Robotics and Autonomous Systems 24 (1998) 57–69

Robotics and Autonomous Systems

Mobile robot self-localisation and measurement of performance in middle-scale environments

Tom Duckett [1], Ulrich Nehmzow *

Department of Computer Science, University of Manchester, Manchester M13 9PL, UK

Received 31 October 1997; accepted 30 April 1998

Abstract

This paper addresses the question of self-localisation in autonomous mobile robot navigation, i.e., the task of identifying places after previous exploration and map building by the robot. We present a novel localisation system which accumulates both exteroceptive and proprioceptive sensory evidence over time to localise, without requiring prior knowledge of the robot's position. We show that the system relocalises successfully on a real robot in middle-scale environments containing transient changes such as moving people.

In addition, a general performance metric and a standard experimental procedure are introduced, allowing disparate localisation systems to be compared on the same robot in the same environment. To demonstrate the utility of the approach taken, we test the evidence-based localisation system in six different environments, comparing its performance to that of localisation using dead reckoning or currently observable landmarks alone. In addition, the results provide us with some useful quantitative measures for characterising different environments.

Keywords: Mobile robot navigation; Mobile robot self-localisation; Quantitative analysis of robot behaviour

1. Introduction

This paper addresses firstly the problem of self-localisation in autonomous mobile robot navigation, i.e., the task of identifying places after previous exploration and map building by the robot. In particular, we are concerned with the worst case of relocalisation where the robot has no prior knowledge of its position, e.g., after becoming lost through some arbitrary circumstance. Piasecki [7] calls this global localisation. In Section 3, we describe our own novel approach to this problem.

At the present time, various localisation systems have been proposed for navigating mobile robots. However, quantitative comparison of the different approaches is largely impossible, because researchers use different robots, apply different quality measures, and conduct different experiments to validate their implementations. In Sections 4 and 6, we therefore introduce a general method for quantifying localisation performance, allowing disparate localisation systems to be compared on the same robot in the same environment.

Thirdly, to be really useful, we believe that navigating robots must be tested in "real world" environments, rather than specially constructed environments in the robotics laboratory. Here, we refer to three different

* Corresponding author. E-mail: ulrich@cs.man.ac.uk.

[1] E-mail: duckettt@cs.man.ac.uk.

0921-8890/98/$ – see frontmatter
PII: S0921-8890(98)00022-0

Fig. 1. *The Manchester ‘FortyTwo’*. Exploration was carried out using the translational and rotational motors located in the base of the robot. A separate behaviour was used to rotate the turret at small speeds towards “North” as indicated by a compass (see Section 5 for full details).

categories of navigation taken from experimental biology (see e.g., [10]).

1. *Small-scale.* Navigating within the vicinity of a home or goal location (e.g., view-based navigation by rats).
2. *Middle-scale.* Leaving the vicinity of the home or goal, and navigating within the wider environment (e.g., foraging desert ants).
3. *Large-scale.* Navigating over very large distances between different environments (e.g., intercontinental navigation by migratory birds).

In mobile robotics, we use “small-scale” to refer to navigation within the confines of the robot lab. “Middle-scale” means leaving the robot lab and entering unmodified public areas, e.g., corridors and offices, which are subject to unpredictable variations. In doing so, any localisation system will inevitably encounter problems of scalability, including perceptual aliasing, [2] computational efficiency and changes to the environment such as moving people.

For the Nomad 200 used in these experiments (see Fig. 1), middle-scale means navigating in environments of several hundred square metres in size. In Sections 7 and 8, we therefore measure the performance of our localisation system in six different environments around Manchester University’s Computer Building.

[2] This refers to the situation where several places are perceptually similar enough to be confused by the robot.

2. The lost robot problem

In mobile robot navigation, self-localisation refers to the task of establishing one’s position with respect to the known world. In particular, this paper is concerned with the worst case of relocalisation, where the robot has no idea of its approximate location, having become lost through some arbitrary circumstance. In this situation, the robot is unable to localise using past experience, i.e., no a priori estimate of position is available.

As we are interested in the use of autonomous mobile robots in unmodified environments, we avoid the use of pre-installed maps or external devices such as markers or beacons for localisation. To be fully autonomous, the robot must therefore rely on its own perceptions to localise. The space of possible perceptions available to the robot for carrying out this task may be divided into two categories:

1. *Exteroception.* The robot’s current perceptions of the outside world. A robot’s exteroceptors may include range-finding sensors, tactile sensors, video cameras, etc.
2. *Proprioception.* The robot’s perceptions of its own interactions with the world. In particular, odometry refers to the proprioceptor mechanism used for dead reckoning in mobile robots.

A localisation system based solely on proprioception will be unsuitable for two reasons. Firstly, as no a priori information will be available to the robot when trying to relocalise, it will not be possible to initialise dead reckoning. Secondly, any proprioceptive sensor system will be subject to drift errors, which cannot be compensated through proprioception.

Exteroception offers potential solutions to these problems, allowing places to be identified and drift errors to be corrected on the basis of perceived environmental features. However, this also introduces another problem; an exteroception-based localisation system will be affected by perceptual aliasing sooner or later in reasonably complex environments. Misclassification of places can also be caused by sensor noise or perceptual changes, e.g., moving people, in an environment.

This research therefore addresses the question of how the different forms of perception available to the robot might be combined to provide a solution to the lost robot problem. In the experiments presented here, the robot had to relocalise after being moved to a randomly chosen location, its sensors being disabled during that move. In addition, moving people were added to one of the environments during localisation.

3. Evidence-based localisation

The main principle of the evidence-based localisation system presented here is that, first, the robot builds a map of its environment. Having completed the mapping phase, the robot is then moved to a random position in the environment, its sensors being disabled during that move. The robot then begins to explore, and attempts to relocalise on the basis of evidence accumulated from both exteroceptive (sonar and infrared) and proprioceptive (relative changes in odometry) sensory input. Using this sensory evidence, the robot is able to choose between a competing set of place memories. A winning location "hypothesis" emerges once a perceptually unique path has been traced through the robot's map.

During localisation, the robot's confidence in its estimated position is stored as a probability distribution over a set H of possible location hypotheses. The current set of hypotheses is generated solely on the basis of the robot's exteroceptive sensory input, i.e., landmarks, so the robot can relocalise successfully without requiring prior knowledge of its position.

The mechanism used for localisation is described in Section 3.2. This works by comparing old and new information at each iteration, or in other words, by considering the *changes* in the moving robot's perceptions over time. The first stage of localisation is to determine the stored place memory which most closely resembles the current sonar and infrared input. Because of perceptual aliasing, typically a number of place memories emerge as likely candidates for the robot's current position. The robot then performs an exploratory movement of an arbitrary nature, whose *relative* displacement in Cartesian space is observed by the localisation system. The current perception, previous perception and displacement are then used to narrow down the list of likely candidates for the robot's current location, using Bayes rule to update the probability distribution over H.

3.1. Map building

The map constructed by the robot consists of a set M of place memories, each place m being associated with a Cartesian coordinate (x_m, y_m) obtained by averaging the robot's odometer readings within that particular place (see Section 6.4 for details of the odometry correction method applied here). Each place m also stores a normalised sonar signature S_m and a normalised infrared signature I_m, obtained by averaging the robot's sensor readings within that particular place.

A new place is entered into the map whenever the robot has travelled by more than a prespecified distance D from the nearest stored place in the map, [3] or when the current sonar and infrared readings differ from the stored signatures of the current place by more than a threshold T. The current sensor difference d_m for place m is calculated using the following weighted sum, where S_c and I_c refer to the normalised current sonar and infrared readings, respectively (in these experiments, $D = 0.375\,\mathrm{m}$, $T = 1.5$, $w_\mathrm{son} = 1$ and $w_\mathrm{ir} = 1$):

$$d_m = w_\mathrm{son}\|S_\mathrm{c} - S_m\| + w_\mathrm{ir}\|I_\mathrm{c} - I_m\|. \tag{1}$$

3.2. Localisation mechanism

A schematic diagram of the localisation mechanism is shown in Fig. 2. The algorithm explained here takes as input a prior set of location hypotheses $H = \{h_1, h_2, \ldots, h_n\}$ from the previous iteration. On initialisation, this set will be empty. A probability distribution $P = \{p(h_1), p(h_2), \ldots, p(h_n)\}$ is associated with set H. Each hypothesis $h_i \in H$ is also associated with a Cartesian coordinate (x_{h_i}, y_{h_i}) and a variance measure v_{h_i} used for Kalman filtering.

Localisation begins by formulating a new set of candidate hypotheses H' based on the current sonar and infrared readings. This is done by finding the current sensor difference d_m for all the possible locations m contained in the map M, again using Eq. (1). Only

[3] The maximum distance between stored places will therefore be $2D$, due to the averaging of (x, y) coordinates within places.

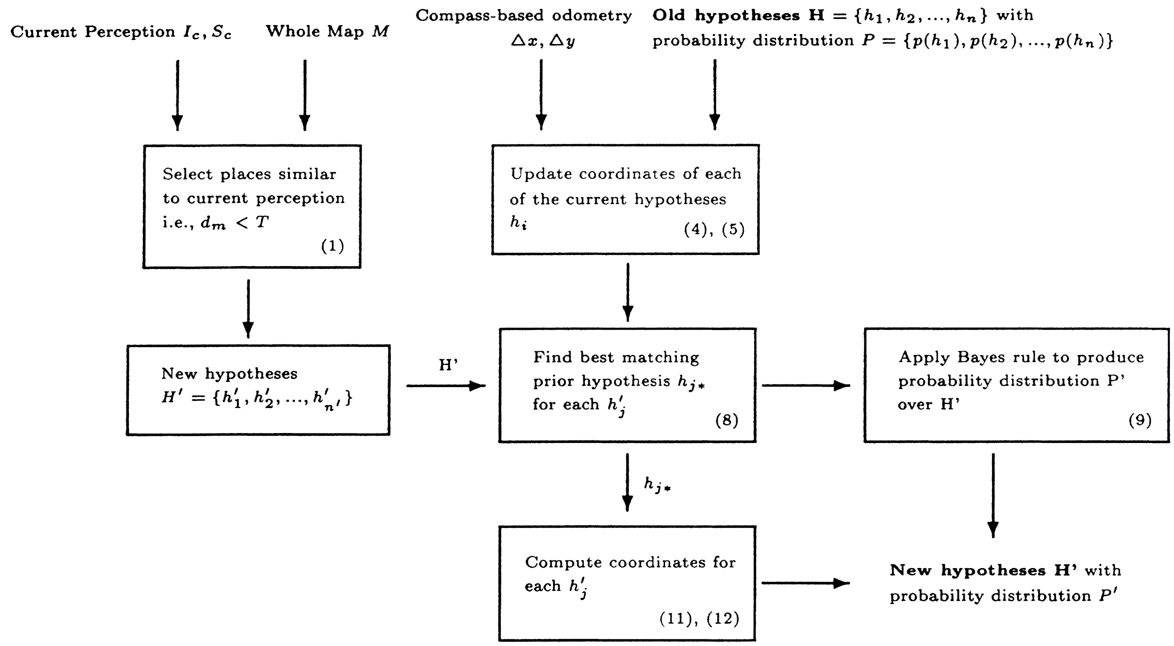

Fig. 2. The localisation mechanism.

the places where $d_m < T$ are included in set H'. For each $h'_j \in H'$, the relative likelihood $L(S_c, I_c|h'_j)$ of obtaining the current sensor readings from that particular place is estimated using

$$L(S_{\mathrm{c}}, I_{\mathrm{c}}|h'_j) = \mathrm{Gaus}\left(\frac{d_m}{w_{\mathrm{ex}}}\right), \tag{2}$$

where the Gaussian function $\mathrm{Gaus}(z) = \mathrm{e}^{-z^2}$. A value of $w_{\mathrm{ex}} = 1.50$ was used for these experiments.

If the old set of hypotheses H is empty, as on initialisation, the probability distribution over the new set H' is calculated for all h'_j using

$$p(h'_j) = \frac{L(S_{\mathrm{c}}, I_{\mathrm{c}}|h'_j)}{\sum_k L(S_{\mathrm{c}}, I_{\mathrm{c}}|h'_k)}. \tag{3}$$

Otherwise, localisation proceeds as follows. Firstly, the coordinates (x_{h_i}, y_{h_i}) of each of the prior hypotheses h_i are updated using

$$x_{h_i}(t) = x_{h_i}(t-1) + \Delta x, \tag{4}$$

and

$$y_{h_i}(t) = y_{h_i}(t-1) + \Delta y, \tag{5}$$

where the vector $(\Delta x, \Delta y)$ refers to the robot's own displacement in Cartesian space as observed by the robot since the previous iteration (see Section 5.2 for details of the on-line compass-based odometry used here). In addition, the additional uncertainty due to odometer drift is modelled using

$$v_{h_i}(t) = v_{h_i}(t-1) + \gamma v_{h_i}(t-1), \tag{6}$$

where a value of $\gamma = 0.05$ was assumed for these experiments.

A matching process between the two sets H and H' then follows. This attempts to match each new hypothesis h'_j to its 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

$$\mathrm{Eval}(h'_j, h_i) = \mathrm{Gaus}\left(\frac{\|(x_{h'_j}, y_{h'_j}) - (x_{h_i}, y_{h_i})\|}{w_{\mathrm{pr}}}\right) p(h_i). \tag{7}$$

The constant w_{pr} effectively determines the relative weighting of proprioceptive and exteroceptive sensory information in the matching process; a value of $w_{\mathrm{pr}} = 4D$ was used in these experiments.

For each h'_j, the best matching prior hypothesis h_{j*} is therefore defined by

$$\forall j : \forall i \neq j* : \text{Eval}(h'_j, h_{j*}) > \text{Eval}(h'_j, h_i). \tag{8}$$

In the event of a tie, one of the best matching hypotheses is picked at random. (In practice, this is highly unlikely, and did not occur in any of the experiments presented here.) The values produced by the match evaluation function are then used to provide a prior probability for each h'_j according to

$$p_{\text{prior}}(h'_j) = \frac{\text{Eval}(h'_j, h_{j*})}{\sum_k \text{Eval}(h'_k, h_{k*})}, \tag{9}$$

and a new probability distribution over H' is calculated using

$$p_{\text{posterior}}(h'_j) = \frac{L(S_c, I_c|h'_j)\, p_{\text{prior}}(h'_j)}{\sum_k L(S_c, I_c|h'_k)\, p_{\text{prior}}(h'_k)}, \tag{10}$$

which is equivalent to Bayes rule.

A separate Kalman filter is then used to update the Cartesian coordinates of each h'_j, taking into account the coordinates of both h'_j and h_{j*}, according to the following equations:

$$x'_{h'_j} = x_{h_{j*}} + K[x_{h'_j} - x_{h_{j*}}], \tag{11}$$

$$y'_{h'_j} = y_{h_{j*}} + K[y_{h'_j} - y_{h_{j*}}], \tag{12}$$

$$v'_{h'_j} = v_{h_{j*}} - K v_{h_{j*}}, \tag{13}$$

where

$$K = v_{h_{j*}} / (v_{h_{j*}} - v_{h'_j}). \tag{14}$$

In these experiments, the variance $v_{h'_j}$ was always initialised to the same arbitrary value (we could probably produce a better model of the uncertainty here, though this would be outside the scope of this paper). Thus, an optimal position estimate is maintained for each of the possible locations in the robot's position model.

The robot then waits until it has moved by a distance $2D$ or the current sensor readings change by T before starting the next iteration of the localisation algorithm.

3.3. Related work

Kurz [2] associated stored sonar prototypes with Cartesian coordinates, and used a Kalman filter for position estimation. However, this approach depends on prior position knowledge and reliable landmark identification to localise.

Piasecki [7] used multiple hypothesis tracking and multiple Kalman filters. However, in this system, temporal evidence accumulation was restricted to a fixed time "window" because of the exponential growth in the number of solutions evaluated. Also, this system was tested only in simulation, and not on a real robot.

Oore et al. [6] used Bayes rule to accumulate temporal evidence over a grid of possible locations. However, this choice of representation means that this system would only be computationally tractable in real-time (on our robot at least) over small-scale environments. Again, this system was only tested in simulation.

By restricting the search to a small subset of the locations stored in the map, and using an efficient map representation (similiar to that of Zimmer [12]), we have produced a system which localises successfully on a real robot in middle-scale environments.

4. Measurement of performance

Various different measures have been used by mobile robotics researchers to assess localisation quality. Oore et al. [6] used absolute error, i.e., the distance between the robot's predicted and actual position in a Cartesian reference frame. However, this approach was only viable here because the authors tested their system in simulation. The exact position of a real moving robot is very difficult to measure, especially in middle-scale environments. Also, such a metric does not facilitate comparison with systems which produce a non-Cartesian response to their perceived location, e.g., Kohonen feature map [5].

Another approach is to evaluate a percentage correct figure for the system under investigation (see e.g., [11]). However, this calculation requires that the human observer is able to interpret the meaning of the robot's response with respect to the robot's environment. Such an interpretation would not be readily available in many cases, e.g., hippocampal [1] and self-organising systems [5], where the robot forms its own internal representation of the world.

In our approach, we therefore adopt a "black box" model (or stimulus-response model) of the robot's interactions with its environment. A performance metric based on the concept of mutual information [8] is detailed in the following section. This effectively measures the extent to which the robot's response predicts its true location, without requiring any semantic interpretation of the robot's internal world model by the observer.

4.1. Contingency analysis

The question we wish to ask with the metric is *"How much information does the robot's response provide about its true location?"*.

The basis of this method of quantitative analysis is a data structure known as a contingency table. In the example given in Fig. 3, a sample consisting of 100 data points has been collected. Each data point has two attributes; one corresponding to the location predicted by the robot (the robot's *response*, R), and the other to the actual location of the robot measured by an observer (the robot's *true location*, L). By convention, the rows of the table are used to represent the response, and the columns to represent the location. For example, Fig. 3 shows one cell containing 19 data points where the robot's response was measured as row 3 and the location as column 5.

Location (L)

Response (R)						
	0	2	15	0	1	18
	10	10	0	0	0	20
	0	2	1	0	19	22
	5	7	3	1	1	17
	0	0	0	23	0	23
	15	21	19	24	21	100

Fig. 3. *Example contingency table.* The rows correspond to the response produced by the particular localisation system under investigation, and the columns to the "true" location of the robot as measured by an observer. This table represents 100 data points, and also shows the totals for each row and column.

For a contingency table, the *row totals* for each response r, *column totals* for each location l and the *table total* are calculated using

$$N_{r\bullet} = \sum_{l} N_{rl}, \tag{15}$$

$$N_{\bullet l} = \sum_{r} N_{rl}, \tag{16}$$

and

$$N = \sum_{r,l} N_{rl}, \tag{17}$$

respectively, where N_{rl} refers to the number of data points contained in the cell at row r and column l. For the example table, the total of row 3 ($N_{3\bullet}$) is 22, the total of column 5 ($N_{\bullet 5}$) is 21, and the table total is 100.

The *row, column* and *cell probabilities* are then calculated using

$$p_{r\bullet} = \frac{N_{r\bullet}}{N}, \tag{18}$$

$$p_{\bullet l} = \frac{N_{\bullet l}}{N}, \tag{19}$$

and

$$p_{rl} = \frac{N_{rl}}{N}. \tag{20}$$

For the example table, the probability of a data point lying in row 3 is 0.22, the probability of a data point lying in column 5 is 0.21, and the probability of a data point lying in cell $(3, 5)$ is 0.19.

The next set of equations are used to calculate the entropy of the variables under consideration, i.e., the amount of information required to remove any uncertainty in these quantities. Here, we define the *Entropy of L* (Eq. (21)), the *Entropy of R* (Eq. (22)) and the *Mutual Entropy of L and R* (Eq. (23)).

$$H(L) = -\sum_{l} p_{\bullet l} \ln p_{\bullet l}, \tag{21}$$

$$H(R) = -\sum_{r} p_{r\bullet} \ln p_{r\bullet}, \tag{22}$$

$$H(L, R) = -\sum_{r,l} p_{rl} \ln p_{rl}. \tag{23}$$

In particular, we are interested in measuring the useful information provided by R in predicting the value of L. (We are not concerned with the reverse relationship; for example, if two responses both predict the same location, this should not have a negative impact on the metric.) Therefore, the *entropy of L given R* is obtained using

$$H(L|R) = H(L, R) - H(R), \tag{24}$$

where

$$0 \leqslant H(L|R) \leqslant H(L). \tag{25}$$

This last property (Eq. (25)) means that the range of values for $H(L|R)$ will be dependent on the size of the environment, because $H(L)$ increases as the number of location bins increases. For making comparisons between different environments, an alternative statistic is the *uncertainty coefficient of L given R* which is obtained using

$$U(L|R) \equiv \frac{H(L) - H(L|R)}{H(L)}, \tag{26}$$

where

$$0 \leqslant U(L|R) \leqslant 1. \tag{27}$$

More precisely, this gives us the *proportion* of the maximum possible information which R provides about L. A value of $U(L|R) = 0$ means that R provides no useful information about L, and implies that the robot's response never predicts its true location. A value of $U(L|R) = 1$ means that R provides all the information required about L, and implies that the response always predicts the true location. It should also be noted that the ordering of the rows and columns in the contingency table makes no difference to the outcome of this calculation.

5. The robot

The experiments presented here were conducted using the Nomad 200 mobile robot shown in Fig. 1. The robot has 16 sonar and 16 infrared sensors mounted 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.

Two reactive behaviours were used to control the robot's motor actions during these experiments. A wall-following behaviour was used to control the translational and rotational motors (see Section 6.2). A separate behaviour was used to control the turret as follows.

5.1. Compass sense

Here, the robot's turret was rotated at small speeds in the direction of "North" as indicated by the robot's compass. The effect of this behaviour was to smooth out local variations in the magnetic field of the environment.

Using the compass provides the robot with a single view of each location, i.e., the appearance of locations to the robot depends on the robot's position alone, not its orientation, regardless of any major variations in the magnetic field.

5.2. 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. Because the robot's turret was anchored to "North" by the compass sense, this had the effect of removing the accumulated angular drift affecting the robot's raw odometry (see Fig. 4), leaving a translational drift of approximately 2–5% of distance travelled (e.g., an error of 4 m over a route of 146 m).

6. Standard experiment

In this section, we describe an experimental procedure for evaluating different localisation systems on the same robot in the same environment which has the following characteristics. This includes a novel mechanism for tracking the true location of the robot (Section 6.4).

6.1. Middle-scale environment

The experiment requires a large environment, as we are interested in the robot's ability to recognise distinct

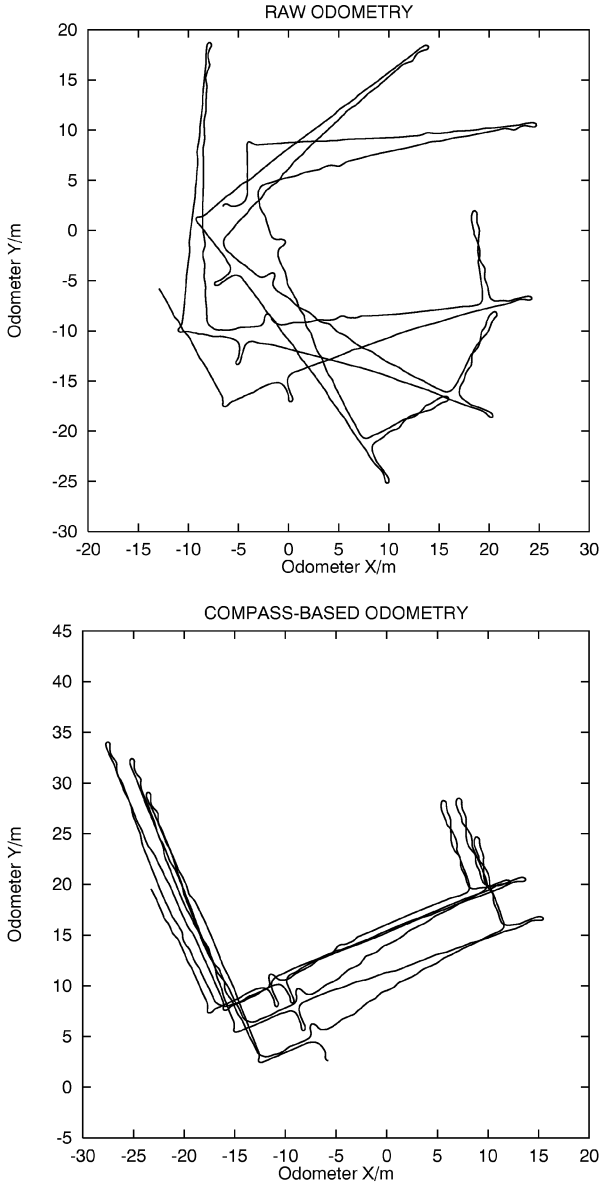

Fig. 4. *Compass-based odometry.* The accumulated rotational drift in the robot's raw odometry was removed on-line using a compass sense (see Section 5.2).

places in the "real world", rather than precise positioning over a small area. The experiments presented here were conducted in environments of several hundred square metres in size, with route lengths between 23 m and 146 m.

6.2. Exploration using wall-following

In our experiments, a reactive wall-following strategy was used for exploration with an average speed of $0.122\,\mathrm{ms}^{-1}$ (see [4] for details). Wall-following was used because it is easily reproducible, and also because the purely reactive nature of this strategy means that sensor data can be recorded and played back for later experiments, whilst preserving the full complexity of robot–environment interaction.

6.3. Trace-replay mechanism

As the robot explored its environment, a trace–replay mechanism (as in [3]) was used to record all of the sensory information available to the robot into a datafile at one second intervals. The collected data could then be played back to assess the performance of different localisation systems as required. Using the same recorded data for each environment ensured that all experiments were conducted under identical conditions.

6.4. Location binning mechanism

The problem addressed here was how to find some independent means of recording the "true" location of the robot over time for later comparison with the responses of a particular localisation system. The robot's raw odometry data are unsuitable for location tracking because of the familiar problems of accumulated rotational and translational drift errors due to wheel slippage (see Fig. 4). However, using the robot's online compass-based odometry, we were able to overcome the problem of the rotational drift (as explained in Section 5.2). The remaining translational error was then removed retrospectively as follows.

Firstly, the recorded data were manually divided into laps by finding an obvious landmark (e.g., corner) in the odometer trace. (This could easily be done automatically by using an external sensor to detect the completion of another lap by the robot, e.g., using an overhead cross-bar sensor as in [9].) For each lap, the accumulated drift error was then removed by correcting each data point by an amount proportional to the distance travelled along the route. Finally, an offset was applied to each of the successive laps to cancel out the remaining translational error between them.

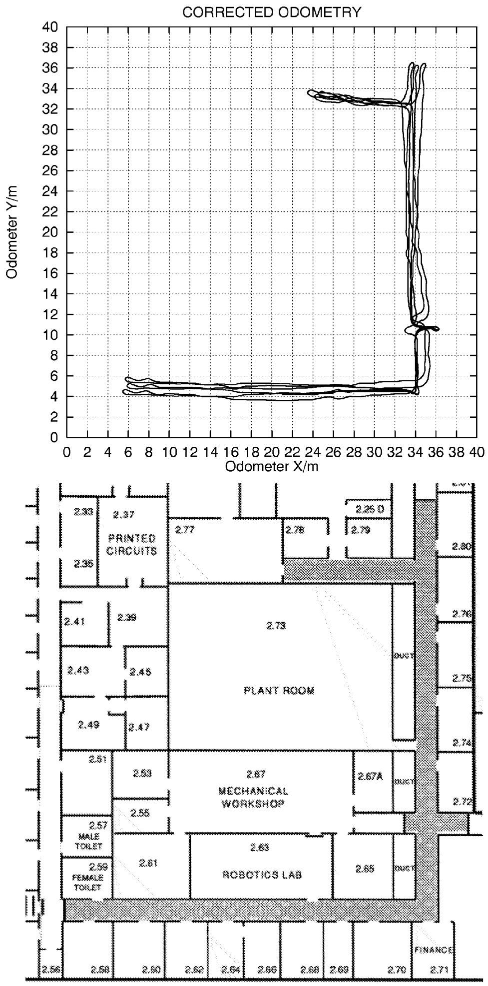

Fig. 5. *Location binning mechanism.* The remaining translational error in the robot's compass-based odometry was corrected off-line (see Section 6.4). The dotted grid was used to coarse-code the location data into bins. The second figure shows a floor plan of the corresponding environment (environment C in Section 7).

Hence, having obtained a method of tracking the robot's true location, the recorded data were coarse-coded into equally sized bins (2 m × 2 m in these experiments) as illustrated in Fig. 5. The orientation and positioning of the dotted grid shown in Fig. 5 over the corrected odometer trace was determined using a search procedure to minimise the number of bins occupied, and then to maximise the value of $H(L)$ where several possible grid positions produced the same number of bins.

It should be noted that global distortions in the corrected odometer trace will not adversely affect the results, provided that the assignment of bins to locations is consistent between successive laps of the environment by the wall-following robot.

6.5. *Measuring performance over distance travelled*

During the experiment, each of the localisation systems under investigation is tested the same number of times ("trials"), using the same played-back perceptions recorded by the robot. Each trial begins from a different location along the route traversed by the robot, and lasts for a fixed distance. In these experiments, we used trials of distance 20 m, starting at 5 s intervals along the recorded route data.

In the subsequent analysis, we assume that all trials begin nominally at distance 0 m and end at distance 20 m. After running a localisation system for a large number of trials, the resulting data were split into 201 separate contingency tables, one for distance 0 m plus one for each 0.10 m travelled by the robot. The performance measures $H(L|R)$ and $U(L|R)$ were then calculated for each of the tables.

7. Experiments conducted

Using the experimental procedure described above, we can quantify the performance of our localisation system over the distance travelled by the robot. However, for the results to be meaningful, we need to be able to compare the performance of our system (qualitatively) against that of other localisation systems. In these experiments, we therefore decided to compare our system with two "base-line" localisation strategies; localisation using dead reckoning (see Section 7.1) and localisation using only currently observable landmarks (see Section 7.2).

The experimental procedure described above was repeated in six different environments around Manchester University's Computer Building, which are summarised in Table 1. In each experiment, the first lap of the recorded robot data were used for map building if required, and the remaining data were used for testing. In each case, the number of trials was carefully chosen so that each part of the environment was equally represented in the data.

Table 1
Characterisation of environments

	Description	Route length (m)	Location bins	No. of trials	Places in EBL map
A	Drinks-machine area	60	24	298	88
B	T-shaped hallway	54	14	263	71
C	L-shaped corridor	146	40	474	185
D	Small unfurnished room	23	8	232	33
E	Single corridor	51	14	248	61
F	E plus moving people	51	14	249	61

Environments A to E remained unchanged throughout the experiments. To assess the impact of changes in an environment on the different systems, we added moving people to environment E to obtain a "dynamic" environment F. We left this environment unchanged during map building. However, during the recording of the data used for testing, 29 persons walked past the robot (inbetween the robot and the wall it was following in 11 of the cases). A further nine persons stood in the corridor or in doorways as the robot went past, thus adding extra landmarks not present in the map. In addition, on four occasions, fire-doors in the immediate vicinity of the robot were left open for several seconds, thus removing landmarks present in the map (although we were careful not to allow the robot to escape into uncharted territory here).

7.1. *Uncorrected dead reckoning*

Here, the robot was allowed to use only its raw odometer readings to localise. At the start of each trial, the robot's uncorrected odometry was initialised to the "correct" position and orientation taken from the corrected odometer trace shown in Fig. 5. The recorded robot data were then played back, using dead reckoning to produce (x, y) coordinates over time. (In our implementation, this meant calculating position and orientation offsets from the initialised values at the start of the trial, and then applying these offsets to the played-back odometer readings.)

To obtain the response R, the (x, y) coordinates produced by the dead reckoning strategy were coarse-coded into bins, using the same dotted grid as shown for the location binning mechanism. To maintain a finite number of possible responses, whenever the position estimate occupied one of the bins not occupied by the corrected odometer trace, the response was classified as being "outside" the environment and was assigned to a separate bin number for the rest of that trial.

7.2. *Nearest neighbour classifier*

Here, the robot was allowed to use only its current sonar and infrared readings to localise, i.e., currently observable landmarks. In this system, the current sensory input is classified according to its nearest neighbour amongst a set of stored prototypes. Each stored prototype consists of a normalised sonar signature and a normalised infrared signature. Classification is decided by normalising the current sonar and infrared readings, and using Eq. (1) (as in the evidence-based localiser) to determine the nearest neighbour.

To facilate direct comparison with the evidence-based localiser, we used exactly the same stored prototypes created by the map building component of that system. During testing, the output class (i.e., the nearest neighbour) was taken as the robot's response R.

8. Results

As we would expect, the results in Fig. 6 show that the performance of dead reckoning worsens over time, while the system using only currently observable landmarks performs at a roughly constant level and the performance of the evidence-based system improves over time. The evidence-based localiser took longer to localise in the dynamic environment, but still eventually achieved the same overall level of performance. The performance of both the landmark-based localiser and dead reckoning worsened in the dynamic environment (avoiding the people meant more turns, and

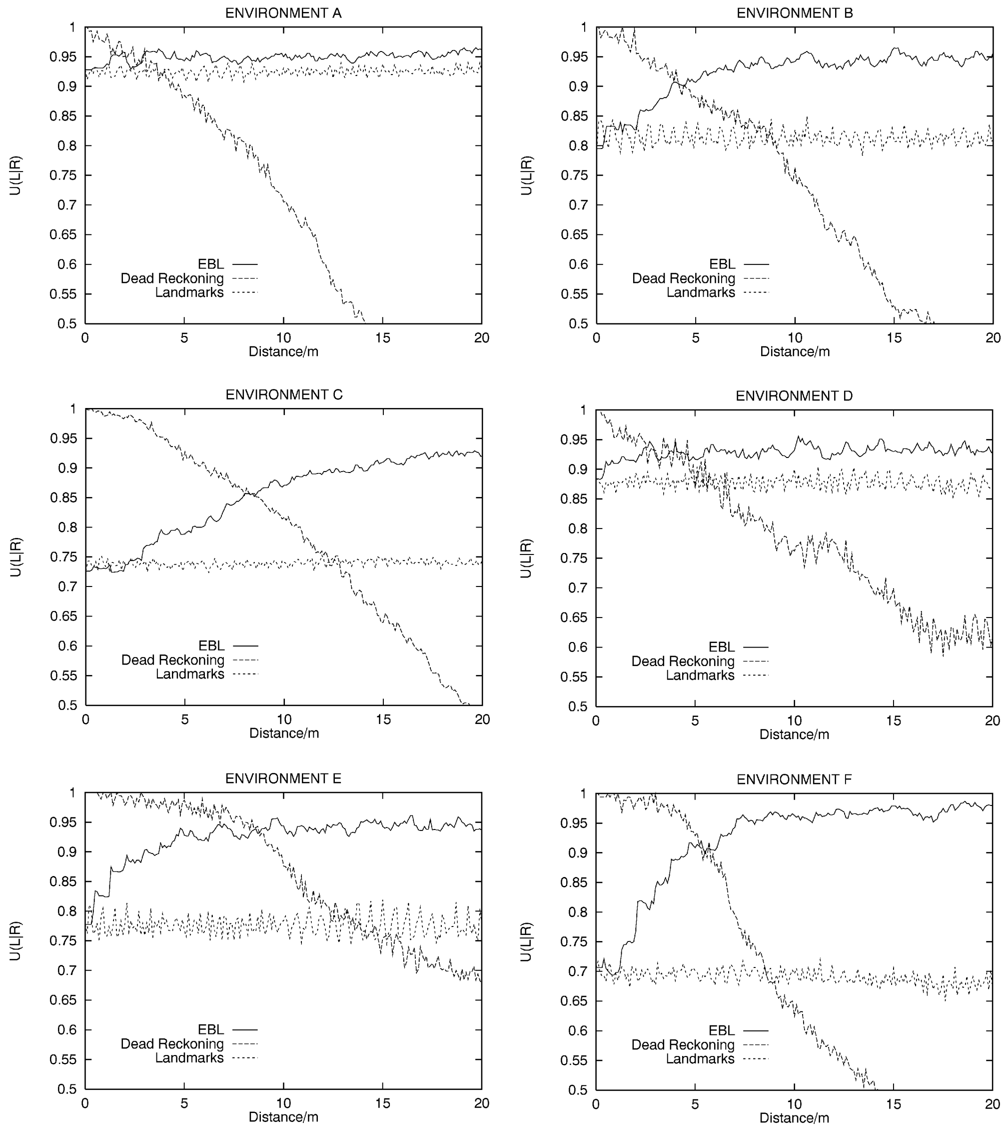

Fig. 6. Results in the six different environments.

thus more accumulated rotational drift in the robot's odometry).

The measures obtained in the different experiments are also summarised in Table 2. Here, the mean values of $U(L|R)$ for the landmark-based system and dead reckoning reflect, respectively, the overall levels of perceptual aliasing and odometer drift which occurred in the different environments. The mean values of $H(L|R)$ also take into account the size of a particular environment (as discussed previously in Section 4.1).

9. Summary

In this paper, we have presented a novel localisation system which accumulates exteroceptive and proprioceptive sensory information over time to localise. This

Table 2
Summary of results

	EBL distance (m) for $U(L\|R) = 0.9$	Landmark classifier mean $U(L\|R)$	Landmark classifier mean $H(L\|R)$	Dead reckoning mean $U(L\|R)$	Dead reckoning mean $H(L\|R)$
A	0.0	0.925	0.227	0.663	1.013
B	3.8	0.814	0.487	0.724	0.719
C	13.6	0.739	0.980	0.785	0.807
D	0.5	0.878	0.236	0.786	0.413
E	3.0	0.776	0.585	0.864	0.355
F	4.7	0.690	0.808	0.686	0.819

system can localise even when lost, is computationally efficient in middle-scale environments and robust in dynamic environments, showing a graceful degradation in performance in the presence of moving people.

We have also presented a general performance measure and an experimental procedure which enable different localisation systems to be compared on the same robot in the same environment. The performance metric measures the information content of the responses of the particular system under investigation in predicting the observed location of the robot. A novel mechanism, based on a compass sense and retrospectively corrected odometry, was used for tracking the location of the robot.

Our experiments were conducted using logged sensor data recorded by a Nomad 200 following a circular path by wall-following. Using recorded data, obtained from a real robot, ensured that all comparisons were conducted under identical conditions, whilst retaining the complexity of robot–environment interaction. Using the methods presented in this paper, we were thus able to compare the performance of our own novel localisation system against that of localisation using dead reckoning or currently observable landmarks alone.

Evidence-based localisation does not require a global Cartesian reference system, a highly desirable property given the fundamental unreliability of navigation by dead reckoning (see Fig. 4). The localisation system presented here, besides using exteroceptive information, relies on *local* odometry only, and our experiments show that reliable localisation in real world environments of middle-scale dimensions is possible using this method.

10. Conclusions and future work

The work presented here is solely concerned with robot self-localisation in static environments, along one specific route through the environment.

In future work, we intend to address the issue of fully autonomous map building using "lifelong learning" (as in [12]), where both map building and localisation would be run continuously and in parallel. Relocalisation would be used to maintain global consistency in the map, recognising places seen before by the robot. To achieve this, we will also need to investigate exploration strategies which take into account the robot's map rather than the purely reactive strategy used here.

Yamauchi [11] distinguished transient from lasting changes in an environment. So far, we have only considered the former of the two. Lifelong learning would also address the problem of lasting changes, allowing the robot to "forget" as well as to aquire map knowledge.

In conclusion, we believe that mobile robot navigation systems will only be generally useful when they have been tested successfully in the "real world", i.e., when the problems of scalability which occur in middle-scale environments have been addressed by experiments in precisely these middle-scale environments. Put more simply, robots tested in specially contructed environments are in danger of solving only specially constructed problems! Towards this end, we believe that using quantitative methods more, while using mere existence proofs less, will move mobile robotics further towards an exact science.

References

[1] N. Burgess, J.G. Donnett, J. O'Keefe, Robotic and neuronal simulation of hippocampal navigation, Proceedings AISB-97 Workshop on Spatial Reasoning in Mobile Robots and Animals, University of Manchester, Technical Report Series UMCS-97-4-1, ISSN 1361–6161, 1996.

[2] A. Kurz, Constructing maps for mobile robot navigation based on ultrasonic range data, IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 26 (2) 1996.

[3] D.C. Lee, The map-building and exploration strategies of a simple, sonar- equipped mobile robot; An experimental, quantitative evaluation, Ph.D. thesis, University College, London, UK, 1995.

[4] U. Nehmzow, Autonomous acquisition of sensor-motor couplings in robots, Department of Computer Science, University of Manchester, Technical Report Series UMCS-94-11-1, 1994.

[5] U. Nehmzow, T. Smithers, J. Hallam, Location recognition in a mobile robot using self-organising feature maps, in: G. Schmidt, (Ed.), Information Processing in Autonomous Mobile Robots Springer, Berlin, 1991.

[6] S. Oore, G.E. Hinton, G. Dudek, A mobile robot that learns its place, Neural Computation 9 (1997) 683–699.

[7] M. Piasecki, Global localization for mobile robots by multiple hypothesis tracking, Robotics and Autonomous Systems 16 (1995) 93–104.

[8] C.E. Shannon, W. Weaver, The mathematical theory of communication, University of Illinois Press, Champaign, IL, 1949.

[9] T. Smithers, On quantitative performance measures of robot behaviour, Robotics and Autonomous Systems 15 (1–2) (1995) 107–133.

[10] R. Wehner, M. Lehrer, W.R. Harvey (Eds.), Animal Navigation (Special issue), The Journal of Experimental Biology 199 (1996).

[11] B. Yamauchi, P. Langley, Place recognition in dynamic environments, in: Mobile Robots (Special issue), Journal of Robotics Systems 14 (2) (1997) 107–120.

[12] U.R. Zimmer, Self-localization in dynamic environments, Proceedings IEEE/SOFT International Workshop BIES'95, Tokyo, Japan, 1995.

Tom Duckett studied his B.Sc. in Computer and Management Sciences at Warwick University. After a three-year spell in industry, he returned to academic life and was awarded an M.Sc. in Knowledge Based Systems with distinction by Heriot–Watt University. His M.Sc. project was carried out in Germany at Karlsruhe University, and involved working on a project for supporting architectural design. Tom is currently pursuing his Ph.D. at Manchester University in the field of mobile robot navigation.

Ulrich Nehmzow is a lecturer in Artificial Intelligence and Robotics at the Department of Computer Science of the University of Manchester, leading the Robotics Research Group. He graduated in 1988 from the Technical University of Munich with a Diploma in Electrical Engineering and Information Science, and with a Ph.D. in Artificial Intelligence from Edinburgh University in 1992. Following appointments as research associate at the Department of Artificial Intelligence, Edinburgh University, and postdoctoral research associate at the Department of Psychology, Edinburgh University, he became lecturer at Manchester in 1994. He is a member of the Institution of Electrical Engineers, member of the IEE Northwest Center committee, member of the Society for Artificial Intelligence and Simulation of Behaviour, and has worked as Visiting Research Scientist at Carnegie Mellon University, Bremen University, and, most recently, as a Royal Society/STA Fellow at the Electrotechnical Laboratory in Tsukuba (Japan). His research interests are autonomous mobile robotics, navigation, robot learning, and artificial neural networks.