Applied Soft Computing 38 (2016) 269–280

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Multi-channel Bayesian Adaptive Resonance Associate Memory for on-line topological map building

Wei Hong Chin[a], Chu Kiong Loo[a], Manjeevan Seera[b,*], Naoyuki Kubota[c], Yuichiro Toda[c]

[a] Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia
[b] Faculty of Engineering, Computing and Science, Swinburne University of Technology (Sarawak Campus), Malaysia
[c] Faculty of System Design, Tokyo Metropolitan University, Tokyo 191-0065, Japan

ARTICLE INFO

Article history:
Received 6 June 2015
Received in revised form
25 September 2015
Accepted 25 September 2015
Available online 22 October 2015

Keywords:
Adaptive resonance theory
Bayesian
Robot navigation
Simultaneous localization and mapping
Topological map

ABSTRACT

In this paper, a new network is proposed for automated recognition and classification of the environment information into regions, or nodes. Information is utilized in learning the topological map of an environment. The architecture is based upon a multi-channel Adaptive Resonance Associative Memory (ARAM) that comprises of two layers, input and memory. The input layer is formed using the Multiple Bayesian Adaptive Resonance Theory, which collects sensory data and incrementally clusters the obtained information into a set of nodes. In the memory layer, the clustered information is used as a topological map, where nodes are connected with edges. Nodes in the topological map represent regions of the environment and stores the robot location, while edges connect nodes and stores the robot orientation or direction. The proposed method, a Multi-channel Bayesian Adaptive Resonance Associative Memory (MBARAM) is validated using a number of benchmark datasets. Experimental results indicate that MBARAM is capable of generating topological map online and the map can be used for localization.

1. Introduction

With growth in robotics research, it is possible for autonomous robots to move in complex, unknown environments and execute assigned tasks with little or no human intervention. A human-friendly autonomous guided robot with little knowledge about its whereabouts can navigate safely within its environment in order to achieve certain goals [1]. It can construct a map of the environment based on its position and posture (mapping), and estimating its position and posture using a built environment map (localization). Building the representation of the map is crucial to autonomous navigation for improving map maintenance, map-based localization, and path planning in any given environment.

In mobile robotics, representations of the world fall into three broad groups, i.e. metric maps [2], topological maps [3], and hybrid models [4,5] that combine both metric and topological information. 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 [6]. In the grid occupancy approach, the environment is mapped as an array of cells. This approach however requires complex computation for feature matching which is not reliable for large environments. Pure metric map methods are vulnerable to inaccuracies in both map building and robot position estimation [7].

In the topological framework, the environment is described by a collection of places linked by path [8]. Places are defined by information gathered from sensors placed in the environment, which is then stored in nodes. Some of the robot's odometry information, which is 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.

One of the advantages of topological maps is they do not require a metric sensor model to convert sensor data into a 2D frame of reference [9]. However, topological maps lack the details found in an environment. To overcome these problems, hybrid approaches [10,11] that combine the topological and metric paradigms have been proposed to compensate for the weakness of each approach. A crucial aspect that restricts the practice of topological maps is the absence of consistence semantics associated with them. For example in [8], nodes are represented as places distinguished by sensor

[*] Corresponding author. Tel.: +60 82 260 958.
E-mail addresses: weihong118118@gmail.com (W.H. Chin), ckloo.um@um.edu.my (C.K. Loo), mseera@gmail.com (M. Seera), kubota@tmu.ac.jp (N. Kubota), toda-yuuichirou@ed.tmu.ac.jp (Y. Toda).

http://dx.doi.org/10.1016/j.asoc.2015.09.031
1568-4946/© 2015 Elsevier B.V. All rights reserved.

data, and edges as paths between places distinguished by control strategies. On the other hand, maps are constructed by partitioning a probabilistic occupancy grid into regions isolated by narrow paths based on a measure of local clearance [12]. In [13], grid map is transformed into a greyscale image and analyzed for retrieving the topological information. This method however requires massive memory allocation for the grid map storing. As an alternative, the generation of a straightforward topological map using the generalized Voronoi graph (GVG) is presented in [13,14].

GVG is however susceptible in dynamic, large-scale environments with sharp-edged obstacles and it requires high computational resources for node extraction and matching. A thinning approach to construct a topological map from a binary grid map is proposed in [15]. Although this method requires less computational resources than the GVG method, the method is based on an existing grid map that limits its application in online map building [16]. Another important factor that impedes the application of the topological map is the online detection and recognition of topological nodes. While artificial landmarks such as wireless beacons, visual patterns, or reflective tape provide reliable recognition of a specific location, we have to consider the possibility of the artificial landmarks not available in unknown environments.

Another area of research 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 appears to respond to the animal's spatial location [17]. Some researchers have uncovered that beta oscillations occur during the learning of hippocampal place cell receptive fields in novel environments [18]. The aim of the Psikharpax project is to create an artificial robotic rat, driven by mapping and navigation algorithms that mimic place cells [19]. Other methods for biologically inspired navigation include RatSLAM [20,21], which builds a topological map with metric information by separating the topological and metric layer. This approach requires proper scaling of the map in advance and is unable to update efficiently during operation.

In this paper, we present the Multi-channel Bayesian Adaptive Resonance Associative Memory (MBARAM) for online topological map learning and generation. Each topological node represents a particular region of the environment with location of the robot, while edges connect nodes and stores the robot actions and relative positions of connecting nodes. MBARAM is developed from the Bayesian Adaptive Resonance Theory (Bayesian ART) framework [22]. Despite several optimization algorithms [23–26] being used to solve the SLAM problem, we chose the ART framework for its fast, online, unsupervised learning abilities, and it has been used in explaining place cell learning [27]. In addition, ART addresses the *stability-plasticity* dilemma [28], which explains how the brain can both quickly learn to categorize information in the real-world and remember it, without forgetting previously-learned knowledge. With these advantages, the proposed method overcomes the problem of online detection and recognition of topological nodes. In addition, MBARAM handles and learn multiple sensory information simultaneously during the mapping process. MBARAM does not require a metric map, it performs self-learning, and maintains the topological map that makes it suitable for work in natural environments.

The proposed method has a number of contributions. The first is an incremental and unsupervised learning method that enables map building with little to no human intervention. Next, it does not require high-level cognitive and prior knowledge to make it work in a natural environment. Third, it can process multiple sensory sources simultaneously in continuous space, which is crucial for real-world robot navigation. Finally, the ART architecture overcomes the *stability-plasticity* dilemma without forgetting previously-learned knowledge. This paper is organized as follows. Section 2 introduces the theoretical framework of the proposed online topological map building approach. The experimental setup and results are detailed in Section 3 with discussions in Section 4. Concluding remarks are finally presented in Section 5.

2. Multi-channel Bayesian Adaptive Resonance Associative Memory (MBARAM)

MBARAM integrates the Bayesian ART [22] and an incremental topology-preserving mechanism. Bayesian ART is used as a learning framework for its ability to reduce category proliferation with better classification accuracy. The topology-preserving mechanism was used to construct a topological mapping, in which nodes are connected by edges. In addition, we modified the ARAM network [29] in becoming a multi-channel architecture. This enable it to learn multiple mappings simultaneously across multi-modal feature patterns from multiple sensors.

The architecture of MBARAM consists of two layers, as shown in Fig. 1. The input layer is formed by multiple Bayesian ARTs to categorize multiple sensory sources as place regions (nodes) and transmit to the memory layer. In the memory layer, place regions (node) connects to one another by edge when transitions between regions are experienced. Each particular connection contains robot's orientation and bearing when the transition is happened whereas nodes represent distinct places of the environment.

As mentioned in Section 1, the nature of MBARAM learning process is biologically inspired, but different from other biologically inspired method, such as RatSLAM [20]. The work in [30] used different techniques for online topological map building. Table 1 shows the difference between MBARAM, RatSLAM and work in [30].

2.1. Building a topological map online with MBARAM

MBARAM learns in an environment that is perceived from continuous sensory information provided by multiple sensors. The *M*-dimensional sensory information is transmitted to the Bayesian ART and the map learning undergo three main processes, i.e. node competition, node matching, and node learning. Details of the learning process is detailed in Section 2.1.2.

In order to generate a topological map, a number of nodes are created that corresponds to the number of distinct regions determined by the robot, i.e. one node for each region. MBARAM incrementally produces nodes to represent the environment as multiple regions throughout the navigation. It is therefore possible to identify where the robot has already been through a simple comparison of the current sensory information with the nodes already in the topological map (online detection and recognition).

2.1.1. Node definition

In our topological map, nodes are defined as regions, in which perceptions are similar given the robot's position in the area. This method solves the point-of-view problem. The definition is obtained directly through Bayesian ART categorization of sensory information, the category of a perception corresponding to the place where the robot is positioned. Each node contains a multidimensional Gaussian distribution, with mean vector $\hat{\mu}_j$, covariance matrix $\hat{\Sigma}_j$, and a prior probability $\hat{P}(w_j)$. Table 2 summarizes the content of the node in topological map.

The network is initialized with three parameters: the maximal hypervolume S_{MAX}, the initial covariance matrix $\hat{\Sigma}_{\text{init}}$, and initial prior probability $\hat{P}(w_j)_{init}$. Such node definitions are based solely on the robot's perceptual capacities and does not rely on a human definition of the place it is supposed to be. This makes places easier to recognize from sensory information.

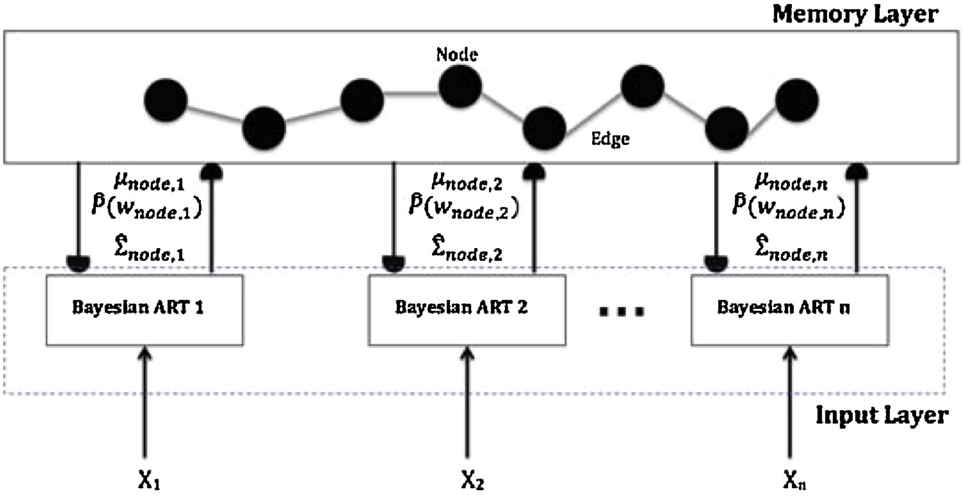


Fig. 1. Multi-channel Bayesian Adaptive Resonance Associative Memory architecture.

Table 1
Comparison between MBARAM, RatSLAM and [30].

	MBARAM	RatSLAM	[30]
Node information	Raw sensory information and robot odometry	Metric information	Opening or intersection between different routes
Map type	Topological map	Topological map and metric map	Topological map
Preprocessing step	Do not require scaling of map in advance and predefined node information	Require proper scaling of map in advance	Predefined node information (doorway, corner of corridor)
Environment	Indoor &outdoor	Indoor &outdoor	Indoor only

Table 2
MBARAM parameters definition.

Parameters	Meaning
$\hat{\mu}_j$	Mean vector – weight parameter
$\hat{\Sigma}_j$	Covariance matrix – weight parameter initialized by initial covariance matrix $\hat{\Sigma}_{\text{init}}$
$\hat{P}(w_j)$	Prior probability and initialized by initial prior probablity $\hat{P}(w_j)_{init}$
S_{MAX}	Maximal hypervolume – vigilance parameter initialized by value falls between optimized range from 10^{-8} to 10^2
N_j	Node learning rate

2.1.2. Map learning process

The learning algorithm grows its network from one node at the first perception. The topological map is continuously updated according to information from the robot's sensors. During learning, the Bayesian ART 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. The winning node, W_J is the node that has the highest match value, G_J and satisfies the vigilance test.

The algorithm consists of three main stages, i.e. node competition, node matching (vigilance test), and node learning.

(1) *Node competition:* In this stage, all existing nodes compete with each other to represent an input pattern. The a posteriori probability of the *j*th node to represent the *M*-dimensional pattern $\boldsymbol{x}$ is calculated as follows:

$$M_j = \hat{P}(w_j|x) = \frac{\hat{p}(x|w_j)\hat{P}(w_j)}{\sum_{l=1}^{N_{cat}} \hat{p}(x|w_l)\hat{P}(w_l)} \tag{1}$$

where N_{cat} is the number of nodes and $\hat{P}(w_j)$ is the estimated prior probability of the *j*th node. The likelihood of w_j with respect to $\boldsymbol{x}$ is estimated using all patterns that have already been associated with the multivariate Gaussian node w_j:

$$\hat{p}(x|w_j) = \frac{1}{(2\pi)^{M/2}|\hat{\Sigma}_j|^{1/2}} \exp\{-0.5(x-\hat{\mu}_j)^T \hat{\Sigma}_j^{-1}(x-\hat{\mu}_j)\} \tag{2}$$

where $\hat{\mu}_j$ and $\hat{\Sigma}_j$ are the estimated mean and covariance matrix of the *j*th node.

If training involves *K* sensory channels, the M_j for each node is:

$$M_j = \sum_{k=1}^{K} \alpha_k [\hat{P}(w_{(j,k)}|x_k)] \tag{3}$$

$$M_j = \sum_{k=1}^{K} \alpha_k \left[\frac{\hat{p}(x_k|w_{j,k})\hat{P}(w_{j,k})}{\sum_{l=1}^{N_{cat}} \hat{p}(x_k|w_{l,k})\hat{P}(w_{l,k})} \right] \tag{4}$$

where

$$\hat{p}(x_k|w_{j,k}) = \frac{1}{(2\pi)^{M/2}|\hat{\Sigma}_{j,k}|^{1/2}} \exp\{-0.5(x_k-\hat{\mu}_{j,k})^T \hat{\Sigma}_{j,k}^{-1}(x_k-\hat{\mu}_{j,k})\} \tag{5}$$

and α_k influence factor for each channel and the sum of α_k is 1.

The winning node $\boldsymbol{J}$ is the one with the maximum a posteriori probability (MAP):

$$J = arg\max(M_j) \tag{6}$$

This proves that the ***J***th node w_J is either more populated than other nodes (i.e. having high $\hat{P}(w_j)$) or more likely to be the correct node for $\boldsymbol{x}$ (i.e. having high $\hat{p}(x|w_j)$ since it is the closest node to $\boldsymbol{x}$) or both. Based on both probabilities and Bayes' theorem [14], the MAP criterion is expected to choose a winning node more accurately than if using only one of the probabilities. For example, the MAP criterion may prefer a category having a priori probability which is higher than that of another node although the normalized by the covariance distance of the former to the pattern is larger than that of the latter.

(2) *Node matching* (Vigilance Test): The node match is to ensure that the chosen node is able to represent the current environment where the robot is located. The test restricts the *J*th node hypervolume S_J to the maximal hypervolume allowed for a node S_{MAX}:

$$S_{J,k} \leq S_{\text{MAX},k} \tag{7}$$

where the hypervolume is defined as the determinant of the Gaussian covariance matrix. For a diagonal covariance matrix, this hypervolume is reduced to the product of the variances each for a dimension:

$$S_{J,k} \triangleq \det(\Sigma_{J,k}) = \prod_{d=1}^{M} \sigma^2_{J_d,k} \tag{8}$$

If the winning node fulfills the vigilance criterion in Eq. (7), learning is then performed. The edge connecting the winning node and previous winning node is updated as well. However, if both nodes are not connected, then a new edge is created to link both nodes and store their adjacent angle. If the winning node fails the vigilance criterion, the node is removed from the competition for this sensory input and the Bayesian ART determines another node till one complies with Eq. (7).

If all existing nodes in topological map fail the vigilance test, it means the robot is located at a new place. A new node that contains a mean vector which is the input pattern and an initial covariance matrix Σ_{init} that enables meeting Eq. (7) to represent this distinct place will be added to the map. A new edge is then created to connect the new node and previous winning node, and to store the adjacent angle between the two nodes.

(3) *Node learning*: When a chosen node fulfills the maximal hypervolume criteria in Eq. (7), the node elements are then updated as follows:

$$\hat{\mu}_{J,k(\text{new})} = \frac{N_J}{N_J+1}\hat{\mu}_{J,k(\text{old})} + \frac{1}{N_J+1}x_k \tag{9}$$

$$\hat{\Sigma}_{J,k(\text{new})} = \frac{N_J}{N_J+1}\hat{\Sigma}_{J,k(\text{old})} + \frac{1}{N_J+1}(x_k - \hat{\mu}_{J,k(\text{new})}) \times (x_k - \hat{\mu}_{J,k(\text{new})})^T * I \tag{10}$$

$$\hat{P}(w_J) = \frac{N_J}{\sum_{l=1}^{N_{node}} N_l} \tag{11}$$

$$N_J^{\text{new}} = N_J^{\text{old}} + 1 \tag{12}$$

where N_J is the number of times that ***J***th node have been chosen as winner for learning before receiving the current input sensory information and ***I*** is the identity matrix.

Thus, MBARAM incrementally produces the number of nodes necessary to learn and remember feature patterns for representing the explored environment as regions. The overall algorithm for topological map building is summarized in Algorithm 1.

Algorithm 1. Algorithm for MBARAM Topological Map Building Process

```
Require:
  Sensor Data $x$
  Covariance Matrix $\hat{\Sigma}$
  Prior Probability $\hat{P}(w)$
  Maximal Hypervolume $S_{\text{MAX}}$
Ensure:
  Nodes $D$
  Edges $\varepsilon$
  Input the sensor data $x$.
  if No node in the TopoMap then
    Add x to the TopoMap as new node.
    $D \leftarrow D_i \cup \{x\}$; $N_i = 0$
    Update the weights of the node $D_i(N, \hat{\mu}, \hat{\Sigma}, \hat{P}(w))$ using Eqs. (9)–(12)
  else
    Determine the winner node using Eqs. 1–6
    Execute the vigilance test for the winner node using Eq. (7)
    if Vigilance test fulfilled: $S_winner \leq S_{\text{MAX}}$ then
      Update the weights of winner node $D(N_J, \hat{\mu}_J, \hat{\Sigma}_J, \hat{P}(w_J))$ using Eqs.
  1–6
      if No edge between winner and prevWinnerthen
        Add edge between the current winner and previous winner nodes
  $\varepsilon \leftarrow \varepsilon \cup \{winner, prevWinner\}$
      else
        Update Edge that connecting winner and prevWinner
      end if
    else
      Reset the winner node and find a new winner from the TopoMap
  until it fulfill the vigilance test
      if All nodes are failed with vigilance test then
        Add $x$ to the TopoMap as new node.
        $D \leftarrow D_{i+1} \cup \{x\}$; $N_{i+1} = 0$
        Update the weights of the node $D_{i+1}(N, \hat{\mu}, \hat{\Sigma}, \hat{P}(w))$ using Eqs.
  (9)–(12)
        Add edge between the new node and previous winner nodes
  $\varepsilon \leftarrow \varepsilon \cup \{prevWinner, newnode\}$
      end if
    end if
  end if
```

2.1.3. Node localization

In topological framework, one of the localization method is to search the node that encodes the sensory information the most similar to the current sensor data. Note that for direct comparison, each node of the map must be distinct with each other. In our proposed method, recognizing a node is performed by comparing the current sensor data with the information stored in each of the nodes. To do this, Eq. (1) is used to search for the best node, and vigilance test in Eq. (7) is to ensure that the encoded information of the node is capable to represent the current sensor data. Fig. 2 illustrates the concept of the node localization. As mentioned in Section 2.1.1, each node of the map generated by MBARAM contains sensory information with the particular robot location. Therefore, all nodes of the map are unique for representing the environment, which helps to cope with perceptual ambiguities. Eq. (13) shows an example during the nodes comparison. The recognized node holds the highest value of M_j that fulfilled the vigilance test.

$$M_j = M_j^{\text{laser front}} + M_j^{\text{laser back}} + M_j^{\text{odometry}} \tag{13}$$

3. Experimental setup and results

In this section, the experimental setup and results obtained from the benchmark datasets simulation (Section 3.3) and physical robot implementation (Section 3.4) are detailed. The topological map forms by set of nodes and edges. Nodes are plotted as red color circle at the (x, y) coordinates in the stored position and edges are black color line that connecting all nodes in the map.

For simulation experiments, the performance of MBARAM in a topological map construction was verified using various benchmark datasets. Several measures have been used to evaluate the quality of a topological map. A commonly used measurement is the quantization error, which measures the average distance between

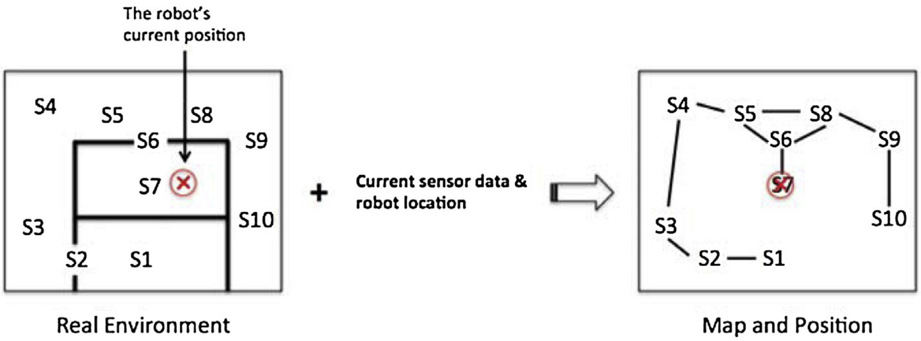


Fig. 2. Topological map constructed by MBARAM, in which each node encodes a sensory information and robot location, is available, recognizing the node which respects to the current sensor data allows the robot's metric position to be determined.

each data vector and its best matching unit (BMU). Here, the BMU is the winning node that has the highest match value and matches the vigilance parameter. Eq. (14) shows the calculation of the QE, where N is the number of data vectors and $m_{\vec{x}_i}$ is the best matching node in the corresponding $\vec{x}_i$ vector:

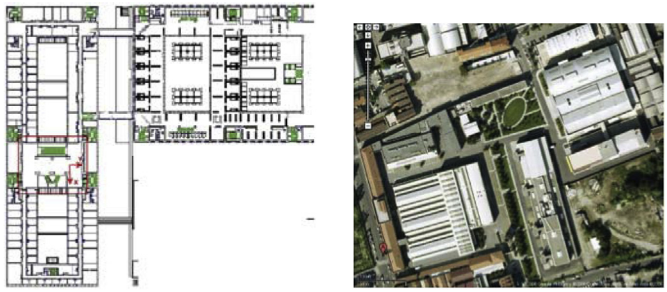
$$QE = \frac{1}{N}\sum \left\| \frac{\vec{x}_i - m_{\vec{x}_i}}{\vec{x}_{i(\max)}} \right\| \tag{14}$$

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 [31].

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. (15).

$$\mathrm{CR} = \frac{\text{Original robot's path size}}{\text{Topological map size}} \tag{15}$$

The benchmark datasets were downloaded from the Rawseeds Project, which aims to produce benchmarking tools for robotic systems. 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 [32,33]. These datasets were used as training samples to validate our proposed method, supplemented by odometry and laser scanner datasets. Fig. 3a and b illustrates the locations available to the robot during Rawseeds data gathering. Several indoor and outdoor scenarios have been defined by Rawseeds that each generate a dataset.



(a) Layout of the indoor environ ment

(b) Satellite view of the Politecnico di Milano campus

Fig. 3. Map for Rawseeds indoor and outdoor datasets collection (Source: Rawseeds Deliverable 2.1).

3.1. Parameters setup

According to [22], the Bayesian ART maximal category hypervolume parameter was optimized over the range $10^{-8} \le S_{\mathrm{MAX}} \le 10^2$. For diagonal covariance matrix case, each variance in the initial covariance matrix should be set to a small value to ensure that the node could grow and change its region during map learning [22]. We follow setup in [22] as the optimized values for the Bayesian ART parameters have been determined.

First, we set the initial covariance matrix to be spherical $\Sigma_{\mathrm{init}} = \sigma^2_{\mathrm{init}} I$, where I is the identity matrix, hence the parameter $\sigma^2_{\mathrm{init}} \ll (S_{\mathrm{MAX}})^{1/M}$ (7), (8) to confirm that the initial category hypervolume is much smaller than the maximal hypervolume allowed. Next, we set the maximal category hypervolume parameter value (S_{MAX}) as 1. Then, we set the value of the initial parameter ($\sigma^2_{\mathrm{init}} \ll 1^{1/682}$) to be two orders of magnitude lower than the maximal category hypervolume parameter which is 0.01. In addition, we set the $\hat{P}(w_j)_{\mathrm{init}}$ to 1 for making the mapping between a region deterministic instead of probabilistic.

We have configured MBARAM to three channels for taking front and rear laser scanner data and odometry information in our experiments. Therefore, we set the importance factor $\alpha = 0.4$ for each laser scanner channel as the laser scanner is scanning the environment and the data is representing the environment, which is the main factor for map learning process. We set the importance factor $\alpha = 0.2$ for odometry channel as odometry information is used to cope with perceptual ambiguities and not representing the environment condition. Thus, odometry channel important factor must be lower than laser scanner channel. Table 3 shows the parameters setting in our experiments.

3.2. Experiment design

The learning algorithm was written in C++ and compiled with Xcode. In both simulation and physical robot implementation, we have configured MBARAM to three channel for taking front laser scanner data, rear laser scanner data and odometry information. Note that odometry information is involved in map learning

Table 3
Indoor and outdoor datasets experiment parameters setting.

Parameters	Indoor experiment	Outdoor experiment
α	0.2 (odometry channel) 0.4 (front laser scanner channel) 0.4 (rear laser scanner channel)	0.2 (odometry channel) 0.4 (front laser scanner channel) 0.4 (rear laser scanner channel)
S_{MAX}	1	1
σ^2_{init}	0.01	0.01
$\hat{P}(w_j)_{\mathrm{init}}$	1	1

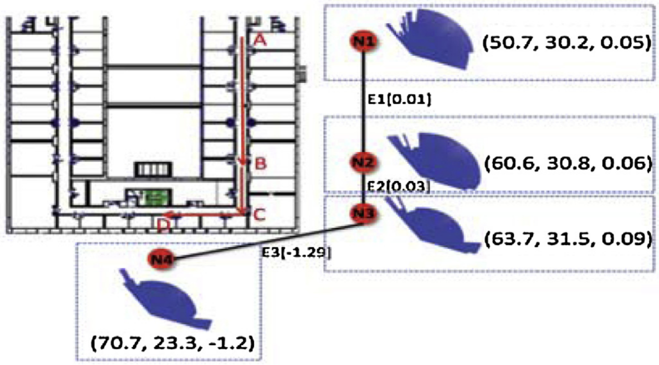


Fig. 4. Topological map building process.

because the information is to help to cope with the perceptual ambiguities. In the simulation experiment, we used the Rawseeds laser front dataset, laser rear dataset and odometry dataset for map learning. The procedures of map learning experiments as follow.

- Configure MBARAM to 3 channels ($k = 1–3$) and initialize its parameter value, such as α, S_{MAX}, σ^2_{init}, and $\hat{P}(w_j)_{\text{init}}$.
- Input M-dimensional laser front data ($x^f = x^f_1, \ldots, x^f_M$) to channel ($k = 1$), l-dimensional laser back data ($x^r = x^r_1, \ldots, x^r_l$) to channel ($k = 2$) and robot odometry data ($V = V_x, V_y, V_\phi$) to channel ($k = 3$) to MBARAM to activate map learning.
- MBARAM learning starts by adding one node ($n = 1$) to the map that store the first laser front ($\mu^{k=1}_{j=n} = x^f$), laser back data ($\mu^{k=2}_{j=n} = x^r$) and robot starting location($\mu^{k=3}_{j=n} = V$).
- For iteration (t), MBARAM obtains time-frame (t) sensors data.
- Then, calculate $M_{j,j=1,\ldots,n}$ in Eq. (4) using time-frame (t) sensors data and information that is encoded in all existing nodes to determine the node J that has the highest M_J (Node competition).
- Next, calculate winner node J hypervolume S_J in Eq. (8) (Node Matching). If S_J is larger than S_{MAX}, update node J information using Eqs. (9)–(12) (Node update).
- Else, reset node J and repeat step 5.
- Calculate the adjacent angle between node J and previous winner node, and update the edge between them.
- If all S_J less than S_{MAX}, add a new node to store all time-frame (t) sensors data and robot location.
- Add edge to connect the new node ($n + 1$) and previous winner node, and store the adjacent angle.
- Map building process stops when all time-frame sensors data are processed.

For the physical robot implementation, MBARAM receives sensor data from front laser range finder, rear laser finder and robot odometry system that mounted on the robot. The setup is detailed in Section 3.4.

3.3. Simulations

All simulation results using the Rawseeds datasets is shown in Sections 3.3.1 and 3.3.2. For better understanding, part of the topological map learning and building process are illustrated in Figs. 4 and 5. Fig. 4 shows that four nodes are generated when robot travel from place A to B, B to C, and C to D. Each node contains robot odometry information and sensory information that represent the particular region. Edges contain connected nodes angle information.

In addition, the node learning process was illustrated in Fig. 5. As shown in Fig. 5, the robot has no knowledge about the environment when it was located at place A (no node in the topological map). Therefore, node **1** was added to the topological map. Node **1** stores the current laser scanners data and robot location (node adding). During exploration from A to B, node **1** is chosen as a winner and its information was updated based on received sensory information as it fulfilled the vigilance test in Eq. (7) (node matching and learning). However, sensory information was different half way which caused node **1** to be mismatched and failed the vigilance test ($S_1 < S_{\text{MAX}}$). Therefore, node **2** was added to represent place B environment by storing sensory information collected at place B. A new edge was then added to connect node **1** and node **2**. When robot traveled from place B to C, node **2** is selected to update its information based on received sensory information, as node **2** has the highest match value in Eq. (4) compared to node **1** (Node Competition) and node **2** also fulfilled the vigilance test. Finally, node **3** was added and updated when the robot reached at place C because node **1** and node **2** both failed the vigilance test, which means these nodes are unable to represent environment C. The map learning and building process continues until robot received instructions from human operator.

3.3.1. Results: indoor datasets

In the first experiment, the odometry dataset and the front and rear laser scanner datasets were used as inputs. Fig. 6 shows the exact path the robot took in different environmental conditions such as (from left to right) dynamic with natural lighting, static with artificial lighting, dynamic with artificial lighting, and static with natural lighting. Fig. 7 illustrates MBARAM topological map learning result. Results showed all generated maps are almost identical to the robot exploration path. The map learning process is insusceptible to lighting conditions.

Table 4 summarizes the QE and CR of each topological map generated by MBARAM. Results showed that QE of each topological map are relatively small and almost similar because all datasets are gathered in the same environment. In addition, the maximum CR of the topological maps can reach to 224:1.

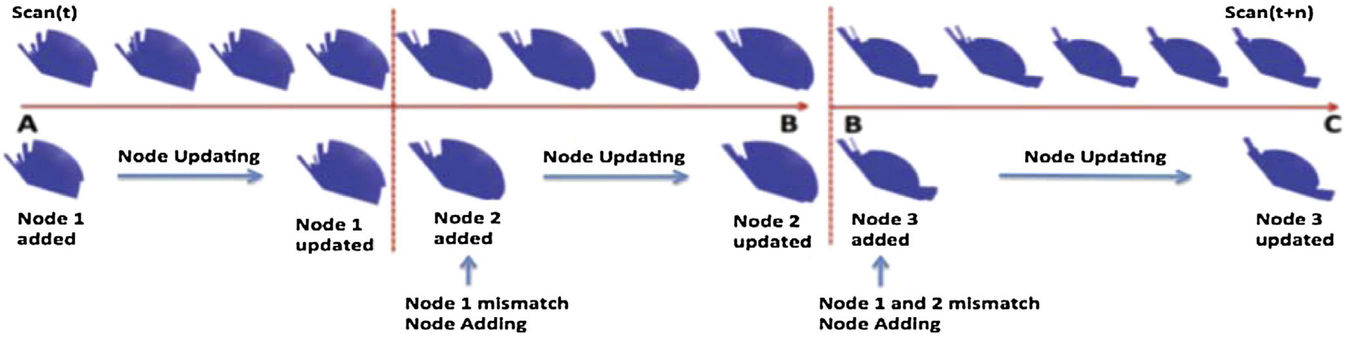


Fig. 5. Nodes learning process.

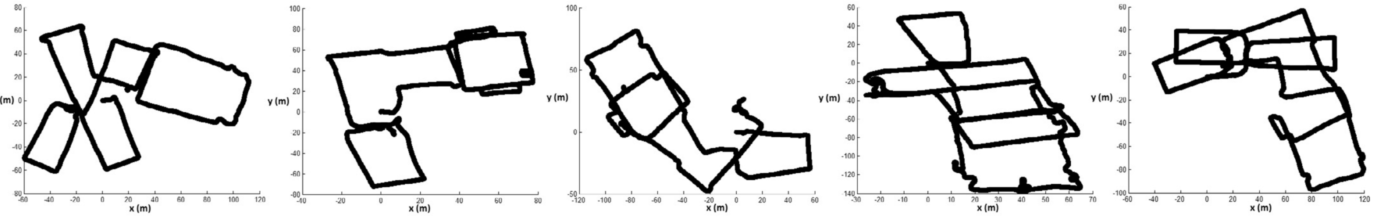


Fig. 6. The path taken by the robot through a pair of buildings at the University Milano-Biccoca in different environmental conditions for datasets gathering. From left to right: Bicocca_25a_Dynamic_Natural, Bicocca_25b_Static_Artificial, Bicocca_26a_Dynamic_Natural, Bicocca_26b_Dynamic_Artificial and Bicocca_27a_Static_Natural.

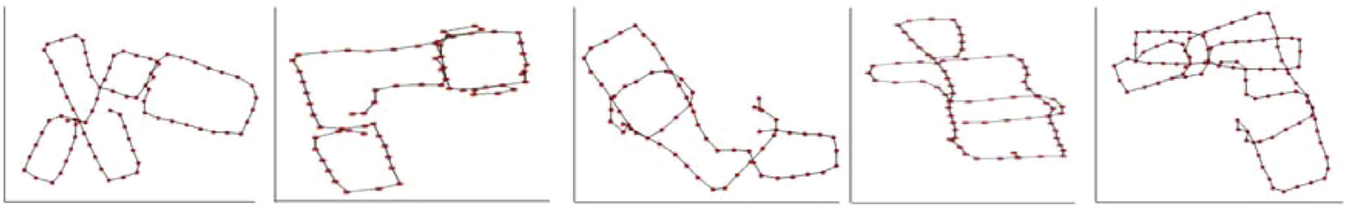


Fig. 7. The topological map constructed by MBARAM using the laser scanner and odometry dataset in the different environmental conditions corresponding to Fig. 6.

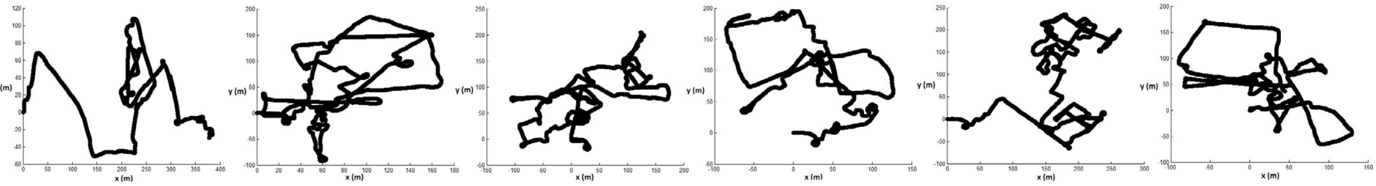


Fig. 8. Actual robot navigation path when gathering datasts (from left to right) Bovisa_2008-10-04_Static, Bovisa_2008-10-07_Dynamic, Bovisa_2008-10-11b_Static, Bovisa_2008-09-01_Static, Bovisa_2008-10-11a_Static and Bovisa_2008-10-06_Dynamic.

Table 4
MBARAM topological map evaluation for indoor datasets.

Datasets	Total nodes	QE	CR
Bicocca_25a_Dynamic_Natural	80	0.02	174:1
Bicocca_25b_Static_Artificial	181	0.03	190:1
Bicocca_26a_Dynamic_Natural	68	0.03	190:1
Bicocca_26b_Dynamic_Artificial	75	0.02	174:1
Bicocca_27a_Static_Natural	96	0.02	224:1

Table 5
MBARAM topological map evaluation for outdoor datasets.

Datasets	Total nodes	QE	CR
Bovisa_2008-10-04_Static	184	0.04	117:1
Bovisa_2008-10-07_Dynamic	171	0.02	98:1
Bovisa_2008-10-11b_Static	131	0.03	153:1
Bovisa_2008-09-01_Static	255	0.01	80:1
Bovisa_2008-10-11a_Static	273	0.02	89:1
Bovisa_2008-10-06_Dynamic	174	0.03	116:1

3.3.2. Results: outdoor datasets

As with the indoor datasets experiments, the same three channels were configured, i.e. odometry, the preprocessed front, and rear laser scanner dataset. The parameter settings are shown in Table 3. Fig. 8 shows the exact robot navigation path for gathering datasets. Fig. 9 illustrates the topological maps produced by the MBARAM. From the results, it is shown that the MBARAM was able to learn feature patterns from the outdoor and mixed datasets and produced topological maps that represent the environment.

Results showed that the maximum CR can up to 150:1 which almost similar to indoor datasets learning despite the complexity of outdoor environment. In addition, the QE of each generated topological maps are relatively low. Therefore, the learning of MBARAM is stable for both indoor and outdoor environment. Table 5 summarizes the number of nodes, QE and CR for each topological map. The number of nodes created in each map and is slightly higher than for indoor datasets because the areas explored are larger.

3.3.3. Results: Nodes localization

After MBARAM generated topological map for each dataset, the sufficiency of the map for representing the explored environment is verified. In the first experiment, we input all three datasets (odometry, front, and rear laser scanner) to the corresponding map, and MBARAM determines the best matching node for node localization using Eq. (1). If the winning node's weight is matched with

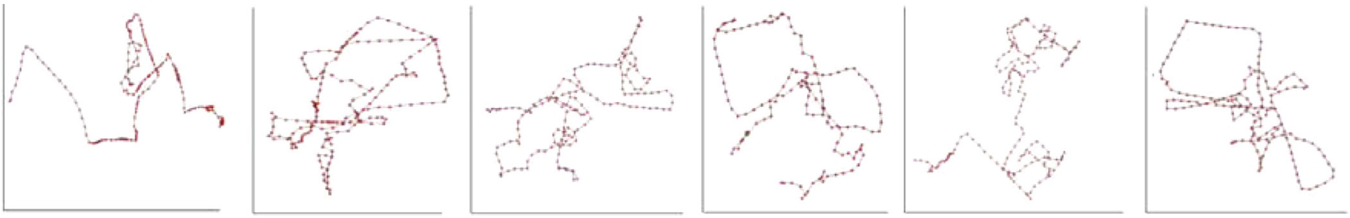


Fig. 9. Topological map generated by MBARAM for each datasets corresponding to Fig. 8.

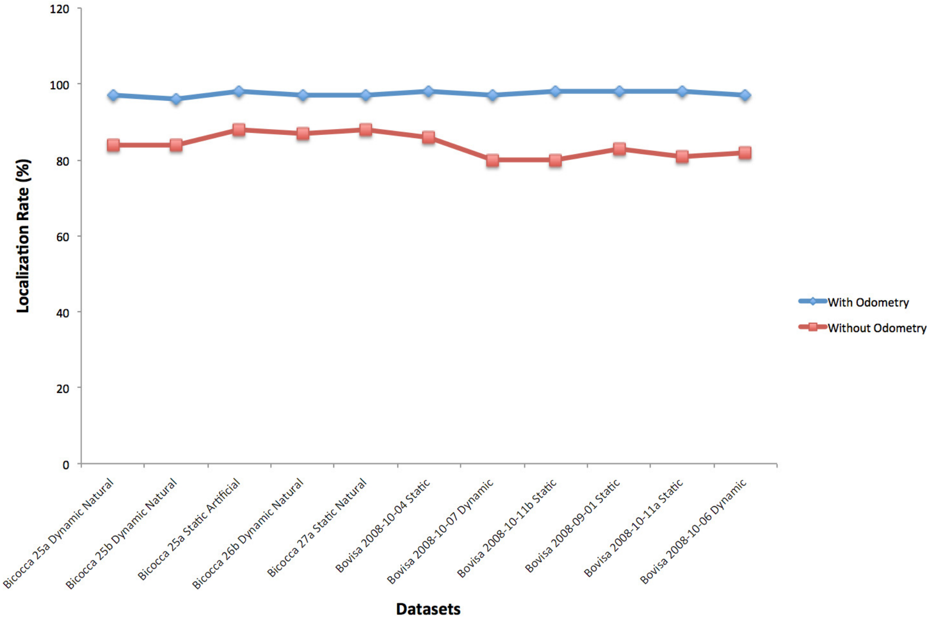


Fig. 10. Nodes localization rate for Rawseeds datasets.

robot's current sensory information and fulfils the criterion (7), then the winning node is chosen for localized. Eqs. (16) and (17) are used for further localization validation. If the euclidean distance between the winning node and the robot current location is less than a threshold, the robot is localized successfully and vice versa.

$$\text{Success localization} = \sqrt{(x_{\text{node}} - x_{\text{target}})^2 + (y_{\text{node}} - y_{\text{target}})^2} < 1 \qquad (16)$$

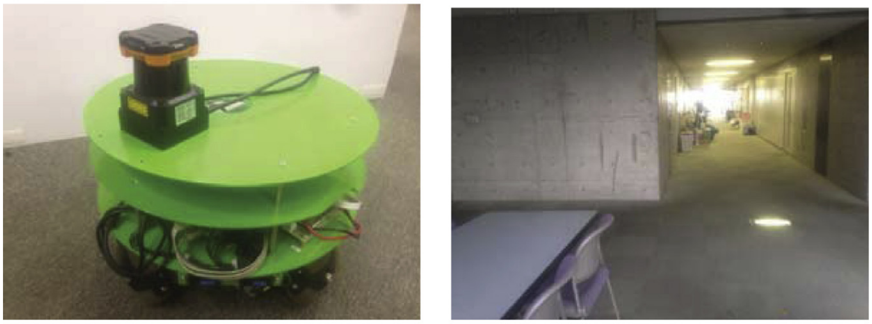
$$\text{Fail localizationn} = \sqrt{(x_{\text{node}} - x_{\text{target}})^2 + (y_{\text{node}} - y_{\text{target}})^2} > 1 \qquad (17)$$

In the second experiment, we input only the front and rear laser scanner to verify the localization rate without odometry information. Result of nodes localization for both experiments are shown in Fig. 10. With odometry information, the average of success rate of localization was approximately 97.4%, whereas without odometry information, the success rate was approximately 83.9%. The failure of localization for the first experiment is due to none of nodes in the map match with certain time step sensory information. However, sensory information that matched with topological nodes have similar content with these mismatch sensory information. Therefore, the overall performance is not affected by these mismatches. As in the second experiment, localization failures happened more often due to absence of odometry information which failed to disambiguate similar sensory information at different locations.

3.4. *Physical robot*

The following experiments are conducted using an omnidirectional robot, equipped with a Hokuyo UTM-30LX front laser scanner and an odometry system as shown in Fig. 11a. Signals from the sensors were sampled at 10 Hz. All computation and logging was ran on-board on an Intel Core i5 1.4 GHz processor. The omnidirectional robot typical operation speed is 0.1–0.5 m s^{-1}, it has obstacle avoidance and wall following capabilities.

The experiments were conducted in an area as shown in Fig. 11b, located in an university laboratory corridor that is moderately populated during the day and night. The size of the environment is 5 m × 40 m. The environment was by no means is static, with moving people, re-arrangement of furniture and equipment, changing door states, and lighting conditions. We set the experiment environment to be as natural as possible because we want to validate



(a) Ominorobot used in the experiment

(b) Photo of the environment of the experiment

Fig. 11. Physical robot experimental setup.

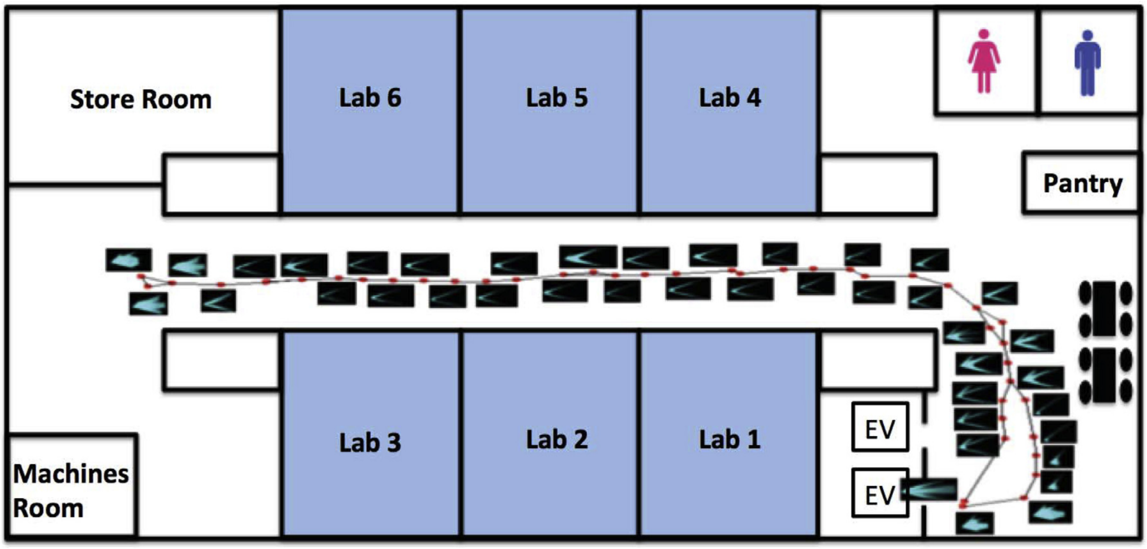


Fig. 12. Topological map built by MBARAM.

MBARAM is capable of dealing with random noise such as pedestrians and different object placements, without any assumptions or modifications, which is crucial for robot mapping research.

Parameters for the robot system and MBARAM algorithms were set as follows: $\alpha_{\text{odo}} = 0.2$, $\alpha_{\text{laser front}} = 0.4$ and $\alpha_{\text{laser rear}} = 0.4$; $S_{\text{MAX}} = 1$; $\sigma^2_{\text{init}} = 0.01$ and $\hat{P}(w_j)_{\text{init}} = 1$.

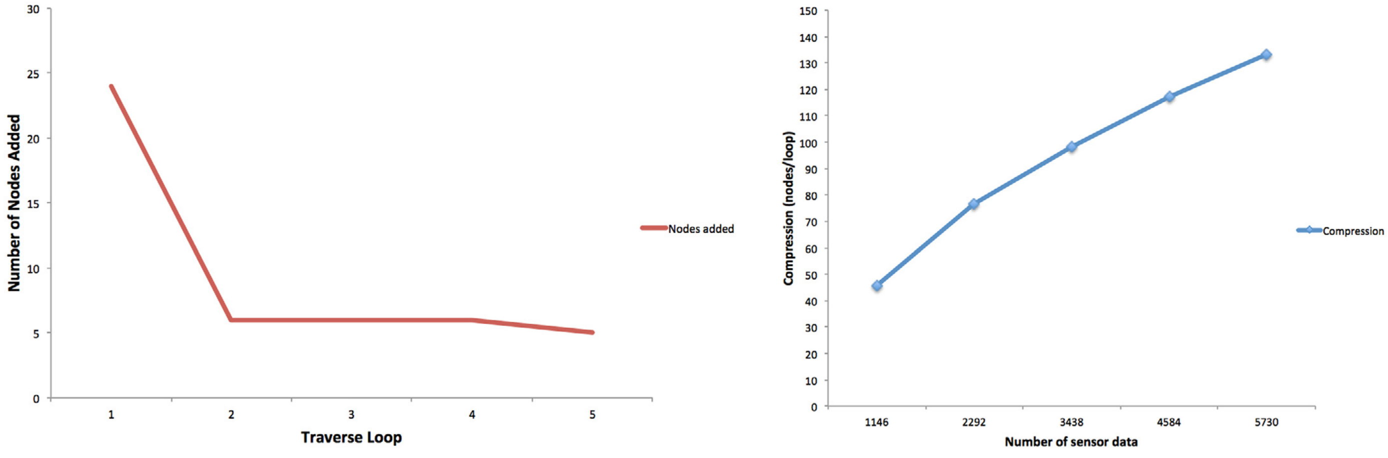
3.4.1. Results: environment learning and topological map generation

The first experiment was to drive the robot through the corridor, while collecting laser data and odometry for topological map construction. During exploration, the robot would automatically learn and build the map by recognizing whether the current sensory data corresponds to a node in the map. If the robot could not recognize the current sensory data, it meant the robot is located at new region, and a new node is added to the map for representing the new region. The robot will continue learning and update the map automatically until it receives instructions from operator.

Next, the robot traversed the corridor five times to verify the environment learning and map building. The final topological map built by MBARAM contained 47 nodes for representing the explored environment as shown Fig. 12. Nodes are plotted as red color circles at the (x, y) coordinates in the stored position, where linked nodes are joined with edges (black color line). Each topological node contains robot odometry information and laser range finder weight that represent particular region. Edges provide adjacent angle between connected nodes. Then, the topological map was updating continuously for the remaining exploration. The map is not highly accurate in a Cartesian sense, but need not be for effective navigation.

Fig. 13a shows the number of nodes added to the topological map for each traverse. During the first exploration, MBARAM has no knowledge about the environment, which causes it to detect many novel information and the 24 nodes were added to represent the places explored. However, the number of nodes added to the map decreased as the robot continued traversing the corridor. This is due to the topological map contains previously-learning knowledge about the environment after the first exploration, and this knowledge was used to detect new places or places that have been visited for the remaining traverses.

Noted that, small amounts of nodes (5–6 nodes in this experiment) are added for the remaining traverse as the robot travel path and environment conditions (pedestrian and placement of objects) vary in each traverse. As such, MBARAM detected some novel



(a) Number of nodes added for each traverse

(b) Compression for each traverse

Fig. 13. Performance of MBARAM in real environment.

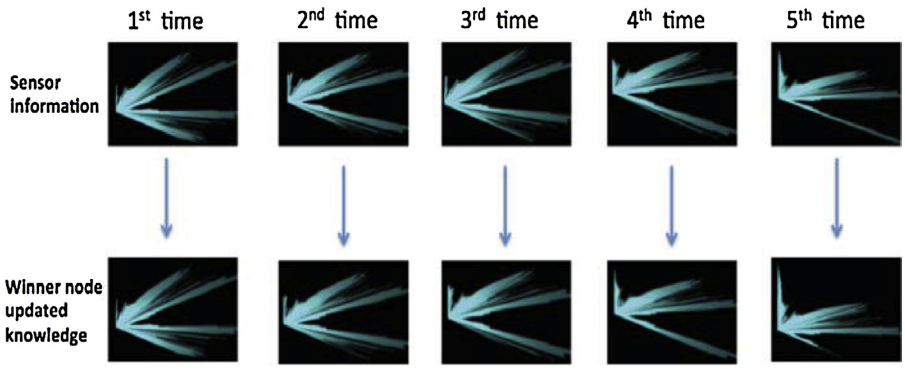


Fig. 14. Winner node learning and updating process. Top row: the laser scanner data when robot visit a same spot for five times. Bottom row: the same winner node was selected and updated its previously learned knowledge based on the received laser scanner data.

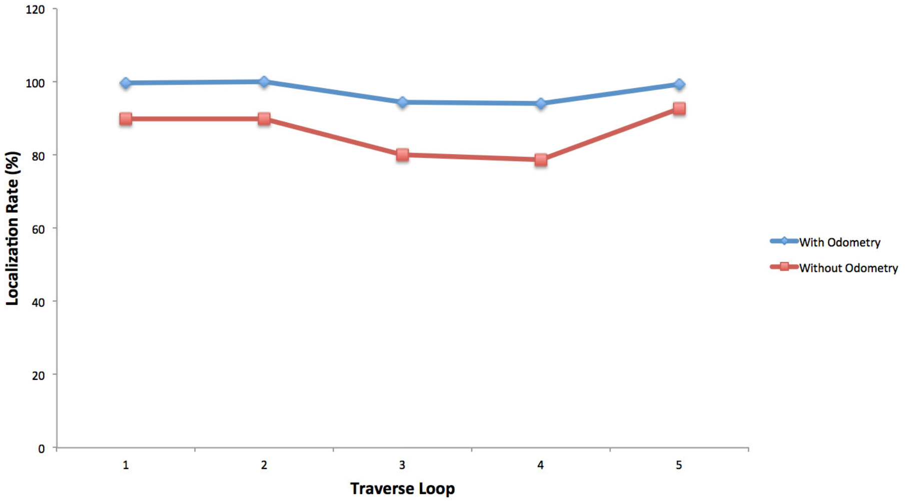


Fig. 15. Nodes localization rate.

sensory information and the map was updated according to the received sensor data. In addition, Fig. 13b shows that the compression results. Compression in this experiment refers to the number of nodes in map per traverse loop. The compression increases for each traverse as the robot receives 1146–1200 scans of sensor data (each loop), and total nodes in the map has small difference from second loop onwards (5 or 6 nodes difference). This proves that MBARAM is converging and capable of maintaining the map during exploration.

The map learning is then verified by examining one of the node update process when the robot visited the same area again. Fig. 14 shows one of the node learning process when the robot visits the same spot of the corridor for five times. As shown in Fig. 14, the sensory information varies despite the robot traveling to the same spot of the corridor. Thus, MBARAM selects and updates the node that encodes the sensory information that is most similar to the current sensor data to ensure that the topological map is up-to-date for representing the explored environment.

3.4.2. *Results: robot localization*

After obtaining the topological map of the environment, we placed the robot at the starting point and then instructed the robot to traverse the corridor ten loops for robot localization verification. During navigation, the robot kept searching nodes in the topological map based on its current sensory information for localization. If the winning node's weight matched with robot's current sensory information and fulfilled (as in Eq. (16)), the robot is localized successfully and vice versa.

In order to verify the advantages of taking multiple sensory sources, we enabled the robot to obtain odometry and laser scanner information for localization during the first five traverse. After the fifth traverse, we disabled the odometry channel and only laser scanner information fed to the robot. Fig. 15 shows the result of robot localization for each traverse. The average of success rate of localization with odometry information was approximately 97.5%, while without odometry information, the robot was still capable of performing localization but the average rate was 86%. Results showed that localization rate is improved when robot obtained more than one sensory source.

4. Discussions

We have shown that the developed MBARAM algorithm is able to construct a topological map from scratch using multiple raw sensory information and does not require any prior knowledge about the environment. The topological map is driven by ART modules, which has been used to explain place cell learning. The Bayesian ART module incrementally produces nodes for representing the explored environment as regions in real time. Each region is defined as multidimensional Gaussian elements that being used to measure their perceptual similarity with the robot's current sensory information.

The constructed topological map is structured into a set of linked regions with particular sensory information, which provides the flexibility and maintainability required for robot navigation. New nodes are constantly added and existing nodes are updated during environment learning. New nodes are connected with existing nodes by edges. These edges contain adjacent nodes angle to guide robot to travel from one location to another. Each node representing a region of the environment and it will be selected for learning when it matched with robot current sensory information. This proves that MBARAM will not forget previously-learned knowledge, but also builds the map according to robot's traverse path. From the simulation results, we shown that all the topological maps generated by MBARAM are almost identical to the original path taken by the robot.

In addition, the physical robot implementation results showed that number of nodes added to map significantly decreased after the first traverse, which means the MBARAM learning converged. Besides, in a real-world environment, sensory information may vary due to moving people or difference of objects placement, despite the robot visits the same spot. Experiment in Section 3.4.1 shows that MBARAM capable of selecting node and updating it's knowledge when the robot visits the same spot with varying sensory information. Therefore, MBARAM is capable of handling noise from the environment and the topological map remains up-to-date during exploration which is crucial for robot mapping.

The training process in MBARAM not only takes into account sensors measurement, but also the odometry of the robot. We assume that the robot's odometry is always accurate and this can be achieved by implementing indoor/outdoor GPS localization or Bluetooth localization. Therefore, MBARAM is able to disambiguate locations where the sensory information is very similar, which overcomes problem of online detection and recognition of topological nodes.

In the physical robot implementation experiment, we see that many nodes are added to remember the environment during the first traverse as MBARAM has no knowledge about the environment. However, the longer the robot traverses the corridor, the lesser nodes are added to the map as MBARAM has stored previously-learned knowledge which helps the robot to recognize the explored environment. Besides, the compression increases for each loop as the number of sensor data detected increases but nodes in the map remains almost the same after first loop (only 5 or 6 new nodes difference). If the amount of added nodes is the same or higher than the previous loop, the compression will not be increased, instead it is decreased. The experimental results show the learning of MBARAM is converging when the robot loops the same environment.

In addition, in the nodes matching and localization experiment, there are some cases where the robot failed to determine node for localization during navigation. This is due to the robot encountered a sudden change of the environment, such as pedestrian walking through the corridor and door changing state, which caused none of topological nodes match with these sensory information. This issue can be overcomed by comparing several time stamps of sensory information at the same location with the topological nodes for localization.

5. Conclusions

A new network, MBARAM for topological map learning and building is proposed in this paper. 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. With odometry information, the proposed method can disambiguate similar sensory information which overcomes the problem of online detection and recognition during learning process. Thus, it does not require high-level cognitive and prior knowledge about the environment to produce a topological map in natural environments. Our method was validated using benchmark Rawseeds datasets and physical robot implementation. Experimental results proved that the 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 provides sensory information, regions location, and traversability information that enables the robot to perform self localization and actions for navigation.

In future work, we will utilize other sensor data such as images to improve the map reliability and usability. In addition, we will integrate path planning into MBARAM for robot to perform path planning and to fully utilize the topological map. More experiments in more complex indoor and outdoor environments with different parameters value will be conducted for further validation.

Acknowledgements

This research is supported by University of Malaya Grant UM.C/625/1/HIR/MOHE/FCSIT/10.

References

[1] D. Kortenkamp, R.P. Bonasso, R. Murphy (Eds.), Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems, MIT Press, Cambridge, MA, USA, 1998.

[2] A. Schultz, W. Adams, Continuous localization using evidence grids, in: Robotics and Automation, 1998, Proceedings. 1998 IEEE International Conference on, vol.4, 1998, pp. 2833–2839, http://dx.doi.org/10.1109/ROBOT.1998.680595.

[3] H. Choset, K. Nagatani, Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization, IEEE Trans. Robot. Autom. 17 (2) (2001) 125–137, http://dx.doi.org/10.1109/70.928558.

[4] C. Estrada, J. Neira, J. Tardos, Hierarchical slam: real-time accurate mapping of large environments, IEEE Trans. Robot. 21 (4) (2005) 588–596, http://dx.doi.org/10.1109/TRO.2005.844673.

[5] N. Tomatis, I.R. Nourbakhsh, R. Siegwart, Hybrid simultaneous localization and map building: a natural integration of topological and metric, Robot. Autonom. Syst. 44 (1) (2003) 3–14, http://dx.doi.org/10.1016/S0921-8890(03)00006-X.

[6] G. Leivas, S. Botelho, P. Drews, M. Figueiredo, C. Häffele, Sensor fusion based on multi-self-organizing maps for slam, in: 2010 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010, http://dx.doi.org/10.1109/MFI.2010.5604482.

[7] R. Chatila, J. Laumond, Position referencing and consistent world modeling for mobile robots, in: Robotics and Automation. Proceedings. 1985 IEEE International Conference on, vol. 2, 1985, pp. 138–145, http://dx.doi.org/10.1109/ROBOT.1985.1087373.

[8] K. Benjamin, B. Yung-Tai, A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations, J. Robot. Autonom. Syst. 8 (1991) 47–63.

[9] J.-A. David Filliat, Map-based navigation in mobile robots: I. a review of localization strategies, Cogn. Syst. Res. 4 (4) (2003) 243–282, http://dx.doi.org/10.1016/S1389-0417(03)00008-1.

[10] H. Chang, C.S.G. Lee, Y. Hu, Y.-H. Lu, Multi-robot slam with topological/metric maps, in: Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, 2007, pp. 1467–1472, http://dx.doi.org/10.1109/IROS.2007.4399142.

[11] C. Chang-Hyuk, S. Jae-Bok, C. Woojin, Topological map building based on thinning and its application to localization, in: Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, vol. 1, 2002, pp. 552–557, http://dx.doi.org/10.1109/IRDS.2002.1041448.

[12] S. Thrun, Learning metric-topological maps for indoor mobile robot navigation, Artif. Intell. 99 (1) (1998) 21–71, http://dx.doi.org/10.1016/S0004-3702(97)00078-7.

[13] D. Van Zwynsvoorde, T. Simeon, R. Alami, Incremental topological modeling using local voronoi-like graphs, in: Intelligent Robots and Systems, 2000. (IROS 2000), Proceedings. 2000 IEEE/RSJ International Conference on, vol. 2, 2000, pp. 897–902, http://dx.doi.org/10.1109/IROS.2000.893133.

[14] Y. Tarutoko, K. Kobayashi, K. Watanabe, Topological map generation based on delaunay triangulation for mobile robot, in: in: SICE-ICASE, 2006. International Joint Conference, doi:10.1109/SICE.2006.315477, 2006, pp. 492–496.

[15] T.-B. Kwon, J.-B. Song, Thinning-based topological exploration using position possibility of topological nodes, Adv. Robot. 22 (2–3) (2008) 339–359, http://dx.doi.org/10.1163/156855308X292619.

[16] C. Shi, Y. Wang, J. Yang, Online topological map building and qualitative localization in large-scale environment, Robot. Auton. Syst. 58 (5) (2010) 488–496, http://dx.doi.org/10.1016/j.robot.2010.01.009.

[17] J. O'Keefe, J. Dostrovsky, The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat, Brain Res. 34 (1) (1971) 171–175, URL http://view.ncbi.nlm.nih.gov/pubmed/5124915.

[18] J. Berke, V. Hetrick, J. Breck, R. Greene, Transient 23–30 Hz oscillations in mouse hippocampus during exploration of novel environments, Hippocampus 18 (5) (2008) 519–529, http://dx.doi.org/10.1002/hipo.20435.

[19] M. Jean-Arcady, G. Agnès, G. Benoît, K. Mehdi, P. Patrick, B. Alain, The psikharpax project: towards building an artificial rat, Robot. Autonom. Syst. 50 (4) (2005) 211–223, http://dx.doi.org/10.1016/j.robot.2004.09.018.

[20] M. Milford, G. Wyeth, D. Prasser, Ratslam: A hippocampal model for simultaneous localization and mapping, in: Proceedings of the International Robotics and Automation (ICRA '04), Vol. 1, 2004, pp. 403–408, http://dx.doi.org/10.1109/ROBOT.2004.1307183.

[21] D. Ball, S. Heath, J. Wiles, G. Wyeth, P. Corke, M. Milford, Openratslam: an open source brain-based slam system, Autonom. Robots 3 (2013) 149–176, http://dx.doi.org/10.1007/s10514-012-9317-9.

[22] B. Vigdor, B. Lerner, The bayesian artmap, IEEE Trans. Neural Netw. 18 (6) (2007) 1628–1644, http://dx.doi.org/10.1109/TNN.2007.900234.

[23] A.-C. Zˇavoianu, G. Bramerdorfer, E. Lughofer, S. Silber, W. Amrhein, E.P. Klement, Hybridization of multi-objective evolutionary algorithms and artificial neural networks for optimizing the performance of electrical drives, Eng. Appl. Artif. Intell. 26 (8) (2013) 1781–1794, http://dx.doi.org/10.1016/j.engappai.2013.06.002.

[24] R.-C. David, R.-E. Precup, E.M. Petriu, M.-B. Rˇadac, S. Preitl, Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity, Inform. Sci. 247 (2013) 154–173, http://dx.doi.org/10.1016/j.ins.2013.05.035.

[25] N. El-Hefnawy, Solving bi-level problems using modified particle swarm optimization algorithm, Int. J. Artif. Intell. 12 (2) (2014) 88–101.

[26] F. Valdez, P. Melin, O. Castillo, An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms, Appl. Soft Comput. 11 (2) (2011) 2625–2632, http://dx.doi.org/10.1016/j.asoc.2010.10.010.

[27] S. Grossberg, Beta oscillations and hippocampal place cell learning during exploration of novel environments, Hippocampus 19 (9) (2009) 881–885, http://dx.doi.org/10.1002/hipo.20602.

[28] G.A. Carpenter, S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, Comput. Vision Graph. Image Process. 37 (1) (2014), http://dx.doi.org/10.1016/S0734-189X(87)80014-2.

[29] A.-H. Tan, Adaptive resonance associative map, Neural Netw. 8 (3) (1995) 437–446, http://dx.doi.org/10.1016/0893-6080(94)00092-Z.

[30] C. Shi, Y. Wang, J. Yang, Online topological map building and qualitative localization in large-scale environment, Robot. Autonom. Syst. 58 (5) (2010) 488–496.

[31] S. McGlinchey, M. Pe na, C. Fyfe, Quantization errors in the harmonic topographic mapping, in: Proceedings of the 5th WSEAS International Conference on Signal Processing, SIP'06, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2006, pp. 105–110, URL http://dl.acm.org/citation.cfm?id=1983937.1983961.

[32] S. Ceriani, G. Fontana, A. Giusti, Rawseeds ground truth collection systems for indoor self-localization and mapping Autonomous Robots, vol. 27, Kluwer Academic Publishers, Hingham, MA, USA, 2009, pp. 353–371, pp. 353–371, http://dx.doi.org/10.1007/s10514-009-9156-5.

[33] G. Fontana, M. Matteucci, D.G. Sorrenti, Rawseeds: building a benchmarking toolkit for autonomous robotics, in: F. Amigoni, V. Schiaffonati (Eds.), Methods and Experimental Techniques in Computer Engineering, Springer Briefs in Applied Sciences and Technology, Springer International Publishing, 2014, pp. 55–68.