Topological Map Building Based on A Genetic Algorithm for Simultaneous Localization and Mapping

Hironobu Sasaki¹, Naoyuki Kubota¹², Kazuhiko Taniguchi³

¹ Department of System Design, Tokyo Metropolitan University, Tokyo Japan
² "Interaction and Intelligence" SORST, Japan Science and Technology Corporation (JST)
³ Kinden Corporation, Kyoto R and D Center, JAPAN

Abstract: This paper proposes a method of self-localization and map building based on a steady-state genetic algorithm and a growing topological neural network for a mobile robot. The growing topological neural network sequentially adds a node to the map according to the measured distance by a laser range finder. When the difference between the measured distance and the map data is large, the proposed method corrects the self-location of the robot by using the steady-state genetic algorithm, and updates the map to be more accurate. Finally we show experimental results of the proposed method.

1. Introduction

Recently, various types of robots and robot technologies have been developed for the use in real world fields such as manufacturing systems, building industry, aerospace development, and human society instead of human labor [1-16]. Such robots are often used in unknown and dynamic environments. Therefore, intelligent technologies [17-19] are required for autonomous execution by robots in the environments. One of the important intelligent capabilities is to build an environmental map and to estimate and correct the self-location. Map building by mobile robots has a long history [20-28]. The map building method can be divided into two main methods of metric and topological approach.

In the metric approach, an environment is represented by finite discrete space or a set of polygons. For example, in a cell decomposition method, a two-dimensional workspace is often divided into \(M \times N\) rectangular cells. Generally, a cell is represented by geometrically simple shape. If the least size of a cell is larger than the size of the robot, we don’t need to take into account the size of the robot in the search space. It is a general problem to choose the resolution of decomposition. A feasible path might not be found if the size of cells is large, while the search space becomes big if the size of cells is small. The size of cells should be adaptively chosen according to the state of search.

In the topological approaches, an environment is represented by a list of connectivities of places. Skeletonization methods directly generate intermediate points and paths, while the cell decomposition methods generate collision-free space. In the skeletonization methods, collision-free paths are basically generated according to the polygonal objects approximated in a workspace. Visibility graph consists of edges connecting visible pairs of vertices of the polygonal objects. In the visibility graph, the shortest path between two points can be generated easily by selecting edges. However, it is dangerous for a mobile robot to move along the generated path, because the path is adjacent to the vertices of the polygonal objects. To overcome this problem, a Maklink graph can be used to generate a safe path. This method can be considered as one of the approximated Voronoi diagrams. In the Maklink graph, a candidate point is represented as a middle point between two vertices, and a path is generated by connecting some intermediate points. Although the generated path is safe, it might not be the shortest.

In our previous works [29], we used image processing to estimate the self-location of the robot based on the cell decomposition method, but it is very difficult to deal with the environmental lighting conditions. Furthermore, the robot must correct the self-location if the difference between the estimated self-location and the actual self-location owing to the inaccuracy of measurement in the rotary encoder. Therefore, we use the measured distance of a laser range finder to estimate the self-location in this paper. The measured distance is used for matching with an environmental map, but this map should be also generated by the robot itself. This problem is well known as a simultaneous localization and mapping (SLAM) [23-26]. In our previous results based on the cell decomposition method, the accuracy of the map building depends strongly the granularity of the map. Therefore, we propose a topological map building method based on a growing neural network as a topological approach. A growing neural network can add neurons and their connections to the network. Furthermore, we apply a steady-state GA (SSGA) to
update the estimated self-position of the robot by using the measured distance and topological map. The proposed method is applied to SLAM in a city hall and a parking area, and we discuss the effectiveness of the proposed method through the experimental results.

This paper is organized as fellows. Section 2 explains the hardware and control method of the mobile robot, a method for topological map building based on growing neural gases and steady-state genetic algorithm. Section 3 shows that the robot can perform simultaneous localization and mapping by using the proposal method.

2. Topological Map Building

2.1. A Mobile Robot

We use Pioneer 2 developed by Active Media Robotics for the map building (Fig.1). This robot is provided with 8 ultrasonic sensors, 2 encoders, and a laser range finder. The laser range finder can measure the distance 50 [m], angle 180°, and the number of the measuring direction is 360 points. The measured distance is used for Simultaneous Localization And Mapping (SLAM). In order to build the environmental map, the robot should perform the wall following, collision avoidance, and target tracing behaviors.

We explain a fuzzy controller for collision avoidance behavior and wall following behavior of the mobile robot with eight ultrasonic sensors. To simplify the problem, two linguistic values of 'dangerous' and 'safe' are used to represent the degree of danger based on the measured distance. A Gaussian membership function is generally described as below,

\[ \mu_{A_i}(x_j) = \exp \left( -\frac{(x_j - a_{ij})^2}{b_{ij}} \right) \]

where \( x_j \) is the distance used as the \( j \)th input; \( b_{ij} \) is width and \( a_{ij} \) and the central position of a membership function. Next, we obtain the \( k \)th resulting output (\( y_k \)) by weighted average as follow,

\[ y_k = \frac{\sum_{i=1}^{r} \mu_{i} \cdot w_{ik}}{\sum_{i=1}^{r} \mu_{i}} \]

where \( w_{ik} \) is the output weight; and \( r \) is the number of rules.

2.2. SLAM

This section describes the historical background of simultaneous localization and map building (SLAM) and related works. Various methods for path planning of a mobile robot have been proposed in intelligent robotics so far. In general, path planning is performed on a built map, but a robot might deal with unknown environment. Sensor-based navigation enables a robot to explore an unknown environment and build a map of the environment. Most of the researches on vision-based navigation have focused on the problem of building full or partial three-dimensional representations of the environment. In order to reduce the search space size, the environmental map can be transformed into various types of search spaces from the visual point of view. However, the robot must know the self-location to update a map. Therefore, a mobile robot requires both of reliable localization and sufficiently precise map to perform navigation tasks in unknown environment. When the robot has an environment map, the robot can estimate the self-location by referring the map. However, when the robot does not have an environment map and does not know the exact self-location, the robot tries to build a map incrementally, while using the same map to estimate the self-location and posture in the environment. This problem is known as SLAM.

2.3. Growing Topological Map Building

Map building can be regarded as one of unsupervised learning approaches where sampling data are noisy and imprecise, because the measurement noise is included in the measured data. Self-organizing map is often applied for extracting a relationship among measured data, since SOM can learn the hidden topological structure from the data [33]. The original SOM used the pre-defined number of nodes. A neural gas have been also used for constructing a topological
map, and furthermore, growing neural gas is used for incremental learning of the topological structure [34-36]. Local error measures are used for determining where to insert new nodes. The competitive Hebbian rule generates the edges between nodes. The addition of nodes and the generation of the edges between nodes can be applied to topological map building. Therefore, we propose a topological map building method based on growing neural gases. We explain the method in the following. At the first measurement, the measurement points are added as the initial nodes of the topological map. Afterward, the topological map is updated according to the measured data. The inputs to the topological map building method is the position \((x, y)\) calculated by the distance from the measured point.

When the \(i\)th reference vector of the topological map is represented by \(r_i\), the Euclidean distance between an input vector and the \(i\)th reference vector is defined as

\[
d_i = \sqrt{v - r_i}^2
\]

Where \(r_i = (r_{1,i}, r_{2,i}, \ldots, r_{N,i})\). Next, the \(k\)th output node minimizing the distance \(d_i\) is selected by

\[
k = \arg\min_i \{d_i\}
\]

Furthermore, the reference vector of the \(i\)th output node is trained by

\[
r_i \leftarrow r_i + \xi \cdot \zeta_{k,i} \cdot (v - r_i)
\]

where \(\xi\) is a learning rate \((0 < \xi < 1.0)\); \(\zeta_{k,i}\) is a neighborhood function \((0 < \zeta_{k,i} < 1.0)\). Accordingly, the selected output node is the nearest point on the topological environmental map.

The number of nodes \(n_{\text{node}}\) is gradually increased when there is no node corresponding to input data. (Fig.2 (a)). We show the procedure of the topological map building:

**Step 1:** Initialization of the map based on the first measurement.

**Step 2:** Distance measurement

**Step 3:** Calculate \(d_i\)

**Step 4:** if \(d_i > D_{\text{max}}\) then \(n_{\text{node}}++\); add \(r_{n_{\text{node}}}\) (Fig.2 (a)) otherwise, update \(r_k\)

**Step 5:** Generate a set \(S_i\) composed of near nodes with respect to the \(k\)th node.

**Step 6:** if the number of nodes in \(S_i\) is larger than the predefined number \(n_{\text{MAX}}\) then the least selected node is removed from the topological map (Fig.2 (b)).

**Step 7:** go to step 2

Figure 3 shows the localization at the robot based on the referring to the topological map. The position and posture of the robot are updated according to the difference between the measured position and the point on the map. This localization is performed by SSGA.

2.4 Steady-state Genetic Algorithm for Localization

As one stream of evolutionary computing, genetic algorithms (GAs) have been effectively used for solving optimization problems in robotics [28-32]. GAs can produce a feasible solution, not necessarily an optimal one, with less computational cost. SSGA simulates the continuous model of the generation, which eliminates and generates a few individuals in a generation (iteration). A candidate solution called an individual is composed of numerical parameters of the revised values to the current position \((g_{i,1}, g_{i,2})\) and rotation \((g_{i,3})\). In SSGA, only a few existing solutions are replaced by new candidate solutions generated by genetic operators in each generation [32]. In this paper, the worst candidate solution are eliminated and
replaced with the candidate solution generated by the crossover and mutation. We use the elitist crossover 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 performed to the generated individual,

\[ g_{ij} \leftarrow g_{ij} + \left( \alpha_j \cdot \frac{f_i - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}} + \beta_j \right) \cdot N(0,1) \]  

where \( f_i \) is the fitness value of the \( i \)th individual, \( f_{\text{max}} \) and \( f_{\text{min}} \) are the maximum and minimum of fitness values in the population; \( N(0,1) \) indicates a normal random value; \( \alpha_j \) and \( \beta_j \) are the coefficient and offset, respectively. In the adaptive mutation, the variance of the normal random number is relatively changed according to the fitness values of the population. Fitness value is calculated by the following equation,

\[ f_{\text{fit}} = \lambda_1 (g_{1,1}^2 + g_{1,2}^2) + \lambda_2 g_{2,1}^2 + \lambda_3 \sum_{k} d_k \]  

where \( d_k \) is the distance between the measured point and its nearest node in the topological map; \( \lambda_1 \), \( \lambda_2 \), and \( \lambda_3 \) are weight parameters for multi-objective optimization. These weight parameters are heuristically determined. Therefore, this problem results in the minimization problem.

3. Experimental Results

We show two experimental results of SLAM. The number of nodes in the map is 500. The population size of SSGA is 50.

3.1. Experimental Results at the hall

We show experimental results of SLAM at a city hall (Fig.4). The workspace size at the experimental environment is \( 30[m] \times 35[m] \). The 90% of the materials in the wall of the city hall is good for the measurement by the laser range finder, but the rest are materials such as curtains. The maximal range of the measurement is 50 [m], but we use the maximal range of 30 [m] because of the reliable measurement. Figure 5 shows (a) the trajectory of the robot and (b) the built map as a result of SLAM at the hall by the proposed method. First of all, the robot moves along the wall according to wall following behavior. Next, the robot takes zigzag running behavior to search the inside of the workspace. When the robot turns, the data contains a lot of noises because ground is a carpet. However, the robot completed SLAM without losing the self-position.

3.2. Experimental Results at A Parking Area

We show experimental results of SLAM at a parking area (Fig.6). There are many cars parked in the parking area. Figure 7 shows a comparison result of the built map without SLAM and the built map of the proposed method. In the experimental results, the robot lost the exact self-position if the SLAM was not
performed (Fig.7 (a)), but the robot can successfully build the map if the proposed method is applied (Fig.7 (b)).

4. Summary

This paper discussed the SLAM of a mobile robot based on computational intelligence. We proposed the topological map building method based on SLAM algorithm by using growing neural gases and a steady-state genetic algorithm. In the experiment at actual hall, the robot moved long distance and rotated many times. As a result, the error of rotary encoder data by slipping was large in the dead-reckoning. However, the map building was successfully done by the proposal method, and the self-position is correctly updated.

As a future work, we intend to perform experiments in the corridor in a large size of floor in order to show the effectiveness of the proposed method. Furthermore, we will develop a topological map building method based on the temporal reliability in a dynamic environment.

References