School of Computing

Visually Guided Autonomous Robot Navigation: An Insect Based Approach

Keven Weber

This thesis is presented as part of the requirements for
the award of the Degree of Doctor of Philosophy
of the
Curtin University of Technology

August, 1998

Abstract

Giving robots the ability to move around autonomously in various real-world environments has long been a major challenge for Artificial Intelligence. New approaches to the design and control of autonomous robots have shown the value of drawing inspiration from the natural world. Animals navigate, perceive and interact with various uncontrolled environments with seemingly little effort. Flying insects, in particular, are quite adept at manoeuvring in complex, unpredictable and possibly hostile environments.

Inspired by the miniature machine view of insects, this thesis contributes to the autonomous control of mobile robots through the application of insect-based visual cues and behaviours. The parsimonious, yet robust, solutions offered by insects are directly applicable to the computationally restrictive world of autonomous mobile robots. To this end, two main navigational domains are focussed on: corridor guidance and visual homing.

Within a corridor environment, safe navigation is achieved through the application of simple and intuitive behaviours observed in insect, visual navigation. By observing and responding to observed apparent motions in a reactive, yet intelligent way, the robot is able to exhibit useful corridor guidance behaviours at modest expense. Through a combination of both simulation and real-world robot experiments, the feasibility of equipping a mobile robot with the ability to safely navigate in various environments, is demonstrated.

It is further shown that the reactive nature of the robot can be augmented to incorporate a map building method that allows previously encountered corridors to be recognised, through the observation of landmarks en route. This allows for a more globally-directed navigational goal.

Many animals, including insects such as bees and ants, successfully engage in visual homing. This is achieved through the association of visual landmarks with a specific location. In this way, the insect is able to 'home in' on a previously visited site by simply moving in such a way as to maximise the match between the currently observed environment and the memorised 'snapshot' of the panorama as seen from the goal. A mobile robot can exploit the very same strategy to simply and reliably return to a previously visited location.

This thesis describes a system that allows a mobile robot to home successfully.

Specifically, a simple, yet robust, homing scheme that relies only upon the observation of the bearings of visible landmarks, is proposed. It is also shown that this strategy can easily be extended to incorporate other visual cues which may improve overall performance.

The homing algorithm described, allows a mobile robot to home incrementally by moving in such a way as to gradually reduce the discrepancy between the current view and the view obtained from the home position. Both simulation and mobile robot experiments are again used to demonstrate the feasibility of the approach.

Preface

The original work presented in chapters 4, 5, and 6, of this thesis has previously been published in various forms including book chapters, refereed journal articles and refereed conference papers.

The corridor guidance chapter presents research conducted during the period 1995–1996. This has been published in: *Proceedings of the Asian Conference on Computer Vision* (Chahl et al., 1995), *Proceedings of the International Conference on Pattern Recognition* (Weber et al., 1996a), and *From Living Eyes to Seeing Machines*, Chapter 11 (Weber et al., 1997).

The corridor discrimination chapter describes work conducted and completed in 1996. This has been published in: *Proceedings of the International Conference on Automation, Robotics and Computer Vision* (Weber et al., 1996b).

The most recent body of research was conducted between 1997 and 1998 on robot homing. This has been published in: *Proceedings of the International Conference on Pattern Recognition* (Weber et al., 1998), and *Adaptive Behavior* (Weber et al., 1999).

The corridor guidance research has also appeared in several review articles on the application of insect-inspired techniques to robot navigation: *Computational Intelligence: A Dynamic System Perspective* (Srinivasan et al., 1995), *Proceedings of the International Conference on Field and Service Robotics* (Srinivasan et al., 1997), and *Robotics and Autonomous Systems* (Srinivasan et al., 1998).

Acknowledgments

Firstly, warmest thanks must go to Svetha for her supervisory prowess, support, guidance, overall encouragement, and her patience in awaiting perfected drafts.

Many thanks must also go to Srini for his encouragement, attention to detail, and for his assistance in injecting the right amount of biological and ethological ingredients into the mixture. Thanks also to Javaan for his help in the design and construction of *Kenneth* the robot.

Thanks to Tony for suitably distracting me, and for playing devil's advocate in proof reading the final draft. Thanks also to Kim, especially for his helpful, and sometimes not so helpful, LaTeX advice.

A special thanks must also go to the callipygious Rochelle for her linguistic advice, proof reading skills, and above all everything else.

Last, but not least, a special thanks should also go to Kenneth for just being there...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

No sensible person will deny that the works of Nature are in the highest degree simple, necessary and as economical as possible. Therefore machines devised by mankind will doubtlessly likewise attain most success if they are as far as possible modelled on works of Nature.

Giovanni Alfonso Borelli, 1680
(Italian mathematician and astronomer)
De moto animalium

Getting robots to navigate autonomously in the real world has proven to be a difficult task. Despite 30 years of intensive research and great advances in technology and computational power, only modest advances have been made towards the goal of constructing truly autonomous machines, capable of intelligently interacting with the world to perform useful tasks. One of the main stumbling blocks, has been the inability to instruct a machine to perceive, reason and interact with an inherently complex and unpredictable world in both a robust and efficient manner.

When designing a machine to perform a difficult task, it can help to investigate and draw inspiration from another machine – artificial or living – that can also perform the task. However, trying to copy how humans perceive, understand, and reason about the world is not an easy undertaking. Human-level cognition is very complex and remains little understood. A natural alternative is to examine how simpler animals solve the problems of navigating in unknown environments. It is intriguing, that small living organisms, such as insects, have evolved effective solutions to this problem despite having relatively simple nervous systems and restricted processing capacities. The ability to both robustly and efficiently

perceive and react to the real world has essentially been a requisite for survival. Given this, it seems likely that insects employ 'short cuts' to navigate in the real world. It is these principles which may be exploited to aid the development of artificial autonomous machines.

Invertebrates, such as insects, are particularly attractive in the present context by virtue of their pervasiveness in virtually all earthly environments, their rich and diverse behaviour, their relative simplicity, and their ease of study. Despite possessing small brains, insects exhibit many of the desired capabilities requested of autonomous robots, such as highly skilled sensorimotor behaviour, efficient communication, learning and cooperation. Insects are thus highly analogous and applicable to autonomous robotics.

Insects may well be described as the ultimate miniature machines, having evolved physical and neural hardware that, by virtue of its elegance and robustness, has withstood the test of time and endured the hazards and uncertainties of the environment. Inspired by this miniature machine view of insects this thesis endeavours to transfer some of the successes of insect navigation to the world of mobile robotics.

1.1 Aims and Approaches

The main aim of this thesis is to show that the guiding principles of autonomy and navigation as observed in the natural world, specifically through the study of insects, are of value in solving some of the real-time, real-world issues of autonomy and navigation in mobile robotics.

Rather than following traditional anthropomorphic or engineering-based approaches, this thesis will explore a subset of navigational strategies that are inspired by findings in insect ethology. The underlying rationale is the expectation that the parsimonious solutions offered by insects can be usefully incorporated into algorithms for robot navigation. The main aim is to avoid needless complexity and improve robustness by concentrating on only the key qualitative elements of successful navigation. Such solutions, can be of extreme benefit in the computationally restrictive world of real-time autonomous robot navigation.

To this end, an exploration of the applicability of various insect-inspired visual guidance behaviours, to the autonomous control of a mobile robot is undertaken. Through a combination of both simulation and real-world experiments it will be

shown that various simple visual cues and behaviours observed in insects can be advantageously applied to autonomous robot navigation.

1.2 Contribution

The major contributions of this thesis involve the application of various visually-driven, insect-inspired behaviours, to the efficient, robust and autonomous control of a mobile robot.

Firstly, it is shown that very simple motion cues and behaviours, inspired by the visual navigation of flying insects, can be used profitably to provide a mobile robot with the ability to traverse a corridor environment. Specifically, it is shown how relative motion cues can be utilised to efficiently drive useful robot behaviour. Apparent motion cues are used to: (i) gauge range, (ii) keep lateral walls equidistant, (iii) slow down when approaching an object, (iv) avoid collisions, (v) regulate forward speed, and (vi) provide visual odometry.

Secondly, a method is proposed for augmenting the reactive nature of the robot through the incorporation of a 'map' building regime that provides for a route recognition capability. By observing landmarks en route, a previously encountered corridor can be recognised as such. This knowledge can consequently be used to help direct a more globally-oriented navigational goal.

Thirdly, a qualitative, landmark-based, robot homing strategy, inspired by the visual homing behaviour of bees and ants, is presented. Despite being simple, intuitive and computationally cheap, the homing technique provides robust homing, superior to that of previous approaches. Landmark avoidance characteristics are also shown to be implicit in the algorithm. Several simple, approximate, and computationally efficient landmark correlation techniques are explored, and are shown to be sufficient for reliable homing. Both qualitative and quantitative investigations of various landmark correspondence techniques and homing methods are made. It is shown that a robot can reliably return to a previously visited site using only a parsimonious representation of the panoramic environment. This is performed in a continuous and reactive fashion, without the need for any explicit 3D modelling or reconstruction of the environment. The homing behaviour is predominantly driven directly by visual stimuli.

1.3 Thesis Outline

This thesis is organised as follows:

Chapter 2 presents a survey of the background literature relevant to the work presented in the following chapters, and also canvases some of the philosophical issues concerning the construction of autonomous and intelligent machines.

Chapter 3 describes the specifics of the mobile robot that was employed in the various real-world experiments throughout the thesis.

Chapter 4 describes an approach to visually guided autonomous robot navigation, inspired by insect visual behaviour. Various visual behaviours, based on relative motion cues, are used to control the mobile robot as it traverses corridor-type environments. In each case, however, there is no need for a 3D reconstruction of the environment, the robot behaviours are driven directly by visual motion cues elicited when the robot is in motion.

Chapter 5 describes an attempt to provide the robot, equipped with very reactive corridor guidance behaviours, with a higher-level ability to recognise, and discriminate between, previously traversed corridor environments. In this way, it is shown how the purely reactive robot can augment, in a relatively seamless fashion, its knowledge of its environment in a way to obtain a more global view, suitable for higher-level navigational goals.

Chapter 6 presents a qualitative landmark-based local homing technique, inspired by visual homing in bees and ants. Utilising very simple landmark-based cues a mobile robot is capable of reliably returning to a previously visited location by simply moving continuously so as to make the current panorama more similar to the panorama seen from that prior location.

Chapter 7 presents a summary of the main contributions of this thesis and some final comments.

In addition to these previous chapters, two additional Appendices have been included:

Appendix A presents a more comprehensive set of real-world corridor guidance results. These provide a better impression of the typical robot behaviour experienced within the context of the experiments described in chapter 4.

Appendix B presents some of the miscellaneous homing results and source data. Also presented, is a more detailed comparison of an image-based approach to visual homing and the landmark-based approach presented in chapter 6.

Chapter 2

Background

This chapter attempts to canvas the majority of the background literature that is directly relevant to work presented in the following chapters, and also discusses the philosophical underpinnings of the major theme of the thesis. In this way a metaphorical picture will be painted of autonomous robot navigation within which the research presented in this thesis contributes.

The format of this chapter is as follows: Firstly, the basic background of autonomous robot navigation is explored (§2.1). This will cover the criticisms and perceived failures of traditional approaches, which have in turn led to the development of several new design philosophies, such as behaviour-based robotics (§2.1.3) and biologically-inspired robotics (§2.2). Following this, several aspects of robot vision are discussed (§2.3). The next major section (§2.4) will outline the rather large body of work dealing with the subject of determining image motion from a sequence of camera images. Image motion can elicit a great amount of structural information about the surrounding environment. Following this, the subject of insect ethology will be presented (§2.5). This will cover the visual behaviours directly relevant to the later chapters, but also give a more general feel for the subject of visually-guided autonomous navigation from the insect's point of view. The final two sections (§2.6 and §2.7) will cover the background literature pertaining directly to the proceeding chapters on corridor guidance and robot homing, respectively.

2.1 Autonomous Robot Navigation

Building truly independent robots capable of intelligent and useful behaviour has long been a major goal of Artificial Intelligence (AI) research.

From the earliest days of science fiction and AI the ideal of constructing the quintessential artificial human being, capable of possessing all of the nuances and qualities associated with a thinking, reasoning, and feeling being, has been successfully portrayed as part of the natural evolution of artificial computation.

Classic fictional anthropomorphic robots such as *C3PO* from *Star Wars* and *Data* from *Star Trek: TNG*, despite their characteristic idiosyncrasies, are able to negotiate and interact with the world in a very general and human manner. They are capable of seeing, perceiving, reasoning and moving about the world with the skill and purpose of a human.

However, the great chasm between science fiction and reality, in this regard, remains undiminished. Despite the 30 years of AI research, intelligent real-time control of autonomous mobile robots operating in unstructured domains still proves to be a very difficult problem to solve.

Emulating a human's ability to understand, and intelligently interact and deal with the inherent richness and complexity of the unconstrained real world has proven to be much more difficult than first thought. As it is still as yet unknown exactly how the human mind and brain actually function, perhaps it is not too surprising that attempts to emulate human 'thought processes' and thus human behaviour are meeting with limited success. Although computers are well suited for solving certain classes of problems, such as numerical problems, they are not at all well suited to performing some of the tasks that we humans ordain to be, more 'intelligent'. Some of the tasks humans appear to perform very easily, such as understanding and recognising what we perceive,[1] are exceedingly difficult to instruct an artificial machine to do. Similarly, in the world of autonomous robot navigation it has been difficult to instruct a machine to perceive and understand its environment in a robust human-like way, as a prerequisite for human-level intelligent behaviour. The attempted codification of perceived human-level perceptual understanding and reasoning does not yet appear to be a very fruitful methodology for the design of autonomous agents capable of surviving in the real world for extended periods of time.

[1]For a human, recognising a familiar face or facial expression, for example, takes a mere fraction of a second.

In the last decade, however, there has been a significant diversion from the traditional anthropomorphic or engineering based approaches toward new approaches inspired by simpler biological organisms. The main underlying rationale of this approach is the fact that autonomy and intelligent behaviour are also exhibited by much simpler animals, and hence intuitively are more likely to be more easily understood and easier to emulate. Understanding the principles which underlie robust autonomy in simpler organisms, may also provide clues and insights into the requirements of the successful autonomy and 'higher-level' intelligence, as exhibited by humans.

The main dissatisfaction with traditional robotics and AI is its apparent inability to deliver robust real-time autonomous performance in an unstructured, dynamic world. Despite the constant advances in computer hardware, in terms of price, speed and size, real-time robotics still operates in a relatively computationally restrictive environment.

By drawing inspiration from the neurobiology, physiology, ethology, and sociology of relatively simple animals the new approaches hope to solve the major difficulty of dealing intelligently with the real-world in real-time. These approaches usually involve a tight coupling of sense and action, with a minimum or parsimonious representational view of the world. By concentrating on only the 'important' elements of successful autonomous behaviour, as is exhibited by relatively simple organisms, one is able to reduce some of the inherent complexities assumed by traditional approaches, and produce both efficient and robust solutions. Rather than have a traditional modularised process involving perception, world modelling, planning, and action, new approaches attempt to use the sensed world in a more direct and decentralised way to guide appropriate behaviour. The paradigm shift is essentially, away from the traditional reductionistic *sense-model-plan-act* architecture towards a more behavioural *sense-react* style of operation.

2.1.1 Traditional robotic navigation

Two of the earliest attempts at autonomous robot navigation epitomise the traditional approach and consequently set the tone for the next two decades of robotics research.

The foundations of the traditional approach to autonomous robotics were first laid down in the late 1960s and early 1970s with the development of the mobile

robot *Shakey* (Nilsson, 1984), at the Stanford Research Institute.[2] Shakey's task was to autonomously navigate within a specially prepared environment. The environment consisted of several well-illuminated rooms which were empty except for several large, coloured blocks and wedges which acted as obstacles. Equipped with a single onboard black-and-white camera and touch sensors, Shakey's objective was to satisfy a goal provided by an outside human observer. Depending on the prescribed goal the robot was to be able to autonomously navigate to a desired destination, navigate around obstacles, push obstacles out of the way, and move an obstacle to a new desired location.

Shakey succeeded in performing these tasks by following a computed plan of action based on an internal 3D model of the world described and prescribed by the visual scene. The images obtained from the camera were transmitted to, and analysed by, an off-board computer. A logical description of what was seen by the camera was then incorporated into a symbolic representation of the world. A planning algorithm, STRIPS, then used this current world model[3] to generate a sequence of physical actions.

The relative success of Shakey, however, was primarily due to its carefully engineered environment. The environment was uniformly coloured and contained only two classes of objects, blocks and wedges, making a world model representation much easier to construct and maintain. In this 'semi-artificial' world, Shakey needed only to concern itself with recognising and distinguishing between two obstacles, blocks and wedges. Each object was also carefully constructed to have differently coloured facets. This eased surface edge detection, aiding both object and pose recognition. The numbers of obstacles was also kept small to reduce problems caused by visual occlusion.

By constraining the working environment to fall within acceptable limits it was thus possible to design a viable system for autonomous navigation. Possessing and maintaining such a perfect world model is, however, very unrealistic when dealing with more general, real-world environments and conditions.

Robotics and AI had, nonetheless, adopted the idea of building a complete three-dimensional model of the world, with which one could reason about the world and one's physical interaction with it. The role of computer vision, within this system, was thus to recover the three-dimensional environment that produced

[2]Now SRI International.

[3]Based on first order predicate calculus.

the two-dimensional image. Running under the assumption that this can be realised, this then led to the standardisation and study of several subproblems, such as generating a planned, collision-free, path through the model, which an autonomous vehicle could successfully follow (Lozano-Pérez and Wesley, 1979; Yap, 1987). The real value of these reductionist methods to the overall goal of producing a viable system capable of real-time autonomous and intelligent behaviour, is debatable. Brooks (1986a; 1991c; 1991b), for example, argues that this 'subprobleming' has distracted attention from the real issues involved, and thus will lead to the solving of irrelevant problems and not the eventual creation of an artificial being.

Around the early 1970s, another interesting project was under way at the Massachusetts Institute of Technology (MIT). The purpose of the MIT robot (Winston, 1972), was to view a *toy world* scene, consisting entirely of a set of stacked wooden blocks, and make a copy of the scene using a robotic arm. Although the programs to do this were very specific to the simple and precisely controlled blocks world, this again reinforced the idea of producing a 3D description from a 2D scene. This also reinforced the reductionistic idea of separating and modularising perception, modelling and reasoning, and action. Assuming 'human-level' perception was a realisable and separate task, this then also legitimised the work of others whose programs reasoned about perfect internal representations of the world. The toy blocks world became a popular AI domain in this respect (Winograd, 1972; Dreyfus, 1981). Involving simple and uniform semantics, the state of this world was represented explicitly and completely, making AI reasoning fairly straight forward.[4]

The traditional role of AI was thus mainly to take descriptions of the world, build an internal representation and using *a priori* knowledge of how the world works, solve problems, make plans, and reason about the world. It was assumed that the abstraction could be connected to the real world without great difficulty. However, it soon became clear that perfect descriptions and models of the real world could not be generated from real sensors.

Another early attempt at building a truly autonomous vehicle was carried

[4]See (Lozano-Pérez et al., 1989) for a more recent example. Although task-level planning is performed with a real robotic manipulator for pick-and-place assembly tasks, the work environment is still precisely modeled. The 3D positions of all the parts within the work area are assumed to be known with a high degree of accuracy.

out at Stanford in the late 1970s (Moravec, 1983). The Stanford Cart[5] was a mobile robot developed to autonomously navigate through cluttered spaces guided only by the input from its on-board camera system. While relaxing the strict environmental requirements observed in the Shakey experiments, the Stanford Cart still functioned under the same traditional *sense-model-plan-act* framework.

The cart used stereopsis to determine the surrounding three-dimensional structure of its environment and the result of its own motion. By moving the camera horizontally, perpendicular to the line of sight, the cart could capture several (nine in all) stereo images. These images were captured at precisely 6.5 cm intervals, allowing range to be deduced from the parallax-induced displacement of image features. From this range information an internal three-dimensional world model was constructed, which was then used to plan an obstacle-avoiding path, through the environment, to the desired new location. The cart could navigate quite successfully for short runs (20 metres), but was however very slow. Navigating through a relatively simple obstacle course, the cart moved, in lurches of approximately 1 metre, every 10 to 15 minutes. After each move a new plan was generated. In this way the cart would sequentially, visually locate obstacles, model its perceived environment, plan a course of action, and finally execute a small part of the plan. This sense-model-plan-act procedure was repeated until the desired new location was reached.

Although providing promising results and improving on the earlier Shakey experiments, the fragility of the cart under real world conditions was still quite apparent. The errors caused by an imperfect sensory description of the world, were compounded by the subsequent augmentation of the internal world model and in combination with the cart's imperfect self-positioning model this eventually caused severe misconceptions about path safety.[6]

Inaccurate sensors, world unpredictability, and imperfect modelling and control all contribute to the failure of traditional planning and navigation systems that rely on complex internal models of an objective external world. Unfortunately, brittleness in the face of unanticipated contingencies and novel situations is still typical of the state of affairs in modern autonomous robotics.

[5]See (Moravec, 1977; Moravec, 1979; Moravec, 1980; Moravec, 1981; Moravec, 1983).

[6]Absolute coordinate systems are a major source of cumulative error and thus contribute to the fragility of purely metric methods.

2.1.2 Reductionism versus Holism

Traditional approaches toward the design of 'true' and complete AI have employed a top-down deconstruction of intelligence through notions of thought and reason. The daunting task of creating an artificial 'human-like' cognitive mind has thus been functionally decomposed and subdivided into smaller, more manageable pieces. It is thought that, eventually, as these pieces to the intelligence puzzle are put together and combined, a truly autonomous and intelligent machine will naturally emerge. One of the criticisms of this approach is that AI tends to gets bogged down in subproblems of intelligence which have debatable relevance to the main goal. Another criticism concerns the gestalt qualities of intelligence and autonomy. There are aspects of the whole which are simply not evident in the sum of its parts.

Opposing this traditional reductionist philosophy, new approaches to designing autonomous robots have been directed towards a more holistic and evolutionary view of intelligence. Essentially, to study and create intelligence from the bottom up. Rather than tackle isolated aspects of human-level intelligence the emphasis is on building completely autonomous, insect-level, intelligent control systems for mobile robots that exist in and interact with the real world.[7]

It has thus been argued that the way forward for both autonomous robotics and AI, is with the construction of robust and completely autonomous creatures or *artificial insects* which are physically grounded in the real world (Brooks, 1986a; Brooks, 1988; Brooks, 1990; Brooks, 1991a). Brooks (1991d) suggests that the new approaches to robotics essentially be concerned with creating robots which are:

(i) embodied — having a direct physical relationship with the real world,

(ii) situated — requiring immediate and direct interaction with the world,[8] and

(iii) autonomous — being separated from any further human input.

Having achieved this complete and robustly autonomous artificial insect, the task is then to incrementally improve the system by incorporating more 'intelligent' capabilities. The creation of highly intelligent beings can now be

[7]This strategy essentially involves an initial trading of computational depth (high-level tasks) for computational width (robust autonomy).

[8]The world is essentially viewed and used as its own model. The system thus continuously refers to its sensors rather than to an internalised world model.

viewed as essentially an evolutionary process. Whilst still maintaining the major tenets of embodied autonomy, the 'creatures' are incrementally improved to incorporate more and more aspects of so called high-level intelligence.[9] Further, the experience and knowledge gained through the study, design, and construction of completely autonomous creatures will assist in the creation of increasingly more complex incarnations. Also, it is perhaps not unreasonable to suggest that some insights into the requirements of higher-level intelligence and autonomy will only emerge through the grounded real-world study of lower-level autonomy.

2.1.3 Behaviour-based Robotics

In the past decade or so, a relatively new approach to designing autonomous agents has been developed, that of *behaviour-based robotics* (Maes, 1990). One of the key tenets of the behaviour-based philosophy is concerned with using the perceived world to direct behaviour rather than using a centralised internal model view of the world. The emphasis is on the intimate and continuous interaction with the environment. Essentially, a more direct linking is sought between perception and action with minimal representation and 'cogitation'. Apart from the antirepresentation bias, other common themes include decentralised control and the incremental layering of competences to provide greater robustness to error and failure.

In the sense that traditional knowledge-based approaches to autonomous control impose a sense-model-plan-act framework, the new *behavioural* approaches propose a more direct sense-react framework, removing, to various degrees, the need to model the world before acting upon it. The question of the extent to which an autonomous robot needs to model or reason about the world, before 'intelligently' interacting with it, has subsequently been queried by several researchers (Brooks, 1991c; Nelson and Aloimonos, 1989; Aloimonos and Rosenfeld, 1991; Brooks, 1991b; Aloimonos, 1992; Ballard and Brown, 1992; Prescott, 1996).

The behaviour-based paradigm has been applied to a variety of tasks, such as

[9]This paradigm shift, can also be viewed as agent-directed improvement as opposed to environment-directed improvement. Rather than building autonomous robots that operate in very constrained worlds (e.g. Shakey) and then go about improving the system by slowly increasing the complexity of the *environment*, the system is initially built to cope with the totally unconstrained real world and then the complexity of the *creature* is gradually increased.

the control of robotic arms (Connell, 1989; Asteroth et al., 1992),[10] underwater vehicles (Bellingham et al., 1990; Bonasso, 1992; Payton et al., 1992) and robots (Brooks, 1997), flying robots (Lewis et al., 1993; Montgomery et al., 1995), legged robots (Brooks, 1989a), planetary exploration robots (Brooks, 1997), and the time-delayed teleoperation of robots (Stein and Paul, 1994). The flying robot (Montgomery et al., 1995), for example, used behaviour-based control with low-level, reflexive behaviours responsible for craft survival and higher-level behaviours responsible for tasks such as navigation and object location.

2.1.3.1 Subsumption Architecture

The traditional AI decomposition for an intelligent control system is to break processing into a hierarchy of sequential information processing modules from sensing to action (fig. 2.1(a)). The subsumption architecture,[11] developed by Brooks (1985; 1986b), proposes a more independently layered decomposition where each behaviour-based layer directly connects sensing to action (fig. 2.1(b)). Within this framework new task-achieving behaviours are added incrementally, as separate layers, to allow the robot to operate at increasing levels of competence. In this way, the architecture is very much based on an evolutionary approach. The control system evolves through the gradual accretion of new layers.

Although dependencies may exist between the layered behaviours, there is no explicit sequential execution of specific layers. While higher layers may subsume lower ones to achieve certain goals, these lower layers still continue to function 'independently'. Behavioural conflicts are resolved through a fixed priority arbitration scheme.

A given layer is realised by processes linking sensing to action through a network of asynchronous, message-passing, augmented finite state machines (AFSMs). The basis of the framework is that a given AFSM can suppress or inhibit the inputs or outputs of another AFSM, for a given time period. In this way, a higher layer can easily be integrated with and grafted onto the current system structure.

Brooks used this behaviour-based architecture to construct several robots which were truly embodied, situated and autonomous. The first of these was

[10] See (Yamauchi and Nelson, 1991) for an interesting behaviour-based robot 'juggling' application.

[11] See also (Connell, 1987; Connell, 1989; Brooks, 1989b; Chatila, 1989; Connell, 1990; Matarić, 1991; Gat et al., 1993; Jones and Flynn, 1993; Lammens et al., 1995).

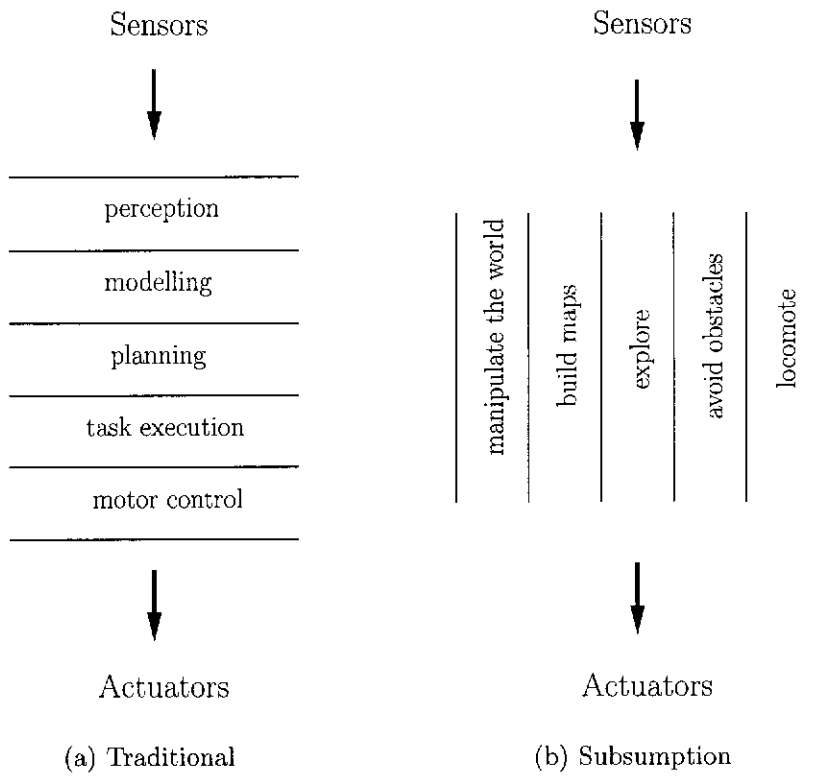


Figure 2.1: The Subsumption Architecture. (a) The traditional hierarchical and functional decomposition of an intelligent control system. The modularised process links sensing to action through a hierarchical chain of information processing modules. (b) The subsumption architecture decomposition is in terms of behaviour generating modules each of which directly connects sensing to action. (Figure adapted from Brooks (1991d).)

the simple robot *Allen* (Brooks, 1986b), which consisted of only two layers. The lower, reactive, layer used sonar readings to avoid static and dynamic obstacles, whilst the upper, non-reactive layer, selected heading directions in which to wander. The successful fusing of both reactive and non-reactive behaviours[12] allowed the robot to successfully explore its environment whilst concurrently avoiding collisions and keeping away from moving obstacles, such as people. A more impressive implementation, however, was provided by the six legged robot *Genghis* (Brooks, 1989a), which was programmed to walk over rough terrain. In this case, the behavioural layers provided first the ability to stand up, then to walk without feedback, then to adjust for rough terrain and obstacles via force feedback, then finally to modulate this accommodation based on pitch and roll sensors. Using only 57 AFSMs the robot successfully navigated rough terrain with little computation, no explicit internal model of the world, and no hierarchical control.

The key consequences of the subsumption approach are:

(i) each layer can deal with a different goal enabling the robot to cope with multiple goals,

(ii) multiple sensors can be used according to the needs of each individual layer without requiring any global representation,

(iii) the system is robust,

(iv) the system is incrementally buildable and testable, and

(v) the system structure is easily extensible. New behavioural layers can be added incrementally, when and if required.

The key differences between this approach and traditional approaches are:

(i) there is no central model[13] of the world explicitly represented within the system.

(ii) there is no separation of data and computation, and

[12]The concept of a two tier system combining both reactive and non-reactive components has become increasingly popular, as the reactive component is seen as a way of dealing with the real-time issues, whilst the more the traditional AI component deals with the long-term planning issues (§2.1.4).

[13]Active (distributed) representation has however subsequently been incorporated into the subsumption architecture (Matarić, 1990; Matarić, 1992).

(iii) there is no central locus of control.

The subsumption architecture is attractive in its elegance, simplicity, and reactive robustness. The main criticism of the approach is its possible practical limitations with respect to scalability. As a system grows in complexity, 'proving' its correctness also becomes increasingly difficult.

2.1.4 Reactive Systems versus Hierarchical Planning

Whilst there is an active perception that reactive and hierarchical planning approaches are diametrically opposed to each other and thus mutually exclusive, this need not necessarily be the case.[14] By exploiting the strengths of each method a synthesis of paradigms can yield improvements in robustness, flexibility, and generality.

The strength of a reactive navigation system is in its intimate connection with the perceived world. Much robustness can be gained through this tight coupling of sense and action and minimal reliance on representation. Several reactive approaches have been proposed, such as Brooks' subsumption architecture (Brooks, 1985; Brooks, 1986b), Payton's reflexive behaviours (Payton, 1986), Kadonoff et al.'s arbitration strategies (Kadonoff et al., 1987), Agre and Chapman's Pengi system (Agre and Chapman, 1987), Kaelbling's intelligent reactive system (Kaelbling, 1987), Arkin's motor schemas (Arkin, 1987a; Arkin, 1987b; Arkin, 1989a), and Zapata et al.'s risk zone (Zapata et al., 1994).

The hierarchical knowledge-based approach to navigation, on the other hand, is best suited to the integration of world knowledge and high-level user intent. Being the basis of the traditional planning approach much of the early literature illustrates this methodology (§2.1.1). Classical planning systems[15] essentially consider a plan to be a partial order over a set of actions. Despite the differences between the various individual planning models, the main identifying feature of this hierarchical and functional decomposition is the heavy reliance on world

[14]See (Firby, 1987; Georgeff and Lansky, 1987; Schoppers, 1987; Arkin, 1989b; Arkin, 1990; Mitchell, 1990; Payton, 1990; Payton et al., 1990; Gat, 1992; Hexmoor and Nute, 1992; Collins et al., 1993; Hexmoor et al., 1993a; Hexmoor et al., 1993b; Lammens et al., 1993; Quinlan and Khatib, 1993; Cameron and Probert, 1994; Langer et al., 1994b; Langer et al., 1994a; Robinson and Jenkin, 1995).

[15]See (Fikes and Nilsson, 1971; Sacerdotti, 1977; Brooks, 1982; Brooks, 1983; Wilkins, 1984; Chatila and Laumond, 1985; Crowley, 1985; Tate, 1995; Elfes, 1986; Goto and Stentz, 1987; Harmon, 1987; Lozano-Pérez, 1987; Noriels and Chatila, 1989).

models and explicit representation. Whilst this is useful for reasoning about the world with respect to explicit high-level tasks, it is less appropriate for low-level real-time interactions with the real world.

Thus instead of viewing reactive systems and knowledge-based planning approaches as opposites, they can be thought of as addressing the two ends of the same problem. Putting it very simply, the knowledge-based approach can be viewed as high-level 'plan formulation', prior to execution, whereas the reactive approach can be viewed as low-level 'plan execution'.

The reactive paradigm can thus conceivably be used to overcome the brittleness observed in many knowledge-based navigational planning approaches, while the reverse can improve overall generality and flexibility, and provide for high-level user intent. Truly 'high-level' intelligent autonomy may well ultimately lie in this integration of methodologies.[16] Interestingly, studies in psychology (Shiffrin and Schneider, 1977; Norman and Shallice, 1986) and neuroethology (Ewert, 1980; Guthrie, 1980) appear to encourage this view.

2.2 Biologically Inspired Robotics

Driven by findings in animal neurobiology, physiology, ethology, and sociology, biologically inspired robotics draws many of its methods and inspirations directly from the living world. Having already evolved effective solutions to many of the 'autonomous agency' problems of interacting and surviving in the real-world, living organisms can provide some of the best clues and insights into how a similar, yet artificial, autonomous agent might be constructed.[17]

While nature's way may not be the only way, or even the best way, of achieving a particular objective, it is still of value by virtue of the very fact that invariably it is the most direct and economical way.

[16]Although, several researchers (Brooks, 1991c; Brooks, 1991b) still argue that purely behaviour-based systems are all that are required. It is still debatable whether reactive systems are truly compatible with solutions that rely on a centralised world model view. Hybrid systems offer a compromise by employing a reactive system for low-level control and a planner for higher level decision making. This can simply be viewed as separating the control system into two communicating but basically independent parts.

[17]Interestingly, mobile robots have also been used to help verify hypotheses about specific control mechanisms in certain animals, such as: phonotaxis in crickets (Lund et al., 1996; Webb and Hallam, 1996), chemical orientation strategies in lobsters (Grasso et al., 1996), and the extraction of compass information and dead reckoning in desert ants (Lambrinos et al., 1998).

2.2.1 Insect Inspired Robotics

Invertebrates, such as insects, are particularly attractive in the present context by virtue of their pervasiveness in virtually all earthly environments, their rich and diverse behaviour, their relative simplicity, and ease of study. Despite having relatively simple nervous systems, insects exhibit many of the desired capabilities requested of autonomous robots, such as highly skilled motor behaviour, efficient communication, learning[18] and cooperation. Insects are thus highly analogous and applicable to autonomous robotics.

Modern robotics still humbles in comparison to some of the feats exhibited by insects. For example, the highly efficient locomotion of the cockroach (Camhi, 1980), the visual and auditory communication among honeybees (von Frisch, 1967; Kirchner and Towne, 1994) and the cooperative talents of ants (Hölldobler and Wilson, 1990). By unearthing some of the mechanisms and behaviours underpinning some of the extraordinary feats exhibited by insects, the possibility exists for not only advantageously applying them to autonomous robotics, but also providing insights into how other more complex behaviours may be generated from relatively simple mechanisms.[19] The economy inherent in insect behaviour is directly applicable to the computationally restrictive world of autonomous mobile robots.

2.2.1.1 Physiology and Neurobiology

Insects have shown that they have solved some of the low-level real time vision problems which have been so difficult to solve in AI and robotics. These tasks, such as collision avoidance, are all the more impressive given the insect's relatively simple nervous system and their use of a rather coarse mosaic of passive electro-

[18]It is worth noting here the misconception that many insects may simply have behaviours or reflexes that are 'hard wired' and hence involuntary. This perceived inability to learn new connections between sensory input and action, is false (Heisenberg and Wolf, 1993).

Insects, for example, have powerful mechanisms of flight steering based primarily on the processing of visual cues, such as motion (optic flow). In a novel experiment (Wolf et al., 1992) it has been demonstrated that, as in humans, in the fly *Drosophila*, flow field information is readily available to motor output systems not specifically designed for this type of sensory data, thus suggesting that correctional steering in insects is more sophisticated than some automatic mechanism. This experiment involved a fly being harnessed to a 'flight simulator' where the force of its legs was used to move the visual panorama (in this case a single vertical dark stripe) accordingly. It was consequently shown that the fly was able to learn, in the course of half an hour, to manipulate (stabilise) the visual panorama.

[19]As has previously been shown, complex behaviours need not necessarily emerge from complex sensory-motor interactions (Braitenberg, 1984).

optical sensors. As some of the signal processing and optical principles[20] at work in insect visual systems are unearthed they may be advantageously applied in the design of mobile robots (Franceschini, 1996).

The visuomotor system developed by Pichon et al. (1989) and Franceschini et al. (1992), for example, is based heavily on the visual system of the common house fly, and is used to successfully drive a mobile robot through a cluttered environment without collision. In attempting to solve real-time visuomotor problems they have shown how the sole ability to perceive visual motion (based on passive sensors) might suffice to steer a robot through a complex environment.

At the moment, there is no visually guided mobile robot that comes anywhere near to matching the real-time visuomotor prowess of the humble housefly. The fly navigates swiftly in unconstrained environments and avoids obstacles without making use of sonars, radars or laser range-finders. An essential requisite of visually guided behaviour in flies is the ability to detect visual motion (Götz, 1968). Motion-sensitive neurons have long been described in the fly's visual system (Franceschini et al., 1989; Franceschini, 1992).

A genuine compound eye with as many facets (100) as the fly's eye (in its equatorial plane) was installed on a wheeled mobile platform (Franceschini et al., 1992). An array of electro-optical EMDs (elementary motion detectors), copied from those of the fly, was wired down underneath the photosensor layer. The EMDs operate concurrently and drive a (parallel) obstacle avoidance algorithm, which controls the steering of the robot in real time. The robot, however, is only partially sighted: it perceives only relative motion and depends on egomotion to evaluate the hazards in its environments. Nevertheless, the robot is able to operate swiftly and autonomously in the real world and in real time, and without any world model or central representation. Rather, action is mediated through the real-time local visual feedback of the environment.

More recently, the retinal movements observed to occur in the compound eyes of flies has inspired the proposal of two eye movement strategies to enhance the ability of a terrestrial mobile robot to perceive visual motion (Mura et al., 1996).

Biological inspiration, however, need not be limited to perceptual systems. The physiology and neuroethology of insects also provide examples of elegant solutions to the basic problem of robust and efficient locomotion. Within the insect world, not to mention the terrestrial animal kingdom in general, one of the

[20]See (Laughlin, 1987; Stavenga and Hardie, 1989; Hausen and Egelhaaf, 1989) for reviews.

most pervasive and successful approaches is legged locomotion (Wilson, 1966). The advantages of possessing legs are many and varied, however, in the present context the sheer robustness of legged locomotion, especially over rough terrain, make it very attractive for use in mobile robotics. Many of the principles of walking in animals, especially insects, can be incorporated into the design and control of legged robots.[21] The amazingly skilled talents of insects in legged locomotion, such as the speed and efficiency of the humble cockroach, are thus highly envied. Studies of the neural basis of cockroach walking (Pearson et al., 1973; Pearson, 1976) have, in fact, been used to design a distributed[22] neural controller for a six-legged walking robot (Beer et al., 1989; Beer et al., 1992; Chiel et al., 1992). More recently, explorations (Espenschied et al., 1996) were made in applying the mechanisms that coordinate walking in the stick insect *Carausius morosus* (Cruse, 1990). However, beyond maintaining stable gaits on level ground, insects must utilise a greater variety of strategies and reflexes to cope with rough terrain (Pearson and Franklin, 1984). The incorporation of these strategies have also been investigated for use with the hexapod robot over rough terrain (Espenschied et al., 1993; Espenschied et al., 1994).[23]

Other forms of locomotion have also been explored for mobile robots, such as flying (Shimoyama et al., 1992; Kubo et al., 1993; Kubo et al., 1994) inspired by flight in insects (Chapman, 1982; Azuma et al., 1985; Ennes, 1988a; Ennes, 1988b; Wooton, 1990), or even swimming (Kukuda et al., 1994).

2.2.1.2 Ethology

Insect ethology,[24] the study of insect behaviour, also has something to offer to the world of autonomous robot navigation.[25] Alongside physical characteristics,

[21] See (Donner, 1987; Brooks, 1989b; Brooks, 1989a; Beer et al., 1991; Espenschied et al., 1993; Kleiner, 1994; Zill and Seyfarth, 1996; Beer et al., 1997).

[22] One of the key characteristics of insect locomotion is distributed control. Local leg 'controllers' decentralise control and thereby decrease computational demand and increase overall flexibility and robustness. (This is especially important to ensure a graceful degradation in behaviour, in response to physical damage.) Stable gaits arise from the cooperative interactions between many separate components.

[23] See (Payton and Bihari, 1991; Simmons and Krotkov, 1991) for a more traditional AI approach to 'autonomous' legged navigation, the human-piloted ASV. See (Simmons and Krotkov, 1991) for a more recent, and more autonomous, example.

[24] Insect ethology is explored in much more detail in section 2.5.

[25] Naturally, various aspects of animal behaviour, such as the mechanisms which drive it, can provide much inspiration for the control of autonomous robots (Anderson and Donath, 1990; Beer et al., 1990; Schnepf, 1991; Nehmzow, 1995a). However, trace fossils (the fossilised remains

insects have evolved many efficient neural and behavioural mechanisms or 'short cuts' to successfully interact and deal with the complexities of the world. Their small brains and relatively simple nervous systems have necessitated the development of efficient ways of perceiving and reacting to the world. In so doing they have also solved some of the real-time issues plaguing autonomy in machines. The parsimonious nature of the insect behaviour is thus directly applicable to the computationally restrictive world of autonomous mobile robots.

Insects have, for example, been shown to successfully exploit apparent motion cues, which are in turn used to drive many useful visually-mediated behaviours. In essentially the same way an autonomous robot can use simple cues and behaviours to efficiently, robustly and intelligently guide its interaction with the world.

Recent studies of insect visual behaviour and navigation reveal a number of elegant strategies that can be profitably applied to the design and control of autonomous mobile robots (Srinivasan et al., 1995; Srinivasan and Venkatesh, 1997; Srinivasan et al., 1997; Srinivasan et al., 1998).

2.2.1.3 Sociology

Many species of insect, such as bees and ants,[26] are also very sociable. They communicate and cooperate with others of their own group in order to perform certain tasks, such as foraging for food and nest building (Wilson, 1971; Hermann, 1982; Lewis, 1984; Franks, 1989). The group intelligence (Bonabeau and Theraulaz, 1994) and the *safety in numbers* survival strategy are just other evolved solutions to the problem of achieving the overall survival and prosperity of both the social group and specie.

Efficient communication, cooperation and distributed yet collective behaviour, however, are also traits that would be very useful to incorporate into groups of small autonomous robots. In hazardous domains, the inherent robustness of a cooperating colony of agents can be an enormous advantage. The interchangeability and inherent redundancy of the individual makes for a more

of animal behaviour) can also be used as inspiration for simple robotic behaviour (Prescott and Ibbotson, 1997a). Early fossil records have been shown to provide important insights into the early evolution of animal spatial behaviour (Seilacher, 1967; Raup and Seilacher, 1969; Seilacher, 1986). Further, it is tentatively suggested (Prescott and Ibbotson, 1997a) that the capacity for intelligent behaviour exhibited by current behaviour-based robots (§2.1.3) is in fact quite similar to that of animals of the early Cambrian period, approximately 530 million years ago.

[26] See (Topoff, 1971; Dumpert, 1981; Hermann, 1982; Hölldobler and Wilson, 1990).

stable and robust community. Individual robots become more expendable as their loss to the community does not threaten the behaviour of the whole. The economics of this approach also make it attractive through the mass production of specific 'classes' of robots. In space/planetary exploration, for example, where there is little chance of rescue or maintenance, there are clear advantages to sending a colony of small, inexpensive robots rather than a single, expensive one (Brooks and Flynn, 1989).

Communication is a very useful social skill for embodied robots (Arkin et al., 1993; Dautenhahn, 1995; Billard and Dautenbahn, 1997a). Billard and Dautenhahn (1997a), for example, show the value of a simple imitative behaviour as a means to develop a vocabulary about another agent's perceptions and environmental states. Using a *teacher-learner* or *mother-child* scenario a two robot[27] experiment focuses on a bidirectional following behaviour as a means for the teacher robot to communicate its 'labels' for locations, based on its sensory stimulation, to the learner. When separated from the teacher, the learner robot is then able to use the teacher's continued transmissions to assist in the searching task. In this case, communication is not a built in behaviour but rather develops through the external *teaching by example* social interaction. Communication, however, need not necessarily take place through an explicit and immediate transfer of information. For example, ants use pheromone trail laying as a very useful and effective means of indirect chemical communication. In this case the communication is based on an in-built behaviour.

Robot cooperation has been investigated for performing various tasks, such as collective transportation (Hashimoto et al., 1991; Stilwell and Bay, 1993; Evans et al., 1997), where a swarm of ant-like[28] robots are used to move palletised loads from one location to another. This would be useful in an automated factory or warehouse application, for example.

Questions about how an insect society functions has provoked interest in self-organisation, collective behaviour and emergent behaviour. A major difficulty, however, is deducing collective activity from individual behaviour. Collective behaviour is not simply the sum of its parts (Pasteels and Deneubourg, 1987).

[27]The robots need not necessarily be homogeneous. In this case, they are both physically different and use different sets of sensors.

[28]Army ants frequently form groups to retrieve large items of prey. This super-efficient cooperative behaviour allows a group of ants to carry a disproportionately heavier load (Franks, 1986). The group performance is much more than the sum of the performances of its members.

Individual behaviour may seem inefficient and chaotic, but the sum produces global behaviour capable of seemingly extraordinary feats. Attempts have been made to model and explain the organisations of recruitment and foraging[29] within ant colonies (Pasteels et al., 1987; Deneubourg et al., 1987; Deneubourg et al., 1992; Fletcher et al., 1995). Self-organisation in groups of robots has also been proposed based on wasp colony function (Theraulaz et al., 1991).

Emergent behaviour can be explored through the simple interactions of many individuals, such as observed in simulated fly swarms (Poggio and Poggio, 1994). Interestingly, emergent behaviour has also been explored in terms of *emergent strategies*, where the emergent strategies obtained from the local interactions between simple agents, are used to play games of strategy, such as chess (Drogoul, 1993).[30]

Biologically-inspired collective and cooperative group behaviours have also been experimented with, using real robots (Beckers et al., 1994; Gaussier and Zrehen, 1994; Martinoli and Mondada, 1995; Maris and te Boekhorst, 1996; Martinoli et al., 1997). For example, a group of ten Khepera (Mondada et al., 1993) miniature mobile robots were used in an experiment involving the gathering and clustering of small cylinders (Martinoli and Mondada, 1995; Martinoli et al., 1997). Similar clustering behaviours are observed in ant colonies (Deneubourg et al., 1991). In a second experiment the group of robots were expected to remove long sticks from holes, requiring synchronous collaboration between a pair of robots (Martinoli and Mondada, 1995).

Ant colony function, such as efficient foraging behaviour (Goss et al., 1992; Hölldobler and Wilson, 1990; Goss et al., 1990; Beckers et al., 1992), has also inspired a new computational paradigm. For example, a new search methodology, involving positive feedback, distributed computation, and greedy heuristics, has successfully been applied to classical optimisation problems such as the traveling salesman problem (Dorigo et al., 1996; Dorigo and Gambardella, 1997). Another example of using an ant colony algorithm is in the control of load balancing in communications networks (Schoonderwoerd et al., 1997).

[29]Interestingly, different species of ants (10,000+) utilise slightly differing recruitment and foraging strategies, depending on such factors as the food distribution within the local environment (Bernstein, 1975).

[30]The multi-agent chess approach views each chess piece as an autonomous agent, with its own behaviour and perception area.

2.3 Robot Vision

Among the many sensors available for mobile robots,[31] such as sonar, tactile and laser range finding, visual sensors arguably provide the greatest single source of information about the surrounding environment.[32] Given our deep understanding of the world around us, even a single static visual scene possesses and can provide us with, a wealth of structural and functional information about the sensed environment.

Visual sensors can also be regarded as very economical in terms of information per unit cost. Specific and sophisticated sensing hardware such as laser scanning rangefinders are still quite expensive[33] in comparison to the mass-produced CCD camera.

The value of vision is certainly not lost on nature. Animal survival and 'fitness' is determined to a large extent by the quality and speed of both the perception of, and reaction to, the dynamic world. To this end, it is not surprising then, that most animals have evolved effective vision systems in their relentless quest for the slightest edge over competition and prey.[34]

Despite the obvious sensory power of visual perception, however, vision has traditionally not been the sensor of choice for real-time mobile robot navigation. Unfortunately, information-rich sensory data, such as provided by vision, usually also entail the most computation, making real-time application more difficult. In this case, this computational expense is due, only in part, to the large volumes

[31] See Everett (1995) for a detailed study and review of both the theory and application of the vast majority of sensors available for robots.

[32] This, however, does not imply that other more limited sensors, such as touch sensors, are necessarily superfluous. Although information gathering at a distance, can be argued to be of greater value due to its ability to allow an agent time to form judgements and to anticipate, and hence perform 'corrective' action before potentially dangerous physical contact is made, other forms of sensing still have a valuable and active role to play. Utilising sensory information from various sources clearly has advantages in allowing a more distributed and comprehensive form of perception. Sensor fusion is also an intelligent way of compensating for the inherently imperfect nature of sensors and perception, and thus provide for a much more robust system. Consider the situation if humans did not possess the sense of touch, an entire world of perception would be lost. A sense of touch is crucial in many of the tasks we do, from grasping and manipulation to simply perceiving when and where physical contact is made with the surrounding environment.

[33] Approximately A$10,000–100,000 (Jarvis, 1997).

[34] Despite the seemingly ubiquitous nature of vision in most reasonably sophisticated animals, there are of course the inevitable exceptions. Bats are a popular example, having developed a very successful echo location mechanism to cope with the inadequacies of vision. Naturally, vision can be quite limited in light restricted environments. Some species of burrowing mole have even done away with vision completely.

of data that usually accompany a vision system. The problem is mostly due to the computational cost of traditional forms of image analysis, such as the reconstructive approach to visual perception, used to extract the 'necessary' high-level navigational cues. Attempting to perceive and understand visual stimulus in terms of high-level structure has shown itself to be a difficult, brittle, and computationally expensive undertaking.

However, relatively simple animals such as insects suggest that this need not necessarily be the case. Given their restricted processing capacity they cannot perceive the world in the same manner as humans do (or are thought to do). Instead, they utilise parsimonious solutions or 'short cuts', based to a large extent on relative motion cues, to effectively and quickly perceive and react to the world. This kind of almost reflex-like visual behaviour has been shown to be used by many insects, especially flying insects, to achieve much of their remarkable navigational success (§2.5).

Thus by concentrating on only the 'important' elements of visual stimuli, such as motion, that are actually needed for specific behaviours, it is possible to achieve effective control at much reduced computational expense. In this way, effective real-time visual guidance for a mobile robot becomes very much achievable, despite limited processing power.[35] There is no inherent need for an 'understanding' of the perceptual world, to anywhere near the degree that we possess, simply to achieve intelligent and real-time autonomy.

A similar train of thought has more recently emerged from research concerned with various aspects of perception and action. Specifically, the value that can be gained through an intimate relationship between the two.[36] One of the emerging ideas, closely allied with the behaviour-based and biologically inspired philosophies previously outlined, is the concept of *visual servoing* where action elicits and simplifies perception, which in turn directly drives behaviour.[37] The real-time advantages of having visual stimuli directly drive useful behaviour are obvious. It is no coincidence that insects, especially flying insects, rely so heavily

[35]The increasingly economical nature of the perceptual process also allows (more) resources to be redirected toward higher-level tasks.

[36]The value of an intimate connection between action and perception has been discussed previously in various guises, such as active perception (Bajcsy, 1988; Aloimonos, 1993), active vision (Aloimonos et al., 1988; Blake and Yuille, 1992), animate vision (Ballard, 1991), and purposive vision (Aloimonos, 1990).

[37]See (Espiau et al., 1992; Clark and Ferrier, 1992; Fermüller, 1993; Raviv and Herman, 1993; Sandini et al., 1993).

on such visual behaviours.

An example application of this type of visual behaviour for mobile robot navigation is provided by Santos-Victor and Sandini (1997c). Driven purely by visual stimuli, in this case from first-order optical flow, a real-time robot docking behaviour is elicited without any need for a 3D reconstruction of the environment or any initial calibration procedures. Visual behaviours can thus be designed for a variety of mobile robot related tasks (Santos-Victor et al., 1997a; Santos-Victor and Sandini, 1997a), such as obstacle detection (Santos-Victor and Sandini, 1995a), for example.

2.4 Image Motion

Two-dimensional image motion is the projection of the three-dimensional motion of the real-world, relative to the visual sensor, onto the imaging plane.

The importance of visual motion in understanding and interacting with the three-dimensional world cannot be overstated (see §2.5). Visual motion can supply useful information about both the three-dimensional motion of the observer as well as the three-dimensional structure of the scene.[38]

2.4.1 Optical Flow

The *optical flow field*, or alternatively the *image velocity field*, refers to the estimated two-dimensional image motion observed between multiple images or frames. The flow field can represent either instantaneous image velocities or, alternatively, discrete image displacements.

The optical flow field can be used to perform many useful functions, such as 3D sensor motion estimation, depth perception and general scene reconstruction,[39] motion and object segmentation,[40] object shape and orientation estimation,[41] time-to-collision calculation,[42] focus of expansion (FOE) estimation,[43] and stereo disparity measurement.[44]

A large amount of research has been conducted, over the last two decades, into reliable and accurate ways of estimating optical flow. Many different methods have been put forward in this endeavour. The vast majority of these, however, can

[38]See (Gibson, 1950; Gibson and Gibson, 1957; Gibson, 1966; Hay, 1966; Braunstein, 1976; Gibson, 1979; Ullman, 1979; Koenderink, 1986)

[39]See (Hay, 1966; Nakayama and Loomis, 1974; Braunstein, 1976; Prazdny, 1979; Ullman, 1979; Longuet-Higgins and Prazdny, 1980; Prazdny, 1980; Longuet-Higgins, 1981; Tsai et al., 1982; Tsai and Huang, 1984; Adiv, 1985; Adiv, 1989; Barron et al., 1990; Negahdaripour and Lee, 1992; Giachetti et al., 1994)

[40]See (Nakayama and Loomis, 1974; Jain, 1984; Adiv, 1985; Murray and Buxton, 1987; Duncan and Chou, 1992; Rognone et al., 1992; Zhang and Faugeras, 1992; Bouthemy and François, 1993)

[41]See (Longuet-Higgins and Prazdny, 1980; Tsai et al., 1982; Tsai and Huang, 1984; Longuet-Higgins, 1984; Subbarao, 1988; Subbarao, 1989)

[42]See (Subbarao, 1990; Cipolla and Blake, 1992; Meyer and Bouthemy, 1992; Micheli et al., 1993)

[43]See (Longuet-Higgins and Prazdny, 1980; Regan and Beverley, 1982; Jain, 1983; Overington, 1987; Sundareswaran, 1992)

[44]See (Yakimovsky and Cunningham, 1978; Barnard and Thompson, 1980; Cornilleau-Pérès and Droulez, 1990; Jenkin, 1990; Langley et al., 1990; Langley et al., 1991; Jenkin et al., 1991)

be classified into a handful of basic types:[45] intensity-based differential methods, frequency-based methods, correlation-based methods, multiple motion methods, and temporal refinement methods. Each class of method operates under its own subset of assumptions and suffers slightly differing problems and limitations. Some common assumptions, for example, include uniform scene illumination, Lambertian surface reflectance, and a single-surface hypothesis. As idealistic assumptions about the environment and visual perception are violated, to varying degrees, the reliability and accuracy of the various methods, in determining optical flow, are thus also affected. Each method has its own set of advantages and disadvantages which determine its suitability to particular scenarios.[46] However, no single method yet exists which can boast ubiquitous reliability, accuracy and density.

2.4.1.1 Initial Constraints

The initial assumption in measuring image motion is that the intensity profiles, of local image regions, are approximately constant for at least a short amount of time (Horn and Schunck, 1981). Formally, if $I(x, y, t)$ is the image intensity function, then this local constraint can be expressed as

$$I(x, y, t) \approx I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{2.1}$$

where Δx and Δy define the displacement, of the local image region, after time Δt. Using a Taylor series expansion, ignoring second-order partial derivatives and higher order terms, this yields

$$I_x u + I_y v + I_t = 0 \tag{2.2}$$

where I_x, I_y and I_t are the first-order partial derivatives of $I(x, y, t)$ with respect to x, y, and t, respectively — the spatial and temporal intensity gradients — and $\mathrm{v} = (u, v)$ is the 2D image velocity. Equation 2.2 is commonly referred to

[45] See (Aggarwal and Nandhakumar, 1988; Beauchemin and Barron, 1995) for a review of both feature-based approaches and optical flow based approaches.

[46] See (Aggarwal and Nandhakumar, 1988; Barron et al., 1994a; Barron et al., 1994b). Hybrid models combining more than one method, such as in Ogata and Sato (1992), appear promising in this regard.

as the *optical flow constraint equation*, or alternatively the *image flow constraint equation* (Schunck, 1986).

Thus the component of the movement in the direction of the intensity gradient (I_x, I_y) is given by

$$\|\mathrm{v}_\perp\| = \frac{I_t}{\sqrt{I_x^2 + I_y^2}} \tag{2.3}$$

In fact, the normal velocity component $\mathrm{v}_\perp$ is the only one that can be estimated. This constraint equation alone is not sufficient to compute both components of v, the two unknowns u and v. This is commonly known as the *aperture problem* (Ullman, 1979).

The two-dimensional image motion (u, v) cannot be computed locally without introducing additional constraints,[47] such as, for example, an optical flow smoothness constraint, where the velocity field varies smoothly over the image region. These constraints can, however, introduce new difficulties, such as in this case when flow discontinuities arise due to visual occlusion.

Several forms of visual phenomena, such as occlusion, transparent motions,[48] and dynamic objects, can cause problems in the determination of image motion (see §2.4.1.5).

2.4.1.2 Intensity-Based Differential methods

Intensity-based differential methods[49] compute image velocity from spatiotemporal derivatives of image intensity.[50] The image domain is thus assumed to be differentiable in space and time. Methods based on the optical flow constraint equation 2.2, such as that expressed by equation 2.3, can be used to compute the optical flow.

[47]While this is true for this type of motion representation, the aperture problem cannot be generalised to all motion detection schemes. The correct two-dimensional velocity can, in principle, be calculated from purely local mechanisms without reference to additional global constraints (Reichardt et al., 1988).

[48]Transparent motions are caused by the occlusion of a surface by a translucent object. See (Irani et al., 1994).

[49]See (Longuet-Higgins and Prazdny, 1980; Horn and Schunck, 1981; Glazer, 1987a; Enkelmann, 1988; Lee et al., 1988; Aisbett, 1989; Srinivasan, 1990; Sobey and Srinivasan, 1991; Burgen et al., 1992; Schnörr, 1992; Fleet and Langley, 1995)

[50]Egomotion and scene structure may also be recovered directly from the image intensity derivatives (Zinner, 1986; Horn and Weldon, 1987; Negahdaripour and Horn, 1987; Heel, 1990).

Different individual techniques may also use different sized neighbourhoods from which to draw a velocity estimate. Local methods, for example, may use the normal-velocity information in small, local neighbourhoods, to perform a least mean square minimisation to find the best fit for (u, v). Surface or contour models may also be used to integrate normal velocities into full velocities (Bergholm, 1988).

Hierarchical approaches, applying differential techniques in a coarse-to-fine framework, may be used to help alleviate some of the problems caused by large 2D image motion and aliasing effects. Glazer (1987a), for example, shows how first order gradient-based methods may be extended to cope with large disparities by formulating a hierarchical generalisation of the single level method. A coarse-to-fine control strategy is used to progressively refine individual motion estimates. This strategy can also be applied to correlation-based image motion detection algorithms (Glazer, 1987b).

2.4.1.3 Frequency-Based methods

Frequency-Based methods[51] compute image velocity using orientation sensitive filters in the Fourier domain of time-varying images.

The Fourier transform of equation 2.1 is

$$\widehat{I}(k,\omega) = \widehat{I}_0(k)\delta(v^T k + \omega) \tag{2.4}$$

where $\widehat{I}_0(k)$ is the Fourier transform of $I(x, y, 0)$, δ is a Dirac delta function, and (k, ω) denotes spatiotemporal frequency. This yields the optical flow constraint equation, in frequency space, as

$$v^T k + \omega = 0 \tag{2.5}$$

This shows that the velocity of a translating 2D intensity signal can be expressed as a function of its spatiotemporal frequency and forms a plane through the origin of the Fourier space.[52]

[51] See (Adelson and Bergen, 1985; Watson and Ahumada, 1985; Heeger, 1988; Fleet and Jepson, 1990; Grzywacz and Yuille, 1990; Heeger and Jepson, 1992; Beauchemin and Barron, 1995)

[52] Interestingly, it has been shown that certain frequency-based methods are equivalent to

2.4.1.4 Correlation-Based methods

Image matching approaches attempt to match image features, corresponding to three dimensional object features in the scene, such as edges and corners, between images.[53] Such matching methods naturally rely heavily on the presence and extraction of relatively sparse but highly discriminatory features. Image displacements and consequently image velocities are determined by correspondence.[54] However, even when matchable image features are available, establishing and maintaining correct correspondence can be problematic and unreliable.

Correlation-based matching[55] attempts to solve this problem by defining image velocities, in terms of image displacements, by finding the best fit between contiguous time-varying image regions. This is accomplished by maximising a similarity measure, or alternatively, minimising a dissimilarity measure. A correlation coefficient between two functions $f(x)$ and $g(x)$ can be defined by the integral

$$\int (f(x + \Delta x) - g(x))^2 \, dx \tag{2.6}$$

Assuming $f(x + \Delta x)$ does indeed equal $g(x, y)$, finding (Δx) which minimises this integral is equivalent to finding the displacement between f and g.

Similarly, the image motion that has occurred between instantiations of the time-varying intensity function $I(x, y, t)$ of an image, can be determined by minimising the double summation

$$\sum\sum (I(x + \Delta x, y + \Delta y, t + \Delta t) - I(x, y, t))^2 \, dx \, dy \tag{2.7}$$

Finding the value of $(\Delta x, \Delta y)$ which minimises this summation is equivalent to finding the image displacement, and hence image velocity, between $I(x, y, t)$ and $I(x, y, \Delta t)$.

---

correlation-based methods and to certain gradient-based methods (Adelson and Bergen, 1985; Barron et al., 1994b).

[53] See (Aggarwal and Nandhakumar, 1988) for a review.

[54] See (Ullman, 1979; Anstis, 1980)

[55] See (Barnard and Thompson, 1980; Sutton et al., 1983; Scott, 1987; Anandan, 1989; Kalivas and Sawchuk, 1991; Zheng and Chellappa, 1993)

Due to the fact that correlation matching operates under a constant velocity model, the size of correlation windows is an important parameter for correlation-based region matching techniques. The optimal correlation window size depends heavily on the structure of the underlying signal. The window must be large enough to encompass enough signal variation for reliable correlation, whilst not too large to significantly violate the constant velocity assumption. Hierarchical approaches[56] can once again assist in this regard.

2.4.1.5 Multiple motion methods

One of the main problems that still affects most optical flow techniques is that of discontinuous motion or multiple motions. In real-world images, motion discontinuities are usually caused by visual occlusion. Occlusion boundaries are problematic for gradient-based approaches because of the requirement for intensity derivatives, and are problematic for correlation-based approaches because of possible appearance or disappearance of image structure. Several strategies[57] have been proposed to deal with the problem of multiple image motion, such as explicitly modelling motion discontinuities or by relaxing smoothness constraints.

2.4.1.6 Temporal refinement methods

Temporal refinement methods propose an incremental or iterative approach to determining optical flow. The advantages of incorporating motion estimates from earlier calculations include improved accuracy and computational efficiency (Beauchemin and Barron, 1995; Fleet and Langley, 1995).

2.4.2 Image Interpolation

Image motion can also be estimated with respect to synthetic models of expected image motion. Image interpolation makes use of the relative motion differences between a set of known reference images and the actual motion images.

[56] See §2.4.1.2 and (Glazer, 1987b; Anandan, 1989; Okutomi and Kanade, 1990; Bergen et al., 1992).

[57] See (Adiv, 1985; Mitiche et al., 1988; Aisbett, 1989; Schunck, 1989; Black and Anandan, 1990; Little and Gillett, 1990; Nagel, 1990; Schnörr, 1992; Burgen et al., 1992; Negahdaripour and Lee, 1992; Jepson and Black, 1993; Irani et al., 1994; Beauchemin and Barron, 1995)

This method computes the motion of a rigid, textured planar surface by a single-stage, non-iterative process which interpolates the average image motion with respect to a set of reference images (Srinivasan, 1994; Nagle and Srinivasan, 1996).

By assuming that the image motion between two images f_0 and f is continuous and linear, one can interpolate f from f_0 with respect to reference images synthetically generated from f_0. This assumption holds provided that both the image motion and the referential motion is small compared to the period of the highest frequency image intensity structures.

The main advantages of this technique are:

(i) no identification or tracking of features is required

(ii) no measurement of high-order spatial or temporal derivatives is required

(iii) no iterative calculation is required

(iv) it is robust to noise

2.4.2.1 Calculating Image Translation

Consider two successive frames, $f_0(x, y)$ and $f(x, y)$, captured by a moving camera. To calculate the horizontal image motion that has occurred between f_0 and f, two reference images, f_1 and f_2, must first be generated. The two reference images, f_1 and f_2, are synthetically generated by shifting f_0 by $+x_{ref}$ and $-x_{ref}$ along the x axis:

$$f_1(x, y) = f_0(x - \Delta x, y) \tag{2.8}$$

$$f_2(x, y) = f_0(x + \Delta x, y) \tag{2.9}$$

Assuming that the captured images have been appropriately smoothed (to remove aliasing effects, etc.), an estimate of f can be expressed as a linear interpolation between the reference images f_1 and f_2. Assuming the image deforms continuously and linearly from f_0 to f, the interpolated estimate of f can be expressed as

$$\widehat{f} \simeq f_0 + 0.5 \left(\frac{\widehat{\Delta x}}{\Delta x_{ref}} \right) (f_2 - f_1) \tag{2.10}$$

The image translation between f_0 and f can now be calculated by determining the value of $\widehat{\Delta x}$ that minimises the difference between f and the interpolated estimate $\widehat{f}$. This can be done by minimising the mean-square error ε between f and $\widehat{f}$ over some window of the image, defined by the function Ψ:

$$\varepsilon = \int\int \Psi(f - \widehat{f})^2 \, dx \, dy \tag{2.11}$$

or equivalently

$$\varepsilon = \int\int \Psi \left\{ f - \left[f_0 + 0.5 \left(\frac{\widehat{\Delta x}}{\Delta x_{ref}} \right) (f_2 - f_1) \right] \right\}^2 dx \, dy \tag{2.12}$$

By differentiating ε with respect to $\widehat{\Delta x}$, and setting this to zero, horizontal image translation is given by

$$\widehat{\Delta x} = \frac{2\Delta x_{ref} \int\int \Psi(f - f_0)(f_2 - f_1) \, dx \, dy}{\int\int \Psi(f_2 - f_1)^2 \, dx \, dy} \tag{2.13}$$

where Ψ represents the weighting function, such as a 2D gaussian, applied to the window.

To calculate image motion along both x and y axes, the image interpolation technique is simply extended through the use of additional reference images. In this case, additional reference images are generated by shifting f_0 by $+y_{ref}$ and $-y_{ref}$, along the y axis. This generates a set of simultaneous equations, one for each unknown, which can be solved by matrix inversion.

2.4.2.2 Calculating Surface Slope

Surface slope is calculated in a similar way to translation. Consider the situation where a moving camera views a vertical planar surface that is sloped at an angle with respect to the horizontal axis.

To compute surface slope, both image translation and image compression must first be computed. To calculate image compression two additional reference frames, f_3 and f_4, are generated. These reference frames are the compressed and expanded versions of the original image f_0:

$$f_3(x,y) = f_0(x(1-\Delta c_{ref}),y) \tag{2.14}$$
$$f_4(x,y) = f_0(x(1+\Delta c_{ref}),y) \tag{2.15}$$

Now the interpolated estimate of $\widehat{f}$, assuming horizontal image motion and compression, can be expressed as

$$\widehat{f} \simeq f_0 + 0.5\left(\frac{\widehat{\Delta x}}{\Delta x_{ref}}\right)(f_2 - f_1) + 0.5\left(\frac{\widehat{\Delta c}}{\Delta c_{ref}}\right)(f_4 - f_3) \tag{2.16}$$

giving the minimisation equation

$$\varepsilon = \int\int \Psi\left\{f - \left[f_0 + 0.5\left(\frac{\widehat{\Delta x}}{\Delta x_{ref}}\right)(f_2 - f_1) + 0.5\left(\frac{\widehat{\Delta c}}{\Delta c_{ref}}\right)(f_4 - f_3)\right]\right\}^2 dx\,dy \tag{2.17}$$

By again differentiating ε with respect to the unknowns, Δx and Δc, and setting the resulting expressions to zero, image translation and compression is given by the two simultaneous equations

$$\begin{aligned}\left(\frac{\widehat{\Delta x}}{\Delta x_{ref}}\right)\int\int \Psi\ (f_2 - f_1)^2\,dx\,dy + \\ \left(\frac{\widehat{\Delta y}}{\Delta y_{ref}}\right)\int\int \Psi\ (f_4 - f_3)(f_2 - f_1)\,dx\,dy \\ = 2\int\int \Psi\ (f - f_0)(f_2 - f_1)\,dx\,dy\end{aligned} \tag{2.18}$$

$$\begin{aligned}\left(\frac{\widehat{\Delta x}}{\Delta x_{ref}}\right)\int\int \Psi\ (f_2 - f_1)(f_4 - f_3)\,dx\,dy + \\ \left(\frac{\widehat{\Delta y}}{\Delta y_{ref}}\right)\int\int \Psi\ (f_4 - f_3)^2\,dx\,dy \\ = 2\int\int \Psi\ (f - f_0)(f_4 - f_3)\,dx\,dy\end{aligned} \tag{2.19}$$

The two unknowns, translation and compression, can be solved by matrix inversion. Surface slope can now be calculated using the computed image translation and compression:

$$slope = \arctan \frac{\omega \sin\theta}{\omega \cos\theta + compression \sin\theta} - \theta \tag{2.20}$$

where ω represents image translation (i.e. angular velocity) and θ the angle between the direction of motion and the camera viewing direction.

2.5 Insect Behaviour

Insects show, by their behaviour, that they perceive the world in three dimensions. However, unlike vertebrates, insects possess immobile eyes with small interocular separations and fixed-focus optics, implying that the range of an object cannot be inferred by convergence cues, stereopsis[58] or the refractive (focusing) power required to bring the image into focus on the retina (Wehner, 1981; Collett and Harkness, 1982; Horridge, 1987; Srinivasan, 1993). However, insects are very adept at exploiting cues based on image motion to infer depth (see Srinivasan (1992a; 1992b; 1993) for a review).

When the eye of an insect moves in a straight line, the images of stationary objects move on the retina. The speed of this apparent motion depends on distance: closer objects appear to move more rapidly. Thus, image motion can be used to infer object range. The most striking application of this strategy is the peering behaviour of locusts. Before jumping, a locust will sway its head and body from side to side. They use this lateral peering motion of their heads to estimate the range of a nearby target in terms of the motion of its image on the retina. In this way the locust is able to adjust the power of its jump accordingly (§2.5.3).

Considerable evidence now suggests that even flying insects are able to infer the ranges of objects from the apparent motion of their images across the eye. The closer a stationary object, the higher the apparent velocity of its image on the retina. Thus, if an insect knows its speed of motion, it can estimate the range of an object from the apparent angular velocity of its image.

2.5.1 Utilising Apparent Motion

Apparent-motion cues are utilised extensively by many insects for a great variety of visually mediated behaviours.

2.5.1.1 Honeybees

Honeybees, for example, have shown that, whilst in flight, they are able to locate a textured figure when presented raised above a textured background.

[58] One notable exception to this is the praying mantis which appears to use stereopsis as well as motion-parallax cues to gauge depth (Rossel, 1983; Rossel, 1986). The praying mantis seems to rely, at least to some extent, on binocular triangulation when estimating the distance of prey.

This figure-ground discrimination ability has been investigated and shown to be accomplished by using cues based on the apparent motion of the figure relative to the background.[59] This was shown, through behavioural experimentation, by training bees to associate a reward with a raised figure (e.g. a disc) and then observing and analysing the spatial distribution of the bees' landings when the reward is removed.

These experiments (Srinivasan et al., 1990) resulted in several interesting findings:

(i) The bees are unable to detect a textured disc that is placed directly on a similarly textured background.

(ii) The bees do not use differences in apparent texture density between disc and background as a cue in locating a raised, textured disc against a similarly textured background.

(iii) The majority of landings occur at the boundary of the figure, irrespective of its shape.

(iv) The bees tend to land facing the inside (centre) of the figure.

(v) Bees trained to detect a raised disc perform equally well at detecting a raised square or triangle when this is offered instead of the disc. In fact, they are unable to distinguish the disc and a similarly textured figure of another shape (when offered simultaneously).

These findings, in turn, allow some deductive reasoning concerning the underlying behavioural processes. For example, the observation that the bees' performance is not influenced by differences in texture density, between figure and ground, appears to rule out the possibility that this is used as a cue to detect the raised figure. The bees appear to be detecting the raised figure by using a cue derived from the motion of the image on the retina. The experiments using a range of different patterns on the raised figures indicate that the important parameter seems to be the relation between the motions perceived on either side of the boundary (i.e. 'boundary parallax') between figure and ground. In

[59]See (Lehrer et al., 1988; Srinivasan et al., 1990; Lehrer and Srinivasan, 1993). Relative motion cues also explain the figure-ground discrimination ability of the fly (Reichhardt and Poggio, 1979).

honeybee vision, the boundaries of objects appear to play an important role in visual processing and behaviour (Lehrer et al., 1990). Their visual systems are particularly sensitive to edges that usually occur at the boundary between object and background.

2.5.1.2 Ladybirds

Another example of a visually mediated behaviour, based on apparent-motion cues, is that exhibited by ladybirds in their approach behaviour to nearby stalks. Essentially, a visuo-motor reflex helps walking ladybirds turn towards and approach nearby, plant-like objects (Collett, 1988).

Experiments examining the turning responses of ladybirds to moving visual stimulus, indicate that the attraction for close objects is due to the fact that during walking such objects will generate a higher retinal image velocity than more distant ones. The visual 'filters' mediating the turn are tuned to higher image velocities.

The way the ladybird appears to use optic flow is to make the size of its turns depend on image velocity and hence distance. Simulations of this strategy resulted in a spiral path towards a target which, to some degree, resembles the real approaches by ladybirds.

2.5.1.3 Male Hoverflies

Males of many species of hoverfly spend much of the day hovering in one spot ready to dart instantly after any passing female. This pursuit behaviour is so primed that the hoverfly will chase many inappropriate targets, ranging from pebbles to distant birds (Collett and Land, 1975a).

An investigation (Collett and Land, 1978) of this pursuit behaviour of single male hoverflies, by shooting various projectiles at them and observing their behaviour, indicated that

(i) The fly's first movement is not directly towards the target but rather in approximately the same direction as that of the target. This suggests an interception rather than a 'tracking' course.[60]

[60]This is in contrast to the chasing behaviour observed in male houseflies (Land and Collett, 1974).

(ii) Once the target is detected the fly moves off at a rapid, uniform acceleration which shows no correlation with the speed of the target or its image velocity on the retina.

(iii) The initial flight course is calculated and maintained without recourse to sensory feedback; the fly does not respond immediately to a dramatic reversal of direction of the projectile (at the beginning of a chase). However, at a later stage the retinal position of the target image does have a much greater control over the fly's orientation.

Knowing only the angular position and angular velocity of the target at the moment of sighting is not enough to specify uniquely the course the male should take to intercept. But it can be assumed the male hoverfly 'knows' three other things: (1) the expected velocity of the female, (2) the distance at which the female is likely to become visible, and (3) the male's own acceleration. If all these quantities are known (i.e. the fly assumes it's intercepting a female) then a planned interception course is possible. Indeed, the interception course calculated for a non-female projectile is, as expected, incorrect.

2.5.1.4 Waterstriders

Waterstriders show how apparent-motion cues can be utilised for self-motion stabilisation (Junger and Varjú, 1990; Junger, 1991; Junger and Dahmen, 1991). Visual and self stabilisation underpin much of insect visual behaviour and guidance (Collett et al., 1993). For flying and swimming insects, without a firm hold on the ground it is important to monitor self-motion. Waterstriders, for example, are one of the fastest insects that live on the water's surface. They compensate for displacements on the surface, caused by wind or water flow, by making occasional, high speed jumps to maintain an average stationary position. In fact, when waterstriders are placed on water with non-uniform flow velocity, which induce both rotation and translation, they are still able to compensate for displacement as well as rotation. They do this by jumping (to compensate for drift) and moving their legs (to compensate for rotation). This implies that in the case of simultaneous translation and rotation the insects react to each component with a separate behavioural sequence. That is, they can discriminate visually between both rotation and translation. Other similar experiments show that the waterstriders are able to distinguish rotation and translation monocularly, and

that the discrimination of rotation is only possible if an extended area of the rigid and contrasted world is visible.

2.5.1.5 Dragonflies

The detection of moving objects is very important to animals in general. This task is easy when the animal is stationary, but when it is itself in motion the task becomes much more difficult. The animal experiences not only the motion of actual moving objects, but also the movement of the entire environment (due to its own motion).

The dragonfly, like many insects, must distinguish between small moving objects such as prey and predators, and the rest of the visual world, in spite of the added visual motion caused by egomotion. To shed some light on this process, Olberg (1981) examined the responses of individual object-movement detectors and self-movement detectors, to moving visual stimulus, to see how two groups of interneurons (in the dragonfly ventral nerve cord) are able to discriminate between 'small object' and 'world'. The key finding was that object-movement detectors discriminate primarily based upon the extent of the object perpendicular to the direction of its motion, whereas, self-movement detectors discriminate primarily based upon the extent of the pattern in the direction of motion.[61]

2.5.2 Range Perception

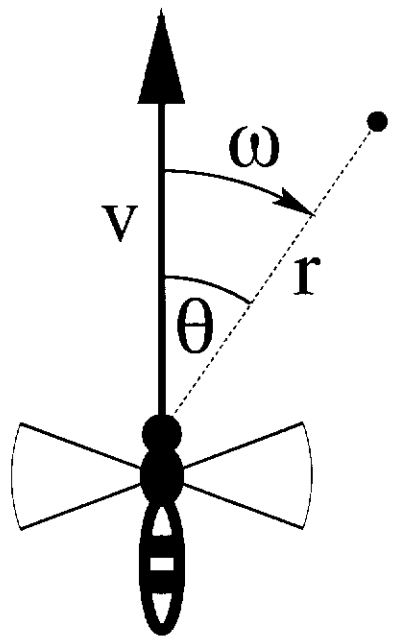
Insects show by their behaviour that they perceive the world in three dimensions. This is accomplished through adept exploitation of relative motion cues. Moving insects are able to infer the ranges of objects from the apparent motion of their images across the eye.[62]

The range (r) of an object can be inferred from its apparent angular velocity (ω), its bearing (θ), and the linear velocity (v) of the eye (Whiteside and Samuel, 1970; Nakayama and Loomis, 1974) (see fig. 2.2):

[61]The way in which freely flying honeybees discriminate between stationary and moving objects has also been studied (Lehrer and Srinivasan, 1992). By exploiting the honeybees' spontaneous preference for moving objects a series of experiments were designed to examine the cues by which they detect a moving object. This preference for moving objects, is in fact, irrespective of reward.

[62]See (Wallace, 1959; Collett, 1978; Eriksson, 1980; Goulet et al., 1981; Collett and Harkness, 1982; Horridge, 1986; Collett, 1988; Lehrer et al., 1988; Kirchner and Srinivasan, 1989; Srinivasan et al., 1989; Sobel, 1990; Srinivasan et al., 1991).

$$r = \frac{v}{\omega}\sin(\theta) \tag{2.21}$$



$$r = \frac{v}{\omega}\sin(\theta)$$

Figure 2.2: Range from apparent velocity. The principle of motion parallax, or more accurately velocity parallax, can be used to gauge the distance to a stationary object. The observer, the insect in this case, can assess the distance r to a static object if it is able to estimate its own translatory speed v and the angular speed ω of the object crossing its field of view.

2.5.3 Peering

Motion parallax is the phenomenon in which an observer's motion (through a stationary environment) results in the perception of apparent motion of objects. As previously described, the apparent motion depends on distance: closer objects appear to move faster. If the observer's motion is known, the distance to an object can be determined (§2.5.2).

The most readily observed example of a scanning movement used to obtain motion parallax information is the peering behaviour of locusts (Wallace, 1959; Collett, 1978; Sobel, 1990). Before jumping onto a nearby object, a locust will sway its head and body from side to side,[63] in order to gauge the distance and

[63]Peering is typically 5–10 mm in amplitude.

hence the power of its jump.

Sobel (1990) describes experiments used to analyze this peering behaviour used by the locust to estimate distances. The experiments involve a locust jumping from a platform onto a moveable target. The target is moveable to enable the alteration of parallax during peering.

The following results were obtained from the study:

(i) Jump velocity increases monotonically with target distance, implying a reliable measure of the locust's intended jumping distance (hence its estimate of target distance).

(ii) The movement of the target during peering indicated locusts do use motion parallax to judge distance. (The movement of the target can simulate the parallax of an object at any specified distance.)

(iii) In computing target distance, locusts seem to ignore the relationship between the directions of head and image movement.

(iv) Locusts seem to perceive less motion parallax with one eye occluded, and ordinarily the motion parallax observed in each eye seems to be combined (e.g. summed or averaged).

(v) Locusts appear to use the extent of lateral head motion in their computation of distance.

2.5.4 Centring

Flying insects, such as honeybees, use information gleaned from image motion to safely navigate around obstacles or fly a straight equidistant path between obstacles. It has been shown (Kirchner and Srinivasan, 1989; Srinivasan et al., 1991) that honeybees centre their flight paths between obstacles simply by balancing the speeds of image motion on their two eyes. For instance, when flying down a textured tunnel, bees strive to maintain equidistance from the two flanking walls by effectively balancing the apparent angular speeds of their images on the two eyes.

2.5.5 Looming

One of the most important flight manoeuvres flying insects perform is that of landing. They have to detect the approaching landing site, and perform the necessary motor skills (e.g. extending their legs) for a safe landing on various surfaces (Goodman, 1960; Wagner, 1982; Borst, 1986; Borst and Bahde, 1988).

It has been shown (Braitenberg and Taddei-Ferretti, 1966; Wehrhahn et al., 1981) that landing behaviour can be induced by an apparent visual pattern expansion (i.e. looming), in front of the fly, which mimics the fly's approach towards a landing site.

By studying the landing responses of tethered flying flies under these synthetic visual conditions it is possible to investigate the sensory cues which lead a fly to initiate landing. Experimental results (Borst and Bahde, 1988) suggest that landing behaviour is induced by a temporal accumulation of the outputs of movement detectors. Further, the response to a movement stimulus is proportional to the number of stimulated movement detectors (if their position-dependent weight is taken into account). Accordingly, the output signals of different movement detectors (from either eye) simply add to form the signal, which after being integrated in time, triggers landing.

Wagner (1982) argued that the critical cue which initiates the onset of deceleration (which is regarded as the beginning of the landing phase) is the relative retinal expansion velocity (RREV). Wagner's investigation involved the observation of freely flying female houseflies (*Musca domestica* L.) as they approached and landed on black stationary spheres. Analyses were made of the landing trajectories (i.e. examining variables such as the fly's velocity, distance and velocity relative to the target, angle subtended by the target on the retina, and the target's retinal expansion velocity) from which conclusions about sensory cues were drawn. Essentially, the RREV is the inverse of *time-to-collision*.[64] The RREV gives information about the range of a target in relative units, dependent on both the velocity and direction of flight with respect to the target; distance is tagged with time. It is therefore implied that the important cues for landing are times, not distances.

Once again it is an apparent-motion cue which drives behaviour.

[64] Also known as *time-to-contact*, this is the time needed to reach the target directly, at constant velocity. See (Lee, 1976). See also equation 4.1 and §2.4.1.

2.5.6 Regulating Forward Speed

Honeybees have also been shown (Srinivasan et al., 1996) to regulate flight speed by monitoring the speed of apparent motion. Essentially, they strive to hold apparent image speed constant. For example, when forced to fly down a tapered tunnel, bees slow down as they approach the narrowest section and speed up again as the tunnel widens once more. Similar behaviours have also been observed in the visual control of flight speed in the fruitfly (David, 1982).

It is clearly an advantage to slow down in tight situations. However, this visual behaviour also has an advantage when it comes to landing. When applied to a landing surface this visual response has the effect of automatically ensuring forward flight speed decreases with altitude, and is close to zero at touchdown. This very same behaviour is also observed in the landing trajectories of bees. As a bee prepares to land on a horizontal surface it again holds constant the apparent speed of the approaching surface and consequently also maintains a safe forward speed approximately proportional to altitude.

Once again, a simple visual behaviour allows intelligent flight control without the need for explicit knowledge concerning the three-dimensional structure of the environment, such as the concept of altitude.

2.5.7 Measuring distance travelled

Recent investigations (Esch and Burns, 1995; Esch and Burns, 1996; Srinivasan et al., 1996) have indicated that foraging honeybees also use image motion to gauge the distances travelled to food sources. The findings revealed that honeybees measure distances to goals by integrating, over time, the apparent motion observed en route. This provides the bee with a visually driven 'odometer' that is robust to variations in travel speed and energy expenditure.[65] Although perhaps not the only information used to gauge long distances,[66] this cue does appear to be significant in measuring short distances. However, due to the fact that apparent motion is dependent on the distances of objects in the visual panorama, this visual odometer is also necessarily route dependent. As such, it can only provide an accurate distance measure for previously travelled routes.

[65]Foraging desert ants have also been shown to use self-induced visual motion to gauge travel distance (Ronacher and Wehner, 1995).

[66]See (Goller and Esch, 1990; Esch et al., 1994; Kirchner and Braun, 1994).

2.5.8 Homing by Path-Integration and Search

An interesting problem faced by many foraging insects is that of successfully returning home after searching for food.[67] Two separate navigational systems have been found to be of prime importance in achieving this task. Firstly, foraging bees and ants, use a vector navigation strategy (von Frisch, 1967; Wehner, 1972; Wehner, 1976) to return approximately to their starting position. Immediately following this they engage in a goal localisation strategy to pinpoint the exact position of the goal, and thus finally reach it.

While foraging for food, bees and ants continually maintain an up-to-date record of their position relative to home through simple path integration.[68] The insect continuously monitors the angles steered and the distances travelled whilst foraging, and integrates this information to compute the current mean home vector. Both bees and ants use skylight compasses for measuring directions (Wehner, 1989). Following the completion of the stereotypical foraging behaviour, it is this home vector that is followed, in a straight line, back to their starting point (home). This dead-reckoning mechanism, however, is inherently subject to cumulative error, and only capable of returning the insect to a point near home. Having returned to a location close to the nest the insect now utilises an alternative strategy, namely goal localisation.

Both ants and bees are able to accomplish this final goal localisation by using nearby landmarks surrounding the goal to pinpoint and 'home in' on their location (§2.5.9). However, in environments which do not possess any obvious local landmarks, the insect must make use of yet another strategy for finding its nest. In such situations the insect has little alternative but to simply search for its nest.

The desert ant *Cataglyphis* lives and forages in just such an environment (Wehner et al., 1983). The search for food, such as dead arthropods, may take the ant as much as a hundred metres away from the nest and last for up to two hours. Despite the meandering outward path, dictated by its foraging strategy, the ant always follows a direct path back to home using its computed mean home vector. Although the dead-reckoning mechanism is accurate enough to allow returns to the nest within a few percent of the entire homing distance, the

[67] Or, for example, successfully returning to the location of a previously visited food source.

[68] In foraging desert ants this path integration has been shown to be solved, not by true vector summation, but through a computationally simple non-trigonometric approximation (Müller and Wehner, 1988).

ant must still actively search for the small inconspicuous hole which leads to its subterranean colony. However, this search is not simply a random walk strategy based on brownian motion. The desert ant, in fact, engages in a systematic search behaviour[69] which successfully weights the amount of searching in a particular area by the probability of finding the nest in that area (Wehner and Srinivasan, 1981; Müller and Wehner, 1994). This highly efficient search strategy involves the ant performing a number of loops of ever-increasing size, starting and ending at the origin of the search and pointing in different azimuthal directions. This ensures that the centre area, where the nest is most likely to be, is searched the most extensively. Further, this search strategy is effectively very similar to other efficient search strategies based purely on theoretical grounds where the location of the nest is assumed to follow a two-dimensional Gaussian probability density function. The desert ant effectively builds a search density function which successfully mimics the assumed Gaussian probability density function for the location of home, providing a very efficient method of finding its nest when landmark cues are unavailable.

2.5.9 Visual Homing

A large number of experiments (see Collett (1992; 1996) for a review) have shown that many insects are able to 'home in' on a specific location, such as a nest, by using visual cues provided by landmarks in the vicinity.[70] This ability is developed most highly in central place foragers, like bees[71], wasps[72] and ants.[73] In fact, insects use terrestrial landmarks not only for pinpointing the location of important places, but also for guiding their way along frequently travelled paths (Collett et al., 1992).

[69]Similar search behaviour has also been observed in the wasp (Peckham and Peckham, 1905) and the desert woodlouse *Hemilepistus reaumuri* (Hoffmann, 1978; Hoffmann, 1983b; Hoffmann, 1983a; Hoffmann, 1984). The foraging excursions of the woodlouse, however, are much shorter in comparison to that of the desert ant, taking it only a metre or so away from its burrow.

[70]At present there is no conclusive evidence to suggest that insects make use of cognitive maps, rather they navigate using vectors and routes (Wehner et al., 1990; Wehner and Menzel, 1990; Dyer, 1991; Kirchner and Braun, 1994; Wehner et al., 1996). (This is contrary to previous claims that insects, such as the honeybee *Apis mellifera*, use landmark-based cognitive maps (Gould, 1986; Gould and Towne, 1987).)

[71]See (Wehner, 1981; Cartwright and Collett, 1983; Lehrer, 1991; Lehrer, 1993; Brünnert et al., 1994; Zeil et al., 1996).

[72]See (van Iersel and van den Assem, 1964; Zeil, 1993a; Zeil, 1993b).

[73]See (Wehner and Räber, 1979; Wehner, 1983; Wehner et al., 1983; Wehner, 1992).

Ethological experiments seem to suggest that insects search for a specific location by using a form of image matching. The insect behaves as though it is striving to 'home in' by moving in such a way as to maximise the match between the current retinal image and a 'snapshot' of the panorama as seen from the goal, acquired on an earlier visit (Cartwright and Collett, 1983; Cartwright and Collett, 1987). In this way the insect is continually and locally guided by the desire to reduce this discrepancy between current and home snapshots until it becomes zero. Moreover, it seems that insects 'process' the snapshot images for features such as edges. Information such as the position, orientation, range and colour of features can be used (Cheng et al., 1987).

It is also suggested that the insects make use of an external frame of reference, a celestial compass, to keep track of snapshot orientations (Cartwright and Collett, 1983; Collett, 1992; Collett and Baron, 1994; Dickinson, 1994; Collett, 1996). It is already known that insects obtain information on absolute direction by making use of a celestial compass (based on the position of the sun or the pattern of skylight polarisation)[74] as well as a magnetic compass (Collett and Baron, 1994; Frier et al., 1995). Even distant landmarks can be used as an external frame of reference (Wehner et al., 1983). Homing, by familiar landmarks, becomes much simpler and more certain when the snapshots have a common orientational frame of reference.

Visual homing by way of a single memorised snapshot is however, quite limiting. Due to the fact, that as the homing distance increases so too does the dissimilarity between the visual panorama and the snapshot taken from home, eventually visual homing must fail. Visual homing cannot function without at least some correspondence between memorised snapshot and current retinal image. The obvious solution to this limitation is to employ a host of memorised visual snapshots, taken en route, to allow a return journey via a reverse chaining of homing targets. In this case, each snapshot need only be similar to its immediate neighbour to allow visual homing over extended distances. Indeed, biological evidence for homing by multiple snapshots has recently been reported in wood ants (Judd and Collett, 1998; Srinivasan, 1998).

[74]See (Lindauer, 1960; Collett and Land, 1975b; Rossel and Wehner, 1986; Müller and Wehner, 1988; Wehner, 1989; Wehner, 1992; Rossel, 1993; Wehner, 1994; Lambrinos et al., 1998).

2.5.9.1 Snapshot Theory

Cartwright and Collett (1983) describe experiments undertaken to discover how honey bees use nearby landmarks to guide their way to a food source. The experiments involve training bees to collect sugar solution from a feeder location defined only by an arrangement of matt black landmarks. Then, by observing the bees' search behaviour when the food source is removed one can deduce the cues bees use to home in on the learnt location. When the food source is removed in the testing phase, the bees will search mainly at the position where the feeder was previously located. By changing the arrangement of landmarks slightly between training and testing phases the resultant change in bee search behaviour can suggest which landmark details are learnt and used in guiding their return.

For example, when trained on a position defined by a single cylindrical landmark, and tested on the same landmark, the bees search at the expected feeder site. However, when the size of the landmark is changed between training and testing phases, the bees search at a location where the apparent size of the landmark (i.e. the visual angle subtended by the landmark) is identical to that viewed from the feeder site. These experiments suggest that the bees learn only the apparent size and bearing of landmarks as seen from the target location. There is also some evidence that bees learn the distances of landmarks from goal, possibly through cues based on optic flow (Cheng et al., 1987; Lehrer, 1996). Further, in returning to a previously visited goal, a bee appears to move to a position where the retinal image best matches the memorised image of the landmarks.

When trained on a more complex arrangement of three landmarks, the bees search where the compass bearings of the landmarks on its retina are the same as they had been when it was at the food source.

Experiments also suggest that the bearings of the landmarks observed are learnt with respect to external compass bearings. Bees were unable to learn the location of the feeding site if the orientation of the landmark array was varied (Cartwright and Collett, 1983).

Cartwright and Collett (1982; 1983) have also implemented computational models to discover how bees might use a memorised image of the target location to guide their return. The models simulate the situation in which a bee takes a two-dimensional snapshot of its surroundings at its target location and when returning continuously compares this snapshot with its current view. The simulated bee

uses the difference between the two views to guide its return.

The model which most closely mimicked the bee's true behaviour involved computing homing directions from differences between the angular extents and bearings of matched areas in the snapshots. An independent compass was also required to maintain a constant orientation between snapshots.

The homing direction was computed by firstly segmenting the snapshot and retinal image into regions of light and dark. Each region in the snapshot is then paired with the closest region on the retina with the same sign (i.e. dark with dark and light with light). Every pairing (not necessarily one-to-one) then generates two correctional unit vectors, and the bee's direction of flight is given by their sum. One correctional vector is used to improve the angular size of the region and the other its bearing. To improve angular size one moves towards or away from the landmark, whereas to improve the bearing one moves perpendicular to the landmark.

2.6 Behaviour-based Corridor Navigation

Several investigations of 'corridor based navigation' have been carried out, from the behaviour-based point of view. These implementations attempt to use the qualitative elements of optic flow to help visual guidance. For corridor guidance this has, in the majority of cases, been inspired by the visual behaviour of flying insects, specifically honeybees (see §2.5.4). By balancing the lateral optic flow, honeybees are able, very simply, to exhibit a centring behaviour within an enclosed space, such as a corridor.

One such investigation, inspired by this visual behaviour, is by Santos-Victor et al. (1993; 1995) for robot navigation down a corridor. Their experimental setup is based on a mobile platform with two cameras pointing laterally in opposite directions, one at each side wall. When the robot (*Robee*) is in motion, the average image velocities (1D optical flow) seen by the two cameras are compared, in real time, and used to control the robot's direction and velocity. Range, however, is measured in terms of image velocity. No attempt is made to compute range in metric terms. The robot simply attempts to balance the average image velocities observed through each camera. The robot moves forward at approximately 8 cm/s.

The flow difference is essentially treated as a misplacement *error signal* and used by a PID controller to appropriately control the robot's rotational speed:

$$u = K_p \left[e(t) + K_i \int_n e(t-n)\, dt + K_d \frac{d\, e(t)}{dt} \right]$$

where u is the applied rotational speed and $e(t)$ is the misplacement error at time t.

This system suffers from several deficiencies, however. The first is due to the fact that the two cameras are pointed laterally. Although this setup gives sufficient motion information (i.e. optical flow, assuming adequate visual texture) to enable centring within a corridor, it may not provide adequate coverage of upcoming obstacles and other environmental conditions. Since there is no frontal view, the robot is virtually looking where it has been, rather than where it is going. If the corridor environment is fairly constant, then this is not a major problem. The range of the lateral walls will not rapidly change. However, if the corridor is changing significantly the side views will not be very indicative of the upcoming environment. Sharp cornering and the narrowing of the corridor requires the robot to move at a much slower, safer speed.

Another deficiency is that the image velocities seen by each camera are spatially averaged before the inter-camera comparison is made. This is acceptable when navigating down an empty corridor, but when there are obstacles (some nearer, some farther), averaging the image velocity will destroy this structural information. For example, the large image velocity produced by a relatively small, yet near (and potentially dangerous), obstacle may be swamped, or at least made less significant, by the image velocities of more distant objects and background.

A further short coming is that the system does not take robot rotation into account when comparing image velocities. Rotation induces an illusion of greater depth on one side (the side being rotated towards), and of less depth on the other side. Clearly, this effect can be important when trying to navigate around obstacles. Corrective motion generally requires some degree of robot turning and therefore introduces a rotational component to the perceived optic flow. The greater the rotation, the greater the distortion of range perception. However, efforts to reduce this rotational effect have been made through judicious placement of cameras with respect to the centre of rotation of the robot, and control over forward speed. Since robot rotation in this particular system is fairly slow, usually less than 3° per second, and the side views are far removed from the

focus of expansion (FOE), this deficiency is not as significant as it might otherwise be. However, limiting the speed at which the robot can implement corrective action can have an obvious effect on performance. It effectively places further limits on the ability to effectively deal with the speed at which the changing environment is perceived.

Another mobile robot "bee-bot", also inspired by the centring behaviour of bees, is that described by Coombs and Roberts (1992; 1993). It uses low resolution motion vision over large fields of view to steer between obstacles and along a corridor. The system uses a single, forward facing, wide-angle camera with a 115° field of view. This field is split into three equal sub-fields, of which the extreme left and extreme right sub-fields are treated as the left and right fields of view, respectively. The central sub-field is ignored. The response from each field of view is the largest optical flow measured in the field, which usually arises from the nearest obstacle. Active gaze stabilisation is used to overcome the effect of rotation which can contaminate the flow field. The largest optical flow in the left and right fields of view are then compared, in real time, to steer between obstacles. Again no attempt is made to compute metric ranges of objects observed in the image. The processing of optic flow in each field is implemented in two parts. First, gradient-parallel optical flow is estimated, using local neighbourhood operations. Then the maximum flow is identified by examining a histogram of the flows. The robot moves forward at approximately 30 cm/s.

The steering control system is again based on deriving a corrective change in heading given the difference in optic flow observed to the left ω_L and right ω_R:

$$\delta R = k(|\omega_L| - |\omega_R|)$$

where the desired change in heading δR is simply proportional to the difference in lateral optic flow.

This implementation has more recently been augmented with a frontal view (Coombs et al., 1995a). Looming cues, from the frontal view, warn the robot of impending collisions. Corridor following combined with dead-end deflection thus allows the robot to safely wander for extended periods.

This implementation also has some deficiencies despite addressing several earlier ones.[75] Firstly, the optical flow is estimated as if the fields of view were

[75]It should, however, be noted that the works of Santos-Victor et al. (1993) and Coombs and

strictly sideways-looking, even though they are in reality pointed approximately 50° forward of side-looking. Furthermore, the bearing of the perceived motion relative to the heading direction (the direction of FOE) is not taken into account. If two objects produce equal image velocities, the one with the smaller bearing will be nearer (see figure 2.2 and equation 2.21 in §2.5.2). This can be very significant since the maximum flow, observed on each side of the robot, is used exclusively for the comparison and hence the determination of appropriate corrective action. It is assumed that this maximum flow is indicative of the effective range of either side. Or, more precisely, indicative of the range of the closest object on either side. However, as can be seen from equation 2.21, this need not be the case. The angle at which an object is observed can play a very significant role in determining its apparent angular velocity or lack there of. The closest object may not generate the maximum observed apparent motion due to its proximity to the FOE. The closer to the FOE the more significant the effect. In this implementation, the oblique-looking views observe motion from as far as 60° from the FOE, to as close as 20° from the FOE. This implies that the range of the object producing the maximum observed flow can be up to 2.5 ($\sin(60°)/\sin(20°)$) times as far away as the closest object. This problem can of course be compounded (or in fact cancelled out) by the possible delusions on the opposite side of the robot. The objects within an environment can therefore conspire to delude the robot into an incorrect, and possibly unsafe, 'corrective' movement. Although, steering between obstacles with this system can be suspect, basic corridor guidance remains reasonably sound.

The more recent improvement, with regard to the additional frontal view, removes the obvious and debilitating blind spot directly in front of the robot. However, simply adding a frontal view does not significantly help estimates of frontal range, because the frontal view provides only weak looming cues. The frontal view is thus only used to warn the robot of impending collisions. Finally, it can be quite difficult and cumbersome to compensate for rotations through active gaze stabilisation.

Another robot built recently by Duchon and Warren (1994) again uses the strategy of balancing lateral image velocities to centre the robot in a corridor.

Roberts (1992; 1993) were conducted at almost the same time and were derived independently of each other.

Inspired by behaviour-based robotics[76] and the Gibsonian[77] "phenomenal" viewpoint of 'ecological psychology'[78] and how optic flow may be used by animals to directly guide their actions,[79] formal control laws,[80] based on optic flow, are proposed for the control of the robot. To demonstrate the applicability of these laws of control to the domain of behaviour-based robotics, two laws were devised for the obstacle avoidance problem of mobile robotics. The two obstacle avoidance strategies proposed were a "Balance Strategy" and an "Avoid-Closest Strategy". The balance strategy acts to equalise the rate of optic flow observed in left and right halves of the visual field, as observed in honeybees (§2.5.4):

$$\delta R = k(|\omega_L| - |\omega_R|)$$

where the prescribed relative rotation (R) is proportional to the difference between left (ω_L) and right (ω_R) apparent motion speeds. The avoid-closest strategy acts to turn the robot away from the place in the visual field with the lowest time-to-contact[81] (i.e. the closest object):

$$\delta R = k(\frac{1}{t_{min}} \times \frac{1}{pos(t_{min})})$$

where the prescribed relative rotation R is inversely proportional to the lowest time-to-contact t_{min} and the relative angular distance of this point from the heading direction $pos(t_{min})$. The frontal obstacles are essentially detected by measuring looming (image expansion) cues. Unfortunately, looming cues are weak and become detectable only when the obstacle is dangerously close. The robot moved forward at approximately 4 cm/s.

[76] See section 2.1.3.

[77] See (Gibson, 1950; Gibson, 1958; Gibson, 1966; Gibson, 1979).

[78] Animals and their environments are viewed as inseparable. As such, the environment should not be described in terms of physics but rather in 'ecological' terms. Animals perceive phenomena not noumena. The ecological approach *ecological optics* thus proposes a *direct perception* view of animal behaviour (Gibson, 1966; Gibson, 1979). See also (Turvey et al., 1981).

[79] See (Gibson, 1958).

[80] Warren (1988) formalised Gibson's (1958) descriptions of how animals might use optic flow, by proposing laws of control which might regulate the flight of flies.

[81] The time needed to reach the target directly, at constant velocity. See §2.4.1 and (Lee, 1976; Lee and Reddish, 1981; Nelson and Aloimonos, 1989; Enkelmann, 1990; Tresilian, 1991; Ancona, 1992; Burlina and Chellappa, 1993).

2.7 Robot Homing

Robot homing, as the name suggests, involves a mobile robot returning from an arbitrary location to a previously visited location. In this sense it is identical to that observed in many animals (Papi, 1992), including insects (§2.5.9).

Although there are various ways in which robot homing can be achieved,[82] such as through dead-reckoning or by triangulating on radio beacons, visual based homing is perhaps the most interesting. As mentioned previously, much real-world robustness can be achieved through the tight coupling of the senses and behavioural action, with minimal reliance on abstract internal representation. Similarly, using the visually sensed environment to directly control a homing behaviour implies improved robustness with little reliance on artificial information, such as radio beacons, or on explicit internal representations of position that require regular and accurate augmentation, such as map-building or dead-reckoning.

More interestingly, however, is the fact that relatively simple animals, such as insects, also regularly perform visual homing. Insects appear to accomplish this task by simply memorising the appearance of the environment as viewed from home and using the discrepancy between this and what is currently observed, to successfully guide their return. The insect essentially strives to move in such a way as to reduce this visual discrepancy to zero. In this way there is no need for an explicit three-dimensional model of the world and one's location within it. The visually sensed world can simply be used to drive an incremental improvement in homing position.

Apart from the obvious vehicle guidance applications, visual homing may also be used for a variety of other tasks, such as directing docking manoeuvres, tool positioning and grasping.

There have been several previous implementations of vision-based robotic homing, utilising visual *snapshots* of the environment.[83] However, the many slightly different approaches can largely be classified into two main groups.

One main approach is to attempt to derive a correct homing vector directly from the manipulation or analysis of the raw images captured from the differing

[82]Interesting examples of robot homing have also been produced using evolutionary methods (Floreano and Mondada, 1996).

[83]Once again, most of the work in this area has its inspirational roots in insect ethology. In this case, specifically visual homing (§2.5.9).

views of home and current location. These methods — image-based homing — attempt to establish the correct correspondence between views at the pixel or sub-pixel level. The underlying assumption being that the image adjacency will not be altered significantly between views. Over relatively short distances this will generally be true. As an observer moves within a static environment, the image motion perceived will always be in the opposite direction to the direction of movement. Visual motion emerges from the focus of expansion, the direction of movement, and recedes to the focus of contraction. The image adjacency assumption is therefore only violated through occlusion caused by the relative and differing distances of objects in the surrounding environment. Unfortunately, visual occlusion is very common in the real-world. This is especially so in a natural outdoor environment.

The other main approach — landmark-based homing — can be described as the methods which attempt to detect salient features, such as landmarks, in the differing views and then derive a homing solution from the discrepancies herein. Again, a set of correspondences are established between the features in each view. However, in this case, no major assumptions need be made concerning image adjacency. The method simply relies on the reliable detection and matching of corresponding visual features between snapshots.

2.7.1 Image-based Homing

An example of the image-based homing approach is that by Franz et al. (1997a). They reduce the problem of homing to one of optimisation, by assuming that the visual panorama is uniformly distant from the position at which the goal view was obtained. Given this assumption, it is possible to determine how much the robot should rotate (A), the direction in which it should translate (B), and the size of the translation that it should make (C) in order to make the next view resemble the 'goal' view as closely as possible. The expected panoramic view, after a hypothetical movement defined by parameters A, B and C, is computed by appropriately *warping* the current view. The warping function $\delta(\theta)$ can be described as

$$\delta(\theta) = \arctan\left(\frac{\nu \sin(\theta - \alpha)}{1 - \nu \cos(\theta - \alpha)}\right) - \psi \tag{2.22}$$

where θ denotes image position, α the direction in which the robot has moved, ψ the change in orientation, and ν the relative distance moved.[84]

Since the environment is not uniformly equidistant in reality, and absolute range information is not available, this method does not take the robot to the goal in a single step. Successive applications of the procedure allows the robot to home in on the goal. However, as with all procedures that rely on image matching, the method fails when the current and goal snapshots are too different to permit sensible correspondence, or obtain usable values for the parameters A, B, and C which decide how the current snapshot should be warped to obtain the goal snapshot. Thus the size of the area within which the robot, when released, can reliably 'home in' on the goal successfully — the 'catchment area' — can be restricted. With this method one must also choose the instantiations of the image warping parameters A, B, and C judiciously. How much of the search space (the warping space) should be sampled? There is an obvious tradeoff here between computational cost and reliability or accuracy.[85]

Robot experiments were conducted in an arena of size 118 × 102 cm. The homing environment contained several (6) model houses, which were arranged to form an approximate ring around the centre of the arena. A modified Khepera miniature robot,[86] connected to an SGI Indy workstation, is shown to successfully 'home in' on the centre of the arena, from various starting positions.[87] However, the experiments also highlighted the failure of the system to reliably home from a starting point outside of the immediate ring of landmarks. The size of the catchment area thus appears to be limited to the interior region of the homing

[84]The relative homing distance is the ratio between the distance to the goal and the average landmark distance. This relative parameter is due to the equal distance assumption of the visual panorama.

[85]Some simulation results using this image warping method are presented in Appendix B.1. A comparison is also made of the performance and computational cost of the image warping approach and the proposed landmark based approach. As is shown in §6.7.1.2 and in more detail in §B.1, the image based method suffers from a significant computational cost, the reduction of which adversely affects homing performance. The anecdotal examples also show that the effective reliable catchment area, using the image warping approach, is generally restricted to the interior region defined by the immediately surrounding environment. By comparison, the proposed landmark based approach is shown to be vastly superior in both computational economy and homing performance.

[86]A panoramic view of the environment is provided by a single miniature camera placed looking upwards at a conical mirror.

[87]Varying the location of home, the algorithm was shown to perform robustly up to an average of 15 cm from the goal. At larger distances, visual occlusion started to affect homing performance. In an office environment successful homing was achieved at distances of up to 200 cm.

space defined by the closest landmarks. This is not surprising, given the main assumptions of the approach.

Inspired by Cartwright and Collett's (1983; 1987) work with honeybees and the robot homing approach of Hong et al. (1991; 1992),[88] Röfer (1995b; 1995a) presents a similar approach. However, instead of extracting explicit landmarks with which to compute a homing vector, correspondences are made at the pixel level. Assuming the strict pixel topology that neighbouring points in one image will be neighbours in the other, correspondence is computed via a modified version of Kohonen's (1982) self-organising feature map. The modified Kohonen network is first initialised with the target image and then trained with the current image. Successful training results in the weight vectors of the target image moving to corresponding vectors in the current image. Having found a correspondence between the current and target images at the pixel level, every pixel pair is used in the computation of a single homing vector. The method used is essentially the same as that proposed by Hong et al. (1991; 1992). However, in this case the bearing of each pixel is treated as the bearing of a separate landmark. Each pixel, together with its corresponding pixel in the other image, is used to compute a correctional vector. The correctional vector C_i is perpendicular to the bisection of B_T and B_S, as shown in figure 2.3.[89] The summation of these correctional vectors thus provides the homing vector.

Homing experiments were performed with a non-rotating robot which captured panoramic views of the environment through a rotating photoreceptor. Within an environment, consisting of three lamps, the mobile robot was able to successfully home from distances of up to 1 metre away.

Röfer (1997a) describes another slightly different image-based approach to visual homing. Correspondences are again computed between two panoramic images at the pixel level. However, the result is treated as a flow field. The direction of the Focus of Expansion (FOE) between the current and target images thus provides the correct homing direction. Pixel correspondence is achieved via a simulated annealing type approach[90] whereby the area within which pixel correlations can be made, shrinks with time. The algorithm involves an iterative

[88] See §2.7.2.

[89] Again, this is the same strategy used by Hong et al. (1991; 1992). The 'merits' of this strategy are discussed in detail later (§6.4.2.4).

[90] This is again inspired by Kohonen's (1982) self-organising feature map, except with a stricter form of neighbourhood preservation.

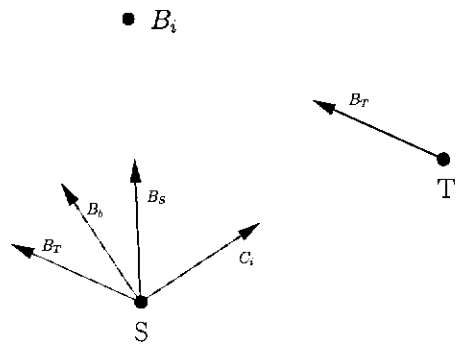


Figure 2.3: Computing a Correctional Vector. The correctional vector C_i, is perpendicular to the bisection of the current bearing of landmark B_i from location S (B_S) and the bearing of the same landmark as observed from home T (B_T). In this case, a correctional vector C_i is computed for each pixel i, the sum of which produces the homing vector.

process whereby a selected pixel is correlated, within the prescribed area of the other image, to find the best match. Each 'best match' results in the linear warping of the image to reflect this correspondence but retain the strict neighbourhood topology of the original image. Having produced a flow field of the virtual motion between target and current positions, the homing direction is computed by summing the (unit) flow vectors.[91]

Homing experiments are performed in the form of path control for a wheelchair traversing within a room and a corridor. A path is prescribed by a sequence of homing targets.[92] Illustrated trajectories of the wheelchair show successful path following in a 7×8 metre room and in a 2 metre wide corridor. Results also show that the directional error is much greater when traversing within a corridor as opposed to an open space. The average directional error within the corridor was 46°, whereas within the room the average error was 11° and 21° (for two different paths). Many more homing targets were thus needed for successful path following within the corridor.

2.7.2 Landmark-based Homing

The homing scheme proposed by Basri and Rivlin (1993a; 1993b; 1995) is based on representing the scene as a set of 2D views and interpolating the novel views

[91]Robot rotation is determined directly from the optical flow field. This rotation component is then subtracted from the flow to leave the purely translational flow field, which is used to determine the homing direction.

[92]This is again very much akin to the work of Hong et al. (1991; 1992).

by linear combinations of the model views. The method uses identifiable features from within the model views to model, under weak perspective conditions, the transformation from one view to another.[93] Unlike other homing approaches, only a limited field of view is used. The actual homing algorithm operates in a very qualitative and simple manner, and is divided into two basic stages. In the first stage the robot fixates on an identifiable point and progressively circumnavigates it (in the appropriate direction) until the point coincides with the corresponding point in the target view. In the second stage the robot simply moves toward or away from the identifiable point until the target position is reached.

Consider the homing task depicted in figure 2.4(a). The model of the scene is composed of two images, P_1 and P_2, separated by an angle γ. The target image P_T is separated from P_1 by an angle θ, and the current image P_C is separated from P_1 by an angle α.

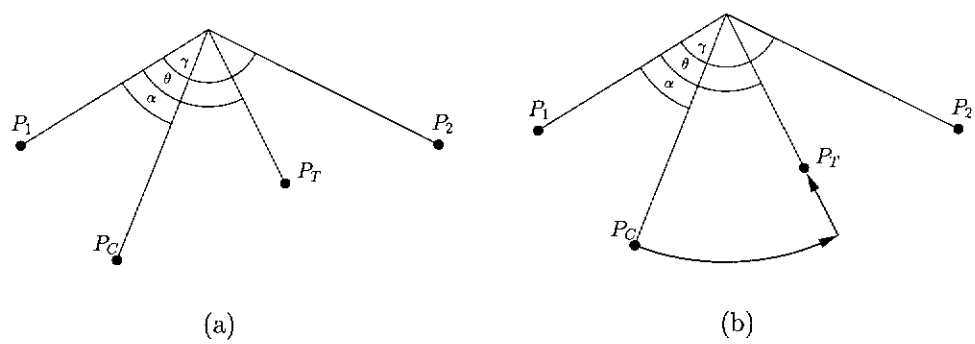


Figure 2.4: A Simple Homing Task. The robot homes in a systematic fashion using the discrepancies observed between a set of model views (P_1, P_2), the target view (P_T), and the current view (P_C). The homing task is divided into two basic stages: (i) the robot fixates on an identifiable feature and proceeds to circumnavigate it, until the feature coincides with the corresponding feature in the target view, and (ii) the robot moves toward or away from the visual feature until the target position is reached (home).

By comparing the transformation coefficients between the model and target image, to the coefficients between the model and the current image, the robot can move in the appropriate direction so as to reduce $|\theta - \alpha|$. The robot takes a new image of the scene after each movement. In the first stage of the homing algorithm each new current image results in a small Δx movement perpendicular to the line of sight (i.e. the fixation point) until $|\theta - \alpha|$ is reduced to zero. The

[93] See also (Wilkes et al., 1994).

resulting path is approximately circular around the point of focus (fig. 2.4b). The line of sight now coincides with the line of sight of the target image. At this point the robot need only advance toward or retreat from the fixated point to find the position from which the target image was taken, i.e. home (fig. 2.4b). Again a comparison of the transformation coefficients is used to determine the correct direction of motion.

Although Basri and Rivlin (1993a; 1995) test their method on real-world images, no results are presented for explicit homing in either simulation or robot experiments.

One of the disadvantages of this approach to homing is the need for appropriately placed model images of the scene portrayed within the target image. There is also the need for identifiable features within the images, for which weak perspective also holds, to enable the interpolation of novel views. The limitations in using a limited field of view are also evident. It seems likely that many 2D model views of the environment are required for robust homing. This is then quite similar to associative homing, also suffering from many of the same drawbacks (see §2.7.3).

Hong et al. (1990; 1991; 1992) have developed a system which uses a landmark-based local homing algorithm to allow a mobile robot to move from one location to another by successively homing in on a sequence of target locations situated en route.[94] In their experiment a mobile robot successfully traverses a corridor by homing on successive target locations which are evenly spaced along the corridor, approximately one foot apart. The Denning DRV-1 non-rotating mobile robot used for the experiment, was equipped with a camera placed looking upward at a spherical mirror, thus producing the necessary panoramic views of the environment.

The homing paradigm is again set around the exclusive use of the two views attained from the target and current positions. Like Cartwright and Collett (1983), Hong et al. (1990; 1991; 1992) attempt to extract a homing vector from the discrepancies observed between corresponding landmarks viewed from the current and target locations.

The key to successful homing is again a set of correct correspondences between the target and current views. In this case, identifiable landmarks are used for

[94]See also (Tsuji and Zheng, 1990b; Tsuji and Zheng, 1990a; Tsuji and Zheng, 1992). Indeed, there is recent evidence that ants use a series of snapshots, acquired at various distances from the goal, for navigation on subsequent trips (Judd and Collett, 1998).

this purpose. The method by which Hong et al. (1990; 1991; 1992) determine landmark correspondence is based on image correlation.[95]

Firstly, a one-dimensional, circular luminance intensity signature, is extracted from the panoramic views, for each snapshot position. This 'snapshot signature' represents the physical position at which it was obtained. Potential landmark positions are then identified, from the current snapshot, by seeking the locations of large intensity changes within regions of monotonically increasing or decreasing intensity. The top fifteen most prominent potential landmarks are then selected as the image features used in the matching process. A correlation function is used to find the best match of the selected image features, in the intensity profile of the current snapshot, within the intensity profile of the snapshot taken from home. In a sense this particular aspect of the process is similar to what is attempted in an image-based approach, and hence suffers from some of the same drawbacks. Image features in the current intensity profile are correlated with image regions, not image features, in the home intensity profile. (Image features are not extracted from the home snapshot.)

Once landmark correspondence has been established the homing vector is primarily computed using a simple vector summation technique. Each pairing of landmarks produces a correctional vector, the sum of which represents an approximate homing vector. Landmark bearings are used for this purpose in a similar way to that proposed by Cartwright and Collett (1983). A correctional vector represents the motion required to improve the landmark bearing disparity observed between a pair of corresponding landmarks. By summing over the complete set of correctional vectors an approximate homing vector is produced. The final homing vector, however, is the result of several minor refinements. The 'merits' of one of these refinements is discussed in detail later (§6.4.2.4).

A slightly different approach to the homing problem is articulated by Pinette (1991). A mobile robot is again given:

1 the bearings of landmarks viewed from the current position,

2 the bearings of landmarks viewed from the target position, and

3 the correspondences between 1 and 2.

However, instead of computing a single homing direction from this information,

[95]See also (Zhang et al., 1991) for a proposed extension of this work.

a range of possible homing directions is logically deduced.[96]

Consider the homing task depicted in figure 2.5(a). The homing objective is again to move in such a way as to reduce the discrepancies between the corresponding landmark bearings observed in each snapshot. In this case to reduce the angular landmark disparities of $\angle A_S A_T$ and $\angle B_S B_T$. To reduce the disparity $\angle A_S A_T$ the robot must move, from starting point S, in a direction defined by the angular interval $(A_S, \bar{A_S})$[97] (fig. 2.6a). Choosing any direction within this region will result in movement of the current bearing A toward the target bearing A_T, thus reducing $\angle A_S A_T$. The same logic can be equally applied to all remaining landmark pairs (fig. 2.6b). Now a feasible region, the range of homing directions which will reduce all landmark disparities, can simply be defined as the intersection of all the valid angular intervals computed for each landmark pair (fig. 2.6c).

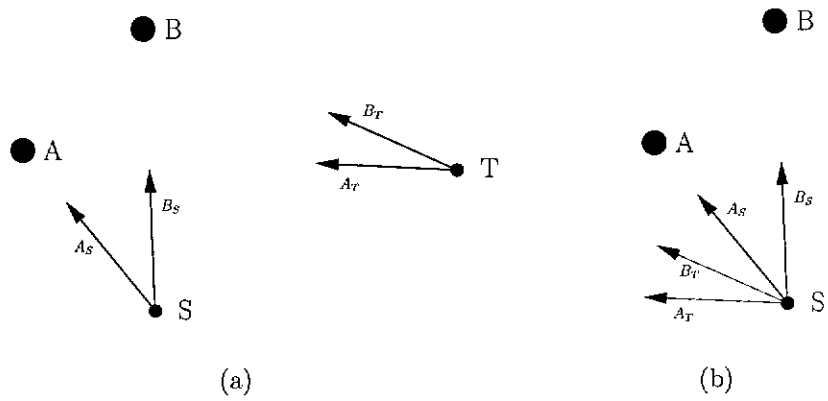
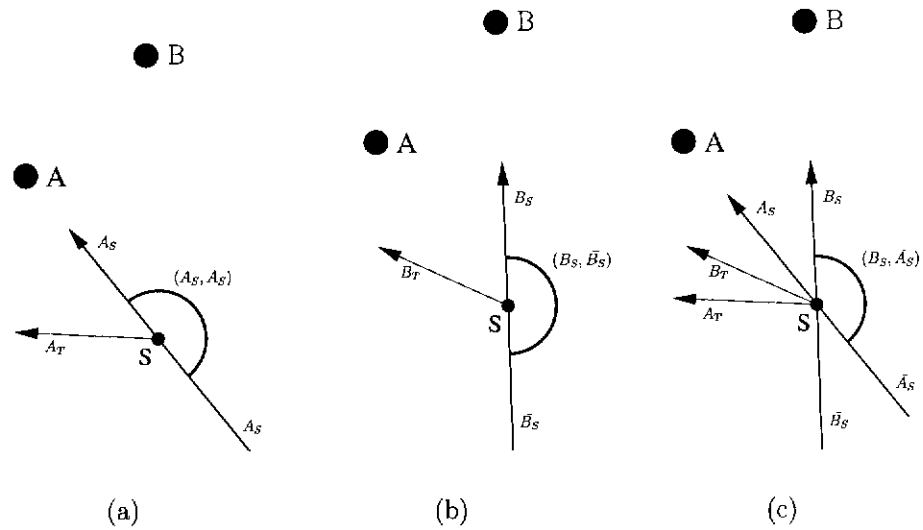


Figure 2.5: An Example Homing Task. The homing task is again defined as moving to reduce the bearing disparities between corresponding landmarks, observed from the current (S) and home (H) locations. In (a) this equates to reducing $\angle A_S A_T$ and $\angle B_S B_T$. For ease of description, (a) can be presented as in (b).

However, simply reducing landmark disparity is not enough to guarantee successful homing. It is possible to continuously choose a direction from the feasible region resulting in the robot never reaching home (i.e. the target position). For example, in figure 2.7(a) if the robot continuously chooses a

[96]This deduction, however, is dependent upon a perfectly correct (consistent) correspondence between landmarks of current and target positions.

[97]The notation used here for an angular interval is in the form (δ, β) where δ defines the counter-clockwise boundary and β the clockwise boundary of the interval.

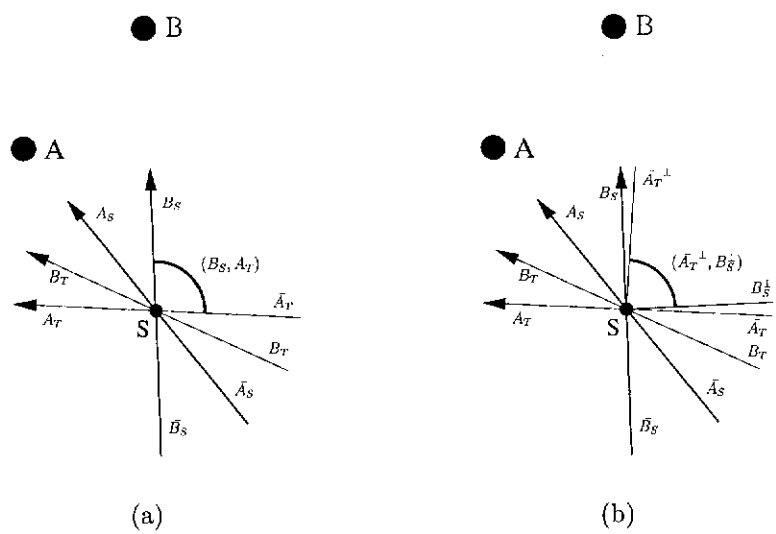


(a) (b) (c)

Figure 2.6: Improving Landmark Bearings. (a) To reduce the landmark disparity between A_S and A_T, the robot must move in a direction defined by the angular interval $((A_S, \bar{A}_S))$. (b) The same logic holds for all remaining landmarks (i.e. B). (c) To reduce the bearing disparities caused by both landmarks A and B, a homing direction must be chosen from the angular interval $((B_S, \bar{A}_S))$.

heading from the angular interval $(\bar{A}_T, \bar{A}_S)$ it will never reach home, despite continually improving the landmark disparities. To guarantee a finite homing path, Pinette (1994) refines the feasible region for a landmark pair to be bounded by the current bearing and the bearing opposite that of the corresponding target bearing. Thus the feasible region now becomes $(B_S, \bar{A}_T)$ as shown in figure 2.7(a).

Having computed a range of plausible homing directions which guarantee successful homing it is now possible to refine this even further (Pinette, 1991). Assuming the feasible homing region is less than 180°, it can be further constrained to also give a guarantee of improving the homing distance as distinct from merely improving the perceived landmark bearings. In essence, it constrains the robot to 'efficiently' home by continually improving its physical proximity to home. This distinction is important in that a movement which improves the bearing discrepancy does not necessarily imply an (immediate) improvement in physical position. To improve the proximity of home the robot must move to a point within the imaginary circle centred at home with radius equal to the distance of home. The range of directions which are guaranteed to (instantaneously) improve physical position is thus given by the intersection of



(a) (b)

Figure 2.7: Restricting the Range of Homing Vectors. (a) To guarantee a finite homing path, a homing direction must be chosen from the angular interval $((B_S, \bar{A}_T))$. (b) To guarantee an (instantaneous) improvement in both landmark bearings and physical location, a homing direction must be chosen from the angular interval $(\bar{A}_T^{\perp}, B_S^{\perp})$.

half planes $H(B_S^{\perp}, B_S)$[9899]and $H(\bar{A}_T^{\perp}, A_T^{\perp})$. The resulting region being the interval $(\bar{A}_T^{\perp}, B_S^{\perp})$ (fig. 2.7b).

Now the range of homing directions that have been deduced not only guarantee successful homing, but also 'efficient' homing.

Interestingly, the method by which Pinette (1994) generates a set of landmark correspondences between two panoramic snapshots, already produces a homing vector. The method involves image warping in a similar fashion to the Franz et al. (1997a) approach described earlier (§2.7.1). The main difference being, where Franz et al. assume a fixed relative 'landmark range' and warp the entire view accordingly, Pinette warps segments of the image at varying hypothesised 'landmark ranges'. In both cases the process automatically generates an hypothesised homing direction, although, Pinette also generates a set of landmark correspondences. At this point it seems a little superfluous to continue with the deduction of a feasible homing region when a homing direction has already been

[98]The notation used here for a half plane is in the form $H(A, B)$ where the half plane is bounded by line A and contains line segment B. In other words, the interval between A and $\bar{A}$ that contains B.

[99]A is perpendicular to $A^{\perp}$.

computed.[100]

As mentioned earlier, the logical deduction of a homing direction cannot be successfully accomplished without a perfectly consistent set of landmark correspondences. Although an algorithm for removing incorrectly matched landmarks[101] is proposed (Pinette, 1994), the ramifications or relative success of this is not clear. The method involves a voting scheme which not only removes incorrectly paired landmarks but also as a consequence correctly paired ones, in an attempt to generate a consistent set of landmark pairs. The overall result of this action on the homing behaviour is not at all clear. Graceful degradation under increasing degrees of mismatch is similarly not apparent.

Unfortunately, Pinette (1994) provides very little in the way of experimental results. Although no results are obtained for explicit homing, a real-world mobile robot was used to gather real data (i.e. snapshots) upon which the algorithms were tested. Interestingly, much more accurate homing directions were generated from the landmark correspondence algorithm than the logical deduction process itself. The image warping process produced a homing vector that was within $\pm 3°$ of the true direction, while the deduction process produced, on average, a feasible homing region of size $80°$.

2.7.3 Associative Robot Homing

Associative vision-based homing is related to the homing methods described earlier, in the sense that it makes primary use of snapshots. However, it differs in the computation of the homing vector.

Associative homing involves taking snapshots at a number of different locations in the environment, and associating each snapshot with a vector which specifies the appropriate homing direction for that location. The homing procedure then involves comparing the current snapshot with those stored in memory to determine the homing direction. This can be done in a number of ways. Nelson (1989a; 1989b; 1991), for example, in a robot arm implementation, uses the homing vector of the associated stored snapshot which best matches the current snapshot. Whereas, Zipser's (1988) simulation averages all homing vectors, weighting each by the degree of match between the corresponding and

[100] In this sense it might be more accurate to classify this work as an image-based approach.

[101] Incorrectly matched landmarks can only be detected when they result in a logical inconsistency, i.e. when they are not consistent with a single target direction.

current snapshot, and then uses this average bearing as the correct homing vector. The observer, essentially, interpolates the direction of home from the distribution of homing vectors associated with the snapshots of the homing area.

The main disadvantages of this associative homing approach are a consequence of the number of required snapshots. Before homing can take place, many reference snapshots must be taken, at reasonably regular intervals, throughout the required homing area. Exactly how many reference snapshots are required and to what density they need to be distributed, for reliable visual homing, will depend largely on the specific homing environment: what is sufficient in one environment may not be sufficient in another. In fact, sufficient distribution may well be far from uniform even within the same homing environment.

Each reference snapshot must also have a correct homing vector associated with it. A correct homing vector, however, need not necessarily be one directly indicative of home. It is sufficient to prescribe a direction which will 'improve' the robot's position, and thus indirectly achieve the homing objective.

Establishing the necessary information base can thus be a costly and time-consuming process, especially if done empirically. This *training period* may well be totally unacceptable in certain autonomous scenarios. The storage and computational requirements of this approach can also be excessive. Every reference snapshot must be stored and every one of these must be matched against the current view in order to generate a homing vector. As a result, these computational costs, without access to parallel architecture, may prohibitively hinder real-time performance.

Chapter 3

The Mobile Robot

Real-world experiments were conducted with a small, relatively cheap,[1] custom-built[2] mobile robot. This robot was specifically designed to cater for the requirements of applying various visually-based insect behaviours in the autonomous robotics domain.

3.1 Hardware and Optics

The mobile robot is tethered to a workstation. The umbilical connection provides the communications link between robot and workstation.[3] Full duplex, asynchronous communication allows the concurrent transmission of both codified commands (e.g. movement instructions) to the robot, and sensed data (e.g. odometry, vision) from the robot.

Robot platform stability is provided by exactly three contact points with the ground, consisting of two drive wheels at the front and a neutral trailing castor at the rear (fig. 3.1). Flexibility of movement is by way of differential steering. The drive wheels are independently controllable, allowing the robot to move along a straight line or a curve, or even spin on the spot. The robot's wheel base is 26 cm wide.

[1]For a price tag of only A$1000 the robot provided an excellent testbed for the various experiments.

[2]The robot was machined and assembled at the Research School of Biological Sciences, Australian National University.

[3]The robot's umbilical line also provides the link to an external power source which provides for all on-board power consumption.

Powered by a pair of 60 rpm electric motors[4] the mobile robot is capable of a maximum forward speed of approximately 15 cm/s. Proficient manoeuvring speed ranges from a slow 5 cm/s to approximately 12 cm/s.[5] Although higher speed motor components (330 rpm) were experimented with, these did not provide enough torque for acceptable performance.

The vision for the robot is provided by a miniature CCD video camera[6] (768 × 576 pixels) with a field of view of 75° × 55°. This camera is placed looking upwards at a mirror assembly[7] which directs two lateral and two straight-ahead views onto the imaging plane of the camera (fig. 3.1 a,c). A pair of mirrored surfaces,[8] arranged in the form of a V, directs the two lateral views of the environment onto two strips of the imaging plane of the camera. Another pair of mirrored surfaces, also arranged in the form of a V, directs light from two additional mirrors oriented to capture two straight-ahead views of the environment. These views are also imaged onto two strips of the imaging plane. The centre-to-centre distance between these two mirrors determines the distance between the axes of the two straight-ahead views. This distance ultimately determines the range of depths that can be computed accurately, as well as the resolution of depth within this range. Each lateral view is 55° × 15° and each frontal view is 38° × 20°. A single camera is used to capture all four views, thus avoiding difficulties associated with balancing automatic gain control systems of multiple cameras. An illustration of the mapping of the four views onto the single imaging plane of the camera is shown in figure 3.1(b). For each view, the motion in the image plane is parallel to the pixel columns. This simplifies the generation, in software, of the reference images required by the interpolation algorithm, to calculate image motion.

On-board hardware[9] is used to control the motors driving the two wheels. The Philips 80C552 8-bit microcontroller also controls all of the on-board monitoring

[4]The 12 Volt d.c. electric motor components are single, encapsulated and sealed combination motor and gear-box units.

[5]At speeds slower than approximately 5 cm/s robot motion becomes somewhat unstable as maintaining forward momentum becomes more difficult.

[6]The Samsung mono chrome CCD camera module is fitted with a 3.8 mm (F2.0) lens, providing a 92° diagonal field of view. The camera module weighs a mere 35 grams and fits inside a box 40 × 45 × 29 mm.

[7]The mirror assembly was constructed from machined aluminium.

[8]Highly polished aluminium surfaces.

[9]The on-board circuitry is the same as that designed for an earlier prototype (see Chahl and Srinivasan (1996)).

functions. Drivemotor speeds are monitored by optical counters attached to the motor armatures, and robot velocity and odometry information is thus provided.

Basic control firmware is supplied by an on-board EPROM. Using a cross-compiler and an EPROM burner, custom-purpose control, diagnostic, and communication software has been written and installed on-board. Motor control is implemented via pulse width modulation (PWM) using PID control. This degree of on-board motor control is essentially to achieve and maintain prescribed wheel velocities.

3.2 Design Considerations

The most important design consideration was the vision system. The design of the robot, especially its novel imaging system, is eminently suited for our corridor guidance experiments. The mirror assembly allows both lateral and frontal views to be projected onto a single camera image. Further, the mapping of the individual views onto the imaging plane is simple and allows straight forward extraction and analysis without the need for any image warping or de-warping. The perceived (horizontal) motion by the robot, in each view, is perpendicular to the image rows on the image plane.

Although providing an elegant sensory solution for the task of visually guided corridor navigation this solution is not ideal for visual homing. The disadvantage for visual homing is the restricted field of view offered by the camera setup. A full 360° view, a panoramic sensor, is what is ideally required for vision-based homing. This can be achieved by setting a camera to point directly up at a conical[10] or spherical[11] mirror. In this way the entire panoramic view can be directed onto a single image. However, due to the absence of a panoramic sensor for the homing task, an alternative, behavioural solution must instead be employed to achieve a panoramic view.

3.3 Software

Communication between the robot and the SGI Indy workstation is via a serial umbilical cable. Codified commands, such as target wheel speeds, are transmitted

[10]See (Yagi et al., 1995; Chahl and Srinivasan, 1996; Franz et al., 1997a).
[11]See (Hong et al., 1991).

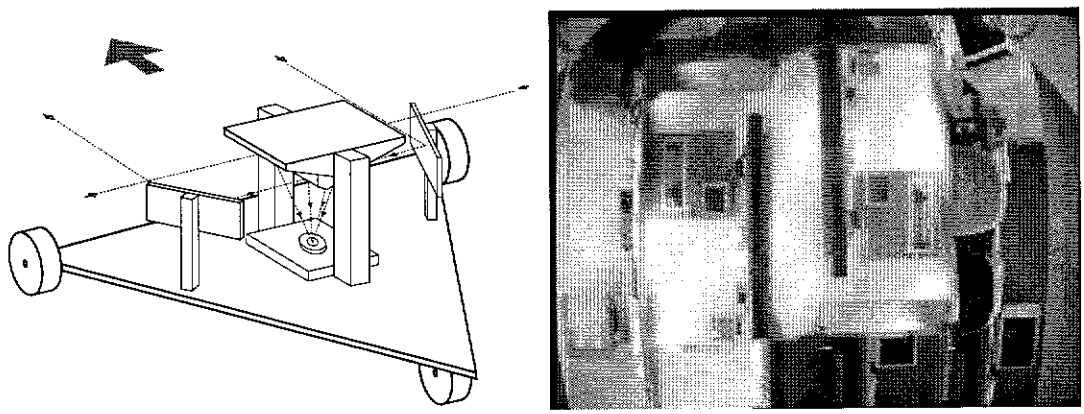
asynchronously to the robot. Correspondingly, the robot also communicates data asynchronously.

The vision information supplied by the on-board camera is also transmitted down the umbilical line via coax to the workstation for processing.

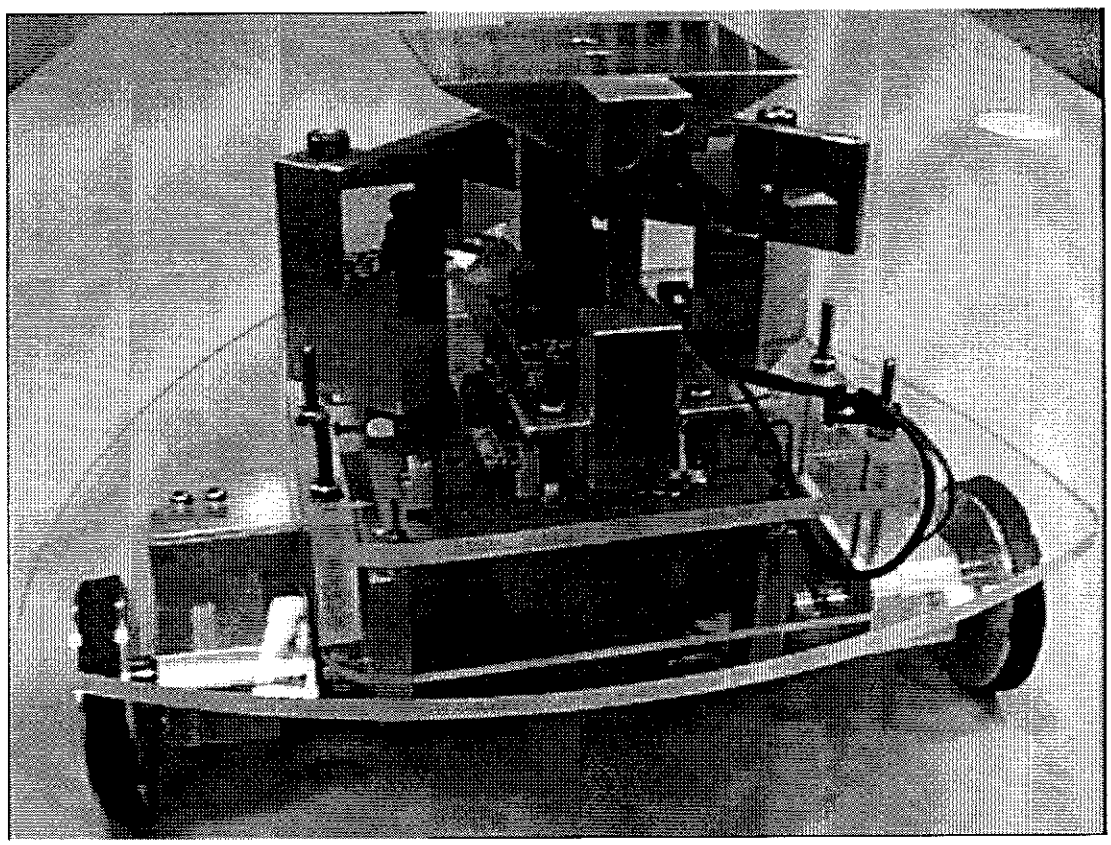
The basic control loop for the robot involves sensing, updating, analysis and finally action. The faster this sensing/action loop can be performed, the better. The sensing stage involves reading current odometric and visual information. The updating stage involves updating egomotion variables, i.e. wheel velocities. The analysis stage involves analysing the sensory input, both vision and egomotion, and determining an appropriate response. Finally, the action, or perhaps more accurately, reaction stage, implements the selected response.

3.4 Conclusions

Despite the physical and visual limitations of the mobile robot it has served its purpose quite well. Through an elegant design, both optic and platform, the mobile robot provided a relatively simple and inexpensive real-world testing platform for the various visually based tasks presented in the following chapters.



(a) (b)



(c)

Figure 3.1: *Kenneth* the Mobile Robot. (a) Using a mirror assembly, a single camera captures the four separate views of the environment; (b) An example camera image showing the mapping of the two lateral and two frontal views; (c) The mobile robot.

Chapter 4

Corridor Guidance

Giving robots the ability to move around autonomously in various real-world environments has long been a major challenge for Artificial Intelligence. To this end, it is vital for robots to be able to perceive their surroundings in 3D; they must be able to estimate the distances of obstacles in their path.

Animals navigate through various uncontrolled environments with seemingly little effort. Flying insects, in particular, are quite adept at manoeuvring in complex, unpredictable and possibly hostile environments.

This chapter will demonstrate, through both simulation and real-world experiments, the feasibility of equipping a mobile robot with the ability to navigate a corridor environment, in real time, using principles borrowed from insect-based visual guidance. In particular, this will be done using the honeybees' visual navigation strategy of measuring object range in terms of image velocity. The viability and usefulness of various other insect behaviours and strategies for corridor navigation will also be shown. These include: (i) keeping walls equidistant, (ii) slowing down when approaching an object, (iii) regulating speed according to tunnel width, and (iv) using visual motion as a measure of distance travelled.

4.1 Introduction

Common approaches to the problem of autonomous robot navigation include the use of specialised equipment, such as laser rangefinders, sonars, and inertial navigation systems. However, such equipment is expensive, and even sophisticated inertial navigational systems accumulate positional error

requiring periodic correction. Alternative approaches, employing passive vision, are attractive, in that visual sensors constitute a rich, yet relatively cheap source of information about the surrounding three-dimensional environment. However, visual sensors are also the ones that entail the most computation. Consequently, there is a strong motivation to explore techniques that make simple, qualitative observations of the important properties of a scene, rather than to use comprehensive, exact approaches that entail needlessly expensive computations.

One way to tackle this problem is to examine how relatively simple animals, such as insects, overcome the problems of autonomous navigation. Given their small size and relatively simple nervous systems, it seems likely that insects employ 'short cuts' to navigate in the real world. These are the principles that may be unearthed and applied advantageously to robot navigation. Recent investigations[1] are now showing what sort of visual cues flying insects use to achieve their navigational prowess.

The main aim of this chapter is to show that very simple motion cues and behaviours, inspired by the visual navigation of flying insects, can be used to provide a mobile robot with the ability to successfully traverse various environments, in real-time. The viability of this approach is demonstrated in both real-world and simulated experiments.

4.2 Background

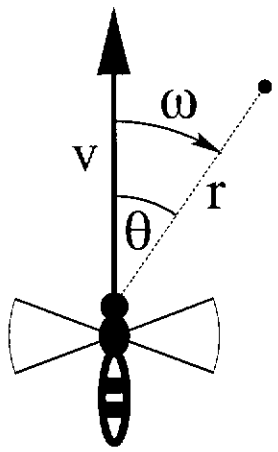
4.2.1 Measuring range via apparent motion

Insects show by their behaviour that they perceive the world in three dimensions. This is accomplished through adept exploitation of relative motion cues. Insects are able to infer the ranges of objects from the apparent motion of their images across the eye (§2.5).

The range (r) of an object can be inferred from its apparent angular velocity (ω), its bearing (θ), and the linear velocity (v) of the eye (see fig. 4.1):

$$r = \frac{v}{\omega}\sin(\theta) \tag{4.1}$$

[1] See §2.5, (Collett et al., 1993; Heisenberg and Wolf, 1993; Hengstenberg, 1993; Srinivasan et al., 1991).



$$r = \frac{v}{\omega}\sin(\theta)$$

Figure 4.1: Range from Apparent Velocity. If the speed (v) of the observer is known then the distance (r) to a static object can be gauged by its bearing (θ) and the angular speed (ω) at which it appears to move.

4.2.2 Insect Behaviour

The following subset of insect behaviour,[2] is considered, as a basis for the development of robust algorithms for corridor traversal.

(i) Peering — before jumping, a locust will sway its head and body laterally to estimate the range of a nearby target in terms of the motion of the target's image on the retina.

(ii) Trying to keep obstacles or walls equidistant — honeybees centre their flight paths between obstacles by balancing the speeds of lateral apparent motion on their two eyes.

(iii) Responding to looming by slowing down — one of the most important flight manoeuvres flying insects perform is that of landing, which is, in many flying insects, triggered by a rapidly expanding retinal pattern.

(iv) Keeping motion speed constant — honeybees regulate flight speed by monitoring the speed of apparent motion. Essentially, they strive to hold constant the speed of laterally perceived motion.

(v) Utilising observed motion as a measure of distance travelled — honeybees measure distances to goals through the integration of apparent-motion speeds observed en route.

[2]See section 2.5 for a detailed discussion of these and other insect behaviours.

4.3 Copying Bee Behaviour

Inspired by the way honeybees use apparent motion cues for a variety of visual guidance tasks, an attempt is made to use the same simple cues and behaviours to provide a mobile robot with the ability to successfully navigate along a corridor environment.

Several similar investigations (see §2.6), also inspired by the navigational behaviour of honeybees,[3] have been carried out. However, as pointed out, these implementations suffer from a variety of deficiencies.

4.3.1 Improvements upon Previous Approaches

To combat some of the deficiencies of earlier approaches the following improvements have been implemented and tested:

(1) In a simple way, the system takes robot rotation into account when calculating range. This is crucial for the robot to be able to change direction as it moves. When the robot is rotating as well as translating, the effects of rotation on the image motion must be discounted before equation 4.1 can be applied to estimate range. The rotational component is most significant in the heading direction. In regions at or close to the Focus of Expansion (FOE), rotation can even cause the perceived image motion to reverse. Robot rotation can, however, also have a major effect on apparent motion even in the side view (i.e. 90° from the FOE). Pure rotation of the robot (i.e. an on-the-spot turn) yields no information on range: all points of the image at a given elevation then move at the same angular velocity. Pure translation, on the other hand, provides the most direct method of computing range from image velocity, as specified by equation 4.1.

The rotation and translation of the robot are calculated by monitoring the rotational speeds of the left and right wheels. The angular velocity of robot rotation is calculated by dividing the difference in wheel coverage by the width of the wheelbase, while the linear velocity of forward translation is obtained from the average speed of the two wheels. The image velocity that would have been obtained with pure translation is calculated by subtracting

[3]Specifically, by the way honeybees use lateral apparent motion to guide an equidistant path between obstacles (§2.5.4).

the angular velocity of rotation of the robot from the measured image velocity. The corrected image velocity can then be used to calculate range. This method avoids the need for active gaze stabilisation.

(2) An image interpolation method is used to calculate apparent motion (see §2.4.2). The advantages of this technique over previous methods are that there is no need for (i) identification or tracking of features, (ii) measurement of high-order spatial or temporal derivatives, or (iii) iterative calculation. Furthermore, unlike earlier approaches, this technique delivers the slope as well as the image motion of a surface. This is extremely useful in the present context of corridor navigation.

(3) Lateral images are used to measure not only the distances to the two side walls, but also to measure their orientation. This additional information can be used to help control the lateral position of the robot, as well as its heading direction.

(4) A 'virtual motion' strategy is presented for the detection of obstacles in the forward direction, such as the end-walls in a T-junction or a sharp L-turn. Two frontal views are captured along parallel axes that are laterally displaced by a known distance. By measuring the apparent motion observed between the two images, it is possible to gauge the distance to the frontal obstructing surface. This strategy is analogous in many ways to the sideways 'peering' head movements of locusts, which serve as a means of estimating the range of frontal targets in terms of the apparent motion of their images on the eye (§2.5.3). Furthermore, measurement of horizontal compression (or expansion) between the two frontal images can provide information on the orientation of the frontally-located surface. This information, in combination with information from the laterally-directed views, can be used to help determine the nature of an oncoming obstacle (obstructing object, sharp turn, shallow turn, T-junction or dead-end), to reduce the speed of the robot, and to control preparatory manoeuvres accordingly.

(5) Further, three additional insect behaviours which also assist corridor navigation, are incorporated. Firstly, that of slowing down in narrow corridors and speeding up in wider ones. This is achieved by regulating the robot speed so as to maintain an approximately constant (maximum) image

speed. Secondly, that of regulating robot speed according to the proximity of frontal obstruction. Thirdly, the use of apparent-motion as a measure of distance travelled, is explored. This alternative distance measure can be very useful for times when wheel-based odometry fails or is inappropriate.

4.4 Proposed Paradigm

4.4.1 Simulation

Early experiments[4] were conducted solely in the domain of simulation, and were therefore not as constrained by practical considerations. This naturally resulted in slightly different methods of image motion estimation and consequently of course control, than what was eventually implemented on the real robot. This thesis, however, presents the simulations and models developed expressly for, and based upon the specifics of the real mobile robot implementation (§4.5).

The basis of the simulation is the modelling of the robot's motion in reaction to visual stimuli observed through several camera views. Raytracing is used to provide the images that would be seen by the cameras. In simulation, each camera had a 30° field of view and provided both frontal and lateral views (fig. 4.2(a)). The simulated environment (i.e. the corridor) is made up of textured walls. The simulations were run on a Silicon Graphics Indy workstation.

The simulated robot is based primarily on the real mobile robot which was used for real-world experiments. The robot moves via two drive wheels, located at the front on either side of the robot. These drive wheels are controlled independently and move at a specified velocity. Turning of the robot is achieved by having one wheel run at a higher speed than the other. In this way the robot is capable of moving in a straight line or moving through a curve, or even spinning on the spot by having the wheels move in opposite directions.

4.4.2 Calculation of Image Motion

As previously described, image motion is computed using an image interpolation technique (§2.4.2). The advantage of this technique for the present application is that it allows relatively cheap and reliable computation of both image velocities and their spatial gradients.

[4]See (Weber et al., 1994; Weber et al., 1995).

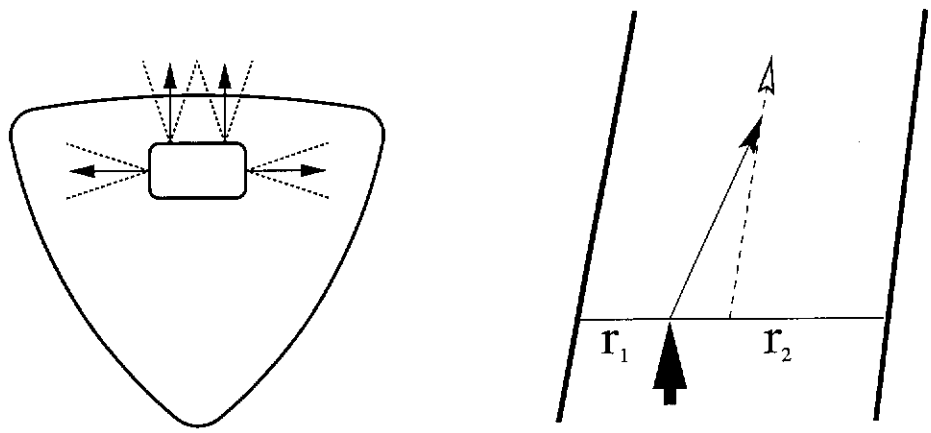
4.4.3 Course Correction

The course correction strategy uses both the range and slope of the walls in determining correct robot heading. Assume, for the moment, that the robot is neither positioned on the tunnel axis nor pointing along it (fig. 4.2(b)). The control rule makes the robot head for a point that is a specified distance away on the tunnel axis. In the present experiments, this distance is set to 100cm. By altering this distance, the rapidity of the control action can be varied. The closer this point of attraction to the robot, the more rapid the corrective action.[5] For implementing the control algorithm, the tunnel axis (dotted arrow) is defined as the line positioned halfway along the width of the tunnel (specified by the mean of the distances r_1 and r_2 measured to the left and right walls, respectively) and oriented in a direction corresponding to the mean of the orientations of the two walls. The apparent speeds and orientations of the two lateral walls are measured by the image interpolation algorithm. Distances are in turn supplied by equation 4.1.

This simple course correction strategy thus ensures that the robot tends towards the corridor centre, and once centred, strives to maintain this centred position. As the dynamics of the perceived corridor change, or as obstacles are observed, corrective action is taken to maintain a centred path whilst avoiding obstacles placed near either side wall.[6] The speed of this corrective action can easily be altered, as required.

[5]Depending on the perceived urgency of required corrective actions, which are usually determined by the dynamics of the environment, this 'gain' can be manipulated to improve overall performance. Depending on the expected or even currently perceived terrain, the judicious manipulation of this additional control variable allows for more sensible behaviour and overall a more robust system. For example, when a straight section of corridor is being traversed the 'point of attraction' for the robot should ideally be further away than for the case of a winding path. In a straight corridor, there is no need for any quick corrective action. Anthropomorphically speaking, the robot is 'kept on its toes' in a state of 'readiness', when there is no need. Consequently, by decreasing the 'urgency' of corrective action, the robot is allowed to 'settle down', resulting in a much smoother path where the natural side-to-side meanderings of the robot are reduced. Ideally, the 'point of attraction' would be perfectly tuned to the specifics of the upcoming environment. However, without perfect world knowledge the upcoming environment remains, to a great extent, unknown, and so care must be taken to allow corrective speed to be high enough to cope with rapid and unforeseen changes in the environment, whilst not too high to introduce severe overcompensation. Care must also be taken due to the fact that a limited view of the environment can cause visual perceptions to change significantly, as the robot moves and changes its orientation. This in turn can cause misconception or misinterpretation in a highly reactive system such as this, if the robot changes direction too far too quickly.

[6]Obstacles are essentially treated simply as extensions of a wall.



(a) (b)

Figure 4.2: Robot Views and Course Correction. (a) The robot is equipped with four separate views, two forward and two lateral. (b) Course correction for the centring behaviour is performed by heading for a point located at a certain prescribed distance along the corridor axis.

4.5 Real-World Robot Setup

The mobile robot[7] used for the real-world corridor guidance experiments is shown in figures 4.3(a,c). Movement is provided by the two separately controllable drive wheels located at the front of the robot. The four required views, two frontal and two lateral, are provided by a single camera placed looking upwards at a mirror assembly (fig.4.3(a)). An example image is shown in figure 4.3(b).

The test environment is provided by cardboard walls covered with artificial (random block) texture (fig. 4.3(d)). Although the degree and type of artificial texture is quite unrealistic, given the practical constraints of the robot and basic assumptions of the approach, this was not considered inappropriate. The direct reliance on visual stimuli, specifically apparent motion, made the use of artificial texture within the fabricated environments mandatory. The specific type of texture was considered unimportant, as long as it was capable of conveying useful motion information. Given the limited processing time available and the purely reactive nature of the system the entire viewable arena was supplied with texture. This ensured apparent motion was always observable throughout the lateral views and thus avoided any unnecessary computation directed towards finding alternate

[7]Described in detail in chapter 3.

motion.

The important point is not how optic flow is measured but given the ability to observe and measure apparent motion, how these visual cues can be used to direct intelligent and useful behaviour. In this case, corridor navigation.

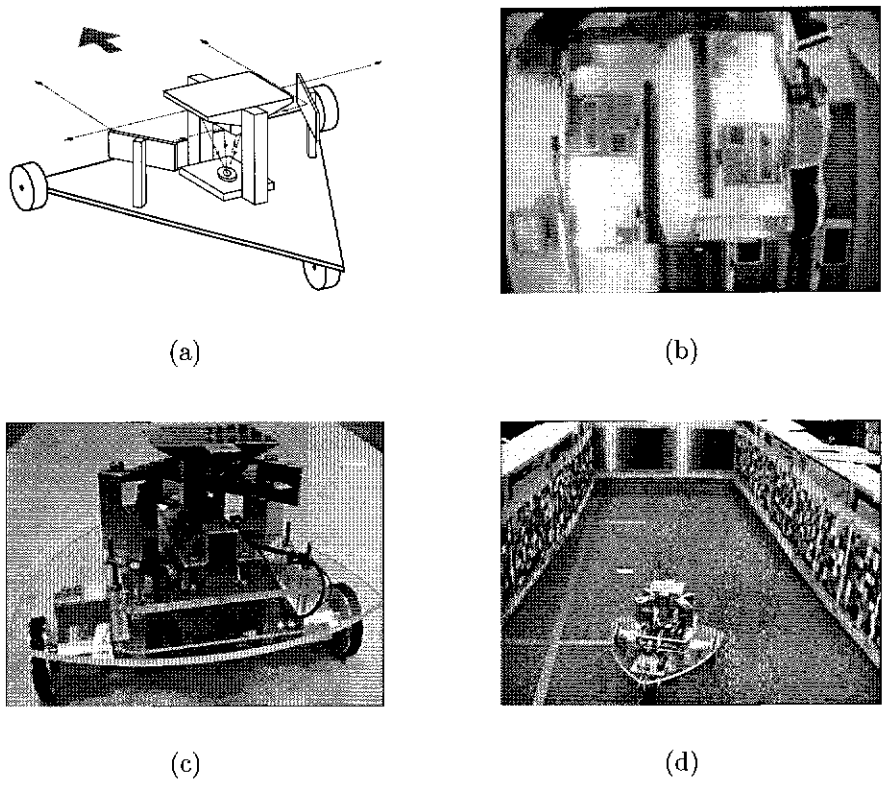


Figure 4.3: Robot and environment. (a) A schematic of the robot shows how the two frontal and two lateral views are captured by a single camera by way of the mirror assembly. (b) An example camera image shows how the four separate views are combined into a single image. (c) The physical robot. (d) The test environment.

4.6 Results

A variety of visually-mediated insect behaviours have been implemented for corridor navigation: (i) keeping walls equidistant, (ii) slowing down when approaching an object, (iii) regulating robot speed by keeping apparent-motion speed constant, and (iv) integrating visual motion as a measure of distance travelled.

4.6.1 Keeping walls equidistant

This centring strategy — keeping walls equidistant — is the basis of the mobile robot's control. Figure 4.4 shows the behaviour of the simulated robot as it traverses various corridor environments. In each case the corridor is 2 metres wide. The trajectories show successive positions of the leading edge (wheel base) of the robot as it progresses up the corridor. The width of the robot trajectory represents that of the robot's wheel base, which is 26 cm wide.

Figure 4.4(a) shows a simple example of the robot traversing a straight section of a corridor. Starting at an off-centre position, it quickly corrects its trajectory by approaching the corridor axis and continues up the corridor maintaining this central position. Note also however, how the centring control rule prescribes proportional corrective action that results in very little unnecessary overshooting of the corridor centre. (In this idealised case, no overcorrective action is observed.) The robot essentially strives to approach the corridor axis without necessarily crossing it. Figures 4.4(b–g) show the behaviour of the robot as it traverses several non-straight corridors at varying speeds. Each of the three increasingly more acute corridor segments are traversed twice. Once at an average speed of 20 cm/s and the other at 40 cm/s. In each case the robot successfully traversed the corridor, avoiding walls and striving to maintain a centred position. At the faster speed of 40 cm/s, it can be seen that the robot tends to understeer more, especially around the sharp corners. This is a natural consequence of having less time to react, but also due in part to the fact that a maximum turn rate has been imposed upon the robot.[8]

Figures 4.5 and 4.6 show several sets of results obtained from the real-world experiments: straight sections of corridor 4.5, curved sections 4.6(a–d), and U-turns 4.6(e,f). In each case the mobile robot managed to successfully traverse the corridor segments by striving to centre itself within the corridor and maintain its centre. This proved quite successful irrespective of corridor width and shape. For these results, the robot moved at an average speed of approximately 10–12 cm/s.

In these 'real world' experiments, however, the environment is very well textured (fig. 4.3d). To test the effect a poorly textured environment can have on a visual stimuli based reactive system, further centring experiments were conducted with reduced texture. In these texture reduced experiments half of

[8]Given that the simulation was to emulate as close as possible 'relevant' real-world conditions and limitations, this was considered an appropriate restriction.

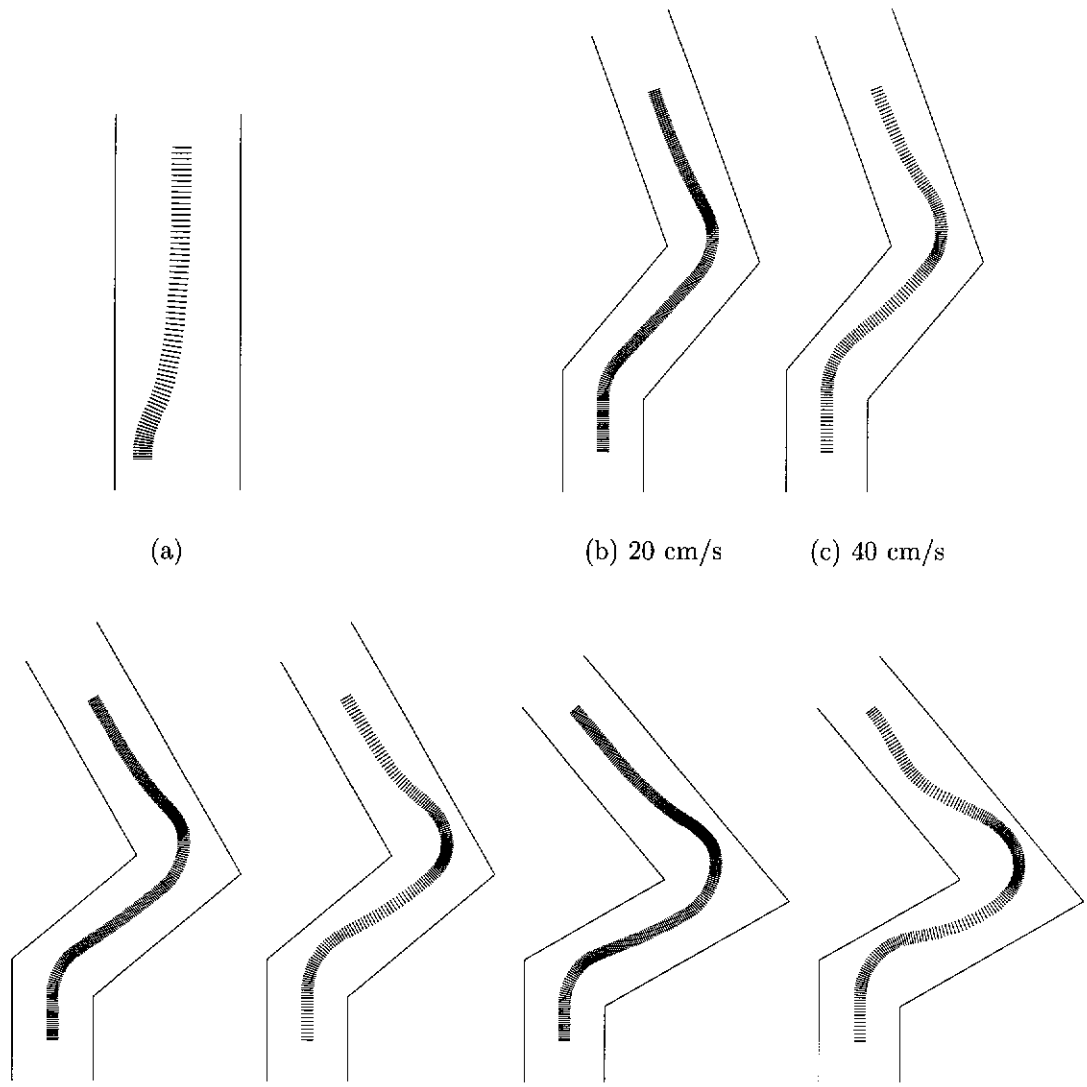


Figure 4.4: Centring Behaviour (simulation). The centring behaviour of the robot simulant is shown in increasingly more acute corridor segments. The effect of robot speed is also shown as each segment is traversed at two different speeds (20 cm/s and 40 cm/s). In each case, the robot starts at the bottom of the corridor and moves upward.

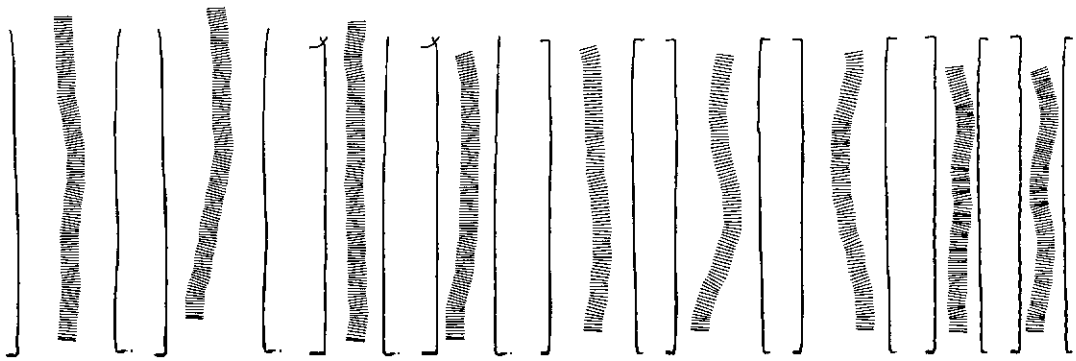


Figure 4.5: Centring Behaviour (real world). The centring behaviour of the mobile robot is shown in various straight sections of corridor. Irrespective of corridor width, the robot is able to attain and maintain an approximately centred position within the corridor. In each case, the robot starts at the bottom of the corridor and moves upward.

the texture making up the environment were removed in A4 sized pieces. The resultant behaviour is shown in figure 4.7. As expected the absence of relatively large chunks of texture from the environment caused significant problems to the purely reactive behaviour of the robot. Errors become much more pronounced and complete failures (i.e. hitting the side walls) also occur.

Although this is indicative of the behaviour expected as a result of the absence of stimuli, it is also somewhat misleading. Given the practical constraints and limitations imposed on the mobile robot this effect is naturally exaggerated. This is due in part by the limited fields of view but mainly by the fact that only a single image patch is analysed for motion. If this small patch is not indicative of the environment then delusional problems will naturally arise. In this case a perceived absence of texture gives no useful information about the proximity or even existence of a lateral wall. This situation may of course be improved through active foveation or simply measuring image motion at more positions throughout the entire visual field. If this cannot be accomplished, for practical reasons, then the purely reactive nature of the present solution may need to be augmented to compensate for a poorly textured environment.

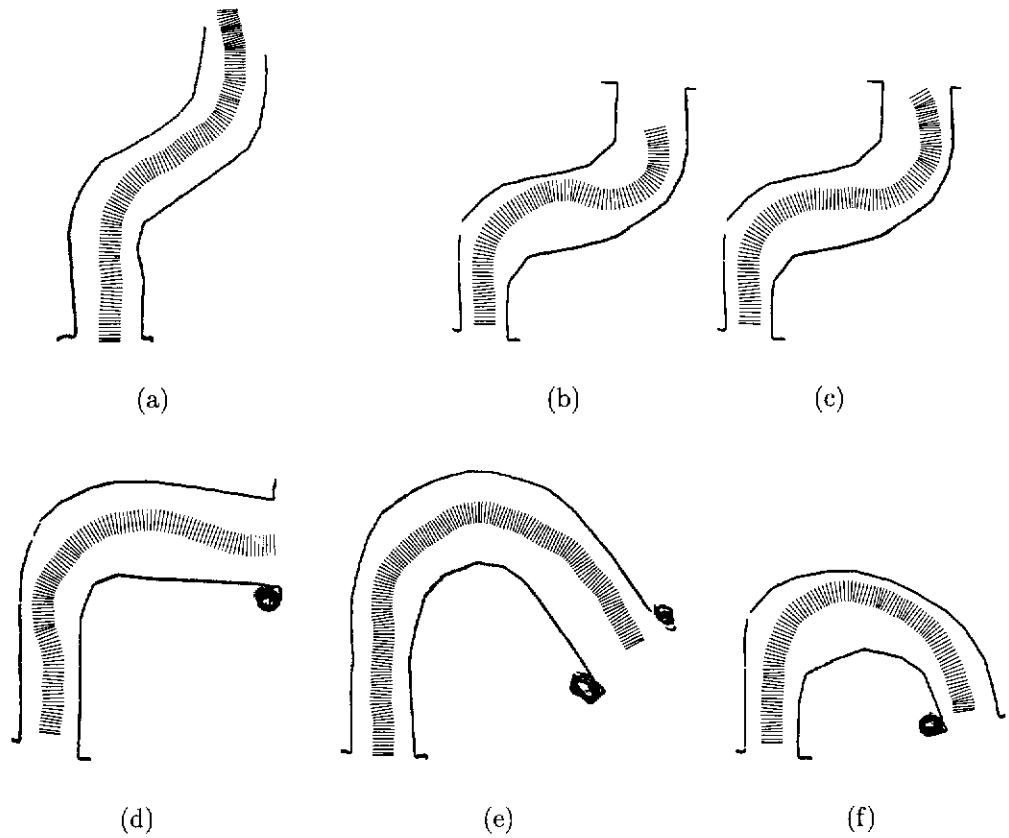


Figure 4.6: Centring Behaviour (real world). The centring behaviour of the mobile robot is shown in various non-straight sections of corridor. In each case the robot is able successfully traverse the corridor segment by continually striving to maintain an approximately centred position.

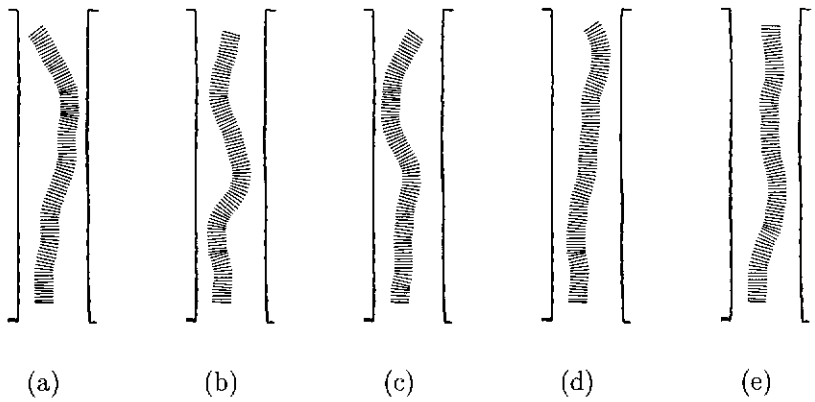


Figure 4.7: Centring in a texture reduced environment (real world). The corridor navigation performance of the mobile robot is shown to be affected by the temporary absence of observable texture.

4.6.2 Slowing down due to frontal obstruction

Figure 4.8(a) shows the simulated robot as it approaches a dead-end passage. As the robot approaches the dead-end it slows down, and will eventually stop and turn around (turning around is not shown). The robot is moving at approximately 40 cm/s before it slows down. Since the motion of the robot is shown by incremental positions of equal temporal spacing, the relative speed of the robot can be seen by the spatial separation.

The mechanism used for slowing down is as follows. If the frontal range is less than some threshold (200 cm) then the regulated speed of the robot is determined by a linear relationship between the maximum speed allowed at this range (30 cm/s) and 0. The task is essentially to slow down gradually as a frontal obstacle is approached. Thus, the manoeuvring speed is determined by the current frontal range (R) and the linear function between 30 cm/s (at range 200 cm) and 0 cm/s (at range 0):

$$TargetSpeed = \vartheta \frac{R}{\rho} \qquad where \quad R < \rho$$

where ρ is 200 cm in this case, and ϑ is 30 cm/s. However, the speed is monitored even at ranges greater than ρ to ensure that the robot is given enough time to slow down. This can be important when the robot is navigating a very sparse environment at a relatively high speed (see §4.6.3, §4.6.4).

Figures 4.8(b–d, e–g) show the same behaviour for the real robot. These demonstrate the use of the frontal views in addition to the lateral ones. In the first example sequence (b–d), the robot travels up the corridor until it perceives the frontal range to be dangerously small at which point it stops (fig. 4.8(b)). The robot then spins on the spot until the frontal range becomes large enough for it to safely resume locomotion (fig. 4.8(c)). Due to the fact that the corridor comes to a dead-end, the robot performs a 180° spin. Figure 4.8(c) then shows the continuation of the robot as it moves back down the corridor, until again it reaches a dead-end, and the procedure is repeated (fig. 4.8(d)). The same stereotypical behaviour is shown in the second example sequence depicted in figures 4.8(b–d).

Due to the physical limitations on the speed of the real robot (12 cm/s maximum) and hence the limited speed range (4–12 cm/s) within which to manoeuvre, the act of slowing down, in response to frontal obstacles, is not clearly visible in figures 4.8(b–g). The robot is essentially still moving at its top speed

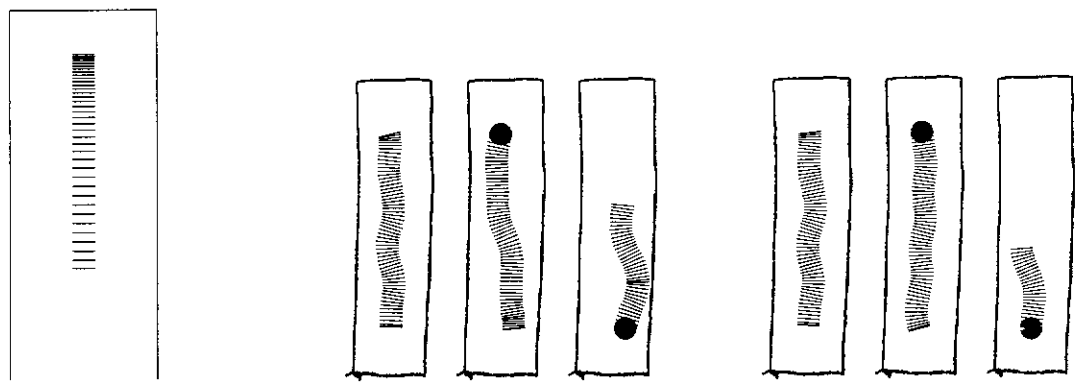


Figure 4.8: Slowing down for frontal obstacles; (a) simulation; (b–g) real-world. Forward speed is regulated according to the proximity of frontal obstructions. Subfigures (b–d) and (e–g) show two separate sequences of the robot behaviour within a dead-end passage. The solid circles are a result of the spinning manoeuvres.

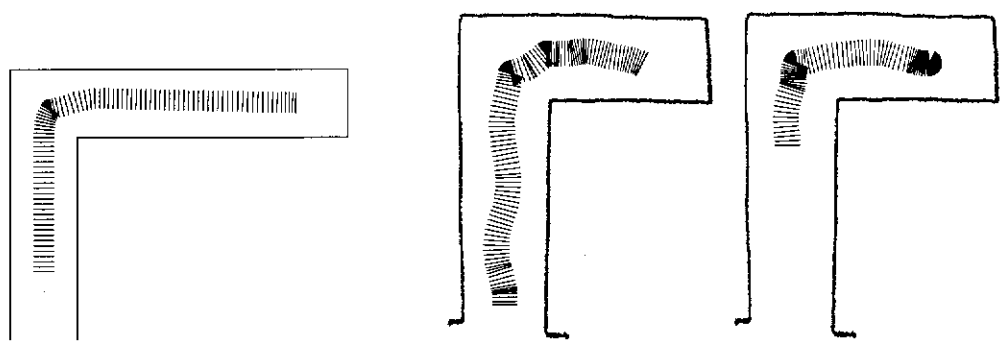
when the frontal object is perceived to be too close for comfort (60 cm) and it responds by stopping, rather abruptly, and then spinning.

4.6.2.1 Turning a sharp corner by spinning

Although, strictly speaking, this is not behaviour copied from insects, it is linked to the previous section on regulating speed according to the proximity of frontal obstruction. When the robot gets itself into a very 'tight' situation where normal manoeuvring is unsafe, it must make use of a different manoeuvre to extricate itself. This has been implemented as a spinning manoeuvre. An example of this was seen in the previous section when the robot approached a dead-end passage.

The spinning behaviour is simply triggered when a frontal obstacle is dangerously close. When the frontal range becomes too small to allow safe manoeuvring (< 60 cm) the robot simply stops and spins on the spot until the frontal range becomes large enough again (> 200 cm) to resume forward motion. This behaviour is shown in figure 4.9, where (a) shows the simulated result and (b,c) the corresponding real-world result.

One difficulty, however, is deciding in which direction to spin. Currently this is determined by the polarity of the average wall slope (i.e spin in the direction of average wall slope). However, this is not a perfect strategy and will not work in all situations. Thus, higher-level processes, not pursued here, would be required to detect and correct erroneous decisions.



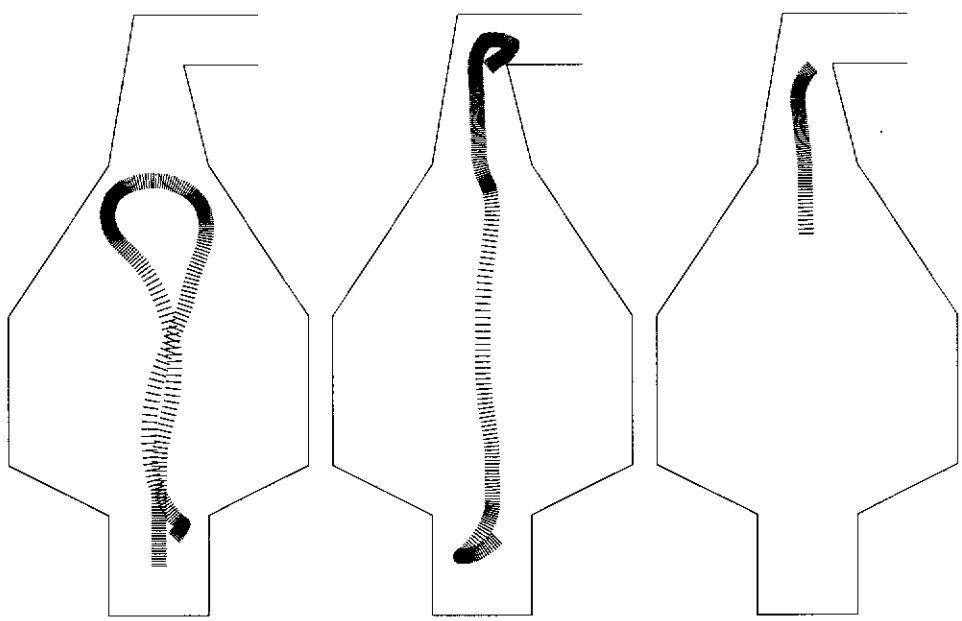
(a) (b) (c)

Figure 4.9: Spinning to aid sharp cornering; (a) simulation; (b, c) real-world. When approaching a sharp corner the frontal range can be used to induce a spinning manoeuvre which can assist in taking a sharp turn.

4.6.3 Tailoring robot speed to tunnel width

Figure 4.10 shows the simulated robot exploring an environment whilst striving to keep the observed (maximum) motion speed approximately constant. This is a very useful strategy for regulating forward speed, in that it forces the robot to slow down in tight situations, such as where the walls of a corridor become quite narrow, and allows a higher speed in more open and 'safer' environments.

In subfigures 4.10(a, b) it can be seen that the robot increases its speed in the wide portion of the corridor and subsequently slows down in the narrow portion. Robot speed also changes significantly towards the end of the run, where frontal obstacles appear. The robot performs several spin moves when the frontal range becomes dangerously small.



(a) (b) (c)

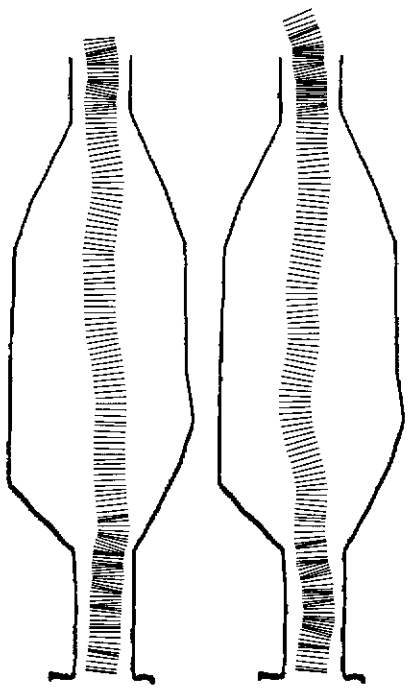
Figure 4.10: Regulating robot speed (simulation). Robot speed is regulated by striving to maintain a constant (maximum) apparent-motion speed. Subfigures (a–c) show the trajectory, in sequence, of a single experiment, with (b) being the continuation of (a), and (c) the continuation of (b).

The behaviour of the robot also illustrates one of the problems with the purely reactive centring behaviour. In the large room the robot tends to get a little lost, as it loses its direction due to the attraction of the large open space in the middle (fig. 4.10(a)). Also, in 4.10(b), it can be seen that as soon as the robot follows

the right-hand turn of the corridor (top), the attraction of the open space to its right draws the robot into a collision course with the corner of the small corridor. This induces another spin manoeuvre, which then sets the robot on a course back down the corridor (fig. 4.10(c)).

This is probably a good situation in which to abandon the centring behaviour and switch to a behaviour in which only one of the walls is followed. Another possibility would be to utilise additional frontal range information for course control; essentially using frontal range to probe for openings. In this way, the attraction of open space directly in front of the robot could offset or override the attraction of open space to the side.

Figures 4.11(a, b) show the real-world results. In trying to maintain a constant maximum speed of observed apparent motions, the mobile robot is clearly seen to change its speed. Robot speed varies from 4–12 cm/s, approximately. In the initial and final stages, where the walls are extremely close, the robot slows down to almost stalling speeds, whereas in wider sections the robot increases its manoeuvring speed to maximum.



(a) (b)

Figure 4.11: Regulating robot speed (real world). Robot speed is regulated by striving to maintain a constant (maximum) apparent-motion speed. Subfigures (a) and (b) show two example experiments; in each case the robot can clearly be seen to appropriately speed up and slow down in response to perceived corridor width.

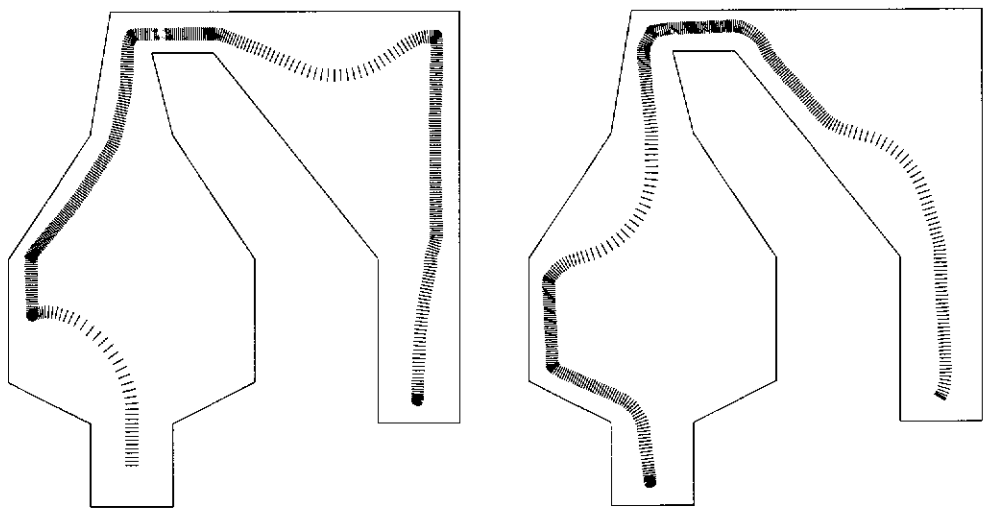
4.6.4 Wall Following

As was seen in the previous section, there is a clear need for a wall-following behaviour. Wall-following has been implemented in essentially the same way as the centring control behaviour. The only major difference is that, instead of using the ranges and slopes of both walls to determine correct direction, only the range and slope of the wall to be followed are taken into account. The robot strives to maintain a pre-specified distance from the followed wall.

The trigger that instigates the wall following behaviour is simply a threshold on the average range of the left and right walls. In essence, corridor width. If the average range exceeds 150 cm the robot will attempt to follow the closest wall. Wall-following is discontinued, and centring resumed when the average range becomes less than a given threshold (125 cm).

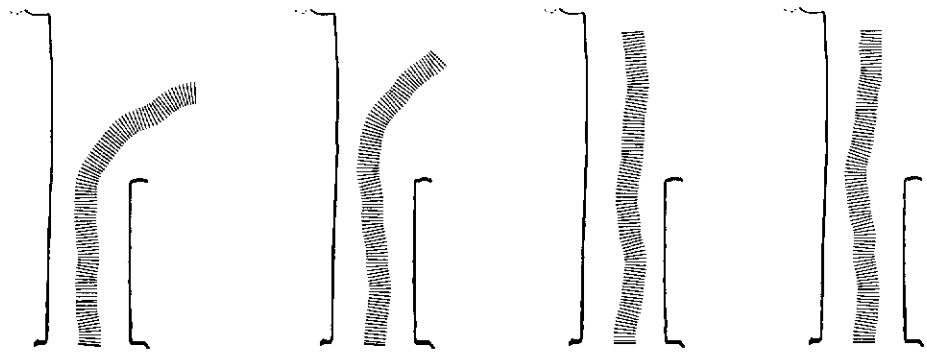
Figures 4.12(a, b) show how well the wall-following behaviour works in the environment that caused problems in the previous section (fig. 4.10). As can be seen, the robot simulant is now able to successfully navigate from one open area to another via a connecting corridor. In large open spaces, where the notion of 'centring' can be considered quite arbitrary, wall following can provide a more secure route. The wall-following behaviour essentially allows the robot to navigate 'around' open areas in a more reliable and predictable fashion.

Figure 4.13 shows the effect of the wall-following behaviour in the real-world experiments. Figures 4.13(a,b) show the original results, without the wall-following behaviour activated. As expected, the robot turns toward the open space just after clearing the right wall. Figures 4.13(c,d), on the other hand, show the behaviour of the robot with wall-following activated. The mobile robot continues its almost straight path, ignoring the space that opens up to the right. The robot tries to maintain a distance of 50 cm from the left wall.



(a) (b)

Figure 4.12: Wall following (simulation). Triggered by corridor width, wall following provides a much more reliable path across a large open space. In this case the robot simulant is able to (a) successfully navigate from one open space (left), through a narrow connecting corridor (top), to another open space (right); and (b) successfully return.



(a) (b) (c) (d)

Figure 4.13: Wall following (real world). (a,b) Without a wall following behaviour the robot naturally heads into open space once clear of the right wall in an attempt to centre its position. (c,d) With a wall following behaviour the robot successfully maintains its straight ahead trajectory ignoring the lateral absence of a wall.

4.6.5 Using image motion to gauge distance travelled

As we move, the motion of the image on our retina provides information not only about the environment, but also about our own motion. As suggested earlier (§4.2.2), integrated optic flow can provide a viable measure of distance travelled. This alternative measure can be very useful when wheel-based odometry fails or is inappropriate. The integrated optic flow as observed in the results presented, is the summation of observed pixel displacements in the left and right fields of view when the robot is in forward motion. When spinning, flow is neither computed nor integrated.

Integrated optic flow provides a measure of distance travelled, but is not directly proportional to it. This is because the flow depends upon the environment—specifically, upon the ranges to the surfaces and objects on either side. However, integrated flow can be used to calibrate distance travelled along a fixed route. Here the repeatability of using integrated flow to measure distance travelled in a constant environment, is assessed over several runs. The expectation is that the natural variation between runs would be averaged out somewhat by the very nature of the integration process. Errors arising from the side-to-side meandering of the robot, for example, would not be integrated into the calculation because, as the robot approaches one of the side walls, inducing greater optic flow on that side, it is simultaneously moving away from the other one, and hence reducing the optic flow induced on that side. This results in a fairly constant average optic flow, which is more representative of pure forward motion.

Figure 4.14 shows a few simple (simulation) runs, over a straight 5 metre section of corridor and table 4.1 shows the results of integrating the optic flow observed en route. In figures 4.14(a, b) the robot is moving at approximately 20 cm/s whereas in figures 4.14(c, d) it is moving at a much faster 40 cm/s.

There are two ways in which the integrated flow values can be computed. First, the integrated flow can be computed as the sum of the left and right apparent angular velocities accumulated over the path. That is

$$flow1 = \sum(\omega_L + \omega_R)$$

Alternatively, the integrated flow can be computed using the sum of the reciprocals of the left and right angular velocities. That is

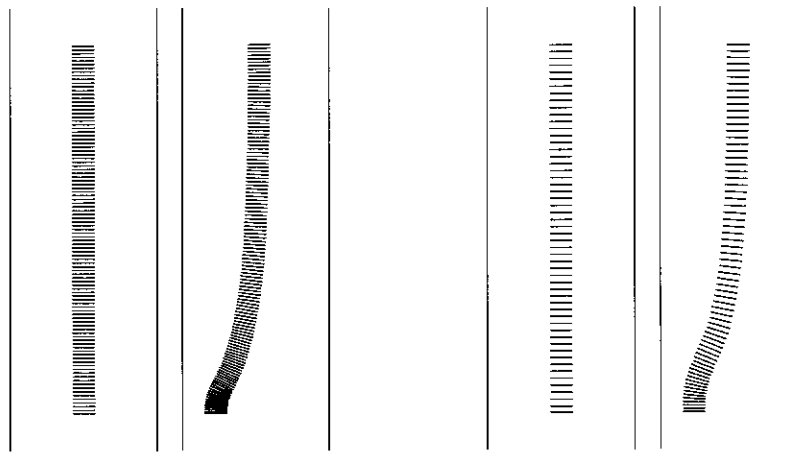


Figure 4.14: Using apparent motion to gauge distance travelled (simulation). An identical corridor is traversed at two different speeds (a,b slower than c,d) from two slightly different starting points (a,c versus b,d). For each run the lateral apparent motion observed en route is summed and recorded.

Run	Speed (cm s^{-1})	Distance (cm)	$flow1$	$flow2$
4.14(a)	20	500.5	9509.5	9509.4
4.14(b)	20	508.6	9955.8	9508.5
4.14(c)	40	501.2	9530.8	9530.8
4.14(d)	40	507.9	10101.5	9502.5

Table 4.1: Integrated optic flow statistics (simulation). The total (lateral) apparent motion observed on each run depicted in figure 4.14 is shown. As expected, $flow1$ provided a good measure of distance travelled, however, $flow2$ provided a better, more consistent measure.

$$flow2 = \sum \frac{4}{(\frac{1}{\omega_L} + \frac{1}{\omega_R})}$$

Note that the denominator of the expression represents the sum of the distances to the left and right walls, that is, the width of the corridor.

Considering first $flow1$, we find that there is a noticeable difference between the integrated flow values (table 4.1) for the straight path (a and c) and the path taken when centring (b and d). This is primarily because as the robot moves closer to one of the side walls, the optic flow from that side increases faster than the flow on the other side decreases. The result is an overall increase in integrated optic flow, for non-centred runs through straight corridors.

$Flow2$, on the other hand, is more independent of the robot's position along the width of the corridor. This is because this measure depends upon the sum of the distances to the two walls, which is the same regardless of the robot's position. As can be seen in table 4.1, $flow2$ does indeed give a more consistent measure of distance.

It is as yet unknown whether honey bees utilise plain integrated optic flow, in spite of the error, or integrate some 'corrected' version of the perceived optic flow, as in $flow2$. Since the bee (or the robot) strives to remain centred within the corridor anyway, this error may well be insignificant with respect to the overall length of the path.

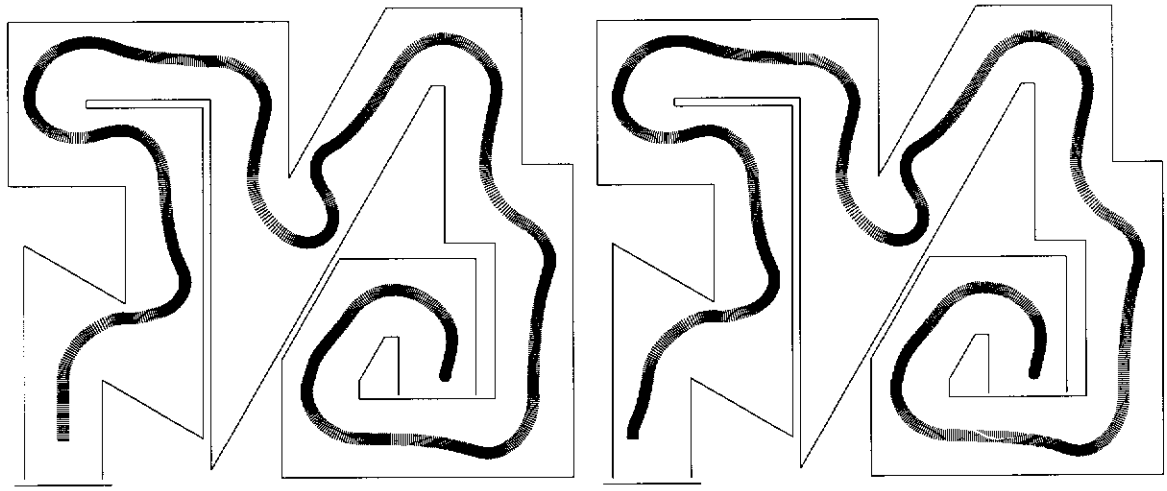
This is illustrated in table 4.2, which compares the results obtained using $flow1$ and $flow2$ for a relatively long circuitous path (fig. 4.15). However, it should be noted that this result is somewhat of an artifact resulting from the very nature of the simulation. Despite the initial position of the robot being different in figures 4.15(a, b and c), the robot quickly 'locks' onto a standard trajectory. As a result, the three paths are in fact very similar, for a major fraction of their length. To ensure a non-standard trajectory the same experiment was repeated varying instead robot speed. The resulting paths are illustrated in figure 4.16. As can be seen in table 4.2, this now illustrates once again the advantage of $flow2$ over $flow1$. Considering $flow1$ and $flow2$ observations from runs 4.15(a), 4.16(a) and 4.16(b), $flow2$ can be seen to have provided a significantly better route distance measure than did $flow1$. When exact traversal paths vary over a given fixed route $flow2$ will generally provide a significantly better measure of route distance than $flow1$.

In the simulation, the paths taken by the robot in figures 4.15 and 4.16 are naturally fairly similar. However, in a real-world situation, the expectation is that the robot's trajectory would be much more chaotic. In this case, the advantage of using flow2 may well be significantly greater than the simulation results suggest.[9]

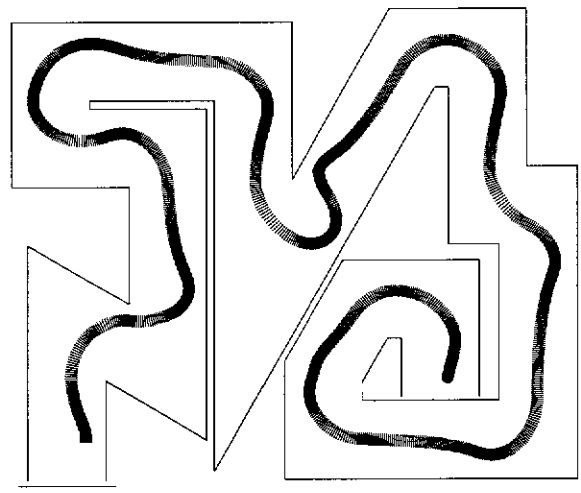
Run	Speed (cm s^{-1})	Distance (cm)	$flow1$	$flow2$
4.15(a)	20	5716	98049	89444
4.15(b)	20	5725	98544	89899
4.15(c)	20	5716	97676	89298
4.16(a)	30	5969	103031	86582
4.16(b)	10	5441	92811	90142

Table 4.2: Integrated optic flow statistics (simulation). The total (lateral) apparent motion observed on each run depicted in figures 4.15 and 4.16, is shown. As expected, $flow2$ again provided a better, more consistent measure of route distance.

[9]Other than sheer simplicity, $flow1$ does not seem to have any advantages over $flow2$.

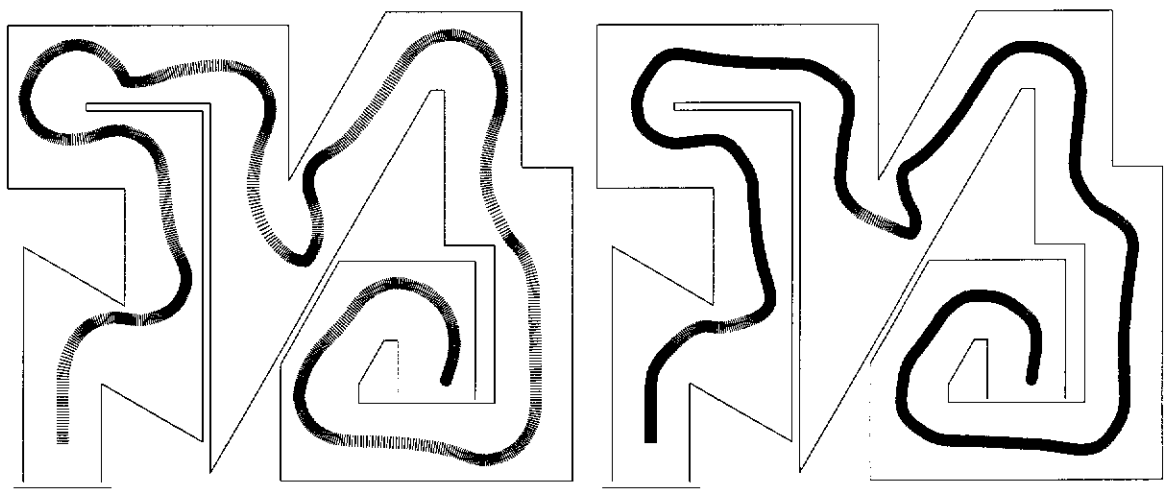


(a) 20 cm s^{-1} (b) 20 cm s^{-1}



(c) 20 cm s^{-1}

Figure 4.15: Using apparent motion to gauge distance travelled (simulation). In each of the above cases the robot simulant starts at a slightly different starting point, but quickly 'locks' onto a common trajectory. Despite the circuitous nature of the corridor the integrated optic flow observed en route is thus expected to be very similar. The robot maintained an average speed of 20 cm/s in all three examples.



(a) 30 cm s^{-1} (b) 10 cm s^{-1}

Figure 4.16: Using apparent motion to gauge distance travelled (simulation). Slightly different trajectories are induced by altering the robot's speed. In (a) the simulant averages 30 cm/s, and in (b) 10 cm/s. These can also be compared with the 20 cm/s trajectory illustrated in figure 4.15(a).

The experiments in figure 4.17 show the behaviour of the real robot traversing a given section of corridor repeatedly. In experiments (a–d) the robot consistently starts in the centre of the corridor and performs its standard centring behaviour whilst maintaining an average velocity of approximately 12 cm/s. Experiments (e–p) show the behaviour when the robot is started off-centre. Finally, the group of experiments (q–t) show the same robot behaviour whilst maintaining a slower speed of 7.5 cm/s.

As expected, the real-world results (table 4.3) do confirm the simulation observations and show that integrated optic flow does indeed provide a consistent measure of physical distance travelled in a static environment. Also from table 4.3 it can be seen that $flow2$ does provide a slightly better, more consistent measure than $flow1$.

Run	Min.	Max.	Mean	Standard deviation	SD as % of mean
4.17(a–d)	22.624 (21.404)	23.831 (21.806)	23.187 (21.588)	0.551 (0.155)	2.375 (0.717)
4.17(e–h)	22.661 (20.962)	25.469 (21.230)	24.093 (21.137)	1.008 (0.103)	4.185 (0.489)
4.17(i–l)	22.991 (21.120)	25.411 (22.022)	23.641 (21.418)	1.023 (0.354)	4.329 (1.655)
4.17(m–p)	22.148 (21.034)	23.215 (21.393)	22.771 (21.212)	0.398 (0.129)	1.748 (0.607)
4.17(q–t)	21.885 (19.154)	24.406 (20.632)	23.157 (20.046)	1.004 (0.580)	4.334 (2.893)
4.17(a–t)	21.885 (19.154)	25.469 (22.022)	23.370 (21.080)	0.956 (0.629)	4.089 (2.982)

Table 4.3: $Flow1$ (and $Flow2$) statistics (real world). This table shows the integrated (lateral) optic flow statistics for the experiments depicted in figure 4.17. As can be seen, $flow2$ provided a better, more consistent measure of distance travelled, than did $flow1$.

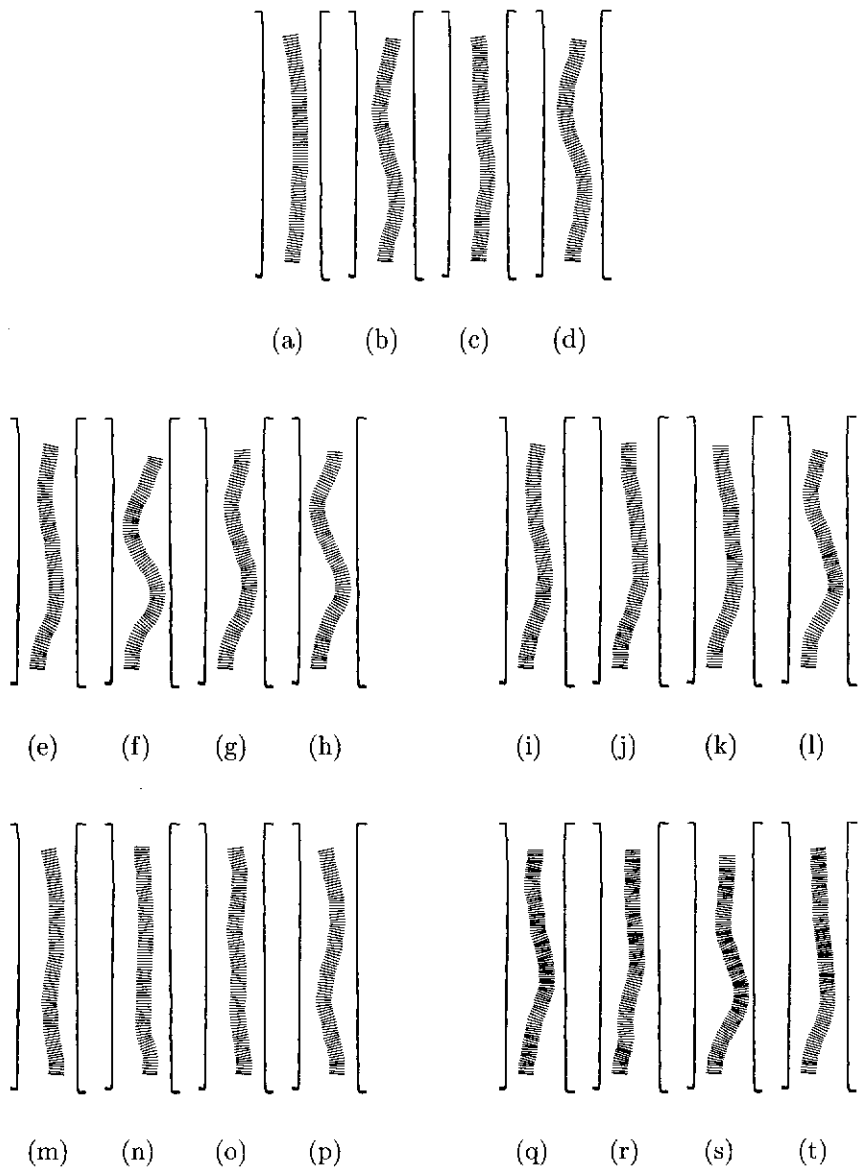


Figure 4.17: Optic flow integration experiments (real world). Repeated exposure to the same route allows a statistical analysis of how well an 'integrated optic flow' measure of distance travelled coincides with true physical distance. Note that the example runs, shown in groups of four, were selected to encompass a variety of robot speeds and starting positions.

4.7 Experimental Limitations

In spite of hardware limitations, both robotic and computing, the real robot showed quite satisfactory behaviour and control. Indeed, the control was not as 'tight' as in simulation, however, this was not unexpected due to the slower control loop.[10] In the real-time experiments the frequency of the control loop was approximately 3.2 Hz for the basic centring (i.e. analysing only side views) and 2.3 Hz for the full system (i.e. analysing both frontal and side views). In comparison, the frequency of the control loop for the simulation was 12.5 Hz. Naturally, this allowed for much more reactive and robust behaviour.

Unfortunately, due mainly to practical restrictions such as a limited umbilical cord length, the physical experimental space was also quite limited. Consequently, various behaviours were tested and observed in isolation. Further large scale experimentation would have been useful in observing the interaction and interplay of all the implemented behaviours at the one time. With all the behaviours activated at once a more holistic 'behaviour' could have been presented.

4.8 Conclusions

In this chapter, it has been shown that very simple motion cues and behaviours, inspired by the visual navigation of flying insects, can be used profitably to provide a mobile robot with the ability to traverse a corridor environment.

Specifically, through both simulation and real-world experiments, the feasibility of equipping a mobile robot with the ability to navigate a corridor environment in real time, using principles based on insect-based visual guidance, has been demonstrated. In particular the viability and usefulness of various insect behaviours has been shown: (i) utilising apparent motion to gauge range, (ii) keeping walls equidistant, (iii) slowing down when approaching an object, (iv) regulating speed according to tunnel width, and (v) using visual motion as a measure of distance travelled.

Providing an autonomous robot with a real-time navigational control system, capable of operating successfully in the real world, constitutes a major challenge for AI and robotics. It is in this area that insect based navigation has something of value to offer. As has been shown, useful behaviour can be achieved using

[10]The control loop is the entire repetitive process of grabbing images of the environment, analysing them and performing some corrective action.

relatively simple strategies and mechanisms. Clearly, there are benefits to be gained by emulating simple biological systems.

Chapter 5

Corridor Discrimination

For a mobile robot to perform some sort of useful 'global' navigation function it usually must have some sort of global understanding of its environment. This is usually expressed in the form of a map.

In the previous chapter a mobile robot, inspired by insect visual guidance, was shown to be able to navigate safely along corridor-type environments. This was achieved through simple and intuitive behaviours observed in insect visual navigation. By observing and reacting to observed apparent motions in a reactive, yet intelligent, way, the robot was able to exhibit useful corridor guidance behaviours at modest expense.

This chapter presents real-world experiments, using the same robot and the same set of behaviours, to show how the robot might extend its knowledge of its environment in a way to obtain a more global view. The mobile robot's world knowledge is extended by the ability to recognise and discriminate between previously encountered corridor environments. This is achieved through the building and refinement of maps based on simple, and homogeneous, landmarks observed en route.

Although, this chapter is somewhat of a diversion from the main theme of this thesis, it does show, that the simple reactive behaviour of corridor following can be seamlessly augmented, using traditional AI techniques, to give a more 'global' goal oriented capability.

5.1 Robot Augmentation

Inspired by the way flying insects navigate through the real world, with seemingly little effort, a mobile robot has been equipped with the ability to autonomously navigate along corridor-type environments using only the apparent motion observed, while in motion.

However, for a mobile robot to do something more useful usually implies knowing, to some degree, its location. Being able to move from location to location, purposefully, requires some knowledge of the world and one's place within it.

In an effort to augment the purely reactive behaviour of the robot, the possibility of building some form of representation (a map) to adequately describe specific sections of corridor has been explored. This will allow the recognition of previously traversed corridors.

Given the inherently structured nature of corridors, corridor recognition could be quite efficient in an indoor structured environment, such as a complex complex. Assuming corridor-type structures connect various parts of the complex, it could be easier to recognise a connecting corridor, and hence one's location, than to try to recognise some unstructured area at the end of a corridor.

However, due to the structured nature of corridors, it can also be quite difficult to recognise a specific piece of corridor by shape alone. Although the shape of a corridor can be quite a useful discriminating feature, and hence assist recognition, it can also be very ambiguous. Some other discriminating factor must be included.[1] Simple landmarks seem very useful for this purpose. A pattern of landmarks, together with the shape of the corridor, can quite easily and dramatically reduce possible ambiguities.

5.2 Insect Behaviour

Dead reckoning is used by several insects to aid in navigation. The desert ants of the genus *Cataglyphis*, for instance, have been shown (Wehner, 1972) to use a vector navigation strategy to keep a bearing and distance of their nest, when out

[1] Although promising results were obtained in simulation, the somewhat erratic nature of the robot's path, in the real-world experiments, also made a pure path-representation approach, quite clumsy, and frankly untenable. Other environmental features, in this case simple landmarks, are needed to viably segment the robot's path.

foraging, to allow an efficient return when finished. Expending the least possible amount of energy and time, under the hot sun, is clearly a worthwhile objective.

Landmarks themselves are also used by insects. Many insects, especially foraging insects such as bees (Cheng et al., 1987; Lehrer, 1993), wasps and ants (Wehner and Flatt, 1972), exploit landmark cues to *pilot* their way back home. A piloting-by-familiar-landmarks strategy enables the insect to find its way by linking the currently observed landmark panorama with the memorised panorama observed around home and en route.

5.3 Landmarks

Inspired by the fact that insects can guide their way along a previously travelled route via the association of landmarks observed en route, it seems reasonable to assert the same for robots. While the methods by which insects and robots operate, in this regard, need not be the same, the principle is sound enough.

Although corridor discrimination is a slightly different problem to that of path guidance, the same principles still apply. Having the robot's path confined to that of a corridor can also simplify the task.

5.4 Robot Setup

As corridor discrimination, through the observation of simple landmarks, is built seamlessly on top of the already established corridor navigation, the robot setup remains identical to that described earlier (chapter 3 and §4.5).

The side views are again used to measure the lateral apparent motion required for the corridor centring behaviour. However, they are now also used in the detection of simple landmarks. The frontal view, on the other hand, while being used to provide virtual motion information for the estimation of frontal range, does not play a role in the detection of landmarks. It is used purely to aid visual guidance.

5.5 Map Building

In an effort to augment the purely reactive nature of the mobile robot, a map building and refinement regime using simple landmarks has been implemented.

Simple dark and light patches in the environment were chosen to be landmarks, as these can be detected both quickly and cheaply.

In this implementation, wheel-based odometry is used to provide the data to allow dead-reckoning between landmarks. Despite the cumulative error that is usually associated with dead-reckoning of this kind, it is not significant in this case. Given that dead-reckoning is really only used between a pair of, hopefully not too distant, landmarks, the cumulative effect of error should be negligible.

In order to assemble a set of landmarks and put them into a useful form, the robot endeavours to build and refine a map. The map consists of a consecutive list of dead-reckoned landmarks as observed en route. Each landmark node records the (dead-reckoned) position at which the landmark was observed. Since only the lateral robot views are used for this purpose, each landmark node also records the direction in which the dark/light region was observed. This is done by recording the current robot orientation together with the side of the robot on which the landmark was observed. The cumulative odometry is recorded, allowing analysis of distances travelled between specific, adjacent landmarks. A 'reliability' measure has also been attached to each landmark node. This serves as an estimate of the importance of the landmark when matching and refining the relevant map.

Thus each landmark node consists of (i) its relative (x, y) position, (ii) the side of the robot the landmark was observed from, (iii) the cumulative robot orientation relative to the starting position, (iv) odometry, and (v) the reliability count for the node.

Thus a typical node m_i is given as:

$$m_i = node((x, y), side, orientation, s, count)$$

A map M consists of a sequence of nodes m_i and is given as:

$$M = \{m_1, m_2, ..., m_n\}$$

5.5.1 Matching

For the purpose of matching the current map (pattern of landmark nodes observed en route) to previous 'learned' maps, a straight forward depth-first tree searching algorithm has been implemented. Utilising a divide-and-conquer strategy an

attempt is made to find the best match between two maps, relatively cheaply, by matching pairs of landmark nodes. Consequently, simple matching can be performed incrementally, as landmarks are encountered, making it more amenable to a real-time implementation.

Specifically, the degree of match between two pairs of landmark nodes is calculated as a 'probability', assuming a gaussian error distribution (fig. 5.1). The match probability is calculated using the displacement error between the landmark nodes and the relative angles of the observed landmarks. A partial path match, between two sets of adjacent landmark nodes, is thus judged by comparing the euclidean, dead-reckoned, displacements between each pair and the relative angles at which the landmarks were observed between each pair. A standard deviation of 20cm and 30° is used for the displacement error and relative landmark-angle error, respectively. (This was estimated from empirical observation.) Given an error of zero, a match probability of 1.0 is returned. As the error increases the match probability tends toward zero, based on a gaussian distribution. The algorithm to compute the degree of match is shown below:

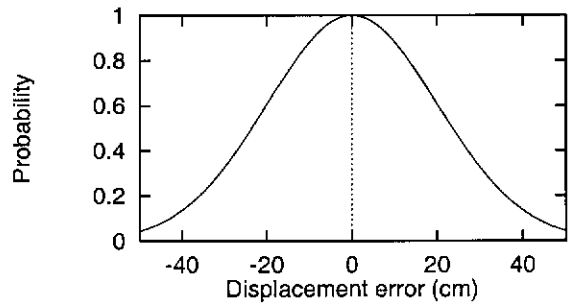


Figure 5.1: Gaussian error distribution

Learned Map: $\{m_1, m_2, ..., m_n\}$
Traversed Pattern: $\{p_1, p_2, ..., p_m\}$

node_pair_prob$((m_1, m_2), (p_1, p_2))$

node pairs are first translated so that both m_1 and p_1 are at origin $(0, 0)$
$D = (x, y)$ positional displacement between m_2 and p_2
$A_m = m_2 orientation - m_1 orientation$
$A_p = p_2 orientation - p_1 orientation$
if $(m_1 side \neq m_2 side)$ then $A_m = A_m + 180$
if $(p_1 side \neq p_2 side)$ then $A_p = A_p + 180$

$A = A_m - A_p$
if ($|A| > 180$) then
  $\pm$ multiples of 360, until $|A| \leq 180$
P_1 = probability of match based on D,
  assuming a gaussian error distribution ($\mu = 0, \sigma = 20cm$)
P_2 = probability of match based on A,
  assuming a gaussian error distribution ($\mu = 0, \sigma = 30°$)
if ($m_1 side \neq p_1 side$ and $m_2 side \neq p_2 side$) then
  $P_2 = P_2 * 0.1$
$node\_pair\_prob = P_1 * P_2$

The matching algorithm is essentially a recursive process. At every level of the recursive matching procedure (each level of the search tree) there are several matching possibilities which are examined:

(i) identical (M_1) — A one-to-one correspondence between nodes (fig. 5.2). Two strictly adjacent landmarks (m_i, m_{i+1}), from the 'memorised' map, are matched against two equally adjacent landmarks from the currently observed 'pattern' (p_j, p_{j+1}).

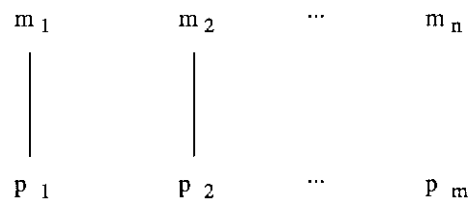


Figure 5.2: One-to-one correspondence

(ii) transposed landmarks (M_2) — Between two valid landmark nodes there are two landmarks which are transposed with one another (fig. 5.3). This can easily occur due to the natural side-to-side meanderings of the robot. If two landmarks are observed very close to each other on opposite sides of the corridor, then, depending on the particular position and motion of the robot, either landmark can be observed first. A penalty is imposed, based on the distance separating the supposedly transposed nodes. Clearly the further the two landmark nodes are apart, the less likely they could be confused with each other. Also, for the two nodes to have been transposed it seems obvious that the two landmarks should have been observed on opposite sides of the robot. An additional penalty is imposed if this is not the case.

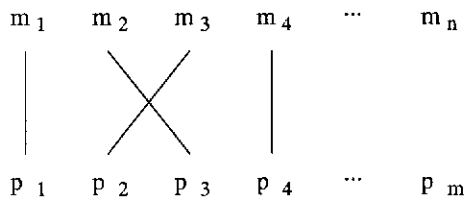


Figure 5.3: Transposed landmarks

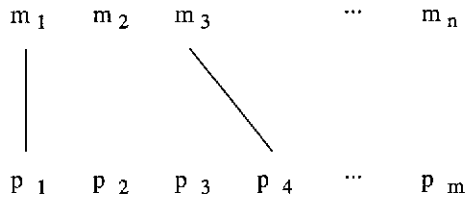


Figure 5.4: Missing and/or additional landmarks

(iii) combination of missing and additional landmarks (M_3) — Between two valid landmark nodes there are assumed to be k missing landmarks as well as j additional landmarks (fig. 5.4); $0 \leq k \leq C$ and $0 \leq j \leq (C - k)$; where C is the maximum allowable number of extraneous nodes between two valid ones. Figure 5.4, for example, shows the situation where there is one missing node (m_2) and two additional nodes (p_2, p_3), between two pairs of matched nodes ((m_1, m_3) and (p_1, p_4)). An explicit penalty is applied to the degree of match depending on the number and importance of any missing/additional landmarks. The importance of a missing landmark is determined by its 'reliability' count $m_i count$:

$$missing\_node\_penalty(m_1..m_n) = 1 - \sum_{i=1}^{n} \frac{m_i count}{tcount}$$

$$tcount = \sum^{i} m_i count$$

$$additional\_node\_penalty(p_1..p_n) = x^n$$

where $tcount$ is the sum of all landmark-node counters in the map and $x = 0.8$ (in this case).

The maximum degree of match from (i), (ii) and (iii) is hence returned as the best solution. The algorithm *best_match* is shown below.

best_match($m_1..m_n, p_1..p_m$)

/* one-to-one correspondence */

$M_1 = node\_pair\_prob((m_1, m_2), (p_1, p_2))+$
$best\_match(m_2..m_n, p_2..p_m)$
/* transposed nodes */
$M_2 = node\_pair\_prob((m_1, m_2), (p_1, p_3))+$
$node\_pair\_prob((m_2, m_3), (p_3, p_2))*$
$transposed\_prob(m_2, m_3)+$
$node\_pair\_prob((m_3, m_4), (p_2, p_4))+$
$best\_match(m_4..m_n, p_4..p_m)$
/* missing/additional nodes */
$M_3 = max\{$
$node\_pair\_prob((m_1, m_{(2+k)}), (p_1, p_{(2+j)}))*$
$missing\_node\_penalty(m_{(l+1)}..m_{(k+1)})*$
$additional\_node\_penalty(p_{(l+1)}..p_{(j+1)})+$
$best\_match(m_{(2+k)}..m_n, p_{(2+j)}..p_m)$;
where $0 \leq k \leq C$,
$0 \leq j \leq (C - k)\}$
$best\_match = max\{M_1, M_2, M_3\}$

transposed_prob(m_1, m_2)
$D = (x, y)$ positional displacement between
m_1 and m_2
$P =$ probability of match based on D,
assuming a gaussian error distribution
$(\mu = 0, \sigma = 40cm)$
if $(m_1 side = m_2 side)$ then
$P = P * 0.1$
$transposed\_prob = P$

Given the depth first nature of the matching procedure, the search tree can be pruned using the maximum valid match attained so far. The search is therefore discontinued down a branch if the resulting search is guaranteed not to better the current best match. To prune the tree further, a restriction on the minimum possible match probability, between a pair of nodes, within a valid solution is also imposed (0.2 in this case).

Although the matching process is most easily described and understood in recursive terms, it is not necessary to implement it as such. To minimise computational overhead, the matching process can be performed in an

incremental fashion, as landmarks are encountered. In this way the computation can be distributed over time. This can be a significant advantage as it reduces the otherwise spasmodic computation which may draw processing power away from more critical systems and tasks, at inopportune times. Within a real-time autonomous system the need to pause to think may have serious consequences. Having said this, most intelligent real-world tasks have an inherent asynchronicity due to the spasmodic occurrence of events or encounters. In a real-world navigation scenario, for example, asynchronism could be caused by the emergence of, not necessarily static, obstacles. However, the ideal of distributed computation, where possible, is still a valid one, by reducing the maximum amplitude of the asynchronous computation.

In the present context an incremental solution of the map matching task is proposed. As each landmark is encountered a further part of the overall solution is computed. Although the need for an incremental implementation was not apparent for the experiments so far conducted, the need may well arise upon scaling.[2] By utilising a divide-and-conquer strategy the matching task is inherently amenable to an incremental solution. The degree of match between two maps is defined by the degrees of match of its constituent nodes. By only augmenting the search tree, as opposed to recomputing it, as each new landmark is observed, the inherent computational cost of the recursive search is distributed over the time it takes to navigate the corridor segment defined by the map.

5.5.2 Map Augmentation and Refinement

If a successful match has been found the matched map is augmented with any additional information observed in the currently traversed map. Currently a successful match is basically defined by the 'average' match of the solution. If the average match is greater than some threshold (0.5 in this case) then the solution is considered to be a valid match. However, if an incremental matching algorithm is used here, one must also take into account the number of landmark

[2]The experiments shown in §5.6 involved the matching of maps made up of approximately one dozen landmark nodes each. Despite a recursive implementation, the matching time on an Indy workstation was invariably still below one tenth of a second per map. Due to the low number of maps required to be matched in the experiments, the total computational overhead, of only a few tenths of a second, was not considered serious. However, when these experiments are scaled up by either increasing the sizes of the maps (i.e. numbers of nodes) or the number of maps, the need for incremental matching should become self-evident.

nodes observed. Clearly, the fewer landmarks actually observed, the less likely the solution (currently attained) is reliably valid.

Specifically, the average match for a solution is calculated as:

$$average\_match = \frac{best\_match}{solution\_size - 1 + penalty}$$

$$penalty = \sum_{i \in M} \frac{m_i count * \#missing}{tcount} + \sum_{i=1}^{\#added} 0.2$$

where *solution_size* is the number of pairs of matched nodes in the solution; M is the set of map nodes not mentioned in the solution; $\#missing$ is the size of M; $\#added$ is the number of pattern nodes which are missing from within the solution; *tcount* is again the sum of all landmark-node reliability counts within the learned map. The rationale for this formula is to express a quality of 'map match' that is largely 'independent' of actual map size. An 'average' match is therefore devised, using the average match between landmark pairs, whilst also including a penalty component for any extraneous landmarks.

Given a valid solution, the best matched of the previously 'learned' maps can now be augmented and improved. If additional landmarks are contained in the pattern, they are added to the matched map. On the other hand, if there are missing landmarks in the pattern then the corresponding landmarks in the matched map are augmented to indicate the lesser certainty pertaining to these. This is accomplished through maintaining a simple 'reliability' counter with each landmark node. Each missing landmark node has its corresponding counter decremented by one. However, if the count associated with a landmark is decremented to zero it is removed from the map. Each successful matching of landmarks also increases this count by two. In this way landmarks which are detected only one third or one half of the time may still provide useful guidance information.

Successfully matched landmarks are also improved, in the sense that their position and relative landmark-direction are refined in accordance with the new data. The position of the landmark node and the relative direction of the landmark are improved using a weighted average between the current pattern of landmark nodes and the matched map. The landmark count is used as the weight, such that, as the landmark counters are incremented over time, less of an improvement is made to the map landmark nodes. In this way the learned map

should tend toward a more 'accurate' representation over time, such as where the side to side meandering nature of the robot is cancelled out.

5.6 Results

The set of results shown in figure 5.5 shows the process by which a specific section of corridor, S_1, is 'learned'. Figures 5.5(a,c,e,g,i,k) show the initial raw landmark maps which are constructed from the robot's dead-reckoning as it moves along the corridor, and concurrently matched to learned ones. Every alternate figure (figs. 5.5b,d,f,h,j,l) shows the evolution of the matched (learned) map. Each successive map is the updated and improved version of the previous, given the successfully matched current map in each case.

Figure 5.5(a) shows the landmarks observed on the initial run. Since no maps have been learned at this stage, a new map (fig. 5.5b) is created from the initial pattern. Figure 5.5(c) shows the second run along the identical portion of corridor with the same landmark layout. This is successfully matched ($average\_match = 0.622$) with the current learned map (5.5(b)). The learned map is then improved (fig. 5.5d) given the fresh information contained within 5.5(c). The third through to the sixth run also matched successfully (see table 5.1 and fig. 5.6) with their current learned map. At each stage the learned map is refined. The fifth refinement of the learned map (fig. 5.5l) can be seen to be marginally straighter and more centred than the first few. The improvement is quite slight due to the natural tendency of the robot, in that specific environment, to meander to the right at the beginning of the run.

S_1 Run	Average Match with latest refined map from 5.5
5.5(a)	
5.5(c)	0.622
5.5(e)	0.693
5.5(g)	0.808
5.5(i)	0.510
5.5(k)	0.720

Table 5.1: Match data for S_1 (fig. 5.5)

Having successfully learned one piece of corridor the robot is put into a new corridor to see how it performs. In fact, the new corridor, S_2, has simply a slightly

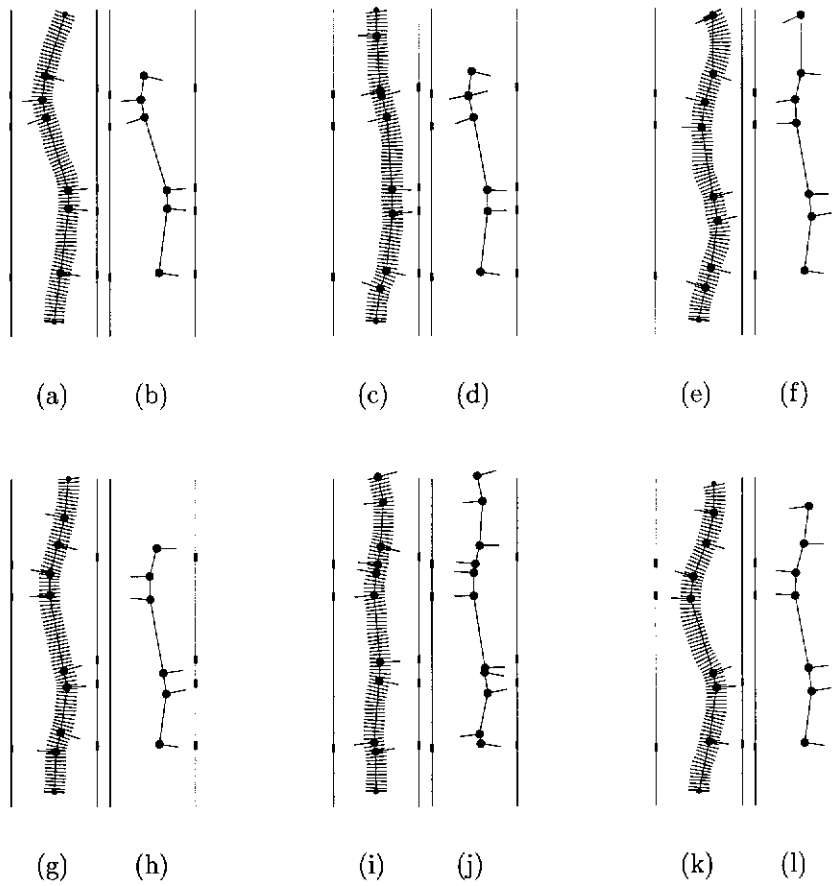


Figure 5.5: Learning a landmark map through repeated traversals of corridor S_1.

different arrangement of landmarks. Figure 5.7 shows the maps built during this phase. Again, figures 5.7(a,c,e,g,i,k) show the patterns of landmarks as observed by the robot on successive runs, whereas the remaining alternate figures show the evolution of the learned map.

The first run through the new corridor is shown in figure 5.7(a). The new pattern of landmark nodes is again matched against all previously learned maps. The degree of match for the previous learned map (fig. 5.7l) was 0.415 and hence failed. A new map is therefore created from the new pattern, shown in figure 5.7(b). The rest of the runs were again successfully matched with the refined versions of 5.7(b) (see table 5.2 and fig. 5.8). In each case the previous learned map 5.5(l) failed to match, as shown in table 5.2.

Having shown that the robot can correctly match and discriminate between a couple of straight sections of corridor, it should be noted that in some sense this can be considered a worst case scenario. A straight corridor is difficult to

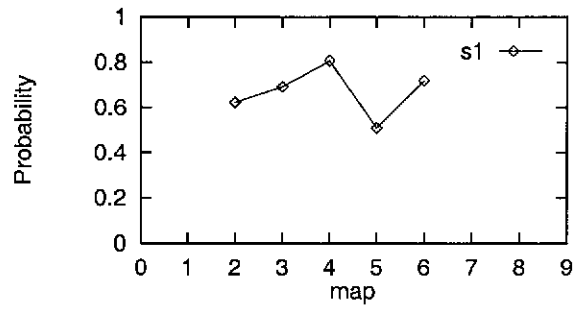


Figure 5.6: Match data for S_1 (from table 5.1)

S_2 Run	Average Match with map 5.5(l)	Average Match with latest refined map from 5.7
5.7(a)	0.416	
5.7(c)	0.451	0.654
5.7(e)	0.281	0.864
5.7(g)	0.462	0.736
5.7(i)	0.479	0.606
5.7(k)	0.354	0.697

Table 5.2: Match data for S_2 (fig. 5.7)

discriminate from another straight section, as there is very little difference in the shape of the corridor. Hence the pattern of dead-reckoned landmark nodes becomes increasingly important. The discrimination essentially comes down to the 'uniqueness' of the pattern of landmarks. This can be quite a limiting factor on the number of corridors which can be successfully discriminated, given a maximum map size. As the number of corridor sections, to be successfully discriminated between, increases, so does the required size, or complexity, of the maps.

If, however, a straight corridor were to be matched against a curved one, or even simply one with turn or kink in it, the shape differences are clearly going to increase the chances of discrimination.

Having successfully learned two straight sections of corridor the performance within a curved section of corridor, C_1, is now examined. Once again, figures 5.9(a,c,e,g,i,k) show the sequence of dead-reckoned landmark maps generated en route. The alternate maps in figures 5.9 again show the evolution of the learned map.

The initial run through the new corridor (fig. 5.9a) failed to match either of the two previously learned maps and hence a new map is generated. As shown in

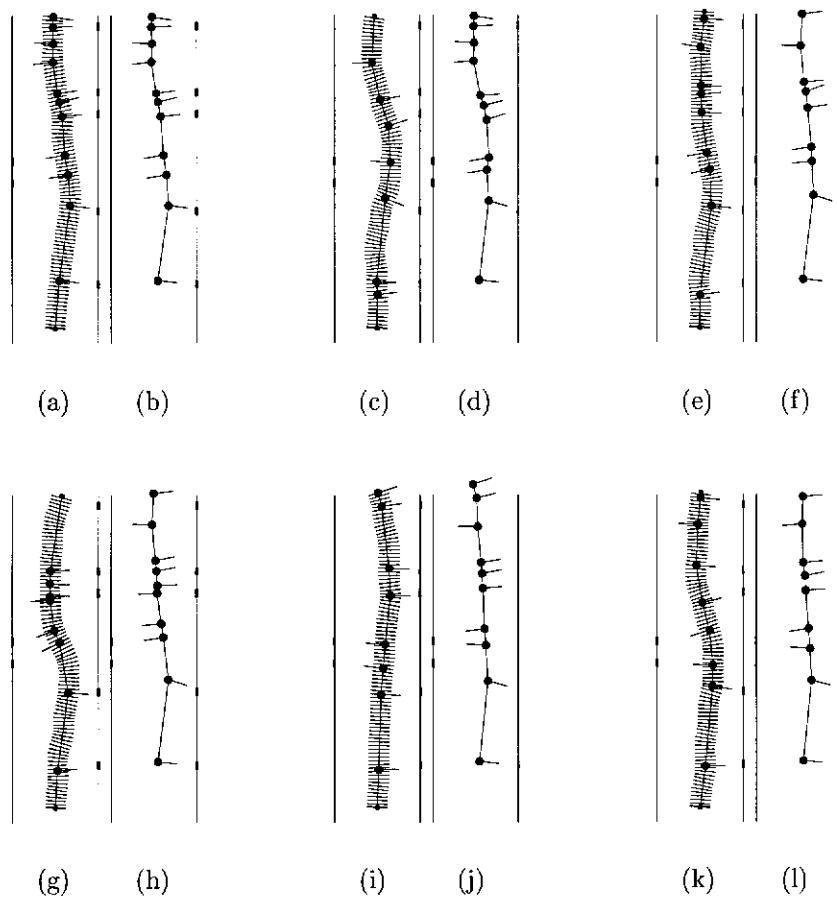


Figure 5.7: Learning a second landmark map through repeated traversals of corridor S_2.

table 5.3, each successive run was successfully matched with the current learned map and failed to match the two previously learned ones. As expected, it can also be seen (table 5.3 and fig. 5.10) that the degree to which the curved corridor is distinguished from the straight sections is significantly greater than the degree with which the straight sections were distinguished from one another (fig. 5.8).

It should be noted that, although the corridor segments are 'learned', in the sense of reinforcement learning, there is no 'training' period in the traditional sense. The results were grouped into batches of six runs purely for ease of description.

As has been shown the robot can now successfully distinguish between three different corridors, S_1, S_2 and C_1, via the evolved maps depicted in figures 5.5(l), 5.7(l), and 5.9(l), respectively. Given this state, the robot is again presented with a succession of six runs through corridor S_1. This is depicted in figure 5.11. As

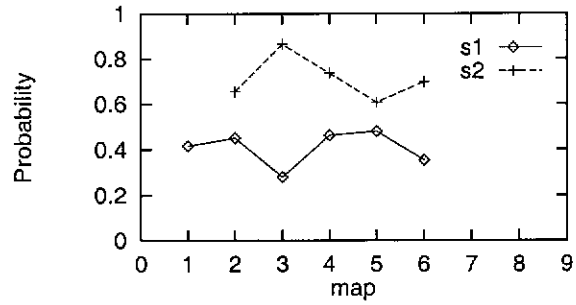


Figure 5.8: Match data for S_2 (from table 5.2)

Run	Average Match with map 5.5(l)	Average Match with map 5.7(l)	Average Match with latest refined map from 5.9
5.9(a)	0.174	0.117	
5.9(c)	0.205	0.163	0.661
5.9(e)	0.188	0.330	0.568
5.9(g)	0.282	0.178	0.746
5.9(i)	0.180	0.208	0.523
5.9(k)	0.268	0.349	0.790

Table 5.3: Match data for C_1 (fig. 5.9)

shown in table 5.4 and figure 5.12, the robot successfully recognises the corridor as being S_1.

The robot is now presented with six more traversals through corridor S_2. This is shown in figure 5.13. However, as table 5.5 and figure 5.14 show, run 5.13(i) fails to successfully match any previously memorised maps. A new map is consequently created (fig. 5.13(j)). The next run, 5.13(k), now poses a small problem. It matches successfully with two memorised maps, 5.13(h) and 5.13(j), with probabilities 0.765 and 0.704, respectively. Which one should be chosen to be correct? Intuitively, the one providing the best match is chosen (5.13(h)). However, a further decision is also made concerning the validity of the learned map 5.13(j). Due to its high degree of match with another memorised map and its relatively low 'maturity', it is expunged. Map maturity can be defined by the average 'reliability' of the nodes that make it up. In this case map 5.13(j) has a maturity of only 2.0, whereas map 5.13(h) has a maturity of 10.2.

Finally, the robot is presented with a further succession of runs through corridor C_1, shown in figure 5.15. These are once again easily recognised as being C_1, but also highly distinguished from both S_1 and S_2 (table 5.6 and figure 5.16).

From figures 5.6, 5.8, 5.10, 5.12, 5.14, and 5.16 it can also be seen that degree

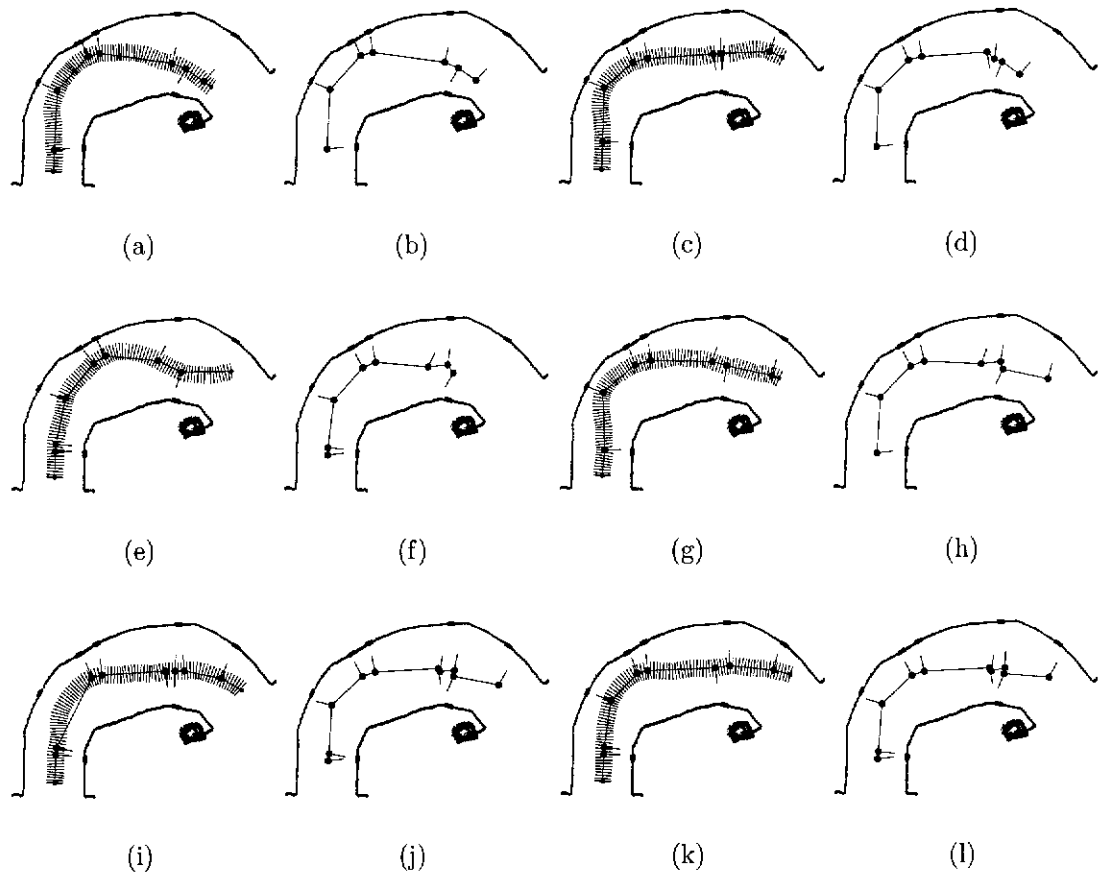


Figure 5.9: Learning a third landmark map through repeated traversals of corridor C_1.

with which the second batch of runs (figs. 5.12,5.14,5.16) is distinguished is much more consistently than the first. The degree of discriminatory power, provided by the more established maps, seems also to have, in the main, marginally improved.

5.7 Conclusions

The robot's purely reactive behaviour has been augmented to incorporate an, on the fly, map building regime for the express purpose of recognising previously navigated corridors. As was shown, the robot was able to discriminate between similar environments by virtue of distinguishing features observed en route. These features being the shape of the navigated path, which is confined by the corridor environment, and the relative positions of simple landmarks observed en route.

Although, in the present implementation, the dead-reckoning data was

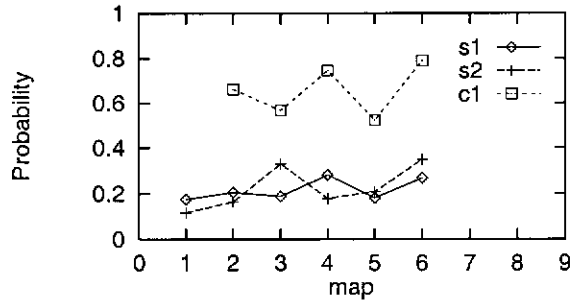


Figure 5.10: Match data for C_1 (from table 5.3)

Run	Match with map 5.11(l)	Match with map 5.13(l)	Match with latest map 5.11
5.11(a)	0.755	0.349	0.150
5.11(c)	0.615	0.421	0.152
5.11(e)	0.685	0.417	0.206
5.11(g)	0.718	0.411	0.211
5.11(i)	0.739	0.398	0.162
5.11(k)	0.859	0.425	0.197

Table 5.4: Continued match data for S_1 (fig. 5.11)

provided by the robot's wheel-based odometry, this need not be the only source. If wheel-based odometry is unavailable or unreliable then 'visual odometry' may be a viable alternative. As was shown in the previous chapter, the integration of the optic flow (e.g. lateral apparent motion) observed en route, can provide an alternate measure of distance travelled. At the very least, visual odometry can provide an additional source of discriminatory information for matching purposes.

Although the map building scheme outlined is not an optimal solution, given the computation restrictions of working in real-time and the simple entomological nature of the mobile robot (e.g. its meandering path), the performance was quite successful. If computational overhead was not a consideration, a brute force matching regime, examining complete patterns and maps as a whole, could be imposed. This would produce superior results but the costs would be prohibitive, especially in a real-time application.

Obviously, there are several improvements and extensions which can and should be made, to provide a specific mobile robot with the ability to perform the navigational function needed of it. However the basic premise, as shown, is quite sound.

One such extension is a mechanism by which learned maps are linked

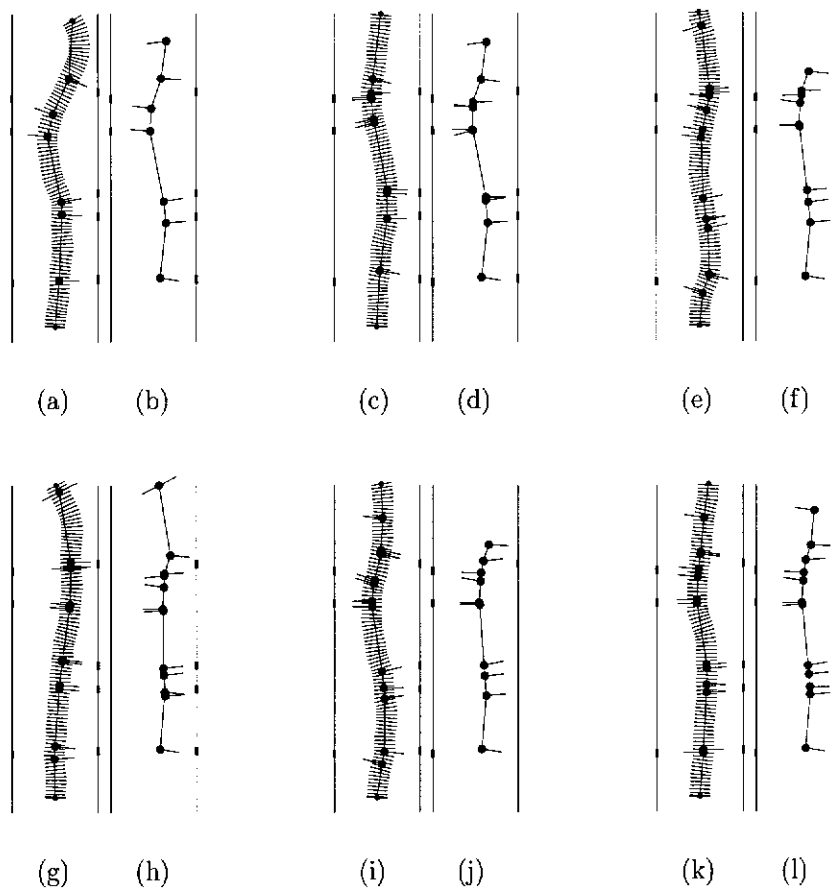


Figure 5.11: Continued Learning of landmark map S_1

together. Self-location is only one piece of the autonomous navigation puzzle. Adding further levels of abstraction can aid and improve 'global' knowledge and navigation.

Certain other situation-specific parameters also need to be nailed down. These include the maximum and minimum size of a map, when to end a pattern and start a new one, and so on. One must also consider how a reliability judgement can be made. How reliable is the current information and to what extent does this affect decision making? For example, when comparing current en route experience to memorised knowledge when does recognition become reliable. Reliability here can be gauged by the 'uniqueness' of the pattern. Although a very relative concept, the uniqueness or distinguishing power of a corridor map can be loosely defined by its size and complexity. These details, however, would usually strongly depend on specific environmental characteristics, and so, by effect, on maps already learned.

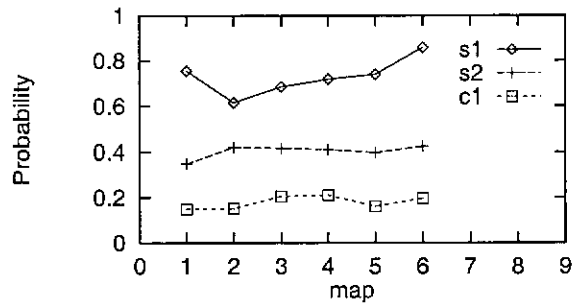


Figure 5.12: Continued match data for S_1 (from table 5.4)

Run	Match with map 5.11(l)	Match with map 5.13(l)	Match with latest map 5.13
5.13(a)	0.402	0.726	0.076
5.13(c)	0.476	0.797	0.209
5.13(e)	0.543	0.700	0.171
5.13(g)	0.411	0.647	0.112
5.13(i)	0.467	0.083	0.105
5.13(k)	0.432	0.765	0.159

Table 5.5: Continued match data for S_2 (fig. 5.13)

Run	Match with map 5.11(l)	Match with map 5.13(l)	Match with latest map 5.15
5.15(a)	0.059	0.060	0.721
5.15(c)	0.288	0.170	0.718
5.15(e)	0.154	0.215	0.804
5.15(g)	0.254	0.174	0.755
5.15(i)	0.250	0.126	0.771
5.15(k)	0.280	0.153	0.842

Table 5.6: Continued match data for C_1 (fig. 5.15)

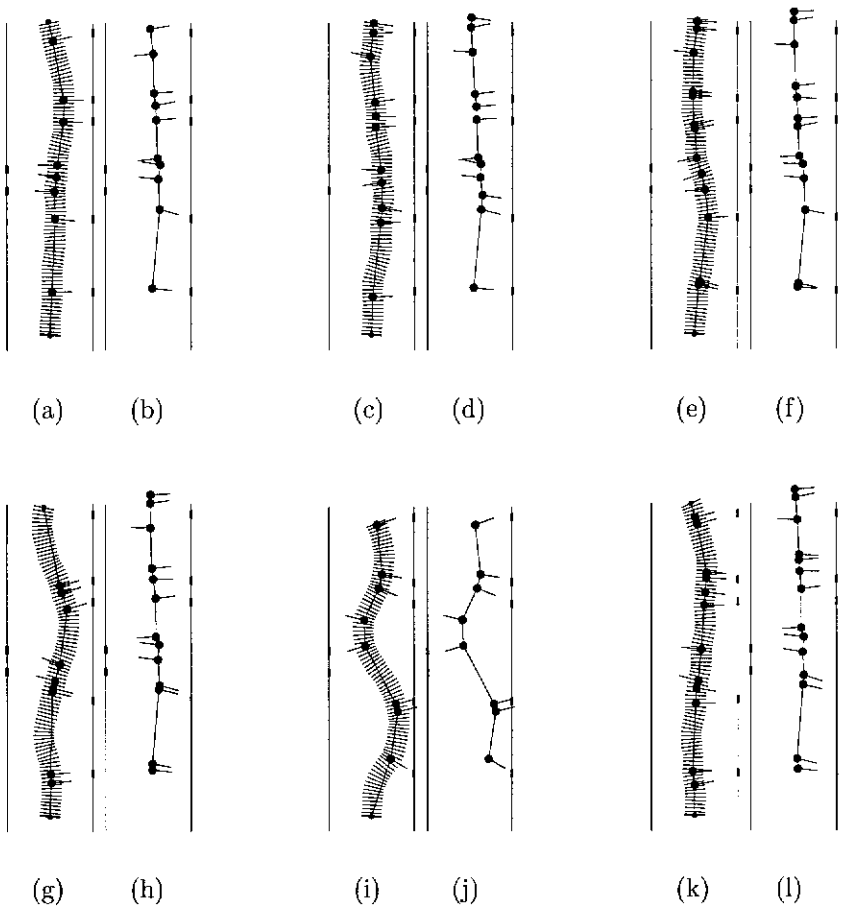


Figure 5.13: Continued learning landmark map S_2

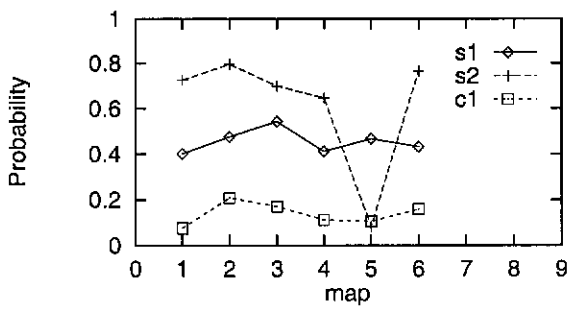


Figure 5.14: Continued match data for S_2 (from table 5.5)

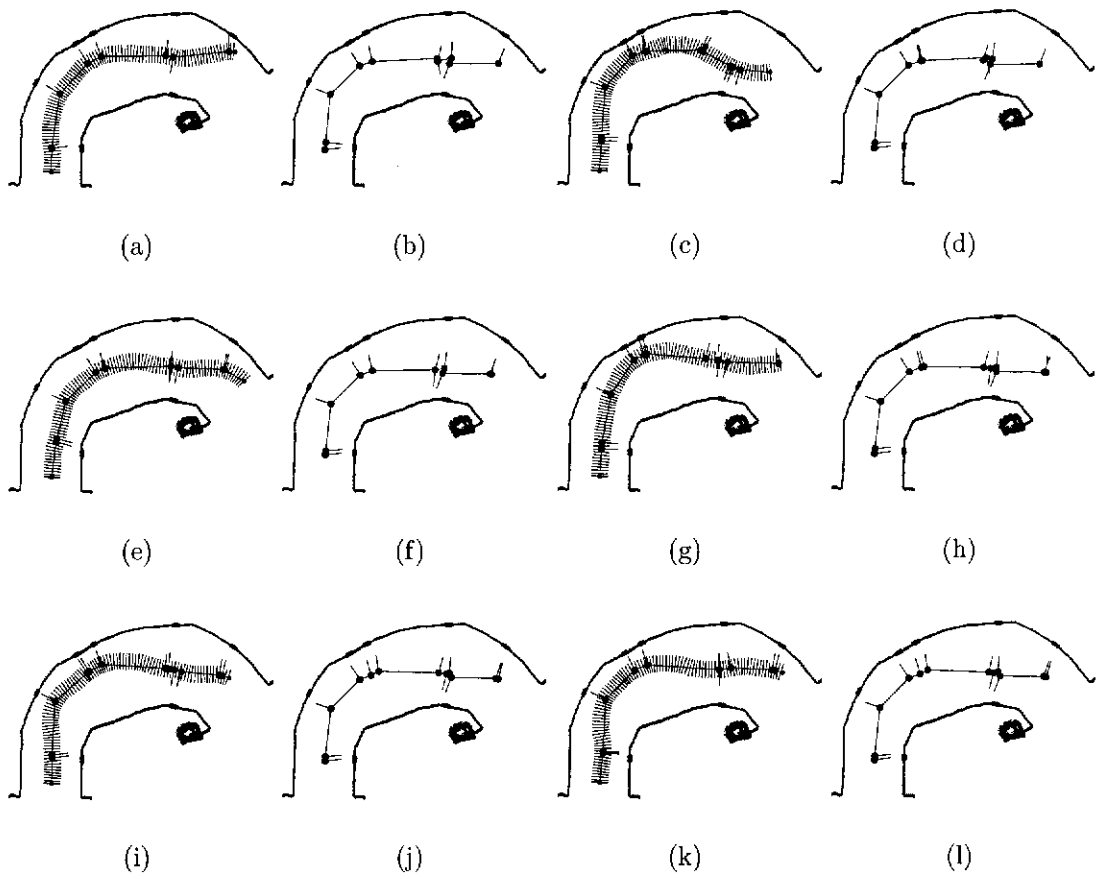


Figure 5.15: Continued learning landmark map C_1

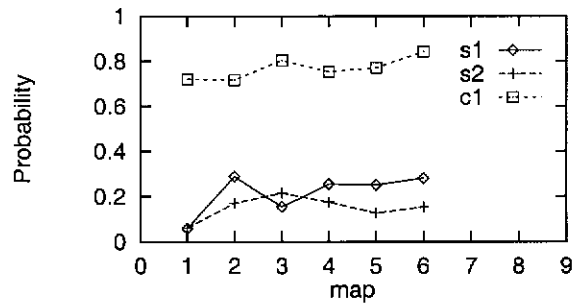


Figure 5.16: Continued match data for C_1 (from table 5.6)

Chapter 6

Homing

Many animals, including insects, successfully engage in visual homing. This chapter describes a system that allows a mobile robot to home. Specifically, a simple, yet robust, homing scheme that only relies upon the observation of the bearings of visible landmarks, is proposed. However, this can easily be extended to include other visual cues which may improve overall performance.

The homing algorithm allows a mobile robot to home incrementally by moving in such a way as to gradually reduce the discrepancy between the current view and the view obtained from the home position. Both simulation and mobile robot experiments are used to demonstrate the feasibility of the approach.

6.1 Introduction

Many animals are very adept at finding their way back home after foraging for food (Papi, 1992). Insects, particularly hymenopterans such as bees and certain species of ants, rely heavily on visual cues for homing. Such creatures are particularly attractive subjects to emulate in the present context. The Saharan desert ant, for example, navigates flawlessly back to its nest after going out on foraging bouts that take it several thousand body lengths away from its nest (Wehner, 1992). Similar principles could conceivably be applied to the design of navigational strategies for a planetary rover that may be deployed to explore unknown territory over the course of an entire day, and to return to its 'home base' at nightfall to recharge batteries or transfer data.

This chapter presents a simple, robust strategy for robotic homing, inspired by the visual homing behaviour of bees and ants.

6.2 Background

6.2.1 Insect Homing

A large number of experiments (see §2.5.9) have shown that many insects are able to 'home in' on a specific location, such as a nest, by using visual cues provided by landmarks in the vicinity. This ability is developed most highly in central place foragers, such as bees and ants.

Ethological experiments (see §2.5.9) seem to suggest that insects search for a specific location by using a form of image matching. The insect behaves as though it is striving to 'home in' by moving in such a way as to maximise the match between the current retinal image and a 'snapshot' of the panorama as seen from the goal, acquired on an earlier visit. In this way the insect is continually and locally guided by the desire to reduce this discrepancy between current and home snapshots until it becomes zero. Moreover, it seems that insects 'process' the snapshot images for features such as edges. It is also suggested that the insects make use of an external frame of reference, a celestial compass, to keep track of snapshot orientations. Homing, by familiar landmarks, becomes much simpler and more certain when the snapshots have a common orientational frame of reference.

6.2.2 Robot Homing

There have been several previous implementations of vision-based robotic homing (see §2.7). However, there are two main approaches to vision-based homing using the panoramic information as seen from home. Firstly, one can try to derive a correctional vector from discrepancies seen in the raw images captured from the differing views of home and current location. This will be referred to as image-based homing. Secondly, one can opt to try to detect salient features, such as landmarks, in the views and then derive a solution from the discrepancies herein. This will be referred to as landmark-based homing.

The homing approach most similar to the one presented here is that of Hong et al. (1990; 1991; 1992). They have developed a system which uses a landmark-based local homing algorithm[1] to allow a mobile robot to move from one location to another by successively homing in on a sequence of target locations

[1]See §2.7.2 for details.

situated en route. In their experiment a mobile robot traverses a corridor by homing on successive target locations evenly spaced, approximately one foot apart, along the corridor.

6.2.2.1 An Improved Approach

By matching the bearings of features extracted from panoramic views and using a vector summation technique to extract a homing vector, a simple, parsimonious, and yet quite robust robotic homing algorithm can be constructed. Specifically, however, the proposed approach:

(i) is very simple. Vector summation is an intuitive, simple, and computationally cheap method of generating a homing vector.

(ii) can function well using only a parsimonious representation of the environment. The homing algorithm requires only landmark bearing information, and hence a snapshot need essentially consist of only a list of bearings. Homogeneity of landmarks is therefore of no concern. The homing algorithm may, however, easily be extended through the incorporation of additional cues, to further assist in the visual homing task. Landmark apparent-size cues, for example, allow improved homing as well as improved landmark avoidance behaviour.

(iii) is very powerful in terms of catchment area. In these experiments it is shown that the size of the catchment area, for the homing algorithm, is only limited by the detectability of landmarks. In general, given that the same set of landmarks are detectable, the homing algorithm will provide a successful path back home irrespective of how different the snapshots are. This is not the case with previous vision-based homing approaches.

(iv) does not require any complex correspondence techniques. It is shown that a very simple, and approximate, correspondence between landmarks is all that is required for successful homing. Further, it is shown that an $O(n^2)$ landmark correspondence method provides an approximation which performs equally well to that of perfect correspondence.

(v) has an inherent tendency to avoid collisions with landmarks. This very useful feature emerges from the vector summing technique used in

calculating the homing vector.[2] It is also shown that one of the refinements that Hong et al. (1991) make to their homing vector computation actually interferes with this feature.

The main differences between the Hong et al. (1990; 1991; 1992) approach and that presented here are subtle but significant. It is shown that the landmark correspondence technique can be approximate, computationally parsimonious and yet still sufficient for homing. Despite this approximation, however, the homing algorithm still produces a superior catchment area to that of previous approaches. Also shown is the quite dramatic effect that a seemingly small and intuitive 'improvement' on homing vector computation can have on the global homing behaviour. In this case the landmark avoidance behaviour of the homing algorithm is corrupted by such an improvement. Finally, the homing algorithm can easily be extended by incorporating a richer source of homing cues. This can significantly improve homing performance. As will be shown, the additional use of landmark angular size cues not only improve the robustness of the homing approach but also provide a method of reinforcing landmark avoidance.

6.3 Robot Setup

The mobile robot[3] used for the real-world visual homing experiments is depicted in figures 6.1(a,c). Movement is provided by the two separately controllable drive wheels located at the front of the robot. The views, two frontal and two lateral, are provided by a single camera placed looking upwards at a mirror assembly (fig. 6.1(a)). An example image showing the mapping of the two lateral and two frontal views onto the single camera image, is shown in figure 6.1(b).

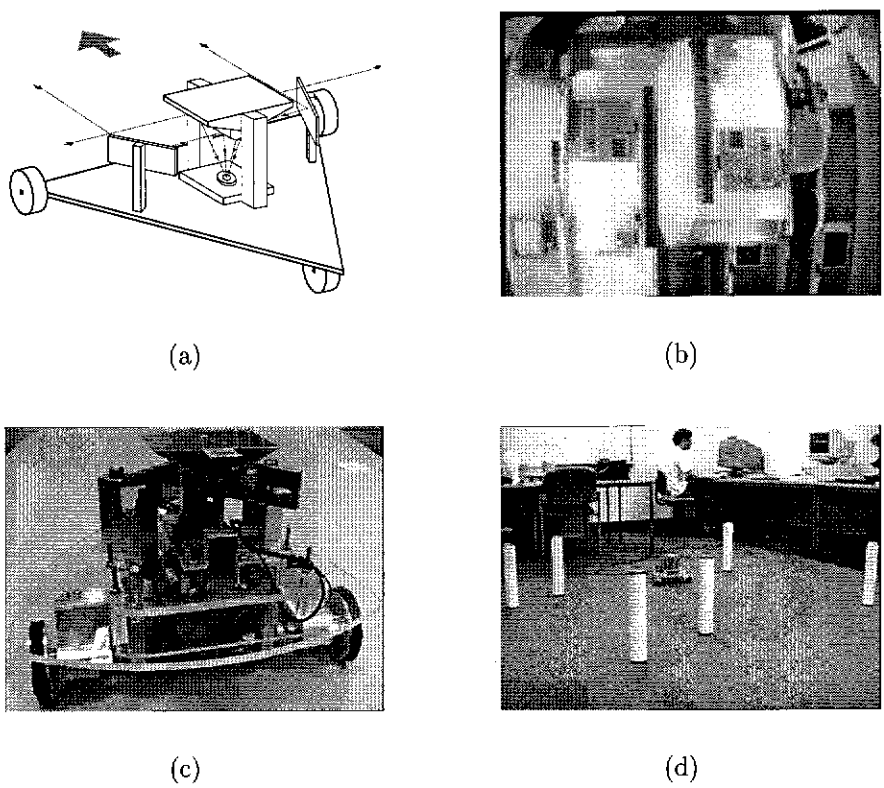
The test environment is constructed with the use of artificial,[4] cylindrical, matt white landmarks (fig. 4.3(d)). The use of artificial landmarks allowed for the explicit placement of landmarks at various distances. Thus providing a very flexible homing environment in terms of specific arrangements of landmarks.

[2]To my knowledge, none of the previous studies have considered landmark avoidance explicitly.

[3]Described in detail in chapter 3.

[4]Typically strongly dependent on environment specifics, general real-world landmark recognition is a difficult problem in and of itself and hence was not considered an essential element to be pursued here. The key issue pursued in this chapter is not how landmarks are identified or recognised, but given this ability, how these visual cues can be used to direct intelligent and useful behaviour, in this case, visual homing.

The landmarks are considered homogeneous for both greater generality and simplicity. Although identifiable landmarks allow more reliable landmark recognition and correspondence they can also introduce another source of error as the recognition process becomes less certain under less than ideal, real-world circumstances. As will be shown, there is no inherent need for heterogeneous landmarks. An arrangement of homogeneous landmarks, or simply landmarks treated as being homogeneous, is all that is required for visual homing.



(a) (b)

(c) (d)

Figure 6.1: Robot and environment. (a) A schematic of the robot shows how the two frontal and two lateral views are captured by a single camera by way of the mirror assembly. (b) An example camera image shows how the four separate views are combined into a single image. (c) The physical robot. (d) The test environment.

As described previously, the mobile robot was custom designed specifically to cater for experimentation in applying visual-based insect behaviours. However, one major disadvantage of our robot, for the task of visual homing, is the restricted field of view offered by the camera setup. The current imaging system

was not designed with homing in mind.[5] A full 360° view, a panoramic sensor, is what is ideally required for vision-based homing. This can be achieved by setting a camera to point directly up at a conical[6] or spherical[7] mirror. In this way the entire panoramic view can be directed onto a single image.

To achieve a panoramic view for this system a behaviour-based rather than hardware-based solution, has been employed. This solution unfortunately interferes with the pure homing behaviour and consequently compromises control. In essence, the robot must physically rotate (on the spot) a full 360° every time a snapshot of the environment is to be obtained. A panoramic snapshot of the environment provides the most reliable and complete view of the salient features used for homing. The spinning behaviour, consequently forces the robot into a stop/start stepwise refinement behaviour rather than a smooth, continuous, more efficient behaviour.[8] Despite the practical restrictions, however, the resulting behaviour has been quite successful. As will be shown, the real robot not only exhibits successful homing behaviour but also confirms the general homing behaviour observed in simulation.

6.4 Robotic Homing

6.4.1 Image Processing

As previously described, due to the absence of a panoramic sensor, the robot must spin on the spot each time it wishes to grab a snapshot of the environment. A panoramic view of the environment is created by piecing together the sequence of images observed through one of the frontal views, as the robot spins. This is done on the fly, by correlating each new image, seen as the robot rotates, with the current panoramic image being created. Odometric information is used

[5]See chapter 4.

[6]See (Yagi et al., 1995; Chahl and Srinivasan, 1996; Franz et al., 1997a).

[7]See (Hong et al., 1991).

[8]Due to the constant spinning behaviour, global orientation had to be manually refreshed periodically, as the wheel-based odometry was not accurate enough to maintain a common frame of reference for the visual snapshots. Ideally, a magnetic compass sensor would be used for this purpose, although, the use of an optical gyroscope is also quite attractive since the objective is to merely provide a common frame of reference and not specifically maintain an accurate account of magnetic north. Optical gyroscopes are becoming very attractive for mobile robotics. With little or no moving parts, such rotation sensors are virtually maintenance free and display no gravitational sensitivities, eliminating the need for gimbaled mounting (Everett, 1995).

to guide this process by providing an initial registration estimate for each new image. Odometric information also assists in the identification of the wrap-around point. In essence, the degree of physically sensed rotation is used to assist in the calculation of visually sensed rotation by providing initial estimates. Image correlation, however, is used to fine tune the piece-wise assembly. Each image is smoothed by a gaussian filter before being correlated with and added to the evolving panoramic image.

To detect landmarks in the panoramic image, a moving window is used to identify contiguous regions of significantly light and dark luminance. For current purposes, landmarks are simply defined as very light regions in the image. The bearings of these landmarks (i.e. centre of gravity) are then noted. This is the only information needed by the homing algorithm. A snapshot can therefore be represented by a simple list of landmark bearings:[9]

$$S = [\theta_1, \theta_2, \theta_3, ..., \theta_n] \qquad \text{where} \quad \theta_i \dot{<} \theta_{i+1}$$

6.4.2 Homing Algorithm

The homing algorithm primarily computes a homing direction, given the snapshot information as taken from home, and the snapshot information taken from the present location. However, the homing algorithm, in this case, is required to give a target distance as well as direction. This homing vector essentially directs the robot to the next snapshot position.

6.4.2.1 Landmark Correspondence

The first task is to match up corresponding landmarks (bearings) from each snapshot as best as possible. Each landmark in one snapshot is paired with exactly one landmark from the other snapshot. In the current case this is straight forward due to the fact that all landmarks are considered to be homogeneous.

To achieve an 'optimum' pairing of landmarks, and hopefully a correct one, a search is conducted for the matching that gives the least mean square error of the pairing error. The pairing error is defined as the angular difference between

[9]Note that the dot notation in this thesis is used to signify the operators that are used to do arithmetic with bearings. These operators, such as $\dot{<}$ and $\dot{-}$, appropriately handle the wrap-around effects observed with bearings.

paired landmarks. This search can be performed using a standard depth first search strategy. Essentially, the search strives to find the minimum of:

$$\sum_{i=1}^{n} |\theta_i - \beta_{p(i)}|^2 \tag{6.1}$$

where θ_i and $\beta_{p(i)}$ represent the bearings of paired landmarks and n the number of said pairs. The pairing function $p(i)$ serves to provide the mapping between landmarks observed from the home snapshot and those currently observed.

Although this snapshot similarity measure is not guaranteed to provide a 100% correct correspondence between landmarks, it is sufficient. An approximation is all that is required for the homing algorithm to work.

The only problem with exhaustively searching for an 'optimal' pairing of landmarks, despite tree pruning, is that when the number of landmarks becomes large the computational requirements may easily become excessive. Exhaustive search has a complexity of the order $O(n!)$.

There are several solutions to this problem. One can, for example, throw away some of the landmarks, perhaps the less distinctive or less prominent ones. However, solutions involving disregarding possibly valuable information are unattractive for obvious reasons. The visual appearance of landmarks may well change significantly from different perspectives, for example. Although eliminating erroneous landmarks, such as those caused by unreliable landmark detection, is a worthwhile objective, this is also fraught with danger if this mechanism is also unreliable. Usually a better, more robust, option is to be able to cope with erroneous data in an implicit fashion.

To overcome this computational complexity problem several alternative matching methods have been investigated. Experiments have shown that the homing algorithm is very tolerant of less than optimal and blatantly wrong matchings. This is due to the fact that, in general, the further away from home, the greater the discrepancy between snapshots and the greater the chances of incorrect pairing, the less need for correct matching. In other words, the increase in snapshot dissimilarity and consequent landmark correspondence error that occurs with increased homing distance is at least in part compensated for by the fact that the margin of error also increases as the distance from home increases. In this chapter it will be shown that very good homing behaviour can be achieved even with very simple matching and correspondence techniques, with far less

computational complexity than the exhaustive search procedure described above.

6.4.2.2 Alternative Landmark Correspondence Methods

As stated earlier, an exhaustive search ($O(n!)$), for the 'optimal' pairing of landmarks, can really only be safely applied when a small number of landmarks are being used.

To combat this problem several alternative landmark correspondence methods have been explored, that are simple and computationally safe. The four methods, $h2, h3, h4$, and $h5$, are at worst $O(n^2)$. Method $h6$ is $O(n^3)$. (Let us assume, for the moment, that the numbers of landmarks within the home and current snapshots are equal.) The relative success of each method will be discussed in detail in the results section. (Method $h1$ refers to the exhaustive search approach.)

The conceptually simplest and computationally cheapest is method $h2$. This method consecutively pairs each home snapshot landmark (θ_i) with the closest matching landmark, with respect to bearing, in the current snapshot (β_i). So, for example, θ_1 is paired with the closest bearing within $\{\beta_1, \beta_2, .., \beta_n\}$. Then θ_2 is paired with the closest unpaired bearing within $\{\beta_1, \beta_2, .., \beta_n\}$. And so on for all θ_i. This reduces the worst case complexity to $\Theta(\frac{n(n-1)}{2})$, and although only a rough approximation, and hence more susceptible to error, this still results in successful homing.

Method $h3$ simply involves correlating the two snapshots with each other using the same matching function as was used in the search ($h1$). The resulting correspondence is given by the least mean square error. This method differs from $h1$ by not altering the adjacency of the landmarks within a snapshot. In other words, if θ_i is paired with β_j, then θ_{i+1} must be paired with β_{j+1}. (For two sets of n landmarks there are n possible sets of pairings.)

Method $h4$ utilises a simple local sort concept. Starting from the default initial correspondence, where snapshot landmarks are consecutively paired (i.e. θ_i is paired with β_i), adjacent pairings are exchanged with each other if the result is an improvement in the sum of the square error involved. If θ_i is paired with β_i then, for example, β_1 can be exchanged with β_2 so that θ_1 is now paired with β_2 and θ_2 with β_1. If this new correspondence reduces the error, it is kept. This bubbling improvement is continued for all adjacent pairs until no further improvement is possible.

Method $h5$ is merely a concatenation of methods $h2$ and $h4$. In essence, the

initial correspondence for $h4$ is provided by the output from $h2$.

Method $h6$ is similar to $h2$, however a preferential voting scheme is used to provide a better overall match. Each landmark within a snapshot has a list of preferences of which landmarks, within the other snapshot, it wishes to be paired (i.e. the closest). This is implemented by treating the problem as a stable marriage problem (Sedgewick, 1988). In this way, a 'stable' pairing configuration is established in the sense that, there does not exist the situation where a landmark prefers another (in the other snapshot) over their existing partner and the preferred landmark 'feels' the same. If there exist two landmarks which prefer each other over their respectively assigned partners then the pairing configuration is said to be 'unstable'.

For reasons of completeness, a one-to-many landmark correspondence method ($h7$) is also examined. Method $h7$ is essentially identical to method $h2$, except that a one-to-one correspondence is not enforced. The landmark bearing from one snapshot is now paired with the closest landmark bearing from the other snapshot, irrespective of any other pairing.

6.4.2.3 Computing The Homing Direction

We have chosen a very intuitive method of computing the homing direction given only a list of paired bearings (landmarks).

Essentially what is required, is to move the robot in a direction which brings the bearings of landmarks seen at the current position closer to the bearings of the landmarks seen from home. Given only bearing information, the most obvious answer is to move perpendicularly to the current bearing of a landmark,[10] in the appropriate direction to bring it closer to the correct bearing as seen from home. For each pairing of bearings, there is a correctional vector. By summing over all the correctional vectors we arrive at our homing direction. However, these correctional vectors are also weighted by the error (difference in bearing) between the bearing pairs. In this way the homing algorithm will strive to correct the worst pairings faster than the best pairings. (The effect of using unit correctional vectors[11] instead of proportional correctional vectors is also examined in §6.5.)

Consider the situation depicted in figure 6.2. The robot wishes to home

[10] Although this perpendicular movement is the most obvious, it is certainly not the only option available, as described in detail in §6.4.2.4.

[11] In this thesis, the use of unit correctional vectors is signified by a 'u' appended to the landmark correspondence method (e.g. h7u).

from starting point S, back to its home position H, given the bearings of three landmarks A, B, and C. Firstly a hypothesised correspondence is established between the landmarks currently observed and those seen from home. For this simple case it can safely be assumed that this has been successfully accomplished; a 100% correct correspondence has been established. Now, the correctional vectors (V_A, V_B, and V_C) can be computed, such that each vector is in a perpendicular direction to the (currently) observed landmark. Moreover, they are in a direction which brings the bearings of the landmarks closer to that observed from home. For example, consider the correctional vector for landmark A (V_A). To improve the current bearing of A (A_S), i.e. make it more similar to that observed from home (A_H), one moves in the direction V_A.

The length of each correctional vector is determined by the angular difference between the landmark bearing currently observed, and the corresponding landmark bearing as observed from home. This is a linear relationship: the vector size is proportional to this error. As seen in figure 6.2, V_B is small relative to V_A and V_C due to the relatively small angular difference between B_S and B_H. The correctional vectors are now summed to produce the homing direction H_S.

$$\begin{aligned}
\vec{H} &= \sum_{i=1}^{n} \vec{V_i} \\
\vec{V_i} &= |\theta_i \dot{-} \beta_{p(i)}| \quad \angle\delta_i \\
\delta_i &= \begin{cases} \beta_i + 90° & : \quad \theta_i \dot{<} \beta_i \\ \beta_i - 90° & : \quad \theta_i \dot{\geq} \beta_i \end{cases}
\end{aligned}$$

In summary, a homing vector $\vec{H}$ is given by the vector summation of all correctional vectors $\vec{V_i}$, where the home bearings θ_i are paired with the currently observed bearings $\beta_{p(i)}$. (In the above notation a vector is expressed as a magnitude M and a bearing θ pair: $M\angle\theta$.)

The only task remaining is to compute the distance to be traversed before the next snapshot. (This distance information is needed due to the discretising effect of our robot's panoramic capturing limitations.) This distance is proportional (within set limits) to the total error between paired bearings. (This error is again calculated by summing over the angular differences observed between paired landmarks, i.e. $\sum_{i=1}^{n} |\theta_i \dot{-} \beta_{p(i)}|$ where $p(i)$ is the pairing function.) The smaller the error the less distance the robot should cover before taking another snapshot.

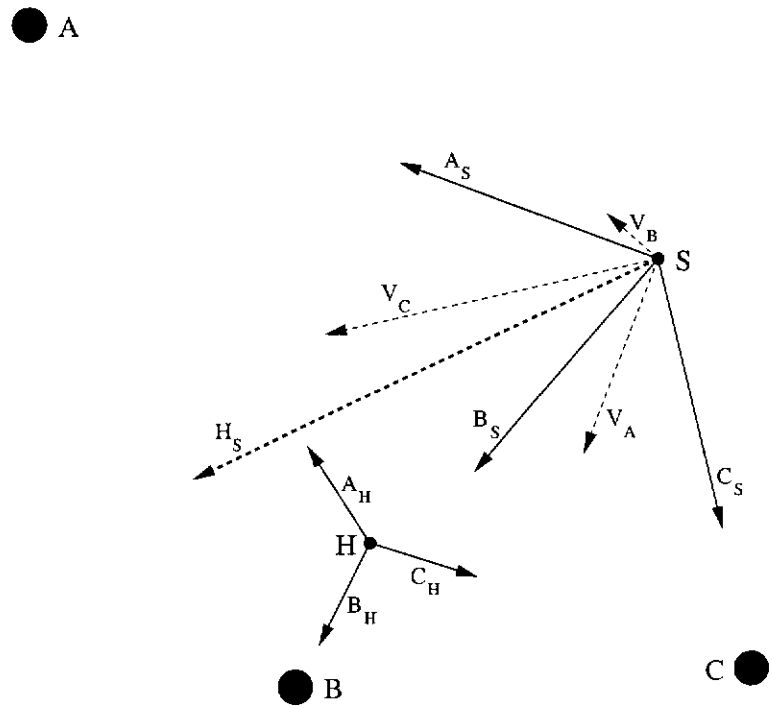


Figure 6.2: Computing the Homing Direction. Each landmark i produces a local correctional vector V_i, the summation of which determines the homing direction H_S. Given only landmark bearing information, each correctional vector attempts to improve the perceived bearing of its landmark to better match that observed from home H.

The robot therefore 'slows' (i.e. makes smaller steps between snapshots) as the error decreases in response to the increasing proximity of home. In this way the robot should successively home in on its target without overshooting significantly.

In summary, the homing procedure involves successively moving in such a way as to try to incrementally improve the perceived position with respect to the goal. This is achieved by moving in response to a homing vector derived from a comparison of remembered and perceived landmark bearings. At each new position, landmarks are detected, correspondences made, and a homing vector computed. The process is repeated until either, the current and home views are sufficiently alike, a local minimum is detected, or a threshold on the maximum number of homing steps has been reached.

6.4.2.4 Analysis of Modified Homing Direction

It can be argued that one can improve on and refine the homing direction as calculated in the previous section.

For example, consider the computed homing vector for the simple situation depicted in figure 6.3. Given the two landmarks, what homing vector should be followed to return from the current position S back to position T (home)? The homing vector H_S is computed as usual by summing the correction vectors V_A and V_B. However, now consider the situation in reverse. What is the computed homing vector if the robot was at home (T) and was trying to get to its current position (S)? If the original homing vector were correct then it would make sense that the homing vector of the reversed situation would simply be 180 degrees out of phase. It is not.

The reason for this can be seen quite clearly (fig. 6.3) to be a result of the differing bearings, of the perceived landmarks, as seen from the two vantage points. The landmark bearings, as seen from one vantage point (T), are obviously going to be different from the landmark bearings observed from a different vantage point (S). This therefore implies differences in the corresponding correctional vectors and hence the resulting homing vector. Depending on the situation, this difference in the effective homing vector can be large or small. In figure 6.3, the homing direction computed from the reversed scenario H_R ($H_T + 180°$) is significantly different from H_S.[12]

[12]This 'reversed homing scenario' is a naturally intuitive way of understanding the logic of the improvement. However, the homing vector computed via this construct is, in fact, equivalent to

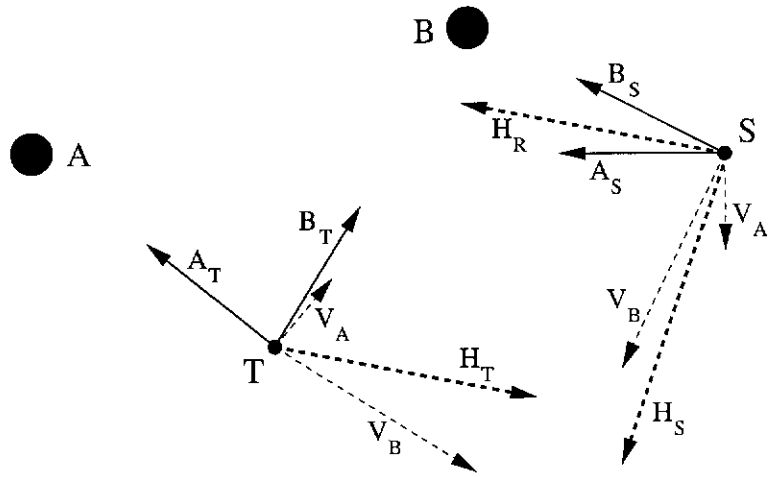


Figure 6.3: 'Improving' the Homing Direction. When homing from position S to T, the homing direction H_S may be 'improved' by incorporating the homing direction calculated from the reversed scenario H_R (i.e. homing from T to S).

Given this alternative view, one can attempt to 'improve' the homing vector by defining the new homing vector as being the average of the homing vectors in the two scenarios. This is in fact exactly what Hong et al. (1990; 1991; 1992) have done.

However, this avenue of improvement has been ignored here because it interferes with the inherent landmark avoidance feature of the homing algorithm. Although this is hinted at in figure 6.3, it is only when the resulting behaviour is viewed that the real significance of the effect can be realised.

Figure 6.4 shows a few examples of how this 'improvement' can affect the homing behaviour. Each subfigure shows three robot paths corresponding to the use of the three homing vector variations: (i) original homing vector (H_S), (ii) 'reversed scenario' homing vector (H_R), and (iii) average of (i) and (ii). These paths will be referred to as path O, path R, and path A, respectively.

In figure 6.4(a) path O follows a curved path, safely circumnavigating the closest landmark, ending up at home (open circle). Path R, however, is immediately drawn to the closest landmark and is almost caught in what is effectively a local minimum. This 'fatal' attraction and the resulting inefficient zig-zag path is caused by the fact that the 'reversed scenario' effectively dictates a mirroring path to that dictated by the original homing vector. This can

simply having the correctional vectors (in the original scenario) direct movement perpendicular to the bearings observed in the home snapshot (rather than those in the current snapshot).

be interpreted as an attempt to circumnavigate the landmark on the opposite side. This mirroring behaviour, however, is essentially flawed because it simply causes the robot to get caught at the cusp between the two navigation options. Eventually the robot reaches a point where the original homing scenario perceives that the best improvement can now be gained by circumnavigating the landmark on the opposite side. These reversals of homing direction thus cause the zig-zag approach and continue until the 'offending' landmark is passed. Although resultant path A managed to successfully reach home, in perhaps a more efficient manner than O, it did make a much closer pass by the landmark.

In another example, figure 6.4(b) shows path O again giving a wide berth to the closest landmark before successfully homing in on its target. Path R also reached home, however it did so by effectively passing through the landmarks en route. Path A also successfully reached home, but again passed quite close by the landmarks.

In figures 6.4(c,d) path O again follows an indirect, yet safe path home. Paths R and A, however, both head straight for the nearest landmark and pass directly 'through' to eventually reach home.

Further, it is worth noting here that, due to the fact that, at this stage, only the bearings of landmarks are observed and taken into account to compute a homing vector, the homing paths are unaffected by landmark size.[13] Since, the actual size and proximity of landmarks are unknown it would seem best to avoid unnecessarily increasing this proximity whilst homing.

To show that these anecdotal example scenarios are not merely pathological cases that rarely occur, some basic statistics were gathered from a randomly generated set of simulated homing scenarios. For a given number of landmarks, 1000 random arrangements of landmarks were generated within which 50 random homing starting positions were chosen. Table 6.1 shows the average landmark distances produced by each of the correctional vector methods O, A, and R. Minimum landmark distance for a specific homing scenario, refers to the minimum distance[14] between the simulant and a landmark attained during the entire homing procedure, i.e. the closest approach to a landmark. The maximum landmark distance, on the other hand, refers to the largest observed separation

[13]One important exception to this, which will be pursued later, is caused by the visual occlusion of landmarks.

[14]The distance between robot and landmark is calculated centre-to-centre since they are treated simply as point-like objects at this stage.

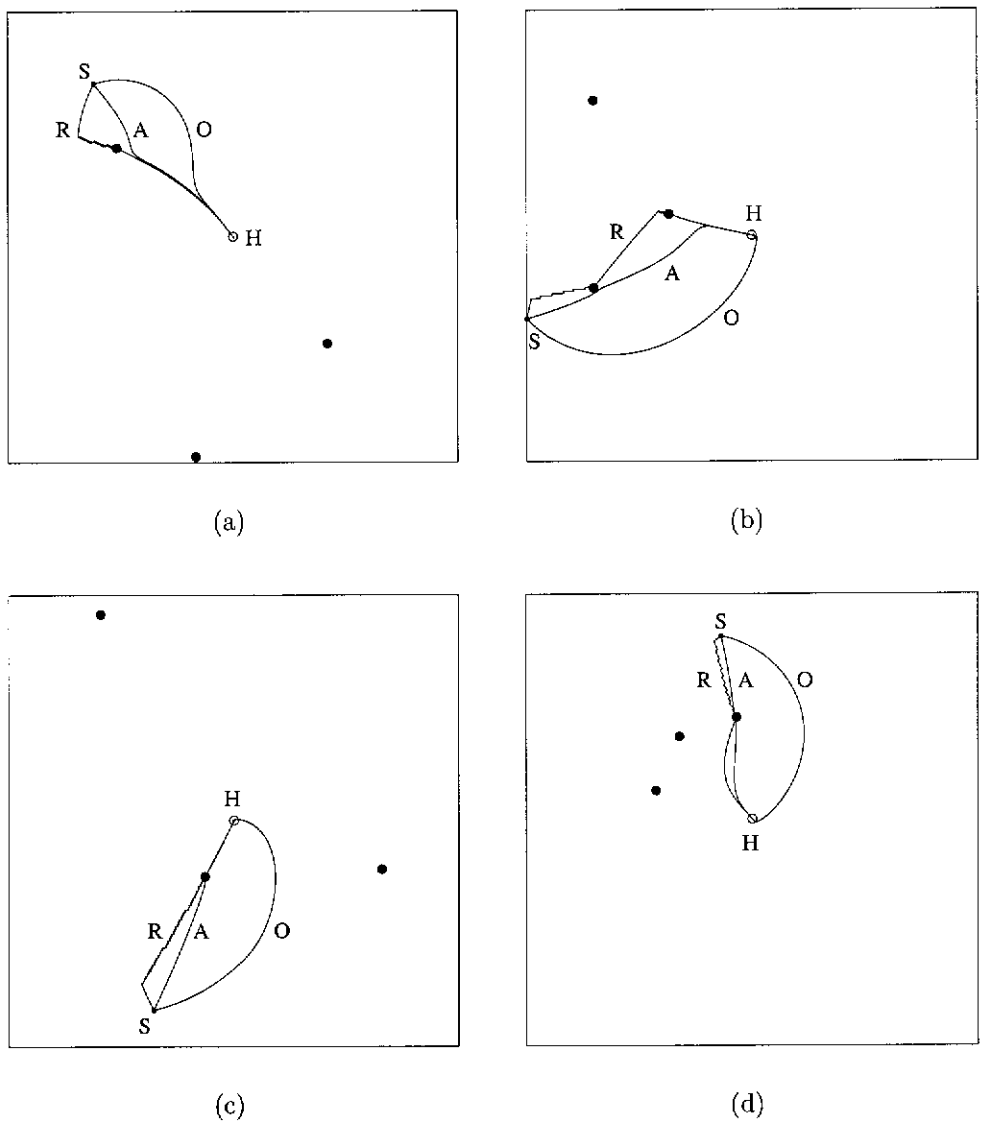


Figure 6.4: Homing Paths (simulation). Slightly different ways of computing the homing direction can result in significantly different homing trajectories: (O) Original homing vector; (R) Reversed scenario; (A) Average of O and R. The landmark avoidance characteristics of path O suggest the safest option. Each path starts from the same starting point (S), and returns to home (H).

between the robot and the current closest landmark.

Number of landmarks	Average min. landmark distance (cm) (Av. max. landmark distance)		
	(O)riginal	(A)verage	(R)eversed
4	165 (346)	145 (340)	123 (337)
6	129 (296)	113 (291)	94 (288)
8	107 (263)	93 (258)	77 (256)
12	80 (222)	70 (218)	57 (216)
20	54 (180)	48 (178)	41 (176)
50	25 (121)	23 (120)	21 (118)
100	14 (90)	13 (89)	12 (89)

Table 6.1: Average landmark proximity during homing. The average minimum and maximum proximity to the closest landmark observed during a set of simulated homing runs, show clear landmark avoidance differences between various methods of computing correctional vectors (O, A, and R). As can be seen, the method O provides for the greatest inherent landmark avoidance behaviour, followed by methods A, and R, respectively.

Although, these "average" statistics do belittle the significant path differences that can and do occur, they do show the consistent trends involved. Method O consistently evokes the safest path around landmarks, followed by methods A and R, respectively.[15] As expected, average (minimum) landmark distance also decreases with respect to increasing numbers of landmarks, irrespective of method.

[15]Homing performance can, however, also be affected. For example, for the 4 landmark case above, the set of simulation experiments showed a noticeable degradation in homing performance using methods A and R. If, for the sake of argument, we consider a final homing distance greater than 20 cm to be a failure then the failure rates for the three variants O, A, and R were 1.87%, 2.12% and 3.13%, respectively. Further, the maximum observed final homing distance for experiments using methods O, A, and R were 173, 221, and 637 cm, respectively.

6.5 Results I

A variety of results were obtained in both simulation and real-world experiments. Various ideas were pursued in simulation to determine theoretical performance, and then the lessons learned in simulation were used and verified in the real-world robot experiments.

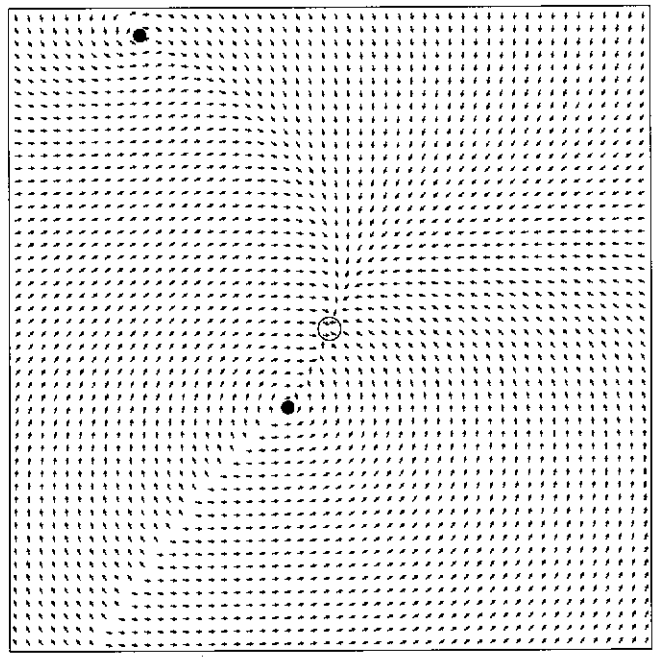
6.5.1 Simulation

Although the simulation provides an idealised and somewhat unrealistic environment, it is still of value in the testing of various homing strategies. The complete control over all parameters allows for a very powerful comparative tool. It can also quickly show the theoretical limitations of various approaches.

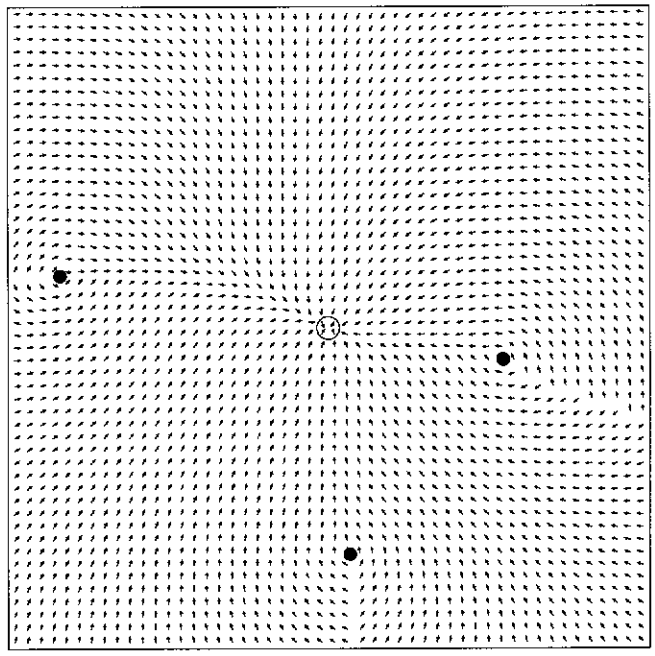
One particular area in which the simulator was found to be very useful was in showing the homing-vector field and resulting catchment area of an arbitrary arena, under various conditions. (This would be very time consuming if performed by the mobile robot, especially given the image acquisition technique described earlier.) The catchment area is defined by the points from which the robot can successfully home. For example, figures 6.5 and 6.6 show the homing vector fields for a variety of landmark configurations. The unfilled circle in the centre of the arena indicates the position of home, whereas the filled circles indicate landmark positions. Each arrow within the vector field indicates the home direction calculated for that point. The homing vector field essentially provides a more condensed view of the homing algorithm's performance. Although discretised, much behavioural information can still be gleaned from such views.

From these vector fields (figs. 6.5 and 6.6) it can be seen that the catchment area is essentially the entire arena. In other words, the robot can successfully reach home from any position within the arena. Reliable homing occurs even when the robot starts at locations well outside and far away from the constellation of landmarks (fig. 6.6). Of course in practice this is limited by the reliable detection of the landmarks involved. In the present set of simulations, utilising only bearing information, the robot simulant could always see the landmarks irrespective of position or distance.

A closer look at the homing vector fields, also reveals some of the implicit behavioural tendencies that the homing algorithm provides. One useful feature of the homing algorithm (independent of the landmark correspondence method),

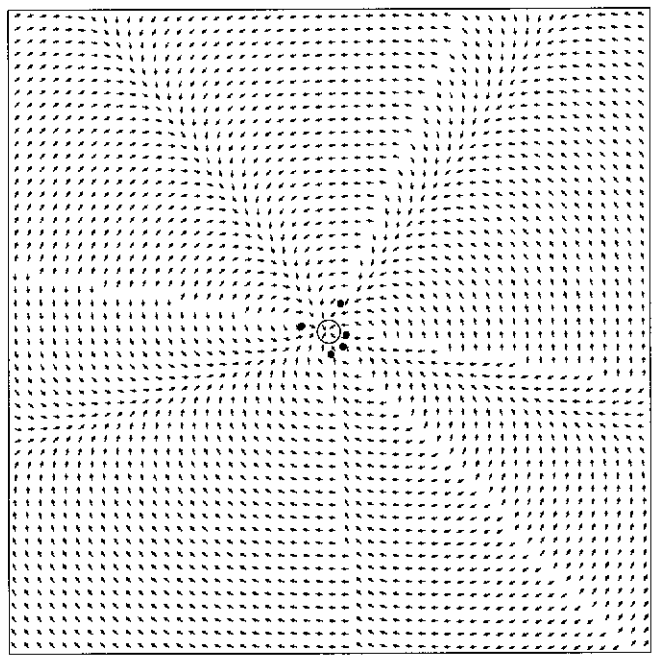


(a)

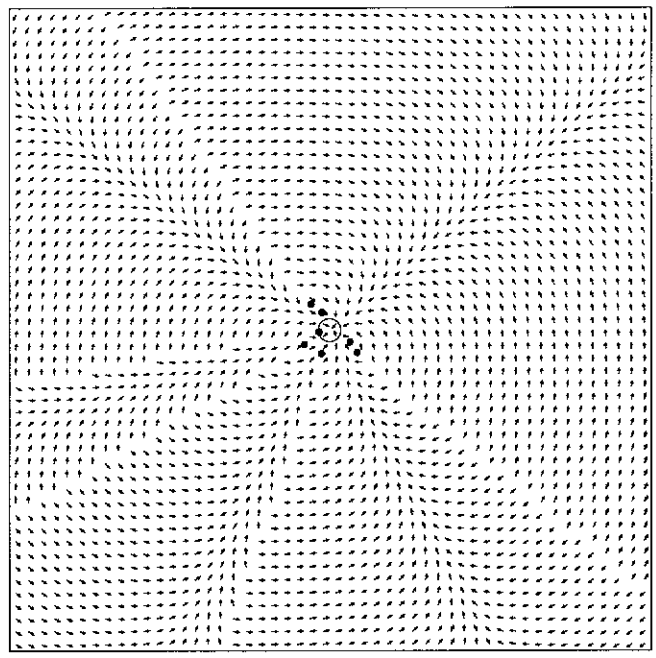


(b)

Figure 6.5: Homing-Vector Field (simulation, h1). Homing vectors are computed at regular intervals throughout the entire arena, showing the extent of the resulting catchment area. The open circle in the centre of each vector field indicates the position of home and the solid circles indicate the positions of landmarks. (a,b) Landmark avoidance is most prominent when homing with respect to a small number of landmarks.



(a)



(b)

Figure 6.6: Homing-Vector Field (simulation, h1). Homing vectors are computed at regular intervals throughout the entire arena, showing the extent of the resulting catchment area. The open circle in the centre of each vector field indicates the position of home and the solid circles indicate the positions of landmarks. (a,b) Successful homing also occurs when relatively far from home.

which is borne out by the simulation results, is the implicit tendency for the robot to avoid landmarks. This can be seen quite clearly in figure 6.5, where the numbers of landmarks are small. Here the homing vectors show the tendency of the robot to circumnavigate some of the closer landmarks first, instead of immediately heading straight for home. This is a natural byproduct of the homing algorithm tending to move the robot in such a way as to correct the (instantaneous) bearings of the landmarks as fast as possible. Landmark avoidance is primarily a result of directing correctional movement perpendicular to currently observed landmarks, but is also partially a result of weighting the correctional vectors to induce motion that will tend to correct the bearings of the landmarks exhibiting the greatest error, faster than those exhibiting less error.

Figures 6.7(a-f) show the simulated robot successfully homing from arbitrary positions within various landmark configurations. These figures show the path of the robot as it homes from arbitrary positions back to the centre. For these results the distance traversed between successive snapshots was kept small (5cm) to give approximately continuous results. Again the tendency of the robot to avoid landmarks is quite distinctive in several of the runs (e.g. figs. 6.7(a,b)).

Figures 6.8(a-f) show the same homing behaviour, except in this case the distance between snapshots is much larger. These discretised results are essentially what is expected in the real-world experiments, given our panoramic capturing restrictions.

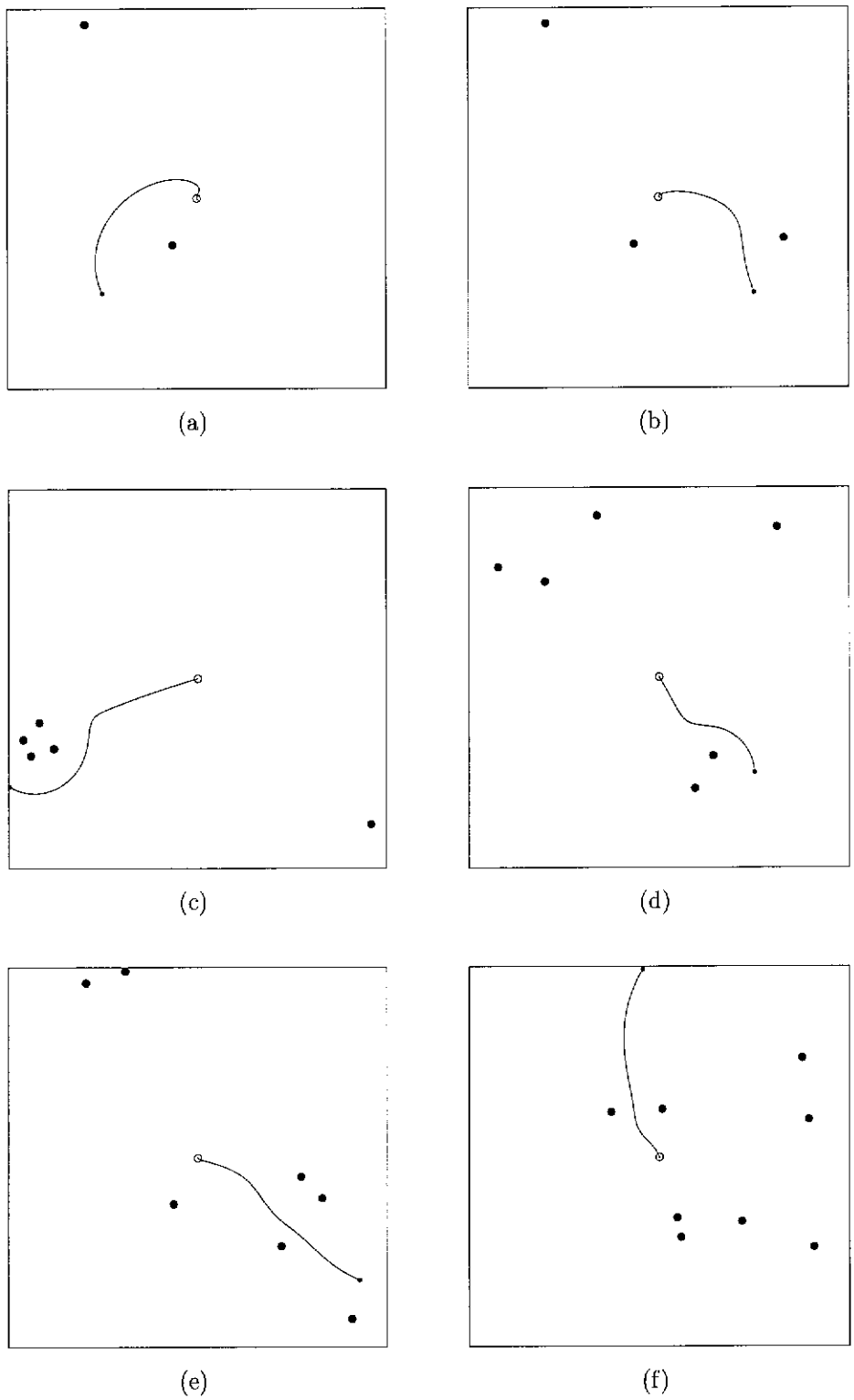


Figure 6.7: Continuous Homing Behaviour (simulation, h1). Individual homing trajectories show the path taken by the robot simulant on its return journey, in various arenas. A small inter-snapshot distance is used to approximate a continuous homing response.

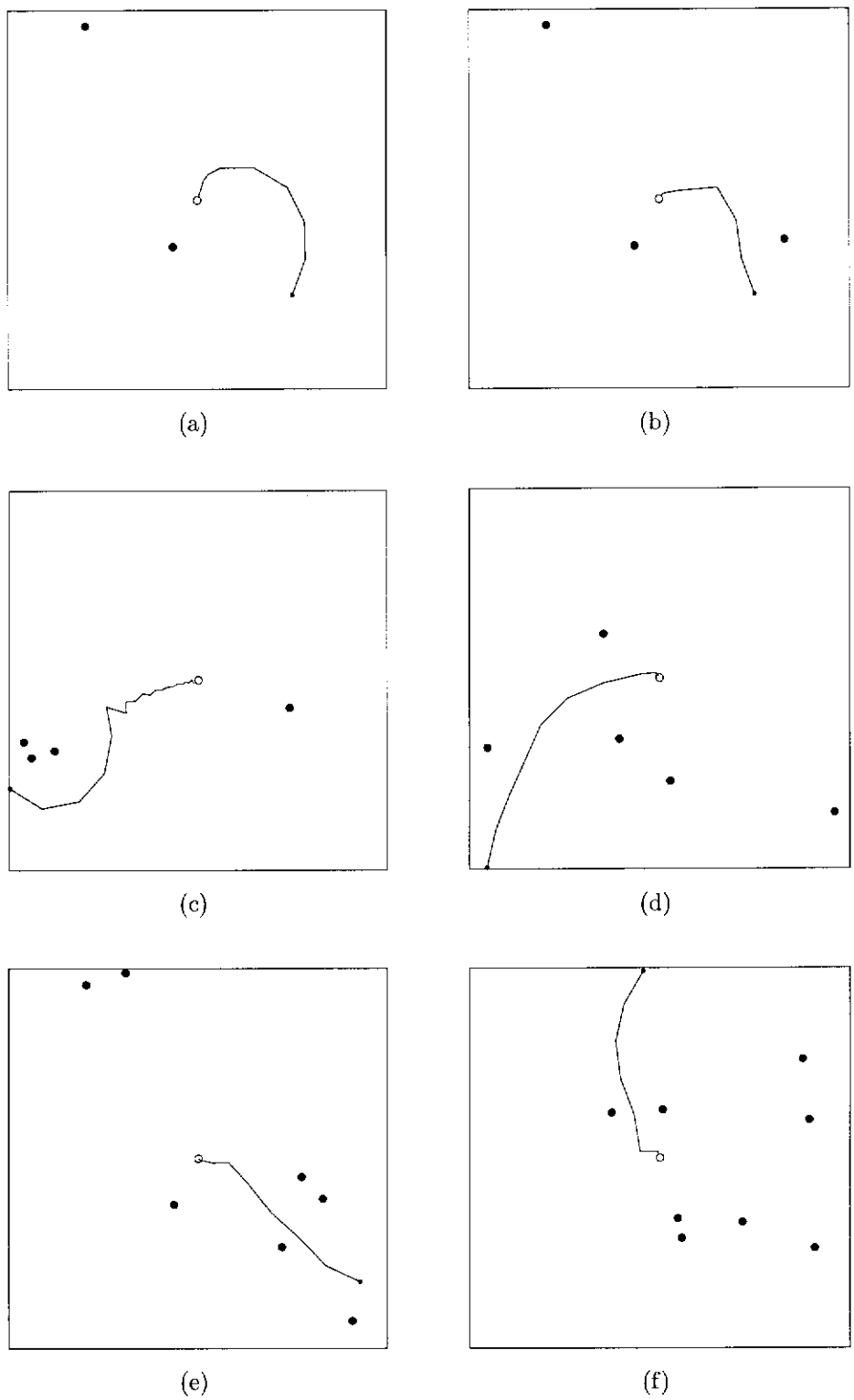


Figure 6.8: Discrete Homing Behaviour (simulation, h1). A variable inter-snapshot distance reduces the required numbers of snapshots whilst still retaining approximately the same homing path as the 'continuous' case.

6.5.1.1 Analysis of Simpler Landmark Correspondence

As stated earlier, several simpler, computationally safe, landmark correspondence methods have been explored. Although an exhaustive search produces optimal solutions, with respect to the snapshot similarity function expressed in equation 6.1, it can really only be safely applied when a small number of landmarks are being used.

It is therefore useful to have alternative correspondence methods available which are computationally cheap, with respect to the order of complexity. Figures 6.9, 6.10, 6.11, 6.12, and 6.13 compare the resultant behaviour of methods h1, h2, and h3, via homing-vector fields.

Differences can be seen between the homing vector fields generated with correspondence methods h1 and h2 (figs. 6.9–6.12). This is due to the inevitable increase in the chances of erroneous pairing up of landmarks when using the simplified method. However, this tends to only affect the homing path, not ultimate homing success.

Figure 6.13 shows the homing vector field produced by the homing algorithm using h2 and h3, within a very dense (50 landmarks) arena.[16] The superiority of method h3 over h2 can clearly be seen. (The number of landmarks, in this example, is too large to make method h1 computationally tractable.) Figure 6.13(b) shows that the h3 approach provides a perfect homing vector field. Whereas, figure 6.13(a) shows that with the h2 approach there is a small hole in the catchment area, close to home, where homing fails. This occurs where the computed homing vector is shown to point away from home and is indicated by the boxed regions. Essentially, this indicates a small local minimum near the goal. Although this seems minor it does show (anecdotally perhaps) that the h2 method does not provide an accurate enough landmark correspondence when close to the goal. The closer one is to home, the less the margin for error.

For a more detailed analysis of the alternative landmark correspondence methods, statistics for the degree of home-vector error (fig. 6.14), homing-vector error (fig. 6.15), and angular pairing error (fig. 6.16), rates of homing failure (tables 6.2 and 6.3), and final homing position (figs. 6.17, 6.18), are presented. In these graphs, method h0 refers to perfect landmark correspondence.

These statistics were generated by calculating, for each combination of correspondence method and number of landmarks, the vector fields for 1000

[16]See figure B.14 for the performance of h4, h5, h6, and h7 within this scenario.

random arrangements of landmarks. Each vector field is made up of 50^2 observations. We also gathered statistical data from actual simulated homing runs. Here again the homing performance is tested, varying both method and number of landmarks, on 1000 random arenas each. For each arena there were 50 random starting positions from which the robot was required to home. For the purposes of statistical evaluation the test arenas were given a scale. The area of each test arena is 1000^2 cm^2. For each experiment data concerning the degree of home-vector error, homing-vector error, rates of homing failure, and final homing position, were gathered. Home-vector error was calculated by comparing the computed homing-vector direction (γ) with the true bearing of home (ϕ). Thus, the home-vector error ($\sqrt{av(|\gamma \dot{-} \phi|^2)}$) is a measure of the accuracy of the homing procedure. The Homing-vector error, however, was calculated by comparing the computed homing-vector direction (γ) with that obtained under perfect correspondence (φ). Thus, the homing-vector error ($\sqrt{av(|\gamma \dot{-} \varphi|^2)}$) is more a measure of the navigational error arising from the mismatching of landmarks.

As expected, the rates at which landmarks are incorrectly paired (irrespective of method) monotonically increases with respect to both distance from home (i.e. range) and number of landmarks. However, for a true comparison of the correspondence methods, the effective homing vector statistics resulting from their use are examined. Figure 6.14 shows the relative performances resulting from each correspondence method. The exhaustive search method (h1), as expected, provided correspondences capable of supporting very accurate homing vectors. However, more surprisingly, method h3 is shown to be equally good, if not better, followed by h5, h6, h4, h2, and h7, respectively. Interestingly, h3 appears to be slightly more accurate in determining the true home vector than even h0. The one-to-many correspondence method h7, however, does not appear to be well suited to this 'bounded universe' domain.[17] As the number of landmarks increases the performance resulting from h7 decreases, especially when close to the goal (fig. 6.14(b)). Figure 6.14 also shows that the use of unit correctional vectors (h5u, h7u)[18] does not appear to be appropriate in this case either. Interestingly though, method h3u performed remarkably well, managing to consistently perform almost as well as h3 itself.

[17]This is despite the fact that method h7 actually produces a landmark pairing that is very low in terms of angular pairing error (fig. 6.16).

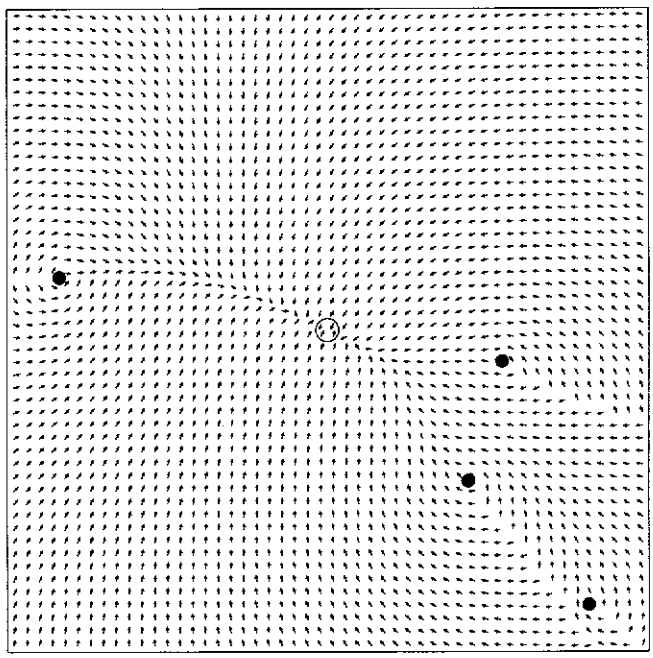
[18]Method h7u is a close approximation of the model proposed by Cartwright and Collett (1983), with the exception that, in this case, apparent size cues are ignored.

Failure rates are shown in tables 6.2 and 6.3. For the purposes of demonstration, homing failure occurs when the final homing position is further than 20 cm from home. For the vector field data, homing failure is defined as occurring when the catchment area is less than 100%. These results show that the failure rates of h0, h1, and h3 are almost identical. The efficient h3 method not only performs as well as the exhaustive search but also as well as is possible given perfect correspondence knowledge (h0). All of the correspondence methods, except h2 and h7, tend toward perfection as the number of landmarks increases. Interestingly, the accuracy of the h2 approximation seems to be beyond the threshold at which more landmarks would normally aid successful homing. Instead the addition of landmarks seems to be a hindrance rather than a help. The correctional vectors contributing to a correct homing vector are beginning to be outweighed by the erroneous ones. This is especially the case close to the goal where the margin for error is smaller. When close to the goal the correct correctional vectors will naturally be small, whereas the erroneous ones tend not to be. Although the failure rate of h2 does eventually peak (at around 20–50 landmarks) and starts to drop as the numbers of landmarks become very large, the failure rates of the other methods peak at a much earlier stage (i.e. relatively small numbers of landmarks).

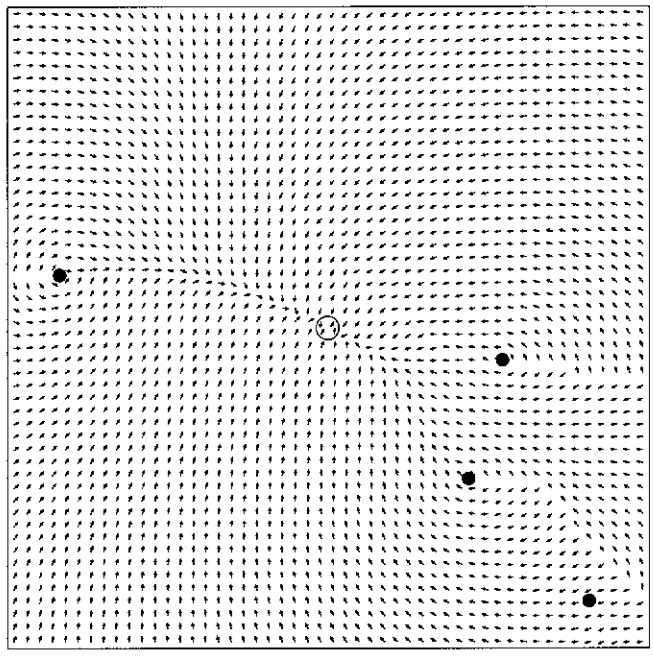
Finally, figures 6.17 and 6.18 show the behavioural performance with respect to the final homing position. Homing success can be gauged by how close to the home, on average, the robot is able to return. These results again show the superiority of method h3. The $O(n^2)$ landmark correspondence approximation still results in an equal performance to that of perfect knowledge.

It should be noted that in the simulations even perfect landmark correspondence (h0) has a non-zero failure rate. This is due to the chance occurrence of pathological arrangements of landmarks. When the landmarks tend to form a line running through home there can be an enormous bias with regard to the computed homing direction. Consider the situation when the robot is also positioned on this line, somewhere between the inner two landmarks around home. All of the correctional vectors will tend to be small and perpendicular to (and toward) this line and thus result in a small homing vector perpendicular to home. Once the robot has homed in on this line it tends to get caught in what is effectively a local minimum. The resultant behaviour is similar to that observed when homing using a single landmark as the only visual reference. That is, the robot will home in an arc about the landmark until it intersects the line formed

by the landmark and home. Naturally, as the numbers of landmarks increase, the chance of randomly generating a pathological test arena decreases exponentially.

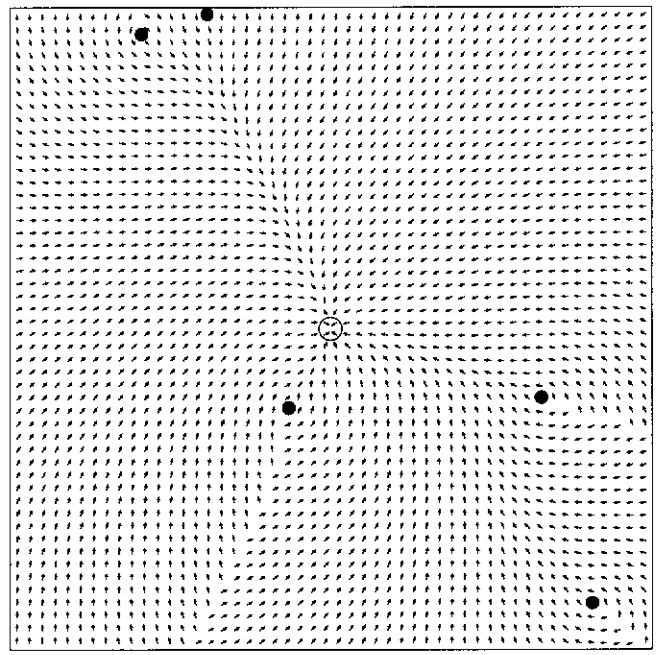


(a) h1

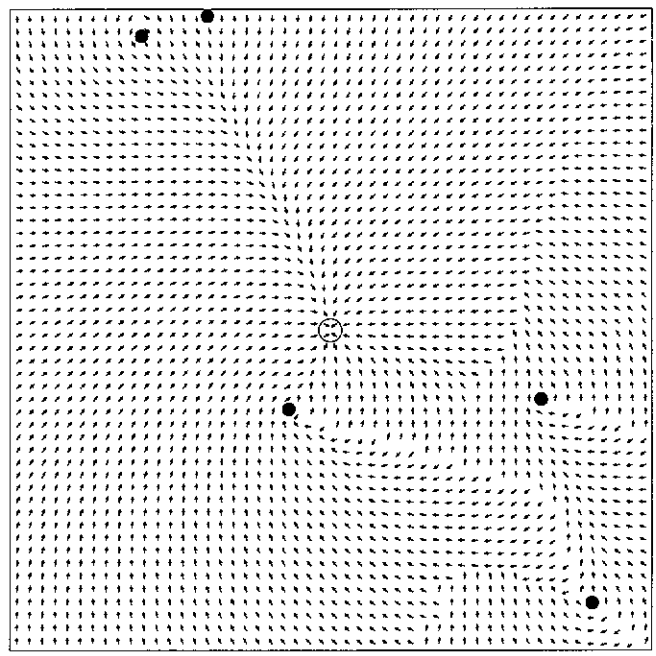


(b) h2

Figure 6.9: Homing-Vector Field (simulation, h1,h2). A comparison is made between landmark correspondence methods h1 and h2 by way of the resulting homing-vector fields. In the given scenario, differences can be observed in the resulting homing vectors and the effective homing trajectories they imply.

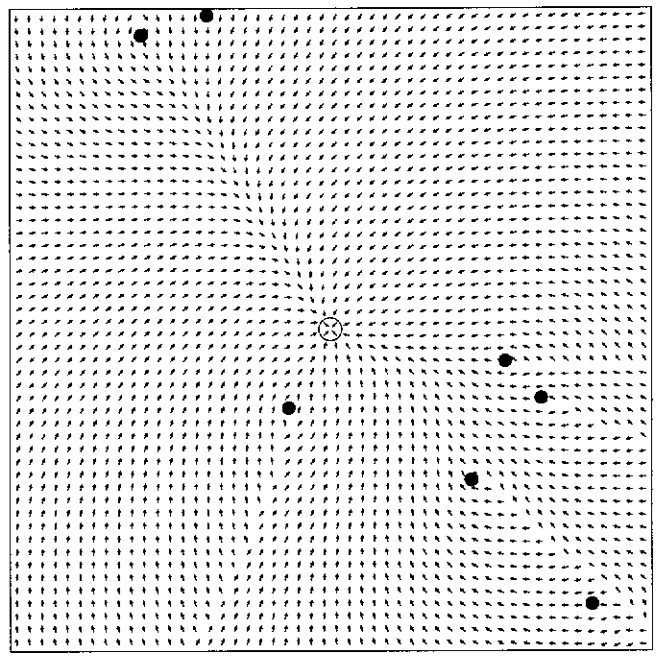


(a) h1

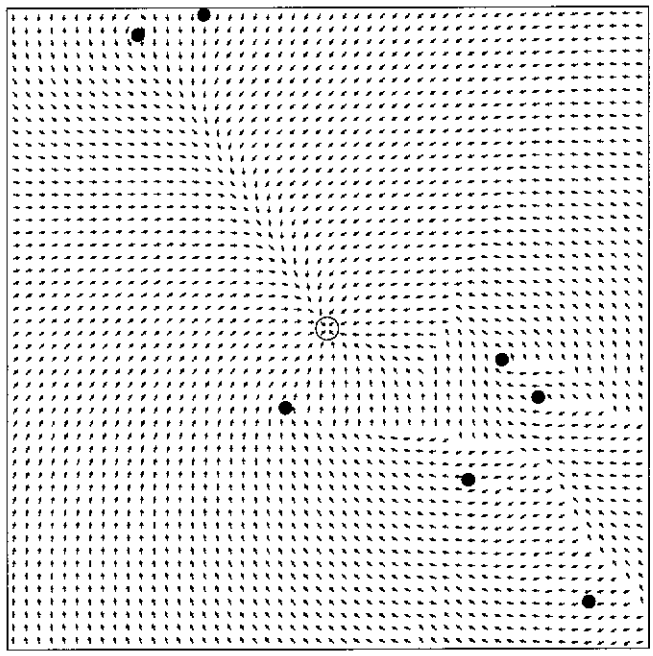


(b) h2

Figure 6.10: Homing-Vector Field (simulation, h1,h2). A comparison is made between landmark correspondence methods h1 and h2 by way of the resulting homing-vector fields. In the given scenario, differences can be observed in the resulting homing vectors and the effective homing trajectories they imply.

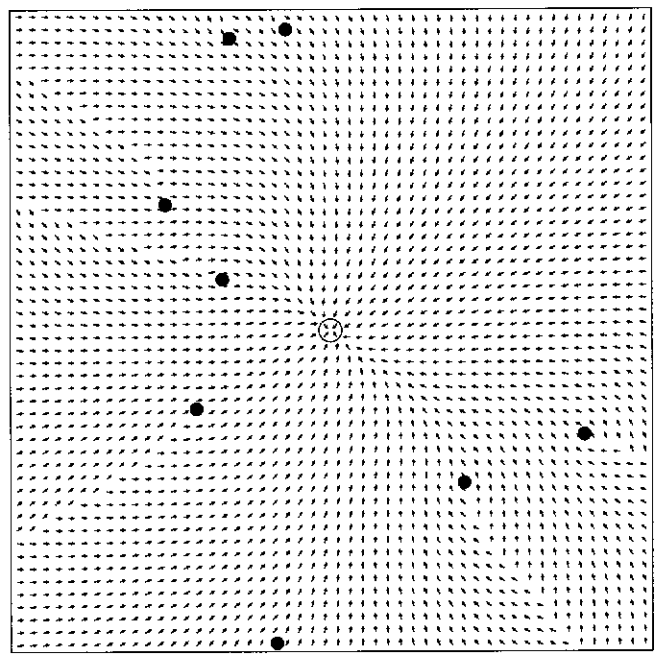


(a) h1

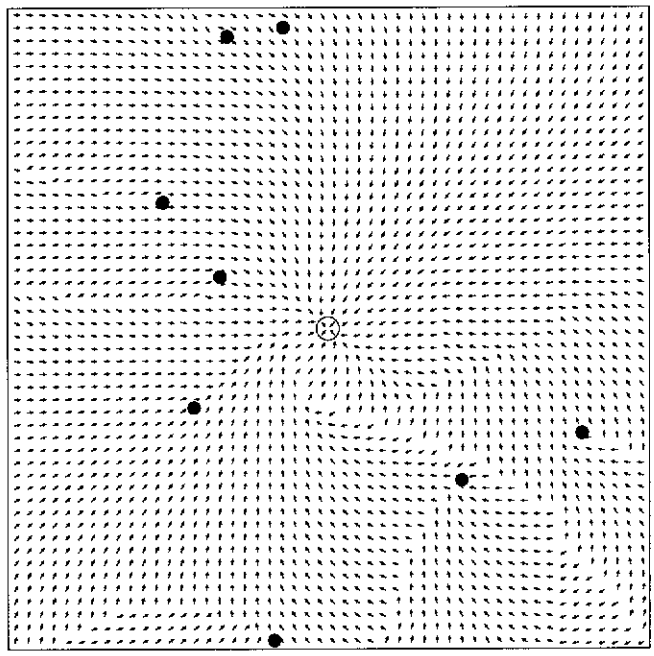


(b) h2

Figure 6.11: Homing-Vector Field (simulation, h1,h2). A comparison is made between the homing behaviour resulting from landmark correspondence methods h1 and h2, within increasingly dense arrangements of landmarks. Resulting homing behaviour can be seen to differ in the example scenario.

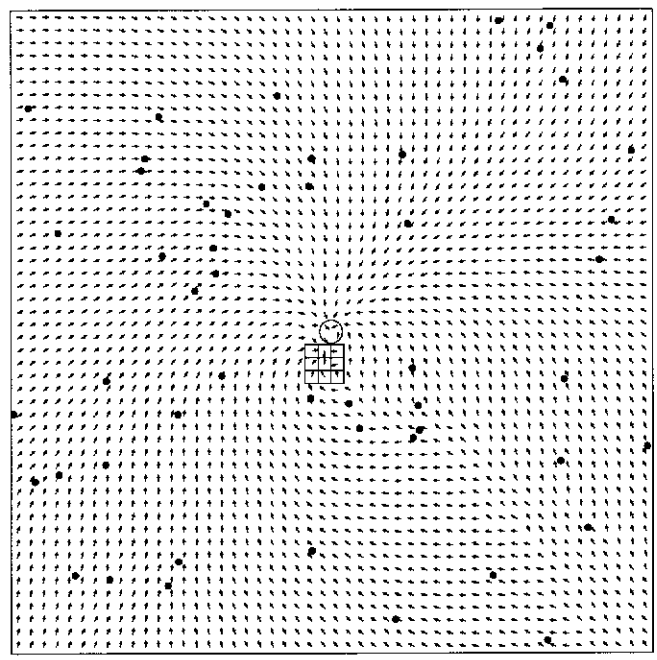


(a) h1

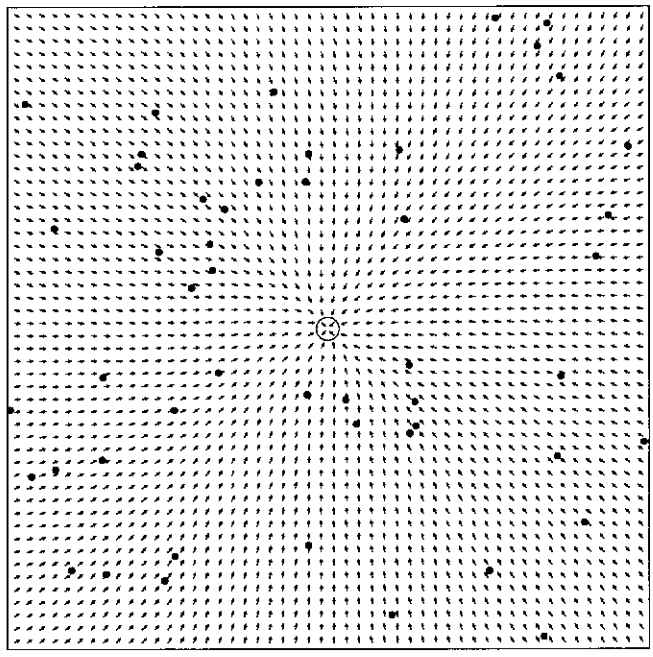


(b) h2

Figure 6.12: Homing-Vector Field (simulation, h1,h2). A comparison is made between the homing behaviour resulting from landmark correspondence methods h1 and h2, within increasingly dense arrangements of landmarks. Resulting homing behaviour can be seen to differ in the example scenario.

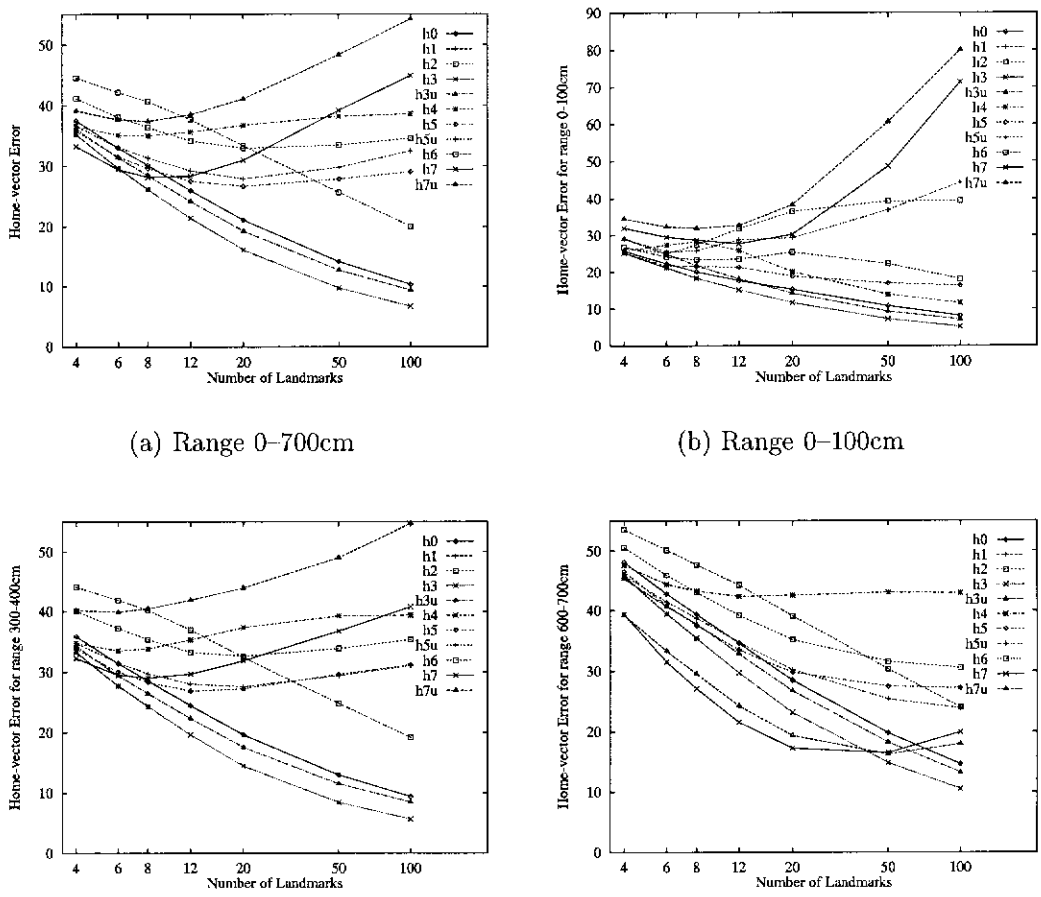


(a) h2



(b) h3

Figure 6.13: Homing-Vector Field (simulation, h2,h3). A comparison is made between the homing performance resulting from landmark correspondence methods h2 and h3, within a very dense arrangement of landmarks. Method h2 is seen to be both less accurate and less reliable than h3. The boxed region in (a) indicates a hole in the catchment area.



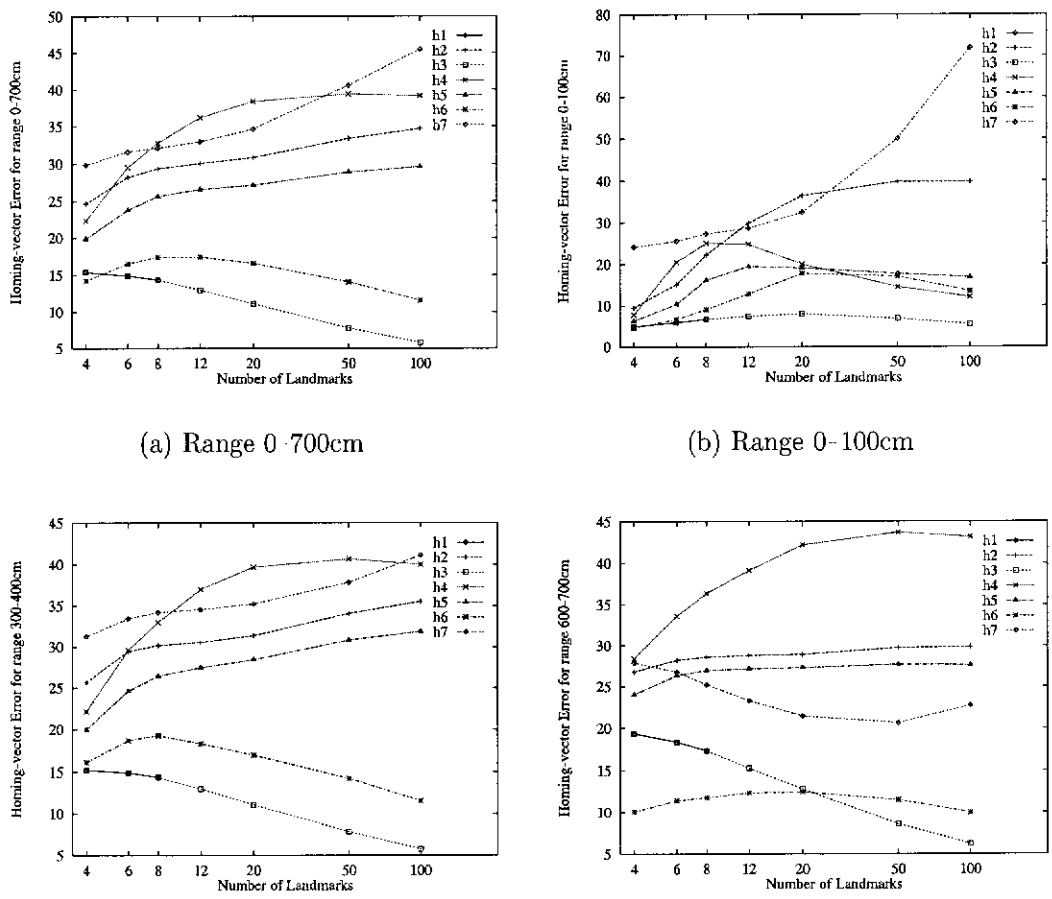
(a) Range 0–700cm

(b) Range 0–100cm

(c) Range 300–400cm

(d) Range 600–700cm

Figure 6.14: Average Home-Vector Error (simulation). Homing accuracy is gauged by the average angular difference between the computed homing-vector direction and the true bearing of home ($\sqrt{av(|\dot{\gamma-\phi}|^2)}$). This error is graphed for the various correspondence methods and presented over four homing ranges.



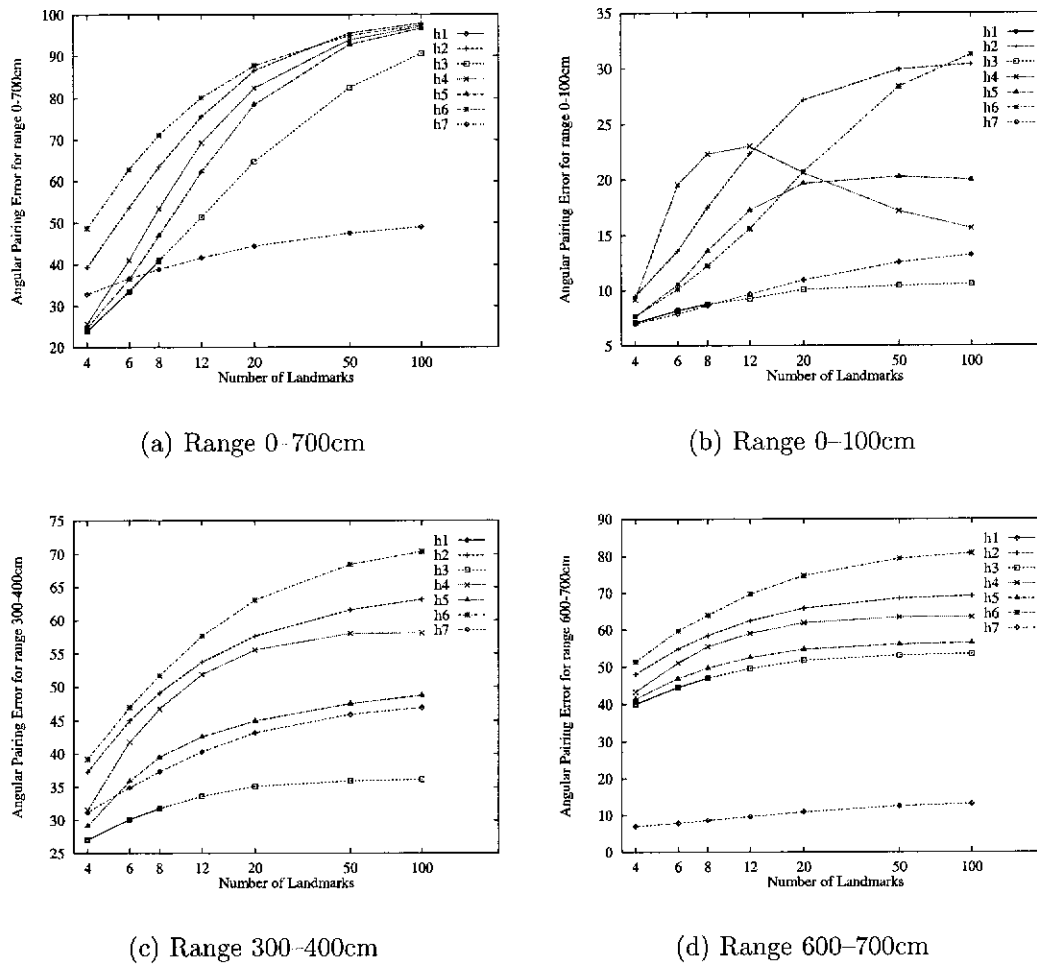
(a) Range 0–700cm

(b) Range 0–100cm

(c) Range 300–400cm

(d) Range 600–700cm

Figure 6.15: Average Homing-Vector Error (simulation). Navigational error (arising from the mismatching of landmarks) is gauged by the average angular difference between the computed homing-vector direction and that obtained under perfect correspondence ($\sqrt{av(|\dot{\gamma}-\dot{\varphi}|^2)}$). This error is graphed for the various correspondence methods and presented over four homing ranges.



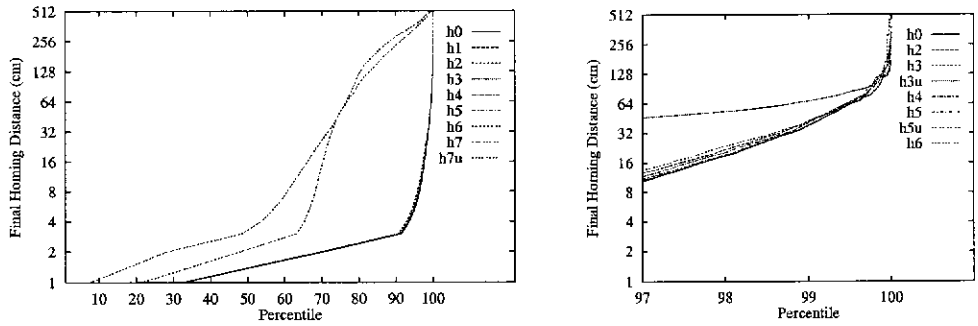
(a) Range 0–700cm

(b) Range 0–100cm

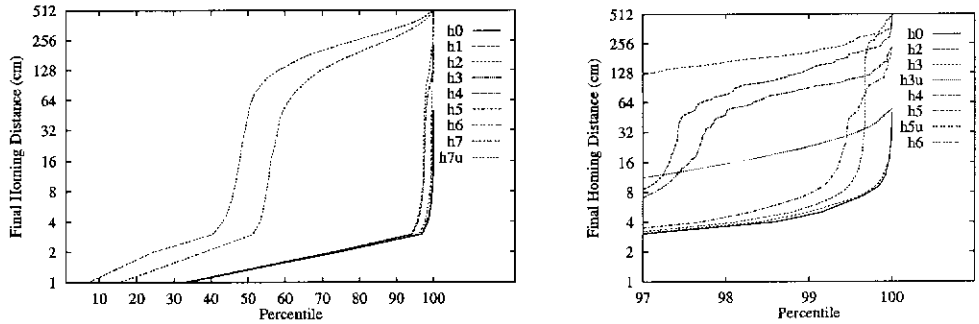
(c) Range 300–400cm

(d) Range 600–700cm

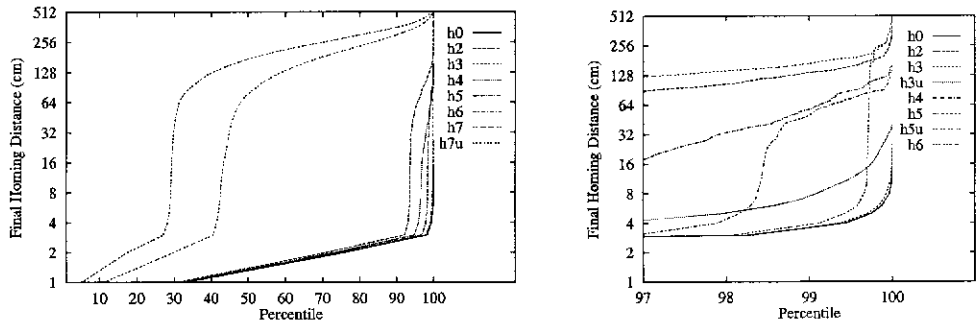
Figure 6.16: Average Angular-Pairing Error (simulation). The angular-pairing error is calculated from bearing disparities observed between the home landmarks ($theta_i$) that have been paired and the true corresponding home landmarks ($theta_{\phi(p(i))}$) that should ideally have been paired. This error provides a measure of how well the two sets of landmarks have been paired with each other ($\sqrt{av(|\theta_i - \theta_{\phi(p(i))}|^2)}$). This error is graphed for the various correspondence methods and presented over four homing ranges.



(a) 4 landmarks

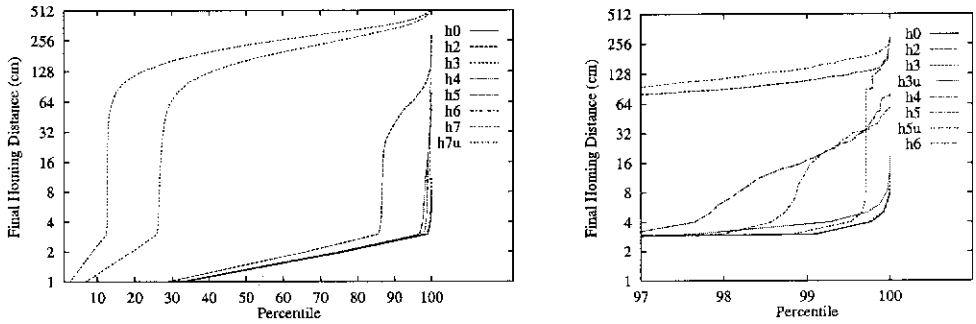


(b) 8 landmarks

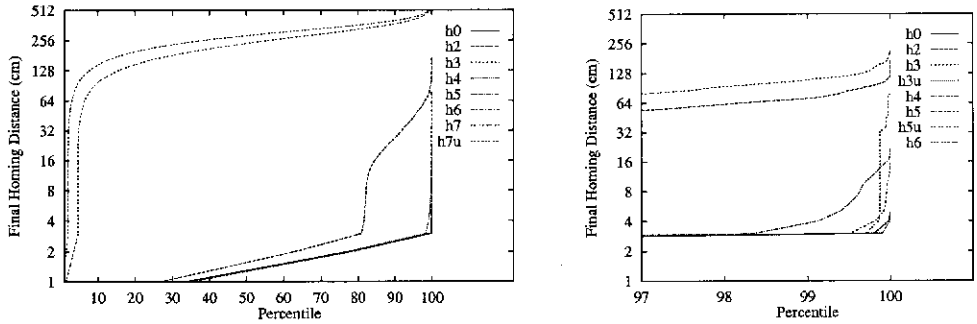


(c) 12 landmarks

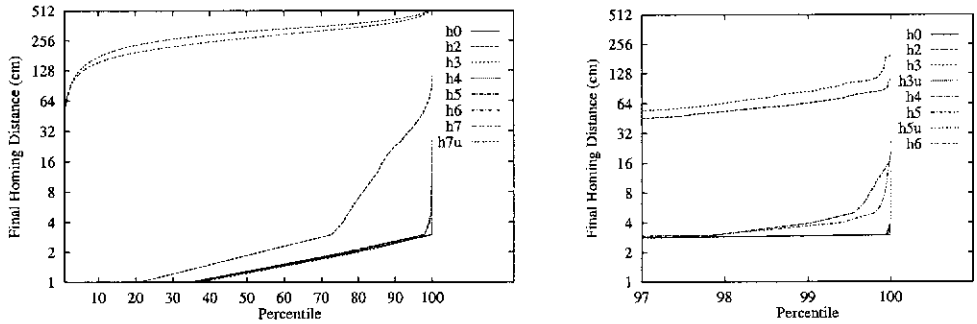
Figure 6.17: Final Homing-Distance Percentiles (simulation). The relative homing success of each correspondence method is gauged by how close the robot simulant is able to return to the true position of home. This is presented for various numbers of landmarks.



(a) 20 landmarks



(b) 50 landmarks



(c) 100 landmarks

Figure 6.18: Final Homing-Distance Percentiles (simulation). The relative homing success of each correspondence method is gauged by how close the robot simulant is able to return to the true position of home. This is presented for various numbers of landmarks.

Number of landmarks	Homing-Vector Field Failure (%)										
	h0	h1	h2	h3	h3u	h4	h5	h5u	h6	h7	h7u
4	2.2	2.2	3.1	2.2	2.3	2.6	2.2	2.3	3.1	71.5	73.5
6	0.5	0.5	3.6	0.5	0.3	4.4	0.6	6.0	2.5	85.0	91.5
8	0.0	0.0	8.8	0.0	0.2	5.5	2.7	11.7	2.8	93.2	97.8
12	0.0	—	19.5	0.0	0.0	4.5	2.2	18.1	2.0	97.1	99.6
20	0.0	—	29.5	0.0	0.0	0.7	0.6	18.1	0.3	99.6	100.0
50	0.0	—	21.0	0.0	0.0	0.0	0.0	12.4	0.1	100.0	100.0
100	0.0	—	12.0	0.0	0.0	0.0	0.0	5.9	0.0	100.0	100.0

Table 6.2: Homing failure rates for various correspondence methods. Homing failure is indicated by the percentage of simulation trials that produced a non-perfect catchment area, for various numbers of landmarks.

Number of landmarks	Homing Failure (% > 20cm)										
	h0	h1	h2	h3	h3u	h4	h5	h5u	h6	h7	h7u
4	1.874	1.872	2.12	1.872	10.80	2.017	1.909	10.9	2.270	29.1	32.3
6	0.476	0.476	1.46	0.476	3.52	2.046	0.556	4.8	0.780	37.6	42.1
8	0.022	0.030	2.64	0.030	1.34	2.346	0.598	5.6	0.365	43.2	51.9
12	0.000	—	6.29	0.004	0.17	2.861	1.528	9.2	0.298	56.2	70.6
20	0.000	—	12.89	0.000	0.00	0.807	0.815	13.2	0.294	72.9	87.2
50	0.000	—	13.34	0.000	0.00	0.004	0.000	11.9	0.128	95.3	98.0
100	0.000	—	11.40	0.000	0.00	0.000	0.000	8.4	0.002	99.4	99.5

Table 6.3: Homing failure rates for various correspondence methods. Homing failure is indicated by the percentage of simulation trials that ended with a final homing distance of greater than 20 cm, for various numbers of landmarks.

6.5.1.2 The Perception Horizon Problem

Although this chapter is not primarily concerned with the perception horizon problem, it is still worth mentioning that if the required homing distance significantly affects the observability of the same set of landmarks as that observed from home, then homing performance can also be significantly affected.

Up until now it has been assumed that all landmarks within the homing arena are perceivable. Furthermore, it has been assumed that landmarks outside of the homing arena are not perceived. What are the consequences of relaxing these assumptions and placing limitations on the range of perception?

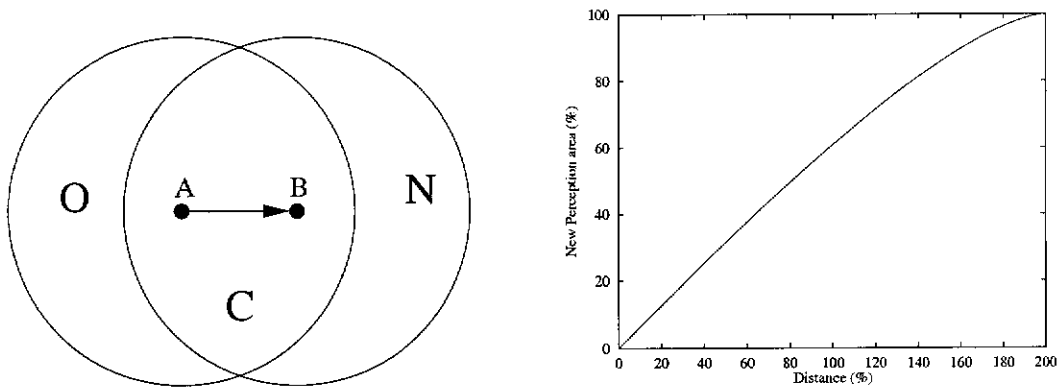
A 'fog of war' concept is a realistic one. No autonomous agent can realistically hope to possess complete knowledge of its environment. There will always be limitations on what is and can be perceived. Many factors can contribute to the limitations placed on perception and perceptual range. These can include such things as lighting, camouflage, proximity, apparent visual size, orientation, time, and occlusion.

Homing within what is effectively an infinite landscape, for example, necessarily places practical limitations on perception. The same can be said for homing within an open or outdoor environment, such as an orchard or perhaps a boulder strewn planetary surface. These scenarios introduce some interesting questions. Such as, how much of the perceived homing environment needs to be in common, between what is observed at the current and target locations, to make visual homing feasible? Or phrased in a slightly different way, at what point are additional snapshots required to maintain homing efficacy?

Consider the situation where visual navigation is attempted on the surface of a sphere, upon which a 'forest' of homogeneous visual landmarks reside. Due to the curved surface a perceptual limit is placed on the observability of the landmarks. In this case, the perception horizon can be defined by a circle centred at the observer with radius equal to the distance to the visual horizon. The detectability of a landmark is simply determined by its physical proximity.

A moving observer is now confronted by the fact that moving may cause current visual landmarks to disappear and new landmarks to appear (fig. 6.19). This complicates visual homing because the set of landmarks that are observed from one position may be significantly different to the set observed from another position. Further, due to the fact that the visual landmarks in question are considered to be homogeneous, there is no reliable way of determining which

landmarks are in common between two visual snapshots. The degree to which these sets of landmarks are dissimilar depends not only on the displacement distance between snapshots (relative to size of the perception horizon), but also on the specific relative spatial arrangements of the landmarks. Determining correct landmark correspondence thus becomes a much more difficult problem. In general, as the number of landmarks in common between two snapshots decreases, so do the chances of determining an appropriate homing vector. The resulting catchment area for successful homing within a 'visually unbounded' or open environment is therefore significantly restricted.



(a) (b)

Figure 6.19: Perception Horizon. In an open environment a perception horizon may be defined by the given distance to the visual horizon. Only landmarks located within this circle are detectable by the observer. (a) As the observer moves from location A to B new landmarks will be perceived in region N, whereas landmarks within region O will be lost. (b) The percentage of newly perceived terrain (N) is approximately linear to the displacement distance between snapshots ($\bar{AB}$).

As can be seen from figures 6.20 and 6.21, the robot can fail in determining an appropriate homing-vector despite being close to home. In these homing-vector fields, the perception horizon is indicated by the large open circle centred at home. Only the landmarks located within this radius (250cm), centred at the observation point, are detectable.

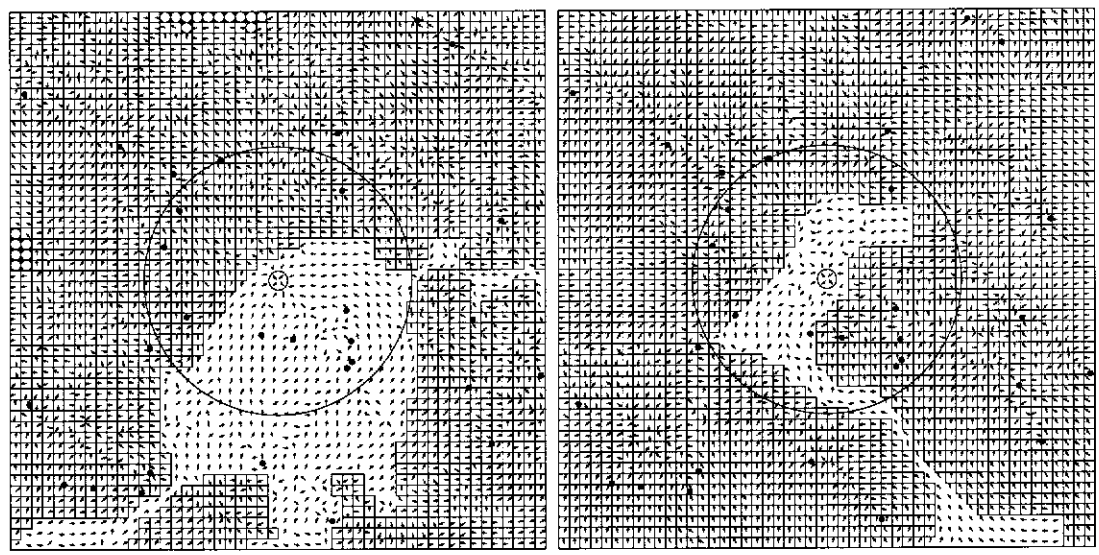
The homing performance of the landmark correspondence methods, both one-to-one (h5) and one-to-many (h7), are quite similar in this domain. Depending largely on individual scenarios, one correspondence method may provide a 'better' catchment area than another. However, none of the correspondence methods

examined can guarantee reliable homing at significant distances. However, within this domain, as indicated in figures 6.20(c,d), the use of unit correctional vectors (h5u, h7u) can now significantly improve the size and placement of the catchment area. Although, as figures 6.21(c,d) show, this improvement is not always guaranteed.

These results from the perception horizon experiments were, at first, quite surprising. As can be seen from the homing vector fields, even small movements, relative to the perception horizon, can cause significant problems in determining a viable homing vector. This is despite the fact that the majority of perceived landmarks may still be in common between snapshots. After closer examination, however, it becomes clear as to what is actually happening. The problem is primarily due to the fact that as the robot, and hence the circular perception horizon, moves, landmarks will tend to disappear and appear on opposing sides of the horizon (fig. 6.19). As the robot moves forward, landmarks will disappear from behind and new landmarks will appear from infront. This essentially equates to having to deal with erroneous landmarks having bearing errors of magnitudes approaching 180°. Despite being small in number, the erroneous landmarks are large in bearing based error. The resulting effect on the computation of a viable homing vector can thus be quite significant. Unfortunately, this is especially so when close to home. Ignoring for the moment the problems associated with correct landmark correspondence, remember that the inherent margin of error is less, the closer one is to home. This is due to the fact that as one approaches home the correctional vectors resulting from the correctly paired and perceived landmarks will decrease in magnitude. The same cannot be said for incorrectly paired or perceived landmarks. Therefore the corrupting influence of error, on the determination of a homing vector, increases with the proximity of home. Unfortunately, in the present scenario this effectively results in a 'no win - no win' situation. The closer one is to home the more corrupting erroneous landmarks are, and the further away from home, the less of the environment is common between the two snapshots, resulting in increased numbers of erroneous landmarks.

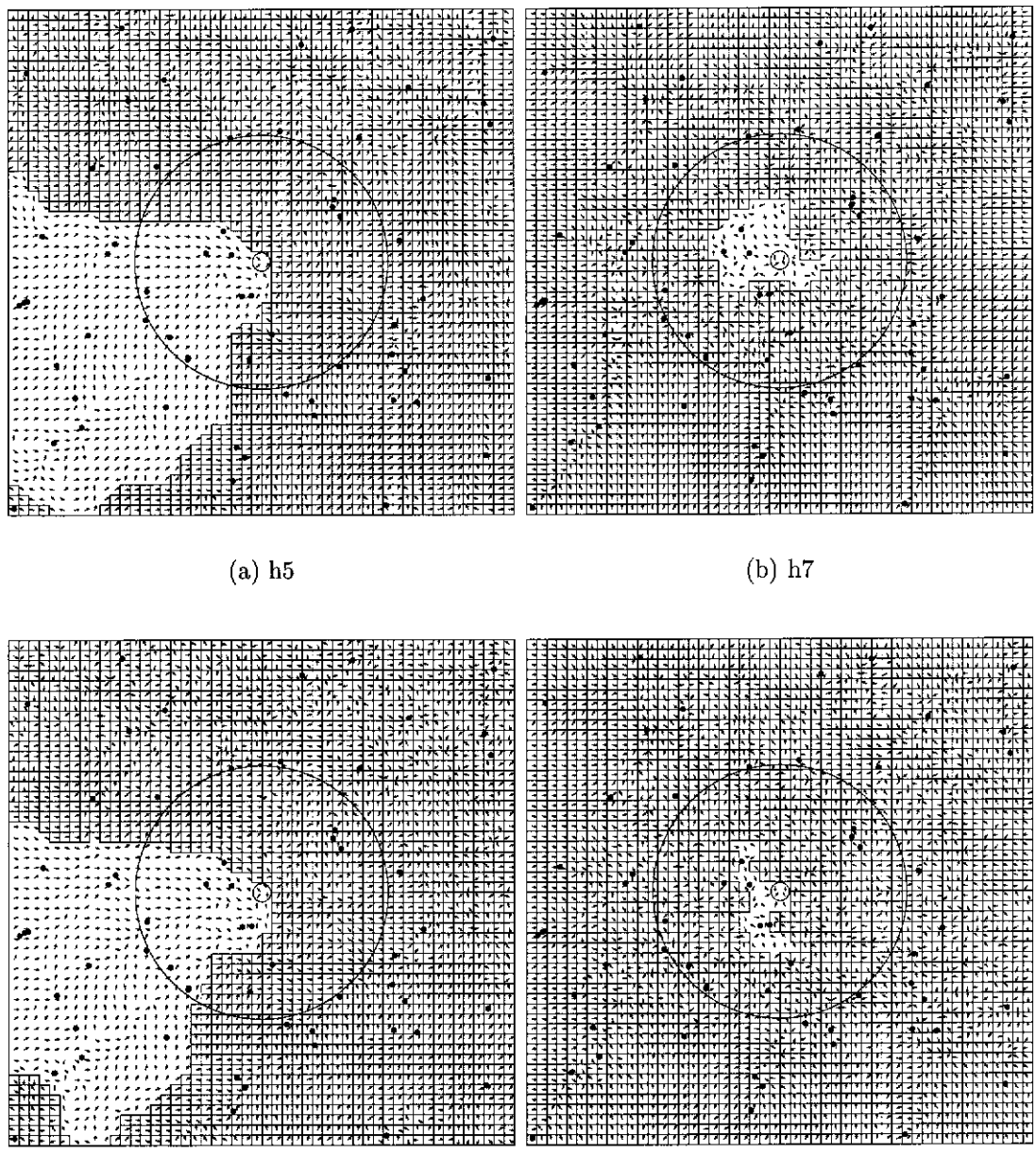
Although the "inherent margin of error" problem can be suppressed by simply using unit correctional vectors, it does not solve the "homing with a perception horizon problem". There still remains the underlying landmark correspondence problem. Erroneous landmarks of the form discussed above also have a significant disruptive effect with regard to landmark correspondence. This is not surprising given the anonymity of the landmarks and the limited perceptual information.

(a) h5 (b) h7



(c) h5u (d) h7u

Figure 6.20: Homing with a Significant Perception Horizon (simulation, h5,h7,h5u,h7u). As indicated, neither a one-to-one correspondence method (h5) nor a one-to-many correspondence method (h7) is able to provide reliable homing at significant distances within an 'unbounded' domain. However, unit correctional vectors (h5u, h7u) can now be used to improve the catchment area. In these examples the large open circle indicates the limit of possible landmark perception, from the observation point.



(a) h5 (b) h7

(c) h5u (d) h7u

Figure 6.21: Homing with a Significant Perception Horizon (simulation, h5,h7,h5u,h7u). As indicated by their catchment areas, the use of unit correctional vectors (h5u, h7u), in determining a homing direction, cannot always guarantee an improvement in homing performance.

Given the assumption of homogeneous landmarks, it is not possible to guarantee reliable homing at significant distances, in a reactive (i.e. no tracking of landmarks) fashion, within an environment which imposes a significant perception horizon problem. That is to say, reliable homing cannot be guaranteed, using a single memorised snapshot of the environment. Extended visual homing in certain domains necessarily demands the use of multiple snapshots of the environment.

6.5.1.3 Error Models

Up to this point the simulations have been run under the assumption of perfect and precise measurement and perception. This is unrealistic. In an effort to explore the effects of various forms of perceptual error, on the homing process, simulation experiments were conducted with varying degrees and types of error.

To this end, two basically separate error models were explored. One error model simply involved introducing gaussian error to the measured bearings of the landmarks. The other involved introducing error to the actual perception of landmarks. In this case, randomly chosen landmarks were removed, added, or replaced.

The results of introducing a gaussian error distribution to the measured bearings of perceived landmarks, is shown in figures 6.22–6.26.[19] Figure 6.22 shows the average home-vector error observed during sets of simulations.[20] As expected, the average home-vector error gradually increases with respect to the amount of landmark bearing error (fig. 6.22(a)). Although the amount of error has little effect on the calculation of a homing vector at large homing ranges (fig. 6.22(c)), its effect increases with the proximity of home (fig. 6.22(d)). When close to home the corrupting influence of bearing error is naturally at its peak (fig. 6.22(b)). However, as also shown, increasing numbers of landmarks help offset this disrupting influence.

Figure 6.23 shows how the introduced error, effects homing performance in terms of how close the robot is able to reliably return home. As can be seen, increasing amounts of bearing based error produces a gradual degradation of homing performance. In response to the increasing error, the robot is increasingly restricted by how close it can reliably home. Once again, this is offset somewhat by increased numbers of visual landmarks. The additional triangulation information effectively helps counteract the implicit error contained within a single bearing measurement. As increasingly more evidence is accumulated the individual error effects become increasingly diluted, thus assisting homing accuracy.

Finally, the homing performance is examined in terms of homing-vector fields. As is also suggested by figures 6.24, 6.25, and 6.26, homing performance tends

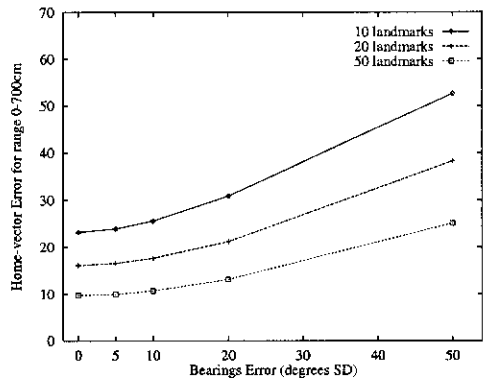
[19]The tabularised source data is detailed in §B.2, tables B.10–B.12.

[20]Essentially the same testing regime was carried out, as was previously conducted for the statistical performance analysis of the various correspondence methods (§6.5.1.1).

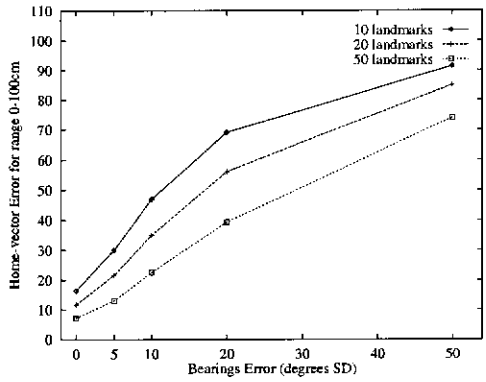
to degrade quite gracefully under increasing amounts of noise. As the degree of error within landmark bearing measurements increase, so too does the distance within which the robot can reliably return to home. The irregularities in the computed homing-vectors can be seen to increase with respect to the closeness of home. Further, these irregularities spread out further and become much more pervasive as the degree of error increases. Landmark bearing noise essentially effects homing accuracy, effecting how close the robot is able to reliably return to home. Once again, as the number of landmarks increases, this effect can be seen to decrease as the additional triangulation information improves the overall accuracy of the homing vector.

The second error model explored, involved removing, adding, and replacing randomly selected landmarks. In this way, an unreliable or simply imperfect landmark detection scheme can be simulated and thus help ascertain some of the effects this might have on the homing process. In the simulation experiments exactly the same basic behavioural and performance trends were observed as for the earlier bearing error model. As increasingly more landmarks are corrupted, homing performance degrades accordingly. Additional landmarks once again have a positive effect on performance. However, one notable difference observed with this model was the increased accuracy of homing vectors close to home. Due to the fact that landmarks were only being removed and added, not systematically altered, the chances of producing an accurate homing vector close to home were somewhat improved (as compared to further away from home), since extraneous landmarks are simply ignored when correspondences are made. When close to home, the legitimate landmarks will tend to be where they ought to be, and hence more likely be matched correctly with the corresponding landmark in the other snapshot. The ignored extraneous landmarks will thus tend to be the erroneous ones.

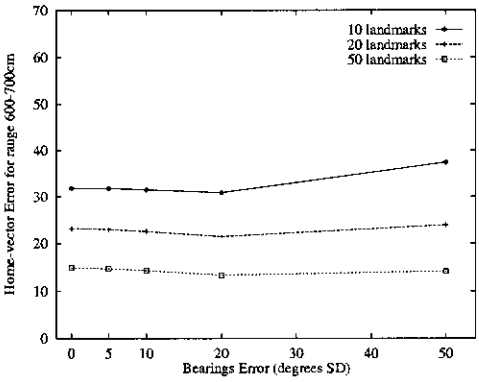
Although these error models are artificial and somewhat unrealistic, they do show the general trend of effects that various degrees of error have on the determination of a homing vector, and hence, homing performance. It is also evident what effect the proximity of home and the number of landmarks utilised in the homing task, has in relation to this.



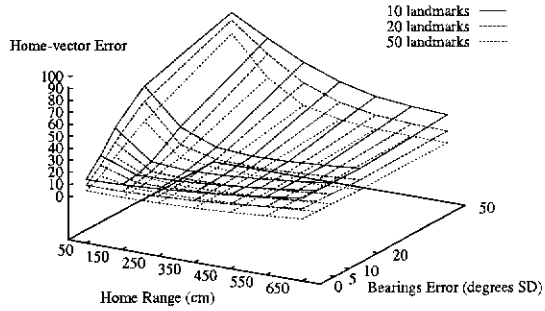
(a) Range 0–700cm



(b) Range 0–100cm

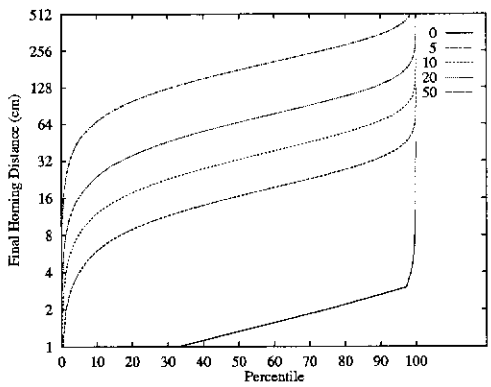


(c) Range 600–700cm

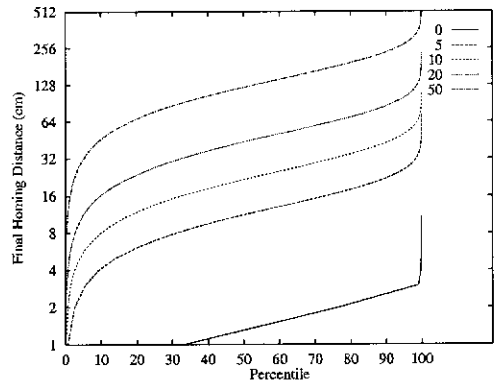


(d)

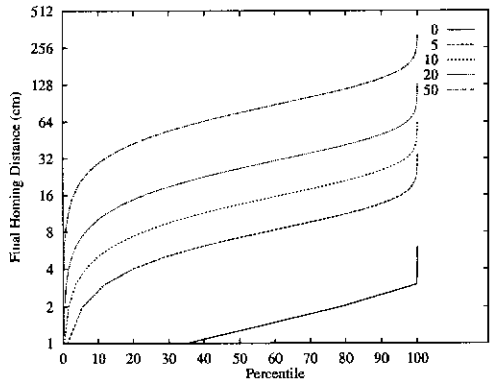
Figure 6.22: Home-Vector Error from Bearing Error Model (simulation, h3). Homing accuracy is again gauged by the average angular difference between the computed homing-vector direction and the true bearing of home ($\sqrt{av(|\dot{\gamma}-\dot{\phi}|^2)}$). The effect of various amounts of bearing-error (artificial error introduced to the measured bearings of perceived landmarks) on homing accuracy, is examined. Gaussian error is added in standard deviations of 0°, 5°, 10°, 20°, and 50°. As shown, the effects on short range performance (b) are much more severe than that at long range (c). The accuracy of the homing vector gradually degrades with respect to both increases in error and decreases in homing range. Increased numbers of landmarks, however, do tend to offset this degradation in homing performance (d).



(a) 10 landmarks

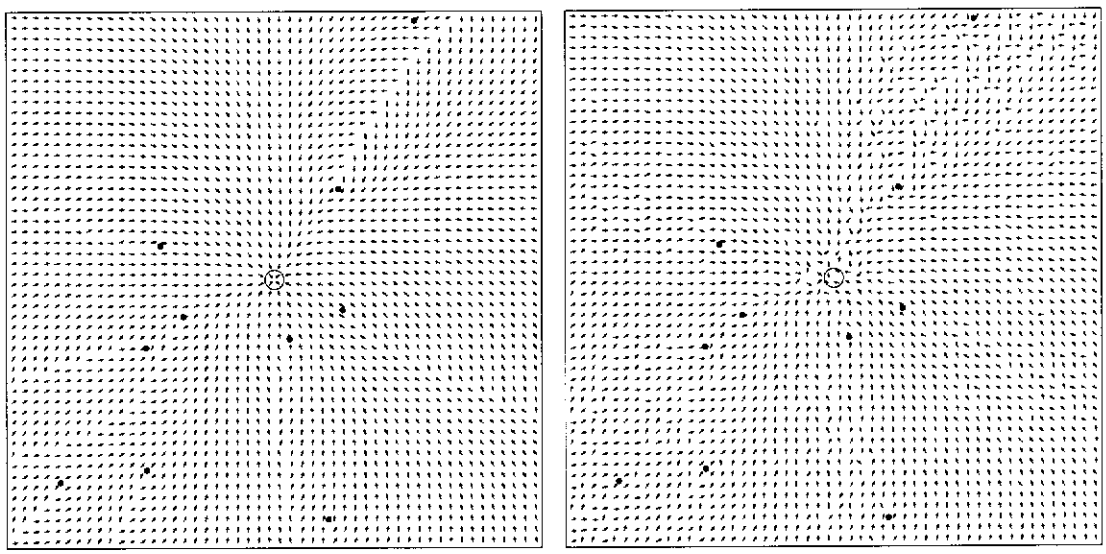


(b) 20 landmarks

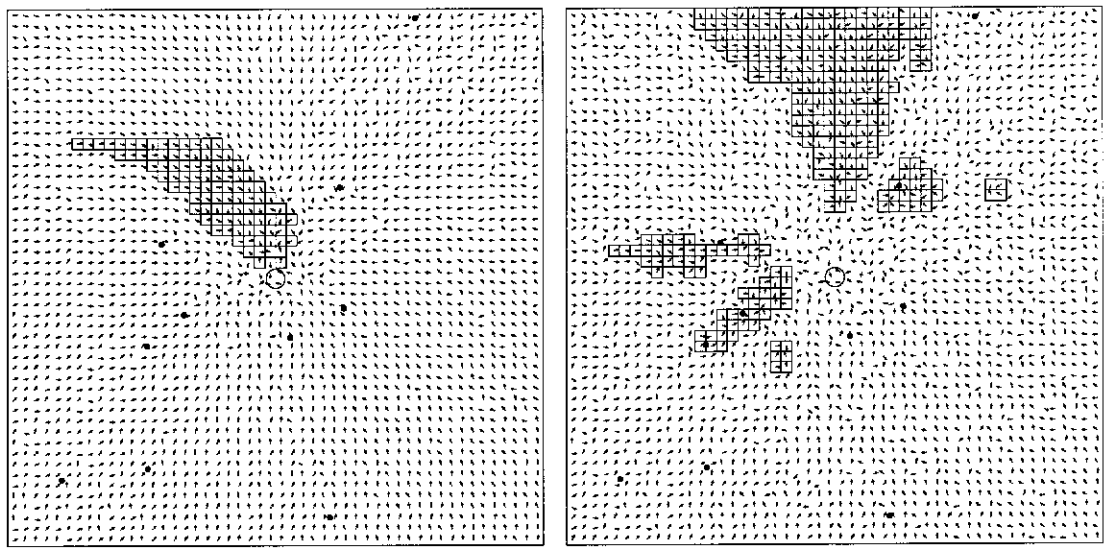


(c) 50 landmarks

Figure 6.23: Final Homing-Distance Percentiles (simulation, h3). How close the robot simulant is able to return to the true position of home, under various amounts of bearing-based error, is examined. Final homing-distance percentiles are shown for 5 error magnitudes: standard deviations of 0°, 5°, 10°, 20°, and 50°. Homing success, defined by final home proximity, is shown to gradually degrade under increased amounts of error (a, b, c). Again increased numbers of landmarks help offset this influence (c).

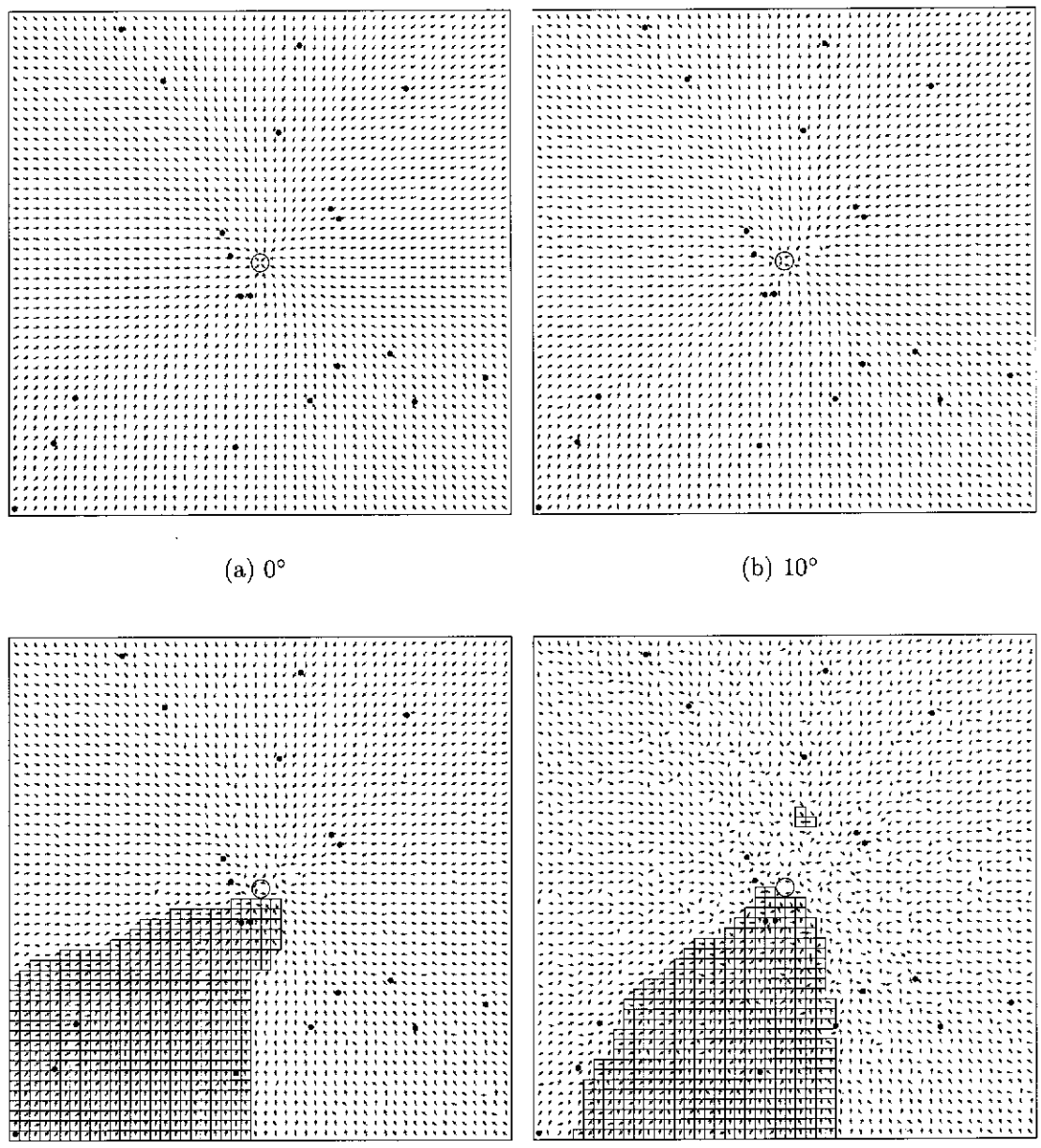


(a) 0° (b) 10°



(c) 20° (d) 50°

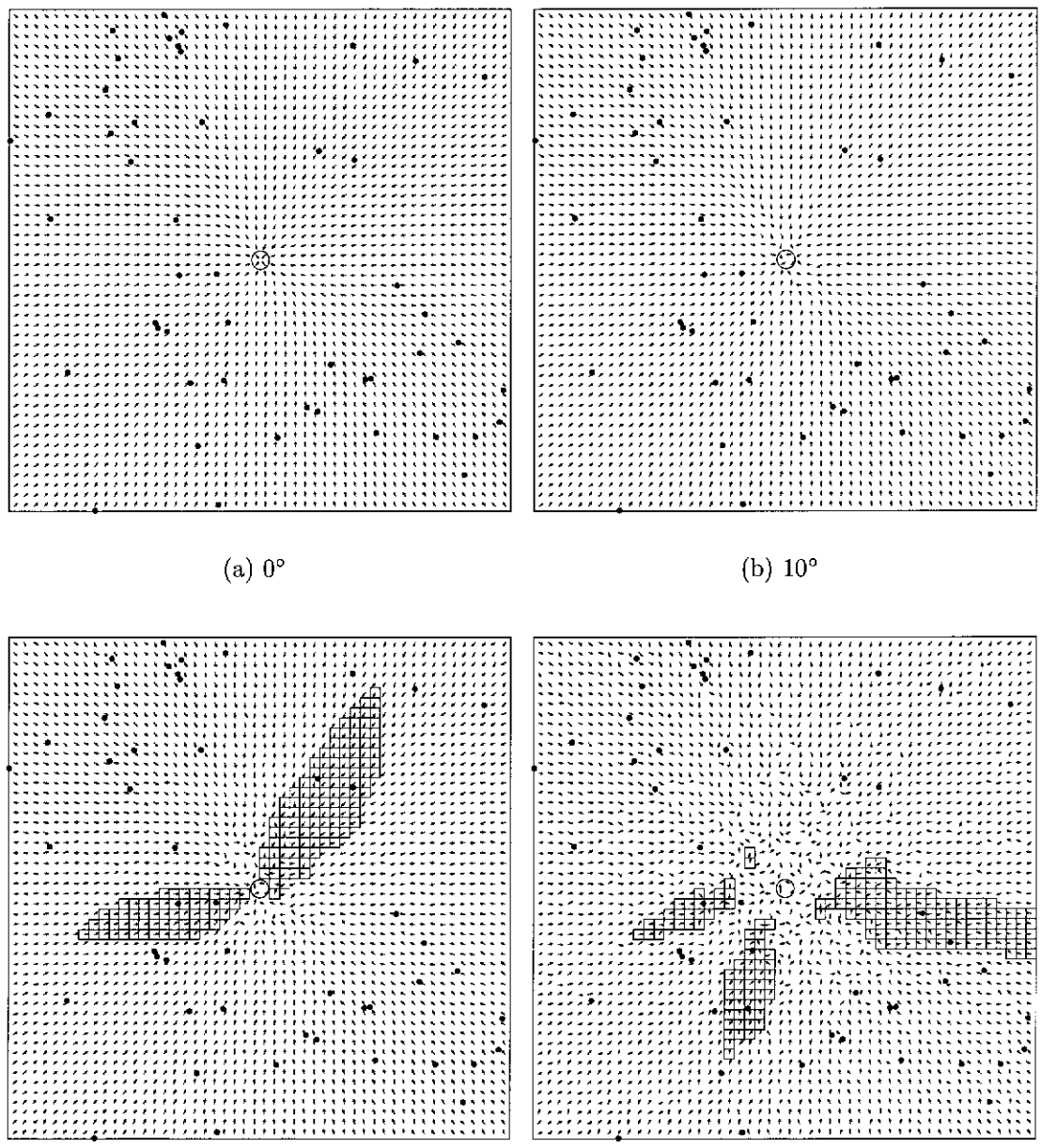
Figure 6.24: Homing-Vector Fields with Bearing-Based Error (simulation, h3). These homing fields show the gradual disruptive effect increased amounts of gaussian error (added to the measured bearings of perceived landmarks) has on the computation of homing vectors and the consequent degradation in homing performance. Gaussian error is added in standard deviations of 10°, 20°, and 50°. Homing-vector irregularities first emerge in the centre, close to home, and gradually increase and move outward, as the degree of error is slowly increased. In this scenario 10 landmarks are utilised for the visual homing task.



(a) 0° (b) 10°

(c) 20° (d) 50°

Figure 6.25: Homing-Vector Fields with Bearing-Based Error (simulation, h3). These homing fields show the gradual disruptive effect increased amounts of gaussian error (added to the measured bearings of perceived landmarks) has on the computation of homing vectors and the consequent degradation in homing performance. Gaussian error is added in standard deviations of 10°, 20°, and 50°. Homing-vector irregularities first emerge in the centre, close to home, and gradually increase and move outward, as the degree of error is slowly increased. In this scenario 20 landmarks are utilised for the visual homing task.



(a) 0° (b) 10°

(c) 20° (d) 50°

Figure 6.26: Homing-Vector Fields with Bearing-Based Error (simulation, h3). These homing fields show the gradual disruptive effect increased amounts of gaussian error (added to the measured bearings of perceived landmarks) has on the computation of homing vectors and the consequent degradation in homing performance. Gaussian error is added in standard deviations of 10°, 20°, and 50°. Homing-vector irregularities first emerge in the centre, close to home, and gradually increase and move outward, as the degree of error is slowly increased. In this scenario 50 landmarks are utilised for the visual homing task.

6.5.2 Mobile robot

Figures 6.27 and 6.28 show several examples of the robot homing behaviour in our real-world experiments. Here, the large, filled circles show explicit landmark locations, and the smaller circles show robot snapshot positions. The start position S is the location of the first snapshot whilst the small, filled circle represents the initial home snapshot position. Each snapshot position also has several vectors associated with it, as shown. The vectors emerging from the centre of the circle (snapshot position) show the bearings of perceived landmarks from that position. The other arrowed (and usually larger) vector shows the calculated homing vector which directs the robot to the location of its next snapshot and eventually home.

In figures 6.27(a,b), the homing behaviour of the real robot within a simple, minimal arrangement of landmarks is shown. For comparison purposes, simulation equivalents are also shown (figs. 6.27(c,d)).[21] Despite the aberrations caused by the imperfect panoramic capturing process and phantom landmarks, the real-world robot performed quite well and the homing behaviour observed matches that seen in simulation very well.[22]

In the examples depicted in figures 6.27(a,b) and 6.28(a,b) the mobile robot homes successfully, from position S to home, in only a few incremental steps. As can also be seen, the homing vector for each successive move is tending to decrease in length. As the robot gets closer to home, each snapshot tends to become more similar to the home snapshot, the resulting error becomes less and hence the reduction in homing vector size. This avoids significant over-shooting and oscillation whilst still keeping the number of time-consuming snapshot operations to a minimum.

Figure 6.28(c) shows a larger homing example where the robot is required to home from over 3 metres away via 8 explicit landmarks. Although there were only 8 explicit landmarks, there were in addition several 'implicit' landmarks (i.e. noise) detected sporadically at various positions in the environment. These extraneous landmarks do cause some interference with the homing. However, if the numbers of stable landmarks outweigh the transitory ones, the problem is not significant. The effect should only be in relation to the path taken, not overall

[21] For a more complete comparison see §B.3, figures B.19–B.21.

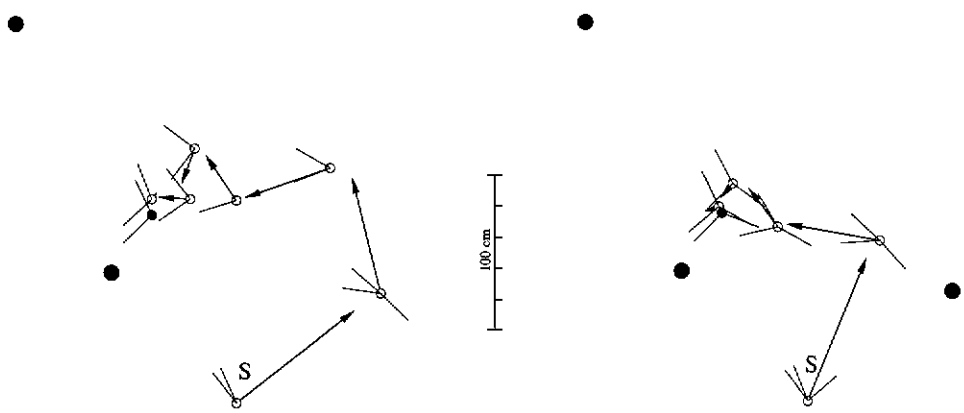
[22] Due to the relatively small numbers of landmarks used in the mobile robot experiments the correspondence method h1 was used.

success or failure. As the homing domain is restricted to a limited or 'visually bounded' indoor environment there is no significant perception horizon problem.

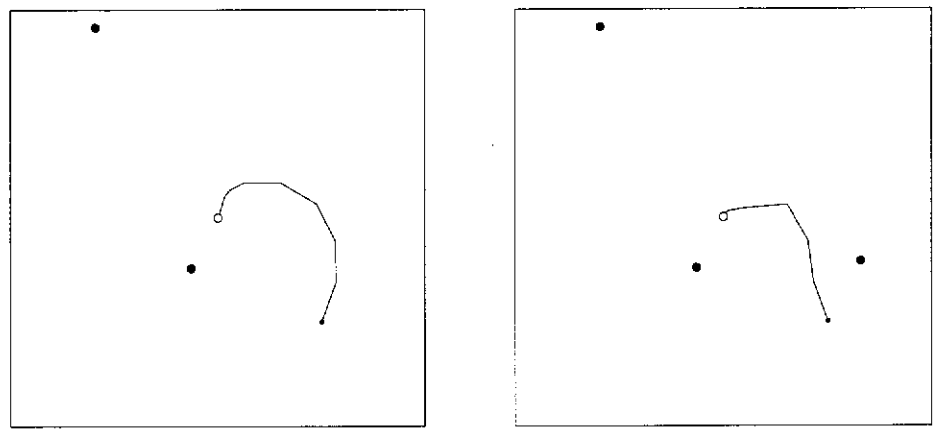
The reason for the seemingly high number of snapshots in figure 6.28(c), relative to 6.28(a) and (b), is to do with the dynamic threshold on the maximum allowed step size. (The step size being the prescribed distance between snapshots.) Although the size of the homing vector is primarily dictated by the error observed between paired landmarks, it is also limited by prescribed thresholds. The actual step size is determined by a linear relationship between the minimum and maximum expected error and the minimum and maximum allowed step size. In this way the homing vector is constrained between a minimum and maximum size whilst still providing a proportional magnitude with respect to the error. In an effort to intelligently deal with the unusual situation in which the error grows between successive snapshots instead of shrinking, the maximum step size is altered accordingly. When the error increases between successive snapshots the maximum step size decreases. Conversely, when the error decreases again the maximum step size increases. In this way the magnitude of the homing vector tends to decrease in response to an unusual trend in the observed error between successive (and presumably closer) snapshots and the home snapshot. This can be seen as a safety mechanism in response to an unexpected situation, i.e. when in doubt, move cautiously.

Within figures 6.28(a, b, c) there is a notable discrepancy between the target position of the next snapshot, shown by the homing vector, and the actual position of the next snapshot. This is due to the over-shooting nature of the robot's traversals and spins. This naturally makes little difference to the overall homing behaviour.

The end condition for the homing behaviour was simply a threshold on the average difference between paired landmarks. This compromise results in a reasonably close final position without having to resort to too many snapshots. With the discrete case there will always be a trade off between accuracy and required number of snapshots. As can be seen in figure 6.28(c) there is clearly still room for improvement after the end condition is met. The final homing vector is shown (although not acted upon) to improve the final position.



(a) real-world (b) real-world



(c) simulation (d) simulation

Figure 6.27: Homing Behaviour (real-world and simulation). A comparison of the homing behaviour seen in the robot experiments with that seen in simulation shows them to be quite similar. In the two examples depicted, the robot trajectories are very similar.

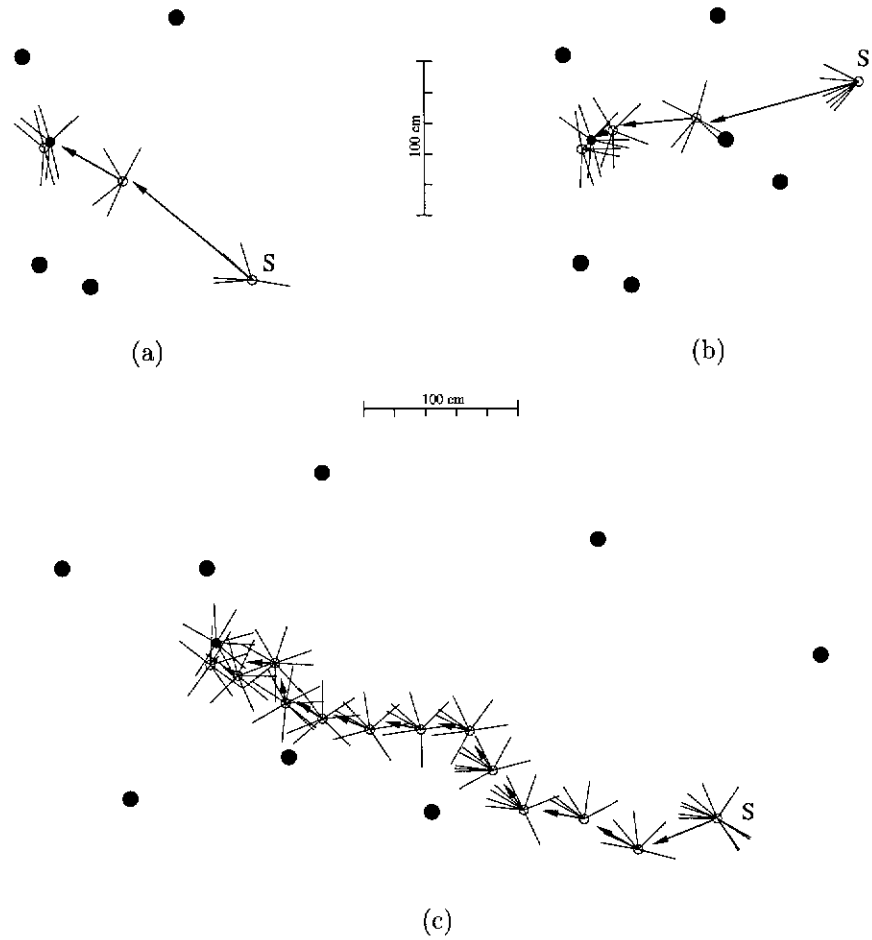


Figure 6.28: Homing Behaviour (real-world). Within increasingly more complex arrangements of landmarks, as shown, the mobile robot continues to exhibit successful homing.

6.6 Extending and Improving the Homing Algorithm

The basic homing algorithm can easily be extended to incorporate a richer source of visual homing cues. Additional landmark apparent-size cues, for example, can be utilised to overcome some of the limitations resulting from using only landmark bearing cues. Apparent-size cues, however, improve not only the robustness of the homing behaviour, but also reinforce its landmark avoidance characteristics. Visual occlusion is now also realistically incorporated into the simulations.

A snapshot is now represented by a list of bearings (θ_n) and angular sizes (α_n) and is given as:

$$S = [(\theta_1, \alpha_1), (\theta_2, \alpha_2), (\theta_3, \alpha_3), ..., (\theta_n, \alpha_n)] \quad \text{where} \quad \theta_i \dot{<} \theta_{i+1}$$

6.6.1 Computing The Homing Direction

Although bearing information alone is usually enough for successful homing, the incorporation of additional apparent-size information can further assist in this task. This involves simply adding further correctional vectors, correcting for the apparent size of landmarks. This is implemented in exactly the same way as before (§6.4.2.3, fig. 6.2), except the correctional vectors will be directed towards or away from the landmarks in order to better match their angular sizes (fig. 6.29). If a landmark in the current view appears larger than its corresponding landmark (as viewed from home), then the correctional vector directs movement away from the landmark in order to decrease its apparent size. Conversely, if the landmark appears too small, the correctional vector is directed towards the landmark in order to better match that observed from home. Again, the correctional vectors are weighted, but in this case by the differences in angular size between landmark pairs. This has the additionally useful effect of providing a (proportional) repulsive force, directing movement away from extraneously looming landmarks.

The final and improved homing vector is thus computed by adding the homing vector computed for apparent-size cues to the original homing vector for bearing cues.

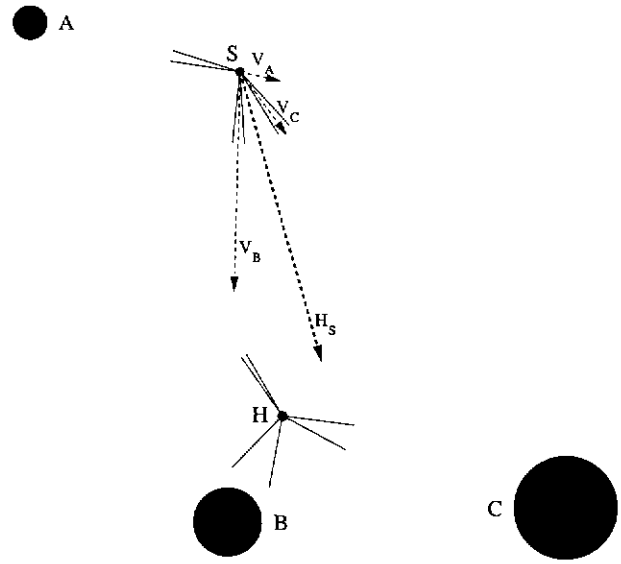
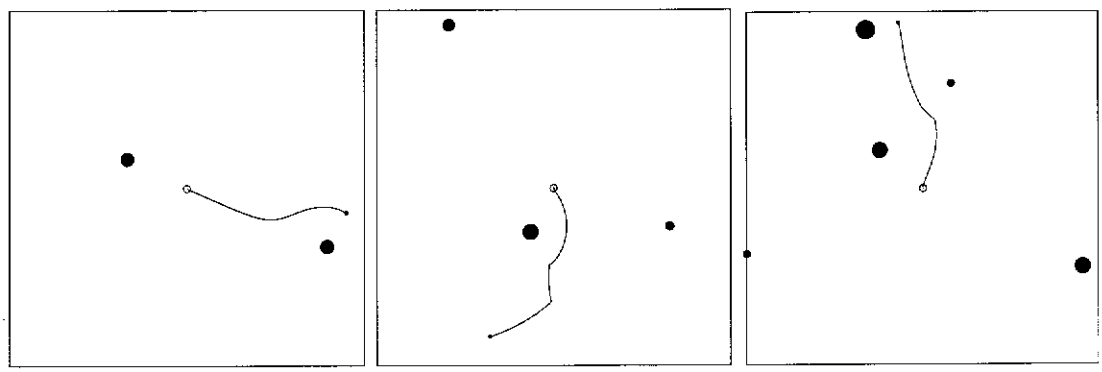


Figure 6.29: Computing Correctional Vectors (using landmark apparent-size information). Given landmark angular-size information, each correctional vector V_i now attempts to improve the perceived size of its landmark (A,B,C) to better match that observed from home H.

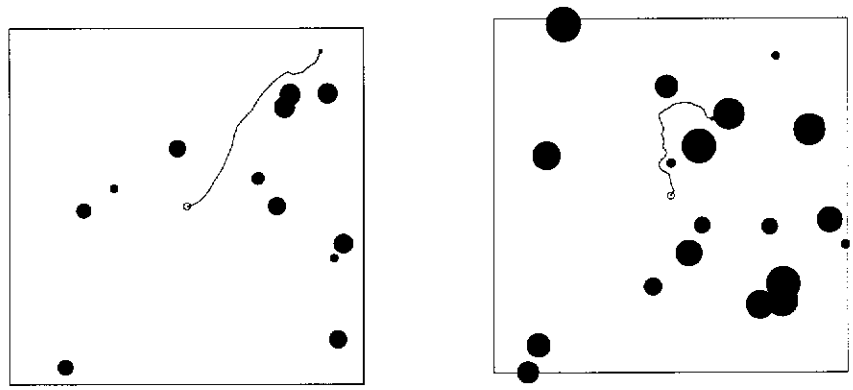
6.7 Results II

6.7.1 Simulation

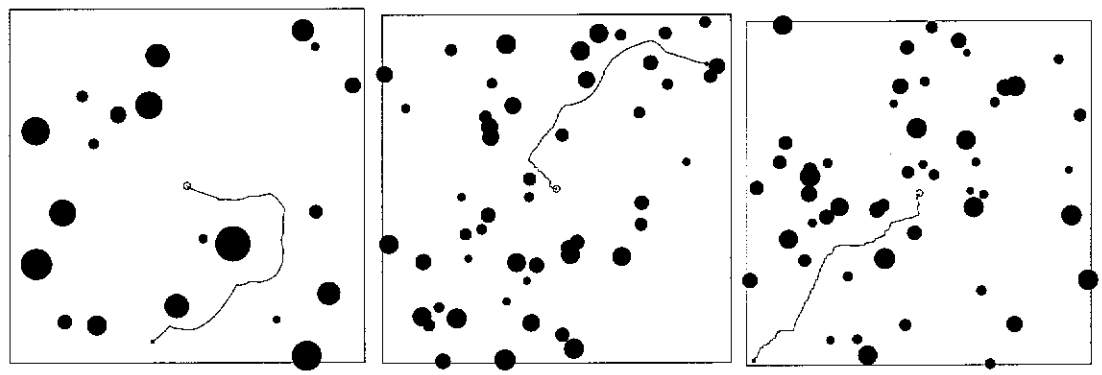
Figure 6.30 shows the simulated homing behaviour, utilising both landmark bearing and apparent-size cues, for various test arenas. Figure 6.30(a) shows how the homing behaviour performs in a simple arena. It is worth noting that this arena would cause problems if one were only using bearing information. In these situations where the landmarks and the target location lie along a line, ambiguity would arise if the additional apparent size cues were not utilised. Figures 6.30(b,c) not only show the landmark avoidance behaviour but also show the effect occlusion can have on the otherwise smooth homing path. Figures 6.30(d,e,f) show how the homing behaviour performs in increasingly more cluttered environments. Apparent size cues greatly assist in avoiding looming landmarks. Finally, figures 6.30(g,h) show the ability of the robot simulant to home even in a very densely packed arena. Despite the visual occlusion problems, the robot was able to home successfully in each case.



(a) h3 (b) h3 (c) h3



(d) h3 (e) h5



(f) h5 (g) h5 (h) h5

Figure 6.30: Homing Behaviour (simulation, h3,h5). Individual homing trajectories show the path taken by the robot simulant on its return journey, in increasingly cluttered environments. Using both landmark bearing and apparent-size cues, the robot is capable of successful homing despite visual occlusion.

6.7.1.1 Visual Occlusion Ambiguities

At this point, it should be noted that visual-homing ambiguities still arise. These are now mainly caused by the visual occlusion of landmarks. As the observer moves, closer landmarks in the foreground will naturally obscure other more distant ones within the homing environment. The partial occlusion of the visual field can thus make the panorama from two physically different locations appear very similar. This phenomenon, as expected, can mislead a visual homing robot into falsely interpreting the proximity or direction of home.

When two physically different locations appear practically identical there is effectively little, in the way of visual homing, that can be done to resolve the ambiguity and improve the situation. If an identical alternative location is homed-in on, there is essentially no way to recognise this fact. The very fact that a panorama is identical to that observed from home, for all intents and purposes, suggests a successful return. Fortunately, more than one 'global minimum' is very rare.

Local minima, on the other hand, are not so rare. Fortunately, these can be recognised as such by the significant differences in perceived panorama. If a false minimum can at least be detected there is hope for evading it, or escaping from it.

A good example of the homing problems that can occur is shown in figure 6.31. The vector field, identifies four local minima within which the homing robot may find itself trapped. In each case, the small pool of incorrectly computed homing vectors is a direct result of visual occlusion.

Consider first the bottom-most problem area, indicated in figure 6.31 by the boxed region. For ease of description, the landmarks within the arena are sequentially numbered from left to right and top to bottom. From the problem location, defined by the erroneous homing vectors, landmarks 1 and 2 are fully eclipsed by landmarks 3 and 4. From this vantage point only landmarks 3, 4, 5 and 6, are observed. Given that both landmarks 1 and 2 are missing from view, the correspondence algorithm (h3) determines the best pairing, between current (C) and home (H) snapshots, to be: $((C_3, H_1), (C_4, H_2), (C_5, H_3), (C_6, H_4))$. The key mistake here is the misinterpretation that treats the bottom two landmarks (5 and 6) as if they were the middle two (3 and 4). The computed homing direction is thus understandably misdirected away from the true position of home.

The same homing problem can also be seen to occur when close to a

landmark.[23] The other two problem areas, occurring very close to landmarks 3 and 4 (fig. 6.31), are essentially caused by the same general problem.[24] The visual occlusion of landmarks causes a misleading landmark pairing, which in turn provides an incorrect homing vector. However, in this case, the problem is caused by a single looming landmark. This makes corrective action much easier. A simple landmark circumnavigation behaviour, for example, has shown to be quite effective in resolving the impasse. The robot is effectively transported to the other side of the offending obstacle, from where successful visual homing can once again, safely resume.

In the previous problem area, however, the culprits of the occlusion are not so easily identified. No simple reactive behaviour can reliably extricate the robot from its predicament. A more systematic behaviour needs to be employed. This may either involve an intelligent modification of the landmark pairing or a more behavioural solution. There are many possibilities in this regard. One possible solution, for example, is the utilisation of some kind of systematic search behaviour akin to that observed in insects (§2.5.8). Although in this case, instead of being used to find home, the search would perhaps more intelligently be directed towards finding a more promising location from which to recommence visual homing.

6.7.1.2 Landmark-Based Homing versus Image-Based Homing

For a comparison of the homing performance offered by the proposed landmark-based approach, with that of image-based approaches, consider the two example scenarios depicted in figure 6.32.

The homing-vector fields resulting from the landmark-based (L) method indicate that homing is quite reliable from any part of the arena, irrespective of visual occlusion problems. The size of the catchment area is generally only dependent on the detectability of the same set of landmarks, visual occlusion notwithstanding, as those seen from home. Given a 'bounded universe'

[23] This is not surprising given the greater degree of occlusion a closer landmark can cause.

[24] A 'deflection behaviour' is already being utilised to avoid the stalemate that can occur directly behind a landmark when the bearing and apparent-size correctional vectors, are combined destructively. However, in this particular case, this has little bearing on the occlusionary problem being discussed. The deflection behaviour simply directs the homing vector perpendicularly to the offending landmark. This successfully avoids much of the problem associated with looming landmarks.

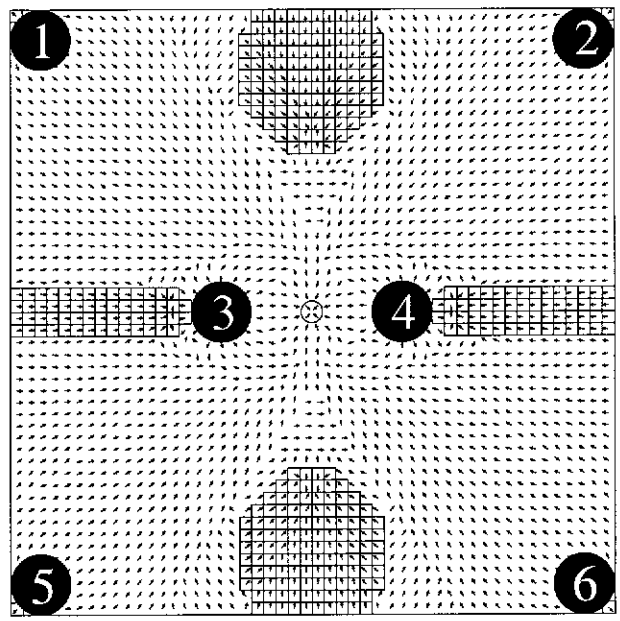


Figure 6.31: Homing Ambiguity Caused by Occlusion (simulation, h3). The occlusion of landmarks can lead to a misleading set of landmark correspondences, which in turn, as shown above, can create a 'local minimum' within which the robot may become trapped. Naturally, this occurs most often when landmarks are close, thus obscuring more of the visual field, but may also occur when landmarks are further away.

assumption (i.e. no significant perception horizon problem),[25] there exists essentially no position where the visual snapshots are so different as to render visual landmark-based homing unworkable.

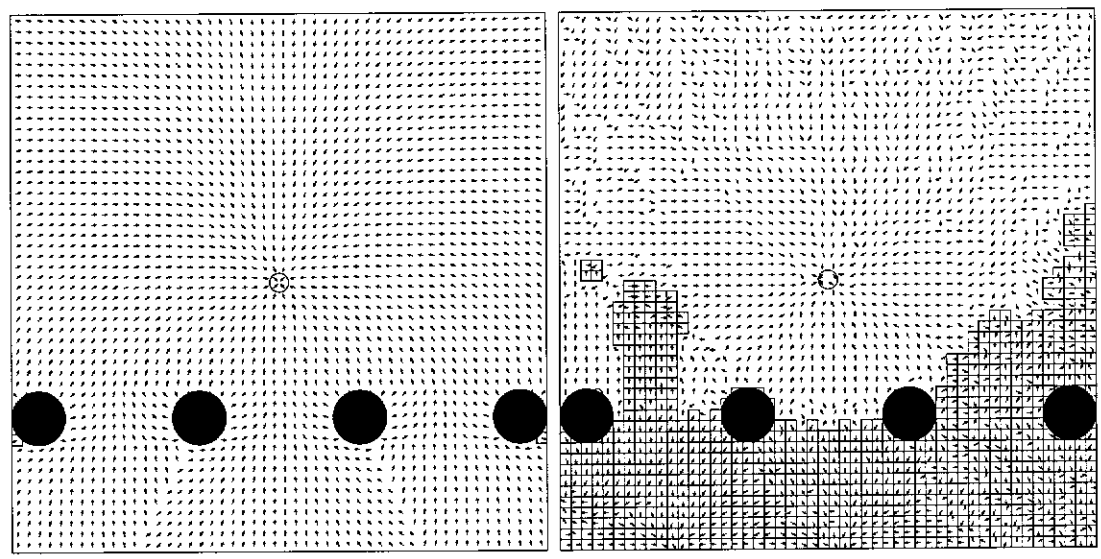
The image-based approach, used for comparative purposes in figure 6.32, is that proposed by Franz et al. (1997a) (see §2.7.1). The homing-vector fields resulting from this image-warping (W) method show its inherent limitations. The extent of the resulting catchment area is primarily restricted to the interior region, defined by the surrounding environment. Homing becomes unreliable when presented with a starting position that is outside of the space defined by the immediately surrounding landmarks at the home position. This result is not surprising given the basic equi-distant panorama and pixel adjacency assumptions of the approach.

For the purpose of figures 6.32(b,d), the 'warped space' was sampled with 12 directions of movement $(0°, 30°, 60°, .., 330°)$ and 15 displacement distances (0cm, 10cm, 20cm, 40cm, 80cm, 140cm, 200cm, 260cm, 320cm,.., 680cm). The panoramic image consisted of 360 pixels, one per degree of view. The average range of the initial panorama, as seen from home, was synthetically computed and used as the relative range value. Although unrealistic, this allowed, for the purposes of simulation, a much more confined search space. In the real-world, lack of environmental knowledge must be compensated for by increasing the search space, unfortunately however, at increased computational expense.

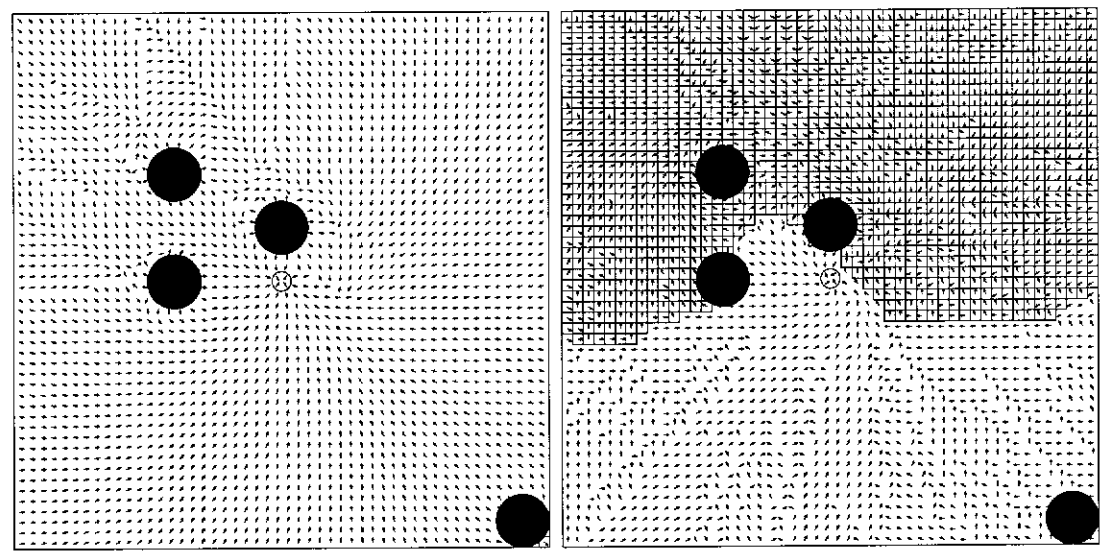
The asymmetry of the catchment area, observed in figure 6.32(b), is mainly due to the heterogeneous nature of the landmarks. By way of further assisting the image-warping approach the landmarks were given unique luminance values. Landmarks, however, remained homogeneous for the landmark-based approach.

A final comment on the computational cost of the image-warping method should also be made. Increasing the sampling rate, at which a matching snapshot is searched for in 'warped space', naturally increases homing reliability, up to a point. However, due to the combinatorial effect that increasing the size of the sampling set has on computational cost, this can quickly become prohibitive. Any substantial increase in image resolution also incurs a similar cost. Without access to parallel architecture, the image-warping approach may not be viable in terms of real-time performance, at least not without compromising homing performance. On an SGI Indy workstation, using the parameters described above,

[25] See §6.5.1.2 for an example of the effect a perception horizon can have on visual homing.



(a) L (b) W



(c) L (d) W

Figure 6.32: Landmark Based Homing versus Image Based Homing (simulation). The landmark based approach (L) and an image warping approach (W) to visual homing are compared by way of homing-vector fields and their consequent catchment areas. In the two example scenarios, the boxed homing vectors indicate positions from which homing is unsuccessful.

an image-based homing vector could be computed in approximately 10 seconds. When the panoramic image resolution is reduced to 90 pixels, a homing vector is computed in approximately 0.7 seconds.[26] By comparison, for the examples above, the landmark-based approach computed a homing vector in approximately 0.003 seconds, which represents a 230 fold improvement. The computational requirement of the landmark-based approach is also, essentially, unaffected by image resolution. Once the landmarks are extracted from an image, the image is no longer required.

6.7.2 Mobile robot

Figure 6.33 shows another example of the robot homing behaviour observed in our real-world experiments. In this case however, both landmark bearings and landmark apparent sizes are observed and used to visually 'home in' on home. Home is represented by the small solid circle in the centre. The open circles indicate successive snapshot positions with the directions of observed landmarks also shown. Again, the discrete nature of the homing path is a consequence of the way the robot must capture its panoramic images. The mobile robot successfully homes by incrementally improving its position, in discrete steps, until a close match between the current snapshot and memorised home snapshot is attained.

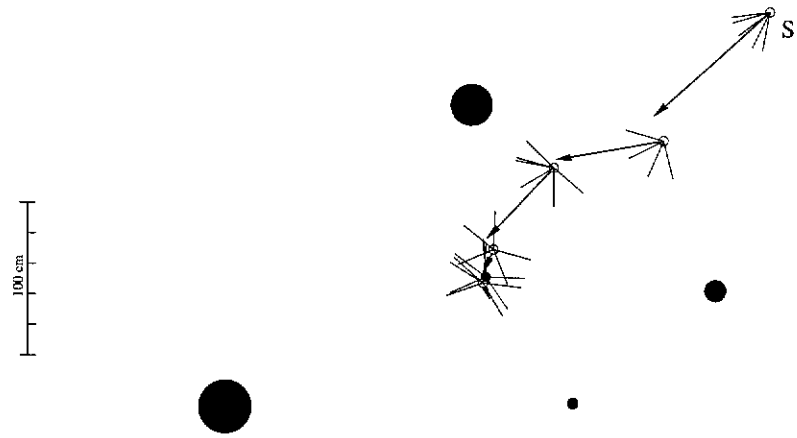


Figure 6.33: Homing Behaviour (real-world). An example robot homing trajectory using both landmark bearing and apparent-size cues is depicted.

[26]Several other alternatives exist to help reduce the computational expense of this image warping approach. These include utilising various heuristics to help prune the search or even simply using a compass sensor to remove the robot orientation dimension from the search space. See §B.1 for a more detailed analysis.

6.8 Conclusions and Discussion

This chapter has presented a very simple, intuitive, computationally cheap, and yet very robust (especially in terms of catchment area), qualitative landmark-based homing technique. The experiments detailed here, have shown that if even simple, homogeneous landmarks can be detected within an environment and an external frame of reference (i.e. compass) is available, then landmark-based homing can provide a very powerful method of real-time navigation.

Although the basic algorithm, utilising only bearing information, performs well and provides stable homing, it can easily be extended to incorporate a richer source of visual homing cues. The additional use of landmark apparent-size cues, for example, is shown to improve not only the robustness of the homing behaviour, but also reinforce landmark avoidance.

It has been shown that the landmark correspondence, required to determine a homing vector, need only be approximate. Of the efficient landmark correspondence methods tested,[27] method h3 provided the best pairing overall for the homing procedure. Further, method h3 was shown to provide a pairing that produced equal, if not better, homing performance than perfect landmark correspondence. As figure 6.14 shows, method h3 consistently produced the most accurate homing vector. However, as this method is based on maintaining the landmark topology within each snapshot, it is best only applied when the numbers of landmarks in each snapshot are equal. If the numbers of landmarks in the snapshots are large and unequal then method h3 may become computationally prohibitive. This is due to its computational complexity being $O(\max(n^2, n\binom{m}{n}))$, where n and m are the numbers of landmarks in each snapshot. In these cases, when method h3 is inappropriate, method h5 appears to provide the next best landmark pairing for the homing task. Method h5 is computationally safe irrespective of the numbers of landmarks in each snapshot ($O(mn)$). In a real homing scenario it would thus pay to choose a correspondence method, on the fly, depending on the numbers of landmarks involved. Another option of course is to base a strategy on utilising more than one method concurrently, by choosing and combining various estimates.

A one-to-one landmark correspondence (e.g. h3) method is also shown to be much more appropriate in a 'visually bounded' domain than a one-to-many

[27] See §6.4.2.2 for a detailed description of the various landmark correspondence methods evaluated.

correspondence (i.e. h7). This becomes increasingly apparent as the number of visual landmarks increases, and as the proximity of home increases. As the numbers of landmarks increase, the accuracy of the computed homing-vector, produced from a one-to-many correspondence, tends to decrease. The opposite is true for a one-to-one correspondence.

Given a set of landmark correspondences, a homing-vector is simply computed by summing all of the correctional vectors. However, there are several slightly different ways in which a correctional vector may be defined. Although the basic idea for each correctional vector is to attempt to correct the perceived bearing of its associated landmark, there is no unique direction that achieves this. The most obvious direction is perpendicular to the currently observed landmark, the rationale being that, given no range information, this will produce the fastest bearing correction. Alternatively, the correctional vector may be directed perpendicularly to the landmark bearing as it was observed from home. A third option is the bisection, or some other combination, of the two. Although the consequences of each approach are not immediately evident, the resulting homing behaviour is seen to be quite different. As is shown, if the correctional vector is perpendicular to the current bearing, an inherent landmark avoidance behaviour can be observed. However, incorporating home landmark-bearing information interferes with this landmark avoidance feature. Although a compromise homing-vector may produce a more efficient homing path, in terms of distance travelled, the overriding concerns of safety suggest the original scenario is best. A landmark-avoidance behaviour also reduces problems caused by visual occlusion. The further away one is from a landmark, the less its tendency to occlude other features and landmarks in the environment.

In its present form, the proposed algorithm is only proficient at homing with respect to a single constellation of landmarks, where there is no significant perception horizon problem. In other words, the homing region is assumed to be either 'visually bounded,' such as an indoor environment, or the robot is assumed to be close enough to home for the same general set of landmarks to be observable. Under these conditions landmark-based homing can be extremely reliable. In an open, 'visually unbounded,' outdoor environment, for example, it is assumed that the robot is capable of returning, via other means such as dead-reckoning, to an approximate home position from which visual-based homing is possible. This behaviour is also observed in insects. When far away from their nest, insects are known to return approximately, via alternative means, and then 'home in' more

accurately on their nest (§2.5.8).

Future work is required in this area to provide an elegant and reliable algorithm which will function well despite a significant perception horizon problem. Although all visual homing algorithms will eventually fail in this domain, the task is to provide a solution which will make the best of a difficult situation. If a certain homing distance can generally be guaranteed, perhaps defined by a percentage of the perceptual radius, then homing via multiple snapshots can provide a viable navigational compromise. Indeed, biological evidence for homing by multiple snapshots has recently been reported (Judd and Collett, 1998). In this way, it will be possible to reliably home within a practically infinite, homogeneous arena, such as a lunar-surface. Through the accommodation and memorisation of multiple snapshots the robot can conceivably increase its potential homing range without bound. Learning a topological arrangement of visual snapshots by exploration can thus be used to navigate successfully through an environment that is too large to be encompassed by a single snapshot. In this way a cognitive map or 'view graph' (Schölkopf and Mallot, 1995; Franz et al., 1998) can be constructed and used to navigate successfully beyond the perceptual horizon of a single snapshot. This can be extremely useful if a dead reckoning behaviour (e.g. odometric capability) is unavailable or inappropriate. A return trip can be made by retracing one's steps by 'homing in' on, in reverse order, the sequence of visual snapshots memorised on the outward journey. In this way one can use a generic homing behaviour as an intermediary navigational step in prescribing and achieving a larger objective. By giving the robot successive homing targets, one can generate a host of interesting global behaviours, by dividing a complex navigational problem into smaller homing subproblems. However, as the numbers of visual snapshots, required to be memorised, increase, so does the need for parsimonious representations.

If, under certain circumstances or in certain situations, visual homing is unreliable or inappropriate, then a hybrid system may be used. Using an associative homing technique, the agent may also be directed, via a simple dead-reckoning vector. In this way a displacement vector may be acted upon, to elude a perceived local minimum, before the next homing stage. On an outward journey, the need for an additional snapshot can easily be deduced by monitoring the return vector, computed via the homing algorithm. When this does not coincide with that expected for a successful return then a new snapshot must be memorised to maintain a traversable chain of homing locations. However, if the need for a

new snapshot is almost immediate upon leaving the last snapshot location, such as may be observed in relation to a perception horizon problem (e.g. fig. 6.21(a)), then an associated homing vector may be assigned.

Another avenue for improvement is in making the homing algorithm more robust to occlusionary ambiguities. As discussed in section 6.7.1.1, the visual occlusion of landmarks can cause a misleading landmark pairing to be adopted, which in turn can lead the robot into what is effectively local minimum. To extricate itself from a false homing location, the robot must either search for a more promising visual homing starting-point, or take corrective measures in terms of the landmark correspondence pairing.

The main advantages of landmark-based homing over image-based homing are the larger size of the resulting catchment area, and the economy of snapshot representation and computational cost. As we have shown, with landmark-based homing there is essentially no position in which the discrepancy between current and home views is so bad as to render homing untenable. This is not however, the case with image-based homing. Image-based homing is usually, only reliable within an area defined by the immediately surrounding environment. Within a constellation of landmarks this generally, limits the catchment area to the open area defined by the closest landmarks. The key difficulty in image-based homing is dealing with the visual panorama in a holistic and yet still accurate way. As an observer moves, objects at different distances and bearings will appear to move by different amounts. Further, objects at different distances will also move in front of and behind others, changing their relative position or possibly obscuring them. Therefore, the degree with which the dynamics of the actual visual panorama fits the expected dynamics determines its basis for success. The key reason for the success of landmark-based homing is the absence of any range or image adjacency or landmark topology assumptions. The landmark-based homing method proposed is capable of very reliable homing from anywhere within a single, finite constellation of landmarks, by utilising both the global characteristics of the cluster and the triangulation of observed landmark bearings.

Both image-based and landmark-based methods have their own set of assumptions, advantages and disadvantages. Depending on the current environment and the specific homing requirements at the time, one method may be much more suitable than another. For an autonomous robot navigating in unknown domains, it seems prudent to have more than one method of performing a given task at its disposal. In this way, a more robust overall behaviour can be

produced by tuning the use of particular methods to appropriate circumstances. Although the concurrent application of multiple methods may be considered overkill in most cases, the possible advantages of incorporating alternate strategies should not be overlooked.

The landmark-based approach to visual homing can, however, be easily extended into the three-dimensional domain. The bearings of visual landmarks simply incur an additional elevation dimension. The computation of a correctional vector remains essentially unchanged, but for the fact that the vector is now orientated in three-dimensional space. The 3D correctional vectors would again simply be summed to produce a 3D homing vector. The landmark-based approach would also be little affected in terms of homing performance and computational cost, the same cannot be said so easily for image-based approaches.

Clearly, the methods and algorithms proposed in this chapter are primarily concerned with providing solutions for autonomous robot navigation. Although inspired by findings in insect ethology, specifically visual homing behaviour, no direct insights into insect 'cognition' are being claimed. There are obvious difficulties in determining exactly how an insect performs a task. An insect cannot be opened and read like a book. One must try to deduce function from observable inputs and outputs. From ethological evidence one can make basic deductions and draw broad conclusions. Exact details, however, in the quest to fully understand the 'thought' processes which drive an insect, remain elusive. Despite this, insect ethology suggests that insects perceive and interact with their environment in a minimalist, yet successful fashion. Inspired by this philosophy, one can strive to imitate the successful behavioural strategies in similarly simple, elegant, and robust ways. It is this ethos which can be of prime significance in solving some of the difficult problems of autonomy in artificial agents. Insects prove by example that the underlying principle of simplicity and elegance in behavioural mechanisms is quite sound. The complexities and richness of behaviour need not necessarily be derived from complicated mechanisms.

Chapter 7

Conclusions

7.1 Summary

The main aim of this thesis has been to show that the guiding principles of autonomy and navigation as observed in the natural world, specifically through the study of insects, are of value in solving some of the real-time, real-world issues of autonomy and navigation in mobile robots.

In chapter 4, it has been shown how very simple motion cues and visual behaviours, inspired by the visual guidance of flying insects, can be used profitably to provide a mobile robot with the ability to simply and efficiently perceive, interact and move within real environments, such as in this case, in corridor-type environments.

Chapter 5 showed how the reactive robot behaviour could be seamlessly augmented to allow a 'route recognition' and 'route discrimination' ability. By observing landmark features en route, a previously encountered corridor can be recognised as such, and thus be used to help direct a more global goal-oriented activity.

Chapter 6 presented a very simple, intuitive, computationally cheap, and yet robust, qualitative landmark-based homing technique. This was similarly inspired by the visual homing behaviour of bees and ants. It was shown that if even simple, homogeneous landmarks can be detected within an environment and an external frame of reference is available (e.g. a compass), then landmark-based homing can provide a very powerful method of real-time navigation. Although the basic algorithm, utilising only bearing information, performs well and provides stable homing, it can easily be extended to incorporate a richer source of visual homing

cues. The additional use of landmark apparent-size cues, was shown to improve not only the robustness of the homing behaviour, but also reinforce landmark avoidance. The algorithm can also be easily extended into the third dimension.

7.2 Future Directions

From this body of research, one main avenue of future work and improvement has to do with the experimental environments and overall robustness. Although the experiments were conducted in the real-world using a real robot, they were still quite contrived in nature. As outlined in the previous chapters, this was mainly due to the limitations imposed by both physical and practical characteristics of the robot and project, as well as the desire to maintain experimental flexibility and control. However, the testing of a completely autonomous system in totally unprepared, realistic environments is a necessary step forward.

As with all of the experiments described in this thesis, the reliance upon visual stimuli is paramount. The inherent 'real-world' limitations of relying upon purely visual stimuli, as a source of navigational information, are obvious. As the 'quality' of the visual stimuli and perception decreases so too does behavioural competency. To compensate for this environmentally induced 'incompetency' it will become necessary to either alter behaviour to actively improve the richness of perception (and quality of interpretation), or more likely, increase reliance upon the perception and interpretation of alternative stimuli, and thus help drive appropriate behaviour. To this end, more work needs to be conducted to successfully fuse a multitude of extra-sensory behaviours to produce more robust interaction with the real-world, in the face of a paucity of quality stimuli.

Fortunately, there is no shortage of examples of how the world could conceivably be perceived: there exists no richer source of sensory and perceptual strategies than that observable in the natural world. Within such a diverse global environment, animals have evolved a miriad of sensory techniques with which to efficiently and successfully perceive, interpret and survive a hostile world.

More specifically, the situation as it stands at the end of this project, could be improved in several ways, in terms of both hardware and software, as well as experimental improvements. In terms of hardware there is ample room for improvement with regard to the physical characteristics of the 'autonomous' robot; such as, removing the restrictive umbilical, improving the quality and quantity of the visual sensing (eg. incorporating a panoramic sensor), the use

of additional non-visual sensors, and general mechanical improvements aimed at improving the physical robustness of the platform. The use of additional sensors would naturally also imply the use of additional sensory behaviours. A necessary next step will also be the testing of the system within more realistic (relatively unprepared) and natural environments.

Unfortunately, due mainly to the practical restrictions on experimental work space and limited umbilical cord length, it was not possible to conduct large-scale experiments as part of this investigation. Consequently, various behaviours were tested and observed in isolation. Further large scale experimentation would be very useful in observing the interaction and interplay of all the implemented behaviours at the one time. With all the behaviours activated at once a more 'holistic behaviour' could be observed and studied. The integration of behaviours is a challenging and interesting problem and would help bridge the gap between a set of low-level navigational tools and the solution of specific navigational problems. All of the behaviours presented within this thesis could conceivably be combined to solve a single large-scale navigation problem; for example, a homing problem requiring the robot to visit a sequence of intermediate 'homes' en route, through a sequence of corridor-type environments, to a final destination. Such an experiment would provide valuable information on how well the various behaviours work and 'gel' together to help achieve a larger objective.

In the visual homing work, two problem areas were identified as requiring further study. The first problem involved the visual homing ambiguity arising from a perception horizon. If the set of perceived landmarks changes significantly from location to location then reliable visual homing is also affected. The second problem also involved homing ambiguity, but in this case it was caused by visual occlusion. Visual occlusion may cause an incorrect landmark correspondence to be established, which in turn can lead to an incorrect homing vector and thus a 'local minimum'. The problem posed by visual occlusion is perhaps the easier of the two to solve, but both problem areas provide clear avenues for investigation and overall improvement in terms of the visual homing algorithm.

As previously mentioned, far broader future directions include the additional use of other non-visual cues and sensory behaviours. Visual guidance behaviours help solve one small piece of the autonomy puzzle. An ensemble of strategies based on a multitude of sensory sources can and should be employed to deal efficiently and robustly with the uncertainties of the real world. Sensory fusion, through distributed control, can provide a useful way of compensating for the inherently

imperfect nature of sensors and perception, and thus provide for a much more robust system. This will hopefully lead to the constructive combination of many behaviours upon which higher-level behaviours can be constructed.

The desire to cut the umbilical and thrust the robot into the real real-world, is always present. Hopefully with a 'critical mass' of appropriately combined 'sensory behaviours', a truly autonomously functioning system can be built that will survive in the real world for extended periods, and from which further higher-level behavioural advances can then be made.

Appendix A

Further Corridor Guidance Results

This appendix presents a more comprehensive set of real-world corridor guidance results. This will give a better impression of the typical robot behaviour experienced within the context of the experiments conducted chapter 4. Apart from showing the 'typical' nature of the visual behaviours, this section will also present some of the not so perfect results and thus indicate some of the problems involved.

Figures A.1–A.4 show a variety of examples of the robot using the centring behaviour to safely navigate down various straight sections of corridor. Note that variations in robot behaviour are also partly a result of variations in target speed, start position, and corridor specifics. Figures A.5–A.9 show the same centring behaviour but in a variety of differently shaped corridor.

Figure A.10 shows several examples of how the spinning behaviour can be used to turn a sharp corner. As suggested, this was probably the most unpredictable behaviour of all. Due to the meandering nature of the robot, and the fact that spinning is triggered by frontal range, the resulting behaviour varied greatly. There was also the problem of determining in which direction the robot should spin. As shown, this was not perfectly determined. Thus, on occasion, the robot would fail to turn in the appropriate direction.

Figure A.11 shows several examples of the wall following behaviour. The effects of both wall-following and non-wall-following are examined and compared.

Figure A.12 shows several examples of how the robot can regulate its forward speed according to tunnel width by maintaining a constant (lateral) apparent

motion speed.

Finally, figure A.13 and table A.1 present the results of an earlier version of the visual odometry experiments.

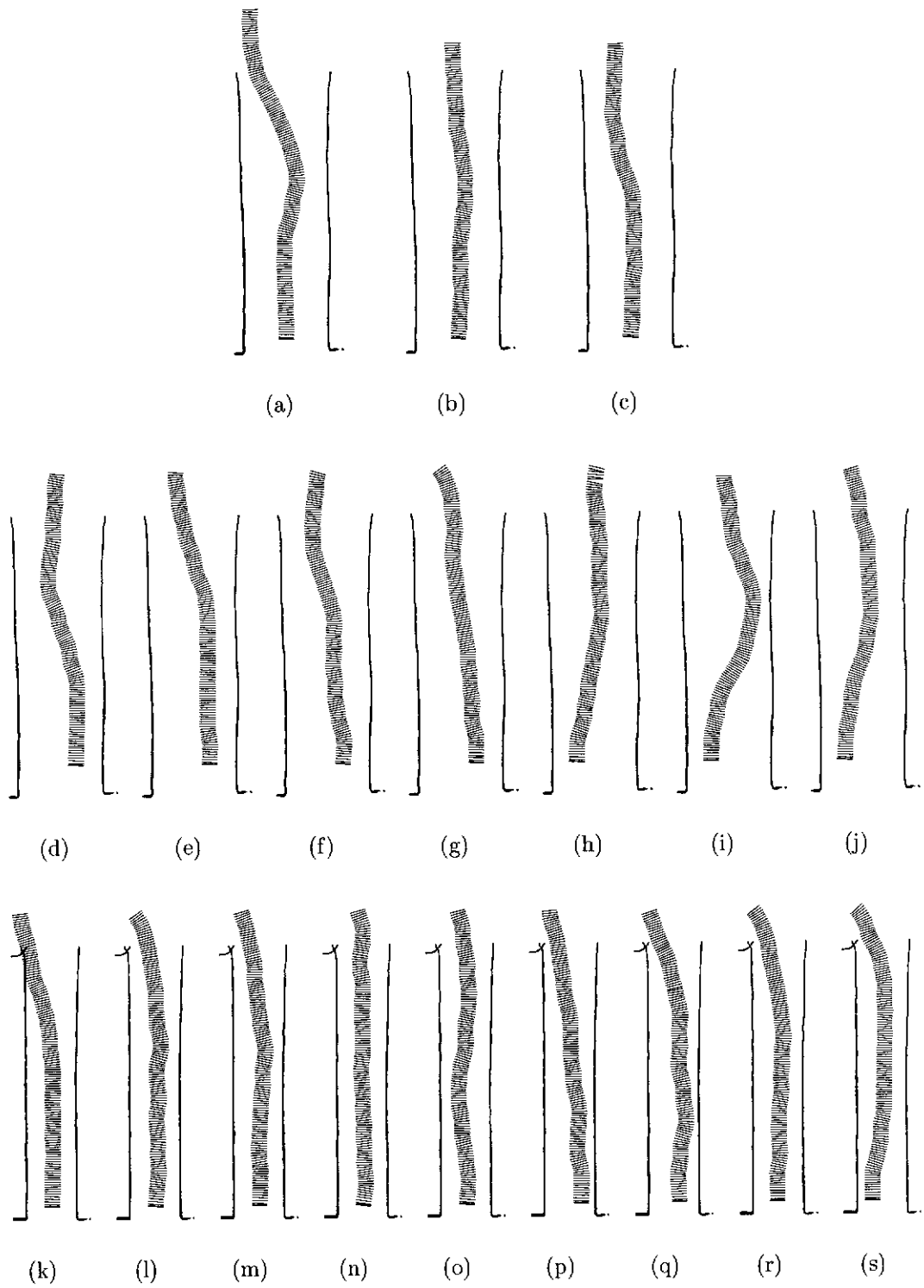


Figure A.1: Early Centring Behaviour within a Straight Corridor (real world)

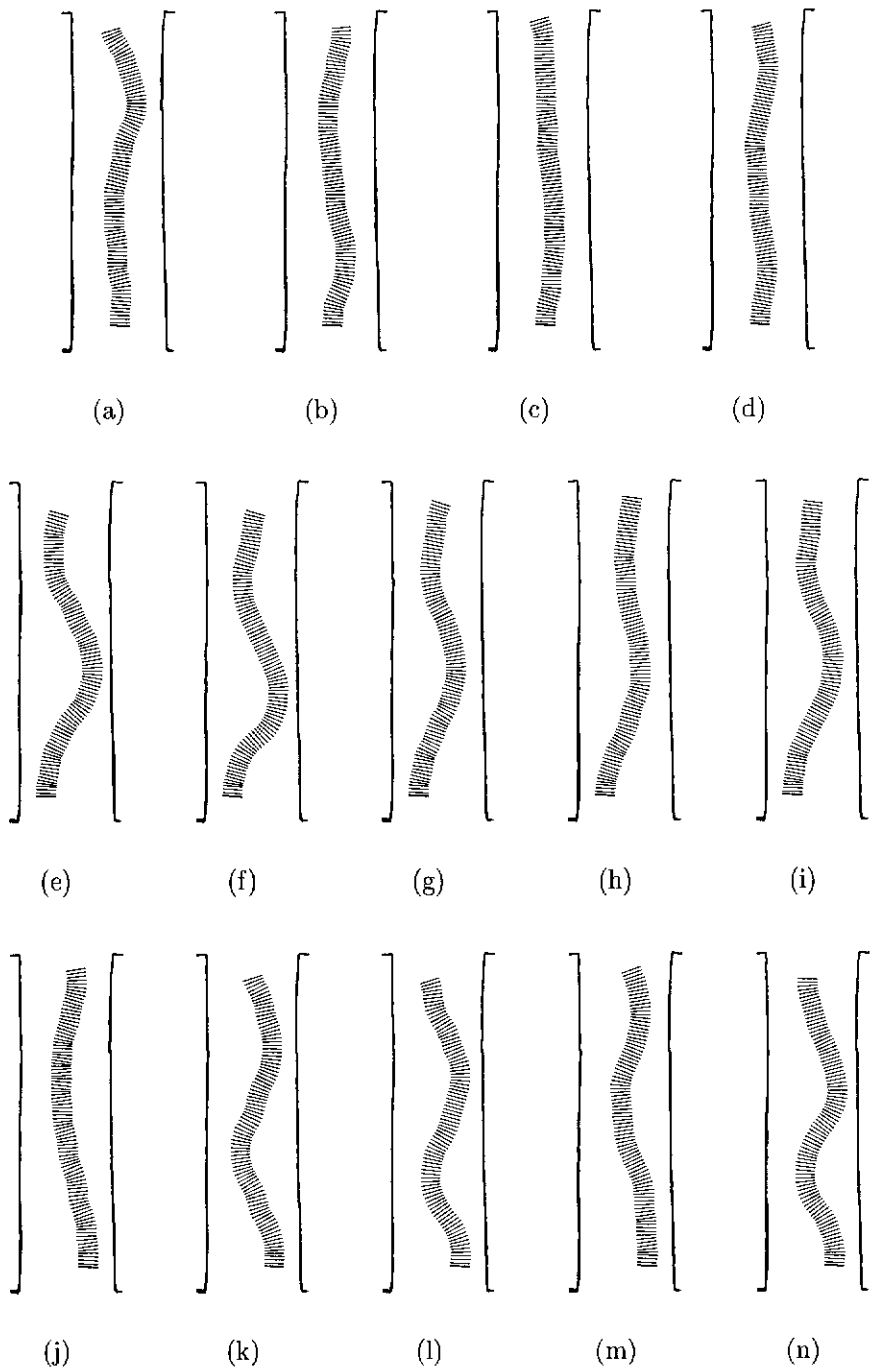


Figure A.2: Centring Behaviour within a Straight Corridor (real world)

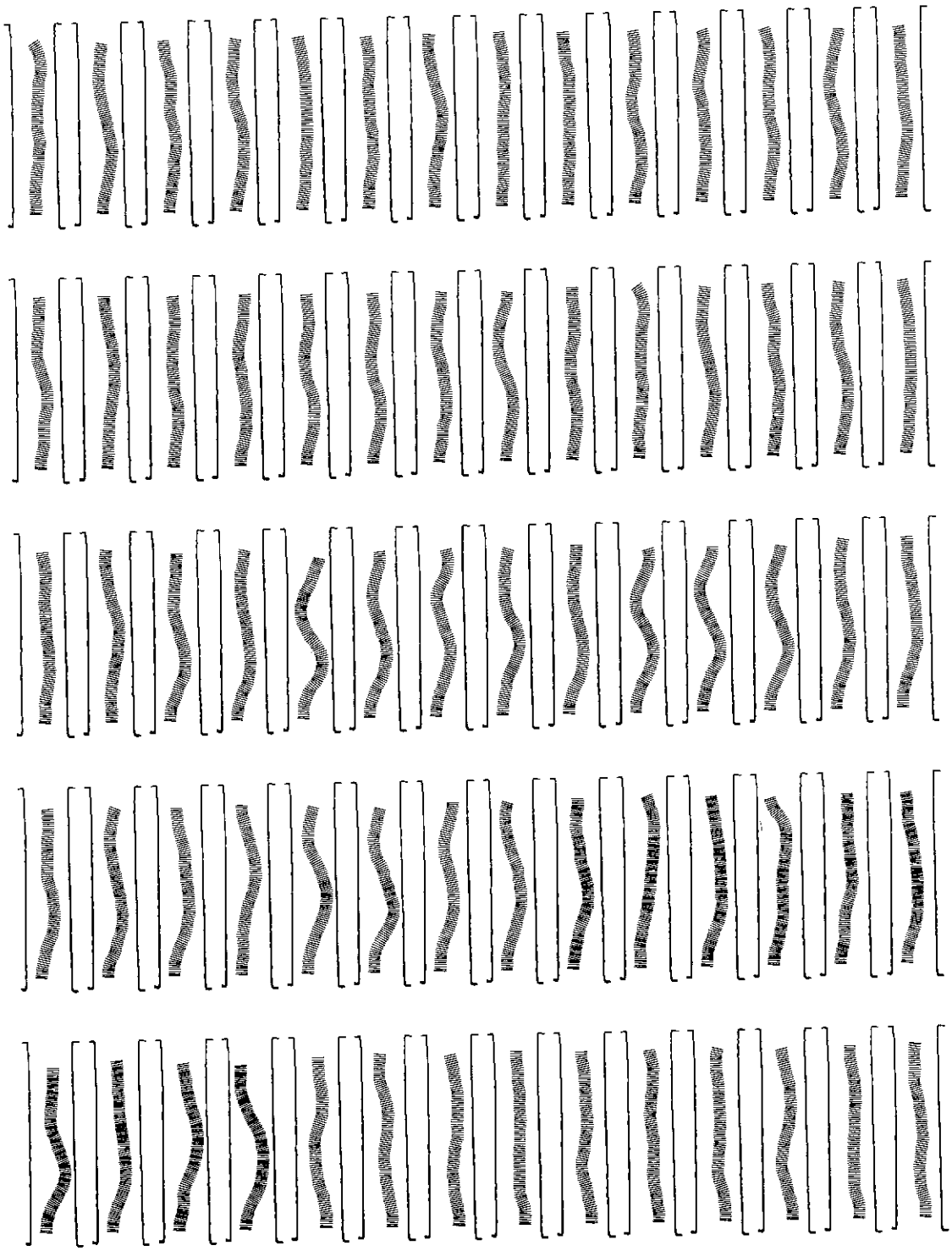


Figure A.3: Centring Behaviour within a Straight Corridor (real world)

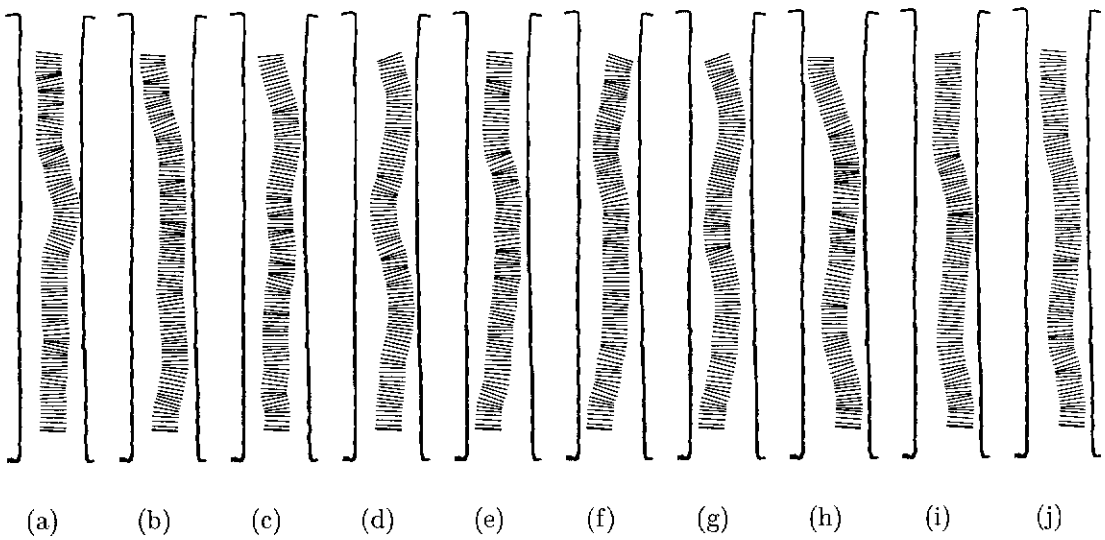


Figure A.4: Centring Behaviour within a Narrow Corridor (real world)

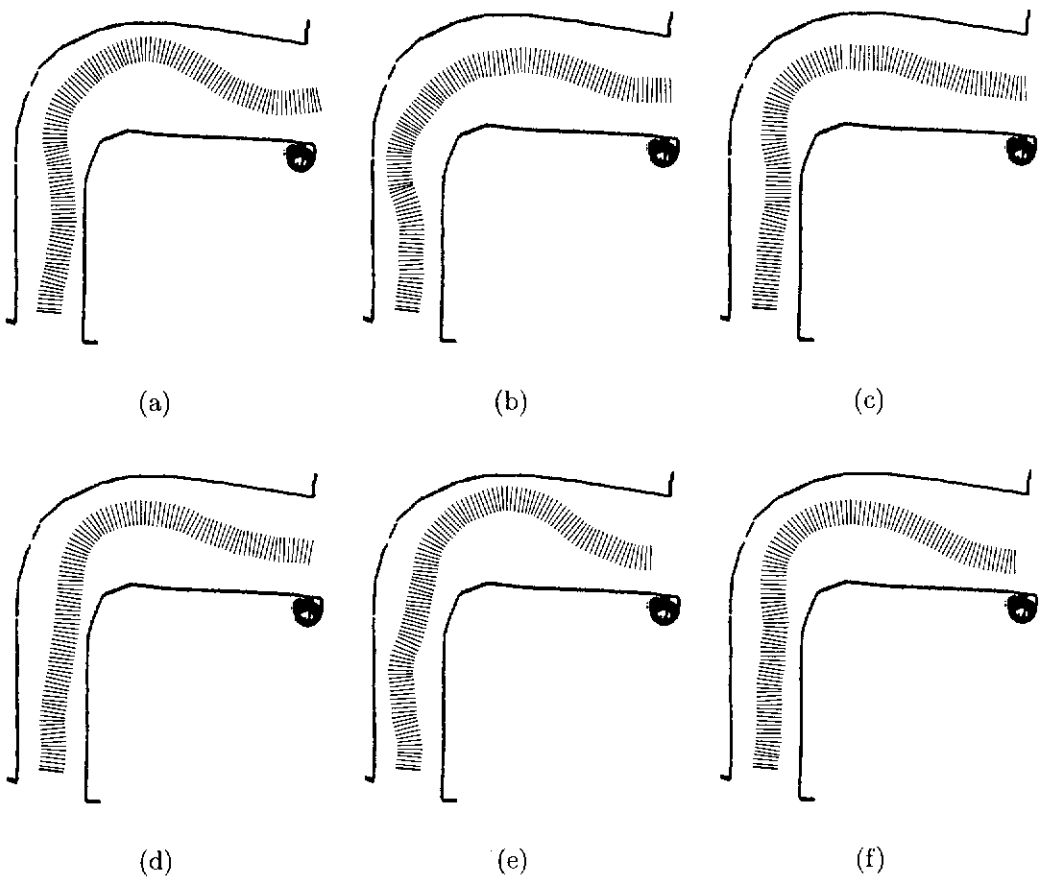


Figure A.5: Centring Behaviour within an L-Shaped Corridor (real world)

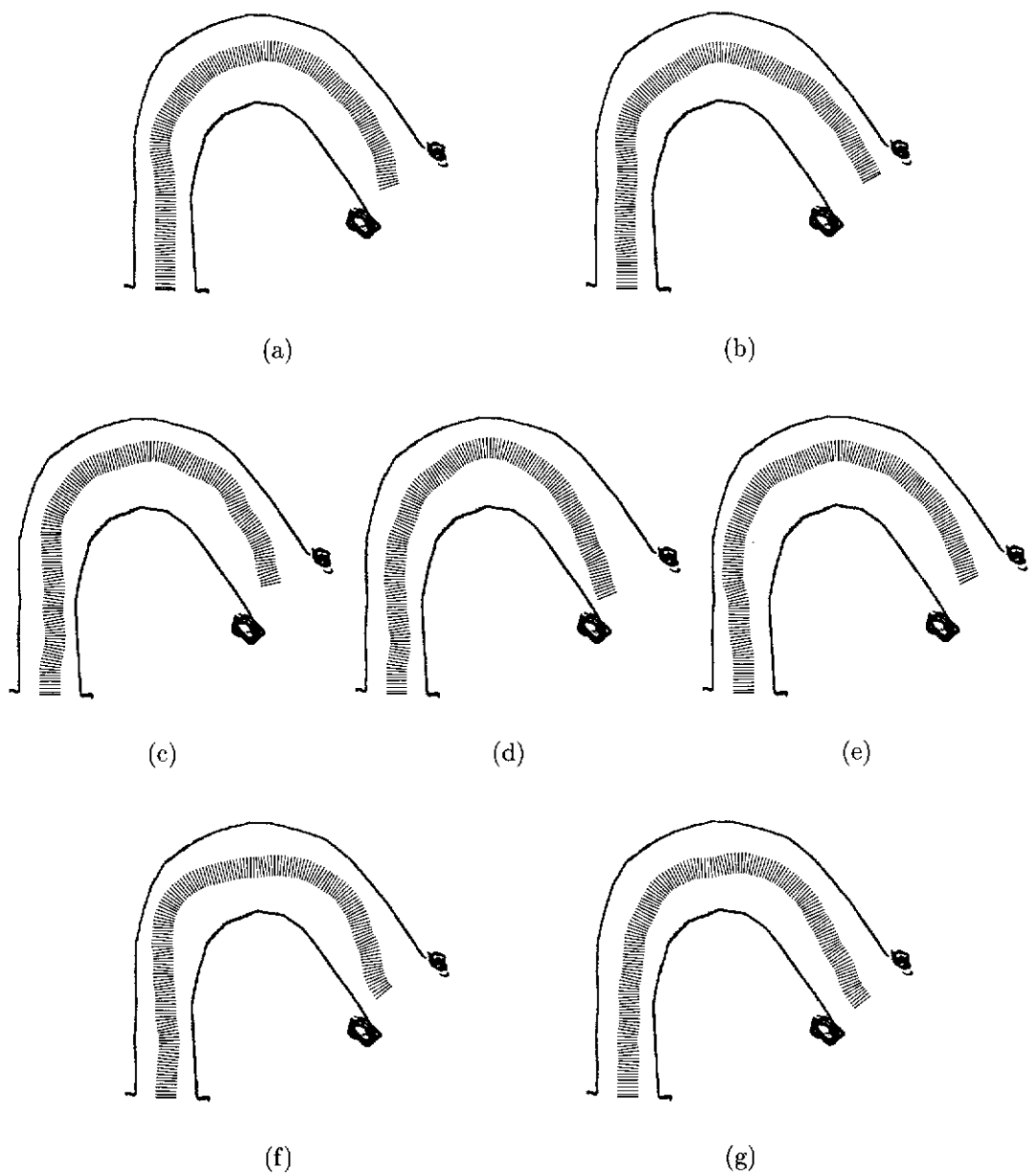


Figure A.6: Centring Behaviour within a Curved Corridor (real world)

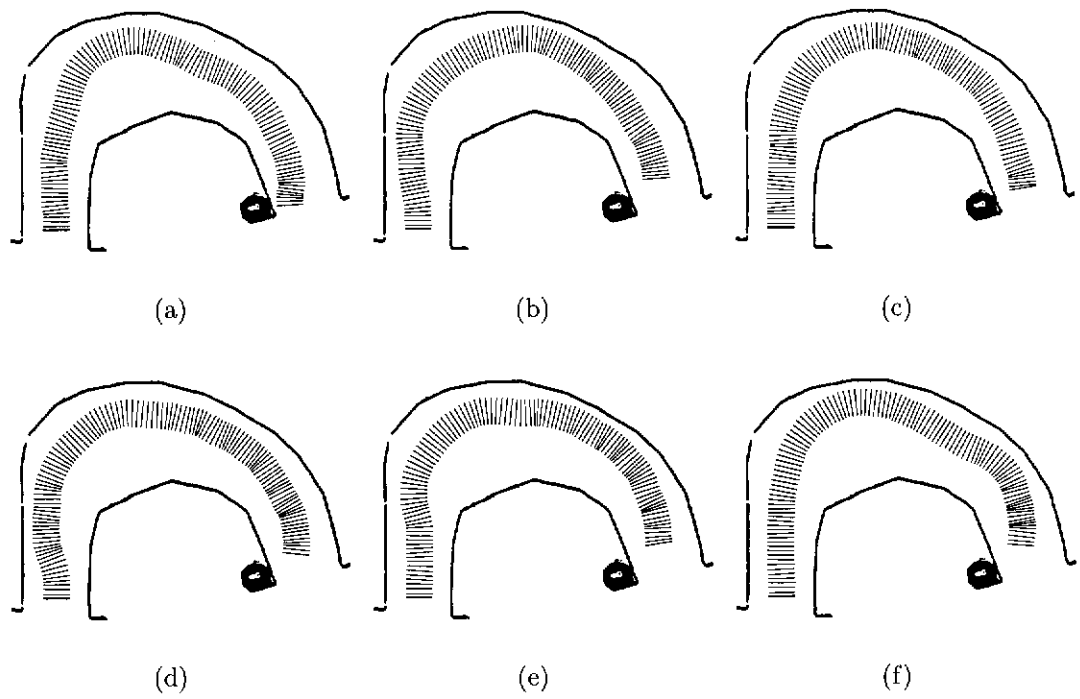


Figure A.7: Centring Behaviour within a Curved Corridor (real world)

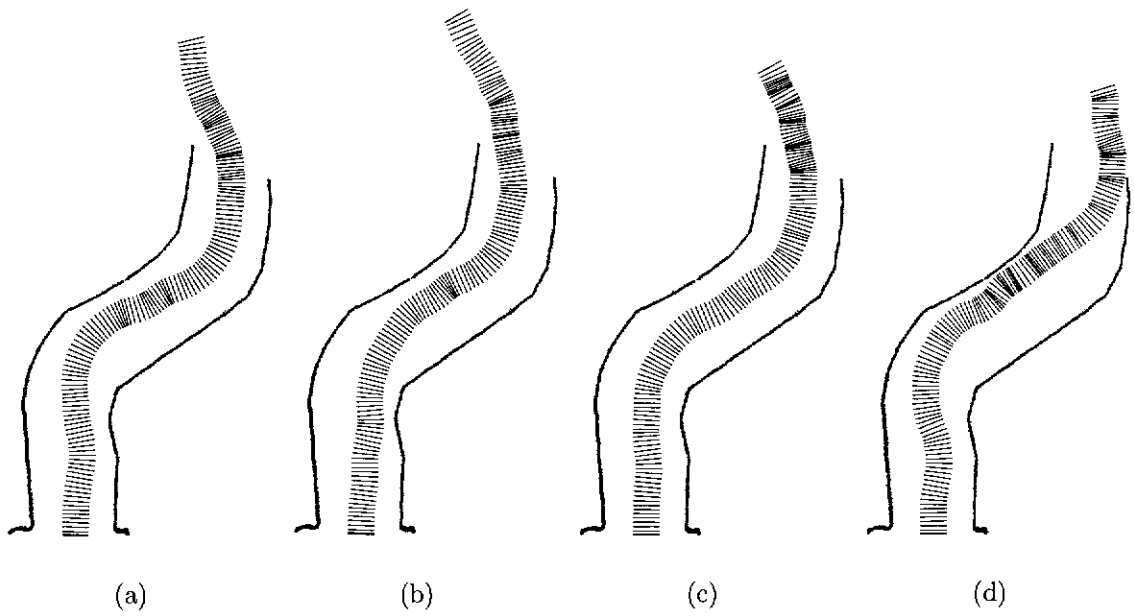


Figure A.8: Centring Behaviour within an S-Shaped Corridor (real world)

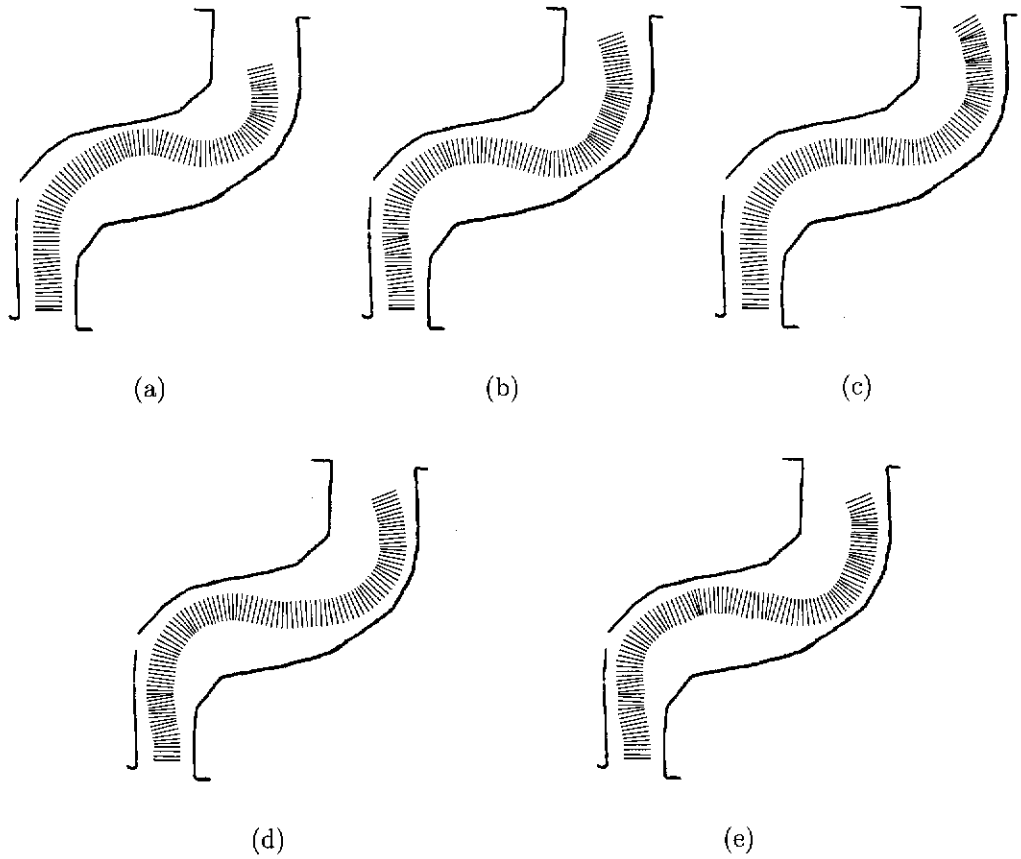


Figure A.9: Centring Behaviour within an S-Shaped Corridor (real world)

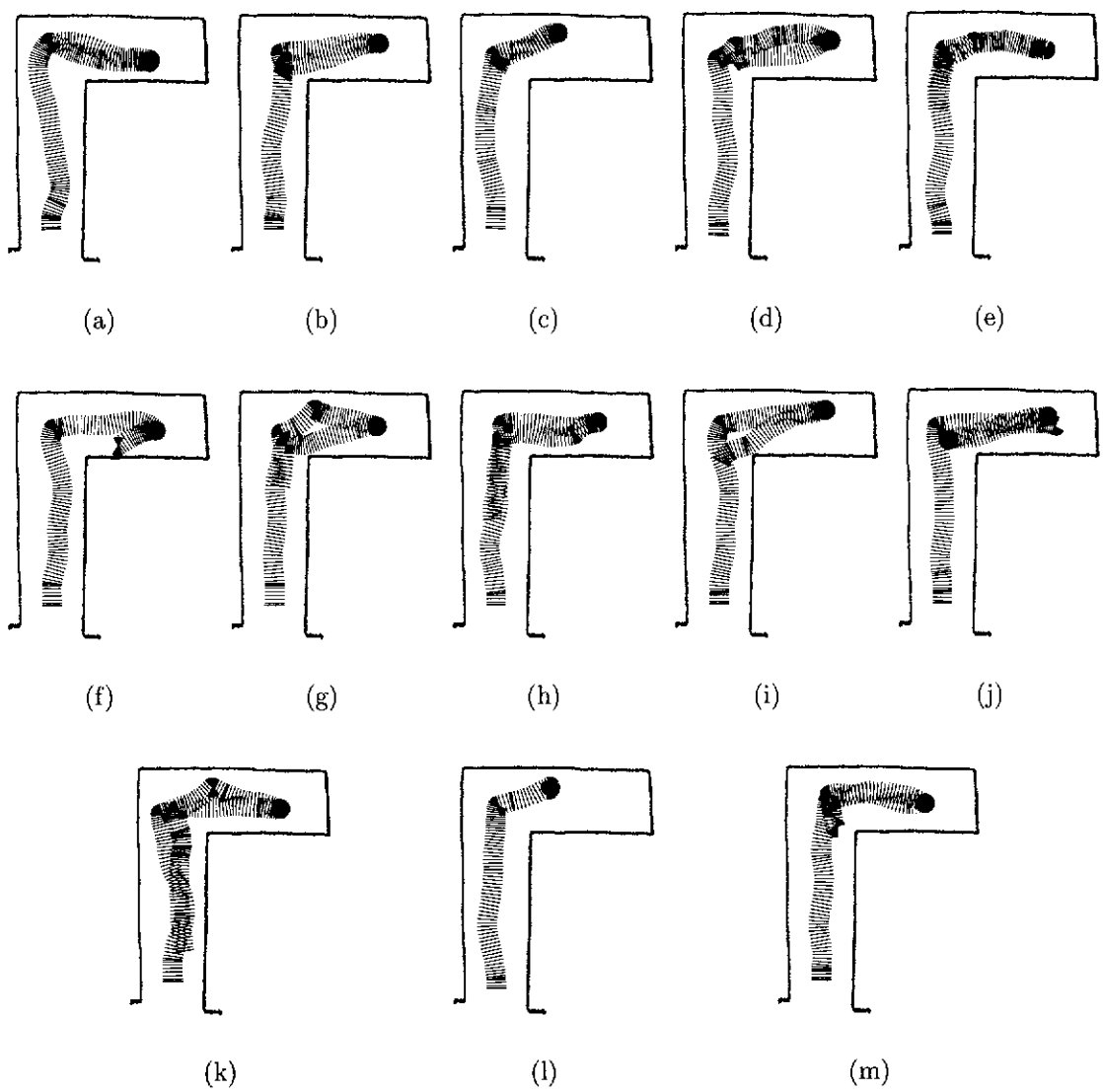


Figure A.10: Spinning to Aid Sharp Cornering (real-world). The problem of correctly determining in which direction to spin is illustrated in subfigures (j) and (m). Note, however, that accurate robot odometry is most seriously affected by spinning. This should be kept in mind when viewing these figures as the spinning, stopping and starting behaviour can cause a noticeable mismatch between real paths and displayed paths.

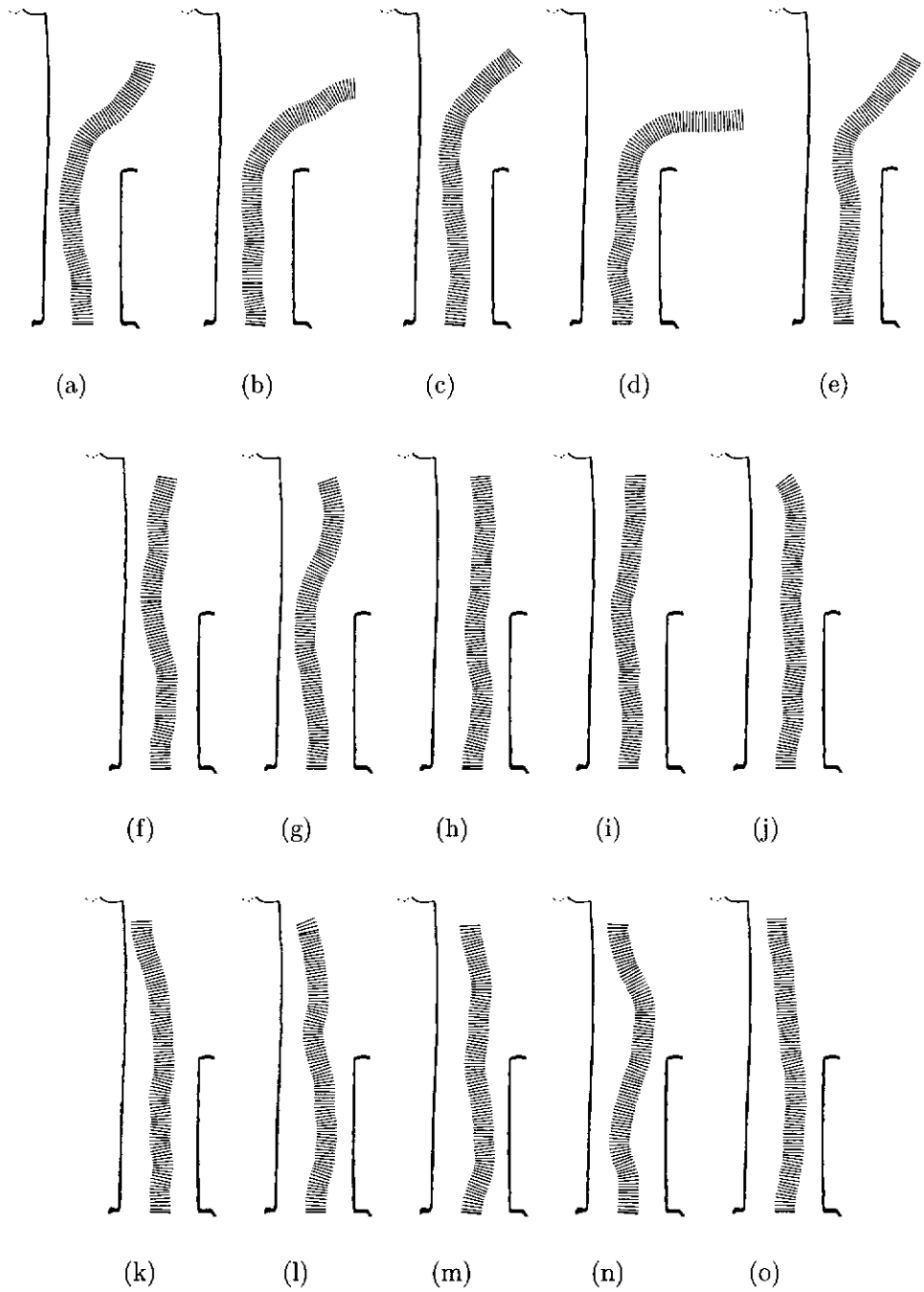


Figure A.11: Wall-Following Behaviour (real world). (a–e) Without a wall following behaviour the robot naturally heads into open space, once clear of the right wall, in an attempt to centre its position. (f–o) With a wall following behaviour the robot successfully maintains its straight ahead trajectory, ignoring the lateral absence of a wall.

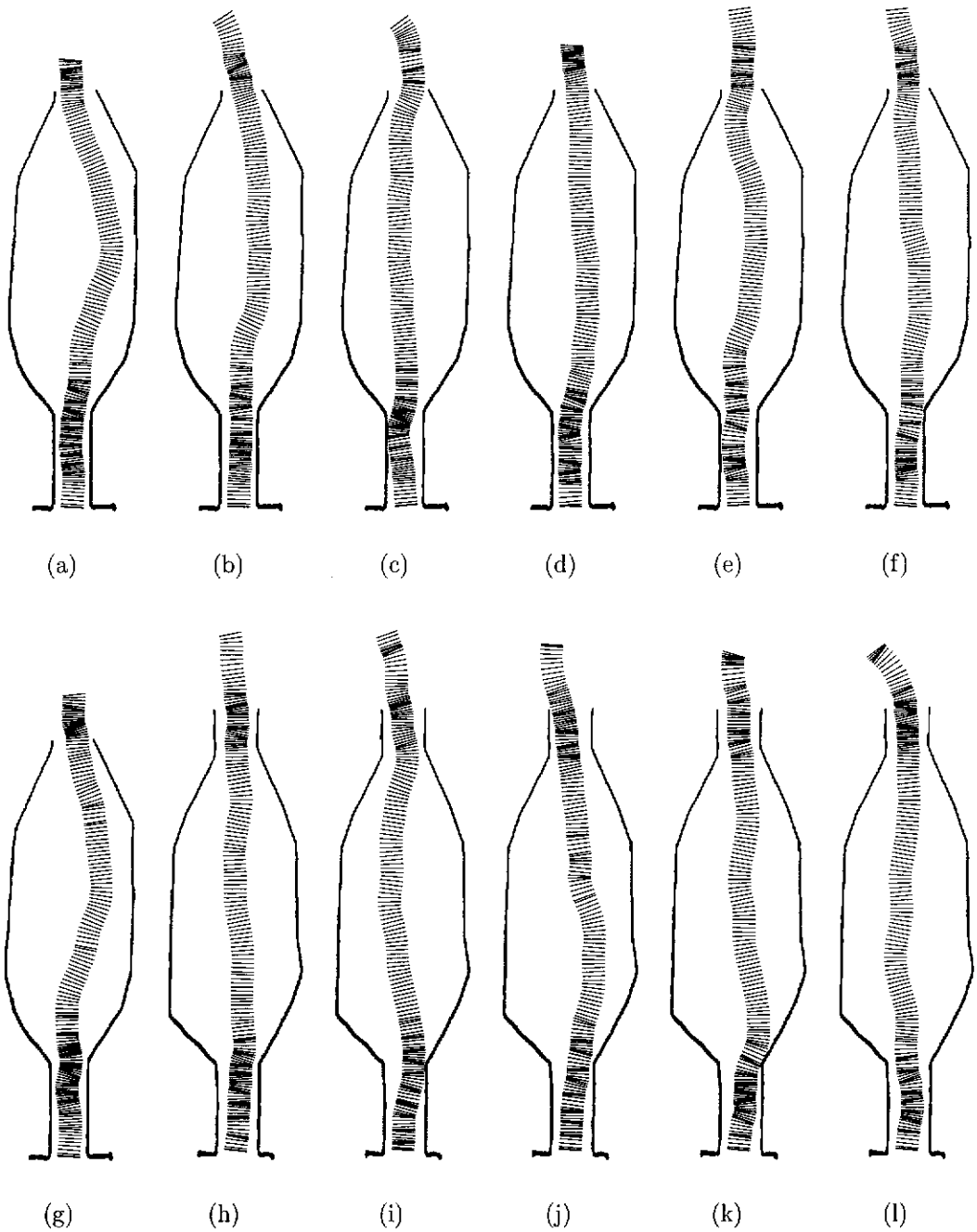


Figure A.12: Regulating Forward Speed (real world). Striving to hold constant the (maximum) speed of laterally perceived motion has the useful effect of regulating robot speed according to tunnel width.

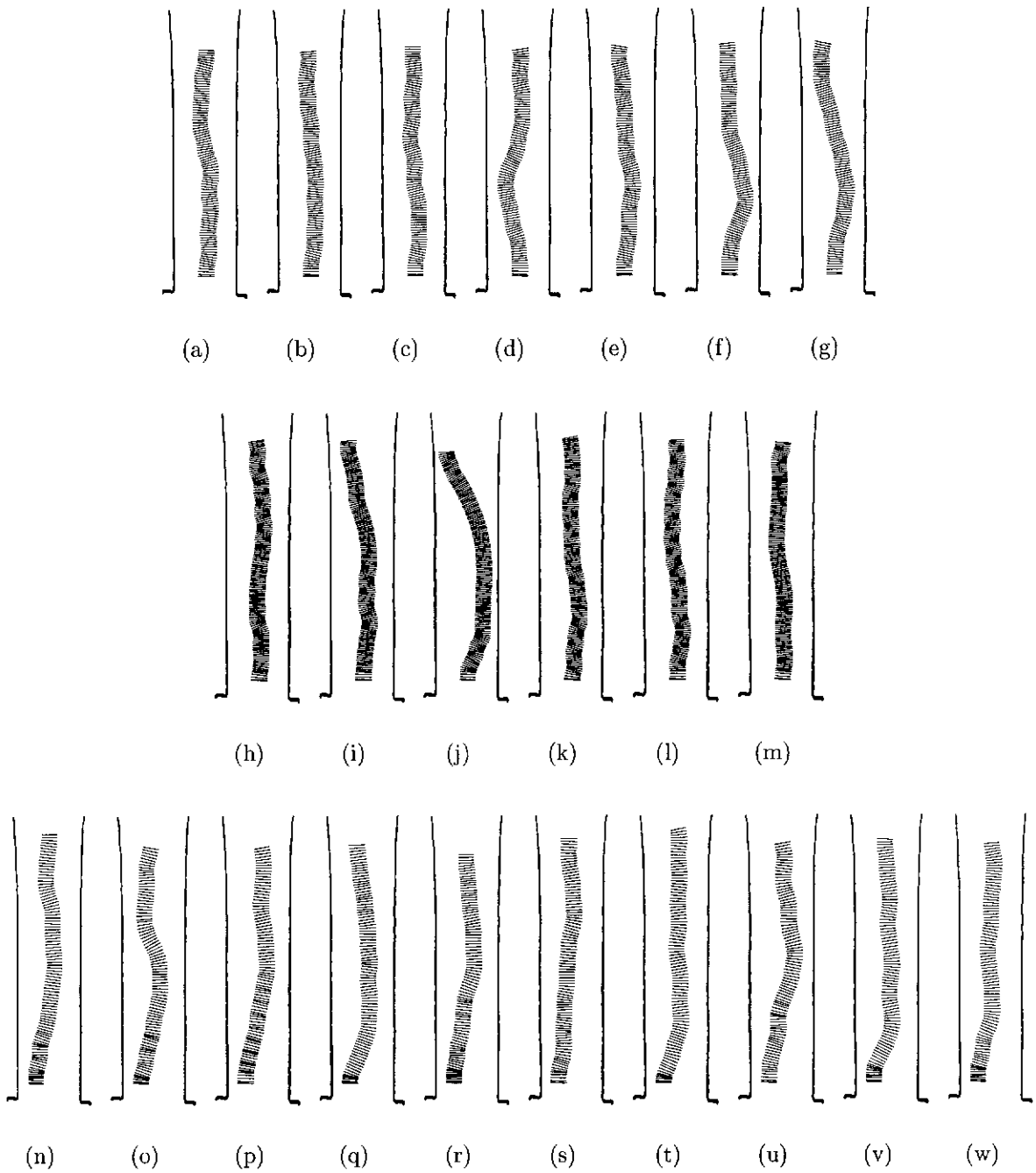


Figure A.13: Early Visual Odometry Experiment (real world). The laterally observed apparent motion statistics are shown in table A.1.

Run	Min.	Max.	Mean	Standard deviation	SD as % of mean
A.13(a–g)	23.76 (20.72)	25.50 (21.49)	24.47 (21.29)	0.53 (0.25)	2.17 (1.18)
A.13(h–m)	25.04 (20.17)	28.55 (22.41)	26.26 (21.20)	1.28 (0.90)	4.89 (4.23)
A.13(n–w)	23.84 (22.01)	25.82 (23.07)	24.74 (22.41)	0.48 (0.32)	1.94 (1.42)
A.13(a–w)	23.76 (20.17)	28.55 (23.07)	25.05 (21.76)	1.07 (0.78)	4.27 (3.59)

Table A.1: *Flow*1 (and *Flow*2) statistics (real-world). This table shows the integrated (lateral) optic flow statistics for the experiments depicted in figure A.13. As can be seen, *flow*2 provides a better, more consistent measure of distance travelled, than *flow*1.

Appendix B

Further Homing Results

This appendix presents some of the miscellaneous homing results. This includes some of the raw data, presented in tabular form, used to construct the graphs presented in chapter 6.

B.1 Homing via image based warping

This section presents an anecdotal comparative study of the homing results obtained from an image-based warping approach[1] and the landmark-based approach presented in this thesis (§6). Both homing performance and computational cost are examined.

As described in §2.7.1, the image-based warping approach to visual homing, functions by hypothesising the new visual panorama expected given the current panorama and a hypothetical displacement vector. The homing procedure therefore searches through the many hypothetical movements for the one that generates the most similar expected panorama to that which was observed from home. The movement resulting in the most similar expected visual panorama, to that of home, should therefore also be very indicative of the way home.

Thus the image warping approach requires the 'warped space' to be searched. The search space is thus sampled with the intent of finding a match with that observed from home. Although a perfect match will not generally be found, the assumption is that the closest match found represents a location physically closer to the home position than the current position. In this way the homing procedure can be repeated until home is reached.

[1] See §2.7.1, specifically the work of Franz et al. (1997a).

The three-dimensional[2] 'warped space' is inherently infinite and continuous and thus must be sampled in a cost effective way, in an effort to find the visual match that coincides with home. The granularity of this sampling, however, affects not only the computational cost but also homing performance.

Figures B.1–B.10 show the homing performance of both the image-warping and landmark-based methods within several custom-designed[3] arenas.

To gauge the effect that various degrees of sampling had on the homing performance of the image-warping method, three different sampling sets were experimented with (W_1, W_2, W_3). Sampling set W_1 was the most comprehensive and consisted of 18 directions of movement $(0°, 20°, 40°, .., 340°)$ and 36 displacement distances (0cm, 20cm, 40cm, 60cm, 80cm,.., 700cm). Sampling set W_2 was less comprehensive consisting of 12 directions of movement $(0°, 30°, 60°, .., 330°)$ and 15 displacement distances (0cm, 10cm, 20cm, 40cm, 80cm, 140cm, 200cm, 260cm, 320cm,.., 680cm). Sampling set W_3 was least comprehensive, consisting of only 8 directions of movement $(0°, 45°, 90°, .., 315°)$ and 5 displacement distances (0cm, 20cm, 80cm, 300cm, 700cm).

As expected, homing performance was significantly affected by the degree of sampling. The more comprehensive the sampling the better the homing performance. However, this is at the cost of increased computation. Sampling set W_1 provided the best homing performance but was also computationally prohibitive. The inherent complexity of determining the homing vector can be estimated by the product of the number of samples in each dimension of the search space. In these image-warping experiments the panoramic image consisted of 360 pixels, one per degree of view. The inherent complexity of W_1 is therefore approximately $18 \times 35 \times (360x + 360^2y)$, where $360x$ refers to the computation of the warped panorama[4] and 360^2y the computation of robot orientation (i.e. panorama correlation). This translated to approximately 37 seconds computation[5] per homing vector, making the process completely

[2]The three dimensions are defined by the three movement parameters: rotation, displacement distance, and displacement direction. The displacement vector, however, is only relative to the hypothesised distance of the surrounding panoramic environment.

[3]These arenas were specifically designed to give a good impression of the overall performance of the homing methods and also to show some of the inherent homing difficulties associated with each method. The main difficulty for the image-warping approach is homing from a point outside of the immediately surrounding environment. The landmark-based approach, on the other hand, has most of its homing difficulties tied up with visual occlusion problems.

[4]See equation 2.22, §2.7.1.

[5]The simulations were performed on an SGI Indy workstation equipped with R4400

untenable for a real-time application.[6] Table B.1 gives a summary of the computational costs experienced in each case. Of the three basic sampling sets, W_3 is the only one approaching the response requirements of a real-time scenario.

To achieve some semblance of real-time performance whilst still retaining reasonable homing behaviour the search space must be further pruned in alternative ways. Apart from the obvious tactic of simply reducing the number of pixels making up the panoramic image,[7] there are several other options.

Perhaps the most attractive option is simply utilising a compass sensor to provide accurate robot orientation information. The effects of this would be two-fold. Firstly, the search space would be significantly reduced by removing the robot rotation dimension. The optimisation problem would then simply be a matter of determining the two remaining unknowns, home direction and distance, significantly reducing computation (see W_{2c} and W_{3c} in table B.1). Secondly, providing an accurate frame of reference for the snapshots would also tend to improve effective homing behaviour. This can be seen in figures B.11 and B.12 where the homing behaviour is again shown for the example test arenas, except in this case a common frame of reference (i.e. a compass sensor) is used to provide robot rotation information. Although some of the catchment areas do not appear to be significantly improved, a closer examination of the computed homing vectors does reveal a noticeable improvement in overall homing accuracy.

Instead of using a compass sensor, however, one could always simply use earlier

processors. Although these processing times are obviously machine dependent, they do give a good impression of the true trend. The inherent computational complexity of the algorithm involved remains the most significant factor.

[6] Or at least requiring parallel architecture to make the process viable for a real-time scenario.

[7] Although a further order of magnitude improvement in computational overhead can most easily be achieved by reducing the granularity of the panoramic image, the effects of such an action on homing quality need also be a consideration. For example, reducing the number of pixels from 360 to 100, reduces the required computation by a factor of 11. Using sample set W_3, this would lower the computational overhead to a fraction of a second (0.17s), providing good real-time performance. Depending on the application, W_2 could possibly become workable, providing some semblance of real-time performance with an overhead of only 0.9 seconds. Sampling set W_1 would, however, remain too computationally expensive for effective real-time performance (3.3s). Again the adverse effects of reducing the visual acuity may override the gains made in reactive speed. Nevertheless, this is highly speculative as the whole argument is really only relevant with respect to the specifics of the environment and required performance criteria. The homing performance effects of this change have not been examined in any detail and it is unclear how much of an adverse effect this would have on the image matching component of the process and consequentially also on homing. Given the inherent synthetic nature of the environment in the simulations, these tests would not be appropriate to assess this and results would remain inconclusive. Thus real-world experimentation would need to be conducted for a more conclusive analysis of effects.

approximations for parameters to help direct the search. Previous determinations of homing distance and direction could possibly provide close initial estimates, and thus assist in further pruning the search. Assuming the robot has not moved very far since the last snapshot was taken the homing distance and direction should be quite similar. However, this type of strategy does possess several flaws. As inevitable errors creep into calculations they can be carried through to possibly contaminate future estimates (i.e. misdirect the search) and consequently adversely effect homing behaviour.

Another alternative to reducing the computational requirements of the search is through another form of heuristic pruning. An image difference measurement, for example, can be quite indicative, up to a point, of homing distance. A large difference between home and current snapshots would indicate a larger homing distance than a small image difference. By inferring spatial distance from measured image distance (i.e. minimum snapshot difference) the sampling can be significantly weighted to concentrate the search at specific homing ranges. A linear approximation for the relationship between spatial and image distance is, in fact, used by Franz et al. (1997a) to gauge the distance of home and thus heavily prune the search space. The speed of the algorithm can thus be drastically improved. However, as with all heuristics optimal performance is never guaranteed. In this case, it is assumed that this correlation will remain valid and be reasonably predictable, for the various homing environments and homing distances required.

As is indicated in table B.1, there is a stark difference between the computational overhead of the landmark-based approach and the image-warping variants. Consider that the average computational overhead for the landmark-based approach (L_{h3}) was 0.003 seconds per homing vector calculation, which is almost three orders of magnitude (a factor of 630) less than that required by the image-warping variant W_3. Although this computation time is dependent on the number of visual features that are required to be paired, there will always be less features than pixels.[8]

With the notable exception of occlusion there were no perceptual limitations included in this set of experiments. The panorama is perfectly observable. However, a few perceptual embellishments were included to aid the image-based

[8]For a comparison, consider the fact that 100 visual features increases the average computational overhead of L_{h3} to only ≈ 0.01 seconds. This is still an order of magnitude improvement over W_{3c}.

	Complexity	CPU time	
		vector	vector field
W_1	$18 \times 35(360x + 360^2y)$	37 secs	26 hours
W_2	$12 \times 14(360x + 360^2y)$	10 secs	7 hours
W_3	$8 \times 4(360x + 360^2y)$	1.9 secs	1.3 hours
W_{2c}	$12 \times 14 \times 360(x + y)$	0.61 secs	25.6 mins
W_{3c}	$8 \times 4 \times 360(x + y)$	0.12 secs	5 mins
L_{h3}	—	0.003 secs	7.2 secs

Table B.1: The Computational Complexity of Producing a Homing Vector. This table shows the computational costs involved in computing a homing vector. The three basic variants of the image-warping approach (W_1, W_2, W_3) using differing sampling sets are shown to incur correspondingly high computational expense. The use of a compass sensor (W_{3c}, W_{3c}) is shown to significantly reduce computational overheads and improve speed. The landmark-based approach (L_{h3}), using correspondence method $h3$, is shown to be the most computationally efficient.

warping approach. Firstly, the relative range of the panoramic environment was provided artificially[9] by simply averaging the ranges of the objects within the arena. This assists the warping approach by making the scope of the required sampling set much more well defined. Secondly, the objects within the environment were given unique grey-scale luminance values.[10] The landmark-based homing method, however, still perceives the objects as being identical.

The homing-vector fields shown in figures B.1–B.10 provide a good snapshot of the homing performance, while at the same time depicting the failures inherent in the system. As expected, the image-warping approach to homing, starts to fail when the underlying assumptions about image adjacency and constant panoramic-range are violated. Reliable homing is thus generally restricted to the interior region defined by the immediately surrounding environment.[11] This

[9]Although unrealistic, this really only affects the degree of required sampling. The absence of this knowledge can be compensated for by providing a more extensive sampling set. The disadvantage being the additional computational expense of increasing the size of the sampling set or, alternatively, the performance degradation caused by increasing the sampling granularity. It is thus very important to be able to estimate beforehand, as close as possible, the required homing distances relative to the distance of the surrounding environment.

[10]The justification for this was that in the real world this would not be an unreasonable assumption. A real-world panorama will usually provide a kaleidoscope of luminance values. This, in turn, significantly improves the matching process, which is inherently pixel-based. See figure B.13 for a few example catchment areas resulting from the use of homogeneous landmarks.

[11]Image-warping, and in fact image-based approaches in general, are not well suited for

can be seen quite clearly by comparing figures B.1(a) and B.2(a). When the four objects are equidistant and, more importantly, surrounding the required homing region (fig. B.1a), there is no significant homing difficulty. The robot can successfully home from anywhere within this region. However, as shown in figure B.2(a), problems arise when homing is attempted from outside of the initial surrounding environment. Further good examples of this are shown in figures B.6, B.4, B.5, B.8, and B.9. The reliable catchment area is essentially restricted to the most interior region defined by the objects within the homing environment.[12]

The landmark-based approach to homing does not suffer from this problem. As shown in figures B.1–B.10, the W_1 approach is very robust in terms of catchment area. There is no implicit limitation on how different the views can be in order to still be capable of reliable homing. Identifying, not necessarily heterogeneous, visual features relaxes the constraints on visual adjacency. However, this does not preclude problems associated with the observability of the visual features, which are used exclusively in determining the homing direction. Visual occlusion being the main culprit in this case. Occlusion has been observed to cause two main classes of problems. The first class of problem is the occasional formation of small pools of local minima directly behind landmarks where the robot may become trapped. This is the problem most frequently manifested.

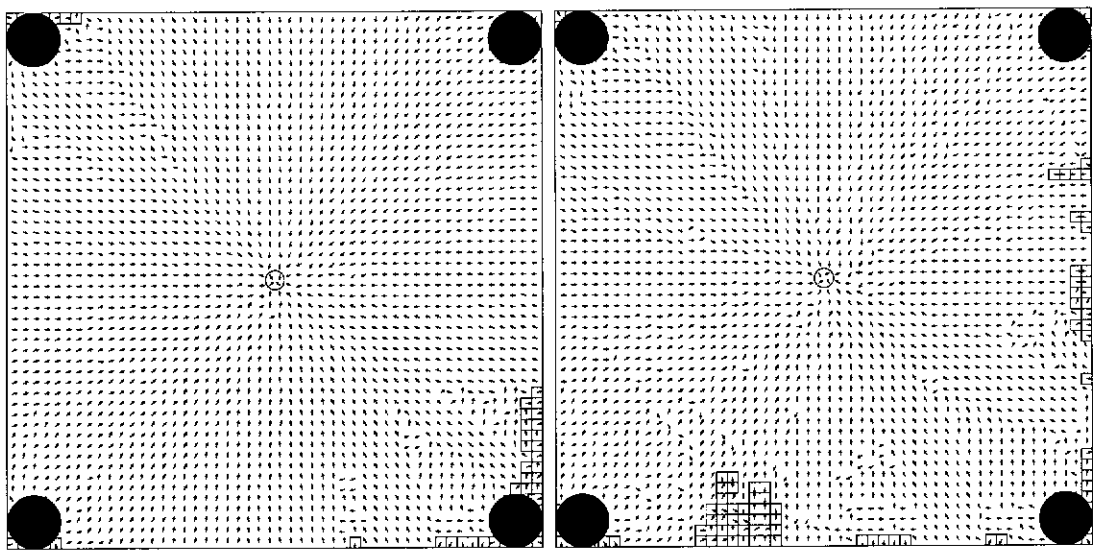
This can be seen quite clearly in figures B.7(d) and B.8(d). Although, the discrete nature of the vector field does exaggerate this phenomenon, this problem does arise often enough to justify the use of additional behaviours to evade such traps. Several solutions have been proposed and implemented to enable the robot to successfully escape these situations.[13] However, given the nature of the homing-vector field generation procedure this is not indicated in the figures. The second class of problem arises when there exist multiple locations within the homing region which have identical, or at least very close, visual panoramas. This is essentially a problem of multiple 'global' minima. Unfortunately, there is no "visual homing" solution to this problem without relaxing some of the previous assumptions (see §6.7.1.1). See for example figure B.6. In this case, visual occlusion causes multiple locations to appear similar to that observed from

obstacle rich environments, where pixel adjacency is violated to any great extent.

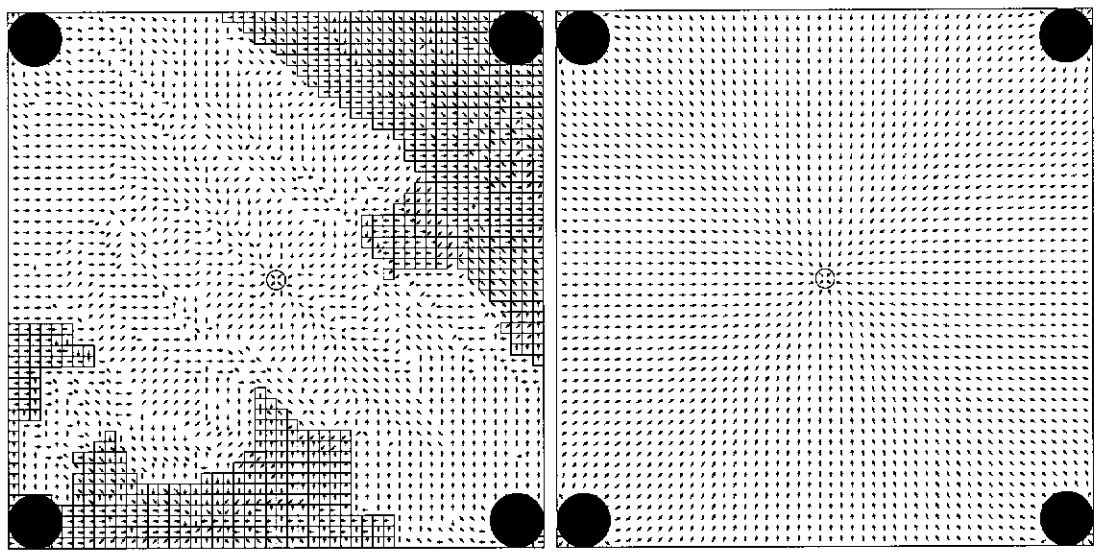
[12]The asymmetrical nature of these homing-vector fields is primarily due to the heterogeneous nature, in terms of luminance, of the landmarks. See figure B.13 for example results using homogeneous landmarks.

[13]See §6.7.1.1. Using a circumnavigatory behaviour the offending landmark can be bypassed and visual homing can successfully resume on the other side.

home. Making the best of what is currently perceived, the homing algorithm ignores missing landmarks, which in this case may result in a misleading landmark correspondence and hence lead to a false minimum.

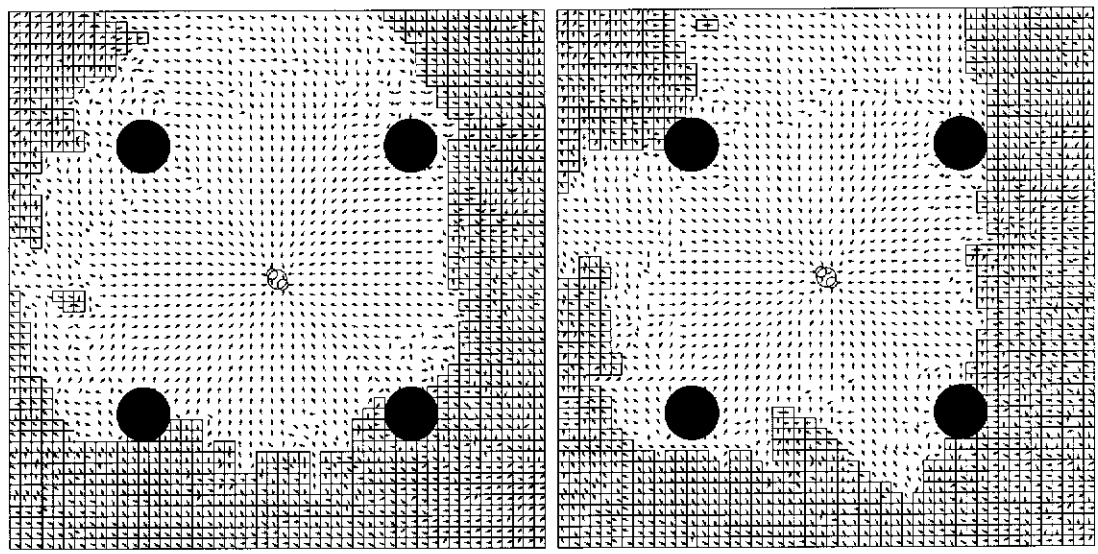


(a) W_1 (b) W_2

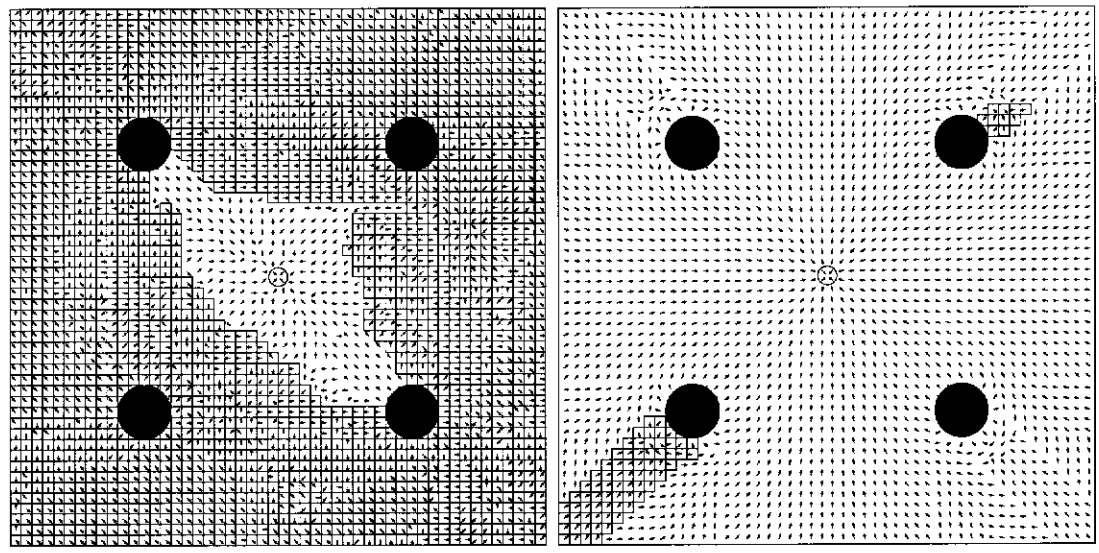


(c) W_3 (d) L_{h3}

Figure B.1: Image Warping versus Landmark Based Homing (simulation). The performance of variants of the image warping approach (W_1, W_2, W_3) are compared with the landmark-based approach (L_{h3}). The homing-vector fields show the extent of the resulting catchment area in each case.

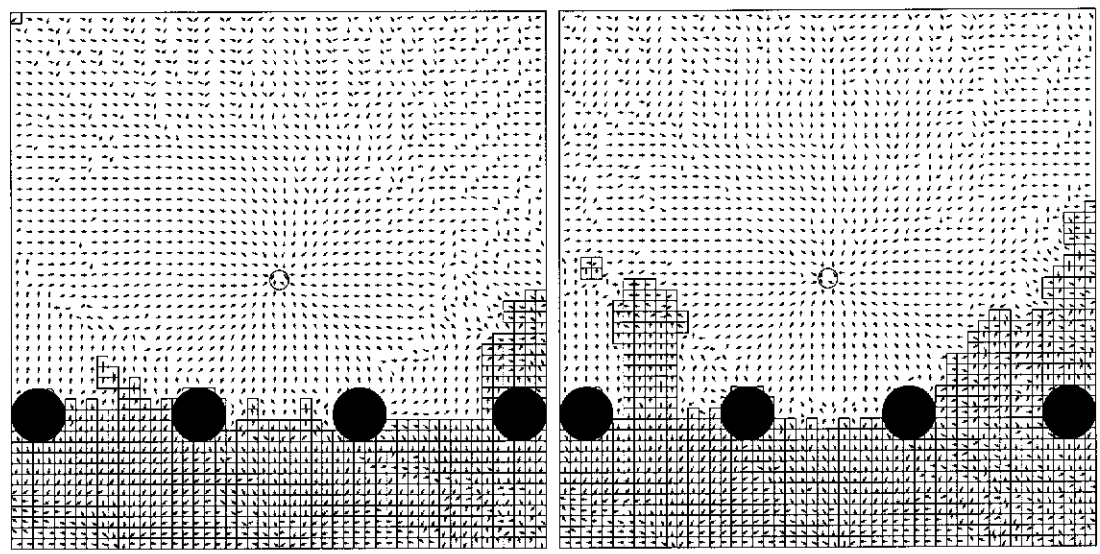


(a) W_1 (b) W_2

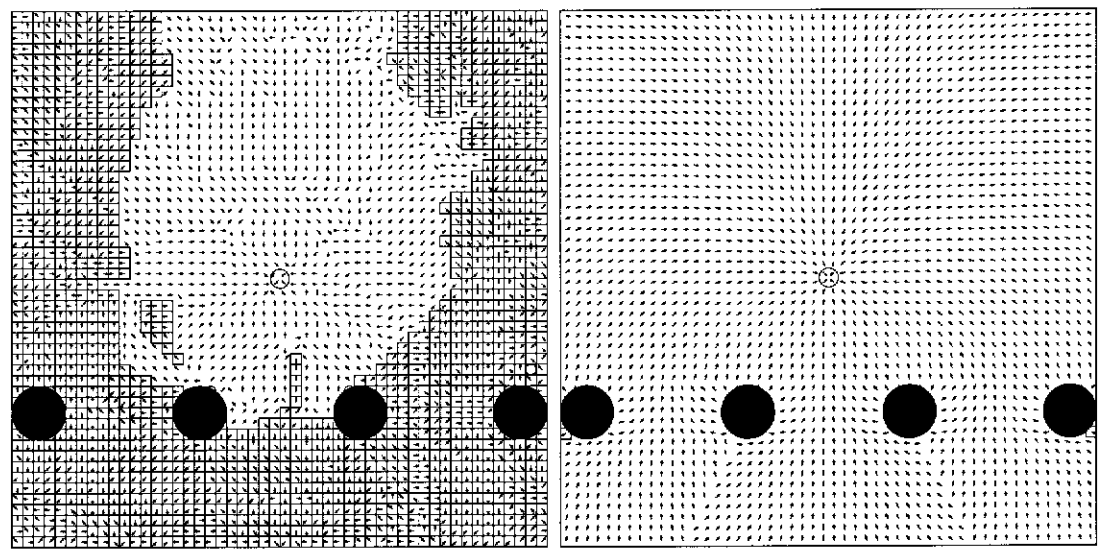


(c) W_3 (d) L_{h3}

Figure B.2: Image Warping versus Landmark Based Homing (simulation). The performance of variants of the image warping approach (W_1, W_2, W_3) are compared with the landmark-based approach (L_{h3}). The homing-vector fields show the extent of the resulting catchment area in each case.

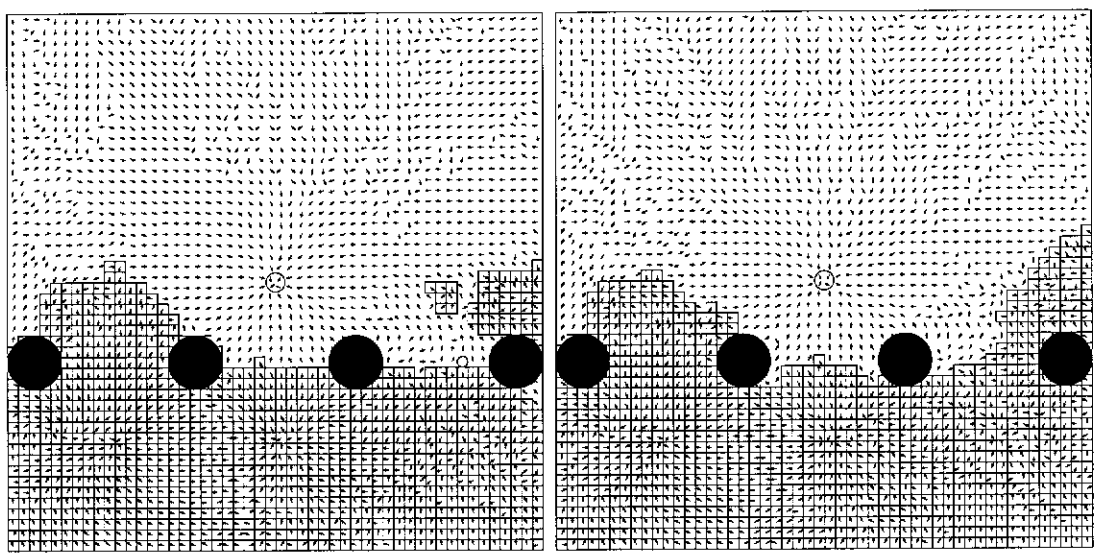


(a) W_1 (b) W_2

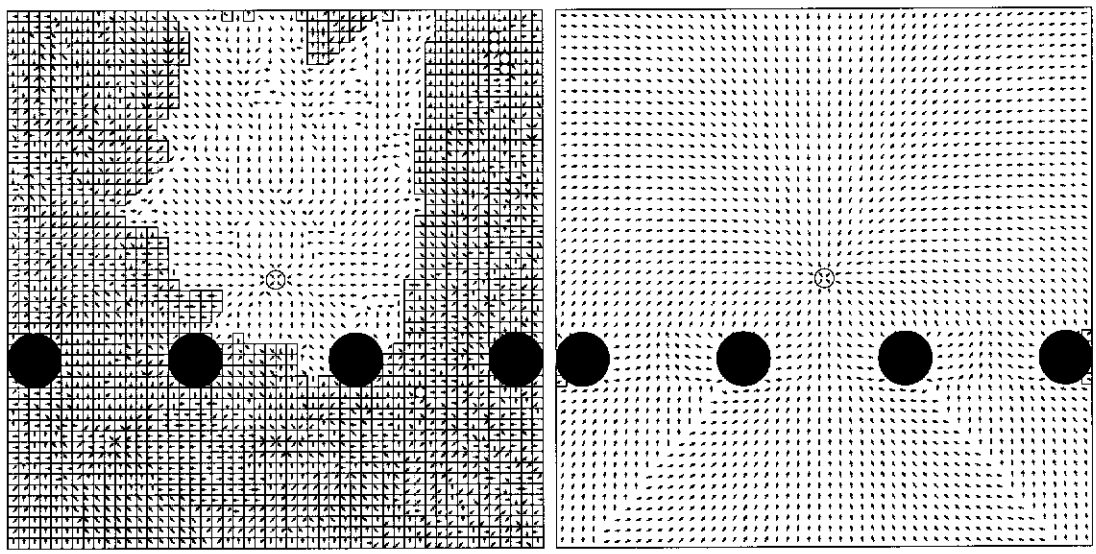


(c) W_3 (d) L_{h3}

Figure B.3: Image Warping versus Landmark Based Homing (simulation). The performance of variants of the image warping approach (W_1, W_2, W_3) are compared with the landmark-based approach (L_{h3}). The homing-vector fields show the extent of the resulting catchment area in each case.

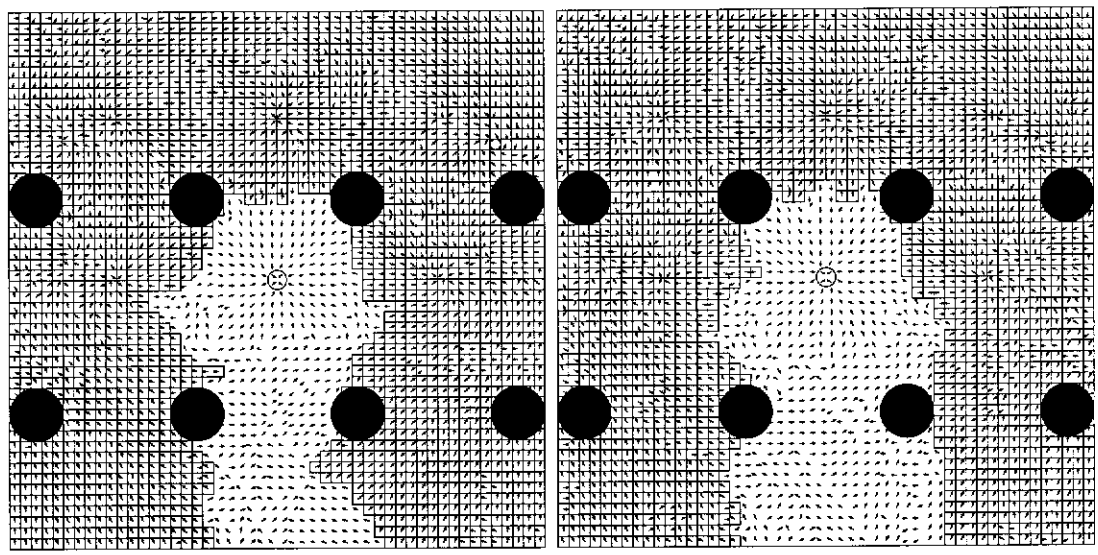


(a) W_1 (b) W_2

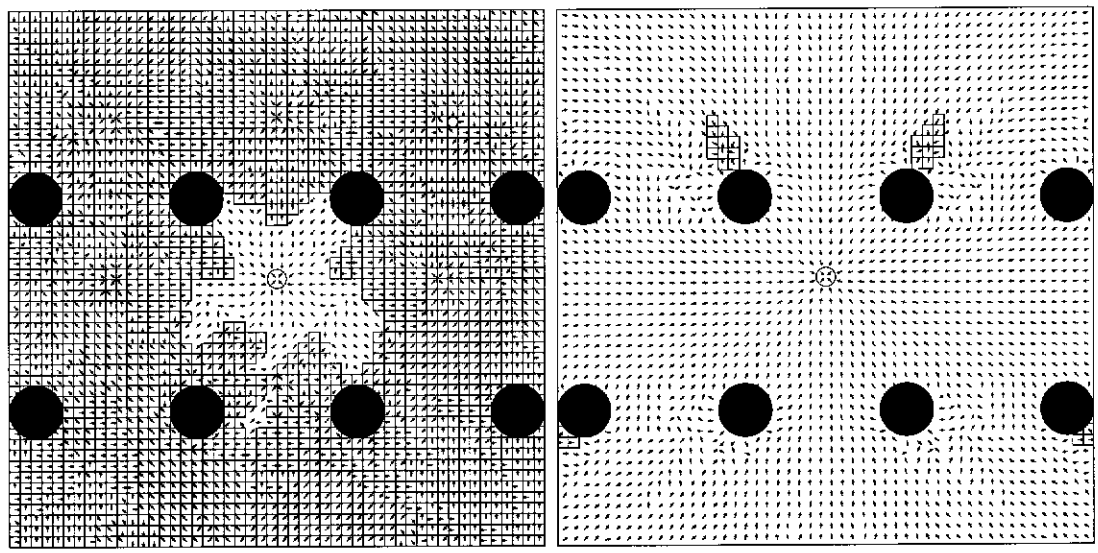


(c) W_3 (d) L_{h3}

Figure B.4: Image Warping versus Landmark Based Homing (simulation). The performance of variants of the image warping approach (W_1, W_2, W_3) are compared with the landmark-based approach (L_{h3}). The homing-vector fields show the extent of the resulting catchment area in each case.

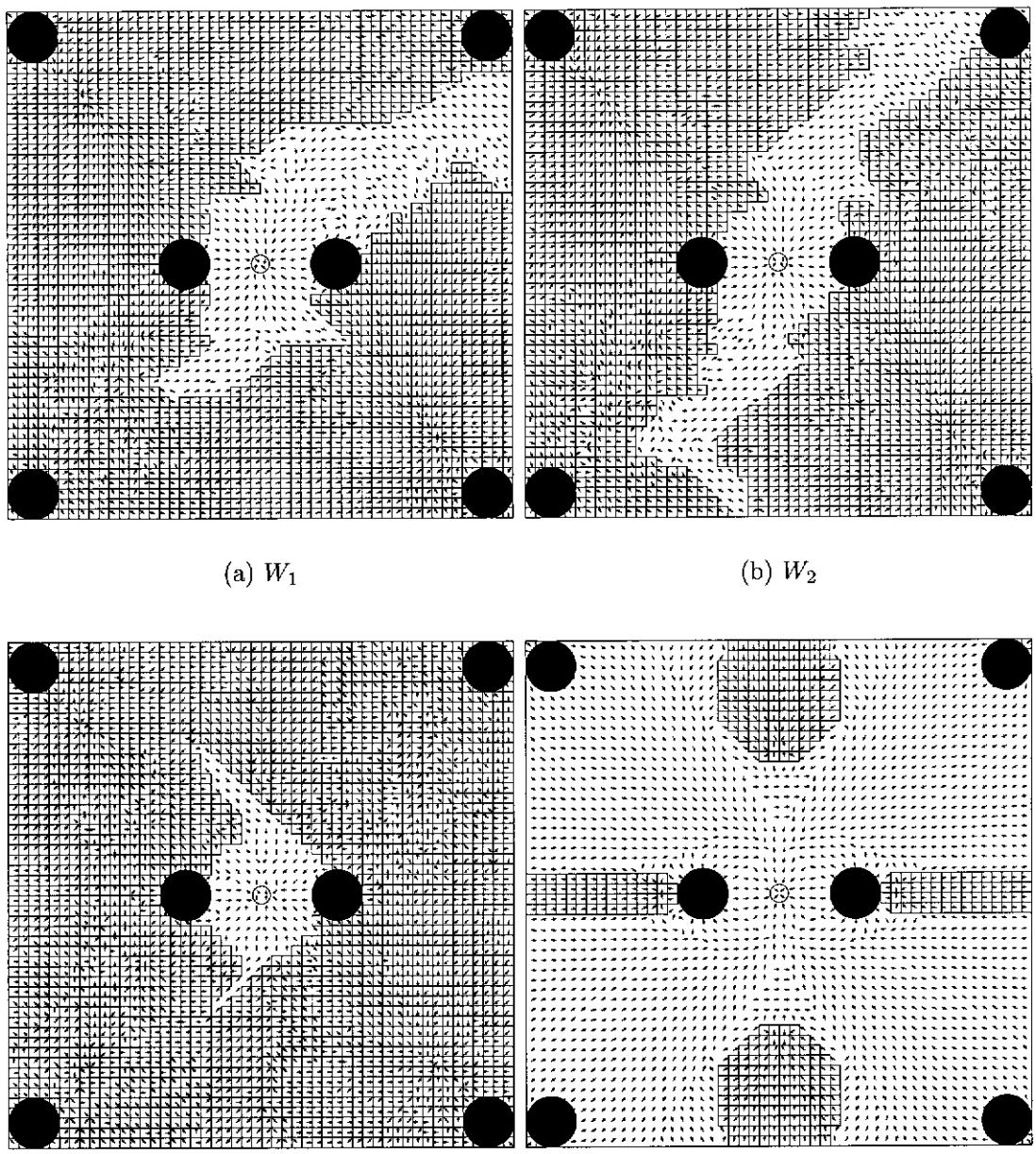


(a) W_1 (b) W_2



(c) W_3 (d) L_{h3}

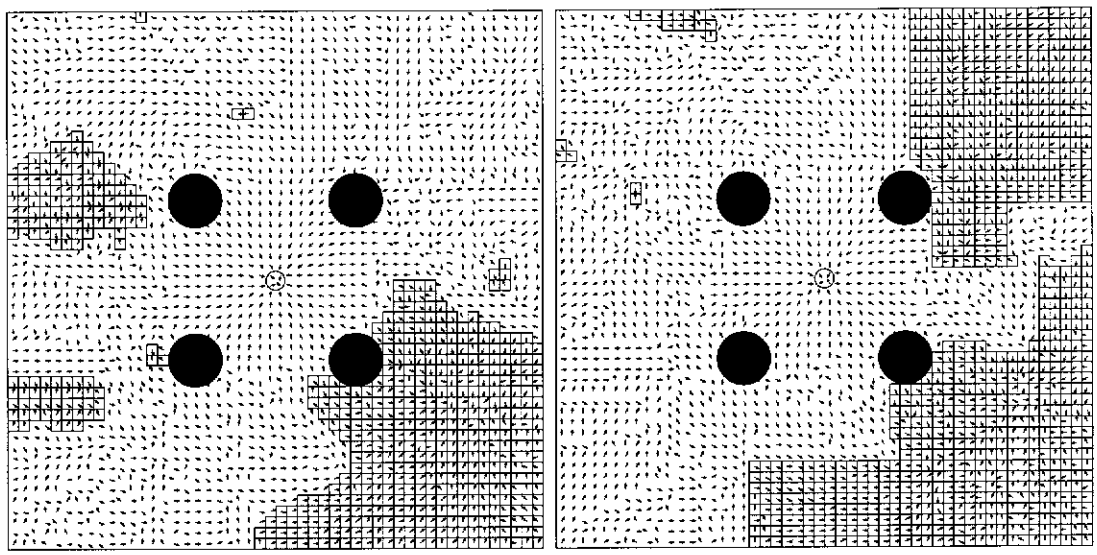
Figure B.5: Image Warping versus Landmark Based Homing (simulation). The performance of variants of the image warping approach (W_1, W_2, W_3) are compared with the landmark-based approach (L_{h3}). The homing-vector fields show the extent of the resulting catchment area in each case.



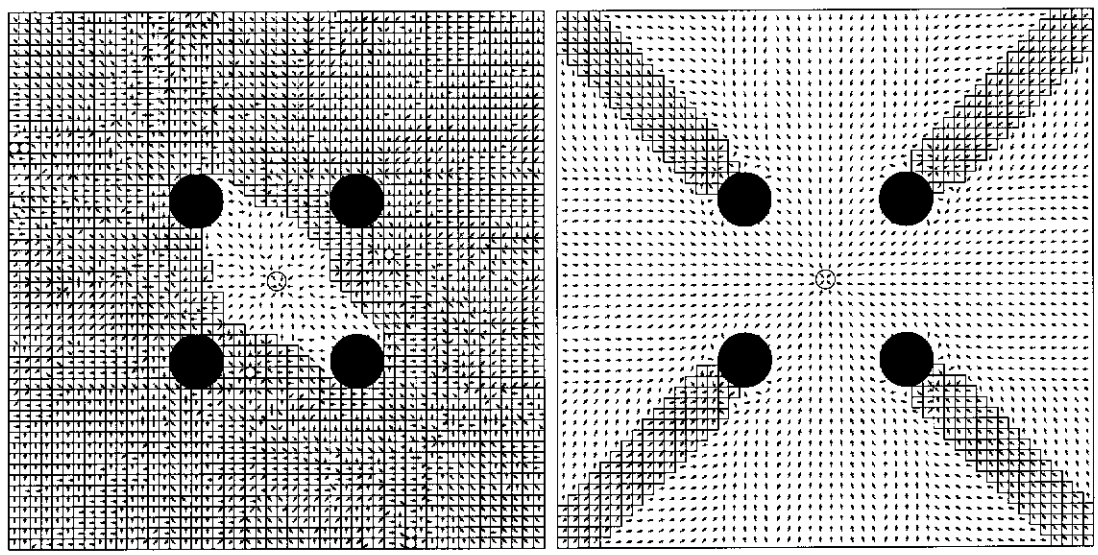
(a) W_1 (b) W_2

(c) W_3 (d) L_{h3}

Figure B.6: Image Warping versus Landmark Based Homing (simulation). The performance of variants of the image warping approach (W_1, W_2, W_3) are compared with the landmark-based approach (L_{h3}). The homing-vector fields show the extent of the resulting catchment area in each case.

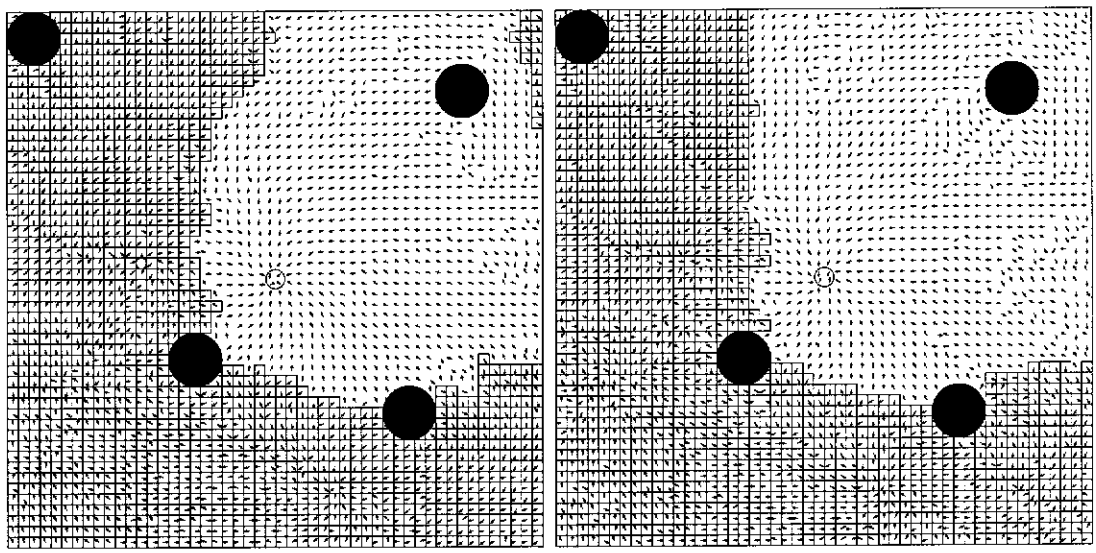


(a) W_1 (b) W_2

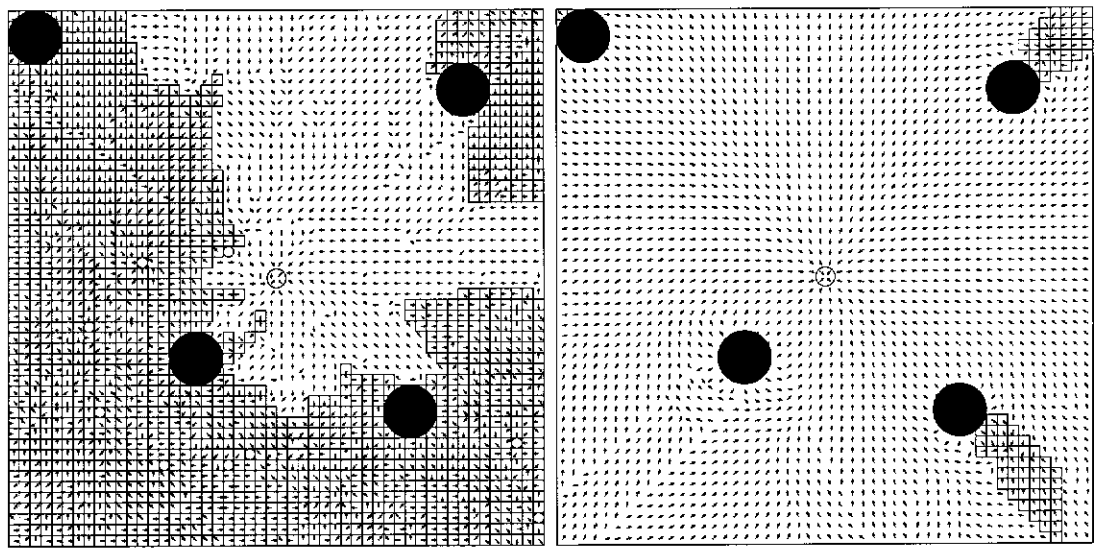


(c) W_3 (d) L_{h3}

Figure B.7: Image Warping versus Landmark Based Homing (simulation). The performance of variants of the image warping approach (W_1, W_2, W_3) are compared with the landmark-based approach (L_{h3}). The homing-vector fields show the extent of the resulting catchment area in each case.

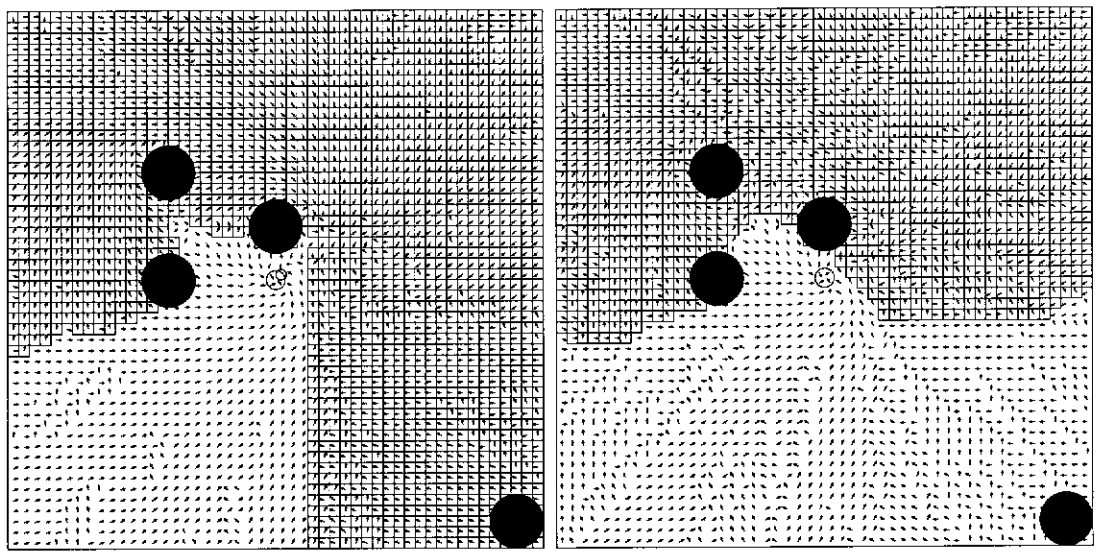


(a) W_1 (b) W_2

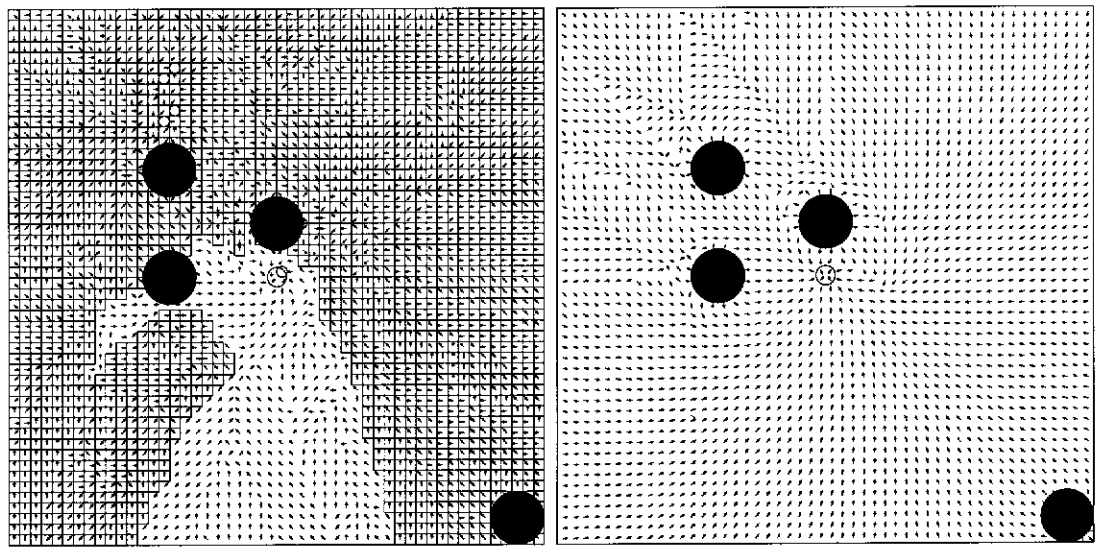


(c) W_3 (d) L_{h3}

Figure B.8: Image Warping versus Landmark Based Homing (simulation). The performance of variants of the image warping approach (W_1, W_2, W_3) are compared with the landmark-based approach (L_{h3}). The homing-vector fields show the extent of the resulting catchment area in each case.

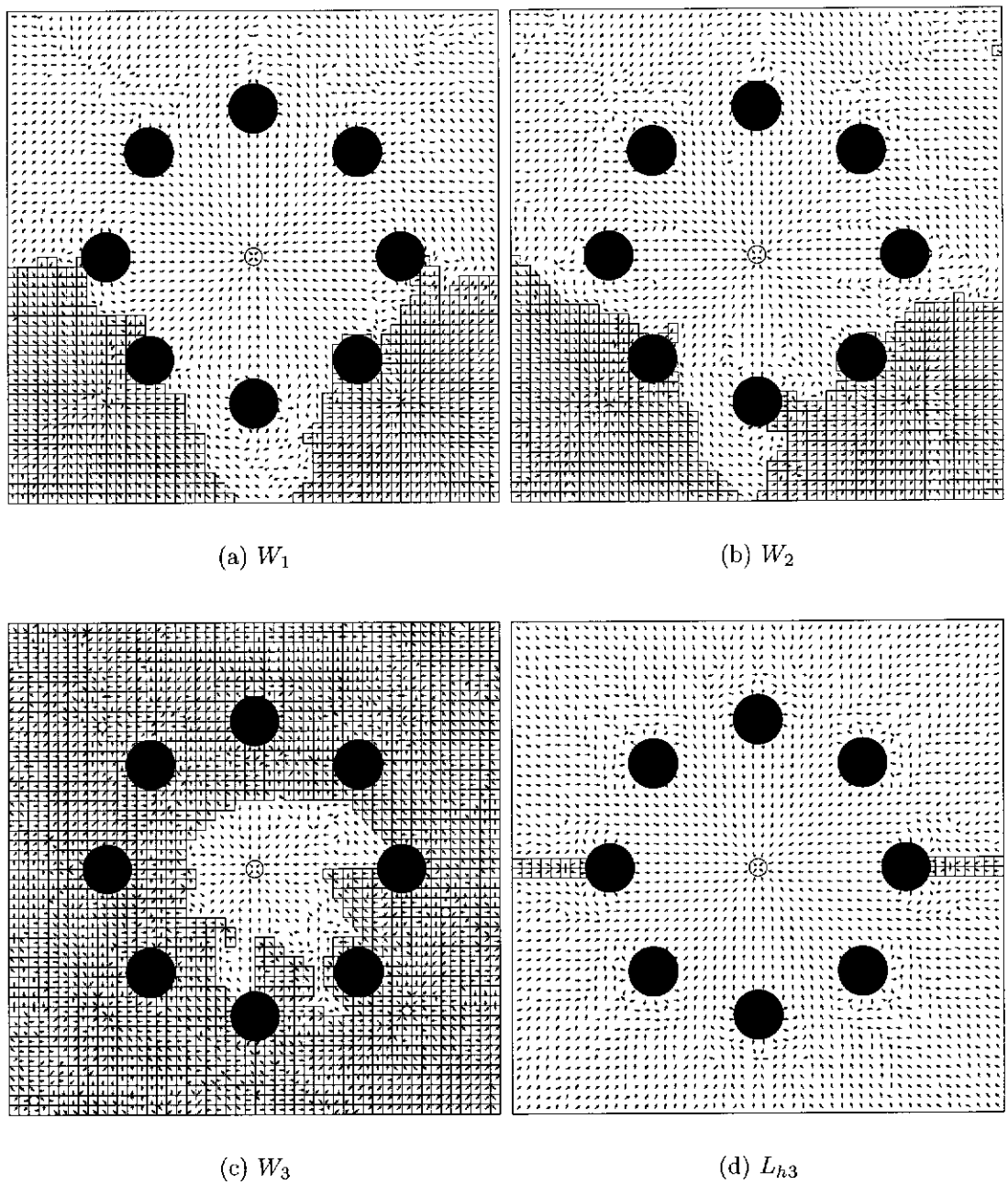


(a) W_1 (b) W_2



(c) W_3 (d) L_{h3}

Figure B.9: Image Warping versus Landmark Based Homing (simulation). The performance of variants of the image warping approach (W_1, W_2, W_3) are compared with the landmark-based approach (L_{h3}). The homing-vector fields show the extent of the resulting catchment area in each case.



(a) W_1 (b) W_2

(c) W_3 (d) L_{h3}

Figure B.10: Image Warping versus Landmark Based Homing (simulation). The performance of variants of the image warping approach (W_1, W_2, W_3) are compared with the landmark-based approach (L_{h3}). The homing-vector fields show the extent of the resulting catchment area in each case.

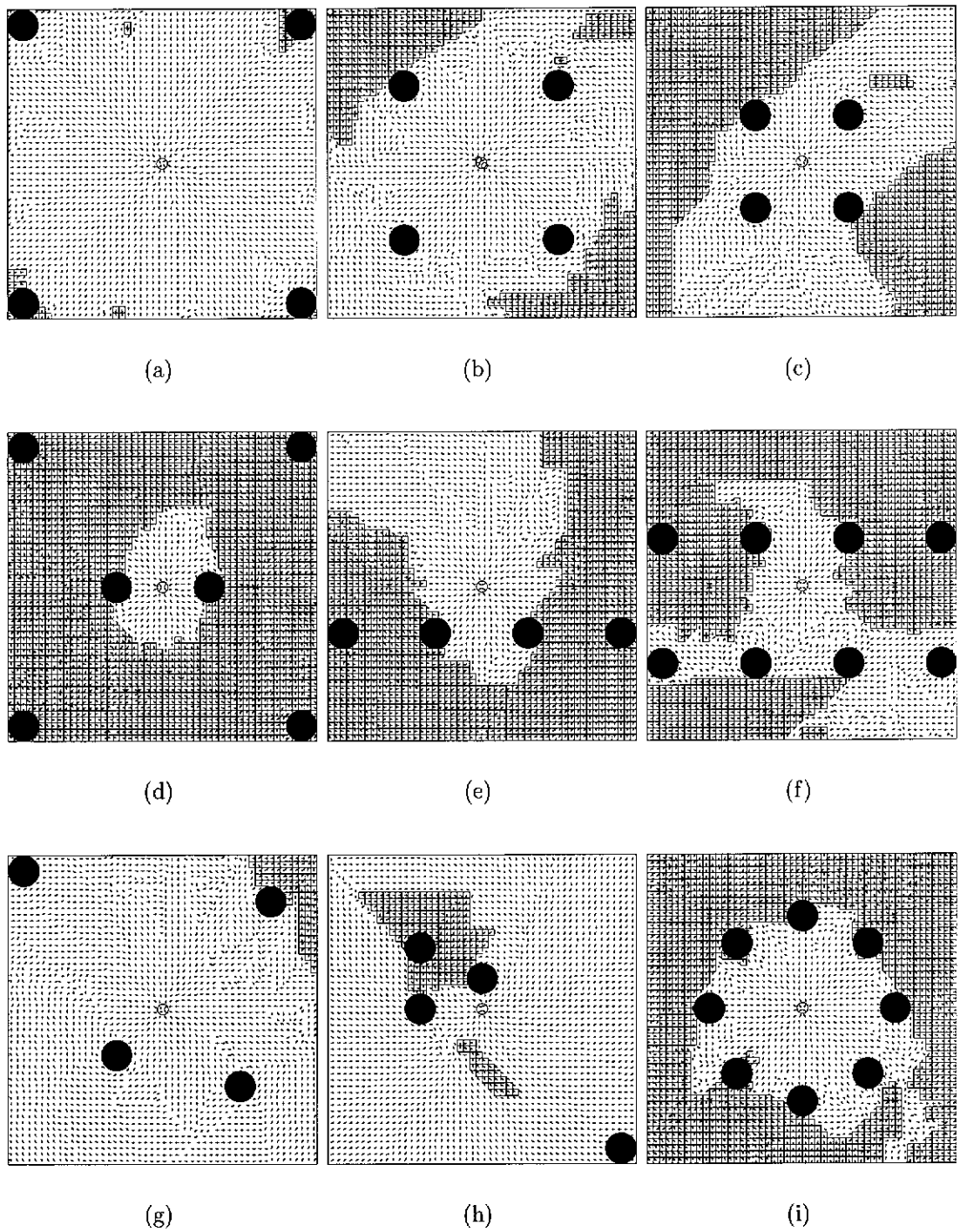


Figure B.11: Homing with a Compass and Image-Based Warping (simulation). The homing behaviour resulting from a variant of the image warping approach utilising a compass sensor (W_{2c}), is shown. A compass sensor reduces computation and increases homing accuracy. The homing-vector fields show the extent of the resulting catchment area for each example scenario.

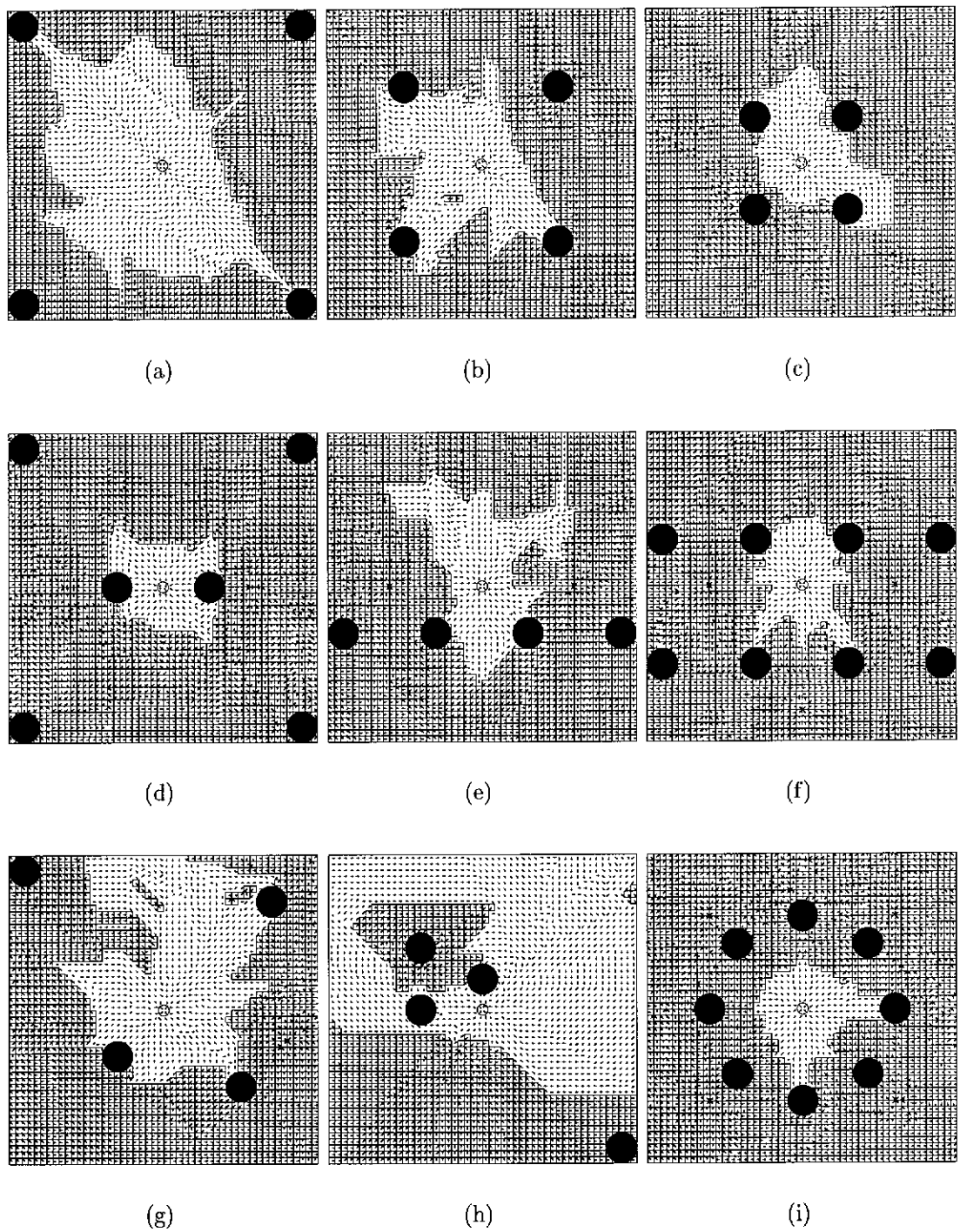


Figure B.12: Homing with a Compass and Image-Based Warping (simulation). The homing behaviour resulting from a variant of the image warping approach utilising a compass sensor (W_{3c}), is shown. A compass sensor reduces computation and increases homing accuracy. The homing-vector fields show the extent of the resulting catchment area for each example scenario.

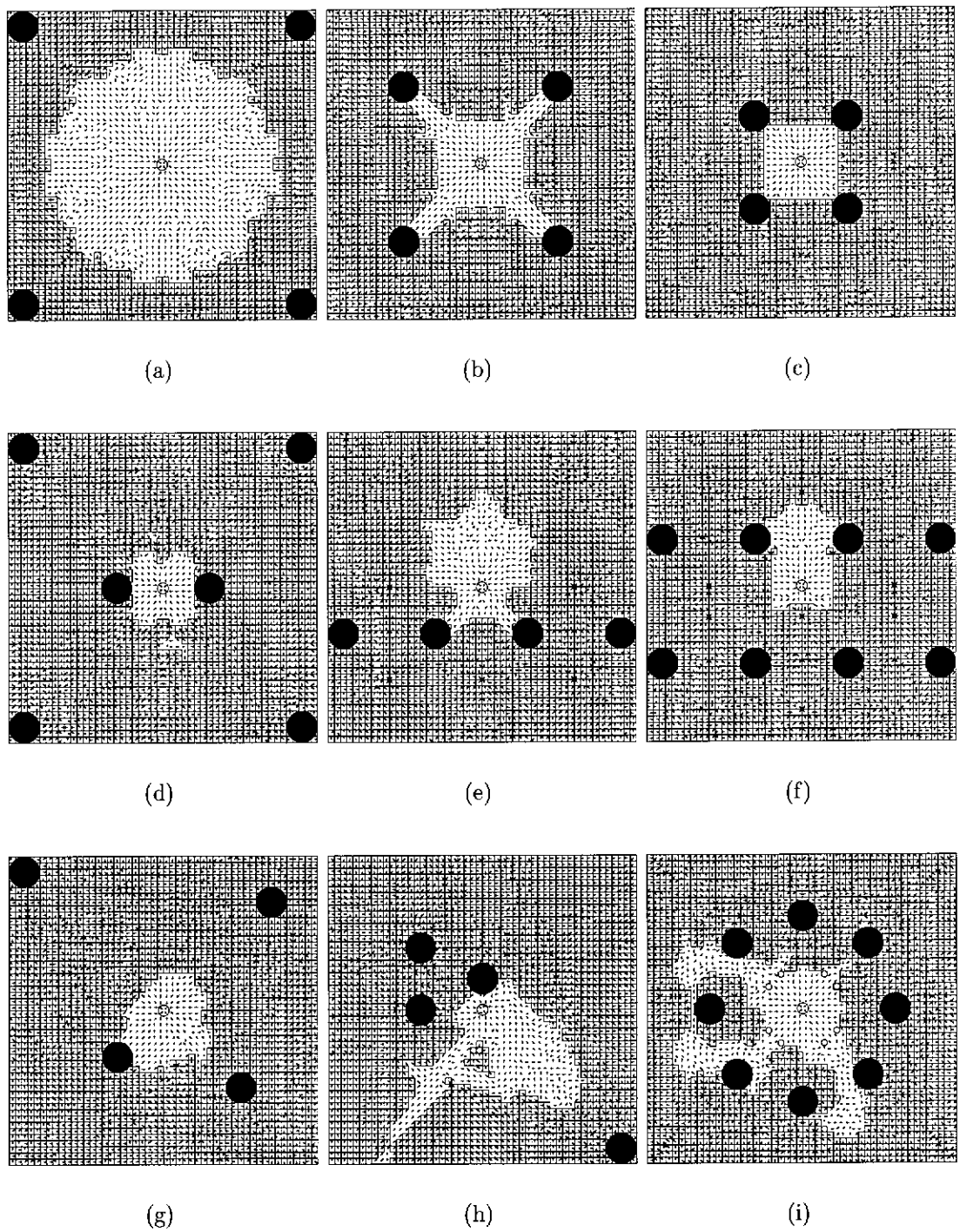


Figure B.13: Homing via Image-Based Warping with Homogeneous Landmarks (simulation, W_3). The homing-vector fields show the performance of the image warping approach (using sampling set W_3), within an arena consisting of homogeneous landmarks. The extent of the resulting catchment area is shown for each sample scenario.

B.2 Correspondence-Method Comparisons and Source Data

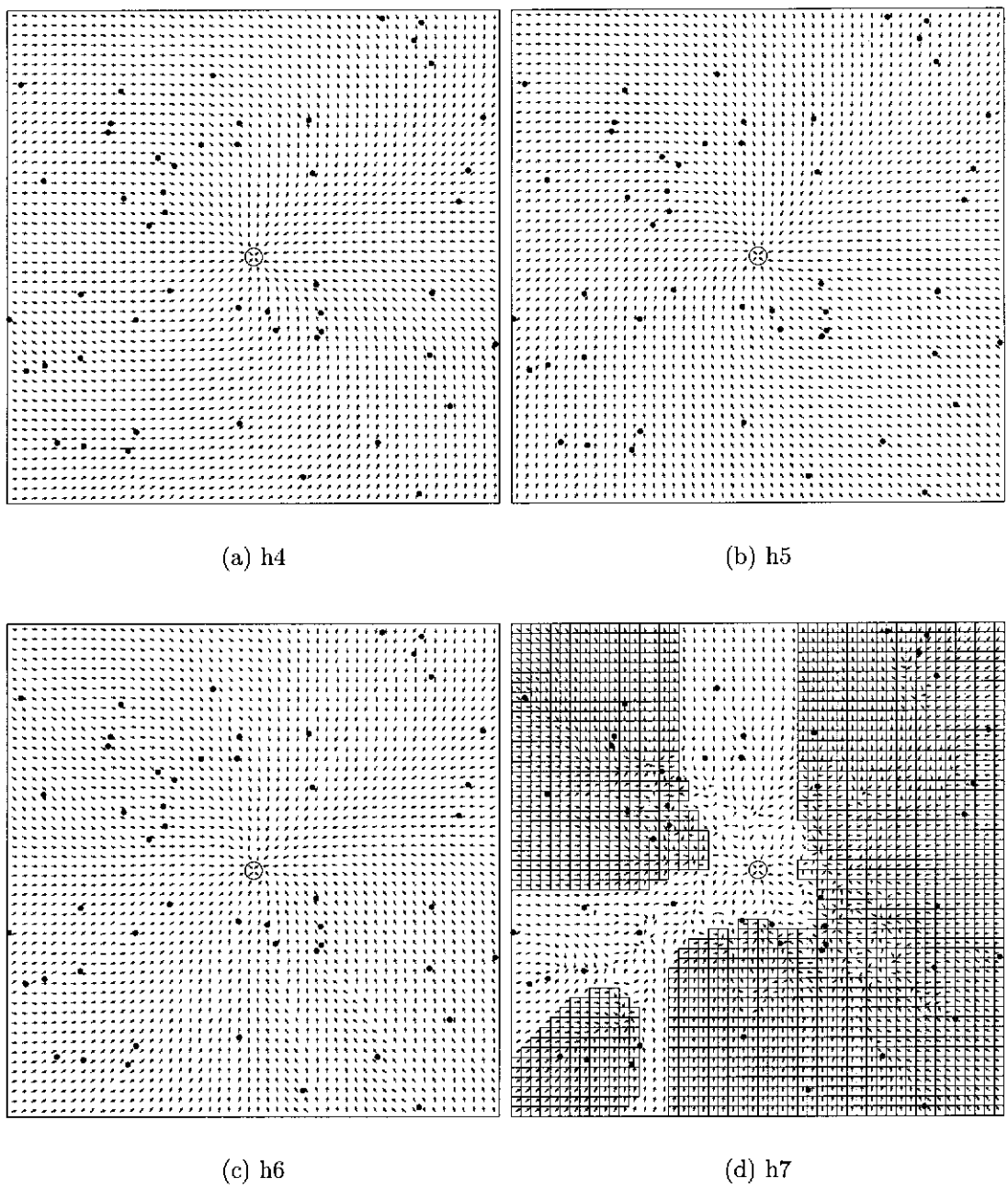
This section presents some of the miscellaneous findings comparing the results of the performance criteria for various landmark correspondence methods. Also presented are the source data for these statistical comparisons and the source data for the bearing-based error model described in §6.5.1.3.

The vector fields depicted in figure B.14 show how the correspondence methods h4, h5, h6, and h7 perform within the same densely packed arena as was presented in figure 6.13 for methods h2 and h3. Differences in the field patterns can clearly be seen, even between methods h4, h5 and h6.

Figures B.15 and B.16 show the average home-vector error experienced with all the tested correspondence methods and unit vector variants. The effect of both homing range and numbers of landmarks on this error is shown. Figures B.17 and B.18 show the home-vector and angular pairing errors, respectively. Again, the relationship between these error criteria and both homing range and landmark numbers, is shown graphically.

Tables B.2, B.3, and B.4 detail the source data for the home-vector, homing-vector, and angular pairing errors, respectively. For comparison, the data for 'non-squared error' versions of the home-vector error, homing-vector error, and angular-pairing error, is also shown (tables B.7, B.8, and B.9). Tables B.5 and B.6 show the mean and standard deviation of the final homing distances, observed with each of the correspondence methods and selected unit vector variants there of.

Tables B.10, B.11, and B.12 detail the source data for the home-vector, homing-vector, and angular pairing errors, observed using correspondence method h3 under varying degrees of artificially introduced bearing-based error. Again, for comparison, the data for 'non-squared error' versions of the home-vector error, homing-vector error, and angular-pairing error, are also shown (tables B.14, B.15, and B.16). Table B.13 shows the mean and standard deviation of the final homing distances, observed under varying degrees of bearing-based error.



(a) h4 (b) h5

(c) h6 (d) h7

Figure B.14: Homing-Vector Field (simulation, h4–h7). A comparison is made between the homing performance resulting from landmark correspondence methods h4, h5, h6, and h7, within a very dense arrangement of landmarks. The catchment area provided by method h7 is clearly inferior to the other methods.

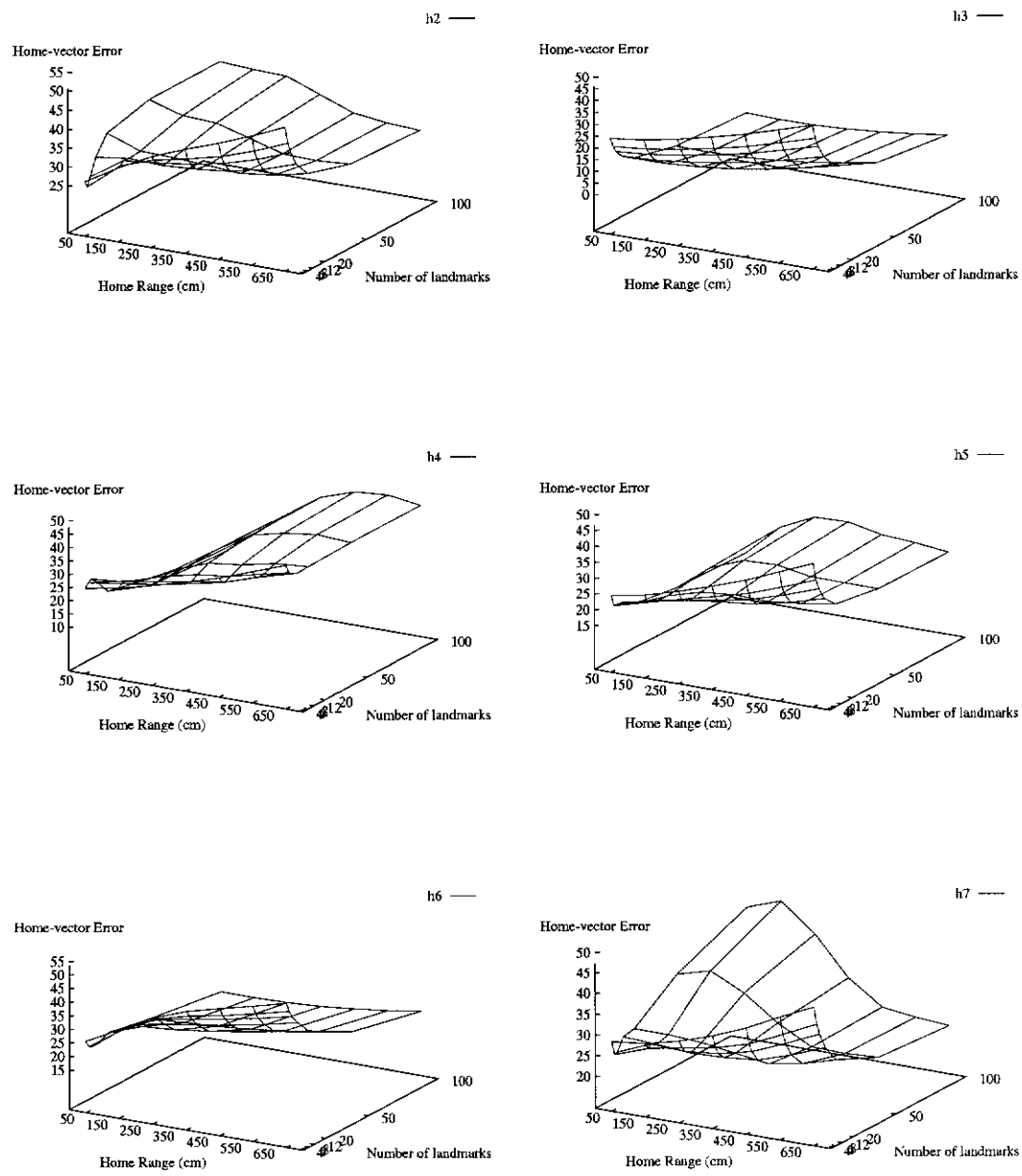


Figure B.15: Average Home-Vector Error (simulation, h2–h7). Homing accuracy is gauged by the average angular difference between the computed homing-vector direction and the true bearing of home ($\sqrt{av(|\gamma \dot{-} \phi|^2)}$). This error is graphed for the correspondence methods h2, h3, h4, h5, h6, and h7. The graphs show how this error is affected by both homing range and landmark numbers, for each method.

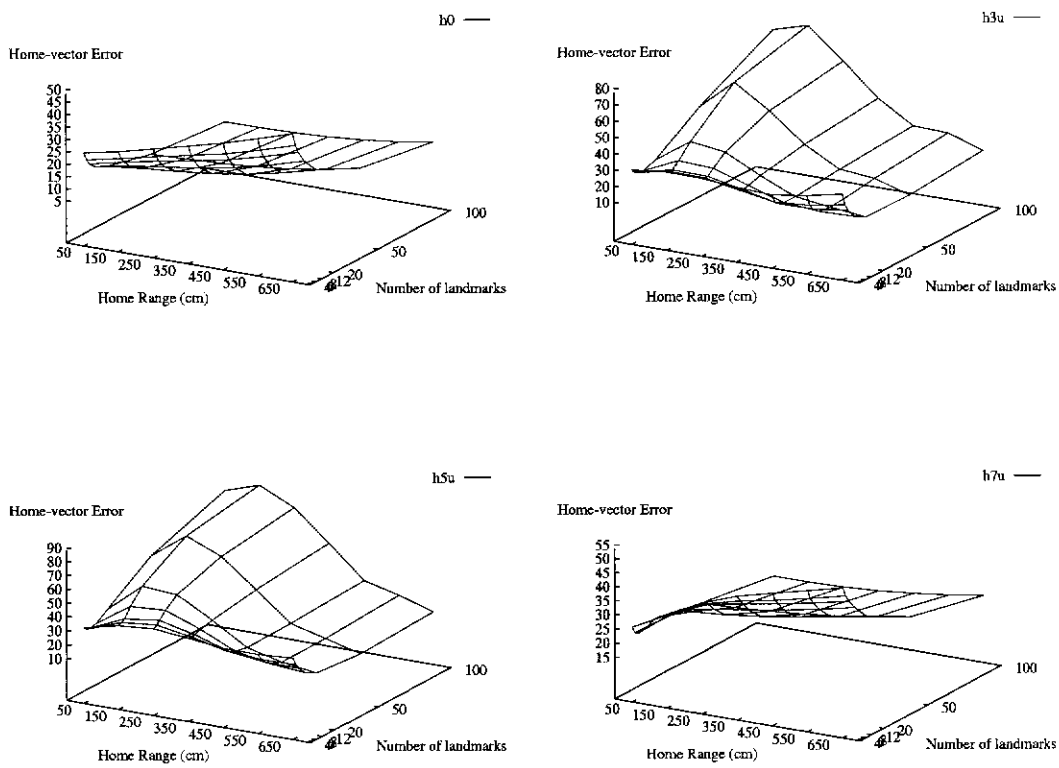


Figure B.16: Average Home-Vector Error (simulation, h0,h3u,h5u,h7u). Homing accuracy is gauged by the average angular difference between the computed homing-vector direction and the true bearing of home ($\sqrt{av(|\dot{\gamma}-\dot{\phi}|^2)}$). This error is graphed for the correspondence methods and unit vector variants h0, h3u, h5u, and h7u. The graphs show how this error is affected by both homing range and landmark numbers, for each method.

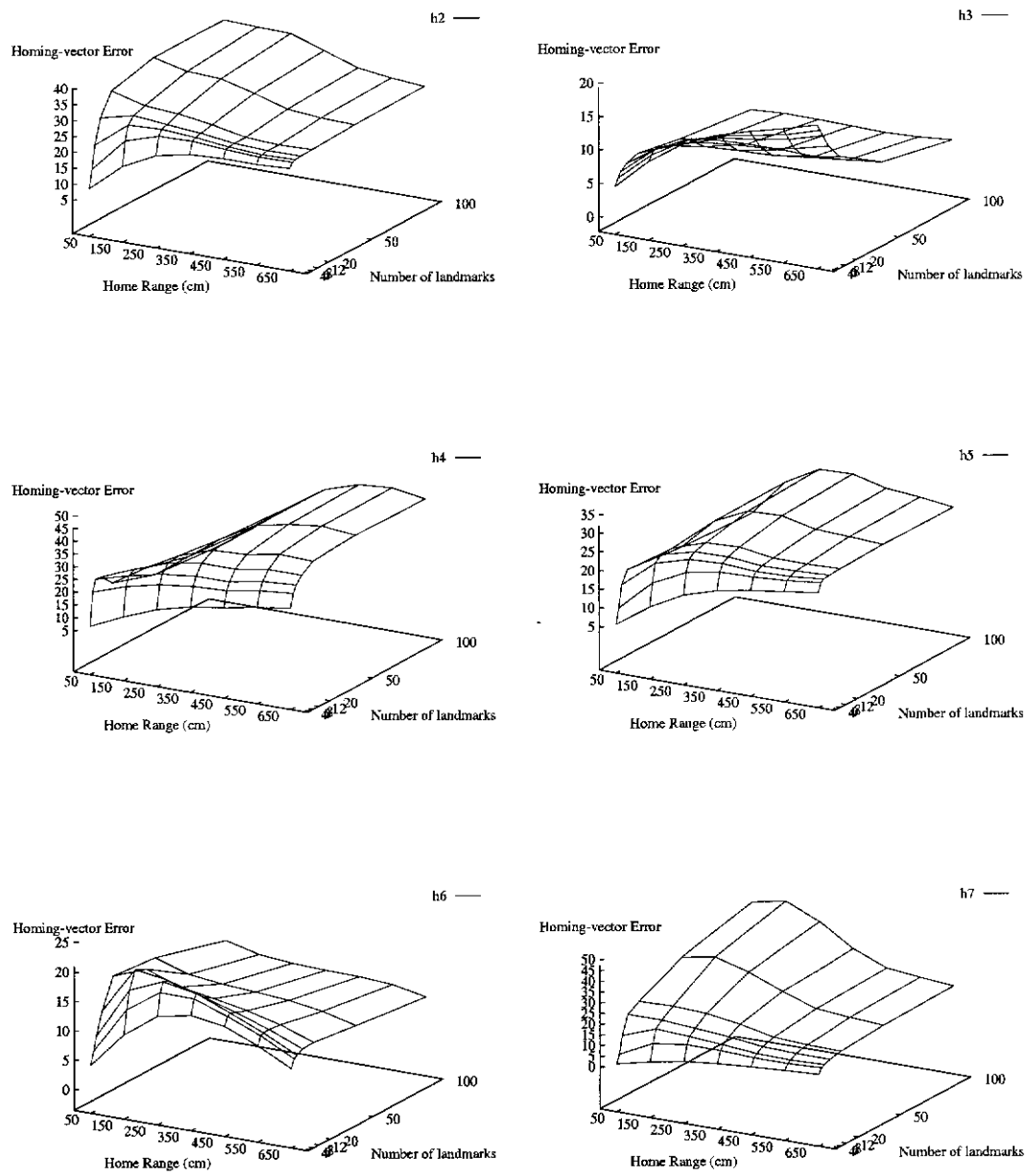


Figure B.17: Average Homing-Vector Error (simulation, h2–h7). Navigational error (arising from the mismatching of landmarks) is gauged by the average angular difference between the computed homing-vector direction and that obtained under perfect correspondence ($\sqrt{av(|\dot{\gamma}-\dot{\varphi}|^2)}$). This error is graphed for the correspondence methods h2, h3, h4, h5, h6, and h7. The graphs show how this error is affected by both homing range and landmark numbers, for each method.

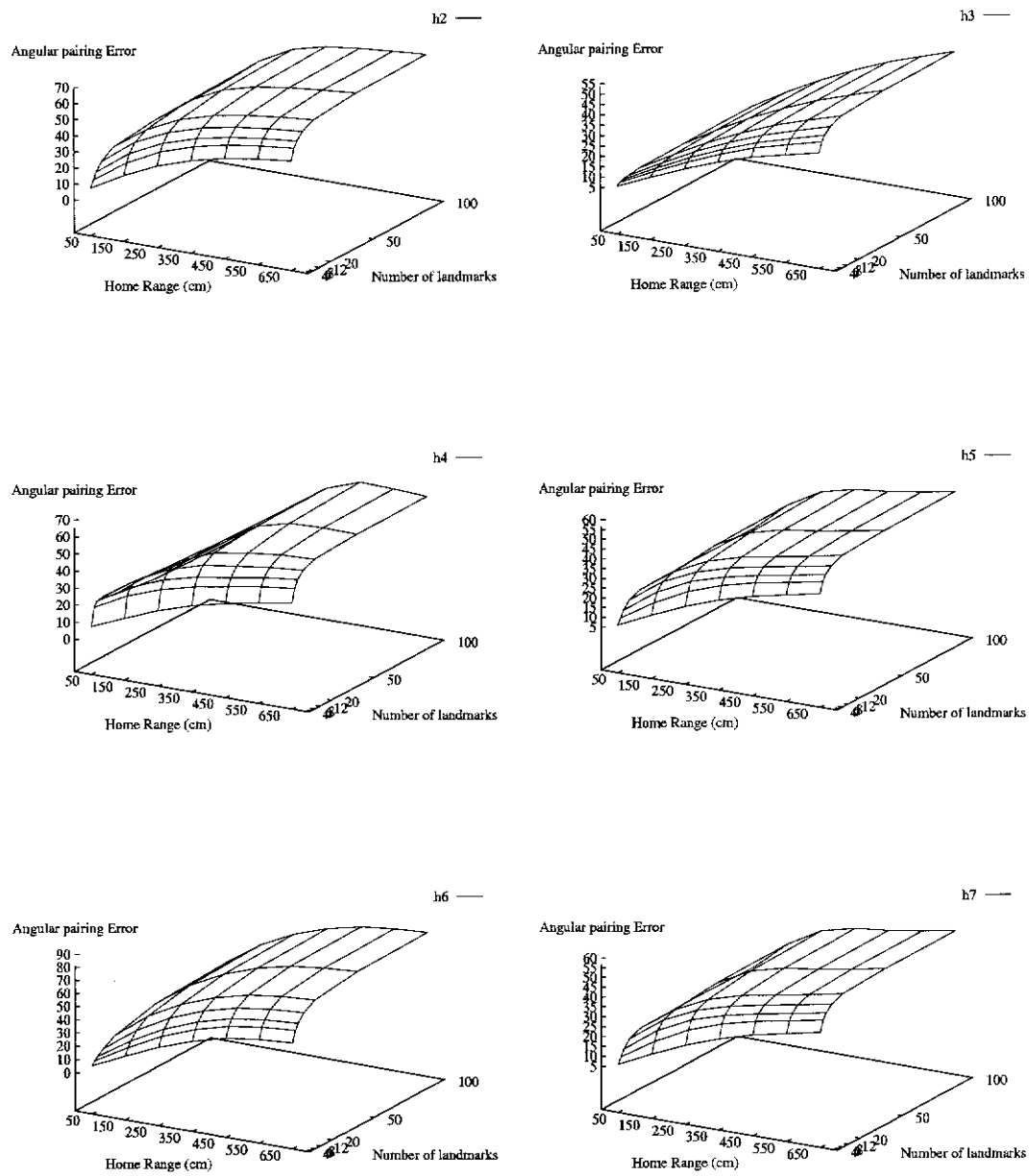


Figure B.18: Average Angular-Pairing Error (simulation, h2–h7). The angular-pairing error is calculated from bearing disparities observed between the home landmarks ($theta_i$) that have been paired and the true corresponding home landmarks ($theta_{\phi(p(i))}$) that should ideally have been paired. This error provides a measure of how well the two sets of landmarks have been paired with each other ($\sqrt{av(|\dot{\theta}_i - \dot{\theta}_{\phi(p(i))}|^2)}$), and is graphed for the correspondence methods h2, h3, h4, h5, h6, and h7. The graphs show how this error is affected by both homing range and landmark numbers, for each method.

Number of landmarks	Range (cm)	Average home-vector error ($\sqrt{av(\|\gamma-\phi\|^2)}$)										
		h0	h1	h2	h3	h3u	h4	h5	h5u	h6	h7	h7u
4	0-100	25.78	25.22	26.80	25.22	29.08	25.72	25.29	29.11	26.70	32.01	34.54
4	100-200	28.65	27.07	32.03	27.07	29.86	27.92	27.34	30.09	32.99	34.26	40.23
4	200-300	32.22	29.92	36.55	29.92	31.64	31.16	30.37	32.08	39.57	34.07	42.46
4	300-400	35.82	33.28	40.08	33.28	34.24	34.73	33.81	34.74	44.10	32.28	40.15
4	400-500	39.48	36.99	43.21	36.99	37.53	38.48	37.62	38.11	47.34	31.39	37.35
4	500-600	43.70	41.35	46.76	41.35	41.55	43.05	42.05	42.16	50.36	34.61	37.51
4	600-700	48.11	45.80	50.57	45.80	45.34	47.57	46.51	45.93	53.54	39.32	39.24
4	0-700	37.54	35.24	41.15	35.24	36.16	36.67	35.80	36.66	44.59	33.26	39.13
6	0-100	22.33	21.16	25.11	21.16	24.96	27.37	21.68	25.61	24.21	29.54	32.37
6	100-200	25.27	22.59	32.27	22.59	25.67	29.16	24.54	28.49	32.40	33.63	41.44
6	200-300	28.47	24.91	35.28	24.91	27.17	31.12	27.38	30.04	38.68	33.42	44.02
6	300-400	31.41	27.73	37.17	27.73	29.42	33.49	30.03	31.53	41.88	29.49	39.93
6	400-500	34.56	31.05	39.31	31.05	32.48	36.14	32.89	33.86	44.25	25.95	34.40
6	500-600	38.47	35.18	42.34	35.18	36.61	40.27	36.69	37.55	47.06	27.56	32.72
6	600-700	42.80	39.46	45.91	39.46	40.70	44.39	40.87	41.48	50.12	31.53	33.27
6	0-700	33.01	29.66	38.10	29.66	31.41	35.12	31.55	33.12	42.17	29.43	37.66
8	0-100	20.09	18.35	27.42	18.35	21.76	28.27	21.61	25.87	23.42	28.70	31.97
8	100-200	23.20	19.62	33.18	19.62	22.49	29.27	24.64	29.20	32.85	33.73	42.92
8	200-300	26.03	21.74	34.16	21.74	24.10	31.56	26.52	28.84	37.89	33.48	46.23
8	300-400	28.57	24.33	35.32	24.33	26.46	33.79	28.35	29.73	40.23	29.04	40.48
8	400-500	31.49	27.49	37.11	27.49	29.62	35.84	30.50	31.65	42.40	24.03	32.77
8	500-600	35.20	31.35	39.69	31.35	33.65	39.77	33.74	35.01	44.87	24.35	29.90
8	600-700	39.34	35.36	43.06	35.36	37.71	43.27	37.52	38.71	47.64	27.11	29.53
8	0-700	30.13	26.20	36.39	26.20	28.47	35.03	29.62	31.37	40.62	28.17	37.41
12	0-100	17.66	—	31.82	15.16	18.19	25.98	21.35	28.84	23.60	27.75	32.73
12	100-200	20.33	—	32.98	15.90	18.90	28.54	23.61	28.45	32.72	37.47	49.57
12	200-300	22.45	—	32.65	17.49	20.28	32.42	25.58	27.74	35.12	36.47	51.13
12	300-400	24.48	—	33.25	19.66	22.35	35.28	26.88	28.05	36.90	29.68	41.92
12	400-500	27.03	—	34.25	22.42	25.16	36.82	28.03	28.88	39.04	22.70	30.90
12	500-600	30.48	—	36.29	25.90	28.85	39.98	30.46	31.48	41.47	20.94	26.16
12	600-700	34.56	—	39.19	29.71	32.83	42.38	33.46	34.70	44.28	21.56	24.26
12	0-700	26.01	—	34.18	21.42	24.21	35.64	27.53	29.22	37.69	28.34	38.48
20	0-100	15.26	—	36.57	11.69	14.20	20.09	18.88	29.33	25.46	30.28	38.37
20	100-200	16.88	—	33.30	11.85	14.78	26.22	23.27	29.30	29.79	45.48	59.68
20	200-300	18.12	—	33.15	12.87	15.90	32.78	26.30	28.80	30.85	42.60	57.97
20	300-400	19.64	—	32.63	14.51	17.60	37.30	27.27	27.61	32.43	31.88	43.95
20	400-500	21.78	—	32.05	16.82	20.01	38.87	26.93	26.45	34.38	22.51	29.01
20	500-600	24.82	—	33.16	19.78	23.27	41.56	28.15	27.96	36.57	19.69	22.99
20	600-700	28.51	—	35.12	23.17	26.76	42.57	29.82	30.28	39.10	17.28	19.37
20	0-700	21.10	—	32.95	16.12	19.27	36.76	26.71	27.89	33.37	30.97	41.08
50	0-100	10.75	—	39.28	7.18	9.27	13.85	16.92	36.94	22.28	48.64	60.65
50	100-200	11.32	—	36.72	6.96	9.71	22.58	25.06	39.18	22.91	67.23	79.59
50	200-300	12.00	—	36.35	7.42	10.43	31.78	29.24	35.19	23.73	53.50	68.86
50	300-400	13.03	—	33.84	8.44	11.53	39.25	29.63	29.42	24.83	36.70	48.99
50	400-500	14.55	—	31.23	10.07	13.17	42.17	27.77	25.09	26.36	25.03	29.07
50	500-600	16.81	—	31.01	12.23	15.50	43.85	27.63	24.88	28.12	23.09	22.87
50	600-700	19.78	—	31.58	14.86	18.22	43.04	27.50	25.36	30.33	16.52	16.30
50	0-700	14.17	—	33.47	9.71	12.74	38.20	27.88	29.78	25.70	39.16	48.36
100	0-100	8.10	—	39.43	5.12	7.07	11.59	16.28	44.34	18.08	71.37	80.04
100	100-200	8.32	—	38.86	4.79	7.38	20.98	25.92	47.21	18.05	77.54	88.36
100	200-300	8.70	—	38.63	5.00	7.80	30.80	30.90	40.44	18.44	59.65	76.23
100	300-400	9.40	—	35.28	5.69	8.48	39.36	31.21	31.07	19.21	40.77	54.69
100	400-500	10.54	—	32.00	6.88	9.56	43.62	28.98	25.46	20.49	27.62	31.68
100	500-600	12.26	—	31.03	8.50	11.25	44.69	28.30	24.48	22.03	27.02	25.52
100	600-700	14.65	—	30.57	10.53	13.30	42.94	27.21	23.86	24.01	19.93	17.95
100	0-700	10.31	—	34.59	6.68	9.34	38.60	29.02	32.49	20.05	44.93	54.16

Table B.2: Home-Vector Error Data

Number of landmarks	Range (cm)	Average homing-vector error ($\sqrt{av(\|\gamma-\varphi\|^2)}$)							
		h0	h1	h2	h3	h4	h5	h6	h7
4	0-100	0.00	5.00	9.55	5.00	7.81	6.36	4.66	24.13
4	100-200	0.00	9.56	18.24	9.56	13.69	12.56	10.80	29.79
4	200-300	0.00	13.00	23.44	13.00	18.78	17.24	14.96	32.01
4	300-400	0.00	15.20	25.67	15.20	22.15	19.97	16.11	31.26
4	400-500	0.00	16.68	26.21	16.68	24.06	21.58	15.23	29.29
4	500-600	0.00	17.92	26.21	17.92	26.43	22.78	12.86	28.34
4	600-700	0.00	19.34	26.80	19.34	28.43	24.02	10.06	27.90
4	0-700	0.00	15.37	24.66	15.37	22.36	19.88	14.21	29.87
6	0-100	0.00	6.00	15.12	6.00	20.54	10.36	6.78	25.57
6	100-200	0.00	10.28	25.89	10.28	24.94	18.56	14.85	33.20
6	200-300	0.00	13.32	29.19	13.32	27.97	22.95	18.74	35.38
6	300-400	0.00	14.89	29.43	14.89	29.57	24.60	18.67	33.45
6	400-500	0.00	15.85	28.61	15.85	30.21	24.83	16.69	30.50
6	500-600	0.00	16.81	28.04	16.81	32.41	25.35	14.37	28.56
6	600-700	0.00	18.33	28.22	18.33	33.61	26.35	11.43	26.79
6	0-700	0.00	14.88	28.20	14.88	29.55	23.78	16.55	31.65
8	0-100	0.00	6.75	22.27	6.75	25.04	16.15	9.09	27.29
8	100-200	0.00	10.84	30.43	10.84	28.68	23.49	17.91	35.31
8	200-300	0.00	13.37	30.93	13.37	31.59	26.03	20.37	36.77
8	300-400	0.00	14.36	30.18	14.36	33.00	26.45	19.26	34.22
8	400-500	0.00	15.00	28.96	15.00	33.12	25.90	17.02	30.36
8	500-600	0.00	15.86	28.31	15.86	35.55	26.09	14.62	27.98
8	600-700	0.00	17.31	28.60	17.31	36.32	26.97	11.77	25.23
8	0-700	0.00	14.34	29.39	14.34	32.80	25.65	17.44	32.16
12	0-100	0.00	—	29.89	7.49	24.79	19.46	12.83	28.68
12	100-200	0.00	—	32.41	10.85	30.12	24.95	20.92	39.95
12	200-300	0.00	—	31.50	12.39	34.69	27.23	20.07	39.77
12	300-400	0.00	—	30.56	12.99	36.97	27.47	18.29	34.58
12	400-500	0.00	—	29.17	13.29	36.97	26.60	16.61	29.78
12	500-600	0.00	—	28.66	14.03	39.08	26.76	14.70	26.97
12	600-700	0.00	—	28.79	15.25	39.09	27.17	12.34	23.29
12	0-700	0.00	—	30.10	12.95	36.15	26.60	17.47	32.99
20	0-100	0.00	—	36.48	8.10	20.02	19.11	17.81	32.47
20	100-200	0.00	—	33.64	10.11	28.25	25.08	19.99	47.68
20	200-300	0.00	—	32.91	10.82	35.38	28.06	17.95	44.66
20	300-400	0.00	—	31.37	11.09	39.68	28.43	16.95	35.24
20	400-500	0.00	—	29.21	11.24	40.27	27.06	15.90	28.38
20	500-600	0.00	—	28.64	11.81	42.60	27.01	14.48	25.94
20	600-700	0.00	—	28.93	12.81	42.19	27.30	12.45	21.44
20	0-700	0.00	—	30.89	11.13	38.39	27.14	16.61	34.64
50	0-100	0.00	—	39.77	6.93	14.53	17.74	17.05	50.02
50	100-200	0.00	—	37.36	7.62	23.84	26.39	15.29	67.89
50	200-300	0.00	—	36.88	7.84	33.16	30.56	14.61	54.01
50	300-400	0.00	—	34.08	7.82	40.65	30.82	14.21	37.81
50	400-500	0.00	—	30.89	7.71	43.28	28.74	13.77	27.53
50	500-600	0.00	—	29.99	8.02	44.96	28.27	12.95	26.49
50	600-700	0.00	—	29.72	8.57	43.66	27.70	11.50	20.62
50	0-700	0.00	—	33.38	7.82	39.37	28.89	14.04	40.56
100	0-100	0.00	—	39.70	5.59	12.11	16.86	13.54	71.87
100	100-200	0.00	—	39.25	5.89	21.61	26.65	12.20	77.66
100	200-300	0.00	—	39.00	5.86	31.42	31.60	11.76	59.81
100	300-400	0.00	—	35.54	5.75	40.00	31.86	11.57	41.11
100	400-500	0.00	—	32.01	5.64	44.12	29.56	11.48	28.59
100	500-600	0.00	—	30.70	5.84	45.18	28.77	11.10	28.59
100	600-700	0.00	—	29.84	6.24	43.14	27.64	9.97	22.72
100	0-700	0.00	—	34.69	5.79	39.13	29.63	11.56	45.46

Table B.3: Homing-Vector Error Data

Number of landmarks	Range (cm)	Average angular-pairing error ($\sqrt{av(\|\theta_i - \theta_{\phi(p(i))}\|^2)}$)							
		h0	h1	h2	h3	h4	h5	h6	h7
4	0-100	0.00	7.09	9.45	7.09	9.16	7.61	7.69	7.00
4	100-200	0.00	14.14	20.77	14.14	17.42	15.62	19.15	15.80
4	200-300	0.00	20.96	30.37	20.96	25.31	23.05	30.28	24.20
4	300-400	0.00	27.05	37.32	27.05	31.55	29.12	39.19	31.19
4	400-500	0.00	32.29	42.38	32.29	36.18	34.08	45.90	36.88
4	500-600	0.00	36.07	45.52	36.07	39.98	37.70	49.45	40.56
4	600-700	0.00	39.99	48.00	39.99	43.24	41.35	51.35	43.45
4	0-700	0.00	28.90	38.13	28.90	32.76	30.63	40.52	32.80
6	0-100	0.00	8.20	13.61	8.20	19.55	10.52	10.10	7.89
6	100-200	0.00	16.06	27.70	16.06	29.62	21.17	24.43	17.89
6	200-300	0.00	23.53	38.14	23.53	36.78	29.93	37.27	27.35
6	300-400	0.00	30.12	45.00	30.12	41.77	35.93	46.95	34.96
6	400-500	0.00	35.95	49.72	35.95	45.32	40.22	53.90	41.05
6	500-600	0.00	40.25	52.70	40.25	48.67	43.44	57.72	45.15
6	600-700	0.00	44.50	54.87	44.50	51.06	46.86	59.71	48.08
6	0-700	0.00	32.23	45.30	32.23	42.27	36.65	47.98	36.61
8	0-100	0.00	8.78	17.51	8.78	22.34	13.58	12.26	8.64
8	100-200	0.00	17.14	32.62	17.14	33.71	26.01	28.24	19.22
8	200-300	0.00	24.98	42.59	24.98	41.66	34.33	41.87	29.29
8	300-400	0.00	31.77	49.08	31.77	46.73	39.48	51.68	37.32
8	400-500	0.00	37.82	53.58	37.82	49.73	43.20	58.65	43.66
8	500-600	0.00	42.43	56.38	42.43	53.19	46.33	62.17	47.66
8	600-700	0.00	46.96	58.42	46.96	55.44	49.69	63.96	50.41
8	0-700	0.00	33.99	49.23	33.99	46.78	39.96	52.41	38.88
12	0-100	0.00	—	22.37	9.26	23.05	17.26	15.58	9.66
12	100-200	0.00	—	37.78	18.17	36.14	29.74	33.99	21.30
12	200-300	0.00	—	47.43	26.48	45.97	37.62	47.89	31.93
12	300-400	0.00	—	53.70	33.64	51.85	42.61	57.62	40.30
12	400-500	0.00	—	57.87	39.92	54.60	46.10	64.39	46.75
12	500-600	0.00	—	60.56	44.85	57.42	49.26	67.81	50.72
12	600-700	0.00	—	62.52	49.56	58.99	52.52	69.72	53.36
12	0-700	0.00	—	53.64	35.93	51.15	42.98	58.04	41.71
20	0-100	0.00	—	27.19	10.07	20.61	19.67	20.71	10.95
20	100-200	0.00	—	41.68	19.25	35.17	31.82	40.20	23.30
20	200-300	0.00	—	51.50	27.75	47.53	40.01	53.63	34.46
20	300-400	0.00	—	57.62	35.15	55.54	44.95	63.05	43.20
20	400-500	0.00	—	61.33	41.64	58.72	48.11	69.56	49.74
20	500-600	0.00	—	63.91	46.84	61.06	51.33	72.86	53.80
20	600-700	0.00	—	65.88	51.75	61.94	54.73	74.70	56.23
20	0-700	0.00	—	57.27	37.54	54.19	45.12	63.26	44.48
50	0-100	0.00	—	29.91	10.42	17.19	20.27	28.39	12.57
50	100-200	0.00	—	45.48	19.81	31.80	34.00	46.52	25.23
50	200-300	0.00	—	55.93	28.46	46.18	43.07	59.40	36.57
50	300-400	0.00	—	61.57	35.95	57.98	47.47	68.41	45.91
50	400-500	0.00	—	64.55	42.60	63.09	49.85	74.47	53.00
50	500-600	0.00	—	66.69	47.99	63.82	52.79	77.56	57.91
50	600-700	0.00	—	68.53	53.07	63.44	56.04	79.34	60.32
50	0-700	0.00	—	60.71	38.44	56.29	47.14	68.37	47.51
100	0-100	0.00	—	30.37	10.59	15.63	20.00	31.22	13.23
100	100-200	0.00	—	46.82	20.02	30.13	34.82	48.94	25.78
100	200-300	0.00	—	57.83	28.65	44.77	44.59	61.61	37.20
100	300-400	0.00	—	63.15	36.17	58.05	48.71	70.36	46.89
100	400-500	0.00	—	65.65	42.88	65.00	50.66	76.20	54.55
100	500-600	0.00	—	67.47	48.31	64.50	53.33	79.12	60.43
100	600-700	0.00	—	69.16	53.48	63.46	56.44	80.82	63.54
100	0-700	0.00	—	61.95	38.70	56.76	48.05	70.20	49.02

Table B.4: Angular-Pairing Error Data

Number of landmarks	Final Home distance (cm) — μ (σ)						
	h0	h1	h2	h3	h4	h5	h6
4	2.610 (7.851)	2.621 (7.853)	2.860 (9.908)	2.621 (7.853)	2.722 (8.598)	2.636 (7.867)	2.963 (12.601)
6	1.714 (3.044)	1.726 (3.116)	3.353 (17.638)	1.726 (3.116)	3.367 (14.582)	1.823 (4.735)	2.526 (16.622)
8	1.471 (1.087)	1.481 (1.117)	4.895 (23.345)	1.481 (1.117)	3.492 (14.121)	1.991 (7.601)	2.580 (20.171)
12	1.417 (0.790)	— (—)	7.098 (24.573)	1.412 (0.822)	3.021 (10.439)	2.352 (8.140)	2.167 (14.393)
20	1.397 (0.721)	— (—)	9.352 (23.092)	1.387 (0.727)	1.821 (3.849)	1.698 (3.196)	1.798 (7.792)
50	1.367 (0.672)	— (—)	7.437 (15.426)	1.343 (0.674)	1.436 (1.022)	1.383 (0.722)	1.427 (1.805)
100	1.333 (0.669)	— (—)	6.822 (13.021)	1.308 (0.665)	1.421 (0.949)	1.413 (0.826)	1.349 (0.676)

Table B.5: Final Homing Distance Statistics (simulation, h0–h6)

Number of landmarks	Final Home distance (cm) — μ (σ)			
	h3u	h5u	h7	h7u
4	7.569 (14.384)	7.616 (14.621)	57.481 (113.351)	67.954 (127.804)
6	3.894 (7.069)	6.540 (26.380)	72.404 (116.299)	96.901 (136.305)
8	2.752 (3.976)	9.879 (36.952)	87.275 (122.340)	122.561 (142.003)
12	1.915 (1.694)	11.628 (34.056)	114.857 (126.891)	170.448 (140.616)
20	1.573 (0.786)	10.910 (28.431)	161.151 (129.013)	221.035 (124.803)
50	1.342 (0.675)	9.353 (22.531)	236.421 (108.155)	278.602 (101.310)
100	1.286 (0.680)	6.144 (16.137)	271.519 (93.414)	305.410 (96.003)

Table B.6: Final Homing Distance Statistics (simulation, h3u,h5u,h7,h7u)

Number of landmarks	Range (cm)	Average home-vector error ($av(\|\gamma-\phi\|)$)										
		h0	h1	h2	h3	h3u	h4	h5	h5u	h6	h7	h7u
4	0-100	20.52	20.23	21.21	20.23	23.81	20.44	20.26	23.82	21.23	23.05	26.86
4	100-200	22.79	21.84	25.36	21.84	24.49	22.34	22.00	24.63	26.46	24.86	30.35
4	200-300	26.19	24.65	29.96	24.65	26.07	25.54	24.97	26.36	32.98	25.43	31.77
4	300-400	29.86	28.00	33.91	28.00	28.40	29.24	28.44	28.82	37.90	24.97	30.52
4	400-500	33.69	31.73	37.45	31.73	31.25	33.17	32.33	31.81	41.48	25.20	29.10
4	500-600	38.11	36.17	41.30	36.17	34.56	37.98	36.90	35.24	44.76	29.02	30.03
4	600-700	42.84	40.83	45.45	40.83	37.45	42.82	41.60	38.17	48.29	34.21	31.88
4	0-700	31.35	29.63	34.82	29.63	29.75	30.90	30.12	30.21	38.12	26.14	30.17
6	0-100	17.31	16.92	19.14	16.92	20.33	19.75	17.17	20.58	18.94	20.52	24.89
6	100-200	19.88	18.25	24.89	18.25	20.98	22.52	19.46	22.20	26.02	23.60	30.79
6	200-300	22.94	20.46	28.80	20.46	22.31	25.38	22.38	23.93	32.48	24.13	32.63
6	300-400	25.94	23.11	31.27	23.11	24.35	28.28	25.16	25.81	36.00	22.09	29.86
6	400-500	29.15	26.26	33.72	26.26	27.10	31.32	28.06	28.22	38.51	20.23	26.19
6	500-600	33.05	30.22	36.90	30.22	30.74	35.67	31.78	31.57	41.45	22.51	25.77
6	600-700	37.45	34.41	40.65	34.41	34.19	39.92	35.93	34.94	44.74	26.72	27.12
6	0-700	27.23	24.62	31.94	24.62	25.86	29.62	26.33	27.06	35.97	22.27	28.39
8	0-100	15.57	14.69	19.32	14.69	17.67	19.46	16.05	19.19	18.02	19.25	24.06
8	100-200	18.16	15.92	25.49	15.92	18.28	22.65	19.22	21.57	26.61	23.18	31.51
8	200-300	20.92	17.81	28.05	17.81	19.67	26.00	21.77	22.70	32.09	23.82	33.82
8	300-400	23.49	20.13	29.70	20.13	21.80	28.79	23.80	24.25	34.57	21.42	29.73
8	400-500	26.38	23.01	31.71	23.01	24.67	31.16	25.92	26.36	36.77	18.50	24.60
8	500-600	30.03	26.64	34.46	26.64	28.31	35.37	29.05	29.56	39.38	19.74	23.45
8	600-700	34.13	30.47	37.96	30.47	31.86	39.02	32.71	32.83	42.31	22.68	24.07
8	0-700	24.71	21.56	30.44	21.56	23.35	29.69	24.67	25.46	34.67	20.86	27.62
12	0-100	13.45	—	21.19	12.15	14.66	18.06	15.47	19.23	17.96	18.17	23.80
12	100-200	15.83	—	25.82	12.87	15.25	22.38	18.97	21.11	27.03	25.44	35.97
12	200-300	18.04	—	27.06	14.24	16.44	27.04	21.32	21.92	29.68	25.59	37.25
12	300-400	20.03	—	28.01	16.09	18.26	30.36	22.66	22.78	31.45	21.47	30.21
12	400-500	22.40	—	29.19	18.46	20.74	32.19	23.73	23.92	33.60	17.02	22.66
12	500-600	25.66	—	31.31	21.61	24.07	35.79	25.99	26.44	36.08	16.68	20.16
12	600-700	29.61	—	34.18	25.20	27.67	38.33	28.81	29.42	39.04	17.70	19.54
12	0-700	21.15	—	28.60	17.37	19.66	30.46	23.01	23.61	32.04	20.18	27.39
20	0-100	11.35	—	24.80	9.36	11.39	14.64	14.35	19.67	19.77	19.38	26.98
20	100-200	13.16	—	26.81	9.52	11.84	20.84	19.34	21.95	24.70	31.06	44.09
20	200-300	14.56	—	28.00	10.37	12.74	27.80	22.40	22.87	25.77	29.36	42.48
20	300-400	15.96	—	27.88	11.71	14.18	32.73	23.25	22.43	27.30	22.48	31.25
20	400-500	17.83	—	27.41	13.62	16.26	34.45	22.86	21.78	29.19	16.73	20.92
20	500-600	20.52	—	28.48	16.15	19.10	37.86	23.95	23.30	31.35	15.44	17.60
20	600-700	23.86	—	30.28	19.13	22.18	38.95	25.41	25.52	33.82	13.82	15.43
20	0-700	16.97	—	27.78	12.83	15.41	31.86	22.57	22.47	28.07	21.10	28.21
50	0-100	8.00	—	28.26	5.72	7.41	10.72	13.82	25.06	17.84	31.95	44.04
50	100-200	8.91	—	31.21	5.54	7.76	18.77	21.86	30.77	18.72	50.30	63.14
50	200-300	9.60	—	31.98	5.91	8.36	27.76	25.91	28.99	19.48	38.62	52.38
50	300-400	10.47	—	29.92	6.72	9.26	35.23	26.16	24.49	20.49	25.95	34.73
50	400-500	11.75	—	27.40	8.04	10.59	38.14	24.20	20.93	21.89	18.79	20.57
50	500-600	13.65	—	27.12	9.82	12.52	41.01	24.03	20.84	23.51	18.88	17.90
50	600-700	16.17	—	27.51	12.02	14.82	40.13	23.80	21.27	25.56	12.46	12.53
50	0-700	11.26	—	29.03	7.58	10.09	33.69	24.30	24.06	21.21	26.64	32.72
100	0-100	6.18	—	28.86	4.09	5.65	9.23	13.79	31.87	14.53	52.42	62.73
100	100-200	6.62	—	33.69	3.82	5.90	17.94	23.11	38.59	14.66	61.42	72.50
100	200-300	6.97	—	34.55	3.97	6.24	27.30	27.72	33.74	15.00	45.07	59.81
100	300-400	7.54	—	31.85	4.52	6.80	35.54	27.91	26.36	15.69	29.43	39.50
100	400-500	8.47	—	28.76	5.49	7.68	39.79	25.66	21.64	16.81	20.85	22.29
100	500-600	9.88	—	27.66	6.80	9.06	42.41	25.01	20.82	18.18	23.33	21.04
100	600-700	11.88	—	26.97	8.47	10.76	40.39	23.85	20.22	19.93	15.28	13.95
100	0-700	8.17	—	30.57	5.19	7.39	34.27	25.67	26.34	16.37	31.58	37.50

Table B.7: Home-Vector Error Data

Number of landmarks	Range (cm)	Average homing-vector error ($av(\|\gamma-\varphi\|)$)							
		h0	h1	h2	h3	h4	h5	h6	h7
4	0-100	0.00	0.48	2.03	0.48	0.80	0.61	1.55	8.61
4	100-200	0.00	1.58	6.19	1.58	2.58	2.19	5.32	15.65
4	200-300	0.00	2.77	9.53	2.77	4.77	4.02	8.55	19.97
4	300-400	0.00	3.74	11.07	3.74	6.56	5.42	9.68	21.21
4	400-500	0.00	4.43	11.25	4.43	7.68	6.35	9.19	20.37
4	500-600	0.00	4.92	10.64	4.92	8.91	6.94	7.60	19.02
4	600-700	0.00	5.50	10.15	5.50	9.87	7.46	5.91	17.53
4	0-700	0.00	3.75	10.01	3.75	6.57	5.35	8.17	19.31
6	0-100	0.00	0.86	4.52	0.86	5.00	1.56	2.93	12.38
6	100-200	0.00	2.43	12.14	2.43	9.41	5.22	9.03	21.26
6	200-300	0.00	3.97	16.20	3.97	12.63	8.56	12.78	25.22
6	300-400	0.00	4.99	16.89	4.99	14.39	10.23	13.08	25.14
6	400-500	0.00	5.65	15.88	5.65	14.98	10.59	11.67	23.38
6	500-600	0.00	6.08	14.62	6.08	16.42	10.78	9.91	21.54
6	600-700	0.00	6.76	13.62	6.76	16.79	11.17	7.94	19.40
6	0-700	0.00	4.90	15.13	4.90	13.95	9.45	11.17	23.02
8	0-100	0.00	1.24	8.18	1.24	8.17	3.54	4.51	15.23
8	100-200	0.00	3.23	17.40	3.23	14.06	9.52	12.01	24.17
8	200-300	0.00	4.90	20.26	4.90	18.12	13.10	14.95	26.97
8	300-400	0.00	5.80	20.03	5.80	19.95	14.05	14.42	26.10
8	400-500	0.00	6.33	18.48	6.33	19.97	13.61	12.82	23.74
8	500-600	0.00	6.76	16.85	6.76	21.82	13.43	10.97	21.74
8	600-700	0.00	7.50	15.72	7.50	21.73	13.65	8.91	19.36
8	0-700	0.00	5.66	18.28	5.66	19.12	12.89	12.68	24.01
12	0-100	0.00	—	14.53	1.80	10.49	7.14	7.41	17.79
12	100-200	0.00	—	22.56	4.13	18.35	14.61	15.58	28.13
12	200-300	0.00	—	23.66	5.71	24.10	17.94	15.70	29.19
12	300-400	0.00	—	22.93	6.43	26.73	18.32	14.52	26.17
12	400-500	0.00	—	21.15	6.75	26.51	17.12	13.31	23.24
12	500-600	0.00	—	19.60	7.15	28.32	16.71	11.84	21.22
12	600-700	0.00	—	18.33	7.86	27.36	16.46	10.05	18.50
12	0-700	0.00	—	21.48	6.23	25.22	16.86	13.59	24.57
20	0-100	0.00	—	22.85	2.71	10.50	11.03	12.15	21.37
20	100-200	0.00	—	26.12	5.01	19.50	18.33	15.91	33.99
20	200-300	0.00	—	26.48	6.15	26.96	21.45	14.48	32.16
20	300-400	0.00	—	25.24	6.57	31.51	21.65	13.86	26.01
20	400-500	0.00	—	23.06	6.70	31.94	19.98	13.16	21.93
20	500-600	0.00	—	21.72	7.07	34.25	19.41	12.07	20.31
20	600-700	0.00	—	20.72	7.75	32.81	18.94	10.52	17.01
20	0-700	0.00	—	24.04	6.41	29.60	20.00	13.45	25.08
50	0-100	0.00	—	28.13	3.36	9.77	13.42	13.27	33.78
50	100-200	0.00	—	31.00	4.95	18.59	21.80	12.30	51.40
50	200-300	0.00	—	31.71	5.47	27.48	25.79	11.87	39.55
50	300-400	0.00	—	29.39	5.54	34.85	25.90	11.65	27.35
50	400-500	0.00	—	26.19	5.45	37.49	23.58	11.41	21.07
50	500-600	0.00	—	24.80	5.69	39.91	22.82	10.83	21.06
50	600-700	0.00	—	23.86	6.10	37.33	21.81	9.78	15.83
50	0-700	0.00	—	27.95	5.43	33.02	23.74	11.51	28.42
100	0-100	0.00	—	28.71	3.30	8.98	13.69	10.78	53.31
100	100-200	0.00	—	33.48	4.28	17.77	23.05	9.84	61.69
100	200-300	0.00	—	34.36	4.43	27.00	27.61	9.53	45.35
100	300-400	0.00	—	31.60	4.38	35.19	27.73	9.43	29.92
100	400-500	0.00	—	28.21	4.27	39.38	25.30	9.43	21.69
100	500-600	0.00	—	26.47	4.43	41.90	24.42	9.18	23.72
100	600-700	0.00	—	25.11	4.72	38.94	23.03	8.37	17.35
100	0-700	0.00	—	30.02	4.34	33.85	25.37	9.44	32.20

Table B.8: Homing-Vector Error Data

Number of landmarks	Range (cm)	Average angular-pairing error ($av(\lvert\theta_i - \theta_{\phi(p(i))}\rvert)$)							
		h0	h1	h2	h3	h4	h5	h6	h7
4	0-100	0.00	0.86	2.16	0.86	1.15	0.92	2.18	1.73
4	100-200	0.00	3.00	7.54	3.00	3.90	3.37	8.21	6.00
4	200-300	0.00	6.02	13.61	6.02	7.70	6.79	15.69	11.15
4	300-400	0.00	9.47	18.60	9.47	11.62	10.42	22.28	15.83
4	400-500	0.00	12.99	22.45	12.99	15.12	13.95	27.39	19.75
4	500-600	0.00	15.92	24.76	15.92	18.24	16.86	29.64	22.26
4	600-700	0.00	19.12	26.53	19.12	21.15	19.90	30.44	24.24
4	0-700	0.00	10.54	18.70	10.54	12.46	11.37	22.29	16.26
6	0-100	0.00	1.22	3.76	1.22	4.70	1.64	3.47	2.42
6	100-200	0.00	4.11	12.11	4.11	10.69	6.00	12.33	8.17
6	200-300	0.00	8.03	20.19	8.03	15.94	11.35	22.20	14.91
6	300-400	0.00	12.36	26.07	12.36	20.11	15.89	30.23	20.68
6	400-500	0.00	16.83	30.23	16.83	23.39	19.58	36.08	25.29
6	500-600	0.00	20.50	32.73	20.50	26.56	22.58	38.73	28.27
6	600-700	0.00	24.35	34.44	24.35	29.09	25.87	39.65	30.37
6	0-700	0.00	13.70	25.76	13.70	20.47	16.38	29.89	21.00
8	0-100	0.00	1.50	5.61	1.50	6.76	2.78	4.76	3.04
8	100-200	0.00	4.95	16.00	4.95	14.75	9.45	15.41	9.71
8	200-300	0.00	9.47	24.70	9.47	21.41	15.64	26.36	17.36
8	300-400	0.00	14.29	30.80	14.29	26.04	19.96	35.03	23.85
8	400-500	0.00	19.22	35.01	19.22	28.87	23.29	41.32	28.91
8	500-600	0.00	23.35	37.42	23.35	32.22	26.29	43.87	31.89
8	600-700	0.00	27.61	39.08	27.61	34.53	29.58	44.68	33.88
8	0-700	0.00	15.73	30.29	15.73	25.83	20.16	34.50	24.03
12	0-100	0.00	—	8.66	1.87	8.73	5.39	6.70	3.93
12	100-200	0.00	—	21.27	6.00	18.96	14.42	19.89	12.18
12	200-300	0.00	—	30.65	11.29	27.86	20.93	31.91	20.94
12	300-400	0.00	—	36.85	16.77	33.43	24.99	41.12	28.08
12	400-500	0.00	—	40.83	22.20	35.94	27.84	47.70	33.40
12	500-600	0.00	—	43.09	26.77	38.61	30.70	50.24	36.36
12	600-700	0.00	—	44.58	31.37	40.09	33.83	51.18	38.24
12	0-700	0.00	—	36.00	18.26	32.25	24.88	40.37	28.00
20	0-100	0.00	—	12.91	2.49	9.21	9.27	9.59	5.19
20	100-200	0.00	—	26.24	7.34	20.85	19.29	24.40	14.80
20	200-300	0.00	—	36.11	13.24	32.09	25.94	36.95	24.53
20	300-400	0.00	—	42.37	19.24	39.80	29.68	46.68	32.29
20	400-500	0.00	—	45.96	25.05	42.54	31.85	53.52	37.71
20	500-600	0.00	—	48.00	29.97	44.62	34.57	55.97	40.68
20	600-700	0.00	—	49.38	34.80	45.05	37.68	56.87	42.29
20	0-700	0.00	—	41.17	20.76	37.61	29.24	45.77	31.91
50	0-100	0.00	—	16.76	3.30	9.44	13.11	13.30	7.14
50	100-200	0.00	—	31.63	8.91	21.48	24.72	28.75	17.69
50	200-300	0.00	—	42.36	15.23	34.30	32.04	42.20	27.88
50	300-400	0.00	—	48.27	21.43	45.34	34.84	52.36	36.47
50	400-500	0.00	—	51.10	27.42	49.81	35.76	59.18	42.47
50	500-600	0.00	—	52.52	32.48	50.06	37.95	61.31	46.14
50	600-700	0.00	—	53.56	37.44	48.72	40.76	62.10	47.38
50	0-700	0.00	—	46.49	22.94	42.37	33.79	51.11	36.18
100	0-100	0.00	—	17.81	3.81	9.34	14.21	14.49	7.99
100	100-200	0.00	—	33.59	9.60	21.32	26.88	30.47	18.45
100	200-300	0.00	—	44.91	15.92	34.31	34.78	44.30	28.78
100	300-400	0.00	—	50.55	22.14	46.54	37.09	54.48	37.90
100	400-500	0.00	—	52.89	28.19	52.77	37.35	61.19	44.65
100	500-600	0.00	—	53.90	33.29	51.75	39.18	63.02	49.34
100	600-700	0.00	—	54.72	38.33	49.54	41.77	63.71	51.07
100	0-700	0.00	—	48.41	23.67	43.77	35.66	53.04	38.05

Table B.9: Angular-Pairing Error Data

Number of landmarks	Range (cm)	Average home-vector error $\sqrt{av(\lvert\dot{\gamma-\phi}\rvert^2)}$				
		0°	5°	10°	20°	50°
10	0-100	16.43	29.97	46.95	69.11	91.38
10	100-200	17.36	18.93	23.64	39.95	75.33
10	200-300	19.10	19.67	21.35	28.17	60.57
10	300-400	21.37	21.66	22.49	25.78	49.72
10	400-500	24.28	24.41	24.79	26.43	42.55
10	500-600	27.88	27.88	27.92	28.42	39.20
10	600-700	31.80	31.74	31.47	30.83	37.42
10	0-700	23.19	23.90	25.61	30.84	52.57
20	0-100	11.69	21.57	34.94	56.15	85.02
20	100-200	11.85	12.83	15.54	25.34	62.25
20	200-300	12.87	13.20	14.19	17.97	43.58
20	300-400	14.51	14.63	15.05	16.92	32.42
20	400-500	16.82	16.83	16.94	17.72	26.83
20	500-600	19.78	19.67	19.52	19.46	24.75
20	600-700	23.17	22.93	22.49	21.42	23.88
20	0-700	16.12	16.54	17.64	21.18	38.25
50	0-100	7.18	13.02	22.52	39.37	74.00
50	100-200	6.96	7.52	9.14	14.25	42.70
50	200-300	7.42	7.59	8.21	10.54	24.82
50	300-400	8.44	8.51	8.81	10.06	18.08
50	400-500	10.07	10.07	10.17	10.68	15.41
50	500-600	12.23	12.15	12.04	11.92	14.55
50	600-700	14.86	14.64	14.26	13.31	14.11
50	0-700	9.71	9.94	10.70	13.14	25.08

Table B.10: Home-Vector Error Data (Bearing Error).

Number of landmarks	Range (cm)	Average homing-vector error ($\sqrt{av(\|\dot{\gamma}-\dot{\varphi}\|^2)}$)				
		0°	5°	10°	20°	50°
10	0-100	7.31	26.83	45.73	69.08	91.56
10	100-200	11.12	13.95	20.80	39.78	76.14
10	200-300	12.95	14.38	17.91	28.11	62.30
10	300-400	13.74	15.11	18.16	25.67	52.69
10	400-500	14.22	15.90	19.27	26.17	47.19
10	500-600	15.04	17.33	21.27	28.22	45.76
10	600-700	16.52	19.70	24.21	31.09	46.49
10	0-700	13.74	16.24	20.94	30.71	55.87
20	0-100	8.10	20.23	34.54	56.41	85.31
20	100-200	10.11	11.73	15.47	26.40	63.12
20	200-300	10.82	11.82	14.05	19.53	45.40
20	300-400	11.09	12.10	14.12	18.45	35.48
20	400-500	11.24	12.54	14.82	19.06	31.50
20	500-600	11.81	13.68	16.44	20.96	31.42
20	600-700	12.81	15.51	18.76	23.59	33.04
20	0-700	11.13	12.93	16.10	22.46	41.34
50	0-100	6.93	13.15	22.86	39.76	74.21
50	100-200	7.62	8.55	10.46	15.52	43.40
50	200-300	7.84	8.40	9.50	12.22	26.31
50	300-400	7.82	8.38	9.41	11.69	20.54
50	400-500	7.71	8.52	9.81	12.25	19.09
50	500-600	8.02	9.31	10.99	13.75	19.83
50	600-700	8.57	10.63	12.78	15.90	21.56
50	0-700	7.82	8.90	10.76	14.58	27.33

Table B.11: Homing-Vector Error Data (Bearing Error).

Number of landmarks	Range (cm)	Average angular-pairing error ($\sqrt{av(\lvert\theta_i - \theta_{\phi(p(i))}\rvert^2)}$)				
		0°	5°	10°	20°	50°
10	0-100	28.75	31.84	43.47	83.90	221.91
10	100-200	56.22	59.25	69.24	105.95	230.66
10	200-300	81.98	85.52	96.20	132.77	243.21
10	300-400	104.34	108.75	121.25	159.74	257.10
10	400-500	123.99	129.65	144.40	185.00	269.82
10	500-600	139.44	146.65	164.00	206.05	279.37
10	600-700	154.39	163.50	183.02	225.26	286.68
10	0-700	111.58	116.93	130.80	169.53	260.47
20	0-100	45.05	51.62	72.48	134.79	316.96
20	100-200	86.11	91.91	108.81	162.93	330.51
20	200-300	124.12	130.25	147.06	198.19	349.76
20	300-400	157.21	164.29	182.45	234.53	370.73
20	400-500	186.21	194.64	214.98	269.85	390.47
20	500-600	209.47	219.71	242.77	300.08	405.21
20	600-700	231.44	243.72	269.40	327.83	416.70
20	0-700	167.88	175.94	195.48	248.75	376.19
50	0-100	73.67	88.28	125.44	218.28	500.71
50	100-200	140.05	150.84	178.85	259.95	523.16
50	200-300	201.21	211.39	237.30	314.92	555.89
50	300-400	254.22	265.12	292.22	372.35	591.22
50	400-500	301.23	313.64	343.72	429.11	624.43
50	500-600	339.37	353.91	388.04	478.25	648.83
50	600-700	375.28	392.48	430.96	524.07	667.99
50	0-700	271.82	283.91	313.28	395.91	600.35

Table B.12: Angular-Pairing Error Data (Bearing Error).

Number of landmarks	Final Home distance (cm) — μ (σ)				
	0°	5°	10°	20°	50°
10	1.472 (1.210)	18.607 (11.363)	36.891 (22.595)	73.795 (45.068)	196.748 (118.974)
20	1.387 (0.727)	12.349 (7.171)	23.860 (13.830)	47.803 (27.702)	133.439 (75.348)
50	1.343 (0.674)	7.773 (4.241)	14.538 (8.038)	28.650 (15.747)	82.175 (45.179)

Table B.13: Final Homing Distance Statistics (Bearing Error).

Number of landmarks	Range (cm)	Average home-vector error ($av(\|\gamma-\phi\|)$)				
		0°	5°	10°	20°	50°
10	0-100	13.12	21.48	33.82	53.15	75.91
10	100-200	13.98	15.27	18.73	29.83	59.52
10	200-300	15.52	15.99	17.34	22.23	46.17
10	300-400	17.54	17.78	18.43	20.86	37.47
10	400-500	20.09	20.19	20.46	21.62	32.32
10	500-600	23.36	23.35	23.33	23.50	30.17
10	600-700	27.16	27.08	26.71	25.79	29.11
10	0-700	18.87	19.41	20.54	23.84	39.13
20	0-100	9.36	14.99	23.95	41.07	69.10
20	100-200	9.52	10.31	12.38	19.11	47.18
20	200-300	10.37	10.63	11.39	14.27	32.12
20	300-400	11.71	11.80	12.13	13.57	24.38
20	400-500	13.62	13.62	13.70	14.28	20.69
20	500-600	16.15	16.05	15.90	15.78	19.38
20	600-700	19.13	18.91	18.51	17.48	18.83
20	0-700	12.83	13.12	13.78	15.83	27.08
50	0-100	5.72	8.98	14.62	26.94	57.80
50	100-200	5.54	5.99	7.25	11.12	31.04
50	200-300	5.91	6.05	6.55	8.40	18.72
50	300-400	6.72	6.78	7.03	8.04	14.15
50	400-500	8.04	8.04	8.13	8.55	12.20
50	500-600	9.82	9.75	9.66	9.56	11.57
50	600-700	12.02	11.83	11.51	10.69	11.23
50	0-700	7.58	7.74	8.17	9.51	16.76

Table B.14: Home-Vector Error Data (Bearing Error).

Number of landmarks	Range (cm)	Average homing-vector error ($av(\lvert\gamma-\varphi\rvert)$)				
		0°	5°	10°	20°	50°
10	0-100	1.54	16.56	31.09	52.14	75.69
10	100-200	3.75	8.62	14.60	28.64	60.00
10	200-300	5.40	8.28	12.19	21.05	47.68
10	300-400	6.26	8.52	12.01	19.44	40.10
10	400-500	6.69	8.91	12.61	19.82	36.32
10	500-600	7.13	9.69	13.94	21.43	35.75
10	600-700	7.92	11.05	15.92	23.66	36.81
10	0-700	6.08	9.17	13.56	22.24	42.39
20	0-100	2.71	12.58	23.01	41.10	69.26
20	100-200	5.01	7.70	11.45	19.81	48.01
20	200-300	6.15	7.68	10.25	15.31	33.82
20	300-400	6.57	7.89	10.28	14.51	27.05
20	400-500	6.70	8.21	10.83	15.02	24.63
20	500-600	7.07	9.03	12.10	16.55	24.94
20	600-700	7.75	10.40	13.95	18.75	26.51
20	0-700	6.41	8.39	11.42	16.67	30.41
50	0-100	3.36	8.62	14.91	27.36	57.98
50	100-200	4.95	6.21	8.13	12.15	31.73
50	200-300	5.47	6.23	7.43	9.71	20.03
50	300-400	5.54	6.27	7.38	9.30	16.17
50	400-500	5.45	6.41	7.72	9.77	15.20
50	500-600	5.69	7.06	8.68	11.00	15.88
50	600-700	6.10	8.16	10.13	12.77	17.35
50	0-700	5.43	6.59	8.14	10.79	19.32

Table B.15: Homing-Vector Error Data (Bearing Error).

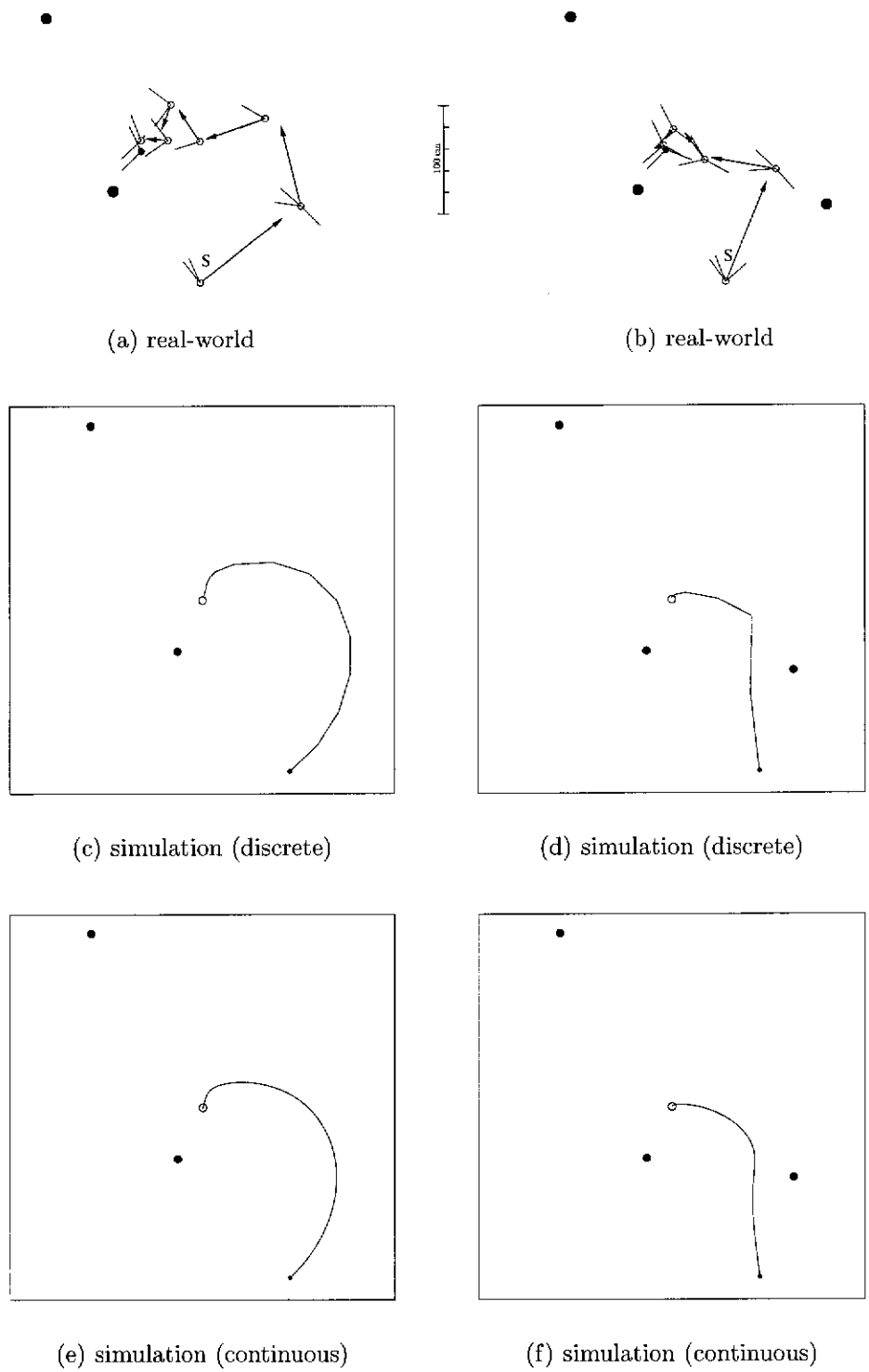
Number of landmarks	Range (cm)	Average angular-pairing error ($av(\lvert\theta_i - \theta_{\phi(p(i))}\rvert)$)				
		0°	5°	10°	20°	50°
10	0-100	1.69	2.80	5.67	14.91	51.00
10	100-200	5.52	6.66	9.66	19.23	53.50
10	200-300	10.49	11.79	15.12	25.30	57.27
10	300-400	15.73	17.29	21.18	32.25	61.68
10	400-500	20.98	22.90	27.49	39.41	65.91
10	500-600	25.46	27.87	33.23	45.76	69.22
10	600-700	30.01	33.06	39.15	51.85	71.82
10	0-700	17.23	19.01	23.23	34.52	62.77
20	0-100	2.49	4.38	8.65	20.22	54.26
20	100-200	7.34	9.11	13.20	24.27	56.90
20	200-300	13.24	15.03	19.16	30.04	60.86
20	300-400	19.24	21.19	25.57	36.70	65.40
20	400-500	25.05	27.29	32.08	43.73	69.93
20	500-600	29.97	32.64	37.98	50.12	73.44
20	600-700	34.80	37.99	43.90	56.29	76.29
20	0-700	20.76	22.92	27.58	39.05	66.63
50	0-100	3.30	6.26	11.51	22.80	55.89
50	100-200	8.91	11.20	15.75	26.49	58.58
50	200-300	15.23	17.24	21.45	32.05	62.71
50	300-400	21.43	23.45	27.72	38.62	67.45
50	400-500	27.42	29.63	34.23	45.72	72.18
50	500-600	32.48	35.07	40.16	52.22	75.81
50	600-700	37.44	40.50	46.16	58.57	78.76
50	0-700	22.94	25.21	29.82	41.10	68.72

Table B.16: Angular-Pairing Error Data (Bearing Error).

B.3 Mobile Robot Homing and Simulation Comparisons

This section presents a more comprehensive comparison of the homing behaviour observed in the mobile robot experiments against identical simulated versions. For each of the real-world homing runs presented in §6.5.2 the equivalent continuous and discrete simulated homing version is shown.

Figures B.19, B.20, and B.21 demonstrate that the real-world robot homing compares favourably with the simulated versions. In each case essentially the same general homing behaviour can be observed. The slight differences between the actual homing path taken by the robot and that observed in simulation, is primarily due to the occasional erroneous observation of landmarks. This is most apparent in figure B.21, where the robot homing path is significantly different to that generated by the simulation. At each snapshot position, the direction of a sensed landmark is noted. Due to the spasmodic identification of erroneous landmarks, in addition to the occasional failure to observe legitimate landmarks, the computed homing vector is not as accurate as those generated in simulation. However, despite this the homing algorithm was able to produce homing vectors accurate enough to incrementally improve the robots position and thus successfully home.

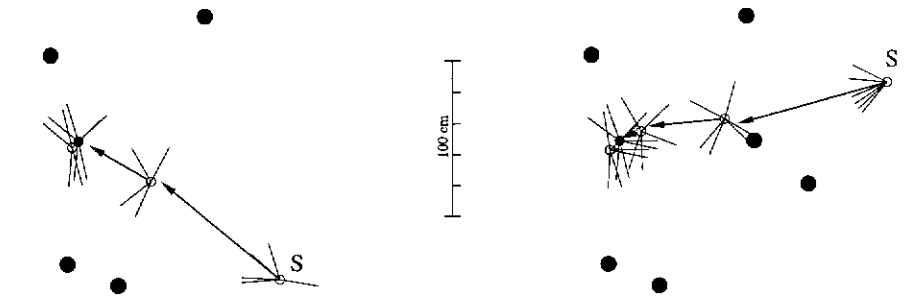


(a) real-world (b) real-world

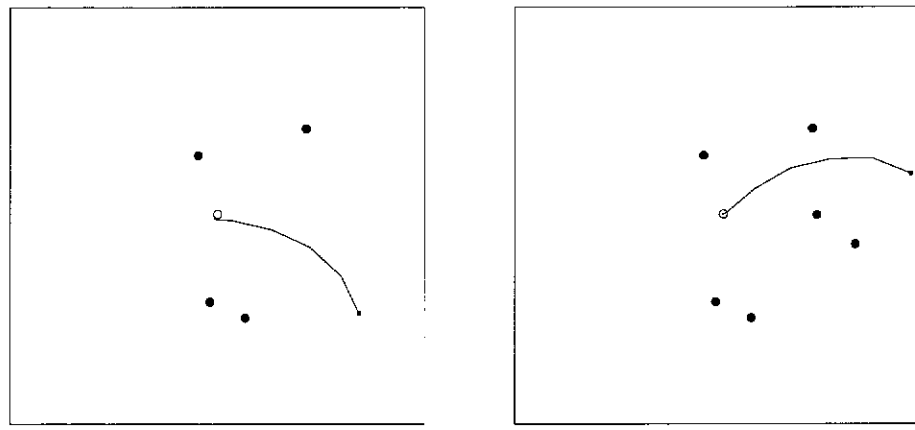
(c) simulation (discrete) (d) simulation (discrete)

(e) simulation (continuous) (f) simulation (continuous)

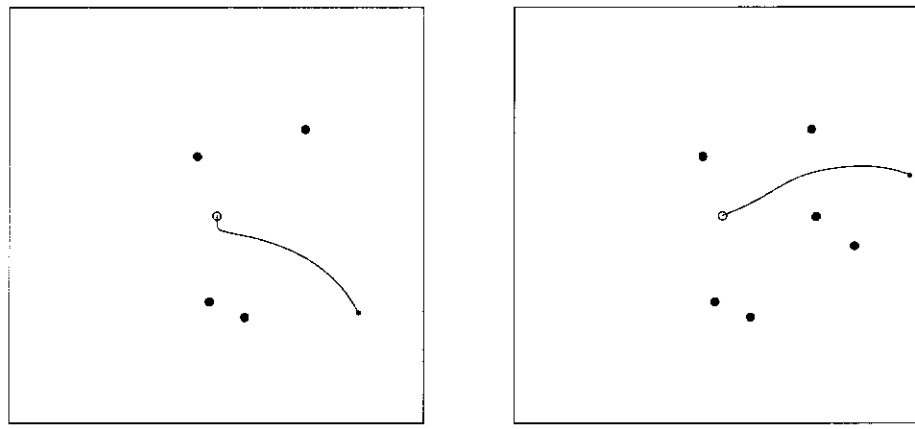
Figure B.19: Mobile Robot Homing and Simulation Comparisons. The real-world homing experiments using the mobile robot (a, b) compare well with identical scenarios run in simulation (c–f).



(a) real-world (b) real-world

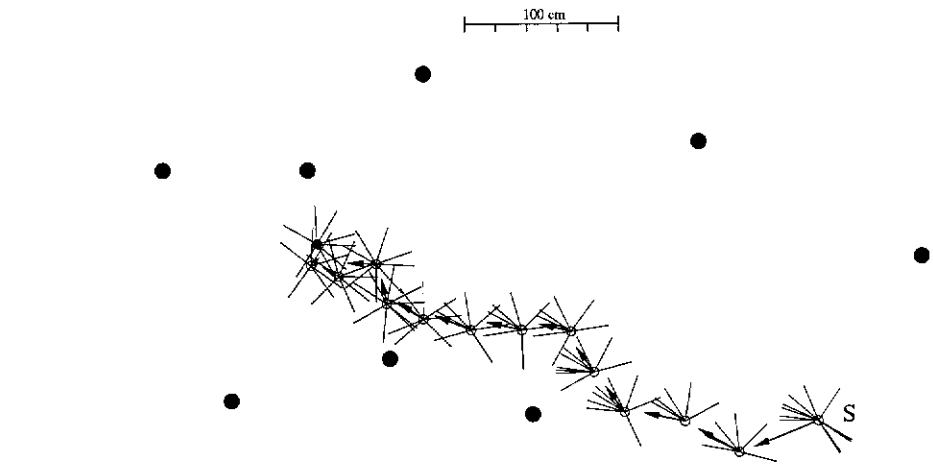


(c) simulation (discrete) (d) simulation (discrete)

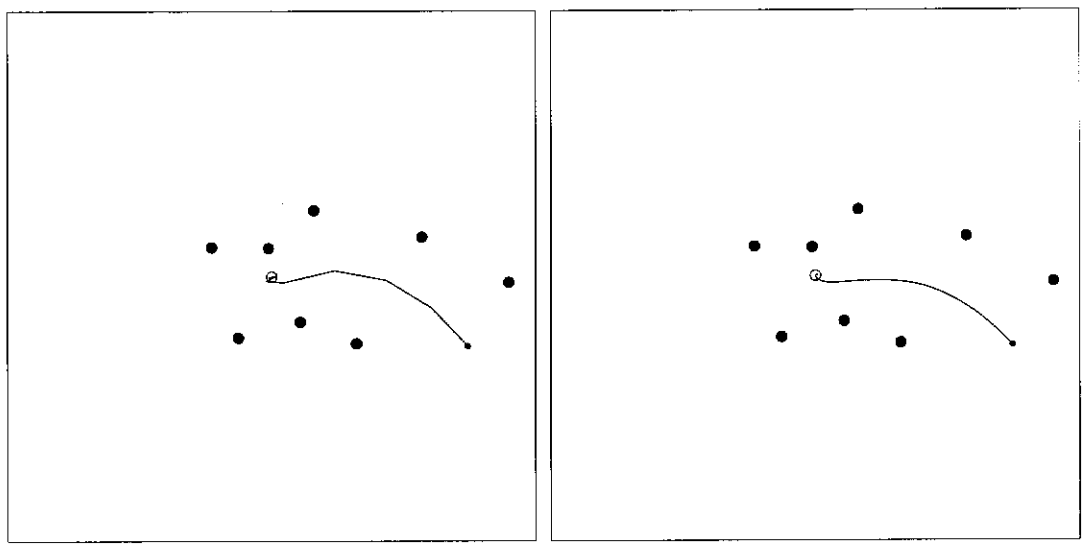


(e) simulation (continuous) (f) simulation (continuous)

Figure B.20: Mobile Robot Homing and Simulation Comparisons. The real-world homing experiments using the mobile robot (a, b) compare well with identical scenarios run in simulation (c–f).



(a)



(b) simulation (discrete) (c) simulation (continuous)

Figure B.21: Mobile Robot Homing and Simulation Comparison. The real-world homing experiment using the mobile robot (a) compares reasonably well with identical scenarios run in simulation (b, c). Path differences are due to the erroneous perception of landmarks.

Bibliography

Adelson, E. H. and Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A: Optics and Image Science*, 2(2):284–299.

Adiv, G. (1985). Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(4):384–401.

Adiv, G. (1989). Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–489.

Aggarwal, J. K. and Nandhakumar, N. (1988). On the computation of motion from sequences of images—a review. *Proceedings of the IEEE*, 76(8):917–935.

Agre, P. E. and Chapman, D. (1987). Pengi: an implementation of a theory of activity. In *Proceedings of the National Conference on Artificial Intelligence*, pages 268–272, Seattle, Washington.

Aisbett, J. (1989). Optical flow with an intensity-weighted smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):512–522.

Aloimonos, Y. (1990). Purposive and qualitative active vision. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 346–360. IEEE Computer Society Press.

Aloimonos, Y. (1992). Is visual reconstruction necessary? *Journal of Robotic Systems*, 9(6):843–858.

Aloimonos, Y., editor (1993). *Active Perception.* Laurence Erlbaum, New Jersey.

Aloimonos, Y. and Rosenfeld, A. (1991). Computer vision. *Science*, 253:1249–1254.

Aloimonos, Y., Weiss, I., and Banddophaday, A. (1988). Active vision. *International Journal of Computer Vision*, 1(4):333–356.

Anandan, P. (1989). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*,

2:283–310.

Ancona, N. (1992). A fast obstacle detection method based on optical flow. In Sandini, G., editor, *Proceedings of the European Conference on Computer Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 267–271. Springer-Verlag, Berlin.

Anderson, T. L. and Donath, M. (1990). Animal behavior as a paradigm for developing robot autonomy. *Robotics and Autonomous Systems*, 6:145–168. Reprint available in (Maes, 1990).

Anstis, S. M. (1980). The perception of apparent movement. *Philosophical Transactions of the Royal Society of London B*, 290:153–168.

Arkin, R. C. (1987a). Motor schema based navigation for a mobile robot: an approach to programming by behaviour. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 264–271, Raleigh, NC.

Arkin, R. C. (1987b). Reflexive/reactive navigation for an autonomous vehicle. In *Proceedings of the AIAA Computers in Aerospace VI*, pages 298–306, Wakefield, Massachusetts.

Arkin, R. C. (1989a). Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112.

Arkin, R. C. (1989b). Towards the unification of navigational planning and reactive control. In *Working Notes of the AAAI Spring Symposium on Robot Navigation*, Stanford University.

Arkin, R. C. (1990). Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105–122. Reprint available in (Maes, 1990).

Arkin, R. C., Balch, T., and Nitz, E. (1993). Communication of behavioral state in multi-agent retrieval tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 588–594. IEEE Computer Society Press.

Asteroth, A., Fischer, M. S., Möller, K., and Schnepf, U. (1992). Tracking and grasping of moving objects: a behaviour-based approach. In *Proceedings of the International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 195–204.

Azuma, A., Azuma, S., Watanabe, I., and Furuta, T. (1985). Flight mechanics of a dragonfly. *The Journal of Experimental Biology*, 116:79–107.

Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE*, 76(8):996–1005.

Ballard, D. H. (1991). Animate vision. *Artificial Intelligence*, 48:57–86.

Ballard, D. H. and Brown, C. M. (1992). Principles of animat vision. *Computer Vision, Graphics, and Image Processing*, 56(1):3–21.

Barnard, S. T. and Thompson, W. B. (1980). Disparity analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(4):333–340.

Barron, J. L., Beauchemin, S. S., and Fleet, D. J. (1994a). On optical flow. In Plander, I., editor, *Artificial Intelligence and Information-Control Systems of Robots'94*, pages 3–14. World Scientific, Singapore. Proceedings of the Sixth International Conference on Artificial Inelligence and Information-Control Systems of Robots.

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994b). Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77.

Barron, J. L., Jepson, A. D., and Tsotsos, J. K. (1990). The feasibility of motion and structure from noisy time-varying image velocity information. *International Journal of Computer Vision*, 5(3):239–269.

Basri, R. and Rivlin, E. (1993a). Homing using combinations of model views. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1586–1591.

Basri, R. and Rivlin, E. (1993b). Localization using combinations of model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 226–230, Berlin. IEEE Computer Society Press.

Basri, R. and Rivlin, E. (1995). Localization and homing using combinations of model views. *Artificial Intelligence*, 78:327–354.

Beauchemin, S. S. and Barron, J. L. (1995). The computation of optical flow. *ACM Computing Surveys*, 27(3):432–467.

Beckers, R., Deneubourg, J. L., and Goss, S. (1992). Trails and U-turns in the selection of the shortest path by the ant *Lasius Niger*. *Journal of Theoretical Biology*, 159:397–415.

Beckers, R., Holland, O. E., and Deneubourg, J. L. (1994). From local actions to global tasks: Stigmergy and collective robotics. In Brooks, R. A. and Maes, P., editors, *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 181–189, Cambridge, Massachusetts. The MIT Press.

Beer, R. D., Chiel, H. J., Quinn, R. D., Espenschied, K. S., and Larsson, P. (1992). A distributed neural network architecture for hexapod robot locomotion. *Neural Computation*, 4(3):356–365.

Beer, R. D., Chiel, H. J., and Sterling, L. S. (1989). Heterogeneous neural networks for adaptive behavior in dynamic environments. In Touretzky, D. S., editor, *Neural Information Processing Systems*, pages 577–585. Morgan Kaufmann.

Beer, R. D., Chiel, H. J., and Sterling, L. S. (1990). A biological perspective on autonomous agent design. *Robotics and Autonomous Systems*, 6:169–186. Reprint available in (Maes, 1990).

Beer, R. D., Chiel, H. J., and Sterling, L. S. (1991). An artificial insect. *American Scientist*, 79:444–452.

Beer, R. D., Quinn, R. D., Chiel, H. J., and Ritzmann, R. E. (1997). Biologically inspired aproaches to robotics. *Communications of the ACM*, 40(3):31–38.

Bellingham, J. G., Consi, T. R., Beaton, R. M., and Hall, W. (1990). Keeping layered control simple. In *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology*, pages 3–8.

Bergen, J. R., Anandan, P., Hanna, K. J., and Hingorani, R. (1992). Hierarchical model-based motion estimation. In Sandini, G., editor, *Proceedings of the European Conference on Computer Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 237–252. Springer-Verlag, Berlin.

Bergholm, F. (1988). Motion from flow along contours: A note on robustness and ambiguous cases. *International Journal of Computer Vision*, 3:395–415.

Bernstein, R. A. (1975). Foraging strategies of ants in responce to variable food density *(Hymenoptera Formicidae). Ecology*, 56:213–219.

Billard, A. and Dautenbahn, K. (1997a). Grounding communication in situated, social robots. In *Proceedings of the TIMR Conference, Towards Intelligent Mobile Robots*, Manchester. Manchester University. Available as a technical report (Billard and Dautenbahn, 1997b).

Billard, A. and Dautenbahn, K. (1997b). Grounding communication in situated, social robots. Technical Report UMCS-97-9-1, Department of Computer Science, Manchester University. Available from http://www.cs.man.ac.uk/cstechrep/titles97.html.

Black, M. J. and Anandan, P. (1990). A model for the detection of motion over time. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 33–37. IEEE Computer Society Press.

Blake, A. and Yuille, editors (1992). *Active Vision.* The MIT Press, Cambridge, Massachusetts.

Bonabeau, E. G. and Theraulaz, G. (1994). *Intelligence Collective.* Hermés, Paris,

France.

Bonasso, P. (1992). Using parallel program specifications for reactive control of underwater vehicles. *Applied Intelligence*, 2(3):201–224.

Borst, A. (1986). Time course of the housflies' landing response. *Biological Cybernetics*, 54:379–383.

Borst, A. and Bahde, S. (1988). Visual information processing in the fly's landing system. *Journal of Comparative Physiology A*, 163:167–173.

Bouthemy, P. and François, E. (1993). Motion segmentation and qualitative dynamic scene analysis from an image sequence. *International Journal of Computer Vision*, 10(2):157–182.

Braitenberg, V. (1984). *Vehicles: experiments in synthetic psychology.* The MIT Press, Cambridge, Massachusetts.

Braitenberg, V. and Taddei-Ferretti, C. (1966). Landing reaction of *Musca domestica* induced by visual stimuli. *Naturwissenschaften*, 53:155–156.

Braunstein, M. L. (1976). *Depth perception through motion.* Academic Press, London.

Brooks, R. A. (1982). Symbolic error analysis and robot planning. *International Journal of Robotics Research*, 1(4):29–68.

Brooks, R. A. (1983). Solving the find-path problem by good representation of free space. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(3):190–197.

Brooks, R. A. (1985). A layered intelligent control system for a mobile robot. In *Proceedings of the Third International Symposium of Robotics Research*, pages 365–372.

Brooks, R. A. (1986a). Achieving artificial intelligence through building robots. AI Memo 899, Massachusetts Institute of Technology. Available from ftp://publications.ai.mit.edu/ai-publications.

Brooks, R. A. (1986b). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23.

Brooks, R. A. (1988). Engineering approach to building complete, intelligent beings. *SPIE Proceedings of Intelligent Robots and Computer Vision: Seventh in a Series*, 1002:618–625.

Brooks, R. A. (1989a). A robot that walks: Emergent behaviors from a carefully evolved network. *Neural Computation*, 1:253–262.

Brooks, R. A. (1989b). A robot that walks; Emergent behaviors from a carefully evolved network. AI Memo 1091, Massachusetts Institute of Technology.

Available from ftp://publications.ai.mit.edu/ai-publications.

Brooks, R. A. (1990). Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15. Reprint available in (Maes, 1990).

Brooks, R. A. (1991a). Challenges for complete creature architectures. In Meyer, J.-A. and Wilson, S. W., editors, *Proceedings of the International Conference on Simulation of Adaptive Behavior*, From Animals to Animats, pages 434–443, Cambridge, Massachusetts. The MIT Press.

Brooks, R. A. (1991b). Intelligence without reason. AI Memo 1293, Massachusetts Institute of Technology. Available from ftp://publications.ai.mit.edu/ai-publications.

Brooks, R. A. (1991c). Intelligence without representation. *Artificial Intelligence*, 47:139–159.

Brooks, R. A. (1991d). New approaches to robotics. *Science*, 253:1227–1232.

Brooks, R. A. (1997). Behavior-based robots for dangerous and difficult environments. In *Proceedings of the International Conference on Field and Service Robotics*, pages 2–4, Canberra. Australian Robot Association Incorporated, Panther Publishing.

Brooks, R. A. and Flynn, A. M. (1989). Fast, cheap and out of control: a robot invasion of the solar system. *Journal of The British Interplanetary Society*, 42(10):478–485.

Brünnert, U., Kelber, A., and Zeil, J. (1994). Ground-nesting bees determine the location of their nest relative to a landmark by other than angular size cues. *Journal of Comparative Physiology A*, 175:363–369.

Burgen, J. R., Burt, P. J., Hingorani, R., and Peleg, S. (1992). A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):886–896.

Burlina, P. and Chellappa, R. (1993). Time-to-X: analysis of motion through temporal parameters. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 461–468, Washington. IEEE Computer Society, IEEE Computer Society Press.

Cameron, S. and Probert, P., editors (1994). *Advanced Guided Vehicles: Aspects of the Oxford AGV Project*, volume 9 of *Robotics and Automated Systems*. World Scientific, Singapore.

Camhi, J. M. (1980). The escape system of the cockroach. *Scientific American*, 243(6):158–172.

Cartwright, B. A. and Collett, T. S. (1982). How honey bees use landmarks to

guide their return to a food source. *Nature*, 295:560–564.

Cartwright, B. A. and Collett, T. S. (1983). Landmark learning in bees: experiments and models. *Journal of Comparative Physiology*, 151:521–543.

Cartwright, B. A. and Collett, T. S. (1987). Landmark maps for honeybees. *Biological Cybernetics*, 57:85–93.

Chahl, J., Weber, K., Srinivasan, M. V., and Venkatesh, S. (1995). Centring behaviour for mobile robots using insect based cues. In *Proceedings of the Asian Conference on Computer Vision.*

Chahl, J. S. and Srinivasan, M. V. (1996). Visual computation of egomotion using an image interpolation technique. *Biological Cybernetics*, 74:405–411.

Chapman, R. F. (1982). *The Insects: Structure and Function.* Harvard University Press, third edition.

Chatila, R. and Laumond, J.-P. (1985). Position referencing and consistent world modeling for mobile robots. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pages 138–145.

Chatila, R. G. (1989). A robust layered control system for a mobile robot, by Rodney A. Brooks. In Khatib, O., Craig, J. J., and Lozano-Pérez, T., editors, *The Robotics Review 1*, pages 103–108. The MIT Press, Cambridge, Massachusetts.

Cheng, K., Collett, T. S., Pickhard, A., and Wehner, R. (1987). The use of visual landmarks by honeybees: bees weight landmarks according to their distance from the goal. *Journal of Comparative Physiology A*, 161:469–475.

Chiel, H. J., Beer, R. D., Quinn, R. D., and Espenschied, K. S. (1992). Robustness of a distributed neural network controller for a hexapod robot. *IEEE Transactions on Robotics and Automation*, 8(3):293–303.

Cipolla, R. and Blake, A. (1992). Surface orientation and time to contact from image divergence and deformation. In Sandini, G., editor, *Proceedings of the European Conference on Computer Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 187–202. Springer-Verlag, Berlin.

Clark, J. J. and Ferrier, N. J. (1992). Attentive visual servoing. In Blake, A. and Yuille, editors, *Active Vision*, pages 137–154. The MIT Press, Cambridge, Massachusetts.

Collett, T., Nalbach, H.-O., and Wagner, H. (1993). Visual stabilisation in arthropods. In Miles, F. A. and Wallman, J., editors, *Visual Motion and its Role in the Stabilisation of Gaze*, chapter 11, pages 239–263. Elsevier Science Publishers B.V.

Collett, T. S. (1978). Peering — a locust behaviour pattern for obtaining motion parallax information. *The Journal of Experimental Biology*, 76:237–241.

Collett, T. S. (1988). How ladybirds approach nearby stalks: a study of visual selectivity and attention. *Journal of Comparative Physiology A*, 163:355–363.

Collett, T. S. (1992). Landmark learning and guidance in insects. *Philosophical Transactions of the Royal Society of London B*, 337:295–303.

Collett, T. S. (1996). Insect navigation *en route* to the goal: Multiple strategies for the use of landmarks. *The Journal of Experimental Biology*, 199:227–235.

Collett, T. S. and Baron, J. (1994). Biological compasses and the coordinate frame of landmark memories in honeybee. *Nature*, 368:137–140.

Collett, T. S., Dillmann, E., Giger, A., and Wehner, R. (1992). Visual landmarks and route following in desert ants. *Journal of Comparative Physiology A*, 170:435–442.

Collett, T. S. and Harkness, L. I. K. (1982). Depth vision in animals. In Ingle, D. J., Goodale, M. A., and Mansfield, R. J. W., editors, *Analysis of Visual Behaviour*, chapter 4, pages 111–176. The MIT Press, Cambridge, Massachusetts.

Collett, T. S. and Land, M. F. (1975a). Visual control of flight behaviour in the hoverfly *Syritta pipiens* L. *Journal of Comparative Physiology*, 99:1–66.

Collett, T. S. and Land, M. F. (1975b). Visual spatial memory in a hoverfly. *Journal of Comparative Physiology*, 100:59–84.

Collett, T. S. and Land, M. F. (1978). How hoverflies compute interception courses. *Journal of Comparative Physiology*, 125:191–204.

Collins, T. R., Arkin, R. C., and Henshaw, A. M. (1993). Integration of reactive navigation with a flexible parallel hardware architecture. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 271–276, Los Alamitos, California. IEEE Computer Society Press.

Connell, J. H. (1987). Creature design with the subsumption architecture. In McDermott, J., editor, *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1124–1126. Morgan Kaufmann.

Connell, J. H. (1988). A behavior-based arm controller. AI Memo 1025, Massachusetts Institute of Technology. Available from ftp://publications.ai.mit.edu/ai-publications.

Connell, J. H. (1989). A behavior-based arm controller. *IEEE Transactions on Robotics and Automation*, 5(6):784–791. Available as a technical report (Connell, 1988).

Connell, J. H. (1990). *Minimalist Mobile Robotics*, volume 5 of *Perspectives in Artificial Intelligence*. Academic Press Inc., Boston.

Coombs, D., Herman, M., Hong, T., and Nashman, M. (1995a). Real-time obstacle avoidance using central flow divergence and peripheral flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 276–283. IEEE Computer Society Press. Available as a technical report (Coombs et al., 1995b).

Coombs, D., Herman, M., Hong, T., and Nashman, M. (1995b). Real-time obstacle avoidance using central flow divergence and peripheral flow. Internal Report 5605, National Institute of Standards and Technology, Intelligent Systems Division, Gaithersburg.

Coombs, D. and Roberts, K. (1992). "Bee-bot": using peripheral optical flow to avoid obstacles. *SPIE Proceedings of Intelligent Robots and Computer Vision XI*, 1825:714–721.

Coombs, D. and Roberts, K. (1993). Centering behavior using peripheral vision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 440–445, New York City, NY. IEEE Computer Society Press.

Cornilleau-Pérès, V. and Droulez, J. (1990). Stereo correspondence from optical flow. In Faugeras, O., editor, *Proceedings of the European Conference on Computer Vision*, volume 427 of *Lecture Notes in Computer Science*, pages 326–330. Springer-Verlag, Berlin.

Crowley, J. L. (1985). Navigation for an intelligent mobile robot. *IEEE Journal of Robotics and Automation*, RA-1(1):31–41.

Cruse, H. (1990). What mechanisms coordinate leg movement in walking arthropods? *Trends in Neural Science*, 13:15–31.

Dautenhahn, K. (1995). Getting to know each other — artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16:333–356.

David, C. T. (1982). Compensation for height in the control of groundspeed by *Drosophila* in a new,'Barber's Pole' wind tunnel. *Journal of Comparative Physiology A*, 147:485–493.

Deneubourg, J. L., Fresneau, D., Goss, S., Lechaud, J. P., and Pasteels, J. M. (1987). Self-organisation mechanisms in ant societies (II): Learning during foraging and division of labour. In Pasteels, J. M. and Deneubourg, J. L., editors, *From Individual to Collective Behaviour in Social Insects*, pages 177–196. Birkhäuser, Basel.

Deneubourg, J. L., Goss, S., Franks, N., and Pasteels, J. M. (1992). The blind leading the blind: modeling chemically mediated army ant raid patterns. *Journal of Insect Behavior*, 2(5):719–725.

Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chrétien, L. (1991). The dynamics of collective sorting: Robot-like ants and ant-like robots. In Meyer, J.-A. and Wilson, S. W., editors, *Proceedings of the International Conference on Simulation of Adaptive Behavior*, From Animals to Animats, pages 356–363, Cambridge, Massachusetts. The MIT Press.

Dickinson, J. A. (1994). Bees link local landmarks with celestial compass cues. *Naturwissenschaften*, 81:465–467.

Donner, M. D. (1987). *Real time control of walking*. Birkhäuser, Boston.

Dorigo, M. and Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(1):29–41.

Dreyfus, H. L. (1981). From micro-worlds to knowledge representation: AI at an impasse. In Haugeland, J., editor, *Mind Design: Philosophy, Psychology, Artificial Intelligence*, pages 161–204. The MIT Press, Cambridge, Massachusetts.

Drogoul, A. (1993). When ants play chess (Or can strategies emerge from tactical behaviours?). In *Proceedings of the Fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 13–27.

Duchon, A. P. and Warren, W. H. (1994). Robot navigation from a Gibsonian viewpoint. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 2272–2277, San Antonio.

Dumpert, K. (1981). *The Social Biology of Ants*. Pitman books, London.

Duncan, J. H. and Chou, T.-C. (1992). On the detection of motion and the computation of optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):346–352.

Dyer, F. C. (1991). Bees acquire route-based memories but not cognitive maps in a familiar landscape. *Animal Behaviour*, 41:239–246.

Elfes, A. (1986). A sonar-based mapping and navigation system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco.

Enkelmann, W. (1988). Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics, and Image Processing*, 43:150–177.

Enkelmann, W. (1990). Obstacle detection by evaluation of optical flow fields from image sequences. In Faugeras, O., editor, *Proceedings of the European Conference on Computer Vision*, volume 427 of *Lecture Notes in Computer Science*, pages 134–138. Springer-Verlag, Berlin.

Ennes, A. R. (1988a). The importance of torsion in the design of insect wings. *The Journal of Experimental Biology*, 140:137–160.

Ennes, A. R. (1988b). The inertial cause of wing rotation in *Diptera*. *The Journal of Experimental Biology*, 140:161–169.

Eriksson, E. S. (1980). Movement parallax and distance perception in the grasshopper (*Phaulacridium vittatum* (Sjöstedt)). *The Journal of Experimental Biology*, 86:337–340.

Esch, H. E. and Burns, J. E. (1995). Honeybees use optical flow to measure the distance of a food source. *Naturwissenschaften*, 82:38–40.

Esch, H. E. and Burns, J. E. (1996). Distance estimation by foraging honeybees. *The Journal of Experimental Biology*, 199:155–162.

Esch, H. E., Goller, F., and Burns, J. E. (1994). Honeybee waggle dances: the 'energy hypothesis' and thermoregulatory behaviour of foragers. *Journal of Comparative Physiology B*, 163:621–625.

Espenschied, K. S., Quinn, R. D., Beer, R. D., and Chiel, H. J. (1996). Biologically-based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot. *Robotics and Autonomous Systems*, 18:59–64.

Espenschied, K. S., Quinn, R. D., Chiel, H. J., and Beer, R. D. (1993). Leg coordination mechanisms in the stick insect applied to hexapod robot locomotion. *Adaptive Behavior*, 1(4):455–468.

Espenschied, K. S., Quinn, R. D., Chiel, H. J., and Beer, R. D. (1994). Biologically-inspired hexapod robot control. In *Proceedings of the Fifth International Symposium on Robotics and Manufacturing*, pages 14–18, Maui, Hawaii.

Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326.

Evans, K. S., Ünsal, C., and Bay, J. S. (1997). A reactive coordination scheme for a many-robot system. *IEEE Transactions on Robotics and Automation*,

27(4):598–610.

Everett, H. R. (1995). *Sensors for mobile robots: theory and application.* A. K. Peters, Wellesley, Massachusetts.

Ewert, J. O. (1980). *Neuroethology.* Springer-Verlag, Berlin.

Fermüller, C. (1993). Navigation preliminaries. In Aloimonos, Y., editor, *Active Perception*, pages 103–150. Laurence Erlbaum, New Jersey.

Fikes, R. E. and Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 5:189–208.

Firby, R. J. (1987). An investigation into reactive planning in complex domains. In *Proceedings of the National Conference on Artificial Intelligence*, pages 202–206, Seattle, Washington.

Fleet, D. J. and Jepson, A. D. (1990). Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104.

Fleet, D. J. and Langley, K. (1995). Recursive filters for optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):61–67.

Fletcher, R. P., Cannings, C., and Blackwell, P. G. (1995). Modelling foraging behaviour of ant colonies. In Morán, F., Moreno, A., Merelo, J. J., and Chacón, P., editors, *Proceedings of the European Conference on Artificial Life*, volume 929 of *Lecture Notes in Artificial Intelligence*, pages 772–783, Berlin. Springer-Verlag.

Floreano, D. and Mondada, F. (1996). Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(3):396–407.

Franceschini, N. (1992). Sequence discriminating neural network in the eye of the fly. In Eeckman, F. H., editor, *Analysis and Modeling of Neural Systems*, pages 189–197. Fluwer Academic Publishers, Boston.

Franceschini, N. (1996). Engineering applications of small brains. *FED Journal*, 7(Suppl. 2):38–52. R&D Association for Future Electron Devices.

Franceschini, N., Pichon, J.-M., and Blanes, C. (1992). From insect vision to robot vision. *Philosophical Transactions of the Royal Society of London B*, 337:283–294.

Franceschini, N., Riehle, A., and Nestour, A. L. (1989). Directionally selective motion detection by insect neurons. In Stavenga, D. G. and Hardie, R. C., editors, *Facets of Vision*, chapter 17, pages 360–390. Springer-Verlag, Berlin.

Franks, N. (1986). Teams in social insects: group retrieval of prey by army

ants (*Eciton burchelli*, Hymenoptera: Formicidae). *Behavioral Ecology and Sociobiology*, 18:425–429.

Franks, N. (1989). Army ants: a collective intelligence. *American Scientist*, 77:139–145.

Franz, M. O., Schölkopf, B., and Bülthoff, H. H. (1997a). Homing by parameterized scene matching. In Husbands, P. and Harvey, I., editors, *Proceedings of the European Conference on Artificial Life*, pages 236–245, Cambridge, Massachusetts. The MIT Press. Available as a technical report (Franz et al., 1997b).

Franz, M. O., Schölkopf, B., and Bülthoff, H. H. (1997b). Homing by parameterized scene matching. Technical Report 46, Max-Planck-Institut für biologische Kybernetik, Tübingen, Germany. Available as /pub/mpi-memos/TR-046.ps via anonymous ftp from ftp.mpik-tueb.mpg.de or from http://www.mpik-tueb.mpg.de/bu.html.

Franz, M. O., Schölkopf, B., Georg, P., Mallot, H. A., and Bülthoff, H. H. (1996). Learning view graphs for robot navigation. Technical Report 33, Max-Planck-Institut für biologische Kybernetik, Tübingen, Germany. Available as /pub/mpi-memos/TR-033.ps via anonymous ftp from ftp.mpik-tueb.mpg.de or from http://www.mpik-tueb.mpg.de/bu.html.

Franz, M. O., Schölkopf, B., Mallot, H. A., and Bülthoff, H. H. (1998). Learning view graphs for robot navigation. *Autonomous Robots*, 5:111–125. Available as a technical report (Franz et al., 1996).

Frier, H. J., Smith, C., and Collett, T. S. (1995). Magnetic fields can provide a directional framework for visual pattern learning in honeybees. In Elsner, N. and Menzel, R., editors, *Proceedings of the Göttingen Neurobiology Conference*, Learning and Memory.

Gat, E. (1992). Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots. In *Proceedings of the National Conference on Artificial Intelligence*, pages 809–815, San Jose, CA.

Gat, E., Behar, A., Desai, R., Ivlev, R., Loch, J., and Miller, D. P. (1993). Behavior control for planetory exploration: interim report. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 567–571, Los Alamitos, California. IEEE Computer Society Press.

Gaussier, P. and Zrehen, S. (1994). A constructivist approach for autonomous agents. In Thalmann, D., editor, *Artificial Life and Virtual Reality*, pages

97–113. John Wiley and Sons, London.

Georgeff, M. P. and Lansky, A. L. (1987). Reactive reasoning and planning. In *Proceedings of the National Conference on Artificial Intelligence*, pages 677–682, Seattle, Washington.

Giachetti, A., Campani, M., and Torre, V. (1994). The use of optical flow for the autonomous navigation. In Eklundh, J.-O., editor, *Proceedings of the European Conference on Computer Vision*, volume 800 of *Lecture Notes in Computer Science*, pages 146–151. Springer-Verlag, Berlin.

Gibson, J. J. (1950). *The perception of the visual world.* Greenwood Press, Westport, Connecticut.

Gibson, J. J. (1958). Visually controlled locomotion and visual orientation in animals. *The British Journal of Psychology*, 49:182–194.

Gibson, J. J. (1966). *The senses considered as perceptual systems.* Houghton Mifflin Company, Boston.

Gibson, J. J. (1979). *The ecological approach to visual perception.* Houghton Mifflin Company, Boston.

Gibson, J. J. and Gibson, E. J. (1957). Continuous perspective transformations and the perception of rigid motion. *The Journal of Experimental Biology*, 54:129–138.

Glazer, F. (1987a). Hierarchical gradient-based motion detection. In *Proceedings of the DARPA Image Understanding Workshop*, pages 733–748. DARPA, Morgan Kaufmann.

Glazer, F. (1987b). *Hierarchical motion detection.* PhD thesis, University of Massachusetts, Amherst. CMPSCI TR 87-02.

Goller, F. and Esch, H. E. (1990). Waggle dances of honey bees: Is distance measured through energy expenditure on outward flight? *Naturwissenschaften*, 77:594–595.

Goodman, L. J. (1960). The landing responses of insects I: The landing response of the fly, *Lucilia sericata*, and other Calliphorinae. *The Journal of Experimental Biology*, 37:854–878.

Goss, S., Aron, S., Deneubourg, J. L., and Pasteels, J. M. (1992). Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76:579–581.

Goss, S., Beckers, R., Deneubourg, J. L., Aron, S., and Pasteels, J. M. (1990). How trail laying and trail following can solve foraging problems for ant colonies. In Hughes, R. N., editor, *Behavioral Mechanisms of Food Selection*, volume G 20 of *NATO ASI Series*, pages 661–678. Springer-Verlag, Heidelberg.

Goto, G. and Stentz, A. (1987). The CMU system for mobile robot navigation. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 99–105.

Götz, K. G. (1968). Flight control in *Drosophila* by visual perception of motion. *Kybernetic*, 4:199–208.

Gould, J. L. (1986). The locale map of honey bees: Do insects have cognitive maps? *Science*, 232:861–863.

Gould, J. L. and Towne, W. F. (1987). Evolution of the dance language. *American Naturalist*, 130:317–338.

Goulet, M., Campan, R., and Lambin, M. (1981). The visual perception of relative distances in the wood-cricket *Nemobius sylvestris*. *Physiological Entomology*, 6:357–367.

Grasso, F., Consi, T., Mountain, D., and Atema, J. (1996). Locating odor sources in turbulence with a lobster inspired robot. In Maes, P., Matarić, M. J., Meyer, J. A., Pollack, J., and Wilson, S. W., editors, *Proceedings of the International Conference on Simulation of Adaptive Behavior*, volume 4 of *From Animals to Animats*. The MIT Press, Cambridge, Massachusetts.

Grzywacz, N. M. and Yuille, A. L. (1990). A model for the estimation of local velocity. In Faugeras, O., editor, *Proceedings of the European Conference on Computer Vision*, volume 427 of *Lecture Notes in Computer Science*, pages 331–335. Springer-Verlag, Berlin.

Guthrie, D. M. (1980). *Neuroethology: An Introduction*. Blackwell Scientific Publications, Oxford.

Harmon, S. Y. (1987). The ground surveillance robot (GSR): An autonomous vehicle designed to transit unknown territory. *IEEE Journal of Robotics and Automation*, 3:266–279.

Hashimoto, M., Oba, F., and Nakahara, H. (1991). Trajectory generation and tracking control methods for a multiple transfer robots system. In *Proceedings of the IEEE/RSJ International Workshop IROS*, pages 799–804, Osaka, Japan.

Hausen, K. and Egelhaaf, M. (1989). Neural mechanisms of visual course control in insects. In Stavenga, D. G. and Hardie, R. C., editors, *Facets of vision*, chapter 18, pages 391–424. Springer-Verlag, Berlin.

Hay, J. C. (1966). Optical motions and space perception: An extension of Gibson's analysis. *Psychological Review*, 73(6):550–565.

Heeger, D. J. (1988). Optical flow using spatiotemporal filters. *International*

Journal of Computer Vision, 1:279–302.

Heeger, D. J. and Jepson, A. D. (1992). Subspace methods for recovering rigid motion I: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117.

Heel, J. (1990). Direct estimation of structure and motion from multiple frames. AI Memo 1190, Massachusetts Institute of Technology. Available from ftp://publications.ai.mit.edu/ai-publications.

Heisenberg, M. and Wolf, R. (1993). The sensory-motor link in motion-dependent flight control of flies. In Miles, F. A. and Wallman, J., editors, *Visual Motion and its Role in the Stabilization of Gaze*, chapter 12, pages 265–283. Elsevier Science Publishers B.V.

Hengstenberg, R. (1993). Multisensory control in insect oculomotor systems. In Miles, F. A. and Wallman, J., editors, *Visual Motion and its Role in the Stabilization of Gaze*, chapter 13, pages 285–298. Elsevier Science Publishers B.V.

Hermann, H. R., editor (1982). *Social Insects*. Academic Press, New York.

Hexmoor, H. and Nute, D. (1992). Methods for deciding *what to do next* and learning. Technical Report TR-92-23, Department of Computer Science, State University of New York, Buffalo, New York. Available from ftp://ftp.cs.buffalo.edu/pub/tech-reports/92-23.ps.Z.

Hexmoor, H. H., Lammens, J. M., Caicedo, G., and Shapiro, S. C. (1993a). Behavior based AI, cognitive processes, and emergent behaviors in autonomous agents. Technical Report TR-93-15, Department of Computer Science, State University of New York, Buffalo, New York. Available from ftp://ftp.cs.buffalo.edu/pub/tech-reports/93-15.ps.Z.

Hexmoor, H. H., Lammens, J. M., and Shapiro, S. C. (1993b). An autonomous agent architecture for integrating "unconscious" and "conscious", reasoned behaviors. Technical Report TR-93-37, Department of Computer Science, State University of New York, Buffalo, New York. Available from ftp://ftp.cs.buffalo.edu/pub/tech-reports/93-37.ps.Z.

Hoffmann, G. (1978). *Experimentelle und theoretische Analyse eines adaptiven Orientierungsverhaltens: Die „optimale" Suche der Wüstenassel* Hemilepistus reaumuri, *Audouin und Savigny (Crustacea, Isopoda, Onicoidea) nach ihrer Höhle*. PhD thesis, Regensburg.

Hoffmann, G. (1983a). The random elements of the systematic search behavior of the desert isobod *Hemilepistus reaumuri*. *Behavioral Ecology and Sociobiology*,

13:81–92.

Hoffmann, G. (1983b). The search behavior of the desert isobod *Hemilepistus reaumuri* as compared with a systematic search. *Behavioral Ecology and Sociobiology*, 13:425–429.

Hoffmann, G. (1984). Homing by systematic search. In Varjú and Schnitzler, editors, *Localization and Orientation in Biology and Engineering*, pages 192–199. Springer-Verlag, Berlin.

Hölldobler, B. and Wilson, E. O. (1990). *The Ants*. The Belknap Press of Harvard University Press, Cambridge, Massachusetts.

Hong, J., Tan, X., Pinette, B., Weiss, R., and Riseman, E. M. (1991). Image-based homing. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 620–625, Sacremento, CA.

Hong, J., Tan, X., Pinette, B., Weiss, R., and Riseman, E. M. (1992). Image-based homing. *IEEE Control Systems*, 12(1):38–45.

Hong, J.-W., Tan, X., Pinette, B., Weiss, R., and Riseman, E. M. (1990). Image-based navigation using 360° views. In *Proceedings of the DARPA Image Understanding Workshop*, pages 782–791. DARPA, Morgan Kaufmann.

Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

Horn, B. K. P. and Weldon, E. J. (1987). Computationally-efficient methods for recovering translational motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2–11. IEEE Computer Society Press.

Horridge, G. A. (1986). A theory of insect vision: velocity parallax. *Proceedings of the Royal Society of London B*, 229:13–27.

Horridge, G. A. (1987). The evolution of visual processing and the construction of seeing systems. *Proceedings of the Royal Society of London B*, 230:279–292.

Irani, M., Rousso, B., and Peleg, S. (1994). Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16.

Jain, R. C. (1983). Direct computation of the focus of expansion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(1):58–63.

Jain, R. C. (1984). Segmentation of frame sequences obtained by a moving observer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(5):624–629.

Jarvis, R. (1997). Intelligent robotics — where are we at and where are we going? In *Proceedings of the International Conference on Field and Service Robotics*,

pages 21–25, Canberra. Australian Robot Association Incorporated, Panther Publishing.

Jenkin, M. R. M. (1990). On the use of trajectory information to assist stereopsis in a dynamic environment. In Faugeras, O., editor, *Proceedings of the European Conference on Computer Vision*, volume 427 of *Lecture Notes in Computer Science*, pages 341–345. Springer-Verlag, Berlin.

Jenkin, M. R. M., Jepson, A. D., and Tsotsos, J. K. (1991). Techniques for disparity measurement. *CVGIP: Image Understanding*, 53(1):14–30.

Jepson, A. D. and Black, M. (1993). Mixture models for optical flow computation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 760–761. IEEE Computer Society Press.

Jones, J. and Flynn, A. (1993). *Mobile Robots: Inspiration to Implementation.* A. K. Peters, Wellesley, Massachusetts.

Judd, S. P. D. and Collett, T. S. (1998). Multiple stored views and landmark guidance in ants. *Nature*, 392:710–714.

Junger, W. (1991). Waterstriders (*Gerris paludum* f.) compensate for drift with a discontinuously working visual position servo. *Journal of Comparative Physiology A*, 169:633–639.

Junger, W. and Dahmen, H. J. (1991). Response to self-motion in waterstriders: visual discrimination between rotation and translation. *Journal of Comparative Physiology A*, 169:641–646.

Junger, W. and Varjú, D. (1990). Drift compensation and its sensory basis in waterstriders (*Gerris paludum* f.). *Journal of Comparative Physiology A*, 167:441–446.

Kadonoff, M., Benayad-Cherif, F., Flanklin, A., Maddox, J., Muller, L., Sert, B., and Moravec, H. (1987). Arbitration of multiple control strategies for mobile robots. *SPIE Proceedings of Mobile Robots*, 727:90–98.

Kaelbling, L. P. (1987). An architecture for intelligent reactive systems. In Georgeff, M. P. and Lansky, A. L., editors, *Reasoning About Actions and Plans*, pages 395–410. Morgan Kaufmann, Los Altos, CA.

Kalivas, D. S. and Sawchuk, A. A. (1991). A region matching motion estimation algorithm. *Computer Vision, Graphics, and Image Processing*, 54(2):275–288.

Kirchner, W. H. and Braun, U. (1994). Dancing honey bees indicate the location of food sources using path integration rather than cognitive maps. *Animal Behaviour*, 48:1437–1441.

Kirchner, W. H. and Srinivasan, M. V. (1989). Freely flying honeybees use image

motion to estimate object distance. *Naturwissenschaften*, 76:281–282.

Kirchner, W. H. and Towne, W. F. (1994). The sensory basis of the honeybee's dance language. *Scientific American*, 270:52–59.

Kleiner, K. (1994). Look to the insect... *New Scientist*, 144(1951):27–29.

Koenderink, J. J. (1986). Optic flow. *Vision Research*, 26(1):161–179.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.

Kubo, Y., Shimoyama, I., Kaneda, T., and Miura, H. (1994). Study on wings of flying microbots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 834–839. IEEE Computer Society Press.

Kubo, Y., Shimoyama, I., and Miura, H. (1993). Study of insect-based flying microbots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 386–391. IEEE Computer Society Press.

Kukuda, T., Kawamoto, A., Arai, F., and Matsuura, H. (1994). Mechanism and swimming experiment of micro mobile robot in water. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 814–819. IEEE Computer Society Press.

Lambrinos, D., Maris, M., Kobayashi, H., Labhart, T., Pfeifer, R., and Wehner, R. (1998). An autonomous agent navigating with a polarized light compass. *Adaptive Behavior*, 6(1):131–161.

Lammens, J. M., Hexmoor, H. H., and Shapiro, S. C. (1993). Of elephants and men. Technical Report 93-13, Department of Computer Science, State University of New York, Buffalo, New York. Available from ftp://ftp.cs.buffalo.edu/pub/tech-reports/93-13.ps.Z.

Lammens, J. M., Hexmoor, H. H., and Shapiro, S. C. (1995). Of elephants and men. In Steels, L., editor, *The Biology and Technology of Intelligent Autonomous Agents*, volume 144 of *NATO ASI Series F: Computer and Systems Sciences*, pages 312–344. Springer-Verlag, Berlin. Available as a technical report (Lammens et al., 1993).

Land, M. F. and Collett, T. S. (1974). Chasing behaviour of houseflies (*Fannia canniculatis*): A description and analysis. *Journal of Comparative Physiology*, 89:331–357.

Langer, D., Rosenblatt, J. K., and Hebert, M. (1994a). A behavior-based system system for off-road navigation. *IEEE Transactions on Robotics and Automation*, 10(6):776–783.

Langer, D., Rosenblatt, J. K., and Hebert, M. (1994b). An integrated system for

autonomous off-road navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 414–419. IEEE Computer Society Press.

Langley, K., Atherton, T. J., Wilson, R. G., and Larcombe, M. H. E. (1990). Vertical and horizontal disparities from phase. In Faugeras, O., editor, *Proceedings of the European Conference on Computer Vision*, volume 427 of *Lecture Notes in Computer Science*, pages 315–325. Springer-Verlag, Berlin.

Langley, K., Atherton, T. J., Wilson, R. G., and Larcombe, M. H. E. (1991). Vertical and horizontal disparities from phase. *Image and Vision Computing*, 9(5):296–302.

Laughlin, S. (1987). New views from compound eyes. *Nature*, 328:292–293.

Lee, D., Papageorgiou, A., and Wasilkowski, G. W. (1988). Computational aspects of determining optical flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 612–618. IEEE Computer Society Press.

Lee, D. and Reddish, P. (1981). Plummeting gannets: a paradigm of ecological optics. *Nature*, 293.

Lee, D. N. (1976). A theory of visual control of breaking based on information about time-to-collision. *Perception*, 5:437–459.

Lehrer, M. (1991). Bees which turn back and look. *Naturwissenschaften*, 78:274–276.

Lehrer, M. (1993). Why do bees turn back and look? *Journal of Comparative Physiology A*, 172:549–563.

Lehrer, M. (1996). Small-scale navigation in the honeybee: active acquisition of visual information about the goal. *The Journal of Experimental Biology*, 199:253–261.

Lehrer, M. and Srinivasan, M. V. (1992). Freely flying bees discriminate between stationary and moving objects: performance and possible mechanisms. *Journal of Comparative Physiology A*, 171:457–467.

Lehrer, M. and Srinivasan, M. V. (1993). Object-ground discrimination in bees: why do they land on edges? *Journal of Comparative Physiology A*, 173:23–32.

Lehrer, M., Srinivasan, M. V., and Zhang, S. W. (1990). Visual edge detection in the honeybee and its chromatic properties. *Proceedings of the Royal Society of London B*, 238:321–330.

Lehrer, M., Srinivasan, M. V., Zhang, S. W., and Horridge, G. A. (1988). Motion cues provide the bee's visual world with a third dimension. *Nature*, 332:356–

357.

Lewis, M. A., Fagg, A. H., and Bekey, G. A. (1993). The USC autonomous flying vehicle: an experiment in real-time behavior-based control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 422–429, Los Alamitos, California. IEEE Computer Society Press.

Lewis, T., editor (1984). *Insect Communication.* Academic Press, London.

Lindauer, M. (1960). Time-compensated sun orientation in bees. In *Cold Spring Harbor Symposium on Quantitative Biology*, volume 25, pages 371–377.

Little, J. J. and Gillett, W. E. (1990). Direct evidence for occlusion in stereo and motion. In Faugeras, O., editor, *Proceedings of the European Conference on Computer Vision*, volume 427 of *Lecture Notes in Computer Science*, pages 336–340. Springer-Verlag, Berlin.

Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135.

Longuet-Higgins, H. C. (1984). The visual ambiguity of a moving plane. *Proceedings of the Royal Society of London B*, 223:165–175.

Longuet-Higgins, H. C. and Prazdny, K. (1980). The interpretation of a moving retinal image. *Proceedings of the Royal Society of London B*, 208:385–397.

Lozano-Pérez, T. (1987). A simple motion-planning algorithm for general robot manipulation. *IEEE Journal of Robotics and Automation*, RA-3(3):224–238.

Lozano-Pérez, T., Jones, J. L., Mazer, E., and O'Donnell, P. A. (1989). Task-level planning of pick-and-place robot motion. *Computer*, 22:21–29.

Lozano-Pérez, T. and Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570.

Lund, H. H., Webb, B., and Hallam, J. (1996). A robot attracted to the cricket species *Gryllus bimaculatus.* In *Proceedings of the European Conference on Artificial Life*, pages 246–255.

Maes, P., editor (1990). *Designing Autonomous Agents: theory and practice from biology to engineering and back.* The MIT Press, Cambridge, Massachusetts.

Maris, M. and te Boekhorst, R. (1996). Exploiting physical constraints: Heap formation through behavioral error in a group of robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'96*, Osaka, Japan.

Martinoli, A. and Mondada, F. (1995). Collective and cooperative group behaviours: Biologically inspired experiments in robotics. In *Proceedings of*

the Fourth International Synposium on Experimental Robotics, pages 3–10, Stanford, USA. Springer-Verlag.

Martinoli, A., Yamamoto, M., and Mondada, F. (1997). On the modelling of bio-inspired collective experiments with real robots. In *WWW Proceedings of the 4th European Conference on Artificial Life ECAL'97*, Brighton, UK.

Matarić, M. J. (1990). A distributed model for mobile robot environment-learning and navigation. AI Memo 1228, Massachusetts Institute of Technology.

Matarić, M. J. (1991). Navigating with a rat brain: A nerobiologically-inspired model for robot spatial representation. In Meyer, J.-A. and Wilson, S. W., editors, *Proceedings of the International Conference on Simulation of Adaptive Behavior*, From Animals to Animats, pages 169–175, Cambridge, Massachusetts. The MIT Press.

Matarić, M. J. (1992). Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3):304–312.

Meyer, F. and Bouthemy, P. (1992). Estimation of time-to-collision maps from first order motion model and normal flows. In *Proceedings of the International Conference on Pattern Recognition*, pages 78–82.

Micheli, E. D., Torre, V., and Uras, S. (1993). The accuracy of the computation of optical flow and of the recovery of motion parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):434–447.

Mitchell, T. M. (1990). Becoming increasingly reactive. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1051–1058, Boston, MA.

Mitiche, A., Grisell, R., and Aggarwal, J. K. (1988). On smoothness of a vector field—application to optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):943–949.

Mondada, F., Franzi, E., and Ienne, P. (1993). Mobile robot miniaturization: A tool for investigation in control algorithms. In *Proceedings of the Third International Synposium on Experimental Robotics*, pages 501–513, Kyoto, Japan.

Montgomery, J. F., Fagg, A. H., and Bekey, G. A. (1995). The USC AFV-I: A behaviour-based entry in the 1994 international aerial robotics competition. *IEEE Expert*, pages 16–22.

Moravec, H. P. (1977). Towards automatic visual obstacle avoidance. In *Proceedings of the International Joint Conference on Artificial Intelligence*, page 584.

Moravec, H. P. (1979). Visual mapping by a robot rover. In *Proceedings of the*

International Joint Conference on Artificial Intelligence, pages 598–600.

Moravec, H. P. (1980). *Obstacle avoidance and navigation in the real world by a seeing robot rover.* PhD thesis, Stanford University.

Moravec, H. P. (1981). Rover visual obstacle avoidance. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 785–790.

Moravec, H. P. (1983). The stanford cart and the CMU rover. *Proceedings of the IEEE*, 71(7):872–884.

Müller, M. and Wehner, R. (1988). Path integration in desert ants, *Cataglyphis fortis. Proceedings of the National Academy of Sciences of the United States of America*, 85:5287–5290.

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

Mura, F., Martin, N., and Franceschini, N. (1996). Biologically inspired eye movements for the visually guided navigation of mobile robots. In Verleysen, M., editor, *Proceedings of the European Symposium on Artificial Neural Networks*, pages 135–147.

Murray, D. W. and Buxton, B. F. (1987). Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(2):220–228.

Nagel, H.-H. (1990). Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow. In Faugeras, O., editor, *Proceedings of the European Conference on Computer Vision*, volume 427 of *Lecture Notes in Computer Science*, pages 139–148. Springer-Verlag, Berlin.

Nagle, M. G. and Srinivasan, M. V. (1996). Structure from motion: determining the range and orientation of surfaces by image interpolation. *Journal of the Optical Society of America A: Optics and Image Science*, 13(1):25–34.

Nakayama, K. and Loomis, J. M. (1974). Optical velocity patterns, velocity-sensitive neurons, and space perception: a hypothesis. *Perception*, 3:63–80.

Negahdaripour, S. and Horn, B. K. P. (1987). Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1):168–176.

Negahdaripour, S. and Lee, S. (1992). Motion recovery from image sequences using only first order optical flow information. *International Journal of Computer Vision*, 9(3):163–184.

Nehmzow, U. (1995a). Animal and robot navigation. *Robotics and Autonomous Systems*, 15:71–81. Available as (Nehmzow, 1995b).

Nehmzow, U. (1995b). Animal and robot navigation. In Steels, L., editor, *The Biology and Technology of Intelligent Autonomous Agents*, volume 144 of *NATO ASI Series F: Computer and Systems Sciences*, pages 258–274. Springer-Verlag, Berlin.

Nelson, R. C. (1989a). Visual homing using an associative memory. In *Proceedings of the DARPA Image Understanding Workshop*, pages 245–262. DARPA, Morgan Kaufmann.

Nelson, R. C. (1989b). Visual homing using an associative memory. In *Proceedings of the AAAI Spring Symposium Series*, Palo Alto, CA.

Nelson, R. C. (1991). Visual homing using an associative memory. *Biological Cybernetics*, 65:281–291.

Nelson, R. C. and Aloimonos, J. Y. (1989). Obstacle avoidance using flow field divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1102–1106.

Nilsson, N. J. (1984). Shakey the robot. Technical Note 323, AI center, SRI International, Menlo Park, California.

Noriels, F. R. and Chatila, R. G. (1989). Control of mobile robot actions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 701–707.

Norman, D. and Shallice, T. (1986). Attention to action: willed and automatic control of behavior. In Davidson, R., Schwartz, G., and Shapiro, D., editors, *Consciousness and Self-regulation: Advances in Research and Theory*, volume 4, pages 1–18. Plenum Press, New York.

Ogata, M. and Sato, T. (1992). Motion detection model with two stages: spatiotemporal filtering and feature matching. *Journal of the Optical Society of America A: Optics and Image Science*, 9(13):377–387.

Okutomi, M. and Kanade, T. (1990). A locally adaptive window for signal matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 190–199. IEEE Computer Society Press.

Olberg, R. M. (1981). Object- and self-movement detectors in the ventral nerve cord of the dragonfly. *Journal of Comparative Physiology*, 141:327–334.

Overington, I. (1987). Gradient-based flow segmentation and location of the focus of expansion. In *Proceedings of the Alvey Vision Conference*, pages 169–177.

Papi, F., editor (1992). *Animal Homing*. Animal Behaviour Series. Chapman and

Hall, London.

Pasteels, J. M. and Deneubourg, J. L., editors (1987). *From Individual to Collective Behaviour in Social Insects.* Birkhäuser, Basel.

Pasteels, J. M., Deneubourg, J. L., and Goss, S. (1987). Self-organisation mechanisms in ant societies (i): Trail recruitment to newly discovered food sources. In Pasteels, J. M. and Deneubourg, J. L., editors, *From Individual to Collective Behaviour in Social Insects*, pages 155–176. Birkhäuser, Basel.

Payton, D. (1986). An architecture for reflexive autonomous vehicle control. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1838–1845.

Payton, D. (1990). Interalized plans: A representation for action resources. *Robotics and Autonomous Systems*, 6:89–103. Reprint available in (Maes, 1990).

Payton, D. W. and Bihari, T. E. (1991). Intelligent real-time control of robotic vehicles. *Communications of the ACM*, 34(8):48–63.

Payton, D. W., Keirsey, D., Kimble, D. M., Krozel, J., and Rosenblatt, K. (1992). Do whatever works: a robust approach to fault-tolerant autonomous control. *Applied Intelligence*, 2(3):225–250.

Payton, D. W., Rosenblatt, J. K., and Keirsey, D. M. (1990). Plan guided reaction. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1370–1382.

Pearson, K. G. (1976). The control of walking. *Scientific American*, 235:72–86.

Pearson, K. G., Fourtner, C. R., and Wong, R. K. (1973). Nervous control of walking in the cockroach. In Stein, R. B., Pearson, K. G., Smith, R. S., and Redford, J. B., editors, *Control of posture and locomotion.* Plenum Press, New York.

Pearson, K. G. and Franklin, R. (1984). Characteristics of leg movements and patterns of coordination in locusts walking on rough terrain. *International Journal of Robotics Research*, 3:101–112.

Peckham, G. W. and Peckham, E. G. (1905). *Wasps, social and solitary.* Constable, Westminster.

Pichon, J.-M., Blanes, C., and Franceschini, N. (1989). Visual guidance of a mobile robot equipped with a network of self-motion sensors. *SPIE Proceedings of Mobile Robots IV*, 1195:44–53.

Pinette, B. (1991). Qualitative homing. In *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, pages 318–323. 13-15 August,

Arlington, Virginia.

Pinette, B. (1994). *Image-based navigation through large-scale environments.* PhD thesis, University of Massachusetts, Amherst. CMPSCI TR 94-87.

Poggio, M. and Poggio, T. (1994). Cooperative physics of fly swarms: an emergent behavior. AI Memo 1512, Massachusetts Institute of Technology. Available from ftp://publications.ai.mit.edu/ai-publications.

Prazdny, K. (1979). Motion and structure from optical flow. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 702–704.

Prazdny, K. (1980). Egomotion and relative depth map from optical flow. *Biological Cybernetics*, 36:87–102.

Prescott, T. J. (1996). Spatial representation for navigation in animats. *Adaptive Behavior*, 4(2):85–123.

Prescott, T. J. and Ibbotson, C. (1997a). The early evolution of spatial behaviour: robot models of trace fossils. In Sharkey, N. and Nehmzow, U., editors, *Proceedings of the AISB workshop on Spatial Reasoning in Mobile Robots and Animals.* Available as a technical report (Prescott and Ibbotson, 1997b).

Prescott, T. J. and Ibbotson, C. (1997b). The early evolution of spatial behaviour: robot models of trace fossils. Technical Report UMCS-97-4-1, Manchester University. Available from http://www.cs.man.ac.uk/cstechrep/titles97.html.

Quinlan, S. and Khatib, O. (1993). Elastic bands: connecting path planning and control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 802–807. IEEE Computer Society Press.

Raup, D. M. and Seilacher, A. (1969). Fossil foraging behaviour: computer simulation. *Science*, 166:994–995.

Raviv, D. and Herman, M. (1993). Visual servoing from 2-D image cues. In Aloimonos, Y., editor, *Active Perception*, pages 191–226. Laurence Erlbaum, New Jersey.

Regan, D. and Beverley, K. I. (1982). How do we avoid confounding the direction we are looking and the direction we are moving? *Science*, 215:194–196.

Reichardt, W., Egelhaaf, M., and Schlögl, R. W. (1988). Movement detectors provide sufficient information for local computation of 2-D velocity fields. *Naturwissenschaften*, 75:313–315.

Reichhardt, W. and Poggio, T. (1979). Figure-ground discrimination by relative movement in the visual system of the fly. Part 1: experimental results. *Biological Cybernetics*, 35:81–100.

Robinson, M. and Jenkin, M. (1995). Reactive control of a mobile robot. In

Archibald, C. and Kwok, P., editors, *Research in Computer and Robot Vision*, pages 55–70. World Scientific, Singapore.

Röfer, T. (1995a). Controlling a robot with image based homing. Technical Report 3/95, Universitët Bremen. Available from http://www.informatik.uni-bremen.de/~roefer/public.htm.

Röfer, T. (1995b). Image-based homing using a self-organizing feature map. In Fogelman-Soulié, F. and Gallinari, P., editors, *Proceedings of the International Conference on Artificial Neural Networks*, volume 1, pages 475–480, Nanterre, France. EC2. Available from http://www.informatik.uni-bremen.de/~roefer/public.htm.

Röfer, T. (1997a). Controlling a wheelchair with image-based homing. In Sharkey, N. and Nehmzow, U., editors, *Proceedings of the AISB workshop on Spatial Reasoning in Mobile Robots and Animals*. Available as a technical report (Röfer, 1997b).

Röfer, T. (1997b). Controlling a wheelchair with image-based homing. Technical Report UMCS-97-4-1, Manchester University. Available from http://www.cs.man.ac.uk/cstechrep/titles97.html.

Rognone, A., Campani, M., and Verri, A. (1992). Identifying multiple motions from optical flow. In Sandini, G., editor, *Proceedings of the European Conference on Computer Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 258–266. Springer-Verlag, Berlin.

Ronacher, B. and Wehner, R. (1995). Desert ants *Cataglyphis fortis* use self-induced optic flow to measure distances travelled. *Journal of Comparative Physiology A*, 177:21–27.

Rossel, S. (1983). Binocular stereopsis in an insect. *Nature*, 302:821–822.

Rossel, S. (1986). Binocular spatial localization in the praying mantis. *The Journal of Experimental Biology*, 120:265–281.

Rossel, S. (1993). Navigation by bees using polarized skylight. *Comparative Biochemistry and Physiology A*, 104:695–708.

Rossel, S. and Wehner, R. (1986). Polarization vision in bees. *Nature*, 323:128–131.

Sacerdotti, E. D. (1977). *A Structure for Plans and Behavior*. Elsevier Science Publishers, North-Holland.

Sandini, G., Gandolfo, F., Grosso, E., and Tistarelli, M. (1993). Vision during action. In Aloimonos, Y., editor, *Active Perception*, pages 151–190. Laurence Erlbaum, New Jersey.

Santos-Victor, J., Bernardino, A., and Silva, C. (1997a). On the design of visual behaviors for autonomous systems. In *Proceedings of the IEEE International Symposium on Industrial Electronics*, Guimaraes, Portugal. Available as a technical report (Santos-Victor et al., 1997b).

Santos-Victor, J., Bernardino, A., and Silva, C. (1997b). On the design of visual behaviors for autonomous systems. VisLab-TR 06/97, Instituto Superior Técnico, Lisboa, Portugal.

Santos-Victor, J. and Sandini, G. (1995a). Visual-based obstacle detection: a purposive approach using the normal flow. In *Proceedings of the International Conference on Intelligent Autonomous Systems*, Karlsruhe, Germany. Available as a technical report (Santos-Victor and Sandini, 1995b).

Santos-Victor, J. and Sandini, G. (1995b). Visual-based obstacle detection: a purposive approach using the normal flow. VisLab-TR 02/95, Instituto Superior Técnico, Lisboa, Portugal.

Santos-Victor, J. and Sandini, G. (1997a). Embedded visual behaviors for navigation. *Robotics and Autonomous Systems*, 19(3–4). Available as a technical report (Santos-Victor and Sandini, 1997b).

Santos-Victor, J. and Sandini, G. (1997b). Embedded visual behaviors for navigation. VisLab-TR 04/97, Instituto Superior Técnico, Lisboa, Portugal.

Santos-Victor, J. and Sandini, G. (1997c). Visual behaviors for docking. *Computer Vision and Image Understanding*, 67(3). Available as a technical report (Santos-Victor and Sandini, 1997d).

Santos-Victor, J. and Sandini, G. (1997d). Visual behaviors for docking. VisLab-TR 10/97, Instituto Superior Técnico, Lisboa, Portugal.

Santos-Victor, J., Sandini, G., Curotto, F., and Garibaldi, S. (1993). Divergent stereo for robot navigation: Learning from bees. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 434–439, New York City, NY. IEEE Computer Society, IEEE Computer Society Press.

Santos-Victor, J., Sandini, G., Curotto, F., and Garibaldi, S. (1995). Divergent stereo in autonomous navigation: From bees to robots. *International Journal of Computer Vision*, 14(2):159–177.

Schnepf, U. (1991). Robot ethology: A proposal for research into intelligent autonomous systems. In Meyer, J.-A. and Wilson, S. W., editors, *Proceedings of the International Conference on Simulation of Adaptive Behavior*, From Animals to Animats, pages 465–474, Cambridge, Massachusetts. The MIT

Press.

Schnörr, C. (1992). Computation of discontinuous optical flow by domain decomposition and shape optimization. *International Journal of Computer Vision*, 8(2):153–165.

Schölkopf, B. and Mallot, H. A. (1994). View-based cognitive mapping and path planning. Technical Report 7, Max-Planck-Institut für biologische Kybernetik, Tübingen, Germany.

Schölkopf, B. and Mallot, H. A. (1995). View-based cognitive mapping and path planning. *Adaptive Behavior*, 3(3):311–348. Available as a technical report (Schölkopf and Mallot, 1994).

Schoonderwoerd, R., Holland, O., Bruten, J., and Rothkrantz, L. (1997). Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, 5(2).

Schoppers, M. J. (1987). Universal plans for reactive robots in unpredictable environments. In McDermott, J., editor, *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1039–1046. Morgan Kaufmann.

Schunck, B. G. (1986). The image flow constraint equation. *Computer Vision, Graphics, and Image Processing*, 35:20–46.

Schunck, B. G. (1989). Image flow segmentation and estimation by constraint line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1010–1027.

Scott, G. L. (1987). 'Four-line' method of locally estimating optic flow. *Image and Vision Computing*, 5(2):67–72.

Sedgewick, R. (1988). *Algorithms.* Addison-Wesley, Reading, Massachusetts, second edition.

Seilacher, A. (1967). Fossil behaviour. *Scientific American*, 217:72–80.

Seilacher, A. (1986). Evolution of behaviour as expressed in marine trace fossils. In Nitecki, M. H. and Kitchell, J. A., editors, *Evolution of Animal Behaviour.* Oxford University Press.

Shiffrin, R. and Schneider, W. (1977). Controlled and automatic human information processing. *Psychological Review*, 84:127–190.

Shimoyama, I., Miura, H., Suzuki, K., and Ezura, Y. (1992). 3D structure of an insect-based microbot with external skeleton. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 693–698. IEEE Computer Society Press.

Simmons, R. and Krotkov, E. (1991). An integrated walking system for the ambler

planetary rover. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 2086–2091.

Sobel, E. C. (1990). The locust's use of motion parallax to measure distance. *Journal of Comparative Physiology A*, 167:579–588.

Sobey, P. and Srinivasan, M. V. (1991). Measurement of optical flow by a generalized gradient scheme. *Journal of the Optical Society of America A: Optics and Image Science*, 8(9):1488–1498.

Srinivasan, M. V. (1990). Generalized gradient schemes for the measurement of two-dimensional image motion. *Biological Cybernetics*, 63:421–431.

Srinivasan, M. V. (1992a). Distance perception in insects. *Current Directions in Psychological Science*, 1(1):22–26.

Srinivasan, M. V. (1992b). How bees exploit optic flow: behavioural experiments and neural models. *Philosophical Transactions of the Royal Society of London B*, 337:253–259.

Srinivasan, M. V. (1993). How insects infer range from visual motion. In Miles, F. A. and Wallman, J., editors, *Visual Motion and its Role in the Stabilization of Gaze*, pages 139–156. Elsevier Science Publishers B.V., New York.

Srinivasan, M. V. (1994). An image-interpolation technique for the computation of optical flow and egomotion. *Biological Cybernetics*, 71:401–415.

Srinivasan, M. V. (1998). Ants match as they march. *Nature*, 392:660–661.

Srinivasan, M. V., Chahl, J. S., Nagle, M. G., Weber, K., Venkatesh, S., and Zhang, S. W. (1995). Low-level vision in insects, and applications to robot navigation. In Palaniswami, M., Attikiouzel, Y., Marks, R. J., Fogel, D., and Fukuda, T., editors, *Computational Intelligence: A Dynamic System Perspective*, pages 312–326. IEEE Press, New York.

Srinivasan, M. V., Chahl, J. S., Nagle, M. G., Weber, K., Venkatesh, S., and Zhang, S. W. (1997). Robot navigation inspired by principles of insect vision. In *Proceedings of the International Conference on Field and Service Robotics*, pages 18–20, Canberra. Australian Robot Association Incorporated, Panther Publishing.

Srinivasan, M. V., Chahl, J. S., Weber, K., Venkatesh, S., Nagle, M. G., and Zhang, S. W. (1998). Robot navigation inspired by principles of insect vision. *Robotics and Autonomous Systems*. Special issue on Field and Service Robotics. (To appear).

Srinivasan, M. V., Lehrer, M., and Horridge, G. A. (1990). Visual figure-ground discrimination in the honeybee: the role of motion parallax at boundaries.

Proceedings of the Royal Society of London B, 238:331–350.

Srinivasan, M. V., Lehrer, M., Kirchner, W. H., and Zhang, S. W. (1991). Range perception through apparent image speed in freely flying honeybees. *Visual Neuroscience*, 6:519–535.

Srinivasan, M. V., Lehrer, M., Zhang, S. W., and Horridge, G. A. (1989). How honeybees measure their distance from objects of unknown sizes. *Journal of Comparative Physiology A*, 163:605–613.

Srinivasan, M. V. and Venkatesh, S., editors (1997). *From Living Eyes to Seeing Machines*. Oxford University Press.

Srinivasan, M. V., Zhang, S. W., Lehrer, M., and Collett, T. S. (1996). Honeybee navigation *en route* to the goal: Visual flight control and odometry. *The Journal of Experimental Biology*, 199:237–244.

Stavenga, D. G. and Hardie, R. C., editors (1989). *Facets of vision*. Springer-Verlag, Berlin.

Stein, M. R. and Paul, R. P. (1994). Operator interaction, for time-delayed teleoperation, with a behavior-based controller. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 231–236. IEEE Computer Society Press.

Stilwell, D. J. and Bay, J. S. (1993). Toward the development of a material transport system using swarms of ant-like robots. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 766–771.

Subbarao, M. (1988). Interpretation of image flow: rigid curved surfaces in motion. *International Journal of Computer Vision*, 2:77–96.

Subbarao, M. (1989). Interpretation of image flow: A spatio-temporal approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(3):266–278.

Subbarao, M. (1990). Bounds on time-to-collision and rotational component from first-order derivatives of image motion. *CVGIP: Image Understanding*, 50:329–341.

Sundareswaran, V. (1992). A fast method to estimate sensor translation. In Sandini, G., editor, *Proceedings of the European Conference on Computer Vision*, volume 588 of *Lecture Notes in Computer Science*, pages 253–257. Springer-Verlag, Berlin.

Sutton, M. A., Wolters, W. J., Peters, W. H., Ranson, W. F., and McNeill, S. R. (1983). Determination of displacements using an improved digital correlation method. *Image and Vision Computing*, 1(3):133–139.

Tate, A. (1995). A review of knowledge-based planning techniques. *The Knowledge Engineering Review*, 1(3):4–17.

Theraulaz, G., Goss, S., Gervet, J., and Deneubourg, J. L. (1991). Task differentiation in Polistes wasp colonies: a model for self-organizing groups of robots. In Meyer, J.-A. and Wilson, S. W., editors, *Proceedings of the International Conference on Simulation of Adaptive Behavior*, From Animals to Animats, pages 346–355, Cambridge, Massachusetts. The MIT Press.

Topoff, H. R., editor (1971). *Army Ants: A Study in Social Organization.* Freeman, San Francisco.

Tresilian, J. R. (1991). Empirical and theoretical issues in the perception of time to contact. *The Journal of Experimental Biology*, 17:865–876.

Tsai, R. Y. and Huang, T. S. (1984). Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):13–27.

Tsai, R. Y., Huang, T. S., and Zhu, W.-L. (1982). Estimating three-dimensional motion parameters of a rigid planar patch, II: Singular value decomposition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-30(4):525–534.

Tsuji, S. and Zheng, J. Y. (1990a). Panoramic representation of scenes for route understanding. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 161–167.

Tsuji, S. and Zheng, J. Y. (1990b). Qualitative representation of scenes along a route. In *Proceedings of the AAAI Workshop on Qualitative Vision*, pages 67–71, Boston.

Tsuji, S. and Zheng, J. Y. (1992). Panoramic representation of scenes for route recognition by a mobile robot. *International Journal of Computer Vision*, 9(1):55–77.

Turvey, M. T., Shaw, R. E., Reed, E. S., and Mace, W. M. (1981). Ecological laws of perceiving and acting: In reply to Fodor and Pylyshyn. *Cognition*, 9:237–304.

Ullman, S. (1979). *The Interpretation of Visual Motion.* The MIT Press, Cambridge, Massachusetts.

van Iersel, J. J. A. and van den Assem, J. (1964). Aspects of orientation in the digger wasp *Bembix rostrata*. *Animal Behaviour Suppl.*, 1:145–162.

von Frisch, K. (1967). *The Dance Language and Orientation of Bees.* The Belknap

Press of Harvard University Press, Cambridge, Massachusetts.

Wagner, H. (1982). Flow-field variables trigger landing in files. *Nature*, 297:147–148.

Wallace, G. K. (1959). Visual scanning in the desert locust *Schistocerca gregaria* Forskål. *The Journal of Experimental Biology*, 36:512–525.

Warren, W. H. (1988). Action modes and laws of control for the visual guidance of action. In Meijer, O. G. and Roth, H., editors, *Complex movement behaviour: The motor-action controversy*, pages 339–390. Elsevier Science Publishers, North-Holland.

Watson, A. B. and Ahumada, A. J. (1985). Model of human visual-motion sensing. *Journal of the Optical Society of America A: Optics and Image Science*, 2(2):322–342.

Webb, B. and Hallam, J. (1996). How to attract females: further robotic experiments in cricket phonotaxis. In Maes, P., Matarić, M. J., Meyer, J. A., Pollack, J., and Wilson, S. W., editors, *Proceedings of the International Conference on Simulation of Adaptive Behavior*, volume 4 of *From Animals to Animats*. The MIT Press, Cambridge, Massachusetts.

Weber, K., Venkatesh, S., and Kieronska, D. (1994). Insect based navigation and its application to the autonomous control of mobile robots. In *Proceedings of the International Conference on Automation, Robotics and Computer Vision.*

Weber, K., Venkatesh, S., and Srinivasan, M. V. (1995). Entomologically based algorithms for the autonomous control of mobile robots. Technical Report 6/1995, Department of Computer Science, Curtin University of Technology, Perth, Western Australia.

Weber, K., Venkatesh, S., and Srinivasan, M. V. (1996a). Insect inspired behaviours for the autonomous control of mobile robots. In *Proceedings of the International Conference on Pattern Recognition*, pages 156–160, Vienna, Austria.

Weber, K., Venkatesh, S., and Srinivasan, M. V. (1996b). Landmark based corridor discrimination. In *Proceedings of the International Conference on Automation, Robotics and Computer Vision*, pages 272–276.

Weber, K., Venkatesh, S., and Srinivasan, M. V. (1997). Insect inspired behaviours for the autonomous control of mobile robots. In Srinivasan, M. V. and Venkatesh, S., editors, *From Living Eyes to Seeing Machines*, chapter 11, pages 226–248. Oxford University Press.

Weber, K., Venkatesh, S., and Srinivasan, M. V. (1998). An insect-based approach

to robotic homing. In Jain, A. K., Venkatesh, S., and Lovell, B. C., editors, *Proceedings of the International Conference on Pattern Recognition*, volume I, pages 297–299, Brisbane, Australia. IAPR, IEEE Computer Society Press.

Weber, K., Venkatesh, S., and Srinivasan, M. V. (1999). Insect-inspired robotic homing. *Adaptive Behavior*, 7(1):65–97.

Wehner, R. (1972). Visual orientation performances of desert ants, *Cataglyphis bicolor*, towards astromenotactic directions and horizon landmarks. In Galler, S. R., editor, *Proceedings of the AIBS Symposium on Animal Orientation and Navigation*, pages 421–436, Washington. US Print Office.

Wehner, R. (1976). Polarized-light navigation in insects. *Scientific American*, 235:106–115.

Wehner, R. (1981). Spatial vision in arthropods. In Autrum, H., editor, *Comparative Physiology and Evolution of vision in invertebrates*, pages 287–616. Springer-Verlag, Berlin. Handbook of Sensory Physiology, Vol VII/6c.

Wehner, R. (1983). Celestial and terrestrial navigation: Human strategies—insect strategies. In Huber, F. and Markl, H., editors, *Neuroethology and Behavioural Physiology*, pages 366–381. Springer-Verlag, Berlin.

Wehner, R. (1989). Neurobiology of polarization vision. *Trends in Neuroscience*, 12:353–359.

Wehner, R. (1992). Arthropods. In Papi, F., editor, *Animal Homing*, volume 2 of *Animal Behaviour*, chapter 3, pages 45–144. Chapman and Hall, London.

Wehner, R. (1994). The polarization-vision project: Championing organismic biology. In Schildberger, K. and Elsner, N., editors, *Neural Basis of Behavioural Adaptations*, volume 39 of *Forschritte der Zoologie*. Gustav Fischer Verlag, Stuttgart.

Wehner, R., Bleuler, S., Nievergelt, C., and Shah, D. (1990). Bees navigate by using vectors and routes rather than maps. *Naturwissenschaften*, 77:479–482.

Wehner, R. and Flatt, I. (1972). *The visual orientation of desert ants,* Cataglyphis bicolor*, by means of terrestrial cues*, pages 295–302. Springer, Berlin Heidelberg New York.

Wehner, R., Harkness, R. D., and Schmid-Hempel, P. (1983). *Foraging Strategies in Individually Searching Ants Cataglyphis bicolor (Hymenoptera: Formicidae)*, volume 1 of *Information Processing in Animals*. Gustav Fisher Verlag, Stuttgart.

Wehner, R. and Menzel, R. (1990). Do insects have cognitive maps? *Annual Review Neuroscience*, 13:403–414.

Wehner, R., Michel, B., and Antonsen, P. (1996). Visual navigation in insects: Coupling of egocentric and geocentric information. *The Journal of Experimental Biology*, 199:129–140.

Wehner, R. and Räber, F. (1979). Visual spatial memory in desert ants, *cataglyphis bicolor* (hymenoptera: Formicidae). *Experientia*, 35:1569–1571.

Wehner, R. and Srinivasan, M. V. (1981). Searching behaviour of desert ants, genus *Cataglyphis* (formicidae, hymenoptera). *Journal of Comparative Physiology A*, 142:315–338.

Wehrhahn, C., Hausen, K., and Zanker, J. (1981). Is the landing response of the housefly driven by motion of a flowfield? *Biological Cybernetics*, 41:91–99.

Whiteside, T. C. D. and Samuel, G. D. (1970). Blur zone. *Nature*, 225:94–95.

Wilkes, D., Dickinson, S., Rivlin, E., and Basri, R. (1994). Navigation based on a network of 2D images. In *Proceedings of the International Conference on Pattern Recognition*, volume A, pages 373–378. IEEE Computer Society Press.

Wilkins, D. E. (1984). Domain-independent planning: Representation and plan generation. *Artificial Intelligence*, 22:269–301.

Wilson, D. M. (1966). Insect walking. *Annual Review of Entomology*, 11:103–122.

Wilson, E. O. (1971). *The Insect Societies*. The Belknap Press of Harvard University Press, Cambridge, Massachusetts.

Winograd, T. (1972). *Understanding Natural Language*. Academic Press, New York.

Winston, P. H. (1972). The MIT robot. *Machine Intelligence*, 7:431–463.

Wolf, R., Voss, A., Hein, S., and Heisenberg, M. (1992). Can a fly ride a bicycle? *Philosophical Transactions of the Royal Society of London B*, 337:261–269.

Wooton, R. J. (1990). The mechanical design of insect wings. *Scientific American*, pages 66–72.

Yagi, Y., Nishizawa, Y., and Yachida, M. (1995). Map-based navigation for a mobile robot with omnidirectional image sensor COPIS. *IEEE Transactions on Robotics and Automation*, 11(5):634–648.

Yakimovsky, Y. and Cunningham, R. (1978). A system for extracting three-dimensional measurements from a stereo pair of TV cameras. *Computer Graphics and Image Processing*, 7:195–210.

Yamauchi, B. and Nelson, R. (1991). A behavior-based architecture for robots using real-time vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1822–1827. IEEE Computer Society Press.

Yap, C. K. (1987). Algorithmic motion planning. In Schwartz, J. T. and Yap, C. K., editors, *Algorithmic and Geometric Aspects of Robotics*, volume 1, pages 95–143. Laurence Earlbaum.

Zapata, R., Lépinay, P., and Thompson, P. (1994). Reactive behaviours of fast mobile robots. *Journal of Robotic Systems*, 11(1):13–20.

Zeil, J. (1993a). Orientation flights of solitary wasps (*Cerceris*; Sphecidae; Hymenoptera): I. Description of flight. *Journal of Comparative Physiology A*, 172:189–205.

Zeil, J. (1993b). Orientation flights of solitary wasps (*Cerceris*; Sphecidae; Hymenoptera): II. Similarities between orientation and return flights and the use of motion parallax. *Journal of Comparative Physiology A*, 172:207–222.

Zeil, J., Kelber, A., and Voss, R. (1996). Structure and function of learning flights in ground-nesting bees and wasps. *The Journal of Experimental Biology*, 199:245–252.

Zhang, Z. and Faugeras, O. D. (1992). Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *International Journal of Computer Vision*, 7(3):211–241.

Zhang, Z., Weiss, R., and Riseman, E. M. (1991). Feature matching in 360° waveforms for robot navigation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 742–743, Los Alamitos, California. IEEE Computer Society, IEEE Computer Society Press.

Zheng, Q. and Chellappa, R. (1993). Automatic feature point extraction and tracking in image sequences for unknown image motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 335–339. IEEE Computer Society Press.

Zill, S. N. and Seyfarth, E.-A. (1996). Exoskeletal sensors for walking. *Scientific American*, 274:70–74.

Zinner, H. (1986). Determining the kinematic parameters of a moving imaging sensor by processing spatial and temporal intensity changes. *Journal of the Optical Society of America A: Optics and Image Science*, 3(9):1512–1517.

Zipser, D. (1988). Biologically plausible models of place recognition and goal location. In McClelland, J. L. and Rumelhart, D. E., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2: Psychological and Biological Models, chapter 23, pages 432–470. The MIT Press.