



Topological Gaussian ARAM for biologically inspired topological map building

Wei Hong Chin¹ · Chu Kiong Loo¹

Received: 19 April 2014 / Accepted: 20 July 2016 / Published online: 6 August 2016
© The Natural Computing Applications Forum 2016

Abstract This paper presents a new neural network for online topological map building inspired by beta oscillations and hippocampal place cell learning. The memory layer represents the hippocampus, the input layer represents the entorhinal, and the ρ is the orientation system. In this model, multiple-scale entorhinal grid cell activations form the input layer feature patterns, which are categorized by hippocampal place cells (nodes) and act as spatial categories in the memory layer. Top-down attentive matching and mismatch-mediated reset (beta oscillations), which are triggered by the orientation system, overcome the stability-plasticity dilemma and prevent the catastrophic forgetting of place cell maps. In our proposed method, nodes in the topological map represent place cells (robot location), while edges connect nodes and store robot action (i.e., orientation, direction). Our method is based upon a multi-channel Adaptive Resonance Associative Memory (ARAM) network architecture to obtain multiple sensory sources for topological map building. It comprises two layers: input and memory. The input layer collects sensory data and incrementally clusters the obtained information into a set of topological nodes. In the memory layer, the clustered information is used as a topological map where nodes are associated with actions. The advantages of the proposed method are: (1) it does not require high-level cognitive processes and prior knowledge to make it work in a natural environment; and (2) it can process multiple sensory sources simultaneously in continuous space, which is crucial for real-world robot navigation. Thus, we

combine our Topological Gaussian ARAM method (TGARAM) with incremental principle component analysis to constitute a basis for topological map building. Lastly, the proposed method was validated using several standardized benchmark datasets.

Keywords Adaptive resonance theory · SLAM · Place cell learning · Topological map · Unsupervised learning

1 Introduction

Autonomous mobile robots are able to move in a given environment and can perform desired tasks and navigate in unstructured environments with little or no human intervention.

A human-friendly autonomous guided robot knows at least a little about where it is and how to reach a particular location in order to achieve certain goals. It can construct a map of the environment according to its position and posture (mapping) and estimate its position and posture using a built environment map (localization). Building the representation of the map is crucial to autonomous navigation and therefore a reliable map not only improves maintenance, but also map-based localization and path planning in any environment.

In mobile robotics, representations of the world are grouped into metric maps, topological maps and hybrid models that combine both metric and topological information [32]. In the metric mapping framework, the environment is represented as a set of objects with coordinates in a 2D space. The construction of the map is based on a grid occupancy or feature map approach [23]. In the grid occupancy approach, the environment is mapped as an array of cells. However, this approach is limited by the

✉ Wei Hong Chin
weihong118118@gmail.com

¹ Faculty of Computer Science and Information Technology,
University of Malaya, Kuala Lumpur, Malaysia

high computational cost of feature matching. Pure metric map approaches are susceptible to inaccuracies in both map-making and the dead-reckoning abilities of the robot [11].

In the topological framework, the environment is represented by a set of distinct places [3] and how a robot travels from one place to another. Places are defined by information gathered from sensors placed in the environment, which is stored in nodes. Some of the robot's odometry information (gathered while it travels from one place to another) is stored in the links of the map. Therefore, a topological map is a sparse representation of the environment that only includes important places used for navigation gathered from sensor information, and connections between these places gathered from the robot's odometry information.

The first advantage of topological maps is that they do not require a metric sensor model to convert sensor data into a 2D frame of reference [12]. However, topological maps can lack the details found in an environment. To overcome these problems hybrid approaches [9, 10], which combine the topological and metric paradigms, have been proposed to compensate for the weakness of each.

However, an important factor that limits the use of topological maps is the lack of uniform semantics associated with them. For example, [3] represents nodes as places characterized by sensor data and edges as paths between places characterized by control strategies. On the other hand, [31] developed maps by dividing a probabilistic occupancy grid into regions separated by narrow passages according to a measure of local clearance. Van Zwansvorde et al. [33] obtained topological information from a grid map by analyzing it as a grayscale image. However, this approach requires a large memory allocation to store the grid map. As an alternative, the construction of a straightforward topological map using the generalized Voronoi graph (GVG) was described in [30, 33]. However, GVG is vulnerable in dynamic, large-scale environments with sharp-edged obstacles and node extraction and matching is computationally demanding. Kwon and Song [22] introduced a thinning method to build a topological map from a binary grid map. Although this approach requires less computational power than the GVG method, the method is based on an existing grid map, which limits its application in online map building [28].

The other important factor that impedes the application of the topological map is the online detection and recognition of topological nodes. While many types of artificial landmarks (such as ultrasonic beacons, visual patterns or reflective tape) are available, which offer fast and stable recognition of a particular location, the problem remains that there are no artificial landmarks in unknown environments.

1.1 Emulating biological system

Another area of research has focused on emulating the biological systems thought to be the basis for mapping and navigation in animals. The hippocampus of rodents is one of the most studied brain regions of any mammal. Early work with rats identified place cells in their hippocampus that appeared to respond to the animal's spatial location [27]. Other research has discovered that beta oscillations occur during the learning of hippocampal place cell receptive fields in novel environments [4].

Beta oscillations explain how place cells may be learned as spatially selective categories from the feedback between the entorhinal cortex and the hippocampus. The aim of the Psikharpx project is to create an artificial robotic rat, driven by mapping and navigation algorithms that mimic place cells [20]. More recent approaches have used grid cells [15] to enable navigation and mapping on a robot in a larger room-sized area. Weitzenfeld and Barrera have developed a biologically inspired robot architecture with spatial cognition and navigation capabilities. Their system is capable of learning and forgetting specified locations and can navigate to them from any point on their map [2].

Another interesting method for biologically inspired navigation is RatSLAM [1, 26], which builds a topological map with metric information, by separating the topological and metric layer. This approach requires a proper scaling of the map in advance, because it cannot be changed efficiently at runtime.

In this paper, we describe a hippocampus-inspired neural model for online topological map construction that we call Topological Gaussian ARAM (TGARAM). Nodes in the topological map represent place cells (the location of the robot), while edges connect nodes and store robot action (i.e., orientation, direction). TGARAM has been developed from the Adaptive Resonance Theory (ART) framework. We selected the ART framework because it is a fast, unsupervised learning algorithm and has been used to explain how place cells learn [16]. Moreover, ART addresses the stability-plasticity dilemma [6], which explains how the brain can both quickly learn to categorize information in the real world and remember it without forgetting previously learned knowledge.

Research to date suggests that beta oscillations are triggered during mismatch states of ART learning mechanisms. Furthermore, the model has been extended to explain and simulate data at multiple levels of brain organization [17]. Based on these existing strengths, our proposed method models place cell learning in the hippocampus and overcomes the problem of online detection and recognition of topological nodes. The proposed learning model can simultaneously build and maintain the topological map and works in natural environments.

The metric map does not need to be constructed in advance, and because it is based on a simple mathematical model, our proposed method requires less computational power than any of the other map building methods mentioned above.

The remainder of this paper is organized as follows. Section 2 introduces the construction of the online topological map using TGARAM. The experimental results of map building are summarized in Sect. 3 and discussed in Sect. 4. Finally, Sect. 5 presents some conclusions.

2 Proposed method

TGARAM integrates the Gaussian ART [34] and the incremental topology-preserving mechanism. The reason we used the Gaussian ART as a learning algorithm was because it is more robust to noise and learns a more efficient internal representation of a mapping. The topology-preserving mechanism of the Growing Neural Gas (GNG) model was used to construct a topological node map in which nodes are connected by edges.

Moreover, we modified the ARAM network becomes multi-channel architecture [29] to learn multiple mappings simultaneously across multi-modal feature patterns from multiple sensors. In order to construct a topological map based on the TGARAM method, a number of nodes are created that correspond to the number of distinct places determined by the robot (i.e., one node for each place). Nodes are connected by directed edges.

To build the topological map, we divided the system into two layers, as shown in Fig. 1. The memory layer represents the hippocampus, the input layer represents the entorhinal, and the ρ is the orientation system. In this model, multiple-scale entorhinal grid cell activations form the input layer feature patterns, which are categorized by hippocampal place cells (nodes) and act as spatial categories in the memory layer. Top-down attentive matching and mismatch-mediated reset (beta oscillations), which are triggered by the orientation system, overcome the stability-

plasticity dilemma and prevent the catastrophic forgetting of place cell maps. The first layer is the input layer, which obtains sensory information and clusters it. To do this, we utilized the Gaussian ART.

As mentioned in previous section, TGARAM is difference with other biologically inspired method, for instance RatSLAM. Table 1 shows the difference between TGARAM and RatSLAM.

2.1 Beta oscillations and place cells learning

Berke et al. [4] have stated that beta oscillations occur during the learning of hippocampal place cell receptive fields in novel environments. Beta power was every low during the first lap of exploration, grew to full strength as a mouse traversed a lap for the second and third times, became low again after the first two minutes of exploration and remained low on following days of exploration. Beta oscillation power also correlated with the rate of which place cells became spatially selective. Given the rapidity with which place cell learning happened, and the high increase in beta activity during the second exposure to the environment, it could be a highly selective learning mechanism at work.

Grossberg [16] proposed an explanation of these data that consolidates two parallel streams of modeling activity and that suggested testable predictions aimed at analyzing the underlying neural mechanisms. In our proposed method, Gaussian ART can be explained to accomplish the place cell learning. The first property is the primary fact of *fast learning and stable memory of place cells*. The second property concerns how *top-down expectations* are learned. Table 2 shows the explanation of how TGARAM achieves these place cell learning properties.

This line of reasoning calls attention to the following basic question: Is there a relationship between mismatch states and beta oscillations? Were this the case, then a simple explanation would be expected of why beta oscillations are not seen during the first lap (on the first learning

Fig. 1 TGARAM architecture and learning: the memory layer represents the hippocampus, the input layer represents the entorhinal, and the ρ is the orientation system

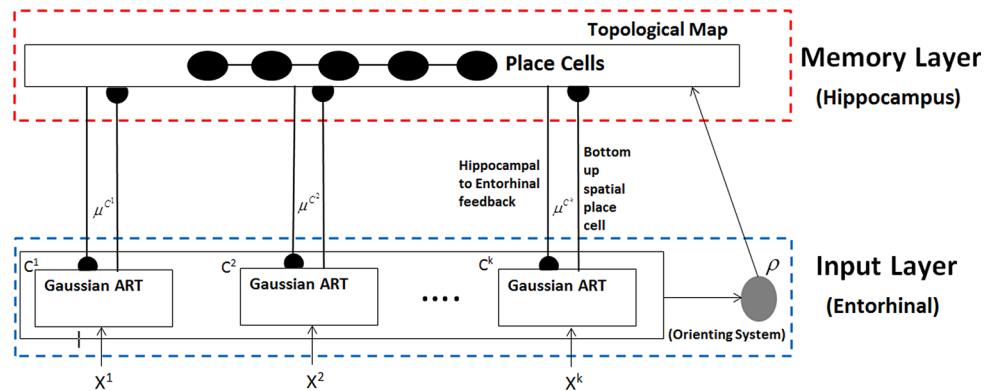


Table 1 Comparison between RatSLAM and TGARAM

RatSLAM	TGARAM
Topological map with metric information	Topological map with sensory information and robot odometry
Has separate topological map and metric map	Has only one topological map
Require proper scaling of map in advance	Does not require scaling of map in advance

Table 2 Relationship between the principle of the TGARAM and the place cell learning

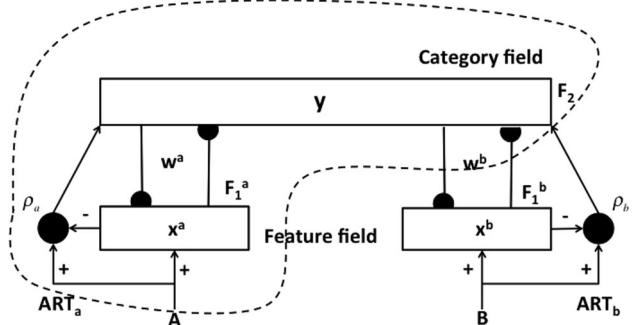
Place cell learning property	TGARAM property
Fast learning and stable memory of place cells. <i>Every Input Pattern Can Initially be Matched</i>	Top-down adaptive weights start out large, so that they can match incoming input pattern New node's mean μ will be set to the input pattern, and its standard deviation σ will be set to a large value of γ Subsequent learning trials refine these adaptive weights by updating the Gaussian distribution
How <i>top-down expectations</i> are learned. <i>Beta Oscillations during Mismatch and Reset</i>	Updating the Gaussian distribution during the learning of a particular node's top-down expectation gradually selects a pattern of critical features which constitute the attentional focus that is activated by the node Following learning trials that refine a node and its critical feature pattern, and that select new nodes for learning, will cause mismatches Mismatches inhibit node learning Nodes recognition is controlled by a vigilance parameter ρ

trial, there are no mismatches), why they begin during the second lap and are corresponded with the rate at which place cells became selective (mismatches occur when learning is refined), and finally why they are attenuated after place cell learning stabilizes (no more mismatches occur).

The mathematical mechanisms that cause beta oscillations during mismatch states are not specific to the features that are categorized and learned. Thus, Grossberg [16] stated if it were the case that place cells are learned as spatial categories in the hippocampal system, and that top-down attentive feedback helps to stabilize their rapid learning through time through a matching process, then it explains why beta oscillations occur with the properties that they exhibit during place cell learning. Moreover, Berke et al. [4] reported that “the extent of beta-entrainment predicted the improvement in spatial specificity between the first 2 min.” When place cell receptive fields are stabilized, there will be fewer mismatches, and beta oscillations will fade.

2.2 Comparison between ARAM and TGARAM

In this section, we will explain about the conventional ARAM and highlight the difference between ARAM and our proposed method. Adaptive Resonance Associative Map (ARAM) is an extension of unsupervised learning

**Fig. 2** Adaptive Resonance Associative Map architecture

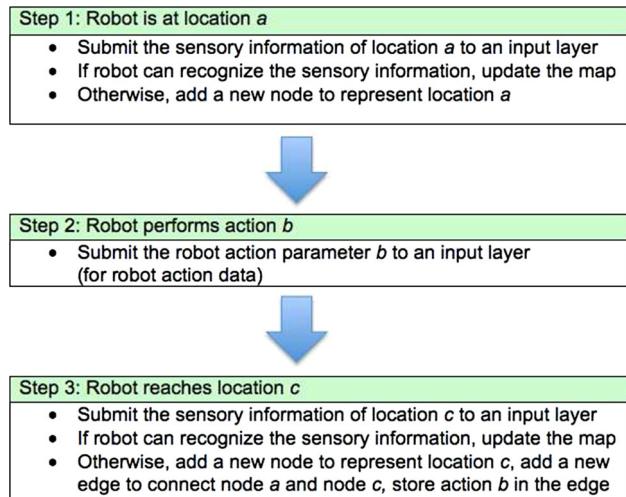
Adaptive Resonance Theory (ART) for rapid, yet stable, heteroassociative learning. ARAM can be visualized as two overlapping ART networks sharing a single category field. Figure 2 shows the ARAM architecture, and Table 3 summarizes the comparisons of ARAM and TGARAM.

2.3 Building a topological map online with TGARAM

TGARAM learns in an environment that is perceived from continuous sensorial information provided by multiple sensory sources. The M-dimensional sensorial information is transmitted to the Gaussian ART channel for online

Table 3 Comparison between ARAM and TGARAM

ARAM	TGARAM
Utilize fuzzy ART	Utilize Gaussian ART
$T_j = \gamma \frac{ A \wedge w_j^a }{z_a + w_j^a } + (1 - \gamma) \frac{ B \wedge w_j^b }{z_b + w_j^b }$	$G_j = \sum_{k=1}^K \alpha^k \left[\exp \left(-\frac{1}{2} \sum_{i=1}^M \left(\frac{x_i^k - \mu_{ji}^k}{\sigma_{ji}^k} \right)^2 \right) \right]$
Only two ART modules in input layer	More than two ART modules in input layer
Two vigilance parameter	Only one vigilance parameter
No topological map in upper layer	Topological map is in upper layer
Update first and second winner	Update only first winner
$w_j^{a(\text{new})} = (1 - \beta_a)w_j^{a(\text{old})} + \beta_a(A \wedge w_j^{a(\text{old})})$	$\mu_{ji,\text{new}}^k = (1 - n_{j,\text{new}}^{-1})\mu_{ji,\text{old}}^k + n_{j,\text{new}}^{-1}x_i^k$
$w_j^{b(\text{new})} = (1 - \beta_b)w_j^{b(\text{old})} + \beta_b(B \wedge w_j^{b(\text{old})})$	

**Fig. 3** Topological map building process

detection and learning, and this information is stored in a node of the topological map. Odometry information is transmitted to the Gaussian ART channel, both for learning purposes and to distinguish similar sensory information in different places.

In topological map building and learning it is often necessary to be able to recognize whether the current sensory information corresponds to a node in the map, and adding a new node if not; link information is then updated based on the robot's action.

In summary, the proposed method is a robust online and incremental learning system, which is suitable for robot mapping and navigation in a real-world environment. TGARAM incrementally produces nodes to learn the new feature patterns detected by the robot at a particular location as it navigates through the environment. Therefore, it is possible to identify where the robot has already been through a simple comparison of the current sensorial information with the nodes already in the topological map (online detection and recognition).

Consequently, the TGARAM learning system mimics place cell learning described earlier. Furthermore, as nodes are connected by edges and form a topological map, the robot can perform path planning from one location to another. The overall process is as shown in Fig. 3, and learning algorithm is as shown in Algorithm 1.

Algorithm 1: TGARAM algorithm

```

TGARAM
Data: Sensor information
Result: Topological map
if memory layer < one node then
    add node to layer;
    return true
else
    search nodes to determine winner;
end
if winner > vigilance then
    update winner;
    if winner and previous winner no edge then
        add edge;
    else
        update edge;
    end
else
    reset winner;
    add node to layer ;
    add edge;
end
    
```

2.4 Node definition

In our topological map, nodes are defined as areas where perceptions are similar, given the robot's position in the area. This method solves the point-of-view problem. Place definitions are directly obtained through Gaussian ART categorization of sensory information, the category of a perception corresponding to the place where the robot is positioned.

Each node contains an input vector, \mathbf{V} encoded from robot's odometer. It is defined as a Gaussian distribution, with mean, μ_j , standard deviation, σ_j in each dimension,

and a priori probability. Such node definition is based solely on the robot's perceptual capacities and does not rely on a human definition of what a place is supposed to be. This makes places easier to recognize from sensory information.

2.5 Node detection and matching

The learning algorithm grows its network beginning with one node and the first perception. Next, the topological map is continuously updated according to information from the robot's sensors. During learning, the Gaussian ART filter activates nodes in the topological map that learn recognition categories, or compressed representations, of sensory information.

These nodes compete with one another to choose the node that receives the largest total input (node detection). The winning node, W_J , is the node that has the highest match value, G_J , and satisfies the match criterion, which is determined by a vigilance parameter, ρ (node recognition). Match is a measure, obtained from the node's unit height Gaussian distribution, of how similar an input is to the node's mean, relative to its standard deviation, calculated as follows:

$$G_j = \exp\left(-\frac{1}{2} \sum_{i=1}^M \left(\frac{x_i - \mu_{ji}}{\sigma_{ji}}\right)^2\right) \quad (1)$$

If the training involves K sensory channels, the match value, G_J for each node, is:

$$G_j = \sum_{k=1}^K \alpha^k \left[\exp\left(-\frac{1}{2} \sum_{i=1}^M \left(\frac{x_i^k - \mu_{ji}^k}{\sigma_{ji}^k}\right)^2\right) \right] \quad (2)$$

$$\alpha^k = \frac{1}{K} \quad (3)$$

α_k decides the influence of each channel, and the sum of α_k is 1. If G_J is larger than the vigilance parameter, ρ , the winning node is most similar to current sensory information. Next, it learns feature patterns from multiple sensory channels; its count, mean and standard deviation are updated to represent feature patterns:

$$n_{J,\text{new}} = n_{J,\text{old}} + 1 \quad (4)$$

$$\mu_{Ji,\text{new}}^k = (1 - n_{J,\text{new}}^{-1})\mu_{Ji,\text{old}}^k + n_{J,\text{new}}^{-1}x_i^k \quad (5)$$

$$\sigma_{Ji,\text{new}}^k = \begin{cases} \sqrt{(1 - n_{J,\text{new}}^{-1})\sigma_{Ji,\text{old}}^{2k} + n_{J,\text{new}}^{-1}(x_i^k - \mu_{Ji,\text{new}}^k)^2} & \text{if } n_{J,\text{new}} > 1 \\ \gamma & \text{otherwise} \end{cases} \quad (6)$$

The initial standard deviation, γ , determines the isotropic spread in the feature space of a new node's distribution about its first sample. A mismatch occurs if any channel k

of the winning node's match value, G_J^k , is smaller than the vigilance parameter, ρ ; the node is then reset and it remains inactive until the next input arrives (beta oscillation). Next, the node with the largest match value is chosen for learning.

If no existing node satisfies the match criterion, the robot is located in a new place. Then a new node is added, and the robot's position and Gaussian elements corresponding to current sensory information are stored in the node to represent this new place. Thus, TGARAM incrementally produces the number of nodes necessary to learn and remember feature patterns from the environment.

2.6 Node localization

After the topological map construction, the node detection and matching are achieved by comparing sensory information around current robot location with nodes of topological map. However, the TGARAM has to cope with perceptual ambiguities, this strategy is useful when a never-seen-before sensory situation is encountered, but an already-known sensory situation may correspond either to an already visited place or to a new place. To differentiate between these two cases, ideothetic information has to be taken into account. Figure 4 explains this situation and shows how TGARAM topological map can overcome the problem.

In our proposed method, Eq. 3 will be used to determine the best matching node each time. If the best matching node's G_J is higher than vigilance parameter ρ , then we conclude that the best matching node is recognized the sensory information around current robot location. During the matching process, the ideothetic information (odometry dataset) is playing an important role to disambiguate similar sensory information at different locations. For example, consider there are two nodes that have similar sensor information ($G_A^{\text{laser}} = G_B^{\text{laser}}$) and equations are shown as follows:

$$G_A = G_A^{\text{laser front}} + G_A^{\text{laser back}} + G_A^{\text{odometry}} \quad (7)$$

$$G_B = G_B^{\text{laser front}} + G_B^{\text{laser back}} + G_B^{\text{odometry}} \quad (8)$$

However, the ideothetic information (odometry) is distinct for every node. Thus, $G_A^{\text{odometry}} \neq G_B^{\text{odometry}}$ and lead to $G_A \neq G_B$.

2.7 Topological map maintenance

The topological map algorithm will continue to add nodes corresponding to changes in the environment as the robot operates. This means that the robot continually updates and maintains its representation of the environment. For instance, if one day the robot travels along a corridor with an open door, it creates a node that captures the open door.

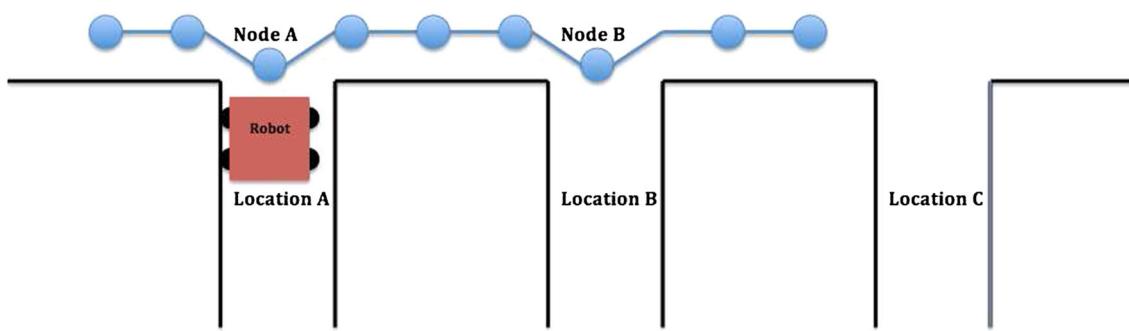
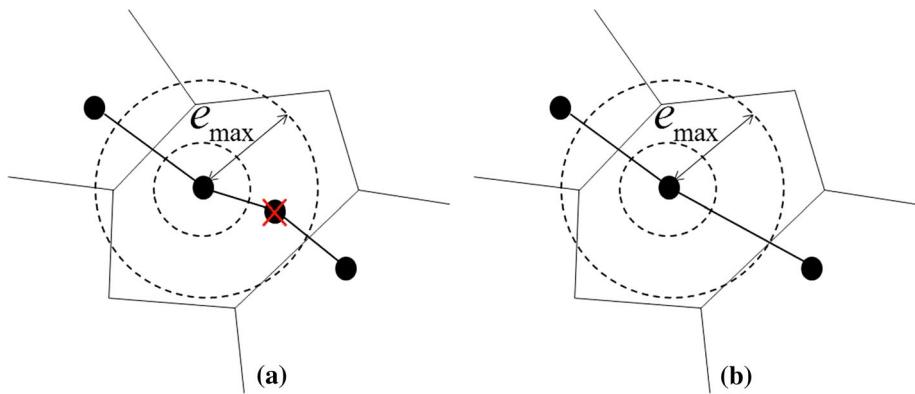


Fig. 4 Location A and location B are almost identical in the map. If robot is placed at location A, node A's G_A will be higher than node B's G_B and meet vigilance parameter; thus, node A is chosen for localization. Although node B has similar sensory information, its location is far from the robot that causes its G_B become smaller and

will not choose for localization. However, if robot is placed at location C, both node A and node B will not be chosen for localization because both nodes are far from the robot. Therefore, robot will detect it as novel situation and new node is added to record the environment

Fig. 5 Topological map pruning. **a** Nodes are removed to maintain a node in a given radius e_{\max} . **b** Transitions to or from removed nodes are reconnected to remaining nodes



If the next day the door is closed, a new node will be created with the door closed. Overlapping nodes mean that the same position with different door conditions can be mapped. This property is a direct result of the algorithm's biological origins.

The problem with this map maintenance method is that the robot remembers *all* environmental changes throughout its lifetime, which creates an unmanageable list of nodes to search and update. Therefore, the list of nodes must be pruned in order to be able to navigate and map in a dynamic environment over an indefinite time period.

Node pruning is accomplished by combining parts of the topological map that are beyond a maximum spatial density threshold. Therefore, the number of nodes grown is proportional to the size of the environment that has been explored, but not time. Figure 5 illustrates the node pruning process. The pruning algorithm used in this paper is based on the Instantaneous Topological Map (ITM) technique [21].

The algorithm prunes nodes when a new node is created in the topological map. If, as a result of the new node, other nodes lie within the Thales sphere with given radius e_{\max} as shown in Eq. 9, then these nodes are removed from the map.

Transitions to deleted nodes are either deleted or updated. For example, odometric information for the new link can be obtained from the current position of nodes in the map.

$$\text{Distance}_{\text{nodes,new}} = \sqrt{(x_{\text{nodes},x} - \mu_{\text{new},x})^2 + (x_{\text{nodes},y} - \mu_{\text{new},y})^2} \quad (9)$$

2.8 Incremental principle component analysis (IPCA)

It requires high computational cost to process high-dimensional sensory information such as laser scanner and image. To reduce the computational cost and high storage demands, we adapted Incremental Principle Component Analysis (IPCA) [8, 18, 19, 35] to reduce the dimension of sensory information before feeding to TGARAM for topological map building. IPCA offers means of representing the laser scanner measurements in a low-dimensional subspace, which allows for efficient matching and recognition.

The process of IPCA is updating the eigenvectors, eigenvalues and mean value by a single new laser scanner

measurement. We assume that a set of eigenvectors $u_j, j = 1 \dots p$ is built, after having used the laser scanner measurement $x_i, i = 1 \dots n$ as an input. The corresponding eigenvalues are λ , and the mean laser scanner measurement is \bar{x} . We would like to update this information to take into account a new laser scanner measurement x_{n+1} .

In addition, we summarize the method proposed in [19] and integrate it with TGARAM. First, we update the mean:

$$\bar{x}' = \frac{1}{n+1} (n\bar{x} + x_{n+1}) \quad (10)$$

The covariance matrix can be updated as well:

$$C' = \frac{n}{n+1} C + \frac{n}{(n+1)^2} (x_{n+1} - \bar{x})(x_{n+1} - \bar{x})^T \quad (11)$$

Then, we have to update the set of eigenvectors in order to reflect the additional laser scanner measurement x_{n+1} . Since the residual vector $h_{n+1} = (Ua_{n+1} + \bar{x}) - x_{n+1}$ is orthogonal to each eigenvector in U , its normalized equivalent is a suitable candidate for the additional vector in the basis:

$$\hat{h}_{n+1} = \begin{cases} \frac{h(n+1)}{\|h(n+1)\|_2} & \text{if } \|h(n+1)\|_2 \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Next, we obtain the new $m \times (k+1)$ matrix of eigenvectors U' by appending \hat{h} to the eigenvectors U and rotating them:

$$U' = [U \hat{h}_{n+1}] R \quad (13)$$

where R is a $(k+1) \times (k+1)$ rotation matrix. R is the solution to the eigenproblem of the following form:

$$DR = R\Lambda' \quad (14)$$

D is a $(k+1) \times (k+1)$ matrix consisting of known components of Λ and x_{n+1} , and Λ' is a diagonal matrix of the new eigenvalues. According to [19], we can construct D as

$$D = \frac{n}{n+1} \begin{bmatrix} \Lambda & 0 \\ 0^T & 0 \end{bmatrix} + \frac{n}{(n+1)^2} \begin{bmatrix} aa^T & \gamma a \\ \gamma a^T & \gamma^2 \end{bmatrix} \quad (15)$$

where $\gamma = \hat{h}_{n+1}(x_{n+1} - \bar{x})$ and $a = U^T(x_{n+1} - \bar{x})$.

There are many ways for constructing D . However, only the method mentioned in [19] allows for the updating mean. Other methods for instance [8, 24], assume the mean vector is at the origin.

2.9 Updating sensor representations

If we want to execute a true online incremental algorithm without keeping the original laser scanner data in

the memory until the model has been constructed, we have to manage the laser scanner data, respectively. Note that in the process of learning, after we update the subset of eigenvectors, we also have to update all of the representations of all the laser scanner data corresponding to the new basis set. This is due to the fact that these estimations are the only form in which we store the laser scanner data throughout the process. The priority focuses on how to update and preserve these estimations.

During the process of learning at a discrete time n , we learn n laser scanner data $x_i, i = 1 \dots n$, which has generated a space of k eigenvectors $u_j, j = 1 \dots k$. The laser scanner data are demonstrated with coefficient vectors $a_{i(n)}, i = 1 \dots n$.

When a new observation x_{n+1} arrives, we calculate the new mean \bar{x}' using (10), and we then compose the intermediate matrix D (15) and solve the eigenproblem (14). Equation (13) then generates an updated subspace base U' , but with no laser scanner representations.

In order to remap the coefficients $a_{i(n)}$ into this new subspace, we first have to recalculate each laser scanner data using the old eigenvectors U :

$$x_{i(n)} = Ua_{i(n)} + \bar{x}, i = 1 \dots n, \quad (16)$$

and project it to the new subspace of eigenvectors:

$$a_{i(n+1)} = (U')^T(x_{i(n)} - \bar{x}'), i = 1 \dots n+1, \quad (17)$$

where $x_{n+1(n)} = x_{n+1}$.

We can combine (13) and (16) into (17) and calculate the new coefficients directly:

$$a_{i(n+1)} = (R')^T \begin{bmatrix} a_{i(n)} \\ 0 \end{bmatrix} + \eta, \quad (18)$$

where η is a vector computed only once for all coefficients as

$$\eta = [U\hat{h}_{n+1}]^T(\bar{x} - \bar{x}') \quad (19)$$

The transformation mentioned above provides a model that represents the input laser scanner data with the same accuracy as the previous one; thus, we can now eliminate the old subspace and the coefficients that represented the laser scanner data in it. x_{n+1} is represented precisely as well, so we can safely eliminate it. The representation of all $n+1$ laser scanner data is possible because the subspace is extended by $k+1$ eigenvectors, which is an *increase* of the model's dimensionality.

Since we want to keep the size of the representation in low dimension, we apply method [14] to determine the dimensionality k . Next, the reduced dimension laser scanner date will feed into TGARAM.

3 Benchmarking and experimental results

The performance of TGARAM in topological map construction was verified in several experiments with different benchmark datasets. The quantization error (QE) measures the quality of the topological map and has been used to benchmark a series of maps that have been trained on the same dataset. For any given benchmark dataset, the quantization error can be reduced by simply increasing the number of nodes, as data samples are distributed more sparsely on the map.

Benchmark datasets were downloaded from the Rawseeds project, which aims to produce benchmarking tools for robotic systems. The toolkit is designed to significantly reduce the time and effort needed to develop successful novel algorithms or products by eliminating the need for expensive data acquisition operations.

Figures 6 and 7 illustrate the location available to the robot during Rawseeds data gathering. Several indoor and outdoor scenarios have been defined by Rawseeds that each generate a dataset.

The project was funded by the European Commission under the Sixth Framework Programme, and it has successfully created and published a high-quality benchmarking toolkit [7, 13]. These datasets were used as training samples to validate our proposed method, supplemented by odometry and laser scanner datasets.

3.1 Quantization error

Several measures have been used to evaluate the quality of a topological map. A widely used measurement is the quantization error, which measures the average distance

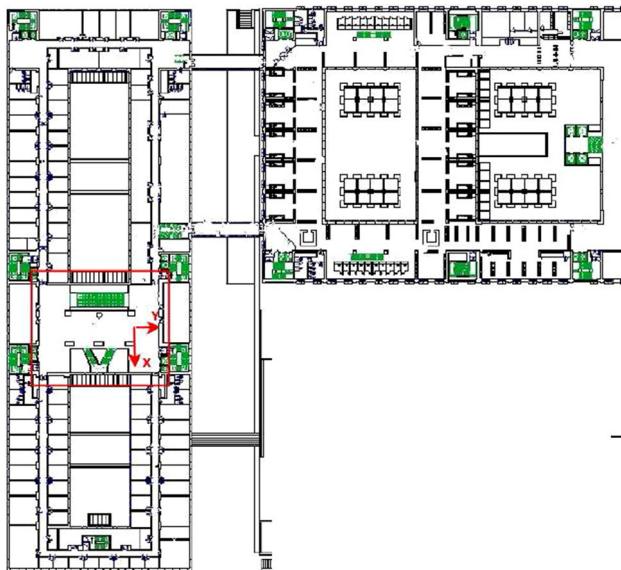


Fig. 6 Layout of the environment where the ground truth collection systems were located



Fig. 7 Satellite view of the Politecnico di Milano campus (*Source:* Rawseeds Deliverable 2.1)

between each data vector and its best matching unit (BMU). In our case, the BMU is the winning node that has the highest match value and matches the vigilance parameter. Equation 20 shows the calculation of the QE, where N is the number of data vectors and $m_{\mathbf{x}_i}$ is the best matching node in the corresponding \mathbf{x}_i vector:

$$QE = \frac{1}{N} \sum \left\| \frac{\mathbf{x}_i - m_{\mathbf{x}_i}}{\mathbf{x}_{i(\max)}} \right\| \quad (20)$$

This error evaluates the fit of the topological map to the robot's actual path. Thus, the optimal topological map is expected to yield the smallest average QE. The smaller the QE, the smaller the average of the distance from the best matching nodes to the robot's actual path, which means that the topological map is closer to the original path [25].

3.2 Compression ratio

Compression ratio (CR) is another measurement used to evaluate the map. It is defined as the ratio between the topological map generated by TGARAM and the robot's actual path and is calculated as shown in Eq. 21.

$$CR = \frac{\text{Original robot's path size}}{\text{Topological map size}} \quad (21)$$

3.3 Experiment design

The topological map forms the basis for robot navigation. Nodes are plotted as a color circle at the (x, y) coordinates in the stored position. Linked nodes are joined with a line. The Gaussian elements in the node are important for the efficient

Table 4 Indoor and outdoor dataset experiment parameters setting

Parameters	Indoor experiment	Outdoor experiment
α	0.2 (odometry channel)	0.2 (odometry channel)
	0.4 (front laser scanner channel)	0.4 (front laser scanner channel)
	0.4 (rear laser scanner channel)	0.4 (rear laser scanner channel)
γ	2 and 4	2 and 4
	0.1 to 0.9	0.1 to 0.9

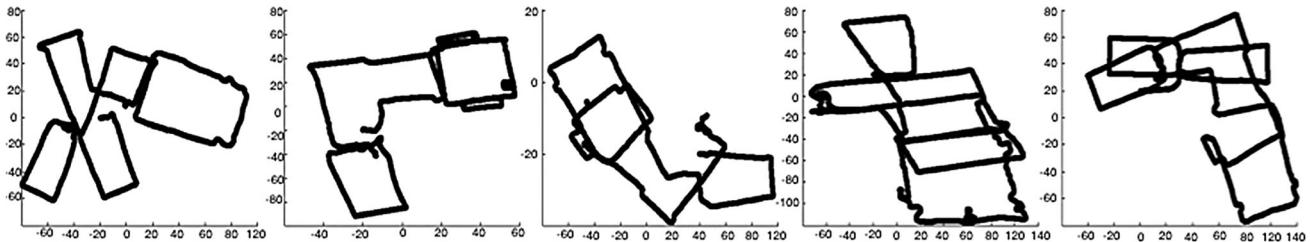


Fig. 8 The path taken by the robot through a pair of buildings at the University Milano-Bicocca in different environmental conditions. From left to right: dynamic with natural lighting, static with artificial

lighting, dynamic with natural lighting, dynamic with artificial lighting and static with natural lighting

operation of the online learning algorithm, while topological integrity is important for path planning and navigation. It has been configured to three channels that each obtains its input from the various datasets obtained from each learning step. The procedures of map learning experiments are as follows:

- Start program and configure number of channels of TGARAM
- Configure TGARAM parameters: α , γ and ρ
- Input sensor datasets based on the number of channels of TGARAM, for example, if TGARAM configured to 3 channels, 3 different sensor datasets have to input to the TGARAM algorithm for learning purpose
- Start TGARAM learning and map building process
- Each map learning iteration, TGARAM, obtains every timeframe (t) sensors data continuously
- Next, TGARAM compares the (t) timeframe sensors data with the ($t - 1$) timeframe sensors data to decide to add nodes, update nodes or remove nodes
- Map building process is stopped when all timeframe sensors data are processed.

3.4 Parameters

There are two important TGARAM parameters that affect topological map construction. First, the initial standard deviation, γ , determines the isotropic spread in feature space of a new node's distribution about its first input, whereas the vigilance parameter ρ controls the matching and compression process. A large value of ρ causes the TGARAM to be less resistant to noise and vice versa. Further discussion of these two parameters can be found in Sect. 4.

In our experiments, we follow the γ parameter setting stated in Williamson's paper [34] by setting $\gamma = 2$ for fast learning and $\gamma = 4$ for slow learning. In addition, according to Carpenter [5], $\rho = 0$ is a baseline to maximize the matching and compression process and $\rho = 0.9$ is a baseline to minimize the matching and compression process. Table 4 shows the parameter setting for our experiments.

3.5 Results: indoor datasets

In the first experiment the odometry dataset and the pre-processed front and rear laser scanner datasets were used as inputs. Figure 8 shows the exact path the robot took in different environmental conditions (dynamic with natural lighting, static with artificial lighting, dynamic with artificial lighting and static with natural lighting).

Figure 9a, b shows the topological map quantization error for fast learning ($\gamma = 2$) and slow learning ($\gamma = 4$). Result shows that quantization error is lower if using higher vigilance parameter (ρ) for map learning.

Next Fig. 10a, b shows the compression ratio of topological map with respect to original robot navigation map. Result showed that the maximum compression ratio can be up to 60:1 with the effect of topological map maintenance algorithm. In addition, compression ratio of the topological map for fast learning is slightly lower than slow learning because fast learning tends to create more topological nodes for representing the environment.

Figure 11 illustrates one of the fast learning and slow learning topological map result with $\rho = 0.9$. The dataset is

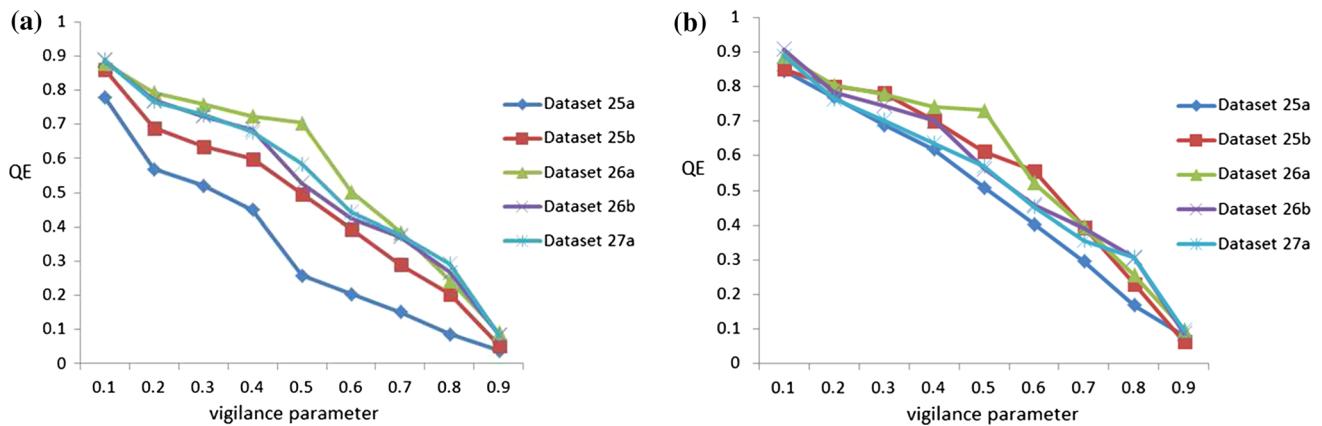


Fig. 9 Quantization error experiment result for indoor dataset. **a** Fast learning, **b** slow learning

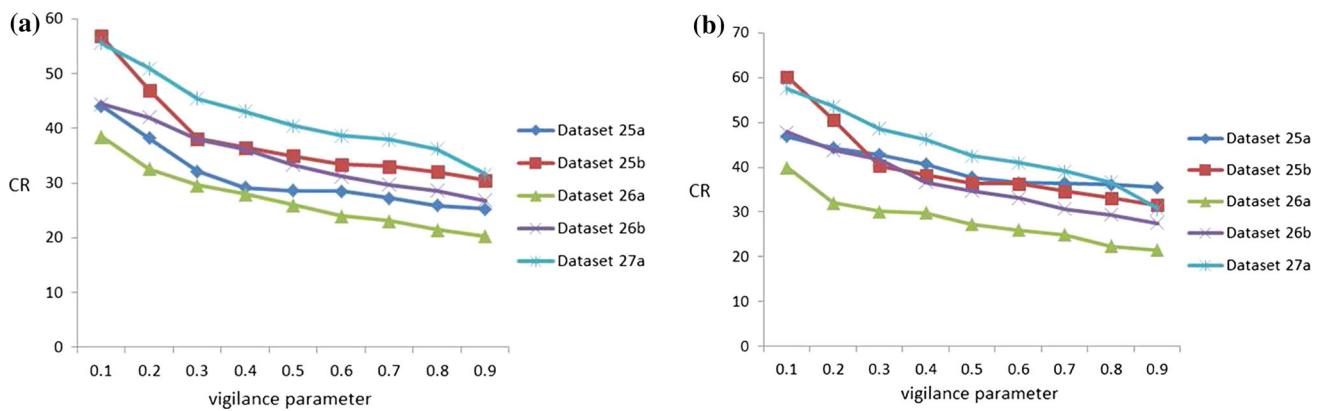


Fig. 10 Compression ratio experiment result for indoor dataset. **a** Fast learning, **b** slow learning

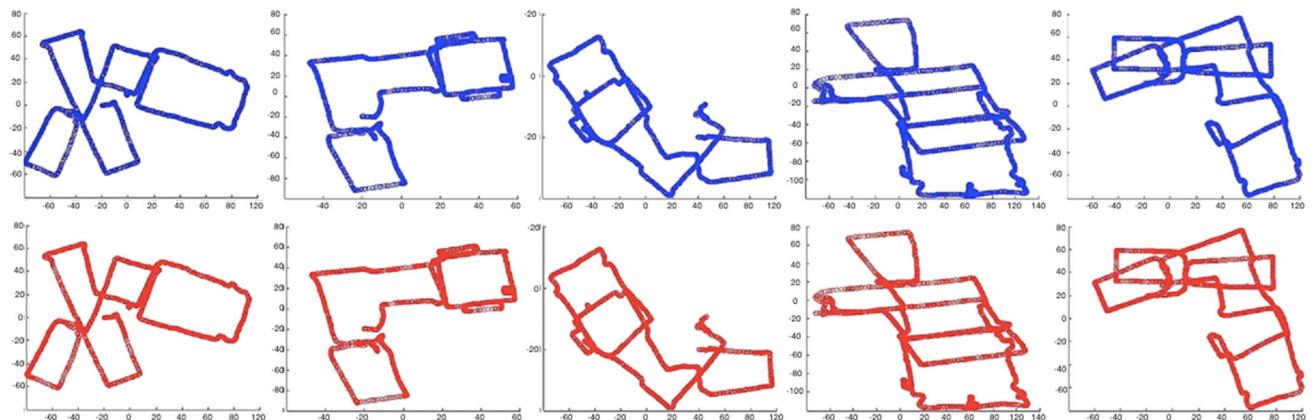


Fig. 11 The topological map constructed by TGARAM using the laser scanner and odometry dataset in the different environmental conditions corresponding to Fig. 12. *1st Row* fast learning results with $\rho = 0.9$. *2nd Row* slow learning results with $\rho = 0.9$

gathered from a large indoor environment; therefore, the built topological maps contain many nodes and difficult to show in full scale.

Figure 12 illustrates one example for the maps connectivity and concept of TGARAM map building. Figure 12a

shows the robot full travel path and 4 laser scanner data gathered when robot traveled from location A to B. Figure 12b shows topological map, and Fig. 12c shows TGARAM learns the environment by adding one node and stores one landmark for representation.

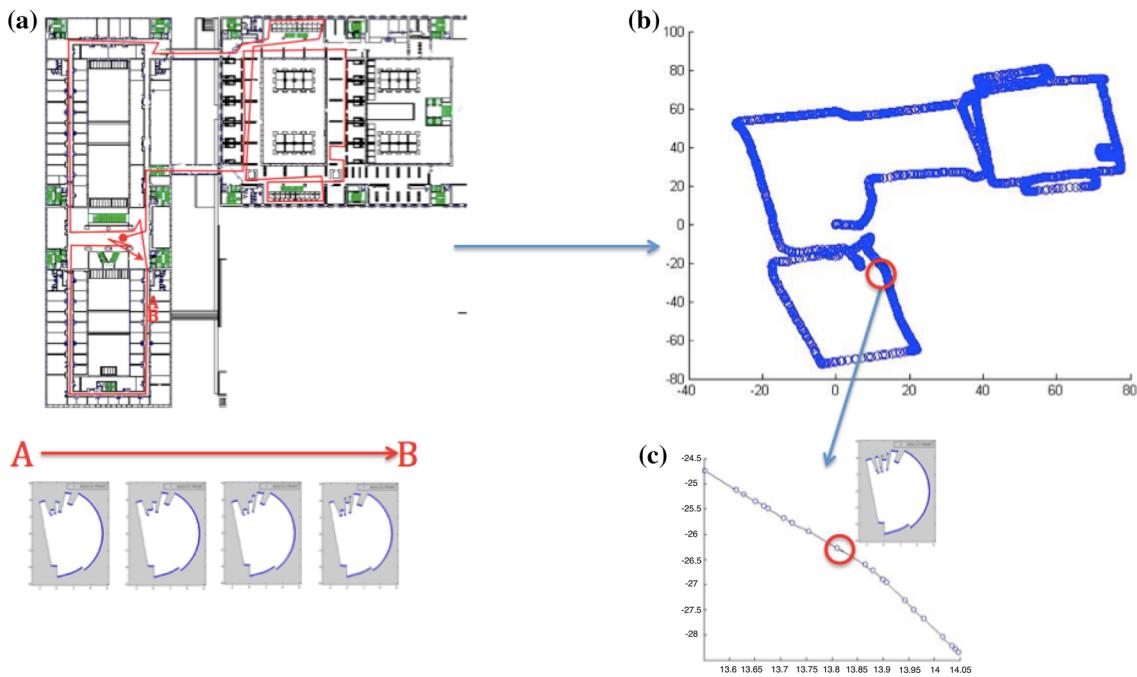


Fig. 12 **a** Original robot path; **b** topological map, and **c** partially enlarged topological map

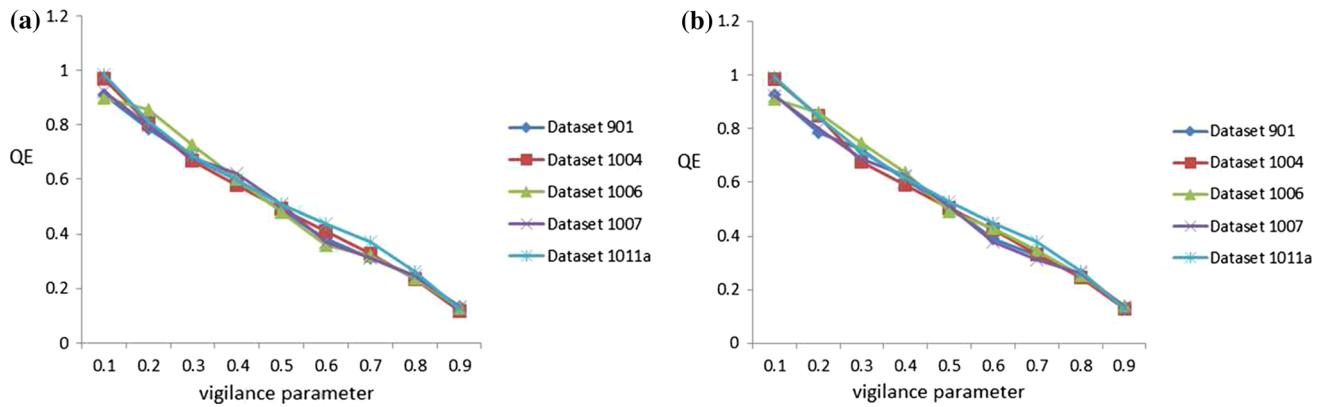


Fig. 13 Quantization error experiment result for outdoor dataset. **a** Fast learning, **b** slow learning

3.6 Results: outdoor datasets

Like experiments on indoor datasets, the same three channels were configured. The parameters setting is as shown in Table 4 in Sect. 3.4. Figure 13a, b shows the topological map quantization error for fast learning ($\gamma = 2$) and slow learning ($\gamma = 4$). Result shows that quantization error is lower if using higher vigilance parameter (ρ) for outdoor datasets map learning.

Next Fig. 14a, b shows the compression ratio of topological map with respect to original robot navigation map from outdoor datasets. Result showed that the maximum compression ratio can be up to 50:1 because outdoor environment and navigation path are more complex than indoor environment.

In addition, compression ratio of the topological map for fast learning is almost similar to slow learning. Figure 15 illustrates the topological maps produced by the TGARAM with fast learning and slow learning.

3.7 Results: map maintenance analysis

In this section, we extract parts of learning iterations to show the node growing patterns with and without map maintenance as shown in Figs. 16 and 17. With map maintenance feature, node pruning is triggered every learning iteration if Eq. 9 is less than e_{\max} . Thus, causing the topological map with fewer nodes requires less memory to store the map, which is crucial for large environment navigation.

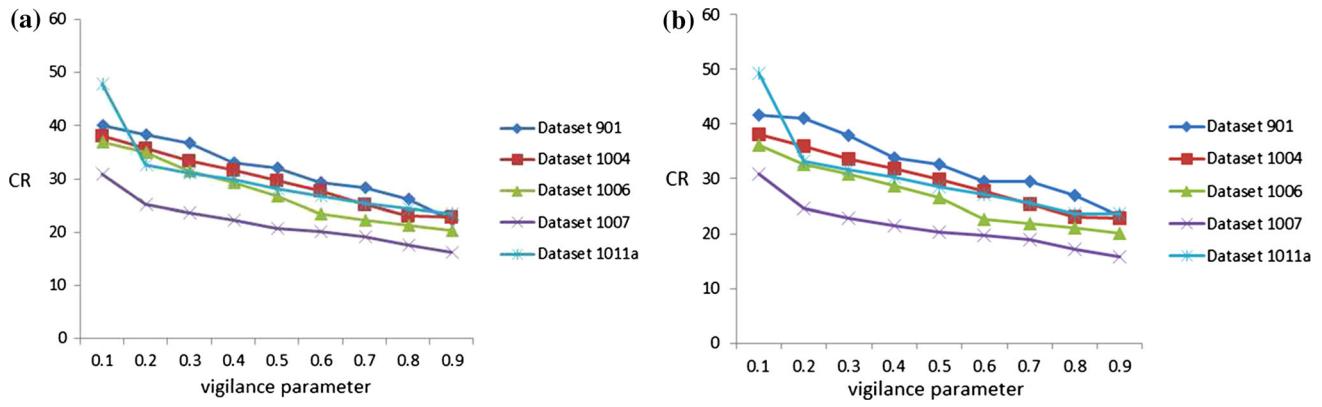


Fig. 14 Compression ratio experiment result for outdoor dataset. **a** Fast learning, **b** slow learning

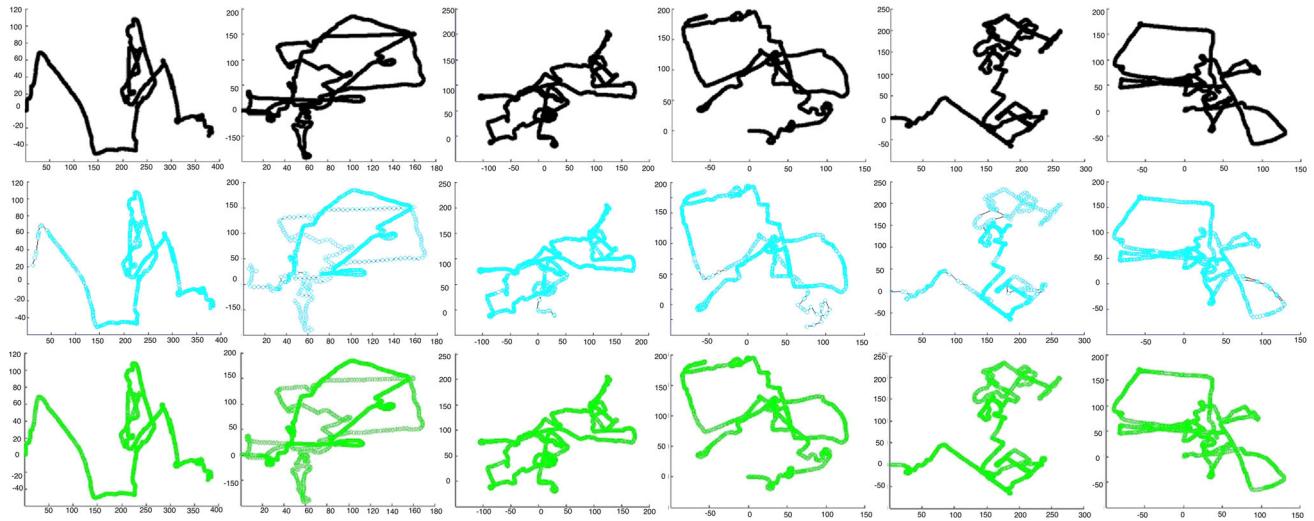


Fig. 15 1st Row actual robots navigation path when gathering (from left to right) Bovisa_2008-10-04_Static datasets, Bovisa_2008-10-07_Dynamic datasets, Bovisa_2008-10-11b_Static datasets, Bovisa_2008-09-01_Static datasets, Bovisa_2008-10-11a_Static

datasets and Bovisa_2008-10-06_Dynamic datasets. 2nd Row fast learning topological map using the datasets mentioned above with setting $\rho = 0.9$. 3rd Row slow learning topological map generated with setting $\rho = 0.9$

3.8 Results: beta oscillation analysis

Figure 18 shows the beta power during rodent traverse in a new environment. Next, Figs. 19 and 20 show the number of mismatch during the topological map learning, respectively. Table 5 shows relationship between the beta oscillations in rodent place cell learning and number of mismatch in TGARAM map learning, which further validate the explanation in Sect. 2.1.

3.9 Results: node localization

In our experiment, we input the same datasets (odometry dataset and Hokuyo laser scanner datasets) again to the corresponding topological map and use Eq. 3 to determine

the best matching node for localization. Equations 22 and 23 are the definition of success and fail localization, respectively.

Success localizationn

$$= \sqrt{(x_{\text{node}} - x_{\text{target}})^2 + (y_{\text{node}} - y_{\text{target}})^2} < 0.1 \quad (22)$$

$$\text{Fail localizationn} = \sqrt{(x_{\text{node}} - x_{\text{target}})^2 + (y_{\text{node}} - y_{\text{target}})^2} > 0.1 \quad (23)$$

Figures 21a, b, and 22a, b illustrate one of the localization results with two nodes having similar sensory information. It is clear that the node B's matching value is much lower than node A because ideothetic information

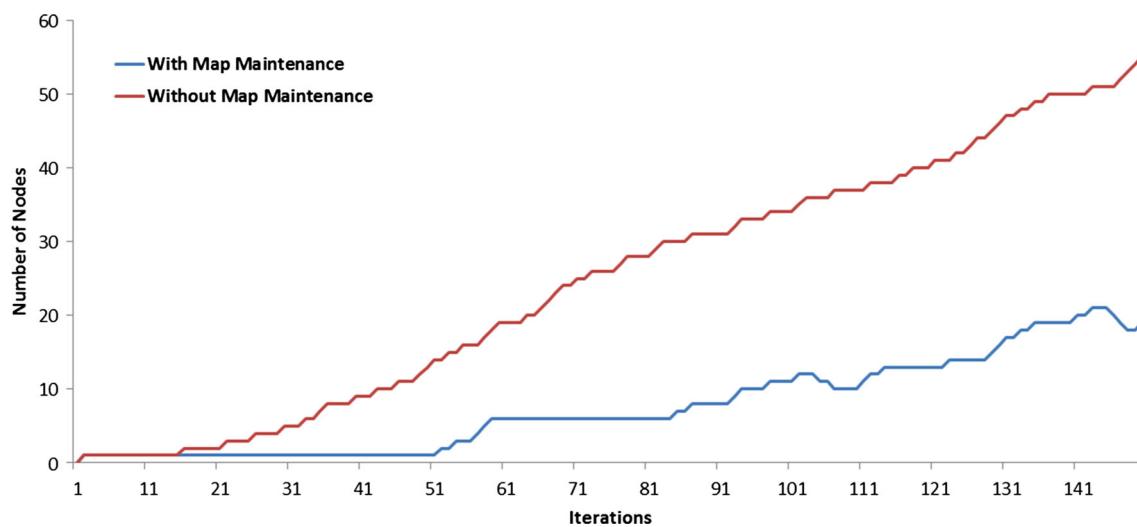


Fig. 16 Nodes growing with and without map maintenance for dynamic, natural lighting dataset

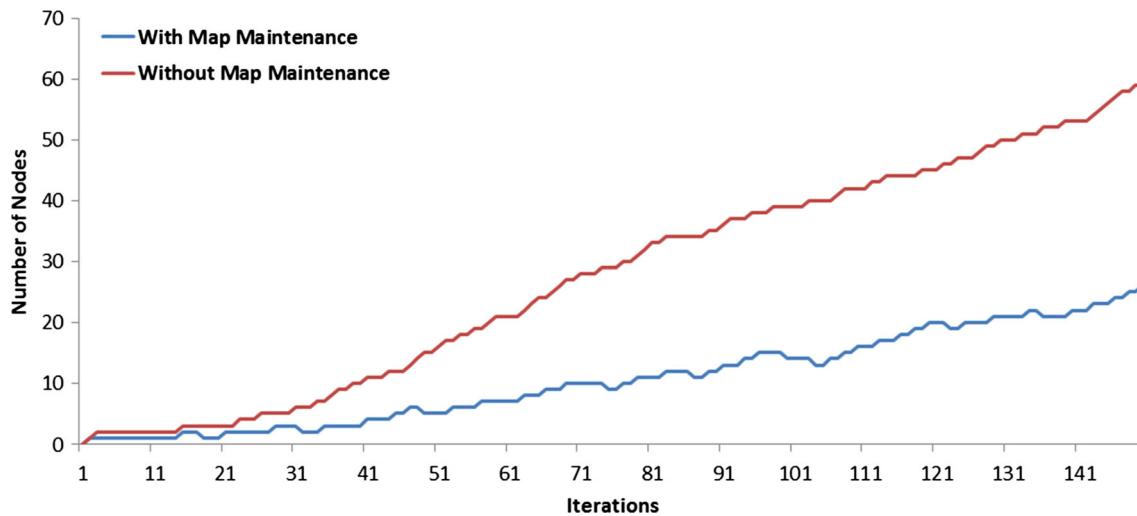


Fig. 17 Nodes growing with and without map maintenance for Bovisa_2008-10-07_Dynamic dataset

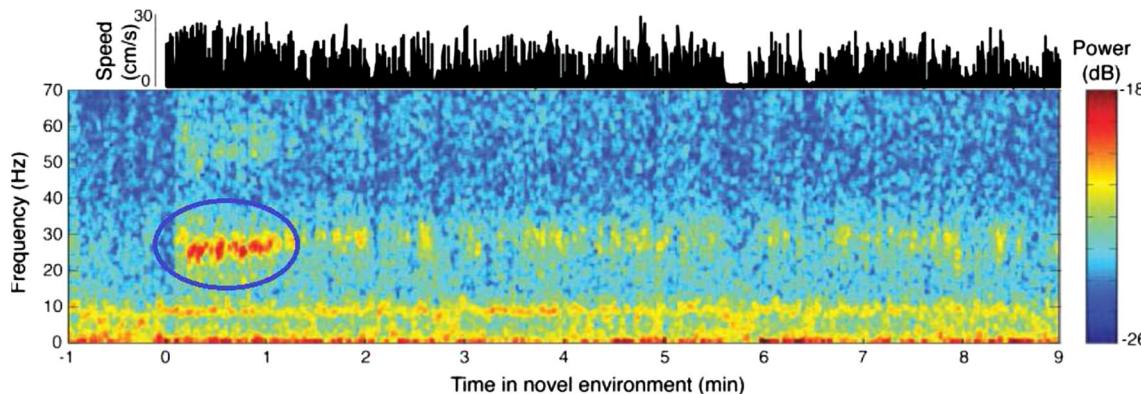


Fig. 18 Beta power (red color) was every low during the first lap of exploration, grew to full strength as a mouse traversed a lap for the second and third times; became low again after the first two minutes of exploration [6] (color figure online)

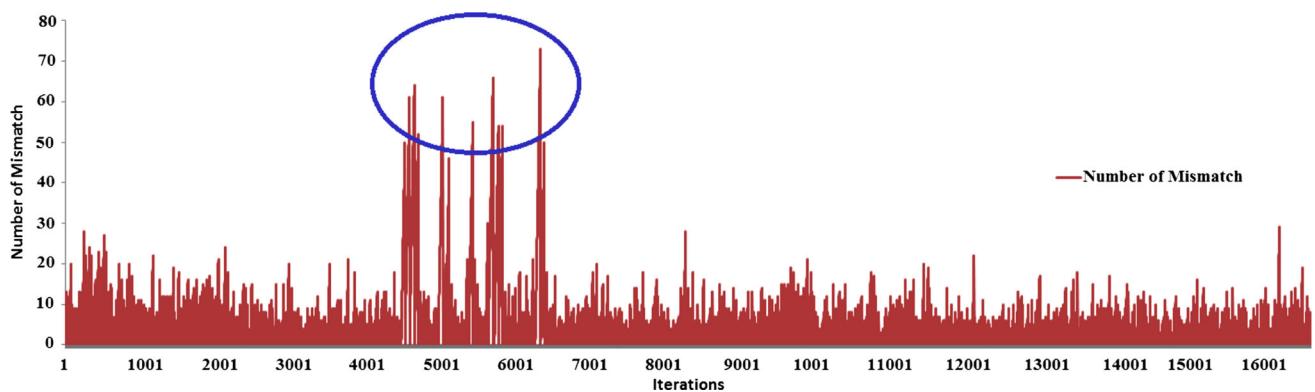


Fig. 19 Number of mismatch during topological map learning for dynamic, natural lighting datasets

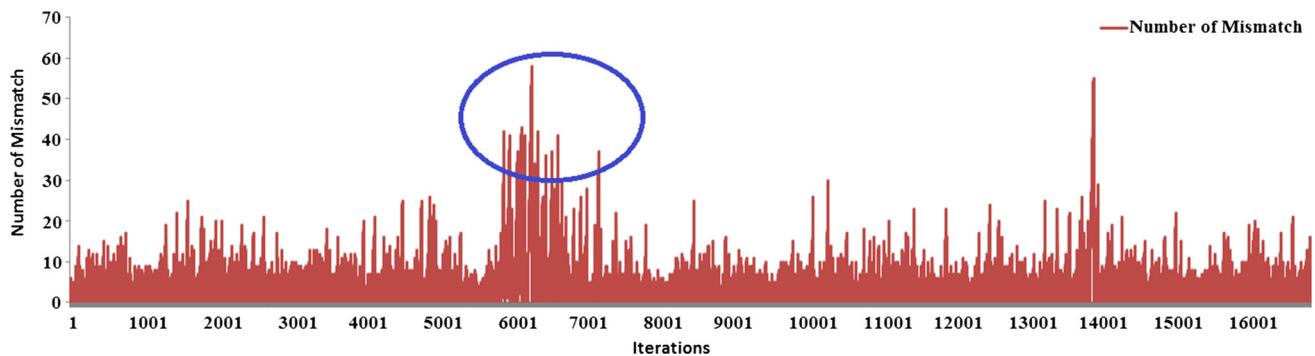


Fig. 20 Number of mismatch during topological map learning for Bovisa_2008-10-06_Dynamic datasets

Table 5 Relationship between the principle of TGARAM and place cell learning

Beta oscillations	TGARAM mismatch state
Beta power is low for the first trial	Number of mismatch is low for the first few thousand iterations because the nodes top-down adaptive weights start out large, so they can match with any input and less mismatch
Beta power grows high at first two minutes for the second and third times	Number of mismatch grows high during 4500 to 7000 iterations, nodes adaptive weight and its critical feature pattern will be refined by updating the Gaussian distribution and, that select new nodes for learning, will cause mismatches
Beta power becomes low again after first two minutes exploration	Number of mismatch remains low after 7000 iterations. Nodes learning is stable, less mismatch

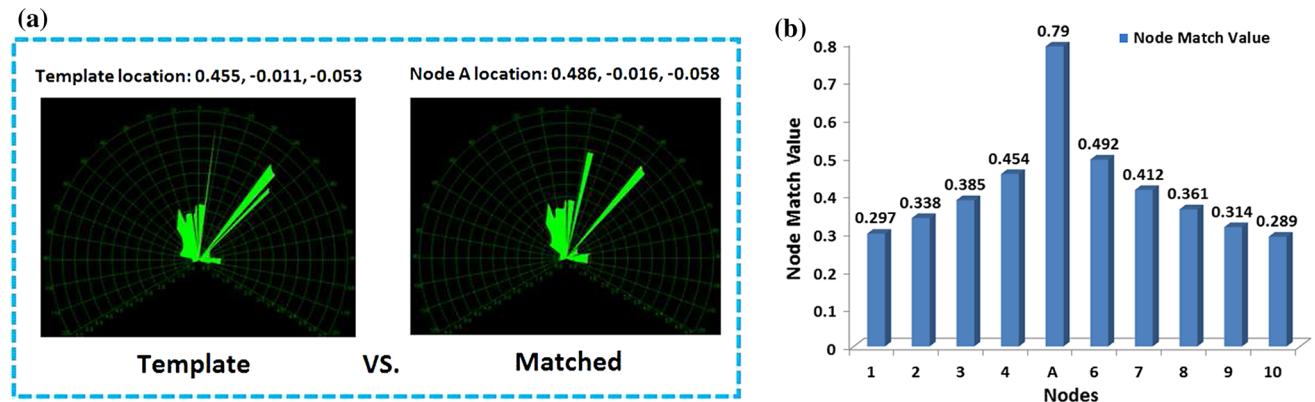


Fig. 21 Template and matching result for node A. **a** Laser data, **b** nodes match value

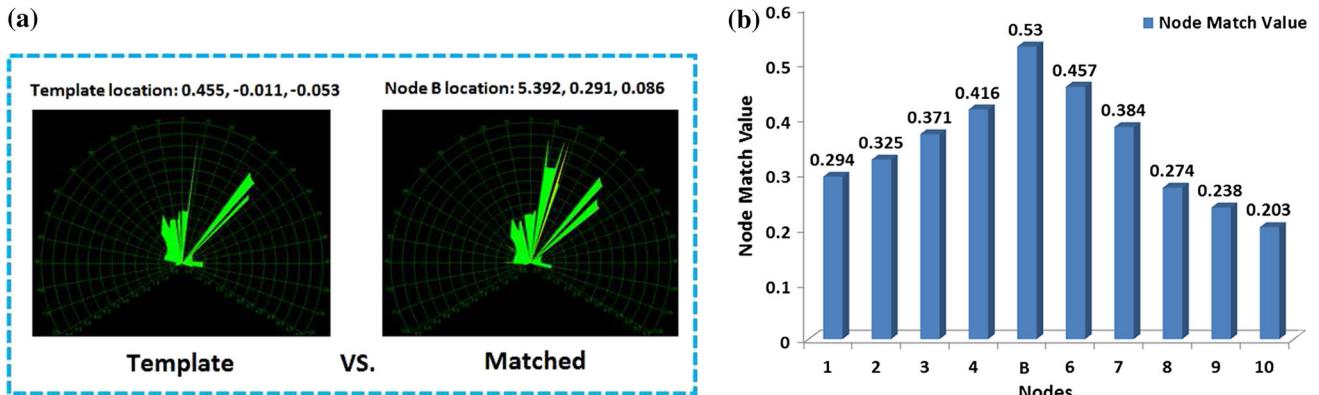


Fig. 22 Template and matching result for node B. **a** Laser data, **b** nodes match value

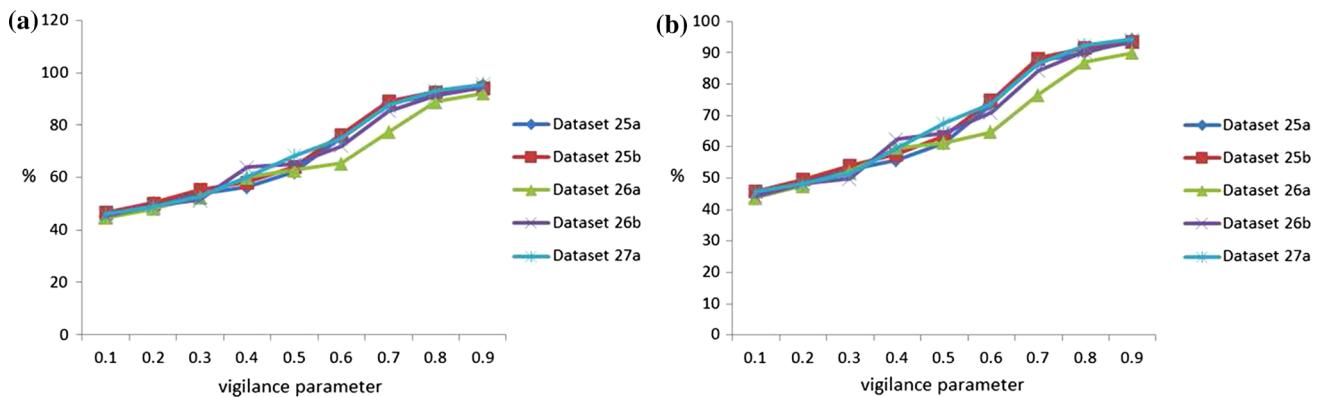


Fig. 23 Localization result for indoor dataset. **a** Fast learning, **b** slow learning

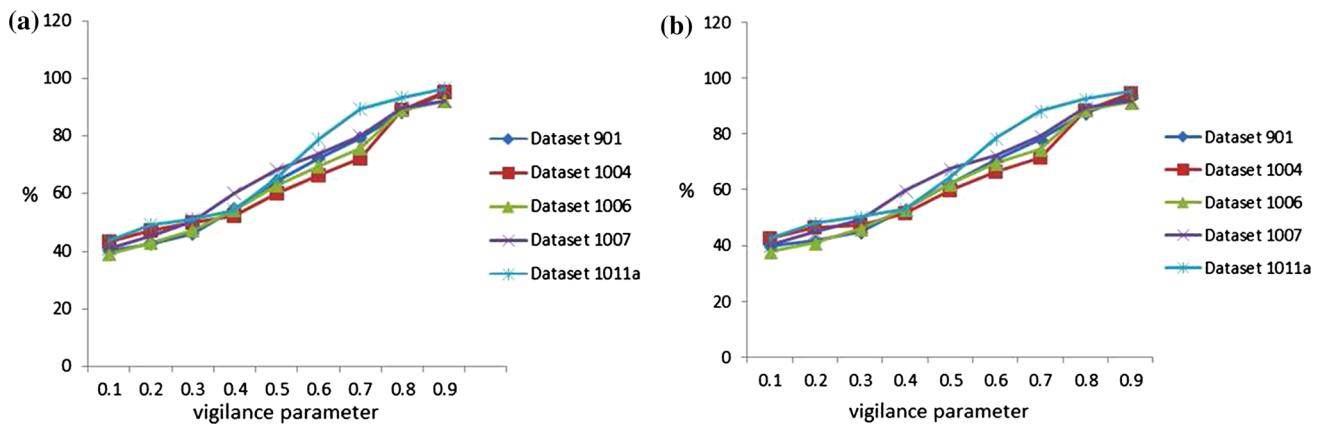


Fig. 24 Localization result for outdoor dataset. **a** Fast learning, **b** slow learning

helps to disambiguate similar sensory information. Therefore, node A will be chosen for localization.

Figures 23a, b and 24a, b show the localization success rate for fast learning and slow learning of TGARAM for both indoor and outdoor datasets. Result shows that using small value of ρ can maximize the compression rate but decrease localization success rate and vice versa.

4 Discussion

We have shown that the TGARAM algorithm we developed is able to construct a topological map from scratch using nearly unprocessed sensory information about the environment. The topological map is driven by Gaussian ART modules, which enable place cell learning, as

explained in previous sections. The Gaussian ART module incrementally produces nodes from the remembered Gaussian elements by measuring their perceptual similarity with the robot's current sensory information.

The TGARAM training process not only takes into account sensor measurements but also the odometry of the robot. Therefore, TGARAM is able to disambiguate locations where the sensory information is very similar, which overcomes problem of online detection and recognition of topological nodes. The TGARAM algorithm relies on access to comprehensive sensory information in order to generate and maintain the topological map. Missing information as well as irrelevant information would significantly degrade the reliability of the map.

The constructed topological map provides the flexibility and maintainability required for robot navigation. New nodes are constantly added as the environment is learned, and they are linked to existing nodes by edges. These edges provide the information necessary in order to travel from one location to another. In the previous section, we showed that all the topological maps generated by TGARAM are almost identical to the original path taken by the robot and experiment in Sect. 3.9 showed that the topological map can be used for node localization, which means that the topological map is suitable for robot localization and navigation.

In addition, map resolution depends on the vigilance parameter (ρ). With a small value of ρ , minor changes in sensory information are detected by TGARAM as novelty and nodes are added to the map. In this case, the map contains a lot of unnecessary information about the environment. On the other hand, if the ρ value is set too high, TGARAM only adds nodes to the map when there is a distinct difference in sensory information. Thus, the map might miss important information and be insufficient to represent the environment. As a result, both situations cause the robot to fail to adequately perform localization.

In this paper, we proposed a pruning method based on Thales sphere radius to manage the number of nodes. The advantage of this pruning system is that it is not affected by the dynamics of objects or the timescale of changes. If the robot travels around an area of the map that has been pruned, it will automatically learn and integrate new nodes, which encode changes of the environment and nodes that fall within the Thales sphere radius will be deleted to ensure that the map remains up to date. Therefore, the pruning algorithm overcomes the nodes proliferation problem for map building. This approach also reduces the amount of memory needed to memorize the map, especially when the area is very large. However, the drawback of the pruning system is that it cannot efficiently handle rapid geometric changes in the environment.

5 Conclusion

In summary, we proposed a biologically inspired learning method for topological map building. It is an incremental and unsupervised learning method that overcomes the stability-plasticity dilemma, which explains how the brain can quickly learn to categorize information in the real world and to remember it without forgetting previously learned knowledge. Our proposed method can simulate beta oscillations and hippocampal place cell learning, which overcomes the problem of online detection and recognition. Thus, it is able to produce a consistent and stable topological map in natural environments. The combination of our method with IPCA was validated by benchmarking on Rawseeds datasets. These high-quality datasets represented different environmental conditions. Our experimental results proved that our proposed method is capable of processing more than one sensory source for map learning, which reduces error measurement effects caused by uncertain sensor data. The topological map contains sensor features, robot location and traversability information, which enables the robot to update its self-location without human intervention.

In future work, we plan use other sensor data such as image to improve the map reliability and usability. Lastly, the TGARAM should be implemented in a real robot for further validation.

Acknowledgments The authors would like to acknowledge a scholarship provided by the University of Malaya (Fellowship Scheme). This research is supported in part by HIR grant UM.C/625/1/HIR/MOHE/FCSIT/10 from the University of Malaya.

References

- Ball D, Heath S, Wiles J, Wyeth G, Corke P, Milford M (2013) OpenRatSLAM: an open source brain-based slam system. *Autonom Robots* 3:149–176. doi:[10.1007/s10514-012-9317-9](https://doi.org/10.1007/s10514-012-9317-9)
- Barrera A, Weitzenfeld A (2008) Biologically-inspired robot spatial cognition based on rat neurophysiological studies. *Autonom Robots* 25(1–2):147–169. doi:[10.1007/s10514-007-9074-3](https://doi.org/10.1007/s10514-007-9074-3)
- Benjamin K, Yung-Tai B (1991) A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *J Robot Autonom Syst* 8:47–63
- Berke J, Hetrick V, Breck J, Greene R (2008) Transient 23–30 hz oscillations in mouse hippocampus during exploration of novel environments. *Hippocampus* 18(5):519–29. doi:[10.1002/hipo.20435](https://doi.org/10.1002/hipo.20435)
- Carpenter GA (2003) Default aromap. In: Proceedings of the international joint conference on neural networks (IJCNN03), pp 1396–1401
- Carpenter GA, Grossberg S (1987) A massively parallel architecture for a self-organizing neural pattern recognition machine. *Comput Vis Graph Image Process*. doi:[10.1016/S0734-189X\(87\)80014-2](https://doi.org/10.1016/S0734-189X(87)80014-2)

7. Ceriani S, Fontana G, Giusti A (2009) Rawseeds ground truth collection systems for indoor self-localization and mapping. In: Autonomous robots. Kluwer, Hingham, MA, USA, vol 27, pp 353–371. doi:[10.1007/s10514-009-9156-5](https://doi.org/10.1007/s10514-009-9156-5)
8. Chandrasekaran S, Manjunath BS, Wang YF, Winkeler J, Zhang H (1996) An eigenspace update algorithm for image analysis. Technical report, Santa Barbara, CA, USA
9. Chang H, Lee CSG, Hu Y, Lu YH (2007) Multi-robot slam with topological/metric maps. In: IEEE/RSJ international conference on intelligent robots and systems, 2007, IROS 2007, pp 1467–1472. doi:[10.1109/IROS.2007.4399142](https://doi.org/10.1109/IROS.2007.4399142)
10. Chang-Hyuk C, Jae-Bok S, Woojin C (2002) Topological map building based on thinning and its application to localization. In: IEEE/RSJ international conference on intelligent robots and systems, 2002, vol 1, pp 552–557. doi:[10.1109/IRDS.2002.1041448](https://doi.org/10.1109/IRDS.2002.1041448)
11. Chatila R, Laumond J (1985) Position referencing and consistent world modeling for mobile robots. In: Proceedings 1985 IEEE international conference on robotics and automation, vol 2, pp 138–145. doi:[10.1109/ROBOT.1985.1087373](https://doi.org/10.1109/ROBOT.1985.1087373)
12. David Filliat JA (2003) Map-based navigation in mobile robots: I. A review of localization strategies. *Cogn Syst Res* 4(4):243–282. doi:[10.1016/S1389-0417\(03\)00008-1](https://doi.org/10.1016/S1389-0417(03)00008-1)
13. Fontana G, Matteucci M, Sorrenti D (2014) Rawseeds: building a benchmarking toolkit for autonomous robotics. In: Amigoni F, Schiaffonati V (eds) Methods and experimental techniques in computer engineering. Springer International Publishing, SpringerBriefs in Applied Sciences and Technology, Berlin, pp 55–68
14. Ghahramani Z, Hinton GE (2000) Variational learning for switching state-space models. *Neural Comput* 12(4):831–864. doi:[10.1162/089976600300015619](https://doi.org/10.1162/089976600300015619)
15. Giovannangeli C, Gaussier P (2008) Autonomous vision-based navigation: goal-oriented action planning by transient states prediction, cognitive map building, and sensory-motor learning. In: IEEE/RSJ international conference on intelligent robots and systems, 2008, IROS 2008, pp 676–683. doi:[10.1109/IROS.2008.4650872](https://doi.org/10.1109/IROS.2008.4650872)
16. Grossberg S (2009) Beta oscillations and hippocampal place cell learning during exploration of novel environments. *Hippocampus* 19(9):881–885. doi:[10.1002/hipo.20602](https://doi.org/10.1002/hipo.20602)
17. Grossberg S, Versace M (2008) Spikes, synchrony, and attentive learning by laminar thalamocortical circuits. *Brain Res*. doi:[10.1016/j.brainres.2008.04.024](https://doi.org/10.1016/j.brainres.2008.04.024)
18. Hall P, Marshall D, Martin R (1998a) Merging and splitting eigenspace models. *IEEE Trans Pattern Anal Mach Intell* 22:2000
19. Hall PM, Marshall D, Martin RR (1998b) Incremental eigen-analysis for classification. In: British machine vision conference, pp 286–295
20. Jean-Arcady M, Agns G, Benot G, Mehdi K, Patrick P, Alain B (2005) The Psikharpx project: towards building an artificial rat. *Robot Autonom Syst* 50(4):211–223. doi:[10.1016/j.robot.2004.09.018](https://doi.org/10.1016/j.robot.2004.09.018)
21. Jockusch J, Ritter H (1999) An instantaneous topological mapping model for correlated stimuli. In: International joint conference on neural networks, 1999, IJCNN'99, vol 1, pp 529–534. doi:[10.1109/IJCNN.1999.831553](https://doi.org/10.1109/IJCNN.1999.831553)
22. Kwon TB, Song JB (2008) Thinning-based topological exploration using position possibility of topological nodes. *Adv Robot* 22(2–3):339–359. doi:[10.1163/156855308X292619](https://doi.org/10.1163/156855308X292619)
23. Leivas G, Botelho S, Drews P, Figueiredo M, Haffele C (2010) Sensor fusion based on multi-self-organizing maps for slam. In: 2010 IEEE conference on multisensor fusion and integration for intelligent systems (MFI). doi:[10.1109/MFI.2010.5604482](https://doi.org/10.1109/MFI.2010.5604482)
24. Levey A, Lindenbaum M (2000) Sequential Karhunen–Loeve basis extraction and its application to images. *IEEE Trans Image Process* 9(8):1371–1374. doi:[10.1109/83.855432](https://doi.org/10.1109/83.855432)
25. McGlinchey S, Peña M, Fyfe C (2006) Quantization errors in the harmonic topographic mapping. In: Proceedings of the 5th WSEAS international conference on signal processing, world scientific and engineering academy and society (WSEAS), Stevens Point, Wisconsin, USA, SIP'06, pp 105–110. <http://dl.acm.org/citation.cfm?id=1983937.1983961>
26. Milford M, Wyeth G, Prasser D (2004) RatSLAM: a hippocampal model for simultaneous localization and mapping. In: Proceedings of the international robotics and automation (ICRA'04), vol 1, pp 403–408. doi:[10.1109/ROBOT.2004.1307183](https://doi.org/10.1109/ROBOT.2004.1307183)
27. O’Keefe J, Dostrovsky J (1971) The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res* 34(1):171–175
28. Shi C, Wang Y, Yang J (2010) Online topological map building and qualitative localization in large-scale environment. *Robot Auton Syst* 58(5):488–496. doi:[10.1016/j.robot.2010.01.009](https://doi.org/10.1016/j.robot.2010.01.009)
29. Tan AH (1995) Adaptive resonance associative map. *Neural Netw* 8(3):437–446. doi:[10.1016/0893-6080\(94\)00092-Z](https://doi.org/10.1016/0893-6080(94)00092-Z)
30. Tarutoko Y, Kobayashi K, Watanabe K (2006) Topological map generation based on Delaunay triangulation for mobile robot. In: International joint conference on SICE-ICASE, 2006, pp 492–496. doi:[10.1109/SICE.2006.315477](https://doi.org/10.1109/SICE.2006.315477)
31. Thrun S (1998) Learning metric-topological maps for indoor mobile robot navigation. *Artif Intell* 99(1):21–71. doi:[10.1016/S0004-3702\(97\)00078-7](https://doi.org/10.1016/S0004-3702(97)00078-7)
32. Tomatis N, Nourbakhsh IR, Siegwart R (2003) Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robot Autonom Syst* 44(1):3–14. doi:[10.1016/S0921-8890\(03\)00006-X](https://doi.org/10.1016/S0921-8890(03)00006-X)
33. Van Zwijnsvoorde D, Simeon T, Alami R (2000) Incremental topological modeling using local voronoi-like graphs. In: Proceedings 2000 IEEE/RSJ international conference on intelligent robots and systems, 2000 (IROS 2000), vol 2, pp 897–902. doi:[10.1109/IROS.2000.893133](https://doi.org/10.1109/IROS.2000.893133)
34. Williamson JR (1996) Gaussian ARTMAP: a neural network for fast incremental learning of noisy multidimensional maps. *Neural Netw* 9:881–897. doi:[10.1016/0893-6080\(95\)00115-8](https://doi.org/10.1016/0893-6080(95)00115-8)
35. Winkeler J, Manjunath B, Chandrasekaran S (1999) Subset selection for active object recognition. In: IEEE computer society conference on computer vision and pattern recognition, 1999, vol 2, pp –516. doi:[10.1109/CVPR.1999.784729](https://doi.org/10.1109/CVPR.1999.784729)

Neural Computing & Applications is a copyright of Springer, 2018. All Rights Reserved.