Neural sensor fusion for spatial visualization on a mobile robot[1]

Siegfried Martens[a,b], Gail A. Carpenter[a], Paolo Gaudiano[a,b]

Boston University
[a]Department of Cognitive and Neural Systems
and [b]The Neurobotics Lab

ABSTRACT

An ARTMAP neural network is used to integrate visual information and ultrasonic sensory information on a B14 mobile robot. Training samples for the neural network are acquired without human intervention. Sensory snapshots are retrospectively associated with the distance to the wall, provided by on-board odometry as the robot travels in a straight line. The goal is to produce a more accurate measure of distance than is provided by the raw sensors. The neural network effectively combines sensory sources both within and between modalities. The improved distance percept is used to produce occupancy grid visualizations of the robot's environment. The maps produced point to specific problems of raw sensory information processing and demonstrate the benefits of using a neural network system for sensor fusion.

Keywords: Mobile robot, neural network, sensor fusion, occupancy grid, sonar, range

1. INTRODUCTION: NEURAL SENSOR FUSION

Mobile robots require accurate representations of their surroundings for navigation. Acquiring these representations involves stages of increasing abstraction, transforming analog streams of sensor values into a symbolic view of the world. Sensor fusion is a topic of great current scientific interest[1,2,3,4]. Individual sensors tend to have shortcomings limiting their applicability; sensory data can be fused, however, and the fused percept can be more veridical than that provided by any single sensor. A robot's perception of its local environment must be as accurate as possible. To increase this perceptual accuracy, we use a neural network to integrate up to sixteen sensory inputs, from both sonar and visual sources. The estimate of range is based on several sources, and is less vulnerable to noise or other errors from the individual sensors. This work subsumes an earlier project on integrating sensors with a neural network[5], applying the fused sensor to mapping the robot's local environment.

Two important aspects of robot motion are reactive control and planned navigation. Reactive control is concerned with generating the robot's next motor command, based on immediate sensory information. Navigation takes a more global approach. When the robot needs to attain a spatial goal, navigation is required to avoid any obstacles, or to plan complex routes. By finding which of fifteen combinations of sensor inputs yields the best predictive performance with an ARTMAP[6,7] neural network, this project first develops a fused sensor adequate for reactive control. The performance measures of these neurally integrated combinations are compared with one another and with a simpler fusion scheme. This more reliable sensor makes reactive control safer, minimizing the chances of collision. Navigation, however, requires a map of the environment for path planning. In the second part of this paper, the best performing combinations of inputs are used to produce maps of the environment. These maps are produced by mapping sensor predictions onto a grid-based representation. Two forms of sensor fusion are thus developed: the first integrates raw sensor values, yielding a virtual range sensor, and the second integrates the virtual range sensors into a map.

Occupancy grids provide a spatial framework for the fusion of sensory data. To be fused, sensory streams must be converted to a common reference frame. Elfes[8,9] uses a discretized cartesian grid as this framework. Predictions from the range sensors are described by a model which specifies the expectancies of free space and obstacle locations based on the reported range. These predictions are iteratively applied to a global map, according to the position and orientation of the sensor. A map of the environment is thus constructed, describing the locations of obstacles and of free space. The map encodes uncertainty information as intermediate occupancy values, which may be used to adjust robot speed, or to weight a heuristic search algorithm.

---

1. 677 Beacon St., Boston, MA 02215, USA, {sig, gail, gaudiano}@cns.bu.edu, http://neurobotics.bu.edu

image represent the lowest edge in each column, and the relative heights of these points are the distance measures returned by visual sonar. The sensory input from visual sonar is thus a vector of eight real-valued numbers (***V1-V8***), with low numbers indicating a nearby obstacle and high numbers indicating free space. The visual sonar sensor detects the bases of walls well, but is confused by image discontinuities that do not correspond to obstacles, as resulting for instance from textured carpeting, door sills, or markings on the floor (e.g., Figure 2(b)).

Sonars are the other main sensors used. They emit an ultrasonic beam and use the amount of time until the echo to calculate distance to obstacles. Sonar's input to the neural network consists of a vector of up to eight numbers (***S1-S8***), describing the distance to obstacles as measured by the B14's eight frontal sonars. Sonar values exceeding a maximum of six meters, as occurs when the echo does not return to the sensor, are clipped to the maximum value. The raw sonar data suffer from a variety of limitations: the returning echo may have been emitted by a different sensor, or may have bounced off several surfaces before being detected. Ultrasonic and visual sonar both have sensory limitations, but in different circumstances, and so they seem apt choices for sensor fusion.

2. DATA COLLECTION

A data set of 10,167 data samples was collected over the course of a few hours, each sample consisting of a set of sensor readings and an associated distance reading obtained through odometry as the robot roamed around an enclosed area of our lab. This data set, which is larger than required for training, was collected in order to get enough data for a thorough and meaningful analysis. Training the ARTMAP network with all of these samples takes on the order of a minute or two.

2.1 Collection of samples

The values recorded are ***S1-S8*** for sonar; ***V1-V8*** for visual sonar; the retrospectively computed distance D from the robot to the wall; and the angle of approach to the obstacle, specified by the infrared sensor I_x which detected it. Samples are recorded every 20 cm as the robot travels in a straight line. When an obstacle is encountered, as measured by the infrared proximity detectors, a new direction is chosen randomly and the process is repeated. The data collection is conducted in an empty area, approximately 2 m by 3.5 m, bounded by flat surfaces (walls and styrofoam panels). This training area is L-shaped, so all corners were concave, except for a single convex one.

2.2 Distance metric

As walls are detected using the B14's infrared sensors, the recorded distance D is always slightly less than the true distance, with the error an increasing function of the slant at which the wall is approached. This type of error is not corrected for in this project (but see section 5.1). The recorded distance D is specified as a continuous value. As it provides the teaching signal to a classifier, the distances are binned into 20 discrete categories. Binning is nonlinear, with small distance bins up close and bin size proportional to the square root of the distance. This nonlinearity allows predictions at shorter distances to be more accurate. Figure 3(a) shows these bins, with the column width indicating the range of distances covered by the bin and the height of the column representing the number of samples in the bin. The semicircular sector plot in Figure 3(b) gives a spatial view of the density of the data collected. The data from each distance bin are further subdivided by the impact angle specified by the infrared sensor I_x that detected the obstacle. Notice that in all cases the obstacle detected lies straight ahead; the impact angle reflects the obstacle slant relative to the robot's heading. The radial thickness of each cell in the semicircle also reflects the nonlinear binning of distances. It is also important to note that although in this and subsequent figures we break down the data

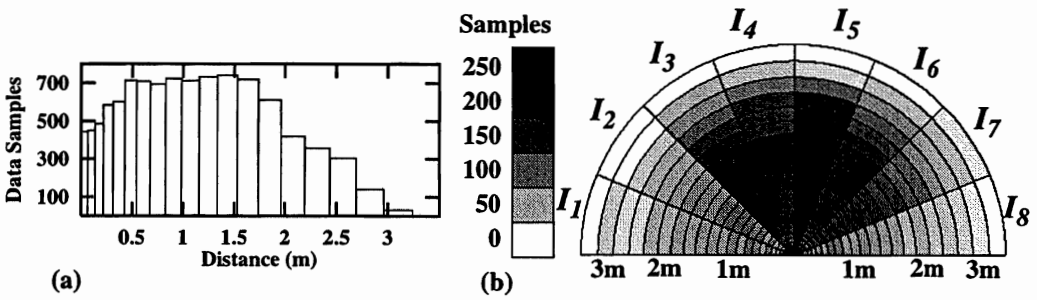


Figure 3. Density of data sampling. (a) Data samples for each distance bin. (b) Data samples for each distance bin broken down by angle of impact. The non-linear increase in size of distance bins is shown in both plots.

Occupancy grids can be the basis for planning, using any of a number of search algorithms, such as the A* algorithm. See Winston[10] for a discussion of this and other search algorithms.

1.1 The B14 robot

This research employs a B14 mobile robot from Real World Interface, Inc. (Jaffrey, NH), a cylindrical robot measuring 14" in diameter (Figure 1), equipped with a synchro drive that permits forward and reverse translation and rotation in place. Arranged around the B14's surface are sixteen infrared proximity detectors and sixteen sonar range finders, distributed uniformly around the robot's perimeter, and a camera mounted on a pan-tilt platform.

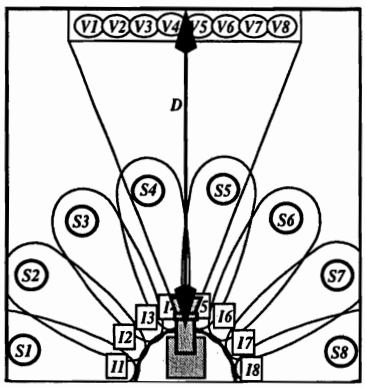


Figure 1. Overhead view of the B14 robot, showing its ranging sensors. Closest to the robot are infrared sensors, labeled ***I1-I8***. These detect obstacles at close range, and specify the angle at which the obstacle is encountered. Sonars are labeled ***S1-S8***, (beams emanating radially from the robot). A camera is mounted on top of the robot, and provides gray-scale images. An edge detection algorithm is applied to these images, yielding *visual sonar*, depicted as sensors ***V1-V8***.

1.2 Self-supervised data collection

The robot learns to predict the frontal distance to obstacles using a new version of the ARTMAP neural network[11,12]. Developed for applications in remote sensing and medicine, this version implements a new category choice rule which helps maximize code compression. The training process is self-supervised, i.e., the robot is not provided with the distance to obstacles. A relative distance measure is obtained using odometry, as the robot randomly explores its training area. Snapshots of the sensory input are recorded as the robot moves in a straight line. When the robot encounters an obstacle detected with the infrared sensors, on-board odometry provides a relative distance to associate with each sensory snapshot. The neural network is trained to learn associations between this sensory input and the distance to the obstacle. The robot thus learns to interpret its sensory input on its own, without human intervention. Self-supervised learning can allow the robot to explore new environments autonomously. This ability is complemented by the ARTMAP network's capacity for one-shot learning, which allows it to form associations between inputs and percepts with a single presentation. Unlike many neural networks, ARTMAP learning is match-based, rather than error-driven. Learning can be fast, as it does not require gradient descent of an error surface, which can require many epochs of training.

1.3 Visual and sonar inputs to the neural network

Input vectors to the neural network are collected from the camera and from the sonar sensors. The camera provides a crude estimate of relative distance to objects through visual sonar[13], obtained by dividing the image into eight columns and searching for edges from the bottom of each column. Under the assumptions that the robot is operating on a flat surface and that obstacles are on the ground, the distance of an edge from the bottom of a column is proportional to the distance of the corresponding obstacle from the robot. Figure 2 provides an example of visual sonar working in an office setting. The points plotted on the

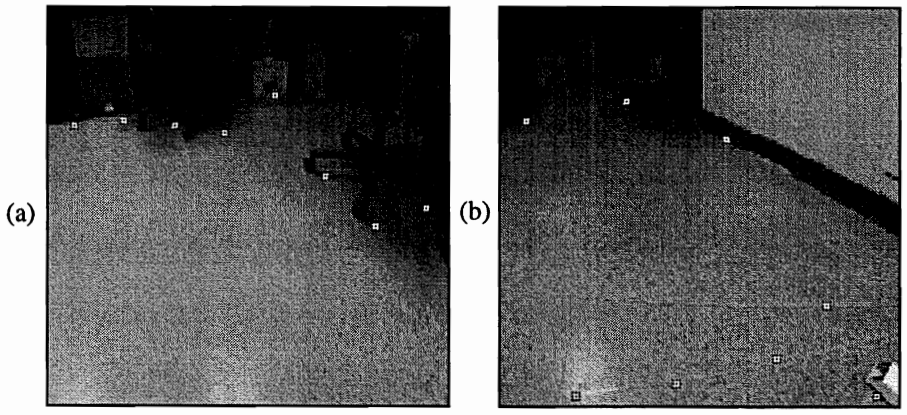


Figure 2. Visual sonar's view of a room. The heights of the points reflect their distance from the robot, and are returned as the values of sensors ***V1-V8***. Perception is accurate in (a), but in (b) markings on the floor are mistaken for an obstacle.

by the impact angle for clarity, the neural network is trained by lumping all inputs from a given distance regardless of slant. Thus, the neural network learns to predict the distance to obstacles that lie straight ahead, regardless of the obstacle's slant.

3. CROSS-VALIDATED TESTING

3.1 Evaluating ARTMAP predictions

The fuzzy ARTMAP neural network is applied to the data set, learning to associate different combinations of sensory inputs with distance. For each data sample i, the network produces a predicted distance p_i, which is compared to the recorded distance D_i. Producing one entire set of n predictions is called a simulation, though this term does not imply that the data themselves are simulated. Simulations are compared using ζ, the average absolute predictive error over the test set:

$$\zeta = \frac{\sum_{i=1}^{n} |p_i - D_i|}{n} \tag{1}$$

Fifteen simulations are compared, each with a different combination of inputs provided to the neural network for fusion. Combinations range from sonar only or visual only to mixtures of both sensory modalities. Specifically, the sensor combinations use two, four or eight of the available values (sonar or visual), in each case using the most central sensors, starting with ***S4***, ***S5*** or ***V4***, ***V5*** (Figure 1).

3.2 Cross-validated testing

Five-fold cross validation ensures that the training and test sets are always disjoint. More specifically, the data set is divided into five partitions. One at a time, each of these data partitions is reserved as the test set while the network is trained with the remaining four. This process is repeated five times, each time using a separate data partition for testing, and the remainder of the data set for training. Moreover, to enhance the accuracy of prediction, and the repeatability of the results, five copies of the neural network are trained, each with a separate ordering of the training set. The final predicted distance p_i is the average of the predictions of these five networks. Thus, for each reported simulation, twenty-five copies of the neural network are used, one for each of the five orderings of the five training sets.

3.3 Comparing input variations

Table 1 compares the average absolute error for the fifteen simulations performed. The best result, highlighted in boldface, uses the two most central sonar sensors and the four most central visual sensors (*2S+4V*). However, all of the results using any of the sonars and at least four visual sensors are nearly as good. Table 1 demonstrates the advantage provided by sensor fusion, both within and between sensory modalities. When using only one type of sensor, performance is proportional to the number of sensors used. The predictive error is always less, however, when both sonar and visual data contribute to the prediction.

Table 1: Average absolute error (cm), for all fused combinations of sonar and visual sensors.

		Sonar			
		0	**2**	**4**	**8**
Visual	**0**		31.1	15.5	13.1
	2	34.7	12.6	11.2	11.1
	4	25.0	**10.3**	10.5	10.7
	8	16.7	10.6	10.7	10.7

3.4 Predictive value of raw sonar

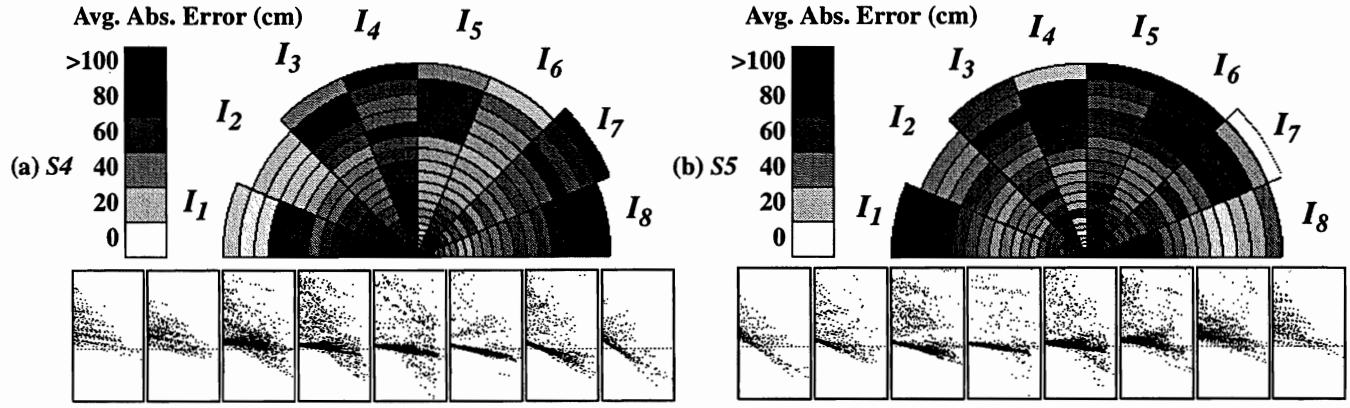


Figure 4. Predictive error of the raw sonars. (a) ***S4***, and (b) ***S5***. Semicircular plots show average absolute error level in each radial/distance bin (bins not shown were not sampled in data collection). Small dot plots show error for each datum collected, one plot for each of the eight impact angles. The Y axis shows predicted minus recorded distance, and is in the range [-200, 300] (cm). The X axis range is [0, 325] (cm).

Before looking at our results with the ARTMAP network in more detail, we illustrate the accuracy obtained with our data set using the raw sonar data. Figure 4 shows, for the two frontal sonars ***(S4, S5)***, the average absolute error as given by Equation (1), broken down by impact angle ***(I1-I8)***. Below each semicircle, the raw error data are plotted as a function of distance, again broken down by angle of impact, with the leftmost box corresponding to obstacles detected by ***I1*** and the rightmost box corresponding to ***I8***. Each point in these scatter plots is the calculated difference between measured and actual distance.

If each sonar were "perfect", all the points would be zero, i.e., the data would lie on the horizontal midline. These error plots show some interesting trends: the majority of the points fall above the horizontal midline, suggesting that sonar tends to overestimate, probably due to echoes bouncing against more than one surface or cross-talk between sensors. However, the data show a tendency to underestimate at large distances. It is unclear whether this is due to the sonar itself (e.g., through reflected echoes), or to the fact that we are using odometry to measure "actual" distance, and odometry is known to be flawed over large distances. It should also be noted that Figure 4 shows that the accuracy of sonar at high impact angles is only marginally worse than accuracy for head-on obstacles. One might expect performance to decline in proportion to the angle of approach because of the directional nature of sonar. This phenomenon might be due to the substantial amount of styrofoam bounding the training area (about 60%). Styrofoam is known to enhance the performance of sonar at high slant angles.

3.5 Fusion of two central sonars: *min(S4,S5)*

Figure 4 makes clear that relying on raw sonar data can yield very noisy distance estimates. A simple way of getting rid of some of the noise is to take the minimum value returned by (***S4***, ***S5***). This assumes most of the errors are overestimate, e.g., from missing the obstacle altogether. The validity of this idea is shown in Figure 5, which shows the predictive accuracy of a sensor based on ***min (S4, S5)***. Accurate prediction is extended to most of the central angular bins, representing relatively orthogonal impact angles (Figure 5(a)). It is interesting that the ***min(S4, S5)*** seems to do poorly for obstacles straight ahead at large distances, as is evident in the outermost cells for sectors ***I4*** and ***I5*** in Figure 5(a). The individual scatter plots in Figure 4 suggest the reason for this problem: each sonar, especially at the central angles, tends to underestimate at larger distances, so that ***min(S4, S5)*** worsens the results. In Figure 5(b), the absolute error of the prediction is shown for each sample in the data set (dots). The averages and the standard deviations of the absolute error values within each of the distance bins are plotted. The lowest level of error is seen in the middle range, between one and two meters. This is problematic for sonar-based navigation, as the high level of error close to obstacles increases the likelihood of collisions. Figure 5(a) shows that much of the error at short distances occurs during approaches to slanted obstacles. In this case, sonar sensors often receive an improper echo. This can result from the interception of another sensor's ultrasonic beam, or from receiving the correct beam after more than one reflection off an obstacle. The result is an overestimate of the distance to the obstacle. Error rises again at higher distances, between two and three meters, but this is less of a problem than the error at short distances. Having seen this illustration of the problems of prediction based on raw sonar, we can now look in more detail at the result of fusing sensory information with fuzzy ARTMAP.

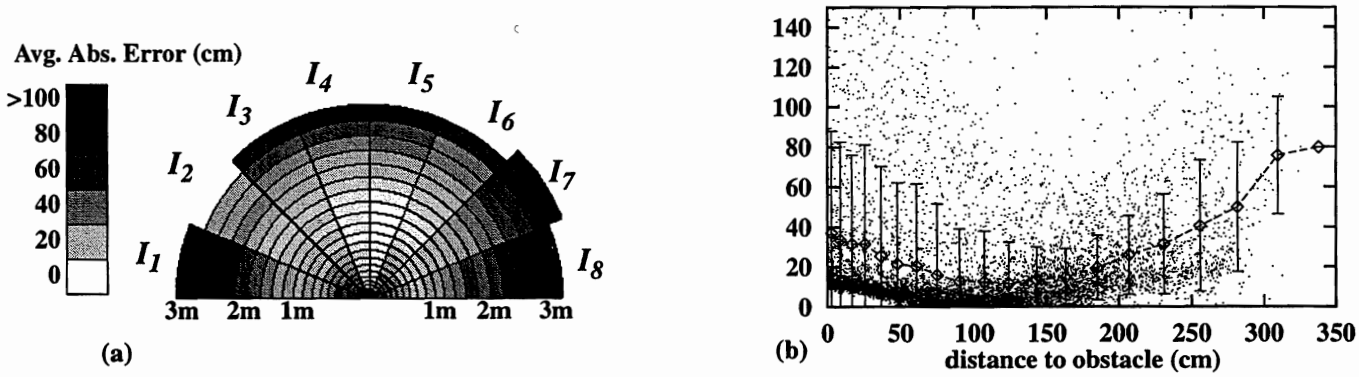


Figure 5. Predictive error of the ***min(S4, S5)*** model. (a) Average absolute error in each sampling bin. Model predictions of distance are accurate at short distances and head-on approaches, but performance is impaired at longer distances or more oblique impact angles (b) Absolute error by distance. Dots show absolute predictive error for each sample in the data set, dotted line shows average absolute error for each distance bin, and error bars indicate one standard deviation.

3.6 Neural fusion of two ultrasonic and four visual sensors

Figure 6 illustrates the predictive performance of the neural network fusing ***2S+4V***, i.e., the two most central sonars and the four most central visual sonars. Table 1 showed this to be the combination of sensors yielding the overall lowest average absolute error. Figure 6(a) demonstrates predictive improvements for nearly all angle and distance bins. Performance at high impact angles and distances is dramatically improved. Even more important, prediction at close range is now quite accurate at all impact angles. This is more clearly visible in Figure 6(b), which shows that average absolute predictive error approximates a nondecreasing function of distance. Predictive accuracy thus now has the desirable property of being roughly proportional to distance. Moreover, the standard deviation of the error is significantly smaller than in Figure 5, and it is an approximate function of distance. As the error for the ARTMAP method is less variable, and this variability is smallest near the walls, the fused sensors are a safer basis for navigation than the raw sonar data.

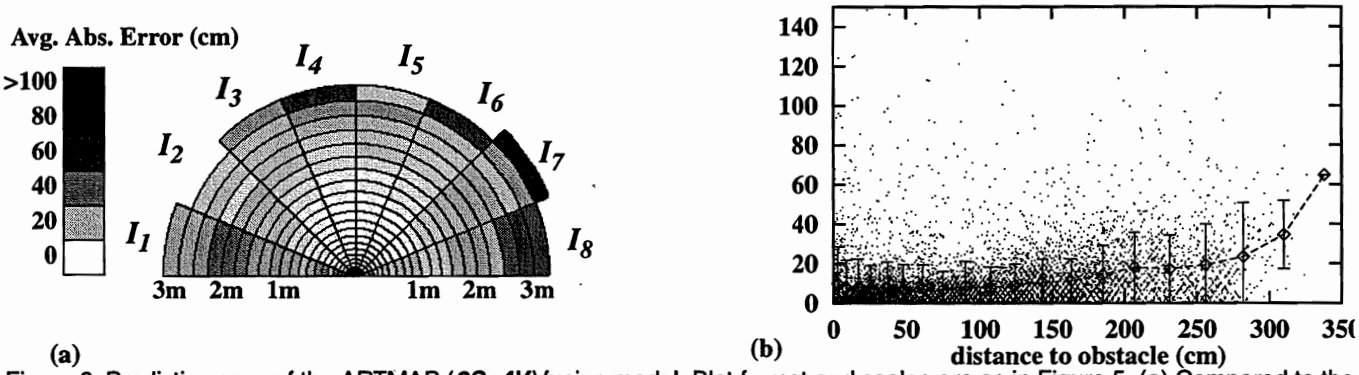


Figure 6. Predictive error of the ARTMAP (***2S+4V***) fusion model. Plot format and scales are as in Figure 5. (a) Compared to the ***min(S4,S5)*** model, predictive accuracy is equal or better at all distance and angle bins. Prediction of distance is improved at high angles and at large distances. (b) Average absolute error is less at all distances, and variance of error is smaller.

3.7 Discussion of integration results

The results in Figure 6 and Table 1 demonstrate the viability of ARTMAP as a methodology for integrating data across and within sensory modalities. Using two sonar and four visual sensors, a fused system has been devised which outperforms the raw predictions of distance of the two frontal sonars. The obtained system is robust in several key aspects. It is more accurate at high impact angles to an obstacle, compensating for sonar's difficulty in detecting obstacles at oblique angles. The errors made by the fused system are small at short ranges, and the variance of the error follows the same trend. This allows the robot to navigate with confidence at close quarters. We now use this robust sensor to produce maps of the robot's environment.

4. OCCUPANCY GRID VISUALIZATION

4.1 The Gaussian sensor model

Sensor models used to represent the reading from an ultrasonic sensor are generally composed of Gaussian waveforms. An idealized sensor as discussed by Elfes[8] is shown in Figure 7. The abscissa represents distance from the robot (range), and the coordinate represents the probability of occupancy. A sensor reading of *r* produces an occupancy value of 1 at the distance *r*, of 0 between the robot and *r* (the "free space hypothesis"), and values of 0.5 (uncertainty) beyond the obstacle. Such a model is appropriate if the sensor is perfect. For real sensors, this model will yield bad results, as noise will produce obstacles in the wrong place. This justifies the use of a Gaussian model to represent the positional uncertainty of the reading.

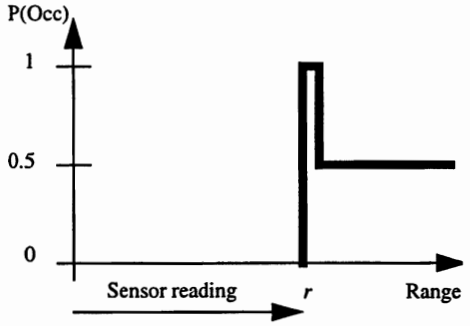


Figure 7. Occupancy probability profile for an ideal sensor, given a range measurement *r*.

Three Gaussians are used: two with positive amplitude describe the obstacle in two dimensions; a negative one represents the free space between robot and obstacle. We use a slightly different convention than Elfes for occupancy values: unknown values are 0, positive values represent obstacles, and negative values represent free space. A magnified version of the model is shown in Figure 8. The sensor model shown is 7 pixels by 50 pixels. One gaussian is wide and negative and centered on the robot (the left side of the figure). This gaussian is shown in light shades and represents the empty space near the robot. The other gaussians are narrow and positive, and represent the obstacle. The value zero is depicted as a middle grey shade, as shown at right.

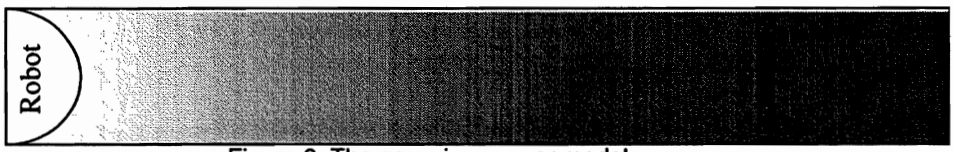


Figure 8. The gaussian sensor model

The model in Figure 8 is used to fill in values in an occupancy grid map. At start-up, each map cell is initialized to zero. As predictions accumulate, the map is filled in with white space for empty areas and in black for the obstacles. For each sensor prediction, the sensor model is applied to the occupancy grid at the appropriate position, based on the position and orientation of the robot. Newer predictions keep accumulating, dynamically updating the map as information becomes available. Black specifies a value of +1 in the occupancy grid, and is associated with a high probability of being occupied, and white represents an occupancy value of -1. The value zero represents a lack of information. Occupancy values saturate at +/-1, though later information can move the occupancy values away from these extremes. This introduces a bias in the map, which may or may not be desirable: recent information can easily override established obstacles, as occupancy values are clipped once the extreme values have been reached.

4.2 Recording of visualization data

A short recording of sensor values provides the data for visualization. It consists of 37 samples, each specifying sixteen sonar values (mm to obstacle). In contrast to the beginning of this chapter, we now fuse the sonar sensors alone. The sonar provides a view all around the robot, allowing a rapid filling in of the map. If the visual sonar were used, it could contribute only to a single sensor aimed in the forward direction, and so it is not used. With sonar, sixteen ARTMAP-derived virtual sensors can be evenly distributed about the robot. For these visualizations, the robot moves forward 2.25 meters in the training area. A straight line motion is specified, but the unevenness of the floor causes the actual trajectory to veer slightly to the right. This curve of the map due to the robot's drift complicates comparison between maps. Since correcting for drift is not the focus of this research, a simulation of the robot's drift is used to correct for the curvature of the path. A turn is added to the robot's path, proportional to distance traveled. This method is used in the rest of the paper to correct the curvature error due to odometry.

4.3 Occupancy grid view of raw sonar recording

Figure 9 shows the occupancy grid after the application of 1, 11 and 37 sensor samples. The occupancy grid is shown evolving in time, as the robot's motion provides information from different viewpoints. Each sample consists of 16 raw sonar values. The distortion produced by sonar when near a wall can be seen on the left of the room, where the robot started, and is seen

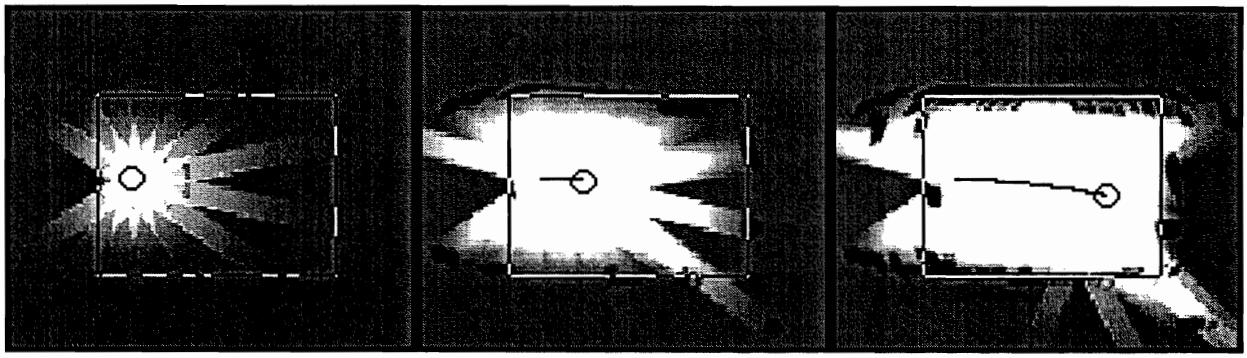


Figure 9. Occupancy grid map based on sonar data. *(l-r)* Views after 1, 11 and 37 samples (225 cm total travel). The robot's final position is shown as a circle, its trajectory is the line behind it, and the rectangular outline shows the room boundaries

starting to form on the right, as the robot approaches. These horn-shaped misperceptions wrongly indicate free space, which could be dangerous to a robot navigating in an unknown environment. Perception of the top and bottom walls is accurate either because the robot never approaches them too closely, or because they are made of styrofoam. The walls on the left and right sides are smoother, and more difficult for sonar to perceive. The line extending to the left of the robot shows the trajectory followed. The actual shape of the room is superimposed on the occupancy grid map.

5. ARTMAP PREDICTIONS

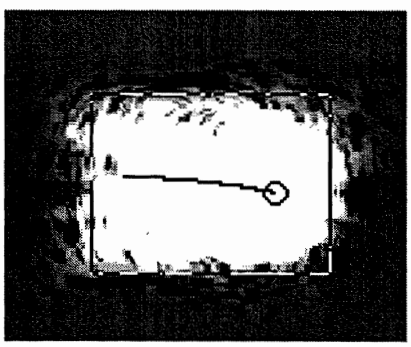
5.1 Preprocessing of the data set

The data used for training the neural network have incorporated a systematic source of error, namely, the robot detects the wall when the infrared sensor nearest to the wall is activated. The robot is usually at a perpendicular distance of approximately 10 cm from the wall when it is detected. Depending on the robot's angle of approach, the actual distance remaining before the robot would have physically encountered the wall can be from 10 cm when the wall is detected by ***I4*** or ***I5***, all the way up to 120 cm when the wall is detected by ***I1*** or ***I8***. Since which infrared was activated was recorded for the entire data set, all the data were reprocessed to partially offset this systematic source of error. For each of the 10,167 samples, the recorded distance has a number added to it, proportional to the tangent of the angle of approach. Most samples are at the middle angles of approach, and so increase only slightly with correction. Before this reprocessing, the maps produced were smaller than the room. Reprocessing the data set moves the perception of the walls to the appropriate place, i.e., the room perceived is the right size in both height and width.

5.2 ARTMAP occupancy maps for two input combinations

Figure 10. ARTMAP predictions applied to the occupancy grid. In (a) the two central sonars are used (**2s**), in (b) the network is trained with eight sonar inputs (**8s**).

(a) 2s



(b) 8s

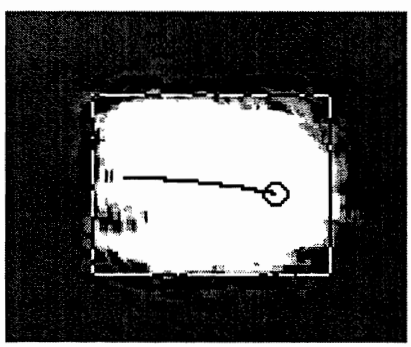


Figure 10 shows occupancy maps produced using ARTMAP. In (a) prediction is based on fusing only the two most central sonars (***S4***, ***S5***) and is fairly unreliable, whereas in (b), which shows the fusion of all eight sonars, it is reasonable. The **8s** case draws the walls more precisely than does **2s**. Table 2 shows the average absolute error in both cases. The preprocessing of the data make these errors slightly larger than those seen in Table 1.

Table 2: Average absolute error obtained with two and eight inputs to the neural network.

Inputs	2s	8s
Error (cm)	31.6	15.2

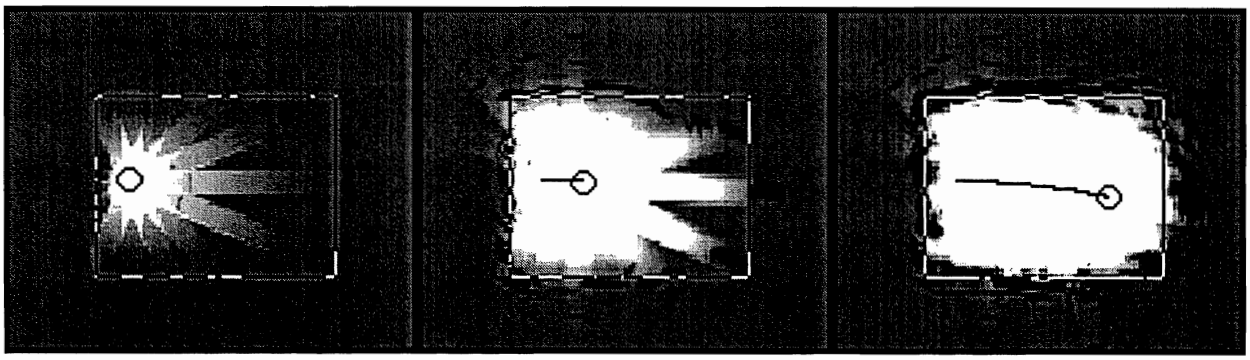


Figure 11. Sensor prediction based on median of five votes. Input is **8s**. Occupancy grid is shown at beginning, middle and end of robot's motion. The final map produced is the most faithful to the actual room boundaries.

Five ARTMAP voters are used to produce the maps in Figure 10. Since each voting ARTMAP network is trained with a re-ordering of the same training set, no additional data are required. Each of the five votes is combined here by averaging. Using the average of five ARTMAP votes as the prediction of distance sharpens the resulting map, but the average can be vulnerable to the influence of outliers. Using the median of the votes rather than the average can reduce this effect (Figure 11).

The map seen in Figure 11 matches the boundaries of the room on all four sides, and the corners are accurately rendered. This map provides a good basis for spatial perception and more high-level processes to operate upon. More importantly, the holes seen in the corners with raw sonar (Figure 9) could pose serious problems for a path planning algorithm, which might direct the robot to travel through the illusory holes.

5.3 Code compression with instance-count pruning

While the map produced using the median of five voters is accurate, the need for 8375 categories (across five voting networks), with the corresponding need for memory and processing time, is excessive. Any method of reducing that number without significantly impairing map accuracy would be useful. A variation of ARTMAP called instance-counting[15] can serve this pur-

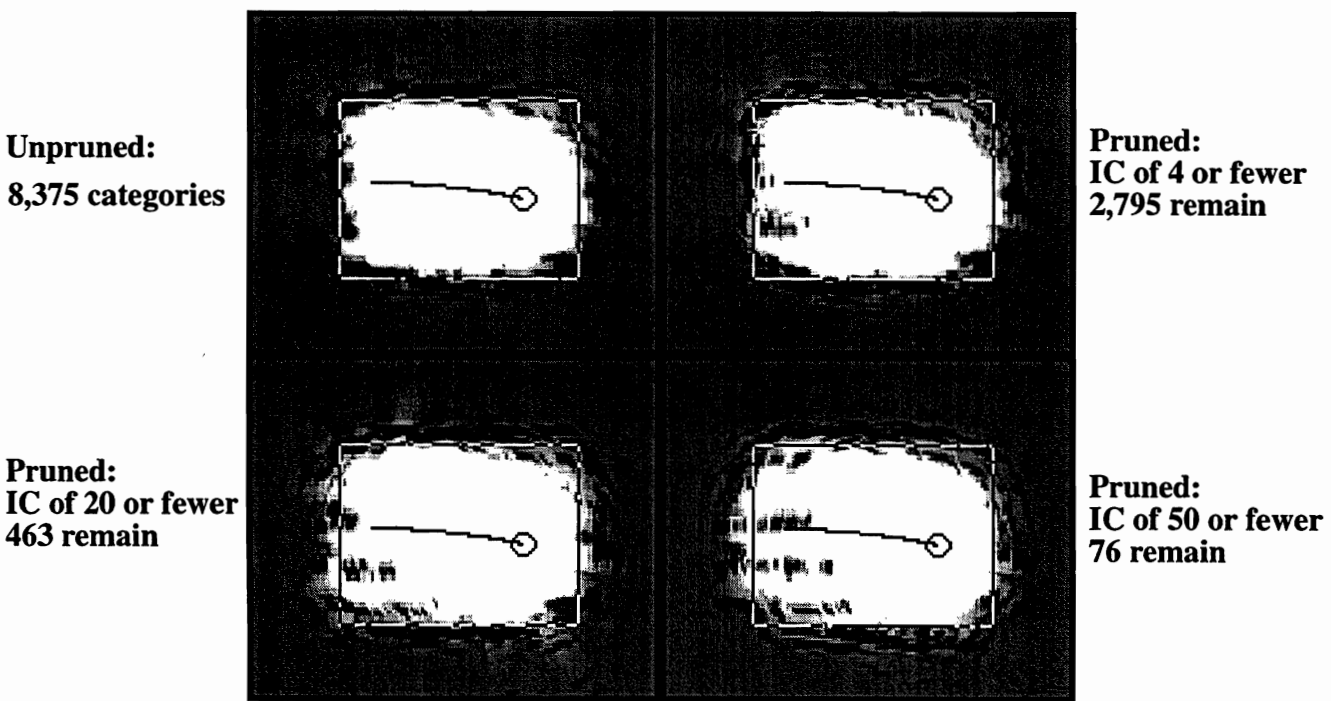


Figure 12. Pruning categories: Maps shown are produced after removing a large fraction of the category templates. The instance count (IC) threshold and the number of categories remaining are shown.

pose. The instance-count (IC) tracks the number of samples that have been incorporated into a category template. The lower a category's instance count, the fewer instances the category is based on, and the less the impact of removing it from the network's memory. By throwing out categories with instance counts below a threshold, the networks can be pruned of the least experienced categories. In the map shown in Figure 11, about a third of the categories have instance counts of 1, while half have count two or less. As shown in Figure 12, removing one half or even two-thirds of the categories formed during training has little effect on the final map, hinting that many categories are due to noise in the training set.

5.4 Discussion of visualization results

These visualizations convey the advantage of the fused virtual sensor over the raw sensory information. As shown by Figure 9, the raw sonar tends to make mistakes when the robot is near a wall. The fused virtual sensor overcomes this limitation by considering information from several sources. ARTMAP learns to recognize the patterns that result from seeing walls at an angle and associates them with the distance metrics learned in training. This fusion of sensory inputs allows the network to recognize when the raw signal is in error.

When the sonar does perceive the wall, it does so fairly accurately, as seen in the top and bottom walls in Figure 9. In contrast, ARTMAP's prediction is more diffuse. This is mainly due to the limited resolution of ARTMAP's perceptual output, since it is trained to predict one of 19 range categories. Its predictions are thus discrete, in contrast to the raw sonar's continuous valued output.

It should be noted that the resolution of the maps shown here is greater than would be needed in an actual navigational or reactive robotics application. This level of resolution is used to make more obvious the differences between the models, and is not necessarily optimal for a real-time system. In particular, while the maps shown are produced in slightly less than real-time, maps with a bit less resolution could easily be produced in real-time, while preserving enough detail for navigation and obstacle avoidance.

A result which has not been reported so far predicts the angle of the wall relative to the robot's approach, in addition to its distance. It was hoped that the angle information might sharpen the map by allowing the gaussian representing the wall to be oriented appropriately. A new network is trained to predict which infrared sensor detected the wall, providing a rough indication of its orientation. As shown in Figure 13, the map obtained is reasonable, but not a distinct improvement over the map without the angle information (Figure 11).

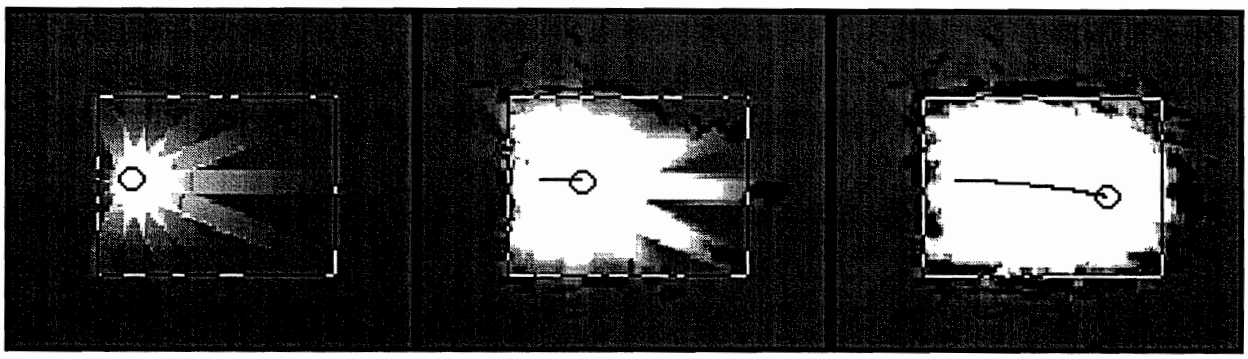


Figure 13. Prediction of angle as well as range (**8s**, median of 5 voters)

A related approach is seen in Ohya, Nagashima and Yuta[14]. Rather than using a neural network to predict the angle of the wall, the authors use a specially designed sonar sensor. The sonar has a single transmitter, but differs from the usual ultrasonic sensor by using two receptors. The timing difference of the echo returns indicates the orientation of the wall relative to the sensor. This sensor is used to augment grid maps with vector representations of walls.

6. CONCLUSION

There are other possibilities for reducing the error inherent in the raw sonar signal. Borenstein and Koren[15] have proposed an "error eliminating rapid ultrasonic firing" algorithm to reduce the number of erroneous readings returned by the sonars. While apparently yielding robust results, this method requires detailed control over the timing sequence of firing the sonars, a degree of control not always available. The method presented here requires nothing more than the sensory data already available.

Others have studied the application of neural networks to the fusion of robot sensor data[16]. In particular, Racz and Dubrawski[17] use the fuzzy ARTMAP network to classify a robot's position within the neighborhood of a door. In their study, however, the position of the robot within its environment is explicitly specified at the start of data collection. In our approach no such external information is provided. It is also interesting to compare our work to that of Thrun[18], in which a neural network is trained with backpropagation to predict the probability of occupancy of cells within a grid, by combining information from four sonars. Our approach is different from Thrun's in that we are estimating only a single frontal distance value rather than individual occupancy predictions, and in that we are fusing two sensor types. In addition, as with Racz and Dubrawski's work, Thrun's approach requires knowledge of the actual position of obstacles during training, whereas our approach does not require any external information. All of the results presented here are based on training and testing off-line using a data set that was collected on-line. The fact that training is done off-line should not be taken as evidence that the ARTMAP network could not have learned on-line. Rather, off-line learning is used here so that various fusion models may be tested against the same set of data, yielding comparable results. As mentioned above, although it took a few hours to collect the 10,167 data samples, it only took a few minutes to train the neural network on these data.

This work makes two novel contributions to the field of robotic sensory integration: it successfully applies an ARTMAP neural network to the task of integrating sensor data from multiple sources, producing a cleaner distance measure than is possible with either sensor type alone; and it does so with a training method that is self-supervised, allowing the robot to learn by itself the relationship between its sensory and odometric senses of distance.

7. ACKNOWLEDGMENTS

This work is supported by DARPA, the Office of Naval Research and the Naval Research Laboratory through grants ONR-00014-96-1-0772, ONR-00014-95-1-0409, and ONR-00014-95-0657.

8. REFERENCES

1. Luo, R.C. and Kay, M.G. (1989) Multisensor Integration and Fusion in Intelligent Systems, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19(5), pp. 901-931.

2. Huntsberger, T.L. (1992) Data Fusion: A Neural Networks Implementation, in Abidi, M.A. and Gonzalez, R.C., *Data Fusion in Robotics and Machine Intelligence* (1992) Academic Press, San Diego, CA. pp. 507-535.

3. Murphy, R.R. (1994) Sensor Fusion, in Arbib, M.A. *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, MA. pp. 857-860.

4. Murphy, R.R. (1996) Biological and Cognitive Foundations of Intelligent Sensor Fusion, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 26(1), pp. 42-51.

5. Martens, S., Gaudiano, P., and Carpenter, G.A. (1998) Mobile Robot Sensor Fusion with Fuzzy ARTMAP, to appear in *Proceedings of ISIC/CIRA/ISAS '98*

6. Carpenter, G.A., Grossberg, S., and Reynolds, J.H. (1991) ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network, *Neural Networks*, vol. 4, pp. 565-588.

7. Carpenter, G.A., Grossberg, S., Markuzon, N. Reynolds, J.H., and Rosen, D.B. (1992) Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional maps, *IEEE Transactions on Neural Networks*, vol. 3, pp. 698-713.

8. Elfes, A. (1989) Using Occupancy Grids for Mobile Robot Perception and Navigation, *IEEE Computer*, vol. 22(6), pp. 46-57.

9. Elfes, A. (1990) Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception, *Proceedings of the Sixth Conference on Uncertainty in AI*, pp. 60-70.

10. Winston, P.H. (1984) *Artificial Intelligence*, Addison-Wesley, Reading, MA.

11. Carpenter, G.A., Gopal, S., Martens, S., and Woodcock, C.E. (1997) Evaluation of Mixture Estimation Methods for Vegetation Mapping, Technical Report CAS/CNS-97-014.

12. Carpenter, G.A., and Markuzon, N. (1998) ARTMAP-IC and Medical Diagnosis: Instance Counting and Inconsistent Cases, *Neural Networks*, vol. 11(2), pp. 323-36.

13. Horswill, I. (1994) Visual Collision Avoidance by Segmentation, *Image Understanding Workshop Proceedings*, vol. 2, pp. 1135-1141.

14. Ohya, A., Nagashima, Y., and Yuta, S.I. (1994) Exploring Unknown Environment and Map Construction Using ultrasonic Sensing of Normal Direction of Walls, *Proceedings 1994 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 485-492.

15. Borenstein, J., and Koren, Y. (1995) Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidance, *IEEE Transactions on Robotics and Automation*, February 1995, Vol. 11(1), pp 132-138.

16. van Dam, J.W.M., Kröse, B.J.A., and Groen, F.C.A. (1996) Neural Network Applications in Sensor Fusion For An Autonomous Mobile Robot. *Proceedings of the International Workshop on Reasoning with Uncertainty in Robotics,* pp. 1-19.

17. Racz, J. and Dubrawski, A. (1995) Artificial Neural Network for Mobile Robot Topological Localization, *Robotics and Autonomous Systems*, Vol. 16, pp. 73-80.

18. Thrun, S. (1998) Learning Maps for Indoor Mobile Robot Navigation, *Artificial Intelligence* (Accepted for publication).