# Knowing Your Place in Real World Environments*

Tom Duckett and Ulrich Nehmzow
Department of Computer Science
University of Manchester
Manchester M13 9PL
United Kingdom
duckettt@cs.man.ac.uk, ulrich@cs.man.ac.uk

**Abstract**

*The topic of mobile robot self-localisation is usually divided into the sub-problems of global localisation and position tracking. Both are now well understood individually, but few mobile robots can deal simultaneously with the two problems in large, complex environments. While efficient solutions have been found for metric maps, topological maps have, by nature of their compactness, the potential for representing environments which are several orders of magnitude larger than those which can be tractably navigated using metric maps.*

*In this paper, we present a unified approach to global localisation and position tracking which is based on a topological map augmented with metric information. The method was validated through a series of experiments conducted in four real world environments, including its integration into a complete navigating mobile robot. Quantitative performance measures were used to assess localisation quality versus computational efficiency. The results show that our robot can localise and navigate reliably in large, complex environments using only minimal computational resources.*

## 1 Introduction

Perhaps the most fundamental competence required for navigation by a mobile robot is that of self-localisation (location recognition). Without the ability to identify locations reliably, a mobile robot will inevitably become lost, and therefore be unable to move purposefully between target locations.

A wide variety of localisation methods have been proposed, and a number of successful laboratory prototypes have been developed (see [1] for a detailed review). Some of these systems have been validated in larger environments, generally consisting of untreated, enclosed areas within public buildings (section 1.1). In real world environments, a navigating robot needs to cope with various problems of scalability, including perceptual aliasing (where several places are perceptually similar enough to be confused by the robot), unpredictable variations caused by other inhabitants of the environment, and limited computational resources — e.g., radio communication with external computers may not always be possible.

The topic is usually divided into two related sub-problems, namely *position tracking*, which assumes that the initial location of the robot is known, and *global localisation*, which entails being able to relocalise under global uncertainty, i.e., to recover from becoming lost. The first problem is well understood, and a number of successful approaches have been applied to the second problem in recent years. However, few systems can deal efficiently with both — one exception is the approach of Burgard *et al* [2], which uses a variable-resolution metric map to handle varying degrees of uncertainty in the robot's location estimates.

While efficient solutions have been found for metric maps, topological maps have, by nature of their compactness, the potential for representing environments which are several orders of magnitude larger than those which can be tractably navigated using metric maps.

In this paper, we therefore present a method for solving both the global localisation and position tracking problems using a topological map augmented with metric information. This method combines cross-correlation techniques for matching detailed sonar scans with a probabilistic algorithm for accumulating sensory evidence over time. The resulting system is thus able to determine not only the most likely place occupied by the robot in the topological map, but

*Proc. EUROBOT'99, 3rd European Workshop on Advanced Mobile Robots, Zurich, Switzerland, September 6-8, 1999.

also the most likely Cartesian position estimate of the robot for each of the possible places.

The method was implemented on a mobile robot and validated through a series of experiments conducted in four different real world environments. The results show that our robot can localise and navigate reliably in large, complex environments using only minimal computational resources. The approach is therefore well-suited to situations in which no radio link to external processors is available, and operation within real-time constraints is required.

### 1.1 Related Work

Yamauchi and Langley [13] developed a topological localisation method, where each place is represented by a local occupancy grid [10]. During localisation, a recognition grid is constructed from the robot's immediate sensor readings. A hill climbing procedure is then used to compare the recognition grid with the previously stored grids in the robot's map, searching the space of possible translations and rotations to find the best match. The main disadvantage of this approach is its high computational requirements; for example, relocalisation using 43 stored grids takes 5 minutes on a Decstation 3100.

A related approach is that of Weiss and von Puttkamer [12], using cross-correlation of laser scans to identify both the place occupied by the robot and the robot's position within that particular place. In this approach, however, the scans are reduced to histograms before matching takes place; firstly, angle histograms are convolved to address the problem of self-orientation, then $x$ and $y$ histograms are used to determine robot's position. We therefore apply histograms to overcome the high cost of the matching local occupancy grids.

Neither of the above approaches is guaranteed to solve the global localisation problem in environments with high levels of perceptual aliasing. A variety of methods have been proposed for resolving perceptual ambiguity by accumulating sensory evidence over time. The most popular is the probabilistic approach known as Markov localisation (e.g., [2, 5, 7, 8], *etc.*), which can be applied to either topological or metric maps.

The main paradigm for probabilistic navigation using topological maps is that of Hidden Markov Models, and their extension to Partially-Observable Markov Decision Processes [7, 8]. Here, the robot maintains a probability distribution over a set of discrete locations, known as the robot's belief. Similarly, possible landmarks and actions are typically defined according to a set of human-defined categories, e.g., possible landmarks might be 'doors', 'junctions', *etc.*, and possible actions might be 'Go North', 'Go West', *etc.* Our approach differs from other topological methods in that possible location estimates are continuous valued Cartesian coordinates, actions are described by arbitrary displacements within Cartesian space, and landmarks are defined by arbitrary grid patterns.

Probabilistic localisation methods using high resolution metric maps have the advantage of high accuracy, but typically require a great deal of computational resources. An elegant solution to this problem is provided by Burgard *et al* [2], where a variable resolution mapping strategy is used to trade off global uncertainty against accurate positioning. The approach uses Markov localisation to identify possible sub-areas of the whole state space in which the robot might be located, then position tracking is carried out only on these sub-areas, "zooming in" to a higher level of resolution when the robot has a high degree of certainty in its location. Our approach differs in that the full state space in the topological map is always searched, and efficient matching algorithms are used to resolve the problem of limited computational resources.

## 2 The Robot

Figure 1: The Nomad 200 Mobile Robot "FortyTwo". A flux gate compass was used to keep the turret, and therefore the sensors, at a constant orientation during map building.

The experiments presented here were conducted using the Nomad 200 robot shown in figure 1. The robot is equipped with sixteen ultrasonic range-finding sensors (range up to $6.5m$) and sixteen infrared sensors (range up to $60cm$) spaced at 22.5 degree intervals around its turret, which can rotate independently relative to the base of the robot (the camera shown was not used here). Two other motors located in the base of the robot are used to control the translational and rotational movement of the robot.

## 3 Self-Orientation

A separate behaviour was used to rotate the robot's turret at small speeds in the direction of 'North', as indicated by a flux gate compass. The effect of this behaviour was to smooth out local fluctuations in the magnetic field of the robot's environment. Using the compass in this way gave the robot a single view of each location, i.e., the appearance of locations to the robot depended on the robot's position alone, not its orientation.

While this method is robust in dealing with minor variations in the magnetic field, severe compass errors caused by ferrous building materials could pose a problem in some environments. A more reliable compass sense could be obtained by integrating perceptual information from the robot's exteroceptive and proprioceptive sensors, as in the self-orientation system described by Li *et al* [9], or by using correlation with a vision-based map of the ceiling as in Thrun *et al* [11].

### 3.1 Compass-based Odometry

Instead of using the robot's rotational wheel encoders for the on-line dead reckoning, we used the relative angular displacement of the turret against the base of the robot. This had the effect of removing the accumulated angular drift affecting the robot's raw odometry (see figure 2), because the turret was anchored to 'North' by the compass sense. Using compass-based odometry leaves a translational drift error of approximately 2-5% of distance travelled.

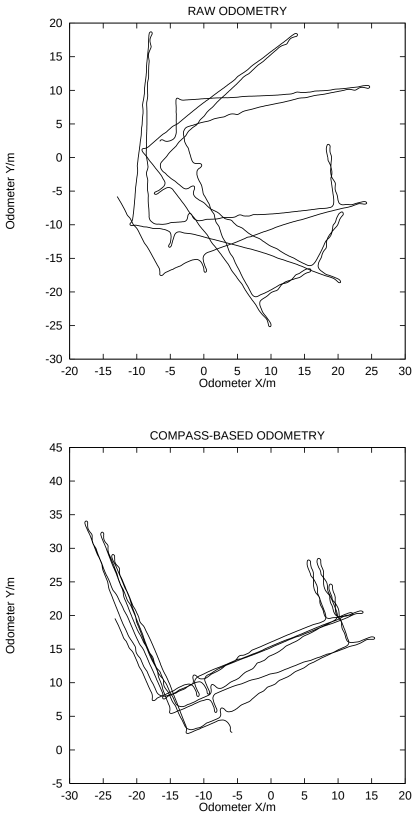


Figure 2: a.) Raw Odometry. b.) Compass-Based Odometry. The accumulated rotational drift in the robot's raw odometry was removed on-line using the compass sense.

## 4 Representation

The robot's environment model consists of a set of $N$ stored places, the centre of each place $i$ being associated with a Cartesian coordinate $(x_i, y_i)$. Landmark information is also attached to each of the places as follows. Firstly, the robot takes a detailed sonar scan at its current location and a local occupancy grid consisting of $64 \times 64$ cells is then constructed (as in Yamauchi and Langley [13]). However, in our system, we do not store or match the occupancy grids themselves. Instead, each grid is reduced to a pair of histograms (one in $x$, and one in $y$), which is then used as a stored signature for that place in the topological map (see figure 3).

Each occupancy grid cell represents an area of $15 \times 15cm$, and is considered as being in one of three possible states; occupied (O), empty (E) or unknown (U), depending on the corresponding probability of occupancy for that cell, i.e.,

$$State(c_{xy}) = \begin{cases} O & \text{if } p(c_{xy}) > 0.5 \\ U & \text{if } p(c_{xy}) = 0.5 \\ E & \text{if } p(c_{xy}) < 0.5 \end{cases}$$

where $p(c_{xy})$ refers to probability of occupancy for the cell at column $x$ and row $y$. One histogram is then derived by adding up the total number of occupied, empty and unknown cells in each of the 64 columns, and the other by adding up the totals for each of the 64 rows.

The robot's location model consists of a set of location hypotheses $H = \{h_1, ..., h_N\}$, one for each place $i$. Each hypothesis consists of a Cartesian coordinate $(x_{h_i}, y_{h_i})$, a probability value $p(h_i)$, which reflects the robot's belief in that place being its true location, and a variance measure used for Kalman filtering (each lo-

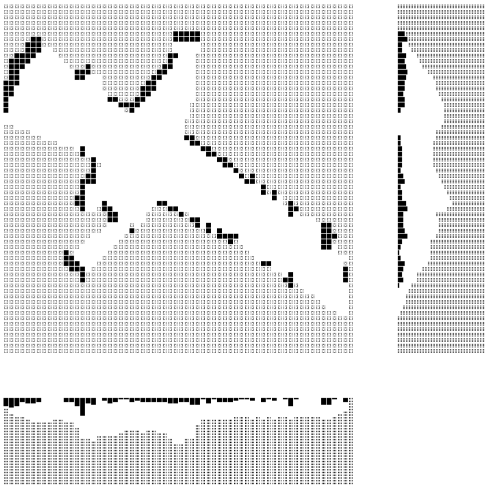

Figure 3: Example Occupancy Grid and Histograms. Occupied cells are shown in black, empty cells in white and unknown cells in grey.

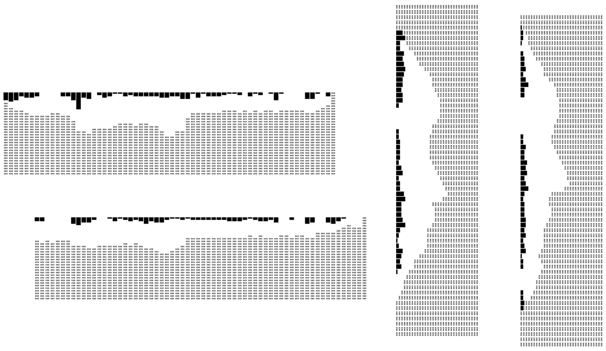

Figure 4: Landmark Recognition. The $x$ and $y$ histograms constructed from the current sensor data are convolved with the corresponding histograms stored in the robot's map.

cation hypothesis is modelled by a separate Gaussian distribution). The initial values for the probability distribution and coordinates are obtained using the histogram matching procedure in section 5 and updated using the algorithm in section 6.

## 5 Spatial Evidence

To begin localisation, the robot takes a new sonar scan. Again, the resulting occupancy grid is processed to produce a pair of histograms. For each stored place $i$ in the robot's map, the matching procedure yields two useful quantities:

1. The strength of the match between the current and stored histograms — this is used to provide a likelihood $L(S \mid h_i)$ of obtaining the current sensor readings $S$ from each place hypothesis $h_i$.

2. The most likely displacement $(rx_i, ry_i)$ of the robot in Cartesian space from the centre of the stored grid pattern, i.e., the position in which the sonar scan for that place was originally taken.

The first quantity is derived from the product of two separate metrics; one obtained by convolving the current and stored $x$ histograms, and the other by convolving the respective $y$ histograms (see figure 4). The strength of the match between two histograms $T^a$ and $T^b$ is calculated using the following evaluation function

$$Match(T^a, T^b) = \sum_j \left[min(O^a_j, O^b_j) + min(E^a_j, E^b_j) + min(U^a_j, U^b_j)\right],$$

where $O_j$, $E_j$ and $U_j$ refer to the number of occupied, empty and unknown cells contained in the $j$th element of histogram $T$. In the convolution, the stored histogram is kept stationary and the recognition histogram is translated against it, using the above function to calculate the best match over the 64 elements of the stored histogram. Any non-overlapping elements in the recognition histogram due to the translation are assumed to consist entirely of unknown cells.

Finally, having obtained the most likely translations for matching the two sets of histograms, i.e., the translations producing the highest match scores, the relative displacement $(rx_i, ry_i)$ of the robot within each place $i$ is obtained by multiplying by the dimensions of one grid cell (i.e., $15 \times 15cm$). The coordinates for each $h_i$ are then obtained using

$$x_{h_i} = x_i + rx_i,$$
$$y_{h_i} = y_i + ry_i,$$

i.e., by combining the coordinates of the place centre and the relative displacement values produced by histogram matching.

## 6 Temporal Evidence

After carrying out the above matching process, the place which yielded the highest match score could simply be taken as the winner, as in Yamauchi and Langley [13]. However, because of factors such as perceptual aliasing and sensor noise, the best matching place is not always the true location of the robot. To overcome this problem, we use a succession of sonar scans taken from different positions over time, and apply the following algorithm for accumulating sensory evidence over time.

At each iteration of the localisation algorithm, a new set of location hypotheses $H' = \{h'_1, ..., h'_N\}$ is constructed from a new sonar scan as described above. The coordinates of each of the old hypotheses $h_i$ are then updated using

$$x_{h_i}(t) = x_{h_i}(t-1) + \triangle x,$$
$$y_{h_i}(t) = y_{h_i}(t-1) + \triangle y,$$

where the vector $(\triangle x, \triangle y)$ refers to the robot's own displacement in Cartesian space as observed by the robot using its odometry since the previous iteration.

A matching process between the two sets $H$ and $H'$ then follows. This attempts to match each new hypothesis $h'_j$ to its one most likely equivalent hypotheses $h_i$ from the previous iteration. Each $h'_j$ is therefore compared to each $h_i$, and the quality of the match is obtained using

$$Eval(h'_j, h_i) = G\left(\|(x_{h'_j}, y_{h'_j}) - (x_{h_i}, y_{h_i})\|\right) p(h_i),$$

where the function $G(z) = e^{-\eta z^2}$, and the constant $\eta = 0.25$ in these experiments. For each $h'_j$, the best matching prior hypothesis is denoted by $h_{j*}$, and is used to provide the prior probability $p_{prior}(h'_j)$, by normalising the values produced by the above evaluation function over all $j$. A new probability distribution over $H'$ is then calculated using Bayes' theorem as

$$p_{posterior}(h'_j) = \frac{L(S_c \mid h'_j) p_{prior}(h'_j)}{\sum_k L(S_c \mid h'_k) p_{prior}(h'_k)},$$

A separate Kalman filter is used to update the Cartesian coordinates of each $h'_j$, taking into account the coordinates of both $h'_j$ and $h_{j*}$ (see [3] for details). The robot then continues to explore, periodically taking a new sonar scan and updating its estimate of its true location by repeating the above process.

| | Description | Approx. Size | $N$ |
|---|---|---|---|
| A | T-shaped hallway | $16m \times 13m$ | 25 |
| B | Conference room | $16m \times 11m$ | 38 |
| C | L-shaped corridor | $34m \times 33m$ | 68 |
| D | Long straight corridor | $53m \times 3m$ | 52 |

Table 1: Characterisation of Environments. The number of places $N$ in the respective maps is also shown.

| | Occupancy Grids | Histogram Matching | Nearest Neighbour |
|---|---|---|---|
| A | 0.764 | 0.807 | 0.692 |
| B | 0.892 | 0.899 | 0.809 |
| C | 0.721 | 0.696 | 0.623 |
| D | 0.598 | 0.497 | 0.407 |
| $\mu$ | 0.744 | 0.725 | 0.633 |
| $T$ | $13051t$ | $31t$ | $t$ |

Table 2: Localisation Quality $U(L \mid R)$ for each environment and the mean value $\mu$ over all 4 environments, and Mean Time Per Place Match $T$, where $t = 1.83 \times 10^{-5}s$ as measured on a Sparcstation 10.

## 7 Experiments

### 7.1 Landmark Recognition

In this experiment, the question asked was "Which landmark recognition mechanism is best suited for self-localisation by a mobile robot?" We compared the performance of occupancy grid matching[1] with that of histogram matching and a third "baseline" method for identifying landmarks, namely a simple 'nearest neighbour' classifier. In the latter system, the sensor patterns are normalised, and the dot product is used to determine the best matching place for the current sensor data.

To assess localisation quality, we used the following entropy-based statistic known as the *Uncertainty Coefficient of $L$ given $R$*, which measures the extent to which the robot's response $R$ predicts its true location $L$ (see also [3]). This is defined by

[1]The self-orientation component of the occupancy grid classifier [13] was disabled here, using our compass sense instead, in order to enable a fair comparison between systems.

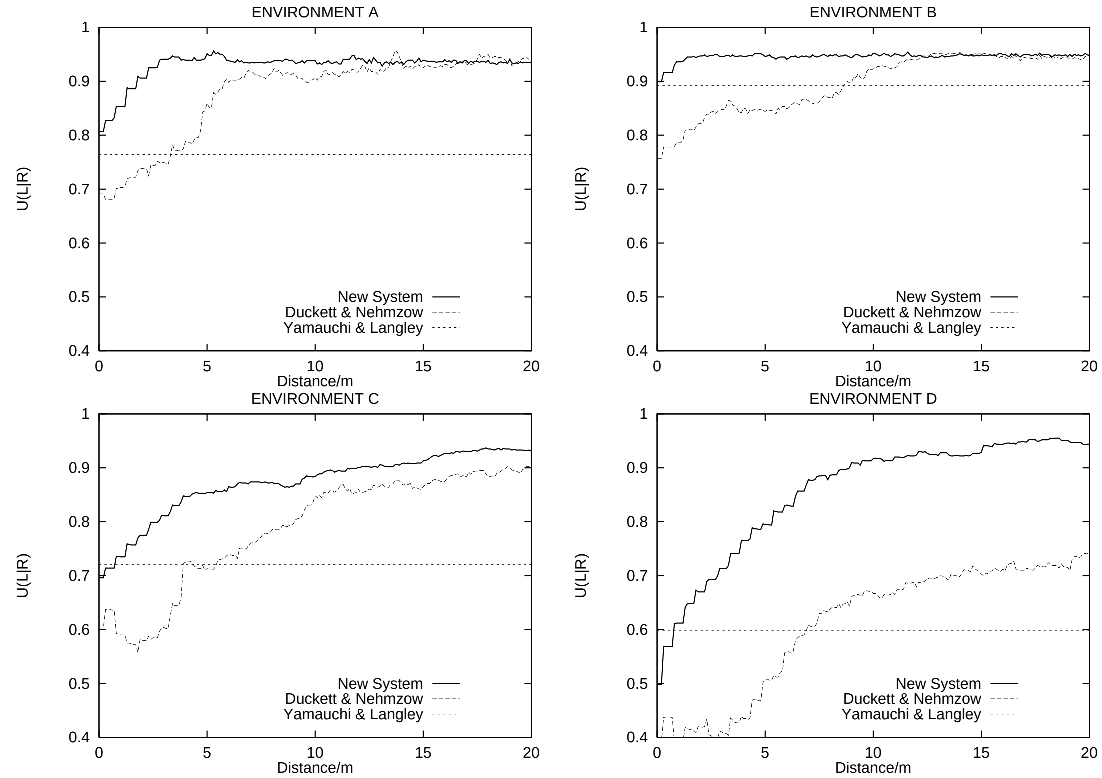


Figure 5: Localisation Quality $U(L \mid R)$ Over Distance Travelled. Here, we measure the worst case relocalisation performance of a wall-following robot after becoming lost in the four environments described in Table 1.

$$
\begin{aligned}
U(L \mid R) &\equiv \frac{H(L) - H(L \mid R)}{H(L)}, \\
H(L) &= -\sum_j p_{\bullet j} \ln p_{\bullet j}, \\
H(L \mid R) &= -\sum_{i,j} p_{ij} ln \frac{p_{ij}}{p_{i\bullet}},
\end{aligned}
$$

where $p_{\bullet j} = \sum_i p_{ij}$, $p_{i\bullet} = \sum_j p_{ij}$, and $p_{ij}$ refers to the probability of the robot's response being $i$ when the robot's true location was $j$. A value of $U(L \mid R) = 1$ implies that the robot's response always predicts its true location, while $U(L \mid R) = 0$ implies that the response never predicts the true location. To assess computational efficiency, we measured the mean time taken to localise on a Sparcstation 10.

The experiment was repeated in four different environments, as described in table 1. The same map and recorded robot data was used to test each of the different systems in each environment. The places in the map were spaced at regular intervals in Cartesian space ($1m$). The apparatus and experimental procedure used was that described in [3]. The results in table 2 show clearly that there is little loss in performance (i.e., loss in information) caused by using histograms rather than full occupancy grids, and that this is achieved at a greatly reduced cost in processing requirements.

## 7.2 Global Localisation

In this experiment, we measured the ability of the robot to recover from being lost, i.e., to relocalise under global uncertainty, using wall-following for exploration. The performance of the new localisation system was compared to that of occupancy grid matching [13] and our previous approach [3]. The latter system used an equivalent mechanism to the 'nearest neighbour' classifier described above for spatial evidence and the iterative localisation algorithm described for accumulating temporal evidence. The results shown in figure 5 indicate the improvement in performance obtained using the new approach.

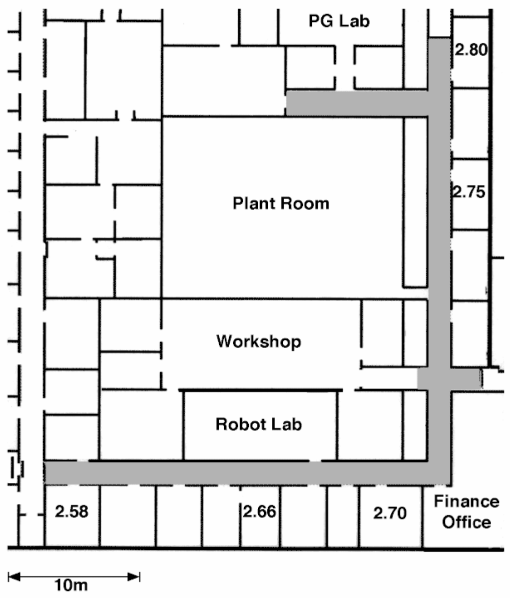


Figure 6: Corridor Environment. The navigation experiment was carried out in the shaded area shown, which is a busy thoroughfare in our computer building (environment C in Table 1).

## 7.3 Application to Robot Navigation

Finally, we validated the new localisation system through its integration into a complete navigating mobile robot (see [4] for more details). Here, we attempted to measure the ability of the robot to perform a delivery task, navigating between a series of user-specified locations shown in figure 6. We compared the performance of the robot when navigating from a known starting location, i.e., using position tracking, to its performance under global uncertainty, i.e., being lost at the start of each navigation attempt.

In this experiment, the robot was supplied with a map which had been previously learned by the robot. Global consistency was maintained in the map using a relaxation algorithm which is analogous to the spring system model described by Golfarelli *et al* [6]. Dijkstra's algorithm was used for path planning and the robot's motor actions were determined from the most likely position estimate produced by the localisation algorithm. In order to assess the ability to navigate under limited resources, the ethernet radio link to the robot was switched off, and all computation was carried out on board the robot's 486 processor. The robot then had to find ten successive routes through the corridor environment (table 3). For the lost robot, the localisation mechanism was disabled and reinitialised at the start of each navigation attempt. The whole experiment was repeated ten times.

| | From | To |
|---|---|---|
| 1 | Workshop | Robot Lab |
| 2 | Robot Lab | Room 2.58 |
| 3 | Room 2.58 | Room 2.66 |
| 4 | Room 2.66 | Finance Office |
| 5 | Finance Office | Plant Room |
| 6 | Plant Room | Room 2.80 |
| 7 | Room 2.80 | Postgraduate Lab |
| 8 | Postgraduate Lab | Room 2.75 |
| 9 | Room 2.75 | Room 2.70 |
| 10 | Room 2.70 | Workshop |

Table 3: Routes Traversed. (See also Figure 6).

The overall frequency of successful navigation attempts was 100% using position tracking, and 92% for the lost robot. These results show firstly that under normal operation using position tracking the navigation system we have developed is highly reliable, always managing to reach the desired goal locations. However, the lost robot was not quite as successful, being affected by various errors as it attempted to recover its position.

Perceptual aliasing or sensor noise could sometimes lead to the robot believing falsely that it had reached the goal location, when in fact more exploration was required to relocalise successfully. This tended to happen when the robot started to navigate from somewhere in the middle of a long, featureless corridor, e.g., on route 6. The lost robot could also become trapped in a local minimum, e.g., moving backwards and forwards between two incorrectly identified places, and taking a very long time to accumulate enough sensory information to be able to relocalise correctly. For example, two attempts at route 3 were abandoned after 15 minutes because the robot had still not managed to relocalise and find the goal.

These errors can be explained by the fact that the robot's motor actions depend upon its current position estimate, so there is no guarantee that the robot will take appropriate actions to relocalise itself should it become lost. One solution to this problem would be to choose actions designed to improve localisation quality, as in Fox *et al* [5]. Alternately, the robot could revert to wall-following in order to relocalise whenever it believed it might be lost, e.g., using the entropy-based novelty filter described in [5] to detect a decrease in localisation quality.

## 8 Summary

In this paper, we have introduced an approach for combining global localisation and position tracking using a topological map augmented with metric information. We have shown through the above experiments that our robot can localise and navigate reliably in large, complex environments with only minimal computational resources using this method.

## 9 Conclusions and Future Work

To achieve reliable self-localisation, a mobile robot must depend on its ability to recognise places using landmarks rather than dead-reckoning. Towards this end, Yamauchi and Langley's approach [13] overcomes some of the problems of using occupancy grids for mobile robot navigation. A globally consistent metric map is not required, because a separate grid is used to represent each place visited by the robot. Also, the approach does not depend critically on accurate range-finding sensors; similar perceptions produce similar grid patterns, despite the specular effects associated with sonar sensors.

However, two major problems remained with this approach until now. Firstly, occupancy grids require a high level of computation, and secondly, they suffer from perceptual aliasing. We have overcome the first of these problems by introducing a matching procedure based on the use of histograms, rather than entire occupancy grids. To tackle perceptual aliasing, we have applied a probabilistic localisation algorithm for accumulating sensory evidence over time, which also increases the robustness of the system in dealing with transient changes to an environment such as moving people.

Localisation in the manner described in this paper requires geometrically consistent coordinates in the robot's map. These coordinates cannot be obtained using dead-reckoning, due to the cumulative drift errors which affect odometry. We are currently investigating mechanisms for obtaining Cartesian coordinates in a topological map using only local information, i.e., the relative displacement of the robot between topologically connected places.

## References


[1] J. Borenstein, H. R. Everett and L. Feng, *Navigating Mobile Robots: Systems and Techniques*, A. K. Peters Ltd., Wellesley, MA, 1996.

[2] W. Burgard, A. Derr, D. Fox, and A. B. Cremers, *Integrating Global Position Estimation and Position Tracking for Mobile Robots: The Dynamic Markov Localization Approach*, Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 1998.

[3] T. Duckett and U. Nehmzow, *Mobile Robot Self-Localization and Measurement of Performance in Middle-Scale Environments*, Robotics and Autonomous Systems 24 (1–2), pp. 57-69, 1998.

[4] T. Duckett and U. Nehmzow, *Self-Localisation and Autonomous Navigation by a Mobile Robot*, Proc. TIMR UK 99, Bristol UK, Manchester University Technical Report UMCS-99-3-1, ISSN 1361-6161, 1999.

[5] D. Fox, W. Burgard and S. Thrun, *Active Markov Localization for Mobile Robots*, Robotics and Autonomous Systems, to appear, 1998.

[6] M. Golfarelli, D. Maio and S. Rizzi, *Elastic Correction of Dead-Reckoning Errors in Map Building*, Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 1998.

[7] J. Hertzberg and F. Kirchner, *Landmark-Based Autonomous Navigation in Sewerage Pipes*, Proc. EUROBOT 1996, IEEE Computer Society Press, pp. 68-73, 1996.

[8] S. Koenig and R. G. Simmons, *A Robot Navigation Architecture Based on Partially Observable Markov Decision Process Models.*, in D. Kortenkamp, R. P. Bonasso and R. Murphy, eds., *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, AAAI Press, 1998.

[9] G. Li, B. Svensson and A. Lansner, *Self-orienting with On-line Learning of Environmental Features*, AISB-97 Workshop on Spatial Reasoning in Mobile Robots and Animals, Manchester University Technical Report UMCS-97-4-1, ISSN 1361-6161, 1997.

[10] H. Moravec and A. Elfes, *High Resolution Maps from Wide Angle Sonar*, Proc. IEEE Int. Conf. Robotics and Automation, 1985.

[11] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Daellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte and D. Schulz, *MINERVA: A Second-Generation Museum Tour-Guide Robot*, Proc. IEEE Int. Conf. Robotics and Automation, 1999.

[12] G. Weiss and E. v. Puttkamer, *A Map Based on Laserscans Without Geometric Interpretation*, Intelligent Autonomous Systems-4 (IAS-4), Karlsruhe, Germany, March 1995, pp. 403-407.

[13] B. Yamauchi and P. Langley, *Place Recognition in Dynamic Environments*, Journal of Robotic Systems 14 (2), pp. 107-120, 1997.