CONCURRENT MAP BUILDING AND SELF-LOCALISATION FOR MOBILE ROBOT NAVIGATION

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

October 2000

By
Thomas D. Duckett
Department of Computer Science

Contents

List of Tables

List of Figures

Abstract

This thesis addresses the problem of navigation by a mobile robot operating in large, real world environments which have not been modified for the purpose of robot navigation. Maps are essential for mobile robot control in complex environments, being needed for self-localisation, path planning and human-robot interaction. In attempting to navigate in *unknown* environments, a self-governing robot is faced with a fundamental dilemma: to explore and build maps of uncharted territory, the robot needs to know its location, but in order to know its location, the robot needs a map.

A unified solution to the problems of simultaneous map building and self-localisation is presented, which is embedded in a hybrid deliberative-reactive control architecture. A particular contribution of the work is that all of the environment and location models, feature models and sensor-motor competences required for navigation are acquired independently by the robot. The learning techniques include self-organisation, where no teaching signal is required, and self-supervised learning, where all of the training examples are generated by the robot. Consequently, the new system is able to build its own maps and navigate in many different, indoor environments that are unfamiliar, without requiring intervention by a human operator.

During an exploration phase, the robot builds a graph-like representation of its environment, in which each location is identified by a description of the robot's sensory information known as a *landmark*. To determine an appropriate landmark recognition mechanism, an experimental procedure was developed which permitted different algorithms to be compared under identical experimental conditions. With this method, existing approaches to landmark recognition were evaluated, and a new self-localisation system was developed. To overcome problems such as perceptual aliasing — the fact that landmarks may not be unique to individual places — a self-localisation algorithm was developed which accumulates sensory

evidence over time so that the robot can recover its position even after becoming lost. To maintain geometric consistency in the robot's map, an optimisation algorithm was developed, which is proved to converge to a globally optimal solution. To explore unknown environments, an artificial neural network was trained to recognise areas of uncharted territory. Quantitative performance measures were applied throughout the work, guiding the development of the new algorithms and allowing the effects of individual system components to be investigated.

Finally, the research was validated by building a complete, self-navigating mobile robot capable of operation in complex, untreated environments of several hundred metres squared. In this system, human intervention is only required to specify a goal location, and all processing is carried out in real-time on board the robot, thereby increasing the autonomy of the robot.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Copyright

Acknowledgements

I am indebted firstly to my Supervisor, Dr. Ulrich Nehmzow, for his guidance, feedback, good humour, commitment and enthusiasm for the work over the past four years (I will always remember the bothy trips to the mountains). Secondly, I thank my Advisor, Prof. David Bree, for his guidance and advice, for providing a stimulating and supportive research environment within the AI Group at Manchester, and for his invaluable assistance during the writing of the final drafts of this thesis. I am indebted also to Dr. Jon Shapiro for many useful discussions, and who has taught me much by example (probably without realising it), particularly during my service as assistant for his labs and examples classes. I thank also my external examiner, Prof. Andreas Kurz for providing a friendly and informative assessment of this work.

To my fellow robotics PhD students, Stephen Marsland, Ten-Min Lee, Carl Owen and Jason Fleischer, for their support, advice, prolific tea drinking, five-a-side football and collective sense of humour. I thank Stephen especially for his numerous readings and comments on my papers and previous drafts of the thesis, and our lab technician, Andrew Pickering, for keeping mobile robot *FortyTwo* up and running. In addition, I thank the whole of the AI Group at Manchester for their friendship, constructive criticism and many interesting Wednesday lunchtimes, and the rest of the Department, especially the porters, for accepting my many experiments around the building with good grace and humour.

Thankyou to the lab organisers and staff responsible for my part-time work as a demonstrator, particularly Dr. Mary McGee Wood for her generosity throughout the first year AI labs. I gratefully acknowledge a scholarship provided by the Department of Computer Science, and the help of Dr. Norman Paton and Dr. Nick Filer in arranging and initiating my PhD studentship at the University of Manchester. I acknowledge also the support of the British Council and the facilities provided by Bremen University during a three week research visit by myself and Ulrich earlier this year.

I would also like to extend this acknowledgement to the many educators prior to my stay in Manchester, including those at Heriot-Watt, Karlsruhe and Warwick Universities, and the teachers of Bourne Grammar School in Lincolnshire, who made it possible for me to do this. And to my new employers at Örebro University in Sweden, particularly Dr. Alessandro Saffiotti for his comments on the penultimate draft of the thesis, and to the anonymous reviewers who have provided much useful feedback on the papers published. Finally, I dedicate this work to my family and friends, without whom I could never have made it.

Chapter 1

Concurrent Map Building and Self-Localisation for Mobile Robot Navigation

About this chapter. This chapter begins by introducing the problem addressed by this thesis. The scientific context is then discussed, including the relevance of the thesis to mobile robotics and other disciplines. This is followed by details of the experimental method used to tackle the problem and a summary of the contributions made to mobile robotics.

1.1 Motivation

In recent years, there has been a great deal of interest in the use of mobile robots for automation. These machines have the potential to carry out many tasks which are considered undesirable or difficult for humans, for example, due to hazardous working conditions (nuclear reactors) or a shortage of skilled labour (healthcare). Other reasons for using robots include freeing human labour from menial and repetitive work (transportation and domestic service), increasing safety and reliability by augmenting human labour with robot assistance (inspection and surveillance), increasing productivity (farming and mining), and applications in education and leisure (tour guides and toys).

Navigation is an essential competence for mobile robots, because most of the potential applications require the ability to move purposefully between locations.

Without navigational skills, a mobile robot would have to resort to random movement, which would severely limit its possible uses.

There have been some well-publicised successes in the field of mobile robotics (for example, in extra-terrestrial exploration). However, almost every case so far has consisted of a one-off exercise in systems engineering for a specific application, and to a greater or lesser extent, these systems cannot operate without human intervention. For example, some robots are dependent on remote teleoperation (Mars rovers), while others require modifications to their environment, e.g., using specially placed beacons or induction loops to guide the robot (automatic guided vehicles or AGVs). The few systems which can operate without on-line assistance typically depend upon pre-installed world knowledge, e.g., using a predefined CAD model of the environment to navigate. Once the robot is moved to a new and unfamiliar set of surroundings, these systems are incapable of purposeful activity, requiring reprogramming for each new application.

This thesis takes a step towards general purpose robots which are capable of independent operation in many different applications and environments. The basic idea is to avoid pre-installation of world knowledge by the system designer by enabling the robot to adapt its own internal representations to whatever features are naturally present in a given environment. In this approach, the world models required for navigation are acquired independently by the robot. The ultimate goal was to produce a mobile robot which can navigate from scratch in unmodified environments that are unfamiliar, without requiring intervention by a human operator.

Finally, a further motivation for studying mobile robotics lies in understanding the underlying principles of successful navigation systems. In building complete navigating robots, the results of mobile robotics research can inform and validate hypotheses of spatial cognition in biological systems as well as robots.

1.2 The Problem

Gallistel (1990, p. 35) defines *navigation* as follows:

> "Navigation is the process of determining and maintaining a course or trajectory from one place to another. Processes for estimating one's position with respect to the known world are fundamental to it. The

known world is composed of the surfaces whose locations relative to one another are represented on a map."

It follows from this definition that if a robot is to be self-navigating, it needs both some representation of the environment, in general a map, and the ability to locate itself within that representation in order to navigate between arbitrary locations. Successful mobile robots have been developed where some form of map is pre-installed by the human designer. However, in order to operate in *unknown* environments, a navigating mobile robot is faced with a fundamental dilemma: to explore and build maps of uncharted territory, the robot needs to know its location, but in order to know its location, the robot needs a map.

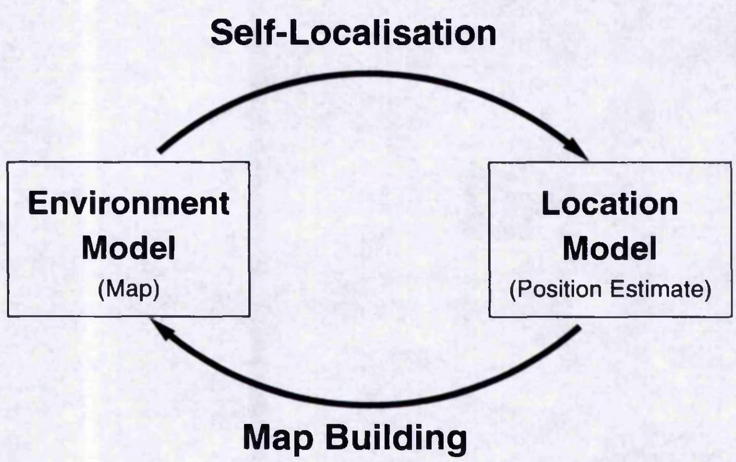

Figure 1.1: The problem of concurrent map building and self-localisation.

The circular nature of the problem is illustrated in figure 1.1. The robot has to maintain two representations at the same time: firstly, an *environment model* or map, and secondly, a *location model* or position estimate. For example, the environment model might consist of a set of discrete locations, and the corresponding location model might be a probability distribution over these locations. This thesis assumes that both of these models are initially completely unknown to the robot. Therefore, the robot must be able to acquire its own map and simultaneously maintain a sufficiently accurate position estimate for useful map building to be possible.

In fact, a more complete picture is provided by the diagram in figure 1.2. Here, the problem is shown within the context of a navigating robot interacting

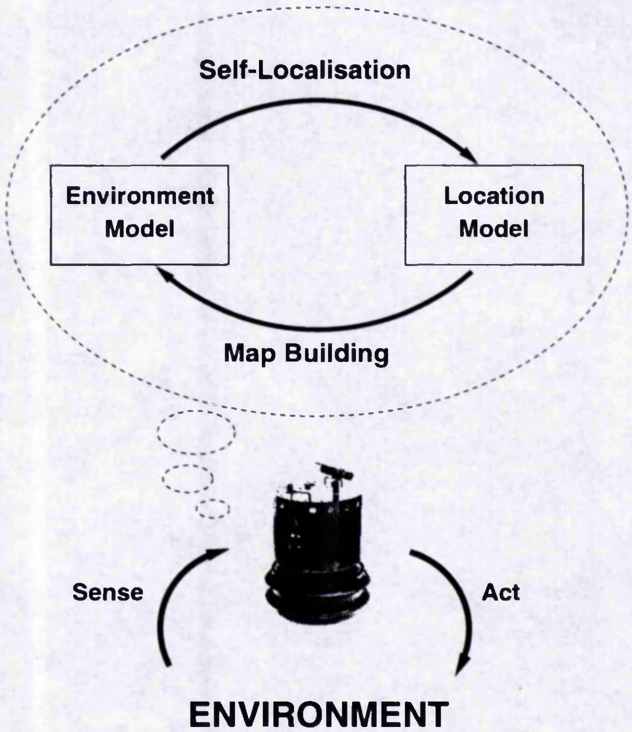

Figure 1.2: The situated problem.

with a real world environment. Map building and self-localisation are active perceptual processes, through which the robot must make decisions about how to sense and act in the world, i.e., how to *explore* the environment in order to obtain useful sensory information. Updating the map requires both the current location and new sensory information, while self-localisation requires the current environment model and new sensory information. In order to obtain this new sensory information, the robot needs to be capable of finding its way through existing charted territory and into new areas of unexplored territory. This last requirement entails being able to determine and follow routes using the map.

In summary, the following competences are required by a mobile robot to solve the problem of concurrent map building and self-localisation:

- *Exploration.* Where to travel in order to obtain useful sensory information.
- *Map Learning.* How to represent the environment.
- *Self-Localisation.* How to establish the robot's location in the map.

For building maps in complex environments, "exploration" includes the competence of *way finding*, which refers to the ability to plan paths and monitor progress towards reaching a goal location.

In this thesis, novel mechanisms are combined with existing techniques for exploration, map learning and self-localisation. These mechanisms are then integrated to produce a solution to the problem of concurrent map building and self-localisation. A further product of this research is a complete mobile robot which can navigate independently and reliably between specified locations in complex, real world environments which have not been modified for the purpose of robot navigation.

1.3 The Context

There is no generally accepted view on what exactly constitutes a robot. For the purposes of this thesis, Arkin's definition of an *intelligent robot* (1998, p. 2) seems the most appropriate:

> "An intelligent robot is a machine that is able to extract information from its environment and use knowledge about its world to move safely in a meaningful and purposeful manner."

The ability to move purposefully rather than randomly between known locations is an essential requirement for most applications of mobile robots. Much research in mobile robotics has therefore concentrated on the topic of navigation.

Navigation has also received much attention in psychology and biology. Psychologists have studied the development of cognitive maps in humans, and there has been much debate among biologists over whether animals use maps at all for navigation (see for example, O'Keefe & Nadel (1977) versus Bennet (1996)). This thesis does not enter into this particular debate — I make no claims concerning the biological validity of the mechanisms presented, and Gallistel's definition of navigation is assumed from the start.

Many psychologists and biologists regard the ability to construct "short cuts" — that is, to infer novel routes from stored spatial information — as a defining characteristic of a map. In this thesis, path planning techniques are used to find new routes through previously explored territory, and also to infer possible routes through territory which has not yet been visited by the robot (described in section 9.6).

A useful analogy concerning the scale of the navigational tasks under consideration is available in the biological literature (see e.g., Wehner (1996)), where the following categories are used:

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 sensory range of the home or goal location, and navigating within the wider environment (e.g., foraging desert ants).

3. *Large-Scale.* Navigating over very large distances between different environments (e.g., inter-continental navigation by migratory birds).

In mobile robotics, the navigational tasks which can be accomplished depend greatly on the type of environment model used by the robot. Some robots use detailed metric maps, consisting of high resolution geometric representations with an explicit Cartesian reference frame, for example, CAD models (Stevens *et al.* 1995) or occupancy grids (Moravec & Elfes 1985). Others use qualitative topological maps, where the environment is represented as a graph of interconnected places, as shown in the example of figure 1.3 (Mataric 1991; Kortenkamp & Weymouth 1994).

While metric maps enable very precise positioning by the robot, topological maps have, by nature of their compactness, the potential for representing environments which are several orders of magnitude larger than those which can be tractably navigated using metric maps. In my opinion, metric maps are therefore better suited to small-scale navigation tasks by mobile robots. Grid-based maps in particular are useful for tasks such as homing where the target location lies within the robot's immediate sensory range (see for example, Yamauchi & Beer (1996)).

This thesis, however, is concerned with the question of navigation in *middle-scale* environments. For mobile robots, middle-scale navigation means leaving the robot lab and entering unmodified public areas, such as corridors and offices, which are subject to unpredictable variations. A fundamental problem for navigation in middle-scale environments is *perceptual aliasing*, which refers to the situation where several places are perceptually similar enough to be confused by

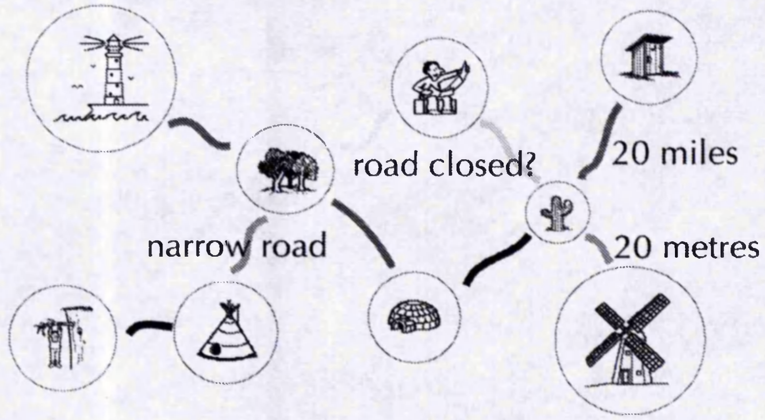

Figure 1.3: Example of a qualitative topological map, taken from Zimmer (1996). The environment is represented as a connected set of places, each place being identified by some perceived environmental feature or "landmark". Extra information such as approximate distances may also be added describing how to move between the places.

the robot. While many robot navigation systems have been tested in small-scale, laboratory environments (see Borenstein *et al.* (1996) for a detailed review), relatively few systems have been validated in middle-scale environments that have not been altered for the purpose of robot navigation (see Kortenkamp *et al.* (1998) for some examples).

Finally, to complete the analogy, large-scale navigation is used here to refer to robots which must leave indoor environments and navigate over much larger distances, e.g., farm robots. For these robots, global positioning systems (GPS), which use transmitted information from earth-orbitting satellites, might provide a suitable mechanism for self-localisation (Borenstein *et al.* 1996). GPS would be unsuitable for use in experiments described in this thesis, however, because it cannot be used indoors, and the dependence on external agents (satellites) would reduce the autonomy of the robot. In the absence of external assistance, a robot must rely on its own perceptions and internal representations to navigate.

This last aspect raises a much more difficult philosophical issue, namely the nature of "autonomy", which can be defined as "being capable of existing independently without outside control" (Merriam-Webster 1999). In my opinion, the term has been widely misused within the mobile robotics community, and has lost much of its original meaning as a result. For example, various authors have claimed that their navigating robots are autonomous, when the environment model used by the robot was actually pre-installed by the system designer, or the robot had to be guided around the environment by a human operator in order to acquire its own map. Others have used the term to refer to robots which require modifications to their environment, for example, using specially placed beacons for self-localisation. At the other extreme, some researchers have argued that an agent cannot be considered to be autonomous if its behaviour is dictated in some way by the experimenter (see e.g., Steels (1993)).

Perhaps the most important issue concerning the true autonomy of mobile robots is the question of knowledge acquisition by the robot. A distinction is drawn here between *pre-installation* of world knowledge by the system designer and *self-acquisition* of this knowledge by the robot, particularly with respect to the world models required for navigation. World models include not only environment and location models, but also implicit models of robot-environment interaction introduced by the system designer. For example, if the roboticist pre-installs hand-crafted rules for obstacle avoidance or wall-following, then an implicit model of robot-obstacle interaction or robot-wall interaction has been introduced. Similarly, if the roboticist writes a special algorithm to detect whether a door is open or closed, then an implicit model of robot-door interaction has been introduced.

Machine learning techniques such as neural networks present an ideal alternative to pre-installation. In this thesis, all of the environment models, location models, feature models and sensor-motor competences required for navigation are acquired independently by the robot. The learning techniques include self-organisation, where no teaching signal is required, and self-supervised learning, where techniques from supervised learning are used but all of the training examples are generated by the robot itself through trial and error. In particular, the robot is able to construct its own maps without requiring any pre-installed knowledge of the environment. As a consequence, human intervention is only required in this system to specify a goal location; all environment learning prior to human-robot interaction is achieved by the robot itself. The term *self-navigating* is therefore used to refer to this aspect of the system, namely the fact that the robot is entirely responsible for its own operation unless otherwise instructed.

A further contribution of this thesis is the development of quantitative methods for evaluating navigation performance. One of the main problems faced by any scientist when confronting the robotics literature is the lack of objective criteria for comparing different approaches. Unfortunately, robotics research so far has been dominated by qualitative descriptions of robot behaviour. Quantitative comparison of different systems is largely impossible.

In biology, replication and comparison of results for navigation experiments involving rodents has been made possible through widespread use of the Morris watermaze (Morris *et al.* 1982). This is a standard experimental set-up in which the animal has to find a submerged platform in a bowl of opaque water. One approach to achieving objective comparison of results in mobile robotics would be to devise a set of benchmarks for mobile robots, i.e., standard tests in standardized environments with task-specific performance measures, e.g., time taken to reach the goal, distance travelled, etc. However, the problem with benchmarks for mobile robotics is that the tasks and environments under investigation vary widely, and a set of tests for one application would be of little use to another. For example, a benchmark test for an office delivery robot would not apply to a sewer robot. Experience in the computing industry has also shown that where industry-wide benchmarks exist, designers tend to heavily engineer systems to perform well on these tests rather than in general, real world situations.

The alternative to benchmarks and application-specific measures of performance is to use general performance measures, such as localisation quality and map quality, which are applicable to many different robots, tasks and environments. In the same way that ethologists investigate the behaviour of animals in their natural habitat, the aim is to study navigating robots in their target environments. For example, the quantitative measure of localisation performance described in chapter 5 allows different self-localisation mechanisms to be compared in a variety of middle-scale, real world environments. From a scientific perspective, quantitative performance measures help us to generate better explanations of complex behaviour by mobile robots, and increase our understanding of the mechanisms required for truly autonomous operation. From an engineering perspective, this understanding helps us to make better robots!

1.4 The Method

Concurrent map building and self-localisation presents the robot with a "chicken and egg" problem — map building requires the ability to self-localise, but self-localisation requires a map. Some means of "pulling yourself up by the bootstraps" is needed to resolve this dilemma. In this thesis, the sub-problem of self-localisation is tackled first, using a pre-installed map provided by the system designer, then the sub-problem of autonomous map building is addressed once a successful self-localisation system has been produced. Finally, system integration and validation experiments are conducted to assess the performance of the complete system.

All of the results presented were obtained using a real robot operating in a series of unmodified, real world environments. In the intermediate stages of the research, the real sensor data of the robot was sometimes recorded and then played back in later experiments, allowing different mechanisms to be evaluated under the same experimental conditions. Because current simulator technology uses simplified numerical models which cannot capture the true complexity of robot-environment interaction (Lee *et al.* 1998), simulation was only used for initial prototyping of algorithms. At the present time, only experiments conducted on a real robot can answer the questions asked in this thesis. Even if a much more accurate simulator technology was to become available, there is always the danger that "unless you saddle yourself with all the problems of making a concrete agent take care of itself in the real world, you will tend to overlook, underestimate, or misconstrue the deepest problems of design" (Dennett 1998, p. 166).

1.4.1 The Robot, Task and Environment

The experiments were conducted using a Nomad 200 robot, which is shown in figure 1.4. This robot is equipped with coarse 360 degree range-finder sensing, consisting of sixteen ultrasonic sensors (range up to 6.5 m) and sixteen infrared sensors (range up to 60 cm). Both sets of sensors are spaced at 22.5 degree intervals around the robot's turret, which can be rotated independently relative to the base of the robot using an electric motor. Two other motors located in the base of the robot were used to control the translational and rotational movement of the robot. A flux-gate compass provided the robot with an approximate estimate of the orientation of its turret. The camera shown was not used in this thesis.

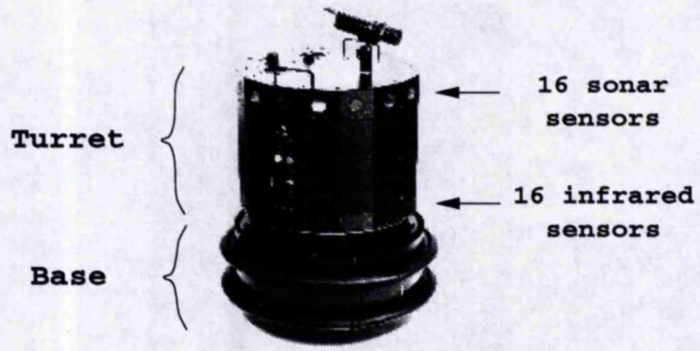

Figure 1.4: The Nomad 200 mobile robot *FortyTwo*.

For fully autonomous operation, no tether or radio link to external processors was allowed, and all computation was carried on board the robot's 486 processor. The only human intervention permitted during the final experiments was to recharge the robot's batteries, switch the robot and its navigation system on and off, and to specify a goal location.

To validate the research carried out, an office delivery task was considered, in which the robot had to find a series of arbitrary routes through an untreated, middle-scale environment. The environments used for the experiments throughout the thesis were assumed to be unmodified, initially unknown to the robot and *semi-structured* — that is, they may contain many unpredictable variations such as moving people, doors opening and closing, etc., but the basic structural elements, such as walls and corridors remain constant with respect to time. Therefore, all environment learning was carried out in a single exploration phase. A possible mechanism for dealing with environments which are subject to structural changes is suggested as a topic for future research (section 11.2.3).

1.4.2 Self-Localisation

To operate without human intervention, any mobile robot must depend only on its own sensory perceptions for location recognition. The space of possible perceptions available to the robot may be divided into two categories:

1. *Exteroception.* The robot's current perceptions of the outside world (external stimuli). A robot's exteroceptors may include range-finding sensors, tactile sensors, video cameras, etc.

2. *Proprioception.* The robot's perceptions of its own body movements (internal stimuli). In particular, odometry refers to the proprioceptor mechanism used for dead reckoning in mobile robots.

A localisation system based solely on proprioception would be unsuitable for two reasons. Firstly, no *a priori* information may be available to the robot when trying to self-localise. For example, if the robot becomes lost, then it will not be possible to initialise any form of dead reckoning. Secondly, any proprioceptive sensor system will be subject to cumulative drift errors, e.g., due to wheel slippage, which cannot be compensated through proprioception alone. The further the robot travels, the more inaccurate the position estimates produced by dead reckoning will become.

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 or "landmarks". In the navigation system developed, the robot *FortyTwo* uses a graph-based representation of its environment, where the nodes correspond to places and the links to possible transitions between places. Each place is identified by a description of the robot's sensory information at that place — it is this description or place "signature" which is referred to as a *landmark* in the rest of the thesis.

Measurement of Localisation Performance

One important aim of this thesis was to establish an objective set of criteria for assessing localisation performance. Experimental procedures and quantitative performance measures were developed, which permitted different algorithms for landmark identification and self-localisation to be compared under identical experimental conditions. Two important problems were addressed:

- Measurement of localisation quality.
- Tracking the robot's true location.

To assess localisation quality, an entropy-based statistic was used, which measures the extent to which the particular mechanism under investigation predicts the robot's true location (see chapter 5 for full details). In order to calculate this statistic, some means of recording the true location of the robot was required. In initial experiments, this was measured by hand, but this process was time consuming and prone to human error. A novel mechanism for measuring the robot's actual location was therefore developed which is based on retrospectively corrected odometer data. The approach has the advantage that no interpretation of the "correct" response by the robot is required, and no optimum standard has to be established by the observer.

Comparative Study of Landmark Recognition Systems

The next stage of the research, described in chapter 6, involved examining existing approaches to landmark recognition in the robotics literature, in order to find an appropriate mechanism for the Nomad 200 robot. Two different types of landmarks were found:

- Designer-determined landmarks.
- Robot-determined landmarks.

In the first approach, the categories assigned to environmental features are determined *a priori* by the system designer; for example, features such as ceiling lights, doors and junctions might be used. Problems with this approach are that the designer might not select the most appropriate landmarks for robot navigation, due to the different sensors of robots and humans, and that it can only be used in environments which contain the appropriate objects (the general make-up of the environment must be known in advance by the designer). Designer-determined landmarks were not used in this thesis because one of the main aims was to avoid pre-installation of world knowledge by the system designer.

Instead, systems which determine their *own* landmarks were considered. These included statistical clustering techniques, self-organising neural networks and occupancy grids. In these approaches, the representation schema is determined in advance by the system designer, but the robot is able to represent its own, *arbitrary* sensor patterns by filling in the details. The various approaches were evaluated in a number of different environments, using quantitative performance measures which took into account both localisation quality and computational efficiency.

Novel Self-Localisation System

The results of the comparative study were then used to guide the development of a novel self-localisation system. A key problem for mobile robots operating in middle-scale environments is that of *perceptual aliasing*, where several different places in the environment may share the same perceptual signature. Therefore, currently observable landmarks alone may not be sufficient to uniquely identify the robot's true location. Landmark misclassification can also be caused by sensor noise and the movements of other inhabitants of the environment. To overcome these problems, a self-localisation algorithm was developed which accumulates sensory evidence over time to identify places. As a result, the robot should be able to relocalise even after becoming completely lost in environments containing no guarantee of unique perceptual cues.

The topic of mobile robot self-localisation is usually divided into the sub-problems of *global localisation*, which means being able to relocalise under global uncertainty (an example is the "lost robot problem", where the robot has no prior information about its location), and *position tracking*, which means being able to accurately determine the robot's position once its general location is known[1]. While individual solutions exist for each of the problems, few robots can deal efficiently with both at the same time. In chapter 7, a unified and computationally efficient solution to the two problems is presented which is suitable for robot navigation in complex, middle-scale environments.

1.4.3 Map Building

The next part of the thesis addresses the question of how to explore and build maps of an unknown environment. In previous research, many systems have relied on maps which are pre-installed by the system designer (Stevens *et al.* 1995), or use passive mechanisms to build maps while the robot is steered manually around the environment by a human operator (Kortenkamp & Weymouth 1994; Engelson 1994). In other systems, the sensor-motor data required for map learning is first collected by the robot under manual control, then an off-line learning algorithm is used to find the best map to fit the data (Shatkay & Kaelbling 1997; Thrun *et al.* 1998b).

While both of these methods have their merits, manual intervention is by nature costly and prone to human error. Similarly, reactive behaviours such as wall-following, though often very robust, cannot be guaranteed to build complete

[1]Note that the terms "global localisation" and "position tracking" are used here to refer to mobile robot self-localisation *within previously mapped territory*; it would be unrealistic, for example, to expect the robot to solve the lost robot problem in completely unknown territory.

maps in large, complex environments. The most flexible approach is for the robot itself to acquire its own maps through a process of autonomous, map-based exploration. In other words, the robot should be able to identify regions of unexplored territory, navigate to the identified areas using its own map, and update its representation of the environment incrementally at the same time.

A version of this strategy was used here, described in chapter 9, in which the robot continuously tries to expand the territory which has already been charted. The robot attempts to recognise areas of open space in the environment which it has not visited before. Places which are presumed to exist, but have not yet been visited by the robot are added tentatively to the robot's map. Subsequent motion by the robot is used to verify whether these predicted places actually exist or not. The process is repeated until the robot has built a complete map of the target environment.

Two key problems were addressed, namely how to detect areas of open space, and how to deal with errors in the robot's map resulting from miscalculations in the dead reckoning — both are explained in further detail as follows.

Learning a Model of Open Space

To explore an environment in the manner described, some mechanism is required to detect areas of unexplored territory. Individual range-finder readings are not well suited for this purpose because of sensor noise and occlusions caused by people. An alternative would be to write a special feature detection algorithm to determine whether an area of open space exists in front of the robot, but this would involve the pre-installation of another world model by the system designer. Instead, an artificial neural network was used to learn the concept of "open space", combining noisy information obtained from many sensor readings. All of the data required for training the network, including the ability to travel in a particular direction, was collected by the robot itself, thus avoiding the need for manually labelling the training examples with the desired output categories. In this approach, the training data is generated by trial and error, i.e., "move until you hit something".

Correction of Dead Reckoning Errors in the Map

A fundamental problem for robot map building is that of obtaining globally consistent metric information in the map. The robot can only produce consistent position estimates by dead reckoning over very short distances, due to the inevitable problem of odometry drift. Over longer distances, these drift errors accumulate and the position estimates quickly become unreliable.

A global coordinate system in the robot's map is desirable for a number of different purposes, for example, inferring possible areas of unexplored territory, self-localisation, inferring novel routes and human interpretation of the robot's map. In particular, because the new self-localisation algorithm uses local odometric information between observed landmarks to disambiguate similar looking places, it requires that the places in the robot's map are labelled with Cartesian coordinates.

An optimisation algorithm was developed for assigning geometrically consistent coordinates to the places in the robot's map using only local odometric information, i.e., the relative displacement of the robot between topologically connected places, described in chapter 8. This algorithm is self-organising, using only local information and local interactions to converge upon a stable solution. It is proved that the algorithm will alway converge to the same globally consistent solution given the same local metric information. Subsequent experiments with the real robot demonstrate that the maps produced are of sufficient quality to enable reliable self-localisation and navigation by the robot.

1.4.4 The Complete System

The final stage of the research involved combining the various mechanisms to produce a complete navigating mobile robot. A hybrid deliberative-reactive architecture was developed, as shown in figure 1.5. In this architecture, the high-level activities of path planning, map learning and self-localisation are carried out in the deliberative layer. Low-level motor control is achieved using a set of previously acquired sensor-motor competences in the reactive layer, such as obstacle avoidance and wall-following. The interface between the deliberative and reactive layers consists firstly of feature detectors, such as the landmark recognition mechanism developed in the work of self-localisation and the model of open space acquired by the robot in the work on exploration. In addition, a set of intermediate control routines are used to select an appropriate combination of behaviours in the reactive layer, depending on the required heading of the robot. Thus, exploration and way finding are achieved through a combination of distributed control modules in the three different layers. The resulting system is

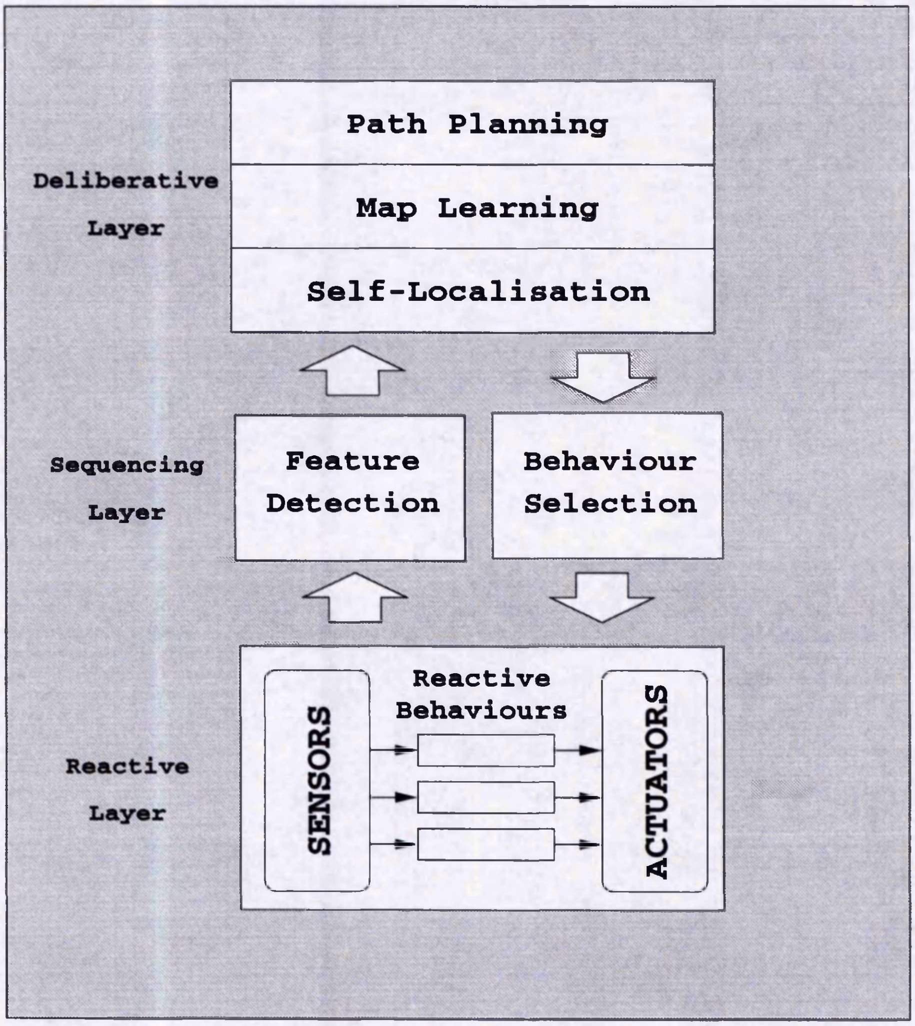

Figure 1.5: System architecture.

able to operate in complex, untreated environments of several hundred metres squared in real-time without requiring human assistance.

To validate the work carried out, an office delivery experiment was conducted in a busy university department building. Here, the robot was first allowed to explore the test environment fully, then had to navigate between a series of user-chosen locations. Reliability and efficiency-based measures were used to evaluate the overall navigation performance. To demonstrate the robustness of the system, the ability of the robot to recover from becoming lost was also considered.

The final results confirmed that the complete navigating robot is able to build its own maps through free exploration, determine its own location in the map and navigate reliably between user-specified locations. Furthermore, the robot can recover from error, and does not require any external processing or simplifying assumptions about the structure of environments.

1.4.5 Limitations

The navigation system presented in this thesis is specific to a holonomic mobile robot equipped with (1) a compass, (2) sonar and infrared range-finder sensors, and (3) odometry accurate enough to measure distances of up to 1 m with small errors (e.g., 5% of distance travelled on the Nomad 200). Nevertheless, most of the techniques presented should be applicable to most similar robots with only minor changes to take into account the different sensors.

The maps built by the robot — in common with all maps — will eventually become out of date. If the underlying structure of an environment changes, then the robot would need to build a new map. While this would not be a major drawback for the mobile robot presented in this thesis, as it can build its own maps, a better solution might be to allow the robot to continuously adapt its existing model of the environment during normal operation. A discussion of this problem of "lifelong learning" can be found in section 11.2.3.

The system as it stands only builds consistent maps in indoor, semi-structured environments which do not contain large cycles or "loops". Large cyclical environments present a seriously non-trivial challenge for current mobile robots. After traversing a large cycle, the robot needs to be able to recognise that it has returned to a previously visited location, otherwise it will mistakenly create multiple (inconsistent) representations in the map for the same physical location. A detailed discussion of this open question is given in section 11.2.4.

1.5 Contributions to Mobile Robotics

This thesis describes all of the necessary parts, elements and steps required to build a self-navigating mobile robot. All of the internal representations, sensor-motor competences and feature detection mechanisms required for navigation are acquired independently by the robot, without requiring human intervention. The specific contributions of this research include:

1. The development of quantitative methods for assessing navigational competence in mobile robots; in particular, measures of self-localisation performance and map quality were considered.

2. Replication and a comparative study of existing work on mobile robot self-localisation, including work by other researchers, using quantitative performance measures.

3. A new method of matching occupancy grids, which has a much lower computational cost than previous methods.

4. A solution to the lost robot problem.

5. A unified algorithm for solving the problems of global localisation and position tracking within previously mapped territory, which has been validated through its integration into a complete navigating mobile robot.

6. A relaxation algorithm for assigning geometrically consistent Cartesian coordinates to the places in a topological map using only local odometric information, which is proved to converge to a globally optimal solution.

7. A novel technique for exploring an unknown environment using an artificial neural network to learn the concept of open space, in which all of the training examples are generated by the robot itself through trial and error.

The ultimate contribution of the thesis is an untethered mobile robot which builds its own maps through free exploration, identifies locations, determines routes and navigates reliably between user-specified locations in unmodified, middle-scale environments, without requiring off-line processing. The thesis also represents a case study in "quantitative robotics"; that is, an application of quantitative performance measures to the design, testing and validation of a complete, self-navigating mobile robot.

1.6 Thesis Outline

The remainder of the thesis is organised as follows.

Chapter 2 reviews previous work on navigating mobile robots, including details of control architectures, world models and methods for map building and self-localisation.

Chapter 3 provides a detailed, high level analysis of the requirements of building a mobile robot that can navigate in middle-scale environments.

Chapter 4 briefly describes the basic mechanisms for sensing and low-level motor control and the assumptions used in the following chapters.

Chapter 5 describes an experimental procedure and a general, quantitative performance measure for evaluating self-localisation performance.

Chapter 6 presents an experimental comparison of previous algorithms for landmark recognition using quantitative performance measures.

Chapter 7 describes the complete system developed for self-localisation, including global localisation, and its quantitative analysis.

Chapter 8 describes the algorithms used for map learning by the robot, including an optimisation algorithm for maintaining geometric consistency.

Chapter 9 presents an exploration system for mapping an unknown environment, in which the environment model is acquired incrementally by the robot.

Chapter 10 describes the integration of the mechanisms for self-localisation and map building into a complete navigation system, and validation experiments involving an office delivery task.

Chapter 11 provides a summary, open questions and conclusions.

Chapter 2

Review of Navigating Mobile Robots

About this chapter. This chapter reviews related work on robot navigation. The motivation for concentrating on complete robots is first discussed, followed by details of common control architectures and world models used for navigation. Some representative examples of navigating robots are then described, particularly with respect to the mechanisms used for map building and self-localisation..

2.1 Introduction

2.1.1 Why Study Complete Robots?

Dennett (1998, p. 308) listed five simplification strategies which are commonly used by AI researchers:

1. Ignore learning and development; model the mature competence, postponing questions of how it could arise.

2. Isolate a subcomponent, ignoring problems of how it might be attached to the larger system.

3. Limit the domain to a toy problem, hoping that scaling up will be straightforward.

4. Bridge the gaps in one's model with unrealistic or miraculous stopgaps.

5. Avoid the complexities of real-time, real world coordination by ignoring robotics.

The problem with all simplifications is in deciding which factors are important, and which factors to leave out. In studying a subcomponent such as map building or self-localisation in isolation (item 2), it is possible to avoid the real problems that will affect the whole system. "Bridging the gaps" (item 4) amounts to the same thing; by assuming a pre-installed map or guiding the robot around an environment by hand, the designer may miss some other requirement for truly autonomous operation. Similarly, work based on simulations or idealised assumptions is unlikely to reveal much about real robots (item 5), and systems which are only tested in specially constructed laboratory environments are in danger of solving only specially constructed problems (item 3).

Consequently, this review concentrates on research which has been validated through developing complete navigating robots operating in middle-scale environments. The robots described in this chapter were selected for their particular relevance to the problem of concurrent map building and self-localisation.

Perhaps the most common simplification made in mobile robotics is to ignore learning and development (item 1). This is not particularly surprising, given that most roboticists are more interested in building robots that work than in producing realistic cognitive models. However, there is a very good reason for studying robots which learn. It is common practice in mobile robotics to handcraft the necessary behaviours, feature detectors, etc., required for a particular application. If the robot is then transferred to a new environment, the preinstalled competences may fail. By contrast, a learning robot need not be given all of the details of its environment by the system designer, and its sensors and actuators need not be finely tuned (Dorigo & Columbetti 1998). Learning offers the ability to *adapt* to new situations — this issue is discussed in further detail in the following sections.

2.1.2 Robot Control Architectures

Most mobile robots can be classified according to one of three possible control architectures:

1. The Functional Approach.

2. Behaviour-Based Control.

3. Hybrid Architectures.

The first approach is based on a series of functional modules. *Sensing* involves transforming incoming sensor data into a central world model, *planning* involves using the model to predict an appropriate set of actions, and *acting* involves executing the plan by controlling the robot's actuators. However, this approach can be brittle (failure in one module leads to failure in the whole system) and slow to react in dynamic environments. Perhaps its most serious drawback is the well-known *frame problem*, that is, the inability of the world model to predict all of the changes which can occur in the real world.

One response to the shortcomings of the functional approach is to eliminate the central world model and the planning module completely, instead decomposing the robot controller into a set of parallel, task-achieving behaviours — see in particular the subsumption architecture proposed by Brooks (1986). Each behaviour consists of a tightly coupled link between sensing and acting, with only minimal communication between behaviours. However, this approach has its own disadvantages; while robust and quick to react to real world situations, it is limited in its ability to carry out the kinds of goal-directed activity required for many applications, e.g., delivery, inspection, etc.

For these reasons, many researchers agree that a hybrid approach presents the best option, combining the high-level, model-based abilities of the functional approach with the low-level, sensor-motor capabilities of behaviour-based control. All of the robots reviewed in the rest of this chapter use, to a greater or lesser extent, a combination of explicit world models and reactive behaviours. While relatively few hybrid systems so far have a well-defined architecture (see chapter 6 of Arkin (1998) for some exceptions), a useful decomposition is provided by the following three-layer description, after Gat (1998):

1. Deliberative Layer.

2. Sequencing Layer.

3. Reactive Layer.

The deliberative layer is responsible for high-level activities such as planning, the reactive layer for low-level sensor-motor control, and the sequencing layer describes the interface between the deliberative and reactive layers.

2.2 Models Used By Navigating Robots

The world models used by a robot may take many forms. Some may be explicit, others may be implicit in the robot's control software. Zimmer (1997) distinguished general models such as behaviours, where "decisions are based on local, immediate sensing", from specific models such as maps, which contain information about "time, place or state". In order to make clear exactly what constitutes a world model in this thesis, I devised the following taxonomy:

1. *Environment Models.* The models such as maps used to represent the robot's environment.

2. *Location Models.* The models such as probability distributions used to represent the robot's location within an environment model.

3. *Behaviour Models.* The models used to implement primary sensor-motor skills such as obstacle avoidance, by translating sensor information directly onto motor actions.

4. *Feature Models.* The models used to translate sensor or motor information into internal concepts (percepts) such as landmarks.

5. *Sensor Models.* The numerical models used to approximate the true function of a robot's sensors.

6. *Motion Models.* The numerical models used to approximate the true function of a robot's actuators.

7. *Noise Models.* The numerical models (often implicit) used to represent the noise in the robot's sensors and actuators.

Note that these categories are intended neither to be exhaustive nor mutually exclusive. Some of these models may be composed of other lower level models. For example, an environment model may typically be defined in terms of feature models. In this thesis, standard techniques are used to provide the sensor, motion and noise models for the navigating robot. However, in contrast to previous work, all of the environment, location, behaviour and feature models required for navigation are acquired independently by the robot, instead of being pre-installed by the system designer. The possible models considered are discussed as follows.

2.2.1 Environment Models

The environment models used for robot navigation are often divided into two approaches, namely topological and metric maps.

A topological map records a set of recognisable locations and the traversable paths between the locations. The paths may be augmented with distance and angle information, and the locations are usually identified by observed environmental features or landmarks. Some approaches assume that the landmarks are assigned uniquely to individual places (Mataric 1991; Kortenkamp & Weymouth 1994), or use an exploration strategy to disambiguate similar looking places by taking into account the sequence of landmarks observed (Nehmzow *et al.* 1991; Kuipers & Byun 1991).

In a metric map, the locations of objects in the robot's environment are specified in a global coordinate system. Some approaches use a feature-based representation, where the map consists of a set of geometric primitives such as line segments (Leonard *et al.* 1990; MacKenzie & Dudek 1994). Other approaches use a grid-based representation, such as occupancy grids, where each grid cell contains some measure of the certainty that the corresponding area is occupied by any object. Possible representation schemes for occupancy grids include probabilistic models (Moravec & Elfes 1985), fuzzy logic (Oriolo *et al.* 1998) and Dempster-Shafer theory (Hughes & Murphy 1992; Pagac *et al.* 1996). Another alternative is to use quad trees (Zelinsky 1992), where the geometric space is decomposed recursively into smaller grid cells.

While many different proposals for robot maps exist in the literature, relatively few approaches have been applied successfully in complete, navigating robots. For example, the assumption of perceptually unique landmarks can never be guaranteed in practice, so systems based on this assumption are unlikely to succeed in practice. Systems which use feature-based representations tend to be brittle because they suffer from the *correspondence problem* of matching noisy sensor readings to the designer-determined primitives. Computational efficiency is another important consideration; metric maps require large amounts of memory and processing, and also depend critically on accurate position information for map building.

Some successful applications of robot map building are described later in this chapter. These include robots which use metric and topological maps, and also some approaches which attempt to integrate both topological and metric representations.

2.2.2 Location Models

The location model used by a robot depends closely on its environment model. The simplest approaches are unimodal — that is, the robot maintains a single estimate of its position within the map. In a topological map, this may be implemented by a winner-takes-all strategy, using a matching process between the current sensor data and the stored landmarks in the map to determine the most likely location of the robot.

In a metric map, the robot's location is typically represented by a Cartesian coordinate and updated on the basis of the positions of perceived environmental features in the map. A common technique for combining position estimates obtained from different observations over time is the Kalman filter (Gelb 1974; Maybeck 1990), where the uncertainty in the robot's location estimate is represented by a unimodal (typically Gaussian) probability density function. At each iteration, the filter is used to combine an existing position estimate updated by dead reckoning with a new position estimate obtained from sightings on known environmental features, according to the respective uncertainty in each of these measurements.

While unimodal approaches can produce very accurate position information, they also tend to be brittle, because they require an *a priori* position estimate in order to resolve perceptual ambiguity, i.e., to deal with situations where several locations in the map appear similar enough to be confused by the robot. This means that the robot cannot be guaranteed to recover its position if it becomes lost. A mobile robot operating in complex, real world environments is bound to become lost eventually, regardless of the accuracy of its sensors, especially in environments which are subject to unpredictable variations over time.

Multimodal approaches offer a more reliable alternative, being able to represent situations where the robot is uncertain of its true location, but has some idea of possible regions of the map in which it might be located. An example of this approach is used in Hidden Markov Models (Koenig *et al.* 1996), where the robot's location model consists of a probability distribution over a set of discrete states corresponding to possible robot locations and orientations. The Markov localisation approach has also been applied to high resolution grid-based maps, maintaining the probability distribution over the cells in the grid (Burgard *et al.*

1998b).

In this thesis, a multimodal location model is used which provides a generalisation on the Kalman filter. A set (or mixture) of competing location "hypotheses" is maintained, each with its own separate probability density function over Cartesian space, thereby combining the reliability of discrete probabilistic methods with the precision of the Kalman filter.

2.2.3 Feature Models

The feature models used by many robots are hand-crafted by the system designer to suit a particular application. Example features used by navigating robots include walls, doors and ceiling lights. However, the problem with pre-installation is that the feature detectors are unlikely to work in different environments. A good example is provided by Nourbakhsh (1998) in describing the 1994 AAAI National Robot Contest. To the surprise of many contestants, the walls of the test environment at the competition were constructed from smooth sheet plastic. This had the effect of confusing many of the sonar-based robots, "causing them to veer into walls and to pick up large numbers of false features".

The alternative to using hand-crafted features is for the robot to acquire its own feature models. One approach is to use supervised learning, such as artificial neural networks trained by back-propagation. For example, Hertzberg & Kirchner (1997) trained a multi-layer perceptron to recognise the different types of junctions found in sewage pipes. Similarly, Mahadevan *et al.* (1998) trained a network to learn the concepts of "door", "opening", "wall" and "undefined" in an indoor corridor environment.

While designer-determined feature categories may be ideal for one specific application in a known domain, they cannot be used for mapping new environments which may contain many unknown features. Learning mechanisms based on self-organisation present one possible alternative, including Kohonen networks (Kohonen 1993), ART networks (Carpenter & Grossberg 1987) and Growing Neural Gas networks (Fritzke 1995). Another approach is provided by self-detailing feature models, such as local occupancy grids (Yamauchi & Langley 1997), which are also able to represent arbitrary sensor patterns.

A further alternative investigated in this thesis is to use self-supervised learning, where all the training examples required to learn a particular classification function are discovered by the robot itself through trial and error. In the exploration mechanism presented in chapter 9, a self-acquired feature detector is used to recognise areas of open space in the robot's environment. This function was obtained by training a neural network to associate recorded sensor readings with the robot's own ability to travel in a given direction.

2.2.4 Behaviour Models

Pre-installed sensor-motor competences are also prone to human error and failure in unknown environments. For example, many robots use a set of hand-crafted rules to avoid obstacles. A common method is described by Nourbakhsh (1998), in which each range-finder sensor reading contributes to the forward and rotational velocities of the robot according to a weighted sum. However, the problem here is how to set the weights, particularly as the human designer "has no real intuition" on how to fine-tune the robot's behaviour.

An alternative method is to obtain the weight values by supervised or self-supervised learning. Nehmzow *et al.* (1991) developed a mechanism in which a single perceptron learns behaviours such as obstacle avoidance and wall-following in a small number training steps, by associating sensor patterns with required motor actions (figure 2.1). The teaching signal is provided by feedback from the robot's sensors, using a predetermined set of "instinct rules" to guide the acquisition of the sensor-motor behaviours. A further advantage of this approach is that the robot can adapt to changes in its morphology such as sensor failure by adjusting its behaviour during operation. Alternatively, the training signal can be provided directly by a human teacher via a joystick (Martin & Nehmzow 1995).

2.3 Robots with Metric Maps

In the rest of this chapter, a number of research robots are reviewed, discussing their relative merits with respect to the problem of concurrent map building and self-localisation. This first section covers systems which use metric maps, that is, environment models in which an explicit Cartesian reference frame is used to represent objects in the robot's environment. In particular, the two most common approaches are considered, namely feature-based and grid-based maps.

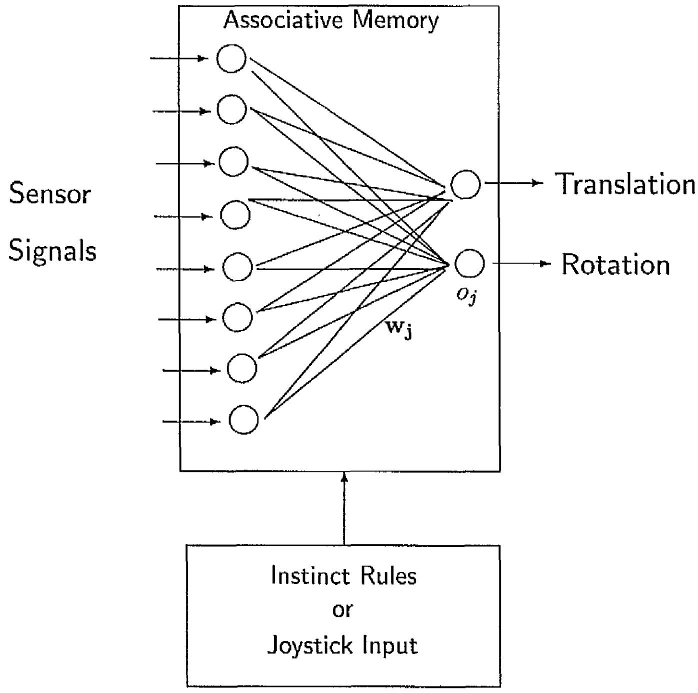

Figure 2.1: Pattern associator for learning behaviours, adapted from Martin & Nehmzow (1995). A perceptron is trained to associate sensor patterns with required motor actions. The training signal comes either from a predefined set of "instinct rules", which monitor and correct the behaviour of the robot, or from a human teacher by joystick input.

2.3.1 Feature-Based Maps

AGVs

Leonard & Durrant-Whyte (1992) investigated an approach for robot navigation using sonar sensors and a precise metric map, which was implemented on various Robosoft automatic guided vehicles (AGVs). The map was built from a set of predefined geometric features including planes, cylinders, corners and edges. These features were made up of primitives known as "regions of constant depth", consisting of sets of adjacent sonar returns of nearly the same range. Self-localisation was achieved while the robot was in motion by tracking stationary features in the environment and applying an extended Kalman filter to combine the inferred position estimates. An important contribution of this work was the development of an improved sonar sensor model for robot navigation.

One of the drawbacks of this approach is that very accurate position information is required for map building. The authors sketched a hypothetical framework for simultaneous map building and self-localisation, but the two mechanisms were only implemented separately, using a hand-measured map for localisation and precise *a priori* position information for map acquisition. These competences were also only tested in a small-scale, static environment, so it is unclear whether they would scale to more complex environments.

A fundamental weakness of the feature-based approach is that it suffers from the *correspondence problem* of matching the raw sensor readings to the human-defined feature categories. Systems which need to solve the correspondence problem tend to be brittle, particularly in dynamic environments where there may be many temporary occlusions and false sensor returns. Any incorrectly identified features would be added to the map, which in turn would result in positioning errors, leading to a mutually destabilising effect between map building and self-localisation during concurrent operation.

Cox & Leonard (1994) proposed a multiple hypothesis tracking technique to deal with the problems of map building in dynamic environments. In this approach, multiple environment models are maintained at each time step according to different possible interpretations of the robot's sensor data. Each model is associated with a probability reflecting its likelihood of being the "correct" model of the environment. However, this technique was only tested in a small-scale environment, and it is questionable whether such an approach would be tractable in a complex, real world setting.

ARNE

David Lee (1995) implemented a map building system for a mobile robot known as *ARNE*, which was equipped with a single rotating sonar sensor. This system also used a feature-based metric map and an extended Kalman filter for self-localisation, and was tested in a static, laboratory environment. A separate grid-based map was derived from the feature-based map for the purpose of path planning. Again, this system relied upon precise position information and accurate feature detection, so it would be unlikely to work in real world, middle-scale environments.

The principal contribution of this work was its quantitative, experimental evaluation of various exploration strategies used by the robot to build its maps (Lee

& Recce 1997). In this analysis, the performance of the automated exploration strategies was compared to that of an optimal exploration behaviour determined by a human observer. A comparative measure of the quality of the grid-based maps generated by the robot was used to assess performance. The authors found that the best results were achieved by a hybrid model-based/reactive exploration strategy, consisting of wall-following with some map-based interventions according to a predefined set of heuristics.

2.3.2 Grid-Based Maps

Some of the most successful approaches to robot navigation have used the Cartesian occupancy grid representation developed by Moravec and Elfes (Moravec & Elfes 1985; Elfes 1987). Each cell in the grid model contains the probability that any object occupies the corresponding location in the robot's environment. The probabilities are obtained by using a pre-installed sensor model to project the robot's range-finder readings onto the corresponding grid cells, and applying a Bayesian update rule to combine multiple readings over the same cell. Self-localisation is achieved through correlation of a grid constructed from the current sensor readings with the stored map, finding the displacement and rotation which produces the best match between the two grids.

The approach avoids the correspondence problem because it does not need to identify the source of the robot's sensor returns. Occupancy grids also provide a natural representation for combining different sensor modalities, provided that a good model is available for each different type of sensor. For example, Thrun *et al.* (1998a) used an occupancy grid to fuse range information from stereo vision and sonar. The disadvantages of the approach are that it takes up a large amount of memory, requires precise position information for map building and depends on accurate range-finder sensing. For example, the specular effects associated with sonar sensors often result in geometric errors in the map.

ARIEL

Yamauchi *et al.* (1998) developed an autonomous robot known as *ARIEL* which explored and built its own map of an unknown environment (see figure 2.2). The system was implemented on a Nomad 200 robot equipped with a planar laser

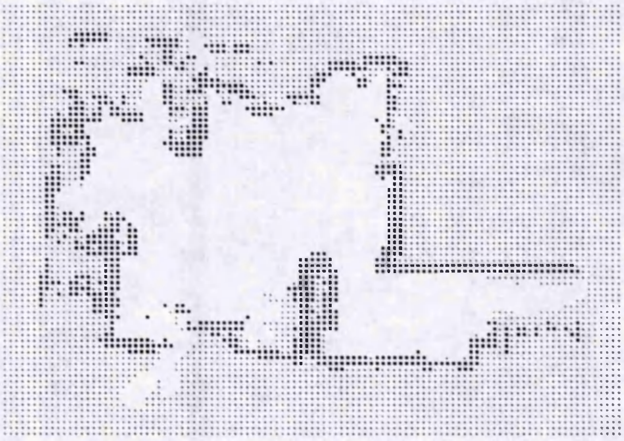

Figure 2.2: An example occupancy grid acquired by the *ARIEL* system, taken from Yamauchi *et al.* (1997).

range-finder in addition to its sonar and infrared sensors. The research integrated a map building strategy known as frontier-based exploration (Yamauchi 1997) with a continuous localisation technique for correcting the robot's odometry (Schultz & Adams 1998).

The system used a sensor scanning technique in which specular reflections affecting the robot's sonar sensors were corrected by the laser range-finder. A process analogous to edge detection and region extraction in computer vision was then used to detect regions between open and unexplored space in the global grid model known as "frontiers". During exploration, the robot attempted to navigate to the nearest frontier in the current map. When the robot reached this frontier, it carried out a new sensor scan and updated the map, adding any new frontiers detected to the list of unexplored goal locations.

Self-localisation was performed by matching a short term, local occupancy grid constructed from the robot's recent perceptions to the long term, global map. The matching process was restricted to a small space of possible translational and rotational errors in the global grid centred around the current position estimate produced by dead reckoning. Note that this dependence on prior position knowledge for self-localisation means that the system would be unable to recover from becoming lost.

ARIEL was tested successfully in a 21 m long corridor environment. A disadvantage of the approach is its high computational requirements, needing radio

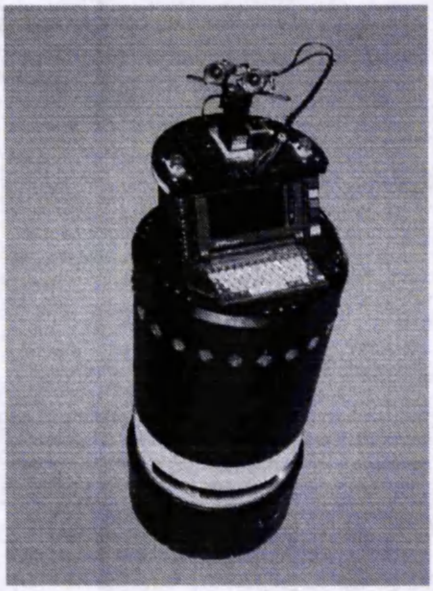

Figure 2.3: *RHINO*, a RWI-B21 robot at Bonn University.

communication with two external Sparcstation 20s. In addition, the approach relies on the accuracy of its range-finder sensors, so it would be unlikely to produce geometrically correct maps in dynamic environments where there may be many unpredictable variations in the robot's sensor data.

RHINO

RHINO is a RWI-B21 mobile robot equipped with a stereo camera system and a ring of 24 sonar sensors (figure 2.3), which has been used as a test-bed for many innovations and state-of-the-art techniques for navigation using metric maps (Thrun *et al.* 1998a; Burgard *et al.* 1998a). For example, later versions of the system have used Markov localisation (Burgard *et al.* 1998b) to avoid dependence on *a priori* position information from dead reckoning (see also section 2.4.2), and an "entropy filter" to detect the presence of humans within the robot's sensory range (Fox *et al.* 1998).

The system is able to build accurate metric maps using information from both its vision and sonar sensors. A neural network is used to translate the sonar readings into occupancy probabilities in the grid, combining information from neighbouring sensors to reduce specular effects and increase the geometric accuracy of the map. The vision system is able to recover depth information from vertical edges detected in the camera images, allowing the robot to detect objects which cannot be perceived using sonar.

Another interesting aspect is that the system can extract a topological map from the grid-based map, using critical points detected in a Voronoi diagram to partition the unoccupied space in the grid into a set of discrete locations (Thrun 1998b). The topological map is then used for path planning. Thrun claims that this approach allows the robot to exploit the "orthogonal strengths" of metric and topological maps, although the robot is still faced with all of the consistency problems and computational overheads involved in maintaining a full metric map.

RHINO has recently been tested extensively as a tour guide in a busy museum environment (Burgard *et al.* 1998a), though the map required for this task was constructed by the designers from accurate distance measurements taken by hand. This application incorporated other areas of AI research, including high-level problem solving and human-robot interaction. A major drawback of the system is its high computational requirements, needing two on-board processors and a radio link to several Sparcstations for normal operation. The system also depends on an assumption that walls are always parallel or perpendicular to each other, a constraint which, while frequently met, cannot be guaranteed in practice.

Work is currently in progress on a successor to *RHINO* called *MINERVA* (Thrun *et al.* 1999; Roy *et al.* 1999), implemented on a RWI-B18 robot, which is based on the same underlying principles and software architecture as *RHINO*. Its innovations, so far, include extra sensors — for example, an upwards-pointing camera is used to learn a map of the ceiling (Thrun *et al.* 1999), and "coastal navigation" (Roy *et al.* 1999), wherein the robot chooses motor actions designed to improve localisation quality.

2.4 Robots with Topological Maps

The problems with maintaining high-resolution geometric maps, namely the high computational requirements and the need for accurate position information, have lead many researchers to investigate the use of topological maps. In this approach, the environment is represented as a graph of connected places, and the problem of self-localisation becomes that of *place recognition* (Kortenkamp & Weymouth 1994). These systems have the advantage that the robot does not need to know its exact position for map building. Some pioneering work on navigation using topological maps was carried out by Kuipers & Byun (1991), introducing the notion of locally distinctive places, but this research was only carried out in simulation. Wan Yik Lee (1996) implemented Kuipers' "Spatial Semantic Hierarchy" on a real robot, but the system was only tested in a small-scale laboratory environment consisting of three cardboard corridors.

Toto

Mataric (1991) developed a mobile robot known as *Toto*, which used a topological map and explored its environment by wall-following. Sonar sensors and a compass were used to identify landmarks according to a designer-determined set of categories. Each distinctive landmark was represented by a node in the map, recording the links to the previous and next landmarks identified en route. During navigation, a path to a goal location was found by spreading activation from the destination node. A fundamental drawback of this approach is that it relies upon the correct identification of landmarks, so it is likely that the system would fail in the presence of sensor noise or perturbations in a real world environment. The approach is also only relevant to environments which can be explored by wall-following.

2.4.1 Self-Organising Robots

Alder and Cairngorm

Nehmzow (1992) developed a reactive controller for the Fischertechnik robots *Alder* and *Cairngorm* (figure 2.4), using wall-following and robot-determined landmarks to identify places. All of the world models used in this system were acquired autonomously by the robots. The sensor-motor competences were learned by a simple feedforward neural network, as described in section 2.2.4. A self-organising feature map (Kohonen 1993) was used for map building, allowing the robot to construct its own internal representation of the environment by clustering together similar groups of sensor readings. These mechanisms have also been validated on a Nomad 200 robot operating in untreated, middle-scale environments, for example, using a self-organising feature map for route-learning (Owen & Nehmzow 1997) and wall-following for relocalisation after becoming lost (Duckett & Nehmzow 1998).

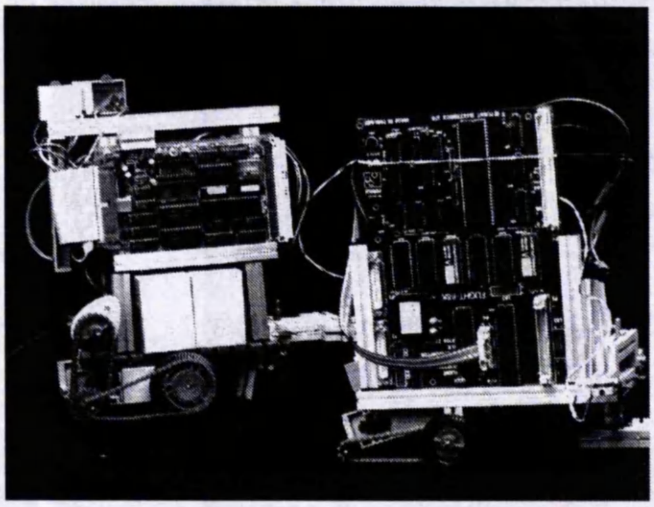

Figure 2.4: *Alder and Cairngorm*, Fischertechnik robots at Edinburgh University.

This research addressed the problem of perceptual aliasing by incorporating history of the robot's sensor-motor experience into the recognition of locations, adding previous sensor readings and motor commands to the input vector of the feature map. A disadvantage of this approach is that location recognition depends on visiting locations in the same sequence as used for map building, therefore restricting the behaviour of the robot to following fixed paths. In addition, location recognition is limited by the duration of the stored history, and does not generalise to cover arbitrarily large areas of perceptual ambiguity such as long, featureless corridors. The self-organising feature map also requires a settlement period to achieve a stable clustering of its input space, and has a number of critical parameters such as the network size which have to be determined in advance by the designer.

ALEF

Kurz (1996) developed a topological mapping system for a modified RWI-B12 robot known as *ALEF* (figure 2.5). In this system, classification algorithms such as self-organising feature maps were used to group together similar sets of sonar readings. The resulting feature categories were then used to partition the robot's environment into contiguous regions known as "situation areas" (see figure 2.6).

During map building, a graph-based representation of the environment was constructed, recording both the topological relations between the situation areas

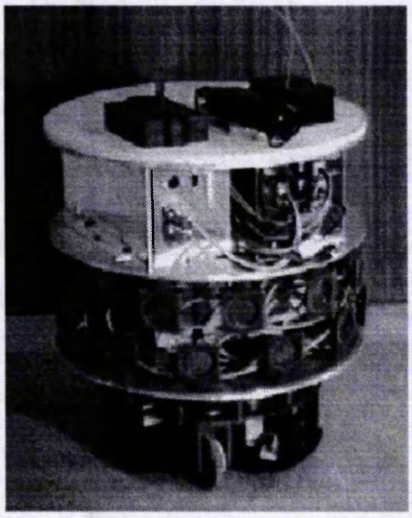

Figure 2.5: *ALEF*, a RWI-B12 robot at Darmstadt University.

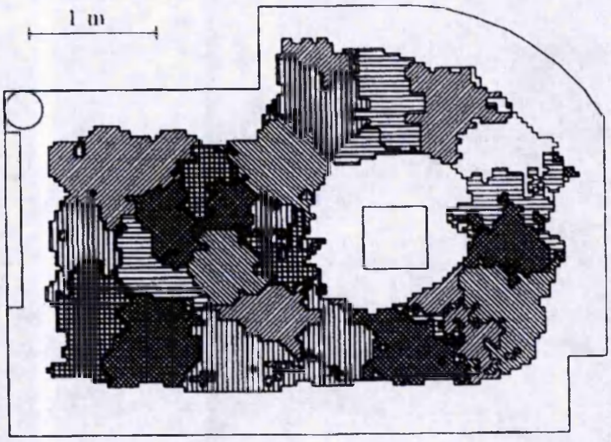

Figure 2.6: Representation of an environment using *situation areas*, taken from Kurz (1996). A self-organising classifier is first used to cluster the robot's sonar readings into prototypical "situations"; the resulting classifications are then used to partition the explored territory into regions.

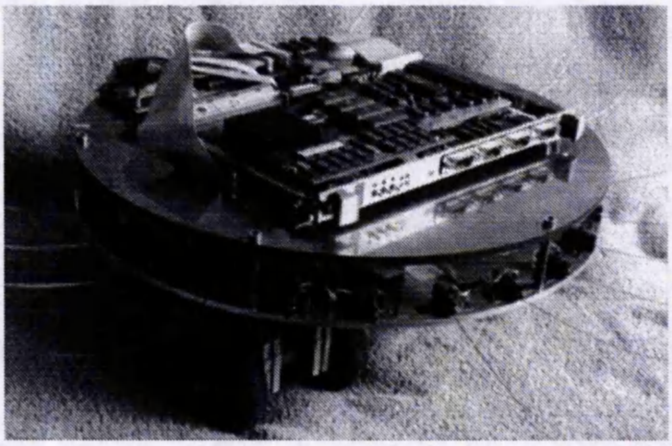

Figure 2.7: *ALICE*, a custom-built robot at Kaiserslautern University.

and position information obtained from the robot's odometry. In this approach, the coordinates produced by dead reckoning in a particular situation area were averaged to produce a single coordinate for each node in the graph.

The robot was also able to navigate to arbitrary locations in the map using the $A*$ algorithm (Nilsson 1980) for path planning. Self-localisation was achieved by a Kalman filter, using the coordinates of the observed situation areas to correct the odometry drift errors. A disadvantage of this approach is that the robot has to rely on dead reckoning when exploring unknown areas, thus limiting the size of the environments which can be mapped according to the accuracy and reliability of the robot's odometry. Furthermore, the system contains no mechanism for recovery after becoming lost, so localisation errors caused by perturbations or perceptual aliasing could lead to failure of the whole system.

ALICE

Zimmer (1995a) developed a concurrent map building and self-localisation system for a custom-built robot known as *ALICE* (figure 2.7). Although this system was only tested in a small-scale laboratory environment, the robot was deliberately designed to use only low resolution, low reliability sensors, thereby allowing the designer to investigate some of the fundamental problems which affect any mobile robot operating in an unknown environment. *ALICE* was equipped with a ring of passive light sensors and touch-sensitive whiskers for location recognition, and a basic dead reckoning mechanism affected by drift errors of up to 25%. In contrast to many other robots, this system had no separate phases for map learning and navigation, instead being able to continuously adapt its internal representations through a process of lifelong learning.

For map building, *ALICE* used an extension of the Growing Neural Gas network (Fritzke 1995), consisting of a set of stored sensor prototypes augmented with Cartesian coordinates and the topological connections between them. The input vector to the network was constructed from the light sensors, tactile sensors and odometer readings. A new node was added to the map whenever a distance measure of the similarity between the current input and the nearest matching prototype exceeded a predefined threshold. At the same time, the robot's odometry was continuously recalibrated through correlation of the current sensor readings with the map. Exploration was carried out using a reactive controller, which was subject to top-down influence from various "instincts", such as trying to reach areas of unexplored territory, or trying to improve localisation quality by moving through areas of previously charted territory.

One of the most important contributions of this research was the handling of the concurrent updates to the robot's environment and location models. In particular, it was found that simultaneous updates led to instability in the world models, because each representation was updated directly with the errors and noise from the other (Zimmer 1995b). The solution found to this problem was to adapt the robot's environment model more slowly than the location model, delaying the integration of the current sensor information into the map by a predetermined time interval. In addition, the robot's map was adapted by gradient descent, moving the stored prototypes in the direction of the sensory input, rather than by a single step.

2.4.2 Hidden Markov Models

So far, all of the navigating robots described have used a unimodal location model, in which self-localisation is achieved either by a winner-takes-all mechanism or through correction of an *a priori* Cartesian position estimate. The problem with these approaches is that localisation errors such as those caused by perceptual aliasing are often fatal, because the robot cannot recover from losing its position. To overcome these problems, many researchers have investigated systems with multimodal location models, such as Hidden Markov Models (HMMs) and their extension to Partially Observable Markov Decision Process models (POMDPs)

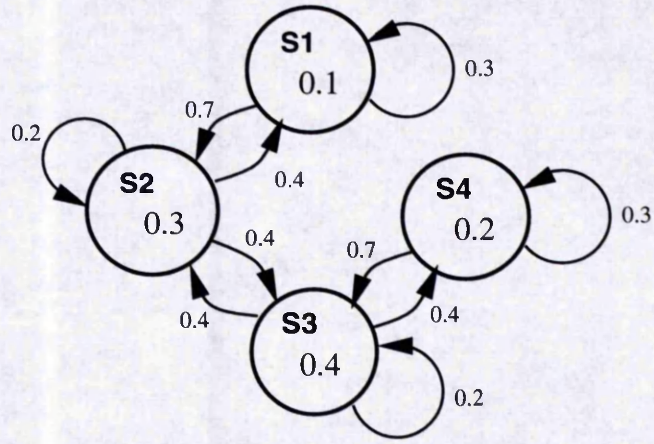

Figure 2.8: Example of a Hidden Markov Model with four states $S1$ to $S4$. Each state contains a probability of being the "true" state of the system. The arcs are labelled according to a probability transition matrix, which describes the likelihood of moving from one state to another. Another probability distribution, not shown, associates possible observations with each of the states. (Adapted from a figure by Magnus Rattray.)

(Simmons & Koenig 1995; Cassandra *et al.* 1996; Hertzberg & Kirchner 1997). Using this approach, the robot can represent the global uncertainty in its true location. For example, the robot might be "almost certain that it is in the North-East corner of either the fourth or seventh floors, though it admits a chance that it is on the fifth floor as well" (Kaelbling *et al.* 1998).

An HMM is a stochastic variant of a finite state automaton, in which possible robot locations and orientations are represented by states, and the robot's estimated location is represented by a probability distribution over the states, as the true state of the system is "hidden" or unknown. The robot's ability to move between states is defined by a probability matrix over all of the possible state transitions (see figure 2.8). Observable environmental features are associated with the states according to another probability distribution. The POMDP model extends the HMM by incorporating information about possible actions in each state, and the rewards for taking each action in each state. Key problems addressed by research into HMMs and POMDPs for robot navigation include how to obtain the models and how to estimate the true state using only the robot's sensory perceptions.

Dervish

Nourbakhsh (1998) implemented a navigation system on a customised Nomad 100 robot known as *Dervish*, in which hand-crafted feature detectors were used to identify landmarks such as doors or junctions. These routines were not particularly reliable, for example, the closed door detection algorithm only gave the correct classification 70% of the time. However, this low-level inaccuracy was overcome by using a high-level self-localisation algorithm known as state set progression.

In this approach, the robot's location model consisted of a "state set", containing a subset of the possible locations in a pre-installed topological map. The set was initialised from the features detected in the current sensor data, as determined by the landmarks associated with the locations in the map. Progression of a state consisted of removing it from the set and replacing it with all of the possible subsequent states, depending on the new sensor data and the robot's direction of travel.

A probability value was assigned to each state using a hand-crafted "certainty matrix", which represented the likelihood of obtaining the observed features from the actual landmarks in the environment. These values were then propagated in Bayesian fashion during state set progression. *Dervish* used the most likely state to plan a path to a goal location, stopping to replan if the robot was no longer on the intended path, for example, because the robot relocalised itelf to a different location in the map or overshot a turn into another corridor.

This approach was not used in this thesis because it requires a pre-installed topological map, and the feature models and certainty matrix have to be handcrafted to suit a particular environment.

Xavier

Koenig & Simmons (1996; 1998) developed a navigation system for performing delivery tasks in an office environment, which was implemented on a RWI-B24 robot known as *Xavier*. In this approach, the environment and location models were defined more formally as a Partially Observable Markov Decision Process (POMDP), though the underlying mechanisms for feature detection and self-localisation were much the same as those used by *Dervish*.

Xavier also required a pre-installed topological map, but had some ability to adapt its environment model through on-line learning. The user-defined map was augmented with approximate information about the distances between locations and the robot's sensors and actuators. The pre-installed models were translated into the POMDP representation by a specially written compiler. An extended version of the Baum-Welch algorithm (Rabiner 1989) was then applied during navigation to improve the compiled distance, sensor and actuator models by adjusting the corresponding probabilities in the POMDP model.

The Baum-Welch algorithm is an expectation maximisation (EM) method for acquiring HMMs and POMDPs from data. This algorithm has a number of drawbacks, for example, it requires a large amount of data, is slow to converge and is subject to local maxima in the likelihood space. Koenig and Simmons avoided the problem of local maxima by pre-installing the initial model, and made learning in real-time possible by restricting the amount of training data to a sliding "time window" (Koenig & Simmons 1996). The size of this window could be varied according to the computational resources available. The algorithm was also speeded up by enforcing a number of constraints in the model based on the designers' world knowledge.

The main disadvantage of this approach is its dependence on pre-installed world knowledge. *Xavier* relied on several assumptions about indoor environments which cannot be guaranteed in practice, for example, that corridors are always straight and perpendicular to each other. The compiled POMDP model also used a somewhat *ad hoc* representation for areas of open space.

Ramona

Shatkay & Kaelbling (1997) extended the Baum-Welch algorithm to allow off-line acquisition of the map using pre-recorded sensor data collected by a modified RWI-B21 robot known as *Ramona*. This was achieved by extending the basic Hidden Markov Model to incorporate odometric information, and adapting the expectation maximisation algorithm to maintain geometric consistency in the model. A special clustering algorithm was developed to provide the initial model, based on local odometric relations extracted from the recorded sensor data.

The new version of the algorithm was shown to produce better models from less data and fewer training iterations, and was capable of learning models for environments containing loops. However, this approach still requires a large amount of data for re-estimating the probabilities in the model, and this data has to be collected manually by driving the robot around every section of the environment many times. The improved algorithm remains computationally expensive and is also affected by local maxima, so there is no guarantee that it would find a map which is topologically or geometrically correct.

A major drawback of the HMM and POMDP models is that they cannot represent arbitrary robot positions and orientations, because the robot's environment model has to be quantized into a set of discrete states. Similarly, landmarks have to be quantized into a discrete set of possible observations, so the approach cannot represent arbitrary sensor patterns. For these reasons, HMMs and POMDPs were not used in this thesis.

2.5 Robots with Hybrid Maps

Some researchers have made attempts to integrate topological and metric representations (Edlinger & Weiss 1995; Yamauchi & Beer 1996; Simhon & Dudek 1998; Gasós & Saffiotti 1999). In this approach, the environment is represented globally as a graph of connected regions. Each region is then represented separately by a small-scale, local metric map. This approach has the advantage that a globally consistent metric map is not required. The topological representation is used for path planning and middle-scale navigation, and the local metric maps are used for small-scale navigation. For example, Simhon & Dudek (1998) addressed the question of how to select good regions of the environment in which to establish local metric maps, based on the robot's ability to carry out precise positioning using its immediate sensory information.

ELDEN

Yamauchi & Beer (1996) developed a navigation system known as *ELDEN* for a Nomad 200 robot, which was tested in an 14 m $\times$ 8m laboratory environment. Exploration was carried out by a reactive controller, using an arbitration mechanism to combine a number of hand-crafted behaviours such as wandering and obstacle avoidance. Dead reckoning was used to construct a topological map. adding a new place whenever the distance to the nearest stored place exceeded a predefined threshold. To correct the odometry drift errors, the robot had to return to its starting location periodically to carry out a special recalibration procedure. Here, an occupancy grid constructed from the current sonar readings

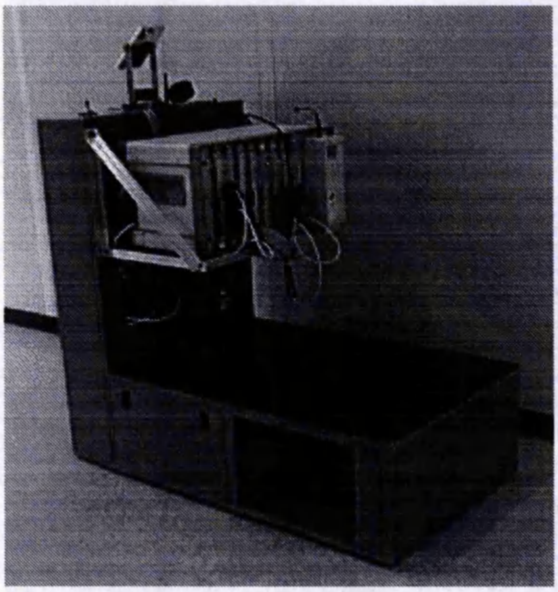

Figure 2.9: *MOBOT-IV*, a robot vehicle at Kaiserslautern University.

was compared to a previously stored grid, using hill climbing to find the transformation producing the best match between the two grids. This transformation was then used to correct the robot's odometry.

The need for regular homing imposes severe scaling limitations on this system, and the reliance on odometry means that the robot would be unable to recover from becoming lost. *ELDEN* was later extended to use a set of stored occupancy grids for recalibration, one for each place node in the map (Yamauchi & Langley 1997) (see also section 6.2.4). However, there was no mechanism for dealing with perceptual aliasing or for correcting the dead reckoning errors between the places in the map, so it is unlikely that the system would scale well to larger environments. Perhaps the most significant contribution of this work is the ability to adapt the robot's map constantly in dynamic environments, using variable-confidence links to represent the uncertainty in the topological relations. Simple rules were used to strengthen and weaken the confidence levels, so that the system could plan an alternative route to a goal location if one of the previously explored paths became blocked by some object.

MOBOT-IV

Edlinger & Weiss (1995) developed a navigation system for the mobile vehicle *MOBOT-IV* (figure 2.9), in which the internal representation acquired by the robot was used to guide the exploration of an unknown environment. This system used a global topological map for navigation, storing the connections between a set of local metric maps obtained using a 360 degree laser range-finder. The laser scanner was first used to construct a temporary metric map from the current sensor data. This representation was then used for self-localisation and to detect areas of unexplored territory. The current sensor map was added to the global map whenever the distance of the robot from the nearest stored node exceeded a predefined threshold.

Self-localisation was achieved through cross-correlation of the current sensor map with the nearest stored scan in the global map (Weiss & von Puttkamer 1995). In this approach, angle histograms were first constructed and convolved (by searching the space of possible translations between the histograms for the current and stored scans) to find the most likely rotation of the robot, as in Hinkel & Knieriemen (1988), then x and y histograms were matched to determine the most likely translation. To detect regions of unexplored territory, a "passage" was defined as a region of open space in the current sensor scan with a width greater than that of the robot. During map building, the detected passages were added to a stack of goal destinations for the navigating robot. Path planning was carried out on the topological map using the $A*$ algorithm.

The *MOBOT-IV* system was tested in a static corridor environment of size 70 m $\times$ 55 m, demonstrating the reliability and scalability of the approach. However, the dependence on high resolution range-finding means that the system would be unlikely to work in a dynamic environment, where many temporary occlusions and variations may be caused by other inhabitants of the environment. The system also depends upon prior position knowledge for self-localisation, so it would be unable to recover its position after becoming lost.

2.6 Concluding Remarks

Navigating mobile robots vary widely, especially in the details of their control architectures and the models used to represent the environment. Nevertheless, many researchers now agree that a hybrid deliberative-reactive architecture provides the most flexible approach for robot control, combining high-level, model-based reasoning with low-level, sensor-motor behaviours (see section 2.1.2).

Two competing paradigms exist for representing the environment. The first consists of detailed metric maps, in which the locations of objects in the robot's environment are specified in a global coordinate system. Of the different proposals for metric maps, the occupancy grid representation developed by Moravec & Elfes (1985) has been tested the most extensively over large environments. This includes perhaps the most successful robot navigation system to date (at least in terms of the number of PhD theses published during the 1990s), namely the *RHINO/MINERVA* project, described in section 2.3.2. In my opinion, this approach has a number of fundamental weaknesses, which are common to all navigation systems based on full metric maps:

1. *Dependence on accurate position information for map building.* Metric maps can only be updated consistently using precise estimates of the robot's position, a requirement which is particularly hard to fulfil in complex, middle-scale environments. This dependency also requires very accurate sensor information for both map building and self-localisation, discussed as follows.

2. *Dependence on accurate sensing.* Robots building metric maps depend critically on a geometrically accurate interpretation of sensor data, both to update the map and to achieve the accuracy of localisation required for map learning. The task of high precision sensing is often undermined by the unpredictable variations which occur in the real world. To meet this requirement, the *RHINO* and *MINERVA* systems have incorporated a series of increasingly sophisticated sensor systems, including laser-range finders and stereo vision. This leads in turn to the following drawback.

3. *High computational requirements.* Robots using detailed metric maps require extremely large amounts of memory and processing power even to navigate in small environments. As a result, these systems are often dependent on radio communication with external processors, a requirement which would be hard to fulfil in many real world environments. In my opinion, metric maps are therefore better suited to small-scale navigation tasks.

Topological maps provide a compact alternative to metric maps which can represent much larger areas using the same resources, and have a much lower dependency on accurate positioning and accurate sensing for map building. A further possibility is provided by hybrid metric-topological maps, consisting of a topologically connected set of *local* metric maps. However, both topological and metric-topological maps can suffer from localisation errors caused by perceptual aliasing and sensor noise. In fact, any navigating robot is bound to become lost at some stage in environments of any real complexity. Without the ability to recover from localisation errors, the robot will be incapable of building a globally consistent map of its environment. A fundamental question is therefore how to identify places reliably which have been visited before by the robot. These issues are explored further in the next chapter, where the requirements for building a mobile robot capable of autonomous navigation in middle-scale environments are determined.

Chapter 3

Requirements Analysis

About this chapter. An overview of the requirements for autonomous robot navigation in real world environments is first provided. This is followed by details of the specific needs of self-localisation and the map building scheme required, together with a summary of requirements.

3.1 Autonomous Mobile Robot Navigation

This chapter discusses the requirements for building a navigating robot capable of operating in indoor, middle-scale environments. In particular, these environments will be *unknown* — the robot will have no pre-installed map, *unmodified* — the environments will not be altered for the purpose of robot navigation, and *semi-structured* — while the basic structure of an environment may be constant over time, there will be many variations in the appearance of that environment to the robot which cannot be predicted.

Perhaps the most essential competence required for robot navigation (other than staying operational and avoiding collisions) is that of self-localisation. Without the ability to identify locations reliably, a mobile robot will inevitably become lost, and therefore be unable to move reliably between target locations.

Proprioceptive sensor systems such as wheel encoders are unsuitable for position estimation over distances of any real significance because of drift errors caused by wheel slippage. Dead reckoning also depends on *a priori* knowledge of the robot's position, and thus cannot relocate the robot should it become lost. Therefore, a navigating robot must depend primarily on its perception of external environmental features or *landmarks* for self-localisation. A key question is which features to extract from the robot's sensor readings.

Localisation in turn requires a map of the environment. To be capable of independent operation, the robot needs the ability to construct its own maps online using only its own sensory impressions of the target environment. To obtain this sensory information, the robot needs to be capable of *goal-directed* exploration; that means being able to identify possible areas of unexplored territory, navigating towards these areas using its own map, and incrementally updating the map at the same time. Basic competences are required for avoiding collisions and following planned paths through the environment.

There are a number of issues in the map learning problem which also need to be addressed. The sensory information available to the robot is noisy, and can produce errors when integrated into the robot's map. In particular, the robot's odometry is subject to drift errors, which can lead to an inconsistent mapping of the environment. To obtain a coherent representation of the environment which can be reconciled with future sensory perceptions, some means of maintaining consistency in the map is required.

It is inevitable at some point that the robot will become lost. In real world environments, unpredictable variations in the robot's sensor data will occasionally result in localisation errors, regardless of the accuracy of the actual sensing and position tracking methods employed. The ability to relocalise after becoming lost, a (hardest) special case of the problem of *global localisation*, is therefore essential if the robot is to be capable of recovery from such errors.

Finally, all of the above requirements need to be handled on a robot platform which has limited computational resources. Navigation in unmodified, middle-scale environments means that communication with external computers may not always be possible. Therefore, all of the computational mechanisms developed must be tractable and allow for real-time operation.

3.1.1 The Need for Quantitative Performance Measures

So far, mobile robotics research has had only limited success in achieving the objective of building robots capable of autonomous navigation in unmodified, real world environments. This is due in part to the numerous scientific challenges discussed briefly in the above section. However, in the opinion of a growing number of researchers (Smithers 1995; Lee & Recce 1997; Nehmzow 1997), there is a second reason why progress has been slow; objective comparison of existing work is often impossible, because researchers use different robots, apply different quality measures and conduct different experiments to validate their work. Without the ability to make meaningful comparisons between results, it is perhaps not surprising that overcoming the scientific challenges has been difficult.

In this thesis, quantitative measures of localisation performance and map quality are required. These measures would provide not only the criteria for assessing the success of the research, but also a means of developing and refining the mechanisms for map building and self-localisation, allowing the effect of individual parameters and system sub-components to be assessed. In fact, localisation quality and map quality are closely related; for example, the quality of a map can be assessed using the robot's ability to localise itself using that same map. The required performance measures should be applicable to a wide variety of different mechanisms for map building and self-localisation, allowing disparate systems to be compared under the same experimental conditions.

3.2 Self-Localisation

3.2.1 Self-Orientation

In attempting to recognise locations, the robot is first faced with a basic problem, namely that the appearance of locations to the robot depends on the direction from which the robot views the location. One solution to this problem would be to make sure that the robot always approached locations from the same direction, by constraining the behaviour of the robot to following fixed paths. However, this approach will not be suitable for exploring and navigating in middle-scale environments of any real complexity.

Self-orientation is often treated as an integral part of the self-localisation problem in mobile robotics. A common approach involves matching the robot's current sensory information to a stored map, searching the space of possible rotations as well as translations to find the best match (see e.g., Moravec & Elfes (1985)). The alternative would be to treat the problem of self-localisation separately, using a compass sense to first remove the problem of self-orientation.

There is certainly some evidence that biological navigation systems may rely on the latter approach. For example, pigeons may use either a geomagnetic compass or an internal function based on the position of the sun at the horizon to calculate their orientation (O'Keefe & Nadel 1977, p. 65) (see also section 11.2.1). In rats, separate neurons have been found which signal the orientation of the animal's head, irrespective of the animal's actual location, known as "head-direction" cells. These contrast with the "place cells" which fire maximally at a particular location, irrespective of the animal's orientation (see citations in McNaughton *et al.* (1996)).

Using a compass on the mobile robot would have the advantage of greatly simplifying the self-localisation problem, reducing the search space in matching sensor readings to the map by an order of magnitude. It also has the disadvantage of reliance on one particular sensor, which will be subject to noise in real world environments. On the Nomad 200 robot, the flux-gate compass is subject to variations in the magnetic field of the environment, and occasionally fails completely to find magnetic North. Therefore, in order to use this compass reliably, some method of dealing with the noise in the compass readings is required.

3.2.2 Landmark Recognition

Self-localisation is possible using *artificial* landmarks such as beacons. However, this research is concerned with navigation in unmodified environments, so the robot must depend solely on its own perceptions for landmark recognition. This is complicated by the problems of using sensors in the real world. For example, due to sensor noise, the robot will often obtain different sensor readings when revisiting a previously encountered location — individual sensory perceptions may be inconsistent and unreliable.

In order to cope with sensor noise, the robot should be capable of generalising on its perceptions in order to extract the salient features in a given situation, without being too distracted by the finer detail of individual sensor patterns. For robots equipped with sonar sensors, such as the Nomad 200 robot used in this thesis, other problems which must be overcome include *cross-talk*, where the echo from one sonar transmitter is picked up by a different receiver, and *specular reflection*, where a sonar receiver fails to detect an echo from the first object encountered by the transmitted sound pulse, instead receiving a reflected return from some other object. The robot's range readings may also be affected by occlusions caused by other inhabitants of the environment.

A further problem is that people observe the world differently to robots, a problem described as *perceptual discrepancy* by Nehmzow & Mitchell (1995). This means that the designer may not select the most appropriate landmarks for the robot, due to the mismatch between the perceptual apparatus of the robot and that of the human designer. It is for precisely this reason that landmark recognition systems based on artificially constructed maps such as CAD models can be brittle, and may suffer from the *correspondence problem*, where the robot fails to match its sensory perceptions to the corresponding components of the map. Rather than attempting to identify specific objects in the robot's environment, e.g., walls, doors, etc., the main requirement for reliable landmark recognition is the ability to recognise distinct sensory impressions of an environment according to their similarity. The required mechanism should be able to distinguish sensor patterns which share common features from those which do not.

Landmark recognition systems based on self-organisation, such as Kohonen networks (Kohonen 1993), and ART networks (Grossberg 1988), present one possible means of achieving these objectives. These systems cluster together similar sensor patterns, avoiding many of the difficulties of sensor noise, specular reflection and cross-talk because they do not attempt to extract explicit metric information from their sensory inputs. Ultimately, however, the appropriate mechanism for landmark recognition on a particular robot can only be validated through experiments on the robot itself. The decision on which mechanism to choose was therefore based on the results of an experimental comparison of systems described in chapter 6.

3.2.3 Global Localisation

In addition to the problems described above, the task of localisation is made non–trivial by the fact that many places may look the same to the robot, a characteristic of sensory perception known as *perceptual aliasing*. Thus, the currently perceivable landmarks alone may not be sufficient to uniquely identify the true location of the robot. In many localisation systems, this problem is avoided in the first instance by *position tracking*, using prior position information from odometry to constrain the landmark recognition process to a small area in the map around the robot's current position estimate, thereby eliminating many other places which may look the same to the robot. The odometry is then corrected using the known positions of the perceived landmarks in the map. However, this approach is bound to fail eventually when the robot becomes lost, because the robot's position cannot be recovered without using some other mechanism for disambiguating perceptually aliased locations.

Global localisation is generally recognised as one of the hardest problems in mobile robot navigation. It requires exploration, since the robot often needs more than just its sensory impressions of a single location to relocalise, and correlation of the subsequent sensory perceptions with the map. It also needs methods for representing and reasoning with the uncertainty in the robot's position, since this can never be established with absolute certainty. Given the difficulty of the task, and the fundamental importance of this particular competence above all others, I would argue strongly that the robot should try to use as much of the available sensory information as possible in trying to localise itself. This includes information from odometry about the *relative* displacement of the robot between observed landmarks (since odometry can only produce reliable position estimates over *small* distances) as well as the landmarks themselves. In some situations, for example where an environment contains repeated constellations of very similar looking places, reliable landmark-based localisation might only be possible with the help of extra sensory information from odometry. Methods for combining information from both the robot's exteroceptive and proprioceptive sensors are therefore required, and metric information would also need to be incorporated somehow into the robot's map.

3.3 The Choice of Map

3.3.1 Topological versus Metric Maps

I have argued for the use of metric information as well as landmarks in the robot's map for self-localisation, on the grounds that many sources of sensory information will be better than one. Yet in the preceding chapter, consideration of the previous work on navigating mobile robots lead to the conclusion that topological maps are better suited to navigation in middle-scale environments than metric maps. The low computational cost of maintaining topological representations, together with their reduced dependence on accurate sensing and accurate position estimation for map building, were particular reasons for reaching this conclusion.

Lee (1995, p. 33) introduced a classification of robot maps based on the notion of geometric *strength*, consisting of the following categories:

1. *Recognisable Locations.* The map consists solely of a set of locations which

can be identified by the robot — no geometric information can be recovered.

2. *Topological Maps.* The map consists of recognisable locations, and information about the connectivity between visited locations can also be recovered.

3. *Metric Topological Maps.* In addition to topology, metric information can be recovered about paths which have been travelled.

4. *Full Metric Maps.* Metric information can be recovered about any objects in the map.

In determining the requirements for a delivery application, Lee argued that his robot needed a full metric map because topological maps would not allow the robot to take the "short cuts" which he claimed were necessary for efficient performance on the delivery task (1995, p. 47). However, Lee's robot experiments were conducted in a small-scale, laboratory environment, where the impact of dead reckoning errors would have been relatively minor. It is very unlikely that such an approach would scale to a larger environment, where accurate positioning becomes much harder to achieve, since precise position information is needed to update full metric maps. In middle-scale environments, robots trying to build accurate metric maps face a losing battle in reducing the uncertainty in the robot's estimated position against problems such as wheel slippage, sensor noise, specular reflection, cross-talk, perceptual aliasing and becoming lost, even without the presence of humans and changes to the environment.

At the same time, I concur with Lee on the limitations of purely topological maps. Without using any stored metric information, it would be very difficult for a robot to identify perceptually aliased locations, explore complex environments and plan shortest paths. To solve this dilemma, some means of reconciling topological and metric representations is required. Thrun (1998b) developed a dual mapping system in which a topological map is derived from an underlying metric map, described in section 2.3.2. However, this approach means that the robot would still be affected by all of the problems of maintaining a full metric map. The alternative investigated in this thesis is to use a topological map to provide the basic underlying representation in the robot's environment model, then to augment this with additional metric information as required. The requirements of such a scheme are discussed as follows.

3.3.2 The Use of Metric Information

The use of odometry for navigation is the most important consideration, since this will dictate the required representation of metric information in the robot's map. One possible mapping scheme would be to record only *local* metric relations in map, that is, to add information concerning the relative distances and angles between stored locations. Self-localisation would then consist of correlating the observed sequence of sighted landmarks and the measured displacement of the robot between sightings with the stored relations in the map. However, to build such a map by itself, the robot would have to explore every possible transition in the map between neighbouring places to obtain the metric information needed for localisation (a very inefficient map building strategy in itself). This in turn would require the ability to recognise previously visited locations — that would, however, be very difficult because some of those places might themselves be perceptual aliased locations, which could then only be identified using an additional exploration strategy known as a *rehearsal procedure* (Kuipers & Byun 1991).

An alternative approach would be to incorporate *global* metric information into the robot's map, that is, to assign Cartesian coordinates to each of the stored locations in the map. Self-localisation would then involve correlation of the perceived metric relations between sighted landmarks and the equivalent, inferred relations within the Cartesian coordinate system. This approach would have the advantage that perceptually aliased locations could be recognised even when approached from an unfamiliar direction, using the global metric information in the map to resolve perceptual ambiguity and eliminating the need for a rehearsal procedure. In addition, a global coordinate system would be useful for mapping an unknown environment, providing a framework for integrating new sensory information and adding new territory to the map.

3.4 Map Building

The above argument for global metric information in the robot's map is based on the assumption of *geometric consistency.* In practice, however, odometry drift errors mean that it is very difficult to maintain a globally consistent coordinate system using dead reckoning. The position estimates obtained from wheel encoders quickly become unreliable, leading to inconsistencies with the existing metric information in the map. Mechanisms for performing odometry correction such as the Kalman filter cannot solve this problem either, because these methods can only produce accurate location estimates within territory which has already been mapped correctly. In order to obtain reliable global metric information in a robot's map, some other means of enforcing geometric consistency is required.

Additionally, to operate in an unknown environment, the robot needs to be able to acquire its own maps through exploration. In complex environments, this means being able to use the map itself to navigate to possible areas of unexplored territory. The robot must then head off into unfamiliar territory to obtain the new sensory information required to extend the map. However, this introduces a new problem, namely how to self-localise in uncharted territory. One possibility is to stay within range of existing landmarks in the map, another is to use a local dead reckoning strategy. The robot also needs to decide which way to go. One method would be to head off in a random direction, but this could be inefficient since it might take many attempts to reach some parts of the environment. Alternately, the robot could use some mechanism for detecting possible areas of uncharted territory, such as areas of open space which are not present in the map.

3.5 Summary of Requirements

In order to build a mobile robot capable of navigating in unmodified, middle-scale environments from scratch, the following requirements were determined:

1. *Landmark Recognition.* The problems of odometry drift mean that the robot must rely on its perception of external environmental features for navigation. The robot needs to generalise on its perceptions to extract the salient features from its raw sensor data, and to overcome problems including sensor noise and specular reflection.

2. *Self-Orientation.* In order to recognise landmarks from arbitrary orientations, the robot needs a compass sense. This can be obtained either using a real compass or an equivalent mechanism for recovering the robot's orientation unambiguously from external environmental features.

3. *Global Localisation.* The robot needs the ability to re-identify locations, especially after becoming lost. In order to overcome the problem of perceptual aliasing (similar looking places) the robot needs to explore the environment.

Extra sensory information from relative odometry between perceived landmarks can help to relocalise the robot.

4. *Metric-Topological Map.* Due to limited computational resources, a robot needs an efficient representation of a middle-scale environment — topological maps are more compact, and require less accurate position information for map learning than metric maps. However, tasks such as global localisation and map building are very difficult without metric information in the robot's map — a global coordinate system would be the most flexible approach.

5. *Map Learning.* The robot needs the ability to build its own maps to operate in unknown environments. Incorporating a global coordinate system into the robot's map introduces a further requirement, that of *geometric consistency.* A globally consistent coordinate system cannot be obtained using dead reckoning — therefore, some other mechanism for enforcing geometric consistency is required.

6. *Exploration.* Global localisation and map building both require exploration in order to collect useful sensory information. To extend an existing map, the robot needs to be able to navigate towards areas of unexplored territory. This in turn requires basic sensor-motor competences for avoiding collisions and following planned paths through the environment.

To evaluate the performance of the system and its sub-components, quantitative measures are required, in particular for assessing localisation quality and map quality. In the following chapter, we turn to some of the basic mechanisms and assumptions underpinning the rest of this research.

Chapter 4

Basic Mechanisms and Assumptions

About this chapter. This chapter provides a brief description of the mechanisms developed for sensing and low-level motor control and the assumptions behind the research presented in the following chapters.

4.1 System Overview

In the following chapters, the development of a complete system for concurrent self-localisation and map building by a navigating mobile robot is described. The research was carried out on the Nomad 200 mobile robot *FortyTwo*, using its sonar, infrared and odometry sensors and an on-board flux-gate compass. The robot has three degrees of freedom, being able to rotate the turret independently of the base of the robot, which contains separate motors to control the robot's translational and rotational movements (see figure 1.4).

In this thesis, the turret was controlled separately using the compass to provide the robot with a canonical view of locations, described in section 4.2. The compass sense was also used to remove the rotational drift affecting the robot's odometry, described in section 4.3. Landmark recognition was carried out using the sonar sensors, while tasks requiring close proximity sensing, such as collision avoidance, were carried out using a combination of sonar and infrared.

All of the subsequent experiments were conducted in a series of unmodified, middle-scale environments in the Computer Science building at Manchester University. The environments were assumed to be unknown, unmodified and semi-structured, being subject to transient variations in the robot's sensor data. People were free to move in the immediate vicinity of the robot, but it was assumed that the robot's path would never be completely blocked during exploration, so that the robot could construct its maps in a single tour of the environment.

The high-level control routines used for map building and self-localisation are described in subsequent chapters. For low-level motor control, previously acquired behaviours for obstacle avoidance and wall-following were used, described in section 4.4.

4.1.1 Metric-Topological Map

For the reasons discussed in the previous chapter, it was decided to use a hybrid metric-topological map. The basic underlying representation consisted of a topologically connected set of places, each place being identified by a description of the perceived environmental features or landmarks at that location. To improve the reliability of landmark recognition, a sensing strategy designed to increase the resolution of the robot's sensors was used, described in section 4.5. The experiments conducted to determine an appropriate landmark recognition mechanism for the Nomad 200 are described in chapter 6.

In addition, the map was augmented with metric information describing the relative displacement of the robot between connected places. The optimisation algorithm described in chapter 8 was used to assign a globally consistent set of Cartesian coordinates to the places in the map based on the local metric relations. The coordinates were used for a number of purposes, including self-localisation (chapter 7) and exploration of unknown environments (chapter 9). The local metric information used for both map building and self-localisation was obtained from the dead reckoning mechanism described in section 4.3.

Full details of the representation schema can be found in the chapters on global localisation (section 7.2.1) and map learning (section 8.2).

4.2 Compass Sense

A basic problem for landmark-based robot navigation is that the appearance of locations to the robot depends on the direction in which the robot's sensors are

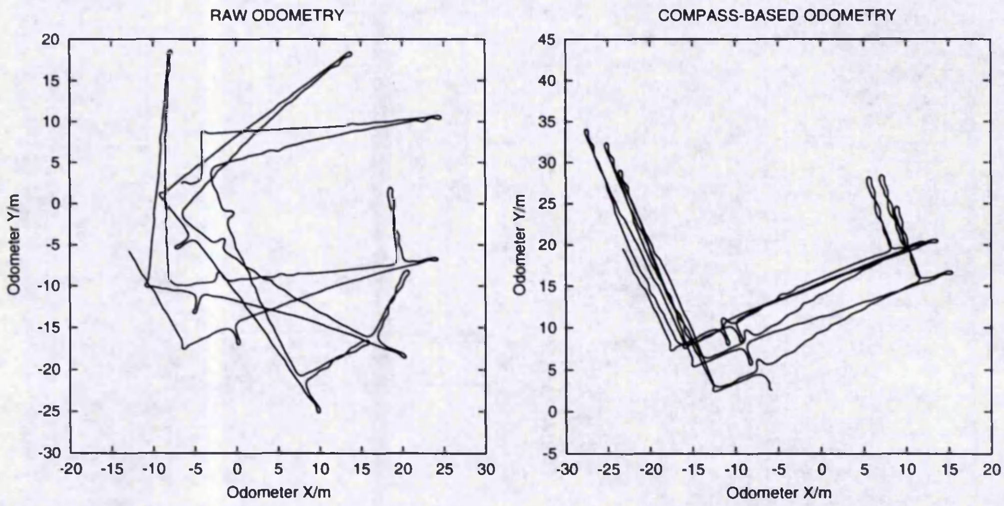

Figure 4.1: Left: raw odometry. Right: compass-based odometry. The accumulated rotational drift in the robot's raw odometry was removed on-line using the compass sense.

facing. To eliminate this problem, a separate behaviour was used to rotate the turret at small speeds, while the robot was in motion, so that the turret always had the same orientation, as indicated by the flux-gate compass. The effect of this behaviour was to smooth out local fluctuations in the magnetic field of the environment, so that the robot explored with its sensors facing in the same global direction throughout rather than the direction of travel. Consequently, the appearance of locations depended on the robot's position alone, not its orientation.

While this method is robust in dealing with minor variations in the magnetic field, ferrous building materials could cause severe compass errors in some environments. However, for the purpose of landmark recognition, the important point is that the appearance of locations remains invariant with respect to the robot's heading, regardless of global magnetic deviations. For the purpose of global self-orientation, a more reliable compass sense could be obtained by integrating perceptual information from the robot's exteroceptive and proprioceptive sensors, as in the self-orientation system described by Li *et al.* (1998), or by using correlation with a vision-based map of the ceiling as in Thrun *et al.* (1999).

4.3 Dead Reckoning

4.3.1 Compass-Based Odometry

Dead reckoning using odometry consists of integrating small measurements from the robot's wheel encoders. Separate encoders track the robot's rotational and translational movements, both of which are subject to cumulative drift errors. In this thesis, however, the rotational drift affecting the robot's odometry was removed as follows. Instead of using the robot's rotational wheel encoders for the on-line dead reckoning, the relative angular displacement of the turret against the direction of travel was used (see figure 4.1). The robot's (x, y) coordinates were continually recalculated according to the following equations:

$$x' = x + \epsilon_t \cos \alpha_t \tag{4.1}$$

$$y' = y + \epsilon_t \sin \alpha_t \tag{4.2}$$

where ϵ_t refers to the distance travelled, as measured using the robot's raw odometry, and α_t to the measured angle at each time step t. This eliminated the accumulated angular drift because the turret was anchored to magnetic North by the compass sense. Using compass-based odometry on the Nomad 200 leaves a translational drift error of up to 5% of distance travelled. Comparison of the diagrams in figure 4.1 shows clearly the two components of dead reckoning error, translational and rotational.

4.3.2 Representing the Uncertainty

In mobile robotics, the inherent uncertainty of dead reckoning means that position estimates are typically represented as a probability density function rather than a single point value. Because the compass sense was used to remove the rotational error in the robot's odometry, it was possible to use a very simple noise model to represent the uncertainty in the robot's distance measurements. For a given point (x, y), the noise is assumed to be distributed equally in all directions according to a Gaussian function of the distance from (x, y). Thus, the area in which the robot may be located with non-negligible probability is modelled by a circle, and the uncertainty in any point measurement is represented by a single variance σ^2 (see figure 4.2).

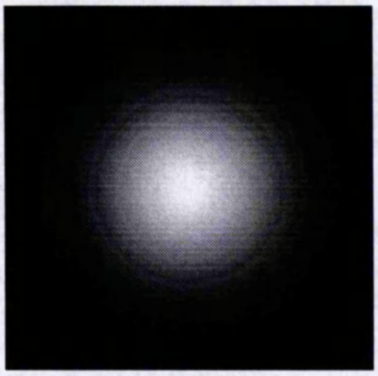

Figure 4.2: The assumption of circular, Gaussian noise. The noise in the robot's position estimates is assumed to be distributed equally in all directions according to a Gaussian distribution; thus, position estimates are represented by a triple $\langle x, y, \sigma^2 \rangle$ where (x, y) is the mean position and σ^2 the corresponding variance (see section 4.3.2).

The assumption of circular noise means that the robot's actuators are not modelled accurately. In some cases, it might be preferable to use a model which captures more accurately the underlying physics of robot motion, for example, with separate components in x, y and θ (see e.g., Smith *et al.* (1990)). The assumption of Gaussian noise also means that each distance measurement is considered to be independent, and that the model cannot deal with cumulative phenomena such as battery drain. Nevertheless, the results presented later in the thesis show that the simple model is sufficient for robust navigation performance in the real world. The model is also attractive for real-time operation in middle-scale environments because of its low computation cost.

4.3.3 Retrospective Odometry Correction

The above sections described the on-line dead reckoning method (section 4.3.1) and the corresponding model of the accumulated dead reckoning error (section 4.3.2). In this section, I introduce an off-line method for dealing with the remaining translational error affecting the robot's compass-based odometry. Provided that clearly identifiable landmarks are available, this error can be removed post-hoc by applying the following procedure (see figure 4.3).

Firstly, the recorded data is divided manually into laps by finding some prominent feature, such as a corner, in the odometer trace. (This could easily be

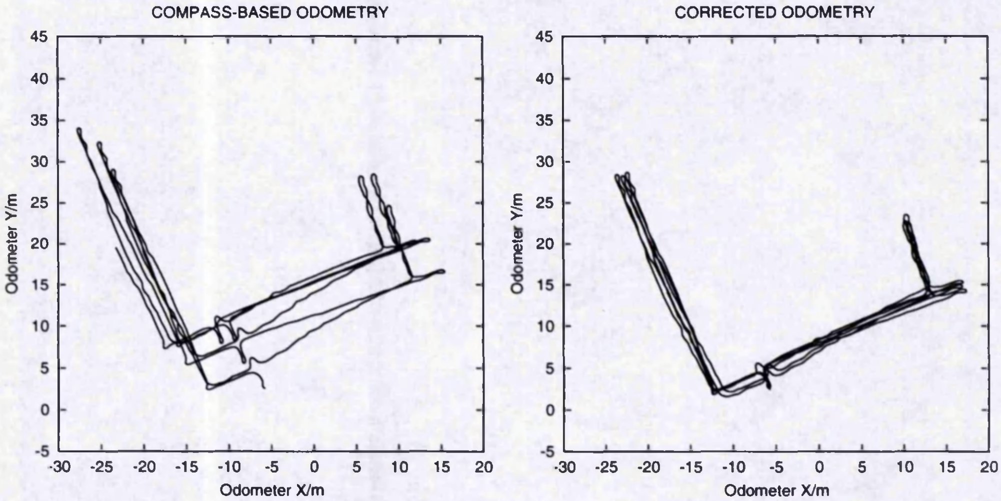

Figure 4.3: Left: compass-based odometry. Right: retrospectively corrected odometry. The remaining translation error in the robot's compass-based odometry was removed post-hoc through manual identification of prominent landmarks in the recorded odometer trace (see section 4.3.3).

done automatically by using an external sensor to detect the completion of another lap by the robot, for example, using an overhead cross-bar sensor as in Smithers (1995).) For each lap, the accumulated drift error is then removed by correcting each data point by an amount proportional to the distance travelled along the route. Finally, a translation is applied to all of the laps but the first one so that all laps start and end at the same (x, y) position.

Clearly, this technique cannot be used for real-time robot navigation. However, it is very useful both in studying the problem of self-localisation and in analysing the performance of the navigating robot. The application of this method — including its limitations — to the measurement of localisation performance is discussed in section 5.3.1.

4.4 Reactive Behaviours

In order to avoid pre-installation of control knowledge by the system designer, the low-level sensor-motor behaviours required for navigation were acquired independently by the robot using an artificial neural network. This mechanism consisted of a single perceptron which was trained to associate the robot's sonar and infrared sensor readings with continuous-valued motor velocities (see figure 2.1), as in Nehmzow (1994).

The sensor input was first pre-processed to take into account the heading of the robot, since the robot's sensors were fixed to magnetic North by the compass sense. The 11 sonar and 11 infrared readings centred around the direction of travel were then presented in the input vector to the perceptron. Two output units were used to control the robot's motor actions, one to determine the translational velocity and the other the rotational velocity.

The training signal was determined using a predefined set of "instinct rules", as in Nehmzow (1992). To obtain the required behaviour for avoiding obstacles, the robot was provided with two instinct rules, one to teach the robot to move faster if its actual translational velocity fell below a prespecified threshold, and one to teach it to turn away from objects detected within a certain sensor range. In order to obtain the required behaviour for wall-following, a third instinct rule was added, teaching the robot to turn towards the nearest perceived object if the distance to that object exceeded a given threshold. The three instinct rules can be summarised as follows:

1. Move forwards.

2. Turn away from obstacle if nearest infrared reading < 5.

3. Turn towards obstacle if nearest infrared reading > 10.

This approach has the advantage that the neural network can be quickly retrained to work in new environments. In these experiments, the required behaviours were first acquired in a small-scale environment, then the weights were fixed before the system was used in middle-scale environments. In fact, the weights in the network could be continuously adapted on-line using the instinct rules if required, as in Nehmzow (1992), though this was found to be unnecessary for the experiments conducted here.

4.5 Sensing Strategy

The sensing strategy used for map building and self-localisation consisted of periodically stopping the robot and then rotating its turret to obtain a detailed sensor scan, before resuming exploration. By taking nine sets of sensor readings at 2.5

degree intervals, a scan consisting of 144 sonar and 144 infrared readings was obtained. This sensory information was then used as input to the mechanisms for landmark recognition, self-localisation and map acquisition.

The motivation for using this sensing strategy was firstly to increase the resolution of the robot's sensory apparatus, thereby improving the accuracy of feature detection and landmark recognition by the robot. Secondly, the strategy reduces the effect of variations in the robot's environment. For example, if a person walks past the robot, it is likely that several of the robot's sonar readings may be occluded, which can often result in landmark identification errors. By taking a succession of sensor readings at small time intervals while the robot rotates its turret, this effect can be dramatically reduced, since a much smaller proportion of the total number of sensor readings will be affected.

However, while using more exteroceptive sensory information reduces the likelihood of localisation errors, this strategy can never be guaranteed to eliminate these errors completely, due to the problem of perceptual aliasing. For this reason, the evidence-based localisation algorithm described in chapter 7 was developed.

4.6 Summary

Environments are assumed to be unknown, unmodified and semi-structured, and are represented using a metric-topological map. A geomagnetic compass is used to remove the problem of self-orientation and the accumulated rotational drift errors affecting the robot's odometry. The uncertainty in the robot's distance measurements is represented using a simple noise model; in all of the algorithms in the following chapters, position estimates in Cartesian space are represented by an (x, y) coordinate associated with a single variance measure. Self-acquired reactive behaviours for obstacle avoidance and wall-following are used for low-level sensor-motor control. A technique for taking detailed range-finder scans is used to increase the resolution of the robot's sensors for landmark recognition.

Chapter 5

Measurement of Localisation Performance

About this chapter. This chapter addresses the question of how to measure the performance of mobile robot self-localisation systems in middle-scale environments. Experimental procedures and a general performance metric are introduced, for which no semantic interpretation of the robot's environment model is required.

5.1 Introduction

So far, relatively few attempts have been made in mobile robotics to quantify robot-environment interactions or to conduct experimental comparisons of navigating robots. Exceptions include the work of Schöner & Dose (1992), Schöner *et al.* (1995) and Smithers (1995), where fundamental sensor-motor behaviours were analysed in terms of dynamical systems theory; Lee (1995) and Lee & Recce (1997), where various exploration strategies for map building were evaluated; and Gutmann & Schlegel (1996), Gutmann *et al.* (1998) and Thrun (1998a), where different algorithms for self-localisation were compared. In this thesis, a general method of evaluating localisation quality over middle-scale environments was required to determine reliable mechanisms for landmark recognition and self-localisation.

A number of different measures have been used by mobile robotics researchers to assess localisation quality. Absolute error, measured as the difference between the robot's predicted and actual position in an external Cartesian reference frame, is one approach (Gutmann & Schlegel 1996; Oore *et al.* 1997; Thrun 1998a). However, this metric is really only feasible for simulations and small-scale environments, because a great deal of time and effort is required to measure the exact position of a moving robot in a middle-scale environment. Also, such a measure does not facilitate comparison with systems which produce a non-Cartesian response to their perceived location, such as self-organising feature maps (Nehmzow *et al.* 1991).

A second approach is to evaluate a percentage correct figure for the system under investigation (Yamauchi & Langley 1997). 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, for example, in robotic implementations of models of the hippocampus (Touretzky *et al.* 1994; Burgess *et al.* 1997; Bethell 1996) and self-organising systems (Nehmzow *et al.* 1991; Keuchal *et al.* 1993; Lambrinos *et al.* 1995), where the robot forms its own internal representation of the world.

A third method was used by Gutmann *et al.* (1998) in an experimental comparison of two self-localisation systems based on metric maps, using sensor-motor data previously recorded by a navigating robot. Again, the accuracy of localisation was measured using absolute error, but global localisation performance was also assessed by measuring the number of times the robot would have become lost using a particular localisation method. However, the latter metric also requires interpretation of the robot's position estimates to determine whether the robot has become lost, and would be difficult to apply to systems which generate a non-Cartesian response.

Instead, the approach taken here was to adopt a "black box" or stimulus-response model of the robot's interactions with its environment. A performance measure based on the statistical notion of mutual information (Shannon & Weaver 1949) is detailed in the next 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 environment model by the observer.

Section 5.3 describes the basic experimental procedure used to calculate this statistic. This includes mechanisms for recording the robot's sensor data and estimating the true location of the robot, since this information is required to assess localisation performance. Section 5.4 describes an extension to this procedure known as the "lost robot experiment", in which the localisation performance of the robot is measured over time as it attempts to relocalise from an unknown starting position.

5.2 The Performance Metric

The calculation of the performance measure is based on a data structure known as a contingency table (Press *et al.* 1992, p. 628). In the example shown in figure 5.1, 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 (known as the robot's *response*, R), and the other to the actual location occupied by the robot (known as the robot's *location*, L). The robot's true location must therefore first be quantized into a set of "bins" — a procedure for automatically recording the actual location of the robot is described in the following section. By convention, the rows of the table are used to represent the response, and the columns to represent the location. For example, figure 5.1 shows one cell containing 19 data points where the robot's response was measured as row 3 and the location as column 5.

For a contingency table, the *Row Totals* for each response i (equation 5.1), *Column Totals* for each location j (equation 5.2) and the *Table Total* (equation 5.3) are calculated as follows, where N_{ij} refers to the number of data points contained in the cell at row i and column j:

$$N_{i\bullet} = \sum_j N_{ij}, \tag{5.1}$$

$$N_{\bullet j} = \sum_i N_{ij}, \tag{5.2}$$

$$N = \sum_{i,j} N_{ij}. \tag{5.3}$$

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.

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

Figure 5.1: 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 recorded by the location binning mechanism described in section 5.3.1. This table represents 100 data points, and also shows the totals for each row and column.

The *Row, Column and Cell Probabilities* can then be calculated using

$$p_{i\bullet} = \frac{N_{i\bullet}}{N}, \tag{5.4}$$

$$p_{\bullet j} = \frac{N_{\bullet j}}{N}, \tag{5.5}$$

$$p_{ij} = \frac{N_{ij}}{N}. \tag{5.6}$$

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 evaluate the entropy of the variables under consideration, i.e., the amount of information required to remove any uncertainty in these quantities. The *Entropy of L* (equation 5.7), the *Entropy of R* (equation 5.8) and the *Mutual Entropy of L and R* (equation 5.9) are defined as

$$H(L) = -\sum_j p_{\bullet j} \ln p_{\bullet j}, \tag{5.7}$$

$$H(R) = -\sum_i p_{i\bullet} \ln p_{i\bullet}, \tag{5.8}$$

$$H(L, R) = -\sum_{i,j} p_{ij} \ln p_{ij}. \tag{5.9}$$

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 as

$$H(L \mid R) = H(L, R) - H(R) \tag{5.10}$$

where

$$0 \leq H(L \mid R) \leq H(L) \tag{5.11}$$

This last property (equation 5.11) means that the range of values for $H(L \mid R)$ will be dependent on the size of the environment, because $H(L)$ increases as the number of location bins increases. For making comparisons across different environments, an alternative statistic is the *Uncertainty Coefficient of L given R*, where the performance measure is scaled to lie between 0 and 1, given as

$$U(L \mid R) \equiv \frac{H(L) - H(L \mid R)}{H(L)}. \tag{5.12}$$

The question asked by this metric is *"How much information does R provide about L?"*. A value of $U(L \mid R) = 0$ means that R provides no useful information about L, and implies that the robot's response never predicts its true location. A value of $U(L \mid R) = 1$ 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, hence no external interpretation of the robot's responses is needed.

5.3 Basic Experimental Procedure

This section describes the basic experimental set-up used to evaluate localisation quality. It has the following components:

1. *Middle-Scale Environment.* The experiment requires a large environment, as we are interested primarily in the robot's ability to recognise distinct locations over middle-scale distances, rather than precise positioning over a small area.

2. *Exploration by Wall-Following.* Wall-following was used 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.

3. *Data Collection.* As the robot explored each environment, a data recording mechanism was used to record the robot's range-finder and odometer sensor readings into a datafile at regular intervals. The sonar and infrared readings were recorded by stopping the robot at 0.50 m intervals, and then rotating the robot's turret to obtain a detailed sensor scan, as described previously in section 4.5. The collected data could then be played back to assess the performance of different localisation systems as required. Using the same recorded data throughout ensured that all experiments were conducted under identical conditions.

4. *Location Recording.* The problem addressed here was to find some way of measuring the "true" location L of the robot for later comparison with the responses of a particular system. In earlier experiments (Duckett & Nehmzow 1997b), this was done manually by drawing a grid on the floor of the environment and recording the time whenever the robot moved into a new grid cell. However, this process was time-consuming and prone to human error, so an automated location recording mechanism was developed, described in the next section.

5. *Quantitative Performance Measure.* The uncertainty coefficient $U(L \mid R)$ (equation 5.12) was used to determine the strength of the association between the responses of the localisation system under investigation and the

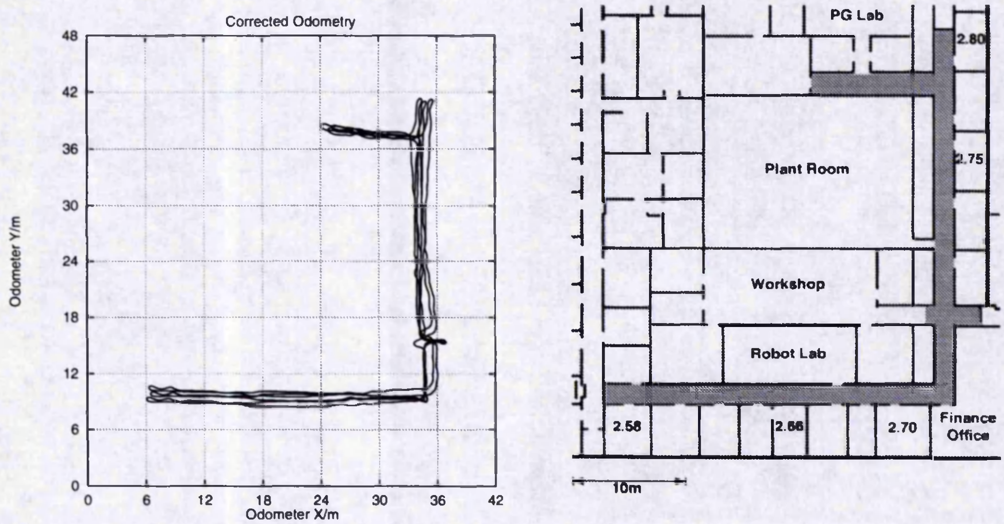

Figure 5.2: Left: location binning mechanism. Right: corresponding floor plan. The recorded odometry data was first corrected using the technique described in section 4.3.3 (see also figure 4.3). The dotted grid was then used to coarse-code the corrected location data into bins.

true location of the robot. Some heuristics for choosing an appropriate number of responses and location bins are discussed in section 5.3.2.

5.3.1 Location Binning Mechanism

To determine the robot's true location L, the technique for retrospective odometry correction described in section 4.3.3 was applied. Recall that this procedure involves applying a post-hoc correction to the recorded odometry data for each successive lap of the environment by the wall-following robot. The corrected data was then coarse-coded into equally sized bins as illustrated in figure 5.2. The orientation and positioning of the dotted grid shown in figure 5.2 over the corrected odometer trace was determined using an exhaustive search procedure to minimise the number of bins occupied, and then to maximise the value of $H(L)$ (equation 5.7) where several possible grid positions produced the same number of bins. The choice of bin size is discussed in the following section.

For performance evaluation, the recorded sensor data from the robot's first lap of the environment was used for map building (or landmark learning), and the data from the subsequent laps was used for testing.

Note: The procedure for correcting the recorded odometry data assumes that drift error occurs equally over distance travelled. In practice, the robot's wheels tend to slip more on turns than on forward motion. Wheel slippage can also vary according to the speed and acceleration of the robot. There can also be magnetic variations in the environment which affect the robot's on-line compass-based odometry; this is noticeable, for example, in figure 5.2 when the robot passed the workshop, a room containing heavy machine tools. However, such distortions in the corrected odometer trace should not adversely affect the performance measures, provided that the assignment of bins to locations is consistent between successive laps of the environment by the wall-following robot. Note also that any variations in the recorded sensor data will affect all systems equally in any performance comparisons; sensor noise is a natural, inseparable aspect of robot-environment interaction, which should be included in any assessment of navigating mobile robots.

5.3.2 Choosing the Experimental Parameters

There are a number of important parameter values which must be determined before calculating the uncertainty coefficient. Firstly, the number of data points used for performance evaluation should be selected so that all parts of the environment are equally represented, otherwise the performance measures will be biased towards one particular region. For data collected by wall-following, this means using data from a whole number of laps (circuits) of the environment.

The next decision is to select the number of responses N_R by configuring the individual system parameters of the particular localisation mechanism under investigation (for example, many systems have some critical parameter which determines the sensitivity to perceptual detail, and therefore the number of different landmarks recognised). The higher the number of responses a system produces, the more information it can give about its true location. Therefore, if several different systems are to be compared, these systems should be configured as closely as possible to produce the same number of responses N_R in the same environment.

The size of the location bins, and therefore the number of location bins N_L, depends on the particular experiment being conducted. For assessing global localisation, the bins can be very large, whereas a fine resolution would be required to measure the accuracy of a robot's position estimates.

Another important consideration is the effect of quantization errors due to the grid representation used by the location binning mechanism (see figure 5.2). For some systems, there may be a big discrepancy between the shape of the locations recognised by the robot and the square cells of the grid — for example, consider the "situation areas" shown in figure 2.6 for the maps built by the ALEF robot (Kurz 1996). In this case, big location bins should be used, so that N_R is several times larger than N_L (i.e., each location bin contains several robot responses) in order to minimise discrepancies caused by quantization effects, particularly if the experiment involves comparing a number of disparate systems.

A detailed example is presented in the next chapter, comprising an experimental comparison of landmark recognition systems.

5.4 The Lost Robot Experiment

As discussed in the requirements analysis, a mobile robot needs the ability to recover from becoming lost if it is to navigate robustly under realistic operating conditions, such as dynamic environments. To measure this ability, I designed the following experiment. The basic idea is that the robot is first subjected to a "virtual kidnapping" (after Engelson (1994)), being transported to an *a priori* unknown location with its sensors "blind-folded" and odometry disabled during this move. The performance of the particular localisation system under investigation is then measured against the distance travelled by the robot from the starting position.

The "kidnapping" is implemented by re-initialising the localisation system at the start of each experimental trial. The system is then tested using a stream of previously recorded sensor-motor data, as collected by the wall-following robot, and the responses of that system to the played-back sensory data are logged against distance travelled. This procedure is repeated many times ("trials") from a series of different starting positions, and the uncertainty coefficient $U(L \mid R)$ is then calculated against distance, using all of the logged location-response data. Implementing this procedure requires some careful treatment of the recorded data, described as follows.

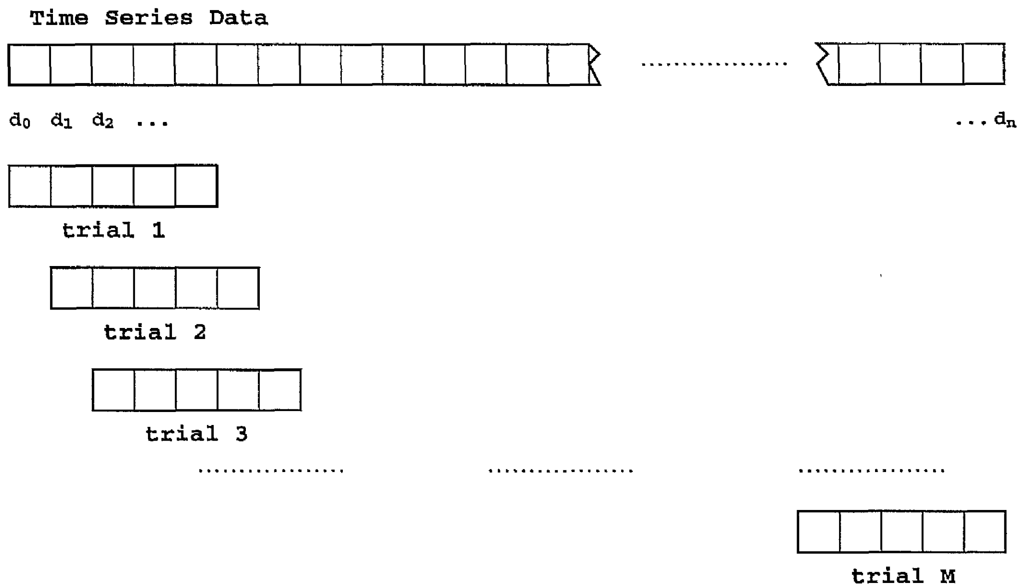

Figure 5.3: Treatment of time series data in the lost robot experiment. The recorded sensor-motor data is played back to test the localisation system under investigation. Each trial $(1, 2, ..., M)$ lasts for a fixed distance l and begins from a different position along the recorded route data $(d_0, d_1, ...d_{n-l})$.

5.4.1 Measuring Performance Against Distance Travelled

Again, the first lap of the recorded sensor-motor data is used for map building. The localisation system under investigation is then tested over a large number of trials using the remaining laps. Each trial begins from a different location along the route traversed by the robot, and lasts for a fixed distance (see figure 5.3). There is some overlap between successive trials, but each starts from a different position along the recorded time-series data. Again, the number of the trials should be chosen carefully so that each part of the environment is represented equally in the test data.

In the experiments conducted in this thesis, trials of length 20 m were used, starting at 0.5 m intervals along the recorded route data. For the subsequent analysis, it is assumed that all trials begin nominally at distance 0 m and end at distance 20 m. After testing the localisation system for the appropriate number of trials, the resulting location-response data is split into 201 separate contingency tables, one for the starting point plus one for each 0.10 m travelled by the robot. The performance measure $U(L \mid R)$ is then calculated for each of the tables. A number of examples where the uncertainty coefficient is plotted against distance can be found in chapter 7 (section 7.4).

5.5 Concluding Remarks

This chapter presented the experimental procedures used for evaluating localisation performance in the rest of the thesis, including an entropy-based performance measure (the uncertainty coefficient U) and a novel mechanism for recording the true location of the robot. The approach has the advantage that it requires no interpretation of the "correct" response made by the robot and no optimum standard has to be established by the observer. The entropy-based performance measure is consistent with the "Infomax" principle (Linsker 1988), which states that a perceptual device should organise itself to transmit the maximum possible information about its inputs. Furthermore, only approximate location recording is required to evaluate localisation quality, because the uncertainty coefficient is measured over a set of discrete location bins rather than an exact Cartesian reference frame. In the next chapter, the basic experimental procedure is used to compare a number of previous approaches for landmark recognition.

Chapter 6

What is a good landmark?

About this chapter. Due to the fundamental unreliability of navigation by dead reckoning, a robot must depend on its perception of landmarks for self-localisation. An empirical study is presented in which a number of algorithms for landmark recognition are compared quantitatively in four different environments.

6.1 Introduction

This chapter examines possible approaches for landmark recognition by a navigating mobile robot. A variety of landmark recognition mechanisms have been proposed by previous researchers (see Borenstein *et al.* (1996) for a detailed survey). Two approaches are commonly found:

1. *Designer-determined landmarks.* One method is to provide the robot with predetermined feature categories such as doorways (Nourbakhsh 1998) or ceiling lights (King & Weiman 1990). Geometric features such as arcs or line segments can also be used (Leonard & Durrant-Whyte 1992; Lee 1995). However, a major problem here is that the designer may not select the most appropriate landmarks for robot navigation, due to the different perception of an environment by the designer and the robot.

2. *Robot-determined landmarks.* The alternative is to avoid pre-installation of the feature categories by the system designer, instead allowing the robot to

represent its own *arbitrary* sensor patterns and to exploit whatever features are naturally present in a given environment. Possible representation schemes include statistical clustering techniques (Kurz 1996), self-organising neural networks (Nehmzow *et al.* 1991), occupancy grids (Moravec & Elfes 1985) and various techniques for matching dense sensor scans (Weiss & von Puttkamer 1995; Lu & Milios 1997b).

It was decided here to use robot-determined landmarks, since one of the aims of the thesis was to avoid pre-installation by the system designer. For topological map building, the space of possible perceptions available to the robot needs to be quantized into a set of landmark descriptions or "place signatures". The robot's location in the map can then be estimated by matching the robot's incoming sensor readings to the stored place signatures in the map. The comparative study presented in this chapter examines a number of possible matching schemes, with the aim of choosing the best mechanism for recognising landmarks on the Nomad 200 robot *FortyTwo*. This was achieved using the basic experimental procedure described in the previous chapter.

Here, landmark recognition is treated as a pattern classification problem (Duda & Hart 1973). For each of the systems examined, the robot's current sensory perception is classified according to the best matching pattern among a set of stored place signatures. The paradigm of pattern classification is particularly well-suited to the problem of landmark recognition by a mobile robot for several reasons. Firstly, this approach provides the generalisation on perception which is needed to cope with sensor noise, as discussed in section 3.2.2. Secondly, the approach avoids many of the problems associated with sonar sensors, such as cross-talk and specular reflection, because it does not rely on an accurate geometric interpretation of the robot's sonar returns. The most important point is that similar perceptions produce similar patterns, irrespective of the actual range-finder readings themselves.

6.1.1 Related Work

Gutmann & Schlegel (1996) examined three algorithms for matching 2D laser scans in an indoor environment, namely the Cox algorithm (Cox 1991), which works by assigning points in the scans to line segments and then matching the segments; the cross-correlation function (Weiss & von Puttkamer 1995), which matches stochastic histograms derived from the points in the scans; and the *idc* algorithm (Lu & Milios 1997b), which carries out a point-to-point correspondence to calculate the scan alignment. The algorithms were extended to estimate an error covariance matrix (an ellipse) as well as an (x, y) coordinate for the mobile robot. The authors found that the all of the methods provided accurate localisation in two small-scale environments, namely a square room and a corridor. However, the extended *idc* algorithm failed to produce an appropriate covariance matrix for the corridor environment, and the Cox algorithm could not be used in a non-polygonal environment.

Gutmann *et al.* (1998) conducted an experimental comparison of two self-localisation methods, namely Markov localisation, which was used to maintain a probability distribution over a grid of possible locations (Burgard *et al.* 1996), and a combination of laser scan matching and Kalman filtering (Lu & Milios 1997b; Gutmann & Schlegel 1996). The results showed that Kalman filtering produced more accurate position estimates, but Markov localisation was more robust in the presence of sensor noise or perceptual ambiguity.

Thrun (1998a) developed a localisation system in which a mobile robot learned to choose its own landmarks by training a neural network to minimise the expected deviation between the estimated and true location of the robot. An experimental comparison of the system with two other systems based on designer-determined landmarks, namely ceiling lights in King & Weiman (1990) and doors in Simmons & Koenig (1995), was then conducted. The systems were evaluated by measuring the percentage reduction in absolute error; this meant that different results could be compared in the same scale, irrespective of the magnitude of the actual error. A drawback of this landmark learning system is that accurate *a priori* position information is required for training the network, and training times can be very large (up to 12 hours on a Pentium Pro to learn a route of 89 m).

6.2 Landmark Recognition Systems Investigated

6.2.1 Restricted Coulomb Energy (RCE) Classifier

The first approach considered was one of the clustering mechanisms used by Kurz (1996), which is a simplified version of the Restricted Coulomb Energy

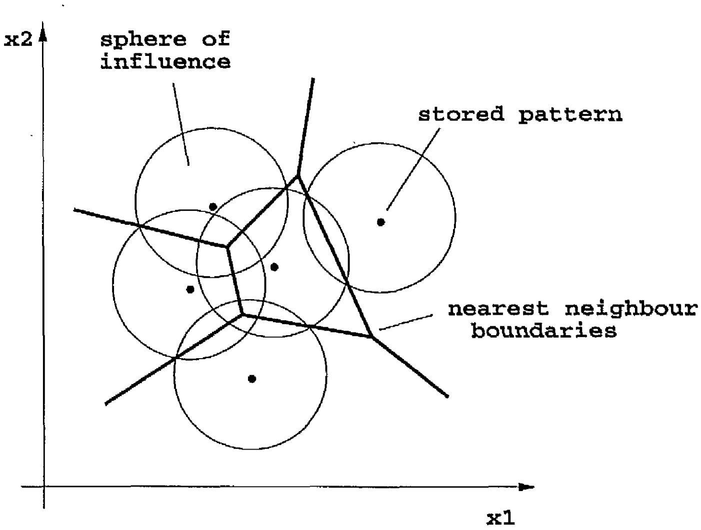

Figure 6.1: RCE classifier (example in 2 dimensions). A new sensor pattern is created if the input pattern fails to lie within a fixed threshold of any of the stored patterns. Otherwise, the sensory input is classified according to the nearest stored pattern. Adapted from Kurz (1996).

network (Reily *et al.* 1982). In this system, the robot's sonar readings are classified according to the nearest neighbour among a set of stored sensor patterns. Each pattern is represented by a vector taken directly from the raw sensor readings. Training the RCE classifier involves determining the stored patterns. A new pattern is stored if the sensory input failed to lie within the "sphere of influence" of any of the existing stored patterns (see figure 6.1).

In the implementation used here, all of the stored patterns were provided with influence spheres of the same size, as in Kurz (1996). The input patterns were normalised, and the dot product was used to compare vectors. A fixed threshold value was used to decide whether a new stored pattern should be added, i.e., to determine the size of the clusters. If the similarity between the input and the nearest stored pattern exceeded this value, then the input pattern was assigned to that particular class, otherwise a new pattern was created from the input data.

6.2.2 Adaptive Resonance Theory (ART2) Classifier

Several authors (Racz & Dubrawski 1995; Balkenius & Kopp 1996; Duckett & Nehmzow 1997a) have considered self-localisation using neural networks based on Adaptive Resonance Theory (Grossberg 1988). The basic ART architecture

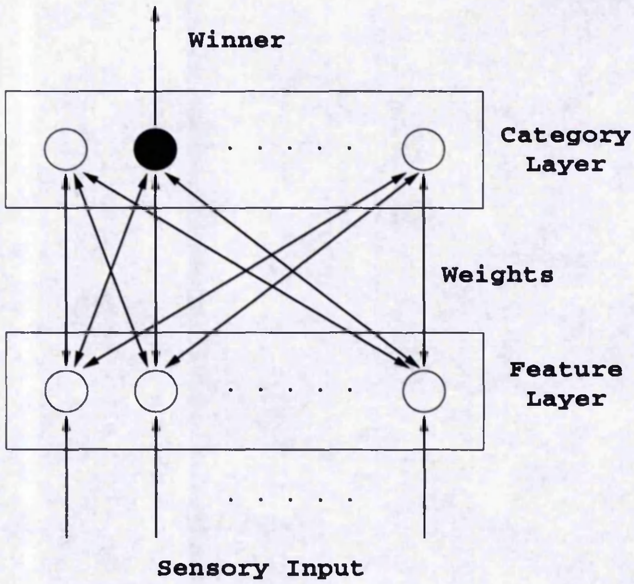

Figure 6.2: Basic ART architecture. Adapted from Racz & Dubrawski (1995).

consists of two fully connected layers of units (figure 6.2). There are two sets of weights between the layers, corresponding to feedforward and feedback connections. A "winner-takes-all" criterion is used during the feedforward phase, and a similarity criterion is used to accept or reject the resulting categorisation in the feedback phase. During training, the weights for the winning unit are adapted to be more similar to the sensory input.

The principal difference between ART and feedforward classifiers such as RCE is that with ART there are different criteria for winning in the two different phases, so that the best matching unit in the feedforward phase may be rejected in the subsequent feedback phase. On presentation of a sensor pattern at the feature layer, the feedforward weights are used to determine the best matching unit in the category layer. The response from the category layer is then compared to the actual input through the feedback weights. If the similarity between the sensory input and the stored pattern in the feedback weights exceeds a predetermined threshold, known as the "vigilance", then both sets of stored weights for the winning node are modified to be more similar to the input. Otherwise a "reset" occurs, wherein the responding unit in the category layer is disabled, and the network searches for another node to match the input pattern. If none of the stored patterns is similar enough to the input, then the weights for a new category unit are initialised from the sensory input.

The different criteria for winning in the two phases are used to implement the *self-scaling property*, which prevents any pattern that is a subset of another from being classified in the same category (see Grossberg (1988) for details). Grossberg calls this "the discovery of critical features in a context sensitive manner". The motivation for including the ART network in this study was to discover whether this property would offer any advantages for landmark recognition on a mobile robot.

In the ART2 implementation used here, an additional layer of preprocessing units was added to reduce noise and enhance contrast in the input patterns, as in Carpenter & Grossberg (1987) — see Duckett & Nehmzow (1996) for full details. A number of different parameters had to be predetermined, the most critical being the vigilance parameter, which determines the sensitivity to perceptual detail, i.e., the number of stored patterns. In this study, the vigilance was determined by experiment, and the other parameters were set by default to the values in Carpenter & Grossberg (1987).

6.2.3 Growing Cell Structures (GCS) Classifier

A number of authors (Nehmzow *et al.* 1991; Kurz 1996; Janet *et al.* 1995; Morellas *et al.* 1995) have proposed self-localisation systems based on self-organising feature maps as developed by Kohonen (1993). In this approach, the sensory input is classified according to its nearest neighbour among a network of stored patterns. Like the ART network, the weights of the best matching unit are adapted to be more similar to the sensory input during training. In addition, the other patterns which are directly connected to the best matching unit are also adapted by a smaller amount. This has the effect of preserving neighbourhood relations in the input data, whereby similar input patterns produce similar responses. As a result, the distribution of the stored patterns in a trained Kohonen network is found to reflect the underlying distribution of the input data (Kohonen 1993).

However, the structure and size (number of stored patterns) of the Kohonen network have to be fixed in advance by the designer, which means that the system cannot be used to map arbitrary, unknown environments. To overcome this problem, Fritzke (1994) developed a growing self-organising network which can store an arbitrary number of patterns depending on the amount of training data presented to the network. Like the Kohonen network, the stored patterns are topologically connected according to their similarity, and the neighbouring patterns of the best matching unit are adapted during training. In addition, at regular intervals, a new pattern is inserted into the most adapted region of the network, and the network topology is modified accordingly — see Fritzke (1994) for full details.

6.2.4 Occupancy Grid Classifier

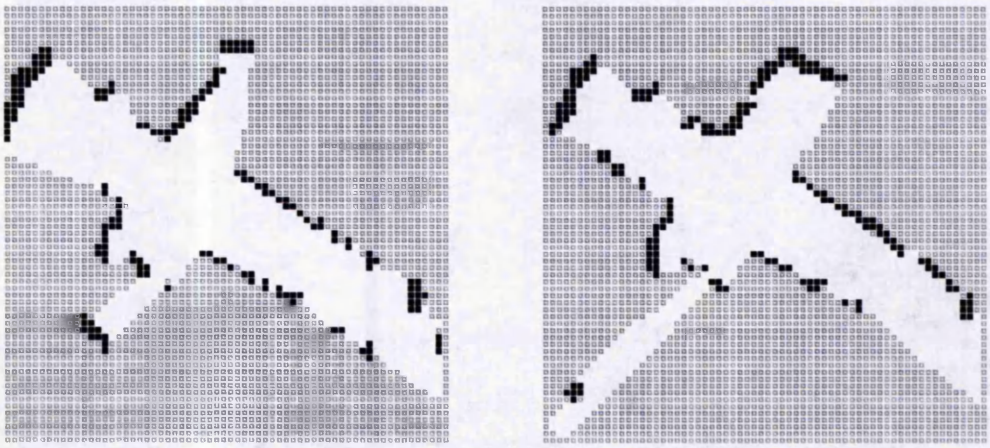

Figure 6.3: Example occupancy grids. In the occupancy grid classifier investigated (Yamauchi & Langley 1997), grids of 64 × 64 cells were used, where each cell represents an area of 15 cm × 15 cm.

Courtney & Jain (1994) and Yamauchi & Langley (1997) both considered mobile robot localisation by classification of occupancy grids. The environment is represented by a set of *local* grids, each corresponding to a small part of the environment observed from a single viewpoint, so a global metric map is not required. This means that odometry is not required to construct the actual grid models, and that the grids do not need to be geometrically accurate — as with the other classifiers, the important factor is that similar perceptions produce similar grid patterns. Therefore, this approach avoids many of the problems associated with grid models.

For this study, the system developed by Yamauchi & Langley (1997) was considered. In this approach, a recognition grid is first constructed from the robot's immediate sensor readings. Localisation then consists of matching the recognition grid with a set of previously stored grids (figure 6.3), using a hill climbing procedure to search the space of possible rotations and translations between the recognition grid and each stored grid. Here, the recognition grid is translated and rotated to find the best match with the stored grid, using an evaluation function to assess the quality of the match. The best matching grid is then used to determine the location of the robot.

This system requires an external procedure to decide when to add new grid patterns to the map. In these experiments, a new grid was added every time the robot had travelled by more than a prespecified distance (1.5 m, as in Yamauchi & Beer (1996)) from the position of the nearest stored grid in the map. The position estimates were obtained from the retrospectively corrected odometry trace used for measuring the "true" location of the robot (see section 4.3.3). (Obviously, for autonomous map building, this information must come from some other source, described later in chapter 8.) The self-orientation component of the occupancy grid classifier was disabled here, using the compass sense instead, in order to enable a fair comparison between systems.

6.2.5 Nearest Neighbour Classifier

The inclusion of the occupancy grid classifier in the study introduces an artefact into the experimental comparison, namely the external procedure used to determine when to add new grids patterns. In order to investigate the effect of this procedure, a fifth classifier mechanism was considered. This system is similar to the RCE classifier, but differs in one important respect. As in RCE, the current sensory input is classified according to its nearest neighbour amongst a set of normalised sensor patterns. Classification is decided by normalising the robot's current vector of sonar readings, and using the dot product to determine the nearest stored pattern. However, unlike the RCE classifier, this system uses *a priori* position information (from the location recording mechanism in these experiments) to decide when to add new patterns to the robot's map. In order to facilitate direct comparison with the occupancy grid classifier, exactly the same sensor readings were used for map building as in that system, i.e., new sensor patterns were added to the map at 1.5 m intervals.

	Description	Approx. Size	Route Length	Data Points	N_L	N_R
A	T-shaped hallway	16 m × 13 m	54 m	623	5	16
B	Conference room	16 m × 11 m	49 m	668	6	25
C	L-shaped corridor	34 m × 33 m	147 m	854	12	42
D	Long straight corridor	53 m × 3 m	111 m	645	9	32

Table 6.1: Characterisation of environments. N_L denotes the number of location bins, and N_R the average number of responses used in the calculation of the uncertainty coefficient (see section 5.2). The number of data points used for performance evaluation is also indicated (see also figure 6.5).

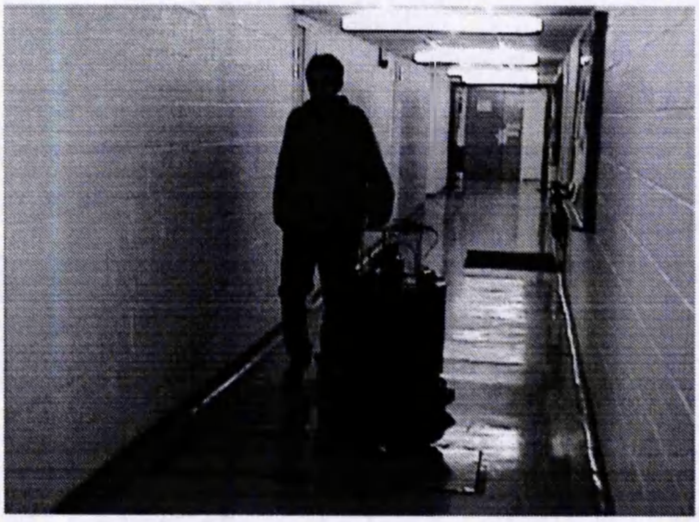

Figure 6.4: Corridor environment. This is a busy public area containing few distinctive landmarks (environment C in table 6.1).

6.3 Experimental Procedure

Data was collected by the Nomad 200 in four environments in the computer building at Manchester University (table 6.1). Wall-following was used for exploration, stopping at 0.50 m intervals to take detailed scans consisting of 144 sonar readings, as described in section 4.5. The environments were chosen to test the different systems under a variety of conditions, including very high levels of perceptual aliasing, specular reflection and cross-talk (see figure 6.4). Environment D is an extreme case; this consisted of a very long corridor with few distinctive features. All of these environments were subject to unpredictable variations in the sensor data, for example, due to people walking past the robot or doors being opened and closed.

The localisation quality obtained by the different systems was then assessed with the basic experimental procedure described in section 5.3, using the first lap of recorded robot data for landmark learning and the remaining laps for testing. Following the heuristics introduced in section 5.3.2, location bins of size 6 m $\times$ 6 m were used for performance evaluation (see figure 6.5). This bin size was chosen because (1) we are primarily interested here in the problem of global localisation (recall that the maximum sensor range of the robot is 6.5 m), and (2) the systems under comparison are very different, so the number of bins N_L should be smaller than the number of responses N_R in order to reduce the impact of quantization effects.

Initial experimentation revealed that the performance of the self-organising classifiers (RCE, ART2, GCS) was highly dependent on the parameter values used to determine the number of stored patterns. This was especially true for the RCE classifier, where the optimal value of this parameter was different for each environment; the value which produced the best performance in one environment could lead to worse performance in another. In addition, while the RCE and ART2 classifiers were capable of one-shot learning, it was found that the GCS network needed to be presented with the training data a large number of times (100 in these experiments) to obtain a stable clustering of the sensory input.

Another problem found with all of these mechanisms was that better performance could generally be obtained by configuring the system parameters to produce a large number of stored patterns. This meant that the classifier had effectively memorised the training data rather than providing a generalisation on perception. This generalisation is essential if the robot is to be capable of navigating freely rather than just following a fixed route, as in these experiments, and also to make navigation computationally tractable.

Some policy was therefore needed to facilitate an objective comparison between systems. In these experiments, the systems were configured as closely as possible to store the same number of patterns, and therefore to produce the same number of responses in each environment (see also section 5.3.2 on the choice of experimental parameters). For the occupancy grid and nearest neighbour classifiers, the number of stored patterns was determined using the retrospectively corrected odometry trace (section 4.3.3), adding new patterns to the robot's map[1] at 1.5 m intervals. The parameters of the other classifiers were then adjusted to yield the same number of responses in each of the different environments[2], as shown in table 6.1.

6.4 Results

Environment	**RCE**	**ART2**	**NstNbr**	**GCS**	**OccGrd**
A	0.554	0.573	0.650	0.719	0.732
B	0.552	0.669	0.715	0.770	0.879
C	0.462	0.538	0.551	0.502	0.644
D	0.220	0.265	0.350	0.340	0.487
Mean, μ	0.447	0.509	0.567	0.582	0.686
Cost, C	t	$218t$	t	t	$13051t$

Table 6.2: Localisation quality $U(L \mid R)$ for the RCE classifier, ART2 classifier, nearest neighbour classifier (NstNbr), growing cell structures (GCS) and occupancy grid classifier (OccGrd) in environments A to D; the mean value μ over the four environments; and the mean cost per match per landmark C, where $t = 1.8 \times 10^{-5}s$ as measured on a Sparcstation 20.

The results of the comparative study are shown in table 6.2. A statistical test was also performed to evaluate the significance of the comparative measures of

[1]Here, a "map" is taken to mean a set of recognisable locations, following Lee's taxonomy (1995, p. 33) — see section 3.3.1.

[2]For RCE, the threshold value $\gamma_A = 0.960, \gamma_B = 0.860, \gamma_C = 0.974, \gamma_D = 0.984$ was used for environments A to D. For ART2, parameter values $a = 5.0, b = 5.0, c = 0.225, d = 0.8, e = 0.001, \theta = 0.1$ were used throughout; for environments A to D, $\rho_A = 0.910, \rho_B = 0.918, \rho_C = 0.945, \rho_D = 0.953$. For GCS, parameter values $k = 6, \epsilon_b = 0.06, \epsilon_n = 0.002, \alpha = 1.0, \beta = 0.0005$ were used throughout; for environments A to D, $\lambda_A = 1040, \lambda_B = 500, \lambda_C = 800, \lambda_D = 840$.

p_{H_0}	**ART2**	**NstNbr**	**GCS**	**OccGrd**
RCE	0.06	0.01	0.04	0.01
ART2		0.04	0.16	0.01
NstNbr			0.60	0.01
GCS				0.05

Table 6.3: Paired Student's t-test results for the comparative study. Each pair of systems in table 6.2 was compared in turn, computing the probability of obtaining these results assuming the null hypothesis H_0 that their performance U is really the same over the 4 different environments.

localisation quality $U(L \mid R)$. This consisted of a pairwise comparison of the systems to test the null hypothesis H_0, for each pair of systems, that the results are really the same over environments A to D. Given the large differences between the environments, it would be meaningless to conduct a standard Student's $t-$test to compare the mean localisation quality μ of both systems. Instead, Student's t-test for *paired samples* was applied (Press *et al.* 1992, p. 618). The four data points for environments A to D in table 6.2 for each pair of systems were paired by environment, and the probability of obtaining these results assuming the null hypothesis p_{H_0} was computed. The probability values in table 6.3 indicate significant differences between all of the systems ($p_{H_0} \preceq 0.05$), except in the comparison of the nearest neighbour and GCS classifiers, where $p_{H_0} = 0.60$. There is a slight anomaly in the comparison of ART2 and GCS, where a value of $p_{H_0} = 0.16$ was observed, though we should expect some variations given the relatively small size of the samples.

In addition, the computational efficiency of the various algorithms was measured using the time taken to localise on a SparcStation 20 (referred to as the cost C in table 6.2). This was computed as the mean time taken per match per landmark during testing (the time taken to initialise the stored patterns during training was not included). The motivation for using a SparcStation here was simply to reduce the amount of time required to conduct the comparative study; we are, of course, really interested in recognising landmarks in real-time on the Nomad 200 robot.[3]

Of the one-shot self-organising classifiers, ART2 performed better than RCE

[3]The SparcStation 20 has a Dhrystone V2.1 performance of 112.9 MIPS, compared to the Nomad 200 486 PC's performance of 39.3 MIPS. Source: Performance Database Server (Netlib 1999).

($p_{H_0} = 0.06$), indicating the usefulness of the self-scaling property for landmark recognition. Both of these systems were outperformed by the GCS network ($p_{H_0} = 0.16$), though this was only achieved after conducting a series of experiments to determine an optimal set of parameter values to maximise the value of $U(L \mid R)$. This would suggest that the GCS network might not be well suited for use in unknown environments, since *a posteriori* knowledge of the system's performance was required to obtain a set of stored patterns which best reflected the underlying distribution of the input data.

The nearest neighbour classifier achieved a similar level of performance to the GCS network ($p_{H_0} = 0.60$). This can be explained as follows. Using corrected odometry data to decide when to add new sensor patterns ensured that the distribution of the stored patterns produced a good approximation to the distribution of the input data, since the stored patterns were drawn more or less evenly from all parts of the environment. This assumes of course that such position knowledge would always be available to the robot – in some environments, for example, due to extreme wheel slippage, this might not be the case, and a self-organising mechanism such as ART2 or GCS would be better suited for landmark recognition.

The best results were obtained by the occupancy grid classifier ($p_{H_0} = 0.05$). The main difference between this system and the other systems is that it uses information about the angular displacement of the robot's sonar readings as well as the actual range-finder measurements themselves. The probabilistic method used to update occupancy grids effectively uses this information to give more weight to the robot's sonar readings where they "agree" with each other, and less weight where they disagree (Elfes 1987). However, the main disadvantage of this approach is its high computational requirements; in Yamauchi & Langley (1997), location recognition using 43 stored grids took 5 minutes on a Decstation 3100[4]. The other systems can be used to recognise locations in real-time on our Nomad 200 mobile robot, whereas the occupancy grid classifier cannot.

These results strongly influenced the design of the self-localisation system described in the next chapter. The motivation was to obtain the level of performance (and beyond) attained by the occupancy grid classifier, while using only minimal computational resources, as in the simpler mechanisms for landmark identification.

[4]The Decstation 3100 has a Dhrystone V2.1 performance of 13.4 MIPS (Netlib 1999).

6.5 Discussion

To achieve reliable self-localisation over realistic distances, a mobile robot must depend on its ability to recognise places using landmarks rather than by dead reckoning alone. This chapter considered approaches for landmark recognition in which the robot determines its own landmark categories, in order to allow the robot to adapt its internal representations to the features which are naturally present in a given environment. Quantitative performance measures were used to compare five different algorithms under a realistic set of operating conditions, namely a mobile robot traversing a series of unmodified, real world environments. The results are summarised in tables 6.2 and 6.3.

All of the landmark recognition mechanisms investigated need to maintain a balance between over-generalisation and over-fitting to the sensory input. For example, in the one-shot classifier mechanisms investigated, some distance metric is used to determine when to add new patterns to the robot's map. If this metric is set too high, the classifier will over-generalise on the data and not recognise sufficient features for useful navigation. However, if the metric is too low, the robot will become too dependent on perceptual detail, and be incapable of interpolating between known environmental features.

For the RCE classifier, it was found that the simple distance threshold applied to the sensor data was too sensitive to be generally useful in unknown environments; in the study presented, it was only possible to choose a suitable parameter value for a particular environment by *a posteriori* examination of the robot's performance in that environment. ART2 was found to be better suited for navigation in unknown environments, due to its internal mechanism for distinguishing sensor patterns which share some common features but differ in others. However, both of these mechanisms would suffer from the problem of adding spurious landmark categories in the presence of severe sensor noise or unpredictable variations in the sensor data caused by other inhabitants of the environment.

The GCS network was found to achieve the highest level of performance among the self-organising clustering mechanisms by replicating the underlying distribution of the pre-recorded sensor data for a given environment. However, again this was only made possible by retrospective examination of the robot's performance. The strategy of adding new nodes to the network at regular time intervals would make little sense in an unknown environment, where there would be no obvious stopping criterion for determining when the network had reached an appropriate balance between generalisation and over-fitting of the sensory input. It is unclear therefore whether this network offers any real advantage over the Kohonen network for the purpose of landmark recognition.

It was found that the best method for obtaining a representative clustering of the robot's sensor data was by using prior position information to determine when to create new landmark categories. In particular, the best performance was obtained by Yamauchi and Langley's method of matching local occupancy grids (1997). However, this was achieved at an extremely high computational cost, preventing this system from being used for location recognition in real-time on the Nomad 200 robot without using external processors. In addition, despite the use of detailed sensory information, this approach still does not overcome the problem of perceptual aliasing, and would therefore be unable to relocalise a navigating robot under global uncertainty.

6.6 Concluding Remarks

This chapter presented a comparative study of previous approaches to landmark recognition for mobile robot navigation, including replication of the work of other researchers. In the next chapter, these results are used to guide the development of a new landmark recognition mechanism for the Nomad 200 robot. For a different robot, different results might be obtained, for example, due to different sensory apparatus. However, the important point is that an objective set of criteria was developed and used to determine the best mechanism for one particular platform, rather than relying on the intuition of the system designer.

In the experiments presented in this chapter, the various mechanisms were assessed using a "winner-takes-all" criterion for landmark recognition. However, as will be explored in the next chapter, it is often desirable to represent situations in which the robot is unsure of its exact position, and to assign appropriate levels of belief to many possible locations. This means using a multimodal location model, rather than a single "winner-takes-all" position estimate, based on the degree of match between the current sensor readings and each of the stored landmark descriptions.

None of the landmark recognition systems investigated were able to overcome the problem of perceptual aliasing, and therefore cannot be used alone to localise the robot under global uncertainty. In environments of any real complexity, there can be no guarantee of uniquely identifiable landmarks; there may often be many places which share the same perceptual signature, particularly in the uniform corridors and hallways which are common to many office buildings. The results confirmed that reliable location recognition could not be achieved using the robot's current sensory perception alone. The next chapter therefore addresses this problem by using a *sequence* of sensory perceptions to identify locations.

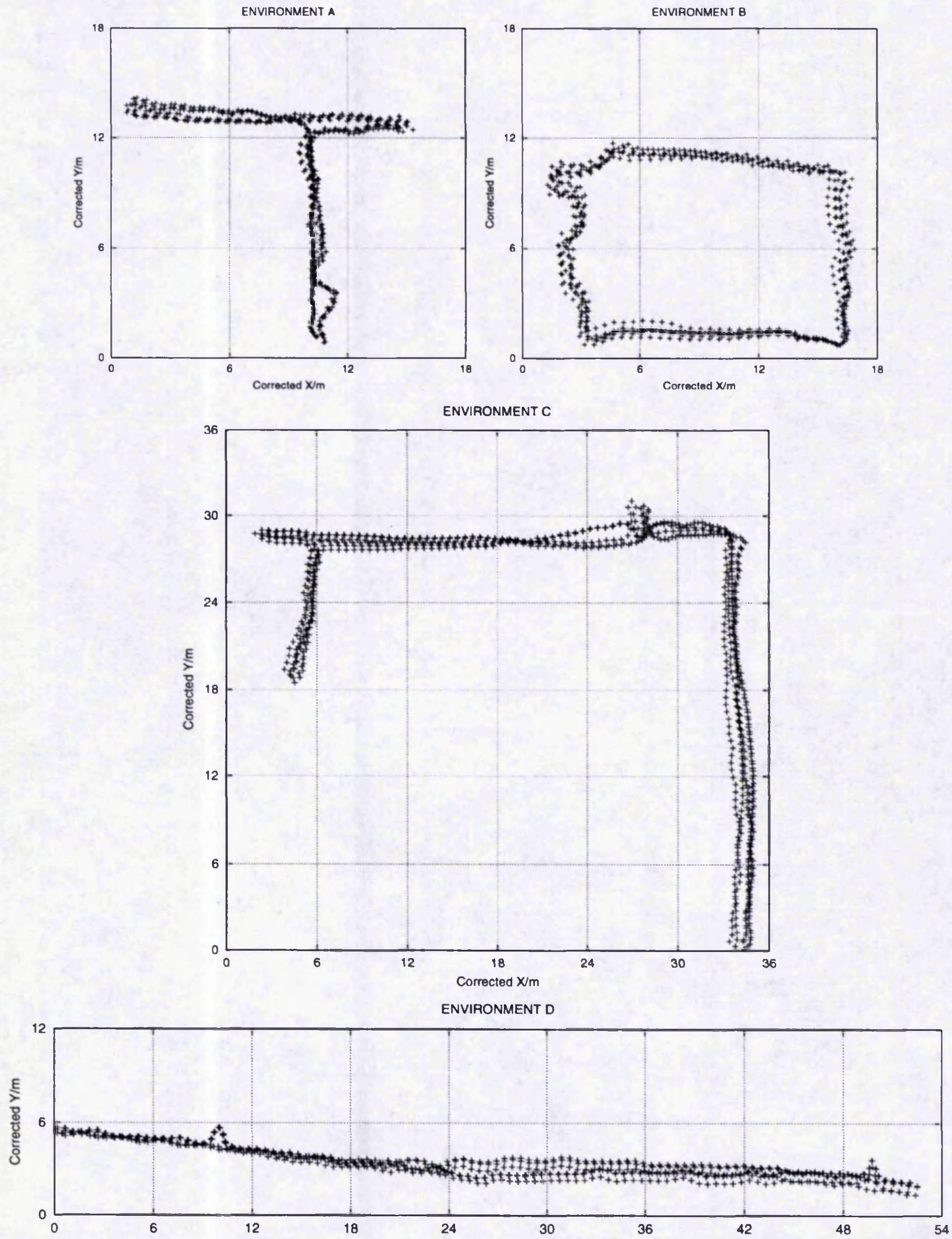

Figure 6.5: Data points used for performance evaluation. The corrected odometer positions for the test data are plotted for the four different environments (see also table 6.1). The dotted grid shows the location bins used for calculating the uncertainty coefficient.

Chapter 7

So where am I?

About this chapter. A key question addressed by the thesis is that of global localisation, using a previously obtained map for the task of identifying places. A complete self-localisation system is presented which accumulates both exteroceptive and proprioceptive sensory evidence over time, allowing the robot to recover its position even after becoming lost.

7.1 Introduction

This chapter describes the development of a new self-localisation system for the Nomad 200, based on the results of the previous chapter. The topic of mobile robot self-localisation is often divided into two related sub-problems:

1. *Global Localisation.* This involves being able to localise under global uncertainty. An example is the "lost robot problem", or the "kidnapped robot problem" (Engelson 1994), where the robot has no prior information about its true location in the map. The uncertainty is typically represented over a *discrete* state space, corresponding to possible robot locations in the map (see e.g., Hidden Markov Models (Koenig *et al.* 1996)), and the task of localisation involves identifying the most likely state or location occupied by the robot.

2. *Position Tracking.* This involves being able to accurately estimate the position of the robot based on approximate knowledge of the robot's global location in the map. Here, the uncertainty is typically represented over a *continuous* state space. The most common solution is the Kalman filter

(Gelb 1974; Maybeck 1990), where the robot's location model consists of a Gaussian probability density function, the mean and variance of which represent the most likely Cartesian coordinate for the robot and the uncertainty in this estimate.

This chapter is concerned primarily with the problem of global localisation. However, the research presented shows that overall self-localisation performance can be improved by *combining* mechanisms for global localisation and position tracking. The result is a unified solution to the two problems, based on a topological map augmented with metric information. The new system solves the global localisation problem by tracking multiple Gaussian location hypotheses over the space of possible locations in the robot's map. It solves the position tracking problem by calculating the most likely (x, y) coordinate for *each* of the possible places.

For landmark recognition, I developed a new method of matching local occupancy grids, which overcomes the high computational requirements of the grid matching algorithm described in the previous chapter. In order to overcome the problems of perceptual aliasing and misclassification errors due to sensor noise, I developed an iterative self-localisation algorithm, which works by accumulating sensory evidence over time as the robot explores its environment.

A quantitative, experimental evaluation of the new self-localisation system was conducted, in which global localisation performance was assessed over a series of middle-scale environments. This included a set of controlled experiments to determine the influence of different sub-components on the overall system; these "lesion experiments" were conducted by removing the sub-components in turn and assessing the change in performance.

7.1.1 Related Work

A wide variety of localisation methods have been proposed for mobile robots, and a number of laboratory prototypes have been developed (Borenstein *et al.* 1996). However, relatively few of these systems have been tested in middle-scale environments, generally consisting of enclosed areas within office buildings such as corridors and hallways.

As described earlier, Yamauchi & Langley (1997) developed a localisation system based on classification of local occupancy grids. In the previous chapter, it was shown that this method of matching occupancy grids produced better self-localisation performance than a number of alternative mechanisms such as self-organising neural networks. In addition to performing place recognition, this approach exploits the spatial information content of the robot's sensor readings, enabling the robot to estimate its relative position within the local grid model, without using odometry.

A related approach is that of Weiss & von Puttkamer (1995), using cross-correlation of laser scans to identify both the current place occupied by the robot and the robot's position within that particular place. In the latter approach, however, the scans are reduced to histograms before matching takes place, then the techniques for scan matching developed by Hinkel & Knieriemen (1988) are applied. Angle histograms are first convolved to address the problem of self-orientation, then x and y histograms are convolved to determine the robot's position. Histograms were applied in this chapter to overcome the high cost of the matching local occupancy grids.

However, neither of the above approaches is guaranteed to solve the global localisation problem in environments of any real complexity, due to the problem of perceptual aliasing. A number of methods have been proposed for resolving perceptual ambiguity by using a sequence of sensory perceptions over time. Perhaps the most popular method in recent years is the probabilistic approach known as Markov localisation, for example, see (Burgard *et al.* 1998b; Cassandra *et al.* 1996; Fox *et al.* 1998; Hertzberg & Kirchner 1997; Simmons & Koenig 1995), etc., which can be applied to either topological or grid-based maps.

For topological maps, the main paradigm for probabilistic localisation is that of Hidden Markov Models (HMMs), and their extension to Partially-Observable Markov Decision Process (POMDP) models (Simmons & Koenig 1995; Cassandra *et al.* 1996; Hertzberg & Kirchner 1997). Here, the robot maintains a probability distribution over a set of discrete locations, known as the robot's *belief* about its possible location. Similarly, possible landmarks and possible actions are typically defined according to a set of designer-determined categories. For example, possible landmarks might be "doors", "junctions", etc., and possible actions might be "Go North", "Go West", etc. The localisation system presented in this chapter differs from other topological approaches in that possible location estimates are represented by continuous valued Cartesian coordinates, actions are described by arbitrary displacements within Cartesian space, and landmarks are defined by arbitrary sensor patterns.

Probabilistic localisation methods have also been applied to high resolution, grid-based maps. This approach has the advantage of high accuracy, but typically requires a great deal of computational resources. An elegant solution to this problem is provided by Burgard *et al.* (1998b), where a variable resolution mapping strategy is used to trade off global uncertainty against accurate positioning. The approach uses Markov localisation to identify possible sub-areas of the whole state space in which the robot might be located, then position tracking is carried out only on these sub-areas, "zooming in" to a higher level of resolution when the robot has a high degree of certainty in its location. The localisation system presented in this chapter differs in that the full state space in the robot's map is always searched, and efficient matching algorithms are used to overcome the problem of limited computational resources.

More recently, a number of successful self-localisation systems have applied Monte Carlo methods (also known as the condensation algorithm), in which the underlying probability density function for the robot location is approximated by a large set of "samples" or particles (Dellaert *et al.* 1999; Thrun *et al.* 2000; Jensfelt *et al.* 2000). During localisation, these methods enumerate random weighted samples which estimate the posterior distribution by taking into account the previous samples and new sensor information. However, these approaches require a high amount of computation to work in large environments, as they suffer from poor degradation to small sample sets, and cannot be guaranteed to recover from becoming lost once the particles have converged around one location in the map (Thrun *et al.* 2000).

7.2 New Self-Localisation System

7.2.1 Representation

Environment Model

The robot's environment model consists of a set of N stored places, the centre of each place i being associated with a Cartesian coordinate (x_i, y_i). In the localisation experiments presented in this chapter, these coordinates were obtained using the *a posteriori* odometry correction technique described in section 4.3.3. The subject of autonomous map acquisition by the robot is covered in chapters

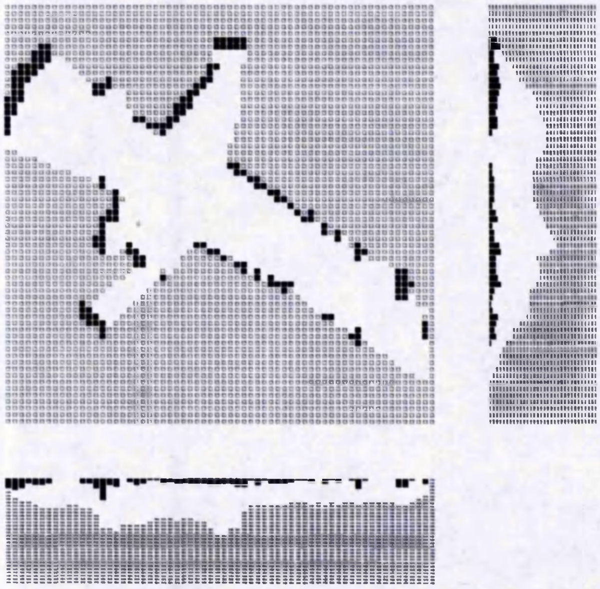

Figure 7.1: Example occupancy grid and histograms. Occupied cells are shown in black, empty cells in white and unknown cells in grey. A separate pair of histograms is used to represent each individual place in the robot's map.

8 and 9.

Landmark information is also attached to each of the places as follows. Firstly, the robot takes a detailed sonar scan at its current location and a local occupancy grid consisting of 64×64 cells is then constructed, where the robot location is taken as the centre of the grid, as in Yamauchi & Langley (1997). However, in the system presented here, the occupancy grids themselves are not stored or matched. Instead, each grid is reduced to a pair of histograms (one in x direction, and one in y direction), which is then used as a stored signature for that *place* in the robot's map, as shown in figure 7.1. Angle histograms are not required here, because the compass sense described previously in section 4.2 was used to remove the problem of self-orientation.

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

$$State(c_{xy}) = \begin{cases} O & \text{if } p(c_{xy}) > 0.5 \\ U & \text{if } p(c_{xy}) = 0.5 \\ E & \text{if } p(c_{xy}) < 0.5 \end{cases} \tag{7.1}$$

where $p(c_{xy})$ refers to probability of occupancy for the cell at column x and row y. These probabilities were obtained using the standard method for updating occupancy grids developed by Moravec & Elfes (1985). One histogram is then derived by adding up the total number of occupied, empty and unknown cells in each of the 64 columns, and the other by adding up the totals for each of the 64 rows. Note that the probability $p(c_{xy}) = 0.5$ is the default probability used to initialise the cells; this value usually indicates that the cell has not yet been updated because the robot's view of the corresponding location is occluded by some other object.

Location Model

The robot's location model consists of a set of competing location hypotheses $\mathcal{H} = \{h_1, ..., h_N\}$, one for each place i. A probability distribution $\mathcal{P} = \{p(h_1), p(h_2), ..., p(h_N)\}$ is associated with set $\mathcal{H}$, reflecting the robot's belief in each of the hypotheses being its true location. Each location hypothesis consists of a Cartesian coordinate (x_{h_i}, y_{h_i}), and a variance v_{h_i} which is used for position tracking. Thus, each hypothesis is represented by a separate Gaussian density function as in the circular noise model described in section 4.3.2. The initial values for the probability distribution and the coordinates of the location hypotheses are obtained using the histogram matching procedure described in section 7.2.2, and updated using the iterative localisation algorithm described in section 7.2.3.

7.2.2 Landmark Recognition

To begin localisation, the robot takes a new sonar scan. Again, the resulting occupancy grid is processed to produce a pair of histograms. These histograms are then convolved with the corresponding stored histograms for all of the places in the robot's map. For each stored place i, the matching procedure yields two useful quantities:

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

2. The most likely offset (r_{x_i}, r_{y_i}) of the robot in Cartesian space from the centre of the stored grid pattern, i.e., the position in which the sonar scan for that place was originally taken.

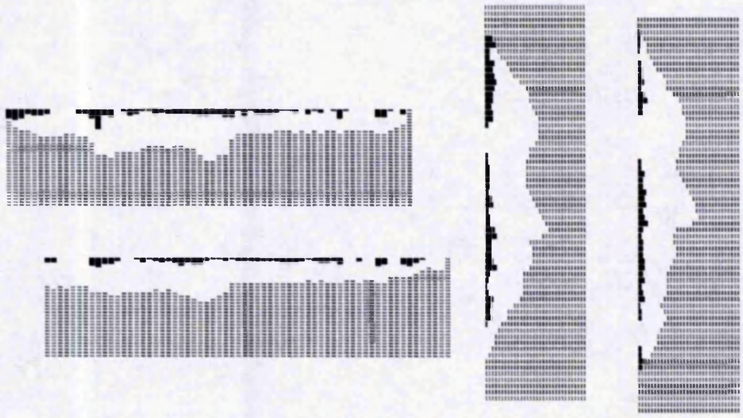

Figure 7.2: Matching the x and y histograms. The new histograms are convolved with the stored histograms for each place in the robot's map to find the best match.

The first quantity is derived from the product of two separate metrics; one obtained by convolving the current and stored x histograms, and the other by convolving the respective y histograms (figure 7.2). The strength of the match between two histograms T^a and T^b is calculated using the following evaluation function

$$Match(T^a, T^b) = \frac{1}{w} \sum_j \left[min(O_j^a, O_j^b) + min(E_j^a, E_j^b) + min(U_j^a, U_j^b) \right], \qquad (7.2)$$

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 $w = 64 \times 64$ is a normalising constant such that $0 \leq Match() \leq 1$. In the convolution, the stored histogram is kept stationary and the recognition histogram is translated against it, using the above function to calculate the best match over the 64 elements of the stored histogram. Any non-overlapping elements in the recognition histogram due to the translation are assumed to consist entirely of unknown cells.

The likelihood $L(S \mid h_i)$ is then calculated from the best match scores as

$$L(S \mid h_i) \propto M_x^{i^*} \times M_y^{i^*}, \qquad (7.3)$$

where M_x^{i*} refers to the value of $Match()$ produced by the best matching alignment of x histograms for place i.

The most likely displacement (r_{x_i}, r_{y_i}) of the robot from the centre of each place i is obtained by multiplying the translations for the x and y histograms by the dimensions of one grid cell (i.e., 15 cm × 15 cm). The coordinates for each h_i are then calculated as

$$x_{h_i} = x_i + r_{x_i}, \tag{7.4}$$

$$y_{h_i} = y_i + r_{y_i}, \tag{7.5}$$

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

Finally, to obtain an estimate of the measurement error in the scan matching, the following heuristic function was used:

$$v_{h_i} = \frac{k_1}{(M_x^{i*} - \bar{M}_x^i)^2} + \frac{k_2}{(M_y^{i*} - \bar{M}_y^i)^2}, \tag{7.6}$$

where $\bar{M}_x^i$ refers to the mean value of $Match()$ in the convolution of x histograms for place i, and the constants $k_1 = k_2 = 2.5$ m^2 in these experiments.

7.2.3 Accumulation of Sensory Evidence

After carrying out the above matching process, the place which yielded the highest match score could be taken as the winner. However, this simple "winner-takes-all" strategy, i.e., using only the current sensory input, is bound to fail in middle-scale environments due to factors such as perceptual aliasing and sensor noise. To overcome these problems, I developed the following algorithm for accumulating sensory evidence, which uses a succession of sonar scans taken from different positions over time.

A schematic diagram of the localisation algorithm is given in figure 7.3. At each iteration, the algorithm takes as input a prior set of location hypotheses $\mathcal{H} = \{h_1, h_2, ..., h_N\}$ and the corresponding probability distribution $\mathcal{P} = \{p(h_1), p(h_2), ..., p(h_N)\}$ from the previous iteration. On initialisation, sets $\mathcal{H}$ and $\mathcal{P}$ will be empty. The algorithm can be explained in the following steps.

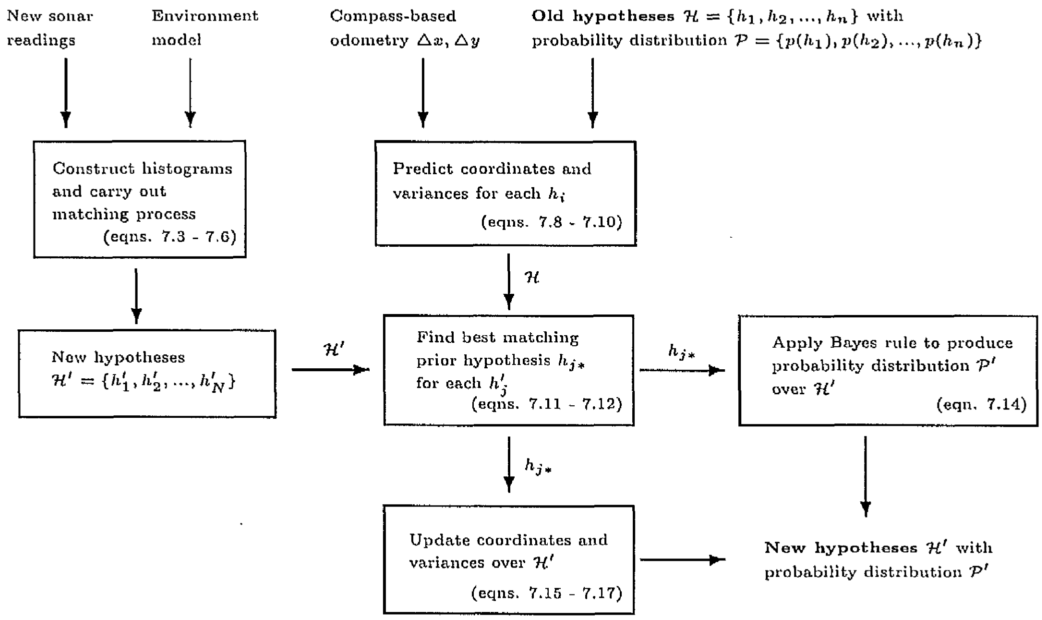

Figure 7.3: The localisation algorithm.

Initialisation

Localisation begins by taking a sonar scan and constructing a set of location hypotheses $\mathcal{H} = \{h_1, ..., h_N\}$, as described in section 7.2.2. For each of these hypotheses, the likelihood $L(S \mid h_i)$ is obtained using equation 7.3 and the coordinates (x_{h_i}, y_{h_i}) are obtained using equations 7.4 and 7.5. The initial probability distribution over $\mathcal{H}$ is then calculated using

$$p(h'_j) = \frac{L(S \mid h'_j)}{\sum_k L(S \mid h'_k)}. \tag{7.7}$$

After initialisation, localisation proceeds as follows. This algorithm is best explained as a three-step *predict–match–update* cycle, after Crowley (1995).

Predict Step

Firstly, the robot waits until it has travelled a further 0.5 m, then the coordinates (x_{h_i}, y_{h_i}) of each of the prior hypotheses h_i are translated to take into account the robot motion, using

$$x_{h_i}(t) = x_{h_i}(t-1) + \Delta x, \tag{7.8}$$

$$y_{h_i}(t) = y_{h_i}(t-1) + \triangle y, \tag{7.9}$$

where the vector $(\triangle x, \triangle y)$ refers to the robot's own displacement in Cartesian space observed since the previous iteration, using its on-line compass-based odometry (section 4.3.1).

The additional uncertainty due to the robot motion is approximated by increasing the variance v_{h_i} for each of the prior hypotheses as

$$v_{h_i}(t) = v_{h_i}(t-1) + k_3, \tag{7.10}$$

where a value of $k_3 = 1$ m^2 was used for these experiments. This constant is based on an approximate estimate of the odometry drift (given that the robot moves a constant distance between scans) plus some extra noise. This has the effect of "blurring" the density function for each of the prior hypotheses.

Match Step

The robot then takes a new sonar scan, and a second set of candidate hypotheses $\mathcal{H}' = \{h'_1, ..., h'_N\}$, is created from the new sonar information. In the algorithm presented here, exactly one hypothesis is generated for each place in the map. (Without a compass, it might be necessary to generate several hypotheses per place, corresponding to possible orientations of the robot.)

A matching process between the two sets $\mathcal{H}$ and $\mathcal{H}'$ then follows. For each new hypothesis h'_j, this attempts to find the *one* most likely equivalent prior hypotheses h_i (since only one hypothesis can actually be the "true" location of the robot). Each h'_j is therefore compared to every h_i, and the likelihood $L(h'_j \mid h_i)$ of obtaining each h'_j from each predicted h_i is calculated as

$$L(h'_j \mid h_i) \propto Gaus\left(\|(x_{h'_j}, y_{h'_j}) - (x_{h_i}(t), y_{h_i}(t))\|\right) p(h_i), \tag{7.11}$$

where the Gaussian function $Gaus(z) = e^{-\eta z^2}$ is used to model the noise in the robot's position estimates. This is weighted here by the prior probability $p(h_i)$ in order to take into account the relative "mass" of evidence afforded to that particular prior hypothesis. The constant η effectively determines the relative weighting of exteroceptive and proprioceptive sensory information in the localisation algorithm; a value of $\eta = 0.25$ was used in these experiments. For each h'_j, the best matching prior hypothesis h_{j*} is therefore defined by

$$\forall j : \forall i \neq j* : L(h'_j \mid h_{j*}) > L(h'_j \mid h_i). \tag{7.12}$$

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.

Update Step

The likelihood values $L(h'_j \mid h_{j*})$ produced by the match step are used to provide a prior probability $p_{prior}(h'_j)$ for each h'_j according to

$$p_{prior}(h'_j) = \frac{L(h'_j \mid h_{j*})}{\sum_k L(h'_k \mid h_{k*})}, \tag{7.13}$$

and a new probability distribution over $\mathcal{H}'$ is calculated using Bayes rule as

$$p_{posterior}(h'_j) = \frac{L(S \mid h'_j)p_{prior}(h'_j)}{\sum_k L(S \mid h'_k)p_{prior}(h'_k)}. \tag{7.14}$$

This step effectively combines a "sensor model" $L(S \mid h'_j)$ and a "motion model" $L(h'_j \mid h_{j*})$ for each location hypothesis h'_j.

The following equations are then used to update the Cartesian coordinates of each h'_j, taking into account the coordinates of both h'_j and h_{j*}.

$$x_{h'_j} = x_{h_{j*}} + \frac{v_{h_j*}}{v_{h_j*} + v_{h'_j}}(x_{h'_j} - x_{h_{j*}}), \tag{7.15}$$

$$y_{h'_j} = y_{h_{j*}} + \frac{v_{h_j*}}{v_{h_j*} + v_{h'_j}}(y_{h'_j} - y_{h_{j*}}), \tag{7.16}$$

$$\frac{1}{v_{h'_j}} = \frac{1}{v_{h_j*}} + \frac{1}{v_{h'_j}}. \tag{7.17}$$

The robot then continues to explore, taking a new sonar scan at 0.5 m intervals and updating its estimate of its true location by repeating the above process.

7.3 How It Works

The robot's location model can be thought of as a collection of different sized "blobs" scattered over the robot's map. The relative mass of the different blobs reflects the amount of belief assigned to each of these location "hypotheses". Each blob actually consists of a Gaussian density function, the parameters of which represent the Cartesian coordinate (the mean) and the variance for that particular position estimate. When the robot moves, the whole distribution gets shifted according to the observed robot motion; the blobs are translated across the map and "blurred" a little to reflect the added uncertainty due to odometer drift (this is the *predict* step described above).

After moving, the robot takes a new sonar scan. The landmark description (histograms) derived from the new sonar readings is compared to all of the stored place signatures in the map to produce a second collection of "blobs". At this stage, the robot has two alternate sets of location estimates, one predicted from its old location model and the observed change in odometry, and one from the currently observable landmarks. The *match* step described above tries to find a correspondence between the two sets, matching each blob in the new set to its most likely neighbour in the old set (blobs in the old set may therefore be matched to more than one hypothesis in the new set).

Finally, the *update* step merges together the two sets of location estimates, using Bayes rule to propagate the belief assigned to the different hypotheses and simple position tracking equations to merge the density functions for each pair of matched blobs. The result is a new collection of blobs which improves the robot's location model based on all of the the available information. At the next iteration, this whole distribution will be shifted and blurred, matched to a new distribution based on the landmarks perceived and updated again.

7.3.1 Relationship to the Kalman Filter

In the Kalman filter, the robot's location model is unimodal, and the probability density function evolves as a Gaussian. By contrast, the new self-localisation algorithm maintains a whole set of competing location hypotheses, each with its own Gaussian density function. In this respect, the approach can be seen as a multi-modal generalisation on the Kalman filter, because the robot's location model consists of a mixture of Gaussians — each updated by a separate filter — rather a single position estimate. However, it should be noted that the algorithm makes some approximations in its treatment of uncertainty which are modelled more accurately in many Kalman filter implementations (see Gelb (1974) on optimal linear filtering). For example, the constants k_1, k_2, k_3 and η were determined

	Description	Approx. Size	Route Length	No. Trials	N_L	N_R
A	T-shaped hallway	16 m × 13 m	54 m	623	5	16
B	Conference room	16 m × 11 m	49 m	668	6	25
C	L-shaped corridor	34 m × 33 m	147 m	854	12	42
D	Long straight corridor	53 m × 3 m	111 m	645	9	32

Table 7.1: Characterisation of environments. N_L denotes the number of location bins, and N_R the number of responses (equal here to the number of mapped locations N) used in the calculation of the uncertainty coefficient. The number of trials used for testing is also shown (this is equal to the number of data points used in the landmark recognition experiments).

largely by trial and error, so that the noise in the robot's sensors is only modelled approximately — a more rigorous approach might investigate using some statistical techniques to optimise the values of these parameters.

7.3.2 Complexity Analysis

The computational complexity of the algorithm in its current implementation would be $O(n^2)$, where n is the number of location hypotheses. However, the complexity could be reduced by an order of magnitude to $O(n)$ by restricting the search over the space of prior hypotheses $\mathcal{H}$ in the *match* step to those places lying within a prespecified distance of a given posterior hypothesis h'_j. This could be implemented using efficient data structures, e.g., by storing a linked list of nearby map locations for each of the places in the map. Then the amount of computation required for self-localisation would grow only linearly with the size of the map. In fact, the current algorithm already has a very low computational cost for maps of the scale considered in this thesis — for the biggest map of 42 places (environment C in table 7.1), one complete cycle of the new localisation algorithm takes 0.002 sec on a 600 MHz Pentium III processor.

Environ.	**Occupancy Grids**	**Histogram Matching**	**Nearest Neighbour**
A	0.732	0.806	0.650
B	0.879	0.850	0.715
C	0.644	0.632	0.551
D	0.487	0.439	0.350
Mean, μ	0.686	0.682	0.567
Cost, C	$13051t$	$31t$	t

Table 7.2: Landmark recognition performance. This shows the localisation quality $U(L \mid R)$ for each environment, the mean value μ over all 4 environments, and mean cost per match per landmark C, where $t = 1.8 \times 10^{-5}s$ as measured on a Sparcstation 20.

7.4 Experiments

7.4.1 Landmark Recognition

The new landmark recognition mechanism was assessed using the basic experimental procedure (section 5.3) and the same recorded robot data as in the previous chapter (figure 6.5). The environments used for testing are summarised once more in table 7.1. Location bins of size 6 m $\times$ 6 m were used for performance evaluation, and new places were entered into the robot's map at 1.5 m intervals. Again, the "winner-takes-all" rule was used to determine the robot's response R. The performance of the new mechanism was compared to that of the previous method for matching local occupancy grids developed by Yamauchi & Langley (1997) (section 6.2.4) and a third "baseline" method for identifying landmarks, namely the nearest neighbour classifier (section 6.2.5). Student's t-test for paired samples was used to determine the significance of the performance comparisons, as described in section 6.4.

The results given in table 7.2 indicate no significant difference in performance between histogram matching and classification of full occupancy grids ($p_{H_0} = 0.90$), and that this was achieved at a greatly reduced cost in processing requirements. For reference, both of the occupancy based methods performed significantly better than the nearest neighbour classifier ($p_{H_0} = 0.01$). Typically only a single iteration of temporal evidence accumulation, i.e., a second sonar scan, was then required for the performance of the new system to overtake that of the occupancy grid classifier, as described in the following experiment.

7.4.2 Lost Robot Experiment

To assess global localisation performance, the lost robot experiment was conducted, using the experimental procedure described in section 5.4 to calculate the uncertainty coefficient $U(L \mid R)$ against distance travelled. In addition, a further metric was introduced to assess the robot's degree of certainty or "confidence" in the location estimates produced by the new system. This was achieved by calculating the entropy of the probability distribution in the robot's location model after each iteration of localisation as

$$H(\mathcal{P}) = -\sum_i p(h_i) \ln p(h_i). \tag{7.18}$$

The lower the value of $H(\mathcal{P})$, the more confident the robot becomes in its estimated location. An equivalent measure of localisation certainty has been proposed independently by various researchers (Donnett 1993, p. 155)(Cassandra *et al.* 1996)(Fox *et al.* 1998).

In figure 7.4, the complete localisation system is compared again to the local occupancy grid classifier and the nearest neighbour classifier. Note the improvement in performance over time for the new system as the robot traversed the environment. These graphs actually represent the *worst case* performance for the different localisation algorithms, since failure to localise correctly in any single trial will reduce the value of the uncertainty coefficient.

The corresponding "confidence" levels are shown in figure 7.5. Comparison of figures 7.4 and 7.5 shows that the robot continues to become more certain of its estimated location for some distance after relocalising itself successfully. For example, in environments A and B, the entropy over the robot's location model continued to fall for about 5 m after the uncertainty coefficient reached its maximum. This effect provides the robot with a degree of "inertia" in its location model; if the robot's immediate sensory perceptions are affected by unpredictable changes such as doors being opened and closed, the "correct" location hypothesis will persist for some distance (e.g., 5 m in environments A and B) before localisation errors can occur.

7.4.3 Lesion Experiments

A further set of experiments were conducted in order to evaluate the effect on system performance of three particular mechanisms; (1) the position tracking

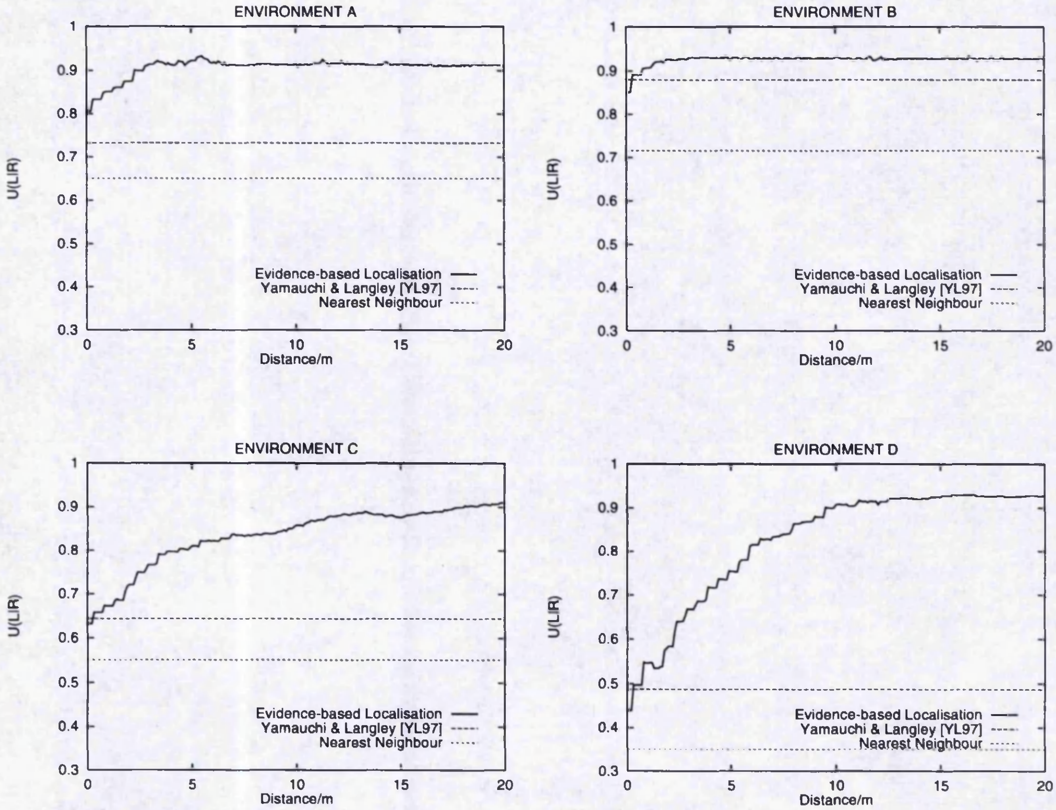

Figure 7.4: Localisation quality $U(L \mid R)$ for the lost robot using the new evidence-based localisation system, measured against the distance travelled by wall-following, compared to the full occupancy grid classifier and the nearest neighbour classifier.

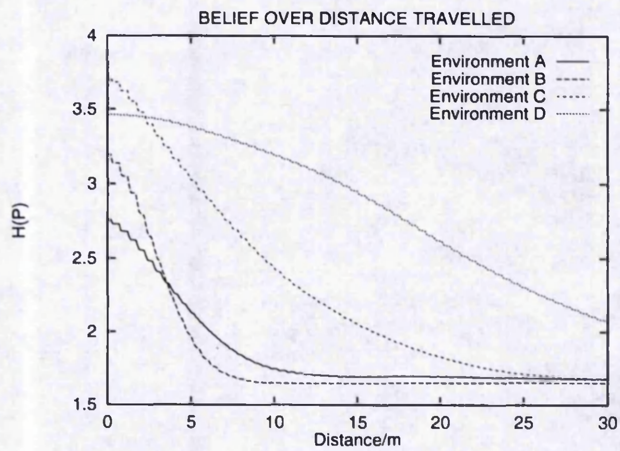

Figure 7.5: Localisation confidence $H(\mathcal{P})$ for the lost robot, measured against the distance travelled by wall-following.

equations (section 7.2.3), (2) the histogram matching mechanism (section 7.2.2) and (3) the high resolution sensing strategy used for landmark recognition (section 4.5). Here, the complete localisation system was used as a control, its performance being used as a baseline against which the following modified versions of the system were compared:

1. *The position tracking equations were disabled.* Here, instead of applying equations 7.15 to 7.17 after each iteration of the self-localisation algorithm, the coordinates of the new hypotheses $(x'_{h'_j}, y'_{h'_j})$ were left unchanged.

2. *The histogram matching mechanism was removed.* Instead, the nearest neighbour classifier (section 6.2.5) was used for landmark recognition, using the dot product to provide the likelihood values $L(S \mid h_i)$ instead of equation 7.3. This meant also that the robot could no longer estimate its most likely offset (r_{x_i}, r_{y_i}) within the current location from its current sonar readings. The offset values in equations 7.4 and 7.5 were therefore always set to zero, and a constant measurement error of $v_{h_i} = 2.5\ \text{m}^2$ was assumed instead of using equation 7.6.

3. *The high resolution sensing strategy was omitted.* The robot was allowed to use only its first 16 sonar readings for landmark recognition instead of the usual detailed scan of 144 readings. Again, the nearest neighbour classifier was used for landmark recognition, this time based on sonar signatures consisting of a normalised vector of 16 sensor readings (in other words, this particular lesion also includes lesion 2).

The results are shown in figure 7.6. Environments A and B produced similar results for the four different systems. Both environments were rich in useful landmarks; as a result, omitting the position tracking equations made little difference to overall system performance, and systems 2 and 3 were able to "catch up" with the control system after a further 10 to 15 m of travel. Comparison of systems 2 and 3 shows that extra sensory information made little difference to system performance here, the environmental cues being strong enough to be perceived using only the basic ring of 16 sonar sensors on the robot.

A different outcome was observed in environments C and D, both consisting of long, largely featureless corridors. In both environments, a clear drop in performance was observed for system 1, indicating the importance of the position

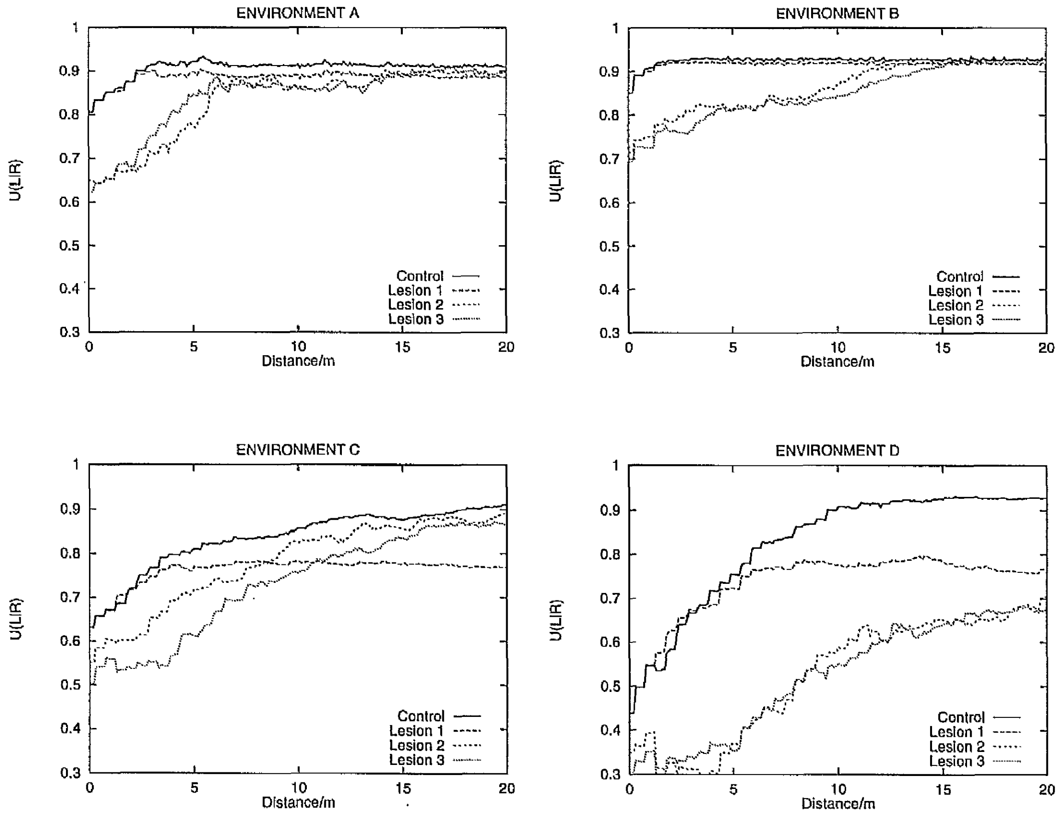

Figure 7.6: Localisation quality $U(L \mid R)$ for the lesion experiments, measured against the distance travelled by wall-following, compared to the "control" performance of the complete evidence-based localisation system.

tracking equations for processing the robot's proprioceptive sensory input and localising the robot as it traversed the corridors. The biggest difference in performance between systems 2 and 3 was noticed in environment C. This environment contained some distinctive landmarks, such as the junctions and corners between corridors, and adding exteroceptive sensory information helped the robot to relocalise itself more quickly. Environment D contained no such perceptual cues, consisting of a single, straight corridor which is subject to an extreme level of perceptual aliasing. Neither system 2 or 3 was able to relocalise the robot here within the 20 m travelled by the robot. (A further experiment conducted over 50 m confirmed that both systems were eventually able to catch up with the control after approximately 40 m of travel.) This indicates the importance of the histogram matching mechanism in extracting as much of the available information as possible from the robot's sonar sensor readings for global localisation.

7.5 Summary of Results

The new self-localisation system was tested in a number of real world environments, using quantitative performance measures to assess its performance. Controlled experiments were also used to investigate the influence of individual system components. The new histogram matching mechanism for landmark recognition proved to be particularly effective, requiring only minimal computational resources. The lost robot experiment showed that the new self-localisation system relocalises the robot reliably under global uncertainty in middle-scale environments. Lesion experiments demonstrated the importance of the new landmark recognition mechanism and the position tracking equations; both mechanisms improved the quality of global localisation.

7.6 Concluding Remarks

This chapter described the development and quantitative analysis of a complete self-localisation system for a Nomad 200 robot. A cross-correlation technique for matching local occupancy grids using histograms was developed to overcome the high computational requirements of Yamauchi and Langley's approach (1997). This mechanism was combined with a novel self-localisation algorithm for accumulating sensory evidence over time, allowing the robot to relocalise under global uncertainty. This algorithm combines exteroceptive and proprioceptive sensory information without being dependent on *a priori* position knowledge from dead reckoning for global position estimation.

In this approach, the robot's location model consists of a mixture of Gaussian density functions, which are updated by an iterative self-localisation algorithm on the basis of new sensory evidence. The result is a unified solution to the problems of global localisation and position tracking. Furthermore, the system requires only minimal computational resources due to the efficiency of its matching algorithms. This approach is therefore well-suited to situations in which no radio link to external processors is available, and operation within real-time constraints is required. To the best of my knowledge, it is the only such system at the current time which can operate in environments of the scale presented using only the robot's on-board computing power.

In the experiments presented, the robot was provided with a pre-installed map and explored the environments using a reactive wall-following behaviour. At the end of this chapter, the question of autonomous map building by the robot still remained unanswered. In particular, two key problems needed to be solved; firstly, the maintenance of a geometrically consistent environment model, and secondly, the exploration of an unknown environment to obtain the sensory information required for map learning. These problems are therefore addressed in the following chapters.

Chapter 8

On-Line Map Learning

About this chapter. This chapter describes the algorithms used by the robot for map learning. A fundamental problem is that dead reckoning cannot be used for global position estimation during map building because of cumulative drift errors caused by wheel slippage. Instead, an optimisation algorithm was developed to maintain global consistency in the robot's map, using only local dead reckoning to obtain the metric relations between places.

8.1 Introduction

In order to achieve concurrent map building and self-localisation, the robot needs to solve a "chicken and egg" problem; self-localisation requires a map of the environment, while map building requires the ability to self-localise. The previous chapters considered the topic of self-localisation in isolation, assuming that a previously acquired environment model was provided to the robot. This chapter describes the algorithms developed to allow the robot to acquire and maintain a map of its environment.

As described previously, the map consists of a list of places, each identified by two histograms, and a list of links which connect some pairs of places. Each link is labelled with local metric information, describing the distance and angle between the two places it connects. In addition, the places in the robot's map are located within a Cartesian coordinate system. These coordinates are useful for a number of purposes, including:

- *Self-Localisation.* The relative locations of perceived environmental features in the coordinate system were used by the localisation algorithm described in the last chapter.

- *Exploration of Uncharted Territory.* The coordinates are used by the exploration system described in the following chapter to infer possible regions of unexplored territory in the robot's environment.

A fundamental problem for robot map building is that odometry can only produce accurate coordinate values by dead reckoning over short distances. Over longer distances, drift errors caused by wheel slippage accumulate, and the position estimates quickly become unreliable. I therefore developed an optimisation algorithm to assign Cartesian coordinates to the places in the robot's map, using only the *local* metric relations between places. In section 8.4.2, it is proved that this algorithm will always converge to a globally optimal solution, thereby maintaining geometric consistency in the map.

8.1.1 Related Work

Engelson & McDermott (1992) developed a passive mapping system, in which they attempted to detect and diagnose errors in a topological map when further, conflicting sensory information was obtained. For example, their system contained rules for merging together different nodes in the map which were found to correspond to the same physical location. However, this approach was only implemented in simulation, and much of the world knowledge it required was embedded in the simulator, so it would be unlikely to work on a real robot.

Lu & Milios (1997a) 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 Lu & Milios (1997b). 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 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.

A similar approach, using a graph-based model of the robot's environment, was developed by Golfarelli *et al.* (1998). Their 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. The x and y components of the elasticity parameters for both springs were derived, defining a 4×4 "stiffness matrix" for each link. A globally consistent set of coordinates was then determined by applying a procedure known as the "stiffness method" (Martin 1966, p. 21). In this procedure, the individual stiffness matrices are superimposed to form a $4m \times 4m$ matrix, where m is the number of nodes, then the equilibrium position for the whole structure is calculated. This calculation requires inversion of the $4m \times 4m$ matrix, so the approach would be likely to be at least as expensive as that of Lu and Milios.

Shatkay & Kaelbling (1997) and Shatkay (1998) 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 (consistency of distance measurements between places) and anti-symmetry (all links assumed to be bi-directional) were enforced directly in the re-estimation procedure for obtaining the probabilities in the HMM. This algorithm is heavily dependent on a good initial model to avoid local maxima. 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.

Both approaches described above, namely matrix methods and expectation maximisation, are computationally expensive. The matrix methods generate a globally consistent map in a single step, by solving a set of N simultaneous equations, where N is the number of places in the map. This operation requires the inversion of a large matrix every time that new information is added to the map. By contrast, the new optimisation algorithm presented in this chapter solves the problem by minimising an energy function over many small steps, as in a Hopfield network (Hopfield 1982). It is particularly efficient because it does not need to recompute the whole coordinate system from scratch every time that new information is added — instead, the existing solution is refined. In contrast to expectation maximisation algorithms, which are subject to local maxima, the new method is guaranteed to find a globally optimal solution.

8.2 Map Representation

The map built by *FortyTwo* consists of a set of N place nodes, and a set of links which connect some pairs of places. Each place i consists of a Cartesian coordinate (x_i, y_i) and a variance v_i reflecting the uncertainty in the calculation of that coordinate, according to the noise model described in section 4.3.2. Each link connects two places i and j, and is associated with a confidence level $0 \leq c_{ij} \leq 1$ reflecting the robot's belief that the link can be successfully traversed. A vector (d_{ij}, θ_{ij}) is also attached to each link, where d_{ij} refers to the distance and θ_{ij} to the heading observed by the robot in moving from place i to place j. Again, the uncertainty in the measurement of this vector is represented by a single variance u_{ij} according to the noise model in section 4.3.2. In this thesis, the links were constrained to be bi-directional, that is, $c_{ij} = c_{ji}$, $d_{ij} = d_{ji}$ and $\theta_{ij} = \theta_{ji} + \pi$.

8.3 Map Acquisition

Whenever the robot moved between two places i and j for the first time, a new topological connection was recorded in the map. The confidence level c_{ij} for each link was initialised to 0.5 and adapted according to the following rules taken from Yamauchi & Beer (1996). During subsequent traversals of an existing link in the map, the confidence level was increased using

$$c'_{ij} = \lambda + (1 - \lambda)c_{ij}, \tag{8.1}$$

where the link adaptation rate, λ, was 0.5 in these experiments. Conversely, whenever the robot failed to traverse a given link, for example, because the robot reached a different place to the intended destination produced by path planning, the confidence value was decreased using

$$c'_{ij} = (1 - \lambda)c_{ij}. \tag{8.2}$$

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

In addition, the distance d_{ij} and heading θ_{ij} of the robot between the two places was recorded, and the variance u_{ij} in this the measurement of this vector (d_{ij}, θ_{ij}) was estimated as 5% of the distance travelled. These measurements were obtained by simple vector arithmetic, taking into account both the displacement of the robot between the two places, measured by odometry, and the most likely offset of the robot from the centre of each place, determined by the histogram matching procedure described in section 7.2.3. These calculations are summarised as follows:

$$d_{ij} = \sqrt{(\triangle x_{ij})^2 + (\triangle y_{ij})^2}, \tag{8.3}$$

$$\theta_{ij} = atan\left(\frac{\triangle y_{ij}}{\triangle x_{ij}}\right), \tag{8.4}$$

where

$$\triangle x_{ij} = r_{x_i} + \triangle x - r_{x_j}, \tag{8.5}$$

$$\triangle y_{ij} = r_{y_i} + \triangle y - r_{y_j}, \tag{8.6}$$

where the vector $(\triangle x, \triangle y)$ refers to the robot's own displacement in Cartesian space observed using its on-line compass-based odometry (section 4.3.1), and the (r_x, r_y) refer to the mostly likely offsets for places i and j calculated by the histogram matching mechanism (section 7.2.2). The local metric relations (d_{ij}, θ_{ij}) are defined by Polar rather then Cartesian coordinates to emphasize the *relative* nature of the measurements.

8.4 Maintaining Global Consistency

The following relaxation algorithm was developed to assign globally consistent coordinates to the places in the robot's map using only the local metric relations between places. In this approach, the coordinates of the places are treated as free variables which are continuously refined over time towards an optimal solution. The basic idea is to pick each node in turn, and move it to "where its neighbours think it should be". By repeated application of this rule, the map gradually converges upon a globally consistent set of coordinates.

8.4.1 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). 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. A new coordinate (x_i, y_i) is then calculated by taking a weighted mean of the estimates (x'_{ji}, y'_{ji}) as

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

This last step has the effect of moving the node to the average of where the neighbouring nodes "think it should be", weighted by the variance in each of these estimates. 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 experiments presented in this thesis, the algorithm was run for a single iteration each time the robot moved to a different place in the map.

An example illustrating the propagation of uncertainty by the relaxation algorithm is shown in figure 8.1. This diagram shows that where there are several measurements for a place coordinate, i.e., because the node is connected to several neighbours, the variance in that coordinate tends to be low. Conversely, where a node is only connected to one neighbour, its variance tends to be high. In the complete system for concurrent map building and self-localisation, the variance measures assigned to the places by the relaxation algorithm are combined with the variance estimates produced by scan matching (equation 7.6) to obtain the measurement errors in the self-localisation algorithm (see section 7.2.3).

8.4.2 Proof of Convergence

Each link in the map can be thought of as a spring which connects two adjacent places i and j (Lu & Milios 1997a; Golfarelli *et. al.* 1998; Shatkay 1998). The spring reaches minimum energy when the relative displacement between the

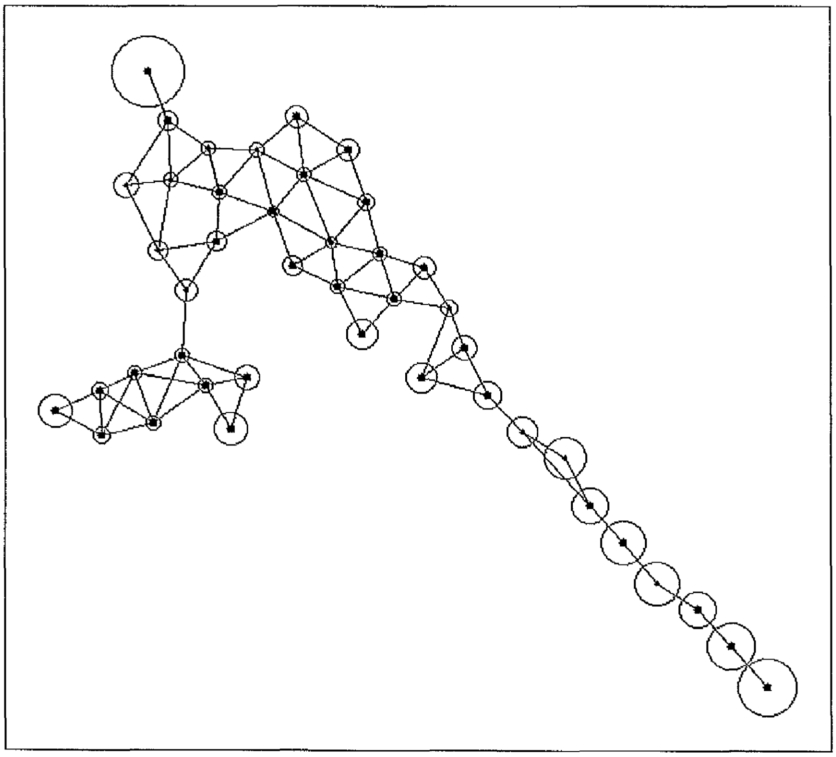

Figure 8.1: Example showing the propagation of uncertainty by the relaxation algorithm. In this figure, the nodes are spaced at approximately equal intervals of 1 m. The variance in each of the coordinates after the relaxation algorithm converges is indicated by the radius of the corresponding circle. The variances were exaggerated by a factor of 10 here for illustration purposes.

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:

$$E = \sum_i \sum_j{}' (x_j - x_i + d_{ji}\cos\theta_{ji})^2 + (y_j - y_i + d_{ji}\sin\theta_{ji})^2, \tag{8.7}$$

where $\sum_j'$ refers to the sum over the neighbours of a given node. In order to prove convergence of the relaxation algorithm, it is sufficient to show that the algorithm is always guaranteed to minimise this energy function. The following proof consists of examining the change in energy after updating a single node in the map and showing that this must always be less than zero.

In order to simplify the notation, it is assumed that all of the variances in the map are equal to 1, since the actual values of the variances will not affect the generality of the proof. The following notation is also adopted:

$$C_{ji} = x_j + d_{ji}\cos\theta_{ji}, \tag{8.8}$$

$$S_{ji} = y_j + d_{ji}\sin\theta_{ji}. \tag{8.9}$$

Consider how an arbitrary node with coordinates (x_i, y_i) is updated by its neighbours:

$$x_i' = \frac{1}{N_i}\sum_j{}' C_{ji}, \tag{8.10}$$

$$y_i' = \frac{1}{N_i}\sum_j{}' S_{ji}. \tag{8.11}$$

where $N_i = \sum_j' 1$ refers to the number of neighbours of node i.

The total change in energy after updating any node i is defined by

$$\Delta E = 2\left\{-\sum_j{}' \left((C_{ji} - x_i)^2 + (S_{ji} - y_i)^2\right) + \sum_j{}' \left((C_{ji} - x_i')^2 + (S_{ji} - y_i')^2\right)\right\}. \tag{8.12}$$

There is a coefficient of 2 because each link is bi-directional and therefore updates both of the nodes that it connects. It is sufficient to show that $\Delta E \leq 0$ to prove convergence, i.e., that the total energy before updating any node is always greater than or equal to that afterwards. Consider the change in energy for the x coordinate of the updated node,

$$\begin{aligned} \Delta E_x &= \sum_j{}' (C_{ji} - x_i')^2 - \sum_j{}' (C_{ji} - x_i)^2 && (8.13) \\ &= \sum_j{}' \left(C_{ji} - \frac{1}{N_i} \sum_j{}' C_{ji} \right)^2 - \sum_j{}' (C_{ji} - x_i)^2. && (8.14) \end{aligned}$$

The energy function is defined in terms of the relative positions of the points rather than absolute coordinates. It is therefore possible to add a constant vector to all points without affecting the energy. Thus, without loss of generality, let $x_i = 0$. Hence,

$$\begin{aligned} \Delta E_x &= \sum_j{}' \left(C_{ji} - \frac{1}{N_i} \sum_j{}' C_{ji} \right)^2 - \sum_j{}' C_{ji}^2 && (8.15) \\ &= \sum_j{}' C_{ji}^2 - \frac{2}{N_i} \sum_j{}' \left(C_{ji} \sum_j{}' C_{ji} \right) + \frac{N_i}{N_i^2} \left(\sum_j{}' C_{ji} \right)^2 - \sum_j{}' C_{ji}^2 && (8.16) \\ &= -\frac{2}{N_i} \left(\sum_j{}' C_{ji} \right)^2 + \frac{1}{N_i} \left(\sum_j{}' C_{ji} \right)^2 && (8.17) \\ &= -\frac{1}{N_i} \left(\sum_j{}' C_{ji} \right)^2 && (8.18) \\ &\leq 0. && (8.19) \end{aligned}$$

In other words, updates always result in a decrease in energy until an equilibrium is reached. The algorithm will 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.

8.4.3 Complexity Analysis

In contrast to previous methods, the algorithm is computationally inexpensive. During on-line operation, it was 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, in which each node is connected to every other node, 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 $O(n)$. This compares favourably with the $O(n^3)$ complexity of matrix inversion methods. In the map building system presented here, the robot attempts to space the nodes at equal intervals, so the maximum connectivity per node is 6. (If the nodes are spaced at equal intervals, then for any node, the minimum angle of incidence between any two of its neighbours will be 60^o, where the three nodes form an equilateral triangle. Therefore, the maximum connectivity per node is $360^o/60^o = 6$.)

8.5 Results

To demonstrate the convergence of the relaxation algorithm, an off-line experiment was conducted in which the coordinates of a previously acquired map were reinitialised to randomly selected values. The algorithm was then iterated until visual inspection revealed that a steady solution had been obtained (see figure 8.2). By fixing one of the points as the origin throughout, it was confirmed that the coordinates in the map always converged to exactly the same solution, as implied by the proof of convergence.

In order to assess the accuracy of the maps produced by the complete on-line map building system, an experiment was conducted in which a geometrically "correct" map was recorded by a human observer for comparison with the robot's self-acquired map. A laboratory environment of size $5.38m \times 3.88m$ was used to allow manual recording of the robot's position with a tape measure (see figure 8.3). The robot explored the environment from a variety of different starting positions, and the actual position of the robot was recorded whenever it added a new place to the map using the exploration strategy described in the previous chapter. Here, the exploration software was modified to make sure that the robot physically traversed every link in the map, so that the robot obtained a measurement of the local metric relation between every pair of connected places.

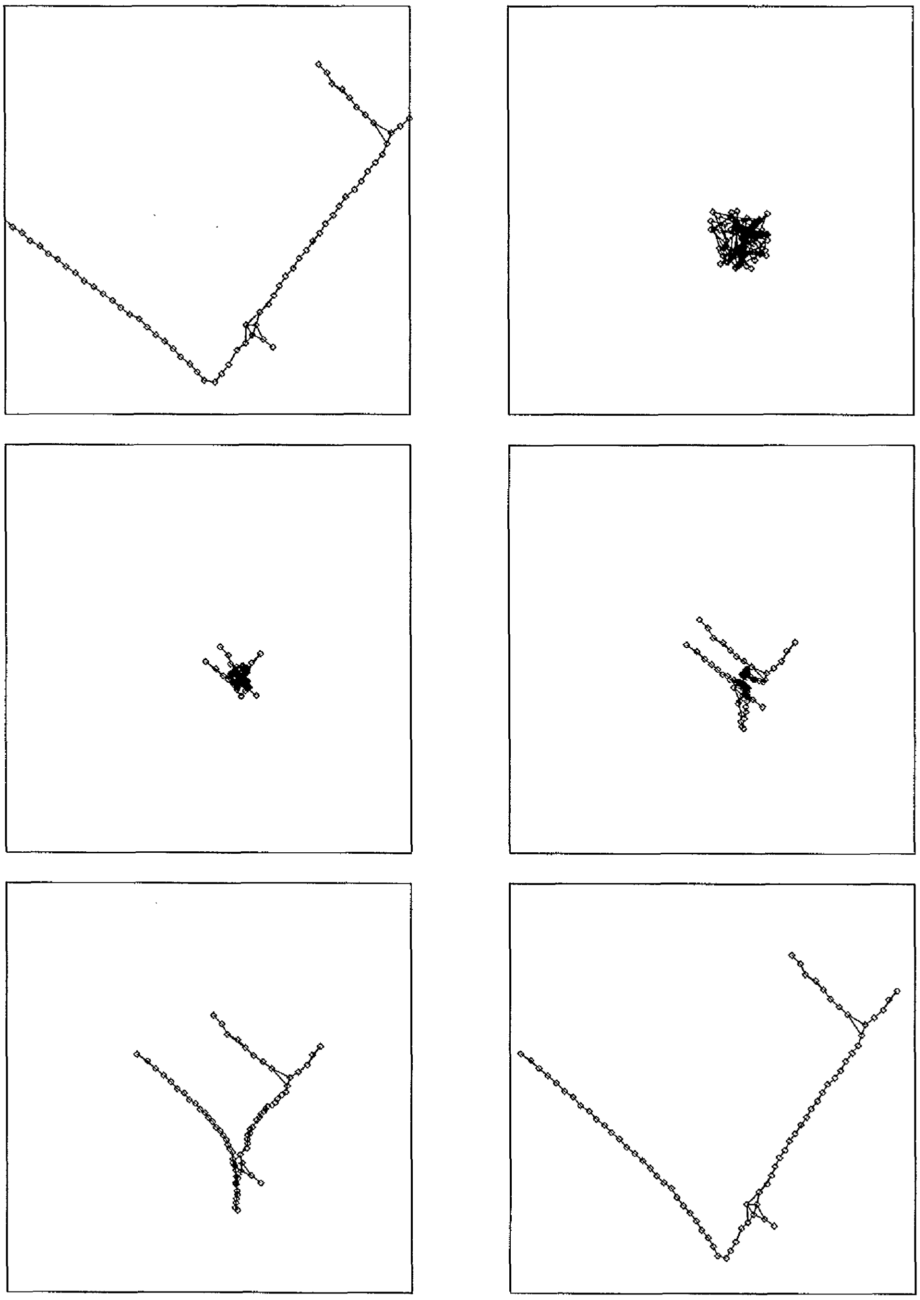

Figure 8.2: Convergence of the relaxation algorithm. The first picture shows a topological map constructed incrementally by the robot during exploration of a corridor environment (see figure 10.1 for the corresponding floor plan). In the second picture, the coordinates in the map have been randomly reinitialised. The remaining pictures show the map after 5, 25, 50, 250 and 500 iterations respectively of the relaxation algorithm in section 8.4.1.

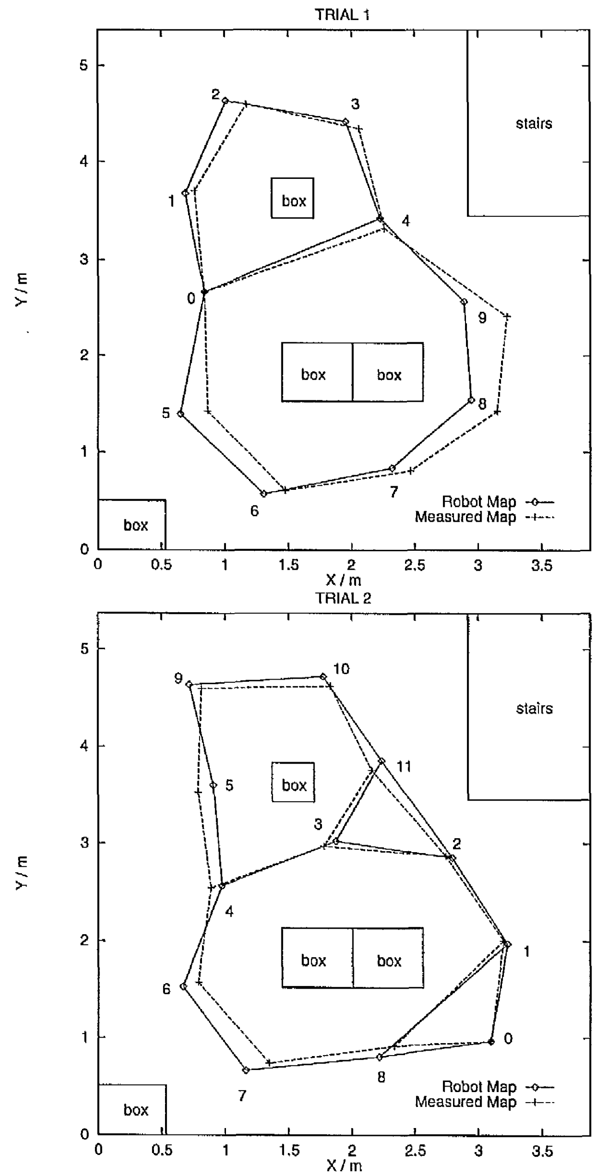

Figure 8.3: Accuracy of the maps produced by the complete system. The maps produced by the robot are compared to the equivalent maps created by a human observer. The numbers indicate the sequence in which the places were added to the robot's map. A laboratory environment of size $5.38m \times 3.88m$ was used to allow manual recording of the robot's position.

The results shown in figure 8.3 reflect the accuracy of the whole system, and therefore represent the worst case results for the relaxation algorithm, since the maps produced will also incorporate any errors in self-localisation, histogram matching and local dead reckoning by the robot. Nevertheless, the maps obtained show a good correspondence with the measurements taken by the human observer, indicating the efficacy of the overall approach. The higher the connectivity of the robot's map, the closer it tends to be to the equivalent human-measured map.

8.6 Discussion

Leonard & Durrant-Whyte (1992, p. 146) considered rock-climbing to be a useful analogy for concurrent map building and self-localisation. They regarded map building

> "... as a methodical, incremental process, in which each successive foothold (vehicle position) is made secure before the next advance. The uncertainty introduced into the system by each move of the vehicle must be eliminated as early and often as possible."

In other words, mapping errors are avoided by trying to prevent erroneous information from being added to the map in the first place. However, this approach is bound to fail sooner or later for the following reasons. Firstly, the models used by robots to interpret sensory information can only approximate the real world, so mapping errors are inevitable on a real robot. Secondly, even if perfect sensors were available, errors would still occur, because real world environments are subject to variations which cannot always be predicted by an autonomous agent. The ability to correct existing errors in the map based on new, conflicting sensory information is essential for navigation in middle-scale environments.

In the map learning scheme described in this chapter, the robot has the ability to modify its map on-line in order to correct possible sensing errors. Places and links can be added or deleted, allowing the robot to adapt its internal representation to the observed topography of the environment. The relaxation algorithm is then applied to find the best map to fit all of the available information, using all of the local metric relations in the map to obtain the place coordinates. In contrast to many previous map building systems, for example, Leonard & Durrant-Whyte (1992), Kurz (1996) and Yamauchi & Beer (1996), the algorithm can thus correct existing errors in the map as well as mapping new territory, because all of the place coordinates are recalculated at each iteration.

Solutions to the consistency maintenance problem exist which are based on matrix methods (Lu & Milios 1997a; Golfarelli *et al.* 1998). However, in comparison to the relaxation algorithm presented here, these methods are complicated to implement and require a large amount of computation to recalculate the whole map whenever new sensory information is obtained. The relaxation algorithm is particularly efficient because it does not throw 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 complete map building system developed, it was found that it was only necessary to run the relaxation algorithm for a single iteration at each cycle of the exploration process.

A further advantage of the approach is that it is possible to adapt the robot's map slowly in the direction of the global minimum in the energy function, rather than jumping straight to the "correct" solution. This is useful for concurrent map building and self-localisation, because simultaneous updates to the robot's environment and location models can lead to the mutually destabilising effect reported by Zimmer (1995b), in which each representation becomes corrupted by the errors in the other. The technique of adapting the map by a single iteration each time counters this effect by providing the robot with a degree of "inertia" in its environment model.

8.7 Concluding Remarks

In this chapter, a relaxation algorithm for maintaining global consistency in the robot's map was presented. This algorithm is self-organising, using only local information and local interactions to converge upon a globally optimal solution. It works by minimising an energy function over many small steps, as in a Hopfield network (Hopfield 1982). Apart from being simple and easy to implement, the algorithm is very effective because it uses all of the local metric information in the map to obtain the place coordinates. The algorithm is also very efficient (its complexity is $O(n)$ compared to the $O(n^3)$ complexity of matrix inversion methods), enabling fast, on-line map learning by the robot in middle-scale environments. Furthermore, in contrast to expectation maximisation algorithms (Shatkay &

Kaelbling 1997; Thrun *et al.* 1998b), which are subject to local maxima, it has been proved that the algorithm will always converge to a global solution. The next chapter considers how to explore an unknown environment in order to obtain the sensory information required for map learning.

Chapter 9

Exploration of an Unknown Environment

About this chapter. In order to build a map of an unknown environment, a robot needs to travel to uncharted territory. A map-based exploration system is presented, in which the environment model is acquired incrementally by the robot. The basic mechanisms include an artificial neural network which is trained to detect areas of open space in the environment.

9.1 Introduction

Various possible strategies for exploration of an unknown environment are described in the robotics literature. The following taxonomy is taken from Lee (1995, p. 19):

1. *Human control.*

2. *Reactive control.*

3. *Approaching the unknown.*

4. *Optimal search strategies.*

In the first approach, the robot is guided around the environment by a human operator. A map is constructed either on-line using a passive mapping system (Kortenkamp 1993; Engelson 1994), or using an off-line learning algorithm after exploration has been completed (Shatkay 1998). This approach requires human intervention in the map building process, and is therefore unsuitable for use by a self-navigating mobile robot.

A reactive exploration strategy is another alternative (item 2), for example, wall-following (Mataric 1991; Nehmzow & Smithers 1992) or random exploration. However, while reactive behaviours are often very robust, they cannot be guaranteed to build complete maps in middle-scale environments. Wall following cannot be used to explore areas of open space such as rooms, and random exploration is unsuitable for environments of any real complexity; the robot may get "trapped" in one part of the environment or take many hours, if at all, to cover the whole area.

In the strategy known as *approaching the unknown* (item 3), the robot tries to move towards the regions of the environment about which it knows the least. The map acquired by the robot is used to guide the exploration process, directing the robot towards areas of uncharted territory. The new sensory information obtained by the robot as it moves into new territory is used in turn to update the map. This process is repeated until the whole environment has been covered.

A version of the last strategy, in which the robot continuously tries to expand the territory which has already been charted, was used here. The basic idea is that the robot travels to the edge of the existing map, and then uses its range-finder sensors to detect new regions of uncharted territory. The new territory is added to the map, then the robot tries to reach the next unexplored edge of the map. The sequence is repeated until the robot has traversed the entire environment.

The approach differs from previous work in that it does not require high precision sensing or depend upon simplifying assumptions about particular environments, and has been tested in populated, real world environments in experiments reported in this chapter. An artificial neural network is used to detect areas of unexplored territory, fusing together information from the robot's sonar and infrared sensors. All of the data required for training the network is collected by the robot itself, avoiding the need for the system designer to determine the training signal. The complete system requires only minimal computational resources, thereby eliminating the need for off-line processing and increasing the autonomy of the robot.

In addition to the exploration strategies described above, Lee identified a fourth category, namely *optimal search strategies*. In the new exploration system, the robot uses a greedy, sub-optimal strategy, namely "head for the nearest area of unexplored territory". However, extending the system to use an optimal strategy would be straight-forward, since deciding which place to visit next is equivalent to the well-known Travelling Salesman Problem (Gibbons 1985).

9.1.1 Related Work

Yamauchi (1997) developed a technique called "frontier-based exploration". In his system, a global occupancy grid was used to represent the environment. Image segmentation techniques were used to extract regions in the grid between charted and unknown territory known as "frontiers". Exploration was then directed towards the frontiers. A disadvantage of this approach is that it depends critically upon accurate laser sensors and precisely corrected odometry, because exact position information is needed to update grid-based maps.

Thrun (1998b) also developed a map building system based on a global grid model. An artificial neural network was trained to translate neighbouring groups of sonar readings onto occupancy values in the grid. Exploration was then directed towards areas of high uncertainty in the acquired map. The required training examples were obtained using a simulator, though the trained neural networks were shown to work well on the real robot. However, this system depends on an assumption that environments are rectilinear, i.e., that surfaces are always parallel or perpendicular to each other.

Edlinger & Weiss (1995) developed a robot map building system in which the map consisted of a set of laser range-finder scans and the topological relations between the scans. Their system attempted to detect obstacle-free segments in the scans known as "passages", that is, regions of open space which are wide enough for the robot to move into. The detected passages were added to a stack of unexplored locations, which were visited in turn until the whole environment had been covered by the robot. This system was tested successfully in a static office environment of middle-scale dimensions, though the sensing strategy used could fail in populated environments, because possible passages in the laser scans might be occluded by moving people.

Kunz *et al.* (1999) developed an automatic mapping system called *Inducto-Beast* which used a topological map and dead reckoning to identify places. To correct the rotational drift errors affecting the robot's odometry, environments were assumed to be rectilinear. An interesting feature of this system is its use of inductive learning during map building, hypothesizing the existence of unexplored hallways based on knowledge of the symmetries which commonly occur in office buildings. However, this approach depends on a set of simplifying assumptions about office environments, such as rectilinearity, known corridor widths and minimum distances between junctions. This means that the system would fail in environments which do not conform to these *a priori* assumptions.

Of the above studies, the approaches based on topological maps would seem the most promising, due to the higher accuracy required for updating grid-based maps. The assumption of rectilinearity, however, while an attractive simplification for many indoor environments, cannot be used in environments which contain non-rectangular objects such as furniture and plants. For example, the recreation area used for one of the experiments presented here contained cylindrical rubbish bins, concave vending machines and a number of objects, including a wall, in orientations which would cause this assumption to fail. The approach presented here is closest in spirit to that of Edlinger & Weiss (1995), though it does not depend upon accurate range-finder sensing and takes various steps to eliminate possible occlusions and variations in the robot's sensor data (see section 9.3).

9.2 Exploration Strategy

In the new exploration system, the robot builds a topological map which is augmented with metric information concerning the distance and angles between connected places. The map contains two different types of places (see figure 9.1):

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

Exploration consists of continuously trying to expand the territory already mapped by the robot, using a neural network to add new predicted places to the map. Subsequent movement by the robot is used to verify whether the predicted places actually exist or not. This exploration strategy is carried out as follows.

From its initial location, the robot takes a new sensor scan. It examines this scan, searching *in all directions* for possible areas of uncharted territory, using the neural network to identify areas of open space. The robot adds the first set of predicted places to the map, and then attempts to navigate to the nearest

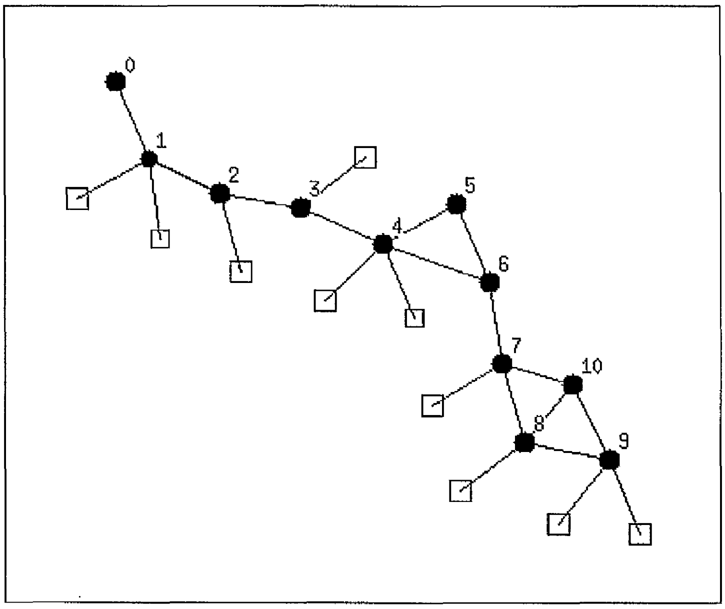

Figure 9.1: Example of topological map building. Places predicted by the neural network but not yet visited by the robot are shown by squares. Places actually visited by the robot are shown by filled circles. The numbers indicate the sequence in which the nodes were visited.

predicted place (choosing one at random if there are several options). If the robot is able to move to a physically distinct new location in the environment without encountering any obstacles, the predicted place is replaced by a confirmed place, otherwise it is deleted. If any obstacles *are* encountered, reactive sensor-motor behaviours steer the robot away from possible collisions. A local dead reckoning strategy is used to decide whether to confirm the predicted places. Whenever another place is confirmed in the map, the neural network is used again to predict more new places. In addition, connections are inferred to any other confirmed places lying within a certain distance (1 m in these experiments) of the added node within the global coordinate system, 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.

9.2.1 Basic Mechanisms

In order to implement this strategy, the following mechanisms were required:

1. *Compass Sense.* To determine the direction of unexplored territory and to control the robot's heading during exploration, the robot needs to be able to determine its orientation (section 4.2).

2. *Open Space Detection.* Some mechanism was also required to add the new predicted places to the map, i.e., to detect areas of unexplored territory in a particular direction. Individual range-finder readings are not well suited for this purpose because of problems such as occlusions, sensor noise, cross-talk and specular reflections. Instead, an artificial neural network was trained to learn the concept of "open space", combining noisy information obtained from many sensor readings (section 9.3).

3. *Local Dead Reckoning.* To determine whether a predicted place should be confirmed and added to the map, a local dead reckoning strategy was used, based on the robot's on-line compass-based odometry (section 4.3.1). If the robot managed to travel by a pre-specified distance threshold (1 m in these experiments) from the nearest stored place in the map, as measured in the global coordinate system, then a new confirmed place was added to the map.

4. *Map Learning and Self-Localisation.* The map learning mechanisms employed by the robot were described in the previous chapter. The robot also needs the ability to determine its location within that map, as described in chapter 7.

5. *Way Finding.* Path planning was carried out using Dijkstra's algorithm. The robot's heading was controlled by taking into account the robot's current location in the map, the compass sense and the shortest path to the goal location. A reactive controller was used for moving forward while avoiding obstacles (section 9.4).

9.3 Learning a Model of Open Space

To detect regions of unexplored territory, I developed a mechanism which uses an artificial neural network to estimate the likelihood of the robot being able to move into open space — that is, space which is unoccupied by any object — in a given direction. A fully connected, feedforward network with 6 inputs, 3

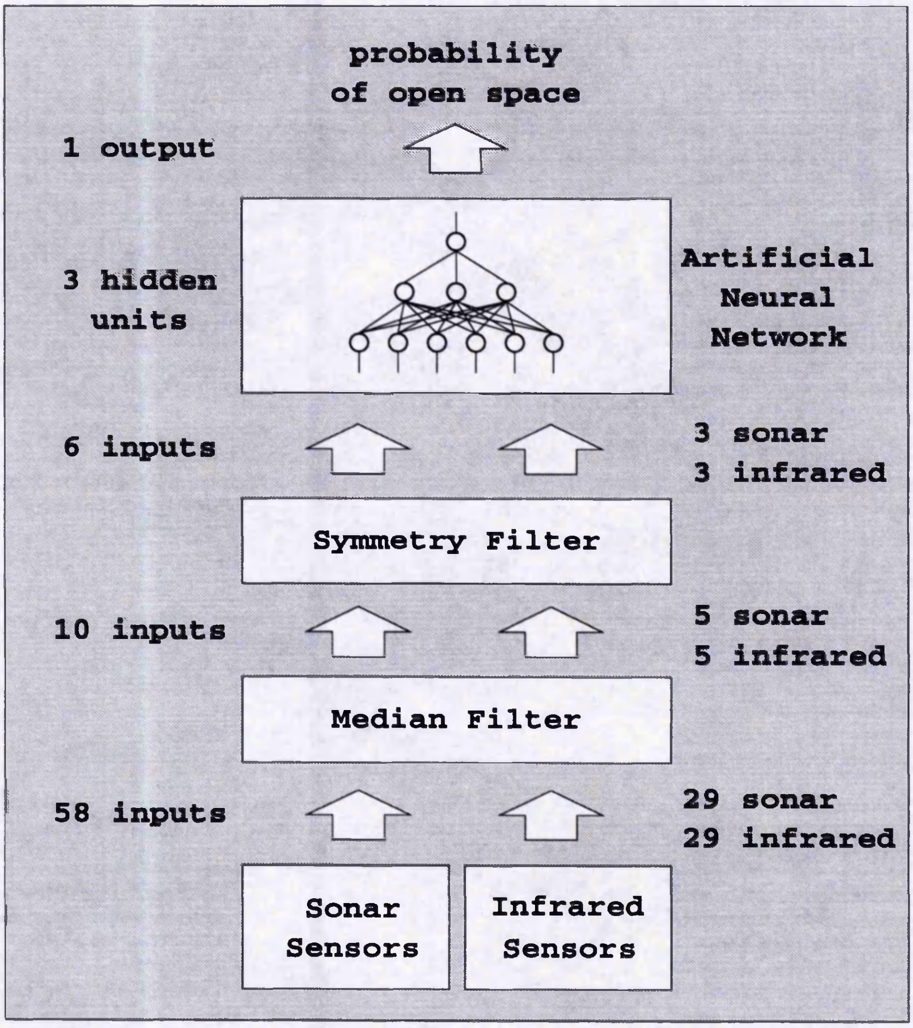

Figure 9.2: Architecture for open space detection. Two pre-processing functions were applied to the sensory input, then an artificial neural network was used to detect the presence or absence of open space in a given direction (see section 9.3).

hidden units and 1 output was trained to associate the sensory input in the given direction with the robot's ability to travel by a pre-specified distance (1 m) in that direction. The output of the network was the probability of open space (see figure 9.2). All of the data required for training the network, including the ability to travel in a particular direction, was collected by the robot itself, thus avoiding the need for manually labelling the training examples with the desired output categories, as in the concept learning mechanism described by Mahadevan *et al.* (1998).

9.3.1 Data Collection

The sensing strategy used by the robot consisted of rotating its turret to obtain a detailed scan, consisting of 144 sonar and 144 infrared readings at 2.5 degree intervals (see section 4.5). For data collection, a scan was first taken, then the robot attempted to move as far as possible in an arbitrary direction until an obstacle was detected within 15 cm of the robot, recording both the sensor readings from the scan and the distance travelled. Data was collected in several different areas of the computer building at Manchester University, including the robotics laboratory, a junction and a corridor.

9.3.2 Pre-Processing

The sensor readings were first processed to take into account the heading of the robot. A subset of 58 of the 288 readings, centred around the direction in which the prediction of open or occupied space was to be made, was used as input to the classification mechanism. (For training and testing, this was the recorded direction of travel. During on-line operation, the sensor scan was subjected to a series of 144 rotations, the subset of 58 sensor readings being extracted from each rotated scan in turn before presentation to the classifier, thus enabling the prediction of open space in all 144 directions). The following functions were applied to the selected subset of sensor readings.

1. **Median Filter.** The robot's raw sensor readings rarely give accurate range measurements; the values may be too high, e.g., due to specular reflections, or too low, e.g., due to cross-talk or occlusions caused by moving people. To reduce these effects, groups of 5 or 6 adjacent sensor readings (of the same sensor modality, sonar or infrared) were combined to produce a single

reading, by taking their median value. This resulted in five sonar and five infrared inputs to the next pre-processing stage. An alternative would be to use the mean value, but this would be susceptible to the effects of outliers in the data; the median is recognised as a more robust statistic (Bishop 1995).

2. **Symmetry Filter.** This function was used to exploit the bilateral symmetry inherent in this classification task. For example, the left-most sonar reading was combined with the right-most sonar reading by taking the minimum of the two values (i.e., the nearest of the two obstacles detected). The middle-left and middle-right readings were similarly combined. This resulted in three sonar and three infrared inputs to the neural network, since the centre readings were not affected by this operation. An alternative would be to synthesize new training examples by manually reflecting the sensor readings, as in the training of the network used to steer the ALVINN autonomous vehicle (Pomerleau 1989), but this would double the size of the training set and necessitate a more complicated network architecture.

9.3.3 Training and Testing

A key issue was that of misclassification errors. Though the performance of the network used here was very good (see section 9.5), any classification mechanism is bound to make some errors. These errors will either be *false positives*, where the robot predicts open space when the space is actually occupied, or *false negatives*, where the robot predicts occupied space when the space is actually open. In the exploration system presented here, false positives are not a major problem, because subsequent movement by the robot is used to verify whether the predicted places actually exist. However, false negatives would pose a problem because they might cause the robot to miss some area of unexplored territory.

The solution adopted here was to bias the classifier mechanism into over-estimating the likelihood of open space in a given direction, thereby producing more false positives but fewer false negatives (none in the experiments presented here). The network was trained off-line by back-propagation, using the sensor-motor data previously collected by the robot, each training example consisting of a pre-processed sensor scan and a target output taken from the distance travelled by the robot (equal to 1 if the robot could travel by the pre-specified distance of

1 m, and 0 otherwise). The network was trained to produce outputs which can be interpreted as the posterior probability of open space by using the *cross-entropy* error function, given by Bishop (1995, p. 231) as

$$E = -\sum_{n=1}^{N} \{t^n \ln y^n + (1 - t^n) \ln(1 - y^n)\}, \tag{9.1}$$

where N is the number of training examples, t^n is the target output and y^n is the actual output of the network for a given training example $n = \{1, ..., N\}$. This can be compared to the *sum-of-squares* error function used most often for training neural networks by back-propagation, given as

$$E = \frac{1}{2} \sum_{n=1}^{N} (t^n - y^n)^2 . \tag{9.2}$$

During testing and on-line operation, a bias value (0.125 in these experiments) was added to the actual output of the network in order to produce the desired over-estimates. An input pattern was thus classified as "open space" if the actual output y^n was greater than 0.5, and "occupied space" otherwise.

9.4 Way Finding

9.4.1 High-Level Control

Dijkstra's shortest path algorithm (Gibbons 1985) was used for planning routes in the topological map. This is an exhaustive search algorithm which is guaranteed to find the shortest path to the goal location from all of the other nodes in a graph in the lowest possible computational complexity. An alternative would be to use a heuristic search method, the optimal choice being the A^* algorithm (Nilsson 1980), though the result would be exactly the same.

During exploration, the robot stopped to take a new sensor scan (section 4.5) every time it had travelled 0.5 m from the position in which the previous scan was taken. The new scan was used for both self-localisation, as described in chapter 7, and open space detection, as described above.

Whenever the self-localisation algorithm detected a change in the robot's current location in the map, a new heading was determined based on the direction to the next node on the shortest path to the goal location. The robot's steering orientation was then adjusted relative to the orientation of the turret, since the turret was already fixed in the direction of North by the compass sense.

9.4.2 Low-Level Control

After setting the new heading, control was passed to a set of reactive behaviours until a further 0.5 m had been travelled by the robot. During this phase, the orientation of the robot's turret was controlled using the compass sense. The initial behaviour used by the robot was to move forwards. If an obstacle was detected within 0.6 m of the robot by the forward facing sensors, control of the translational and rotational motors was passed to the previously acquired behaviour for wall-following described in section 4.4. If the detected obstacle was on the left side of the robot, a left-hand wall-following behaviour was used, otherwise right-hand wall-following was used.

Again, this raises the problem of sensor noise and occlusions. If the decision on whether to go left or right is based on a spurious sensor reading, the wrong decision might be made. The solution adopted here was to use a pair of "virtual" sensors; the "left" sensor being taken as the median value of the four infrared sensor readings to the immediate left of the direction of travel, and the "right" sensor as the corresponding median value on the other side.

The reason for using wall-following was as follows. In initial experiments, the learned behaviour for obstacle avoidance was used, which is based on the two instinct rules for moving forwards and turning away from obstacles (described in section 4.4). Sometimes, however, this behaviour would result in the robot failing to reach predicted places. For example, if the robot tried to turn into a narrow opening such as an intersecting corridor to reach a new place, the obstacle avoidance behaviour could sometimes cause the robot to miss the opening completely. Instead, it was found that the wall-following behaviour produced the best results, because the third instinct rule for turning towards obstacles meant that the robot would still avoid collisions but stay close enough to the original intended trajectory to find its way through relatively narrow spaces (e.g., the corridors in our building are 1.75 m wide).

9.5 Results

The neural network was trained to perform the open space detection task using a training set of 276 examples and a testing set of 92 examples, resulting in a

	Description	**Approx. Size**	**Map-Based** Distance/m	**Wall-Following** Distance/m
A	Long straight corridor	53 m × 3 m	106.64 ± 1.48	110.91 ± 0.09
B	L-shaped corridor	34 m × 33 m	146.33 ± 3.45	146.68 ± 0.31
C	T-shaped hallway	16 m × 13 m	58.02 ± 2.64	×
D	Recreation area	20 m × 11 m	91.86 ± 2.86	×

Table 9.1: Test environments for exploration. The total distance travelled with standard deviation is shown for the map-based exploration strategy in all four environments. Only the corridor environments A and B could be explored by wall-following, therefore no comparison between the two strategies is shown for environments C and D.

test error of 7.6%. This mechanism was then validated through its integration into the map-based exploration system, using a different set of environments to those used for training and testing. Using the proportion of predicted places which were not confirmed during map building, a validation error of 4.0% was observed. That this is lower than the test error can be explained by the fact that the data used for training and testing contained a higher proportion of "difficult" examples, such as junctions and corners.

The map-based exploration strategy was tested in a number of middle-scale environments in the computer building at Manchester (see table 9.1), which contained transient changes such as people walking past the robot and doors being opened and closed. An example of one of the exploration trials is shown in figure 9.4, illustrating the incremental acquisition of the robot's map. Two of these environments, A and B, were composed entirely of corridors, so could therefore also be explored systematically using wall-following. The performance of the map-based exploration system was compared to that of wall-following in these environments by taking the total distance required to conduct a complete tour of the environment and then return to the starting location. The results show that the performance of the map-based exploration strategy is comparable to that of wall-following in these environments, while having the additional benefit of being able to operate in open environments such as C and D.

To assess the quality of the maps obtained, the ability of the robot to navigate using a self-acquired map was considered. Firstly, the robot's ability to relocalise under global uncertainty, i.e., to recover its position after becoming lost, was assessed. Secondly, the map building system was validated through its integration

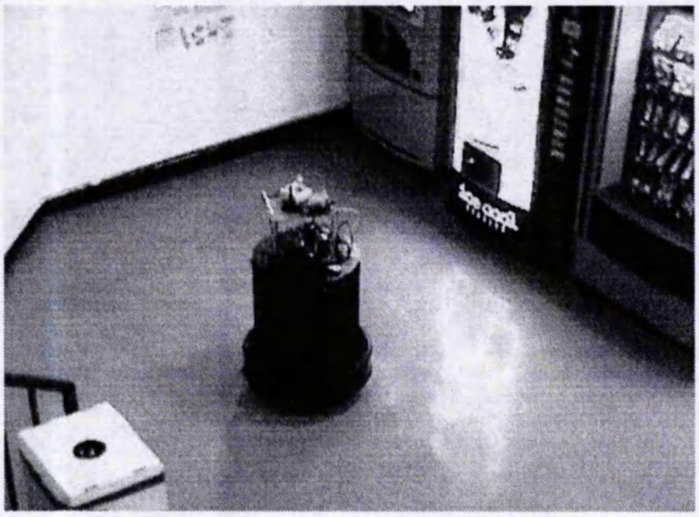

Figure 9.3: *FortyTwo* explores part of the recreation area near to some vending machines (environment D in table 9.1).

into a complete navigation system, which was tested on a delivery task in a middle-scale environment. These results are reported in chapter 10.

9.6 Discussion

In this chapter, a map-based exploration system for a mobile robot was presented. The basic mechanisms used included a compass, a topological map augmented with metric information and a neural network trained to detect areas of open space. The neural network can easily be retrained to work with other environments or sensors, thus increasing the overall flexibility of the system. In the experiments presented, a number of unmodified, real world environments of several hundred metres squared were mapped independently by the robot, without requiring off-line processing or human intervention in the map building process. Furthermore, no simplifying assumptions concerning the structure of the environment were required, as in previous approaches (Thrun 1998b; Kunz *et al.* 1999).

In the new exploration system, path planning takes place over the whole map, including both the predicted and confirmed places; this means that the robot is able to infer routes across territory which it has not yet visited. Subsequent exploration by the robot is used to confirm whether or not the inferred places

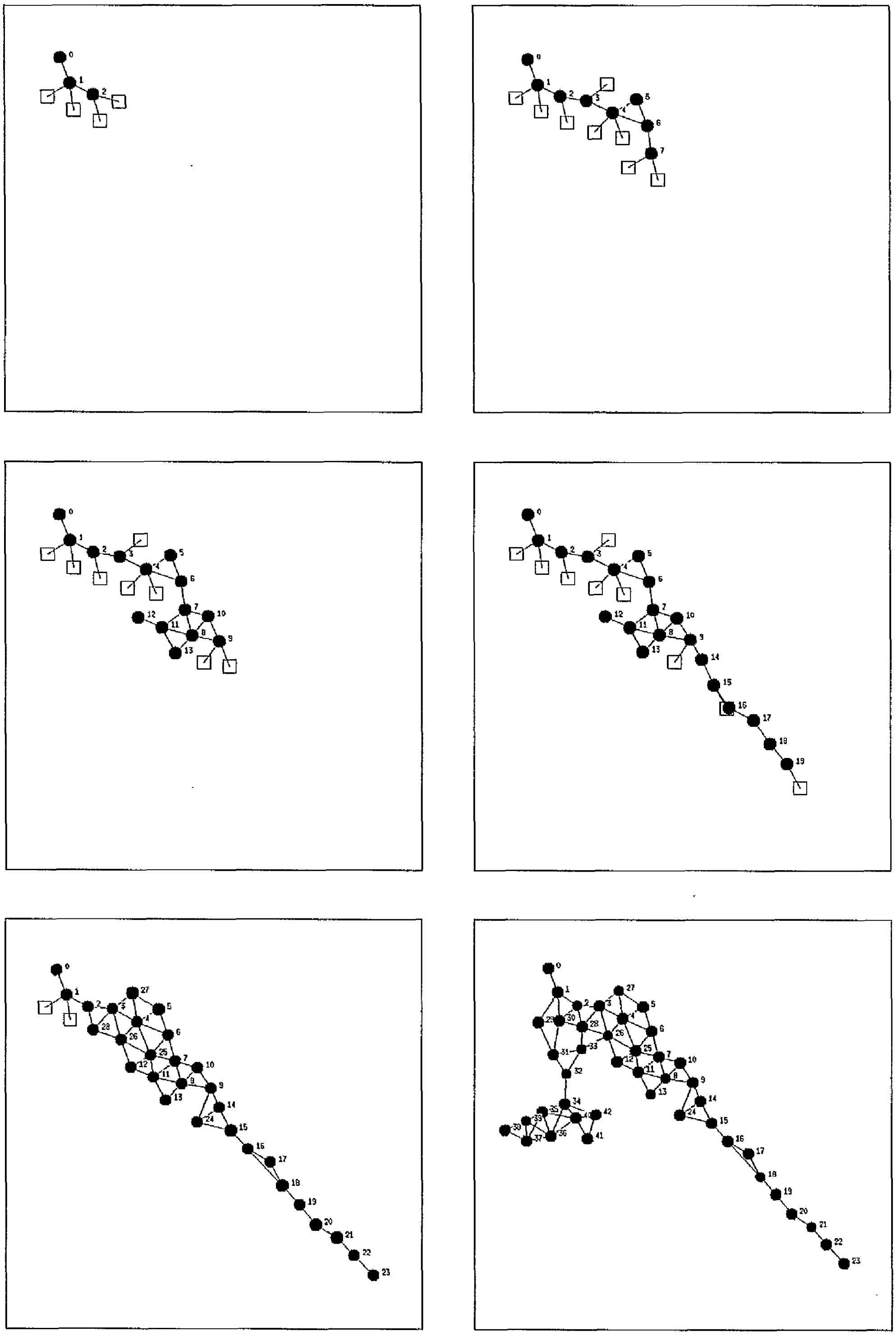

Figure 9.4: Testing of the exploration system. The map acquired by the robot in the recreation area (environment D in table 9.1) is shown after 2, 7, 17, 25, 55 and 80 steps of exploration, where the nodes in the map are spaced at approximately equal intervals of 1 m. The last region to be added to the map (see bottom-right picture) was the vending machine area in figure 9.3.

actually exist. If the intended route is found to be blocked, then the robot simply replans a new route to the goal location using the updated version of the map.

The way finding mechanism described in section 9.4 employs a somewhat *ad hoc* collection of routines to coordinate the high-level activities of path planning and path execution with the low-level sensor-motor behaviours. This is sometimes referred to as the *task integration* problem (Saffiotti 1997), that is, the problem of maintaining the right correspondence between the achievement of goals at the higher level and the execution of behaviours at the lower level. According to Saffiotti (1997, p. 17), "the problem of how to coordinate the activity of a set of behaviours remains the Achilles' heel of behaviour-based robotics". A proper treatment of this problem would warrant another PhD thesis in its own right. The routines used here were determined largely by trial and error, since the focus of this research was primarily on the high-level algorithms required for concurrent map building and self-localisation.

This chapter only considered exploration as a map learning problem, assuming that reliable location recognition is always possible; in some environments, the robot might occasionally need to travel back through previously charted territory in order to improve localisation quality, as in Zimmer (1995a) and Beetz *et al.* (1998). This issue is discussed further in section 10.3.

9.7 Concluding Remarks

This chapter described an automated mapping system based on a single tour of an unknown environment. An artificial neural network was trained to recognise areas of unexplored territory, by learning a model of open space, thus avoiding the pre-installation of a world model for this purpose by the system designer. This approach could be extended to allow continuous exploration of environments which are subject to structural changes over time. The relaxation algorithm presented in the previous chapter would allow constant adaptation of the robot's environment model depending on perceived changes to the environment. At the moment, the place signatures and the local metric relations in the map are fixed by "one-shot" learning; these would also need to be continuously modifiable.

To recap, chapters 6 and 7 considered the topic of self-localisation, assuming a previously acquired map of the robot's environment. This chapter and the previous chapter considered the topic of map building, assuming that the robot had the ability to localise itself. In combination, these chapters present a complete solution to the problem of concurrent map building and self-localisation by an autonomous mobile robot operating in middle-scale environments. The next chapter documents the integration testing and final validation experiments conducted to assess the performance of the complete system.

Chapter 10

The Complete System

About this chapter. This chapter describes the experiments conducted to test the integration of the mechanisms for self-localisation and map building into a single robot controller. A validation experiment for the complete navigating robot involving an office delivery task is then presented.

10.1 Integration Testing

To complete the research, all of the mechanisms described in the preceding chapters were integrated into a single controller for the Nomad 200 robot *FortyTwo*. An overview of the system architecture was provided in section 1.4.4. To facilitate human interaction with the controller, a basic interface was provided, allowing the user to initiate map building in a new environment or to specify a goal destination in the self-acquired map when the map building phase had been completed.

The map building abilities of the complete system were then tested in a number of different environments around the computer building at Manchester University. Results concerning the robot's ability to explore an unknown environment were presented in chapter 9. To assess the quality of the maps obtained, the robot's ability to navigate using its own self-acquired map was then evaluated. In particular, the robot's ability to recover its position after becoming lost was considered.

In chapter 7, relocalisation performance was assessed using a map which was created manually using retrospectively corrected odometry data. To validate the work of the last two chapters on autonomous map building, relocalisation

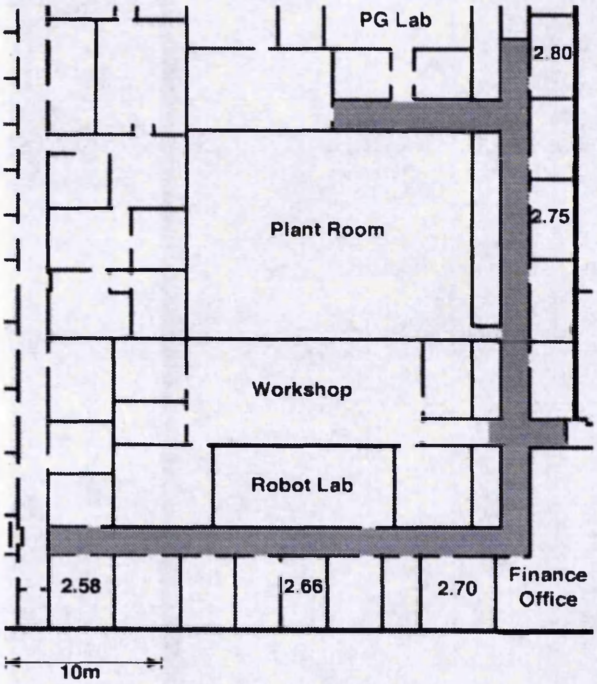

Figure 10.1: Floor plan of the corridor environment. The experiments were carried out in the shaded area shown in the upper storey of the computer building at Manchester (approximate size 34 m × 33 m).

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

Table 10.1: Routes traversed in the delivery experiment (see figure 10.1).

performance was assessed using a self-acquired map.

The experiment was conducted in the corridor environment shown in figure 10.1 (referred to as environment C in table 6.1). This area is a busy thoroughfare affected by many transient changes such as moving people, doors opening and closing, etc. The environment was also chosen because it is subject to very high levels of perceptual aliasing; in chapter 7 it was shown that a wall-following robot needed to travel as far as 18 m to relocalise successfully here after becoming lost.

To begin the experiment, the robot built its own map of the environment. The sensor-motor data required for measuring localisation quality was then collected by wall-following, and the experimental procedure described in section 7.4.2 was used to calculate the uncertainty coefficient $U(L \mid R)$ against the distance travelled by the robot. The performance was then compared to that of self-localisation using a manually constructed map (exactly as described in chapter 7). The first lap of the recorded data was reserved for constructing the pre-installed map and the remaining 2 laps were used for testing both systems. A total of 290 localisation trials were used in the calculation of the performance measures.

10.1.1 Results

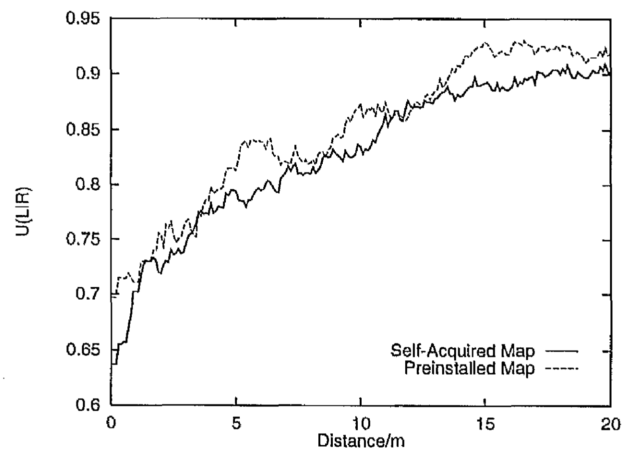

Figure 10.2: Localisation quality using a self-acquired map, compared to the performance using a pre-installed map.

The results in figure 10.2 show the localisation performance obtained using a self-acquired map compared to that using the manually installed map. The localisation quality achieved by the complete system is nearly as good as that achieved with a "perfect" pre-installed map. Note that this is only one experiment conducted using one set of data. The experiment would need to be repeated many times to show any kind of significance in this result. Such a study would require a great deal of time and effort; instead, it was decided to use the limited amount of time available to assess the complete system's performance by a more ecological method, measuring its ability to carry out an office delivery task.

10.2 Delivery Experiment

In order to validate the complete navigation system, a delivery experiment was conducted, in which the robot had to navigate between a series of user-specified locations. This was carried out firstly to test overall performance of the system under real world operating conditions. In addition, to assess the robustness of the system, its ability to recover from error in navigating to a goal location was considered. Here, the robot was "blindfolded" and physically moved to an unknown starting position at the start of each navigation attempt.

The robot first built its own map of the environment. The robot then had to find ten successive routes through the corridors, navigating from one user-chosen location to the next (see table 10.1). These locations form a path connecting a list of offices, designed to test the system on a likely delivery "mission". The experiment was carried out firstly with the robot knowing its initial location (the "control"), and then with the "lost" robot, where the robot's localisation mechanism was disabled and re-initialised at the start of each navigation attempt. This procedure was repeated 10 times to allow for experimental variations between trials, giving 200 trials in total. To assess the robot's performance, two measures were considered:

- *Effectiveness of navigation.* This was measured using the percentage of trials in which the robot navigated successfully to the goal location.

- *Efficiency of navigation.* This was measured using the length of the paths traversed by the robot in the successful trials. To compare the performance of the two systems, the ratio of their respective mean path lengths was taken.

Route	Control		Lost Robot		t-Test	Effic.
	Success	Distance/m	Success	Distance/m	p_{H_0}	Ratio
1	10	15.13 ± 0.88	10	15.84 ± 1.15	0.14	1.05
2	10	18.85 ± 1.10	8	$> 18.55 \pm 0.53$	0.45	0.98
3	10	12.32 ± 1.06	7	$> 14.08 \pm 1.74$	0.02	1.14
4	10	15.33 ± 1.10	9	$> 18.09 \pm 2.01$	0.02	1.18
5	10	13.26 ± 1.04	10	13.37 ± 0.92	0.82	1.01
6	10	16.72 ± 1.25	8	$> 17.52 \pm 2.01$	0.31	1.05
7	10	8.80 ± 1.21	10	9.30 ± 1.65	0.44	1.06
8	10	11.87 ± 0.64	10	11.20 ± 1.12	0.29	0.94
9	10	28.80 ± 1.48	10	28.77 ± 1.90	0.97	1.00
10	10	11.40 ± 0.92	10	11.52 ± 1.23	0.80	1.01

Table 10.2: Results of the delivery experiment, showing the number of successful trials (out of 10) and mean distance travelled with standard deviation for each route. In routes 2, 3, 4 and 6 the mean distances for the lost robot will be underestimates, indicated by $>$, because they do not include the failed trials. Also shown are the Student's t-test results, indicating the probability of the null hypothesis H_0 for each route that the mean distances are really the same, and the efficiency ratio of the mean distances for both systems.

10.2.1 Results

The effectiveness measures shown in table 10.2 show firstly that under "normal" operation using prior position knowledge, the navigation system is highly reliable, always managing to reach the desired goal locations. However, the lost robot was not quite as successful, being affected by various errors as it attempted to recover its position — the overall success rate of the lost robot was 92% compared to 100% for the control.

Perceptual aliasing or sensor noise could sometimes lead to the robot believing falsely that it had reached the goal location, when in fact more exploration was required to relocalise successfully. This tended to happen when the robot started to navigate from a location somewhere in the middle of a long, featureless corridor, for example, on route 6 in this experiment.

The robot could also become trapped in a "local minimum", moving backwards and forwards between two incorrectly identified places and taking a very long time to acquire sufficient sensory information to relocalise correctly (at the moment, the robot itself has no idea of whether it has produced the "correct" location estimate — see discussion). For example, two attempts at route 3 were abandoned after 10 minutes because the robot had still not managed to relocalise and find the goal. These errors can be explained by the fact that the robot's motor actions depend upon its current position estimate, so there is no guarantee that the robot will take appropriate actions to relocalise itself should it become lost.

The efficiency measures obtained are shown in the right-most column of figure 10.2. Note that the mean distances travelled for the lost robot will be underestimates in routes 2, 3, 4 and 6 because they do not include the failed trials. A standard Student's t-test (Press *et al.* 1992, p. 616) was conducted to determine the probability of obtaining the mean distances for each route, assuming the null hypothesis H_0 that the distances for both systems are really the same.

On the whole, the distance efficiency of the lost robot was as good as that of the control. A significant difference between the mean distances travelled was found only on routes 3 and 4 ($p_{H_0} < 0.05$). This can be explained by the fact that many of the perceptually aliased locations in the long corridors were assigned similar headings by the path planning algorithm. Thus, the robot often managed to perform the correct action, even though it had actually recognised the wrong location. Ballard & Whitehead (1992) distinguished "beneficial" perceptual aliasing, where the same actions are required at similar-looking locations, from "destructive aliasing", where different actions are required. In this particular experiment, the lost robot would seem to have been assisted by beneficial aliasing on several occasions, but hindered by destructive aliasing on others.

10.3 Discussion

In this chapter, the performance of the complete system for concurrent map building and self-localisation was evaluated, including its integration into a navigation system for a mobile robot. The experiments showed that under normal operation, the entire system is very reliable, enabling autonomous navigation by the robot between user-specified locations.

The integration experiments confirmed that the robot was able to relocalise itself reliably using a self-acquired map of the environment by wall-following. Following a canonical path through the environment guaranteed that the robot would eventually experience a unique sequence of perceptions, allowing successful relocalisation even if no single place in the robot's environment had a unique perceptual signature. However, in the complete navigation system as it stands, the robot's motor actions depend upon its current position estimate, so there is no guarantee that the robot will follow such a path and thus relocalise correctly should it become lost. This lead to the errors described in section 10.2.1.

One solution to this problem would be to choose actions designed to improve localisation quality, as in Fox *et al.* (1998) and Roy *et al.* (1999). Alternately, the robot could revert to wall-following in order to relocalise whenever it believed it might be lost. To detect when it might be lost, the robot would measure its confidence in its current location estimate, using the entropy-based measure given in equation 7.18. Recall that this measure decreases as the robot becomes more certain of its position. In the results given in figure 7.5, this measure reached a minimum of approximately $H(\mathcal{P}) = 1.7$. Therefore, to improve localisation quality, the robot would resume wall-following whenever $H(\mathcal{P}) > 1.7$.

Chapter 11

Summary and Conclusions

About this chapter. This chapter provides a summary of the main results in the thesis, followed by a description of the open questions and conclusions arising from the research conducted.

11.1 Summary of Results

This thesis presents a solution to the problem of concurrent map building and self-localisation by a self-navigating mobile robot operating in unknown, unmodified environments of middle-scale dimensions. In contrast to previous research, all of the environment and location models, feature models and sensor-motor competences required for navigation are acquired independently by the robot. As a result, the complete navigation system is able to operate in many different, real world environments which are initially unknown to the robot, without requiring intervention by a human operator or modifications to its surroundings.

The research began by looking at the sub-problem of self-localisation in isolation, assuming a pre-installed map provided by the system designer (chapters 6 and 7). After developing a successful self-localisation system, the sub-problem of autonomous map building by the robot was considered (chapters 8 and 9). System integration and validation experiments were then conducted to assess the performance of the complete system (chapter 10).

Due to the fundamental unreliability of navigation by dead reckoning, self-localisation based on external perceptual cues or "landmarks" was considered. In order to determine the best mechanism for landmark recognition on the Nomad 200 robot, an experimental procedure was developed to enable the comparison of different mechanisms under the same experimental conditions. An information-theoretic performance measure was used to assess localisation quality. It was found that a system for matching local occupancy grids produced the best results, but only at a computational cost which was prohibitive for real-time operation.

The results of the comparative study of landmark identification mechanisms were then used to guide the development of a complete self-localisation system for the Nomad 200. A new technique for matching local occupancy grids was developed, enabling precise positioning at a low computational cost. This was combined with a probabilistic algorithm for accumulating sensory evidence over time, enabling relocalisation under global uncertainty. Quantitative performance measures demonstrated the ability of the robot to recover from becoming lost in complex, middle-scale environments.

To enable autonomous map building by the robot, an exploration system was developed in which the robot's motor actions were guided by the current state of the robot's environment model. In this approach, the robot continuously expands the area which it has already mapped until the whole environment has been covered. To detect areas of uncharted territory, an artificial neural network was trained to learn the concept of open space. This approach fuses perceptual information from different types of sensors without requiring pre-installed world models.

In order to maintain geometric consistency in the robot's map, a relaxation algorithm was developed for assigning Cartesian coordinates to the places in the map. This algorithm is self-organising, using only local interactions and local odometric information to converge upon a globally optimal solution. In addition, the algorithm propagates the uncertainty in the robot's distance measurements through the map, this information being used in turn to improve the quality of self-localisation by the robot. A system integration experiment was conducted to measure the robot's ability to build its own map and then relocalise after becoming lost using the self-acquired map.

Finally, to assess the performance of the complete navigation system, a delivery task was considered in which the robot had to navigate between a series of arbitrary, user-chosen locations. Reliability and efficiency measures were used to assess the overall level of navigational competence.

11.2 Open Questions

11.2.1 Extensions to the Basic System

Improved Compass Sense

The basic compass sense consists of a simple behaviour designed to rotate the robot's turret in the direction of magnetic North according to a flux-gate compass (section 4.2). While this mechanism proved to be robust in the environments considered in this thesis, environments containing severe variations in the magnetic field, e.g., due to heavy machinery or ferrous building materials, could cause it to fail. On metal boats, there is a technique of installing heavy iron blocks symmetrically on either side of a compass, so lessening the effect of exterior ferrous material. Alternately, a more robust compass sense might be obtained by combining perceptual information from other sensor modalities.

Biological systems may use several redundant sources of sensory information for self-orientation. For example, the adult pigeon can use either the position of the sun or a geomagnetic compass sense (O'Keefe & Nadel 1977, p. 65). The position of the sun above the horizon is combined with information from an internal clock and an ephemeris function to calculate where the sun is located at that particular time of day. Alternately, when the sky is overcast, the pigeon will resort to information from its geomagnetic compass sense.

On the robot, the other sensor modalities which could be used would include sonar, infrared and odometry. One possible method of matching information from the robot's range-finder sensors would be to apply the technique for matching angle histograms constructed from detailed sensor scans developed by Hinkel & Knieriemen (1988). Alternately, the robot's camera (not used in this thesis) could be pointed upwards, either at the ceiling or a conical mirror, as in Franz *et al.* (1998), to obtain more detailed 360 degree sensory information.

Improved Sensor-Motor Control

This thesis was concerned primarily with the high-level algorithms required for self-localisation and map learning, and relatively little time was spent developing the mechanisms used for low-level sensor-motor control. The routines for obstacle avoidance and wall-following are quite coarse grained, and the robot cannot navigate through gaps of less than 1.2 m (the diameter of the robot is approximately

0.60 m). Navigation through narrow doorways is a difficult control problem which would require further investigation.

The progress of the robot in exploring new environments and carrying out navigation tasks can sometimes seem slow, due to the strategy of periodically stopping to take a new sensor scan. It would be relatively straightforward to integrate this sensing strategy with the other sensor-motor behaviours, rotating the robot's turret independently of the translational and rotational motors while the robot is in motion. This would speed up the whole navigation process and give the appearance of a more "intelligent" behaviour to the uninitiated observer.

More Rigorous Treatment of Uncertainty

Similarly, coarse grained techniques were used in the thesis for representing the uncertainty in the robot's sensor readings. For example, the model used to represent the noise in the robot's odometry measurements (section 4.3.2) is very simple. More accurate position information could be obtained by deriving a more elaborate noise model, e.g., using a covariance matrix (Smith *et al.* 1990).

However, its should be noted that no model would ever be likely to capture the true physics of robot sensors. For example, the covariant model of odometry drift used by Leonard & Durrant-Whyte (1992) does not take into account local variation in the environment; some areas of the floor might produce different levels of wheel slippage to others. This thesis has shown that reliable robot navigation is possible using simple sensor models which only capture an approximate or "naive" physics (Hayes 1979) of robot-environment interaction.

11.2.2 Active Selection of Landmarks

All of the landmark recognition systems considered in chapter 6 used a fixed distance metric, either in Cartesian space or some abstract perceptual space, to determine when to add new landmarks to the robot's map. This strategy could be inefficient, as some parts of the environment might contain richer perceptual cues than others.

Simhon & Dudek (1998) proposed a mapping scheme in which only the most "distinctive" regions of the environment were represented in full using local metric maps. These regions were linked topologically in the robot's global map by fuzzy, semi-unknown areas which were not represented in any detail. The "distinctive" regions were chosen according to their reliability for performing accurate localisation within the local metric representation.

An alternative approach has recently been investigated by Duckett & Nehmzow (1999); see Nehmzow *et al.* (2000) for some further results. Here, the most distinctive landmarks are seen as those which are the most unusual or "surprising", and therefore the least predictable to the robot as it explores the environment. A simple neural network was trained off-line to predict the robot's next sensory perception based on the immediate, preceding perception. During on-line operation, the prediction error was calculated at the following time step from the difference between the predicted and actual sensor readings. Smoothing functions were then applied to the series of error values generated, and the perceptions which produced local maxima in the smoothed error curve were taken as candidate landmarks for inclusion in the robot's map.

These approaches are certainly much closer in spirit to the cognitive mapping schemes which have been postulated for human navigation (see e.g., Yeap (1988) and citations therein), and to the notion of a qualitative topological map as depicted in figure 1.3.

11.2.3 Lifelong Learning

In this thesis, the environments were assumed to be semi-structured so that the robot could construct a complete map using a single tour of the target environment. This assumption would fail if the environment was subject to structural changes, for example, if the robot's path became blocked during subsequent navigation[1].

Extending the system to enable continuous adaptation of the robot's environment model over time would be fairly straight-forward — in fact, the algorithms for map learning presented in chapter 8 would already support this. The map uses variable-confidence links to model the uncertainty in the perceived topological relations in the environment, as in Yamauchi & Beer (1996), and the relaxation algorithm (section 8.4.1) continuously adapts the whole map in order to maintain a globally consistent representation.

[1]if the robot's path did become blocked, the system as its stands might well be fooled into believing that it was in a different location

In this approach, map learning would be carried out continuously during normal operation. Whenever the robot had no particular task to perform in its target application, exploration would be resumed, repeatedly traversing the previously mapped territory to look for changes to the environment.

However, in some situations a dynamic environment might lead to localisation errors and would therefore produce errors in the map during concurrent map building and self-localisation. One solution to this problem would be maintain multiple versions of the environment model, as in Cox & Leonard (1994), corresponding to different possible interpretations of the robot's sensor data. The "correct" map could then survive temporary localisation errors and re-emerge as the most likely environment model upon subsequent exploration. According to O'Keefe & Nadel (1977, p. 96):

> "The updating of maps does not imply that the old map is literally erased. Some representation of every experienced state of the environment must be maintained, along with information as to which representation is current and which is no longer so."

There would be several issues to address here, including on which version of the map to base decisions, where to move and computational tractability. The main disadvantage of Cox and Leonard's approach is the exponential growth in the tree of possible maps. Expectation maximisation (EM) algorithms can also be used to backwardly re-estimate the most likely environment model based on new, contradictory sensor information (Thrun *et al.* 1998b), but they suffer from local maxima as well as high computational cost. Some fast, on-line method of resolving these conflicts would be required.

A further extension would be to introduce on-line learning of system parameters — at the moment, the system contains a number of parameters, such as the distance threshold for adding new places to the map, which are determined manually by the system designer. Adapting the parameters on-line would require some feedback signal to indicate the success of the current set of parameters in achieving the system's navigational objectives, e.g., efficiency-based measures of the distance travelled to a goal location could be used.

11.2.4 Closing the Loop

A fundamental problem for any navigating robot is to build consistent maps in environments containing very large loops. Here, the robot's self-localisation mechanism needs to recognise when a previously visited location has been reached once the robot has made a complete circuit of a loop, otherwise the robot will keep on adding new, duplicate copies of the same physical locations to the map *ad infinitum*. This cannot be achieved using exteroception alone, because of perceptual aliasing, but equally cannot be achieved using proprioception, because of the inevitable drift errors.

Humans can typically only solve the loop closing problem in large, complex environments by reading signs, such as street names or door numbers. However, since artificial "markers" are excluded from our definitions of "autonomous navigation" and "self-navigation", this alternative was not considered in this thesis.

So far, navigating mobile robots have only been able to "close the loop" by using accurate range-finding sensors and precisely corrected odometry. For example, the system described by Thrun (1998b) has successfully mapped a circular route of 160 m using laser range-finder sensors and a high resolution metric map. However, this approach will inevitably fail once the size of the environment is increased beyond the accuracy limits of the robot's dead reckoning mechanism. Gutmann & Konolige (1999) have recently proposed a method for mapping cyclic environments which combines the Lu and Milios algorithm for obtaining geometrically consistent metric maps (1997a) with a variant of Markov localisation for recognising when the starting location has been reached once more. However, again this method relies ultimately on the accuracy of dead reckoning, because it will fail if the wrong location is identified as the start location. A manual method of closing the loop using retrospectively corrected odometry was introduced in section 4.3.3, but my attempts at autonomously mapping an environment containing a circuit of length 190 m with *FortyTwo* have so far been unsuccessful.

It is perhaps unrealistic to expect a "perfect" solution to this problem, particularly when biological systems can fail at closing the loop too. For example, lost humans can travel for long distances without realising that they have returned to a previously visited location. Conversely, perceptual aliasing can lead to identification of the wrong place as the starting location. The solution must lie in using all of the available perceptual information, and finding appropriate noise models to ensure that previously visited locations can possibly be recognised after travelling long distances. Again, it may be necessary to maintain multiple versions of the map, since further exploration might be needed to correct mapping errors. The complete map building and self-localisation system developed in this thesis has great potential here, because the efficiency of its representations and algorithms means that it could easily be generalised to maintain many alternative versions of the robot's environment and location models at the same time.

11.2.5 Universal Navigation Architecture

A further open question concerns the generality of the mechanisms developed in this thesis to work on other robot platforms. The mechanisms for landmark identification evaluated in chapter 6 were all based on a holonomic robot design with sonar sensors spaced at equal intervals around the robot's turret, so it is unlikely that all of these methods would work with non-holonomic robots. However, the important point is that a standard experimental procedure was used to select the best landmark identification mechanism for one particular robot. Future work could investigate automating this process (perhaps using evolutionary learning) so that the best landmarks could be selected automatically for an *a priori* unknown robot morphology.

At present, developing navigation software for mobile robots is an extremely labour intensive process, usually requiring a complete redesign from scratch for each new robot. Widespread use of mobile robots is unlikely to occur until generic software for navigation becomes available. However, as the technology progresses, it is likely that robot hardware in turn will change, making existing control software obsolete.

A solution to this problem would be to develop a universal software architecture for robot navigation, which would be applicable to many different robots, including automatic guided vehicles, robotic wheelchairs, domestic service robots, etc. This would consist of a common core, perhaps based on the probabilistic algorithms for self-localisation and map learning developed in this thesis, together with the facility to "plug in" different sensor models and sensor-motor competences for different robots. Learning would be utilised at every level of the architecture, so that the system could adapt itself to work with almost any present or future robot platform.

In this approach, there would be an initial learning phase, during which primary sensor-motor skills and feature detectors would be acquired through self-organisation and self-supervised learning (or reinforcement learning when only a positive or negative feedback signal is available for training). This might be followed by a secondary learning phase, in which the internal representations of the robot would be adapted for a specific application. During the final deployment of the system, the robot would then begin concurrent map building and self-localisation in the target environment. After completing the initial learning of the environment, the desired function of the navigation system would be similar to that of an operating system on a stand-alone PC, monitoring the progress of its user applications and being responsible for executing required actions, whilst running in the background and consuming minimal resources.

11.3 Conclusion

Levitt & Lawton (1990) defined the task of navigation by the three questions: "Where am I?", "Where are other places relative to me?" and "How do I get to other places from here?". In this thesis, these questions were answered through the development of a complete navigating robot equipped with concurrent, interdependent mechanisms for self-localisation (question 1), map learning (question 2), exploration and way finding (question 3). Furthermore, this was achieved with minimal pre-installation of world knowledge by the system designer. Nehmzow (1992, p. 196) states that

> "People as designers of robot controllers do not have the experience of a robot, undeniably so, which makes its impossible for them to reliably determine all the features a robot will require *a priori*. The more dependent on designer-defined knowledge a robot controller is, the less flexible it will inevitably be when in operation."

The robot developed in this thesis can be placed in an *a priori* unknown environment, build its own map through free exploration and then use this map for navigation, without requiring external assistance or modifications to the target environment. The representations and algorithms developed are so computationally cheap that self-navigation in middle-scale environments was achieved using only *FortyTwo*'s own internal 486 processor. To the best of my knowledge, it is the only navigating mobile robot so far with the ability to operate from scratch in unmodified, populated environments of middle-scale dimensions using only on-board computation and its own self-acquired models for feature detection, sensor-motor control, map learning and self-localisation.

At the same time, the system as it stands stills suffers from many of the limitations common to all current mobile robots. Place it in a building site, and it would happily trundle through any unglazed window frames towards an untimely death, with no self-awareness or realisation of the consequences of its actions. We could fix the immediate problem by providing the robot with downward pointing sensors and a new control program (preferably self-acquired, personal injury to the robot notwithstanding) to detect this danger – some robots are indeed equipped with such sensors. However, the basic underlying problem would remain, namely that currents robots lack the means for dealing with situations unforeseen by the system designer. Clearly, far more sophisticated techniques for robot learning are required, which go beyond the current paradigm of generalisation on examples and include other forms of inference and reasoning. To be truly successful, I believe that mobile robotics research must integrate itself with many other areas of AI research, including machine learning, vision, planning and natural language processing, to name but a few, if robots are to operate with any real degree of autonomy within human-oriented applications.

The new navigation system was developed only after conducting numerous experiments with robot *FortyTwo* over middle-scale distances, including the experimental procedure described in chapter 5. Some of the early prototyping work was done first on the Nomadic simulator, but only the experiments on the real robot could confirm whether this work would transfer to the real world. This is not particularly surprising, given that no simulator can capture the true complexity of robot-environment interaction; models can only ever approximate the real world. This experience confirmed my earlier argument in section 2.1.1 on why we should be interested in building complete robots navigating over middle-scale environments.

However, the "existence proof" of building a complete navigating robot does not in itself constitute a meaningful contribution to the scientific study of robotic systems. According to a very famous quotation by William Thomson (Lord Kelvin), taken from MacHale (1993),

"When you measure what you are speaking about and express it in

> numbers, you know something about it, but when you cannot express it in numbers your knowledge about it is of a meagre and unsatisfactory kind."

Without quantitative metrics, it would be very difficult for roboticists to evaluate performance, to compare competing theories, and to analyse and formulate hypotheses about robot behaviour. As yet, however, there have been very few studies which attempt to quantify robot-environment interactions or make experimental comparisons of navigating robots (see chapter 5 for some examples). In short, measures of robotic performance would serve to increase our understanding of the mechanisms which underlie intelligent behaviour, and thus to advance the state-of-the-art in real world applications of robotics.

I believe that the quantitative measures required to describe robotic systems are already available in computer science, particularly from Shannon's mathematical theory of communication ~~theory~~ ("information theory") (Shannon & Weaver 1949). My view is that the fundamental processes of robot-environment interaction, robot-robot interaction and human-robot interaction could all be formulated as a communication process involving a transfer of information between the respective agents and the robot controller. For example, a sensor can be seen as a noisy communication channel which transmits information from the robot-environment system to the controller. Conversely, actuation can be seen as the reverse process, resulting in a change of state in the robot-environment system.

Robot-environment interaction is to some extent a "black box" — we can know precisely what goes into the robot controller and measure its effect, but we cannot isolate the function of the robot controller from that of its interaction with the real world. Lelas (1993, p. 425) points to the situation in quantum physics in which theory cannot avoid reference to experimental arrangements. Similarly, noise and variations are an inherent and vital part of robot-environment interaction which cannot be excluded from any theory of autonomous mobile robot navigation.

In this thesis, an information-theoretic performance metric was used to assess both localisation quality and map quality, by measuring the information content of a robot's responses in predicting its true location. This approach allowed a navigating mobile robot be studied within a series of middle-scale environments, including all of the phenomena associated with using real sensors and real actuators in the real world. The influence of individual system parameters and sub-components was assessed through controlled experiments. As such, this thesis represents a case study in "quantitative robotics"; that is, an application of quantitative measures of robot-environment interaction to the design, testing and validation of a complete navigating mobile robot. I hope that this will be a useful contribution towards the science of mobile robotics.

Bibliography

[Arkin 1998] Arkin, R. C. 1998. *Behavior-Based Robotics.* Cambridge, MA: MIT Press.

[Balkenius & Kopp 1996] Balkenius, C., and Kopp, L. 1996. The XT-1 vision architecture. In *Proceedings of the Symposium on Image Analysis, Lund University, Sweden*, 39–43.

[Ballard & Whitehead 1992] Ballard, D., and Whitehead, S. 1992. Learning visual behaviours. In Wechsler, H., ed., *Neural Networks for Perception.* New York: Academic Press.

[Beetz *et al.* 1998] Beetz, M.; Burgard, W.; Fox, D.; and Cremers, A. 1998. Integrating active localisation into high-level robot control systems. *Robotics and Autonomous Systems* 23(4).

[Bennet 1996] Bennet, A. 1996. Do animals have cognitive maps? *Journal of Experimental Biology* 199:219–224.

[Bethell 1996] Bethell, A. 1996. A hippocampally-inspired connectionist model for mobile robot localisation. Master's thesis, Department of Computer Science, University of Manchester.

[Bishop 1995] Bishop, C. 1995. *Neural Networks for Pattern Recognition.* Oxford, England: Oxford University Press.

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

[Brooks 1986] Brooks, R. 1986. A robust layered control system for a mobile robot. Technical Report 864, Massachusetts Institute of Technology.

[Burgard *et al.* 1996] Burgard, W.; Fox, D.; Henning, D.; and Schmidt, T. 1996. Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, 896–901.

[Burgard *et al.* 1998a] Burgard, W.; A.B.; Cremers; Fox, D.; Hähnel, D.; Lakemeyer, G.; Schulz, D.; Steiner, W.; and Thrun, S. 1998a. The interactive museum tour-guide robot. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98)*.

[Burgard *et al.* 1998b] Burgard, W.; Derr, A.; Fox, D.; and Cremers, A. B. 1998b. Integrating global position estimation and position tracking for mobile robots: The dynamic Markov localization approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*.

[Burgess *et al.* 1997] Burgess, N.; Donnett, J.; Jeffery, K.; and O'Keefe, J. 1997. Robotic and neuronal simulation of the hippocampus and rat navigation. *Philosophical Transactions of the Royal Society B* 352.

[Carpenter & Grossberg 1987] Carpenter, G., and Grossberg, S. 1987. ART2 : Self–organization of stable category recognition codes for analog input patterns. *Applied Optics* 26(23):4919–4930.

[Cassandra *et al.* 1996] Cassandra, A.; Kaelbling, L. P.; and Kurien, J. 1996. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'96)*.

[Courtney & Jain 1994] Courtney, J. D., and Jain, A. K. 1994. Mobile robot localization via classification of multisensor maps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'94)*.

[Cox & Leonard 1994] Cox, I. J., and Leonard, J. J. 1994. Modeling a dynamic environment using a Bayesian multiple hypothesis approach. *Artificial Intelligence* 66:311–344.

[Cox 1991] Cox, I. 1991. Blanche - an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation* 7:193–204.

[Crowley 1995] Crowley, J. L. 1995. Mathematical foundations of navigation and perception for an autonomous mobile robot. In *Workshop on Reasoning With Uncertainty in Robotics.* Tutorial paper.

[Dellaert *et al.* 1999] Dellaert, F.; Fox, D.; Burgard, W.; and Thrun, S. 1999. Monte Carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'99).*

[Dennett 1998] Dennett, D. 1998. *Brainchildren.* Penguin.

[Donnett 1993] Donnett, J. 1993. *Analysis and Synthesis in the Design of Locomotor and Spatial Competences for a Multisensory Mobile Robot.* Ph.D. Dissertation, Edinburgh University.

[Dorigo & Columbetti 1998] Dorigo, M., and Columbetti, M. 1998. *Robot Shaping: an experiment in behavior engineering.* MIT Bradford Press.

[Duckett & Nehmzow 1996] Duckett, T., and Nehmzow, U. 1996. A robust perception-based localisation method for a mobile robot. Technical Report UMCS-96-11-1, Department of Computer Science, University of Manchester.

[Duckett & Nehmzow 1997a] Duckett, T., and Nehmzow, U. 1997a. Experiments in evidence based localisation for a mobile robot. In *Proceedings of the AISB'97 Workshop on Spatial Reasoning in Animals and Robots*, 25–34. Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-97-4-1.

[Duckett & Nehmzow 1997b] Duckett, T., and Nehmzow, U. 1997b. Knowing your place in the real world. In *ECAL-97 Fourth European Conference on Artificial Life.*

[Duckett & Nehmzow 1998] Duckett, T., and Nehmzow, U. 1998. Mobile robot self-localisation and measurement of performance in middle scale environments. *Robotics and Autonomous Systems* 24(1–2):57–69.

[Duckett & Nehmzow 1999] Duckett, T., and Nehmzow, U. 1999. Learning to predict sonar readings for mobile robot landmark selection. Report of the 4th visit in the Academic Research Collaboration (ARC) funded by the British Council and Deutscher Akademischer Austauschdienst, June 24 to July 13 1999, Bremen University, Germany.

[Duda & Hart 1973] Duda, R., and Hart, P. 1973. *Pattern Classification and Scene Analysis.* J. Wiley and Sons.

[Edlinger & Weiss 1995] Edlinger, T., and Weiss, G. 1995. Exploration, navigation and self-localisation in an autonomous mobile robot. In *Autonomous Mobile Systems.*

[Elfes 1987] Elfes, A. 1987. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation* RA-3(3):249–265.

[Engelson & McDermott 1992] Engelson, S. P., and McDermott, D. V. 1992. Error correction in mobile robot map learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'92)*, 2555–2560.

[Engelson 1994] Engelson, S. 1994. *Passive Map Learning and Visual Place Recognition.* Ph.D. Dissertation, Department of Computer Science, Yale University.

[Fox *et al.* 1998] Fox, D.; Burgard, W.; and Thrun, S. 1998. Active Markov localisation for mobile robots. *Robotics and Autonomous Systems* 25:195–207.

[Franz *et al.* 1998] Franz, M.; Schoelkopf, B.; Mallot, H.; and Buelthoff, H. 1998. Learning view graphs for robot navigation. *Autonomous Robots* 5:111–125.

[Fritzke 1994] Fritzke, B. 1994. Growing cell structures - a self-organizing network for unsupervised and supervised learning. *Neural Networks* 7(9):1441–1460.

[Fritzke 1995] Fritzke, B. 1995. A growing neural gas network learns topologies. In Tesauro, G.; Touretzky, D. S.; and Leen, T. K., eds., *Advances in Neural Information Processing Systems 7*, 625–632. Cambridge, MA: MIT Press.

[Gallistel 1990] Gallistel, C. R. 1990. *The Organization of Learning.* Cambridge, MA: MIT Press.

[Gasós & Saffiotti 1999] Gasós, J., and Saffiotti, A. 1999. Integrating fuzzy geometric maps and topological maps for robot navigation. In *Proceedings of the 3rd International ICSC Symposium on Soft Computing (SOCO'99)*, 754–760.

[Gat 1998] Gat, E. 1998. Three layer architectures. In Kortenkamp, D.; Bonasso, R.; and Murphy, R., eds., *AI-based Mobile Robots: Case Studies of Successful Robot Systems.* Cambridge, MA: MIT Press.

[Gelb 1974] Gelb, A. 1974. *Applied Optimal Estimation.* Cambridge, MA: MIT Press.

[Gibbons 1985] Gibbons, A. 1985. *Algorithmic Graph Theory.* Cambridge, England: Cambridge University Press.

[Golfarelli *et al.* 1998] 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)*, 905–911.

[Grossberg 1988] Grossberg, S. 1988. *Neural Networks and Natural Intelligence.* MIT Bradford Press.

[Gutmann & Konolige 1999] 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.*

[Gutmann & Schlegel 1996] Gutmann, J.-S., and Schlegel, C. 1996. AMOS : Comparison of scan matching approaches for self-localization in indoor environments. In *Proc. EUROBOT '96, 1st European Workshop on Advanced Mobile Robots.* IEEE Computer Society.

[Gutmann *et al.* 1998] Gutmann, J.-S.; Burgard, W.; Fox, D.; and Konolige, K. 1998. An experimental comparison of localization methods. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98).*

[Hayes 1979] Hayes, P. 1979. The naive physics manifesto. In Boden, M., ed., *The Philosophy of Artificial Intelligence.* Oxford, England: Oxford University Press.

[Hertzberg & Kirchner 1997] Hertzberg, J., and Kirchner, F. 1997. Landmark-based autonomous navigation in sewerage pipes. In *Proc. EUROBOT '97, 2nd European Workshop on Advanced Mobile Robots.*

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

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

[Hughes & Murphy 1992] Hughes, K., and Murphy, R. 1992. Ultrasonic robot localization using dempster-shafer theory. In *Neural and Stochastic Methods in Image and Signal Processing.* SPIE Vol. 1766.

[Janet *et al.* 1995] Janet, J.; Gutierrez-Osuna, R.; Chase, T.; White, M.; and Luo, R. 1995. Global self-localization for autonomous mobile robots using self-organizing Kohonen neural networks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'95).*

[Jensfelt *et al.* 2000] Jensfelt, P.; Austin, D.; Wijk, O.; and Anderson, M. 2000. Feature based condensation for mobile robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*

[Kaelbling *et al.* 1998] Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101.

[Keuchal *et al.* 1993] Keuchal, H.; von Puttkamer, E.; and Zimmer, U. R. 1993. SPIN: Learning and forgetting surface classification with dynamic neural networks. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'93).*

[King & Weiman 1990] King, S., and Weiman, C. 1990. Helpmate autonomous mobile robot navigation system. In *Proceedings of the SPIE Conference on Mobile Robots*, 190–198.

[Koenig & Simmons 1996] Koenig, S., and Simmons, R. 1996. Unsupervised learning of probabilistic models for robot navigation. In *Proceedings of the IEEE International Conference Robotics and Automation (ICRA'96).*

[Koenig & Simmons 1998] Koenig, S., and Simmons, R. 1998. Xavier: A robot navigation architecture based on partially observable markov decision process models. In Kortenkamp, D.; Bonnasso, R.; and Murphy, R., eds., *Artificial Intelligence Based Mobile Robots: Case Studies of Successful Robot Systems.* MIT Press.

[Koenig *et al.* 1996] Koenig, S.; Goodwin, R.; and Simmons, R. 1996. Robot navigation with markov models: A framework for path planning and learning with limited computational resources. In Dorst, L.; van Lambalgen, M.; and Voorbraak, L., eds., *Lecture Notes in Artificial Intelligence*, volume 1093. Springer, Berlin. 322–337.

[Kohonen 1993] Kohonen, T. 1993. *Self-Organization and Associative Memory, 3rd ed.* Springer, Berlin.

[Kortenkamp & Weymouth 1994] Kortenkamp, D., and Weymouth, T. 1994. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the Twelth National Conference on Artificial Intelligence (AAAI'94)*, 979–984.

[Kortenkamp *et al.* 1998] Kortenkamp, D.; Bonasso, R. P.; and Murphy, R. 1998. *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems.* Cambridge, MA: MIT Press.

[Kortenkamp 1993] Kortenkamp, D. 1993. *Cognitive Maps for Mobile Robots: A Representation for Mapping and Navigation.* Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, University of Michigan.

[Kuipers & Byun 1991] Kuipers, B., and Byun, Y. 1991. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representation. *Robotics and Autonomous Systems* 8:47–63.

[Kunz *et al.* 1999] Kunz, C.; Willeke, T.; and Nourbakhsh, I. 1999. Automatic mapping of dynamic office environments. *Autonomous Robots, in print.*

[Kurz 1996] Kurz, A. 1996. Constructing maps for mobile robot navigation based on ultrasonic range data. *IEEE Transactions on Systems, Man and Cybernetics — Part B: Cybernetics* 26(2):233–242.

[Lambrinos *et al.* 1995] Lambrinos, D.; Scheier, C.; and Pfeifer, R. 1995. Unsupervised classification of sensory-motor states in a real world artifact using a temporal Kohonen map. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'95).*

[Lee & Recce 1997] Lee, D., and Recce, M. 1997. Quantitative evaluation of the exploration strategies of a mobile robot. *International Journal of Robotics Research* 16(4):413–447.

[Lee *et al.* 1998] Lee, T.-m.; Nehmzow, U.; and Hubbold, R. 1998. Mobile robot simulation by means of acquired neural network models. In *European Simulation Multiconference.*

[Lee 1995] Lee, D. C. 1995. *The Map-Building and Exploration Strategies of a Simple, Sonar-Equipped, Mobile Robot; An Experimental, Quantitative Evaluation.* Ph.D. Dissertation, University College London.

[Lee 1996] Lee, W. Y. 1996. *Spatial Semantic Hierarchy for a Physical Robot.* Ph.D. Dissertation, The University of Texas at Austin.

[Lelas 1993] Lelas, S. 1993. Science as technology. *British Journal of Philosphical Science* 44:423–442.

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

[Leonard *et al.* 1990] Leonard, J.; Durrant-Whyte, H.; and Cox, I. 1990. Dynamic map building for an autonomous mobile robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'90)*, 89–96.

[Levitt & Lawton 1990] Levitt, T., and Lawton, D. 1990. Qualitative navigation for mobile robots. *Artificial Intelligence* 44:305–360.

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

[Linsker 1988] Linsker, R. 1988. Self-organization in a perceptual network. *IEEE Computer* 21(3):105–117.

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

[Lu & Milios 1997b] Lu, F., and Milios, E. 1997b. Robot pose estimation in unknown environments by matching 2D range scans. *Intelligent and Robotics Systems* 18:249–275.

[MacHale 1993] MacHale, D. 1993. *Comic Sections.* Boole Press.

[MacKenzie & Dudek 1994] MacKenzie, P., and Dudek, G. 1994. Precise positioning using model-based maps. In *Proceedings of the International Conference on Robotics and Automation (ICRA'94)*, 1615–1621. Los Alamitos, CA: IEEE Computer Society Press.

[Mahadevan *et al.* 1998] Mahadevan, S.; Theocharous, G.; and Khaleeli, N. 1998. Rapid concept learning for mobile robots. *Autonomous Robots* 5.

[Martin & Nehmzow 1995] Martin, P., and Nehmzow, U. 1995. "Programming" by teaching: Neural network control in the Manchester mobile robot. In *Proceedings Intelligent Autonomous Vehicles.*

[Martin 1966] Martin, H. 1966. *Introduction to Matrix Methods of Structural Analysis.* McGraw-Hill.

[Mataric 1991] Mataric, M. 1991. Navigating with a rat brain: A neurobiologically inspired model for robot spatial representation. In *From Animals to Animats 1.*

[Maybeck 1990] Maybeck, P. S. 1990. The Kalman filter: An introduction to concepts. In Cox, I., and Wilfong, P., eds., *AI-based Mobile Robots: Case Studies of Successful Robot Systems.* Springer, Berlin. 194–204.

[McNaughton *et al.* 1996] McNaughton, B.; Barnes, C.; Gerrard, J.; Gothard, W.; Jung, M.; Knierim, J.; Kudrimoti, H.; Qin, Y.; Skaggs, W.; Suster, M.; and Weaver, K. 1996. Deciphering the hippocampal polyglot: The hippocampus as a path integration system. *Journal of Experimental Biology* 199:173–185.

[Merriam-Webster 1999] Merriam-Webster. 1999. *WWWebster Dictionary.* `http://www.m-w.com/netdict.htm.`

[Moravec & Elfes 1985] 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)*, 116–121. St. Louis, Missouri: IEEE Computer Society Press.

[Morellas *et al.* 1995] Morellas, V.; Minners, J.; and Donath, M. 1995. Implementation of real time spatial mapping in robotic systems through self-organizing neural networks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'95).*

[Morris *et al.* 1982] Morris, R.; Garraud, P.; Rawlins, J.; and O'Keefe, J. 1982. Place-navigation impaired in rats with hippocampal lesions. *Nature* 297:681–683.

[Nehmzow & Mitchell 1995] Nehmzow, U., and Mitchell, T. 1995. The prospective student's introduction to the robot learning problem. Technical report, Department of Computer Science, University of Manchester.

[Nehmzow & Smithers 1992] Nehmzow, U., and Smithers, T. 1992. Using motor actions for location recognition. In Varela, F., and Bourgine, P., eds., *Toward a Practice of Autonomous Systems.* MIT Press, Cambridge MA.

[Nehmzow *et al.* 1991] Nehmzow, U.; Smithers, T.; and Hallam, J. 1991. Location recognition in a mobile robot using self-organising feature maps. In Schmidt, G., ed., *Information Processing in Autonomous Mobile Robots.* Springer Verlag.

[Nehmzow *et al.* 2000] Nehmzow, U.; Gelder, D.; and Duckett, T. 2000. Automatic selection of landmarks for mobile robot navigation. Technical Report UMCS-00-7-1, Department of Computer Science, University of Manchester.

[Nehmzow 1992] Nehmzow, U. 1992. *Experiments in Competence Acquisition for Autonomous Mobile Robots.* Ph.D. Dissertation, Department of Artificial Intelligence, University of Edinburgh.

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

[Nehmzow 1997] Nehmzow, U. 1997. Scientific methods in mobile robotics?! In *Proceedings of TIMR'97, Towards Intelligent Mobile Robots, Manchester.* Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-97-9-1.

[Netlib 1999] Netlib. 1999. *The Performance Database Server.* `http://performance.netlib.org/performance`.

[Nilsson 1980] Nilsson, N. 1980. *Principles of Artificial Intelligence.* Palo Alto, CA: Tioga.

[Nourbakhsh 1998] Nourbakhsh, I. 1998. Dervish: An office-navigating robot. In Kortenkamp, D.; Bonasso, R.; and Murphy, R., eds., *AI-based Mobile Robots: Case Studies of Successful Robot Systems.* Cambridge, MA: MIT Press. 73–90.

[O'Keefe & Nadel 1977] O'Keefe, J., and Nadel, L. 1977. *The Hippocampus as a Cognitive Map.* Oxford, England: Oxford University Press.

[Oore *et al.* 1997] Oore, S.; Hinton, G. E.; and Dudek, G. 1997. A mobile robot that learns its place. *Neural Computation* 9:683–699.

[Oriolo *et al.* 1998] Oriolo, G.; Ulivi, G.; and Vendittelli, M. 1998. Real-time map building and navigation for autonomous robots in unknown environments. *IEEE Transactions on Systems, Man and Cybernetics* 28(3):316–333.

[Owen & Nehmzow 1997] Owen, C., and Nehmzow, U. 1997. Middle scale navigation - a case study. In *Proc. AISB 97 Workshop on Spatial Reasoning in Animals and Robots.* Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-97-4-1.

[Pagac *et al.* 1996] Pagac, D.; Nebot, E. M.; and Durrant-Whyte, H. 1996. An evidential approach to probabilistic map building. In *Reasoning with Uncertainty in Robotics*, number 1093 in Lecture Notes in AI, 164–170. Springer.

[Pomerleau 1989] Pomerleau, D. 1989. ALVINN: An autonomous land vehicle in a neural network. In Touretzky, D., ed., *Advances in Neural Information Processing Systems 1.* Morgan Kaufman.

[Press *et al.* 1992] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; and Flannery, B. P. 1992. *Numerical Recipes in C, 2nd. edition.* Cambridge, England: Cambridge University Press.

[Rabiner 1989] Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2).

[Racz & Dubrawski 1995] Racz, J., and Dubrawski, A. 1995. Artificial neural network for mobile robot topological localisation. *Robotics and Autonomous Systems* 16:73–80.

[Reily *et al.* 1982] Reily, D.; Cooper, L.; and Elbaum, C. 1982. A neural model for category learning. *Biological Cybernetics* 45:34–51.

[Roy *et al.* 1999] Roy, N.; Burgard, W.; Fox, D.; and Thrun, S. 1999. Coastal navigation: Robot navigation under uncertainty in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '99).*

[Saffiotti 1997] Saffiotti, A. 1997. The uses of fuzzy logic in autonomous robot navigation: A catalogue raisonne. *Soft Computing* 1(4):180–197.

[Schöner & Dose 1992] Schöner, G., and Dose, M. 1992. A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems* 10:253–267.

[Schöner *et al.* 1995] Schöner, G.; Dose, M.; and Engels, C. 1995. Dynamics of behavior: theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems* 16:213–245.

[Schultz & Adams 1998] Schultz, A., and Adams, W. 1998. Continuous localization using evidence grids. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98).*

[Shannon & Weaver 1949] Shannon, C., and Weaver, W. 1949. *The Mathematical Theory of Communication.* University of Illinois Press.

[Shatkay & Kaelbling 1997] 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.*

[Shatkay 1998] Shatkay, H. 1998. *Learning Models for Robot Navigation.* Ph.D. Dissertation, Department of Computer Science, Brown University.

[Simhon & Dudek 1998] Simhon, S., and Dudek, G. 1998. Selecting targets for local reference frames. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98)*, 2840–2845.

[Simmons & Koenig 1995] Simmons, R., and Koenig, S. 1995. Probabilistic robot navigation in partially observable environments. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence.*

[Smith *et al.* 1990] Smith, R.; Self, M.; and Cheeseman, P. 1990. Estimating uncertain spatial relationships in robotics. In Cox, I., and Wilfong, P., eds., *AI-based Mobile Robots: Case Studies of Successful Robot Systems.* Springer, Berlin. 167–193.

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

[Steels 1993] Steels, L. 1993. *Building agents out of autonomous behaviour systems.* New Haven: Lawrence Erlbaum.

[Stevens *et al.* 1995] Stevens, A.; Stevens, M.; and Durrant-Whyte, H. 1995. 'OxNav': Reliable autonomous navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (IROS'95)*, 2607–2612.

[Thrun *et al.* 1998a] Thrun, S.; Bücken, A.; Burgard, W.; Fox, D.; Fröhlinghaus, T.; Henning, D.; Hofmann, T.; Krell, M.; and Schmidt, T. 1998a. Map learning and high-speed navigation in RHINO. In Kortenkamp, D.; Bonasso, R.; and Murphy, R., eds., *AI-based Mobile Robots: Case Studies of Successful Robot Systems.* Cambridge, MA: MIT Press.

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

[Thrun *et al.* 1999] Thrun, S.; Benewitz, M.; Burgard, W.; Cremers, A. B.; Dellaert, F.; Fox, D.; Haehnel, D.; Rosenberg, C.; Roy, N.; Schulte, J.; and Schulz, D. 1999. MINERVA: A second-generation museum tour-guide robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'99).*

[Thrun *et al.* 2000] Thrun, S.; Fox, D.; and Burgard, W. 2000. Monte Carlo localization with mixture proposal distribution. In *Proceedings of the National Conference on Artificial Intelligence (AAAI).*

[Thrun 1998a] Thrun, S. 1998a. Bayesian landmark learning for mobile robot localisation. *Machine Learning* 33(1).

[Thrun 1998b] Thrun, S. 1998b. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence* 99.

[Touretzky *et al.* 1994] Touretzky, D.; Wan, H.; and Redish, D. 1994. Neural representation of space in rats and robots. In Zurada, J.; Marks, R.; and Robinson, C., eds., *Computational Intelligence: Imitating Life*, 57–68. IEEE Press.

[Wehner 1996] Wehner, R. 1996. Middle-scale navigation: The insect case. *Journal of Experimental Biology* 199.

[Weiss & von Puttkamer 1995] 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*, 403–407.

[Yamauchi & Beer 1996] Yamauchi, B., and Beer, R. 1996. Spatial learning for navigation in dynamic environments. *IEEE Transactions on Systems, Man and Cybernetics Section B* 26(3).

[Yamauchi & Langley 1997] Yamauchi, B., and Langley, P. 1997. Place recognition in dynamic environments. *Journal of Robotic Systems, Special Issue on Mobile Robots* 14(2).

[Yamauchi *et al.* 1997] Yamauchi, B.; Schultz, A.; Adams, W.; Graves, K.; Grefenstette, J.; and Perzanowski, D. 1997. ARIEL : Autonomous robot for integrated exploration and localization. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 804–805. extended abstract.

[Yamauchi *et al.* 1998] Yamauchi, B.; Schultz, A.; and Adams, W. 1998. Mobile robot exploration and map-building with continuous localisation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'98)*.

[Yamauchi 1997] Yamauchi, B. 1997. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation.*

[Yeap 1988] Yeap, W. 1988. Towards a computational theory of cognitive maps. *Artificial Intelligence* 34(3):297–360.

[Zelinsky 1992] Zelinsky, A. 1992. Mobile robot exploration algorithm. *IEEE Robotics and Automation* 8(6).

[Zimmer 1995a] Zimmer, U. R. 1995a. *Adaptive Approaches to Basic Mobile Robot Tasks.* Ph.D. Dissertation, University of Kaiserslautern.

[Zimmer 1995b] Zimmer, U. R. 1995b. Self localisation in dynamic environments. In *IEE/SOFT International Workshop.*

[Zimmer 1996] Zimmer, U. R. 1996. Robust world modelling and navigation in a real world. *Neurocomputing, Special Issue 1996* 13(2–4).

[Zimmer 1997] Zimmer, U. R. 1997. General and specific models in complex robotics systems - a critique and a proposal. Technical Report 1063, GMD.

Appendix A

Publications from this Thesis

The published papers are available on-line at the following web pages:
`http://www.cs.man.ac.uk/~duckettt`
`http://aass.oru.se/People/tom.html`

A.1 Bibliographic Details

A.1.1 Journal Articles

1. Tom Duckett and Ulrich Nehmzow, Mobile Robot Self-Localisation and Measurement of Performance in Middle Scale Environments, *Robotics and Autonomous Systems*, 24(1–2):57–69, 1998.

2. Tom Duckett and Ulrich Nehmzow, Mobile Robot Self-Localisation Using Occupancy Histograms and a Mixture of Gaussian Location Hypotheses, *Robotics and Autonomous Systems*, to appear, 2000.

Two further submissions to international journals are currently under review.

A.1.2 Conference Papers

1. Tom Duckett and Ulrich Nehmzow, Performance Comparison of Landmark Recognition Systems for Navigating Mobile Robots, *Proceedings of the Seventeenth National Conference on Artificial Intelligence* (AAAI'2000), Austin, Texas, July 30–August 3 1999.

2. Tom Duckett and Stephen Marsland and Jonathan Shapiro, Learning Globally Consistent Maps by Relaxation. *Proceedings of the IEEE International Conference on Robotics and Automation* (ICRA'2000), San Francisco, CA, April 24–28 2000.

3. Tom Duckett and Ulrich Nehmzow, Exploration of Unknown Environments using a Compass, Topological Map and Neural Network, *Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, November 8–9 1999.

4. Tom Duckett and Ulrich Nehmzow, Knowing Your Place in Real World Environments, *Proceedings of EUROBOT '99, 3rd European Workshop on Advanced Mobile Robots,* IEEE Computer Press, 135–142, 1999.

5. Tom Duckett and Ulrich Nehmzow, Self-Localisation and Autonomous Navigation by a Mobile Robot, *Proceedings of TIMR'99, Towards Intelligent Mobile Robots*, Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-99-3-1, Bristol, March 26 1999.

6. Tom Duckett and Ulrich Nehmzow, Knowing Your Place in the Real World. Accepted for presentation at the ECAL-97 Fourth European Conference on Artificial Life, Brighton, Sussex, July 1997.

7. Tom Duckett and Ulrich Nehmzow, Quantitative Analysis of Mobile Robot Localisation Systems, *Proceedings of TIMR'97, Towards Intelligent Mobile Robots*, Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-97-9-1, Manchester, September 5 1997.

8. Tom Duckett and Ulrich Nehmzow, Experiments in Evidence Based Localisation for a Mobile Robot, *Proceedings of the AISB'97 Workshop on Spatial Reasoning in Animals and Robots*, Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-97-4-1, Manchester, April 7–8 1997.

A.1.3 Technical Reports

1. Tom Duckett and Ulrich Nehmzow, *A Robust Perception-based Localisation Method for a Mobile Robot*, Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-96-11-1, 1996.

2. Ulrich Nehmzow and David Gelder and Tom Duckett, *Automatic Selection of Landmarks for Mobile Robot Navigation*, Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-00-7-1, 2000.

A.2 Abstracts

Tom Duckett and Ulrich Nehmzow, Mobile Robot Self-Localisation and Measurement of Performance in Middle Scale Environments, *Robotics and Autonomous Systems*, 24(1–2):57–69, 1998.

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.

Tom Duckett and Ulrich Nehmzow, Mobile Robot Self-Localisation Using Occupancy Histograms and a Mixture of Gaussian Location Hypotheses, *Robotics and Autonomous Systems*, to appear, 2000.

> *The topic of mobile robot self-localisation is often divided into the subproblems of global localisation and position tracking. Both are now well understood individually, but few mobile robots can deal simultaneously with the two problems in large, complex environments. In this paper, we present a unified approach to global localisation and position tracking which is based on a topological map augmented with metric information. This method combines a new scan matching technique, using histograms extracted from local occupancy grids, with an efficient algorithm for tracking multiple location hypotheses over time. The method was validated with experiments in a series of real world environments, including its integration into a complete navigating robot. The results show that the robot can localise itself reliably in large, indoor environments using minimal computational resources.*

Tom Duckett and Ulrich Nehmzow, Performance Comparison of Landmark Recognition Systems for Navigating Mobile Robots, *Proceedings of the Seventeenth National Conference on Artificial Intelligence* (AAAI'2000), Austin, Texas, July 30–August 3 1999.

> *Self-localisation is an essential competence for mobile robot navigation. Due to the fundamental unreliability of dead reckoning, a robot must depend on its perception of external environmental features or landmarks to localise itself. A key question is how to evaluate landmark recognition systems for mobile robots. This paper answers this question by means of quantitative performance measures. An empirical study is presented in which a number of algorithms are compared in four environments. The results of this analysis are then applied to the development of a novel landmark recognition system for a Nomad 200 robot. Subsequent experiments demonstrate that the new system obtains a similar level of performance to the best alternative method, but at a much lower computational cost.*

Tom Duckett and Stephen Marsland and Jonathan Shapiro, Learning Globally Consistent Maps by Relaxation. *Proceedings of the IEEE International Conference on Robotics and Automation* (ICRA'2000), San Francisco, CA, April 24–28 2000.

> *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 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.*

Tom Duckett and Ulrich Nehmzow, Exploration of Unknown Environments using a Compass, Topological Map and Neural Network, *Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, November 8–9 1999.

> *This paper addresses the problem of autonomous exploration and mapping of unknown environments by a mobile robot. A map-based exploration system is presented, in which a topological map of the environment is acquired incrementally by the robot, using an artificial neural network to detect new areas of unexplored territory. Using this approach, no manual intervention in the map acquisition process is required, and all computation is carried out in real-time on board the robot. Experiments are presented in which a Nomad 200 robot successfully mapped and navigated complex, real world environments containing transient changes such as moving people.*

Tom Duckett and Ulrich Nehmzow, Knowing Your Place in Real World Environments, *Proceedings of EUROBOT '99, 3rd European Workshop on Advanced Mobile Robots,* IEEE Computer Press, 135–142, 1999.

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

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

Tom Duckett and Ulrich Nehmzow, Self-Localisation and Autonomous Navigation by a Mobile Robot, *Proceedings of TIMR'99, Towards Intelligent Mobile Robots*, Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-99-3-1, Bristol, March 26 1999.

This paper provides an overview of a three year project conducted on the subject of self-localisation and autonomous navigation by a mobile robot. The research was carried out in two stages:

Self-Localisation. *Due to the fundamental unreliability of dead reckoning, landmark-based methods were investigated. Several related issues were addressed, including performance evaluation, replication and comparison of existing work, and the development of a novel localisation system based on the results obtained.*

Autonomous Navigation. *This work included the development of novel mechanisms for exploration and map building, as well as validating the previous work on self-localisation, in a complete, navigating mobile robot. Quantitative assessment of the different competences required for navigation was also carried out.*

Tom Duckett and Ulrich Nehmzow, Knowing Your Place in the Real World, *ECAL-97 Fourth European Conference on Artificial Life*, presented paper, 1997.

> *This paper addresses the scalability of existing techniques for mobile robot navigation to work in large, unstructured environments. In particular, we are interested in the problem of relocalisation after the robot has been placed at some arbitrary location (unknown to the robot) in a previously explored environment. In the first instance, we assess the problems found when a system previously shown to work in a small-scale environment was transferred to a middle-scale environment.*
>
> *Through analysis of the results, we are able to show that successful performance in the middle-scale environment requires a global compass sense. We provide a suitable quantitative measure for assessing localisation quality in the larger environment, and are able to quantify* post hoc *the performance which would be obtained given an appropriate compass mechanism. In addition, we use the same quality metric to evaluate various different approaches for pre-processing the robot's sensory information.*

Tom Duckett and Ulrich Nehmzow, Quantitative Analysis of Mobile Robot Localisation Systems, *Proceedings of TIMR'97, Towards Intelligent Mobile Robots*, Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-97-9-1, Manchester, September 5 1997.

> *This paper addresses the question of how to measure the performance of mobile robot localisation systems. A general performance metric and a standard experimental procedure are introduced, where no semantic interpretation of the robot's internal world model is required. A novel mechanism for tracking the "true" location of the robot is also presented.*
>
> *Together, these methods facilitate the comparison of disparate localisation systems on the same robot in the same environment, and also the replication of such experiments by other researchers. To demonstrate the utility of the approach taken, three different localisation systems are compared using the same data collected by a real robot travelling over a route of* 175*m.*

Tom Duckett and Ulrich Nehmzow, Experiments in Evidence Based Localisation for a Mobile Robot, *Proceedings of the AISB'97 Workshop on Spatial Reasoning in Animals and Robots*, Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-97-4-1, Manchester, April 7–8 1997.

> *This paper addresses the problem of localisation in autonomous mobile robot navigation, i.e., the task of identifying places after prior exploration and mapbuilding by the robot. In particular, the work is concerned with the more general problem of relocalisation without using past experience (i.e., knowing roughly where you are to start with), referred to here as 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. The robot therefore had no* a priori *knowledge of its position, and had to use current sensory perceptions and map knowledge alone to relocalise.*
>
> *A perception-based localisation method is presented which is resilient to the problem of perceptual aliasing (i.e., perceptual identity of distinct locations), and is capable of relocalising even in environments where no single place has a unique perceptual signature. During an exploration phase, the robot builds a map of its environment, using a self-organising neural network to cluster its perceptual space. The robot is then moved to an arbitrary location, where it will attempt to relocalise. By actively exploring, and accumulating evidence through the use of relative odometry between competing place memories, the robot is able to establish its location with respect to perceptual landmarks very quickly.*

Tom Duckett and Ulrich Nehmzow, *A Robust Perception-based Localisation Method for a Mobile Robot*, Department of Computer Science, Manchester University, Technical Report Series, ISSN 1361-6161, Report UMCS-96-11-1, 1996.

> *This paper addresses the problem of localisation in autonomous mobile robots, which is the task of identifying places that have been visited before by the robot. In particular, this paper is concerned with the more general problem of relocalisation, where the robot is unable to*

localise using past experience (in other words, the robot is "lost"). A perception based localisation algorithm is presented, which operates independently of any global reference frame. The method is resilient to the problem of perceptual aliasing (i.e., perceptual identity of distinct locations), and is capable of localising even in environments where no single place has a unique perceptual signature.

During an exploration phase, the robot builds a map of its environment, using a self–organising neural network to cluster its perceptual space. The robot is then moved to a randomly chosen position, where it will attempt to localise. By actively exploring, and accumulating evidence through the use of relative odometry between local landmarks, the robot is able to determine its location with respect to perceptual landmarks very quickly.

Index