Cognitive Inspired Mapping by an Autonomous Mobile Robot

Chee Kit Wong

A thesis submitted to
Auckland University of Technology
in fulfillment of the requirements for the degree of
Doctor of Philosophy

Primary Supervisor:
Professor Wai-Kiang (Albert) Yeap

School of Computing and Mathematical Sciences

29th May 2008

Table of Contents

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Wai-Kiang (Albert) Yeap. I am greatly indebted for his guidance and for sharing his vast knowledge, experience and ideas. I really appreciate the fact that his door is always open for a lively and enthusiastic discussion. Thank you for your encouragement, honesty, patience and the support you have given me. It has been a privilege to work with Professor Yeap.

I would also like to thank all the friends at the Centre for Artificial Intelligence Research. In particular, I would like to thank Dr. Jochen Schmidt for his contributions and discussions which have been significant towards the outcome of this thesis. To the rest of the group, past and present, thank you for all your support. I would also like to extend my gratitude to Professor Sapiyan who visited our group from the University of Malaya. His discussions have been very helpful to this study.

I wish to thank late Dr. Margaret Jefferies for the support she has given me. Unfortunately Margaret passed away whilst I was completing the thesis. May you rest in peace.

A special thank you to Malcolm Stenersen for his encouragement, and enthusiasm on my work. Thank you for volunteering to proofread the thesis on such a short notice.

I cannot finish without saying how grateful I am to my family. To my wife, Michelle, thank you for your unwavering love, support, patience and understanding. You kept me going, through the highs and lows. Thank you for believing in me, entertaining me, and for proofreading this thesis. To David, Mutsumi, Minaho, Momiji, my parents-in-law, uncles, aunts and cousins; whom have all given me a loving environment and encouragement.

Finally, and most importantly, I would like to thank my parents. Obviously, I am not where I am without my Mom and Dad. I would like to thank them for their constant and never-ending love, dedication and support. I owe my parents much for what I have become. To them I dedicate this thesis.

Abstract

When animals explore a new environment, they do not acquire a precise map of the places visited. In fact, research has shown that learning is a recurring process. Over time, new information helps the animal to update their perception of the locations it has visited. Yet, they are still able to use the fuzzy and often incomplete representation to find their way home. This process has been termed the cognitive mapping process.

The work presented in this thesis uses a mobile robot equipped with sonar sensors to investigate the nature of such a process. Specifically, what is the information that is fundamental and prevalent in spatial navigation?

Initially, the robot is instructed to compute a "cognitive map" of its environment. Since a robot is not a cognitive agent, it cannot, by definition, compute a cognitive map. Hence the robot is used as a test bed for understanding the cognitive mapping process. Yeap's (1988) theory of cognitive mapping forms the foundation for computing the robot's representation of the places it has visited. He argued that a network of local spaces is computed early in the cognitive mapping process.

Yeap coined these local spaces as Absolute Space Representations (ASRs). However, ASR is not just a process of partitioning the environment into smaller local regions. The ASRs describe the bounded space that one is in, how one could leave that space (exits) and how the exits serves to link the ASRs to form a network that serves as the cognitive map (see Jefferies (1999)). Like the animal's cognitive map, ASRs are not precise geometrical maps of the environment but rather, provide a rough shape or feel of the space the robot is currently in.

Once the robot computes its "cognitive map", it is then, like foraging and hoarding animals, instructed to find its way home. To do so, the robot uses two crucial pieces of information: distance between exits of ASRs and relative orientation of adjacent ASRs. A simple animal-like strategy was implemented for the robot to locate home. Results from the experiments demonstrated the robot's ability to determine its location

within the visited environment along its journey. This task was performed without the use of an accurate map.

From these results and reviews of various findings related to cognitive mapping for various animals, we deduce that:

Different animals have different sensing capabilities. They live in different environments and therefore face unique challenges. Consequently, they evolve to have different navigational strategies. However, we believe two crucial pieces of information are inherent in all animals and form the fundamentals of navigation: distance and orientation. Higher level animals may encode and may even prefer richer information to enhance the animal's cognitive map. Nonetheless, distance and orientation will always be computed as a core process of cognitive mapping.

We believe this insight will help future research to better understand the complex nature of cognitive mapping.

Chapter One

Introduction

This thesis describes work done using a mobile robot equipped with sonar sensors to compute a "cognitive map" of its environment, and using it, to find its way home. Since a robot is not a cognitive agent, it cannot, by definition, compute a cognitive map. Thus, this work is about using a robot as a test bed for understanding cognitive mapping.

Cognitive mapping, on the one hand, is a complex process which derives its input from an array of sensors and, for humans, is also a knowledge-rich process. Humans' interpretations of their environment are laden with their emotions, memories of past events, social etiquette, and others. All these factors affect the humans' perception of their physical environment. Robots, on the other hand, are sensor-poor and knowledge poor and they never forget. It is therefore not a straightforward task implementing theories of cognitive mapping such as that of Kuipers (1977, 2000); Yeap & Jefferies (1999); Chown, Kaplan & Kortenkamp (1995); and others, on a mobile robot.

Consequently, little cross-fertilization between robot mapping and cognitive mapping has occurred, although there are increasing attempts to do so recently, see Jefferies & Yeap (2008). Understandably, researchers interested in robot mapping focus on developing algorithms that will produce a map useful solely for robot navigation. One representation commonly produced is that of an accurate global metric map, for example, Konolige (2004); Simmons & Koenig (1995); Tardos, Neira, Newman & Leonard (2002); Thrun (2008) and Tomatis, Nourbakhsh & Siegwart (2002). Robotics researchers often refer to their problem as Simultaneous Localization and Mapping (SLAM) – see Thrun (2008) for a comprehensive review of SLAM. In short, the robot must know exactly where it is in the map computed so far.

In contrast, humans are often lost in a new environment and animals do not compute an accurate global metric map. They are easily distracted by what is happening in their surroundings and have to be always on the look-out for danger. In addition, views obtained during an outward journey from home are not the same as the views obtained during the homeward journey. Short-cuts are taken, landmarks are remembered and conceptual views are formed. With their different senses, travel moods and needs, different species, as varied as ants, birds, fish, rats and humans, compute different kinds of fuzzy representations and develop many ingenious algorithms utilizing its fuzzy representation.

My goal in this research is to develop a cognitive map-like algorithm for a robot and use it to further investigate the nature of cognitive mapping. In particular, three questions were addressed in this research:

1. Could the robot be used to compute some kind of a cognitive map?
2. Could the map computed be used to solve a cognitive mapping related task?
3. From the above, could further insights be gained about what is happening in the early cognitive mapping process?

To many robotics researchers, their robots cannot find its way in its environment without having solved the SLAM problem. Thus, the first question is a challenge to get the robot to compute something different, moving this research away from the mainstream research of robotics. Ideally, the robot should behave like an animal, wandering freely in its own environment and compute its own "cognitive map". The map computed should not be mathematically precise for the robot to localize its exact position or thereabout on the map. The map computed does not have the exact shape of each local environment it has visited. In particular, the shape of each local environment could change depending on the direction of travel. Creating such a robot would be like having created a new kind of "animal", with its own peculiar kind of sensors.

Providing an answer to the first question would not be interesting if the robot computes such a map without doing something useful with it. Thus, the second question is to get the robot to use the cognitive map to solve a fundamental cognitive mapping related task. The foraging behavior of animals would suggest that the most basic use of one's cognitive map is to enable one to return home. Could this robot find its way home? The robot used in the experiments conducted in this thesis is equipped with sonar and odometer sensors. It is a deliberate choice not to equip the robot with more powerful sensors such as lasers. Furthermore, no error-correction software is used together with the sensors. This helps create a situation atypical to that faced by robotics researchers. For instance, one could not re-create the same map on the homeward journey so that one could recognize where one is based upon matching. The errors in sonar sensing and from the odometer readings would make it very difficult to do so. This situation forces the development of other strategies for the robot to find its way home and in particular strategies utilizing similar information afforded to animals with their own imprecise maps. Animals were found to make much use of distance and orientation information. Thus, a new algorithm is developed that uses these two pieces of information implicitly available in the robot's cognitive map.

The third, and final question, is a key question related to cognitive mapping and which this research will attempt to address. In Yeap's (1988) theory of cognitive mapping, it was argued that what is computed early in the process are a network of ASRs (Absolute Space Representations). An ASR should not be mistaken as any partitioning of a large environment into many smaller local environments or regions. In particular, an ASR is not the same as the partitioning of the environment into different empty spaces popularized by the work of robotics researchers. An ASR is a representation that affords two crucial pieces of information: the bounded space that one is in and where one could move out of that space (the exits, see Yeap & Jefferies (1999)). One continues to shape and re-shape an ASR as one move through an environment.

However, since Yeap's theory has been proposed, it has rarely been demonstrated how a network of ASRs would eventually emerge in one's exploration of the environment and be useful for solving a cognitive mapping task. From the vast literature on cognitive mapping (see Downs & Stea (1973); Golledge (1999); Kitchin & Freundschuh (2000) and Portugali (1996)) and from an introspection of one's own behavior, one would expect that one remembers some loosely coupled ASRs and landmarks initially. Subsequent explorations of the environment would enable one to build a more well-connected network. However, believing so does not explain how a loosely coupled network of ASRs could later be turned into a well-connected representation. Does this require that the subject be lost several times? If so, how could one recover from being lost and how could subsequent learning improve upon it? What happens to the network created during the confusion? How useful is the initial loosely coupled network of computed ASRs? What could one do from what is learned during the initial exploration?

This research does not attempt to answer all of the above questions related to the initial cognitive mapping process. It will attempt to address the last two. Experimenting with a robot to find its way home using a fuzzy map will help to shed some light into the nature of the early cognitive mapping process. The robot provides a situation where the minimum amount of information is available and implementing an animal-like strategy will allow us to study the problem closely. This exploratory study will show how a robot might be useful for finding answers to the above kinds of questions and more could be discovered.

All the experiments conducted could be described briefly as follows: The robot wanders on its own through an office environment similar to that as shown in Figure 1.1(a). It collects sonar data (Figure 1.1(b)) and odometer readings of the journey and computes ASRs as shown in Figure 1.1(c). *Home* is defined as the starting point of a journey and after exploring for a while, the robot is instructed to turn around and find its way home. On its homeward journey, the robot computes another cognitive map, representing how the environment is perceived on the homeward journey. This map is most likely to be different from the one computed during exploration. Several experiments were conducted and, on most occasions, the robot successfully returned home.

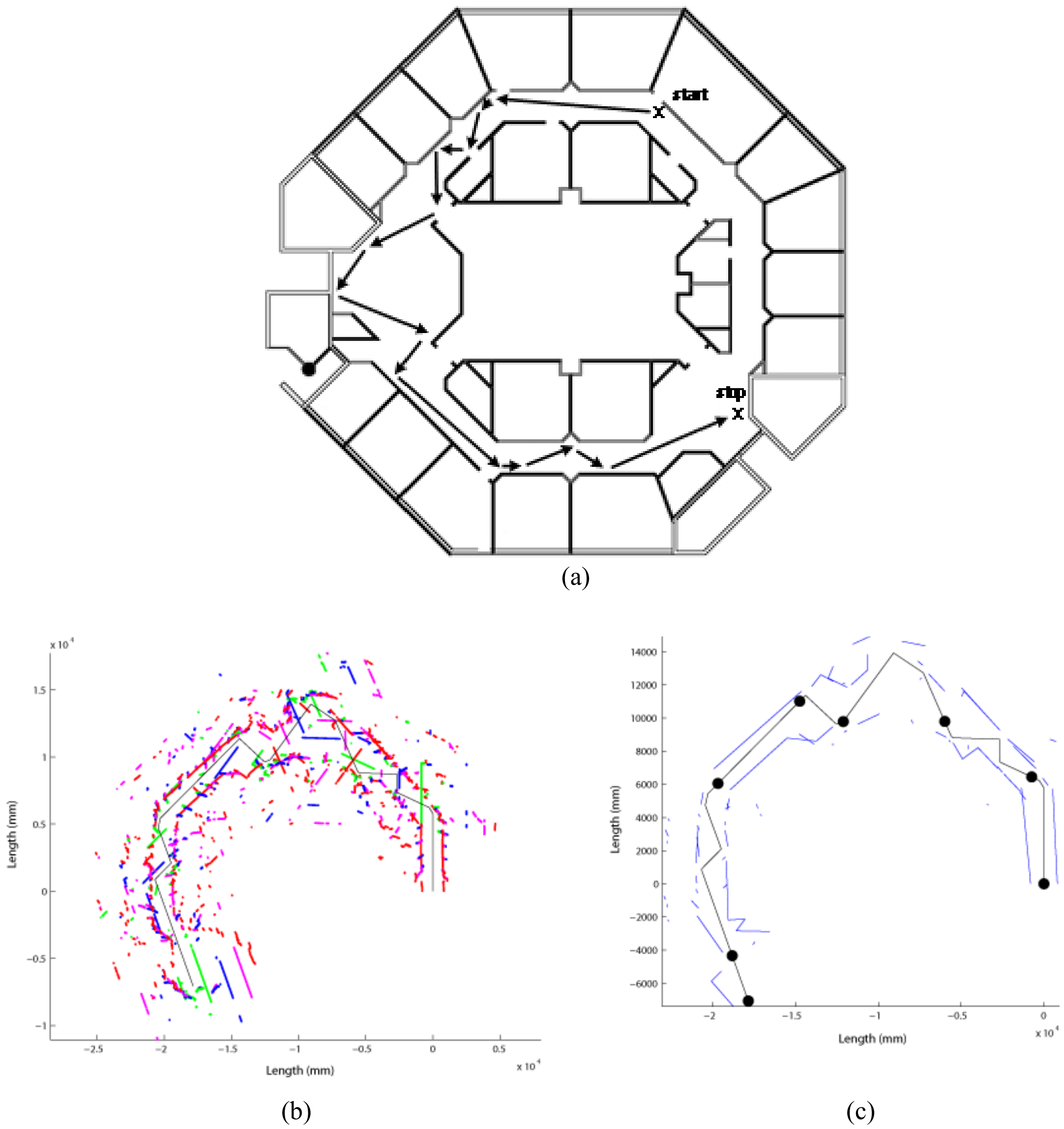


Figure 1.1 (a) Example of an exploration journey through an office environment. The arrows depict the physical movement of the robot during its journey (b) Sonar information collected during the exploration journey. The different colors represent the data collected from the different sonar sensors and (c) the ASRs computed for the environment visited. The ASRs are the spaces between the solid black dots, or split points. The blue lines represent the boundary of the ASRs.

Two new algorithms were developed. The first new algorithm is for computing ASRs and is based upon the well-known Split and Merge method (see Duda & Hart (1973); Niemann (1990) and Pavlidis & Horowitz (1974)), which originated from research in pattern recognition. The basic method for computing ASRs as proposed in Jefferies (1999) has not changed. However, because the robot has very poor sensors, a new algorithm has to be developed. The second new algorithm consists of two parts: the need for a navigation strategy to enable the robot to find its way home and the need for the robot to localize itself while returning home. Four new sub-algorithms are developed for these two parts.

The strategy developed for the robot to find its way home is quite straightforward. Since it is assumed that the robot remembers all the ASRs computed, the strategy for going home is to go back to the entrance used for entering the current ASR and then back to the entrance used to enter the previous ASR and so on until one reaches home. However, the lack of an ability to identify exits poses an interesting localization problem. A general solution to the localization problem is developed, which is based upon combining confidence measure of different kinds of information. Two different kinds of information are used, namely distance traveled in terms of the length of the ASR and the orientation between adjacent ASRs. Fusion of the strategies is done using the Democratic Integration technique which was originally proposed by Triesch (1999) and Triesch & von der Malsburg (2001) for sensor data fusion in computer vision, using images as input data.

Chapter 2 provides a review of different theories of cognitive mapping, and the different attempts to implement them on a mobile robot. In particular, the chapter discusses the theories of Yeap, Kuipers and Chown et al., and three different neural models of cognitive mapping. Chapter 3 provides a review of some recent neurological and behavioral studies about cognitive maps. Recent experiments to help advance our understanding of cognitive mapping are discussed. Chapter 4 describes the implementation of a new algorithm for computing ASRs for a mobile robot equipped with sonar sensors. The basic idea for computing ASRs is still the same as proposed by Jefferies' (1999) but the details are significantly different. This is because the robot is equipped with sonar sensors which do not allow much information to be sensed from a single location in the environment. This chapter also provides details about the robot and its sensors. Chapter 5 describes how the robot finds its way home using the cognitive map computed. Four new algorithms were developed. The first is concerned with the strategy for the robot to navigate home. The remaining three algorithms are concerned with how the robot localized itself in the environment. The localization algorithms make use of the distance of the path between exits of ASRs visited and the relative orientation between adjacent ASRs. Finally, Chapter 6 concludes the thesis with a discussion of its findings and future directions.

Chapter Two

Theories and Implementations of Cognitive Mapping Models

This chapter discusses the various theories and models of cognitive mapping. In particular, those closely related to the one chosen as the basis for this work, namely Yeap's computational theory of cognitive mapping.

Sections 2.1, 2.2, and 2.3 describe the ASR model, SSH model and PLAN model of cognitive mapping respectively. The differences between them and their recent progress will be highlighted. In particular, and of most relevance to the current work, the focus is also on how these theories have been implemented on a mobile robot. In addition to these theories, Section 2.4 describes three different implementations of neural-inspired models of cognitive mapping. These works investigate cognitive mapping from a different angle, i.e. at a neural level, and it is thus of complimentary value to the work presented here. Again, the major interest is on the theory proposed and the way the theory has been tested on a mobile robot. Section 2.5 concludes this chapter with an overall discussion of the various models and their implementations. For a thorough review of earlier works on cognitive mapping, see Yeap & Jefferies (2000).

2.1 ASR Model

Yeap's (1988) computational theory of cognitive mapping is developed from paying attention to how sensor information is used to compute a cognitive map. The key question asked in Yeap's approach is: What needs to be made explicit in the cognitive mapping process, why and how? The input to the process is some form of a viewer-centered representation of surfaces obtained by whatever sensors are available to the viewer. The theory is thus a general one: it accounts for *why* and *how* information captured at the sensor level is transformed, via various stages, into a cognitive map. To date, four key representations were proposed, namely:

1. An Absolute Space Representation (ASR) – An ASR is a representation of each local environment visited. A local environment is identified as any bounded region of space which one has wandered into. The feeling of boundedness is perceptually driven i.e. it depends solely on how subjects perceive the environment. Thus two key pieces of information that define an ASR computed immediately are its boundary/shape and its exits, see Yeap & Jefferies (1999). By computing an ASR, one makes explicit the space that one is in and the exits for moving out of it.

2. A Memory for One's Immediate Surroundings (MFIS) – An MFIS is a global map of one's immediate surroundings. It consists of a network of ASRs described using a global co-ordinate system and centered on the ASR which contains the subject. The MFIS is thus an extension of what an ASR is and provides the subject an awareness of their immediate surroundings. The MFIS is thus useful for knowing what is immediately beyond the current ASR.

3. A Raw Cognitive Map – This consists of networks of ASRs, each describing how local environments are connected together. It is "raw" in the sense that information in it is perceptually driven. The network, if well-connected, enables one to navigate with ease in one's environment and provides the basis for further development into a much richer cognitive map.

4. A Full Cognitive Map – This map contains all the richness that is found from interpreting the world through one's experience and understanding. A minimal representation would be a hierarchy of place representations. Each place representation points to or names one or more groups of ASRs. The way one interacts with one's world could change the shape of ASRs and/or how best they are remembered. Things in them are remembered for various reasons and at different levels of interpretations.

Figure 2.1 shows how the different representations are inter-related. Implementing the notion of an ASR, using a mobile robot is not a straightforward task. For instance, it is not just about computing a representation of the empty spaces traversed. The latter is a popular approach among roboticists who are interested in computing a representation of the robot's environment for the sole purpose of navigation in the environment. An ASR is a complex entity. It begins with a rough description of the shape of each local environment and its exits. Both could change as a result of further explorations of, and social interactions within, the environment. An initial network of ASRs might not be useful for navigation as some ASRs in-between could be forgotten while others might not be detailed enough.

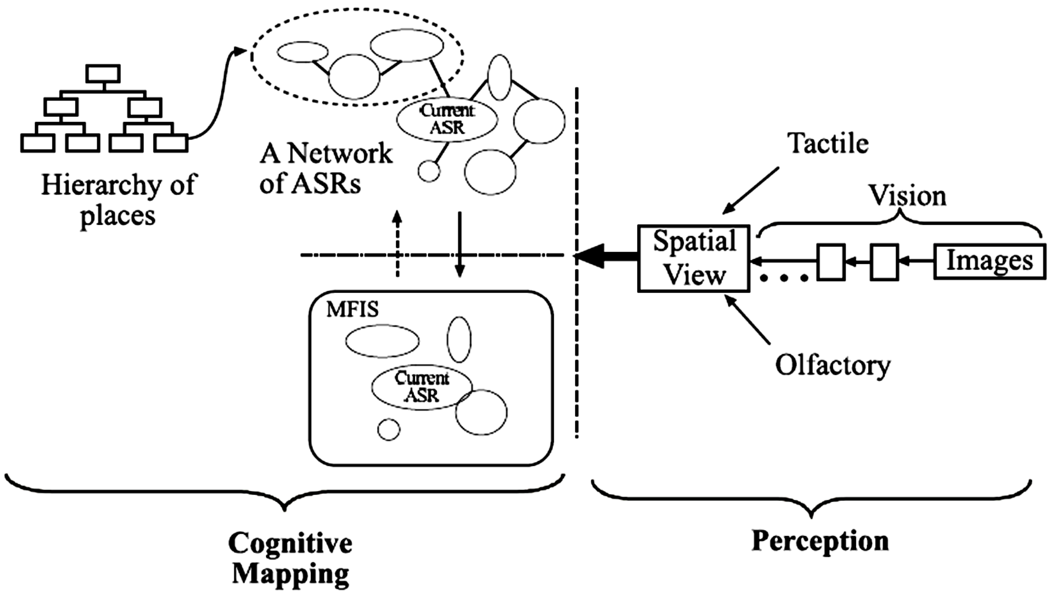
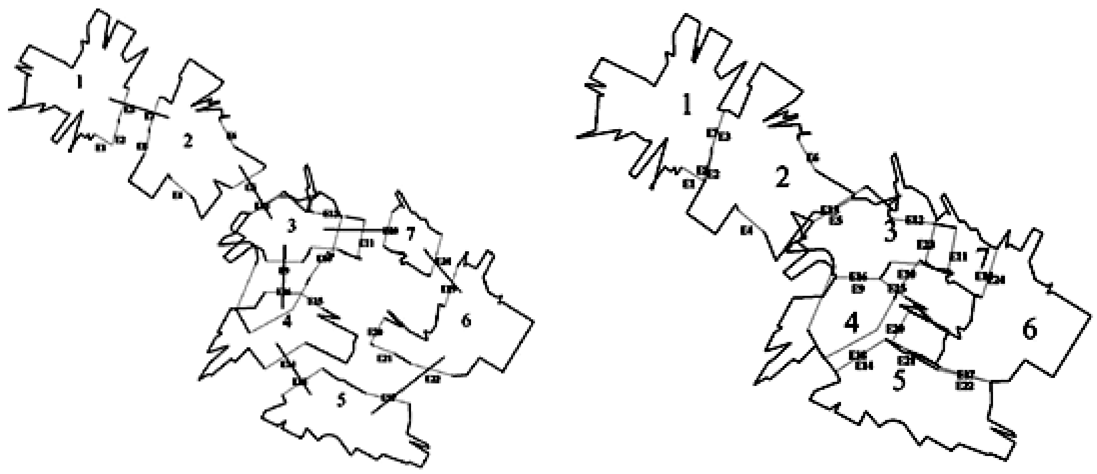


Figure 2.1 Yeap's computational theory of cognitive mapping. Image has been reproduced from Figure 1 in Yeap (2007). Note that although an improved version of the cognitive mapping process was discussed in Yeap (2007), it is premature to consider that model in this research.

Given the complex nature of ASRs, very few attempts have been made in implementing the idea on a mobile robot. Jefferies (1999) noted the importance of exits and used that as an important constraint for computing the boundary of an ASR. Subsequently, and using a mobile robot equipped with a laser sensor, she and her co-workers have implemented two ways to recognize ASRs revisited, namely: (i) the use of a neural network for learning signatures of ASRs, see Jefferies, Weng, Baker & Mayo (2004), and (ii) the use of an MFIS to help recognize adjacent ASRs, see

Jefferies, Weng & Baker (2008). Figure 2.2 shows an example of how an MFIS is used to help recognize ASRs revisited. This ability to recognize ASRs revisited allows them to build useful networks of ASRs for a large environment. The work presented in this thesis continues to investigate this early process of computing ASRs and using them to solve a cognitive mapping problem.



(a) (b)

Figure 2.2 (a) A network of ASRs computed so far, and (b) the MFIS. The robot has re-entered ASR3 from ASR7 but did not compute a new ASR for it. From the MFIS, it realizes that it is in ASR3. Images reproduced from Figure 12.4 of Jefferies et al. (2008).

2.2 SSH Model

Although Yeap (see section above) identified four different representations in a cognitive map, he and his co-workers have focused much on the difficult task of computing ASRs and to a limited extent, on the MFIS. In contrast, Kuipers developed one of the most comprehensive computational models of cognitive mapping to date. His Spatial Semantic Hierarchy (SSH) specifies 5 different levels of knowledge known to exist in a cognitive map. These levels are:

1. **Sensory level** – Interface to the agent's sensory system but at this level, there is neither information on the global structure of the environment nor information on the robot's position or orientation within the environment

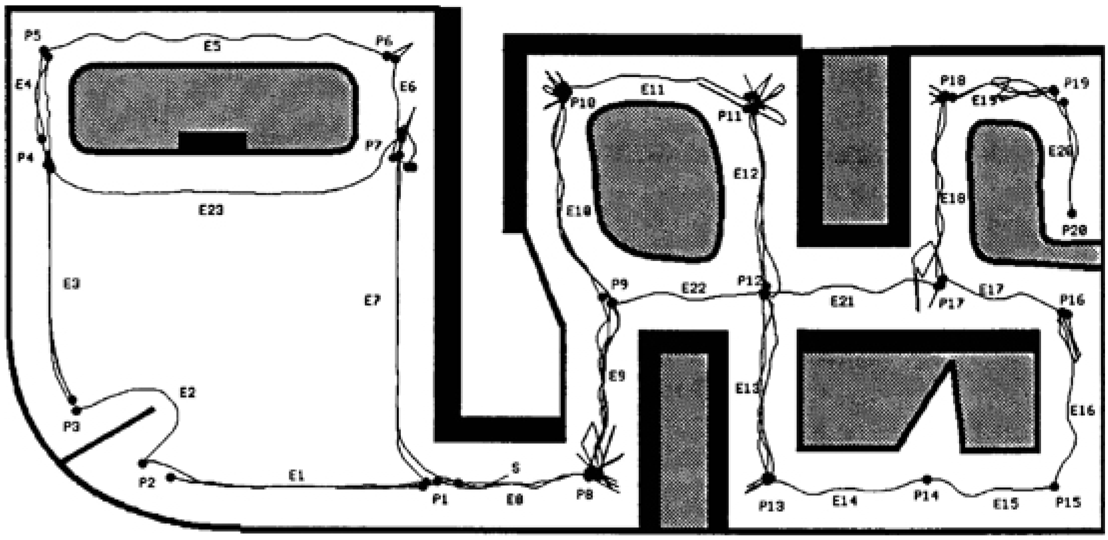
2. **Control level** – Exploration of an unknown environment is performed through the selection of a control law, which depends on the information collected by the sensor(s) of the environment. For example, if the robot is facing free space and there is a wall to its left, then maybe, the best control law is 'Follow the Left Wall'.

3. **Causal level** – The causal level abstracts the sequence of control laws the robot uses to move from one distinctive state to another within the environment. This description is augmented with views that are generated at the distinctive states before and after the agent's movement. Views are sensory information collected at the distinctive states.

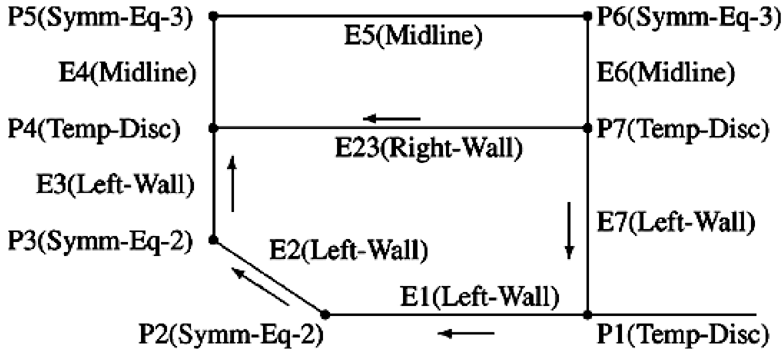
4. **Topological level** - At the topological level, the environment is described using places, paths and regions and their connectivity, order and containment.

5. **Metrical level -** Metrical level involves the generation of a 2-D global map (such as grid map) of the explored environment. This provides quantitative information to the agent's representation of the environment, such as distance between places and angles between paths.

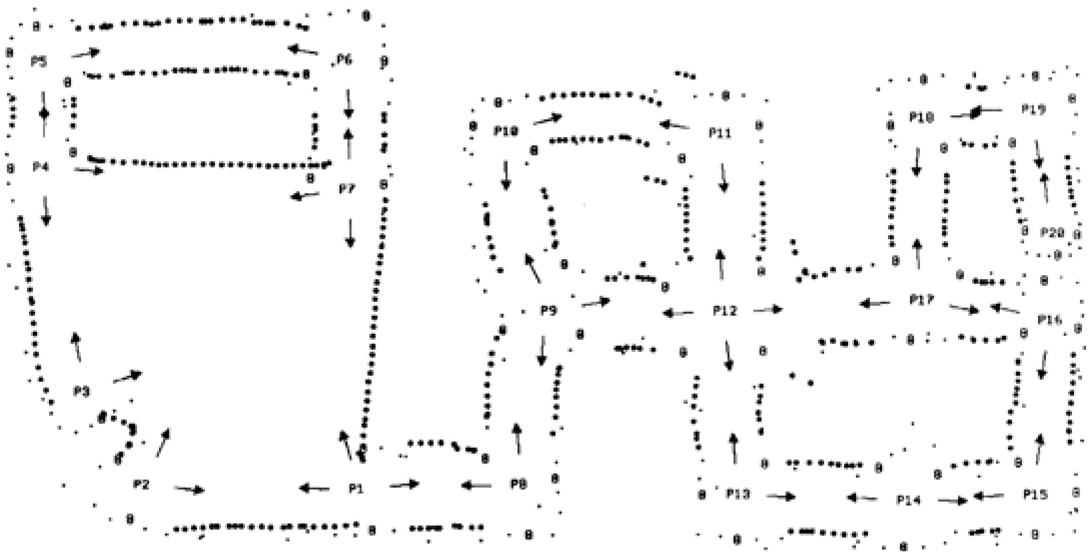
The model operates in a hierarchical nature. That is, each level provides the properties that the next one depends on. In particular, Kuipers stresses that the topological level is prior to the metric level. Kuipers (1977, 1978) initially developed a simulation model known as the *TOUR model* to investigate the topological nature of a cognitive map. The SSH is an extension of the TOUR model and with it, Kuipers and his co-workers have begun investigating other levels of cognitive mapping. For example, using a simulation model, Kuipers & Byun (1987, 1988, 1991) investigated how a robot could reach distinctive states in the environment using simple control laws and how a topological and a global metric map are then abstracted from those states (see Figure 2.3). Other attempts to implement the SSH model are found in Lee (1996) and Remolina (2001).



(a)



(b)

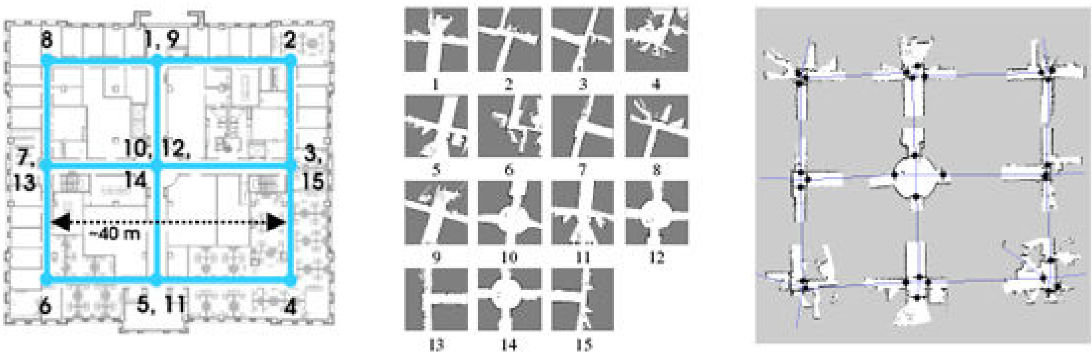


(c)

Figure 2.3 Exploration and mapping performed by the simulated NX robot. (a) The exploration journey and distinctive paths and places (b) Fragment of the topological map which identifies places and paths; and the measures that were used to define them (c) The global metrical map. Images were reproduced from Kuipers (2000).

These early SSH implementations have limited utilization of sensory information. More recently, Kuipers and his co-workers have begun investigating integrating perceptual information from the robot's sensors to the SSH model, see Beeson et al. (2003) and Kuipers, Modayil, Beeson, MacMahon & Savelli (2004). The hybrid SSH is an extension of the SSH which includes metrical description of the places described in the topological map of the SSH. These local descriptions are known as local perceptual model (LPM) and are meant to provide low level metric information that is useful for path planning during navigation.

The LPMs were implemented as occupancy grids, each having their own local frame of reference. They experimented on how to detect locations that are suitable to be places for collecting metrical information in Beeson, Jong & Kuipers (2005) and how the LPMs are linked to each other in Kuipers et al. (2004). The model was tested on a mobile robot in an environment with multiple nested loops as shown in Figure 2.4(a), see Kuipers et al. (2004). The exploration path is depicted by the numbered labels. Kuipers et al. (2004) also investigated on solving perceptual aliasing when revisiting an already encoded LPM (see figure 2.4(b)); and how gateways on the same path are linked as shown in figure 2.4(c) so that the hybrid topological map can later be used to generate a global metrical map of the environment, see Modayil, Beeson & Kuipers (2004).



(a) (b) (c)

Figure 2.4 (a) Robot exploring an environment with multiple nested loops. Numberings denote the order the environment was explored. (b) Local Perceptual Models that were generated at each place denoted by the numberings and (c) shows the unique LPMs that have been selected and connected via gateways that are on the same path. Reproduced from Kuipers (2008).

2.3 PLAN Model

PLAN stands for Prototypes, Location, and Associative Networks. Chown, Kaplan and Kortenkamp (1995) developed the PLAN model of cognitive maps from closely mirroring how humans solve their way-finding problem. They argued that the human cognitive mapping process is about solving the latter problem and the latter could be broken down into four component problems, namely:

1. landmark identification
2. path selection
3. direction selection
4. abstract environmental overviews

Henceforth, PLAN is derived from a solution to each of the above problems. Beginning with landmarks, Chown et al. argued that landmark perception is no different from the perception of other object categories and therefore the development of a general prototype theory for the cognitive mapping process would be adequate to address landmark identification. How prototypes are created and represented would naturally impact upon the development of one's cognitive map. PLAN, at present, suggests the use of a simple associative network and the idea was tested via a connectionist implementation known as NAPS, a Network Activity Passing System, see Levenick (1985; 1991). The need for direction selection in way-finding prompted Chown et al. to propose the use of a local map in addition to a network of landmarks. Local maps tell us how we could orient ourselves to nearby landmarks. Figure 2.5 shows an example of how visual cues are extracted to form a picture plane, which is the observation of a scene from a location from one viewpoint or direction. The view is divided into five viewing directions, each having a locational grid connected to it. Cues extracted from the scenes are mapped onto the cells in the locational grid. Together, the directional and locational grids form the basis of the local map.

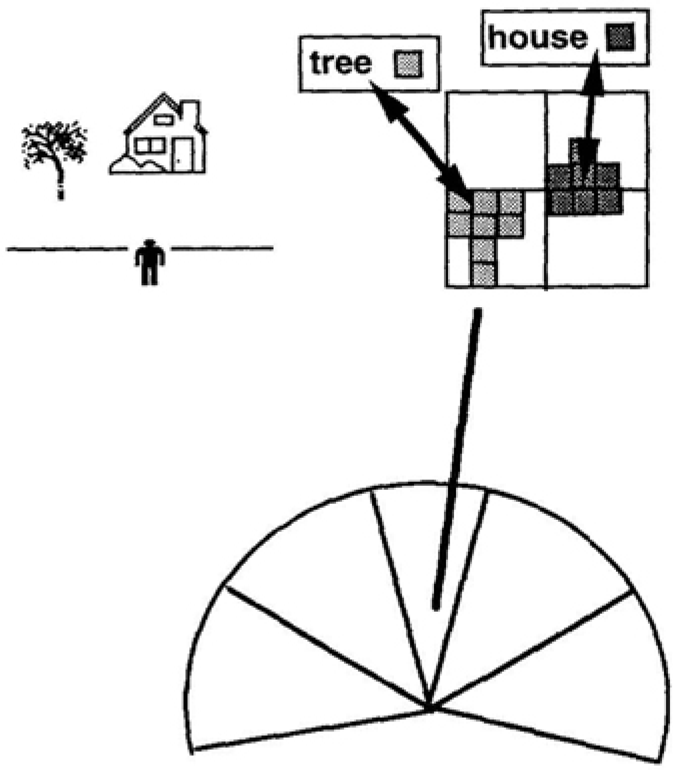


Figure 2.5 Formation of the local map. Top left: Observer's view consisting of a tree and a house. Top right: Extraction of visual cues onto a picture plane. Bottom: Picture plane is stored according to the observation direction. For a place, there are a total of five observation directions. The picture planes in all directions form the local map. Image reproduced from Chown et al. (1995).

It is interesting to note that they further argued that local maps are not constructed at landmarks but rather at choice points in a journey, such as at a fork in a road or a doorway. However, choice points are not represented in the network of landmarks and as such, how can we create local maps? It turns out that another network is proposed: an R-Net which is a network of local maps. Chown et al. (1995, p. 28) speculated that having two could be advantageous, namely: (i) it provides a redundant system of route structures, and (ii) that the two working in conjunction with each other can operate faster and more efficiently. They further speculate that R-Net will eventually supplant the associate network of landmarks as the primary structure for use in way-finding. Since each local map already contain information about landmarks and as Chown et al. (1995, p. 29) have pointed out that R-Net more naturally reflect the experience of a journey, it is a puzzle why an independent network of landmarks is still needed. It would be straightforward to create a single network of mixed landmarks and local maps.

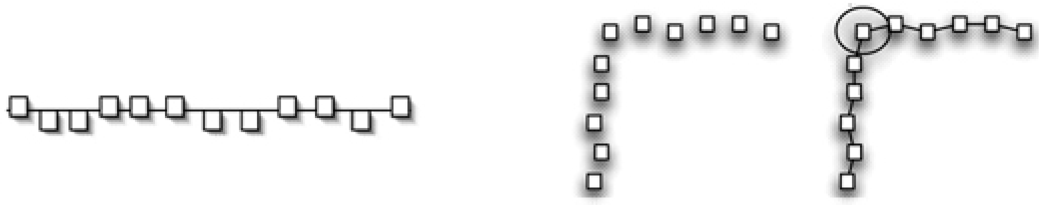
The third, and final, representation proposed in PLAN is a regional map. This representation is meant to capture large-scale spaces that cannot be perceived immediately, in contrast to local maps. Interestingly, Chown et al. proposed that such regions have gateways that mark the entrance to these spaces. Yeap and Jefferies (1999) also argued for the importance of entrance in spaces but these are at the level of ASRs (or local spaces) rather than at the level of regional spaces.

R-PLAN or Robot-PLAN describes an implementation of PLAN onto a mobile robot equipped with sonar sensors and camera, see Kortenkamp (1993) and Kortenkamp & Weymouth (1994). In R-PLAN, they investigated on how sonar sensor information is firstly used for identifying gateways and then how visual cues are extracted from vision. One major deviation from PLAN was the type of visual cues extracted. In PLAN, the cues are abstractions of objects from the scene, e.g. trees, buildings, etc. However, this is a task too complex to implement on a mobile robot. Instead, vertical edges are extracted as cues from the scenes in R-PLAN. In addition, Kortenkamp and Weymouth (1994) investigated how these scenes can be used to perform place recognition.

More recently, Chown and Boots (2008) described a new implementation dubbed C-PLAN (for Corner-PLAN). C-PLAN is conceptually inspired by PLAN but differs significantly in implementation. As the name implies, the approach is based on the corners detected in the environment and even though gateways can be identified, they are not represented as they are in R-PLAN.

In C-PLAN, Chown and Boots investigated how corners can be detected using information collected using a laser scanner. A topological map is used to represent the environment: each node representing a perceived corner and each path is used to form relationships between corners that were perceived simultaneously. Furthermore, the connection between nodes also represents the geometrical relationship between linked corners. Over time, this link is strengthened with repeated simultaneous viewing and weakened when only one corner is viewed at one time.

Chown and Boots investigated on classifying the two features commonly found in laser readings: walls and corners (see Figure 2.6). The authors described walls being features with readings that are 180 degrees to each other and corners are readings that form a 90-degree angle to each other (see Figure 2.6(b)). The justification is that humans are not suited for certain types of environment and can be easily deceived in their perception. Whilst this rationale may be true when learning a new environment, nonetheless, one can speculate that over time, humans can represent environments that consist of various types of corners and other features, such as the environment used in the experiments conducted in this thesis (see Figure 4.1 in Chapter 4).



(a) (b)

Figure 2.6 (a) Laser sensor information that closely resembles a straight line and (b) left: raw laser readings and right: identification of a corner due to a large angle between neighboring readings. Images reproduced from Chown & Boots (2008).

2.4 Neural-inspired Models

Neural-inspired models, as a class, differ from computational models in that, not that rich amount of information is encoded in the model itself. Thus, a neural-inspired model is much more concerned with supplying sufficient information extracted from different parts of the environment visited to activate and train “cells” in a neural network so that these cells could be re-activated when the same part of the environment is re-visited. Their reactivation is interpreted as a successful recognition of that part of the environment and consequently these cells are dubbed “place cells”. In contrast, computational models, as described in the previous section, is more concerned with the rich amount of information stored in a cognitive map and how all that information could be obtained, organized, and utilized. Its notion of a place is a far more complex entity.

Nonetheless, and at least for completeness, it is useful to review some recent neural-inspired network models and observe their progress to date. In particular, attention is paid to the kind of cognitive maps produced in such models and the way in which these models are tested on a mobile robot.

2.4.1 Hafner

In Hafner's (2000a, 2000b) model, her neural-inspired network consists of an input layer of "neurons" connected to an output layer of fully interconnected neurons, each with a weight vector. The output layer would be trained into a topological representation which functions as the cognitive map for the environment visited (see Figure 2.7). To do so, neurons in the map layer are activated by a function with terms concerning feature similarity, neuron connectedness, and angle information of the traveled path.

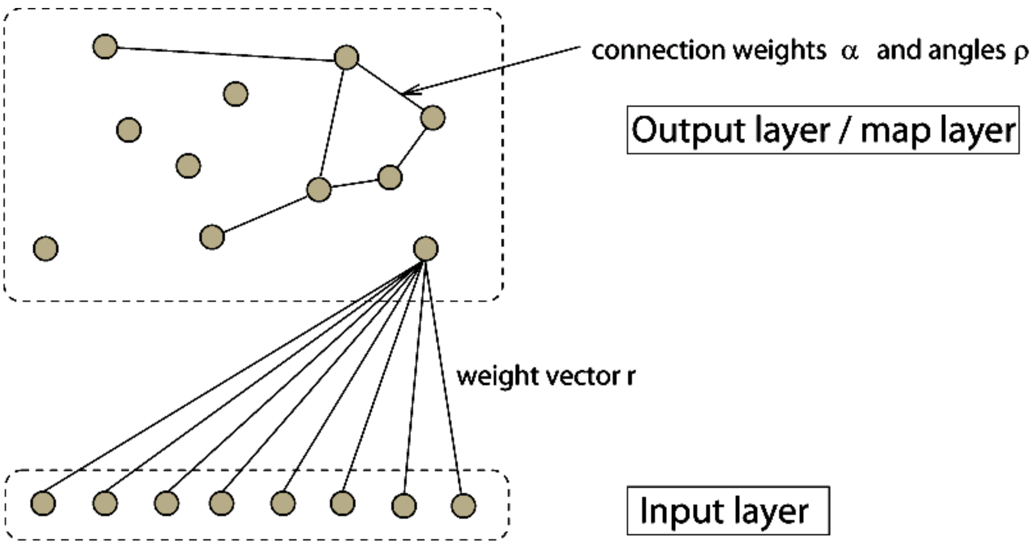
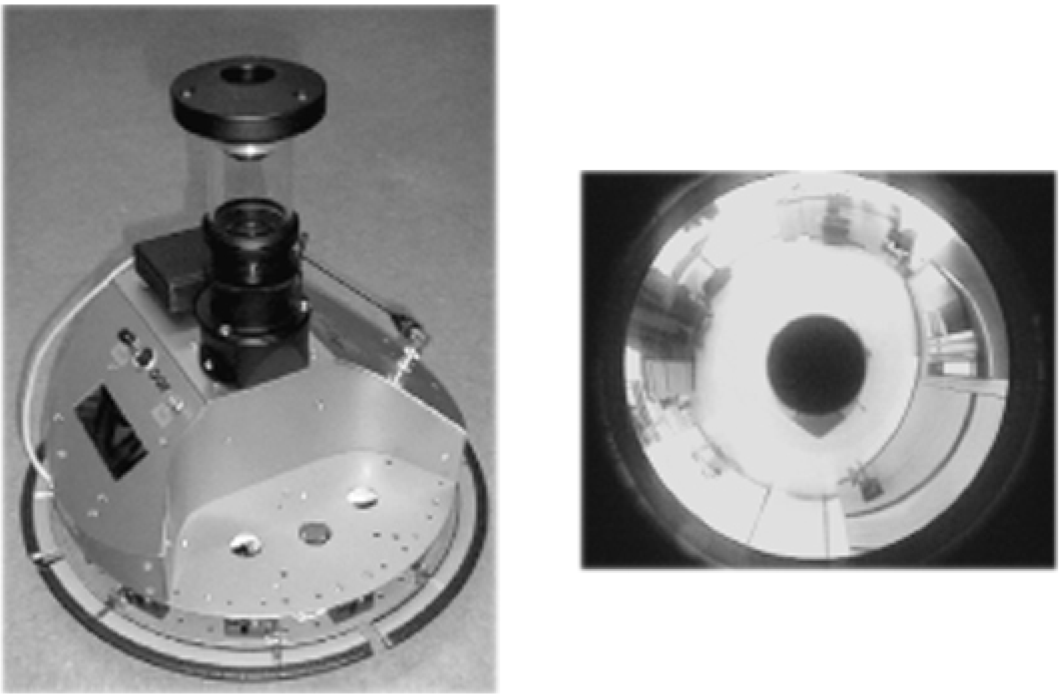


Figure 2.7 Hafner's neural network model. The output layer is a topological representation of the environment and serves as the cognitive map. The connections between the neurons in the output layer also contain connection weights and angle information. Image was reproduced from Hafner (2000b).

The model was initially tested and optimized using a simulator and then implemented on a mobile robot. In both simulation and robot implementations, the robot's headings during exploration are recorded. The simulated robot also collects information on the environment through distance readings which are used as input to the neural network system.

For the robot implementation, Hafner used an omni-directional camera for sensing the robot's surroundings. Figure 2.8(a) shows the Samurai robot used for the experiments. During exploration, the robot is allowed to wander randomly around the environment. At every few time steps, as omni-directional image of the environment is captured (see example in Figure 2.8(b)).



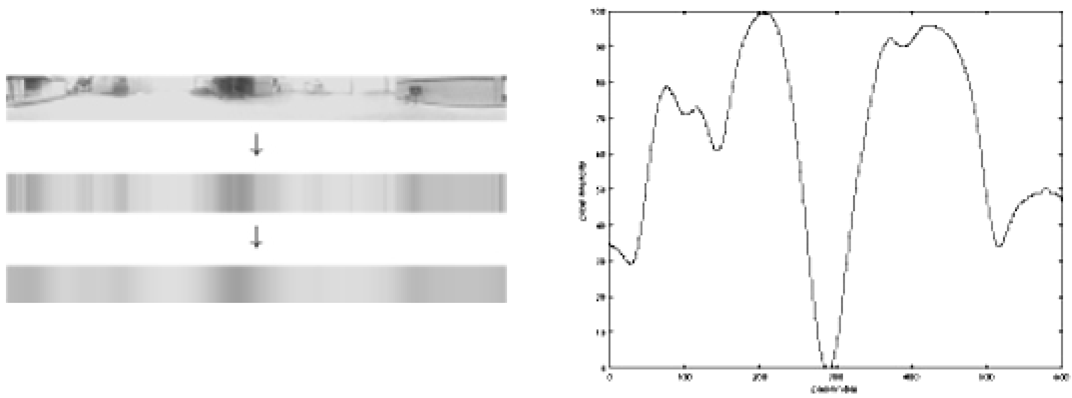
(a) (b)

Figure 2.8 (a) Picture of the Samurai robot which is used by Hafner for her experiments. It is equipped with an omni-directional camera and a magnetic compass and (b) the environment as seen through the camera. Images were reproduced from Hafner (2000b).

The images taken by the camera are then processed to produce a low resolution, orientation-invariant output and then transformed into polar coordinates. Vertical averaging is applied on the images and then smoothened using a Gaussian filter. To ensure rotation invariance, the images are rotated according to the compass reading.

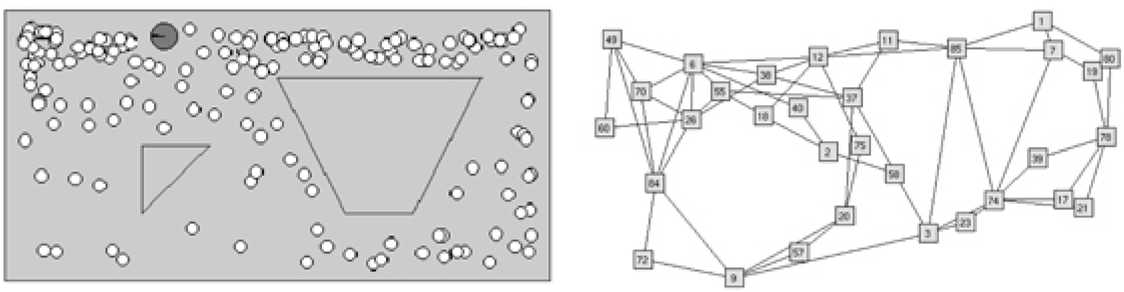
The results of these processing stages can be seen in Figure 2.9(a). Sixteen samples are then collected from the intensity curve (see Figure 2.9(b)), which are used as inputs to the neural network system.

The output of the neural network is a topological layout of the cognitive map, where the nodes represent the center of the place fields and the edges represent the connections between the places (see Figure 2.10). The thickness of the edges signifies the strength of the connection between the place fields.



(a) (b)

Figure 2.9 Extraction of information from the robot's vision (a) Image from Figure 2.8(b) projected onto a polar plane (top); vertically averaged (center); and smoothened using a Gaussian filter (bottom). (b) Intensity curve of the resulting image after processing as shown in the bottom image in (a). Images were reproduced from Hafner (2000b).



(a) (b)

Figure 2.10 An exploration example using the simulation robot. (a) The simulated environment with the white circles depicting the locations where sensor readings were collected. (b) The cognitive map generated from the exploration example in (a). The nodes represent the center of the place fields and the edges represent connections between places. The thickness of the edges is used to represent the strength of the connections. Images were reproduced from Hafner (2000b).

2.4.2 Cuperlier et al.

Cuperlier et al. proposed a model which is based strongly on a model of the hippocampal and prefrontal interactions, see Cuperlier, Quoy, & Gaussier (2007). Of particular interest here is that what is computed as a cognitive map is a network of transition points rather than places. Transition points record transition information (such as the directional information linking the starting and ending location) between places and their links indicate frequency of use between transition points. Figure 2.11 shows their model.

The model begins by using a neural network to learn and identify a constellation of landmarks from each panoramic image taken of the environment and directional information from a compass (see Figure 2.12). Note that the concept of landmark here refers to small portions of the original image that have been extracted based on some image processing criteria and not the tradition sense of landmarks. More specifically, when an image is captured, a gradient image is formed at a reduced resolution. Difference of Gaussian (DOG) filtering is then used to extract curvature points. Small portions of the image (size of 32 x 32 pixels) are then extracted, with the centers being the curvature points. These images are then binarized using a log-polar transform and the resulting images becomes the landmarks. Furthermore, the azimuth (the angular position relative to the North given by a compass) is computed for each curvature point. Together, the binarized images and their corresponding azimuth values form a signature or characteristic of a particular location. For each image, 30 pairs of landmarks and azimuth values are extracted (see Figure 2.12).

Cuperlier et al. termed the coding of each location a “place cell”, similar to that found in the rat’s hippocampus. When a location is reached and the corresponding landmarks and azimuth values computed, a matching function is used to compute the distance between the current set of values with the learned sets. If the distance is below the set threshold, then a new neuron is taken on for encoding this location. The number of locations learned therefore depends on two factors: the threshold and the environment itself. For the latter, the authors explained how more locations are learned near walls and doors due to fast changes in angular positions of near landmarks, or the appearance or disappearance of landmarks, or both.

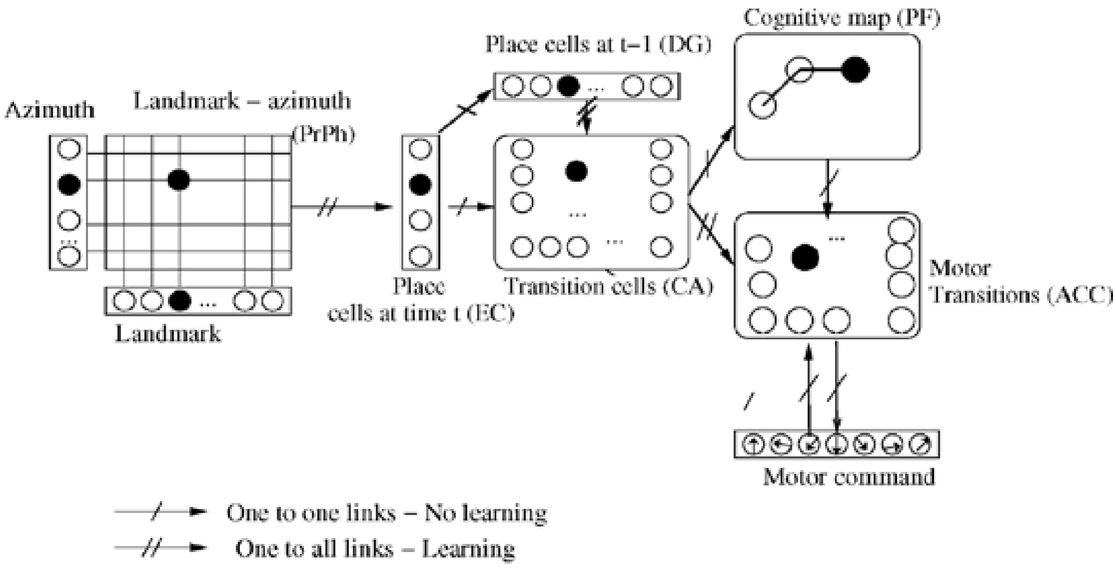


Figure 3. Sketch of the model. From left to the right: merging landmarks (Pr, perirhinal cortex) and their azimuth (Ph, parahippocampal cortex) in a matrix of neurons called product space (PS or PrPh)(maybe localized in the perirhinal and/or parahippocampal cortex), then learning of the corresponding set of active neurons on a place cell (ECs). Two successive place cells define a transition cell (CA). Place cell at time t − 1 is in DG. Transitions are used to build the cognitive map (PF) and are also linked with the integrated movement performed (ACC).

Figure 2.11 Cuperlier et al.'s neural model of cognitive mapping. Diagram and description were extracted from Cuperlier et al. (2007).

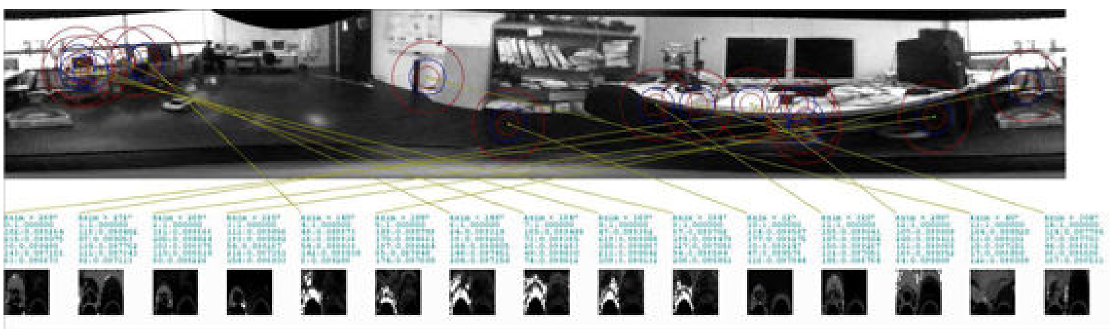


Figure 2.12 Image of the environment captured by the panoramic camera of the robot. Fifteen landmarks extracted are shown below the image and are linked to the region they were taken from. Image is reproduced from Cuperlier, Quoy, Gaussier & Giovannangeli (2005).

Another neural network is used to learn the transition information between adjacent place cells created. This network produces "transition cells"; each contains directional information linking the starting and ending location. For example, transition cell AB links location A to location B with directions on getting from place A to B, relative to the north.

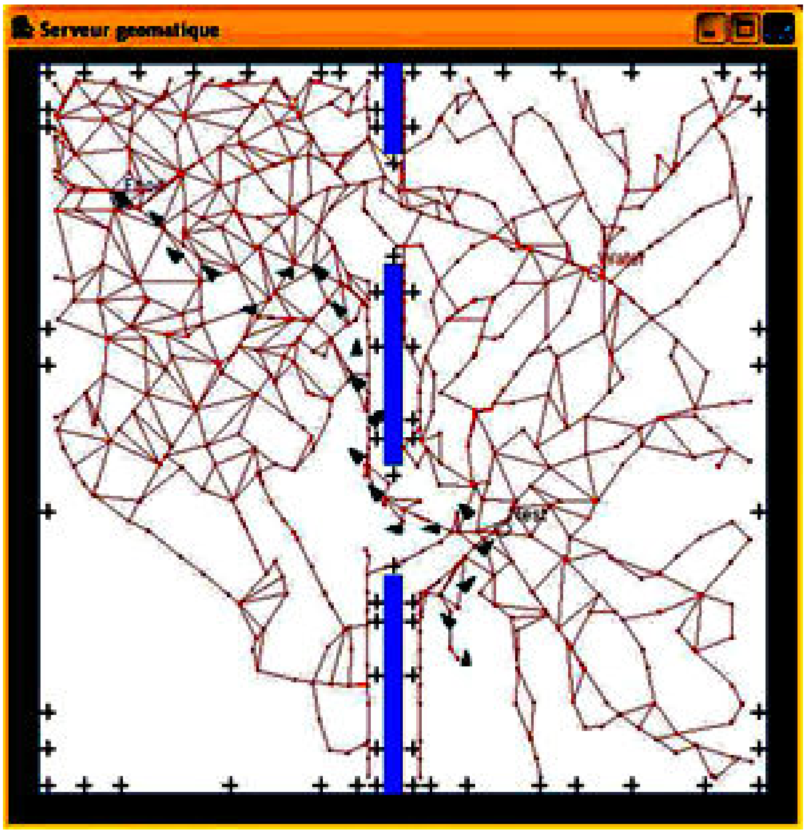


Figure 2.13 Cognitive map generated from the exploration of the robot (in triangles) starting from the bottom right to the goal on the top left. The image was reproduced from Cuperlier et al. (2005).

The cognitive map proposed by Cuperlier et al. is thus composed of nodes and arcs. Each node represents a transition. Each time a transition is used, a link or arc is created with a given value, to connect the transition used with the previous transition. If the link has previously been created, then the value of the link is increased to reinforce its usage. Conversely, values are decreased for links that are not used. Therefore, over time, a graph of nodes and arcs are created, as seen in Figure 2.13, which represents the cognitive map of the robot during a single exploration starting from the bottom right to the goal on the top left of the image. The triangles represent the successive locations of the robot during its journey.

2.4.3 RatSLAM

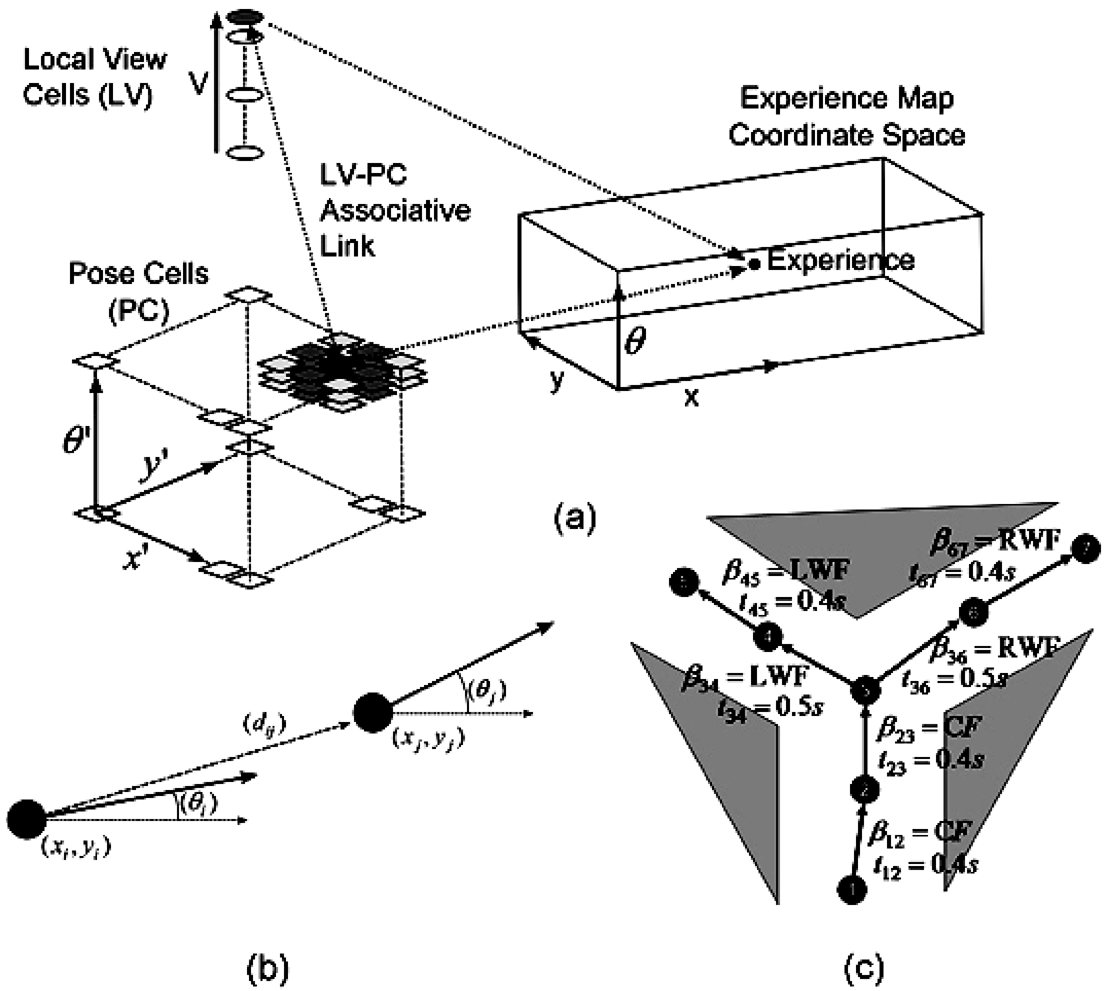


Figure 2.14 The RatSLAM model. (a) Incorporation of an experience map into the representation. The experience map has its own coordinate space and is associated with a certain pose and local view cells. (b) Connectivity between two experiences. The dots and arrows show the pose of the experiences whilst d_{ij} is the odometric distance between the two. (c) Example of the representation of information where the solid dots are the experiences and CF, RWF and LWF are the movements: CF – Centerline Following; LWF – Left Wall Following; RWF – Right Wall Following. Images were redrawn from Milford & Wyeth (2007).

RatSLAM proposed the use of three different representations namely, pose cells (robot's position and orientation), local view cells (containing visual information), and an experience map (see Figure 2.14). Initially, RatSLAM (see Milford & Wyeth (2003)), following proposed models of hippocampus by Arleo (2000) and Redish (1999), uses two separate networks to learn the position and orientation information but this was found to be inadequate, as described in Milford & Wyeth (2007). To ensure proper learning, there is a need to create place cells with different poses, thus giving rise to a single network for pose learning. Unfortunately, doing so could result in different pose cells representing the same physical place and multiple physical places associated with the same pose cells. An additional network is then proposed to learn a unique map (or in their terminology, experience) of the environment from the pose cells and local views. This latter map, known as experience map, has its own its own global co-ordinate system. Each node is a snapshot of activity within the pose cells and the local view cells. Links between nodes capture behavioral information that enables one to move between the nodes.

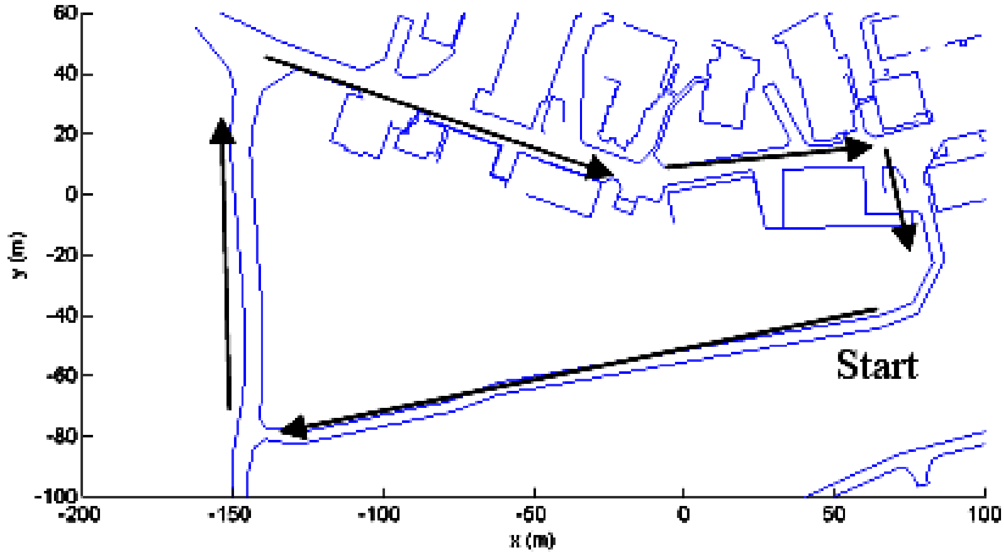


Figure 2.15 Map of the outdoor environment explored by the robot. The path the robot took is represented by the arrows. Image has been reproduced from Prasser, Wyeth & Milford (2004).

RatSLAM has recently been tested in an outdoor environment (see Figure 2.15) using a small tractor equipped with odometry and an omni-directional camera, see Prasser, Milford & Wyeth (2005) and Prasser, Wyeth & Milford (2004). Figure 2.16 shows the trajectory of two laps in the same direction superimposed.

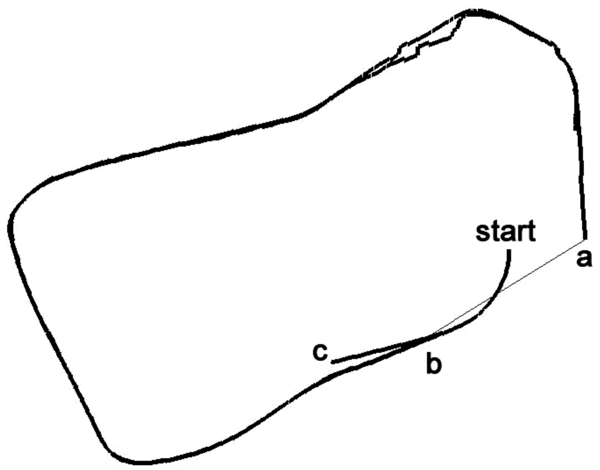


Figure 2.16 Trajectory of the robot performing two laps in the same direction. Image was redrawn from Prasser et al. (2004).

It was pointed out that RatSLAM made an error at the start of the first lap where it incorrectly re-localizes from point c to point b. However, it successfully recognized that it was in a loop during its second lap when it realized that point a is point b after completing the first lap. Figure 2.17 shows the result obtained after RatSLAM has completed three forward laps and one backward lap. In the first lap, the robot went from point b to point a, and then reversed its journey by rotating 180 degrees. It was able to maintain the correct path until point d. In the subsequent 2 laps, the robot is confused. The failures of RatSLAM in both figures were mainly attributed to its poor visual system.

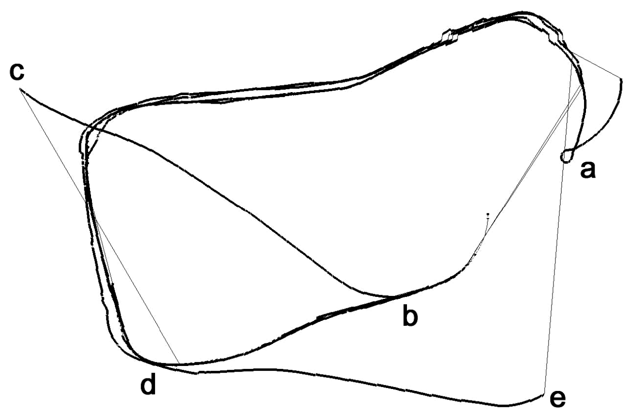


Figure 2.17 Trajectory of the robot performing three laps in the forward direction and one lap in the reverse direction. Image was redrawn from Prasser et al. (2004).

2.5 Discussion

This chapter shows that there has been a significant progress in the development of computational models of the cognitive mapping process and especially, significant efforts have been made towards their implementations on a mobile robot.

The SSH model provides a clear description of the interplay of the many different kinds of knowledge that are involved in a cognitive mapping process. Kuipers and his co-workers have begun serious investigation into the perceptual end of the SSH model and this could lead to significant implementations of the SSH model on a mobile robot.

The PLAN model is similar to the SSH model in that both models are developed from an investigation into the very extensive findings about the human cognitive mapping process. However, they differ from their point of focus: SSH is centered on the notion of a place representation while PLAN is centered on the way-finding problem. In particular, PLAN is influenced by the idea that when finding one's way, one first remembers landmarks and builds local maps along choice-points. Consequently, a cognitive map with multiple networks has been proposed. It will be interesting to see further attempts to implement PLAN on a mobile robot.

The current thesis aims to work towards a better understanding of the ASR model. Like the current work on SSH and PLAN, this thesis is also about implementing the model on a mobile robot. As can be seen in these attempts, implementing a theory of cognitive mapping onto a mobile robot is not a straightforward task. On the one hand, if one were to simplify the theory too much, then the implementation does not capture the essence of that theory. On the other hand, if one were to implement the theory in full, the robot is not a system that is powerful enough (yet) to do so. The approach taken here is to strike a balance – treat the robot as a cognitive agent of its own kind and implement and test aspects of a cognitive mapping theory with the robot.

For completion, three recent neural-inspired models of cognitive mapping were also reviewed. They demonstrated a popular use of vision as input to their mobile robots but that is partly because they could extract with ease some form of a signature of a place to do place learning and recognition using a neural network. They also created a very dense network of places and this might pose a problem when the problem is scaled up. In this respect, RatSLAM is interesting – it was tested in a real-world environment. However, and as noted by its creator, its success depends on successful visual recognition of places visited. This begs the question: is cognitive mapping about remembering everything that one has seen? It would be interesting to see further development in this area using real world environment.

Finally, RatSLAM reinforced the importance of implementing and testing various theories of cognitive mapping on a mobile robot. It has shown that a popular hippocampus model of representing pose using two separate networks is flawed.

Chapter Three

Neurological and Behavioral Studies on Cognitive Mapping

As noted in the Introduction, an idea central to the work reported in this thesis is the development of a robot that can, in a sense, behave like a “cognitive agent”, and perform cognitive mapping in its own environment. This chapter thus reviews some recent research (mostly those published since 2000) on how (real) cognitive agents such as humans, rats, fish, ants and others perform cognitive mapping. Since Chapter 2 provides a review of both computational and neural models of cognitive mapping, this chapter will thus review both neurological and behavioral studies of cognitive mapping.

Note that on the one hand, the cognitive mapping process is, in general, a complex process (see chapter 2) and on the other hand, the robot used here to test ideas about cognitive mapping is an unsophisticated agent. Furthermore, the aim here is to use the robot to help us understand a very early stage of the cognitive mapping process: how could a network of fuzzy ASRs be useful? What is made explicit in these ASRs are the geometrical shape and exits of the local environments. The focus of this review will be on whether cognitive agents pay attention to such information and if so, how? The review is not meant to be a comprehensive review on the behavioral and neurological literature on cognitive mapping.

Sections 3.1 and 3.2 provide a review of recent neurological and behavioral studies of cognitive mapping respectively. For this thesis, the former studies will include all those works that examine the role of the brain neural structure in relation to cognitive mapping and the latter will include all those works that exclude any explicit examination of the role of the brain neural structure. Section 3.3 concludes this chapter with a discussion of the insights gained from this review.

3.1 Neurological studies

In 1971, O'Keefe and Dostrovsky discovered that place cell activity in the hippocampus of rats was closely related to the rodent's location in its environment, see O'Keefe & Dostrovsky (1971). This discovery led O'Keefe and Nadel (1978) to propose that the hippocampus functions as a cognitive map and has since become the cornerstone of many neurological studies. (for alternative viewpoints, see Eichenbaum, Dudchenko, Wood, Shapiro & Tanila (1999) ; Frank & Brown (2000) and Jeffery, Gilbert, Burton & Strudwick (2003)). Figure 3.1 shows where the hippocampus lies in the human brain.

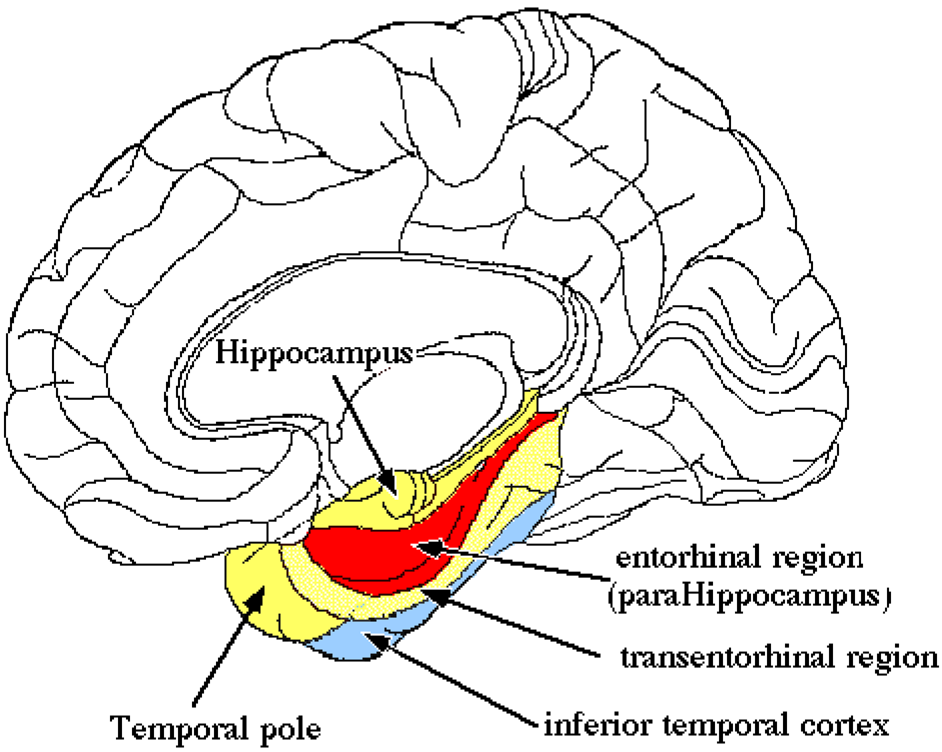


Figure 3.1 Location of the Hippocampus and entorhinal region within the human brain. Image was reproduced from (*Memories*, 2006).

Section 3.1.1 presents some recent studies that continue to support the role of the hippocampus as a cognitive map. Sections 3.1.2 and 3.1.3 present some recent studies that investigated the learning of geometric information at the neural level, the hippocampus and the parahippocampal region respectively. Section 3.1.4 concludes with a cautionary note on the interpretation of such work. For a review of earlier neurological studies, see Best and White (1998).

3.1.1 Hippocampus and Cognitive Maps

Rivard et al. (2004) investigated the rat's ability to adapt rapidly to dynamic environments. In their study, a rat was placed in a cylindrical arena which contains a tall barrier. From their experiments, they found a new class of cells within the animal's hippocampus which relates to the proximity to the physical barrier. The following observations were recorded during the experiment:

1. When the barrier was fixed, the cells appeared to be ordinary place cells,
2. When the barrier was moved, the activity of these cells moves equally,
3. When the barrier was removed, the cells stop firing, and
4. When the barrier was placed in a different arena (and hence place cells was totally changed from previous arena), these cells continue to discharge at the barrier.

From their findings, Rivard et al. concluded that the barrier and place cell activity represents the current arrangement of the environment.

Kobayashi et al. (2003) investigated the changes in the neural activity as the rat learns to navigate efficiently to acquire rewards. Hippocampal activity was recorded as the rat learned the locations of the rewards, and gradually developed efficient navigation strategies. They reported a change in spatial firing in some neurons, as learning proceeded and intensive firing when the rat is near the reward site, once efficient navigation was established. They consequently suggested that the hippocampal neurons play a crucial role in the formation of efficient navigation.

Gagliardo et al. (1999) investigated whether the hippocampus is needed for landmark learning. They compared the differences between hippocampal lesioned pigeons and control pigeons in the way they acquire the spatial representation of the environment. The control pigeons were found to be able to learn new landmarks which allow them to perform "pilotage", or the use of landmarks without reference to the sun to guide their flight home. The lesioned pigeons relied exclusively on the sun to guide their way home. They were only able to learn to use familiar landmarks at the training location, to recall the compass orientation based on the sun, referred as "site-specific compass orientation". They thus concluded that the hippocampal formation is required for the birds to learn a spatial representation based on numerous independent landmarks, which can then be used later to directly guide them home. For a similar study, see White, Strasser, & Bingman (2002).

Maguire et al. (1997; 2000) investigated the neural activity of London's taxi drivers. In Maguire et al.'s (1997) experiment, PET (Positron Emission Topography) of experienced taxi drivers showed activation of a network of brain regions, including the right hippocampus, when they were asked to recall complex routes around the city. The cab drivers were then asked to recall famous landmarks which they have no knowledge of their location, resulting in similar activation of the brain regions, except for the right hippocampus. Consequently, it was suggested that the hippocampus is involved in processing spatial information that is established over a long time, whilst other parts of the brain maybe responsible for any topographical stimulation. See also Ekstrom et al. (2003) and Rosenbaum et al. (2005).

3.1.2 Hippocampus and Geometric Information

Vargas et al. (2004) investigated the connection between the hippocampal formation and the learning of geometric information. In the study, pigeons with lesions to the hippocampal formation were trained to search for food in a rectangular arena with a wall of a different color. The impaired birds were found to have relied exclusively on the wall of a different color for navigation (featural information). The control pigeons were able to encode and use both geometric information (namely the shape of the environment) and color cues; and when one was absent, the birds used on the other. They concluded that the results demonstrate that the hippocampus is also important for learning geometric information of space in pigeons.

Similar results were obtained by Tommasi et al. (2003) from their study with domestic chicks (*Gallus gallus*). In the experiment, chicks were trained to search for food at the centre of an enclosure, which is next to a landmark. When the landmark was removed, the sham-operated chicks and chicks with lesion of the left hippocampus were able to correctly locate the food source by utilizing geometric information provided by the enclosure. However, chicks with lesion of the right hippocampus or both hippocampi were completely disorientated, when the landmark was removed; and searched around the landmark, when the landmark was displaced. Like the pigeons, this result shows that the chicks encoded both landmark and geometric information when available and in the absence of one information source, the other would be used for localization. More importantly, this result also shows that the left and right avian hippocampi have different roles in spatial cognition.

Lesion study on rats by McGregor et al. (2004) also found strong evidence which suggest that the hippocampus is responsible for mediating geometric information. Control rats showed that the animal is able to utilize the shape information provided by physical barriers (walls), and array of landmarks. However, rats with excitotoxic lesions of the hippocampus were impaired and were unable to use the geometrical information.

Long term exposures to differently shaped environments have also been found to affect the hippocampal place cell representations (see Lever et al. (2002)). Place cell activities show gradual and incremental divergence according to the shape of the environment. More importantly, this divergence was further observed when the subjects were transferred to new enclosures of the same shape, further indicating the encoding of geometric information in the hippocampus.

3.1.3 Parahippocampal and Geometric Information

Epstein et al. (1998) observed that there is a region within the parahippocampal cortex which responses more strongly when subjects are exposed to navigation visual stimuli such as street scene, buildings and landscapes. This occurs even if it is only a change in viewpoint of the scene, than when they are exposed to other visual stimuli such as faces, objects, etc (see Epstein et al. (2003)). These responses were found even when subjects were not engaged with navigational tasks. Consequently, they named this region the “Parahippocampal Place Area” or PPA (see Figure 3.1). An interesting discovery is the strong activation of the PPA region when in the presence of information about the shape and layout of the immediate environment, regardless of objects within the environment. In particular, it responds more strongly to depictions of surfaces that in some sense “enclose” the observer and define a space within which one can act, than to depictions of surfaces that define objects that the observer can act upon.

Furthermore, they also recorded that the PPA reacted no more strongly to a photograph of a room filled with furniture and other objects than to a photograph of an empty room. However, these activations are reduced when the room is rearranged to no longer define a coherent space. Consequently, they propose that the PPA represents places by encoding the geometry of the local environment.

In Epstein et al. (1999), the authors reported that the PPA activity is not affected by the subjects’ familiarity with the place depicted. Furthermore, a study by Epstein et al. (2001) on two patients that damaged the PPA from vascular incidents showed that they suffered memory problems for topographical materials and were unable to navigate unassisted in unfamiliar environments. The subjects showed deterioration in their visual memory performance, namely: scene-like stimuli were significantly weaker than object-like stimuli.

3.1.4 Conclusion

There are many studies that continue to investigate place cell activities in relation to spatial mapping in the environment. A few of these studies were reported in section 3.1.1. What is interesting to note is that the more recent studies are not just concerned with place cell activities in relation to a place but also to other characteristics of a place which are identified as important for cognitive mapping (such as landmark and shape of each local environment).

The discovery that PPA, being another possible site for cognitive mapping, responds strongly to depictions of surfaces that in some sense “enclose” the observer is interesting. This is because such surfaces are exactly those that have been argued by Yeap & Jefferies (1999) to be essential for computing ASRs.

However, it should be noted that it is always problematic trying to explain what truly happens when place cell activities are observed. For instance, not all researchers agree with these findings. Pearce et al. (2004) for example, reported that rats did not use the overall shape of the local environment to locate a goal, but rather relied on local cues of the environment for navigation. Thus rats with lesion of the hippocampus were impaired in their ability to use these cues and not the overall shape of the local environment. In pigeons, some experiments (for example, Pearce et al. (2005)) showed that the hippocampus is not essential for structural discriminations but is important for processing some types of spatial information.

3.2 Behavioral Studies

Psychologists involved in cognitive mapping research are interested in the mental process which handles the acquisition, coding, storing, recall and decoding of information on the spatial environment. In fact, the term *cognitive map* was first coined by an American psychologist, Edward Chace Tolman. Through experiments with rats, Tolman (1948) demonstrated that the animal could encode spatial information of its environment which could then be used later; rather than simply learning responses triggered by environmental stimuli.

In more recent studies, it is widely accepted that Cheng's (1986) experiments with rats provided the initial motivation for studying whether the overall shape, or geometry of the environment, is encoded as part of the cognitive map, and if so, how it is used. In his experiments, Cheng trained the rats to search for food that was hidden in a corner of a rectangular enclosure with various orientations. Each of the corners was marked by a distinct featural cue.

Cheng noted that the rats could have easily solved the task by encoding the featural cue that was of the same corner as the hidden food. However, the experiments revealed that the rat still made errors. Upon examination, Cheng discovered that the rats made more errors in the corner diagonally opposite to the correct corner, than the expected even distribution of all the incorrect corners. This led Cheng to conclude that the rats had only encoded geometrical information of the environment, and as a result, the two corners were not distinguishable since they are both geometrically correct. Cheng termed this event systematic rotational error. The setup of Cheng's experiment is shown in Figure 3.2.

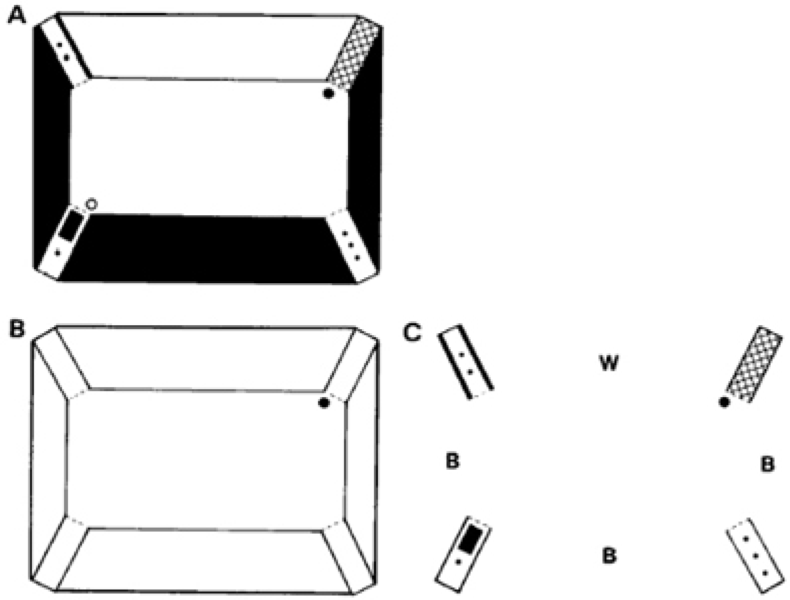


Figure 3.2 (a) Experimental setup where the rectangular box has three black walls and a single white wall. The filled circle identifies the location of the hidden food and the open circle is its geometrically equivalent corner. Each corner has panels with different visual, tactile and olfactory characteristics. (b) Geometrical relation of the test environment when featural cues are absent, showing it is not possible to distinguish between the goal and its geometrical equivalent. (c) Arrangement of featural cues with letters W and B representing the color of the walls. Encoding the goal with respect to the featural information can easily be used to solve this problem. The image was reproduced from Cheng (1986).

Since Cheng's experiment, there was an explosion of interests in trying to find out how animals deal with geometrical information about a local environment and in particular its shape. The review in this section will focus on these experiments. Sections 3.2.1 reviews experiments showing the use of geometrical information in spatial mapping, and section 3.2.2 reviews experiments involving both geometrical and landmark information. In Section 3.2.3, a further review is done on experiments on paths integration by ants. How the ants find their way home would be of particular interests to the research conducted here. This is especially true when the problem solved and the sensors used are both compatible. Section 3.2.4 concludes with a discussion of the significance of the findings.

3.2.1 Utilizing Geometrical Information in Spatial Mapping

Burt De Perera (2004) presented a very interesting study on whether the size and shape parameters are encoded for spatial tasks. The author experimented on the blind Mexican cave fish (*Astyanax fasciatus*) and noted that this species is excellent for this study as it lacks vision; meaning cues from visual landmarks are not available. As the fish glides, a flow-field is created around it. This field is affected by nearby objects and is monitored by the animal's lateral line organ. The faster the fish swims, the more stimulated the organ becomes. Therefore, when the fish encounters unknown environments, it swims at higher velocities to collect information on its new surroundings. Over time, the velocity decreases when the fish gets accustomed to its new enclosure.

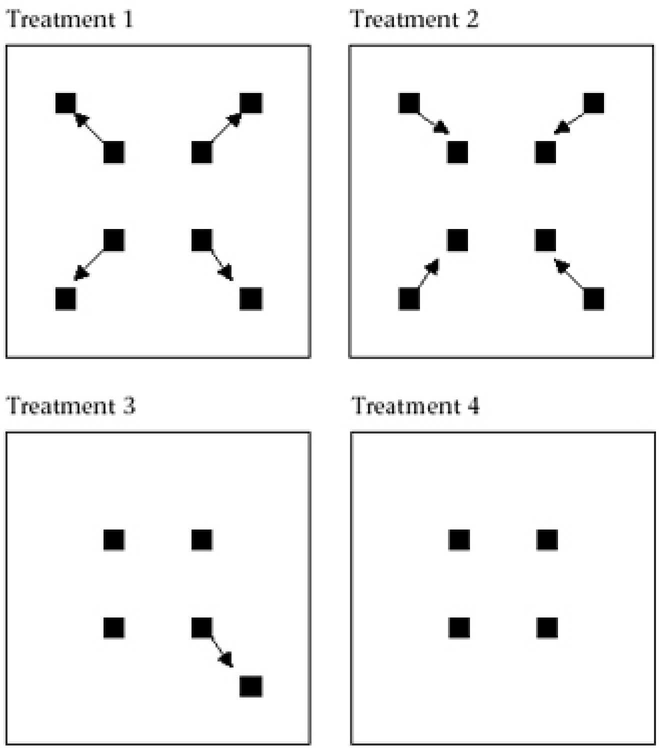


Figure 3.3 Diagram showing the change in the layout of the landmarks for testing the size (Treatments 1 and 2) and shape (Treatment 3) of the fish's surroundings. Treatment 4 represents the original configuration, which sets the benchmark for the experiment. The image was reproduced from Burt De Perera (2004).

In the experiment, the blind Mexican cave fish initially learned an array of four landmarks. To test whether the fish can encode the size and shape information, the array was expanded and contracted (for testing the size parameter); and the array was distorted (for testing the latter parameter). Figure 3.3 shows the changes made to the landmark's arrangement in the study.

The experiments recorded a significant increase in swimming velocity for both conditions and when the array was modified back to the original configuration, there was no significant difference in velocity. Hence, the author concluded that the fish must have reacted to the difference between its perception of the new environments and its representation of the old one, learnt during training. The author suggested that the results show that this species of fish is able to encode the distances between the landmarks and maybe even the shape (from the landmark configurations) within the enclosure.

Experiments by Sovrano et al. (2002, 2003) found that fish (*Xenotoca eiseni*) is also able to use purely geometric information for reorientation. When tested in a rectangular tank without any featural cues, the animal chose the geometrically correct corners more frequently than the other two corners. Featural cue in the form of a blue wall was then introduced into the environment. The fish was clearly able to distinguish and choose the correct corner, which suggests that fish are able to use both geometric and non-geometric information in conjunction for reorientation. Further investigation also revealed that the fish encoded geometric information even when featural information alone sufficed to solve the spatial task. Test also showed that transformations that altered the geometrical relationship between the target and the shape of the environment had more effect on the reorientation performance of the fish. For similar studies but using monkeys, see Gouteux, Thinus-Blanc & Vauclair (2001).

Studies on the use of geometrical information are not limited to non-human animals. Waller et al. (2000) researched into the process by which humans establish a memory of a location. Specifically, the authors looked into the use of distance and angular information between landmarks. In their training experiment, subjects had to repeatedly learn the location of a target in relation to three distinct landmarks in a virtual environment. They were then required to return to the target location during testing, see Figure 3.4.

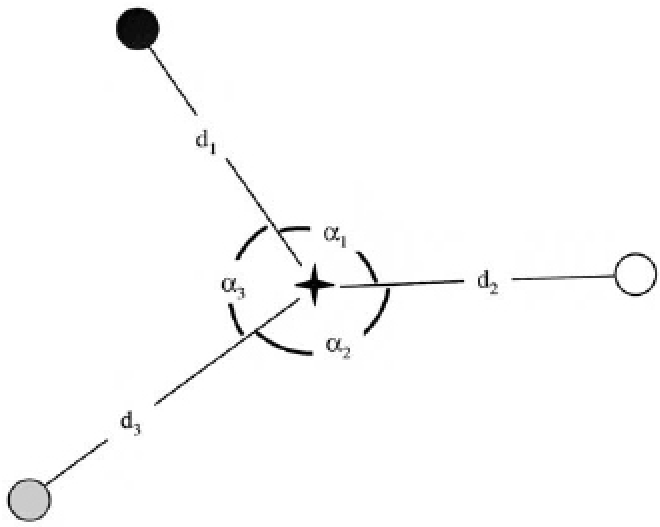


Figure 3.4 Human experiments conducted by Waller et al. Subjects learned the location of the target (star) relative to the three landmarks. The experiment affords geometric information on the distances between the target and each of the landmarks (d_1, d_2 and d_3) and the direction of each landmark (α_1, α_2 and α_3). The image was reproduced from Waller et al. (2000).

To test the reliance on the distance and angular information between the landmarks, the configuration of the landmarks were altered. Subjects displayed greater dependency on distance over angular information, with two exceptions:

1. when the relationship between the target and the enclosure was distorted and;
2. the prominent existence of right angles within the configuration during learning

Waller et al. proposed that these result could suggest the existence of a multi-level organization of spatial memory or information handling such as that suggested by Huttenlocher et al. (1991). Waller et al. argued that there are two level of encoding: a "coarse" and a "fine-grain" level. Using that paradigm, they suggested that the enclosure could serve as "coarse" level information and metric data could serve as "fine-grain" level of information.

The authors also pointed out another important finding from their experiments. That is, the subjects' performance did not depend on the accurate perception of absolute distances. The relative distances to landmarks were sufficient for successfully completing the spatial task. Hence, any errors incurred had no bearing on the subjects' performance.

Kelly and Spetch (2004a) wanted to determine whether human adults encode geometric and featural information. To do so, the authors experimented with human subjects in a two-dimensional schematic of a rectangular room. The subjects were required to locate a hidden goal, which has a position that is constant, relative to featural and geometric cues but its absolute position varied across trials. Results from the experiment showed that adult humans were able to learn and use featural information with ease to find the goal. Cues such as color and shape of distinct features were used to encode the target's location. Even when these features were removed, the subjects used distant features to reorient themselves.

More importantly, as well as featural information, the experiments also showed that the subjects also encoded geometric properties of the environment. However, they found it difficult to learn only geometric properties of the environment. Therefore, it is not surprising that they preferred featural information when geometric and featural information were contradicting each other (unlike pigeons; see Kelly and Spetch (2004b)).

Whilst most researchers agree that the use of geometric information is inherent in animals, what and how geometric information is used remains a contentious issue. In recent experiments by Pearce et al. (2004), rats were trained to find a submerged platform located in a corner of a rectangular arena. The overall shape of the arena was then altered to form the shape of a kite, and the rodents were tested to determine their search pattern in the new environment. The results revealed that the rats focused their search on the geometrically equivalent corner of the arena. Figure 3.5 depicts the experimental setup used for this experiment.

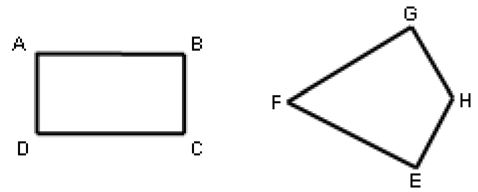


Figure 3.5 Shapes of the enclosures that were used for the experiments by Pearce et al. The image was reproduced from Pearce et al. (2004).

Pearce et al. argued that the animal used references to local features of the environment, rather than the overall shape of the environment. They further suggested these local features could include featural cues such as lengths of walls of the testing environment and geometric properties of the corner where the goal was located.

Tommasi and Polli (2004) presented results that supported this view. In their study, domestic chicks (*Gallus gallus*) were trained to search for food that is located at a corner of a parallelogram shaped enclosure. For each test, the chicks were disorientated and the enclosure rotated. The shape of the enclosure was chosen spontaneously between a rectangle (to test reorientation based on the lengths of the wall) and a rhombus (to test reorientation based on the angles between walls). Reorientation was possible based on these two local cues only. Results from the experiment show that both of these features were encoded, and in the final test where the two features conflicted (mirrored parallelogram), the chicks relied on salience of corner angles. Tommasi and Polli believe that the local information provided by the corners and the walls enabled the animals to reorient themselves in the new environment.

Cheng and Gallistel (2005) however, disagree with the conclusions offered by both Pearce et al. (2004) and Tommasi and Polli (2004). Even though they agree that matching of the similarity of shapes is unlikely in the presented cases, they believe that the results of the two studies can be accounted for by the matching of the shape parameter of the first principal axis. In short, the first principal axis is the long axis of the shape which passes through the centroid of the object. The location of the goal can be encoded relative to the first principal axis.

For example, the target is located to the left of the centre of the principal axis, or the target is at the end of the axis, and so forth. Consequently, they argued that the similarities in goal location relative to the first principle axis between the training and test enclosures enabled the subjects to successfully complete their goal. Figure 3.6 illustrates how this explanation was applied to the results acquired by Pearce et al. (2004) in Experiment 1A (see Figure 3.5). See Cheng and Gallistel (2005) for their explanation of the results recorded in Tommasi and Polli (2004).

However, they do not doubt that animals do use local features for solving spatial tasks and hence, they did not discounting its use completely. They suggested that a combination of global processes, such as matching by an axis of symmetry, and local processes coexists to help decrease the computational complexity for solving spatial tasks.

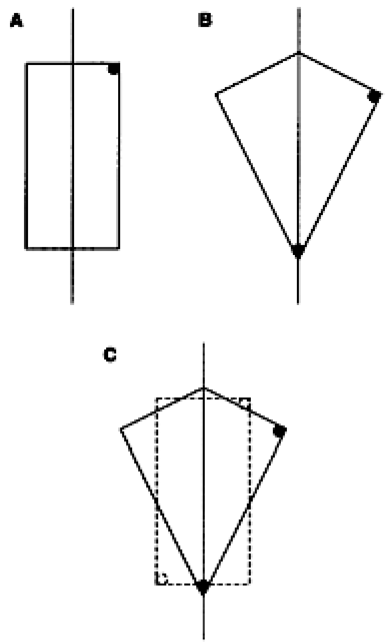


Figure 3.6 Cheng and Gallistel's (2005) explanation of the results recorded by Pearce et al.'s (2004) Experiment 1A. Diagram A: shows the training environment where the rats searched the target location (marked by the solid dot), as well as the diagonally opposite corner. Diagram B: Rats were then transferred to a kite-shaped pool and the results shows that they searched predominantly at the two corners indicated by the solid dots. Diagram C: Training and test environments are superimposed onto the principal axis. Cheng and Gallistel explained that Pearce et al.'s results can be accounted for if the rats chose the corners situated at the end and as far right as possible of the principal axis. The diagram was reproduced from Cheng and Gallistel (2005).

3.2.2 Utilizing Geometrical and Landmark Information in Spatial Mapping

Learmonth, Nadel and Newcombe (2002) specifically wanted to determine if human children are able to use landmarks alongside geometric information for spatial tasks. Two sets of children (above and below the age of six) were tested with reorientation tasks in large and small spaces. The authors discovered that children under the age of six were unable to use featural information such as a blue wall in smaller spaces. In larger spaces, children were reported to be quite good at using the featural information for reorientation purposes.

Gouteux and Spelke (2001) conducted eight experiments on three to four year old children to determine whether the subjects were able to utilize geometric and landmark information for reorientation. The experiments were performed in an open environment with three (arranged in a triangular configuration with unequal sides) to four (rectangular formation) landmarks. The subjects were required to locate an object hidden inside one of the landmarks. When oriented, the children successfully found the target in every experiment. However, the children were unsuccessful in locating the target when they were disoriented. More interestingly, they observed that the children did not use the geometric information between the landmarks.

In contrast, when adults were tested, they were found to use both geometric and non-geometric information. This indicates that the early development of human navigation abilities depend on layout of the permanent surfaces (walls) and not from information provided by the objects within the environment, whether it be geometric or non-geometric information.

Results from previous studies by Learmonth et al. (2002) and Hermer and Spelke (1994, 1996) showed that disorientated children were able to conjoin geometric and landmark information to reorient themselves in large, but not in small spaces. In 2005, similar experiments with fish by Sovrano, Bisazza, & Vallortigara (2005) and chicks by Vallortigara, Feruglio, & Sovrano (2005) were presented but recorded different results. In the respective experiments, fish and chicks did not have any problems in using geometric information in combination with featural cues, both in large and small environments. Moreover, both chicks and fish were able to reorient themselves straight away when they were transferred from large to small experimental spaces and vice versa.

Based on the ease at which these animals accustomed to their new environment, they suggested perhaps the animals encoded and used relative metrics, rather than absolute measurements. Therefore, even when the lengths differed between the two environments, reorientation was not a problem because they distinguished long and short walls, rather they absolute lengths.

However, the two species seemed to make more mistakes in two situations. The fish made more errors based on geometric information when transferred from a small to large tank and made more errors based on landmark information when transferred from a large to small tank. The chicks did not have this problem. Instead, the chicks made more errors based on geometric information in small spaces rather than in large spaces, when geometric relations between the target and the shape of the environment were altered. This pattern is consistent with the fact that the animals prefer geometric information over featural cues in small spaces and, landmarks over geometric information in large spaces, as was confirmed in two more recent studies (see Sovrano & Vallortigara (2006) and Sovrano, Bisazza, & Vallortigara (2006)).

In Sovrano & Vallortigara (2006), the authors found that the birds used featural cue (blue wall) in large spaces but preferred metrical information in small environments. They explained that the reason could be due to the reliability of the information in each situation. In small spaces, one could obtain information on the metrical properties, such as the length of the surfaces, which may be a reliable source of spatial information (for example, the target is at the corner that has a short wall to the right and a long wall to the left). In large environments however, the subjects are not able to reliably acquire geometric information of the space and thus resorts to relying on featural information for reorientation (for example, the target is at the corner that has a blue wall to the right and a white wall to the left). However, they did not address the reason why featural information is not used in both the large and the small environments. See also Chiesa, Pecchia, Tommasi & Vallortigara (2006).

Sovrano, Bisazza, & Vallortigara (2006) investigated how redtail splitfins (*Xenotoca eiseni*) use geometry in large and small tanks. In their experiments, the fish was initially trained to reorient to find a corner that has a featural cue. A blue wall was used as the featural cue (see Figure 3.7). The cue was then moved to the adjacent wall and the fish's selections of the corner were recorded.

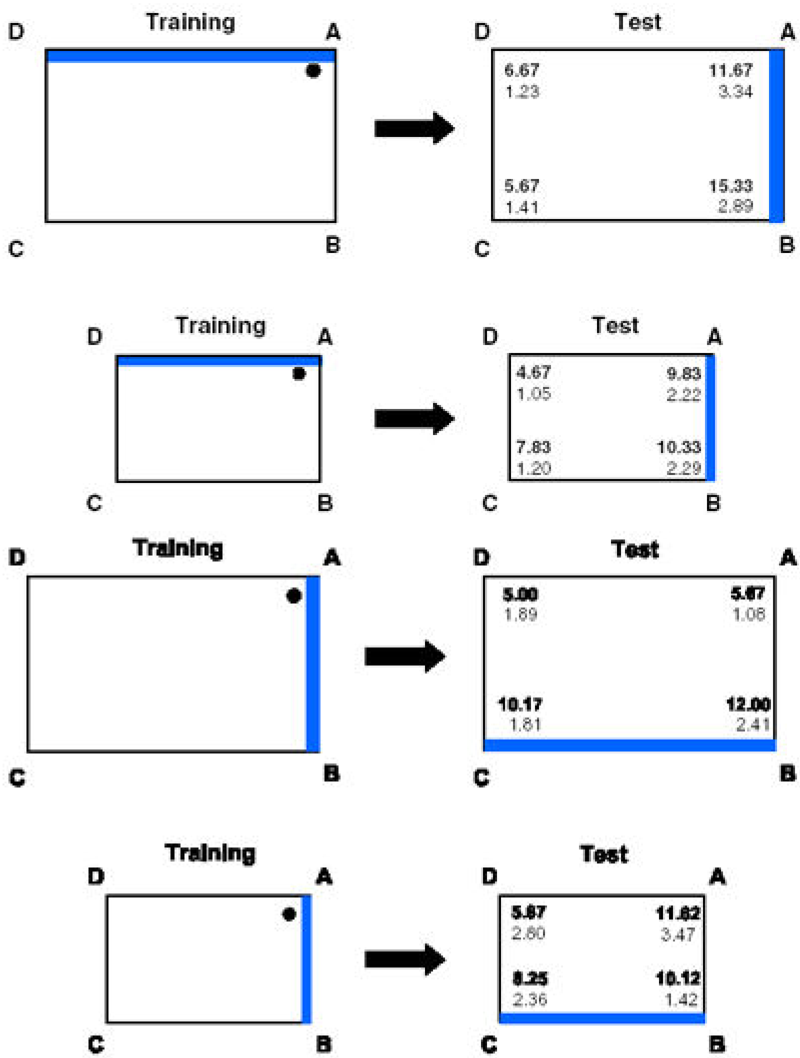


Figure 3.7 Results from the experiments conducted by Sovrano et al. The top and bottom values respectively represent the mean values of choice and the calculated standard error of the mean (SEM) for each corner. In the first set of experiments (rows 1 and 2), the featural wall was moved from the long to short edge. Rows 1 and 2 respectively show the results for the large and small enclosures. Similarly, rows 3 and 4 show the results when the wall was moved from the short to the long edge, for the large and small enclosures respectively. The image was reproduced from Sovrano et al. (2006).

Results show that, in the large tank, the fish chose the corners that have the feature and prefers the corner that preserves the correct arrangement (e.g. feature to the left of the wall, etc). In the small tank, the fish chose the corners with the features and the corner, without the feature but maintains the correct geometric arrangement. The results recorded in their study are shown in Figure 3.7.

However, unlike the chicks studied in the experiments in Sovrano & Vallortigara (2006), geometric information seems to be relatively more important than featural information for redtail splitfins. The authors suggest that this is perhaps due to the difference in species: birds have high spatial resolution, whereas fish has comparatively reduced spatial resolution.

Kelly and Spetch (2001) also reported the preference of pigeons to encode relative geometry, and not absolute geometry of the enclosure. Performed in a rectangular enclosure that eliminates all external cues, the geometric properties of the experimental apparatus were manipulated to alter the geometric information. In tests that preserved relative geometric information but altered absolute geometric information, the pigeons successfully choose the geometrically correct corners. However, when the pigeons were tested in a square enclosure, essentially distorting both relative and absolute geometric information, the birds randomly chose the corners. They suggested that this result provides evidence that relative geometric information was encoded, which is interesting, given that it contradicts earlier results obtained by the authors that showed pigeons encoding landmark arrays in absolute metric form.

Kelly and Spetch suggested that one reason could be due to the type of spatial information available. In this instance, the birds depended on the shape of the enclosure as all external cues have been removed. They suggested that using absolute metric information could be inefficient for this situation as the animal will have to travel to each corner to determine if the geometrical information matches the ones stored in memory. However, absolute distance is likely to be an important piece of information when landmarks are used to pinpoint a goal.

3.2.3 Path Integration

Wehner, Gallizzi, Frei, & Vesely (2002) presented an interesting study on the ability of desert ants (*Cataglyphis fortis*), to find its way home during foraging. This species of ant is able to leave their underground nest from a tiny hole in the desert for tens to hundreds of meters away, before returning to its nest, using only path integration to continually estimate the direction and distance relative to its home (see Wehner (1983), Collett & Collett (2000) and Wehner & Srinivasan (2003)).

Even though, the ants' behavior was already reported in 1986 in Wehner & Wehner (1986), the ants' journey to home is worth mentioning. As with any path integration system, the ant's homing system is affected by cumulative errors. Upon its departure to find home, the ant first navigates using the home vector. As a result of the odometric errors, the ants would not reach home, but would reach a location that is close to its nest. From that point, the ant switches to a systematic search strategy to locate its nest (see Wehner & Srinivasan (1981), Wehner & Wehner (1986) and Müller & Wehner (1994) for detailed account on the ant's systematic search). Furthermore, landmarks provide additional guidance in the ants to correct for odometry errors.

In the current study, Wehner et al. (2002) wanted to investigate the flexibility of the ants' path integration module to adapt to the situation where the point of arrival to the feeder from the nest is not the same as the point of departure back to the nest. In other words, vectors of the homeward and outbound journeys are not 180 degrees of each other. To do so, when the ants arrive at the feeder, they were relocated by 2.5 meters, 5 meters, 7 meters, 7.5 meters or 10 meters eastwards. Care was taken to prevent the ants from seeing any surrounding landscape. Figure 3.8 gives an overview of the experimental setup.

As before, the ants started the homeward journey along the home vector, as if they were departing from the feeder. When the animal completed the home vector, it started a systematic search for the nest. Once the food has been offloaded, the ant returned back to the feeder and the testing process was repeated. Ants that were displaced by 10m had difficulties locating the nest, and would at times take 30 minutes or more to return home. Subsequently, the animal usually did not return to the feeder the same day.

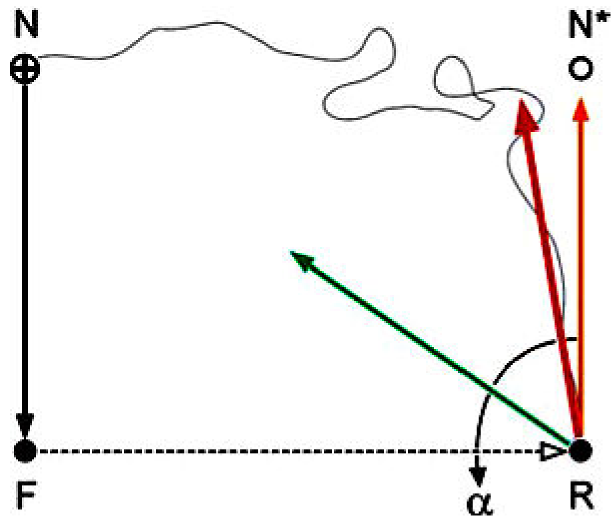


Figure 3.8 Experimental setup where N represents the actual nest and N* projects the fictitious nest for ants that have been displaced from the feeder F to the release point R. The orange arrow shows the inverse of the ant's outbound journey from the nest (N→F) and α represents the angular displacement from R→N*. The green arrow is the direction of the nest from the release point and the red arrow represents the mean vector course of the ants when subjected to the experiment. The thin line is an example of an ant's trajectory after subjected to over 50 displacements. The image has been redrawn from Wehner et al. (2002).

From the experiments, the authors found the ants recalibrated their homebound journey by shifting the home vector towards the true direction of the nest. Hence, the home vector is shifted slightly closer to the nest's direction each time the experiment was repeated. Interestingly though, the authors noted that the recalibration only lasted until the 4th repetition of testing. After that, home vector remained constant, regardless of additional training. This is something the authors were not able to explain. However, they conclude that vector calibration within the ants is indeed a fast and flexible process, but for some reason incomplete.

The experiments described by Wehner et al. (2002) investigated the ability of the ants when the direction of the homeward journey was altered. In a separate study, Cheng and Wehner (2002) experimented with the same species of ants (*Cataglyphis fortis*), to determine how the ants cope, when the distance of the outbound journey to the feeder is different from the distance back to the nest. By releasing ants at various distances away from the nest, and comparing their behavior to the control group (where control ants experienced outbound and homebound journeys of the same distance), the authors found that the ants did not increase their estimate of the global vector for the homebound journey. However, the experiments revealed the search pattern of the ants altered, with significant bias towards the nest. Consequently, the authors conclude that the desert ants can learn to modify their search pattern based on their previous experiences.

3.2.4 Conclusion

There are a number of important findings that surfaced out of the review on behavioral studies. Various species show different abilities or preferences to use different types of information.

Studies show that the representation of spatial information does not need to be an accurate description of one's environment. Humans in Waller et al.'s (2000) study for example, did not have precise distance or angular measurements to their targets. Being imprecise enables one to be flexible in adapting to new environments, hence yielding quick completion of spatial tasks. When errors occur, the performance is not affected significantly. Perhaps this is the reason why the fish and chicks in Sovrano, Vallortigara and colleagues (see Sovrano et al. (2002, 2003, 2005, 2006); Sovrano & Vallortigara (2006) and Vallortigara et al. (2005)) were not affected by the size of the environment.

Moreover, Wehner et al.'s desert ants (see Cheng & Wehner (2002); Wehner et al. (2002) and Wehner & Wehner (1986)) and Burt de Perera's (2004) blind Mexican fish showed that learning is a recurring process where new information helps the animal to constantly readjust their perception of the environment. In doing so, the representation improves with experience, rather than generating an accurate representation right at the beginning. This makes sense as the environment is a highly dynamic entity. There is no need to generate a highly accurate representation when the surroundings change constantly.

Another interesting idea is that animals have a multi-level representation of the environment, such as described by Waller et al. (2000) where different levels of detail; or types of information are stored and used depending on one's need. Cheng and Gallistel (2005), for example, hypothesized that global geometric information of the environment exist in conjunction with local features. In doing so, computation complexity can be kept to a minimum.

Furthermore, multiple strategies also increase flexibility to deal with a lack of information. For instance, the chicks in Chiesa et al.'s (2006) experiment still managed to locate the goal when featural cues were removed, albeit in a less efficient manner. Even though the chicks performed better when landmark information was available, they could also rely on geometric information, since it was simultaneously encoded during training.

Lastly, some animals appear to use different strategies to help them in their navigation or reorientation. The desert ants in Wehner et al. for example, initially used a simple strategy to get themselves close to the nest (see Cheng & Wehner (2002), Wehner et al. (2002) and Wehner & Wehner (1986)). Once within the vicinity, it started a more complex searching strategy to pinpoint home.

Another example is the different behaviors observed when animals were experimented in different enclosure sizes in Sovrano et al. (2005, 2006), Sovrano & Vallortigara (2006) and Vallortigara et al. (2005). Animals used featural cues in large spaces but preferred geometrical information in smaller environments. Perhaps in smaller spaces the animals needed to be certain of their surroundings. Therefore more accuracy was necessary to efficiently navigate within smaller environments. Consequently, geometric information was preferred over featural cues. In larger environments, on the other hand, landmarks may be the preferred cue because using geometric information requires larger memory and computation capacities. Hence, features are used as orientation cues which approximately guide animals to the required destination, and upon arrival, geometric information would then more accurately guide them to their final objective.

3.3 Discussion

The review conducted here, at both neural and behavioral level, has demonstrated that recent research focused much on the importance of geometrical information in spatial mapping. It is interesting to note that even blind fish developed a special organ, the lateral line organ, to sense their surroundings in high details. This shows nature's emphasis on computing the shape of one's local environment or ASRs according to Yeap's theory of cognitive mapping.

It is a pity that most, if not all, of these experiments focused on the ability of the subjects to re-locate the goal in a changed environment once that environment is learned. Consequently they tell us little regarding how computing the geometrical information of each local environment will help the individuals to navigate in the larger environment when they first had their experience of the environment.

The method which ants used to find their way home after foraging for food provides an interesting case study. The ants would appear to be using more primitive sensors than our robot. However, the algorithm they used demonstrated an ingenious use of distance information to return home and the ants' ability to deal with a dynamic environment. If the shape of each local environment becomes available when using more powerful sensors, more powerful strategies could be developed.

Chapter 5 investigates what kind of strategy could be developed for a mobile robot exploring its environments with sonar sensors. As we shall soon see, the strategy developed bears some similarity to the ant's strategy. But first, the next chapter investigates how the robot with its sonar sensors and odometers computes a cognitive map of its environment.

Chapter Four

ASR Computations

This chapter describes how the robot explores its environment and computes ASRs. Initially, the robot is positioned somewhere in the corridor in an office surrounding and is allowed to explore the environment until it is told to stop. The starting location is referred to as home. There are no restrictions as to where the robot can travel and no modifications are made to the environment. That is, things that already existed in the environment (such as rubbish bins, chairs, flower pots, cabinets, etc.) remain there but may or may not be there during subsequent journeys. In addition, doors leading into offices are closed or opened depending on the time of the experiment.

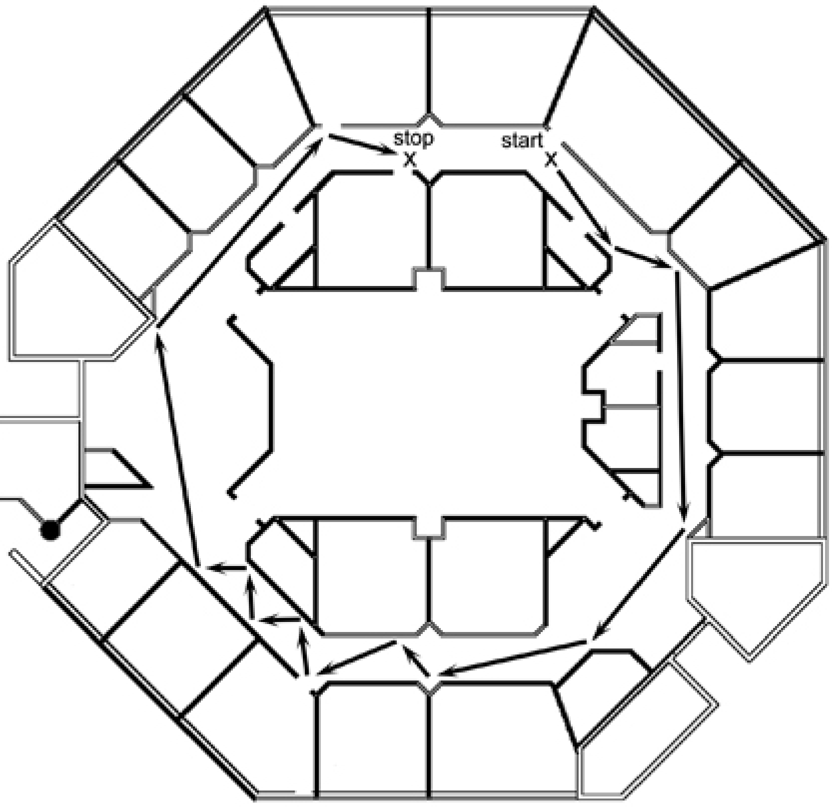


Figure 4.1 Example of a robot's journey during an exploration run.

The robot's exploration algorithm is simply to move forward in a "straight-line" manner until it is impeded by an obstacle. When it stops, it "looks" for an empty space by calculating which direction has the most free space. The robot would then turn towards that direction and continues its journey. Figure 4.1 shows an example of a journey of the robot during an exploration. Note that the environment is one section of a single floor of a three section, three level building. It has a floor space of approximately 100 m^2. Each of the arrows represents a straight-line movement, which is also known as a *path*. In certain situations (such as encountering a dead end or a tight corner), the sonar sensors would report that the robot is impeded in all directions. The experiments will then be aborted. However, such cases rarely happened in our experiments.

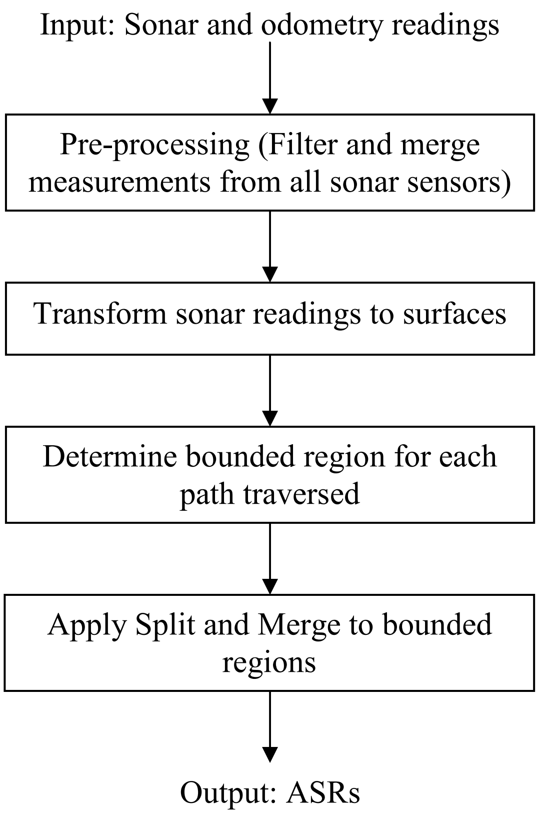


Figure 4.2 Flowchart of processes involved in computing ASRs for the environment

Information collected from the robot's sonar and odometric readings during any single exploration run is processed in 4 stages to produce the required ASRs (see Figure 4.2). Section 4.1 describes the robot and the sonar sensors used in this research. Section 4.2

describes the pre-processing of the sonar readings into surfaces. Section 4.3 describes the construction of ASRs for each path (as defined above) traversed. Section 4.4 describes the construction of ASRs for the environment experienced and Section 4.5 presents the conclusion for this chapter.

4.1 The Robot and Its Sensors

The robot platform used for testing is a Pioneer 3DX mobile robot from MobileRobots Inc. (*MobileRobots Inc*), which was formerly known as ActivMedia Robotics. It is equipped with eight ultrasonic (or sonar) sensors for collecting information of the environment and integrated wheel encoders for computing the movement of the robot. However, it is worth noting that the proposed algorithm is not restricted to using sonar sensors. In fact, the performance will improve when more densely sampled data (e.g. from laser) are available.

Sonar sensors are time-of-flight devices that work on the following principle: distance from the sensor to an object is proportional to the time taken for a signal to travel from the sensor to an object and back. Figure 4.3 gives an illustration of the physical arrangement of the eight sonar sensors, each with a maximum sensing distance of approximately 3.5 meters.

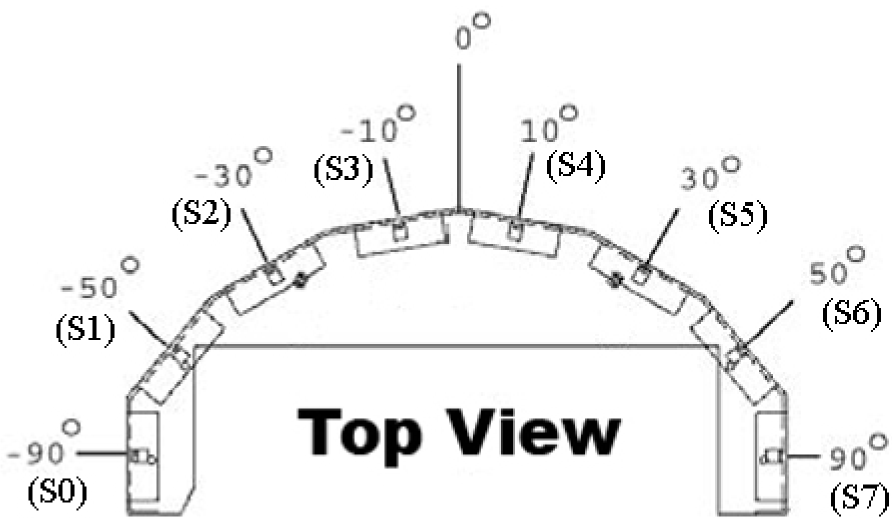


Figure 4.3 Alignment of the robot's eight sonar sensors, labeled from S0 to S7. The image was redrawn from Figure 12 in the Pioneer 3™ & Pioneer 2™ H8-Series Operations Manual (2003) and is not to scale.

Sonar sensors however, have a few major drawbacks, namely:

- They are highly sensitive to the angle of reflection from objects. If the surface of the object is not near perpendicular to the sonar signal, the signal would not be reflected back to the sensor. Consequently, the sensor would think that there is nothing in front of it. For this reason, sonar sensors are not deemed to be very reliable.
- Short measurement distance – As mentioned above, the maximum range of sonar sensors is approximately 3.5 meters. Most rooms are beyond this capacity and compared to lasers, which have a maximum range of approximately 100 meters, sonar sensors are very restricted.
- Large beamwidth (coverage area) and the physical dimension of sonar sensors, means that this type of sensors produces sparse readings. Unlike laser, which collects 360 readings over 180 degrees, only eight sonar sensors are used to cover the same range.

The robot has two solid rubber tires, driven by a two-wheel differential, reversible drive system. A caster wheel is positioned at the rear of the robot to provide balance. However, it is also a major source of odometric errors. When the robot rotates, the caster wheel provides resistance to the movement, which results in the robot rotating less than it intended to. In addition, the robot platform also suffers severely from odometric errors during translational movement. A slight change of conditions in the surface of the ground causes the robot to veer slightly to one direction. However, the robot still thinks that it is moving in a straight-line. As a result of both rotational and translational errors, the robot has a highly distorted representation of the environment.

Figure 4.4 gives an example of the effect of odometric errors on the robot's map. 'O' marks the starting point of the journey and 'X' marks the stop location. The robot's "map" (in red) is overlaid onto the actual map of the environment. Even after a short 15-meter journey, the robot's estimate of its own location is off by more than 5 meters. Since odometric error is accumulative, its estimate becomes poorer and poorer over time.

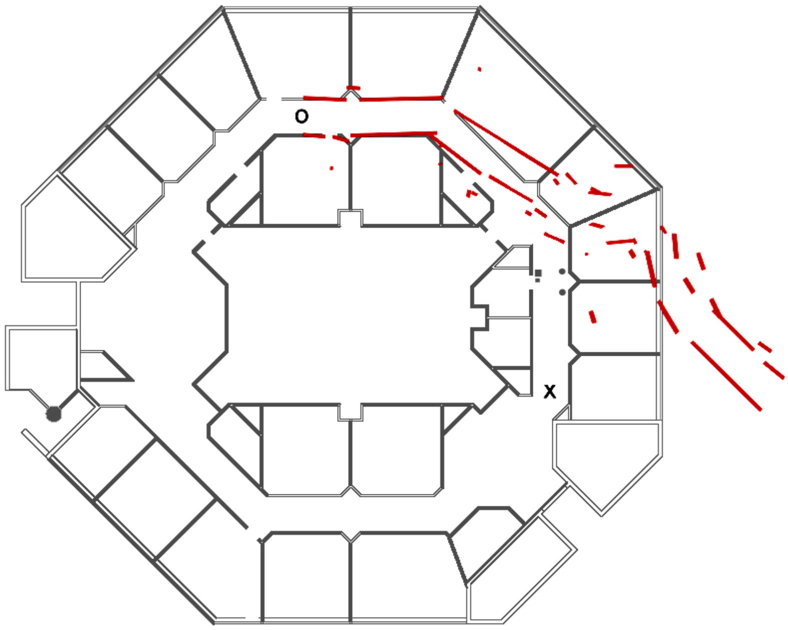


Figure 4.4 Accumulation of odometric errors significantly affects the robot's representation of the environment and its own location. The robot started its exploration at 'O' and stopped at 'X'. Red lines signify walls and objects detected by the robot's sonar sensors. The odometric error is more than 5 meters after a short 15-meter journey.

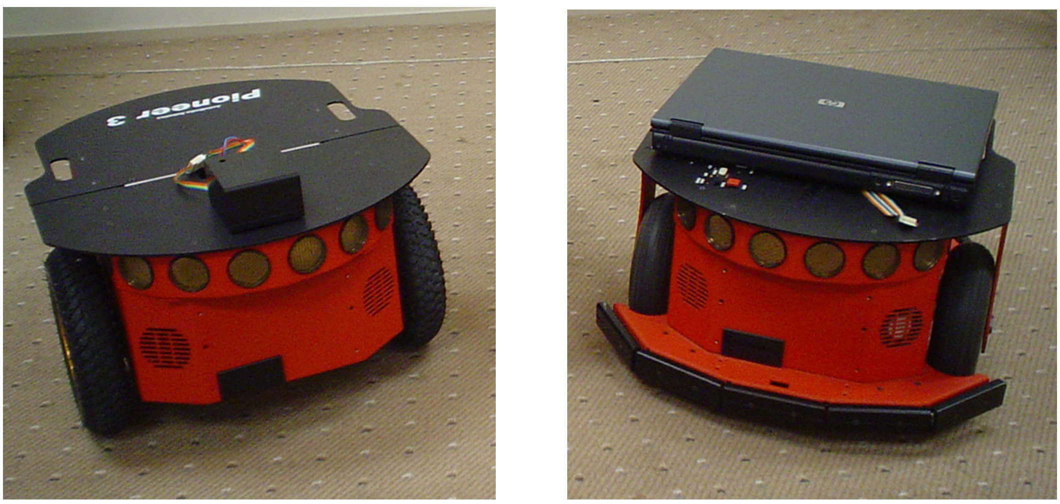


Figure 4.5 Robots used for this thesis. Both are equipped with eight sonar sensors and optical wheel encoders. The robot on the right is also equipped with bump sensors but they were not used in the experiments.

A laptop computer is used for processing information collected by the robot's sensors. It is placed in the center, on top of the robot to avoid affecting the robot's balance. If the balance of the robot is changed, it would cause the robot to drift towards one side, compounding the problem of odometric drift. The laptop is connected to the onboard microprocessor via a USB-to-serial connector. Processing can be performed "onboard" by the laptop or alternatively, the laptop can also act as a gateway through a wireless network connection so that information can be processed at other preferred terminals. Figure 4.5 shows the two robots that were used for the experiments performed for this thesis.

4.2 Pre-processing of Raw Data

The Pre-processing stage is responsible for:

1. Eliminating noise
2. Turning sonar points into surfaces
3. Combining surfaces generated from different sonar sensors to form a single representation of surfaces as perceived by all the sensors

During the data collection phase, the sonar sensors either return:

1. A valid sonar reading (a value which approximately equates to the distance to the closest object)
2. An invalid sonar reading (random values due to specular reflections)
3. A 9999 reading (which means no object detected or free space)

Invalid readings that are beyond the sonar sensors' maximum detection range (approximately 3.5 meters) are easy to nullify. Basically, any readings above 3.5 meters were considered useless, and eliminated using a filter. Figures 4.6 and 4.7 respectively show plots of the sonar readings of the environment before and after the filter was applied. The black line represents the physical movement of the robot during exploration.

The color of the plot corresponds to the sensor the data was collected from:

- Red – readings from sonar sensors 0 and 7
- Magenta – readings from sonar sensors 1 and 6
- Blue – readings from sonar sensors 2 and 5
- Green – readings from sonar sensors 3 and 4

Depending on the size of the environment tested, the amount of data from an experiment can reach an enormous proportion. Raw range readings are hard to handle and computationally expensive to process because the information is of too low level. The information must therefore be converted to a higher level representation.

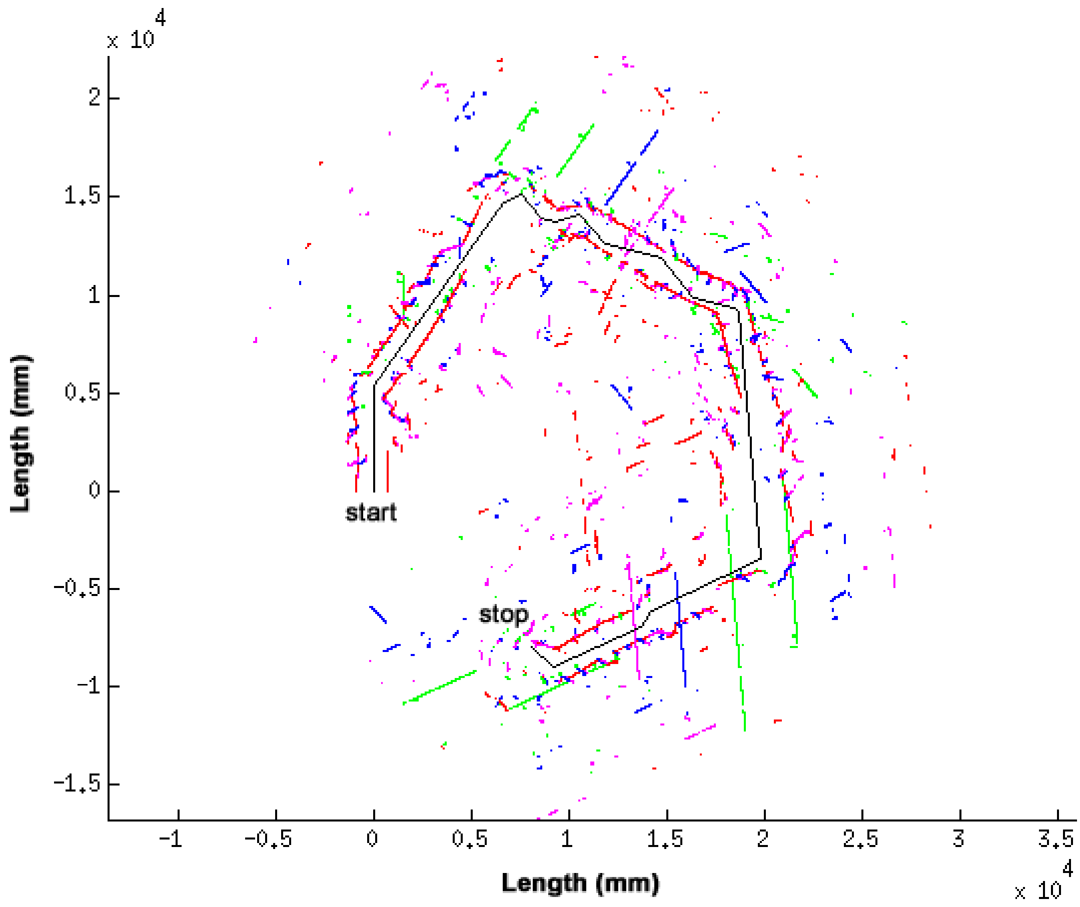


Figure 4.6 Raw sonar information collected by the robot as it moved through the environment shown in Figure 4.1. The colors correspond to the sensor that collected the data. Red: Sensors 0 and 7. Magenta: Sensors 1 and 6. Blue: Sensors 2 and 5. Green: Sensors 3 and 4. The black line represents the robot's journey.

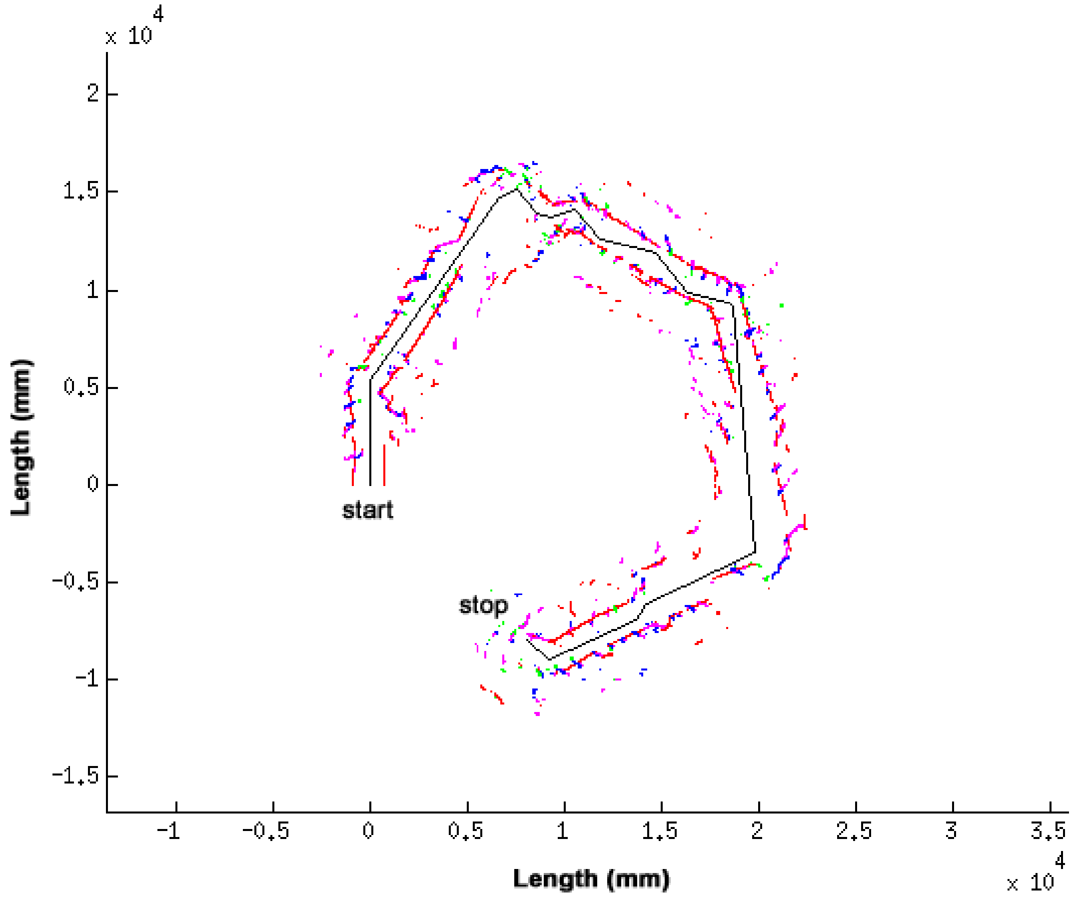


Figure 4.7 Sonar plot after readings that were above 3500mm were removed. The colors correspond to the sensor that collected the data. Red: Sensors 0 and 7. Magenta: Sensors 1 and 6. Blue: Sensors 2 and 5. Green: Sensors 3 and 4. The black line represents the robot's physical movement.

In the case of the sonar information collected here, the range readings are put through a line segmentation algorithm to generate linear representations of the environment. These linear representations correspond to the surfaces of walls and objects detected, and henceforth will be referred to as surfaces. There are many algorithms that provide linear approximations from points. Two examples are Split and Merge and Least Squared methods. For this thesis, the surfaces are simply computed by determining the spatial relationship between neighboring sonar information.

Recall that the robot has eight sonar sensors (see Figure 4.3). As the robot moves forward, sonar points collected from each sensor are first turned into surfaces independently. The line-forming algorithm is described below:

1. Cluster formation – The sonar points collected from each sonar sensor are first grouped into different clusters based upon the Euclidean distance between readings. If this value exceeds a certain threshold then a new cluster is formed.

2. Cluster segmentation – For each of the clusters formed, a further check is made to see if that cluster contains one or more surfaces. This is done by dividing the cluster into smaller clusters and then calculating their respective average gradient. If the gradients between the smaller clusters are found to differ significantly, then the original cluster would be segmented further, eventually producing two or more surfaces (see Figure 4.8).

3. Surface formation – Surfaces are formed by drawing a straight line between the first and last sonar readings of each of the clusters formed in the second step.

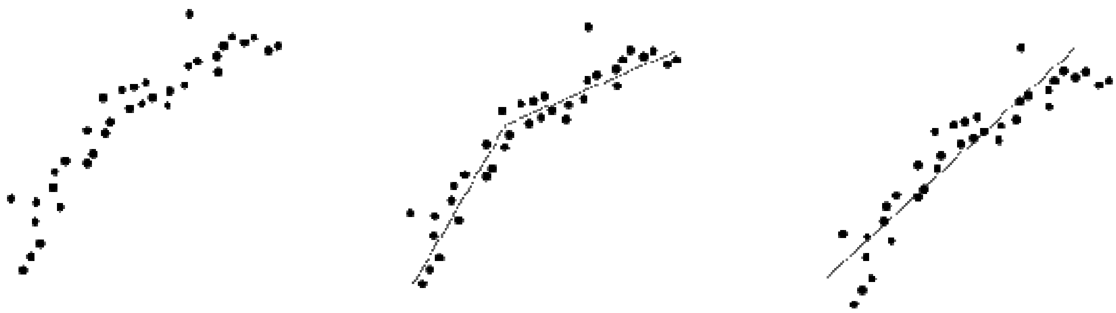


Figure 4.8 Example showing the need for cluster segmentation. The image on the left shows a cluster of sonar readings. The image in the middle shows two line segments used to represent the cluster and on the right, the image shows a single line of best fit. Even though both representations are correct, the extra information from the angles between surfaces maybe useful for later computations.

The above line-forming algorithm is a crude algorithm for generating a linear representation from the sonar points. However, it is adequate for the purpose here since what is needed is only a rough description of the environment. Figure 4.9 illustrates the linear representation were formed from the information collected from each sonar sensor.

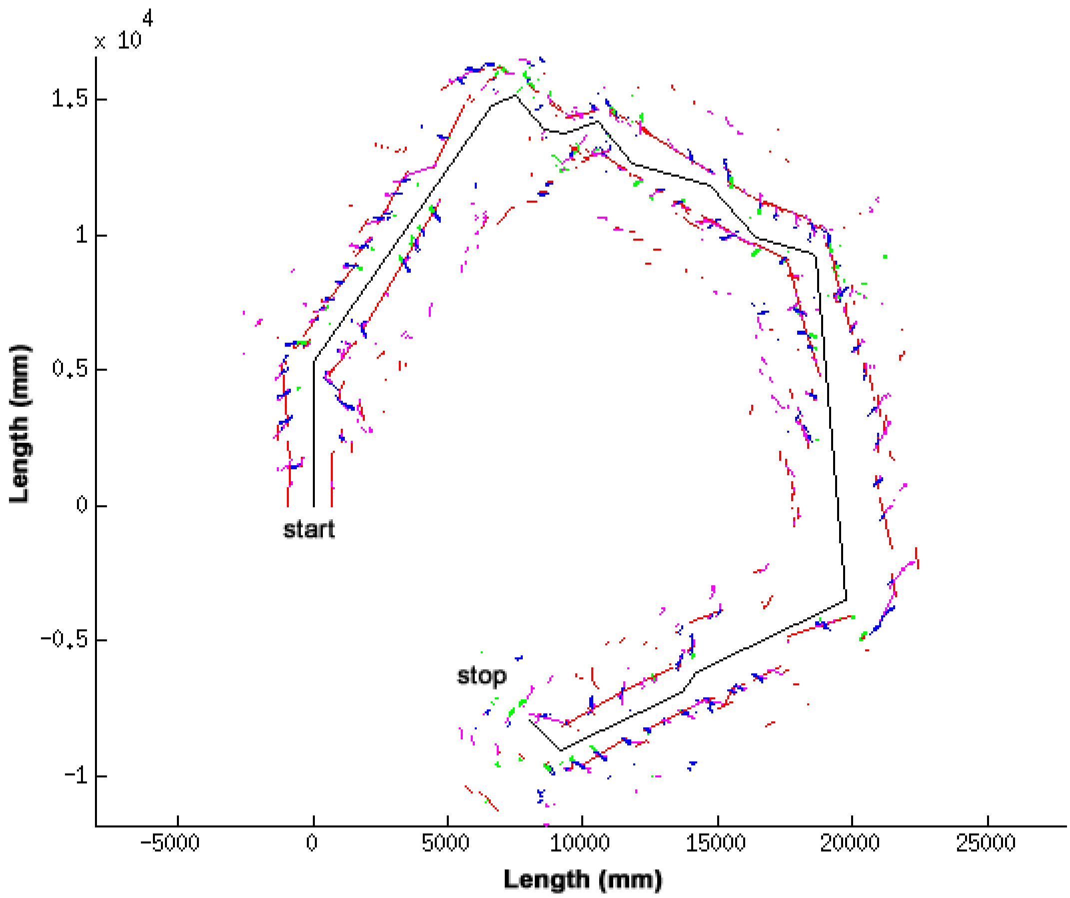


Figure 4.9 Surfaces formed from the sonar readings collected from each sensor. The colors of the surfaces correspond to the sensor that collected the data. Red: Sensors 0 and 7. Magenta: Sensors 1 and 6. Blue: Sensors 2 and 5. Green: Sensors 3 and 4. The black line represents the robot's journey.

Applying the above algorithm, we get four sets of surfaces on both sides of the path. Surfaces on the left side are generated from sensors 0, 1, 2, and 3. Surfaces on the right side are generated from sensors 4, 5, 6, and 7. These surfaces need to be merged into a single set on each side, representing the final surfaces perceived by the robot. To merge them, it is observed that for sonar sensors, readings indicating the presence of an object nearby are more reliable than readings indicating the presence of an object far away. Henceforth the merging algorithm simply samples a point from the nearest surface on each side and for each discrete step along the path. By reapplying the above line-forming algorithm to these points collected, the points are then turned into the final set of surfaces as seen by the robot for that path.

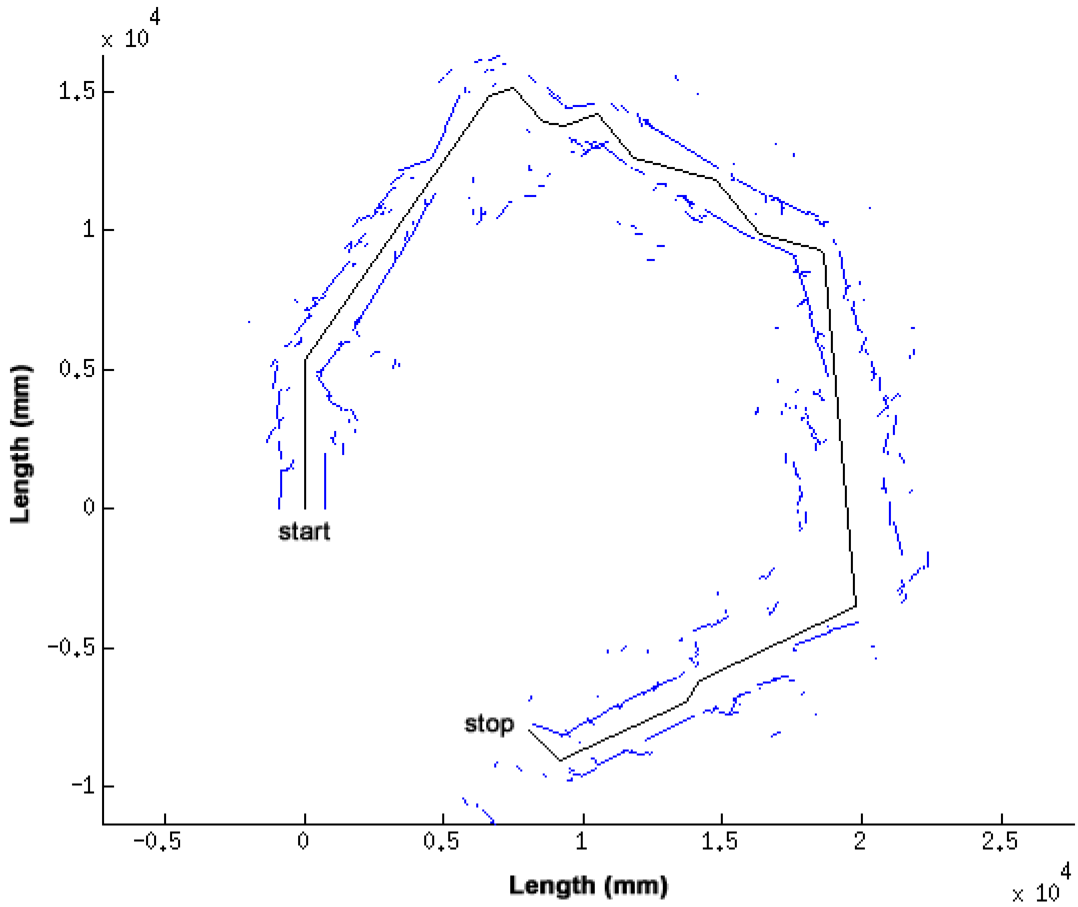


Figure 4.10 Surfaces (in blue) of the environment explored after information from all sensors have been merged. The black line represents the robot's journey.

The advantage of the above method is that it can be applied to a variable number of sensors. By adding more sensors, one would obviously obtain a more complete picture of the environment. The approach can also be used to test the reliability of particular sensors by using only those sensors to sense the environment. Figure 4.10 shows the overall representation of the environment after surfaces from all the sonar sensors have been combined together.

4.3 Computing ASRs of Paths

According to Yeap's (1988) theory of cognitive mapping, the significance of computing an ASR is the immediate identification of a region of space that constitutes one's local environment. The perceiver feels it is bounded and identifies exits out of the bounded region (see Yeap & Jefferies (1999)). If the robot has more powerful sensors such as vision, an initial ASR would be computed from a single view of the environment and will subsequently be modified or updated with information from subsequent views (see Jefferies, Cree, Mayo, & Baker (2004); Jefferies, Weng & Baker (2008); Jefferies, Weng, Baker & Mayo (2004) and Jefferies, Yeap, Cosgrove, & Baker (2005)). However, since the robot has only sonar sensors, the information obtained from a single view is not rich enough for computing an ASR. In this sense, my robot could be considered as “partially blind”. Consequently, a new algorithm is developed for computing ASR for each path traversed rather than from each sonar reading of the environment.

After traversing each path, the robot would have obtained many surfaces of various lengths on both sides of the path. To form an ASR, the robot needs to select the appropriate surfaces on both sides of the path to form a boundary description for the ASR. The new algorithm first considers, independently for each side, all the larger surfaces “in view” to be boundary surfaces. If these surfaces together form a significant boundary, say exceeding 70% of the distance traveled (path length), then these surfaces adequately represent the boundary of the ASR and the smaller surfaces perceived will be ignored. If not, the algorithm iteratively uses smaller surfaces to form the boundary, until a significant proportion of the boundary is formed.

The reasons for choosing the larger surfaces first are twofold. Firstly, larger surfaces are more likely to correspond to some real surfaces in the physical world and, if they exist, they are also more likely to act as a boundary for the current local space. Secondly, and again, what is needed is only a rough description of the shape of the local environment. Hence, once we have established this rough description using the larger surfaces, the algorithm can be terminated.

The algorithm is described in detail below. It consists of two steps (the threshold values used are intuitively chosen):

Step 1: Selecting boundary surfaces – For each side of the path do (four possible iterations):

a) Select all perceived surfaces that are greater than 700mm in length. If the sum of the surfaces selected is greater than 70% of the distance traveled, then go to Step 2.
b) Select all perceived surfaces that are greater than 500mm in length. If the sum of the surfaces selected is greater than 70% of the distance traveled, then go to Step 2.
c) Select all perceived surfaces that are greater than 300mm in length. If the sum of the surfaces selected is greater than 70% of the distance traveled, then go to Step 2.
d) Select all perceived surfaces that are greater than 200mm in length (the surfaces that are less than 200mm are considered as too small for consideration). Go to Step 2.

Step 2: Given the surfaces from Step 1, compute the boundary for the ASR:

a) For each side, determine the Euclidean distance of the gap between neighboring surfaces.
b) For all gaps greater than the 500mm threshold, leave the gap as it is.
c) For all gaps less than the 500mm threshold, replace it with a virtual surface. That is, a line is added from the end point of the first surface to the start point of the second surface.
d) For all virtual surfaces, calculate and compare the gradients of the virtual surface and the first surface it is connected to. If the gradients do not differ significantly, then it is not necessary to keep two surfaces. The two surfaces are replaced with a single surface by simply connecting the start point of the first surface with the end point of the second surface. If the gradients differ significantly, the virtual surface will be considered as part of the boundary for the ASR.

The threshold that signifies a significant gap was set at 500mm because the robot would not be able to safely move through a gap that is narrower than 500mm. In most cases, the gaps less than 500mm exist due to the sonar sensor's inconsistencies and should nonetheless be treated as if a surface is present.

The progression of the boundary selection process of Step 1 is applied to the surfaces shown in Figure 4.10. The results after each of the four iterations are shown in Figure 4.11. It clearly illustrates those paths that contain large surfaces do not need to retain smaller surfaces. However, when larger surfaces are absent, smaller surfaces are relied upon to provide spatial description of the surroundings.

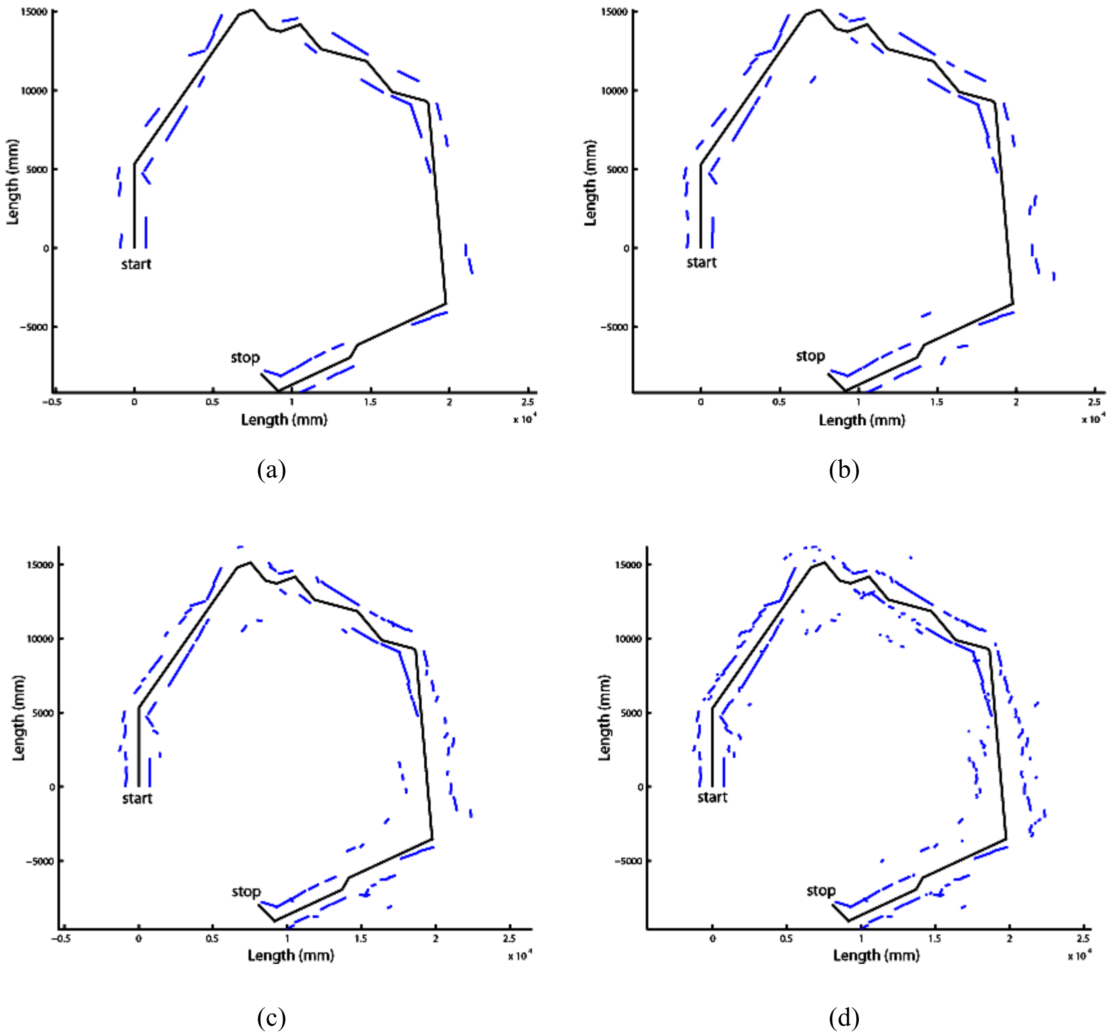


Figure 4.11 Progression of the boundary selection process from the first Iteration (a) to the last iteration (d). Smaller surfaces are not used if large surfaces are sufficient for computing the boundary.

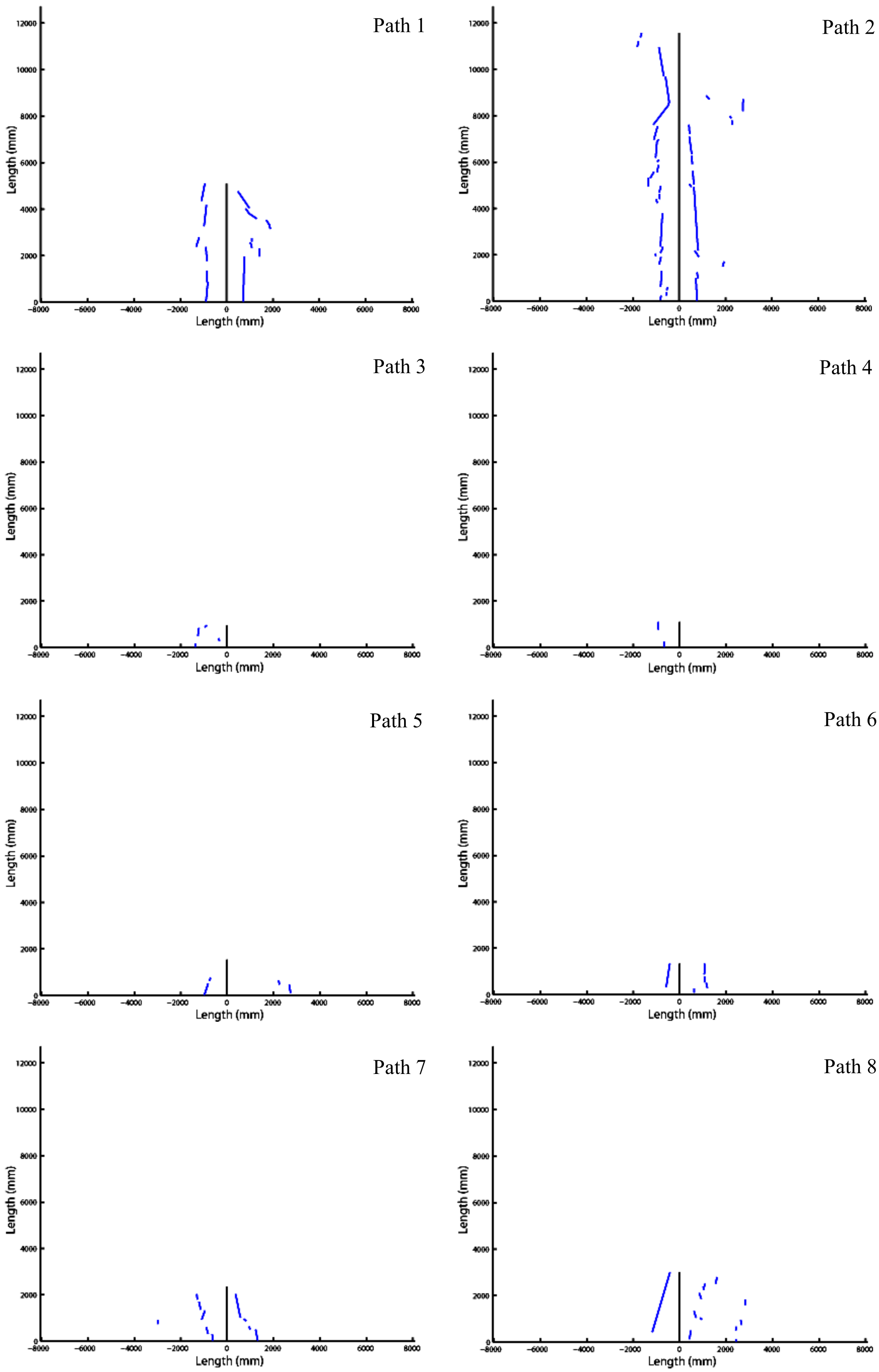


Figure 4.12 Individual plots of the surfaces selected for paths 1 to 8

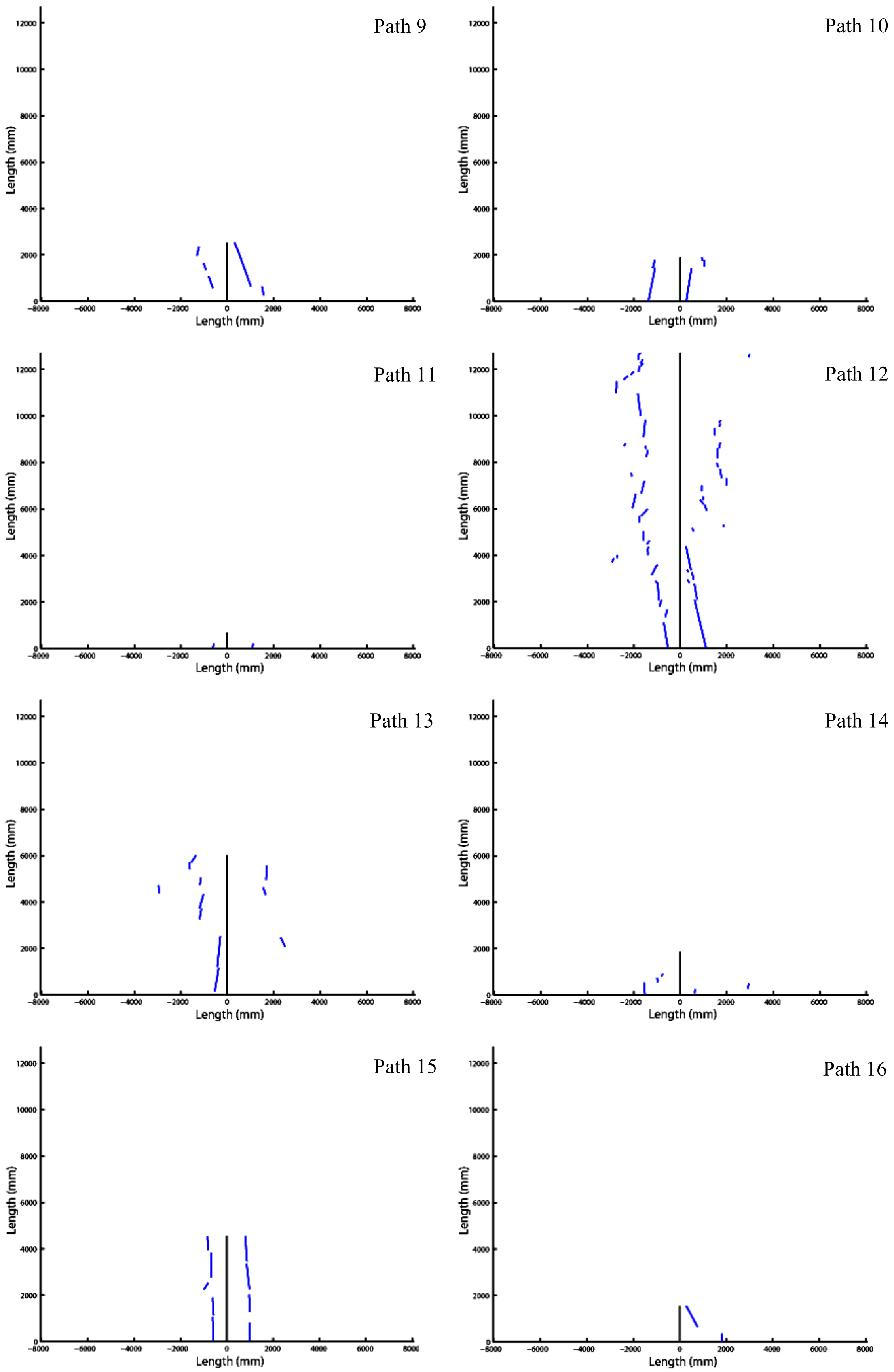


Figure 4.13 Individual plots of the surfaces selected for paths 9 to 16

The actual surfaces selected for each path for computing an ASR are shown in Figures 4.12 and 4.13 in their own local coordinate system. The final boundary computed after applying Step 2 is shown in Figure 4.14. This can now be used to compute the ASRs for the environment as a whole.

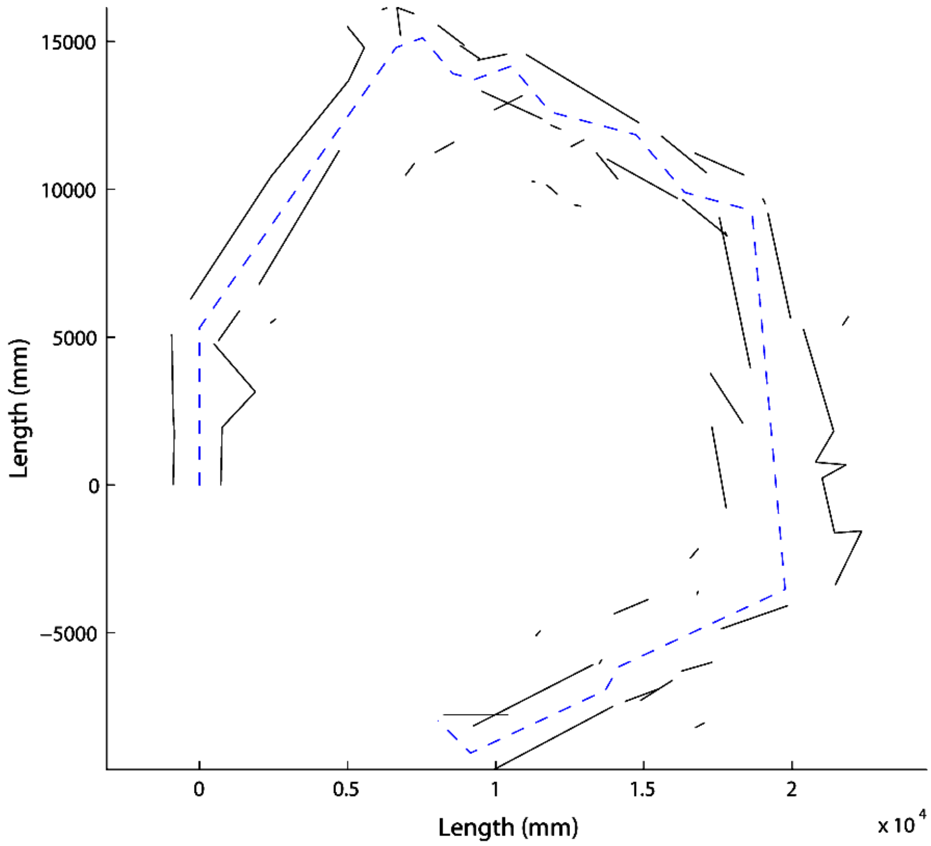


Figure 4.14 The final boundary computed for each path which is also the ASR for each path. The result is then used to compute the final ASRs for the environment as a whole (see next section).

4.4 Computing ASRs of the Environment

As noted above, once ASRs for each path are computed, one needs to combine or split ASRs to better reflect the shape of each local environment experienced. Such combining or splitting will take into consideration of other spatial characteristics of ASRs. For example, if the robot moves down a straight corridor in a zigzag manner, the ASRs for these paths are combined as a single ASR representing the corridor.

Similarly, if the robot moves out of a corridor into a junction and straight into another corridor, the single ASR computed for the path is split into three different ASRs.

The basis of the ASR generating algorithm for this final step is the well-known Split and Merge method, which originated from research in pattern recognition (see Duda & Hart (1973); Niemann (1990) and Pavlidis & Horowitz (1974)). A classic application of this algorithm is finding piecewise linear approximations of contour points that have been detected in an image. A variety of other applications has been proposed, including segmentation of image regions given a homogeneity criterion, e.g., with respect to color or texture (see Niemann (1990)). Additionally, Split and Merge has also been popular in robotics for generating geometric maps from range data by extracting lines (for example, Baltzakis & Trahanias (2002); Borges & Aldon (2000) and Newman, Leonard, Tardds, & Neira (2002)). However, it has rarely been utilized before for generating topological information from a metric map, as described here. The Split and Merge algorithm forms the core part of the process resulting in a topological ASR representation of the environment.

The objective here is to transform the initial network of ASRs based upon path movements of the robot into a network of ASRs based upon the spatial characteristics of the environment experienced. Splitting is done along the robot movement path, using an objective function that computes the quality of a region, based on local metric features derived from the geometric map, such as the average room width (corridors are long and narrow compared to rooms) and overall direction (e.g., a corridor is separated from another one by a sharp bend in the wall).

The step by step operation of the original Split and Merge algorithm described by Niemann (1990) are as follows:

1. Start with an initial set R^0 consisting of n_0 regions $R_0^0, \ldots, R_{n_0-1}^0$.
2. Split R_i^k into two parts: R_j^{k+1} and R_{j+1}^{k+1} if $h\left(R_j^{k+1}, R_{j+1}^{k+1}\right) \geq \theta$. Repeat as long as $h\left(R_j^{k+1}, R_{j+1}^{k+1}\right) \geq \theta$ for any $j = 0, \cdots, n_{k+1} - 2$

3. Merge two adjacent regions R_i^k and R_{i+1}^k into one new region R_j^{k+1} if $h\left(R_i^k, R_{i+1}^k\right) < \theta$. Repeat until merging is not possible any more.
4. Shift the split point shared by two adjacent regions R_i^k and R_{i+1}^k to left and right while leaving the overall number of parts fixed. Keep the split that reduces the overall error, repeat until no further changes occur.

Split and Merge is a recursive algorithm. It starts with an initial partitioning of the map, and then refines it until a stable configuration has been reached. The algorithm results in a piecewise approximation of the original points, where every single residual error is below a given threshold θ.

To apply Split and Merge, the following initialization steps must be completed first:

1. The input data, consisting of contour points, which is to be approximated, is sorted.
2. Choose a parametric function F (for example, lines, curves, etc.) to be used for approximating the contour points.
3. Choose a method for computing the residual error ε of the resulting approximation (usually root mean square error), or, when used for regions, a homogeneity or quality criterion,
4. Set the threshold θ which determines whether a split or merge is required.

Split and Merge is applied twice to generate the final ASR representation: the first, to generate initial partitioning of the map; and the second, to refine the initial split points to give a better approximation of the regions.

4.4.1 Initial Split and Merge

Before the region Split and Merge algorithm on the map can be applied, it is necessary to create an initial split of the map. This can be generated using a number of different approaches. However, the robot's map in this case is already split into different ASRs where each split is based upon a straight line path. Consequently, the start and end points of each path provide the necessary initial split points for the algorithm. As such the initial Split and Merge algorithm for ASRs excludes the split part of the algorithm. What needs to be done is to see whether the initial ASRs need to be merged together. One instance where it is needed is when the robot moves through a corridor in a zigzag fashion. Such paths through a corridor create several ASRs which should be merged into one. The merging algorithm for ASRs is described below.

Merging: For all ASR_1 to ASR_n, do:

1. Take the first two ASRs (say, ASR_1 and ASR_2) with their respective split points (e.g. SP_1, SP_2 and SP_3).
2. Connect SP_1 and SP_3 using a straight line (call this line SP1←→SP3).
3. Sub-divide the two paths between SP_1 and SP_3 into equidistant points (currently set to 500mm apart), starting from SP_1.
4. For each of the points, calculate the distance of the norm from the division point to the line SP1←→SP3.
5. If any of the distances (and hence the residual error) is above a threshold, then repeat the process with ASR_2 and the rest of the ASRs. In other words, no merging of ASR_2 with ASR_1.
6. If all of the distances (and hence the residual error) are below the threshold, then the split point SP_2 is removed and ASR_1 and ASR_2 becomes a single ASR, ASR_1. Repeat the process.

Note that the original Split and Merge algorithm specifies a shifting step after the merging phase. The objective of this is to fine tune the split points. However, this level of "accuracy" is not needed since the ASRs are only an approximate representation of the environment. For this reason, the shifting process was not implemented as part of the algorithm for computing the ASRs.

4.4.2 Region Split and Merge

The initial Split and Merge will group the ASRs that have the same orientation. These ASRs are caused by zigzag movements along a confined space, say, a corridor. However, the above does not take into consideration any spatial characteristics of ASRs. To do so, the Split and Merge algorithm is repeated using a different residual error function $h(ASR_i, ASR_j)$. This function compares two ASRs, ASR_i and ASR_j, and computes the homogeneity of the two regions (low values of $h(ASR_i, ASR_j)$ mean homogeneous, high values very inhomogeneous).

The residual error function is used during the splitting phase for deciding whether an ASR_i^k will be split at a given position into two new regions ASR_j^{k+1} and ASR_{j+1}^{k+1}, and in the merge phase to determine whether two adjacent ASRs can be joined together. When the homogeneity is above a given threshold θ_r, the region will be split into two ASRs. Conversely, two neighboring regions are merged when the homogeneity is below the threshold.

One of the criteria to distinguish one local region from another is through spatial characteristics. Based on this concept, the basic idea is to use the average width of a region in the map as a criterion for splitting, as a width change resembles a changing environment, e.g., a transition from a corridor to a big room. The homogeneity (residual) function used is:

$$h(ASR_i, ASR_j) = \frac{\max\left\{ fw(ASR_i),\ \ fw(ASR_j) \right\}}{\min\left\{ fw(ASR_i),\ \ fw(ASR_j) \right\}} + s_r r\left(ASR_i, ASR_j\right) \qquad \textit{Equation } 4.1$$

where $fw(ASR_i)$ is the average width of region ASR_i, and $r\left(ASR_i, ASR_j\right)$ is a regularization term that takes care of additional constraints during splitting. The factor s_r controls the influence of $r\left(ASR_i, ASR_j\right)$.

Obviously, the average width is given by $fw(ASR_i) = \frac{A_{R_i}}{l_{R_i}}$, where A_{R_i} is the area of ASR_i, and l_{R_i} is its length. The definition of the length of an ASR in particular is not always obvious, but can be handled using the robot movement paths, which are part of each ASR. The length l_{R_i} is then defined by the length of the line connecting the start point of the first robot path of an ASR and the end point of the last path of the ASR, i.e., the line connecting the exits of an ASR, which the robot used while traveling through the environment. This is an approximation of a region's length having the advantage that disturbance caused by zigzag movement of the robot during mapping does not affect the end result.

For area computation, the gaps contained in the map have to be taken into account, either by closing all gaps, or by using a fixed maximum distance for gaps. Closing a gap is a good approach if it originated from missing sensor data, but may distort the splitting result when the gap is an actual part of the environment. Closing it would make the ASR appear smaller than it actually is. The implementation uses a combination of methods: small gaps are closed in a pre-processing step; and large ones are treated as distant surfaces.

Depending on how gaps are handled, the algorithm possibly creates a large number of very small ASRs. This is where the regularization term $r(ASR_i, ASR_j)$ comes in: it ensures that ASRs do not get too small. It penalizes small ASRs but still allows their creation if the overall quality is very good. A sigmoid function centered at n is used, where n is the desired minimum size of a region:

$$r(ASR_i, ASR_j) = \frac{1}{1 + \exp\left(-\frac{\min\left\{ A_{R_i}, \ A_{R_j} \right\}}{A_{\max}} + n \right)} - 1 \qquad \textit{Equation } 4.2$$

This function can assume values between −1 and 0. The exponent is basically the ratio of the area of the smaller one of two adjacent ASRs to the maximum area $A_{\max}$ of the smallest possible ASR that the algorithm is still allowed to create. Thus, the smallest ratio is 1. It increases when the region gets larger. Figure 4.15 shows a graph which roughly approximates the regularization function.

The regularization term only has an influence when an ASR is already small, by reducing the likelihood the region will be split again. As the sigmoid reaches zero asymptotically, it has virtually no influence when a region is large. The overall influence of the regularization can be controlled by the factor s_r in Equation 4.1. It is given by $s_r = s\theta_r$, where $0 \le s \le 1$ is set manually and defines the percentage of the threshold θ_r that is to be used as a weight. θ_r is the threshold introduced earlier, which determines that a region is to be split into two when the first region is θ_r times larger than the second one.

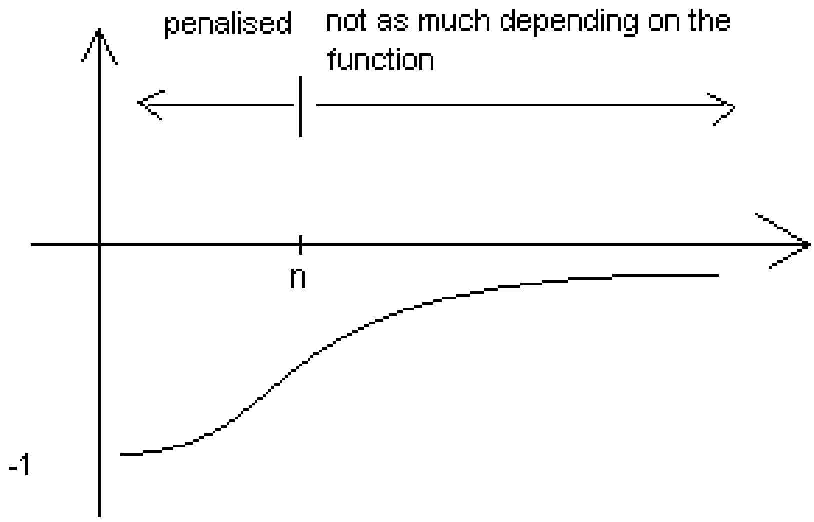


Figure 4.15 Graph showing the characteristics of the regularization function

The following is a summary of the region Split and Merge algorithm:

Step 1: Splitting

1. Take an ASR (e.g. ASR_1) with its respective split points (e.g. SP_1 and SP_2)
2. At each of the division points along the path of the robot, have a temporary split. (Let us denote the region to the left of the temporary split as ASR_{1L} and the region to the right as ASR_{1R}.)
3. Calculate the areas of ASR_{1L} and ASR_{1R}
4. Calculate the lengths from the regions split points (SP_1 and SP_2) to the current temporary split point
5. Calculate the average widths for ASR_{1L} and ASR_{1R}
6. Calculate the homogeneity function *h(ASR_{1L} and ASR_{1R})*
7. If *h(ASR_{1L} and ASR_{1R})* is LESS THAN the threshold θ_r then proceed with next division point and repeat steps 1 – 7.

8. If $h(ASR_{1L}$ *and* $ASR_{1R})$ is MORE THAN the threshold θ_r then this confirms that the current split point as an actual split point.
9. The splitting procedure continues with the newly created region and all other remaining regions.

Step 2: Merging

1. Take three regions (e.g. ASR_1, ASR_2 and ASR_3) with their respective split points (e.g. SP_1 SP_2, SP_3 and SP_4)
2. Temporarily "remove" the split point between ASR_1 and ASR_2 (i.e. SP_2) to temporarily merge the two ASRs (call it $Temp_R$)
3. Calculate the areas of $Temp_R$ and ASR_3
4. Calculate the lengths of both $Temp_R$ and ASR_3
5. Calculate the average widths of both regions
6. Calculate the homogeneity function $h(Temp_R, ASR_3)$
7. If $h(Temp_R, ASR_3)$ is above the threshold θ_r, then the split point remains and the regions remain as ASR_1, ASR_2 and ASR_3. The process continues with the remaining regions.
8. If $h(Temp_R, ASR_3)$ is below the threshold θ_r, then permanently remove split point SP_2 to merge regions ASR_1 and ASR_2 together. The splitting process continues with the newly created region, with the remaining regions.

For the same reason mentioned earlier, the shifting step of the Split and Merge algorithm was not implemented for the region Split and Merge. The result of applying this algorithm onto the surfaces in Figure 4.14 is shown in Figure 4.16. The black dots mark the split points whilst the regions between the split points represent the ASRs.

The split points would be generated at different locations by varying the two parameters θr and s. Their influences are shown in the following figures. Figures 4.17 and 4.18 depict splitting results for different values of θr whilst Figures 4.19 and 4.20 show the results for different values of s.

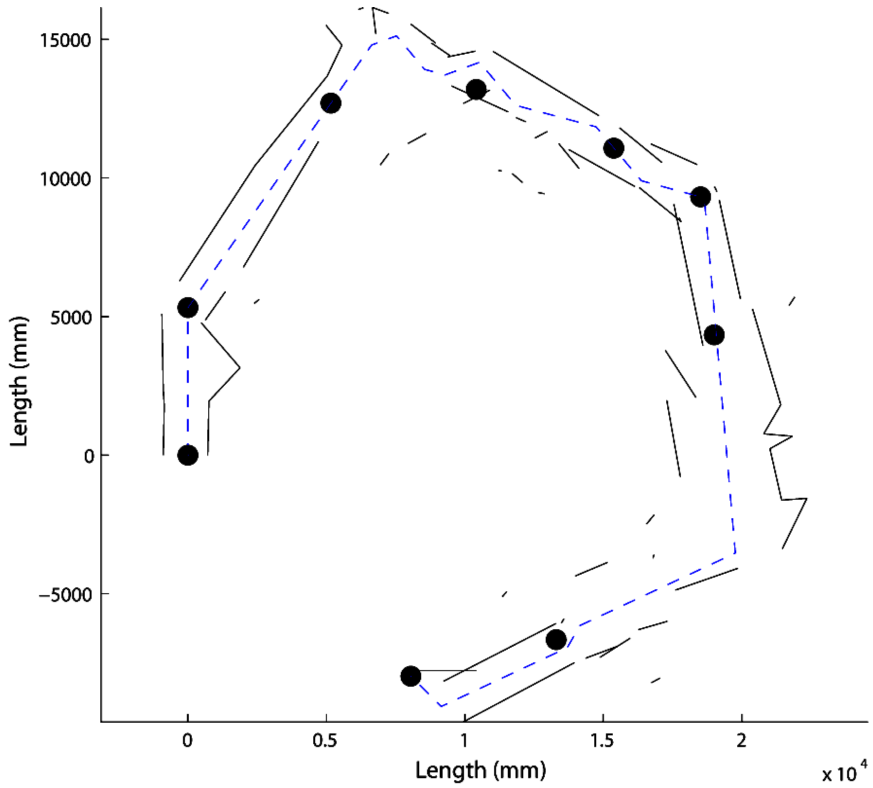


Figure 4.16 ASRs generated from the boundaries shown in Figure 4.14 using θr = 1.7 and s = 0.1. The black dots represent the split points, which divide the environment into ASRs.

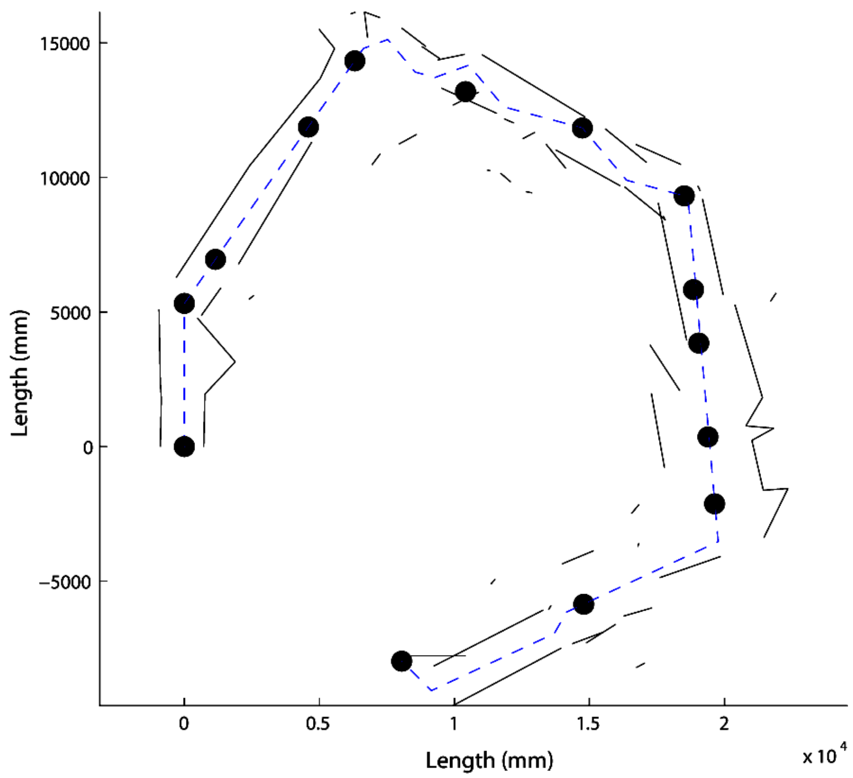


Figure 4.17 ASRs computed using parameter values θr = 1.3 and s = 0.1

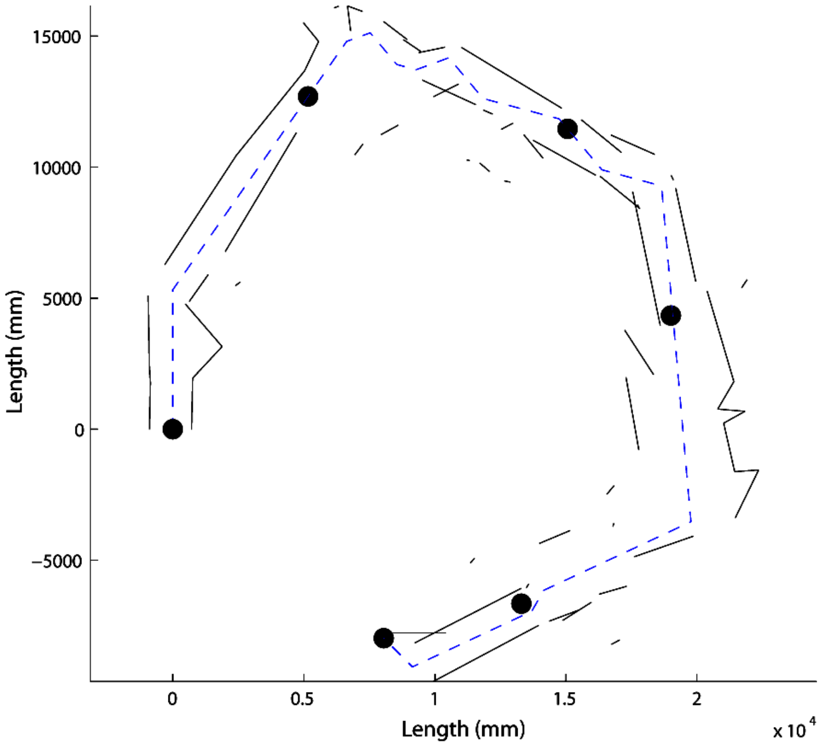


Figure 4.18 ASRs computed using parameter values θr = 2.2 and s = 0.1

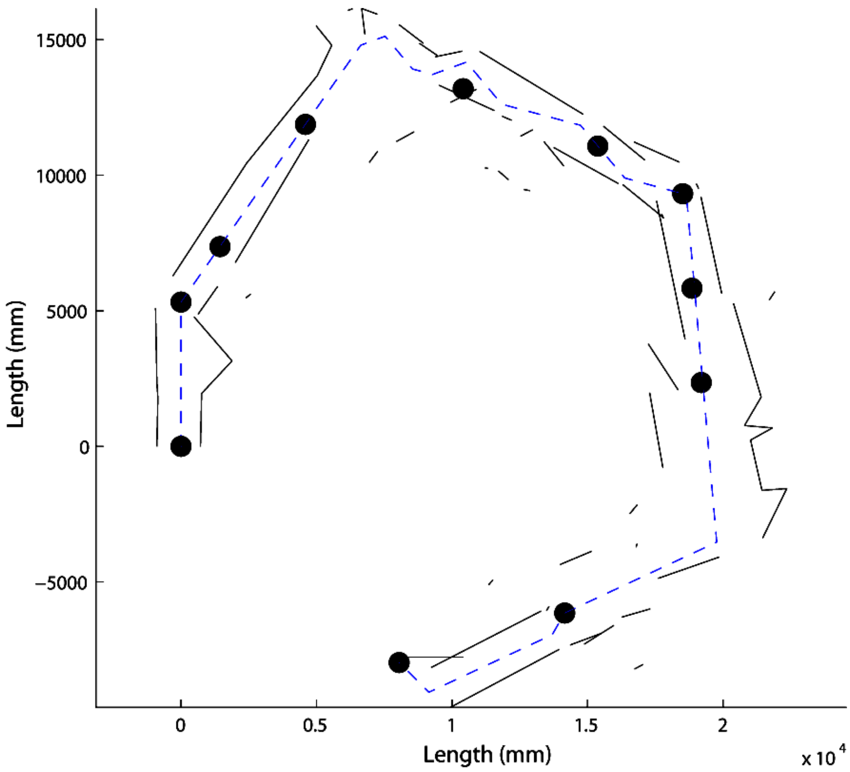


Figure 4.19 ASRs computed using parameter values θr = 1.7 and s = 0.005

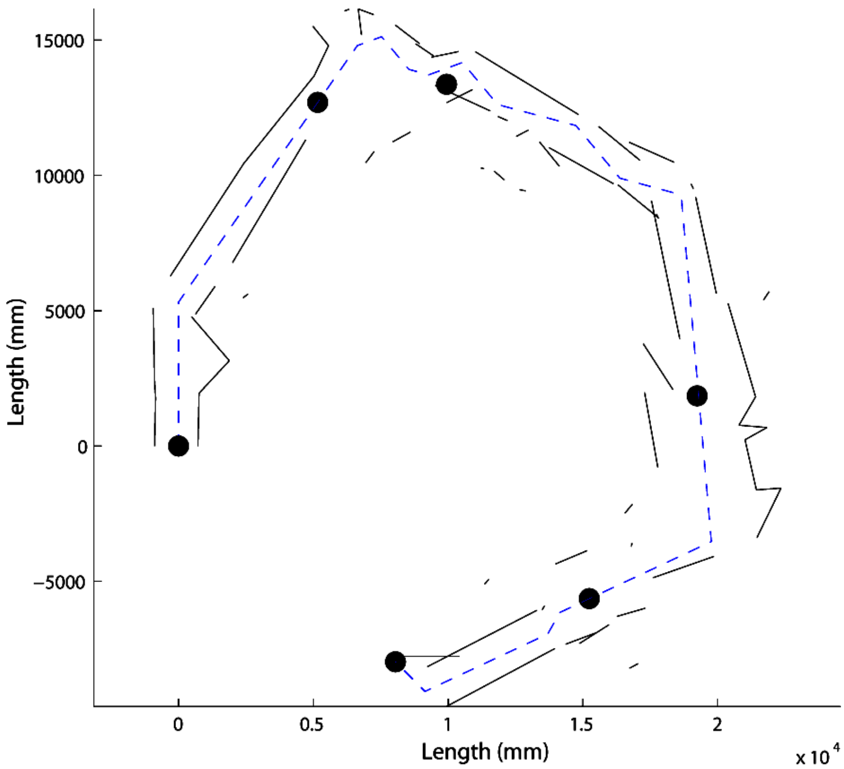
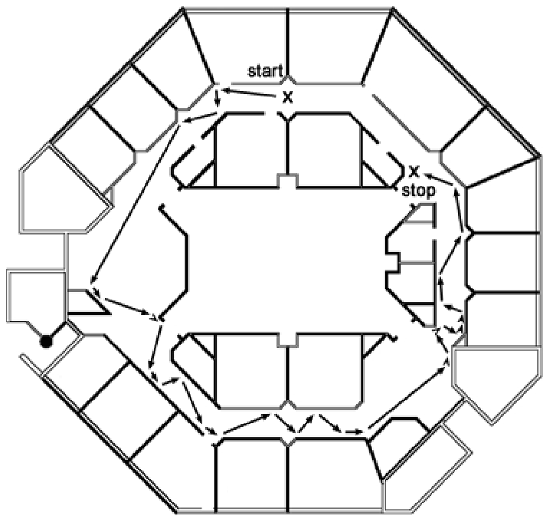


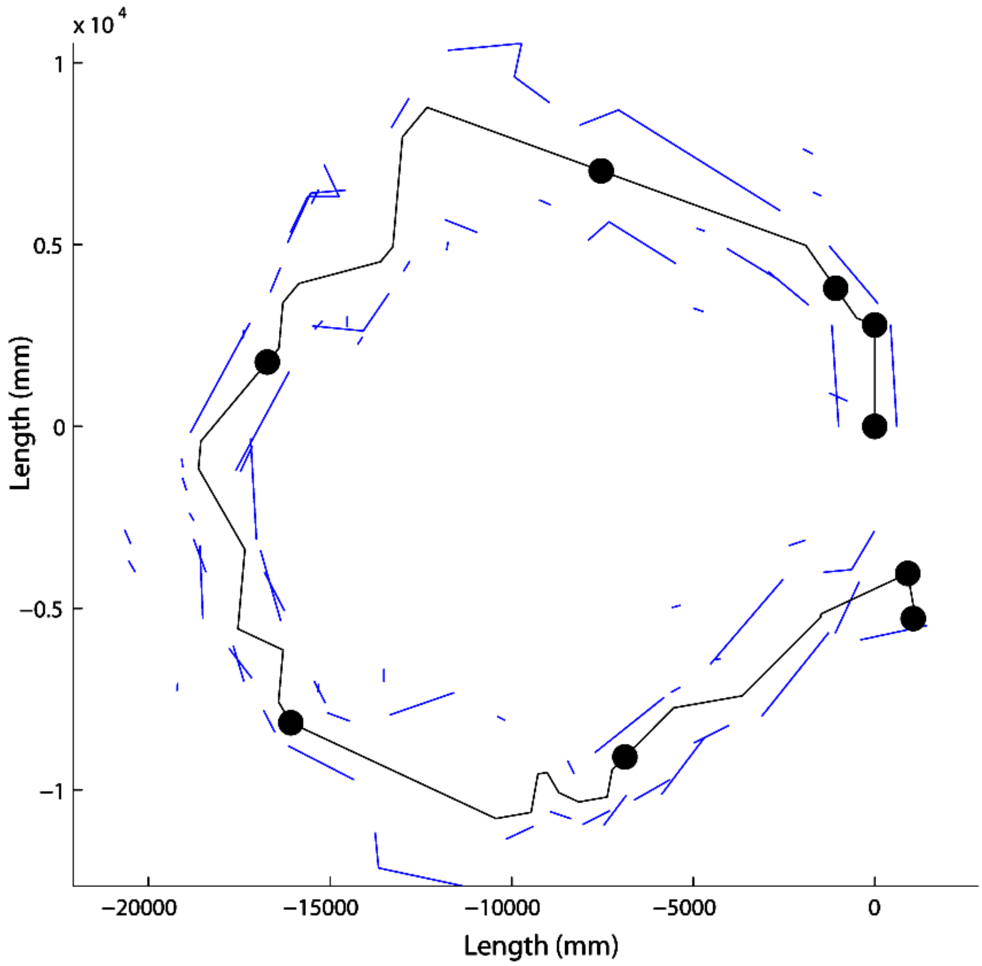
Figure 4.20 ASRs computed using parameter values θr = 1.7 and s = 0.95.

The maps in Figures 4.17 and 4.18, were computed using θ_r values of 1.3 and 2.2 respectively, while the other parameter was fixed to s = 0.1. In Figures 4.19 and 4.20, the maps were generated using s values of 0.005 and 0.95 respectively, while θ_r was fixed to 1.7. Since θ_r controls the ratio of the average width of adjacent ASRs, increasing this parameter results in fewer splitting points and therefore larger ASRs. The parameter s shows a similar behavior, i.e., the higher its value the higher the influence of the regularization term, which results in larger regions on average.

The following are more examples of the different exploration journeys and their corresponding ASR representations. Figure 4.21 presents the robot mapping the environment in the opposite direction compared to the journey presented in Figure 4.1. Figure 4.22 shows the robot successfully explore the large space situated in the middle of the section. Interestingly, the robot performed a couple of loops during its journey through the middle section and yet produces a single ASR representing that space. Finally, Figure 4.23 shows the robot's exploration through two sections of the building.

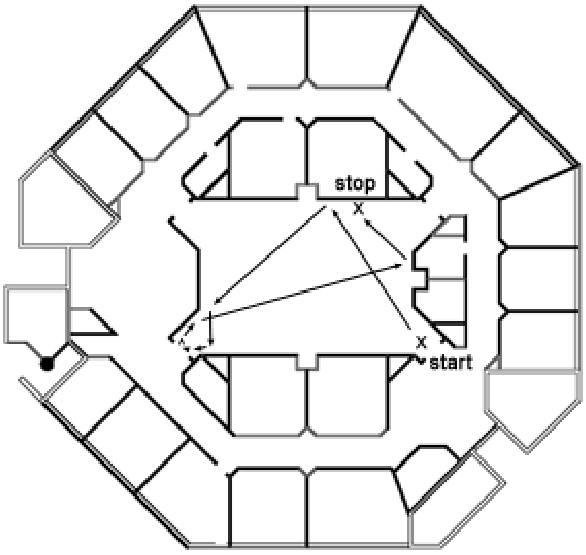


(a)

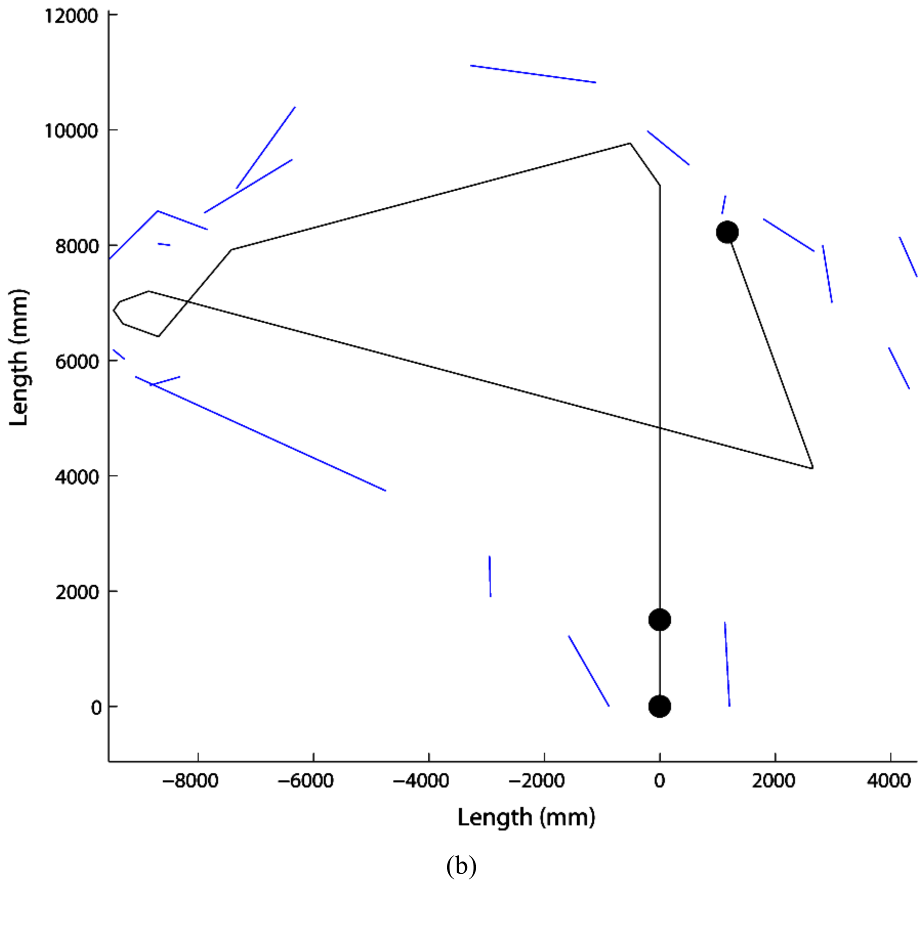


(b)

Figure 4.21 Exploration of the environment in the opposite direction to the journey in Figure 4.1. (a) Journey of the robot in the experiment and (b) the ASRs generated - (0,0) relates to the start location

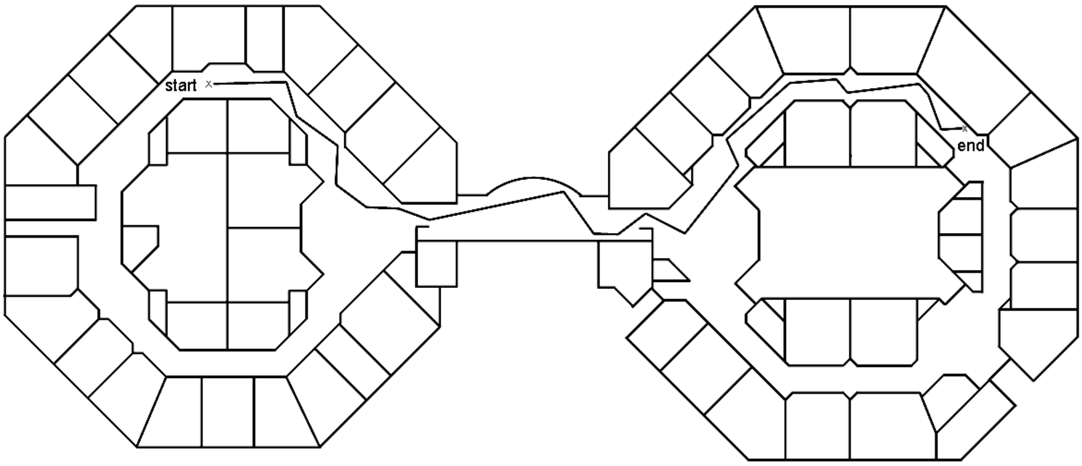


(a)

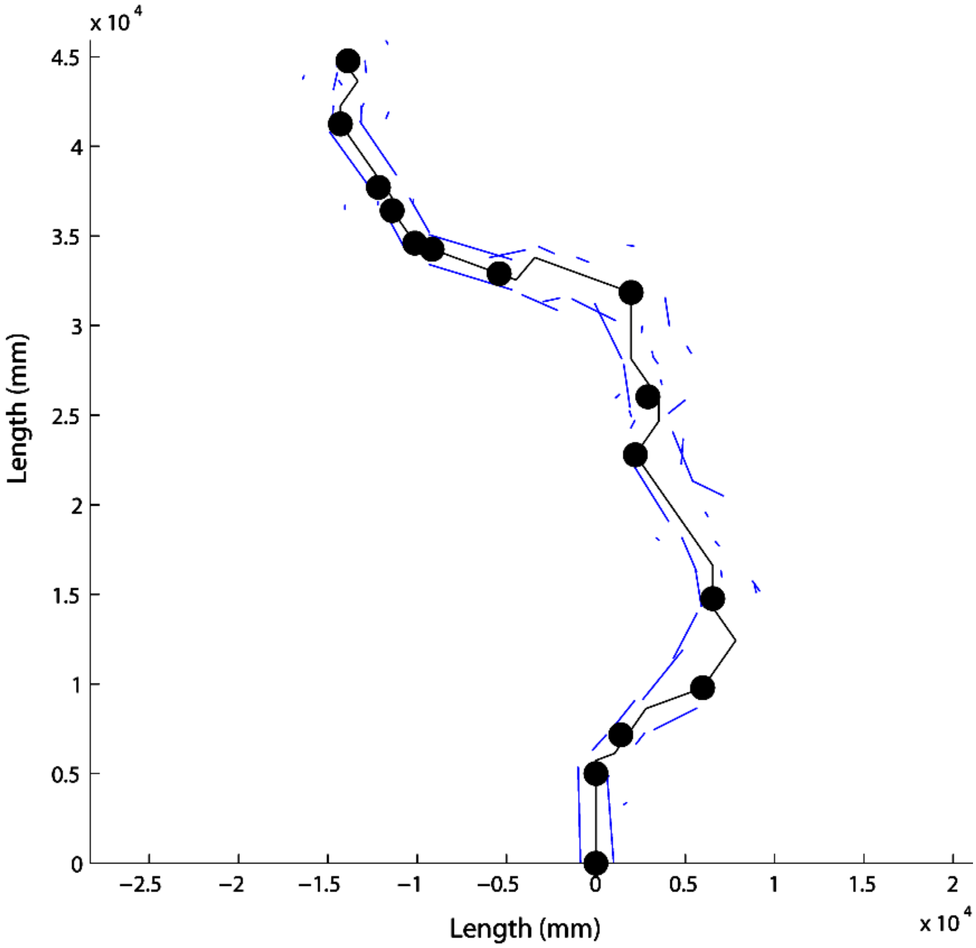


(b)

Figure 4.22 Exploration of the large space in the middle section of the floor (a) Physical movement of the robot took during the journey and (b) the ASRs generated - (0,0) corresponds to the start location



(a)



(b)

Figure 4.23 Exploration of an extended environment (a) The robot's journey and (b) the ASRs generated - (0,0) corresponds to the start location

The parameters used in the last three examples were identical to that used for generating the points in Figure 4.16. Their values are $\theta_r = 1.7$ and $s = 0.1$. These values are empirically determined. These results show that once the parameter values are set, the Split and Merge algorithm is successful for computing ASRs for different types of maps. No "tweaking" is found to be necessary, hence enabling the robot to be truly operating without human intervention.

4.5 Discussion

The basic algorithm for computing an ASR is the same as the algorithm reported in Jefferies (1999). Thus, there is no change to the theory of ASR computations. However, Jefferies tested her algorithm using a simulator whereas the algorithm developed here works for a robot equipped with sonar sensors. This is the first time that an algorithm is developed for computing ASRs for a robot equipped with sonar sensors.

With sonar sensors, the robot collects very little and possibly unreliable information about its environment from a single position in space. Consequently, the robot can only "sense" where it is (via the act of computing an ASR that it is in) after it has gone down a path through the environment. This posed two major new challenges:

1. How do you compute a boundary for each path traversed from the eight sonar sensors available to the robot?, and
2. How do you combine ASRs computed for each path into ASRs computed for the whole environment?

This chapter presented a solution to these problems using a multi-level representation of surfaces and a modified version of the Split and Merge algorithm developed for pattern recognition research. ASRs are identified using two criteria:

1. Similarity in the path traversed; and
2. Similarity in the bounded space.

The algorithm developed has been well tested by the robot making numerous explorations in an office-like environment. The algorithm did not fail to generate ASRs and, for most of the time, the ASRs generated were judged by humans to partition the environment well. The latter judgment is not important, though, as the view of the robot is different to that of the humans.

It is argued that the new algorithm is cognitively interesting because it has the following characteristics:

1. It generates ASR with approximate shape rather than precise co-ordinates;
2. The resulting ASRs computed for an environment are dependent on the direction of travel.

Both these properties are commonly observed characteristics of cognitive maps of cognitive agents. A strategy to use such a map to find its way home is presented in the next chapter.

Chapter Five

Returning Home: Algorithm, Experiments, and Results

With the successful implementation of an algorithm for computing ASRs for the robot, this chapter describes the design and implementation of several algorithms that will enable the robot to find its way home. As noted in Chapter 1, the ability to return home is a crucial task for foraging animals and in Chapter 3, it was noted that animals make significant use of distance and orientation information to find their way home. These two pieces of information are available implicitly in the network of ASRs computed. This chapter explores how such information could be extracted from a fuzzy map and used by the robot to find its way home.

The returning home experiment begins with the robot being positioned approximately where it stopped during the exploration journey. Like the exploration journey, no modifications were made to the environment in the homeward journey. That is, no changes were made to ensure that the environment be the same for both journeys. Quite often, the homeward journey is made days, weeks or even months after the original exploration journey. Thus the environment could have changed significantly. For example, doors were open during the initial exploration but were shut during the going home journey; and vice versa.

Like in exploration, the robot generates a "cognitive map" of its environment during the homeward journey. However, and as shown in the previous chapter, this map is likely to be different from the map computed during exploration. Furthermore, while it was sufficient during the exploration stage to collect all the data first and then process it at the end, the robot will now have to process the sensor information into ASRs every time it stops. This is because the robot now needs to decide whether it has reached home or not. If not, it needs to know where it is and where to move next. In short, during the homeward journey, the robot needs a navigation strategy and a method to localize itself.

To develop a suitable navigation strategy, recall that in chapter 3, ants find their way home by traveling towards the home direction and at an approximate distance away from it. When they are near their home, they search for it. Similarly, observe here that the robot knows which ASR it is in when it turns around to go home. It also knows how far it needs to travel from its current position in order to reach the end of this ASR and into the next. Thus a simple strategy would be to travel approximately the required distance to the exit and then begin to search for the exit whereabouts. However, the robot is unable to use its sensors to detect if it has reached the end of that ASR if it were to travel the required distance. Furthermore, when going home, it does not follow the wall and thus attempts to travel in a straight line to reach the end of the corridor. This strategy is not always useful anyway since the robot is also tested moving through a large empty room. The solution to this problem is that the robot needs to somehow localize itself whenever it reaches the end of an ASR.

At any particular moment in the homeward journey, two representations of the environment are available: A cognitive map generated during the exploration journey (henceforth referred to as CM-in-memory) and a cognitive map generated during the homeward journey but only up to the point where the robot stops (henceforth referred to as CM-current). Since the shape of the ASRs computed in either cognitive map are not accurate enough to match and localize oneself, a new algorithm is developed which makes use of the length of the ASRs computed and the relative orientation between adjacent ASRs.

If the robot, say, has computed the 3rd ASR in its homeward journey, it could calculate the distance it has traveled in terms of the sum of the length of the three ASRs computed. From that, it could calculate roughly where it is in the CM-in-memory and which ASR in the CM-in-memory that it is in. The robot could use a variety of information to increase its confidence in localizing itself. In the experiments conducted here, two pieces of information are used, namely length of ASR and relative orientation between adjacent ASRs.

Section 5.1 describes the navigation strategy used by the robot to find its way home. Section 5.2 describes the localization algorithm and section 5.3 describes the results of the returning home experiments conducted with the robot. Section 5.4 concludes this chapter with a summary of the results obtained.

5.1 Navigation Strategy for Returning Home

When the robot is trying to find its way home, it needs a navigation strategy, not random exploration. Initially, the robot knows where it is i.e. the last ASR computed in the CM-in-memory. Let us label the ASRs in CM-in-memory as ASR_1, ASR_2, … ASR_n. ASR_1 is the home ASR and ASR_n is the last ASR computed at the end of the exploration (i.e. the ASR that the robot is in, at the start of the going home journey). The robot also knows which exit to go to and how far it is from its current position. The navigation strategy used by the robot can be described as follows:

Navigation Strategy (d: estimated distance to exit):

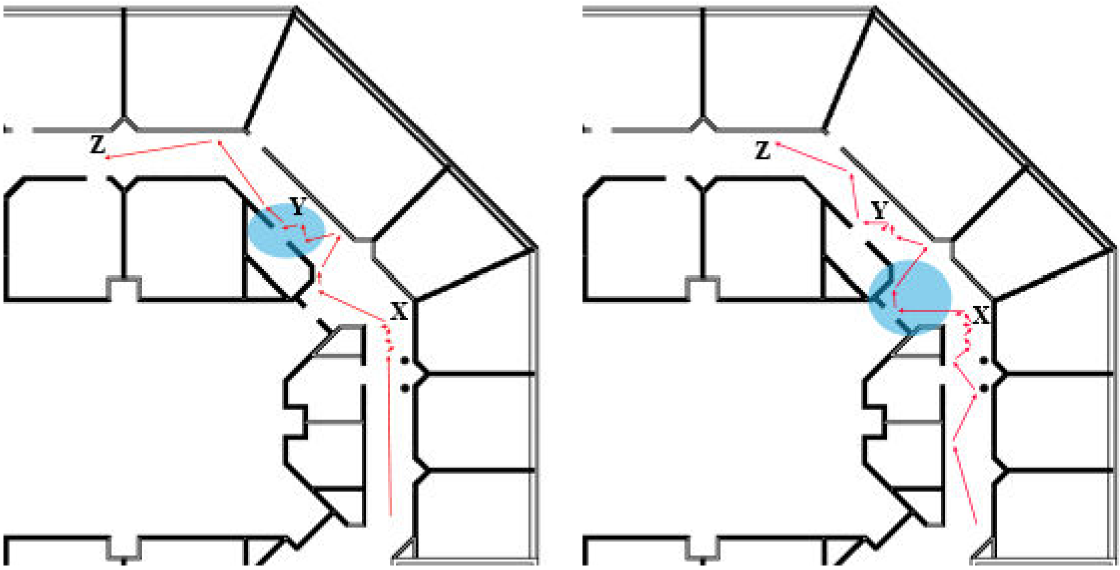
1. Turn towards the exit.
2. Move forward d-0.5 meter.
3. If an obstacle is encountered before having traversed d-0.5 meter, avoid the obstacle and continue to move forward the remaining distance calculated in steps 4 and 5.
4. Localize where the robot is using the algorithm described in Section 5.2. The localization algorithm works out which ASR the robot is most likely to be in and how far it has moved into that ASR.
5. If the robot is still in the current ASR, then work out how much more to travel to get into the next ASR. Set d to the required distance. Go to step 2.
6. If you have moved into the next ASR, check if this ASR is the home ASR. If so, proceed towards home and stop.
7. Calculate d, the distance from the current position to the exit of the current ASR. Go to step 1.

The above strategy bears some similarity to how some lower level cognitive agents (such as ants and bees) find their way home. Using distance and orientation information, they work their way close to home and when they are near their home, they carry out a search to find it. The above strategy gets the robot close to each exit and then localizes itself for the exit.

Note that an earlier version of the above algorithm was implemented and reported in Wong, Yeap, & Sapiyan (2005). The algorithm is as follows:

1. Move to the other end of the current ASR by traveling L-x distance. L is the perceived length of the ASR and x is a small distance so that the robot arrives at the decision point of the journey.
2. Once there, re-position by moving forward and backward to detect two side walls.
3. Move to a point which is equidistant from the two side walls.
4. Check whether the next ASR is on its left or right. If it is on the left (right), search for the open space on one's left (right).
5. Move into that open space and one has now moved into the next ASR.
6. Repeat the algorithm until all ASRs have been re-visited.

The earlier algorithm assumes that an ASR is always connected left or right of the current ASR. This constraint is found to be too restrictive and is derived from observing that the environment being tested at that time consists of corridors only. Furthermore, as noted above, moving a distance of L-x does not guarantee one reaches the end of the ASR computed during exploration stage. Unless the robot happens to orient straight towards the end of the corridor, the distance traversed could fall well short. Indeed, and as shown in Figure 5.1, that was the case and the robot failed to reach home. What is lacking in the earlier algorithm is the ability for the robot to localize itself in the homeward journey and this is now solved with the addition of step 4 in the new algorithm.

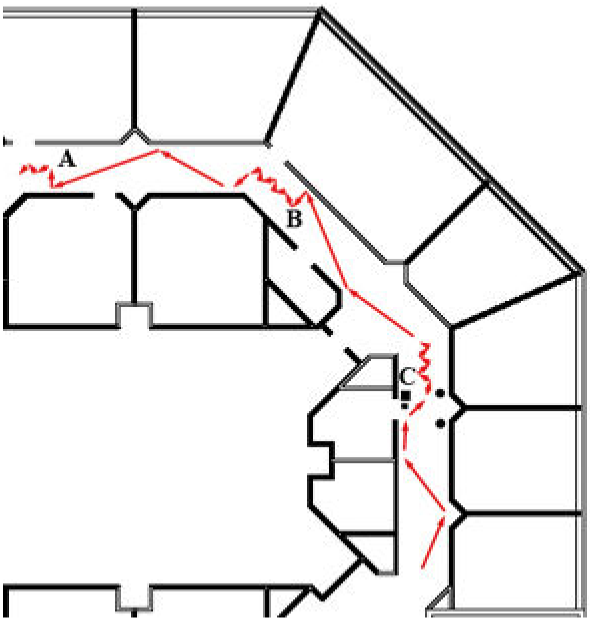


Unsuccessful journey 1 Unsuccessful journey 2

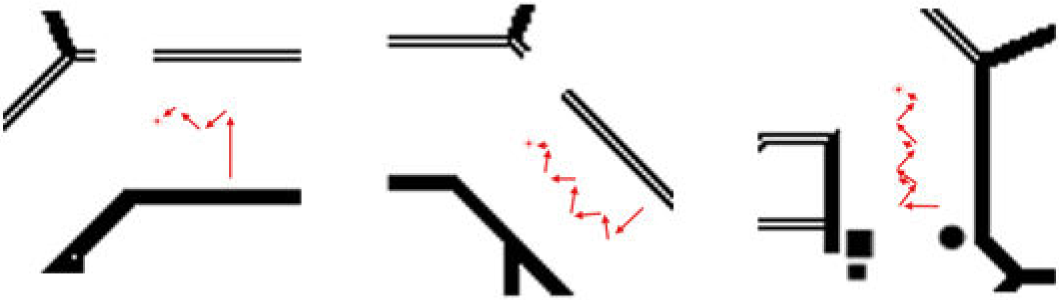
Figure 5.1 Unsuccessful attempts by the robot to get home using the previous algortihm: The shaded region is the area where the robot was confused. The robot failed to go straight to the end of the corridor and thus mis-localized itself.

It is interesting to note that both versions of the algorithm produce interesting observable robot behavior at the exits. Figure 5.2 shows in detail the robot's movements at some of the decision points. It shows that the robot was cautious and was looking for a gap on its left.

Figure 5.3 shows the robot's cautions when crossing the exit of an ASR using the latest algorithm developed. The robot was free to wander from the start up until location 1, which is where the robot thinks it is about 0.5m away from the exit. At that location, it calculates to see if it is still within the old ASR or in the next ASR. The former is true and so it moves forward to location 2, believing it has crossed the exit. It has not and it re-calculates how far to move forward again so as that it could cross into the ASR. At location 3, it found itself in the next ASR and it rotates towards the blue arrow (direction of the next split point). However, it senses an obstacle and it rotates away from it.



(a)



A B C

(b)

Figure 5.2 (a) Example of a successful going home journey and (b) The robot's cautious movements at decision points A, B and C are shown in detail. Each arrow indicates that the robot made a turn to inspect the environment. A longer arrow also indicates that it has moved forward too.

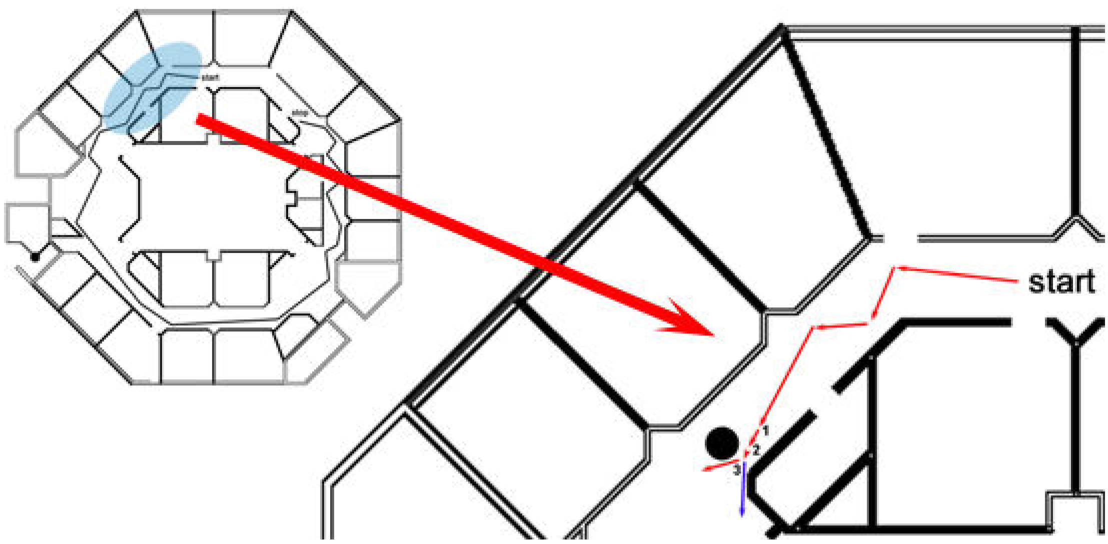


Figure 5.3 Detailed movement of the robot going home through the first ASR. The black dot represents the split point or exit of the first ASR.

5.2 Localization Strategies on the Way Home

The robot could localize itself using a variety of information and in this section, a new algorithm is developed which uses two key pieces of information implicitly available in the network of ASRs namely, distance and orientation. Sections 5.2.1 and 5.2.2 describe the individual localization strategy using distance and orientation information respectively. Section 5.2.3 describes the fusion of the two localization results to tell the robot where it is. Fusion is done using the Democratic Integration technique which was originally proposed by Triesch (1999) and Triesch & von der Malsburg (2001) for sensor data fusion in computer vision, using images as input data. The use of this technique allows integration of other information when they become available later.

5.2.1 Localization Strategy – Distance

Just as humans have a rough notion of how far they have walked starting from a certain location, so could a robot. However, if one were to measure it based upon the actual distance traveled as provided by the odometer, then the information collected would be difficult to use. This is because the robot often moves in a zigzag fashion across the environment and because of the accumulated errors in comparing such measurements would be excessive.

In the representation shown in Figure 4.16, the start and end points of an ASR are depicted by dark dots (split points) located on a set of connected lines representing the physical movement of the robot. Figure 5.4 shows the actual paths taken by the robot through one of these ASRs. The zigzag movement of the robot in between two splits is clearly visible, and can be quite different from the line connecting start and end points. This is especially true if the ASR describes a large empty space.

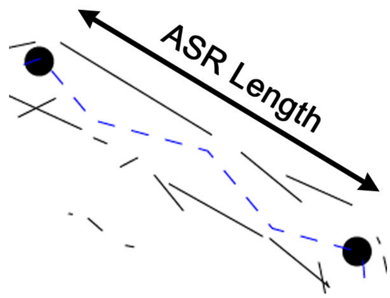


Figure 5.4 Section extracted from Figure 4.16 showing the paths of the robot (blue dashed line between the split points) can be quite different to the ASR length.

To localize using the distance traveled, it is better to use the *effective* distance traveled as measured by the total length of the ASRs visited. The length of an ASR is measured as the straight line distance between the entrance and the exit that the robot uses for moving through that ASR.

When the robot stops at point X in its homeward journey and wants to localize itself, it has two pieces of information. It can calculate the effective distance it has traveled so far using the ASRs computed in the CM-current. Let this be d_{RJ}, effective distance traveled so far during the return journey. It can also calculate the effective distance it needs to travel to reach the entrance of all the ASRs already computed in CM-in-memory i.e. those computed during the exploration phase. Let this be d_{ASRn}, effective distance needed to travel from the end point of exploration to the entrance of ASR_n. By comparing d_{RJ} and d_{ASRn} (for n = 1 to m where m is the number of ASRs computed in the outward journey), one can compute a local confidence map, $Cd_{RJ}(t)$ of all ASRs in memory to tell us the likelihood that the robot is in ASR_n.

Gaussian distribution is used for modeling the confidences for each ASR, as it allows for a smooth transition between regions, and the width can be easily adjusted by altering the standard deviation. The latter feature is rather useful. The further the robot travels, the overall distance traveled gets more and more unreliable due to slippage and drift. This error can be reflected by increasing the standard deviation that broadens the Gaussian, which in effect decreases confidence levels over time. For the plot, the horizontal axis is the distance traveled in millimeters, centered at the current overall effective distance traveled d_{RJ}. The Gaussian's standard deviation σ was chosen as $\sigma = 0.05d$. Note that although a Gaussian distribution is used here, we do not try to model a probability density function, but rather make use of its bell-shape curve. The normal distribution is computed using the following equation:

$$f(x) = \frac{e^{-(x-\mu)^2/\left(2\sigma^2\right)}}{\sigma\sqrt{2\pi}} \qquad \textit{Equation } 5.1$$

The confidence value for a particular ASR is determined by sampling the Gaussian at the position given by the accumulated (ASR) distances from the origin (i.e., where the robot started the homeward journey) to the entrance of this ASR. Figure 5.5 shows an example of this computation.

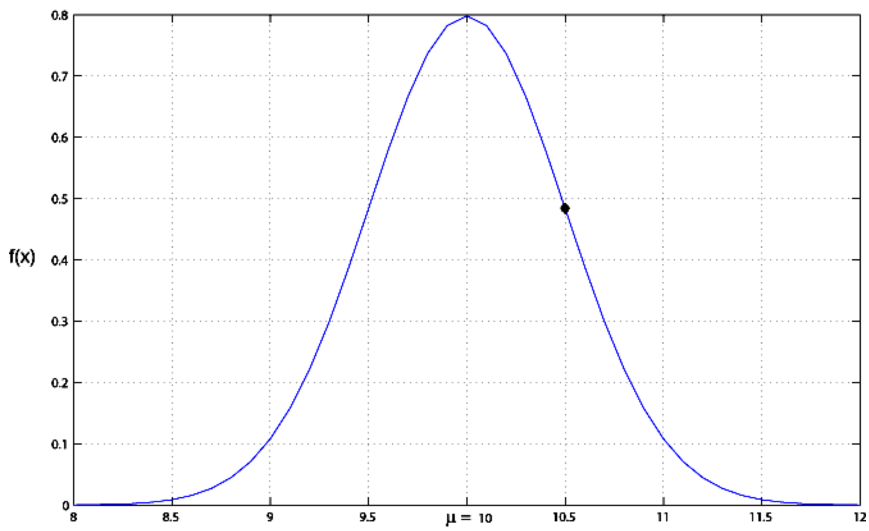


Figure 5.5 Example of a normal distribution curve used for determining the confidence level based on the distance strategy. The mean of this example is 10 and it has a standard deviation of 0.5.

Here, the robot has moved effectively 10 meters in the homeward journey. Hence the mean is 10 and standard deviation is 0.5. The confidence level is sampled along this curve. For instance, if ASR 8 in CM-in-memory has an accumulative distance of 10.5, then its confidence level is approximately 0.48, as shown by the black solid dot. After a value for each ASR is computed, the local confidence map $c_{Dist}(t)$ is normalized to the interval [0; 1] by dividing the values by the maximum value, which is $f(\mu)$.

5.2.2 Localization Strategy – Relative Orientation

The second method for computing local confidence maps containing estimates of the robot's position with respect to the CM-in-memory is based on using relative orientation information between adjacent ASRs. During its journey, the robot enters an ASR at one location and exits at a different one, usually involving multiple movements of various directions in between. We define the direction of an ASR as the direction of the line connecting the entrance and exit points.

As before, direction information varies every time the robot travels through the environment. However, the overall shape between adjacent ASRs is relatively stable. We therefore propose to use angles between ASR directions as a local measure of the current position of the robot. Note that this information is not very effective on its own, because the same angles (i.e., direction changes) can be found at different locations of the environment. This is especially true when the robot has made a lengthy exploration journey. Nonetheless, by combining this strategy with others, it can help to choose between position estimates that would otherwise be indistinguishable.

Firstly, all angles $\alpha_1, \ldots, \alpha_{N-1}$ between adjacent ASRs in the CM-in-memory are computed once the exploration journey has been completed. In the re-mapping process while returning home, new ASRs are computed in the new map based on data gathered while the robot travels. Using the direction information contained in this map, the angle β between the current ASR and the previous one can be computed. Comparing this angle to all angles of the original map gives a clue (or multiple clues) for the current location of the robot.

The comparison of angles is done by computing the difference angle between the current angle β obtained from the newly generated map and all angles α_i of the original map. This difference angle is then mapped linearly to the interval [0; 1]. Hence the confidence level c_{Dir_i} is give by:

$$c_{Dir_i} = -\frac{1}{\pi}\left|\alpha_i - \beta\right| + 1, \qquad i = 1,\ldots,N-1 \qquad \textit{Equation } 5.2$$

Another option would be to use the cosine of the difference angle instead of the linear mapping. However, this “compresses" confidence values for similar angles, which is not a desirable effect.

The mapping given by this function results in high values for similar angles, and low values for dissimilar ones. The confidence map computed this way is used for further processing. Since the overall reliability of the relative orientation strategy as described above is rather low compared to the confidence values from other methods (in this case using distance information), the confidence values from relative orientation are reduced by a constant factor. As the data fusion method implemented is capable of adjusting the relative weights of the different localization strategies, this is non-critical, because it will change automatically over time, depending on the reliability of the other methods as well.

5.2.3 Fusion by Democratic Integration

The Democratic Integration technique was originally proposed in Triesch (1999) and Triesch & von der Malsburg (2001) for sensor data fusion in computer vision, using images as input data. The method was extended and embedded into a probabilistic framework for 3D object tracking in Denzler, Zobel, & Triesch (2002); Kahler & Denzler (2005) and Kahler, Denzler, & Triesch (2004). It is also used for segmentation purposes in Tang, Garrett, & Malsburg (2005).

The separate local confidence maps are merged into a global confidence map by computing a weighted sum of all local maps. The main advantage of the Democratic Integration algorithm is that it allows for the weights to be adjusted dynamically and automatically over time, depending on the reliabilities of the local map.

Given M number of strategies with their respective local confidence maps $c_{li}(t)$ at time t, the global map $c_g(t)$ is computed as:

$$c_g(t) = \sum_{i=0}^{M-1} w_i(t) c_{li}(t) \qquad \textit{Equation } 5.3$$

where $w_i(t)$ are weighting factors that add up to one.

An estimate of the current position of the robot with respect to the original map can now be computed by determining the largest confidence value in $c_g(t)$. The index *n* that produces the highest confidence value is equivalent to the index of the ASR that the robot believes it is in. Furthermore, the value $c_{g_m}(t)$ itself is a good indicator of how reliable the position estimate is in absolute terms, while comparing it to other ASRs shows the relative reliability.

One problem of having various strategies is not knowing which strategies are reliable, and which ones are not. For this reason, it is an advantage to use Democratic Integration because the weighting factors $w_i(t)$ are automatically updated after each time step. They are adjusted according to the reliability of the confidence output. If a particular method constantly reports low confidences, its influence on the overall confidence map decreases over time. This means that the fusion will automatically detect unreliable localization strategies and decrease their influence over time.

To update the weighting factors, the local confidence maps $c_{li}(t)$ have to be normalized first, so that the quality of each local map can be computed. The normalized map $c'_{li}(t)$ is given by:

$$c'_{li}(t) = \frac{1}{N} c_{li}(t) \qquad \textit{Equation } 5.4$$

where *N* is the total number of ASRs in the outward map.

The idea when updating the weights is that local confidence maps that provide very reliable data get higher weights than those which are unreliable. Different ways for determining the quality of each local confidence map are presented in Triesch & Malsburg (2001). We use the normalized local confidence values at index *m*, which has been determined from the global confidence map as described above, i.e., the quality $q_i(t)$ of each local map $c_{li}(t)$ is given by $c'_{l_m}(t)$. Normalized qualities $q'_i(t)$ are computed by:

$$q'_i(t) = \frac{q_i(t)}{\sum_{j=0}^{M-1} q_j(t)} \qquad \textit{Equation } 5.5$$

The new weighting factors $w_i(t+1)$ can now be computed from the old values using the following equation:

$$w_i(t+1) = w_i(t) + \frac{1}{t+1}\left(q'_i(t) - w_i(t)\right) \qquad \textit{Equation } 5.6$$

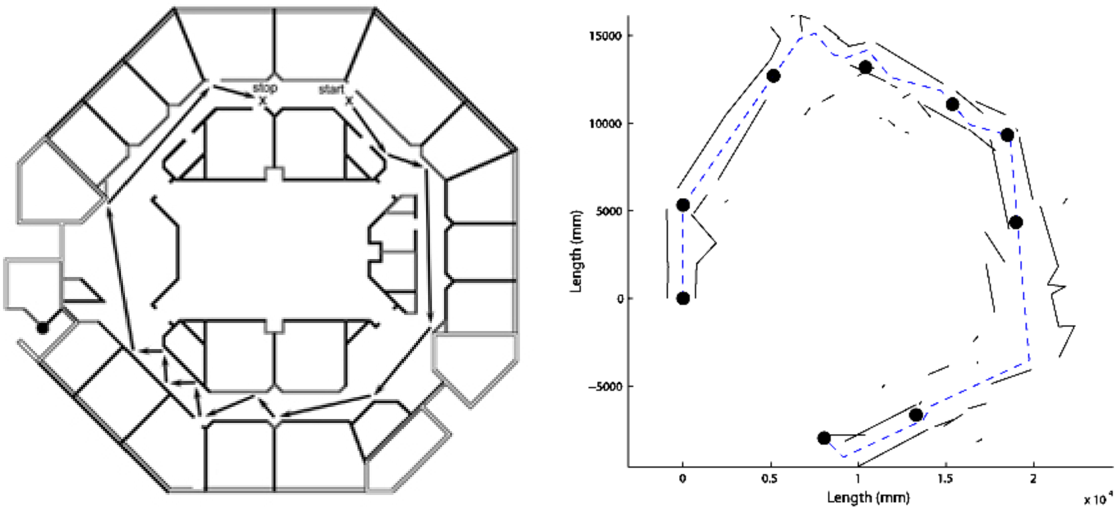
This is a recursive formulation of the average over all qualities from time zero to time *t*. Using this update equation and the normalization of the qualities in *Equation 5.5* ensures that the sum of the weights equals one at all times (see Triesch (1999) and Triesch & Malsburg (2001) for details).

5.3 Experiments

Various experiments were conducted to ascertain whether the robot is able to successfully determine its location based on its cognitive map of the environment. This section describes three such experiments.

5.3.1 Experiment 1

Figure 5.6 shows the journey the robot took during exploration and the CM-in-memory computed (which are from Figures 4.1 and 4.16). The robot was then instructed to return to home. The physical movement of the robot during the going home journey is as shown in Figure 5.7 and its representation of the environment when it stopped is shown in Figure 5.8.



(a) (b)

Figure 5.6 (a) The robot's journey during the exploration of Experiment 1 and (b) the CM-in-memory computed for this journey.

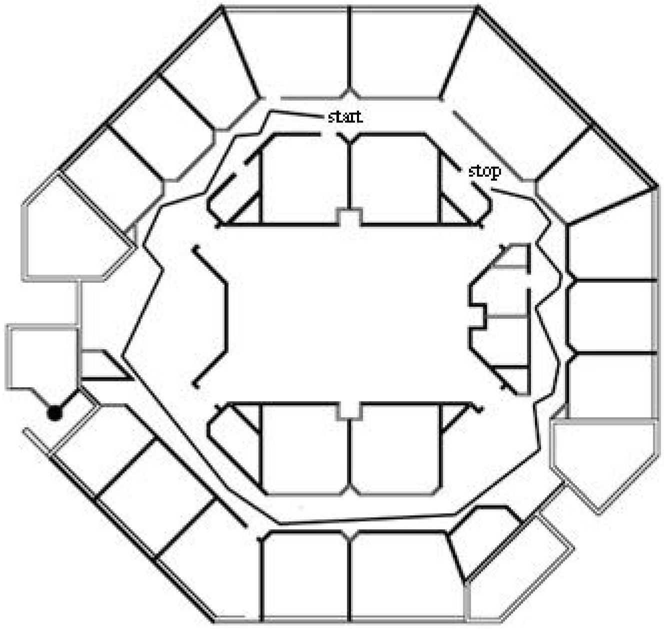


Figure 5.7 The robot's return home journey.

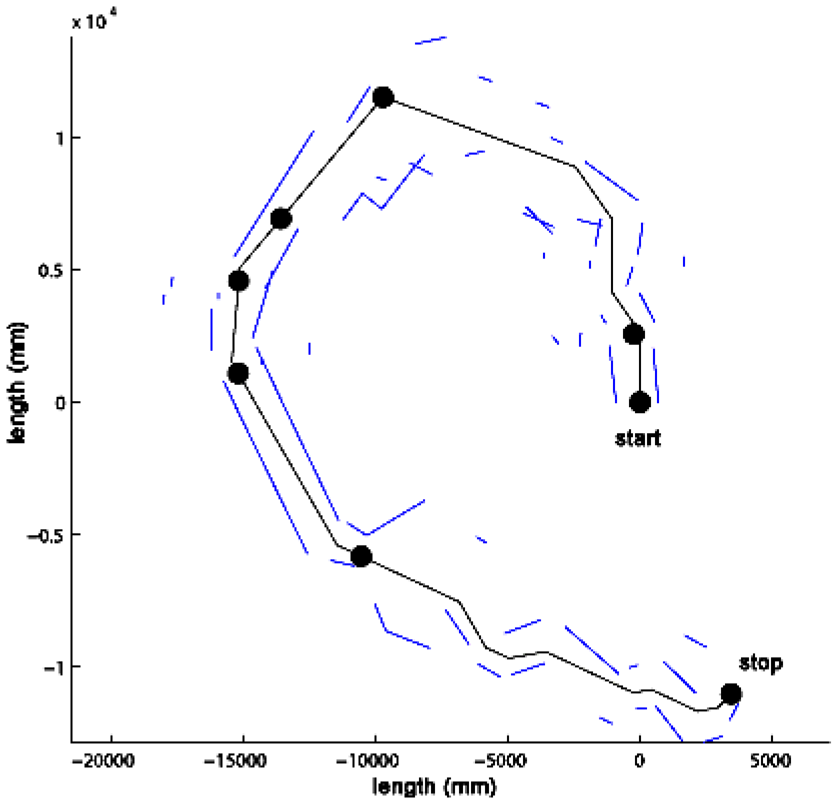


Figure 5.8 The robot's representation of the environment computed at the end of the going home journey.

The graphical plots in Figure 5.9 show the confidence maps computed at four different locations during the return home journey. The light dotted lines show the ASR estimate using the ASR length information (distance method) and the dark dashed lines depicts the ASR estimate using the angles between ASRs (relative orientation method). The solid line is the overall ASR estimate for the corresponding ASR (horizontal axis). Note that the confidences have values between 0 and 1 (vertical axis) but do not sum up to 1.

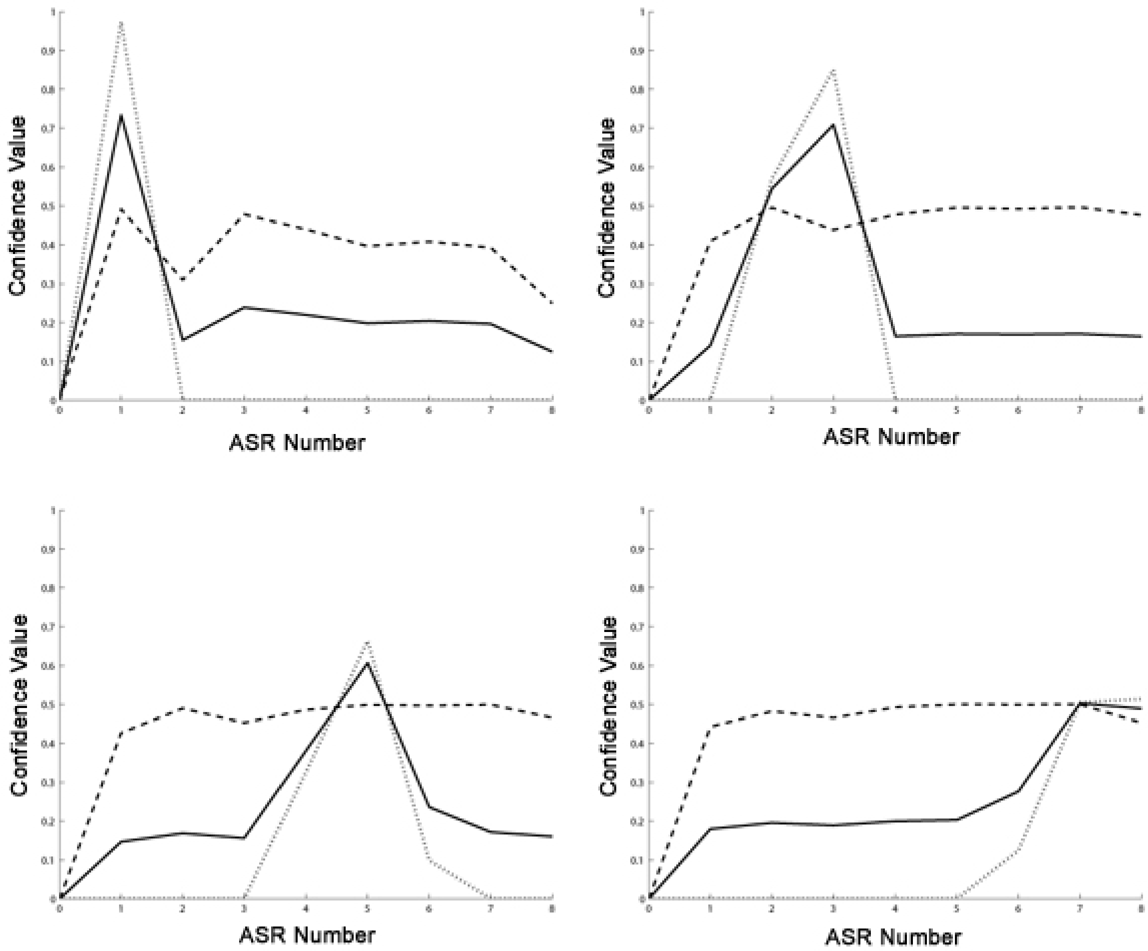


Figure 5.9 Confidence maps generated at 4 different locations during the going home journey. The light dotted line represents confidences generated using the ASR distance method, dark dashed line is the confidences from the ASR relative orientation and the solid line represents the overall confidence map.

In Figure 5.9, the top left plot shows a narrow peak for the overall confidence at ASR 1, with corresponding peaks for both distance and relative orientation. The narrow peak signifies the robot being very confident of being in ASR 1. The top right graph however is quite different. It has two peaks, one slightly higher than the other. It shows that the robot is around the transition region between ASR 2 and 3. The higher confidence value for ASR 3 shows that the robot feels that it has already moved into ASR 3. The bottom left plot shows a similar account except that this time, both neighboring ASRs (4 and 6) have higher values, resulting in a wider peak. Finally, the bottom right map shows that ASRs 7 and 8 have similar confidence levels, indicating that the robot is unsure which of these two ASRs it is in.

Figure 5.10 shows the distribution of weighting used on the localization strategies for calculating the overall confidence values. The solid line is the plot of distance method weights and the dashed line represents the weights of the relative orientation method. For the first three ASRs, the weights for both distance and orientation are initialized to be of equivalent values (0.5 in this case since there are only two strategies) because we only start computations after the third ASR. The weight for ASR distance then increases, because the confidence on ASR distance is higher than the confidence computed from the orientation method. As time progresses, the weights of the distance method decrease due to a decrease in confidence, and vice versa for confidence from the orientation method.

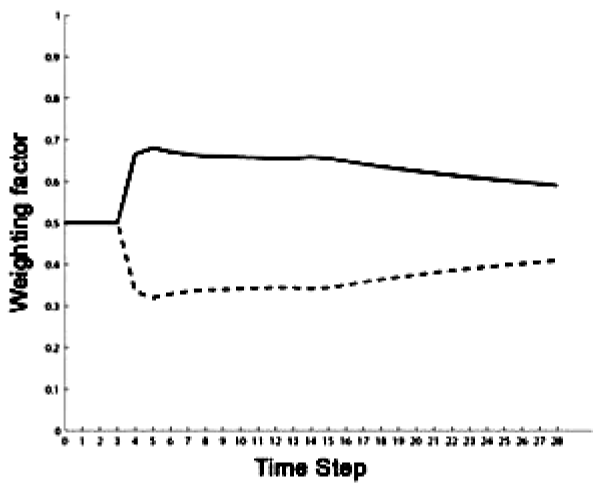


Figure 5.10 Adjusting the weighting factor to reflect the confidences on each of the strategies used. Weights for ASR length (solid line) and weights for ASR relative angles (dotted line).

The current experiment presented a simple environment which consisted primarily of corridors and a couple of small spaces. This experiment was repeated more than 20 times and achieved a success rate of 80%. The experiment was considered success if the robot reached +/-3m from its home position. For this particular example, the robot stopped two meters short of home. The next experiments look at traversing more complex environments, including across a large space, where large spaces are defined as areas that are larger than the robot's sensing capabilities. Furthermore, it is also important to note the robot's path crossed over on more than one occasion in both the exploration and return home journeys. If conventional robotic mapping was used, the algorithm would have needed to account for surfaces that have been detected more than once.

5.3.2 Experiment 2

For the second experiment, the robot moved through the large room in the middle of the building. The journey the robot took during the exploration journey is shown Figure 5.11.

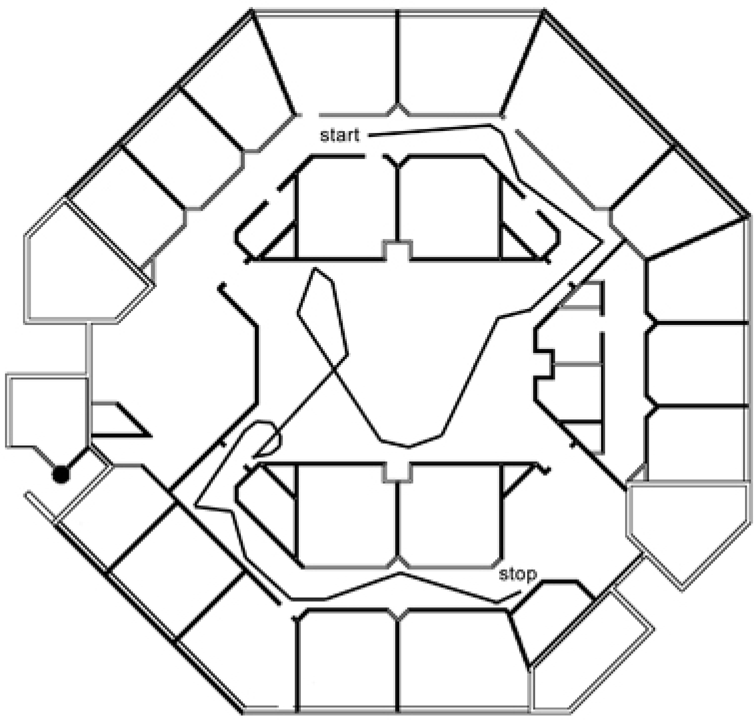


Figure 5.11 Physical movement of the robot during the exploration journey of Experiment 2.

The robot's cognitive map of the environment, which becomes CM-in-memory for this experiment, is shown in Figure 5.12. Note that the robot's starting location is at coordinates (0, 0). ASR segmentations for the explored environment are represented by the black dots.

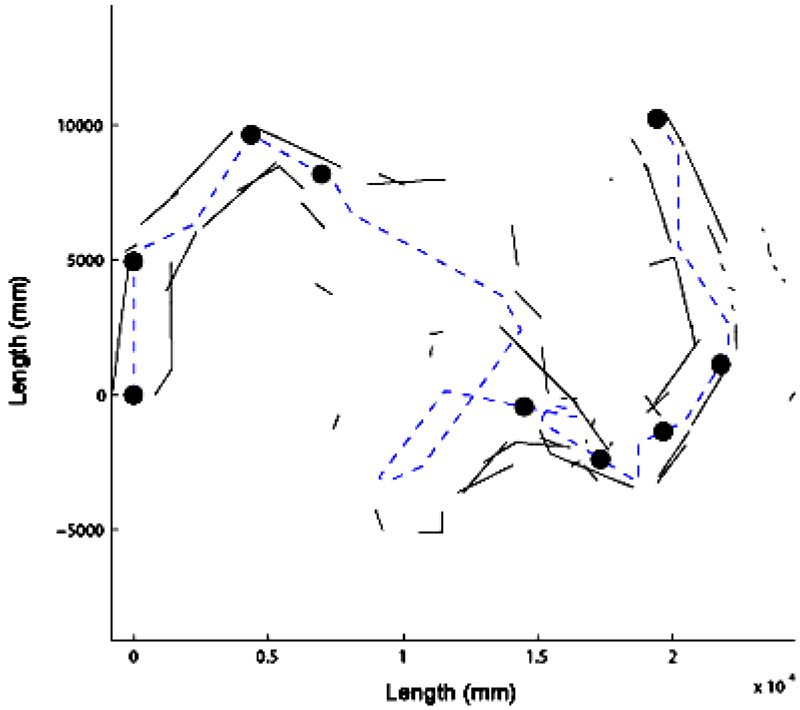


Figure 5.12 Representation of the environment generated by the robot during the exploration journey. The solid black dots represent the split points computed. Hence, the spaces between the split points are the ASRs.

The robot's journey to return home is depicted in Figure 5.13, which shows very different paths compared to the exploration journey. The ASRs generated during the exploration journey are indicated by ellipses and are numbered from zero starting at the last ASR (so that during the homeward journey ASRs are visited in ascending order).

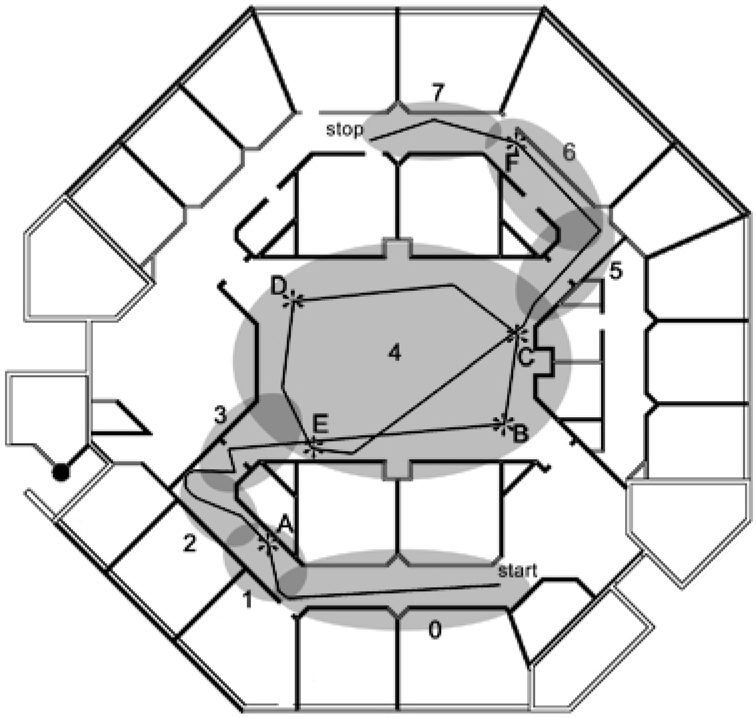


Figure 5.13 Physical movement of the robot during the homebound journey (solid line). The ellipses (with their respective labels 0-7) represent the ASRs computed during the exploration journey (i.e. see Figure 5.12). The asterisks (with their respective labels A-F) are locations where confidence maps are displayed in the plots in Figures 5.15 and 5.16.

The cognitive map of the robot computed during the homebound journey is shown in Figure 5.14. Note that due to the re-initialization of the robot before returning home, the starting point for both, outward and homeward journeys, is the origin of the coordinate system, which also results in the map depicted in Figure 5.12 to be upside-down with respect to the map in Figure 5.14.

As in the previous example, with maps generated during both journeys, it is obvious that different representations are computed, in particular a different number of ASRs, and split points generated at different locations. Furthermore, and as shown in Figure 5.14 (see the dotted path crossing over a solid line), it is also interesting to note that the CM-current is so distorted that the robot "moved" through an earlier detected wall to exit the large room.

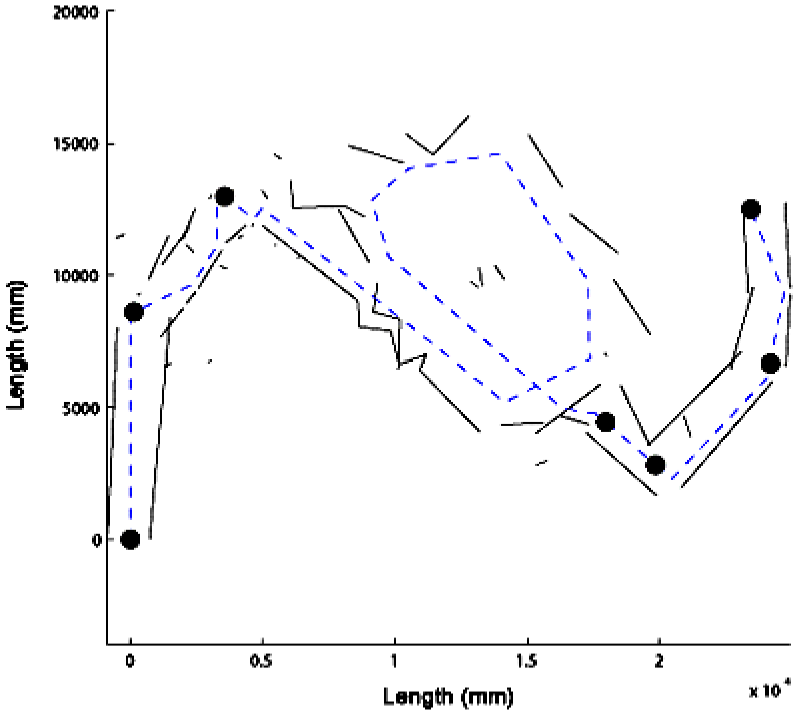
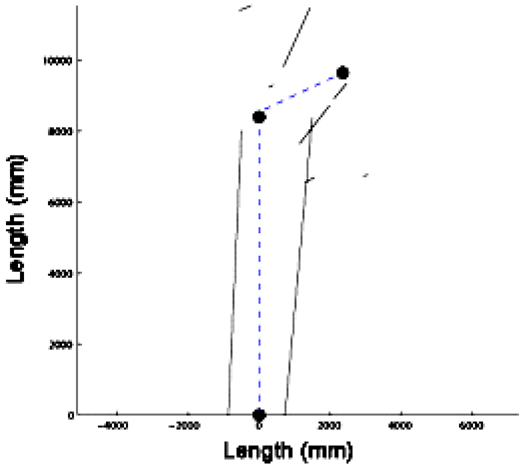
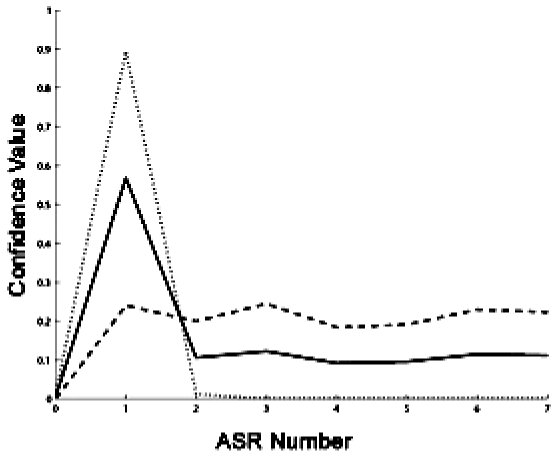


Figure 5.14 Representation of the environment generated by the robot during the homebound journey.

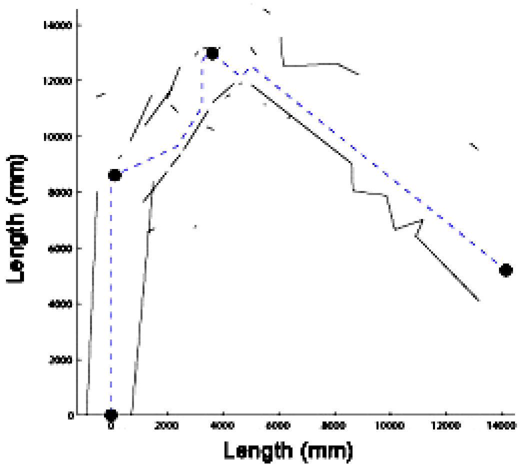
The position of the robot is always given with respect to the ASR representation of the original outward journey map. What can be clearly seen in all maps generated in both experiments is that the algorithm is capable of representing the large room as a single ASR, independent of the path that the robot took while mapping. The algorithm is capable of separating the corridors from bigger rooms (e.g., ASRs 3 and 4 in Figure 5.13), and corridors from other corridors. To evaluate the performance of the localization algorithm, confidence maps have been selected at six positions on the homeward journey. These physical locations are marked by asterisks, labeled A - F in Figure 5.13.

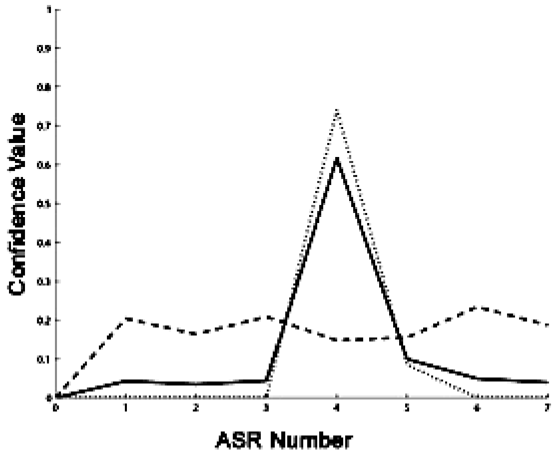
Firstly, the plots of Figures 5.15 (a), (c) and (e) show the intermediate maps generated at positions A, B and C in Experiment 2 (see Figure 5.13). The corresponding confidence maps are depicted in the graphs of Figures 5.15 (b), (d) and (f). Similarly, Figures 5.16 (a), (c) and (e) show the intermediate maps generated at positions D, E and F and their corresponding confidence maps are presented in Figures 5.16 (b), (d) and (f).



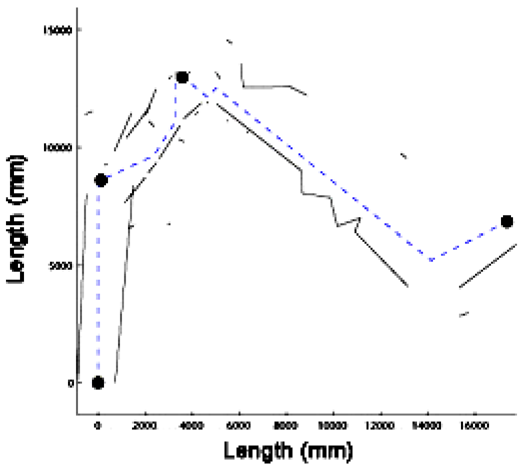


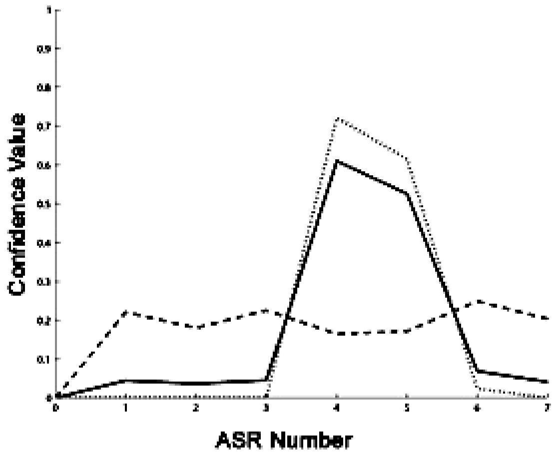
(a) (b)





(c) (d)





(e) (f)

Figure 5.15 Cognitive maps (a, c, e) and confidence maps (b, d, f) generated at locations A, B and C respectively (see Figure 5.13).

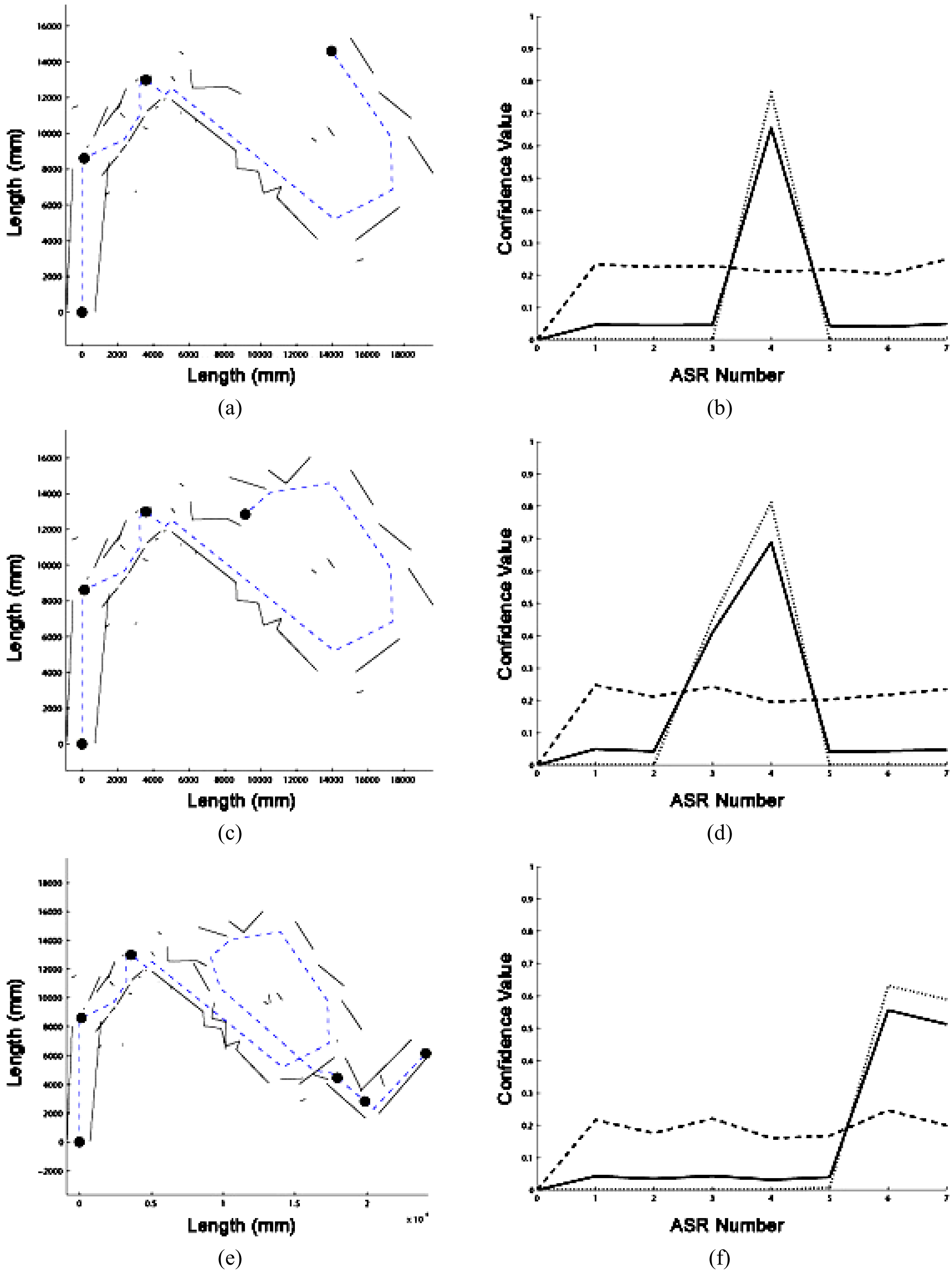


Figure 5.16 Cognitive maps (a, c, e) and confidence maps (b, d, f) generated at locations D, E and F respectively (see Figure 5.13).

In the confidence graphs, the light dotted line shows the ASR estimate using the ASR length information (distance method) and the dark dashed line depicts the ASR estimate using the relative angles between ASRs (relative orientation method). The solid line is the overall estimate after fusion. The horizontal axis corresponds to the index of the ASR, the vertical axis to the confidence level, which can have values between zero and one.

The confidence graph at position A (Figure 5.15 (b)) shows a peak for the overall confidence at ASR 1, signifying the robot is very confident of being in this particular local space. The same is true for the graph in Figure 5.15 (d) corresponding to position B, where the robot has moved far into the big room in the centre. The confidence graph at position B shows a peak for ASR 4. Note that at this point, the robot is attempting to move out of ASR 4. It calculates the distance to travel and moves to C. Note that it moves towards C because that was the only forward motion possible from position B. At C, it is close to the border between ASRs 4 and 5, which is reflected by high confidence values for both ASRs in the confidence map in Figure 5.15 (f), where the robot gets more and more unsure about whether it is still in the big room or whether it has entered the next region yet (ASR 5). Nonetheless, it calculates the necessary distance to move out of ASR 4 from point C but it headed in the wrong direction.

Consequently, it moves further away from the exit and back into the room, reaching position D. At D, it is very confident again that it is still in ASR 4, not having actually exited the big room (see Figure 5.16 (b)). Position E is close to where the robot has entered the room, coming through ASR 3. Again, this can be observed in the confidences plotted in Figure 5.16 (d), which shows that the robot is still quite confident of being in ASR 4, but where the value for ASR 3 has increased considerably, i.e., the robot “knows" that it is close to where it was when it entered the room. Finally, the graph in Figure 5.16 (f) was generated at position F, which is at the border between ASRs 6 and 7, reflected by high confidences for both.

The robot continued to travel into ASR 7 and stopped when it thought it has reached home. In this experiment the robot stopped to within one meter of the actual location of home.

5.3.3 Experiment 3

The final experiment reported here is similar to Experiment 2. The physical movement of the robot during the exploration journey and the cognitive map generated during this journey are shown in Figures 5.17 and 5.18 respectively. The homebound journey is shown in Figure 5.19 and the map it generated is depicted in Figure 5.20.

As before, the ASRs generated during the exploration journey are indicated by ellipses and are numbered from zero starting at the last ASR. Six locations have also been chosen from the going home journey (labeled A-F) to determine the performance of the localization algorithm. The intermediate maps generated at these locations and their corresponding confidence maps are shown in Figure 5.21 and 5.22.

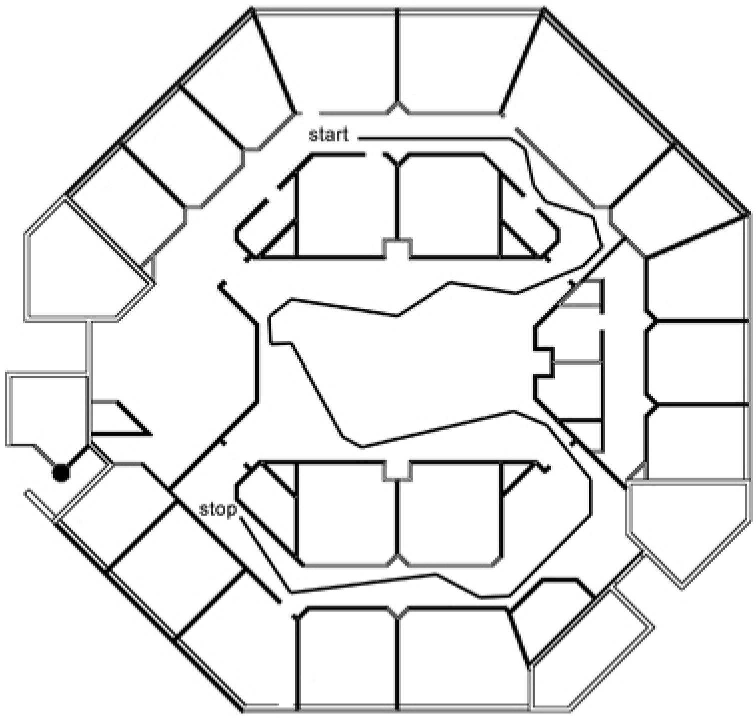


Figure 5.17 Physical movement of the robot while exploring the environment in Experiment 3.

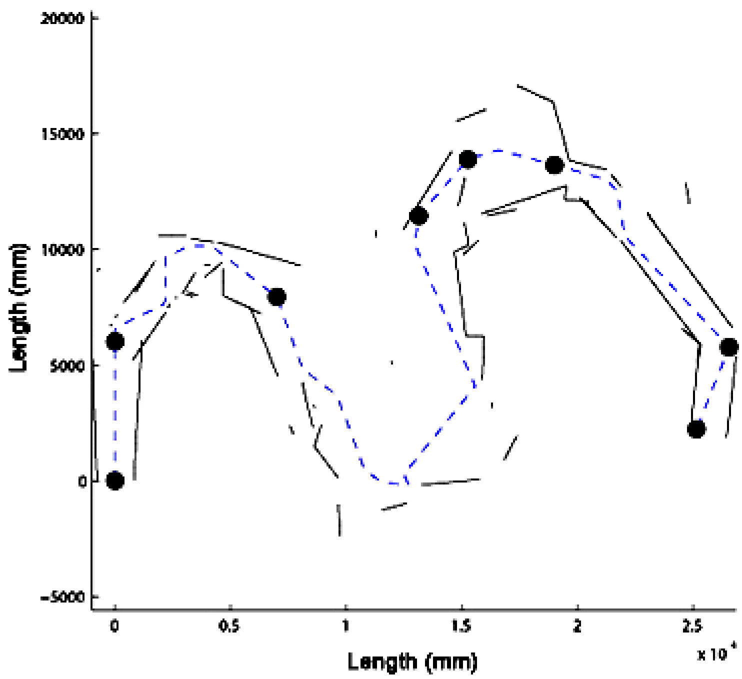


Figure 5.18 Representation of the environment generated by the robot during the exploration journey. The solid black dots represent the split points computed.

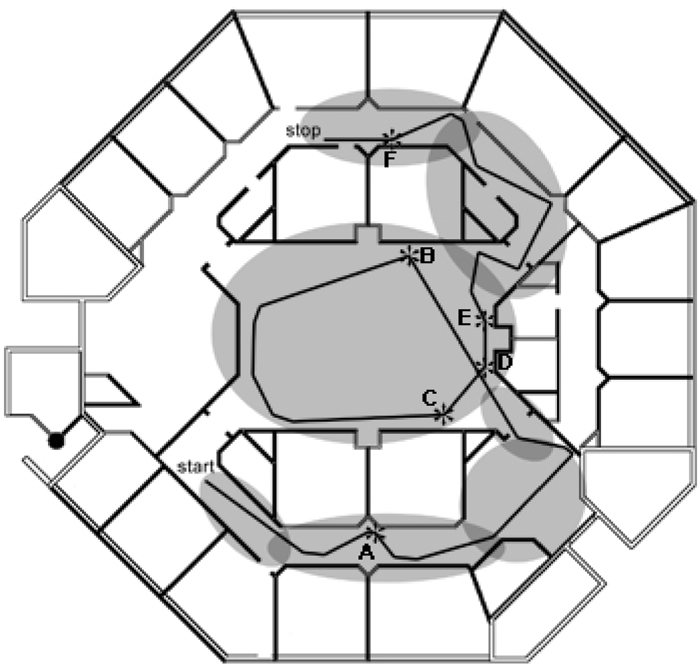


Figure 5.19 Physical movement of the robot during the homebound journey. The ellipses (with their respective labels 0-7) represent the ASRs computed during the exploration journey. The solid dots (with their respective labels A-F) are locations where confidence maps are displayed in the plots in Figures 5.21 and 5.22.

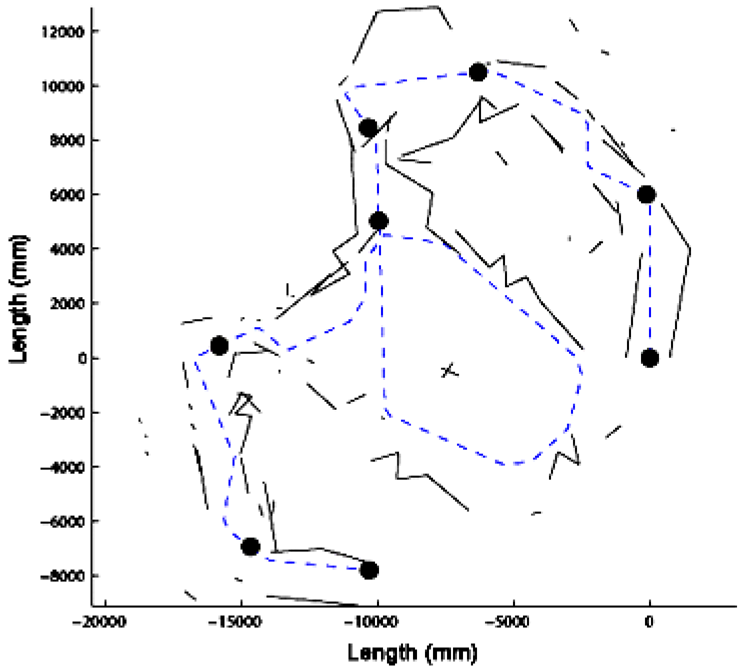


Figure 5.20 Representation of the environment generated by the robot during the homebound journey.

Figures 5.21 and 5.22 show intermediate maps generated at positions A - C and D - F respectively for Experiment 3; and the corresponding confidence maps. As in the previous experiment, when reaching position A the robot is very confident about being in ASR 1, indicated by the distinct peak in the graph in Figure 5.21 (b). At position B it has entered the big room being ASR 4. As can be seen in Figure 5.21 (d) it is not sure about whether it is still in ASR 4, or whether it has reached the next ASR already, the confidences for both being equally high, with a slight bias towards ASR 4.

Taking into account odometric inaccuracies and the information that is available to the robot using the present localization methods, this behavior makes perfect sense. When it reaches position C after traveling through the big room in a loop, the robot is highly confident (Figure 5.21 (f)) that it is still in ASR 4 (which is true for the positions in between as well, but not shown here).

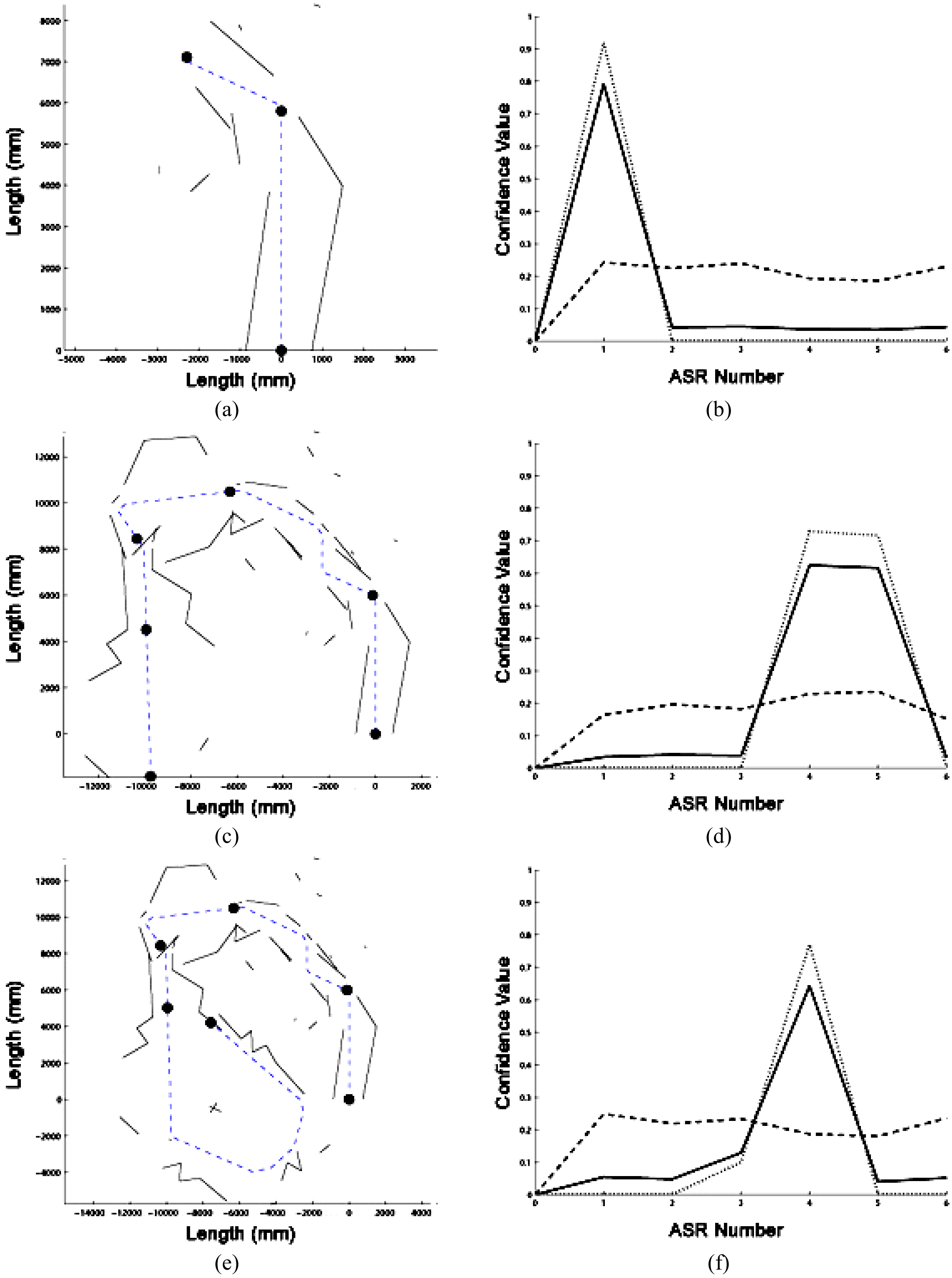


Figure 5.21 Cognitive maps (a, c, e) and confidence maps (b, d, f) generated at locations A, B and C respectively (see Figure 5.19).

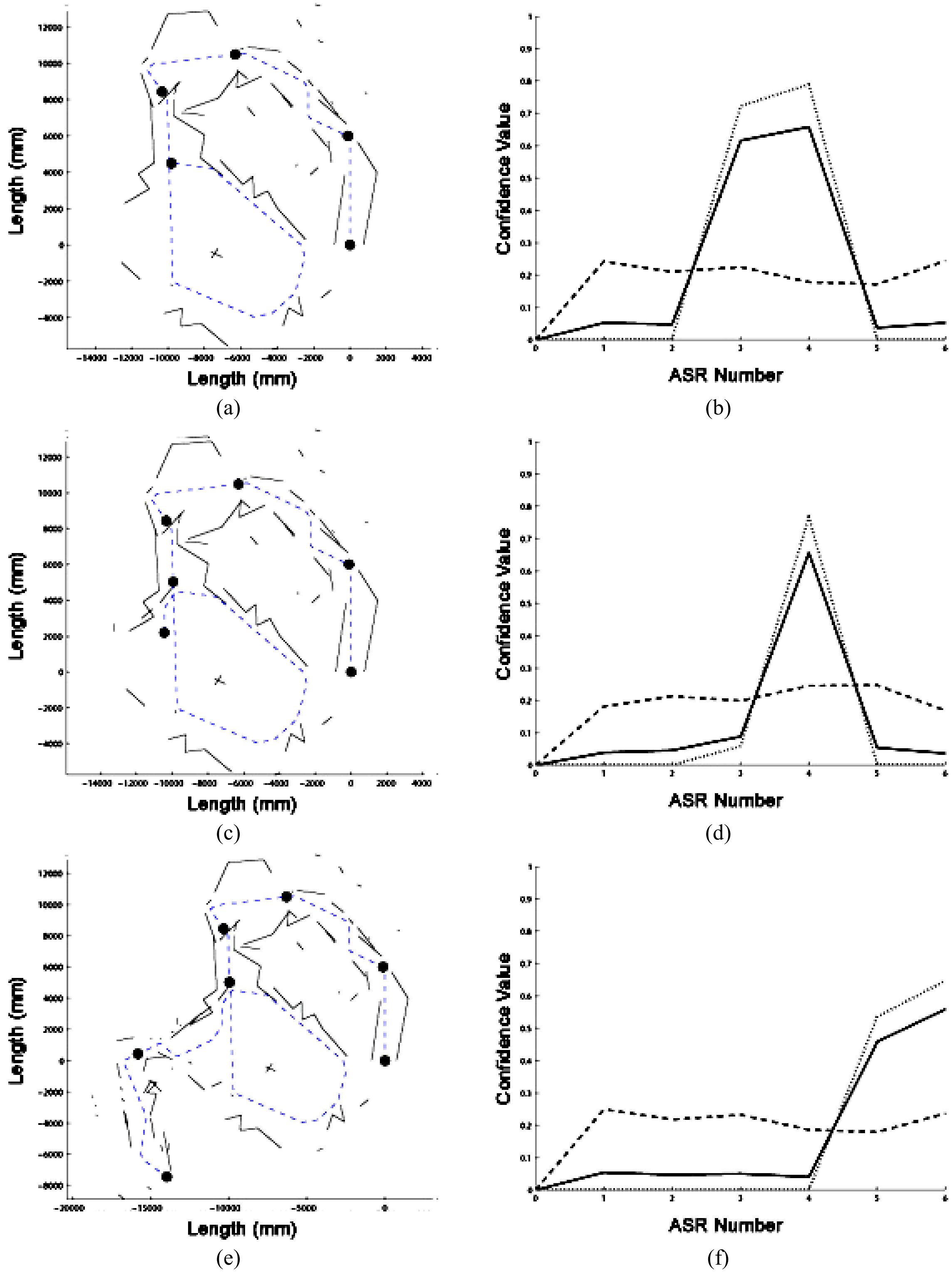


Figure 5.22 Cognitive maps (a, c, e) and confidence maps (b, d, f) generated at locations D, E and F respectively (see Figure 5.19).

Being close to where it has entered the room, the confidence for ASR 3 starts to increase at this location. This effect becomes more prominent in Figure 5.22 (b), when the robot is at position D, which is at the border between ASRs 3 and 4. Again, it is unsure about which ASR it is in; meaning that it can infer that it actually is at the border between two regions, near the entrance to the room. Moving away from the border to position E (Figure 5.22 (d)), it is again confident about being in ASR 4. At the last position F, the robot is very close to its home position in ASR 6 already, which is reflected in the high confidence values shown in Figure 5.22 (f).

In this experiment, the robot stopped to within 0.5 meter of the home location. More than 10 similar experiments were conducted and the success rate was about 70%.

5.4 Discussion

More than 30 experiments were conducted and they demonstrated consistent results, with over 75% of them successfully reaching home. When the robot was able to successfully locate home, it was usually within +/- 2m from the target in the physical world.

Many strategies have been developed by animals in nature to help them find their way home or to migrate to faraway places. For example, they could perform vector calculation, orient towards the magnetic pole and reason about the geometry of space. The question explored in this chapter is: Could the robot find its way home after computing a fuzzy map of its environment i.e. without the need to compute a precise metric map?

The answer is yes. A new algorithm has been successfully developed for the robot to find its way home using a fuzzy map consisting of a network of ASRs. The navigation strategy is similar to that used in animals – try to get near to your goal (exit in this case) and then search for it. The localization algorithm makes use of the effective distance traversed as measured between the entry and exit point of each ASR and the relative orientation between adjacent ASRs. Separate confidence maps are created for each kind of information used for localization and a fusion algorithm is implemented to combine them to produce a global confidence map.

While one does not expect the relative orientation between adjacent ASRs to be effective (and especially in such an environment), the use of the distance between exits of ASRs is surprisingly effective as a localization tool. It helps the robot to get home. However, its limitation can be seen when the robot wanders through the large middle room. When the robot is searching around in the middle room on its way home, it does not quite know where it is. It was just trying to travel the next x meters (which it calculates itself) to reach the next exit. After trying a few times, it manages to get out in the right exit and heads for home. The limitation here is due to the inability of the robot to orient towards its goal and to recognize a place. The robot can calculate an initial orientation towards the goal when it first enters a new ASR but, after that, it will not be able to orient towards the exit it wants to go to.

Work reported in this chapter has been published in Wong, Schmidt & Yeap (2007) and Schmidt, Wong & Yeap (2007).

Chapter Six

Conclusion

One significant contribution of this thesis is to show how a mobile robot could be used to investigate the nature of cognitive mapping. Since the mobile robot was invented, the term, robot mapping, has become synonymous with the process of creating a map for the robot to use. Consequently, the map created is crisp: it is complete, accurate, and useful for navigation. Cognitive mapping, however, describes how cognitive agents gain their environmental knowledge. A cognitive map is thus incomplete, fuzzy, and often cause confusion when used initially for finding ones way.

Nonetheless, a mobile robot is an autonomous system and there is no reason why it could not be programmed to do cognitive mapping. The real question is *how*. This research demonstrated that we could program a robot to compute a fuzzy cognitive map and then experiment with animal-like strategies to perform interesting cognitive mapping tasks. Some implications to be drawn from this research about cognitive mapping will be discussed in the sections below. Note that much of the work done here is still considered as exploratory and much more still needs to be done.

In summary, different theories and implementations of cognitive mapping models were discussed in Chapter 2. Recent experimental works on cognitive mapping were reviewed in Chapter 3. Three key questions were asked in this research, namely:

1. Could the robot be used to compute some kind of a cognitive map?
2. Could the map computed be used to solve a cognitive mapping related task?
3. From the above, could further insights be gained about what is happening in the early cognitive mapping process?

Chapters 4 and 5 have discussed the work done to answer questions 1 and 2 above respectively. These works will be discussed again respectively in Sections 6.1 and 6.2

with respect to the insights gained about the cognitive mapping process, thus providing an answer to question 3 above. Section 6.4 concludes this chapter with a discussion on possible future work.

6.1 The Robot and Its Cognitive Map

There are many interesting questions one could ask about the nature of one's cognitive map and the early cognitive mapping process during the initial explorations of an environment. Some of these questions include:

1. The initial map is believed to be fuzzy and with loosely-coupled ASRs and sporadic landmarks. How could we get a robot to compute such a map?
2. Subsequent explorations would improve the map's contents but how does that learning process work?
3. To be able to have subsequent explorations, one would assume one could find one's way back home. How could one use the initial fuzzy map to find one's way home?
4. What happens to the information perceived on the way home and how does that affect the map computed? Should we merge them to form a more consistent map?
5. What happens when one is confused while exploring a large environment with loops?

This thesis has developed a robot that computes a fuzzy map of ASRs. Each ASR consists of a rough description of the shape of the local space traversed and possible exits. ASRs are identified from the environment explored using two criteria:

1. Similarity in the path traversed; and
2. Similarity in the bounded space.

However, due to the use of sonar sensors and odometer, the shape of each ASR computed is unreliable and its exits could be walls that the robot has failed to detect.

The robot also could not recognize objects and the use of landmarks cannot be investigated. The robot does not forget and so none of the ASR computed is forgotten.

The map computed is thus not loosely-coupled. However, it has one of the key properties of a cognitive map i.e. it is a fuzzy representation of each local environment visited. Furthermore, the representation computed is shown to be dependent on the direction of travel. With such a map, I have demonstrated how it could be used by the robot to find its way back home. Much is made use of following the direct path between exits of ASRs (but see next section for further discussion on the role of direction and distance information) to get back to home. The lack of accurate measurements is both unimportant and unnecessary. One only needs to travel close to a point of interest and then switches to a different strategy to locate the place accurately.

It is tempting to investigate how one could merge the map computed in the return journey with the map computed during the outward journey; thus allowing us to explore questions 2 and 4 above. However, without understanding the nature of why certain ASRs are not remembered, developing an algorithm to do so appears to be premature. To develop an algorithm to merge two representations is a straightforward task from a computational standpoint. It is important to know why. Furthermore, many of the above questions are beginning to probe deeper into the nature of cognitive map and its processes and it might not be appropriate to answer all of them using our current robot equipped with very poor sensors. After all, one does not expect, say, the ants to swim. The current robot provides us a demonstration as to how one might compute an initial cognitive map and use it to find one's way home. To investigate other questions, we will need to introduce a robot with more powerful sensors.

6.2 The Robot and Its Journey Home

Although the map computed by the robot bears an important characteristic of a cognitive map, it would not be very interesting if the robot could not use it to solve any cognitive mapping related problems. The returning home problem is identified and inspired by the foraging behavior of animals. It is a nice problem to investigate since one could begin studying the problem using a very simple environment. The lessons learned could help us develop more sophisticated algorithms for wandering further away from home and for dealing with more complex environments.

Prior to developing an algorithm for the robot to find its way home, the following has been observed:

1. Animals make much use of distance and orientation information to find their way home.
2. Animals develop ingenious algorithms using such information, and others, to find their way home.

What is implicitly available in a network of ASRs is precisely, distance and orientation information but, how one could utilize the information to find one's way home is unclear. Nonetheless, from observation (2) above, if animals could develop several ingenious algorithms then it is likely that one could develop an interesting algorithm for the robot. The robot could be treated as a species of its own kind. From that perspective, it is a species which is, perhaps, at the level of sophistication of the ant. Unlike the ant, it can compute ASRs but unlike higher species, it cannot recognize them from a distance. It could measure distances traveled in a crude way and could orient, very roughly, towards the end of an ASR. The robot thus created an interesting micro-world to study the kind of algorithms one could develop for the robot to find its way home and from the algorithms developed what could one say about cognitive mapping for simple animals?

The algorithm developed makes much use of distance information. Note that the type of orientation information being used is proven to be ineffective. With hindsight, I have used orientation information as an additional piece of information for identifying which ASR the robot is in and developed a general algorithm for combining information from different sources. The orientation information used is not the kind that helps the robot to orient itself towards home. With hindsight, the robot's ability to orient towards the end of the ASR, no matter how crudely; turns out to be crucial for the robot's success in finding its way home. In the earlier version of the navigation algorithm, the times when the robot failed to find its way home is because it failed to orient properly towards the end of the ASR. In the latest version, the inability to orient also shows up to be problematic for the robot finding its way home in the middle of the room.

There is no doubt that distance and orientation information are important for all cognitive mobile agents and this has been noted many times in many psychological experiments on animals' way-finding behavior. However, experimenting with a robot with such limited sensors has highlighted the significance is on the *sensing ability* of any mobile agents to orient themselves, be they cognitive or not. It is not just the ability to compute orientation information from the map but that the agent must have the appropriate sensor to orient. Our robot could compute the orientation of the exits but it does not have the ability to orient towards it, except for the initial entry into the ASR.

With this new emphasis, it is interesting to note how lower species (such as ants, bees and birds), who need to travel far in their foraging behavior, have evolved fascinating mechanisms to orient themselves. Furthermore, it is interesting to note that the evolution to more powerful sensors (such as vision) actually produce sensors which combine the ability to orient with other capabilities (such as recognition). Researchers interested in finding out how a new species find their way home would do well in searching for how that species orient themselves.

6.3 Conclusion and Future Work

This research has successfully developed a robot to compute a fuzzy cognitive map and uses it to find its way home. It has developed four new algorithms for use by a robot exploring an environment using sonar sensors. From the experiments, the significant lessons learned are:

1. The importance of paths between exits of an ASR;
2. The importance of having accurate sensors to orient towards a goal;

3. Distance information need not be very accurate as long as one knows roughly how far to travel;
4. A simple but useful navigation strategy is to find oneself close to home and when there, search for its exact location.

From an algorithm point of view, it would be interesting to add new algorithms to compute more information about the environment for the robot to localize itself. One such piece of information is the rough extent of ASRs. However, improving the performance of the robot in this way does not, in my view, add much to our understanding of cognitive mapping. That is, the current robot species with its very limited sensors is too primitive to be used for further investigations into the nature of cognitive mapping. In particular, sonar sensing failed to enable the robot to orient itself towards a target. This capability is found to be extremely important for the study of cognitive mapping.

A more interesting and useful extension is to incorporate more powerful sensors (such as the use of compass, laser, and vision) into the robot. This will enable us to investigate other interesting questions about the cognitive mapping process such as:

1. The use of short-cuts – could the robot return home via short-cuts? How are short-cuts computed from a cognitive map?
2. The use of landmarks – what is a landmark? Why are they computed sporadically rather than in, say, every ASR?
3. The use of different environments – How easy is it to implement a similar algorithm to learn a large environment with loops?
4. Could the algorithm be extended to take the robot out onto the streets or into a non-office like environment (ignoring problems related to mobility issues)?

The last 2 questions posed are interesting also from a robot mapping point of view. If the answer is yes for an office-like environment (i.e. one could use similar algorithms to learn a large office-like environment with loops in a straightforward manner), then we would have a different approach for robots to learn an office-like environment. That is, an approach that does not depend heavily upon the accuracy of the sensors and is not about computing a mathematically precise map of the environment. This new approach might turn out to be more powerful and simpler to implement.

References

Arleo, A. (2000). *Spatial Learning and Navigation in Neuro-mimetic Systems: Modeling the Rat Hippocampus.* Doctor of Philosophy, Swiss Federal Institute of Technology, Lausanne, Switzerland.

Baltzakis, H., & Trahanias, P. (2002). An Iterative Approach for Building Feature Maps in Cyclic Environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)* (pp. 576-581). Lausanne, Switzerland.

Beeson, P., Jong, N. K., & Kuipers, B. (2005). Towards autonomous topological place detection using the Extended Voronoi Graph. In *IEEE International Conference on Robotics and Automation (ICRA-05)* (pp. 18-22). Barcelona, Spain.

Beeson, P., MacMahon, M., Modayil, J., Provost, J., Savelli, F., & Kuipers, B. (2003). Exploiting local perceptual models for topological map-building. In *IJCAI-2003 Workshop on Reasoning with Uncertainty in Robotics (RUR-03)* (pp. 15-22). Acapulco, Mexico.

Best, P. J., & White, A. M. (1998). Hippocampal cellular activity: A brief history of space. *Proceedings of the National Academy of Sciences, 95*(6), 2717–2719.

Borges, G., & Aldon, M.-J. (2000). A Split-and-Merge Segmentation Algorithm for Line Extraction in 2-D Range Images. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR'00)* (Vol. 1, pp. 441-444). Barcelona.

Burt de Perera, T. (2004). Spatial parameters encoded in the spatial map of the blind Mexican cave fish, *Astyanax fasciatus*. *Animal Behaviour, 68*(2), 291-295.

Cheng, K. (1986). A purely geometric module in the rat's spatial representation. *Cognition, 23*, 149-178.

Cheng, K., & Gallistel, C. R. (2005). Shape parameters explain data from spatial transformations: comment on Pearce et al. (2004) and Tommasi & Polli (2004). *Journal of Experimental Psychology: Animal Behavior Processes, 31*(2), 254-259.

Cheng, K., & Wehner, R. (2002). Navigating desert ants (*Cataglyphis fortis*) learn to alter their search patterns on their homebound journey. *Physiological Entomology, 27*(4), 285-290.

Chiesa, A. D., Pecchia, T., Tommasi, L., & Vallortigara, G. (2006). Multiple landmarks, the encoding of environmental geometry and the spatial logics of a dual brain. *Animal Cognition, 9*(4), 281-293.

Chown, E., & Boots, B. (2008). Learning Cognitive Maps: Finding Useful Structure in an Uncertain World. In M. E. Jefferies & W. K. Yeap (Eds.), *Robot and Cognitive Approaches to Spatial Mapping*. Berlin / Heidelberg: Springer Verlag.

Chown, E., Kaplan, S., & Kortenkamp, D. (1995). Prototypes, Location, and Associative Networks (PLAN): Towards a Unified Theory of Cognitive Mapping. *Cognitive Science, 19*, 1-51.

Collett, M., & Collett, T. S. (2000). How do insects use path integration for their navigation? *Biological Cybernetics, 83*(3), 245-259.

Cuperlier, N., Quoy, M., & Gaussier, P. (2007). Neurobiologically inspired mobile robot navigation and planning. *Frontiers in NeuroRobotics, 1*(1).

Cuperlier, N., Quoy, M., Gaussier, P., & Giovannangeli, C. (2005). Navigation and planning in an unknown environment using vision and a cognitive map. In *IJCAI-05 Workshop: Reasoning with Uncertainity in Robotics (RUR-05)* (pp. 129-142). Edinburgh, Scotland.

Denzler, J., Zobel, M., & Triesch, J. (2002). Probabilistic Integration of Cues From Multiple Cameras. In *Workshop Dynamic Perception* (pp. 309–314). Bochum, Germany.

Downs, R. M., & Stea, D. (Eds.). (1973). *Image and Environment: Cognitive Mapping and Spatial Behavior*. Chicago: Aldine Publishing Coy.

Duda, R. O., & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. New York, USA: John Wiley & Sons.

Eichenbaum, H., Dudchenko, P., Wood, E., Shapiro, M., & Tanila, H. (1999). The Hippocampus, Memory, and Place Cells: Is It Spatial Memory or a Memory Space? *Neuron, 23*, 209-226.

Ekstrom, A. D., Kahana, M. J., Caplan, J. B., Fields, T. A., Isham, E. A., Newman, E. L., et al. (2003). Cellular networks underlying human spatial navigation. *Nature, 425*, 184-187.

Epstein, R., DeYoe, E. A., Press, D. Z., Rosen, A. C., & Kanwisher, N. (2001). Neuropsychological Evidence For A Topographical Learning Mechanism In Parahippocampal Cortex. *Cognitive Neuropsychology, 18*(6), 481-508.

Epstein, R., Graham, K. S., & Downing, P. E. (2003). Viewpoint-Specific Scene Representations in Human Parahippocampal Cortex. *Neuron, 37*, 865-876.

Epstein, R., Harris, A., Stanley, D., & Kanwisher, N. (1999). The Parahippocampal Place Area: Recognition, Navigation, or Encoding? *Neuron, 23*, 115-125.

Epstein, R., & Kanwisher, N. (1998). A cortical representation of the local visual environment. *Nature, 392*(6676), 598-601.

Frank, L. M., & Brown, E. N. W., M. (2000). Trajectory Encoding in the Hippocampus and Entorhinal Cortex. *Neuron, 27*(1), 169-178.

Gagliardo, A., Ioale, P., & Bingman, V. P. (1999). Homing in Pigeons: The Role of the Hippocampal Formation in the Representation of Landmarks Used for Navigation. *The Journal of Neuroscience, 19*(1), 311-315.

Golledge, R. G. (Ed.). (1999). *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes*. Baltimore & London: The Johns Hopkins University Press.

Gouteux, S., & Spelke, E. S. (2001). Children's use of geometry and landmarks to reorient in an open space. *Cognition, 81*, 119-148.

Gouteux, S., Thinus-Blanc, C., & Vauclair, J. (2001). Rhesus monkeys use geometric and non geometric during a reorientation task. *Journal of Experimental Psychology: General, 130*(3), 505-519.

Hafner, V. V. (2000a). Cognitive maps for navigation in open environments. In *Sixth International Conference on Intelligent Autonomous Systems (IAS-6)* (pp. 801-808). Venice: IOS Press.

Hafner, V. V. (2000b). Learning Places in Newly Explored Environments. In B. Meyer, Floreano, Roitblad and Wilson (Ed.), *From Animals to Animats. SAB2000 Proceedings Supplement Book* (pp. 111-120).

Hermer, L., & Spelke, E. (1994). A geometric process for spatial reorientation in young children. *Nature, 370*, 57-59.

Hermer, L., & Spelke, E. (1996). Modularity and development: The case of spatial reorientation. *Cognition, 61*(3), 195-232.

Huttenlocher, J., Hedges, L. V., & Duncan, S. (1991). Categories and particulars: Prototype effects in estimating spatial location. *Psychological Review, 98*, 352-376.

Jefferies, M. E. (1999). *Cognitive Maps: Understanding How Local Environments are Computed.* Doctor of Philosophy, University of Otago, Dunedin.

Jefferies, M. E., Cree, M., Mayo, M., & Baker, J. T. (2004). Using 2D and 3D Landmarks to Solve the Correspondence Problem in Cognitive Robot Mapping. In *Spatial Cognition IV. Reasoning, Action, and Interaction* (Vol. 3343/2005, pp. 434-454). Berlin / Heidelberg: Springer.

Jefferies, M. E., Weng, W., & Baker, J. T. (2008). Robot Cognitive Mapping - A Role for a Global Metric Map in a Cognitive Mapping Process. In M. E. Jefferies & W.-K. Yeap (Eds.), *Robotics and Cognitive Approaches to Spatial Mapping* (Vol. 38, pp. 664-672): Springer-Verlag, Berlin.

Jefferies, M. E., Weng, W., Baker, J. T., & Mayo, M. (2004). Using Context to Solve the Correspondence Problem in Simultaneous Localisation and Mapping. In C. Zhang, H. W. Guesgen & W.-K. Yeap (Eds.), *Proceedings of the Eighth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2004)* (pp. 664-672). Auckland, New Zealand: Springer-Verlag, Berlin.

Jefferies, M. E., & Yeap, W. K. (Eds.). (2008). *Robot and Cognitive Approaches to Spatial Mapping*. Berlin / Heidelberg: Springer Verlag.

Jefferies, M. E., Yeap, W. K., Cosgrove, M. C., & Baker, J. T. (2005). Using Absolute Metric Maps to Close Cycles in a Topological Map. *Journal of Intelligent Manufacturing, 16*(6), 693-702.

Jeffery, K. J., Gilbert, A., Burton, S., & Strudwick, A. (2003). Preserved Performance in a Hippocampal-Dependent Spatial Task Despite Complete Place Cell Remapping. *Hippocampus, 13*(2), 175–189.

Kahler, O., & Denzler, J. (2005). Self-organizing, adaptive data fusion for 3d object tracking. In *ARCS 2005 - Organic and Pervasive Computing, Workshop Proceedings* (pp. 109–116). Innsbruck, Austria.

Kahler, O., Denzler, J., & Triesch, J. (2004). Hierarchical Sensor Data Fusion by Probabilistic Cue Integration for Robust 3-D Object Tracking. In *IEEE Southwest Symp. on Image Analysis and Interpretation* (pp. 216–220). Nevada.

Kelly, D. M., & Spetch, M. L. (2001). Pigeons encode relative geometry. *Journal of Experimental Psychology: Animal Behavior Processes, 27*(4), 417-422.

Kelly, D. M., & Spetch, M. L. (2004a). Reorientation in a Two-Dimensional Environment: I. Do Adults Encode the Featural and Geometric Properties of a Two-Dimensional Schematic of a Room? *Journal of Comparative Psychology, 118*(1), 82–94.

Kelly, D. M., & Spetch, M. L. (2004b). Reorientation in a Two-Dimensional Environment: II. Do Pigeons (*Columba livia*) Encode the Featural and Geometric Properties of a Two-Dimensional Schematic of a Room? *Journal of Comparative Psychology, 118*(4), 384-395.

Kitchin, R., & Freundschuh, S. (Eds.). (2000). *Cognitive Mapping: Past, Present and Future*. London: Routledge.

Kobayashi, T., Tran, A. H., Nishijo, H., Ono, T., & Matsumoto, G. (2003). Contribution of hippocampal place cell activity to learning and formation of goal-directed navigation in rats. *Neuroscience, 117*(4), 1025-1035.

Konolige, K. (2004). Large-scale map-making. In *Proceedings of the National Conference on AI (AAAI)* (pp. 457-463). San Jose, CA: AAAI Press / The MIT Press.

Kortenkamp, D. (1993). *Cognitive maps for mobile robots: A representation for mapping and navigation.* Doctor of Philosophy, University of Michigan.

Kortenkamp, D., & Weymouth, T. (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)* (pp. 979-984). Seattle, Washington.

Kuipers, B. (2008). An intellectual history of the Spatial Semantic Hierarchy. In M. E. Jefferies & W. K. Yeap (Eds.), *Robot and Cognitive Approaches to Spatial Mapping* (pp. 243-264). Berlin / Heidelberg: Springer Verlag.

Kuipers, B., Modayil, J., Beeson, P., MacMahon, M., & Savelli, F. (2004). Local metrical and global topological maps in the hybrid Spatial Semantic Hierarchy.

In *IEEE International Conference on Robotics and Automation (ICRA-04)* (Vol. 5, pp. 4845- 4851). New Orleans, Louisiana, USA.

Kuipers, B. J. (1977). *Representing Knowledge of Large-Scale Space.* Doctor of Philosophy, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Kuipers, B. J. (1978). Modeling spatial knowledge. *Cognitive Science, 2*, 129-153.

Kuipers, B. J. (2000). The spatial semantic hierarchy. *Artificial Intelligence, 119*(1-2), 191 - 233.

Kuipers, B. J., & Byun, Y.-T. (1987). A qualitative approach to robot exploration and map-learning. In *AAAI Workshop on Spatial Reasoning and Multi-Sensor Fusion* (pp. 390-404). St. Charles, Illinois.

Kuipers, B. J., & Byun, Y.-T. (1988). A robust qualitative method for spatial learning in unknown environments. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-88)* (pp. 774-775). Los Altos, CA.

Kuipers, B. J., & Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems, 8*, 47-63.

Learmonth, A. E., Nadel, L., & Newcombe, N. S. (2002). Children's Use of Landmarks: Implications for Modularity Theory. *Psychological Science, 4*, 337-341.

Lee, W. Y. (1996). *Spatial Semantic Hierarchy for a Physical Mobile Robot.* Doctor of Philosophy, The University of Texas at Austin, Austin, Texas.

Levenick, J. R. (1985). *Knowledge representation and intelligent systems: from semantic networks to cognitive maps.* Doctor of Philosophy, The University of Michigan, Ann Arbor, Michigan.

Levenick, J. R. (1991). NAPS: A connectionist implementation of cognitive maps. *Connection Science, 3*(2), 107-126.

Lever, C., Wills, T., Cacucci, F., Burgess, N., & O'Keefe, J. (2002). Long-term plasticity in hippocampal place-cell representation of environmental geometry. *Nature, 416*, 90-94.

Maguire, E. A., Frackowiak, R. S. J., & Frith, C. D. (1997). Recalling Routes around London: Activation of the Right Hippocampus in Taxi Drivers. *The Journal of Neuroscience, 17*(18), 7103-7110.

Maguire, E. A., Gadian, D. G., Johnsrude, I. S., Good, C. D., Ashburner, J., Frackowiak, R. S. J., et al. (2000). Navigation-related structural change in the hippocampi of taxi drivers. *Proceedings of the National Academy of Sciences, 97*(8), 4398–4403.

McGregor, A., Hayward, A. J., Pearce, J. M., & Good, M. A. (2004). Hippocampal Lesions Disrupt Navigation Based on the Shape of the Environment. *Behavioral Neuroscience, 118*(5), 1011-1021.

Memories. (2006). from http://the-half-decent-pharmaceutical-chemistry-blog.chemblogs.org/archives/2006/10/18/memories

Milford, M., & Wyeth, G. (2007). Spatial Mapping and Map Exploitation: A Bio-inspired Engineering Perspective. In S. Winter, M. Duckham, L. Kulik & B. Kuipers (Eds.), *Spatial Information Theory, 8th International Conference, COSIT 2007* (Vol. 4736, pp. 203-221). Melbourne, Australia: Springer.

Milford, M. J., & Wyeth, G. (2003). Hippocampal Models for Simultaneous Localisation and Mapping on an Autonomous Robot. In *Australian Conference on Robotics and Automation*. Brisbane, Australia.

MobileRobots Inc. from http://www.mobilerobots.com/

MobileRobots Inc. (2003). *Pioneer 3™ & Pioneer 2™ H8-Series Operations Manual*.

Modayil, J., Beeson, P., & Kuipers, B. (2004). Using the topological skeleton for scalable global metrical map-building. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-04)* (pp. 1530 - 1536). Sendai, Japan.

Müller, M., & Wehner, R. (1994). The hidden spiral: systematic search and path integration in desert ants, *Cataglyphis fortis Journal of Comparative Physiology A, 175*(5), 525-530.

Newman, P., Leonard, J., Tardds, J., & Neira, J. (2002). Explore and Return: Experimental Validation of Real-Time Concurrent Mapping and Localization. In *International Conference on Robotics and Automation* (pp. 1802–1809). Washington, DC.

Niemann, H. (1990). *Pattern Analysis and Understanding* (2nd ed. Vol. 4). Berlin, Germany: Springer.

O'Keefe, J., & Dostrovsky, J. (1971). The hippocampus as a spatial map: preliminary evidence from unit activity in the freely moving rat. *Brain Research, 34*, 171–175.

O'Keefe, J., & Nadel, L. (1978). *The hippocampus as a cognitive map*. Oxford: Clarendon Press.

Pavlidis, T., & Horowitz, S. L. (1974). Segmentation of Plane Curves. *IEEE Transaction on Computers, C-23*, 860 – 870.

Pearce, J. M., George, D. N., Haselgrove, M., Erichsen, J. T., & Good, M. A. (2005). The Influence of Hippocampal Lesions on the Discrimination of Structure and on Spatial Memory in Pigeons (*Columba livia*). *Behavioral Neuroscience, 119*(5), 1316-1330.

Pearce, J. M., Good, M. A., Jones, P. M., & McGregor, A. (2004). Transfer of Spatial Behavior Between Different Environments: Implications for Theories of Spatial Learning and for the Role of the Hippocampus in Spatial Learning. *Journal of Experimental Psychology, 30*(3), 135-147.

Portugali, J. (Ed.). (1996). *The Construction of Cognitive Maps*: Springer Netherlands.

Prasser, D., Milford, M., & Wyeth, G. (2005). Outdoor Simultaneous Localisation and Mapping using RatSLAM. In *International Conference on Field and Service Robotics* (pp. 143-154). Port Douglas, Australia.

Prasser, D. P., Wyeth, G. F., & Milford, M. J. (2004). Experiments in outdoor operation of RatSLAM. In N. Barnes & D. Austin (Eds.), *2004 Australasian Conference on Robotics and Automation* (pp. 1-6). Canberra, Australia.

Redish, A. D. (Ed.). (1999). *Beyond the Cognitive Map: From Place Cells to Episodic Memory*: Massachusetts Institute of Technology, Massachusett.

Remolina, E. (2001). *A Logical Account of Causal and Topological Maps.* Doctor of Philosophy, The University of Texas at Austin, Austin, Texas.

Rivard, B., Li, Y., Lenck-Santini, P. P., Poucet, B., & Muller, R. U. (2004). Representation of Objects in Space by Two Classes of Hippocampal Pyramidal Cells. *The Journal of General Physiology, 124*, 9-25.

Rosenbaum, R. S., Gao, F., Richards, B., Black, S. E., & Moscovitch, M. (2005). "Where to?" Remote Memory for Spatial Relations and Landmark Identity in Former Taxi Drivers with Alzheimer's Disease and Encephalitis. *Journal of Cognitive Neuroscience, 17*(3), 446-462.

Schmidt, J., Wong, C. K., & Yeap, W. K. (2007). Spatial Information Extraction for Cognitive Mapping with a Mobile Robot. In *Conference on Spatial*

Information Theory: COSIT'07 (Vol. 4736 of Lecture Notes in Computer Science, pp. 186-202). Melbourne, Australia: Springer-Verlag.

Simmons, R., & Koenig, S. (1995). Probabilistic Robot Navigation in Partially Observable Environments. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1080-1087). Montreal, Canada.

Sovrano, V. A., Bisazza, A., & Vallortigara, G. (2002). Modularity and spatial reorientation in a simple mind: encoding of geometric and nongeometric properties of a spatial environment by fish. *Cognition, 85*, 51-59.

Sovrano, V. A., Bisazza, A., & Vallortigara, G. (2003). Modularity as a Fish (*Xenotoca eiseni*) Views It: Conjoining Geometric and Nongeometric Information for Spatial Reorientation. *Journal of Experimental Psychology, 29*(3), 199-210.

Sovrano, V. A., Bisazza, A., & Vallortigara, G. (2005). Animal's use of landmarks and metric information to reorient: effects of the size of the experimental space. *Cognition, 97*, 121-133.

Sovrano, V. A., Bisazza, A., & Vallortigara, G. (2006). How fish do geometry in large and in small spaces. *Animal Cognition, 10*(1), 47-54.

Sovrano, V. A., & Vallortigara, G. (2006). Dissecting the Geometric Module: A Sense Linkage for Metric and Landmark Information in Animals' Spatial Reorientation. *Psychological Science, 17*(7), 616-621.

Tang, X., Garrett, D., & von der Malsburg, C. (2005). Self-Organized Figure-Ground Segmentation by Multiple-Cue Integration. In *Proceedings of the International Conference on Signal and Image Processing (SIP)* (pp. 376–382). Honolulu, USA.

Tardos, J., Neira, J., Newman, P., & Leonard, J. (2002). Robust mapping and localization in indoor environments using sonar data. *The International Journal of Robotics Research, 21*(4), 311-330.

Thrun, S. (2008). Simultaneous localization and mapping. In M. E. Jefferies & W. K. Yeap (Eds.), *Robot and Cognitive Approaches to Spatial Mapping* (pp. 13-41). Berlin / Heidelberg: Springer Verlag.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review, 55*(4), 189-208.

Tomatis, N., Nourbakhsh, I., & Siegwart, R. (2002). Hybrid Simultaneous Localization and Map Building: Closing the Loop with Multi-Hypotheses Tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2002)* (Vol. 3, pp. 2749 - 2754). Washington DC, USA.

Tommasi, L., Gagliardo, A., Andrew, R. J., & Vallortigara, G. (2003). Separate processing mechanisms for encoding of geometric and landmark information in the avian hippocampus. *European Journal of Neuroscience, 17*(8), 1695-1702.

Tommasi, L., & Polli, C. (2004). Representation of two geometric features of the environment in the domestic chick (*Gallus gallus*). *Animal Cognition, 7*(1), 53-59.

Triesch, J. (1999). *Vision-Based Robotic Gesture Recognition*. Aachen: Shaker Verlag.

Triesch, J., & von der Malsburg, C. (2001). Democratic Integration: Self-Organized Integration of Adaptive Cues. *Neural Computation, 13*(9), 2049–2074.

Vallortigara, G., Feruglio, M., & Sovrano, V. A. (2005). Reorientation by geometric and landmark information in environments of different size. *Developmental Science, 8*(5), 393 –401.

Vargas, J. P., Petruso, E. J., & Bingman, V. P. (2004). Hippocampal formation is required for geometric navigation in pigeons. *European Journal of Neuroscience, 20*, 1937–1944.

Waller, D., Loomis, J. M., Golledge, R. G., & Beall, A. C. (2000). Place learning in humans: The role of distance and direction information. *Spatial Cognition and Computation, 2*(4), 333-354.

Wehner, R. (1983). Celestial and terrestrial navigation: Human strategies - insect strategies. In F. Huber & H. Markl (Eds.), *Neuroethology and Behavioral Physiology* (pp. 366-381). Berlin Heidelberg New York: Springer.

Wehner, R., Gallizzi, K., Frei, C., & Vesely, M. (2002). Calibration processes in desert ant navigation: vector courses and systematic search. *Journal of Comparative Physiology A, 188*(9), 683-693.

Wehner, R., & Srinivasan, M. V. (1981). Searching behaviour of desert ants, *genusCataglyphis* (*Formicidae, Hymenoptera*). *Journal of Comparative Physiology A, 142*(3), 315-338.

Wehner, R., & Srinivasan, M. V. (2003). Path integration in insects. In K. J. Jeffery (Ed.), *The Neurobiology of Spatial Behaviour* (pp. 9-30). Oxford: Oxford University Press.

Wehner, R., & Wehner, S. (1986). Path integration in desert ants. Approaching a long-standing puzzle in insect navigation. *Monitore zoologico italiano, 20*(3), 309-331.

White, A. R., Strasser, R., & Bingman, V. P. (2002). Hippocampus Lesions Impair Landmark Array Spatial Learning in Homing Pigeons: A Laboratory Study. *Neurobiology of Learning and Memory, 78*(1), 65–78.

Wong, C. K., Schmidt, J., & Yeap, W. K. (2007). Using a Mobile Robot for Cognitive Mapping. In *International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 2243-2248). Hyderabad, India.

Wong, C. K., Yeap, W. K., & Sapiyan, M. (2005). A Mobile Robot that Maps Naively But Plans Intelligently. In *Proceedings of the Artificial Intelligence and Application Conference* (pp. 562-567). Innsbruck, Austria.

Yeap, W. K. (1988). Towards a Computational Theory of Cognitive Maps. *Artificial Intelligence, 34*(3), 297-360.

Yeap, W. K. (2007). From Spatial Perception to Cognitive Mapping: How is the Flow of Information Controlled? In H. Schultheis, T. Barkowsky, B. Kuipers & B. Hommel (Eds.), *Control Mechanisms for Spatial Knowledge Processing in Cognitive / Intelligent Systems* (pp. 59-61). Menlo Park, California: AAAI Press.

Yeap, W. K., & Jefferies, M. E. (1999). Computing a Representation of the Local Environment. *Artificial Intelligence, 107*(2), 265-301.

Yeap, W. K., & Jefferies, M. E. (2000). On early cognitive mapping. *Spatial Cognition and Computation, 2*(2), 85-116.