A comparison of three neural networks for building local grid maps

JOSÉ CRUZ[1], URBANO NUNES[2], JOSÉ METRÔLHO[1] and EURICO LOPES[1]

[1] Technology High School
Polytechnic Institute of Castelo Branco
6000 Castelo Branco, Portugal

[2] Institute for Systems and Robotics
University of Coimbra – Polo II
3030 Coimbra, Portugal

Abstract: This paper addresses the local grid map-building problem. Sensor readings are interpreted using a feedforward neural network, and a Bayesian rule is used to update the occupancy probabilities of the grid cells. A comparison of three neural network configurations is made in the map building of indoor environments, using the sonar sensors of the Nomad 200 simulator. The architecture with best results in simulations was tested in a real mobile robot.

Key-words: Mobile Robots, Local Grid Maps, Neural Networks.

1 Introduction

Autonomous robots must be able to learn and maintain models of environments, particularly when robots operate in dynamic environments. To navigate in indoor environments and build a good model of the environment, the mobile robot must know where it is. Certainty grids are a probabilistic finite-element representation of robot spatial knowledge. These grids allow the efficient accumulation of small amounts of information from individual sensor readings into increasingly accurate and confident maps of robot's surroundings [8].

Ultrasonic range sensors are simple in construction, simple in operation and mechanically robust providing a very low cost process for environment perception. These sensors provide distances to surrounding obstacles located within their radiation cone. The time elapsed between the transmission of a wave and the reception of its echo allows the computation of a range reading. However, these sensors have some properties that make map building a non-trivial process. These sensors have a very poor angular resolution, and exhibit specular reflections. Experimental based models derived from data collection have been presented in [7,10]. The ultrasonic pulse may not be reflected back to the sensor, therefore some surfaces may appear to be invisible and the mobile robot can collide. To overcome sonar limitations a probabilistic model in occupancy grids with a Bayesian cell's updating formula was adopted in our work. In order to minimize errors, sensor's readings are interpreted by means of a neural network before the Bayes's update. This paper extends our previous work on grid map building [6] by exploiting the performance of three neural network configurations. The knowledge of the surroundings provided to the mobile robot controller, acts as a safety device, and can ensure a navigation free of collisions. Additionally local maps allow trajectories and manoeuvres planning.

The paper is organized as follows. Next we present previous work in map building. Section 2 describes the map building architecture. Section 3 describes in detail the grid map building algorithm. Experimental results are shown in section 4. Some conclusions are reported in section 5.

1.1 Related Work

Occupancy grid mapping was proposed by Moravec and Elfes [4], and since then has been adopted in numerous robotic systems. Sonar readings are interpreted through probability profiles to determine empty and occupied areas. In the map, beyond empty and occupied cells (areas), unknown areas were explicit represented. Borenstein and Koren [1] applied the occupancy grid map representation in fast obstacle avoidance. A good performance was achieved with a simple and fast method for update cell´s value. In [9] is described a stereovision based mapping and navigation for mobile robots. The algorithm integrates stereovision, occupancy grid

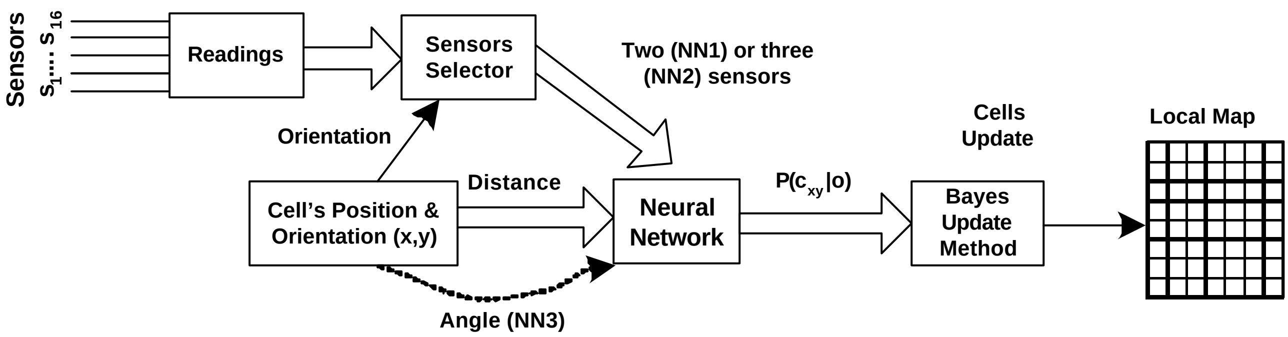

Fig. 1. Map-building architecture.

mapping and potential field path-planning techniques. Another way of building maps was presented in [11], where occupancy grid is associated with fuzzy logic theory. Chow et al. proposed in [3] a probabilistic grid mapping where the probability distribution function is tuned by fuzzy rules. A method to build topological maps using grid-based maps is presented in [12] where the information of environment is acquired by sonars and stereovision.

2 LOCAL MAP-BUILDING PROCESS

Figure 1 shows the map building architecture. The updating process of a given cell (x,y) starts with the Sensors Selector module that chooses two (or three) sensors with orientations closest to the orientation of the cell. The range readings of the selected sensors are provided to the neural network. The function of the neural network (figure 2) is to provide the conditional probability $P(c_{xy}|o)$ given actual sensory observation o. The cell is finally updated by using the Bayes update formula. In next sub-sections this procedure is explained in more detail.

2.1 Neural Networks

In the map building architecture a feedforward neural network is used to determine de probability $P(c_{xy}|o)$ of a cell (x,y) being occupied given actual sensory observation o. In this paper we compare the map building method using three different feedfoward neural networks: NN1, NN2 and NN3. For a given cell (x,y), the input layer of the neural networks consists of:

NN1 (Figure 2a)

1) The observation $o1$=(s_1, s_2) of the two sensors oriented in the direction of cell (x,y);

2) The distance of the center of the cell (x,y) with respect to the mobile robot coordinate system, as illustrated in the example of figure 3 for a circular mobile robot used in the experiments.

NN2 (Figure 2b)

1) The observation $o2$=(s_1, s_2, s_3) of the three sensors oriented in the direction of cell (x,y);

2) The distance of the center of the cell (x,y) with respect to the mobile robot coordinate system.

NN3 (Figure 2c)

1) The observation $o3$=(s_1, s_2) of the two sensors oriented in the direction of cell (x,y);

2) The polar coordinates (distance and angle) of the center of the cell (x,y) with respect to the mobile robot coordinates system.

Therefore, the input layer of the neural network has three nodes when we use two sensors (NN1) and has four nodes if we use three sensors (NN2) or two sensors and the polar coordinates of the center of the cell (NN3). The output layer has only one node which produces $P(c_{xy}|o)$. In each case the network was trained off-line with a back-propagation algorithm [5]. The training examples were generated with the mobile robot simulator. Placing the robot in a known environment, a set of examples was made recording sensor readings at various situations and for adequate ranges according to the cell's size and the size of the grid maps. After training, the network gives values in the range [0,1] that can be interpreted as probabilities of occupancy.

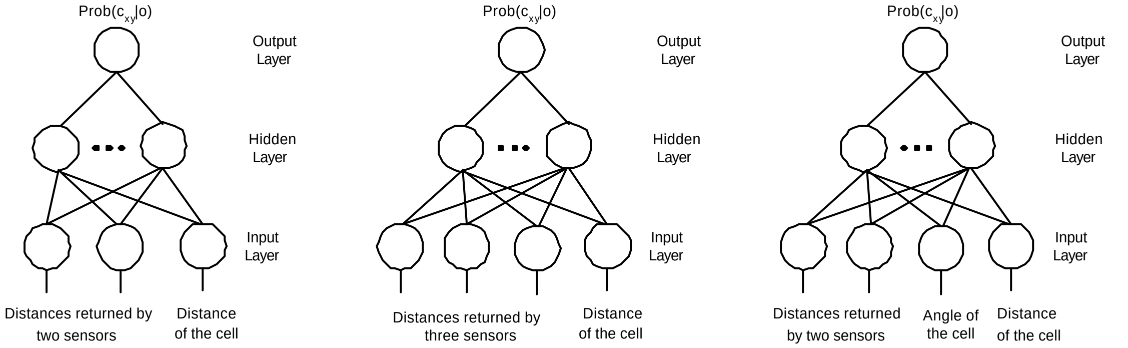

Fig. 2. Feedforward neural networks a) NN1; b) NN2; c) NN3.

Since the neural network is trained based on examples, it can easily be adapted to new situations. Another advantage is its capacity to interpret several sensor readings simultaneously. Interpreting sensor readings in context of their neighbours generally yields more accurate results [12].

2.2 Bayesian-based cells updating

The local mapping consists of estimating the occupancy of a specific area around the robot that moves with it. Let c_{xy} denotes "*cell (x,y) occupied*". So c_{xy} denotes a discrete random variable with events in the universe {0, 1}, i.e $c_{xy} = 1$ stands for cell occupied, and $c_{xy} = 0$ stands for cell free. The mapping can be seen as the problem to estimate the conditional probability:

$$P(c_{xy}|o^{(1)},\ldots\ldots, o^{(N)})$$

where $o^{(1)}$, denotes the first (in time) observation and $o^{(N)}$ the last observation. Based on Bayes theorem, we can express the conditional probability of cell (x,y) to be occupied, given a sequence of observations, as follows [2]:

$$P(c_{xy}|o^{(1)},\ldots\ldots, o^{(N)}) = \frac{P(o^{(N)}|c_{xy})P(c_{xy}|o^{(1)},\ldots,o^{(N-1)})}{P(o^{(N)}|o^{(1)},\ldots,o^{(N-1)})} \quad (1)$$

Applying the concept of "*odds of cell (x,y) to be occupied*" [8], under the assumption of statistic independency of sonar readings, obtained at different instants of time, and with mathematical manipulation we can express the conditional probability as follows:

$$P(c_{xy}\,|\, o^{(1)},\ldots\ldots, o^{(N)}) = 1-(1 + b)^{-1} \quad (2)$$

where

$$b=\frac{P(c_{xy}|o^{(N)})}{1-P(c_{xy}|o^{(N)})}\cdot\frac{1-P(c_{xy})}{P(c_{xy})}\cdot\frac{P(c_{xy}|o^{(1)},\ldots,o^{(N-1)})}{1-P(c_{xy}|o^{(1)},\ldots,o^{(N-1)})} \quad (3)$$

In equations (2) and (3) $P(c_{xy}|o^{(N)})$ is given by the neural network, $P(c_{xy})$ is the initial probability of occupancy of cell (x,y), (equal to 0.5), and $P(c_{xy}\,|\, o^{(1)},\ldots\ldots, o^{(N-1)})$ represents the probability before the actual update. Using equation (2) we can evaluate iteratively the probability of occupancy of each cell, which means that only one value per cell needs to be stored in the local map.
The value of the map cell (x,y) represents the probability of the corresponding space to be occupied (near 1) or free (near 0). Initially all cell values are set to 0.5, i.e. unknown case. Every time a cell seems to be occupied, its value increases, on the contrary, its value decreases. Due to the mathematical characteristics of the update equations (2) and (3), if the cell value is 1 or zero, in the following iterations, the result remains always 1 or zero respectively, independently of the value of $P(c_{xy}|o^{(N)})$). In the experiments described in section 4, it was used the range [0.01, 0.99].

3 LOCAL MAP BUILDING

The occupancy grid approach has the following advantages: simplicity, robustness and adaptability to several environments. In this method, the environment is shaped as a 2D discrete grid. As explained in section 2.2, the value of the cells should be limited to the range]0, 1[.

3.1 Cell Update Algorithm

1. Initialization: $P(c_{xy}) = 0.5$

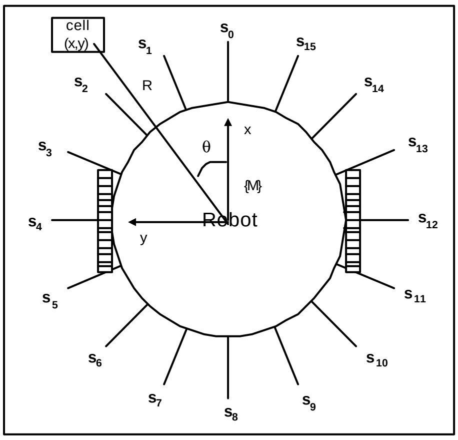

Fig. 3. The mobile robot and sensor's orientation. Each Si represents a sonar sensor. {M} defines the robot's coordinate system. R and θ are the polar coordinates for the cell (x,y) related to {M}.

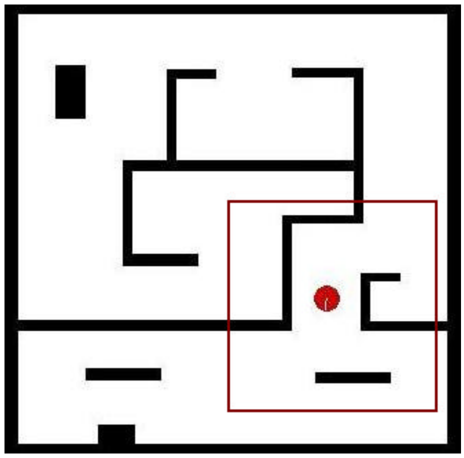

Fig. 4. Simulation environment.

2. For each cell (x,y) and for each new observation $o^{(N)}$ (selected for this cell) the neural network gives as output $P(c_{xy}|o^{(N)})$.

3. Cell's value update:

$$P(c_{xy})[k]=1-\left(1+\frac{P(c_{xy}|o^{(N)})}{1-P(c_{xy}|o^{(N)})}\cdot\frac{P(c_{xy})[k-1]}{1-P(c_{xy})[k-1]}\right)^{-1} \quad (4)$$

where $P(c_{xy})$[k] denotes the actual cell's value, that is $P(c_{xy})$[k]= $P(c_{xy}|o^{(1)},\ldots\ldots, o^{(N)})$. Equation (4) is equation (2) after the following operation. Since in equation (2) $P(c_{xy})$ denotes the cell's initial value that is 0.5, then:

$$\frac{1-P(c_{xy})}{P(c_{xy})}=1 \quad (5)$$

3.2 Selecting Sensor Readings

Whenever a cell (x,y) is proposed to be updated, it is necessary to find the sensors more adequate to provide information about distances, in cell's direction. We use the angle of the polar coordinates of each cell (x,y) to choose the two (or three) sensors whose orientations are the nearest to the cell's orientation.

3.3 Local Maps

When the robot is in motion, a window (grid of cells) is moving with it, therefore, it is necessary to have a relative position estimation, in order to update correctly the cells in the local map. In the local map the position of the robot relative to the window of cells is fixed (at the centre). When the robot is moving we take into account the displacement of the robot in cell's updates. The odometry information plays an important role in this point. This information allows the correct slide of the window of cells (local map) according with the movements of the robot.

4 EXPERIMENTAL RESULTS

The following results were obtained using the Nomad 200 simulator and its 16 sonar proximity sensors. The sensors are disposed around the robot as described in figure 3 (top view of the robot). The simulation environment is depicted in figure 4, which consists of an area of 12x12m^2. Figure 5 shows three local maps obtained by the map-building architecture using the different neural-network configurations: NN1 on the left side map; NN2 on the centre map; NN3 on the right side map. The square frame around the robot (small circle) shown in figure 4 represents the portion of space mapped in the local maps of figure 5. The local grid map has 3600 cells, each one representing an area of 5x5 cm^2. As can be observed the grid maps are very representative of the local space.

In figure 6, a comparison of the errors of the local grid maps of figure 5, for some directions, is shown. We can observe that the NN2 generates errors between 1.5 cm and 34.7 cm, which are in some cases slightly above to the corresponding errors generated by NN1. NN3 generates the smallest errors. The map-building method, for the three NN cases, was also analysed according to the following properties: robustness, adaptability and the impact of robot rotations. The robustness was analysed for a static environment observing the map after initial

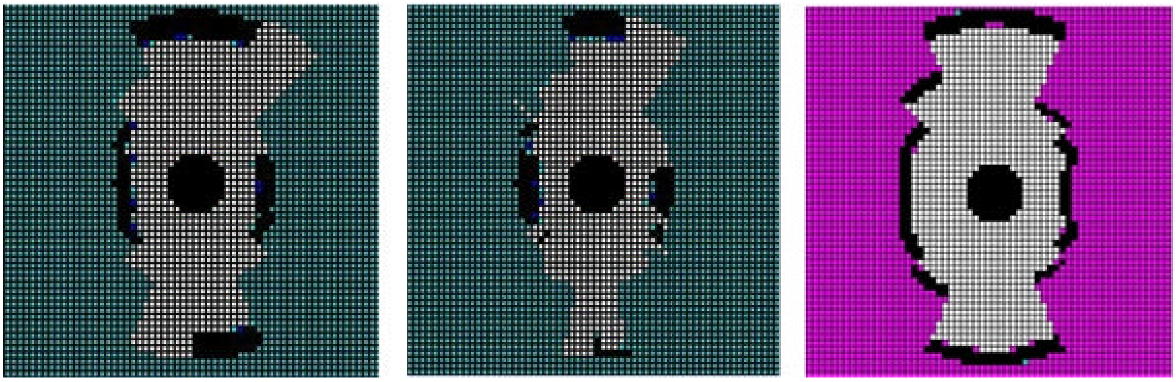

Fig. 5. a) Local map using NN1; b) Local map using NN2; c) Local map using NN3.

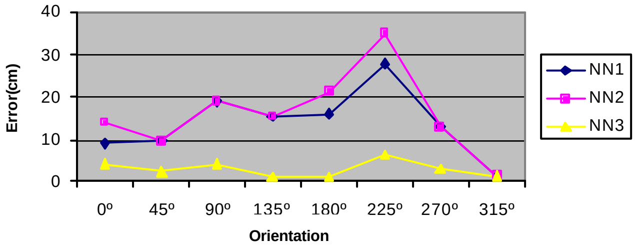

Fig. 6. Error analysis for the three approaches.

Fig. 7. Real environment.

updates (the map is initialised with all cells with value 0.5) and observing the map after multiple updates. The robustness is inferred by comparing the two local maps. The robustness is higher as much as the similarity is higher. All the neural networks proved to have high robustness with results marginally better for the NN3. Adaptability was analysed as follows: first an object is introduced in the environment (in the surroundings of the robot), next the reaction of the method was analysed for the first update and after 15 updates of the map. The procedure was repeated when the object was removed. If the method adjusts the local map quickly to the changes in the environment then the method shows good adaptability. It was observed that the method for the three NN cases reacts quickly to environment changes. This property is very important particularly when the robot operates in dynamic environments. The impact of robot rotations was analysed by rotating the robot a few degrees and observing the changes in the local map. When the robot rotates, sensor's orientation changes and naturally the distance information provided by each sensor may change. Sometimes these changes are enough to make a local map quite different from the correct local map. The method with NN2 proved to be less sensitive to robot rotations and the method with NN3 achieved slight better results than NN1. Since NN2 uses information from three sensors it integrates better the changes derived by the rotations than the other neural networks that make use of only two sensors. However, for example, increasing the number of sensors this problem can be minimized.

Extensive simulations were performed and in general the local map-building method with NN3 achieved the best results.

In order to test this method in real environments we used the Nomad's Scout II mobile robot. Figure 7 shows an example of a real environment used in the experiments with the mobile robot. A local map built using real sonar data, with the top view of the real environment superimposed, is shown in figure 8. Observing this figure we can conclude that the method has good performance real situations with good adaptability and robustness. Figure 9 shows a local map, in a real situation, where one of the small objects near the robot is not detected due to a robot rotation of 66º, in clockwise direction.

5 CONCLUSIONS AND FUTURE WORK

The paper describes a method to build local grid maps. The map building process uses neural networks to interpret the readings of sonar sensors, and a Bayesian rule for cell's updating. A comparative analysis was made for three different architectures of neural networks, in a simulation

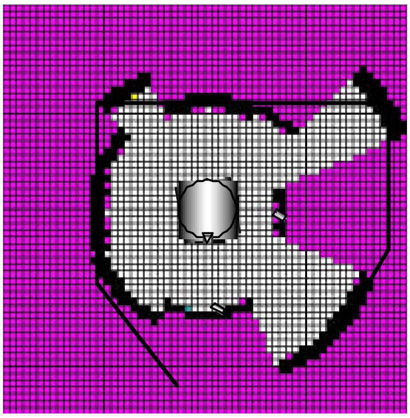

Fig. 8. Local map superimposed with top view of the real environment.

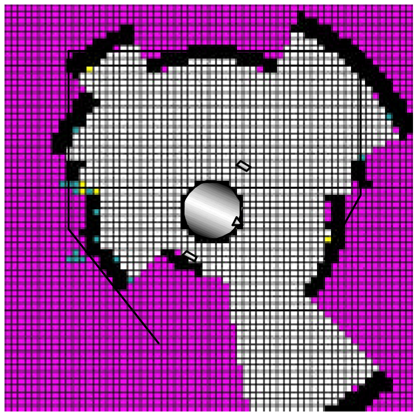

Fig. 9. Local map superimposed with top view of the real environment, after a rotation of 66 degrees. One of the objects is not detected.

environment, in terms of accuracy, robustness, adaptability and impact of robot rotations. The method using NN3 generates local maps with the smallest errors, with good adaptability and robustness. This method was tested in a real mobile robot showing good results as well. However, in some cases, the impact of robot rotations in local maps can be high as mentioned in section 4.

The architecture proposed (figure 1) to build grid maps uses a set of sensors to update cell's value. The sensor with smaller distance predominates in the presence of sensors with greater distances. Therefore, it may arise high errors when a set of sensors used to update the cells gives measures with significantly disparate values.

The map-building algorithm will be further investigated in the following directions: 1) integration of other sources of sensory data; 2) application of the algorithm in the navigation of a real mobile robot; 3) building an indoor global map by integrating local maps information.

6 ACKNOLEDGEMENTS

This project was partially supported by FCT (Portuguese Science and Technology Foundation) project POSI/33594/SRI/2000.

References:

[1] Borenstein, J. and Yorem K., The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots. In: *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, June 1991.

[2] Burgard, W., A. B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, Experiences with an Interactive Museum Tour-Guide Robot. In:*T.R.CMU-CS-98-139*, June 1998.

[3] Chow, J., A. Rad, Y.L. Ip, Enhancement of Probabilistic Grid-based Map for Mobile Robot Applications, In: *Journal of Intelligent and Robotic Systems*, 34(2): 155-174, June 2002.

[4] Elfes, Alberto, Sonar-Based Real World Mapping and Navigation. In: *IEEE J. Robotics and Automation*, Vol. RA-3, No. 3, June 1987.

[5] Fausett L., Fundamentals of neural networks-architectures, algorithms and applications, Prentice Hall Int. Editions 1994.

[6] Cruz, J., and U. Nunes, Generating Local Maps for Mobile Robots. In: *4th Portuguese Automatic Control Conference*, October 2000, pp. 412-417.

[7] Moita F., A. Feijão, U. Nunes, and A.T. de Almeida, Modelling and Calibration of an Ultrasonic Ranging System of a Mobile Robot. In: *First Portuguese Automatic Control Conference*,Vol.1, IST September 1994, pp61-65.

[8] Moravec, H. P., and Mike Blackwell, Learning Sensor Models for Evidence Grids. In: *CMU Annual Research Review*, Sept.1993, pp. 8-15.

[9] Murray, Don, and Cullen Jennings, Stereo Vision based Mapping and Navigation for Mobile Robots. In: *Proceedings of the IEEE ICRA 1997*.

[10] Harris, K., and M. Recce, Experimental modeling of time-of-flight sonar. In: *Robotics and Autonomous Systems*, 24 (1998), 33-42.

[11] Oriolo, G., G. Ulivi and M. Venditteli, "Fuzzy Maps: A New Tool for Mobile Robot Perception and Planning", *IEEE Trans. on Systems, Man and Cybernetics, part B: Cybernetics*, vol. 28, No. 3, 1998, pp. 316-333.

[12] Thrun, S., Learning metric-topological Maps for indoor Mobile Robots Navigation. In: *Artificial Intelligence,* 99(1) 1998, pp 21-71.