniss, C., Leonard, J.J., Thrun, S.: Simultaneous localization and mapping. In: Springer Handbook of Robotics, (2016). https://doi.org/10.1007/978-3-319-32552-1_46
5. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., et al.: Fastslam: A factored solution to the simultaneous localization and mapping problem. Aaai/iaai **593598** (2002)
6. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Trans. Robot. **23**(1), 34–46 (2007). https://doi.org/10.1109/TRO.2006.889486
7. Silva, G., Costa, J., Magalhães, T., Reis, L.P.: Cyberrescue: A pheromone approach to multi-agent rescue simulations. In: Information Systems and Technologies (CISTI), 2010 5th Iberian Conference On, pp. 1–6 (2010). IEEE
8. Bakhshipour, M., Jabbari Ghadi, M., Namdari, F.: Swarm robotics search & rescue: a novel artificial intelligence-inspired optimization approach. Appl. Soft. Comput. **57**, 708–726 (2017). https://doi.org/10.1016/j.asoc.2017.02.028
9. Cai, Y., Chen, Z., Min, H.: Improving particle swarm optimization algorithm for distributed sensing and search. In: 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 373–379 (2013). https://doi.org/10.1109/3PGCIC.2013.64
10. Ranjbar-Sahraei, B., Tuyls, K., Caliskanelli, I., Broeker, B., Claes, D., Alers, S., Weiss, G.: 13 - bio-inspired multi-robot systems. In: Ngo, T.D. (ed.) Biomimetic Technologies. Woodhead Publishing Series in Electronic and Optical Materials, pp. 273–299 (2015). https://doi.org/10.1016/B978-0-08-100249-0.00013-6. https://www.sciencedirect.com/science/article/pii/B9780081002490000136
11. Calvo, R., Oliveira, J.R.d., Figueiredo, M., Francelin Romero, R.A.: A distributed, bio-inspired coordination strategy for multiple agent systems applied to surveillance tasks in unknown environments. In: The 2011 International Joint Conference on Neural Networks, pp. 3248–3255 (2011). https://doi.org/10.1109/IJCNN.2011.6033652
12. Zeno, P.J., Patel, S., Sobh, T.M.: Review of neurobiologically based mobile robot navigation system research performed since 2000. J. Robot. **2016**, 17 (2016). https://doi.org/10.1155/2016/8637251
13. Tang, H., Yan, R., Tan, K.C.: Cognitive Navigation by Neuro-Inspired Localization, Mapping, and Episodic Memory. IEEE Trans. Cogn. Dev. Syst. **10**(3), 751–761 (2018). https://doi.org/10.1109/TCDS.2017.2776965
14. Yu, F., Shang, J., Hu, Y., Milford, M.: NeuroSLAM: a brain-inspired SLAM system for 3D environments. Biol. Cybern. (2019). https://doi.org/10.1007/s00422-019-00806-9
15. Li, J., Li, Z., Chen, F., Bicchi, A., Sun, Y., Fukuda, T.: Combined sensing, cognition, learning, and control for developing future neuro-robotics systems: a survey. IEEE Trans. Cogn. Dev. Syst. **11**(2), 148–161 (2019). https://doi.org/10.1109/TCDS.2019.2897618
16. McNaughton, B.L., Battaglia, F.P., Jensen, O., Moser, E.I., Moser, M.-B.: Path integration and the neural basis of the'cognitive map'. Nat. Rev. Neurosci. **7**(8), 663 (2006). https://doi.org/10.1038/nrn1932
17. O'Keefe, J.: Place units in the hippocampus of the freely moving rat. Exp. Neurol. **51**(1), 78–109 (1976). https://doi.org/10.1016/0014-4886(76)90055-8
18. Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., Moser, E.I.: Microstructure of a spatial map in the entorhinal cortex. Nature **436**(7052), 801 (2005). https://doi.org/10.1038/nature03721
19. Taube, J., Muller, R., Ranck, J.: Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. J. NeuroSci. 10(2), 420–435 (1990). https://doi.org/10.1523/JNEUROSCI.10-02-00420.1990
20. Milford, M.J., Wyeth, G.F., Prasser, D.: Ratslam: a hippocampal model for simultaneous localization and mapping. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, vol. 1, pp. 403–4081. https://doi.org/10.1109/ROBOT.2004.1307183 (2004)
21. Ball, D., Heath, S., Wiles, J., Wyeth, G., Corke, P., Milford, M.: Openratslam: an open source brain-based slam system. Auton.

Robot. **34**(3), 149–176 (2013). https://doi.org/10.1007/s10514-012-9317-9

22. Milford, M., Jacobson, A., Chen, Z., Wyeth, G.: RatSLAM: Using models of rodent hippocampus for robot navigation and beyond. In: Springer Tracts in Advanced Robotics, vol. 114, pp. 467–485 (2016). https://doi.org/10.1007/978-3-319-28872-7_27
23. Prasser, D., Milford, M., Wyeth, G.: Outdoor simultaneous localisation and mapping using ratslam. In: Corke, P., Sukkariah, S. (eds.) Field and Service Robotics, pp. 143–154. Springer (2006). https://doi.org/10.1007/978-3-540-33453-8_13
24. Milford, M.J., Wyeth, G.F.: Mapping a suburb with a single camera using a biologically inspired slam system. IEEE Trans. Robot. **24**(5), 1038–1053 (2008). https://doi.org/10.1109/TRO.2008.2004520
25. Milford, M., Wyeth, G.: Persistent navigation and mapping using a biologically inspired slam system. Int. J. Robot. Res. **29**(9), 1131–1153 (2010). https://doi.org/10.1177/0278364909340592
26. Menezes, M.C., de Freitas, E.P., Cheng, S., de Oliveira, A.C.M., de Almeida Ribeiro, P.R.: A neuro-inspired approach to solve a simultaneous location and mapping task using shared information in multiple robots systems. In: 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 1753–1758 (2018). https://doi.org/10.1109/ICARCV.2018.8581270
27. McDonald, J., Kaess, M., Cadena, C., Neira, J., Leonard, J.J.: Real-time 6-DOF multi-session visual SLAM over large-scale environments. Robot. Auton. Syst. **61**(10), 1144–1158 (2013). https://doi.org/10.1016/j.robot.2012.08.008
28. Labbé, M., Michaud, F.: Long-term online multi-session graph-based SPLAM with memory management **42**, 1133–1150 (2018). https://doi.org/10.1007/s10514-017-9682-5
29. Labbé, M., Michaud, F.: Online global loop closure detection for large-scale multi-session graph-based slam. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2661–2666 (2014). https://doi.org/10.1109/IROS.2014.6942926
30. 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. **37**(6), 1874–1890 (2021)
31. Daoud, H.A., Sabri, A.Q.M., Loo, C.K., Mansoor, A.M.: Slamm: Visual monocular slam with continuous mapping using multiple maps. PloS one **13**(4), 0195878 (2018)
32. Schneider, T., Dymczyk, M., Fehr, M., Egger, K., Lynen, S., Gilitschenski, I., Siegwart, R.: maplab: An open framework for research in visual-inertial mapping and localization. IEEE Robot. Autom. Lett. **3**(3), 1418–1425 (2018)
33. McDonald, J., Kaess, M., Cadena, C., Neira, J., Leonard, J.J.: Real-time 6-dof multi-session visual slam over large-scale environments. Robot. Auton. Syst. **61**(10), 1144–1158 (2013)
34. Wang, Y., Huang, S., Xiong, R., Wu, J.: A framework for multi-session rgbd slam in low dynamic workspace environment. CAAI Trans Intell Technol **1**(1), 90–103 (2016)
35. Burguera Burguera, A., Bonin-Font, F.: A trajectory-based approach to multi-session underwater visual slam using global image signatures. J Marine Science and Engineering **7**(8), 278 (2019)
36. Labbé, M., Michaud, F.: Multi-session visual slam for illumination invariant localization in indoor environments (2021). arXiv:2103.03827
37. Milford, M.J., Wiles, J., Wyeth, G.F.: Solving navigational uncertainty using grid cells on robots. PLOS Computational Biology **6**(11), 1–14 (2010). https://doi.org/10.1371/journal.pcbi.1000995
38. Besl, P.J., McKay, N.D.: Method for registration of 3-D shapes. In: Schenker, P.S. (ed.) Sensor Fusion IV: Control Paradigms and Data Structures, vol. 1611, pp. 586–606. International Society for Optics and Photonics (1992). https://doi.org/10.1117/12.57955
39. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361, IEEE (2012)
40. Walther, T., Diekmann, N., Vijayabaskaran, S., Donoso, J.R., Manahan-Vaughan, D., Wiskott, L., Cheng, S.: Context-dependent extinction learning emerging from raw sensory inputs: a reinforcement learning approach. Scientific Reports **11**(1) (2021). https://doi.org/10.1038/s41598-021-81157-z
41. Smith, M., Baldwin, I., Churchill, W., Paul, R., Newman, P.: The new college vision and laser data set. The International Journal of Robotics Research **28**(5), 595–599 (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Matheus Menezes is pursuing a Ph.D. in Computer Science at the Federal University of Maranhão (UFMA). He also completed an M.Sc. in Computer Science from the same university. His research focuses on space navigation, specifically in the interdisciplinary domains of robotics and computational neuroscience. Matheus applies computational methods such as artificial neural networks, reinforcement learning, and simultaneous localization and mapping to conduct scientific research and develop engineering solutions.

Mauro Muñoz is a researcher and scholar specializing in Artificial Intelligence. He recently completed a postdoctoral fellowship at the Federal University of Maranhão (UFMA). Dr. Muñoz holds a Ph.D. in Electronic Systems in Artificial Intelligence from the renowned Polytechnic School of the University of São Paulo (POLI/USP). His academic journey and expertise cover a wide range of disciplines, including Engineering and Computer Science, with a strong focus on Artificial Intelligence. His research primarily explores decision-making systems, informatics for biodiversity, species distribution modeling, artificial neural networks, genetic algorithms, robotics, and mobile applications.

Edison Pignaton de Freitas has a Ph.D. in computer science and engineering from Halmstad University, Sweden (2011), an M.Sc. in computer science from UFRGS (2007), and a computer engineering degree from the Military Institute of Engineering (2003). Currently, he holds an associate professor position at UFRGS, acting in the Graduate Programs on Computer Science (PPGC) and Electrical Engineering (PPGEE), developing research mainly in the following areas: computer networks, robotics, real-time systems, and (multi)unmanned aerial vehicles.

Sen Cheng is a Professor of Computational Neuroscience in the Institute for Neural Computation in the Faculty of Computer Science at Ruhr University Bochum. He studies the neural and cognitive mechanisms of episodic memory and spatial navigation using a variety of computational methods such as biological neural networks, cognitive modeling, and machine learning. His interdisciplinary research combines philosophical, psychological, neuroscientific and computational approaches. He founded and currently heads the DFG-funded research unit FOR 2812 "Constructing scenarios of the past: A new framework in episodic memory".

Areolino de Almeida Neto received his B.S. degree in electrical engineering from the Federal University of Maranhão (UFMA - S. Luís, Brazil) in 1990, his M.S. degree in mechanical engineering from the Aeronautical Institute of Technology (ITA - S. J. dos Campos, Brazil) in 1998 and his Ph.D. degree also from ITA in 2003. From 1991 until 1993, he worked as a computer engineer at Alcântara Space Center in Maranhão, Brazil. From 1993 until 1995, he was the CEO of Trend Systems Engineering in S. Luís. Since 1994 he has been a professor at UFMA, where he also was Head of the International Office (2007–2012) and Director of the Innovation and Entrepreneurship Department (2012–2017). His main research interests are neural networks, reinforcement learning, intelligent control, and mobile robotics. He is a member of the Brazilian Society of Neural Networks.

Paulo Ribeiro received the Ph.D. degree in Neuroscience from the University of Tübingen (Germany) in 2015, Master degree in Mechatronics Engineering from the University of Minho (Portugal) in 2012, and the Bachelor degree in Computer Science from the Federal University of Maranhão (Brazil) in 2009. He is an Assistant Professor (tenured position), since 2016, within the Department of Computer Engineering at the Federal University of Maranhão (UFMA) in Brazil. His research interests include Robotics, Autonomous Mobile Robot, Control and Automation Systems, Data Science, Artificial intelligence (AI), Neuroscience and Brain-Computer Interface (BCI).

Alexandre Oliveira earned his Ph.D., specializing in hybrid evolutionary algorithms for continuous and discrete search space, in 2004 at the renowned Brazilian Institute for Space Research. Dr. Muniz de Oliveira successfully defended the Memorial for Progression in the Career of the Superior Teaching in Brazil in 2021, ascending to Professor. Since 2022, he has embarked on a postdoctoral research position at the University College Cork (UCC) Insight Center for Data Analytics, focusing on investigating optimization-simulation closed-loop systems applied to Smart Manufacturing Systems.

Authors and Affiliations

Matheus Menezes[1] · Mauro Muñoz[1] · Edison Pignaton de Freitas[2] · Sen Cheng[3] · Areolino de Almeida Neto[1] · Paulo Ribeiro[1] · Alexandre Oliveira[1,4]

Matheus Menezes
menezes.matheus@lacmor.ufma.br

Mauro Muñoz
mesmunoz@gmail.com

Edison Pignaton de Freitas
edison.pignaton@inf.ufrgs.br

Sen Cheng
sen.cheng@rub.de

Areolino de Almeida Neto
areolino.neto@ufma.br

Paulo Ribeiro
paulo.ribeiro@ecp.ufma.br

[1] Federal University of Maranhão, São Luís, Brazil

[2] Graduate Program on Computer Science, Federal University of Rio Grande do Sul, Porto Alegre, Brazil

[3] Institute for Neural Computation, Ruhr University Bochum, Bochum, Germany

[4] School of Computer Science and Information Technology, University College Cork, Cork, Ireland