Occupancy grid map algorithm with neural network using array of infrared sensors

N A Yatim¹, N Buniyamin², Z M Noh¹ and N A Othman³

¹Centre for Telecommunication Research & Innovation (CeTRI), Fakulti Kejuruteraan Elektronik & Kejuruteraan Komputer (FKEKK), Universiti Teknikal Malaysia Melaka (UTeM) Malaysia
²Faculty of Electrical Engineering (FKE), Universiti Teknologi MARA (UiTM) Shah Alam, Selangor, Malaysia
³Faculty of Electrical and Electronics Engineering (FTKEE), Universiti Malaysia Pahang (UMP), Malaysia

Abstract. Occupancy grid map is a map representation that shows the occupancy of spaces, whether there is any object in a particular area or it is a free space. This map representation is also commonly known as a grid map. However, the accuracy of the occupancy grid map is highly dependent on the accuracy of the sensors. In this paper, low cost and noisy sensors such as infrared sensors were used with the occupancy grid map algorithm integrated with a neural network. The neural network was used to interpret adjacent sensor measurements into cell’s occupancy value in the grid map. From the simulation experiments, it is observed that, that neural network-integrated algorithm has a better map estimate throughout robot’s navigation with mean of 28% more accurate compared to occupancy grid map algorithm without neural network. This finding is beneficial for implementation with simultaneous localization and mapping or commonly known as SLAM problem. This is because SLAM algorithm makes use of both estimations of environment’s map and robot’s state. Thus, a better map estimate throughout the robot’s journey can improve a robot’s state estimate as well.

1. Introduction
There are three main elements in autonomous navigation for a mobile robot which are localization, mapping and path planning. In brief, path planning is how a robot makes the shortest or safest trajectory to reach a destination. Localization means to determine the robot’s position in its environment and mapping is to map the environment that the robot is currently at. Localization in an unknown environment, without a priori map, would need the robot to map its environment as well. The combination of these two problems in the literature is known as Simultaneous localization and mapping (SLAM) problem, as illustrated in Figure 1. In this paper, the focus is on the problem of mapping for the autonomous mobile robot.
Figure 1. Elements in autonomous navigation for mobile robot and its combination. This paper focuses on mobile robot mapping problem.

Occupancy grid map is a map representation of the environment surrounding the mobile robot. It has been widely used with SLAM algorithm [1]–[5]. This map representation divides the environment into grid cells. Each cell in the grid map can be either equal in size [4] or with different resolution [6]. These cells represent space occupancy of an area in the map. Occupancy grid map is commonly used for indoor environments [7]. The advantage of this map representation is that the output map is easier to understand compared to landmark-based map. However, grid-based mapping often adopt high-end sensors such as laser range finder. Observation from low-cost sensors is often noisy and inadequate to correctly localize the robot’s position and produce dense map such as occupancy grid map [3].

In this research, low-cost sensors such as infrared sensors were used instead of high-end sensors. This could reduce the overall cost of the mobile robot. The aim is to evaluate the performance of the occupancy grid map algorithm integrated with a trained neural network, $\mathcal{N}$. Overall, in this paper, section 2 describes the methodology used to carry out the simulation experiments, which consists of the simulation setup to test the occupancy grid map algorithm. Consequently, section 3 reports the results of the simulation and the performance of the mapping algorithm. Lastly, section 4 conclude the finding of this paper and suggests future works.

2. Methodology

In this research work, an array of infrared sensor module is chosen as the robot’s perception. This is because infrared sensors have low computation cost and low power consumption. Furthermore, there has been promising works done in mapping and SLAM using infrared sensors, although not specifically in the configuration of an array [8], [9]. The neural network was used to interpret adjacent measurements from infrared sensors into cell’s occupancy value in the grid map. Thus, this section describes the methodology adopted to carry out the experiments and the method used to analyse the performance of the algorithm.

2.1. Occupancy grid map algorithm

The occupancy grid map algorithm is derived from the static state binary Bayes filter. The static state binary Bayes filter is used to estimate a state that has two possible outcomes from a sequence of sensor measurements. Occupancy grid map is an example of binary Bayes filters with the static state, where the outcome is a fixed binary quantity (i.e. a cell can be either occupied or free of obstacles). Static state binary Bayes filter is often adapted when the state of measurements are more complex than a binary state [10]. Thus, in this case, the state of measurements range can take on possible values from zero to maximum range and comes from multiple sensor measurements to estimate the occupancy of each cell which is a binary state.

Algorithm 1 outlined the overall occupancy grid map algorithm. Here, $m_i$ is the $i$th cell in map, $m$, robot’s observation is $z_{1:t}$, and robot’s state is $x_{1:t}$. Line 2 to 4 indicates that only cells that fall in the perceptual field of the current observation $z_t$ are to be updated. Otherwise, the occupancy value of that cell remains the same. The function prob_to_log_odd convert probability value in log odd notation and the inverse, log_odd_to_prob revert back log odd value to probability. The algorithm return the estimate of cell’s occupancy value as $p(m_i|x_t, z_t)$.
Algorithm 1: Occupancy grid mapping

1. **Require**: $l(m_i|x_{1:1-t}, x_{1:t-1}), p(m_i|x_{1:t})$
2. **If** $m_i$ in perceptual field of $x_t$ **then**
   3. $l(m_i|x_{t}, x_t) = \text{prob\_to\_log\_odd} \cdot p(m_i|x_{t}, x_t)$
   4. $l(m_i|x_{1:t}, x_{1:t}) = l(m_i|x_{1:t-1}, x_{1:t-1}) - l(m_i)$
3. **else**
   4. $l(m_i|x_{1:t}, x_{1:t}) = l(m_i|x_{1:t-1}, x_{1:t-1})$
7. **end if**
8. $p(m_i|x_{1:t}, x_{1:t}) = \text{log\_odd\_to\_prob} \cdot l(m_i|x_{1:t}, x_{1:t})$
9. **return** $p(m_i|x_{1:t}, x_{1:t})$

2.2. Mobile robot platform: Khepera III and infrared sensor

In this research, the Khepera III mobile robot is chosen as the robot platform. Khepera III is a differential drive robot produced by K-Team Corporation. The robot is equipped with two types of low-cost sensors; ultrasonic sensors and infrared sensors located on its base model (i.e. without any extension module). The array of infrared sensors on Khepera III is not distributed equally apart due to the physical build of the robot. Figure 2 shows the infrared sensors arrangement and its numbering order in the robot simulator.

![Figure 2. Infrared sensors arrangement for Khepera III robot. The numbering of sensors are as documented in the simulator.](image)

The infrared sensor on Khepera III has a maximum range of 30 cm, as stated in the Khepera III specification. The infrared sensor works on triangulation manner, where the sensor module sends out an LED pulse. Then a receiver, typically a photodiode, will detect where the pulse falls by using optical calculation. The angle varies upon the distance of the object detected. Triangulation has two major implications. The resolution of the sensor is higher for a closer object, lower when it is further away. The second implication is the sensor reading may become spurious when reading is taken while the robot is moving.

2.3. Test environment setup

The size of the environment considered is not too big as a ratio to the robot’s diameter. In autonomous mapping domain, the performance of the intelligent algorithm developed is related to the size of the environment to be mapped [18]. Thus, the size of the environment needs to be considered as a proportion to the Khepera III mobile robot.
In this research, since the Khepera III robot platform has a diameter of approximately 13 centimeter, a comparatively small test arena with uncommon structure was designed. This test arena consists of walls and corners that is not 90 degrees. Figure 3 shows the test arena designed, which has a dimension of 4m × 2m squared area. The environment is inspired by Magnenat et al., work [8]. The environment contains multiple unsymmetrical objects and corners with different angles. Rocks formation is included to mimic objects with irregular shapes.

Figure 3. Test indoor environment with Khepera III mobile robot with corners that is not 90 degrees and unsymmetrical objects.

2.4. Experiment development
To obtain a neural network learner, \(N\) that interprets sensor measurements into grid cells occupancy, the neural network need to be trained with a training data set. For this purpose, the training data set was collected by letting the robot to navigate in a simple squared environment. The robot was set to navigate randomly while avoiding the walls. During navigation, the robot collected its sensor and wheel encoder measurements that will be processed as training input for neural network, \(N\). The output of the training data for \(N\) is the ground truth grid cells values (i.e. 0 or 1) that within the range of sensor measurements.

Originally, the training data set consists of 11 set of data. They are: nine measurements values for each sensor, \(z^k_t, k \in \{1,2,\ldots,9\}\), coordinate of \(j\)th cell, \(\{x_j, y_j\}\), and occupancy value of each cell in the map, \(m_j, m_j \in \{0,1\}\). From this training data set, the neural network was trained to classify its input into the right cell’s occupancy value. In this work, instead of using all sensor measurements as the input to the neural network such as in [11] the training data was preprocessed beforehand to fit with the neural network, \(N\) configuration. The configuration of \(N\) inputs are as follows:

- Four sensors measurements, \(z^k_t, k \in \{1,2,3,4\}\) that are closest to cell, \(m_i\)
- Encoded cell’s position using the distance, \(d_{m_i}\), and angle, \(\theta_{m_i}\).

Figure 4 illustrates the measurements of \(d_{m_i}\) and \(\theta_{m_i}\). The green structure is the array of infrared sensor module on the Khepera III mobile robot. In this illustration, the closest sensor to cell \(m_i\) is sensor three, thus, the position of the cell \(m_i\) is encoded using the distance between \(m_i\) and sensor three, \(d_{m_i}\), and the angle of \(m_i\) to the same sensor, \(\theta_{m_i}\).
Figure 4. The position of cell $m_i$ is encoded using $d_{m_i}$ and $\theta_{m_i}$.

After the data set was pre-processed, neural network, $\mathcal{N}$ was trained with the training data set. The training was done via MATLAB neural network toolbox. Figure 5 shows the configuration of the neural network, $\mathcal{N}$.

In the second part of the experiment, the Khepera III mobile robot was set to navigate through the test environment in Figure 3. Since the maximum range of the infrared sensor is only 30 cm, to get the maximum observation, the robot had to make multiple loops in the test environment. To build the map, the occupancy grid map algorithm in section 2.1 was implemented where the output of the neural network, $\mathcal{N}$ was used as the probability of cell’s occupancy, $p(m_i|x_t,z_t)$.

At each time step, the neural network, $\mathcal{N}$ will build a local occupancy grid map, $\mathcal{G}$, with $n \times n$ cells as depicted in Figure 5. The size of each cell was set to 2 cm$^2$. The size of $\mathcal{G}$ was determined based on the maximum range of infrared sensor used, which is 30 cm. In order to obtain a local grid map $\mathcal{G}$ that can cover the maximum measurements, a squared local grid map with the size of 73 cm$^2$ was built at each time step. As the diameter of Khepera III is 13 cm and the maximum range of each side of the robot is 30 cm. Thus, the size of $\mathcal{G}$ in cell unit was approximately $36 \times 36$ cells, as illustrated in Figure 6.

Figure 6. Local occupancy grid map, $\mathcal{G}$ with $n = 36$.

2.5. Method for performance analysis
To evaluate the accuracy of the map estimate, two methods of analysis are adopted in this research. Firstly, is to compare the resulting maps qualitatively from the map details. Secondly, is to compare the
resulting maps quantitatively by using a fitness score and a completeness measure adapted from Moshiri et al. in [11] and Lee et al. in [12] respectively.

The fitness score in (1) is used to evaluate occupied cells accuracy, while the completeness measure in (2) is used to evaluate empty cells accuracy. Different methods are adopted because occupancy value for each cell is not either 0 or 1, but rather a probability value of 0 until 1. For occupied cells, grid cells’ values of the ground truth grid map, \( M_{\text{truth}} \), are compared with grid cells of the map estimate, \( M \). The sum of the differences is divided by the number of occupied cells compared, \( N_{\text{occ\_cells}} \). A perfect match will result to a score of 1, while a completely wrong map will result in a score of 0.

\[
f_{\text{occ}}(M,M_{\text{truth}}) = 1 - \frac{\sum_{c \in \text{occ\_cells}} |M(c) - M_{\text{truth}}(c)|}{N_{\text{occ\_cells}}}
\]

(1)

As for free cells, the probability of having unoccupied cells are higher in each sweep of the sensor reading. Thus, an empty cell should have a value very close to 0 rather than some occupancy value in it. The completeness measure formulation in equation (2) termed as \( M_{R_{\text{free}}} \), is adapted. An indicator function \( I \) in (3) is used to indicate whether the cell occupancy value is close to 0. Here, the threshold, \( oc_{c}\_th \) is set to a small value of 0.05.

\[
M_{R_{\text{free}}}(M,M_{\text{truth}}) = \frac{\sum_{c \in \text{free\_cells}} I(M(c))}{N_{\text{free\_cells}}}
\]

(2)

\[
I(t) = \begin{cases} 
1 & \text{if } 0 < t < oc_{c}\_th \\
0 & \text{otherwise}
\end{cases}
\]

(3)

To evaluate the map estimate as a whole, the same fitness function to evaluate occupied cells is used. Here, all cells in map estimate are taken into account in the fitness function, as stated in Equation (4).

\[
f_{\text{map}}(M,M_{\text{truth}}) = 1 - \frac{\sum_{c \in \text{all\_cells}} |M(c) - M_{\text{truth}}(c)|}{N_{\text{all\_cells}}}
\]

(4)

In order to analyze the increase in performance by integrating neural network, \( \mathcal{N} \), the same data set was apply to the occupancy grid map algorithm alone. This algorithm used beam endpoint model where the cell at beam endpoint is marked as occupied and the rest of the cells are marked as free cells. At the end, the accuracy of both maps was compared and analyzed.

3. Results and analysis

In the first part of the experiment, the neural network, \( \mathcal{N} \) was trained with the input configuration described in the experimental development section. The training was executed using the neural network tool in MATLAB, and managed to obtain an MSE value of 0.0965.

To evaluate the performance of the occupancy grid map algorithm with and without neural network, the accuracy of the global maps was analyzed. Figure 7 and 8 show the resulting maps. Both algorithms used the same dataset from the array of the infrared sensor. The green line indicates the robot’s trajectory when navigating in the test environment.

Visually, it can be seen the global map that was built by the occupancy grid map algorithm alone managed to capture all walls and two objects in the test environment (refer to Figure 7). However, there was an incorrect mapping where there was a visible line at the center of the map. This should be empty spaces instead. On the other hand, in the global map by the neural network, \( \mathcal{N} \) integrated algorithm (see Figure 8), there was no such incorrect mapping occurred. The overall structure of the environment
managed to be mapped adequately. However, there were a few areas of the environment that were not mapped clearly. Such as corners on the right side of the test environment. It is observed though, that at this area, the distance of the wall and robot’s trajectory is larger compared to walls that are mapped correctly. The larger distance has caused the neural network, $\mathcal{N}$ not to be able to interpret the measurements accordingly.

**Figure 7.** Global map of occupancy grid map algorithm without neural network (XNN).

**Figure 8.** Global map of occupancy grid map with neural network (NN) algorithm.

To further analyse the global maps, the accuracy of both maps was compared by using the fitness function, and MR function in equation (1) to (4). Figure 9 shows the result of $f_{occ}$, $MR_{free}$ and $f_{map}$ scores of both global maps at the end of the robot’s trajectory. In this graph, the global map by occupancy grid map algorithm with a neural network is termed as global map NN and the global map of the same algorithm without a neural network is termed as global map XNN. From figure 9, it can be observed that both algorithms have approximately the same score for $f_{occ}$, $MR_{free}$ and $f_{map}$. However, for $f_{occ}$ which reflects on accuracy of occupied cells the integrated neural network, $\mathcal{N}$, algorithm produces a global map with about 10% more accurate.

**Figure 9.** Fitness and MR scores for occupancy grid map algorithm with and without neural network at the end of the robot’s trajectory.

Although at the end of robot’s exploration, significant improvement in map accuracy was not detected, however, in application such as in Simultaneous Localization and Mapping (SLAM) algorithm it is important to have a consistent map accuracy throughout robot’s exploration. This is because at each time step, the map estimate is used to estimate the robot’s state position simultaneously. Thus, the accuracy of the map estimate is plot throughout the robot’s trajectory, as shown in Figure 10.
Figure 10. Global map score using neural network (NN) and without neural network (XNN) algorithm during robot’s exploration time.

From Figure 10, it can be observed that the score of a neural network-integrated algorithm for occupied cells, free cells and overall global map estimate consistently higher than the algorithm without neural network. In fact, the mean score for occupied cells shows that neural network-integrated algorithm has 28% more accurate than the occupancy grid map algorithm without neural network. This is a significant finding for this research because the SLAM algorithm makes used of occupied cells value in computing measurement likelihood. Furthermore, in the SLAM algorithm, the map estimate is used concurrently in order to estimate both map and robot’s state estimate. Thus, a better map estimate throughout the robot’s journey can improve the robot’s state estimate as well.

4. Conclusion and future works
In this paper, the objective is to investigate the performance of the occupancy grid map algorithm integrated with a neural network by using infrared sensors measurements. A neural network is used to interpret adjacent sensor measurements into a grid cell’s occupancy. The performance of the occupancy grid map algorithm with a neural network is then compared with the performance of the occupancy grid map algorithm without neural network. It is found out that having a neural network interpreting the adjacent sensor measurements gives a more consistent accuracy throughout the robot’s exploration, where the mean of maps accuracy is 28% higher. However, due to mapping error in different parts of the environment, the final map accuracy does not show significant improvement.

For future works, the neural network-integrated algorithm will be tested with the SLAM problem where algorithm should estimate the robot’s state as well considering errors from wheel encoder of the mobile robot or dead reckoning.

Acknowledgments
The authors would like to thank the Centre for Telecommunication Research & Innovation (CeTRI), Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer (FKEKK), Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Universiti Teknikal Malaysia Melaka (UTeM) and the Ministry of Higher Education for moral and operational support throughout the project.

References
[1] C. Stachniss and H. Kretzschmar, “Pose graph compression for laser-based SLAM,” in Robotics Research, Springer, 2017, pp. 271–287.
[2] C. Georgiou, S. Anderson, and T. Dodd, “Constructing informative Bayesian map priors: A multi-objective optimisation approach applied to indoor occupancy grid mapping,” Int. J. Rob. Res., vol. 36, no. 3, pp. 274–291, 2017.
[3] N. M. Yatim and N. Buniyamin, “Indoor Mapping with Machine Learning Algorithm using
Khepera III Mobile Robot,” *J. Telecommun. Electron. Comput. Eng.*, vol. 8, no. 9, pp. 61–66, 2016.

[4] C. Schröter, H.-J. Böhme, and H.-M. Gross, “Memory-Efficient Gridmaps in Rao-Blackwellized Particle Filters for SLAM using Sonar Range Sensors.,” in *EMCR*, 2007.

[5] B. Ristic, D. Angley, D. Selvaratnam, B. Moran, and J. L. Palmer, “A random finite set approach to occupancy-grid SLAM,” in *Information Fusion (FUSION), 2016 19th International Conference on*, 2016, pp. 935–941.

[6] E. Einhorn, C. Schröter, and H.-M. Gross, “Finding the adequate resolution for grid mapping-cell sizes locally adapting on-the-fly,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1843–1848.

[7] B. Hampton, A. Al-Hourani, B. Ristic, and B. Moran, “RFS-SLAM robot: An experimental platform for RFS based occupancy-grid SLAM,” in *Information Fusion (Fusion), 2017 20th International Conference on*, 2017, pp. 1–8.

[8] S. Magnenat *et al.*, “Affordable slam through the co-design of hardware and methodology,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 5395–5401.

[9] B. G. Sileshi, J. Oliver, R. Toledo, J. Gonçalves, and P. Costa, “Particle Filter SLAM on FPGA: A Case Study on Robot@ Factory Competition,” in *Robot 2015: Second Iberian Robotics Conference*, 2016, pp. 411–423.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[11] B. Moshiri, M. R. Asharif, and R. HoseinNezhad, “Pseudo information measure: A new concept for extension of Bayesian fusion in robotic map building,” *Inf. Fusion*, vol. 3, no. 1, pp. 51–68, 2002.

[12] K. Lee and W. K. Chung, “Effective maximum likelihood grid map with conflict evaluation filter using sonar sensors,” *Robot. IEEE Trans.*, vol. 25, no. 4, pp. 887–901, 2009.