Indoor Mapping with Machine Learning Algorithm using Khepera III Mobile Robot

Norhidayah Mohamad Yatim[1,2], Norlida Buniyamin[2]
[1]*Department of Computer Engineering, Faculty of Electronic and Computer Engineering,*
Universiti Teknikal Malaysia Melaka (UTeM), Malaysia.
[2]*Department of System Engineering, Faculty of Electrical Engineering.*
Universiti Teknologi Mara (UiTM), Malaysia.
hidayah.yatim@gmail.com

***Abstract*—Small robots can be beneficial in many applications as they have the advantage of reaching small spaces. For these robots to be truly autonomous, ability to map their surrounding is essential. Accuracy of mapping is related closely to sensor's precision. However, small robots can only be equipped with small sensor that is typically has noisy characteristic with cheaper cost, such as sonar sensor and infrared sensor. To enhance the quality of map build by noisy and low-cost sensor, machine learning algorithm integration is a good approach. In this work, multiple learners, which are Naïve Bayes, Decision Tree, Neural Network and AdaBoost, were experimented with occupancy grid map algorithm using Khepera III robot platform. Then, the results of their fitness score according to the maps build were compared. The results show that Neural Network performed the best with the occupancy grid map algorithm.**

***Index Terms*—Indoor Mapping; Khepera III; Machine Learning; Occupancy Grid Map.**

I. INTRODUCTION

One of the essential challenges in robotics is to reduce robot's size as well as its cost. Small robots have the advantage of reaching and exploring narrow places to perform any specific task. Small or miniature robots have variety of applications in domestic, industrial or humanitarian field. This includes inspection system, medical, cleaning, mowing, de-mining and even in search and rescue (SAR) efforts to name a few.

One possible area is inspection of aircrafts' turbines that are highly exposed to tear-and-wear condition. A cost effective inspection system can be done by using miniature robots [2]. These robots have to be very small to fit in the aircraft's turbine to inspect the blades inside.

In domestic cleaning, robotic vacuum cleaners have gained significant accomplishment. Although these robots are not miniature, they are considerably small. It is reported more than 2 million units of domestic service robots are sold every year from 2012 and approximately 95% of the sold units are robotic vacuum cleaners [3]. Vacuum cleaner robots have become more intelligent with the ability to memorise the area that it has explored. This allows the robot to clean more efficiently.

A riskier application of mini robots is their involvement in search and rescue mission. A very unfortunate event of tsunami in Tohoku region in Japan had claimed many lives. In the event of locating survivor in closed space, such as collapsed building, underground passages and tunnels, where rescuers cannot get closer to the rubbles due to various reasons, mini robots would be an obvious alternative to do the job. In response to the Tohaku disaster, two types of robots on ground and air were designed to track victims in SAR mission, particularly in underground malls [4].

A majority of researches in the area of mini robot applications focus on the design and the development of the robot such as, the method of locomotion, sensing and communication. Thus, works that discuss mini robots and their capability to execute autonomous behaviour such as localisation, navigation, mapping and path planning are often based on either commercialised mini robots or custom-made for education and research.

For a robot to be truly autonomous, it needs to be aware of its environment. To acquire this information, robot essentially has to know where it is in the environment and how the environment looks like. Concisely, there are three main tasks for a robot to learn efficiently about its environment, which are mapping, localisation and path planning [5].

In this research work, we focus on mapping the problems of small robot. Our approach, adopting machine-learning method is described to obtain map using a ring of infrared sensor on a small mobile robot platform.

Section II describes previous works in mapping problem using small robot with low-cost sensor. Section III describes the simulation setup of this research, such as the robot platform and range model used for robot's perception. Section IV elaborates the utilised of mapping algorithm, including the machine learning method adopted for comparison. The results and discussion are included at the end of this section. Lastly, Section V concludes this paper and suggests future development.

II. PREVIOUS WORKS

A. Mapping with small robot

Mapping the environment using small robot has been done previously by multiple researchers [6]–[9]. In these works, two types of map representation were used; line segment map [6] and occupancy grid map [7]–[9] representation.

Grid-based map representation is a metric map, where the map of environment is divided into grid cells. The occupancy value of each cell depends on the measured distance, where the cells at the end of the sensors' rays are considered occupied (i.e. mark as black cells) and the cells in between are considered free (i.e. mark as white cells). While, line segment map uses algorithm, such as Hough-transform [6], [10], [11], Region of Constant Depth (RCD) [12], and iterative end point fit [13] are needed to extract line features from the sensors' measurements.

Feature-based map and occupancy grid map have their advantages and disadvantages. Feature based-map is a more compact map representation, but it needs to make certain assumption on the environment. For example, some of the works made orthogonality or geometric assumption of the environment[11], [13]. On the other hand, occupancy grid map is more flexible but with high memory consumption. Table 1 summarises the advantage and disadvantages of feature-based map and grid map.

Table 1
Comparison of feature-based map and occupancy grid map algorithm

	Feature Based Map		Occupancy Grid Map
+	Compact map representation	-	Memory consumption increases with resolution of grid cells
-	Make assumption on structure of features in environment	+	Make no assumption on features of the environment
-	Need data association to landmarks	+	No data association to landmarks needed

In this research, we adapted occupancy grid map representation, due to its flexibility of making no assumption on the feature of the environment. Thus, the mapping algorithm can be used in the application, such as inspection, where actual condition of the environment is required for evaluation.

B. Machine Learning Integration

Small robots have limitation on the size of sensors that can be equipped onboard. For high-end sensor, such as laser rangefinder, using occupancy grid map algorithm alone can produce a good quality map. However, laser rangefinders are bulky and not suitable for small robots.

Among sensors that are small in size are sonar sensors and infrared sensor. These sensors are much lower in cost, but have significant noise in their sensors' measurements.

Implementation of low-cost sensor could not be treated the same way like the high-end sensor implementation in grid-based mapping. The sensor measurements need to be pre-processed to gain a better map accuracy. For example, sonar sensors' reflections can be misinterpreted due to many reasons.

Machine learning method is advantageous for interpreting noisy sensor measurements. Neural network learner was used in [1] to evaluate multiple adjacent sonar sensors. In [14], neural network was also used with infrared rangefinder to produce a more accurate map.

Other than Neural Network, there are many algorithms developed in machine learning domain. Machine learning has evolved greatly in the past decade. With cheaper Graphical Processing Unit (GPU) that is available in the market, high computation cost of machine learning methods are still manageable.

In order to interpret noisy sensor measurements to produce an occupancy grid map, one way is to train a learner to classify sensors data into a grid cell's occupancy[1]. This problem can be treated as classification method.

However, instead of using the classification value (i.e. 0 for free cell and 1 for occupied cell), the probability of the output is used. The probability value is then fed into the occupancy grid map algorithm.

In this work, there are multiple algorithms that were experimented to learn the sensor inputs. They are Naïve Bayes, Decision Tree, AdaBoost (for adaptive boosting) and Neural Network learners.

III. Simulation Setup

A. Robot Platform

In this research, we used Webots robot simulator, which is a simulator developed by Swiss Federal Institute of Technology, Lausanne (EPFL). We selected Khepera III mobile robot from the many robot prototypes provided (see Figure 1). Khepera III is a differential drive robot produced by K-Team Corporation. Khepera III satisfies the criteria of being small with the size of approximately 12 cm diameter and it is equipped with two types of low-cost sensor. They are five ultrasonic sensors and nine infrared sensors. In this research work, we used the array of infrared sensor because infrared sensor has smaller beam characteristic compared to sound wave by ultrasonic sensor. The array of infrared sensors is not equally apart due to the physical build of the robot.

Figure 2 shows the infrared sensors arrangement and its numbering order in Webots robot simulator. There is a newer version of the robot, Khepera-IV. However, this robot bears a higher cost.

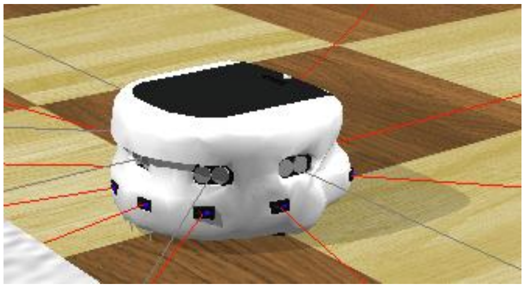


Figure 1: Khepera III mobile robot in Webots robot simulator

B. Range Model for infrared sensor

In occupancy grid map algorithm, the quality of map is highly related to the sensor's characteristic. Thus, it is important to analyse the characteristic of the sensor.

The measurement of infrared sensor in Webots simulator depends on the reflectance on colour properties of the object's surface. The infrared sensor equipped on Khepera III is a TCRT5000 reflective optical sensor from Vishay Telefunken. It has a maximum range of 30 cm.

To analyse the characteristic of the sensor, we changed the location of a white plate in front of the sensor mounted on the mobile robot. Measurements were taken at every 1 cm away from the sensor, for distances of 0 to 30 cm.

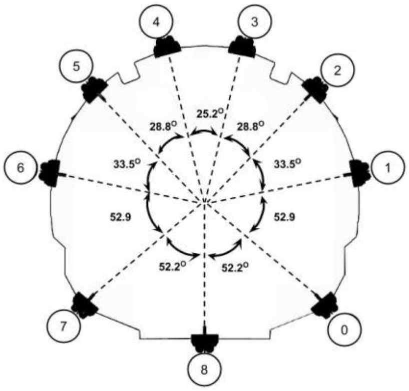


Figure 2: Infrared sensors arrangement for Khepera III robot. The numbering of sensors is as documented in Webots simulator.

For each distance, 10,000 measurements were collected. The sensor measurements at distance of 2 cm were plotted on a histogram in Figure 3. It can be seen that the noise of the sensor's measurements is set to follow Gaussian distribution in Webots simulator. This shows that the simulator uses a fairly realistic infrared noise model.

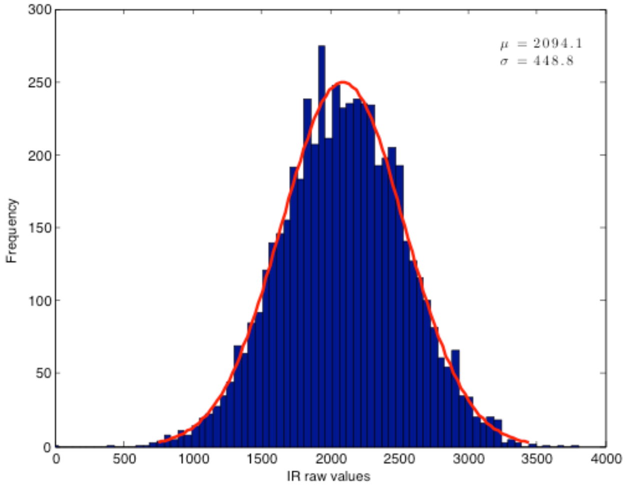


Figure 3: Histogram fit of 5500 measurements at a distance of 2cm.

After the initial calibration, a range model implementation was adapted from the work by Proroks et al. [15]. For each set of measurements with distance in range $R = \{0,1,\ldots,30\text{cm}\}$, mean, μ_i and standard deviation, σ_i, where $i \in \{R\}$, were determined.

In order to elaborate the sensor values, an ensemble Ω are build from sensor values obtained at R. The ensemble Ω is defined as Equation (1). We denote v_i and d_i as the measurements value and actual distance at each R. Then, the ensemble Ω were created by sampling S samples from a normal distribution, $\mathcal{N}(\mu_i, \sigma_i)$, which reflect the distribution of the infrared sensor. Each sample is denoted by v_j.

$$\Omega = \bigcup_{i \in R} \{(v_j, d_i) | v_j \sim \mathcal{N}(\mu_i, \sigma_i), j \in [0 \cdots S]\} \tag{1}$$

In order to obtain the information with respect to sensor values' axis, v, (i.e. $\mu(v)$ and $\sigma(\text{v})$) a rectangular function, $\prod$, (see Equation (2)) was applied to the set of points in Ω.

$$\prod(t) = \begin{cases} 0 & \text{if } |t| > \frac{1}{2} \\ 1 & \text{if } |t| < \frac{1}{2} \end{cases} \tag{2}$$

We defined a set of values for $v \in \{0, 50, \cdots, 4000\}$, and selected N points for each v from Ω using a sliding window, $w = 40$ as in Equation (3).

$$N = \sum_{(v_j, d_i) \in \Omega} \prod\left(\frac{v_j - v}{w}\right) \tag{3}$$

These selected points were used to calculate the mean, $\mu(v)$, and standard deviation, $\sigma(v)$, of each v value. Equation (4) and (5) are used for this purpose.

$$\mu(v) = \sum_{(v_j, d_i) \in \Omega} \frac{d_i \cdot \prod\left(\frac{v_j - v}{w}\right)}{N} \tag{4}$$

$$\sigma(v) = \sqrt{\frac{1}{N} \sum_{(v_j, d_i) \in \Omega} \left(d_i - \mu(v)\right)^2 \cdot \prod\left(\frac{v_j - v}{w}\right)} \tag{5}$$

Figure 4 shows the result of infrared sensor range model from the method described. A polynomial fit is performed on the mean values and obtained $R^2 = 0.9888$. It is noted that an exponential decay fit can also be used to obtain the range model.

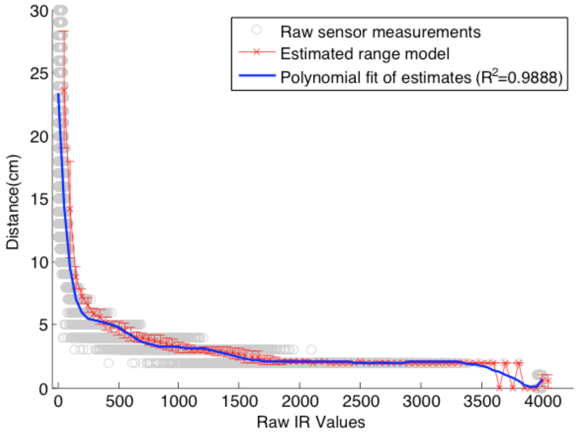


Figure 4: Estimated range model for infrared sensor obtained from raw values in grey points. The standard deviations are marked by error bars.

IV. Methodology

A. Machine learning integration

From the sensor characteristic of infrared sensor, it can be seen that the range model is highly non-linear due its noisy sensor characteristic.

Using the machine learning method is advantageous. We interpreted multiple sensors reading concurrently, rather than interpreting the range measurement of infrared sensor independently. By doing this, it allows to capture more information for sparse sensor, such as an array of infrared sensors [1], [16].

To use the machine learning method, we need a training data to train the learner. For this, we use a simple squared environment with multiple objects and let the robot run randomly while collecting the data of the sensor and the corresponding occupancy cell values. Figure 5 shows the occupancy grid map of our training environment.

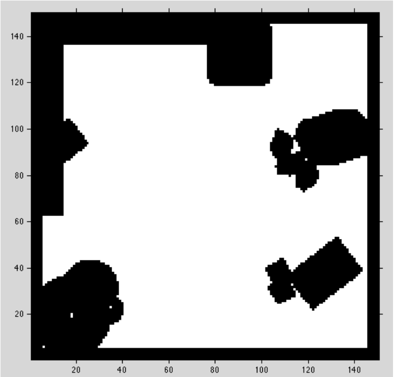


Figure 5: Occupancy grid map for the training purpose. Black cells are occupied area and white cells are free area.

The machine learning algorithm determines the decision surface that separates a class with another class so that a new data input can be classified accordingly. Thus, a machine learner takes data and transforms it into a decision surface. Firstly, we wanted to visualise the data to see whether there is a good chance that the data could be classified into 0 and 1.

For this, we determined the input of our classifier based on a method in [1]. The author used neural network to determine the probability of cell occupancy with the following inputs:

- Four sensors measurements, $\tilde{z}_t^k, k \in \{1,2,3,4\}$ that are the closest to a particular cell at x and y position.
- Encoded position of cell using the distance, $d_{x,y}$ and angle, $\theta_{x,y}$, of cell to the closest sensor with respect to the robot position

In order to visualise the data, we reduced the dimensionality of the input. Regression method was used, where the four closest sensor inputs a feed to a regression to obtain four coefficients (i.e. $\alpha_1, \alpha_2, \alpha_3$, and α_4) leading to a single value, z_t. z_t was computed using (6), where c is a constant value.

$$z_t = c + \sum_{i}^{4} \alpha_i \tilde{z}_t^i \tag{6}$$

The training data was visualised in three dimensional with sensor regression output, z_t, $d_{x,y}$, and $\theta_{x,y}$ as showed in Figure 6. In Figure 6, the red circles are cells with value 1 and blue circles are cells with value 0. By observation, it can be seen that there is a decision boundary that can be obtained where the upper area are cells with value 1 and the bottom area is occupied by cells with value 0. However, in the middle area, there are overlapping data between the two classes.

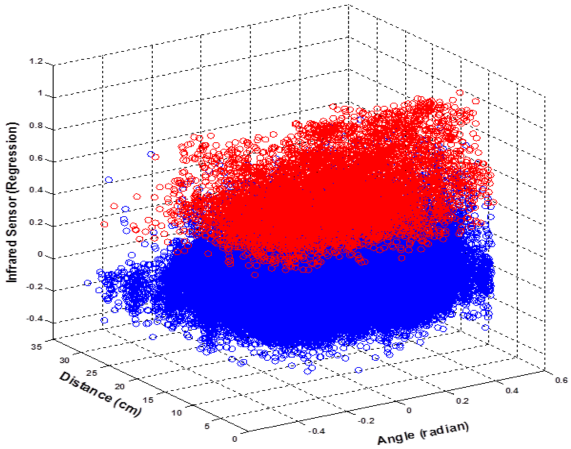


Figure 6: Train data visualization in 3 dimensional view, the red circles are cells with value 1 and blue circles are cells with value 0

We then treated this problem as classification problem with machine learning. As mentioned before, the supervised learning methods that were experimented were Naïve Bayes, Decision Tree, AdaBoost and Neural Network. The inputs are z_t, $d_{x,y}$, and $\theta_{x,y}$ and output is the probability of cell's occupancy. Figure 7 shows the input and output structure for Neural Network configuration. The probability of cell occupancy is denoted as $p(m_i|x_t, z_t)$, which describes the probability of cell m_i is occupied given the robot's current position, x_t and latest sensor measurements, z_t. Here, m_i is the ith cell in map, m. Function $f(\cdot)$ is the regression function in (6).

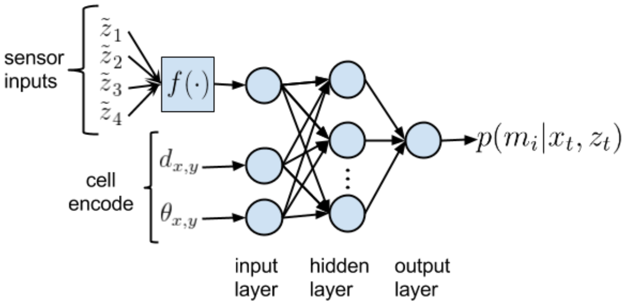


Figure 7: Neural network input output configuration. Here, f(·) is the regression function in Equation (6).

B. Occupancy grid map

Occupancy grid map algorithm was first developed by Moravec and Elfes in the 80's [17]. The algorithm is derived from the static state binary Bayes filter. Here, the probability form was converted to log odds notation stated in Equation (7) for computation efficiency. Note that, the probability $p(m_i|x_t, z_t)$ obtained from the machine learning algorithm was used as the probability value.

$$l(x) = \log\left(\frac{p(x)}{1-p(x)}\right) \tag{7}$$

By using this substitution, the log odd notation form of each cell in the grid map was computed using Equation 8. $l(m_i|z_{1:t}, x_{1:t})$ denotes the log odd value of m_i given all robot's observation, $z_{1:t}$ and all robot's state $x_{1:t}$. The last term in Equation 8 is the initial value of the cell, m_i. The initial probability is set to 0.5. To get the occupancy probability, the log odd value, $l(m_i|z_{1:t}, x_{1:t})$ is then converted back to probability using equation 9, which is the inverse of Equation 7.

$$\begin{aligned} l(m_i|z_{1:t}, x_{1:t}) = l(m_i|z_t, x_t) + l(m_i|z_{1:t-1}, x_{1:t-1}) \\ - l(m_i) \end{aligned} \tag{8}$$

$$p(x) = 1 - \left(\frac{1}{1+e^{l(x)}}\right) \tag{9}$$

V. SIMULATION

A. Results

After implementing the method described in the previous section on Khepera III mobile robot, the robot was set to run through environment in Figure 8. This unstructured environment is inspired by Magnenat et al. work [8]. The environment contains multiple unsymmetrical objects and corners with various angles.

It is logical that the larger the environment relative to the robot's perception range, the more challenging for it to acquire a map. Thus, to get maximum observation due to the short range of infrared sensor, the robot has to do multiple loops in the test environment.

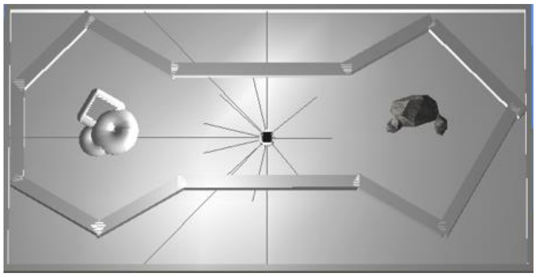


Figure 8: Test indoor environment with Khepera III mobile robot.

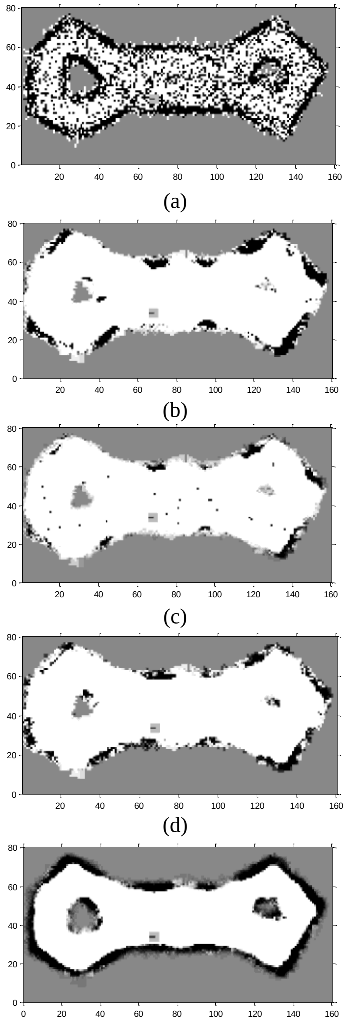
Figure 9 (b)-(e) shows the resulting map obtained with Naïve Bayes, Decision Tree, AdaBoost and Neural Network learners respectively. To compare the performance of each learner that integrates with the occupancy grid map algorithm described in previous subsection, a fitness function in equation 10 is used. Grid cells' values of the ground truth grid map, M_{truth}, are compared with grid cells of the map obtained using machine learning method, M. The sum of difference is divided by the number of cells compared, N_{cells}. This value is deducted from 1 to obtain the fitness score. A perfect match will result in a score of 1, while a complete wrong map will result in a score of 0.

$$f(M, M_{\text{truth}}) = 1 - \frac{\sum_{c \in \text{cells}} |M(c) - M_{\text{truth}}(c)|}{N_{\text{cells}}} \tag{10}$$

Table 2 shows the result of each learner. The first column shows Mean Squared Error (MSE) of each learner in the training phase. To calculate the MSE, we separated our collected data into train and test portion. The next column shows the fitness scores that each learner obtained. There are two fitness scores; 1) for all free cells and occupied cells in sensors' range, 2) for only occupied cells in sensors' range. The last column is the average computation time of each learner.

B. Discussion

From Table 2 fitness score, it can be observed that the difference is not apparent on the fitness score of all detected cells. However, the maps obtained in Figure 9(b)-(e) show that Neural Network has far more accuracy than the other three learners.



(a)

(b)

(c)

(d)

(e)

Figure 9: Resulting occupancy grid maps of different machine learning algorithm using 2.5cm2 grid cells. (a) without any machine learning algorithm, only occupancy grid map algorithm. (b) Naïve Bayes learner, (c) Decision Tree learner (d) Adaboost learner (e) Neural Network learner

Table 2
Performance of Machine Learning Algorithm

Method	Mean Squared Error (MSE)	$f(M, M_{\text{truth}})$		Time step (second)
		all	occupied	
Naïve Bayes	0.1504	0.7822	0.2879	1.23
Decision Tree	0.1227	0.7577	0.1776	1.13
Neural Network	0.1269	0.8641	0.7410	1.43
AdaBoost	0.1169	0.7814	0.2695	2.59

This can be seen quantitatively from the fitness score of only occupied cells, where the score of Neural Network is the highest among the learners and followed by Naïve Bayes and AdaBoost which had about 30% accuracy. Meanwhile, Decision Tree had the lowest performance among the learners. Another observation was that only Neural Network learner can detect the rock formation on the right area of the environment. However, objects on the left compartment were still vague.

We analysed the training data using the Neural Network classifier.

Figure 10 shows the training data that are classified using Neural Network. The data points with marker × and o of the same colour (i.e. red or blue) showed data points that are classified correctly. While, the data points that have both red and blue markers are data points that are not classified correctly. It can be observed that some of the data with label 1 are interpreted as 0 on the red area. Neural Network is naturally probabilistic, which explains why it has better performance compared to the other learners.

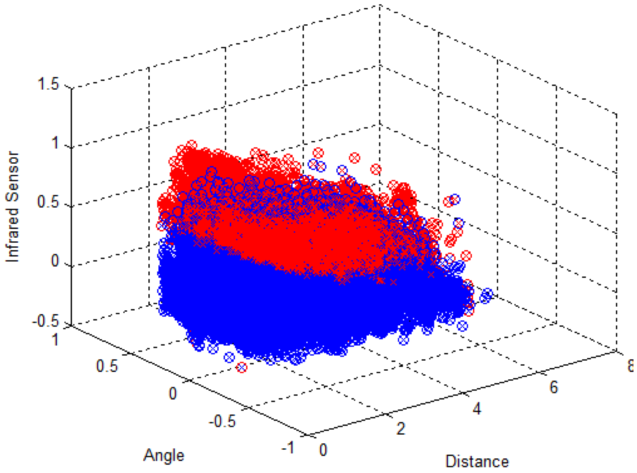


Figure 10: Classification using Neural Network. Circles and x marks with the same color (i.e. red or blue) shows data that is interpreted correctly by Neural Network. While circle and x marks that are not the same color show data that is misclassified by Neural Network

VI. Conclusion and Future Works

Although the training of neural network learner could not manage to get a lower MSE value (i.e. MSE < 0.01), by integrating this learner with occupancy grid map, we managed to get an adequate map. Compared to the map without any learner (see Figure 9(a)), machine learning is a good approach as a probabilistic classifier. The drawback of the learner is the consumption of significant time due to computation cost. Table 2 shows that all the learners took more than 1 second for a time step. This is due to the computation of each learner on each cell in the grid. Smaller grid size will result in more cells to be computed, hence more execution time of the machine learner. However, further optimisation can be done in order to produce a more accurate and faster learner for mapping with infrared sensor.

References

[1] S. Thrun, “Learning metric-topological maps for indoor mobile robot navigation,” *Artif. Intell.*, vol. 99, no. 1, pp. 21–71, 1998.

[2] N. Correll and A. Martinoli, “Multirobot inspection of industrial machinery,” *Robot. Autom. Mag. IEEE*, vol. 16, no. 1, pp. 103–112, 2009.

[3] F. C. Vaussard, “A Holistic Approach to Energy Harvesting for Indoor Robots,” 2015.

[4] N. Shimoi, Y. Takita, and H. Madokoro, “Development of a wheel robot and micro fling robot using for rescue scenarios,” *Am. J. Remote Sens.*, vol. 1, pp. 61–66, 2003.

[5] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, “An experiment in integrated exploration,” in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, 2002, vol. 1, pp. 534–539.

[6] F. Abrate, B. Bona, and M. Indri, “Experimental EKF-based SLAM for Mini-rovers with IR Sensors Only.,” in *European Conference on Mobile Robots*, 2007.

[7] C. M. Gifford, R. Webb, J. Bley, D. Leung, M. Calnon, J. Makarewicz, B. Banz, and A. Agah, “Low-cost multi-robot exploration and mapping,” in *Technologies for Practical Robot Applications, 2008. TePRA 2008. IEEE International Conference on*, 2008, pp. 74–79.

[8] S. Magnenat, V. Longchamp, M. Bonani, P. Rétornaz, P. Germano, H. Bleuler, and F. Mondada, “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] P. Tarquino and K. Nickels, “Programming an E-Puck Robot to Create Maps of Virtual and Physical Environments,” in *Robot Intelligence Technology and Applications 2*, Springer, 2014, pp. 13–28.

[10] J. D. Tardós, J. Neira, P. M. Newman, and J. J. Leonard, “Robust mapping and localization in indoor environments using sonar data,” *Int. J. Rob. Res.*, vol. 21, no. 4, pp. 311–330, 2002.

[11] T. N. Yap and C. R. Shelton, “SLAM in large indoor environments with low-cost, noisy, and sparse sonars,” in *IEEE International Conference on Robotics and Automation (ICRA), 2009.*, 2009, pp. 1395–1401.

[12] J. J. Leonard and H. F. Durrant-Whyte, *Directed sonar sensing for mobile robot navigation*, vol. 448. Kluwer Academic Publishers Dordrecht, 1992.

[13] Y.-H. Choi, T.-K. Lee, and S.-Y. Oh, “A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot,” *Auton. Robots*, vol. 24, no. 1, pp. 13–27, 2008.

[14] Y.-S. Ha and H.-H. Kim, “Environmental map building for a mobile robot using infrared range-finder sensors,” *Adv. Robot.*, vol. 18, no. 4, pp. 437–450, 2004.

[15] A. Prorok, A. Arfire, A. Bahr, J. R. Farserotu, and A. Martinoli, “Indoor navigation research with the Khepera III mobile robot: An experimental baseline with a case-study on ultra-wideband positioning,” in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, 2010, pp. 1–9.

[16] N. M. Yatim and N. Buniyamin, “Development of Rao-Blackwellized Particle Filter (RBPF) SLAM Algorithm Using Low Proximity Infrared Sensors,” in *9th International Conference on Robotic, Vision, Signal Processing and Power Applications*, 2017, pp. 395–405.

[17] H. P. Moravec, “Sensor Fusion in Certainty Grids for Mobile Robots,” *AI Magazine*, vol. 9, pp. 61–74, 1988.