Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Learning Globally Consistent Maps by Relaxation

Tom Duckett
Department of Technology
University of Örebro
SE-70182 Örebro
Sweden
Tom.Duckett@aass.oru.se

Stephen Marsland & Jonathan Shapiro
Department of Computer Science
University of Manchester
Manchester M13 9PL
England
{marslans,jls}@cs.man.ac.uk

Abstract

Mobile robots require the ability to build their own maps to operate in unknown environments. A fundamental problem is that odometry-based dead reckoning cannot be used to assign accurate global position information to a map because of drift errors caused by wheel slippage. This paper introduces a fast, on-line method of learning globally consistent maps, using only local metric information. The approach differs from previous work in that it is computationally cheap, easy to implement and is guaranteed to find a globally optimal solution. Experiments are presented in which large, complex environments were successfully mapped by a real robot, and quantitative performance measures are used to assess the quality of the maps obtained.

1 Introduction

Maps are essential for mobile robot navigation in complex environments, being needed for self-localisation, path planning and human-robot interaction. While successful navigating robots have been developed using pre-installed maps, such as CAD models [12], 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 wheel slippage, which can lead to an inconsistent mapping of the environment. To maintain a coherent representation of the environment which 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 globally optimal solution. Experiments are presented in which large, complex environments were successfully mapped by a real robot, and quantitative performance measures are used to assess the quality of the maps obtained.

The map representation consists of a topologically connected network of places, where each link is labelled with local metric information describing the relative distance and 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 (as in Lu and Milios [8]), and the algorithm finds an optimal set of coordinates using only the local metric relations between places.

In this paper, we assume that the robot has the ability to determine its orientation using a compass, and the ability to recognise its own place in the map. Full details of the self-localisation mechanism used in these experiments can be found in [4].

1.1 Previous Work

Lu and Milios [8] considered the problem of enforcing geometric consistency in a metric map. 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 [9]. A maximum likelihood algorithm was then used to derive a position estimate for each of the frames, by minimising the Mahalanobis distance between the actual and derived relations over the whole network of frames. A drawback of this method is that it re-

0-7803-5886-4/00/$10.00© 2000 IEEE

quires 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.

A similar approach is described by Golfarelli *et al.* [5], using a graph-based model of the robot's environment. The system was based on the analogy of a mechanical spring system, in which each link in the graph is modelled 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. Again, however, this approach requires the inversion of a large matrix.

Shatkay and Kaelbling [11, 10] addressed the problem of incorporating metric information from odometry into robot maps based on Hidden Markov Models (HMMs) and enforcing geometric consistency in these maps. The sensor-motor data from which the models were acquired were first collected by the robot under manual control, then an expectation-maximisation (EM) algorithm was used to find the map which best fitted the recorded data. In this approach, the conditions of additivity and anti-symmetry were enforced directly in the reestimation procedure for obtaining the probabilities in the HMM. This algorithm is heavily dependent on a good initial model 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.

2 The Robot

The experiments presented here were conducted using the Nomad 200 robot shown in Fig. 1. The robot is equipped with sonar and infrared sensors mounted around its turret, which can be rotated independently relative to the base of the robot. Two other motors located in the base of the robot are used to control the translational and rotational movement of the robot.

2.1 Compass Sense

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. 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 sensors, as in the self-orientation system described by Li *et al.* [7], or by using correlation with a vision-based map of the ceiling as in Thrun *et al.* [13].

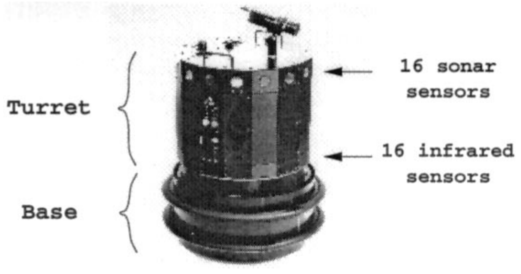

Figure 1: The Nomad 200 mobile robot. A flux gate compass was used to keep the turret, and therefore the sensors, at a constant orientation during map building.

3 Dead Reckoning

The compass sense was also used for the on-line dead reckoning. We used the robot's wheel encoders to measure the distance travelled, but the rotational component was obtained from the relative angular displacement of the turret against the direction of travel. This had the effect of removing the accumulated rotational drift affecting the robot's raw odometry (see Fig. 2), because the turret was anchored to 'North' by the compass sense, leaving a translational drift error of up to 5% of distance travelled.

3.1 The Noise Model

Due to the use of the compass sense to remove the rotational error in the robot's odometry, we were able to use a very simple noise model to represent the uncertainty in the robot's distance measurements. We assume 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, and the area in which the robot may be located with non-negligible probability is therefore modelled by a circle. Thus, the uncertainty in any point to point measurement can be represented by a single variance measure. For dead reckoning, this was taken to be 5% of the distance travelled by the robot.

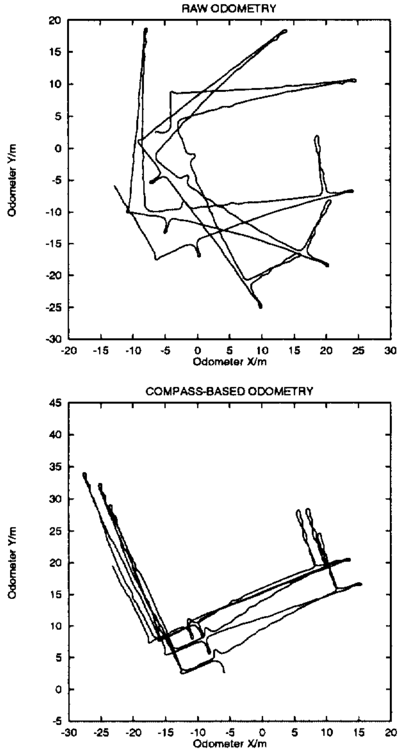

Figure 2: Top: raw odometry. Bottom: compass-based odometry. The accumulated rotational drift in the robot's raw odometry was removed on-line using the compass sense.

An alternative would be to model the area of non-negligible probability as an ellipse, using a covariance matrix to represent the noise in two distinct dimensions (or three without the compass sense), as in [6]. We found that a covariant noise model was unnecessary for the experiments presented here, though it would be relatively straightforward to generalise the algorithm presented in this paper to use covariance matrices rather than variances.

4 Map Learning

The map consists of a set of n place nodes, and a set of links which connect some pairs of places. Each place i is associated with a Cartesian coordinate (x_i, y_i) and a variance v_i which represents the uncertainty in the calculation of that coordinate. Each link connects two places i and j, and is associated with a metric relation (d_{ij}, θ_{ij}) which describes the relative distance d_{ij} and angle θ_{ij} between the two places, where the angle is an absolute measurement obtained from the compass sense. The uncertainty in the measurement of this relation is represented by a variance u_{ij}. In this paper, the links were constrained to be bi-directional, that is, $d_{ij} = d_{ji}$ and $\theta_{ij} = \theta_{ji} + \pi$.

The maps were acquired incrementally using the exploration strategy presented in [3] — lack of space prevents a detailed description here. In this approach, the robot continuously tries to expand the territory which has already been charted, moving to the edge of the existing map to look for new unexplored territory. In these experiments, places were added to the map at 1 m intervals. Whenever a new place is added to the map, or a previously unexplored link between existing places is traversed, the distance d_{ij} and heading θ_{ij} of the robot between the two places is recorded. In addition, the variance u_{ij} in this measurement was estimated as 5% of the measured distance.

To maintain geometric consistency in the map, the following relaxation algorithm was applied.

4.1 The Relaxation Algorithm

At each iteration of the algorithm, a two-step procedure is carried out for each node i of the map in turn:

Step 1. For each of the neighbours j of node i, i.e., the places which are topologically connected to i, an estimate (x'_{ji}, y'_{ji}) of the coordinate of node i is obtained using

$$\begin{aligned} x'_{ji} &= x_j + d_{ji}\cos\theta_{ji}, \\ y'_{ji} &= y_j + d_{ji}\sin\theta_{ji}, \end{aligned}$$

where the coordinate of j is denoted by (x_j, y_j), and the variance v_{ji} in this estimate is obtained using

$$v_{ji} = v_j + u_{ji},$$

where v_j refers to the variance for node j and u_{ji} to the variance for the link from j to i.

Step 2. The position estimates (x'_{ji}, y'_{ji}) for all j are then combined to produce a new coordinate for node i. First, the new variance v_i for node i is calculated as

$$\frac{1}{v_i} = \sum_j{}' \frac{1}{v_{ji}},$$

where $\sum'_j$ refers to the sum over the neighbours of node i. The new coordinate (x_i, y_i) is then obtained by calculating the mean of the estimates (x'_{ji}, y'_{ji}) weighted by $1/v_{ji}$ as

$$\begin{aligned} x_i &= \sum_j{}' \frac{x'_{ji} v_i}{v_{ji}}, \\ y_i &= \sum_j{}' \frac{y'_{ji} v_i}{v_{ji}}. \end{aligned}$$

The algorithm is repeated until some arbitrary stopping criterion is reached, for example, when the total change in the coordinates falls below some threshold. In the on-line map building experiments presented, the algorithm was run for a single iteration each time the robot moved to a different place in the map, taking less than 1 second on a 486 PC.

The basic principle behind the relaxation algorithm can be explained as follows. The idea is to pick each node in turn, and then move it "to where its neighbours think it should be". By repeated application of this rule, the coordinates in the map converge upon a globally optimal solution.

4.2 Proof of Convergence

Each link in the map can be thought of as a spring which connects two adjacent places i and j [8, 5, 10]. The spring reaches minimum energy when the relative displacement between the coordinates of i and j is equal to the vector $(d_{ij},\ \theta_{ij})$ 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 minimising the following energy function:

$$E = \sum_i \sum_j{}' \left[(x_i - x_j + d_{ij} \cos\theta_{ij})^2 + (y_i - y_j + d_{ij} \sin\theta_{ij})^2 \right],$$

where $\sum'_j$ refers to the sum over the neighbours of a given node. To prove convergence, it can be shown that updates to any node will always result in a decrease in energy (see [1] for details). The algorithm will therefore always converge to a minimum in the energy function, because this function is bounded below by zero. Since the energy function is quadratic, this can only be a global minimum.

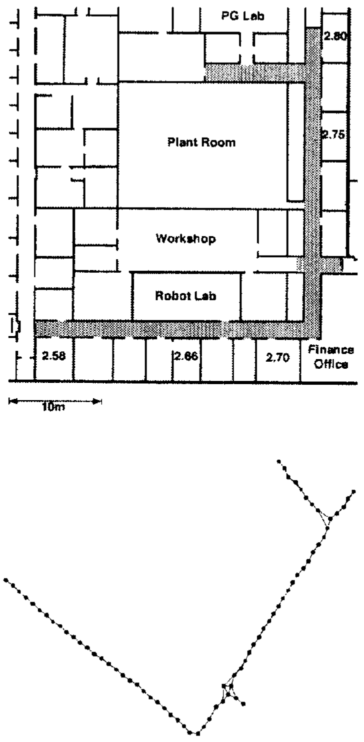

Figure 3: Top: floor plan of a corridor environment. Bottom: the corresponding map acquired by the robot (see also uncorrected odometry in Fig. 2).

5 Results

The complete map building system was tested successfully in a number of untreated, real world environments, which were subject to transient changes such as moving people, doors opening and closing, etc. An example map acquired by the robot in a corridor environment of size $34m \times 33m$ is shown in Fig. 3.

In order to illustrate the convergence of the relaxation algorithm, we also conducted an off-line experiment in which the coordinates of a self-acquired map were randomly reinitialised to arbitrary values, then the algorithm was repeated until the map returned to its globally consistent solution (see Fig. 6). Also, to illustrate the ability of the algorithm to propagate the uncertainty in the robot's distance measurements, we plotted the variances in the pose estimates which it produced in a non-corridor environment (see Fig. 4).

To assess the quality of the maps obtained, we considered the robot's ability to localise itself using its own self-acquired map. Here, we considered the robot's ability to relocalise under global uncertainty, i.e., to

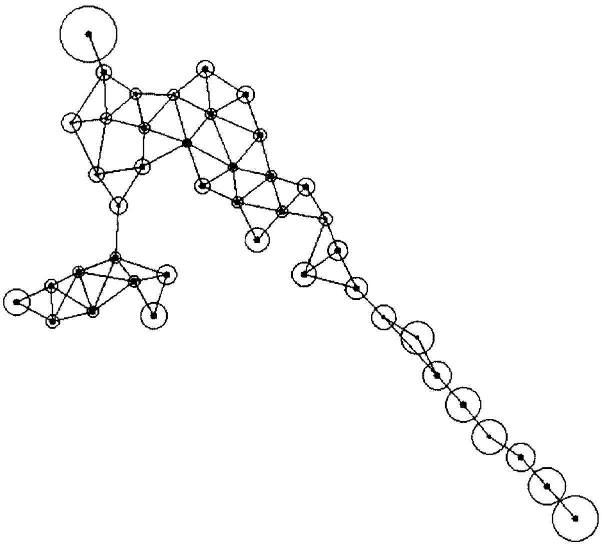

Figure 4: Propagation of uncertainty by the relaxation algorithm. The variance in each of the coordinates after applying the relaxation algorithm is indicated by the radius of the corresponding circle (the variances were exaggerated by a factor of 10 here for illustration purposes).

recover its position after becoming lost. To assess localisation performance, the *Uncertainty Coefficient* $U(L \mid R)$ was measured against the distance travelled by the robot from an unknown starting position using wall-following (see Fig. 5). This statistic measures the extent to which the robot's response, R (the location estimates produced by self-localisation) predicts the robot's true location, L, as

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

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 and the robot's true location being j. Full details of the experimental procedure used to assess localisation performance can be found in [2].

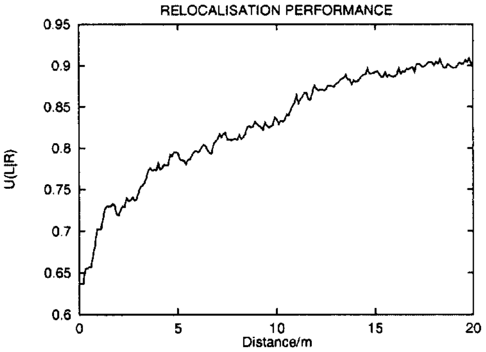

Figure 5: Relocalisation performance $U(L \mid R)$ under global uncertainty in the corridor environment of Fig. 3 using the self-acquired map.

6 Conclusion

In this paper, we presented a relaxation algorithm for maintaining geometric consistency in a robot's map. The method uses three sources of perceptual information; (1) some external place recognition system, (2) a global orientation obtained from a compass, and (3) local distance information from odometry.

The new algorithm is computationally cheap compared to previous approaches. During on-line operation, we found that only a single iteration of the algorithm was required at each stage of the map building process. For the worst case of a completely connected graph, the complexity of the algorithm would be $O(n^2)$, where n is the number of nodes. However, for a map, the number of links per node will not grow with the size of the map, so the complexity is linear or $O(n)$. (In our system, the robot attempts to space the nodes at equal intervals, so the maximum connectivity per node is 6.) This compares favourably with the $O(n^3)$ complexity for matrix methods such as the Lu and Milios algorithm [8].

Our method works by minimising an energy function in lots of small steps. We have proved that the algorithm always converges to a global solution, in contrast to expectation-maximisation algorithms [11, 14], which are subject to local optima.

Finally, to demonstrate the quality of the maps obtained, we measured the robot's ability to carry out global localisation using its own self-acquired map. Future work will include building maps in large environments containing loops.

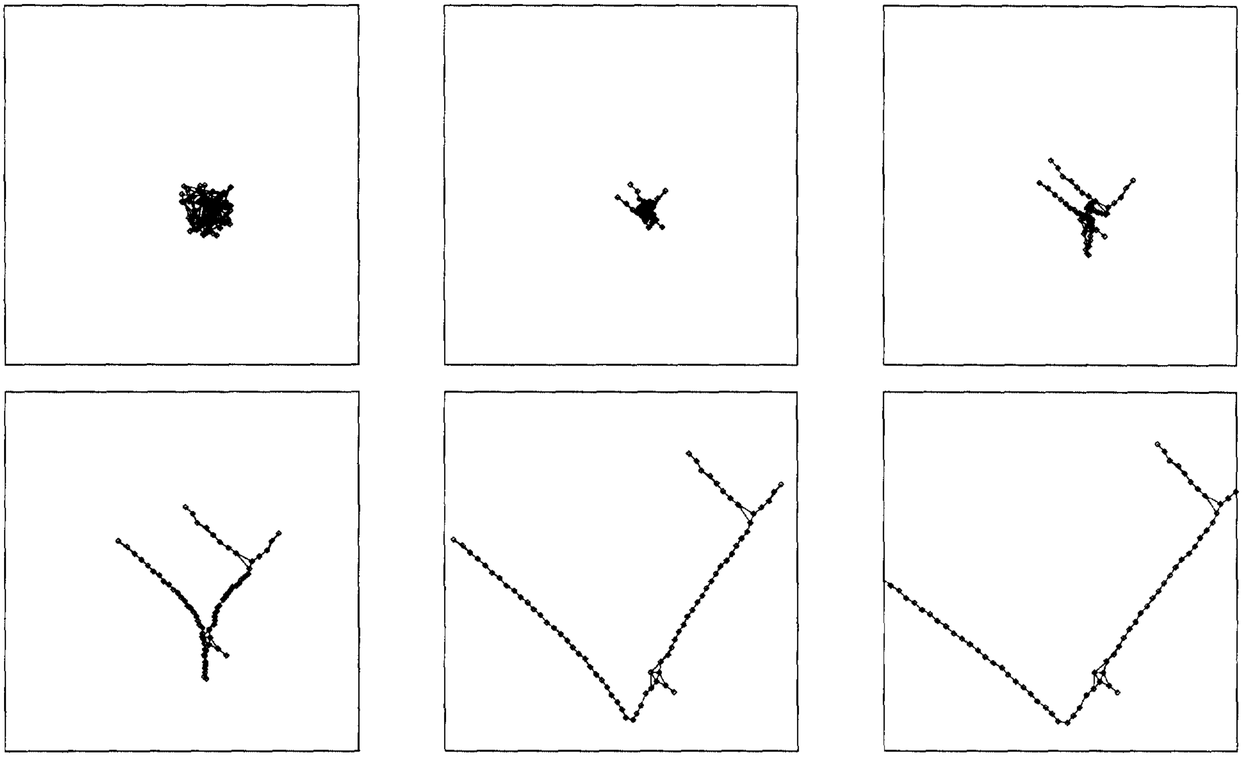

Figure 6: Convergence of the relaxation algorithm. In the first picture, the coordinates of the self-acquired map shown in Fig. 3 have been randomly reinitialised. The remaining pictures show the map after 5, 25, 50, 500 and 1000 iterations respectively of the relaxation algorithm.

References

[1] T. Duckett. *Concurrent Map Building and Self-Localisation for Mobile Robot Navigation.* PhD thesis, Dept. of Computer Science, Manchester University, 1999.

[2] T. Duckett and U. Nehmzow. Mobile robot self-localisation and measurement of performance in middle scale environments. *J. Robotics and Autonomous Systems*, 24(1–2), 1998.

[3] T. Duckett and U. Nehmzow. Exploration of unknown environments using a compass, topological map and neural network. In *Proc. IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, 1999.

[4] T. Duckett and U. Nehmzow. Knowing your place in real world environments. In *Proc. EUROBOT '99, 3rd European Workshop on Advanced Mobile Robots.* IEEE Computer Society Press, 1999.

[5] M. Golfarelli, D. Maio, and S. Rizzi. Elastic correction of dead-reckoning errors in map building. In *Proc. IROS*, pages 905–911, 1998.

[6] J.J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation.* Kluwer Academic Publishers, 1992.

[7] G. Li, B. Svensson, and A. Lansner. Self-orienting with on-line learning of environmental features. *Adaptive Behaviour*, 6(3/4):535–566, 1998.

[8] F. Lu and E.E. Milios. Globally consistent range scan alignment for environment mapping. *J. Autonomous Robots*, 4:333–349, 1997.

[9] F. Lu and E.E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Proc. IROS*, 18:249–275, 1997.

[10] H. Shatkay. *Learning Models for Robot Navigation.* PhD thesis, Dept. of Computer Science, Brown University, 1998.

[11] H. Shatkay and L.P. Kaelbling. Learning topological maps with weak local odometric information. In *Proc. IJCAI*, 1997.

[12] A. Stevens, M. Stevens, and H. Durrant-Whyte. OxNav: Reliable autonomous navigation. In *Proc. ICRA*, pages 2607–2612, 1995.

[13] 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. In *Proc. ICRA*, 1999.

[14] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localisation for mobile robots. *Machine Learning*, 31(5):1–25, 1998.