THE AUSTRALIAN
NATIONAL UNIVERSITY

A Multi-Level Spatial Memory for Vision-Based Mobile Robot Localisation

A thesis submitted for the degree of
Doctor of Philosophy
of The Australian National University.

Simon Thompson
Bachelor of Information Technology H1 (Australian National University)

August 2002

Statement of Originality

These doctoral studies were conducted under the supervision of Professor Alexander Zelinsky. The work submitted in this thesis is a result of original research carried out by myself, in collaboration with others, while enrolled as a PhD student in the Department of Systems Engineering at the Australian National University. It has not been submitted for any other degree or award.

Simon Thompson

Acknowledgements

First and foremost thanks to my supervisor in this endeavour, Professor Alexander Zelinsky, who keep me on track and provided encouragement and advice throughout. Also credit for building a friendly and professional lab environment which makes the hard yards all that more bearable.

To the others in the Robotic Systems Laboratory , past and present, who contributed so much to that environment and to myself and my work, thank-you very much. Thanks to Gordon Cheng for giving me his "hand me downs" at the start of the thesis, Yoshio Matsumoto for his advice on vision processing in the early days, Sebastian Rougeaux for the widening distribution of his MMX code, Luke Fletcher and David Austin for keeping the robot running just enough of the time, Uwe Zimmer for some reminders about science, and all the administration staff who helped along the way.

Special thanks to Dr. Toshihiro Matsui of the Intelligent Systems Division, Electro-Technical Laboratory and the members of that lab for supporting me for five enjoyable months in Tsukuba, Japan.

Thanks also to Professor Sebastian Thrun and colleagues of the Robot Learning Laboratory at Carnegie Mellon University for kindly donating their robot navigation software to the ANU. This software was used in evaulating the performance of our research.

My most heartfelt thanks and appreciation for my wife, Rieko, without who I would never have made it. And to the as yet nameless one, who's arrival is a blessing of joy and inspiration,

this is for you

Abstract

Robust and reliable navigation systems are required before robots will join us in the real world. The major problem in creating robotic systems which can navigate reliably is that of forming a representation of an arbitrary environment which can provide reliable localisation information. Mobile robot localisation encompasses the conflicting tasks of accurate local position estimation and efficient global localisation. This thesis documents a multi-level spatial representation which is specifically designed to facilitate multiple navigation strategies and to solve the conflicting localisation tasks.

The three levels of spatial representation are directly inspired by three levels of biological organism's navigation systems: those of honeybees, rats and humans. The three levels are: visual landmarks, local space profiles and indirect landmarks, or disambiguating features. Visual landmarks allow for low-level navigation strategies and provide accurate local position information and a unique representation for place discrimination, but matching at this level of representation is expensive. Local space profiles are a minimalistic representation of the extent of local space surrounding a mobile robot and as such cannot provide unambiguous localisation information, but is lightweight and comparatively cheap to match. The level of disambiguating features seeks to actively search out discriminating features in snapshots of the sensory view at places in an attempt to define the feature which most successfully discriminates between the two places. Methods for acquiring multi-level representations of places in the environment are developed as well as methods for constructing topological maps with transitions and cycles.

The visual landmark representation is shown to provide accurate local position estimation and inefficient, but accurate, global localisation in a topological map. The mid-level representation of local space profiles can constrain the global localisation search of lower levels of representation by 60%. By combining the visual landmark and local space representations, a system which performs continuous local position estimation and global localisation is produced. This system is applied to the kidnapped robot problem in a topological map with 100% local position estimation recovery.

Contents

List of Figures

List of Tables

Chapter 1

Overview

For most of the past few decades robots have been restricted to manufacturing plants, research labs or other highly controlled areas. Today we are seeing the emergence of robots in our everyday environments, our work, our homes and our playgrounds. In order for robots to function in the real world, it is usually necessary for them to move about the environment in a meaningful manner. Office assistant or tour guide robots must move through complex, dynamic environments reliably to fulfill their given tasks. Robot pets, to interact more meaningfully with humans, must move more freely about our homes. Thus navigation is a crucial behaviour which underpins mobile robot functionality in almost all application areas, and its current limitations are also a major restriction in robot placement in the real world.

Tour guide robots have been tested in various museums around the world, but usually have to be monitored and cannot perform complex tours in dynamic environments for extended periods of time. Robot pets are cute and amusing, but cannot do more than wander around blindly at this stage. Robust and reliable navigation systems are needed for robots to perform unsupervised tasks throughout their active life. The research field of mobile robot navigation is concerned with developing robot systems that can autonomously perform navigation tasks reliably over an extended period of time.

1.1 Mobile Robot Navigation

The problem of mobile robot navigation can be broken into three subcomponents: mapping, localisation and planning. It is now a relatively easy task for a robot to plan and physically execute a path between two points in a known environment, the challenge now is to form an internal representation of the robot's environment and with this representation, work out where in the environment the robot is currently located. This involves two sub-components of the navigation task: that of mapping and localisation.

Traditional approaches to mobile robot mapping and localisation predominantly try to map out the total environment in order to navigate in it successfully, building complete two or even three dimensional spatial maps from recorded sensor data (Moravec and Elfes, 1985; Thrun, Burgard and Fox, 1998). Robots can then, given they know their own location, plan and execute any trajectory within the map. Recently robotic navigation systems inspired by successful biological systems, have concentrated on extracting only the necessary information from sensors to achieve specific navigation tasks such as homing to a place.

1.1.1 Multiple Levels of Spatial Representation

When thinking about the navigation task, and in particular our own navigation strategies, it is clear that it is necessary to employ a variety of different strategies for different situations. No one strategy will successfully work in all situations, and each different strategy may require different sensory cues, levels of processing and spatial memory to work. Spatial memory refers to the internal views of the spatial characteristics of the environment that organisms form in order to interact with the environment. This decomposition of navigation is reflected in the literature (see Chapter 2) where a large number of navigation systems have been proposed, each solving a specific, or small group of navigation tasks, by forming specific spatial representations of the world.

An example of employing different navigation strategies can be observed in a situation as simple as a person moving down a corridor and passing through a particular door. Navigation along the corridor simply involves keeping to the center of the corridor and moving forward. A complete internal representation of the corridor and the relative position of the person moving along it is not necessary. But the person must perceive the existence of doors and must be able to distinguish the goal door from others along the corridor. After detecting the door the person must have some strategy for passing through it (assuming it is open), for example by focusing on the door frame on one side of the opening and plotting a trajectory relative to that frame. If the opening is very small, the person might also place part of their body against the frame to provide more accurate position information. These different navigational strategies require different internal spatial representations of the environment. So even seemingly simple navigation tasks can require multiple navigation strategies and also multiple levels of internal representation about the environment and also the mobile agent's location within it.

1.1.2 Biological Inspiration

The above example highlights the insights which can be gained from observing successful biological navigation systems. Even simple biological systems such as honeybees are capable of performing complex and precise navigation behaviours with comparatively simple computational mechanisms. This thesis aims to use such systems as inspiration when designing practical robotic systems, while remembering the limitations inherent in current robotic systems and sensors.

1.1.3 The Localisation Problem

Multiple navigation strategies and multiple levels of spatial representation require multiple levels of localisation. A system must be able to perform localisation on the various levels of spatial representation in order to trigger the appropriate navigation behaviour. In the example above, the corridor navigation behaviour requires localisation to a corridor like environment, while the door entry behaviour requires corridor and door recognition, plus an estimate of the robot's position relative to the two door jambs.

The need for multiple levels of internal representation and their localisation equivalent is highlighted by the two conflicting extremes of the mobile robot localisation problem:

1. *Global Localisation:* the robot system must be able to identify its current general

location within its internal map representation. Only when it's current location is known can it plan a path to the goal position. This form of localisation requires a search of the entire internal map to find the most likely location in the map from which the robot captured the current sensor data. Because the entire map must be searched this procedure is computationally expensive.

2. *Local Positioning:* the robot system must accurately identify and maintain an estimate of its position relative to some local reference frame in order to perform precise navigation movements. This form of localisation requires a fine search of a localised area within the internal map, in order to precisely identify the most likely location from which the robot captured the current sensor data. This search is locally constrained and therefore can be performed relatively cheaply.

In order to perform global localisation efficiently, the robot system's internal representation of the world must be sparse in order for the localisation task to be computationally tractable. In order to perform accurate local position estimation, the internal representation must be fine enough to provide enough positional cues to execute precise navigational behaviours.

An additional problem in mobile robot localisation is how to represent the robot's belief that it is in a particular location. Obviously the degree to which a robot is certain it is in a specific location is going to affect it's navigation behaviour. By introducing multiple levels of spatial representation, the representation of this belief is complicated and a method for combining the various levels of localisation into a central belief is required. The conflict between global localisation and local positioning highlights the need for multiple levels of spatial memory.

The kidnapped robot problem is often used as a bench mark for mobile robot localisation systems to evaluate their ability to solve the above challenges. In such a situation the robot is navigating through a pre-mapped environment with a high belief of its location, when it is suddenly kidnapped. The robot is subsequently released at another location in the map with the belief it is still at the original location. The task for the localisation system is to recognise it has been moved and subsequently re-localise its position within its internal map. This involves adjusting the robot's belief in the presence of conflicting sensor data, performing global localisation and recovering accurate local position estimation.

1.2 Principle Objectives

This thesis proposes to develop a biologically inspired robot localisation system which facilitates the use of multiple navigation strategies by representing the world using different levels of spatial memory. The various levels of spatial representation and subsequent potential navigation strategies should be inspired by navigation and spatial representation methods in both robotic and biological systems.

In particular the proposed system should use multiple levels of spatial representation to:

1. *Solve the conflicting localisation problems of global localisation and local position estimation.* To do so the multiple levels of spatial representation must enable efficient global localisation while maintaining accurate local position estimation.

2. *Maintain a central belief as to the robots position within its internal map.* This central belief should reflect the localisation information produced by the various levels of spatial representation.

3. *Solve the kidnapped robot problem.* The kidnapped robot problem is largely unsolved and can be used as an ultimate test for practical mobile robot localisation systems.

By using multi-level spatial representation and localisation, a navigation system can operate in the real world, robustly and reliably, switching navigation strategies according to the situation. In doing so the goal of robots joining us in our everyday environments is nearer. The primary focus of this thesis will be on the required levels of internal representation or spatial memory of a robot, and how multiple levels can work together to allow for accurate and efficient localisation.

1.3 Outline

This chapter has provided an overview of the main concepts that will be be developed in this thesis and a preview of the main objectives that this thesis seeks to achieve. The remainder of this chapter outlines chapter by chapter the contents of the rest of this thesis.

- *Mobile Robot Navigation*

  The concepts involved in navigation and the state of present research in this field are presented in Chapter 2. In particular details of the general navigation problem and a hierarchy of navigation behaviours is identified. The idea of navigation as a robotic behaviour and what can realistically be achieved in this area is discussed. A brief description of robot systems as biological analogies is presented and the design philosophy behind the current approach is defined. From there, some background in the field of mobile robot mapping and localisation is presented and examples from biological and robotic systems are described.

- *Multi-Level Spatial Representation*

  This leads to the proposal of the multi-level spatial memory as a solution for the mapping and localisation problem in Chapter 3. Specific details of a three level system are introduced and motivated. The three levels of representation are: visual landmarks, local space profiles and disambiguating features. In addition the robot system which is used in experiments validating the proposed multi-level spatial representation is briefly introduced..

- *Visual Landmarks*

  Chapter 4 presents the details of representing places in a topological map by visual landmarks from panoramic images. Landmarks are chosen to represent a place by identifying potential landmarks from a static image and evaluating their tracking performance over a biologically inspired Turn Back and Look movement. Visual landmarks are selected which are locally unique and can be reliably recognised over the area surrounding the reference position from which the place is learnt. A method for estimating the depth of landmarks is also presented and experimental results validating the depth estimation as well as landmark recognisability are reported.

- *Local Space Profiles*

  The details of the proposed second level of spatial representation are presented in Chapter 5. A method for determining the extent of local space surrounding the robot from a panoramic image is presented. The profile of local space from a given panoramic image can be used to represent the local space surrounding a particular place in a topological map. A method for inexpensive matching between local space profiles is also presented.

- *Disambiguating Features*

  Disambiguating features can be used to discriminate between two places in a topological map in pathological localisation situations. A panoramic image captured from the reference position where the place was learnt provides a snap shot memory from which disambiguating features can later be extracted. Chapter 6 describes the process of extracting these features from two panoramic images, and using them to discriminate between places in a topological map.

- *Topological Maps*

  Chapter 7 describes a method for constructing topological maps using the multiple levels of spatial representation presented earlier. In particular the chapter deals with when to acquire a new place in the topological map and how to form transitions between nodes in the topological map. Experiments verify the system's ability to construct topological maps including the detection of cycles in the topological graph structure, denoting circular routes in the physical environment.

- *Local Positioning*

  Chapter 8 describes how the multi-level spatial representation can be used to perform mobile robot local position estimation within topological maps. A method of accurately estimating local position within places is presented and experimentally validated. The estimate can then be passed between places in the topological map using the transition information to achieve position tracking. An experiment validates position tracking and the systems ability to overcome odometric drift.

- *Global Localisation*

  Chapter 9 describes how the multi-level spatial representation can be used to solve the the conflicting problems of global localisation and local position estimation. Global localisation can be solved with the visual landmark level of representation, although this proves to be computationally expensive. The second and third levels of spatial representation can be used to restrict the global localisation search space, drastically reducing the computation costs, and improving the efficiency of the global localisation search. A method of combining local position estimation with global localisation is introduced and the complete system is successfully applied to the kidnapped robot problem. Experimental results supporting these methods are reported.

- *Conclusions*

  To conclude this thesis, Chapter 10 summarises the contributions and key points of this thesis in addition to some ideas for further work.

Chapter 2

Review of Mobile Robot Navigation

The field of mobile robot navigation literature is wide and varied, with the topics of path planning, localisation and navigation forming large sub-disciplines of their own. This chapter aims to introduce the field of mobile robot navigation, focusing on the two sub-components of mapping and localisation, and discuss the approach the proposed system will take in designing a solution. A brief background in traditional approaches to these problems is given and examples of biological and biometric approaches are reported.

Section 2.1 details the general navigation problem and identifies a hierarchy of navigation behaviours. Section 2.2 explores the idea of navigation as a robotic behaviour and discusses what can realistically be achieved. Section 2.3 provides a brief description of robot systems as biological analogies. Section 2.4 describes the philosophy used in this thesis when designing a solution to the localisation problem. Section 2.5 briefly describes some of the more common approaches to navigation and details the recent advances in those areas. In particular the use of topological maps versus metric maps is discussed. In addition, the topic of probabilistic reasoning is introduced as a way of representing a robot's internal beliefs and allowing for simultaneous multiple hypotheses. Section 2.6 the navigation systems of various levels of biological organisms are presented, specifically honeybees, rats and humans. Their spatial memory and methods of localisation and mapping are discussed in relation to the level of navigation behaviour they exhibit. Section 2.7 goes on to report several biomimetic approaches to mobile robot navigation which exhibit simple navigational strategies such as homing and route following. These approaches are directly inspired by experimental results from the biological sciences. The chapter ends with a summary of the key concepts covered in this chapter and reiterates the current problems in the field of mobile robot localisation.

2.1 What is Navigation?

In order to construct a system that can navigate successfully, it is important to define exactly what navigation is, and what this means in the context of robotics. It is more so, given that mobile robot literature has in the past, referred to a number of seemingly different robot behaviours under the the same banner as navigation. Franz and Mallot (2000) provide a discussion of the definition of navigation in regard to robotic systems.

Navigation historically refers to the guidance of ocean going vessels. The word navigation itself is derived from Latin, *navis*: ship, and *agere*: to drive. In this context the process of navigation has three steps:

1. Determine the ships position on a chart.
2. Relate the current position to destination.
3. Set course of the ship.

(Levitt and Lawton, 1990) describe the same process in robotics, defining navigation as answering the three questions:

1. "Where am I?"
2. "Where are other places with respect to me?"
3. "How do I get to other places from here?"

Franz and Mallot (2000) argue that many biological organisms are able to navigate without answering all of these questions. Instead the only question that needs answering is "How do I reach the goal?". This demands a broader definition of navigation:

> Navigation is the process of determining and maintaining a course or trajectory to a goal location (Franz and Mallot, 2000).

Navigation Behaviour	*Behavioural Prerequisite*	*Navigation Competence*
Local Navigation		
Search	Goal recognition	Finding the goal without active goal orientation
Direction-following	Align course with local direction	Finding the goal from one direction
Aiming	Keep goal in front	Finding a salient goal from a catchment area
Guidance	Attain spatial relation to the surrounding objects	Finding a goal defined by its relation to the surroundings
Way-finding		
Recognition triggered response	Association sensory pattern-action	Following fixed routes
Topological navigation	Route integration, route planning	Flexible concatenation of route segments
Survey navigation	Embedding into a common reference frame	Finding paths over novel terrain

Table 2.1: Franz and Mallot's (2000) hierarchy of navigation behaviours.

Using this definition, therefore, the only requirements for navigation are to be able to move in free space and to determine whether or not the goal has been reached. Navigation then, incorporates all spatial behaviours that include motion with reference to a goal location, and excludes other behaviours such as exploration, foraging, obstacle avoidance or course stabilisation. Furthermore Franz and Mallot (2000) divide navigation behaviours into two groups: local navigation and way-finding.

Local navigation only requires the recognition of one goal location. The robot chooses its actions based on the current sensory input and internal state, without needing representations of places or objects that lie beyond the sensory horizon. Way-finding by comparison, requires the recognition of several places and representation of places beyond the sensory horizon. Within these categories lie navigation behaviours of various complexity and competences. In the next section, Franz and Mallot's (2000) navigation behaviour hierarchy is described.

2.1.1 Navigation Behaviour Hierarchy

When discussing navigation and navigation behaviours it is useful to classify the difference between strategies in a hierarchy of increasing complexity and competences. This also allows for comparison of biological and traditional robotic navigations systems. The categorisation of navigation behaviours into local navigation and way-finding described above lead Franz and Mallot (2000) to extend Trullier, Wiener, Berthoz and Meyer's (1997) navigation hierarchy as shown in Table 2.1. Local navigation includes such behaviours as search, direction following, aiming and guidance, listed in increasing order of competence. Way finding is divided into three levels: recognition triggered response, topological navigation and survey navigation. A robot at a given level of local navigation or of way-finding has all the navigation competences of the lower levels in that group, but if an agent is capable of way-finding, that does not automatically mean that it is proficient at all levels of local navigation. The individual navigation behaviours are described in the following sections.

Search

Search is the simplest form of navigation. It entails searching for the goal position, and as such only requires the robot to be able to move and to recognise the goal area. Obviously this type of navigation is not efficient and an agent can take a long time to randomly encounter the goal. Due to the light amount of information needed during navigation this strategy can be used as a backup when all other strategies fail.

Direction-following and path integration

Direction-following requires the robot to follow a course determined by some locally available instructions. This direction may be extracted from external cues such as a landmark, or internal cues such as a compass. Using direction-following then, an agent can navigate to a goal if it is on the path and can accurately extract direction information. If a robot is not on the path or deviates from the path due to noisy data, then this type of navigation will fail. If the distance to the goal is also known, a robot can use path integration (odometry) to determine whether it has passed the goal location without detection and should adopt another navigation strategy. Also if a robot using path integration deviates from a trail, it can return to its original position as long as the path integration has not been disrupted.

Aiming

Aiming is similar to direction following, but involves navigating towards a specific goal or beacon and does not require the robot to follow a specific trail to the goal. As long as the robot can perceive the beacon then the robot can navigate towards it. A goal location then, has a catchment area or navigation field surrounding it, in which the beacon can be perceived. Goal locations are limited to those locations which have a suitable beacon.

Guidance

If a suitable beacon is not present at a goal location, a robot can use other features in the surrounding environment to guide its movements towards the goal. This guidance is provided by the relation between the current perceptual view and a memorised view of surrounding landmarks or features. By moving so as to make these two views the same, the robot is in fact navigating towards the goal location.

Recognition-triggered response

The previous three navigation strategies are all local strategies. These methods only work when the information needed to navigate to the goal position lies within the current sensory horizon. Recognition-triggered responses associate current sensory information with an action to perform. Actions are typically local navigation strategies, connecting the place where the response was triggered to a local goal position. By building a sequence of these recognition-triggered responses, robots can form *routes* incorporating a number of locations. It is important to note that this form of wayfinding does not involve knowledge of the future route, simply the triggering of a particular action given the current sensory information. If any of the local navigation strategies employed at places along the route fail, then the robot must resort to a simpler strategy, such as *search*, until its perception of the environment triggers another response. If the robot has a number of goal locations, each one must have its own route and associated recognition/response representation of places, and no integration between routes is possible.

Topological navigation

Topological navigation allows for route integration by building a spatial representation of places along routes independent of goals. If the robot can detect that a place is common to two or more routes then it can pass from one route to another, thus reaching a goal along a previously untravelled route. The integration of routes in this fashion forms a topological map of the environment. The robot can then navigate to a goal location by planning a path through the integrated routes.

Survey Navigation

Topological navigation though, does not allow novel route generation. Using such a navigation strategy the robot must travel along segments of previously learnt routes. In order to generate novel routes, learnt places must be embedded into a common frame of reference. This requires the spatial relationships between all learnt places to be available to the robot from any given place. In survey navigation, this information is available, allowing the robot to plan novel routes through previously unvisited areas of the environment by inferring potential connections between places from their embedded spatial relationships.

2.1.2 The Components of Navigation

Assuming that a navigation hierarchy exists, and that various levels of competence can be observed in biological organisms, it is necessary to identify the functional subcomponents necessary in order for navigation strategies to work. In particular what are the components which are common across all navigation strategies? By identifying such common components, a better understanding of how different navigation strategies are related and how it might be possible to switch between the strategies is reached.

The Components of Traditional Navigation Systems

The process described by the initial nautical definition of navigation, and the traditional definition in the field of mobile robotics, can be divided easily into three subtasks: mapping, localisation and planning.

- *Mapping*: the acquisition and maintenance of an internal representation of knowledge about the spatial characteristics of the environment.

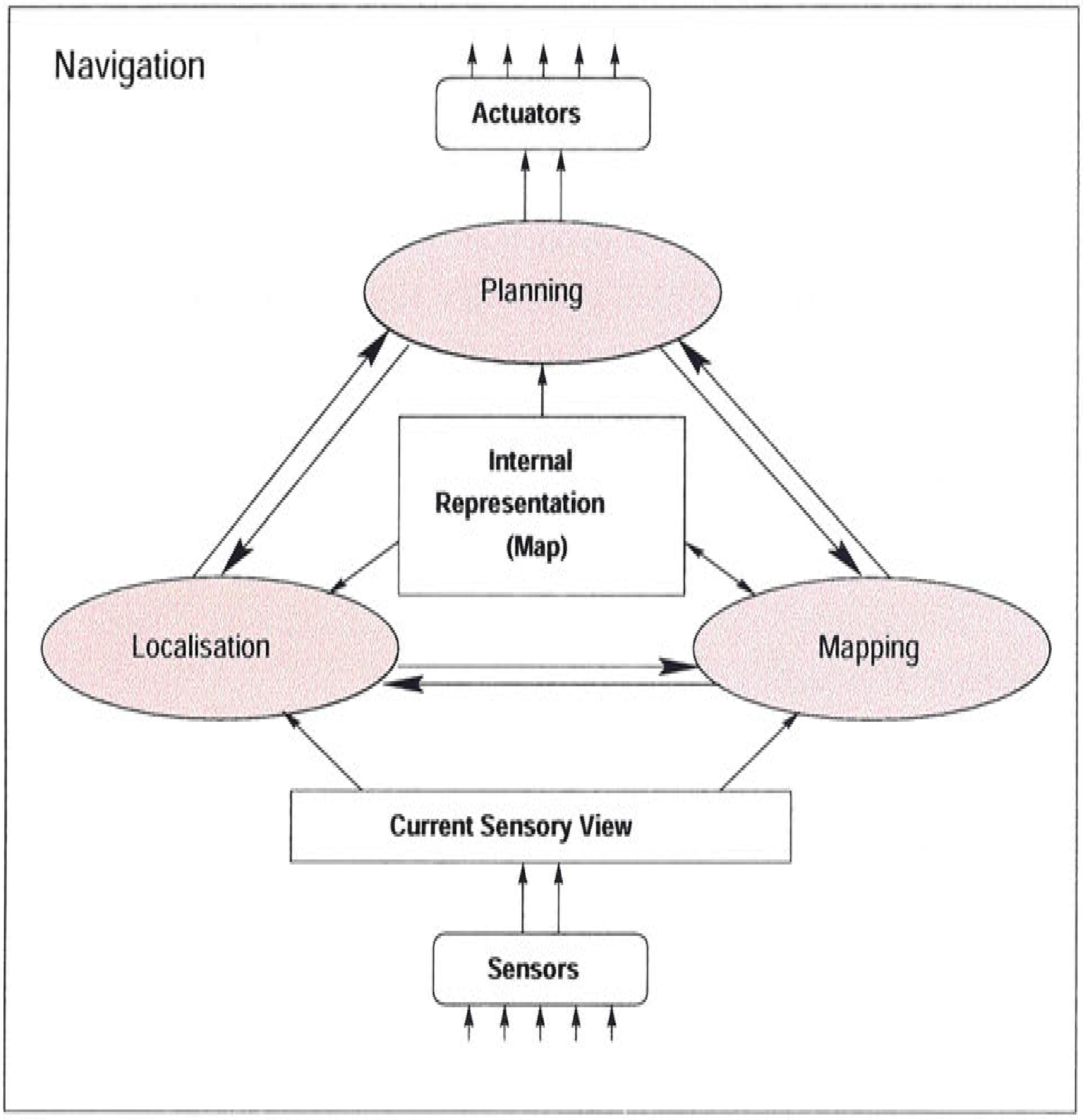


Figure 2.1: The components and tools of the navigation task.

- *Localisation*: the process of determining a robot's current position relative to the internal map, from current sensor readings of the local environment.

- *Planning*: path planning based on the internal knowledge the robot has about its own state, the map, and its desired behaviour.

For a robot that is navigating through a dynamic environment, continually re-evaluating its internal state, goals and sensory view, these sub-tasks must work interdependently to produce successful navigation behaviour. Figure 2.1 illustrates this relationship between the subcomponents. In this figure the behavioural subcomponents are illustrated using the grey filled ovals.

As mentioned earlier, the three sub-tasks rely on internal knowledge in order to function. Mapping, localisation and planning all require internal representations of the current environment and the previously learnt/experienced environment. These two representations can be referred to as the *current sensory view* and the *map* and their relationship to the functional components is shown in Figure 2.1 as the rectangular boxes. The current sensory view is an abstraction of the raw sensor data representing the current environment. The map is an internal abstraction of multiple previous sensory views of the environment. In general the level of abstraction involved in both representations has strong consequences in the degree of navigation competence a robot can achieve. For example, being in possession of a highly detailed map, such as a floor plan, allows a robot to plan to navigate anywhere within the floor plan. While only knowing if one has reached a goal or not, such as in the search behaviour described above, will only allow navigation to a single place, with almost no planning in between.

The Components of Hierarchical Navigation Systems

By redefining navigation as above, Franz and Mallot (2000) move away from the traditional decomposition of the navigation process, saying, in reference to local navigation:

> ... this notion of navigation does not imply that the current location must be recognised, nor that a map-like representation must be used to find the goal (Franz and Mallot, 2000).

This definition only requires the subcomponents *recognise goal* and *move in free space*, rather than the traditional mapping, localisation and planning. It also argues against the need for an internal map.

But this is not the case. The local navigation behaviours can be described as simplified instances of the mapping, localisation and planning view of navigation. Taking the simplest of the local navigation behaviours, i.e *Search*, as the case in point, it can be shown that all the traditional components of navigation are present. Mapping is present because the robot must still define some relation between its own internal representation of the goal and its current sensory input. It does not matter that the "map" only distinguishes between "found goal" and "have not found goal" in the simplest case. Localisation is present as a decision as to the robot's location based

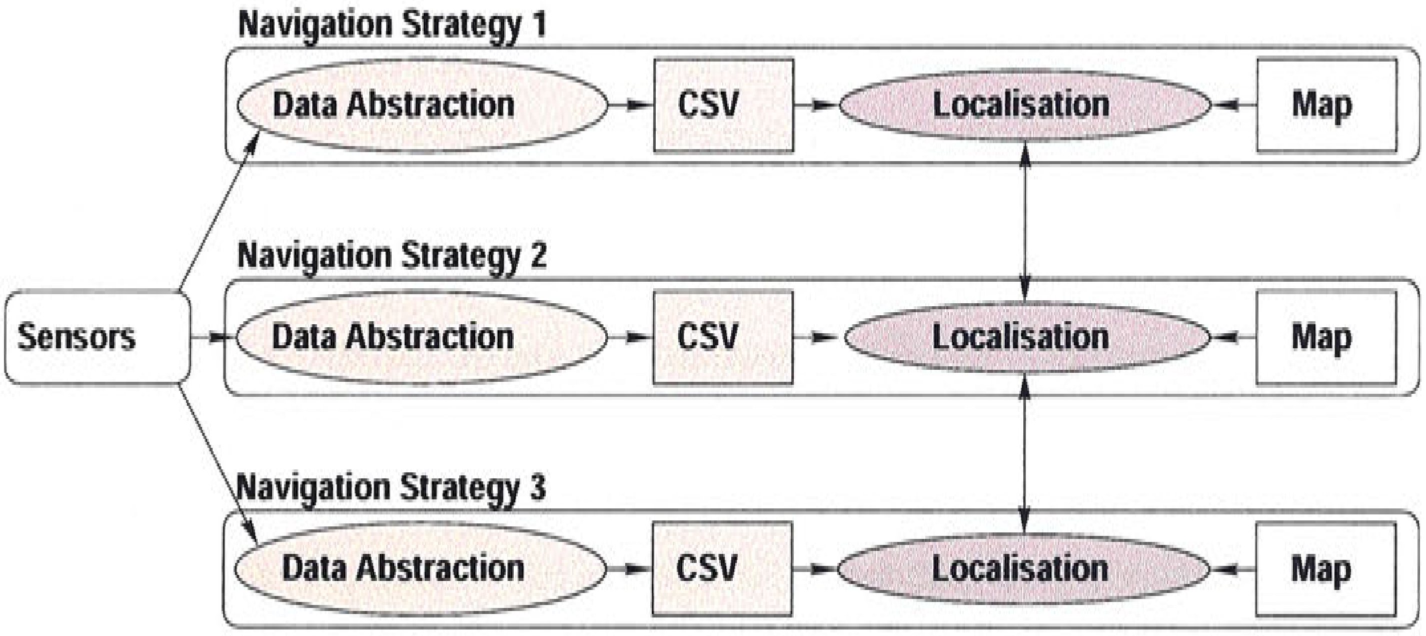


Figure 2.2: Localisation in a multiple strategy navigation system

on the internal representation and its current sensory input must be made. For the *Search* behaviour, this could be just the decision "I am not at the goal yet". Planning is present since once having made a decision as to its location the robot performs certain actions to reach its goal. In the case of "I have not reached the goal yet" this could be to simply move in a random direction.

Navigation systems which display different navigation behaviours then, do not necessarily have different sub-components. In addition, all need internal representations of the current sensory view and the internal map. These sub-components and representations, however, might be simplified and might only exist implicitly in the navigation behaviour rather than explicitly in the robot's reasoning.

2.1.3 A Consistent Framework for Navigation Strategies

This important distinction as described above, although seeming insignificant and semantic, allows the various navigation behaviours, including the two major categories of local navigation and way-finding, to share common behavioural subcomponents. This common behavioural framework means that multiple navigation behaviours in a robot can help reinforce the robot's knowledge about mapping, localisation and planning, and can contribute to more successful navigation behaviour by the robot.

Consider the case of a robot navigating down a corridor to a particular door employing two navigation strategies. Strategy 1 uses odometry to estimate the distance travelled down the corridor, with the robot reaching the goal by travelling the known distance between the start location and the goal door location. Strategy 2 uses door and office number recognition to form a topological map, with the goal being reached when it detects a door with the correct office number. Both strategies will work in the right situations, each having a form of internal representation and associated mapping and localisation behaviours. In general the office number recognition solution would be superior due to the noise inherent in odometry measurements. But what would happen if the office numbers were removed and all doors looked identical? The robot would then not be able to identify the correct door. However by combining the door recognition and odometry navigation strategies, the robot would be able to locate a door at approximately the correct distance down the corridor. This solution requires the robot to have multiple spatial maps of the same environment, or a spatial representation which facilitates the use of different strategies, and also for the combination of localisation information extracted from both navigation strategies.

In a robot that exhibits multiple navigation strategies then, it is desirable for it to have a common behavioural framework for navigation strategies and it also must have the capacity for multiple levels of internal representation for both the current sensory view and also the internal map. Figure 2.2 shows multiple navigation strategies and levels of internal representation for the localisation component of navigation. The general idea is that a robot captures different forms of spatial representation from the current sensory view, and can use these multiple levels to form maps from which localisation can be performed. The localisation process can then be occurring at different levels, with each using different cues from the environment and information passing between levels, resulting in an agent with more robust and reliable knowledge about it's position in it's environment.

2.2 Navigation as a Robotic Behaviour

Navigation, as described above, is a complex behaviour, with interdependent subcomponents operating on a number of levels. To achieve robust and reliable navigation a mobile robot system must model the complexity and multi-level representations of biological organisms. Given that robot and biological systems are inherently different, design and computationally wise, what level of navigation functionality can realistically be expected from robots?

On a basic level, all the navigation strategies previously described can be achieved by mobile robots. Given a particular environment, a particular navigation strategy and a particular choice of internal representation, robot systems can be built that can successfully navigate for most of the time within those environments. The problem is moving to more general or large scale environments, where single strategies and internal representations will not always work. By adding more strategies and internal representations it is proposed that a mobile robot will be able to function more robustly and more reliably in more generalised, larger environments, but adds additional computation costs to the navigation process. Given the real time constraints of navigating a mobile robot in the real world, computation must be kept to a minimum and internal representations that do not scale well with the size of the environment are not appropriate.

The bulk of computation in navigation systems is in the continuous processes of interpretation of sensory data and localisation. In robotic systems, localisation is the computationally intensive task of matching the current sensory view to the internal map. The current position is determined by the location in the map which matches best with the current view. Therefore the choice of internal representations greatly affects the efficiency of the localisation process.

Another problem that arises when developing robotic navigation systems is that of the robot maintaining a belief of where it is. How does a robot interpret the results of matching and decide where it is located in it's internal map? A related question is how does it change it's belief if it is wrong? Biological organisms are excellent at forming a belief, acting appropriately and switching to alternate hypotheses when necessary.

Biological navigation systems, and in particular those of humans, do not exist in the knowledge vacuum that robotic systems do. Humans have access to a level of knowledge and structure of their environment far above that of a robotic system. This leads to far more complex navigation cues and representations, which leads to a more generic navigation system. A form of knowledge representation is needed to combine localisation information from different navigation strategies over time to form a belief of current position, while continuously forming and maintaining alternate hypotheses.

In addition, biological systems exist in a highly goal driven and purposive organism. Bees can navigate from their hive to flowers and back again. Success can be measured by collecting pollen and delivering it to the larvae in the hive. The bee has knowledge (not necessarily explicit knowledge) of when these things are complete. A robot with a particular mission in an office environment, has very limited understanding of when the goal is complete, excluding any hardwired flags that are programmed in. For example for a mission of navigating to a particular office and deliver a verbal message requires significant knowledge aside from that required by simple navigation to be able to achieve the task. In this research, the navigation task to be performed by the robot, in absence of sematic knowledge, is point to point navigation within an internal map.

2.3 Biological Analogies in Robotics

Robot systems have traditionally been anthropormorphised by popular culture. When people imagined robots of the future they inevitably imagined human like creatures such as C3P0 from Star Wars. Even when robotic helpers were not humanoid they invariably took the shape of other familiar creatures such as dogs and cats. Recently, after decades of exposure to non-humanoid robots such as factory robots, there has been a resurgence of research directed towards humanoid and biologically inspired robots in general. There are several reasons for this persistence in the anthropomorhication of robotics, ranging from aesthetics and psychology to imitation and functionality:

1. Human expectation of appearance and behaviour of "intelligent systems".
2. Facilitation of human-robot interaction.
3. Robots should operate in the real world which is designed and built for humans and their companions.
4. Biological systems can accomplish the desired task so why not imitate them?

All these reasons hold true when considering artificial navigation systems. Mobile robots must navigate in our environment, they must behave predictably in order to successfully navigate with and around humans. Biological navigation systems have these qualities as well as being successful at navigation. There are also examples of biological systems with relatively simple computational components that can perform complex and robust navigation behaviours. Bees for example, perform complex takeoff, route following, goal detection and landing procedures with limited computational resources. It is hoped by further studying biological organisms, successful, low-cost behavioural algorithms can be developed.

Throughout this thesis, the navigation problem is considered from a biological viewpoint, with biological systems capable of equivalent navigation behaviour inspiring solutions to particular problems. At a deeper design level, another important lesson can be gained from the study of biological organisms. The development of biological solutions are dependent on the past evolutionary pathway, with the development of new structures and algorithms dependent on the existing functionality. Likewise with robot systems the choice or availability of sensors and processing capability leads to certain representations, algorithms and behaviours being more appropriate than others.

Another lesson for robot researchers comes from the field of ethology. Biological systems do not exist by themselves. They form part of a complicated eco-system and are highly evolved to fill an ecological niche. As such they cannot be viewed as separate from their environment. For robotic systems, this means when designing robotic systems, not only think about what behaviours the robot should be capable of but also what environment will the robot exist in, and how does this impact on the robots desired behaviours. Obviously a robot relying on a passive vision system will not be much use if it is required to work in dark environments. More subtly, a robot navigating by extracting certain features from the environment, will not be able to function in environments where those features are sparse.

However it must be remembered that robots are not biological organisms. Robot subsystems have not evolved in parallel, they have been been specifically designed and built for varying tasks, which may or may not include the task at hand. Robotic systems also suffer from a low knowledge base when compared to biological systems. As mentioned in Section 2.2 robots do not have the deep level of semantic knowledge about the world that some biological organisms have. They do not have the wide array of multi-modal cognitive information that biological systems experience and cannot be expected to perform at similar levels until this type of knowledge is available to them. Therefore, while it can be extremely useful to draw inspiration from successful biological systems, it must be remembered that there are inherent design differences between robotic and biological systems, which can limit the applicability of biologically inspired solutions, and also limit the degree of functionality that robotic systems can achieve.

2.4 Philosophy of Approach

In the construction of complex systems such as the navigation systems for robots, a consistent design philosophy can lead to more complete, internally consistent and ultimately superior solutions. Also the formation and application of such a design philosophy itself can provide novel insights into the system under construction.

In the system under consideration, just as the behavioural algorithms can imitate those of successful biological systems, so the current design approach can take inspiration from the ultimate designer, nature.

In looking for a solution to navigation then, it is important to view the system as a whole, incorporating its environment and its evolutionary history. The biological solution, for example, is dependent on the particular organisms physical structure, the type of sensors available, its level of knowledge about the environment it is in, and its own behavioural and instinctual predispositions. In robotics, this means identifying available (and suitable) hardware, sensors, existing software, as well as the nature of the desired behaviour and the environment in which the robot is to exist. From there individual subcomponents of the system can be designed that work within the limitations, and utilise the advantages of the existing system. Subcomponents of the system should also be designed with knowledge that they are interdependent behaviours and each contribute to the success or failure of the others.

This research then, aims to develop a solution to the navigation problem that follows this design philosophy, using interdependent subcomponents designed with respect to the system as a whole to produce a successful navigation system.

2.5 Traditional Navigation

Traditional navigation approaches have concentrated on producing robots capable of exhibiting movement above all other considerations. Concerns such as mapping and localisation were largely ignored in deference to building systems that could move from point A to point B. Maps and starting location were considered *a priori* knowledge, and operational environments were limited to single rooms or corridor environments. These assumptions and limitations lead to navigation methods that relied on large amounts of sensor data and computationally expensive matching algorithms for localisation.

Recently, the short-comings of these approaches when confronted with large complex environments has led to the emergence of mapping and localisation as important research areas. The questions being how to form efficient representations of large scale maps and how to best search those maps when localising. These questions are examined in detail with respect to mobile robot literature, and two general forms of maps have been identified, namely metric maps and topological maps. Contemporary research in this field is looking at forming hierarchical maps and integrating the two approaches, and some of this recent work is reported.

2.5.1 Mapping and Localisation

Robot localisation using self-made maps in arbitrary environments is an enduring problem for the field of mobile robotics. Solving this problem involves matching an internal representation of the world which has been abstracted/interpreted from sensor data, with an internal representation of the current view of the world which again, is abstracted/interpreted from sensor data. Accurate solutions have been found by constraining the basic problem, such as by introducing a priori information (pre-existing maps), using artificial landmarks, limiting the size of the environment, or reducing the required localisation accuracy. These approaches try to limit the complexity of the matching task between the environment and the map, by limiting the complexity of one or the other. Typically limiting the complexity of the environment results in scalability problems while limiting the complexity of the map results in coarse localisation accuracy.

Sensors and Map Representation

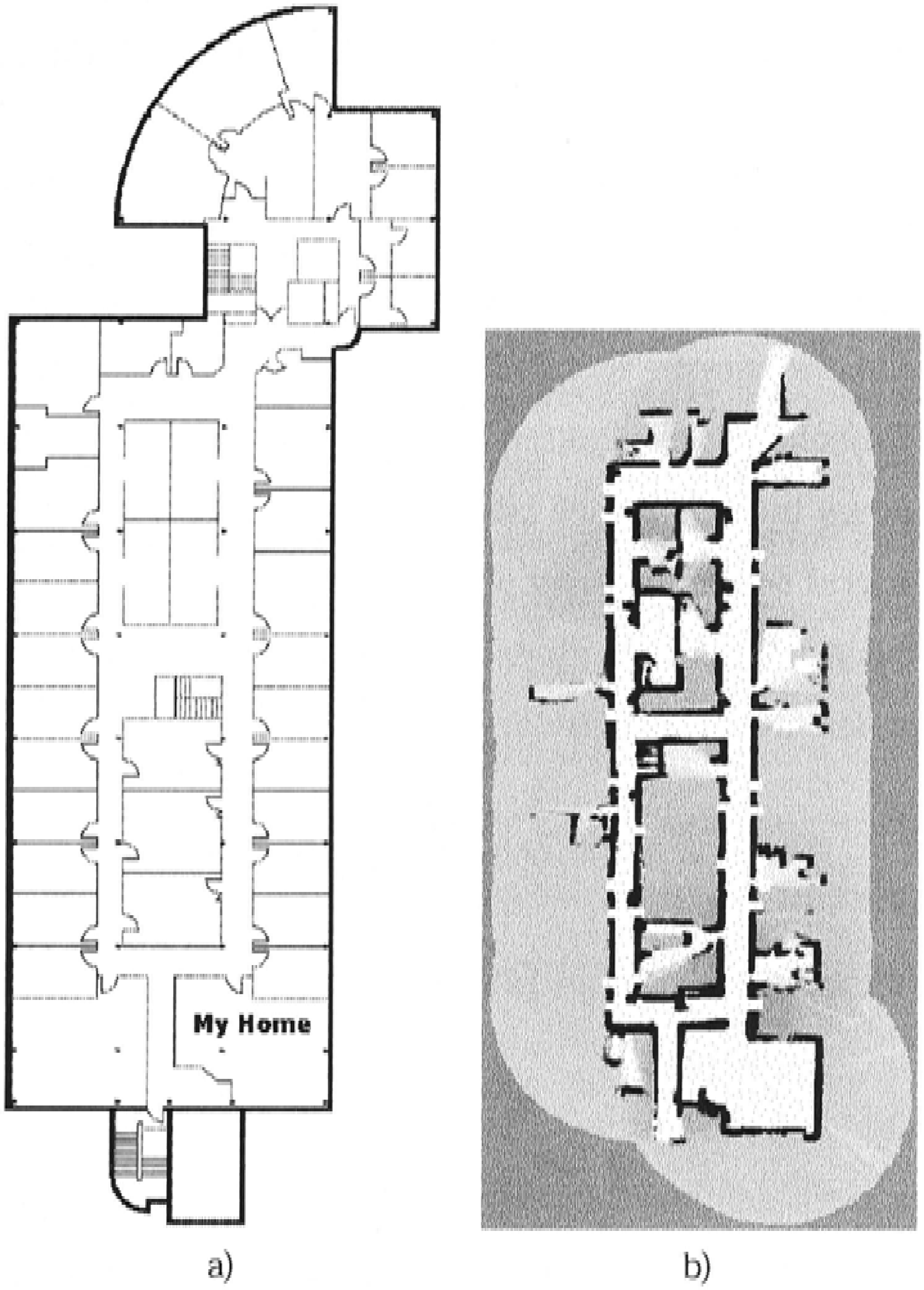
Representations which have a low level of abstraction (close to the sensor level) typically require a large amount of information to map the environment, and also have high computation cost associated with the matching for localisation. On the other hand, representations which have a high level of abstraction require less storage and matching, but require a lot of computation and complexity to form the level of abstraction from the raw sensor data, first when acquiring the map, and second, when localising. They also can abstract the view of the environment to such a degree that details which uniquely identify the current robot position can be lost. Thus the type of sensors used, the data representation and data matching methods are crucial for the tasks of mapping and localisation.

There are traditionally two opposing approaches for mapping: metric versus topological (Borenstein, Everett and Feng, 1996). Metric maps form highly detailed, low level representations of the environment, while topological maps try to abstract information into a less data intensive, semantically meaningful form.

Metric Maps

Metric maps try to capture the exact two or three dimensional structure of the environment. They are easily comprehensible by humans and are similar to maps we use in our everyday lives such as street directories, architectural plans and atlases. This is not surprising as their use in robot navigation originated from when maps were assumed to be known, and humans provided the maps to robots. For example compare the maps of the robotics laboratory at the Australian National University in Figure 2.3, an architectural drawing and a metric map constructed from laser data captured on a mobile robot (map produced by Thrun, Beetz, Bennewitz, Burgard, Cremers, Dellaert, Fox, ahnel, Rosenberg, Roy, Schulte and Schulz's (2000) mapping system).

Metric maps provide a fine level of representation and can therefore be used to track the position of the robot accurately. The amount of information such maps contain however can limit the scale of the maps, and increases the difficulty of the global localisation task. Also when constructing metric maps the accrual of odometry errors can lead to large distortions in the spatial representation.



a)

b)

Figure 2.3: Metric maps of the Robotic Systems Laboratory. Part a) shows a architectural drawing, part b) a map constructed from laser range finder data.

Metric maps are typically constructed using range information from sensors such as sonar and laser range finders. Range sensors provide accurate range information from which accurate metric maps can be constructed and accurate local robot position estimates made. Range sensors and metric maps, however, are often incapable of perceiving and representing details in the environment which can provide important cues for the global localisation task.

An example of metric maps are the evidence grids proposed by Moravec and Elfes (1985). In this approach maps are constructed from sonar data by evaluating the probability that each grid cell in the map is occupied or vacant. Over time the evidence for each cell builds up as more and more sensor data is processed. When performing localisation by matching between a current local map and the pre-built map of the global environment, the sum of the products of corresponding cells is calculated. In general, global localisation in evidence grid based approaches, involves the matching of current sensor data with all possible positions in the global grid. Thus the computational cost of localisation in evidence grids grows with the size of the environment and the granularity of the grids.

As an example, say a particle filter (see Chapter 8) based localisation system using a metric map, could perform local position estimation with 1000 particles in time $T_M = 10ms$, taking $0.01ms$ to evaluate the probability of each particle. Particle filters attempt to approximate the Probability Distribution Function (PDF) of the robot position over the space of robot poses by importance based resampling of a set of sample poses, or particles. Given a grid cell granularity of $10cm$ and $1\deg$, a $10 \times 10m$ map would require 3,600,000 matches to perform an exhaustive global localisation search, again taking $0.01ms$ to evaulate the probability of each grid cell:

$$gridsize = 3,600,000 = 100 \times 100 \times 360$$

resulting in a computation time of $36s$ and a ratio of global localisation to local position estimation time of:

$$\frac{36,000}{10} = 3,600$$

Global localisation in a metric map may not require an exhaustive search space, but the computation requirements for global localisation grow quickly. In addition, if each cell in the metric map requires only 1 byte for storage, the metric map would require $\sim 3MB$ of memory.

Thus scalability is a problem in all metric map approaches, whether the representation be evidence grids, collections of line segments, or geometrically correct maps of landmarks or features.

Topological Maps

Topological maps are typically coarse, graph like representations of the environment. In these representations, nodes correspond to significant places in the environment while edges in the graph correspond to transitions between places. Example topological maps are shown in Figure 2.4, both with exact metric transitions (a), and with metrically incorrect but still functional transitions (b). Topological maps, theoretically, scale well with the size of the environment and lend themselves well to graph based path planning and navigation. They typically have a coarse representation which limits the accuracy of position tracking and in practice do not scale efficiently due to the problems of discriminating between nodes (Thrun, Gutmann, Fox, Burgard and Kuipers, 1998). While the coarse spatial representation limits the local positioning accuracy, it conversely simplifies the global localisation task, that is, localising the robot from an unknown start position. In practice however it is hard to find a representation which uniquely identifies each place, and such representations are usually complex or time consuming to extract, and localisation can become time consuming. Because topological maps define the transitions relative to each place, the odometry error does not accrue, and when navigating from the map odometry estimates can be reset at the recognition of each place.

Topological maps typically use rich, data intensive sensors such as vision. The richness of data captured by a sensor such as vision allows the robot system to extract an internal representation from the environment that uniquely identifies the location from which the sensor data was captured. As mentioned above the ability to form unique representations of locations in the environment is crucial to the concept of topological mapping.

A good example of topological maps is presented by Kuipers and Byun (1991). Forming one level of the spatial semantic hierarchy, the topological map is constructed of distinctive places connected by distinctive travel paths. It is proposed that distinctive places are determined by a distinctiveness measure defined on a subset of sensory features which are maximised at the place. When the robot is in the neighbourhood of the place, it can use the distinctiveness measure to perform a hill climbing search and so move towards the distinctive place.

Just as distinctive places are maximised over a two dimension area, distinctive paths can be identified by a distinctiveness criterion which defines a set of one dimensional points. Thus Kuipers and Byun (1991) report the ability to be able to navigate down the mid-line of a corridor and along the edge of a room, but cannot navigate in open

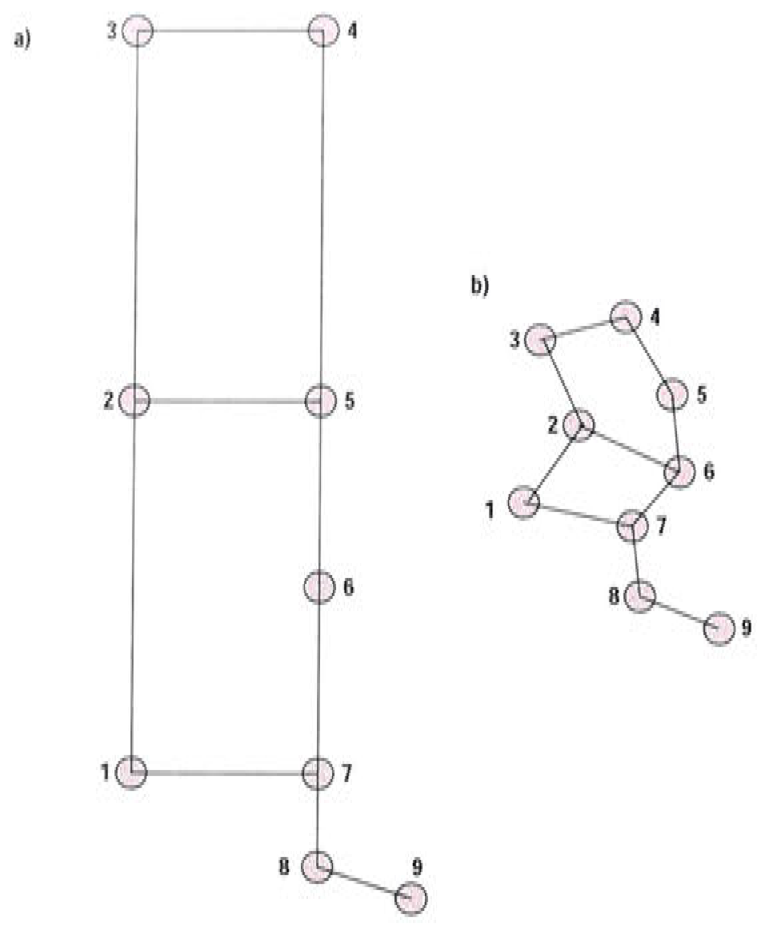


Figure 2.4: An example topological map of the Robotic Systems Laboratory, showing exact metric transitions a), and metrically incorrect transitions

space. This implementation of a topological map does not address the global localisation as individual places are not unique, but it does eliminate odometry errors by periodically resetting an odometric counter.

In comparison to the metric map example, a topological map with a node density of 1 per $1m^2$ would require 100 matches to perform an exhaustive global localisation search. Matching computation time is $100 \times T_T s$, where T_T is the time taken to match one instance of a given topological representation. The value of T_T is dependent on the computation costs of the particular topological representation used. The ratio between local position estimation and global localisation in such a topological map is

$$\frac{100T_T}{T_T} = 100$$

which is much less than the equivalent metric ratio.

In general the matching time cost for instances of topological representations are expensive, with $T_T > T_M$, however as the size of the map grows the spareness of representation overcomes the initial overheads:

$$T_T(N/G_T) < T_M(N/G_M) \tag{2.1}$$

for large values of map size N, where G_T and G_M are the granularity of the topological and metric map representations.

Typical topological localisation would result in a robot position estimation accuracy of less than $1m^2$. The memory requirements also depend on the chosen topological representation.

Topological maps scale better with the size of the environment than metric maps, but have high matching overheads and are comparatively inaccurate.

Integration of Topological and Metric Maps

Since both topological and metric maps have their disadvantages recently research has started to examine integrating the two approaches to take advantage of the strengths of both approaches. For example the topological system reported above (Kuipers and Byun, 1991) was just one level of a spatial semantic hierarchy. Below the topological representation, was a path following control mechanism, reminiscent of the local navigation strategies, while at the highest level was a metric representation, which was refined over time as the robot familiarised itself with the environment. Following this, Thrun, Gutmann, Fox, Burgard and Kuipers (1998) report an approach to mapping which uses a topological map to correct for large odometric errors in a metric map, under a common statistical framework. Asoh and Matsui (1999) take this idea further, directly integrating topological and metric maps. Both these latter approaches however, still generate fine grained maps, and as such suffer from the scalability problems discussed above, while Kuipers and Byun (1991) does not address global localisation problems at all.

Rather than focusing on combining the two methods, it makes sense to look at what each approach gives us and how that contributes to the goal of autonomy in mobile robots. As mentioned above, topological maps give good scalability and simplify global localisation if places are sufficiently unique. Metric maps, by contrast, provide accurate position tracking. Global localisation is needed in the case where the robot is started in an unknown position and must localise itself in the map. Position tracking is needed for accurate navigation and path planning tasks. On closer inspection the accuracy required for the navigation and path planning tasks is only needed relative to the local environment. Examples include navigating around an obstacle or positioning the robot near a docking station. As long as the robot has an accurate position estimate relative to a local point in the environment, it does not need metric information about the rest of the map.

Limiting the Localisation Search Space

Given that a major problem in navigation is efficient global localisation, it is desirable to limit the required computation as much as possible when performing such a task. One method of doing this is by reducing the localisation search space in the map. In fact once the global position has been estimated, the tasks of initial local position estimation and subsequent position tracking are just localisation in a vastly reduced search space. The traditional approach to this problem has just been to assume that the robot has a rough estimate as to its location and search local space for fine positioning. This assumption is invalid in the kidnapped robot problem.

Recently efforts have been made to overcome this problem in a more satisfying manner. Dellaert, Fox, Burgard and Thrun (1999) use a particle filter to subsample potential locations of the robot from a grid based metric map. The samples can then be shifted to areas in which it is likely that the robot is positioned according to current sensor data. In practice however, it takes a large number of samples to effectively cover a large environment and the scalability problem arises once more. This led Jensfelt, Wijk, Austin and Andrsson (2000) to propose several augmentations to the algorithm to more efficiently search the localisation space. Further discussion on this topic is provided below.

Another approach is to construct a topological map using metric maps of particular locations (Courtney and Jain, 1994). Metric localisation tasks can then be constrained by first localising on the topological map, and accumulated error on the metric level is eliminated due to the decomposition of the metric representation. Of course in this approach, two separate searches must take place, one on the topological level and one on the metric level.

2.5.2 Data, Features and Landmarks

The above approaches to metric mapping involved creating maps of accurate two dimensional space and localisation was achieved by matching complete sensor data readings with that map. This led to scalability problems when robots were presented with large environments. Such representations which utilise the entire sensor view in both mapping and localisation can be found in topological mapping as well. An example of this is Principle Component Analysis, where the entire sensory view at a place is analysed to find its maximally discriminating components of an image in relation to views of other places (Krose, Vlassis, Bunschoten and Motomura, 2001). Another example is reported by Matsumoto, Inaba and Inoue (1997), who use raw, low resolution panoramic images to represent places along a learnt route. Global localisation can be achieved by exhaustively matching a current panoramic view to a repository of stored views.

Another way to approach this problem is to explicitly pick out distinctive information from the sensory views and use this distinctive information to construct maps and provide localisation cues. In metric maps this means building maps with these distinctive features positioned precisely on a two dimensional map (Leonard and Durrant-Whyte, 1991a) (Jensfelt and Christensen, 1999), and localising by detecting those features and calculating relative robot pose. In the case of topological maps, this information would be used to uniquely identify individual places, and localisation would occur by finding those particular distinctive features in the current view.

This approach has traditionally been called *landmark navigation*. But inside that all-encompassing title there are really two distinct methodologies. The first is to extract information that is truly unique to that area. This is analogous to representing a place using a monument such as the Eiffel tower, a truly distinct landmark. A location can be defined in relation to such a landmark or set of landmarks. This is true *landmark navigation*, and such unique regions of the environment can be referred to as *landmarks*. Alternatively, distinctive features in the environment can be extracted which occur regularly in the environment, such as the identification of door frames, line segments, or corner points. In this case a location can be defined in relation a specific configuration of local features. These repeating regions shall be referred to as *features* and navigation using such cues as *feature navigation*.

The only distinction between *landmarks* and *features* that this research makes is that *landmarks* are unique to one specific location in the environment, whereas a specific *feature* may occur at a number of locations throughout the environment.

Selecting Landmarks and Features

What makes a distinct landmark? When are features appropriate and when are landmarks? Recent work has lent towards letting the robot select appropriate landmarks rather than humans defining them. An example of this is found in Thrun's (1998) description of a landmark learning algorithm. This algorithm, called Bayesian Landmark Learning (BaLL), let a neural network extract landmarks from visual images of the environment, by minimising the error in position estimation of the robot. Thrun (1998) found that a variety of landmarks were found, including, doors, dark spots, wall colour, hallways, and blackboards. In one spot the almost invisible (to humans) change in illumination of an otherwise visually sparse wall was chosen as a landmark.

By introducing different levels of uncertainty into the robot's position estimate, Thrun (1998) found they could manipulate the characteristics of landmarks chosen. By introducing high levels of position estimation uncertainty, and thus forcing the robot to solve the global localisation problem, the algorithm would tend to select landmarks such as wall colour that differentiate large portions of the environment. If the network was trained with low levels of position estimation uncertainty then the algorithm tends to select local features such as doors or hallways.

This study shows that the appropriateness of landmarks depends on the task at hand. Global localisation requires distinct landmarks in order to differentiate places while local position tracking can be achieved using local features. Of course over time the observation of a specific set of features in a particular configuration can lead to global localisation, but in general, encountering distinct landmarks achieves this much more efficiently.

The Matching Problem

The matching problem is the problem of comparing the current sensory view with a robot's internal map representation to perform localisation. This process is performed irrespective of the level of representation: data, features or landmarks.

Matching raw data in topological or appearance based representations such as entire panoramic images (Matsumoto et al., 1997) is computationally expensive while matching with sparse data (Krose et al., 2001) leads to ambiguity in localisation due to spatial aliasing and data association problems.

A landmark representation can identify unique regions of the sensor data which helps eliminate spatial aliasing. A feature representation is less unique and data association is a problem. The case for landmarks over features is not so cut and dry. The matching process upon which localisation depends is different for landmarks and features. Features, such as door jams, can be identified in the current sensory view before matching occurs. Thus it is only necessary to match the current configuration of features with the configuration of features in the map. Landmark matching on the other hand, requires all landmarks in the map to be matched against every possible position in the current sensory view. So while landmarks are distinct the matching process can be computationally prohibitive compared to feature matching, even when taking into account the extra processing involved in identifying features.

An ideal situation would be to combine the attention attraction capabilities of features with the distinctiveness of landmarks. Sim and Dudek (1999) attempts to do this by identifying potential landmarks in a visual scene by detecting features comprised of dense regions of edge elements. Groups of landmarks are then selected based on the principle components analysis of potential landmarks from different viewpoints. Localisation then involves first extracting the edge density maxima features and matching them with the landmark sets contained in the map. This approach to landmark selection and matching appears promising, but the experimental environment was very small and the system was not tested on a real robot, with associated real time localisation constraints.

2.5.3 Internal Belief Representation

Robots operate in real world environments and as such they have to deal with noisy sensor data. Localisation achieved by simply matching discrete sensor data at each time will result in poor estimates of robot position due to the errors introduced by noisy and ambiguous data. One of the biggest criticisms of early navigation systems was their inability to handle uncertainty in the robot's understanding of the environment. This meant that there was no strategy for integrating sensor measurements over time, nor for recovering from incorrect localisation estimates and no means to maintain multiple hypotheses as to the robot position. This problem can be stated as the problem of a robot maintaining an internal belief as to its own current position.

The previous description of purposive robotic behaviour identifies point to point navigation as a goal of this research. An internal representation of belief as to where a robot is located in the environment should be able to inform decisions as to whether this goal has been reached. This decision typically can not explicitly be made with traditional topological representations. Topological representations are typically driven by behavioural based methods of navigation. In these approaches the sensor data drives the robot control towards a goal state (Khatib, 1985)(Latombe, 1991)(Arkin, 1998). Behaviour based robot architectures typically have a behaviour arbitration system which makes it difficult to define just when a goal has been accomplished.

This section looks at internal belief representation in metric based maps and how such representations impact on the localisation task.

Gaussian Estimations

A common method for representing the internal belief of a robot's location within an internal map is using a Gaussian Probability Density Function (PDF). The mean of the Gaussian PDF represents the estimate of the robot's position in the environment while the variance represents the uncertainty associated with that measure. The noise evident in a system is likely to be composed of noise from many small sources (Maybeck, 1979). A Gaussian probability density closely approximates the summation of many small sources of noise, regardless of the shape of the constituent densities.

The Kalman filter is a recursive algorithm for the optimal estimation of linear system's which assumes that noise in the system is Gaussian and white (Maybeck, 1979). Maybeck (1979) describes the key characteristics of a Kalman filter by stating:

> ...[a Kalman filter] processes all available measurements, regardless of their precision, to estimate the current value of the variables of interest, with use

> of (1) knowledge of the system and measurement device dynamics, (2) the statistical description of the system noises, measurement errors, and uncertainty in the dynamics model, and (3) any available information about the initial conditions of the variables of interest.

Durrant-Whyte (1994) provides a guide to performing mobile robot pose tracking using a Kalman Filter. Using a Kalman filter Leonard and Durrant-Whyte (1991a) incorporated noisy observations of features into an estimate of robot pose, given an *a priori* estimate. An application of the Kalman filter to a mapping and localisation problem is presented in Chapter 4.

The Kalman Filter approach to internal belief representation can handle noisy sensor data and produce an optimal estimate of local position, but assumes a relatively accurate knowledge of the starting position. Therefore the use of Kalman filters for position estimation is not suitable for global localisation. Also if the position estimate diverges from the true position, it cannot re-localise as the distribution model is uni-modal.

Summation of Gaussian PDF's can be used to model multi-modal distributions for pose tracking. Jensfelt and Christensen (1999) use a Gaussian PDF to represent the position of a mobile robot in a minimalistic metric feature map. As ambiguous sensor data is introduced the system spawns off alternate hypotheses of robot location. These hypotheses also are characterised as Gaussian PDF. The summation of the Gaussian PDF's for each hypothesis represents the mobile robot position PDF. An extended Kalman filter is used to update the multiple Gaussian hypotheses. As a new hypothesis is created for every ambiguous sensor reading, the hypothesis tree becomes large and pruning is required.

Jensfelt and Christensen (1999) use the multiple Gaussian hypotheses approach to perform global localisation. Detection of features in the sensor data generate robot pose hypotheses. The summation of the hypotheses' Gaussian PDF's gives a global PDF. Due to the minimalistic features and careful management of the hypothesis tree, global localisation can be performed with a suitable exploration strategy. The minimalistic feature map, however, can result in long exploration routes before all sources of ambiguity in robot localisation are eliminated. Jensfelt and Christensen (1999) also use a manually constructed map.

Arbitrary Probability Density Function Estimation

Although the summation of multiple Gaussian PDF's can be used to construct an arbitrary PDF over a search space, this approach introduces extra problems such as hypothesis management to limit the amount of generated hypotheses. This can be avoided by attempting to represent the robot's position belief as a probability density function over the entire space of possible robot locations. Of course this problem is not tractable for real-world environments and some approximation of the probability density over the environment must be made in order for the technique to be useful for mobile robot localisation.

Position probability grids discretise the environment and define an approximation of the PDF (Burgard, Fox, Henning and Schmidt, 1996). Each cell in a position probability grid represents the probability that the robot is in that particular location in the map. This representation is computationally expensive to update, as the probability of each cell must be calculated for every localisation step. The computational cost grows exponentially with the granularity of the grid.

Particle filter approaches attempt to approximate the probability density function by use of sampling methods. The approximation is achieved by distributing samples or "particles" throughout the search space, and recursively reselecting with replacement those samples which the sensor model predicts are more likely based on current observations. This technique has been applied to mobile robot localisation in grid based maps by Dellaert, Fox, Burgard and Thrun (1999) using laser range data, and Dellaert, Burgard, Fox and Thrun (1999) using visual data, under the names of Monte-Carlo localisation and the Condensation algorithm (from Isard and Blake's (1998) work with computer vision) respectively. Using this method allows for the formation of multimodal hypotheses as to the robots current position. One problem with pure particle filter approaches, is that the robot can still get centered on wrong hypotheses, and the sample re-selection does not allow for exploration of unseen areas of the probability density function. Jensfelt et al. (2000) propose some adjustments to this re-selection process to allow for the exploration of novel hypotheses.

Global localisation using particle filter approaches is also problematic. A particle set of a size which allows for efficient position estimation generally cannot effectively sample the extent of the search space when attempting global localisation. To perform global localisation effectively, the particle filter would have to sample from the majority of cells in the grid map. In doing so, particle filter localisation would be reverting to a position probability approach.

The tradeoff between accurate position tracking and efficient global localisation can be seen in both the multiple Gaussian hypothesis and the grid based particle filter approaches. The multiple Gaussian hypothesis approach attempts to simplify the matching process (by reducing the number of hypotheses) using a minimalistic feature based method, while the sampled PDF approach of particle filters attempts to sample from a rich map representation. Both forms of location belief are dependent on the underlying internal representation. The dependence on a particular representation limits what is achievable in the conflicting tasks of accurate local position estimation and efficient global localisation.

2.5.4 Simultaneous Localisation and Mapping (SLAM)

Mapping and localisation are not independent. If a robot is constructing a map, it must be accurately localised in order to sensibly integrate current sensor data into an existing map representation. If a robot is to perform localisation, it must have a map representation in which to localise itself. Therefore to to operate in arbitrary environments, a robot system must perform the mapping and localisation behaviours concurrently. In contemporary mobile robot literature this process is referred to as Simultaneous Localisation and Mapping (SLAM).

Smith, Self and Cheeseman (1987) performed SLAM using an extended Kalman Filter by modelling both the robot pose and the map parameters in the state vector. Observations of current sensor data were used to update the map parameters as well as the robot pose. A map could be extended by adding additional parameters to the state vector when exploring previously unseen regions of the environment. This approach does not scale well as each additional map feature adds an extra dimension to the filter.

Many attempts have been made to reduce the complexity of Kalman Filter based SLAM: Leonard and Durrant-Whyte (1991b) restricted the filter update to "confirmed" robot poses or feature observations; Uhlmann (1998) with the use of covariance intersection; and Dissanayake, Durrant-Whyte and Bailey (2000) by removing unnecessary features from the state vector. Jensfelt (2001) suggest a hierarchical form of SLAM which attempts to reduce the complexity of the state vector by breaking the map into a number of smaller more manageable sub maps based on the notion of gateways. While all approaches provide some relief from the map scaling problem, it's effects are only delayed and not eradicated.

Another characteristic of Kalman Filter based SLAM approaches is that all map parameters are estimated relative to a global coordinate system. As a robot moves away from the starting point the uncertainty associated with the robot's position accumulates and is reflected in subsequent feature acquisition and position estimation.

While most approaches to SLAM have relied on range sensors such as sonar or laser, Davidson and Murray (2002) report an approach to SLAM which uses an active vision sensor. This approach uses a corner detector to identify map features and estimates the features position in three dimensional space. Like Dissanayake et al. (2000) a minimal amount of map parameters are maintained. Davidson and Murray (2002) also introduce the idea of incorporating sparse prior knowledge into maps to reduce the problem of increasing uncertainty in the world coordinate system.

The particle filter approach to localisation is not suitable for the SLAM problem. Estimation of map parameters by a search through the space of all possible maps is computationally intractable for all but the smallest of maps. Proponents of the particle filter have performed SLAM using scan-matching to build a map and the particle filter approach to perform localisation (Thrun, Burgard and Fox, 2000).

SLAM attempts to perform map acquisition and maintenance concurrently with localisation. SLAM accentuates the conflict between local position estimation and global localisation by introducing additional computational and representational requirements. Current approaches to SLAM highlight the map scaling problem and reinforce the need for hierarchical representations.

2.5.5 Panoramic Vision

Panoramic vision is a sensor which is becoming popular in the field of mobile robotics. It is also the sensor of choice in our research and as such a brief review of panoramic vision in the literature is necessary. From our own experience we know that vision sensors provide a rich source of information for the localisation task. Single visual snapshots of most environments provide us with enough cues to localise. The extension of the visual scene to a panoramic view increases the amount of information available. This wealth of information has led to the application of panoramic vision sensors to the problem of mobile robot localisation.

Panoramic Vision Sensors

Real time panoramic vision can be achieved by a variety of means. Early methods included the use of fish-eye lenses (Cao, Oh and Hall, 1986) and conic (Yagi and Kawato, 1990) and spherical (Hong, Tan, Pinette, Weiss and Riseman, 1991) mirrors.

More recently, more sophisticated mirror shapes have been suggested. (Yamazawa, Yagi and Yachida, 1995) propose a hyperboloidal mirror shape which has the optical property that all light rays which would pass through the focal point of the mirror are reflected by the mirror surface to pass through a second common focal point. By positioning the camera so its focal point is in this secondary location, the transformation from warped panoramic image to a cylindrical panorama image or common perspective image is simplified (Yamazawa et al., 1995).

Chahl and Srinivasan (1997) report a family of mirror shapes described by polynomial functions which preserve a linear relationship between the changes in the angle of incidence and the angle of reflection in light rays striking the mirror surface. This property simplifies the image transformation and results in a constant angular resolution of elevation in the unwarped cylindrical panoramic image.

These mirror shapes all produce polar camera images in which the pixel density per elevation angle increases with the radius of the polar image. This results in unwarped images which vary in image quality. (Conroy and Moore, 1999) have designed a mirror shape which achieves spatial-resolution invariance over the unwarped image by ensuring pixel density in the polar image is constant irrespective of the angle of elevation (Conroy and Moore, 1999) also design coaxial mirror shapes for panoramic stereo. The depth perception of these stereo systems, however, is very limited due to the small baseline between the two coaxial surfaces.

Applications to Mobile Robotics

There has been many applications of panoramic vision to the problem of mobile robot localisation. A few notable approaches are presented here.

Matsumoto et al. (1997) use the entire unwarped image to build View-Sequence maps of routes through a corridor environment. Images are memorised when the correlation of the current panoramic image with the last stored image falls below a threshold. Localisation is achieved by matching the current panoramic image with the set of stored images. Although this matching process is expensive for global localisation, the panoramic image representation allows for navigation along the learnt route in both directions.

Franz, Scholkopf, Mallot and Bulthoff (1998) used a similar appearance based approach using a pixel average of rings in an warped panoramic image to build a view graph of an environment. Matsui, Thompson and Asoh (2000) also has used a single ring in the warped panoramic image to perform correlation for mobile robot localisation. Both approaches took advantage of the wide field of view of panoramic images and their rotational invariance to extract a minimalistic representation of the environment. Localisation in these appearance based approaches is limited to the granularity of stored views and is susceptible to occlusion.

Paletta, Frintrop and Hertzberg (2001) match overlapping segments of panoramic images with a view based map. This approach overcomes occlusion in the current image by matching each segment of the view and fusing the correlation results. Although occlusion is overcome, this approach introduces additional matching requirements and adds further complexity to the global localisation task.

To limit the complexity of the matching task, panoramic images can be reduced to a minimalistic representation through Principle Components Analysis (PCA) (Vlassis and Krose, 1999). This reduction however often produces features with poor discriminatory ability between robot positions. Vlassis, Bunschoten and Krose (2001) argue that this is because of the unsupervised nature of PCA and report a supervised liner feature extractor which takes advantage of odometric knowledge between image sampling to produce better robot position estimates.

Vlassis, Terwijn and Krose (2002) report the use of particle filter position estimation in a panoramic image PCA feature map. To overcome the problem of in-optimal sampling of the PDF the use of auxiliary particles are proposed. Auxiliary particles are targeted towards the region of the PDF where the prior distribution and the observation likelihood distribution overlap. This targeting of sampling is similar to that proposed by Jensfelt et al. (2000).

Yagi, Hamada, Benson and Yachida (2000) use a panoramic camera to perform alternate pose estimation and map generation with no knowledge of robot motion. The azimuth angle of vertical edges is analysed to obtain a least squares estimate of robot pose and map configuration. This form of SLAM operates in real time on a small map and produces accurate local position estimations with an error of $\sim 10cm$. The map generation step incorporates knowledge about the reliability of map feature estimates in relation to the magnitude of the change in azimuth. Global localisation is not attempted.

Panoramic vision based SLAM has also been reported by (Drocourt, Delahouche, Marhic and Cleretin, 2002). In this approach, stereo panoramic sensors provide location estimations of vertical line features. Like the SLAM approaches reported previously, this approach focuses on local position estimation and does not mention global localisation.

Ulrich and Nourbakhsh (2000) use colour panoramic vision to build topological maps of indoor and outdoor environments. Places in the topological map are represented by image histograms detailing the colour composition of panoramic images from specific locations in the environment. Place recognition is achieved by forming similar histograms of the current panoramic scene and performing matching and a unanimous voting categorisation technique. This method results in very reliable place recognition performance, for very little computational expense, but does not allow for local position estimation within places. The topological map is manually constructed.

Several other panoramic vision based approaches to mobile robot localisation are reported later in this chapter under the section detailing biomimetic approaches.

This section has briefly reviewed the topic of panoramic vision for mobile robot localisation. It has highlighted the ability of panoramic vision to reduce the complexity of internal representations or increase accuracy of local position estimation due to the extent of the visual field. The underlying problem of conflict between the dual goals of local position estimation and global localisation has not been addressed.

2.6 Biological Systems

As mentioned in Chapter 1 the study of biological organisms can provide insights into the design of artificial systems. The study of biological navigation systems is especially advantageous as navigation is a behaviour seen in a wide variety of organisms of differing complexity. It is also an example of terminal addition and as such the navigation behaviours can be seen to build upon each other as the complexity of biological organisms increase. By studying organisms of increasing complexity, the requirements for increasing navigation competences can be identified. For that reason the navigation behaviours, in particular spatial memory, mapping and localisation, of honeybees, rats and humans are presented below.

2.6.1 Honeybee Navigation

Honeybees are relatively simple organisms which are capable of a variety of sophisticated navigation behaviours. Not only can they reliably navigate to and from a food source but they can communicate the direction and distance to such goal locations to other bees in their hive. What sort of internal representations do bees use to store this navigation information and how do they construct these maps and localise themselves from them?

Visual landmarks

There is a large amount of evidence to suggest that in addition to the path integration information of distance and direction to goal, bees memorise a sequence of visual images along a route (Collett, 1996). When traversing the route, a bee can compare the stored images with the current sensory image and set it's course appropriately. It does this by using visual landmarks in three separate ways (Collett, 1996):

1. *Recognising scenes*: if bees are kidnapped from their hive and released in a familiar location, the bee will proceed to fly in the direction of the hive as though it had recalled a goal vector associated with the scene.

2. *Aiming at beacons*: bees can approach distant goals by aiming for familiar landmarks on or near a direct line from start location to finish. By following these beacons bees can move into an area close to the goal positions.

3. *Image matching*: Once a bee gets close to the goal location, it can use stored images to guide them to the goal position by moving so as to align corresponding landmarks in the stored and current images.

These three uses of visual landmarks correspond to the way-finding behaviour recognition-triggered response, and the local navigation behaviours of aiming and guidance. Collett (1996) suggests that the multiple strategies of landmark guidance

> ... may have arisen by an opportunistic grafting of visual pattern learning onto pre-existing navigational and visuo-motor control mechanisms.

Although these behaviours use a common representation of visual landmarks in stored images, different types of landmarks are more appropriate for different behaviours, and thus they are acquired in different ways.

Turn Back and Look Flights

It has been observed that honeybees, when leaving the hive or a feeding location for the first time, make a flight of small arcs looking back at the start position. These systematic manoeuvres have been called flights of learning, or turn back and look movements ((Lehrer, 1993; Zeil, Kelber and Voss, 1996; Collett and Zeil, 1996)). Zeil et al. (1996) provide a discussion of the structure and function of these flights with regard to the different navigation strategies bees employ.

The structure of the flight appears consistent across a variety of species of bees and wasps (an example flight is shown in Figure 2.5. In the case of leaving the hive, these

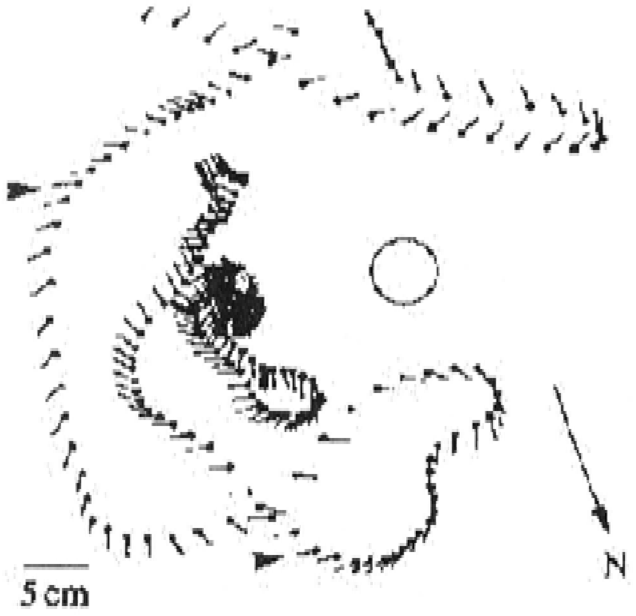


Figure 2.5: The flight path of a typical Turn Back and Look flight. Figure from Zeil et al. (1996).

insects hover for a few seconds looking back at the hive entrance, then fly a series of ever increasing arcs moving away from the hive, while also gaining height at a consistent rate. The last part of the flight involves the bee circling the hive at a height of a few meters.

While flying these arcs the honeybee keeps the start location in the lateral visual field so as to maintain a constant visual image of the hive, except for the extremes of the arc where the bee turns back the other way, momentarily capturing the hive in the high resolution ventral visual field, before it passes into the opposite lateral field. This has the added benefit of requiring only two goal vectors relative to the longitudinal axis of the bees body over the entire flight. At any stage during an arc the goal vector is approximately 45 degrees to the left or right of the bees current orientation.

The full significance of these flights has not yet been determined but the evidence suggests that honeybees use these flights to do two things:

> First, they allow insects to inspect and record the local scene around a goal from distinct positions and along directions that are determined by celestial or earth-based compass cues or by the bearing of close landmarks relative to the goal. Second, they allow insects to acquire information about the true distance of nearby landmarks (Zeil et al., 1996).

Collett (1996) proposed that the purpose of these flights is to learn about landmarks visible from the start location. The distance of landmarks from the goal location determine how appropriate they are for the different navigation strategies. Landmarks that lie along the correct direction and are at relatively far distance from the goal will not change position in the visual field during these flights and as such can be used as homing beacons. Landmarks that are close to the goal, on the other hand, will move through the visual field as the bee moves, allowing for accurate guidance back to the goal location.

Therefore up to three levels of landmarks are learnt. A general view of the local environment may be recorded for scene recognition. Distance landmarks may be learnt in association with goal vectors. Finally, close landmarks may be learnt for accurate positioning near the goal location.

Path Integration

Aiming behaviours work much better if they are used in conjunction with path-integration. If an organism has the ability to determine distance travelled in the direction to a goal, it can use this distance knowledge to cue other search behaviours at the appropriate time. Also in bees the direction and distance can be communicated to other bees in the hive via "dance" movements (Esch, Zhang, Srinivasan and Tautz, 2001).

Path integration in honeybees, as well as other flying insects, is complicated by wind patterns and load carrying. Bee flight can be hindered or assisted by the direction of the wind, and bees are loaded with pollen on return flights to the hive. Therefore such possible path integration methods such as counting wing-beats, energy consumption, and integrating airspeed are not reliable.

By conducting experiments which eliminated these potential cues, Srinivasan, Zhang and Bidwell (1997) showed that honeybees can reliably measure short flight distances and they do so by analysing image motion. This visual odometry is primarily determined by motion in the lateral visual fields and information from the ventral field is only used in absence of other information. This use of lateral field image motion, allows the bee to acquire a reliable odometric signal independent of the height the bee is flying at. Esch et al. (2001) showed that it is distance based on this total image motion which is communicated by dancing bees rather than a measure of true distance. In this experiment honeybees found food after flying through a tunnel which induced exaggerated visual odometry. Upon returning to the hive the honeybees communicated the location of the food source to hive mates. The tunnel was removed thus reducing the image motion along the route to the food source. The hive mates now travelled exaggerated distances seeking the food source.

Learnt Routes

Such path integration, even using visual odometry, will accumulate excessive error over long distance navigation. Srinivasan et al. (1997) report that odometry is "reset" along the route when prominent landmarks are encountered. In fact Collett (1996) suggests that recognising landmarks along the way trigger the recall of associated goal vectors. Thus at each prominent landmark a new goal vector is obtained and the computation of distance is recommenced.

The construction of routes with segments triggered by visual stimuli is supported by a honeybee maze learning experiment by Zhang, Bartsch and Srinivasan (1996). In this experiment bees learnt to navigate mazes by following colour marks. Trained bees could then navigate through novel mazes by following the marks, and could even follow marks of different colours though with less success. In addition, bees could navigate learnt mazes when the colour marks had been removed, although again, were not as proficient as when the colour marks where present. This suggests that not only were the bees remembering colour cues but were also acquiring at least a set of motor commands defining the correct sequence of turns through the maze.

Zhang, Lehrer and Srinivasan (1999) go on to investigate the learning of multiple routes by honeybees. If a bee is foraging for food at more than one site, it needs to not only memorise a separate sequence of landmarks for each site, but must be able to retrieve from memory the set of landmarks appropriate for each route. They trained honeybees on two routes using two distinct sets of three visual landmarks. The results showed that honeybees can indeed store visual stimuli for more than one route at a time. In addition, when exposed to a landmark from a set of stimuli from a particular route, this triggered recall of the other two landmarks in the set. The associative recall of landmarks was largely independent of the sequence in which they were learnt.

That is they could follow a path marked by three landmarks independent of the order in which the landmarks were presented. Zhang et al. (1999) suggest this could be because of the training method, as other experiments have shown the sequence of stimuli to be important (Collett, Fry and Wehner, 1993).

Honeybees have been shown to exhibit multiple navigation strategies. They use specific flights of learning to learn about the visual scene surrounding important locations. Visual landmarks are chosen for specific navigation tasks. Honeybees also possess sophisticated visual odometry mechanisms which can perform reliable path integration in adverse weather conditions and when carrying loads. These two abilities are used in conjunction to build routes from the hive to feeding sites using conspicuous landmarks along the way to reset the visual odometer. Multiple routes can be remembered simultaneously and exposure to a landmark from one route triggers an associative recall of all other landmarks and related goal vectors along that particular route.

2.6.2 Rat Navigation

Rats can display similar levels of local navigation and way-finding as honeybees but also have more complex internal representations of space which allow them to perform more complex navigational behaviour.

Experiments have shown that rats are capable of all levels of the local navigation hierarchy. Rats have been shown to use path integration and to be able to follow pheremone trails. They can follow routes and exhibit the guidance strategy to navigate to goal locations.

Place Learning

Rats are also capable of learning specific places in relation to the general environment and are capable of navigating straight to those places along untravelled paths. Tolman (1948) reports an experiment where rats where trained to navigate the maze shown in Figure 2.6. The rats had to follow the path from A through locations B,C,D,E,F in order to reach food situated in location G. When training was complete, the maze was swapped with one shown in Figure 2.7. The rat was introduced to the maze at the

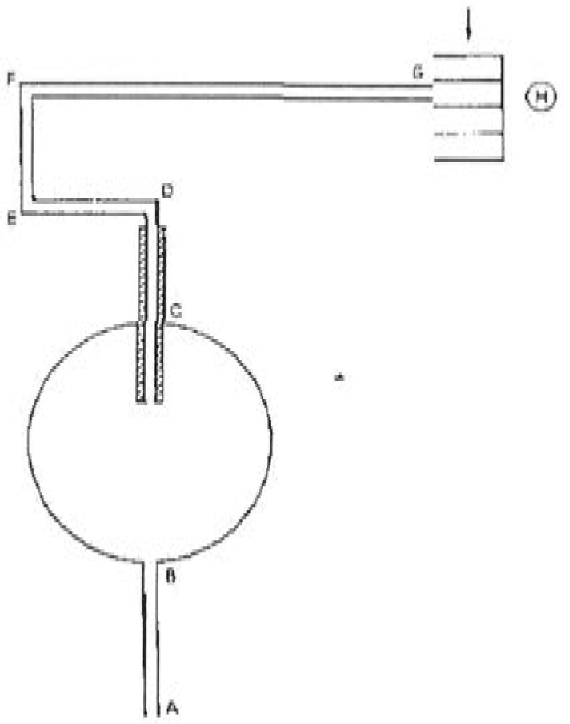


Figure 2.6: Maze used for learning in Tolman's (1948) experiment. Rats learnt to follow the path A,B,C,D,E,F to the goal G. Figure from O'Keefe and Nadel (1978).

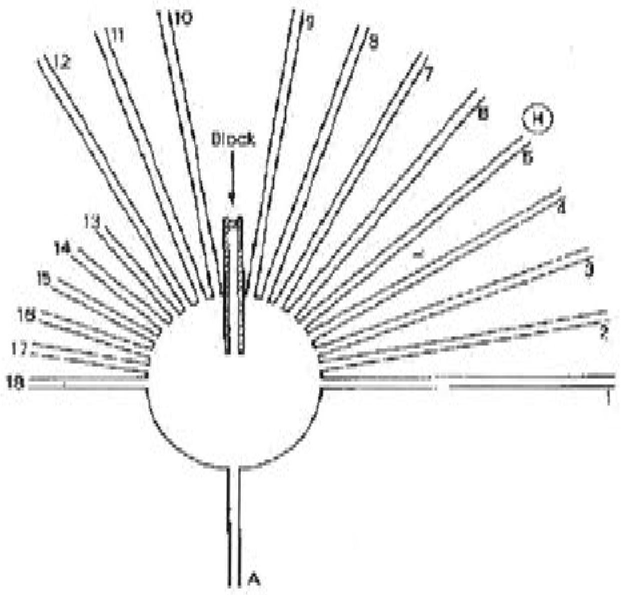


Figure 2.7: Maze used for testing in Tolman's (1948) experiment. The majority of rats choose the path which led straight to the goal located neat the light source at H. Figure from O'Keefe and Nadel (1978).

same start position A and the food reward was still positioned at the same location. Upon release the majority of the rats proceeded down the arm of the maze pointing directly at the goal. The rats were going to the place associated with the food reward rather than making a particular response to current visual stimuli. Tolman (1948) concluded this was evidence for an internal cognitive map, although there was a light source positioned at H which weakens this conclusion. This idea is developed further by (O'Keefe and Nadel, 1978) and they report further experiments supporting this hypothesis. O'Keefe and Burgess (1996) propose the hippocampus as the site in the brain where these cognitive maps reside. Rats which have lesions to the hippocampus are incapable of performing the above task, but can still successfully use landmarks to perform beacon homing. They call this *locale learning*, and describe navigation using this information as the cartographic strategy, which is equivalent to the survey navigation behaviour identified in Chapter 1.

Geometric Shape of Environment as a Cue

The above experiment does not prove that an explicit cartographic map of the environment is being made. The experimental setup does not eliminate reasoning about path-integration or visual cues leading the rats to the food reward location. Bennett (1996) state that these two alternatives must be eliminated before the cognitive map explanation becomes necessary.

Whether rats require explicit cognitive maps to exhibit this detour behaviour or not, they are performing a survey navigation strategy (see navigation hierarchy defined earlier) and require a more complex internal representation than bees do. Another situation which requires rats to use a survey navigation strategy is when they become disorientated (Cheng, 1986). Rats were trained to dig for food in their cage at a specific location. This location was partially specified by the cage geometry (ie a rectangular cage has ambiguous locations, and fully specified by the presence of distinctive odours and patterns placed about the cage. Rats that have been disoriented were placed in the cage and then must reorient themselves and dig for food accordingly. Surprising Cheng (1986) found that rats rely primarily on the geometric information from the cage walls to spatially reorient themselves, disregarding the more accurate odour and pattern cues.

This result is confirmed by Ramos (2000). In this experiment rats learnt to navigate a four arm maze with an open room that was inside a visually complex environment as shown in Figure 2.8. Rats released in any arm learnt to navigate to the particular arm in which a food reward was located. The use of pheremone trail following, olfactory detection of food or enacting a route sequence as possible navigation strategies was eliminated by experimental procedures such as rotating the maze between each trial and attaching inaccessible food rewards to each arm of the maze. Ramos (2000)

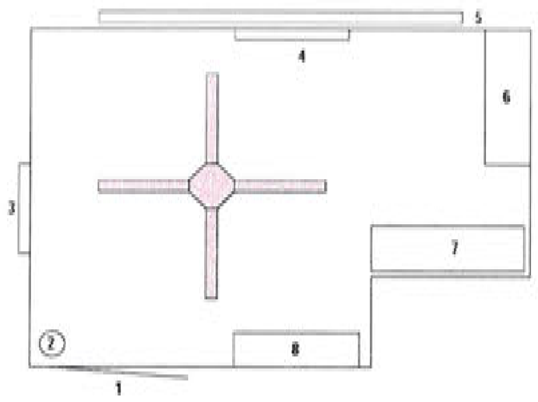


Figure 2.8: Experimental setup used by Ramos (2000) to test spatial reorientation cues in rats. The numbers label extra-maze visual stimuli.

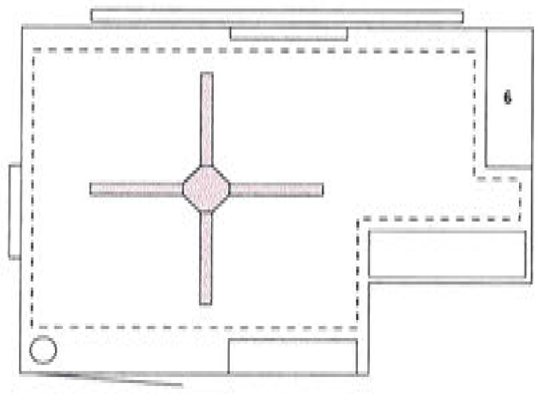


Figure 2.9: Room with visual stimuli covered by curtain which maintains the general geometry of the room (Ramos, 2000).

concludes that the rats were navigating using extra-maze visual stimuli as navigation cues. Continuing the experiment, the extra-maze landmarks were eliminated from the experimental environment. A white curtain was hung along the walls of the room, covering all distinct individual visual landmarks but still displaying the same room shape (Figure 2.9). Further testing showed that rats could still locate the food reward in the absence of all visual landmarks, the success rate for finding the food dropping

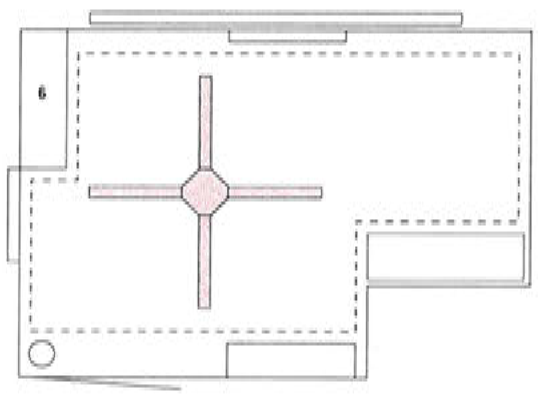


Figure 2.10: Room with visual stimuli covered by curtain which changes the general geometry of the room (Ramos, 2000).

from near 100 percent to around 80 percent. Only the deformation of the perceivable extra-maze environment (as shown in Figure 2.10) led to drastic reductions in the success rate, the figure dropping to around 40 percent.

This shows that the rats were not relying solely on visual landmarks in the extra-maze environment to reach the reward and led Ramos (2000) to state

> the neurologically intact animals in our study did not use a guidance strategy because their performance did not worsen significantly when the extramaze stimuli were eliminated.

Furthermore, Ramos (2000) provides the explanation that the rats build a cognitive map of the environment and the shape of the perceivable environment plays an important part in this construction. Again it is debatable whether this is in fact proof of a cognitive map as such, or the use of multiple cues to perform guidance. In this case using the geometric shape of the perceivable environment as a "visual landmark" which guides the rat to the correct location.

This distinction notwithstanding, Ramos (2000) concludes that:

> These findings favour the hypothesis that the geometric shape of the space that surrounds the animal is a behaviourally important component of the extramaze space influencing the orientation of the animals under conditions that elicit the cartographic strategy.

The fact that the degradation in performance due to covering the visual landmarks is minimal, and could be attributed to the slight change in the geometry of the room caused by the curtain, reinforces Cheng's (1986) conclusion that geometric information was the primary cue for spatial reorientation. Although in the absence of geometric information rats can still navigate using distinct landmarks, be they visual or olfactory.

2.6.3 Human Navigation

Humans can also spatially reorient themselves from geometric cues. Unlike rats, human adults can also use other distinctive cues to solve geometrically ambiguous situations. This is not surprising given the complexity of navigation tasks that we as humans perform everyday. What is surprising is that human children suffer from the same reorientation limitations that rats exhibit. Hermer-Vazquez, Moffet and Munkholm (2001) present a review of these findings and present experiments detailing the cognitive change in humans which can overcome these limitations. Inspired by Cheng's (1986) work on spatial reorientation in rats, Hermer-Vazquez et al. (2001) set out to investigate this process in humans. In particular they were interested in the emergence of human-specific cognitive traits through the evolutionary process of terminal addition. Hermer-Vazquez et al. (2001) suggest that terminal addition provides a logical way for studying these cognitive traits:

> find a trait for which young children show the phylogentically older and more common mechanism but for which human adults show distinctive flexibility, and then study the developmental change in depth.

In order to investigate the development of spatial reorientation the following experiment was carried out. Humans of varying ages were shown where a reward was located in the experimental room. This room was rectangular with red boxes in each corner and one of the short walls was painted bright blue while all other walls were white. The reward was hidden under a box which was ambiguously specified by the geometry of the room, but could be distinctly specified by the geometry in conjunction with the bright blue wall. Subjects were introduced to environment, the blue wall was specifically pointed out and the reward location was revealed. Subjects were then disoriented by spinning about with their eyes closed. When they were told to stop spinning and told to locate the reward, their search patterns were observed.

Human adults immediately search for the reward in the correct location. Young children below the ages of 5-7 years, split their searches between the two geometrically ambiguous locations, ignoring the additional cue of the blue wall. These findings suggest that children below this age cannot spatially reorient themselves using indirect landmarks such as the blue wall. Furthermore, the authors found a correlation between the ability to spatially reorient using indirect landmarks and the onset of specific language producing capabilities, namely the ability to specify verbally the exact information needed to solve the task. This correlation was also found in a movable object search task. Hermer-Vazquez et al. (2001) conclude that the development of human adult-like performance in these cognitive tasks is associated with the ability to form linguistic representations of the information required to perform the tasks in a more flexible manner.

2.7 Biomimetic Systems

By studying the navigation systems of biological organisms, solutions to mobile robot navigation have been proposed. This section looks at several of those solutions inspired by honeybees and rats and classifies them with relation to the navigation behaviour hierarchy. Of course robot navigation systems have been inspired by other organisms besides those discussed below and those systems exhibit behaviours covering almost all of the navigation hierarchy. For example, Lambrinos, Moller, Labhart, Pfeifer and Wehner (2000) have developed a navigation system based on the desert ant Cataglyphis which displays the navigation behaviours of systematic search, path integration and visual guidance. But for the most, biomimetic robots take their inspiration from honeybees and rats, in part due to the extensive research undertaken with these animals, and also because of their navigational prowess and their use of vision as the primary sensor.

2.7.1 Honeybee Inspired Robots

Mobile robots which have navigation systems inspired by honeybees are popular because of the relatively simple neural structure of honeybees, and the promise of computationally inexpensive algorithms that that simplicity implies. Researchers using bee inspired solutions are hoping to gain insights into the type of information required to do basic navigation tasks and also to test neuro-biological theories of bee behaviour.The section below details bee inspired systems according to their navigation competence.

Path Following

A robot system has been developed which can navigate along corridor environments using a centering behaviour inspired by bees (Coombs and Roberts, 1992). The centering behaviour is achieved by the robot balancing the optic flow in the periphery of each lateral visual field. When the robot is moving down the center of the corridor the optic flow is equal to each side, if it is closer to one wall, then the optic flow in that lateral field is greater than the opposing field so the robot steers to the center. The implementation of the centering approach is simplified by a single camera sensor facing forward with a wide field of view and an active gaze stabilising system.

This type of navigation is found only to be suitable in corridor environments. When operating in open areas the centering behaviour tended to make the robot move towards walls or into corners, as this type of motion also acts to balance the optic flow fields to each side. Therefore Coombs and Roberts (1992) suggest that an additional forward looking obstacle avoidance behaviour needs to be used in conjunction with the current approach.

Path Integration

Honeybees use visual odometry for path integration as well. Srinivasan, Chahl, Weber, Venkatesh, Nagle and Zhang (1999) describe a system which uses a centering behaviour, similar to the one above, in conjunction with visually mediated odometry. This system computes the distance travelled along a path by integration the image motion in the lateral fields over time. Of course the distance computed is not an actual distance measure, but a measure of the image motion experienced to travel a certain distance along a particular path. Provided the robot travels over the same path, missions of the same length can repeatedly be performed. If, for example, the distance between the two walls of the corridor were doubled, the robot would traverse twice as far, as the perceived image motion would be halved for a set distance.

Guidance

The use of visual landmarks to perform guidance navigation is popular in biomimetic literature. Bianco and Zelinsky (1999) stands out by mimicking the Turn Back and Look flights of bees and wasps to evaluate landmarks. In order to learn a place Bianco and Zelinsky's (1999) robot first selects potential landmarks from the visual scene at the goal location by an adapted interest operator (Moravec, 1977; Mori, Matsumoto, Shibata, Inaba and Inoue, 1995). Then the robot makes a series of short movements in ever increasing arcs while facing back at the goal location and tracking the potential landmarks. Landmarks which track well over this movement are considered *reliable*. The area from which these landmarks can be recognised form a navigation field, inside which the robot can successfully be guided back to the goal position. Guidance is achieved by moving so as to reduce the error in image space between the landmarks' current and reference positions.

This approach could perform homing from within the navigation field, which was a pie slice shape with the vertex at the goal and approximately 30 degrees wide and up to 10 meters deep. If the robot was located outside this field, then it could not find its way to the goal location.

Recognition-Triggered Response

Although not directly mentioning the inspiration of bees in their motivation, Gaussier, Joulain, Zrehen, Banquet and Revel (1997) propose a biologically inspired system capable of the recognition-triggered response behaviour evident in bees. From a central goal position, the robot moved around the local environment and captured 4 panoramic image views at four places surrounding the goal. Features are extracted from these views by collapsing intensity values for each column of the panoramic images into a 1 dimensional intensity histogram. The derivative of this signal and its local maxima and minima are used to define the places.

Gaussier et al. (1997) then use PerAc, a neuro-computation architecture, to perform recognition-triggered responses to navigate towards the goal. PerAc consists of:

> an action level (a hardwired pathway able to play the role of a reflex mechanism) and a perception level trying to recognise particular situations and to associate them with the correct action through an associate or a reinforcement learning rule (Gaussier et al., 1997).

Given a current panoramic view of the environment, the robot when trying to home to the goal location, tries to recognise the distinct place closest to the current view. If a place view recognition occurs, then this triggers the action of moving to the goal in the direction associated with the recognised place. With just the four different views and associated movements, the robot can successfully navigate to the goal location from anywhere in the surrounding open room environment.

2.7.2 Rat Inspired Robots

In addition to robot navigation systems mimicking the navigation behaviours of honeybees, there has also been research directed at producing systems inspired by rat behaviour. Although there is no longer the close correlation between observed behaviours and implementable strategies that is evident in bee inspired systems, rat navigation studies have provided many insights for robot navigation behaviour. The main reason for this is the large body of evidence in rats for the hippocampus being the site of so called "place cells", specific cells in rats, and other animals, which fire when the rat encounters specific places in the environment (O'Keefe and Nadel, 1978)(Eichenbaum, Stewart and Morris, 1990)(Oliveira, Bueno, Pomarico and Gugliano, 1997). This has led to the proposal of many topological map based systems in biomimetic literature. Building rat inspired robot navigation systems also allows neuro-physiologists to test hippocampal models of spatial learning and navigation behaviour.

Modelling Hippocampal Function

A simple model of hippocampal function with regard to place cells was proposed by Burgess, Reece and O'Keefe (1994). In this model spatial reorientation is defined by the firing rates of several hippocampal place cells. These place cells correspond to observed distances to walls in the environment. So a particular firing pattern over these places cells would define an open region of space in the rats field of view (approximately 220 degrees). This model is supported by rats spatial reorientation limitations described by Cheng (1986).

This model was then tested on a mobile robot platform (Burgess, Donnett, Jeffrey and O'keefe, 1997). The robot in this study learnt places by first visually estimating the distance to surrounding walls. Then it used a competitive learning mechanism to select which place cells should be used to represent the observed distances. Using this representation of places and path integration, the robot could localise and navigate between places.

Topological Maps

The more common approach to rat inspired robotic navigation systems is to use the idea of places cells in the hippocampus as sites for places in topological maps. An example of such a system is that of Bachelder and Waxman (1995) who used the configuration of observed landmarks to define places in an open room. These places were learnt by a neural architecture in which place cells differentiated between sensor views gathered from different locations in the environment. Another level of the system learnt the connections between place cells and associated them with movements. Although this system showed that indeed such places could be learnt and recognised, the environment in which the experiment was carried was quite artificial and it was not tested on a real robot.

Mallot, Bulthoff, Georg, Scholkopf and Yasuhara (1995) use the idea of topological maps comprised of local views from places in the robot environment. The view graph is connected by vertices representing movement similar to Bachelder and Waxman (1995). The robot only operated in a maze environment and all junctions had there own distinctive markings. The robot could, after learning a maze consisting of 12 places and 24 views, navigate between any pair of views by the shortest possible path.

This approach was tested in more realistic environments by Franz et al. (1998). In this experiment a small robot navigated around a miniature town using a panoramic vision sensor. Views consisted of a one dimensional intensity vector from the panoramic image taken at a particular place. View nodes were connected by adjacency alone and did not hold any distance information. A place was only added to the view graph when the current view was sufficiently different to all other previously stored views in the view graph. This meant that no two nodes in the view graph were similar enough to cause recognition ambiguity. Although it did limit the areas of the environment in which the robot would learn places, therefore limiting subsequent navigation tasks.

2.7.3 Combining Navigation Strategies

From the biomimetic mobile robot navigation systems described above, it is obvious that to perform any useful behaviour it is necessary to combine navigation strategies.

Experiments with visually mediated odometry combined path following with path integration. All way-finding strategies required a local navigation behaviour in addition to the recognising of places.

Gaspar, Winters and Santos-Victor (2000) present a navigation system which explicitly makes use of this combination of simple strategies and notes the emergence of a powerful navigation system. In their system panoramic visual images are used to create what the authors call a topological map, but is more like a learnt route with places spaced 50cm apart and each place being represented with a panoramic image. The images' eigenspace are used to reduce matching in localisation, but the search is restricted to local position estimates as this method is sensitive to perceptual aliasing. In the route, transitions are specified by one of two navigation behaviours: corridor following or visual path following. Corridor following is achieved by picking out line segments where the wall meets the floor. This process is simplified by converting the warped panoramic image into a birds eye view, in which these lines are straightened. Visual path following is realised by tracking features in the birds eye view images. Features are corner points defined by the intersection of long edge segments. Corners provide more accurate information, while long edge segments track more reliably. Recognition of a particular place along the route which requires visual path following, triggers the recall of a set of features and a desired trajectory. In this way the robot can move accurately in special situations such as navigating through doorways or through cluttered environments. At the moment the user must initialise the features.

This system illustrates the importance of multiple navigation strategies in mobile robots. Although in this experiment strategies were switched in specific situations, so only one had exclusive control at a particular time, in general simultaneous activation of strategies is desirable.

2.8 Summary

The general theme throughout this literature review has been that navigation systems depend on an interplay between their internal representations and their navigation strategies. In traditional robotic systems, there has been a predominance of fine grained representations and high level navigation strategies. Biomimetic approaches try to use parsimonious representations of the environment to execute simple navigation strategies.

Traditional approaches grapple with the problem of computational tractability when localising in large scale maps. Current efforts are focused on reducing the computational expense of the matching phase in localisation in order to produce efficient global localisation. This can be done by finding efficient representations of spatial knowledge or by intelligently directing the global localisation search. Biomimetic approaches have implemented some simple navigation strategies, but now face the problem of imitating more complex navigation behaviours that are exhibited by biological organisms. True survey navigation has not been demonstrated in biomimetic systems, and the problem of embedding places in a topological map into a common frame of reference has not been attempted. Although it has not been proved that this level of cognitive map is present in biological systems either.

In both approaches there appears to be an emergence of systems which attempt to integrate navigation behaviours into more robust and successful solutions. In traditional robotics this is seen by the integration of metric and topological maps and the introduction of hierarchical maps, while in biomimetics, the focus is on combining navigation strategies. It is interesting to note that traditional approaches seem to concentrate on integrating the internal representations, while biomimetic approaches try to integrate navigation strategies. By looking at the limitations of systems representing each side it is clear that both spatial representations and navigation strategies need to be integrated together to form successful navigation systems.

The evolution of these systems is converging on the original goal of mobile robot navigation: that of robots which have navigation abilities equivalent to those of humans. From evolution of the human navigation system, important insights into the integration of different levels of navigation strategies and internal representations can be gained. In particular the study of the cognitive abilities related to navigation tasks which develop under the evolutionary process of terminal addition can be used as a model from which integrated navigation systems can be designed. The progression of navigation ability from that exhibited by bees, to rats and finally humans is of particular note. The studies reported in Section 2.6 show spatial knowledge cues starting at landmarks for bees, progressing to a sense of open space at the expense of landmarks in rats, and then the combination of this sense of space with landmarks in humans. The latter being linked to the onset of linguistic reasoning capability. Although these results can provide a starting point for robot navigation system design, a deeper understanding of the cognitive processes involved with navigation in mammals and humans need to be formed before robotic navigation systems with equivalent behaviour can be developed.

The goal of this research is to implement a navigation system for a mobile robot that can robustly and reliably navigate around an office environment using a learnt map. Navigation in this case, in the absence of specific goals, means point-to-point navigation within the learnt map, allowing for accurate positioning within 5-10cm of goal locations. Also the robot should be able to perform equally accurate localisation at any time should the situation arise, ie a dynamic obstacle moves close to the robot. The robot should also be able to recover from any localisation errors or failures such as the kidnapped robot problem.

The selected environment means that there are still variations within it that require different navigation strategies (offices, corridors and open rooms). In particular maps must be created that have the necessary information to localise within these different environments. Also the environment is of a sufficient size that intelligent search strategies in the localisation process must be utilised.

The thesis will concentrate on the mapping and localisation components of navigation, in particular developing internal representations, or spatial memories, that limit the computational cost and enhance the reliability of these components.

Chapter 3

Multi-Level Spatial Representation

This thesis proposes a multi-level spatial memory for mobile robot localisation. It is directly inspired by the three levels of navigation behaviour shown by honeybees, rats and humans: recognition triggered responses and visual homing in honeybees, spatial orientation and navigation using the a sense of space in rats, and the ability to pair this sense of space with disambiguating landmarks in humans. The increase in knowledge required in these navigation tasks directly relates to the amount of distinct visual information extracted from the environment and thus influences the organism's performance of the global localisation task. The level of abstraction needed in each level of navigational ability is inversely proportional to the accuracy of position estimation. Therefore a successful solution must have low level navigation strategies to navigate accurately and also must have higher levels to perform global localisation in large, visually ambiguous environments. These navigation behaviours need three distinct levels of internal spatial knowledge:

1. precise and distinct features in the local environment with which to perform position estimation;
2. general features in local environment which may be similar between some places but are easily extracted from sensory views;
3. specific areas of sensory scene which disambiguate between similar places.

Section 3.1 further motivates the proposed multi-level spatial memory solution to the mapping and localisation problem and introduces some issues involved in such an

Spatial Knowledge	*Representation*	*Biological Inspiration*	*Advantages*
Precise distinct features	Unique visual landmarks	Honeybees	Accurate local position estimate
General, easy to extract features	Extent of local space	Rats	Guide global localisation search
Disambiguating features	Distinct image regions	Humans	Eliminate spatial aliasing

Table 3.1: The three levels of spatial representation

approach. Section3.2 provides details of the chosen sensor, a panoramic camera. Initial details of each level of spatial memory proposed in this research are described in Sections 3.3, 3.4 and 3.5 . A solution for the integration of multiple levels of representation is introduced in Section 3.6. Integration of the levels of representation is needed for the robot to form an internal belief as to its location and also to help restrict the global localisation search space. Section 3.7 provides a brief introduction of the robot hardware, software and sensor platform that is used in this research to validate the proposed multi-level representation for spatial memory.

Section 3.8 provides a summary of the proposed representation and restates the primary objectives of this research with regards to the multi-level representation This section and indeed the chapter concludes by listing the key contributions of our research to the field of mobile robot localisation.

3.1 Multi-Level Spatial Representation for Mobile Robot Localisation

By building topological maps which incorporate these three spatial representations into the idea of places, the following can be achieved:

1. accurate position tracking in local navigation;
2. the global localisation search can be reduced by targeting matching towards similar areas;
3. perceptual aliasing can be eliminated by directly choosing disambiguating features from conflicting places on a global level.

Perceptual aliasing is the problem of falsely categorising a percept due to ambiguous or incomplete perceptual input. Applied to the problem of mobile robot localisation this means the possibility of mis-localising due to the similar appearance of samples of sensor data captured from different locations in the robot's environment. In mobile robot literature this is also referred to as the *data association* problem.

A good choice of internal representation at each level would lead to cues from individual levels contributing to the goals of other levels. For example a good choice of visual landmarks for maintaining accurate local positioning could also lead to good place level discrimination, while a mid level feature which captures some geometric information could contribute to accurate positioning.

The specific representations chosen for this system are unique visual landmarks for localising between places and accurate positioning, a sense of local space for directing the global search, and distinct regions of images for ultimate disambiguation between places. These representations, their inspirations and their contributions to the mobile robot problem is given in Table 3.1.

A system which is comprised of multiple levels of representations needs some way of combining the information from each level. In navigation systems such as this, this means that localisation as a whole must be distributed throughout the many levels, and must take into account information provided at all levels, while maintaining a central internal belief as to the robots current position. Approximations of probability densities have been used to maintain this belief in robotic systems and they can be applied to hierarchical problems (Jensfelt, 2001). This system will use a particle filter with samples distributed according to the information available from the levels of spatial knowledge.

3.2 Sensor

Sensors are used to perceive the environment. In general the more information a robot has about the environment, the better decisions and actions it can take to accomplish a goal or task. Decisions and actions, however, must be made in a timely fashion, therefore the amount of incoming sensor information and the degree to which it can be processed is limited by the computational power available.

Chapter 2 introduced the concept that the choice of sensor in a robot localisation system directly impacts the choice of internal abstraction with which the robot system represents the external environment. In order to form a representation from which it is possible to accurately estimate the local position of a robot, the sensor must provide sufficient positional information from the environment. Range sensors such as laser range finders provide accurate positional information (Dellaert, Fox, Burgard and Thrun, 1999) (Jensfelt et al., 2000). Vision sensors have not typically been used to provide accurate positional information for the mobile robot localisation task, although the information is present in the visual scene (Davidson and Murray, 2002).

Representations that allow a unique description of places in the environment require a sensor that provides a rich source of information from the environment, from which unique properties of specific places can be extracted. In typical environments, range data and thus range sensors, cannot provide enough information from the environment to form such unique representations. Vision as a sensor captures a wide variety of information from the environment and therefore can be used as a source from which unique representations can be extracted.

To perform the competing tasks of local position estimation and global localisation, a sensor must be chosen which provides both accurate position information and the level of detail necessary to extract unique representations of locations in the environment.

The choice of sensor is also related to the desired behaviour of the agent. A sensor must perceive enough of the environment to accomplish a given task, using the available computational power. As mentioned above, the choice of sensor also impacts on the map representation in a navigation system. Because the choice of sensor is crucial to the success of the overall system, it makes sense to find inspiration in successful navigation systems such as biological systems. Biological systems have developed complex navigation systems while being tightly constrained by evolution on the 'choice' of sensors and processing power.

Biological Sensors

Almost all evolutionary pathways led to biological systems using vision as the dominant sense involved in navigation. Ignoring the debate on which sense is easier to evolve, this raises the question why is vision as a sensor for navigation systems so common? The simple answer is vision sensors rely on a physical medium (light) which can represent very complex, dynamic, high resolution data about the environment with relatively low cost (compare this to the sense of smell, which relies on airborne molecules which simply have a distinct odour). Accepting that vision is the best sense for navigation, only brings up further questions, namely:

- should vision be used in isolation?
- what information needs to be extracted?
- what vision sensor characteristics should be used?

These questions need to be evaluated in the light of available computational resources and desired behaviours. A good way to do this is to compare biological organisms with simple vision systems to those with more complex vision systems. Insects have a relatively simple neural structure and limited processing capacity yet can still exhibit complex navigational behaviours. They do this by mainly relying on visual input from large (compared to overall body size), fixed position eyes with a large field of view, and very limited stereopsis. Biological organisms with more complex nervous systems (such as mammals) on the other hand, use vision sensors that have different combinations of high resolution, limited field of view, foveated visual attention, stereopsis, and are integrated with other senses to direct visual attention. Eyes with limited fields of view are usually combined with bodies with flexible necks and appropriate behaviour patterns to search the entire environment.

From this it can be reasoned that computationally simple navigation can be achieved using simple sensors if they cover a large field of view. This navigation is limited however, by the resolution of the sensors and the lack of integration from other sensors and internal knowledge. From another point of view, the use of visual sensors with a large field of view can bypass the need for complex computation involving multimodal sensing and attention direction, given the goal of basic navigation tasks. For more complex navigation, requiring more indepth knowledge about the environment such as 3D object reconstruction and semantic classification, more complex sensors, and integration between sensors is needed.

Panoramic Vision

The implementation of successful robot navigation systems, is limited by the current computation paradigm and the associated computational power. Vision systems which can reconstruct 3D objects in real time are starting to appear but the underlying semantic classification and reasoning to use these objects in intelligent navigation is still not available. Likewise, sensor integration for basic sensing is achievable but the integration between sensors, actuators and a semantic reasoning level is still a long way off. Without an underlying system which is capable of integrating multi-modal senses and internal knowledge, complex reasoning and planning behaviours, and complex navigation tasks on the level that humans are able to display are unachievable.

Therefore it makes sense to chose sensors which fit in with the current level of artificial reasoning capability and computing power, and use sensing and navigation strategies from biological organisms which posses a nervous system with a comparable limit to real-time reasoning and computation.

The sensor used should also be able to capture all information necessary to form the required levels of spatial representation. One sensor which provides all this information without requiring extensive sensor fusion or active attention direction is the panoramic vision sensor. This sensor provides a 360 degree visual image of the environment using one camera. This is achieved by pointing a normal video camera at a conical mirror as shown in Figure 3.1. The raw sensor data is then a view of the entire environment but in a polar coordinate form about the axis of the camera. Vertical lines in the environment are converted into radial lines in the *warped* panoramic image, but horizontal lines become distorted and are hard to recognise. In this warped view, it is hard to track visual features as even rigid motion causes non-linear deformations in the image space. Point features, such as those identified by the Kanade-Lucas-Tomasi (KLT) tracker (Lucas and Kanade, 1981) can be tracked successfully in a warped panoramic image stream (Strelow, Mishler, Singh and Herman, 2001), but

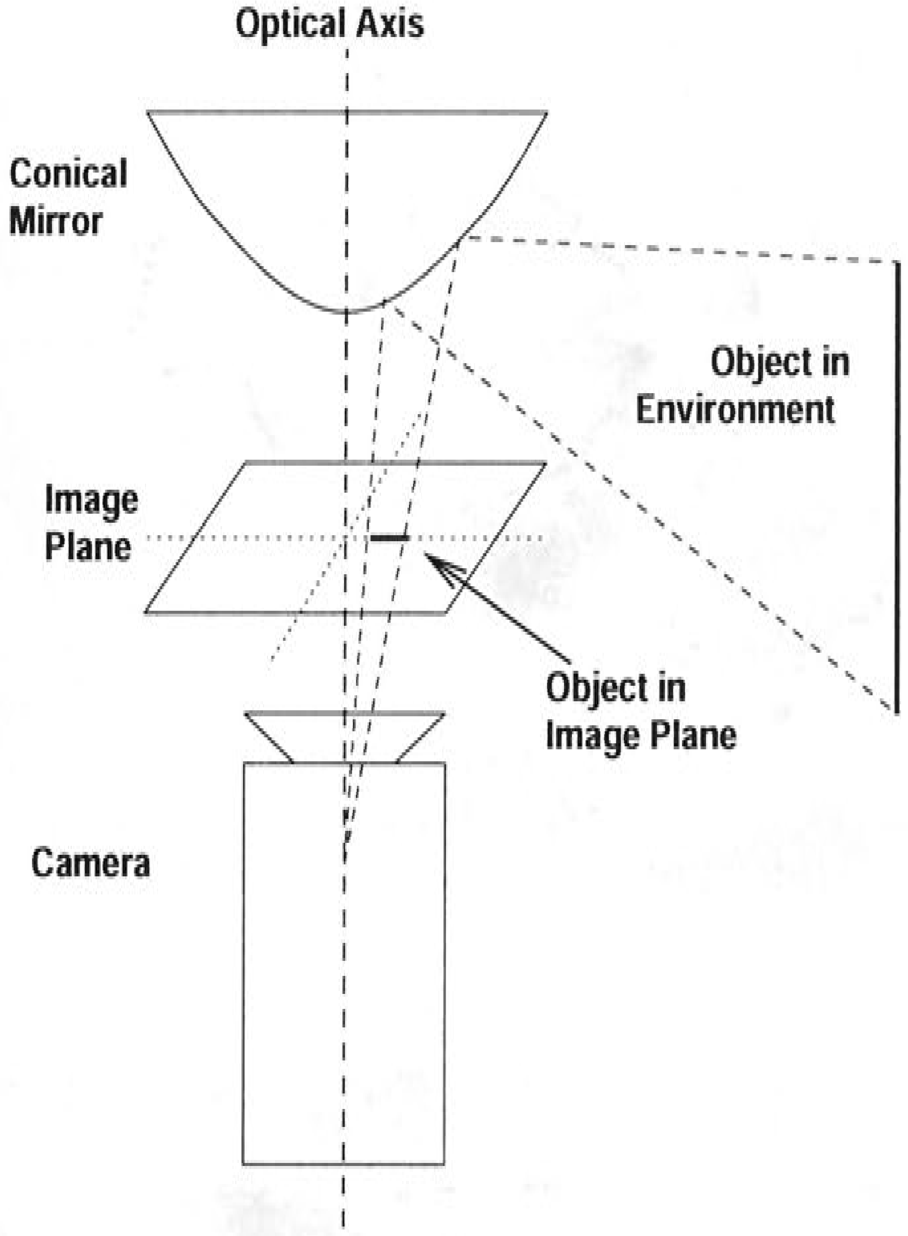


Figure 3.1: Panoramic sensor configuration.

such features are not unique in the image. The lack of uniqueness in the representation of KLT features means that individual features cannot be uniquely identified and recognised on revisiting a location in the environment. This failure could lead to errors in global localisation and possible data association problems. Appearance based visual landmarks have a unique representation, and although tracking in a warped panoramic image stream is possible using deformable templates, recognition of such landmarks when revisiting learnt places in the environment is difficult to achieve.

An example warped image is shown in Figure 3.2. This image can be *dewarped* in software to form a more recognisable image, and one in which visual features can be tracked more reliably. An example dewarped panoramic image is shown in Figure 3.3. Notice that the resolution in panoramic images is greatly reduced and is not even throughout the entire image. Additional information on the panoramic sensor used in this research is presented in Section 3.7.

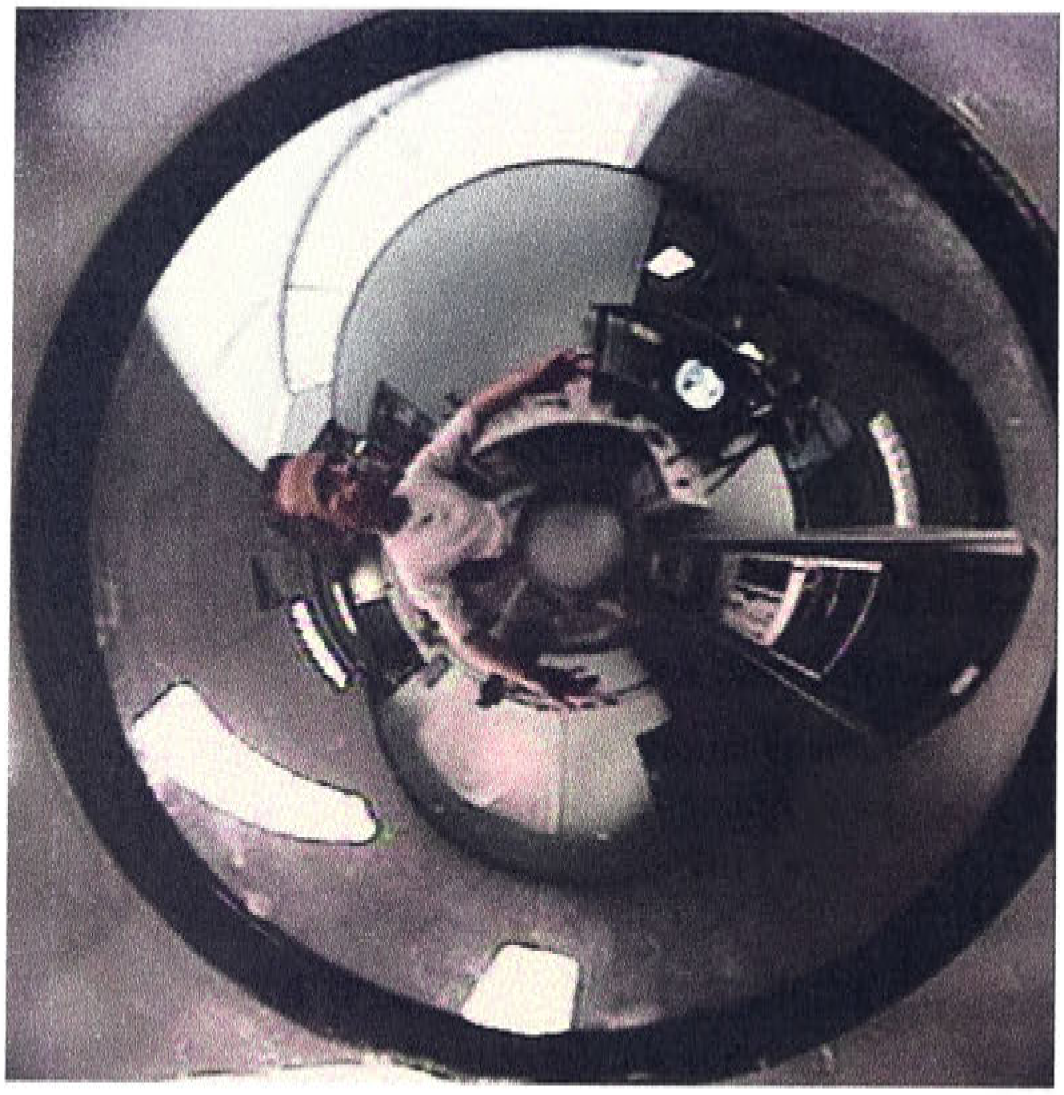


Figure 3.2: An example of a warped panoramic image, captured by camera.

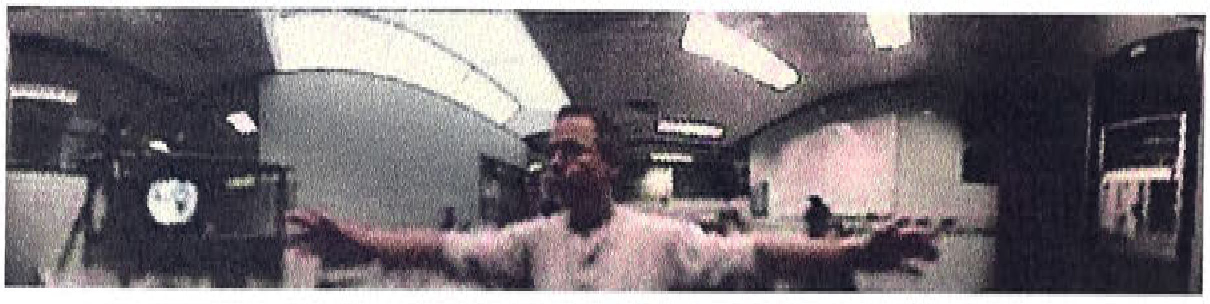


Figure 3.3: An example of a panoramic image, dewarped by software.

The present system uses these dewarped images as the main sensory data for navigation. In addition to this visual sensor, the navigation system receives odometric information from the encoders in the wheel motors of the robot platform.

3.3 Low Level: Unique Visual Landmarks

Unique visual landmarks will be used for accurate local positioning relative to a learnt place. A set of such landmarks will also be used as the low level representation for individual places in a topological map. The selection of landmarks for place represen-

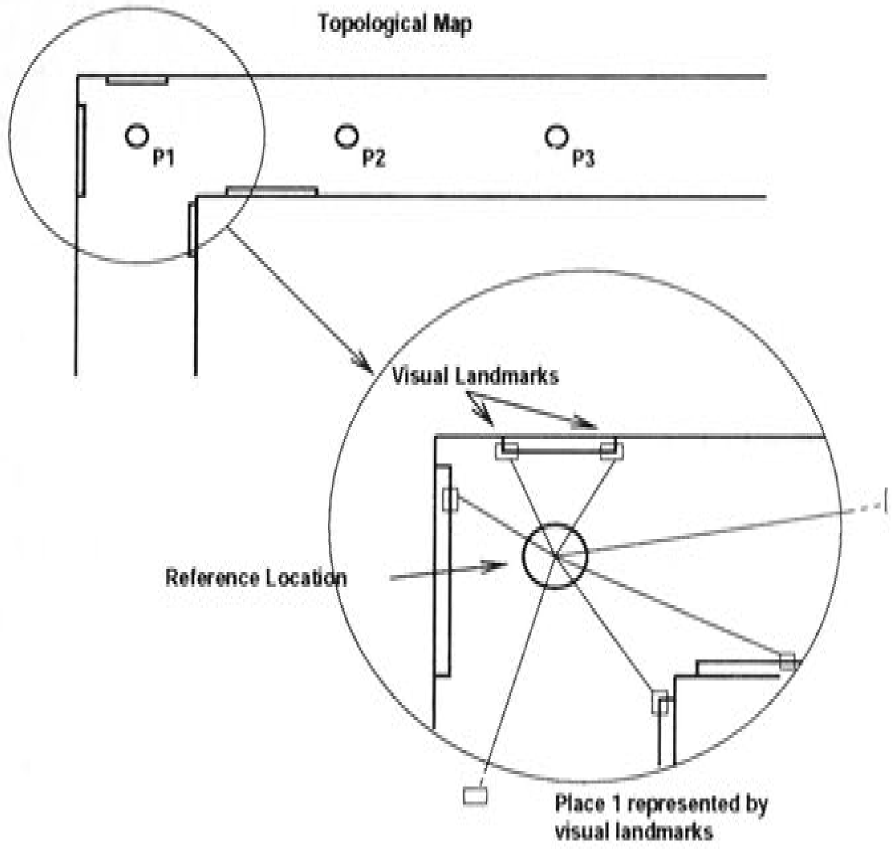


Figure 3.4: A set of visual landmarks representing a place in a topological map.

tation is crucial for the performance of the system. By choosing unique landmarks, the representation of individual places will be more distinct and localisation between places will be easier. But by using unique landmarks instead of features, the cost of matching increases as mentioned in Chapter 2, and therefore strategies to reduce the amount of matching are necessary.

The selection of landmarks also determines the coarseness of the topological map. The larger the area in the environment represented by a place, the coarser (and subsequently less complex) the topological map can be. But this coarseness might come at the cost of local position accuracy. In general visual landmarks representing objects close to the robot location provide accurate local positioning information but can only be recognised in a small area of the environment. Landmarks which are from objects located far from the robot can be recognised over a greater area but do not allow for accurate position information.

3.3.1 Landmarks in Panoramic Images

Panoramic images allow for landmarks to be selected from all directions surrounding the robot. This large field of view increases the possibility of selecting useful landmarks and also adds to the reliability and accuracy of local positioning. By ensuring

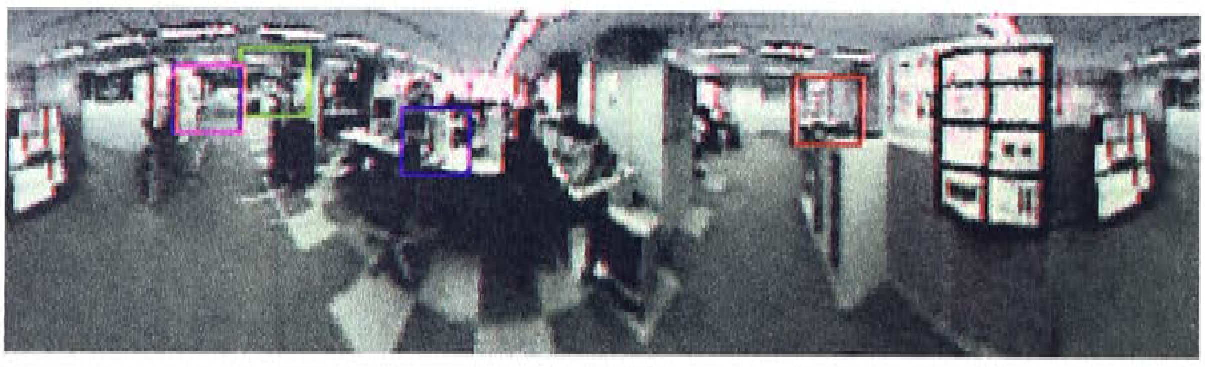


Figure 3.5: An example of unique visual landmarks in a panoramic image.

the set of landmarks representing a place are distributed throughout the panoramic image, the robot can minimise triangulation error and can also handle occlusion of landmarks due to dynamic objects moving through the visual field. For ease of matching, landmarks in this system are small, square image regions. Figure 3.4 illustrates the use of a set of landmarks in representing a place in a topological map. An example of landmarks in a panoramic image can be seen in Figure 3.5.

3.3.2 Automatic Landmark Selection

Landmarks must be acquired automatically, and should be selected according to their ability to contribute to place recognition and local positioning. This system primarily focuses on selecting landmarks which allow for reliable place recognition as the use of the panoramic sensor and its triangulation ability should provide for reasonably accurate local position estimation. When learning a place, a panoramic image is captured from the reference position and the automatic landmark process begins. The selection process involves first a static phase where potential landmarks are extracted based on their local uniqueness, and a second, dynamic phase, where there recognition reliability is evaluated over a series of movements about the reference position. This dynamic evaluation is directly inspired by the Turn Back and Look behaviour in bees and wasps, and first applied to biomimetic robots by (Bianco and Zelinsky, 1999).

3.3.3 Landmark Matching and Localisation

In a system using visual landmarks, localisation becomes a process of locating the visual landmarks of a particular place in the current sensory view. Global localisation in a topological map is the process of finding the set of landmarks associated with a particular place which have the best match with regions of the current image. Local positioning involves using the position of matched landmarks in the current image to estimate the robot pose relative to the associated place's reference position. Because landmarks are unique visual patterns, each landmark must be matched against the entire image in order to find the best match. Given that landmarks are initially selected on the basis of a measure of local uniqueness, it would be interesting to experiment with matching reference landmarks with areas of high local uniqueness in the current image, similar to the work of (Sim and Dudek, 1999), and evaluating the cost/benefit of such a approach. Particle filters specific to individual places can integrate the landmarks observations over time to provide more robust position estimation.

This system uses a set of unique visual landmarks automatically selected from panoramic images of the environment to represent places in the topological map. Automatic visual landmark selection allows the selection of locally unique features in the environment which leads to a more unique place representation than finding generic features such as doors, walls etc. The use of a panoramic image sensor, as well as the landmark selection process allows for a greater coverage of the environment for each learnt place. The combination of panoramic sensing and unique landmarks also allows for more accurate local positioning. The details of the implementation of this level of spatial representation can be seen in Chapter 4, and the associated localisation process in Chapter 8.

3.4 Mid Level: Local Space

Knowledge of the extent of local space can be used to restrict the global localisation search. Places in the topological map can be eliminated from the matching process if the extent of open space in the robots local environment is not similar to their own. Local space is a useful cue in this process as it is necessary for obstacle avoidance and motion planning and therefore must be calculated anyway. In addition, once global localisation has been achieved, knowledge of local space can be used to provide additional local position information if required. It also has the opportunity to provide extra metric information if more complex spatial representations are to be formed.

Matsumoto, Inaba and Inoue (2000) report a mobile robot system with a panoramic sensor which uses coarse depth information to guide exploration. The panoramic sensor produces lateral depth estimates based on optic flow while a compact stereo vision system produces sparse disparity maps of the frontal view. These depth estimates are used to annotate a View-Sequenced Map for further exploration. This information, although pertinent, is not used in the localisation process.

An alternative mid level representation providing a constraint on the global localisation search space would be the colour band histrogram representation reported by Ulrich and Nourbakhsh (2000) and described breifly in Chapter 2. This approach provided efficient and effective place recognition using histograms of colour composition in panoramic images. Restricting the global localisation search space using this representation could not, however, provide additional local position information. Unlike knowledge of the extent of local space surrounding a robot, this representation is not required for other navigation tasks such as obstacle avoidance, therefore would mean additional computational expense.

In this system, when learning a place, a sense of local space is extracted from the current sensory view. This local space is matched with a set of local space primitives and an association is formed with the most similar local primitive. When localising the current view of local space is extracted from the sensory data and is compared with the set of primitives and the subsequent landmark matching process is limited to places with a similar local space primitive. This notion of local space primitives is inspired by the evidence of place cells in the rat's hippocampus being associated with measurements of open space (Burgess et al., 1994).

3.4.1 Determining Local Space

The extent of open space around a mobile robot provides information crucial for motion planning involving obstacle avoidance. Since this information is being extracted from sensory data for other operation critical robot behaviours, it makes sense to take advantage of this spatial information when performing localisation.

The use of panoramic vision as the primary sensor simplifies the task of determining local space immensely. Because the vertical axis in the dewarped panoramic image

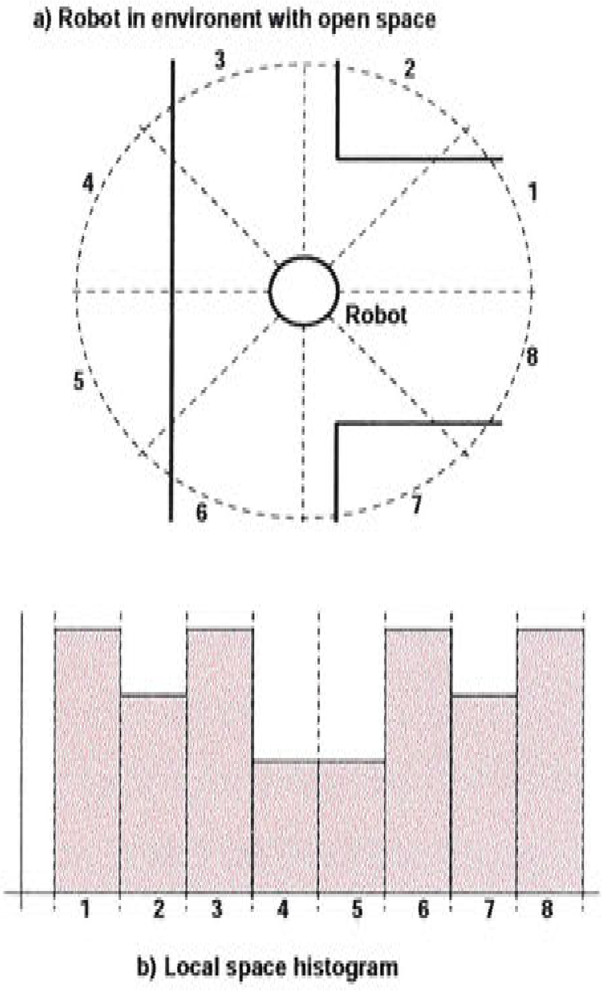


Figure 3.6: Local Space around a robot can be represented as a histogram.

corresponds to radial directions about the robot platform a sense of space along these radial directions can easily be evaluated by finding a free space boundary throughout the panoramic image. This is equivalent to finding the ground plane in the panoramic image, an easy approximation of which can be obtained by using colour cues. This can be done in a variety of methods but for ease of implementation, the current system will use carpet colour matching, the details of which are given in Chapter 5.

The resolution that a view of open space requires depends on what we are using it for. For the current task of limiting the localisation search by matching with local space profiles, the resolution can be relatively low. If in addition to matching with the local space profiles, the open space information was being used to provide local position information then the required resolution would be greater. In this system, for speed of matching between two local space profiles, the resolution of the local space measurement has been limited to sixteen discrete directions about the robot platform, which can be represented as a histogram. Figure 3.6 shows an example of a low (eight directions) resolution view of open local space about a robot platform, and its associated local space histogram.

3.4.2 Local Space Profiles of Places

Rather than representing each place with a unique local space histogram, it is possible to instead associate them with a local space primitive. This means that a quick search can be performed against the limited set of local space primitives rather than matching a current local space profile against each place in the topological maps local space profile. Then computational resources in the global localisation search can be targeted to those places which are associated with currently activated local space primitives. An example of local space primitives being associated with places in a topological map is given in Figure 3.7. In this example the environment is simple and only two local space primitives are needed. For more general environments more primitives will be required but their number will not grow in proportion to the size of the map. Environments incorporating large regions of open space can be categorised by a local space profile with maximum depth measures in all directions. The use of such a set of primitives depends on their ability to successfully categorise the local space profiles of places in the topological map. If the set of primitives can successfully group similar places together then they can make a useful contribution to the localisation task. If the set of primitives do no contribute significantly to the localisation task, then it would be better to represent each place with a unique local space profile. Experiments evaluating the contribution of local space primitives are presented in Chapter 5.

In the current system, a sense of open local space is used to constrain the global localisation search. The search is restricted to places which have a similar local space profile as the current sensory view. The use of panoramic images simplifies the detection of local space. In the future, higher resolution local space profiles could be used to aid local position estimation.

Figure 3.7 shows places in a topological map with a local space profile representation. It also shows how a set of primitives could be used to classify local space profiles to limit the global localisation search. When performing localisation, the local space profile of the current panoramic view could be matched against the local space histograms for each place or the set of local space primitives. The results of the matching process could be used to restrict the global localisation search.

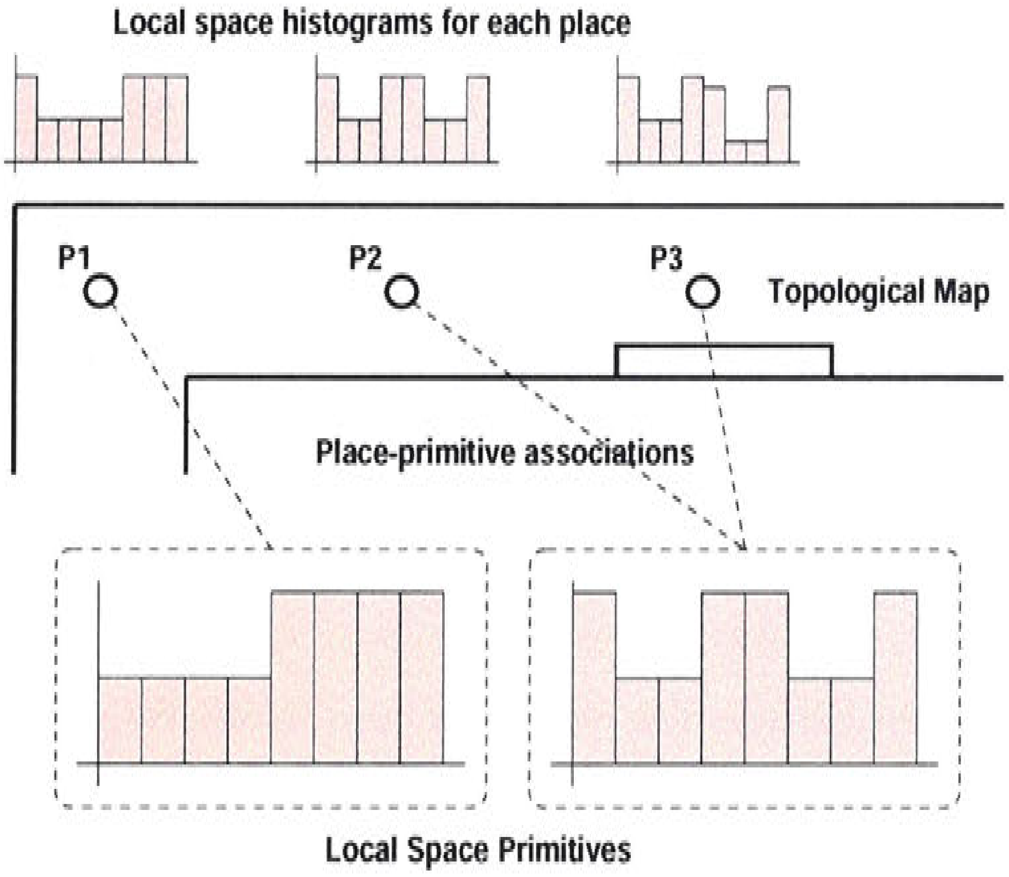

Local space histograms for each place
P1
P2
P3
Topological Map
Place-primitive associations
Local Space Primitives

Figure 3.7: Mid level place representation: associating places with local space primitives.

3.5 High Level: Disambiguating Features

In problematic cases the situation can arise in which neither the defining set of visual landmarks nor the shape of the local space can differentiate one place from another. A solution to this would be to define another type of feature with which to classify places. But what happens when this feature again is in-discriminable between places? At some point the addition of feature classifications will start to provide diminishing returns. In fact the combination of unique visual places with a sense of local space will be enough to globally localise in most cases. When this isn't the case and a decision between two or more places is needed urgently, a direct comparison of both places can be made, and specific differences between places can be used to make the localisation decision. This is inspired by the emergence in humans of the use of indirect landmarks to spatially reorient themselves. The appearance of this ability is correlated with that of the ability to produce language containing the exact information needed to solve the task. In this system, a search of the visual scene at similar places is made and regions of the image are identified which contain the exact information necessary to solve the task. This requires that complete visual scenes of each place are stored when they are

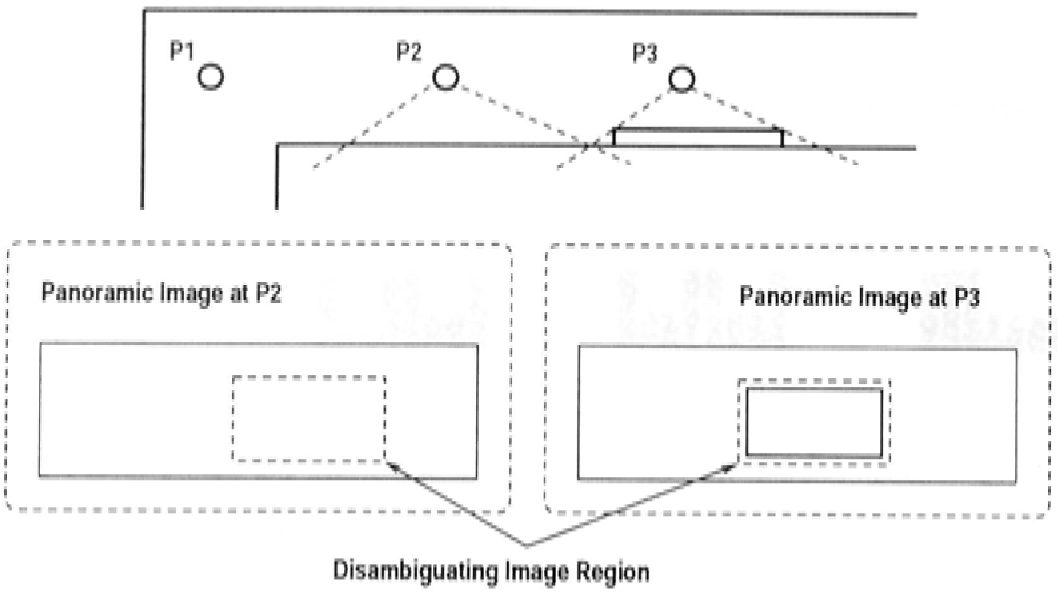


Figure 3.8: High level place representation: Complete panoramic image stored and later compared to find disambiguating features.

first learnt. By storing this information, revision of other levels of representation also becomes possible.

In panoramic images, regions of difference can be identified by matching entire images over all possible rotations of the image. The area of greatest difference can then be matched with the panoramic image of the current sensory view. The representation of places at this level and an example of identifying a disambiguating feature is shown in Figure 3.8. Of course this type of comparison will be computationally expensive, but is only applied when the localisation based on the other levels of spatial representation fails.

3.6 Integrating the Spatial Knowledge

The previous sections have detailed individually the three levels of representation that the system uses to represent a place in the topological map. For any given learnt place, the map contains all three levels of representation as shown in Figure 3.9.This information is to be used for more accurate, efficient and reliable localisation, and ultimately navigation, behaviours.

When using multiple levels of spatial representation, a framework for integrating the information between levels is needed. In addition, there must be some way of maintaining an internal belief as to the robots current position. As noted in the previous

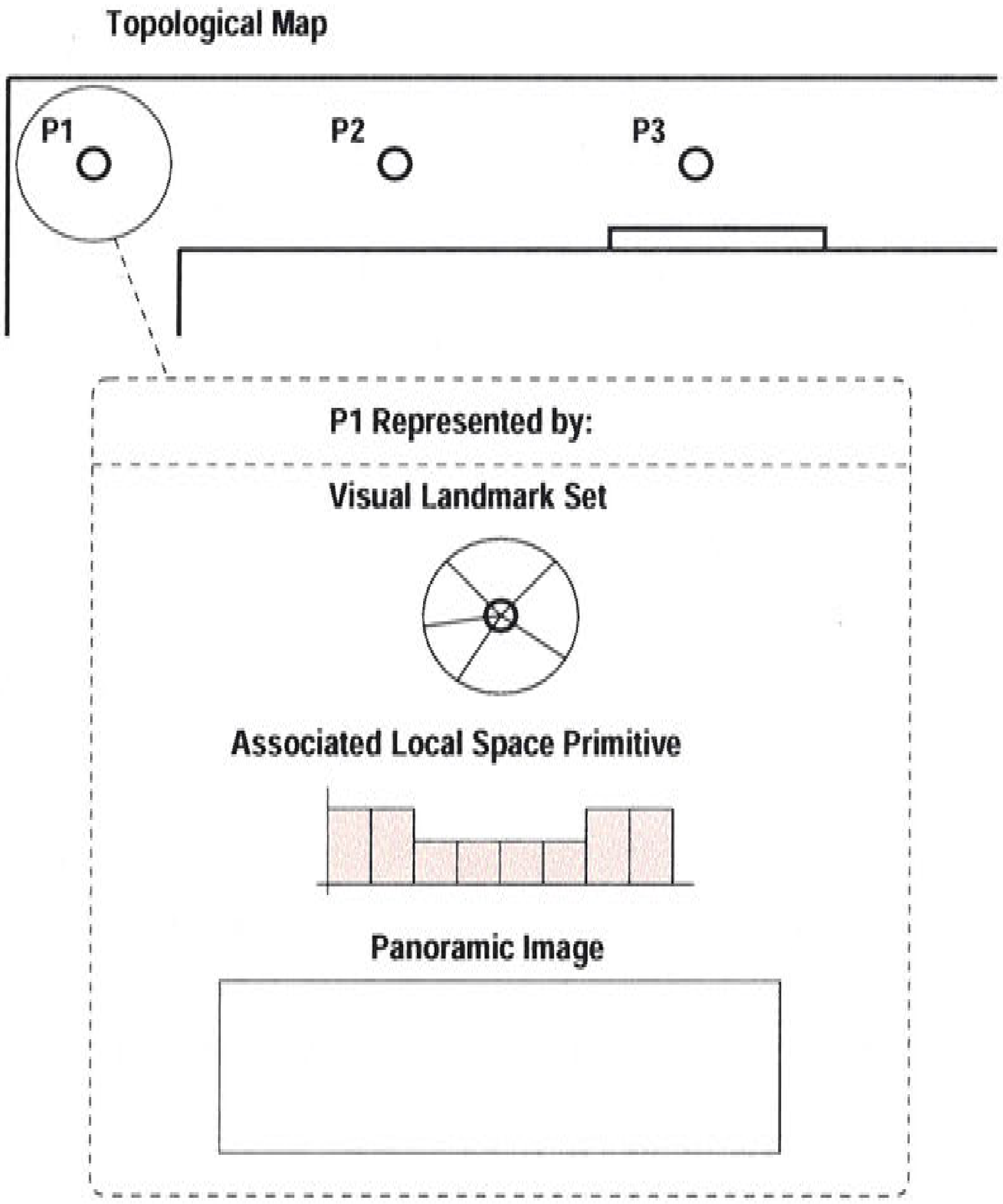


Figure 3.9: The levels of spatial memory in the topological map.

chapter, this belief must be able to form and maintain multiple hypotheses. Previous systems have used particle filters to approximate the Probability Density Function of the robot's position over a map. This system uses this method, but adapts it for a multi-level representation. The key problems in applying a particle filter to a topologically based, multi-level representation is in distributing the particles sensibly throughout the system and forming a current belief and maintaining that belief in transitions between places.

3.6.1 Guiding the Localisation Search

In this system individual particles will measure the probability of the robot being in a specific pose relative to a particular place in the topological map. Particles must be initially distributed throughout likely places. This choice of likely places is determined by the matching of local space profiles. But simply distributing particles based on the relative activation of these local primitives is not desirable because of the large computational cost involved in matching the current sensory view with the landmark sets of each likely place. A maximum number of places that can be matched within the real time constraints of localisation needs to be defined, and the distribution of particles restricted to this number of places. Alternatively, distribute them further in accordance with the local place primitives and perform matching with time switching, matching more likely places more often than less likely places. In this way no information is lost. Figure 3.10 illustrates the distribution of particles in the multi-level map, assuming there are only enough computational resources for matching the current scene against two distinct places per time step. In this figure particles are distributed randomly throughout the two places which have the the most similar local space profile when compared to the current scene, as determined by the local space primitives.

3.6.2 Maintaining an Internal Belief

Not only is there a need for distributing the particles throughout likely places, but there is also a need for evaluating the single most likely place that the robot is at any given moment. This evaluation is the robots internal belief as to where it is in relation to its spatial knowledge about the environment. But given that the information that is available is a sample driven approximation to a probability density function, how should this instantaneous position estimation be achieved? Should it be simply the location of the sample with the highest probability? Should it be the mean position of all samples? Obviously with such a multi-modal distribution such as robot localisation there is no trivial solution, and a heuristic solution suitable to most cases should be applied. In this system, as sensor observations are based on observed sets of landmarks representing places, there is a separation between position in terms of the places in the topological map and position in terms of local positioning within those places. Local position estimation is therefore taken to be the most probable position of all locally sampled positions.

Another issue that arises because of the segmentation of the environment into distinct places is that of passing knowledge about position between nodes in the topological

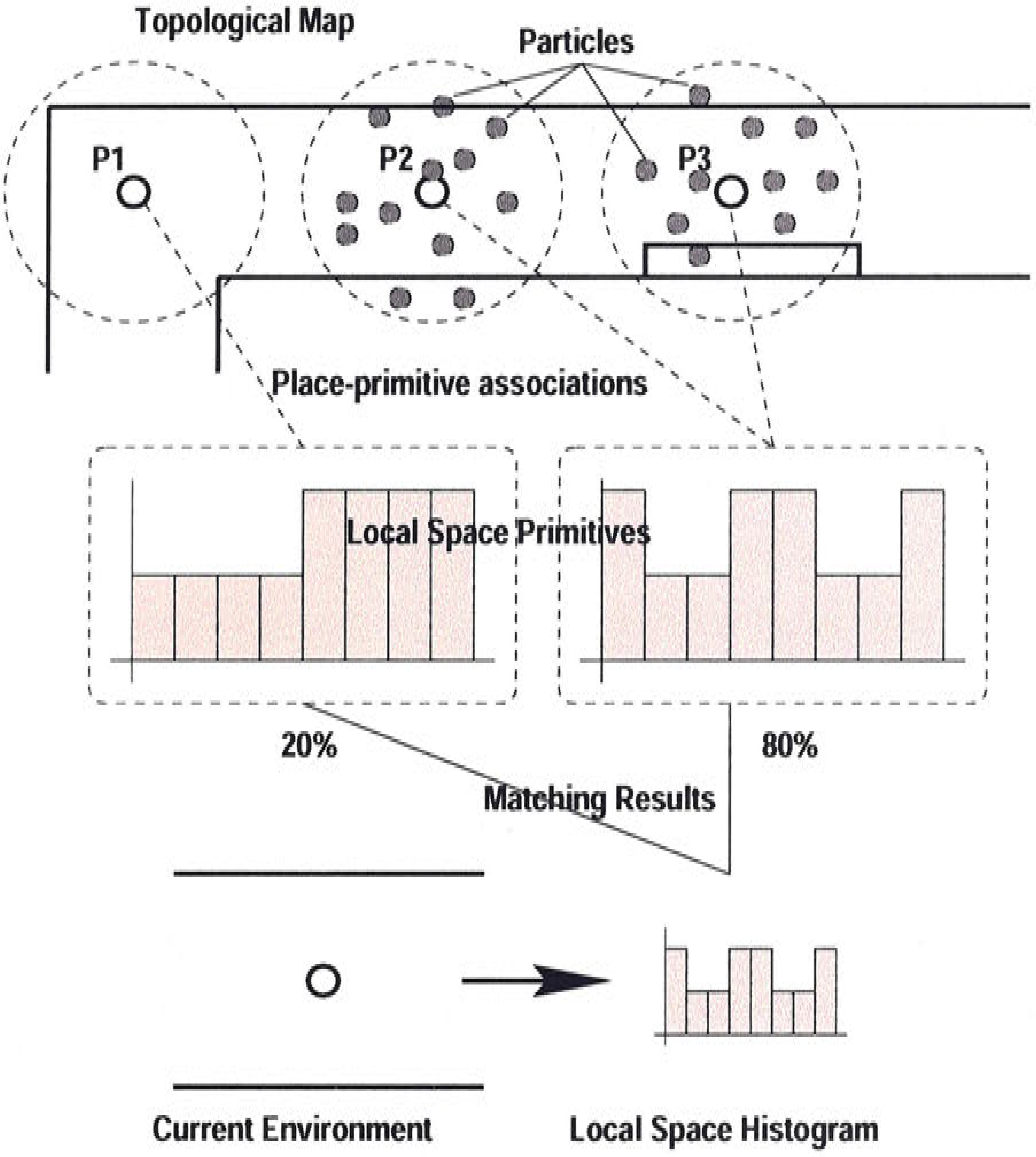


Figure 3.10: Distribution of particles throughout places based on local space matching.

graph. If there is a strong belief that a robot is in a particular position in relation to a particular place, this shown by a large amount of particles grouped tightly together about that position. If the robot leaves that place and makes a transition to an adjacent node in the place graph, then how can the position estimation information be passed to the next place? To a certain degree this will be achieved by sampling new particles at places determined by matching of the local space primitives. Relying on this alone however, will result in loss of the adjacency information captured by the topological map. Therefore some sampling at a position in the adjacent place given by the place transition information is desirable. Exactly how and when the re-sampling to adjacent places is achieved is left to the implementation details described in Chapter 8. In this way, position estimation is resolved on a place by place basis, which helps to limit computation, but still allows for exploration of new hypotheses and the passing of strong hypotheses between adjacent places on the map.

3.7 Robot System

This section details the hardware and software configuration of the robot system used in this thesis. A description of the hardware used in this system is given, in particular details of the Nomad XR4000 mobile robot and the panoramic vision sensor are presented. Then very general details of the software architecture is given and some design considerations are discussed. The environment in which real world robot experiments are performed is also discussed.

3.7.1 Nomad XR4000 Mobile Robot

The Nomad XR4000 is a mobile robot platform manufactured by Nomadic Technologies. A picture of the XR4000 robot is given in Figure 3.11. The panoramic vision system can be seen mounted on the top of the robot. The Nomad XR4000 has a plethora of other sensors which will not be used in the present system, such as a laser range finder, a web camera, as well as ultra-sonic and infra-red sensors.

The robot has holonomic drive and odometry measurements can be accessed in millimetres for motion along the x and y axes, and milli-radians about the axis of rotation. Access of all robot state information and control of robot motion in both position and velocity modes is achieved through the use of Nomadic Technologies *nrobot* control software.

The XR4000 has two on-board 750MHz CPU's running the Linux operating system, one of which has the robot control software running on it. The two CPU's are connected by a standard network connection. The network is also linked via a hub to a wireless Ethernet transmitter.

3.7.2 Panoramic Sensor

The panoramic vision sensor used in this system is shown in Figure 3.12. It is mounted in the center on the top of the mobile robot at the height of $\sim 1400mm$ (from the lens of

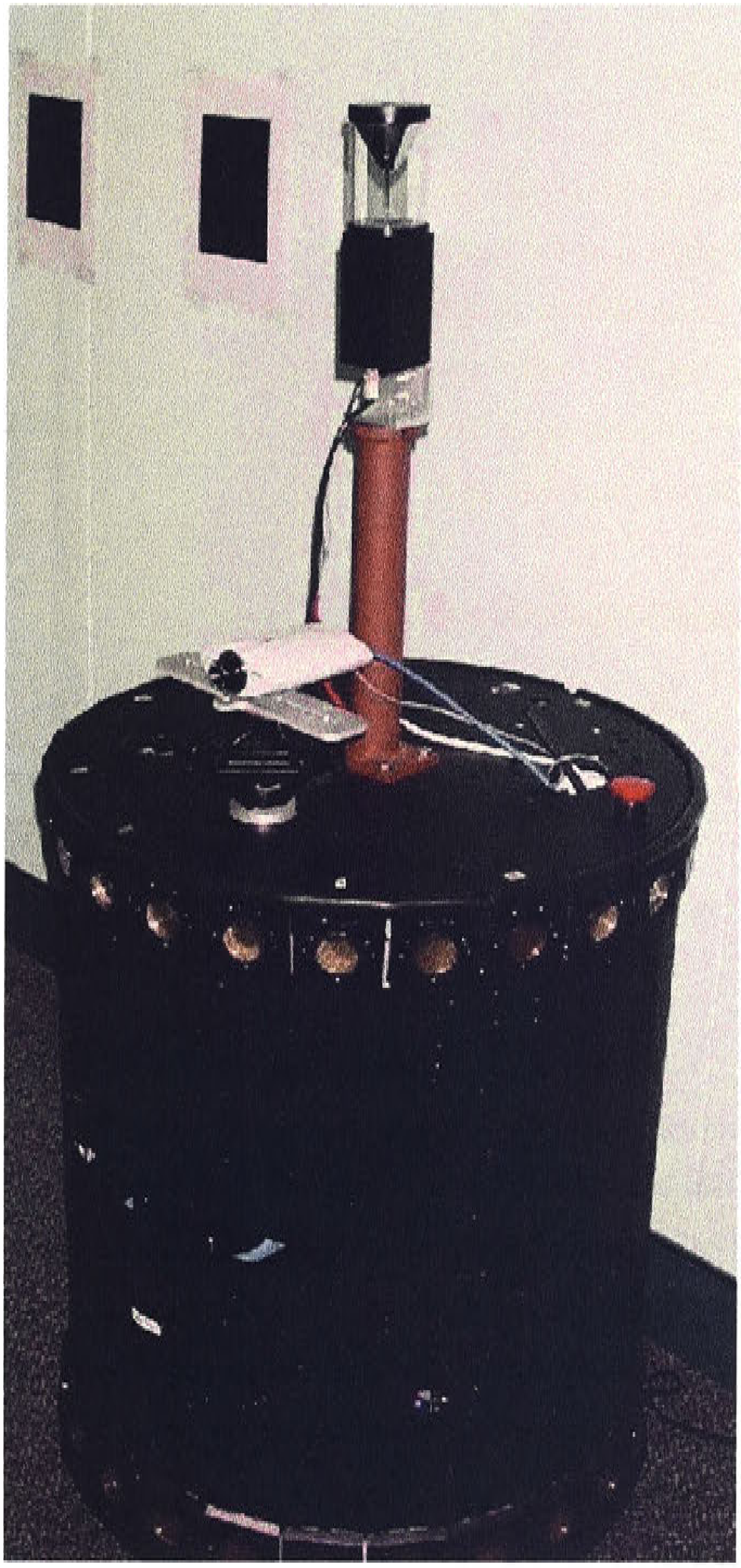


Figure 3.11: The Nomad XR4000 mobile robot by Nomadic Technologies.

the camera to the floor). The panoramic vision sensor is made up of a standard Sony CCD video camera pointed at the peak of a hyperboloidal mirror. The video camera can then capture a reflected image which covers 360 degrees of the surrounding environment. An example image from the video output is shown in Figure 3.2.

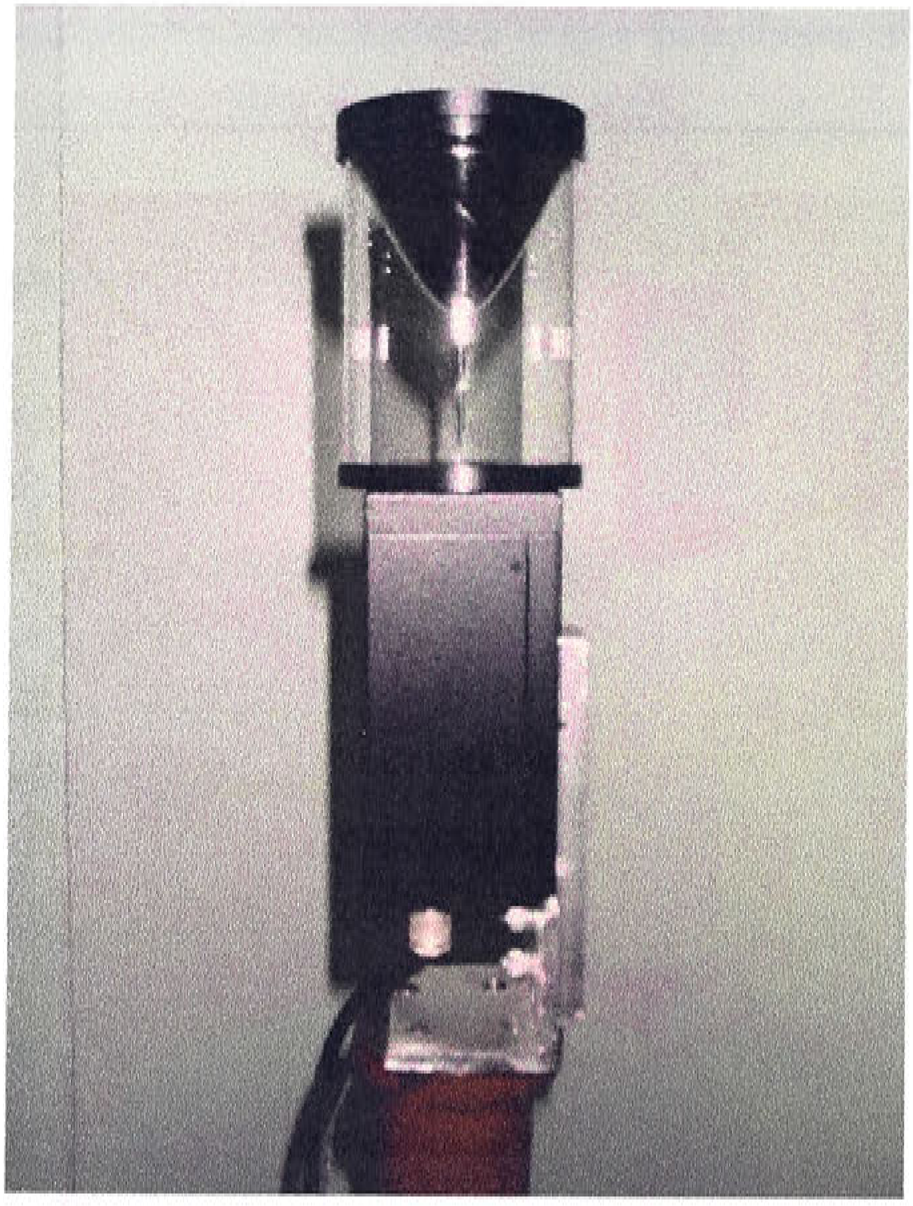


Figure 3.12: The panoramic visual sensor mounted on the mobile robot.

The hyperboloidal mirror shape was chosen due to the simplification of image processing it provides. Our research is not overly concerned with image quality, resolution or calibration of cameras. A panoramic image with sufficient resolution to track visual landmarks over small translations is all that is necessary. Camera calibration is not required as the local position estimation system depends solely on the radial angle of observations which is invariant in panoramic vision sensors.

The optical qualities of the sensor (Yamazawa et al., 1995) used in our research is shown in Figure 3.13. If the image space is defined in polar coordinates (r, θ) and a cylindrical coordinate frame about the mirror's central axis as (R, θ, Z), then the hyperboloidal mirror can be defined as:

$$\frac{R^2}{a^2} - \frac{Z^2}{b^2} = -1 \tag{3.1}$$

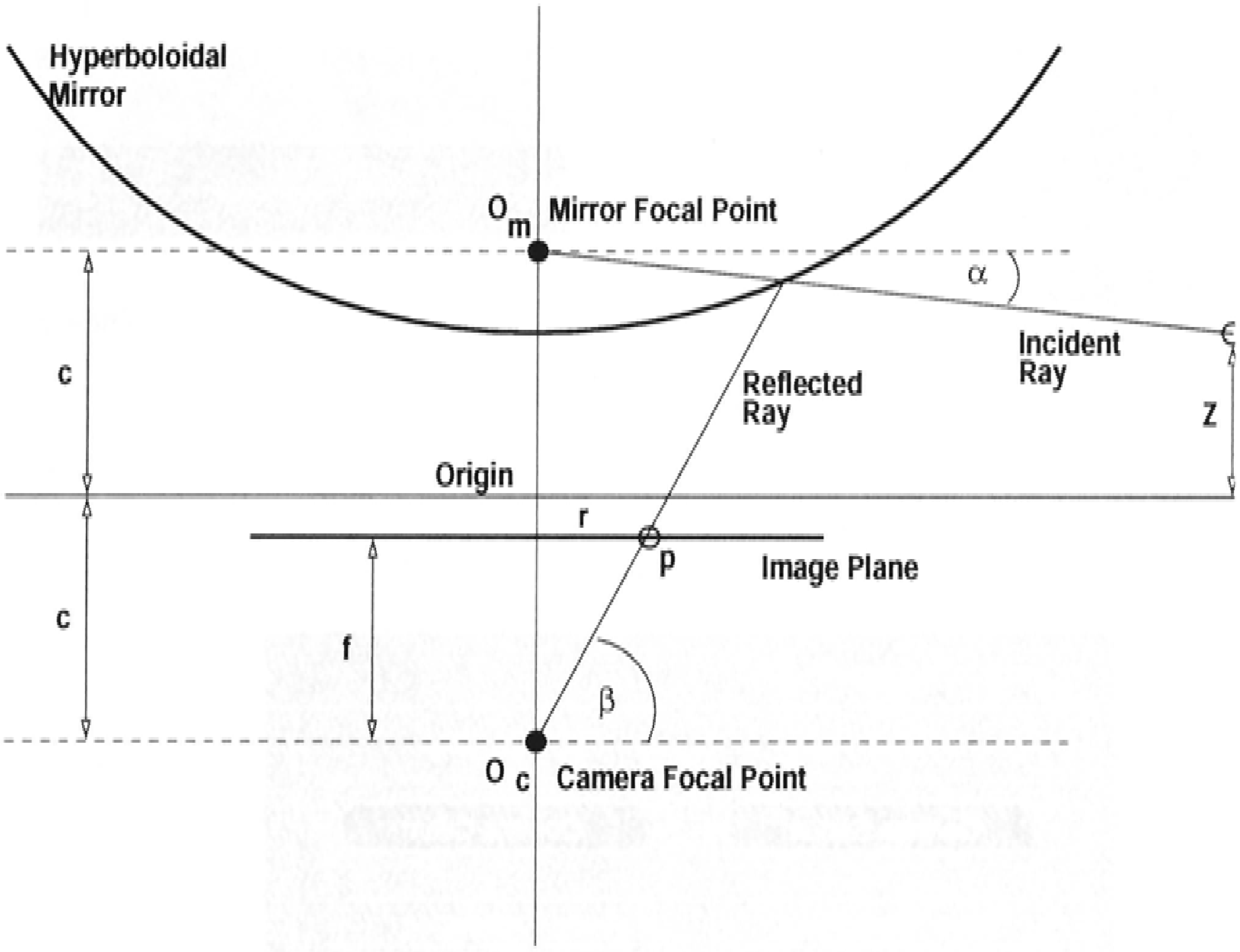


Figure 3.13: Optical configuration of the hyperboloidal mirror.

$$c = sqrta^2 + b^2 \tag{3.2}$$

The point in cylindrical space $(0, 0, +c)$ is the focal point of the mirror, and $(0, 0, -c)$ is the focal point of the camera.

Using basic geometry the range of the vertical axis of a cylindrical panoramic image can be calculated by:

$$r = ftan\beta \tag{3.3}$$

$$\beta = tan^{-1}\left(\frac{(b^2 + c^2)\, cos\alpha - 2bc}{(b^2 - c^2)\, cos\alpha}\right) \quad \left(0 \leq \beta \leq \frac{\pi}{2}\right) \tag{3.4}$$

Using these equations the camera image can be transformed to a cylindrical panoramic image. Figure 3.14 shows this concept pictorially. The circles in the warped camera image are un-warped into horizontal rows in the cylindrical panoramic image. As the varying thickness of the rings in the camera image shows, the pixel resolution

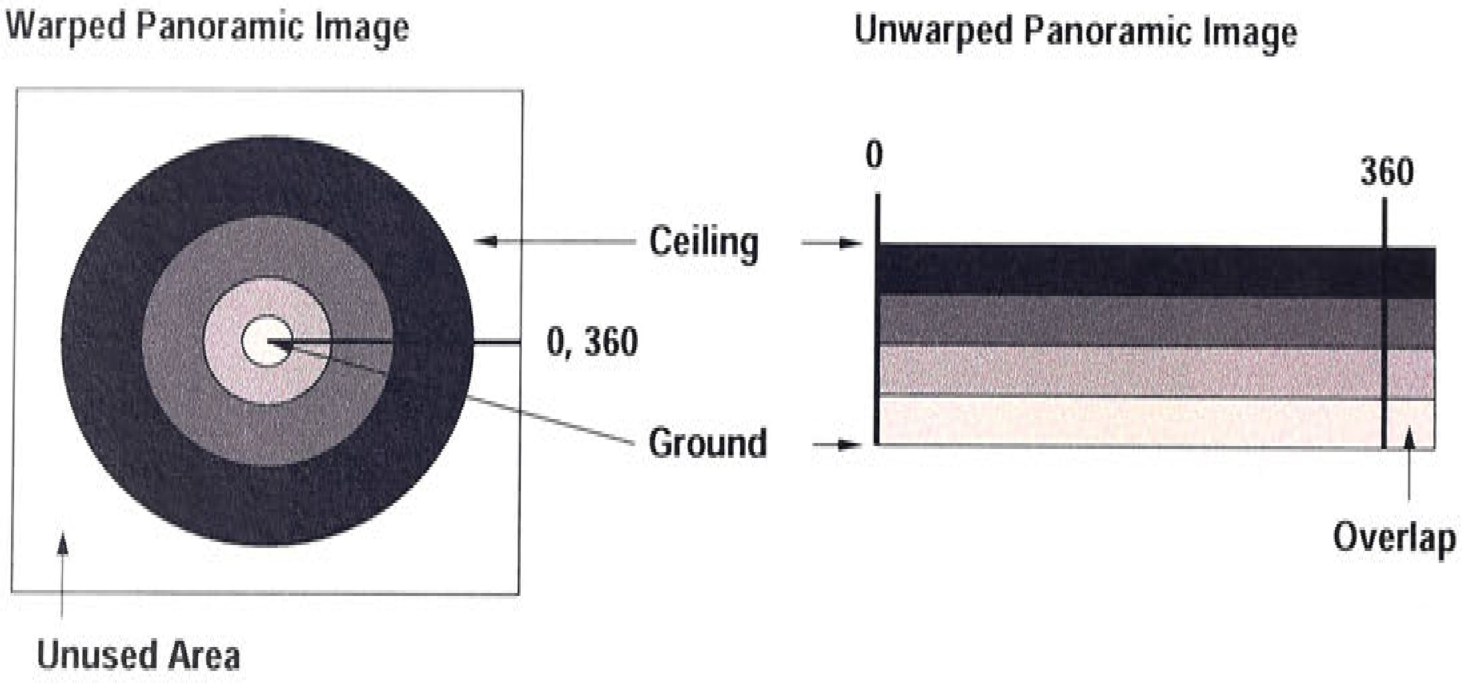


Figure 3.14: Unwarping the panoramic image.

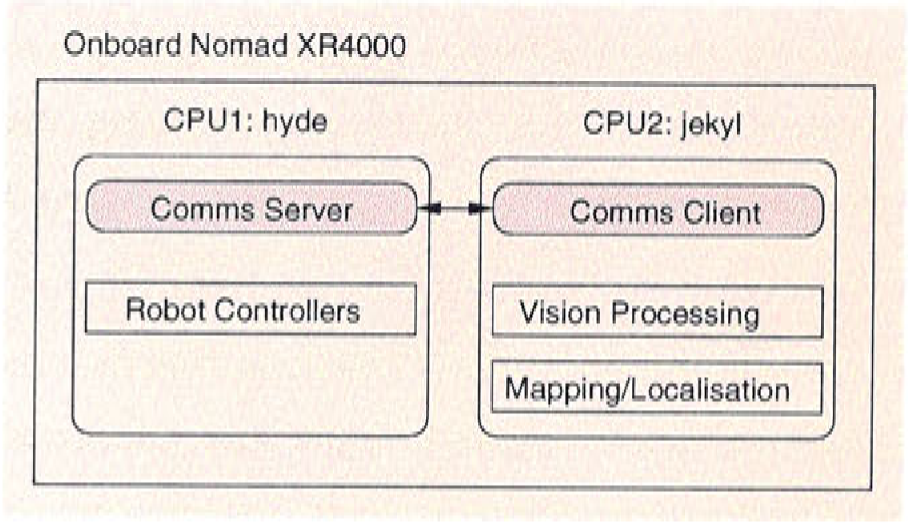


Figure 3.15: Software distribution over the two processors.

in the image depends on the elevation of the point in space. This transformation is completed in software. A look-up table, constructed using Equations 3.3 and 3.4, is used to speed this process up. The resulting cylindrical panoramic image repeats a small portion of the warped image. The left and right extremes of the image both portray the same area of the environment. This characteristic is intentional as it aids in tracking landmarks which move beyond the horizontal borders of the panoramic image.

3.7.3 Software Configuration

The main ideas in this thesis are directed towards mobile robot localisation. As was discussed in Chapter 2 localisation is a sub-component of the broader robot behaviour of navigation. As such the software developed for this thesis concentrates on solving the localisation problem and is not meant to provide a stand alone robot system capable of full autonomous navigation behaviour. Therefore to test the localisation software test harness programs were written to exercise the functions being tested in lieu of a fully functioning navigation system.

This means that there is no overlying software or control architecture, such as behavioural or reactive architectures, controlling the mobile robot. In fact the system being developed is designed to become a component module in such a robot architecture. As it stands the software configuration of the system is illustrated by Figure 3.15. This diagram shows that the system software is split between the two processors: robot control on the process named *Hyde*, and vision processing, mapping and localisation on *Jekyl*, with a client/server communication link handling the information passing between the two.

Harness programs are executed on *Jekyl* and call functions from the vision processing, mapping and localisation modules as required. Robot state information and motion commands are requested by the communication client. The harness program provides all movement directions for the mobile robot and it does not rely on the information being computed by the localisation process. A path planning component would be necessary to close the loop, using the localisation estimates to produce true mobile robot navigation.

3.7.4 Experimental Environment

In studying any type of system, biological or robotic, it is important to look at the environment the system will operate in. This section presents a short description of the environment in which a real world robotic systems will operate in order to experimentally validate our research.

In our research the majority of the real world robot experiments took place in the Robotic Systems Laboratory of the Australian National University. The environment is a typical office building with long visually sparse corridors connecting small offices. This environment is typical of mobile robot experiments in the literature. A feature of this environment is a relatively large open room. Localisation using vision sensors has

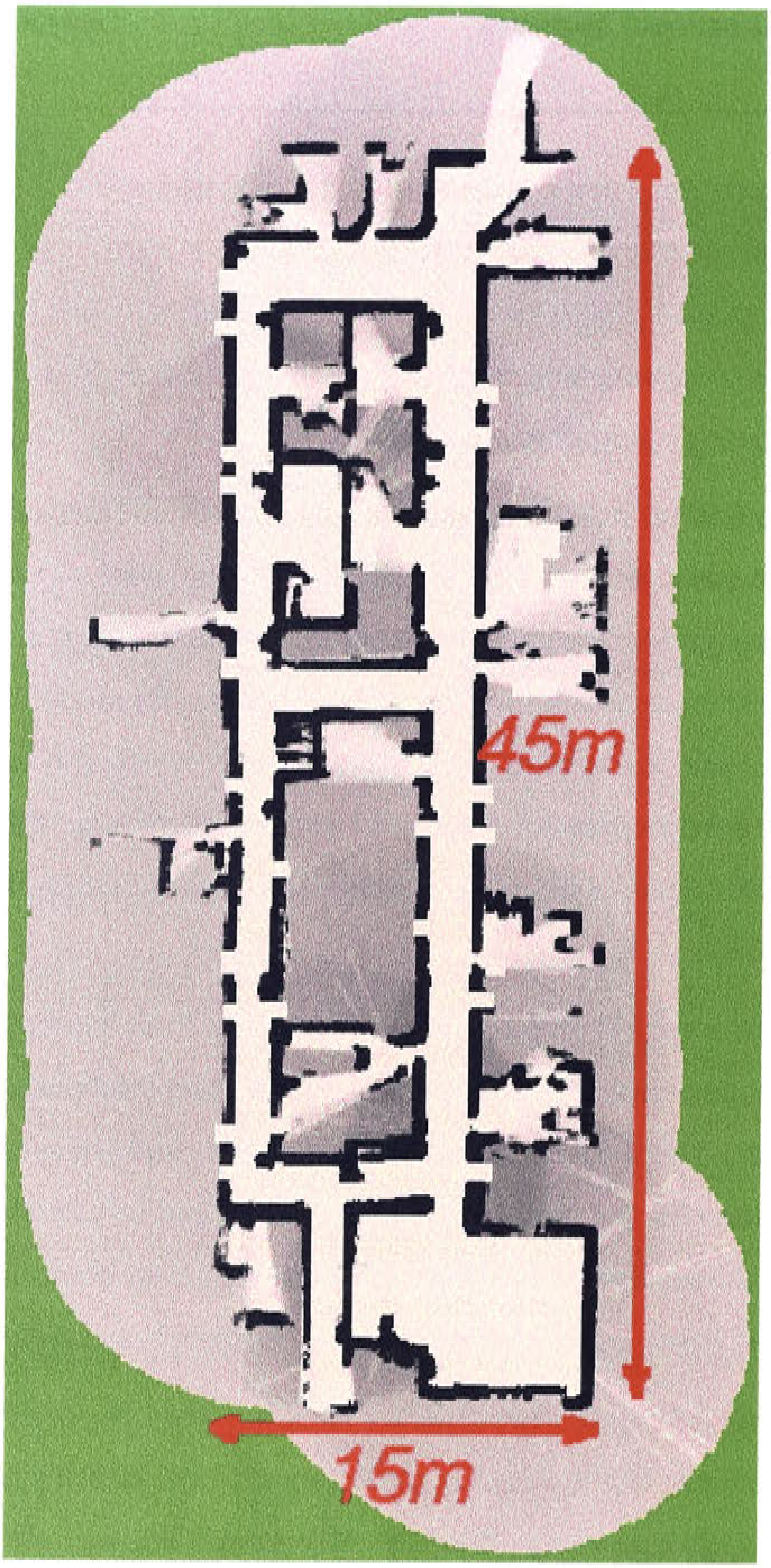


Figure 3.16: A map of the experimental environment.

typically not been applied to such an open area.

Figure 3.16 shows a map of the environment with the gross dimensions marked. This grid based map was constructed using a mapping and localisation system developed by Sebastian Thrun and colleagues of Carnegie Mellon University (Thrun, Beetz, Bennewitz, Burgard, Cremers, Dellaert, Fox, ahnel, Rosenberg, Roy, Schulte and Schulz, 2000), and kindly donated to the Robotics Systems Laboratory of the Australian National University. The large open room is in the bottom left of the map.

The map has a grid size of $10cm$ and has proven very accurate and reliable at position tracking over long paths. Our research, unless otherwise stated, uses this map and Thrun, Beetz, Bennewitz, Burgard, Cremers, Dellaert, Fox, ahnel, Rosenberg, Roy, Schulte and Schulz's (2000) localisation system as a measure of ground truth for real world mobile robot experiments.

3.8 Summary

This chapter has proposed a solution to the mobile robot navigation problem. This solution concentrates on the mapping and localisation components of navigation, in particular the problem of combining efficient global localisation with accurate local position tracking. Inspired from observations in biological systems of different complexity this system proposes the use of a multi-level spatial representation which can over come the above problems. Accurate local positioning is achieved by using unique visual landmarks to orient the robot with respect to a local reference place. Efficient global localisation is achieved by directing the localisation search using a notion of local space. Problematic cases are resolved by directly searching for disambiguating features. Internal belief is maintained by using a form of probabilistic reasoning implemented via a particle filter with special considerations to intelligently distribute particles throughout the representation.

3.8.1 Goal

The goal of our research, as stated in Chapter 1, is to build a localisation system for a mobile robot which solves the conflicting tasks of accurate local position estimation and global localisation. A problem which exemplifies the difficulty in this task is the kidnapped robot problem.

The main concept of this thesis is to use a multi-level representation of spatial memory in order to achieve this goal. Our research uses biological inspiration to design suitable levels of representation: low level visual landmarks; mid-level local space profiles;

and high level disambiguating features. Existing practical methods of mapping and localisation are enlisted and built upon to implement a localisation system which takes advantage of these levels.

The plan for reaching the goal is as follows:

1. *Map Representation:* detail each level of spatial memory representation (Chapters 4, 5, 6).

2. *Map Construction:* describe how maps can be be constructed using our representations for places in a topological map (Chapter 7).

3. *Local Position Estimation:* the low-level representation of visual landmarks are used to provide accurate local position estimation (Chapter 8).

4. *Global Localisation:* the low-level representation can also be used for global localisation. This process is computationally expensive. The higher levels of representation, especially that of local space profiles, can be used to constrain the global localisation search and reduce computational complexity (Chapter 9).

5. *Kidnapped Robot:* the multi-level representation can be combined to perform continuous global localisation and local position estimation. This process is applied to the kidnapped robot problem (Chapter 9).

3.8.2 Key Contributions of Thesis

The key contributions of this thesis in respect to the field of mobile robot navigation are:

- *Formalisation of multi-level spatial memory based on observed biological competences:* this work notes significant levels of behavioural competences and spatial memory that have been observed in biological organisms and proposes a multi-level spatial representation for artificial agents which reflects this range in capability, and the evolution of cognitive behaviours under the effect of terminal addition. This evolutionary process refers to building complex behaviours on top of existing, simpler behaviours. This system takes advantage of this design process.

- *Extension of automatic selection of visual landmarks in panoramic images:* the selection of visual landmarks using a Turn Back and Look behaviour is extended for use with a panoramic sensor and depth estimates of visual landmarks are made using a form of bearing only Simultaneous Localisation and Mapping (SLAM).

- *Geometric landmark sensor models for hypothesis evaluation:* An ellipsoid-line intersection model is developed to evaluate the likelihood a given observation of a landmark in a panoramic image was made from a sample robot state. This effectively lets the system assign a value to the probability that the current observations were made from a hypothesised position in the internal map.

- *Accurate local positioning in a topological map:* visual landmarks are used to not only define places in a topological map but also to provide for accurate local positioning in conjunction with a particle filter based probabilistic position estimator.

- *Division of probabilistic global localisation search by topological mapping:* By using a set of unique landmarks to represent places, this allows for localisation on a topological level by matching landmark sets, thus restricting the global localisation search to areas defined by places in map.

- *Multiple cues for restricting global localisation search:* the mid-level of the spatial memory representation can be used as a cue for restricting the global localisation search. By incorporating cues with low matching costs the global localisation process can be made more efficient.

- *Loss and recovery of localisation belief:* a method of maintaining a central localisation estimate is developed, which can detect when localisation is lost, and trigger relocalisation.

- *A solution to the kidnapped robot problem:* the current system goes a long way towards solving the kidnapped robot problem. Accurate local positioning can be achieved through observations of visual landmarks; the loss of position tracking can be detected through monitoring the robot's belief in it's estimate; relocalisation can be achieved by the restricted global localisation search.

These contributions allow for efficient global localisation and accurate local positioning. In doing so mobile robot systems can be built which can reliably localise themselves in the presence of problematic situations, and subsequently to realise robust navigation behaviour. The following chapters detail the specific implementation details and experimental validation of this solution.

Chapter 4

Visual Landmarks for Low Level Representation

This chapter details the implementation of the first level of spatial representation introduced in Chapter 3: that of visual landmarks. Landmarks are distinct features of the environment whose presence in the sensory view can aid in the localisation task. Upon encountering a novel place in the environment, a mobile robot can form an abstraction of the location in the environment by selecting prominent landmarks from the sensory view from the reference position. A robot which subsequently observes these landmarks can infer that it is located near the reference position from which the abstraction was formed. By observing such distinct features a robot can also perform triangulation calculations in order to accurately determine their local position.

The outline of this chapter is as follows: Section 4.1 describes the need for automatically selecting a set of visual landmarks to form a low-level representation of a place in the environment, and a two step process to do so is outlined. Section 4.2 details the first step in this process, that of selecting static landmarks being selected on the basis of their local uniqueness. The second step, evaluating static landmarks over a dynamic motion phase, is reported in Section 4.3. This dynamic motion phase is biologically inspired and is named the *Turn Back and Look* (TBL) phase as it mimics wasps turning back and looking at their hives before going on foraging flights. Section 4.4 presents a method of recognising landmark sets in sensory views, while Section 4.5 reports on experiments which test landmark recognition performance. Next, in Section 4.6,

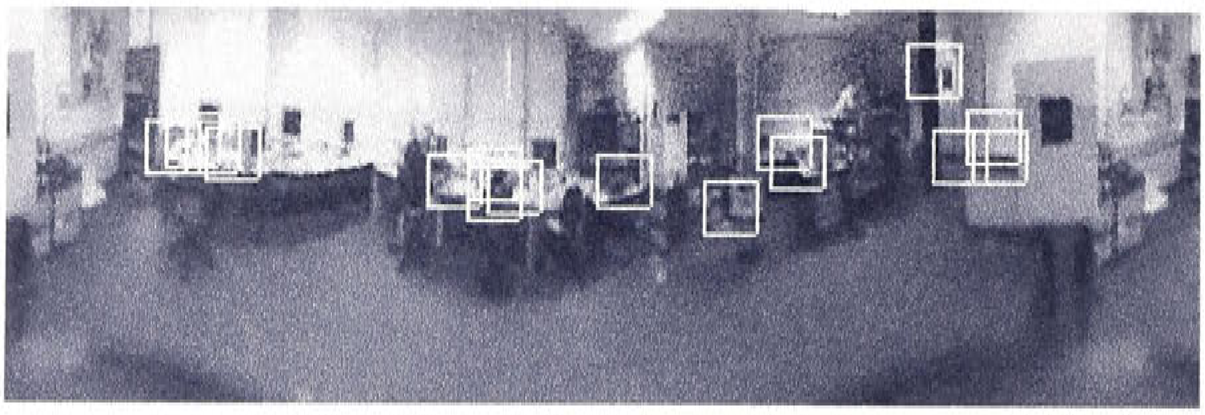


Figure 4.1: An example of visual landmarks in a panoramic image.

the use of the dynamic landmark selection phase to produce an estimate of landmark depth is described and experiments to validate these estimates are presented. Finally, Section 4.7 provides a discussion and summary of the issues associated with implementing the low-level visual landmark representation.

4.1 Automatic Visual Landmark Selection

The process of landmark selection is aimed at producing a set of landmarks which are unique, maximise the area of the environment the place represents, and allow for accurate local positioning. This means that landmarks must be reliable, strongly identifiable, and they must be distributed throughout the image. They also must be able to withstand image variance due to temporal and translational distortions. Bianco and Zelinsky (1999) proposed a method which selects landmarks based on their static uniqueness and their dynamic reliability. This approach has been extended to incorporate the advantages of the panoramic sensor. In this system, visual landmarks are 16×16 pixel regions in the grey-scale panoramic image.

An example of visual landmarks in a panoramic image is shown in Figure 4.1. Of course for a mobile robot to act autonomously it must have a method for extracting these landmarks automatically. Figure 4.2 shows the process this system uses to automatically select visual landmarks. An image is captured from the reference position, from which 32 static landmarks are selected. These landmarks are then tracked and evaluated for their reliability over a Turn Back and Look (TBL) movement in the dynamic phase. The 16 most reliable landmarks are selected to represent the place as a landmark set.

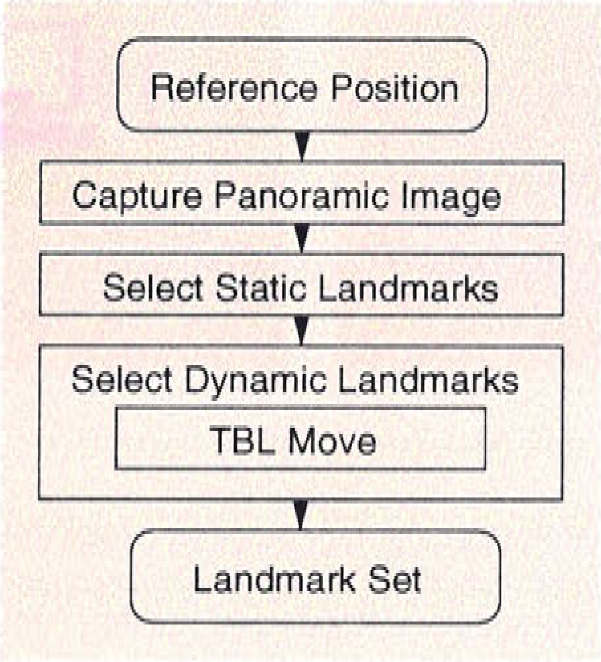
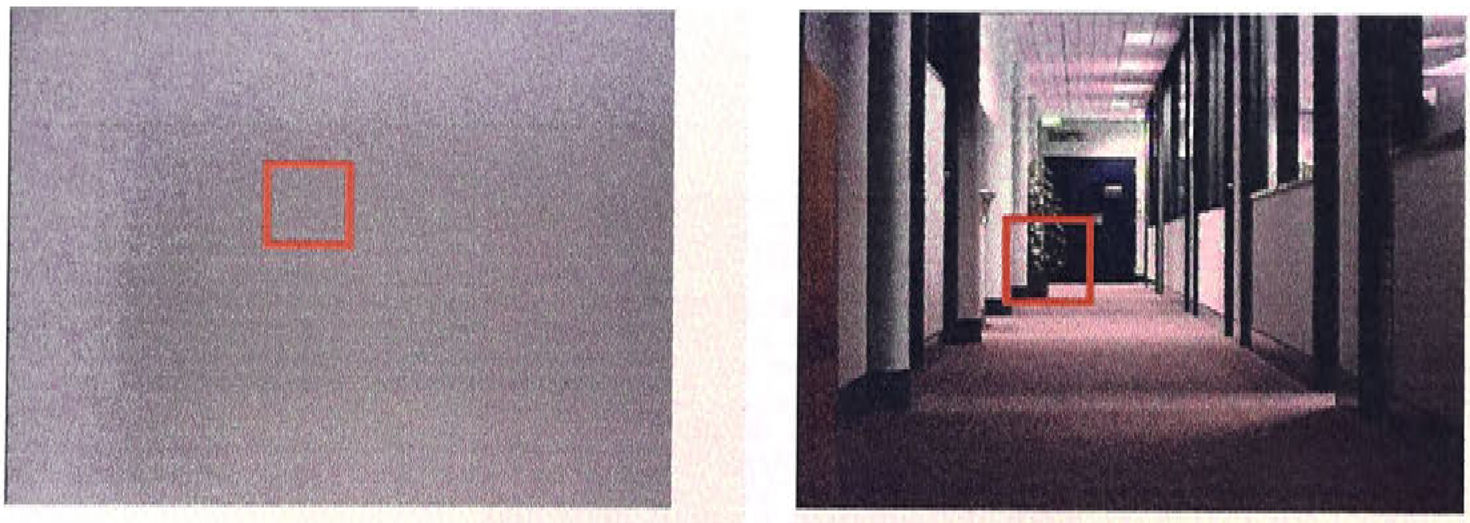


Figure 4.2: The process of automatic visual landmark selection.

The two phases of landmark selection, static and dynamic, are described in the following sections. The static phase is meant to ensure landmarks are distinct in their local region of the visual scene and as such are strongly identifiable. The dynamic phase is meant to select landmarks which are reliable and recognisable over a large area, and this characteristic is tested by comparing the range over which static and dynamic landmarks can be recognised. Landmarks should be recognisable in the presence of varying illumination conditions and a method of achieving this and some results are presented. By associating a depth with landmarks, more accurate local position estimates can be formed and a method for doing this is described in Section 4.6.

4.2 Static Landmark Selection

Static landmarks are selected from a static scene on the basis of their local uniqueness. Two examples of landmarks are shown in Figure 4.3. Image a) shows a landmark which is similar to its surroundings and subsequently useless for localisation or navigation task. Image b) on the other hand shows a landmark which is distinct when it is compared to the image region immediately surrounding it. Obviously an automatic landmark selection system should pick out landmarks which are locally unique, similar to that displayed in image b) as opposed to image a). In this system local uniqueness is defined as the degree to which the landmark template differs from the area of the image immediately surrounding the landmark. This approach is based on



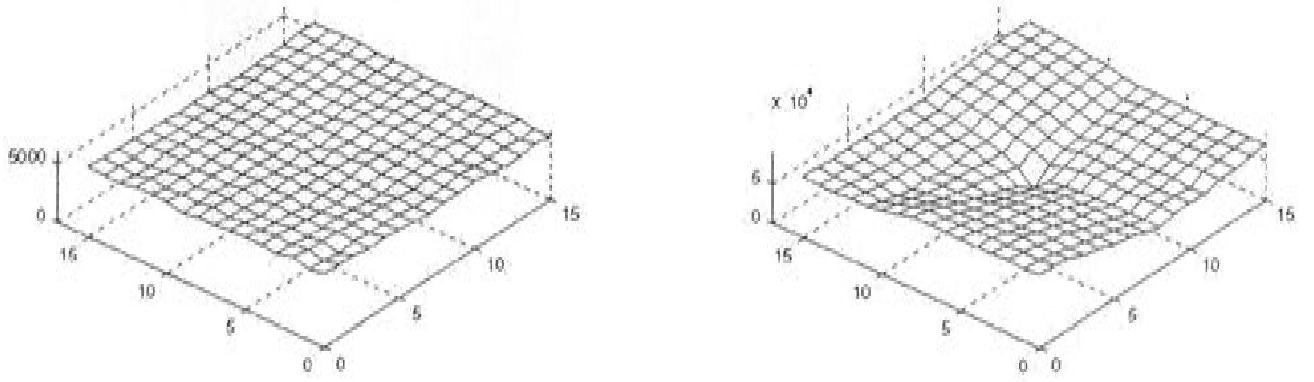
a) b)

Figure 4.3: Examples of landmarks. The landmark in image a) is unreliable and has a low local uniqueness. The landmark in image b) is more reliable and has a higher local uniqueness. Figure from Bianco and Zelinsky (1999).

the 'The Valley Method' proposed by Mori et al. (1995) to generate attention tokens in a scene, which in turn appears to be an instance of a Moravec interest operator (Moravec, 1977) applied to feature tracking. Bianco and Zelinsky (1999) adapt this method for the present task of automatic selection of landmarks. When evaluating an image region as a potential landmark, two panoramic images are captured, the landmark template in the first image is matched with its surrounding region in the second image using correlation matching. Because the scene is static the only difference between the two images are those introduced by camera noise. In this case a standard Sum of Absolute Differences correlation algorithm has been used for the matching process:

$$SAD = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I_{ij} - T_{ij}| \tag{4.1}$$

where a template T of pixel size $M \times N$ is being correlated with an image region I of the same size. It is worthwhile noting here that normalised correlation matching is not desirable here as the process is trying to identify the magnitude of differences within regions of the same scene. Normalised correlation, which uses the mean intensity of pixels in an image to eliminate illumination effects in the correlation process, would



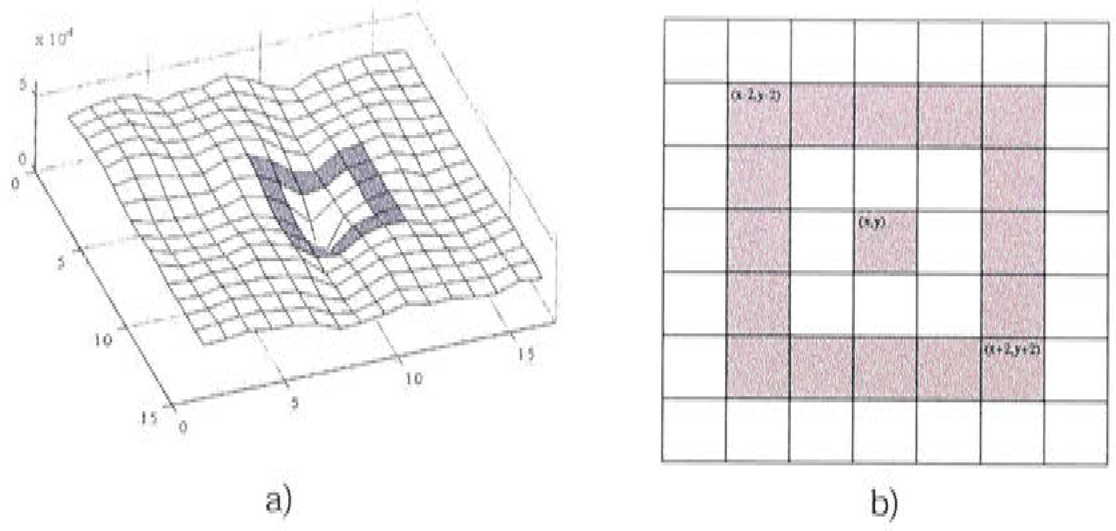
a) b)

Figure 4.4: Examples of distortion matrices from the landmarks in Figure 4.3. The matrix in a) is from local correlation matching around the unreliable landmark, the matrix in b) from the reliable landmark. The horizontal axes denote the matrix resulting from matching the 16×16 pixel template over a 32×32 pixel search window. The vertical axes shows the SAD correlation value for each match. Figure from Bianco and Zelinsky (1999).

tend to exaggerate all differences. A formula for normalised correlation is given in Equation 4.3.

Figure 4.4 shows the correlation results obtained by matching each 16×16 pixel landmark template, from the images in Figure 4.3, on a 32×32 search window centered on the original template. The unreliable landmark produces the image distortion matrix shown in image a) which is uniformly low meaning that the landmark correlated highly with its surrounding image region. In the distortion matrix in image b), the more reliable landmark produces a valley corresponding to where the landmark template was matched with itself, thus having a high correlation (low distortion). By comparing the depth of this valley in relation to the surrounding distortion, a measure of how unique the landmark is in its local region can be obtained. In practice this measure is the ratio of the minimum global matching distortion to the local minimum (from the match of the landmark and the sixteen surrounding values). This idea is illustrated in Figure 4.5 where the local minimum is located in the region highlighted by the grey square, and the global minima is shown as the bottom of the valley. More formally local uniqueness is defined by:

$$r = 1 - g/g' \tag{4.2}$$



a)

b)

Figure 4.5: Finding a measure of landmark reliability: ratio of the global minimum (valley) and the surrounding local minimum (from the area covered by the grey square. Figure b) is a over head view of the surface in a). Figure a) from Bianco and Zelinsky (1999).

where r is the reliability of the landmark, g is the distortion of the landmark matched with itself, and g' is the minimum matching distortion from the surrounding circle of pixels. Given that g should only result in distortion due to noise, then the higher the distortion of the minimum of the surrounding templates, the steeper the valley in the distortion matrix and subsequently the more unique the local template should be. An example of this concept is shown in Figure 4.5.

In order to select the 32 static landmarks that the system uses, two panoramic images are captured from the reference position and an exhaustive search over all possible image regions is carried out to select the 32 image regions or landmark templates which have the highest measure of local uniqueness as determined above.

By dividing the panoramic image into four sectors, roughly corresponding to forward, back left and right, and selecting an equal number of landmarks from each sector, the resulting landmark set is distributed throughout the image. This assists in local position estimation as well as ensuring the visibility of some landmarks when parts of the visual field are occluded.

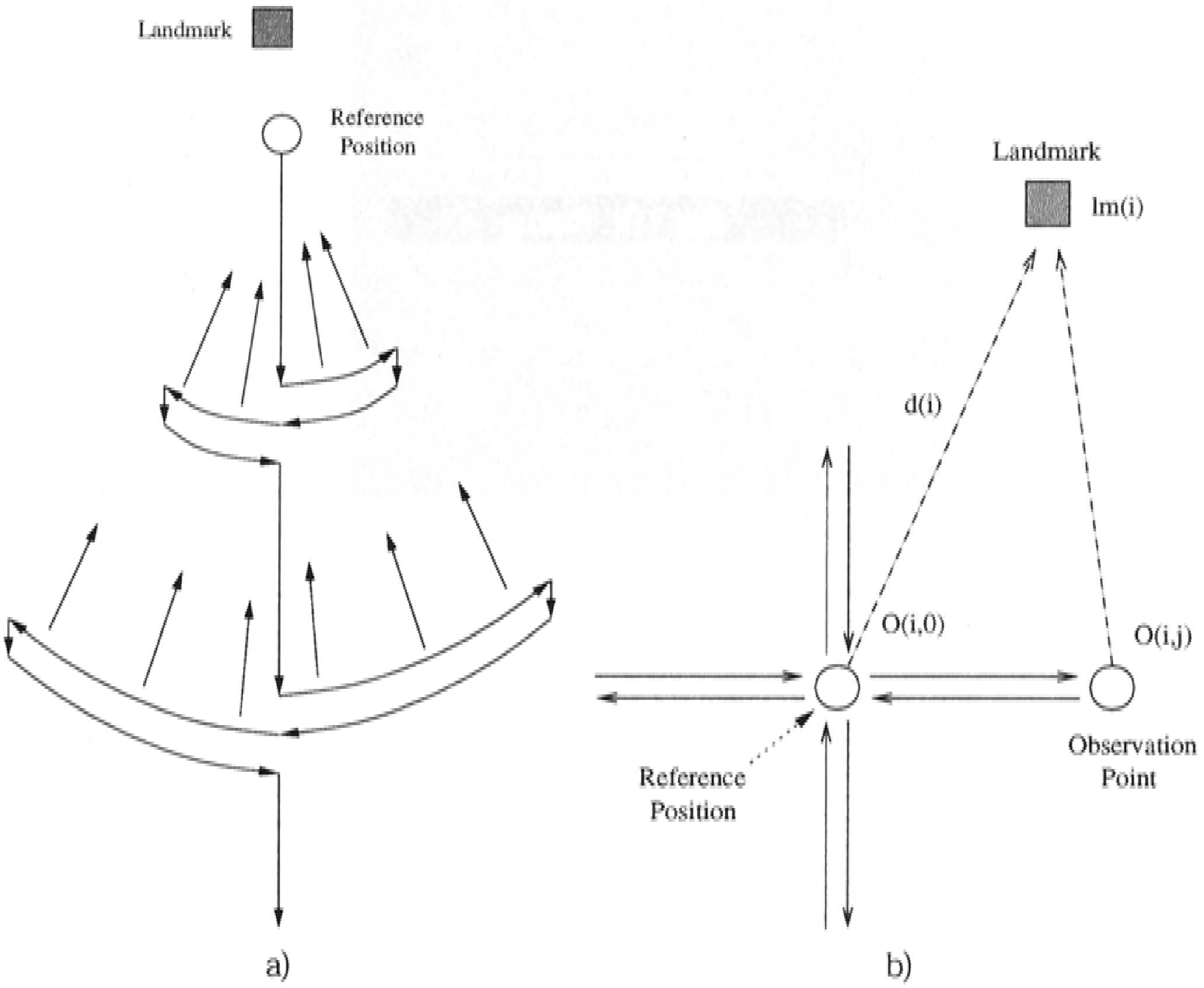


Figure 4.6: TBL paths for learning places with mobile robotic. Image a) is the robot motion path used by Bianco and Zelinsky (1999) and image b) the modified TBL phase for dynamic landmark selection with panoramic vision.

4.3 Dynamic Landmark Selection

Landmarks selected for their static uniqueness are then evaluated for their dynamic reliability. In this phase the robot moves about the reference position from which the static landmarks were selected while observing the potential landmarks. By tracking the landmarks throughout this biologically inspired movement, the landmark's resistance to changes in lighting and perspective can be measured. In this way landmarks which do not significantly distort from their original appearance during these movements can be chosen to represent the place.

4.3.1 Turn Back and Look Movement

This series of movements about the reference position was inspired by insect behaviour and is termed the Turn Back and Look behaviour (Lehrer, 1993) (Collett, 1996).

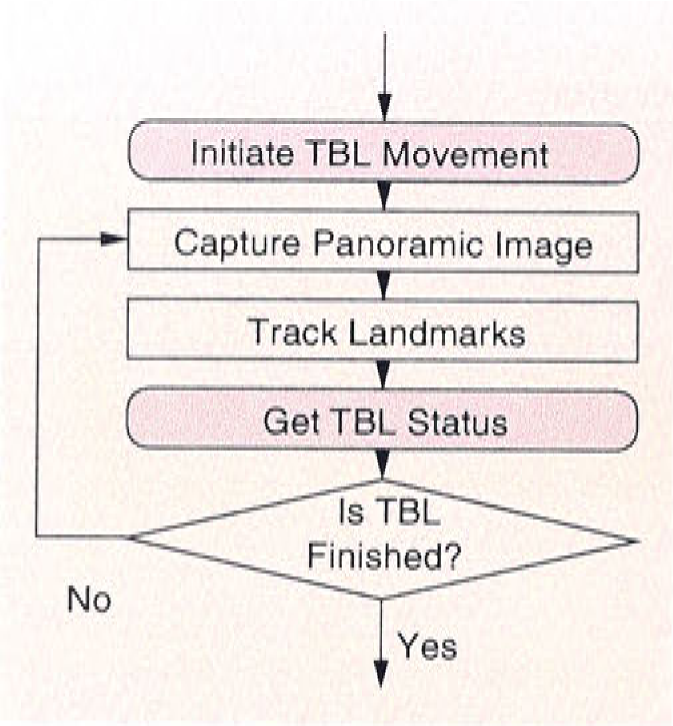


Figure 4.7: TBL movement algorithm on vision processing CPU

Bianco and Zelinsky (1999) first implemented this behaviour on a mobile robot using a normal camera configuration and moving in arcs away from the goal position while facing back at the potential landmarks. In this system the movement has been extended to take into account the greater field of view of the panoramic camera system. The differences in the TBL movements between the two systems can be seen in Figure 4.6.

In the current system dynamic landmark selection is accomplished using the method illustrated in Figure 4.7. This computation occurs on the CPU dedicated to vision processing and localisation. The ovals shaded grey denote stages where communication with the CPU hosting the robot controller occurs. Basically the system initiates the TBL movement then enters a cycle of capturing the panoramic image and tracking landmarks until the TBL movement has finished.

By selecting landmarks which track reliably throughout the TBL move, a landmark set with greater reliability and coverage about the reference position can be selected.

4.3.2 Landmark Tracking

The reliability of landmarks is determined by locating the landmarks throughout the TBL movement by correlation template matching, and averaging their correlation measures over the entire movement. Template matching over this movement is achieved

Table 4.1: Expanding Search Window Landmark Tracking

```
-- initialisation
for each $landmark$ $i$ in $landmark$ $set$
    $ssw^i = 32$
    $(x_i, y_i) = static\_landmark\_position(i)$
end
$t = 0$
-- iteration
for each $image$ $frame$ at step $t$
    for each $landmark$ $i$ in $landmark$ $set$
        if $landmark\_match(i_{t-1}) < threshold$
        then
            $ssw_i = ssw_i + 2$
        else
            $ssw_i = 32$
        end
        $landmark\_match(i_t) = corr(image_t, template_i, x_i, y_i, ssw_i)$
    end
    $t = t + 1;$
end
```

using the normalised cross correlation template matching algorithm:

$$cor = \frac{\left(\left(\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I_{ij} T_{ij}\right) / NM - \overline{IT}\right)}{\sqrt{\sigma_I \sigma_T}} \tag{4.3}$$

where a template T of pixel size $M \times N$ is matched with image region I of the same size, and σ_I and σ_T are the standard deviations of I and T respectively. This equation produces a result between in the range of $[0..1]$ with 1 being perfect correlation. The normalised cross correlation process results in matching between image regions with the same mean intensity level reducing the affects of varying illumination. Landmarks can then be tracked irrespective of lighting conditions resulting in more robust landmark recognition.

Because the exact position of the landmark within the image is initially known and the robot is moving at slow speeds throughout the TBL movement, the process of locating landmarks in each frame can be sped up by tracking the landmarks within a specified search widow of the image, centered on the last known landmark location. The tracking algorithm (shown in Table 4.1) uses an expanding search window ($ssw =$ size of the search window) centered on the landmarks last observed position (x_i, y_i). Initially the search window for each landmark is 32 pixels square and is centered around the image position determined by the results of the static landmark selection phase.

Upon entering the iterative phase, if the matching value for any landmark falls below a *threshold* value then the size of the search window is increased by 2 pixels along the horizontal image axis. If the matching values subsequently rises above the *threshold* then the search window size is reset to 32. This means that if tracking is lost for any given landmark, the image region within which the landmark is matched grows until tracking is regained. This expanding search window is limited to 90 pixels, or a quarter of the visual field, as any region of field likely to be of use as landmarks will not be displaced more than 90 degrees as a result of the small translations involved in the TBL movement. In our system the *threshold* value is set to 0.7.

At some stage movement of the robot will cause tracked landmarks to pass beyond the edge of the image. Due to the panoramic nature of the sensor, and the overlap in the visual image, landmarks that pass beyond the left or right edges of the panoramic image will already be in view in the opposite side of the image. Thus whenever a tracked landmark's search window goes beyond the edge of the image, the system automatically checks the corresponding region on the opposite side. In this way tracking can be maintained even when landmarks pass beyond the edge of the current image.

The cycle of capturing the panoramic image and tracking the 32 landmarks takes on average 129ms (Pentium II 750). With the panoramic image capture and unwarping taking 6.5ms and landmark tracking 25-75ms, the tracking time varying with the size of the search window. Throughout a TBL move the vision and landmark tracking system can process and log or display approximately 400 image frames. If dynamic landmark selection is run without image logging, approximately 1000 frames can be processed. For each landmark a reliability measure can be obtained by averaging the landmark's tracking results over this sequence of frames. This reliability measure can then be used to rank the 32 static landmarks in terms of resistance to distortions introduced by the TBL movement.

From the 32 static landmarks, the four most reliable landmarks in each sector of the panoramic image, as determined by the dynamic selection phase are chosen. Landmarks can simply be represented as x, y locations in the image. These 16 chosen landmarks then form the landmark set which represents the place being learnt. An exam-

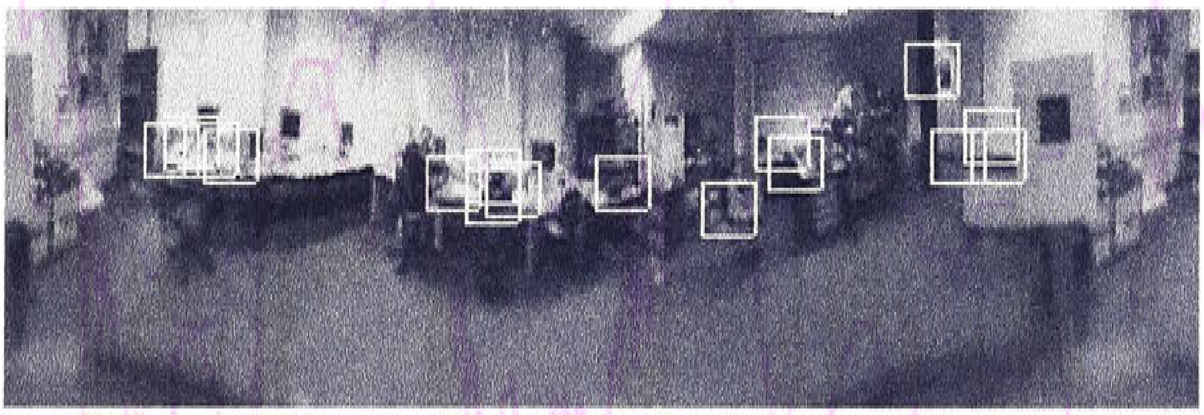


Figure 4.8: An example of landmarks selected to represent a place using static and dynamic landmark selection.

ple of the landmark set selected by this process is given in Figure 4.8.

4.4 Recognising Landmark Sets

The reason for learning a representation of a place is so that the robot can recognise it when it revisits the place. Therefore the best way to evaluate potential methods of acquiring the representation is to measure the degree to which those representations can be recognised. It is important to have an understanding of how these representations will be used, and as such methods for recognising landmark sets in panoramic images are presented here. These recognition methods can also provide some empirical evidence as to suitability of the representation.

The basic method to perform landmark recognition in a panoramic image, is to search the panoramic image for regions containing a similar spatial pattern, that is regions which appear similar to the landmark template. The algorithm should identify the region which is most similar to the landmark template and also it should preferably do this in the shortest amount of time. Although the speed of landmark recognition is not critical to system performance, as landmarks can be tracked at great speed after they are recognised, it is a behaviour whose execution frequency will grow as more places are added to a topological map. Thus a fast recognition rate for multiple places is desirable but not at the expense of recognition performance.

Two methods, brute force template matching and pre-matching feature extraction are investigated. The advantages, disadvantages and the tradeoffs of these two approaches are discussed.

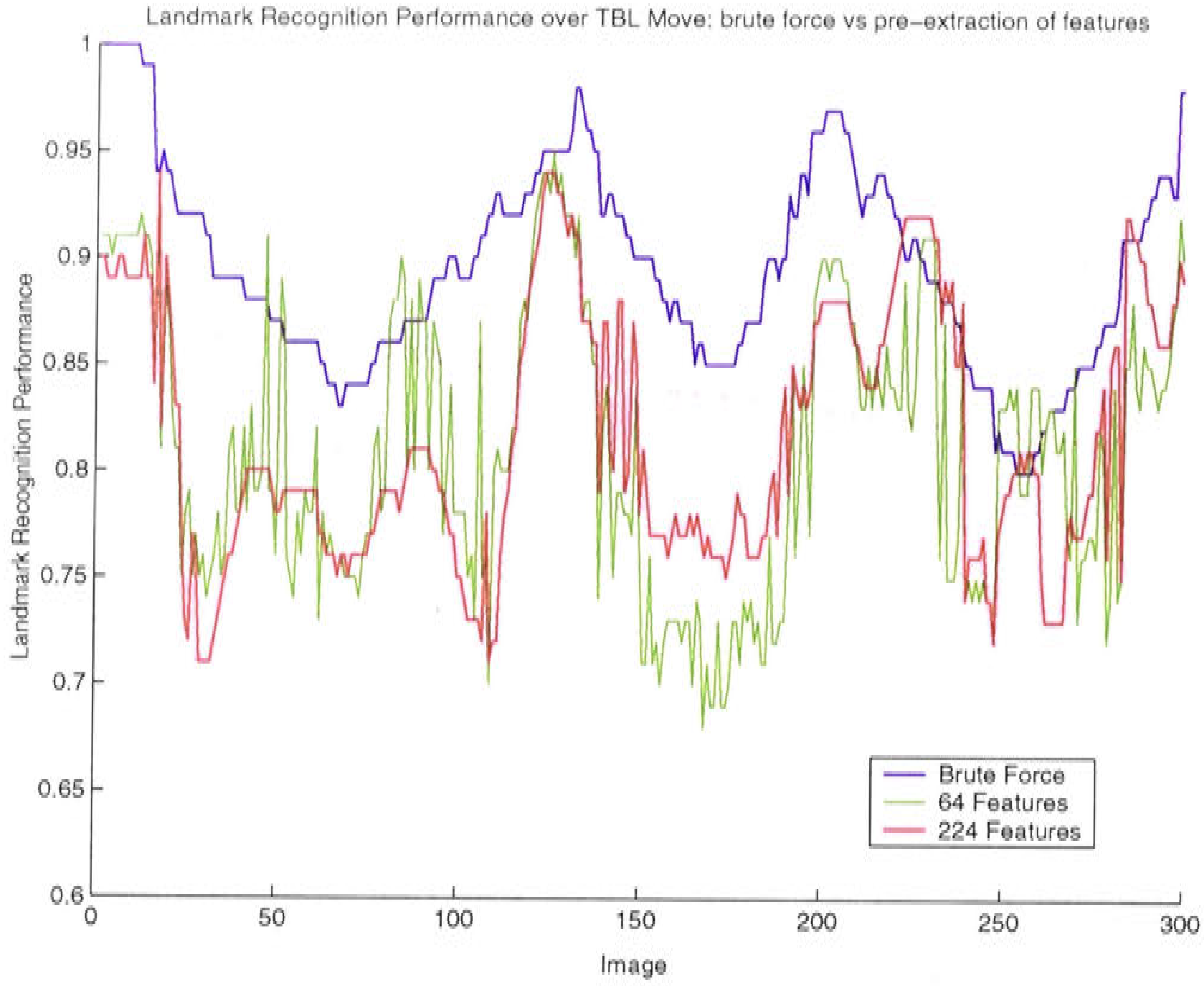


Figure 4.9: Landmark set recognition performance over a sequence of images comparing the brute force search approach versus that of extracting features before matching. The images used in this experiment were captured over a Turn Back and Look path, the high peaks in the brute force graph correspond to the robot passing over the reference position of the place.

4.4.1 Brute Force Landmark Template Matching

A brute force search involves comparing every possible region within the image to the landmark templates to identify the region which looks the most like each template. It will find the image regions which is most similar, pixel to pixel, to each of the templates. This is computationally expensive. The normalised cross correlation method described in Equation 4.3 is used to perform the template matching. This matching identifies the correlation value and the x, y position of the best match in the image for each landmark. A landmark set recognition measure is obtained by averaging the correlation values of all the landmarks. To evaluate its recognition performance and computational intensity, brute force template matching was applied to images captured during a TBL movement. A pre-learnt landmark set was matched with each of the 300 images in the sequence and the average correlation of the landmarks was

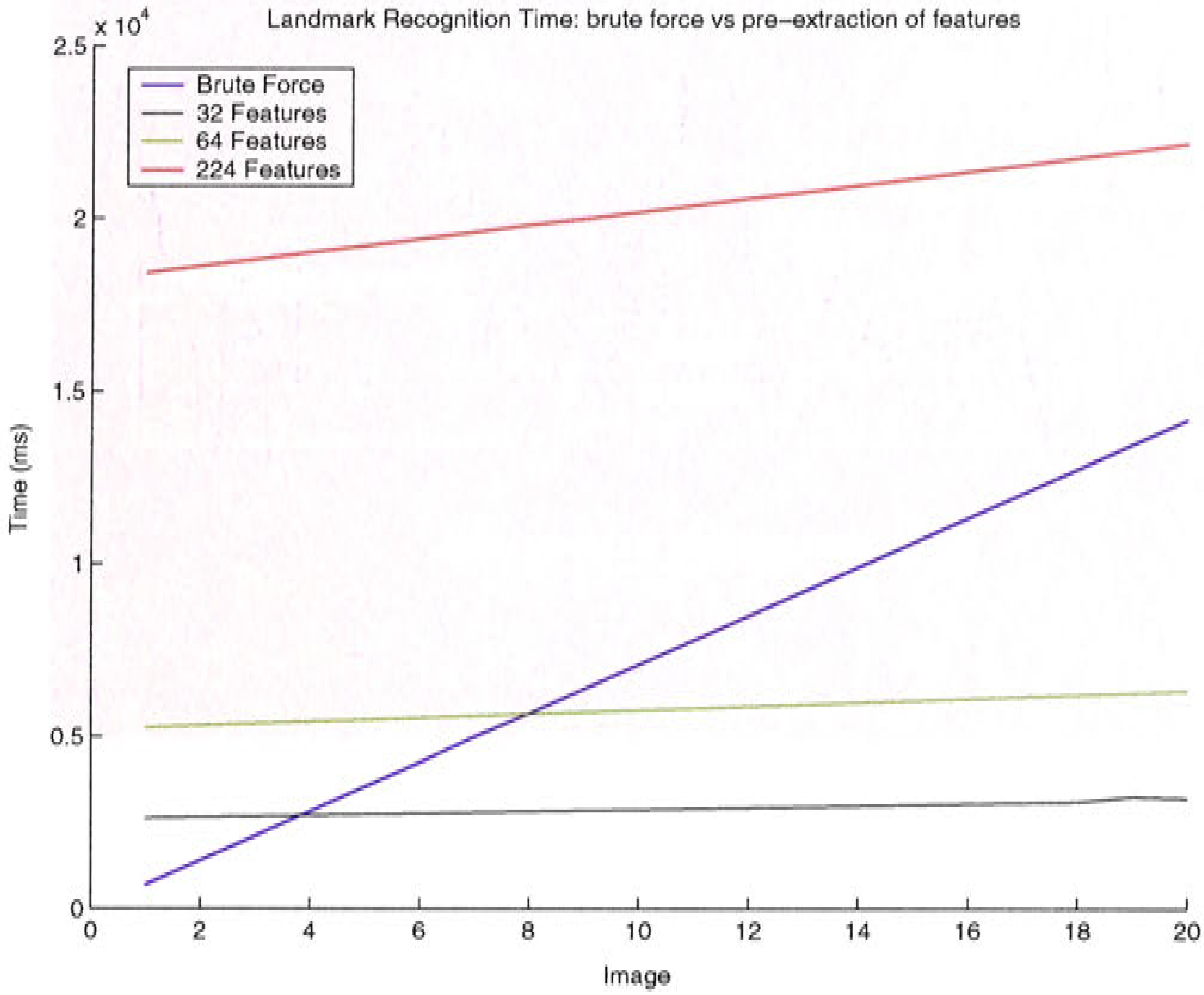


Figure 4.10: Time taken for landmark recognition comparing the brute force search approach versus that of extracting features before matching. The graph shows the time taken for each approach when matching a landmark set with a number of images. This is analogous to matching 1 image with a number of landmark sets as would happen in global localisation.

recorded. The results are shown in Figure 4.9 by the blue line. The peaks in the graph correspond to the where the images in the sequence correspond to the robot passing over the reference position of the place.

The time taken for the computation of a brute force search was measured by performing the search on 1 image with a varying number of places to search for. Figure 4.10 shows that a brute force search (blue line) takes approximately $700ms$ for each landmark set. The linear nature of this relationship means that brute force searches quickly becomes computationally unacceptable as the number of places increases.

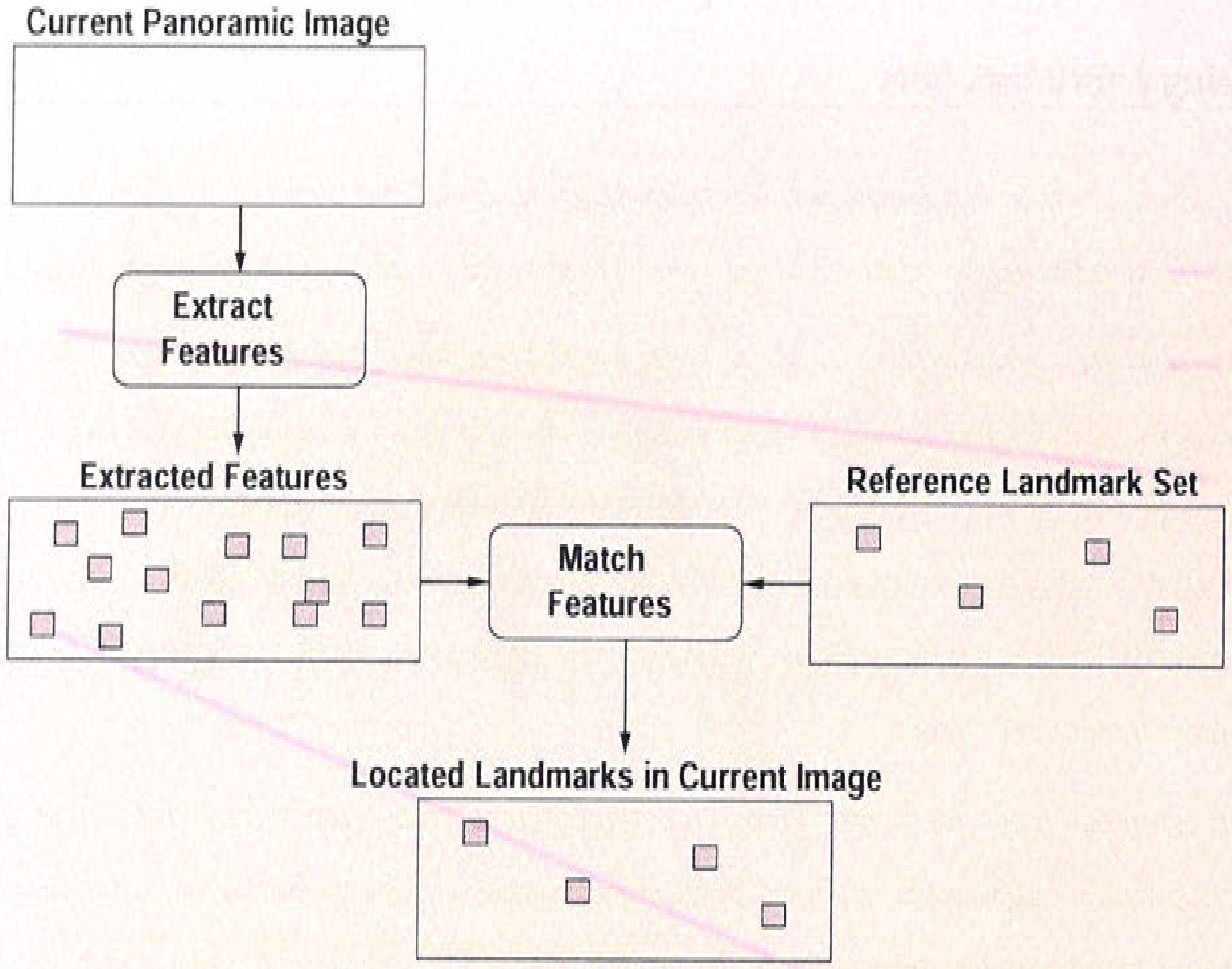


Figure 4.11: Pre-matching feature extraction

4.4.2 Pre-matching Feature Extraction

Brute force searching is computationally expensive because of the large amount of template matching required per landmark set. One way to reduce this computational load is to extract features from the visual scene and then match these features with the landmark sets. This method requires additional processing initially to extract the features, but matching time per landmark set can be drastically reduced. The concept behind pre-matching feature extraction is shown in Figure 4.11.

The features that this system uses are those obtained by applying Mori et al.'s (1995) Valley method. This is applied when selecting static landmarks and can be used again to extract likely features.

The recognition performance of matching pre-extracted features from an image with the landmark templates from a learnt landmark set is shown in Figure 4.9. Features were extracted from each image in the sequence by applying the Valley operator and selecting those features with a higher local 'uniqueness'. These extracted features were then matched against the landmark sets templates using normalised cross correlation matching as above. The features which best matched the landmark template were recorded and their average correlation gave the measure of recognition performance.

The green dashed line shows the recognition performance of matching with 64 pre-extracted features, while the maroon dashed and dotted line shows that of matching with 244 features. It can be seen that neither plot of extracted features can compare with the recognition performance of the brute force search. In addition, whereas it would be thought that extracting more features would lead to greater recognition performance, this was not the case.

Computation time for pre-extracted feature matching was measured in the same way as for the brute force approach, except that time measurements include both feature extraction and matching times. Figure 4.10 displays the results of this experiment, with the yellow dashed line representing the time taken for extracting and matching 32 features, the maroon dotted line 64 features and the green dashed and dotted line 224 features. It can be seen that although there is a substantial penalty for the initial extraction (approximately $1500ms$) the subsequent matching of landmark sets is achieved much faster than for the brute force approach. In fact even with this penalty, pre-extracted feature matching becomes the faster option when matching 5 or more landmark sets.

Unfortunately the computation saving of matching pre-extracted features is made irrelevant by its poor recognition performance. The importance of being able to recognise a learnt place far out-weights that of doing it quickly. The nature of the feature extraction used in static landmark selection must be too volatile when subjected to small changes in the visual scene. Although these landmarks are selected for their reliability under small translations, it obviously does not guarantee that the underlying feature extraction method is similarly reliable under those circumstances. The use of a different method of selecting features might solve this problem, but any new methods would have to be shown to be as recognisable and computationally inexpensive as well. In our research we have decided to persist with a brute force search and later investigate other methods for reducing landmark set recognition time by constraining the amount of landmark sets to be searched, rather than by constraining the search time itself.

4.5 Landmark Reliability Experiments

Landmark recognition experiments were carried out to determine the performance of the system under a variety of conditions described below. All experiments took place in the Electro-Technical Laboratory, Intelligent Systems Division, Tsukuba, Japan. The environment is a semi-structured corridor about 2.5 meters in width and over 15 meters in length. All experiments consisted of an initial phase of learning a place by guiding the robot to the desired place and initiating the automatic landmark selection software. A subsequent phase of guiding the robot to several positions and attempting to locate the learnt landmarks gave the system's landmark recognition performance results. The results are averaged correlation results over the landmark set, with values from 0 to 1, with 1 representing perfect correlation of all learnt landmarks at a particular place in the environment. The desired recognition performance is to achieve high correlation results over the largest possible area, while still maintaining a unique set of landmarks. By having landmarks which can be recognised over a wide area, a coarser topological map can be built, meaning the system can cut back on the amount of storage and processing time needed for global localisation. High correlation within that area can allow more accurate positioning by landmark triangulation and can lead to navigation behaviours such as homing and moving between places. Normalised cross-correlation template matching is used to minimise the intensity differences between templates and search images due to changes in environmental illumination.

4.5.1 Dynamic vs Static Landmarks

First to establish the usefulness of the TBL phase in automatic landmark selection, a comparison of Landmark Recognition Performance (LRP) for a given place was made using landmarks selected by either solely static landmark selection or by both static and dynamic selection. In the place learning phase the most reliable static landmarks are stored in addition to the best dynamic landmarks. Both were used to localise within a 2×2 meter section of the corridor environment, centered on the place that was learnt. Measurements of landmark recognition performance were taken at 20cm intervals.

The results of using static landmarks for recognition are shown in the first plot of

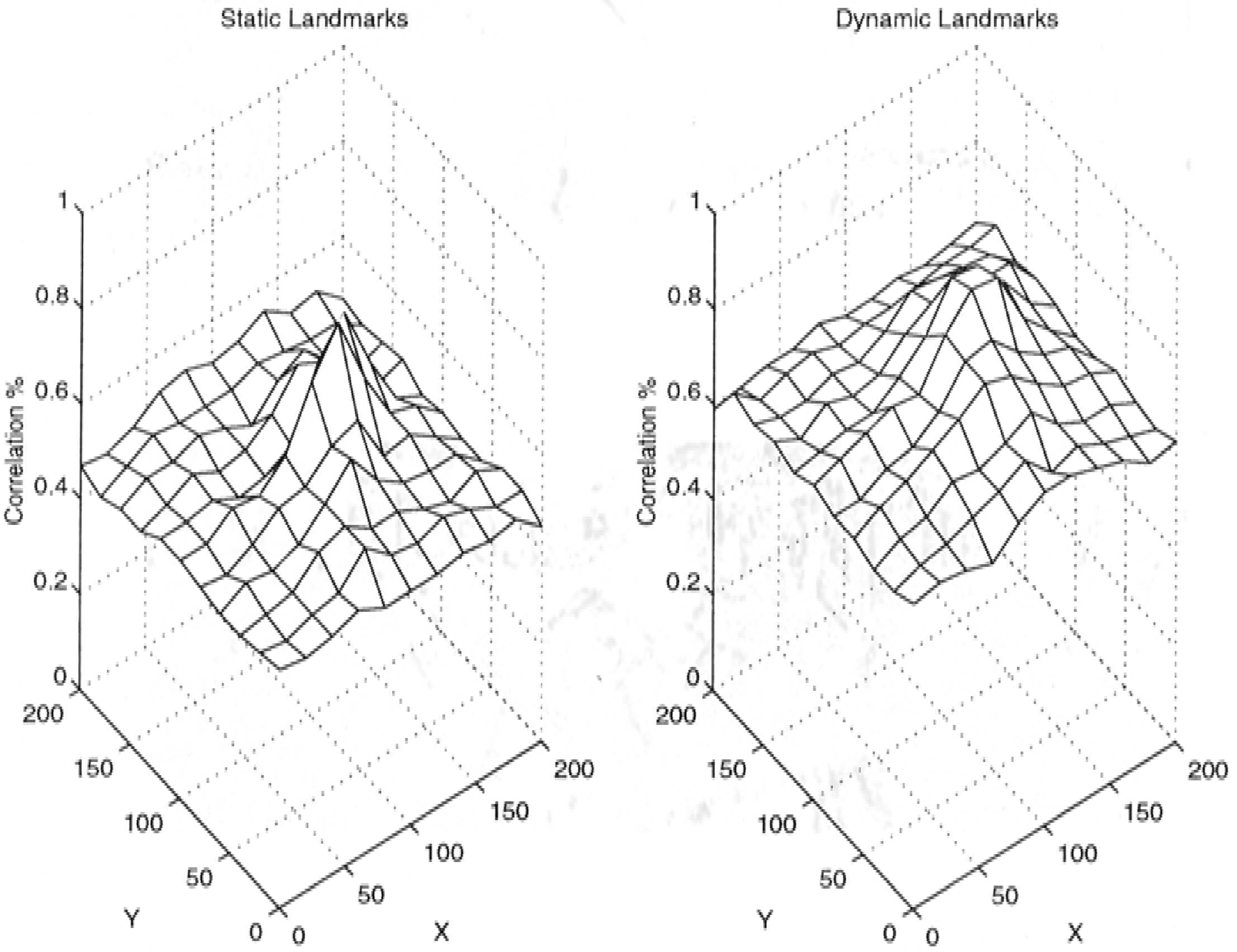


Figure 4.12: Landmark set recognition over a 2 × 2 meter area surrounding a learnt place. The graph on the left shows recognition using a set of statically selected landmarks. The graph on the right shows recognition with a landmark set that has been selected using both the static and dynamic selection phases. X and Y axes show robot position in centimetres.

Figure 4.12. The plot shows the average landmark correlation over the landmark set for various locations about the reference position. The static landmark set had 16 individual landmarks, while the dynamic landmark set had 8 landmarks. A sharp peak is evident near the center, peaking at 0.79, but falling to around 0.50, just 40cm from the peak, and maintaining this to the edges of the graph. The second plot of Figure 4.12 shows the results when using dynamic landmarks. Again there is a peak near the center (0.86) but it is not nearly so sharp and drops less rapidly. At about 60cm from the peak with correlation values around 0.70, the slope of the graph decreases further and eventually falls to approximately 0.63 at the edges of the graph.

Comparing the two graphs shows that the use of dynamic landmarks for recognition results in higher correlation measures over a greater area around the learnt place than when using static landmarks. The two different slopes observed in the dynamic graph

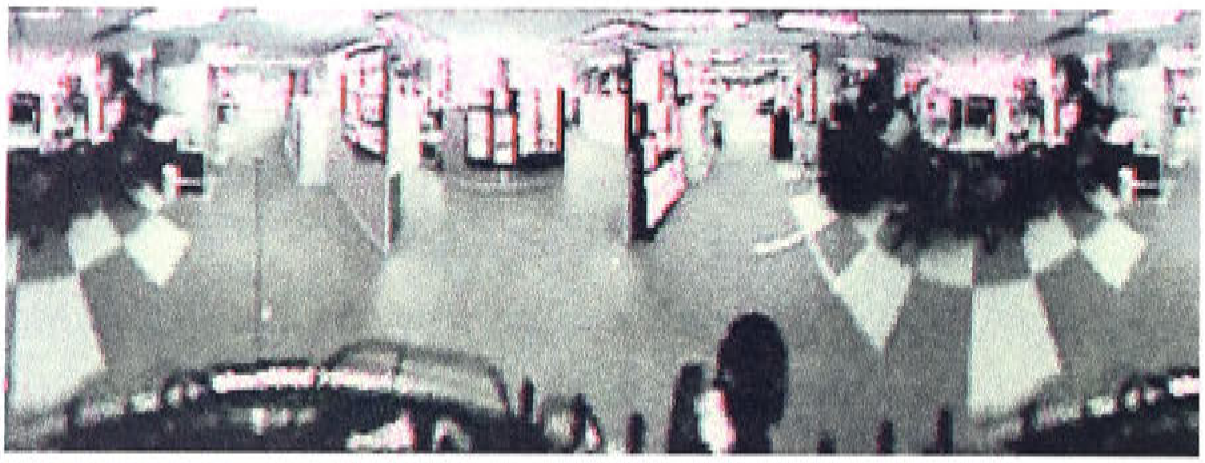


Figure 4.13: An example panoramic image from the recognition experiment captured at 15:00.

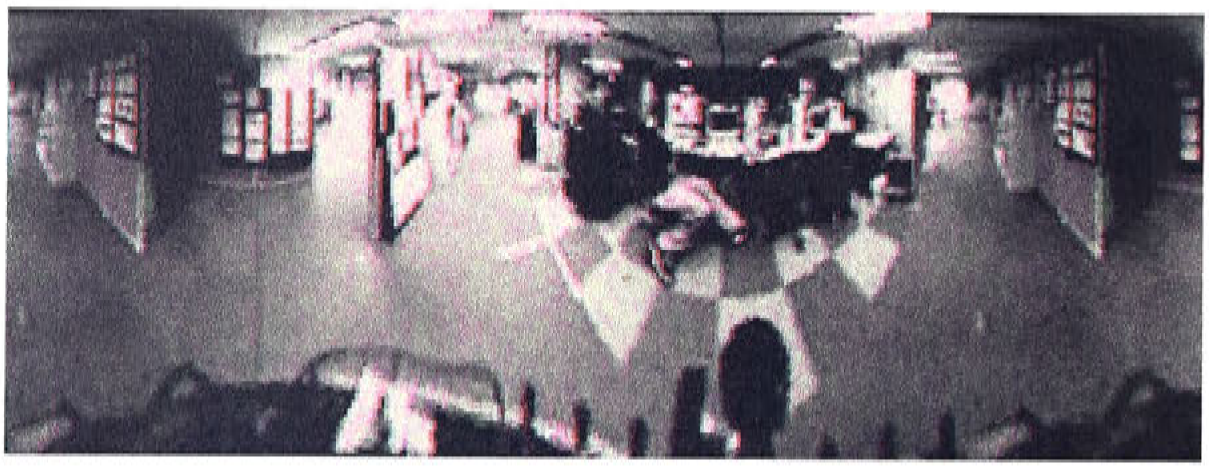


Figure 4.14: An example panoramic image from the recognition experiment captured at 20:00.

can be attributed to the higher distortion of landmarks located on the sides of the corridor (closer to robot), when compared to those at the end of the corridor (further away). From this result it can be seen that including a dynamic landmark phase in the automatic landmark selection process increases the area surrounding the reference position from which the landmark set can be recognised.

4.5.2 Robust Landmark Correlation under Changing Illumination

Next, the effect of changes of illumination on the system's performance was investigated. A place was learnt at 15:00, and the selected landmarks stored. The landmark recognition phase was carried out immediately after learning and again at 20:00 that evening, using the same set of learnt landmarks. Again the results given are for a 2×2 meter section of the corridor centered on the learnt place, and the measurements taken at 20cm intervals. Sample images from the robot during the 15:00 run and the

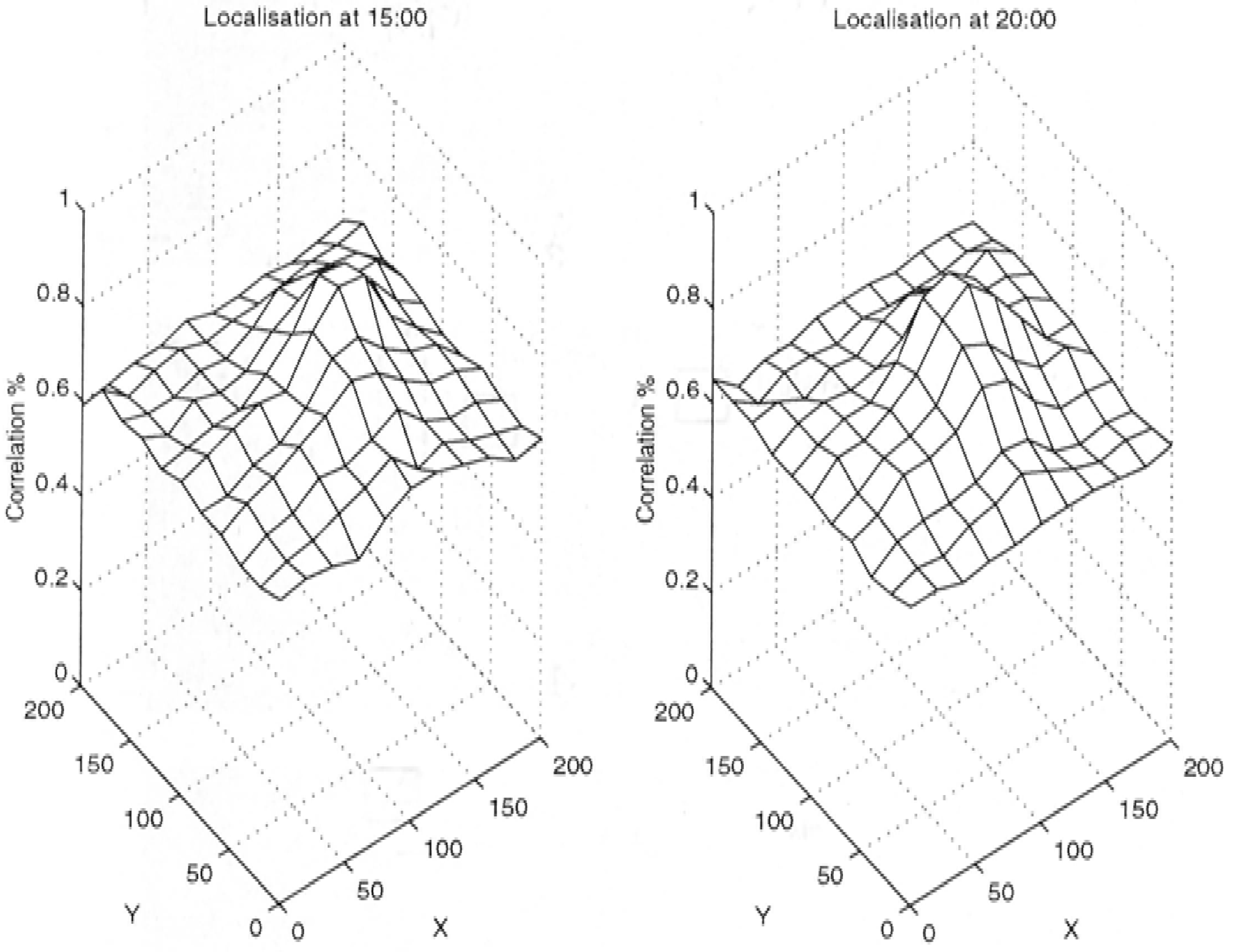
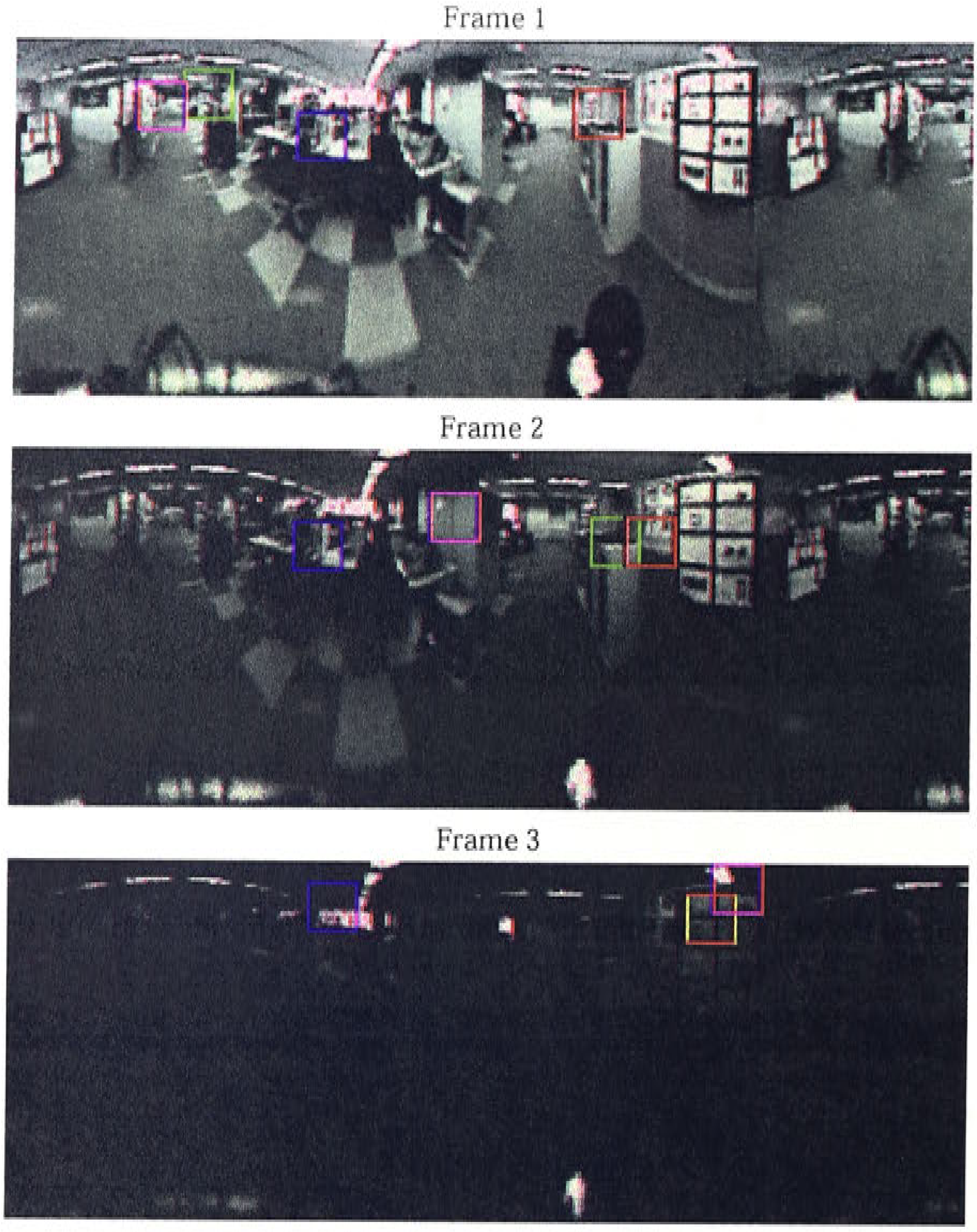


Figure 4.15: Landmark set recognition over a 2×2 meter area surrounding a learnt place in the presence of changing illumination. The graph on the left shows recognition performance from images captured at 15:00. The graph on the right shows recognition performance from images captured at 20:00. The landmark set was learnt at 15:00.

20:00 run are shown in figures 4.13 and 4.14 respectively to demonstrate the variance in illumination between trials.

Figure 4.15 shows the results from the recognition experiments conducted with differing levels of illumination. Both peak at the same position in the graph with similar values (0.86, 0.87 respectively) and both follow the same two step slope described in the previous experiment, with values of around 0.63 at the edges of the graph. This shows that landmarks can be recognised in the presence of varying illumination conditions.

To further display the benefits of using the normalised cross correlation technique when performing template matching in the presence of varying illumination conditions Figures 4.16 and 4.17 shows an example of landmark template tracking over three image frames. In Figure 4.16 the simple SAD correlation is used and landmark



Frame 1

Frame 2

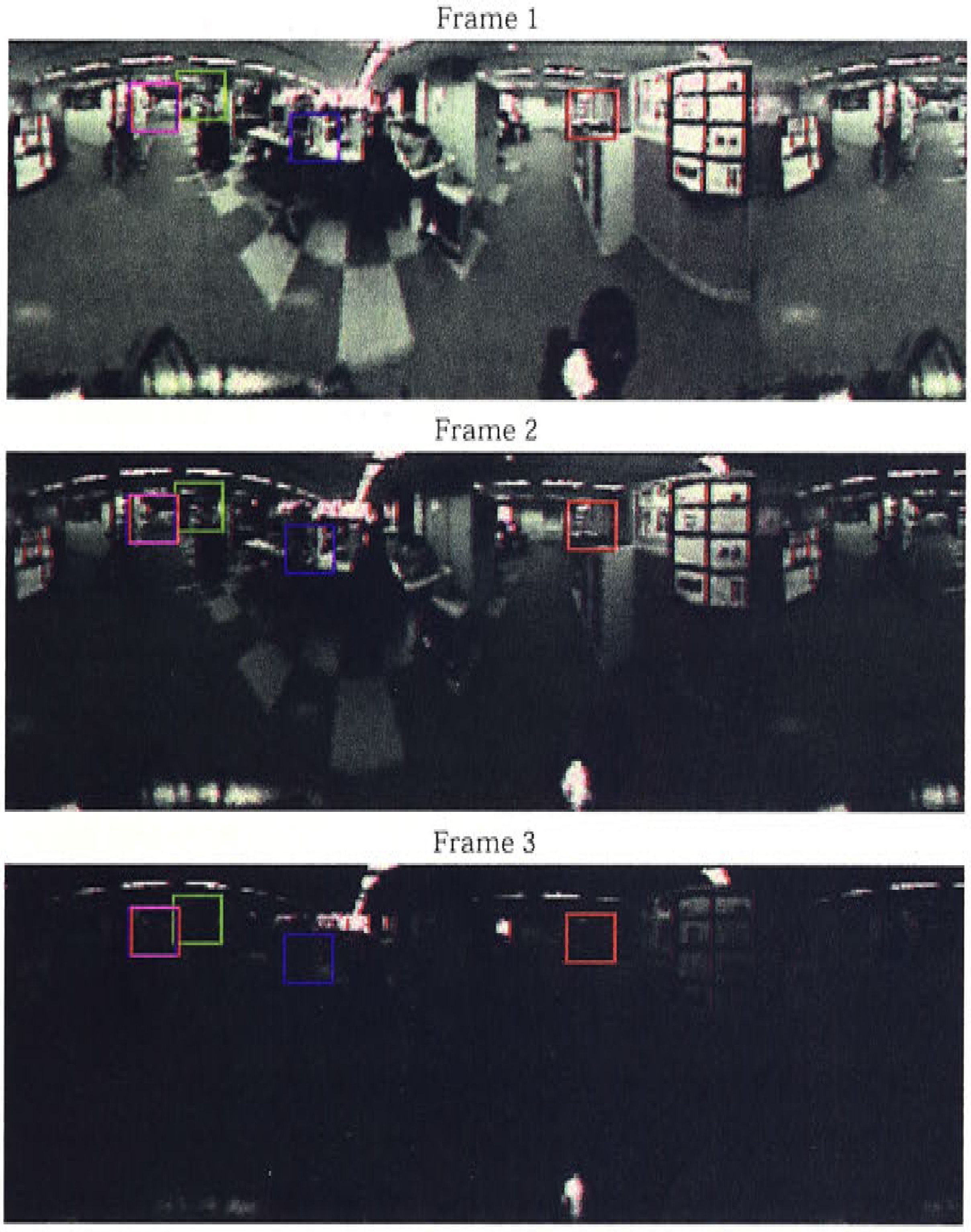
Frame 3

Figure 4.16: Non-normalised (SAD) correlation landmark tracking

tracking is lost quickly due to the rapidly diminishing illumination in the scene. The same sequence is shown in Figure 4.17, this time normalised cross correlation was used for the template matching and it can be seen that landmark tracking was successfully maintained.

4.6 Dynamic Landmark Selection with Depth Estimation

Knowledge of the depth of landmarks in the environment is valuable for robot localisation as is shown in the proliferation of range sensor based robotic solutions. Obtain-



Frame 1

Frame 2

Frame 3

Figure 4.17: Normalised correlation landmark tracking.

ing range from a monocular visual sensor is a difficult problem and becomes worse with the low resolution of the panoramic images used in this system. When looking to solve this problem, inspiration can again be taken from a biological solution. Not only has the TBL movement of wasps been interpreted as a way of selecting landmarks which are stable in the visual field, it has also been suggested that they use this flight to extract depth information about the environment. In the same way, a robot making a TBL move can use this exploration of the environment to extract depth information about potential landmarks.

While performing the TBL move and tracking landmarks, a form of bearing only Simultaneous Localisation And Mapping (SLAM) can be used to make this estimation of landmark position. SLAM is usually implemented using sensors which supply bearing and range information. In parallel with our research (Deans and Hebert, 2000) developed a panoramic bearing only SLAM system which uses the structure from motion technique of bundle adjustment to initialise a Kalman filter before the iterative estimation process can be performed.

In our research SLAM problem is reformulated into estimating the depth and angle of each landmark (as opposed to (x, y) landmark position), from a local reference position. The resulting estimates more directly reflect the uncertainty in the observations, and the local frame of reference eliminates the accumulation of uncertainty evident in SLAM using a global reference frame. Because of this reformulation, with a comparatively accurate first estimate of angle, a suitable initial approximation of depth allows for acceptable landmark position estimates without the need for batch processing before the iterative process can start. (Strelow et al., 2001) report an iterative structure from motion approach to position estimation which also uses this formulation and initialisation scheme. The fact our research estimates the position of landmarks from a known local reference point and the low resolution of the images also support this approach, with landmark depths being being virtually indistinguishable beyond a certain depth. A speed-scale drift error in landmark position estimation, however, is still conceivable.

A common problem in Kalman filter SLAM approaches is that as a map grows and more features or landmarks are obtained, the dimensionality of the filter's matrices grows accordingly and the problem can become computationally intractable. In the current system, the Kalman filter is only operative while performing the TBL movement during place acquisition and therefore the number of dimensions stays low. This limitation means however, that the system cannot continue estimation when the place is being revisited.

4.6.1 Bearing Only Simultaneous Localisation and Mapping

The current system uses a Kalman filter to estimate the state of the system in relation to reference point of the place to be learnt, with the state X defined as:

$$X = \begin{bmatrix} x_R & y_R & \theta_R & d_0 & \gamma_0 & \dots & d_n & \gamma_n \end{bmatrix}^T \tag{4.4}$$

where, (x_R, y_R, θ_R) describe the robot pose, while (d_i, γ_i) describe the depth and angle of landmark i. The filter has the traditional prediction phase:

$$X(k+1|k) = f(X(k|k), U(k+1)) \tag{4.5}$$

$$P(k+1|k) = \nabla f_x(k) P(k|k) \nabla f_x(k)^T + Q(K) \tag{4.6}$$

followed by an update phase:

$$\begin{aligned} X(k+1|k+1) = X(k+1|k) + W(k+1) \\ (z(k+1) - H(X(k+1|k))) \end{aligned} \tag{4.7}$$

$$\begin{aligned} P(k+1|k+1) = P(k+1|k) - W(k+1) \\ S(K+1)W^T(k+1) \end{aligned} \tag{4.8}$$

where U is the action performed, P is the covariance matrix, Q the process noise, W the Kalman gain, z the current observation, and H the expected observation. With

$$W(k+1) = P(k+1|k) \nabla h_x(k)^T S^{-1}(k+1) \tag{4.9}$$

$$S(k+1) = \nabla h_x(k) P(k+1|k) \nabla h_x(k)^T + R(k+1) \tag{4.10}$$

where ∇h_x is the observation Jacobian and $R(k+1)$ the noise in the observation. In the current formulation, $U(k+1)$ is the action peformed since last iteration as defined by the robot's odometric sensors, while $z(k+1)$ and $H(X(k+1|k))$ are the current and expected observation angles of landmarks in the panoramic image. Subsequently the observation Jacobian $\bigtriangledown h_x$ becomes:

$$\bigtriangledown h_x = \begin{bmatrix} \frac{\delta\phi}{\delta x_R} & \frac{\delta\phi}{\delta y_R} & \frac{\delta\phi_i}{\delta\theta_R} & \dots & \frac{\delta\phi_i}{\delta d_i} & \frac{\delta\phi_i}{\delta\gamma_i} & \dots \end{bmatrix} \tag{4.11}$$

with ϕ being the observed angle of the landmark at the present time. The filter can be initialised using the knowledge of the exact reference location of the robot, initial observations of the landmarks and associated noise (resolution of panoramic sensor), and an initial estimate of landmark depth and variance.

$$X = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & d_i^0 & \gamma_i^0 & \dots & 0 \end{bmatrix}^T \tag{4.12}$$

The iterative estimation of the system state can then proceed and continue throughout the TBL movement. The general idea of the filter is to iteratively use a sequence of observations of the angle to a particular landmark in conjunction with odometric measurements to produce successively better and better estimates of the landmark's angle and depth from the original reference position. While doing this the Kalman filter also estimates the uncertainty of the estimates by incorporating process noise in the computations. Process noise in this case includes the noise in angle measurements caused by the low resolution of the panoramic images and the error in odometric measurements due to wheel slippage.

The relationship which drives the filter is that between the observation angle of a given landmark ϕ_i from the robot's current position and the landmark's depth and and angle from the reference position, d_i and γ_i. This relationship is defined by the equation:

$$\phi_i = tan^{-1}\left(\frac{d_i sin(\gamma_i) - y_R}{d_i cos(\gamma_i) - x_R}\right) - \theta_R + \epsilon_\phi \tag{4.13}$$

where ϵ_ϕ is the noise associated with the observation.

The values of the observation Jacobian $\bigtriangledown h_x$ (Equation 4.11) then become:

$$\frac{\delta\phi_i}{\delta x_R} = \frac{d_i sin(\gamma_i) - y_R}{(d_i cos(\gamma) - x_R)^2 + (d_i sin(\gamma_i) - y_R)^2} \tag{4.14}$$

$$\frac{\delta\phi_i}{\delta x_Y} = \frac{d_i cos(\gamma_i) - x_R}{(d_i cos(\gamma) - x_R)^2 + (d_i sin(\gamma_i) - y_R)^2} \tag{4.15}$$

$$\frac{\delta\phi_i}{\delta\theta_R} = -1 \tag{4.16}$$

$$\frac{\delta\phi_i}{\delta d_i} = \frac{\left(\frac{sin(\gamma)}{d_i cos(\gamma_i) - x_R}\right) - \left(\frac{(d_i sin(\gamma_i) - y_R) cos(\gamma_i)}{(d_i cos(\gamma_i) - x_R)^2}\right)}{1 + \left(\frac{d_i din(\gamma_i) - y_R}{d_i cos(\gamma_i) - x_R}\right)^2} \tag{4.17}$$

$$\frac{\delta\phi_i}{\delta\gamma_i} = \frac{(d_i cos(\gamma_i) - x_R)(d_i cos(\gamma_i)) + (d_i sin(\gamma_i) - y_R)(d_i sin(\gamma_i))}{(d_i cos(\gamma_i) - x_R)^2 + (d_i sin(\gamma_i) - y_R)^2} \tag{4.18}$$

This method for bearing only SLAM was implemented and experimentally verified, first in simulation and then on a real world mobile robotics system.

4.6.2 Simulation Results

A simulation of the bearing only SLAM system described above was implemented in Matlab to validate its ability to accurately estimate landmark position. The simulation was to estimate the position of eight landmarks over a TBL movement involving 400 noisy landmark observations.

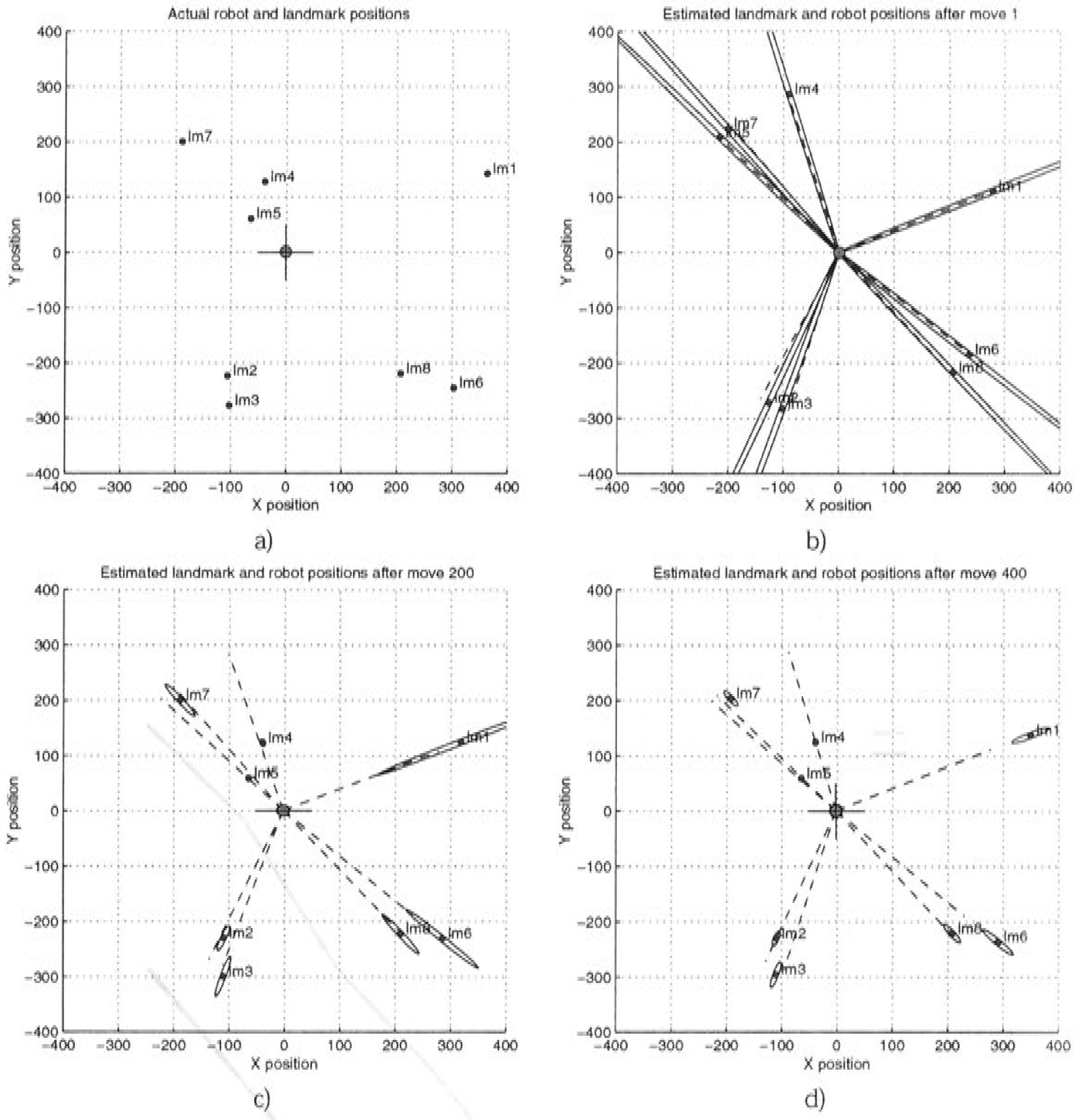
Observations had an error of ±1 degree, while the odometry measures where within 10% of the actual values.

Figure 4.18 part a) shows an example of the setup of the simulated system. Landmarks are spread randomly about the reference position of the robot which is indicated by the filled circle at the center of the plot. The robot then moves in the cross shaped TBL pattern, which is shown by the solid line about the reference position. During the movement landmark observations are calculated geometrically and noise is added. The noisy observations are then incorporated into the filter.

The results of landmark position estimation for this configuration of landmarks are shown in parts b), c) and d) of Figure 4.18. Plots of the estimated state are shown at the beginning (b), middle (c), and end (d) of the TBL movement. In each graph the position of the robot is again shown as the filled circle, the completed section of the robot TBL movement by the solid line, the landmarks estimated positions are shown by the labelled black points, the ellipses display the uncertainty of the estimation and the dotted lines the current observations.

In graph b) the landmarks' depth and angles are at their initial estimates as much the uncertainty associated with their depth is still very large. The system is initialised with variances of 300 in the depth estimate and 1 degree in the angle estimate. As the simulation progresses, and the filter encounters more observations from different locations the uncertainty and position of the landmarks can be seen to change (as seen in sub figures b) and c)). It is easy to see from the plots that the landmarks positions were all estimated quite well, with the estimates of landmarks closer to the robot being more accurate to those further away. This is due to the fact that for landmarks closer to the reference position, the movement along the TBL path causes greater change in the observed angle, than that of landmarks further away. Another thing to notice is that movements that are orthogonal to the direction of the landmark have much greater affect on landmarks' position and uncertainty estimates than that of parallel movements. The estimate in Landmark 1 in plot c) which lies in the direction of the movement along the x axis can be seen to still be quite uncertain, while the other landmark estimates have improved quite noticeably.

The accuracy of the system's landmark position estimation was tested by comparing the actual and estimated position of landmarks at various depths from the reference position such as those shown in Figure 4.19. The system estimated landmark depth over a TBL move as in the previous experiment, while measuring the error between the actual and estimated landmark positions. As the ability of the filter to converge on the correct landmark depth depends on the initialisation of the depth estimate, this was done for four different initial landmark depth estimates of 100, 300, 500 and 700. The units used for the measuring of distance is not important to the accuracy measure. What matters is that the distance depth estimates can be accurately achieved in relation to the size of the TBL movement. In this case the TBL movement extending to a maximum of 50 units from the reference position along all major axes. All results



a)

b)

c)

d)

Figure 4.18: Simulation results of landmark depth and variance estimation over the Turn Back And Look (TBL) movement. Part a) shows the actual positions of landmarks in the simulation; part b) the state of the system at the beginning of the TBL move, depth estimate and variance are at the initial values; b) completed X-axis phase of TBL movement; c) completed TBL movement.

were averaged over 10 trials. The uncertainty measures for landmark depth in the covariance matrix were initialised to 700 for all trials.

The results of this experiment can be seen in Figure 4.20. This graph shows the actual depths of landmarks plotted against the estimation error. As expected the accuracy of estimates for landmarks with short depths is much better than those of long

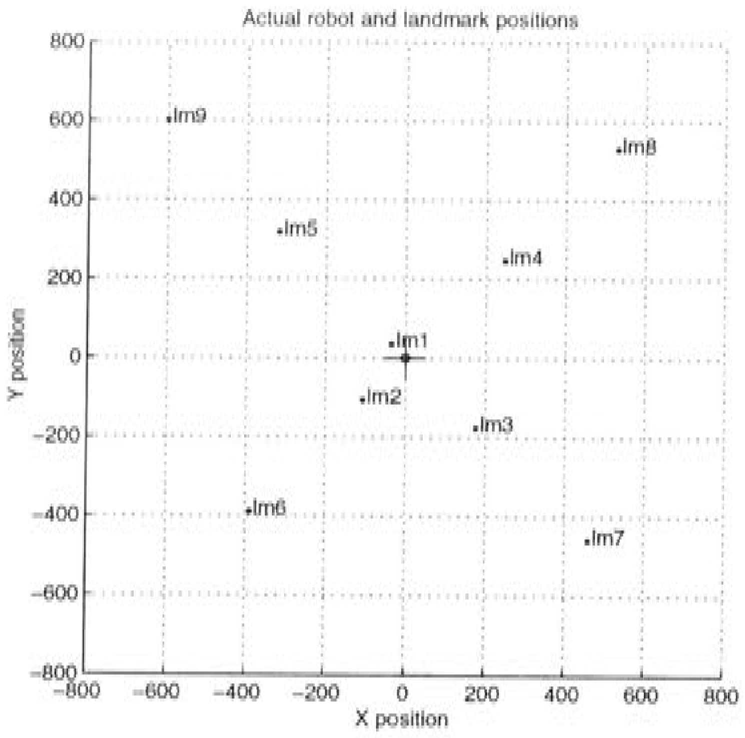


Figure 4.19: Actual position of landmarks and the TBL movement used to evaluate depth estimation accuracy.

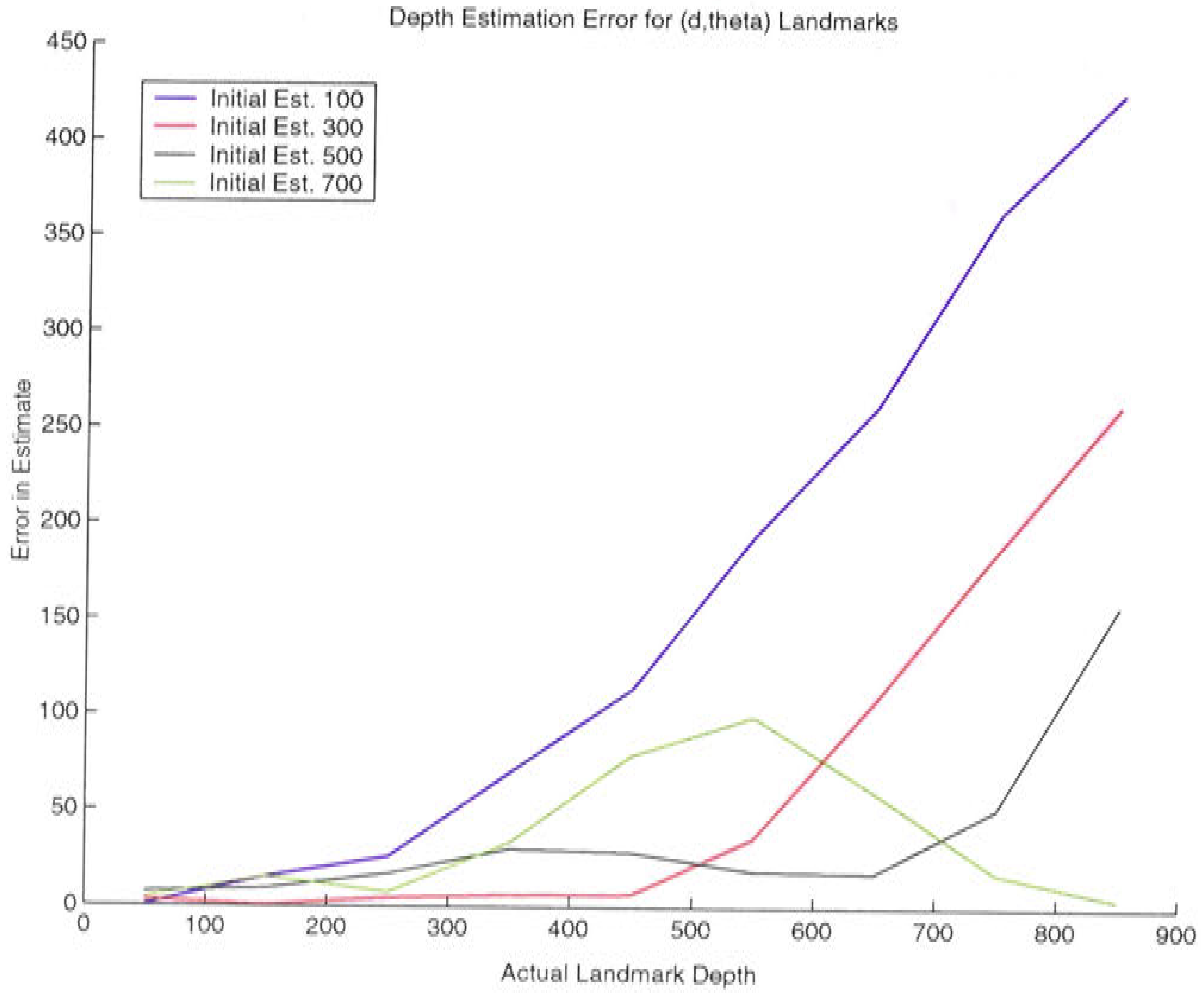


Figure 4.20: Error in depth estimation of landmarks for various initial depths.

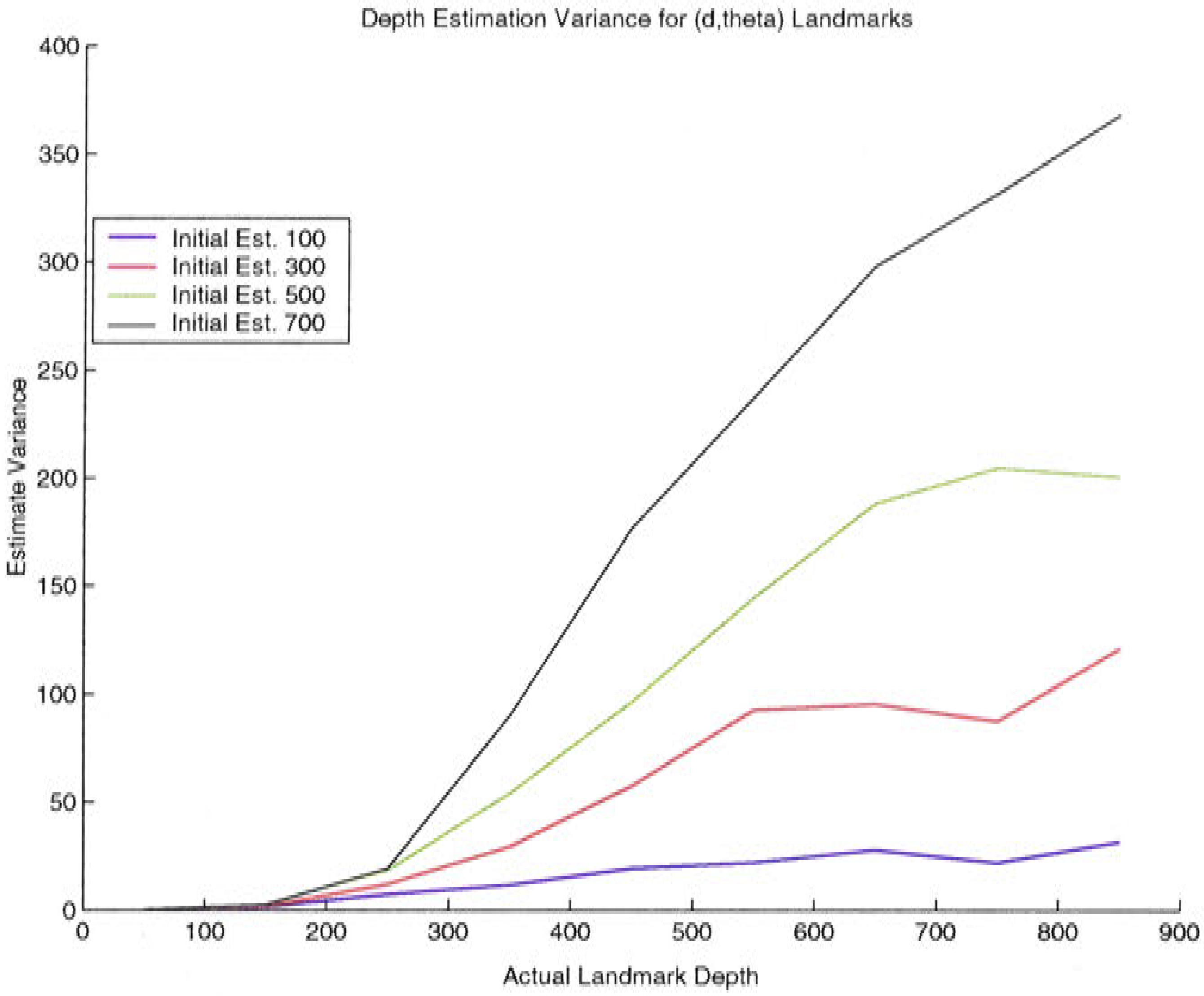


Figure 4.21: Variance of depth estimation of landmarks for various initial landmark depths.

depths. The importance of choosing a suitable initial depth value can also be seen. Landmark estimation using an initial depth estimate of 100 can converge accurately for short landmark depths but are bad for medium and long depths. Initial estimates of 300 or 500 seem to perform quite well for short and medium depths, but the error grows rapidly afterwards. The results for an initial estimate of 700 are surprising, the graph showing accurate estimates for landmarks with a short depth, the error growing through medium depth landmarks and then contracting again for longer landmarks.

As these estimates have associated uncertainty regions, the results must be interpreted in conjunction with the size of theses uncertainties. Figure 4.21 shows the corresponding variance measures for each of the plots in Figure 4.20. Again, for landmarks with a short depth the variances are very low, showing that no matter what the initial depth estimate is the filter can converge on the correct estimate with high certainty.

The plot of the variance associated with an initial depth of 100 shows that the variance does not grow in conjunction with the error measure. This is probably because the filter makes a wrong estimate early in the iteration cycle and because of the depth of the landmark and small amount of information that can be gained from changes in observation, it can not recover. Whatever, the reason, this mismatch between error and variance argues against initialising the filter with short depth estimates. At the other end, initialising long landmarks with a depth of 700 gives an accurate estimate with very high uncertainty. The accuracy in depth estimation of distant landmarks is mainly luck, due to the initial depth being close to actual depth. With landmarks at this range there is not enough information to be gained from observations to either move the estimate from its initial value or to decrease the uncertainty in that estimate, but this is to be expected due to relatively high uncertainty in the observation input. The variance associated with the moderate initialisation values of depth grow along with the error which is a desirable characteristic. If these estimates are going to be used it is necessary to know how much trust to place in them.

Figure 4.21 gives the indication that all the variance plots are starting to level off after landmarks pass a depth of about 600. This again reinforces the limit of the filter to gain information from landmarks located it depths greater than this. Interpreting these two graphs in conjunction it would seem it best to use an initialisation depth of about 3-5 times the extent of the observation movement from the reference position. Also it seems that there is little information to be gained from landmarks greater than 6-7 times this distance.

It is interesting to note that for estimation accuracy it is desirable to spread landmarks throughout the possible range of angles. Landmark configurations where they are bunched around a single angle, such as that shown in Figure 4.22, provided significantly worse depth estimates. This is not because the landmarks contribute to each others depth estimation directly, but rather because the Kalman filter is achieving mapping and localisation simultaneously as the name SLAM implies. Not only is the filter estimating landmark positions, it is also estimating the position of the robot from which the observations are being taken. When landmarks are bunched together there is less information available to correct the robot position, and errors in the odometry can accumulate. Subsequently, observations from an uncertain position cannot provide as much information on landmark location as those from a certain position.

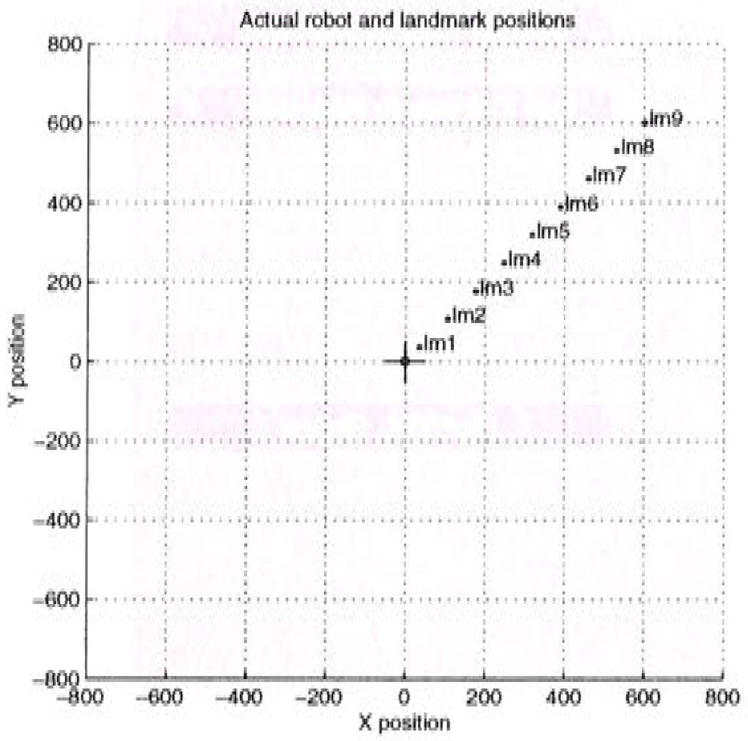


Figure 4.22: An example of a configuration of landmarks which results in poor depth estimates.

Landmark position estimation with a small amount of landmarks suffers from the same fate.

Simulation results from the system show the filters ability to converge to the correct result, although for landmarks at greater depths this process can be slow. The accuracy of the estimate depends on both the depth of the landmark in relation to the size of the TBL movement and the resolution of the sensor which measures the angle to the landmarks.

Of course the uncertainty region is not exactly modelled as it should really be sort of a parallelogram bounded by the angle uncertainty. Modelling uncertainty with ellipses, however is much easier to implement and seems acceptable to the system needs.

4.6.3 Real World Landmark Depth Estimation

The method described above for estimating landmark depth while performing the TBL movement was implemented on the Nomad XR4000 mobile robot. The computation cycle for depth estimation is shown in Figure 4.23. First the filter is initialised using an assumed depth and covariance and the landmark angle observations from the static landmark selection phase. All landmark depths are set to 2500cm as an initial estimate with a variance of 2500cm. The variance of the landmark angle is deter-

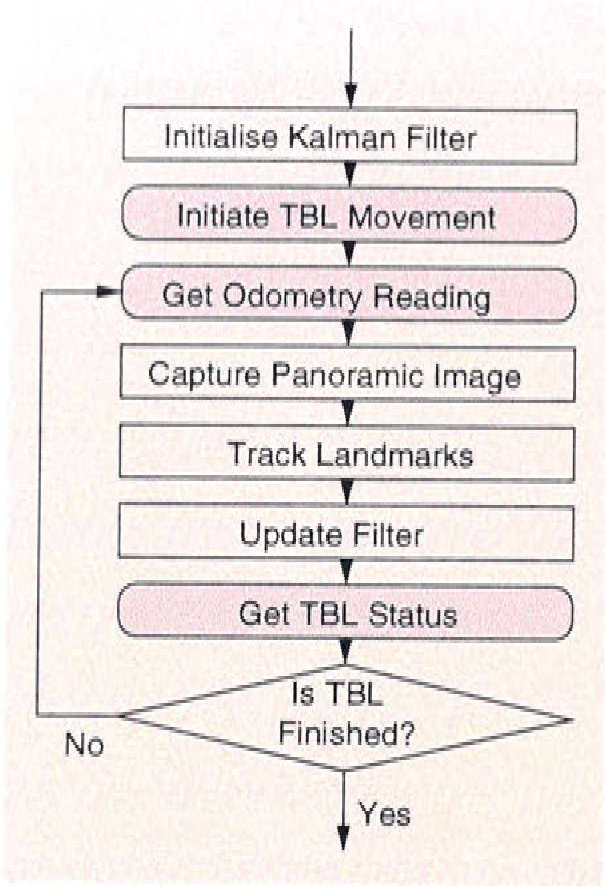


Figure 4.23: Computation cycle on vision processing CPU for estimating landmark depth during the TBL movement.

mined by the resolution of the panoramic sensor which is approximately 0.9444 pixels per degree. The program then sends a signal to the robot controller to initiate the TBL movement and the the iterative estimation of the landmarks' positions begin. The TBL movement extends $50cm$ from the reference position and the robot moves at $20cm/s$.

The current odometry is read from the robot controller and a panoramic image is captured. Landmarks are located in each image frame using the tracking algorithm described in Table 4.1. The odometry, the observed radial angles to the landmarks in the landmark set and their associated measures of noise are used as input to update the Kalman filter as described in the preceding section.

This estimation process is repeated until the robot controller reports that the TBL movement is complete. When this occurs, the estimated position of the landmarks can be read from the state vector of the filter.

The cycle of capturing the panoramic image, tracking the 32 landmarks and estimating the depth takes on average $135ms$ (Pentium II 750). With the panoramic image capture and unwarping taking $6.5ms$, landmark tracking $25 - 75ms$, and the depth estimate $7ms$, the tracking time varying with the size of the search window.

The main difficulty in implementing this algorithm in the real world is of course the noise associated with real world sensing. The noise introduced into the system with each sensor reading must be estimated in order for the filter to correctly incorporate the observation information into the system state. For example consider the situation where during landmark tracking a particular landmark is occluded and tracking is lost, resulting in incorrect landmark angle information being input into the filter. By associating an appropriately high level of noise with the incorrect angle measurements, the filter can place less weight on such observations and rely more on earlier, less noisy observations. The two sources of noise in this system are the sensing of landmark angle from the panoramic images and the odometry readings from the robot controller.

Landmark Angle Observation

The observation of the radial angle of landmarks is achieved by matching landmark templates with the current panoramic image. The landmark tracking algorithm used in this system was presented in Table 4.1. The location of the landmark along the x axis of the image can easily be converted in to an angle measurement. The noise assigned to these measurements is related to the value of the $landmark\_match$ as defined in Table 4.1, which represents the correlation match between the landmark template and region of the current image where the landmark was located. This value is assigned a noise level according to the following formula:

$$\text{if} \quad val < 0.5 \quad \text{then} \quad val = 0.5,$$

$$\epsilon = ((1 - val)\pi + \pi/180)^2 \tag{4.19}$$

where ϵ is the noise variance associated with the observation and val the landmark matching value. The value of val is limited at 0.5 because this is the correlation level for matching with a random background. The first term of the right hand side of the equation converts $1 - val$ into a proportion of π radians. This is then added to the error inherent in the resolution of panoramic sensor (approximately 1 degree). The result is then squared to produce the noise variance. Thus the variance approaches

$(\pi/2)^2$ when landmark correlation is poor and $(\pi/180)^2$ when landmark correlation is perfect.

Although the above method for weighting the importance of angle observations does take into consideration errors in tracking where tracking performance degrades gracefully, it does not fully insure against the problem of data association. The data association problem is that of mapping sensor data with real world objects. In this case it can be a problem if a region in visual field that is not a representation of the actual landmark is significantly similar to that of the landmark template. This situation could cause the system to produce an incorrect landmark angle observation which has a high matching value and subsequently a low noise estimate.

Data association is a common problem in estimation approaches such as the Kalman filter, and no direct attempt to address it is made in this system. However, the chances of the problem in the first place occurring, and secondly affecting the outcome are reduced for the following reasons:

- Landmarks are visual templates and are initially chosen for their locally unique appearance in the static selection process. Visual templates are much less likely to fall victim to the data association problem than features such as corner regions as each template can be distinct. The static selection process further limits the chance that image regions of similar appearance will be located in the immediate vicinity of the landmark.
- Landmark tracking begins on the known landmark location so initialisation is not a problem.
- If an instance of the data association problem does occur there is likely to be some degradation in tracking performance due to the nature of the landmarks (visual templates). Although this degradation may not be sufficient for the system to correctly estimate the noise associated with the observation, part of the dynamic selection process is to choose the landmarks which track best over the TBL movement. Therefore landmarks which suffer from a data association problem are less likely to be chosen in the final landmark set than those which do not.

These reasons do not entirely eliminate the data association problem, however it substantially alleviates its affect on landmark position estimation in this system.

Odometry Readings

The reading of odometry measurements from the wheel encoders are another source of noise in the landmark depth estimation process. Odometry error due to wheel slippage is not likely to be a great factor in this case due to the limited range of the TBL movement. A greater source of noise is the delay between obtaining a odometry reading and capturing the image. Although these two events happen in succession (Figure 4.23) the odometry reading must travel from the computer hosting the robot controller to the computer hosting the vision processing and so time delays between readings can occur. This system contains no time stamping of sensor readings but relies on noise estimates to allow for measurements to be combined accurately. Counter acting this delay is the fact the robot moves slowly ($20cm/s$) and the TBL movement is comprised of constant velocity movements.

The time taken for an odometry request to be sent and answered is $\sim 40ms$. Assuming that the communication overhead on sending and receiving the odometry request are equal this results, on average, in a $\sim 20ms$ delay between capturing the odometry information and the call to initiate capture of the panoramic image.

Odometry noise has been set to 0.01 for translation measurements under translation and 0.10 for translation measurements under rotation. Although this noise level may seem low given the delay between the odometry reading and image capture the following results seem to validate these levels. It is unknown whether tighter coupling of these measurements would have an affect of landmark position estimation in light of the greater amounts of noise being introduced by the landmark angle observations.

4.6.4 Artificial Landmark Results

Since visual landmarks can be composed of visual representations of objects from different depths, it can be difficult to determine a ground truth of selected landmarks, and therefore difficult to validate the depth estimation procedure. Manually selected artificial landmarks, on the other hand, can be constructed to provide this validation.

Artificial landmarks were constructed in an otherwise visually sparse corridor environment as shown in Figure 4.24. The entirety of each landmark lay on a single planar surface (corridor wall) and as such the ground truth of landmark position could be

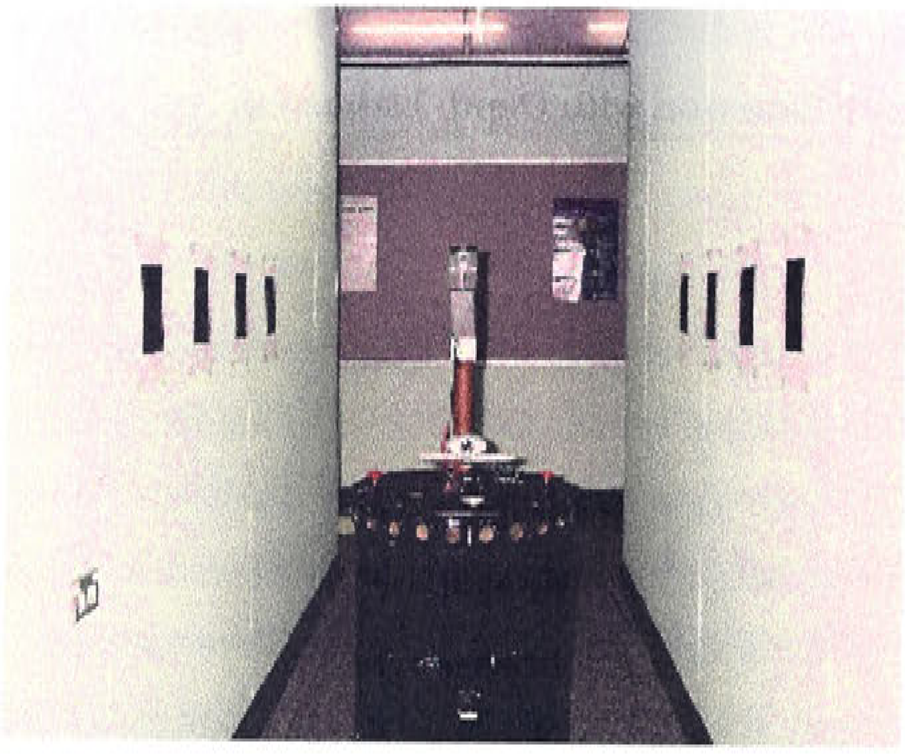


Figure 4.24: Artificial landmarks in corridor environment.

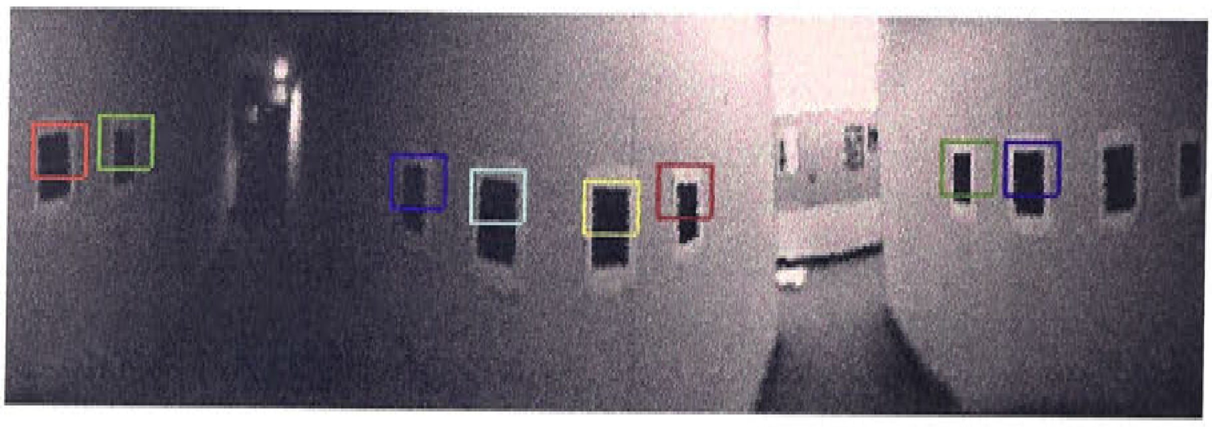


Figure 4.25: Artificial Landmarks

measured.

Landmarks lay on the parallel surfaces of the corridor wall, equidistantly spaced at a height of $140cm$. The corridor was $140cm$ wide and the robot was positioned approximately in the center of the corridor and surrounding landmarks. To ensure landmark positions were measurable, landmarks were manually identified in the reference panoramic image. The landmarks chosen for this experiment are shown in Figure 4.25.

Once the landmarks were identified the robot performed the TBL movement, capturing images and performing the iterative landmark position estimation. Because of the confined space in the corridor the TBL movement was shortened across the x axis.

Obviously the tracking task in this experiment is somewhat simplified with such visually distinct landmarks, leading to more reliable input to the depth estimation filter.

This advantage was counteracted by the proximity of the landmarks to the panoramic sensor, which caused a high level of distortion of landmarks due to the translation of the panoramic sensor while the robot performed the TBL movement.

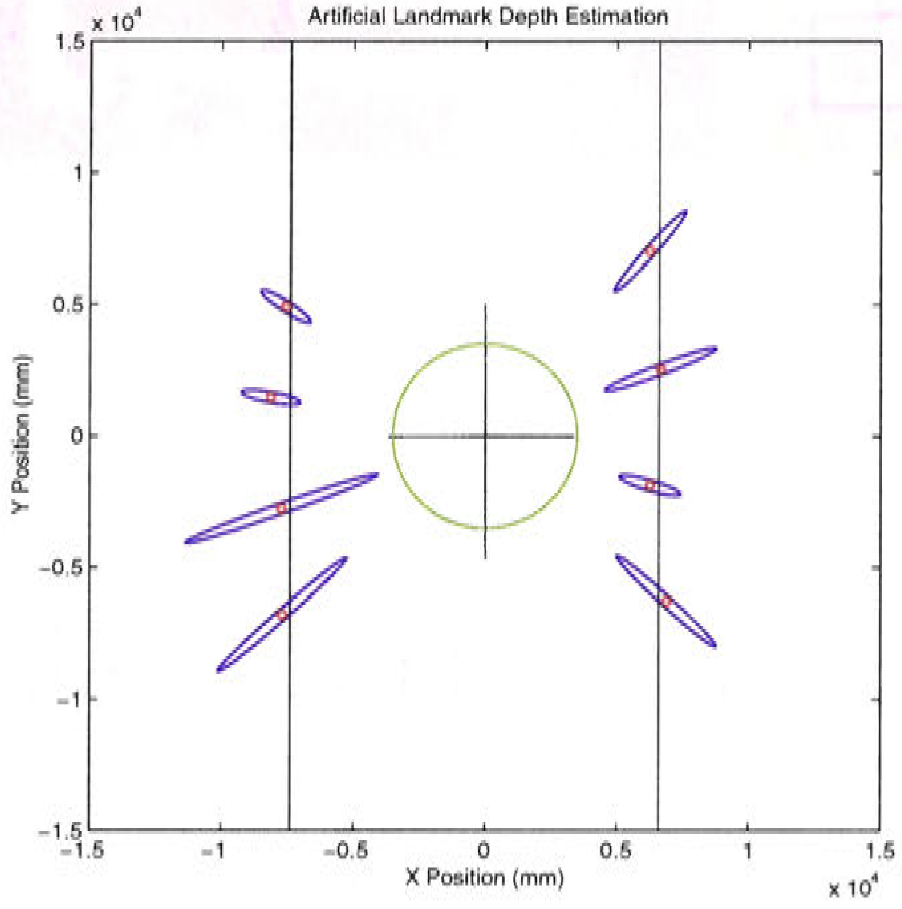


Figure 4.26: Estimated landmark depth and variance from artificial landmarks

Figure 4.26 shows the results of estimating landmark depth using the artificial landmarks. The estimated landmark positions are shown by the small circles and the associated variances by the encircling ellipses. The two parallel lines show the walls of the corridor while the large circle shows the initial position of the mobile robot. The path the robot travelled while performing the TBL movement is shown by the cross shaped lines at the center of the robot. It can be seen that the estimates correspond closely to the walls of the corridor, all estimates are within $\pm 10cm$ of the actual position of the landmarks. A video of this experiment is included in the CD-ROM which accompanies this thesis.

4.6.5 Real World Results

The landmark depth estimation filter was applied to real world data. The context and dimensions of the room in which the experiments were conducted is shown if Figure 4.27. Figure 4.28 shows the results of one such experiment. The initial robot posi-

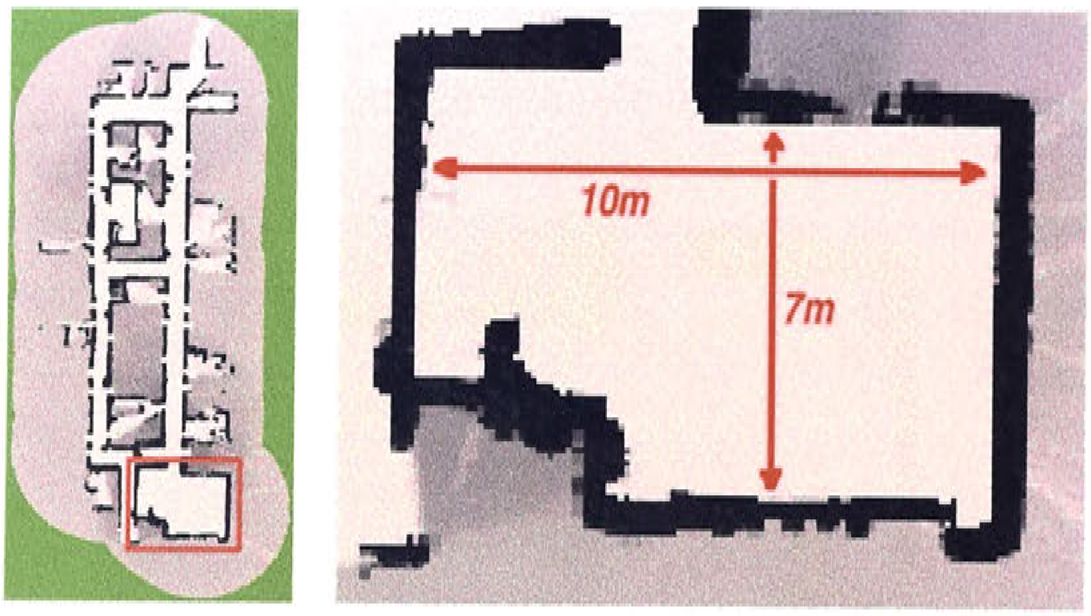
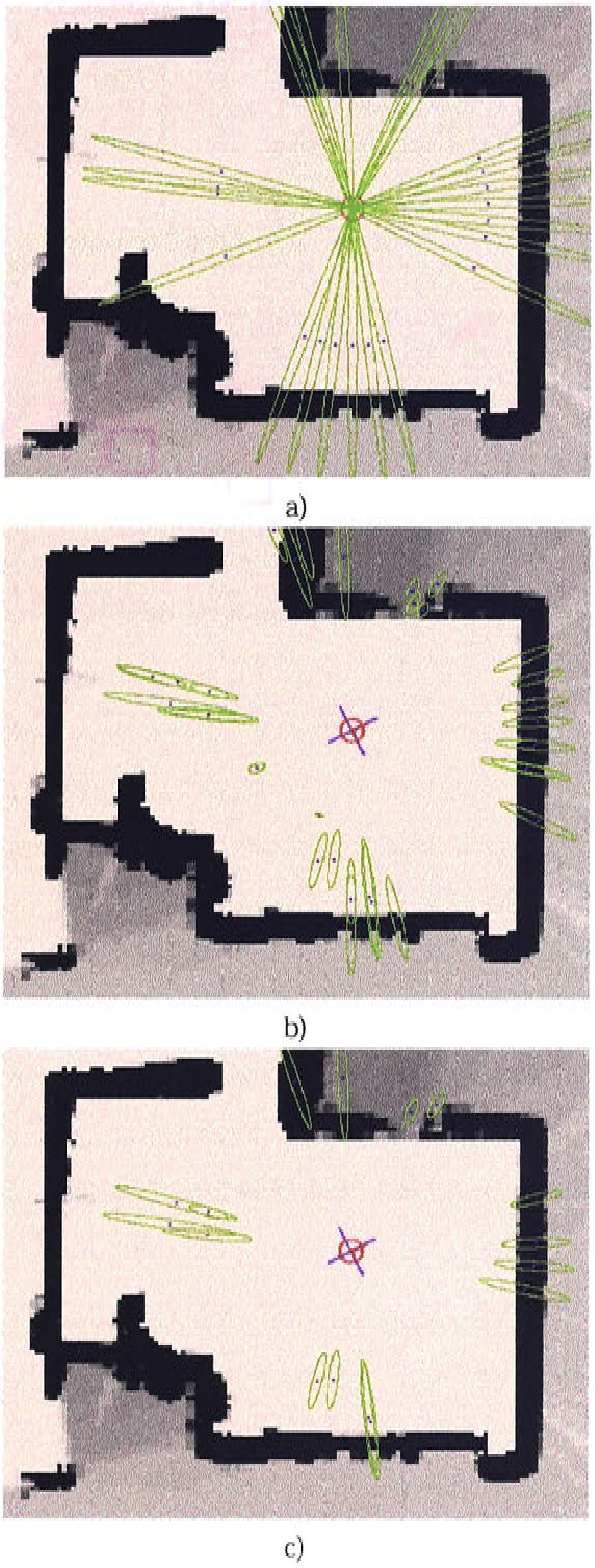


Figure 4.27: Location of real world landmark acquisition experiments.

tion from which the landmarks where selected is shown by the red circle in the center of the room and the landmarks positions and variance by the surrounding squares and ellipses. The line within the red circle represents zero degrees (x axis in robot coordinate system) from the reference position. The TBL movement extended $500mm$ from this reference position, moving along the robot's x and y axes. Initial landmark depth estimates were set at $2500mm$. Approximately 1000 frames were processed throughout the TBL movement. The room is approximately 7×10 meters in area. Sub-figure a) shows the initialisation of the filter before the start of the TBL movement; b) shows the landmark depth estimation after the completion of the TBL movement; and c) shows the landmarks which were then selected to form the reference landmark set to represent the place based on their landmark tracking performance throughout the TBL movement. Manual inspection showed the landmarks were estimated to within roughly $\pm 500mm$, with the exception of those landmarks in the doorway which were within $1000mm$.

The selected static landmarks that were used in this experiment are those shown in Figure 4.29. With the nature of visual landmarks being such that they can contain objects from different depths, the estimation process is much noisier than with a simulated system, and a measurement of ground truth can be equally hard to obtain. The results here provide a good estimate of landmark depth as well as a variance measurement in accordance to the noisy nature of the landmarks used. For example the landmarks shown in the doorway have templates which include pixels from both the door frame and the corridor behind with the depth estimate lying in between and



a)

b)

c)

Figure 4.28: Real world landmark depth estimation experiment 1. The place's reference position is shown by the red circle, landmark position estimates and their variances are displayed as blue boxes and green ellipses. The TBL motion is shown in blue. The figures show landmark depth estimates a) before TBL move (initialised to 2500mm), b) after TBL move. Landmarks selected to represent the place because of their reliable tracking performance are shown in c).

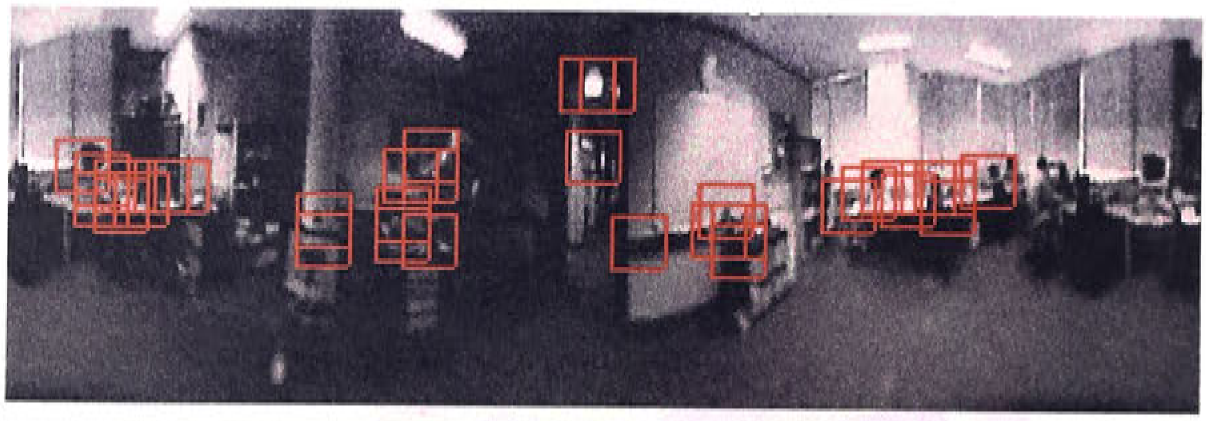


Figure 4.29: Static landmarks used in real world depth estimation experiment 1.

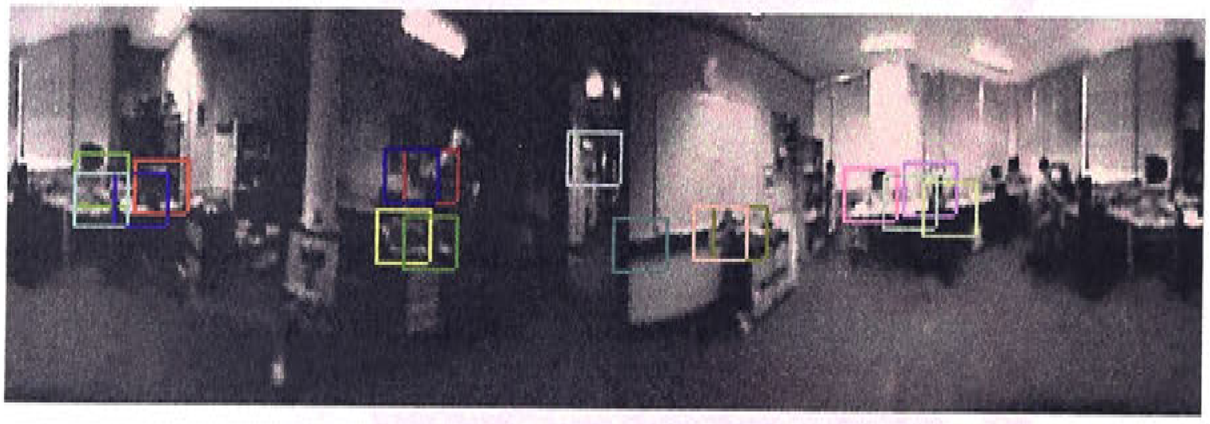
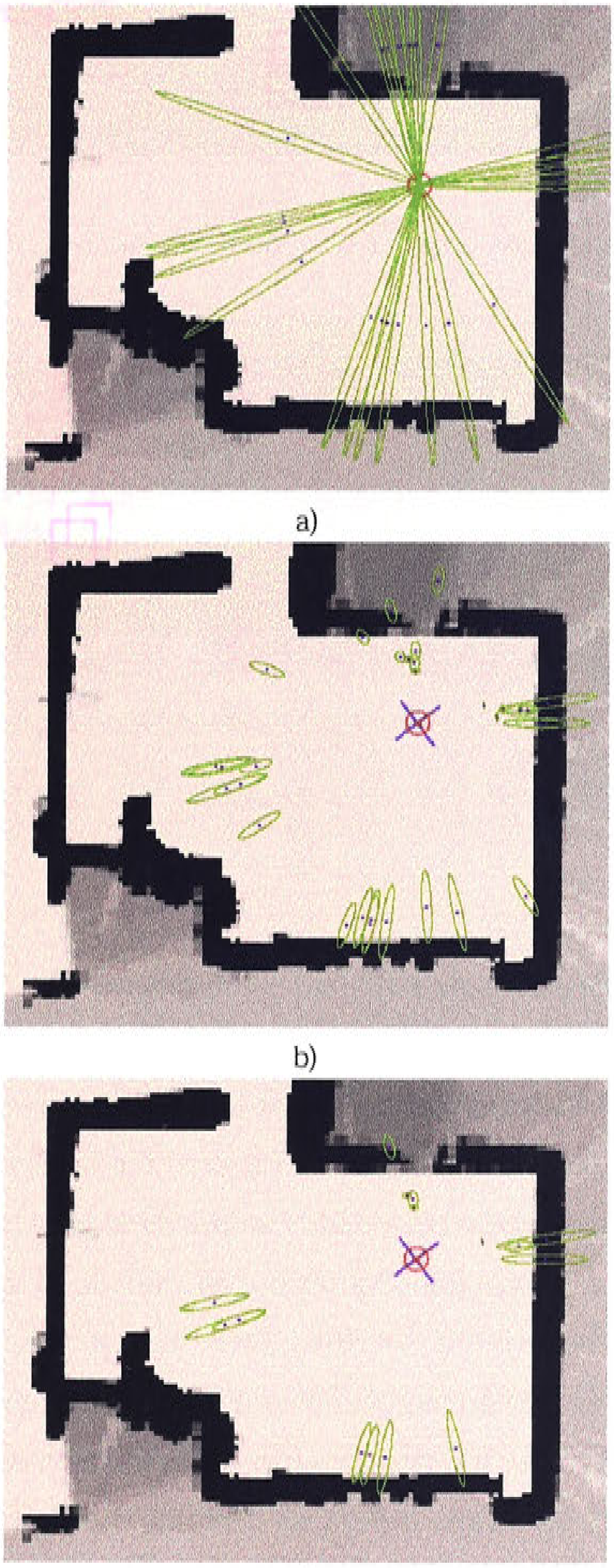


Figure 4.30: Reference landmarks selected after TBL move in real world depth estimation experiment 1.

the variance covering both. The reference landmarks which were subsequently selected because of their reliable tracking performance throughout the TBL movement are shown in Figure 4.30.

Figure 4.31 shows the results of another real world landmark depth estimation experiment, this time with the reference place being closer to the walls of the room. Again sub-figure a) shows the initialisation of the filter before the start of the TBL movement; b) shows the landmark depth estimation after the completion of the TBL movement; and c) shows the landmarks which were then selected to form the reference landmark set to represent the place based on their landmark tracking performance throughout the TBL movement. The results show the system can estimate landmark depths at a range of depths. The static landmarks used in this experiment are shown in Figure 4.32 and the chosen reference landmarks in Figure 4.33.

Figure 4.34 shows the results of real world landmark depth estimation in the presence of occlusion. In this experiment the system selected and estimated landmark positions from the same reference position used in the experiment displayed in Figure 4.32. The



a)

b)

c)

Figure 4.31: Real world landmark depth estimation experiment 2. The place's reference position is shown by the red circle, landmark position estimates and their variances are displayed as blue boxes and green ellipses. The TBL motion is shown in blue. The figures show landmark depth estimates a) before TBL move (initialised to 2500mm), b) after TBL move. Landmarks selected to represent the place because of their reliable tracking performance are shown in c).

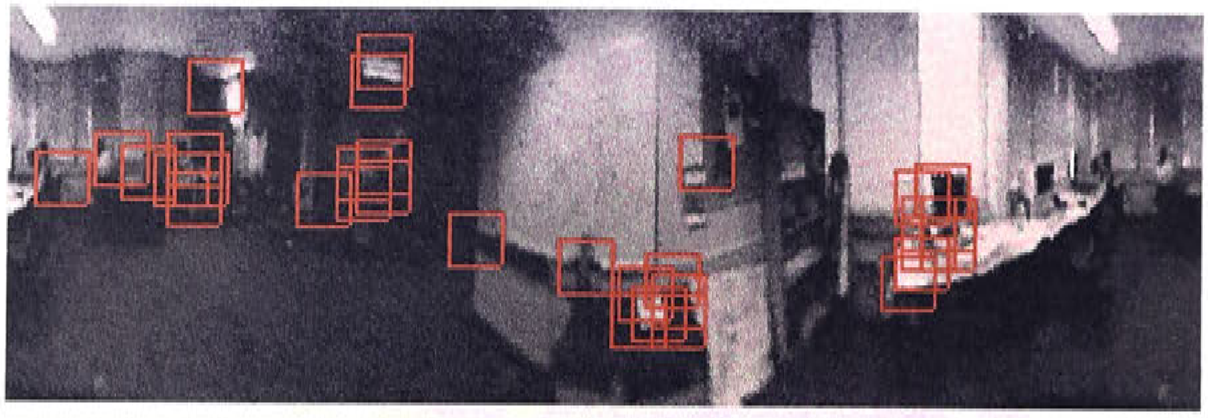


Figure 4.32: Static landmarks used in real world depth estimation experiment 2.

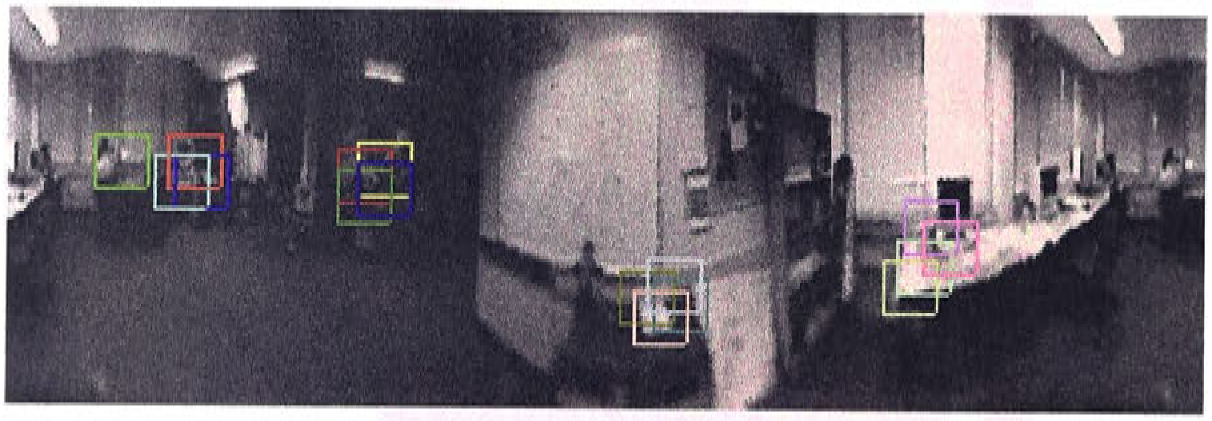
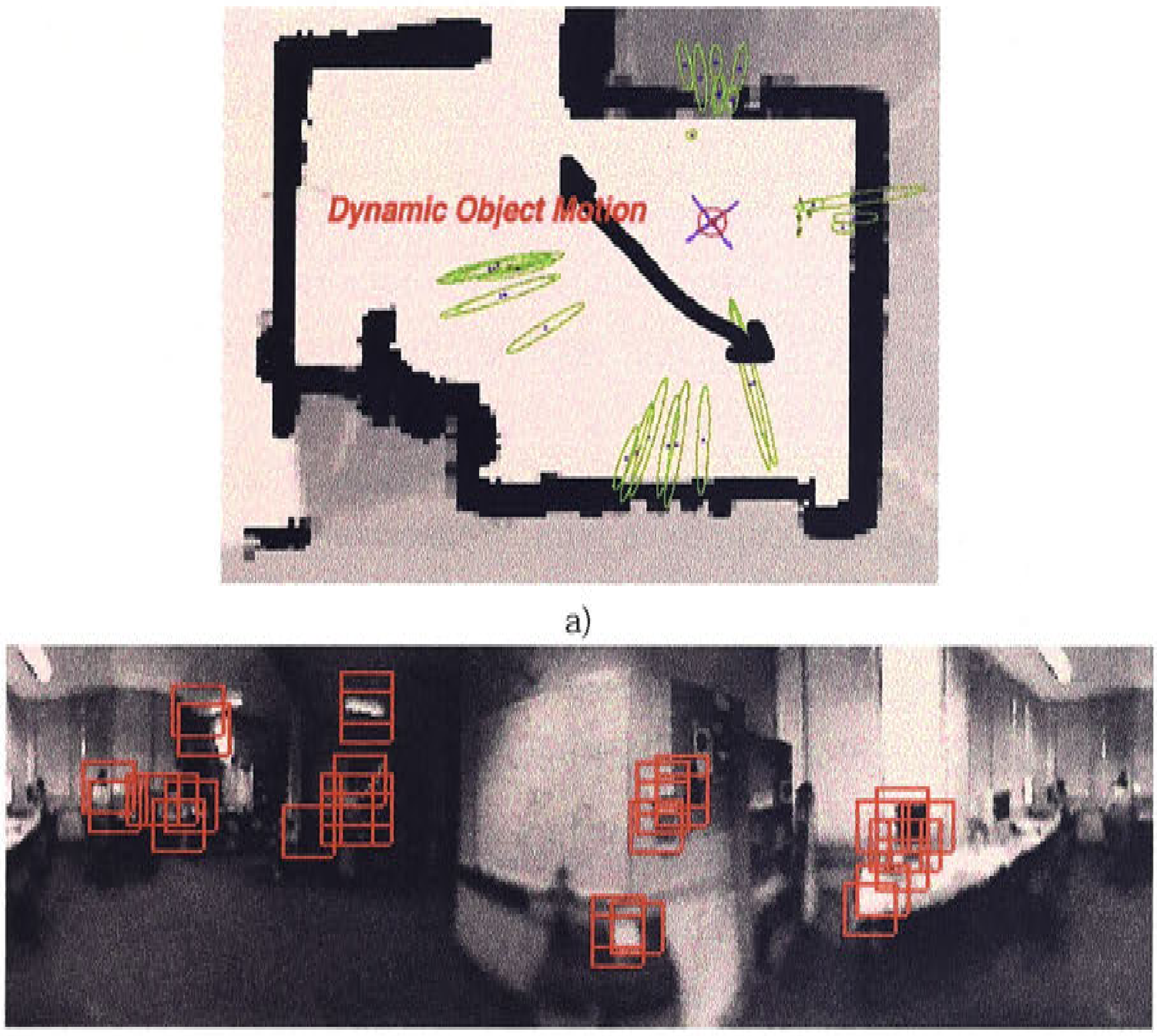


Figure 4.33: Reference landmarks selected after the TBL movement in real world depth estimation experiment 2.

reference position and the estimated landmark positions are shown in part a), while the static landmarks used in this experiment are shown in part b). During the TBL movement a person walked at normal pace back and forth along the path displayed by the thick black line. The path of the person caused the landmarks in the bottom right of the figure to be repeatedly occluded from the panoramic vision sensor. This in turn caused a repeated loss in tracking for the affected landmarks and the introduction of high levels of noise into the Kalman filter. The results show that although the landmarks were not estimated as accurately as before, the system still provides a consistent estimate and through manual confirmation it was observed that the true position of landmarks still lay within the ellipse denoting the estimate uncertainty.

This result in addition to the results presented above involving estimation of automatically selected landmarks demonstrate the systems ability to provide a usable estimate of landmark depth. This estimate can subsequently be used as additional information to the problem of mobile robot localisation which will be addressed in Chapter 8.



a)

b)

Figure 4.34: Estimated landmark depth and variance in the presence of occlusion. Part a) shows the position of estimated landmarks and the path of the dynamic object responsible for occlusion in the visual field during the TBL movement. Part b) shows the static landmark set used in this experiment.

4.7 Summary

This chapter has a presented a low-level representation for spatial knowledge based on visual landmarks. Implementation details of automatic landmark selection and recognition were reported. This chapter has not addressed the localisation system of the proposed system, rather solely concentrating on the acquisition of visual landmark sets as a place representation. The use of this level of representation for localisation is left to Chapter 8.

This chapter presented the following main ideas were developed:

- *Automatic Visual Landmark Selection:* Visual landmarks are distinct and can provide a rich source of information to the localisation task. The task of selecting landmarks to form a representation about a reference position must be automated.

- *Static Landmark Selection:* static landmarks were selected on the basis of their local uniqueness. The valley method can identify potential visual landmarks which are strongly recognisable.
- *Dynamic Landmark Selection:* landmarks identified by the static landmark selection process are evaluated for their reliability throughout a Turn Back and Look movement. During this movement landmarks are tracked using normalised cross correlation template matching.
- *Landmark Set Recognition:* Brute force template matching can be used to recognise sets of landmarks in panoramic images, although this process is computationally expensive. Experiments demonstrated that the dynamic selection phase improves landmark recognition performance.
- *Landmark Depth Estimation:* In addition to ensuring the selection of reliable landmarks, the TBL movement can also be used to estimate the depth of landmarks from the reference position. A form of bearing only SLAM was presented which could accomplish. Results were presented from simulated landmark depth estimation, artificial landmark estimation and real world experiments. These results showed that a depth estimate and its uncertainty can be estimated during the TBL movement.

A landmark set representation of a particular place in the environment is desirable because of the benefits such a representation will have on the task of localisation. Distinct visual landmarks should aid the recognition process as landmark templates will be unique to sensory views captured from near the reference position. Knowledge of the angle and depth of landmarks in relation to the reference position will contribute to the accuracy of local positioning by providing additional information to triangulation calculations. The validity of the ideas developed above, should therefore be discussed in relation to the localisation task and assessed by weather they contribute to or distract from the successful implementation of that task.

The automatic selection of visual landmarks, while providing a workable representation of landmark sets is far from ideal. The current implementation is limited as to what it defines as a "good" landmark. The initial process of feature extraction using the valley method is computationally expensive and the system would be benefited enormously by the use of a feature extraction method which could maintain landmark recognisability while becoming more computationally efficient.

The dynamic selection phase could also be improved by a more intelligent selection landmark criteria. Because an estimate of landmark depth is being made concurrently with the dynamic evaluation procedure, this information could contribute to the choice of landmarks. Landmarks that are close to the reference position would provide more information to the local positioning task, whereas landmarks at greater depths would be recognisable from a greater area surrounding the reference position. A study of these competing needs is needed to determine the desirable depth of landmarks. Also the combination of depth and reliability measures of landmarks in order to rank potential candidates is also an interesting question.

The biggest problem in this implementation of landmarks as a representation for spatial knowledge occurs in the landmark set recognition procedure. As detailed above the brute force recognition process is extremely computationally expensive which limits the efficiency of the global localisation task. As more places are added to a topological map, more of the brute force searches will be required to perform global localisation and the required computation time will increase linearly. This, ironically highlights the need for a multi level spatial representation in mobile robot navigation, with the higher levels constraining the global localisation search in the lower levels. However the computation needed for a brute force search for just one landmark set already pushes the real time constraints of the problem. This problem will only be solved when a computationally inexpensive pre-matching feature extraction method is implemented which can maintain the recognition performance of landmark sets.

The depth estimation process is the strong point of the landmark representation implementation and can only add to the localisation performance of depth ignorant systems. A more accurate depth measure could be obtained by improving the resolution in the panoramic image and by a more rigorous approach to enforcing real time controls on sensor measurement and data fusion.

To end this chapter the proposed landmark representation should provide benefits to the localisation task although the expense of the landmark recognition process may inhibit the applicability of this particular implementation. This problem, however, only highlights the need for a multi-level approach to spatial knowledge representation and these ideas can be verified in this system irrespective of the failure of this level of representation to meet the real time constraints of the task domain.

Chapter 5

Mid Level Representation: Local Space Profiles

This chapter details the implementation of the mid level of spatial representation for mobile robot localisation introduced in Chapter 3 based on the local space profiles concept. Information about the location of a robot can be derived from the amount of open space surrounding the robot. For example, the knowledge that there is open space to the front and to the back and no open space to both sides can inform the robot that it is in a corridor environment. Subsequent investigation into the robot's position, can then be directed with respect to this knowledge.

A local space profile contains information about the extent of local space for a particular position in the environment. A view of the extent of open space surrounding a mobile robot will not be unique when compared to other views as can be imagined in a corridor environment. Thus a representation of local space is not meant to provide a unique solution to the mobile robot localisation task, rather to simply constrain the search to save computation in low-level landmark localisation. Thus the representation of local space should be minimalistic and support fast comparisons between different viewpoints.

The outline of this chapter is as follows. Implementation details of the mid level of spatial representation are given. Section 5.1 describes the process of detecting the extent of local space surrounding a position in the environment from the panoramic visual sensor. First methods of detecting ground planes in panoramic images are discussed and a probabilistic approach to combining different techniques is introduced. Local space detection results from image sequences are then presented, and a comparison to ground truth is made to evaluate carpet matching techniques. Section 5.2 presents a representation for local space profiles which is compact and allows for fast rotation invariant matching. Section 5.3 discusses categorising local space profiles to constrain the localisation search. Finally, Section 5.4 provides a summary and discussion of the implementation issues involved in forming a local space representation.

5.1 Local Space Detection

In order to use knowledge of local open space to aid in localisation, the robot must have a sensing method. The panoramic visual sensor seems to be ideal for this task providing rich visual information from the surrounding environment. However, it is monocular vision and any attempt to recover a sense of local space must detect the depth of visible objects. As shown in Chapter 4 the extraction of the depth of points in a monocular panoramic scene can be achieved, however it requires time and computation therefore it is not applicable to evaluation of the entire visual scene. This extraction of depth information is not a trivial problem and could command a research project in its own right. The problem is the same as that of detecting obstacles in obstacle avoidance for local navigation which needs to be solved for any mobile robot system to be successful. Acknowledging this fact, this research seeks to find a passable solution which highlights the advantages of using such a sense of local space in localisation, without providing a general solution to this particular problem. Also, it must be noted that the proposed solution, which uses carpet detection techniques, is limited to environments in which the ground plane has a constant and consistent colour.

Determining Local Space with Monocular Visual Sensors

The detection of open space is akin to that of obstacle detection.

Horswill (1993) describe a mobile robot, Polly, which used a monocular camera to perform online obstacle detection and avoidance. To accomplish this the system detects the ground plane by forming a Radial Depth Map (RDM) of the area visible in the camera's field of view, and executes navigation strategies according to this map of local space. The RDM is formed using an edge detector to detect texture in the visual scene, labeling regions in the image that contain texture as representing background objects, and those which contain no texture as belonging to the ground plane.

Cheng and Zelinsky (1996) describe a similar robot system with a monocular camera which used template matching to detect carpet regions in the visual field, and thus identify the ground plane. A carpet template was stored and matched with the current visual field using dedicated hardware and regions which are similar to carpet are identified. Using this approach, the mobile robot could perform exploration and goal seeking behaviours at quite high speeds without colliding with walls or other objects.

The reliable performance of Cheng and Zelinsky's (1996) ground plane detection was achieved due to the positioning of the camera and the resulting field of view. The Yamabico robot used in this research was low to the ground standing at no more than 40cm. The camera was mounted on top and angled downwards at the carpet directly in front of the robot. In this way the depth of objects within the visual field was limited and high image resolution was maintained and the appearance of carpet within the field of view was quite consistent.

The affect of a deeper and more extensive visual field was noted by Gaskett, Fletcher and Zelinsky (2000). Using the same same vision processing system, this time mounted on a Nomad200 mobile robot, Gaskett et al. (2000) used detected free space as input for the acquisition of wandering and visual serving behaviours through reinforcement learning. Since the camera was mounted at the greater height walls and tended to correlate well with the carpet template, an additional constraint was added: anything above a non-carpet region in the image, must also be a non-carpet region.

Stepan and Kurlich (2001) have taken carpet matching one step further, using detected carpet regions to build up occupancy grid maps of the environment. Their system uses monocular camera images which appear to be captured from a camera situated close to the floor similar to that of Cheng and Zelinsky (1996). To combat the problem of obstacles and carpets sharing a similar appearance, they built up probability distributions in HSV colour space which modelled the carpet and non-carpet appearance in training images. Occupancy grids can then be constructed using Bayesian probability, with the probability of a current image region being carpet being defined as the similarity of its HSV measure to that of the models describing carpet and non-carpet regions. The map was maintained in 2 dimensional robot coordinates by transforming images from the the camera coordinate system to a floor coordinate system.

This probabilistic approach marks a shift from pure obstacle avoidance to map building. This can be thought of as a shift from the detection of free space to the detection of local space. By maintaining an estimate of surrounding space through time, the presence of temporary, dynamic obstacles can be eliminated from representation describing the extent of local space.

Determining Local Space using Panoramic Visual Sensors

Panoramic cameras produce images with a lower resolution particularly in image regions with a low angle of elevation, which is precisely where the presence of carpet is most likely. For this reason, the template matching approach to detecting free space is not common when using panoramic visual sensors. Alternatively researchers have tried to extract carpet boundaries through feature extraction. Yamazawa et al. (1995) used a panoramic sensor to detect obstacles and form a map of local space in the environment. This was accomplished by transforming the panoramic image into a floor plan perspective, and extracting line edges and direction from the resultant images using a differential operator. The elimination of radial lines from this process produced a floor map of the region. This type of feature extraction is obviously targeted towards environments which are highly structured and contain uninterrupted lines at the floor/wall boundary.

Similarly, Gaspar et al. (2000) present a mobile robot system which uses panoramic vision to navigate through a highly structured environment. Again the panoramic image is transformed into a floor plan, or birds eye view image. From this image parallel lines are extracted and this produced an estimate of the edges of a corridor. From there the robot can steer along the middle of the corridor, resulting in a centering behaviour. This system assumed everything within the parallel lines is free space.

It can be seen from these studies, that true detection of free space from a monocular camera is difficult to achieve. The carpet matching approach suffers from the non-uniformity of carpet appearance while line extraction suffers from limiting the com-

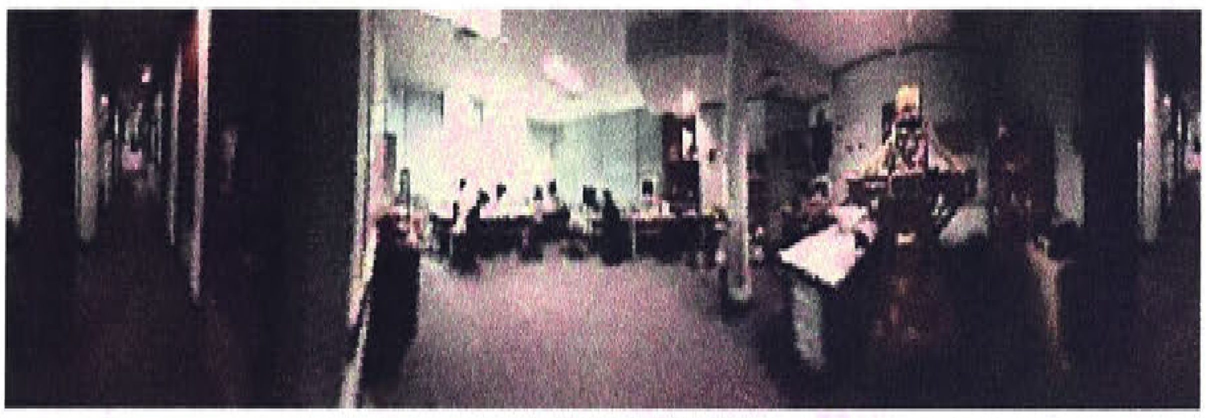


Figure 5.1: An example of carpet in panoramic images.

plexity of the environment to rigid structures. In this system, the desire to detect free space both in structured corridors and in cluttered rooms led to the choice of ground plane detection through carpet matching to achieve measurement of local space.

5.1.1 Carpet Matching

Carpet matching is an imprecise and location specific method of detecting the extent of local space surrounding the robot. When using a panoramic sensor in the environment shown in Figure 5.1 it becomes even more so. The low resolution in the panoramic image means that any distinguishing patterns in the carpets spatial frequency are lost. Further more the blurring of colours results in the carpet being virtually indistinguishable from the walls using standard pixel or region matching techniques. In addition, variations in lighting cause dramatic changes in carpet appearance and again can lead to confusion of carpet with the walls and other surfaces.

Average Pixel Colour Matching

One method for detecting regions of carpet in panoramic images is to calculate an average pixel colour for carpet. By finding the absolute difference between this average pixel and the colour of each pixel in a panoramic image, a measure of how similar each pixel is to the colour of carpet can be made.

Figure 5.2 shows the results of performing this absolute difference calculation on example panoramic images. Two examples are presented: a) an image captured in a large room and b) an image captured at a T-intersection of an office corridor. The average carpet pixel for each image was taken from the region contained within the

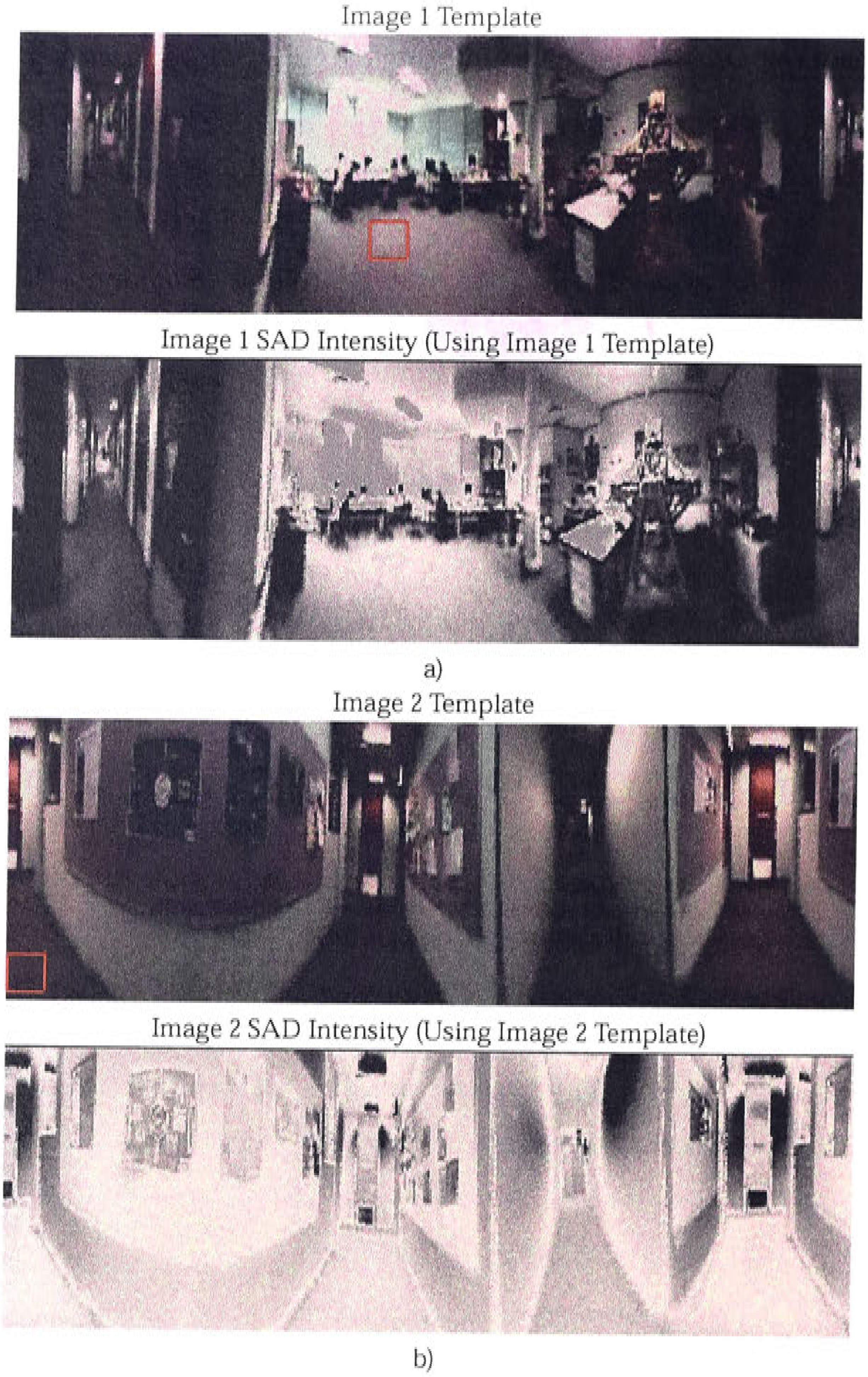


Image 1 Template

Image 1 SAD Intensity (Using Image 1 Template)

a)

Image 2 Template

Image 2 SAD Intensity (Using Image 2 Template)

b)

Figure 5.2: Carpet matching using average pixel colour from identified carpet region. Two examples of carpet matching in panoramic images using this technique are given: a) image captured from a large room and matching results; b) image captured at a T-intersection in a corridor and matching results. In both cases the average carpet pixel was calculated from a template (red box) taken from the captured image. High intensity pixels in the matching results represents a high correlation with the average carpet pixel.

red boxes on the normal panoramic images. The intensity plots under each example image show the average pixel differences. A low pixel difference results in a brighter intensity while high pixel differences are shown with darker intensities. The results show that the matching with carpet regions in both images, produce widely varying results. The intensity levels tends to vary with illumination of the carpet and non-carpet regions. The difficulty in the carpet detection is to correctly identify carpet regions irrespective of illumination changes while limiting the false detection of non-carpet regions.

These examples stress the obvious point that an average pixel carpet measure will only be representative of carpet under a particular illumination level. Figure 5.3 shows the difficulties of trying to match carpet regions using an average carpet pixel that was calculated from example carpet pixels from different images. The carpet matching intensity plots are from the same panoramic image examples contained in Figure 5.2 but in each case the absolute difference calculation was performed with the average carpet pixel from the other image. The intensity plot in Figure 5.3 a) was produced by finding the absolute difference between the panoramic image in Figure 5.2 a) and the average pixel derived from the red boxed region in Figure 5.2 b). These results further highlight the inappropriateness of using illumination variant approaches to carpet matching.

Normalised RG Colour Space Matching

Stepan and Kurlich (2001) reported a successful carpet matching system which used a colour space model of carpet to detect carpet regions. The objective was to build a model of carpet which is independent off the level of illumination in the image.

A colour space model of the colour of carpet can be constructed using carpet colour samples from panoramic images. RG colour space is used as the image obtained from the video capture device is in RGB format. The process of normalising an RGB value makes one of the values redundant, thus leaving only RG colour. Converting an RGB pixel to a normalised RG one is done as follows:

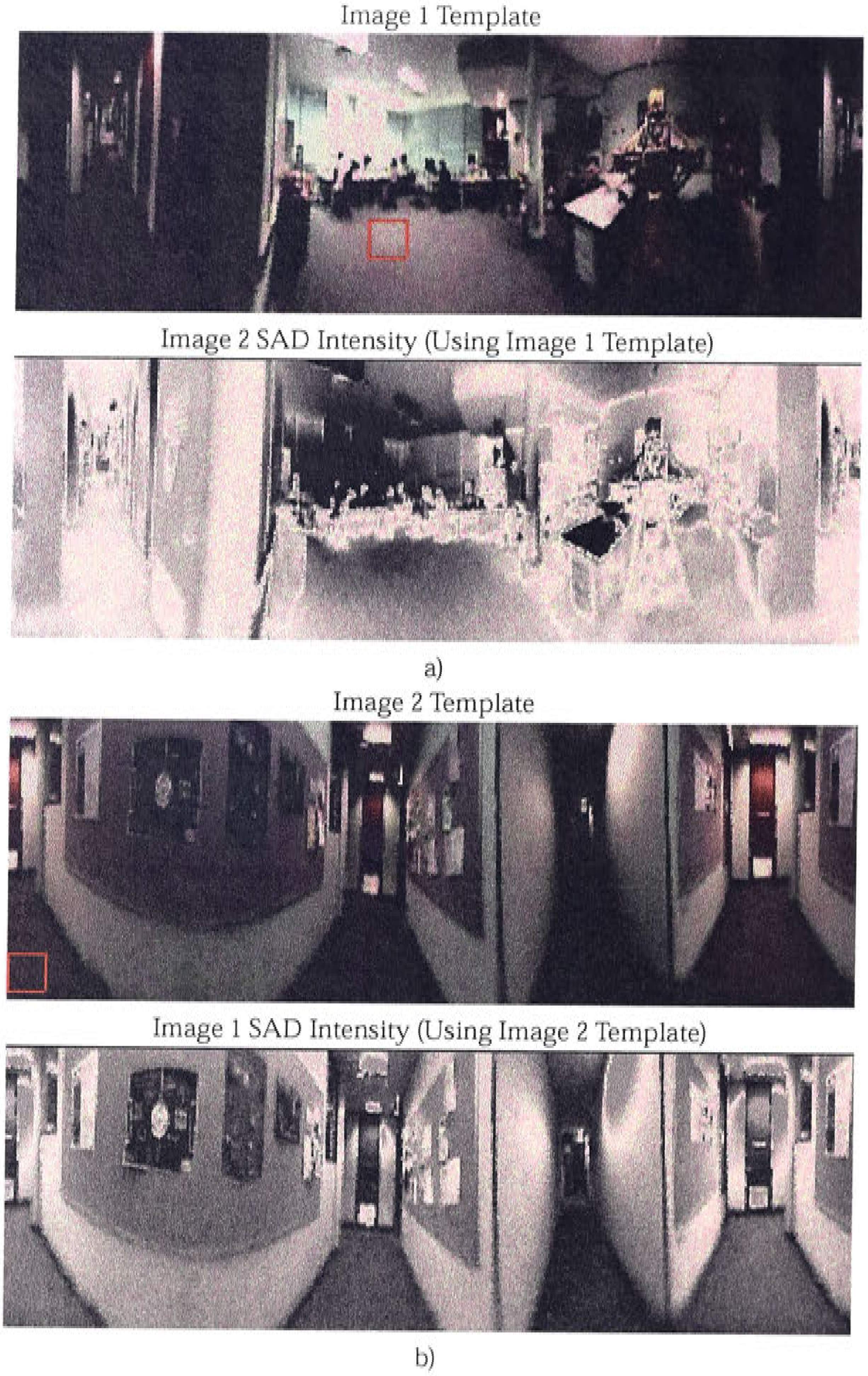


Image 1 Template
Image 2 SAD Intensity (Using Image 1 Template)
a)
Image 2 Template
Image 1 SAD Intensity (Using Image 2 Template)
b)

Figure 5.3: Carpet matching using average pixel colour from identified carpet region in another image. Part a) shows an intensity plot produced by performing an absolute difference calculation on image 2 using the average carpet pixel colour from the template in image 1. Part b) shows the results of matching image 2 with the average pixel colour from the template in image 1.

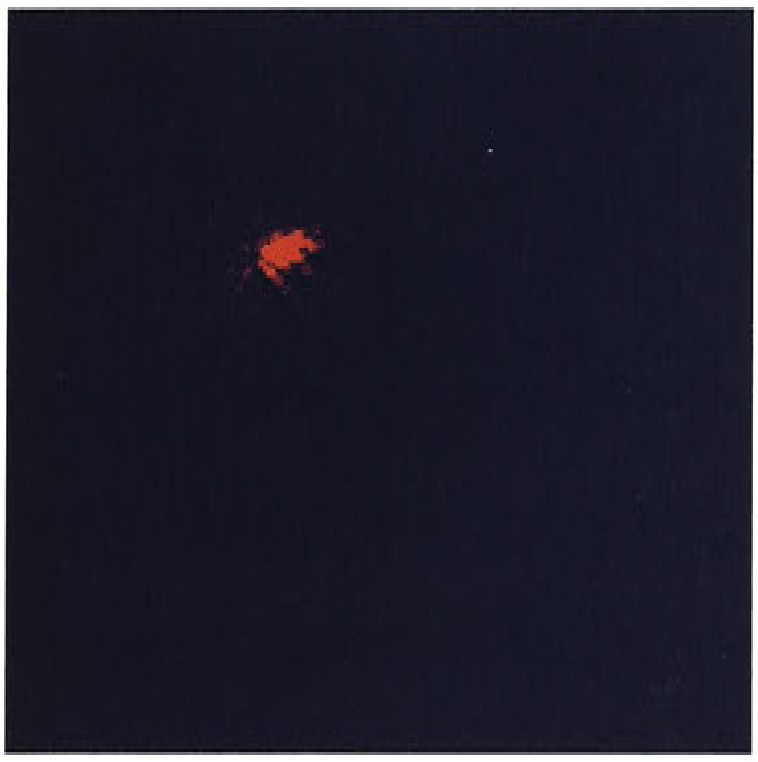


Figure 5.4: Normalised RG colour space of carpet regions from panoramic images. The red dimension lies along the horizontal axis, while the green lies along the vertical.

$$P_{normRG}(R', G') = \left(\frac{R}{R+G+B} * 255, \frac{G}{R+G+B} * 255 \right) \tag{5.1}$$

The colour model is made by sampling pixels of carpet regions from panoramic images and using then to construct an approximation of the probability distribution for carpet colour over the normalised RG colour space. In implementation, this means populating a 255×255 (representing RG values) by incrementing the appropriate cell for each example carpet pixel. The resultant matrix is then convolved with a Uniform 3×3 pixel kernel[1] and normalised to hold an integer value between 0 and 255.

$$K = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{5.2}$$

The value in a particular RG cell then provides a measure of the likelihood that the colour associated with that cell is an instance of carpet colour. Figure 5.4 shows the normalised RG colour model of carpet constructed from sections of carpet from sam-

[1] Gaussian kernels and kernels of varying sizes were tried but a Uniform kernel of size 3×3 proved superior for carpet matching given the small pixel sample size used to construct the colour model

ple panoramic images. The sections of carpet were identified by hand over a small number of training images captured from different locations in the environment.

Using the colour space model of carpet, the probability of each pixel in a panoramic image being the colour defined by the carpet colour model can be calculated in two easy steps:

1. Convert pixel to normalised *RG* form
2. Look up the value associated with that *RG* pixel in the carpet colour model

The results of applying this method to example panoramic images are shown as intensity plots in Figure 5.5. A high pixel intensity level reflects that the corresponding pixel in the panoramic image is likely to be the colour of carpet. Again there are two examples and the original panoramic images are given above the intensity plots. At first glance this does not seem to be much of an improvement on the results from matching with an averaged pixel value, with many regions that are obviously not carpet being strongly identified as carpet. This is true, but it is also the case that all carpet regions have been identified correctly irrespective of the level of illumination.

Carpet Boundary Detection

The carpet detection results gained from matching pixels with a *RG* colour model of carpet show that it is not sufficient to simply use carpet colour to form a reliable estimation of local free space. As mentioned earlier, Gaskett et al. (2000) use an additional heuristic rule to enhance carpet matching performance. In that case, the detection of a non-carpet region disqualified regions above it from being identified as carpet regions. A similar approach is taken in this research. It is assumed that a change in colour gradient defines the transition between carpet and non-carpet regions. The detection of such a transition can then be used to detect the boundary between two different surfaces in the image. The first such transition detected, working on the image in vertical columns from the bottom up, can be assumed to be the transition between carpet and non-carpet regions.

This does not help us when walls that look like carpet start from the very bottom of the image and extend upwards through the entire vertical visual field. Another rule

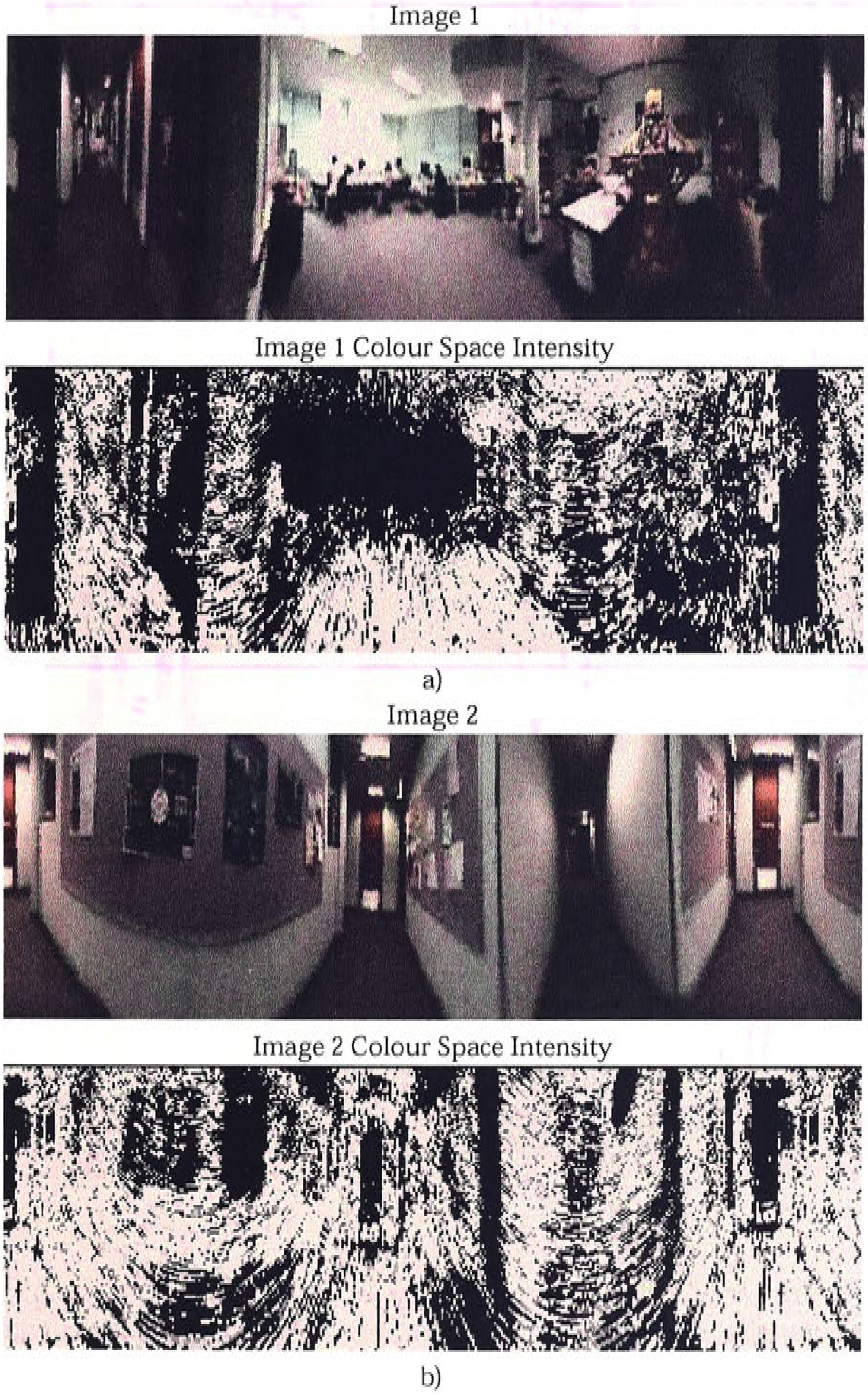


Image 1

Image 1 Colour Space Intensity

a)

Image 2

Image 2 Colour Space Intensity

b)

Figure 5.5: Carpet matching using RG colour space model of carpet colour. The top images show the panoramic images and the bottom images show the intensity plots of detected carpet regions.

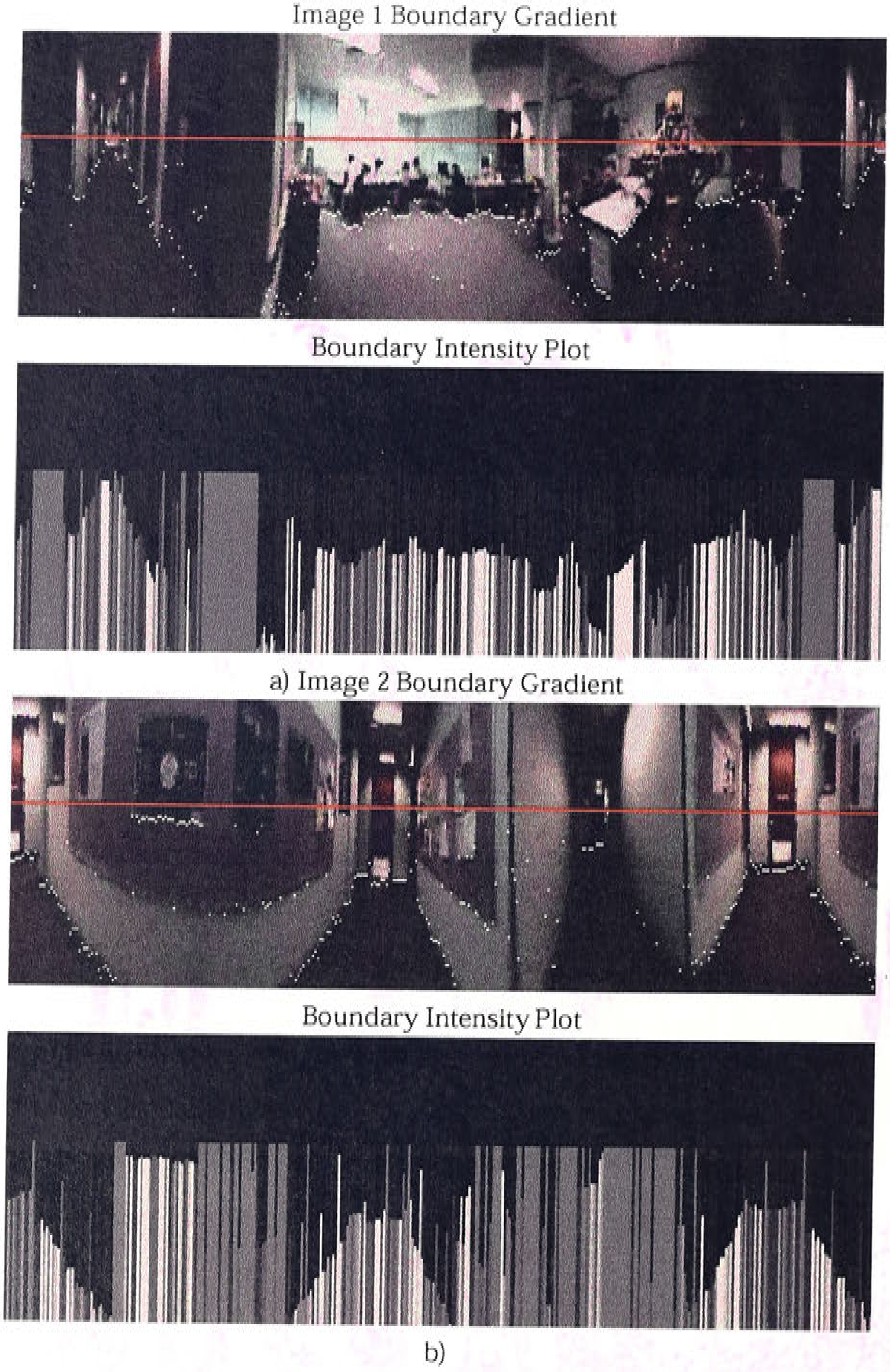


Image 1 Boundary Gradient

Boundary Intensity Plot

a) Image 2 Boundary Gradient

Boundary Intensity Plot

b)

Figure 5.6: Carpet detection using a gradient boundary. The top images in part a) and part b) show the detected boundary regions in each pixel column as white dots. The red lines show the defined horizon in the panoramic images. The bottom images show the carpet detection intensity plots produced based on gradient detection.

Table 5.1: Detecting the gradient boundary in a panoramic image

```
-- iteration
for each pixel column $c$ in Image $I$
   $gradient\_found = FALSE$
   $r = 0$
   -- iterate rows from bottom upwards
   while $r \leq image\_horizon$ and $gradient\_found = FALSE$ do
      $mag = abs(I(r, c) - I(r - 1, c))$
      if $mag > gradient\_threshold$ then
         $gradient\_found = TRUE$
         $gradient\_row = r$
         $gradient\_mag = mag$
      end

   end
   if $gradient\_found = TRUE$ then
      $gradient\_row(c) = gradient\_row$
      $gradient\_mag(c) = gradient\_mag$
   end
end
```

is introduced that states there must be such a transition before the height in the image which corresponds to the horizon in the visual field. It should be noted that no regions of carpet lying on a flat plane along which the robot is translating should ever appear above the horizon. In this way sections of wall which would otherwise be detected as carpet regions due to their similarity with carpet can be avoided.

Figure 5.6 shows two example panoramic images. The top images in both sub-figures a) and b) are overdrawn with detected gradients in white and with the image horizon in red. It can be seen that most boundaries between carpet and non-carpet regions have been correctly identified. There are problems with extremes of illumination as seen in the dark corridor of the second panoramic image.

In this case gradient detection was achieved by performing an absolute difference calculation on vertically adjacent pixels in every column in the image. Starting from the bottom and proceeding upwards, the first such gradient detected above a threshold value was defined to be the carpet boundary. If no gradient above the threshold is found before reaching the image horizon then that pixel column is assumed to have no carpet regions. Any pixels above horizon line can be said with absolute certainty to not be carpet pixels. Gradient boundary detection is shown in algorithmic form in

Table 5.1.

In summary, when assigning a pixel $p(x, y)$ from a panoramic image a carpet probability measure $P(p(x, y))$, assuming gradient detection has already been applied, the following steps are applied:

1. If pixel p is above the image horizon:

$$P(p(x, y)) = 0$$

2. If pixel p lies below a detected gradient of magnitude g_{mag} then:

$$P(p(x, y)) \propto g_{mag}$$

3. If pixel p lies above a detected gradient of magnitude g_{mag} then:

$$P(p(x, y)) \propto 1 - g_{mag}$$

4. If pixel p is below the image horizon and no gradient was detected in p's column:

$$P(p(x, y)) = C$$

where C is a constant which reflects the uncertainty in columns in the image where no gradient boundaries have been detected, which may or may not contain carpet regions. In this system C is set to 0.3.

The results of applying this method of carpet detection to panoramic images is shown in Figure 5.6. For two example images the resulting probabilities that pixels are in regions of carpet are shown as intensity plots. A clear boundary can be seen in both examples, and for the most the lighter pixels in the intensity plot correspond to carpet regions in the panoramic images. There are mis-classifications, particularly in the columns where no boundary was detected. However, the method is sufficiently accurate for our research purposes.

5.1.2 Occupancy Grid of Local Space

All the examples of carpet detection methods have been applied to single images. These images are noisy and sometimes lack the information necessary to correctly identify carpet regions. By promulgating these carpet estimates through time while incorporating new information a better and more robust method of carpet detection can be achieved.

Similar to Stepan and Kurlich (2001), our research treats free space detection as an occupancy grid problem and uses Bayesian probability theory to maintain the free space estimate. In our research the occupancy grid represents the probability that a location in the 'grid' of a panoramic visual scene is 'occupied' by a carpet coloured pixel. It is important to note that the occupancy grid is in image space and not in physical space. The occupancy grid can be constructed using basic probability theory:

$$P_{new}(Occ) = P(Occ)P_{Acc}(Occ) \tag{5.3}$$

which states that the new probability of a cell in the occupancy grid being occupied, $P_{new}(Occ)$, is the probability that is it occupied at this time period $P(Occ)$ multiplied by the probability of it being occupied accumulated from past measurements, $P_{Acc}(Occ)$.

Stepan and Kurlich (2001) uses the Bayesian rule[2] to recursively update the accumulated probability each time a new sensor reading is obtained:

$$P_{Acc'}(Occ) = \frac{P(Occ)P_{Acc}(Occ)}{P(Occ)P_{Acc}(Occ) + (1 - P(Occ))(1 - P_{Acc}(Occ))} \tag{5.4}$$

The carpet matching methods can be applied to a sequence of images and the information of probable carpet regions can be accumulated over these images. Figure 5.7 shows a sequence captured in the middle of a large room from a stationary position. Each carpet matching method previously described was applied to this image sequence using Equations 5.3 and 5.4 to accumulate probabilities. The results of the average carpet pixel, colour space model and gradient methods after frames 1 and 5

[2] as presented in (Press, 1989)

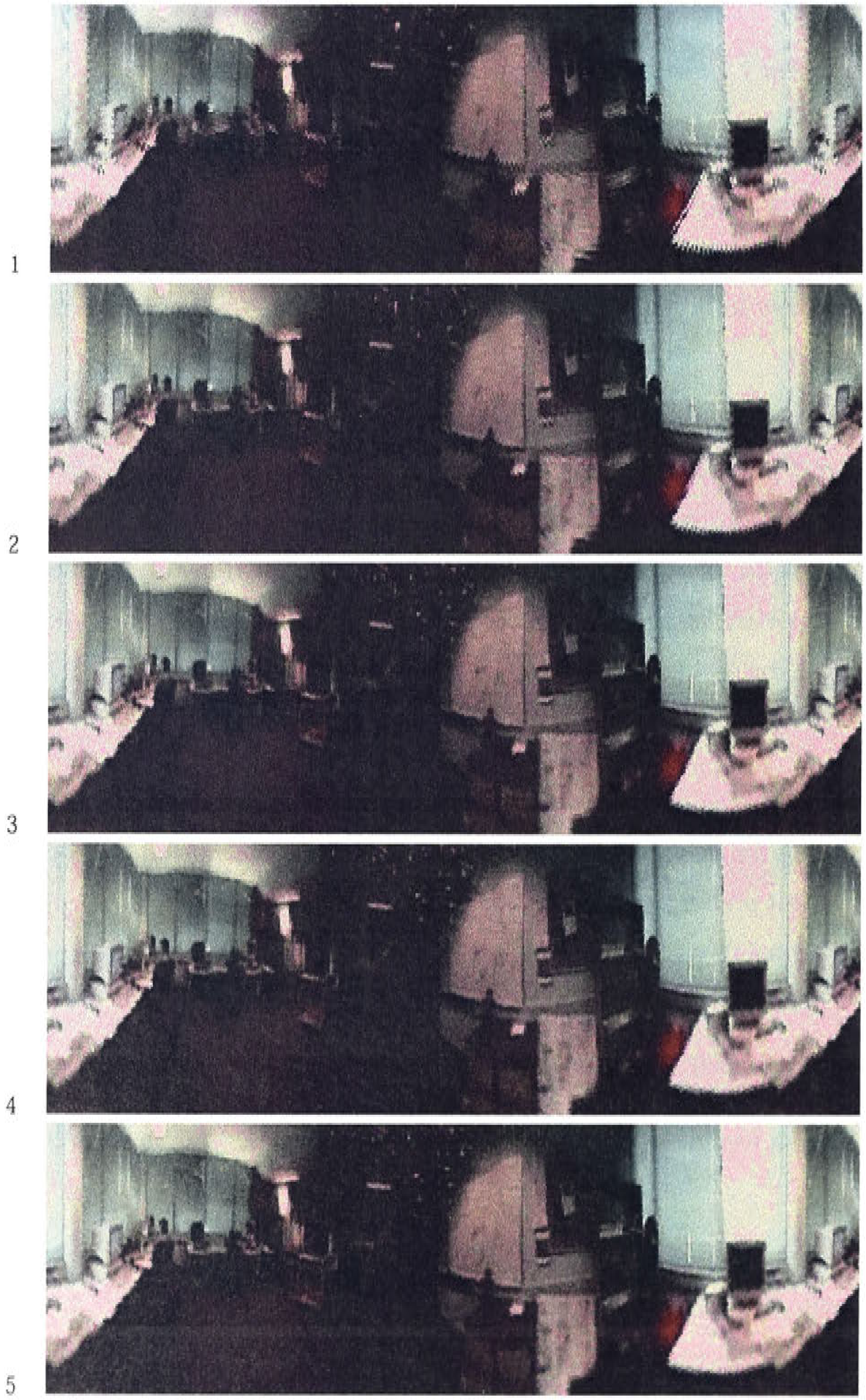


Figure 5.7: A sequence of five panoramic images from a stationary camera

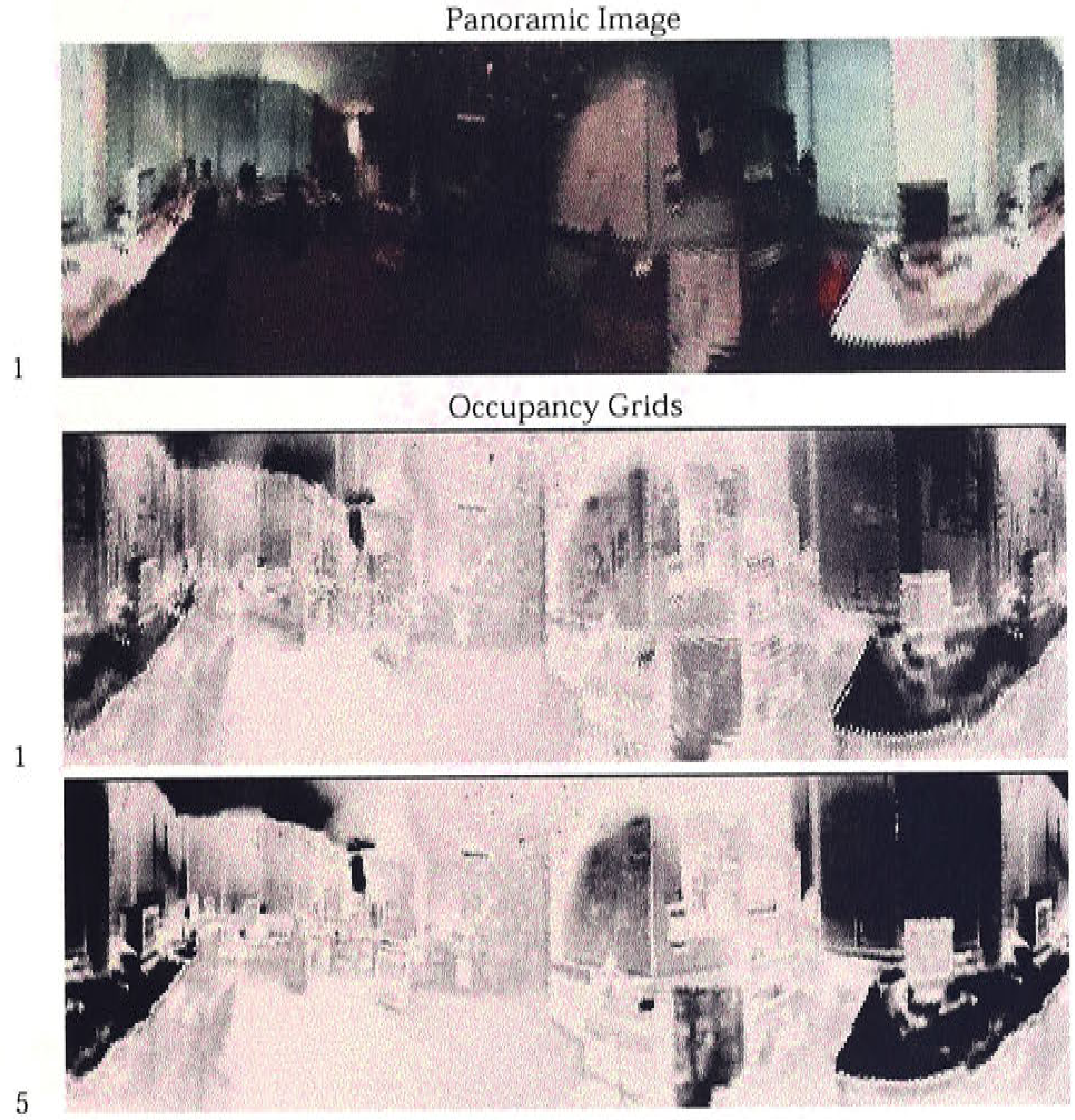


Figure 5.8: Average pixel carpet matching using occupancy grids and Bayesian probability theory. The initial probability grid from frame 1 and the accumulated probability after frame 5.

are shown in Figures 5.8, Figure 5.9 and Figure 5.10 respectively. It can be seen that although there is no great improvement of carpet region detection over the sequence for any of the methods, most noise associated with the use of a single image is eliminated, especially in the gradient detection approach.

Using the Bayes update formula, the occupancy grid can not only be updated over time, but also within the one time step with probabilities from multiple sensor modalities. Therefore the probability of a given cell in the occupancy grid can be updated by multiple carpet detection methods.

Figures 5.11 and 5.12 shows the occupancy grids formed by combining the results of colour matching and the gradient search. Again the probability that a cell is occupied

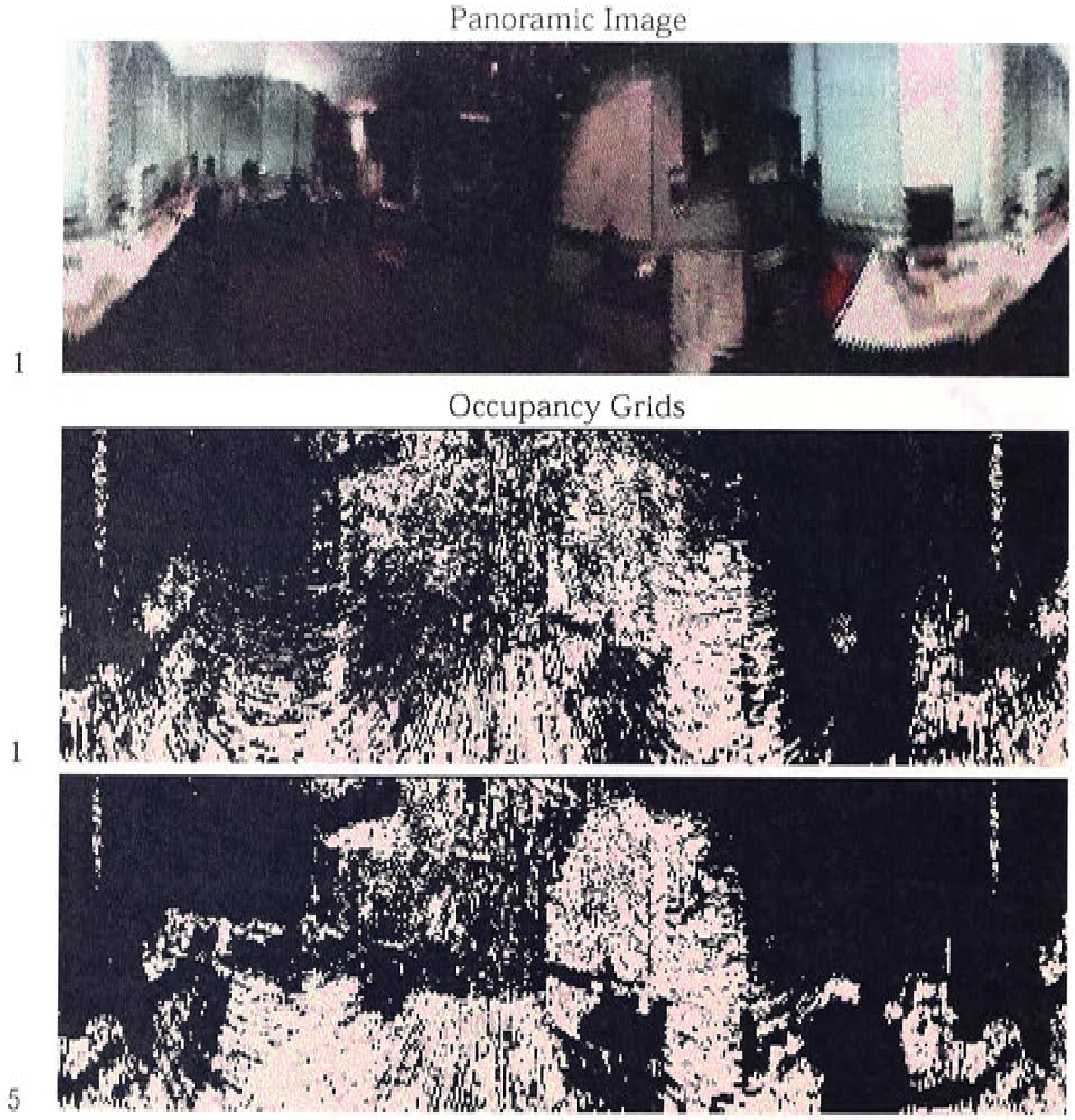


Figure 5.9: Colour space carpet matching using occupancy grids and Bayesian probability theory. The initial probability grid from frame 1 and the accumulated probability after frame 5.

is shown by the intensity levels in the figure. Dark pixels represent regions which are occupied while lighter regions represent free space. Two single image examples are shown. The top image in each figure is the original panoramic image. Next are the results of applying the colour space model, the gradient detection, and finally the combination of both methods through the Bayes update formula. It can be seen that even with the combination of the two sources of information that some mistakes are still made. In particular the example in Figure 5.12 shows that the carpet region located in the darkest corridor is not detected well. This is due to the lack of colour and gradient information available from the excessively dark pixels.

The combined information can then applied over the sequence of images shown in

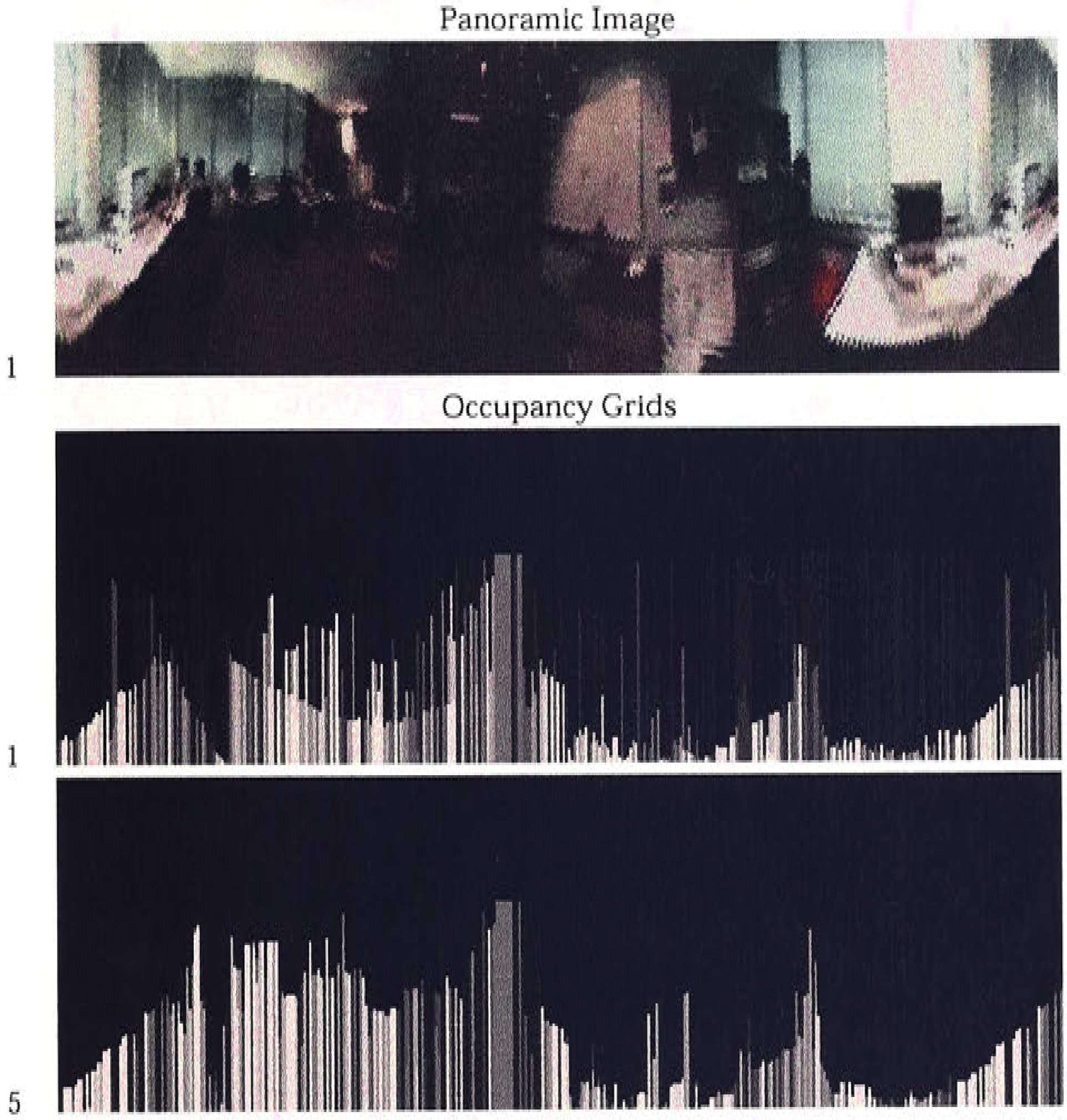


Figure 5.10: Gradient boundary carpet detection using occupancy grids and Bayesian probability theory. The initial probability grid from frame 1 and the accumulated probability after frame 5.

Figure 5.8. The results of combining two methods of carpet detection over the image sequence are shown in Figure 5.13, which combines average carpet pixel and gradient detection and Figure 5.14, which combines carpet colour space model and gradient detection. Figure 5.13 shows higher probabilities than Figure 5.14 but also has more false positive carpet carpet identifications. Combining the colour space and gradient detection methods over a sequence of images provides the best carpet region detection results. The various methods of carpet detection described previously are empirically evaluated in the following section.

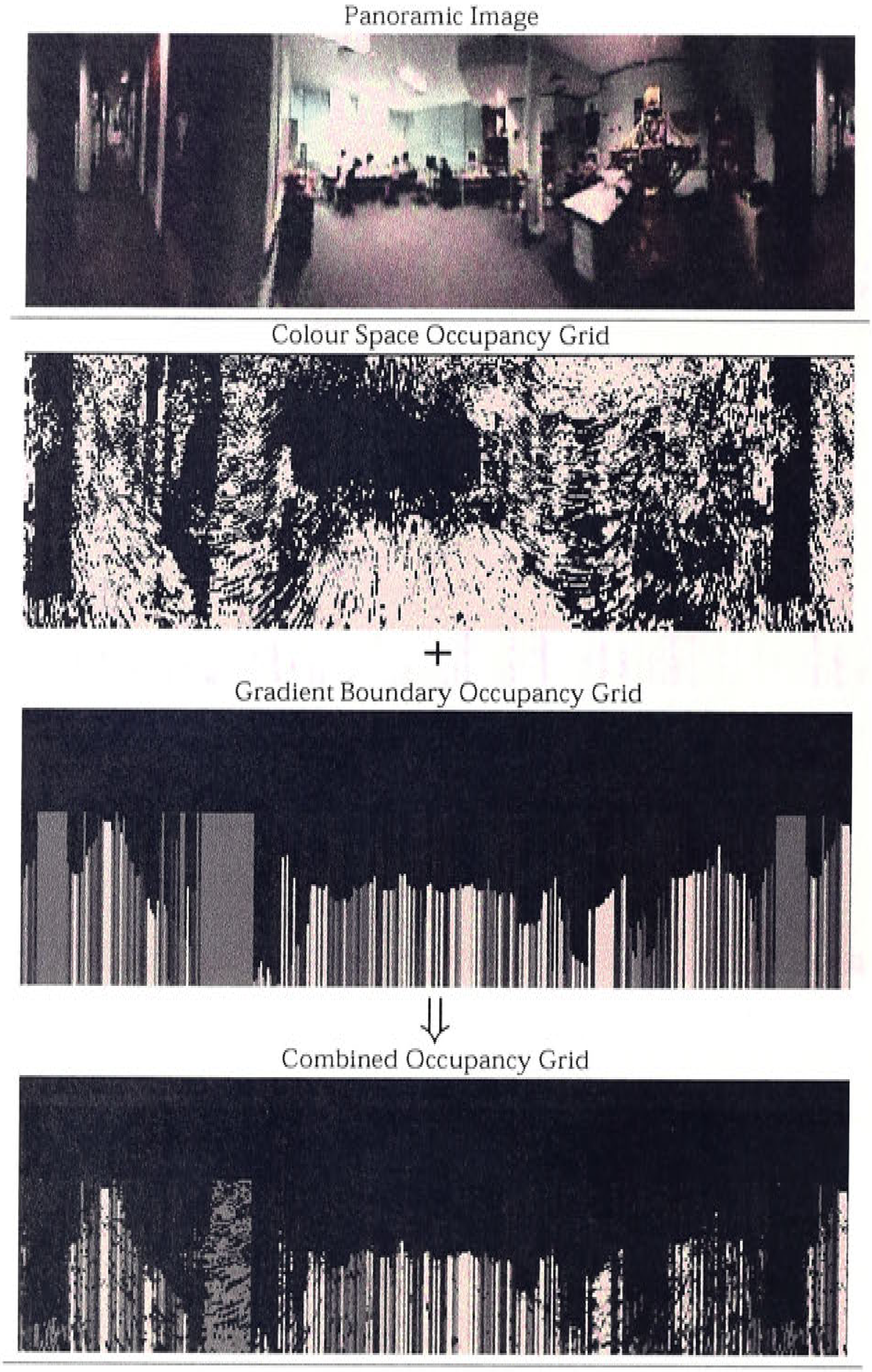
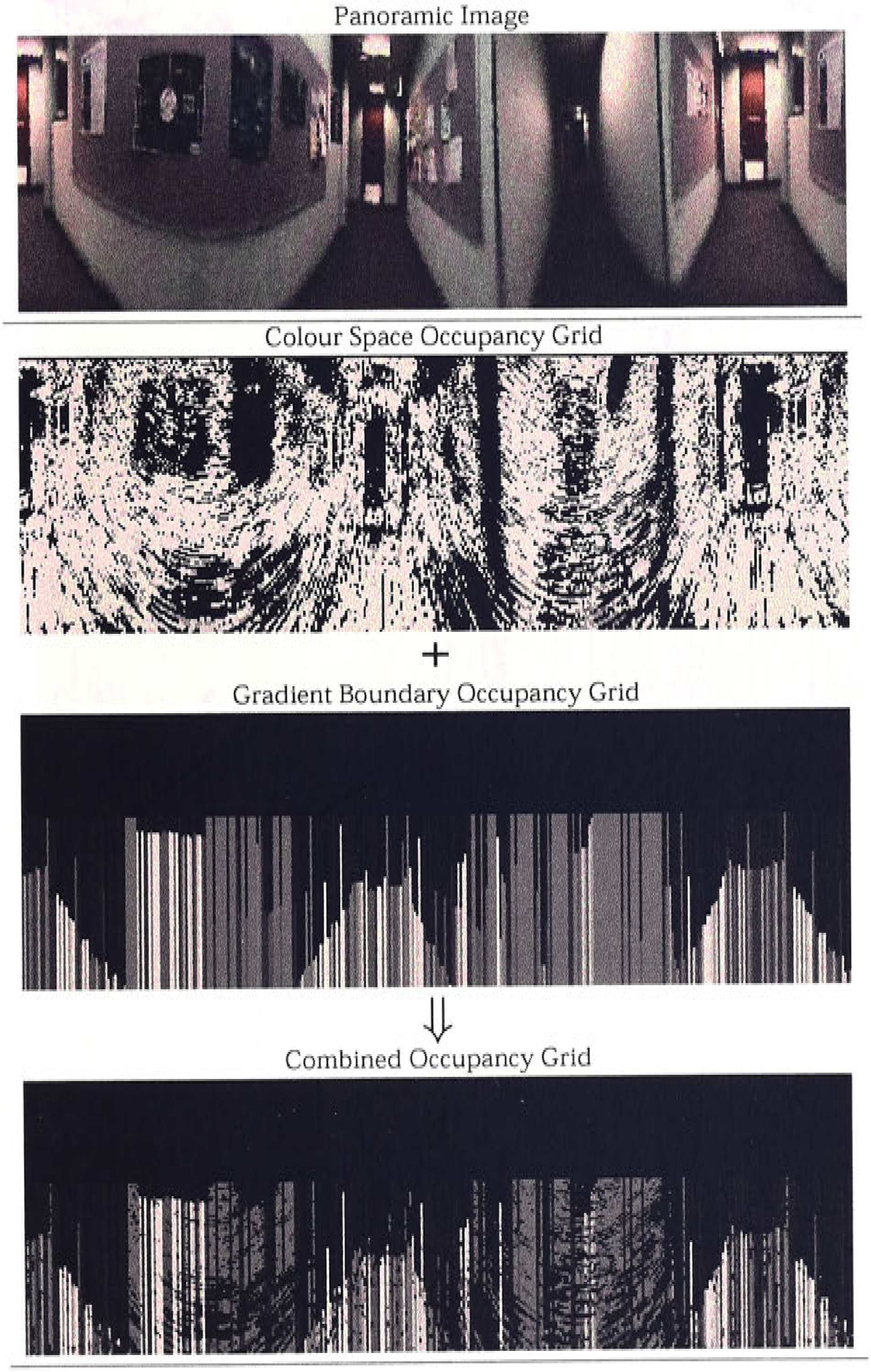


Figure 5.11: Combining gradient boundary detection and carpet colour space model matching using Bayesian probability theory.



Panoramic Image

Colour Space Occupancy Grid

Gradient Boundary Occupancy Grid

Combined Occupancy Grid

Figure 5.12: Combining gradient boundary detection and carpet colour space model matching using Bayesian probability theory.

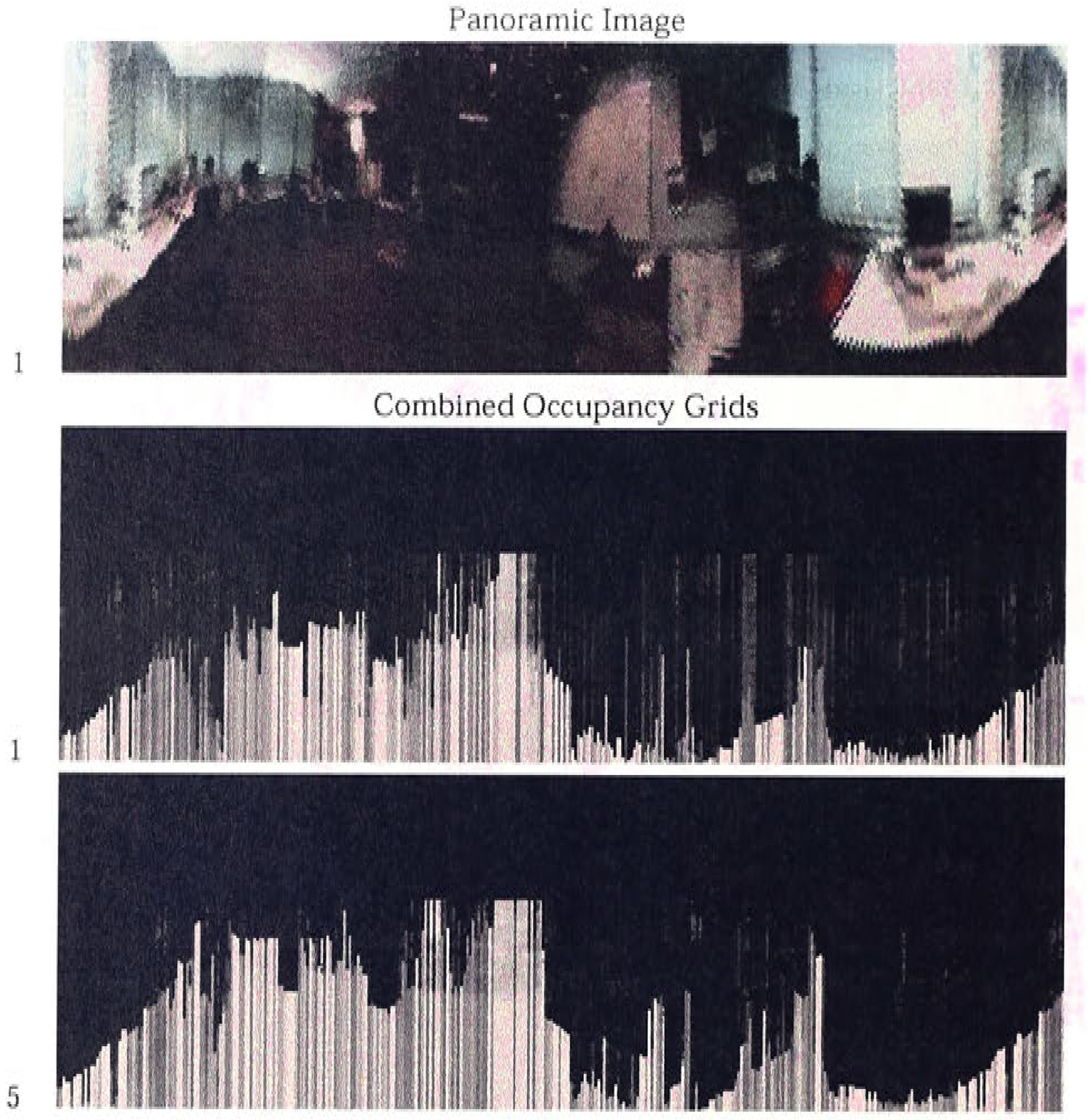


Figure 5.13: Combining average pixel matching and gradient boundary carpet detection techniques over time. Occupancy grids from frames 1 and 5 in the image sequence are shown.

5.1.3 Evaluating Local Space Detection

It is difficult to compare the results of the various carpet matching methods by visual inspection alone. An objective measure is needed to determine which method performs the best. The results of each carpet matching method can be compared with the ground truth. The ground truth, while difficult to obtain exactly, can be approximated by visual inspection and manually identifying carpet regions in panoramic images. In order to enable comparison with the output of the carpet matching methods described above, the ground truth can be represented as an occupancy grid with carpet/non-carpet regions known with absolute certainty. Figure 5.15 gives an example of a a) panoramic image and b) the associated manually identified occupancy grid.

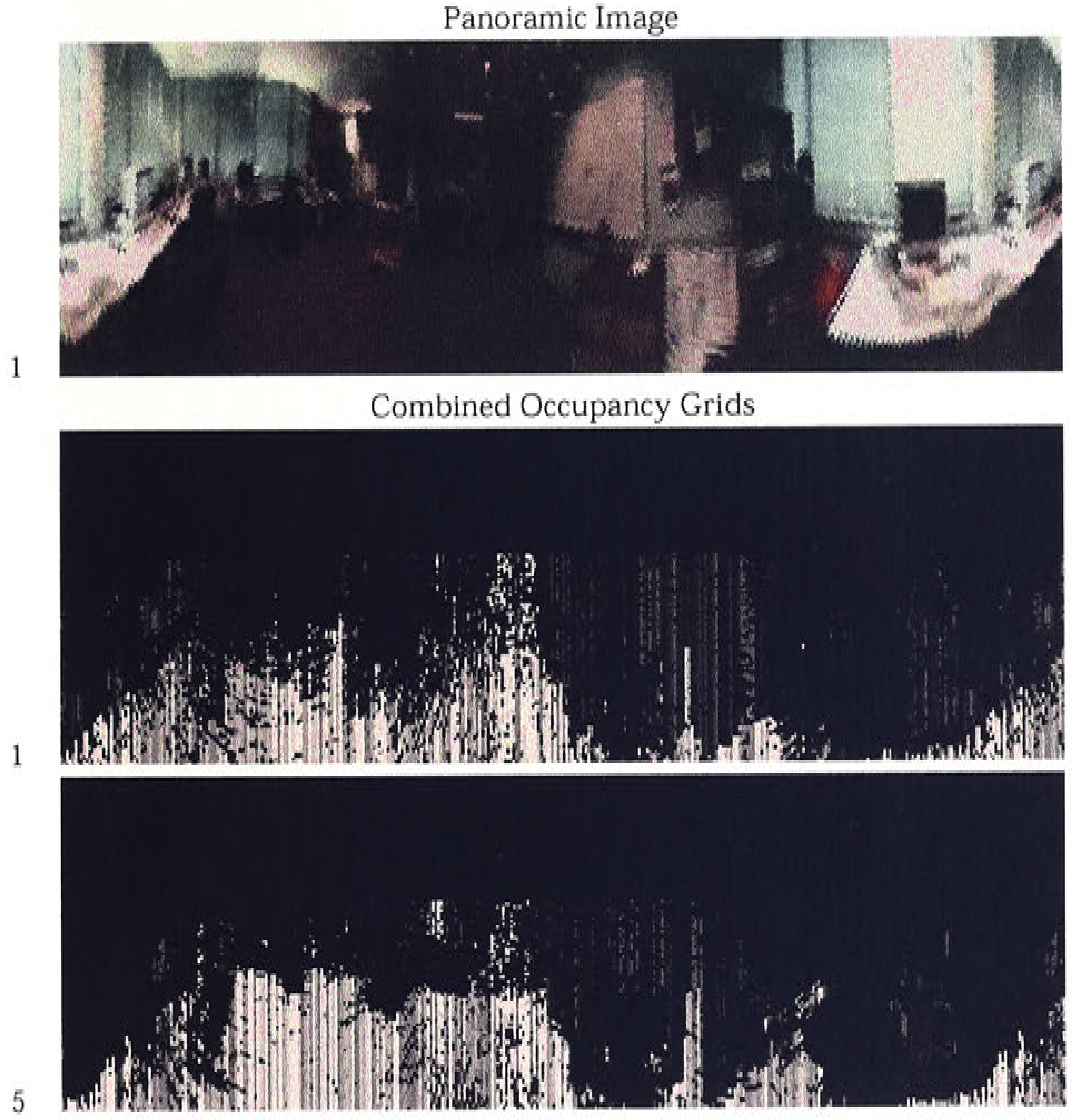
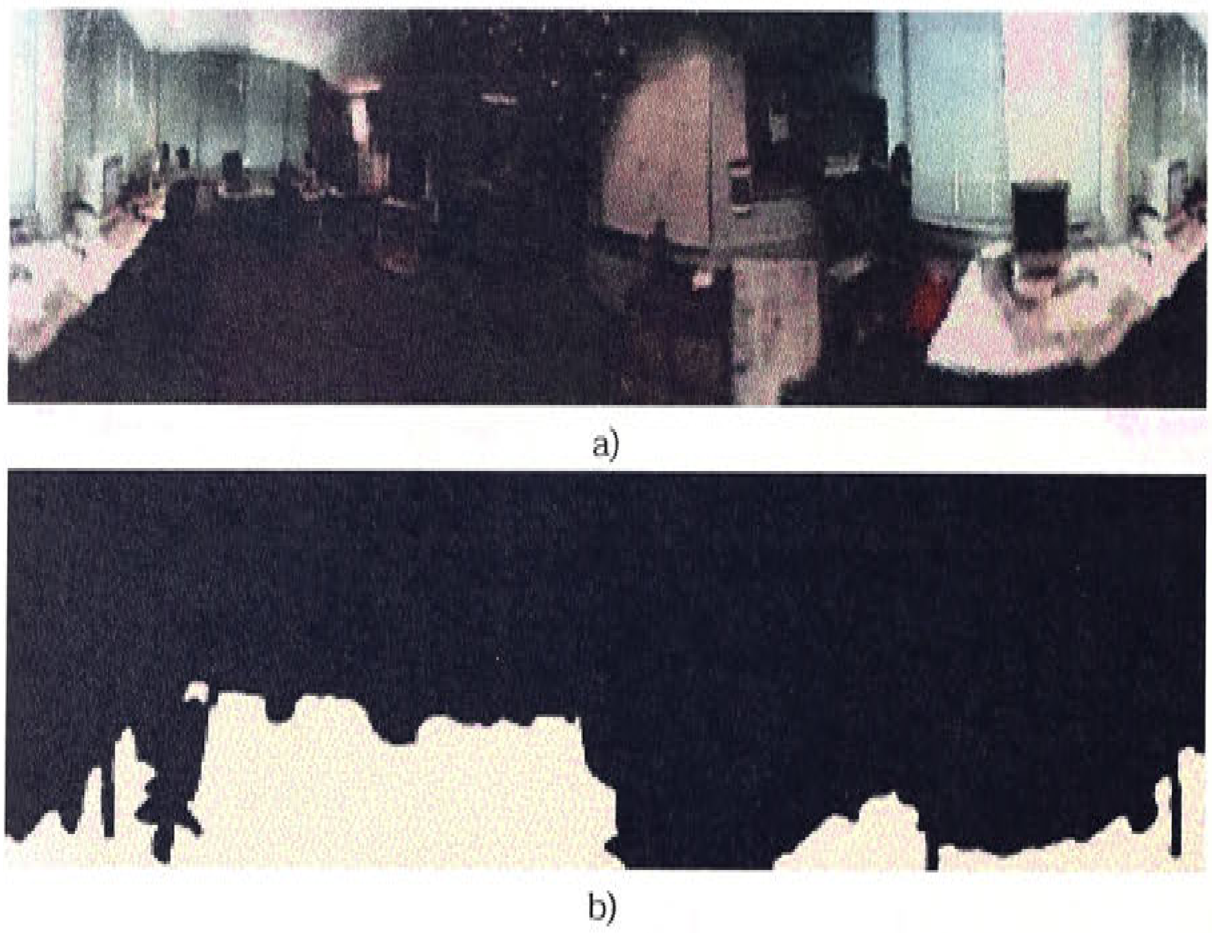


Figure 5.14: Combining colour space matching and gradient boundary carpet detection techniques throughout an image sequence.Occupancy grids from frames 1 and 5 in the image sequence are shown.

A comparison of a carpet matching occupancy grid and the ground truth can then be made by subtracting one grid from the other, with the absolute difference being a measure of the similarity between the two. This process highlights the differences between the two grids and identifies regions of false positives as well as false negatives. Figure 5.16 shows the results of performing this subtraction on occupancy grids produced by applying the carpet matching methods of a) colour space matching, b) gradient detection and c) both colour space and gradient detection.

Summation of the absolute differences from all cells in the grid gives a quantitative measure of how closely a result matched the ground truth. More specifically the difference ϵ between the ground truth grid G^{GT} and the example result grid G^{R} is given



a)

b)

Figure 5.15: Ground truth of carpet matching: a) the panoramic image and b) the manually identified occupancy grid denoting the ground truth of known carpet regions in the image. White pixels represent absolute certainty of carpet regions.

Method	Total: ϵ	Average: ϵ/N
	After 1 image	
Ave. pixel	5498865	127
Colour Model	3025180	70
Gradient	1618066	37
Grad. & Ave. pixel	1454562	33
Grad. & Colour model	1595560	36
	After 5 images	
Ave. pixel	5000955	115
Colour Model	2617984	60
Gradient	1473617	34
Grad. & Ave. pixel	1406868	32
Grad. & Colour model	1323679	30

Table 5.2: Carpet matching performance: total and average pixel value of image subtraction between known and estimated results for different carpet detection methods.

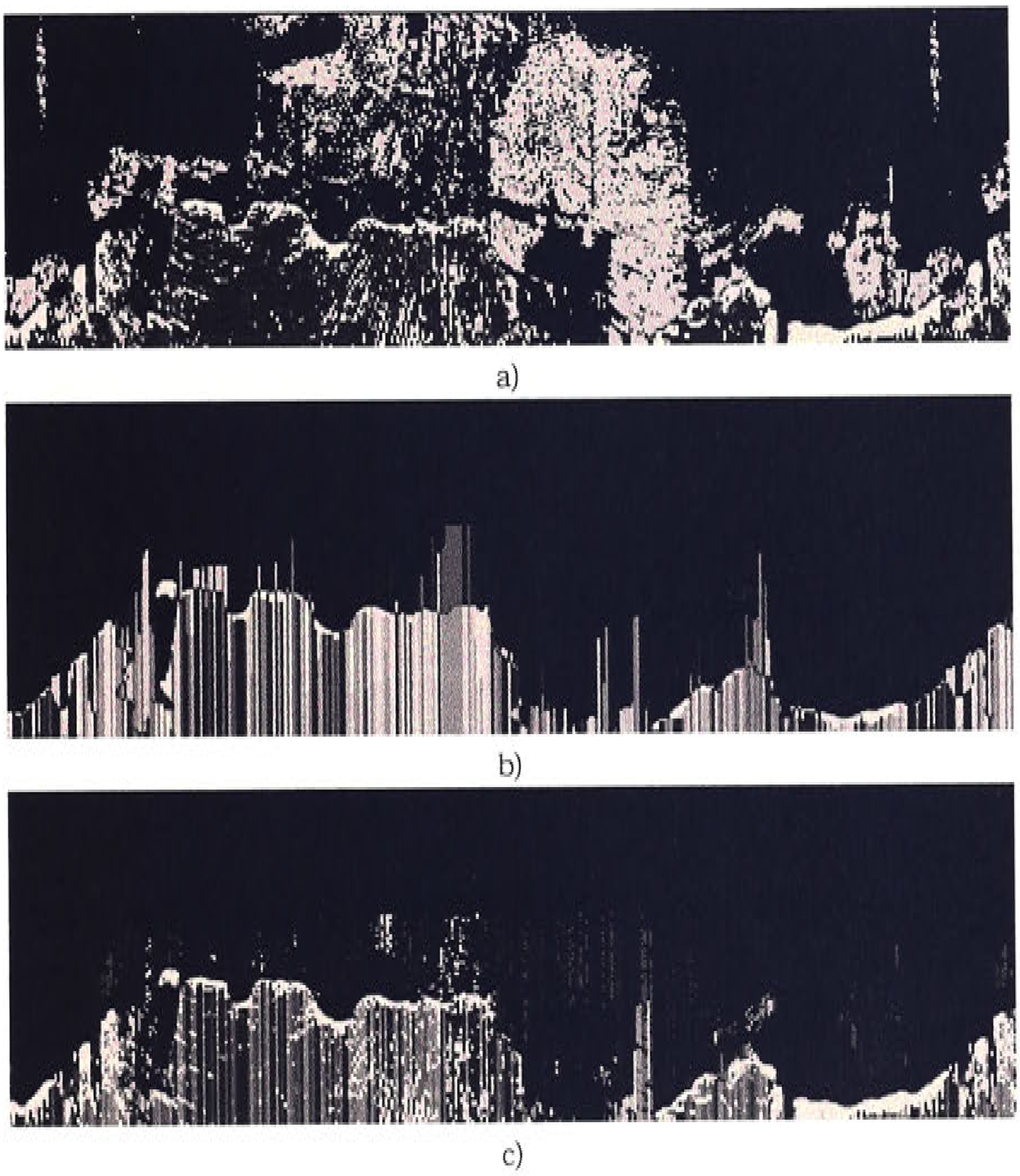


Figure 5.16: Image subtraction of carpet matching results with ground truth: occupancy grids showing subtraction of ground truth with the carpet matching results of a) colour space, b) gradient detection, and c) colour space and gradient detection

by:

$$\epsilon = \sum_{i=0}^{N-1} |G_i^{GT} - G_i^{R}| \tag{5.5}$$

where N is the number of cells in the grid. Table 5.2 presents the values of ϵ produced by applying Equation 5.5 to the occupancy grid results of the various methods of carpet matching discussed previously. From the data presented in the table, it can be seen that the combination of a colour matching and the gradient detection produces the best results, with the gradient detection method seemingly making the greater contribution. It is surprising that colour matching using the average pixel method and that of the colour space method produce such similar results. Remembering that the average pixel method is limited to conditions of constant illumination it is clear that colour space matching is superior.

Combining a colour based carpet matching approach with a heuristic rule limiting the extent of possible carpet regions in the local space produce the best estimates of free local space. In this case the preferred methods are colour space matching and gradient detection. It should still be clear that it is easy to imagine situations where carpet detection methods would fail, and as long as the system consistently mis-classifies the region in question there should not be be any significant affect on the current systems applicability to the localisation problem.

Local Space Detection under Occlusion

The above results have, for the most part, shown carpet detection in a static environment. Carpet detection estimates from sequences of images were combined over time to combat noise in the panoramic sensor, however the environment was essentially static during these sequences. Mobile robots on the other hand, should operate in the dynamic environment of the real world. Changes in the sensed environment can come from two sources (excluding sensor noise), that of objects that are visible moving within the environment, and that of motion of the mobile robot itself, causing the viewpoint of the sensor to change in respect to the environment.

When objects move about in the environment, the instantaneous view of local space changes to reflect the movement. For example when a person walks past the panoramic sensor the carpet is occluded from view for a brief period of time. A local space estimate taken during that time will result in non-carpet regions being identified in the image displaying the person. This detection of the moving object in the scene is necessary for obstacle avoidance but for robot localisation it produces a problem. The change in the perception of free local space due to dynamic objects occluding carpet regions could cause the system to mis-localise when revisiting that region of the environment. Therefore for the task of localisation it is preferable to maintain an estimate of local space which is insensitive to the presence of moving objects.

The promulgation of local space estimates through time using Bayesian probability

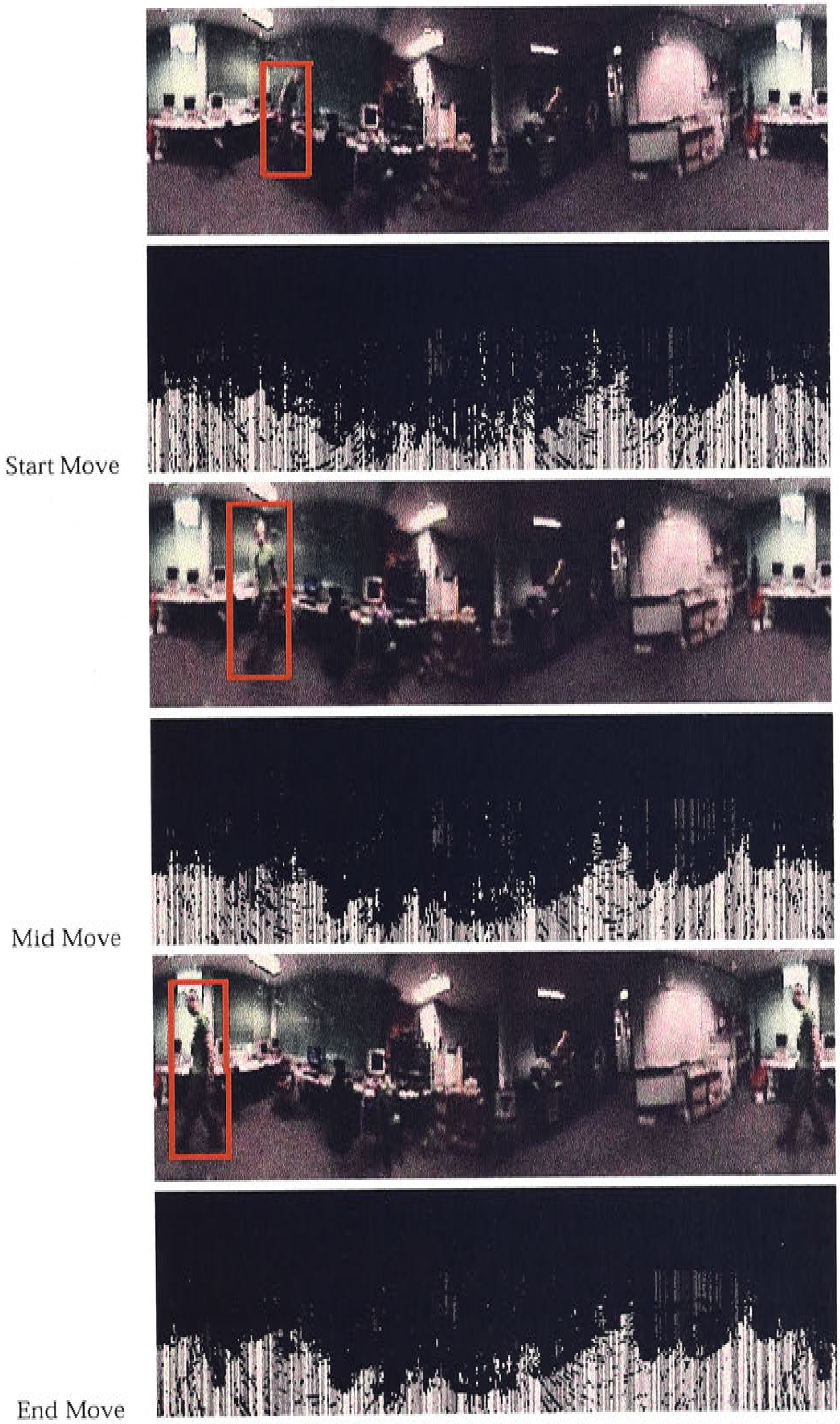


Figure 5.17: Occlusion in local space detection.

theory provides this insensitivity. Temporary observations of non-carpet regions due to moving objects need to overcome the previous observations of carpet regions before they begin to affect the occupancy grid. In this way a stable view of the local environment can be maintained in the presence of dynamic objects. Figure 5.17 shows the results of an experiment where a person walks past the robot. The local space occupancy grid is not dramatically affected. Three panoramic images and the associated occupancy grids are shown from the beginning middle and end of a sequence in which a person walks acrodd the field of view. The locations of the person in the panoramic images are highlighted in red. The middle image and grid show that the occupancy grid can be insensitive to the presence of moving objects although the bottom image and grid displays a noticeable change due to the moving object. This is because the gradient between the carpet and the object was strong in this region, causing a strong input into the occupancy grid. It is not clear if the local space occupancy grid will be insensitive to all object motion through the visual scene. Typically moving objects are quite small in relation to the size of the visual field and as such do not deform the view of local space dramatically. This sort of anomaly could be countered by introducing a saturation into the occupancy grid which additionally reinforces detected carpet regions over time, providing more protection from moving obstacles. Of course this protection does not help in the case when a section of the environment permanently changes causing prolonged exposure to contradicting sensor data, such as when a desk is moved. Permanent deformation in the environment would lead to inconsistent local space views over time. For robot localisation, this would require the acquisition of a new view of local space to replace the old view stored in the internal map representation.

Local Space Detection during Motion

The second cause of change to the local space surrounding a mobile robot is due to the robot's own movement. Detecting changes in local space due to ego-motion is desirable since it reflects movement from one area of the environment to another and provides valuable information to the task of localisation. Changes due to ego-motion can be detected using our system as the steady, constant motion typical of mobile robots produces a similarly steady and consistent change in the perception of local space. The

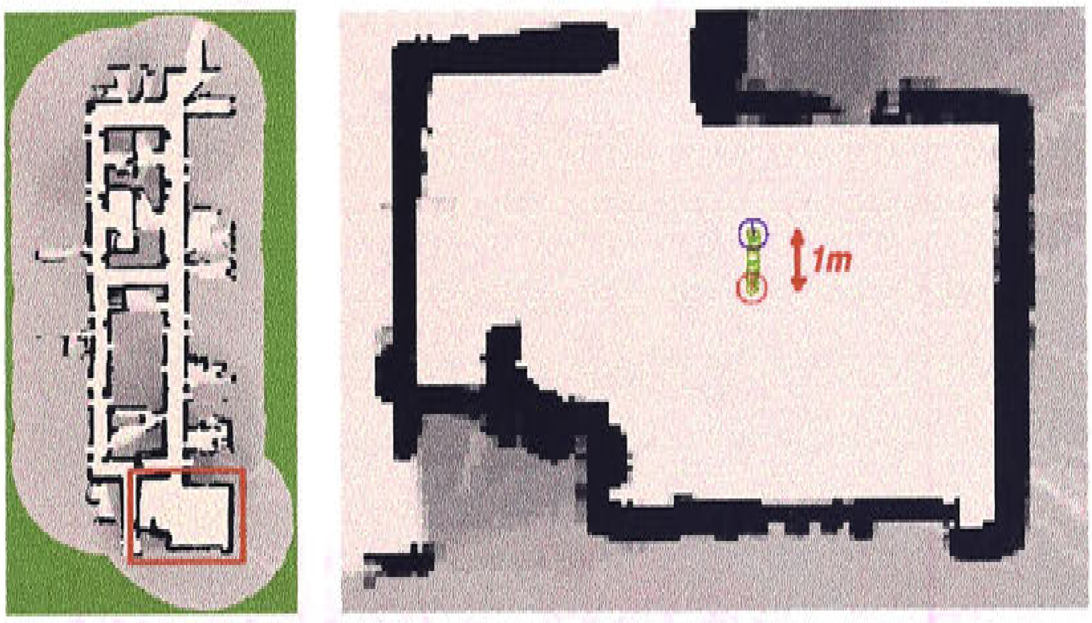
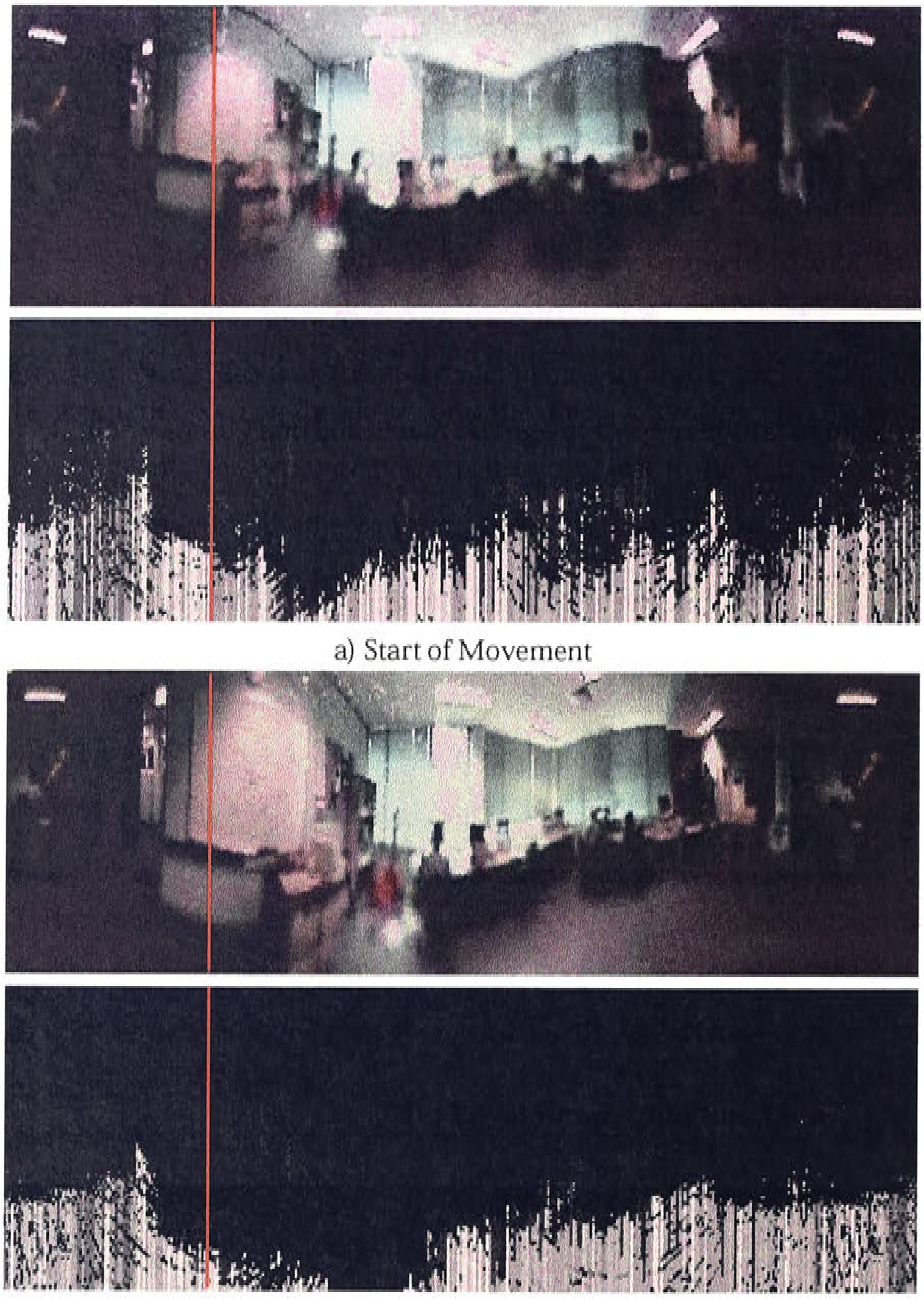


Figure 5.18: Path of robot during local space detection experiment

continual reinforcement of changing areas in the visual field produce lasting change in the occupancy grids.

Figure 5.18 displays the path of the mobile robot over a short movement through a large room, starting from the red circle in the center of the room and finishing up at the location denoted by the blue circle.The total distance travelled in this experiment was approximately 1 meters. During this movement panoramic images were captured and the extent of local space was estimated.

Figure 5.19 shows the panoramic images captured at the start (top) and finish (bottom) of the robot path as well as the associated local space occupancy grids. The direction of motion is indicated by the red line. It can be seen that the robot movement causes the carpet region in the left side of the panoramic view to contract as the robot approaches the wall. This is reflected in the local space results where a similar contraction is seen in the detected carpet regions of the occupancy grid. The difference between the two occupancy grids can be calculated by image subtraction. Figure 5.20 shows the results of subtracting the occupancy grid from the start of the move from the occupancy grid at the end of the move. The result is noisy but definite regions of difference are defined . Our approach to carpet detection therefore can produce occupancy grids which reflect the change in carpet regions due to robot ego-motion.



a) Start of Movement

b) End of Movement

Figure 5.19: Occupancy grids of local space at the beginning and end of short movement. The red line indicated the direction of movement.

5.2 Local Space Representation

The results in the previous sections demonstrate that the system is capable of producing a reasonable estimate of the extent of local free space surrounding the mobile robot. The estimate can handle fluctuations in the level of illumination, it can be pro-

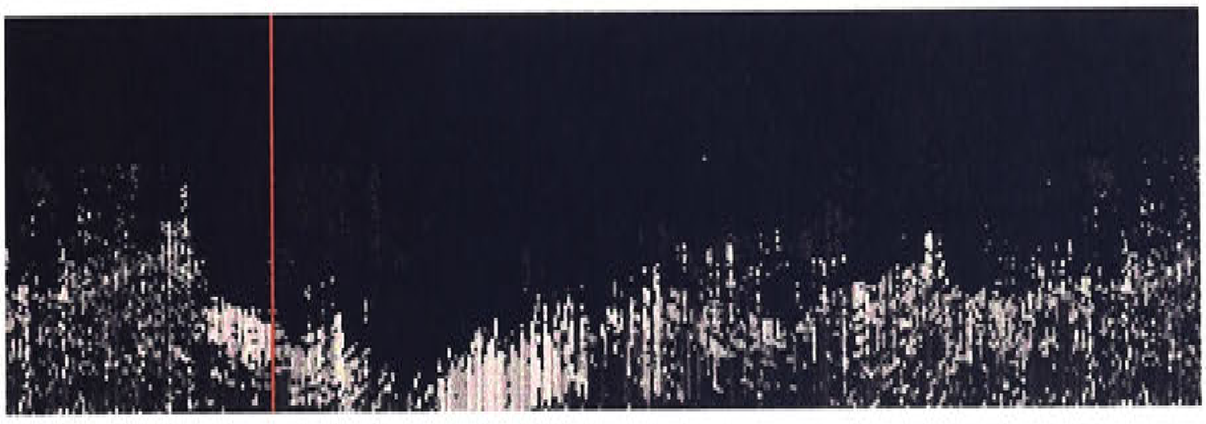


Figure 5.20: Subtraction of start and end move occupancy grids. The high intensity pixels show regions of difference.

mulgated through time, and can cope with dynamic objects moving through the visual scene and also with visual changes due to ego-motion. After producing such an estimate of local space, the question becomes how can this knowledge be used to aid the mapping and localisation tasks?

The desire to extract an estimate of the extent of local space surrounding a mobile robot comes from the need to form representations of places in the environment which can be used to constrain the global localisation search. To achieve this goal the representation of local space must first and foremost be simple enough to enable fast matching between instances of local space estimates. Of secondary importance is the degree to which individual instances are distinct from each other. This contrasts with the low level landmark representation where distinctness of landmarks is the most important concern and speed of matching was sacrificed. Indeed, when used in conjunction the two levels of representation can be used to quickly constrain the global localisation search to a subset of places within which the richer representation can be used to localise.

5.2.1 Histograms of Local Space

A simple way to represent the extent of local space surrounding a mobile robot is to use a histogram. The horizontal axis can represent discrete steps in the radial angle from the robot while the vertical axis can represent the extent of free space along the associated angle. By using a histogram, a one dimensional vector can be used to represent a particular local space profile, which dramatically simplifies the matching task.

Table 5.3: Building a Local Space Histogram from an Occupancy Grid

```
-- initialisation
$x\_step = 20$
$y\_step = 1$
-- iteration
for each column $x \times x\_step$ in Grid $G$
   $free\_space = TRUE$
   $y = 0$
   -- iterate rows from bottom upwards
   while $free\_space$ is $TRUE$ do
      $p = find\_ave\_pixel\_in\_cell(I, x, y, x\_step, y\_step)$
      if $p < threshold$ then
         $free\_space = FALSE$
      end
      $y = y + y\_step$
   end
   $histogram[x] = y - y\_step$
end
```

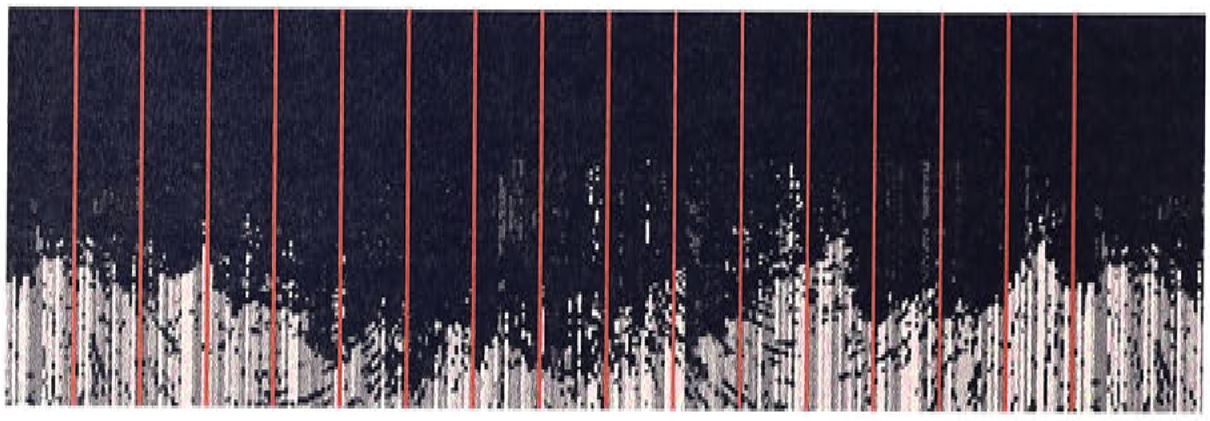


Figure 5.21: Occupancy grid divided into a 16 cell histogram.

An occupancy grid can be converted into a histogram by detecting continuous columns of free space. Table 5.3 gives the algorithm which fits a histogram to a particular instance of a local space occupancy grid. The horizontal (x) axis of the occupancy grid is divided into a number of discrete columns of size $size_x$. The intensity of pixel regions within each column x is then checked starting from the bottom of the grid ($y = 0$) and proceeding upwards with y increasing by steps of $size_y$. If the average intensity of these regions within a particular column falls below a threshold then the magnitude of the local space histogram at cell x is set to the value of y. The resulting histogram then represents a particular profile of local space. In our system the histogram has 16 cells, each of which condenses the pixel intensity from 20 pixel columns in the occupancy grid. The division of the occupancy grid into a 16 cell histogram is shown in

Figure 5.21.

Figure 5.22 shows two examples of local space histograms calculated from occupancy grids. It can be seen that the histogram makes a good fit with the occupancy grid. Although part b) shows that if the occupancy grid is wrong, or there are regions of high uncertainty, the histogram will also be inaccurate. As long as the local space is mis-identified consistently, this will have no affect on the using local space information in localisation tasks. When used to represent a particular place in the environment, the histograms formed by the method described, are said to capture the places local space profile.

The stages in local space detection are shown in Figure 5.23. The final form of representation for the local space profile is the one dimensional vector of size 16 containing the local space histogram.

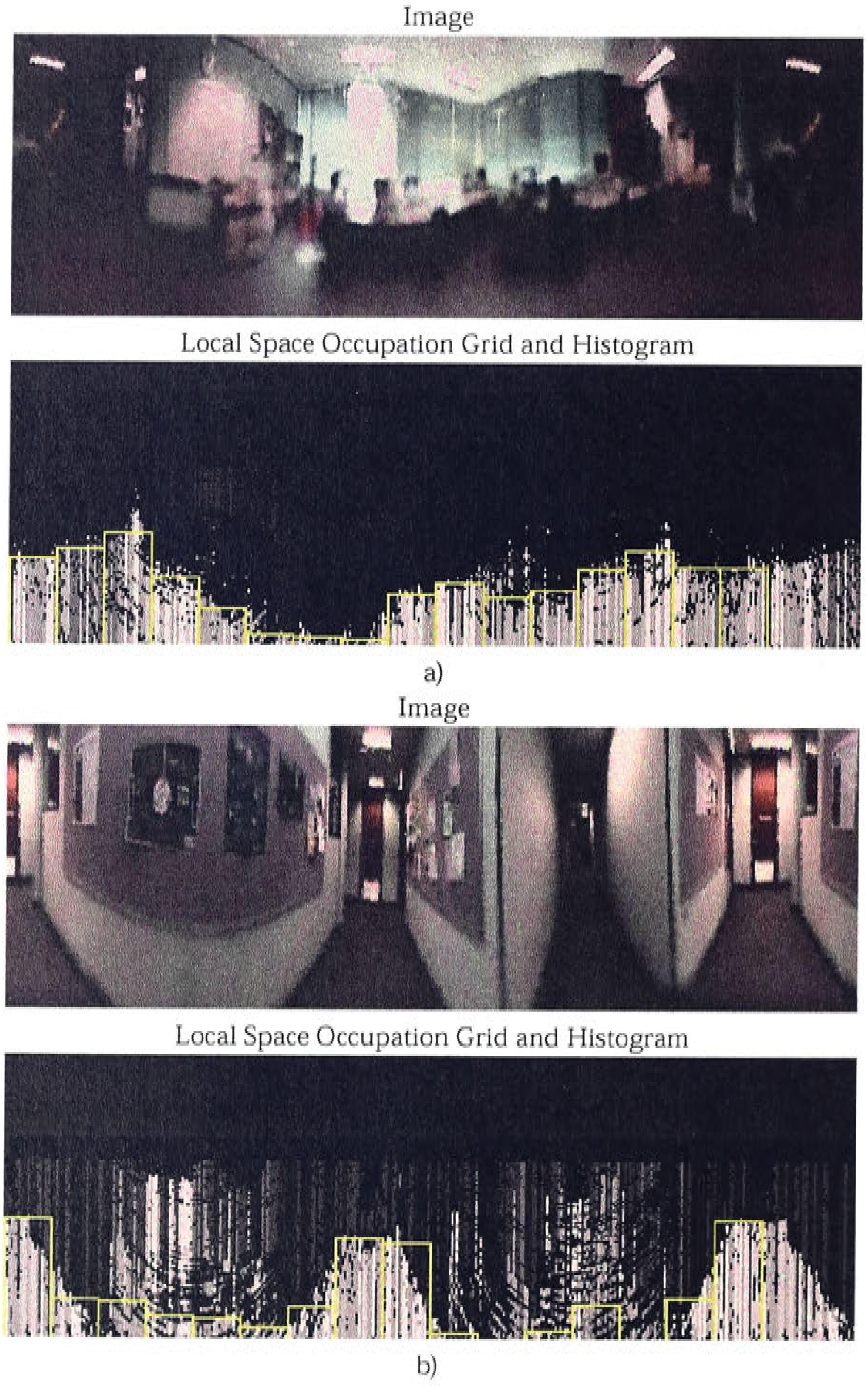
5.2.2 Local Space Matching

In order to use the information contained in local space histograms for mobile robot localisation it is necessary to compare two instances of histograms and produce a measure of their similarity. This is the equivalent of the matching task between landmark templates and panoramic images. However in this case the matching is between two one dimensional vectors.

A Sum of Absolute Differences (SAD) can be used to produce a measure of similarity between two vectors. It must be remembered, however, that the cells of the local space histogram represent the radial angles of the robot. Therefore the rotation of the robot will affect the profile of the histogram, meaning two views of local space captured from the same position in the environment but from a different orientation would result in a shift in the local space histogram.

In using knowledge of local space to constrain a localisation search it is beneficial if the matching process is invariant to orientation. Two local space views from environments with similar free space could then be matched irrespective of the orientation from which the two views were captured.

Figure 5.24 shows how rotation invariance in the matching process can be achieved. A SAD calculation is performed on every possible orientation of one of the histograms



Image

Local Space Occupation Grid and Histogram

a)

Image

Local Space Occupation Grid and Histogram

b)

Figure 5.22: Local space histograms

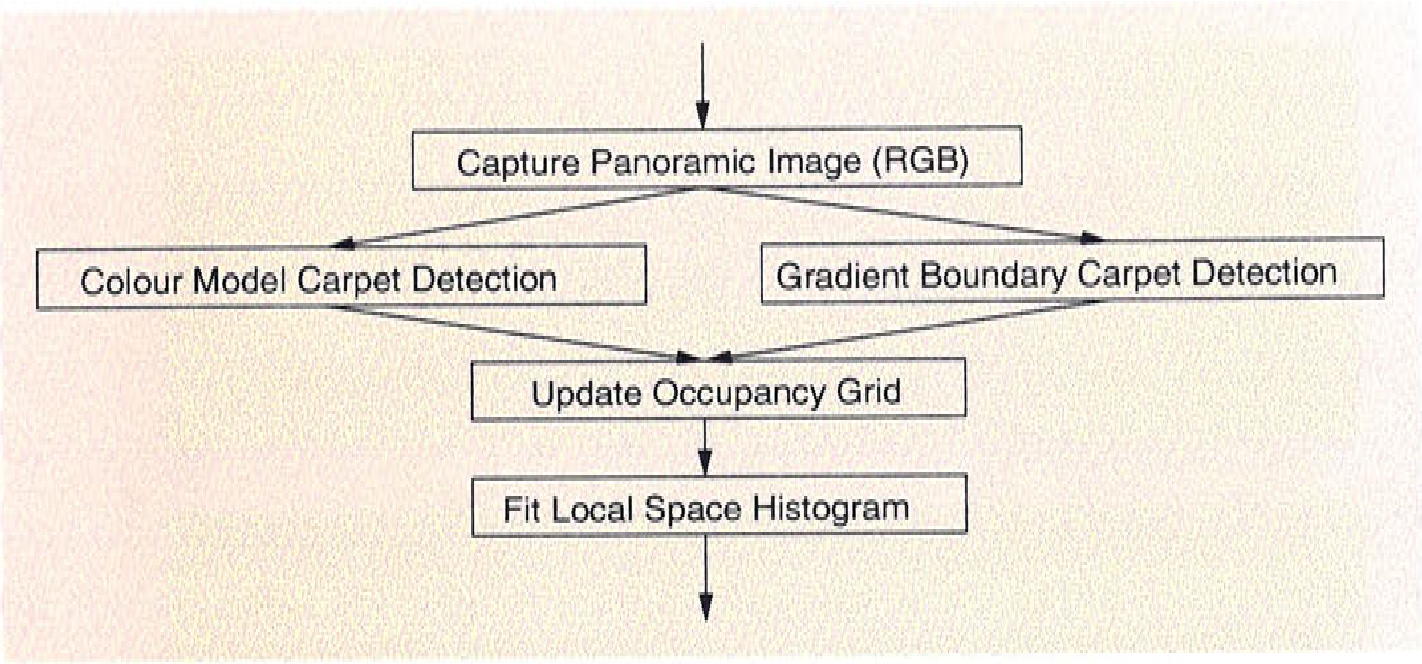

Capture Panoramic Image (RGB)
Colour Model Carpet Detection
Gradient Boundary Carpet Detection
Update Occupancy Grid
Fit Local Space Histogram

Figure 5.23: Stages in extraction of a local space profile.

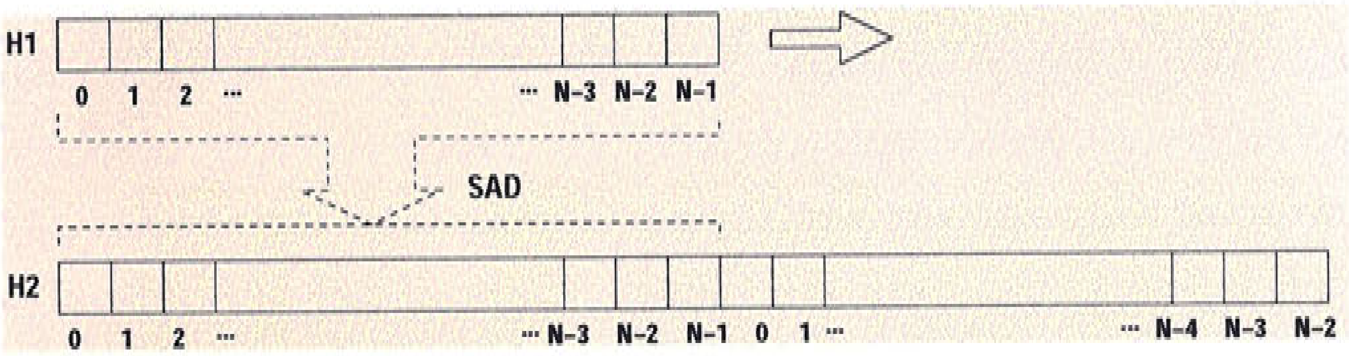


Figure 5.24: Rotation invariant matching of local space histograms

with respect to the other. Given two histograms H^1, H^2, this is achieved by constructing a vector of size $2N-1$ by repeating H^2 such that $H^2_i = H^2_{i+N}, \forall i < N$. This repeated histogram H^2 can then used to produce vectors of size N of the form:

$$\overline{v}_0\{0\ldots N-1\}, \overline{v}_1\{1\ldots N\}\ldots\overline{v}_{N-2}\{N-2\ldots 2N-3\}, \overline{v}_{N-1}\{N-1\ldots 2N-2\} \tag{5.6}$$

which represent all possible orientations of H^2, where $\overline{v}_i\{i\ldots j\}$ denotes the vector of size N from H^2_i to H^2_j.

A measure c which quantifies the correlation between H^1 and H^2 can then be computed:

$$c = 1 - \frac{min_{i=0}^{N-1}(SAD(H^1, \overline{v}_i\{i\ldots i+N\}))}{MAX * N} \tag{5.7}$$

where MAX is the maximum value possible for cells in the histogram, denoting that

Table 5.4: Local space histogram matching results

Match Local Space Profiles	Result
1 & 2	0.91
1 & 3	0.78
2 & 3c)	0.76

carpet has been detected up to the image horizon.

Using this matching method different instances of local space histograms can be compared. Figures 5.25 and 5.26 shows three panoramic images together with their extracted occupancy grids and local space histograms. The panoramic images and histograms shown in a) and b) are views captured from roughly the same position in space but view b) was captured after rotating the robot 180 degrees. View c) is from a position three meters away.

Each of the local space histograms presented in Figure 5.25 were matched with each other used the matching technique described above. The results from this matching are shown in Table 5.4. From this table it can be seen that the local space histograms from views a) and b) correlate very well, whereas matching views a) or b) with view c) does not produce such a high correlation measure. This result shows the ability of the local space histogram matching process to successfully discriminate between like and unlike views as well as demonstrating the rotation invariant property of the matching process.

5.3 Local Space Detection and Localisation

In our research we can construct profiles of local space and can compare these profiles using the matching method previously described. In order to use these ideas to simplify the global localisation task it is necessary to categorise the profiles. If the local space profiles associated with places in a topological map and a current view of local space can be categorised in groups, then the computation resources needed for brute force landmark set matching can be targeted to appropriate places. In this local space profiles can be used to constrain the global localisation search space.

To illustrate this concept Figure 5.27 shows a mobile robot path through an office envi-

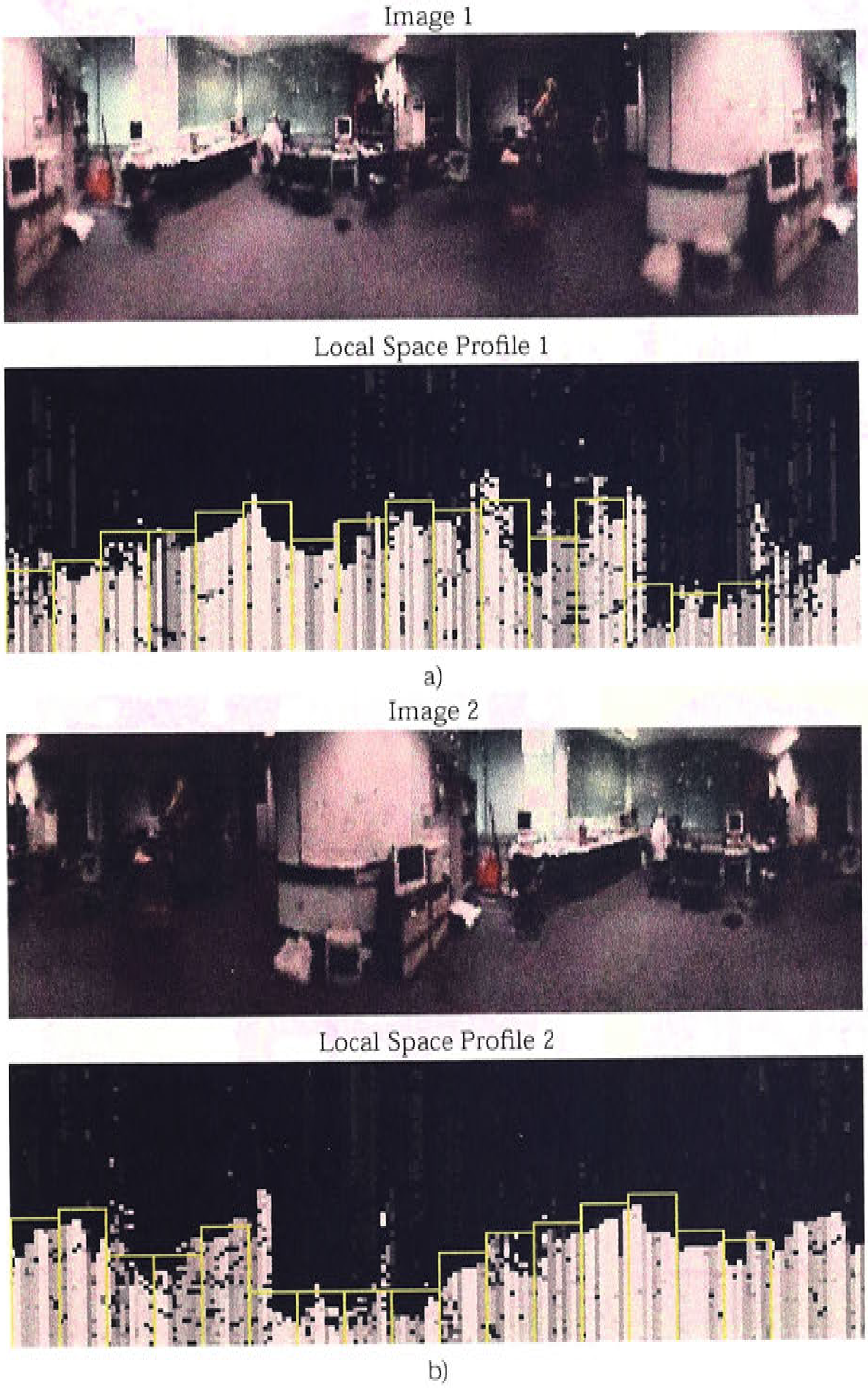


Image 1

Local Space Profile 1

a)

Image 2

Local Space Profile 2

b)

Figure 5.25: Example panoramic images and local space histograms. Views a) and b) were captured at the same position in the environment but are oriented 180 degrees from one another.

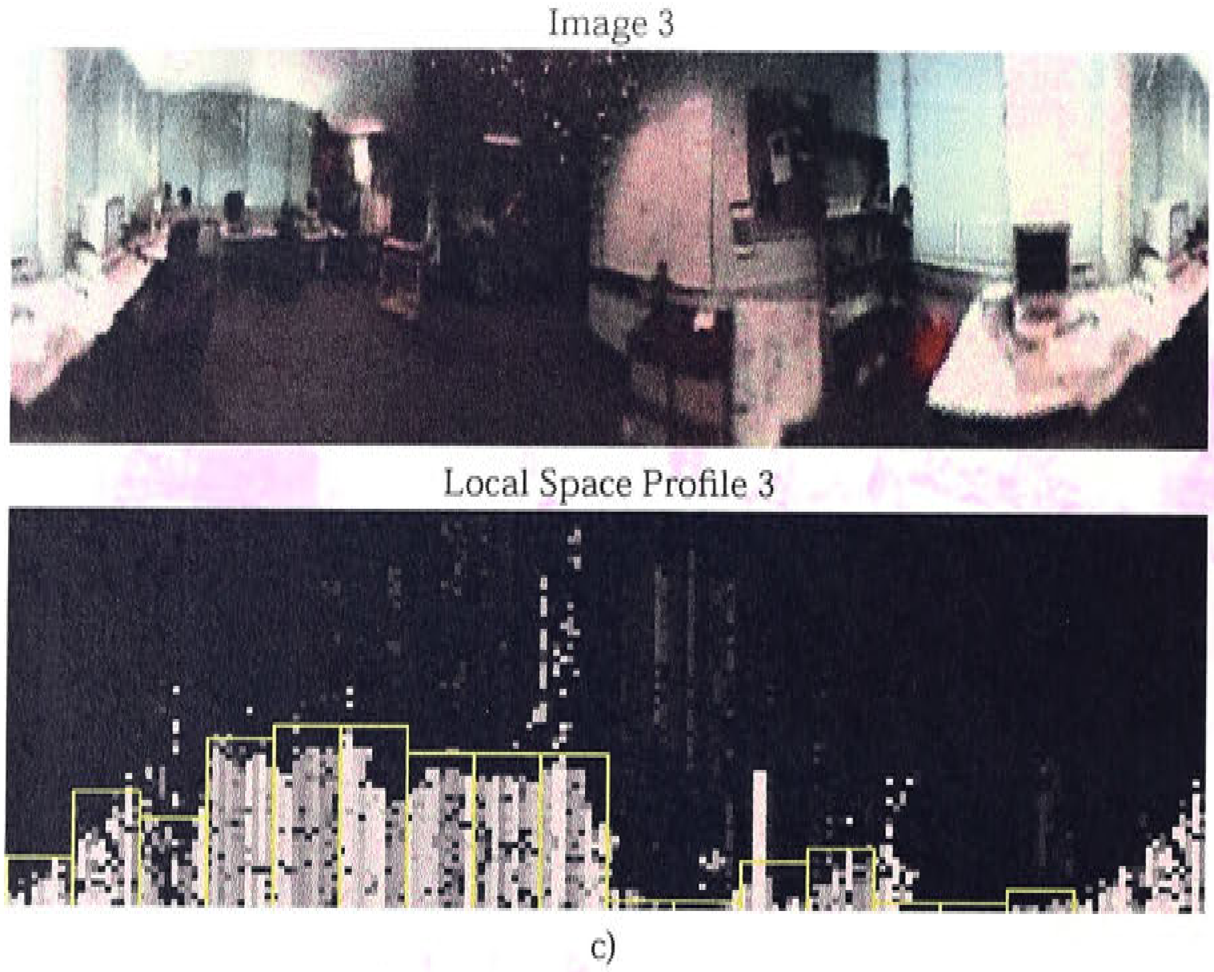


Figure 5.26: Example panoramic images and local space histograms. Image c) is from a location three meters away from those in Figure 5.25

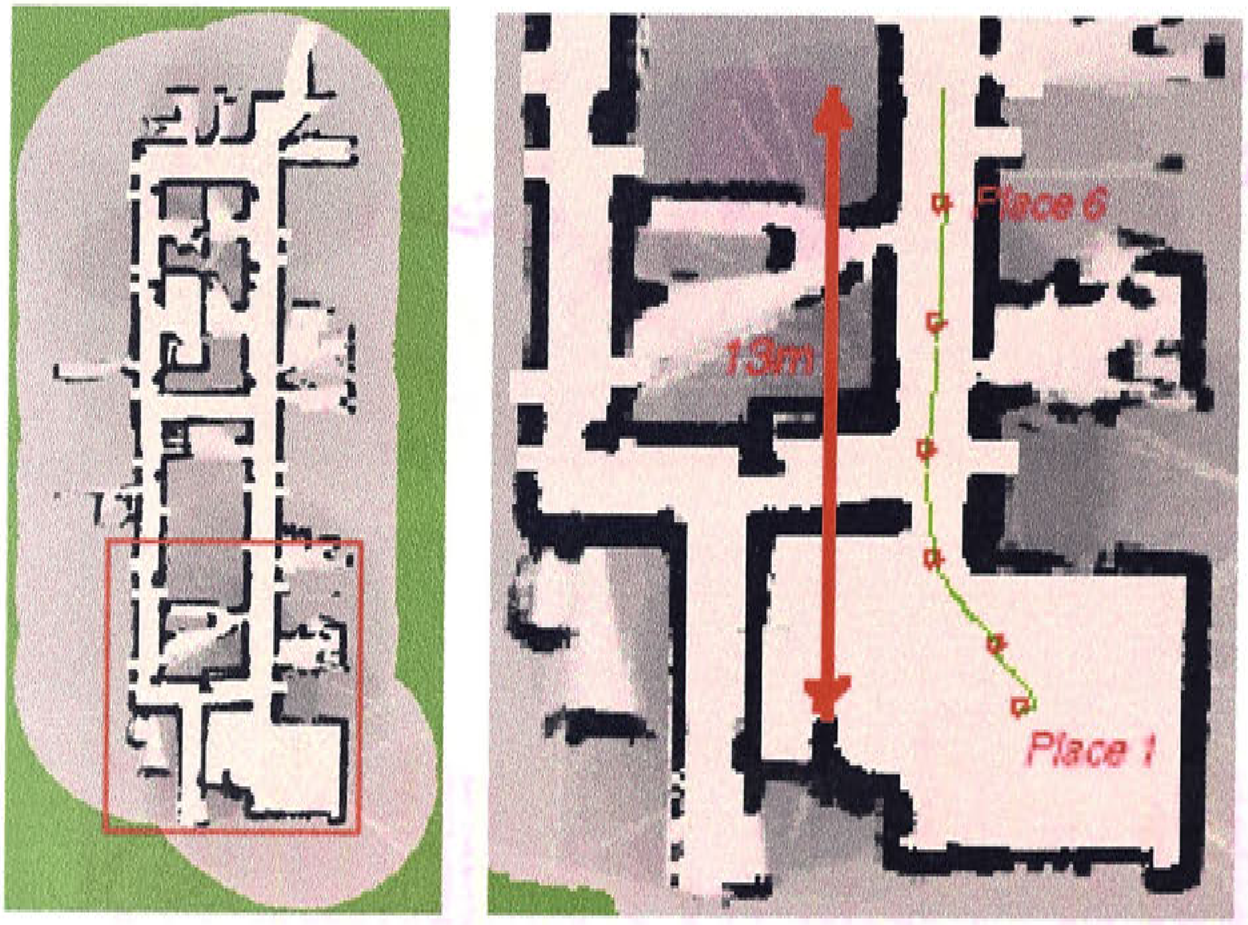


Figure 5.27: Map showing the nine places in the environment where local space profiles were obtained.

ronment. The path is displayed as a green line and the position of the robot is shown at some points along the path by a red circle. The robot started in the large room at the bottom of the map and travelled up the corridor at $20m/s$. Throughout this movement panoramic images were captured and every tenth frame was logged, resulting in an image sequence of 300 images. The local space profiles associated with the image sequence were extracted using the techniques we have developed. A video of this image sequence and the extracted local space profiles is included on the CD-ROM which accompanies this thesis.

Six places were manually chosen at 50 image intervals along the path, with the local space profile detected at each place forming a topological map of the robot's path. The positions of these places are displayed as red circles in the map. The panoramic images and local space profiles of each of the six places are shown in Figures 5.28 and 5.29. It can be seen from the map that a number of the identified places would contain views of local space which are quite similar, especially in the corridor section of the robot's path. If a robot retracing the route perceived a current local space profile which was similar to a corridor, the subsequent localisation search could be limited to regions in the map which shared a similar local space profile. In fact the structured environment of the typical office building, there is a limited set of local space profiles which might be encountered.

5.3.1 Local Space Primitives

A set of commonly seen configurations of local space can be defined as the local space primitives for a particular environment. These primitives can then help to classify sections of the environment and improve the efficiency of the global localisation task. Specifying the set of local space profiles that should constitute the set of primitives requires consideration. In general the chosen primitives should divide the set of places which form the topological map in such a way as to minimise the average localisation search time. This means that there should be an equal distribution of the number of places associated to each local space primitive, and that the number of local space primitives is sufficiently low so that the computation saved by reducing the search low-level search space does not exceed the extra cost associated with matching the current local space profile with the primitive set. In this research a set of primitives

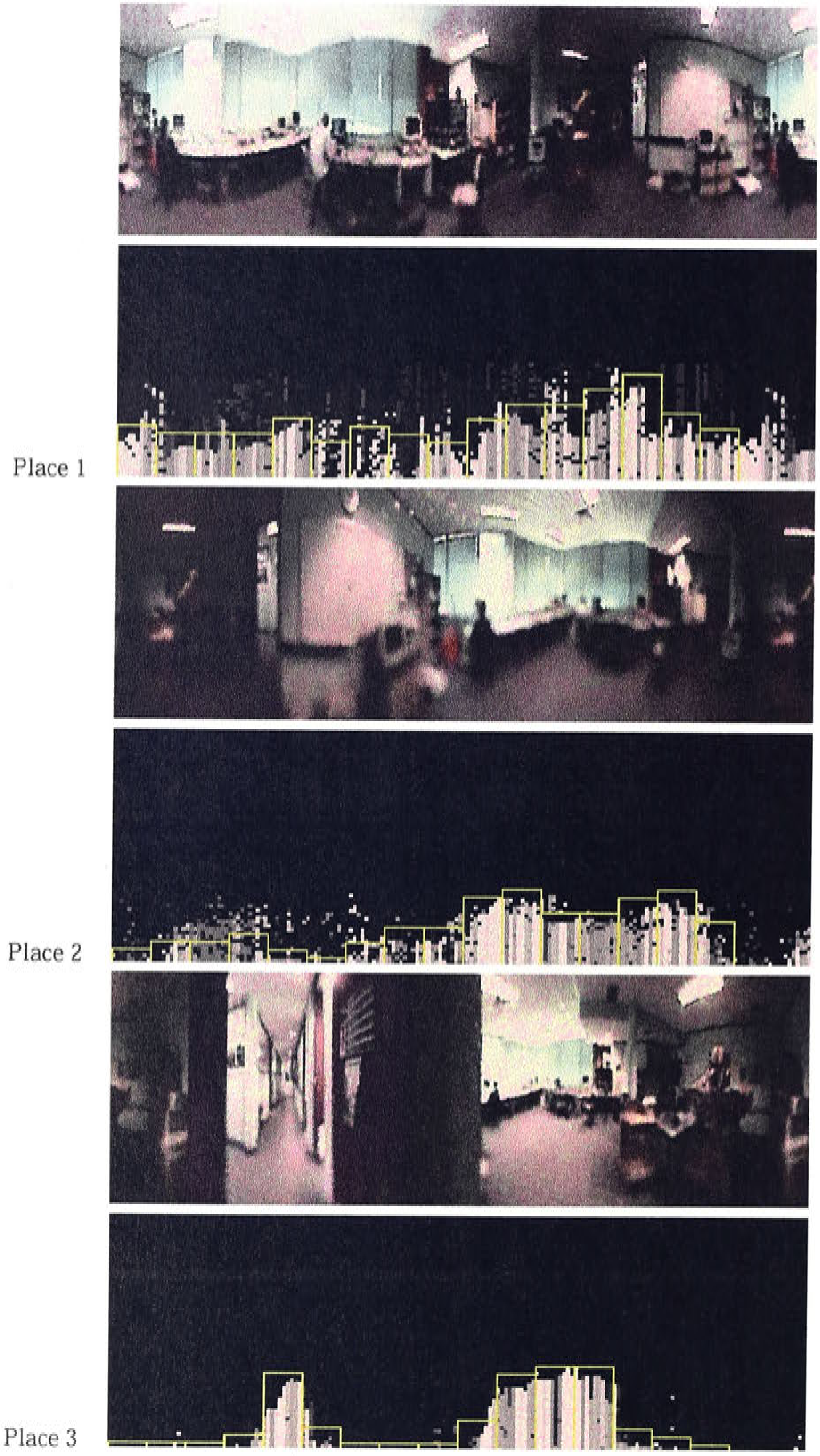


Figure 5.28: Local space profiles of places 1, 2 and 3

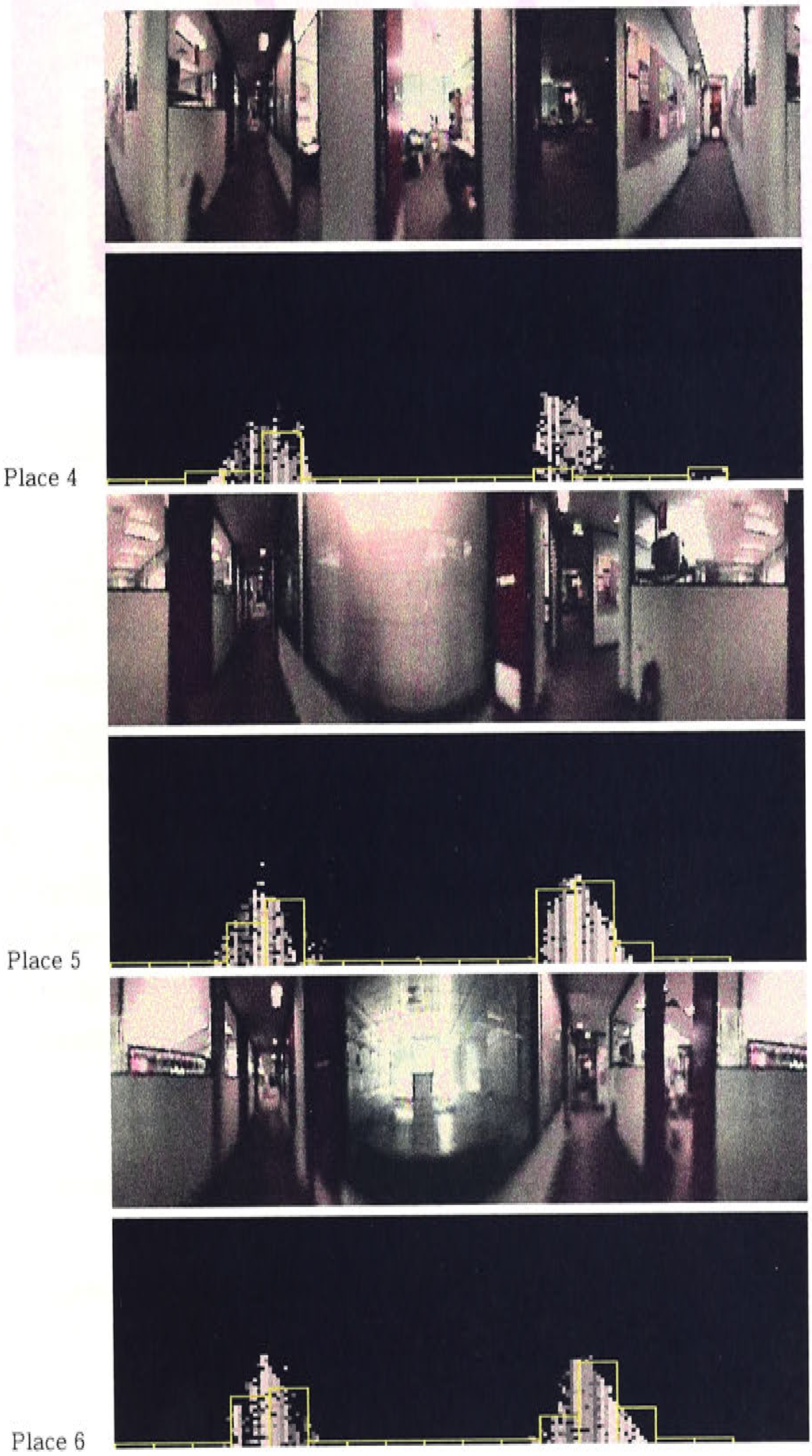


Figure 5.29: Local space profiles of places 4, 5 and 6

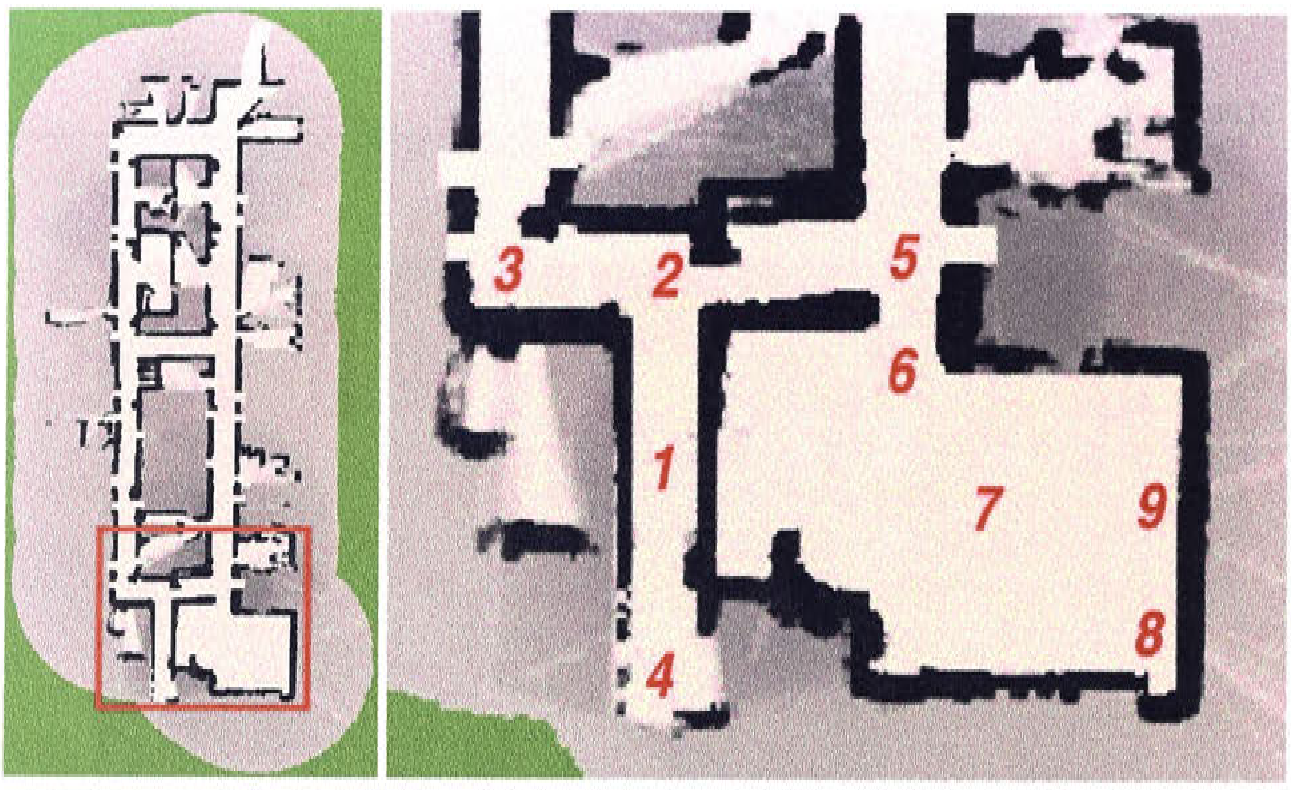


Figure 5.30: Regions of different local space: 1) corridor, 2) T-intersection, 3) turn, 4), dead-end, 5) cross roads, 6) doorway, 7) open space, 8) corner, and 9) wall.

was chosen by hand as the environment of predominantly structured corridors and rooms provided strict delimitation between possible local space profiles. The different types of local space in this environment is illustrated in Figure 5.30. In general it would be beneficial for the system to learn and maintain a set of local space primitives based on the criteria mentioned above and its perception of the explored environment.

The set of local space primitives used in this system corresponds to the areas shown in Figure 5.30 and are shown in Figure 5.31. It includes local space profiles common to every office environment. While human distinctions between such areas as inside a small office and at a dead and of a corridor (areas labelled number 4 in figures) might be lost by this level of discretisation and the choice of representation (eg only open space), it does provide a useful division of the search space.

5.3.2 Matching Primitives

Local space primitives can then be used to categorise views of local space that are perceived in the environment. A set of histograms representing the primitives identified above can be matched against the views of local space extracted from image sequences in an attempt to categorise the environment through which the robot moves. This

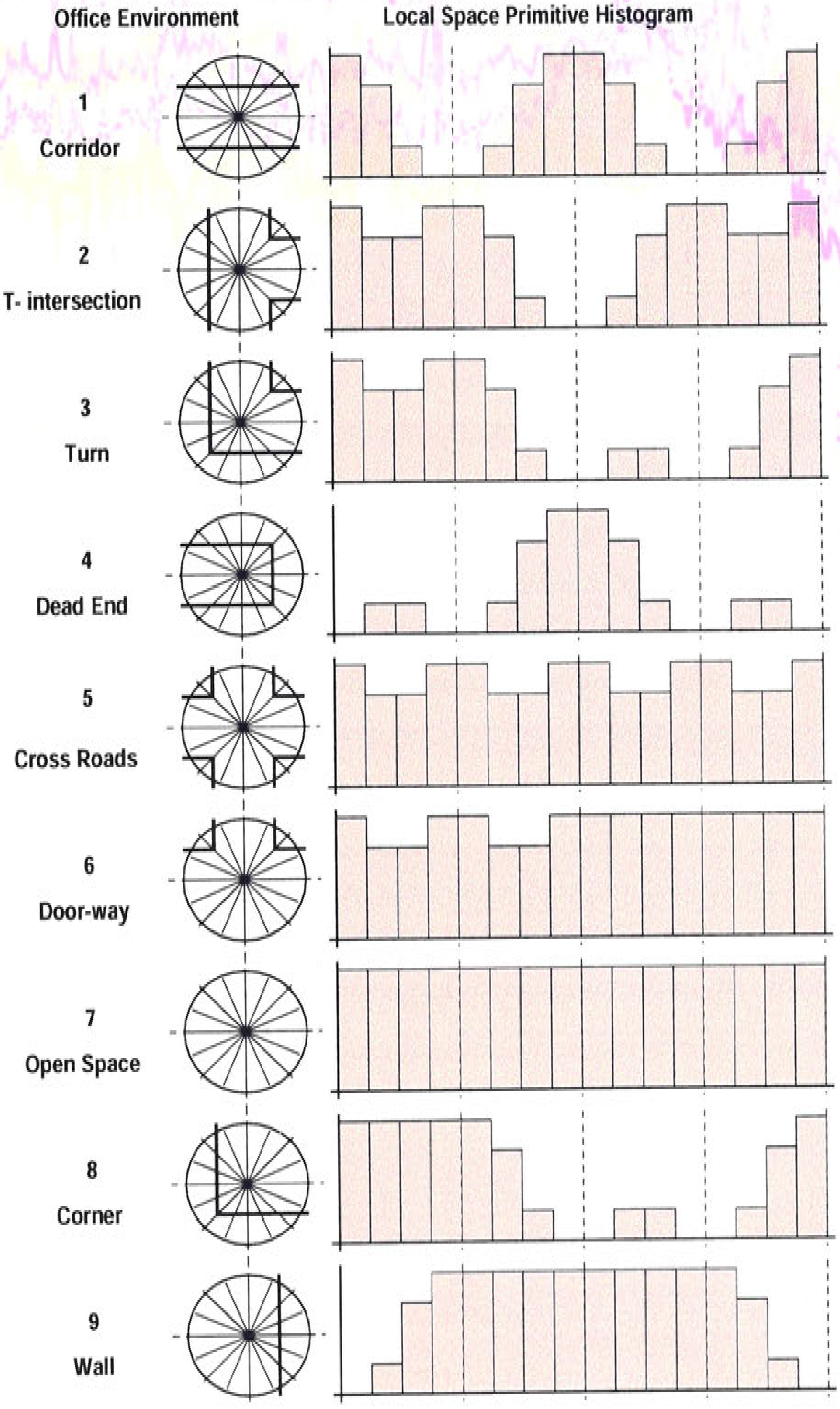


Figure 5.31: An example set of local space primitives from an office environment (Figure 5.30).

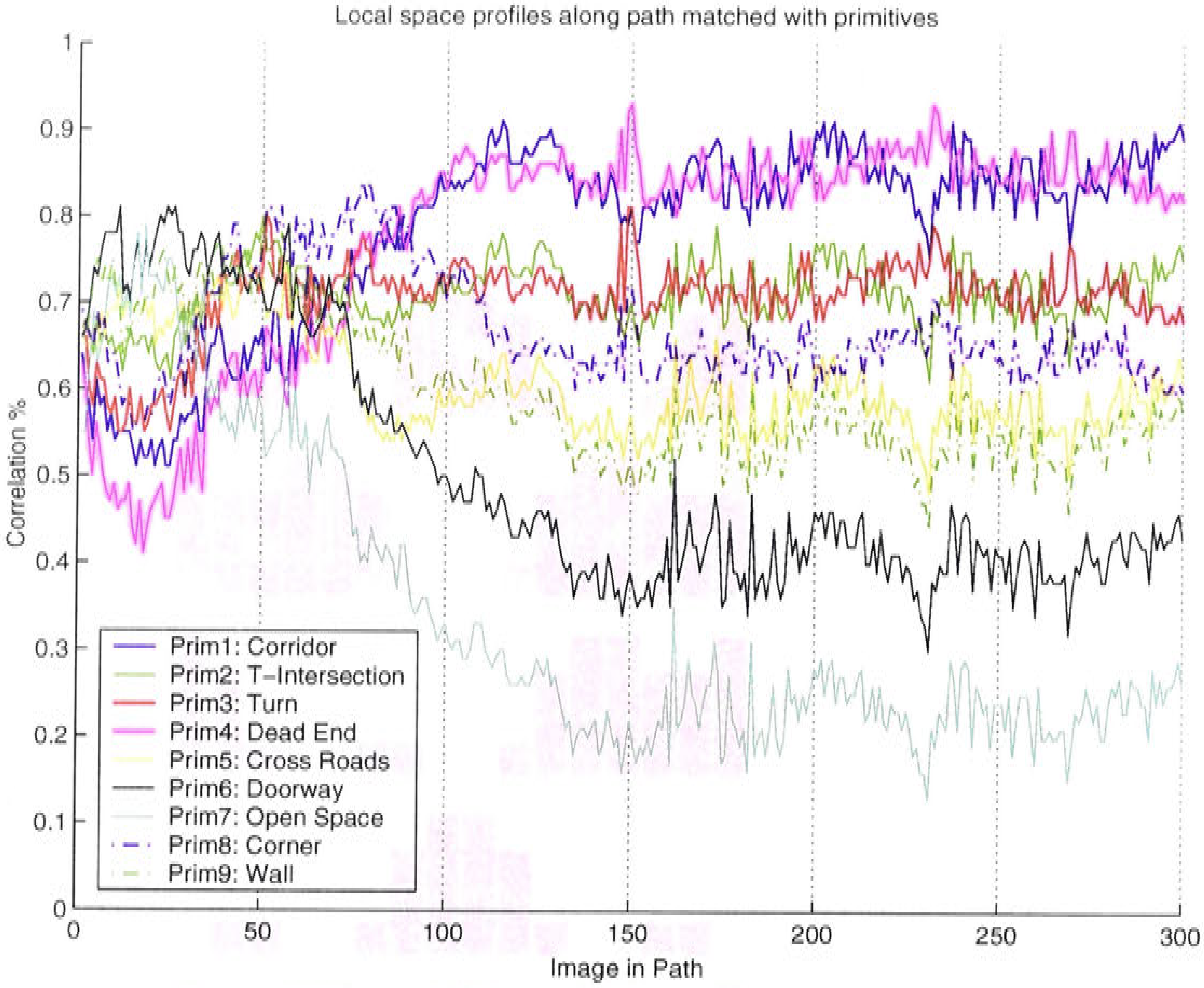


Figure 5.32: Local space profiles extracted from images along a path matched with local space primitives.

matching can be performed using the technique introduced earlier for local space profiles except that the profile to be matched needs to be normalised. This process resizes the magnitude of cells in the histogram to be matched to be consistent with those in the primitives.

Figure 5.32 shows the results of matching the local space profiles extracted from the robot path shown in Figure 5.27 and a subset of primitives identified above. The graph shows the correlation results for matching the 300 images in the sequence with primitives representing the identified categories in the environment. The vertical black dotted lines represent where the chosen places along the path occur in the image sequence.

From this figure it can be seen that while the sections of the path that correspond to corridor regions match highly with the corridor primitive, the results from the section of path in the large room are not so clear. Both the open space primitive and the doorway primitive are matched well here and no definitive result can be identified. This is due to the cluttered nature of the room and the noise in the local space detection process. It is difficult for a discretised approach such as using local space primitives to recover from such problems. It can also be problematic when a place is located on the boundary between two primitives. For example when does a T-intersection become a corridor?

A reduction in the number of primitives helps to overcome these problems. However this reduces the benefits that such a representation brings to the localisation task. Changing the nature of the representation itself could also improve primitive matching results. For example local space could be represented simply by one value through summing the magnitude of cells in the local space histogram. This value can be discretised to categorise the views of local space. This approach , while allowing for noise in local space measurements, would suffer from the discretisation of the categories not being reflected in the system perception of local space.

A clustering approach, where the system forms categories of local space itself would be the best solution to these problems, by attempting to form categories of local space profiles based on real differences in the perception of local space from past experiences. This solution though is beyond the scope of this thesis and is only mentioned as a consideration for further work.

5.3.3 Matching Places

One method of categorisation which truly represents the sensor data and is easy to implement is to simply make each view of local space its own category. That is each place in the topological map has its own profile of local space and no explicit attempt to categorise them further is pursued. This has the benefit of each place retaining the distinctive features of its own profile and subsequently the discretisation problem does not occur. Of course this approach does mean that in the matching process, the entire set of local space profiles which represent places in the topological map must be compared with the current view of local space. Places which correlate well with the current view can then be used to perform localisation using the low-level representation. This approach is similar to that of Matsumoto et al. (1997) who matched entire

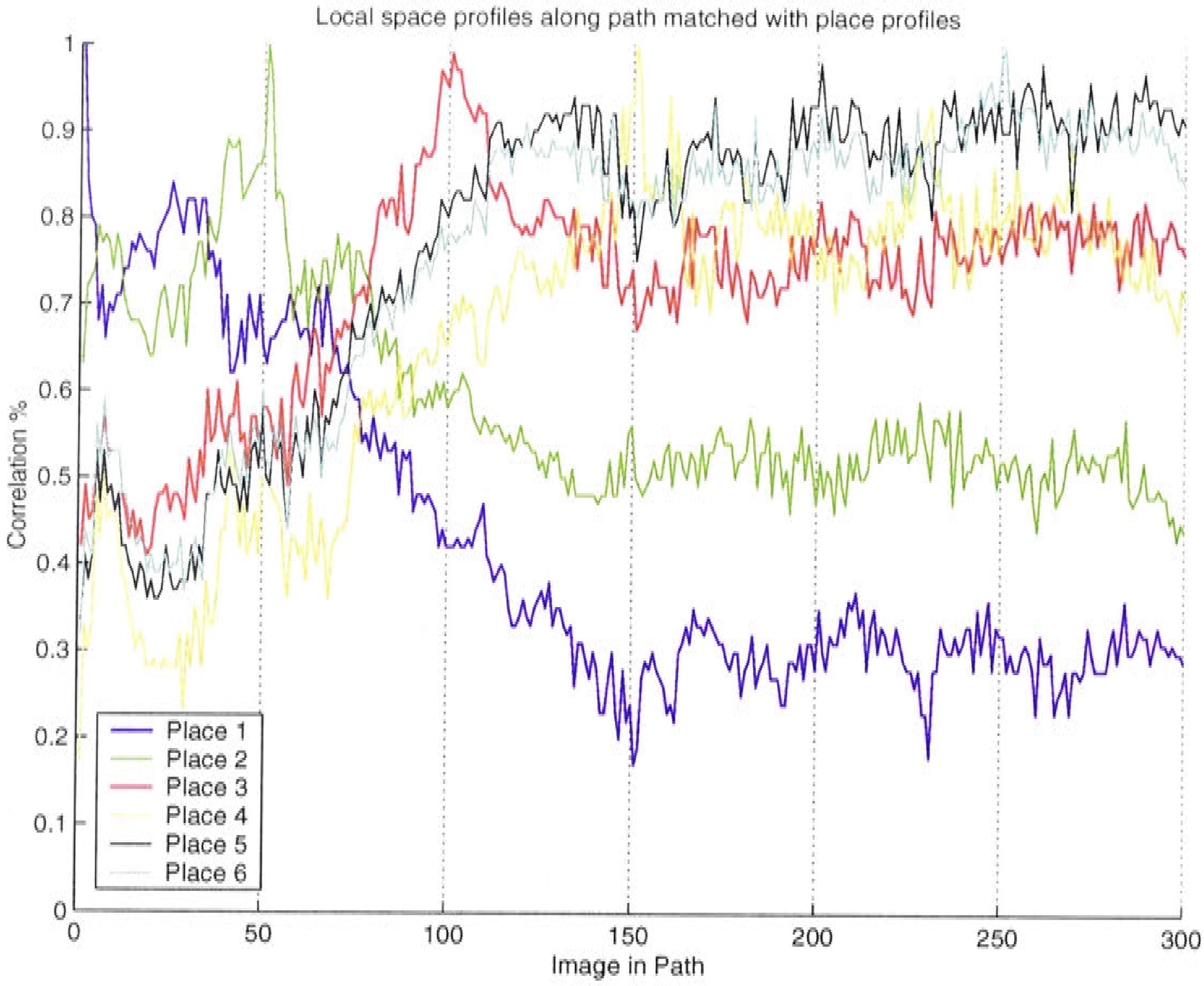


Figure 5.33: Local space profiles extracted from images along a path matched with the profiles of the places identified in Figure 5.27.

panoramic images to perform localisation in a sequence of captured panoramic views.

Figure 5.33 shows the results of matching the local space profiles of places in the topological map with those obtained from the image sequence captured during the robot path. The correlation results for the six places in the path are displayed. Again the dotted black vertical lines show where the places are located in the 300 image sequence.

As with the matching results for primitives, the first thing that becomes apparent from the plot is that the local space profiles from the corridor section of the environment are distinct from the rest and are consistently similar with each other. Unlike the primitive results, the section of the path where the robot was inside the room shows distinct peaks and it could be imagined that some sort of discrimination between local space profiles could be made. Also, the plot of results for each place peaks at the image in the sequence from where the profile for that place was extracted. This could mean that the local space profiles could limit the localisation search down to one place in the topological map. This optimism should be tempered by noting the fall-off of correlation results surrounding the peaks. This shows that the the defining features of the local space profiles which produce the peaks disappear quickly as the robot leaves the reference position from where the place was captured. This phenomenon is particularly evident in the plot of the first place in the path which peaks at the first image. The dramatic decrease in correlation performance after the peak in this case can be attributed to the image being the first captured, and therefore having an occupancy grid which is not as certain and more susceptible to change than those from images further down the path.

5.3.4 Computation Constraints

The advantages of abandoning attempts at categorisation must be balanced against the computation costs of performing one matching calculation for every place in the topological map. The purpose of having multiple levels of spatial representation was so that computationally cheap representations could provide approximate localisation estimates for more expensive levels. Thus the computation costs of forming a local space representation of the environment should not outweigh the computation savings made by the constraints it imposes on the low-level localisation search.

Figure 5.34 shows the computation time involved in matching primitives compared to that of matching places. The dotted red line represents the time in milli-seconds that the system takes to match a set of nine primitives with an instance of a local space profile as a function of the number of places in the topological map. The solid blue line shows the computation time for matching an instance of a local space profile with the set of profiles associated with the places in the map. It can be seen that the computation time for matching primitives is constant regardless of the size of the map, whereas the computation time for matching places increase linearly with the size of the map.

Although the plots of computation time would suggest that the system should use the local space primitives to categorise local space profiles, it should be noted that the time scale of these computations is relatively small when compared to the expense of other parts of the system. A plot of the time taken to capture and unwarp a panoramic

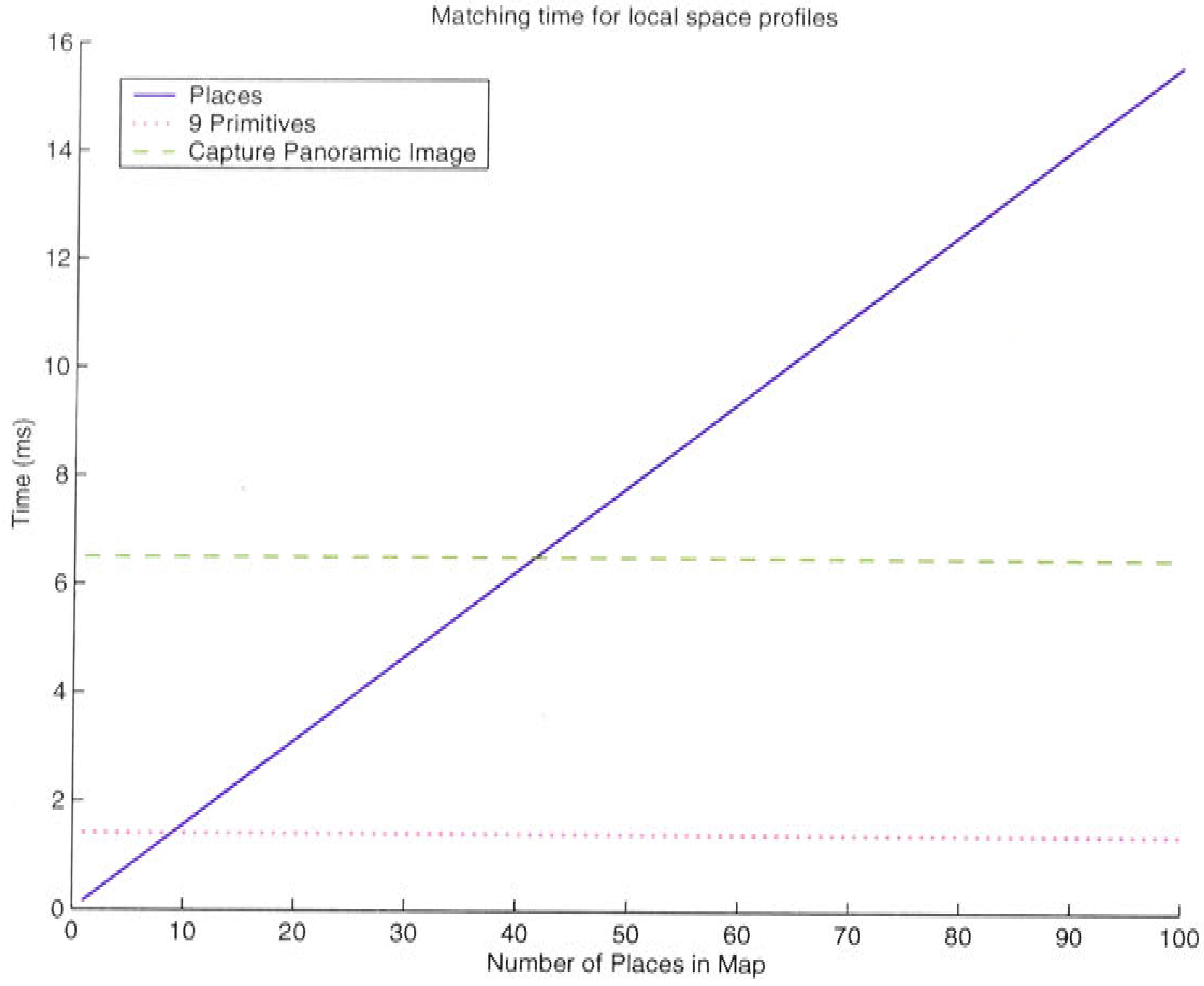


Figure 5.34: Time taken for matching local space profiles against the number of places in a topological map.

image is included in the figure to give a relative scale to the time measurements. We can see that the map must contain at least 40 places before the time taken for matching places exceeds that of capturing and unwarping the panoramic image. It should be remembered that the time taken to perform a brute force landmark template search on a panoramic image is approximately $700ms$. Given that matching with places allows for greater discrimination between places in the topological map and therefore can further restrict the localisation search, the extra cost in local space matching computation time that is accrued will be made up during landmark localisation.

In this research local space profiles are used to restrict the global localisation search by storing individual profiles for each place in the topological map.

5.4 Summary

This chapter has presented a mid level representation of spatial knowledge in the form of local space profiles. Implementation details were provided and discussed with a view towards using such a system for constraining the global localisation search space for lower levels of spatial representation. In particular the following key methods and ideas were developed:

- *Ground Plane Detection in Panoramic Images:* detecting the ground plane using carpet detection in panoramic images is a difficult problem due to the poor resolution inherent in the panoramic sensor. The carpet region detection methods of average carpet colour pixel matching, carpet colour space model matching and gradient boundary detection were implemented and examples of applying them to panoramic images were presented.

- *Occupancy Grids of Local Space:* detected carpet regions can be represented as occupancy grids which provide a probabilistic measure of the presence of carpet in the panoramic image. Instances of occupancy grids are then combined using Bayesian probability theory and provide a mechanism for accumulating local space knowledge over time in addition to fusing the results of different methods. Using a ground truth measure of known carpet space in a panoramic image, it was found that the combination of carpet colour space model matching and gradient boundary detection produced the most accurate carpet detection results. The robustness of the carpet detection system was also tested in the presence of moving objects in the visual scene and through ego-motion of the mobile robot.

- *Local Space Histograms:* the knowledge about the extent of local space contained in occupancy grids can be abstracted to provide a simple, compact representation of local space. A method was presented to convert occupancy grids to one dimensional vectors with 16 values representing the extent of local space in 16 sectors surrounding the mobile robot.

- *Matching Local Space Profiles:* the compact nature of the local space profile allows for fast efficient matching between individual instances. A rotation invariant matching method was presented.

- *Categorising Local Space Profiles:* by categorising local space profiles, places in a topological map which are represented by instances of local space profiles can likewise be categorised. The perception of a particular category of local space profile can then constrain the localisation search to places which have a similar local space profile. The concept of local space primitives was introduced as a way to categorise local space profiles. This categorisation can be contrasted by using individual instances of local space profiles to represent places in the environment. An experiment comparing local space recognition when using local space primitives versus the use of matching individual places was carried out. Matching via places provided much higher recognition rates than matching with primitives. Although using primitives resulted in a large reduction in the local space matching cost, the savings were insignificant compared to the subsequent savings that matching with places would provide by more tightly constraining the low level localisation search.

The use of local space profiles as a mid level representation of spatial memory is intended to reduce the computation costs involved at localisation at lower levels of the representation. To do this the process of extracting such a profile from sensor data should be robust and efficient.

The ground plane detection methods used in this system as a basis to form an abstraction of the extent of local space are inherently unreliable and often produce incorrect results. The use of probabilistic reasoning to combine noisy information and promulgate an occupancy grid estimate over time helps to overcome this problem. The resulting occupancy grids however, are still subject to some inaccuracies, but for the most part a stable representation of the space surrounding the mobile robot is maintained. Possible improvements to this process would include using a more discriminatory colour space with which to build the model of the colour of carpet and introducing existing methods of detecting the ground plane as another information source.

The reduction of the occupancy grid form of representation to that of a local space histogram allows for fast and efficient matching of local space profiles. This reductionism assumes that the savings it brings to local space profile matching process outweighs the loss of detail in the local space measure. This loss of detail will have an effect on the degree to which a representation of local space can constrain the localisation search. A detailed study of the costs and benefits of minimalist in the representation versus the savings in localisation computation needs to be undertaken.

A study of this sort should also include any potential benefits of categorising profiles of local space might provide the localisation search. The notion of primitives introduced in this chapter did not appear to provide any benefits when compared with the use the local space profiles of all places in the topological map. Another categorisation method based on the robot's own experience of the environment might be able to maximise the trade-off between matching time and potential localisation savings.

In summary this chapter has detailed the implementation of local space profiles as a representation of spatial memory. Reasonably reliable estimates of the extent of local space are able to be made and can be successfully compared with each. The use of such a representation to constrain the localisation search at lower levels of spatial memory appears promising although by no means is the current implementation the most efficient one. Experiments applying the idea of local space profiles to the task of mobile robot localisation are reported in Chapter 8.

Chapter 6

High-Level Representation: Disambiguating Features

The previous two chapters have described two different visual cues that are used for mobile robot localisation. Both cues provide information which can be used to solve the localisation task. It is still possible however, for the robot system to encounter situations in the environment, when either or both of these cues fail to provide enough information to perform the localisation task. Adding extra cues would temporarily solve this problem, although again pathological cases would still exist. Rather than continually adding cues when faced with a lack of localisation information, it makes sense to instead actively seek out what are the defining features of a given area in the environment.

As mentioned in Chapter 2, the search for a disambiguating feature is inspired by the use of indirect landmarks by adult humans to perform spatial reorientation. When humans perform this task, they use all their semantic, episodic and linguistic knowledge to define rules for reorientation. Unfortunately robot systems do not have such a wealth of knowledge from which to generate rules to discriminate between places in the environment.

Because of this any such method for extracting disambiguating features will be limited by the knowledge available to the robot system. In this case the knowledge available to the robot system about places in the environment is limited to a panoramic image captured from various locations in the environment. No semantic knowledge about what the pixels in each image represent is assumed. One way to make discriminations between places based solely on this information is to use visual template matching to discover disambiguating regions in the image. On the surface this would appear to be just another form of visual cue to guide the localisation process, but in this case, the robot system is actively searching out features which allow one place to be discriminated from another.

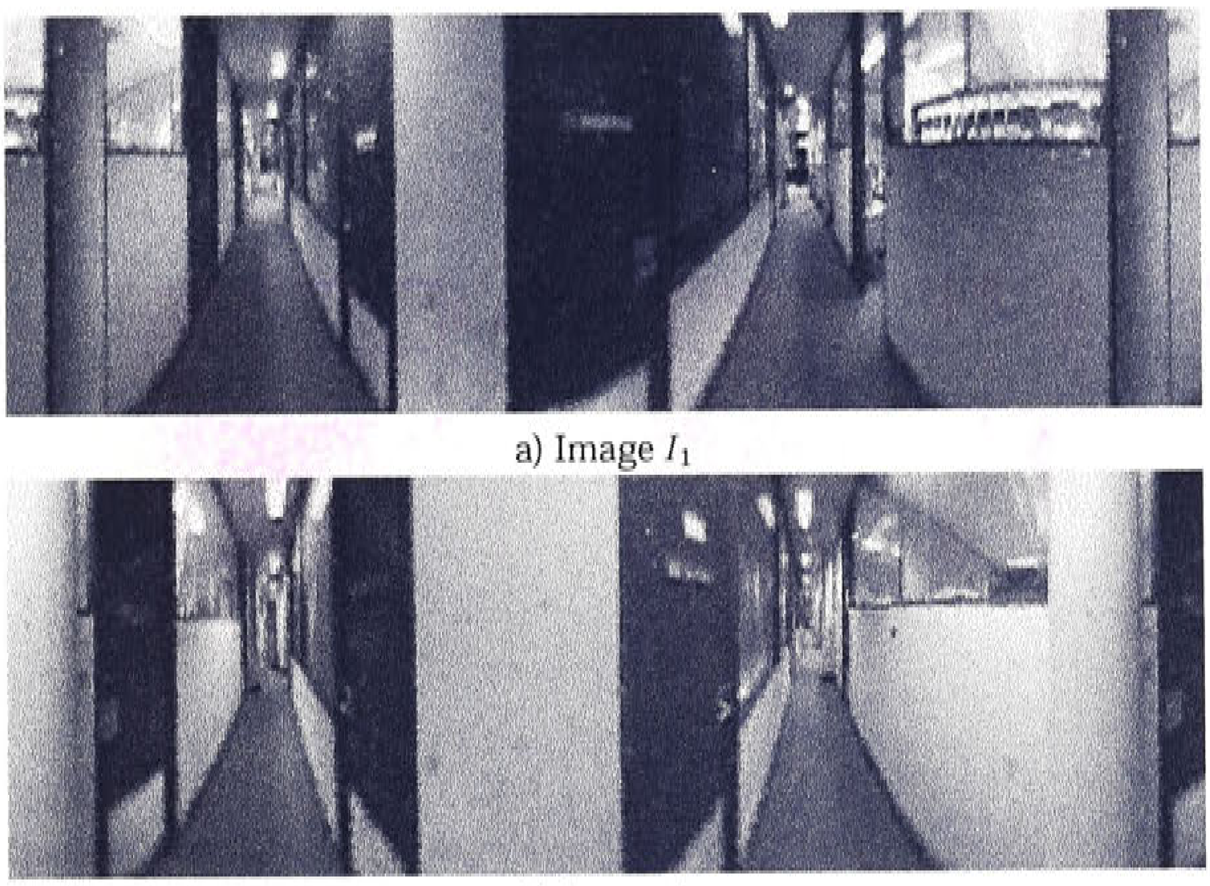
Panoramic images are captured from the reference position of each place in a topological map. These snapshots form the third level of spatial representation. In cases where the localisation information contained by lower levels of representation can not discriminate between the two places, localisation can be solved by extracting visual templates from one or both of the panoramic images which solve the localisation ambiguity.

This chapter describes how this system detects such disambiguating features between two panoramic images and then discusses how such features can be used to perform place discrimination. Finally a brief summary of the issues in this chapter is given.

6.1 Detecting Disambiguating Features

Environments which are visually sparse can result in topological maps which have places that contain similar panoramic views. Figure 6.1 shows two panoramic images which were captured at two distant locations in an environment but which share many similar features. A landmark based representation of these two places might include templates which are common to the corridor environment, leading to place discrimination errors. By actively searching for disambiguating features, these errors can be avoided.

Disambiguating features can be extracted from two panoramic images by using the correlation template matching approach described in Chapter 4. Given two panoramic images, I_1 and I_2, a visual template from I_1 can be evaluated for its ability to discriminate between the two images by performing a pixel by pixel correlation over possible regions of I_2. The worse the resulting correlation score, the better the template can discriminate between the two images. In other words, a feature provides good discrimination between the two images if there are no similar regions of pixel patterns in



a) Image I_1

b) Image I_2

Figure 6.1: An example of two panoramic images from two distant but visually similar areas in the environment.

the opposing image. This can be expressed as:

$$d_{x,y} = correlate(T_{x,y}, I_2) \tag{6.1}$$

where $d_{x,y}$ is the discriminatory ability of the template T at location (x, y) in I_1. The output of the correlation function is a value between zero and one which signifies the correlation value of the region in I_2 which is most similar in appearance to the template. Low values of $d_{x,y}$ indicate the template in question has a high discriminatory ability.

Using this method the discriminatory ability of all potential templates in a panoramic image can be calculated by evaluating $d_{x,y}$ for all values of x, y and all potential template widths and depths. The template which produces the minimum correlation value is the feature in the image which most discriminates the image from the other.

Figure 6.2 shows the disambiguating feature extracted by comparing image 1 to image 2. The identified disambiguating feature is shown by the red rectangle in part a) of the figure, while the region of image 2 which is most similar to the template is shown in

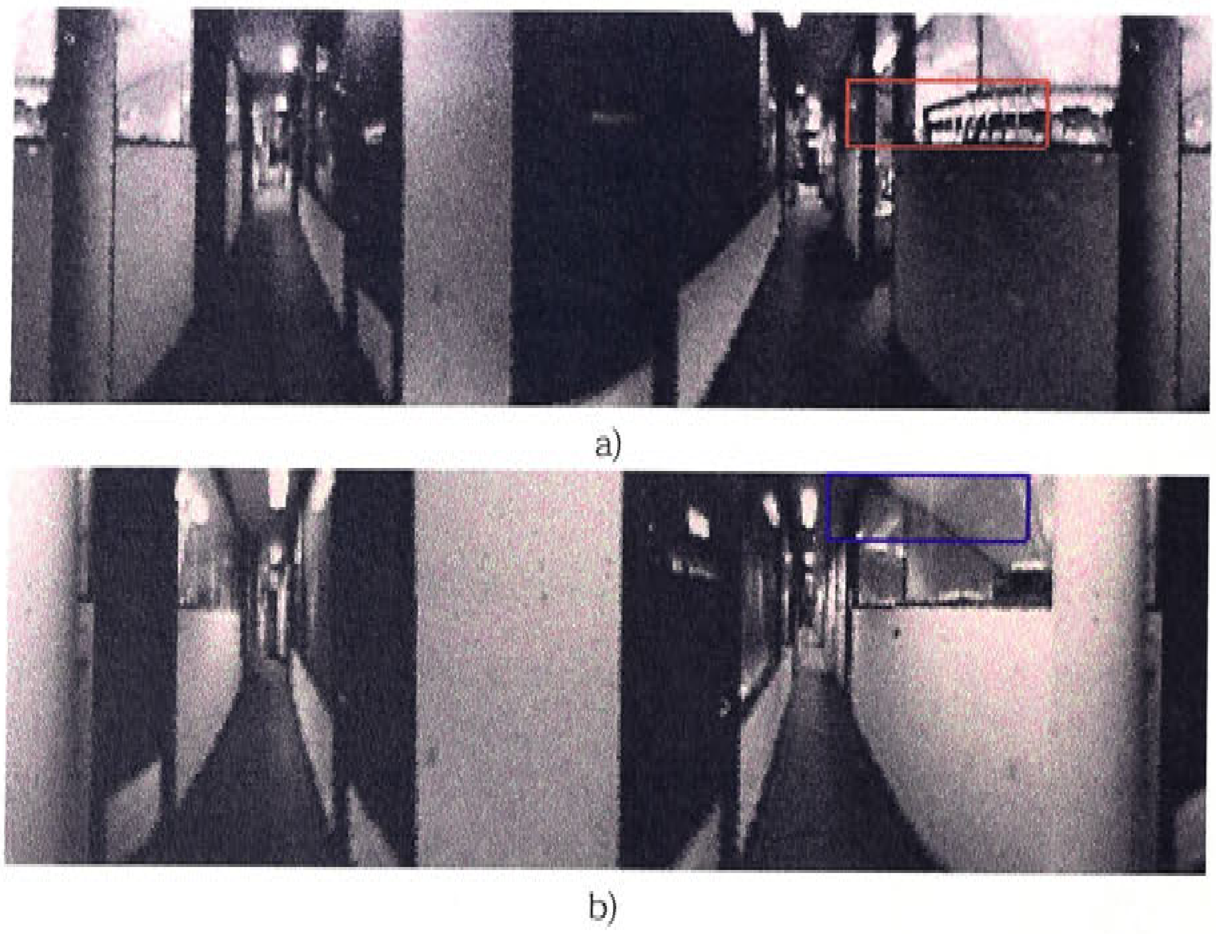


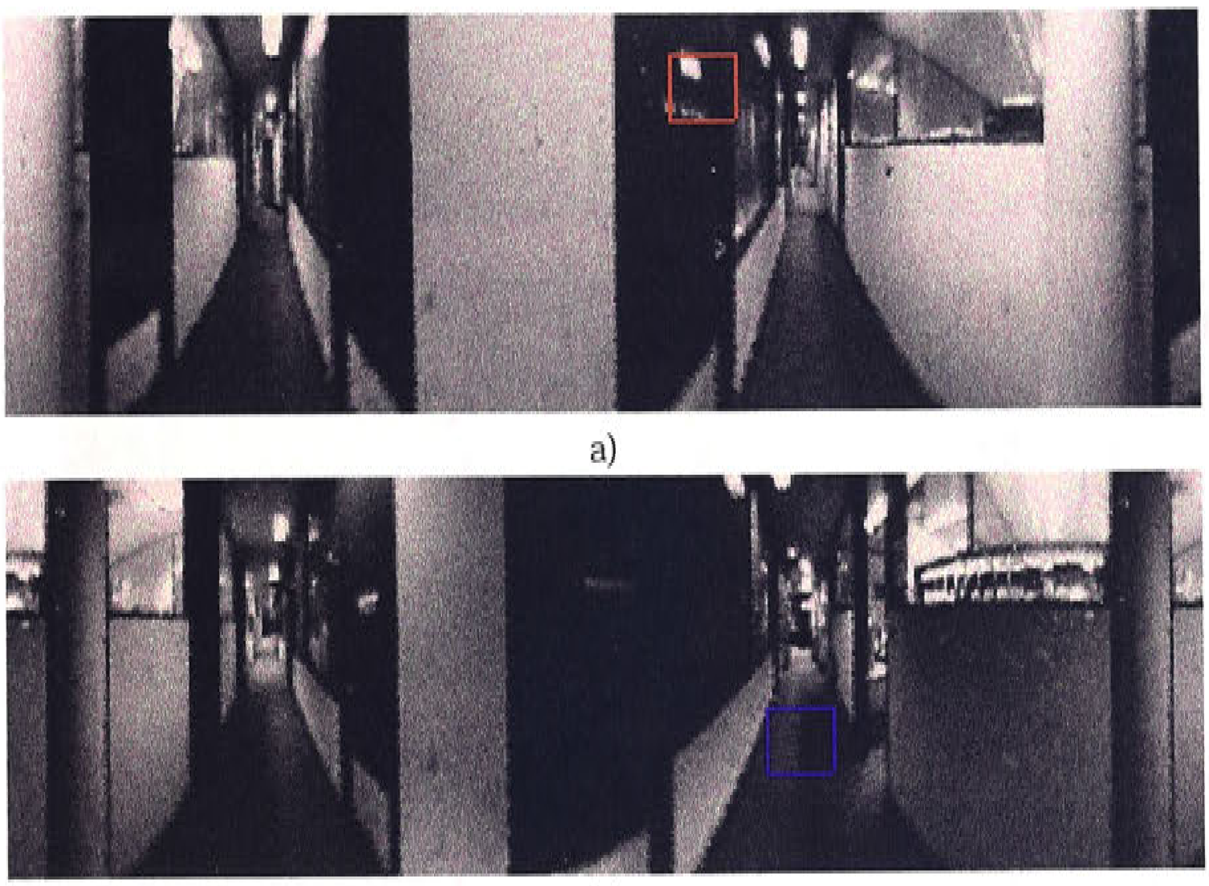
Figure 6.2: The disambiguating feature from image I_1 (a), and its most similar match in image I_2 (b)

part b). Visual inspection shows that these regions are quite dissimilar and can be used to successfully disambiguate image 1 from image 2.

This search over all possible templates and template sizes is of course prohibitively expensive. This method would greatly benefit from attentional cues and semantic knowledge about the content of images which could target computational resources to potential disambiguating features. The present method however does demonstrate the ability of a comparative search to correctly identify disambiguating features in pairs of panoramic images.

6.2 Disambiguating Features for Place Discrimination

Once a disambiguating feature has been extracted it it can be used for the purpose of discriminating between two places in a topological map. Given a panoramic image from an unknown location, a decision as to which of the two places in the map it was most probable to have been captured from can be made by comparing the current image with the disambiguating feature. If the disambiguating feature of one place



a)

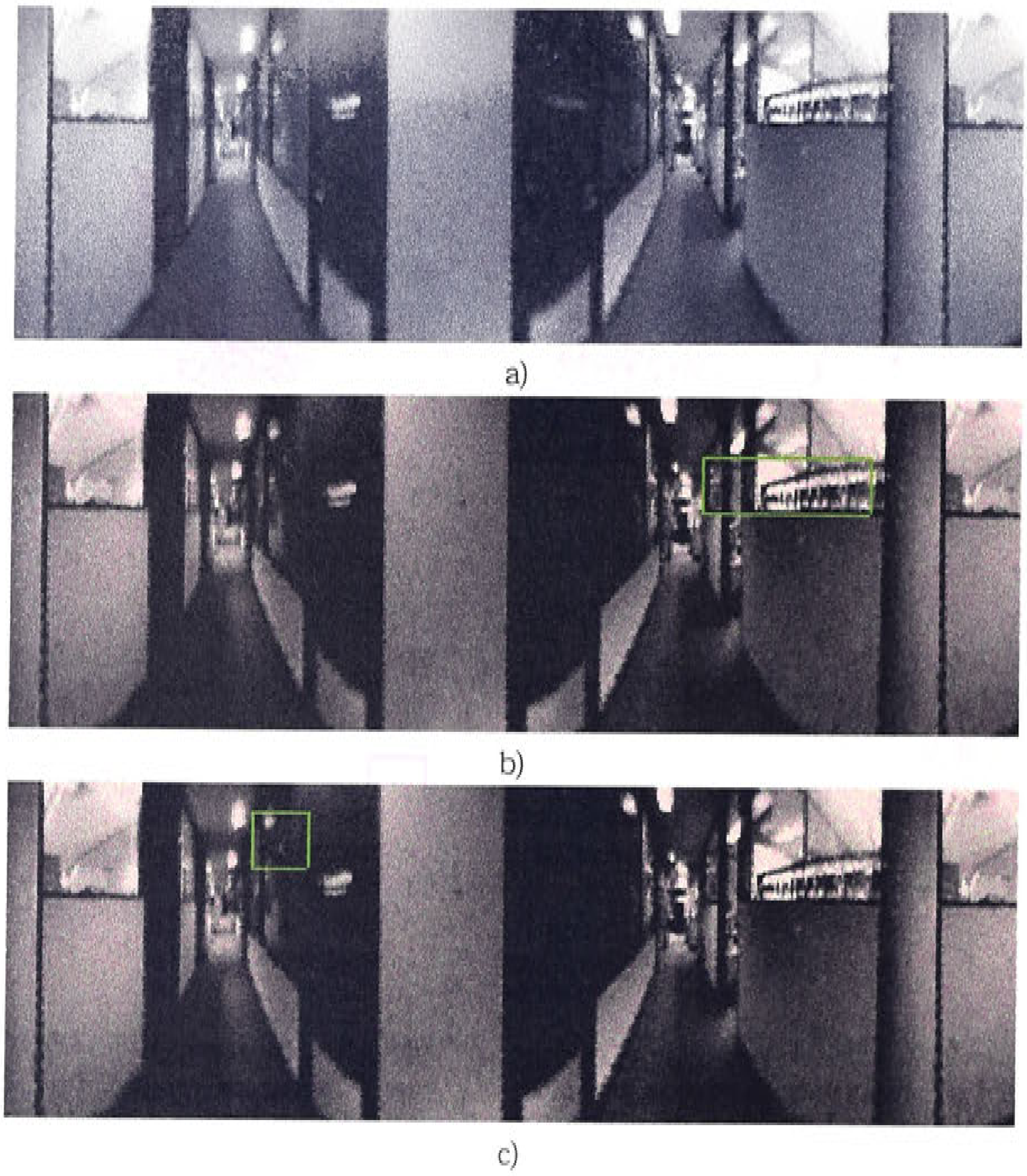
b)

Figure 6.3: The disambiguating feature from image I_2 (a), and its most similar match in image I_1 (b)

can be found with a high level of certainty within the current image, then it is more probable that the image was captured from that particular place in the topological map.

Given a panoramic image I_C captured from an unknown location in the environment and two places in a topological map, place discrimination can be achieved by first extracting the disambiguating features from each of the places' reference images when compared to the other. Figure 6.2 shows the identified disambiguating feature $d^1_{x,y}$ of place 1 when compared to the reference image from place 2, while Figure 6.3 shows the identified disambiguating feature $d^2_{x,y}$ of place 2 when compared to the reference image from place 1.

Both disambiguating features can then be matched against I_C producing a value of how well the templates correlated with regions of the current image. The place associated with the disambiguating feature which has the higher correlation value with the current image is identified as the more likely place from which I_C was captured. Figure 6.4, part a) shows an example image I_C which is similar to I_1 and I_2. Part b) shows the region in image I_C which is most similar to the feature $d^1_{x,y}$, while part c)



a)

b)

c)

Figure 6.4: An example of using disambiguating features to perform place discrimination. Part a) shows an image I_C captured from an unknown location in the environment. Part b) shows the best match of $d^1_{x,y}$ with I_C, part c) the best match of $d^2_{x,y}$ with I_C

shows the region most similar to feature $d^2_{x,y}$. Matching I_C and $d^1_{x,y}$ produced a maximum correlation value of 0.93 while matching I_C and $d^2_{x,y}$ produced a maximum of 0.81. Using this method it can be concluded that the image I_C was more likely to have been captured closer to the location of image I_1 than image I_2.

Although this method of place discrimination works well in the example presented above and demonstrates the applicability of finding discriminating features, it is in general not very reliable. Failure to successfully discriminate arise because of the lack of the simple method for choosing disambiguating features. Although the features do discriminate between the panoramic view captured from the reference position, the method does not attempt to evaluate the feature's distinctiveness or reliability in the presence of changes in perspective as does the local uniqueness measure and the Turn Back and Look movement of visual landmark selection.

These failing can lead to the identification of disambiguating features which successfully discriminate between the snap shots of reference places but not between views captured from regions adjacent to reference places.

6.3 Summary

This chapter has proposed a third level of representation for spatial memory. This representation involves storing complete panoramic images captured from the reference position of the place being acquired. The panoramic snapshots can then be used to perform place discrimination using disambiguating features when it is problematic to localise between places which appear similar at other levels of representation. This is analogous to the use of indirect landmarks in adult humans to resolve ambiguous spatial orientation problems.

Although using this representation to perform localisation is expensive and unreliable it demonstrates how actively searching for features which discriminate between similar places could be used to perform localisation in the most pathological of cases.

Additional cognitive pre-processing methods that direct attention and interpret the visual scene would, in combination with semantic knowledge about objects in the environment, would lead to faster more meaningful and more reliable methods for identifying disambiguating features.

Chapter 7

Topological Maps

This chapter details a method for constructing topological maps using the multiple levels of place representation described in previous chapters. As introduced in Chapter 2 topological maps are typically coarse, graph like representations of the environment. In these representations nodes correspond to significant places in the environment while edges in the graph correspond to transitions between places. In this system, places are represented by the three levels of spatial memory: those of visual landmarks, local space profiles and disambiguating features. The previous chapters presented the details of how to form a representation of a place. This chapter aims to introduce methods of deciding when to add a place to the topological map, and of representing the connections between places.

The outline of this chapter is as follows: Section 7.1 details how to construct a simple topological map. This includes a method of deciding when to learn a new place as well as describing a way to record the transitions between learnt places. Section 7.2gives an example of a topological map constructed using the methods described. Finally, Section 7.3 provides a discussion and summary of the issues involved in building a topological map.

7.1 Building Topological Maps

A topological map of the robot environment can be constructed by learning a series of places along a path and associating them with information which describes the

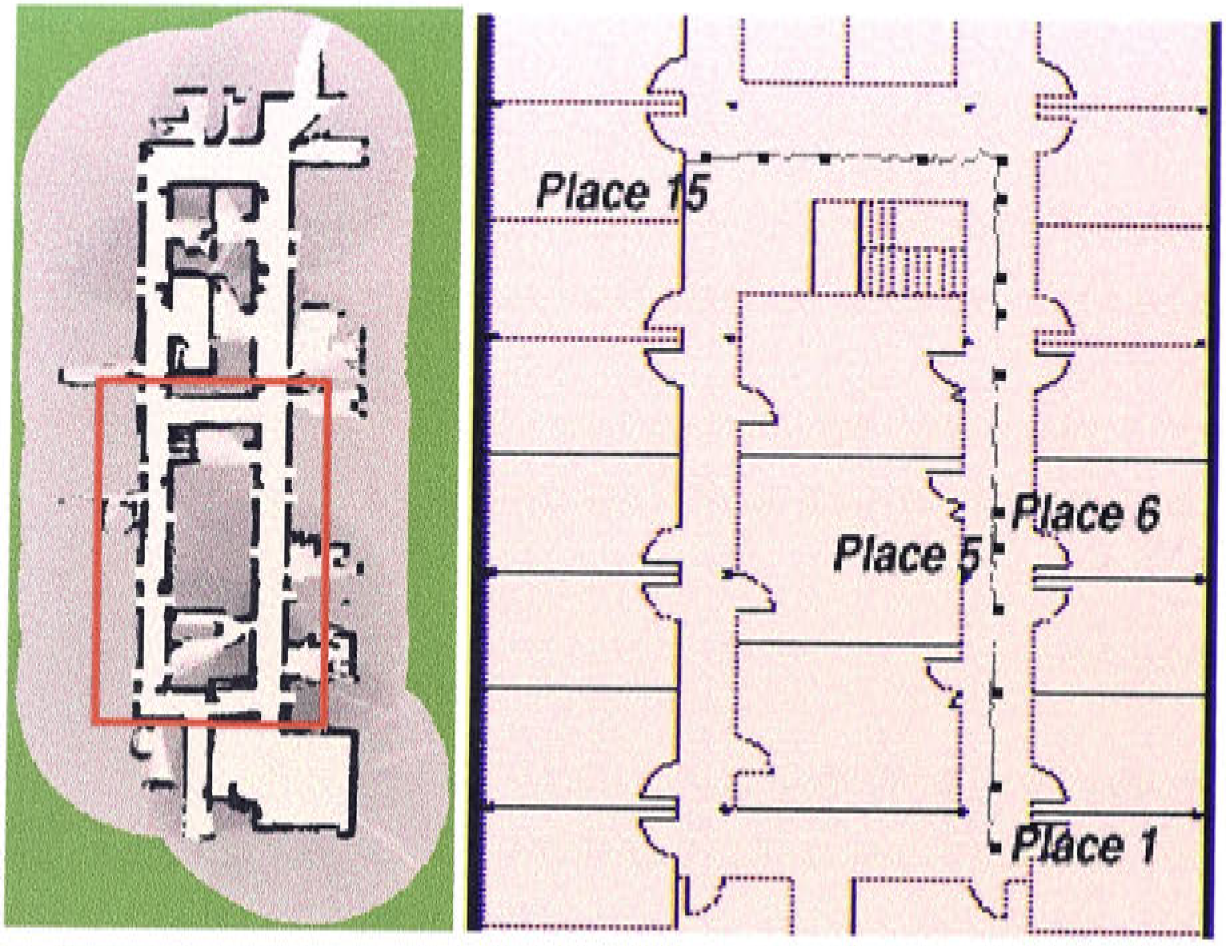


Figure 7.1: Map of learnt places (black filled squares) in corridor environment and actual robot position along an example path (grey path).

transitions between places. Figure 7.1 shows an example topological map constructed in the corridors of the Robotic Systems Lab at the Australian National University. The environment is a typical corridor environment, containing regular doors, windows and light fittings but little else. It contains 15 learnt places over a 25 meter long path. The black squares along the grey path denote the locations along the path where places where learnt. From this map it is evident that a topological map representation must contain two types of information:

- *Place Representation:* each node in the topological map represents a place in the environment. A topological map therefore must contain the spatial representations of each place that has been learnt. This information facilitates the recognition of places when revisiting areas of the map. As mentioned above the chosen place representation of this system is the multi-level spatial representation introduced in earlier chapters.

- *Transitions Between Places:* In order for the map to reflect the topology of place locations which exists in the real world environment, the map must contain information about the connections, or transitions, between learnt places. A transition defines how to move from one node or place in the map to another node.

In the current system, a topological map is represented by an array of learnt places, each of which is accompanied by a list of transitions to adjacent places.

7.1.1 Adding Nodes

The example map above was formed by manually identifying locations in the environment where the robot system should learn a place. This was done at fairly regular intervals with no significance being assigned to the chosen locations. In general it would be advantageous for the system to automatically identify when a new place should be learnt. A robot system which relies on user intervention every time it encounters a novel section of the environment can hardly be classified as an autonomous agent, and such a shortcoming would seriously limit a robot's performance in real world applications.

The question is then, when should the system decide that a new node must be added to the topological map, and how should that determination be made?

Naively, the answer to the first part of the question is that a new place should be learnt when the robot encounters a new area of the environment. Remembering that the robot is using a topological map to represent it's understanding of the environment, specifically that a particular representation is used to represent a place in the environment, the answer can be developed further: A new place should be added to the topological map at the point when the robot can no longer recognise its current location when compared to the representations of the places contained in its existing map. Incidently this also answers the second part of the question by predicating learning a new place with recognition failure.

In a multi-level place representation such as this the question of determining recognition failure is not straightforward. Given that the driving factor of our research is to produce a system which has a sparse map representation while allowing accurate localisation estimation, the addition of a new node should be determined by a loss of recognition in the representation level which provides the accuracy, that of visual landmarks. The visual landmark level also provides the most unique representation of a place which also is important for recognition performance.

A change in the perceived local space profile associated with the second level of place representation might also signify that the robot is entering a significantly different section of the environment, thus warranting a new node in the topological map. This is important considering the desire to constrain the global localisation search by means of the local space profile. If the local space profile changes dramatically within a given place this could lead to miss-constraints in such a search. The need to include this level in node addition determination is countered however by the possibility of adding places to the map which contain similar landmark sets. In general the appearances of a set of visual landmarks and the local space profile for a given section of the environment are independent and it would be problematic to attempt to form a relationship between the two for the purposes of determining recognition failure.

Landmark Recognition Performance (LRP)

In our research, recognition failure in terms of adding a new place to the topological map, is determined solely by the loss of landmark set recognition. The recognition of landmarks sets was mentioned in Chapter 4 and the brute force matching approach is used in this section to demonstrate landmark recognition performance. It must be noted that landmark set recognition and topological map construction presuppose localisation, which will be discussed in the next chapter.

As stated in Chapter 2, mobile robot navigation is a behaviour made up of the simultaneously operating sub-components of planning, localisation and mapping. It is almost impossible to strictly separate the components into sequential chapters as is attempted here, without some mention of components yet to be covered. Therefore in this chapter localisation and landmark recognition will be assumed although details of these procedures will be left until later.

The average landmark correlation measure for a set of landmarks can be used a measure of landmark set recognition. This measure can be stated more formally as:

$$LRP_{ip} = \left(\sum_{j=0}^{N-1} (locatelm(l_j, i)) \right) /N \tag{7.1}$$

where, LRP_{ip} is the landmark recognition performance of place p in image i, N is the number of landmarks in a landmark set, and $locatelm(j, i)$ returns the correlation measure of the best match of landmark l_j in image i. Landmarks l_j being elements of the landmark set represented place p.

Suppose a robot has globally localised and is certain it is within a particular place in the topological map. That is to say the landmark set recognition measure for this place is greater than all other places in the map. The identified place can then be called the *Most Likely Place* (MLP).

Thresholding LRP

A simplistic approach to the use of landmark set recognition as a determinant to adding nodes to the map would be to learn a new place when the robot moves through the environment in such a way as to cause the landmark set recognition measure for the most likely place to fall below an arbitrary threshold. This of course means that the likelihood of being in all other places within the map is also below this threshold. Matsumoto, Inaba and Inoue (1997) apply a threshold to image acquisition in their view-sequence approach to mobile robot localisation. A new view is acquired along a route when the correlation between the current panoramic image and the last stored panoramic view drops below a set threshold level.

In the context of our research such an approach makes an undesirable assumption. It assumes, falsely, that landmark recognition performance for all landmarks will degrade gracefully as the robot moves away from the place where the landmarks were captured, decreasing to a common background level. Background LRP level is the matching result achieved between a set of landmarks and an arbitrary environment, where chance similarities provide a steady level of LRP although the actual landmarks are not necessarily visible. With this assumption it is necessary to nominate a threshold level which is above the chance level of recognition performance, and ensure that landmark sets above this level are defining a unique section of the environment.

Figure 7.2 shows an example of landmark recognition performance for the topological map presented in Figure 7.1. The figure displays the LRP produced when the set of landmarks associated with place five in the map (fifth black square from the bottom)

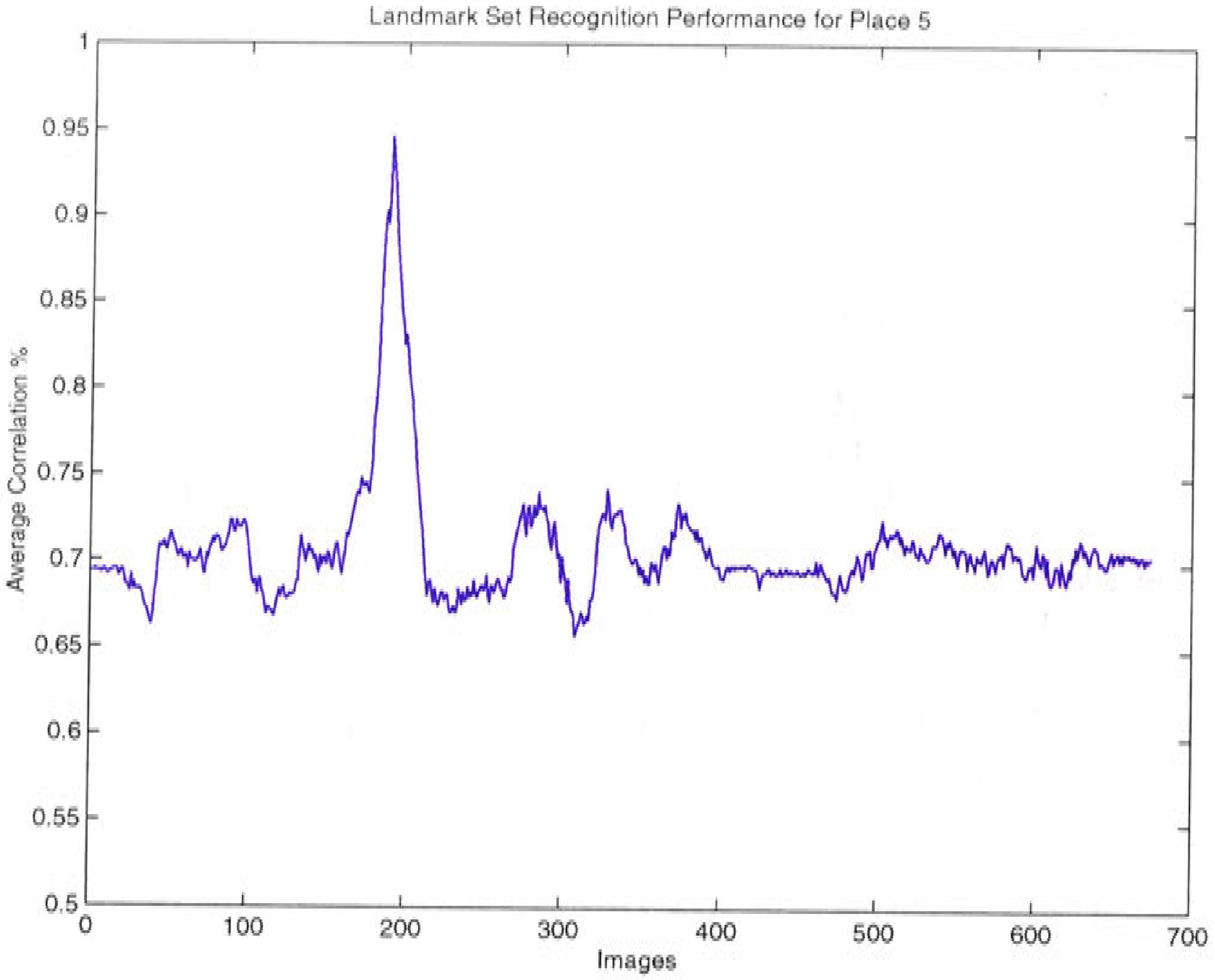


Figure 7.2: The landmark set recognition performance for the landmarks associated with place 5 over the robot path.

are matched with the approximately 700 panoramic images captured over the robot path. The distinct peak in the LRP graph for place five (around image 190) is a result of the robot passing over the position in the environment from where place five was learnt. As the robot moves away from this position the LRP measure quickly decreases and levels off at approximately 0.7. Using a threshold value of slightly above 0.7, the LRP could be used to trigger the acquisition of a new place when the LRP falls past the assigned threshold.

This method, although attractive in its simplicity, suffers from the false assumption mentioned above. The fact is that not all places show the same LRP behaviours and setting a threshold level can be problematic. A landmark set which includes landmarks which remain in view over a large area of the environment, such as those at a far distance will produce much higher back ground LRP levels than those with close landmarks which quickly distort under translation.

This phenomenon is illustrated in Figure 7.3. In this figure the LRP plots from places

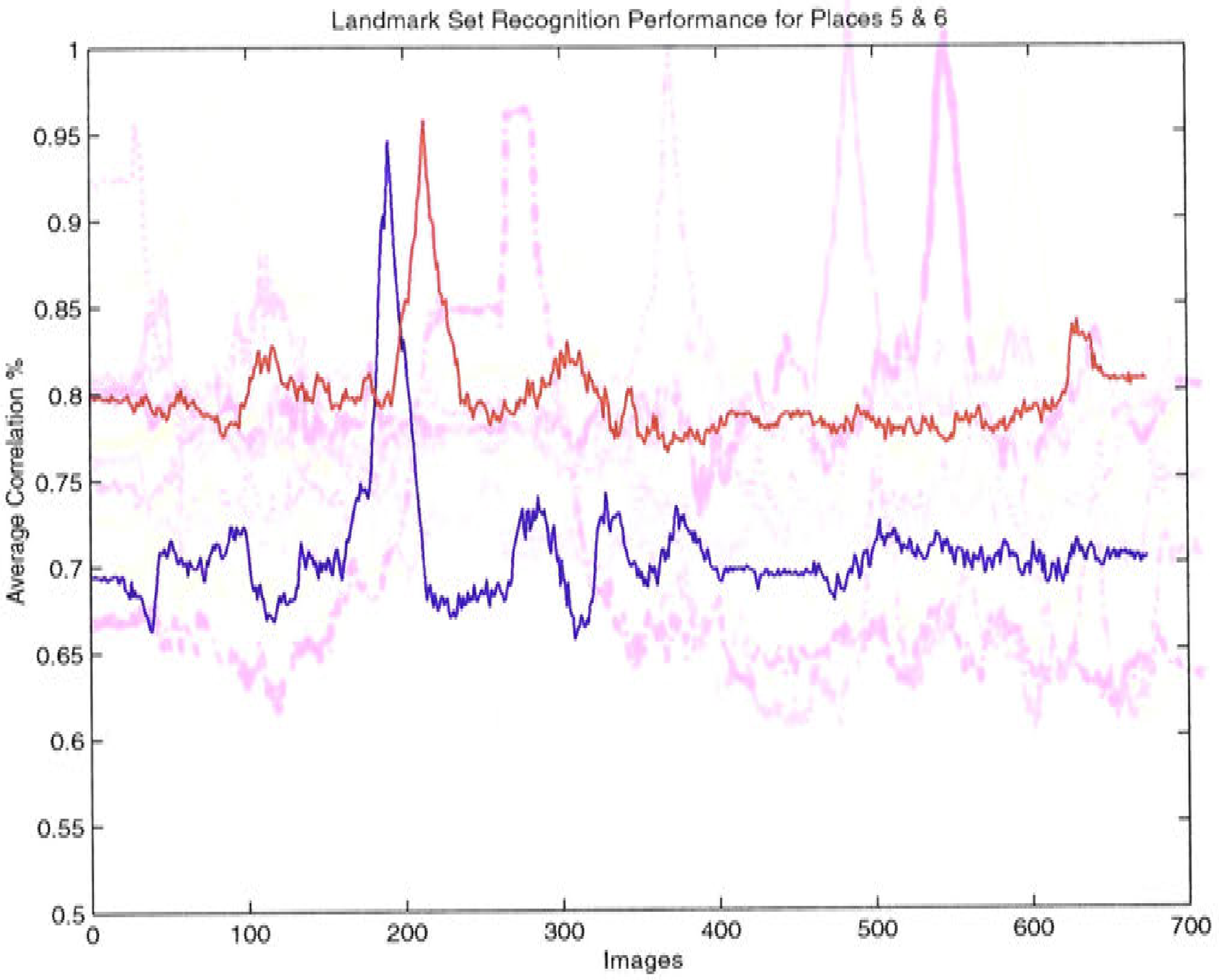


Figure 7.3: The landmark set recognition performance for the landmarks associated with places 5 (blue) and 6 (red) over the robot path.

five (blue) and six (red) in the topological map are shown. Again, the peaks of both places are evident, but what is surprising is the difference between the background levels of both places. Whereas the the background level of place five can again be seen as about 0.7, place six has a background LRP level of approximately 0.8. If the threshold for learning a new place was set at slightly above 0.7 then a new place would never be learnt after place six had been included in the topological map. Setting a threshold significantly higher than 0.8 would result in a map that is very dense and ultimately prove un-scalable.

This example highlights this problem particularly well due to of the nature of the environment where the topological map was made. The office corridor along which the map was built has very few distinguishing features and the most interesting landmarks lie at the ends of the corridor. This sparse visual environment results in landmark sets which contain landmarks that can be seen along the majority of the corridor and therefore produce a high level of background LRP.

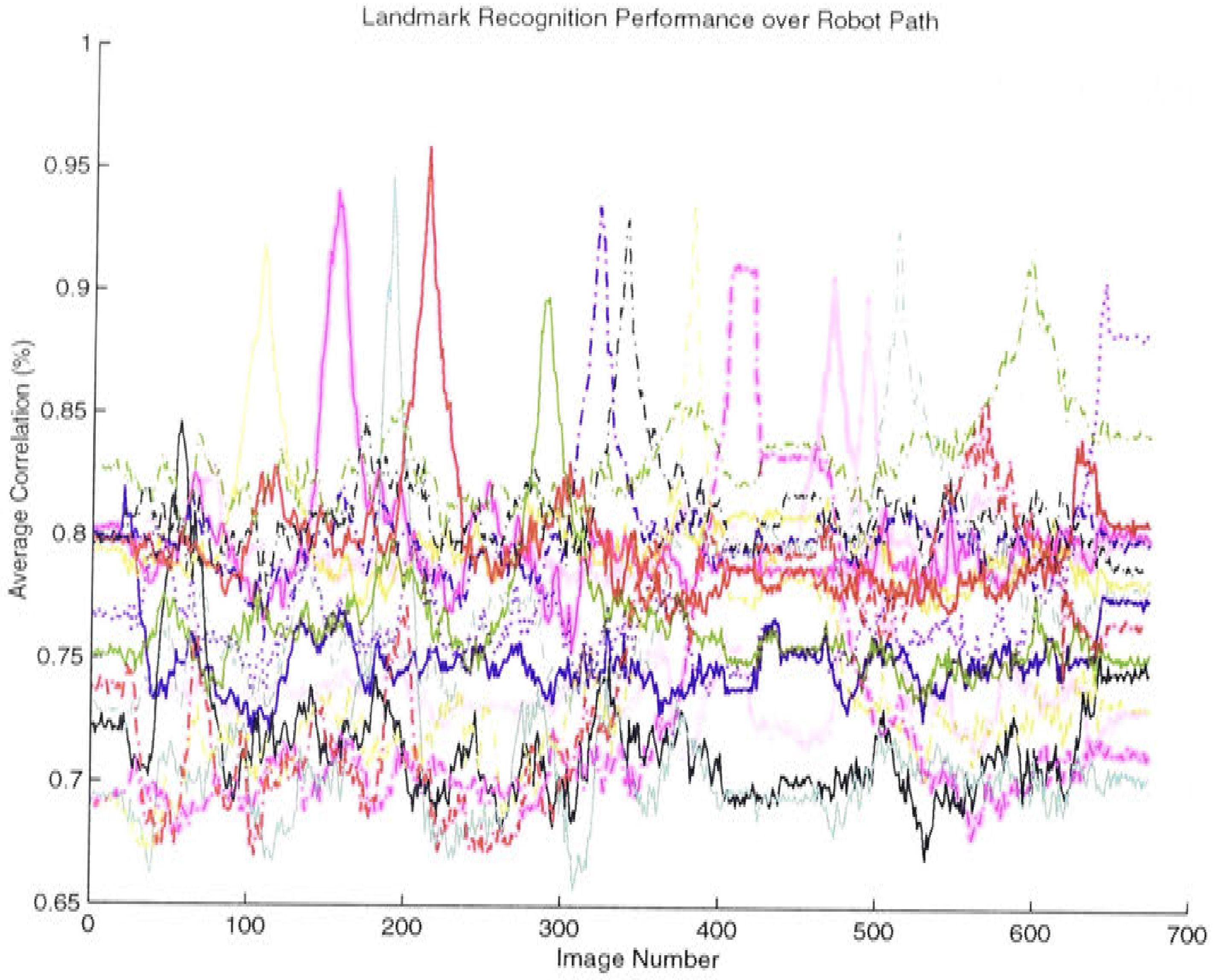


Figure 7.4: The landmark set recognition performance for the 15 places over the robot path.

This large difference in background LRP levels will not be seen in all environments and maps and it is impossible to predict when such problems will occur, and what level the background LRP should be. Therefore it is impossible to set a static threshold level to determine when to learn a new place which will function reliably over arbitrary environments. Figure 7.4 presents the LRP of the 15 places in the example topological map to demonstrate the unpredictability of LRP background levels over a larger region. It is hard to trace individual places LRP levels in this figure, however the varying levels of background LRP are evident. The flat region of the graph between images 400-500 is caused by the robot turning on the spot, which produces a constant LRP.

The variance in the background level of landmark set recognition means that a threshold is not sufficient to determine when to learn new places.

Background LRP Levels

An adaptive threshold which varies with the level of background LRP would seem to be an ideal solution for determining when to learn a new place. This process would require the background levels to be known in advance. Only when the highest level of background LRP is known for a given area of the environment can a suitable threshold level be set. Of course it is impossible to know the background levels before encountering the region, therefore the decision of when to learn a new place can be made only when the background level has been detected. This means that only when the LRP of the landmark set associated with the most likely place has been found to have decayed to its background level will a new place be learnt. This changes the determination process slightly from that of thresholding to one of detecting when the LRP measures for a place has reached background levels.

The difficulty in detecting when the LRP is at background levels is that no prior knowledge can be assumed. If a place has just been learnt, there is no past history of what a landmark set's background LRP level is. Also, although in this example we are assuming that all landmark sets are being monitored for their recognition performance, when the map gets significantly bigger more efficient localisation methods are needed since it will not be possible to track all the sets of landmarks all the time. Therefore a stored recent history of a places LRP can not be assumed, even if it is already part of the topological map.

From the LRP graphs presented, it can be seen that background LRP levels remain at a constant level, although there is considerable noise. The background noise level evident in locations along the path adjacent to any given learnt place is the same for those locations that are distant from the places reference position. If immediately after learning a place the LRP background level for that place is detected and recorded, then this level can be used later to estimate an appropriate threshold level for node acquisition in the topological map. The steps involved in deciding whether to learn a new place are as follows:

1. *Find Most Likely Place:* use landmark template matching to find a place in the map at which the robot is most likely to be given the current sensor readings.
2. *Set threshold:* set the LRP threshold for determining whether a new node should

be added to the map. This is achieved by either

   (a) Setting an estimate according to the recorded LRP background levels of the most likely place determined when that particular place was learnt.

   (b) If the MLP is new, and no background level has been estimated set the threshold to a default low level.

3. *Evaluate LRP:* Evaluate the LRP of places in the map. If the current most likely place is new and the threshold is still at the default level (the background level is not known), attempt to detect the background LRP for the most likely place. Continue evaluating LRP performance until:

   (a) Another place in the map becomes the MLP, in which case start over at step 1.

   (b) The LRP of the most likely place falls below the threshold level. Proceed to step 4.

   (c) If a background level has been detected, store the value and proceed to step 4.

4. *Learn A New Place:* add a new node to the topological map and proceed in acquiring a representation for that place as described in the previous chapters. If the robot is in continuous operation the system would then return to step 1 and continue to repeat the procedure.

In this fashion new places can be added to the topological map when the LRP of the most likely places reaches its background level rather than an arbitrary threshold.

Background LRP Detection

The method for determining when to learn a new place depends on the ability of the robot to detect when the LRP for a particular place is at the background level. Since the method can not assume any historic LRP measurements, it must rely on current landmark recognition measurements or those from the recent history. The method should also be independent of the actual value of LRP as the background levels can change from place to place.

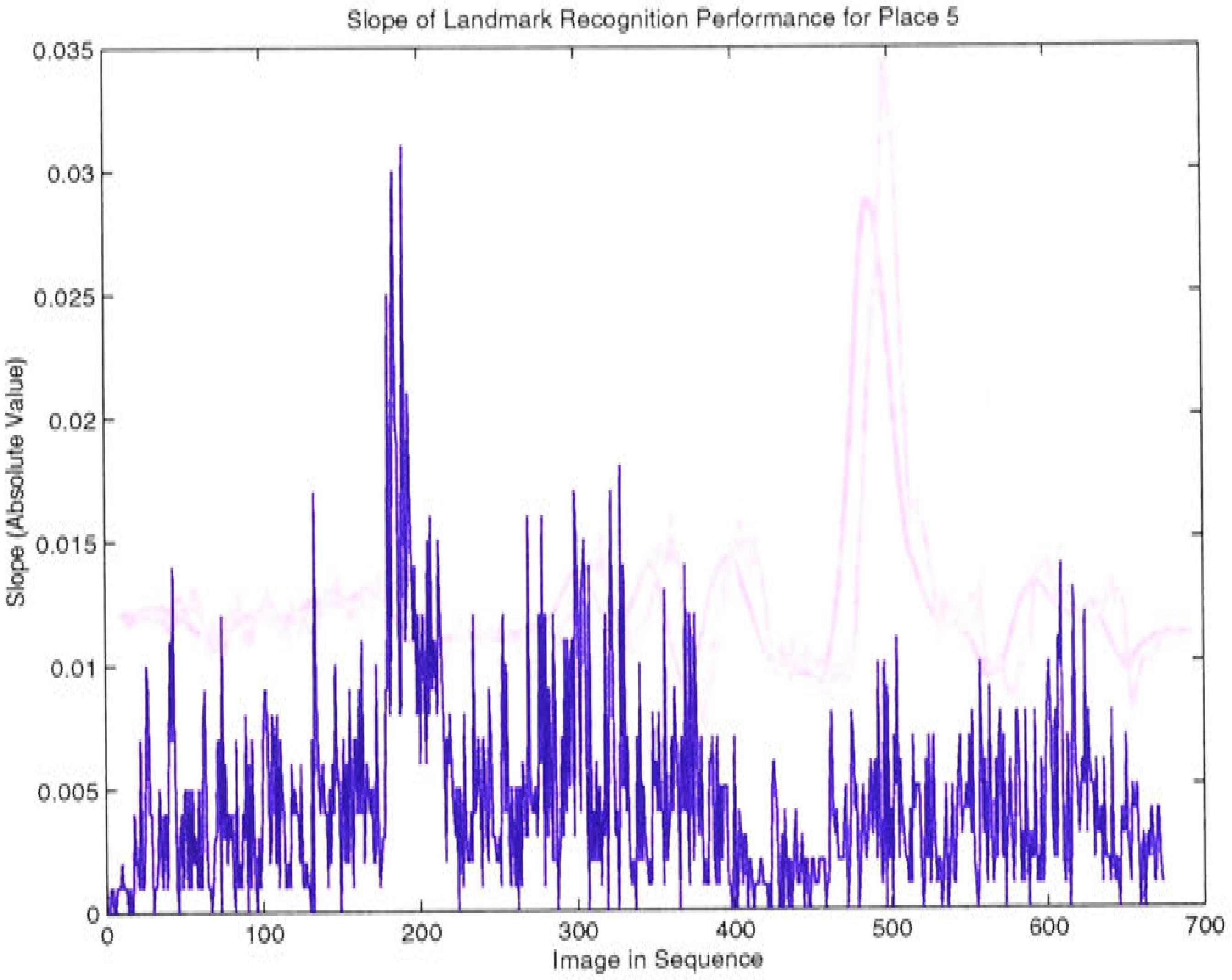


Figure 7.5: Instantaneous slope measurements for place 5 over the robot path.

One way to detect a background level of LRP is to monitor the gradient slope of the LRP between samples. From the graphs above, it can be seen that the LRP peaks sharply in areas associated with places reference positions, but is reasonably constant at other times. In general it can be said that sections of the LRP graph which have a zero slope are representative of background LRP levels. If while monitoring the LRP performance at a given place, the slope between recent measurements is approximately zero, the system determines that the LRP performance is at background level.

Figure 7.5 shows the instantaneous slope of the LRP for place five as illustrated in Figure 7.2. The data in the plot was obtained by simply taking the absolute value of the result of subtracting the LRP of every image in the sequence from the LRP from the previous image:

$$s_i = |LRP_i - LRP_{i-1}| \tag{7.2}$$

where, s_i is the instantaneous slope of the LRP function at image i, and L_i denotes the LRP performance at image i.

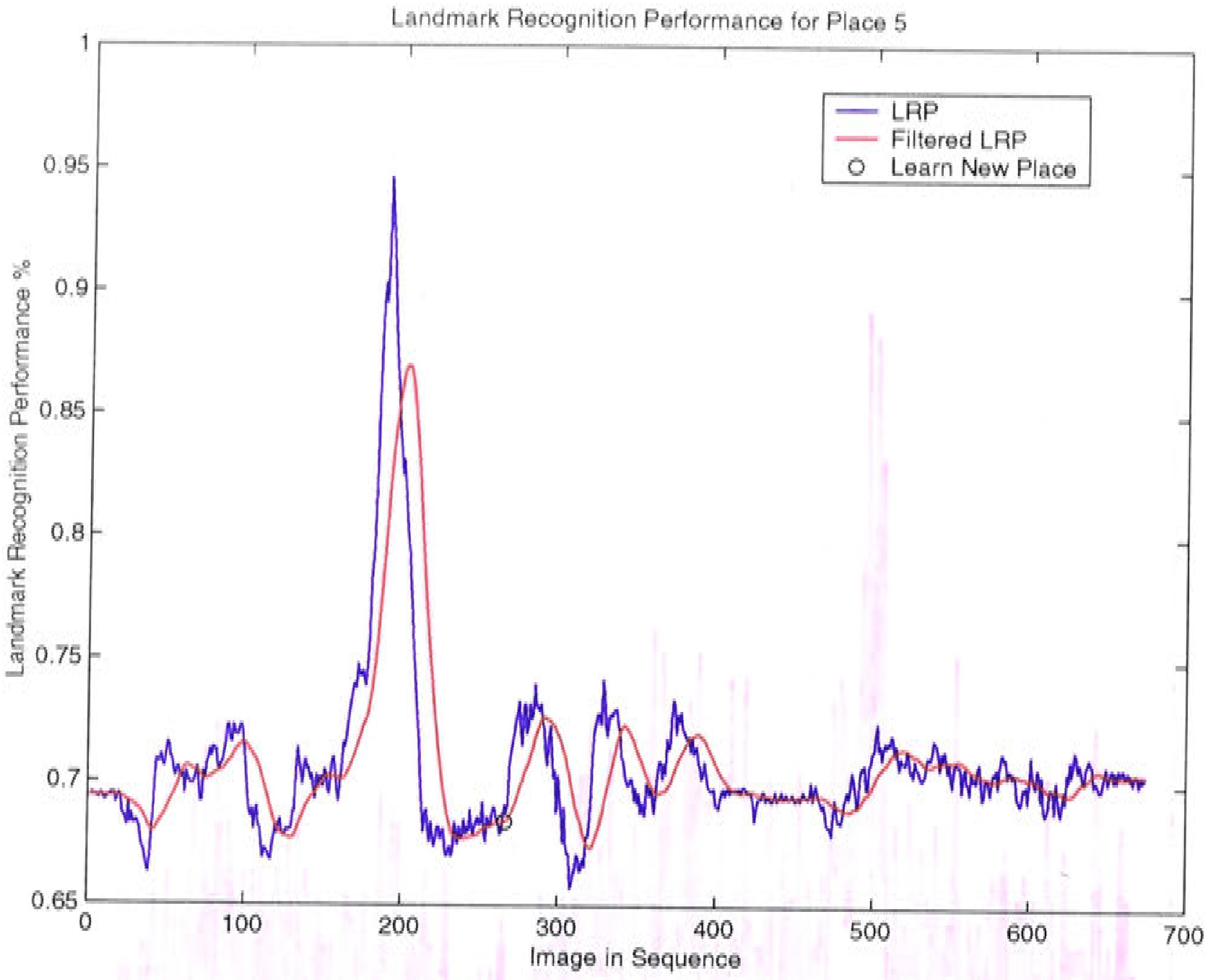


Figure 7.6: Raw and filtered Landmark Recognition Performance (LRP) for place 5 over the robot path. The black circle indicates when a new place would be learnt.

Although a peak in this graph is evident it is still noisy. This reflects the noise in the LRP measurements. This noise can be partially alleviated by applying an averaging filter to the LRP information. This filter averages the LRP measurements over the last N readings:

$$LRP'_i = \frac{\sum_{j=0}^{20}(LRP_{i-j})}{N} \tag{7.3}$$

with the size of N depending on the velocity of the robot and the rate of data sampling.

The results of applying the averaging filter to place five's LRP data is shown in Figure 7.6. The raw LRP measurements are illustrated by the blue line while the filtered measurements are shown in red. Filtering provides a much soother view of the LRP and a peak can clearly be seen at the same position as the one in the LRP graph, albeit delayed due to the windowing affect of the filter. The slope of the filtered LRP

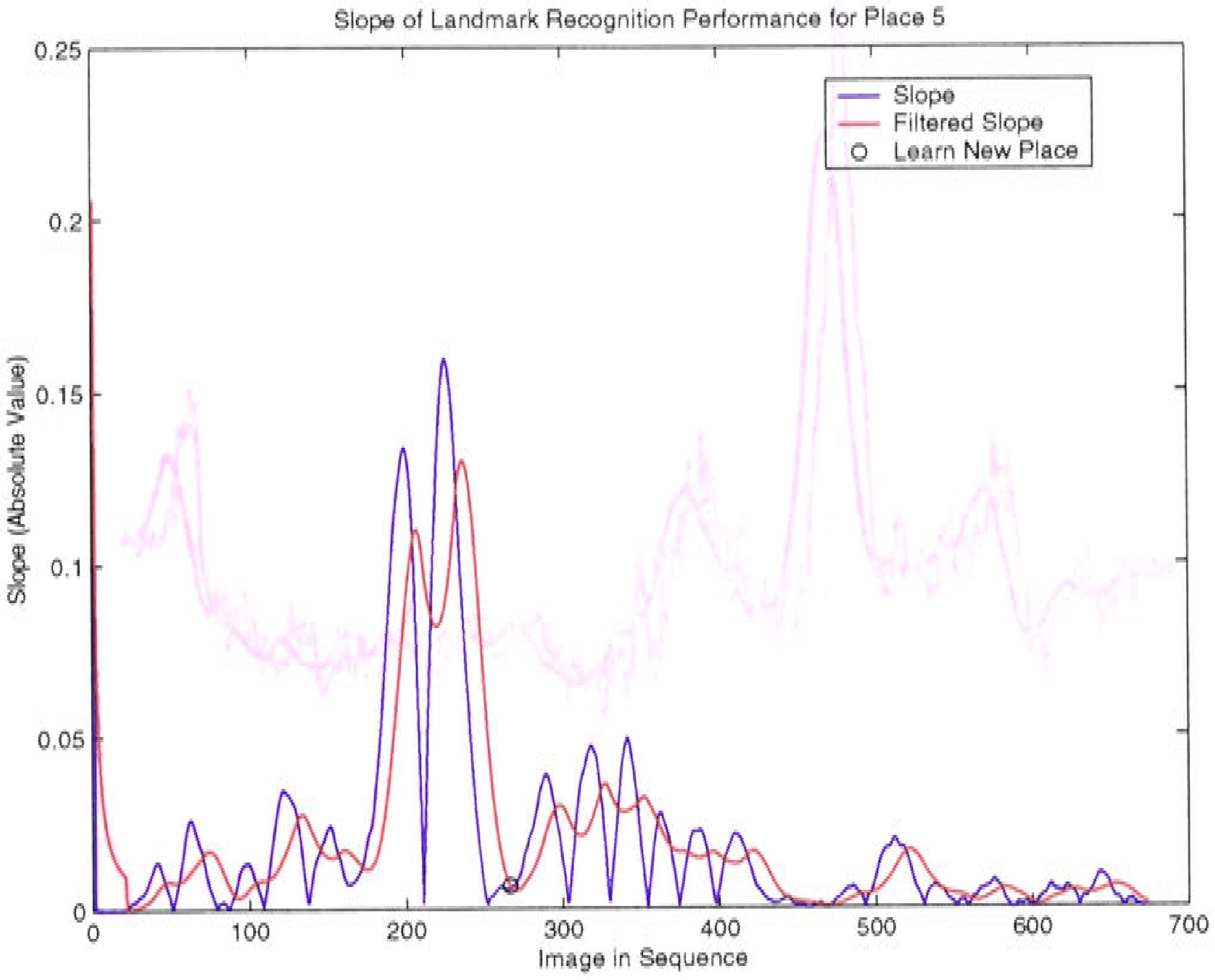


Figure 7.7: Raw and filtered slope measurements for place 5 over the robot path. The slope measurements are calculated from a window over the filtered LRP.

measurements can then be calculated. Given that the instantaneous slope can still reflect local minima, the slope measurements used are those observed over the sampling window:

$$s_i = |LRP'_i - LRP'_{i-N}| \tag{7.4}$$

where, s_i is the slope of the LRP function between images captured at steps i and $i - N$ and LRP'_i denotes the filtered LRP performance at image i.

The output from Equation 7.4 can then then be filtered again to further remove any local minima in the slope function:

$$s'_i = \frac{\sum_{j=0}^{N}(s_{i-j})}{N} \tag{7.5}$$

The results of applying Equations 7.4 and 7.5 to the LRP performance for place five

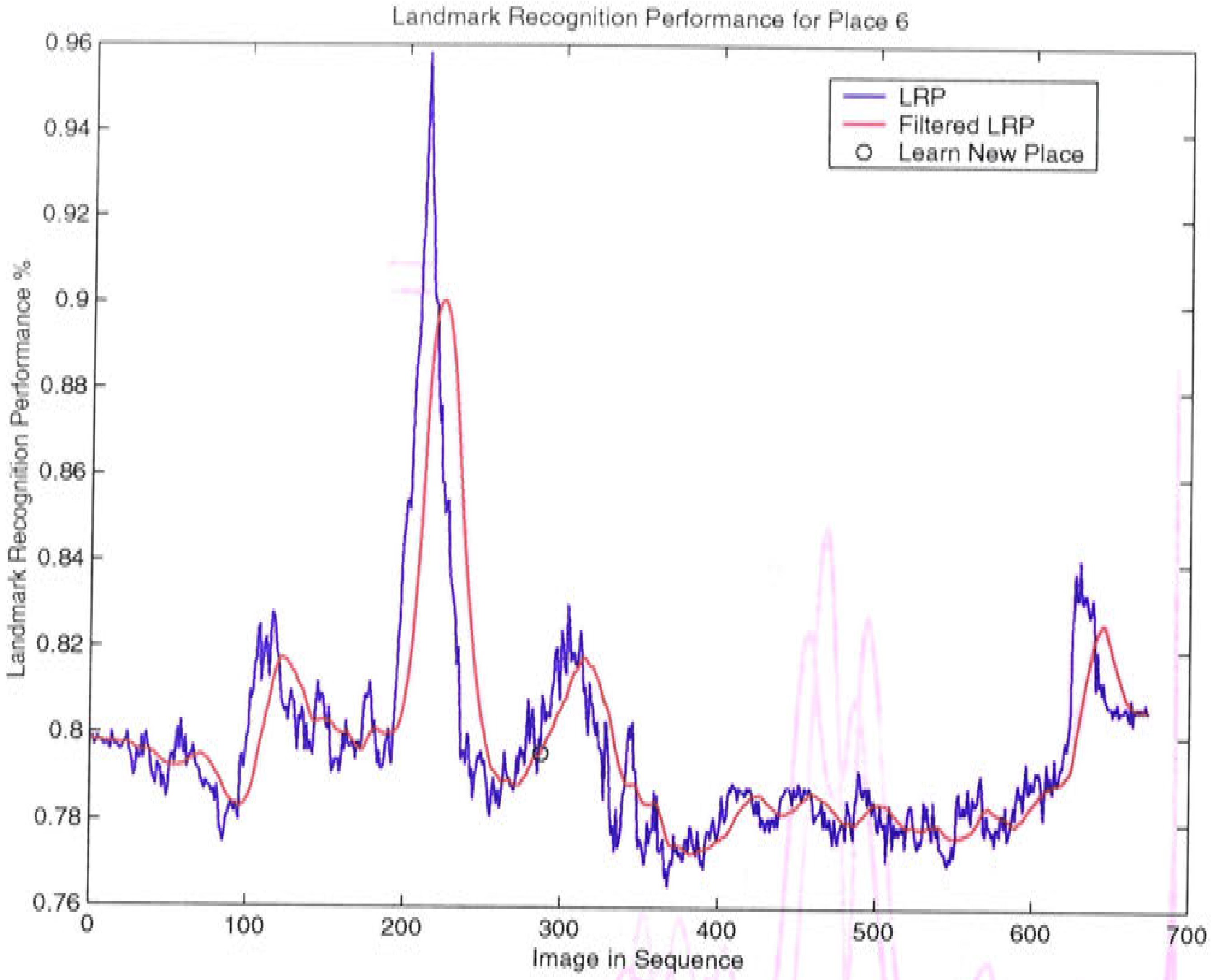


Figure 7.8: Raw and filtered LRP measurements for place 6 over the robot path.

are shown in Figure 7.7 by the blue and red lines respectively. A threshold can then be used on the smoothed slope value to determine when the LRP reaches background levels. The threshold limit used in our experiments was 0.0075. The difference between thresholding on the slope and thresholding on the LRP measurements is that the slope values are independent of the actual background level of the LRP.

Assuming that the process of detecting background LRP levels began at the top of the peak (simulating learning a new place), and continued along the image sequence, the above process with the given threshold will detect background levels of LRP at the point in the graph denoted by the black circle in Figures 7.6 and 7.7. The robot can use this method of detecting background levels of LRP to trigger the acquisition of a new place in the topological map.

To demonstrate that this method of determining when to learn a new place in the topological map is independent of the levels of background LRP, Figures 7.8 and 7.9 show the raw and filtered LRP performance and slope for place six in the topological map.

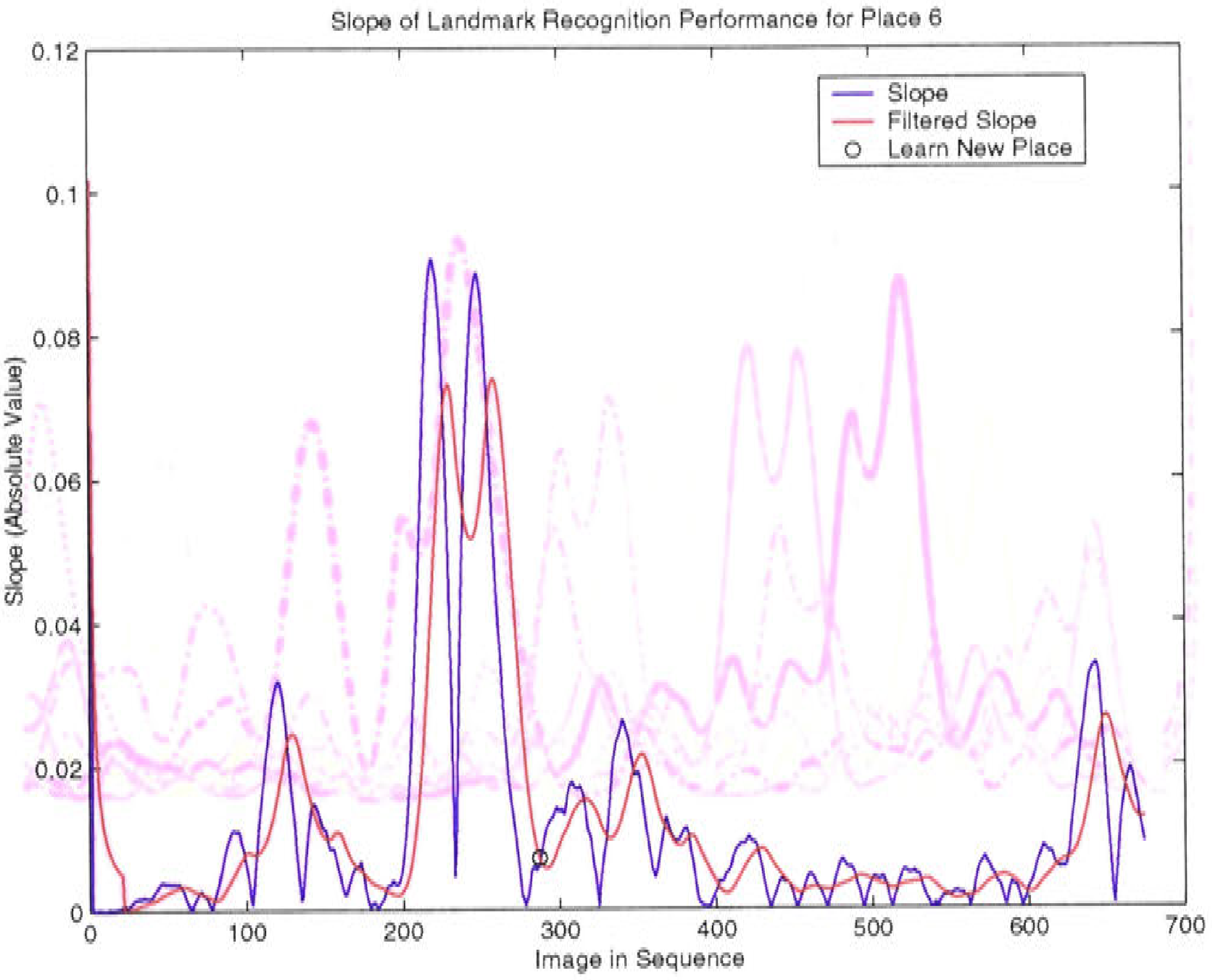


Figure 7.9: Raw and filtered slope measurements for place 6 over the robot path.

The last step to the process of determining the level of LRP at which a new place should be learnt is to provide a default level beyond which the threshold should not drop. A LRP performance measure below this default level is considered too poor to be reliable even though the background level has not been detected. If the LRP performance for a place falls below the default level the robot system treats the default value as the background level. Also to handle the change in slope at peaks and newly acquired places, it is assumed that if the LRP performance is above a maximum level, then it can not be considered to be a background reading. The default LRP background level in this system is empirically set at 0.7, while the default maximum is 0.85. Only if the LRP is between these two default levels do the above criteria apply.

Figure 7.10 shows the filtered slope measurements for all 15 places over the entire image sequence. In comparison to Figure 7.4 it can be seen that the peaks in the slope measurements representing places along the path are higher relative to the noise contained in the background levels than the corresponding peaks in the LRP data.

The LRP of places 13 and 14 in the topological produce unconvincing filtered slope

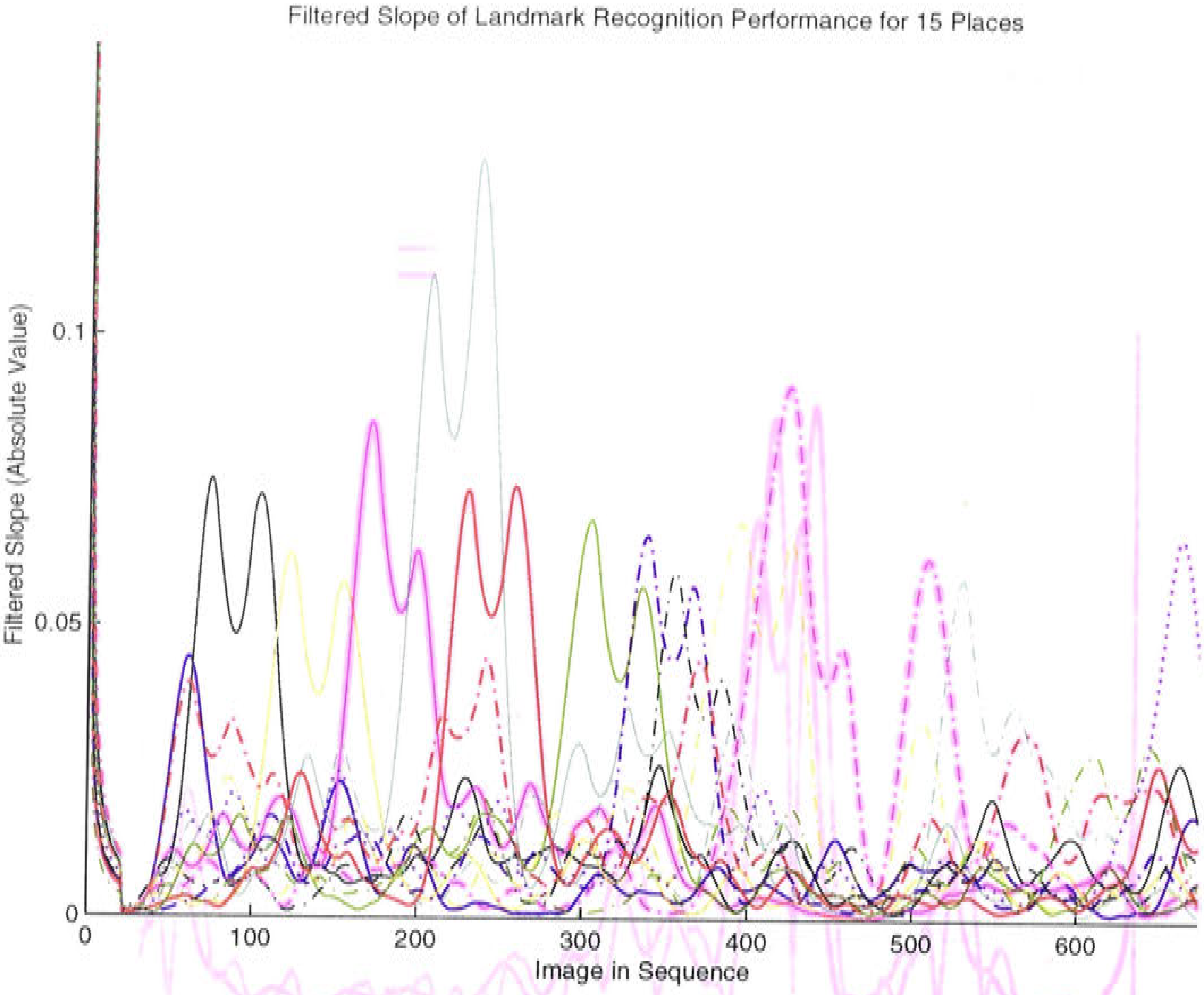


Figure 7.10: The filtered slope for the 15 places over the robot path.

results. The filtered LRP and slope measurements of these two places are shown in Figures 7.11 and 7.12 respectively. Inspection of the LRP results show that both places still have peaks associated with the reference position of each place. The peaks however are quite broad and therefore produce low slope measurements. This would indicate that the landmark sets selected to represent these two places are not unique and can produce a high LRP measure throughout a broad area of the environment. The particular area of the environment which is represented by places 13 and 14 contains filing cabinets which produces very strong and regular features. When applying this method of acquiring new places in a topological map, such low slope values would result in the acquisition of more places. Whether this is a desirable characteristic in topological map construction is left for further work. Our method is not completely reliable. The slope at which the LRP decays as the robot system moves away from the reference position will vary from place to place, although it should always be significant when compared to the background levels. The sample rate at which the robot

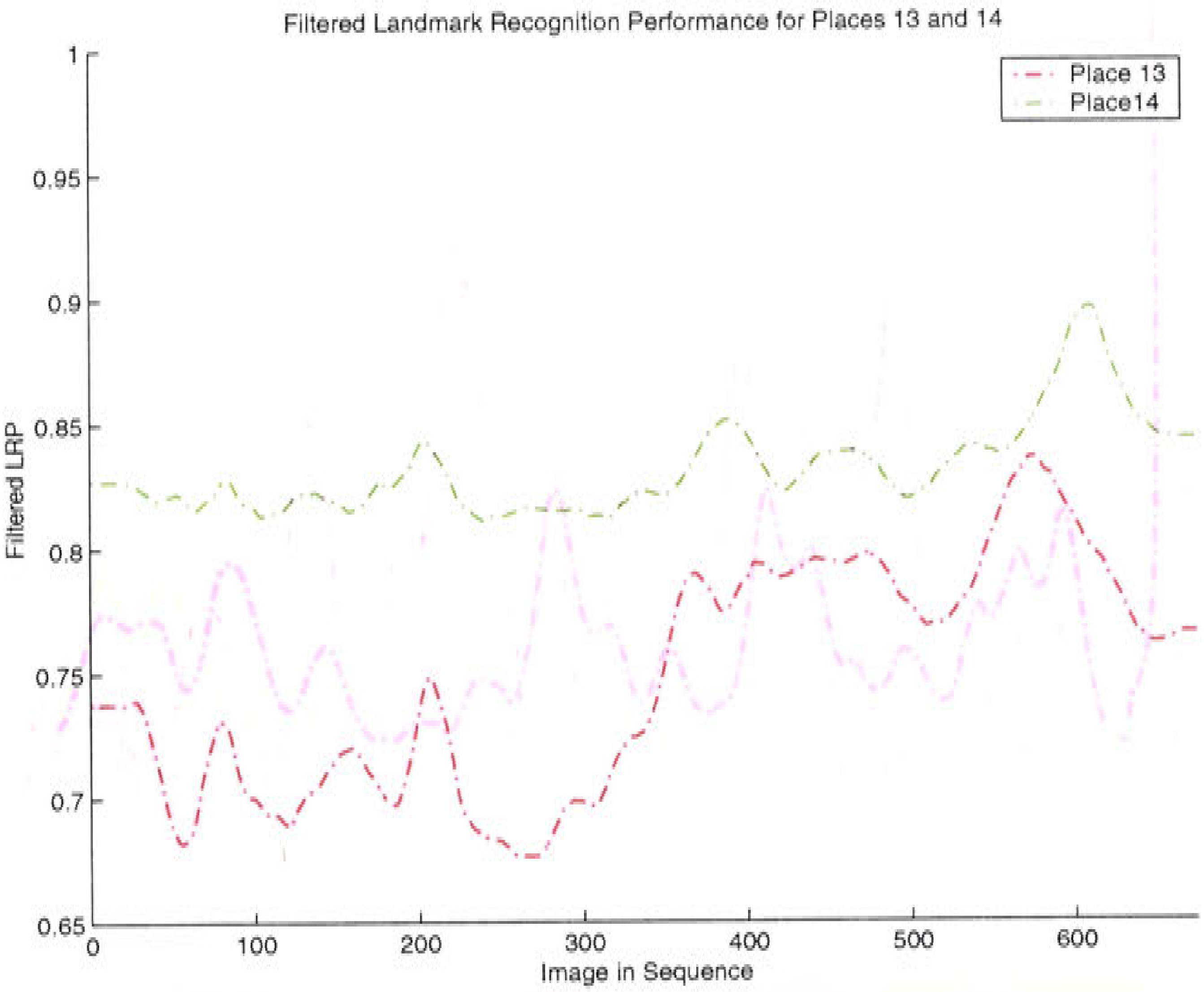


Figure 7.11: Filtered LRP for place 13 and 14.

captures the panoramic images and thus produces LRP measurements also affects the rate of change of the LRP. The direction of motion of the robot and even if the robot is moving also plays a part.

The major drawback for our method is that the LRP slope is measured over a temporal window rather than a spatial one. The measurements and subsequent decisions are made based on measurements captured over a temporal window irrespective of how the robot is moving through space. In this example all the robot movements were made at a constant velocity and either directly away from places or directly towards them. Our system accepts this limitation, noting that the detection of background levels of will only occur directly after learning a place and that motion in a topological map will usually be between nodes within the map. A modification that attempts to solve these issues for the purpose of LRP background detection is that the robot ignores LRP readings from a stationary robot, or where a robot is undergoing purely rotational motion.

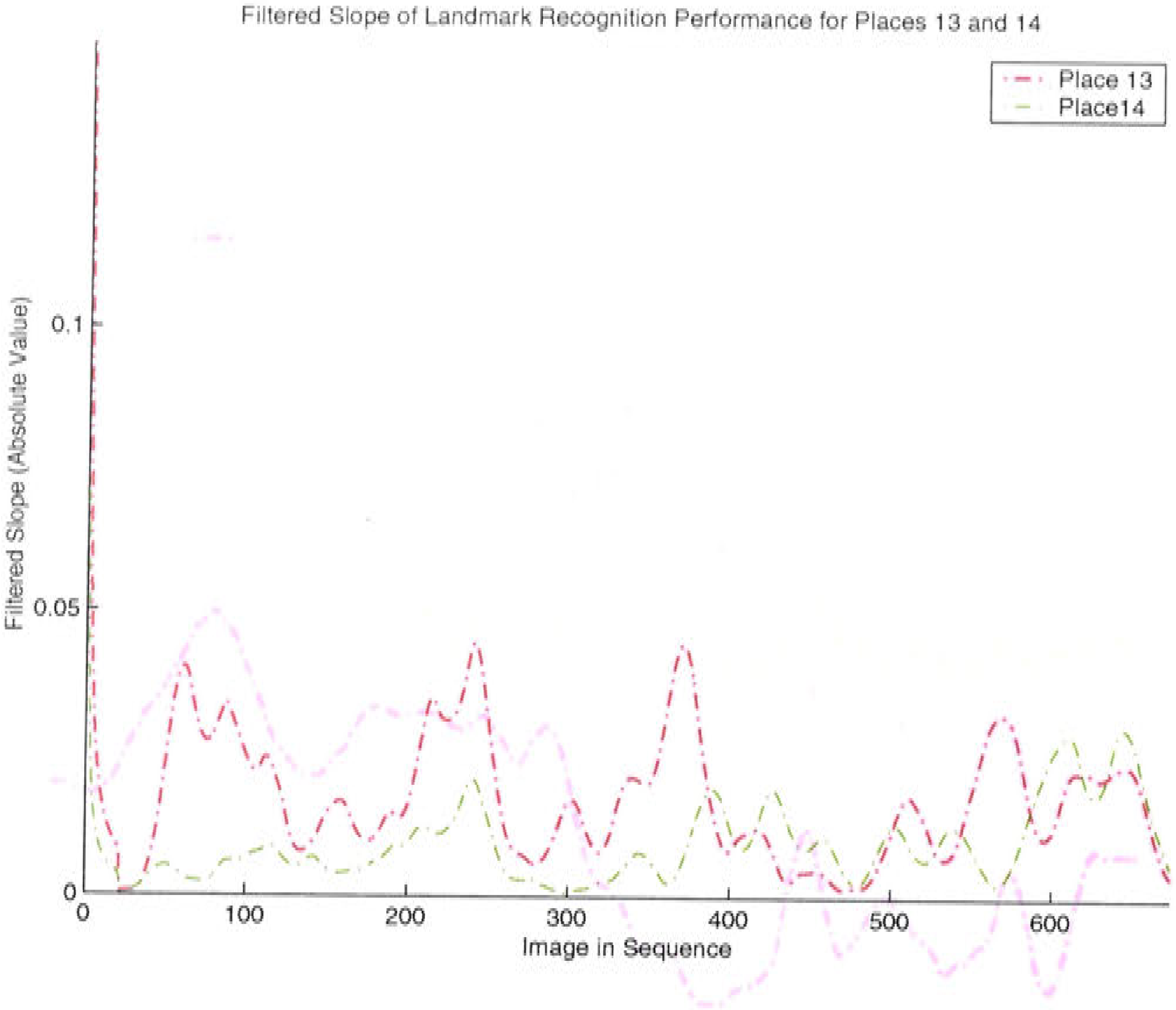


Figure 7.12: The filtered slope for the places 13 and 14

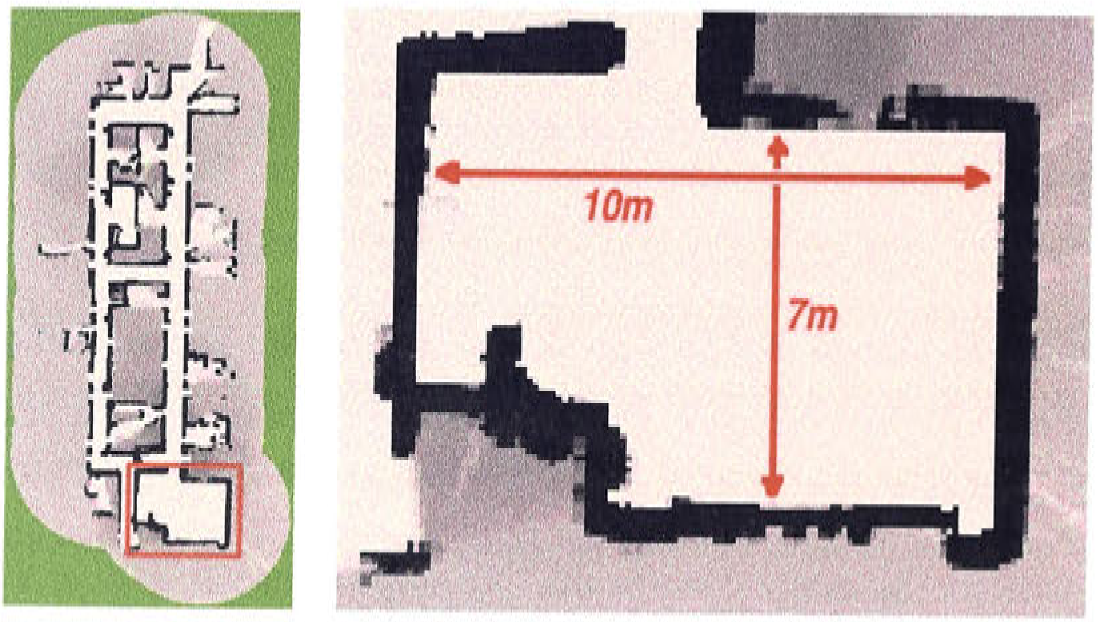


Figure 7.13: Location of place acquisition experiments.

Example Place Acquisition

This section presents an example of the robot system deciding when to learn a new place. Figure 7.13 shows a map of a large room in the the ANU laboratory in which the robot must build a topological map. Figure 7.14 shows a short path (green dots)

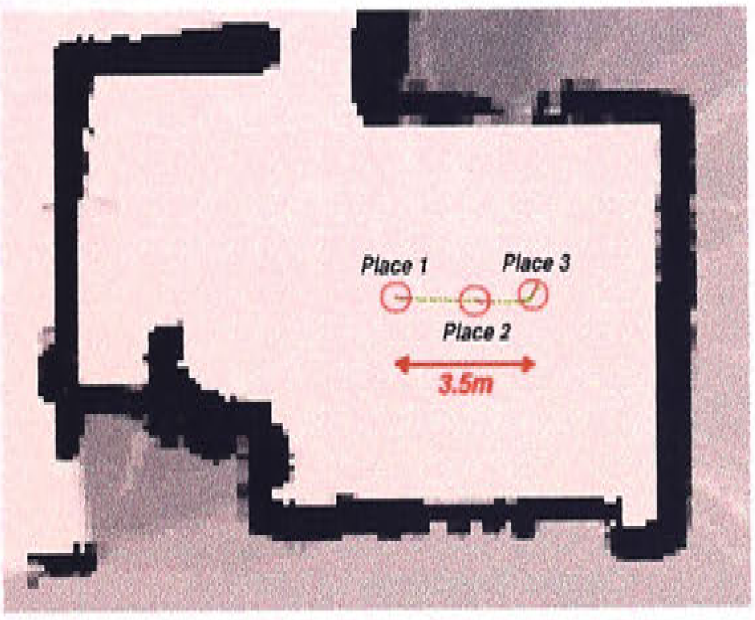


Figure 7.14: An example of place acquisition in a topological map constructed by a robot system operating in a real world environment. Three places were acquired over a short robot path. The robot path is shown by the green dots originating from place 1.

the robot system traversed. While moving along the path the robot was building a topological map using the place acquisition criteria described above. Three places were learnt along the path, shown by the red circles and text in the illustration. At each of these positions, the robot system performed a Turn Back and Look movement and acquired a new place in the topological map. This was an online experiment, with the robot autonomously, in real time, deciding when to acquire new places.

Figure 7.15 shows the flow of control that the robot system uses to perform a movement while constructing a topological map. The variable DEFAULT_BG refers to the default background LRP level. In this experiment, DEFAULT_BG was set to 0.7 and the filtered slope threshold level was set at 0.005.

The value of the LRP and its associated slope were logged as the robot traversed the path. Figure 7.16 shows the raw (blue line) and filtered (red line) LRP performance of the most likely place for each each image captured along the path. Each black circle identifies where the robot system decided to learn a new place. Figure 7.17 shows the raw and filtered slope associated with the LRP measurements. Again the black circles identify where new places were learnt.

The three places were each acquired in different circumstances, which are detailed below:

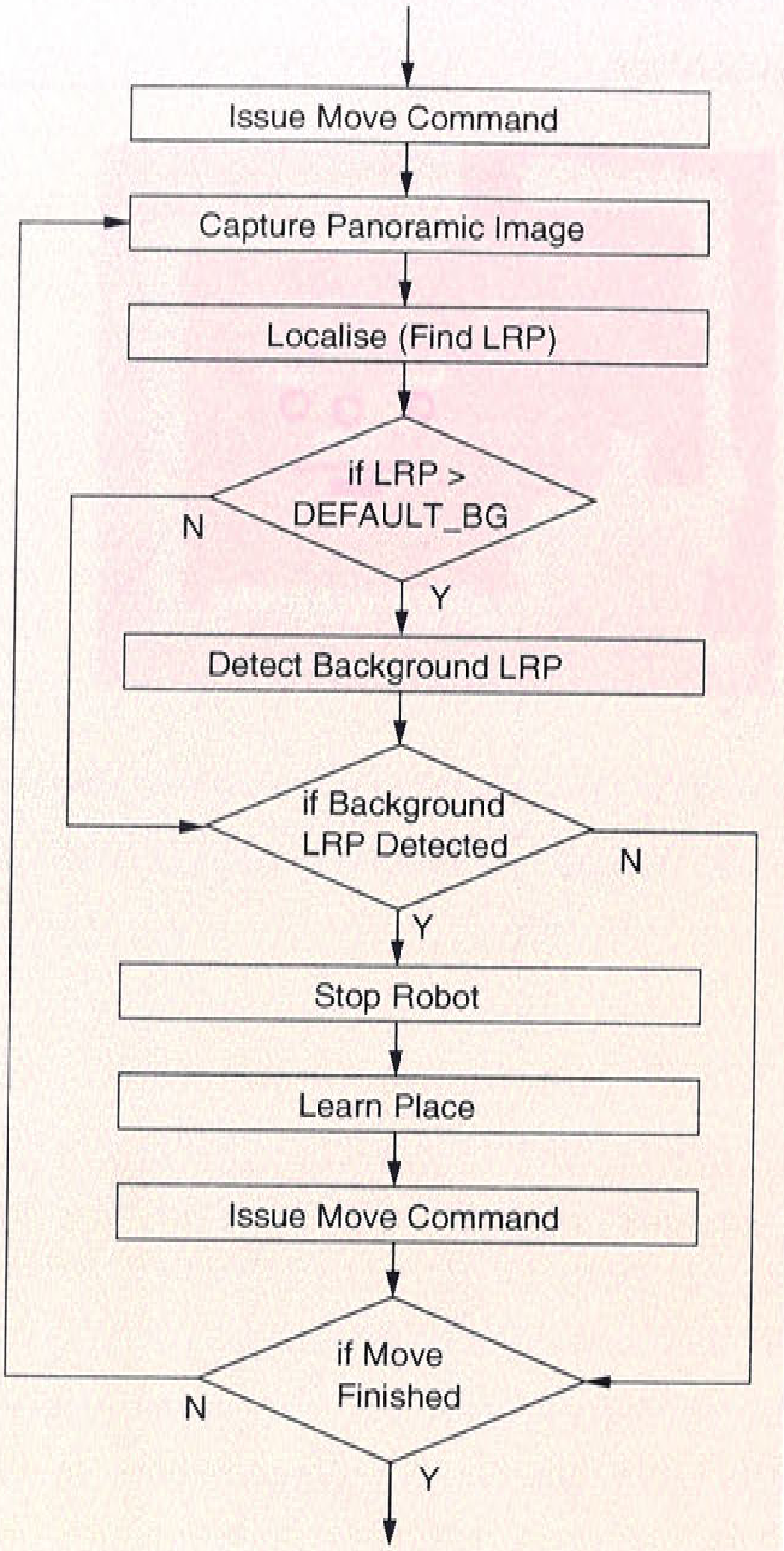


Figure 7.15: The flow of control in the robot system when deciding weather to learn a new place.

- *Place 1:* The robot starts by trying to localise itself with its existing knowledge. Not being able to localise itself in an empty map, the robot learns the first place. This is identified as place 1 in the figures. The robot attempts localisation again, and identifies that place 1 is the place in its map that the robot is the MLP.

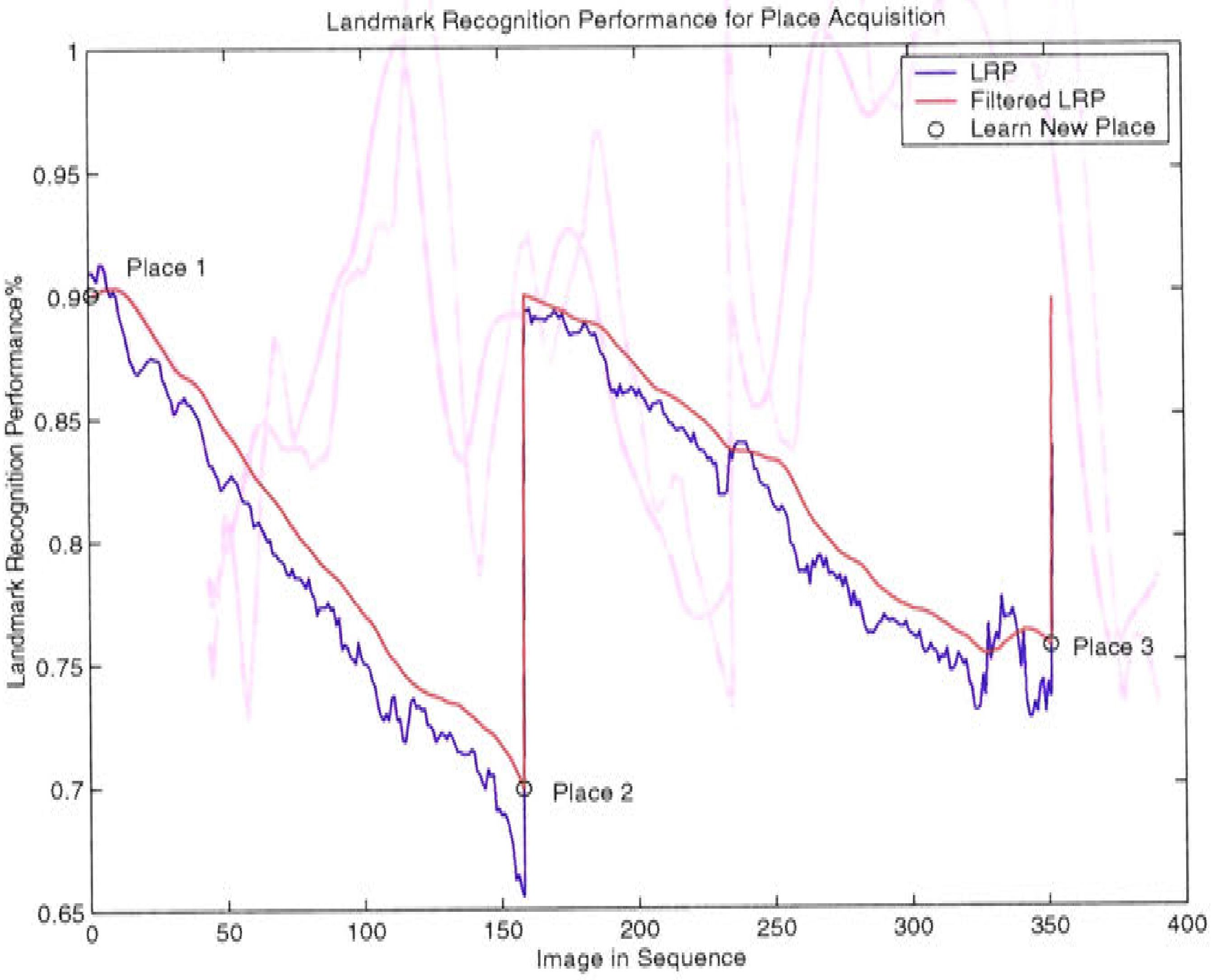


Figure 7.16: The Landmark Recognition Performance for the most likely place while building the example topological map. The black circles denote when along the robot path each place was acquired.

- *Place 2:* From place 1 the robot proceeds to move, monitoring its landmark recognition performance and trying to estimate the background LRP level. As the robot moves away from place 1 the LRP levels falls away as the current view of the landmarks distort. The slope remains relatively stable until it starts falling around the $120th$ image in the sequence where the LRP graph starts to bottom out. It never reaches the slope threshold level however as the filtered LRP level falls below the default background level first. This observation causes the robot system to stop and start the acquisition of place 2. The Localisation procedure now identifies place 2 as the MLP.

- *Place 3:* The robot system recommences its path traversal while monitoring the LRP for place 2 and estimating the background LRP as previously described. The slope of the filtered LRP graph eventually falls below the slope threshold level and the triggers the acquisition of place 3.

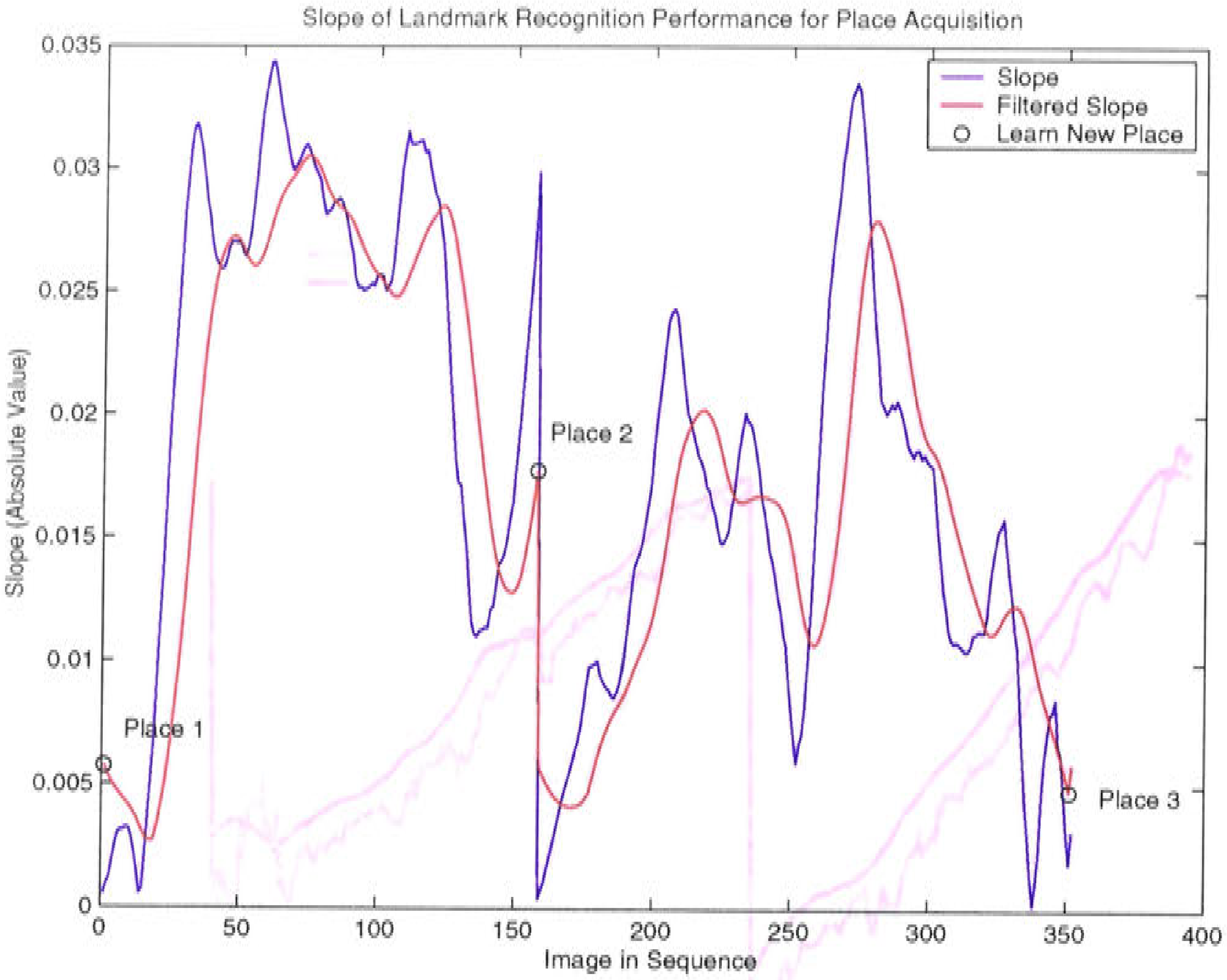


Figure 7.17: The slope of the filtered LRP measurements of the most likely place while building the example topological map. The black circles denote when each place was learnt.

Using this scheme the robot system can decide when to learn new places and add nodes to it's topological map. This approach differs to Matsumoto et al.'s (1997) in that an adaptive correlation threshold level for place acquisition is used rather than a constant threshold.

Other Possible Methods

As hinted in the previous section another way to decide when to learn a new place is to monitor the landmark recognition performance background levels with respect to the distance the robot is away from the reference position from which the current most likely place was learnt. This would eliminate the need to assume that the robot always heads directly away from a learnt place in the topological map at a constant velocity while sampling sensor data at a constant rate.

Another possible method is to learn a new place when the position estimation within the current MLP in the topological map provides uncertain position information. As opposed to monitoring whether the landmark sets used to recognise a particular place are tracking well, an analysis of the uncertainty contained in a local position estimate derived from the landmark observations could be used to determine place acquisition. In other words, learning a new place could be triggered when the robot can no longer be certain about its local position estimate. This change in reasoning could lead to topological maps which allow for a greater level of position estimation accuracy. One foreseeable disadvantage of this would be creating fine grained maps of areas in the environment which do not provide much positional information, such as corridors. In these environments, a certain degree of uncertainty in position estimation is acceptable as long as it can be resolved upon encountering unique regions.

The ability to provide an uncertainty measure while estimating local position is dependent on the method used. For example if the position estimate was being maintained by a Kalman Filter model, the uncertainty of the estimate could be obtained from the covariance matrix. In our research we use a particle filter approach to position estimation, the details of which will be presented in the next chapter. This approach has significant benefits such as the ability to represent multi-modal distributions, but has the disadvantage of not providing an uncertainty estimate without costly data analysis on the spread of particles. Therefore a place acquisition approach which incorporates local positioning uncertainty is not attempted here.

Our research accepts the reduction in position accuracy of the LRP analysis method in the anticipation of the anticipation of fast position estimation and a sparser topological map leading to more efficient global localisation.

7.1.2 Defining Transitions

In addition to storing a representation of locations in the environment, a topological map must also record the connections between adjacent nodes. Using these connections, the robot system can plan paths throughout the map, navigate between two adjacent nodes and also predict position estimates when moving between nodes. In this system, such connections are called transitions, and they are calculated every time a new node is added to the map.

Transitions between places ($T_{S_i \to S_j}$) are represented by a distance and heading measure (d, γ) defined relative to the learnt place's reference position.

$$T_{(S_i \to S_j)} = (d, \gamma)$$

A transition is defined as a connection between two places in the topological map. When attempting to calculate heading and distance values for a transition it is assumed that the robot system has a stable local position estimate in relation to the last most likely place the robot has visited. For example when moving away from a given place S_i and making the decision to learn a new place S_{i+1}, the robot system has an estimate of its position relative to place S_i.

If this assumption holds, a transition can be calculated using simple geometry. Given two places S_1 and S_2 the transition $T_{(S_1 \to S_2)}$ can be defined as follows:

$$\begin{aligned} T_{(S_1 \to S_2)} &= (d_1, \gamma_1) \\ &= \left(\sqrt{x_1^2 + y_1^2}, tan^{-1}\left(\frac{y}{x}\right)\right) \end{aligned} \tag{7.6}$$

where the place S_2 is learnt at the coordinates (x_1, y_1, θ_1) relative to the reference frame defined byS_1. Of course this only defines a one way transition from place 1 to place 2, $T_{(S_1 \to S_2)}$. In order to maintain bidirectional connectivity within the topological map, more calculations must be performed to form the transition from place 2 back to place 1:

$$\begin{aligned} T_{(S_2 \to S_1)} &= (d_2, \gamma_2) \\ &= (d_1, (\gamma_1 - \theta_1) + \pi) \end{aligned} \tag{7.7}$$

which provides a transition which is relative to the new places reference frame. It should be noted here that θ_1 is the angle of the x axis of place S_2's reference frame with respect to place S_1's reference frame.

An example which illustrates Equations 7.6 and 7.7 is given in Figure 7.18. When a new place is learnt, transitions are defined from the last visited place to the new place and vice versa.

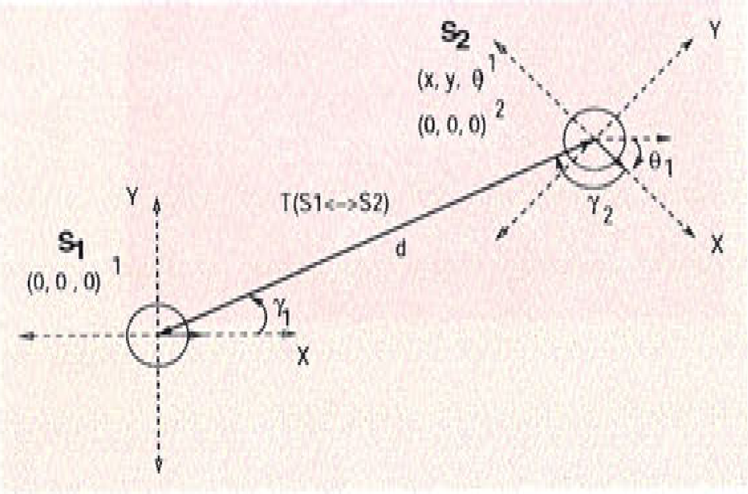


Figure 7.18: The geometry of a transition in a topological map. Two places S_1 and S_2 are connected by the transition $T_{S_1 \leftrightarrow S_2}$.

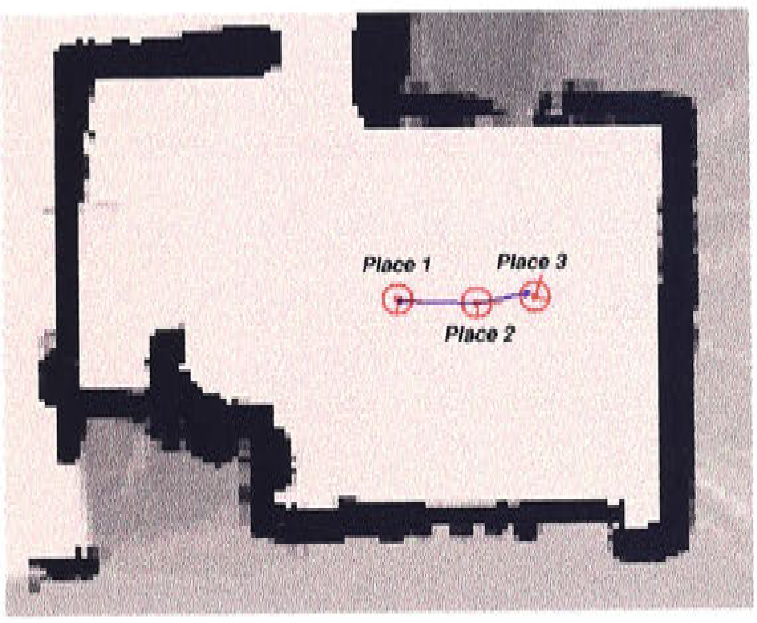


Figure 7.19: Transitions defined from place 2 in the topological map.

Figure 7.19 gives an example of transitions defined according to the implementation details described above. The topological map that is used was learnt in the example in the previous section. In the figure the transitions originating from place 2 can be seen. The transition connecting place 2 to place 1 is an example of a reverse transition, defined when place 2 was learnt, and is very accurate. The connection from place 2 to place 3 is less accurate due to the errors in odometry and position estimation introduced by the rotation in the robots position along the path between places 2 and 3. Although the direction is slightly off, following the transition from place 1 would still position the robot in a location from which it would be able to recognise place 3.

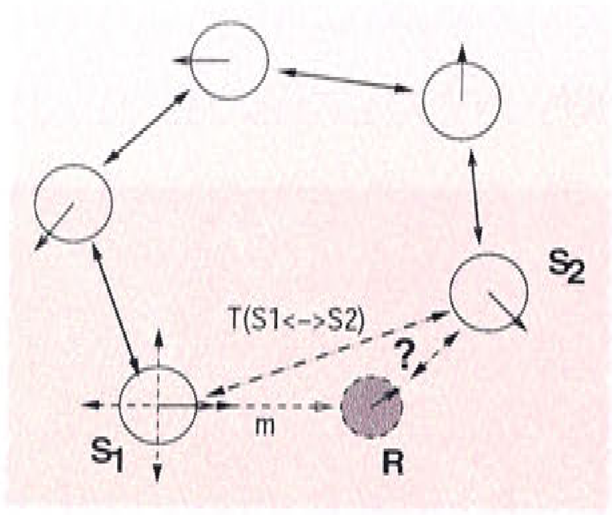


Figure 7.20: An example of a cycle in a topological map. The robot at point R relative to place S_1 has detected a connection between places S_1 and S_2 but does not know the spatial relationship between the two.

7.1.3 Transitions and Cycles

This method of defining transitions works when adding a new node in the topological map depends on knowledge of the relative position of one place to the other. The method fails when a cycle is detected in the graph. Consider the case when a robot is moving away from place S_1 and encounters a previously learnt place S_2, when there is no existing transition connecting the two. The encounter is detected, when the LRP for place S_2 becomes higher than that of S_1 thus signifying a change in the MLP that the robot is occupying in the topological map. The fact that there is no existing connection between the two means that although the robot has an estimate of its current position relative to place S_1, it has no knowledge of the relative position of either place with respect to the other. This situation will most likely occur when the robot detects a cycle in the graph, or when it is connecting two previously disconnected sections of the topological map.

Figure 7.20 shows an example of when a cycle is detected and illustrates the problem of the two places not sharing a common frame of reference. In this figure, the robot R has moved along the vector m away from place S_1. It has now recognised the landmark set from place S_2 but has no idea where it is in relation to the reference position from where place S_2 was learnt. Therefore the transition from S_1 to S_2 can not be defined.

One approach is to try and infer the spatial relationship between the two places by

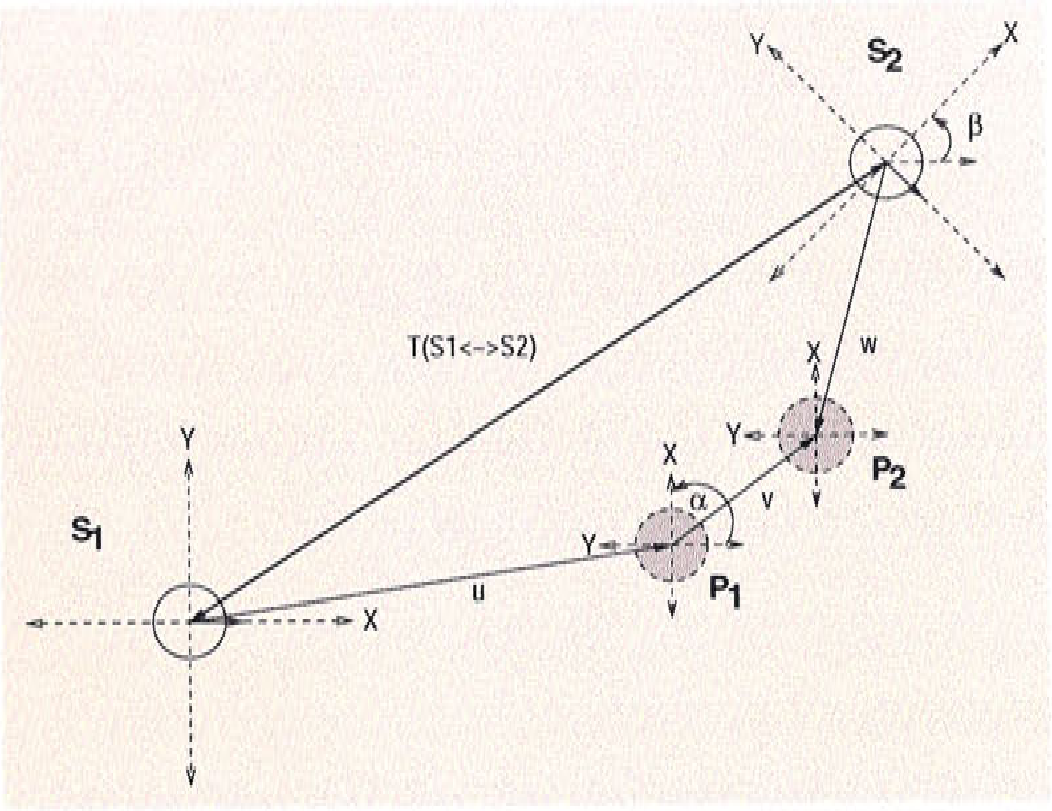


Figure 7.21: The geometry of defining a transition between pre-existing places in a topological map. S_1 and S_2 are two pre-existing places in a topological map, while P_1 and P_2 are local position estimates from the reference positions of S_1 and S_2 respectively.

summing the transition vectors of the nodes in the topological graph which create the cycle. This has two problems: first it assumes a cycle does exist and that the two nodes do not belong to two separate and disconnected sub-graphs; and second, it assumes that there are no odometry errors. The individual connections are relative and contain levels of error which when summed might produce drastic inconsistencies in the topology of the map, especially when dealing with a large number of nodes in the connecting path.

A better approach that does not rely on the connecting path, is to directly discover the spatial relationship between the two nodes and use this relationship to calculate the transitions between the two places.

In order to successfully calculate the transition between S_1 and S_2, by discovering the spatial relationship between the two places, the robot must first localise itself within S_2. By continuing its movement and observing the angles of landmarks the robot system can produce a local position estimate, the details of how this is achieved are presented in the next chapter. Also by using odometry to keep track of its movement while between places, the robot now has three vectors which when combined define the spatial relationship of the two places: $\overrightarrow{u}$ the initial local position estimate from place S_1; $\overrightarrow{v}$ the odometric vector maintained while local positioning is attempted; and $\overrightarrow{w}$ the local position estimate from S_2.

Figure 7.21 illustrates the situation further and identifies the three vectors $\overrightarrow{u}$, $\overrightarrow{v}$ and $\overrightarrow{w}$. In the figure P_1 and P_2 are the two points between which the robot is relying on solely odometric information. Given that S_1, S_2 and the odometry measurements are all relative to different frames of reference it is impossible to calculate $T_{(S_1 \to S_2)}$ without first resolving the differences in coordinate systems. From here on the notation A_C^B will refer to point or vector A_C with respect to coordinate system B. Therefore, the three vectors as they stand can be more correctly defined as:

$$\overrightarrow{u}^{S_1}, \overrightarrow{v}^{S_O}, \overrightarrow{w}^{S_2}$$

where the super scripts S_1 and S_2 refer to the coordinate systems relative to places S_1 and S_2 respectively, and O refers to the odometric frame of reference. With this notation the transition can be defined as follows:

$$T_{(S_1 \to S_2)}^{S_1} = \overrightarrow{u}^{S_1} + \overrightarrow{v}^{S_1} - \overrightarrow{w}^{S_1} \tag{7.8}$$

It should be noted here that the frames of reference of the last two known vectors are different to those used to define the transition. If $F_\phi(x)$ is a linear transformation which rotates a point x through an angle of ϕ, then vectors $\overrightarrow{v}$ and $\overrightarrow{w}$ can be converted to S_1's reference frame as required by Equation 7.8:

$$T_{(S_1 \to S_2)}^{S_1} = \overrightarrow{u}^{S_1} + F_\alpha(\overrightarrow{v}^{S_O}) - F_\beta(\overrightarrow{w}^{S_2}) \tag{7.9}$$

where α and β are the respective difference between coordinate systems of S_1 relative to O, and S_1 relative to S_2. The two angles can be calculated as follows:

$$\alpha = \theta_{P_1}^{S_1} - \theta_{P_1}^{O} \tag{7.10}$$

$$\beta = (\theta_{P_2}^{O} + \alpha) - \theta_{P_2}^{S_2} \tag{7.11}$$

where, as above $\theta_{P_1}^{S_1}$ refers to the robot's orientation θ at point P_1 with respect to the coordinate system S_1.

In this way the transition from S_1 to S_2 can be defined. The reverse transition can then be calculated in the same manner as calculating the reverse transition for newly acquired places presented above. Because the calculation of cycle transitions is based on odometry between local position estimates, the odometric error is small.

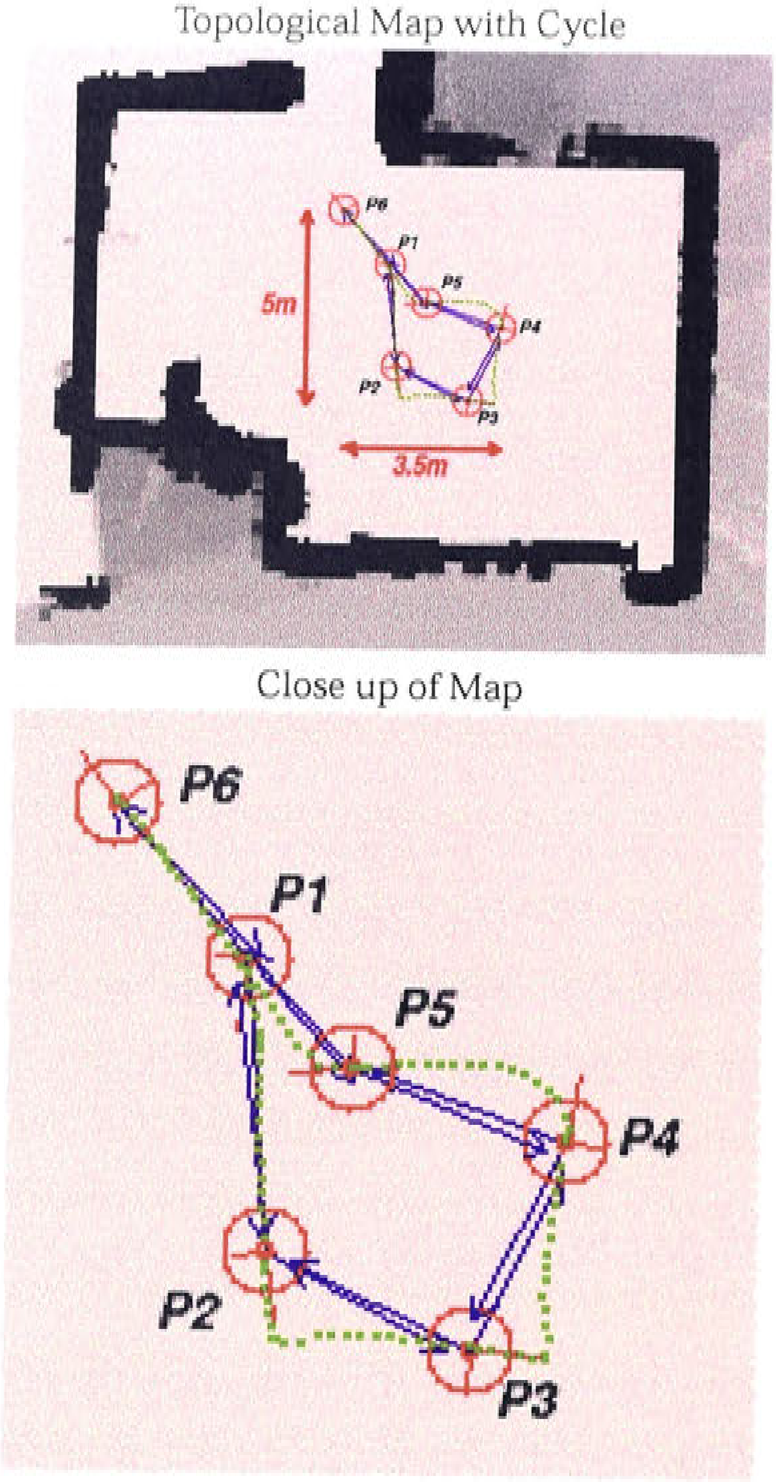
Implementing this on the robot system requires the robot to be able to be able to localise within a place, and also to decide when a reliable local estimate has been achieved. The details of localisation will be discussed in the next chapter, but for now local positioning will be assumed, with the proviso that it takes a number of sample observations for a reliable estimation. An example of forming a cyclical connection in a real world topological map is presented in the next section.

7.2 An Example Topological Map

This section presents a topological map constructed using the place acquisition and transition definition methods described in the previous two sections. The map was learnt in a large room in the Robotic Systems Laboratory at the Australian National University. The robot system traversed a predefined path while attempting to learn a topological map. The resulting map is shown in Figure 7.22. This figure shows the robot path in green, the places learnt along the path in red, and the transitions which connect the places in blue. The orientation at which places were learnt is shown by the red lines radiating from the center of each place circle, with the longer line representing the angle of the robot facing forward.

The graph shows that places were learnt along the path at semi-regular intervals and transitions were defined which link each node in the graph with its adjacent nodes. A cycle was detected between place 5 and place 1, and the resulting transitions successfully capture the relative spatial relationship between those two places.

Although this is a relatively small topological map, the methods used here can be used to build arbitrarily large maps. The methods of place acquisition and transition definition are not directly dependent on the size of the map. In practice however, both of these processes depend on successful localisation within the map, effectively limiting


Topological Map with Cycle

Close up of Map

Figure 7.22: An example topological map constructed in a real world environment. The robot started at the location labelled place 1 and moved along the green path, acquiring new places in the map along the way. Places are shown by the red circles with the reference orientation shown by the red lines (longer line facing forward) and transitions between places by the blue arrows. Note that a cycle was detected as the robot leaves place 5 and the corresponding transitions were correctly defined.

the map to the number of places with which it can localise within the necessary real-time constraints. A larger map constructed using the previously described methods is presented in Chapter 9.

7.3 Summary

This chapter presented a method with which to build topological maps using the place representation described in earlier chapters. Topological maps need to hold representations of places and the transitions between those them. Place representations allow for the robot system to recognise places when revisiting them, while transitions allows for navigation between places in the environment. The two main steps in constructing a topological map, apart from forming a representation of individual places, is deciding when to learn a new place, and defining transitions connecting places.

Determination of when to learn a new place can be achieved by monitoring how well the robot system recognises its current environment. When the robot no longer recognises its current environment a new place is added to the topological map. A method was presented for making this determination using the level of Landmark Recognition Performance (LRP) and the notion of background levels of recognition.

Transitions can be defined by simple geometric calculations in conjunction with global localisation and local position estimation. Detecting cycles in the topological graph cause complications to these calculations, however methods were described which successfully overcame these problems by working out the spatial relationship between the two connecting places.

Together, these two processes allow the construction of topological maps, and a small example of such a map constructed by our robot system operating in a real world environment was presented.

The topological maps constructed using the methods detailed in this chapter can be used to experimentally verify the place representations and localisation concepts which are central to this thesis.

As discussed above, the addition of nodes using the back ground detection method suffers from monitoring the LRP over time, rather than through space. To truly detect when the background level of recognition performance, some account of velocity, both speed and direction, and the data sampling rate must be taken into consideration. A tighter integration of the three sub-components of navigation: mapping, localisation and path planning, would lead to place acquisition determination methods which produce topological maps with a more descriptive and efficient topology.

The method of defining transitions was shown to work accurately. One disadvantage of our system is that it is static. Defined transitions are not refined, and no method of deleting or updating them has been attempted. Revision of transitions could easily be accomplished by weighting the transition estimates and adjusting them every-time the robot system attempts to navigate using the transition information.

The implementation of a robot system which forms topological maps in real world environments highlights the difficulties inherent in the task, some of which have been discussed. Determining when to learn a new place, which seemingly is an 'easy' problem, quickly becomes complex. Indeed forming some comprehensive rules which maximise graph structure in terms of minimising the costs involved in mapping, localisation and path planning is an interesting research topic in itself.

Future work in this area would involve investigating the relationship between the topological structure and the individual subcomponents of navigation applied to the task of determining when to learn a new place. This could lead to topological maps which support efficient and accurate localisation methods.

Chapter 8

Local Position Estimation

The process of localisation requires a robot to answer the question of "Where am I?" By defining a multi-level representation of spatial knowledge, it is hoped that robots will be be able to answer this question reliably and efficiently. As stated in Chapter 2, localisation has two distinct parts: global localisation and local position estimation. Global localisation is the process of identifying the Most Likely Place (MLP) of the robot over the entire internal map. This global estimate can be coarse as local position estimation serves to refine and maintain the robots position from this localised region. Our research proposes to use the low-level representation of visual landmarks to perform both global localisation between places in the topological map, and local position estimation within particular places. The higher levels are there to 1) restrict the global localisation search, and 2) disambiguate between equally likely places.

This chapter describes the process of local position estimation using visual landmarks in detail and reports supporting experimental results. Section 8.1 describes heuristic and probabilistic approaches to local position estimation within a place. Experimental results highlight the benefits of using probabilistic position estimation especially in conjunction with knowledge of landmark depth and an appropriate sensor model. Section 8.2 describes a method of passing the local position estimate between places to maintain position tracking as a robot navigates between nodes in the topological map. Section 8.3 provides a summary of using visual landmarks for local position estimation.

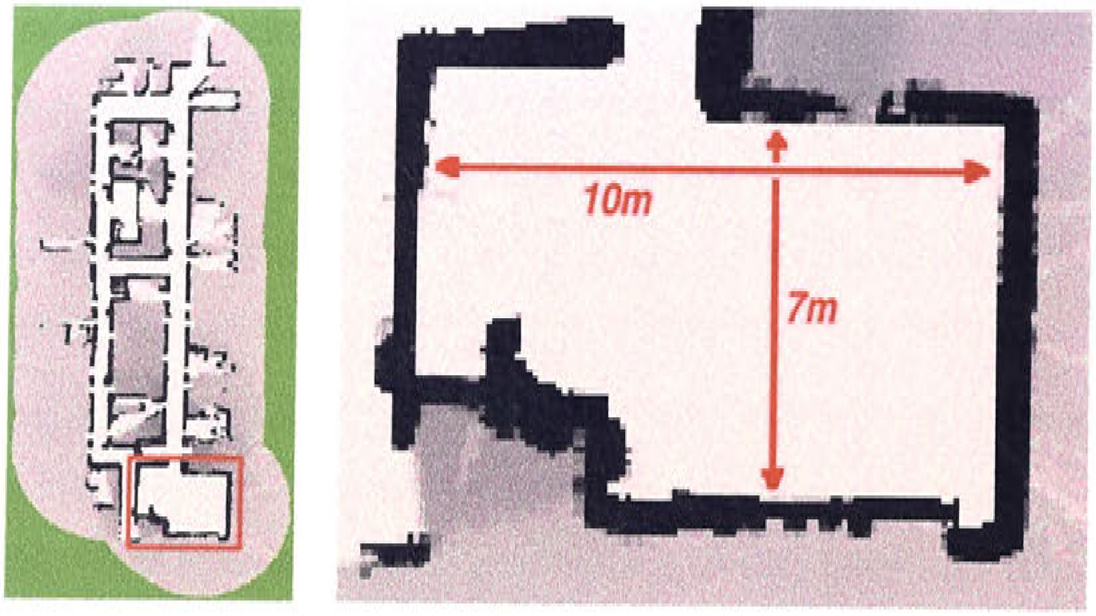


Figure 8.1: Location of local position estimation experiments.

8.1 Local Positioning Within Places

Approaches to the robot navigation problem which use topological maps typically can not provide accurate position information within places in the map: either a robot is "at" a particular place or it is not. In order to develop a topological map based localisation system which can approach the accuracy given by metric maps, the robot system must be able to locate itself relative to the reference position of places within the map. We refer to this process as local positioning within places in the topological map.

This section reports on three methods of local position estimation within a learnt place: odometric, heuristic and probabilistic. All three experiments involve the robot learning a place in the center of the same room (Figure 8.1) as reported in the previous chapter and then subsequently estimating position along a path of captured images originating at the learnt place and following the typical TBL path described in Chapter 4. Odometric positioning is used as a ground truth here as the motion of the path is insufficient to introduce any significant odometric error. It can be appreciated however, that odometric positioning will not be useful for positioning on longer motion paths. Sensor based localisation approaches (e.g. the heuristic and probabilistic approaches) do not suffer from these problems. The robot path as determined from odometric position estimation is shown in Figure 8.2.

The goal of the local positioning experiments described below is to successfully deter-

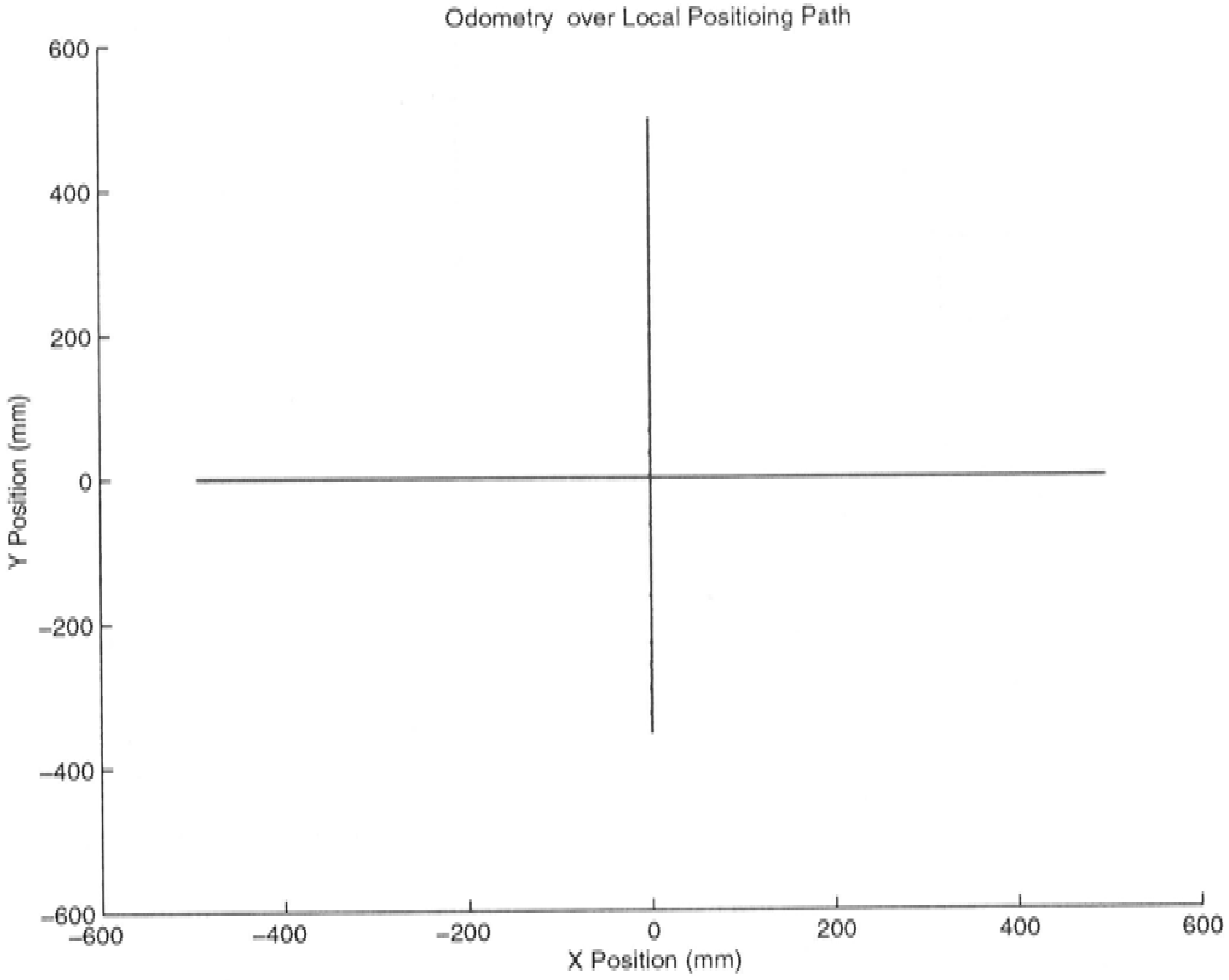


Figure 8.2: Local positioning along TBL path using odometry. Axes are in *mm*.

mine the path the robot travelled with no prior knowledge about the pose of the robot within the area surrounding the learnt place's reference position. The assumption that the initial pose is unknown makes odometry by itself unable to provide any useful localisation measure.

8.1.1 Heuristic Position Estimator

Typically navigation systems which employ topological maps to represent the environment do not attempt to extract local position estimates relative to the reference position of places in the map. Instead they rely on simple recognition/action pairings or homing behaviours which drive the robot to a place without knowledge of relative positioning. Such reactive behaviours often rely on a heuristic rule rather than explicit calculations and thus are referred to in this section as heuristic navigation systems. An example of such an algorithm is presented that attempts to extract a relative position estimation from the heuristic system.

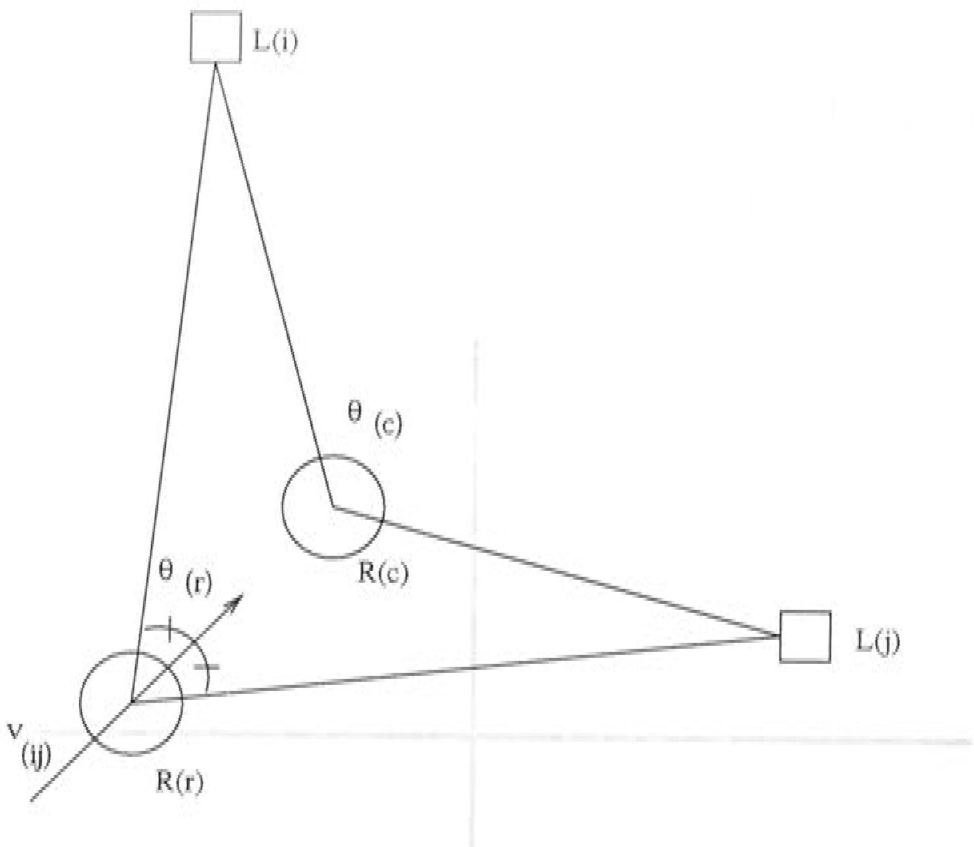


Figure 8.3: Radial contraction position algorithm: vector, v_{ij}, produced from a single landmark pair, (l_i, l_j).

This heuristic localisation method grew out observation of data during the execution of a homing behaviour that uses sensor data to drive the robot to a desired state (Bianco and Zelinsky, 1999) (Collett, 1996). The algorithm uses the contraction/expansion of the observed radial displacement between pairs of landmarks to estimate the position of the current robot state relative to a reference state. The general idea is that if the radial displacement between a landmark pair is larger in image 1 than image 2, then image 1 must have been captured at a location closer to the landmarks in the direction of the bisecting angle. The sensor gives the radial angles of located landmarks from a landmark set. Let this observation be denoted $L = l_0 ... l_n$, where l_i is the observed radial angle of landmark i. Given the two observations L_r, captured from a reference position R_r, and L_c from the current position R_c, then the problem is to estimate the translation vector $\overline{V}$ from R_r to R_c.

This can be done by summing the contraction vectors, $\overline{u_{ij}}$, for every possible landmark pair (l_i, l_j) in the landmark set. A contraction vector is the translation vector needed to cancel the change in radial displacement within a landmark pair between the two observations. An example is given in figure 8.3. More formally, $\vec{u}_j$ is composed of

$\vec{v}_{ij}(mag)$ and $\vec{v}_{ij}(dir)$,where,

$$\vec{v}_{ij}(mag) = (\theta_c - \theta_r) \tag{8.1}$$

$$\vec{v}_{ij}(dir) = \theta_r - (\theta_r/2) \tag{8.2}$$

θ_c and θ_r are the radial displacements of the landmark pair (l_i, l_j) at R_c and R_r respectively. The magnitude of the contraction vector, $\vec{v}_{ij}(mag)$, is then the change in the radial displacement and the direction $\vec{v}_{ij}(dir)$, is the angle bisecting the displacement.

Summing the contraction vectors for every landmark pair, weighted by the reliability of the observation, gives the translation vector:

$$\vec{V} = \sum_{i=0}^{n}\sum_{j=0}^{n} \vec{v}_{ij} w_{ij} \tag{8.3}$$

where

$$w_{ij} = (rel(l_i) + rel(l_j))/2 \tag{8.4}$$

$rel(l_i)$ being a reliability measure of correctly locating landmark i as determined by the results of the normalised correlation template matching process. Template matching for landmark tracking was introduced in Chapter 4 which described acquiring a visual landmark representation of places in a topological map.

The results from applying this algorithm to the captured TBL path are given in Figure 8.4 . As you would expect, this approach allows a general estimate of position but is not suitable for navigation tasks which require accurate positioning. The direction is slightly skewed, due to the irregular distribution of landmarks throughout the image and the simple vector weighting. The scale of results is unknown, although a simple scaling based on assumed depth of landmarks is given in Figure 8.4. The main problem is the sensitivity to noise as shown by the zig-zag nature of the estimated path. It is also vulnerable to false positives in landmark tracking, as the change in displacement determines the magnitude of the vectors. These characteristics are typical of attractor-based methods and do not support accurate local positioning (Arkin, 1998).

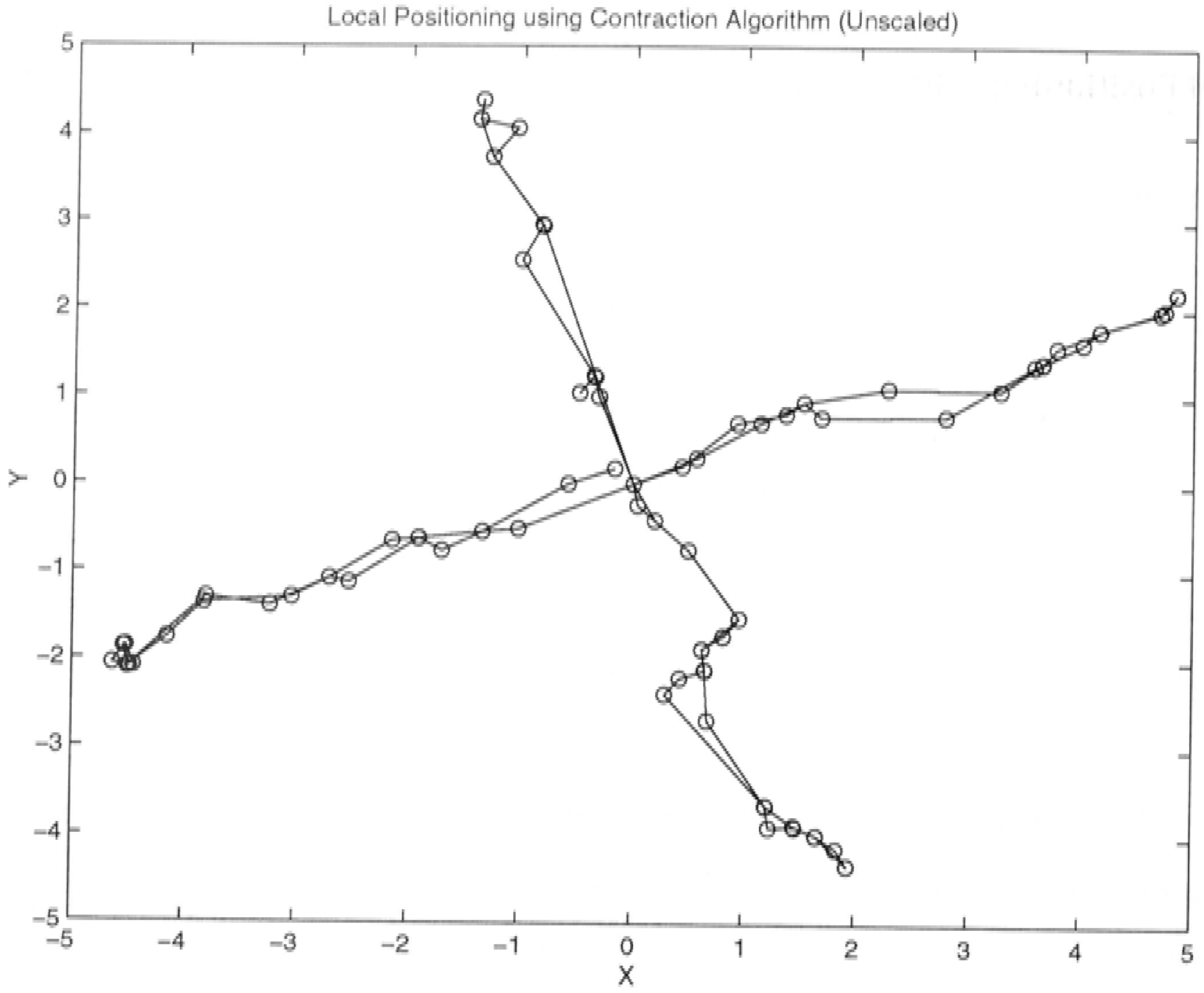


Figure 8.4: Local positioning along TBL path using heuristic algorithm

The heuristic based method for local position estimation was the first attempt of our research to perform localisation within places in a topological map.

8.1.2 Particle Filter Position Estimator

The benefits of probabilistic reasoning in mobile robot localisation has been reported in Chapter 2. By using knowledge of past observations and ego-motion, the position of the robot can be represented as a probability distribution throughout state space of possible robot poses in the environment, allowing for multi-modal hypotheses. Particles filters have been used successfully to reduce the computational requirements of generating the distribution, by random sampling of the state space. The Condensation algorithm (Isard and Blake, 1998; Dellaert, Fox, Burgard and Thrun, 1999) is one such method, and is summarised below . A set of sampled states $S_t^{(0)}...S_t^{(n)}$ (where $n =$ the number of particles or samples used), and their associated probabilities $(\pi_t^{(0)}...\pi_t^{(n)})$, are used to approximate the probability distribution at time t. At each iteration of the

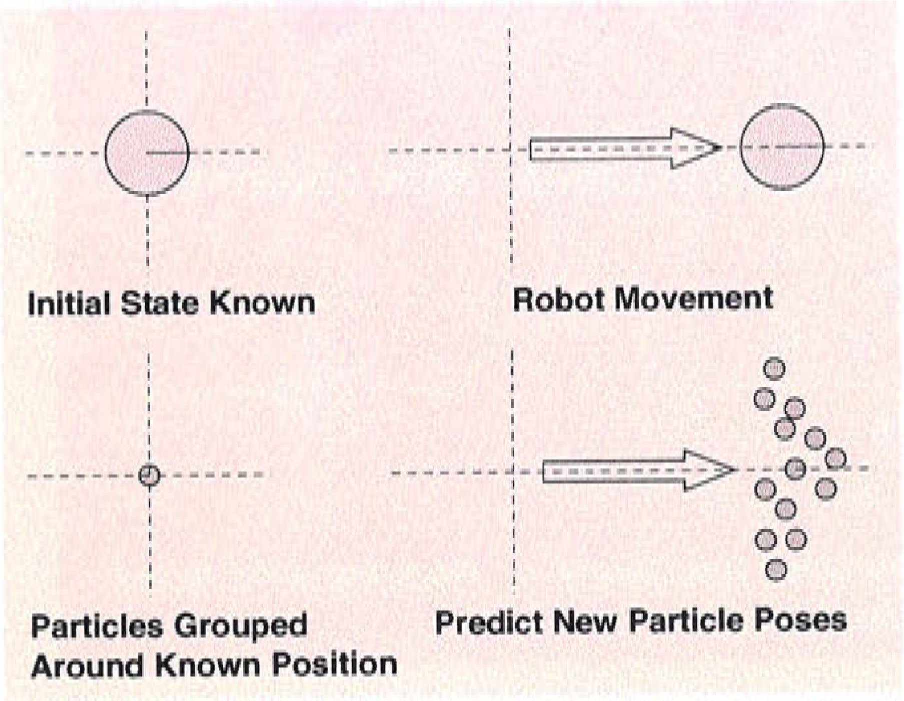


Figure 8.5: Prediction phase of the Condensation algorithm. Particles are initially grouped around a known position. After a noisy motion input, the particles are distributed according to the stochastic noise and deterministic drift in the motion model.

algorithm, the following steps are applied:

1. *Re-sample:* For each particle $S_t^{\prime(i)}$ in S_t', select a random particle $S_{t-1}^{(j)}$ from S_{t-1}. This re-sampling is done with replacement and probability of selecting $S_{t-1}^{(j)}$ is given by $\pi_{t-1}^{(j)}$.
2. *Predict:* For each particle $S_t^{\prime(i)}$ in S_t' predict a new state $S_t^{(i)}$ in S_t by applying the motion model.
3. *Measure:* For each particle $S_t^{(i)}$ in S_t, evaluate the probability $\pi_t^{(i)}$ using the sensor model.

The motion model describes the dynamics of the system to predict the state of a particle after an action has been taken. This model incorporates deterministic drift and stochastic noise into the predictions, capturing the noisy nature of odometric sensors used in this system. Figure 8.5 demonstrates what happens to particles when an action is taken by the robot system. In this illustration the robot system is a mobile robot which starts at a known position, in this case the origin of the drawn coordinate system, and moves to the right. The accompanying particle filter diagrams show the particles initially tightly grouped about the origin, denoting that the initial position is

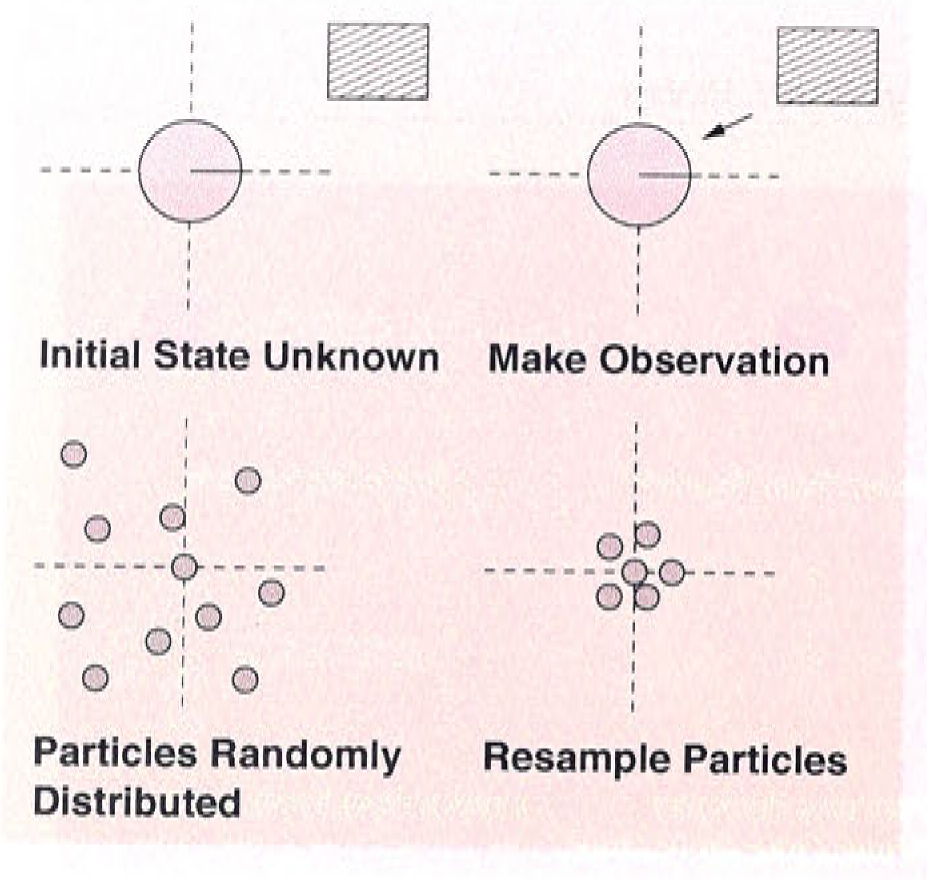


Figure 8.6: Measurement and re-sampling steps of the Condensation algorithm. Particles distributed randomly condense about a hypothesis after an observation has been made. Particles condense around a hypothesis by resampling the particle set according to the probability of making the current observations given the particle's state.

known. After the move the particles are spread out in an arc which approximates the probability density function of the robot position in the presence of noise introduced by the errors in rotational and translation motion of the mobile robot.

The sensor model is used to calculate the probability that a given particle correctly describes the system state based on the current observation. Figure 8.6 demonstrates how a sensor model and re-selection of probable particles can be used to maintain state identification in a particle filter system. In this illustration the pose of the robot is initially unknown. Accordingly, the particles used to represent the probability distribution function of the robot's position are randomly distributed throughout the space defined by the coordinate system. When the robot makes an observation the sensor model can be used to measure the probability that a robot system in state described by each individual particle could have made such an observation. Particles representing robot states which have a high probability of making the current observation have more more chance of being reselected in the re-sampling phase of the Condensation algorithm. This is shown in the figure by the set of particles, after incorporating the

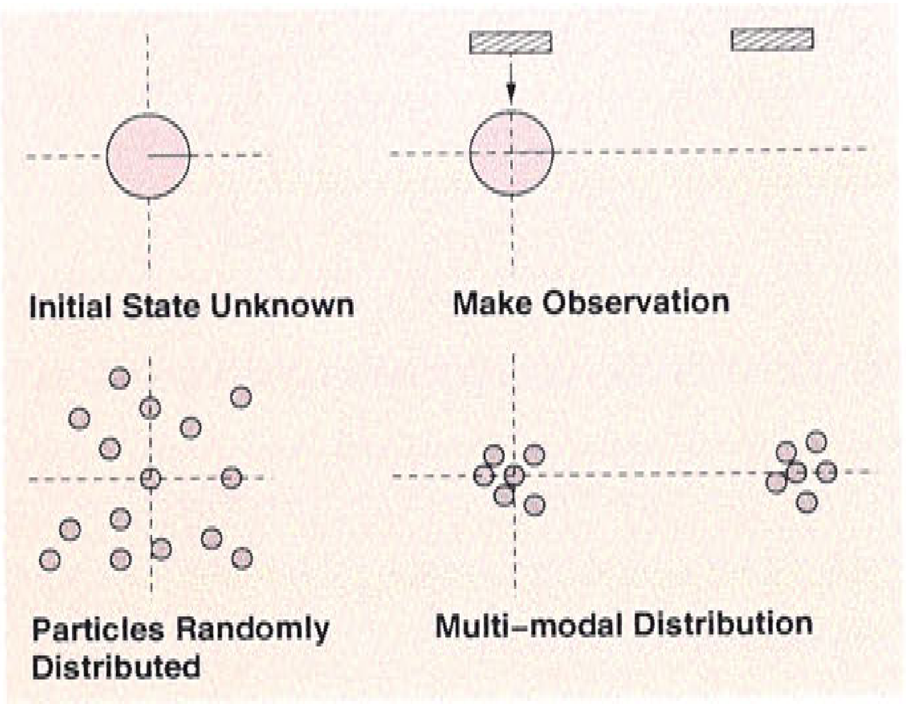


Figure 8.7: The particle filter approach to mobile robot localisation can represent multi-modal distributions in the presence of ambiguous observations.

observation, condensing around the correct position in the coordinate system.

Another feature of particle filters are that they can represent multi-modal hypotheses when the sensor information is ambiguous. Figure 8.7 illustrates this case. Initially the robot position is unknown and the particles are distributed randomly throughout the sample space. After making an observation the probability of making such an observation is calculated for each particle and the most likely particles are re-sampled. In this example, however, they are two positions (e.g. two doorways) which are equally likely to result in the current observation and this is reflected in the bi-modal distribution of the resulting particle set. In this way, bi-modal and even multi-modal probability density functions can be represented by particle filters.

The Condensation algorithm in its simplest form suffers from the likelihood of not exploring crucial parts of the probability distribution and the inability to recover from these types of errors. Jensfelt, Wijk, Austin and Andrsson (2000) propose augmentations of Condensation which overcome these limitations with sampling techniques such as random sampling and planned sampling. In our research the state space is restricted to to the space about individual places, so the space will be covered by an initial random distribution. To ensure good particles are re-sampled, this system automatically re-samples particles which have an observed probability greater than the mean plus the variance. A standard holonomic motion model is used in the prediction

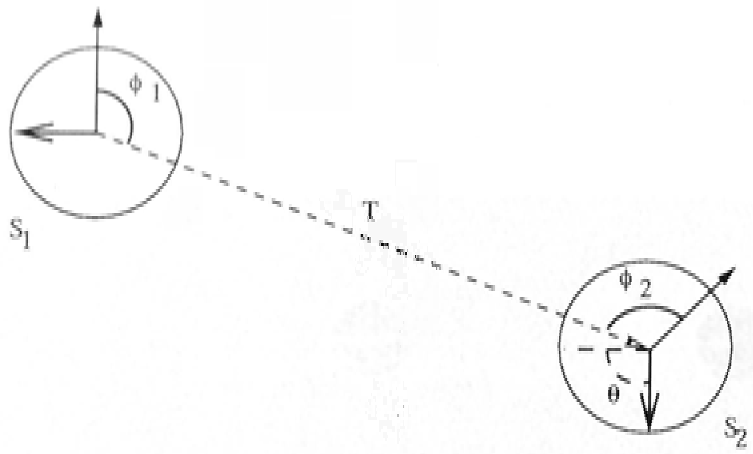


Figure 8.8: Components of a holonomic movement

phase. A sensor model has been derived to suit the landmark based place representation. Below, the motion model and the sensor model are described in detail and experiments in local positioning are reported.

8.1.3 Motion Model

The motion model used is a standard position based holonomic motion model. Using two odometric measurements of robot position, a holonomic move can be defined which incorporates stochastic noise and deterministic drift and can be used to predict the actual movement of the mobile robot. A holonomic move between any two points can be said to made up of the four components shown in Figure 8.8, and listed below:

- ϕ_1 : an initial steering rotation to the heading between the points,
- T : a translation distance across the intervening distance
- ϕ_2 : another steering rotation to the end steering position,
- θ : an orientation rotation to the required orientation position.

Components can be adjusted for noise (sampled from the normal distribution) as follows:

$$\hat{\phi_1} = \phi_1 + normal(\alpha_1\phi_1 + \alpha_2 T) \tag{8.5}$$

$$\hat{T} = T + normal(\alpha_3 T + \alpha_4\phi_1) \tag{8.6}$$

$$\hat{\phi_2} = \phi_2 + normal(\alpha_1\phi_2 + \alpha_2 T) \tag{8.7}$$

$$\hat{\theta} = \theta + normal(\alpha_5\theta + \alpha_2 T) \tag{8.8}$$

where $normal(x)$ returns a sample from the normal distribution between $[-1 : 1]$ with variance x, and α_i represents the levels of noise being introduced into the system. In particular, α_1 noise in steering position while steering, α_2 noise in steering/orientation position while translating, α_3 noise in translation position while translating, α_4 noise in translation position while steering, and finally α_5, noise in orientation position while changing orientation. Given a sample state S_{i-1} and a holonomic move $(\phi_1, T, \phi_2, \theta)$, the motion model predicts an end state S_i by applying the movement $(\hat{\phi_1}, \hat{T}, \hat{\phi_2}, \hat{\theta})$ to S_{i-1}.

8.1.4 Sensor Model

The sensor model is used to calculate the probability of making a particular observation given a sample of the system state as represented by individual particles in the particle filter. In our research the sensor model is derived to suit the landmark based place presentation. The panoramic sensor provides a wealth of information from the environment, while the place representation has an abstracted form. We treat the panoramic vision and landmark matching systems as a landmark sensor, and the sensor model calculates the probability of observations based on the lowest common level of abstraction between the learnt place and the sensor: that of the observed radial angles of landmarks.

The sensor model's input consists of a robot state sample S_i, the radial angles of landmarks from the learnt place, O_R, and radial angles of the observed landmarks from the current image O_C. For a particular observation O_C, there are $o_0..o_n$ angles, where n is the number of landmarks in a landmark set. The probability $P(o_k)$, given state S_i of observing angle o_k, $o_k \in O_C, 0 \leq k \leq n$, is a mixed distribution composed from the following distributions:

- P_{rand}: probability of the sensor returning a noisy random result.

$$P_{rand} = 1/360\,\text{deg} \tag{8.9}$$

This value is basically there to ensure an incorrect landmark observation does not result in a zero probability for the sample state S_i.

- P_{corr}: probability that at state S_i, the sensor correctly observed landmark l_k. If S_i is the correct state, the intersection of the lines formed by the current observation from S_i and the reference observation from S_R should occur at the landmark's position in the environment. We determine the intersection between the reference and current observations as:

$$P_{corr} = rel(o_k)P_{hit} \tag{8.10}$$

where, $rel(o_k)$ is the matching correlation between the reference and current images for landmarkl_k, and P_{hit} is the probability that the reference and current observations intersect at the location of the landmark.

These distributions are combined by a weighted sum based on their relative importance (manually set based on experimental evidence):

$$P(o_k) = P_{rand}w_{rand} + P_{corr}w_{corr} \tag{8.11}$$

and then the product over the number of observations gives the final probability that the current observation was made at the sample state S_i:

$$P(O) = \prod_{k=0}^{n} P(o_k) \tag{8.12}$$

Using this process a particle filter can take inputs from the panoramic sensor and find and re-sample particles representing system states which are consistent with the current observations.

This process depends on defining the probability, P_{hit}, that the current observation intersects with the appropriate landmark. Methods for calculating this probability depend on the complexity of the geometric model which is used to represent the landmark position and the observations. Four methods of calculating this probability have been investigated:

- Assumed Depth Line Intersection
- Estimated Depth Line Intersection
- Ellipsoid and Line Intersection
- Ellipsoid and Arc Intersection

Assumed Depth Line Intersection

In the assumed depth sensor model the reference and current sensor observations, being radial angles of tracked landmarks, are geometrically modelled as lines extending from the center of the robot at the observed angle direction. In addition this method assumes that the system has no knowledge about the depth of the landmark along the ray from the reference position. This model is included to illustrate the importance of having an estimate of the depth of the landmarks in the position estimation task.

The assumed depth line intersection sensor model seeks to find the probability that the reference and current observation rays intersect at the point in the environment where the landmark is located. In the absence of information about the depth of the landmark, this approach assumes that landmarks are at a constant depth and the probability is calculated with this assumption. The probability is determined by the distance of the intersection of the two lines from the assumed landmark depth. If the two lines intersect at the assumed depth along the reference observation ray the probability of making such an observation is set to the maximum, otherwise the probability is defined by a Gaussian distribution centered around the assumed depth. More formally this relationship can be defined as:

$$P_{hit} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{(d_k-\bar{d})^2}{\sigma^2}\right)} \tag{8.13}$$

where d_k is the intersection depth of the current observation α_k with the reference observation, and $\bar{d}$ and σ are the assumed depth and variance of landmarks.

If a landmark had an initial observation angle of γ, a current observation of θ and the

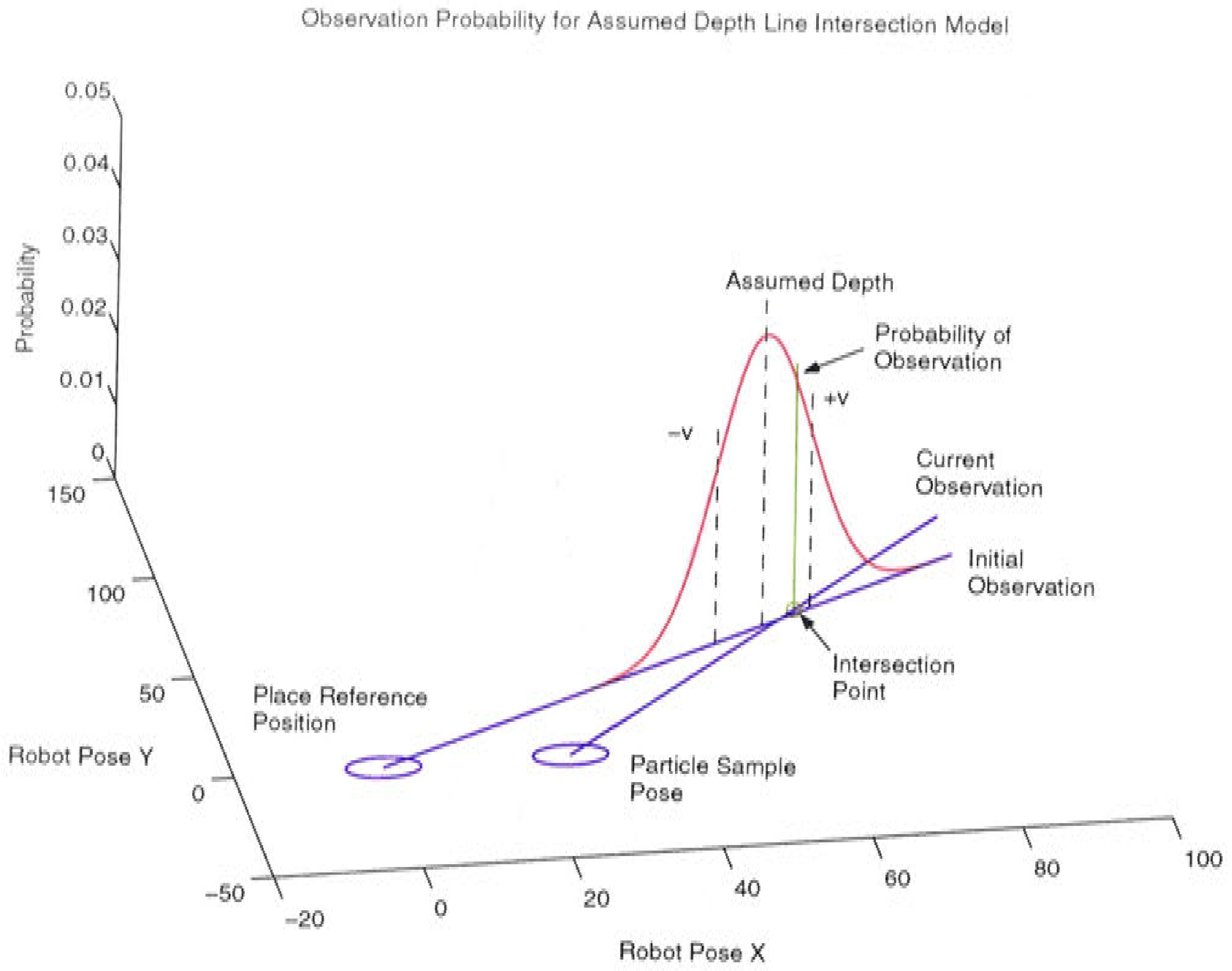


Figure 8.9: The assumed depth sensor model finds the probability that the two lines formed by the current and reference observations intersect at the assumed depth of landmarks from the reference position.

sample particle represented the robot pose $(x, y)_p$, then from standard geometry the intersection $(x, y)_I^T$ of the two lines formed by the reference and current observations can be calculated as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix}_I = \begin{pmatrix} \frac{y_p - x_p tan\theta}{tan\gamma - tan\theta tan\gamma} \\ \frac{tan\gamma(y_p - x_p tan\theta)}{tan\gamma - tan\theta} \end{pmatrix} \tag{8.14}$$

The depth from the reference position at which the current observation intersects with the reference observation is then:

$$d_k = \sqrt{x_I^2 + y_I^2} \tag{8.15}$$

This value can then be substituted into Equation 8.13 to discover the probability of making the current observation from the sample particle's hypothetical pose.

Figure 8.9 shows the assumed depth line intersection model. In this figure the blue

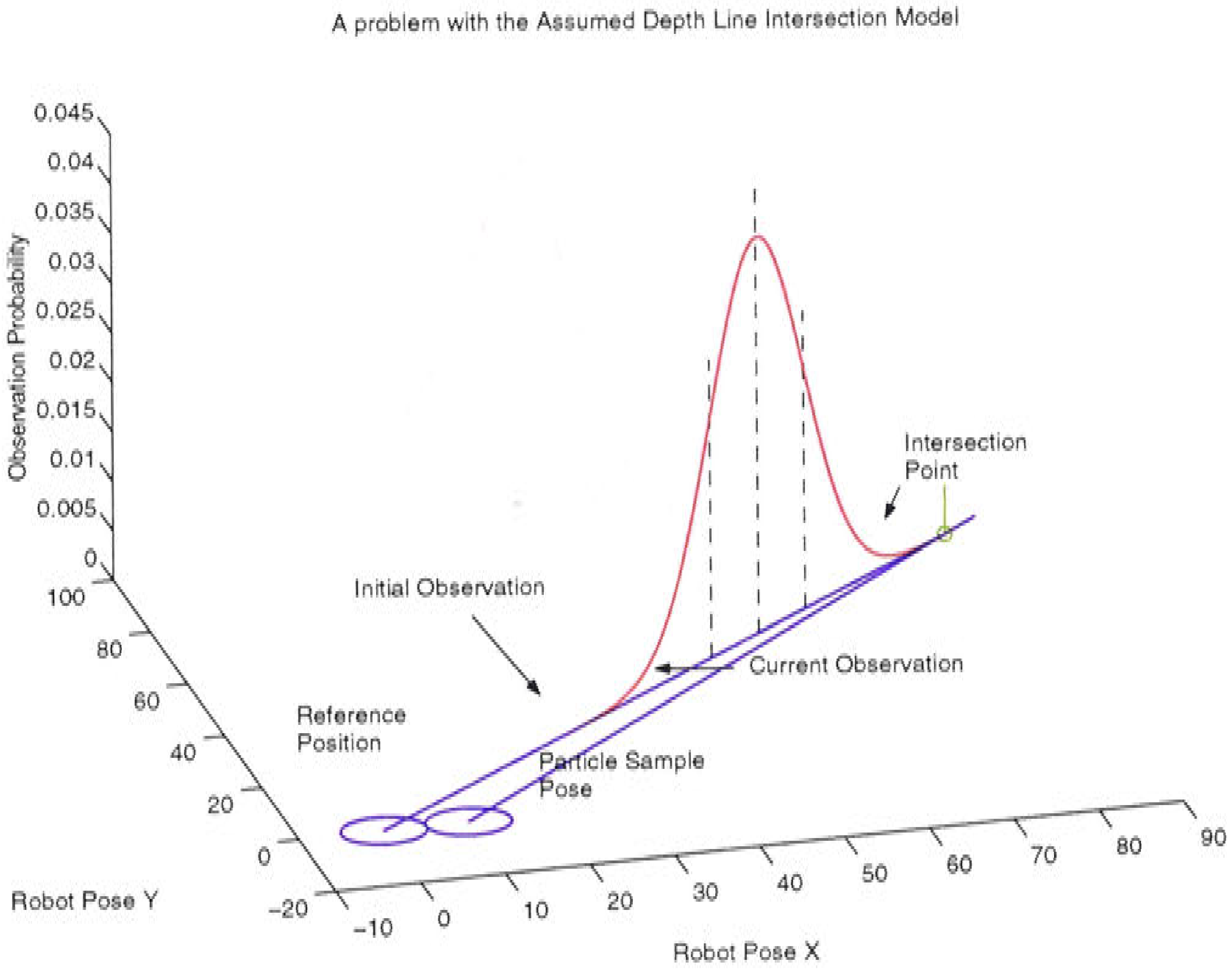


Figure 8.10: The assumed depth line intersection probability calculation is inaccurate when the observation lines approach parallel.

circles represent the place reference position from which the initial observations of landmarks where made, and the hypothetical robot position of the particle being evaluated. The blue lines denote the initial and current observations from the the respective positions. The red curve shows the Gaussian probability distribution centered around the assumed depth while the intersection point of the two observations and the resulting probability value is shown by the green line.

Obviously by using an assumed depth in this model, the calculated probability will not reflect the true relationship between the current observation and the actual landmark position. This highlights the importance of depth estimation in the landmark acquisition process.

Another problem of using a simple geometric model of the observations can be seen in Figure 8.10. A problem occurs when the reference and current observations are parallel. In this situation there will either be no intersection between the two lines or the intersection will be infinite. In fact for particles which have a robot close close to the reference position, current observation which approach parallel with the reference observation will falsely return low probability values as two lines either do not intersect, or intersect at a great distance from the assumed depth. This is particularly a problem when the robot is actually located at the reference position, when all current observation angles to landmarks are equal to the reference angles. It is not problematic when the intersection of only one landmark in the landmark set is not defined as the probability of other landmark observations can overcome the problem.

An ad hoc solution to this problem is to introduce an alternate model when the intersection is not defined. Such a distribution assumes that most situations where there are no intersections in the model occur around the reference position from which the place was learnt. Accordingly it calculates the probability of the observation from the distance of the sample robot pose to the reference position:

$$P_{hit} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x_r^2 + y_r^2}{\sigma^2}\right)} \qquad (8.16)$$

ensuring the sensor model returns a probability measure even when the intersection between the two lines is not defined.

To further explain the various sensor models being discussed, it is advantageous to apply them to an example in the mobile robot domain. Figure 8.11 shows two landmarks being observed from both a reference position $(0,0)$ and the actual current position of the robot $(0,80)$. In this example it assumed that the robot's orientation is known, therefore limiting it to a two dimensional state space.

With prior knowledge about the reference position and the reference observations, knowledge of the the robot's current position can be inferred from the current observation. The effect of the sensor models can then observed by applying them to all possible robot poses throughout the sample environment. It should be noted here that by applying the model over all possible poses, the system is in fact approximating the probability density function of the robot position over the state space of robot poses. In the particle filter approach to mobile robot localisation, each particle represents just one possible pose, so the PDF is only being sampled.

Figure 8.12 shows the results of applying the assumed depth line intersection sensor model with an assumed depth of 40 units to the set of all possible robot poses. In

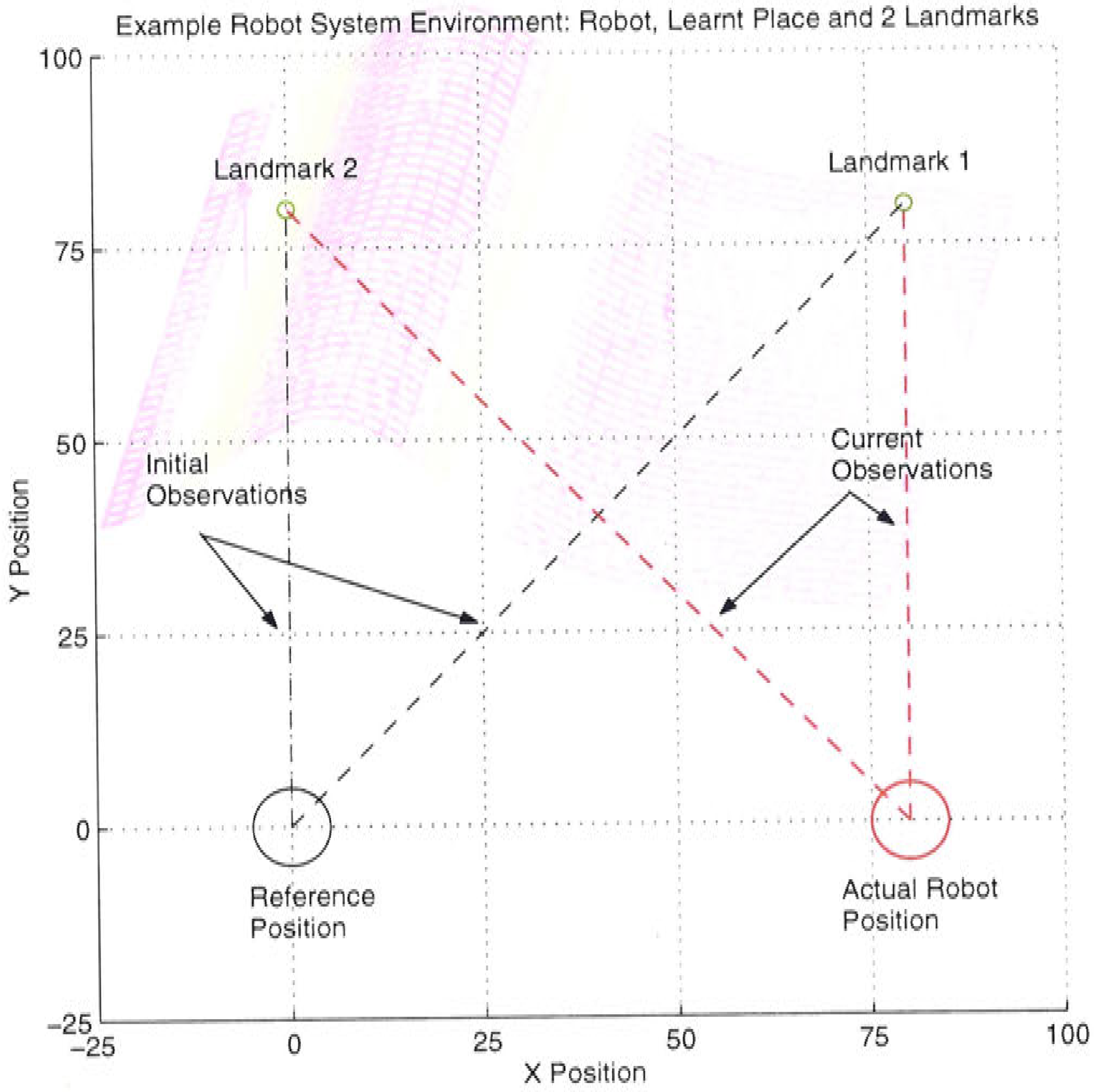


Figure 8.11: An example mobile robot localisation task. The robot has learnt two landmarks from a reference position, and now must perform localisation based on the current observations of the landmarks.

this figure landmark one has been observed at an angle of $\pi/2$ radians. The surface of the graph represents the probability that a robot at a given (x, y) pose in the state space observed landmark 1 at an angle of $\pi/2$ radians. The reference position, actual and assumed landmark location and the actual robot position are shown by the different colour circles. The current observation produces probability high at the assumed landmark depth position. This reflects the fact that at these locations it is more likely to observe the landmark at the assumed position at the angle of $\pi/2$ radians than any other position in the state space.

The peak of this graph lies far from the actual robot position, occurring in the states from which it is more likely to observe the landmark at it's assumed depth rather

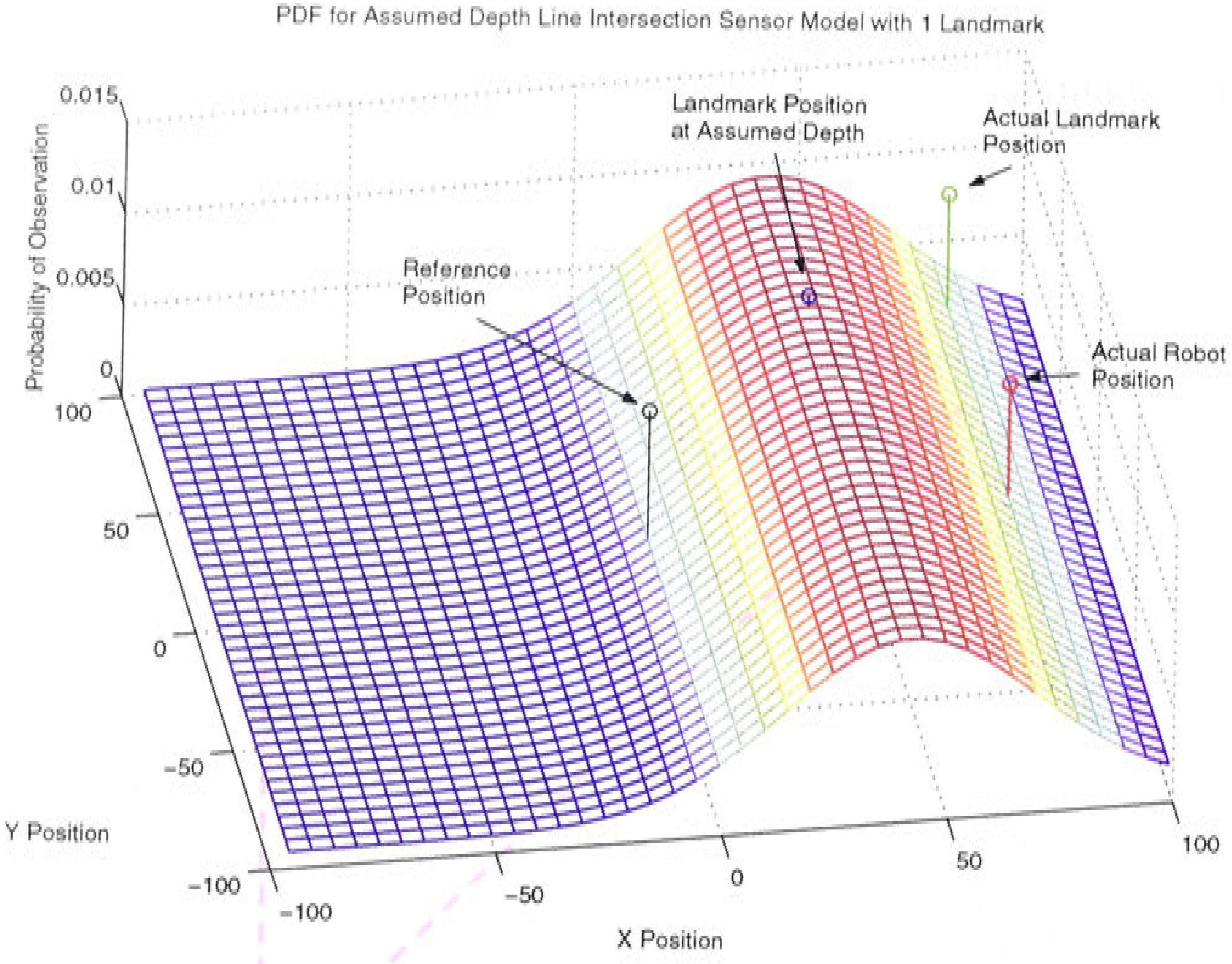


Figure 8.12: The probability distribution produced when applying the assumed depth sensor model over the range of all possible robot positions given the current observation of landmark 1 at 90 deg.

than it's actual depth. This highlights the fact that an estimation of depth is necessary to provide accurate position information to the robot system. The slope of the graph surrounding the peak is also quite shallow reflecting the high variance in the Gaussian distribution. The high variance can assist in alleviating the problem of assumed depth, but subsequently limits the resolution of any positional information the model can produce.

Figure 8.13 shows the results of applying the sensor model to the same example but incorporates the observation of two landmarks. The output of the sensor model for the two landmarks is combined using Equation 8.12. The combination of the probabilities from the two observations reduces the area from where it is most likely to explain the observations. By incorporating more observations into the sensor model, the area of likelihood can be reduced still further. The problem of an assumed landmark depth is still a problem, as the probability peak is still not located near the actual robot position.

Finally, Figure 8.14 shows the PDF produced by the assumed depth line intersection

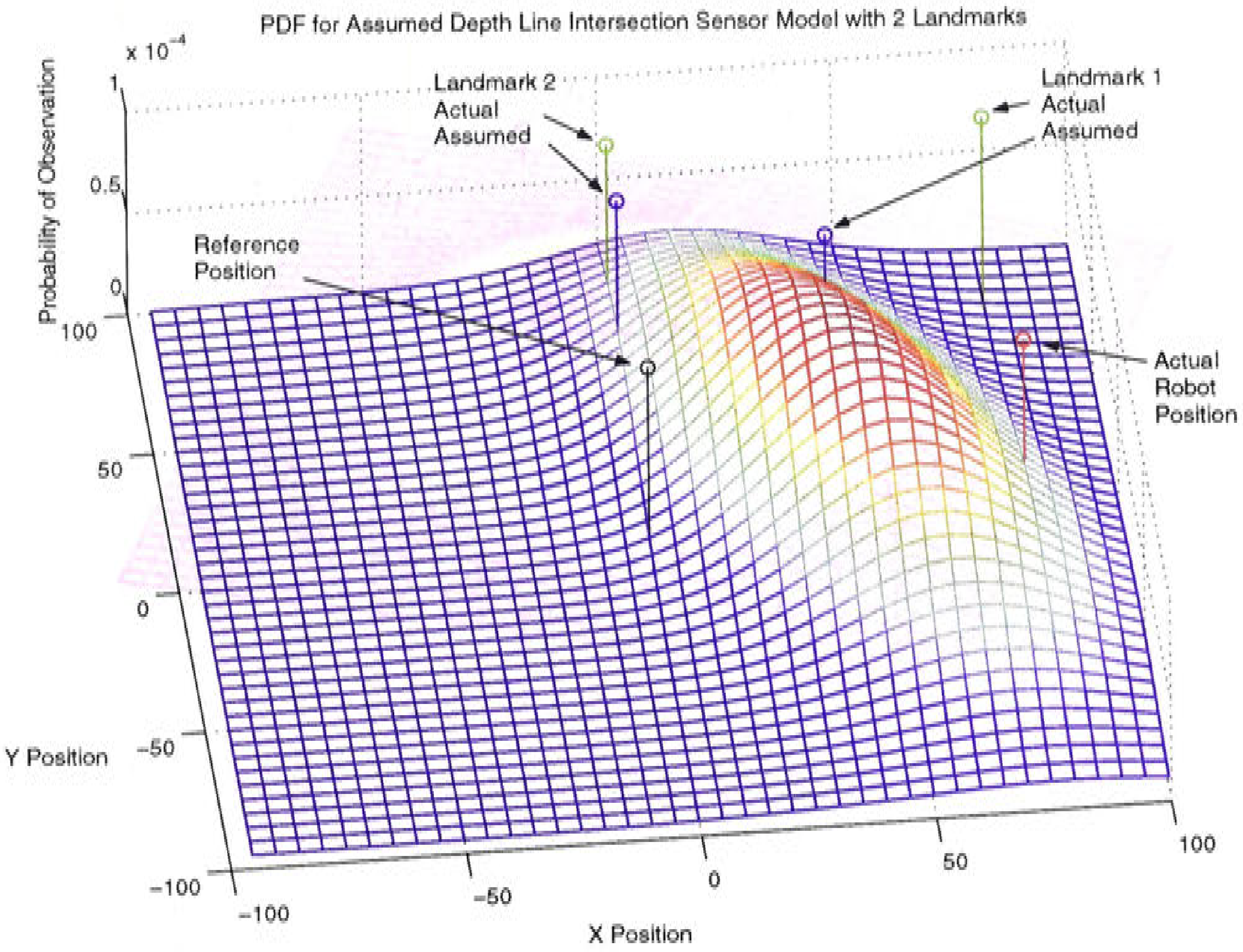


Figure 8.13: The probability distribution produced when applying the assumed depth sensor model over the range of all possible robot positions given the current observations of landmark 1 at 90 deg and landmark 2 at 45 deg.

sensor model with one landmark when the reference and the current observations are parallel. In this case the intersection between the two lines is undefined and the alternative sensor model described by Equation 8.16 is used. If the observation was made from the area surrounding the reference position then this model would correctly identify the most likely robot poses. However in this case the actual robot position is located well outside the identified likely region.

Estimated Depth Line Intersection

The estimated depth line intersection sensor model is similar to the assumed depth model except it takes advantage of the information about the depth of landmarks produced by the Turn Back and Look movement. The probability of making an observa-

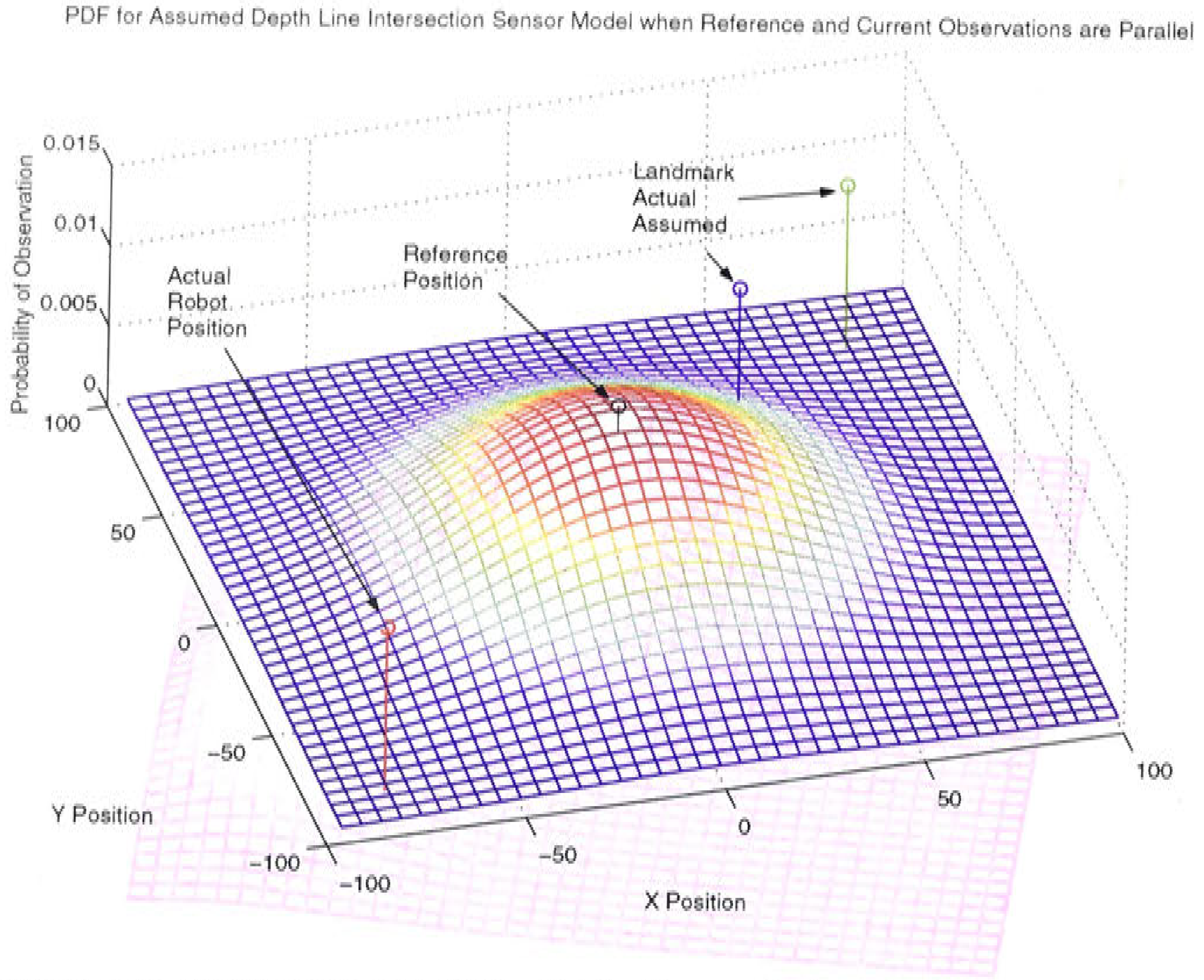


Figure 8.14: The alternate probability distribution used when the lines representing the current and reference observations are parallel.

tion from a sample robot pose is given by:

$$P_{hit} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{(d_k - \bar{d})^2}{\sigma^2}\right)} \tag{8.17}$$

where d_k is the intersection depth of the current observation o_k with the reference observation, and $\bar{d}$ and σ are the estimated depth and depth variance of the landmark.

The estimated depth model still suffers from the problem which occurs when the lines formed by the reference and current observations approach parallel. The alternative model is again applied when this situation occurs.

Figure 8.15 shows the results of applying the estimated depth sensor model to the sample robot environment illustrated in Figure 8.11 and observing one landmark. Again a similar result from that of the assumed depth model can be seen, except that the peak region now covers the actual robot position. This is because the estimated depth of the landmark is much closer than the assumed depth. The slope of the graph in the peak

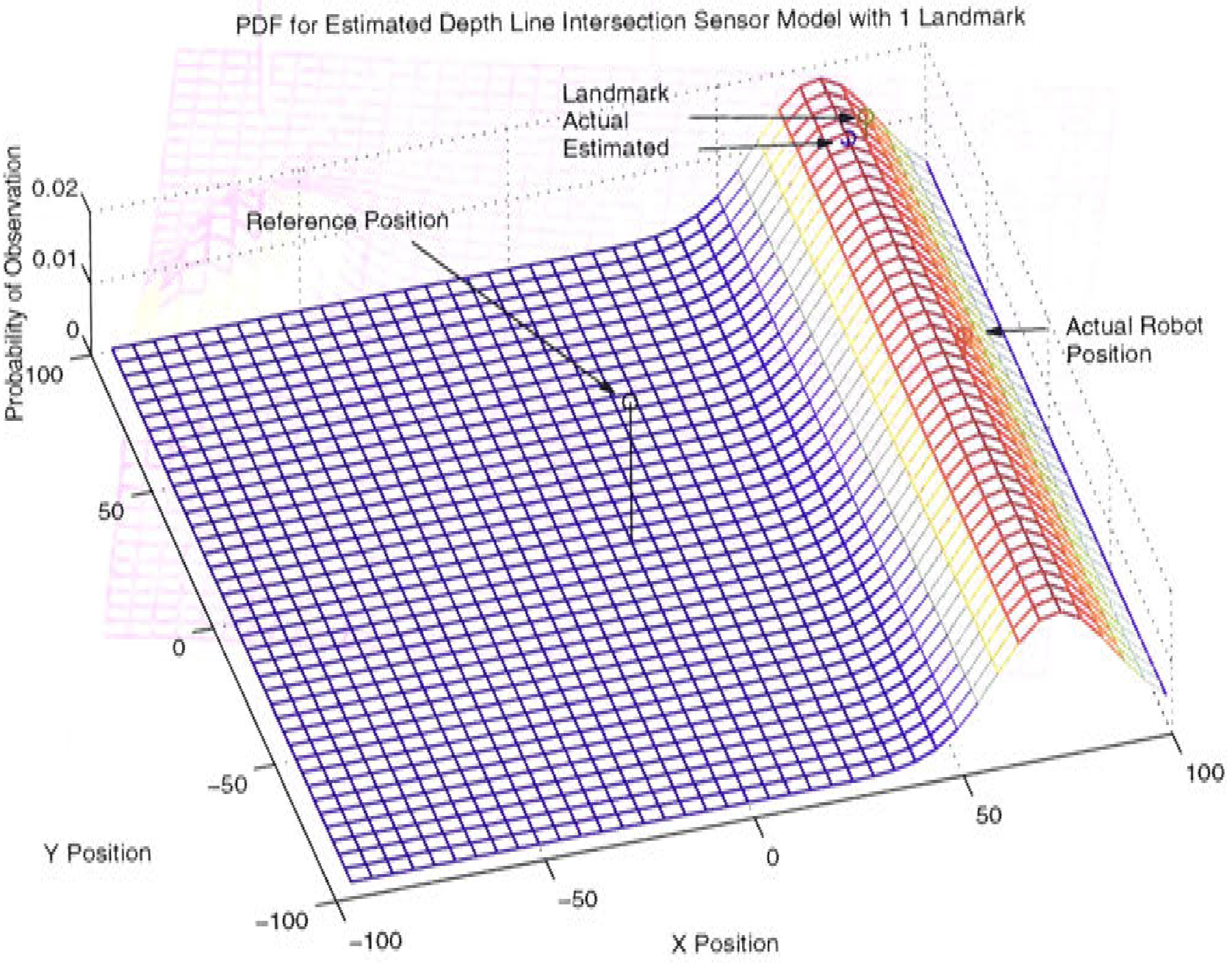


Figure 8.15: The probability distribution produced when applying the estimated depth sensor model over the range of all possible robot positions given the current observation of landmark 1 at 90 deg.

regions is also greater due to the decreased variance in the landmark depth estimate when compared to the assumed depth.

The results incorporating two landmark observations using the estimated depth sensor model are shown in Figure 8.16. The combination of the two landmarks cause a single peak near the actual robot position. Additional landmark observations would strengthen this peak and further eliminate regions in the state space from where it would be unlikely to make such observations.

Ellipsoid and Line Intersection

A more sophisticated geometric model for the observation intersection is to model the reference observation as an ellipsoid defined by the landmark estimate from the Turn Back and Look movement. This means that the model incorporates the uncertainty in the initial angle observation as well as in the depth estimate. The perimeter of the

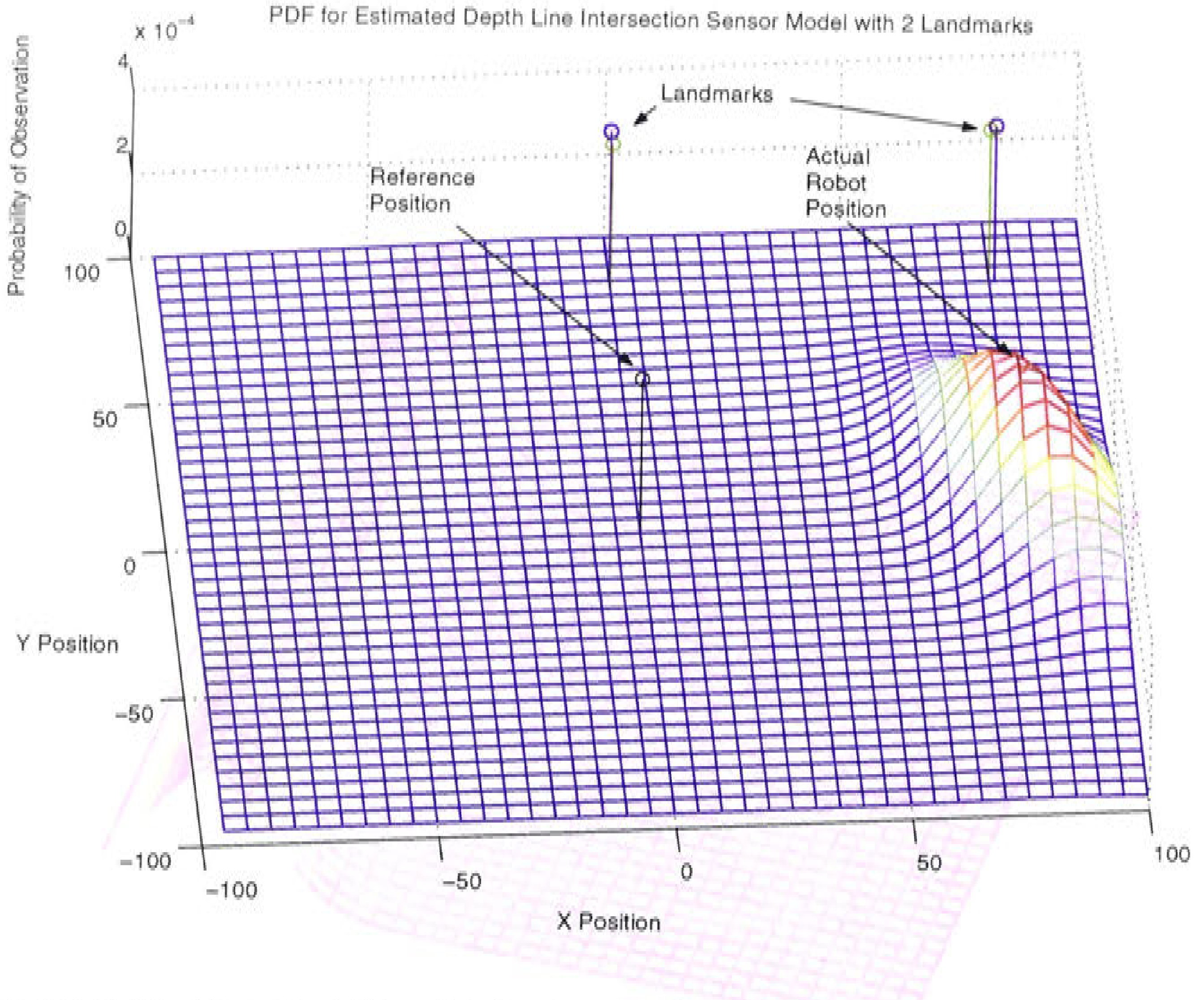


Figure 8.16: The probability distribution produced when applying the estimated depth sensor model over the range of all possible robot positions given the current observations of landmark 1 at 90 deg and landmark 2 at 45 deg.

ellipse defined by the landmark position estimate represents the variance along the depth and angle axes of the estimate. In order to produce a probability measure of making a particular observation from a given sample state, the ellipse can be overlayed with a two dimensional Gaussian distribution centered on the landmark position and scaled to the respective variance measures of the depth and angle axes. The resulting three dimensional surface is not a true ellipsoid, but is referred to as an ellipsoid to distinguish it from the two dimensional ellipse. The surface of the ellipsoid can then be used as a probability measure of the landmark being in the area surrounding the estimated landmark location in two dimensional space. If an observation from a particle hypothesising a particular robot pose intersects with the ellipsoid, then the probability of making the observation can be given by the highest point of the ellipsoid surface through the cross section defined by the observation line.

The situation is seen more clearly in Figure 8.17. In this figure the reference and current robot positions are shown by the blue circles. The estimated position of the landmark

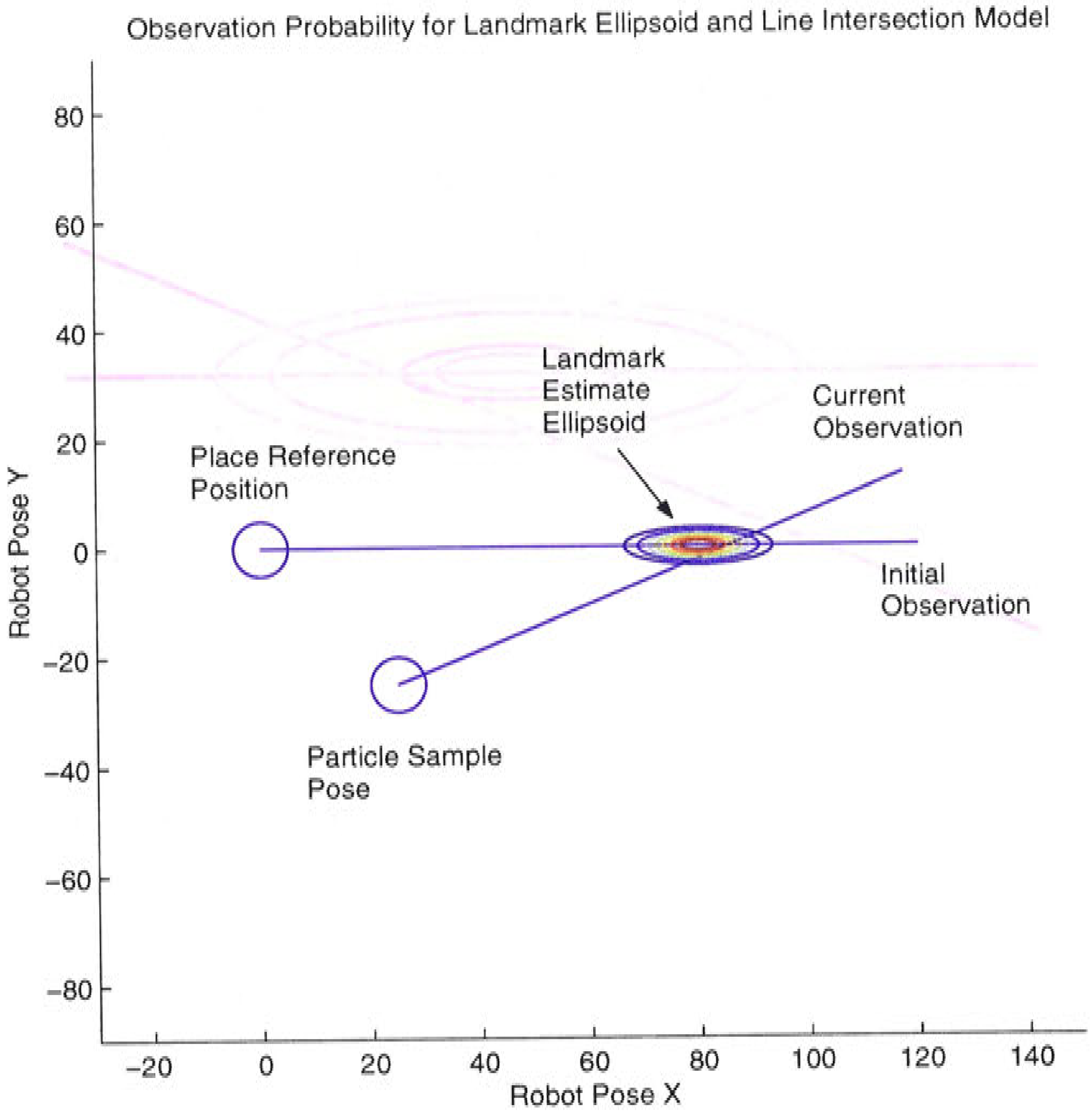


Figure 8.17: The ellipsoid line intersection sensor model takes advantage of the landmark position uncertainty ellipse estimated during the TBL movement to calculate the probability of making an observation from a particular position in the robot state space.

after the TBL movement and its associated ellipsoid is shown by the contour plot along the line representing the reference observation. The line representing the current intersects the ellipsoid and the line segment between the two intersection points defines a cross section of the three dimensional entity.

Figure 8.18 shows a close up of the ellipsoid and the various contours of the ellipsoid can be seen more clearly. It is obvious that the maximum probability of making the observation does not lie on the major or minor axes of the underlying ellipse. This highlights the need for the three dimensional representation of the probability distribution. Also, modelling the landmark probability distribution in this fashion elimi-

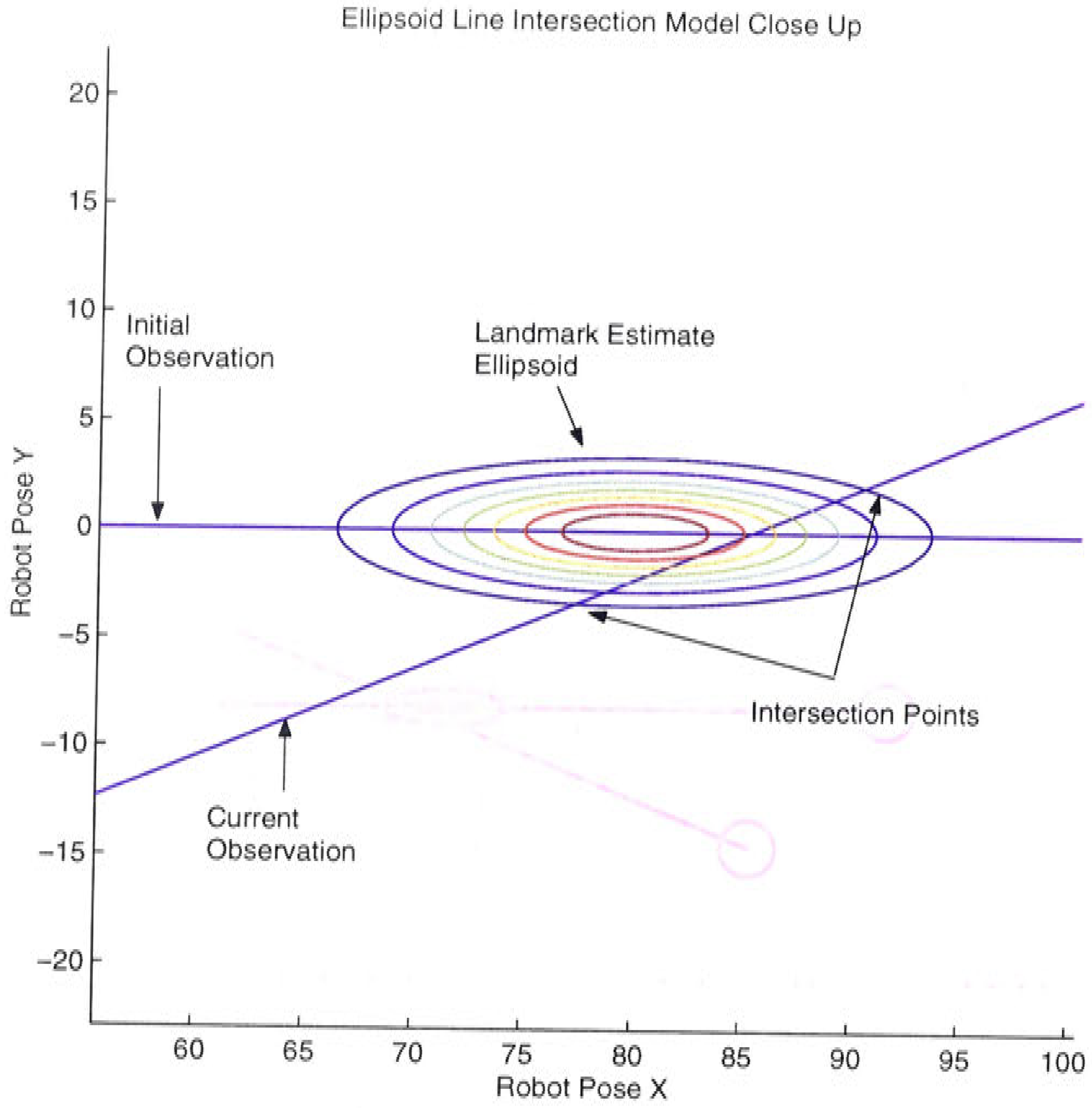


Figure 8.18: A close up of the ellipsoid line intersection showing intersection points and the probability distribution within the ellipse.

nates the problem of parallel reference and current observations that are evident in the two line intersection models.

Figure 8.19 displays the cross section of the ellipsoid surface defined by the intersecting line segment. The maximum probability value in this cross section is used as the probability measure for making the current observation from the sample robot pose.

This sensor model requires the calculation of the intersection points of a ray, defined by the sample robot pose and the angle of the current observation, and an ellipsoid defined by the estimated landmark depth and the angle from the reference position and associated variances. The maximum probability of the intersecting line segment must be found in order to produce a probability measure for the observation. This can

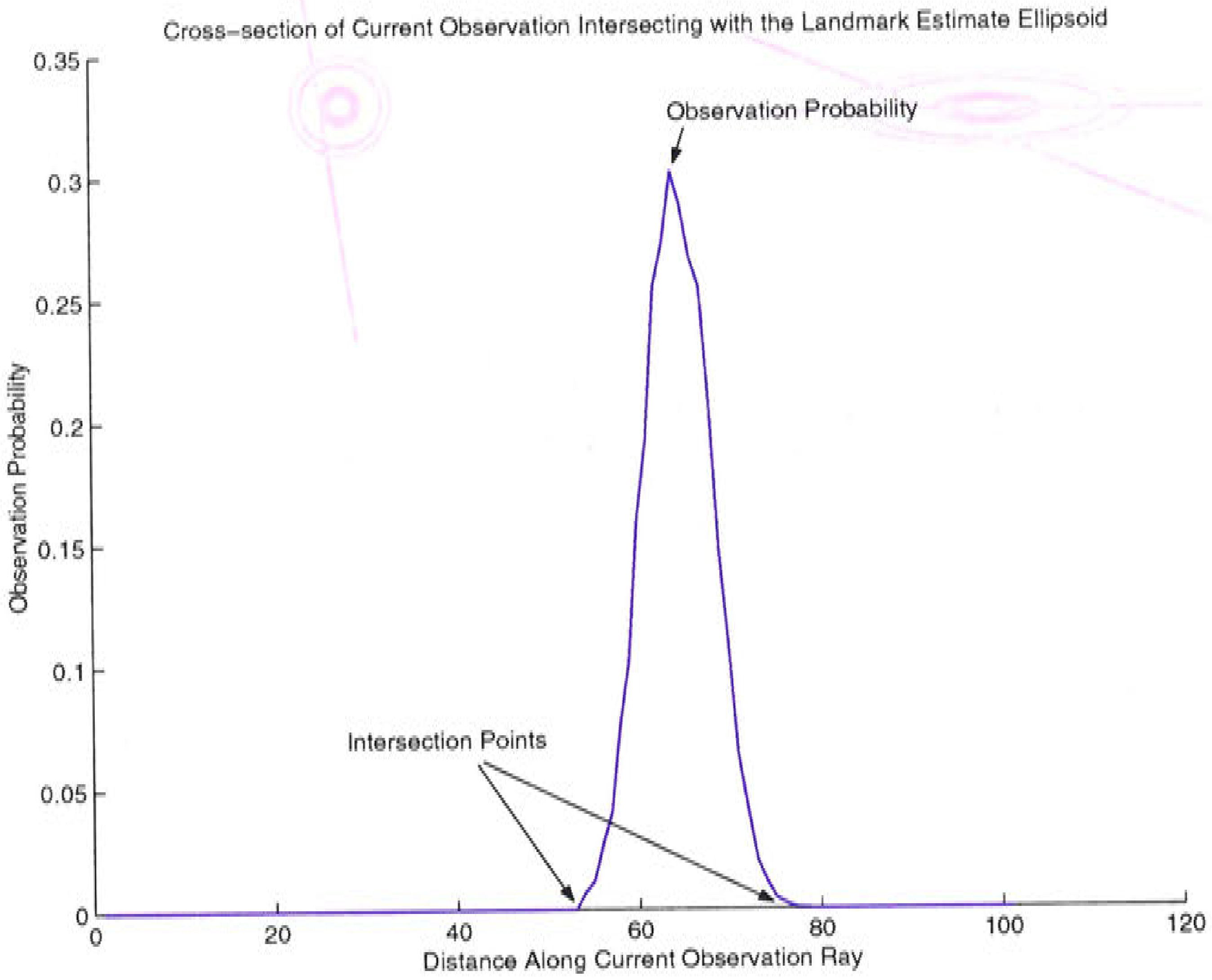


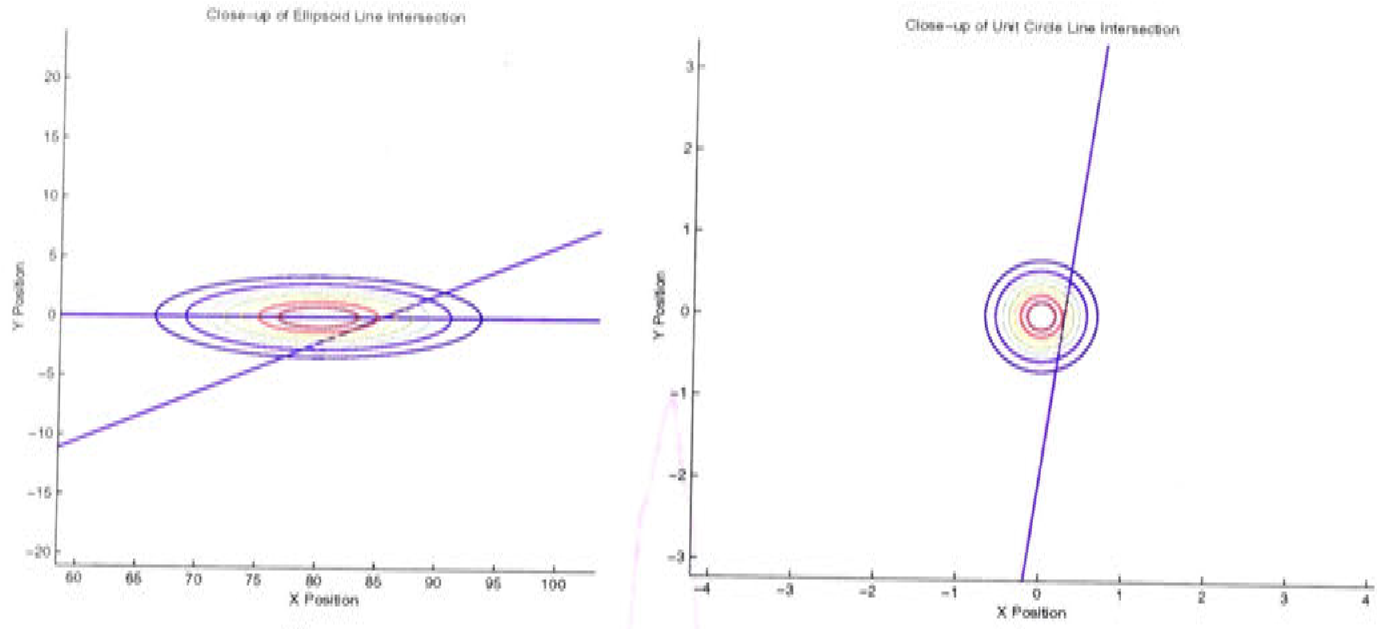
Figure 8.19: The cross section of the ellipsoid probability distribution along the line segment formed by intersection of the current observation and the landmark uncertainty ellipse.

be achieved in three steps using known geometric techniques:

1. Transform the ellipse and ray to a coordinate system where the ellipse is a unit circle.
2. Find the intersection of the transformed ray and the unit circle.
3. Overlay a two dimensional Gaussian distribution over the unit circle and calculate the maximum probability over the intersecting line segment.

By transforming the ellipse into a unit circle the intersection and probability calculations are greatly simplified. The specifics of the calculations involved in these three steps are presented below.

Let there be an ellipse e defined by the parameters (x, y, a, b, γ), where the point (x, y) denotes the center of the ellipse; the values a and b the magnitude of the major and minor axes; and γ represents the orientation of the major axis. Similarly, let there be a



a)

b)

Figure 8.20: The transformation of a) the line ellipse intersection problem into that of a b) line unit circle intersection.

ray r defined by (x, y, θ) where the point (x, y) is the origin of the ray which points in direction θ.

The ellipse can be transformed into a unit circle by the following three steps:

1. *Translate* back to the origin by $(-x_e, -y_e)$.
2. *Rotate* the ellipse by $-\gamma$ so the major axes are aligned to the X axis of the coordinate system.
3. *Re-scale* the major and minor axes by a and b respectively.

This process leaves the ellipse with the parameters $(x = 0, y = 0, a = 1, b = 1, \gamma = 0)$ which describes the unit circle. The parameters defining the ray can be similarly transformed by first performing the translation and rotation transformations:

$$\begin{pmatrix} x'_r \\ y'_r \end{pmatrix} = \begin{pmatrix} cos(-\gamma) & -sin(-\gamma) \\ sin(-\gamma) & cos(-\gamma) \end{pmatrix} \left(\begin{pmatrix} x_r \\ y_r \end{pmatrix} - \begin{pmatrix} x_e \\ y_e \end{pmatrix} \right) \tag{8.18}$$

and:

$$\theta' = \theta - \gamma \tag{8.19}$$

Since scaling along both axes by different factors will affect the orientation of the new ray, an intermediatory step of defining another point at some distance d along the ray must be performed:

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} dcos\theta' \\ dsin\theta' \end{pmatrix} \tag{8.20}$$

Using the extra point step three of the transformation is performed as follows:

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} 1/a & 0 \\ 0 & 1/b \end{pmatrix} \begin{pmatrix} x'_r \\ y'_r \end{pmatrix} \tag{8.21}$$

$$\begin{pmatrix} x'_d \\ y'_d \end{pmatrix} = \begin{pmatrix} 1/a & 0 \\ 0 & 1/b \end{pmatrix} \begin{pmatrix} x_d \\ y_d \end{pmatrix} \tag{8.22}$$

and finally:

$$\theta = tan^{-1} \frac{y'_d - y''_r}{x'_d - x''_r} \tag{8.23}$$

Now the ray has been transformed to a similar coordinate system as the unit circle a ray/circle intersection can be performed to determine whether the observation ray intersects with the landmark estimation ellipsoid. If we let

$$\delta x = cos\theta, \delta y = sin\theta \tag{8.24}$$

and

$$A = (\delta x)^2 + (\delta y)^2 \tag{8.25}$$

$$B = 2(y_r \delta y + x_r \delta x) \tag{8.26}$$

$$C = (x_r)^2 + (y_r)^2 - 1 \tag{8.27}$$

then the roots of the quadratic give the points where the ray intersects with the unit circle by:

$$X^{+/-} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{8.28}$$

$$I_1 = \begin{pmatrix} x \\ y \end{pmatrix}_1 = \begin{pmatrix} x_r + X^{+/ve}\delta x_r \\ y_r + X^{+/ve}\delta y_r \end{pmatrix} \tag{8.29}$$

$$I_2 = \begin{pmatrix} x \\ y \end{pmatrix}_2 = \begin{pmatrix} x_r + X^{-/ve}\delta x_r \\ y_r + X^{-/ve}\delta y_r \end{pmatrix} \tag{8.30}$$

where $X^{+/ve}$ and $X^{-,ve}$ are the positive and negatives roots of the quadratic equation; and I_1 and I_2 are the intersection points of the ray and the unit circle. Of course these are not always defined, if the expression $B^2 - 4AC$ in the quadratic is less than zero there are no intersection points and if it equals zero there is only one intersection point.

If there are two intersection points defined, then the probability of making the original observation can be measured by overlaying a Gaussian distribution over the unit circle and finding the maximum level of the distribution along the line segment defined by the two intersection points. This is accomplished by populating a two dimensional array with a discretised Gaussian distribution and finding the maximum probability value in the cells of the array which lie along the line segment. Geometrically this can be accomplished by finding the the maximum value of the Gaussian distribution at intervals along the line segment. At any position along the line the value of the Gaussian distribution is defined as:

$$v(t) = \frac{1}{2pi\sigma^2} e^{-\frac{1}{2}\left(\frac{(x_1 + t\delta x)^2 + (y_1 + v_t\delta y)^2}{\sigma^2}\right)} \tag{8.31}$$

where $v(t)$ is the value of the Gaussian at the t^{th} interval along the line segment. The value of the Gaussian depends on the distance of the point at the interval from the center of the unit circle. If there are two intersection points, the maximum of this value over the interval gives the probability of the observation:

$$P_{hit} = max(v(0), \ldots, v(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2})); \tag{8.32}$$

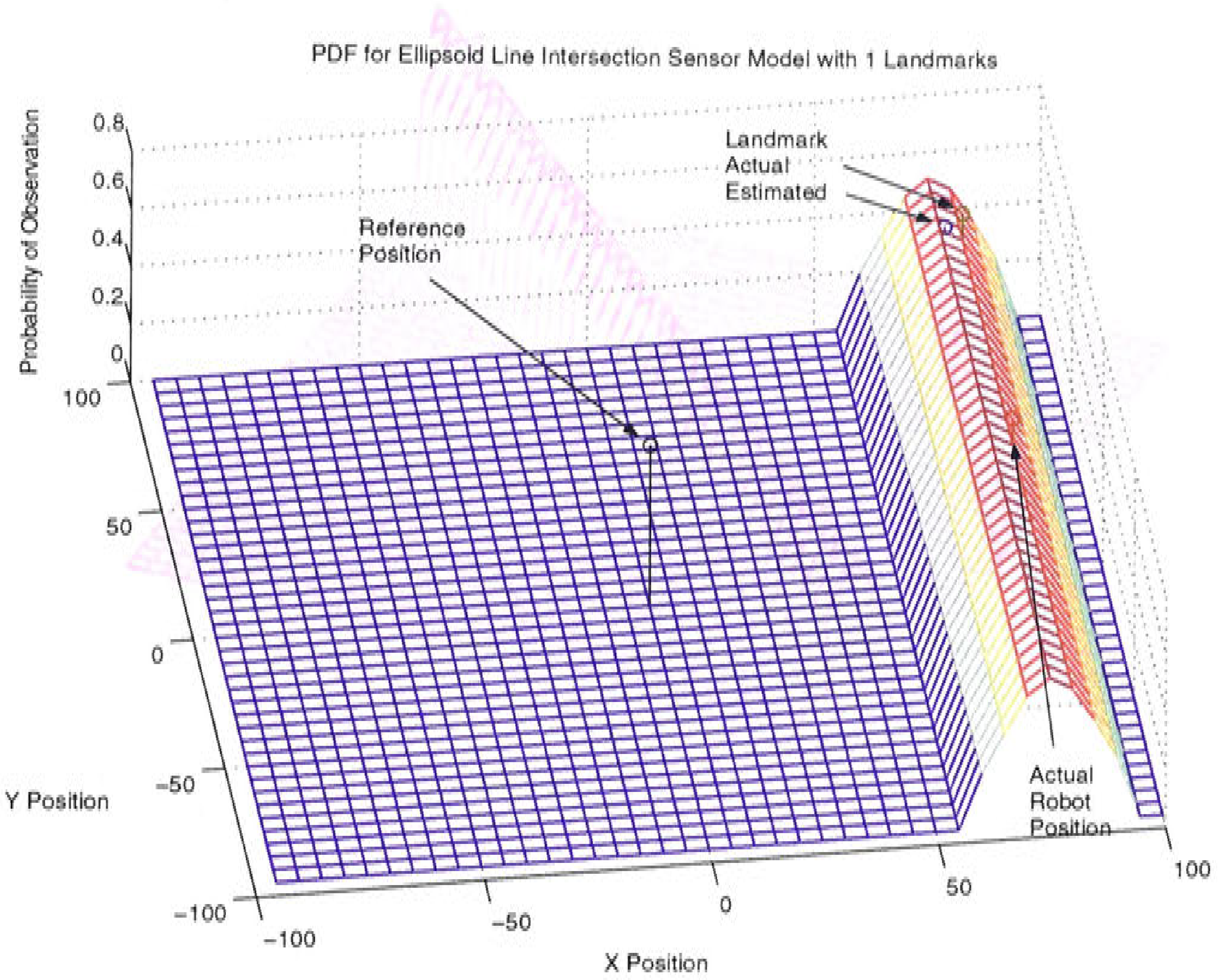


Figure 8.21: The probability distribution which results from applying the ellipsoid line intersection sensor model over the range of all possible robot positions given the current observation of landmark 1 at 90 deg.

In the case where there are either zero or 1 intersection points:

$$P_{hit} = 0 \tag{8.33}$$

This probability value can then be used as an estimate of the probability that the given ray representing an observation of a particular landmark was made from a sample robot pose, irrespective of where the sample pose lies in the robot position state space.

The results of applying the ellipsoid line intersection sensor model over all position states in the robot position state space are shown in Figure 8.21. The surface of the graph shows the probability of observing landmark 1 at an angle of $pi/2$ radians from an (x, y) robot pose. The peak regions of this plot are similar to the results produced by the estimated depth line model, which is to be expected due to the large difference between the reference and current observation angles.

The benefits of using the ellipsoid line intersection model can be appreciated when the

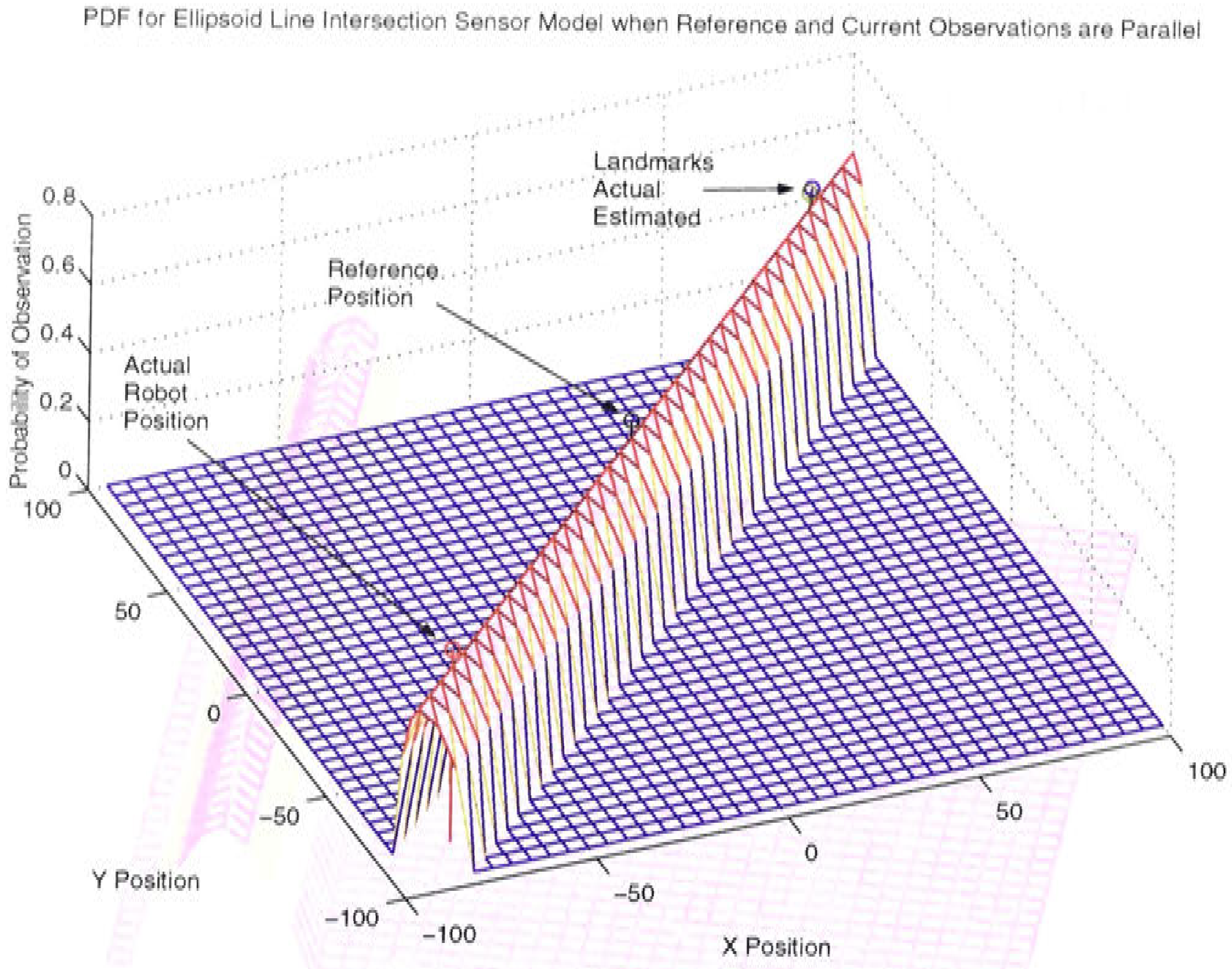


Figure 8.22: The probability distribution produced as a result of applying the ellipsoid line intersection sensor model over the range of all possible robot positions given an observation of landmark 1 which is parallel with the reference observation.

lines defined by the reference and observation angles approach parallel. Figure 8.22 illustrates such a case. The ellipsoid line intersection model returns the correct probability distribution even when the two observation lines are parallel, thus marking an improvement on the sometimes incorrect alternate model necessary when applying the line intersection models described above.

Ellipsoid and Arc Intersection

The ellipsoid line intersection model incorporates the uncertainty of the reference observation angle into the observation probability calculations but fails to take the current observation uncertainty into account. The reliability of the current landmark observation angle depends on the resolution of the panoramic sensor and on the landmark template matching correlation value, as described in Chapter 4. A sensor model which aims to accurately represent the observation probability distribution should in-

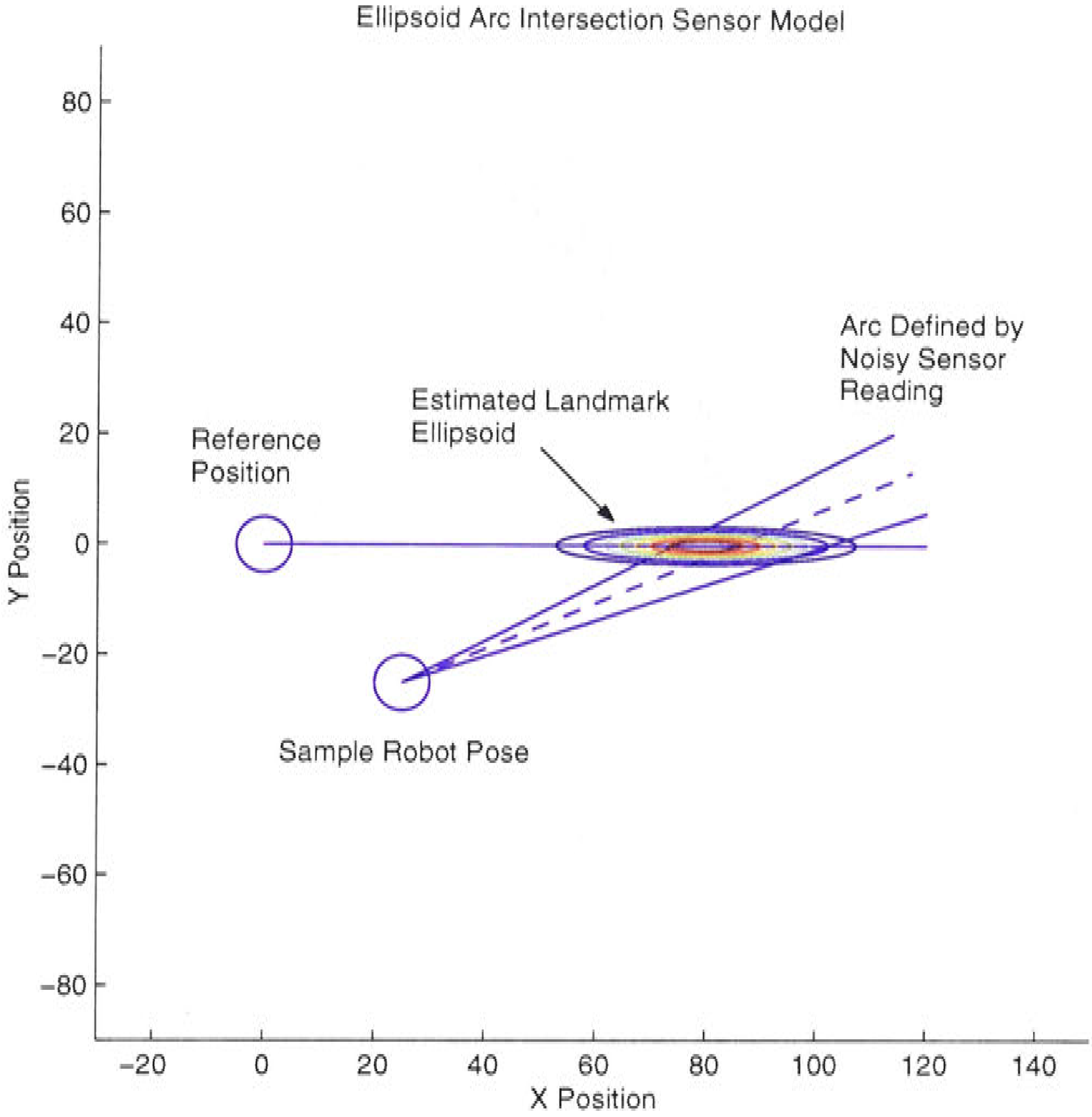


Figure 8.23: The proposed ellipsoid arc intersection sensor model to measure the probability of making a particle observation from a given position in the robot position state space. The ellipsoid and the arc capture the uncertainty in the landmark location and the sensor observation respectively.

corporate all forms of uncertainty in the model.

Figure 8.23 illustrates the situation when incorporating the current observation uncertainty into the sensor model. The noise in the current observation can be represented as an arc centered on the observation angle and displayed as a sector radiating out from the hypothesised robot position. It is unclear as to how best to represent the current observations probability distribution within the sector, as the uncertainty introduced by the resolution of the camera would suggest a uniform distribution between the limits of accuracy while the matching uncertainty could be modelled as a varying Gaussian distribution.

The detection of intersecting regions and the combination of the ellipsoid and sector probability distributions is also computationally expensive and difficult. The calculations need to be performed for each particle in the particle filter at each step of the sense-localise-move cycle, and thus speed of computation is important. It is questionable how much additional accuracy such a complex model will introduce compared to the ellipsoid line intersection model, which depends on the uncertainty in the landmark estimate to model the entire noise in the system.

For these reasons this system employs the ellipsoid line intersection sensor model in the particle filter to solve the local positioning problem within places in the topological map. The uncertainty introduced by the observations is incorporated after the intersection calculation when the probability of making an observation is multiplied by the landmark template matching correlation measure as described by Equation 8.10.

8.1.5 Local Positioning Experiments

Using the holonomic motion model and the various sensor models as defined above, the particle filter can now be applied to the mobile robot local positioning problem. The experimental setup is the same as for the heuristic local positioning method. A place was learnt in the center of a large room (Figure 8.1). From this reference position the robot then traversed a path identical to the Turn Back and Look path used in place acquisition. During this move the images were captured and logged along with odometric readings. Using this data and the particle filter, the robot system attempts to perform local position estimation continually along the path and the results are compared to the odometric ground truth. By performing this experiment, the particle filter approach can be compared to the heuristic method and other known grid based methods. In addition the various sensor models can be evaluated for their contribution to local position estimation in the real world.

Implementation Issues

Implementing the particle filter system described above to operate on a real world mobile robot introduces some uncertainty and timing issues which are not present in the purely geometric models described above. In the real world system sensor data

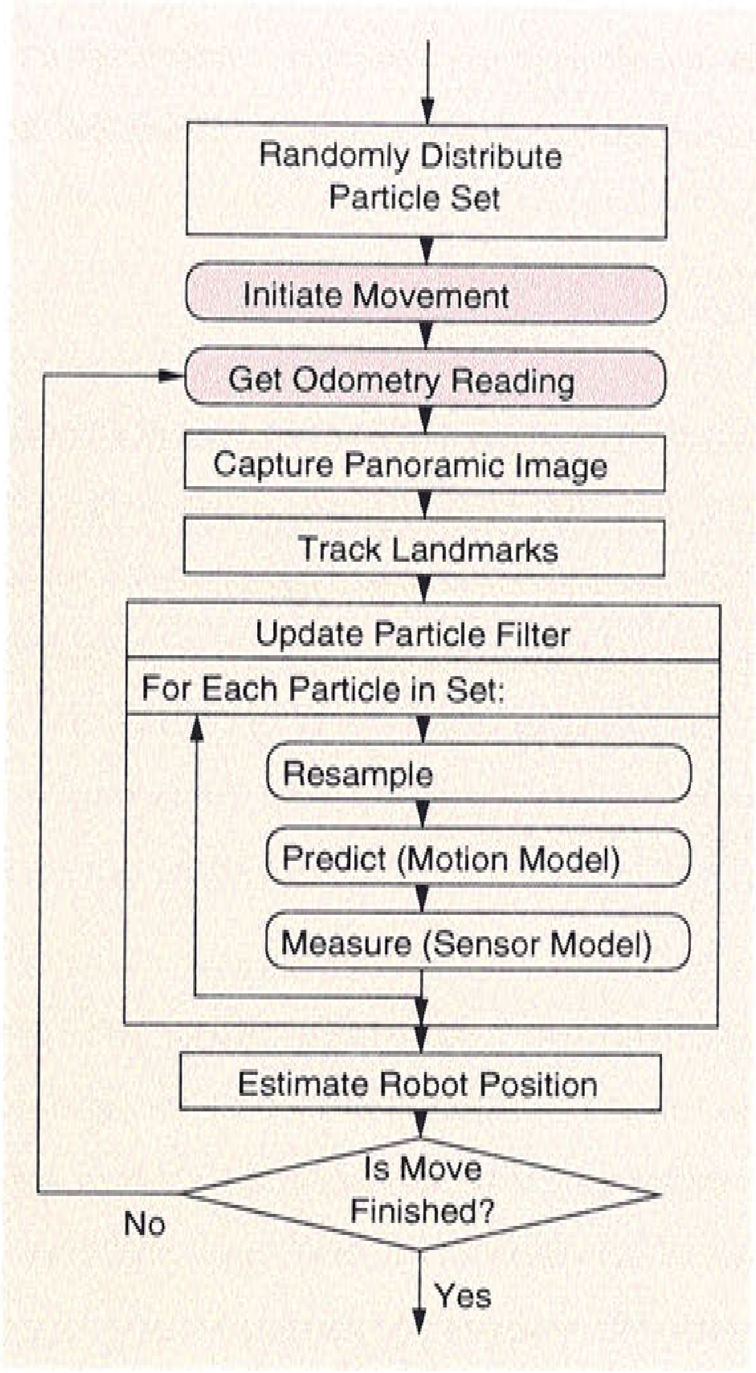


Figure 8.24: The flow of control when performing local positioning on the mobile robot system. The grey shaded functions involve communication across the local area network.

from the panoramic camera and robot odometry are sampled at different instances in time and trying to associate one with the other can be problematic.

Figure 8.24 shows the flow of control when the robot system is performing local position estimation using the particle filter approach. It can be seen that the processes by which the panoramic and the odometric sensor data are sampled are in serial, thereby introducing a time delay between the two sensor modalities. In addition the odometric data must be requested from the robot controller which is operating on a separate processor. This communication over a local area network introduces additional timing delays which further complicates any attempt to temporally synchronise the sensor data.

Rather than enforce hard real time constraints on the sensor sampling process, we assume that the time delay between sampling the two sensor modalities is itself a form of uncertainty in the local position estimation system and can be overcome by the high levels of uncertainty already present in the system or by adding additional uncertainty to the motion and sensor models to factor in this extra source of noise.

Initialisation of Orientation

All the geometric models used to reward robot poses hypothesised by individual particles are sensitive to the orientation of the robot. Obviously a small change in the orientation of the robot will result in a dramatic change in the sensor model response. This means that the particle set distribution must explore potential orientations in the state space thoroughly to produce accurate tracking results. In order to reduce the need for an excessive number of particles to facilitate the orientation search, an initialisation phase can be used to approximate the robot's orientation before local position estimation begins.

Landmarks which have tracked well in the current image provide information about the robots orientation. In general, within the area surrounding a place's reference position the observation angle to a particular landmark does not change dramatically due to translation from the reference position. Any significant changes, therefore are a result of rotation, and can therefore be used to initialise the local positioning system's orientation estimate. The current system performs this initialisation as follows:

Let A be the set of landmarks from a landmark set such that each landmark in the set has a template matching correlation measure with the current image of greater than a threshold value. This ensures only accurately tracked landmarks are selected. In our research this value was set at 0.85. Then the initial orientation θ_0 is set to the average difference between the current observed radial angle of landmarks in set A with the reference angles of landmarks in set A:

$$\theta_0 = \frac{\left(\sum_1^n (\theta_C^i - \theta_R^i)\right)}{n} \tag{8.34}$$

where, n is the number of elements in set A, and θ_C^i and θ_R^i refer to the current and reference angles to landmark i respectively.

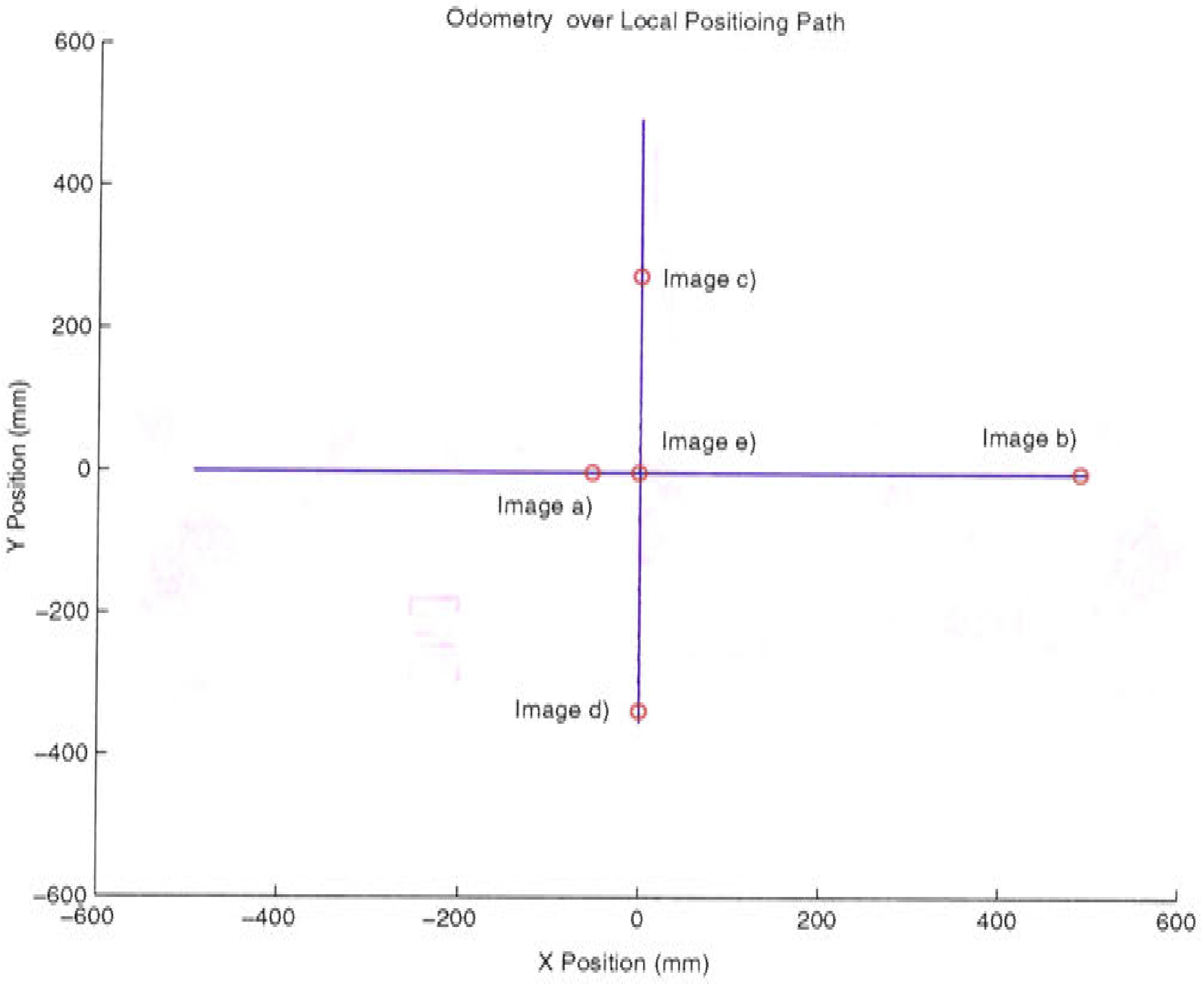
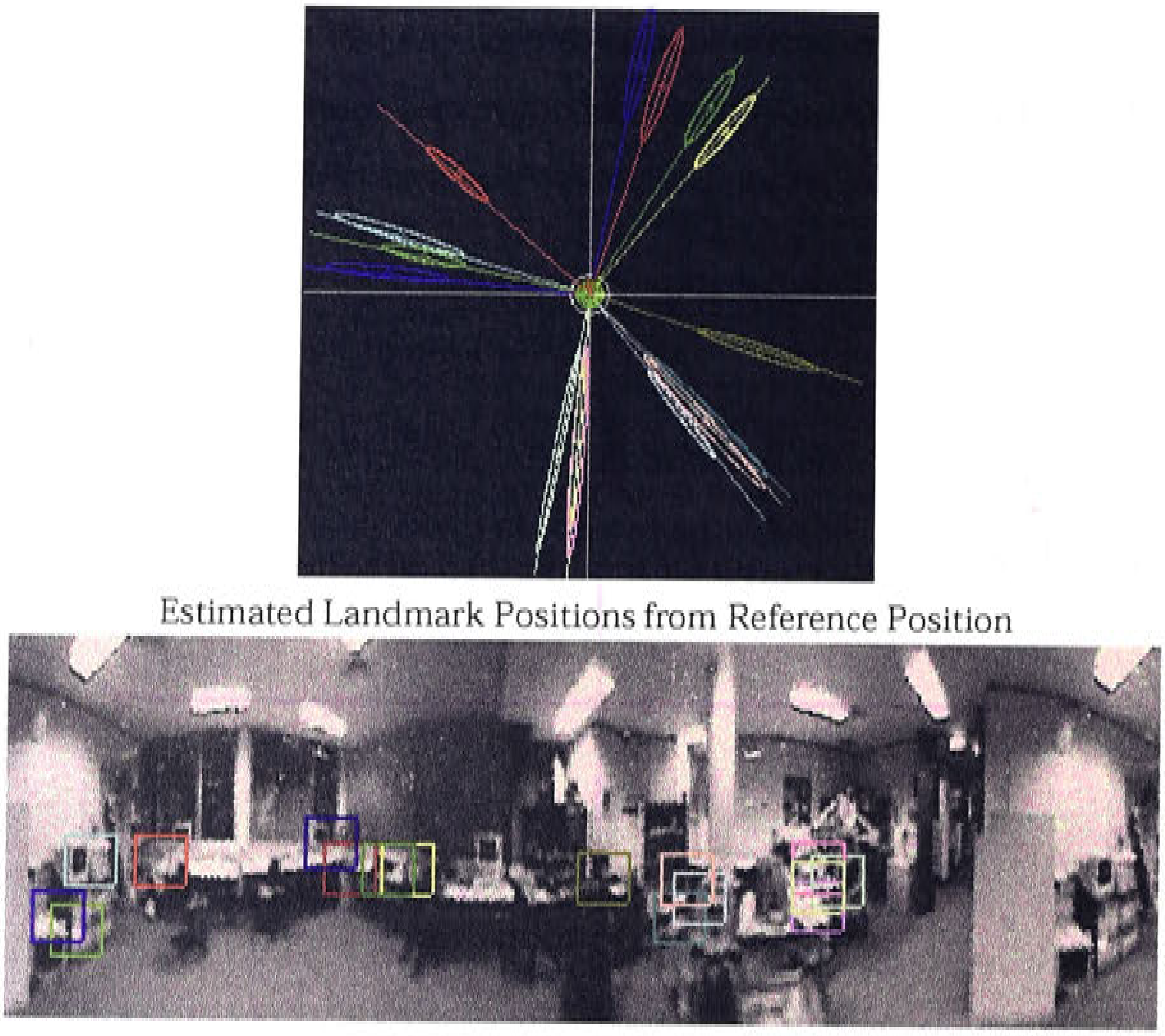


Figure 8.25: Locations of images along the TBL odometric path

The particle set used to explore the state space of robot positions can then be initialised with a Gaussian distribution around the orientation θ_0. By using this initialisation strategy the range in the orientation axis of the local position search space can be reduced. Particles can then be concentrated on exploring the X and Y axes of the search space.

Experimental Setup

An experiment was carried out using images and odometry captured during a TBL movement. The path originated from the reference position of a previously learnt place. Figure 8.25 shows the odometry from the robot path along with locations along the path at which panoramic images of interest were captured. Figure 8.26 shows the estimated positions of the landmarks from the reference position and a panoramic image with the reference landmark templates. Using this information together with images captured along the path and the sensor models described previously, the robot



Estimated Landmark Positions from Reference Position

Reference Landmark Set

Figure 8.26: The setup of the local positioning experiments. The estimated landmark positions relative to the learnt place and the reference landmark set are also displayed.

performs local position estimation for each of the approximately 500 sensor samples along the path.

Throughout the sequence of images the landmark templates are tracked and their observed radial angle and the odometry at each corresponding sample point form the input to the particle filter performing local position estimation. Examples of the tracked landmarks at different locations along the path are included in Figure 8.27. The average landmark template correlation value of the landmark set for each image in the path is presented in Figure 8.28. The landmark tracking performance peaks when the images in the sequence correspond to locations along the path which are close to the reference position.

For each experimental run, the 2000 particles in the particle set were initially randomly distributed around the reference position in a Gaussian distribution with a variance of $100mm$. The local position estimate at each iteration of the particle filter, was said

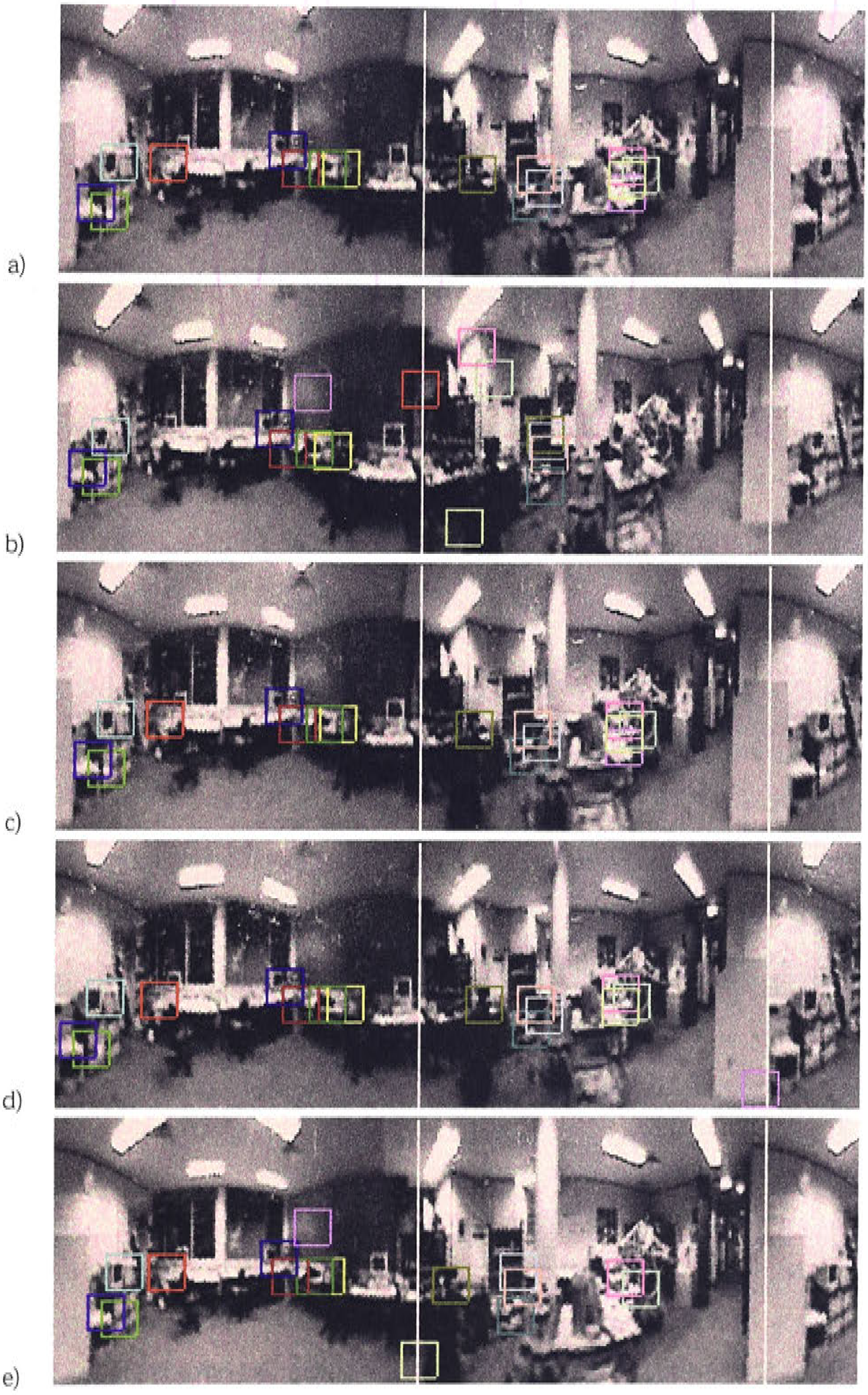


Figure 8.27: Tracked landmarks at identified locations along the TBL motion path.

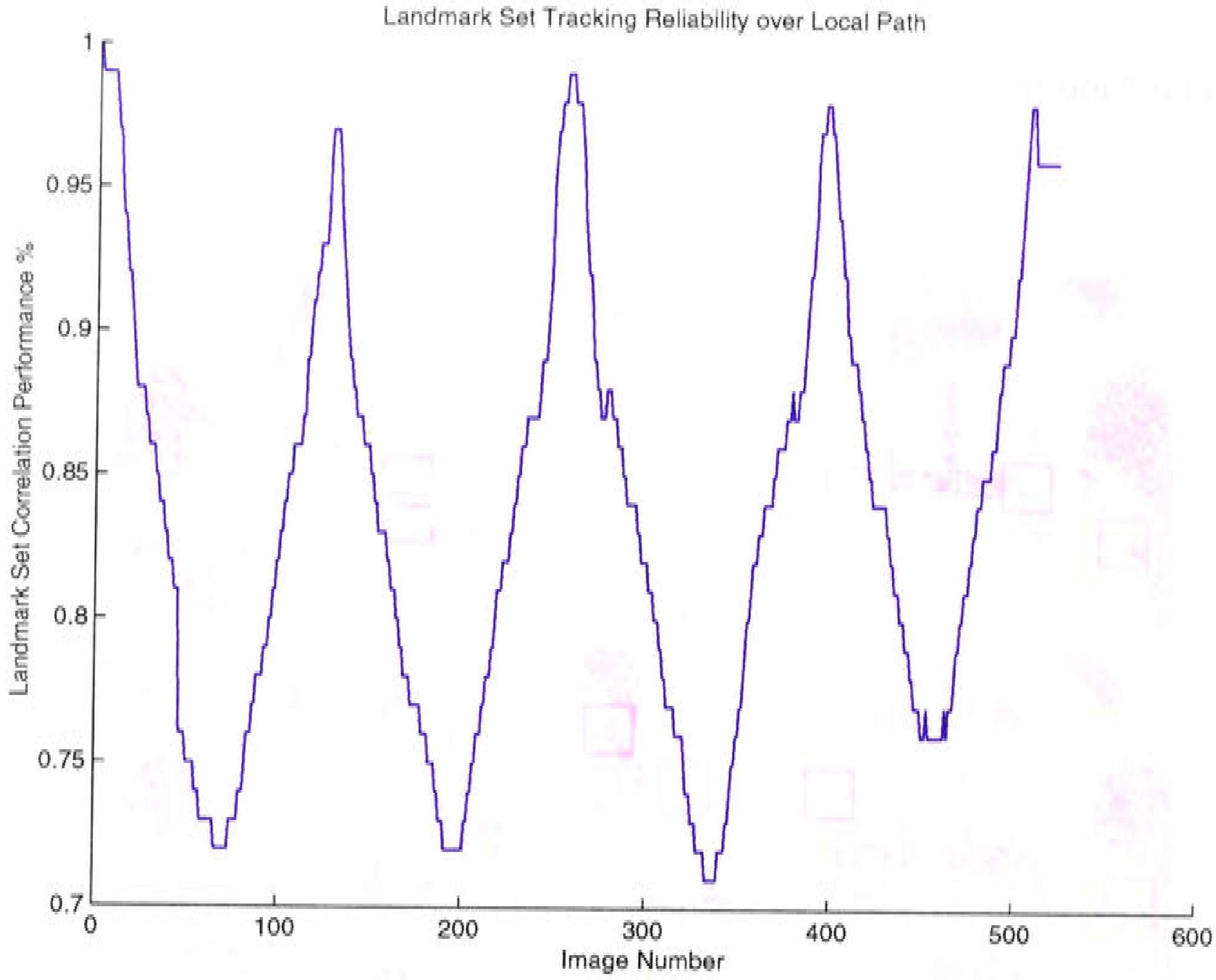


Figure 8.28: The Landmark Landmark Recognition Performance (LRP) for reference landmark set over the path traversed by the robot.

to be the robot pose represented by the particle in the particle set with the highest probability as measured by the sensor model.

The local position estimation using the particle filter approach was applied to the path described above. On three separate trials, the three different sensor models were evaluated for their value to local position estimation.

Assumed Depth Sensor Model Results

Figure 8.29 shows the local position estimation results using the assumed depth line intersection sensor model. The estimate is noisy and continuously jumps around, although the basic shape of the TBL movement can be observed. The noise in the position estimate is as bad as that in the estimate provided by the heuristic approach. Unlike the heuristic approach however, by incorporating the odometric information into the position estimate as occurs using the particle filter approach, the scale of the

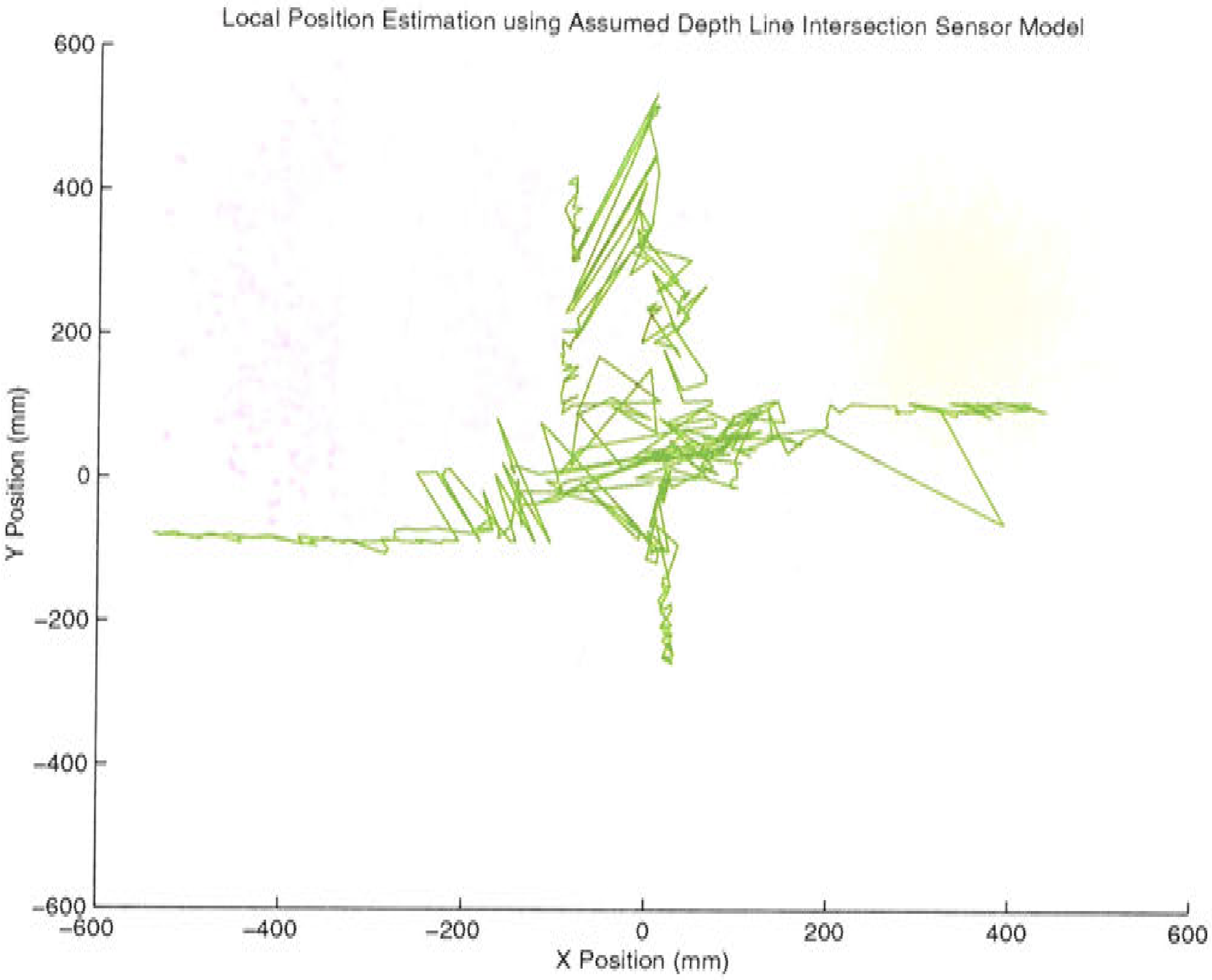


Figure 8.29: The estimated local position along the TBL path using probabilistic algorithm and assumed landmark depths.

robot movement is kept, and the estimated path is not as skewed. The noise present in the estimated path can be attributed to the simplicity of the assumed depth sensor model and the method used for obtaining a position estimate from the particle set.

The assumption that landmarks representing a particular place are all at a constant depth means that the robot pose probability distribution defined by such a model is not always correct and can be inconsistent between samples. In addition, the large depth variance that such an assumption necessitates results in large areas of the state space being rewarded for a given observation. These two problems lead to a particle set that is distributed over a large area and has associated particle probability values that vary greatly.

Figure 8.30 shows the particle set distribution produced by the assumed depth sensor model for various observations over the initial X axis movement of the robot path. The particles are initially distributed around the reference position as shown by the yellow cloud of points. The distribution then moves along the X axis due to the introduction

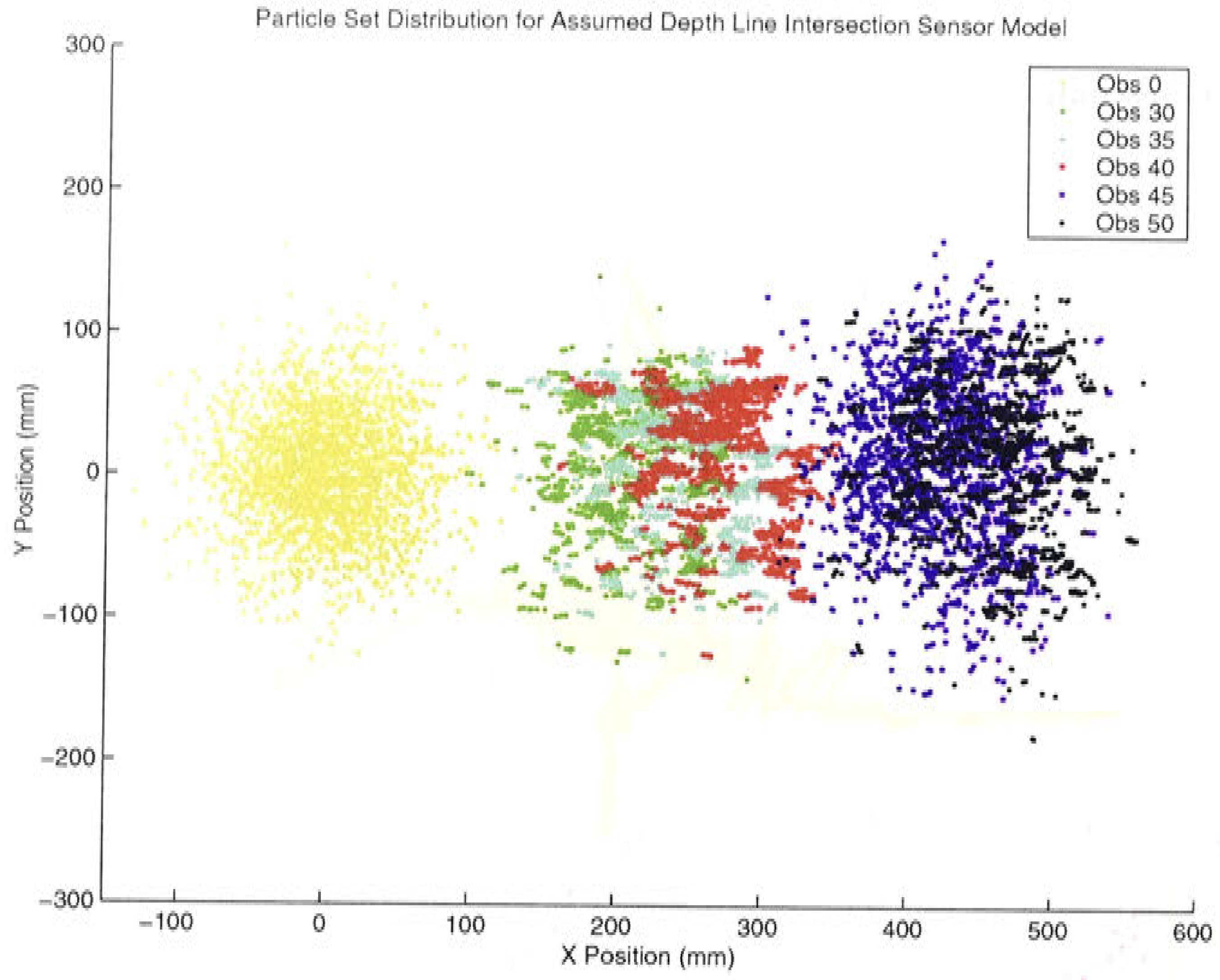
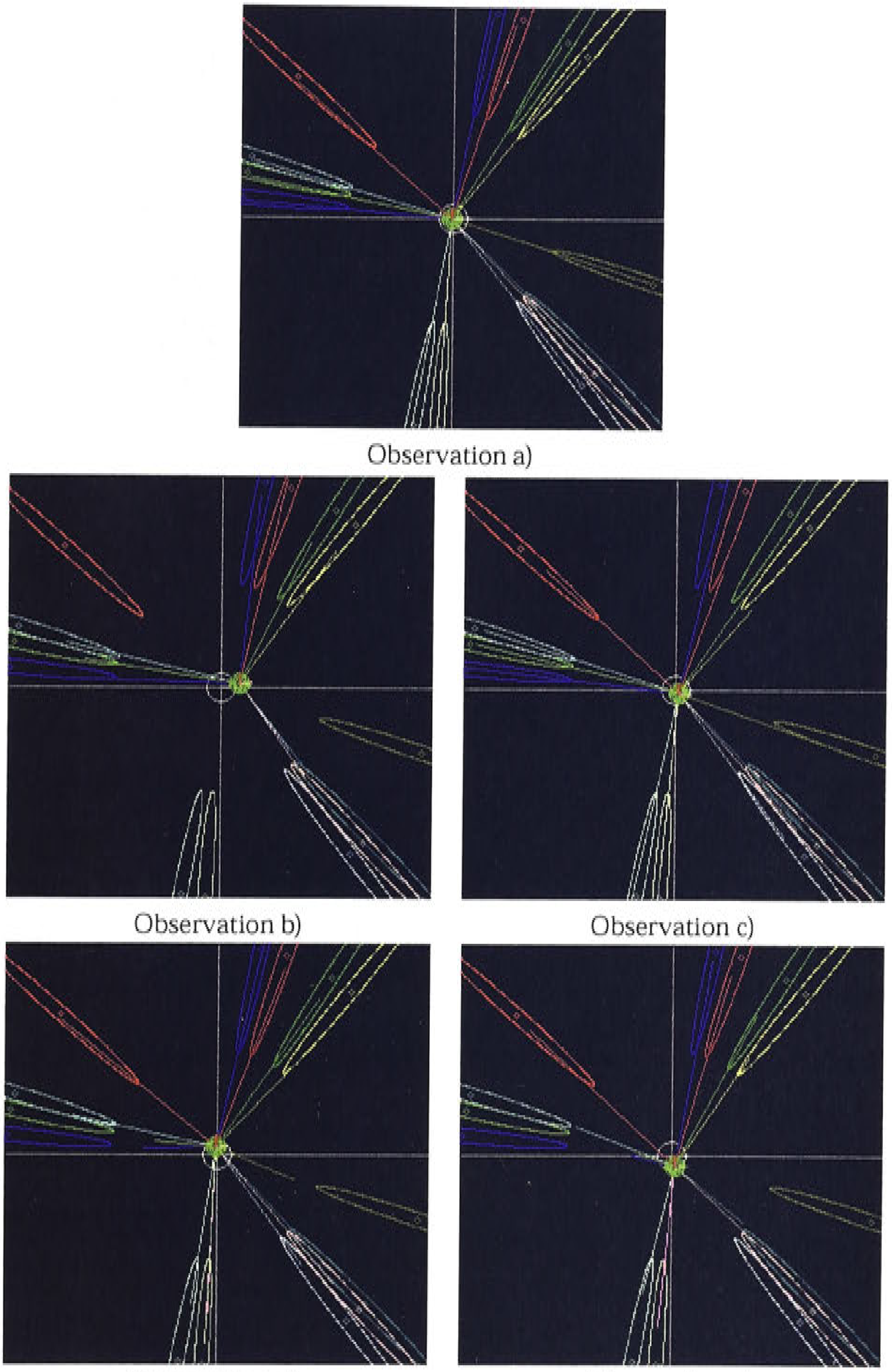


Figure 8.30: The particle set distribution for various observations during the assumed depth sensor model local positioning experiment.

of the odometry information. The assumed depth sensor model does not condense the particle set distribution towards a local position estimate. As a result the underlying distribution is inconsistent, thereby causing the measure of the particle containing the highest probability in the set to jump around between iterations of the particle filter cycle.

Although the sensor model does not condense the particle set, it can be observed that the particle set does not continuously expand as would occur if the sensor model was contributing no information at all. The particle set distribution does expand in one instance in the figure, at observation 45. This is because between observation 40 and 45 the sensor sampling rate decreased sharply and a large amount of noise was introduced into the system, as there was a large time period when no sensor data was incorporated into the filter.

Imprecise geometric modelling leads to the errors in the sensor model as shown in Figure 8.31. Observations from the various identified images along the TBL motion



Observation a)

Observation b)

Observation c)

Observation d)

Observation e)

Figure 8.31: Example observations from the local positioning experiment using the assumed depth sensor model. The images show the observations of landmarks in the landmark set from the most likely particle. The landmarks are shown at their assumed depth. The length of the landmark observation lines reflect the contribution to the particle's probability measure.

path are shown. The observation diagrams show the landmark observation angles from the robot pose associated with the particle with the highest probability measure. The coordinate system is centered about the reference position of the learnt place. The landmarks are drawn as ellipses at the assumed depth and variance. The length of the landmark observation lines is proportional to the contribution of the particular landmark to the particles probability measure. This contribution is dependent on the output of the sensor model and also the template matching correlation value.

It can be seen that at various points along the robot path that it is difficult to identify a single robot pose from which it is possible to observe all landmarks at the assumed depths. This inability to identify one clear, small region of being the most likely current robot pose, leads to the diffuse and inconsistent local position estimation evident when using the assumed depth line intersection model.

Estimated Depth Sensor Model Results

Figure 8.32 shows the results of using the estimated depth line intersection sensor for local position estimation. The accuracy of the position estimate is clearly superior than the assumed depth model. Knowledge of the landmark depths allows the sensor model to produce tighter more consistent robot pose probability distributions. This allows the particle set distribution to condense on the correct robot pose hypothesis and brings stability to the choice of most likely particle. Some error in the position estimate exists but the size and scale of the motion is preserved.

The relatively noise free results produced above are due to the sensor model condensing the particle set around a stable and small probability peak in the state space. The improvement in the particle set distribution is shown in Figure 8.33. The particle set is initially distributed around the reference position. At observations 30, 35 and 40 the particle set has condensed to a stable circular area approximately $50mm$ in diameter.

This distribution spreads out due to the period of motion without sensor data prior to observation 45. This expansion is reversed by observation 50 although it appears the estimate condenses around an incorrect sample pose state. This may be due to an insufficient number of particles in the particle set to model the large motion between delayed observation samples without missing critical parts of the underlying proba-

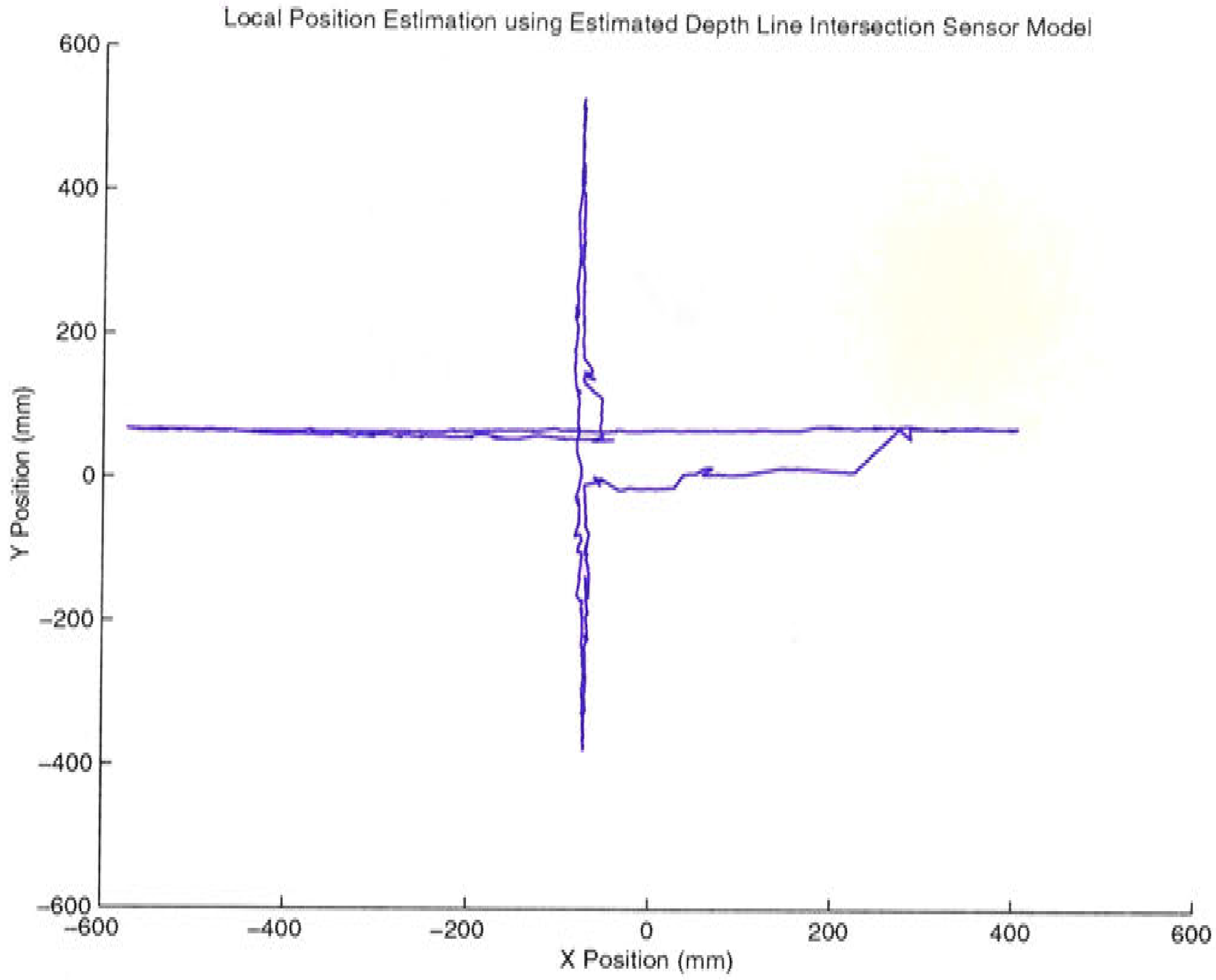


Figure 8.32: The estimated local position along the TBL path using probabilistic algorithm and estimated landmark depths.

bility distribution. The condensation about an incorrect point is evident in the plot of local position estimation, the estimated Y position jumps from 0 to approximately $50mm$. The resulting estimate does not increase in error, this infers that the sensor model is correctly constraining the local position estimation process.

Figure 8.34 shows example observations from the most likely particle for various images along the path. The length of the landmark observation lines denote the contribution to the particle's probability measure. In this model all reasonable landmark measurements make a contribution, resulting in equal lengths in the observation lines.

It is evident the estimated depth sensor model provides a more accurate, tighter, and more stable local position estimate than that obtained using the assumed depth sensor model.

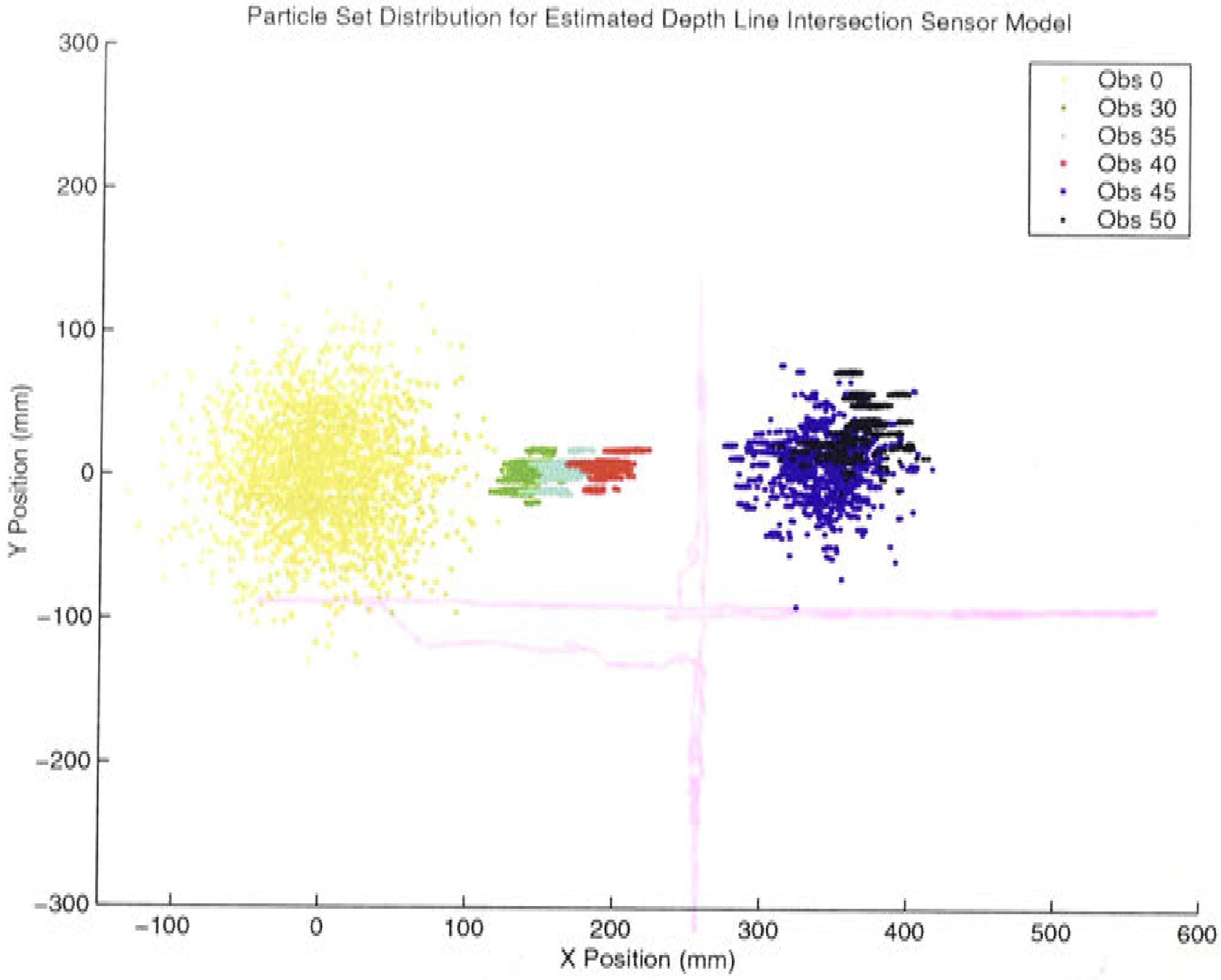
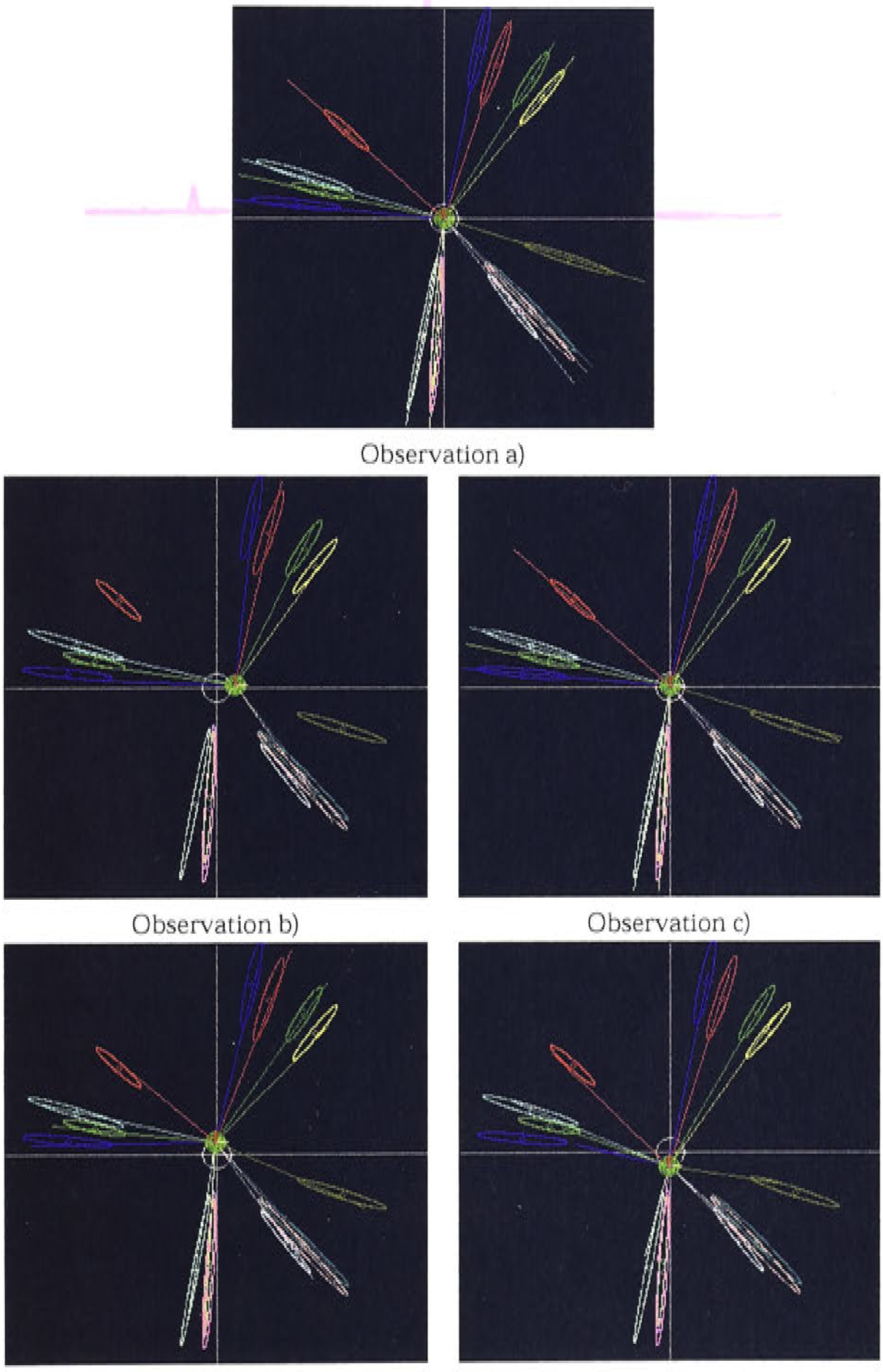


Figure 8.33: The particle set distribution for various observations during the estimated depth sensor model local positioning experiment.

Ellipsoid Sensor Model Results

Figure 8.35 shows the result of estimating the local position of the mobile robot system using the ellipsoid line intersection sensor model. The figure shows that the use of the ellipse model produces an accurate, stable local position estimates when compared to the other two models which are based on simple line intersections. The more complex geometric model allows the sensor model to tightly constrain the underlying robot position probability distribution, especially when observations are made that are parallel to the reference observations.

The sampled probability distribution produced by the particle filter in conjunction with the ellipsoid sensor model at various points of time is presented in Figure 8.36. The initial distribution is randomly spread about the reference position. As the robot moves along the X axis, the ellipsoid sensor model tightly constrains the particle distribution into a small region of the underlying robot pose probability function. This



Observation a)

Observation b)

Observation c)

Observation d)

Observation e)

Figure 8.34: Example observations from the local positioning experiment using the estimated depth sensor model. The images show the observations of landmarks in the landmark set from the most current most probable particle. The landmarks are shown at their estimated depth. The length of the landmark observation lines reflect their contribution to the particles probability measure.

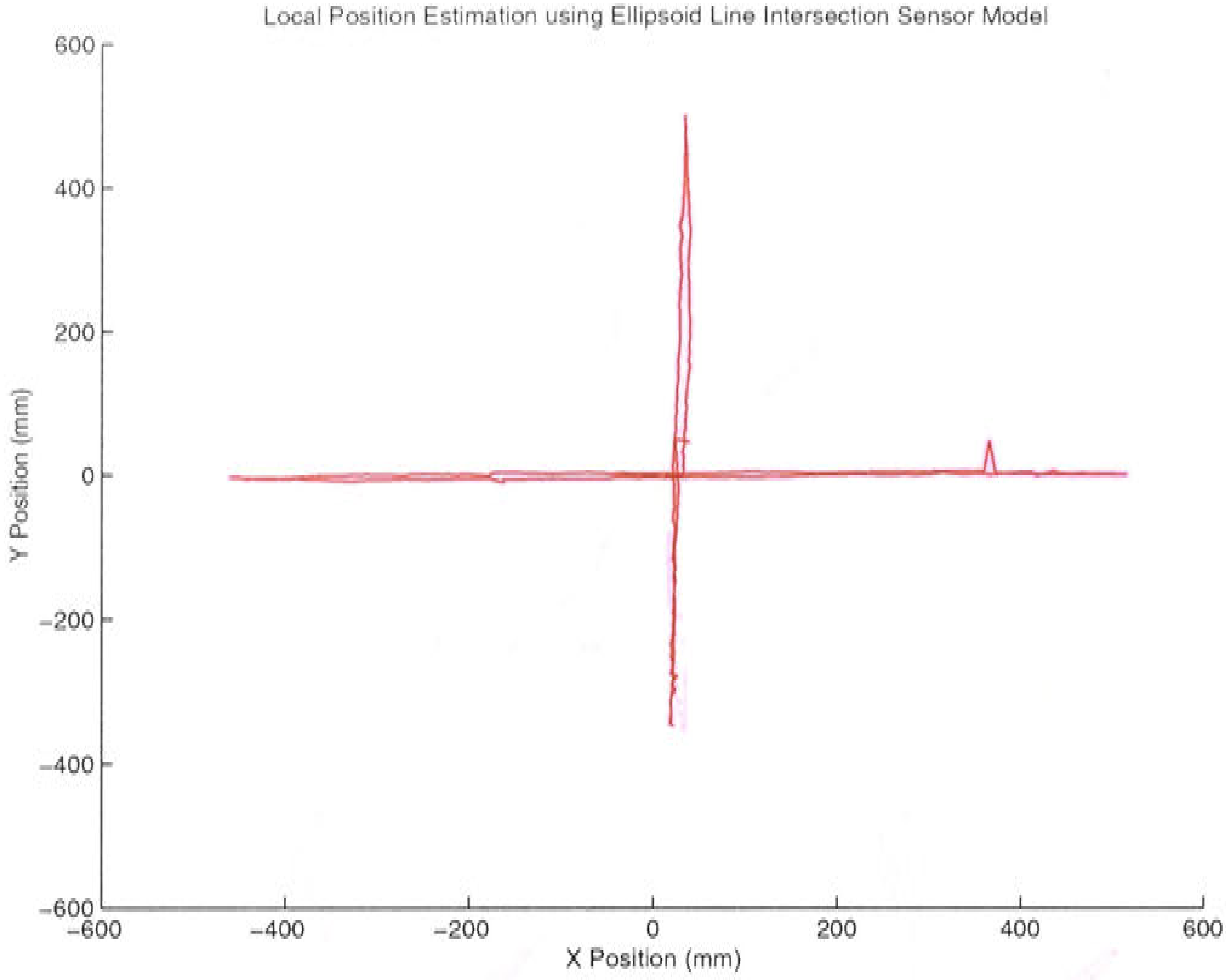


Figure 8.35: The estimated local position along the TBL path using probabilistic algorithm and the ellipsoid line intersection model

identified area accurately reflects the true position of the mobile robot as it traversed the path.

The expansion of the particle set due to the prolonged absence of sensor data between observations 40 and 45 is present, but is curtailed by the improved sensor model, and the sampled distribution of the expanded set condenses onto the correct robot pose by observation 45.

The ability of the ellipsoid sensor model to closely discriminate between hypothesised robot poses when incorporating current observations into the local position estimate is shown in Figure 8.37. The figure shows several example observations from various positions along the robot path. The observations from each figure are plotted from the particle which has been identified as the most likely mobile robot pose by the sensor model. The length of the lines representing the landmark observations are proportional to the observations contribution towards the particle's probability measure. When compared to the observations from the previous two models it is apparent

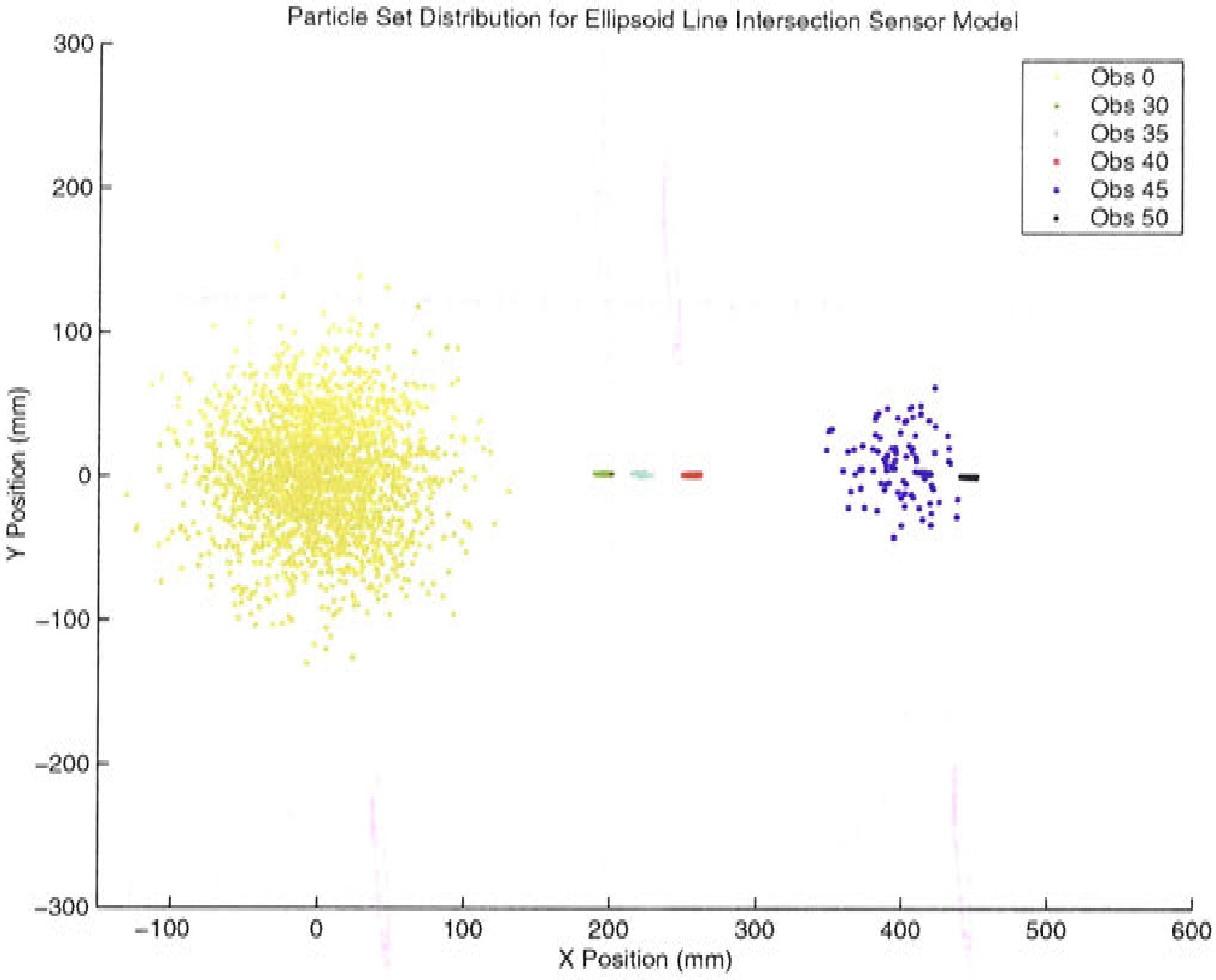


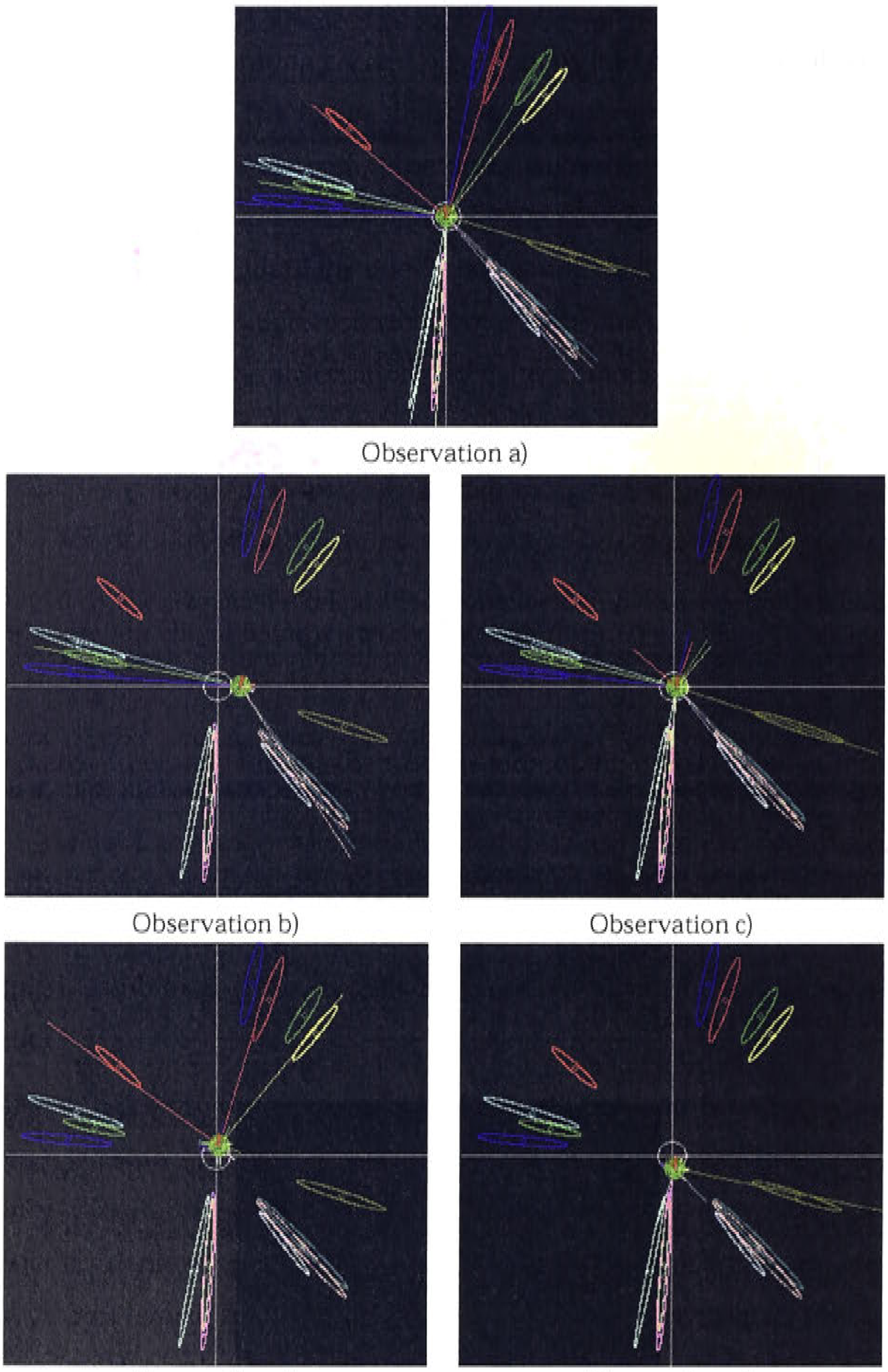
Figure 8.36: The particle set distribution for various observations during the ellipsoid line intersection sensor model local positioning experiment.

that the current model discriminates more finely between observation angles, as there are fewer long observation lines for the ellipsoid model. This shows that the current model only rewards the most probable observations, thus leading to more tightly constrained probability distributions and ultimately a more accurate local position estimate.

The ability of the ellipsoid sensor model to more finely discriminate between competing robot pose hypotheses produces a more accurate robot pose probability distribution and therefore the particle filter condenses around a more accurate local position estimate than the previous two sensor models.

Sensor Model Comparison

The superiority of the ellipsoid line intersection sensor model compared to the assumed depth and estimated line intersection sensor models is shown in Figure 8.38.



Observation a)

Observation b)

Observation c)

Observation d)

Observation e)

Figure 8.37: Example observations from the local positioning experiment using the ellipsoid line intersection depth sensor model. The images show the observations of landmarks in the landmark set from the most current most likely particle. The landmarks are shown at their estimated depth with the uncertainty ellipse surrounding them. The length of the landmark observation lines reflect the contribution to the particle's probability measure.

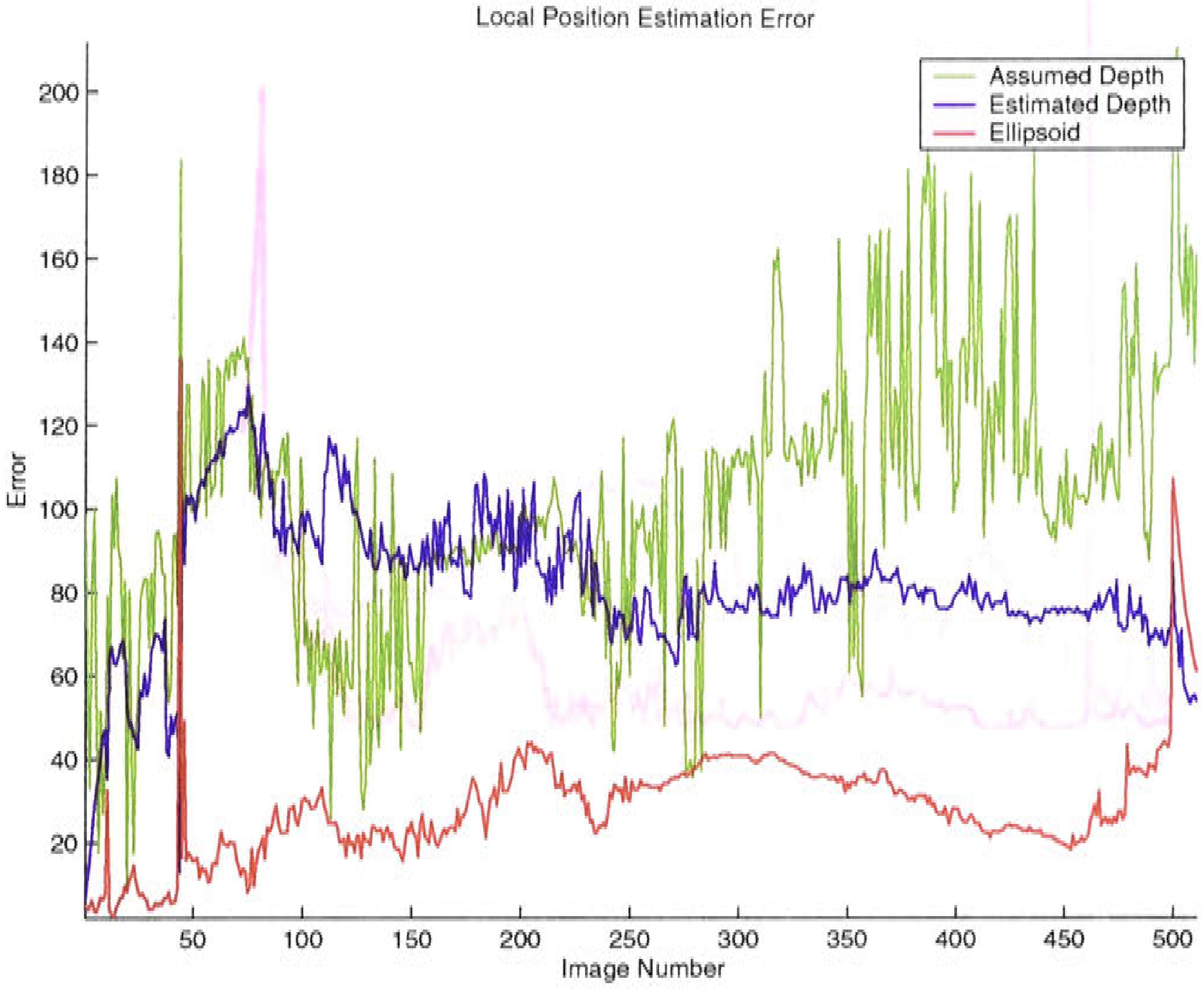


Figure 8.38: Error in the combined X and Y location estimate for the three sensor models.

The error of the local position estimate at each image captured along the example path for each of the three sensor modes is shown. The error (distance of x, y estimate from the corresponding x, y odometry measurement) of the assumed depth model is plotted in green, the estimated depth model in blue and the ellipsoid model in red. The error in the local position estimate obtained with the ellipsoid model appears to reach a peak of approximately $40mm$ (if outliers are removed) with a mean error of approximately $30mm$, which is about half that of its nearest rival.

The peaks in the error plot that can be observed around the $45th$ image in the robot path are associated with the expansion of the particle distribution in all sensor models that occurred due to the delay in sensor sampling discussed above.

Figure 8.39 shows the individual X and Y axes components of the ellipsoid sensor model error measurement presented in Figure 8.38. The increases in error occur mainly along the axis of current motion. The first part of the robot path consisted of

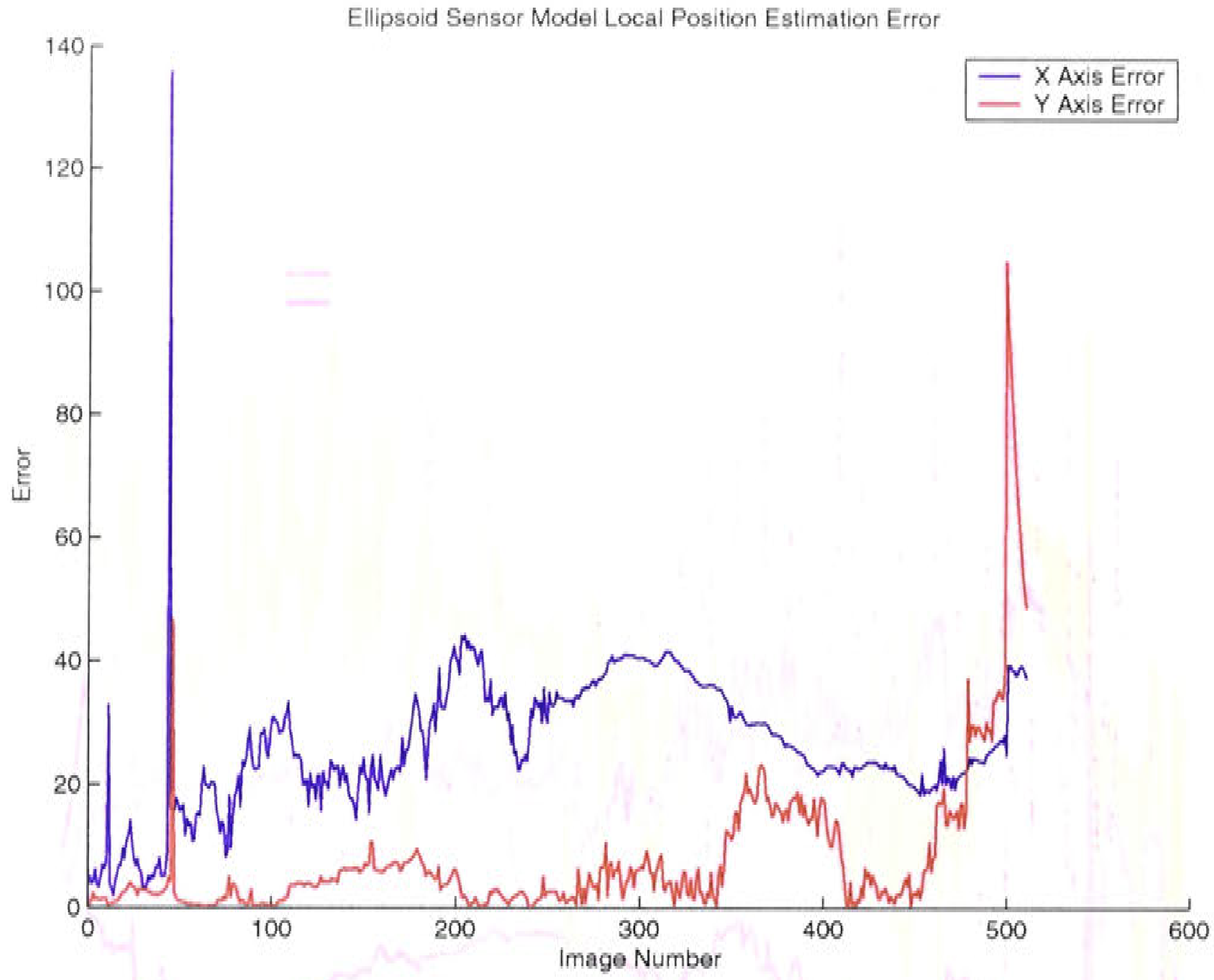


Figure 8.39: The separate X and Y error components for the ellipsoid line model local position estimates.

motion solely along the X axis, whereas the second half of the path involved mostly Y axis motion. In the axes error plots the X axis error can be seen to grow predominantly in the first half of the path, the Y axis in the second half. This implies that the direction of robot motion was estimated accurately, and there was some error in the estimated distance travelled along that direction. An elongated particle filter distribution supports this implication. This result suggests that the ellipsoid model tightly controls the direction of motion and the orientation from which the robot makes the observations.

This idea is called into question when the orientation estimation error is investigated. Figure 8.40 shows the orientation estimation error for each of the three sensor models used in the local position estimation process. The green plot shows the error in the assumed depth model orientation error, the blue line that of the estimated depth model and the red line that of the ellipsoid sensor model. The two simple line intersection model produce orientation estimates which are near the constant odometry orienta-

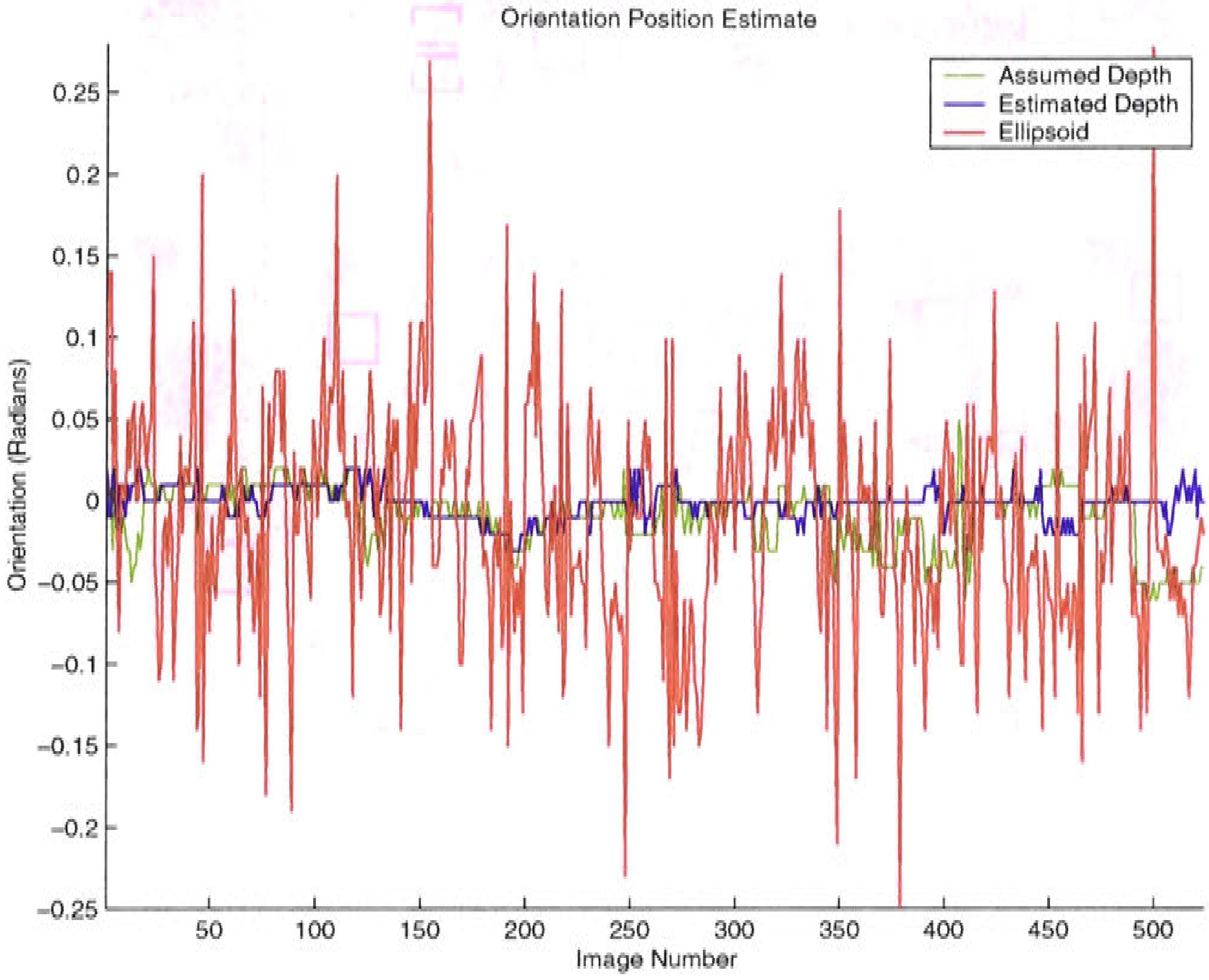
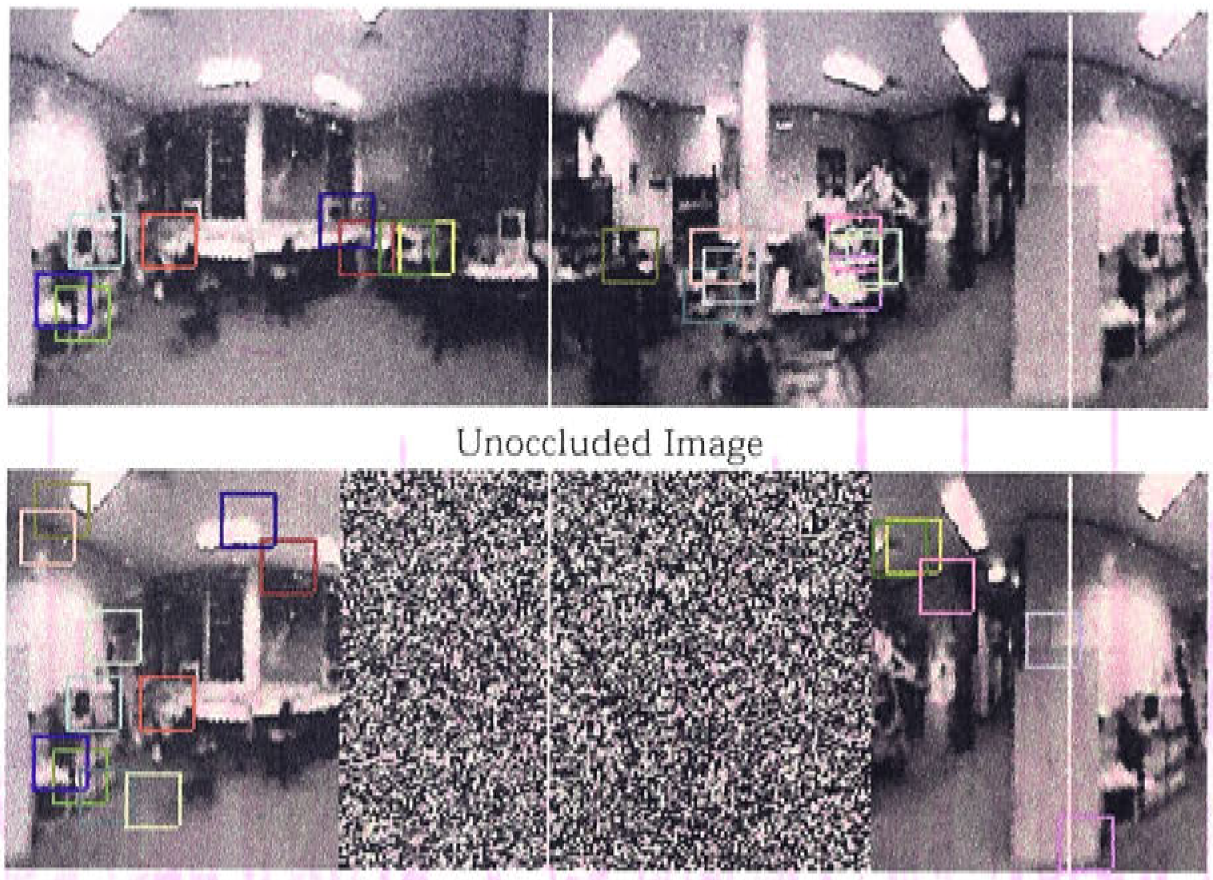


Figure 8.40: Orientation estimate error for the three sensor models.

tion measure of $0\,\mathrm{deg}$ throughout the entire path, whereas the orientation estimate error produced by the ellipsoid model varies greatly in comparison.

This error is alleviated when it is combined with the steering angle estimate, although the error remains large when compared to the orientation error of the other two models. The oscillation in orientation could be caused by the continually adjustment of the steering angle by the robot controller to obtain a straight translation, which would not be reflected in the odometric observation measurements, although steering changes of up to $10 - 15\,\mathrm{deg}$ do seem excessive. This result is strange and it is not currently explained satisfactorily. The fluctuation in steering and observation angles observed in the ellipsoid model, and its inherent sensitivity to orientation results in accurate position estimation of the more easy to validate x and y parameters.



Unoccluded Image

Occluded Image

Figure 8.41: An example of tracked landmarks in an unoccluded and an occluded panoramic image. 180 degrees of the image is occluded by white noise.

Local Positioning under Occlusion

A robot system performing local position estimation in dynamic environments must be able to overcome the problem of landmarks becoming occluded by moving objects. If a moving object passes between the panoramic sensor and the physical landmark in the environment, the landmark tracking process for that particular landmark is interrupted and potentially incorrect observation data will be introduced into the estimation process.

To perform robust local position estimation we need overcome the noise introduced by occlusion. Our research attempts to handle the problem of occlusion by evenly distributing landmarks throughout the image and by incorporating the template matching correlation reliability (Chapter 4) into the probability calculations of the sensor model, as defined previously.

In order to test the ability to estimate a mobile robot's local position in the presence of visual scene occlusion, experiments similar to the local positioning experiments were performed with altered input images. Areas in the panoramic images captured over

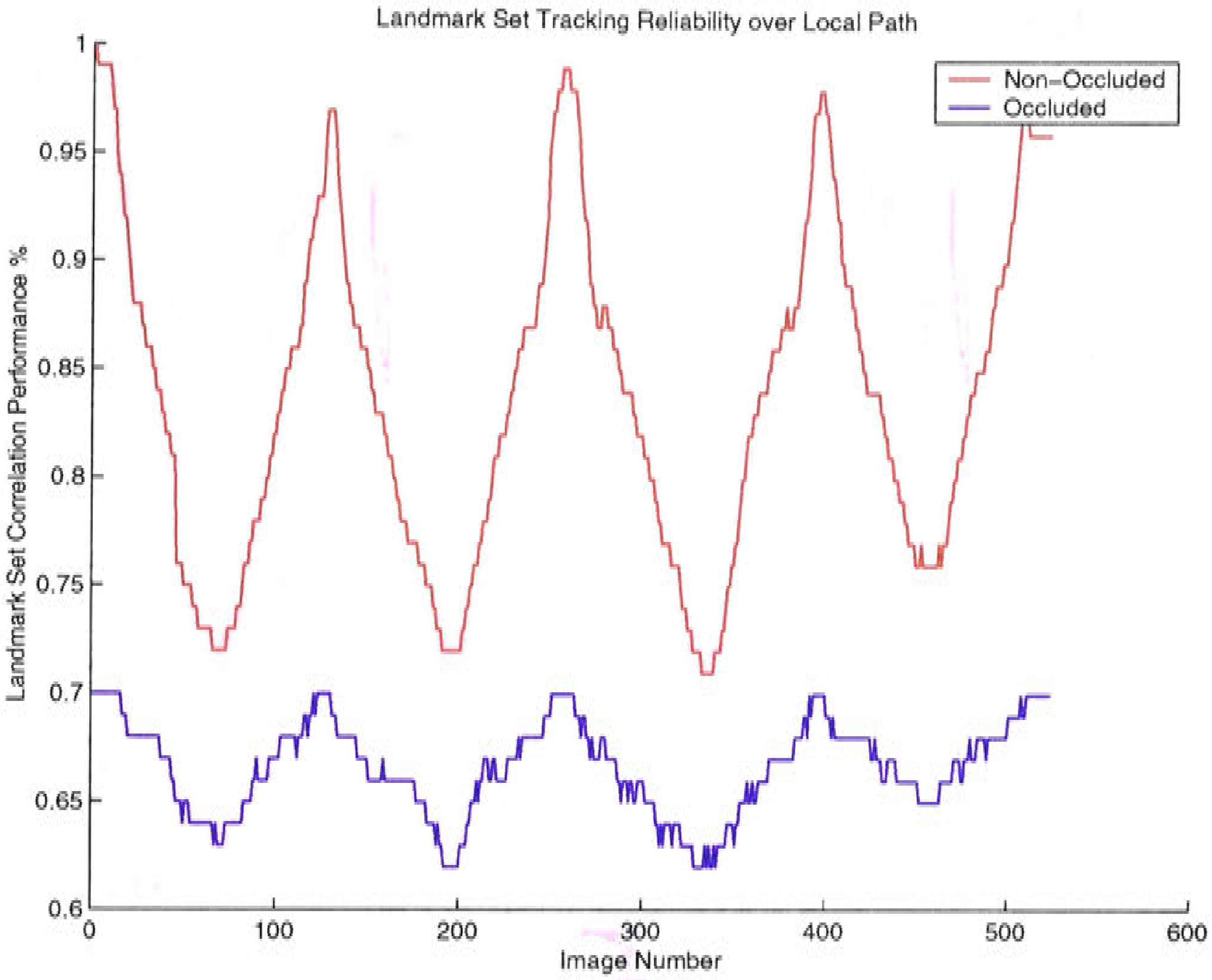


Figure 8.42: The Landmark Recognition Performance (LRP) for the normal and occluded image sets.

an example robot path were drawn over with white noise prior to landmark tracking, thereby eliminating views of the majority of landmarks and causing false observation angles to be reported. White noise was used in order for the landmark tracking system to have no chance in detecting a landmark anywhere near its original position, which might happen in the real world when a small moving object temporarily occludes a landmark from view.

A region of 180 deg of the panoramic images was over drawn with white noise. An example is shown in Figure 8.41. The white noise region occludes three quarters of the landmarks used to represent the learnt place, leaving only four landmarks from one sector of the visual field visible. The average landmark tracking correlation performance for occluded versus non-occluded images over the robot path is shown in Figure 8.42. The LRP performance for the occluded image set is very poor, not rising above the 0.7 mark where the performance for non-occluded images does not fall below the 0.75 mark. An LRP of 1.0 means perfect correlation for all landmarks in the

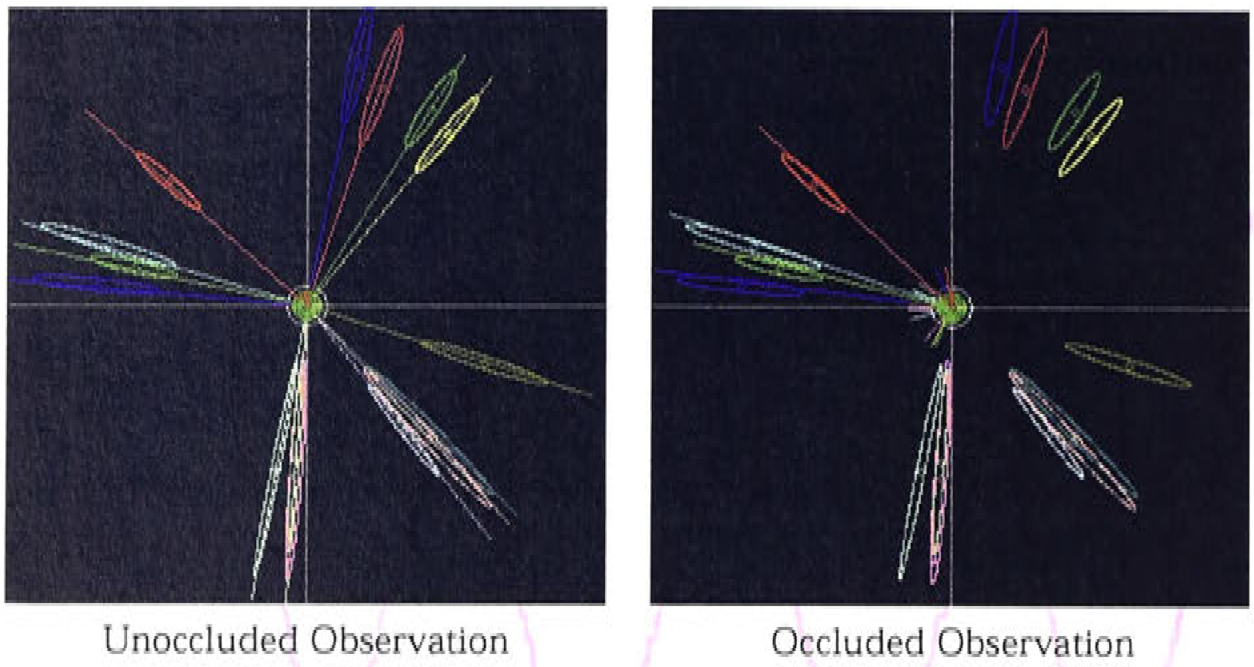


Figure 8.43: An example observation produced by landmark tracking in an unoccluded and occluded image. The observation lines of occluded landmarks are short due to poor tracking results.

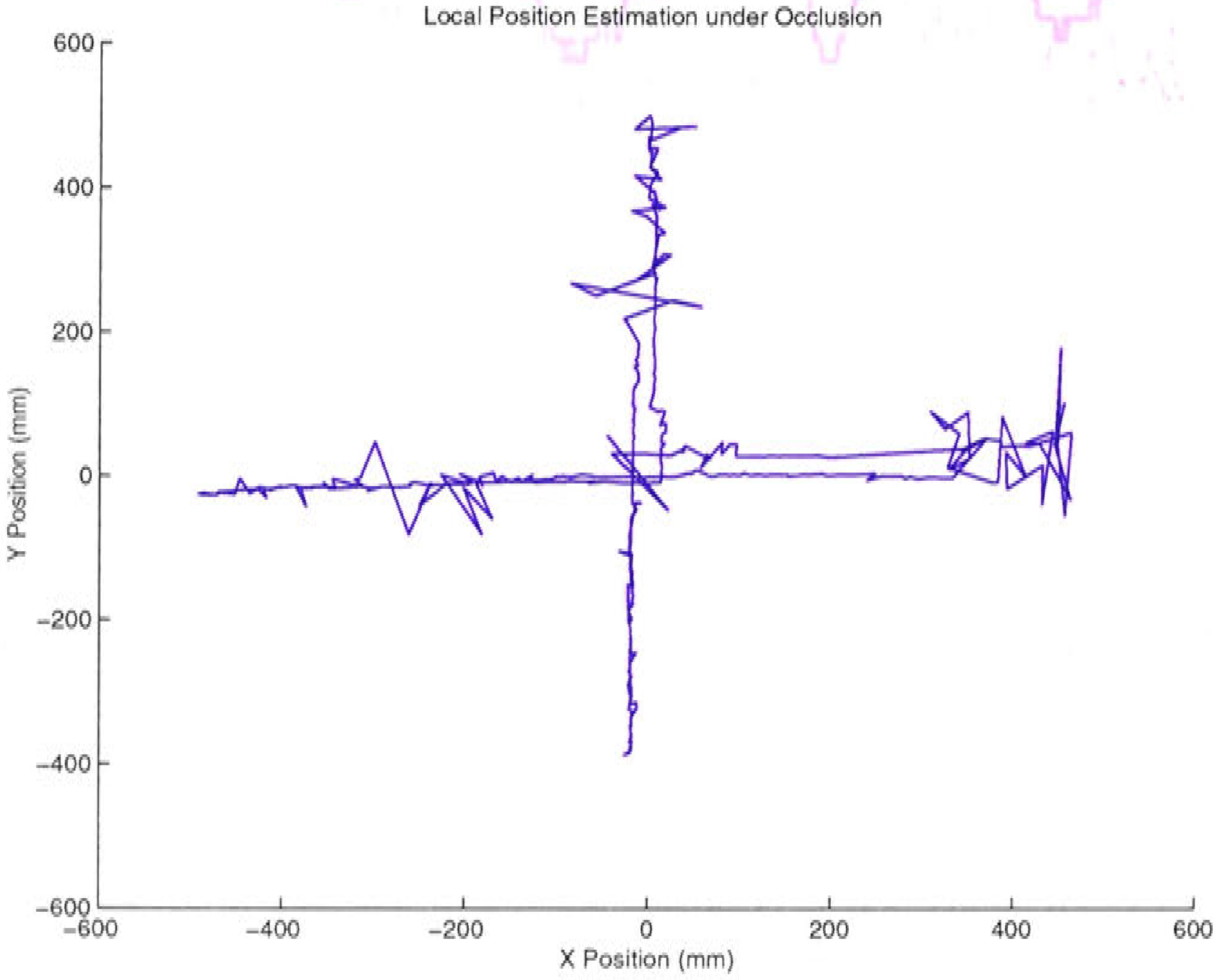


Figure 8.44: The estimated local position along the TBL path using probabilistic algorithm and the ellipsoid line intersection model with an occluded image set.

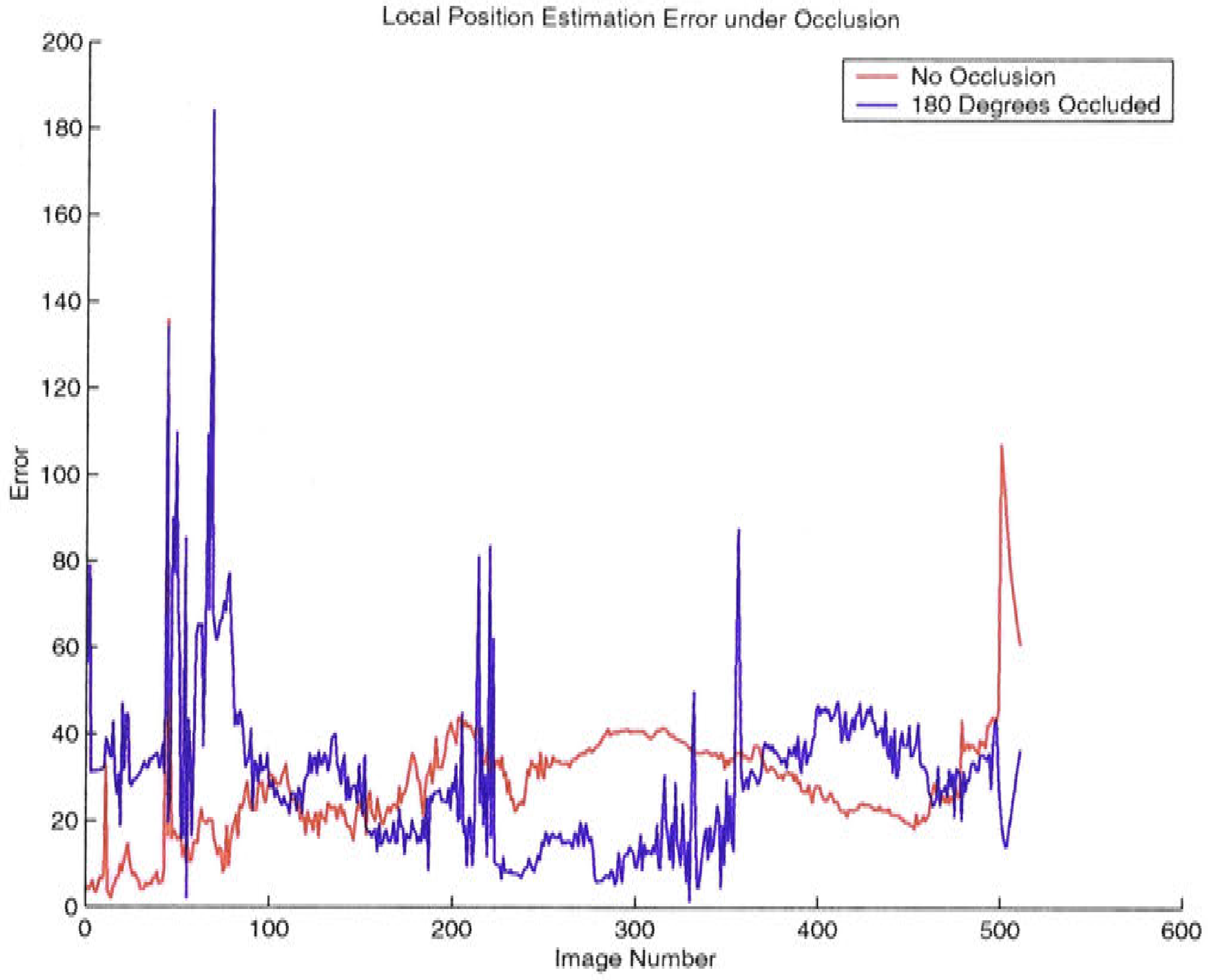


Figure 8.45: A comparison of the error in local position estimate using the ellipsoid line intersection sensor model over normal and occluded image sets.

landmark set. An observation using poor landmark tracking results is presented in Figure 8.43. There are only four reliable landmark observations from which to base the local position estimate, the occluded landmark observations are all at an angle which disagree with the displayed robot pose.

The results of performing local position estimation with the occluded image set are shown in Figure 8.44. Although the local position estimate is noisier than when using non-occluded images, the basic shape of the movement was estimated correctly. In fact the error in the occluded local position estimate is comparatively equal to that of non-occluded position estimation. These results show that the current system can perform accurate local position estimate even in the presence of large scale continuous occlusion, provided that a small number of landmarks can still reliably be observed. The number of landmarks necessary to maintain accurate local position estimation varies on the orientation and depth of the landmarks. In general, three landmarks from distinct sectors of the environment is sufficient for accurate position estimation.

Local Positioning and the Data Association Problem

The data association problem in mobile robot localisation is the problem of matching sensory perceptions with internal representations of the environment. In particular, the difficulty lies in associating the current sensory data with the correct portion of the internal representation. Localisation methods which use abstracted features to represent the environment are especially susceptible to this problem as they can contain many ambiguous situations.

Our research uses a visual landmark representation which actively seeks unique landmark templates therefore incorrect data association is less of a problem. The problem might still occur in situations where there is a sparse visual scene. A data association problem could occur in the matching process between the places landmark set and the current panoramic view, when an incorrect landmark observation is made with a high recognition measure. For incorrect position tracking results to occur, not only would a number of landmarks in a set have to be strongly mis-matched, but the pattern in which they are distributed throughout the visual scene would have to be consistent with the reference observations. Therefore it is highly unlikely that an odd occurrence of mis-matching landmarks with a high correlation measure will significantly affect the process of local position estimation in the current system.

The ability of the current system to perform local position estimation in the face of data mis-association was tested by manually inducing incorrect landmark tracking over the example path. Mis-associated landmarks were reported to have high correlation levels in incorrect regions of the visual scene. Figures 8.46 and 8.47 show an example of landmark tracking and the resulting observation when five landmarks from the landmark set were mis-associated. The observation image shows that unlike the mis-matched observations seen in the occlusion experiment, the observations lines are consistently long, representing the high confidence the system has in these observations. By consistently returning high correlation measures for incorrectly matched landmarks, the local position estimation process is subject to ambiguous and misleading information, as is seen in the data association problem.

Figure 8.48 shows the results of estimating the robot position when five landmarks consistently report mis-associated observations. Although there is noise, especially

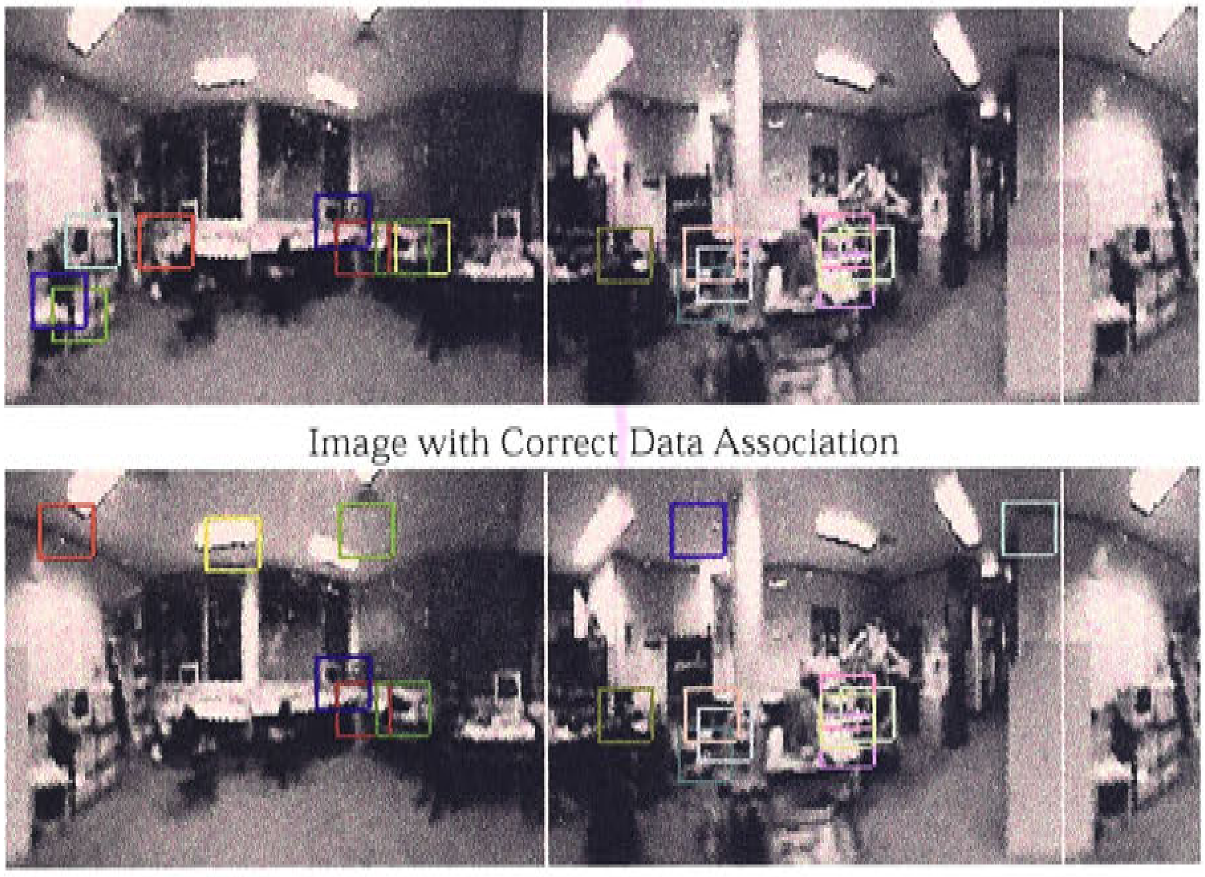
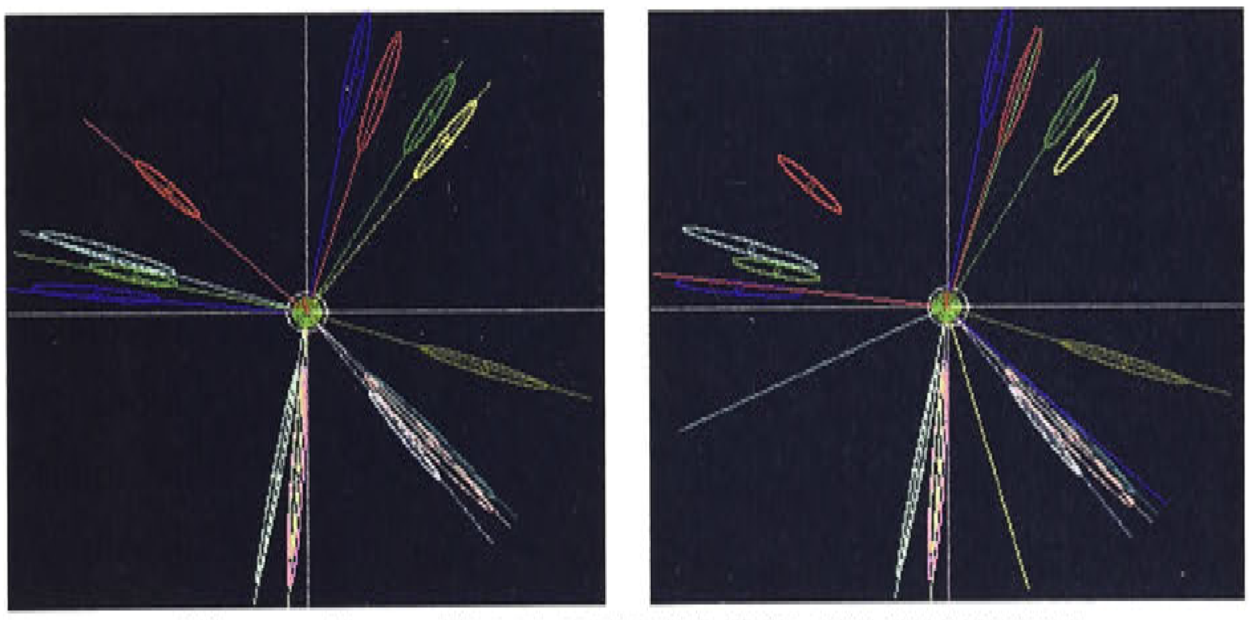


Image with Correct Data Association

Image with Data Mis-association

Figure 8.46: An example of tracked landmarks with and without data mis-association. The five landmarks across the top of the image are mis-associated.



Observation without and with data mis-association

Figure 8.47: An example of data mis-association showing the resulting observation image. The long observation lines which do not intersect with the correct landmarks are observations of mis-associated landmarks.

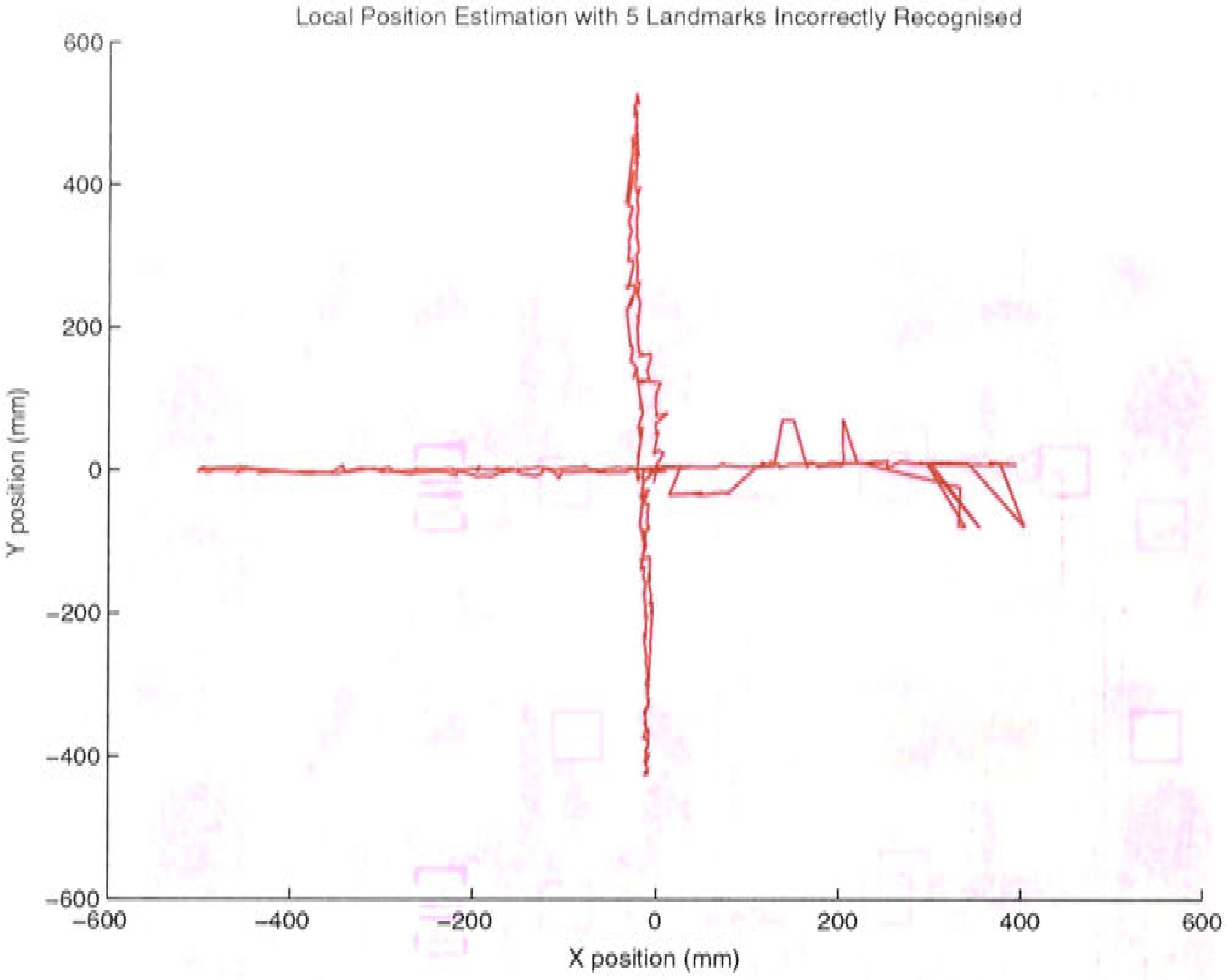


Figure 8.48: The estimated local position along the TBL path where 5 landmarks are incorrectly tracked through the entire path..

away from the center of the movement, the general path of the robot has been estimated well. The estimation error is shown in Figure 8.49 along with the results for experiments with one and three landmarks being mis-associated. The error is considerably greater than when there are no data mis-associations, growing to a maximum of about $12cm$. The peaks in the error plot correlate with locations along the path when the robot is most distant from the reference position. This suggests that data mis-association accentuates the reduction in position information available as the robot leaves the area immediately surrounding the reference position. This leads to an inability to constrain the particle set and a noisier position estimates at the extremes of the TBL movement.

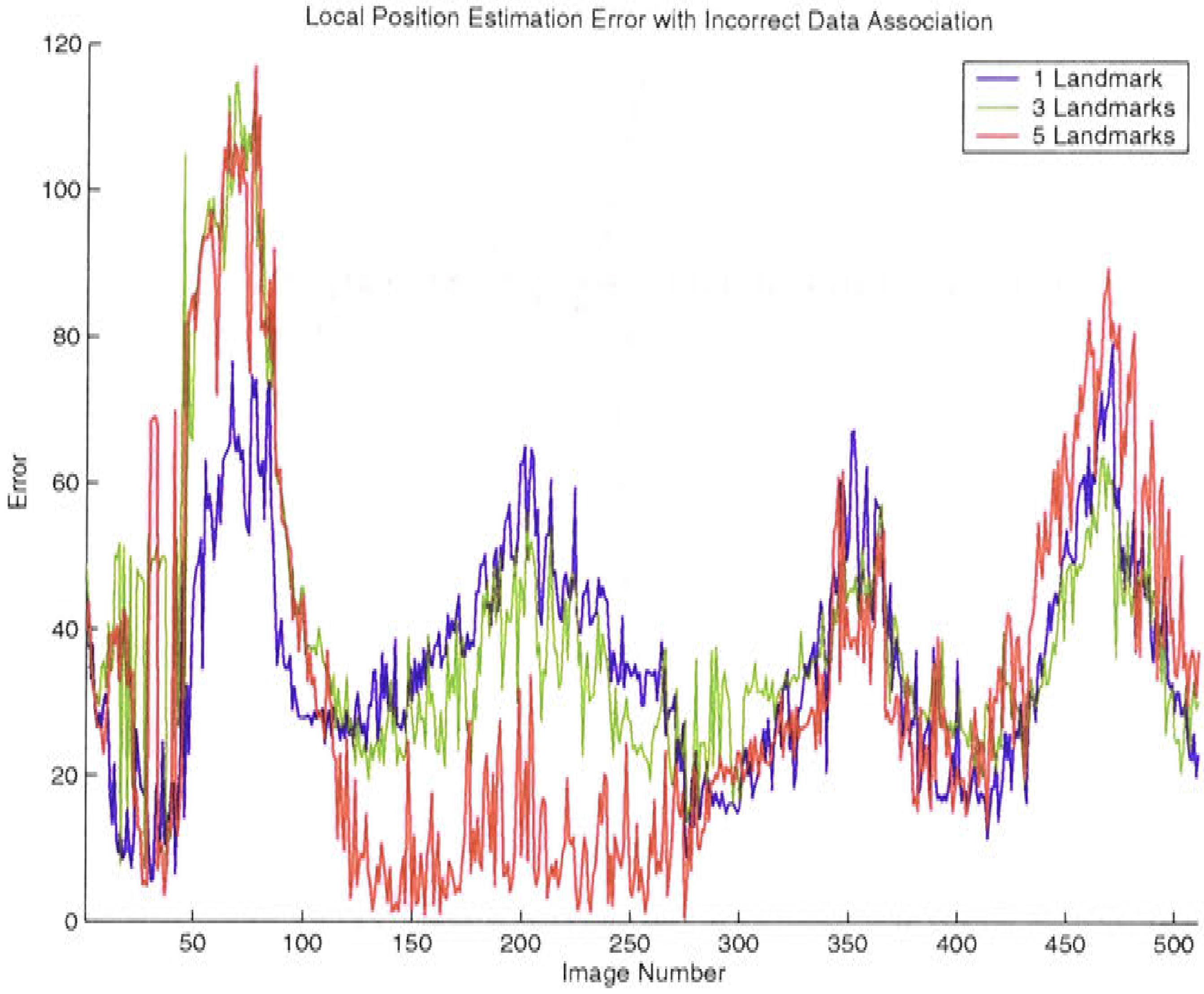


Figure 8.49: A comparison of the error in local position estimate when 1, 3 and 5 landmarks suffer from data mis-association.

8.2 Position Tracking Between Places

The previous section showed how it is possible for a mobile robot system using visual landmarks in panoramic sensors to accurately localise relative to the reference position of a learnt place. An accurate local position estimate allows a robot to perform precise navigation tasks within the area surrounding the learnt place. In order for a mobile robot system to be useful it needs to be able to navigate to places beyond that covered by a single learnt place in a topological map. This means navigating between places in the topological map while maintaining a position estimate.

The benefits of topological mapping derive from their sparse representation. A problem associated with a spare representation is that it can be difficult to implement such a representation which captures the relationships between each place in the map on a global reference frame. The desirability of maintaining a global frame of reference is questionable as preserving the accuracy of such a reference frame over large distances can is a difficult problem in itself. In topological maps position estimates can only be made relative to the reference positions of places in the map. The problem of maintaining a position estimate while navigating between places in a topological map means that at some stage during the movement between two particular places, a robot must conceptually switch its localisation reference frame from one place to another.

Passing Position Estimates Between Places

A mobile robot moving away from a learnt place S_1 performs local position estimation relative to the reference position of that place. As it approaches another place in the map, S_2, the landmark recognition performance for S_1 will be decreasing while that for place S_2 will be increasing. At some point during the movement, the landmark recognition performance for landmarks representing place S_2 will rise above that of S_1, signalling that the robot is now more likely to be nearer to place S_2. At this stage it has a local position estimate relative to the reference position of S_1. The task now is to transform the position estimate so it is relative to place S_2, which is now providing more reliable landmark tracking information.

The key to making this transformation is the existence of a transition $T_{S_1 \leftrightarrow S_2}$ between the two places. Of course if this particular path has not been travelled before, there will not be an existing transition and one must be defined according to the method described in Chapter 7. Assuming that there is already an existing transition, or that one has just been defined, the position estimate can now be passed between places using the spatial information contained in the transition definition.

Figure 8.50 illustrates the situation further. At the moment when the robot system decides to switch its localisation reference frame from place S_1 to place S_2, the robot is at point P^{S_1}, the robot pose P relative to the reference frame defined by S_1. The problem is to calculate a value for P^{S_2}, the robot pose P in relation to the reference frame defined by place S_2. The new pose can be calculated as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix}_{P^{S_2}} = \begin{pmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_{P^{S_1}} + \begin{pmatrix} x \\ y \end{pmatrix}_{S_2^{S_1}} \qquad (8.35)$$

where α is the difference in angle between the two coordinate systems defined by S_1

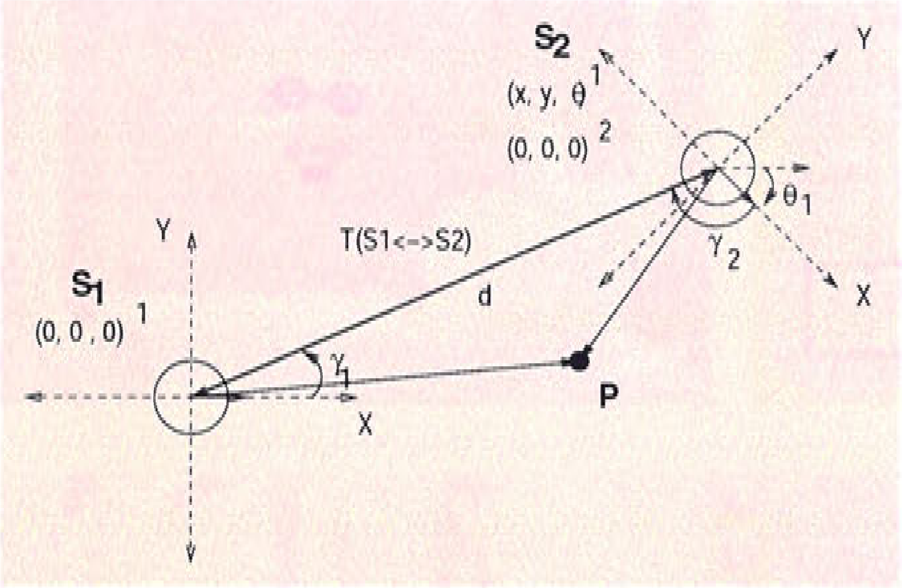


Figure 8.50: The geometric relationship of passing a position estimate between two connected places in the topological map.

and S_2:

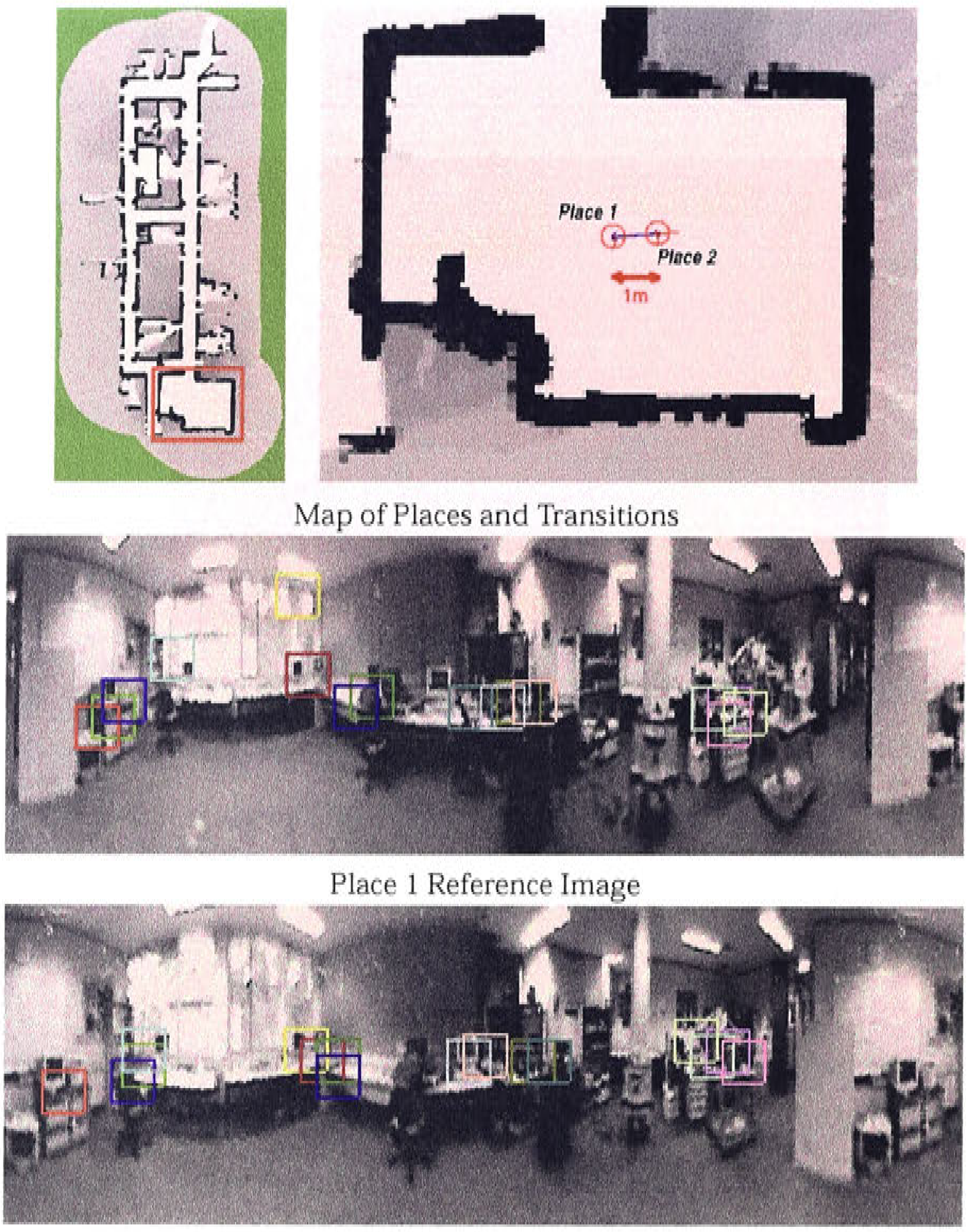
$$\alpha = \gamma_1 - (\gamma_2 - \pi/2) \tag{8.36}$$

remembering $T_{S_1 \to S_2} = (d_1, \gamma_1)$ and $T_{S_2 \to S_1} = (d_2, \gamma_2)$ as presented in Chapter 7. This allows S_1 to be defined in terms relative to S_2 as:

$$\begin{pmatrix} x \\ y \end{pmatrix}_{S_2^{S_1}} = \begin{pmatrix} d_2 cos\gamma_2 \\ d_2 sin\gamma_2 \end{pmatrix} \tag{8.37}$$

In this fashion a position estimate relative to one place can be passed to another place using the spatial relationship defined by the transition information.

Once a local position estimate has been transformed to be relative to the place towards which the robot is travelling, the particles in the particle filter can be distributed around this estimate. By spreading the particle distribution about the estimated position any errors contained within the previous estimate relative to the first place or in the transition information are overcome. The robot proceeds with local position tracking relative to to the new place, until another place transition has been detected.



Map of Places and Transitions

Place 1 Reference Image

Place 2 Reference Image

Figure 8.51: The topological map used in the position passing experiment. The transitions between the two places are drawn in blue. The two place's landmarks are shown in images captured from the place's reference positions.

A Real World Example of Local Position Passing

The passing of local position estimates between places has been implemented on a real world mobile robot. An experiment which displays the robot's ability to track its position between two places is presented.

Figure 8.51 shows a topological map with two places connected by known transitions. Panoramic images containing the reference landmark set for each place in the topo-

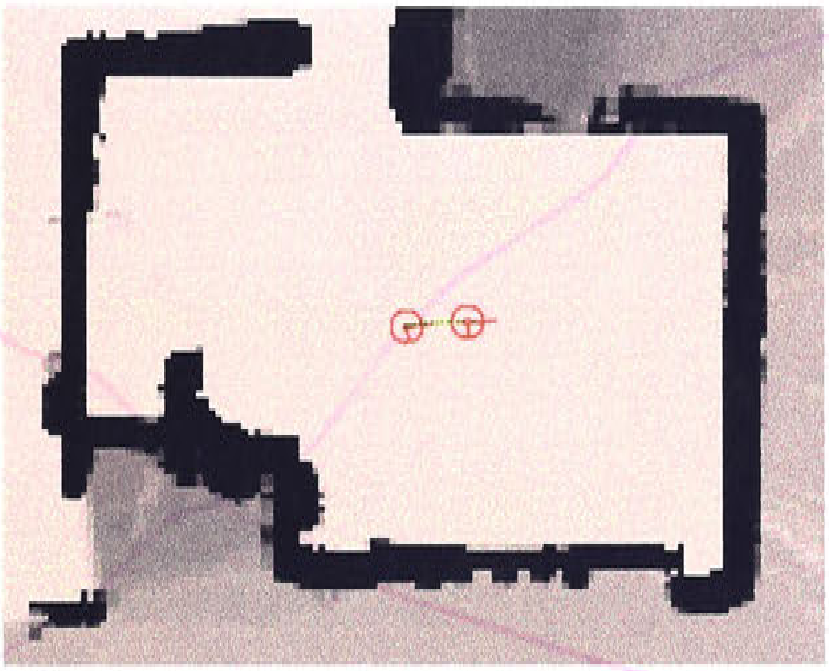


Figure 8.52: A robot path from place 1 to place 2 in topological map.

logical map are also shown.

In this experiment the mobile robot executed a path from place 1 to place 2. The path followed is shown in Figure 8.52. In traversing this path, the robot performs landmark recognition on the landmark sets of each of the two places. At the start of the path the landmark set for place 1 is recognised more strongly, whereas the landmarks of place 2 perform better towards the end of the path, as the robot approaches place 2. The LRP for both places over the complete robot path is presented in Figure 8.53. Due to this difference in tracking performance, it makes sense to estimate the local position of the robot using the landmark set which produces the best recognition results. The goal of this experiment was to successfully track the robot's position over the complete path using local position estimates relative to each place.

Figure 8.54 shows the state of the system in various locations along the example path. In these images, the place with the higher Landmark Recognition Performance (LRP) is drawn as a red circle, and the local position is subsequently estimated relative to that place. Initially the landmarks describing place 1 have the higher LRP and thus the robot position is at first estimated relative to place 1's reference position. The estimated path taken relative to place 1 is shown by the black dots in part a) of Figure 8.54, while the particle filter particle set distribution is plotted by the green dots. At the point where the LRP performance of place 2 rises above that of place 1's, the position estimate relative to place 1 is transformed to be relative to place 2 using the transition information, and a particle filter for local position estimation in place two

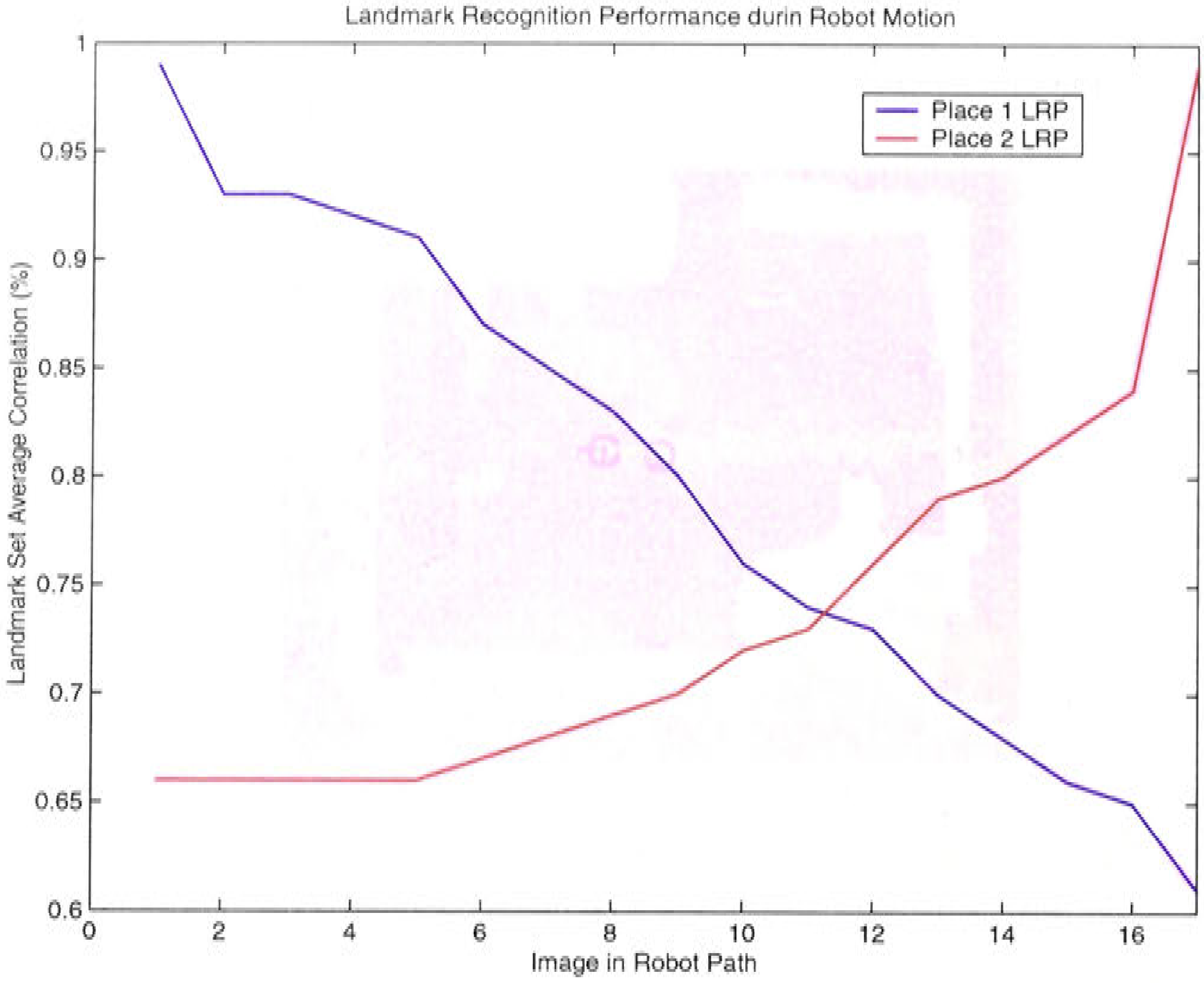
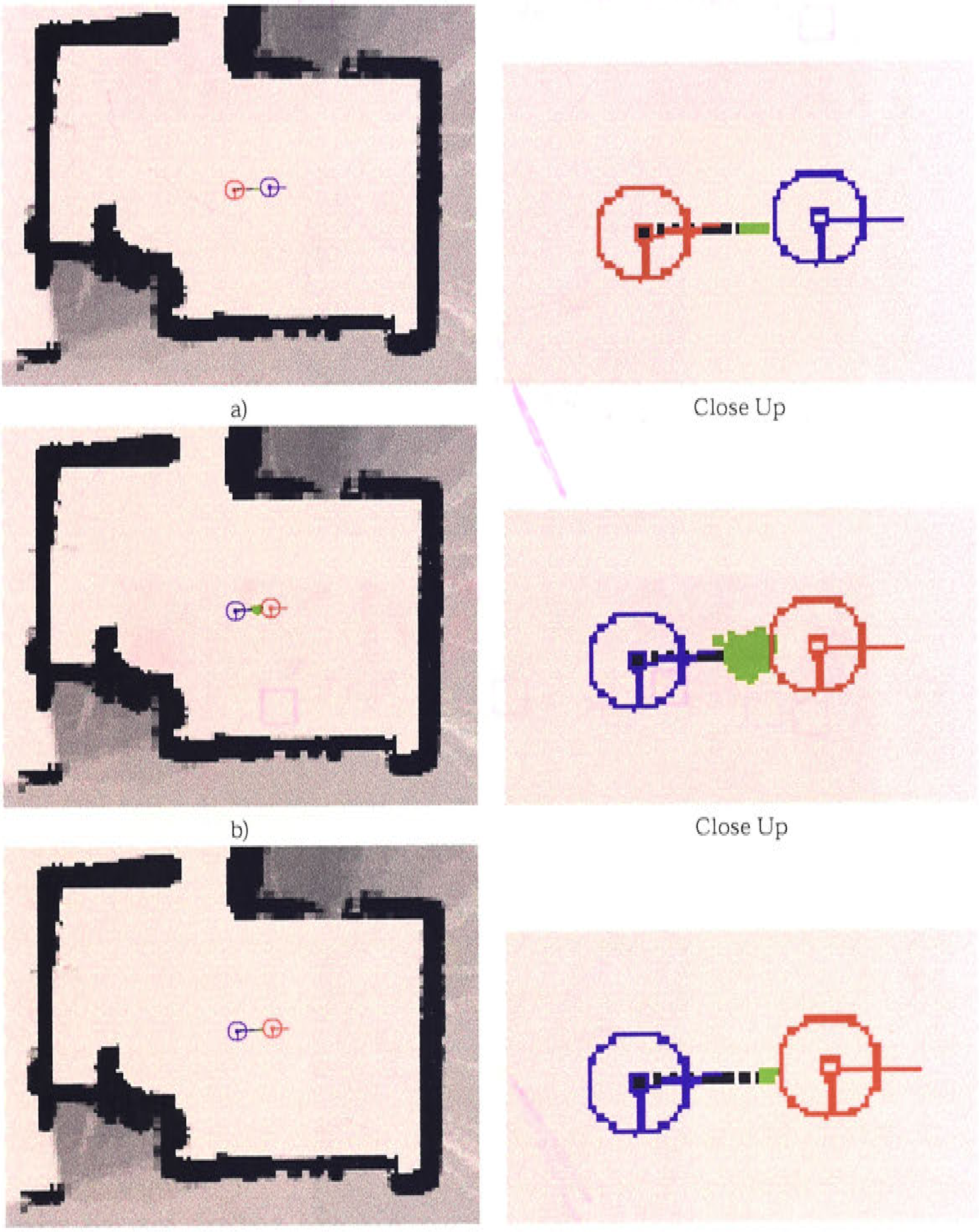


Figure 8.53: The Landmark Recognition Performance of the two places from images captured during the traversal of the path.

is initialised as shown in part b) of the figure. The green blob is the particle set of poses relative to place 2 distributed about the transformed position estimate. Part c) of the figure shows the state of the system at the next step, where the particle set has condensed down to a correct estimate of the robot's local position relative to place 2.

Figure 8.55 shows the input to the particle filters from the panoramic sensor and landmark tracking system prior to and immediately after the passing of the position estimate between the two places. The top two images in the figure show the landmark tracking results of the landmarks belonging to place 1, and the observations they form when used in the sensor model. The bottom two images of the figure show that after the system has detected the switch to place 2, the system is now using landmark observations derived from the landmark tracking results of place 2's landmark set.

Using this method, position tracking was maintained over the entire path that the robot system traversed. The complete estimated path is shown in Figure 8.56.

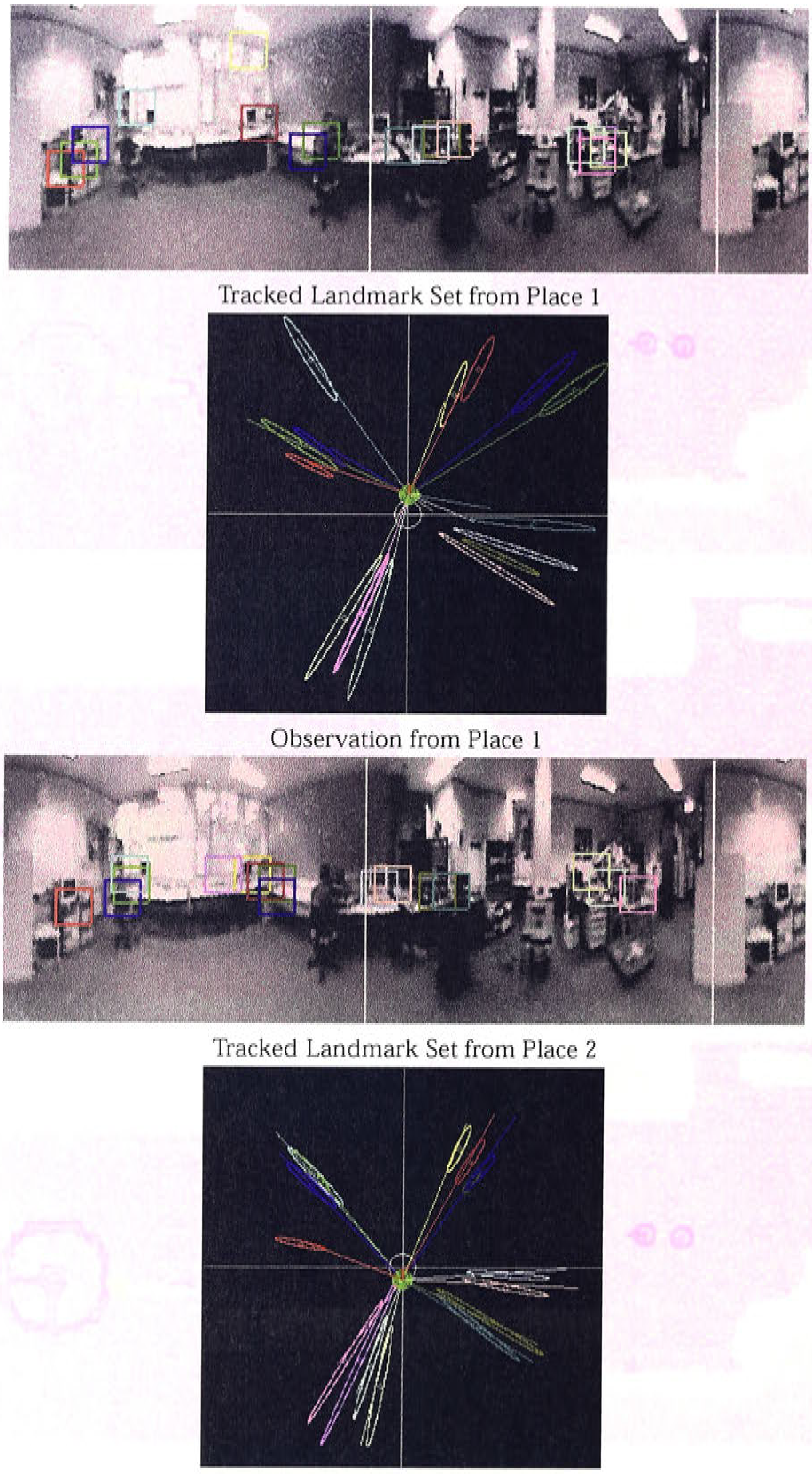


a) Close Up

b) Close Up

c) Close Up

Figure 8.54: Position passing between places in the topological map. Part a) shows the estimated path the robot travelled relative to place 1. At this stage the LRP performance for place 1 is higher than for place 2. Part b) shows the moment after the LRP for place 2 rose above that of place 1 and the particle set is now redistributed around the passed position estimate relative to place 2. Place c) shows a few steps later, the particle distribution condensing around the correct estimate.



Tracked Landmark Set from Place 1

Observation from Place 1

Tracked Landmark Set from Place 2

Observation from Place 2.

Figure 8.55: Tracked landmarks and observations of the two landmark sets from the two places at the image in the sequence when the need for position passing occurs. The observations are drawn from the estimated robot position.

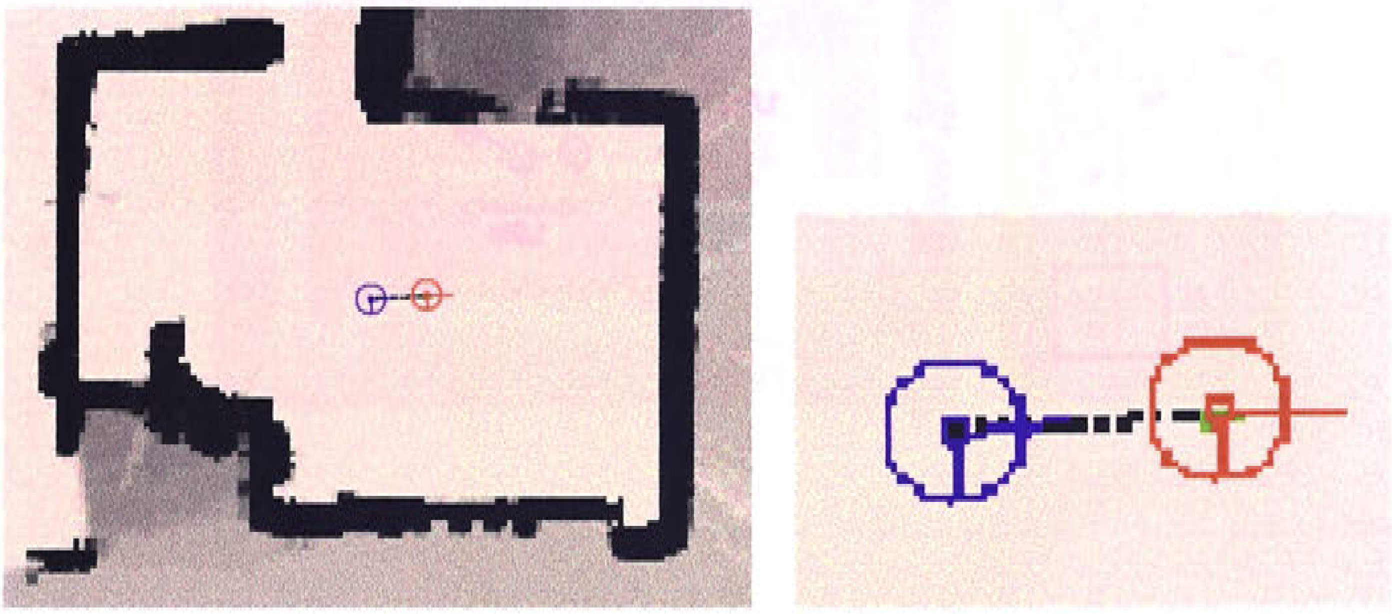


Figure 8.56: The complete estimated path between the two places in the topological map. The path contains estimates relative to both place 1 and place 2.

Position Tracking in a Small Topological Map

The local positioning experiments that have been presented have all been conducted with very short movements. These experiments demonstrated the potential accuracy of the local positioning system. However for such short movements odometric measurements by themselves can provide similar accuracy if the initial position is known. To test the ability of our system to maintain an accurate local position estimate in the face of drifting odometry measurements, it is necessary to perform experiments with longer and cyclical paths. An experiment of this scope is detailed and the systems position tracking estimation results are compared to those produced using only odometric measurements.

In this experiment the measure of robot position ground truth has been provided by a laser range sensor and metric map based localisation system (Thrun, Beetz, Bennewitz, Burgard, Cremers, Dellaert, Fox, ahnel, Rosenberg, Roy, Schulte and Schulz, 2000). This "ground truth" has a granuality of $10cm$ and it has been observed to produce erroneous measures of up to $20cm$, although almost all measurements are within the $10cm$ limit. Interpretation of our system's local position estimation performance should therefore allow for the possibility of errors in the ground truth measure.

A topological map containing a cycle was captured in a large room. The locations at which each of the five places were learnt and the transitions between the places are

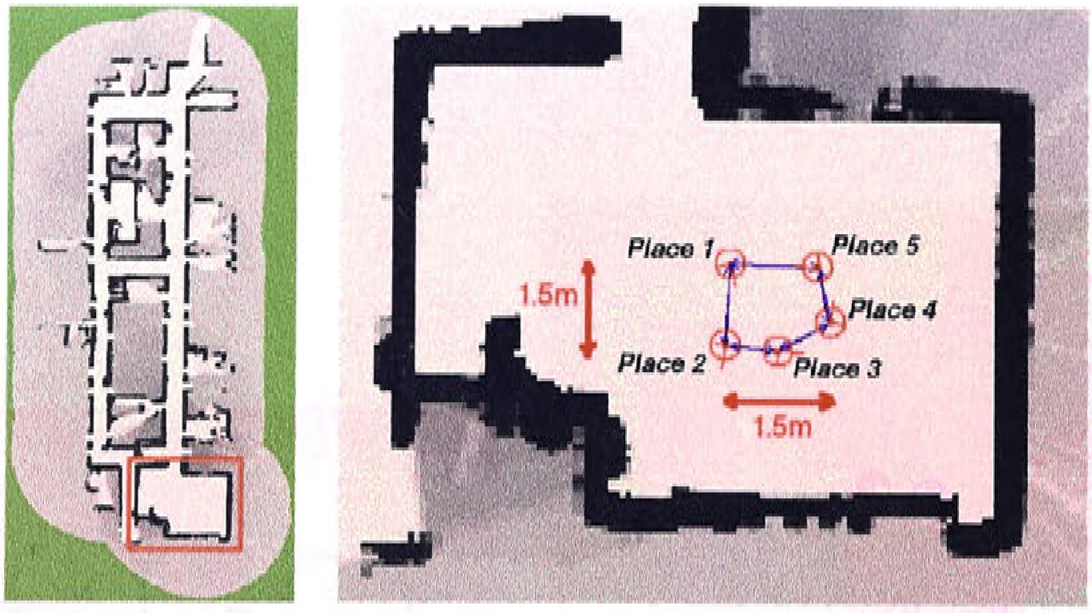


Figure 8.57: A small topological map of five places containing a cycle used for the position tracking experiment.

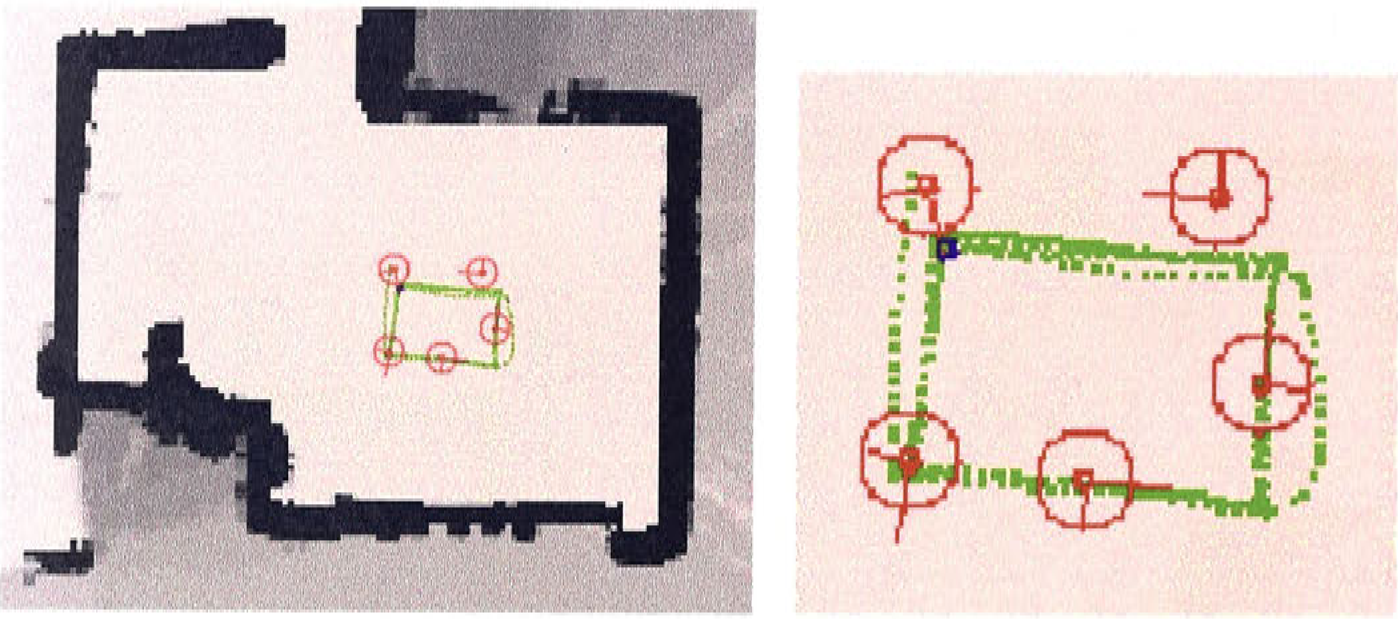


Figure 8.58: The path travelled by the robot in the position tracking experiment. The robot travels around the map four times.

shown in Figure 8.57.

After the map was learnt, the robot then traversed a cyclical path visiting each place in the topological map numerous times. The path travelled originated from place 1 in the map and followed the route shown in Figure 8.58. The displayed path completed the full circuit described by the topological map four times, travelling approximately $25m$ in total. During path execution the robot continuously recorded odometric information and panoramic images. Over the extent of the path 4100 images and odometric measurements were logged. The final position of the robot after the path was complete

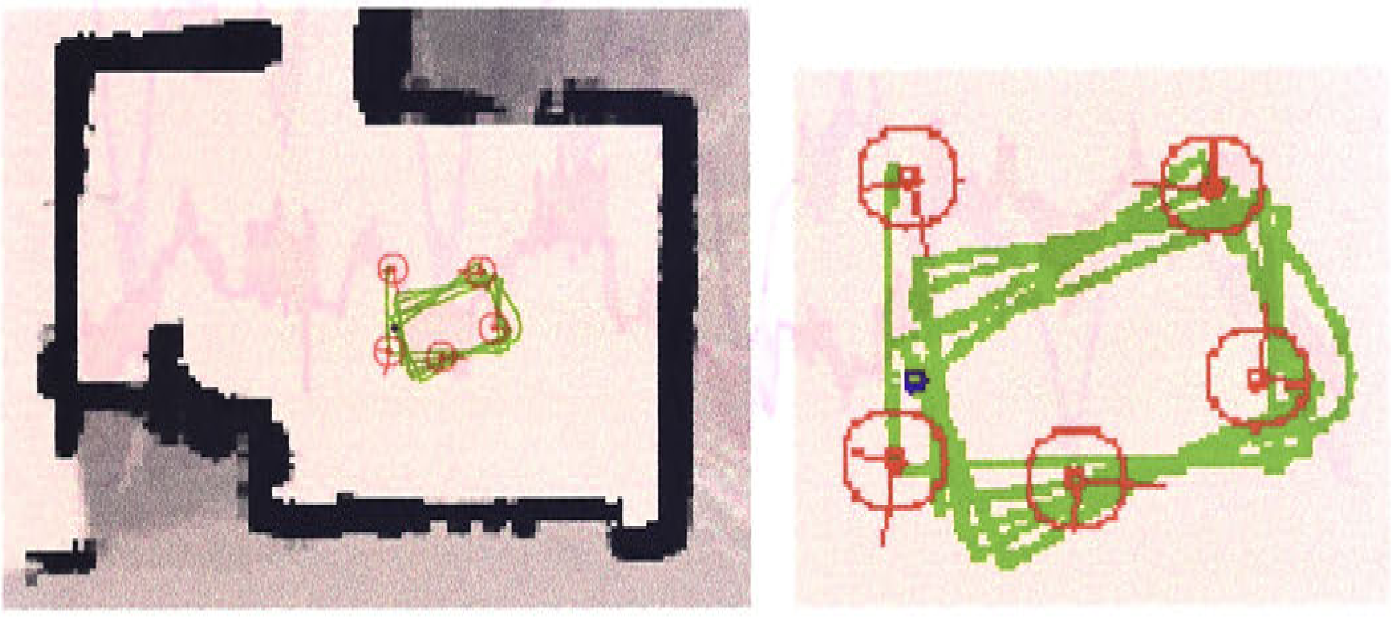


Figure 8.59: The path travelled by the robot according to odometric information.

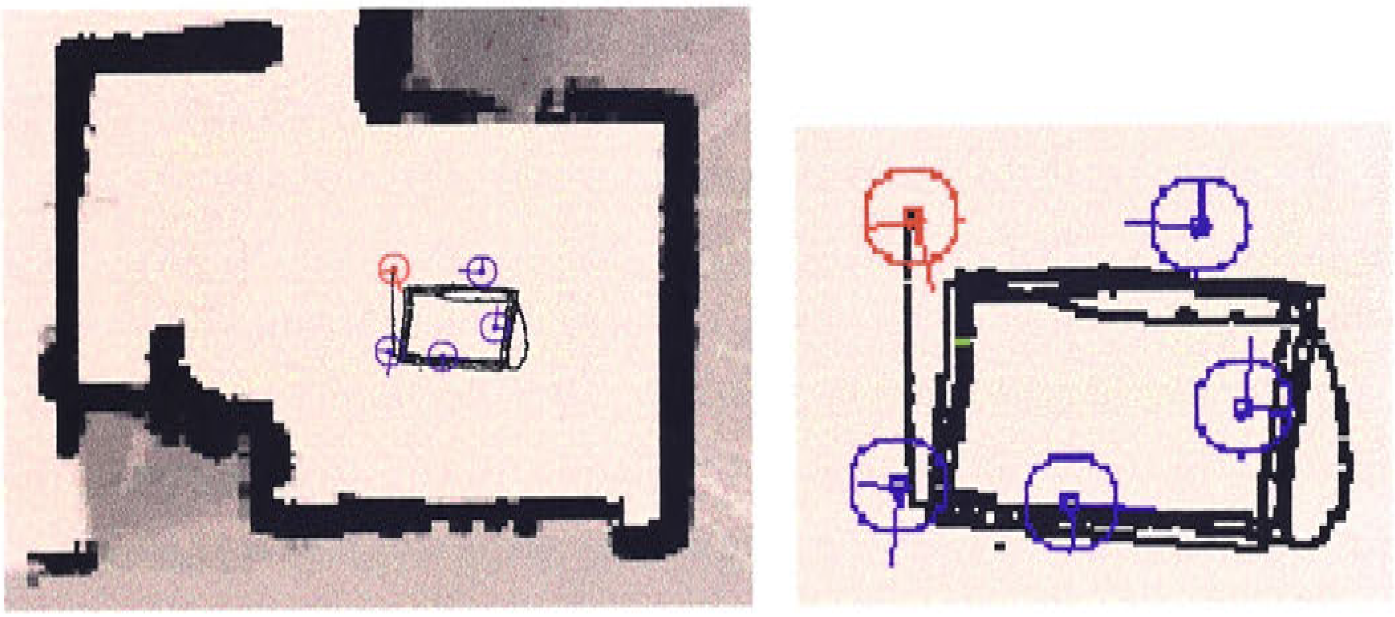


Figure 8.60: The estimated path travelled by the robot. The position estimation process used both odometric and panoramic vision sensor data.

is shown near Place 1 by the small blue circle.

The odometry measurements captured over the path are plotted in Figure 8.59 relative to the initial starting position at place 1. The odometry initially follows the travelled path quite closely but eventually the accumulated error in the odometry measurements causes the estimated path to skew significantly from the true path. The odometrically estimated final position of the robot is shown by the blue circle. The final position is about $50cm$ away from that indicated by the true path plot. This result clearly demonstrates the need for additional sensor data to correct the accumulating odometric error.

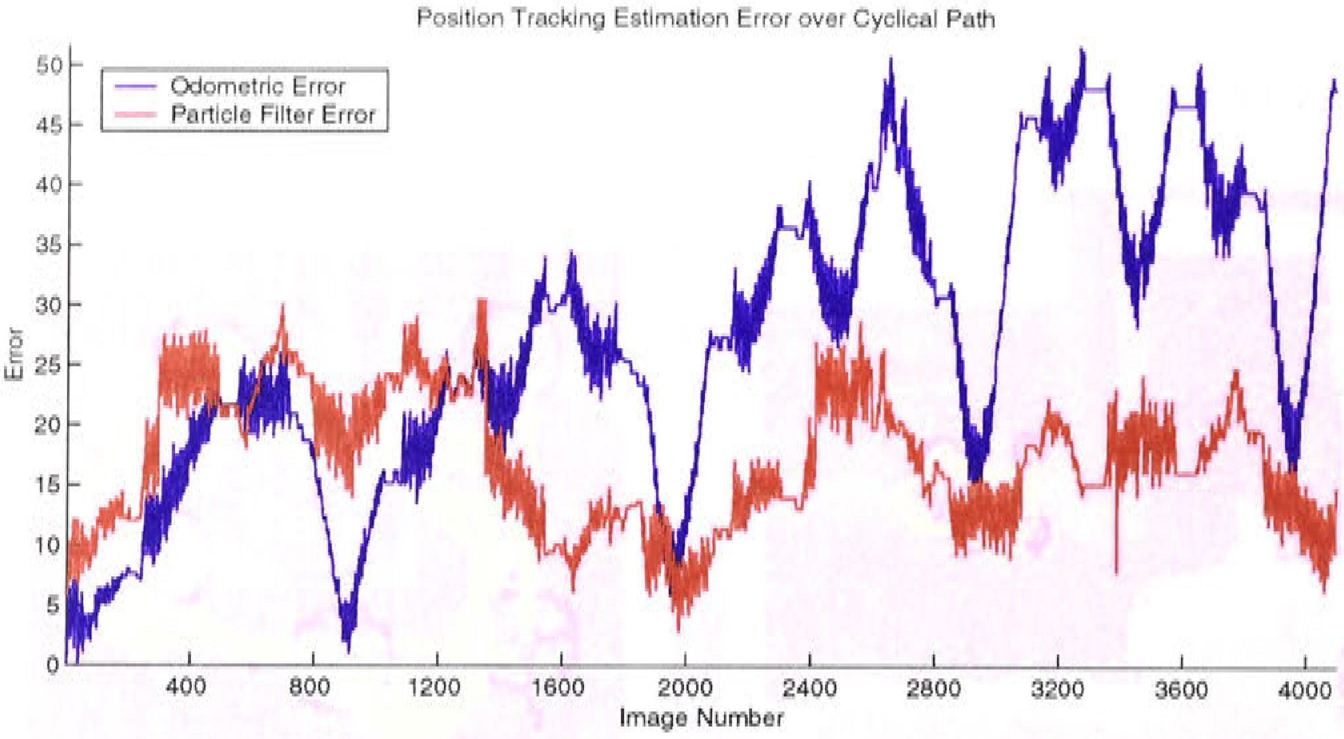


Figure 8.61: The error in position estimation from the true path from the estimated and odometric paths.

The results of performing position tracking over the cyclical path are shown in Figure 8.61. The black dots in this figure show the estimated position of the robot at each point along the path. The green dot shows the position estimate of the robot at the end of the path. After the entire $25m$ path is complete the position estimate produced by the current system is approximately $15cm$ from the ground truth location. This can be compared to the error of approximately $50cm$ for the final odometric estimate. The mean error of the our system's local position estimation over the path is $17.06cm$.

Although the estimated path is noisy it does not suffer from the drift seen in the odometric results. If the robot was to continue traversing the cycle in the topological map, the odometric estimate would continue to grow, whereas as the error in the estimate produced by our system would be maintained at the same level. The noise in the estimate is introduced by poorly tracked landmarks and the inaccuracies of passing position estimates between places. It can be seen however that the error in the local position estimate is bounded, with certain sections of the topological map, especially around place 2 providing very accurate position estimates. At these places, the system uses accurate landmark tracking information to correct and constrain the distribution of the particle set representing the robot's position estimate.

The ground truth measure is subject to an error of approximately $10cm$ itself, so it is difficult to interpret the possible position tracking accuracy of the current system, especially given that the reported local position error remains relatively constant over the entire path. The local position estimation results over the TBL path suggest that an accuracy of less than $5cm$ is possible within an individual place. Of course as the robot moves between places, noise is introduced due to moving away from the reference positions and switching places and this accuracy is lost. Our system *can* achieve position tracking accuracy of within $15 - 20cm$ although much more accurate estimates can be achieved when the robot moves close to the reference positions of places in the map.

8.3 Summary

This chapter has described the application of the proposed multi-level spatial representation for mobile robots to the problem of local position estimation. Methods for performing local position estimation were presented and experimentation validating our approach was performed. In particular the following results were achieved:

- *Particle Filter Approach to Local Position Estimation:* The use of the particle filter state estimation approach to solve the mobile robot localisation problem was introduced. In comparison to other systems, this system attempts to use this approach with a topological representation and visual landmarks.

- *Geometric Sensor Model for Panoramic Vision Sensor:* Novel sensor models for particle probability evaluation of varying geometric complexity were developed and evaluated. In particular the geometric models captured the sensor noise and geometric properties specific to the use of visual landmarks and the panoramic sensor. It was found that modelling the landmark uncertainty as an ellipsoid and the current observation as a ray produced the most accurate probability density functions while maintaining real time constraints.

- *Accurate Local Position Estimation in a Topological Representation:* Using the ellipsoid-line intersection sensor model local position estimation over a path within a given learnt place was achieved with an error of less than $4cm$. Experiments confirming the systems robustness in the presence of occlusion and data mis-association were presented.

- *Position Tracking:* A method of passing position estimates between places to achieve local position tracking along a path was introduced. A position tracking experiment throughout a topological map demonstrated the systems ability to overcome odometric drift and produced estimation results within a $15 - 20cm$ error of the metric map based "ground truth" measure.

The particle filter approach to mobile robot localisation allows for the approximation of arbitrary robot position probability density functions. This is important in the case of the local position estimation in the current system as the noisy sensor information can produce non-Gaussian distributions. When applied to the restricted area surrounding the reference position from which a place was learnt, it can provide very accurate position estimates despite the noisy and often inconsistent information provided by landmark observations. The $4cm$ accuracy in the local position estimate is quite remarkable given the low resolution of the vision sensor and the loose coupling of odometric and visual sensor data. The accuracy is achieved through the discriminatory ability of the ellipsoid-line intersection sensor model and the particle filter's propensity for condensing about the correct estimate. The chosen sensor model balances the computation cost of particle evaluation and the ability to model the process noise in the system.

Position estimation between places was less accurate but was robust to errors due to odometric drift. The accuracy of position estimation within places necessarily is dependent on the distance from the reference position as the more accurate measure in a landmark's position estimate is its angle from the reference position rather than the depth measurement. This results in a range of achievable position accuracy as the robot moves between places in the topological map. It should be assumed that a robot that needs to perform very accurate measurements near a particular location in an environment, would learn a place of that location, thus maximising the potential position estimate accuracy. Although the initial location of landmarks within a panoramic image are computationally expensive, position tracking can be maintained close to real time by tracking landmarks through the image sequence.

The first goal of this research was to realise a mobile robot system which constructed a topological internal representation of its environment which allowed for accurate position estimation on par with contemporary metric map based systems. Given the local position estimate accuracy which is achievable near the reference position of learnt places, it can be concluded that this goal has been achieved. It is easy to imagine that improvements to the resolution of the panoramic sensor and an introduction of a tighter synchronisation between the sensor measurements, that this accuracy could be improved.

A minor draw back to local position estimation in terms of accuracy is the basic nature of the visual landmarks. The nature of visual landmarks themselves mean that they can capture the appearance of objects at different depths of the environment within a single template. This can lead to inaccurate initial depth estimates and noisy observations during local position estimation. A landmark identification system which selected regions of objects at a consistent depth would produce much more accurate results.

This chapter has described methods for performing the task of local position estimation. Experimental results have been presented which confirm the systems ability to perform accurate local position estimation in a topological map representation, using a noisy visual sensor.

Chapter 9

Global Localisation

Global localisation is the process of locating a robot in an internal map from an unknown start position. In our research this means localising to a particular learnt place in the topological map. In this chapter the global localisation problem is investigated and a solution is proposed involving the use of the low-level landmark based place representation. To globally localise the robot system must match the current panoramic view with the landmark sets of all places in the topological map. The results of this matching can be used to discriminate between places on the basis of recognition performance. The method for this solution is described and experimental results on place discrimination are presented. A high place discrimination ability reflects the uniqueness of individual places defined by sets of visual landmarks and validates the underlying low-level representation. Experiments combining the results of global localisation and local position estimation are reported. The computation costs involved in the process of global localisation using the low-level representation are expensive. The global localisation search space can be constrained by the mid-level representation of local space profiles and computational costs reduced. In attempt to achieve continuous global localisation and local position estimation, a method for the detection and recovery of local position estimation is developed. Using these methods for continuous global localisation and local position estimation, the kidnapped robot problem can be solved.

Section 9.1 briefly reviews the matching of landmark sets with panoramic images to perform place recognition. Section 9.2 reports experiments that investigate the use of sets of visual landmarks in discriminating between places in a topological map for the task of global localisation. Section 9.3 explains combining global localisation results with position estimates to track a robots position as it travels through the environment. Section 9.4 describes the use of local space profiles, the second level of representation, to restrict the global localisation search. Section 9.5 reports on the process of disambiguating between places when lower localisation estimates cannot distinguish between multiple hypotheses. Section 9.6 applies these localisation techniques to solve the kidnapped robot problem introduced in Chapter 2. Finally, Section 9.7 presents a summary and discussion about localisation in systems with multi-level representations.

9.1 Matching Places

Global localisation in the low-level representation is a matter of matching sets of landmarks, and their associated place, to the visual scene. A brute force search of landmarks throughout the entire image is undertaken for each set of landmarks, producing a measure of Landmark Recognition Performance (LRP). The robot is initially assumed to be in the place associated with the set of landmarks which have the highest average correlation in the current scene. Landmark set matching and LRP were discussed in detail in Chapter 4.

9.2 Place Discrimination

Place discrimination refers to the systems ability to discriminate between different places in the topological map when attempting to perform global localisation, with the goal of identifying the Most Likely Place (MLP) that the current visual view could have been captured from. This goal can be achieved by monitoring the LRP for all places in the topological map. An obvious method for identifying the most MLP is to nominate the place whose associated landmark set has the highest LRP measure. Stated more formally:

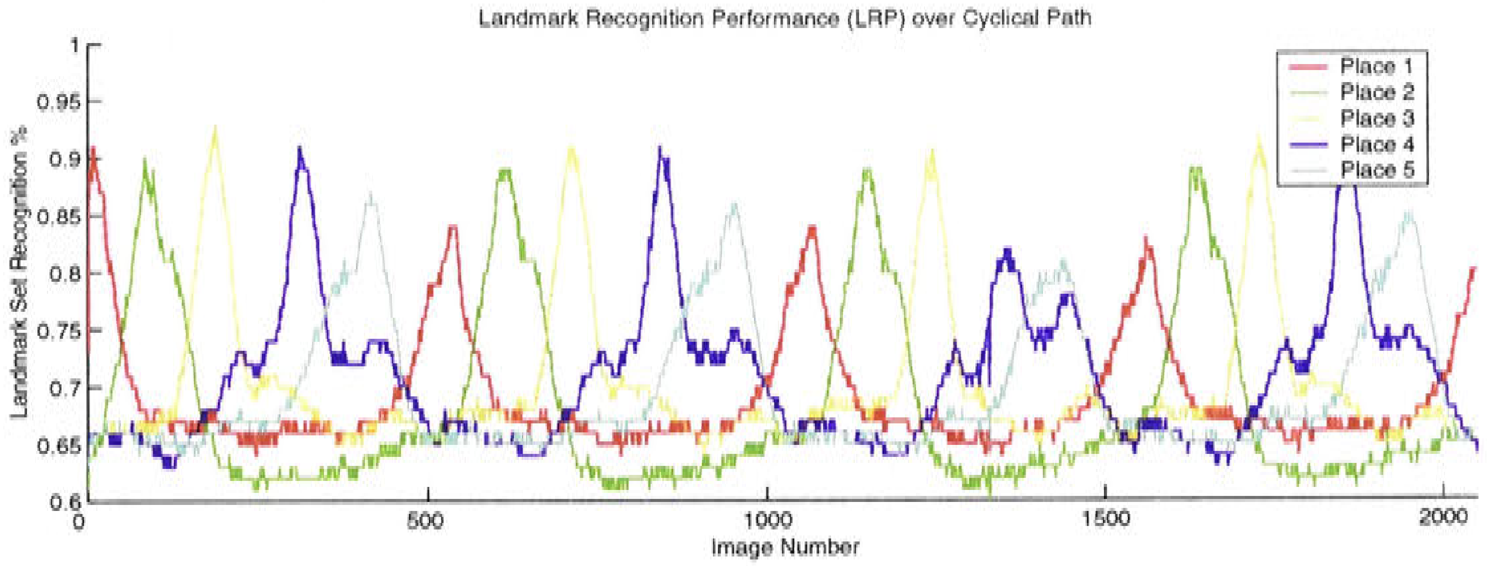


Figure 9.1: Landmark Recognition Performance (LRP) over the example path from the position tracking experiment.

$$P_t = S_i \quad | \quad (LRP(C_t, S_i) = max(LRP(C_t, S_1), \ldots, LRP(C_t, S_N))) \tag{9.1}$$

where P_t is the identified most likely place at time t, and $LRP(C_t, S_i)$ is the LRP measure gained when matching the landmark set of place S_i with that of the current panoramic image C_t.

Identifying the MLP in this fashion works well for maps which contain places where the panoramic visual scene is similar across the whole map. Figure 9.1 shows the LRP performance of all places in the topological map used in the position tracking experiment over a cyclical path for all images captured throughout the path. The LRP plot for each place produces a regular peak, corresponding to the locations along the cyclical path when the robot captured images near the reference position of each place. The place with the highest measure of LRP can be easily identified as the MLP in the topological map at any point along the example path. This result is achieved due to the "uniqueness" of the landmark set for each place despite the fact that all sets were acquired from the same general area in the environment.

This performance is due in part to the similarity of the visual scene from all places in the topological map. Given the highly unstructured and visually dense nature of panoramic images captured in the large room, the robot had plenty of interesting landmarks to select. Additionally, potential landmarks are unlikely to be repeated in such an irregular environment. By comparing landmark sets from regions of the environment that are visually similar, this experiment does not validate the method in situations where places in the map have varying background LRP levels. The importance of background LRP levels for place acquisition in topological maps was discussed in Chapter 7. To validate the place discrimination ability of our method, experiments must be performed with a topological map which covers several areas of the environment which are visually disparate and contain visually sparse scenes.

Figure 9.2 shows a large topological map which contains places located in disparate locations in the environment. The topological map contains 50 places, shown in red, connected by transitions which are shown in blue. In particular the visual scene from places acquired at locations inside the large room at the bottom of the map are very different to those located in the corridors. Figures 9.3 and 9.4 show landmark sets and visual scenes from the places marked in Figure 9.2. Note that the landmark set from corridor environments contain many landmarks which are likely to be repeated in such a visually sparse and regular environment. The entire data set of places in this topological map and paths throughout the map are included in the CD-ROM which accompanies this thesis.

A robot executed a path which traversed the topological map, during which it continually logged odometry information and panoramic images. The path is shown in Figure 9.5. The LRP of each place in the topological map was calculated for every image captured along the example path, and is plotted in Figure 9.6. The figure is obviously crowded as it contains the LRP plots of the 50 places in the topological map[1]. The interesting feature is the peaks in the graph. Each peak represents the high points of individual place's LRP over the motion path.

This result is presented in Figure 9.7 as a surface in image and map space. The LRP of each place is shown for every image along the path. The true path of the robot can be imagined as a diagonal line across the space. The plot of the LRP surface compares well with the true path. The background LRP noise can be seen by the troughs and peaks of the surface which do not lie along the true path.

Overall the result is promising, landmark sets provide a unique response for the ma-

[1] Each successive place is shown as a different colour plot. Unfortunately the plotting software only allowed 7 different colours so colours are repeated. The colour order from place 1 to place 7 is: blue, red, green, cyan, black, yellow, maroon. This is repeated every 7 places.

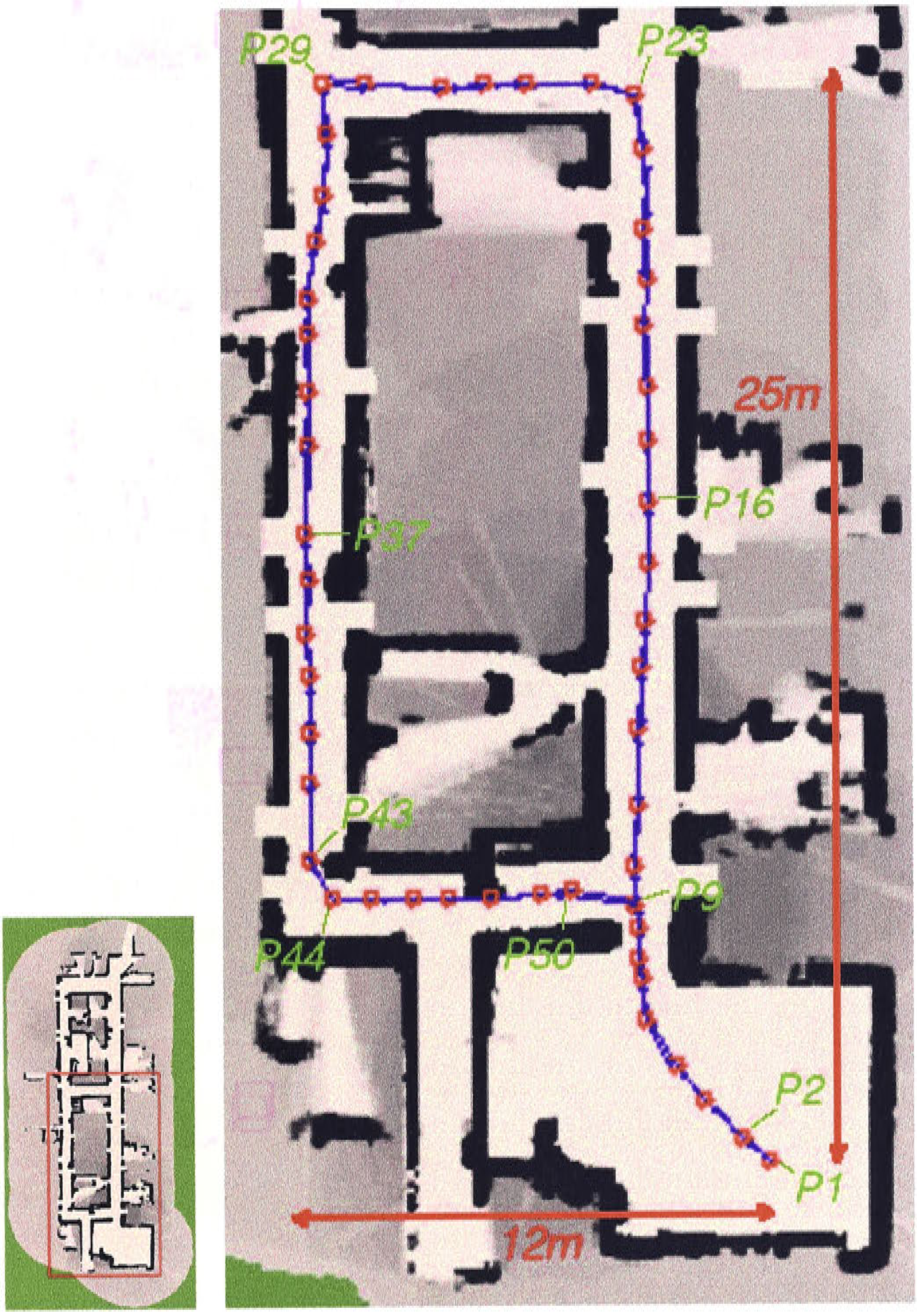


Figure 9.2: Topological map of the test environment for the place discrimination experiment. The 50 acquired places are shown in red, while the transitions between places are shown in blue.

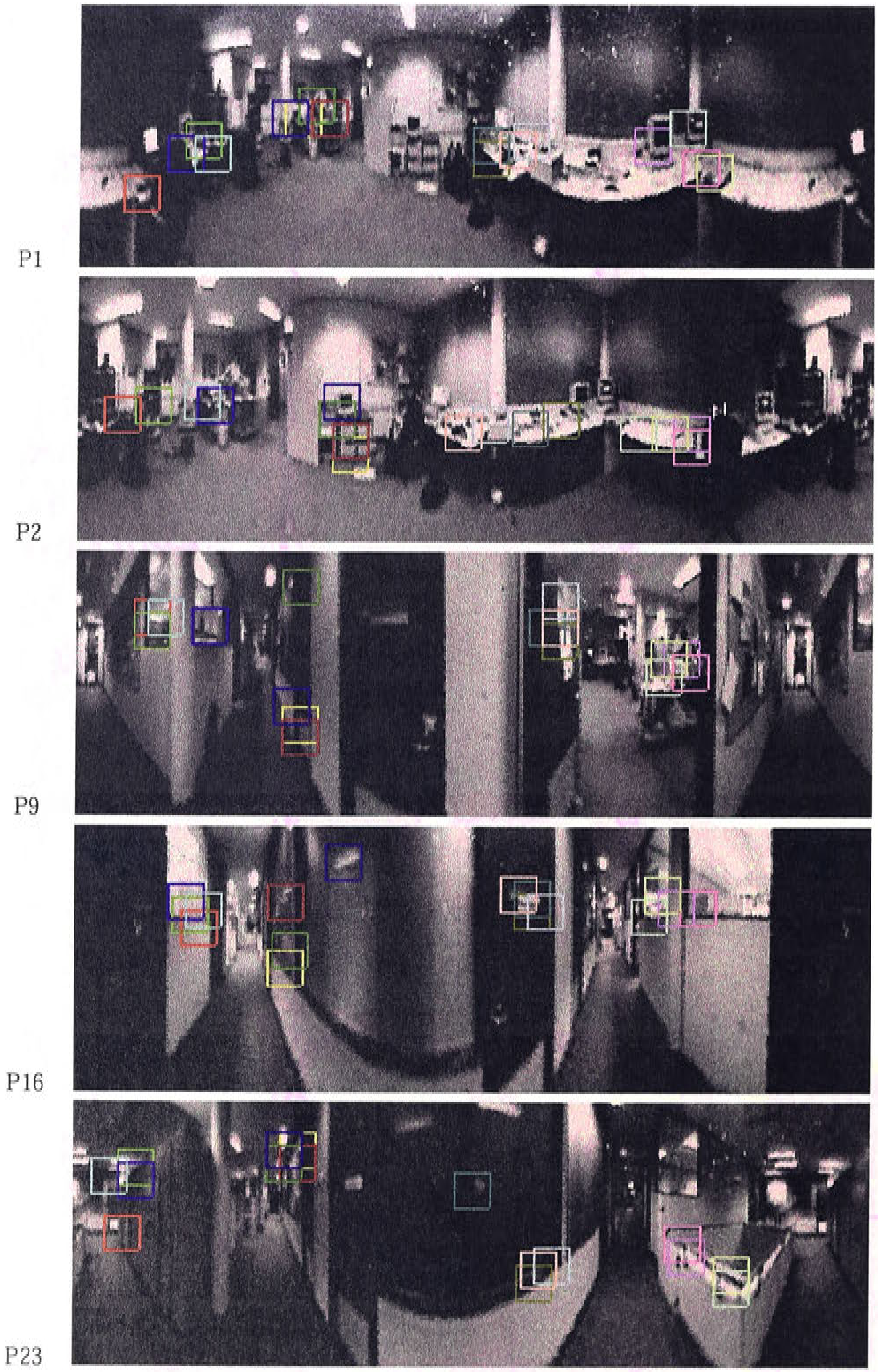


Figure 9.3: Example landmark sets and visual scenes from the reference position of places in the topological map: Places 1, 2, 9, 16, 23.

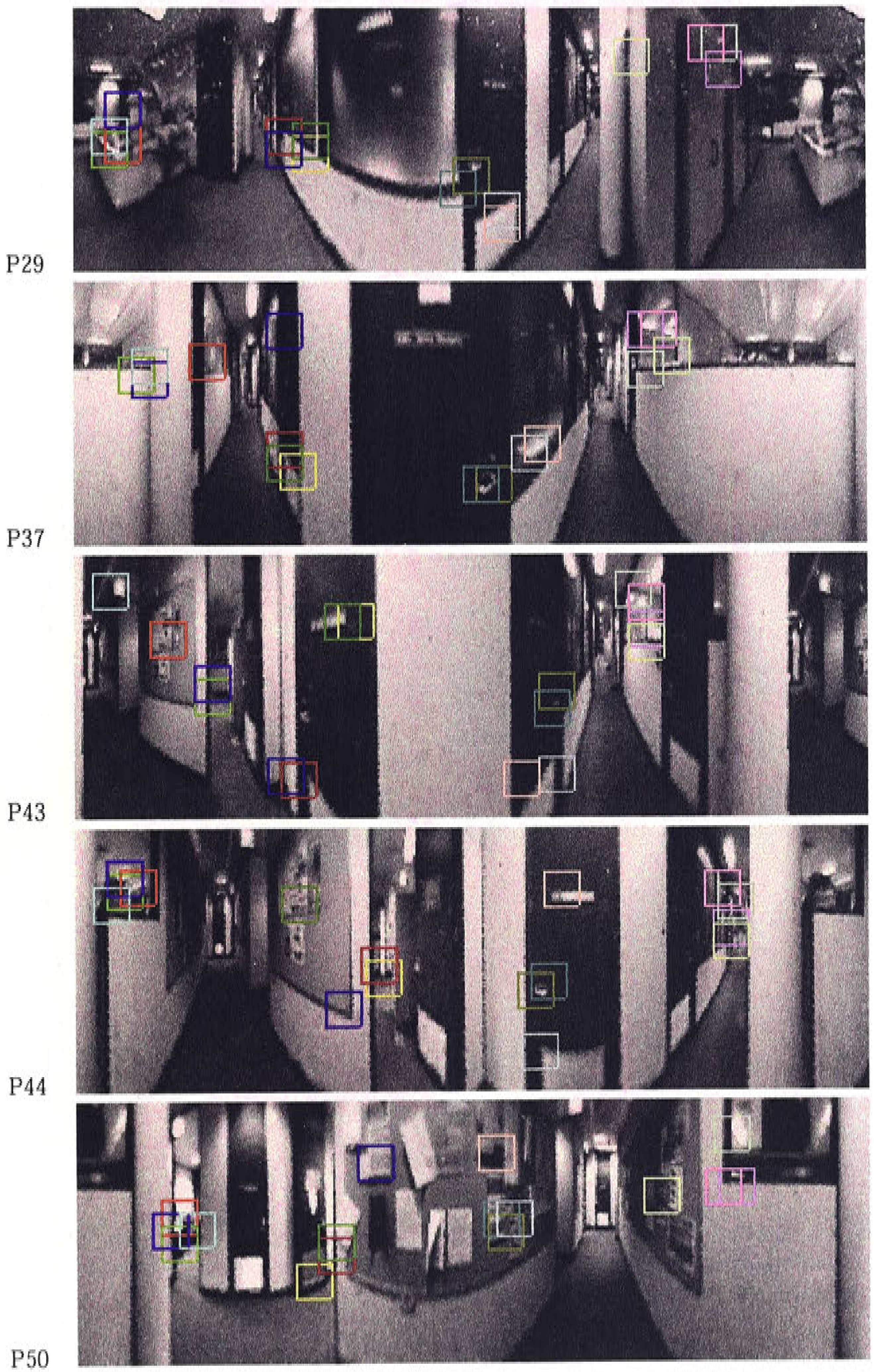


Figure 9.4: Example landmark sets and visual scenes from the reference position of places in the topological map: Places 29, 37, 43, 44, 50.

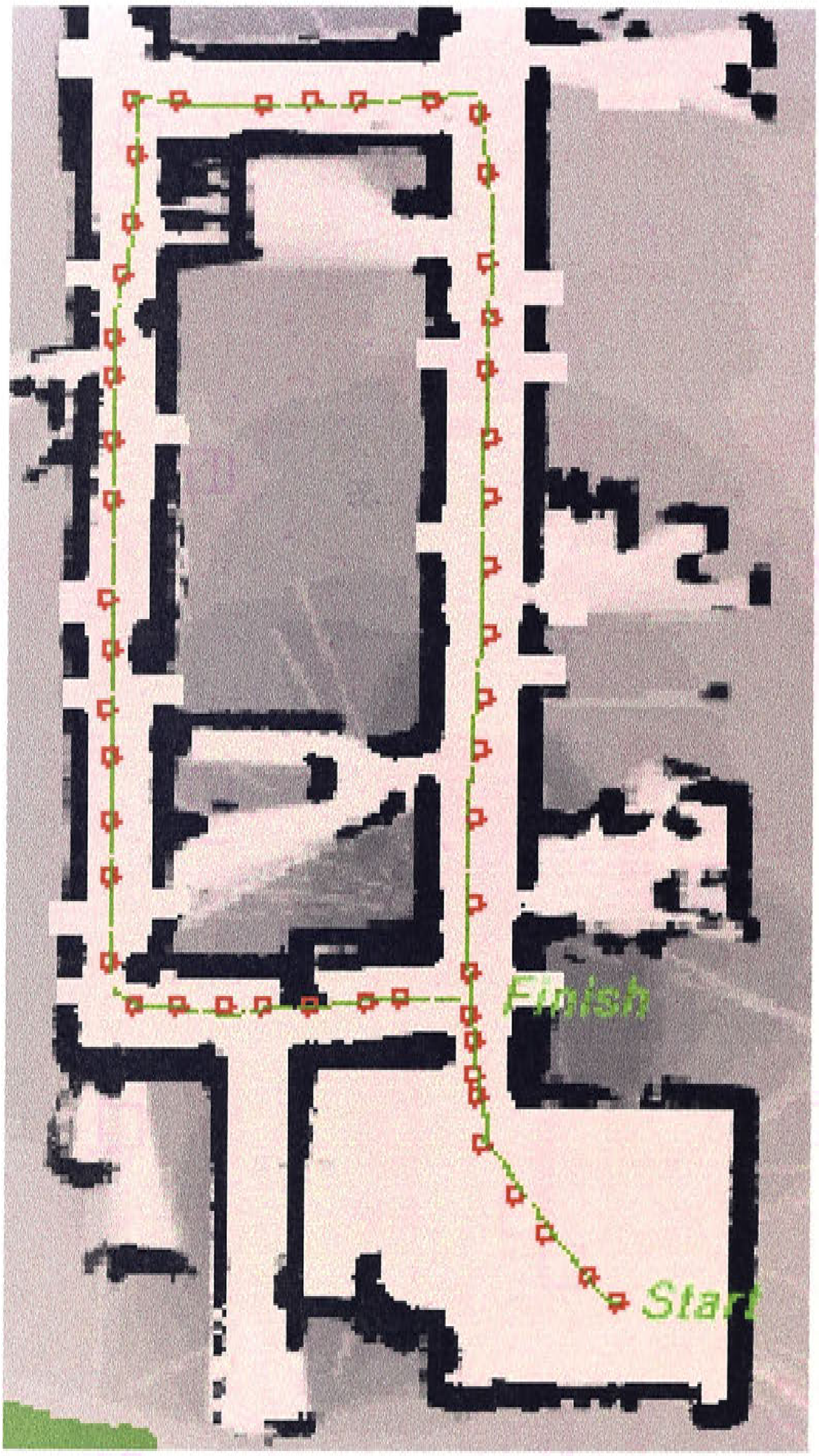


Figure 9.5: Example path which traverses the topological map used in the place discrimination experiment. The 50 places in the topological map are shown in red while the path the robot system travelled is displayed in green.

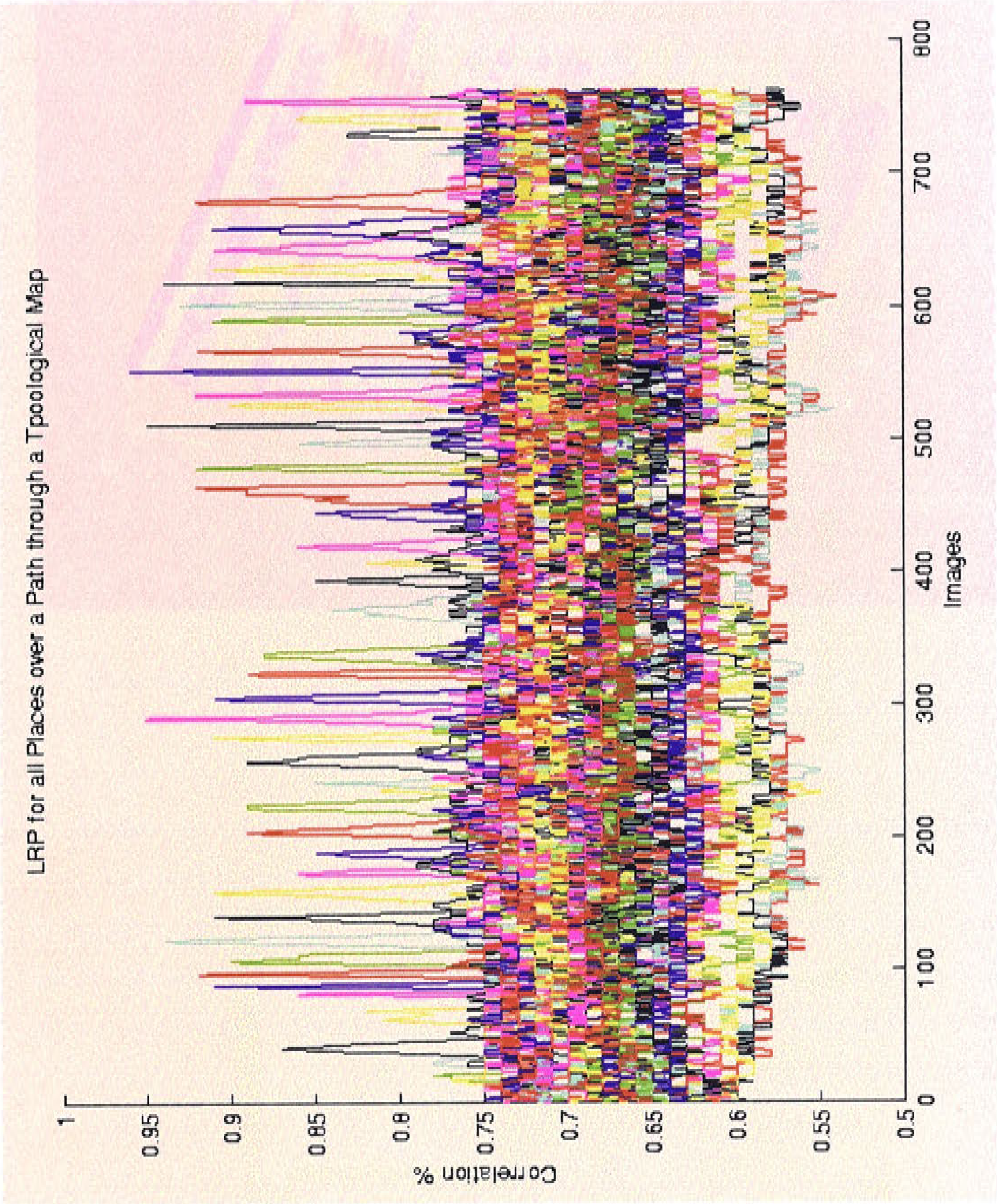


Figure 9.6: The LRP of each of the 50 places in the topological map over the example path.

jority of the path, there are only a couple of places where no peaks are apparent. We can further refine our proposed method for MLP identification.

There is one other measure which is readable available and which impacts greatly on identification of likely places which we have so far ignored. The background LRP

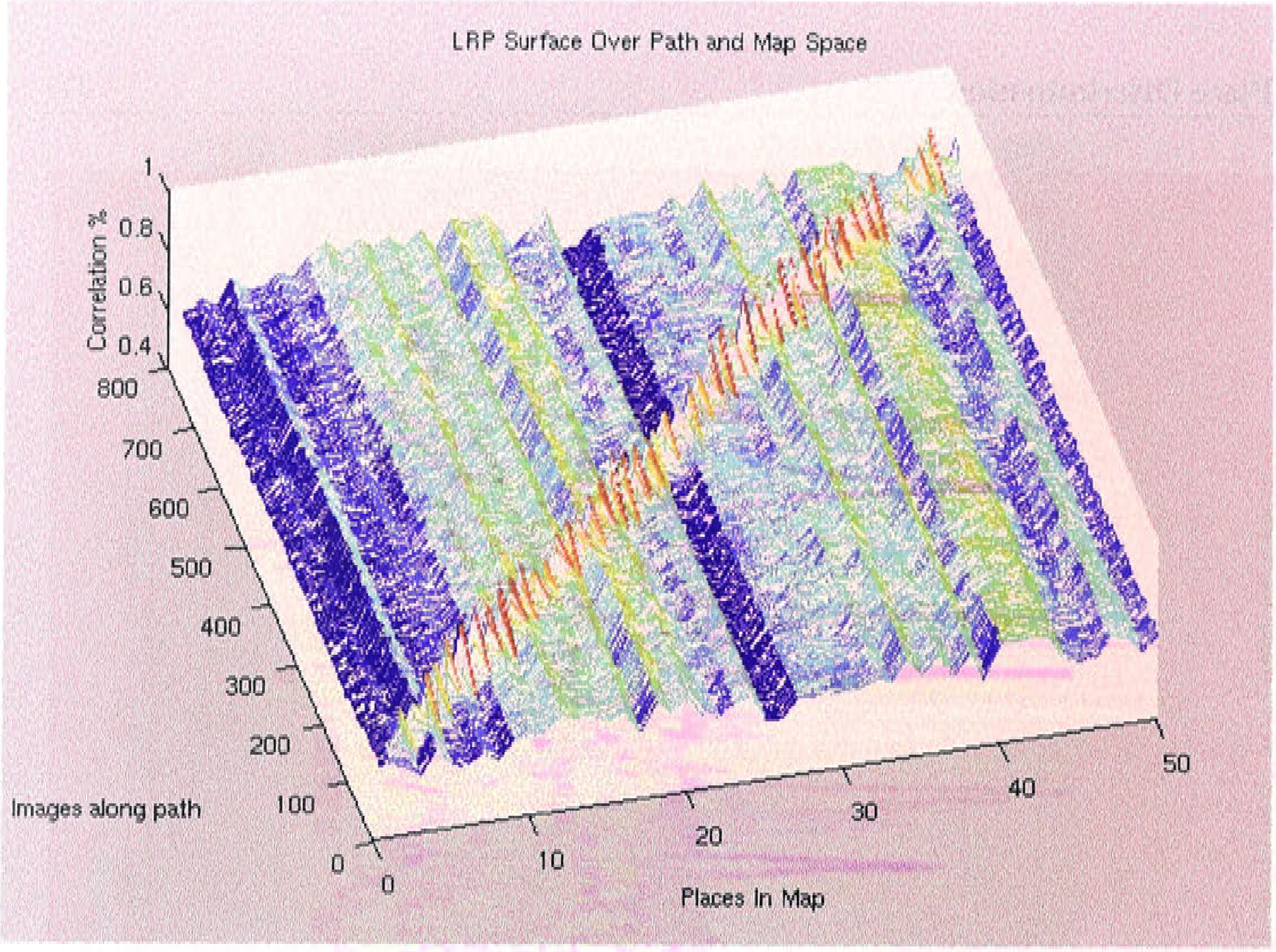


Figure 9.7: The LRP surface of each of the 50 places in the topological map over the example path.

level for each place that was detected and recorded at the time of place acquisition. If the LRP measure for a particular place is subtracted from the background LRP level for that place, then the resulting quantity gives an indication as to the strength of the LRP measure irrespective of the background levels. This is particularly helpful when comparing LRP of places with differing background levels. The formulation for identifying the MLP then becomes:

$$Q_t^i = LRP(C_t, S_i) - S_i^{bgLRP}$$

$$P_t = S_i \quad | \quad (Q_t^i = max(Q_t^1, \ldots, Q_t^N)) \tag{9.2}$$

where again P_t is the identified most likely place at time t, and $LRP(C_t, S_i)$ is the LRP measure gained when matching the landmark set of place S_i with that of the current panoramic image C_t, and now S_i^{bgLRP} is the background LRP level for place S_i, and

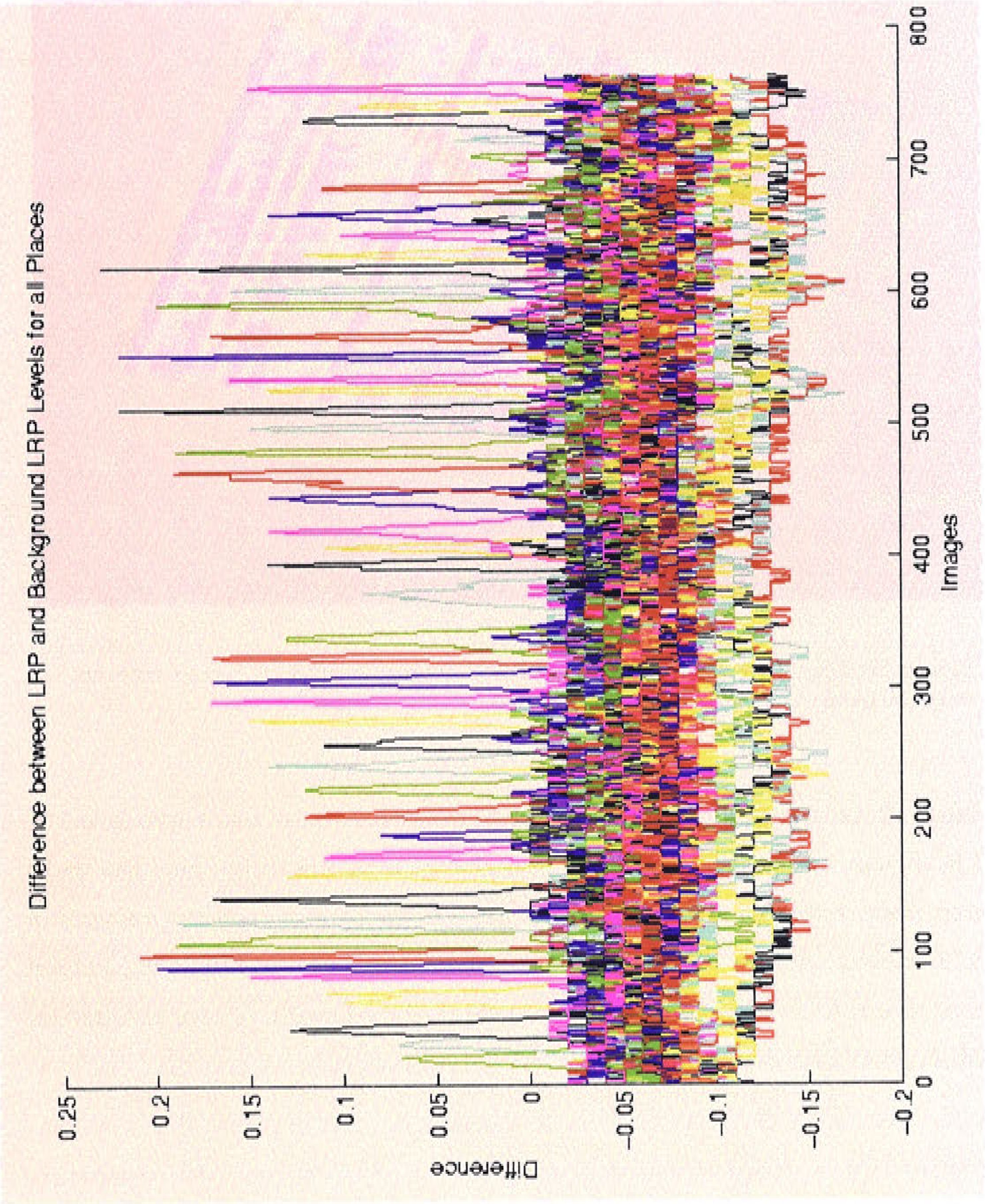


Figure 9.8: The difference between the LRP and the background LRP levels for each of the 50 places in the topological map over the example path.

Q_t^i is the difference between the measured LRP and the background level for place i at time t.

Applying this new formulation for MLP identification to the images captured during the motion path gives the results shown in Figure 9.8. This is shown as a surface over

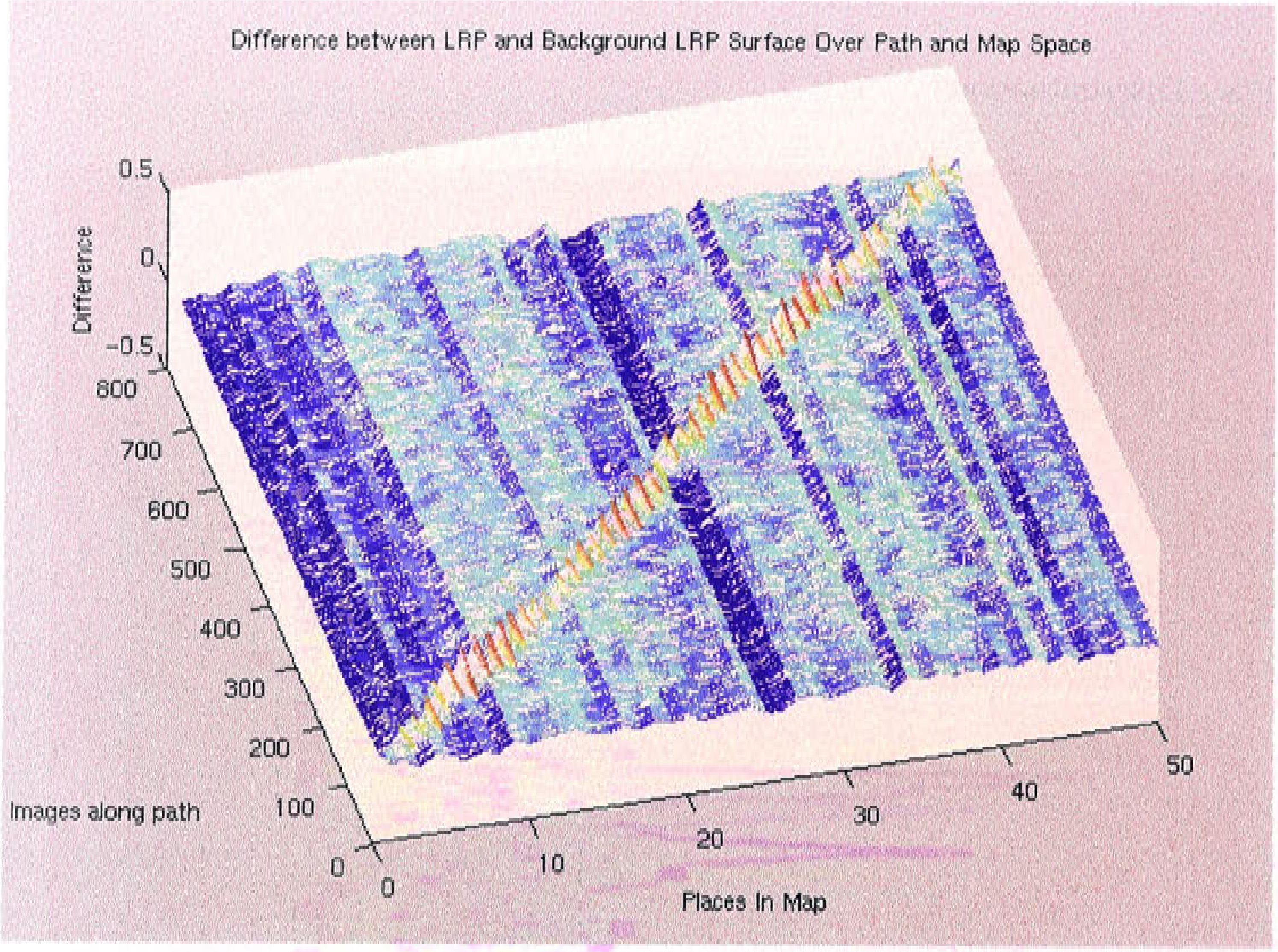


Figure 9.9: The difference between the LRP and the background LRP levels for each of the 50 places in the topological map over the example path.

the image and map space in Figure 9.9. When compared to the results produced by LRP measurements, most of the existing peaks have been strengthened and new peaks have emerged giving an overall better indication of the MLP at each stage along the path. In the surface plot, the magnitude of background peaks and troughs has been reduced. Using this method of place discrimination for exhaustive global localisation, the MLP can be correctly identified over 100% of the images in the path.

This improvement is difficult to see due to the large number of places. The improvement gained by incorporating the background LRP levels can be seen when comparing two places from locations in the map with disparate visual scenes. Figure 9.10 shows the LRP of places 2 and 43 over the example path. Place 2 represents a location in the environment which is in the large room, as shown in Figure 9.2, place 43 on the other hand, was acquired in a corridor environment. The LRP figure shows that although both places display peaks in the plots of their own performance at the correct locations along the path, the background LRP level of place 43 nearly overwhelms the peak LRP

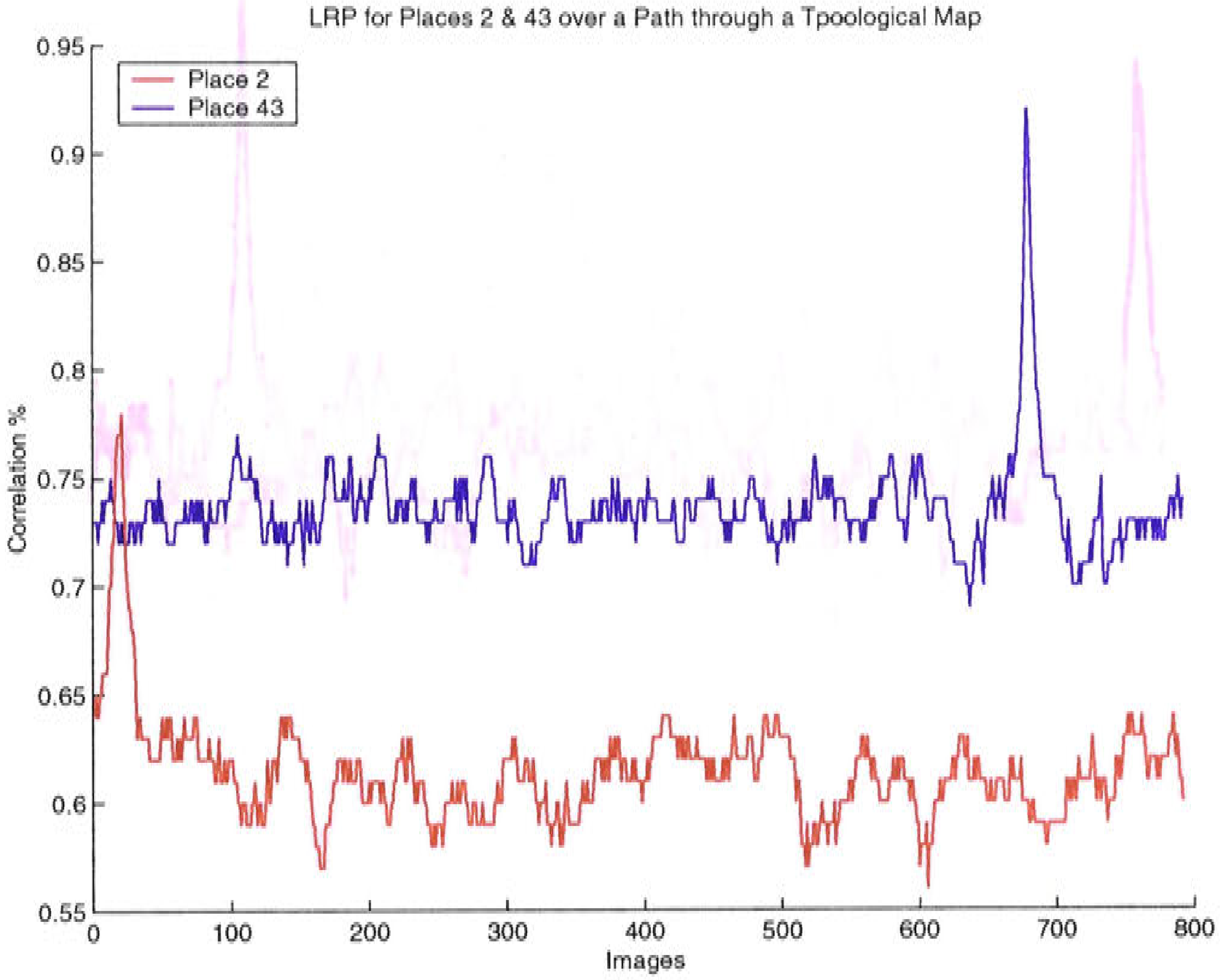


Figure 9.10: LRP of places 2 and 43 over the example path.

level produced by place 2.

In comparison, Figure 9.11 shows the difference between the LRP measurements and the background LRP levels for each of the two places. Now it can be seen that both places form a peak even when compared to the other place, the new formulation overcomes the differences in background LRP levels between the two places.

Global localisation can be achieved throughout the topological map by matching the current panoramic view with the landmark sets from all places in the topological map. The uniqueness of individual landmark sets throughout the entire topological map in conjunction with knowledge about the background recognition levels can successfully identify the MLP from which a robot makes an observation of the environment. The uniqueness of the visual landmark representation can be demonstrated by matching each places' landmark set with the panoramic image captured from the reference position of all other places in the map. Figures 9.12 and 9.13 show that only when matching a reference landmark set with the place's reference image is a high correlation result achieved.

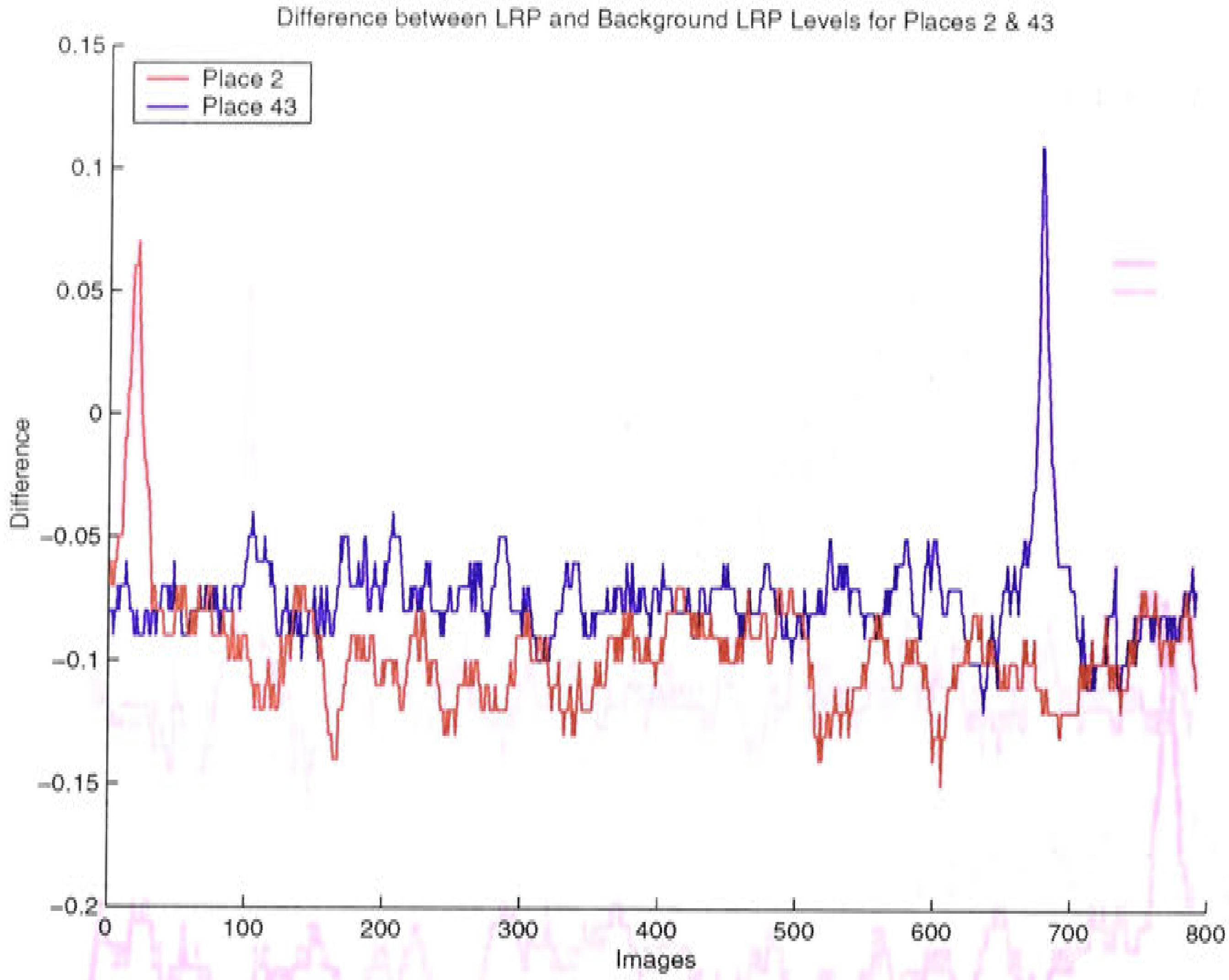


Figure 9.11: The difference between the LRP and the background LRP levels for places 2 and 43 over the example path.

9.3 Combining Global Localisation and Local Positioning

Local Positioning can be used in conjunction with the topological map to estimate the robot's position along a route through a map without initial knowledge as to the robot's position. The global localisation system can be used to identify the MLP at each step along the route. Within the MLP the local position of the robot can be estimated, tracked and passed to the next identified MLP.

In smaller maps with landmark sets that were captured in visually dense environments global localisation and local position tracking can be carried out to produce a reasonable position estimate. When the robot is executing long paths through a dynamic and visually sparse environment however, the robot can become lost, with the particle set distribution, and subsequently the position estimate, diverging from the ground truth position. The nature of the Condensation algorithm which is used to control the distribution of particles in the particle set does not consistently recover

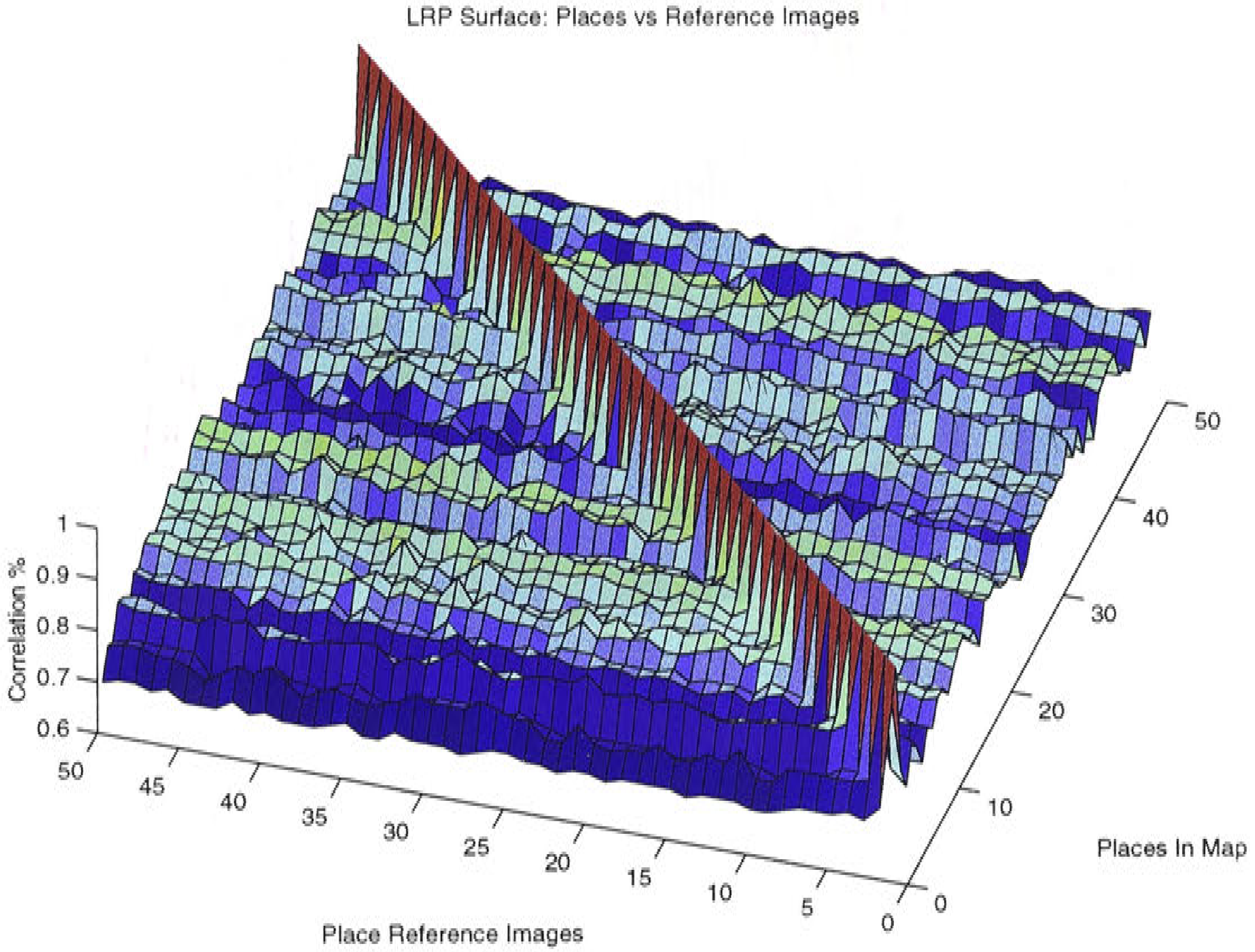


Figure 9.12: The LRP levels for each of the 50 places in the topological map when matched with the places' reference images.

position tracking automatically. When this situation occurs, the robot system must be able to first detect that it has become lost, and second, take steps to recover position tracking.

9.3.1 Detecting Loss of Position Tracking

The robot can detect a loss of position tracking by monitoring its belief as to where it is in the topological map. The robot's belief as to where it is in the topological map is fully represented by the entire distribution of the particle set. The samples approximate the probability distribution of the robot's possible position over the area around the current MLP's reference position. Any attempt to calculate a confidence measure of the robot's local position estimate should depend on the diffusion of the particle set. Such a calculation is problematic in real time. Alternatively, the probability of the most likely particle in the particle set can be used as an instantaneous measure the sys-

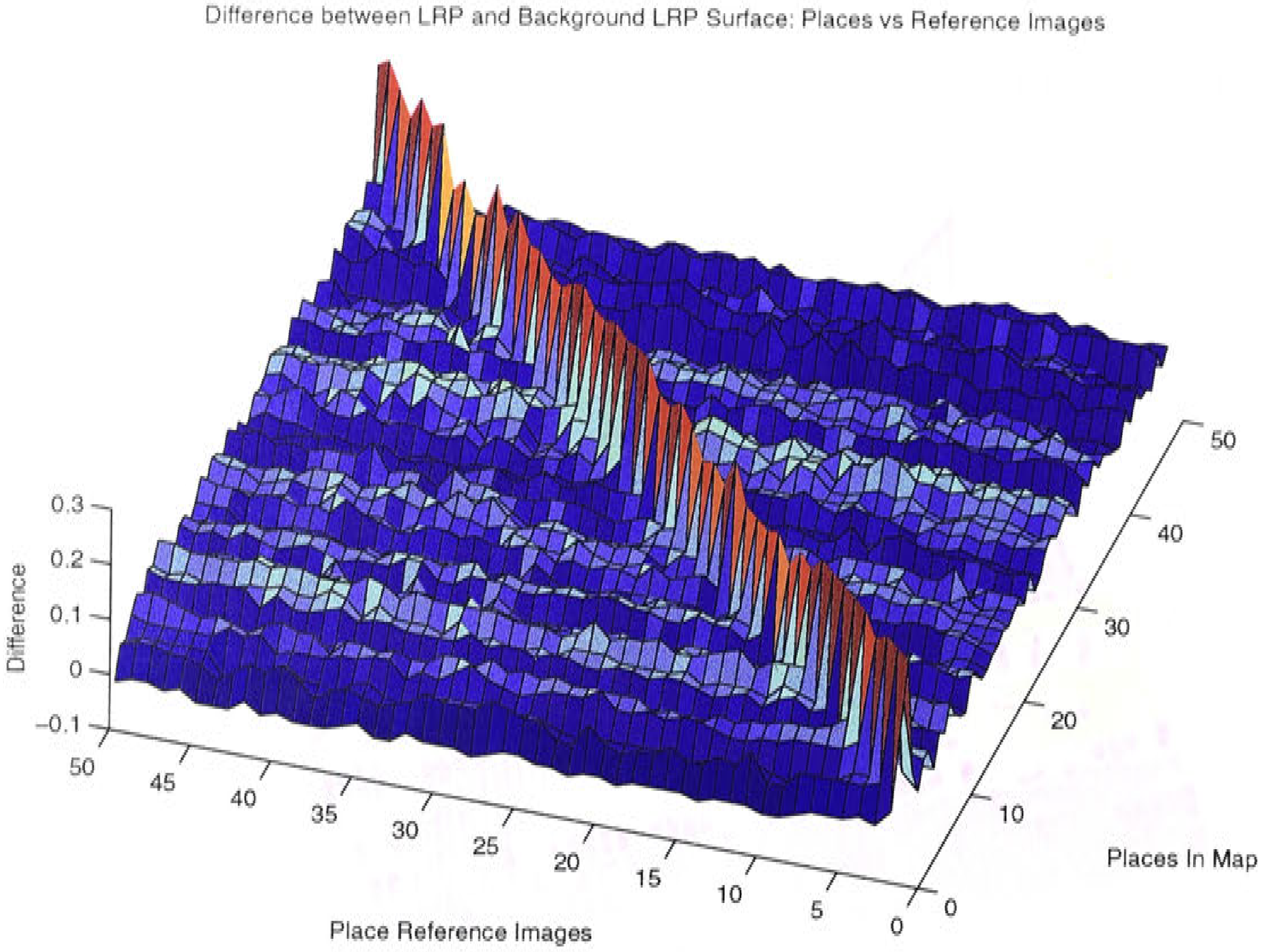


Figure 9.13: The difference between the LRP and the background LRP levels for each of the 50 places in the topological map when matched with the places' reference images.

tem's belief in it's position estimate without significant additional computation. This probability measure is normalised, so the confidence measure relies instead on the raw sensor model output of the most likely particle. This measurement is the output of the sensor model given the current sensory view and the hypothesised robot position as provided by the most likely particle.

Figure 9.14 shows the robot movement between two places in a topological map that was used in the position passing experiment in Chapter 8. Figure 9.17 shows the raw sensor model output for local position estimates along the motion path. The sensor model output is at a high level when the robot makes observations close to the reference position of place 1. The output level decreases as the robot moves away from the reference position. It increases again as the robot approaches the reference position of place 2. This reflects the accuracy of local position information as the observation location moves away from the reference location.

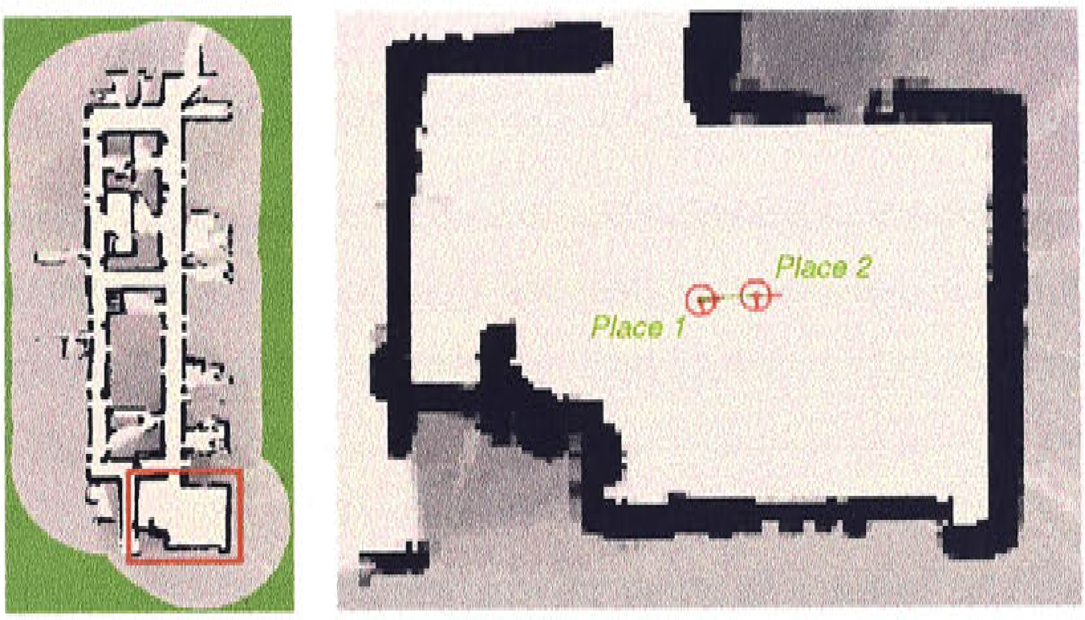


Figure 9.14: The topological map used in the position passing experiment.

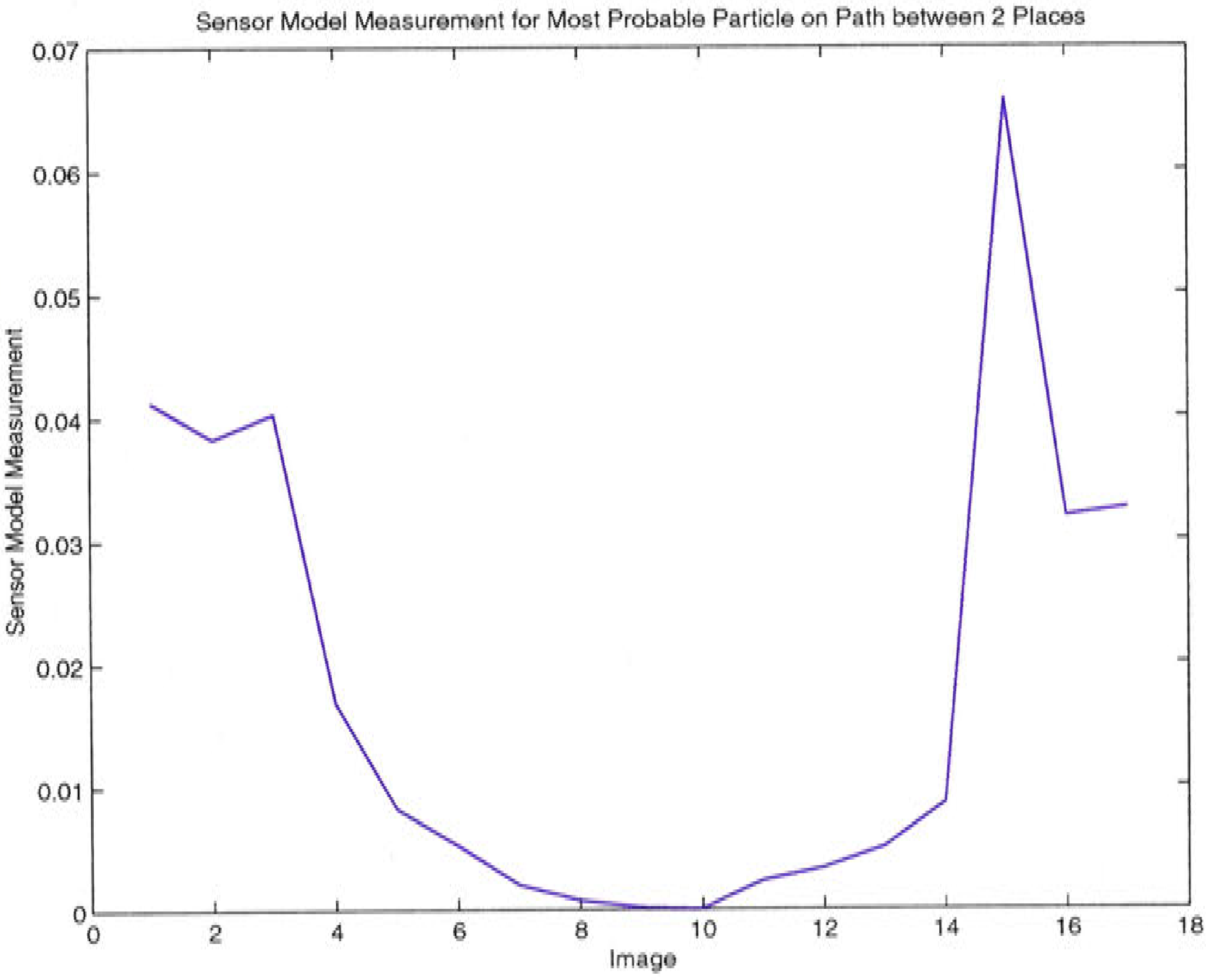


Figure 9.15: Maximum sensor model output for local position estimates along the example robot path.

By reversing the MLP when performing global localisation on the images along the path between the two places, the robot can be fooled into distributing particles at the incorrect place. This produces a local position estimate which is incorrect. This information can be used as a baseline measure for a definition of loss of local position tracking. The peak of the sensor model output for the most probable particle within a mis-identified place is slightly above the sensor model output for pure chance.

With this and other experimental validation the sensor model output threshold, T^{LP} for determining loss of position tracking was set appropriately. The decision as to whether position tracking has been momentarily lost can be made by the predicate:

$$p(t) = P_t^{LP} < T^{LP}$$

where $p(t)$ is the predicate that position tracking is lost at observation t, and P_t^{LP} is the raw sensor model output of the most probable particle at observation t.

The robot position estimation will be unstable if every time position a loss of position tracking as defined causes a mass redistribution of particles. This situation could be caused by momentary occlusion of a large portion of the visual field by dynamic objects in the environment, which is not a true loss of position tracking. To combat this the robot will report a loss of position tracking when the predicate $p(t)$ is true for a number of consecutive observations.

When $p(t)$ has been determined true for a number of consecutive observations, the robot's position tracking is considered to have been lost and the recovery procedure is started.

9.3.2 Recovering Position Tracking

Upon the detection of a loss of position tracking the robot attempts to recover position tracking by redistributing the particle set within the MLP. Such a redistribution can lead to the particle set condensing about the correct region of the robot position state space. This redistribution takes place in two steps:

1. *Local Redistribution:* Upon detecting the loss of position tracking the particle set is distributed about the last estimate in a Gaussian distribution with a variance

of $200mm$. This step attempts to recover from the situation where the particle set has mistakenly condensed into a tight region of the state space, close to the true robot position, and cannot recover through process noise alone.

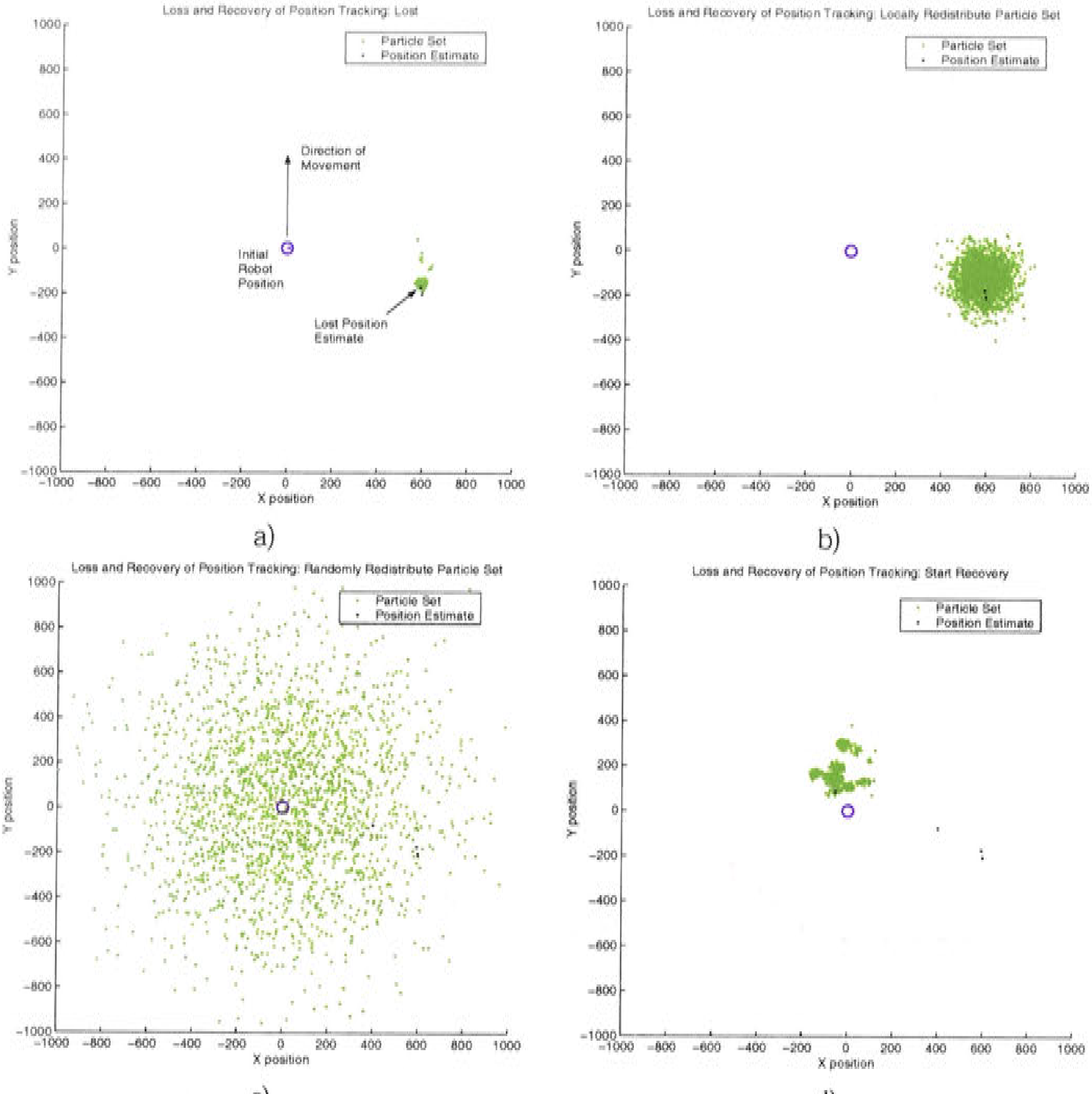
2. *Random Redistribution:* If the local redistribution step does not bring the P_t^{LP} above the threshold, the robot redistributes the particle set randomly throughout the MLP. This redistribution is Gaussian and is centered on the place's reference position and has a variance if $1m$. A redistribution of with a variance of $1m$ should ensure a thorough search of the region in the environment represented by the place, given the distance between places in acquired topological maps.

Position tracking is considered to be recovered when the sensor model output of the most likely particle rises above the threshold value, or the second stage of particle redistribution has occurred, and the count of consecutive observations of lost position tracking is reset.

Figure 9.16 shows an example of position tracking and the two step recovery process. The figure shows position estimation and particle set distribution along the path between the two places from the position passing experiment. Initially the position estimate had been manually set to create a situation where the robot is "lost". Part a) shows the lost position estimate and particle set. The robot detects this loss and redistributes the particles in the local area as shown in part b). After the robot is still lost the particle set as shown in part c) is redistributed randomly throughout the area surrounding the reference position. Finally the particle set condenses about a more accurate position estimate as shown in part d).

Figure 9.17 shows the sensor model output for the position loss and recovery sequence presented above. Initially the out put is essentially zero as the robot is lost, and redistributes the particle set. At image five in the sequence the particle set has been randomly redistributed , and a particle reports improvement in the sensor model output. After that, however it drops briefly as the position estimate is passed between the places in the map, and then grows steadily as the robot approaches the second place in the path.

The robot can detect losses in position tracking within places and also recover to form another sensible local position hypothesis.



a) b) c) d)

Figure 9.16: Loss and Recovery of Position Tracking: a) position tracking is lost; b) robot attempts to recover by redistributing particles around the local area; c) robot attempts to recover by randomly redistributing particles throughout place; d) position tracking has been recovered.

9.3.3 Global Localisation and Local Positioning Experiment

The combination of global localisation and local positioning with loss of tracking and recovery was applied to the path through the large topological map shown in Figure 9.5. The position estimation results are presented in Figure 9.18. Position estimation over the large topological map is noisy however the general path shape can still be observed. Loss of position tracking occurs on several occasions but is recovered each time. The location along the path in at which position tracking was lost are highlighted in Figure 9.18 by light blue boxes. The final position estimate at the end of the

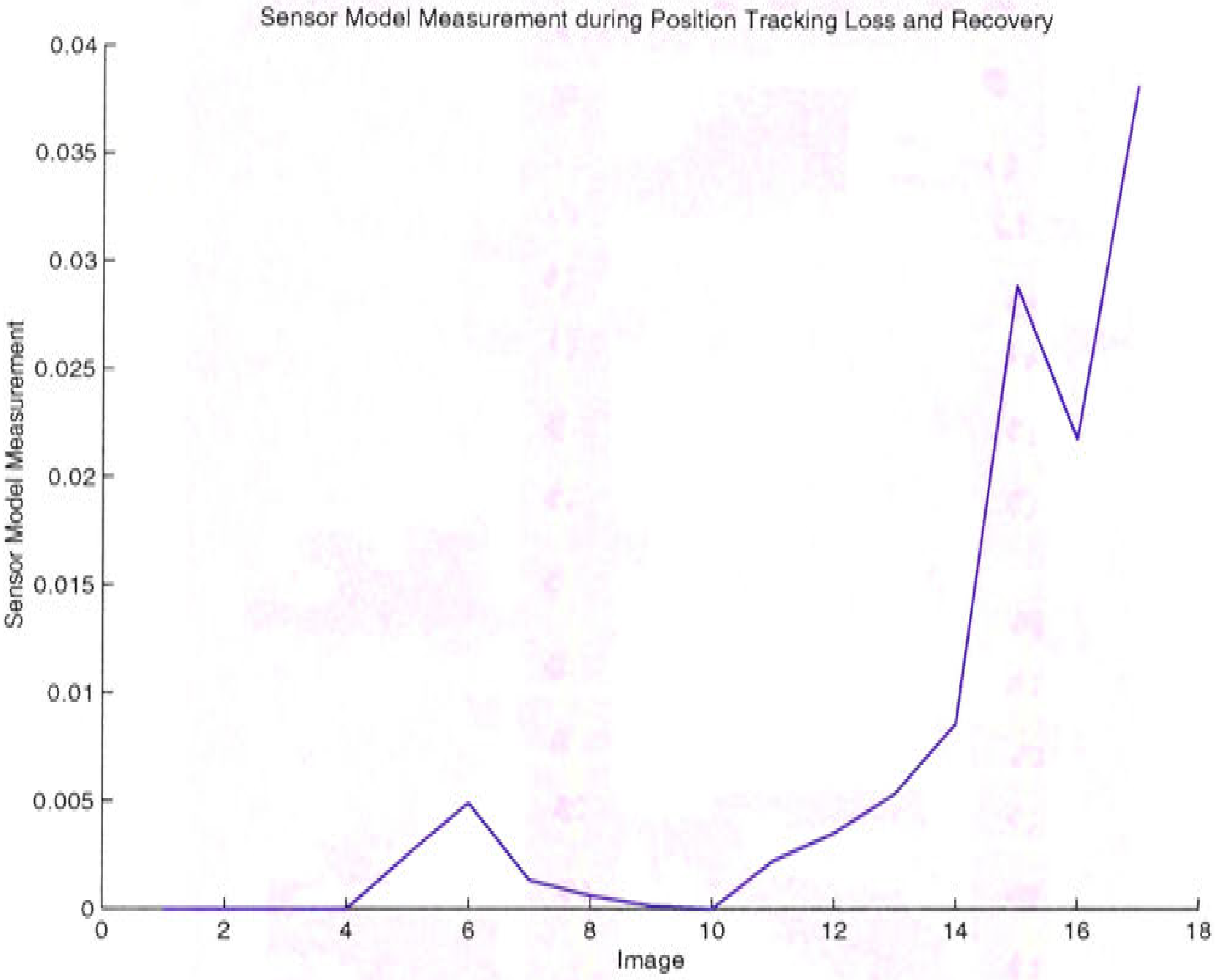


Figure 9.17: Maximum sensor model output for position loss and recovery.

path lies within $15cm$ of the ground truth measurement. When position tracking is maintained the confidence measure is above the loss of tracking threshold and the position estimate stays within a maximum error of approximately $30cm$. Errors in some sections of the estimated path cause the error to move above $1m$ but the robot always recognises that tracking has been lost and takes action that causes eventual recovery of the position estimate.

9.3.4 Computation Costs

The process of global localisation using landmark set matching is computationally expensive. As reported in Chapter 4, matching just one landmark set with one panoramic image takes $700ms$ on a Pentium III $750mHz$ processor. To match all 50 places in the current topological map this results in a total computation time of $35s$. This is unacceptable, even given the non-critical real time constraints this research placed on the global localisation task. In order to make the system usable and provide timely responses to tasks involving human-robot interaction, this computation time must be

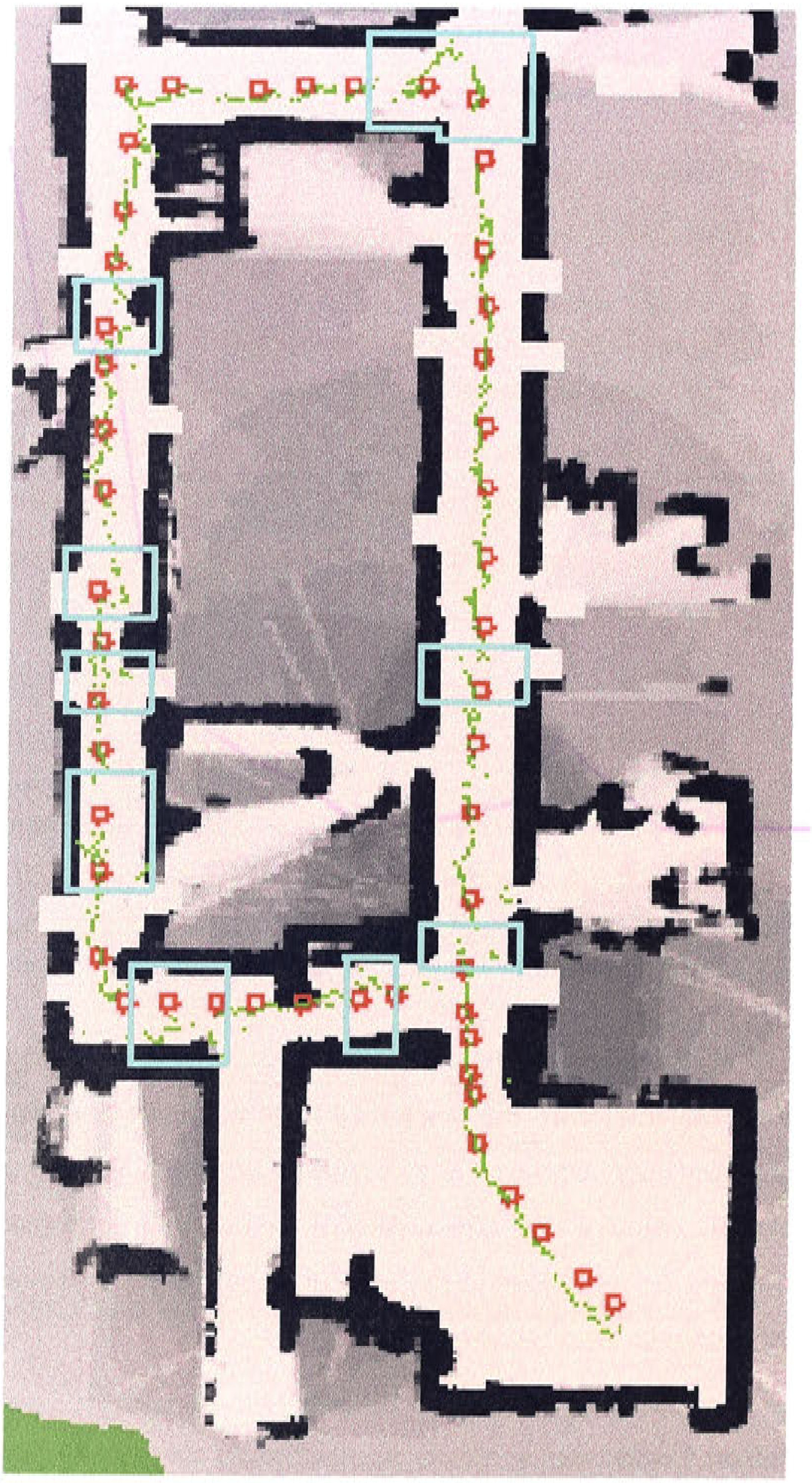


Figure 9.18: Estimated path through large topological map using combined global localisation and local position estimation. The light blue boxes highlight regions where position tracking is poor.

dramatically reduced.

9.4 Constraining the Global Localisation Search

The computation costs involved with performing a brute force global localisation search over the entire topological map are prohibitive except for the most trivial of maps. It is desirable then, to constrain this search as much as possible in order to reduce the computational resources needed for global localisation. We present a method for constraining the global localisation search space using the mid-level spatial representation of local space profiles.

9.4.1 Local Space

The concept of using a profile of the extent of local space to represent a place in a topological map was introduced in Chapter 5. This representation can constrain the global localisation search performed on the lower level landmark representation which is computationally expensive.

When performing global localisation the local space profile extracted from the current sensory view can be matched with the profiles of local space used to represent the places in the topological map. The cost of this matching process is significantly less than that of matching landmark sets. The results of the local space profile matching can then be used to restrict the places in the map that the landmark set matching is applied during the global localisation process.

Local Space Profiles in a Topological Map

The process of forming local space profiles from panoramic images captured by a mobile robot system was described in Chapter 5. Topological maps can be constructed which contain a local space profile of the extent of local space surrounding the reference position of each place. Figure 9.19 shows a topological map containing 10 places which was learnt and which contains local space profiles in it's representations of places in the map. A robot motion path which traverses the map is also shown in the figure.

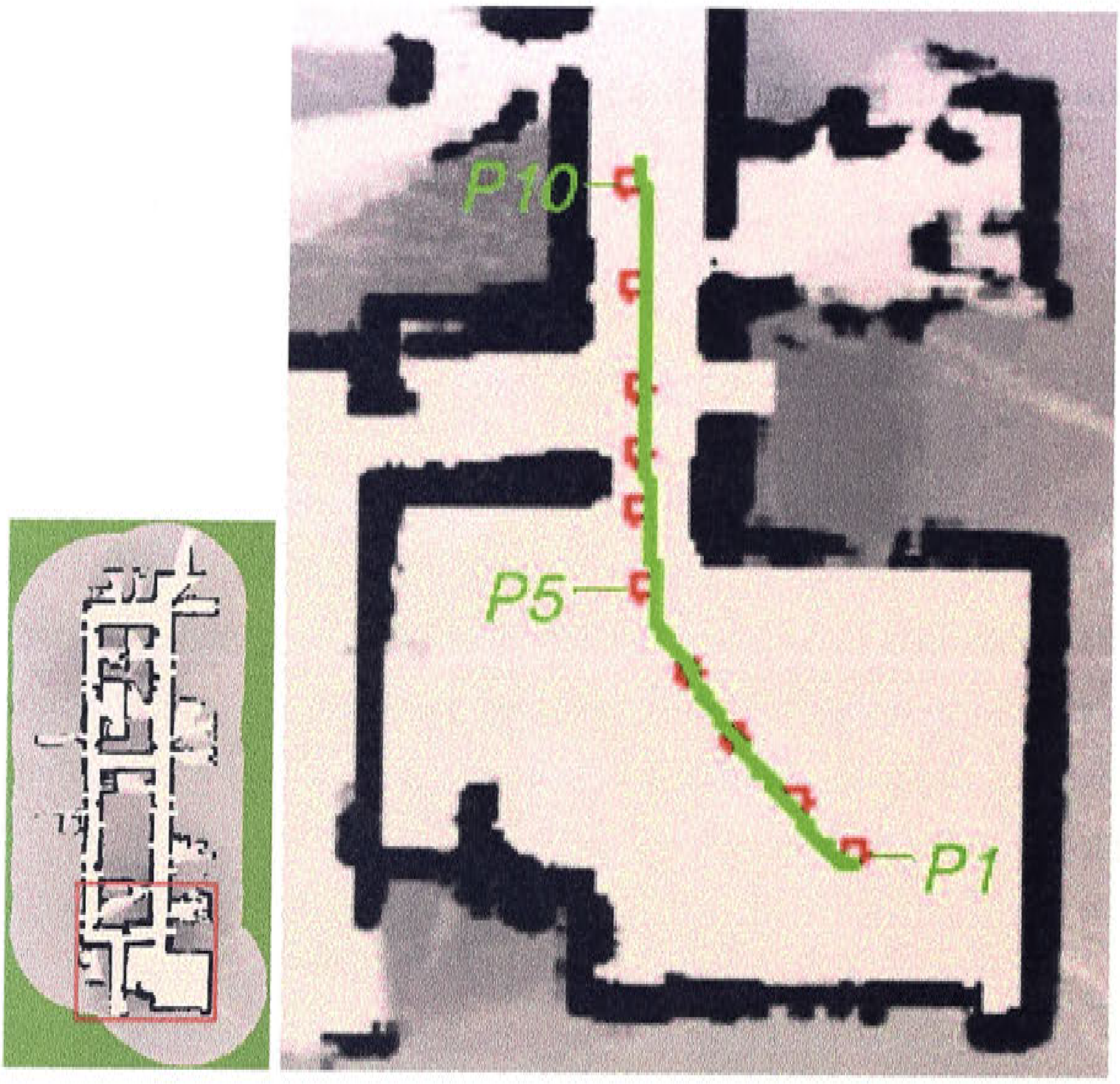


Figure 9.19: A topological map containing local space profiles in the representation for each place.

The local space profiles which represent each of the 10 places in the topological map are shown in Figures 9.20 and 9.21. Each local space profile is displayed as a histogram overlaid on a panoramic image captured at the reference position of the associated place. The local space profiles representing the places in the map can be grouped into three broad categories:

- images captured in the large room and containing a representation of large open spaces;
- images captured around the doorway between the room and the corridor, displaying open areas and constricted regions;
- images captured in the corridor representing restricted open space except in the axis of the corridor.

By comparing local space profiles of the current sensor view with the local space pro-

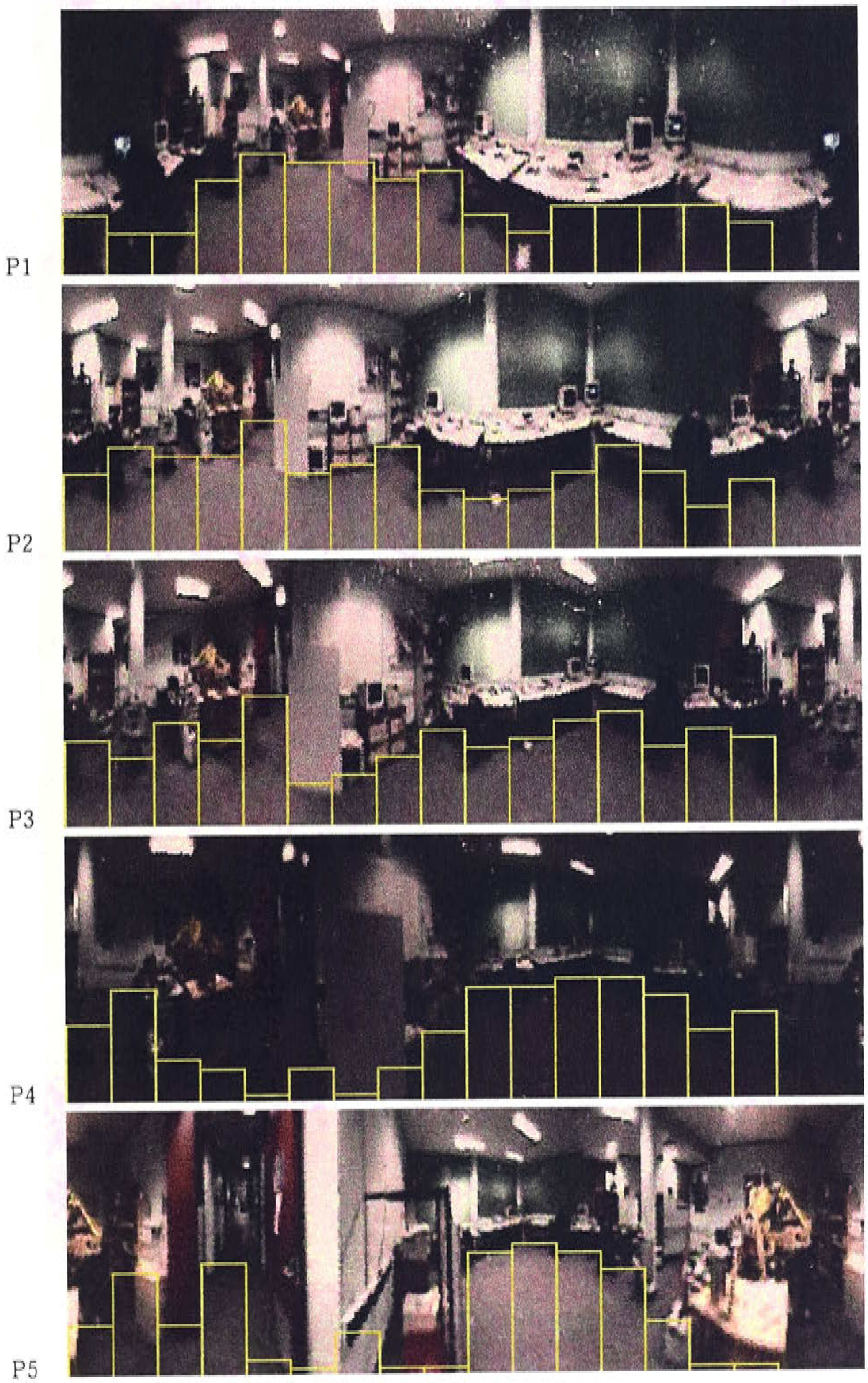


Figure 9.20: The local space profiles of places 1-5 in the topological map.

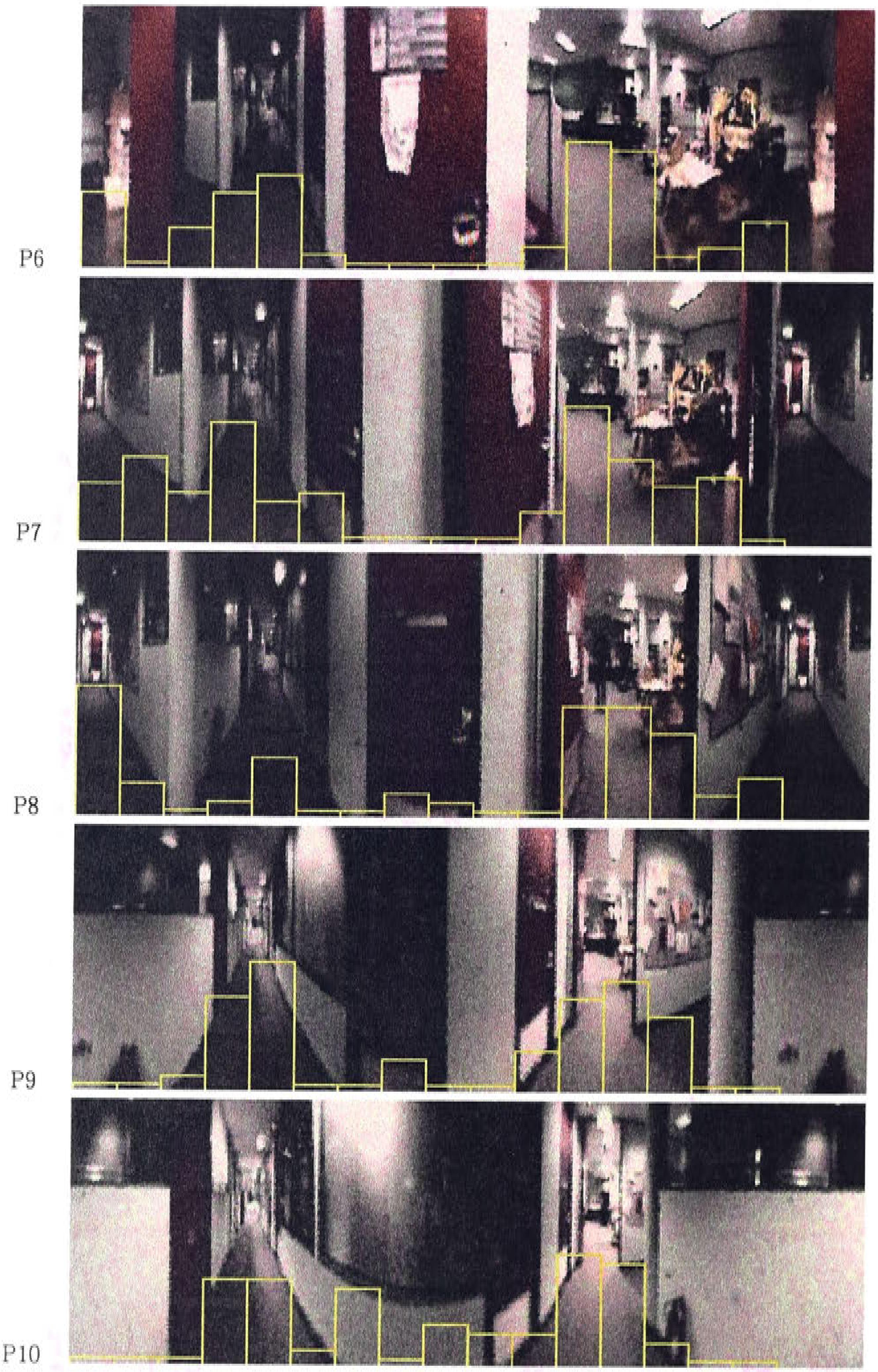


Figure 9.21: The local space profiles of places 6-10 in the topological map.

files if those places in the topological map, the global localisation search space can be at least restricted to places within one of the three categories.

Local Space Matching for Global Localisation

After learning the topological map shown in Figure 9.19, the robot traversed the map following the displayed path. For each image captured along this path the local space profiles were extracted and matched against the local space profiles of the ten places in the topological map.

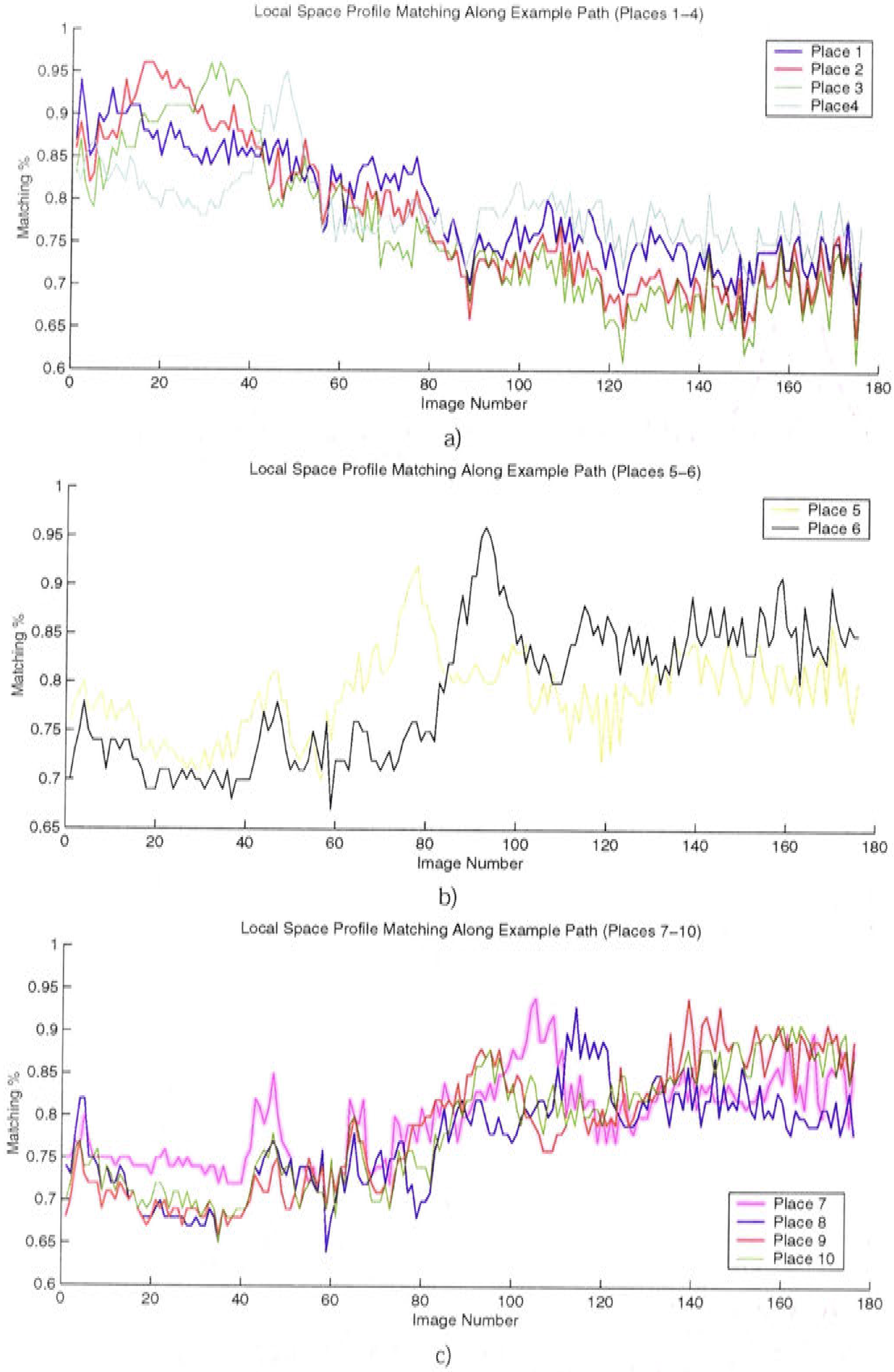
Figure 9.22 shows the results of matching the local space profiles. Each line in the graphs shows the matching performance of one place in the topological map. The results are grouped into the three broad place categories. Part a) shows the matching results for places one to four, part b) places five and six, and place c) places seven to ten.

In general the results show the three categories of local space. There are even rough peaks at the location along the path where each place was learnt. In order to substantiate the local space profiles ability to discriminate between places it is useful to compare the results to those gained from landmark set matching discrimination.

Figure 9.23 shows the Landmark Recognition Performance (LRP) over the example path for each of the landmark sets representing the 10 places in the topological map. The LRP results are also divided into the three categories for comparison with the local space results. The LRP provides a better measure for discriminating between places, and it is unclear that the best local space matching results always coincide with the best LRP results. It is clear that the local space matching results do provide information pertinent to constraining the global localisation search.

Computational Benefits of Local Space Matching

The local space matching results can be used to constrain the landmark based global localisation search. The results must be categorised in a qualitative fashion in order to identify which places in the map to search further. A simple way to do this is to use sets of places which are above a given threshold for local space matching. If this was a static threshold, many occasions could occur where the noisy local space sensor

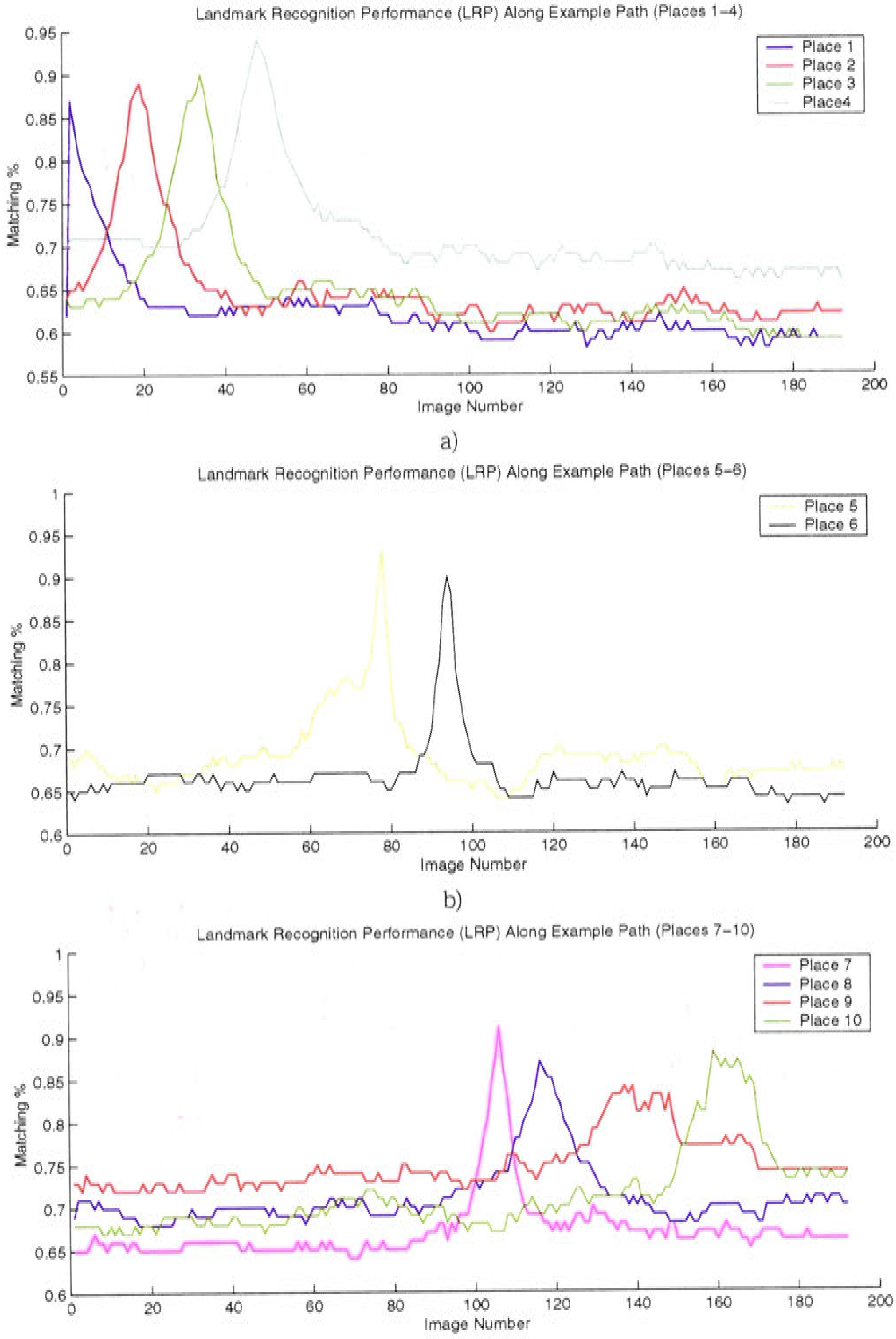


a)

b)

c)

Figure 9.22: Local space profile matching along the example path. Places are grouped into plots with similar response curves.



a)

b)

c)

Figure 9.23: Landmark Recognition Performance along example path. Places are grouped in plots by similarity of the local space profile.

data fails to identify the correct set of places. In our system the threshold limit at each iteration of the global localisations search depends on the current best local space matching results:

$$T_t^{LS} = max(match(C_t^{LS}, S_1^{LS}), \ldots, match(C_t^{LS}, S_N^{LS})) - B \qquad (9.3)$$

where T_t^{LS} is the local space matching threshold value at iteration t, $match(C_t^{LS}, S_i^{LS}$ matches the local space profiles of the current view at time step t with the local space profile for place i, and B is a constant value which is subtracted to give a lower limit on the threshold for inclusion in the matching set.

The set of places with which to perform landmark set matching can be defined as those having a matching value of greater than the threshold:

$$G_t^{LS} = \{S_i \in \{S_1, \ldots S_N\} \,|match(C_t^{LS}, S_i^{LS}) > T_t^{LS}\} \qquad (9.4)$$

where G_t^{LS} is the set of places at step t that is identified for further global localisation searching by the local space matching results.

The value of B is obviously crucial to the success or failure of this approach. The value of B should be chosen to maximise the computational savings which results from constraining the global localisation search space, while maintaining inclusion of the correct place in the identified set. This means that the value of B should minimise the size of the set while achieving acceptable rates of set inclusion for all possible views of the current local space profile S_t^{LS} over the areas covered by the topological map.

To select the best possible value of B, the threshold values T_t^{LS} and the resulting set of places G^{LS} were evaluated for every captured image over the example path shown in Figure 9.19 for values of B ranging from 0.0 to 0.225. Set G^{LS} was evaluated for set inclusion by determining whether the place identified as the best match by landmark matching is an element of the set.

The set inclusion results for the range of lower limits (B) over the example path are shown in Figure 9.24. The horizontal axis of the plot shows the possible values of the lower limit B while the vertical axis shows the percentage of the sets G_t^{LS} over the example path that include the correct place. The percentage of sets which include the

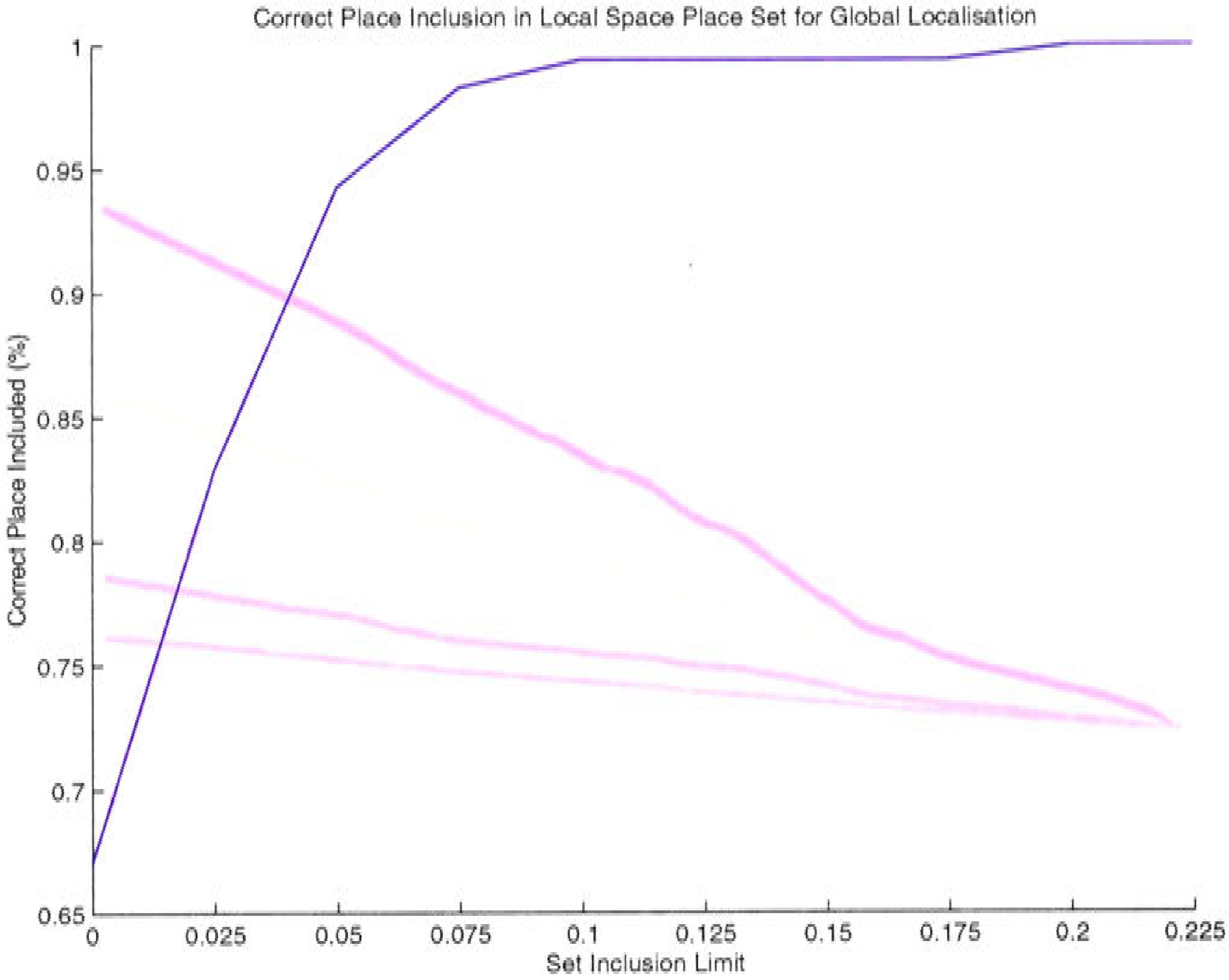


Figure 9.24: Correct place inclusion in local space set for global localisation.

correct place is quite low ($\sim 60\%$) when B is zero. This is equivalent to identifying the MLP purely by the local space matching results. From this it can be seen that it is not possible to rely solely on the local space profile matching for global localisation. The percentage of sets which include the correct place rises quickly however, as the value of B increases and levels of at above 99% at around the value of 0.1.

The set inclusion percentage must be balanced against the global localisation computation costs which grow as the size of the set increases. Figure 9.25 shows the cumulative number of places included in the sets G_t^{LS} over the example path for a range of lower limit values. At each image in the path, the size of the set G_t^{LS} was evaluated for each value of B and this size was summed over the image path. The plot shows that the growth in total required computation presented by the cumulative set size is almost linear and the slope of the relationship depends on the value of the lower limit B.

Figure 9.26 shows the average set size for the various values of the lower limit B. This relationship is also linear with the size of B, meaning that as B increases the size of

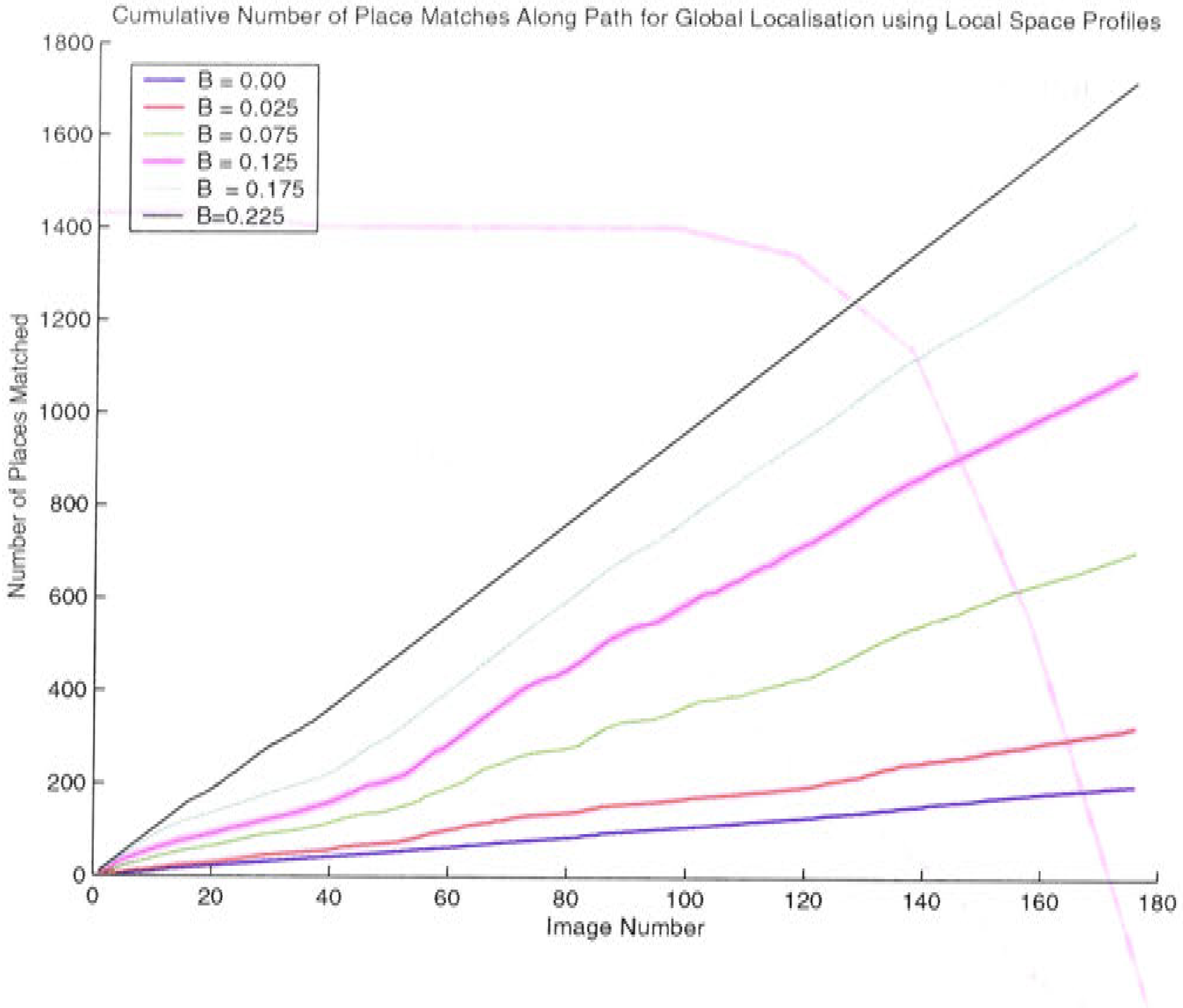


Figure 9.25: Cumulative number of places in the local space set during global localisation over the example path for various lower bounds to the local space matching limit.

the set G_t^{LS} increases proportionally.

Given that there is a sharp plateau of set inclusion percentage and that the computation cost increases linearly, it is clear that the value of B must be chosen before the set inclusion plateauing effect reduces the contribution of any additional computation caused by increasing set size. Therefore in our system the value of B is set to 0.075 which gives a set inclusion percentage of 98%, and a average set size of 4.01. These values can be compared with the a full search of the global localisation space which has 100% set inclusion and an average set size of 10. By using the second level of spatial representation, that of local space primitives, the search space of global localisation can be reduced approximately 60% at the cost of less than 2% in set inclusion performance.

In terms of total computation savings generated by the use of a local space representation, the benefits can be calculated by comparing the cost of global localisation using

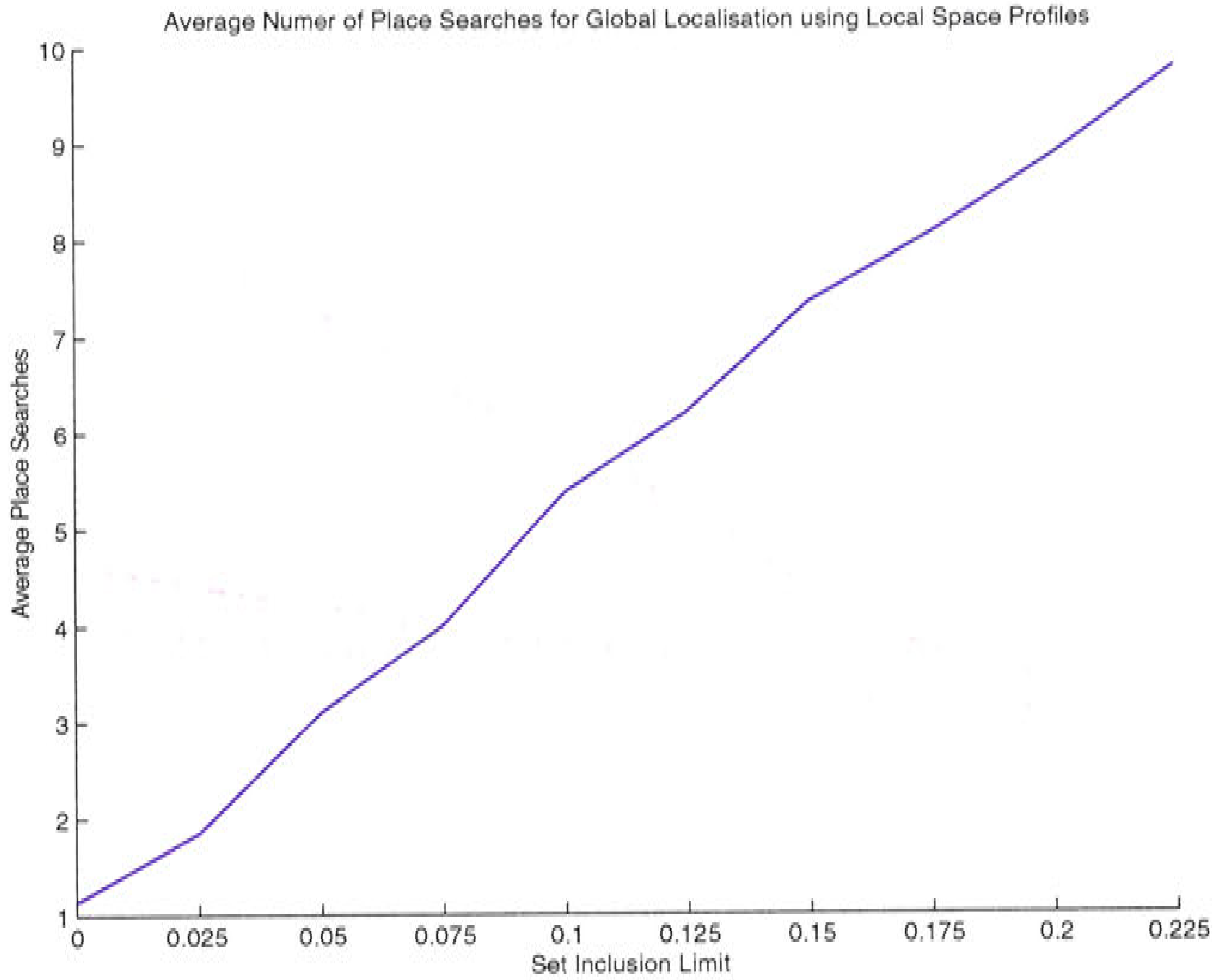


Figure 9.26: Average local space set size during global localisation for various lower bounds to the local space matching limit.

landmark recognition for all places in the topological and the cost of global localisation on the constrained place set plus the additional cost of performing the local space matching.

Given that the cost of matching for one place using local space profiles is less than $1ms$ when comparing it to the cost of matching landmark sets , $700ms$ per landmark set, the additional costs involved with local space matching of places in the topological map are negligible when compared to the savings. Total computation savings derived from the use of the local space representation then, can also be estimated at approximately 60%.

9.4.2 Continuous Global Localisation

The knowledge gained from the local space profiles can be used in conjunction with the particle filter system to provide continuous global localisation and local positioning. In order to achieve this the mobile robot system must perform the following three steps:

1. *Constrain the Search Space:* as described previously, the localisation search space of all places in the topological map must be restricted in order for the localisation process to approach computational tractability.
2. *Perform Global Localisation:* Identify potential places in the map that the robot is most likely to be at at this particular moment in time.
3. *Perform Local Position Estimation:* use the global localisation information to distribute particle filter samples to appropriate places in the topological map to perform local position estimation.

This process can be used iteratively to provide continuous localisation. However it is desirable to make a distinction between the localisation upon initialisation of navigation, and that of localisation with a prior estimation of robot position.

Active Place Set

A subset of places can be defined in the topological map in order to restrict the global localisation search. Membership of this set can be determined using the local space profile representation. This set of likely places is named the *active place set*. Global localisation using landmark recognition is performed with members of the active place set. In addition to the set of places G_t^{LS} identified by local space matching, the active place set, A_t contains as members the MLP from the last iteration P_{t-1}, and its closest adjacent place as defined by the transition information, $P_{t-1}^{\rightarrow}$. The complete active place set can then be defined as:

$$A_t = G_t^{LS} \cup P_{t-1} \cup P_{t-1}^{\rightarrow} \tag{9.5}$$

The previous MLP is added to the set identified by local space matching to provide stability to the local place estimate in the presence of noisy local space estimates that occur when moving objects temporarily occlude large portions of the panoramic camera's visual field. The place associated with the closest transition is also included in order to detect when the current local position estimate passes into the domain of an adjacent place in the map that may not have been identified by the local space matching process.

Examples of active place sets constructed by applying local space matching and Equation 9.5 to real world localisations situations are shown in Figure 9.27. Each of the figures show a topological map and the active place sets derived from observations made at places along a path. Places which are included in the active place sets are drawn in green, whereas the places which are excluded from the set are in red. The positions from which the sensory views which produced the active place sets where taken are displayed in yellow. It can be seen that all identified active place sets include the places closest to the position from which the sensory views were captured. The active set in Part a) contains only two places , thus providing a significant computational saving for the subsequent localisation task, while part b) only managed to restrict the active place set to six places. Part c) identifies three potential places.

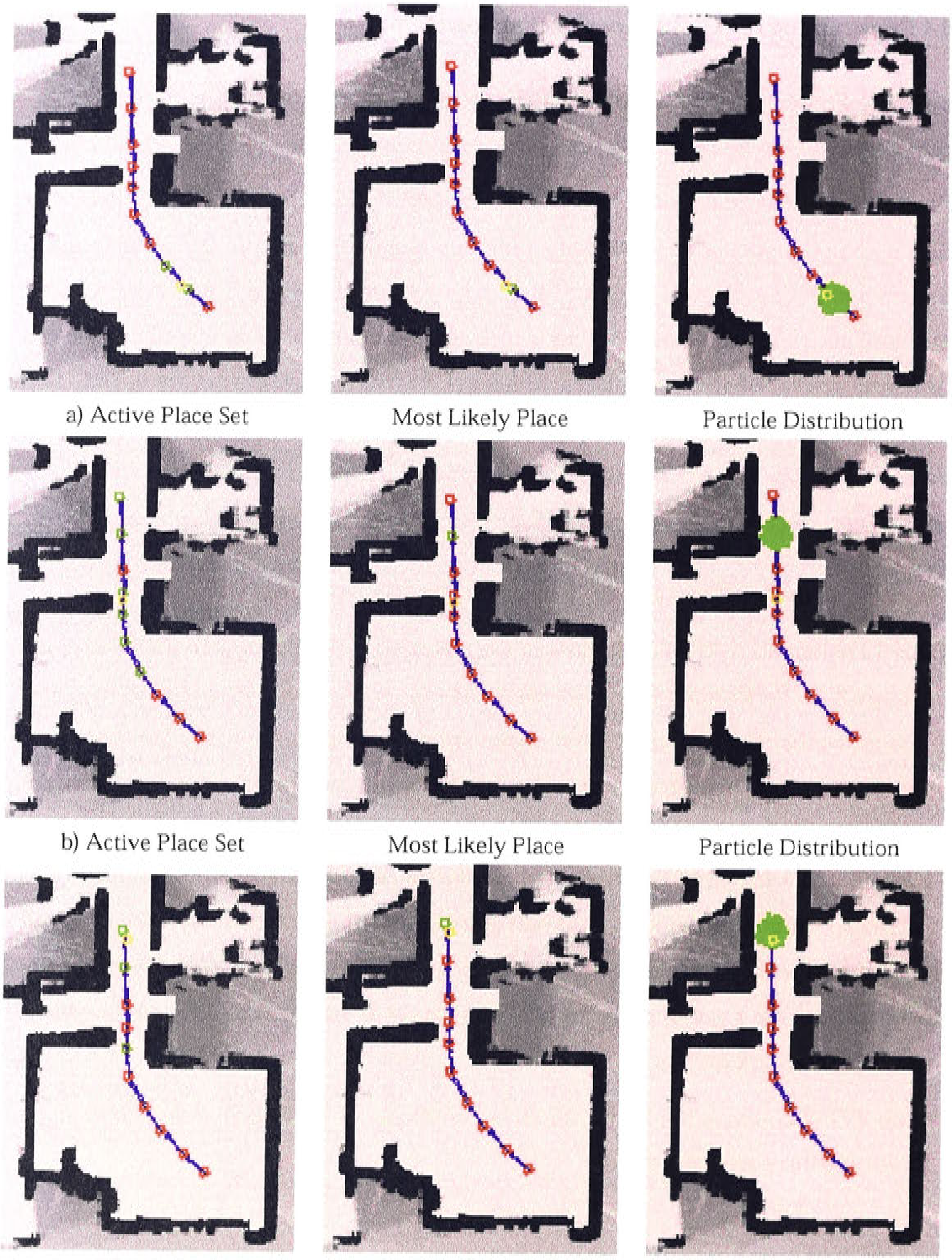
It should be noted that the active place sets in a) and c) are from locations along the path which are close to particular places, thus providing definite place matching results and subsequently small active place sets. The location from which the active place set in part b) was produced is closer to the center of two places, thus providing a weaker matching result and a large active place set.

Initial Global Localisation

If a mobile robot system has been turned-on or if the robot has detected it has become lost, it is necessary to perform global localisation with no prior knowledge as to the robot's location in relation to the topological map. The robot must make a full search of the places in the active place set to successfully localise itself.

If computation resources are scarce, or there is a need for the robot to immediately move, the robot could make an almost random guess as to likely places and proceed with the global localisation process as from a known position. In our system we assume that it is acceptable for the robot to initially perform global localisation at less than real time rate.

To accomplish the initial global localisation step, the robot system simply matches the landmark sets of each place in the active place set, the place with the highest LRP is



a) Active Place Set

Most Likely Place

Particle Distribution

b) Active Place Set

Most Likely Place

Particle Distribution

c) Active Place Set

Most Likely Place

Particle Distribution

Figure 9.27: Three examples of global localisation and local position estimation using the local space profiles to constrain the global localisation search.

Figure 9.27	Places in Active Place Set	LRP	Most Likely Place
a)	2	0.87	2
	3	0.68	
b)	4	0.70	9
	5	0.65	
	6	0.69	
	7	0.73	
	9	0.73	
	10	0.68	
c)	6	0.66	10
	9	0.78	
	10	0.87	

Table 9.1: LRP for places in active place sets

identified as the MLP and global localisation has been achieved.

Table 9.1 shows the LRP of each place in the active set for the three active set examples in Figure 9.27. Performing landmark set matching on the example active sets in a) and c) identifies the place nearest to the observation location as the MLP. The active set in b) fails to produce a strong candidate for the MLP when performing landmark set recognition, although the correct place, place 7, is in a tie for the highest with place 9. This is due to actual robot position lying between places in the topological map. The system can randomly choose the correct place as being the most likely. The rounded figures could reflect a situation where the tie is not actually present. However for the sake of producing a problematic case for analysis, it is assumed that the robot system incorrectly chose place 9 as the MLP.

Figure 9.27 also shows the MLP's identified from places in the active place sets using LRP for the three example cases.

Local Position Estimation

Once the MLP has been identified, local position estimation can proceed. For local position estimation occurring after initial global localisation or when relocalising from a lost state, this means distributing the particle set about the reference position of the learnt place. Examples of distributing the particle set in identified places are shown in Figure 9.27. Parts a) and c) of the figure show particles distributed about correctly identified places while part b) shows what occurs when the identified place does not correspond to the place nearest to where the current observation was made.

If the place in question already has a place estimate or a transition has been detected, local position estimation can proceed. If the particle set is distributed around the incorrect place, the sensor model results will detect this situation and relocalisation can occur through the continuous mobile robot localisation process.

Continuous Mobile Robot Localisation

Once global localisation and an initial distribution of particles within the most likely place has been achieved, the mobile robot system must maintain an estimate of the robot's position while executing movements throughout the environment. In the case that the correct robot position has been estimated, as in Figure 9.27 parts a) and c). This means the robot must perform position tracking to maintain the estimate with limited exploration of other places in the map to detect possible transitions between places. In the cases where the robot has incorrectly identified the MLP, the robot must be able to detect the loss of position tracking and perform a limited relocalisation.

The important feature of position tracking within and between places in continuous mobile robot localisation is the restriction on the size of the active place set. When performing position tracking the robot uses the current active place set identified by the local space profile matching, which also contains the previously identified MLP and the adjacent place reached by the closest transition to the current estimated position.

To perform probabilistic localisation over the entire map the particles should be spread throughout likely places in the map to achieve a correct approximation to the robot position probability density function. However in our system, due to the high recognition accuracy of places and the excessive computation time for landmark set recognition we decided to spread particles about the MLP only . The computation of the likelihood of the robot being located in other places contained in the topological map is left to the local space profile and landmark set matching.

Which places in the active place set should landmark set recognition be applied in order to identify the current MLP? If the computational resources ware available it would be desirable to apply landmark set recognition to all places in the active place set. However, given the high computation cost of landmark set recognition it is desirable to further limit the active place set. While the measure for local position confidence is high then the MLP is assumed to be correct and only the MLP and the adjacent place need to be searched for their LRP:

$$\text{if} \quad P^{LP}_{t-1} < T^{LP} \quad \text{then} \quad A_t = P_{t-1} \cup P^{\rightarrow}_{t-1}$$

where P^{LP}_{t-1} is the sensor model output of the most probable local position estimate in the most likely place P_{t-1}, T^{LP} is the local position estimate threshold and A_t is the resulting active place set.

If local position tracking has been lost then the active place set must be redefined to ensure the localisation system explores the places in the set constructed by the local space matching results:

$$\text{if} \quad P^{LP}_{t-1} \geq T^{LP} \quad \text{then} \quad A_t = B_t \cup P_{t-1} \cup P^{\rightarrow}_{t-1}$$

where B_t is a sub set of places from the set G^{LS}_t.

The formation of set B depends on the computational resources that are available. Obviously the more places that are contained in set B, the greater the chance that the MLP will be correctly identified and the position tracking regained. Conversely, the less places in B the faster the landmark recognition for each place can be performed and another sensor observation can be captured. In such a small map, as used in our examples, B was set to contain all elements from G^{LS}_t.

Figure 9.28 shows the flow of control in the proposed system when continuous global localisation and local position estimation are performed.

Figure 9.29 shows a sequence of continuous global localisation and local position estimation using a restricted global localisation search space. This figure shows example b) from Figure 9.27, where the active place set has incorrectly identified the MLP. In the current figure, the places in the topological map are shown in red, places in the active place set in green , the particle set also in green, and the actual current robot position in yellow. In part a) the robot attempted to perform local position estimation in an incorrect place, and the system detected a loss of position tracking. Part b) shows the system redistributing particles about the local current estimate. After this strategy fails, local space matching is performed to produce a new active set and the

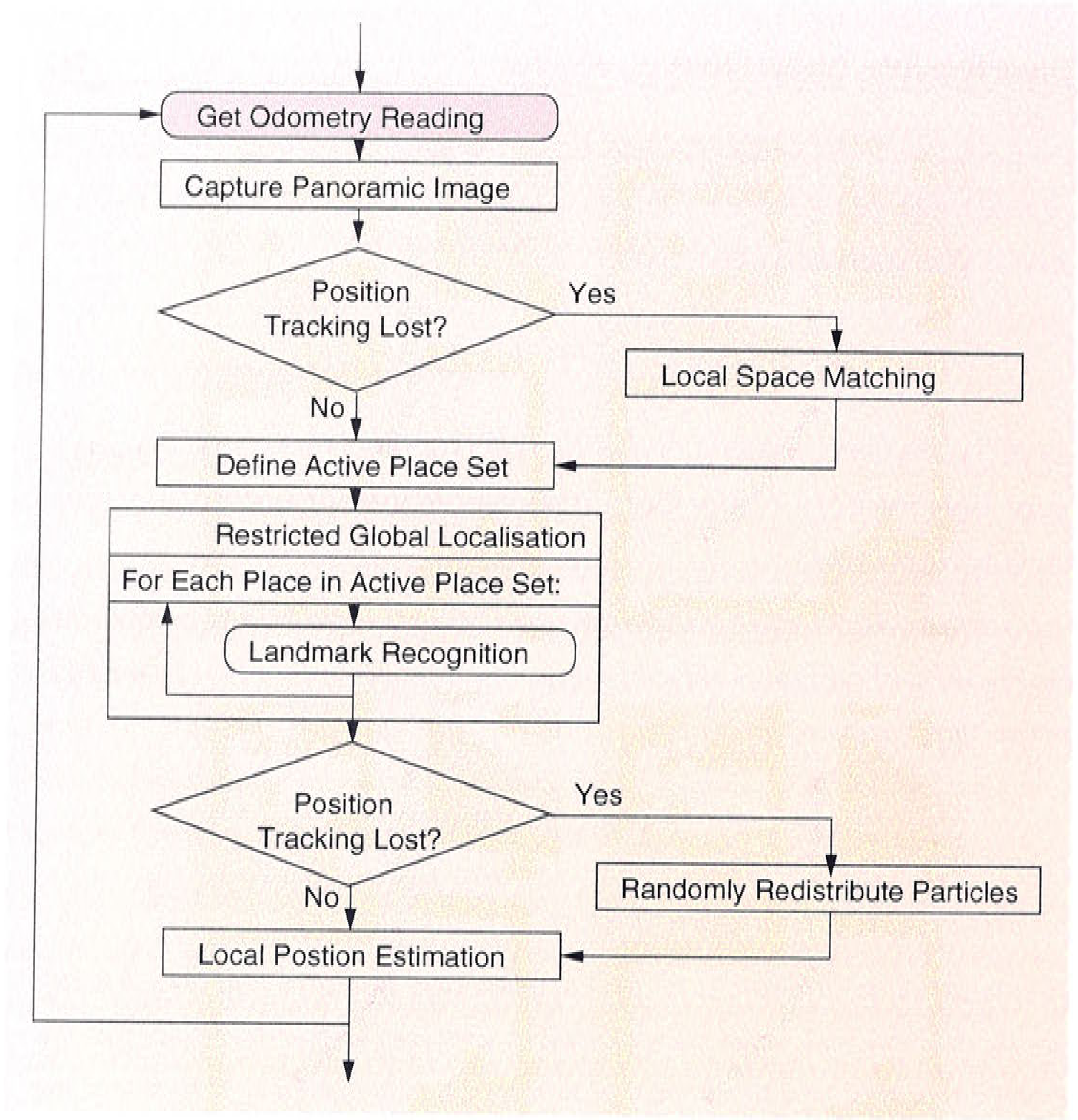


Figure 9.28: The flow of control when performing continuous mobile robot global localisation and local position estimation.

system performs a restricted global localisation search. This produces a correct MLP, and the position estimates are redistributed about this new place, as shown in part c) of the figure. Part d) shows the state of the system after the next observation, when the particle set condenses around a new, more accurate local position estimate.

During this process, the size of the set of places upon which landmark template matching is performed did not rise above 3, compared to a maximum of 10 for unconstrained global localisation.

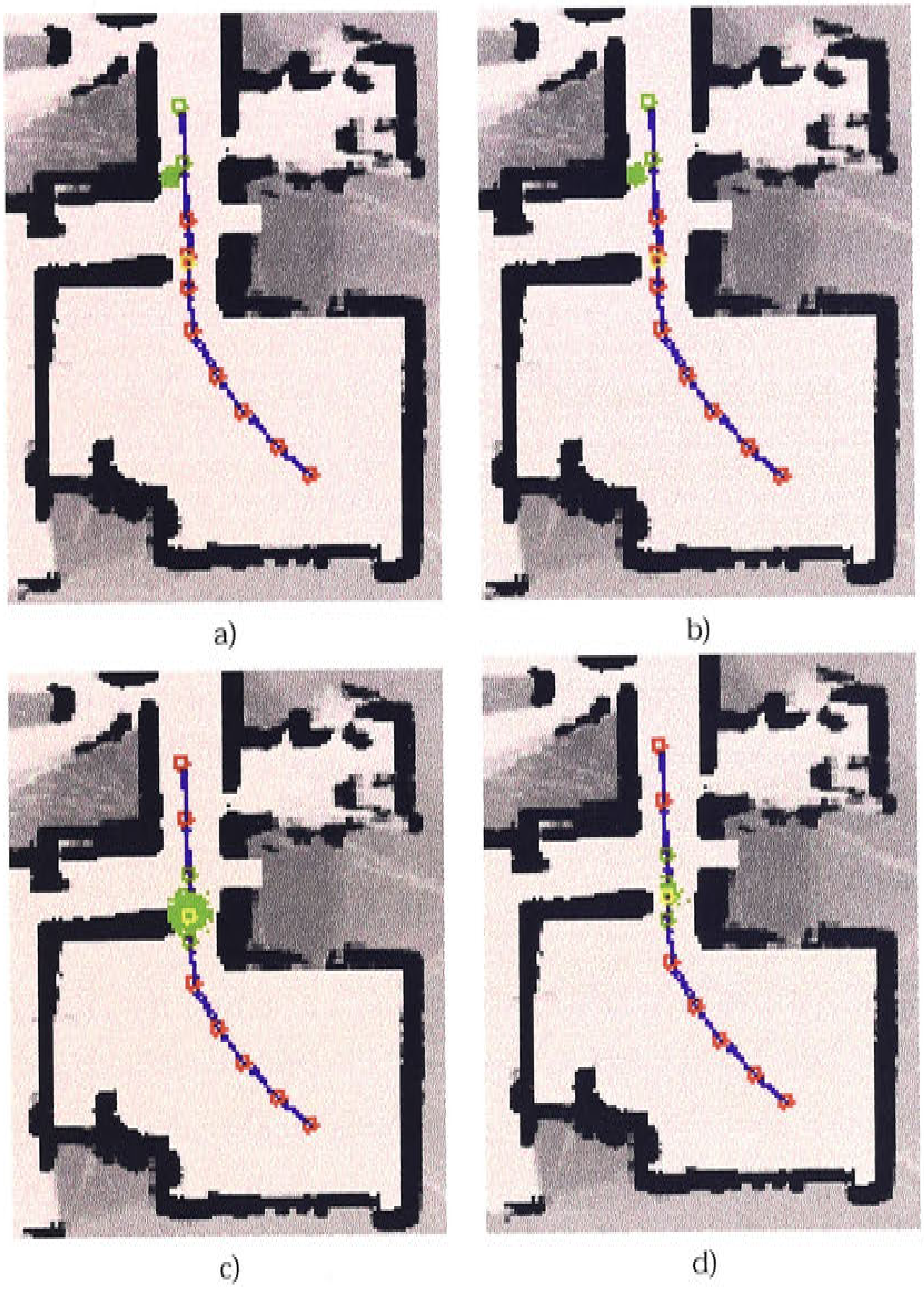


Figure 9.29: Continuous global localisation and local position estimation when the initial position estimate is wrong. The particle set representing the robots current belief in its location is shown in green, while the actual location is drawn in yellow.

9.5 Disambiguating Similar Places

All landmark or feature based localisation systems will encounter situations where the localisation information provided by the underlying spatial representation will be ambiguous. An obvious example of this is a robot system in a featureless corridor environment. Given a spatial memory based on a finite number of representation modalities, a corridor can be imagined where all the modalities could fail to provide unambiguous localisation information. Instead of relying on predefined landmarks or features to guide the localisation process, it can be useful in these pathological cases to actively search for discriminating features with which to eliminate localisation ambiguities. Chapter 6 introduced a method for extracting disambiguating features from panoramic snap-shots of places in a topological map and to use these features to discriminate between places for the purposes of mobile robot localisation.

Our method is expensive and unreliable however, and is not ready for application to a working mobile robot localisation system. In the present case, the use of distinct visual landmarks to perform place discrimination is successful in a vast majority of instances and the addition of the third level of spatial memory to the localisation process does not significantly improve localisation performance. Chapter 6 briefly discusses some options for improving the contribution of this level of spatial representation, but for now, its inclusion is only to highlight the need for actively searching for discriminating features.

Although our system does not attempt to use the third level of disambiguating features in the reported localisation experiments, it is useful to describe the basic steps of integrating a such a level of spatial representation into the existing mobile robot localisation system:

1. *Detect an Ambiguity in Localisation:* a method of detecting the ambiguity arising from the lower levels of spatial representation is needed to trigger the disambiguation process. In the present system, this would involve the Landmark Recognition Performance of two or more places being equal. Of course given the expense of such a step, there should be some continuous time period during which an ambiguity is detected before it is reported. This would eliminate the need to perform disambiguation when the cause of the ambiguity is temporal in nature, such as dynamic objects occluding the visual scene or loss of landmark tracking.

2. *Decide Whether to Disambiguate:* not every ambiguous situation requires disambiguation before purposive navigation can be achieved. A system which uses disambiguating features needs to be able to decide when such an action is appropriate and when navigation in ambiguous circumstances is acceptable. A suggestion for making this determination could consider the potential expense

of mis-planned routes when an ambiguous localisation is accepted.

3. *Perform Disambiguation:* Finally the method of disambiguating features can be used to discriminate between ambiguous places in the topological map. Local position estimation and position tracking can then proceed as normal.

Our mobile robot system does not use the disambiguating features level of spatial representation for the reasons listed previously. A truly robust localisation system, however, would require a method for performing location discrimination like that of disambiguating features to actively search for discriminating features in pathological situations.

9.6 The Kidnapped Robot

The kidnapped robot problem, as introduced in Chapter 1, remains an open problem in contemporary mobile robot localisation systems. The problem restated is:

... given a robot system which has a strong belief in its location within an internal map of the environment, "kidnap" the robot by transporting it to another location in the environment, without the robot being aware of the translocation. The robot now must realise it has been "kidnapped", and further re-localise itself within it's internal map ...

This problem is equivalent to relocalising a lost robot, with the additional nuisance of the robot moving instantaneously from a known location in the map to an unknown without any odometric or clues to the transition. A robust solution to this problem remains elusive due to the various sub-problems which must be first solved:

1. Detection of Loss of Position Tracking

2. Global Relocalisation

3. Recovery of Position Tracking

The global relocalisation step has proven to be the most intractable, especially in environments where large maps are necessary and systems that have real time constraints on robot response.

The previous experimental results in this chapter have addressed the sub-components of the kidnapped robot problem. It has been shown that the multi-level spatial representation can be used to perform recovery of loss from position tracking as well as global localisation with a constrained search space.

9.6.1 Multi-Level Spatial Memory and the Kidnapped Robot

The multi-level spatial memory approach to global localisation can be applied to the kidnapped robot problem in the same way it was used to re-localise a lost robot. Local positioning using the low-level representation can be used to form a strong belief that the the robot is in a particular location in the topological map. Upon kidnapping and release in another location, the robot can detect a loss of position tracking and can attempt to re-localise using first the local space representation to constrain the global search and then the landmark representation to select a MLP and initialise position tracking.

An experiment to test the systems ability to solve the kidnapped robot problem, given a topological map and a set of panoramic images and odometric measurements from an example path through the map, involves the following steps:

1. Identify a *start position* along the path, where the robot will be positioned prior to the kidnapping, and a *release position* along the path from where the robot will be released after kidnapping.

2. For the first 5 frames (frames $0-4$) from the *start position* localise the robot within the topological map. A frame refers to a sensor data sample from a particular location along the example path.

3. After frame 4, the robot is kidnapped. Continue localisation, but present the sensor data associated with the *release position* along the robot path.

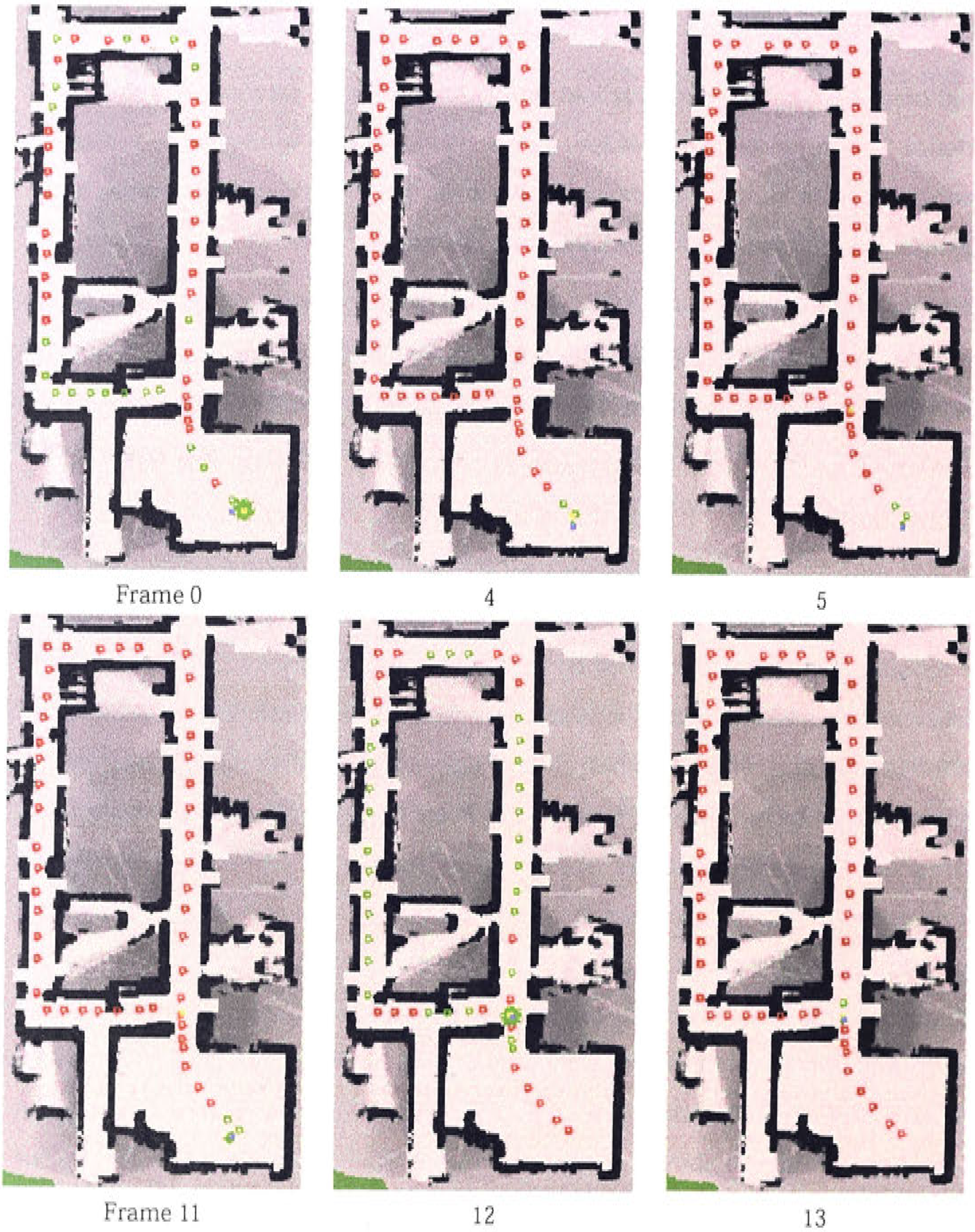
4. During the next 10 frames (frames $5-14$) perform localisation. The system should detect that local position estimation has been lost and attempt to re-localise and regain local position estimation.

The frame number refer to the frames prior to and after the kidnapping. Initially 5 frames are taken from the *start position* to ensure the robot is localised with a strong internal believe. After kidnapping and release at the *release position*, the robot must experience 5 continuous frames of low sensor model output to trigger relocalisation and another two to attempt global re-localisation. The remaining two frames allow the particle filter to start condensing about an estimate.

After frame 14 the identified MLP and the estimated local position within that place can be compared with the ground truth measures to evaluate the robot's performance in solving the kidnapped robot problem. In addition the benefit of constraining the global localisation search space can be determined by recording the size of the active place set during the relocalisation process.

Figure 9.30 shows the results of applying our localisation system to a kidnapped robot situation. In this situation the robot started in a location near place 1 at the bottom of the topological map, was kidnapped then released near place 9, which is close to the doorway between the large room and the corridor environment. In the images presented in the figure the localisation state of the system is shown after incorporating the sensor data for a given frame. In particular in each image, places in the topological map are shown by green circles for places within the current active place set and red for those without. The particle set distribution for the identified MLP is shown by the spread of green points, and the current estimated position as determined by the most likely particle is shown by the blue point. The yellow point represents the actual current location of the robot as determined by the ground truth measures. The frames shown in the figure correspond to the steps taken in solving the mobile robot problem outlines above. Frame 0 shows the system upon initial localisation about the *start position*; frames 4 and 5 before and after the kidnapping occurs; frames 11, 12 and 13 when the robot has detected a loss in position tracking relative to place 1, performs global localisation and redistributes the particle set about the new most likely place (place 9), and condenses about the new position estimate.

In this example our system correctly identified the most likely place of release after the kidnapping and the final position estimate was $15.62cm$ distant from the ground truth measure. The active place set during initial localisation was of size 21, and during the relocalisation process it reached size 31. This represents a global localisation search through 62% of the topological map during relocalisation. Although the estimated local position was accurate just 3 frames after relocalisation in this example, the con-



Frame 0

4

5

Frame 11

12

13

Figure 9.30: An example of the system solving an instance of the kidnapped robot problem. In all images, places in the active place set are drawn in green, the particle set is in green also, the estimated position in blue and the actual position in yellow. Frame 0 shows initial global localisation; frame 4: a strong position estimated has been achieved; frame 5: the robot has been kidnapped and released; frame 11 the robot has lost position tracking tracking; frame 12: the robot performs relocalisation; frame 13: the system has recovered position tracking.

Trials	MLP Identification (1^{st} Attempt)	Ave. Search Size	Ave. Error
100	84%	19 places (38%)	$29.12cm$

Table 9.2: System performance in the kidnapped robot experiment after the first attempt at re-localisation.

strained global localisation search space was still quite large and a further reduction on this is desirable. It should be noted however, that only during the initial localisation at frame 0 and relocalisation at frame 12 is the active place set this large. At all other times during local position position tracking the size of the active place set was constant at just 2 places.

9.6.2 Multi-Level Spatial Memory Localisation Performance

To further evaluate the ability of our system to solve the kidnapped robot problem and to constrain the global localisation search space, this experiment was repeated over 100 separate trials. For each trial the *start position* and the *release position* were chosen randomly from the 792 sample frames which were captured over the example path.

The results of performing localisation using the current system over the 100 trials are presented in Table 9.2. The table shows the percentage of trials in which the most likely place (MLP) was correctly identified, the average size of the active place set during relocalisation (the size of the active place set as a percentage of the total number of places in the topological map is shown in the brackets), and the average error in the local position estimate, for trials in which the most likely place was correctly identified, 3 frames after relocalisation occurs.

The 84% place identification is acceptable given that this is a first attempt at relocalisation. The failures occur when the robot is released in a position which lies between two places on the topological map. This causes ambiguities in the LRP measures and can led to incorrect identification of the MLP. In this case the system continues to detect a loss in position tracking which triggers relocalisation again and again until position tracking has been recovered. Eventual recovery from the kidnapped robot problem in this map is 100%. The worst case observed required three attempts before the MLP was correctly identified.

The size of the average active place set is comparable to that suggested by the local space experiments using a smaller map presented earlier, resulting in global localisation computation savings of approximately 60%. This is surprising giving the similarity of large extents of the current map and may represent a bound on potential saving using the local space matching approach with the current threshold levels. It holds sufficient promise that refinement of this method or the introduction of other cheap but ambiguous matching methods could further reduce the global localisation search space.

The average error in the resulting local position estimate of $29.12cm$ is high, however it must be remembered that this is only the third estimate after relocalisation has occurred and the local position tracking system could be still converging on a better estimate. Within the area surrounding a place's reference position there will be some fluctuation of achievable position estimation accuracy as the information available varies with distance from the reference position.

Our results indicate that the kidnapped robot problem is solvable and that the multi-level spatial memory representation can reduce the computation costs involved in re-localisation. The computation time required for the current system to perform position tracking once global localisation has been been performed is approximately $1.5s$. Global localisation can be achieved on average in approximately $13.5s$. These computation costs are still high for real time localisation for mobile robot navigation. Using range based sensors and a metric map, position tracking can be achieved in milliseconds, far outperforming our current system. During the global localisation task, however, the matching task grows prohibitively, increasing with each additional grid cell in the map space. In comparison, even with the expensive template matching of our current system, the ratio between computation time in global localisation and position tracking is very low:

$$\frac{13.5}{1.5} = 9$$

Reiterating the example from Chapter 2, consider a particle filter based localisation system using a metric map could perform position tracking with 1000 particles in

$10ms$. Given a grid cell granularity of $10cm$ and $1 \deg$, a $10 \times 10m$ map, roughly equivalent to the size of the current topological map, would require $3,600,000$ matches to perform an exhaustive global localisation search:

$$gridsize = 3,600,000 = 100 \times 100 \times 360$$

resulting in a computation time of $36s$ and a ratio of global localisation to local position estimation time of:

$$\frac{36,000}{10} = 3,600$$

If we conservatively estimate the node density of our topological graphs at $1m^2$, then an equivalent multi-level representation topological map would require 100 places to cover the same area. Global localisation in this map can be estimated at $27s$ still under that required by the metric map example. In regards to memory requirements, the metric map was estimated to require $3MB$ of memory. A map built using our system, that contains 100 places represented by visual landmarks and local space profiles would require:

$$(16 \times 16 \times 16 + 16) * 100 =\sim 0.4MB$$

bytes of storage (16 landmarks of size $16 \times 16pixels$), 16 value local space histogram, and 100 places, with a representation of a single place using $4KB$. If the high-level representation of disambiguating features is added to the representation, then the per place memory requirement is $\sim 47KB$; a 100 place map $\sim 4.7MB$. This exceeds that of the metric map, but this level of representation is only required in rare cases and can be stored on a physical storage medium.

Our system benefits from the scalability of topological based localisation system while still maintaining the local position accuracy of metric approaches.

Our system suffers from the computationally expensive process of matching landmark templates with the current visual scene. If a fast, repeatable method for pre-extracting landmarks from a visual scene is developed, this would eliminate the need for matching each landmark set with the entire current visual scene and greatly reduce the computation needed for global localisation.

Our results validate the multi-level mapping approach with representations of varying complexity to solve different aspects of the mobile robot localisation problem. A combination of accurate local position estimation and high place discrimination can be achieved in topological maps due to the low-level, but computationally expensive, representation of visual landmarks. The global localisation search space can be constrained by the comparatively computationally cheap but indiscriminating mid-level representation of local space profiles.

9.7 Summary

This chapter has described the application of the proposed multi-level spatial representation for mobile robots to the problem of global localisation. Methods for performing local position estimation and global localisation were presented and experimentation validating our approach was performed. In particular the following results were achieved:

- *Global Localisation using Place Discrimination over a Topological Map:* Place discrimination was achieved using the low-level representation of visual landmarks. A sensor image was classified as to which place it was captured in by evaluating the Landmark Recognition Performance (LRP) for each place's landmark set in the current image. Global localisation in a topological map can then be achieved by identifying the *most likely place* from which the current observation was made. In a path through a topological map of 50 places, place discrimination using visual landmarks correctly nominated the most likely place 100% of the time.

- *Combining Local Position Estimation and Global Localisation:* A method of detecting and recovering from the loss of local position tracking was introduced and validated experimentally. This method was used to combine global localisation and local position estimation to perform mobile robot localisation on a path through a non-trivial topological map (50 places). Although position estimation over the path was noisy and position tracking was often lost, the system always recovered and the final position estimate was within $15cm$ of the ground truth measure.

- *Constraint of Global Localisation Search Space with Local Space Profiles:* The second level of spatial representation can be used to restrict the global localisation search space by forming active place sets. These sets define the most likely places in the topological map where the robot is located. Experimental results showed that it was possible to reduce the global localisation computation time by up to 60% while maintaining the inclusion of the most likely place in the active place set 98% of the time. It was shown that the active place set can be applied to the problem of continuous global localisation and local position tracking.

- *Disambiguation of Similar Places:* The role of the third level of spatial representation was introduced and a possible application to the current system was discussed. Due to the computationally expensive nature and limited reliability of the implemented disambiguating features, this level of spatial representation was not actually integrated with the lower levels.

- *A solution to the Kidnapped Robot Problem:* The localisation system proposed in this thesis was applied to the notoriously difficult problem of the kidnapped robot as a measure of its ability to provide accurate local position estimation and efficient global localisation. In experimental results over a topological map with 50 places, it was found that the current system could recover from kidnapping 84% of the time with one attempt at relocalisation, and produce a local position estimate within $30cm$ of the ground truth measure within three frames of detection of position tracking loss. Given multiple attempts at relocalisation the system solved the kidnapped robot problem in this map 100% of the time. It was shown that although the computation involved in the method of landmark template matching is prohibitive for real time constraints, the current global localisation system does scale much better than a purely metric based approach.

The second goal of this thesis was to efficiently solve the global localisation problem while maintaining the robot systems' ability to perform accurate local position estimation. Global localisation was achieved as shown by the successful re-localisation observed in the kidnapped robot experiments. The low-level representation of visual landmarks provides a strong indicator when discriminating being places in a topological map. The mid-level of representation also provided a method for reducing the global localisation search space, and goes a long way in providing efficient global localisation. The savings in computation were achieved by using place representation with a cheap method of matching to initially constrain the search space before applying the more discriminatory but expensive matching method. While the results enforce the current multi-level approach to mobile robot localisation and are an improvement on the current approach, they fall short of the goal of efficient global localisation.

The major drawback in the global localisation process in this system is the nature of the visual landmark representation. The correlation method for matching landmark templates with the current panoramic image is too expensive to apply to the entire image space. This results in prohibitive computation costs when performing global localisation in even small topological maps, despite the reduction in search space size gained by the mid-level representation. Unfortunately our system will only approach acceptable real-time operation for global localisation when the computation involved in landmark matching method is drastically reduced. As stated in Chapter 4 this reduction in computation time, must not come at the expense of the landmark recognition performance.

While the landmark matching problem limits the performance of the system, the basic premise that a multi-level representation providing more efficient global localisation has been proven. Additional mid-level cues which are cheap to extract from the environment and match with the internal representation, such as the predominant colour in an image, could further reduce the global localisation costs. The topological representation also provides an obvious level of categorisation in the restriction of the global localisation search space. This discretisation of the global search space removes the need for expensive exhaustive searches over all possible robot positions, while providing a meaningful level of representation upon which to apply mid-level cues. The coarseness of the discretisation also removes the need for any probabilistic exploration of the topological search space, as the mid-level representation is computationally fast enough to apply to all places simultaneously, while the low-level representation is discriminatory enough to identify the correct place with a high measure of success. Add to this the simple rules for maintaining the active place set and the system's ability to detect and recover from loss of position tracking. The resulting system is capable of providing robust and reliable continuous global localisation and position tracking. These traits were demonstrated in the kidnapped robot experiment.

This chapter has described methods for performing the task of global localisation. The ability our system to perform global localisation has also been demonstrated. Our system can also solve the kidnapped robot problem by combining multi-levels of spatial representation to perform continuous local position estimation and global localisation. Although this was achieved outside of acceptable real time constraints, the contribution of multiple levels of spatial representation was demonstrated.

Chapter 10

Conclusions

Mobile robots require robust and reliable navigation systems if they are to join us in the real world. The stumbling block for robotic navigation systems has been in forming efficient and effective internal representations of the environment with which to perform localisation. Representations must be fine enough to allow for accurate local position estimation and coarse enough to permit global localisation. These two conflicting attributes form the core of the current limitations in mobile robot navigation. Much current research is devoted to combining representations in an attempt to overcome these limitations. Without such representations the localisation task is prohibitively expensive computationally, or inaccurate.

This thesis has proposed a multi-level representation of spatial knowledge for mobile robot navigation. This approach was inspired by biological navigation systems which combine different levels of navigation strategies to perform complex navigational tasks. This thesis argues that not only should robots combine multiple navigation strategies but that these strategies ought to be adopted to fit multiple levels of spatial representation as well. Multi-level representations allow for the accuracy provided by low-level approaches while constraining the localisation search with associated higher level representations. This approach, of building higher levels of representation and functionality ontop of existing levels was inspired by biological studies of navigation systems in honeybees, rats and humans.

At a low level, unique visual landmarks were used to represent places on a topological map. Visual landmarks were selected via an automatic static and dynamic selection process which was inspired by the flights of wasps. Performing a TBL movement during place acquisition also allows the estimation of landmarks depths using a form of bearing only SLAM.

The second level of representation is that of local space profiles. A sense of open space in the environment can direct the global localisation search to places known to share such an extent of open space. A histogram representation of the extent of local space surrounding the robot is formed by detecting the ground plane through carpet matching techniques. An occupancy grid in image space is used to promulgate carpet region probability measures through time and across sensor modalities.

The third level is that of disambiguating features. These are used to directly compare the panoramic images at two locations and identify images regions which differentiate between them. This process can help localisation in problematic environments where similar places exist.

The three levels of spatial representation are used to represent places in a topological map. Topological maps are automatically constructed by monitoring the background levels of LRP. Once LRP performance drops to background levels, the robot acquires a new place and connects it to the existing map using transitions. Cycles in the topological map can be detected, and a method for defining transitions in the presence of cycles was developed.

Geometric models of landmark sensor observations were developed and a particle filter method of local position estimation was implemented. Using the visual landmark representation, accurate local position estimation within places was achieved due to the estimated landmark depth and the associated sensor model. Position estimates are passed between places using the information contained in transitions and position tracking over a cyclical path was demonstrated in this fashion.

The uniqueness of the visual landmark level of spatial representation provides good place discrimination ability. Global localisation can be achieved by matching the current image against the landmark sets of all places in the topological map. Unfortunately the computation costs involved in matching with landmark sets make an exhaustive global localisation search prohibitively expensive. The mid-level spatial representation of local space profiles are compact, computationally inexpensive to match, and can be used to constrain the global localisation search. Use of local space profiles in this fashion resulted in a 60% saving in landmark matching when performing global localisation.

By defining a method of detecting when local position tracking has been lost, the multi-level spatial representation was used to perform continuous global localisation and local position estimation. This complete system was then applied to the kidnapped robot problem, with good results.

Chapter 1 outlined three principle objectives of this research:

1. Solve the conflicting localisation problems of global localisation and local position estimation.
2. Maintain a central belief as to the robot's position within an internal map.
3. Solve the kidnapped robot problem.

The localisation system developed in this research uses the multi-level spatial representation to achieve these objectives.

The conflicting global localisation and local position estimation problem was resolved by the targeting of representation levels to the functional and computational requirements of the two tasks. The visual landmark representation level provided the necessary positional and discriminatory ability to perform accurate local position estimation and global localisation. The local space profile representation level provided the a computationally inexpensive way of constraining the global localisation search space. The implementation of the visual landmark representation reduced the efficiency of the system, however the concept of targeting multiple levels of spatial representation to the conflicting localisation task was demonstrated.

A central belief of the robot's position was maintained through the use of active place sets in the topological map and particle filter position estimation within places of the topological map. The *Most Likely Place* that the robot is occupying is identified by local space and landmark set matching while the exact position within a place is identified by the most probable particle. The belief of robot location is spread between the levels of gross topology and the relative reference frames within each place. Methods for monitoring the certainty of this belief and forming and switching alternative hypotheses in the face of conflicting sensory input were developed. Hypotheses formation and switching occurs at both levels of belief, although at the relative reference level, hypotheses are restricted to within the current most likely topographic region (the most likely place).

The multi-level spatial representation solved the kidnapped robot problem by using the representation's ability to perform continuous global localisation and position estimation and to maintain a central belief of it's location.

The overall philosophy of multiple levels of representation providing a basis for multiple layers of localisation and navigation strategies was not fully embodied by the developed system. The high-level representation level proved difficult to implement reliably and was not integrated with the rest of the system. In the limited semantic environment of mobile robot perception, it is not easy to pick out disambiguating visual features without more complex scene analysis. Although we believe that this level of representation is essential for robust robot navigation, this research did not develop it to a level where it provides useful information to the localisation task.

Another discrepancy from the multi-level philosophy appears when analysing what each level of representation contributes to the localisation task. In this research, the low-level representation of visual landmarks provided the majority of the localisation information at the topological and sub-topological level. The mid-level representation served only to restrict the global localisation search within the topological level. In a true multi-level approach, the localisation work-load should be more evenly spread, with the local space level of representation providing more information to the topological and even perhaps the sub-topological level. This is not meant to disregard the need for hierarchies in both representation levels and the localisation task, simply to highlight the possibility that each level of representation could contribute to multiple levels of the localisation task.

The multi-level spatial representation developed in this research has been successfully applied to the domain of mobile robot localisation. The multiple levels of representation have provided the necessary information to solve contemporary problems in mobile robot localisation such as the conflict between global localisation and local position estimation and the kidnapped robot situation. While the current implementation of a multi-level spatial representation based localisation system can not perform global localisation within even soft real time constraints, the contributions of the various levels do reduce the computational complexity.

10.1 Future Work

The road goes ever on and on. This research leaves many openings for future investigation. In the areas where this research is successful, further refinement of the implementation details and the concepts involved are needed. In the areas where this research falls short, total revision is in order.

Avenues for future work specific to the multi-level representation system reported in this research have been mentioned in the relevant chapter summaries. What follows here are some broad possibilities for future research in multi-level representations for mobile robot localisation.

This research built topological maps based on the concept of places defined by a low level of representation. Consequently the topography defined by these maps is relatively fine grained. Research is needed to define what level of representation should be used to drive the construction of topological maps. A possibility is to build maps with multi-levels of representation with appropriate linkages between them. This would be consistent withe the multi-level approach to representation espoused by our research. This would lead to coarser topologies and subsequently reduced complexity of the global localisation task.

Throughout this research the dual role of the visual landmark representation in the localisation has intrigued the authors. Visual landmarks are used as a low-level representation for accurate position information and at a higher level for topological discrimination. The distinctiveness of landmarks derives from their richness and adds to the computation complexity of their use. Visual uniqueness of landmarks in the global environment is desirable when performing global localisation, while visual uniqueness within the current scene is desirable for accurate local positioning. It would be interesting to investigate the role of landmarks in these two roles and how matching complexity could be managed by recognising such a demarcation of responsibilities.

An obvious follow on from our research is to develop other modalities of lightweight spatial representations to contribute to the goal of constraining the global localisation task. A multitude of computationally inexpensive and comparatively indisciminatory representations could be used in conjunction to further reduce the global localisation search space for a richer, more discriminatory representation.

Finally, the further development of a high level representation for active place discrimination would be interesting. Only the active search for disambiguating features will lead to reliable localisation. It is interesting that human ability to perform complex spatial reorientation tasks is linked to the onset of linguistic and reasoning capabilities. A topic for future work would be to investigate just what functional, semantic and symbolic knowledge is required to perform such tasks.

The future work described here is addressed to the same underlying concepts as this research as a whole: the representations of spatial knowledge that are essential for the navigation tasks of mapping and localisation. This thesis has proposed a multi-level representation of spatial knowledge and applied it to the problem of mobile robot localisation with promising results. Multiple representations of spatial knowledge facilitate the solving of the localisation problem.

Bibliography

Arkin, R. (1998). *Behaviour Based Robotics*, MIT Press.

Asoh, H. and Matsui, T. (1999). A unified framework of map learning with a hierarchy of probabilitic maps, *Proceedings of 1999 International Conference on Field and Service Robotics*, pp. 86–91.

Bachelder, I. A. and Waxman, A. M. (1995). A view-based neurocomputation system for relational map-making adn navigation in visual environments, *Robots and Autonomous Systems* **16**: 267–289.

Bennett, A. T. D. (1996). Do animals have cognitive maps?, *The Journal of Experimental Biology* **199**: 219–224.

Bianco, G. and Zelinsky, A. (1999). Biologically inspired visual landmark learning and navigation for mobile robots, *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS '99)*, Vol. 2, pp. 671–676.

Borenstein, J., Everett, H. R. and Feng, L. (1996). *Navigation Mobile Robots: Systems and Techniques*, A. K. Peters Ltd.

Burgard, W., Fox, D., Henning, D. and Schmidt, T. (1996). Estimating the absolute position of a mobile robot using position probability grids, *Proceedings of Naitional Conference on Artifical Intelligence (AAAI'96)*.

Burgess, N., Donnett, J. G., Jeffrey, K. J. and O'keefe, J. (1997). Robotic and neuronal simulation of the hippocampus and rat navigation, *Philosophical Transations of the Royal Society of London: Biological Systems* **352**: 1535–1543.

Burgess, N., Reece, M. and O'Keefe, J. (1994). A model of hippocampal function, *Neural Networks* **7**: 1065–1081.

Cao, Z., Oh, S. and Hall, E. (1986). Dynamic omnidirectional vision for mobile robots, *Journal of Robotic Systems* **3**(1): 5–17.

Chahl, J. and Srinivasan, M. (1997). Reflective surfaces for panoramic vision, *Journal of Applied Optics* **36**(31): 8275–8285.

Cheng, G. and Zelinsky, A. (1996). Real-time visual behaviours for navigating a mobile robot, *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

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

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., Fry, S. N. and Wehner, R. (1993). Sequence learning in honeybees, *The Journal of Comparative Psychology* **172**: 693–706.

Collett, T. and Zeil, J. (1996). Flights of learning, *Journal of the American Psychological Society* pp. 149–155.

Conroy, T. L. and Moore, J. B. (1999). Resolution invariant surfaces for panoramic vision systems, *Proceedings of the IEEE International Conference on Computer Vision (ICCV'99)*, pp. 392–397.

Coombs, D. and Roberts, K. (1992). Centering behvaiour using peripheral vision, *IEEE Conference on Computer Vision and Pattern Reocgnition*, pp. 440–451.

Courtney, J. and Jain, A. (1994). Mobile robot localisation via classification of multisensor maps, *IEEE International Conference on Robotics and Automation*, pp. 1672–1678.

Davidson, A. and Murray, D. (2002). Simultaneous localisation and map-building using active vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**: 865–880.

Deans, M. and Hebert, M. (2000). Experimental comparison of techniques for localization and mapping using a bearingonly sensor, *International Conference on Experimental Robotics*.

Dellaert, F., Burgard, W., Fox, D. and Thrun, S. (1999). Using the condensation algorithm for robust, vision-based mobile robot localization, *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*.

Dellaert, F., Fox, D., Burgard, W. and Thrun, S. (1999). Monte carlo localization for mobile robots, *Proceedings of 1999 International Conference on Robotics and Automation*.

Dissanayake, G., Durrant-Whyte, H. and Bailey, T. (2000). A computationally efficient solution to the simultaneous localisation and map building (slam) problem, *Proceedings of the IEEE Internation Conference on Robotics and Automation*, pp. 1009–1014.

Drocourt, C., Delahouche, L., Marhic, B. and Cleretin, A. (2002). Simultaneous localisation and map construction method using omnidirectional steroscopic information, *Proceedings of IEEE International Conference on Robotics and Automation*.

Durrant-Whyte, H. (1994). Where am i?, *Industrial Robotics* **21**: 11–16.

Eichenbaum, H., Stewart, C. and Morris, R. (1990). Hippocampal representation in place learning, *Journal of Neuroscience* **10**: 3531–42.

Esch, H., Zhang, S., Srinivasan, M. and Tautz, J. (2001). Honeybee dances communicate distances measured by optic flow, *Nature (Lond)* **411**: 581–583.

Franz, M. O. and Mallot, H. A. (2000). Biomimetic robot navigation, *Robots and Autonomous Systems* **30**: 133–153.

Franz, M. O., Scholkopf, B., Mallot, H. A. and Bulthoff, H. H. (1998). Learning view graphs for robot navigation, *Autonomous Robots* **5**: 111–125.

Gaskett, C., Fletcher, L. and Zelinsky, A. (2000). Reinforcement learning for a vision based mobile robot, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Gaspar, J., Winters, N. and Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omnidirectional camera, *IEEE Transactions on Robotics and Automation* **16**(6): 890–898.

Gaussier, P., Joulain, C., Zrehen, S., Banquet, J. P. and Revel, A. (1997). Visual navigation in an open environment without map, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 545–550.

Hermer-Vazquez, L., Moffet, A. and Munkholm, P. (2001). Language, spacce, and the development of cognitive flexibility in humans: the case of two spatial memory tasks, *Cognition* **79**: 263–299.

Hong, J., Tan, X., Pinette, B., Weiss, R. and Riseman, E. (1991). Image-based homing, *Proceedings of IEEE International Conference on Robotics and Automation*.

Horswill, I. (1993). Polly: A vision-based artificial agent, *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*.

Isard, M. and Blake, A. (1998). Condensation – conditional density propagation for visual tracking, *International Journal of Computer Vision* **29**(1): 5–28.

Jensfelt, P. (2001). *Approaches to Mobile Robot Localization in Indoor Environments*, PhD thesis, Department of Signals, Sensors and Systems, Royal Insititute of Technology, Sweden.

Jensfelt, P. and Christensen, H. (1999). Laser based pose tracking, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'99)*, Vol. 4, pp. 2994–3000.

Jensfelt, P., Wijk, O., Austin, D. and Andrsson, M. (2000). Experiments on augmenting condensation for mobile robot localization, *IEEE Intl. Conf. on Robotics and Automation*, pp. 2518–2524.

Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots, *Proceedings of IEEE International Conference on Robotics and Automation*.

Krose, B., Vlassis, N., Bunschoten, R. and Motomura, Y. (2001). A probabilistic model for appearance-based robot localization, *Image and Vision Computing* **19**: 381–391.

Kuipers, B. and Byun, Y. (1991). A robot exploration and mapping strategy based on a sematic hierarchy of spatial representations, *Journal of Robots and Autonomous Systems* **8**: 47–63.

Lambrinos, D., Moller, R., Labhart, T., Pfeifer, R. and Wehner, R. (2000). A mobile robot employing insect strategies for navigation, *Robots and Autonomous Systems* **30**: 39–64.

Latombe, J. C. (1991). *Robot Motion Planning*, Kluwer Academic Publishers, Boston.

Lehrer (1993). Why do bees turn back and look?, *Journal of Comparative Physiology* **172**(A): 549–563.

Leonard, J. and Durrant-Whyte, H. (1991a). Mobile robot localization by tracking geometric beacons, *Robotics and Automation* **7**: 376–382.

Leonard, J. and Durrant-Whyte, H. (1991b). Simultaneous map building and localisation for an autonomous mobile robot, *Proceedings of IEEE International Workshop on Intelligent Robots and Systems*.

Levitt, T. and Lawton, D. (1990). Qualitative navigation for mobile robotics, *Artificial Intelligence* **44**: 305–360.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to setero vision, *Proceedings of Seventh International Joint Conference on Artificial Intelligence*.

Mallot, H., Bulthoff, H., Georg, P., Scholkopf, B. and Yasuhara, K. (1995). View based cognitive map learning by an autonomus robot, *Internationnal Conference on Artifical Neural Networks*, Vol. 2, pp. 381–386.

Matsui, T., Thompson, S. and Asoh, H. (2000). Mobile robot localization using circular correlations of panoramic images, *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS '00)*.

Matsumoto, Y., Inaba, M. and Inoue, H. (1997). Memory-based navigation using omniview sequence, *Int. Conf. of Field and Service Robotics*, pp. 184–191.

Matsumoto, Y., Inaba, M. and Inoue, H. (2000). Exploration and navigation in corridor environment based on omni-view sequence, *Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2000)*, pp. 1505–1510.

Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control*, Academic Press.

Moravec, H. (1977). Towards automatic visual obstacle avoidance, *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Vol. Vision-1, p. 584.

Moravec, H. and Elfes, A. (1985). High resolution maps from wide angle sonar, *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 116–121.

Mori, T., Matsumoto, Y., Shibata, T., Inaba, M. and Inoue, H. (1995). Trackable attention point generate based on classification of correlation value distribution., *JSME Annual Conf. on Robotics and Mechatronics (ROBOMEC 95).*, pp. 1076–1079.

O'Keefe, J. and Burgess, N. (1996). Geometric determinants of the place fields of hippocampal neurons, *Nature* **381**: 425–428.

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

Oliveira, M., Bueno, O., Pomarico, A. and Gugliano, E. (1997). Strategies used by hippocampal and caudate-putamen lesioned rats in a learning task, *Neurobiological of Learning and Memory* **68**: 32–41.

Paletta, L., Frintrop, S. and Hertzberg, J. (2001). Robust localisation using context in omnidirectional imaging, *Proceedings of IEEE International Conference on Robotics and Automation*.

Press, S. J. (1989). *Bayesian Statistics: Principles, models and applications*, John Wiley and Sons, New York.

Ramos, J. M. J. (2000). Influence of the shape of the experimental room on spatial learning in rats, *Physiology and Behaviour* **70**: 351–357.

Sim, R. and Dudek, G. (1999). Learning visual landmarks for pose estimation, *Proc. of the IEEE International Conference on Robotics and Automation*.

Smith, R., Self, M. and Cheeseman, P. (1987). A sotchastic map for uncertain spatial relationships, *4th Internation Symposium on Robotics Research*.

Srinivasan, M. V., Chahl, J. S., Weber, K., Venkatesh, S., Nagle, M. G. and Zhang, S. W. (1999). Robot navigation inspired by principles of insect vision, *Robotics and Autonomous Systems* **26**: 203–216.

Srinivasan, M. V., Zhang, S. W. and Bidwell, N. J. (1997). Visually mediated odometry in honeybees, *The Journal of Experimental Biology* **200**: 2513–2522.

Stepan, P. and Kurlich, M. (2001). Building maps for autonomous mobile robots, *3rd International Conference on Field and Service Robotics*.

Strelow, D., Mishler, J., Singh, S. and Herman, H. (2001). Extending shape-from-motion to non-central omnidirectional cameras, *Proceedings of IEEE International Conference on Intelligent Robots and Systems IROS'01*.

Thrun, S. (1998). Bayesian landmark learning for mobile robot localisation, *Machine Learning*, Vol. 33.

Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A., Dellaert, F., Fox, D., ahnel, D., Rosenberg, C., Roy, N., Schulte, J. and Schulz, D. (2000). Probabilistic algorithms and the interactive museum tour-guide robot minerva, *International Journal of Robotics Research* **19**.

Thrun, S., Burgard, W. and Fox, D. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots, *Autonomous Robots* **5**: 253–271.

Thrun, S., Burgard, W. and Fox, D. (2000). A real time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping, *Proceedings of IEEE International Conference on Robotics and Automation*.

Thrun, S., Gutmann, J., Fox, D., Burgard, W. and Kuipers, B. (1998). Integrating topological and metric maps for mobile robot navigation: A statistical approach, *Proceedings of AAAI-98*.

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

Trullier, O., Wiener, S., Berthoz, A. and Meyer, J. (1997). Biologically based artificial navigation systems; review adn prospects., *Progress in Neurobiology* **51**: 483–544.

Uhlmann, J. (1998). *Dynamic Map Building and Localisation: New Theoretical Foundations*, PhD thesis, Robotics Research Group, Department of Engineeering Science, University of Oxford.

Ulrich, I. and Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization, *Proceedings of IEEE International Conference on Robotics and Automation.*

Vlassis, N., Bunschoten, R. and Krose, B. (2001). Learning task-relevant features from robot data, *Proceedings of IEEE International Conference on Robotics and Automation.*

Vlassis, N. and Krose, B. (1999). Robot environment modelling via principle component regression, *Proceedings of IEEE International Conference on Intelligent Robots and Systems IROS'01.*

Vlassis, N., Terwijn, B. and Krose, B. (2002). Auxilliary particle filter robot localisation from high-dimensional senor observations, *Proceedings of IEEE International Conference on Robotics and Automation.*

Yagi, Y., Hamada, H., Benson, N. and Yachida, M. (2000). Generation of stationary environmental map under unkown robot motion, *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS '00).*

Yagi, Y. and Kawato, S. (1990). Panoramic scene analysis with conic projection, *Proceedings of the 1990 IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS '90).*

Yamazawa, K., Yagi, Y. and Yachida, M. (1995). Obstacle detection with omnidirectional image sensor hyperomni vision., *IEEE International Conference on Robotics and Automation*, pp. 1062–1067.

Zeil, J., Kelber, A. and Voss, R. (1996). Structure and function of learning flights in bees and wasps, *The Journal of Experimental Biology* **199**: 245–252.

Zhang, S. W., Bartsch, K. and Srinivasan, M. V. (1996). Maze learning by honeybees, *Neurobiology of Learning and Memory* **66**: 267–282.

Zhang, S. W., Lehrer, M. and Srinivasan, M. V. (1999). Honeybee memory: Navigation by associative grouping and recall of visual stimuli, *Neurobiology of Learning and Memory* **72**: 180–201.

List of Publications

Matsui, T., Thompson, S. and Asoh, H. (2000). Mobile robot localization using circular correlations of panoramic images, *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS '00).*

Thompson, S., Matsui, T. and Zelinsky, A. (2000). Localisation using automatically selected landmarks from panoramic images, *Proceedings of Australian Conference on Robotics and Automation.*

Thompson, S. and Zelinsky, A. (2002). Accurate local positioning using visual landmarks from a panoramic sensor, *Proceedings of IEEE International Conference on Robotics and Automation.*

Thompson, S., Zelinsky, A. and M.V.Srinivasan (1999). Automatic landmark selection for navigation with panoramic vision, *Proceedings of Australian Conference on Robotics and Automation.*