2015 IEEE Symposium Series on Computational Intelligence

# Genetic Bayesian ARAM for Simultaneous Localization and Hybrid Map Building

Wei Hong, Chin*, Chu Kiong, Loo*, Naoyuki Kubota†, and Yuichiro Toda†
*Faculty of Computer Science and Information Technology
University of Malaya, Kuala Lumpur, 50603 Malaysia
Email: weihong118118@gmail.com, ckloo.um@um.edu.my
† Faculty of System Design
Tokyo Metropolitan University, Tokyo, 191-0065, Japan
Email: kubota@tmu.ac.jp, toda-yuuichirou@ed.tmu.ac.jp

***Abstract*—This paper presents a new framework for mobile robot to perform localization and build topological-metric hybrid map simultaneously. The proposed framework termed as Genetic Bayesian ARAM consists of two main components: 1) Steady state genetic algorithm (SSGA) for self-localization and occupancy grid map building and 2) Bayesian Adaptive Resonance Associative Memory (ARAM) for topological map building. The proposed method is validated using a mobile robot. Result show that Genetic Bayesian ARAM capable of generate hybrid map online and perform localization simultaneously.**

## I. INTRODUCTION

In recent intelligent robotics research, autonomous robot capable of solving complicated tasks in many applications area. A human-friendly autonomous robot should be able to knows where it is and constructs its own map for representing the operating environment. Therefore, building the representation of map is important to maintain autonomous behavior such as robot localization, path planning and collision avoidance.

Robot mapping can be grouped into metric maps, topological maps, and hybrid maps that combine both metric and topological map information [1]. In the metric approaches, the environment is represented as a set of objects with coordinates in a 2D space [2]. The metric maps are build based on feature representation [3] or free-space representation [4]. Pure metric maps are vulnerable to inaccuracies in both map building odometry sytem of the robot [5].

In the topological map approaches, the environment is represented by a set of *distinctive places* where these places are linked to each other by path [6]. Place definition are based on sensory information received from sensors that operating in the environment. In its common way, a topological map is a sparse representation of the environment that only represents important places for navigation. However, topological map approaches require intensive exploration of the environment if higher accuracy for localization is needed. Another crucial factor that impedes topological map effectiveness is the online detection and recognition of topological nodes especially when dealing with unreliable sensors or in a dynamic environment.

More recently, hybrid approaches are developed, which combine metric and topological methods into hybrid maps. These methodologies vary in the structure the specific maps are generated, interconnected, and utilized. For instance, the topological layer is built on top of grid-based map by handling Voronoi diagrams of the thresholded grid [7]. A statistical maximum likehood approach [8] generates a topological map that solves global position alignment problem and also being used for building a fine-grained map. Besides, the map constructed by the Atlas framework contains a list of multiple local maps with a limited size. Each node represents a local coordinate frame of reference and each edge contains the transformation information between local frames [9].

In this paper, we propose a new framework that combining our previous work SSGA [10] and Bayesian ARAM [11] for hybrid map building. The proposed method termed as Genetic Bayesian ARAM which constructs map that embodies both a metric and a topological representation. First, SSGA allows robot to perform self-localization and metric mapping simultaneously based on occupancy grid mapping. Then, Bayesian ARAM utilizes the localization information from the metric map and sensory information from the explored environment to construct the topological map. Nodes in the topological map represent distinct places, while edges connect nodes and store robot's bearing information such as orientation and direction. The metric grid map describes the explored environment outline for human operators understanding and further operations. In addition, it also provides global position of robot for Bayesian ARAM to generate the topological map and overcome the online detection and recognition problem. With the topological map, robot could perform localization and path planning without recalculating the entire metric grid map.

Contributions of this paper are: (i) it is an incremental and unsupervised learning framework that enables robot to perform self-localization and topological metric map building automatically; (ii) it does not require high-level cognitive knowledge, artificial landmark and feature extraction process to make it work in a natural environment; (iii) it can process single or multiple sensory sources for topological map building; does not require odometry system such as encoder and GPS for robot position estimation. The rest of the paper is organized as follows. Section II introduces the theoretical framework of the proposed online hybrid map building. The experimental results shown in Section III while results of map building and localization are discussed in Section IV. Concluding remarks are finally presented in Section V.

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.48

## II. GENETIC BAYESIAN ARAM

In our proposed hybrid mapping method, the *notion* of map has both a metric and a topological feature. On the metric side, the environment is represented by a global occupancy grid map. SSGA simultaneously perform localization and metric map building based on occupancy grid mapping.

On the other hand, Bayesian ARAM obtains robot's position from the metric mapping and sensory information to build the topological map. Bayesian ARAM continuously cluster sensory sources as nodes and create edges to connect one another when transistions between nodes are experienced. Each particular connection contains robot's orientation or bearing while nodes are represent distinct places of the explored environment. The proposed method continuously updates the hybrid map representations with little or no human intervention. The overall process of our proposed framework as shown in Figure 1.

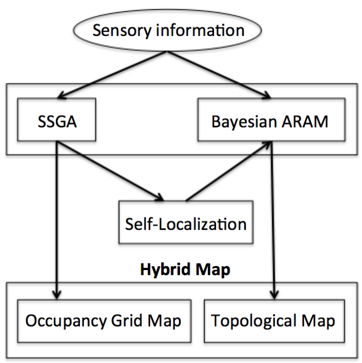


Fig. 1: Genetic Bayesian ARAM overall map building process

### A. Building the metric map and localization

As mentioned in previous section, we utilize the occupancy grid mapping [12], [13] for constructing the metric map. Figure 2 illustrates the concept of the occupancy grid map.

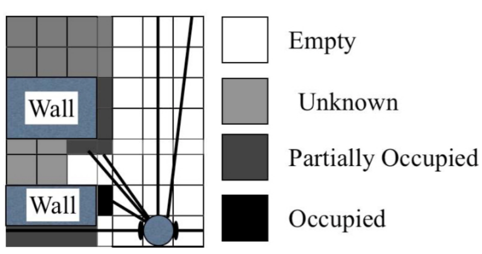


Fig. 2: Concept of the occupancy grid map

The value of each discrete cell is represented as follows:

$$\text{map}_0(x, y) = \begin{cases} 1 & \text{if occupied} \\ 0.5 & \text{if partially occupied} \\ 0 & \text{if unknown} \\ -1 & \text{if empty} \end{cases} \quad (1)$$

Initially, all cells value are set as 0. The measurement value is represent as $(d_i, \theta_i)$, where $i = 1, 2, \ldots, M, j = 1, 2, \ldots, L,$; $d_i$ is the measurement distance from laser range finder and $\theta_i$ is the angle of the measurement direction; $M$ is the number of total measurement directions; $L_i = [\alpha^{\text{Res}} \cdot d_i]$ is the number of resolution for the map generation by the occupancy grid model.

The map update process as shown in Algorithm 1 where $(x_p, y_p)$ is the position of the mobile robot; $r_p$ is the posture; $d_i$ is the measurement distance from laser range finder in the *i*th direction; $\theta_i$ is the angle of the measurement direction; $\alpha^{\text{MAP}}$ is the scale factor mapping from the real world to the grid map; $f(\cdot)$ in equation 4 is a function according to IF-THEN rules as shown in Table I.

**for** *i=1 to M* **do**
  **for** *j=1 to* $L_i$ **do**

$$u_{i,j} = \frac{j}{L_i}\Big(d_i \cos(\theta_i + r_p)\Big) + x_p$$
$$v_{i,j} = \frac{j}{L_i}\Big(d_i \sin(\theta_i + r_p)\Big) + y_p \quad (2)$$

$$x_{i,j} = [\alpha^{\text{Map}} \cdot u_{i,j}]$$
$$y_{i,j} = [\alpha^{\text{Map}} \cdot v_{i,j}] \quad (3)$$

$$\text{map}_0(x_{i,j}, y_{i,j}) = f(map_0(x_{i,j}, y_{i,j}), j) \quad (4)$$

  **end**
**end**

**Algorithm 1:** Map update process

TABLE I: State transitions of map-update

| Condition | | Output |
|---|---|---|
| $j$ | $\text{map}_0(x, y)$ | |
| $j < L$ | 0 | -1 |
| $j < L$ | 1 | 0.5 |
| $j = L$ | 0 | 1 |
| $j = L$ | -1 | 0.5 |

We apply SSGA for the correction of the position and angle. As one stream of evolutionary computing, genetic algorithms (GAs) have been effectively used for optimization problems in robotics [14]. GAs can produce a feasible solution, not necessarily an optimal one, with less computational cost. The main role of GAs in robotics is the optimization in modeling or problem-solving. SSGA simulates the continuous model of the generation, which eliminates and generates a few individuals in a generation. A candidate solution is composed of numerical parameters or revised value to the current position $(g_{k,x}, g_{k,y})$ and rotation $(g_{k,r})$.

In the SSGA, only a few existing solutions are replaced with the candidate solution generated by the crossover and mutation. In this experiment, we utilize the elitist cross over and adaptive mutation. Elitist crossover randomly selects one individual and generates an individual by incorporating genetic information from the selected individual and best individual in order to obtain feasible solutions rapidly.

Next, the following adaptive mutation is applied to the generated individual as follows:

$$g_{k,h} \to g_{k,h} + \left(\alpha^{\text{SSGA}} \cdot \frac{f_{\max} - f_k}{f_{\max} - f_{\min}} + \beta^{\text{SSGA}}\right) \cdot N(0,1) \quad (5)$$

where $f_k$ is the fitness value of the *k*th individual, $f_{\max}$ and $f_{\min}$ are the maximum and minimum of fitness values in the population; $N(0,1)$ indicates a normal random value; $\alpha^{\text{SSGA}}$ and $\beta^{\text{SSGA}}$ are the coefficient and offset respectively. The fitness value of the *k*th candidate solution is calculated using equation .

$$fit_k^{\text{Loc}} = \sum_{i=1}^{M} \text{map}_0(x_{i,L}, y_{i,L}) \quad (6)$$

*B. Building the topological map*

The MBARAM build the topological map beginning with one node (category) at the first perception. Next, the map is updating continuously according to robot location generated from SSGA and sensory information that gathered from robot's sensors. The map update as shown in Algorithm 2.

**Data**: Sensory data and robot position from SSGA
**Result**: Topological map
**if** *map < one node* **then**
  add node to map;
  **return** true
**else**
  search all nodes to determine winner;
  **if** *winner > vigilance* **then**
    update winner node;
    **if** *winner node and previous winner node no edge* **then**
      add edge;
    **else**
      update edge;
    **end**
  **else**
    reset winner;
    add node to map ;
    add edge;
  **end**
**end**

**Algorithm 2:** Topological map building

Place definitions are directly obtained 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 robot location $\vec{V}$ encoded from SSGA. It is defined as a multidimensional Gaussian component such as mean vector, covariance matrix, and prior probability. Such node definition is based solely on the robot's perceptual capacities and does not rely on human definition of what a place is supposed to be. This make places easier to recognize from sensory information.

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

1) *Node Competition:* In this stage, all existing nodes compete 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:

$$M_j = \hat{P}(w_j|x) = \frac{\hat{p}(x|w_j)\hat{P}(w_j)}{\sum_{l=1}^{N_{cat}} \hat{p}(x|w_l)\hat{P}(w_l)} \quad (7)$$

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 $w_j$:

$$\hat{p}(x|w_j) = \frac{1}{(2\pi)^{M/2}|\hat{\Sigma}_j|^{1/2}} \times \exp\{-0.5(x-\hat{\mu}_j)^T\hat{\Sigma}_j^{-1}(x-\hat{\mu}_j)\} \quad (8)$$

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:

$$\begin{aligned} M_j &= \sum_{k=1}^{K} \alpha_k[\hat{P}(w_{(j,k)}|x_k)] && (9)\\ &= \sum_{k=1}^{K} \alpha_k\left[\frac{\hat{p}(x_k|w_{j,k})\hat{P}(w_{j,k})}{\sum_{l=1}^{N_{cat}} \hat{p}(x_k|w_{l,k})\hat{P}(w_{l,k})}\right] && (10)\end{aligned}$$

where

$$\hat{p}(x_k|w_{j,k}) = \frac{1}{(2\pi)^{M/2}|\hat{\Sigma}_{j,k}|^{1/2}} \times \exp\{-0.5(x_k-\hat{\mu}_{j,k})^T\hat{\Sigma}_{j,k}^{-1}(x_k-\hat{\mu}_{j,k})\} \quad (11)$$

and $\alpha_k$ influence factor for each channel and the sum of $\alpha_k$ is 1.

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

$$J = \arg\max(M_j) \quad (12)$$

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

$$S_{J,k} \leq S_{\text{MAX},k} \quad (13)$$

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:

$$S_{J,k} \triangleq \det(\Sigma_{J,k}) = \prod_{d=1}^{M} \sigma_{J_d,k}^2 \quad (14)$$

If the winning node fulfills the criterion (13), learning is performed. Else, the node is removed from the competition for this sensory input and continue searching for another node until one is comply with (13). If all existing nodes failed the vigilance test, it means robot is located at a new place. Then, a new node is added to the map which stores the input pattern and an initial covariance matrix $\Sigma_{init}$ and robot location to represent this distinct place. New edge is added to connect the new node with previous winner node.

3) *Node Learning*: When a chosen node fulfills the maximal hypervolume criteria(13), the node elements are updated as follows:

$$\hat{\mu}_{J,k(\text{new})} = \frac{N_J}{N_J+1}\hat{\mu}_{J,k(\text{old})} + \frac{1}{N_J+1}x_k \tag{15}$$

$$\hat{\Sigma}_{J,k(\text{new})} = \frac{N_J}{N_J+1}\hat{\Sigma}_{J,k(\text{old})} + \frac{1}{N_J+1}(x_k - \hat{\mu}_{J,k(\text{new})}) \times (x_k - \hat{\mu}_{J,k(\text{new})})^T * I \tag{16}$$

$$\hat{P}(w_J) = \frac{N_J}{\sum_{l=1}^{N_{node}} N_l} \tag{17}$$

$$N_J^{\text{new}} = N_J^{\text{old}} + 1 \tag{18}$$

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.

## III. EXPERIMENTAL RESULTS

We have conducted the following experiments using an omni-directional robot with four omni-wheels and DC motors as shown in Figure 3. The robot can move in different omni-direction by changing the combination of output levels of motors. Furthermore, the robot equipped with a Hokuyo laser range finder (UTM-30LX) and obstacle avoidance ability for self-localization and map building. Table II shows the hardware specification of the robot.

Fig. 3: Omni-directional robot

TABLE II: Specifications of omni-directional mobile robot

| | |
|---|---|
| Diameter | 300mm |
| Height | 177mm |
| Maximal Speed | 1.5km/h |
| Operating Time | 1 hour |
| Communication | Wi-Fi (2.4 GHz) |

The experimental place is our university laboratory room and corridor, moderately populated during the day and night. The environment was by no means static, with moving people, re-arrangement of furnitures and equipments and changing door states. Parameters for the robot system and MBARAM algorithms were set as follows: $\alpha_{\text{odo}} = 0.2$, $\alpha_{\text{laser}} = 0.8$, $S_{\text{MAX}} = 1$, $\sigma^2_{\text{init}} = 0.01$ and $\hat{P}(w_j)_{\text{init}} = 1$, parent candidates($\mu$) $= 1000$ and offspring candidates($\lambda$) $= 500$.

Next, we commanded the robot to traverse the corridor 3 loops and to our laboratory room during the last loop for verifying the hybrid map building. During the navigation, robot perform self-localization and build the grid map by using laser range finder measurement data. Then, robot's position are used by Bayesian ARAM for topological map building. Figure 4 shows the hybrid map for the first loop traverse over the corridor. Figure 5 shows the final hybrid map contained a grid map representing the outline of explored corridors and the laboratory room and a topological map with 79 nodes for representing the explored path. Nodes are plotted as red color circles at the (x,y) coordinates; linked nodes are joined with edges (black color line). Each node contains robot position (x,y, $\theta$) generated by SSGA and laser range finder weight that represents particular explored region as shown in Figure 6.

During the first traverse, robot has no knowledge about the environment which cause it added 50 nodes to the map for the first loop. However, only 29 nodes are added to the map as the robot continue traverse the corridor and laboratory room for the second and third times. This is because the map contained previously learned knowledge about the environment after the first traverse. These knowledge were used by robot for detecting new place or places that already visited for the remaining loop.

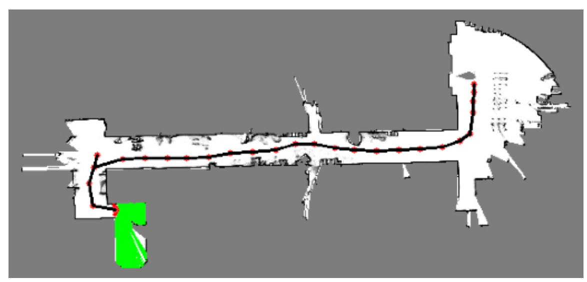

Fig. 4: Hybrid map for first loop

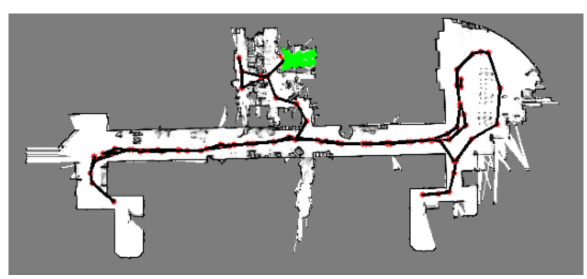

Fig. 5: Final hybrid map

## IV. DISCUSSION

We have shown that the Genetic Bayesian ARAM framework is able to construct a metric-topological hybrid map from

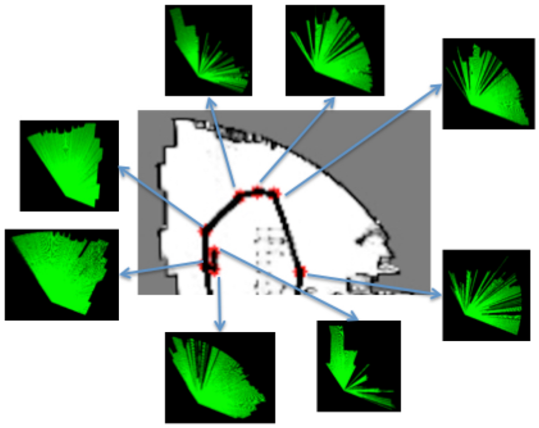

Fig. 6: Enlarged hybrid map with topological node information

scratch using unprocessed sensory information and does not require any feature extraction and prior knowledge about the environment.

In the experiment, SSGA continuously perform localization and metric mapping without using any odometry system. Then, Bayesian ARAM use these localization data and laser scanner data for topological map building. The localization data are important for distinguish places where the sensory information is very similar, which overcomes the problem of online detection and recognition of topological nodes. After building the map, robot can perform path planning by just compare it's current location with topological nodes without recalculating the grid map. These features help to compensate the weakness of metric map and topological map.

## V. Conclusion

The experiments presented show the feasibility of the proposed approach. Both metric and topological maps were generated as expected, correctly representing the environment according to the method. The proposed method does not requires high-level cognitive knowledge or any artificial landmark to construct the hybrid map which make it ready work in natural environment. In addition, robot can perform self-localization automatically without any odometry system.

Future work in this subject will include an analysis of effectiveness of the value of framework parameters. Besides, we will extend our method for robot path planning and loop closing to fully utilize the hybrid map. Lastly, we will conduct more experiments in different kind of indoor and outdoor environment for further validation.

## References


[1] David Filliat, Jean-Arcady Meyer, "Map-based navigation in mobile robots: I. A review of localization strategies," Cognitive Systems Research, Volume 4, Issue 4, Pages 243-282, ISSN 1389-0417, http://dx.doi.org/10.1016/S1389-0417(03)00008-1 (2003).

[2] Sebastian Thrun, "Learning metric-topological maps for indoor mobile robot navigation," Artificial Intelligence, Volume 99, Issue 1, Pages 21-71, ISSN 0004-3702, http://dx.doi.org/10.1016/S0004-3702(97)00078-7, (1999).

[3] Gutmann, J.-S.; Konolige, K., "Incremental mapping of large cyclic environments," Computational Intelligence in Robotics and Automation, 1999. CIRA '99. Proceedings. 1999 IEEE International Symposium on , vol., no., pp.318,325, (1999).

[4] Arleo, A.; del R.Millan, J.; Floreano, D., "Efficient learning of variable-resolution cognitive maps for autonomous indoor navigation," Robotics and Automation, IEEE Transactions on , vol.15, no.6, pp.990,1000, (1999).

[5] Chatila, R.; Laumond, J., "Position referencing and consistent world modeling for mobile robots," Robotics and Automation. Proceedings. 1985 IEEE International Conference on , vol.2, no., pp.138,145, (1985).

[6] Benjamin Kuipers, Yung-Tai Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," Robotics and Autonomous Systems, Volume 8, Issues 12, (1991).

[7] S. Thrun, A. Bucken, W. Burgard, D. Fox, T. Frohlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schmidt, Map learning and high-speed navigation in RHINO. Cambridge, MA, USA: MIT Press, pp. 2152, (1998).

[8] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. J. Kuipers, Integrating topological and metric maps for mobile robot navigation: a statistical approach, in Proc. of the Fifteenth National/Tenth Conf. on Artificial Intelligence/Innovative Applications of Artificial Intelligence, ser. AAAI 98/IAAI 98. Menlo Park, CA, USA: American Association for Artificial Intelligence, pp. 989995 (1998).

[9] M. Bosse, P. M. Newman, J. J. Leonard, and S. Teller, SLAM in Large-scale Cyclic Environments using the Atlas Framework, The International Journal of Robotics Research, vol. 23, no. 12, pp. 11131139, (2004).

[10] Yuichiro Toda, Shintaro Suzuki, and Naoyuki Kubota, "Evolutionary computation for intelligent self-localization in multiple mobile robots based on SLAM," In Proceedings of the 5th international conference on Intelligent Robotics and Applications, 229-239, (2012).

[11] Wei Hong Chin, Chu Kiong Loo, and Naoyuki Kubota, " Multi-channel Bayesian Adaptive Resonance Associative Memory for Environment Learning and Topological Map Building," in 4th International Conference on Informatics, Electronics & Vision (ICIEV), (2015).

[12] Thrun, S. "Learning Occupancy Grid Maps With Forward Sensor Models," Autonomous Robots 25(2), 111127 (2003)

[13] Lee, K., Chung, W.K., "Effective Maximum Likelihood Grid Map with Conflict Evaluation Filter using Sonar Sensors," IEEE Transactions on Robotics 25(4), 887901 (2009)

[14] Fogel,D.B., "EvolutionaryComputation," IEEE Press (1995)