ON VISUAL MAPS AND THEIR AUTOMATIC CONSTRUCTION

Robert Sim

Department of Computer Science
McGill University, Montréal

17 February 2004

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

Your file Votre référence
ISBN: 0-612-98372-2
Our file Notre référence
ISBN: 0-612-98372-2

ABSTRACT

This thesis addresses the problem of automatically constructing a visual representation of an unknown environment that is useful for robotic navigation, localization and exploration. There are two main contributions. First, the concept of the *visual map* is developed, a representation of the visual structure of the environment, and a framework for learning this structure is provided. Second, methods for automatically constructing a visual map are presented for the case when limited information is available about the position of the camera during data collection.

The core concept of this thesis is that of the visual map, which models a set of image-domain features extracted from a scene. These are initially selected using a measure of visual saliency, and subsequently modelled and evaluated for their utility for robot pose estimation. Experiments are conducted demonstrating the feature learning process and the inferred models' reliability for pose inference.

The second part of this thesis addresses the problem of automatically collecting training images and constructing a visual map. First, it is shown that visual maps are self-organizing in nature, and the transformation between the image and pose domains is established with minimal prior pose information. Second, it is shown that visual maps can be constructed reliably in the face of uncertainty by selecting an appropriate exploration strategy. A variety of such strategies are presented and these approaches are validated experimentally in both simulated and real-world settings.

RÉSUMÉ

Cette thèse adresse le problème de construire automatiquement une représentation visuelle d'un environnement inconnu qui soit utile pour la navigation, la localisation et l'exploration robotique. Il y a deux contributions principales. Premièrement, le concept de carte visuelle, une représentation de la structure visuelle d'un environnement, est développé, et une méthodologie pour apprendre cette structure est fournie. Deuxièmement, plusieurs méthodes pour construire automatiquement une carte visuelle sont presentées, pour le cas où l'information disponible sur la position de la caméra pendant la collecte de données est limitée.

L'idée centrale de cette thèse est celle de la carte visuelle qui modélise un ensemble de caractéristiques du domaine de l'image, extraites à partir d'une scène. Celles-ci sont initialement choisies à partir d'une mesure de salience visuelle, et ensuite modelésisées et évaluées pour leur utilité quant à l'évaluation de la pose du robot. Des expériences sont presentées, montrant l'apprentissage des caractéristiques et la fiabilité des modèles impliqués pour l'inférence de la pose d'un robot.

La deuxième partie de cette thèse adresse le problème de rassembler automatiquement des images échantillons et de construire une carte visuelle. Tout d'abord, il est démontré que les cartes visuelles s'organisent automatiquement par nature. De plus, la transformation entre l'image et le domaine de la pose est etablie avec une information préalable minimale sur la pose. Ensuite, nous démontrons comment les cartes visuelles peuvent tre construites de manière robuste malgrè l'incertitude de l'environement en choisissant une stratégie d'exploration appropriée. Un nombre de

ces stratégies sont présentées et ces approches sont validées expérimentalement dans un contexte simulé et réel.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support and encouragement of friends, colleagues and family. In particular, I want to extend my gratitude to Greg Dudek, my supervisor and friend of seven years, who has seen me through two theses and a variety of stimulating projects. Greg has that rare ability to extract academic excellence while maintaining an open and approachable relationship with his students. I would not have chosen the field of robotics had it not been for Greg's contagious enthusiasm for the subject, and I would have not completed the work had it not been for his persistent faith in my abilities.

I would also like to extend my appreciation to the various members of my committee and other faculty members at McGill who have provided invaluable feedback about my work. In particular, Frank Ferrie, Kaleem Siddiqi, Mike Langer, Tal Arbel, Sue Whitesides and Doina Precup have all made a positive mark on this thesis. My appreciation goes also to the past and present support staff at the Centre for Intelligent Machines, with whom it has always been a pleasure to work: Jan Binder, Ornella Cavalliere, Marlene Gray, Cynthia Davison, Danny Chouinard, and Mike Parker.

I would also like to acknowledge my sources of funding. The generous support of the Canadian Natural Sciences and Engineering Research Council, the Canadian Space Agency, IRIS/Precarn and Hydro Québec made all of this possible.

My friends and lab-mates at CIM over the years have provided the kind of community support that every Ph.D. student needs; Eric Bourque, Sylvain Bouix, Saul Simhon, Abril Torres, Ioannis Rekleitis, Scott Burlington, Deeptiman Jugessur,

Richard Unger, Francois Belair, Sandra Polifroni, Nick Roy and Mike Daum have all played key roles in maintaining my sanity. Sylvain Bouix also takes credit for providing the French translation of the Abstract. I owe special thanks as well to Ravi Bhat and Finlay MacNab, fellow students and close friends who understand how difficult a doctorate can be.

Last, but not least, I owe Nisha an enormous debt of gratitude for her love and patience throughout this experience. Being a student puts a special kind of strain on a relationship and she has endured it with grace. I hope I can provide the same support as she embarks on her own doctoral adventure.

This thesis is for Maya, my angel.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1. The Visual Map: A Robot's "Mind's Eye"

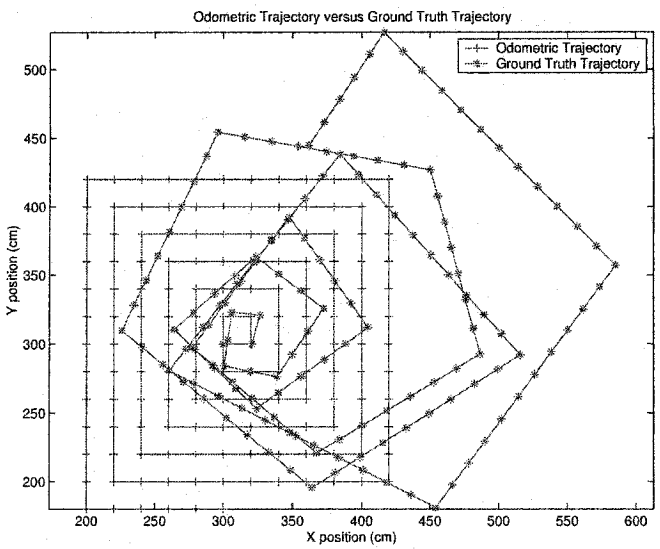
This thesis presents a method to enable a robot to develop its "mind's eye". That is, it addresses the problem of representing the visual world, and using that representation to interact with the environment. In particular, we will apply the visual representation, otherwise known as a *Visual Map*, to the task of robotic localization– that is, allowing a robot to answer the question "Where am I?", by comparing what it sees in the world with its "mind's eye" representation of what it expects to see. An answer to the "localization problem" is a fundamental building block in solving the problems of robotic navigation and autonomy.

An important goal in developing a visual representation will be that of solving the problem of how a robot can automatically *learn* such a representation without human assistance. While in some instances a visual representation could be supplied by a human operator (such as a computer-aided design (CAD) drawing of a building or a road map), there are a variety of contexts in which a human-built representation is insufficient. For example, CAD drawings do not explicitly represent the presence of furniture, road maps will not mark lanes or routes closed for construction, and the satellite-based global positioning system (GPS) will not operate indoors, under water, or at large distances from Earth.

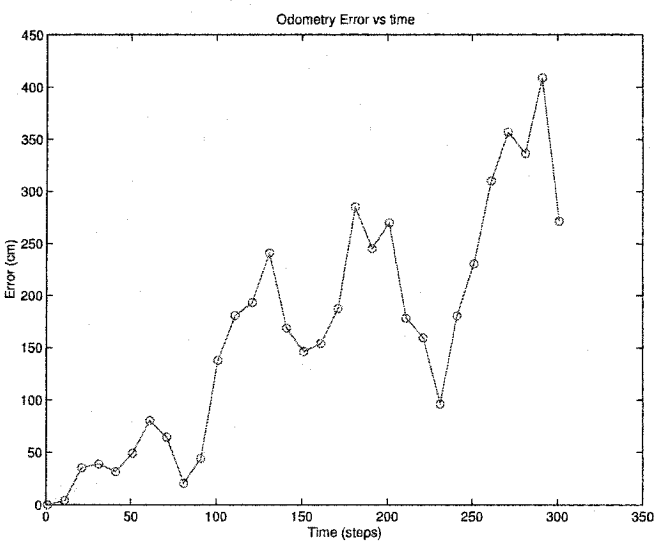
Beyond producing the ability for a robot to navigate using vision, there are a wide variety of applications where producing a visual representation of the environment is important. For example, automatic inspection of hazardous environments, such as radioactive waste disposal facilities, deep-sea salvage and monitoring operations, and planetary exploration all depend on the transmission of meaningful visual information. In each of these scenarios, there is not only a need for the robot to construct the representation, but also to explore the environment autonomously.

1.1. The Localization Problem. In constructing a visual representation, our primary goal is to enable a robot to estimate its position in the environment. The remainder of this section will provide an overview of the state of the art in position estimation methods and motivate the design choices that constitute the visual map representation.

The localization problem is important to a robotic agent for a variety of reasons, not the least of which is that of the need to know where it is before it can decide how to get where it wants to go. A naive approach to measuring robot position involves integrating information from odometers that measure wheel rotations or joint motions. However, odometers have no model of the external world and cannot measure the local effects of wheel slippage or the global effects of transport due to maintenance or other events. Even under highly controlled circumstances, over time a pose estimate derived from odometry can drift away from the true pose such that the error grows without bound. Figure 1.1 illustrates this problem using a simple simulation of a robot whose actions result in normally distributed, zero-mean, 2% variance error at each time step as it attempts to trace a series of concentric squares (the trajectory traced with a series of '+' symbols). The actual trajectory taken by the robot is traced with a series of '*' signs. Clearly, the robot's odometry-based representation of position diverges rapidly from the true trajectory. This divergence is quantified in the lower graph, which plots absolute error between the two trajectories versus time. Over time, this divergence can grow without bound.



(a)



(b)

FIGURE 1.1. a) Odometric trajectory (+) vs actual trajectory (*) for a simulated robot. b) Absolute deviation of odometry from the true trajectory as a function of time. As time progresses, the deviation can grow without bound.

In order to correctly localize in the world, humans resort to external references. For example, as they approach a door, humans judge their relative position to it, employ hand-eye coordination to reach for the door-knob, and ensure that they have passed through the door before they close it behind them. In a larger environment, they might judge their position relative to the edges of a sidewalk, their distance along the sidewalk and the particular street that they are following, all using external cues. Finally, at a much larger scale, they might measure bearings to distant landmarks, such as buildings and trees, mountains, or even stars, to determine their place in the world.

Similarly, a robot must sense external cues or features of the environment and determine its position from them. There is an almost infinite variety of features to select from and it is worthwhile to consider the most common selections used by modern robotic platforms. Among these are features derived from range- or distance-sensing devices, the detection of artificial landmarks, and features dependent on specific assumptions about sensor configuration and the properties of the world.

There is an extensive body of research on position estimation from geometrically determined landmarks. Such landmarks are usually sensed using range sensors that return distance and bearing to each landmark, and estimating pose is a simple matter of triangulation [**136, 2, 137, 68, 78, 74, 83, 36**]. However, extracting and matching corresponding landmarks from a range sensor is a non-trivial task, even to the point that most early work on the problem assumed known correspondence. Furthermore, most range sensors, such as laser range-finders and sonar transducers are active or energy-emitting sensors, emitting sound or light into the environment, which is an undesirable property in a wide variety of applications. In addition, active range sensors can be monetarily expensive devices. Passive range sensors, such as stereo cameras, require careful calibration and depend on assumptions concerning object reflectance in order to achieve accurate pixel correspondence between images [**49, 50, 51, 56, 62, 87, 92, 156**].

As an alternative to determining a set of natural features of the environment, some approaches install a set of artificial landmarks and use them for pose estimation. Perhaps the best known example of pose estimation from artificial landmarks is the satellite-based Global Positioning System (GPS), which enables an object on Earth to determine its position using time-of-flight information and triangulation based on radio signals from a network of satellites in orbit at an altitude of about 20,000km [48]. Other examples of artificial landmarks include bar codes, specialized targets, embedded magnetic tracks, or coloured markers [71, 9, 52, 106]. In some instances, the robot will deploy these targets as it explores an unknown environment, but in most cases the robot is dependent on a human operator installing them *a priori*. The chief disadvantage of such systems is the requirement of a human presence, which is largely impossible in many contexts, such as space or undersea exploration, search and rescue in a danger zone, or other hazardous environments, as well as the constraints imposed by landmark visibility. For example, GPS fails when line of sight is not maintained with at least four orbiting satellites.

Finally, a small class of pose estimation approaches exploit special assumptions about the world, or the sensor. For example, autonomous vehicles that can navigate roadways make domain-specific assumptions about the structure of the roadway and the behaviours of other vehicles on the road [141]. Other methods exploit special environmental properties and camera configurations, such as a map constructed as a mosaic of the ceiling in an indoor environment [20]. Several approaches, such as photogrammetry or methods employing omnidirectional cameras, make concrete assumptions about the imaging geometry and often rely on careful calibration of the apparatus [149, 85]. A useful discussion of all of the approaches mentioned here can be found in Dudek and Jenkin [27].

1.2. Local Features. A core idea in this thesis is that locally computed image features, as opposed to global image properties, are best suited to solving inference tasks. Local features provide resistance to the dynamics of operational environments, where moving people, furniture and illumination can all affect the sensor image and

corrupt globally derived measures. One goal of this work is the inference of which local image features are most stable and resistant to changes in the scene. There are also arguments to be made for improved computational efficiency when employing local features. This fact is evident in biological vision systems which attend to a small set of features in the world in order to construct an internal representation [58].

1.3. Probabilistic Reasoning. The vast majority of position estimation systems take a probabilistic approach to representing both sensor outputs and the computed position estimate. Among the most common representations are particle filters, Kalman Filters, and grid decompositions, exemplified by the works of Dellaert, *et al.*, Smith, *et al.*, and Moravec, respectively [**21, 132, 83**]. The reasons for taking a probabilistic approach are grounded in the fact that the world and the sensors that operate in it are stochastic in nature, and that the same observations from the same position are rarely repeated. Furthermore, it is often the case that different parts of the world will produce similar observations– consider similarities between trees of the same species, intersections in downtown Manhattan, the rooms in a hotel, and the aisles in a warehouse. In practice, resolving ambiguities in a robot's position is a hard problem [**29**], and when a robot's position is estimated it is important to provide information about how precise that estimate is and whether alternative hypotheses exist about where the robot might be.

The current inventory of pose estimation solutions and their related environment representations suggests the enumeration of a set of open problems with respect to methodology and representation. In particular, this thesis aims at developing a representation and approach to the problem that

(i) employs a passive sensor, such as a camera,

(ii) does not depend on assumptions about the structure or other properties of the environment,

(iii) does not depend on explicit, and often expensive, calibration of the sensing apparatus,

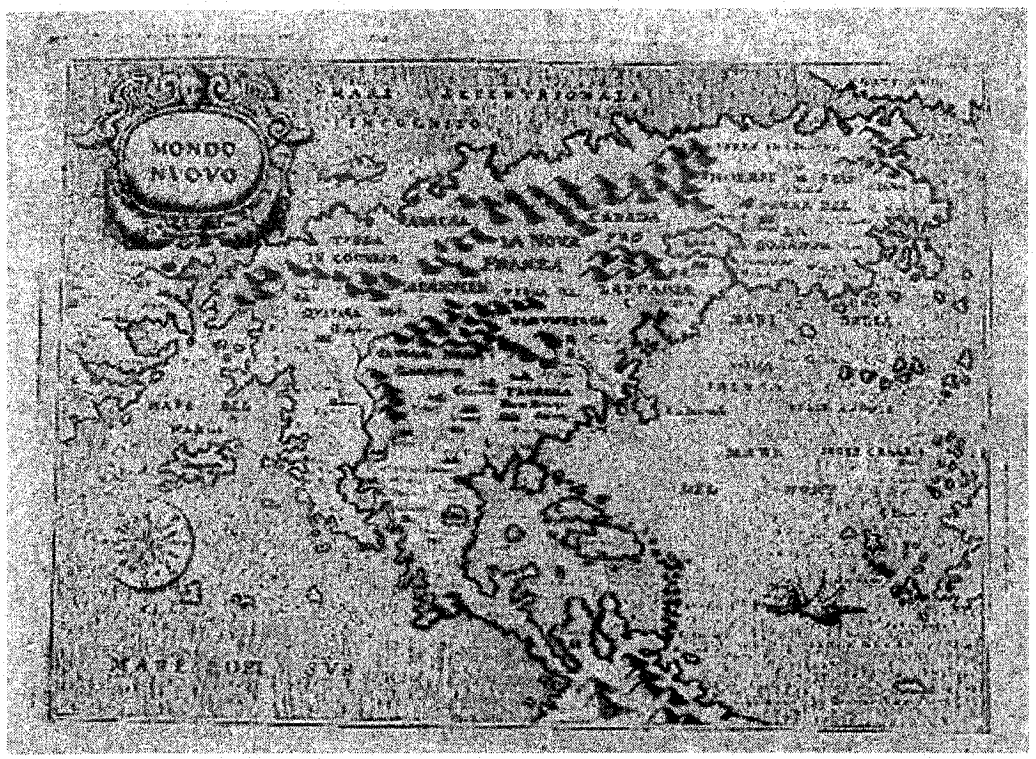


FIGURE 1.2. This sixteenth century map of North America, produced by Tomaso Porcacchi, illustrates the difficulty of simultaneous localization and mapping (SLAM) [**94**]. The obvious distortion of the eastern seaboard is due to the chicken-and-egg nature of the problem.

(iv) does not require any explicit assumptions concerning the imaging geometry of the sensor,

(v) does not depend on modifications to the environment,

(vi) exploits local image features rather than global image properties,

(vii) employs a probabilistic approach, and

(viii) can be learned automatically from the environment.

1.4. Learning and Exploration. The question of how to automatically learn a representation of the world, the last problem stated above, has been considered by a variety of researchers (e.g. [**64, 31, 144**]). These methods are almost universally concerned with minimizing uncertainty in the inferred structure. The problem of uncertainty minimization is compounded by the chicken-and-egg nature of the mapping

problem– how does the robot update its map if it does not know where it is, and how does the robot determine where it is if the map is incomplete (Figure 1.2)? While the tools and methods that have been developed in the robotics context are useful, in most cases they are bound to a particular sensing modality (specifically, range sensors) and furthermore very little attention has been paid to the problem of *how* to explore continuous pose spaces beyond the application of simple heuristics[1]. That is, how should the robot move through the world so as to minimize uncertainty and maximize the knowledge that it gains? To that end, this thesis is also motivated by the need to consider the robot's trajectory as an important factor in simultaneously maximizing map accuracy and coverage. The primary interest is in determining exploration policies that are best suited for developing a visual representation of the world, particularly in consideration of the lack of tools and approaches to mapping using vision.

To briefly summarize the requirements and goals for a complete approach to map construction:

Sensing: Passive device, inexpensive.

Representation: Minimize assumptions about device and world.

Methodology: Principled data collection, probabilistic framework.

This thesis will present an approach that meets all of the stated requirements by applying a machine learning approach to modelling the visual world with a camera.

2. Problem Statement and Objectives

This thesis answers three questions. First, how can the visual properties of a large-scale scene be modelled reliably such that the models we build are useful for robotic navigation and position estimation? Second, how can such models be constructed automatically, particularly when only limited prior information is available concerning the position of the robot as it collects training data? Finally, given solutions to the

[1]There is, however, a rich body of literature on exploring discrete state spaces, such as breadth-first, depth-first and A^* search [14].

first two questions, what exploration strategies are best suited for a moving robot to collect observations of a scene and construct accurate models on-line?

The objectives of this thesis are to answer these questions while exploiting assumptions about the world and about the robot that are as general as possible. Specifically, we will not commit to a specific optical geometry (for example, a perspective-projection pinhole camera versus an omnidirectional camera), nor will we require careful calibration of the camera's intrinsic properties or its extrinsic properties with respect to how it is mounted on the robot. Furthermore, we will not make any assumptions concerning the geometric and illumination properties of the scene being modelled. Finally, recognizing that there is a wide diversity of approaches to representing uncertainty, this thesis aims to present a framework that is compatible with a variety of probabilistic state representations, including particle filtering, grid representations and Kalman Filters.

In achieving these objectives, a framework for computing solutions to the questions at hand will be developed, with a discussion of the motivation for each component. The solutions will in turn be validated experimentally.

3. Contributions

The main contributions of this thesis deal with visual maps, how they are represented, and how they can be constructed. In particular, there are four main contributions:

(i) The first complete framework for visual feature learning, entailing four stages: selection, tracking, modelling, and evaluation;

(ii) The first visual feature representation and model for large-scale space that does not rely on assumptions concerning feature or camera geometry.

(iii) An original framework whereby a visual map can be inferred, taking into account a spectrum of possible *a priori* information concerning the robot's knowledge of its pose as it collects training images, varying from extremely limited to very strong priors; and

(iv) The first quantitative examination of the impact that a robot's exploratory trajectory has on the on-line construction of a map of an unknown environment, and an empirical evaluation of the trajectory properties that lead to accurate maps.

In addition to the primary contributions, there are a variety of secondary contributions. Among these contributions are:

- The design and implementation of an extensive and flexible software package that implements all the components of the visual mapping and exploration frameworks;
- The development of a generative approach to feature modelling, whereby it becomes possible to predict visual feature observations as a function of pose;
- The development of two specific feature models, each of which is tuned to meet a complementary set of computational requirements;
- An original method for feature tracking that takes into account a model of visual attention to provide priors for matching locations, and exploits an evolving model of feature appearance in order to increase the domain over which a feature can be accurately tracked;
- An empirical evaluation of which quantitative attributes of a feature are best suited for feature modelling.
- An empirical evaluation of the visual mapping framework versus approaches to appearance-based robot pose estimation that exploit global image structure;
- The application of the generative feature model to the inference of camera pose using a variety of uncertainty representations, including a discretized posterior distribution representation for global pose estimation, and an Extended Kalman Filter for approximating the posterior given a strong prior.
- A consideration of some of the limitations of visual maps and the development of mechanisms for addressing these limitations in a real-world setting.

In particular, an approach to working in high-dimensional configuration spaces using a low-dimensional representation;

- Extensive experimental results demonstrating all aspects of both the mapping and exploration frameworks.

The visual mapping framework is related to work that is presented in my Master's thesis [**115**], and it is important to delineate the boundary between this thesis and that one. In the prior work the approaches to feature modelling and pose inference were dependent upon crucial assumptions about feature behaviour as a function of pose, namely, that each feature behaved as a one-to-one mapping from observation to pose, an assumption which generally fails in large environments. For example, the position of a feature in the camera image might be identical under a sideways translation followed by a compensating rotation. Additionally, the pose estimates were arrived at without the intermediate step of feature generation, making the models difficult to evaluate, and imposing certain constraints on inferred pose posterior distributions (i.e. that they are composed solely of Gaussian mixture models). A new approach to computing a pose estimate will be presented in Section 3.5. Finally, my Master's thesis did not examine the inference of the poses of the training images (Chapter 4), nor did it consider the simultaneous localization and mapping problem (Chapter 5), but rather depended on the availability of ground truth for constructing the map.

4. Statement of Originality

Portions of the results presented in this thesis work have appeared previously in, or are currently in submission to peer-reviewed journals or conference proceedings [**126, 127, 125, 129, 128, 130, 124, 100, 101, 123, 122, 116, 118, 120, 121, 119, 117**]. In some cases, I have co-authored papers with Ioannis Rekleitis and Evangelos Milios that employ the visual map representation in the context of testing Ioannis' research on collaborative approaches to robot localization and exploration. In those works the distinction between our individual contributions is explicit.

My published work also contains results that are not reported in this thesis. In most cases, new results are presented here duplicating the original experiments (Chapter 3), but using improvements to the original algorithms presented in the published work. In addition, for the purposes of clarity, Section 3.6 presents results from [**125**] comparing the performance of the visual map approach with principal components analysis (PCA), but omits results from the same work that compare the approach with an ad-hoc localization method. This decision was made on the basis that the ad-hoc method is not known to the community and its inclusion would not further the objectives of the thesis. Finally, I have published work related to the software infrastructure underlying the experimentation. To report those results here would also distract from the main goals of the thesis.

5. Outline

The remainder of this thesis will be organized as shown below. Due to the variety of the sub-problems that are examined, in lieu of a monolithic literature review, the prior work on each major sub-problem will be presented in the chapter in which the respective problem is considered. This approach is meant to improve the continuity and readability of the text. The outline of the thesis is as follows:

Chapter 2: A presentation of the visual map representation, including an outline and discussion of the following topics:

- Background material on probabilistic representations,
- Visual attention,
- Tracking,
- Modelling visual features, and
- Evaluating visual features.

Chapter 3: An implementation of the visual map representation and experimental results illustrating its application to a variety of visual tasks.

Chapter 4: Mapping with limited pose information. This chapter examines the problem of learning a visual map when the pose information associated with the training images is incomplete.

Chapter 5: A presentation of simultaneous localization and mapping using the visual mapping framework. We will concentrate on the question of selecting exploration trajectories which are best suited to building an accurate map.

Chapter 6: Implementation details and experimental results addressing the problem of selecting exploration trajectories.

Chapter 7: Discussion of the work and future directions for exploration.

Following the presentation of the visual mapping framework in Chapters 2 and 3, the basic themes of the subsequent chapters revolve around the issue of what prior information is available concerning the robot's exploratory trajectory. Figure 1.3 depicts this spectrum of information, by plotting the exploratory poses of the robot, as determined by the mapping process. On the right, the robot has complete information about its pose as it explores a grid and can infer the visual feature models directly (Chapter 2). On the left, the robot has very limited information about where it was when it collected its training images and so the mapping problem involves inferring the set of training poses as well as the visual representation (Chapter 4). In the middle, the robot has some model of uncertainty about its pose as it explores (represented by the ellipse at each position), and it must reliably infer the map, taking into account this information (Chapter 5).

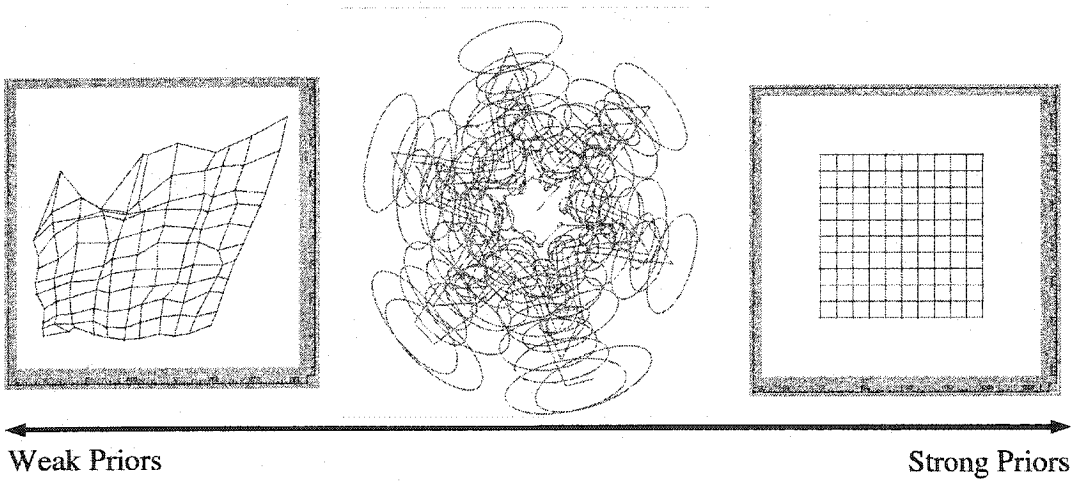


FIGURE 1.3. The spectrum of prior information available for map construction. Weak, or limited ground-truth pose information requires inference of the poses of training images, whereas strong priors (known training poses) provide rigid constraints for inferring the map. In the middle, trajectory information can provide some constraints on the training image poses. Refer to the text for further details.

CHAPTER 2

Visual Maps

1. Overview

This chapter presents the visual mapping framework– a method for learning a set of models that are suitable for representing image-domain features. The first half of the chapter will cover background material that will motivate the development of the framework, including a discussion of the concept of modelling the environment using a generative approach, and how such models are applicable to inference problems, such as robot localization. An overview of how generative models fit into the bigger picture of robot navigation and exploration will be presented. This discussion will lead into the development of the visual mapping framework itself and how generative feature models can be inferred and represented.

The visual mapping approach is motivated by a model of how biological systems process visual information. The approach entails the automatic selection of features using a visual attention mechanism, as well as the synthesis of models of their visual behaviour. We will develop a generative model that smoothly interpolates observations of the features from known camera locations[1]. The uncertainty of each model will be computed using cross-validation, which allows for *a priori* evaluation of the feature attributes that are modelled. This presentation of the learning framework will

[1]In Chapter 4 we will consider an alternative model aimed at computing approximate models when computation time is a critical factor.

provide the basis for the implementation and experimentation presented in subsequent chapters.

2. Previous Work

The purpose of constructing a visual representation in this thesis is to facilitate robotic tasks, such as localization. We will concentrate on approaches to the robot localization problem, addressing representational issues as they arise.

2.1. Geometric Representations. Many early solutions to the pose estimation problem assume that the problem of landmark detection, and sometimes even recognition, is solved. These methods employ a computational geometry approach that triangulates range and bearing information for a set of landmarks in order to arrive at a pose estimate. Triangulation methods are based on traditional methods in cartography and navigation, which use the angles or *bearings* measured between the lines of sight to known landmarks in the environment. The seminal triangulation-based approaches in the domain of mobile robotics rarely involve real-world implementation, allowing the researcher to ignore the problems of landmark detection and recognition, which are often issues that are domain- and sensor-dependent [**136, 2, 137**].

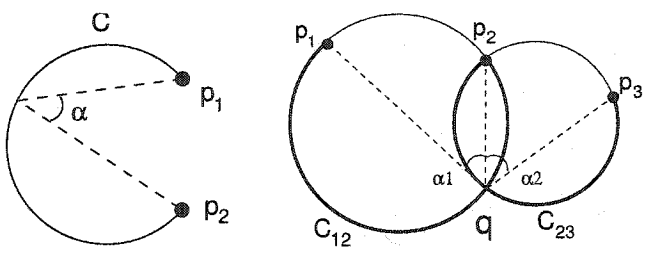
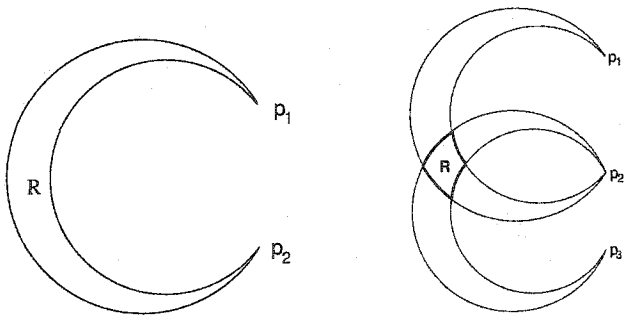


FIGURE 2.1. Pose constraints C and $\mathbf{q}$, given bearings to (a) two landmarks p_1 and p_2, and (b) three landmarks p_1, p_2, and p_3. The constraints represent the space of possible robot poses given the associated bearing measurements.

Given only the angle α measured between two distinguishable landmarks, the pose of the observer is constrained to the arc of a circle, as shown in Figure 2.1(a). In the case where there are three landmarks, providing bearing measurements α_1 and α_2, the pose is constrained to a single point, lying at the intersection of two circles (Figure 2.1(b)), provided that no two landmarks are coincident. When there are four or more landmarks, the system is overdetermined, or may have no solution [**136, 63**]. This result provides a basis for several localization solutions under a variety of conditions. For instance, Sugihara provides a consideration of the problem of localization when the observed landmarks are indistinguishable [**136**]. That work seeks out a computationally efficient method for finding a consistent correspondence between detected landmarks and points in a map. This correspondence method is improved upon by Avis and Imai [**2**]. Both of these results presuppose the reliable extraction of landmarks from sensor data and the accuracy of the bearing measurements – only minor consideration is given to the problem of using uncertain bearings.

Sutherland and Thompson approach triangulation methods from the perspective that the landmark correspondence problem has been solved, but the bearings to observed landmarks cannot be precisely known [**137**]. It is shown that informed selection of the set of landmarks to be used in the map can help to minimize the *area of uncertainty*, that is, the area in which the robot may self-locate for any given error range in visual angle measure. Figure 2.2 shows the area of uncertainty computed for a bounded error range in the cases of a) two and b) three observed landmarks. Sutherland and Thompson demonstrate that the size of the area of uncertainty can vary significantly for different configurations of landmarks. The goal of their work is to select landmarks whose configurations minimize the area of uncertainty.

Betke and Gurvits also consider the problem of localization from uncertain bearings. They are concerned primarily with the efficient computation of a position estimate from an overdetermined set of bearings [**6**]. They derive a complex-domain representation of the positions of the landmarks that linearizes the relationship between the constraining equations and allows the system to be solved in time linear in



(a) (b)

FIGURE 2.2. Pose constraints R given bounded uncertainty on bearings to (a) two landmarks, and (b) three landmarks.

the number of landmarks, provided that certain constraints on the formation of the landmarks are met.

All of the triangulation methods considered above make a strict set of assumptions about the environment and the robot. In every case, the robot is provided with an accurate *a priori* map of the positions of known landmarks, and in some cases assumes the ability to unambiguously distinguish between the observed landmarks. In addition, it is assumed that the robot can always reliably detect landmarks in the sensor data. An important aspect of these solutions is the observation that sensor measurements are not always accurate, and hence it is most reasonable to seek out a solution which minimizes the uncertainty of the position estimate.

2.2. Appearance-based Representations. An alternative to explicit geometric representations of the world is that of employing an appearance-based representation, which computes a compact description of the manifold of observations as a function of pose. Dudek and Zhang used a neural network to compute a representation for pose estimation from vision data [**32, 33**]. Nayar *et al.* and Pourraz and Crowley have examined an appearance-based model of the environment based on principal components analysis (PCA). They perform localization by interpolation in the manifold of training images projected into a low-dimensional subspace [**95, 16, 84**].

Formally, the low-dimensional subspace is determined to be the first k principal components U_k of the matrix Z, whose columns correspond to a set of training vectors. Each training vector is defined by a training image sampled from the pose space of the robot, whereby the image is "unrolled" into a vector by sorting the pixels in a raster-scan order. If U is defined to be the orthonormal matrix determined by the singular values decomposition (SVD) of Z:

$$Z = U\Sigma V^T \tag{2.1}$$

then U_k is defined as the k columns of U corresponding to the largest eigenvalues contained in the diagonal matrix Σ [**96, 39**]. It should be noted that PCA depends on global image properties and as such is susceptible to occlusions and other image degradations.

Some appearance-based methods exploit specific sensor configurations and assumptions concerning the geometry of the environment. For example, Dellaert *et al.* represent the visual world as a planar mosaic, constructed by imaging the ceiling in an indoor environment [**20**]. Such a representation affords a straightforward derivation of the observation model. To reduce the complexity of the distribution, the sensor image is represented by a single intensity measurement corresponding to a pixel near the centre of the image.

While these approaches demonstrate the utility of appearance-based modelling, they can suffer due to the dependency of the result on global sensor information and the assumptions they make concerning the structure of the environment.

Several researchers have also considered local feature models for localization. Recent work by Se *et al.* [**109**], applies a scale- and rotation-invariant feature detector to determine feature matches, and then localize using a least-squares pose estimate based on the geometrically determined positions of the features from the output of a stereo camera. The feature detector locates maximal responses in a difference-of-Gaussian pyramid constructed from the input image. Similar approaches have seen success in the tasks of place recognition (recognizing a scene from an ensemble) in

work by Dudek and Jugessur [**28**], and in object recognition, such as those used by Lowe, Schmid and Shokoufandeh *et al.* [**73, 107, 114**]. These approaches omit any explicit geometry computation and instead rely on the appearance models determined by the feature detector to locate matches. An important aspect of these works is the task of recognizing pseudo-invariants under changes in viewing conditions. In particular, the attention operators developed are insensitive to changes in scale and planar rotation. For the localization problem, it is not only important to be able to recognize pseudo-invariants, but to be able to parameterize the effects that changes in pose have on the feature. While this thesis considers only translation invariance, these prior results indicate the feasibility of readily modelling other parameterizations.

2.3. Active Localization. Both geometric and appearance-based localization approaches define methods for computing pose estimates from an observation taken at a single instance in time. There is a large class of results concerned with localizing a robot over time as it takes observations while moving through the environment. Smith, *et al.*, and Leonard and Durrant-Whyte applied Extended Kalman Filter (EKF) methods to localize a robot from successive range measurements [**133, 68**], and Fox *et al.* developed *Markov Localization*, which can track robot pose in situations where the probability distribution over robot pose is multi-modal [**36**]. These results will be discussed in greater depth later in the thesis. Seiz *et al* also looked at exploration methods for resolving the robot's position [**110**]. A key result in the problem of active localization, demonstrated by Dudek, *et al.*, is that unambiguously determining a robot's pose in a known environment with minimum travel is an NP-hard problem [**29**].

2.4. Visual Attention. The concept of visual attention is central to our visual mapping framework. Visual attention, or selective attention, entails preferentially selecting a subset of image neighbourhoods, based on their content, that are *a priori* considered potentially useful for visual perception. This selection process acts as a first-order filter for conserving limited computational resources– the visual world

is simply too rich to take in all at once. In fact, current psychophysical research suggests that the human brain processes only a tiny fraction of the information that hits the retina and that the perception of a rich and complete visual world is due to the brain "filling-in" most of the scene by applying assumptions about the consistency of the scene under observation [**151, 88**].

Research on mammalian visual attention suggests that a key attribute of the loci of attention is that they are different from their surrounding context [**58, 108, 150**]. Several featural dimensions have been identified that lead to pre-attentive "pop-out" and, presumably, serve to drive short term attention [**148**]. Feature maps used by human attention may include those for colour, symmetry, edge density, or edge orientation, among others, some of which have been implemented computationally [**54, 98**]. Other research demonstrates that attentional processing is characterized by visual saccades to areas of high curvature, or sharp angles [**86**]. In the computational domain, several researchers have investigated a variety of attention operators. For example, Kelly and Levine exploit regions of high edge symmetry as loci of attention [**57**], Harris and Stephens, and Lucas and Kanade implemented corner detectors based on shared principles of maximizing the locality of the image gradient [**43, 75**]. The latter, often referred to as the Kanade-Lucas-Tomasi (KLT) operator, was later popularized by Shi and Tomasi [**43, 113**]. Finally, Bourque and Dudek have demonstrated that the behaviour of an edge-density attention operator on simple stimuli resembles that predicted by the psychophysical literature [**8**]. It is this latter work which forms the basis for the operator employed in this thesis.

The subsequent sections will provide the background for developing a probabilistic representation of the environment, followed by an overview of the feature learning framework. In particular, we will present the feature model, our approach to feature detection and tracking, and finally a method for evaluating the inferred feature models. This will lead into the following chapter which will present experimental results illustrating the utility of the framework for several inference tasks.

3. Generative Models and Bayesian Inference

This thesis approaches probabilistic inference problems from the context of developing a generative model of the environment. In order to facilitate the development of the concept of generative models, consider a mapping $F : \Re^m \rightarrow \Re^n$ from some set of parameters to another, that describes some phenomenon:

$$\mathbf{z} = F(\mathbf{w}) \tag{2.2}$$

where $\mathbf{z} \in \Re^m$ and $\mathbf{w} \in \Re^n$. $F(\cdot)$ might correspond to the physical laws governing the phenomenon, and the parameters $\mathbf{w}$ can be considered as an instance of the event. The vector $\mathbf{z}$ then corresponds to an observation of the phenomenon. Often it is convenient to fix some subset $\mathbf{m}$ of the input parameters $\mathbf{w}$ that describes invariant or intrinsic properties of the world (for example, daytime illumination is due to the sun, as opposed to some other star), and allow other parameters, denoted as $\mathbf{q}$, to vary. For example, in computer graphics a rendered image is a function of many parameters: the objects in the virtual scene, the illumination of the scene, the intrinsic parameters of the virtual camera, such as focal length, and the extrinsic parameters of the camera, such as its position and orientation in the scene. One can describe the rendered image $\mathbf{z}$ with the function

$$\mathbf{z} = F(\mathbf{m}, \mathbf{q}) \tag{2.3}$$

$$= F_m(\mathbf{q}) \tag{2.4}$$

where $\mathbf{q}$ describes the position and attitude of the camera, and $\mathbf{m}$ describes all the remaining parameters of the scene. Using the *generating function* $F_m(\cdot)$, we can vary $\mathbf{q}$ along some trajectory and generate a fly-through of the virtual world described by $\mathbf{m}$.

In the real-world, most observed phenomena are stochastic– the number of photons that strike a photographic film or CCD in some time frame is random and it is nearly impossible to exactly describe $F_m(\cdot)$ for an arbitrary scene, except in the most

constrained circumstances. For example, consider the impact of a partly cloudy day on illumination, or the parametric description of all of the trees, bushes and rocks in a forest (if that seems easy, be sure to take into account the season, or perhaps the impact of a caterpillar infestation). Therefore, sensor observations usually represent samples from an observation model $p(\mathbf{z}|\mathbf{w})$, or in the case of a model with some fixed parameters $p(\mathbf{z}|\mathbf{q}, \mathbf{m})$, that is related in some way to $F_m(\cdot)$. As we will regard $\mathbf{m}$, and therefore $F_m(\cdot)$ as an invariant entity, we will usually just write $p(\mathbf{z}|\mathbf{q})$. In the computer graphics example, $p(\mathbf{z}|\mathbf{q})$ is a probability distribution over images that describes the likelihood that our rendering engine produced image $\mathbf{z}$, given that the virtual camera is placed at position $\mathbf{q}$. If we produce images with a deterministic rendering engine, the conditional likelihood $p(\mathbf{z}|\mathbf{q})$ is a delta function:

$$p(\mathbf{z}|\mathbf{q}) = \begin{cases} 1 & \text{if } \mathbf{z} = F_m(\mathbf{q}) \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

In the real world, and in the absence of a more informative sensor model, the observation model might be approximated by a Gaussian distribution

$$p(\mathbf{z}|\mathbf{q}) \approx k \exp(-\frac{||\mathbf{z} - F_m(\mathbf{q})||^2}{2\sigma^2}) \tag{2.6}$$

where $\exp(x) = e^x$, σ^2 is a variance describing the uncertainty of the observation, and $k = (\sigma\sqrt{2\pi})^{-1}$ is a normalizing constant to ensure the total probability sums to one.

The stochastic nature of the world presents serious problems for a mobile robot that wants to use an image to determine its pose, configuration, or position[2]. Supposing that photons behaved deterministically, and that a robot could be equipped with a "perfect" camera, the complexity of modelling the real world's interactions with the camera is enormous. However, supposing that the robot had a reasonably accurate model of the observation distribution $p(\mathbf{z}|\mathbf{q})$, it is possible to infer the probability distribution $p(\mathbf{q}|\mathbf{z})$ over possible poses of where the robot might be using Bayes'

[2]... or state. Throughout this thesis, I will refer to this entity exclusively as the *pose* of the robot, a vector $\mathbf{q}$ in some pose- (or configuration-, or state-) space $C \in \Re^m$.

Law [**139, 26**]:

$$p(\mathbf{q}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{q})p(\mathbf{q})}{p(\mathbf{z})}. \tag{2.7}$$

The distribution $p(\mathbf{q}|\mathbf{z})$ is referred to as the *a posteriori* (or, the posterior) probability distribution of pose $\mathbf{q}$, given an observation $\mathbf{z}$, and $p(\mathbf{q})$ is referred to as the *a priori* distribution over pose, otherwise known as the prior, which describes any information the robot had about its pose independent of the observation $\mathbf{z}$. The distribution $p(\mathbf{z})$ is referred to as the *evidence*. The evidence is constant with respect to $\mathbf{q}$ and so is usually ignored as a normalizing factor[3].

4. Robot Localization and Navigation

Equation 2.7 describes the key insight to the localization problem– that is, how to infer a robot's pose given an observation. The posterior distribution $p(\mathbf{q}|\mathbf{z})$ could be a delta function, Gaussian, or multi-modal over $\mathbf{q}$. What is significant is that the solution is exact– with accurate descriptions of any prior information and the interaction of the sensor with the world, we can describe all of the information we have about the robot's pose. Even if the posterior is not a delta function, or even unimodal, it can be used to make inferences about the outcomes of future actions, and therefore to plan, inevitably to acquire more information that might further constrain the pose distribution.

Despite the fact that the posterior is unlikely to ever evaluate to a delta function, it is clear that Bayes' Law is a powerful inferential tool, and that in the presence of uncertainty, it is always safer (for the robot, and for the environment) to work with a probability distribution. One example of how the posterior can be applied is in the context of active localization. Suppose at time t the robot has a representation of its pose $p(\mathbf{q}|t)$, it executes some action u, and subsequently acquires an observation $\mathbf{z}$.

[3]The evidence does play a role, however, in evaluating the accuracy of the model $\mathbf{m}$. Recall that for brevity we dropped the intrinsic model parameters $\mathbf{m}$ from the distributions, and that $p(\mathbf{z})$ is more accurately expressed as $p(\mathbf{z}|\mathbf{m})$.

After executing the action u, the pose distribution can be described as

$$p(\mathbf{q}|u) = \int_{\mathbf{q}' \in C} p(\mathbf{q}|u, \mathbf{q}')p(\mathbf{q}'|t)d\mathbf{q}' \tag{2.8}$$

where C is the configuration space of the robot and $p(\mathbf{q}|u, \mathbf{q}')$ is a probability distribution describing the (stochastic) effect of action u on the robot's pose (recall that wheels slip, odometers are inexact, *et cetera*).

Subsequently, the robot takes an observation $\mathbf{z}$. We can apply Equation 2.7, substituting $p(\mathbf{q}|u)$ as the prior:

$$\begin{aligned} p(\mathbf{q}|t+1) &= p(\mathbf{q}|\mathbf{z}, u) && (2.9) \\ &= \frac{p(\mathbf{z}|\mathbf{q}, u)p(\mathbf{q}|u)}{p(\mathbf{z}|u)} && (2.10) \\ &= \frac{p(\mathbf{z}|\mathbf{q})p(\mathbf{q}|u)}{p(\mathbf{z})} && (2.11) \end{aligned}$$

where we assume that $p(\mathbf{z}|\mathbf{q}, u) = p(\mathbf{z}|\mathbf{q})$ by a Markov assumption that the observation depends only on pose and not how the robot arrived at that pose, and also that $p(\mathbf{z}|u) = p(\mathbf{z})$, a more fragile assumption that the distribution of possible observations is unaffected by the robot's actions. In most situations, $p(\mathbf{z})$ will be constant relative to the quantity of interest (that is, $\mathbf{q}$).

Given a series of actions and observations, it is possible to determine the posterior $p(\mathbf{q}|t)$ for any t by applying Equations 2.8 and 2.11 recursively. This process is known as *Markov Localization* [**36**], named as such because the sequence of actions and observations form a *Markov chain* where the outcome of each action at time $t+1$ is dependent only on the pose of the robot at time t and the action it executes.

Markov Localization, and its Gaussian approximation, the Kalman Filter [**133, 68**], have been widely deployed and are considered the *de facto* standards for robotic navigation applications (e.g. [**41**]). The update equations provide a recipe for mapping actions and observations to probability distributions over pose-space. The necessary ingredients for implementing these update equations on a finite-precision computer are:

- A plant model $p(\mathbf{q}|u, \mathbf{q}')$ that describes the outcome of a robot's action;
- a method for computing an approximation of the integral in Equation 2.8;
- the observation model $p(\mathbf{z}|\mathbf{q})$; and
- for navigation, a method for choosing u at each time step.

For continuous pose spaces, Equation 2.8 can only be approximated on a finite-precision computer, and there are a number of methods available for finding approximate solutions. Among the more common are approaches that approximate the probability distributions as Gaussian with a mean and covariance (the Kalman Filter and its cousin, the Extended Kalman Filter) [**133**], or as a finite set of weighted particles (the particle filter, e.g. [**53**]), versus approaches that approximate the pose distributions by discretizing the pose-space into a finite, bounded set of bins [**83**]. In addition, a number of researchers have addressed the problem of modelling the plant model $p(\mathbf{q}|u, \mathbf{q}')$ for various robot drive mechanisms, and we will not consider that problem further [**11, 99**].

The remaining unknowns: the observation model $p(\mathbf{z}|\mathbf{q})$, and the navigation (or exploration) policy, are the focus of this thesis. The next few sections develop an observation model for a camera sensor and demonstrate its utility for localization and navigation, with particular attention paid to inferring the observation model given a set of exemplar observations.

5. Visual Maps

The remainder of this chapter will describe a technique for learning a set of image-domain models of selected features in a scene, and then illustrate how the learned models can be used to approximate an observation model $p(\mathbf{z}|\mathbf{q})$. The models will capture the appearance and image-domain geometry of features as a function of camera pose, effectively learning a generating function $\hat{F}_m(\cdot)$ that approximates $F_m(\cdot)$ for each feature. It is precisely these generating functions that will constitute the robot's "mind's eye", enabling it to visualize the features as a function of pose.

The learned functions will be evaluated so as to deliver likelihood estimates of future observations, and provide a mechanism for selecting reliably modelled features.

5.1. Problem Statement. Formally, this chapter addresses the following problem:

Given:

- $I = \{\mathbf{z}_j\}$, an ensemble of images of an environment, and
- $Q = \{\mathbf{q}_j\}$, ground truth pose information indicating the pose of the camera from which each image was acquired.

Compute: a feature-based visual representation of the environment that enables:

(i) *Extraction*, modelling and evaluation of visual features of the environment from I.

(ii) *Recognition* of modelled features in new images.

(iii) *Prediction* of a feature observation $\mathbf{z}_i^*$, given a pose $\mathbf{q}$, and

(iv) *Evaluation* of the feature observation likelihood $p(\mathbf{z}_i|\mathbf{q})$, for an observation $\mathbf{z}_i$ and pose $\mathbf{q}$.

This work is the first to employ generic image-domain feature models that do not rely on assumptions concerning feature or camera geometry. Image features, rather than global image properties, are explicitly employed because they provide robustness to limited illumination variation, partial occlusion due to scene dynamics and possibly even small changes in camera parameters. Furthermore, the computational complexity of high-level inference is mitigated by using only subregions of an image, a feature that evolutionary biology has exploited with remarkable success. These claims will be supported by the experimental results presented in Chapter 3.

An important aspect of feature modelling is the selection and evaluation of the features themselves. I will approach this problem by employing a model of visual saliency to initially select candidate features, and track them across an ensemble of training images. Given these tracked feature observations, a set of feature models are

constructed and subsequently evaluated and filtered. The result is a set of feature models that have been determined to be reliable for tracking and localization.

Earlier, we demonstrated how robot pose can be inferred from an image using Bayes' Law. As an illustrative example, Figures 2.3 a) and c) depict images from a laboratory environment from two known poses $q_0 = 0$ and $q_1 = 1$ in a one-dimensional pose space. Given the image $\mathbf{z}$ in Figure 2.3 b), taken from an unknown pose q which lies somewhere on the line connecting q_0 and q_1, the task of localization is to compute a probability distribution $p(\mathbf{q}|\mathbf{z})$ and, ideally, a q^* which maximizes the likelihood of the image according to Equation 2.7.

The chief difficulty in producing a posterior distribution is that of computing the observation likelihood $p(\mathbf{z}|\mathbf{q})$. Given the enormous complexity of the visual world, and the equivalent computational complexity involved in managing a global model of the world, it seems reasonable to rely instead only on some subset of the perceptually relevant parts of the scene. In fact, this is a well-established characteristic of the human visual system, particularly as it applies to task-driven perception [**104, 3**]. A set of models of local image features can be used to compute the likelihood of observations of these features from a particular pose.

Formally, pose inference based on the observation of a set of image features can be accomplished by assuming that the observation model $p(\mathbf{z}|\mathbf{q})$ is approximated by the joint likelihood of the set of feature observations $\{\mathbf{z}_i\}$ conditioned on pose $\mathbf{q}$:

$$p(\mathbf{z}|\mathbf{q}) \approx p(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n|\mathbf{q}) \tag{2.12}$$

where we assume the formula is an approximation because we are ignoring any information that might be present in parts of the image other than those occupied by the detected features. The definition of the feature observation vector $\mathbf{z}_i$ will be presented in Section 9.

Deferring for the moment any discussion of precisely how the joint likelihood is computed, let us assume that it is some function of a set of probability distributions

(a)

(b)

(c)

FIGURE 2.3. Images from three successive poses along a line: a) known pose $q = 0$, b) q unknown, c) known pose $q = 1$. The value of q at b) must be inferred from a) and c).

$M = \{p(\mathbf{z}_i|\mathbf{q})\}$ that are instantiated over the set of features[4]. Furthermore, if we can construct a generating function $\hat{F}_i(\cdot)$ that can compute a maximum likelihood observation of a feature as a function of pose:

$$\mathbf{z}_i^* = \hat{F}_i(\mathbf{q}) \tag{2.13}$$

then, all other factors being constant, the distribution $p(\mathbf{z}_i|\mathbf{q})$ can be represented as some function of the observation $\mathbf{z}_i$ and the prediction $\mathbf{z}_i^*$. The visual mapping framework provides a mechanism for constructing an approximation to Equation 2.13 based on a set of feature observations, and for subsequently computing the observation likelihood $p(\mathbf{z}_i|\mathbf{q})$.

The framework operates by automatically selecting potentially useful features $\{f_i\}$ from a set of training images of the scene taken from a variety of camera poses (i.e. samples of the pose-space of the robot). The features are selected from each image on the basis of the output of a visual attention operator and are tracked over the training images. This results in a set of observations for each feature, as they are detected from different positions. The application of an attention operator allows one to focus on the local behaviours of features, which themselves may be easier to model and more robust than global image properties. For a given feature f_i, the modelling task then becomes one of learning the imaging function $F_i(\cdot)$, parameterized by camera pose, that gives rise to the imaged observation $\mathbf{z}_i$ of f_i according to Equation 2.13[5].

Clearly, the imaging function is also dependent on scene geometry, lighting conditions and camera parameters, which are difficult and costly to recover [**135**]. Traditional approaches to the problem of inferring $F_i(\cdot)$ have either focused on recovering geometric properties of the feature under strict surface or illumination constraints

[4]This amounts to an assumption of conditional independence, since we do not consider joint distributions such as $p(\mathbf{z}_1, \mathbf{z}_2|\mathbf{q})$.

[5]The reader should note the distinction between a feature f_i (some visual phenomena), and a feature observation $\mathbf{z}_i$ corresponding to some measured property of f_i, usually conditioned on the pose of the camera.

(e.g. [4]), or developed appearance-based representations derived from the entire image. This thesis bridges the gap between these approaches by modelling both image-domain geometry and appearance in a single framework.

The next section will present the feature learning framework, followed by an elaboration of its components.

6. The Learning Framework

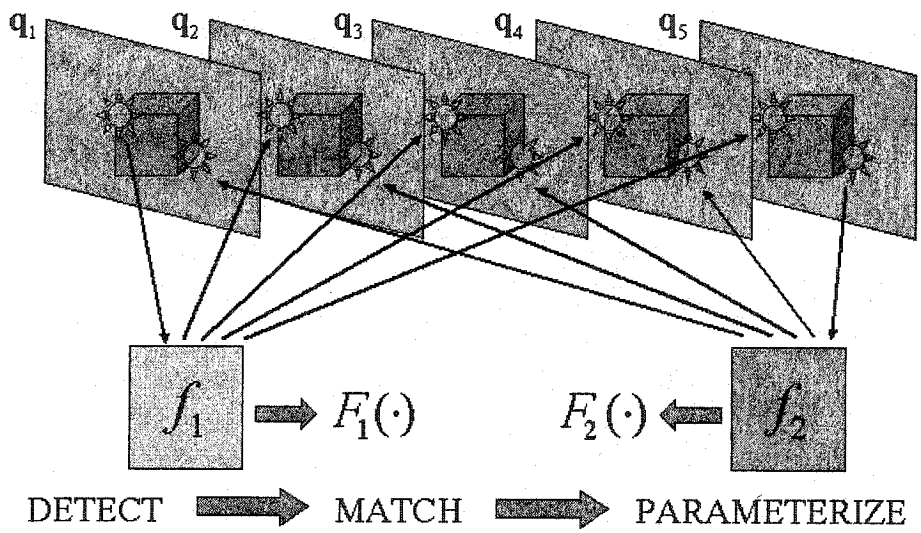


FIGURE 2.4. Learning framework: An ensemble of images is collected (top rectangles) sampling views of the scene from poses $\mathbf{q}_i$. Candidate features f_i are extracted and matched, and subsequently modelled using a generative model $F_i(\cdot)$. Refer to text for further details.

The visual map learning framework constitutes a method for extracting a set of useful feature models from an ensemble of training images. The approach operates as follows (Figure 2.4):

ALGORITHM 2.1.

(i) *The robot explores the environment, collecting images* $I = \{\mathbf{z}_j\}$ *from a sampling of known positions* $Q = \{\mathbf{q}_j\}$.

(ii) *Compute* Z_0*, a subset of images corresponding to training poses that span the explored pose space.*

(iii) *Compute a set of candidate features* $\{f_i\}$ *from the images in* Z_0 *using a model of saliency* Ψ *(Section 2.7).*

(iv) *For each extracted feature* f_i

  (a) *Initialize a generative feature model* $\hat{F}_i(\cdot)$.

  (b) *For each training image* $\mathbf{z}_j$

    (i) *Evaluate* $\hat{F}(\mathbf{q}_j)$ *to determine a matching template* $\mathbf{i}_f$ *for* f_i

    (ii) *Locate an optimal match* $\mathbf{i}^*$ *to* $\mathbf{i}_f$ *in the image* $\mathbf{z}_j$ *(Section 2.8).*

    (iii) *If a match is found, update the generative model* $\hat{F}_i(\cdot)$ *with the new information* $(\mathbf{i}^*, \mathbf{q}_j)$ *(Section 2.9).*

  (c) *Evaluate the quality of the model for* f_i *and remove it from the map if it fails to meet certain criteria (Section 2.11).*

A key point to note is that we are currently considering image ensembles for which ground-truth pose information is available. It is assumed that a mechanism is available for accurate pose estimation during the exploratory stage (such as assistance from a second observing robot [**103**], or the utilization of an expectation-maximization approach to map building [**143**]). This assumption will be relaxed in later chapters.

The matching and model update stages are interleaved, so as to facilitate matching over a wide range of feature appearance. This approach results in an anytime algorithm and the map can be used for localization before it is completed.

7. Feature Detection

Potential features are initially extracted from a subset of the training images using a model of visual saliency. The initial image subset is defined to be the set of images that uniformly sample the training poses such that no two images are closer

together in pose space than some user-defined threshold. The threshold itself depends on the average spacing between training images in the pose space.

In previous work conducted with Polifroni [**130**], we considered the use of several alternative interest operators in the context of constructing visual maps. The conclusions from that work indicated that a wide variety of feature detectors can be applied to the visual mapping problem, and that stability and uniqueness are the important factors in extracting useful features (see also [**113**]).

In the results presented in this thesis, we employ a measure of local edge density as the attention operator, as defined below. The edge map from a given image is convolved with a Gaussian kernel and local maxima of the convolution are selected as salient features. The Gaussian convolution diffuses the presence of an edge to nearby pixels, thereby contributing to those pixels' local measure of edge density. More formally, given an image $\mathbf{z}$, the Canny edge magnitude map is computed, and convolved with a wide Gaussian kernel, resulting in an edge density function $\Psi(\mathbf{x})$, where $\mathbf{x} = [u\ v]^T$ is an image location. For the local maxima computation, define $X = \{\forall \mathbf{x} \in I\}$ as the set of pixel locations in the image I, and the initial set of features, $M_0 = \{\arg\max_{\mathbf{x}\in X} \Psi(\mathbf{x})\}$, that is, the pixel location in the image where the saliency function Ψ is maximal. The algorithm proceeds iteratively. Define the set of candidate locations at the ith iteration to be

$$U_i = \{\mathbf{x} \in X : \forall_{\mathbf{m}_j \in M_i} ||\mathbf{x} - \mathbf{m}_j||_2 > \sigma\} \tag{2.14}$$

where σ is the standard deviation of the Gaussian mask used to define Ψ, and the set of features at the ith iteration to be

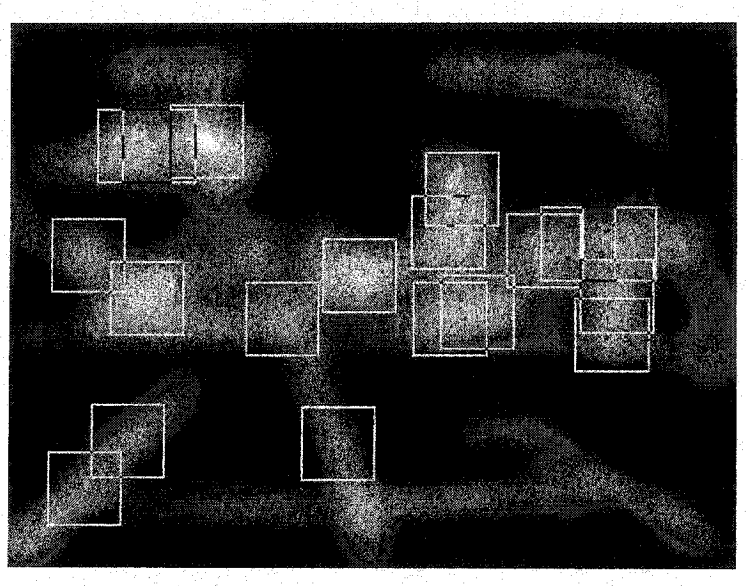
$$M_i = M_{i-1} \cup \{\arg\max_{\mathbf{x}\in U_i} \Psi(\mathbf{x})\} \tag{2.15}$$

Iteration halts when $\max_{\mathbf{x}\in U_i} \Psi(\mathbf{x})$ falls below a threshold which is defined as $t = \mu_D + k\sigma_D$, representing a user-defined k standard deviations from the mean density.

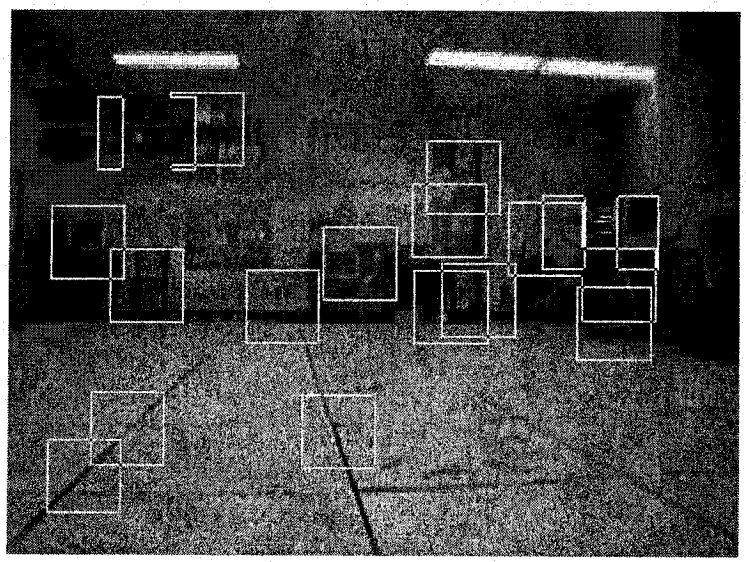
Once the set of feature points has been computed, the local image neighbourhood surrounding each point is presumed to contain useful information, and these feature

windows, along with their positions, are returned as the output of the operator. Figure 2.5 depicts the selected features from an image as superimposed squares over the original, and the convolved edge map. It should be noted that the scale of the edge operator (1 pixel) is significantly smaller than the size of the convolution operator[6]. Empirical experience with this operator suggests that it is stable and reliable for candidate feature selection. However, in some circumstances the edge density operator may not localize the feature with sufficient precision (for example, along the long line defined by a door frame). In such cases, a more precise operator, such as the KLT operator or other measure of saliency may be required (e.g. [**97, 113, 147, 55**]).

[6]Throughout this thesis, a Gaussian convolution operator with a width of 8 pixels is used, yielding feature windows that are 33 by 33 pixels in size. The value of k in the detector threshold t is set to 1.0.



(a)



(b)

FIGURE 2.5. Detected features in an image. a) The convolved edge map or density function, and b) the detected feature imposed on the input image as squares.

8. Feature Matching

Once an initial set of features has been extracted, the next phase involves matching the detected features over the entire training image set. Each training image is searched in sequence for each feature. The camera pose $\mathbf{q}_j$ of any given training image $\mathbf{z}_j$ is known and therefore one can construct a generative model of the feature (Section 2.9) to predict the intensity window $\mathbf{i}_f$ of the feature for the pose of the training image being searched. Define the best match to $\mathbf{i}_f$ in the image to be the image sub-window $\mathbf{i}_x$ centred at position $\mathbf{x}^*$ that has maximal correlation ρ with the predicted image $\mathbf{i}_f$, where ρ is defined as

$$\rho(\mathbf{i}_x) = \cos\theta = \frac{\mathbf{i_x} \cdot \mathbf{i}_f}{||\mathbf{i_x}||\,||\mathbf{i}_f||} \tag{2.16}$$

The problem of maximizing Equation 2.16 can be computationally expensive for large feature windows, even when the camera pose corresponding to the input image is known. This is especially true when no assumptions are made about imaging and scene geometry– as the camera moves through the scene, a feature might move an arbitrary number of pixels in an arbitrary direction through the image from one frame to the next. For example, given the size of the local intensity window that is used for matching (33 by 33 pixels), searching a 240 by 320 pixel image involves 76,800 evaluations of Equation 2.16, which in turn costs about 6537 floating point operations, or a total of 5.02×10^8 operations, notwithstanding potential savings from various optimizations.

In order to reduce the computational cost of matching, we exploit our *a priori* knowledge that the matching template is derived from a model that was originally extracted using a measure of visual saliency Ψ. The implication of this is that the visual saliency map provides a prior distribution over the image as to regions that are likely to be good matches to the template. Assuming that the saliency map can be efficiently computed (which is the case for an edge-density operator, where the Gaussian convolution operator is separable), it is possible to select potential matches by sampling image locations from a probability density function Φ which is

proportional to the saliency measure:

$$\Phi(\mathbf{x}) = k\Psi(\mathbf{x}) \tag{2.17}$$

where k is defined such that $\sum_{\mathbf{x}\in I} k\Psi = 1$ and I is the set of pixel locations in the image.

The matching algorithm operates as follows:

ALGORITHM 2.2.

(i) *Sample an image location* $\mathbf{x}$ *from* I *with probability* $\Phi(\mathbf{x})$.

(ii) *Perform gradient ascent in the image neighbourhood of* $\mathbf{x}$, *until Equation 2.16 is locally maximized.*

(iii) *Repeat from step 1 until the cumulative probability of all the image points examined exceeds some threshold.*[7]

(iv) *Return the image location* $\mathbf{x}^*$ *and local intensity neighbourhood* $\mathbf{i}^*$ *that resulted in maximal correlation.*

Consider a cumulative saliency function Ψ^* defined by summing the saliency of pixel locations in an image sorted in decreasing order of saliency. We will say that an image has n per cent saliency if fifty per cent of the maximal value of Ψ^* is achieved when n per cent of the sorted pixel locations have been added. In general, for structured environments most images will have low n per cent saliency. If we assume that generally $n < 25$, then the cost of matching, compared to global search, is reduced by at least 75%, discounting the costs of applying the attention operator and subsequent importance sampling. In practice, n can be considerably lower. For an edge density operator, and using the example above, the total cost of operation is about $0.25 \times 5.02 \times 10^8$ operations for the correlation and 5.07×10^6 operations for the edge density computation. In addition, sampling will require approximately

[7]In practice, a threshold of 50% is used.

3.8×10^6 operations. This yields a total cost of approximately 1.5×10^8 floating point operations.

When the sub-window maximizing Equation 2.16 is determined, the corresponding intensity neighbourhood and image position $[\mathbf{i}^* \ \mathbf{x}^*]^T$ is added to the feature model for f. When every training image has been considered, a set of matched features is obtained, each of which is comprised of a set of observations from different camera poses. Figure 2.6 depicts one such set, where each observation is laid out at a grid intersection of an overhead view of the pose space corresponding to the location from which it was obtained; grid intersections where there is no observation correspond to locations in the pose space where the feature was not found in the corresponding training image. Note that the generative nature of the matching mechanism allows the appearance of the feature to evolve significantly over the pose space.

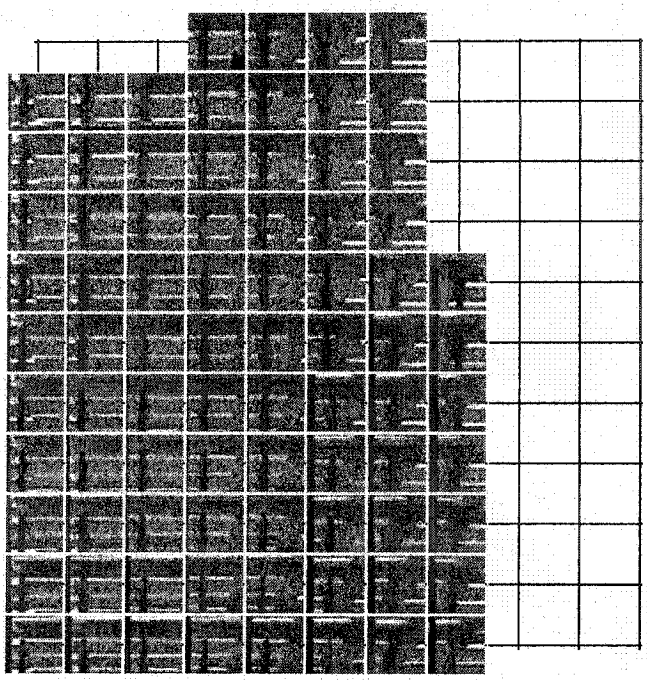
9. The Generative Feature Model

Let us now turn our attention to the problem of inferring a generative feature model. We are interested in learning a pose-dependent model of a scene feature, given a set of observations of the feature from known camera positions. We require that the model will be capable of producing maximum-likelihood virtual observations (predictions) of the feature from previously unvisited poses. It will also be capable of estimating the likelihood $p(\mathbf{z}_i|\mathbf{q})$ of an observation $\mathbf{z}_i$ of feature f_i, given the pose $\mathbf{q}$ from which it might be observed.

An observation $\mathbf{z}$ of a feature f will be represented by the vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{i} \\ \mathbf{x} \end{bmatrix} \tag{2.18}$$

where $\mathbf{x}$ represents the parameters that specify the $[u \ v]$ position of the local-intensity window $\mathbf{i}$ in the image and the intensity window itself is expressed as a vector by "unrolling" it in raster-scan order. While this thesis considers only the position of the feature in the image plane as the space of possible feature transformations- one



(a)

(b)

FIGURE 2.6. a) A set of observations of an extracted scene feature. The grid represents an overhead view of the pose space of the camera, and feature observations are placed at the grid intersection corresponding to the pose where they were observed. Note that the observations capture variation in feature appearance. The lower-left thumbnail is highlighted in the scene, as depicted in Figure b), below.

can also consider rotation and scaling, or any other measurement derived from the observation. The observation $\mathbf{z}$ is a vector-valued function $F(\cdot)$ of the pose of the camera $\mathbf{q}$. The goal is to learn an approximation $\hat{F}(\cdot)$ of this function, as expressed in Equation 2.4. In this thesis, robot poses are assumed to be vectors in a two-dimensional space $\mathbf{q} = [x\ y]^T \in \Re^2$, corresponding to a camera moving through a planar environment at a fixed, but arbitrary, orientation. We will address the problem of orientation recovery in later chapters.

The approach to learning $F(\cdot)$ is to model each element of the feature vector $\mathbf{z} \in \Re^k$ as a linear combination of radial basis functions (RBFs), each of which is centred at a particular robot pose determined by the set of training poses. A radial basis function is any function that exhibits radial symmetry about some central point. In this thesis, an exponentially decaying RBF $G(\cdot,\cdot)$ is used:

$$G(\mathbf{q}, \mathbf{q}_c) = \exp(-\frac{||\mathbf{q} - \mathbf{q}_c||^2}{2\sigma^2}) \tag{2.19}$$

where $\mathbf{q}_c$ represents the centre of the RBF, and the response of the RBF is measured as a function of $\mathbf{q}$. The width, or influence, of the RBF is defined by σ. Figure 2.7 illustrates the response of an RBF centred at the origin and with σ set to one over a two-dimensional pose space.

Given a set of observations, a set of weights $\mathbf{w}_i \in \Re^k$ can be computed such that a linear combination of RBF's interpolates the observations, approximating the function that generated the observations. Formally, given a set of observations from known poses $(\mathbf{z}_i, \mathbf{q}_i)$, a predicted observation $\mathbf{z}$ from pose $\mathbf{q}$ is expressed as

$$\mathbf{z} = \hat{F}(\mathbf{q}) = \sum_i^n \mathbf{w}_i G(\mathbf{q}, \mathbf{q}_i) \tag{2.20}$$

where n is the number of training poses.

The computation of the weight vectors $\mathbf{w}_i$ is well understood in the context of regularization and interpolation theory and is described elsewhere [**145, 93, 45**]. In brief, assuming m observations, n RBF centres, and observation dimensionality k, the optimal weights $W = [w_{ij}] \in \Re^{n \times k}$ are the solution to the linear least squares

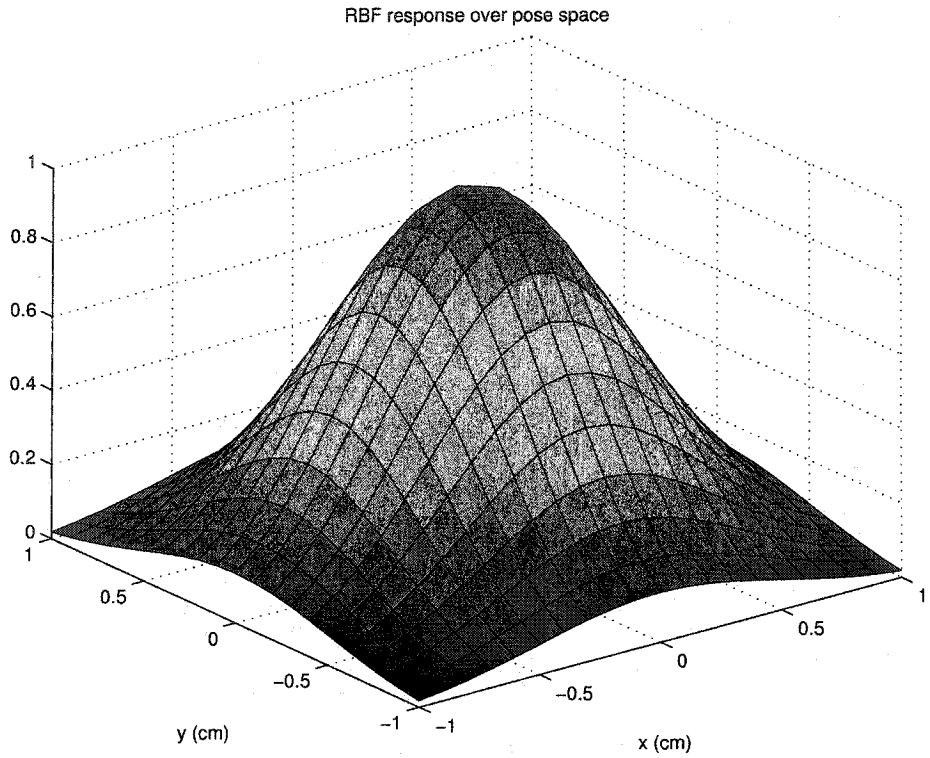


FIGURE 2.7. Radial basis function response as a function of pose.

problem

$$(G + \lambda I)W = Z \tag{2.21}$$

where the elements G_{ij} of the design matrix $G \in \Re^{n \times n}$ correspond to Equation 2.19 evaluated at observation pose i and RBF centre j, $I \in \Re^{m \times n}$ is an identity matrix, and row i of matrix $Z \in \Re^{m \times k}$ corresponds to training observation a pose i. When λ is 0 and G^{-1} exists, the computed weights result in a network whereby Equation 2.20 interpolates the observations exactly. Note that in general for $n < m$, the solution will be over-determined and the presence of noise and outliers as well as the complexity of the underlying function being modelled can result in an interpolation which is highly unstable. The solution can be stabilized by adding a diagonal matrix of regularization

parameters λI to the design matrix G. In this work, these regularization parameters and the RBF width σ are set by hand at the outset[8].

If the design matrix employs every observation pose as a centre for a RBF, the computational cost of computing the weights for n observations is that of an $O(n^3)$ singular value decomposition of an n by n matrix, followed by an $O(n)$ back-substitution for each element of the feature vector $\mathbf{z}$. For computational savings, at the cost of reduced accuracy, the number of RBF centres can be limited to a subset of the observation poses. In practice, we limit the maximum number of centres to 25, and select the centres from the observation poses by ensuring that they cover the pose space uniformly. The drawback to this approach is that features that are visible over a large portion of the pose space may be limited in terms of accuracy.

Figure 2.8 depicts three generated instances of the same feature from different poses. The predicted feature image $\mathbf{i}$ is plotted at the predicted image location $\mathbf{x}$. Note the variation in both appearance and position of the feature in the image.

10. Visibility

As the robot or other observer moves through the environment, features will move in and out of view due to both camera geometry and occlusion. Therefore it is valuable to explicitly model feature *visibility*; that is, whether or not a particular feature is visible from a particular location in pose-space. This information aids the task of localization and is important for the problem of reconstructing the scene. The same regularization framework presented in the previous section is employed to learn a visibility likelihood function $p(visible(f)|\mathbf{q})$, training the function with the binary-valued observability of each feature from each visited pose in the training set[9]. This information is also useful for deciding where to collect new training examples.

[8]For the experiments presented here, $\lambda = 0.01$ and $\sigma = 2D/\sqrt{2M}$ where D is the maximal distance between any two poses in the training set and M is the number of training poses. Furthermore, while ridge regression can be employed to compute the optimal regularization parameters, experience indicates that this approach is not necessary for the distributions of measurements that are being interpolated [47].

[9]The computed RBF model could produce likelihood values less than zero or greater than one– these outputs are clamped when they occur.

11. Model Evaluation

Given an observation $\mathbf{z}_i$ of feature f_i, we can compute the likelihood that it came from pose $\mathbf{q}$, $p(\mathbf{z}_i|\mathbf{q})$ by computing a maximum likelihood observation $\mathbf{z}^*$ using the generative model (Equation 2.20) and comparing the actual and predicted observations using some distance metric $\ll \mathbf{z}, \mathbf{z}^* \gg$. It is not clear, however, how a metric in the space of observations should be defined (recall that an observation is a combination of pixel intensities and transformation parameters). Nor is it clear that the observation space is smooth or continuous. Furthermore, how does the likelihood behave as a function of the metric? In the absence of a more elaborate sensor model, the computed feature models are evaluated using leave-one-out cross-validation, and the feature observation likelihood $p(\mathbf{z}_i|\mathbf{q})$ is modelled as a Gaussian with a covariance R defined by the cross-validation covariance [**90, 153, 59**]. The Gaussian model represents a first-order approximation of the underlying stochastic process that leads to feature observations.

Cross-validation operates by constructing the model with one observation point excluded, predicting that data point using the construction and measuring the difference between the actual point and the prediction. By iterating over several (ideally all) of the training data, and computing the covariance R of the resulting errors, we can build up a measure of how well the model fits the data and, more importantly, how well we might expect it to predict new observations.

Given the high dimensionality of the observation space, the covariance R computed over a Euclidean metric over the observation space will be rank-deficient[10]. This poses problems for numerical stability in the presence of noisy observations. To overcome this problem, the cross-validation covariance R is computed over the 3-space defined by

$$\mathbf{z}_e = \begin{bmatrix} ||\mathbf{i} - \mathbf{i}^*||_2 \\ \mathbf{x} - \mathbf{x}^* \end{bmatrix} \tag{2.22}$$

[10]If the observation space is n dimensional, and there are k observations where $k < n$, then the k cross-validation error vectors lie in a manifold that spans at most a k dimensional space.

where $\mathbf{i}^*$ and $\mathbf{x}^*$ are the intensity and image position components of the maximum-likelihood prediction computed from

$$\mathbf{z}^* = \hat{F}_i^{j-}(\mathbf{q}_j) \tag{2.23}$$

where $\hat{F}_i^{j-}(\mathbf{q}_j)$ is the RBF model for the feature trained with observation j removed and subsequently evaluated at pose $\mathbf{q}_j$. The cross-validation covariance R is then defined as

$$R = \frac{1}{k}\sum_{j=1}^{k} \mathbf{z}_e \mathbf{z}_e^T \tag{2.24}$$

where k is the number of observations of the feature and $\mathbf{z}_e$ is measured for each removed observation j.

Given R, the observation likelihood function $p(\mathbf{z}_i|\mathbf{q})$ is then expressed as a Gaussian distribution:

$$p(\mathbf{z}_i|\mathbf{q}) = c\exp(-0.5\mathbf{z}_e^T R^{-1}\mathbf{z}_e) \tag{2.25}$$

where $c = ((2\pi)^M|R|)^{-1/2}$, M is the dimensionality of the transformed observation space, $|R|$ is the determinant of R, and $\exp(x) = e^x$.

The covariance R is not only useful as a model parameter, but is also a useful measure of model fit. Trained features whose model covariance has a large determinant can be eliminated from the set of features on the basis that the feature is not modelled well and will not be useful for feature reconstruction or robot localization[11].

12. Discussion

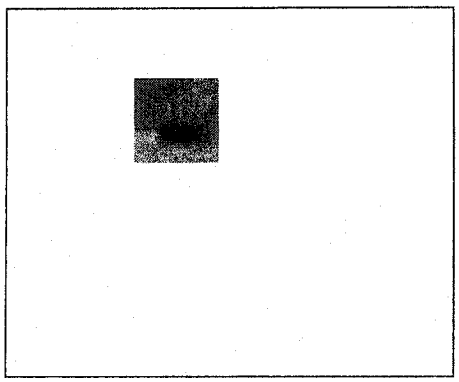
At this point, we have defined all of the elements involved in learning a set of generative models of visual features from an ensemble of training images collected from known robot poses. The models enable the representation of an observation model $p(\mathbf{z}|\mathbf{q})$, which is useful for pose inference and scene reconstruction. The learning framework operates by selecting candidate features using a visual saliency operator,

[11]Another useful measure is the cross-validation error, $e = \sum_j ||\mathbf{z}_e||^2$. However e depends also on the number of observations k, and empirical experience indicates that $|R|$ is more reliable.

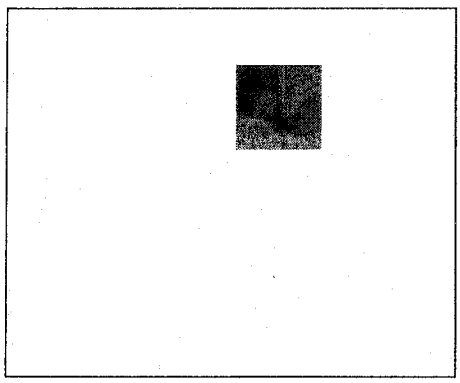
matching image features over a set of training images, and learning a generating function parameterized by the pose of the camera which can produce maximum likelihood feature observations. A radial basis function network is trained for modelling each feature. The system also models the uncertainty of the generated features, allowing for Bayesian inference of camera pose.

The visual mapping framework enables a robot to explore its environment, acquiring an ensemble of observations with a metrically calibrated pose estimate, and then produce a set of generative models of visual features extracted from the observations, selecting those features that appear to be stable. An important property of the framework is that it minimizes *a priori* assumptions about the features being modelled, thus enabling the capture and representation of a wide variety of visual phenomena while employing an arbitrary imaging apparatus.

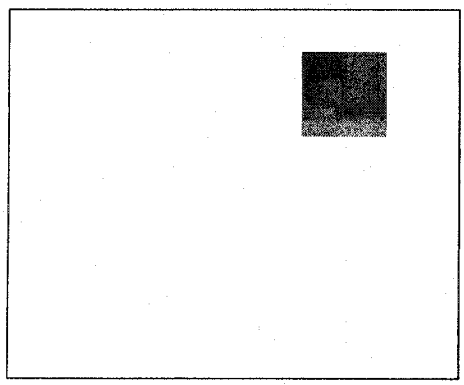
The next chapter will present an implementation of the learning framework in a variety of visual environments and examine its properties when applied to a variety of visual tasks.



(a)



(b)



(c)

FIGURE 2.8. A single feature as generated and rendered from three different camera positions.

CHAPTER 3

Visual Maps: Applications and Experimental Results

1. Overview

The real benefit of constructing a generative model is the ability to predict observations from an arbitrary pose. This ability enables a wide variety of tasks, including model evaluation, scene reconstruction from previously unvisited poses, and robot navigation and localization. This chapter will present experimental results illustrating the implementation of the visual mapping framework, and consider applications of the framework to these problems. Specifically, the problems of model evaluation, scene evaluation, scene reconstruction, and localization will be considered. We will also compare the performance of the mapping framework with that of a common method that infers pose from global image properties.

2. Model Evaluation

The first experiment will evaluate the feature model, both in terms of what attributes are used in the feature representation, and in terms of the reliability of the generative framework used to interpolate the observations. The cross-validation error associated with a particular learned feature represents a measure of the reliability of the model and the feature it has tracked. In order to examine the reliability of

TABLE 3.1. Training set statistics for Scenes I, II and III.

Attribute	Scene		
	I	II	III
Training images	256	121	121
Pose space (cm)	30x30	10x10	200x200
Ground truth accuracy (cm)	0.05	0.05	0.5
Sample spacing (cm)	2	1	20
Features	123	124	125

the representation– that is, the attributes that are modelled generatively, the feature models are cross-validated over individual attributes, such as the image position $\mathbf{x}$, intensity image $\mathbf{i}$, and the edge distribution $\mathbf{e}$ of $\mathbf{i}$.

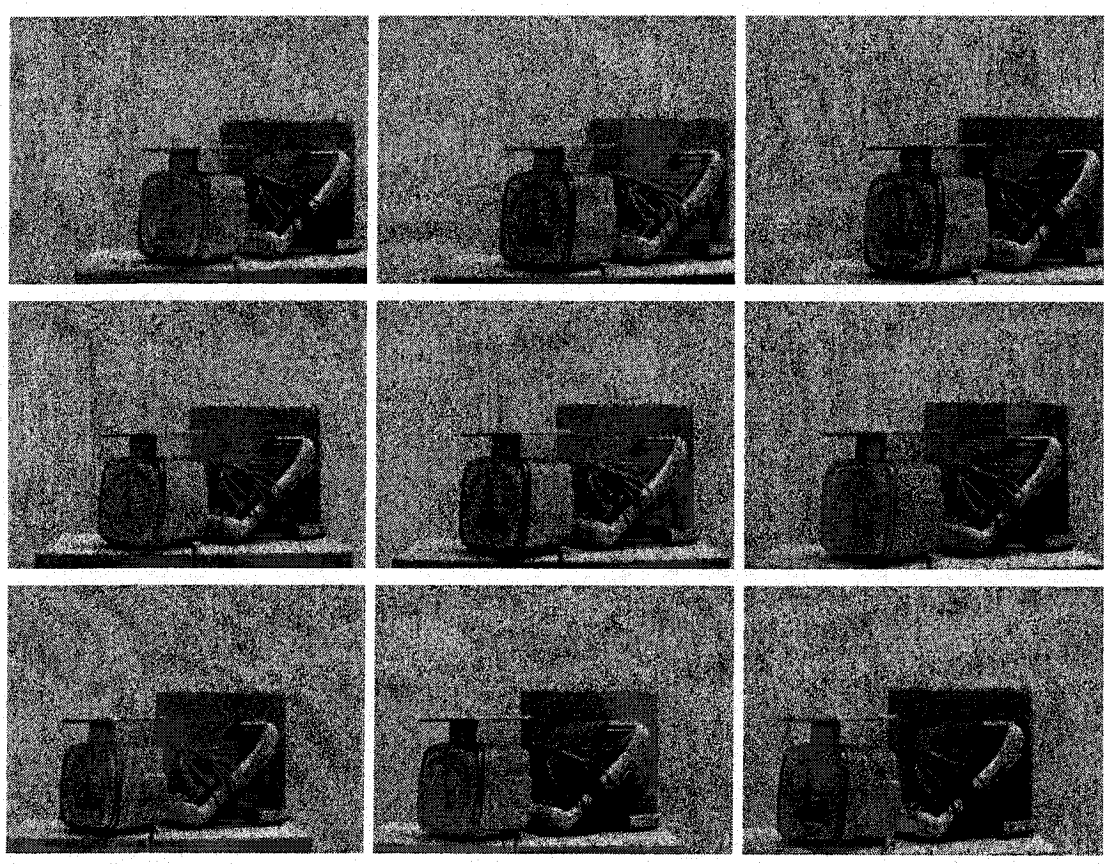


FIGURE 3.1. Images from Scene I used for evaluating feature attributes.

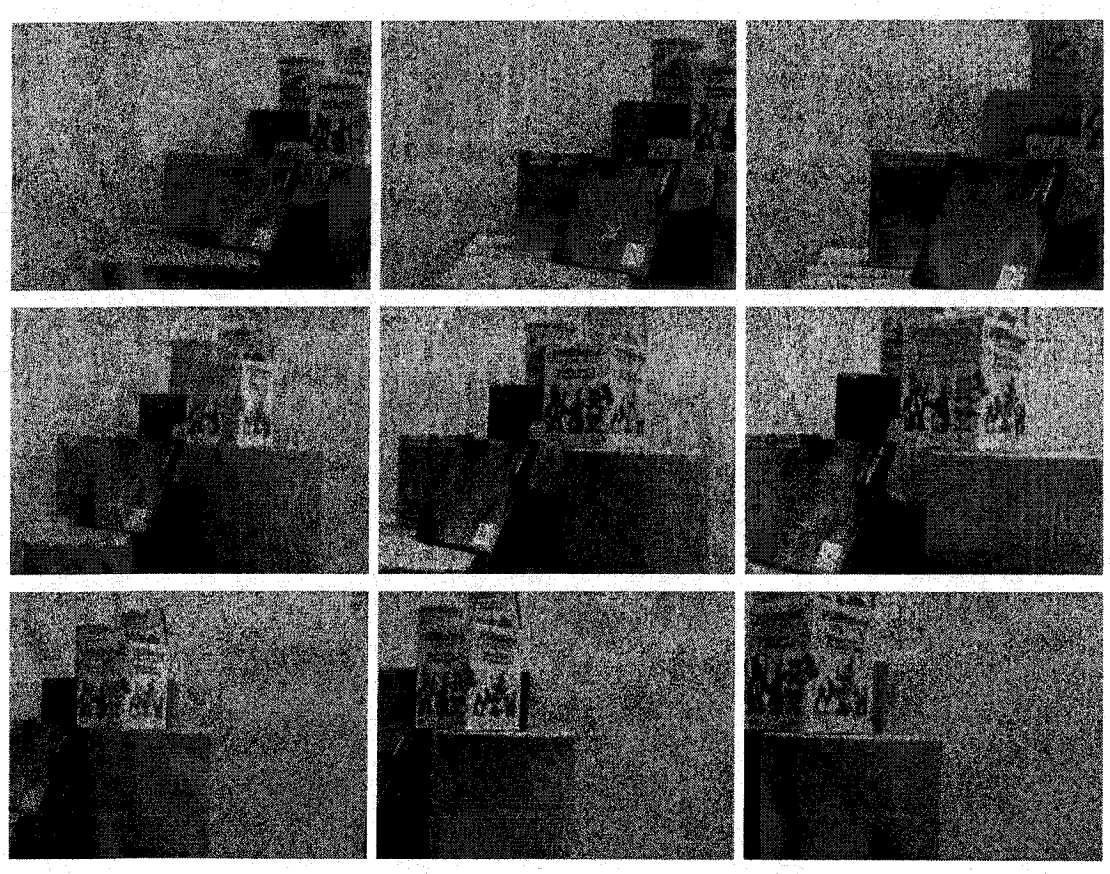


FIGURE 3.2. Images from Scene II used for evaluating feature attributes.

The learning framework was applied to three scenes and the resulting models were evaluated. Figures 3.1, 3.2 and 3.3 depict a selection of training images from the scenes under consideration and Table 3.1 indicates number of training images, pose-space geometry and approximate accuracy of the ground-truth pose information for each scene. In all three cases, the pose space was a square grid in which the robot collected samples at uniform intervals. In addition, the orientation of the camera was held constant. Scenes I and II were obtained using a camera mounted on the end-effector of a gantry robot arm, with a positioning accuracy of approximately 0.05cm. Scene III was collected using a camera mounted on a Nomad 200 robot (Figure 3.4). A laser-pointer was mounted on the robot to point at the floor and the robot's position was measured by hand based on the position of the laser point on the

FIGURE 3.3. Images from Scene III used for evaluating feature attributes.

floor. Notwithstanding human error, the accuracy of these measurements is about 0.5cm.

As an illustrative example, the distribution of feature cross-validation covariances for Scene III is depicted in Figure 3.5. Of 125 modelled features, 56 contained more than four observations. For each of these, the cross-validation covariance R was computed, along with its determinant. Each bar in the histogram corresponds to the number of features for which $\log |R|$ falls within the range defined by the horizontal axis.

Figure 3.6 depicts the five best features modelled for Scene III, according to the determinant of the cross-validation covariance. Each image represents an overhead view of the pose space and the feature observations are plotted at their corresponding

FIGURE 3.4. The Nomad 200 mobile robot

poses. Similarly, Figure 3.7 depicts the five worst features, all of which were rejected by the learning framework. Note that the more reliable features tend to track well over small regions of the pose space, whereas the unreliable features demonstrate failures in tracking whereby different parts of the scene were matched to the same feature, resulting in significant modelling errors.

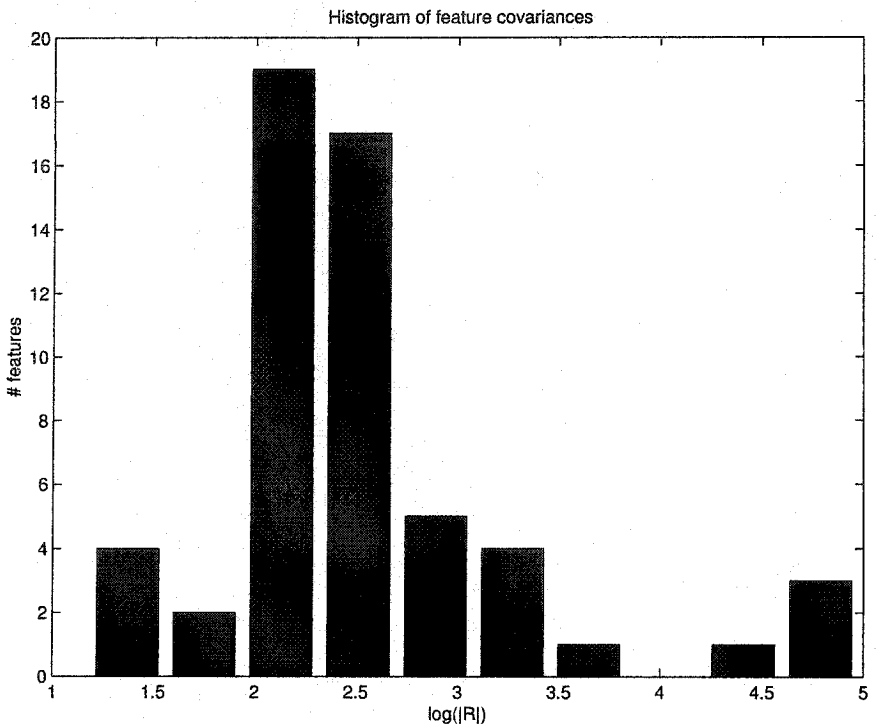


FIGURE 3.5. The distribution of feature cross-validation covariances for Scene III. Each bar tabulates the number of features out of 56 for which $\log |R|$ falls into the range defined by the horizontal axis.

TABLE 3.2. Mean cross-validation error by feature attribute.

Feature Attribute	Scene		
	I	II	III
Position $\mathbf{x}$ (pixels^2)	1.7×10^1	1.4×10^2	3.3×10^1
Intensity Distribution $\mathbf{i}$ (intensity^2)	2.1×10^3	5.2×10^3	2.5×10^3
Edge Distribution $\mathbf{e}$ (intensity^2)	1.7×10^4	2.2×10^4	1.3×10^4

Table 3.2 summarizes the quality of the Position, Intensity, and Edge Distribution attributes mentioned above for the three scenes under consideration, based on the mean cross-validation error. The results are also depicted graphically in Figure 3.8. For any given feature, the image position, intensity distribution and edge distribution of the features were each used to generate a separate model and cross-validation error. Tabulated are the mean cross-validation error of these properties over all observed

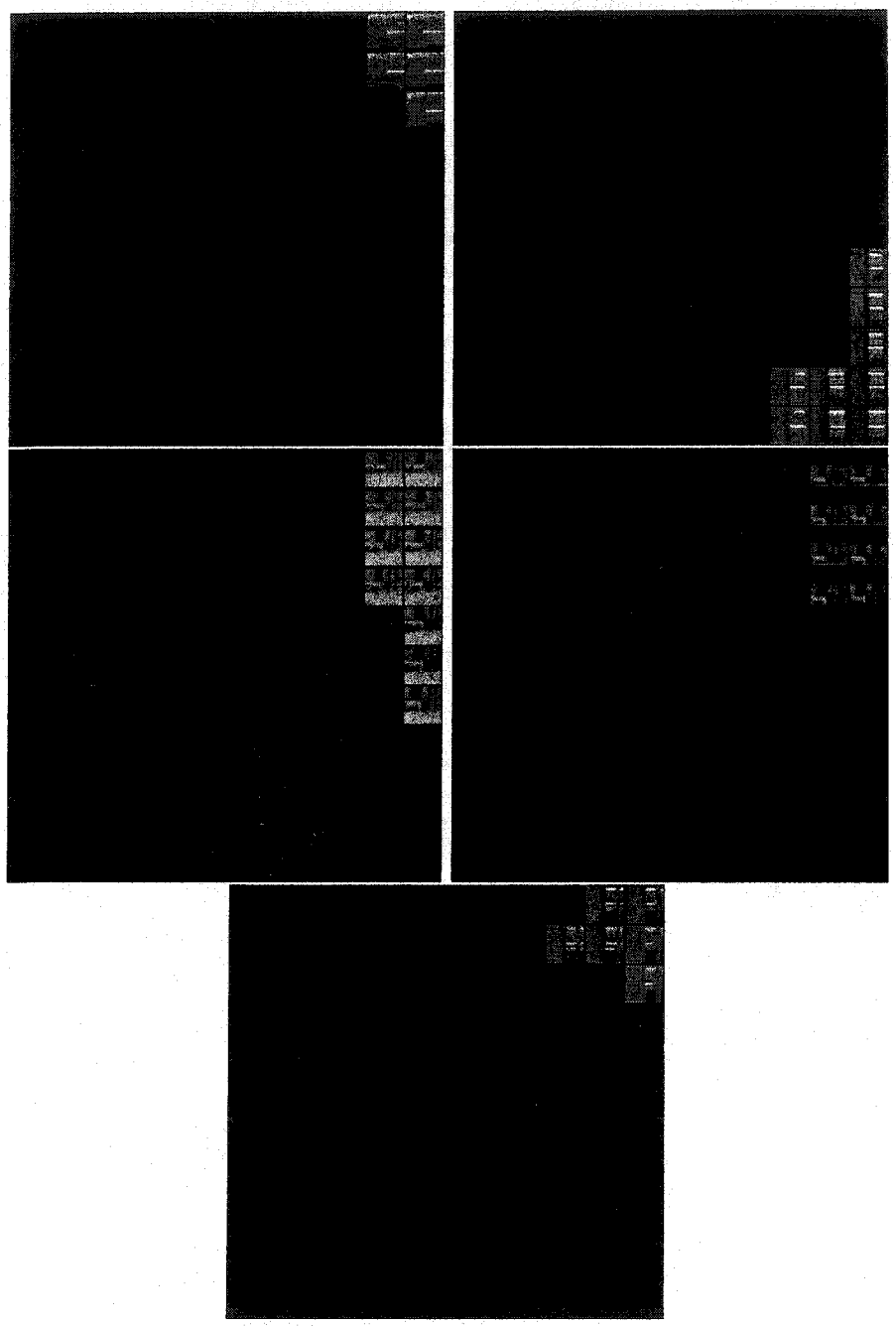


FIGURE 3.6. The five best features modelled for Scene III, by determinant of the cross-validation covariance. Each small thumbnail corresponds to an observation of the feature from the pose corresponding to the thumbnail's position in the image. Dark regions correspond to poses where the feature was not observed.

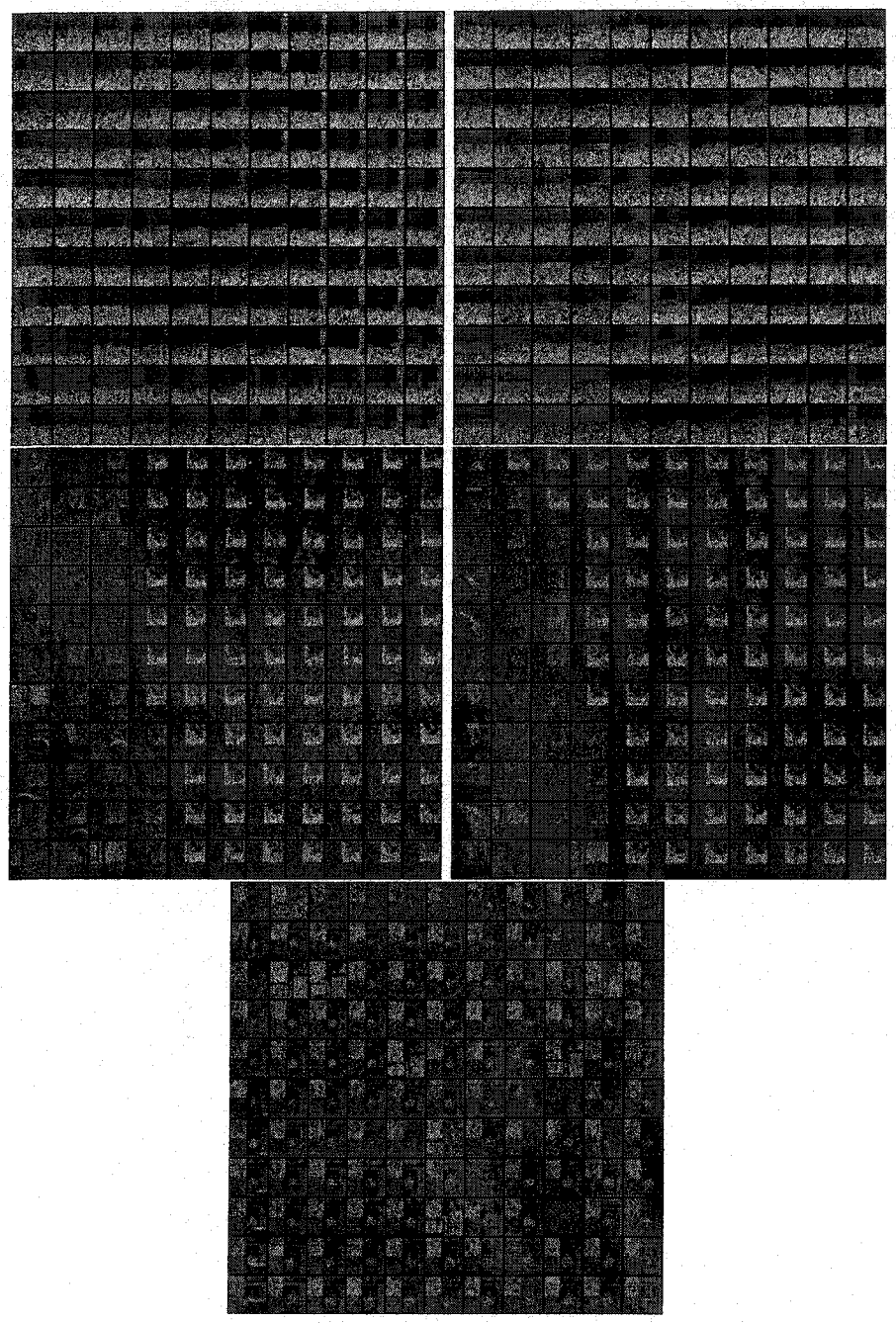


FIGURE 3.7. The five worst features modelled for Scene III, by determinant of the cross-validation covariance. Each small thumbnail corresponds to an observation of the feature from the pose corresponding to the thumbnail's position in the image..

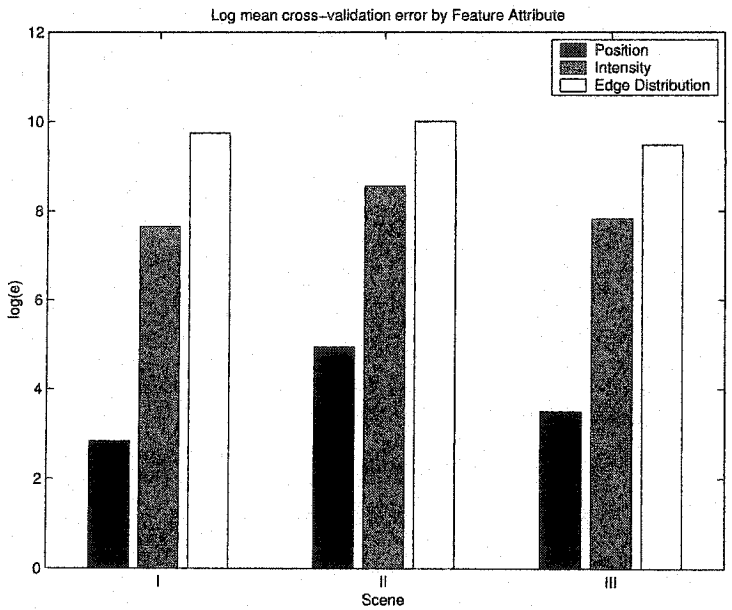


FIGURE 3.8. Log of the cross-validation error e by attribute for each scene. Units are in log pixels for image position, and ln intensities $\in [0, 6]$ for the appearance and edge distribution.

features. Therefore, the smaller the value, the more reliable the attribute can be considered to be.

It is interesting to note that the image position of the feature in the image is in general the most accurately modelled whereas the edge distribution is poorly modelled. The order-of-magnitude difference between the position error and the intensity distribution error is due in part to the significant difference in the dimensionality of the attributes. One can conclude that for the purposes of inference, in most circumstances the image position will be the most useful attribute.

3. Scene Evaluation

In addition to measuring feature quality, it is also possible to evaluate the ability of the model to represent the environment as a function of pose. This is accomplished by computing a quality estimate for the subset of features observable from a given

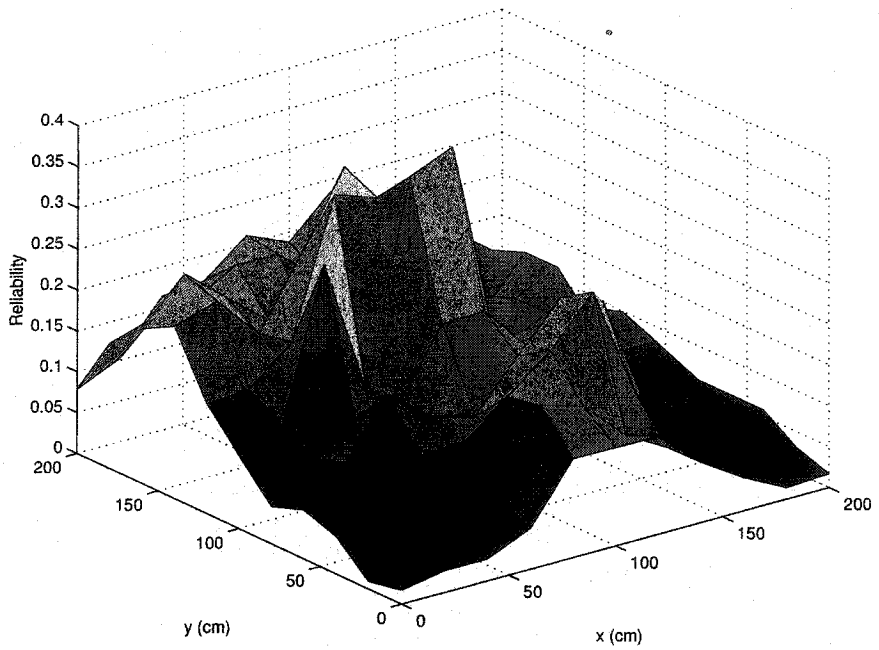


FIGURE 3.9. *A priori* training reliability R as a function of pose for scene III. The camera faces in the negative y direction.

position. At each training pose $\mathbf{q}$, we can compute a measure of reliability

$$R_{\mathbf{q}} = \sum_{f_i \in \Gamma} \frac{1}{|R_{f_i}|} \tag{3.1}$$

where Γ is the set of features which are observed from pose $\mathbf{q}$, and $|R_{f_i}|$ is the determinant of the feature cross-validation covariance R_{f_i}. Note that for poses other than the training poses, a similar measure can be computed by weighting the terms of $R_{\mathbf{q}}$ by their visibility likelihood, $p(visible(f)|\mathbf{q})$. The determinant of the feature covariance is an indication of how weak the pose constraint for a given feature may be. Clearly, $R_{\mathbf{q}} \in (0, \infty)$ and larger values should predict more reliable pose estimates. Figure 3.9 plots $R_{\mathbf{q}}$ as a function of pose for Scene III. In this plot, the orientation of the camera is fixed to face in the negative y direction while the robot moves over a 2m by 2m pose space. Note that the reliability is particularly low for small values of y. This is due to the fact that images in that region of the pose space change dramatically under small changes in pose, leading to difficulty in tracking the features.

4. Scene Reconstruction

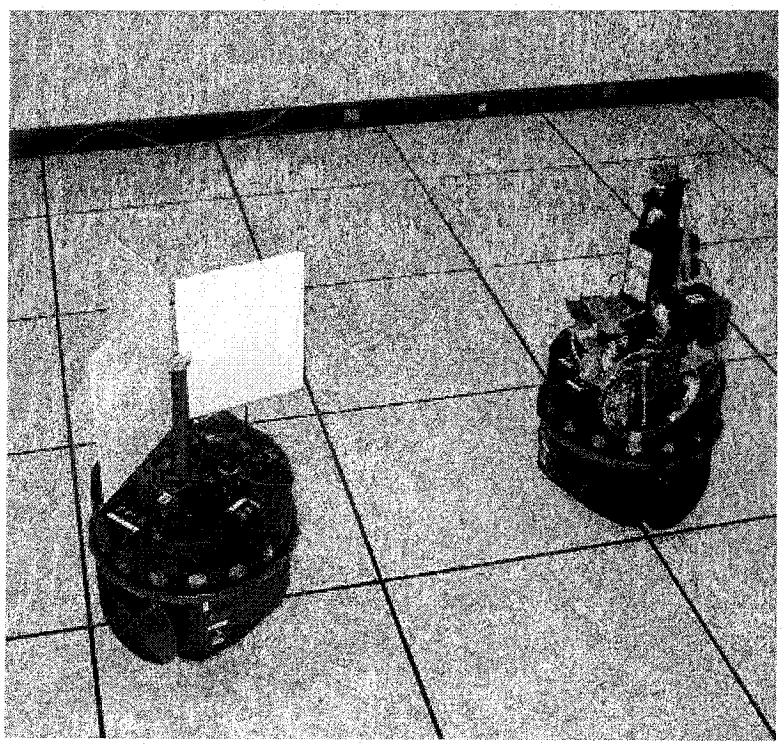


FIGURE 3.10. Robots employed for data collection. The three-plane target mounted on the exploring robot is sensed by the stationary robot, allowing for the computation of pose estimates for the explorer. The pose estimates are employed as an approximation to ground-truth, both for training and evaluating the vision-based localizer.

Given a set of trained features and a particular pose $\mathbf{q}$, one can generate a maximum likelihood reconstruction of the scene features. Given the generated observations, the full image is reconstructed by positioning each feature according to its predicted transformation and mapping the generated intensity image in the image neighbourhood:

$$\mathbf{z}^* = \hat{F}_i(\mathbf{q}). \tag{3.2}$$

In this experiment a large image neighbourhood (61 by 61 pixels) is modelled for each feature in order to predict as much of the image as possible. Where predicted feature windows overlap, the pixel intensity $I(\mathbf{x})$ is determined by selecting the pixel

intensity corresponding to the feature that maximizes the weighting function

$$v = \frac{p(visible(f)|\mathbf{q})}{|R_f|} e^{-\frac{\Delta \mathbf{x}^2}{2\sigma^2}} \tag{3.3}$$

where R_f is the cross-validation covariance for the feature, $\Delta \mathbf{x}$ is the Euclidean distance $||\mathbf{x} - \mathbf{x}^*||_2$ between the pixel $\mathbf{x}$ and the predicted position of the feature $\mathbf{x}^*$, and $\sigma = 30$ pixels is a parameter describing the region of influence of each feature in the image. This winner-takes-all strategy selects a feature whose pixel prediction has the highest confidence, and paints the pixel with that prediction.

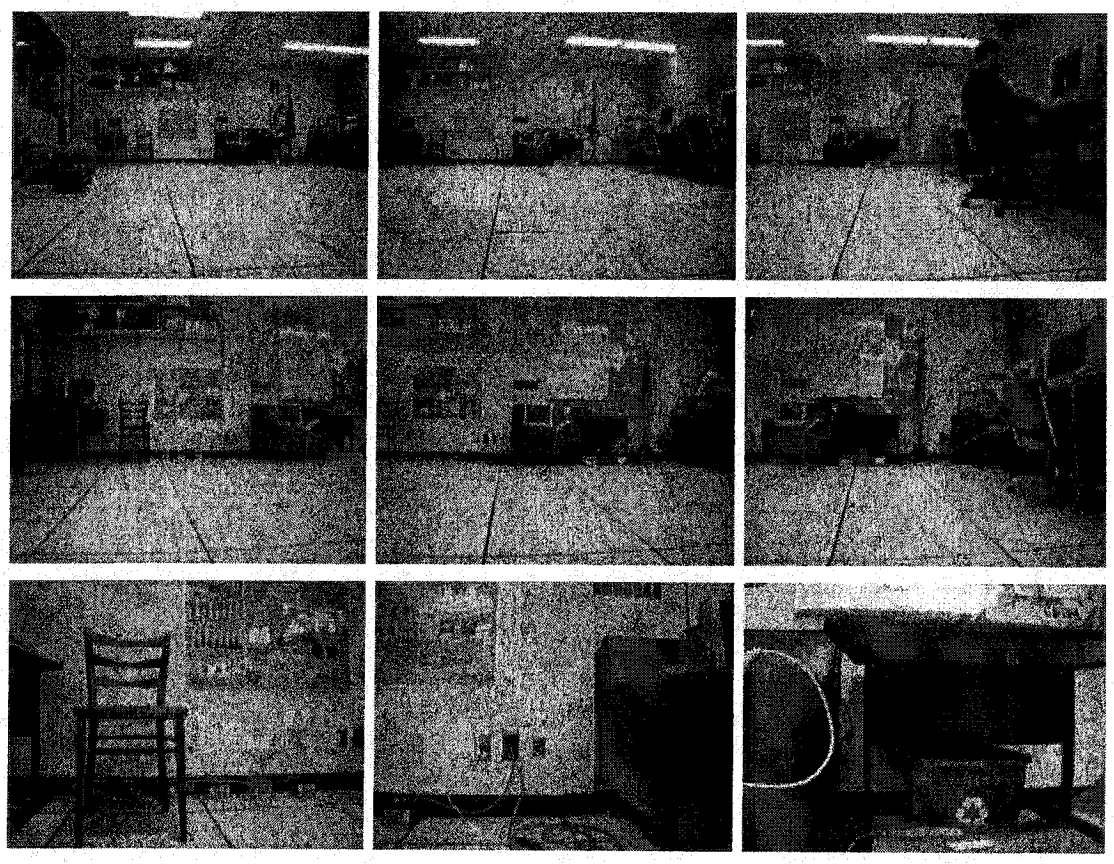


FIGURE 3.11. Images from Scene IV used for demonstrating scene reconstruction.

For this experiment, a new training set, Scene IV, was collected (Figure 3.11). This environment was explored by taking 291 training images at uniform intervals of approximately 25cm over a 3.0m by 6.0m pose space. A second observing robot

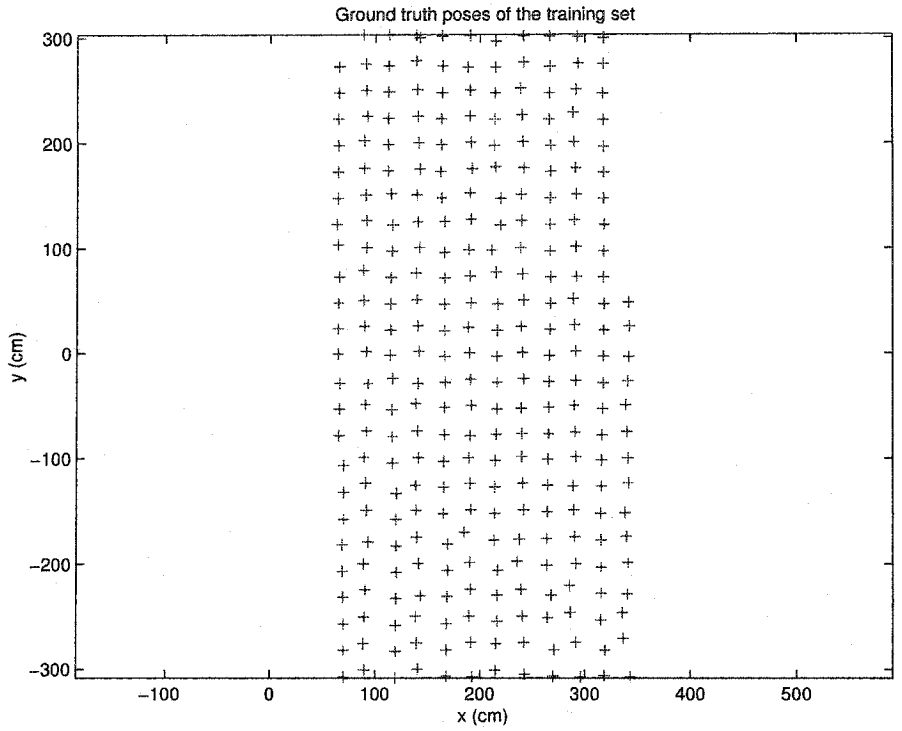


FIGURE 3.12. Ground truth poses of the Scene IV training set, as measured by the robot tracker

equipped with a laser tracking system was deployed to estimate the ground-truth position of the exploring robot to an accuracy of approximately 4cm. The implementation of the laser tracking system is described in [**103, 102**]. The observer employed a laser range-finder to accurately determine the exploring robot's pose from the range and orientation of a three-plane target mounted on the exploring robot (Figure 3.10). For the purposes of this scene, the robot attempted to take training images at the same global orientation. However, uncertainty in the robot's odometry, as well as the observing robot's estimate, led to some variation in this orientation from pose to pose. The set of ground truth training poses is depicted in Figure 3.12.

Given the training set, a visual map was constructed, extracting 325 feature models from the scene. Subsequently, a set of images was rendered from user-selected poses using the scene reconstruction method. Figure 3.13a shows a training image from the scene, and Figure 3.13b depicts the reconstruction of the scene from a nearby

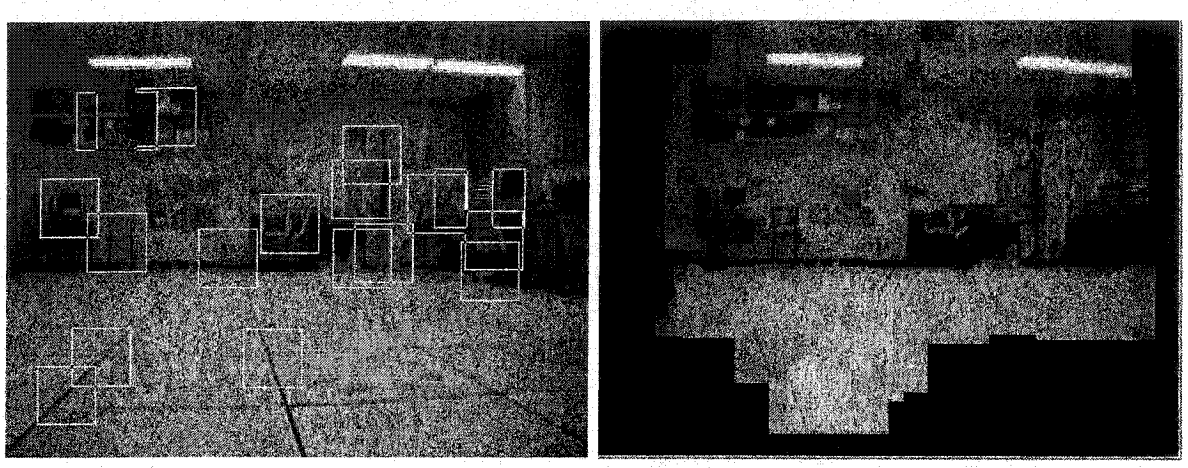


FIGURE 3.13. a) Training image b) A reconstruction of the scene, as predicted from a nearby camera pose.

pose. Several other reconstructed frames of the scene are depicted in Figure 3.14. Note that the reconstruction cannot predict pixels for which there is no visible feature model, and as such, the lower edge of the image is left unshaded. It may be that these regions can be shaded by extrapolating from the nearby texture using Markovian reconstruction methods [**34, 37**].

5. Localization

This section will examine the ability of the feature learning framework to produce a map that is useful for pose inference. Given a set of feature models, the task of robot localization can be performed by applying Bayes' Law, as per Equation 2.7. When a pose estimate is desired, an observation is obtained and optimal matches $\mathbf{z} = \{\mathbf{z}_i\}$ to the learned features are detected in the image using the method described in Section 2.8. Each feature observation $\mathbf{z}_i$ then contributes a probability density function $p(\mathbf{z}_i|\mathbf{q})$, which is defined as the product of the distribution due to the maximum likelihood prediction of the model $p(\mathbf{z}_i|\mathbf{q}, visible(f))$ (Equation 2.25) and the feature visibility likelihood $p(visible(f)|\mathbf{q})$:

$$p(\mathbf{z}_i|\mathbf{q}) = p(\mathbf{z}_i|\mathbf{q}, visible(f_i))p(visible(f_i)|\mathbf{q}). \tag{3.4}$$

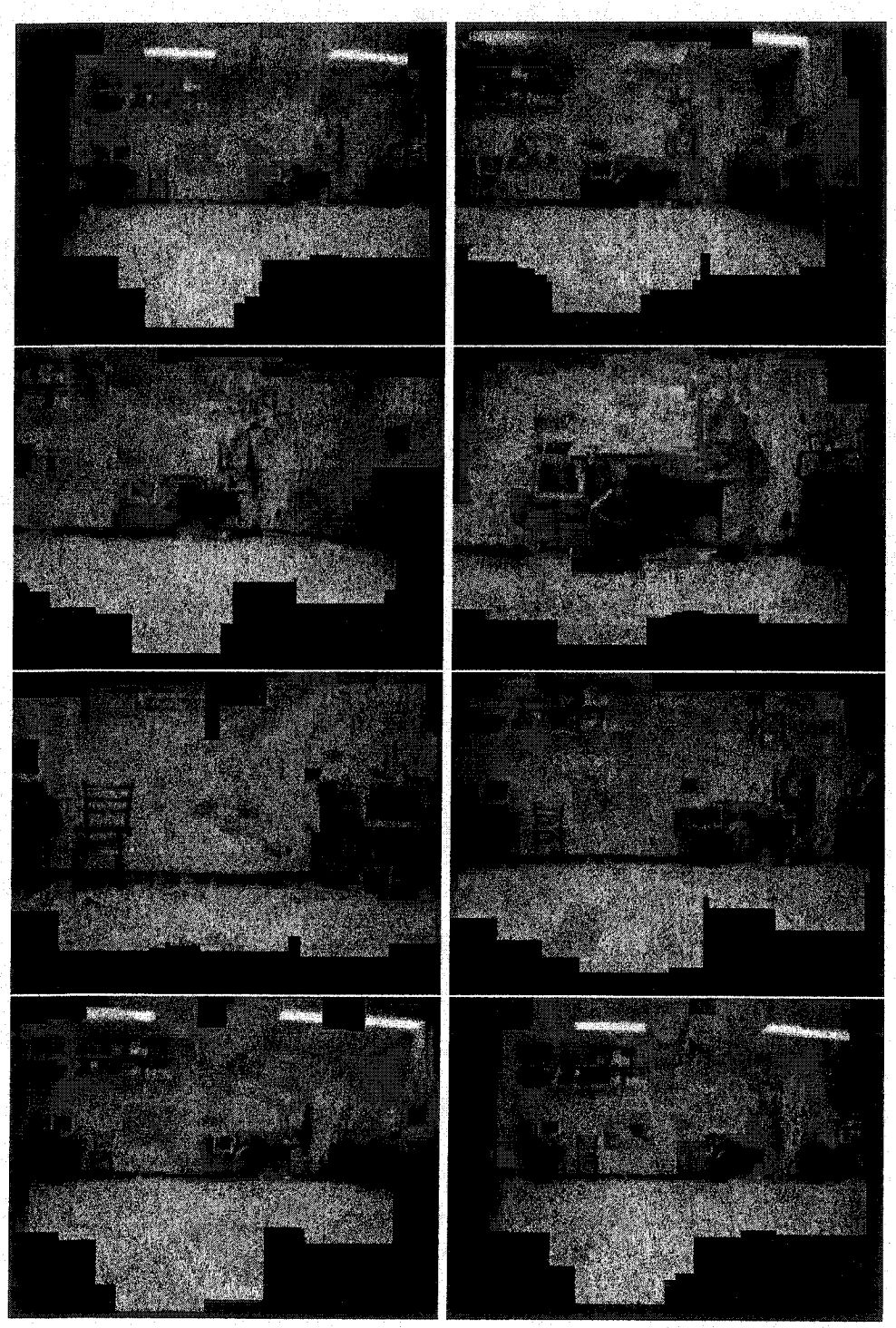


FIGURE 3.14. More reconstructed frames from Scene IV.

Assuming conditional independence between the individual feature observations, the probability of an observed image is defined to be the joint likelihood of the individual observations:

$$p(\mathbf{z}|\mathbf{q}) = \prod_{i=1}^{n} p(\mathbf{z}_i|\mathbf{q}) \tag{3.5}$$

In the absence of informative priors, the pose $\mathbf{q}^*$ that maximizes the joint likelihood of the observations is considered to be the maximum likelihood position of the robot. It is not clear, however, that the conditional independence assumption holds for features derived from a single image and, furthermore, outliers can lead to catastrophic cancellation of the joint distribution. Instead, let us assume that for any feature observation there is a high probability e that it is an outlier. In this case, the probability of an observation can be redefined as a mixture of a uniform distribution and the observation likelihood:

$$p'(\mathbf{z}_i|\mathbf{q}) = \frac{e}{C} + (1-e)p(\mathbf{z}_i|\mathbf{q}) \tag{3.6}$$

where C is the area of the pose space [**79**].

Under this interpretation, for e close to 1, Equation 3.5 can be approximated by a Taylor series expansion:

$$\prod_{i=1}^{n} p'(\mathbf{z}_i|\mathbf{q}) \approx (e/C)^n + (e/C)^{n-1}(1-e)\sum_{i=1}^{n} p(\mathbf{z}_i|\mathbf{q}) \tag{3.7}$$

Hence, maximizing Equation 3.7 is equivalent to maximizing the mixture model defined as the sum of the individual feature likelihoods, and, ignoring the additive constants, the joint likelihood is approximated with this sum:

$$p(\mathbf{z}|\mathbf{q}) \approx \sum_{i=1}^{n} p(\mathbf{z}_i|\mathbf{q}) \tag{3.8}$$

This model takes an extreme outlier approach whereby it is assumed that there is a high probability that any given feature observation is an outlier. Experience indicates that the mixture model provides resistance to outlier matches without the need for computing a full joint posterior, as I will demonstrate in the next section.

For further reading on the application of a summation model in place of a product, refer to the tutorial by Minka [**79**].

In the following experiments, the performance of the learning framework and feature models will be evaluated in terms of their performance in the task of robot localization. The experiments involved Scenes I, II, III and IV, depicted in Figures 3.1, 3.2, 3.3 and 3.11, respectively. For each scene, a set of initial features were extracted from a small subset of the training images, determined to be a uniform sampling of the pose space such that no two images in the subset were less than a scene-dependent threshold apart[1]. From this set of initial features, the models were trained as described in Chapter 2. Those models with large cross-validation error, or with too few observations to construct a useful model, were removed, resulting in a set of reliable feature models. For all four scenes, the intensity $\mathbf{i}$ and position $\mathbf{x}$ of the features were modelled and employed for localization. However, in the case of Scene I, it was found that the feature intensity distribution $\mathbf{i}$ was not informative enough in the "looming direction"[2] and degraded the localization results considerably. This outcome could derive from a variety of factors, such as a low signal-to-noise component in the intensity model as a function of pose, or perhaps over-smoothing the model with respect to the regularization parameter λ. As a result, the results for Scene I were computed a second time wherein only the feature position $\mathbf{x}$ was employed to compute localization estimates. These results are reported as Scene Ib. Note that both the intensity and position attributes were employed for the remaining scenes.

To validate localization performance using the learned models, for each scene an additional set of images was collected from random poses, constrained to lie anywhere within the training space. These validation images were used to compute maximum-likelihood (ML) estimates of the camera's position using Equation 3.8, and these estimates were compared against the ground truth pose information.

[1] 15cm, 15cm, 1m and 1m for Scenes I, II, III and IV, respectively.
[2] The direction parallel to the optical axis of the camera.

The ML estimates themselves were computed by exhaustive search over a multi-resolution discretization of the training space, selecting the $\mathbf{q}$ that maximized Equation 3.8. In particular, the training space was discretized into a 40 by 40 grid covering the entire training space and Equation 3.8 was evaluated at each position in the grid. Subsequently, at the maximal grid location a new 10 by 10 grid was instantiated over a neighbourhood spanning 7 by 7 grid positions in the larger grid and Equation 3.8 was evaluated over the new grid. This process iterated recursively to a resolution of 1% of the intervals between training poses, and the maximal grid pose at the highest resolution was returned. Note that a more efficient estimator, such as Monte Carlo sampling, could be easily deployed for applications where computational resources are limited.

In practical settings, one is not always interested in the ML pose estimate, but sometimes in the entire probability distribution over the pose space, which can provide information about alternative hypotheses in environments which exhibit significant self-similarity. Figure 3.15 depicts the *a posteriori* pose distributions computed for a selection of the Scene III validation images. Each frame in the Figure represents the evaluation of Equation 3.8 computed over a uniform discretization of the 2m by 2m pose space, where darker regions correspond to more likely poses. Figure 3.16 depicts another pose distribution, derived from a Scene IV validation image, in greater detail. The distribution is presented modulo a normalizing constant. The figure clearly indicates a region where the pose is more probable, as well as a second, less probable region. The second region may be due to a mis-classified feature (a failure in the matching stage), or some self-similarity in a trained feature.

Given that each ML estimate has a numerical likelihood, it is possible to reject pose estimates that do not meet a particular confidence threshold. In this way, several estimates in the validation sets were rejected. Interestingly, the majority of these estimates were associated with images that were obtained when the robot was very close to objects in the scene it was facing, where it was difficult to reliably track features at the selected training sample density. This behaviour coincides with that

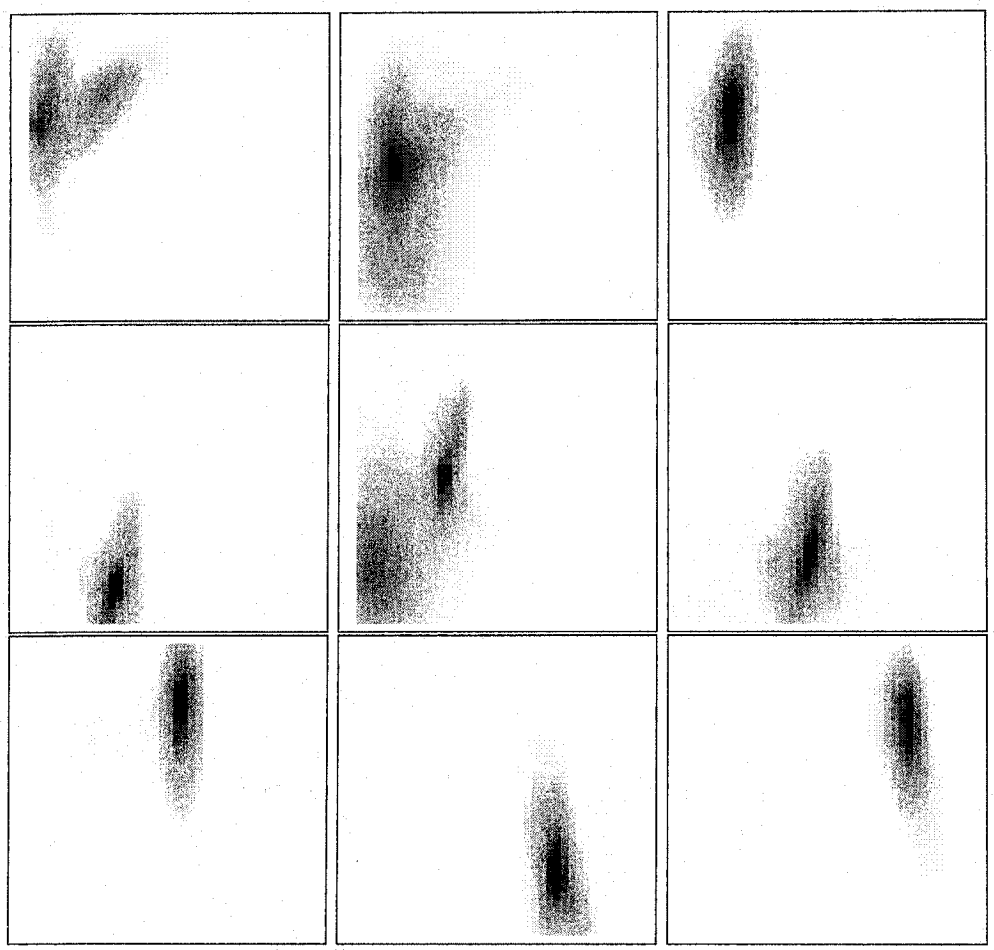


FIGURE 3.15. *A posteriori* pose distributions for a selection of the Scene III validation images. Each image represents an overhead view of the 2m by 2m pose space. Darker regions correspond to more likely poses. Note that several of the distributions are not Gaussian or unimodal.

predicted by our *a priori* evaluation of a similar scene, as exhibited in Figure 3.9, where the reliability measure degrades when the robot approaches the objects in the scene.

Figures 3.17, 3.18, 3.19, 3.20 and 3.21 plot for each scene the location of the unrejected ML estimates for the validation images ('x') against the ground truth camera position ('o') by joining the two points with a line segment for each scene. The length of each line segment corresponds to the magnitude of the error between the corresponding pose estimate and ground truth. The mean absolute error, mean x

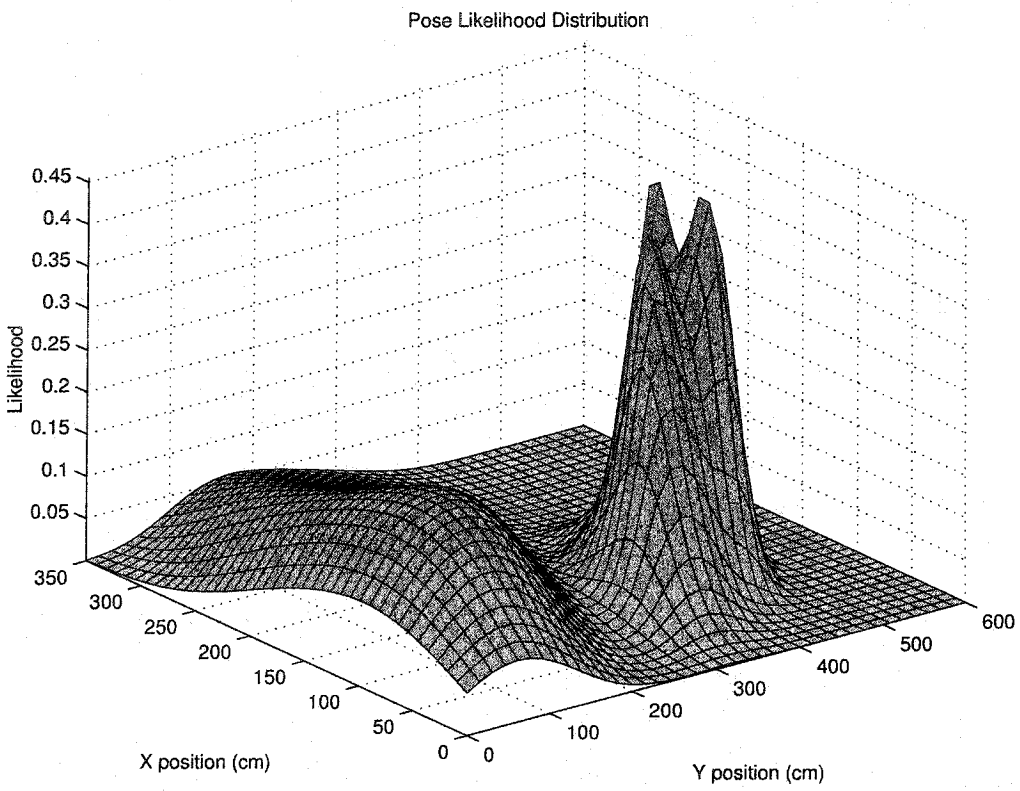


FIGURE 3.16. An example *a posteriori* pose distribution for a validation image from Scene IV.

and y direction errors (corresponding to sideways and looming motion, respectively), minimum and maximum errors and number of retained estimates for each validation set is tabulated in Table 3.3. The larger error in the y (looming) direction corresponds to the fact changes in observations due to forward and backward motion are not as pronounced as changes due to side-to-side motion. This is a well-known result in the ego-motion literature [17]. This is particularly apparent in the results from Scene I.

TABLE 3.3. Summary of Localization Results for Scenes I, II, III and IV.

	Scene				
	I	Ib	II	III	IV
Training images	256	256	121	121	291
Pose space (cm)	30x30	30x30	10x10	200x200	300x600
Ground truth accuracy (cm)	0.05	0.05	0.05	0.5	4.0
Sample spacing (cm)	2	2	1	20	25
Validation Images	100	100	20	29	93
Valid Pose Estimates	99	100	19	28	89
Mean Error (cm)	3.05	0.37	0.19	6.8	17
Mean x Error (cm)	0.57	0.09	0.04	3.6	7.7
Mean y Error (cm)	3.0	0.35	0.17	5.1	14
Minimum Error (cm)	0.08	0.020	0.035	0.64	0.49
Maximum Error (cm)	18.8	1.6	0.73	13.1	76

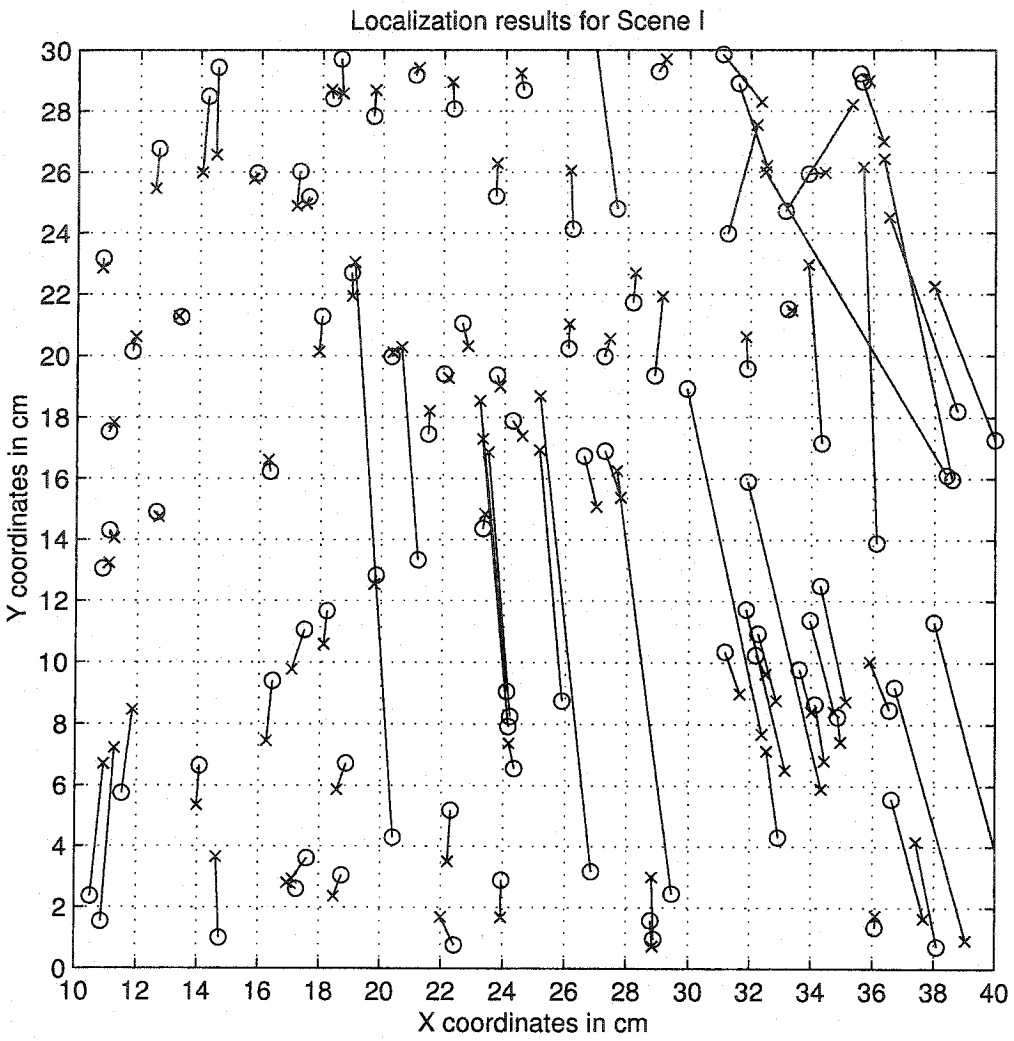


FIGURE 3.17. Localization results for Scene I using Appearance and Image Position as pose indicators. The set of maximum-likelihood pose estimates ('x') plotted against their ground truth estimates ('o'). For this data set, the appearance distributions were found to be non-informative and degraded the results. An alternative estimate, using only Image Position is presented in Scene Ib, below.

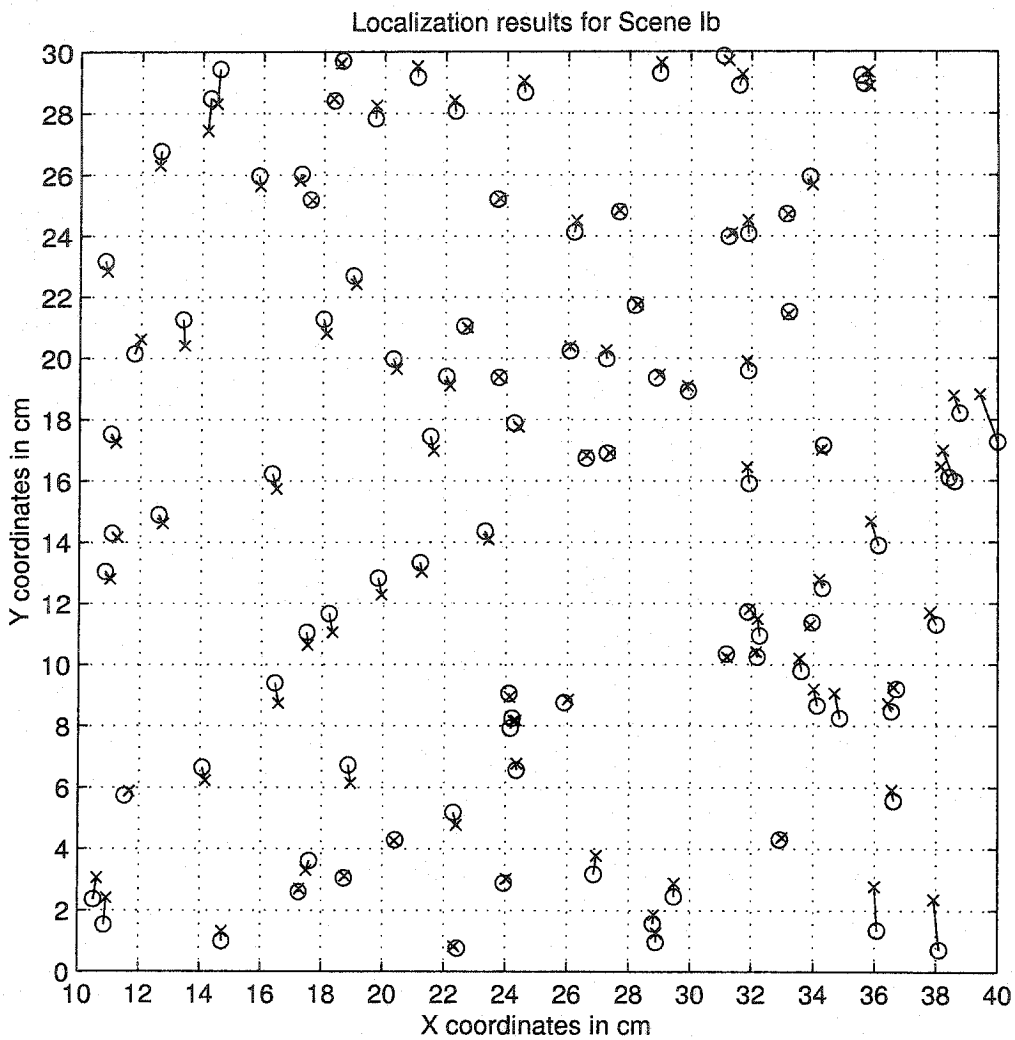


FIGURE 3.18. Localization results for Scene Ib using only Image Position as a pose indicator. The set of maximum-likelihood pose estimates ('x') plotted against their ground truth estimates ('o'). Mean error: 0.37cm

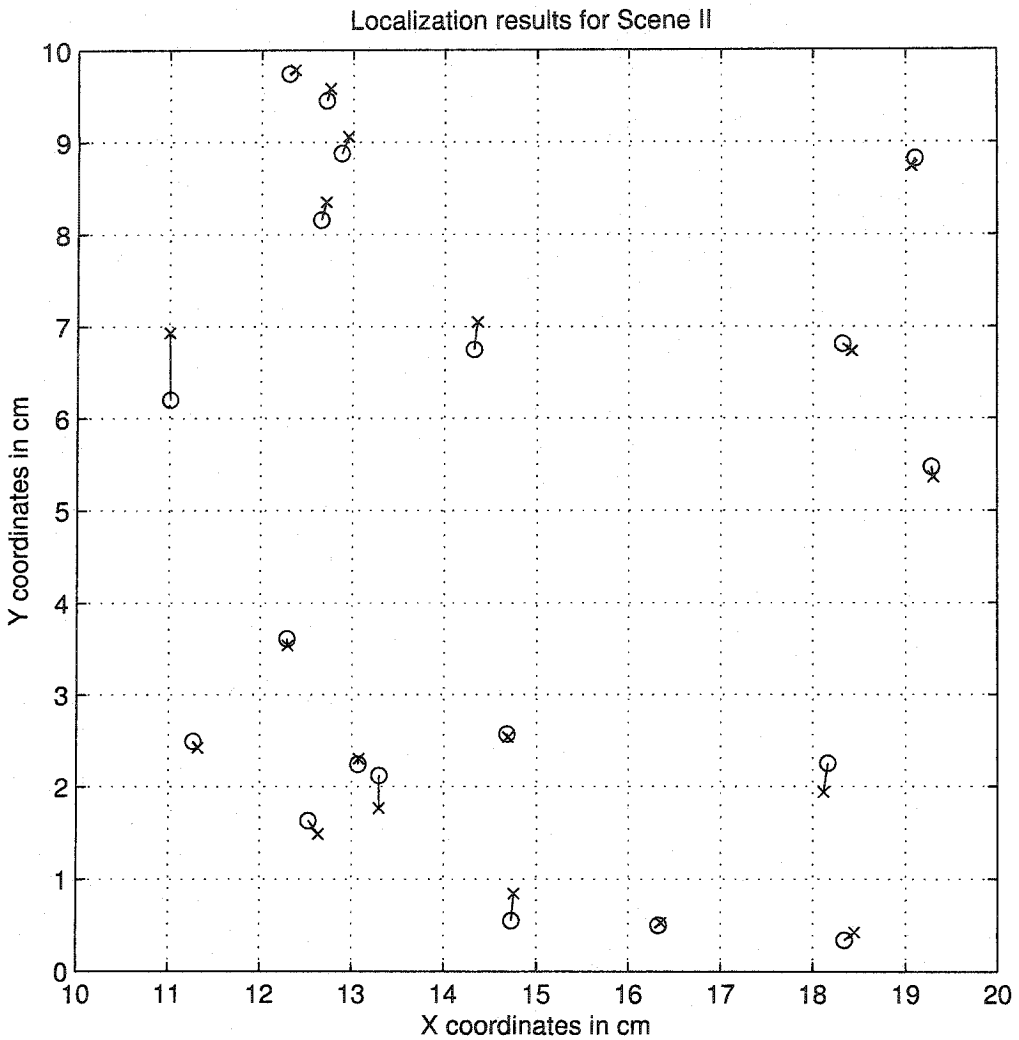


FIGURE 3.19. Localization results for Scene II: the set of maximum-likelihood pose estimates ('x') plotted against their ground truth estimates ('o'). Mean error: 0.19cm

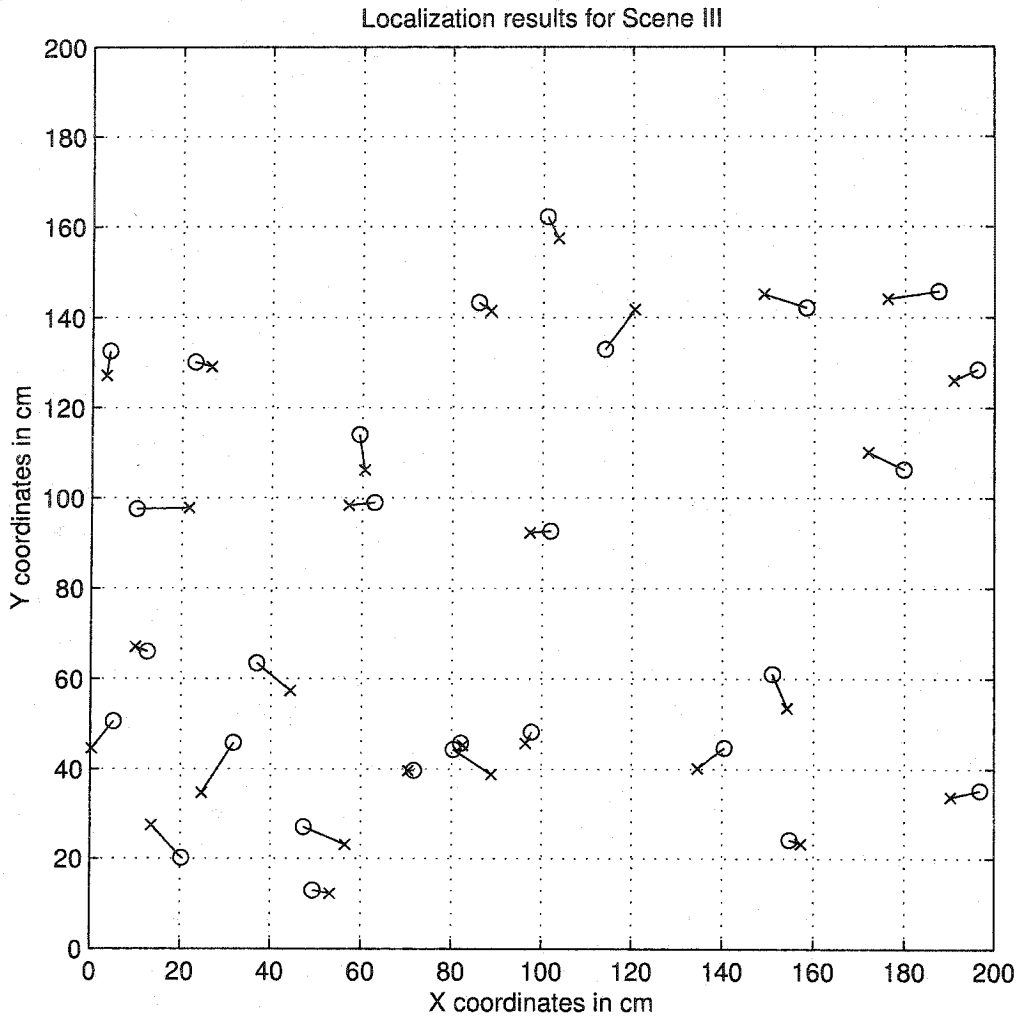


FIGURE 3.20. Localization results for Scene III: the set of maximum-likelihood pose estimates ('x') plotted against their ground truth estimates ('o'). Mean error: 6.8cm

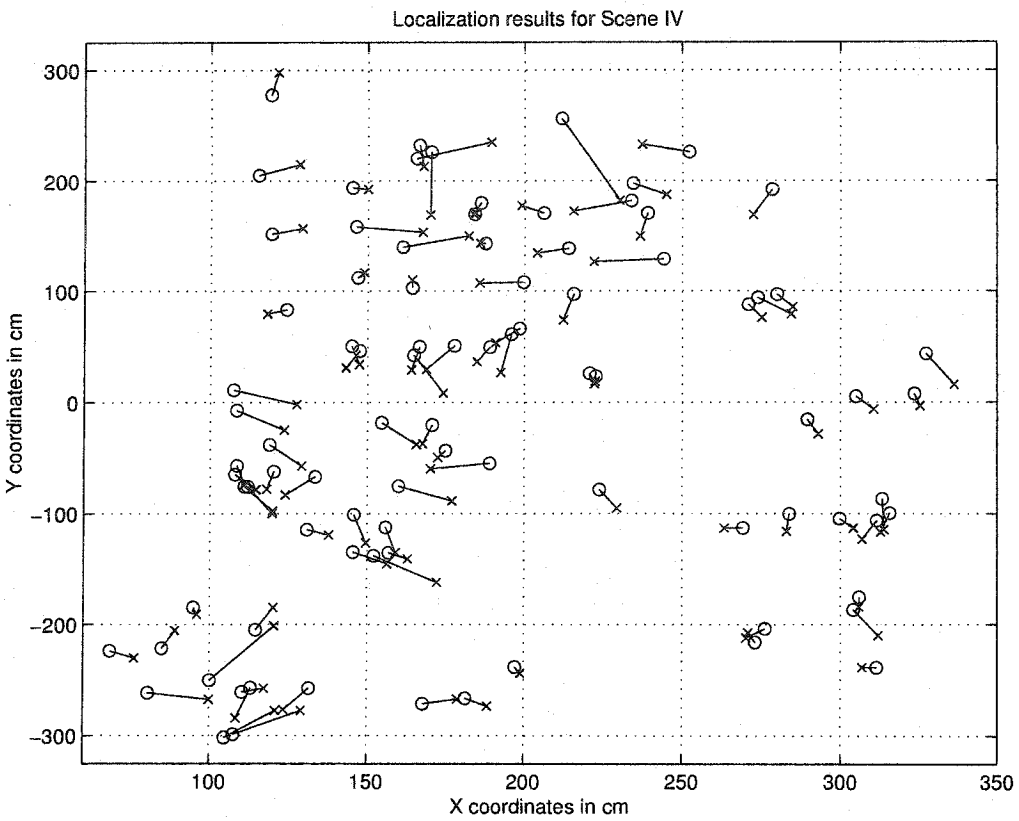


FIGURE 3.21. Localization results for Scene IV: the set of maximum-likelihood pose estimates ('x') plotted against their ground truth estimates ('o'). Mean error: 17cm

It is interesting to note that for Scenes Ib,II,III and IV the mean localization error scales roughly linearly with the sample spacing in the training set (which in turn scaled roughly linearly with the size and distance of the objects in the scene from the camera). As the sample spacing grows, however, feature tracking becomes more difficult and outliers become more significant. Conversely, at smaller scales, a lack of significant variation in feature behaviour can also produce outliers due to a flattening of the inferred probability distribution over pose, as was observed in Scene I. While this was remedied for Scene Ib by eliminating the feature attribute that led to the errors, the question of how to automatically detect which feature attributes will be useful remains open[3]. An interesting direction for future work will be to adaptively collect training examples, varying the sample spacing appropriately as the feature behaviour varies.

6. Comparison with Principal Components Analysis

The final experiment examines the performance of the visual mapping framework versus the approach developed by Nayar, *et al.* based on principal components analysis (PCA) [84]. PCA-based localization computes the principal directions of the subspace spanned by the training images which are interpreted as vectors in a high-dimensional space, and represents images as linear combinations of these principal directions (Section 2.2.2). Note that PCA is a *global* approach in that the entire image is used as input to the localization framework.

A localization framework using PCA can be defined by first computing a PCA subspace for the set of training images, and then constructing an interpolation function $\mathbf{z}_k = \hat{F}(\mathbf{q})$ that approximates the subspace projection $\mathbf{z}_k$ of an image as a function of pose $\mathbf{q}$. For the purposes of this experiment, the interpolation function is determined by computing a Delaunay triangulation of the training poses [25], and for an arbitrary pose $\mathbf{q}$, finding the face of the triangulation that $\mathbf{q}$ falls into and computing

[3]The attribute evaluation performed in Section 2 of this chapter is an indicator of model accuracy and stability, which only partly addresses this question.

$\mathbf{z}_k$ using bilinear interpolation between the projections of the corresponding training images[4].

The observation likelihood function is then defined as

$$p(\mathbf{z}|\mathbf{q}) = k \exp\left(-0.5||\mathbf{z} - \hat{F}(\mathbf{q})||^2\right) \tag{3.9}$$

where k is a normalizing constant and $\mathbf{z}$ is the subspace projection of the observed image. With this likelihood function, the probability distribution $p(\mathbf{q}|\mathbf{z})$ can be computed using Equation 2.7.

In order to provide the localization methods with training images and ground truth pose estimates, the Scene III training set was used. Recall that the orientation of the robot was fixed. The same set of training images was provided to both the PCA and visual mapping framework for preprocessing and training, and the running times for this phase were recorded. In order to optimize the visual map for speed, a triangulation-based feature model was employed (Section 4.4), and feature appearance was ignored in computing the model covariances and evaluating Equation 2.25 (that is, modelling and localization took into account only the image position of the modelled features).

For verification, initially the set of 29 Scene III verification images was employed. Note that these verification images were collected under the same illumination conditions and observed the same static scene as the training images. These images constitute our *Normal* verification set. In addition, a set of occluders, consisting of black tiles, were randomly painted into the images to generate an *Occlusion* verification set (Figure 3.22). The mean area of occlusion in each image was 32%. This set was generated in order to evaluate local versus global localization methods in the face of outliers in the image.

6.1. Experimental Results. Table 3.4 depicts the localization results for the 29 images in the *Normal* verification set. The mean localization error, maximum error, and offline and on-line running times are depicted for each method. The maximum

[4]The details of this method will be presented in greater detail in the next chapter.

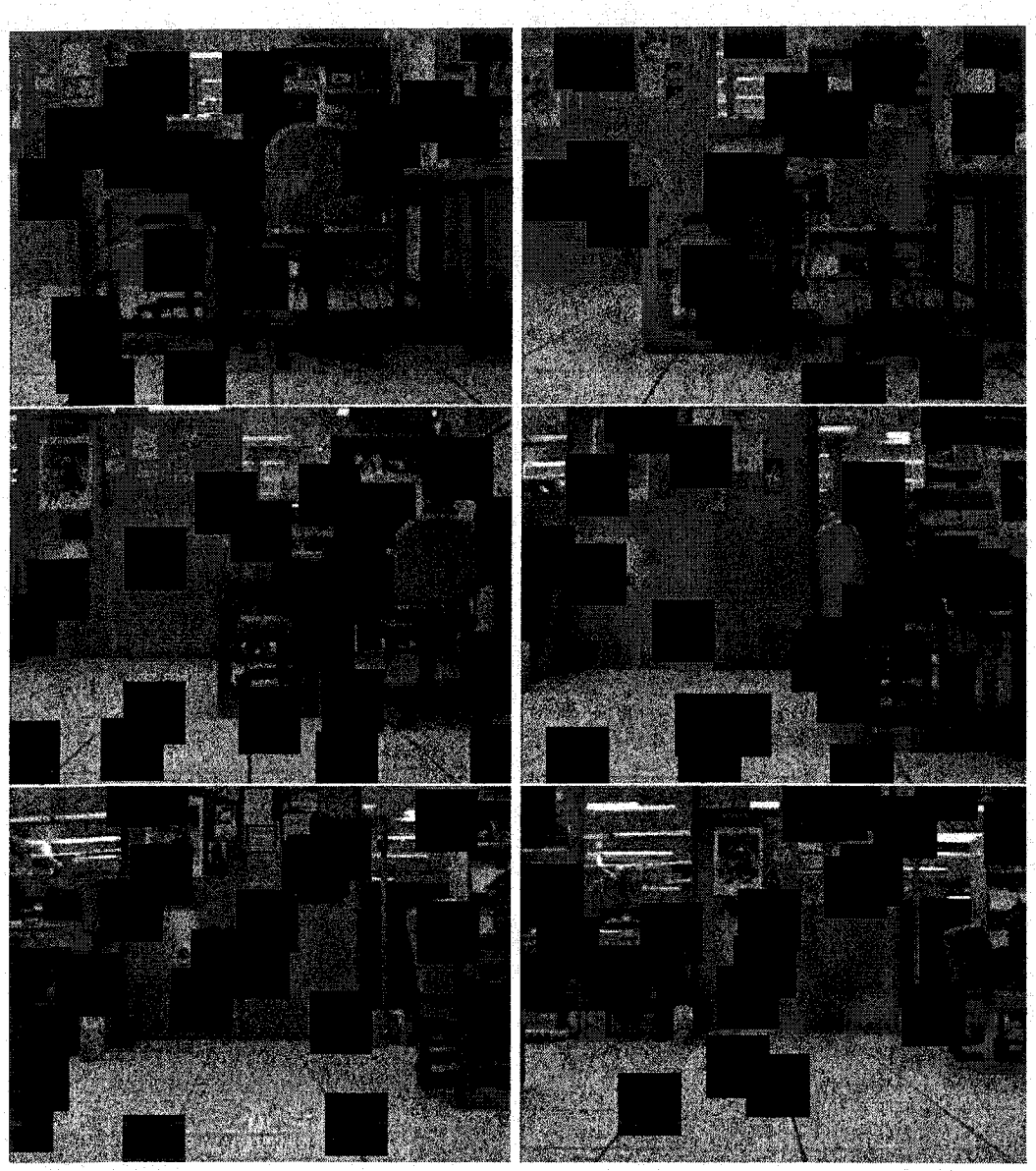


FIGURE 3.22. A selection of images from the *Occlusion* verification set.

errors are reported in lieu of a standard deviation since the errors are bounded from below by 0, and hence the mean and maximum errors provide more informative statistics on the tails of the distributions. Note that PCA presents a large offline computational cost, but at the gain of very fast on-line cost– whereas the visual mapping framework presents some efficiency in training, while it is more expensive to

TABLE 3.4. Results for the *Normal* set.

	Mean Error (cm)	Max Error (cm)	Total Training Time (s)	On-line Preprocessing (s/image)	On-line Localization (s/image)
PCA	6.06	12.4	2170	0.234	0.208
Visual Map	8.49	23.09	1581	145	19.8

TABLE 3.5. Results for the *Occlusion* set.

	Mean Error (cm)	Max Error (cm)
PCA	96.3	222
Visual Map	12.1	48.6

run on-line, primarily due to the need for feature matching and the cost of operating the generative models over the discretized pose space.

Table 3.5 depicts the results for the *Occlusion* verification set. The visual mapping method demonstrates a small degradation in performance, whereas PCA fails completely. These results confirm the utility of feature-based methods at providing robustness to dynamics in the environment, such as the movement of people and furniture.

The mean localization errors for the two experiments are summarized in Figure 3.23. While PCA is quite stable under normal conditions, the global method degrades considerably in the face of occlusions. Conversely, the visual mapping framework is performs slightly worse under normal conditions but provides robustness in the face of occlusion. It should be noted, however, that visual maps do incur significant overhead in terms of the cost of feature matching. These experiments were run without any prior information about feature matches, and the cost of matching can be reduced considerably with useful priors, which are often available in the context of active localization. A more complete examination of the performance of PCA, visual maps, and other methods under a variety of adverse imaging conditions can be found in [**125**]. The full details of that work have been omitted here as they compare

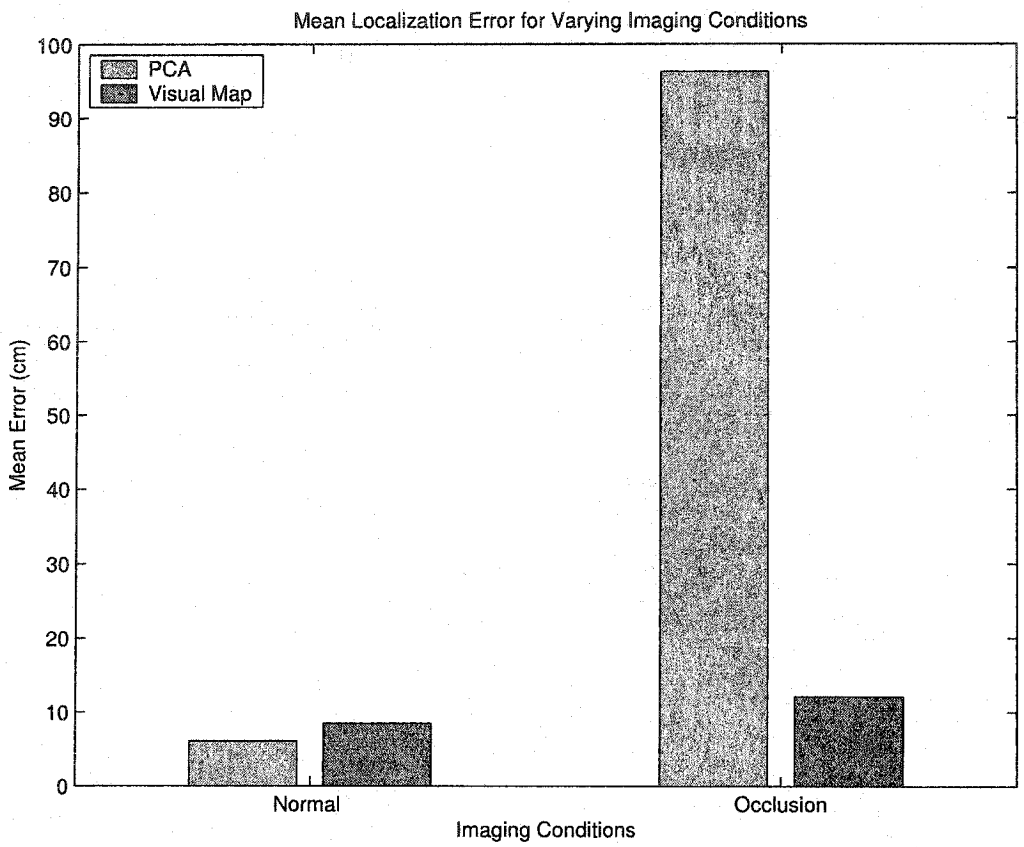


FIGURE 3.23. Summary of localization performance under differing imaging conditions.

the landmark-based localization approach against an ad-hoc global method whose elaboration would be tangential to the goals of this thesis.

7. Discussion

This chapter presented a set of experiments demonstrating the feature learning framework. The experimental results demonstrate the remarkable precision of the feature models when applied to the task of pose estimation, as well as other tasks, such as scene reconstruction. The results illustrate the stability and smoothness of the resulting posterior distribution over camera pose, and we were able to detect most outliers by thresholding the likelihood of the ML estimates. However, important issues are raised with respect to the density of training samples. In order to capture

aspects of the scene that change significantly, one must sample at higher densities. Conversely, in some instances features change too slowly to be informative for pose estimation. Furthermore, a method is required in order to provide ground truth pose information for the training images. The next chapter will examine the problem of inferring the feature models and the ensemble of training poses when only a small number of training poses are known.

CHAPTER 4

Self-Organizing Visual Maps

1. Overview

The previous chapter presented a framework for learning a set of feature models from an ensemble of images acquired from known robot poses. This chapter will examine the problem of automatically inferring a set of feature models while simultaneously determining the spatial distribution (in pose space) of the images in the training ensemble, even with only limited *a priori* information about the training poses. Beyond robotic applications, examples of such ensembles include camcorder and archival footage where some images can be easily localized but most can not (or would be too time consuming to determine). While the solution to this problem can be viewed as an instance of robot mapping it can also be used in other contexts where low-dimensional features can be extracted and correlated from high-dimensional measurements.

The approach taken to this problem is to use the visual mapping framework to initially select and match visual features from the ensemble without updating the appearance models, and to then localize the images by first assembling the small subset of images for which pose information is available (or, for a moving robot, where odometric confidence is high), and sequentially inserting the remaining images, localizing each against the previous estimates, and taking advantage of any priors that

are available. Experimental results validating the approach will be presented, demonstrating metrically and topologically accurate results over two large image ensembles, even given only four initial ground truth poses.

While the self-organizing approach described here depends to a large extent on adequate coverage of the pose space, the last part of this chapter will develop a method for organizing the input ensemble even when coverage is sparse. Furthermore, a mechanism will be provided for incorporating odometric information for the case that it is available.

2. Problem Statement

We are interested in building a visual map of an unknown environment from an ensemble of observations and limited pose information. Formally,

Given:

- I, an ensemble of images of an environment,
- Q, ground truth pose information indicating the pose of the camera from which a *subset* of images in I were acquired, and
- A, optionally, additional prior information providing relative ordering information of the ensemble, or robot control trajectory information.

Compute:

- A visual map of the environment, and
- Q', the set of camera poses of images in I for which ground truth is unavailable.

In the abstract, the goal is to examine the extent to which we can organize a set of measurements from an unknown environment to produce an embedding of the measurements in pose space with little or no knowledge of where in the environment the measurements were obtained. A primary assumption is that, at most, we have limited prior trajectory information, so as to bootstrap the process– the source of this information might be from the first few odometry readings along a trajectory, the general shape of the trajectory, information from an observer, or from a localization

method that is expensive to operate, and hence is only applied at a small subset of the observation poses. While metric accuracy is of interest, the primary aim is to recover an embedding of the ensemble in pose space. That is, metrically adjacent poses in the world are topologically adjacent in the resulting map.

The question of how to bootstrap a spatial representation, particularly a vision-based one, also appears to be relevant to other research areas in computer vision and even ethology. Several authors have considered the use of self-organization in autonomous agent navigation [**138, 5, 23, 111**], often with impressive results. For example, Takahashi *et al.* construct a self-organizing map from odometric data and use reinforcement learning to train the robot to navigate between distant nodes in the inferred graph. This thesis is among the first work to demonstrate how to automatically build a complete map of a real (non-simulated), large-scale (that is, human scale), unknown environment using only monocular vision.

We will approach the problem in the context of the visual map representation. To construct a map, two steps are involved: first, reliable features are selected and correspondences are found across the image ensemble. Second, the quantitative behaviours of the features as functions of pose are exploited in order to compute a maximum-likelihood pose for each image in the ensemble. While other batch-oriented mapping approaches are iterative in nature [**143, 60**], it will be demonstrated that if accurate pose information is provided for a small subset of images, the remaining images in the ensemble can be localized without the need for further iteration and, in some cases, without regard for the order in which the images are localized.

The following section will examine prior work related to the problem; in particular, approaches to self-organizing maps, and the simultaneous localization and mapping problem. We will then proceed to present how the visual mapping framework is applied to organize the input ensemble. Finally, we will examine experimental results on a variety of ensembles, demonstrating the accuracy and robustness of the approach.

3. Previous Work

The construction of self-organizing spatial maps (SOM's) has a substantial history in computer science. In particular, Kohonen developed a number of algorithms for covering an input space with a network of weighted computational units that are distributed in a mesh [**60, 61**]. The units are initialized to have random weights and as the input training vectors are introduced to the network, maximally responding units have their weights adjusted toward the input, along with a local neighbourhood of units defined according to a neighbourhood function which decreases in size over time. If we consider the feature observations $\mathbf{z}_i$ from Chapter 2, and sort them into a set of vectors $\mathbf{z}(t)$, indexed by time t and the unit weights at each unit are defined by a vector $\mathbf{m}_i(t)$, where i is the index of the unit, and $\mathbf{m}_i(t)$ has the same dimensionality as the observations, then the update equation for the neighbourhood of the maximally responding unit at time t is defined to be

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(\mathbf{z}(t) - \mathbf{m}_i(t)) \tag{4.1}$$

where α is a learning rate parameter.

After a certain number of iterations, the unit that responds maximally to input vector i is labelled with that vector, and once all vectors have been assigned to a unit the distribution of labels defines an ordering of the input vectors.

While spatial coverage is used as a metaphor, the problem of representing a data space in terms of self-organizing features has numerous applications ranging from text searching to audition [**146**]. The problem of spanning an input space with feature detectors or local basis functions has found wide application in machine learning, neural nets, and allied areas. In much of this algorithmic work, particularly as it pertains to Kohonen maps, the key contributions have been related to convergence and complexity issues. In fact, one significant difficulty with SOM's is the lack of evidence that in high-dimensional cases the algorithm has a steady, or absorbing, state [**15**].

The issue of automated mapping has also been addressed in the robotics community. One approach to fully automated robot mapping is to interleave the map synthesis and position estimation phases of robot navigation, a process known as simultaneous localization and mapping (SLAM), or sometimes concurrent mapping and localization (CML). As it is generally applied, this entails incrementally building a map based on geometric measurements (e.g. range measurements derived from a laser range-finder, sonar or stereo camera) and intermittently using the map to correct the robot's position as it moves [**70, 157, 19**]. Results in these areas will be considered in greater depth in the subsequent chapters.

When the motion of a robot can only be roughly estimated, a topological representation becomes very attractive. Kuipers and Byun use repeated observation of a previously observed landmark to instantiate cycles in a topological map of an environment during the mapping process [**67, 65, 66**]. The robot navigates between nodes using a set of control mechanisms and the locally defined "landmarks" are based on the observed local maxima of some distinctiveness measure of the environment. Simhon and Dudek apply this idea of distinctiveness to the problem of constructing map of visually distinctive places in the environment, to each of which is attached a local metric map [**131**].

The idea of performing SLAM in a topological context was also been examined theoretically. Dudek *et al.* demonstrated that for a graph-like world where nodes have no distinctive characteristics other than a consistent edge ordering, correct graph inference is impossible without the aid of a marker that can be dropped and later retrieved when it is re-encountered, permitting the robot to recognize when it has closed a cycle [**31**]. Deng and Mirzaian examined the same problem and developed a set of competitive algorithms utilizing a set of n identical markers [**24**].

When trajectory information is available, the probabilistic fusion of uncertain motion estimates and observations has been examined by several authors (e.g., [**132,**

42, 74]). These results deal exclusively with observations derived from range sensors. The use of expectation maximization (EM) has recently proven quite successful although it still depends on estimates of successive robot motions [**22, 112, 142, 12**].

EM is a post-processing method that operates iteratively, alternating an E-step, in which the current map is assumed to be correct and the set of observations is localized against the map, followed by an M-step in which the map is reconstructed based on the assumption that the localized poses are correct. The process repeats until the map reaches a steady state. Unfortunately, convergence is so far guaranteed only for a specific family of exponential probability distributions, and the process is susceptible to finding local maxima in the probability distribution. While outside the scope of this thesis, the results from this chapter can in turn be employed as a reliable prior for subsequent EM-style post-processing.

A closely related problem in computer vision is that of close-range photogrammetry, or structure-from-motion (SFM) [**72, 91, 44, 18**], which involves recovering the ensemble of camera positions, as well as the full three-dimensional geometry of the scene that is imaged. SFM operates by computing a least-squares minimization of the projection error of a set of features for which correspondence has been established between images. In the case where a para-perspective camera model is assumed, the problem is linear and the solution can be computed directly. For more realistic camera models, the problem is non-linear and is computed using an iterative process known as bundle-adjustment [**149**].

The key difference between the SFM problem and inferring pose with a visual map is that a solution to the SFM problem is dependent on explicit assumptions about the optical geometry of the imaging apparatus. In the visual mapping framework we have avoided committing to any such assumptions, and as such the self-organizing behaviour exhibited in the experimental results is equally applicable to exotic imaging hardware, such as an omnidirectional camera. A second major difference between this work and the SFM problem is that SFM depends on image features that correspond to well-defined points in three-dimensional space, whereas the visual mapping framework

can handle features derived from more exotic visual phenomena, such as intersections of occlusion boundaries in depth.

4. An Alternative Feature Model

Effective pose inference requires accurate models for performing localization. In order to build a map on-line, as new observations are added the models must be updated. In the context of visual mapping, model update, and particularly cross-validation is an expensive operation. Computing a feature model from n observations costs $O(n^3)$ time, due to the need for singular values decomposition required to solve Equation 2.21. Similarly, cross-validation costs $nO((n-1)^3) = O(n^4)$, as the model must be fully recomputed for each observation. If the model is updated and cross-validated with the addition of each new observation, the total complexity is given by

$$t = \sum_{i=1}^{n} (c_1 i^3 + c_2 i^4) = O(n^5). \tag{4.2}$$

Equation 4.2 describes the running time for the complete construction of an RBF model with full cross-validation updates after each observation insertion. While there are methods for adding and removing rows and columns from a singular value decomposition (as is the case in cross-validation and model updating), and other techniques exist for taking advantage of the symmetric structure of the matrix, these techniques can save at most one degree of computational complexity [**39**].

In order to reduce the cost of model updating, this chapter and the following chapters will employ a feature model which represents a more efficient approximation to the generating function $F_m(\cdot)$. The benefits of this model will be that observation insertion and removal will be $O(\log n)$, facilitating $n \log(n)$ model construction and cross-validation. The drawback to employing this model is that it will be somewhat susceptible to outlier observations. Of course, once the complete map has been constructed, it is possible to apply the more stable RBF-based model.

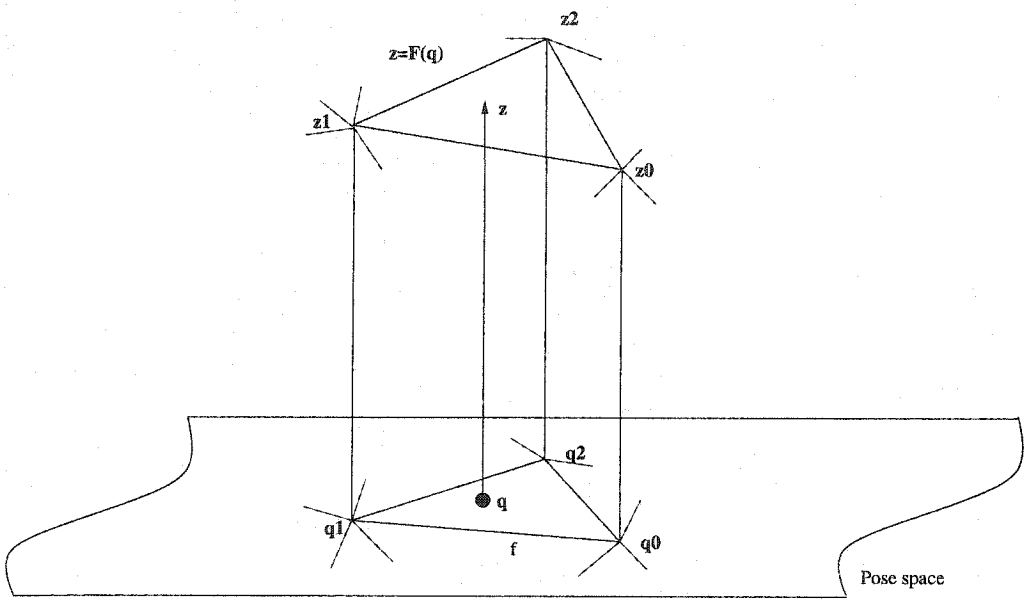


FIGURE 4.1. The approximate model based on a Delaunay triangulation of the pose space. $\hat{F}(\mathbf{q})$ is evaluated by computing the face f that contains $\mathbf{q}$, and interpolating between the observations associated with the vertices of f.

The basic premise of the model approximation is depicted in Figure 4.1. An interpolation model is constructed by computing a Delaunay triangulation of the set of observation poses and performing interpolation at the pose $\mathbf{q}$ by locating the face f of the triangulation that contains $\mathbf{q}$ and employing bilinear interpolation of the observations associated with the vertices of that face in order to generate a predicted $\mathbf{z}^*$ [**25**]. The triangulation mechanism is implemented using a hierarchical data structure that facilitates insertions and lookups into the triangulation in $O(\log n)$ expected time and deletions in $O(\log \log n)$ expected time [**13**].

Interpolation operates as follows:

ALGORITHM 4.1.

(i) *Given input* $\mathbf{q}$ *locate the face* f *that* $\mathbf{q}$ *falls into. If the face does not contain the infinite vertex, goto 4, otherwise*

(ii) *For neighbouring faces to f that do not contain the infinite vertex find the face f' that minimizes $||\hat{\mathbf{q}}||$, where $\hat{\mathbf{q}}$ corresponds to the coordinates of $\mathbf{q}$ relative to the basis defined by the vertices of f'*[1]

(iii) *Set $f = f'$.*

(iv) *Compute the coordinates $\hat{\mathbf{q}} = [x\ y]^T$ of $\mathbf{q}$ relative to the coordinate basis defined by the face f (see below).*

(v) *Compute $\mathbf{z}^* = x(\mathbf{z}_2 - \mathbf{z}_1) + y(\mathbf{z}_3 - \mathbf{z}_1)$ as the linear combination of observations $\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$ corresponding to the coefficients $\hat{\mathbf{q}}$.*

For poses $\mathbf{q}_1$, $\mathbf{q}_2$,and $\mathbf{q}_3$, defining the vertices of the selected basis face f, the coordinates $\hat{\mathbf{q}}$ are determined from $\mathbf{q}$ as

$$\hat{\mathbf{q}} = A^{-1}[\mathbf{q} - \mathbf{q}_1] = [\mathbf{q}_2 - \mathbf{q}_1\ \mathbf{q}_3 - \mathbf{q}_1]^{-1}[\mathbf{q} - \mathbf{q}_1] \tag{4.3}$$

where $\mathbf{q}_2 - \mathbf{q}_1$ and $\mathbf{q}_3 - \mathbf{q}_1$ define the columns of A. This basis defines the position of $\mathbf{q}$ as a linear combination of the vectors $\mathbf{q}_2 - \mathbf{q}_1$ and $\mathbf{q}_3 - \mathbf{q}_1$.

While other, more stable, interpolation algorithms may exist, in practice the triangulation model provides a good local approximation to the generating function $F(\cdot)$, provided that there are no outlier observations in the training set. Even in the presence of a small number of outliers, their effects will be limited to local regions of pose space. Finally, it should be noted that the triangulation model provides a reasonably stable extrapolation mechanism, a feature that RBF models are unreliable at producing. Stable extrapolation is a key requirement for incrementally growing the space over which a feature can be reliably modelled.

5. Self-Organization

We now turn to the problem of inferring the poses of the training images when ground truth is unavailable, or only partially available. The self-organization process involves two steps. In the first step, image features are selected and tracked, and in the second step the set of images is localized.

[1]The purpose of this step is to locate a face that provides maximal stability for interpolation.

5.1. Tracking. Tracking proceeds by considering the images in an arbitrary order (possibly, but not necessarily, according to distance along the robot's trajectory). Given the input image set $Z = \mathbf{z}_i$, the tracking algorithm operates as follows:

ALGORITHM 4.2.

(i) *Apply an edge-density feature detector* Ψ *to the first image* $\mathbf{z}_1 \in Z$ *(Section 2.7).*

(ii) *Initialize a* tracking set T_j *for each detected feature.*

(iii) *Define* $T = T_j$.

(iv) *Initialize* $E = \mathbf{z}_1$.

(v) *For each subsequent image* $\mathbf{z}_i$, *do*

  (a) *Search* $\mathbf{z}_i$ *for matches to each* $T_j \in T$ *(Section 2.8).*

  (b) *Add successful matches to their respective tracking sets* T_j. *Call the set of successful matches* M.

  (c) *Apply* Ψ *is applied to the image and let* S *be the set of detected features.*

  (d) *If* $|M| < |S|$, *then*[2]

    (i) *New tracking sets* T_j *are initialized by elements selected from* S. *The elements are selected first on the basis of their response to the attention operator, and second on the basis of their distance from the nearest image position in* M[3]. *Call this new set of tracking sets* T_S.

    (ii) *For each* $\mathbf{z}_k \in E$ *search for matches to the new tracking sets in* T_S *(Section 2.8), and add the successful matches to their respective tracking set.*

    (iii) $T = T \cup T_S$

  (e) $E = E \cup \mathbf{z}_i$

[2] $|X|$ refers to the cardinality of the set X.

[3] In this way, features in S which are close to previous matches are omitted, and regions of the image where features exist but matching failed receive continued attention.

The template used for by any particular tracking set is defined as the local appearance image of the first feature added to the set. Matching is considered successful when the normalized correlation of the template with the local image under consideration exceeds a user-defined threshold (Section 2.8).

When tracking is completed, we have a set of feature correspondences across the ensemble of images. The process is $O(kn)$ where k is the final number of tracked sets, and n is the number of images.

5.2. Localization. Once tracking is complete, the next step is to determine the position of each image in the ensemble. For the moment, consider the problem when there is a single feature that was tracked reliably across all of the images. If we assume that the motion of the feature through the image is according to a monotonic mapping from pose to image, then the topology of a set of observation poses will be preserved in the mapping from pose-space to image-space.

While the mapping itself is nonlinear (due to perspective projection), it can be approximated by associating actual poses with a small set of the observations and determining the local mappings of the remaining unknown poses by constructing an interpolant over the known poses. The algorithm proceeds as follows:

ALGORITHM 4.3.

(i) *Initialize* $S = \{(\mathbf{q}, \mathbf{z})\}$, *the set of (pose, observation) pairs for which the pose is known.*

(ii) *Compute* $\hat{F}_i(\cdot)$, *the parameterization of* S *as defined by the feature learning framework.*

(iii) *For each observation* $\mathbf{z}_j$ *with unknown pose,*

  (a) *Use* $\hat{F}_i(\cdot)$ *as an interpolant to find the pose* $\mathbf{q}^*$ *that maximizes the probability that* $\mathbf{q}^*$ *produces observation* $\mathbf{z}_i$.

  (b) *Add* $(\mathbf{q}^*, \mathbf{z}_j)$ *to* S *and update* $\hat{F}_i(\cdot)$ *accordingly.*

For a parameterization model based on a Delaunay triangulation interpolant, updating the model $\hat{F}_i(\cdot)$ takes $O(\log n)$ expected time, where n is the number of observations in the model. The cost of updating the covariance associated with each model is $O(k \log n)$, where k is the number of samples omitted during cross-validation.

In addition, the cost of finding the most likely pose with $\hat{F}_i(\cdot)$ is $O(m \log n)$, where m corresponds to the number of poses that are evaluated (finding a face in the triangulation that contains a point $\mathbf{q}$ can be performed in $\log n$ time.). Given n total observations, the entire algorithm takes $O(n(m + k + 1) \log n)$ time. Both m and k can be bounded by constants, although k is typically chosen to be n.

In practice, of course, there is more than one feature detected in the image ensemble. Furthermore, in a suitably small environment, some might span the whole set of images, but in most environments, most are only visible in a subset of images. Finally, matching failures might introduce a significant number of outliers to individual tracking sets. Multiple features, and the presence of outlier observations are addressed by the localization framework; the maximum likelihood pose is computed by maximizing Equation 3.8, and the effects of outliers in a tracked set are reduced by their contribution to the covariance associated with that set.

When it cannot be assumed that the environment is small enough such that one or more feature spans it, we must rely on stronger *a priori* information to bootstrap the process. For example, we might require the initial known poses to be close together, ensuring that they share common features for parameterization. In addition, we might take advantage of knowledge of the order in which images were acquired along a trajectory, ensuring that as one feature goes out of view, new ones are present against which to localize.

The following section will present experimental results on two image ensembles.

6. Experimental Results

6.1. A Small Scene. The first experiment will demonstrate the procedure on Scene III (Figure 3.3, repeated in Figure 4.2), a relatively compact pose space.

FIGURE 4.2. Scene III, repeated.

Recall that the ensemble of 121 images was collected over a 2m by 2m environment, at 20cm intervals, and that ground truth was measured by hand, accurate to 0.5cm.

Given the ensemble, the images were sorted at random and tracking was performed as described in Section 4.5.1, resulting in 91 useful tracked features. (A tracked feature was considered useful if it contained at least 4 observations). The localization stage proceeded by first providing the ground truth information to four images selected at random. The remaining images were again sorted at random and added, without any prior information about their pose, according to the methodology described in the previous section. Figure 4.3 depicts the set of inferred poses versus their ground truth positions. The four 'holes' in the data set at $(20, 200)$, $(40, 140)$, $(60, 0)$, and $(140, 0)$ correspond to the four initial poses for which ground truth was supplied. For the purposes of visualization, Figure 4.4 plots the original grid of poses, and beside it the same grid imposed upon the set of pose estimates computed for the ensemble.

In order to quantify the distortion in the resulting map, the lengths of the mapped line segments corresponding to the original grid were measured, and the average and standard deviation in the segment lengths was recorded. For the ground-truth mesh,

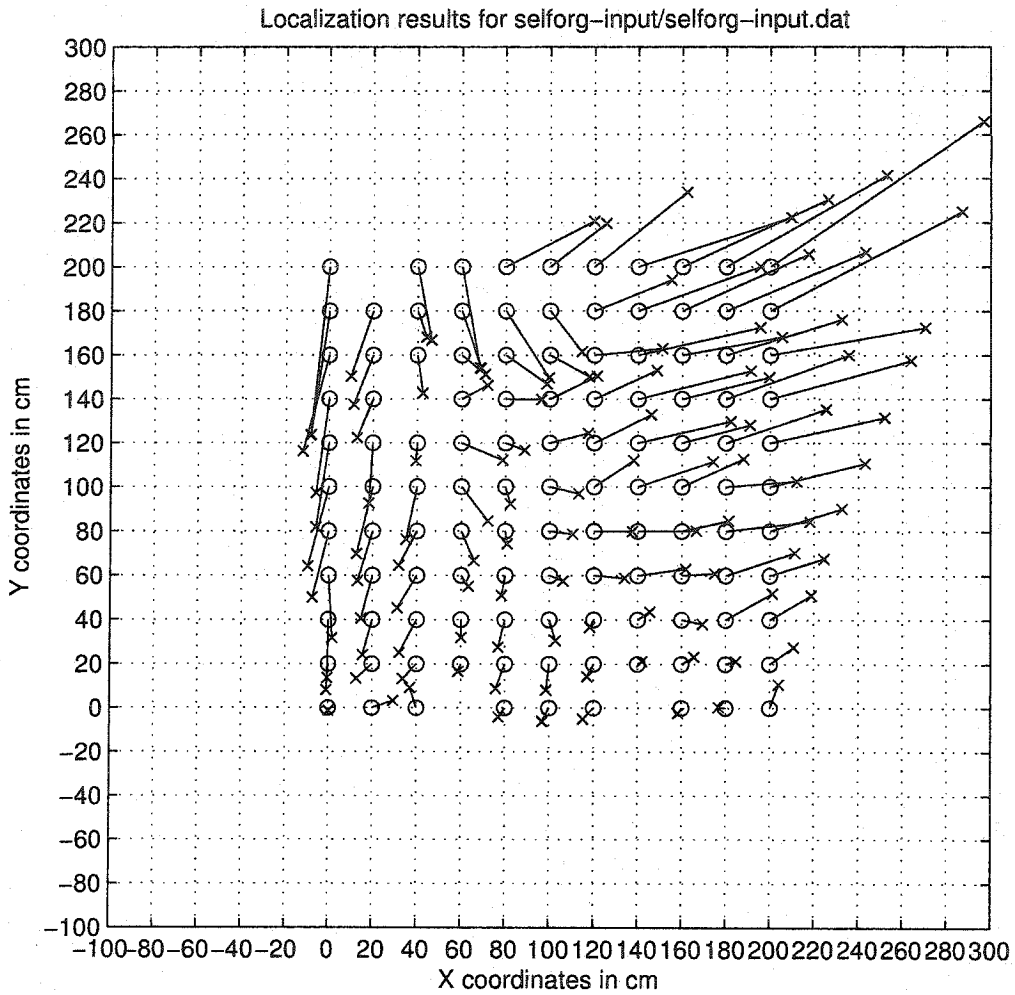


FIGURE 4.3. Self-organizing pose estimates plotted versus ground truth for Scene III.

the average and standard deviation segment length was 20cm and 0cm, respectively (assuming perfect ground truth). In the inferred map, the mean segment length was 24.2cm and the standard deviation was 11.5cm. These results demonstrate that the resulting map was slightly dilated and with variation in the segment lengths of about 11.5cm or 58% of 20cm on average.

While there is clearly some warping in the mesh, for the most part the topology of the poses is preserved. It is interesting to note that the mesh is distorted most as

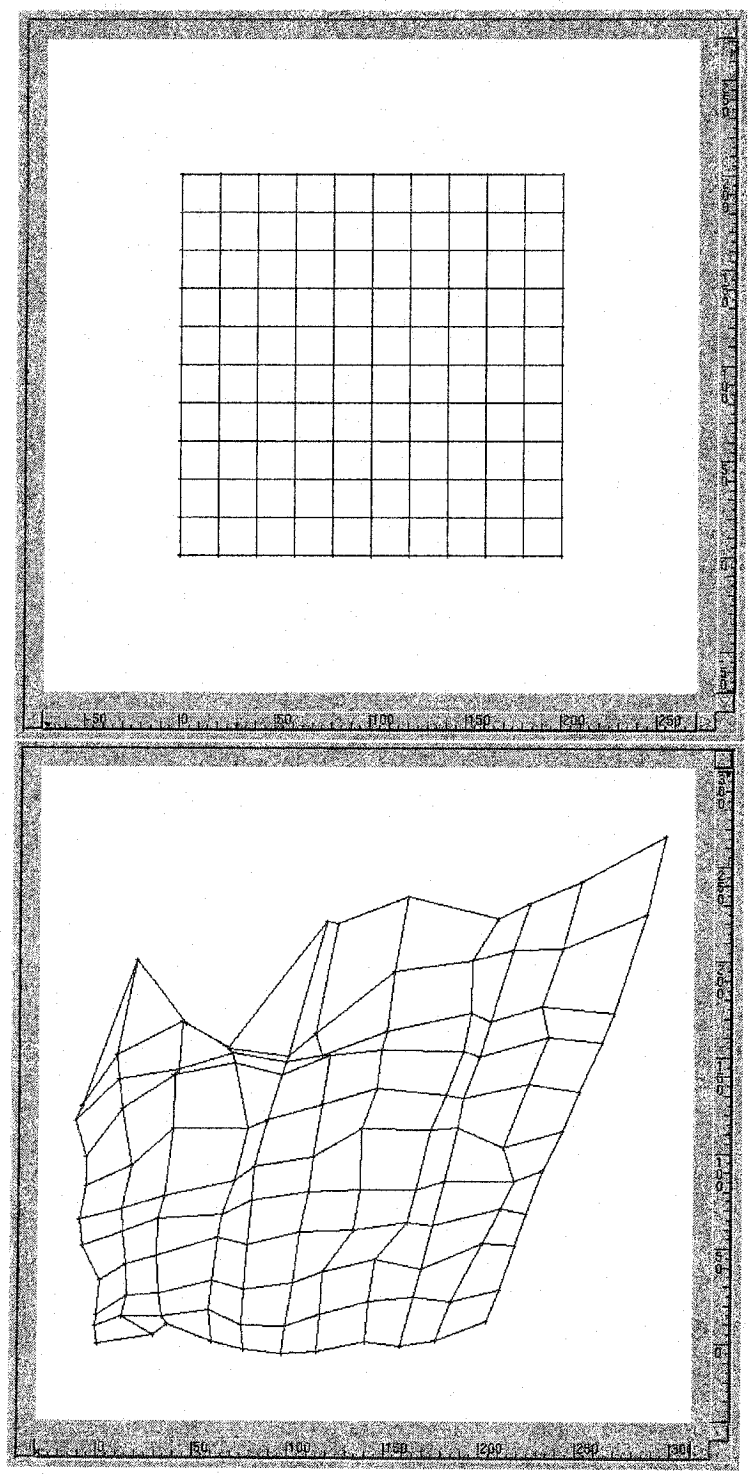


FIGURE 4.4. Ground truth, and the map resulting from the self-organizing process for Scene III.

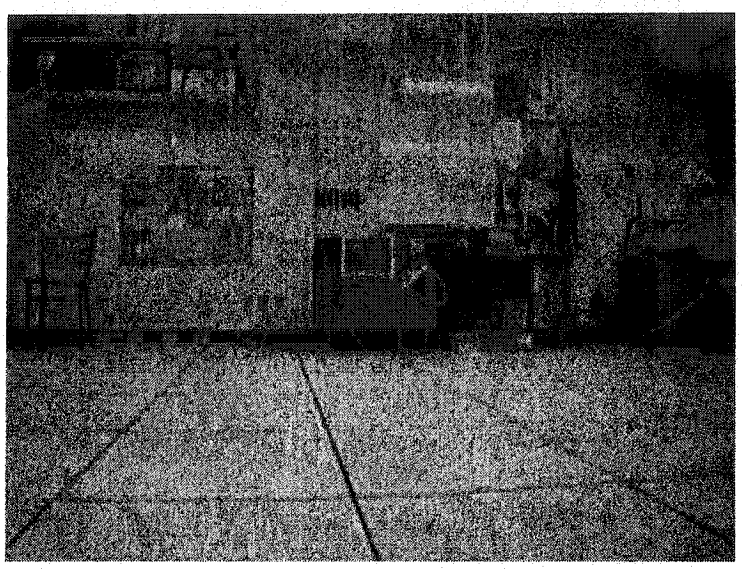


FIGURE 4.5. Scene IV

the y-axis increases, corresponding to looming forward with the camera and, as such, where the nonlinearity of the camera geometry is most pronounced.

6.2. A Larger Scene. For the second experiment, we consider a larger pose space, Scene IV, depicted in Figure 4.5. For this experiment, a section of the training images closest to the wall were removed from the training set after it was found that they had insufficient feature matches to make reasonable inferences. The resulting training set consisted of 252 images, spaced at 25cm intervals. Recall that ground truth for the set was measured using a pair of robots, one of which used a laser range-finder to observe the pose of the moving robot [**101**].

As in the previous experiment, tracking was performed over the image ensemble and a set of 49 useful tracked features were extracted. In this instance, the larger interval between images, some illumination variation in the scene and the larger number of input images presented significant challenges for the tracker, resulting in the smaller number of tracked features.

Given the size of the environment, no one feature spanned the entire pose space. As a result, it was necessary to impose constraints on the *a priori* ground-truth, and the order in which the input images were considered for localization. In addition, a

weak prior $p(\mathbf{q})$ was applied as each image was added in order to control the distortion in the mesh.

Rather than select the initial ground-truth images at random, ground truth was supplied to the four images closest to the centre of the environment. The remainder of the images were sorted by simulating a spiral trajectory of the robot through the environment, intersecting each image pose, and adding the images as they were encountered along the trajectory. Figure 4.6 illustrates the simulated ground-truth trajectory through the ensemble. Finally, given the sort order, as images were added it was assumed that their pose fell on an annular ring surrounding the previously estimated poses. The radius and width of the ring was defined to be at least 25cm away from the previously estimated poses, where 25cm corresponds to the interval used to collect the images. The computed *a priori* distributions over the first few images input into the map are depicted in Figure 4.7. The intent of using these priors was to simulate a robot exploring the environment along trajectories of increasing radius from a home position.

As in the previous section, Figure 4.8 plots the original grid of poses, and beside it the same grid imposed upon the set of pose estimates computed for the ensemble. Again, the positive y-axis corresponds to looming forward in the image, and as such the mesh distorts as features accelerate in image space as the camera approaches them. Note however, that as in the first experiment, the topology of the embedded poses is preserved for most of the grid. Quantitatively, the mean segment length in the original grid is about 25cm, whereas the mean length in the inferred map is 37.9cm with a 20.7cm standard deviation. In this case, the dilation is more pronounced and there is a wider variation in the segment lengths, due primarily to the looming effects.

7. Sparse Pose Spaces and Loop Closing

For many image ensembles, the trajectory of the camera moves along a 1-D curve in 3-space. In these cases, while image features behave smoothly as a function of

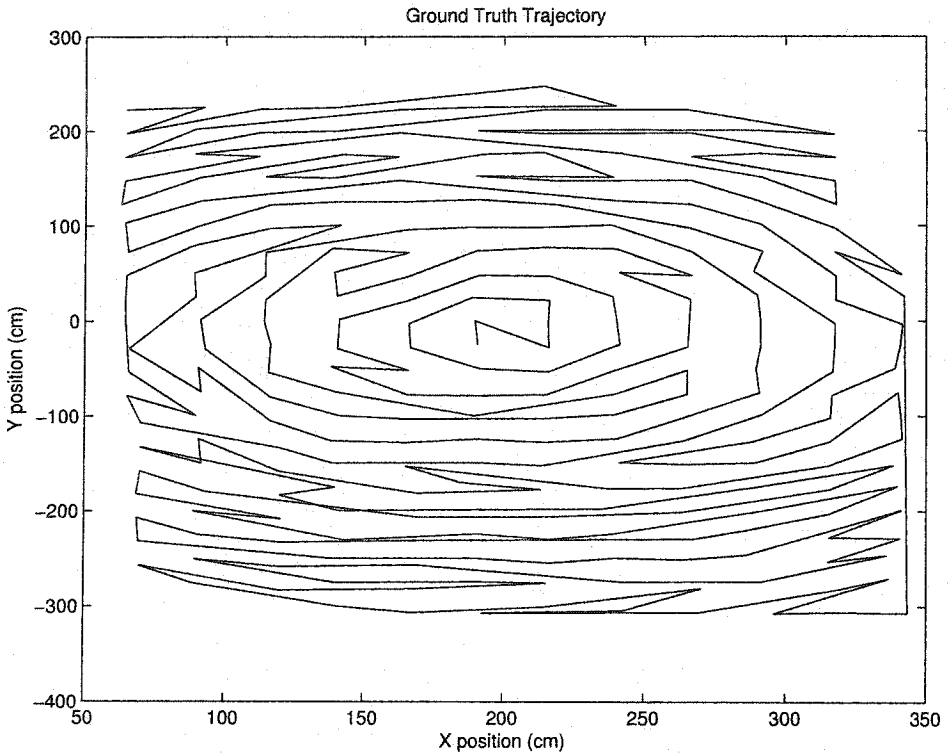


FIGURE 4.6. Ground truth simulated trajectory.

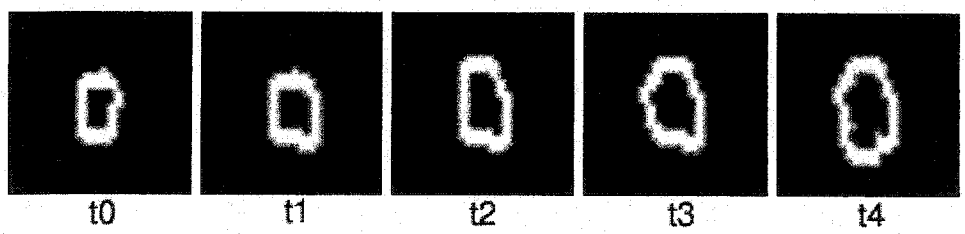


FIGURE 4.7. Evolution of the annular prior $p(\mathbf{q})$ over the first few input images. Each thumbnail illustrates $p(\mathbf{q})$ over the 2D pose space at time t_i.

pose, it is generally difficult to model their general behaviour without providing additional constraints, or making restrictive assumptions. This section will consider the problem of inferring the spatial distribution of an image ensemble derived from a 1-D trajectory through space where weak odometric information is available. The method will operate by inferring proximity between images in pose space based on a measure of their similarity that is computed from the co-visibility of features, and

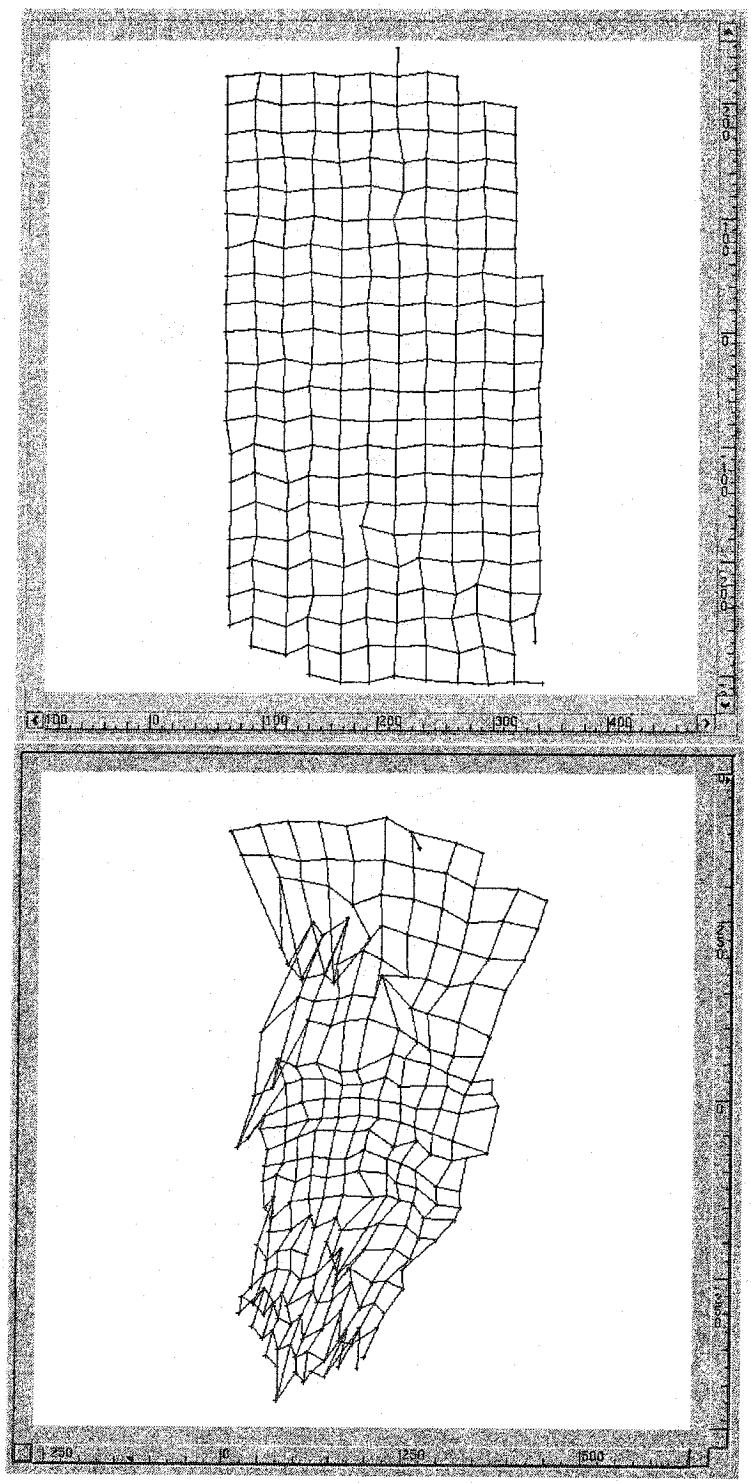


FIGURE 4.8. Ground truth, and the map resulting from the self-organizing process for the environment depicted in Figure 4.2.

conditioned on the odometric input. The inference process will be iterative, stochastically generating possible outcomes from the odometric input and evaluating them using a cost function based on the similarity measure. Experimental results will then be presented, illustrating the convergence and correctness of the method.

7.1. Image Similarity. A simple measure for determining whether two images derive from the same region of pose space is to compute the correlation of their visible features. If the set of features observed in image I_i is Z_i and the set of features observed in image I_j is Z_j, then the correlation ρ_{ij} between the images can be computed as

$$\rho_{ij} = \frac{|Z_i \cap Z_j|^2}{|Z_i||Z_j|} \tag{4.4}$$

where $|X|$ is the set cardinality operator.

Note that $\rho_{ij} \in [0, 1]$, takes on a value of 0 when the two images have no features in common, and 1 when the two images share all the same features. Given that there is a non-zero probability that some features will be outlier matches, and that some image pairs might have very few total features, ρ_{ij} is thresholded on the cardinality of the intersection of the feature sets (the numerator of Equation 4.4), and set to zero when the threshold is not reached. Another way of thinking about the thresholding mechanism is to say that when there is insufficient evidence to support the conclusion that two images are similar, we will assume that they are not.

For a concrete example, Figure 4.9b) plots the correlation matrix $\Xi = [\rho_{ij}]$ for a sequence of images extracted from the Scene IV training set. The trajectory of poses corresponding to the image sequence is plotted in Figure 4.9a), starting in the lower left corner and ending in the centre of the loop. Each element ρ_{ij} of the matrix corresponds to the similarity between images I_i and I_j along the trajectory. The bright diagonal corresponds to the fact that an image's similarity with itself is always 1.

7.2. Conditioning on a Motion Model. The correlation matrix Ξ provides a measure of the likelihood that any two images derive from nearby poses. In some

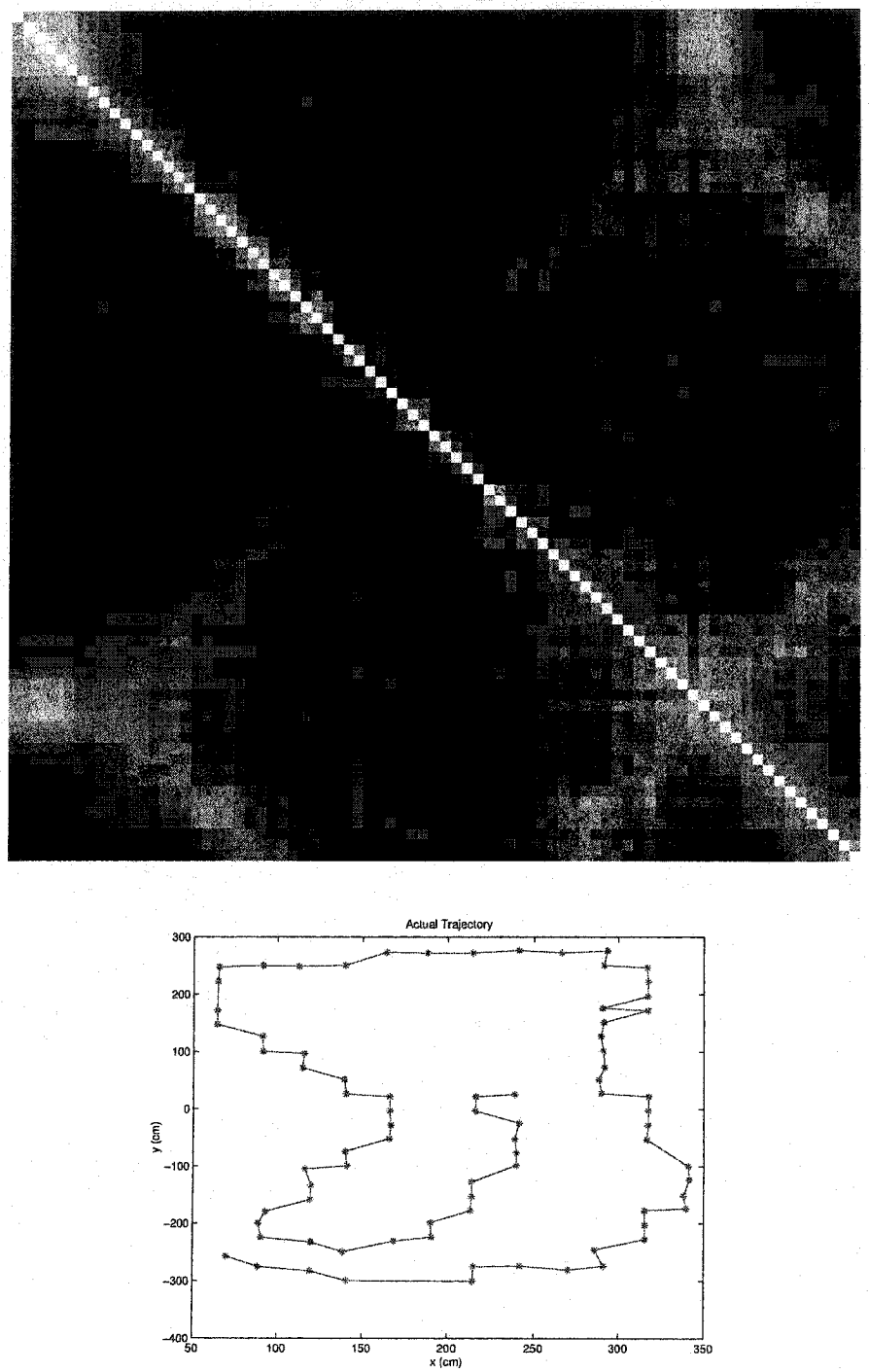


FIGURE 4.9. Similarity matrix for image sequence whose pose-space trajectory is plotted below. Brightness corresponds to higher similarity.

circumstances, Ξ may be insufficient to adequately differentiate between different parts of the world, due to self-similarities and outlier feature matches. One method for improving the correlation measure is to incorporate odometric information, when it is available. For example, suppose we are provided with A, the sequence of (noisy) actions performed by the robot. Define the sequence A_{ij} as the sequence of actions starting immediately after image I_i is acquired and ending immediately before image I_j is acquired.

Given a sequence A_{ij} and a starting pose $\mathbf{q}_i$, we can compute the probability distribution of the pose of the robot at the end of the sequence $p(\mathbf{q}|A_{ij})$, and therefore compute the probability that the pose $\mathbf{q}_j$ is equal to $\mathbf{q}_i$, $p(\mathbf{q}_j = \mathbf{q}_i|A_{ij})$. For a robot with a translation/rotation motion model, the distribution is likely to be non-Gaussian. However, the Extended Kalman Filter (EKF), which linearizes the motion model, can be applied to compute a Gaussian approximation for $p(\mathbf{q}_j = \mathbf{q}_i|A_{ij})$. The EKF is presented in greater detail in the next chapter.

Using the probability distribution $p(\mathbf{q}_j = \mathbf{q}_i|A_{ij})$, we can compute a new similarity measure, conditioned on the *a priori* likelihood that poses i and j are proximal:

$$\theta_{ij} = \rho_{ij} p(\mathbf{q}_j = \mathbf{q}_i|A_{ij}) \tag{4.5}$$

and define the matrix $\Theta = [\theta_{ij}]$ to be the similarity matrix conditioned on the action sequence. The cost of computing Θ is $O(n^2)$ where n is the number of actions, due to the need to compute each EKF for A_{ij}.

For the image sequence whose ground-truth trajectory is presented in Figure 4.9, a noisy control sequence was simulated by running a Monte Carlo simulation that tracked the actual pose of the robot and sampled the input actions based on the robot's stochastic odometry model. The simulated action sequence A is illustrated in Figure 4.10.

Given the simulated action sequence A, the similarity matrix depicted in Figure 4.9 was conditioned on the EKF estimate computed from A. The resulting matrix

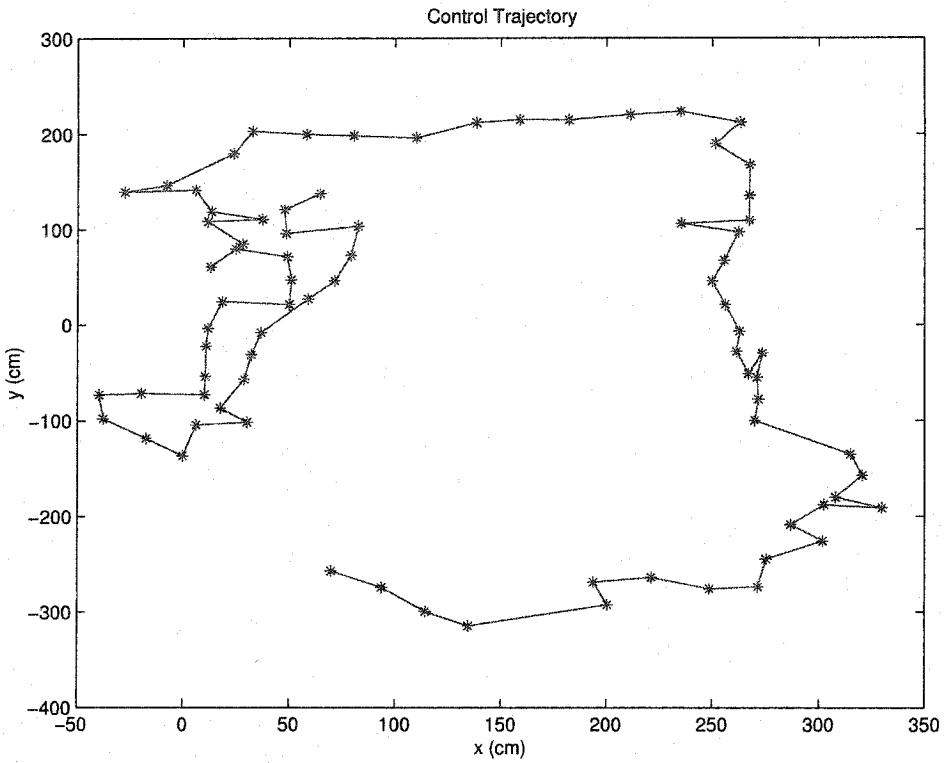


FIGURE 4.10. The simulated action sequence, computed using Monte Carlo simulation from the ground-truth trajectory depicted in Figure 4.9.

is normalized and depicted in Figure 4.11. Note that now a single image pair stands out as being highly proximal, with neighbouring images also indicating proximity.

7.3. Trajectory Inference. Given Θ and the action sequence A, trajectory inference takes place by optimizing a cost function C defined over pose sequences and the similarity matrix. Let $Q = \{\mathbf{q}_i\}$ be a sequence of poses. The cost of Q is given by

$$C(Q, \Theta) = \sum_i \sum_j ||\mathbf{q}_i - \mathbf{q}_j||^2 \theta_{ij} \tag{4.6}$$

While one could optimize C by selecting an initial Q and performing gradient descent over the set of poses, such an approach would fail to ensure that the optimal pose sequence is a reasonable outcome from the action sequence A. Instead, an iterative Monte Carlo mechanism is employed whereby, given an initial sequence Q, an

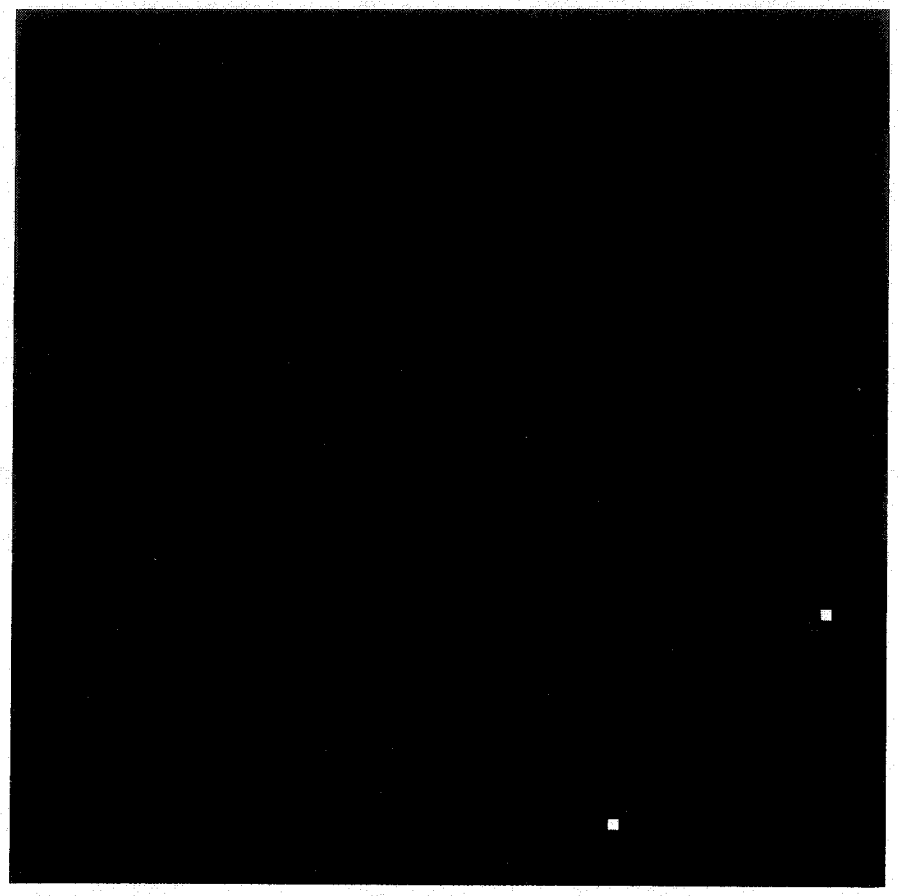


FIGURE 4.11. Conditioned similarity matrix using control trajectory plotted at right.

action A_i is selected at random, a new outcome for that action is sampled stochastically from a motion model for the robot, and the sequence Q is adjusted according to the new outcome, producing a new sequence Q'. Adjusting Q will involve adjusting every pose occurring after the sampled action A_i. If the cost of the adjusted sequence Q' is less than the original sequence, Q is set to the new sequence, otherwise Q' is rejected. The algorithm iterates indefinitely, stopping when a sufficient amount of time has passed without generating an outcome that improves the cost of Q.

Figure 4.12 depicts the evolution of the pose trajectory at iterations 1, 10, 100, 1000 and 10000. From this sequence, it is apparent that convergence occurs within

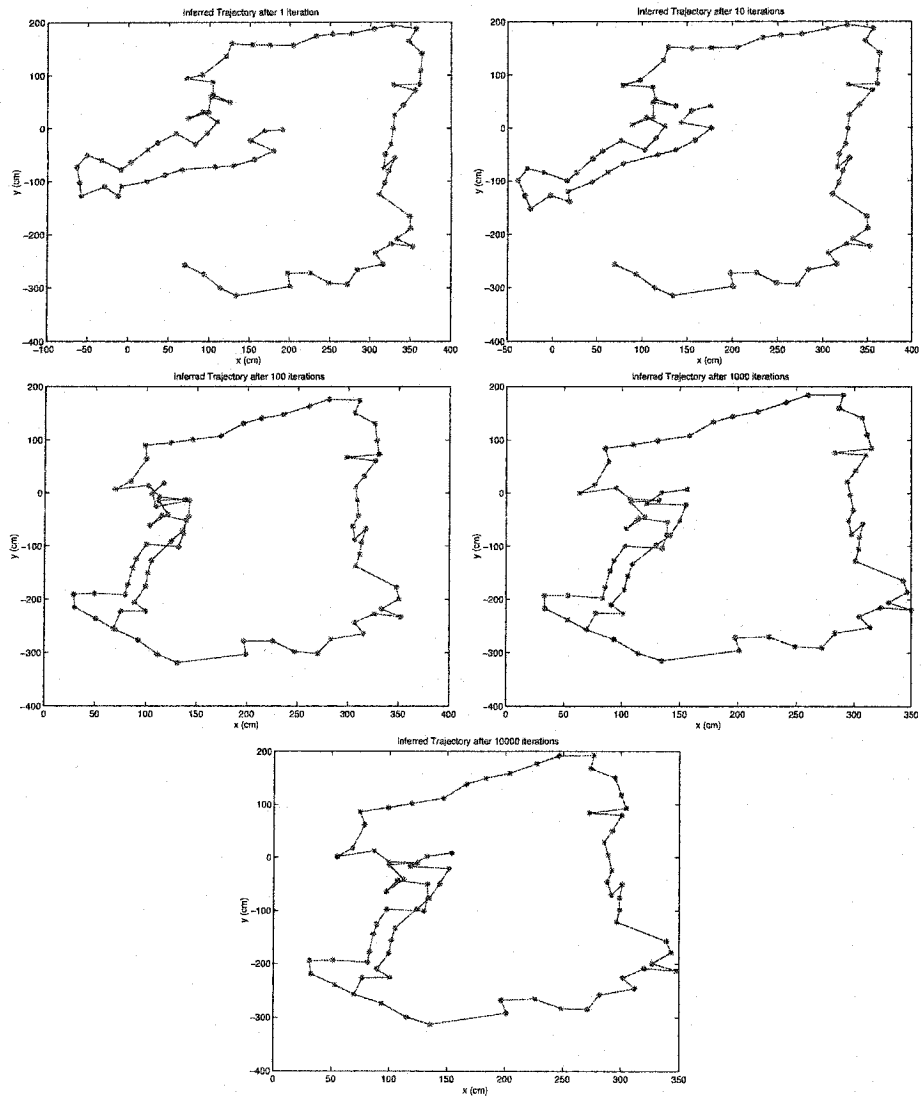


FIGURE 4.12. Inferred trajectory after 1, 10, 100, 1000, and 10000 iterations.

about 1000 iterations, or about ten seconds on a 1.6GHz Pentium 4 computer. While the resulting trajectory is not identical to ground truth (Figure 4.9), it should be noted that the simulated input is derived from a very noisy robot odometry model, which limits the constraints that the action sequence can impose on the result. Clearly there is a trade-off involved between constraining the odometry model and enabling sufficient flexibility in the Monte Carlo optimization to close the loop.

8. Discussion

This chapter presented an approach to spatially organizing images of an unknown environment using little or no *a priori* positional knowledge. The repeated occurrences of visual features in the images allows us to accomplish this. The visual map of the environment that is produced is topologically correct and also demonstrates a substantial degree of metric accuracy. The result can be described as a locally conformal mapping of the environment. This representation can then be readily used for path execution, trajectory planning and other spatial tasks.

While several authors have considered systems that interleave mapping and position estimation, this work is among the first to do this based on monocular image data. In addition, unlike prior work which typically uses odometry to constrain the localization process, mapping can be accomplished with essentially no prior estimate of the position the measurements are collected from. On the other hand, if some positional prior is available we can readily exploit it. In the second example shown we exploited such a prior. Even in this example, it should be noted that the data acquisition trajectory was one that did not include cycles.

The absence of a requirement for an informative *a priori* distribution over position (i.e. an explicit odometry model) makes this approach suitable for unconventional mapping applications, such as the integration of data from walking robots or from manually collected video sequences. The ability to do this depends on the repeated occurrence of visual features in images from adjacent positions. This implies that successful mapping depends on stable feature tracking between successive frames.

For environments with large cycles, the latter part of this chapter demonstrated how the co-visibility of features can be exploited along with odometric information to produce a cost function over trajectories. A mechanism was then presented for inferring a closed loop while maintaining consistency with the control inputs. Consistency with the control input is a feature that can be lost in applications that rely solely on the observations to register the most likely map. Clearly, these results can

be combined with the results from the first half of the chapter to achieve greater accuracy in the inferred map.

Having demonstrated the mapping capabilities of the visual mapping framework, the following chapters will apply the framework to an examination of simultaneous localization and mapping, and will posit the question of how a robot's exploration trajectory impacts the accuracy of the map that it builds.

CHAPTER 5

Simultaneous Localization and Mapping: Evaluating Exploration Strategies

1. Overview

This chapter considers the effect of exploration policy in the context of the autonomous construction of a visual map of an unknown environment. The problem is an instance of simultaneous localization and mapping (SLAM), whereby odometric uncertainty presents the problem of introducing distortions into the map which are difficult to correct without costly on-line or post-processing algorithms. The problem is further compounded by the parameter-free nature of the triangulation-based feature representation, which is designed to accommodate a wide variety of visual phenomena without assuming a particular imaging platform. This path of least commitment precludes the exploitation of constraints provided by assumptions about scene and camera geometry. The representation also presents a requirement for a relatively dense sampling of observations of the environment in order to produce reliable models.

The goal of this chapter is to develop an on-line policy for exploring and collecting observations of an unknown environment which minimizes map distortion while maximizing coverage. As was observed in the self-organizing case, there is a tendency for the inferred feature models to lead to errors in the map, as the nonlinear

effects of imaging geometry accumulate. Rather than applying costly post-hoc expectation maximization approaches to improve the output, an extended Kalman filter (EKF) is used to localize each observation once, and the exploration policy becomes an important factor in ensuring that sufficient information is available to localize the successive observations. This chapter will present the approach taken to the problem, and experimental results will be presented in the following chapter.

2. Introduction

The previous chapter presented an approach to constructing a visual map when limited prior information is available about the camera positions of the training images. In many robotic contexts, such as that presented in Section 4.7, it is usually possible to provide stronger priors, such as an estimate of the robot's trajectory based on control inputs. This chapter will consider the problem of automatically exploring an environment and constructing a visual map. In particular, we are interested in selecting an exploration strategy which minimizes map uncertainty on-line. Such uncertainty is accumulated by errors in odometric sensing and modelling errors and biases, and can grow unbounded over time. This work differs from other exploration techniques in that the map representation is implicit in nature; that is, there is no explicit representation or recovery of the geometry of the interaction between camera and environment. As such, it is not immediately suited to standard Kalman filter, or expectation maximization techniques, which usually estimate the state of the robot and the positions of a set of landmarks in a global frame of reference [**69, 158, 143**].

The approach taken here is aimed using a purely on-line exploration paradigm to produce a map that is suitable for robotic navigation. Of course, in many contexts it might be desirable to post-process such a map to further optimize its accuracy, but the current work is motivated by the supposition that even in such cases a good initial map is helpful. We will examine a variety of exploratory trajectories, both in simulation and using a real robot, and present experimental results.

The reader will recall that visual maps encode visual landmarks in the image domain and do not require camera calibration, enabling the encoding of a wide variety of visual phenomena, including exotic phenomena such as specularity, shadowing and atmospheric hazing, using arbitrary imaging geometry. The challenge posed by such a representation is that, unlike geometric landmarks, it is not well-suited to filters which aim to compute maximum-likelihood parameterizations. This poses an interesting challenge for autonomous mapping– how to maximize the map likelihood without an explicit representation. It should also be noted that, by definition, the visual map representation makes no prior assumptions about imaging geometry, so a further challenge is to infer the correct map without linearizing away the nonlinear interactive properties of the environment and sensor. Finally, the representation also poses the requirement for a relatively dense sampling of observations covering the pose space. This presents additional challenges in that the exploratory trajectory can be quite long, even in a small environment.

With these challenges in mind, the goal is to develop a technique for maximizing coverage of a relatively small pose space in order to generate an accurate visual map. The mapping process can be made more robust by composing a large map using a set of smaller sub-maps (e.g. [**65, 131, 152, 12**]). These principles are applicable to the mapping context described here, but can be viewed as a secondary stage of processing and control.

The subsequent sections will examine previous work on the SLAM problem and then go on to establish a framework for exploration and discuss several candidate exploration policies. The following chapter will present experimental results based on both simulation and validation in the real world using a variety of candidate exploration policies, and discuss their implications.

3. Previous Work

The problem of simultaneous localization and mapping, also known as concurrent mapping and localization (CML) has received considerable attention in the robotics

community [**133, 68, 143, 157, 7, 35, 155, 10, 89, 81**]. The state of the art in SLAM can be broadly subdivided into one of two approaches (and various hybrids). One family of methods collects measurements and incrementally builds the map while the robot moves (i.e. in an on-line fashion), whereas the other family of methods post-processes the trajectory and observation data in an off-line fashion.

Early approaches to SLAM involved approximating the probability distribution describing the pose of the robot as a Gaussian distribution with mean $\hat{\mathbf{q}}$ and covariance P. Smith, *et al.* and Leonard and Durrant-Whyte pioneered this approach in which the map is represented as a set of landmarks derived from a range sensor, and a Kalman filter is employed to propagate the pose distribution according to a *plant*, or *motion* model describing the uncertain outcome of the robot's actions, and an *observation* model describing the relationship between poses and observations [**133, 68**].

Formally, the plant model describes the motion of the robot due to an action $u(t)$ at time t:

$$\hat{\mathbf{q}}(t+1|t) = F(\hat{\mathbf{q}}(t), u(t)) \tag{5.1}$$

$$P(t+1|t) = \nabla F P(t|t) \nabla F^T + Q(t) \tag{5.2}$$

where F is a function that describes the outcome of an action in the absence of noise, ∇F is the Jacobian of F evaluated at $\hat{\mathbf{q}}$, and Q is the covariance of the action noise model[1]

While the plant model describes the uncertain outcome of actions, the observation model provides the ability to improve the pose estimate by relating sensor observations to pose. For an observation $\mathbf{z}(k+1)$ define the innovation $\mathbf{v}(t+1)$ and associated

[1]The reader should take note that the plant model $F(\mathbf{q}, u)$ is a distinctly different function from the visual map observation model $\mathbf{z} = F(\mathbf{q})$. Throughout this chapter, the function $\mathbf{h}(\mathbf{q})$ denotes the observation model.

covariance $S(t+1)$ as

$$\begin{aligned} \mathbf{v}(t+1) &= \mathbf{z}(t+1) - \mathbf{h}(\hat{\mathbf{q}}(t+1|t)) && (5.3) \\ S(t+1) &= E\left[\mathbf{v}(t+1)\mathbf{v}^T(t+1)\right] && (5.4) \\ &= \nabla\mathbf{h}P(t+1|t)\nabla\mathbf{h}^T + R(t+1) && (5.5) \end{aligned}$$

where $\mathbf{h}(\cdot)$ is a generative model describing the expected observation as a function of pose $\mathbf{q}$, R is a covariance matrix describing the sensor noise model, and $E[x]$ indicates the expectation of the random variable x. $\nabla\mathbf{h}$ is the Jacobian of $\mathbf{h}(\cdot)$. The innovation describes the extent to which the current observation differs from what the robot expects to see from its current pose estimate.

Given the innovation $\mathbf{v}(t+1)$, and covariance S the pose estimate can be updated by first computing the *Kalman gain* $W(t+1)$:

$$W(t+1) = P(t+1|t)\nabla\mathbf{h}^T S^{-1}(t+1) \tag{5.6}$$

and applying W to transform the innovation into a pose displacement:

$$\hat{\mathbf{q}}(t+1|t+1) = \hat{\mathbf{q}}(t+1|t) + W(t+1)\mathbf{v}(t+1) \tag{5.7}$$

and covariance

$$P(t+1|t+1) = P(t+1|t) - W(t+1)S(t+1)W^T(t+1). \tag{5.8}$$

For nonlinear F and/or $\mathbf{h}(\cdot)$, the update equations serve only as a first-order approximation and the filter is referred to as an Extended Kalman Filter.

The robustness of the Kalman filter can be improved by partitioning the observation $\mathbf{z}$ into a set of separate landmark observations $\mathbf{z}_i$ and removing those observations for which the innovation is too large, according to the log of their likelihood:

$$\log(p(\mathbf{z}_i|\hat{\mathbf{q}}(t+1|t)) = \mathbf{v}_i^T(t+1)S_i^{-1}(t+1)\mathbf{v}_i(t+1) \tag{5.9}$$

Estimates whose log-likelihood exceeds a user-defined threshold on g^2 are removed:

$$\mathbf{v}_i^T(t+1)S_i^{-1}(t+1)\mathbf{v}_i(t+1) > g^2 \tag{5.10}$$

The retained estimates are subsequently recombined into an observation vector and the pose estimate computed as above. In the case of a range-sensor landmark-based approach, the observation model is equivalent to computing a least-squares triangulation, taking into account the sensitivity of the observation due to viewpoint.

It should be noted that this EKF approach assumes a fixed map. Most modern applications of the EKF to SLAM minimize the total uncertainty of the robot pose and the individual landmark positions, where the vector $\mathbf{q}$ describes the full state of the world: robot pose and landmark position, all with associated covariances [**69, 41**]. While there is a related increase in computational cost, it is mitigated somewhat by the fact that most of the linear systems involved, $S(t+1)$ in particular, are sparse.

While the EKF is computationally elegant, it is a first-order approximation and can diverge from the true distributions being modelled. In response, some researchers have employed particle filters to represent the underlying distributions [**20, 53**]. The particle filter is appealing in that with enough samples any arbitrary probability distribution can be represented. In practise, limited computational resources impose constraints on the number of samples, and as such the particle filter can suffer from issues similar to the divergence of the Kalman Filter. The observation that S is sparse led Montemerlo *et al.* to develop a particle filter-based approach called *FastSLAM* that factorizes the underlying probability density functions, taking advantage of the decoupling between sub-matrices in S [**80, 81**]. This approach scales logarithmically in the number of landmarks, a significant improvement over the quadratic-time Kalman Filter update.

The second family of methods for SLAM involves first collecting measurements and then post-processing them in a batch. The standard post-processing method is to employ Expectation Maximization (EM), again to minimize the total uncertainty of robot poses and landmark positions (Section 4.3) [**22, 143**]. One goal of this chapter

is to develop an on-line exploration method which maximizes the accuracy of the map without resort to expensive map updating.

Of particular relevance to this chapter is the problem of planning a trajectory for minimizing uncertainty while maximizing the utility of the observed data. This question has been examined in other domains. For example, MacKay considered the problem of optimally selecting sample points in a Bayesian context for the purposes of inferring an interpolating function [**77**]. Whaite and Ferrie employed this approach as motivation for their "curious machine", a range-finder object recognition system that selected new viewing angles in order to maximize the information gained about a parameterized model of object shape [**154**]. Arbel and Ferrie further applied this approach to appearance-based object models, selecting the viewing angle that maximized the ability to discriminate between objects [**1**]. It is important to note that MacKay, in particular, and Ferrie *et al.*, to a lesser extent, provide analytical results defining optimal exploration trajectories based on a strict set of assumptions about the objects being modelled and the distribution of their best-fit parameters. For general representations of the environment, and for visual maps in particular, it may be impossible or at least very difficult to extrapolate those analytical results to strong conclusions about ideal exploration methods.

The exploration problem has also received some attention in the robotics community. Moorehead *et al.*, considered the problem of maximizing information gain from multiple sources in order to control an exploratory robot [**82**], Taylor and Kriegman examined a variety of exploration strategies for constructing a topological visual representation of the environment [**140**], and Stachniss and Burgard demonstrated an exploration mechanism exploiting the idea of coverage [**134**]. Finally, Roy *et al.* applied the principle of minimizing uncertainty to the problem of robotic path planning, instantiating the problem as a partially observable Markov decision process [**105**]. The tendency of the path planner is to compute routes where sensor noise is expected to be low and pose constraints are expected to be high, the result being that the robot tends to follow the boundaries of obstacles and walls.

4. Problem Statement

Formally, this chapter considers the following problem:

Given: Γ, a set of policies for exploring an unknown environment and collecting image data.

Execute: each policy in Γ, constructing a visual map as data is collected and updating robot pose from the constructed map.

Evaluate: each policy based on the coverage and accuracy of the resulting visual map.

The following sections will present the approach to exploration that we will employ, followed by a presentation of the exploration policies under consideration.

5. Exploration Framework

We will adapt the EKF localization framework described above and developed in the seminal papers by Smith *et al.* and Leonard and Durrant-Whyte as the basis for the exploration framework used in this thesis [**133, 68**]. While the work by Leonard and Durrant-Whyte employed "geometric beacons" derived from range sensors as landmarks, the visual map representation instead employs feature observations in the visual domain. It should be noted that, unlike EKF implementations which encode both robot pose and global landmark position parameters, the only parameters maintained in this implementation are those of the robot pose.

Throughout the experiments described in this and the next chapter, we will employ the triangulation-based feature model described in Section 4.4. Given the EKF formulation described in the previous section, the following discussion will cover only those aspects of the implementation that are particular to this work.

At each time step t, the robot executes an action $u(t)$, and takes a subsequent observation $\mathbf{z}$. The plant model is updated from u according to Equation 5.1, and a set of matches to known features $\mathbf{z}_i$ are extracted from the observed image using the mechanism described in Section 2.8. Given that the visual map assumes a 2D

configuration space, (that is, fixed orientation) some rehearsal procedure will be required to align the camera prior to taking an observation– we will consider this issue in further detail in presenting the experimental results.

For each successfully matched feature, a predicted observation $\hat{\mathbf{z}}_i$ is generated using the current visual map, and the innovation $\mathbf{v}_i(t+1)$ is computed

$$\mathbf{v}_i(t+1) = \mathbf{z}_i(t+1) - \mathbf{h}_i(t+1, \hat{\mathbf{q}}) \tag{5.11}$$

$$= \mathbf{z}_i(t+1) - \hat{\mathbf{z}}_i(t+1, \hat{\mathbf{q}}) \tag{5.12}$$

where $\hat{\mathbf{z}}_i(t+1)$ is the predicted observation of feature i according to the learned generative model.

The innovation covariance requires estimation of the Jacobian of the predicted observation given the map and the plant estimate. This is defined as the matrix of partial derivatives of the observation given the pose:

$$\nabla \mathbf{h} = \begin{bmatrix} \frac{\delta \mathbf{z}_0}{\delta \mathbf{q}_0} & \cdots & \frac{\delta \mathbf{z}_0}{\delta \mathbf{q}_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta \mathbf{z}_m}{\delta \mathbf{q}_0} & \cdots & \frac{\delta \mathbf{z}_m}{\delta \mathbf{q}_n} \end{bmatrix} \tag{5.13}$$

We compute this Jacobian using the triangulation-based model by computing the slope of the planar patch defined by the observations associated with the face of the pose triangulation containing $\hat{\mathbf{q}}$.

The innovation covariance follows the standard model defined in Equation 5.5 and repeated here:

$$S_i(t+1) = \nabla \mathbf{h}_i P(t+1|t) \nabla \mathbf{h}_i^T + R_i(t+1) \tag{5.14}$$

where P is the pose covariance following the action u, and R is the cross-validation covariance associated with the learned feature model. It is important to note that R serves several purposes– it is simultaneously an overall indicator of the quality of the interpolation model, as well as the reliability of the matching phase that led to

the observations that define the model; finally it also accommodates the stochastic nature of the sensor.

5.1. Outlier Detection. Feature correspondence takes place once an observation image is obtained. However, there may be outlier matches that must be filtered out. As such, the gating procedure described by Equation 5.10 is employed, with the additional constraint that the gating parameter g is computed adaptively. Specifically, we accept feature observations that meet the constraint

$$\mathbf{v}_i^T(t+1)\mathbf{S}_i^{-1}(t+1)\mathbf{v}_i(t+1) \leq g^2 \tag{5.15}$$

where

$$g = \max(g_{base}, \bar{g} + 2\sigma_g), \tag{5.16}$$

g_{base} is a user defined threshold, and $\bar{g}$ and σ_g are the average and standard deviation of the set of gating values computed for each feature observation (that is, the left-hand side of Equation 5.15)[2]. This selection of g allows the filter to correct itself when several observations indicate strong divergence from the predicted observations–indicating a high probability that the filter has diverged and affording the opportunity to correct the error.

5.2. Map Update. Given the set of gated feature observations $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$, the EKF is updated according to the standard formulation, whereby the set of filtered innovation measurements is compounded into a single observation vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_n \end{bmatrix} \tag{5.17}$$

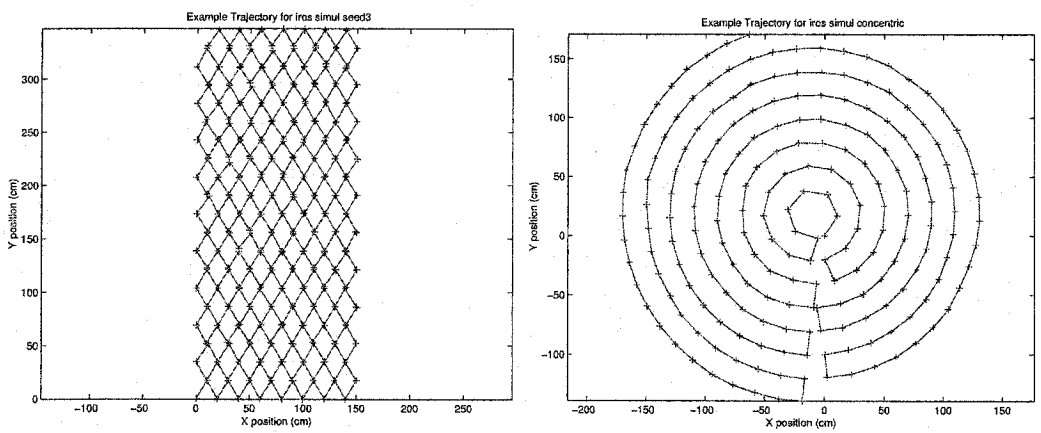
and the Kalman gain W is computed according to Equation 5.6. Combined with the plant model, a pose estimate $\hat{\mathbf{q}}$ and associated covariance P are obtained from Equations 5.7 and 5.8. Once an updated pose estimate is available, the successfully

[2]Throughout the experimentation $g_{base} = 5.0$.

matched features are inserted into the visual map, using the estimated pose as their observation pose. It should be noted that we also insert those observations that were removed by the gating procedure into the map. This approach is taken because it serves to increase the cross-validation covariance associated with the mis-matched feature, thereby reducing its influence for future localization. As such, at the end of the exploration procedure, only those features that serve to match reliably *and* facilitate reliable localization can be selected and retained.

The next section will consider the problem of selecting exploration trajectories that result in an accurate map using the EKF framework.

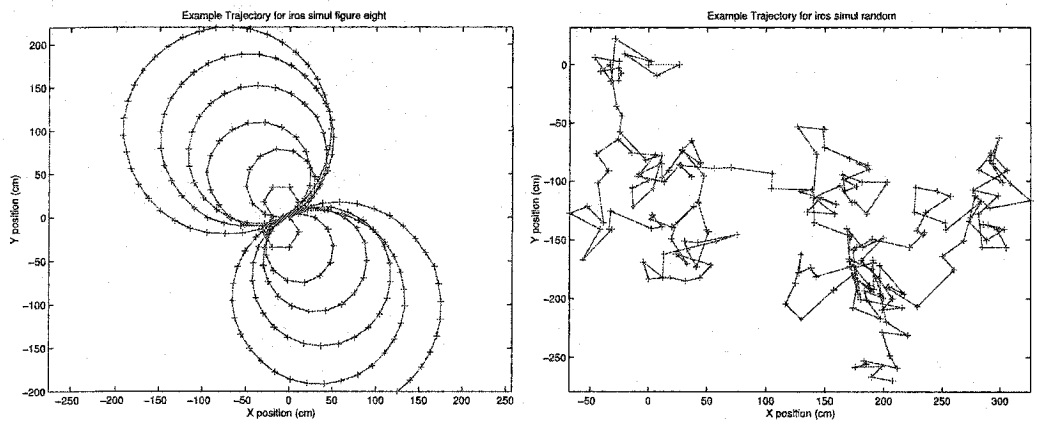
6. Exploration Policies



(a) SeedSpreader (b) Concentric

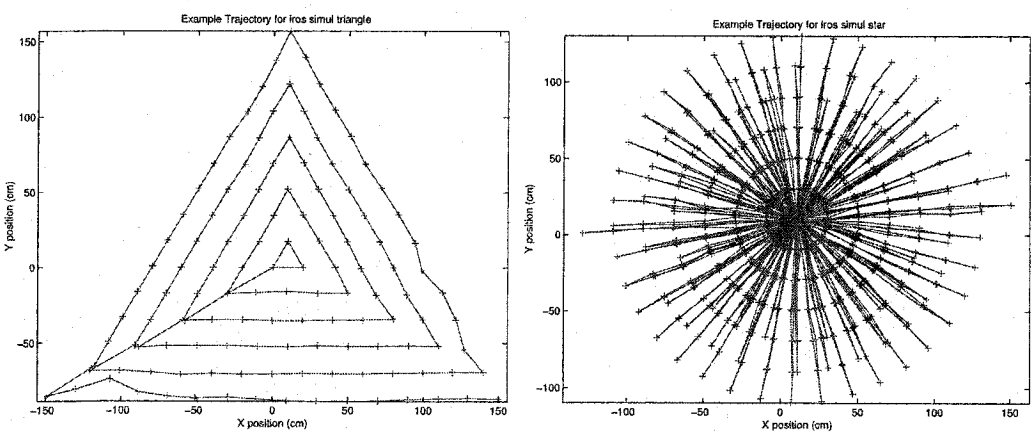
FIGURE 5.1. `SeedSpreader` and `Concentric` policies.

We are interested in comparing candidate robot exploration policies with the goal of balancing two competing interests: coverage and accuracy. In other words, we want to build the largest, most accurate map possible in a finite amount of time. Given that there are an infinite number of possible exploration trajectories, we will restrict our consideration to a set of policies which are either intuitively satisfying or serve to illustrate an extreme case. The particular policies we will examine are described



(a) FigureEight (b) Random

FIGURE 5.2. `FigureEight` and `Random` policies.



(a) Triangle (b) Star

FIGURE 5.3. `Triangle` and `Star` policies.

below. They are `SeedSpreader`, `Concentric`, `FigureEight`, `Random`, `Triangle` and `Star`. An example of each trajectory is depicted in Figures 5.1, 5.2, and 5.3.

`SeedSpreader`. The robot follows a seed-spreader pattern through the environment [**76**]. A variation on this approach is employed by oscillating in a zig-zag motion as the robot moves along the path. This is performed in order to ensure that the visual map spans two dimensions, even along the first pass of the seed spreader.

Specifically, for each sweep of the seed spreader, the robot takes ten steps, rotating to 60 degrees, translating 20cm, rotating to -60 degrees and translating another 20cm. At the end of the sweep, the robot rotates 180 degrees and reverses course.

`Concentric`. The robot traces a series of concentric circles out from its starting point, reversing direction for alternating circles. Each circle is decomposed into line segments of 20cm in length, indicating a rotation and a 20cm translation by the robot, and the interval between concentric circles is also 20cm.

`FigureEight`. Like the `Concentric` pattern, the robot traces a set of concentric circles, but in a series of growing figure-eights, bringing the robot close to its starting point with each pass. In this case, the radius of each successive '8' increases by 20cm, and, like the `Concentric` pattern, the curves are decomposed into segments 20cm in length.

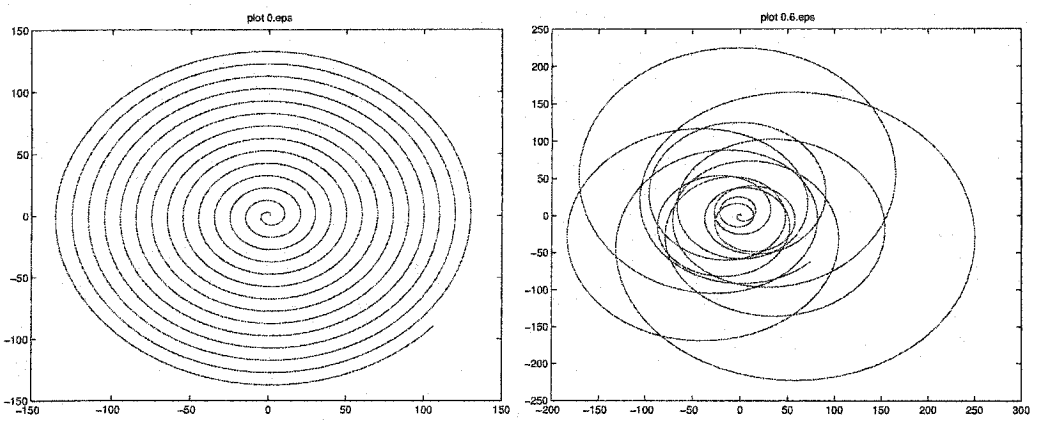
`Random`. The robot takes a random step in a random direction at each time step. The random step is normally distributed with a mean of 20cm and standard deviation of 10cm.

`Triangle`. The robot traces a series of concentric equilateral triangles, taking observations at fixed intervals of 20cm along each side. The advantage of this approach is that the ideal set of observation poses covers the pose space in a uniform tiling. The length of the sides of the triangles increases by 40cm for each successive triangle.

`Star`. The robot oscillates along a set of rays emanating from its starting point. The rays grow in length at 20cm increments over time such that the set of observation poses is roughly the same as that for `Concentric`.

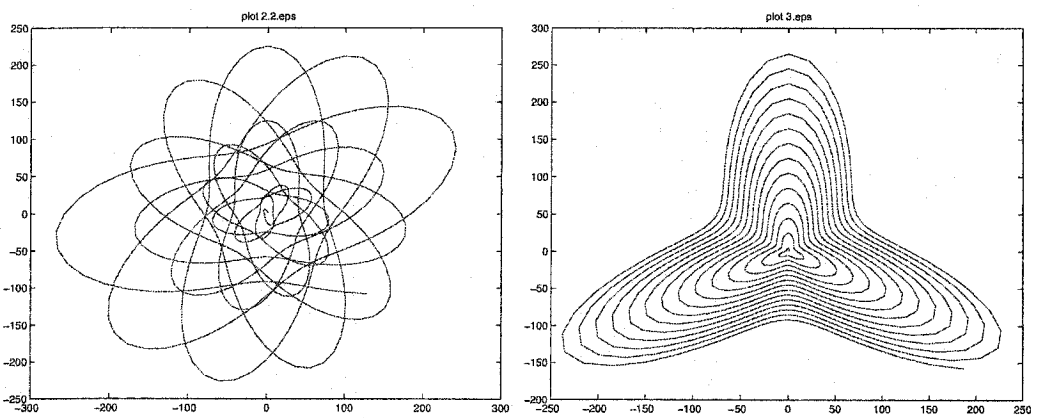
7. Parameterized Trajectories

In addition to the heuristically determined trajectories, it is valuable to consider a family of parametrically defined trajectories that satisfy some of the desirable properties of the heuristic set. Recent work in mathematics suggests that a wide variety of trajectories and shapes can be expressed with a simple parametric function [**38**]. We will examine an analytic family of trajectories, parameterized over a single parameter,



(a) n=0 (b) n=0.6

FIGURE 5.4. Sample trajectories for a variety of values of n.

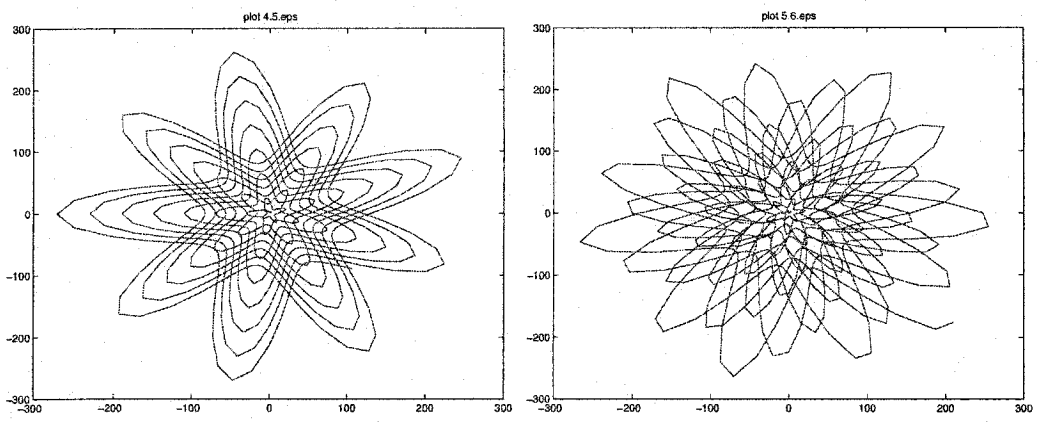


(a) n=2.2 (b) n=3

FIGURE 5.5. Sample trajectories for a variety of values of n.

that aims to capture the variety of properties that are important for accurate and efficient exploration. The specific parametric curve under consideration is expressed as the distance r of the robot from the origin as a function of time:

$$r_n(t) = \frac{kt}{2 + \sin nt} \tag{5.18}$$



(a) n=4.5 (b) n=5.6

FIGURE 5.6. Sample trajectories for a variety of values of n.

where k is a dilating constant that is fixed for the experiments and n parameterizes the curve in order to control the frequency with which the robot moves toward the origin. Some examples of the curve for a variety of values of n are shown in Figures 5.4, 5.5 and 5.6. Note that in the extreme cases, the curve never moves toward the origin ($n = 0$), or will do so with very high frequency ($n \to \infty$). Also of interest are integral values of n, where the curve never self-intersects, and has n distinct lobes. Finally, note that from an efficiency standpoint, the new space covered as a function of t decreases roughly monotonically as n increases, since for larger n the robot spends an increasing amount of time in previously explored territory.

The next chapter will present experimental results illustrating the performance of the exploration framework for the various candidate trajectories.

CHAPTER 6

Evaluating Exploration Strategies: Experimental Results

1. Overview

This chapter presents an experimental analysis of the exploratory policies considered in the previous chapter. The results from the heuristically determined policies, which are presented in both simulated and real environments, will provide further motivation for the parametrically determined policy and will demonstrate that with an effective policy an accurate map can be constructed.

2. Implementation

2.1. Exploration Model. In all cases, the exploration model follows a `Plan, Act, Observe, Update` loop, planning a motion based on the exploration policy and the covered trajectory, executing the action, taking an observation, and updating the Kalman Filter and visual map (Figure 6.1). A single action is either a rotation or a translation, and new images are obtained only after a translation. Furthermore, while the filter is always updated after obtaining a new image, the image is only added to the visual map if there are no previous images from nearby poses in the map. When the robot traverses previously explored territory, it localizes without updating the map.

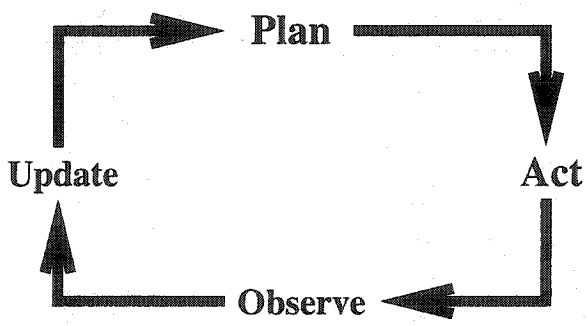


FIGURE 6.1. The `Plan, Act, Observe, Update` cycle. At each time step, the robot formulates a plan and executes it (`Plan, Act`). It subsequently acquires an observation and updates its knowledge about the world– its map, and its pose estimate (`Observe, Update`). The loop repeats indefinitely.

2.2. Safety. In all robotic applications, special attention must be devoted to the safety of the robot and other agents in the environment. Given that the visual map does not encode geometric information, obstacle inference and avoidance requires careful consideration. For the experiments presented here, a sonar ring mounted on the robot acts as a virtual bumper, using the last sonar observation to determine whether an intended action is safe or unsafe to execute.

3. Experimental Results: Heuristic Trajectories

3.1. Simulation. The experimental setup for the simulated experiments is as follows: a simulated robot is placed in a rectilinear room of dimensions 1200cm by 600cm. The camera model is simulated by texture-mapping the walls and ceiling with real images of our laboratory, and rendering the scene based on the ground-truth pose of the robot and a model of a simple perspective camera. It is assumed that the camera has the ability to align itself using a procedure which is external to the robot drive mechanism, possibly using a compass and pan-tilt unit or an independent turret, such as that which is available on a Nomad 200 robot (Figure 3.4). Using this procedure, when an image observation is required, the camera snaps two images, one along the global x axis and one along the global y axis, and returns the composite image (Figure 6.3). Figure 6.2 illustrates a typical image returned by the

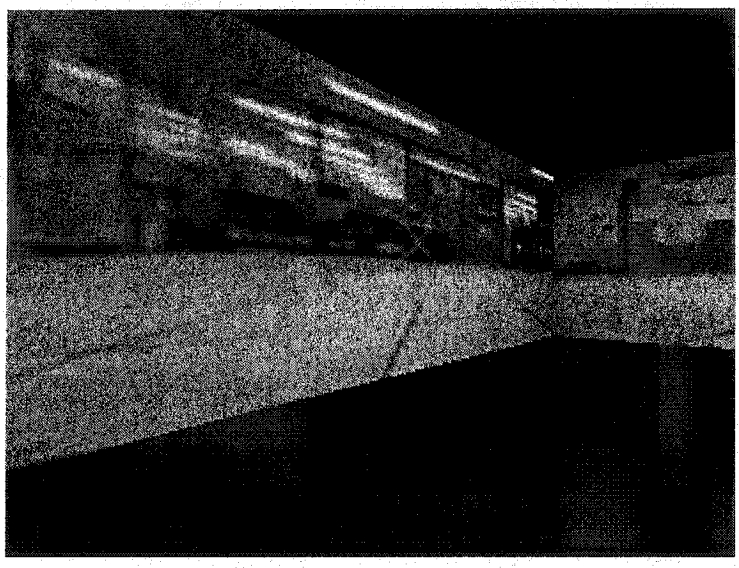


FIGURE 6.2. Simulated camera view.

camera in one direction in the simulated environment. While the environment may seem somewhat unrealistic, particularly with respect to the lack of obstacles, we are concerned primarily with the performance of the visual mapping framework under idealized conditions– in this context the visual features should be straightforward to extract and model and the results will reflect the effects of the exploratory policy.

The simulated robot has a ring of sixteen evenly spaced sonar sensors for detecting collisions, and the robot's odometry model is set to add normally distributed zero-mean, 1% standard deviation error to translations and normally distributed zero-mean, 2% standard deviation error to rotations.

Each exploratory policy was run in the simulated environment, executing actions and taking observations until one of two conditions was met: either the visual map contained two hundred images, or the robot was unable to execute its action safely. The starting pose of the robot was selected to be the centre of the room, except in the case of the `SeedSpreader`, which started in the corner.

The results of the experiments are tabulated in Table 6.1. For each policy the mean deviation between the filter pose estimate and ground truth and the mean deviation between odometry and ground truth are reported. Also recorded is an

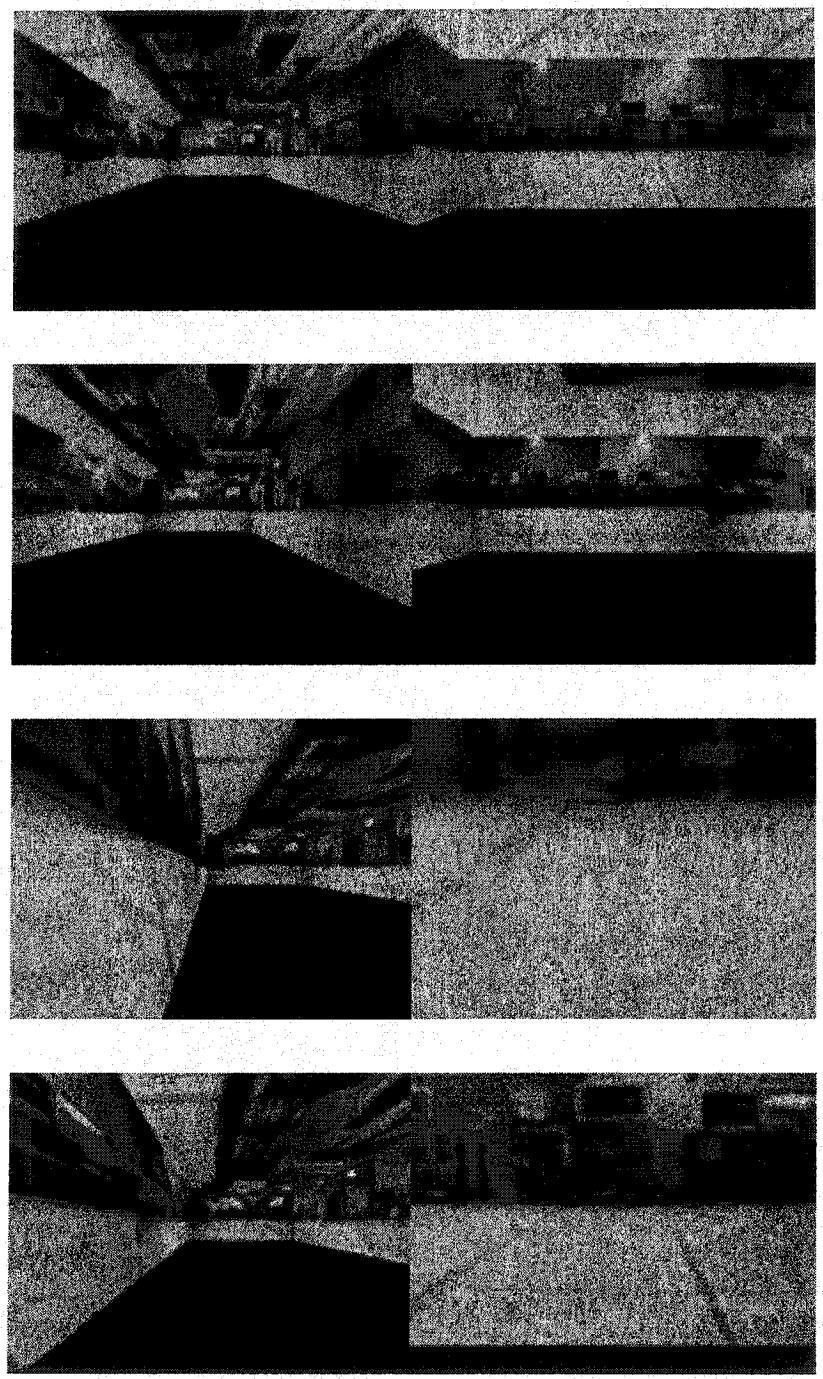


FIGURE 6.3. Example composite observations: A single observation entails acquiring images in two orthogonal directions.

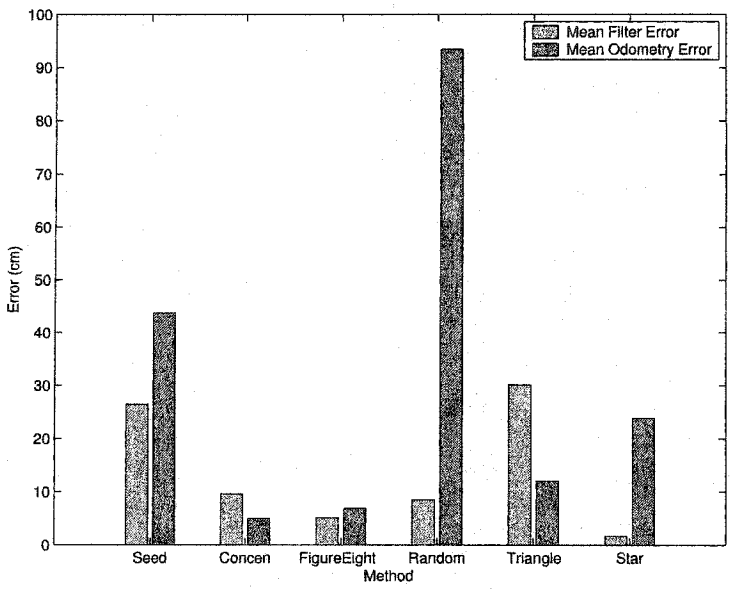
TABLE 6.1. Summary of exploration results by exploration policy.

Method	Mean Filter Error (cm)	Mean Odometric Error (cm)	Exploration Efficiency (images/step)	Maximal Distance from O (cm)
`SeedSpreader`	26.4	43.7	0.178	373
`Concentric`	9.57	4.95	0.496	183
`FigureEight`	5.09	6.87	0.411	242.9
`Random`	8.46	93.4	0.475	347
`Triangle`	30.1	12.0	0.480	173
`Star`	1.63	23.8	<0.001	152

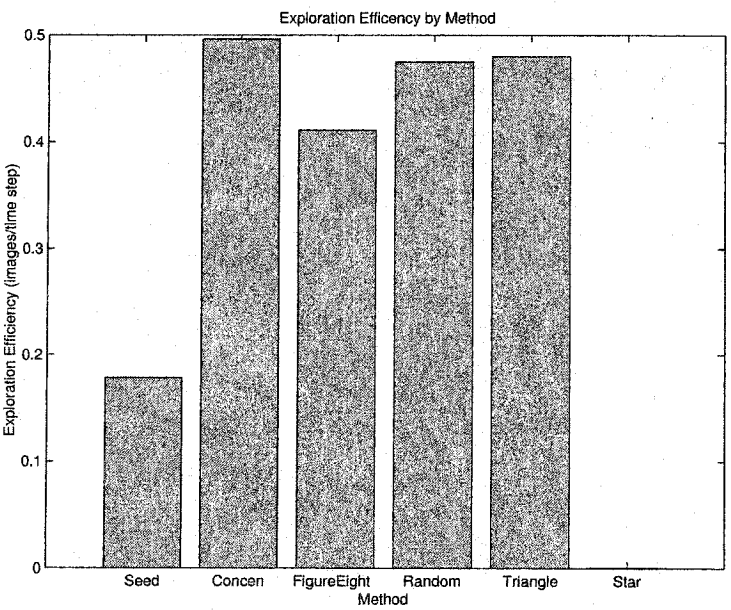
estimate of the space explored per unit time (exploration efficiency), expressed as the total number of observation images inserted into the visual map divided by the total number of actions executed by the robot. A small value indicates that the robot spent most of its time in previously explored space. Finally, the maximal distance achieved from the robot's starting pose is reported.

Figure 6.4a) summarizes the mean filter error and mean odometry error for each method and Figure 6.4b) summarizes the exploration efficiency, expressed as the number of images inserted into the map per robot action. Note that while most of the methods inserted images at a near-optimal rate (after each rotate-translate pair of actions), the `Star` policy is highly inefficient as it repeatedly traverses previously explored terrain.

Figures 6.5, 6.6, and 6.7 depict propagation of error versus ground truth for the filter and odometer for each policy, sampled at intervals of ten time steps. In each figure, the curve marked by '+' corresponds to the filter error and the curve marked by 'o' corresponds to the odometry error. It is clear from these results that the `Star` policy produced the most accurate map, and the `Random` policy performed very well relative to the accumulated error. While the good performance of the `Random` policy is likely due to the fact that it occasionally re-traverses old territory, it is no doubt an unsuitable choice for task-driven robotics. It is perhaps surprising that the `Concentric` and `FigureEight` policies do not perform more accurately. The principal



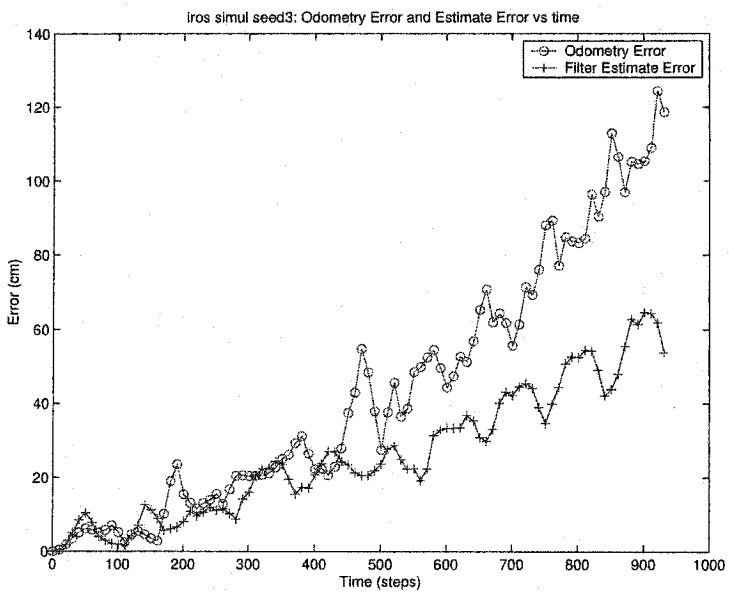
(a) Mean filter error and odometric error.



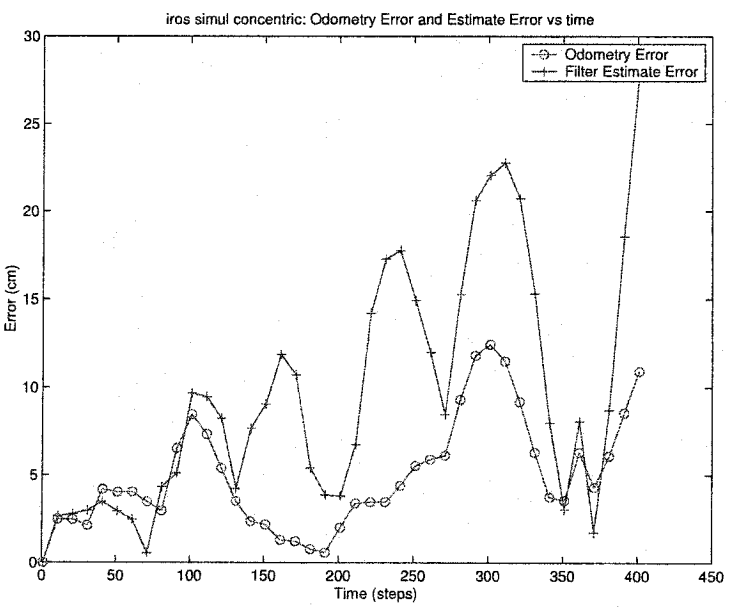
(b) Exploration efficiency.

FIGURE 6.4. a) Mean filter error and odometry error for each method. b) Exploration efficiency for each method.

cause is that as the circles get wider, errors from the inner circles are amplified by the linearization of the filter, and add bias to the localization estimates. In addition, the errors become highly correlated between successive observations, whereas such correlations are avoided using the `Star` method by re-localizing with respect to a reliable reference point (the centre of the star). Finally, note the sudden and extreme divergence of the `Triangle` method, a result of divergence in the filter as the robot failed to correctly turn the corner of one of the points of the triangle.

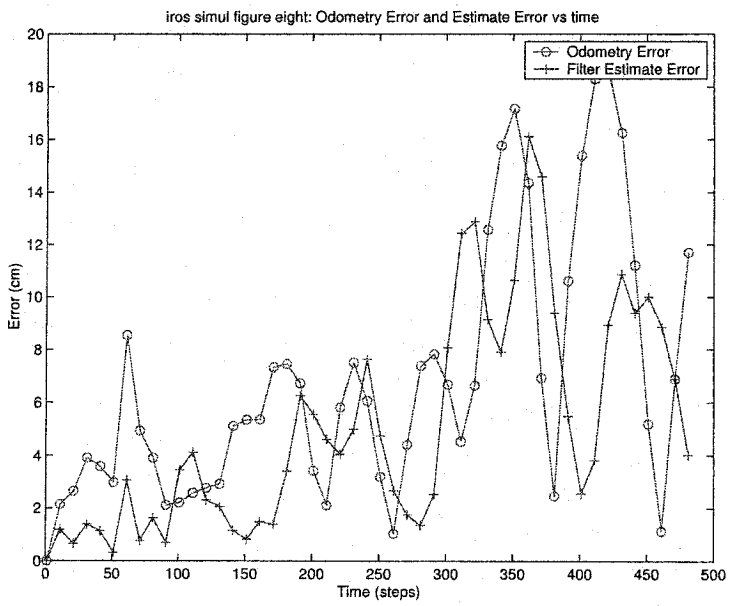


(a) SeedSpreader

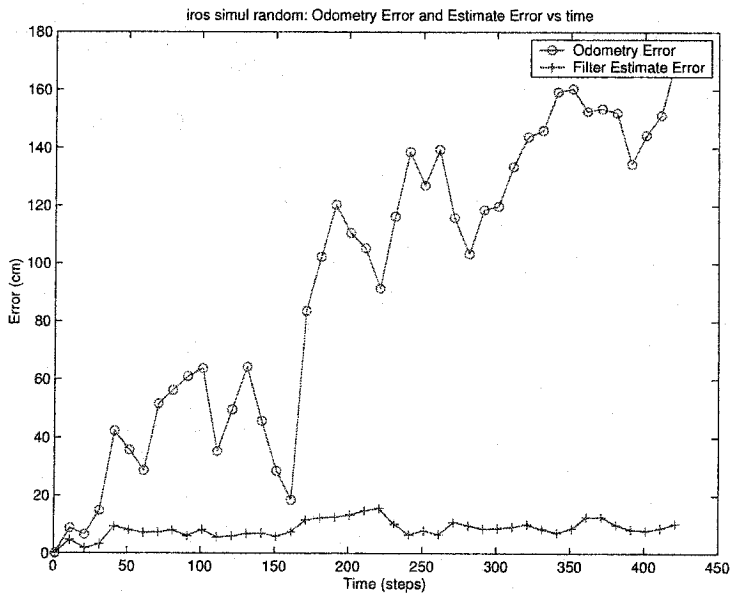


(b) Concentric

FIGURE 6.5. Filter ('+') and odometry ('o') error plotted versus time for `SeedSpreader` and `Concentric`.

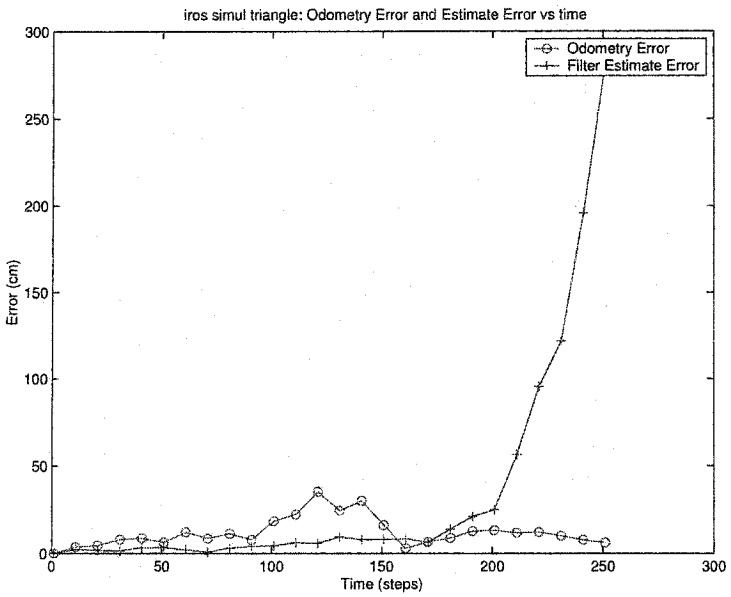


(a) FigureEight

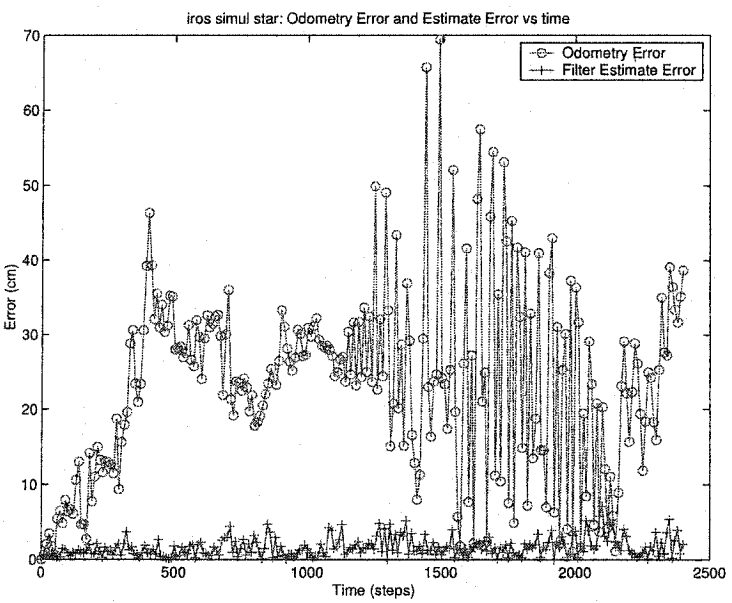


(b) Random

FIGURE 6.6. Filter ('+') and odometry ('o') error plotted versus time for `FigureEight` and `Random`.



(a) Triangle



(b) Star

FIGURE 6.7. Filter ('+') and odometry ('o') error plotted versus time for `Triangle` and `Star`.

TABLE 6.2. Final pose errors by exploration policy for the real robot. All results are in cm, and an ideal result is (0,0).

Method	Actual Final Pose (x, y)	Actual Pose Error	Odometer Pose (x, y)	Odometer Pose Error
`Concentric`	(6,-14)	15	(55,20)	59
`Star`	(21,2)	21	(171,133)	217

3.2. Real world performance. For the real-world experiments, the exploration framework was implemented on a Nomadics Super Scout mobile robot. The Scout platform employs a differential drive mechanism for steering and is particularly prone to rotation errors. The robot was equipped with a monocular camera and a KVH C100 compass. The compass was employed to align the camera– while local variations in magnetic field made the compass useless for navigation, the variations were repeatable as a function of pose and degraded smoothly enough that the robot could be steered to face in the direction of a particular heading when an image was captured. Nonetheless, some noise was observed in observation orientation, and this noise presented itself in relatively large cross-validation errors for the x image position of any given landmark. At each location the robot used the compass to acquire an image in a fixed direction. The robot commenced exploration from the centre of an open space in our lab (Figure 6.8a)). A sample image from the robot's camera is shown in Figure 6.8b).

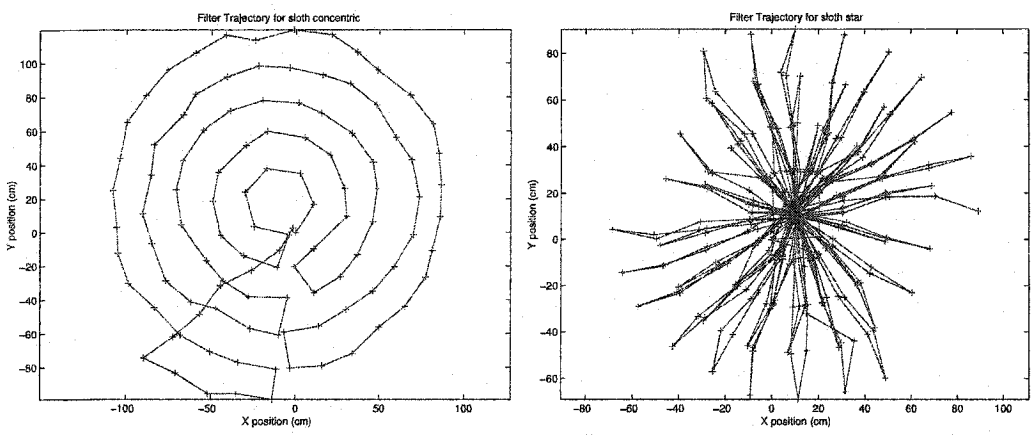
Two experiments were run on the robot, employing the `Concentric` and `Star` exploration policies respectively. Exploration continued until 100 images were inserted into the map. Since the ground truth trajectory was not available, when exploration terminated the robot was instructed to navigate back to its starting position. The discrepancy between the final position and the starting position was measured by hand. Figure 6.9 depicts the filter trajectory for each method.

The discrepancy between the robot's starting and ending positions and the magnitude error are shown in Table 6.2, and are depicted graphically in Figure 6.10. In

(a) The robot (lower left) in the environment.

(b) Robot's eye view of the scene

FIGURE 6.8. The real-world environment and robot's eye view.



(a) Concentric (b) Star

FIGURE 6.9. Filter trajectory for each method.

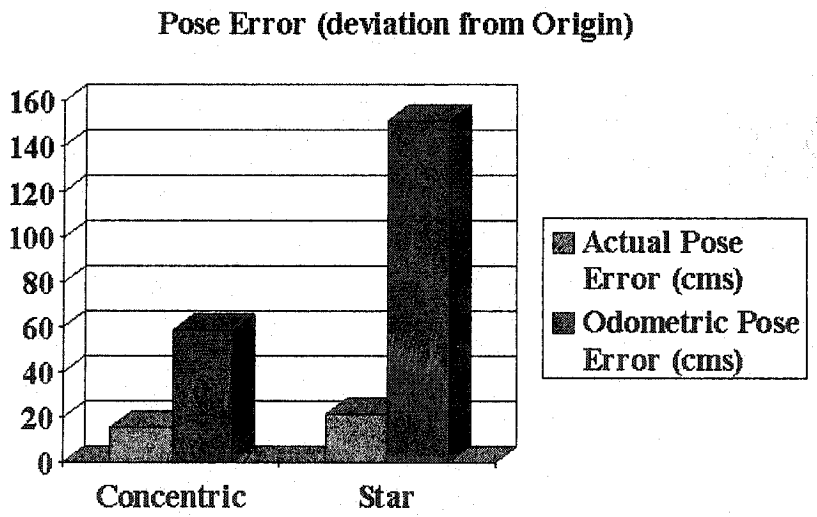


FIGURE 6.10. Pose errors for real robot.

all cases, the robot started at pose $(0, 0)$, and ended the trajectory by homing until the filter indicated a pose within 3cm of the origin. Depicted in the table are the actual ending pose (measured by hand), and the ending pose reported by the robot's odometer. All measurements are in centimetres. The filter easily out-performed odometry in both cases. However, it should be noted that this method of evaluation

does not examine the accuracy of the entire map. An examination of Figure 6.9b) suggests that the Star method did occasionally lead to divergence between the filter and ground-truth, which was later corrected when the robot returned to the origin. Future work should employ a tracker or other device to accurately measure the entire trajectory.

4. Experimental Results: Parametric Trajectories

The outcome of the initial experiments motivate a second examination of the problem using a parameterized family of exploratory trajectories. In particular, we will examine the spiral-shaped trajectory defined in Equation 5.18, parameterized in a way that controls the frequency with which it oscillates toward the origin. We will consider the properties of the generated map, including accuracy and coverage, dependent on the trajectory parameterization.

The experiments in the previous section demonstrated that the visual map representation, and in particular its interpolation-based approach to representing features poses a difficult challenge for accurate map construction. The feature models will tend to locally linearize feature behaviour, biasing the pose estimates computed in the update stage. Furthermore, once the model is updated it specializes in the neighbourhood of the updated pose, resisting the accommodation of new information. As such it is often the case that the model will rely too heavily on the accuracy of the training data, further justifying the goal of selecting an exploratory trajectory that ensures a high standard of accuracy at the outset. In particular, what is required is a training trajectory whose uncertainty is minimized relative to the bias introduced by the models.

The primary results from the previous experiments indicated that the most accurate exploratory policy, the `Star` policy, was also the most inefficient. By contrast, the most efficient policy, `Concentric`, while not the least accurate of the policies examined, demonstrated a clear tendency to propagate and amplify errors over time,

resulting in a map that was accurate near the home position but increasingly inaccurate as the circles grew. These two policies can be thought of as lying at opposite extremes of our parameterized function– at one end, the robot returns to the home position at a maximal frequency, and at the other the robot never returns. The goal of this experiment is to find a reliable middle-ground between the needs for accuracy and efficiency.

4.1. Setup. The experiments presented in this section examine the parametric trajectory defined in Equation 5.18 by applying the trajectory for a variety of parameter settings and evaluating the resulting maps. The same Extended Kalman Filter-based approach was employed as for the previous experiments. The experiments were run in the simulated environment in order to obtain accurate ground truth, and the same odometric error model was applied, with normally distributed zero-mean, 1% standard deviation error added to translations and normally distributed zero-mean, 2% standard deviation error added to rotations.

The experiments were conducted as follows: for values of $n \in [0.0, 8.0]$ at increments of 0.1, the robot was placed at the centre of the room, and the trajectory $r_n(t)$ was executed over five degree increments in t for 1000 time steps (whereby one time step involved a rotation followed by a translation). The constant k in Equation 5.18 was set to 20cm. At each pose, an observation was obtained and the Kalman Filter was updated. The visual map was updated whenever the filter indicated that the robot was more than 6.7cm from the nearest observation in the visual map. The ground-truth pose, the filter pose and the control inputs were recorded for each pose along the trajectory.

4.2. Results. The top graph of Figures 6.11, 6.12, and 6.13 depict a selection of the ground-truth trajectory ('o') plotted against the filter trajectory ('+') for selected values of n. The disparity between the two trajectories is an indicator of the accuracy of the visual map, since the poses of the images inserted into the visual map correspond to the filter poses. Given that small rotation errors near the beginning of

the trajectory can lead to large errors at the edges of the map, even if the map itself is conformal, the orientation of the filter trajectory was rotated around the starting pose to find the best fit against the ground-truth.

The bottom graph of Figures 6.11, 6.12, and 6.13 depict the filter error versus ground truth ('+') and the odometry error versus ground truth ('o') over time for the corresponding values of n. From these figures, one can observe that for some values of n, odometry out-performs the filter, whereas for other values the filter tracks ground truth more accurately.

For each value of n it is possible to compute the mean filter and odometric error over the entire trajectory. Figure 6.14a) plots the mean error values for odometry and the filter as a function of n. It is interesting to note that as n increases the odometry error tends to increase, due to the increasing total magnitude of rotations performed by the robot, but the filter error remains roughly constant. This suggests that mapping accuracy is roughly independent of the choice of n. Note, however, the prominent spikes in the filter error corresponding to neighbourhoods of integer values of n. These values correspond to trajectories that never self-intersect, or demonstrate a high degree of parallelism with nearby sweeps. As such, it appears that errors will propagate significantly or the filter may even diverge when insufficient constraints are available between neighbouring paths[1]. The lone exception to this trend is the value for $n = 0$. In this particular case, however, the small amount of rotation at each time step leads to a well-constrained plant model in the Kalman Filter.

Finally, Figure 6.14b) depicts the length of each trajectory as a function of n. The trajectory length is an approximate measure of the inefficiency of the trajectory for exploration, since the radius of the convex hull of the explored space is bounded from above by kt_{max}, where k is the scaling constant in Equation 5.18 and t_{max} is the maximal time value, a constant across the experiments. Periodic minima in the trajectory length correspond to points where the exploration was terminated prematurely because the robot was unable to safely continue to execute its policy. In

[1]This is an instance analogous to the well-known aperture problem encountered in photometric stereo[**46**].

all of these cases, the filter estimate had diverged significantly from ground truth. As expected, increasing values of n lead to increased inefficiency.

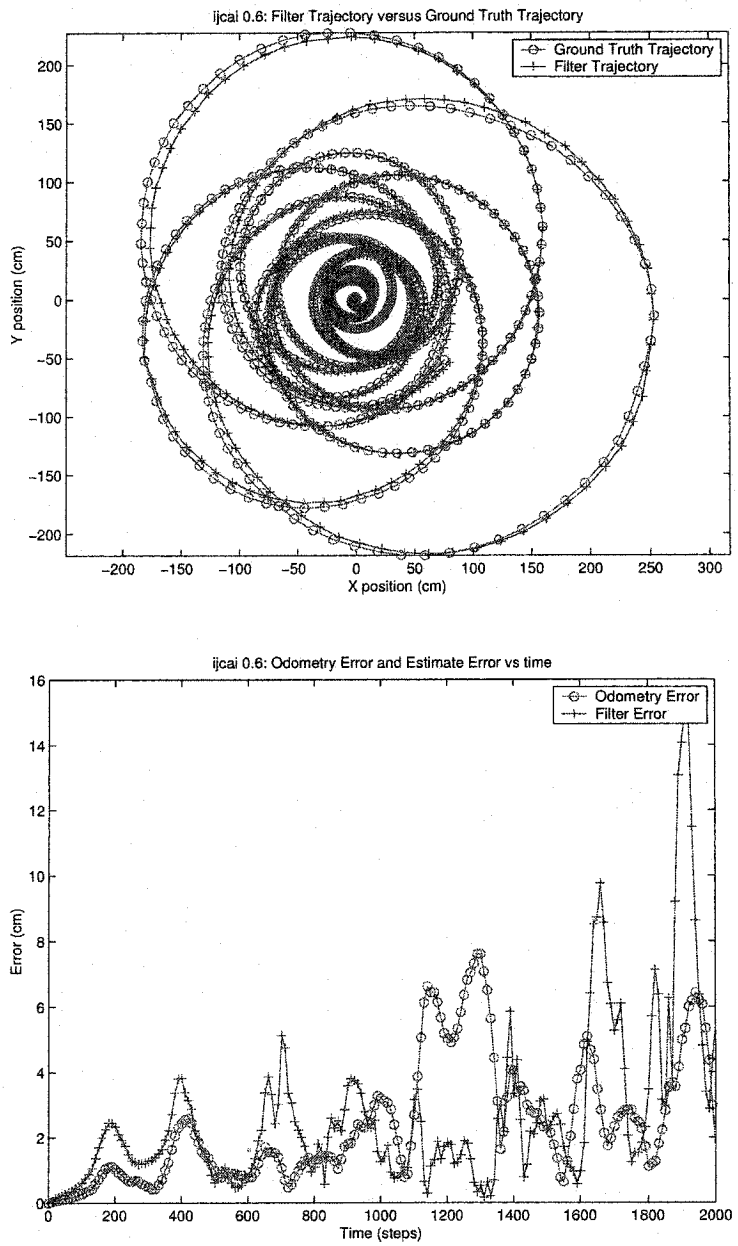


FIGURE 6.11. Top: Filter (+) vs ground truth (o) trajectories for $n = 0.6$. Bottom: Filter error (+) and odometry error (o) versus time for $n = 0.6$.

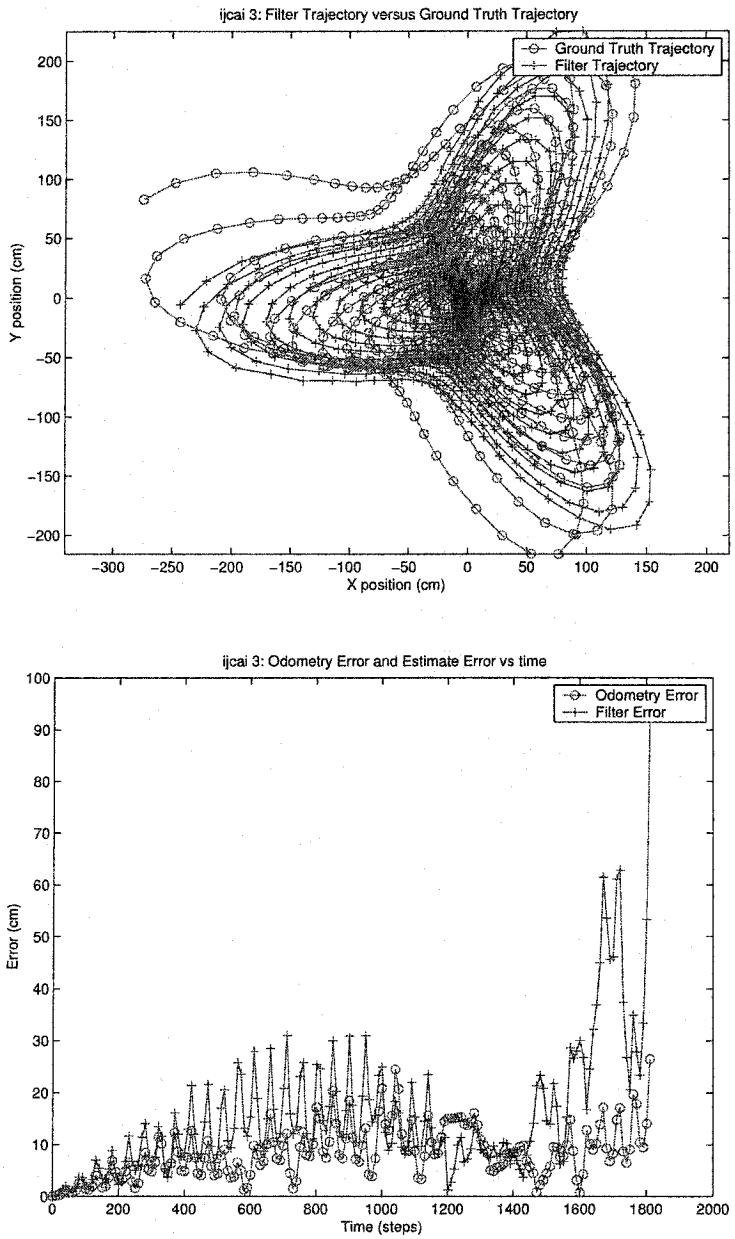


FIGURE 6.12. Top: Filter (+) vs ground truth (o) trajectories for $n = 3.0$. Bottom: Filter error (+) and odometry error (o) versus time for $n = 3.0$.

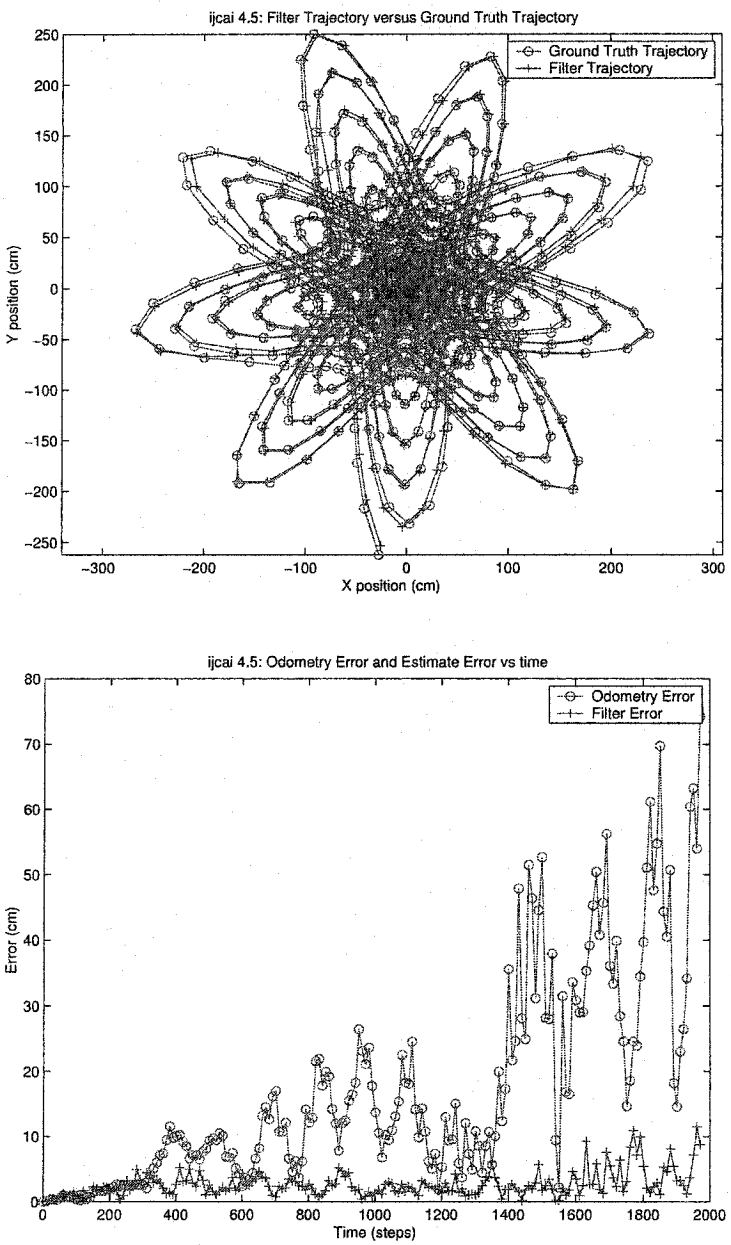
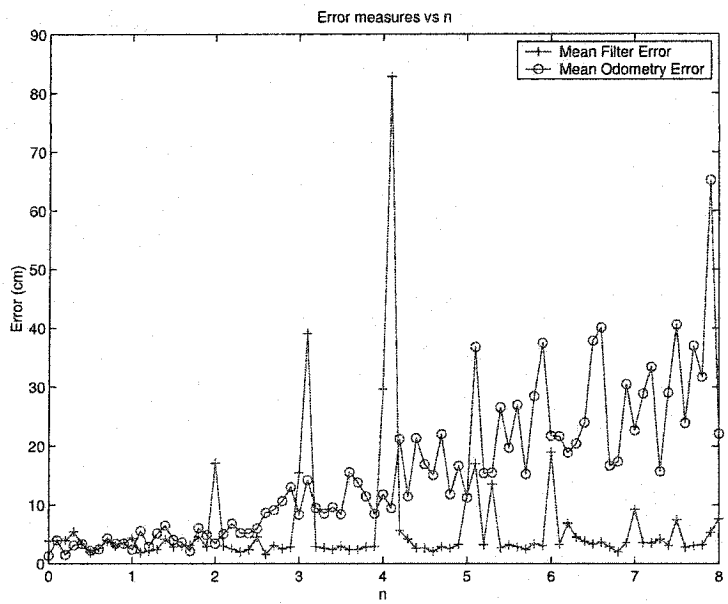
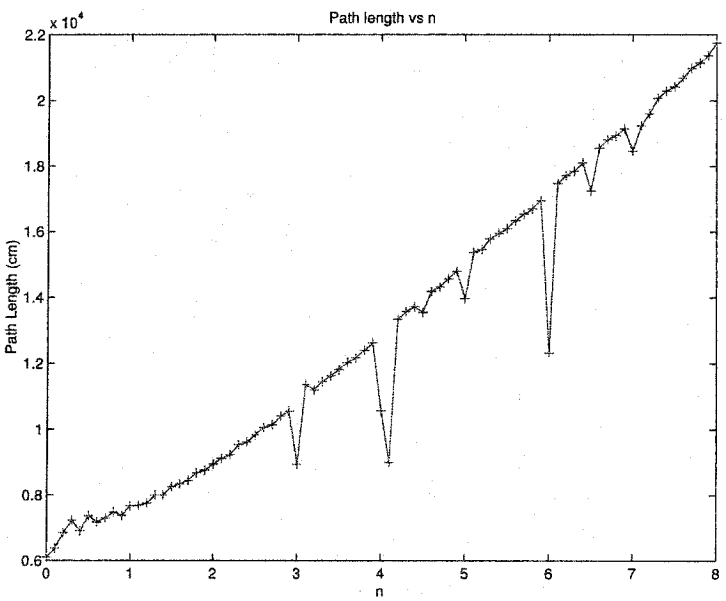


FIGURE 6.13. Top: Filter (+) vs ground truth (o) trajectories for $n = 4.5$. Bottom: Filter error (+) and odometry error (o) versus time for $n = 4.5$.



(a) Mapping accuracy.



(b) Trajectory length vs. n.

FIGURE 6.14. a) Mapping accuracy: Mean filter error and mean odometry error vs n. b) Trajectory length versus n.

5. Discussion

This chapter has experimentally examined the problem of automatically constructing a visual map of the environment, with particular attention paid to selecting an exploration policy that balances accuracy with efficiency. The simulation results on the heuristic set of policies indicate that this particular balance is difficult to strike, as the best way to improve accuracy is to select a highly inefficient method. We also examined results using a real robot and, while the lack of ground truth presents a difficulty in evaluating the results, the homing strategy indicated that the resulting map was useful for navigation and far more accurate than odometry-based navigation.

The outcome of the experiments suggest that some balance can be struck between maintaining map accuracy and exploration efficiency. The parametric family under consideration is in general a suitable choice for exploration in that for most parameterizations the error in the generated map is small relative to odometric error. However, it was interesting to note a subset of parameterizations that systematically led to divergence. A somewhat surprising result was that mapping error was relatively small for the simple spiral trajectory corresponding to $n = 0$. While this can be explained in part by the small amount of accumulated odometric error, it runs contrary to the findings from the `Concentric` policy. Nonetheless, a working hypothesis is that if the exploration were to continue over a longer time interval, the filter will eventually diverge. Furthermore, the potential for the presence of obstacles in a real environment would impose circumnavigation requirements which could introduce the kinds of odometric errors that might lead to divergence. As such, the conclusions are that it is worth the additional effort of "re-homing" the robot from time to time, corresponding to employing a larger value of n.

An important aspect of the exploratory trajectories considered in this work is that they are computed independently of the state of the robot's map or the uncertainty in the filter. An obvious direction for future investigation is to determine a locally optimal trajectory, given the current map and filter state. Furthermore, while it is beyond the scope of this thesis, the family of curves studied pose the problem that the

robot needs to determine a central starting position at the outset, not to mention that the rotational symmetry of the curves make them less suitable for irregularly shaped environments. One potential solution to this problem is to partition the environment and build a separate visual map for each partition. These issues are left open for future development.

CHAPTER 7

Conclusion

This thesis has considered the problem of automatically constructing a visual representation of an unknown environment. The central idea is to learn the generating function that maps camera pose to image features, enabling a robot to develop its "mind's eye". The task of choosing which features to represent was decided by first applying a visual attention mechanism, followed by a *post hoc* evaluation of the system's ability to represent the features that were extracted and tracked. The generating functions were learned using a variety of interpolation mechanisms. Considerable attention was paid to the problem of feature evaluation and several applications of the representation to inference problems in robotics were presented. Some of the key features of the visual map representation include sensor independence, so that the framework is equally applicable to a variety of camera geometries, and environment independence, in that the features do not depend on assumptions that the observations derive from three-dimensional points but can model arbitrary repeated visual phenomena. Experimentally, the feature learning framework demonstrated remarkable precision in estimating robot pose, and significant robustness against scene occlusions. While the framework imposes certain requirements for pose-space coverage and dimensionality, there are a variety of scenarios where these requirements are naturally defined by the task. For example, coverage is a necessary requirement in automated inspection applications.

Where the visual map representation calls for ground truth pose information provided with the training images, we examined a variety of scenarios in which a visual map can be constructed with only limited, or highly uncertain ground truth information. In Chapter 4, an embedding of the image domain in pose space was computed from an image ensemble, and in Chapters 5 and 6 we considered the problem of inferring an accurate map by controlling the uncertainty of the robot's pose as it collects training images. The experimental results demonstrated that real-time on-line map construction is possible and that, if the exploration trajectory is chosen wisely, the resulting map is highly accurate. These results have obvious implications beyond the visual mapping paradigm as mapping with any sensor model involves dealing with the issue of uncertainty management.

1. Future Work

While a broad variety of problems have been covered with respect to the automatic construction of visual maps, there remain several open problems and unanswered questions. These are divided primarily between the problem of representation and the problem of data collection.

With respect to the visual map representation, an obvious extension is to model other feature attributes. In particular, feature shape, scale and orientation can play an informative role in pose inference and reduce residuals in the appearance model. These attributes can be readily incorporated into the existing framework, although they do pose issues with respect to computational cost, particularly in dynamically generating appearance templates for feature tracking.

A key issue with respect to the feature modelling framework is the requirement for complete coverage of the pose space, and the subsequent limitation imposed on the dimensionality of the pose space. While this is a legitimate concern, relaxing these constraints will inevitably require new assumptions concerning the behaviour of the features being modelled. For example, if we assume that the modelled features correspond explicitly to three-dimensional points in space, we can construct a compact

model that requires a small number of observations to model its image-domain positional behaviour (modelling appearance, however, remains difficult). One reasonable approach to this trade-off between assumptions is to incorporate a model evaluation scheme that selects an appropriate model based on the evidence supporting its accuracy. In this way, a compact representation can be applied to three-dimensional points and more general representations can be applied to other phenomena.

Probably the most difficult task for any robotics application is that of unambiguous data association. While it is not reflected in the text, a significant portion of the research and development effort that went in to this work involved developing tracking and matching methods that were robust. One important advantage of the visual mapping framework is that when matching produces outliers, the cross-validation and *a posteriori* distribution estimation mechanisms provide a significant degree of robustness. That being said, robust, unambiguous data association will probably continue to be an important research direction, not only in the visual domain but in other sensing domains as well. The sampling technique based on visual attention presented in this thesis is an important step forward in leveraging low-level vision operators towards producing higher level visual cognition.

An open question with respect to the visual mapping framework is that of representing very large environments with thousands of features. Additional mechanisms beyond the feature evaluation methods presented here might be required to select features based on how informative they are globally about the environment, as well as spatial priors indicating which trajectories and regions of the environment are of interest. Such a mechanism would enable the system to select a minimal set of features that cover the explored space, as well as tune the explored space to meet the robot's task-determined objectives.

With respect to the feature representation and its relationship to visual attention and other feature detectors, an interesting avenue for study would be the development of an attention operator that can learn to extract the most informative localized visual features based on training from an ensemble of images. One would require that, once

trained, such an operator would be able to rapidly extract highly stable features of the scene.

With respect to autonomous exploration, this thesis has only scratched the surface of a problem that is only beginning to draw attention in the robotics community. The general trend in robotic exploration is to apply heuristics to the data collection problem, and even my own work begins with a set of assumptions concerning the kinds of exploratory trajectories that are suitable for constructing an accurate map. Among the unanswered questions are those of discriminating between errors due to actuator and odometer noise versus modelling and sensing noise, determining a general approach for data-driven exploration that accounts for both kinds of noise, and finally determining a framework that explicitly accounts for variation in model choice. As current technology in concurrent mapping and localization matures, these questions will play an increasingly dominant role in the deployment of autonomous robotic systems.

2. Final Thoughts

Visualization is a key component of human navigation and inference systems. The visual mapping framework takes steps toward mimicking the mind's eye phenomenon that enables humans to visualize objects and places based on assumptions about their invariance. All of this is made possible by taking a generative approach to representing the world. As computational resources increase, we can expect the expressiveness of these models to increase similarly. There is little doubt that these systems will have sufficiently rich internal representations so as to enable robotic agents to outperform their human counterparts, even at generic tasks in arbitrary environments. The visual mapping framework represents a solid foundation for the development of these representations, and as the enumeration of future directions suggests, there is enormous promise for its future development.

Visual maps represent a cornerstone for robotic visual cognition. The development of the mapping framework allows robots to enter new, unknown environments

and acquire a visual representation in real-time. This approach opens new avenues to robotic exploration, interaction with humans, and even robotic creativity. Whether it is operating in the domain of space exploration or in a busy warehouse, the visual mapping framework provides the flexibility to enable a new generation of robots to perform their tasks with reliability and precision.

APPENDIX A

The Visual Mapping Software Architecture

The visual mapping architecture consists of several programs that implement various components of the framework. The first four components are highly parallelized and can be run on a networked cluster of PC's. The main programs are as follows.

1. Visual Maps

1.1. landmark.

Synopsis: Extracts features from input images. Default mode is to compute edge-density based features. Also computes KLT [75], and interfaces with several other operators.

Input: a set of portable graymap images (PGM).

Output: `foo.clist` A file containing the list of features extracted from each image.

1.2. batch_track.

Synopsis: Tracks features across an image ensemble.

Input: `foo.clist`, containing a list of extracted candidate features.

Output: `foo.gml`, containing a list of feature models.

1.3. compute_synthcert.

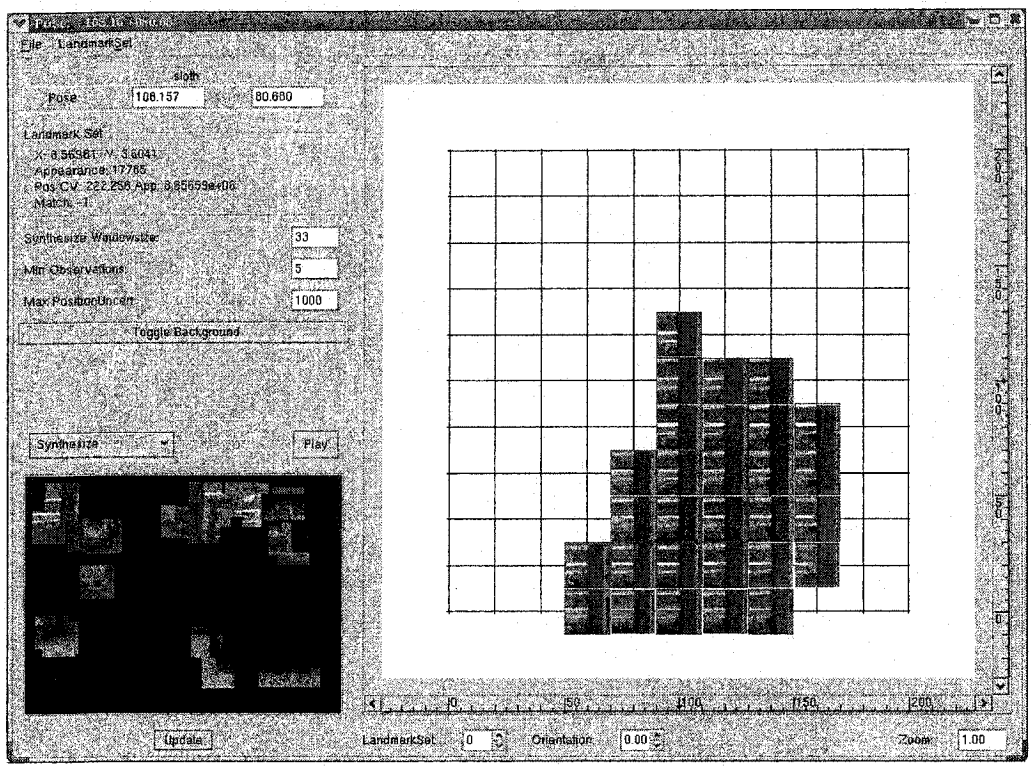


FIGURE A.1. The `rdgui` user interface.

Synopsis: Evaluates a set of feature models and removes unreliable models.

Input: `foo.gml`, containing a list of feature models.

Output: `foo.gml`, containing a filtered list of feature models.

1.4. cluster_loc.

Synopsis: Computes pose estimates given a set of images and a set of feature models.

Input: `img_i.pgm`, A set of PGM images for which a pose distribution is desired, and `foo.gml`, containing a list of feature models.

Output: `img_i.prob` containing a discretized probability distribution of where the image was observed from and `img_i.res`, containing the maximum-likelihood estimate.

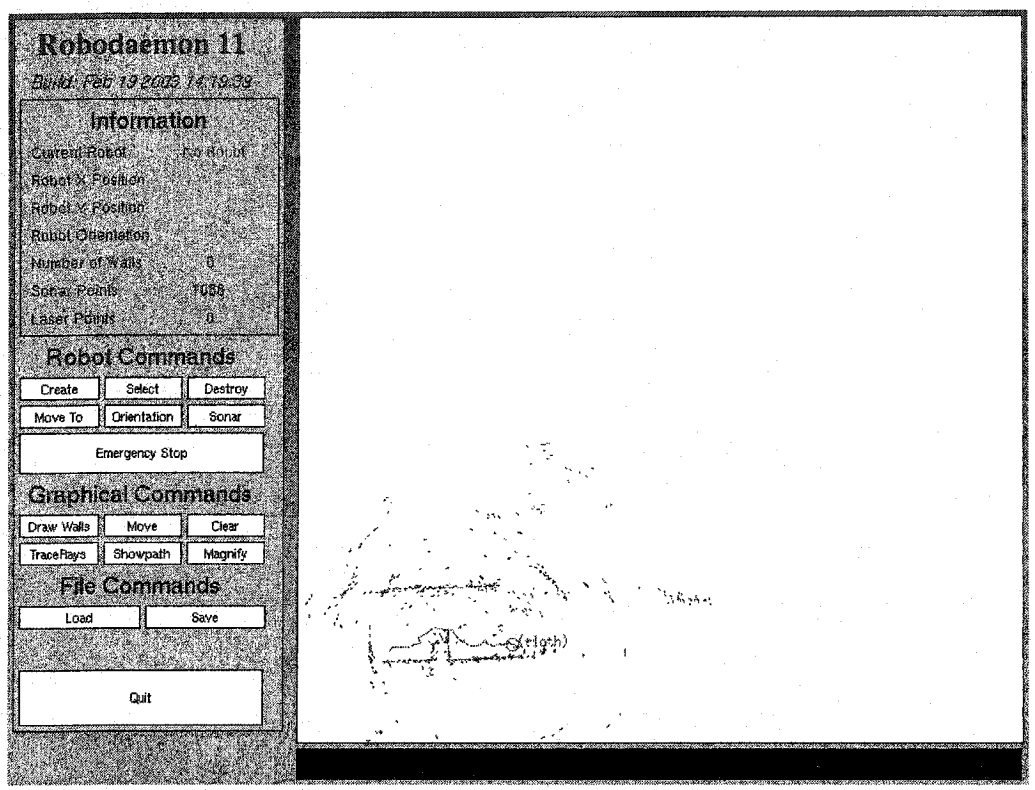


FIGURE A.2. The `Robodaemon` user interface.

2. Mapping and Exploration

In addition to the feature learning framework, there are a number of programs that implement specific applications:

2.1. selforg_track.

Synopsis: Similar to batch_track, performs tracking to enable self-organization.

Input: `foo.clist`, a list of candidate features.

Output: `foo.gml`, a set of tracked features for self-organization.

2.2. selforg.

Synopsis: Self-organizing map inference.

Input: `foo.gml`, a list tracked features, and a set of prior pose constraints.

Output: `foo.gml`, a set of feature models and inferred camera poses.

2.3. loopclose.

Synopsis: Similarity-based loop closure for order image sequences and weak odometric information.

Input: `foo.gml`, set of features tracked over an image sequence, and `actions.dat`, the action sequence associated with the images. constraints.

Output: `foo.gml`, a set of feature models and inferred camera poses.

2.4. kfslam.

Synopsis: Online simultaneous localization and mapping using a Kalman Filter.

Input: `foo.conf`, a configuration file describing the exploration policy, robot type, and feature learning parameters.

Output: `foo.gml`, a set of feature models and camera poses.

2.5. rdgui.

Synopsis: A graphical user interface for visualizing the learned feature models (Figure A.1). Applications such as scene reconstruction are implemented through this program.

Input: `foo.gml`, a set of feature models.

2.6. Robodaemon.

Synopsis: Robot control and simulation software for controlling the real and simulated robots used throughout the experiments [**30**]. Figure A.2 illustrates the Robdaemon interface.

2.7. glenv.

Synopsis: An application for visually simulating Robodaemon environments. This application was used to generate the simulated camera viewpoints in the exploration experiments.

REFERENCES

[1] T. Arbel and F. P. Ferrie, *Viewpoint selection by navigation through entropy maps*, Seventh IEEE International Conference on Computer Vision (Kerkyra, Greece), IEEE Press, 1999, pp. 248–254.

[2] D. Avis and H. Imai, *Locating a robot with angle measurements*, Journal of Symbolic Computation (1990), no. 10, 311–326.

[3] D. Ballard, *Eye movements and visual cognition*, Workshop on Spatial Reasoning and Multisensor Fusion, Morgan Kaufmann, 1987, pp. 188–200.

[4] P. Belhumeur and D. Kriegman, *What is the set of images of an object under all possible lighting conditions?*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recogition (CVPR) (San Francisco, CA), IEEE Press, 1996, pp. 270–277.

[5] G. Beni and J. Wang, *Theoretical problems for the realization of distributed robotic system*, Proceedings of the IEEE International Conference on Robotics and Automation (Sacramento, CA), IEEE Press, April 1991, pp. 1914–1920.

[6] M. Betke and L. Gurvits, *Mobile robot localization using landmarks*, IEEE Transactions on Robotics and Automation **13** (1997), no. 2, 251–263.

[7] H. Blaasvaer, P. Pirjanian, and H. I. Christensen, *AMOR: An autonomous mobile robot navigation system*, IEEE International Conference on Systems, Man, and Cybernetics (San Antonio, TX), October 1994, pp. 2266–2271.

[8] E. Bourque and G. Dudek, *Automated image-based mapping*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop on Perception for Mobile Agents (Santa Barbara, CA), June 1998, pp. 61–70.

[9] J. M. Brady, S. A. Cameron, H. Durrant-Whyte, M. M. Fleck, D. A. Forsyth, J. A. Noble, and I. Page, *Progress towards a system that can acquire pallets and clean warehouses.*, 4th International Symposium on Robotics Research, MIT Press, Cambridge MA, 1987.

[10] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun, *Collaborative multi-robot exploration*, Proceedings of the IEEE International Conference on Robotics and Automation (San Francisco, CA), IEEE Press, May 2000, pp. 476 –481.

[11] K. S. Chong and L. Kleeman, *Accurate odometry and error modelling for a mobile robot*, Proceedings of the IEEE International Conference on Robotics and Automation (Albequerque, NM), April 1997, pp. 2783–2788.

[12] H. Choset and K. Nagatani, *Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization*, IEEE Transactions on Robotics and Automation **17** (2001), no. 2, 125–137.

[13] CGAL Editorial Committee, *Computational geometry algorithms library, version 2.4*, http://www.cgal.org, May 2003.

[14] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to algorithms*, The MIT Press, 1990.

[15] M. Cottrell, J. C. Fort, and G. Pagès, *Two or three things that we know about the Kohonen algorithm*, Proceedings of the European Symposium on Artificial Neural Networks (Brussels, Belgium) (M. Verleysen, ed.), D facto conference services, 1994, pp. 235–244.

[16] J.L. Crowley, F. Wallner, and B. Schiele, *Position estimation using principal components of range data*, Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium), May 1998, pp. 3121–3128.

[17] H. Dahmen, M. O. Franz, and H. G. Krapp, *Motion vision - computational, neural, and ecological constraints*, ch. Extracting Egomotion from Optic Flow: Limits of Accuracy and Neural Matched Filters, pp. 143–168, Springer Verlag, Berlin, 2001.

[18] A. Davison, *Real-time simultaneous localisation and mapping with a single camera*, Proceedings of the IEEE International Conference on Computer Vision (Nice, France), 2003.

[19] A. J. Davison and N. Kita, *3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Lihue, HI), December 2001, pp. 384–391.

[20] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, *Using the CONDENSATION algorithm for robust, vision-based mobile robot localization*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Ft. Collins, CO), IEEE Press, June 1999, pp. 2588–2593.

[21] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, *Monte Carlo localization for mobile robots*, IEEE International Conference on Robotics and Automation (ICRA) (Detroit, MI), May 1999, pp. 1322–1328.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm (with discussion)*, Journal of the Royal Statistical Society series B **39** (1977), 1–38.

[23] J.L. Deneubourg, S. Goss, J. Pasteels, D. Fresneau, and J.P. Lachaud, *Self-organization mechanisms in ant societies (ii): Learning in foraging and division of labor*, From Individual to Collective Behavior in Social Insects, Experientia Supplementum (J.M. Pasteels and J.L. Deneubourg, eds.), vol. 54, Birkhauser Verlag, 1989, pp. 177–196.

[24] X. Deng and A. Mirzaian, *Competitive robot mapping with homogeneous markers*, IEEE Transactions on Robotics and Automation **12** (1996), no. 4, 532–542.

[25] O. Devillers, S. Meiser, and M. Teillaud, *Fully dynamic Delaunay triangulation in logarithmic expected time per operation*, Computational Geometry: Theory and Applications **2** (1992), no. 2, 55–80.

[26] Richard O. Duda and Peter E. Hart, *Pattern classification and scene analysis*, John Wiley & Sons, Inc., 1973.

[27] G. Dudek and M. Jenkin, *Computational principles of mobile robotics*, Cambridge University Press, May 2000, ISBN: 0521568765.

[28] G. Dudek and D. Jugessur, *Robust place recognition using local appearance based methods*, International Conference on Robotics and Automation (San Francisco), IEEE Press, April 2000, pp. 466–474.

[29] G. Dudek, K. Romanik, and S. Whitesides, *Localizing a robot with minimum travel*, Proceedings of the Sixth ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA), January 1995, pp. 437–446.

[30] G. Dudek and R. Sim, *Robodaemon - a device independent, network-oriented, modular mobile robot controller*, Proceedings of the IEEE International Conference on Robotics and Automation (Taipei, Taiwan), IEEE Press, May 2003, p. 7.

[31] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes, *Robotic exploration as graph construction*, Transactions on Robotics and Automation **7** (1991), no. 6, 859–865.

[32] Gregory Dudek and Chi Zhang, *Vision-based robot localization without explicit object models*, Proc. International Conference of Robotics and Automation (Minneapolis, MN), IEEE Press, 1996.

[33] Gregory L. Dudek, *Environment representation using multiple abstraction levels*, Proceedings of the IEEE **84** (1996), no. 11, 1684–1704.

[34] A. A. Efros and T. K. Leung, *Texture synthesis by non-parametric sampling*, IEEE International Conference on Computer Vision (1999), 1033–1038.

[35] J. W. Fenwick, P. M. Newman, and J. J. Leonard, *Cooperative concurrent mapping and localization*, IEEE International Conference on Robotics and Automation (Washington, D.C.), vol. 2, May 2002, pp. 1810–1817.

[36] D. Fox, W. Burgard, and S. Thrun, *Active Markov localization for mobile robots*, Robotics and Autonomous Systems (RAS) **25** (1998), 195–207.

[37] S. Geman and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **6** (1984), no. 6, 721–741.

[38] J. Gielis, *A generic geometric transformation that unifies a wide range of natural and abstract shapes*, American Journal of Botany **90** (2003), 333–338.

[39] G. H. Golub and C. F. Van Loan, *Matrix computation*, John Hopkins University Press, 1991, Second Edition.

[40] H. Gross, V. Stephan, and H. Bohme, *Sensory-based robot navigation using self-organizing networks and q-learning*, Proceedings of the World Congress on Neural Networks, Lawrence Erlbaum Associates, Inc., September 1996, pp. 94–99.

[41] J. Guivant, E. Nebot, and H. Durrant-Whyte, *Simultaneous localization and map building using natural features in outdoor environments*, Sixth International Conference on Intelligent Autonomous Systems (Italy), vol. 1, IOS Press, 2000, pp. 581–588.

[42] J.S. Gutmann and K. Konolige, *Incremental mapping of large cyclic environments*, International Symposium on Computational Intelligence in Robotics and Automation (CIRA'00), 2000.

[43] C.G. Harris and M.J. Stephens, *A combined corner and edge detector*, Proceedings of the Fourth Alvey Vision Conference (Manchester), 1988, pp. 147–151.

[44] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.*, Cambridge University Press, Cambridge, United Kingdom, 2000.

[45] S. Haykin, *Neural networks*, MacMillan College Publishing Company, New York, NY, 1994.

[46] E. Hildreth, *The measurement of visual motion*, MIT Press, Cambridge, MA, 1984.

[47] A.E. Hoerl and R.W. Kennard, *Ridge regression: Biased estimation for nonorthogonal problems*, Technometrics **12** (1970), no. 3, 55–67.

[48] B. Hoffmann-Wellenhof, H. Lichtenegger, and J. Collins., *GPS: Theory and practice*, 3rd ed., Springer-Verlag, New York, 1994.

[49] B. K. P. Horn, *Non-correlation methods for stereo matching*, Photogrammetric Engineering and Remote Sensing **49** (1983), no. 5, 535–536.

[50] D. R. Hougen and N. Ahuja, *Estimation of the light source distribution and its use in integrated shape recovery from stereo and shading*, Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Berlin, Germany), 1993, pp. 148–155.

[51] Katsushi Ikeuchi, *Determining surface orientations of specular surfaces by using the photometric stereo method*, IEEE Transactions on Pattern Analysis and Machine Intelligence **3** (81), no. 6, 661–669.

[52] Luca Iocchi and Daniele Nardi, *Self-localization in the robocup environment*, Proceedings of 3rd RoboCup Workshop, Springer-Verlag, 1999.

[53] M. Isard and A. Blake, *Condensation–conditional density propagation for visual tracking*, International Journal of Computer Vision **29** (1998), no. 1, 2–28.

[54] L. Itti, C. Koch, and E. Niebur, *A model of saliency-based visual attention for rapid scene analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1998), 1254–1259.

[55] ———, *A model of saliency-based visual attention for rapid scene analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998), no. 11, 1254–1259.

[56] M. R. M. Jenkin and P. A Kolers, *Some problems with correspondence*, Tech. Report RBCV-TR-86-10, Researches in Biological and Computational Vision, Department of Computer Science, University of Toronto, 1986.

[57] M. Kelly, *Annular symmetry operators: A multiscalar framework for locating and describing imaged objects*, Ph.D. thesis, McGill University, Montréal, Québec, 1995.

[58] C. Koch and S. Ullman, *Shifts in selective visual attention: towards the underlying neural circuitry*, Human Neurobiology **4** (1985), 219–227.

[59] R. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*, Proceedings of the 1995 International Joint Conference on Artificial Intelligence (IJCAI) (Montréal), August 1995, pp. 1137–1145.

[60] T. Kohonen, *Self-organization and associative memory*, Springer-Verlag, New York, 1984.

[61] ———, *Self-organizing maps*, Springer, Berlin; Heidelberg; New-York, 1995.

[62] D. J. Kriegman, E. Triendl, and T. O. Binford, *Stereo vision and navigation in buildings for mobile robots*, IEEE Transactions on Robotics and Automation **5** (1989), no. 6, 792–803.

[63] E. Krotkov, *Mobile robot localization using a single image*, Proceedings of the IEEE International Conference on Robotics and Automation (Scottsdale, AZ), 1989, pp. 978–983.

[64] B. Kuipers and P. Beeson, *Bootstrap learning for place recognition*, Proceedings of the National Conference on Artificial Intelligence (AAAI) (Edmonton, AB), 2002, pp. 174–180.

[65] B. Kuipers and Y.-T. Byun, *A robot exploration and mapping strategy based on a semantic hierachy of spatial representations*, Robotics and Autonomous Systems **8** (1991), 46–63.

[66] B. J. Kuipers, *The Spatial Semantic Hierarchy*, Artificial Intelligence **119** (2000), 191–233, http://www.cs.utexas.edu/users/qr/papers/Kuipers-aij-00.html.

[67] B. J. Kuipers and Y. T. Byun, *A qualitative approach to robot exploration and map-learning*, Proceedings of the IEEE workshop on spatial reasoning and multi-sensor fusion (Los Altos, CA), IEEE, 1987, pp. 390–404.

[68] J. J. Leonard and H. F. Durrant-Whyte, *Simultaneous map building and localization for an autonomous mobile robot*, Proceedings of the IEEE Int. Workshop on Intelligent Robots and Systems (Osaka, Japan), November 1991, pp. 1442–1447.

[69] J. J. Leonard and H. J. S. Feder, *A computationally efficient method for large-scale concurrent mapping and localization*, Robotics Research: The Ninth International Symposium (London) (J. Hollerbach and D. Koditschek, eds.), Springer-Verlag, 2000, pp. 169–176.

[70] John J. Leonard and Hugh F. Durrant-Whyte, *Mobile robot localization by tracking geometric beacons*, IEEE Transactions on Robotics and Automation **7** (1991), no. 3, 376–382.

[71] C. Lin and R. Tummala, *Mobile robot navigation using artificial landmarks*, Journal of Robotic Systems **14** (1997), no. 2, 93–106.

[72] J.C. Longuet-Higgins, *A computer algorithm for reconstructing a scene from two projections*, Nature **293** (1981), 133–135.

[73] D. G. Lowe, *Object recognition from local scale-invariant features*, Proceedings of the International Conference on Computer Vision (Corfu, Greece), IEEE Press, September 1999, pp. 1150–1157.

[74] F. Lu and E. E. Milios, *Robot pose estimation in unknown environments by matching 2D range scans*, Proceedings of the Conference on Computer Vision and Pattern Recognition (Los Alamitos, CA, USA), IEEE Computer Society Press, June 1994, pp. 935–938.

[75] B.D. Lucas and T. Kanade, *An iterative image registration technique with an application to stereo vision*, IJCAI81, 1981, pp. 674–679.

[76] S. Lumelsky, S. Mukhopadhyay, and K. Sun, *Dynamic path planning in sensor-based terrain acquisition*, IEEE Transactions on Robotics and Automation **6** (1990), no. 4, 462–472.

[77] D. MacKay, *Information-based objective functions for active data selection*, Neural Computation **4** (1992), no. 4, 590–604.

[78] P. MacKenzie and G. Dudek, *Precise positioning using model-based maps*, Proceedings of the International Conference on Robotics and Automation (San Diego, CA), IEEE Press, 1994, pp. 1615–1621.

[79] T.P. Minka, *The 'summation hack' as an outlier model*, http://www.stat.cmu.edu/~minka/papers/minka-summation.pdf, 2003.

[80] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, *FastSLAM: A factored solution to the simultaneous localization and mapping problem*, Proceedings of the AAAI National Conference on Artificial Intelligence (Edmonton, Canada), AAAI, 2002.

[81] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, *Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges*, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03) (San Francisco, CA) (Georg Gottlob and Toby Walsh, eds.), Morgan Kaufmann Publishers, 2003.

[82] S. Moorehead, R. Simmons, and W. R. L. Whittaker, *Autonomous exploration using multiple sources of information*, IEEE International Conference on Robotics and Automation (Seoul, South Korea), May 2001.

[83] H. Moravec, *Sensor fusion in certainty grids for mobile robots*, AI Magazine **9** (1988), no. 2, 61–74.

[84] S.K. Nayar, H. Murase, and S.A. Nene, *Learning, positioning, and tracking visual appearance*, Proc. IEEE Conf on Robotics and Automation (San Diego, CA), May 1994, pp. 3237–3246.

[85] S. Nene and S. K. Nayar, *Stereo using mirrors*, Proceedings of the IEEE International Conference on Computer Vision (Bombay), IEEE Press, January 1998, pp. 1087–1094.

[86] D. Noton and L. Stark, *Eye movements and visual perception*, Scientific American **224** (1971), no. 6, 33–43.

[87] S. I. Olson, *Stereo correspondence by surface reconstruction*, IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990), no. 3, 309–315.

[88] J.K. O'Regan, H. Deubel, J.J. Clark, and R.A. Rensink, *Picture changes during blinks: looking without seeing and seeing without looking*, Visual Cognition **7** (2000), no. 1, 191–212.

[89] M.A. Paskin, *Thin junction tree filters for simultaneous localization and mapping*, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03) (San Francisco, CA) (Georg Gottlob and Toby Walsh, eds.), Morgan Kaufmann Publishers, 2003, pp. 1157–1164.

[90] R. R. Picard and R. D. Cook, *Cross-validation of regression models*, Journal of the American Statistical Association **79** (1984), no. 387, 575–583.

[91] C. J. Poelman and T. Kanade, *A paraperspective factorization for shape and motion recovery.*, IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997), no. 3, 206–218.

[92] G. Poggio and T. Poggio, *The analysis of stereopsis*, Annual Review Neuroscience **7** (1984), 379–412.

[93] T. Poggio and S. Edelman, *A network that learns to recognize 3d objects*, Nature **343** (1990), 263–266.

[94] Tomaso Porcacchi, *Mondo nuovo*, http://www.raremaps.com/cgi-bin/mapbuilder.cgi?America+North-America+3138, 1572.

[95] F. Pourraz and J. L. Crowley, *Continuity properties of the appearance manifold for mobile robot position estimation*, Proceedings of the IEEE Computer Society Conference on Pattern Recognition Workshop on Perception for Mobile Agents (Ft. Collins, CO), IEEE Press, June 1999.

[96] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical recipes in c*, Cambridge University Press, New York, N.Y., 1991.

[97] D. Reisfeld, H. Wolfson, and Y. Yeshurun, *Context free attentional operators: The generalized symmetry transform*, International Journal of Computer Vision **14** (1995), 119–130, Special Edition on Purposive Vision.

[98] Daniel Reisfeld, Haim Wolfson, and Yehezkel Yeshurun, *Context free attentional operators: the generalized symmetry transform*, International Journal Of Computer Vision **14** (1995), 119–130.

[99] I. Rekleitis, G. Dudek, and E. Milios, *Odometric properties of mobile robots using probabilistic analysis*, Submission to IROS 2001. Also available at http://www.cim.mcgill.ca/ yiannis/iros01a.ps.gz, 2001.

[100] I. Rekleitis, R. Sim, G. Dudek, and E. Milios, *Collaborative exploration for map construction*, 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Banff, AB), July 2001, pp. 296–301.

[101] ———, *Collaborative exploration for the construction of visual maps*, 2001 IEEE/RSJ Conference on Intelligent Robots and Systems (IROS) (Hawaii), October 2001, pp. 1269–1274.

[102] I. M. Rekleitis, *Cooperative localization and multi-robot exploration*, Ph.D. thesis, School of Computer Science, McGill University, Montreal, Quebec, Canada, February 2003.

[103] I. M. Rekleitis, G. Dudek, and E. Milios, *Multi-robot collaboration for robust exploration*, Proceedings of the IEEE International Conference on Robotics and Automation (San Francisco, CA), April 2000, pp. 3164–3169.

[104] R.A. Rensink, J.K. O'Regan, and J.J. Clark, *To see or not to see: The need for attention to perceive changes in scenes*, Psychological Science **8** (1997), 368–373.

[105] N. Roy, W. Burgard, D. Fox, and S. Thrun, *Coastal navigation – robot motion with uncertainty*, AAAI Fall Symposium: Planning with POMDPs (Orlando, FL), 1998.

[106] D. Scharstein and A.J. Briggs, *Real-time recognition of self-similar landmarks*, Image and Vision Computing **19** (2000), no. 11, 763–772.

[107] C. Schmid, *A structured probabilistic model for recognition*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Ft. Collins, CO), June 1999, pp. 485–490.

[108] W. Schneider and R. M. Shiffrin, *Controlled and automatic human information processing: I. detection, search, and attention*, Psychological Review **84** (1977), no. 1, 1–66.

[109] S. Se, D. Lowe, and J. Little, *Global localization using distinctive visual features*, Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS) (Lausanne, Switzerland), October 2002, pp. 226–231.

[110] M. Seiz, P. Jensfelt, and H. I. Christensen, *Active exploration for feature based global localization*, Proceedings IEEE International Conference on Intelligent Robots and Systems (IROS) (Takamatshu), October 2000.

[111] O.G. Selfridge, *The organization of organization*, Self-Organizing Systems (M.C. Yovits, G.T. Jacobi, and G.D. Goldstein, eds.), McGregor & Werner, Washington D.C., 1962, pp. 1–7.

[112] H. Shatkay and L. P. Kaelbling, *Learning topological maps with weak local odometric information*, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (Nagoya, Japan), Morgan Kaufmann, 1997, pp. 920–929.

[113] J. Shi and C. Tomasi, *Good features to track*, Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) (Seattle, WA), IEEE Press, 1994, pp. 593 – 600.

[114] A. Shokoufandeh, I. Marsic, and S. Dickinson, *View-based object recognition using saliency maps*, Image and Vision Computing **17** (1999), 445–460.

[115] R. Sim, *Mobile robot localization from learned landmarks*, Master's thesis, McGill University, Montreal, Canada, July 1998.

[116] ———, *Bayesian exploration for mobile robots*, Tech. Report CIM-03-02, Centre for Intelligent Machines, McGill University, Montreal, QC, June 2000.

[117] R. Sim and G. Dudek, *Mobile robot localization from learned landmarks*, Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS) (Victoria, Canada), IEEE Press, October 1998, pp. 1060–1065.

[118] ———, *Learning and evaluating visual features for pose estimation*, Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV) (Kerkyra, Greece), IEEE Press, Sept 1999, pp. 1217–1222.

[119] ———, *Learning environmental features for pose estimation*, Proceedings of the 2nd IEEE Workshop on Perception for Mobile Agents (Ft. Collins, CO), IEEE Press, June 1999, pp. 7–14.

[120] ———, *Learning visual landmarks for pose estimation*, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (Detroit, MI), IEEE Press, May 1999, pp. 1972–1978.

[121] ———, *Visual landmarks for pose estimation*, Canadian Artificial Intelligence (1999), no. 43, 13–17.

[122] ———, *Learning landmarks for robot localization*, Proceedings of the National Conference on Artificial Intelligence SIGART/AAAI Doctoral Consortium (Austin, TX), SIGART/AAAI, AAAI Press, July 2000, pp. 1110–1111.

[123] ———, *Learning environmental features for pose estimation*, Image and Vision Computing, Elsevier Press **19** (2001), no. 11, 733–739.

[124] ———, *Learning generative models of scene features*, IEEE Conference on Computer Vision and Pattern Recognition (Hawaii), IEEE Press, December 2001, pp. 406–412.

[125] ———, *Comparing image-based localization methods*, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI) (Acapulco, Mexico), Morgan Kaufmann, August 2003, pp. 1560–1562.

[126] ———, *Effective exploration strategies for the construction of visual maps*, Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS) (Las Vegas, NV), IEEE Press, October 2003, p. 8.

[127] ———, *Examining exploratory trajectories for minimizing map uncertainty*, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Reasoning with Uncertainty in Robotics (RUR) (Acapulco, Mexico), Morgan Kaufmann, August 2003, pp. 69–76.

[128] ———, *Self-organizing visual maps*, Tech. Report MRL-03-01, Mobile Robotics Lab, McGill University, Montreal, QC, October 2003.

[129] ———, *Learning generative models of scene features*, International Journal of Computer Vision (2004), Accepted for publication, January 2004.

[130] R. Sim, S. Polifroni, and G. Dudek, *Comparing attention operators for learning landmarks*, Tech. Report MRL-03-02, Mobile Robotics Lab, McGill University, Montreal, QC, October 2001.

[131] S. Simhon and G. Dudek, *A global topological map formed by local metric maps*, IEEE/RSJ International Conference on Intelligent Robotic Systems (Victoria, Canada), October 1998.

[132] R. Smith and P. Cheeseman, *On the representation and estimation of spatial uncertainty*, International Journal of Robotics Research **5(4)** (1986), 56–68.

[133] R. Smith, M. Self, and P. Cheeseman, *Estimating uncertain spatial relationships in robotics*, Autonomous Robot Vehicles (I.J. Cox and G. T. Wilfong, eds.), Springer-Verlag, 1990, pp. 167–193.

[134] C. Stachniss and W. Burgard, *Exploring unknown environments with mobile robots using coverage maps*, Proc. of the International Conference on Artificial Intelligence (IJCAI) (Acapulco, Mexico), 2003.

[135] G. P. Stein and A. Shashua, *Model-based brightness constraints: On direct estimation of structure and motion*, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000), no. 9, 992–1015.

[136] K. Sugihara, *Some location problems for robot navigation using a single camera*, Computer Vision, Graphics, and Image Processing **42** (1988), 112–129.

[137] K.T. Sutherland and W.B. Thompson, *Inexact Navigation*, Proceedings of the IEEE International Conference on Robotics and Automation (Atlanta, Georgia), May 1993, pp. 1–7.

[138] T. Takahashi, T. Tanaka, K. Nishida, and T. Kurita, *Self-organization of place cells and reward-based navigation for a mobile robot*, Proceedings of the 8th International Conference on Neural Information Processing, Shanghai (China) (2001), 1164–1169.

[139] A. Tarantola, *Inverse problem theory*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 1986.

[140] C. J. Taylor and D. J. Kriegman, *Exploration strategies for mobile robots*, IEEE International Conference on Robotics and Automation (Atlanta, GA), IEEE Press, May 1993, pp. 248–253.

[141] Charles Thorpe, Romuald Aufrere, Justin David Carlson, David Duggins, Terrence W Fong, Jay Gowdy, John Kozar, Robert MacLachlan, Colin McCabe, Christoph Mertz, Arne Suppe, Chieh-Chih Wang, and Teruko Yata, *Safe robot driving*, Proceedings of the International Conference on Machine Automation (ICMA 2002), September 2002.

[142] S. Thrun, *Finding landmarks for mobile robot navigation*, Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium), May 1998, pp. 958–963.

[143] S. Thrun, D. Fox, and W. Burgard, *A probabilistic approach to concurrent mapping and localization for mobile robots*, Machine Learning **31** (1998), 29–53, also appeared in Autonomous Robots 5, 253–271.

[144] Sebastian Thrun, Dieter Fox, and Wolfram Burgard., *A probabilistic approach to concurrent mapping and localization for mobile robots*, Machine Learning and Autonomous Robots **31,5** (1998), 29–53,253–271.

[145] A. N. Tikhonov and V. Y. Arsenin, *Solutions of ill-posed problems*, V.H. Winston & Sons, John Wiley & Sons, Washington D.C., 1977, Translation editor Fritz John.

[146] Honkela Timo, *Self-organizing maps in natural language processing*, Ph.D. thesis, Department of Computer Science and Engineering, Helsinki University of Technology, 1997.

[147] C. Tomasi and T. Kanade, *Detection and tracking of point features*, Tech. Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[148] A. Triesman, *Perceptual grouping and attention in visual search for features and objects*, Journal of Experimental Psychology: Human Perception and Performance **8** (1982), no. 2, 194–214.

[149] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A. Fitzgibbon, *Vision algorithms: Theory and practice*, Lecture Notes in Computer Science, vol. 1883, ch. Bundle adjustment: A modern synthesis, pp. 298–372, Springer Verlag, 2000.

[150] J. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo, *Modelling visual attention via selective tuning*, Artificial Intelligence **78** (1995), no. 1-2, 507–547.

[151] John K. Tsotsos, *Analysing vision at the complexity level*, Behavioral and Brain Sciences **13** (1990), no. 3, 423–496.

[152] I. Ulrich and I. Nourbakhsh, *Appearance-based place recognition for topological localization*, Proceedings of the IEEE Interantional Conference on Robotics and Automation (San Francisco, CA), IEEE Press, April 2000, pp. 1023–1029.

[153] G. Wahba, *Convergence rates of 'thin plate' smoothing splines when the data are noisy*, Smoothing Techniques for Curve Estimation (1979), 233–245.

[154] P. Whaite and F. P. Ferrie, *Autonomous exploration: Driven by uncertainty*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Los Alamitos, CA, USA), IEEE Computer Society Press, June 1994, pp. 339–346.

[155] S.B. Williams, P. Newman, G. Dissanayake, and H.F. Durrant-Whyte, *Autonomous underwater simultaneous localisation and map building*, IEEE International Conference on Robotics and Automation (San Francisco CA), 2000, pp. 1793–1798.

[156] R. J. Woodham, *Photometric stereo: A reflectance map technique for determining surface orientation from image intensity*, Proceedings of SPIE **155** (1978), 136–143.

[157] B. Yamauchi, A. Schultz, and W. Adams, *Mobile robot exploration and map building with continuous localization*, Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium), IEEE Press, May 1998, pp. 3715–2720.

[158] Guido Zunino and Henrik I Christensen, *Simultaneous mapping and localisation in domestic environments*, Multi-Sensory Fusion and Integration for Intelligent Systems (Baden-Baden, DE) (R. Dillmann, ed.), August 2001, pp. 67–72.

Document Log:

Manuscript Version Final

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-LaTeX — 17 February 2004

ROBERT SIM

CENTRE FOR INTELLIGENT MACHINES, MCGILL UNIVERSITY, 3480 UNIVERSITY ST., MONTRÉAL (QUÉBEC) H3A 2A7, CANADA, *Tel.* : (514) 398-6319

E-mail address: simra@cim.mcgill.ca