Planning, Localization, and Mapping for a Mobile Robot in a Camera Network

David Paul Meger

Master of Science

School of Computer Science

McGill University

Montréal, Quebec

February 2007

A Thesis submitted to McGill University in partial fulfilment of the requirements for the degree of Master of Science

ACKNOWLEDGEMENTS

This thesis could not have been completed without the generosity and support of a large group of people. My supervisor Professor Gregory Dudek has been an amazing source of perspective, insight, and understanding and made the lab a fun place to work. Equally, my co-supervisor Professor Ioannis Rekleitis provided tireless enthusiasm and optimism for research. Despite often being at uncommon times and locations, our meetings never failed to be productive and motivating.

Thanks to those students at McGill's Centre for Intelligent Machines, and particularly the Mobile Robotics Lab who have become great friends and supporters over the last two years. I have gained equal portions of insight into research problems and Montréal culture with their help.

Dimitri Marinakis was extremely generous in providing much of the networking code to run the sensor network. Special thanks to Simon Drouin for translating the Abstract to French.

I thank my family for their calm, unquestioning support and for being there without being asked.

Finally, I would also like to thank the National Science and Engineering Research Council of Canada for having provided me with the scholarship which has made it possible for me to pursue this research.

ABSTRACT

Networks of cameras such as building security systems can be a source of localization information for a mobile robot assuming a map of camera locations as well as calibration information for each camera is available. This thesis describes an **automated** system to acquire such information. A fully automated camera calibration system uses fiducial markers and a mobile robot in order to drastically improve ease-of-use compared to standard techniques. A 6DOF EKF is used for mapping and is validated experimentally over a 50 m hallway environment. Motion planning strategies are considered both in front of a single camera to maximize calibration accuracy and globally between cameras in order to facilitate accurate measurements. For global motion planning, an adaptive exploration strategy based on heuristic search allows compromise between distance traveled and final map uncertainty which provides the system a level of autonomy which could not be obtained with previous techniques.

ABRÉGÉ

Des réseaux de caméras comme on en retrouve dans les systèmes de sécurité d'édifice peuvent être utilisés comme source d'information de localisation pour un robot automoteur si la position et les paramètres de calibration de chaque caméra sont connus. Cette thèse présente un système qui permet d'obtenir cette information automatiquement. Un système automatique de calibration de caméra utilise des marqueurs et un robot automoteur pour améliorer la facilité d'utilisation comparée aux techniques existantes. Une méthode itérative de filtrage non linéraire à 6DDL est utilisée pour cartographier l'environnement. Elle est validée de façon expérimentale dans un réseau de couloirs de 50 m. Différentes stratégies de mouvement sont étudiées: d'abord devant une seule caméra, pour maximiser la précision de la calibration, puis, entre caméras pour améliorer la précision de la cartographie. Pour la planification de mouvement globale, une stratégie d'exploration adaptive basée sur l'algorithme A^* permet de faire un compromis entre la distance parcourue et la précision de la cartographie, ce qui procure au sytème un niveau d'autonomie qui n'avait pu être obtenu avec les techniques existantes.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

Modern technology allows for a vast amount of information about the environment to be collected, measured, and analyzed. For example, temperature, pressure, and moisture conditions across the world are monitored in increasingly fine granularity, visual information is collected for a large percentage of the public spaces occupied by humans and cell phones provide persistent, mobile sources of information for and about their users. "Sensor Networks" is a generic term for the type of systems which collect such information. As the availability of data collected by Sensor Networks increases, so does the opportunity to use this data in applications which benefit humans. Making use of the large volume of real time information that is produced by a Sensor Network is often an overwhelming task, so automated systems are required. The problem studied in this thesis is the automated use of image information from a camera network to aid a mobile robot in navigation.

Networks of cameras are a common type of Sensor Network due to their ability to collect visual information that is immediately salient to a human operator. The images from these camera networks provide a rich source of information about the environment in which they are placed; however, extracting the information for automated processing faces several challenges. Camera networks have not typically been deployed with automated processing in mind. This could mean that neither a careful map of each camera's location nor their properties has been recorded. Also,

the nature of the visual information obtained by cameras makes obtaining geometric position information difficult. Camera images are a 2-D representation of the 3-D world, where depth information has been discarded in the imaging process. Since many of the desired applications for camera network information require at least rough estimates of the position of the events observed in the images, overcoming these two challenges would allow the information obtained by a camera Sensor Network to be much more useful.

Unfortunately, obtaining a map of the camera locations and their internal properties and recovering geometric information from the images is a time consuming process for a human operator using previously existing techniques. Specifically, "camera calibration", which refers to the recovery of the internal camera properties, is a process which typically involves a large amount of operator interaction in the data collection process. Also, placing numerous cameras in a single reference frame can only be accomplished by making measurements of the distances and relative orientations between cameras, which is challenging especially in an indoor environment where there may not be clear line of sight between cameras. So, a more automated system is needed to obtain map and calibration information.

In this thesis, we are specifically interested in using the information from a camera network to localize a mobile robot. Traditionally, this task is accomplished using sensors mounted directly on the robot which readily produce position information, such as laser rangefinders and sonar. In an environment instrumented with a camera network, the problems of mapping and calibration of the network and robot localization and mapping can be interconnected. Measurements of the robot's position

in the camera images can replace the range information from a robot's sensors, and positions of the cameras themselves can be used as the landmarks which comprise the map. Such a system presents several unique challenges including the need to calibrate the cameras as a substep of the mapping task and the fact that the system is only able to make measurements when the robot is within the field of view of a camera, which necessitates intelligent robot motion. This thesis will present one method for solving this problem and demonstrate the feasibility of the approach using the robot platform and network setup shown in Figure 1–1.

1.1 Thesis Outline

Within the larger problem of producing a hybrid robot/camera network localization system, there are several sub-problems which will be addressed in the chapters of this thesis:

Automated Camera Calibration Under the assumption that the cameras in the sensor network have been placed arbitrarily and without record of each camera's properties, the first step in using the images from the cameras is to determine the properties of the cameras. The task of computing camera parameters and obtaining metric measurements is referred to in the literature as *camera calibration* and is well-studied in both photogrammetry and computer vision [12, 34]. Standard techniques for calibration are human operator intensive and not suited for use by an autonomous robot. The typical process begins with a human operator collecting a set of images of an engineered calibration pattern, typically a checkerboard or 3-D object, from numerous viewpoints. Next, calibration points are extracted from each image in a process that involves a

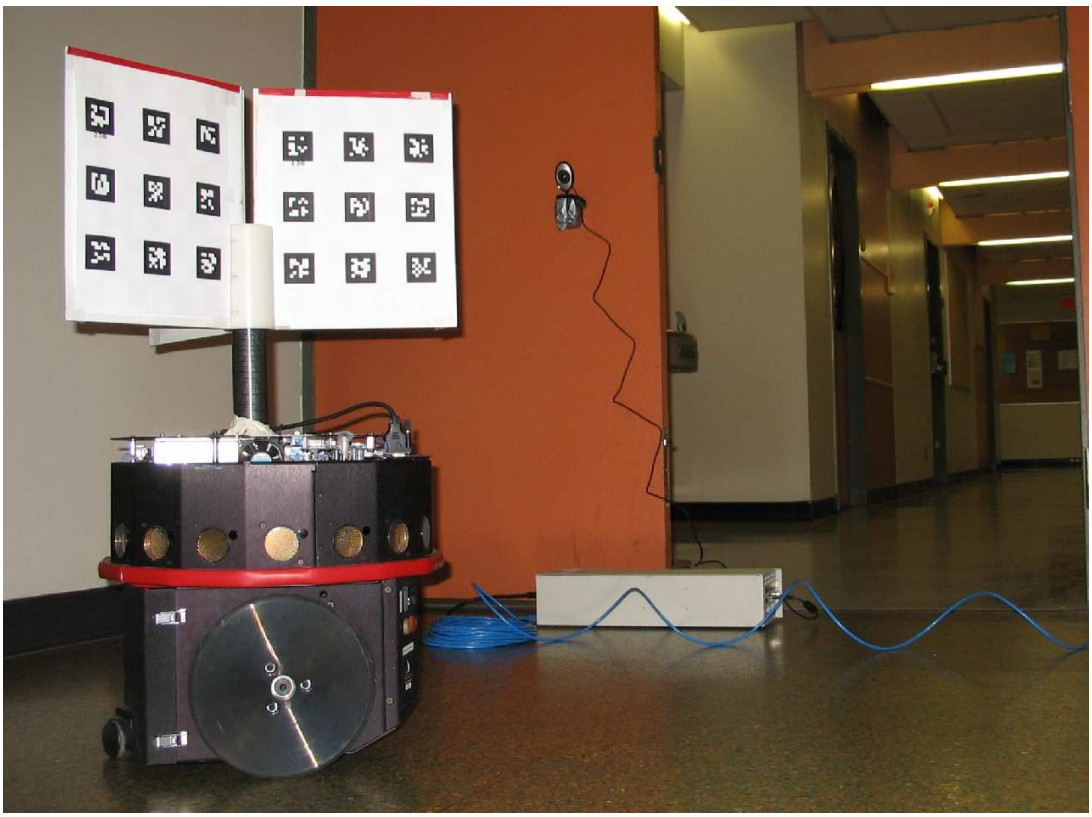


Figure 1–1: The experimental setup used throughout this thesis. The robot carries a calibration target which can be easily detected in images taken by the cameras in the network (such as the one mounted on a door here).

mix of automated detection and manually clicking on image points. Finally, a nonlinear optimization algorithm such as the one proposed in [61] is used to estimate the intrinsic camera parameters as well as the pose from which each image was taken. Section 3.1 will detail an automated solution where the robot replaces the human operator in moving the calibration pattern and collecting images. A system of barcode-like markers (see Figure 2–2) is used along with a detection library [14] so that the calibration points are detected robustly, with high accuracy, and without operator interaction.

Localization and Mapping Framework As with any mapping framework based on noisy relative measurements and a mobile robot without the ability to measure its absolute position, map quantities must be estimated with careful treatment of uncertainty. Section 3.2 will describe the use of measurements from the calibration process to localize the robot and place each camera within a common reference frame. This mapping problem is formulated as an instance of the standard Simultaneous Localization and Mapping (SLAM) problem. Typically, the robot uses its sensors to measure the relative locations of landmarks in the world as it moves. Since the measurements of the robot motion as well as those of the pose of landmarks are imperfect, estimating the true locations becomes a state estimation problem, which is solved using an Extended Kalman filter (EKF). Our situation differs from standard SLAM in that sensor measurements can only be obtained when the robot moves within the field of view of a camera, so robot motion planning is integral to our system's performance.

Motion Planning for Camera Calibration Since the robot replaces a human operator in moving the calibration target during the collection of calibration images, it is necessary for the robot to employ a motion planning scheme to ensure sufficient images can be collected. The calibration literature [61] details several cases where a set of images of a planar target does not provide sufficient information to perform the calibration. The robot must clearly avoid any such situation, but we can hope for more than just this simple guarantee. Section 4 provides analysis of the accuracy of calibration with respect to several different motion strategies and details of the image sets collected.

Exploration Motion Planning During exploration, the order and frequency at which the robot visits the cameras will largely affect mapping accuracy. Two conflicting goals, the desire to cover the space as quickly as possible and to build a final map which is as accurate as possible, lead to a complex problem which is likely intractable to solve optimally. Chapter 5 presents a detailed analysis of several factors of the exploration planning problem, and proposes heuristic exploration strategies which are validated by simulation experiments. Heuristic search using a cost function composed of distance and uncertainty is shown to provide a flexible compromise between the two conflicting goals and is able to exploit information inherent in the EKF estimate in order to achieve favorable results.

1.2 Contributions

The main contribution of this thesis is automated calibration and mapping of a camera Sensor Network by a mobile robot. The calibration and mapping algorithm

consists of the application of well known methods to a new scenario such that the process is fully automated, accurate, and numerically stable. Several novel planning strategies have been developed for both the local calibration and global exploration problems. These strategies allow our system to operate with full autonomy. Beyond that, in several cases, the algorithms used for planning in our context are applicable to other sensing modalities and environments. A list of contributions in this thesis follows:

1. Fully automated camera calibration using fiducial markers and a mobile robot.
2. Adaptation of a known mapping technology for the unique sensing scenario, including derivation of measurement equations.
3. Several heuristic local calibration strategies along with analysis of their effects on calibration accuracy.
4. A study of calibration error as a function of dataset parameters.
5. Implementation of calibration, mapping, and local calibration procedures on experimental hardware which included modules for image processing, communication, robot control, local planning, user interface, and numerous other functions.
6. Several static heuristic trajectories for exploration along with analysis of their effects on exploration and map uncertainty.
7. Adaptation of the heuristic search planning algorithm to minimum distance and uncertainty exploration.

Many of the sub-problems discussed here have been published by the author during the course of development. The corresponding publications [35, 41] include all but 4 and 7 above.

CHAPTER 2
Background

This thesis depends on a relatively broad selection of fields, so the background material will be split into several categories. First, previous work in Sensor Networks, whether composed of cameras or other sensing devices, will be discussed, as several such methods have similar high level goals and hardware. Second, previous work on camera calibration will be provided in sufficient detail to understand its use as a component of our technique. Third, a brief review of robotic mapping techniques will be provided. Finally, papers which have dealt with autonomy during mapping or localization will be discussed in order to justify our choices for exploration strategies.

2.1 Related Sensor Network Applications

Camera networks have previously been applied to the task of detection and tracking [20, 59, 23, 9, 10]. Since the targets tracked in these systems have a variety of appearances and motions, the problem is much more difficult than that of our scenario (due to lack of cooperative targets and a controllable robot). So, these approaches presuppose that camera positions are either known or unnecessary.

In several cases inference of camera network topology from moving targets has also been considered [9, 32, 18]. These methods employ probabilistic inference techniques to find the most likely connections between cameras based on observations. Both Ellis *e*t al. [9] and Funiak *et al.* [18] depend on cameras with overlapping fields of view. Marinakis *et al.* [32] deal with non-overlapping cameras by using time

between detections in subsequent cameras to estimate distance. In each of these approaches, only topological information is inferred while we are interested in producing a metric map of the cameras. Batalin and Sukhatme [4] used the radio signals from nodes in a sensor network for robot localization. The spirit of this system is quite similar to our own, but the use of cameras instead of radio signal strength presents us with a large number of new challenges and also with several advantages. Moreover, the previous work considered only localization, while our system also maps the camera poses.

Cooperative localization of multiple robots can be viewed as a more general case of our work, where many of the agents are mobile instead of only a single moving agent among a stationary network. This scenario has been considered by many authors, *e.g.*, [28, 43, 47, 22]. Coordination problems inherent in a multi-robot system cause the focus of these methods to be slightly different from our own. Also, we are specifically concerned with the sensor calibration problem for our stationary nodes, where the previous work on cooperative localization has assumed each agent's sensor was pre-calibrated.

2.2 Camera Calibration

Camera calibration is the process of recovering the internal parameters of an imaging device which govern the image formation process so that scene geometry can subsequently be determined. Camera calibration is a well established problem; a good summary paper by Tsai [56] outlines much of the previous work. Zhang [61] and Faugeras *et al.* [11] present improvements made more recently. We employ the

calibration procedure suggested by Zhang [61] in our work, so it warrants further discussion.

The starting point for modelling camera behavior is the pin-hole camera model shown in Figure 2–1. This model is represented mathematically as a projective transformation. A camera is a projective device, mapping information about the 3-D world onto a 2-D image plane. A point in the world $M = [X, Y, Z]^T$ is mapped to pixel $m = [u, v]^T$ in the image, under the following projective equation:

$$\underbrace{s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{\tilde{\mathbf{m}}} = \underbrace{\begin{bmatrix} f_x & \alpha & u_x \\ 0 & f_y & u_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} R & t \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\tilde{\mathbf{M}}} \tag{2.1}$$

In matrix A, f_x and f_y represent the focal lengths in pixel related coordinates, α is a skew parameter, and u_x and u_y are the coordinates of the center of the image. Collectively, these elements are referred to as intrinsic camera parameters. The T matrix is a homogeneous transformation made up of the rotation matrix R and the translation vector t and it expresses the position and the orientation of the camera with respect to the calibration-target coordinate frame. The elements of T are referred to as extrinsic parameters because they describe the scene geometry and change every time the camera or the calibration target moves. We will use the T matrix as a measurement in the global mapping process described in Section 3.2.

In order to solve for the parameters, a set of images of a target with known geometry is used to give a number of correspondences $[u, v]^T \rightarrow [X, Y, Z]^T$, which are related by Equation 2.1. This relation allows the intrinsic camera parameters and the extrinsic parameters of each image to be jointly estimated using a two-step process. The first step is a linear solution to initialize the intrinsic parameters. The second step is a nonlinear optimization which includes polynomial distortion parameters. As with any nonlinear optimization, this process is subject to local maxima and divergence, so it is important that the initialization in the first step is quite accurate. It will be important for our further discussion to mention what Zhang [61] calls "degenerate configurations" where additional views of the calibration target do not provide sufficient information for the linear solution required in the first step. The strongest result given is that any two calibration planes which are mutually parallel do not provide sufficient information for calibration. The intuition here is that the rotation matrix R is used to produce constraints on the intrinsic parameters; as the rotation matrices for parallel planes are linearly dependent, they produce an under-constrained system. To avoid this situation, several different local motion strategies, used to obtain an adequate set of images are discussed in Chapter 4.

We are particularly interested in automated calibration performed by a mobile robot and so we seek a method for increasing calibration accuracy based on properties of the image set collected. This is similar to the series of papers by Tsai *et al.* [58, 57] which included a fairly complete study of calibration error as a function of properties of a calibration image set. Unfortunately, these methods only deal with a single camera and use manipulators with accurate joint encoders, *i.e.*, odometry

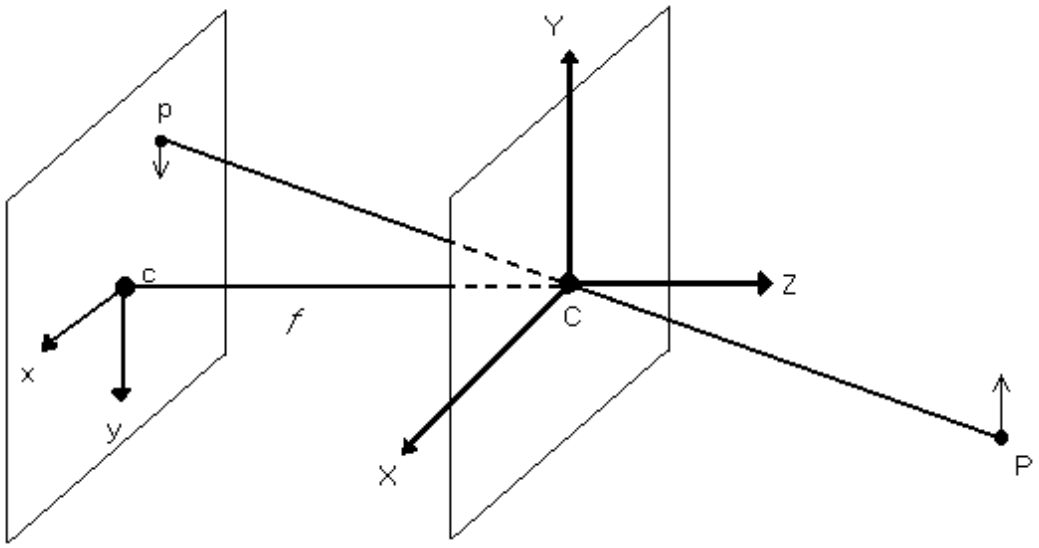


Figure 2–1: The pinhole camera model is a simple description for image formation.

error is not a factor. In the mobile robot context, constraints on the space of feasible motions and the presence of large-scale odometry error makes the problem much more challenging.

2.2.1 Fiducial Markers

One important step in the automation of the camera calibration process is the accurate detection of the calibration pattern placed within a complex scene. The complexities associated with detecting standard targets such as checkerboards, which may often resemble image background, can be avoided through the use of patterns engineered to be reliably detected by a computer vision algorithm. Such systems are referred to as "fiducial markers" and include square planar targets such as Cybercode [40], circular targets such as those discussed by Naimark and Foxlin [38] and many

others. ARToolkit [39] and ARTag [14] are two recent examples developed for use in Augmented Reality applications.

The ARTag markers utilized in our work are square black and white patches with a relatively thick solid outer boundary and an internal 6 by 6 grid (see Figures 2–2 and 2–3). The outer border is used for quad detection and the internal grid uniquely identifies each marker even under arbitrary rotation and reflection. Digital image processing techniques allow for robustness to lighting and error correction which allows detection even under partial occlusion. The advantages of such a digital system are reliable marker detection with low rates of false positive detection and marker confusion. ARTag markers have been previously used for robot localization in [15] where a camera from above viewed robots, each of which had one marker attached to its top. Our system extends this concept to allow multiple cameras in general position. Also, camera calibration based on fiducial markers has been considered [16]. Our work provides an additional level of automation by employing a mobile robot for calibration image collection.

While it may not be feasible or appropriate to permanently place engineered targets inside the buildings where a robot operates, the scenario presented in this thesis is somewhat more realistic. The ARTag target is carried by the robot, so that its presence is temporary and there is no defamation of the environment.

2.3 Map Building

In order for a mobile robot to perform tasks in an environment, it must first be able to answer the question: "Where am I?". When this question is posed with

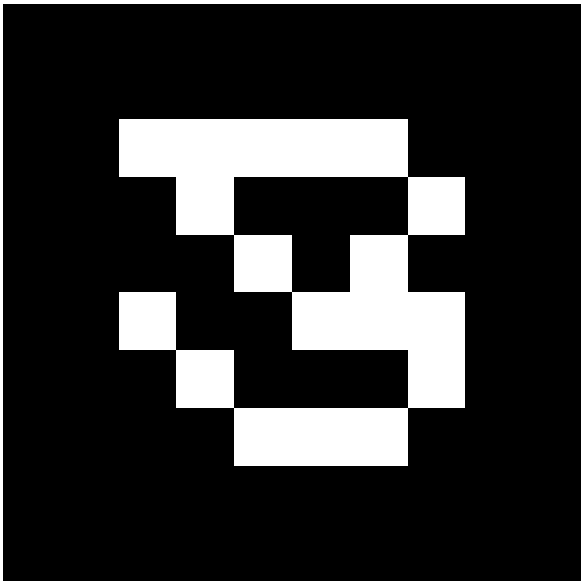


Figure 2–2: A single ARTag marker.

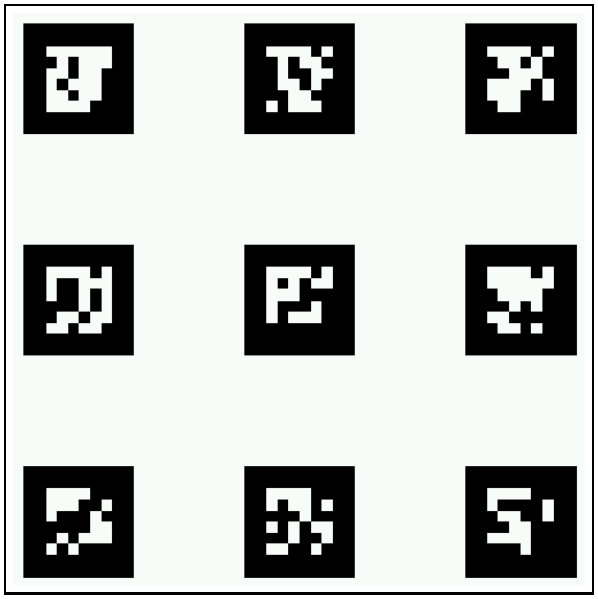


Figure 2–3: The calibration target is formed out of a regular grid of ARTag markers.

respect to a known previously existing map of the environment, the problem is commonly referred to as Localization. Simultaneous Localization and Mapping (SLAM) answers the question in a more general case: where the robot has no map to begin with and must construct a map at the same time as performing Localization. Traditionally, SLAM solutions involve estimating the positions of landmarks in the environment as well as the robot's position based on measurements from robot-mounted sensors and odometry estimates. Numerous solutions to this problem have been considered such as scan matching for alignment [30], the Extended Kalman filter (EKF) [24, 53, 29], FastSLAM or Rao-Blackwellized Particle Filters (PF) [36, 54] and others to form the now very mature SLAM field.

In this thesis, SLAM is solved using the EKF, so it will be discussed in slightly more detail. The standard Kalman filter (KF) developed by Kalman in 1960 [24] is a method to estimate the state, X, and covariance of the estimate, P, of a time-evolving system under certain assumptions about the properties of the system. Specifically, the KF assumes zero-mean, Gaussian distributed error in the process model of the system evolution over time and the measurement model of information obtained about the system during its evolution. Also, the KF assumes that each of the process and measurement models are linear in the state variables. If these assumptions hold, the KF is proveably the best linear unbiased estimator of the state at each timestep. In many real world systems, it is not possible to express the system model using linear equations, and so the EKF is a generalization for nonlinear models. In this case, there are fewer guarantees about the optimality of the state estimates; however, the EKF has been shown to provide positive results in many practical applications

(for example, in the robotic mapping domain as first demonstrated by Smith et al [53]).

An example of previous use of camera networks for localization and mapping using a Particle Filter is Rekleitis and Dudek [46]. Our work extends several aspects of this previous method. First, ARTag markers allow automated detection of calibration target points without human interaction which was required in the previous work. Second, the EKF estimates the joint probability distribution over the 3-D position and orientation of all cameras, which encodes *correlation between estimates* of the camera positions and robot position estimates. The Particle Filter employed in the previous work maintained a probability distribution only over the robot's position, and used a Maximum Likelihood estimate for camera locations. As a result, our state estimation method shows improved convergence and accuracy. Finally, our method examines both local and global planning which has not been previously considered. Each of these extensions gives our system a higher level of autonomy and allows mapping of much larger environments.

2.4 Exploration Planning

We consider robot motion planning, which attempts to achieve three related goals: the environment should be explored efficiently with respect to distance traveled; the robot's position should be estimated accurately; and the camera positions should be estimated accurately. There is a large and active body of work related to achieving each of these three goals either independently, or in combination. This section will provide a summary of the previous work grouped according to the goals listed above. This cannot be an exhaustive summary due to the large volume of

previous work, but instead focuses on the approaches most closely related to our own.

2.4.1 Coverage and Exploration

In order to produce a map of an environment, the robot must move so that it explores each region, or covers the environment. The problem of planning robot motions to achieve coverage has been considered in the case of a topological, or graph-like environment, as pioneered by Kuipers [27]. Various techniques have been proposed, often assuming that minimal information is available [8, 25, 45]. Choset *et al.* [6] study completeness of exploration. Rekleitis *et al.* [44] propose an alternative approach which minimizes uncertainty during exploration.

The coverage problem has also been studied in the case of a continuous metric representation of the environment. One such representation is the grid-based decomposition of the world known as occupancy grids, pioneered by Moravec and Elfes [37], where each grid cell stores the probability of the space being occupied by an obstacle. For occupancy grids, the goal of coverage is to maximize the information contained in the map by ensuring that the sensors have covered as much of the environment as possible. One of the first works in this field, Frontier-Based Navigation, developed by Yamauchi *et al.* [60], is still a commonly used strategy. For exploration of continuous space with an even more general representation, Baeza-Yates *et al.* [3] provide analytical results for the optimality of spiral-search given a particular set of assumptions.

2.4.2 Active Robot Localization

Robot localization is the problem of determining the robot's position within a previously known map using its motion and sensor readings. Due to a phenomenon known as perceptual aliasing, which means the sensor readings from two positions are often indistinguishable, robot motion is necessary in order to produce an unambiguous estimate. The term Active Localization refers to the problem of planning motions in order to infer the true position as quickly as possible. Thrun *et al.* [17] study this problem in the context of a probabilistic model for robot motion and sensing where the robot's position is represented as a probability density function or belief. Paths are planned by selecting points which minimize the resulting uncertainty of this belief. **Entropy** is the measure of the uncertainty in a distribution and is defined as:

$$H(p(\xi)) \equiv -\int p(\xi)\log(p(\xi))d\xi \tag{2.2}$$

Several authors have also considered reducing the uncertainty of the robot's position estimate during localization. A heuristic strategy for robots operating in an occupancy grid map using laser scanners proposed by Roy *et al.* and titled Coastal Navigation [48] recommends that the robot remains close to walls in the environment in order to receive the most informative measurements. This method was implemented for a museum tourguide robot and shown to be successful in a real world situation. Kollar and Roy [26] examine optimizing a continuous low-level path such that the uncertainty accumulated in the estimate of the robot's pose is

minimized while traveling between two landmarks in the environment. The goal of this work is to find the cubic spline that minimizes the resulting entropy in position estimate after odometry accumulation.

2.4.3 Map Error Reduction

The assumption of a previously existing map which is made in localization approaches does not hold for the case in which a robot builds a map of the environment online - the SLAM problem. Robot motion planning is again important in this case, as the estimates of map quantities can be improved through repeated visits by the robot. The exact formulation of the planning problem in this case is quite dependent on the underlying SLAM algorithm employed. Several approaches have considered planning for FastSLAM-like [36] methods where "closing the loop", or returning to previously explored territory is an explicit goal [55, 54]. In work that is more closely related to our own, planning to reduce error in an EKF SLAM estimate has been considered by many authors [52, 13, 21, 49]. These approaches seek to reduce the expected error in the map estimate using methods similar to the entropy reduction techniques used for localization and described in the previous section. For the Gaussian distributions used by an EKF representation of the environment, entropy can be expressed in closed form. Sim and Roy [52] discuss two different measures from information theory for which either the trace or determinant of the covariance matrix provides the final measure for uncertainty.

The majority of previous approaches have employed entropy minimization to facilitate greedy search for planning. This is typically justified because planning optimally over longer distances is exponential for many of the problem formulations

discussed. Sim and Roy have [52] proposed a longer term path planning strategy. They employ a variant of breadth first search where loops are pruned in order to find global paths with computational expense approximately quadratic in the number of possible locations the robot can occupy. This type of planning comes quite close to the goals of our system; however, as presented, it is limited to motion within a bounded region near a set of previously explored landmarks. Neither exploration of free space nor minimum travel distance are considered. We will discuss these two extensions in Chapter 5.

2.4.4 Multiple Objective Functions

As stated initially in this section, the goals of minimizing distance traveled, robot positional uncertainty and map uncertainty are not mutually exclusive. In fact, we would like to achieve all three simultaneously; however, this greatly complicates the planning process. Several authors have previously studied this problem in different contexts. In two papers combining EKF type SLAM with occupancy grids, Makarenko *et al.* [31] and Bourgault *et al.* [5] combine the objectives of minimizing the entropy of the EKF, lowering the entropy of an occupancy grid, and traveling minimal distance. Both papers employ a cost function which is a linear combination with contributions from each of these disparate goals which they name (I)nformation Gain, (N)avigability and (L)ocalizability weighted by arbitrary factors. Their cost function becomes:

$$U^{TOT} = \omega_I U^I + \omega_N U^N + \omega_L U^L \tag{2.3}$$

The authors do not provide a concrete scheme for determining the weights ω, instead showing that they are able to modify the weights in order to produce desireable robot behaviour. This need to choose weights is a limitation of the method as presented, and is an area for potential further research. For example, an automated method for choosing weights such as gradient descent on a set of sample environments may yield improved performance. In more recent work, Amigoni *et al.* [2] have attempted to avoid explicitly choosing weights through the use of a Pareto-optimal criterion for finding dominated goal locations, where each one of the goals can be considered independently. Finally, several authors have studied of the effect of a number of different parameterized trajectories for this problem which explicitly examine the tradeoff between distance traveled for exploration and resulting map error [46, 50, 51].

CHAPTER 3
Calibration and Mapping

This section presents a fully automated system for recovering camera calibration information and a map of the camera locations using the motion of a mobile robot equipped with a calibration target in the environment. Figure 3–1 illustrates a typical scenario in which the robot follows a path in front of four cameras placed in hallways of an office building. In order to build a map in this situation, the position of each camera relative to the robot must be calculated based on the images obtained as the robot passes through the camera field of view. In addition, as the robot moves between different cameras, its position must be tracked using odometry so that all cameras can be placed in a single map. The two steps required to solve this problem are discussed in this chapter: Section 3.1 outlines an automated camera calibration procedure which allows the robot to obtain relative camera position measurements with no human operator interaction; and Section 3.2 provides a mapping algorithm which combines these measurements with the robot's odometry to place each camera in a common reference frame.

3.1 Automated Camera Calibration

We wish to obtain pose information based on the images from the cameras in our sensor network. This means transforming the 2-D image information into 3-D position measurements. This problem has been the focus of many years of study, spawning fields of work referred to as shape from X, with $X \epsilon \{shading, shadows, stereo,$

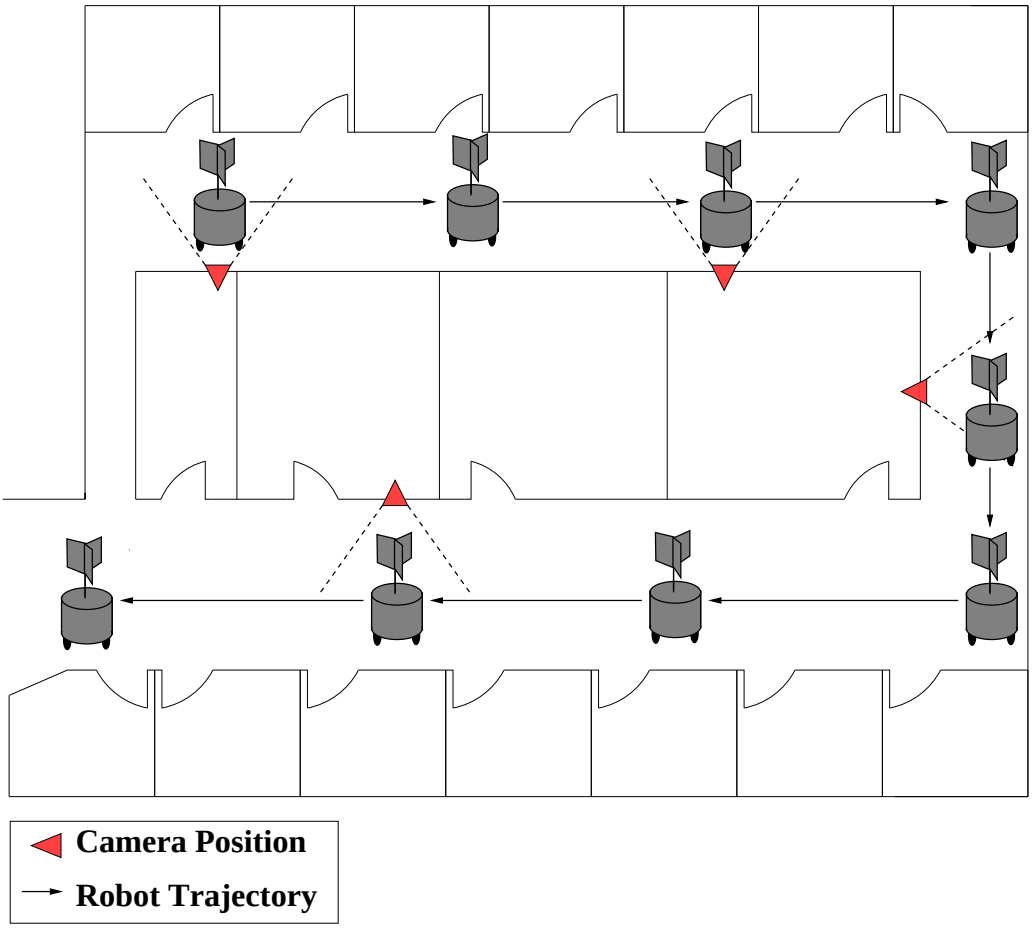


Figure 3–1: The calibration and mapping scenario described in this thesis. The robot moves through the environment, calibrating each camera and estimating both its own position as well as the positions of each camera in a common coordinate frame.

texture, ...}, depending on the cues utilized. In order to avoid many of the challenges present in the general case, an engineered calibration target can be used to simplify the problem. In this case, the following steps must take place in order to provide the necessary information to solve the localization problem:

Detection: The presence of the target must be confirmed when it does indeed appear in an image, without "false detections" which indicate the system fallaciously believes the target is in the image.

Location of Known Target Positions: In order to exploit the known geometry of the target, the image coordinates of critical target points must be located. This process must be accurate, and again must occur without false detections.

Position Inference: Based on known target geometry and the extracted image points, relative position information must be estimated.

A fully automated system is presented for the three tasks involved in camera calibration: collecting a set of images of a calibration target; detecting points in the images which correspond to known 3-D locations in the target reference frame; and performing calibration, which solves for the camera parameters through nonlinear optimization. The key to this process is the calibration target mounted atop a mobile robot as shown in Figure 1–1. The markers on the panel are easily and robustly detected through use of the ARTag software library, so that the system can immediately be aware each time the robot passes in front of a camera. The robot can then move slightly, so that different views of the calibration targets are obtained until a sufficient number is available for calibration. Chapter 4 discusses robot motion strategies which improve the quality of calibration by ensuring the

images are obtained from favorable locations. Each planar panel is comprised of nine square ARTag markers (four corners each), thus providing 36 calibration points in an evenly spaced planar grid. The six panels placed at different orientations allow for at least one to be visible from a large fraction of possible viewing directions. The 3-D locations of each marker corner in the robot frame can be determined through simple measurements and these can be fed in to a calibration routine along with the detected image points to recover calibration information.

The ARTag system requires that each marker occupies a sufficient portion of the image for the relatively fine details of the internal six by six grid to be identified. This imposes a limit of approximately 15 pixels as the minimum marker size in the image for robust detection, which translates into a maximum distance from the camera at which the calibration panel can be identified. The specific distance depends on camera resolution and imaging properties as well as the size of the target. With inexpensive cameras and a letter-sized paper target, approximately a 2 m maximum detection distance is achieved. Of course higher-resolution camera hardware and larger calibration patterns will increase this distance.

Figure 3–2 shows example images taken during the automated calibration process. Robot motion is able to provide sufficient variation in the images collected to avoid the degenerate configurations mentioned in Section 2.2 and the ARTag library detects the image points with extremely low false positive rates. The detected image points are matched to known locations on the calibration targets and this information is used as input to Zhang's [61] nonlinear calibration procedure. The results of calibration are the intrinsic parameters of the camera, such as focal length and image

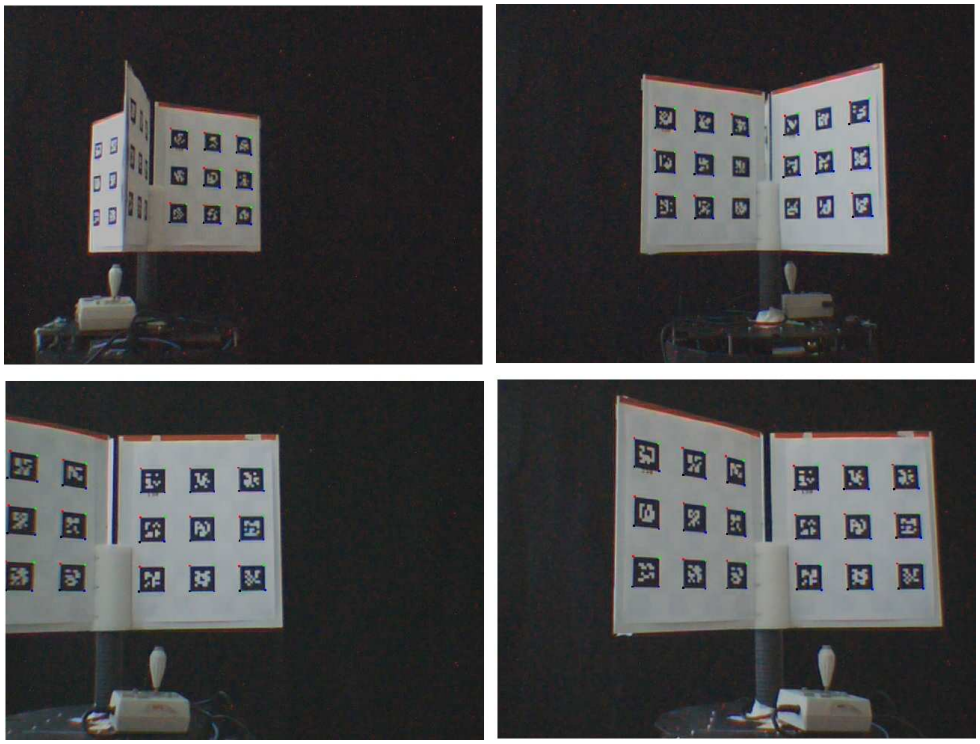


Figure 3–2: Sample images of the robot mounted calibration target which are used as input to the automated camera calibration procedure.

centre, as well as the homogeneous transformation T which relates the pose of the camera to that of the calibration panel. This information is recovered in less than two minutes, with low error rates, and with complete autonomy using our system.

Camera calibration is a nonlinear optimization procedure, which can potentially return sub-optimal solutions in the presence of local minima and may also diverge in some cases. These failure cases would be diagnosed by a human operator in a typical calibration scenario, and must be handled by our system in order to achieve fully autonomous calibration. The path planning methods presented in Chapter 4 ensure that numerous informative images can be collected by the system which reduces the

potential for failure. Further diagnostic steps to detect inaccurate calibration such as comparison of the camera measurements with the robot's odometry are potential future improvements. These approaches are not discussed in this thesis, as the system as presented achieves a high level of robustness and accuracy in practice.

In conclusion, detecting a set of images of the robot-mounted target and then detecting the grid pattern from the corners of the ARTag markers provides enough information to extract the camera intrinsic parameters and then calculate the extrinsic parameters. The extrinsic parameters of the camera provide an estimate of the camera pose relative to the robot. The next section will discuss the use of an Extended Kalman filter to combine these estimates with robot odometry in order to build a map of camera positions.

3.2 Localization and Mapping for a Camera Network

The previous section described a method for obtaining an estimate of the pose of a single camera relative to the robot through the extrinsic parameter matrix, T. These individual measurements, along with the robot's odometry, can be used to build a consistent global map of a set of cameras by adding each camera pose to the map when calibration finishes and by improving the estimates each time the robot returns to a previously mapped camera. When not observed by a camera, the robot's position can be estimated through odometry. Since odometry information

is inaccurate[1] over large distances, repeated measurements of the same camera can be used to improve the estimate. To maintain consistent estimates in this global mapping problem, an Extended Kalman filter (EKF) is used to combine noisy camera measurements and odometry. The approach of Smith *et al.* [53] for applying an EKF to robotic mapping is adapted so that general 3-D camera positions and orientations can be estimated.

The EKF for this problem will attempt to estimate the joint state vector formed by the current robot pose and the fixed camera locations. The robot pose is modeled as position on the floor plane, (x, y) and orientation, θ, which form a joint state vector $X_r = [x_r, y_r, \theta_r]^T$. The cameras may be positioned arbitrarily; so, their 3-D position and orientation must be estimated. Roll, pitch, and yaw angles are used to describe orientation, thus the pose of camera i has six degrees of freedom (6DOF) and is represented as a vector $X_{ci} = [x_{ci}, y_{ci}, z_{ci}, \alpha_{ci}, \beta_{ci}, \gamma_{ci}]^T$. There are numerous other representations for 3-D orientations, each of which has advantages and disadvantages. We have chosen roll, pitch and yaw angles here for the sake of simplicity and intuition. For further discussion of angle representations, see Craig [7].

The EKF tracks the states of the robot and the cameras in two steps: the propagation step tracks the robot pose during motion, and the update step corrects

[1] A careful reader will naturally ask why this inaccurate odometry information is used to locate the cameras. Recall that more accurate methods are desirable when available, but are often impractical as described in Chapter 1.

the robot and the camera poses based on the measurements from the calibration process.

3.2.1 Propagation Equations

The kinematic equations for a differential drive robot based on the control inputs $U_t = [v_t, \omega_t]^T$ where v_t is forward velocity and ω_t turning angle at each timestep are:

$$X_{t+1|t} = f(X_{t|t}, U_t) \tag{3.1}$$

$$\begin{bmatrix} x_{t+1|t} \\ y_{t+1|t} \\ \theta_{t+1|t} \end{bmatrix} = \begin{bmatrix} x_{t|t} + v_t \Delta_t \cos(\theta_{t|t}) \\ y_{t|t} + v_t \Delta_t \sin(\theta_{t|t}) \\ \theta_{t|t} + \omega_t \Delta_t \end{bmatrix} \tag{3.2}$$

This is a nonlinear function, so a Taylor Expansion must be used to linearize. The state vector and the covariance matrix are updated through the following equations:

$$\hat{X}_{t+1|t} = F\hat{X}_{t|t} \tag{3.3}$$

$$P_{t+1|t} = FP_{t|t}F^T + C_v \tag{3.4}$$

where F is the Jacobian of Equation 3.2 and C_v is a matrix representing odometry error.

3.2.2 Measurement Equations

When the robot is observed by a camera, the calibration information allows a 3-D coordinate transformation to be estimated. This information relates two state variables in our EKF: the robot pose and the pose of the camera. This information provides a measurement equation which is a nonlinear expression of the state variables. In order to apply the KF update equations, we must again linearize before

using the Kalman filter update equations. The measurement equation relates two coordinate frames, so the language of homogeneous coordinates transformations is used in order to express the relation. In general, any two coordinate frames are related by a transformation matrix as follows:

$$ {}^{b}_{a}T \;=\; \begin{bmatrix} {}^{b}_{a}R & {}^{b}P_{a_{orig}} \\ 0_{1x3} & 1 \end{bmatrix} \tag{3.5} $$

In this case, the transformation expresses frame a in the coordinates of frame b. ${}^{b}_{a}R$ is the 3×3 rotation matrix which represents the orientation of a as seen in b. ${}^{b}P_{a_{orig}}$ is the translation vector which represents the position of the origin of frame a in coordinates of frame b.

The Kalman filter state encodes position and orientation as a 6-D vector while our derivation of the measurement equation is based on homogeneous transformation matrix representations of position and orientation. Fortunately, this is not a problem, because these two representations can be related through well known formulas (see, for example, Craig [7]). A homogeneous transformation T can be generated from a state vector using Equation 3.6:

$$ \begin{aligned} T(X) &= T(x, y, z, roll, pitch, yaw) \\ &= \begin{bmatrix} cy*cp & cy*sp*sr - sy*cr & cy*sp*cr + sy*sr & x \\ sy*cp & sy*sp*sr + cy*cr & sy*sp*cr - cy*sr & y \\ -sp & cp*sr & cp*cr & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.6} $$

Here trigonometric functions and variable names are abbreviated for brevity. For example, cp is written in place of $\cos(pitch)$ and sy instead of $sin(yaw)$. For the inverse operation, a state vector X can be recovered from a homogeneous transformation using Equation 3.7:

$$
\begin{aligned}
X(T) &= X \begin{pmatrix} R_{3x3} & P_{3x1} \\ 0_{1x3} & 1 \end{pmatrix} \\
\begin{bmatrix} x \\ y \\ z \\ roll \\ pitch \\ yaw \end{bmatrix} &= \begin{bmatrix} P(1) \\ P(2) \\ P(3) \\ arctan(\frac{R(3,2)}{R(3,3)}) \\ arctan(\frac{-R(3,1)}{sqrt(R(1,1)^2+R(2,1)^2)}) \\ arctan(\frac{R(2,1)}{R(1,1)}) \end{bmatrix}
\end{aligned}
\tag{3.7}
$$

Deriving the measurement equation involves comparing the result of the calibration process to the predicted result based on the current EKF state. Calibration provides the position of the panels atop the robot in the camera's coordinate frame and the EKF state encodes the position of both the robot and the camera in world coordinates. There are many ways to compare the two pieces of information, but for simplicity, in the following, they will both be used to obtain the transformation describing the robot in the camera's reference frame, namely ${}^{C}_{R}T$. The structure of the derivation will be to first use the calibration information to produce ${}^{C}_{R}T_{measured}$ and then to use the EKF state information to produce ${}^{C}_{R}T_{predicted}$. The reader is referred to Figure 3–3 as a visual guide.

For the first step, the calibration process estimates the extrinsic parameters which represent the calibration panel in the camera frame, that is ${}^{C}_{P}T$. Since each panel i is rigidly attached to the robot, the transformation between the panel and the robot, namely ${}^{P}_{R}T(i)$ is easily measured and treated as a constant throughout the procedure. When a new measurement arrives, it can immediately be used to relate the camera and the robot coordinates by:

$$ {}^{C}_{R}T_{measured} = {}^{C}_{P}T_{fromCalibration}\ {}^{P}_{R}T_{constant}(i) \tag{3.8} $$

The second step requires the transformation ${}^{C}_{R}T$ to also be expressed in terms of the current state of the EKF, through information contained in the pose of the robot, X_r, and the pose of the camera, X_c. These state vectors encode both the robot and the camera positions in world coordinates and can produce transformation matrices ${}^{W}_{R}T$ and ${}^{W}_{C}T$ using Equation 3.6. These transformations can be combined to express ${}^{C}_{R}T$ in terms of the EKF state vectors as desired as shown in Equation 3.9 below.

$$ \begin{aligned} {}^{C}_{R}T_{predicted} &= {}^{C}_{W}T\,{}^{W}_{R}T \\ {}^{C}_{R}T_{predicted} &= {}^{W}_{C}T^{-1}\,{}^{W}_{R}T \\ {}^{C}_{R}T_{predicted} &= T(X_{ci})^{-1}T(X_r) \end{aligned} \tag{3.9} $$

Equation 3.9 can be expanded and simplified. First, the rules for inverting homogeneous transformations are used to arrive at Equation 3.10. That is, rotation matrices are orthogonal ($RR^T = I$ or $R^T = R^{-1}$). The expression can then be

combined into a single transformation through matrix multiplication as in Equation 3.11.

$$= \begin{bmatrix} {}^{W}_{C}R^{T} & -{}^{W}_{C}R^{T}{}^{W}_{C}P \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{W}_{R}R & {}^{W}_{R}P \\ 0 & 1 \end{bmatrix} \tag{3.10}$$

$$= \begin{bmatrix} {}^{W}_{C}R^{T}{}^{W}_{R}R & {}^{W}_{C}R^{T}({}^{W}_{R}P - {}^{W}_{C}P) \\ 0 & 1 \end{bmatrix} \tag{3.11}$$

In fact, although Equations 3.8 and 3.11 could be used directly as the measurement equation of this system, each expresses a homogeneous transformation which uses twelve entries to express inherently six dimensional information. Since the complexity of computing the Kalman filter updates is related to the dimensionality of the measurement, it is beneficial to use a more compact representation. This problem is easily solved however, because Equation 3.7 allows both transformation matrices to be expressed as state vectors. This leads to our final measurement equation:

$$\begin{aligned} {}^{C}_{R}T_{measured} &\approx {}^{C}_{R}T_{predicted} \\ X({}^{C}_{R}T_{measured}) &\approx X({}^{C}_{R}T_{predicted}) \\ z_{measured} &= h(\hat{X}) + C_{\omega} \end{aligned} \tag{3.12}$$

Measurement noise C_{ω} expresses the uncertainty of transformation parameters from camera calibration. Equation 3.12 is now of the form that can be used in a Kalman filter, although it is nonlinear. This means that h must be differentiated with

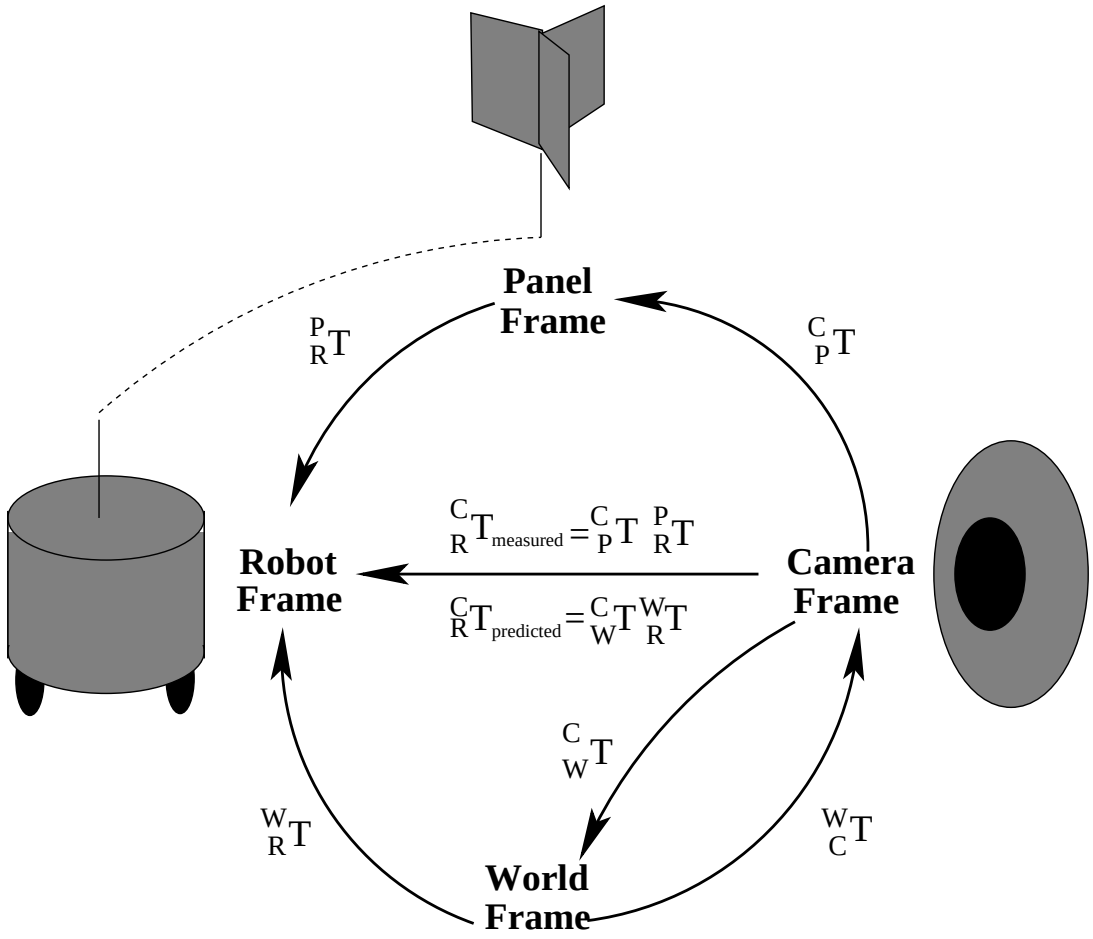


Figure 3–3: Positions of the robot, camera, and panel are related using homogeneous transformation matrices T. Each arc in the diagram represents relative information either computed with the EKF, measured through calibration, or computed through combination of these sources. Deriving the measurement equation involves representing ${}^{C}_{R}T$ using both the EKF information and the results of calibration so that these two results can be compared.

respect to each parameter to obtain a first-order linear approximation $z \approx h(\hat{X}) + H\tilde{X}$ where H is the Jacobian of vector function h in Equation 3.12. The EKF update equations can then be applied as usual:

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + K(z - h(\hat{X}_{t|t-1})) \tag{3.13}$$

$$P_{t|t} = \left[I - KH^T\right] P_{t|t-1} \tag{3.14}$$

$$K = P_{t|t-1}H(HP_{t|t-1}H^T + C_\omega)^{-1} \tag{3.15}$$

3.2.3 UDU Factored Covariance Filter

The KF is proveably optimal in the sense that it tracks the true mean and the correct error covariance under the given assumptions. Unfortunately, the recursions do not provide a numerically stable algorithm. That is, when the filter results are calculated on a computer with finite precision (for example, using double precision floating point arithmetic), roundoff operations cause large errors in the final values. Numerical stability is a property of an algorithm which determines the sensitivity of the solution's accuracy with respect to the error in the input data. In the case of the standard recursions to compute the EKF estimate, the operation which is most problematic for accurate computation is the inverse required for computing the Kalman Gain, K in Equation 3.15. In theory, if P and C_ω are both positive semi-definite, which all covariance matrices should be by definition ($cov(X) \equiv E[X^T X]$), then the inverse should exist and also be positive semi-definite. In real implementations, P can be quite close to singular, in which case computation of the inverse produces large errors. Some causes of this are:

- Portions of the state vector being known very precisely, so the corresponding diagonals (and rows and columns) of P are near zero.
- More commonly, portions of the state vector are known *much better* than others, leading to badly scaled computations.
- A degenerate series of measurements, such as repeated observations of the same landmark from the same position.
- Simple accumulation of many recursions of the filter, leading to slow buildup and compounding of roundoff errors.

These problems are sometimes solved by clever engineering of the filter, such as estimating some linear combination of states to preserve scaling, or treating some states as constants to avoid estimating quantities with near-zero uncertainty. The use of an algorithm with better numerical properties is an option that allows improvement in solution quality without any difficult engineering choices, which can still be made afterwards, if further performance gain is required. There are numerous "square root filters" which are discussed by Maybeck in great detail [33]. For this project, we have chosen the UDU factored covariance filter. This filter is based on the fact that any positive semi-definite matrix can be factored into the product of an upper triangular matrix U, a diagonal matrix D and a lower triangular matrix U^T.

The UDU filter propagates the factors U and D instead of the full covariance matrix. There are numerous benefits to this approach: it is numerically more stable, giving approximately twice the effective precision as the standard recursions; it requires less storage; and it does not require square roots in the computations, unlike

Figure 3–4: Hardware for two nodes in our Sensor Network. Images from the camera at each node are obtained using a low powered computer and are transmitted over the network using a combination of wireless and wired networking.

filters based on other factorizations such as the Cholesky decomposition. The computation time is slightly higher than the traditional EKF; however this assumes only a simple inverse is used in Equation 3.15. Due to numerical errors, the pseudo-inverse is sometimes used in this step, which requires a Singular Value Decomposition and makes the computation much more expensive, so the UDU filter may in fact be more efficient in practice as well as more accurate.

3.3 Large Scale Calibration and Mapping Experiments

To demonstrate the effectiveness of the automated calibration and mapping system, we instrumented an office environment with a seven node camera Sensor Network (Figure 3–4 shows the hardware utilized). The environment consisted of a rectangular loop and a triangular loop connected by a long hallway with length approximately 50 m as shown in Figure 3–5. A Nomadics Scout robot mounted with a target with six calibration patterns was used to perform the calibration and

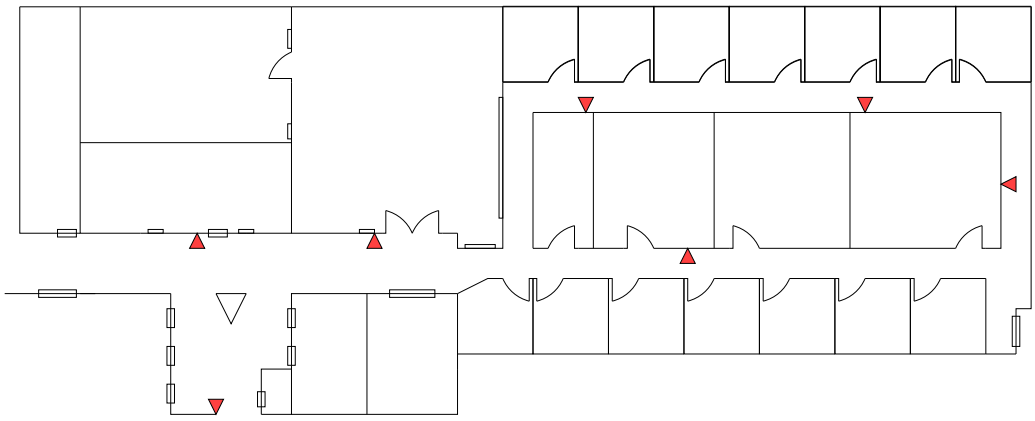


Figure 3–5: A floorplan description of the hallway environment used for calibration and mapping experiments. The seven nodes were placed on walls and doorways throughout the environment as shown with red triangles in the image.

mapping procedure. The robot traversed the two loops of the environment 3 times each and traveled in excess of 360 m in total. When in front of each camera, the robot stayed in place and performed simple rotations until sufficient images were obtained. This local calibration strategy and others will be further discussed in Chapter 4. In these tests, 30 images per camera were collected for calibration, which took less than 1 minute per camera node to collect. While equally accurate calibration is possible with many fewer than 30 images, the frequency of drastic failure during calibration decreases as more images are used for calibration. A set of 30 images was found empirically to provide adequate robustness without excess computational effort. An open source implementation which is part of the OpenCV computer vision library [1] was used for camera calibration.

Figures 3–6 and 3–7 show the odometry path estimate and the path corrected by observations from the cameras. It is visually clear that the use of camera measurements was able to correct for the buildup of odometry error relatively well.

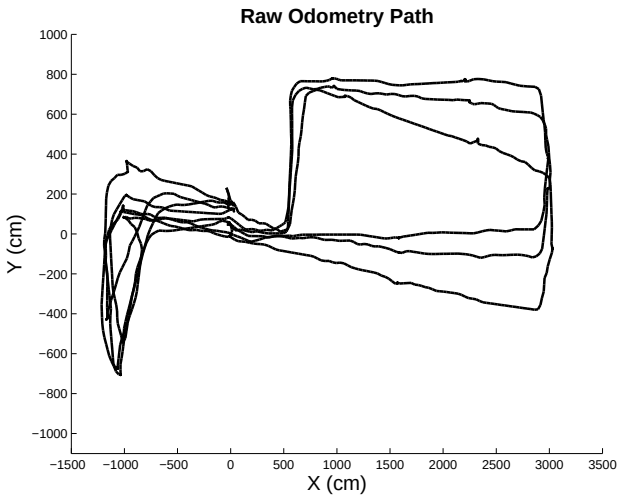


Figure 3–6: Odometry readings for hallway path.

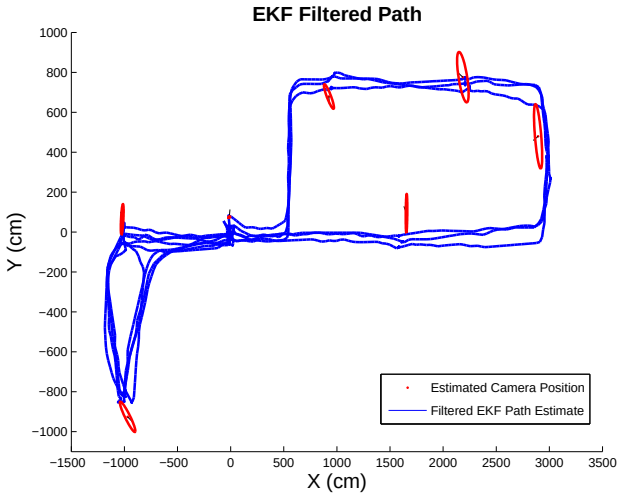


Figure 3–7: EKF estimate of the hallway path. Estimated camera positions with uncertainty ellipses are in red where colour is available.

Unfortunately, there are some regions where the filtered path is still a very rough approximation due to large distances between cameras. These distances are traveled without correction of the odometry error. Since the system does not provide a means for odometry correction between the camera fields of view, this type of behavior is unavoidable. Solutions include placing the cameras much closer together to limit the size of the unobservable regions or performing SLAM with another sensor such as sonar or laser to provide a complementary approach. We will leave further discussion of these possibilities for the conclusions.

No ground truth data was collected for this experiment due to lack of available accurate measuring devices over this size of environment. In previous work, in a similar experiment over a 15 m environment, the camera positions were found to deviate from the true positions by 2.11 cm on average with standard deviation 1.08 cm [42].

It is likely that a systematic manual localization procedure would be able to obtain lower error than that expressed by the covariance ellipses in Figure 3–7; however, this would only be possible with aid of equipment to precisely measure angles and distances. Also, it would likely take a human much more time to collect these measurements, to the extent that the entire system would be much less attractive to a potential practitioner. It is important to remember that the map in Figure 3–7 can be constructed, in principle, *completely free from human interaction*, which makes our system trivial to configure for the operator.

CHAPTER 4
Local Calibration Path Planning

Calibration routines use the locations of points detected in multiple images of a calibration target to estimate camera parameters and the target locations relative to the camera. For any given set of images, these routines produce as good an estimate as possible of the underlying parameters. Previous experimental results have focussed on the scenario where a human operator is able to collect a high quality set of calibration images. Typically a large target (20 by 20 grid points) is carefully mounted on a planar surface and placed extremely close to the camera such that the target occupies the entire field of view. As we will show later in this section, such a configuration is desireable in a number of ways: covering the whole field of view allows any lens distortion to be easily identified; being close to the camera allows for accurate detection of corner points; and perfectly planar targets give no deviation from assumptions made in the calibration routine.

The automated system proposed in this thesis imposes several limitations which may preclude such an ideal set of images from being collected. These limitations include: a very large calibration target may be too heavy or cumbersome to mount on a low powered mobile robot; the camera might be positioned high above the ground such that the robot cannot move the target close enough to cover the entire image; and the robot cannot initially have perfect information about the camera position, so its ability to find the appropriate part of the field of view is limited. In spite of these

limitations, using a robot-mounted target provides a unique opportunity to collect calibration images in an intelligent fashion by controlling robot motion. A path planner for the robot during local calibration can make decisions in order to ensure the calibration data set collected is as informative as possible under the limitations and thus maintain some measure of accuracy in the final result. Such a planner must also attempt to keep the accumulated odometry error to a minimum. It is not immediately clear what the best motion strategy will be. There are numerous sources of error including detecting the corner pixels, approximating the linear parameters, and convergence of the nonlinear optimization. Ideally, the robot should move in such a way that the resulting image set reduces the combined effect of all these error sources and gives the most accurate calibration possible.

In the remainder of this chapter, we will analyze the affects of numerous parameters of the set of images collected on calibration accuracy. We will attempt to determine whether or not there are certain properties of a single image and between the images in a set that can be shown to directly impact calibration accuracy. These parameters can subsequently be used to allow accurate autonomous calibration and mapping for our system.

4.1 Heuristic Local Trajectories

As an initial investigation into this problem, five hand-designed motion strategies were examined in order to explore the spectrum of expected calibration accuracy and odometry buildup. The strategies chosen were those that allowed for intuitively interesting properties of a calibration trajectory to be examined while keeping robot

motions simple and intuitive. These strategies are illustrated in Figure 4–1 and described briefly below.

Stationary: Once in front of the camera, the robot stays in one spot, see Figure 4–1(a). This is as simple a strategy as possible and a point for comparison to ensure robot motion is truly necessary for accurate calibration. Due to the target geometry, two non-parallel panels can be observed by the camera, which provides the minimal amount of information necessary for calibration.

One Panel Translation-only: The robot translates across the camera field of view (FOV) with only a single calibration panel visible and undergoing pure translation, see Figure 4–1(d). This is a degenerate case and did not produce good calibration; it was considered because translation is a simple behavior to implement on the robot and because the single panel undergoing pure translation illustrates the degenerate case in calibration.

Multi-Panel Translation-only: The robot translates across the camera FOV with all calibration panels visible, see Figure 4–1(e). This provides two non-parallel planes for calibration and accumulates a minimal amount of odometry error. Again, this was chosen because translation is a simple motion to implement and allowing multiple visible panels provides for comparison with the previous method.

Rotation-only: The robot rotates in place in the center of the camera FOV allowing the panels to be detected at a variety of angles, see Figure 4–1(b). Rotation is also easy to implement on a mobile robot and collecting images with a wide variety of target orientations contrasts with both of the Translation methods.

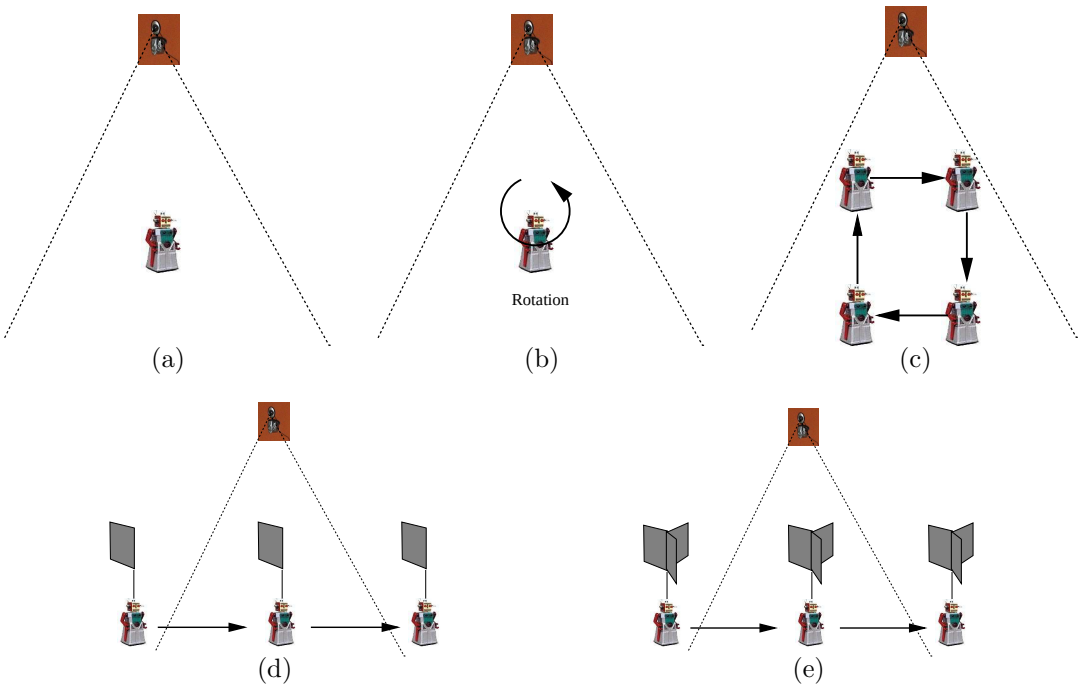


Figure 4–1: The local calibration strategies: (a) **Stationary**; (b) **Rotation-only**; (c) **Square Pattern**; (d) **Single-Panel Translation-only**; and (e) **Multi-Panel Translation-only**

Square Pattern: The robot follows a square-shaped path in front of the camera, alternating translation and rotation by 90 degrees, see Figure 4–1(c). This forms a square with 4 corners and at each corner the robot has two poses with perpendicular orientation. Since a large portion of the image is covered, and there is variation in the detected panel orientation and in depth, this method achieved good calibration accuracy. The combination of rotation and translation caused large odometry error.

Table 4–1: Mean value and percentage standard deviation of the intrinsic parameters for each strategy over 10 trials. **One Panel Translation-only** is omitted due to divergence. Deviations are with respect to the mean; ground truth error is not provided.

Path	Mean Values				Standard Deviation (%)			
	f_x	f_y	u_x	u_y	f_x	f_y	u_x	u_y
Stationary	903.2	856.0	233.5	190.6	6.3	5.6	30.9	17.1
Translation	785.8	784.3	358.0	206.4	2.7	2.3	3.6	5.0
Rotation	787.7	792.0	324.1	236.6	1.6	1.6	3.9	10.3
Square	781.2	793.1	321.4	274.2	1.2	2.0	2.4	13.9

4.1.1 Experimental Evaluation of Heuristics

The effects of the heuristic local trajectories were studied experimentally. The goals of this study were to evaluate the motion strategies in terms of reliable calibration results and magnitude of odometry error as well as verifying that different paths used by the robot have sufficient impact on final results in order for this to be an interesting area for further study. This test was done inside our laboratory with the same robot and calibration target as described in Section 3.3, but using only a single camera. The 5 strategies were performed for 10 trials, each with $n = 30$ calibration panels detected per trial. The automated detection and calibration system allowed for these 50 trials and 1500 pattern detections to occur in under 3 hours (using a Pentium IV 3.2 GHz CPU running Linux for both image and data processing).

Table 4–1 summarizes the intrinsic parameters obtained for each method. The lack of data for the **One Panel Translation-only** path is due to the fact that, as expected, calibration diverged quite badly in all trials with this method. The true intrinsic parameters were not available for the camera used for these experiments. Hence, standard deviations between a number of trials are presented in place of

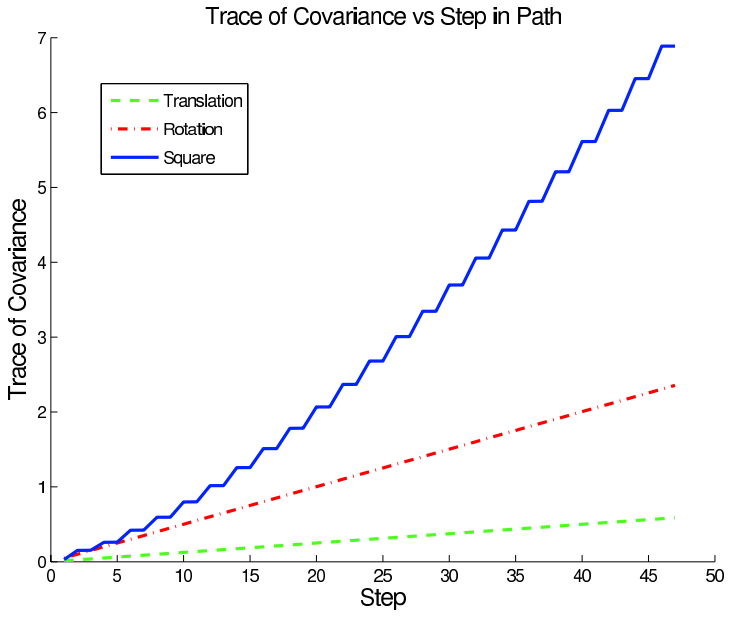


Figure 4–2: Odometry error accumulation for 3 local calibration paths

ground truth errors. This is an approximation of the calibration error which is accurate only under the assumption that there are no systematic biases present in the estimation. This is assumption is supported by the fact that, other than the **Stationary** method, for all the other strategies statistically, the mean parameter estimates are not significantly different. For the analysis in Section 4.1, a "good" calibration result is one which produces small standard deviations.

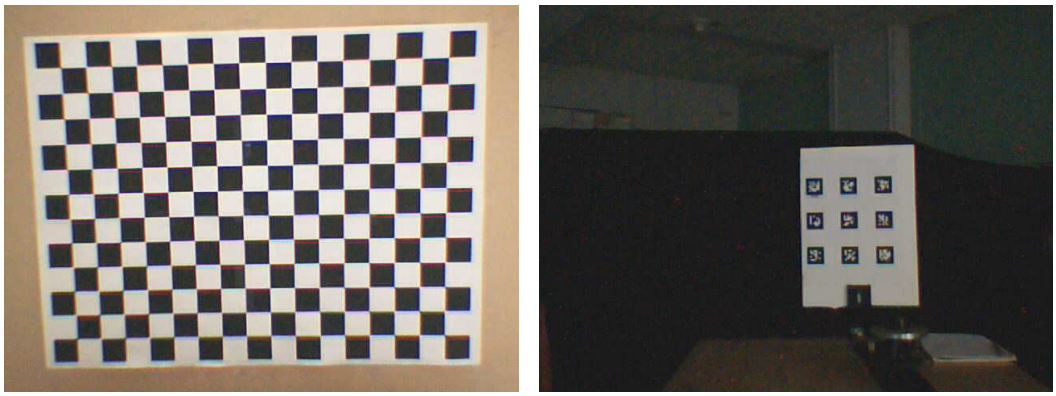
To examine the difference between odometry buildup among the different paths, each of the three paths which involved motion was simulated using an EKF (the **Stationary** approach clearly does not build any odometry error). To ensure a fair comparison, the step size of the **Translation-only** method was set equal to the side length of the square pattern (8 cm each) and the angle step in the **Rotation-only**

method was set to 90 degrees. This meant that the **Square Pattern** translated half of the distance of **Translation-only** and rotated half the angle of **Rotation-only**. Figure 4–2 shows the trace of the covariance matrix as each method progresses. The **Square Pattern** accumulates odometric error at a higher rate than the other two methods, as expected. We must note that the relative slopes in this figure are influenced by the choice of odometry error covariances in the EKF but the values used are representative of our robot's true performance.

4.2 Calibration Error Study

The previous section illustrated that fairly simple robot motions during calibration can lead to successful calibration, but at the cost of relatively large accumulation of robot odometry error. In order to go further towards automated calibration motion planning, it is necessary to relate specific aspects of scene geometry to resulting calibration errors. Intuitively, moving the target closer to the camera will allow the marker corners to be detected with smaller errors (fine detail is more apparent when examined at close range). Also, since Zhang [61] has determined that complete lack of rotation between images will give bad performance, avoiding this situation by making non-trivial rotations between at least some of the images in the calibration set will lead to improved performance as well.

A set of experiments was conducted in order to verify these somewhat loosely formed claims. First, a set of 215 images of a large calibration target with 13 by 17 squares was collected. See Figure 4–3(a) for an example image. For this set of images, the target was moved somewhat arbitrarily by hand, though always within 5 m of the target and with large varieties of angles for each position. Collecting



(a) (b)

Figure 4–3: Sample images from the two calibration data sets: (a) the large target data set; and (b) the ARTag data set with ground truth information.

images in this way (with a high quality target close to the camera) is the technique typically carried out in the calibration literature in order to obtain highly accurate calibration results. While the robot cannot be expected to duplicate such behaviour as previously discussed, this data is useful for obtaining a good estimate of the ground truth camera parameters and also as a means of comparing performance of the autonomous system to a "best case scenario".

A second set of images was collected, which was meant to represent the much less informative images that could be provided by the robot during autonomous operation. Figure 4–3(b) provides an example image. For this data set, careful ground truth regarding the calibration target positions was collected using a positioning device in order to allow principled study of the effects of the parameters. One of the three panels used on the robot (as in Figure 1–1) was moved over a set of grid positions in front of the test camera. The extents were 40 *cm* from the camera at the closest point to 140 *cm* at the farthest and directly in the centre of the camera's

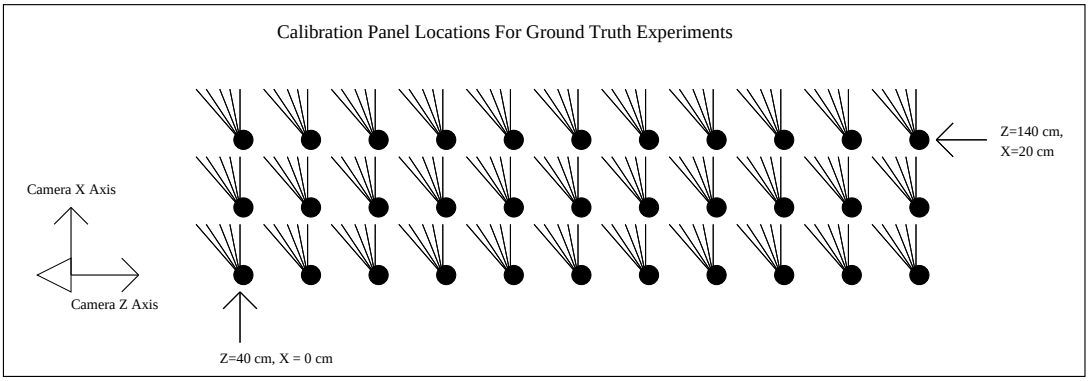


Figure 4–4: The view from above of the experimental setup for the ARTag image set including ground truth position information.

image to 20 *cm* maximum offset. At each grid point, the target was placed at 5 angles ranging from 0 degrees from orthogonal to the camera viewing direction in 10 degree increments to 40 degrees. Figure 4–4 illustrates this setup.

With the 1040 total images collected in these experiments (215 with the large target and 825 with the ARTag target), a series of experiments were run in order to accomplish the objectives set out at the beginning of this section. Their description and results will be presented below.

4.2.1 Ground Truth Parameter Determination

In order to compare further calibration results, the first step taken was to produce an accurate estimate of the ground truth camera parameters. This estimate was made by selecting subsets of images from the sets described above, performing the related calibrations and comparing the results. After preliminary testing, it became clear that larger subsets provided more repeatable calibrations. This did not lead directly to a solution, because even the results from sets as large as one hundred images showed significant variation between trials. For this reason, it was judged that using

a single calibration set, no matter how high quality the images, was not sufficient to estimate the ground truth parameters. Fortunately, as many trials are run with increasingly numbers of images per trial, the mean values of the trials at each stage can be seen to be fairly constant (see Figure 4–5 for example). Also, the variation between trials becomes less as the number of images in each trial increases. The value to which the trials converge is a well supported estimate for the ground truth solution with the spread of the values giving the uncertainty of the estimate. Figures 4–5 through 4–9 present the final estimates. The small difference between mean values computed for each parameter between the large target set and the ground truth set arises from the fact that hand measurement of 3-D target geometry was necessary, and this was not done with extreme precision for either case.

4.2.2 Calibration Error With Respect to Data Set Parameters

As mentioned above, there are many aspects of a calibration image set that could potentially affect calibration accuracy. These include, but are not limited to, the average distance to the camera, the degree of angle variation, the variation in depth, and the variation in horizontal and vertical positions. An essential question that must be answered in designing a local calibration procedure, is: "How do each of these parameters affect the resulting calibration accuracy?" The ground truth information collected with the ARTag data set allowed us to test several of these variables independently. The procedure used was to take numerous sample subsets quasi-randomly, with each subset being controlled in some aspects, such as all images from a constant depth, but random in other aspects, such as horizontal position and angle. The two parameters chosen for initial study were distance from the camera

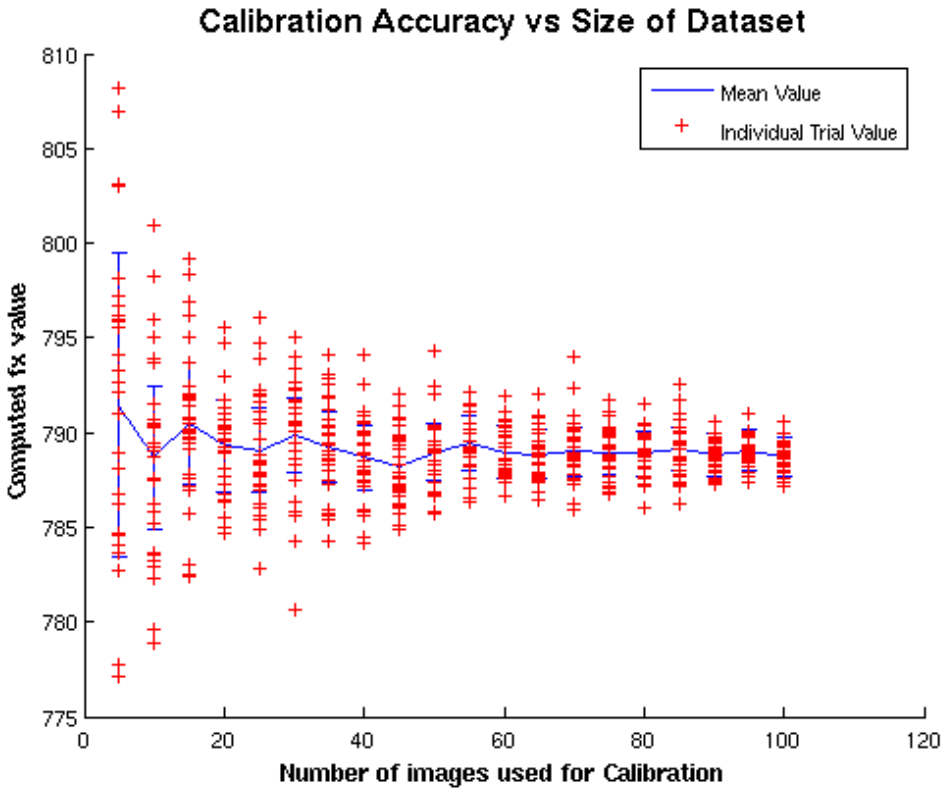


Figure 4–5: Using images from the large calibration target image set, the computed fx values becomes more consistent between trials as the number of images used is increased.

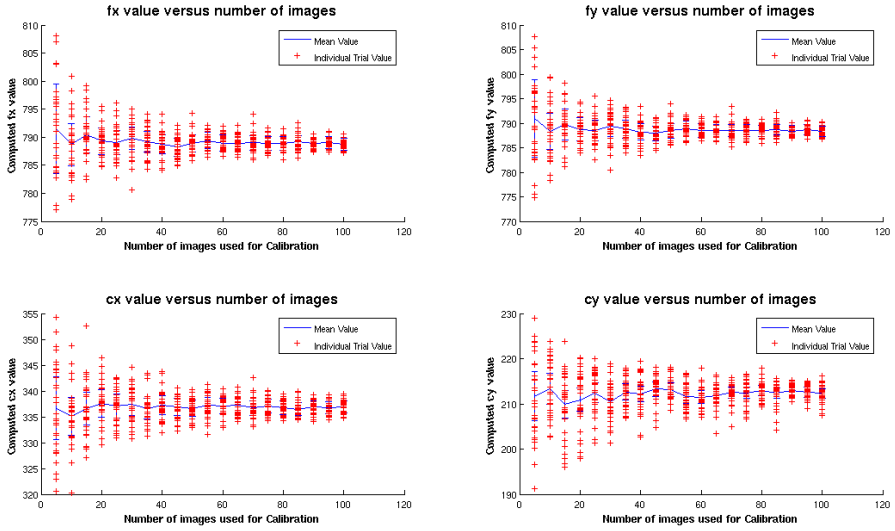


Figure 4–6: Using images from the large calibration target image set, each of the computed linear parameter values become more consistent between trials as the number of images used is increased.

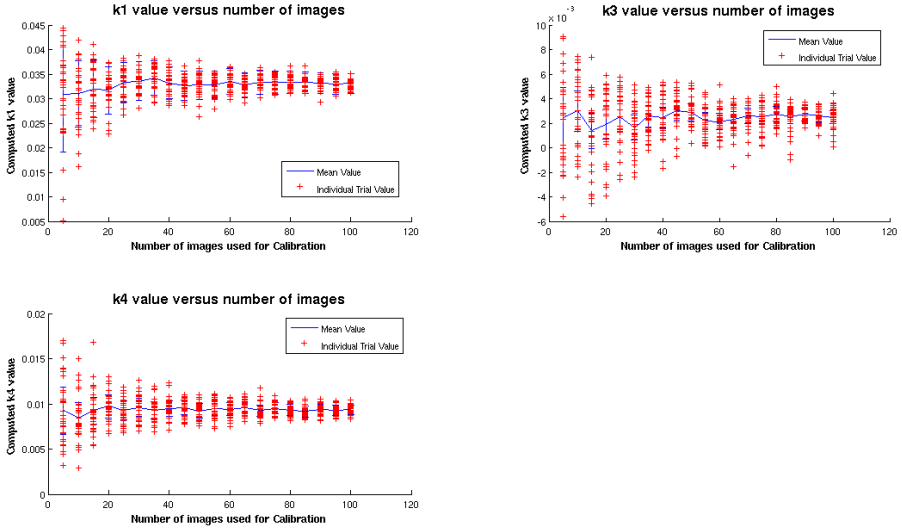


Figure 4–7: Using images from the large calibration target image set, the computed nonlinear distortion coefficients become more consistent between trials as the number of images used is increased.

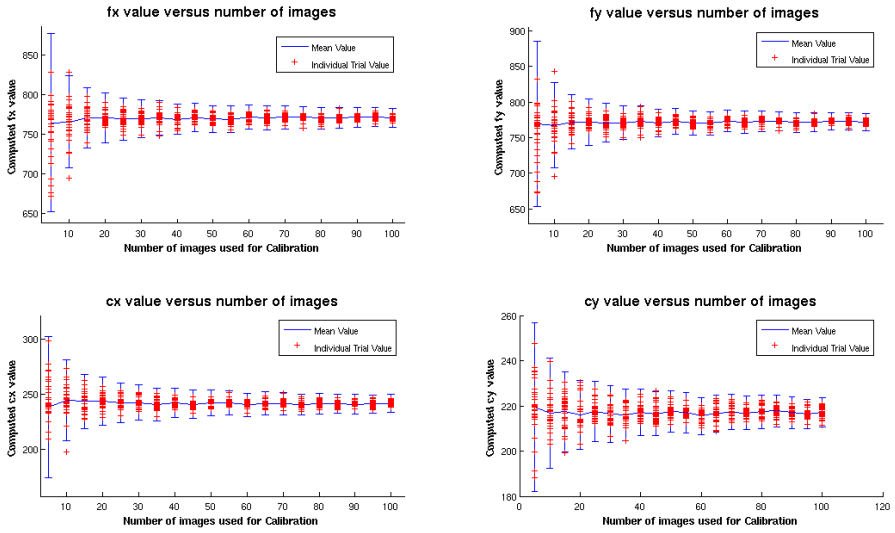


Figure 4–8: Using images from the ARTag ground truth image set, the computed linear parameter values become more consistent between trials as the number of images used is increased.

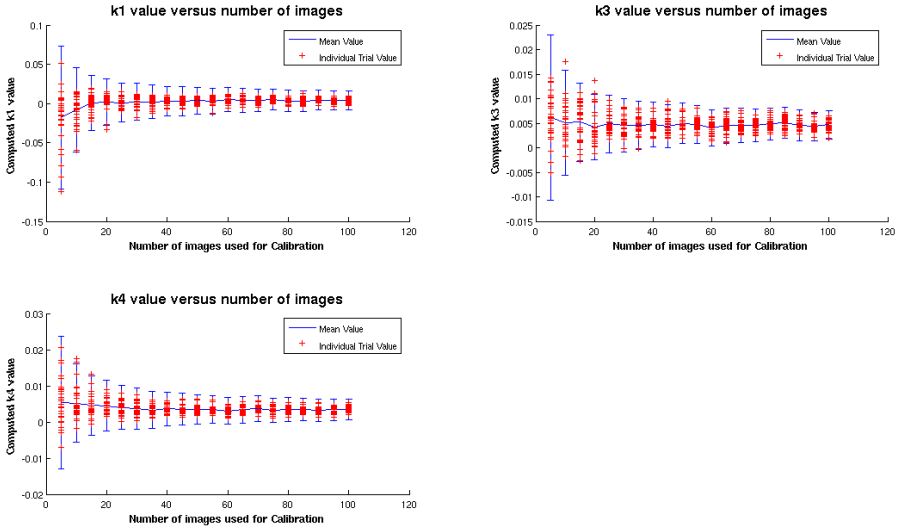


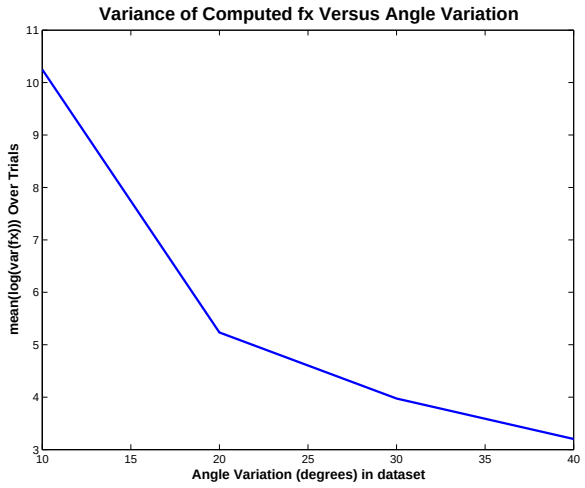
Figure 4–9: Using images from the ARTag ground truth image set, the computed nonlinear distortion coefficient values become more consistent between trials as the number of images used is increased.

and angle variation within the subset. These parameters were intuitively the most promising to directly contribute to calibration accuracy.

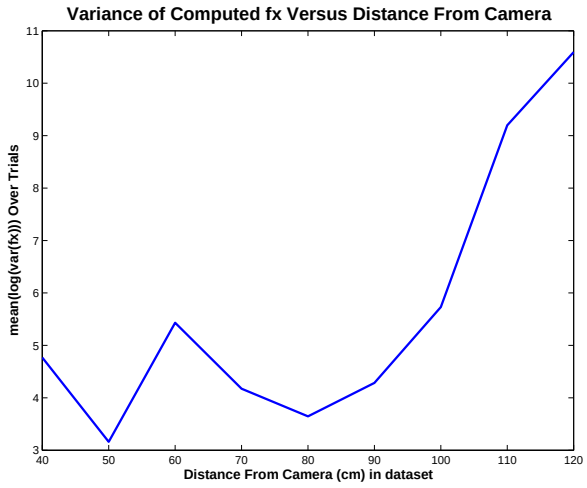
For each subset of images selected, calibration was performed in the manner described in Section 3.1. The results were compared to both the ground truth values in the previous section and between subsets with the same parameters to determine the error and variance of calibration results. In all cases examined, the mean values over trials matched the established ground truth values within the variance of the estimates. It is more interesting to note that the variance of the results depends quite strongly on each of the parameters tested, see Figures 4–10(a) and 4–10(b). In fact, we can conclude that collecting images close to the camera and with large variation between successive images leads to the best calibration accuracy. This fact is not surprising, since it is the standard practice in the previous literature, but does further inform our choice of local calibration trajectory planning.

The experimental setup used in this thesis (see Figure 1–1) inherently satisfies several criteria for image sets which achieve accurate calibration. Each of the panels in our calibration target is separated by at least 100 degrees, so a single image containing two of these panels provides sufficient angular variation to produce accurate calibration. Robot motion is necessary in order to ensure that the calibration panels are close to the camera during the collection of camera images. Achieving this autonomously is complicated by several system limitations such as the need to remain in the camera field of view and to maintain an angle from the viewing direction such that the ARTag markers can be detected. These complications can be overcome using the initial rough estimates of camera position which can be produced from a single

image of the calibration target without performing complete calibration. In short, our experimental system along with the analysis of this section allows for accurate, automated camera calibration.



(a)



(b)

Figure 4–10: The dependence of calibration accuracy on image set parameters can be determined using the ARTag image set with ground truth information. (a) The variation in computed fx value decreases as there is more angular variation within the data set. (b) The variation in computed fx value is relatively constant for distances less than 80 cm and increases for larger distances.

CHAPTER 5
Global Exploratory Trajectories

During Simultaneous Localization and Mapping, the robot's odometry estimates are combined with its sensor readings in order to estimate the robot's path as well as a map of the environment. Choosing the trajectory that a robot follows, or motion planning, is an essential component of SLAM; it determines the accuracy of the final map produced as well as the efficiency of the process. On one hand, accurate mapping is dependent on the robot's position estimate being corrected through repeated measurement of the same landmarks. That is, after initially estimating the position of landmarks when they are first observed, the robot's motion must allow it to revisit the same landmarks in order to obtain additional measurements which improve the estimates of the robot's position and of the landmark positions. On the other hand, efficient map building requires that the robot's path allows its sensors to cover the environment with minimal distance traveled, so returning to an already explored landmark is undesirable. Motion planning is necessary to ensure that landmarks are repeatedly visited and also that the entire environment is covered as efficiently as possible.

Using a camera Sensor Network for SLAM rather than robot mounted sensors makes the robot motion planning problem even more critical. The robot travels over potentially large distances between cameras, using only odometry to track its position, so error will accumulate unchecked unless corrections are made using the

camera measurements. Camera positions serve as the landmarks in the map, so measurements can only be made when the robot moves its calibration target into the field of view of one of the cameras. This must be made explicit in choosing robot paths for our system. By selecting robot trajectories intelligently, we can produce a map more efficiently and more accurately, ensuring that camera measurements are available to correct the robot's position estimate when necessary and that each camera in the network is visited by the robot. The remainder of this chapter will attempt to achieve two goals for exploration trajectory planning: the robot should cover the entire network with minimal distance traveled; and the estimated map should have minimal uncertainty. Previous work discussed in Section 2.4 attempts to achieve similar goals for different sensing modalities through greedy planning or restrictive assumptions about the families of allowable paths. We desire a more general approach which increases efficiency of planning while still considering both distance traveled and map uncertainty.

During exploration, the robot has a partially constructed map of the environment. This map initially contains only the origin position and information is added each time the robot passes into the field of view of a previously unexplored camera. In this scenario, we can partition the exploration planning problem into two distinct sub-problems: choosing the order with which the cameras are explored, and choosing paths through the existing map. The nature of exploration is that we initially have no information about unexplored regions of the environment. This means that planning to optimize goals such as resulting uncertainty are impossible without making

prior assumptions about the nature of the environment. For this reason, the algorithms in this chapter will focus on the second sub-problem, planning paths through the existing map. We will continue to consider the full exploration problem by considering its complexity in Section 5.1.1 and also through an experimental evaluation where hand-crafted solutions are used for initial exploration of unknown regions, as will be discussed further in Section 5.4.3.

Several heuristic planning methods are examined which enable autonomous exploration of a camera network and perform well with respect to the distance and uncertainty criteria. We do not focus on optimal solutions, but instead on those which are computationally tractable and which produce favorable results in practice (optimality is discussed in Section 5.1.1). Several heuristic methods are developed which explicitly balance the inherent distance and uncertainty tradeoff outlined in Section 5.1.2. A family of static trajectories are presented as an initial investigation in Section 5.2 and a more adaptive strategy presented in Section 5.3. A variation of heuristic search which combines distance traveled and uncertainty in the cost function is able to provide flexible planning and exploit structure in the EKF estimate in order to achieve improved performance while keeping computation cost manageable. Simulation results given in Section 5.4 demonstrate the performance of each technique on a variety of randomly generated environments.

5.1 Challenges in Global Planning

This section will discuss the challenges in computing paths which achieve the goals outlined previously: minimal distance traveled and minimal map uncertainty. First, we will give analysis supporting the computation of an optimal solution being

exponentially complex, which means it is intractable except for small environments. Next, we will discuss the conflict inherent between our two goal criteria. This conflict makes producing reasonable results challenging even through the use of hand-created trajectories.

5.1.1 Complexity Analysis of the Exploration Planning Problem

Finding the optimal distance or minimum uncertainty path for exploring a camera network[1] appears to be computationally difficult and we speculate that it is NP-complete. This is not proven formally here for the general case. Instead, we give several intuitive arguments and demonstrate the NP-completeness of a special case. The candidate solutions to the problem of exploring a network are all covering walks through the graph which represents the network (each node in the graph represents a camera and edges indicate navigability). The number of solutions of this type is exponential in the length of the path considered. Of course, this does not guarantee that finding the optimal path requires exponential computation, since it may be found without considering every possible candidate solution.

When only distance is minimized, the problem becomes finding the minimum distance covering walk (MCW) through the graph. This problem is similar to a known NP-complete problem, minimum Hamiltonian Path [19], which has the same formulation except nodes may not be revisited. In fact, the problems are equivalent

[1] For the theoretical analysis of this section, a rough estimate of the entire graph is assumed to be known *a priori*. While this is not completely faithful to the problem faced by a mobile robot, it represents the "Map Verification Problem" and provides a lower bound for exploration of a completely unknown environment.

in the special case of fully connected graphs where the triangle inequality holds, since the MCW is guaranteed not to repeat any nodes in this case.

When uncertainty is considered as part of the cost function, the problem is intuitively harder than for distance only, since this means we must minimize a multivariate function whose values require iterations of the Kalman filter to compute. A formal proof of the NP-completeness of either minimal distance or minimal uncertainty exploration is beyond the scope of this thesis, but is an interesting point to consider for future work.

5.1.2 Distance and Uncertainty Trade-off

If the claims of the previous section hold, then computing optimal paths for our global exploration problem is intractable. Instead, we will seek heuristic trajectories which enable exploration in practice with low, though not minimal, distance and uncertainty. Unfortunately, as mentioned at the outset of this chapter, these two goals are often in conflict. While performing the mapping process, the robot has a partially constructed map, and must travel into previously unvisited territory in order to add new cameras to this map. If the robot was to continually move into unexplored regions, it would be able to cover its environment quickly at the cost of accumulating a large amount of uncorrected odometry error which would lead to inaccurate estimates of camera locations. The robot can slow this error buildup by periodically returning to regions of the map that have already been visited so its position can be corrected by camera measurements. This behavior will be referred to as "relocalizing". Relocalization will allow for mapping with lower uncertainty, but will require the robot to travel farther in order to cover the space. This describes

a trade-off which will be present in any exploration system; covering the space with minimal distance traveled and mapping with minimal uncertainty are conflicting goals.

In view of this conflict, we would like to produce paths which represent a compromise between our two goals. In order to allow automated algorithms to represent each goal in a common framework, we can combine them into a single cost function. Unfortunately, the two are incommensurable, that is, they lack common units for comparison, so care must be taken in combining their values. Several authors have previously proposed an *ad hoc* weighted linear combination, see for example Makarenko *et al.* [31]. Also, we must simplify the multi-dimensional covariance matrix into a scalar quantity which correctly represents entropy. As previously mentioned in Section 2.4.3, for the Gaussian distribution used by the EKF, either the trace or determinant of the covariance matrix can be employed (with Sim *et al.* [52] recently demonstrating that the trace is favorable). The general form of such a cost function is:

$$C(p) \quad = \quad \omega_d \; length(p) + \omega_u \; trace(P(p)) \tag{5.1}$$

where P is the EKF covariance matrix, C is the cost function evaluated over a particular path, p. Choosing weighting factors ω_d and ω_u for the contribution of distance and uncertainty must inherently involve one free parameter which represents the compromise between the two values. We attempt to make the choice of this parameter as intuitive as possible by normalizing the contribution of each quantity by

a rough estimate of its maximum possible value. For example, summing the weights in a spanning tree of the network provides an estimate for the maximum distance of a reasonable path, and the odometry error which would be accumulated by moving the robot from start to goal without correction from camera measurements estimates the maximum uncertainty. Once each quantity has been normalized, a single free parameter α in the range $[0, 1]$ is able to specify contribution of each factor to the sum. Based on this formulation, the weights used in our cost function are:

$$\begin{aligned} \omega_d &= \frac{\alpha}{maxdist} \\ \omega_u &= \frac{1-\alpha}{maxuncert} \end{aligned}$$

It is possible to consider only one of the factors at a time using this cost function. That is, setting $\alpha = 1$ makes C a function of distance only and will produce shortest path planning, while setting $\alpha = 0$ makes the C a function of uncertainty only.

5.2 Static Heuristic Exploration Trajectories

In view of the distance and uncertainty trade-off, we begin our analysis by considering several specific static policies that typify the behavioral extremes with respect to which most other mechanisms can be described. This section will describe these policies along with the rationale for their choice. The three static heuristic strategies considered were:

Depth-first exploration: the robot always moves into unexplored territory, never relocalizing. This strategy provides coverage of the environment with small

distance traveled, but the uncertainty of the robot position grows quickly. See Figure 5–1(a).

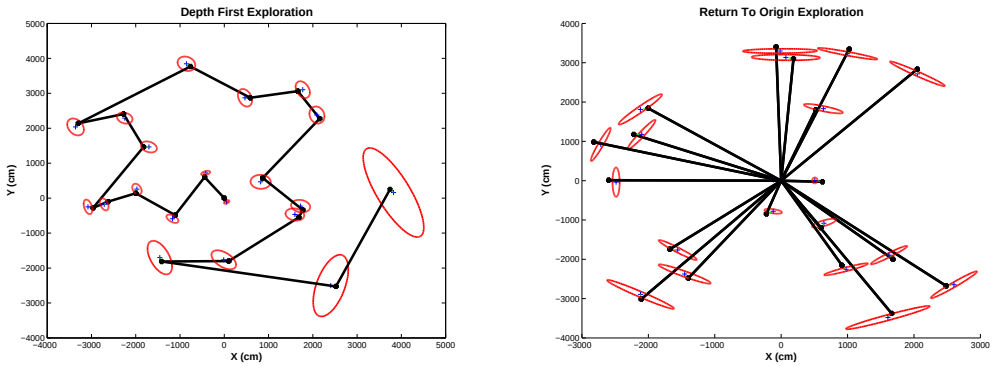
Return-to-Origin: the robot alternates between exploring a new camera position and returning to the first camera it mapped, which has the lowest uncertainty. This strategy allows for accurate relocalization, but means the robot must travel a large distance between each newly explored camera. See Figure 5–1(b).

Return-to-Nearest: in a compromise between the two previous methods, the robot alternates between exploring a new camera position and relocalizing at the nearest previously explored camera. For example, Figure 5–2 shows several stages of the exploration process. The ability to relocalize accurately depends on the uncertainty of the nearest camera only, which might not be mapped as accurately as cameras which are farther from the robot. Only regressing by one camera at a time means the extra distance traveled remains small.

These three strategies do not capture the full range of possibilities, and they provide neither flexibility nor adaptation to the environment. Instead, they are presented as an initial study of the effects of different strategies on the mapping process and as a baseline for comparison with more informed strategies.

5.3 Adaptive Heuristic Exploration

The static trajectories introduced in the previous section allow for exploration with various degrees of relocalization and illustrate the distance and uncertainty spectrum; however, always following the same relocalization strategy appears to be sub-optimal given the robot's uncertainty and that of the surrounding cameras will



(a) (b)

Figure 5–1: Two of the static heuristic exploration strategies. (a) The map estimate after exploration using the **Depth-first** strategy. (b) The map estimate after exploration using the **Return-to-Origin** strategy. For both methods, solid lines show the robot's path and 3σ uncertainty ellipses represent the uncertainty in the final estimate.

vary between each step in exploration. A more adaptive strategy which chooses relocalization paths based on the filter state at each step is more able to take advantage of problem structure and geometry. The challenge for such a system is that the space of possible paths is extremely large, so considering each option is intractable. We require an algorithm which is able to return a reasonable result using a limited amount of computation. The pruned search of Sim and Roy [52] is an example of a method which meets this requirement although the search proposed considers only map uncertainty and not distance traveled. We would like to extend this method by also keeping the distance traveled to a minimum, and providing increased search efficiency. The cost function given in Equation 5.1 provides a method for considering both distance and uncertainty. Achieving more efficient search depends on being

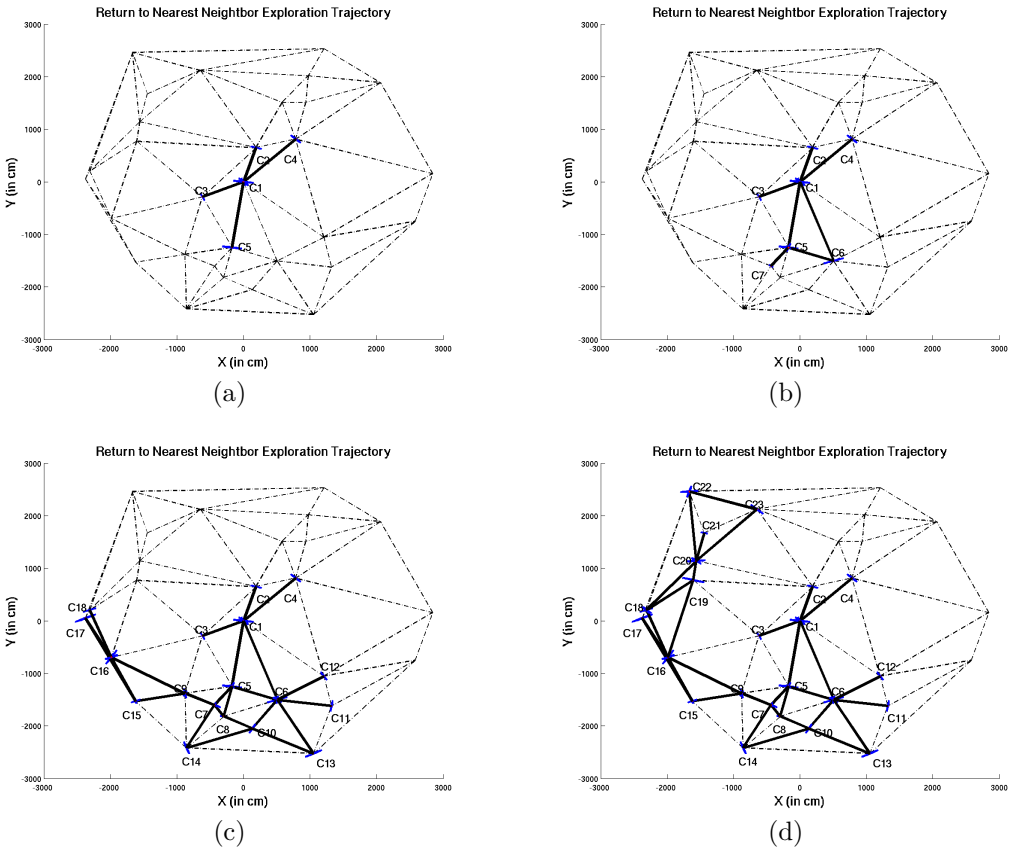


Figure 5–2: **Return-to-Nearest** exploration strategy progress while exploring a random graph at four intermediate steps progressing left to right and top to bottom. Dotted edges have not been followed by the robot while solid edges have been traversed once or more. Labels Ci indicate the node exploration ordering with $C1$ the starting position of the robot.

able to consider a further reduced subset of paths, or to search in a more favorable order.

A solution from classical AI planning that achieves these goals is known as informed search. Informed or best-first search uses a heuristic[2] function which can be evaluated at each node in order to estimate the remaining cost to reach the goal. The information provided by the heuristic function allows for searching through only a portion of possible paths, while still making guarantees about quality of the result. A^* search is an example of informed search which is able to return optimal paths as efficiently as possible given some restrictions on the heuristic function provided. Here, efficient means that the fewest possible nodes have been considered for any method that guarantees the final path is optimal. The remainder of this section will describe the application of heuristic search to the goals of distance and uncertainty planning.

5.3.1 Heuristic Search For Distance and Uncertainty Planning

A^* search requires two pieces of information about each node during its search: $f(n)$, the cost of the best path found so far from the start to node n; and $h(n)$, the heuristic function, which is an estimate of the cost from node n to the goal based on some (hopefully cheap) approximation. Nodes are explored in terms of increasing expected cost C which is a combination of the two terms:

[2] The term heuristic is used here in a different context than it was earlier for heuristic trajectories. In this case, it is a standard term used in the planning literature to describe an approximated cost.

$$C(n) \quad = \quad f(n) + h(n) \tag{5.2}$$

In order to apply A^* to distance and uncertainty planning, we must define the functions f and h. The f function is simply the combined distance and uncertainty cost function: Equation 5.1. The h function is less straightforeward to identify. In order for the guarantees of A^* to hold, this function must be "admissible". That is, it must be a reasonably tight underestimate for the remaining cost that will be required to reach the goal. This is a difficult quantity to compute because we are planning through a partially explored camera network where the uncertainty resulting from visiting a node depends on the existing map and the odometry built-up by the robot along the path. The odometry error accumulated while traveling on the shortest path to the goal without taking measurements from any cameras would provide an **upper** bound. We require the opposite, an underestimate, which demands that we consider uncertainty reduction that takes place when camera measurements are made, in which case there are many complex interactions that preclude any solution method other than exhaustive search. In this case, computing any type of reasonable bound appears to be an exponential problem.

Since we are unable to define an underestimate of uncertainty which is computationally tractable, we set the uncertainty contribution for h to zero. In the general case, A^* using a completely uninformed heuristic ($h(n) = 0 \ \forall \ n$) leads to breadth-first, or uniform-cost, search as only the f function contributes to the cost. In our case, only the uncertainty contribution in our h function is set to zero; the

straight-line distance to the goal remains to guide the search. Since our modified heuristic may not be admissible, the strict properties of A^* search are not met. So we will refer to the resulting search algorithm as heuristic search, or adaptive heuristic search in the following discussion.

It is important to note that this heuristic search is only approximating optimal paths with respect to the chosen cost function because the situation we consider violates the standard assumptions in two aspects. First, our cost formulation allows for loops through the network which reduce uncertainty, and thus have negative costs. Such so-called "negative weight cycles" imply that optimal path has infinite length. To maintain tractable planning, all cycles are pruned in our heuristic search, which is an approximation that prevents optimality. Second, our cost function uses a scalar representation of the EKF covariance matrix and uncertainty in subsequent planning steps cannot be predicted based on such a simple value. In other words, the $trace(P)$ measure used by our planner is not a sufficient statistic which is required for optimal planning. This lack of optimality is unavoidable based on the intuition that the the problem is inherently exponentially complex, but is noted in order to provide a more complete understanding and to provide directions for future work. It is also important to consider the potential for generalization of the method. Since our analysis does not explicitly depend on any properties of the camera sensors used, it could easily be applied to other graph-based exploration problems.

Figure 5–3 shows paths generated for four values of the α parameter which weighs the contributions of distance and uncertainty and illustrates that heuristic search is able to adapt to the environment and plan for both factors. In this example

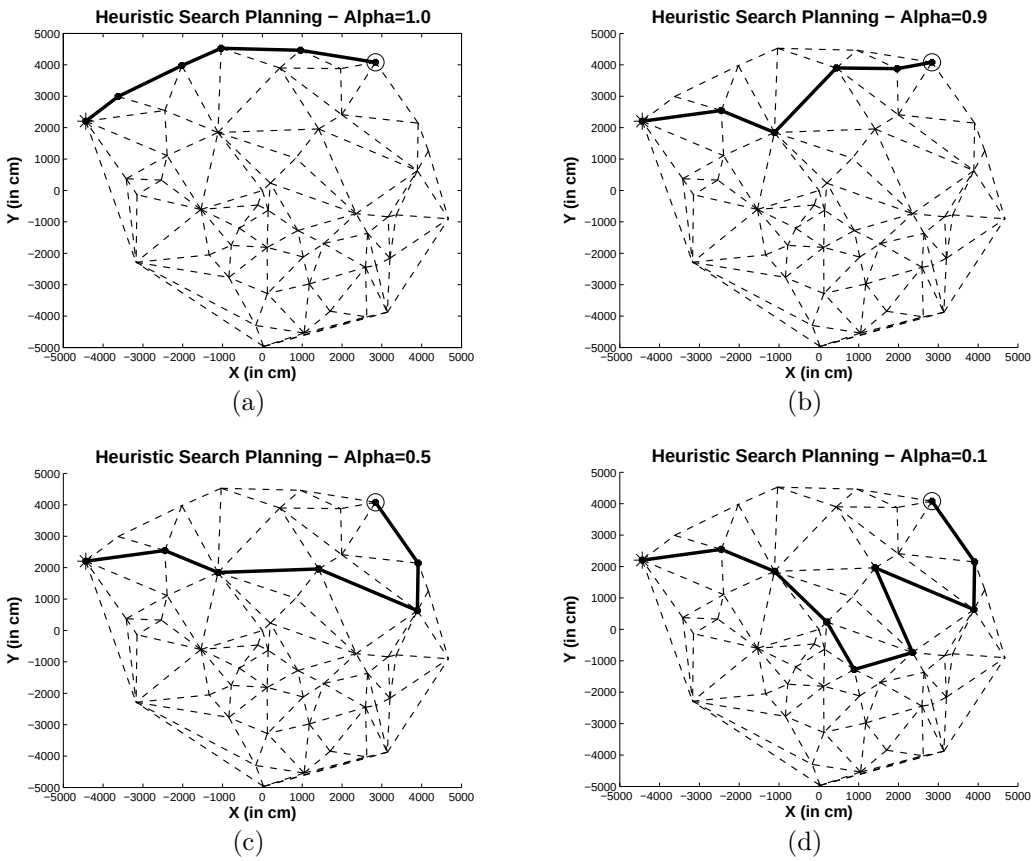


Figure 5–3: Paths generated by adaptive heuristic search using a distance and uncertainty cost function for four values of α. Dark lines indicate the path followed by the robot in each case.

environment, the robot has begun the mapping process at the centre of the graph, and has explored the nearby nodes first. This leads to relatively accurate estimates of the central camera locations compared to those on the periphery of the graph. The paths chosen by the robot are intuitive in that they cover longer distances in order to arrive at these central cameras as α decreases, and uncertainty is weighed more heavily. The next section provides results from further experiments which will evaluate the impact of such adaptation on global exploration.

5.4 Simulated Exploration Results

A simulation environment was developed in order to verify the performance of each of the heuristic exploration planners detailed earlier in this chapter. This environment was meant to emulate the properties of camera networks such as the one used in the hallway experiments described in Section 3.3 as closely as possible. To accomplish this, nodes were chosen from a uniform distribution over free space with approximately the same density as the hallway setup, the camera heights and distances from the robot location were taken as the averages of the hallway tests, and the same EKF implementation was used for state estimation as was used to produce Figure 3–7. Various graph sizes were produced by altering the number of nodes so that trends in the exploration results could be examined. Three families of edge densities were developed to ensure generalization of the results. In the remainder of this chapter, results have either been computed over a mixture of the graph types, or the results for each were so similar that results from a single class are presented as representative. Figure 5–4 shows sample graphs from each of these families. The graphs considered were:

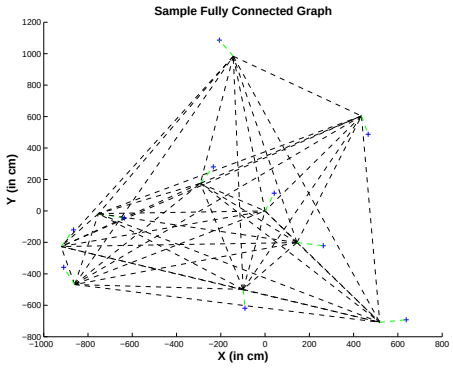
Fully connected graphs: Every pair of nodes is connected by an edge. This represents a scenario where the robot is able to freely traverse the environment without obstacles, and can move directly between any two nodes in the network. See Figure 5–4(a).

Triangulated graphs: Edges are chosen by triangulation of the nodes which produces a planar graph. This again represents obstacle free space, but in this case assumes the robot should move through a sequence of nodes along its path instead of traveling directly to the destination. See Figure 5–4(b).

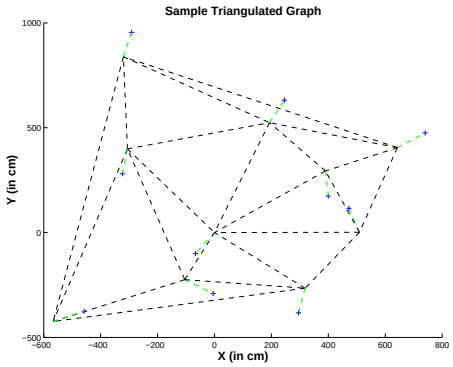
Hospital graphs: The floorplan layout of a real hospital building is used to define obstacles and free space. Nodes are distributed in the free space and edges are chosen by triangulation; however, only edges which do not pass through an obstacle are accepted. The presence of realistic obstacles in this environment makes this family of graphs similar to our real world situation. See Figure 5–4(c).

5.4.1 Static Trajectory Results

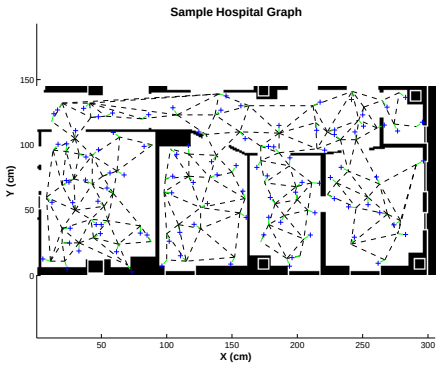
The three static heuristic strategies were each executed on a large set of random graphs from each of the three edge density families and for sizes ranging from ten to fifty camera nodes. Since the **Depth-first** strategy does not perform relocalization, uncertainty accumulates quickly for this method. The two relocalization strategies both limit error buildup more effectively, with **Return-to-Nearest** somewhat surprisingly producing lower uncertainty maps. The larger distance traveled for the **Return-to-Origin** strategy does not produce significant corresponding gain in accuracy. One reason for this effect is likely that nearby landmarks in an EKF have



(a)



(b)



(c)

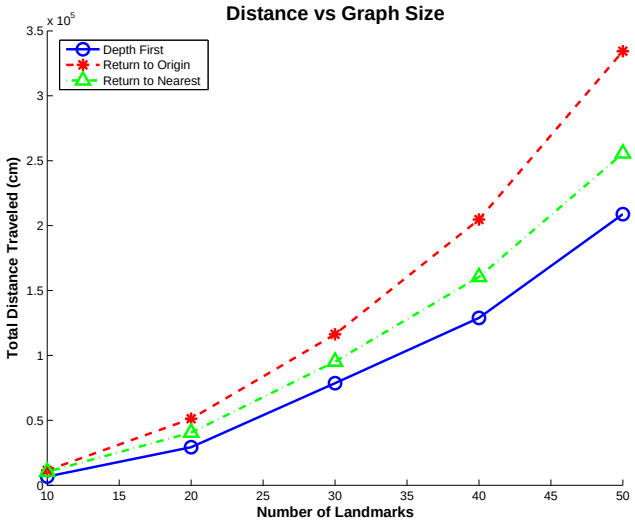
Figure 5–4: The three different families of graphs considered for simulation results. Dotted lines indicate edges between nodes. Camera positions are shown as blue crosses where colour is available.

much more effect on each other than those which are far apart. So, relocalizing at the origin has minimal impact on the estimation accuracy at distant cameras. In addition, upon inspection and analysis of the map estimates produced, **Return-to-Origin** produces a star-like exploration pattern (for example, Figure 5–1(b)), always following rays from the origin of the graph to the next node explored. This produces estimated distributions with the direction of maximum uncertainty perpendicular to the ray to the origin. If all cameras along a path have a similar direction of maximum uncertainty, there is little reduction in entropy possible when subsequent cameras are visited. This illustrates a fundamental limitation of the static heuristics; they do not plan adaptively in order to consider the problem geometry.

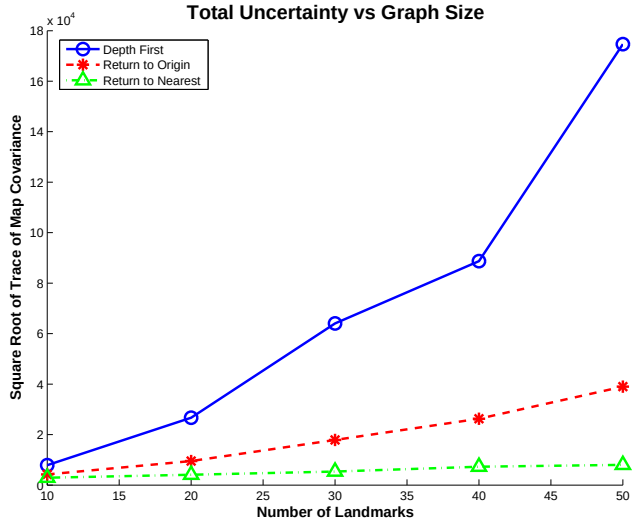
5.4.2 Single Path Results for the Adaptive Heuristic

The goal of our adaptive heuristic planner is to choose a short-term path through the known graph that allows the robot to arrive at a new node with minimal distance and uncertainty. In order to analyze the performance of our algorithm, it was executed in the simulation environment described earlier. For each randomly generated network, the **Return-to-Nearest** strategy was executed for a portion of the network in order to initialize camera estimates in the EKF. A series of planners were then executed to find paths between constant start and goal nodes so that paths returned could be compared. The planners evaluated were heuristic search with four different values for α: $(0.1, 0.5, 0.9, 1.0)$. Note that $\alpha = 1.0$ produces shortest distance planning.

Figure 5–6(a) illustrates the distance traveled for the four choices of α, over twenty instances for each network size. As expected, larger α values produce shorter



(a)

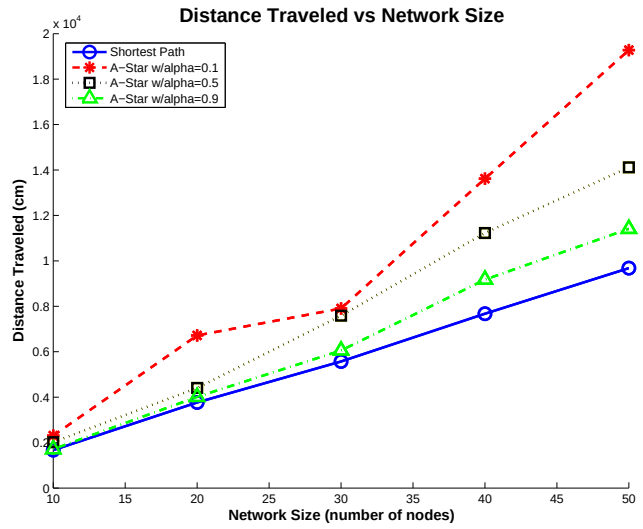


(b)

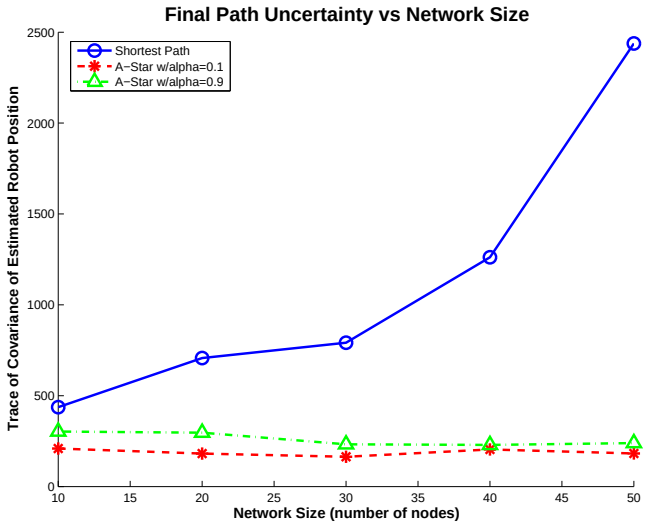
Figure 5–5: (a) Distance accumulated during mapping for the three static heuristic strategies. (b) Final map uncertainty after mapping for the three static heuristic strategies.

path lengths since distance is weighed more heavily in the cost function. The relatively graceful increase in distance traveled as α decreases indicates the ease at which the planners are able to find slightly longer paths that perform better with respect to final uncertainty. That is, there is no catastrophic degradation in distance performance as the weighting is changed.

Figure 5–6(b) shows the final robot uncertainty upon arrival at the chosen goal node for three of the four α values used. When examining these results, it is important to note that a relatively complicated exploration procedure has proceded heuristic search planning, so while the general trends are of interest, explanation of the precise behaviours is somewhat complicated. For example, the fact that the uncertainty decreases with graph size for several values of α is a product of the relation between graph density, frequency of camera visitation at various distances from the center, and the exploration and planning algorithms employed. Results for $\alpha = 0.5$ are excluded because the results for this method lie extremely close to $\alpha = 0.1$ and $\alpha = 0.9$, which makes visualization difficult. Setting $\alpha < 1$ manages to reduce the uncertainty drastically over the $\alpha = 1$ case, where only distance is considered. In fact, the improvement in uncertainty between $\alpha = 1$ and $\alpha = 0.9$ is much larger than that between $\alpha = 0.9$ and $\alpha = 0.1$. Investigation of the paths returned indicates that there is often one camera near the shortest path which has been mapped accurately, in some cases the origin node, where good localization can be obtained. Once this camera is included in the path, the uncertainty is significantly lowered. Continuing to lower the uncertainty beyond this value becomes more difficult, so decreasing α



(a)



(b)

Figure 5–6: (a) The distance required to reach a goal node in a partially explored graph is shortest with $\alpha = 1$ representing shortest path planning and increases as α is decreased. (b) The uncertainty with with the robot reaches the goal node in a partially explored graph is largest with $\alpha = 1$ representing shortest path planning and decreases as α is decreased.

does not reduce final uncertainty as effectively. This is potentially a useful insight for future work.

5.4.3 Global Exploration Results

The previous section clearly indicates that the adaptive localization heuristic is able to produce relocalization paths with intuitively favorable and adaptive properties. This method will be extended to global exploration, which requires iterating two steps:

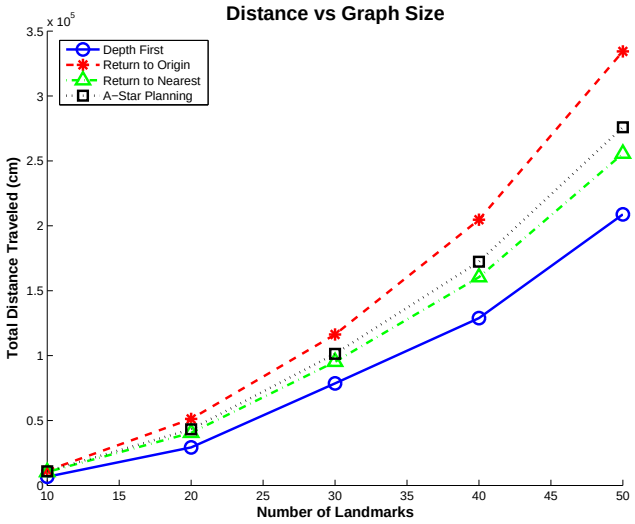
- Selecting an edge from an explored node leading into unexplored territory in order to visit a new camera for the first time.
- Selecting a path through the known graph ending at the next destination node for exploration.

As mentioned previously, for the first step, we are not able to compute expected distance or uncertainty during the initial exploration of a new camera position. Under this condition, a hand-crafted strategy similar to breadth-first search, which explores nodes closest to the origin first, is used for each method compared. A more detailed analysis of the effects of such a strategy as well as development of more sophisticated strategies is an interesting problem for future work. Solutions for the second step have previously been suggested by each of the static heuristics. We will propose a new exploration strategy here based on the adaptive localization heuristic. That is, we execute heuristic search to compute the path which achieves minimal distance traveled and uncertainty at each sub-step of mapping, and the robot follows the path returned. In this way, the adaptive localization strategy is extended to adaptive global exploration.

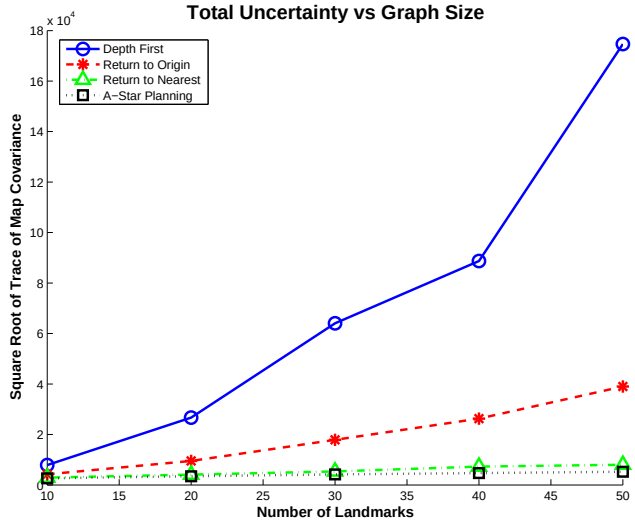
Figure 5–7(a) illustrates the total distances traveled by each of the methods. The results are not surprising; the **Depth-first** strategy covers the environment with the least robot motion, **Return-to-Nearest** requires slightly more motion, the adaptive heuristic slightly more again, and **Return-to-Origin** requires the largest distance traveled. Figure 5–7(b) presents the final map uncertainty results. The adaptive global strategy is able to produce maps with lower uncertainty than any of the static methods due to the fact that it uses all of the information available in order to choose paths which exploit properties of the current estimate.

It is interesting to note that the improvements in uncertainty made by the adaptive exploration strategy over the **Return-to-Nearest** strategy are much less pronounced than those that were made by the adaptive heuristic over shortest distance planning for sub-paths. Upon examination of the global trajectories produced by each method, it is apparent that the entropy reduction made by the adaptive planner results mainly from exploiting the most accurate camera or set of cameras in a region. This behaviour is able to ensure that each sub-path ends with low uncertainty; however, it is not ideal for reducing the uncertainty in the less accurate cameras in the map. That is, these estimates are never corrected and remain quite inaccurate. It may be necessary to add a term in our cost function which allows us to jointly consider final robot path uncertainty as well as reduction in uncertainty in previously mapped nodes in order to achieve further reductions in uncertainty during exploration. This will be left for future work. Even before these potential

improvements, exploration planning using heuristic search based on distance and uncertainty for relocalization is an adaptive process which exploits problem geometry to find paths which effectively reduce uncertainty in the robot's pose estimate.



(a)



(b)

Figure 5–7: (a) The total distance required to explore the environment for each of the strategies considered. (b) The final map uncertainty for each of the strategies considered.

CHAPTER 6
Conclusions

This thesis has presented a method for a hybrid robot/camera system which allows for profitable collaboration between the two systems. Chapter 3 presented an **automated** calibration and mapping framework which is capable of computing a map of the locations of the cameras in an environment along with calibration information for each of these cameras. The use of a robot-mounted ARTag target allows a single camera to be calibrated quickly and accurately, without user interaction. This technique extends the state of the art since vision researchers currently perform calibration by manually collecting images of the calibration target and clicking on grid corners, which is a time consuming and error prone process. Once cameras are calibrated, the metric measurements they provide are combined with the robot's odometry estimates to build a map of the camera locations and localize the robot within that map. This is accomplished using an Extended Kalman filter, which estimates 3-D position and orientation of the camera poses. The performance of the mapping system was verified with experiments in a large hallway environment, which showed even a sparse distribution of cameras could be used for relatively successful navigation, but suggested the use of other types of sensing or more sophisticated inference mechanisms in order to further improve performance.

Adaptive robot motion was investigated in several aspects of our system. Chapter 4 examined robot motion planning while in front of a single camera through

analyzing the dependence of calibration accuracy on parameters in the image set used for calibration. Heuristic local trajectories showed that accurate calibration could be achieved using simple robot motions. In order to obtain a more sophisticated planning method, two image sets were collected and used to study the effect of image set geometry on calibration accuracy. Results from numerous repetitions of calibration with these images showed that maximizing difference in angle between successive frames and moving the target as close to the camera as possible produced the most accurate calibrations. Chapter 5 studied global motion planning between the cameras with the goals of reducing map uncertainty with as little robot motion as possible. While these goals are conflicting and optimal planning likely intractable, heuristic solutions were shown to improve results over naive methods. Adaptive heuristics for local paths were able to vary behaviour based on the current state of the EKF and produce favorable results.

In this work, we propose the use of a 6-DOF EKF for global mapping. While this approach worked quite well even in a large environment, there are several indications that a more sophisticated mapping method would be preferable. The environment has large regions without cameras to provide observations, so filtering alone will not be able to correct entirely for the odometric error accumulated in these areas. It will likely be preferable to adopt a filtering and smoothing method which will allow for better correction of paths in regions with few observations. Also, since we expect to build large odometry errors before seeing a camera, it is expected that the linearization procedure, which is only a good approximation when errors are small, will be highly inaccurate. This effect will be seen increasingly as cameras are spaced

farther apart in the environment. A probabilistic method such as Particle Filtering might give improved results in this context, since linearization is not necessary for such a technique. In addition, the roll, pitch, yaw representation of angles used in our state vector is intrinsically discontinuous, which leads to special cases in the equations which must be dealt with carefully. The use of quaternions to represent angles is a technique which could be adopted to improve our state estimation.

The combination of the camera measurements we study with a SLAM solution based on dense sensor readings has the potential to produce interesting results. Dense SLAM approaches can often be seen to lose global alignment over large regions. Solutions to this have been forcing the robot to close loops during mapping which helps correct orientation, or to perform expensive post-processing of the data. Our method provides an economical approach to allow the correction of error in the robot orientation, because a measurement from the camera provides a second level of sensing and can be processed much more cheaply than post-processing all of the dense range data. Moreover, because of the information inherent in each camera, there is no data-association problem.

The principles of our technique can be applied in numerous other environments. For example, underwater robots are typically unable to use laser range sensing and must depend on much less reliable sonar or stereo vision systems in order to perform traditional mapping. Placing engineered targets such as ARTag markers underwater in the robot's environment could allow for much more accurate mapping in this case. Also, in a multi-robot scenario, the use of engineered targets attached to each robot would allow each robot to be quickly detected, have its identity recovered without

ambiguity and the robots' relative locations measured quickly and easily. Achieving each of these tasks through sensing would allow the robots to collaborate without excessive communication, which is a desireable property for an underwater robot team.

There are several aspects of our method as presented which could be made more principled for the purposes of academic study. In choosing local calibration trajectories, we focussed on intuition and empirical study due to the fact that analysis of the precise properties of the nonlinear optimization is difficult and time consuming. In order to produce an analytically optimal calibration strategy, this work should be pursued. Also, in terms of global exploration planning, we made some intuitive claims about the exponential complexity of minimum uncertainty mapping. At least in simple cases, it should be possible to continue this work to produce a rigorous proof of the NP-hardness of the problem. Neither of these aspects has interferred with the deployment of an autonomous calibration and mapping system, but they would have been desireable results had time permitted.

Perhaps the most promising area of future work related to this thesis is the use of adaptive heuristic trajectory planning in this and other localization and mapping domains. As discussed in Section 2.4, the state of the art in planning for reduction of map uncertainty consists of many greedy planners based on entropy reduction techniques. By nature greedy planning is far from optimal, given that it does not attempt to exploit all of the information available. The adaptive heuristic search method described in Chapter 5 provides for greater adaptation, while managing to limit computation through the use of a heuristic function to guide the search for

solutions. The author believes similar methods can be applied in other mapping domains such as the landmark based EKF, occupancy grid representations such as FastSLAM, and in fact any representation where uncertainty is explicitly modeled in the estimator. The use of such adaptive heuristics will allow robotic mapping to occur with lower error and contribute to the autonomy of robotic agents in general.

REFERENCES

[1] The opencv computer vision library. *http://www.intel.com/research/mrl/ research/opencv*.

[2] F. Amigoni and A. Gallo. A multi-objective exploration strategy for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3861–3866, Barcelona, Spain, 2005.

[3] R. Baeza-Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Information and Computation*, 106:234 – 252, 1993.

[4] M. Batalin, G. Sukhatme, and M. Mattig. Mobile robot navigation using a sensor network. *International Confernce on Robotics and Automation*, 2003.

[5] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte. Information based adaptive robotic exploration. *International Conference on Intelligent Robots and Systems*, 2002.

[6] Howie Choset and Joel Burdick. Sensor based planning, part ii: Incremental construction of the generalized voronoi graph. In *Proc. of IEEE Conference on Robotics and Automation*, pages 1643 – 1648, Nagoya, Japan, May 1995. IEEE Press.

[7] J. J. Craig. *Introduction to Robotics, Mechanics and Control*. Addison-Wesley, 1986.

[8] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes. Robotic exploration as graph construction. *Transactions on Robotics and Automation*, 7(6):859–865, December 1991.

[9] T.J. Ellis, D. Makris, and J. Black. Learning a multicamera topology. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 165–171, Nice, France, October 2003.

[10] D. Estrin, D. Culler, K. Pister, and G. Sukatme. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1(1):59–69, January-March 2002.

[11] O. Faugeras and Q. Luong. *The Geometry of Multiple Images*. The MIT Press, 2001.

[12] O. D. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.

[13] H. Feder, J. Leonard, and C. Smith. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research, Special Issue on Field and Service Robotics*, 1999.

[14] M. Fiala. Artag revision 1, a fiducial marker system using digital techniques. In *National Research Council Publication 47419/ERB-1117*, November 2004.

[15] M. Fiala. Vision guided robots. In *Proc. of CRV'04 (Canadian Confernence on Computer and Robot Vision*, pages 241–246, May 2004.

[16] M. Fiala and C. Shu. Fully automatic camera calibration using self-identifying calibration targets: Nrc publication number: Nrc 48306. 2005.

[17] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 1998.

[18] S. Funiak, C. Guestrin, Mark Paskin, and Rahul Sukthankar. Distributed localization of networked cameras. In *Proceedings of Information Processing in Sensor Networks*, Nashville, TN, USA, 2006.

[19] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1991.

[20] W. E. L. Grimson, C. Stauer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 22–29, 1998.

[21] S. Hang, N. Kwok, G. Dissanayake, Q. Ha, and G. Fang. Multi-step look-ahead trajectory planning in slam: Possibility and necessity. *International Conference on Robotics and Automation*, 2005.

[22] Andrew Howard, Maja J Mataric, and Gaurav S. Sukhatme. Localization for mobile robot teams using maximum likelihood estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 434–459, EPFL Switzerland, 2002.

[23] O. Javed, Z. Rasheed, O. Alatas, and M. Shah. Knight: a real time surveillance system for multiple and non-overlapping cameras. *The fourth International Conference on Multimedia and Expo (ICME 2003)*, 2003.

[24] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[25] S. Koenig and Y. Smirnov. Graph learning with a nearest neighbor approach. In *Proceedings of the Ninth Annual ACM Conference on Computational Learning Theory (COLT)*, pages 19–28, 1996.

[26] T. Kollar and N. Roy. Using reinforcement learning to improve exploration trajectories for error minimization. In *International Conference on Robotics and Automation*, pages 3338–3343, Orlando, Florida, 2006.

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

[28] R. Kurazume and S. Hirose. Study on cooperative positioning system - optimum moving strategies for cps-iii. In IEEE, editor, *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 4, pages 2896–2903, 1998.

[29] John J. Leonard and Hugh F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.

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

[31] A. Makarenko, S. Williams, F. Bourgault, and H. Durrant-Whyte. An experiment in integrated exploration. *International Conference on Intelligent Robots and Systems*, 2002.

[32] Dimitris Marinakis, Gregory Dudek, and David Fleet. Learning sensor network topology through monte carlo expectation maximization. In *Proc. of the*

IEEE International Conference on Robotics & Automation, pages 4581–4587, Barcelona, Spain, Apr. 2005.

[33] P. Maybeck. *Stochastic Models, Estimation ond Control*, volume 1. Academic, New York, 1979.

[34] J. C. McGlone, editor. *Manual of Photogrammetry*. American Society of Photogrammetry, 5th edition, 2004.

[35] David Meger, Ioannis Rekleitis, and Gregory Dudek. Autonomous mobile robot mapping of a camera sensor network. In *The 8th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, pages 155–164, Minneapolis, Minnesota, July 2006.

[36] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the American Association for Artificial Intelligence (AAAI)*, 2002.

[37] H. Moravec and A. Elfes. High-resolution maps from wide-angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 116–121, St. Louis, MO, USA, 1985.

[38] L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision inertial self-tracker. In *In ISMAR: International Symposium on Mixed and Augmented Reality*, Darmstadt, Germany, 2002.

[39] I. Poupyrev, H. Kato, and M. Billinghurst. Artoolkit user manual, version 2.33. *Human Interface Technology Lab, University of Washington*, 2000.

[40] J. Rekimoto and Y. Ayatsuka. Cybercode: Designing augmented reality applications with visual tags. In *In Proceedings of DARE, Designing Augmented Reality Environments*, pages 1–10, 2000.

[41] Ioannis Rekleitis, David Meger, and Gregory Dudek. Simultaneous planning, localization, and mapping in a camera sensor network. *Robotics and Autonomous Systems*, page (in print), 2006.

[42] Ioannis M. Rekleitis. A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, 3480 University St., Montreal, Québec, CANADA H3A 2A7, 2004.

[43] Ioannis M. Rekleitis, Gregory Dudek, and Evangelos Milios. Multi-robot collaboration for robust exploration. In *Proceedings of International Conference in Robotics and Automation*, pages 3164–3169, San Francisco, USA, April 2000.

[44] Ioannis M. Rekleitis, Gregory Dudek, and Evangelos Milios. Multi-robot collaboration for robust exploration. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):7–40, 2001.

[45] Ioannis M. Rekleitis, Vida Dujmović, and Gregory Dudek. Efficient topological exploration. In *Proceedings of International Conference in Robotics and Automation*, pages 676–681, Detroit, USA, May 1999.

[46] Ioannis M. Rekletis and Gregory Dudek. Automated calibration of a camera sensor network. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 401–406, Edmonton Alberta, Canada, Aug. 2-6 2005.

[47] Stergios I. Roumeliotis and George A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, Octomber 2002.

[48] N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation - mobile robot navigation with uncertainty in dynamic environments. *International Conference on Robotics and Automation*, 1999.

[49] R. Sim. Stable exploration for bearings-only slam. *International Conference on Robotics and Automation*, 2005.

[50] R. Sim and G. Dudek. Examining exploratory trajectories for minimizing map uncertainty. *International Joint Conference on Artificial Intelligence*, 2003.

[51] R. Sim, G. Dudek, and N. Roy. Online control policy optimization for minimizing map uncertainty during exploration. *International Conference on Robotics and Automation*, 2004.

[52] R. Sim and N. Roy. Global a-optimal robot exploration in slam. In *International Conference on Robotics and Automation*, pages 661 – 666, 2005.

[53] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, pages 167 – 193, 1990.

[54] C. Stachniss, G. Grisetti, D. Haehnel, and W. Burgard. Improved rao-blackwellized mapping by adaptive sampling and active loop closure. *Workshop on Self-Organization of Adaptive Behavior (SOAVE) (invited presentation)*, 2004.

[55] C. Stachniss, D. Haehnel, and W. Burgard. Exploration with active loop-closing for fastslam. *International Conference on Intelligent Robots and Systems*, 2004.

[56] R. Y. Tsai. Synopsis of recent progress on camera calibration for 3-d machine vision. *The Robotics Review*, pages 147–159, 1989.

[57] R. Y. Tsai and R. K. Lenz. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, pages 323–344, 1987.

[58] R. Y. Tsai and R. K. Lenz. Real time versatile robotics hand/eye calibration using 3d machine vision. *IEEE International Conference on Robotics and Automation*, 1988.

[59] D. Woods, S. McNee, J. Davis, A. Morison, P. Maughan, and K. Christoffersen. Event template hierarchies as means for human-automation collaboration in security surveillance. In *Human Factors and Ergonomics Society Annual Meeting*, Orlando, FL, September 26-28 2005.

[60] B. Yamauchi. A frontier based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, 1997.

[61] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.