University of Alberta

A Probabilistic Camera Sensor Model for Bearing-only Visual SLAM

by

Jing Wu 

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Fall 2007

Your file Votre référence
ISBN: 978-0-494-33374-7
Our file Notre référence
ISBN: 978-0-494-33374-7

Learning without thought is labor lost; thought without learning is perilous.

– Confucius 551 BC - 479 BC

Abstract

In this thesis, we present our method for building a probabilistic camera sensor model for bearing-only visual SLAM. This is the first time that a probabilistic camera sensor model for bearing-only visual SLAM has been studied. Our focus is on modelling the uncertain component of a probabilistic camera sensor model, measurement noise, which represents the inaccuracy of the estimated sensor measurements from the sensor model. We conclude that a camera sensor model with constant measurement noise variance is sufficient for both close and distant landmarks. Simulation results show that our model performs well in visual localization as measured by the accumulated error. However, in a noisy environment, an accurate model of the measurement noise is not critical for the performance of visual localization as long as proper landmark position uncertainty is used.

To my parents,
who bore me, raised me, supported me, taught me and loved me.

Acknowledgements

It is a great pleasure to thank the many people who made this thesis possible.

I would like to express my deep and sincere gratitude to my supervisor, Professor Hong Zhang. Throughout my research in the Robotics Lab, he provided encouragement, sound advice, good teaching and insightfull criticism. Furthermore, thanks to Professor Martin Jagersand and Professor Vicky Zhao for being my thesis examiners.

I would like to thank my fellow colleagues for providing a stimulating and fun environment in which to learn and grow. I am especially grateful to Jonathan Klippenstein and Xiang Wang for detailed discussions and helpful suggestions. Thanks to Chris Parker for his advice on writing. Thanks to Artit Visatemongkolchai for his company during my graduate studies.

Finally, I wish to thank my good friends Jing Li, Jie Teng and Xiangming Wu, for all the emotional support and caring they have provided.

Table of Contents

1 Introduction **1**
1.1 Thesis Motivation . . . 1
1.2 Thesis Objectives and Contributions . . . 3
1.3 Overview . . . 3

2 Background **5**
2.1 SLAM . . . 5
2.2 Solutions to the SLAM Problem . . . 7
2.2.1 EKF-SLAM . . . 7
2.2.2 FastSLAM . . . 9
2.3 Bearing-only Visual SLAM . . . 11
2.4 Localization . . . 12
2.5 Probabilistic Camera Sensor Model . . . 13
2.6 Summary . . . 14

3 Model Development **15**
3.1 Probabilistic Camera Sensor Model . . . 15
3.2 Methodology . . . 17
3.3 Model Building . . . 18
3.3.1 Camera Calibration . . . 18
3.3.2 Reprojection Error . . . 19
3.4 Summary . . . 23

4 Experiments of Building a Camera Sensor Model **24**
4.1 Deterministic Component of a Camera Sensor Model . . . 24
4.2 Uncertain Component of a Camera Sensor Model . . . 24
4.2.1 Normality Test . . . 25
4.2.2 Sensitivity to Range . . . 28
4.2.3 Measurement Noise Covariance Matrix . . . 29
4.3 Summary . . . 32

5 Model Validation **33**
5.1 Experimental Setup . . . 33
5.1.1 Simulation and Experimental Environment . . . 33
5.1.2 Performance Evaluation . . . 36
5.2 Simulation Results . . . 36
5.3 Experimental Results . . . 44
5.4 Summary . . . 50

6 Conclusions **51**
6.1 Camera Sensor Model . . . 51
6.2 Limitations and Future Work . . . 52

Bibliography **54**

A Implementation of Particle Filter Based Localization **56**
A.1 Motion Model . . . 56
A.2 Transformation Between Reference Frames . . . 57
A.3 Innovation Covariance Matrix . . . 59

List of Tables

4.1 Intrinsic camera parameters and their standard deviations. f_x and f_y are the camera focal length in x and y directions respectively. cc_1 and cc_2 represent the principal point on the image plane. k_1 is the radial distortion scale factor of the image. . . . 24

4.2 The results of Geary's test on reprojection error samples at both x and y directions. The number of samples used is 500. The significance level is set to $\alpha = 0.05$. . . . 26

5.1 Results of the close landmark simulation for landmarks approximately one meter away from the camera. The table shows the average and standard deviation of the accumulated localization error for various sensor models. Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The $\star$ refers to data not shown in Figure 5.4. . 39

5.2 Results of the distant landmark simulation for landmarks approximately two meter away from the camera. The table shows the average and standard deviation of the accumulated localization error for various sensor models. Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The $\star$ refers to data not shown in Figure 5.5. . 40

5.3 Results of the close landmark experiment for landmarks approximately one meter away from the camera. The table shows the average and standard deviation of the accumulated localization error for various sensor models. Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$. The $\star$ refers to data not shown in Figure 5.9. . . . 46

5.4 Results of the distant landmark experiment for landmarks approximately two meter away from the camera. The table shows the average and standard deviation of the accumulated localization error for various sensor models. Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$. The $\star$ refers to data not shown in Figure 5.10. . . 47

List of Figures

1.1 The typical approach to solving SLAM is composed of a prediction step and an update step that are performed recursively to estimate both the robot pose and the map. In the prediction step, odometry data is used to predict the robot pose and the map. In the update step, measurements obtained from the sensors are used to update the prediction. 2

2.1 A graphical model of SLAM, describing the evolution of the odometry data $u_{k-1:k+1}$, the states $x_{k-1:k+2}$ and the measurements $z_{k-1:k+1}$. 6
2.2 Example of Extended Kalman Filter estimation of the map and the robot path. Landmarks are denoted by stars and the triangle represents the robot. All the ellipses show the uncertainties around the estimated landmark positions and robot poses respectively. 8
2.3 A pinhole camera model. A three-dimensional landmark is projected onto a two-dimensional image plane. Its intersection of the image plane is the image point of the three-dimensional landmark. 13

3.1 A flowchart of bearing-only visual SLAM. 16
3.2 A diagram of image points associated with a three-dimensional landmark X^W. x is the actual image point on the image plane, $\overline{x}$ is the measured image point and x^I is the estimated image point. 17
3.3 A checkerboard with the origin of the world reference frame is on the left hand corner. Z-axis of the frame is perpendicular to the checkerboard plane. 19
3.4 A flowchart describing the construction of a probabilistic sensor model for a camera [35]. 21
3.5 A camera mounted on an ActiveMedia Pioneer 3 robot facing a checkerboard. . . 22

4.1 The calibration pattern used in the experiments. Arrows represent the reprojection errors between the measured corners and the estimated ones. 25
4.2 Reprojection error standard deviation along both the horizontal and vertical directions. The reprojection error covariance matrix $\Sigma_{reproj1}$ is (0.279^2, 0.006; 0.006, 0.253^2). The landmarks are at a distance of 50 cm from the camera. 27
4.3 Reprojection error standard deviation along both the horizontal and vertical directions. The reprojection error covariance matrix $\Sigma_{reproj2}$ is (0.125^2, -0.001; -0.001, 0.120^2). The landmarks are at a distance of 130 cm from the camera. 27
4.4 Reprojection error standard deviation along both the horizontal and vertical directions. The reprojection error covariance matrix $\Sigma_{reproj3}$ is (0.098^2, 0.000; 0.000 0.106^2). The landmarks are at a distance of 210 cm from the camera. 28
4.5 The relationship between a camera and several images that are taken at a range of 50 cm in the camera coordinate frame. 29
4.6 Reprojection error standard deviation along the x and y directions for landmarks at various distances from the camera. 30
4.7 Plot of the measurement noise standard deviation with respect to various ranges between the camera and the landmarks. The measurement noise covariance matrix $R_k = diag(\delta u^2, \delta v^2)$. δu and δv represent the measurement noise standard deviation in the x and y directions respectively. 31

5.1 An ActiveMedia Pioneer 3 robot mounted with a camera mounted on the robot facing perpendicular to the robot's forward motion. 34

5.2 An ActiveMedia Pioneer 3 robot observes artificial landmarks on the wall from images taken using a mounted camera. Corners detected by the Harris Corner Detector are shown by cross marks. The numbers in the image refer to the landmarks identification number in the map. 35
5.3 A diagram of our ActiveMedia Pioneer 3 robot moving down the hallway of the Department of Computing Science. A camera is mounted on the robot and faces sideways to the wall. The robot observes artificial landmarks placed on the wall. The solid line indicates the robot's real path and the dotted line represents the odometry data obtained from the robot's wheel encoders. 35
5.4 Results of the close landmark simulation. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. 39
5.5 Results of the distant landmark simulation. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. 40
5.6 Comparison of accumulated localization errors between the distant and close landmarks in simulation. The dotted line shows the ratio between the accumulated error for the landmarks $\epsilon_{distant}$ and the accumulated error for the close landmarks ϵ_{close}. The straight line represents the average error ratio over all the sensor models using various levels of measurement noise standard deviation. 41
5.7 Localization simulation in a three-dimensional environment. The robot moves along a straight line and observes close landmarks about one meter away. The length of the robot's real path is approximately 15 meters. A pessimistic sensor model with the measurement noise covariance matrix $R_k = diag(100^2, 100^2)$ is used. The straight line represents the robot's true path, the curve line is the robot's estimated path based on the odometry data, and a set of 50 particles show the robot's estimated path based on the localization algorithm. 43
5.8 Localization simulation in a three-dimensional environment. The robot moves along a straight line and observes close landmarks about one meter away. The length of the robot's real path is approximately 15 meters. A pessimistic sensor model with the measurement noise covariance matrix $R_k = diag(0.1^2, 0.1^2)$ is used. The straight line represents the robot's true path, the curve line is the robot's estimated path based on the odometry data, and a set of 50 particles show the robot's estimated path based on the localization algorithm. 43
5.9 Results of the close landmark experiment. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$. 46
5.10 Results of the distant landmark experiment. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$. 47
5.11 Comparison of accumulated localization errors between the distant and close landmarks. The dotted line shows the ratio between the accumulated error for the distant landmarks $\epsilon_{distant}$ and the accumulated error for the close landmarks ϵ_{close}. The straight line represents the average error ratio over all the sensor models using various levels of measurement noise standard deviation. 48
5.12 Results of the close landmark simulation. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$. 48

5.13 Results of the distant landmark simulation. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$. 49

A.1 Odometry motion model. The robot's motion from time step $k-1$ to k is approximated by translation ϕ_{trans}, rotations ϕ_{rot1} and ϕ_{rot2}. 57

A.2 The relationship between the world reference frame, the robot reference frame and the camera reference frame. R^{WR} and T^{WR} are the rotation and translation from the world coordinate frame to the robot reference frame. R^{RC} and T^{RC} are the rotation and translation from the robot reference frame to the camera reference frame. . . . 58

Chapter 1

Introduction

1.1 Thesis Motivation

If an autonomous robot is sent to an unknown environment, such as the moon, how can we ensure that it knows where it is and where to go? Assuming that the robot is given a map of the moon, it would need to localize itself within this map. This problem is known as *localization*, where the robot's coordinates are estimated relative to an external reference frame. Alternatively if the robot knows its position and orientation relative to a global coordinate frame, referred to as its pose, the problem then becomes one of constructing a map of the environment, which is known as *mapping*. However, in practice, the robot is typically not given a map of the environment, nor does it know its pose. Instead, all it can use are measurement readings about the state of the environment and its relative movement. The problem of constructing a map of the environment, while simultaneously localizing within that map, is called Simultaneous Localization and Mapping (SLAM). SLAM is a fundamental problem in robotics and provides a challenge for robot navigation in an unknown environment.

Placed in an unknown environment, a mobile robot collects sensor measurements and odometry data. The robot uses sensor measurements to obtain information about the state of the environment. For example, a robot might take an image using a camera or collect data using a laser range finder. Furthermore, the robot moves within the environment through control actions that cause the state of the robot to change. The odometry data, obtained from the robot's wheel encoders, describes the robot's movement. Motion induces a loss of knowledge due to noise in the odometry data, whereas measurement data increases the robot's knowledge of its pose and the map.

SLAM requires that the robot estimate both the map of its environment and its pose within the map using sensor measurements and odometry data. A typical probabilistic approach to solving SLAM is shown in Figure 1.1. The approach is composed of a prediction step and an update step that are performed recursively to estimate both the map and the robot pose. In this work, the map of the environment is composed of a set of significant three-dimensional locations called landmarks. In the prediction step, the odometry data is used to predict both the robot pose and the map. However,

the odometry data is often corrupted by noise that reduces the accuracy of both the predicted robot pose and the map. For this reason, measurements obtained from the robot's sensors, referred to as sensor measurements, are used in the update step to correct both the predicted robot pose and the map. How can we effectively use the sensor measurements to improve the accuracy of both the predicted robot pose and the map?

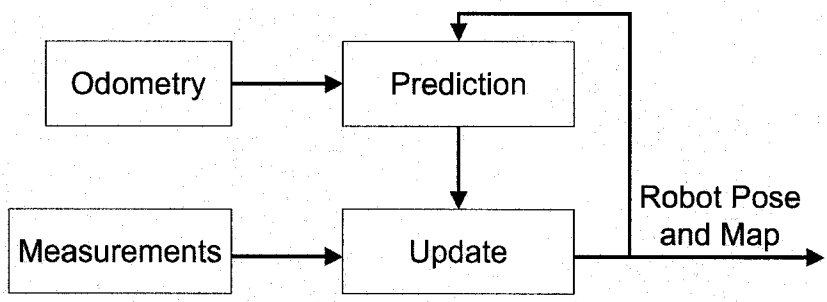

Figure 1.1: The typical approach to solving SLAM is composed of a prediction step and an update step that are performed recursively to estimate both the robot pose and the map. In the prediction step, odometry data is used to predict the robot pose and the map. In the update step, measurements obtained from the sensors are used to update the prediction.

In this thesis, we use a single camera to collect image data of the environment to help solve the SLAM problem. This approach is known as bearing-only visual SLAM. The images contain actual sensor measurements, which are two-dimensional points in the image corresponding to three-dimensional landmarks in the environment. The generation of sensor measurements in the physical world is described by a *camera sensor model*. Unfortunately, the camera sensor model is subject to noise that limit the accuracy of the collected measurement information. The noise associated with the measurement data is referred to as the measurement noise, which is the difference between the actual sensor measurement and the estimated sensor measurement obtained from the camera sensor model. Assuming known landmark positions, the measurement noise comes from both the camera model and the algorithm used to locate the two-dimensional points in the image. The measurement noise is taken into consideration in a probabilistic SLAM algorithm. The *probabilistic sensor model* is a conditional probability distribution $p(z_k|x_k)$, where x_k is the system state, which includes the robot pose and the map of the environment, and z_k is the sensor measurement at time k. A typical representation of a probabilistic camera sensor model consists of a deterministic component, which can be modelled by a pinhole camera model, and an uncertain component, which models how accurate the estimated sensor measurements are. Although we primarily address the development of a probabilistic camera sensor model within the context of the SLAM problem, it is also applicable to other types of visual robot navigation problems, such as visual localization.

To solve the SLAM problem, a probabilistic camera sensor model must account for uncertainty that exists in the real world. Typically, it is assumed that the accuracy of the SLAM algorithm

increases with the accuracy of the camera sensor model. Extensive work has been done in modelling the deterministic component of a camera sensor model, which is also known as camera calibration. However, there has been limited research in modelling the uncertain component: the measurement noise. How do we properly build a probabilistic camera sensor model that can be used in bearing-only visual SLAM? How robust is the SLAM algorithm to errors in the camera sensor model? What is the impact of an inaccurate camera sensor model on the performance of SLAM? This thesis addresses the above questions and provides a thorough study of a probabilistic camera sensor model.

1.2 Thesis Objectives and Contributions

The objective of this thesis is to build a probabilistic camera sensor model for bearing-only visual SLAM. Our focus is on modelling the uncertain component of the sensor model, the measurement noise, which accommodates the non-determinstic aspects of the model. In addition, we want to identify the importance of a probabilistic camera sensor model for bearing-only visual SLAM.

This thesis makes a number of research contributions. We introduce a method for building a probabilistic camera sensor model. This method is motivated by the camera calibration process and the definition of measurement noise. We confirm that for both close and distant landmarks, a probabilistic camera sensor model with constant measurement noise variance is sufficient for solving the bearing-only visual SLAM problem. Using the assumption that a camera sensor model is independent of the visual robot navigation algorithm, we test our camera sensor model on a localization problem. Simulation results demonstrate that our model gives good performance in terms of the accumulated localization error. The accumulated error is 16 percent of the length of the robot's travelling path for close landmarks, and 62 percent for distant landmarks. Under the same experimental setup, we suggest using close landmarks rather than distant landmarks based on the above result. We show that sensor models with a wide range values of measurement noise variance can give nearly optimal performance for the localization problem. In addition, we show that, in general, pessimistic camera sensor models perform better than optimistic models. From the experimental results, we conclude that an accurate model of the measurement noise is not critical for the performance of localization in a noisy environment if proper landmark uncertainty is used. In conclusion, this thesis provides a thorough study of a probabilistic camera sensor model for bearing-only visual SLAM.

1.3 Overview

The rest of this thesis is arranged as follows. Chapter 2 will discuss the background and related work in the field of SLAM and camera sensor models. Chapter 3 will introduce our method for building a probabilistic camera sensor model for bearing-only visual SLAM. In Chapter 4, a probabilistic camera sensor model is built based on data collected from a Dragonfly IEEE-1394 camera. In Chapter 5, visual localization simulation and experimental results are presented to validate our

probabilistic camera sensor model. The effects of the sensor model errors on visual localization are also discussed. Finally in Chapter 6, a general conclusion on building a probabilistic camera sensor model is presented along with possible directions for future work.

Chapter 2

Background

In the previous chapter, we presented the motivation and the contributions of this thesis. In this chapter, we introduce various concepts and definitions that form the background of this thesis. We discuss the problem of Simultaneous Localization and Mapping (SLAM) in mobile robot navigation and its two important solutions: EKF-SLAM and FastSLAM. We also overview the bearing-only visual SLAM problem and its related topics. We then talk about another robot navigation problem: localization using a single camera. Finally, the topic of building a camera sensor model in the context of bearing-only visual SLAM is presented.

2.1 SLAM

Simultaneous Localization and Mapping (SLAM) has been a major research area towards achieving fully autonomous robot systems. SLAM algorithms answer the question: "Can a robot estimate its travelling path and build a map of its environment given a set of measurements and the odometry data?" To express the process of solving the SLAM problem, we need to specify the environment and the robot's path. A *map* of the environment is a list of objects in the environment and their locations:

$$m = [m_1, m_2, \ldots, m_N] \tag{2.1}$$

where N is the number of objects in the environment and m_n, where $n = 1, 2, \ldots, N$, specifies the three-dimensional location of the object. In this thesis, objects are *landmarks*, which are distinct, static features of the environment. To build a map of the environment, the robot estimates the three-dimensional locations of the landmarks. At time k, a robot pose x_k^v describes the location and orientation of the robot. A measurement z_k of a landmark is taken from the robot location at time k. The robot is driven to the state x_k^v and the odometry data u_k is collected. A history of robot locations is the robot's path or trajectory.

Virtually all state of the art algorithms for SLAM are probabilistic [28], since robot odometry data and sensor measurements are noisy, complicated and difficult to accommodate. Probabilistic

techniques can help solve this problem by modelling various sources of noise and incorporating them into the SLAM algorithm.

Currently, the dominating probabilistic scheme for integrating sensor measurements and robot odometry data into the SLAM algorithm is Bayes filters [21]. A SLAM algorithm recursively estimates the system state x_k, including both the robot pose x_k^v and the map m, from a sequence of noisy sensor measurements $z_{1:k}$ and odometry data $u_{1:k}$. Since the robot pose, odometry data, and measurements are all subject to noise, the state estimate is characterized by uncertainty. This uncertainty is represented by a density function over a random variable x_k, $p(x_k|z_{1:k}, u_{1:k})$, conditioned on all odometry data and measurement data available at time k. This density provides an answer to the SLAM question: "What is the probability that the estimate state is x_k if the history of sensor measurements are $z_{1:k}$ and odometry data is $u_{1:k}$, for all possible x_k?"

To make this problem computationally tractable, Bayes filters assume that the dynamic system is Markov, indicating that all historical relevant information is contained in the previous state variable x_{k-1}. Figure 2.1 shows the dynamic Bayes network that characterizes the evolution of odometry data, sensor measurements and states. The state at time k only depends on the state at time $k-1$ and its odometry input u_k. The measurement z_k depends on the state x_k at time k.

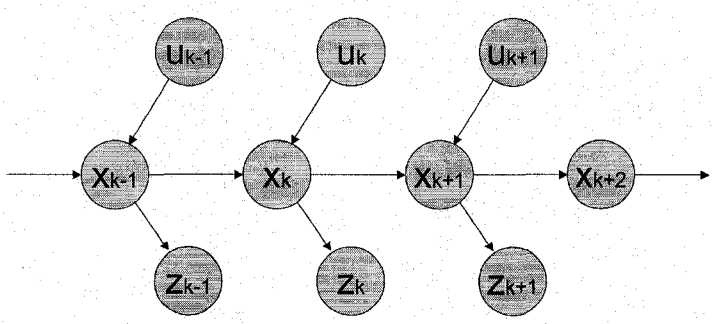

Figure 2.1: A graphical model of SLAM, describing the evolution of the odometry data $u_{k-1:k+1}$, the states $x_{k-1:k+2}$ and the measurements $z_{k-1:k+1}$.

In general, a recursive solution to the SLAM problem is used. For each time step, a SLAM algorithm first predicts the system state (prediction), and then corrects this prediction using available sensor measurements (correction).

Prediction: The state is predicted according to the following update rule:

$$p(x_k|z_{1:k-1}, u_{1:k})^- \leftarrow \int p(x_k|x_{k-1}, u_k)p(x_{k-1}|z_{1:k-1}, u_{1:k-1})d_{x_{k-1}} \tag{2.2}$$

where $p(x_k|x_{k-1}, u_k)$ is the *motion model*, meaning the probability of the system state is x_k, given that its previous state is x_{k-1} and the current odometry data is u_k.

Correction: With the generation of a sensor measurement z_k, the predicted system state is

corrected by the Bayes rule:

$$p(x_k|z_{1:k}, u_{1:k}) \propto p(z_k|x_k)p(x_k|z_{1:k-1}, u_{1:k})^{-} \tag{2.3}$$

where $p(z_k|x_k)$, the *sensor model*, describes the likelihood of making observation z_k given the system state x_k.

Equations 2.2 and 2.3 provide a recursive procedure for calculating $p(x_k|z_{1:k}, u_{1:k})$ based on all the sensor measurements $z_{1:k}$ and odometry data $u_{1:k}$. This recursion is a function of a motion model $p(x_k|x_{k-1}, u_k)$ and a sensor model $p(z_k|x_k)$.

2.2 Solutions to the SLAM Problem

The following section gives an overview of two important Bayes Filter solutions for the SLAM problem: EKF-SLAM and FastSLAM. Finding appropriate representations for both the motion model and the sensor model is required for solving the SLAM problem. Algorithms based on the Extended Kalman Filter (EKF) [13, 14, 16, 30] are the most widely used variant of Bayes filters. It approximates both the motion model and the sensor model in the form of a deterministic state-space model with additive Gaussian noise. Another popular alternative solution, FastSLAM [26, 27], uses a set of samples to represent the probabilistic distribution of the robot pose, along with Gaussians to represent landmarks in the map.

2.2.1 EKF-SLAM

A brief overview of the structure of the EKF framework is described here. Most of the notations are adopted from Welch and Bishop [34]. The goal of SLAM is to estimate a state vector containing the robot pose, x_k^v, and landmark positions $m = (m_1, m_2, \ldots, m_N)$ in the world reference frame. The Extended Kalman Filter represents the probabilistic distribution $p(x_k|z_{1:k}, u_{1:k})$ by the mean x_k and the covarinace matrix P_k as follows:

$$x_k = \begin{pmatrix} x_k^v \\ m_1 \\ m_2 \\ \cdots \\ m_N \end{pmatrix} \tag{2.4}$$

$$P_k = \begin{pmatrix} \sigma_{x_k^v x_k^v} & \sigma_{x_k^v m_1} & \sigma_{x_k^v m_2} & \cdots & \sigma_{x_k^v m_N} \\ \sigma_{m_1 x_k^v} & \sigma_{m_1 m_1} & \sigma_{m_1 m_2} & \cdots & \sigma_{m_1 m_N} \\ \sigma_{m_2 x_k^v} & \sigma_{m_2 m_1} & \sigma_{m_2 m_2} & \cdots & \sigma_{m_2 m_N} \\ \sigma_{m_3 x_k^v} & \sigma_{m_3 m_1} & \sigma_{m_3 m_2} & \cdots & \sigma_{m_3 m_N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \sigma_{m_N x_k^v} & \sigma_{m_N m_1} & \sigma_{m_N m_2} & \cdots & \sigma_{m_N m_N} \end{pmatrix} \tag{2.5}$$

where N is the number of landmarks in the map. The covariance matrix represents the uncertainty in all the variables in the state vector and the correlations between landmarks and the robot pose. The initial position of the robot is taken to be the origin of the world reference frame. At the beginning,

none of the landmark locations are known. As the robot moves, the dimensions of x_k and P_k grow dynamically with the number of landmarks in the map. Figure 2.2 illustrates the simulated image of the map obtained using the EKF algorithm. The map consists of six landmarks denoted by stars. The ellipses around the landmarks show the residual uncertainty as specified by part of the covariance matrix P_k. The dotted line represents the estimated trajectory of the robot. The robot starts from a known initial pose. Similarly, the robot's path is uncertain, as indicated by the ellipses that are shown along the path.

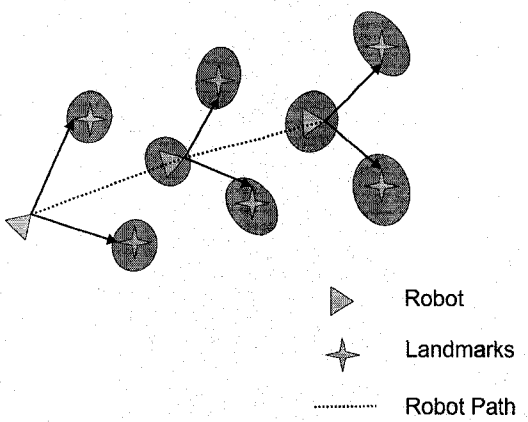

Figure 2.2: Example of Extended Kalman Filter estimation of the map and the robot path. Landmarks are denoted by stars and the triangle represents the robot. All the ellipses show the uncertainties around the estimated landmark positions and robot poses respectively.

The evolution of the EKF involves two steps as the mean x_k and the covariance matrix P_k change. The prediction step uses a motion model to determine the robot's pose and how its uncertainty increases after it moves. An updating step then incorporates the sensor model that describes how the uncertainty of both the robot pose and the map can be reduced. In practice, both the motion model and the sensor model are often non-linear functions. In a general EKF framework, the linearization process is performed on both models. EKF-SLAM makes a Gaussian noise assumption for both the motion model and the sensor model. Such an assumption is sufficient as long as the non-linearity of the system is small.

The motion model is in the form:

$$x_k = f(x_{k-1}, u_k) + w_k \tag{2.6}$$

where $f(x_{k-1}, u_k)$ models the robot kinematics and w_k is a zero-mean additive Gaussian noise with covariance matrix Q_k.

The sensor model is described as the following:

$$z_k = h(x_k) + v_k \tag{2.7}$$

where $h(x_k)$ describes the formation process of sensor measurements and v_k is additive sensor noise, characterized by a zero-mean Gaussian with the covariance matrix R_k.

Assuming the estimated system state x_{k-1} and its covariance matrix P_{k-1} from the last iteration, the prediction step of the Extended Kalman Filter incorporating the odometry data, is defined by the following:

$$x_k^- = f(x_{k-1}, u_k) \tag{2.8}$$

$$P_k^- = A_k P_{k-1} A_k^T + Q_k \tag{2.9}$$

where A_k is the Jacobian matrix of f with respect to x_{k-1}.

The observation update is described as follows:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{2.10}$$

$$x_k = x_k^- + K_k(z_k - h(x_k^-)) \tag{2.11}$$

$$P_k = (1 - K_k H_k) P_k^- \tag{2.12}$$

where H_k is the Jacobian matrix of h with respect to x_k and K is the Kalman gain that minimizes the difference between the true state and the estimated one.

The EKF-SLAM algorithm updates the system state (the robot pose and the landmarks) and the system covariance matrix every time an observation is made. Such an update process requires time quadratic in the number of landmarks in the map. Practical real-time implementations with many landmarks is the main disadvantage of this type of algorithm. However, because of the correlations between all the landmarks and the robot pose, EKF-SLAM gives better performance when the robot closes the loop, which means that the robot comes back to the start position [28]. Another important limitation of the EKF-SLAM approach lies in the Gaussian noise assumption, which is not suitable in non-Gaussian noise situations. More generally, the EKF-SLAM algorithm collapses due to failure in the *correspondence problem*, which is the problem of correctly associating individual sensor measurements with landmarks already in the map.

2.2.2 FastSLAM

The FastSLAM algorithm, introduced by Montemerlo *et al.* [26], represents the system state by a set of weighted particles distributed according to the distribution $p(x_k|z_{1:k}, u_{1:k})$. It uses a particle filter [15] to estimate the robot path, and a separate low-dimension EKF to estimate each individual landmark. Compared to the EKF-SLAM algorithm, FastSLAM can represent a non-linear motion model without techniques that approximate the model via a linear function. As a result, FastSLAM can be used on problems that EKF is not well-suited, for example, when the kinematics

are highly non-linear. An efficient implementation of the FastSLAM algorithm is time logarithmic in the number of landmarks [28]. Therefore, FastSLAM is more efficient than EKF-SLAM in terms of time complexity. Recent research finds that both FastSLAM and EKF-SLAM suffer from inconsistency [4,5], which means both algorithms fail to accurately estimate the uncertainty of the system state.

An important characteristic of the SLAM problem lies in the fact that given a robot pose, landmark locations can be estimated independently of each other. The SLAM system state can be factored into a robot pose component x^v and a conditional map component m:

$$p(x_{1:k}|z_{1:k}, u_{1:k}) = p(m|x^v_{1:k}, z_{1:k})p(x^v_{1:k}|z_{1:k}, u_{1:k}) \tag{2.13}$$

Notice that the probability distribution is conditioned on the trajectory $x^v_{1:k}$ rather than the single pose x^v_k. Given the robot's trajectory, the map landmarks become independent, and the recursive estimate divides the problem into two independent processes: estimation for the robot pose and that for the map. A robot pose is sampled, and assuming that this pose is perfect, the observed landmarks are updated individually as an EKF measurement update from a known robot pose. Thus, the map for a single particle is governed by the accuracy of the robot's trajectory.

Each particle consists of the robot pose, all the landmarks and their uncertainty covariance matrices. Calculating the distribution $p(x_{1:k}|z_{1:k}, u_{1:k})$ from time $k-1$ to time k involves generating a new particle set. This new particle set incorporates new odometry data u_k and sensor measurement z_k. The general FastSLAM 2.0 [17] algorithm is performed in the following steps:

- **Prediction**

  For each particle, compute a proposal distribution $p(x^v_k|x^v_{1:k-1}, z_{1:k}, u_{1:k})^-$, which is the approximation of the true distribution $p(x^v_k|x^v_{1:k-1}, z_{1:k}, u_{1:k})$, and draw a sample robot pose from it:

  $$x^v_k \backsim p(x^v_k|x^v_{1:k-1}, z_{1:k}, u_{1:k})^- \tag{2.14}$$

- **Measurement Update**

  For each particle, perform an EKF update on the observed landmarks as a simple mapping operation with a known robot pose.

- **Importance Weight**

  Samples are given importance weights to compensate for their variation. Calculate weight samples according to the importance function:

  $$w_k = w_{k-1}\frac{p(z_k|x^v_{1:k}, z_{1:k-1})p(x^v_k|x^v_{k-1}, u_k)}{p(x^v_k|x^v_{1:k-1}, z_{1:k}, u_{1:k})^-} \tag{2.15}$$

  which offsets the difference between the proposal distribution and the desired distribution by adjusting the weight of each particle.

- **Resampling**

  The purpose of resampling is so that the particles are not distributed according to the desired distribution. New particles are selected from the previous particle set with replacement according to the importance weight w_k. Selected particles are given uniform weight, $w_k = 1/N$, where N is the number of particles [28].

There are two versions of FastSLAM in the literature. For FastSLAM 1.0 [26], the robot pose is only sampled from the most recent odometry data u_k (see Equation 2.16), not paying attention to measurement z_k.

$$x_k \backsim p(x_k^v | x_{1:k-1}^v)^- \tag{2.16}$$

FastSLAM 2.0 [27], described in the above procedure, proposes an improvement to the proposal distribution by taking the measurement information into account (see Equation 2.14). This new proposal improves the performance of FastSLAM, especially when the accuracy of the motion model is relatively lower than that of the robot's sensor model [28].

2.3 Bearing-only Visual SLAM

Many solutions have been used to successfully solve the SLAM problem with sensors that provide both range and bearing information. However, sensors such as laser range finders are normally either large, heavy or expensive. Using cameras as sensors to help solve SLAM, has recently become an active research topic because of its many benefits over range-bearing sensors. First, cameras can be cheap, light and easily integrated into an embedded robot system. Second, it is possible to recover three-dimensional landmark positions from processing multiple two-dimensional images. Last, many well-known algorithms in computer vision can be used to solve feature detection and data association problems when solving the SLAM problem.

Solving the SLAM problem with a single camera falls into one SLAM algorithm category: bearing-only visual SLAM [12, 23]. Sensor measurements obtained from a camera can be used to calculate the angle to a landmark but not the range. The unavailability of the range information of the landmarks makes solving bearing-only visual SLAM a challenging task. Since the mean and the covariance of the landmark positions cannot be obtained by one observation, EKF or FastSLAM cannot be used directly because a full Gaussian estimation of the landmark position is required. Therefore, when a landmark is first observed, it is not incorporated as a landmark in the map of SLAM until it is well-conditioned, which means that it can be approximated by a Gaussian distribution. At least two observations at two different robot poses can be used to localize the position of the landmark. The process of initializing the three-dimensional location of a landmark is referred to as landmark initialization problem for bearing-only visual SLAM [2, 10, 24].

Another interesting topic in bearing-only visual SLAM is feature detection. Features are image points corresponding to three-dimensional landmarks in the physical environment. The task of

feature detection is to extract features from the measurements, and thus provide useful information from images to be used in the SLAM framework. The Harris Corner detector [7] is one of the most popular feature detection algorithms. This algorithm calculates the value of the corner response and selects a good corner point which has the largest intensity change in all directions. Another widely used feature detection algorithm, David Lowe's SIFT [25], can extract distinctive features from images. SIFT is able to robustly identify objects and is rotation, scale, and intensity invariant.

2.4 Localization

The algorithms described above all address the problem of estimating the robot pose as well as a map of its environment, which is known as Simultaneously Localization and Mapping (SLAM). The simpler case - estimating the robot pose with a known map - is also a basic perceptual problem in mobile robot navigation. Localization has the reputation of being robust and easy to implement compare to SLAM. Most importantly, localization is very useful for our experiments in this thesis because it separates the landmark initialization problem from the bearing-only visual SLAM problem. As a result, localization provides us a more convenient robot navigation algorithm to verify our camera sensor model than SLAM does. A probabilistic camera sensor model describes the formation process of sensor measurements that generated in the physical world. Such a model is independent of various types of robot navigation problems.

The localization problem is essentially the process of matching the map coordinate system and the robot's local coordinate system. With a known map of its environment, the goal of the robot is to determine its pose relative to this map given the perceptions of the environment and its movement. Because the robot pose cannot be measured directly and only one sensor measurement is not sufficient, measurement data has to be integrated over time to determine the robot's pose. Not every localization problem is equally difficult. Assuming a known initial robot pose, position tracking is an easier problem than global localization, of which the initial pose of the robot is unknown. The rest of the thesis talks about localization with known initial robot pose.

Localization using a single camera is a simple version of bearing-only visual SLAM. Visual localization assumes that we have knowledge of the map. In this case, localization does not have the landmark initialization problem, which is quite challenging to solve in bearing-only visual SLAM. Most of the popular algorithms for solving SLAM can also be applied to visual localization, namely EKF localization and particle filter based localization, which is similar to FastSLAM. For EKF localization, the system state vector only consists of the robot pose. Therefore, EKF-SLAM is the same as EKF localization algorithm, except that the algorithm does not need to update the landmark positions in the map any more. As mentioned earlier, FastSLAM decomposes the SLAM problem into two subproblems: localization and mapping. Particle filter based localization is equal to the localization component of the FastSLAM algorithm.

2.5 Probabilistic Camera Sensor Model

A sensor model $p(z_k|x_k)$ describes a robot's expected measurement z_k given its system state x_k. In the context of bearing-only visual SLAM and localization, a probabilistic camera sensor model can be described by Equation 2.7, which consists of a deterministic component $h(x_k)$ and an uncertain component v_k. The first component $h(x_k)$ is essentially a pinhole camera model, which projects the three-dimensional landmarks onto a two-dimensional image plane. The projection process is shown in Figure 2.3. The optical axis passes through the optical centre and is orthogonal to the image plane. Its intersection with the image plane is defined as the principal point. A three-dimensional landmark is projected onto the image plane.

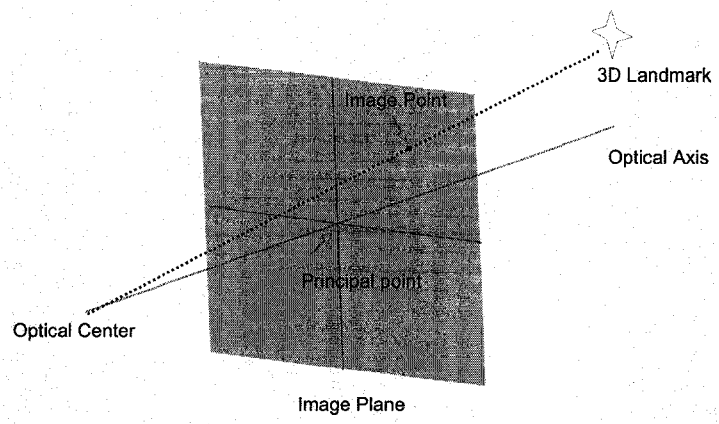

Figure 2.3: A pinhole camera model. A three-dimensional landmark is projected onto a two-dimensional image plane. Its intersection of the image plane is the image point of the three-dimensional landmark.

Much work has been done in camera calibration [20, 32, 36] for establishing a deterministic sensor model, but such a model is insufficient for bearing-only visual SLAM. Depending on the accuracy of the camera intrinsic parameters, the precision of the position of the image features obtained from the pinhole camera model in the two-dimensional space varies. Because the computed parameters will never be the actual ones, it is important to capture the uncertainty of the measurement due to the inaccuracy of the deterministic part of the sensor model. In addition, the noise introduced by the feature detection algorithm influences the accuracy of the sensor measurements. However, there is limited work discussing the development of a probabilistic camera sensor model.

Existing research in bearing-only visual SLAM has used sensor models that are either ad-hoc or based on the physics of the sensor. In a recent paper, Robert Sim *et al.* [29] used a stereo camera to explore an unknown environment. They used a fixed 10.0 pixels as the camera measurement noise. Although their camera sensor model is used for a stereo camera, the measurement noise covariance matrix can still be applicable to a single camera. Such a constant sensor model seems to be sufficient for their system. Nevertheless, theoretical understanding of a camera sensor model has

scientific significance.

Davison's work on monocular SLAM assumed a Gaussian noise sensor model whose covariance was determined by the image resolution [11]. However, detailed information was not provided regarding how the measurement noise covariance was calculated.

In Zucchelli and Kosecka's work [37], the relationship between the coordinates of the normalized feature position $x^n = [x\_n, y\_n, 1]$ and the image coordinates (pixel) $x^I = [u, v, 1]$ was defined by:

$$\begin{pmatrix} x\_n \\ y\_n \end{pmatrix} = \frac{1}{f} \begin{pmatrix} cc_1 - u \\ cc_2 - v \end{pmatrix} \tag{2.17}$$

where $[cc_1, cc_2]$ is the principal point. The distortion of the image is ignored, and both the horizontal focal length and the vertical focal length are the same f. They discussed how to propagate the uncertainty of a camera's intrinsic parameters into a covariance matrix $\Sigma_{\mathbf{x^n}}$ that characterizes the noisy feature positions in the normalized three-dimensional space. However, this still needs to be further propagated into the image space to capture $\Sigma_{\mathbf{x^I}}$ in order for it to be useful for bearing-only visual SLAM.

In a RoboCup application paper [18], a sensor model was represented by the range and bearing data, although the Sony Aibo used a camera to sense the artificial colored landmarks. They obtained the sensor model by calculating the standard deviation between the actual measurements and the estimated ones from the different samples. Our approach to building a probabilistic camera sensor model uses a similar empirical idea.

2.6 Summary

SLAM is an active research topic of both practical and theoretical importance. Over the past decade, much research has focused on understanding the problem of SLAM, implementing different variants of SLAM, and developing more consistent and efficient SLAM algorithms. The key challenge of SLAM in the near future is how to employ SLAM in large-scale or unstructured environments [3]. However, Julier [22] posed an insightful question about current research of SLAM: "many of these discussions have missed a fundamental point: does SLAM actually work?" He has looked at answering the following question: what happens to SLAM if there are data association errors, modelling errors or probability transformation errors? Very limited attention has been received in discussing those fundamental problems. We consider the issues of how to build a probabilistic camera sensor model and the impacts of the camera sensor model on the performance of the visual localization problem. In the next chapter, we will describe how we build a probabilistic camera sensor model.

Chapter 3

Model Development

In the previous chapter, we presented background information and related work in the research areas of SLAM and, more specifically, camera sensor models. In this chapter, we introduce our method for building a probabilistic camera sensor model for bearing-only visual SLAM. We begin by defining what is a camera sensor model. We then discuss the methodology used to build the sensor model. Finally, we describe our probabilistic camera sensor model, which is developed based on the camera calibration process and the definition of measurement noise.

3.1 Probabilistic Camera Sensor Model

In bearing-only visual SLAM, a mobile robot moves around an unknown environment and makes measurements using a single camera. The robot's pose is defined by $x^v = [x_v, y_v, z_v, \psi, \theta, \phi]$, where $[x_v, y_v, z_v]$ is the three-dimensional position and $[\psi, \theta, \phi]$ represents the orientation. A vector $m = [m_1, m_2, ..., m_N]$ describes the three-dimensional locations of landmarks in the map, where N is the total number of landmarks. The system state at time k is represented as

$$x_k = [x_k^v, m_1, m_2, m_3, ..., m_N]^T \tag{3.1}$$

The robot moves to the pose x_k^v at time k and the odometry data $u_{1:k}$ represent all the robot movements so far. $z_{1:k}$ are a set of landmark measurements available at time k. A general SLAM algorithm recursively estimates both the robot pose and the map from a sequence of noisy sensor measurements and the odometry data. For each iteration, the SLAM algorithm works in two steps: state prediction and measurement update as shown in Figure 3.1. First, the odometry data from the robot's wheel encoders is used to predict both the robot pose and the map. When a new measurement z_k is made, the predicted system state is immediately corrected based on the new information using:

$$p(x_k|z_{1:k}, u_{1:k}) \propto p(z_k|x_k)p(x_k|z_{1:k-1}, u_{1:k})^- \tag{3.2}$$

where the probabilistic distribution $p(x_k|z_{1:k}, u_{1:k})$ over x_k represents the uncertainty of the system state created by the noisy odometry data and uncertain measurements. The distribution $p(x_k|z_{1:k-1}, u_{1:k})^-$

is the proposal distribution from the prediction step, and $p(z_k|x_k)$ is the sensor model, describing the likelihood of making measurement z_k given the system state x_k.

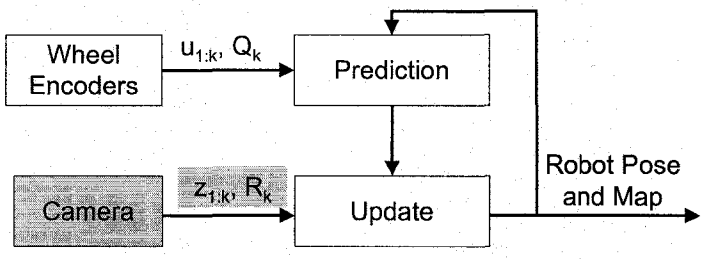

Figure 3.1: A flowchart of bearing-only visual SLAM.

The uncertainty of the system state due to the noisy odometry data and uncertain measurements is represented by a probabilistic distribution $p(x_k|z_{1:k}, u_{1:k})$ over x_k. A measurement z_k is related to the current state x_k and measurement noise v_k, which is the difference between the actual measurement and the estimated measurement, by the function:

$$z_k = h(x_k) + v_k \tag{3.3}$$

where $h(x_k)$ is the deterministic component of the camera sensor model and v_k is the uncertain component.

For bearing-only visual SLAM, let $X^C = [x^C, y^C, z^C]$ be the three-dimensional position of a landmark in the camera reference frame. The landmark is reprojected onto the image plane by:

$$h(x_k) = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} cc_1 \\ cc_2 \end{pmatrix} + (1 + k_1 r^2) \begin{pmatrix} f_x \frac{x^C}{z^C} \\ f_y \frac{y^C}{z^C} \end{pmatrix} \tag{3.4}$$

where

$$r^2 = \left(\frac{x^C}{z^C}\right)^2 + \left(\frac{y^C}{z^C}\right)^2 \tag{3.5}$$

where f_x and f_y are the focal length in the horizontal and vertical directions respectively, $[cc_1, cc_2]$ is the principal point (image center) of the image, and k_1 is the radial distortion coefficient. The camera intrinsic parameters $[f_x, f_y, cc_1, cc_2, k_1]^T$, which describe the camera's internal characteristics, can be estimated from the camera calibration process.

In the SLAM framework, the measurement noise v_k can be captured by a covariance matrix R_k, which determines the accuracy of the measurement information acquired from the sensor model. When R_k approaches zero, the algorithm relies more on the actual measurement than the odometry data. Thus, R_k determines the significance of measurement information on the final estimation of the robot pose and the map of the environment. An accurate camera sensor model properly estimates the measurement noise, an optimistic camera sensor model underestimates the measurement noise, and a pessimistic camera sensor model overestimates the measurement noise. The focus of this

chapter is to model the uncertain component of a camera sensor model, which is applicable to both bearing-only visual SLAM and visual localization.

3.2 Methodology

In this section, we introduce the methodology used to build our probabilistic camera sensor model. The deterministic component of a camera sensor model is defined as a function h (a pinhole camera model) from three-dimensional space to two-dimensional image space as described in Equation 3.4. Our focus is to estimate the measurement noise inherent in the sensor measurements from a camera model.

There are three different image points associated with a three-dimensional landmark X^W in the world reference frame: the actual observed image point x, the measured image point $\overline{x}$ and the estimated image point x^I, as shown in Figure 3.2. The actual observed image point is the ground truth where the three-dimensional landmark is projected onto an image plane through a camera. The position of the actual image point in the image space is determined using feature detection algorithms such as the Harris Corner detector [7] or the SIFT [25] algorithm. However, the measured point $\overline{x}$ is not the actual point x because of the error in the corner detection algorithm. We can approximate the actual image point by projecting X^W into the image space using a pinhole camera model described by Equation 3.4. Thus, the estimated image point $x^I = [u, v]$ equals $h(x_k)$. Notice that the actual image point, the measured image point and the estimated image point are all on the same image plane.

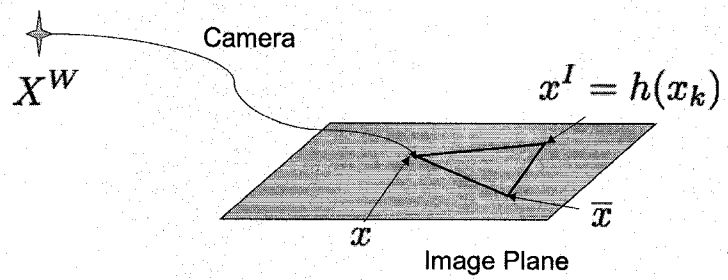

Figure 3.2: A diagram of image points associated with a three-dimensional landmark X^W. x is the actual image point on the image plane, $\overline{x}$ is the measured image point and x^I is the estimated image point.

The reprojection error is defined as the difference between the measured image point $\overline{x}$ and the estimated image point x^I [19]. The difference between the actual observed image point x and the estimated image point x^I is the measurement noise. Their relationship in the vector space is defined by:

$$x - x^I = (x - \overline{x}) + (\overline{x} - x^I) \tag{3.6}$$

where the componet $x - \overline{x}$ is the error introduced by the feature detection algorithm. Since it is impossible to get the position of the actual image point, we cannot estimate the measurement noise or measurement noise covariance matrix R_k directly. Our method outlined in the Section 3.3.2 estimates the covariance matrix R_k indirectly. First, we design experiments to estimate the reprojection errors. We then consider the noise introduced by the feature detection algorithm.

3.3 Model Building

We now introduce our method for building a probabilistic camera sensor model for bearing-only visual SLAM. First, a camera calibration process is introduced to build the deterministic component of the camera sensor model. We then outline our procedure for estimating the uncertain component v_k. Our method is inspired by the camera calibration process, and the model is built based on the definition of measurement noise, which is the difference between the actual image point and the estimated image point.

3.3.1 Camera Calibration

A probabilistic camera sensor model consists of two components: a pinhole camera model $h(x_k)$ and measurement noise v_k. To determine the pinhole camera model, we need to calculate the camera's intrinsic parameters, which can be obtained from the camera calibration process. Various methods for camera calibration can be found in the literature [20, 32, 36]. A commonly used camera calibration toolbox [6] is based on Heikkila and Silven's work.

Camera calibration is the process of determining the intrinsic parameters and extrinsic parameters of a camera. Extrinsic parameters are required to transform object coordinates into camera centered coordinates. In other words, we can calculate the three-dimensional positions of the landmarks in the camera reference frame using the extrinsic parameters. Intrinsic parameters describe the internal geometric and optical characteristics of a camera. The intrinsic parameters used in this thesis include the focal length $[f_x, f_y]$, the principal point $[cc_1, cc_2]$, and the radial distortion coefficient k_1.

The calibration procedure used in this work is performed with a checkerboard as shown in Figure 3.3. The origin of the world reference frame is the left hand corner of the checkerboard, and the z-axis of the frame is perpendicular to the checkerboard plane. The corners on the checkerboard are three-dimensional landmarks, and their corresponding image points in the image are feature points. Based on the knowledge of the grid size, we can calculate the three-dimensional locations of all the corners on the checkerboard in the world reference frame. The procedure first ignores the radial distortion coefficient k_1 to acquire an initial value of the focal length $[f_x, f_y]$ and the principal point $[cc_1, cc_2]$. The resulting transformation from a corner point at location $X^W = [x^W, y^W, z^W]$ to a image point $[u, v]$ is represented by the following equation:

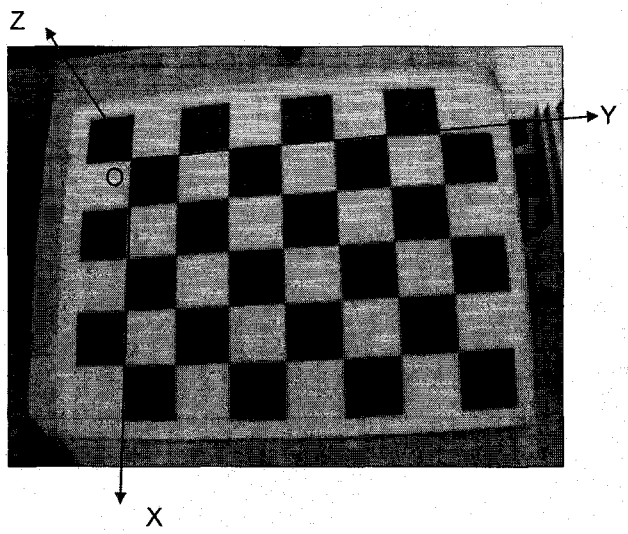

Figure 3.3: A checkerboard with the origin of the world reference frame is on the left hand corner. Z-axis of the frame is perpendicular to the checkerboard plane.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = A \begin{pmatrix} x^W \\ y^W \\ z^W \\ 1 \end{pmatrix} \tag{3.7}$$

where matrix A is a homogeneous 3×4 matrix. Given sufficient pairs of image points and corner locations, the Direct Linear Transformation (DLT) algorithm can be used to solve this linear transformation [1]. Focal lengths, principal point and extrinsic parameters can be extracted from the solved matrix A. Next, an optimization algorithm is used to iteratively calculate all extrinsic parameters and intrinsic parameters including the radial distortion coefficient. Parameters $[f_x, f_y, cc_1, cc_2]$ from the DLT algorithm are used as the initial values for the optimization procedure to avoid local minima. The optimization algorithm minimizes the sum of squared residuals between the observed image points and the estimated image points obtained from the pinhole camera model. The standard deviation of the parameters $[f_x, f_y, cc_1, cc_2, k_1]$ can be calculated using backward propagation of the covariance of residuals between the observed image points and the estimated image points [19].

After camera calibration, the camera's intrinsic parameters are known and a pinhole camera model is determined. The following section describes our method for estimating the uncertain component of the probabilistic camera model.

3.3.2 Reprojection Error

We cannot find the measurement noise directly. However, we can calculate the reprojection error first, and then consider the noise introduced by the feature detection algorithm. The following paragraph outlines how to determine the reprojection error to build a probabilistic camera sensor model.

Figure 3.4 shows the procedure for building a probabilistic camera sensor model [35]. This method is statistical. When we collect a sufficient number of samples of the reprojection error between a measured image point and an estimated image point, we can then determine the reprojection error covariance matrix Σ_{reproj} from the sampled reprojection errors. We first determine the number of samples N to obtain a statistically accurate estimate of the reprojection error covariance matrix. The process starts by first calibrating the camera. The camera calibration technique, described in Section 3.3.1, provides the camera's intrinsic and extrinsic parameters. We then take an image of the camera calibration rig, which in our work is a checkerboard, placed at an arbitrary distance from the camera as shown in Figure 3.5. From the extrinsic parameters, we can now calculate the transformation between the checkerboard and the camera. Using the transformation, the three-dimensional positions of landmarks (the corner points) in the camera reference frame are obtained. Applying the pinhole camera model in Equation 3.4, we can obtain the coordinates of the estimated image points and their difference between the measured image points, which are the reprojection errors.

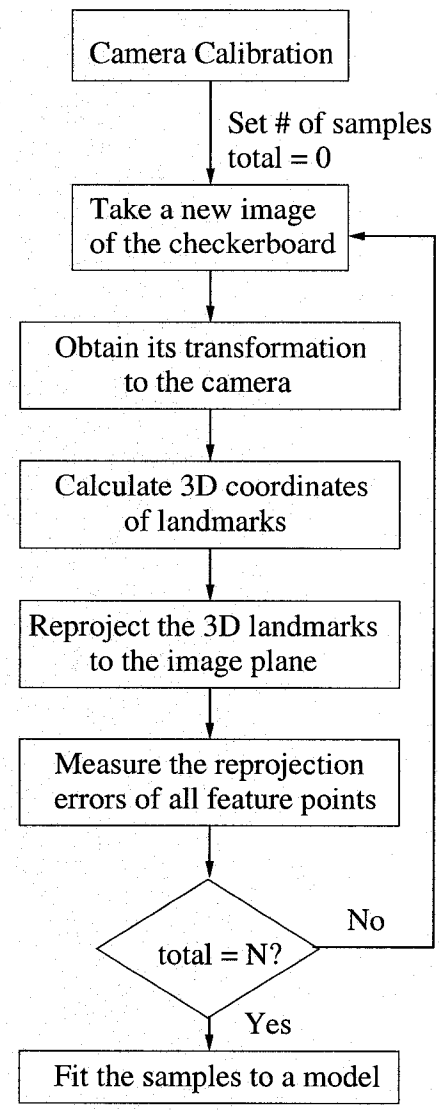

Figure 3.4: A flowchart describing the construction of a probabilistic sensor model for a camera [35].

If the number of feature points is smaller than N, we take another image of the checkerboard and repeat the above process. Otherwise, we use the collected samples of the reprojection errors to build a model to obtain the reprojection error covariance matrix Σ_{reproj}. We repeat the procedure in Figure 3.4 when the distance between the checkerboard and the camera changes. We can then calculate different Σ_{reproj} when the feature points are at various distances from the camera.

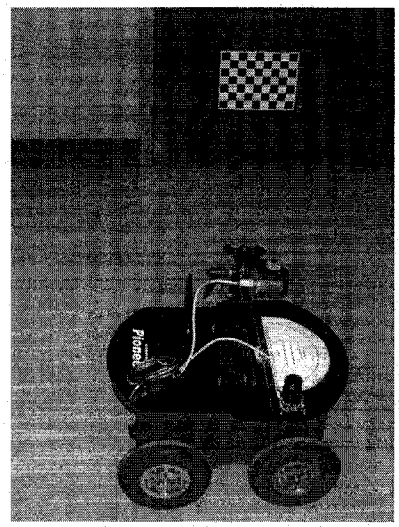

Figure 3.5: A camera mounted on an ActiveMedia Pioneer 3 robot facing a checkerboard.

The measurement noise in bearing-only visual SLAM not only comes from reprojection but also from errors caused by the feature detection algorithm (see Equation 3.6). In this thesis, we use a popular feature detection algorithm, the Harris Corner detector, to find the actual observed feature points x_i in the image. However, as with all feature detectors, the location of the features within the image may not be accurate. In Sojka's work, the accuracy of the Harris Corner Detector was evaluated on both synthesized images and real images [31]. They demonstrated that the average distance between the actual image points and the detected image points is between 0.5 pixels and one pixel. Because the Harris Corner Detector has sub-pixel accuracy, we assume that the variance of the errors introduced by the Harris Corner Detector is bounded by one pixel. Since the error introduced by the Harris Corner Detector is evaluated based on a different process from the reprojection errors, which are determined from the procedure in Figure 3.4, we assume that these two errors are independent from each other. Therefore, the covariance matrix of the measurement noise R_k is equal to the sum of both reprojection error covariance matrix Σ_{reproj} and the detection algorithm error covariance matrix. Numerically, R_k is equal to Σ_{reproj} plus the extra pixel errors introduced by the Harris Corner Detector, which are added to the diagonal elements of the reprojection error covariance matrix. In this thesis, we use the upper bound of one pixel as the noise caused by the Harris Corner Detector.

3.4 Summary

Robot navigation algorithms such as EKF-SLAM and FastSLAM require both deterministic and uncertain components as measurement inputs to the Baye Filters. As we mentioned in the related work, much work has been done in estimating the intrinsic parameters of a camera. Our major concern in this thesis is how to build the uncertain component of a camera sensor model. Since we cannot estimate the measurement noise directly, our method introduces a procedure to determine the measurement noise covariance matrix R_k indirectly.

Chapter 4

Experiments of Building a Camera Sensor Model

In this chapter, we describe the experiments performed to build a probabilistic camera sensor model. First, the deterministic component of a camera sensor model is built by estimating the camera's intrinsic parameters. Then, the procedure to build the uncertain component is introduced and issues related to building a camera sensor model using real data are discussed. Finally, the properties of a camera sensor model are presented.

4.1 Deterministic Component of a Camera Sensor Model

In this work, the deterministic component of the camera sensor model is a pinhole camera model from the camera's intrinsic parameters. We use the camera calibration process to determine the necessary intrinsic parameters and their uncertainties. Our camera calibration is performed using the Camera Calibration Toolbox for Matlab [6] on images taken from a Dragonfly IEEE-1394 camera with 640×480 pixel resolution. The calibration is performed using a planar checkerboard shown in Figure 4.1 with 35 corners used as landmarks. The size of the grid on the checkerboard is 31.5 cm×31.5 cm. Table 4.1 shows the resulting intrinsic camera parameters and their corresponding standard deviations.

f_x[pixel]	f_y[pixel]	cc_1[pixel]	cc_2[pixel]	k_1[pixel]
769.77 ±1.45	775.08 ±1.41	330.43 ±1.09	254.77 ±1.03	-0.36 ±0.003

Table 4.1: Intrinsic camera parameters and their standard deviations. f_x and f_y are the camera focal length in x and y directions respectively. cc_1 and cc_2 represent the principal point on the image plane. k_1 is the radial distortion scale factor of the image.

4.2 Uncertain Component of a Camera Sensor Model

After camera calibration, we proceed to estimate the measurement noise covariance matrix. The following sections present related topics regarding to the use of reprojection errors to build the

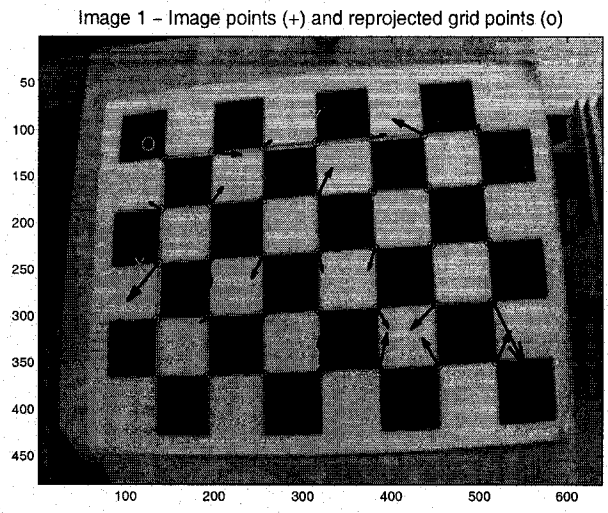

Figure 4.1: The calibration pattern used in the experiments. Arrows represent the reprojection errors between the measured corners and the estimated ones.

uncertain component of a camera sensor model. In addition, a normality test is used to verify the assumption that a Gaussian function can be used to model the measurement noise for both the EKF-SLAM and the FastSLAM algorithms. Finally, a measurement noise covariance matrix is obtained indirectly from the reprojection error covariance matrix.

4.2.1 Normality Test

From the camera calibration process, we obtain the deterministic component of a camera sensor model. We then follow the model building procedure described in Section 3.3.2 to generate the reprojection errors using corner points as landmarks at various distances from the camera. In Figure 4.1, each arrow of the corner represents the reprojection error, which is the difference between the measured corner and the estimated corner. From the resulting set of images, we determine the reprojection error covariance matrix.

The reprojection errors appear to follow a normal distribution according to our experimental results. In order to test this statistical hypothesis, we use Geary's test [33]. This test is based on the ratio of two estimators of the sample standard deviation. For example, suppose there is a set of random samples $X_1, X_2, \ldots, X_n$, where U is defined as

$$U = \frac{\sqrt{\frac{\pi}{2}} \sum_{i=1}^{n} |X_i - \overline{X}|/n}{\sqrt{\sum_{i=1}^{n} (X_i - \overline{X})^2/n}} \tag{4.1}$$

The standard normality Z is transformed from U, using:

$$Z = \frac{U - 1}{0.2661/\sqrt{n}} \tag{4.2}$$

This transformation produces a z-statistic that is converted to a probability by calculating the p-value:

$$p(Z) = \frac{1}{\sqrt{2\pi}} e^{\frac{-z^2}{2}} \tag{4.3}$$

The decision to accept the normality hypothesis is made when the p-value is larger than or equal to the significance value. In our experiments, we set the significance level at $\alpha = 0.05$, which is commonly used in statistical analysis. We then test the normality hypothesis for landmarks at different ranges. The results of Geary's test are shown in Table 4.2.1.

ranges[m]	**p-value (x)**	**p-value(y)**
50 cm	0.5972	0.7771
70 cm	0.2158	0.3092
90 cm	0.0684	0.2939
110 cm	0.0572	0.1018
130 cm	0.3204	0.9498
150 cm	0.0651	0.1362
170 cm	0.8533	0.6023
210 cm	0.2243	0.7927

Table 4.2: The results of Geary's test on reprojection error samples at both x and y directions. The number of samples used is 500. The significance level is set to $\alpha = 0.05$.

Since the p-values of the reprojection error samples along both the x and y directions are larger than α, we are able to conclude that the reprojection errors do indeed follow Gaussian distributions. We proceed to determine the reprojection error covariance matrix using

$$\Sigma_{reproj} = \frac{1}{N} \sum_{i=1}^{N} (X_i - \overline{X})(X_i - \overline{X})^T \tag{4.4}$$

The covariance matrices at the three different ranges: 50 cm, 130 cm and 210 cm, shown in Figures 4.2 to 4.4), are found to be:

$$\Sigma_{reproj1} = \begin{pmatrix} 0.279^2 & 0.006 \\ 0.006 & 0.253^2 \end{pmatrix} \tag{4.5}$$

$$\Sigma_{reproj2} = \begin{pmatrix} 0.125^2 & -0.001 \\ -0.001 & 0.120^2 \end{pmatrix} \tag{4.6}$$

$$\Sigma_{reproj3} = \begin{pmatrix} 0.098^2 & 0.000 \\ 0.000 & 0.106^2 \end{pmatrix} \tag{4.7}$$

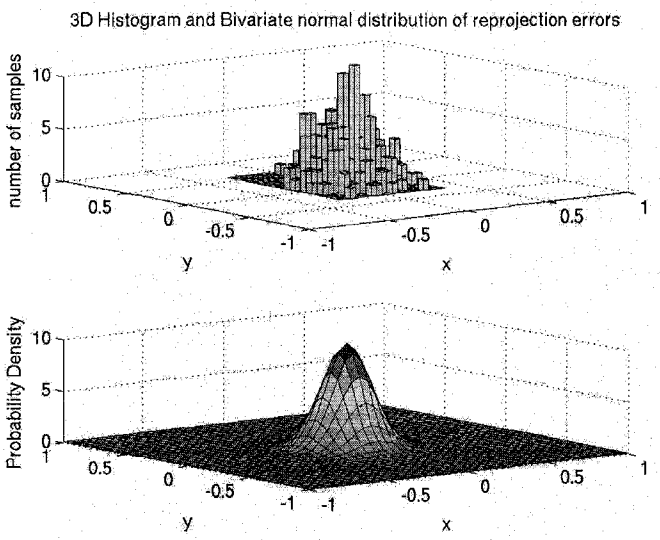

Figure 4.2: Reprojection error standard deviation along both the horizontal and vertical directions. The reprojection error covariance matrix $\Sigma_{reproj1}$ is (0.279^2, 0.006; 0.006, 0.253^2). The landmarks are at a distance of 50 cm from the camera.

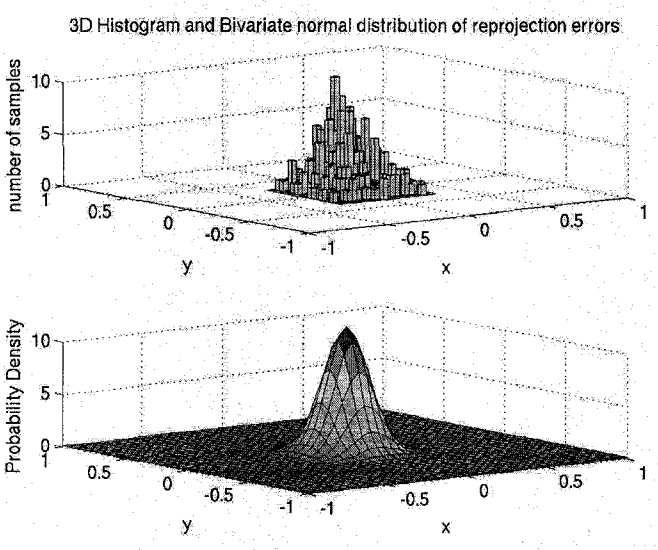

Figure 4.3: Reprojection error standard deviation along both the horizontal and vertical directions. The reprojection error covariance matrix $\Sigma_{reproj2}$ is (0.125^2, -0.001; -0.001, 0.120^2). The landmarks are at a distance of 130 cm from the camera.

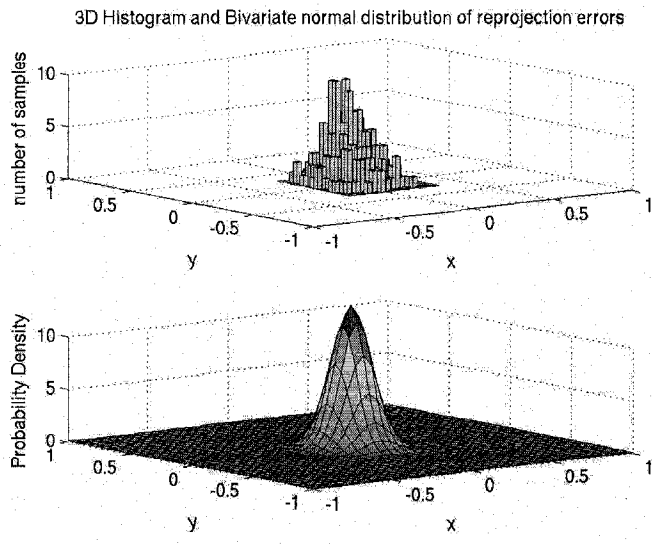

Figure 4.4: Reprojection error standard deviation along both the horizontal and vertical directions. The reprojection error covariance matrix $\Sigma_{reproj3}$ is (0.098^2, 0.000; 0.000 0.106^2). The landmarks are at a distance of 210 cm from the camera.

As mentioned in Section 3.3.2, the error introduced by the Harris Corner Detector algorithm is independent of the reprojection errors. Thus, the measurement noise covariance matrix, which is the sum of the measured noise covariance matrix and the reprojection error covariance matrix, still follows a Gaussian distribution. Since popular SLAM algorithms, such as EKF-SLAM and FastSLAM, assume a Gaussian distribution of the measurement noise, our confirmation of this assumption is a convenient result for vision based SLAM algorithms.

4.2.2 Sensitivity to Range

Our previous experiments show that range, which is the distance between the landmark and the camera, is important. As shown earlier, the reprojection error covariance matrix has the value of $\Sigma_{reproj1}$ = (0.279^2, 0.006; 0.006, 0.253^2), $\Sigma_{reproj2}$ = (0.125^2, -0.001; -0.001, 0.120^2), and $\Sigma_{reproj3}$ = (0.098^2, 0.000; 0.000 0.106^2) for the ranges of 50 cm, 130 cm and 210 cm respectively. To demonstrate the effect of range on the reprojection error standard deviation, we carry out the model building procedure, described in Figure 3.4, at different ranges. Figure 4.5 shows an example of the images that are taken at a range of 50 cm away from the camera. In order to obtain a statistically meaningful number of feature points, we take a set of images at the same range. Limited by the available image resolution of the Dragonfly IEEE-1394 camera, our experimental range between the camera and the landmarks extends from 50 cm to 210 cm. Figure 4.6 shows the relationship between the reprojection error standard deviation and the range in both the horizontal and the vertical directions.

From Figure 4.6, we see that the reprojection error standard deviation in both the horizontal

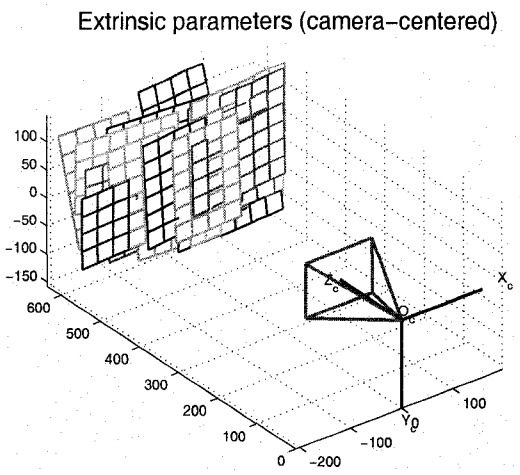

Figure 4.5: The relationship between a camera and several images that are taken at a range of 50 cm in the camera coordinate frame.

and vertical directions decrease when the landmarks are further away from the camera. Ideally, two different three-dimensional points should project to the same image point if they are along the same ray originating from the principal point. Why is the reprojection error standard deviation sensitive to range? There are two possible reasons to explain this. The first possible reason is the discretization error, which is resulting from the fact that the continuous three-dimensional space are represented by discrete two-dimensional digital image space. In the three-dimensional space, more distant landmarks than close landmarks are represented by the same pixel in a digital image. Therefore, the difference between the measured image point and the estimated image point decreases with the range.

The second possible reason is the uncertainty of the three-dimensional positions of the landmarks. For example, there are two three-dimensional landmarks $X_A = [x_A^C, y_A^C, z_A^C]$ and $X_B = t[x_A^C, y_A^C, z_A^C]$. If the scale factor t is larger than one, then the landmark B is further away in the camera reference frame. According to Equation 3.4, these two landmarks are projected onto the same two-dimensional image point $x^I = [u, v]$. The coordinate value along the z axis for landmark B scales the reprojection error. As a result, the same noise in the coordinates of a three-dimensional landmark leads to different noise in the image space for close and distant landmarks. Thus the reprojection errors in the two-dimensional images of distant three-dimensional landmarks are smaller than those of close landmarks [35].

4.2.3 Measurement Noise Covariance Matrix

In order to obtain the measurement noise covariance matrix, we add an extra pixel noise, introduced by the corner detection algorithm, to the diagonal elements of the reprojetion error covariance matrix. Since the off-diagonal values are very small and therefore can be ignored, we define the

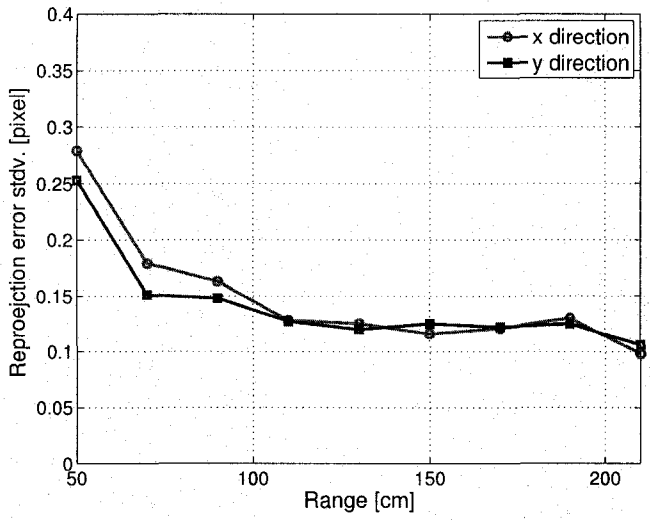

Figure 4.6: Reprojection error standard deviation along the x and y directions for landmarks at various distances from the camera.

measurement noise covariance matrix R_k as $diag(\delta u^2, \delta v^2)$. δu and δv represent the measurement noise standard deviation in the x and y directions respectively. Figure 4.7 shows the final measurement noise standard deviation of various ranges after adding one extra pixel noise to account for error in the Harris Corner Detector. Since the accuracy of the Harris Corner Detector is sub-pixel, therefore, adding one pixel noise sets the upper bound of the measurement noise standard deviation. The experimental results demonstrate that compared to the reprojection error standard deviation, the measurement noise standard deviation is less sensitive to the ranges, since the noise introduced by the Harris Corner Detector algorithm dominated the reprojection error in the measurement noise. Therefore, both the close landmarks and distant landmarks have a similar value of measurement noise standard deviation, of which the maximum difference is approximately 0.03 pixels. Therefore, a constant measurement noise covariance matrix $R_k = diag(1^2, 1^2)$ is sufficient for a set of landmarks in a map regardless of their distance from the camera. Since the pinhole camera model is deterministic, we can claim that a camera sensor model with constant measurement noise variance can be used for bearing-only visual SLAM and localization algorithms.

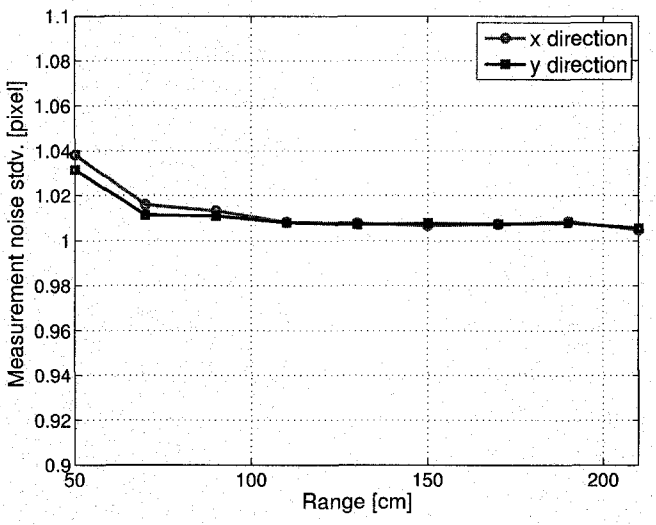

Figure 4.7: Plot of the measurement noise standard deviation with respect to various ranges between the camera and the landmarks. The measurement noise covariance matrix $R_k = diag(\delta u^2, \delta v^2)$. δu and δv represent the measurement noise standard deviation in the x and y directions respectively.

4.3 Summary

In summary, the construction of the camera sensor model can be obtained from the camera calibration process using our reprojection error procedure. In the procedure, we build a camera sensor model based on the definition of measurement noise, which is the difference between the actual measurement and the estimated measurement. According to Geary's test, we conclude that the reprojection errors in both the x and y directions follow normal distributions, which verifies the assumption that the sensor model noise is Gaussian. By repeating the same procedure using different ranges between the camera and the checkerboard, we find that the reprojection error standard deviation decrease as the range increases. Although the resulting changes are not significant with respect to the final measurement noise covariance matrix, it provides an interesting observation of camera sensor models in bearing-only visual SLAM. Our study of a probabilistic camera sensor model helps us understand why a camera sensor model with constant measurement noise variance is sufficient for landmarks at any distance. In addition, we analyze the sources of the noisy measurement and the difference between the measurement noise and the reprojection error. Therefore, our method not only builds a probabilistic camera sensor model but also provides a detailed study of the measurement noise.

Chapter 5

Model Validation

In the previous chapter, we used real data to build a probabilistic camera sensor model. In this chapter, we conduct a set of localization experiments in both a simulated environment and a real environment to verify our probabilistic camera sensor model. We then analyze the effect of sensor model error on the performance of a particle filter based localization algorithm. Finally, simulation and experimental results are presented.

5.1 Experimental Setup

In this chapter, we conduct a series of experiments to validate the sensor model we developed in the previous chapter. As we discussed in Chapter 2.6, feature initialization is a very challenging problem in bearing-only visual SLAM and is beyond the scope of this thesis. We will avoid feature initialization by manually measuring the three-dimensional landmark positions in our real experiments. In the simulation, we always assume that the three-dimensional positions of the landmarks are known. As a result, our experiments focus on the localization aspect of SLAM. Nonetheless, the experimental results are still applicable to SLAM if the feature initialization algorithm can provide accurate landmark positions. Since the probabilistic camera sensor model is necessary in any probabilistic approach to visual robot navigation, our sensor model is applicable to solve both visual localization and bearing-only visual SLAM.

5.1.1 Simulation and Experimental Environment

Our experimental environment is a corridor in the Department of Computing Science at the University of Alberta. Figure 5.1 shows the ActiveMedia Pioneer 3 robot we use in our experiments. Wheel encoders are fitted to the embedded computer to provide odometry data, including current robot positions and headings relative to the initial pose. Using a single camera, sufficient parallax, generated between two image frames, is needed to locate the three-dimensional landmark position. If the camera faces forward, it is difficult to obtain sufficient parallax since the camera faces the same direction as the robot's movement. Therefore, a Dragonfly IEEE-1394 camera is mounted

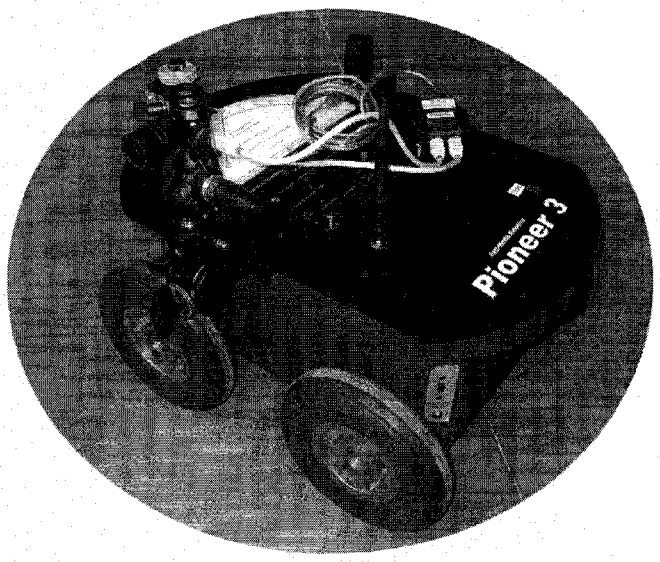

Figure 5.1: An ActiveMedia Pioneer 3 robot mounted with a camera mounted on the robot facing perpendicular to the robot's forward motion.

on the robot facing perpendicular to the robot's forward motion. Artificial landmarks are placed on the wall so that the Harris Corner Detector can easily identify features from the images. The three-dimensional landmark locations are measured relative to the robot's initial pose. Numbers attached to the artificial landmarks help us correctly associate individual sensor measurements with landmarks in the map. Figure 5.2 gives a sample image of what is viewed by the robot when it travels along the hallway.

The robot is driven in a straight line in the hallway of the Department of Computing Science building, shown in Figure 5.3. The robot moves about 15 meters while observing landmarks on the wall and takes an image approximately every 15 centimeters. However, the noisy odometry data indicates to the robot that it moves in a curve. To evaluate the localization performance, it is necessary to know the robot's real path. For this reason, the robot's real path is measured manually to ensure that accurate distances are recorded.

In the simulation we know the robot's real path and landmark positions. We simulate the measurement noise by adding Gaussian noise with zero mean to the image feature position. Therefore, the added Gaussian noise becomes our true measurement noise for our sensor model. We define this probabilistic camera sensor model, using the true measurement noise covariance matrix, as a true camera sensor model. Alternatively, we define inaccurate sensor models as sensor models that either overestimate or underestimate the measurement noise standard deviation. To evaluate the performance of the true sensor model we built, we use various inaccurate sensor models to perform the localization algorithm. Compared with the noisy real environment, our simulation guarantees perfect knowledge of the landmark positions and their transformation to the robot pose.

In both the simulation environment and the real environment, the robot moves on a two-dimensional

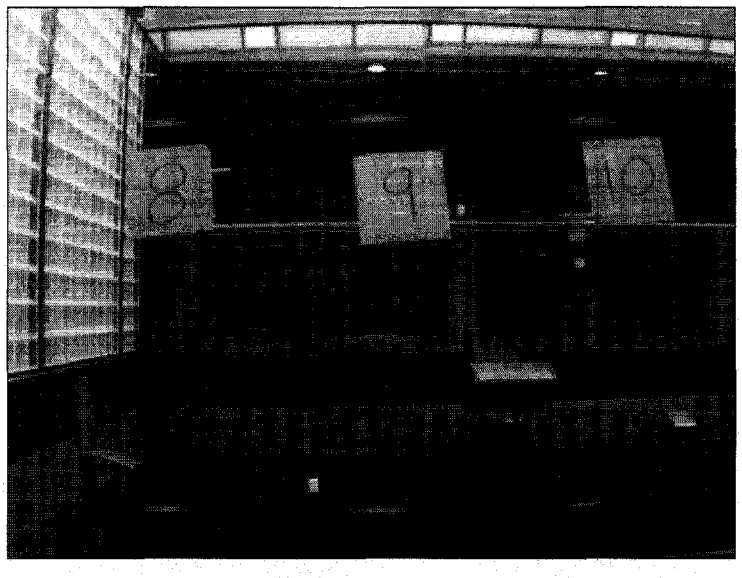

Figure 5.2: An ActiveMedia Pioneer 3 robot observes artificial landmarks on the wall from images taken using a mounted camera. Corners detected by the Harris Corner Detector are shown by cross marks. The numbers in the image refer to the landmarks identification number in the map.

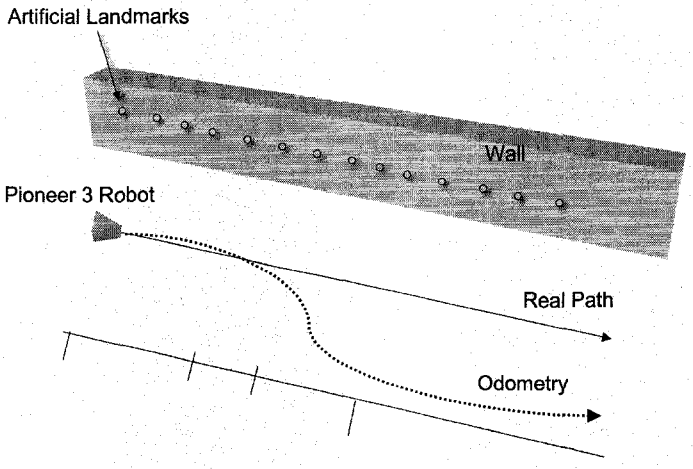

Figure 5.3: A diagram of our ActiveMedia Pioneer 3 robot moving down the hallway of the Department of Computing Science. A camera is mounted on the robot and faces sideways to the wall. The robot observes artificial landmarks placed on the wall. The solid line indicates the robot's real path and the dotted line represents the odometry data obtained from the robot's wheel encoders.

plane, where the robot's centre coordinates (x_v, y_v) and orientation (θ) are estimated with respect to the world reference frame. The robot's start position is the origin of the world reference frame. Landmarks are corner points in the three-dimensional world reference frame and are represented by $X^W = [x^W, y^W, z^W]$. Both the robot and the landmarks are registered in the same reference frame. The landmark positions are transformed into the camera reference frame so they can be incorporated into the localization algorithm. Detailed information about the different reference frame

transformations and other issues related to the localization algorithm implementation can be found in Appendix A.1 and A.2.

5.1.2 Performance Evaluation

The localization algorithm we use is a particle filter based algorithm which uses a mixture proposal distribution. The algorithm is essentially a FastSLAM 2.0 algorithm with a known map. With known landmark positions, each particle contains only the robot's pose. We use 50 particles for each experimental run. For each run, we compare the algorithm's point-estimate of the robot pose, which is the average value over 50 particles, to the ground truth. The accumulated error ϵ is the accumulated distance between the true pose and the estimated pose over the robot's trajectory. We use the accumulated localization error to evaluate the performance of the localization algorithm. Let $x_i^v = [x_{vi}, y_{vi}, \theta_i]$ be the true robot position, and $\hat{x}_i^v = [\hat{x}_{vi}, \hat{y}_{vi}, \hat{\theta}_i]$ be the estimated robot position at each time step i. We define the accumulated error as:

$$\epsilon = \sum_{i=0}^{n} \sqrt{(x_{vi} - \hat{x}_{vi})^2 + (y_{vi} - \hat{y}_{vi})^2} \tag{5.1}$$

In order to verify our assumption that the accuracy of a sound estimator should increase with the accuracy of its sensor model, we not only use the sensor model we built but also a set of inaccurate sensor models that either overestimate or underestimate the measurement noise standard deviation. For each experimental setup, 20 runs are carried out to account for the randomness of the particle filter. The mean and the standard deviation of the accumulated localization errors over the 20 runs will be shown in the result section for each experiment.

5.2 Simulation Results

In our simulation, we use the same intrinsic parameters that we used for the real environment experiment. The uncertainties of the three-dimensional landmark positions and the transformation between the robot centre and the camera can be ignored. This controlled environment allows an accurate analysis of the effects of sensor model error. We consider the following statement: the accuracy of any probabilistic robot navigation algorithm should increase with the accuracy of the probabilistic camera sensor model. We simulate the true noisy sensor model by adding Gaussian noise to the estimated image points obtained from the pinhole camera model. We then repeat the experiment using various sensor models with different measurement noise covariance matrices. The optimal performance of the localization algorithm is expected if we use the true sensor model, where the measurement noise covariance $R_k = diag(\delta u^2, \delta v^2)$ is equal to the added Gaussian noise and the measurement noise standard deviations δu and δv are equal.

Two sets of experiments are carried out. As mentioned before, we conduct 20 runs of each experiment and the results are averaged. The true measurement noise covariance R_k is equal to

$diag(1^2, 1^2)$. The robot travels along the straight line for approximately 15 meters. For the first experiment, the robot can only observe landmarks that are approximately one meter away. In the second experiment, the robot travels along the same path and can only observe distant landmarks that are approximately two meters away. The simulation results using the close landmarks are shown in Figure 5.4 and the corresponding data is listed in Table 5.1. We can see that using the true camera sensor model with the measurement noise standard deviation $\delta u = 1$ pixel gives 2.39 meters of accumulated error (see Equation 5.1), which is 16 percent of the length of the robot's total traveling path. However, the optimal performance with 2.33 meters of accumulated error is given when using a sensor model with the measurement noise standard deviation $\delta u = 0.8$ pixels. In addition, an optimistic sensor model with measurement noise standard deviation $\delta u = 0.1$ pixels gives 62.03 meters of accumulated error, which is about 26 times worse than using the true sensor model. A pessimistic sensor model with measurement noise standard deviation $\delta u = 100$ pixels also gives non-optimal performance. Its accumulated error is 8.00 meters, which is still approximately three times larger than using the true sensor model. Sensor models of δu from half to five times larger than the true δu provide similar accumulated errors to the true sensor model as shown in Table 5.1. The results demonstrate that using our camera sensor model gives small accumulated error although not optimal for close landmarks simulation.

The simulation results using distant landmarks are shown in Figure 5.5 with the corresponding data in Table 5.2. Using the true camera sensor model with the measurement noise standard deviation $\delta u = 1$ pixel gives 9.29 meters of accumulated error, which is 62 percent of the length of the robot's total traveling path. When using a sensor model with the measurement noise standard deviation $\delta u =$ 3 pixels, the optimal performance with accumulated error 6.26 meters can be achived. A pessimistic sensor model with δu five times larger than the true δu gives 1.7 times worse performance than using the true sensor model. An optimistic sensor model with δu 10 times smaller than the true δu gives 22 times worse performance than using the true sensor model. The accumulated error increases dramatically when using sensor models with measurement noise standard deviation smaller than the true measurement noise standard deviation. Pessimistic sensor models with measurement noise standard deviation five times less than the true measurement noise standard deviation gives similar accumulated errors to the true sensor model does as shown in Table 5.2.

Under the same experimental setup as the close landmark simulation, the accumulated localization error of distant landmarks is on average eight times larger than using the close landmarks. As shown in Figure 5.6, the dotted line represents the accumulated error ratio between the distant landmarks and the close landmarks with respect to sensor models using various levels of measurement noise standard deviation. On average, the ratio is approximately 80 percent as shown by the straight line. Even with the same true sensor model, the performance of the close landmark simulation is approximately 40 percent better than that of the distant landmark simulation in terms of accumulated error.

Since a camera can only measure bearing information, two landmarks $X_A = [x_A^C, y_A^C, z_A^C]$ and $X_B = t[x_A^C, y_A^C, z_A^C]$, that are projected onto the same image point, are equivalent to the camera. Why does the landmark range influence the performance of the localization algorithm? We find that in the update step of the localization algorithm, the range scales the Jacobian matrix H_k of the sensor model with respect to the robot pose. The Jacobian matrix H_k influences the proposal uncertainty of the updated robot pose from which the current robot pose is sampled. When t is larger than one, landmark B is further away in the camera reference frame. Thus, the relationship between the two Jacobian matrices is $H_{kB} = H_{kA}/t$. (The derivative of the Jacobian matrix H_k can be found in Appendix A.2.) As a result, the uncertainty of the updated robot pose using the sensor measurement of landmark X_B is larger than using the sensor measurement of landmark X_A. Therefore, the localization algorithm treats the close and distant landmarks differently, where more uncertainty is introduced by the distant landmarks than the close landmarks.

From the above simulation results, we can draw four conclusions. First, for both close and distant landmark situations, the localization algorithm is nearly optimal when the true camera sensor model is used. Secondly, being pessimistic about the measurement information also gives good performance of the localization algorithm. Being optimistic about the observation information can lead to poor performance and it is found to be more so when using distant landmarks. For example, if the localization algorithm is over confident about the estimated sensor measurement, which is not the true measurement, it uses this inaccurate measurement information to update the predicted robot pose. Therefore, it is easy to update the robot pose to the wrong position. Thirdly, according to the result data listed in Tables 5.1 and 5.2, using distant landmarks gives on average 80 percent larger accumulated localization errors than using close landmarks. In addition, the standard deviation of the errors over all simulation runs for close landmarks are approximately 30 times smaller than those for distant landmarks, indicating that close landmarks provide more stable performance. Thus, care should be taken when using sensor measurements of the distant landmarks to correct the robot's predicted pose [9]. Finally, the simulation results reinforce the observation made in Section 4.2 that a camera sensor model with constant measurement noise variance is sufficient for all landmarks at various ranges.

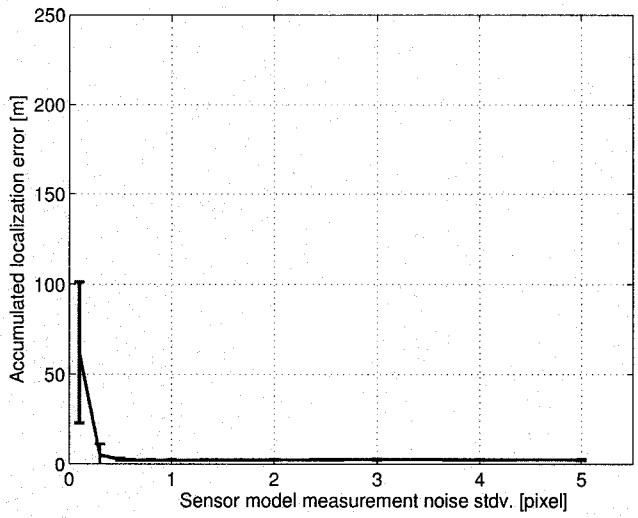

Figure 5.4: Results of the close landmark simulation. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel.

δu [pixel]	accumulated error [m]	standard deviation [m]
0.1	62.03	39.14
0.3	5.03	6.15
0.5	2.69	0.48
0.6	2.38	0.21
0.8	2.33	0.10
1	2.39	0.14
2	2.51	0.15
3	2.73	0.27
4	2.63	0.26
5	2.75	0.32
$\star$ 100	8.00	0.55

Table 5.1: Results of the close landmark simulation for landmarks approximately one meter away from the camera. The table shows the average and standard deviation of the accumulated localization error for various sensor models. Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The $\star$ refers to data not shown in Figure 5.4.

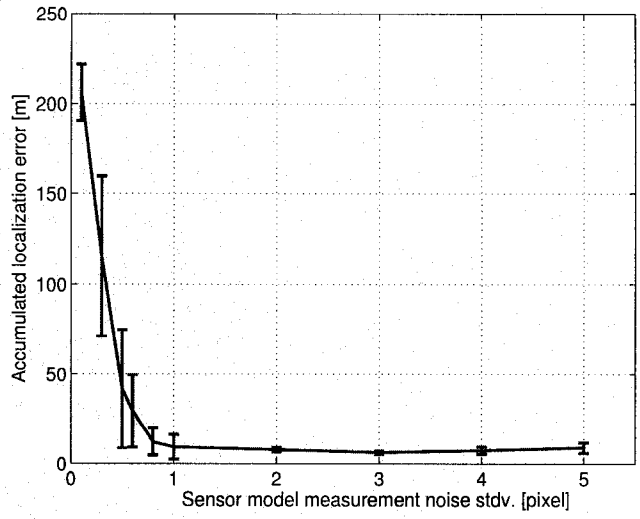

Figure 5.5: Results of the distant landmark simulation. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel.

δu [pixel]	accumulated error [m]	standard deviation [m]
0.1	206.20	15.63
0.3	115.52	44.15
0.5	41.79	33.17
0.6	29.26	20.15
0.8	12.20	7.44
1	9.29	6.82
2	7.75	1.35
3	6.26	0.88
4	7.40	2.00
5	8.94	2.98
$\star$ 100	16.25	2.12

Table 5.2: Results of the distant landmark simulation for landmarks approximately two meter away from the camera. The table shows the average and standard deviation of the accumulated localization error for various sensor models. Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The $\star$ refers to data not shown in Figure 5.5.

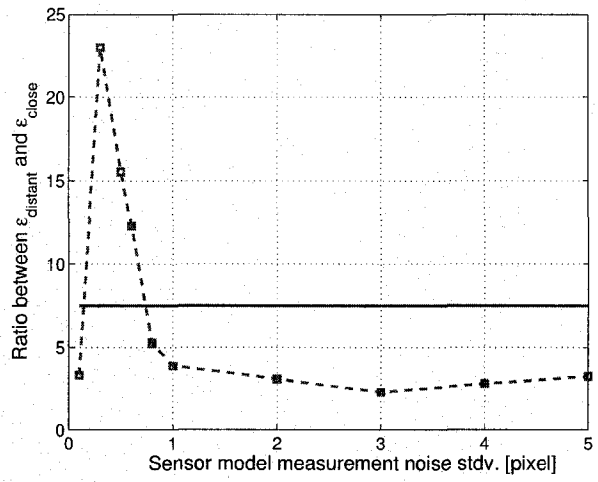

Figure 5.6: Comparison of accumulated localization errors between the distant and close landmarks in simulation. The dotted line shows the ratio between the accumulated error for the landmarks $\epsilon_{distant}$ and the accumulated error for the close landmarks ϵ_{close}. The straight line represents the average error ratio over all the sensor models using various levels of measurement noise standard deviation.

One may expect that an optimal performance of a probabilistic robot navigation algorithm to occur when the true camera sensor model is used and the accuracy of the algorithm should increases with the accuracy of the probabilistic camera sensor model. However, this is not entirely supported by the simulation results. Optimal performance is more likely to occur when using pessimistic camera sensor models than using the true sensor model. We suggest that this results are due to the inconsistency problem of the robot navigation algorithm rather than the sensor model, as mentioned in Section 2.2.2. The inconsistent localization algorithm tends to underestimate the uncertainty of the robot pose. Therefore, the algorithm is in favour of a pessimistic sensor model to overcome its optimistic estimation. Tim Bailey *et al.* show that the improper resampling process may be the cause this inconsistency problem [5].

How does the localization algorithm behave when a pessimistic sensor model or an optimistic sensor model are used? Figures 5.7 and 5.8 show the localization simulation results in a three dimensional environment. When the measurement noise covariance matrix $R_k = diag(100^2, 100^2)$, sensor measurements become noisy and inaccurate, causing the algorithm to depend less on the measurement data and more on the odometry data. Therefore, the estimated robot path follows to a great degree what the odometry data indicates, and it is different from the robot's real path. The set of particles, however, can still keep the average robot pose close to the ground truth as shown in Figure 5.7. When the measurement noise standard deviation is very small, for example when $R_k = diag(0.1^2, 0.1^2)$, the algorithm is over confident about the accuracy of the estimated sensor measurements, which causes the robot's pose to be updated using inaccurate estimated sensor measurements. In the subsequent update step, the particle set spreads over the three-dimensional

space to cover the possible true robot pose. Therefore, the estimated robot pose, which is the average value of the 50 particles, actually jumps considerably back and forth from the robot's real pose. This causes the estimated robot path to lose its track, as shown in Figure 5.8.

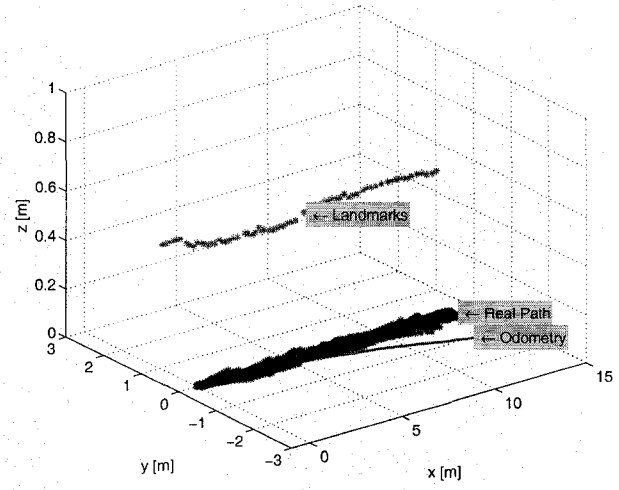

Figure 5.7: Localization simulation in a three-dimensional environment. The robot moves along a straight line and observes close landmarks about one meter away. The length of the robot's real path is approximately 15 meters. A pessimistic sensor model with the measurement noise covariance matrix $R_k = diag(100^2, 100^2)$ is used. The straight line represents the robot's true path, the curve line is the robot's estimated path based on the odometry data, and a set of 50 particles show the robot's estimated path based on the localization algorithm.

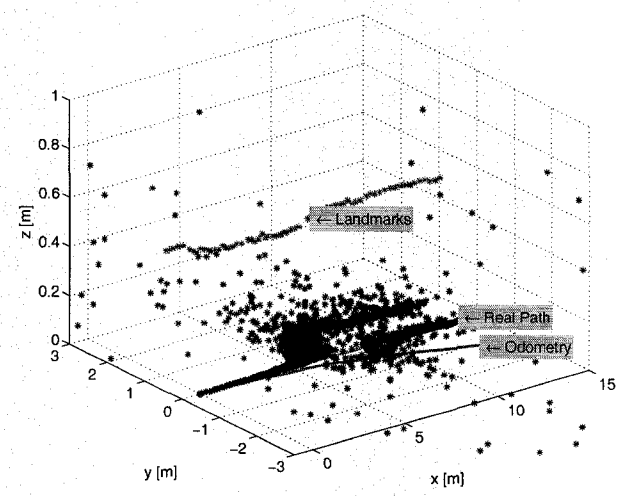

Figure 5.8: Localization simulation in a three-dimensional environment. The robot moves along a straight line and observes close landmarks about one meter away. The length of the robot's real path is approximately 15 meters. A pessimistic sensor model with the measurement noise covariance matrix $R_k = diag(0.1^2, 0.1^2)$ is used. The straight line represents the robot's true path, the curve line is the robot's estimated path based on the odometry data, and a set of 50 particles show the robot's estimated path based on the localization algorithm.

5.3 Experimental Results

The experimental setup for our real environment experiments is similar to that of the simulations. In the real environments, data is often corrupted by the uncertainty of the three-dimensional landmark positions and the noisy translation between the robot pose and the camera. Therefore, it is unrealistic to assume perfect knowledge of the three-dimensional landmark positions. In the particle filter based localization algorithm, both the uncertainty of the landmark position and the measurement noise covariance matrix influence the accuracy of the sensor measurements (see Appendix A.3). Because we manually measure the landmark positions and the robot's real path, five centimetre noise is added to the uncertainty of the landmark position. Thus, the uncertainty of the landmark position P_l is equal to $diag(0.05^2, 0.05^2, 0.05^2)$.

The experimental results from Section 4.3 indicate that a measurement noise covariance matrix $R_k = diag(1^2, 1^2)$ for both close landmarks and distant landmarks is sufficient. In order to compare the performance with the true camera sensor model, we use sensor models with measurement noise standard deviation ranging from 0.1 pixels to five pixels. Two sets of experiments are carried out. In the first set, the robot observes only close landmarks that are approximately one meter away. In the second set, the robot observes distant landmarks that are approximately two meters away.

The experimental results of the close landmarks are shown in Figure 5.9 with its corresponding data in Table 5.3. We can see that sensor models with measurement noise standard deviation ranging from 0.1 pixels to five pixels all give accumulated error of approximately five meters. Even the standard deviation of the accumulated error is quite small, which is on average 0.34 meters. However, a pessimistic sensor model with a measurement noise covariance matrix $R_k = diag(100^2, 100^2)$ provides approximately nine meters of accumulated error, which is 1.7 times worse than using the true sensor model.

The experimental results of the distant landmark experiment are shown in Figure 5.10 and the corresponding data is listed in Table 5.4. We can see that our camera sensor model with measurement noise standard deviation of one pixel gives about 9.9 meters of accumulated error. An optimal performance with accumulated error of 9.57 is given when using a sensor model with the measurement noise standard deviation $\delta u = 5$ pixels. Sensor models with δu or δv ranging from 0.1 pixel to five pixels also lead to similar performance. More specifically, the average accumulated errors over those models vary from approximately 9.57 meters to 11.49 meters. Considering that the average standard deviation of the accumulated error is about 2.7 meters, we can claim that all these sensor models give similar performance. Even using a pessimistic sensor model with $R_k = diag(100^2, 100^2)$, the accumulated error is still approximately 1.2 times worse than the optimal performance of 9.57 meters of accumulated error. Similar to the simulation results, according to the data in both Tables 5.3 and 5.4, the average accumulated error and their standard deviation for all the sensor models are larger when using close landmarks than using distant landmarks. As shown in Figure 5.11, the dotted line represents the accumulated error ratio between the distant landmarks and the close land-

marks with respect to sensor models using various levels of measurement noise standard deviation. On average, the ratio is approximately 18 percent as shown by the straight line.

From the experimental results, we are able to draw three conclusions. First, for close landmark simulation, both the accurate sensor model and the inaccurate sensor models give similar performance in terms of the accumulated error. This observation also holds true for the distant landmark simulation. Secondly, under the same experimental setup, using close landmarks instead of distant ones gives better performance in terms of the accumulated localization error and the standard deviation. For example, using the same sensor model with measurement noise of one pixel, the close landmark experiment gives 5.52 meters of accumulated error while the distant landmark experiment gives 9.9 meters. If we want to make good use of the distant landmarks, they should be incorporated into the robot navigation system differently from the close landmarks. This topic is beyond the scope of this thesis, but related work can be found in recent work by Civera *et al.* [9].

Finally, the uncertain component of a camera sensor model is not as critical as we would expect since the noisy landmark positions in the real environment influences more the performance of the localization algorithm. Compared to the simulation results in Tables 5.1 and 5.2, we conclude that the uncertainty of the three-dimensional landmark position dominates the uncertainties of the sensor model in the performance of the localization algorithm. In order to investigate this observation, we run the simulation again to simulate the noisy landmark positions by adding five centimeters uncertainty to the three-dimensional landmark positions. As we can see from the Figures 5.12 and 5.13, using the same landmark position uncertainty $P_l = diag(0.05^2, 0.05^2, 0.05^2)$, the simulation results are similar to the experimental results in the Figures 5.9 and 5.10. From the comparison of the resulting plots, the same observation can be drawn: sensor models using various levels of measurement noise standard deviation can be used for visual localization algorithm.

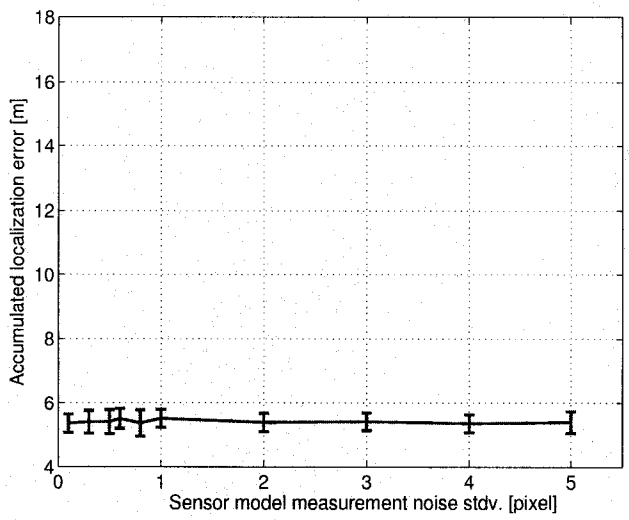

Figure 5.9: Results of the close landmark experiment. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$.

δu [pixel]	accumulated error [m]	standard deviation [m]
0.1	5.36	0.29
0.3	5.40	0.35
0.5	5.41	0.37
0.6	5.51	0.31
0.8	5.37	0.41
1	5.52	0.28
2	5.39	0.29
3	5.42	0.28
4	5.36	0.28
5	5.40	0.34
$\star$ 100	9.30	0.55

Table 5.3: Results of the close landmark experiment for landmarks approximately one meter away from the camera. The table shows the average and standard deviation of the accumulated localization error for various sensor models. Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$. The $\star$ refers to data not shown in Figure 5.9.

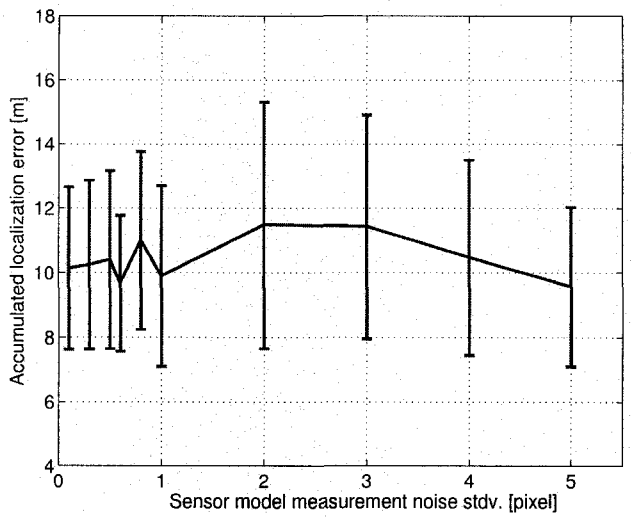

Figure 5.10: Results of the distant landmark experiment. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$.

δu [pixels]	accumulated error [m]	standard deviation [m]
0.1	10.14	2.51
0.3	10.25	2.61
0.5	10.41	2.76
0.6	9.67	2.10
0.8	11.01	2.76
1	9.90	2.79
2	11.49	3.83
3	11.43	3.48
4	10.78	3.03
5	9.57	2.46
$\star$ 100	12.14	1.27

Table 5.4: Results of the distant landmark experiment for landmarks approximately two meter away from the camera. The table shows the average and standard deviation of the accumulated localization error for various sensor models. Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$. The $\star$ refers to data not shown in Figure 5.10.

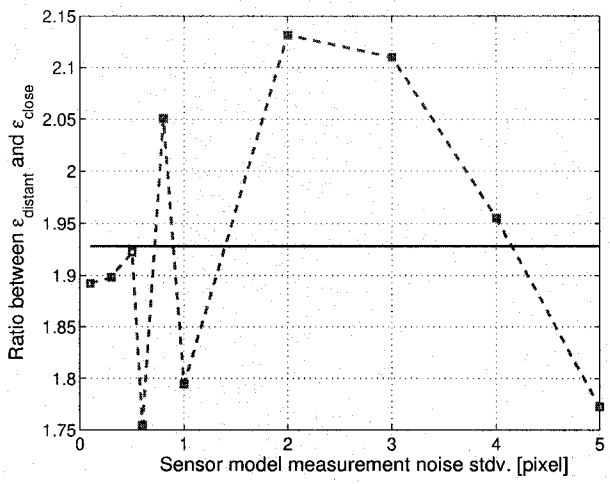

Figure 5.11: Comparison of accumulated localization errors between the distant and close landmarks. The dotted line shows the ratio between the accumulated error for the distant landmarks $\epsilon_{distant}$ and the accumulated error for the close landmarks ϵ_{close}. The straight line represents the average error ratio over all the sensor models using various levels of measurement noise standard deviation.

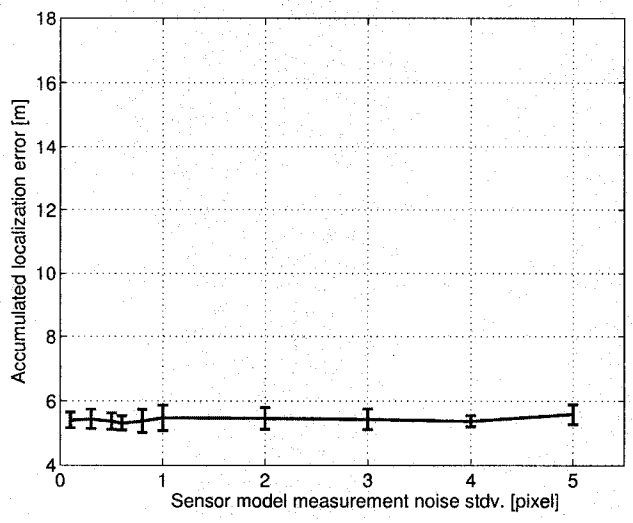

Figure 5.12: Results of the close landmark simulation. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$.

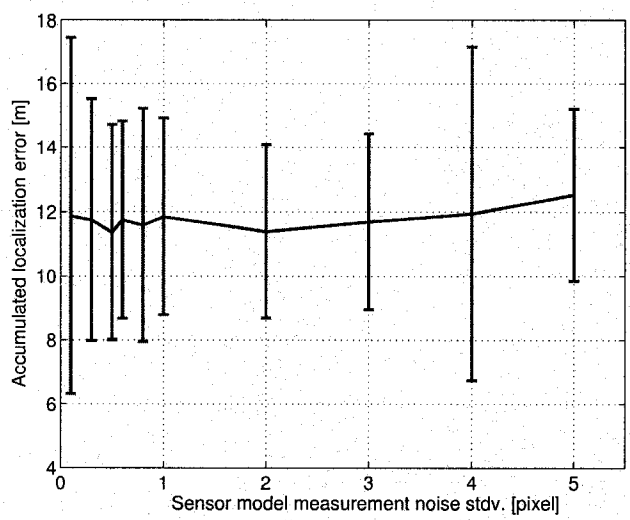

Figure 5.13: Results of the distant landmark simulation. The plot shows the accumulated localization error with respect to the camera sensor models using various levels of measurement noise standard deviation along x and y directions (δu and δv). Sensor models using δu or δv smaller than one pixel are optimistic models, whereas pessimistic sensor models uses δu or δv larger than one pixel. The uncertainty of the landmark position is $P_l = diag(0.05^2, 0.05^2, 0.05^2)$.

5.4 Summary

In this chapter, we use the sensor model built in Chapter 4.3 to run a particle filter based localization algorithm. To analyze the performance of using the true sensor model, a set of optimistic sensor models and pessimistic sensor models are used in the localization algorithm. We discuss the simulation results of the localization algorithm and analyze the effect of the sensor model on the performance of the algorithm. We also discuss the discrepancy between the simulation results and the experimental results. It is found that in the noisy real environment, the uncertain component of the camera sensor model, the measurement noise, is not as critical as in the simulation. This is because that the uncertainties of landmark positions dominate the measurement noise in the performance of the localization algorithm. Since a probabilistic camera sensor model is independent of the robot navigation algorithm, the simulation and experimental results in this chapter can also be applied to bearing-only visual SLAM.

Chapter 6

Conclusions

In the previous chapter, we described the experiments performed to evaluate our probabilistic camera sensor model. In this chapter, a general conclusion summarizes our method for building a probabilistic camera sensor model and the major contributions of this thesis. We also discuss the limitation of the current work and provide suggestions for possible improvements in the future.

6.1 Camera Sensor Model

A probabilistic camera sensor model that accounts for uncertainty in the sensor measurements is critical for solving the SLAM problem. A probabilistic camera sensor model consists of a deterministic component and an uncertain component. Extensive work has been carried out towards building a pinhole camera model, however, limited work has studied how to model the measurement noise. The measurement noise determines the accuracy of the sensor measurements obtained from a camera sensor model. Thus, building an accurate camera sensor model that properly captures the noise in the sensor measurements is important for the performance of bearing-only visual SLAM.

In this thesis, we present our method for building a probabilistic camera sensor model for bearing-only visual SLAM. This method is inspired by the camera calibration process and the measurement noise definition, which is the difference between the actual measurement and the estimated measurement. In order to validate our probabilistic camera sensor model, we test the model on a localization problem in both simulation and real environments.

In the field of using the probabilistic approach to solving visual robot navigation, researchers often use a constant measurement noise covariance matrix. However, no detailed information is given to explain how they choose this value for the matrix. Our research provides a detailed study of modelling the measurement noise covariance matrix used in bearing-only visual SLAM. We verify the assumption that the measurement noise can be modelled by a Gaussian distribution, which is required for the EKF-SLAM and FastSLAM algorithms. In addition, we conclude that a camera sensor model with constant measurement noise standard deviation is sufficient for all landmarks regardless of their distance from the camera.

This thesis identifies the significance of a probabilistic camera sensor model and analyzes the sensor model error on the performance of visual localization. Simulation results show that our camera sensor model provides good performance in terms of the accumulated localization error. An optimistic camera sensor model tends to give poor performance than a pessimistic camera sensor does. More interestingly, sensor model with different levels of measurement noise standard deviation larger than the true measurement noise standard deviation can also give good performance. Finally, our experimental results demonstrate that an accurate model of the measurement noise is not critical for noisy environments as long as proper uncertainties for the landmark positions are used.

We find that landmarks at various distances from the camera have different influence on the performance of the visual localization algorithm. Under the same experimental setup, close landmarks give smaller accumulated localization errors than distant landmarks. In addition, using close landmarks are more stable in the presence of randomness in the localization algorithm. We suggest using close landmarks rather than distant landmarks to correct both the predicted robot pose and the map.

Although we present our probabilistic camera sensor model in the context of bearing-only visual SLAM, our sensor model and the resulting conclusions are also applicable to any other probabilistic robot navigation algorithm. In conclusion, this thesis provides us with a possible technique for constructing a probabilistic camera sensor model and also gives insight into general camera sensor models. This is the first time that a theoretical study of a probabilistic camera sensor model has been undertaken.

6.2 Limitations and Future Work

At this time, we will address a number of limitations in our current work and suggest possible directions for future research. Our method for building a probabilistic camera sensor model suffers from its inability to fully capture the reprojection error, which is the difference between the measured image point and the estimated image point. In our method, the three-diemensional locations of the corner points on the checkerboard in the camera's reference frame are calculated from the camera's intrinsic parameters. Then, we project the corner points back onto the image plane through the same camera's intrinsic parameters. Therefore, the influence of the inaccurate pinhole camera model has been cancelled out through this method. Ideally, the locations of the corner points should be determined independently from the camera sensor model. Future work requires a highly accurate setup to measure the transformation between the camera and the checkerboard.

The experiments on localization in the real environment suffer from noisy landmark positions. Future work in conducting experiments could benefit from constructing a noise-free environment. Our simulation and experimental results on localization contradict the general assumption that the accuracy of a probabilistic estimator should increase with the accuracy of the sensor model. Future research towards properly integrating the sensor model into the SLAM framework is required to realize a sound estimator for solving the SLAM problem.

Although a sensor model should be independent of its statistical estimator, the same sensor model can be used for both bearing-only visual SLAM and visual localization. Future research requires experiments designed specifically for solving a SLAM problem. In addition, our sensor models are only tested for indoor environment. It is worthwhile to verify they can also be applied for outdoor robot navigation.

Bibliography

[1] Abdel-Aziz, Y. I., Karara, and H. M. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proceedings of the Symposium on Close-Range Photogrammetry (pp. 1-18).*, 1971.

[2] Tim Bailey. Constrained initialization for bearing-only slam. In *IEEE International Conference on Robotics and Automations*, 2003.

[3] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localisation and mapping (slam): Part ii - state of the art. In *Robotics and Automation Magazine*, 2006.

[4] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the ekf-slam algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006*, Beijing, China, 2006.

[5] Tim Bailey, Juan Nieto, and Eduardo Nebot. Consistency of the fastslam algorithm. In *IEEE International Conference on Robotics and Automation*, 2006.

[6] Jean-Yves Bouguet. Camera calibration toolbox for matlab, 2006.

[7] Harris C and Stephens M. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, 1988.

[8] Ouyang Cheng, Wang Guangzhi, Zhang Quan, Kang Wei, and DING Hu. Evaluating harris method in camera calibration. In *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005.

[9] Javier Civera, Andrew J. Davison, and J. M. M. Montiel. Inverse depth to depth conversion for monocular slam. In *IEEE International Conference on Robotics and Automations*, 2007.

[10] A. Davison. Real-time simultaneous localization and mapping with a single camera. In *Proc. International Conference on Computer Vision*, 2003.

[11] Andrew. J. Davison. Mobile robot navigation using active vision. Ph.d. thesis, Department of Engineering Science, University of Oxford, 1998.

[12] Matthew Deans and Martial Hebert. Experimental comparison of techniques for thesisization and mapping using a bearing only sensor. In *Proceeding of the ISER'00 Seventh International Symposium on Experimental Robotics*, 2000.

[13] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous thesisisation and map building (slam) problem. In *Working notes of ICRA'2000 Workshop W4: Mobile Robot Navigation and Mapping*, April 2000.

[14] G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous thesisization and map building (slam) problem. In *IEEE Trans. Robot. Automat.*, volume 17-3, pages 229–241, June 2001.

[15] A. Doucet, J. F. G de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proc. of the Conference on Uncertainty in Artificial Intelligence*, 2000.

[16] H. Durrant-Whyte, S. Majumder, S. Thrun, M. de Battista, and S. Scheding. A bayesian algorithm for simultaneous localization and map building. In *Proceeding of the 10th International Symposium of Robotics Research (ISRR'01)*, Lorne, Australia, 2001.

[17] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. In *Robotics and Automation Magazine*, 2006.

[18] Daniel Gohring and Hans-Dieter Burkhard. Multi robot object tracking and self localization using visual percept relations. In *Proceeding of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[19] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. University Press, Cambridge, 2004.

[20] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *CVPR97*, 1997.

[21] A.M. Jazwinsky. *Stochastic Processes and Filtering Theory*. New York, 1970.

[22] Simon J. Julier. Consistency and slam. Technical report, Department of Computer Science, University College London, 2006.

[23] N. M. Kwok and G. Dissanayake. An efficient multiple hypothesis filter for bearing-only slam. In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04)*, 2004.

[24] T. Lemaire, S. Lacroix, and J. Sola. A practical 3d bearing-only slam algorithm. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 05)*, 2005.

[25] David G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, 2004.

[26] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-slam: A factored solution to the simultaneous localization and mapping problem. In *Proc. AAAI Nat. Conf. Artif. Intell.*, pages 593–598, 2002.

[27] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-slam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proc. Int. Joint Conf. Artif. Intell.*, pages 1151–1156, 2003.

[28] Sebastian, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.

[29] Robert Sim and James J. Little. Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters. In *Proceeding of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[30] R. C. Smith and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167–193, Berlin-Heidelberg, 1990.

[31] Eduard Sojka. A new algorithm for detecting corners in digital images. In *Proceedings of the 18th spring conference on Computer graphics*, 2002.

[32] B. Triggs. Autocalibration from planar scenes. In *Proc. 5th European Conference on Computer Vision*, Freiburg, Germany, June 1998.

[33] R. E. Walpole, R. H. Myers, S. L. Mayers, K. Ye, and K. Yee. Probability and statistics for engineers and scientists. In *7-th edition, Upper Saddle River, N. J., Prentice Hall*, 2002.

[34] Greg Welch and Gary Bishop. An introduction to the kalman filter. In *ACM Computer Graphics (SIGGRAPH' 2001)*, 2001.

[35] Jing Wu and Hong Zhang. Camera sensor model for visual slam. In *Fourth Canadian Conference on Computer and Robot Vision*, May 2007.

[36] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV99*, 1999.

[37] Marco Zucchelli and Jana Kosecka. Motion bias and structure distortion induced by calibration errors. In *BMVC Conference*, Machester,England, 2001.

Appendix A

Implementation of Particle Filter Based Localization

In this appendix, we provide details of the implementation of the particle filter based localization algorithm, including the motion model, the transformation between the different reference frames, and the innovation covariance matrix.

A.1 Motion Model

The motion model $p(x_k|x_{k-1}, u_k)$ describes the probability of system state x_k, given the previous system state x_{k-1} and the odometry data u_k. Because only the robot pose is estimated in a localization problem, the system state x_k is the robot pose at each time step k. The current robot pose $x_k^v = (x_{vk}, y_{vk}, \theta_k)$ can be calculated from the previous robot pose $x_{k-1}^v = (x_{vk-1}, y_{vk-1}, \theta_{k-1})$ and the odometry data, which can be obtained from the robot's wheel encoders. We consider $(\phi_{rot1}, \phi_{trans}, \phi_{rot2})$ as relative motion information (see Figure A.1), which is defined by the following:

$$\begin{aligned} \phi_{rot1} &= arctan(y_{vk} - y_{vk-1}, x_{vk} - x_{vk-1}) - \theta_{k-1} \\ \phi_{trans} &= \sqrt{(x_{vk} - x_{vk-1})^2 + (y_{vk} - y_{vk-1})^2} \\ \phi_{rot2} &= \theta_k - \theta_{k-1} - \phi_{rot1} \end{aligned} \tag{A.1}$$

Notice that rotations should be normalized to $[-\pi, \pi]$. The translation and rotations are noisy, and their uncertainties are assumed to follow a Gaussian distribution.

$$\begin{aligned} \sigma^2_{\phi_{rot1}} &= \alpha_1 {\phi_{rot1}}^2 + \alpha_2 \phi^2_{trans} \\ \sigma^2_{\phi_{trans}} &= \alpha_3 \phi^2_{trans} + \alpha_4 \phi^2_{rot1} + \alpha_4 \phi^2_{rot2} \\ \sigma^2_{\phi_{rot2}} &= \alpha_1 \phi^2_{rot2} + \alpha_2 \phi^2_{trans} \end{aligned} \tag{A.2}$$

The process noise covariance matrix Q_k is represented as :

$$\begin{bmatrix} \sigma^2_{\phi_{rot1}} & 0 & 0 \\ 0 & \sigma^2_{\phi_{trans}} & 0 \\ 0 & 0 & \sigma^2_{\phi_{rot2}} \end{bmatrix} \tag{A.3}$$

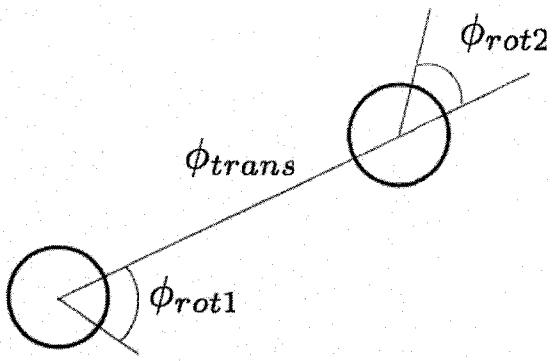

Figure A.1: Odometry motion model. The robot's motion from time step $k-1$ to k is approximated by translation ϕ_{trans}, rotations ϕ_{rot1} and ϕ_{rot2}.

Scale parameters $[\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ determine how noisy the odometry information is. In order to obtain a correct motion model for all our experiments, we simulate the odometry data by corrupting the robot's real motion information with predefined scale parameters.

A.2 Transformation Between Reference Frames

The estimation of the localization algorithm is in the world reference frame, but we also have the robot reference frame and the camera reference frame. Figure A.2 shows a schematic diagram of the robot in the process of observing a landmark $X^C = [x^C, y^C, z^C]$ and the relationship between the different reference frames.

The transformations between the different reference frames are represented by a homogeneous matrix. For example, the transformation between the world reference frame and the robot reference frame can be defined as:

$$\begin{bmatrix} R^{WR} & T^{WR} \\ 0^T & 1 \end{bmatrix} \tag{A.4}$$

where R^{WR} is a 3×3 matrix, T^{WR} a translation 3-vector, and 0^T a null 3-vector. The inverse transformation can be obtained from the inverse of this matrix.

As we mentioned in Chapter 2.6, the sensor model is linearized in order to incorporate it into the robot navigation algorithms. The deterministic component of the sensor model is a pinhole camera model, which is defined in the camera reference frame. Therefore, the Jacobian matrix H_k of the sensor model with respect to the robot pose has to be propagated into the world reference frame. To simplify the calculation, the Jacobian matrix H_k can be generated by:

$$H_k = \frac{\partial h(x_k)}{\partial x_k^v} = (\frac{\partial h(x_k)}{\partial X^C}) R^{CR} (\frac{\partial X^R}{\partial x_k^v}) \tag{A.5}$$

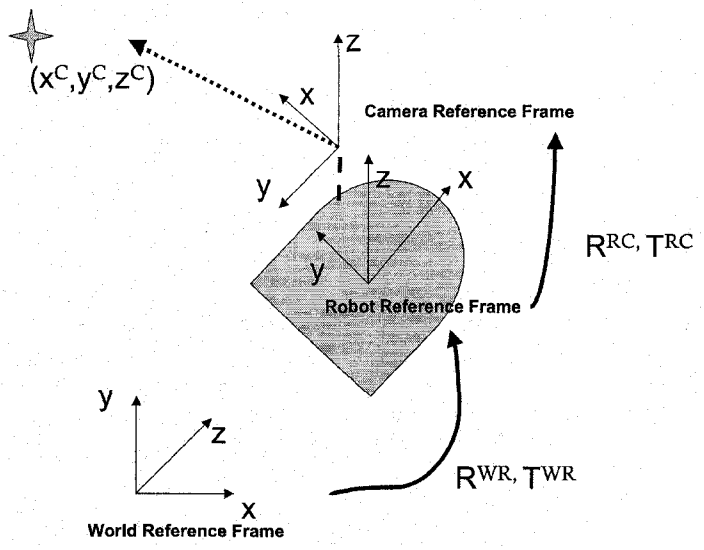

Figure A.2: The relationship between the world reference frame, the robot reference frame and the camera reference frame. R^{WR} and T^{WR} are the rotation and translation from the world coordinate frame to the robot reference frame. R^{RC} and T^{RC} are the rotation and translation from the robot reference frame to the camera reference frame.

where the first term

$$\frac{\partial h(x_k)}{\partial X^C} = \frac{1}{z^C}\begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \end{bmatrix}$$

$$\begin{aligned} J_{11} &= f_x(1 + k_1(3(\frac{x^C}{z^C})^2 + (\frac{y^C}{z^C})^2)) \\ J_{12} &= 2f_x\frac{x^C}{z^C}\frac{y^C}{z^C} \\ J_{13} &= -f_x(\frac{x^C}{z^C} + 3k_1(\frac{x^C}{z^C})^3 + 3k_1\frac{x^C}{z^C}(\frac{y^C}{z^C})^2) \\ J_{21} &= 2f_y\frac{x^C}{z^C}\frac{y^C}{z^C} \\ J_{22} &= f_y(1 + k_1((\frac{x^C}{z^C})^2 + 3(\frac{y^C}{z^C})^2)) \\ J_{23} &= -f_y(\frac{y^C}{z^C} + 3k_1(\frac{y^C}{z^C})^3 + 3k_1\frac{y^C}{z^C}(\frac{x^C}{z^C})^2) \end{aligned} \tag{A.6}$$

The rotation from the camera to robot centre differs from different experimental setup. The last term can be defined by:

$$\frac{\partial X^R}{\partial x_k^v} = \left[\frac{\partial X^R}{\partial x_{vk}}, \frac{\partial X^R}{\partial y_{vk}}, \frac{\partial X^R}{\partial \theta_k}\right] \tag{A.7}$$

where

$$
\begin{aligned}
\frac{\partial X^R}{\partial x_{vk}} &= \begin{bmatrix} -cos\theta_k \\ sin\theta_k \\ 0 \end{bmatrix} \\
\frac{\partial X^R}{\partial y_{vk}} &= \begin{bmatrix} -sin\theta_k \\ -cos\theta_k \\ 0 \end{bmatrix} \\
\frac{\partial X^R}{\partial \theta_k} &= \frac{\partial R^{RW}}{\partial \theta_k} \begin{bmatrix} x^W \\ y^W \\ z^W \end{bmatrix} + \begin{bmatrix} -cos\theta_k y^W + sin\theta_k x^W \\ sin\theta_k y^W + cos\theta_k x^W \\ 0 \end{bmatrix} \\
\frac{\partial R^{RW}}{\partial \theta_k} &= \begin{bmatrix} -sin\theta_k & cos\theta_k & 0 \\ -cos\theta_k & -sin\theta_k & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{A.8}
$$

A.3 Innovation Covariance Matrix

The measurement noise is represented in the SLAM algorithm using covariance matrix R_k. In both EKF-SLAM and FastSLAM 2.0, S_k is the covariance matrix of the innovation, which is the difference between the actual measurement and the estimated measurement from the estimated landmark position. The covariance of the innovation is calculated as:

$$S_k = H_k P_k^- H_k^T + R_k \tag{A.9}$$

where H is the Jacobian of $h(x_k)$ with respect to x_k, P_k^- is the system state covariance matrix, $H_k P_k^- H_k^T$ is the uncertainty of the system state, and R_k is the measurement noise covariance matrix. Without the effect of the noisy robot pose and landmark positions, when $H_k P_k^- H_k^T$ is equal to zero, the covariance matrix of innovation is equal to R_k.

The covariance matrix of the innovation influences the accuracy of the sensor measurements when performing the localization algorithm in a real environment. In the particle filter based localization algorithm, for each update step, we assume perfect robot pose. Thus, the uncertainty of the system state is only the uncertainty of the landmark position in the particle filter based localization algorithm. In this thesis, we use P_l as the uncertainty of the landmark position to represent the system state covariance matrix P_k.