Autonomous Robots 12, 287–300, 2002

Fast, On-Line Learning of Globally Consistent Maps

TOM DUCKETT
Department of Technology, University of Örebro S-70182 Örebro, Sweden
tom.duckett@tech.oru.se

STEPHEN MARSLAND AND JONATHAN SHAPIRO
Department of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK
smarsland@cs.man.ac.uk
jls@cs.man.ac.uk

Abstract. To navigate in unknown environments, mobile robots require the ability to build their own maps. A major problem for robot map building is that odometry-based dead reckoning cannot be used to assign accurate global position information to a map because of cumulative drift errors. This paper introduces a fast, on-line algorithm for learning geometrically consistent maps using only local metric information. The algorithm works by using a relaxation technique to minimize an energy function over many small steps. The approach differs from previous work in that it is computationally cheap, easy to implement and is proven to converge to a globally optimal solution. Experiments are presented in which large, complex environments were successfully mapped by a real robot.

Keywords: simultaneous localization and mapping, concurrent map-building and self-localization, relaxation algorithm, Gibbs sampling, learning and adaptation

1. Introduction

Maps are very useful for mobile robot navigation in complex environments, being needed for self-localization and path planning, as well as enabling human operators to see where the robot has been. While successful navigating robots have been developed using pre-installed maps, to operate in *unknown* environments a robot needs the ability to build its own maps. However, the sensor information available to the robot is noisy and can produce errors when integrated into the map. In particular, the robot's odometry is subject to drift errors caused by factors such as wheel slippage, which can lead to an inconsistent mapping of the environment. To maintain a coherent representation of the environment that can be reconciled with future sensory perceptions, some means of maintaining geometric consistency in the map is required.

This paper introduces a fast, on-line algorithm for obtaining globally consistent maps. The approach differs from previous work in that it is computationally cheap, easy to implement and is guaranteed to find a solution that is statistically optimal. Our algorithm assumes three sources of perceptual information:

(i) a place recognition system,
(ii) a global orientation obtained from a compass, and
(iii) local distance information from odometry.

Experiments are presented in which large, complex environments were successfully mapped by a real robot using ultrasonic range-finder sensors for (i), a flux-gate compass for (ii), and uncorrected odometer sensors for (iii).

The map representation consists of a topologically connected network of places, where each link is labeled with noisy metric information describing the relative distance and absolute angle between the two places it connects. The purpose of our algorithm is to assign a globally consistent set of Cartesian coordinates to the places in the map. In this approach, the coordinates of the places are treated as free variables, and the algorithm finds an optimal set of coordinates using only the local metric relations between places.

The rest of this paper is structured as follows. Section 2 provides a detailed description of the problem. Section 3 derives a map learning algorithm from a statistical model of the noise on the robot's sensors and proves that the algorithm converges to a globally optimal solution, together with a complexity analysis. Section 4 describes the robotic platform used for the experiments, including implementation details of the place recognition system and incremental mapping strategy applied. This is followed by experimental results, related work and conclusions. Finally, two variants of the map learning algorithm are discussed: a simplification of the basic algorithm for use when only approximate geometric information is required (Appendix A), and a generalization for use when the robot is not equipped with a compass (Appendix B).

2. The Problem

When maps are generated from the estimates of distances and angles measured by the robot, the geometry of the space will be non-Euclidean. For example, the angles inside a triangle may not add up to 180°. As the robot is exploring in Euclidean space, this is a problem. Our map building algorithm aims to find an evidence-based way of fixing this problem so that the maps are geometrically consistent.

In our experiments, we use a graph-based model of the environment, in which the nodes correspond to places and the links to traversable paths between places (see Fig. 1). Each node in the graph is associated with a local occupancy grid, constructed using the robot's sonar sensors, which is used as a place signature. For place recognition, we use the self-localization algorithm described in Duckett and Nehmzow (2001), which applies an occupancy grid-matching technique to identify both the robot's current place in the map and the relative displacement of the robot from the center of that particular place (see Section 4 for more details).

For the following analysis, we define our maps as follows:

- The topological component of the map consists of a set of N place nodes and a set of links that connect pairs of places.

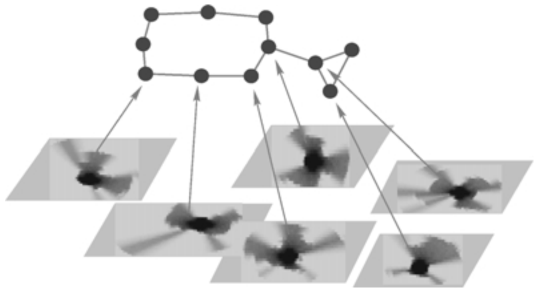

Figure 1. The map representation consists of a topologically connected set of places, each place being associated with a local occupancy grid. The core problem addressed by this paper is to assign geometrically consistent coordinates to the places.

- Each place i is associated with a pair of Cartesian coordinates $\mathbf{r}_i = (x_i, y_i)^T$ that are initially unassigned (or they could be initialized approximately by dead reckoning). The true coordinates are unknown to the robot.
- Each link connects two places i and j, and is associated with an estimated metric relation $\mathbf{l}_{ij} = (d_{ij}, \theta_{ij})$, measured by the robot, that describes the relative distance d_{ij} and angle θ_{ij} between the two places. The angle θ_{ij} is an absolute measurement obtained from the compass. In this paper, the links were constrained to be bi-directional, that is, $d_{ij} = d_{ji}$ and $\theta_{ij} = \theta_{ji} + \pi$.

The algorithm derived in the next section aims to assign a globally consistent set of Cartesian coordinates $\{\mathbf{r}_1, \ldots, \mathbf{r}_N\}$ to the places in the map using the noisy measurements of the distances and angles between the places. This noise means that a perfect map cannot be generated. Instead, we derive a statistically optimal algorithm to deal with this problem in a principled way.

3. The Map Learning Algorithm

3.1. Estimating the Noise

When the robot travels between two map places, a large number of small distances and angles are measured as the robot continually updates its heading (see Fig. 2). In our experiments, the distances are measured with odometry and the angles are measured with a flux-gate compass. Let δ_t be the displacement and α_t the heading

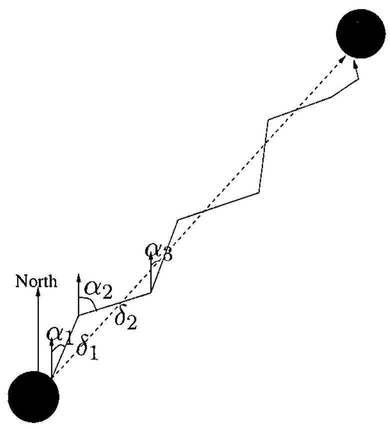

Figure 2. Example of how the robot travels between two map nodes, constantly updating its heading. In reality the number of update steps is higher.

at the tth step. Then

$$d_{ji} = \sqrt{\left(\sum_t^{T_{ji}} \delta_t \cos\alpha_t\right)^2 + \left(\sum_t^{T_{ji}} \delta_t \sin\alpha_t\right)^2}, \quad (1)$$

$$\theta_{ji} = \tan^{-1}\left(\frac{\sum_t^{T_{ji}} \delta_t \sin\alpha_t}{\sum_t^{T_{ji}} \delta_t \cos\alpha_t}\right), \quad (2)$$

where T_{ji} is the number of steps, i.e., the number of updates the robot performs along a path.

Assuming that the number of updates made while traveling between two map places is large and that the statistical properties of each step are independent, the Central Limit Theorem (Reif, 1982) implies that measurements of d_{ji} and θ_{ji} will be normally distributed around their true values.

Suppose that the noise properties are the same along a path between two map places (but not necessarily between paths) and that successive measurements are independent. That is, the measurement of the path from place i to place j is independent of the measurement from place j to place k (this means that the model cannot deal with cumulative phenomena such as battery drain). Then we can write the estimates of the distance travelled and the heading of the robot at each small step as

$$\delta_t = \delta_t^* + \Delta\delta_t \quad (3)$$

$$\alpha_t = \alpha_t^* + \Delta\alpha_t \quad (4)$$

where $\Delta\delta_t$ is the noise in the estimate of the true distance δ_t^* and $\Delta\alpha_t$ is the noise in the estimate of the true angle α_t^*.

Assuming that the noise measurements are small compared to the distances traveled, the covariance matrix C_{ji} of the link measurements (d_{ji}, θ_{ji}) can be calculated using small deviation expansions as

$$C_{ji} = T_{ji} R(\phi_{ji})^{-1} \begin{pmatrix} \Delta_{\delta\delta} & \delta_{ji}\Delta_{\alpha\delta} \\ \delta_{ji}\Delta_{\alpha\delta} & \delta_{ji}^2\Delta_{\alpha\alpha} \end{pmatrix} R(\phi_{ji}), \quad (5)$$

where $\delta_{ji} = (\sum_t^{T_{ji}} \delta_t)/T_{ji}$, that is the average distance between updates, the $\Delta_{\delta\delta}$, etc. are global noise estimates of the subscripted variables, and ϕ_{ji} is the rotational error between the two nodes, where $R(\phi)$ is the rotation matrix

$$R(\phi) \equiv \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}. \quad (6)$$

In our model, it is assumed that $\phi_{ji} = 0$ because of the compass, so the rotation matrix $R(\phi_{ji}) = I$. This model is invariant to global rotations and depends only on the direction of travel. The noise parameters $\Delta_{\delta\delta}$, etc. are calculated by maximum likelihood estimation as part of the relaxation algorithm (Eqs. (12)–(14)).

3.2. *The Algorithm*

The map can be considered as a set of free nodes that are held together by springs, where each spring connects two adjacent places i and j (this analogy can also be found in (Lu and Milios, 1997a; Golfarelli et al., 1998; and Shatkay, 1998)). Each spring reaches minimum energy when the relative displacement between the coordinates of node i and node j is equal to the vector (d_{ij}, θ_{ij}) measured by the robot. Equilibrium is reached in the whole map when the total energy over all of the springs reaches a global minimum. Thus, global consistency is maintained in the map by minimizing the following energy function, which corresponds to the log likelihood:

$$E = \sum_i \sum_j{}' (\mathbf{r}_i - \mathbf{r}_j - \mathbf{D}_{ji})^T C_{ji}^{-1} (\mathbf{r}_i - \mathbf{r}_j - \mathbf{D}_{ji}), \quad (7)$$

where $\sum_j'$ refers to the sum over the neighbors of node i, $\mathbf{r}_i = (x_i, y_i)^T$, and $\mathbf{D}_{ji} = d_{ji}\binom{\cos\theta_{ji}}{\sin\theta_{ji}}$.

Since each C_{ji} is symmetric,

$$\nabla E_i = 2 \sum_j{}' C_{ji}^{-1}(\mathbf{r}_i - \mathbf{r}_j - \mathbf{D}_{ji}), \tag{8}$$

when the position of node i is updated. Therefore, the algorithm that finds the maximum likelihood solution is the one that finds the $\mathbf{r}_i'$ that minimizes E, that is,

$$\mathbf{r}_i' = \left(\sum_j{}' C_{ji}^{-1} \right)^{-1} \sum_j{}' C_{ji}^{-1}(\mathbf{r}_j + \mathbf{D}_{ji}). \tag{9}$$

This can be considered as a form of Gibbs sampling at zero temperature (Reif, 1982). One component is optimized by keeping all of the others fixed.

The algorithm derived from Eq. (9) is given in Fig. 3. The basic principle of the algorithm is to "move each node to where its neighbors think it should be". By iteration, the coordinates in the map converge towards a global minimum in the energy function (Eq. (7)). For on-line map learning, steps 2 and 3 in Fig. 3 are inter-leaved with the rest of the navigation control software so that the map is adapted continually during exploration.

An example illustrating the convergence of the relaxation algorithm is given in Figs. 4 and 5. In this experiment, the coordinates of the self-acquired map in Fig. 4 were randomly reinitialized to arbitrary values, then the algorithm was iterated until the map returned to its globally consistent solution. (Note that this differs from the normal, on-line use of the algorithm, where the coordinates are *not* reinitialized; rather, the existing coordinates are adapted continually over time.)

3.3. Proof of Convergence

To prove convergence of the algorithm, we show here that the algorithm always minimizes the energy function (Eq. (7)). When a node i with position $\mathbf{r}_i$ is updated, its new position $\mathbf{r}_i'$ will be given by Eq. (9). That is, the new position is picked to minimize the value of $\sum_j (\mathbf{r}_i' - \mathbf{r}_j - \mathbf{D}_{ji})$. Hence, the change in energy will be

$$\begin{aligned}\Delta E = \sum_j{}' \big\{ &(\mathbf{r}_i' - \mathbf{r}_j - \mathbf{D}_{ji})^T C_{ji}^{-1} (\mathbf{r}_i' - \mathbf{r}_j - \mathbf{D}_{ji}) \\ &- (\mathbf{r}_i - \mathbf{r}_j - \mathbf{D}_{ji})^T C_{ji}^{-1} (\mathbf{r}_i - \mathbf{r}_j - \mathbf{D}_{ji}) \big\} \quad (15) \\ \leq 0. \quad (16)\end{aligned}$$

Since E is bounded below, the algorithm must converge and any legitimate stopping criterion must be reached. The energy function is quadratic and therefore has a unique minimum, so the algorithm can only converge to a global minimum. The fact that the energy function is quadratic also means that the updates could be computed in a single step by inverting the $(N \times N)$ matrix showing connections between all of the nodes in the map. Our system performs an iterative update instead, which is computationally simple and fast, and therefore more suitable for robotics applications.

3.4. Complexity Analysis

The computational cost of the new algorithm is linear in the number of places in the map. Because the algorithm makes an iterative refinement to the existing solution, rather than recalculating the entire coordinate system from scratch, only one iteration is typically required whenever new information is added to the map. The complexity of the algorithm is thus bounded by $O(NM)$, where M is the maximum number of neighbors per node. For a topological map, the number of links per node will not grow with the size of the map, so M is constant and the overall complexity is approximately $O(N)$. This compares favorably with the worst case $O(N^3)$ complexity of matrix inversion methods such as Lu and Milios (1997a) and Golfarelli et al. (1998) (though that can be reduced to $O(N^2)$ by using a sparse matrix solver if the covariance matrix is not required (Frese and Hirzinger, 2001)).

4. Robotic Implementation

The relaxation algorithm was tested on a Nomad 200 robot as part of a complete system for mapping unknown environments (Duckett, 2000). In this system, the robot attempts to space the place nodes in the map at equal intervals of 1 meter. Some important details of the robotic implementation are given as follows.

4.1. Compass Sense

To pre-process the readings from the robot's flux-gate compass, we used the behavior-based filtering method described in Duckett and Nehmzow (2001). In this approach, a separate behavior is used to rotate the robot's turret at small speeds in the direction of an arbitrary 'North', as indicated by the compass. Then the angular

1. Initialize the covariance matrix C_{ji} for all links. This could be taken to be the identity matrix, or the covariance matrix computed during a previous run of the algorithm (see step 3).
2. For each node i, do:

   a) For each of the neighbors j of node i, i.e., the places that are topologically connected to i, obtain an estimate $\mathbf{r}'_{ji}$ of the coordinates of node i using

   $$\mathbf{r}'_{ji} = \mathbf{r}_j + \mathbf{D}_{ji}, \tag{10}$$

   where $\mathbf{r}_j = (x_j, y_j)^T$ refers to the coordinates of node j, and $\mathbf{D}_{ji} = d_{ji}\begin{pmatrix}\cos\theta_{ji}\\ \sin\theta_{ji}\end{pmatrix}$.

   b) Combine the position estimates $\mathbf{r}'_{ji}$ for all j to produce new coordinates $\mathbf{r}'_i$ for node i using

   $$\mathbf{r}'_i = \left(\sum_j{}' C_{ji}^{-1}\right)^{-1} \sum_j{}' \left(C_{ji}^{-1}\mathbf{r}'_{ji}\right), \tag{11}$$

   where $\sum'_j$ refers to the sum over the neighbors of node i.

3. Re-estimate the noise parameters in the C_{ji} (equation 5) by maximum likelihood estimation using

   $$\Delta_{\delta\delta} = \sum_i \sum_j{}' \frac{1}{NN_iT_{ji}} \left[(R(\phi_{ji})(\mathbf{l}'_{ji} - \mathbf{l}_{ji}))^T\right]_1 \left[R(\phi_{ji})(\mathbf{l}'_{ji} - \mathbf{l}_{ji})\right]_1 \tag{12}$$

   $$\Delta_{\alpha\delta} = \sum_i \sum_j{}' \frac{1}{NN_iT_{ji}} \left[(R(\phi_{ji})(\mathbf{l}'_{ji} - \mathbf{l}_{ji}))^T\right]_1 \left[R(\phi_{ji})(\mathbf{l}'_{ji} - \mathbf{l}_{ji})\right]_2 \tag{13}$$

   $$\Delta_{\alpha\alpha} = \sum_i \sum_j{}' \frac{1}{NN_iT_{ji}} \left[(R(\phi_{ji})(\mathbf{l}'_{ji} - \mathbf{l}_{ji}))^T\right]_2 \left[R(\phi_{ji})(\mathbf{l}'_{ji} - \mathbf{l}_{ji})\right]_2 \tag{14}$$

   where $\mathbf{l}_{ij} = (d_{ij}, \theta_{ij})^T$ refers to the metric relation measured by the robot for the link from node i to node j, and $\mathbf{l}'_{ij} = (d'_{ij}, \theta'_{ij})^T$ is the corresponding metric relation generated by relaxation. N is the number of nodes in the network, N_i is the number of neighbours of node i, and $[\cdot]_i$ denotes the ith component of a vector.

4. Repeat from step 2 using the new values of the noise parameters until the change in energy falls below some pre-defined threshold, or some other stopping criterion is reached.

Figure 3. The maximum likelihood map learning algorithm.

estimates are obtained by measuring the relative displacement of the turret against the direction of travel.

The effect of this behavior is to smooth out any fluctuations in the compass readings caused by electromagnetic disturbances, maintaining a constant orientation around the average value of 'North'. We have found that the performance of this method degrades gracefully with respect to the magnetic variations in the environment. It also has the advantage of keeping the robot's range-finder sensors at a steady orientation,

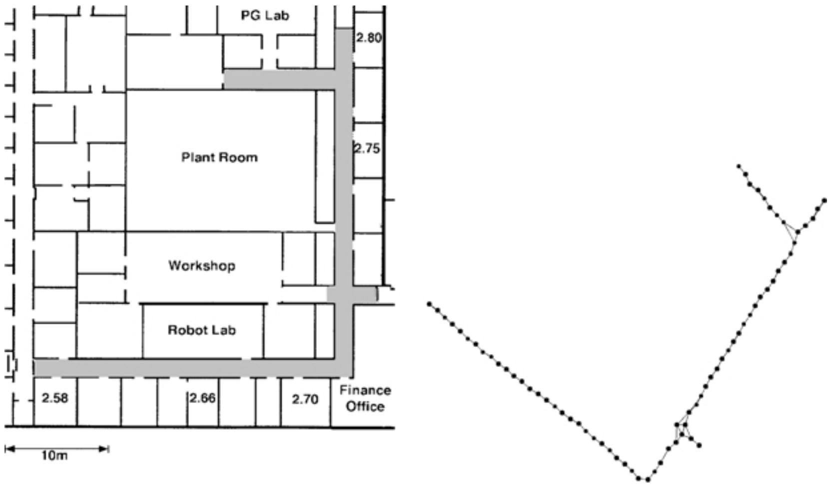

Figure 4. Left: floor plan of a corridor environment at Manchester University (size 34 m × 33 m). Right: the corresponding map acquired by the robot.

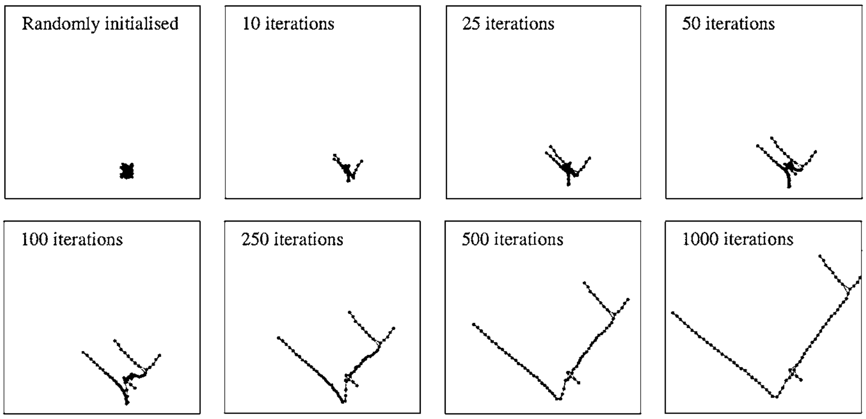

Figure 5. Convergence of the relaxation algorithm. In the first picture, the coordinates of the self-acquired map shown in Fig. 4 have been randomly reinitialized. The remaining pictures show the map after 10, 25, 50, 100, 250, 500 and 1000 iterations respectively of the relaxation algorithm.

which can help to reduce the noise on the ultrasonic sensor readings. This compass sense was also used for the on-line dead reckoning, as illustrated in Fig. 6.

4.2. *Exploration Strategy*

Topological map building was performed using an incremental exploration strategy, in which the robot continually tries to expand the territory that has already been charted (see Fig. 7). To implement this strategy, two different types of place are included in the map:

– *Predicted.* Places presumed to exist but not yet visited by the robot.
– *Confirmed.* Places actually visited by the robot.

The basic exploration strategy consists of trying to move towards the nearest predicted place, using a standard graph-based path planning algorithm to determine the route. An artificial neural network is used to predict new places by classifying the robot's sonar readings in all directions (see Duckett and Nehmzow (1999) for full details). A local dead reckoning strategy is used to decide whether to confirm the predicted places: a predicted place is replaced by a confirmed place if a

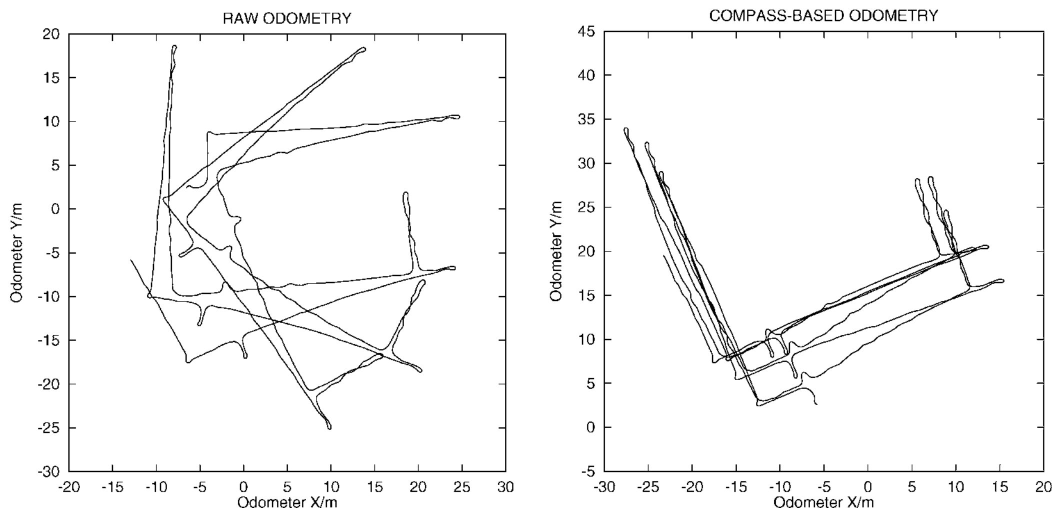

Figure 6. Left: raw odometry. Right: compass-based odometry. The accumulated rotational drift in the robot's raw odometry was removed on-line using the compass sense. In this example, the robot repeatedly traversed the environment of Fig. 4 by wall-following.

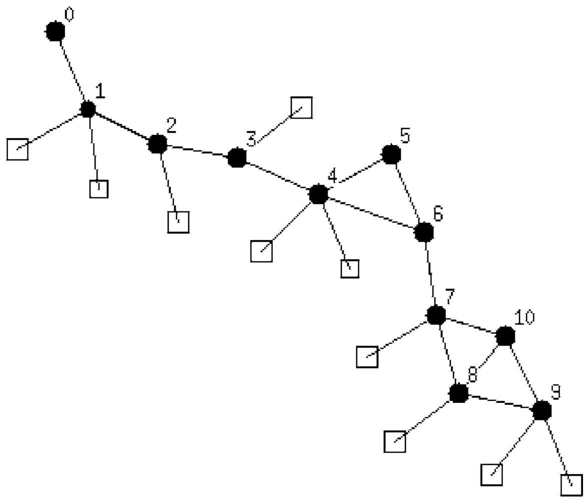

Figure 7. Example of incremental map building. Places predicted but not yet visited by the robot are shown by squares. Places visited by the robot are shown by filled circles (the numbers indicate the sequence in which they were visited).

distance of 1 meter from the nearest existing confirmed place can be traveled, otherwise the predicted place is deleted from the map. The coordinates produced by dead reckoning (relative to the last visited place) are also used to initialize the coordinates of the confirmed places before the relaxation algorithm is applied. Whenever another confirmed place is added to the map, the neural network is used again to predict more places. In addition, connections are inferred to any other confirmed places lying less than 2 meters from the added node, provided that the neural network indicates open space in that particular direction. The whole process is repeated until all predicted places in the map have either been visited by the robot or deleted.

In the experiments presented here, an extra heuristic was added to this strategy to force the robot to close loops, described as follows. Whenever a new confirmed place is added to the map, a search is carried out on each of the adjacent confirmed place nodes to determine the shortest path actually traversed by the robot from that particular node. If the length of that path exceeds a pre-specified threshold of 3 meters, then the robot is forced to travel directly to that node in order to obtain a measurement of the metric relation for the connecting link. Only the physically traversed links are used by the relaxation algorithm in the calculation of the node coordinates (the inferred links are used only for path planning). In the self-acquired map of Fig. 12, the links actually traversed by the robot are shown in bold, while the inferred links are shown by dotted lines.

The links in the topological map are maintained using the following rules taken from Yamauchi and Beer (1996). Whenever a new link is added to the map, a confidence level, c_{ij}, for that link is initialized to a value of 0.5. During repeat traversals of the same link, the confidence level is increased using

$$c'_{ij} = \lambda + (1 - \lambda)c_{ij}, \tag{17}$$

where the link adaptation rate, $\lambda = 0.5$ in these experiments. Conversely, whenever the robot fails to traverse a given link, e.g., because the robot reaches a different destination to the one intended by the path planner, the confidence value is decreased using

$$c'_{ij} = (1 - \lambda)c_{ij}. \tag{18}$$

A link is deleted from the map whenever its confidence level falls below a pre-specified threshold (0.2 in these experiments). A node is deleted from the map if no path can be found to that node from the robot's current location, i.e., when no possible routes exist due to link deletion.

4.3. Use of Local Metric Information

In the derivation of the relaxation algorithm in Section 3, it was assumed that the robot takes measurements of the relative distances and angles between a discrete set of places, which correspond to the nodes of a graph. In reality, the robot moves in a continuous space, and the places are contiguous regions rather than single points in space. So to implement the algorithm, some means of measuring the position of the robot relative to the centre of the current place is required. In our approach, as in others (Weiss and von Puttkamer, 1995; Lu and Milios, 1997a; Gutmann and Konolige, 1999), this is achieved by scan matching. Specifically, we use the local occupancy maps embedded in the hybrid metric-topological representation of Fig. 1.

In our approach, we do not store or match the actual occupancy grids. Instead, we first reduce each grid to a pair of histograms, one in x-direction and one in y-direction, as shown in Fig. 8. This is done by adding up the total number of occupied, empty and unknown cells in each row or column of the grid (see Duckett and Nehmzow (2001) for a full description). A pair of occupancy histograms is stored for each place in the robot's map, and scan matching then consists of convolving a new pair of histograms constructed from the robot's immediate sonar readings with the stored histograms, as in Fig. 9. In the absence of a compass, we would also have to consider angle histograms, as in Hinkel and Knieriemen (1988). For each stored place i, the matching procedure yields two useful quantities:

1. A metric that quantifies the strength of the match between the observation histograms and the stored histograms for place i.

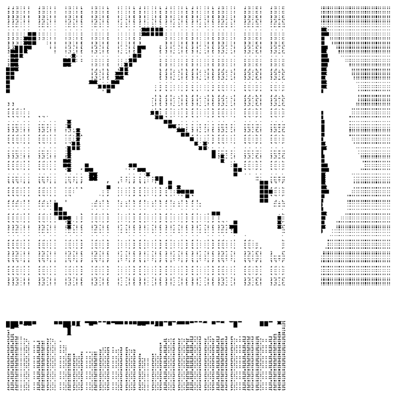

Figure 8. Example occupancy grid and histograms. Occupied cells are shown in black, empty cells in white and unknown cells in gray.

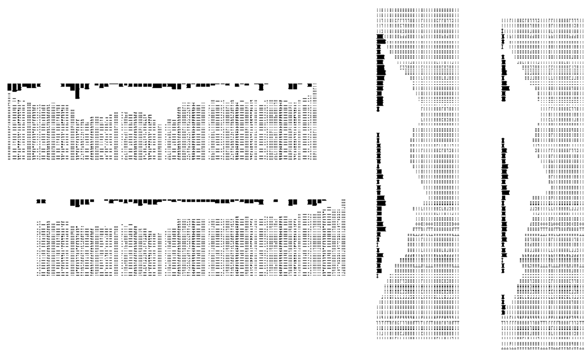

Figure 9. Matching the x and y histograms. The observation histograms are convolved with the stored histograms for each place in the robot's map to find the best match.

2. The most likely offset of the robot from the center of the place, i.e., the position in which the sonar scan for that place was taken.

The strength of the match between two histograms T^a and T^b is calculated using the following evaluation function:

$$\begin{aligned} Match(T^a, T^b) = \frac{1}{w} \sum_j \big[&\min\big(O_j^a, O_j^b\big) \\ &+ \min\big(E_j^a, E_j^b\big) + \min\big(U_j^a, U_j^b\big)\big], \end{aligned} \tag{19}$$

where O_j, E_j and U_j refer to the number of occupied, empty and unknown cells contained in the jth element of histogram T, and w is a normalizing constant such that $0 \le Match() \le 1$. The match scores are used for place recognition: the most likely location of the robot in the map is determined by a Bayesian multi-hypothesis tracking algorithm (Duckett and Nehmzow, 2001), which takes into account the prior probability distribution over the places in the map, the movement of the robot between observations and the new observation match scores.

The most likely offset (r_x^*, r_y^*) of the robot from the centre of the current place is then obtained by multiplying the best-matching translations for the x and y histograms by the dimensions of one grid cell (15 cm × 15 cm). To obtain an estimate of the error in the scan matching, the following heuristic functions are used:

$$\sigma_x^2 = \frac{k}{(M_x^* - \bar{M}_x)^2}, \tag{20}$$

$$\sigma_y^2 = \frac{k}{(M_y^* - \bar{M}_y)^2}, \tag{21}$$

where σ_x^2 refers to the error estimate for the x-histograms, M_x^* refers to the value of *Match*( ) produced by the best-matching alignment of x histograms for the current place, $\bar{M}_x$ refers to the mean value of *Match*( ) in the convolution of x histograms, and the constant $k = 1.0\ \text{m}^2$ in these experiments.

To minimize the overall error due to scan matching, all of the estimates of the offset from the place center that are made by the robot as it travels through the current place are combined together using an iterative filtering technique, which is specified by the following equations:

$$r_x = r'_x + \frac{\sigma_{x'}^2}{\sigma_{x'}^2 + \sigma_{x^*}^2}(r_x^* - r'_x), \tag{22}$$

$$\frac{1}{\sigma_x^2} = \frac{1}{\sigma_{x'}^2} + \frac{1}{\sigma_{x^*}^2}, \tag{23}$$

where r_x is the new x-offset relative to the place centre, r_x^* is the x-offset from the latest observation, and r'_x is the previous x-offset updated by dead reckoning to take into account the movement of the robot between observations. The quantities σ_x^2, $\sigma_{x^*}^2$, and $\sigma_{x'}^2$ are the corresponding error measures. In the experiments presented here, the occupancy histograms were extracted from sonar scans taken at intervals of 50 cm by the traveling robot.

The estimated metric relation, $\mathbf{l}_{ij} = (d_{ij}, \theta_{ij})$, for the link connecting two places i and j is then obtained as

$$d_{ij} = \sqrt{(\Delta x_{ij})^2 + (\Delta y_{ij})^2}, \tag{24}$$

$$\theta_{ij} = \tan^{-1}\left(\frac{\Delta y_{ij}}{\Delta x_{ij}}\right), \tag{25}$$

where

$$\Delta x_{ij} = r_{x_i} + \Delta x - r_{x_j}, \tag{26}$$

$$\Delta y_{ij} = r_{y_i} + \Delta y - r_{y_j}, \tag{27}$$

with the vector $(\Delta x, \Delta y)$ referring to the link measurement obtained by dead reckoning, and the (r_{x_i}, r_{y_i}) referring to the filtered offsets for place i from Eq. (22). In the current implementation, the link measurement is obtained from the first traversal of the link only, and the metric information from any subsequent traversals is discarded.

5. Experimental Results

The system was tested in the office environment shown in Figs.10 and 11. The map acquired by the robot in Fig. 12 shows the position of the places in global coordinates calculated by the new algorithm. To illustrate the accuracy of the acquired map, we have also combined the relaxed coordinates with the recorded sonar data to produce a global occupancy grid model of the environment, using the standard technique developed by Moravec and Elfes (1985). The derived gridmap is shown in Fig. 13. The map has a resolution of 0.15 meters, and should be accurate enough for safe navigation and planning. This can be compared to the gridmap constructed from the robot's compass-based odometry in Fig. 14.

The entire process requires minimal computational resources. Maximum likelihood estimation with the new algorithm was performed on-line as part of the

Figure 10. The Nomad 200 mobile robot *Milou* in the test environment at Örebro University (corresponding to the right-hand side of the map in Fig. 11).

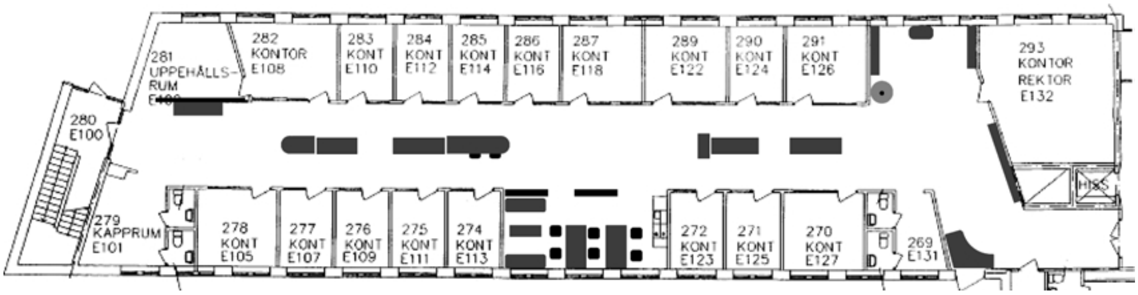

Figure 11. A semi-structured office environment at Örebro University, with an approximate size of 46 × 12 meters.

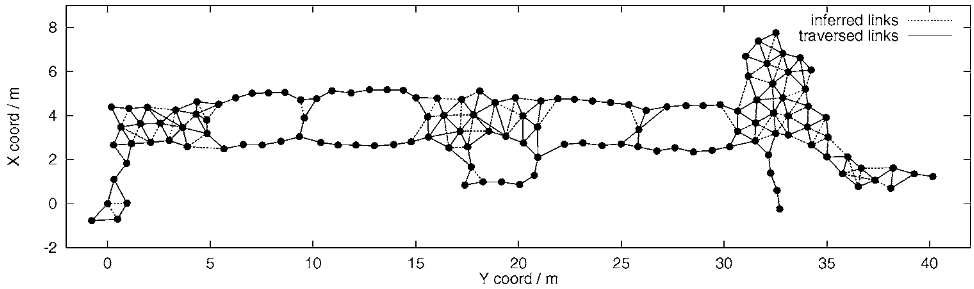

Figure 12. The map acquired with the new map learning algorithm. The links actually traversed by the robot are shown in bold, and the inferred links are shown by dotted lines (only the traversed links were used to estimate the node coordinates).

Figure 13. The global gridmap constructed with the new algorithm, by combining the recorded sonar data with the global coordinates generated by the algorithm.

Figure 14. The global gridmap constructed without the new algorithm, by combining the recorded sonar data with the global coordinates calculated using the compass and odometry. Without the algorithm, the robot fails to build a consistent map.

map building process. One iteration of the algorithm on the full map, consisting of 137 places and 188 physically traversed links, required 20 msec. on a 200 MHz Pentium II processor.

6. Related Work

Lu and Milios (1997a) considered the problem of enforcing geometric consistency in a metric map constructed using laser range-finder sensors without a compass. Their approach maintained a history of all the local frames of sensor data used to construct the map and the network of spatial relations between the frames. The spatial relations were obtained either by odometry or pairwise matching of the range-finder data in adjacent frames, using the scan matching algorithm described in Lu and Milios (1997b). A maximum likelihood algorithm was then used to derive a position estimate for each of the frames, by minimizing the Mahalanobis distance between the actual and derived relations over the whole network of frames. A drawback of this method is that it requires the inversion of a $3n \times 3n$ matrix, where n is the number of frames, so the approach is likely to be computationally expensive in large environments. This approach was extended by Gutmann and Konolige (1999) to build maps in environments containing large cycles.

A similar approach is described by Golfarelli et al. (1998), using a graph-based model of the robot's environment. Their algorithm operates under the assumption of no topological errors, so that the robot recognises a place when it visits it for a second time. Their system was based on the analogy of a mechanical spring system, in which each link in the graph is modeled by a pair of springs, a linear axial spring and a rotational one. The elasticity parameters of the springs were used to represent the uncertainty in the robot's odometry measurements, and the equilibrium position for the whole structure was then calculated, generating a $4n \times 4n$ matrix that requires inverting. The algorithm can be applied with or without a compass, although all of their experiments appear to use a compass. Their results seem qualitatively similar to ours (at least by visual inspection) despite the added complexity of the mechanical spring system model.

Shatkay and Kaelbling (1997) and Shatkay (1998) addressed the problem of incorporating metric information from odometry into robot maps based on Partially Observable Markov Decision Process (POMDP) models and enforcing geometric consistency in these maps, both with and without a compass. The sensor data from which the models were acquired were first collected by the robot under manual control, then an expectation-maximization (EM) algorithm was used to find the map which best fitted the recorded data. While this algorithm does not explicitly make the assumption of an initial topological map, in practice it depends heavily on a sufficiently good initial model (obtained in Shatkay (1998) from the recorded odometer data) to avoid local maxima and hence build topologically correct maps. The approach would not scale well to larger environments due to the large amount of data needed and the high computational cost of the EM algorithm.

A similar approach is described by Thrun et al. (1998), where an EM algorithm was used to learn an occupancy grid model of a large environment (90 m $\times$ 90 m). Again, this method is extremely expensive, requiring up to two hours of computation to generate a grid map with a spatial resolution of 1 meter, and it depends on a manually labeled set of landmarks. This approach was later extended by Burgard et al. (1999) to use a network of local gridmaps constructed from sonar data, as in our approach, instead of the pre-defined landmarks.

7. Discussion

In this paper, we have presented a relaxation algorithm for maintaining geometric consistency in a robot's map. The algorithm is computationally very cheap, enabling globally consistent map learning in real-time. Its $O(N)$ complexity compares favorably to that of matrix inversion methods such as Lu and Milios (1997a). The method is particularly efficient because it does not throw any useful information away; instead of recalculating the entire map from scratch every time, the existing solution is refined. As a result, only small changes to the map are typically required when new information is added. In the experiments conducted, we found it was only necessary to run the relaxation algorithm for a single iteration at each cycle of the map acquisition process.

The method works by minimizing an energy function in lots of small steps, rather like a Hopfield network (Hopfield, 1982). Because this energy function corresponds to the log likelihood, minimizing this function provides us with the maximum likelihood solution to the map learning problem, provided that the map is topologically correct. We have proved that the relaxation algorithm always converges to a globally optimal solution, in contrast to EM algorithms, which are subject to local optima. We should, of course, point out that our method assumes a place recognition system (in other words, no topological errors) before maximum likelihood estimation takes place, an assumption that is not made by Shatkay and Kaelbling (1997) and Burgard et al. (1999). However, all of our experiments were conducted on a real, self-navigating robot using a real self-localization system (Duckett and Nehmzow, 2001), without requiring any pre-installed environment model, thus demonstrating the efficacy of our approach.

Through our experiments, we have demonstrated a complete solution to the problem of simultaneous localization and mapping (SLAM) for indoor, office environments. This solution agrees favorably with the theoretical requirements for an ideal SLAM solution outlined by Frese and Hirzinger (2001), as follows:

1. The hybrid metric-topological representation preserves the "certainty of relations despite the uncertainty of positions".
2. This representation also means that the memory required for storage grows only linearly with the number of nodes in the map.
3. The new map learning algorithm is computationally cheap, having a cost that is linear in the number of places stored in the map.

There is one possible situation where the new algorithm may be significantly slower than usual—that is if the robot closes a very large cycle and the accumulated odometric error is very high. Here it might be faster to use a simultaneous equation solver, as in Lu and Milios (1997a) and Gutmann and Konolige (1999), depending on the magnitude of the initial error in the map (cf. Fig. 5). However, while theoretically possible, this situation tends to be very rare in practice, and relaxation should be faster in the vast majority of cases. The relatively large loops in the test environment of Fig. 11 did not pose a problem in any of the experiments conducted.

A further benefit of the new algorithm is that all of the global noise parameters are continually re-estimated by the algorithm itself, so we do not need to compute the values of these parameters ourselves. This is a distinct advantage for building autonomous robots, since reducing the number of pre-installed parameter values reduces the dependence of the robot on a priori knowledge provided by the system designer.

Future work will investigate more intelligent strategies for selecting the landmarks or place signatures in the topological map (see Marsland et al. (2001) for some first results). The current strategy of adding new places at 1 meter intervals is clearly inefficient, and could be greatly improved. We will also consider building maps in environments containing large cycles.

Appendix A: A Simplification of the Algorithm

In the derivation of the algorithm in Section 3, various assumptions were made, e.g., that small angle approximations could be made, and that the covariance matrix C_{ji} was symmetric. In fact, the most common noise model estimate for a mobile robot's position is an ellipse, with the major axis lying perpendicular to the direction of travel, as in Smith and Cheeseman (1986). However, when a compass is used, the noise on the distance measurement (odometry) could be greater than the noise on the rotation measurement, so that the major axis lies along the direction of travel. Both of these models assume that the robot is perfectly symmetrical, which is not necessarily the case. For instance, the robot may measure turns more accurately anticlockwise than clockwise, or one of the wheels could slip more than the other, so that the robot does not follow a straight line.

In previous experiments (Duckett et al., 2000), we found that surprisingly good results can be obtained by making one further simplifying assumption: that the noise in the robot's position estimates is distributed equally in all directions around points in Cartesian space according to a Gaussian distribution, i.e., a circle rather than an ellipse in used to represent the area in which the robot may be located with non-negligible probability. Thus, the uncertainty in any point to point measurement can be represented by a single variance measure (this makes the covariance matrix C_{ji} proportional to the identity matrix). For dead reckoning, this can be taken as some small proportion, e.g., 5%, of the distance travelled by the robot. The algorithm can then be described in two steps, as given in Fig. 15. While less accurate than the algorithm in Fig. 3, this variant is much easier to implement, and is useful when only approximate geometric information is required in the robot's map.

Appendix B: Relaxation without a Compass

Although in this paper we have used a compass to provide a global 'North' from which all angular measurements are taken, this should not be necessary if the robot is equipped with some other means of measuring the angles that it turns through. Suppose that the robot has the ability to measure the change in its pose when moving between two adjacent places, e.g., by matching laser range-finder scans, as in Weiss and von Puttkamer (1995), Lu and Milios (1997a), and

1. For each node i, do:

   a) For each of the neighbors j of node i, i.e., the places that are topologically connected to i, obtain an estimate r'_{ji} of the coordinates of node i using

   $$\mathrm{r}'_{ji} = \mathrm{r}_j + \mathrm{D}_{ji}, \tag{28}$$

   where $\mathrm{r}_j = (x_j, y_j)^T$ refers to the coordinates of node j, and $\mathrm{D}_{ji} = d_{ji}\binom{\cos\theta_{ji}}{\sin\theta_{ji}}$.

   b) Combine the position estimates r'_{ji} for all j to produce new coordinates r'_i for node i using

   $$\mathrm{r}'_i = \left(\sum_j{}' v_{ji}^{-1}\right)^{-1} \sum_j{}' \left(v_{ji}^{-1}\mathrm{r}'_{ji}\right), \tag{29}$$

   where $\sum_j'$ refers to the sum over the neighbours of node i, and v_{ji} is the variance for the link from node i to node j.

2. Repeat from step 1 untill the change in energy falls below some pre-defined threshold, or some other stopping criterion is reached.

Figure 15. A simplification of the map learning algorithm, based on the assumption of circular noise in the robot's odometry.

Gutmann and Konolige (1999). To maintain geometric consistency in the map, the relaxation algorithm can then be extended to estimate a pose $\mathbf{r}_i = (x_i, y_i, \theta_i)^T$ for each node i of the map, where the angle θ_i corresponds to the orientation of the robot in which the scan for that particular place was taken. Equation (10) of the algorithm in Fig. 3 is rewritten as:

$$\mathbf{r}'_{ji} = \mathbf{r}_i + \mathbf{F}_{ji}, \tag{30}$$

where

$$\mathbf{F}_{ji} = \begin{pmatrix} x_{ji}\cos\theta_i - y_{ji}\sin\theta_i \\ x_{ji}\sin\theta_i + y_{ji}\cos\theta_i \\ \phi_{ji} \end{pmatrix}, \tag{31}$$

with $(x_{ji}, y_{ji}, \phi_{ji})^T$ referring to the relative pose between nodes i and j, after Lu and Milios (1997a). The derived orientation θ_j for node j will be an estimate relative to an arbitrary 'North', i.e., the orientation of the robot at the origin.

With suitable changes to the covariance matrix (i.e., making it 3×3, with estimates of the noise in the angles), the nature of the algorithm can then be preserved. Obviously, this requires the storage of more angles, and there will be noise in the rotational measurements. This is why using a compass, even in indoor environments where there are likely to be a lot of electromagnetic disturbances, is advantageous.

References

Burgard, W., Fox, D., Jans, H., Matenar, C., and Thrun, S. 1999. Sonar-based mapping with mobile robots using EM. In *Proc. of the International Conference on Machine Learning*.

Duckett, T. 2000. Concurrent map building and self-localisation for mobile robot navigation. Ph.D. Thesis, Department of Computer Science, University of Manchester.

Duckett, T., Marsland, S., and Shapiro, J. 2000. Learning globally consistent maps by relaxation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'2000)*, San Francisco, CA.

Duckett, T. and Nehmzow, U. 1999. Exploration of unknown environments using a compass, topological map and neural network. In *Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, pp. 312–317.

Duckett, T. and Nehmzow, U. 2001. Mobile robot self-localisation using occupancy histograms and a mixture of Gaussian location hypotheses. *Robotics and Autonomous Systems*, 34(2/3):119–130.

Frese, U. and Hirzinger, G. 2001. Simultaneous localization and mapping—A discussion. In *Proceedings of the IJCAI-2001 Workshop on Reasoning with Uncertainty in Robotics (RUR'01)*, D. Fox and A. Saffiotti (Eds.).

Golfarelli, M., Maio, D., and Rizzi, S. 1998. Elastic correction of dead-reckoning errors in map building. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*, Victoria, Canada, pp. 905–911.

Gutmann, J.-S. and Konolige, K. 1999. Incremental mapping of large cyclic environments. In *Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA.

Hinkel, R. and Knieriemen, T. 1988. Environment perception with a laser radar in a fast moving robot. In *Symposium on Robot Control (SYROCO'88)*, Karlsruhe, Germany.

Hopfield, J. 1982. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences*, USA, 79:2554–2558.

Lu, F. and Milios, E. 1997a. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349.
Lu, F. and Milios, E. 1997b. Robot pose estimation in unknown environments by matching 2D range scans. *Intelligent and Robotics Systems*, 18:249–275.
Marsland, S., Nehmzow, U., and Duckett, T. 2001. Learning to select distinctive landmarks for mobile robot navigation. *Robotics and Autonomous Systems*, 37(4):241–260.
Moravec, H. and Elfes, A. 1985. High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'85)*, St. Louis, Missouri, pp. 116–121.
Reif, F. 1982. *Fundamentals of Statistical and Thermal Physics*. McGraw-Hill: New York.
Shatkay, H. 1998. Learning models for robot navigation. Ph.D. Thesis, Department of Computer Science, Brown University.
Shatkay, H. and Kaelbling, L.P. 1997. Learning topological maps with weak local odometric information. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*.
Smith, R. and Cheeseman, P. 1986. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68.
Thrun, S., Burgard, W., and Fox, D. 1998. A probabilistic approach to concurrent mapping and localisation for mobile robots. *Machine Learning*, 31(5):1–25.
Weiss, G. and von Puttkamer, E. 1995. A map based on laserscans without geometric interpretation. In *Proceedings of Intelligent Autonomous Systems 4 (IAS-4)*, Karlsruhe, Germany, pp. 403–407.
Yamauchi, B. and Beer, R. 1996. Spatial learning for navigation in dynamic environments. *IEEE Transactions on Systems, Man and Cybernetics Section B*, **26**(3).

Tom Duckett is an Assistant Professor at Örebro University, where he is currently setting up a new research laboratory in robot learning within the Centre for Applied Autonomous Sensor Systems. He got his B.Sc. from Warwick University in Computer and Management Sciences, M.Sc. from Heriot-Watt University in Knowledge Based Systems, and Ph.D. from Manchester University on simultaneous localisation and map-building for mobile robot navigation. His research interests include autonomous robots, machine learning and neural networks, artificial intelligence, and biology inspired sensor systems.

Stephen Marsland recently completed his Ph.D. on unsupervised novelty detection and is currently a lecturer of machine learning at the University of Manchester. Prior to that he completed a degree in Mathematics and Computation at Somerville College, University of Oxford. His research interests include many aspects of theoretical machine and statistical learning, such as neural networks, self-organisation and topology preservation, and Gaussian processes. He is interested in many applications of machine learning techniques to real world domains, such as robotics, medical informatics and bioinformatics.

Jonathan Shapiro got his Ph.D. from UCLA in theoretical condensed matter physics. He works in the fields of theory of neural networks, application of neural networks, particularly to cognitive modelling, and theory of evolutionary algorithms. He is currently Senior Lecturer in Computer Science at the University of Manchester, and External Faculty member of The Santa Fe Institute.