# STEREO VISION BASED THREE DIMENSIONAL SIMULTANEOUS LOCALISATION AND MAPPING

By
Damitha Chandana Herath

SUBMITTED IN FULFILMENT OF THE
REQUIREMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TECHNOLOGY, SYDNEY
August 2007

## CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Production Note:
Signature removed prior to publication.

Damitha Chandana Herath

# Acknowledgments

If there is a person who has the perseverance, patience and wisdom to guide me through the tenacious work of the PhD, trying at times, it is my supervisor Prof. Gamini "dissa" Dissanayake. This thesis is a tribute to his all encompassing wisdom and compassion, Thank you.

In the first year of my work at UTS, I had the privilege of working with Prof. Satoshi Takezawa, a visiting scholar and a gentleman from Japan to whom I am most grateful for his contributions to my early work. The following year, I spent the time at University of Peradeniya and would like to thank all the staff and students of Computer Engineering Department and the Computing Centre, Faculty of Engineering for their cooperation and support. On return from Sri Lanka Dr. Sarath Kodagoda became not only a supervisor of the thesis but also a close friend. I would like to thank Dr. Sarath, his wife Shermila and daughter for being greatest of friends and all the support given professionally and personally.

I would like to thank Dr. Raymond Kwok for being the first person to introduce me to 'Pioneer', SLAM and a host of other things related to my work. It was his coding that got me started. To Dr. Matt Gaston, for the high performance cluster, friendship and support with Linux. Also, I owe a special thank you to Dr. Jaime Vals Miro, Dr. Shoudong Huang and Dr. Jonathan Paxman for sharing their knowledge with me. I would like to thank Dr. Dikai Liu, Dr. Quang Ha and all the academics at CAS UTS for their support throughout my stay.

Time spent at UTS would have been less exciting if it wasn't for the great friendship extended by my colleagues. Especially, I would like to thank Cindy for her charm, Alen for his wit, Haye for his pc wizardry, Ash for his mischief, ZZ for the friendship, Matt for his profound silence, Weizhen for her other worldliness and all the other fellow CAS members who did not share the 'corner' with us. There are numerous friends in Australia and overseas to whom I owe a big thank you for the best of times shared together.

Ms. Gunasmin Lye for her kindness, help and most of all for keeping my Sri Lankan palate alive with her Polos, Chutney and many things in between. Also I like to thank

Prof Hung Nguyen, Dr. Prashanthi Hagare and all the members of the PG support team. Also thanks to Ms. Anya Van Eeuwen.

I have been funded by many contributions throughout my project. I am most grateful to the presidential scholarship scheme of Sri Lanka for the initial funding. Also would like to acknowledge the scholarships offered by the Centre for Autonomous Systems and the Faculty of Engineering, UTS, and the research grant from the National Science Foundation of Sri Lanka. I would like to acknowledge the support received from the ARC centre of Excellence for Autonomous Systems and its research director Prof Hugh Durrant-Whyte for the many illuminating seminars and workshops.

Overall, the time spent at CAS-UTS has been one of the most fulfilling of my life, especially in terms of robotic research, an area that interested me since childhood. However, I would have not reached this far if it wasn't for the support of several great persons back home in Sri Lanka. I am most grateful to Prof. Janaka Ekanayake, who helped me earlier on in my academic career and is a constant source of support and encouragement. I am exceedingly grateful to Dr. Devapriya Dewasurendra, whom I consider to be a great mentor and a scholar that I had the privilege of working with. It was Dr. Devapriya who put me in touch with Prof. Dissa. I am also grateful to Prof. Keerthi Walgama for the inspiring discussions we had, both on academic and philosophical topics over the year I spent in CE at University of Peradeniya. Unfortunately, I could not practice Thai Chi later in Australia, something that I regret. Also, I am thankful to Dr. Ranjith Dissanayake for his support and friendship.

Finally, my greatest thank goes to my family. For my mother, it is her softness and unbridled love that saw me through the most difficult of times. For my father, it is his unimaginable love, guidance and belief in me that prompted me to look beyond the obvious and to keep the creative spark burning. For my brother, for simply being my brother. Lastly, thanks to my wife, Nimali for tolerating my grumpiness at times of struggle and rejoicing at times of triumph and maintaining sanity and order in between. Without her, I would have not survived the last couple of years in my thesis and I now look forward to the life beyond the doctoral years with her and son Dinendra born on the day the thesis was submitted.

# Contents

**Appendix**

# List of Figures

# Abstract

This thesis deals with the problem of stereo vision based three dimensional Simultaneous Localisation and Mapping in the context of autonomous robotic navigation. Simultaneous Localisation and Mapping (SLAM) refers to the problem of mapping landmarks in an environment by the navigating robot and concurrently using the mapped features in the localisation of the robot. This thesis concentrates on the issues that arise from using a short baseline stereo vision system as the primary sensor for observing the environment.

Initially, a stereo vision sensor is empirically studied in the context of SLAM. Several error sources that could potentially affect the performance of SLAM algorithms are identified. It is then shown that the observation model corresponding to the particular vision system is highly nonlinear and as a consequence, traditional filtering techniques such as the Extended Kalman Filter used in solving the SLAM problem generate inconsistent state estimates. This observation leads to the development of a novel nonlinear batch optimisation technique that is shown to produce consistent state estimates.

The next major contribution of this thesis arises from the development of a novel Multi Map (MM) framework for SLAM. The framework was inspired by observations of human navigation habits. The novel representation relies on two

individual maps in the localisation and mapping process. The Global Map (GM) is a compact global representation of the robots environment and the Local Map (LM) is exclusively used for low-level navigation between local points in the robots navigation horizon. The LM in many aspects is similar to prevailing sub map methods and hence, has efficiencies of such representations. However, the combination of two map representations in the MM framework extends the capabilities of hitherto existent algorithms by not only in the way of improving consistency but also by way of increase in efficiency through the compact representation and the unique feature marginalisation strategy. In addition, it aids implementation of novel techniques for loop closure. The framework is highly suited for sensors like vision where map sizes tend to grow rapidly due to the very nature of the sensing techniques used.

Finally, the algorithms are validated with real experimental data collected using a mobile robot platform traversing in an indoor environment.

# Chapter 1

# Introduction

"Making a computer see was something that leading experts in the field of Artificial Intelligence thought to be at the level of difficulty of a summer student's project back in the sixties. Forty years later the task is still unsolved and seems formidable"

*-Olivier Faugeras [1]*

## 1.1 Simultaneous Localisation and Mapping

An automaton exploring the frontier relies on its ability of perception for a successful execution of the mission. Localisation and mapping plays a key role in this process. In the absence of *a priori* information of the environment, mapping and localisation requires simultaneous operation. Simultaneous Localisation and Mapping (SLAM) is thus, the process in which a map is being built of the environment in which the automaton is operating, while the map itself is being utilised for self localisation. Applications of such an algorithm is numerous and diverse in context and could span from robots in search and rescue, under water exploration, military reconnaissance to planetary exploration. Therefore, depending on the application domain, requirements of the SLAM algorithms vary. However, a fundamental requisite is that the algorithm

is robust and has the ability to produce consistent estimates. Secondly, the choice of sensors used for perception dictates the limits of practical viability of an application.

## 1.2 Background and Motivation

From a practical perspective, use of passive, inexpensive and compact sensors for robotic navigation is an appealing area of research. Vision sensors qualify exceedingly well with the above criteria. In addition, vision sensors produce large amounts of data that could potentially be used in further augmenting the navigation capabilities. They have the versatility to be implemented in widely varied application domains. Stereo vision is preferred over monocular vision due to the improved tractability gained through the sensors ability to perceive depth in addition to the bearing observations of a single camera. These desirable attributes motivated the research into the use of a compact stereo vision sensor in the context of SLAM.

### 1.2.1 Stereo Vision and SLAM

This thesis is concerned with issues pertaining to solving the SLAM problem using estimation theoretic techniques and use of stereo vision as the primary sensor for perception. In particular, it discusses the consistency issues in state estimation, arising from using a two-camera vision system with a short baseline.

Stereo vision has its own issues that need to be addressed prior to successful integration in a SLAM context. From a practical perspective, implementation of a stereo vision hardware set-up in a SLAM scenario requires a thorough understanding of the characteristics of the particular sensor. This motivates the study of the characteristics of a stereo vision sensor as presented in chapter 3.

### 1.2.2 Application Environment

An indoor environment with artificial lighting was chosen as the application environment. This has the advantage of having controlled lighting which improves the stability of the stereo sensor. However, there are other issues that need to be

addressed. Particularly, lack of texture, regular nature of the environment and restricted field of view pose considerable challenges to implementation of SLAM using stereo vision.

### 1.2.3 Stochastic Methods for SLAM

Application of Extended Kalman Filters in SLAM is a well researched area. During the past two decades considerable insight has been gained leading to development of robust applications. Specially, with the use of sensors such as laser range finders, millimetre wave radar and sonar, which are now considered to be traditional sensors in the context of SLAM. However, several researchers have reported issues in consistency of the EKF estimations due to nonlinear observation and process models.

As a solution, batch methods that utilise history of observations and/or estimates have gained considerable interest recently. Batch estimators have the advantage of utilising all available observations in estimating all the model parameters resulting in smoother estimates. However, the disadvantage is that, while EKF could naturally be implemented as a recursive algorithm resulting in online estimations, batch methods tend to be computationally expensive.

### 1.2.4 Multi-Map SLAM

This thesis exploits the advantages of both batch and recursive techniques in developing a unique multi-map representation for the SLAM problem. The technique has evolved through observations made on human navigation patterns. It has been observed that humans tend to use a few important visual cues when navigating from one point of interest to the next and tend to discard most of the information utilised in refining the navigation task in between such points along the path.

Consider the scenario depicted in Figure 1.1. A person is required to move from point A to the goal at point B. The person would utilise three important visual cues along the path for successful execution of the task. For instance, when the person

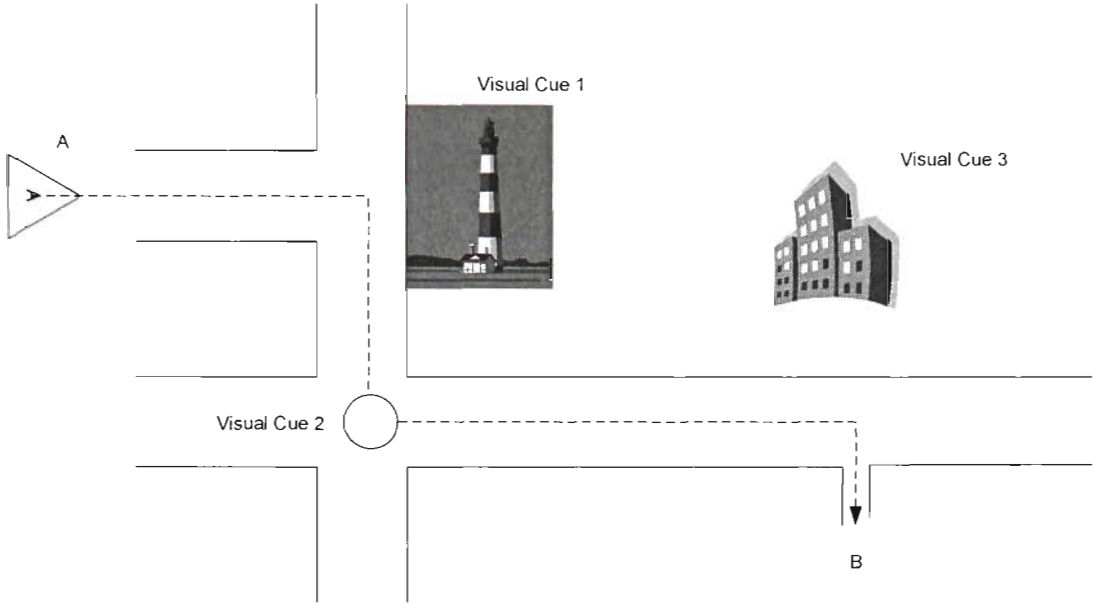



**Figure 1.1: Use of visual cues in a simple human navigation task.**

reaches visual cue 1, he/she would reorganise it and realise the need to turn right. Then the second visual cue, a traffic roundabout, prompts her to take the left turn. Finally, when he/she reaches the third visual cue, he/she reorganise that the goal is nearer. Generally, the person would not pay considerable attention to any other visual cues along the way, apart from using them to help stay on the correct path (e.g. the curb markings on the road, pedestrian crossings, traffic signs and traffic lights).

Similar principle is utilised in the development of the novel Multi-Map approach. The new representation consists of a Global Map, analogous to the three visual cues in the previous example. This map enables the SLAM algorithm to be bounded globally and the corresponding map size grows monotonically in dimensionality as the robots exploration horizon expands. The second map called the Local Map enables the lower level navigation between two visual cues of the global map, analogous to the lane markings, etc. When the navigation is completed, the local map corresponding to the path segment is removed from the robots state vector. Hence, the dimensionality of the local map state vector varies from segment to segment but

does not correlate with the growth of the robots exploration horizon, reducing the computational requirements of the overall SLAM algorithm considerably. This novel representation has the added advantage of realizing estimator consistency through explicit use of batch optimisation techniques in the estimation process.

## 1.3 Contributions

The main contributions of the thesis are as follows,

- Stereo vision sensor characteristics are analyzed through extensive simulations as well as experiments. Various issues pertaining to the unique combination of hardware and software are discussed and potential problems are identified. A signal conditioning algorithm is also proposed for improving the performance of the stereo vision sensor.
- This thesis also provides an insight into the behaviour of EKF based solutions to the SLAM problem using the stereo vision sensor. Particular attention is paid to the inconsistent filter performance due to the nonlinear nature of the observation model. Various potential techniques are discussed as alternative approaches to minimizing the inconsistency in state estimates of the standard EKF implementations. It is demonstrated that conventional recursive filtering mechanisms, known to perform well using traditional sensors, are incapable of producing consistent estimates in the particular vision sensor domain.
- A batch method inspired by the techniques used in Structure from Motion estimation in the computer vision research is developed as an alternative to the recursive estimation. A novel formulation integrates odometric information to the batch estimation. It is shown through empirical evidence that the new batch estimation algorithm produces consistent state estimates.
- A novel Multi-Map framework for SLAM is developed, and is then generalised as a technique for improving efficiency and consistency of the SLAM process. By allocating individual maps to the two fundamental tasks

in the SLAM framework (i.e. observing new features and re-observing old features) considerable computational gains are achieved and the real-time performance capabilities of the recursive algorithm is preserved even in large exploration tasks. By integrating batch optimisation techniques at key points in the navigation task, consistency of the state estimates is also improved.

- The techniques developed in the thesis are demonstrated using real data collected in an indoor office like environment. Various aspects of the implementation are discussed with particular attention to the practicality of the techniques developed.

## 1.4 Publications

Following publications were made during the course of the research presented in this thesis,

D. C. Herath, S. Kodagoda, and G. Dissanayake, "New Framework for Simultaneous Localization and Mapping Multi Map SLAM" in *IEEE/RSJ International Conference on Robotics and Automation (ICRA 2008)*, Pasadena, California, 2008, to appear.

D. C. Herath, K. R. S. Kodagoda, and G. Dissanayake, "Stereo Vision Based SLAM: Issues and Solutions," in *Vision Systems: Applications*, G. Obinata and A. Dutta, Eds.: Pro literatur verlag, 2007, pp. 565-582. (ISBN 978-3-902613-01-1)

D. C. Herath, S. Kodagoda, and G. Dissanayake, "Simultaneous Localisation and Mapping: A Stereo Vision Based Approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, Beijing, China, 2006, pp. 922-927.

D. C. Herath, K. R. S. Kodagoda, and G. Dissanayake, "Modelling Errors in Small Baseline Stereo for SLAM," in *The 9th International Conference on Control, Automation, Robotics and Vision (ICARCV 2006)*, Singapore, 2006.

S. Takezawa, T. Gulrez, D. C. Herath, and G. Dissanayke, "Environmental Recognition for Autonomous Robot using SLAM Real Time Path Planning with

Dynamical Localised Voronoi Division," *International Journal of Japan Society of Mechanical Engineering (JSME)*, vol. 3 C, 2005

D. C. Herath, S. Takezawa, and G. Dissanayake, "Planar Patch Feature Validation for Stereo Vision Based 3D Simultaneous Localization and Mapping," in *The Eleventh Annual Conference of the Institution of Electrical Engineers, Sri Lanka* Colombo 2004.

S. Takezawa, D. C. Herath, and G. Dissanayake, "SLAM in Indoor Environments with Stereo Vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Sendai International Center, Sendai, Japan, 2004

## 1.5 Thesis Outline

**Chapter 2** explores the general problem referred to as SLAM. Stochastic approaches such as, the Extended Kalman Filter (EKF), the nonlinear extension to the Kalman Filter and batch optimisation techniques are discussed in detail. It also discusses the concept of consistency with respect to filter estimates and techniques available for analysis of filter inconsistency are presented.

**Chapter 3** presents an empirical study of the stereo sensor to be used throughout the thesis. Discussion follows from the robotic navigation perspective. Using two representative robotic mapping experiments, various characteristics of the stereo vision sensor are identified and important parameters governing its performance are estimated. This includes the estimation of noise parameters and identification of trends they exhibit in correlation to depth. The analysis includes the development of a sensor model to be used in the SLAM algorithms and discuss the effects of high nonlinearity inherent in the model.

**Chapter 4** begins with a discussion on the nature of the short baseline stereo vision based SLAM with respect to consistency. It shows various formulations that yield different degrees of consistency and discusses the contributing factors. Then the

discussion proceeds to an alternative batch formulation that produces consistent results.

**Chapter 5** presents the Multi Map (MM) approach to the SLAM problem. This novel framework utilise two independent maps with uniquely assigned tasks. The Local Map (LM) consists of vision features that can be quickly tracked between consecutive image frames while the Global Map (GM) relies on descriptive features for loop closure. The vision sensor is inherently capable of generating a dense set of features. On the other hand, large number of features needs to be observed in each image frame for stable operation of SLAM algorithms. This reciprocative nature requires a sound map management scheme in order to maintain the tractability of algorithms. Thus, in the MM approach, LM features are marginalised at scheduled intervals and the modalities of the feature selection for the individual maps allow zero information loss in the process. Therefore, the framework is well suited in the sensor domain considered.

**Chapter 6** presents results from several experiments conducted using a mobile robot in an indoor environment illustrating various aspects of the algorithms developed in the previous chapters. The chapter also presents a novel algorithm that was successfully used during the experimentations for detecting spurious vision observations. A laser range finder based SLAM algorithm is used to generate 'ground truth' in order to compare the estimator results. Several practical aspects related to the sensor are also discussed.

**Chapter 7** concludes the thesis with a summary of the outcomes presented and an outlook to the future of related research.

# Chapter 2

# Simultaneous Localisation and Mapping

## 2.1 Introduction

It has been two decades since the seminal work by Smith, Self and Cheesman [2, 3], that spawned considerable research towards the problem of Simultaneous Localisation and Mapping (SLAM). Large body of work that followed relies on stochastic estimation techniques. Especially, with proven convergent properties [4], the Extended Kalman Filter (EKF) has been the algorithm of choice in many of these implementations [4-7].

Even though EKF has several appealing features, such as its recursive nature and simplicity in implementation, alternative approaches exist to the SLAM problem that addresses various drawbacks in EKF. These include, Monte Carlo methods [5-8], Gaussian Sum approaches [5], the Unscented Kalman Filter (UKF) [6-9], and batch estimation methods [10-13] that perform better with nonlinear system models, and information filter based approaches [14, 15] that scale better with large implementations. Monte Carlo, Gaussian Sum and batch methods are known to perform poorly when the state space expands. However, structure of the SLAM problem has been exploited successfully in batch methods to improve the

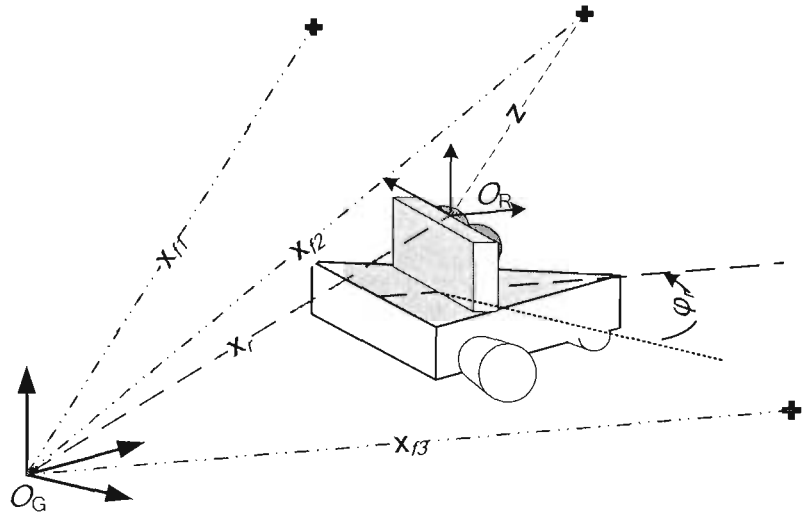



**Figure 2.1: The robot in X-Y plane observing features in 3D space. $O_G$ represents the global coordinate frame. $O_R$ is the robot centric coordinate frame where relative observations are made to the features.**

computational efforts to reach near real-time performance [10]. Sliding window based methods [16] and Loopy Belief Propagation techniques [17] are also gaining considerable interest in the development of fast batch solutions to the SLAM problem. This potential for real-time implementation combined with the advantages mentioned previously has motivated the use of batch methods amongst other alternatives to recursive techniques in this thesis. Thus, from an algorithmic perspective, this thesis is mainly concerned with the application of EKF in SLAM, its apparent inconsistencies in the state estimate due to the nature of the sensor model, and nonlinear batch optimisation techniques that could avoid the problem of inconsistency. A discussion on various localisation and mapping algorithms could be found in [18, 19].

Another prerequisite for an efficient SLAM algorithm is its ability to manage the map of the environment as the robot's navigation horizon expands with time. In [20], it was shown that a large number of features could be removed from the map without compromising the consistency of the estimates. This has been further developed in

[21]. Many other authors have presented sub map [22, 23] and hierarchical map [24, 25] representations as feasible solutions for large scale SLAM implementations. A novel Multi Map solution is proposed in this thesis that not only manages the complexity of the Map but also provides a unique practical solution for stereo vision based SLAM.

### 2.1.1 Three Dimensional SLAM

The specific SLAM problem to be studied consists of a mobile robot traversing on a planar surface equipped with a sensor capable of perceiving point features distributed in three dimensional space (Figure 2.1). A static environment is assumed. The robot state at time $k$ is defined by,

$$\mathbf{x}_r(k) = \begin{bmatrix} x_r(k) \\ y_r(k) \\ \phi_r(k) \end{bmatrix} \tag{2.1}$$

where $x_r$ and $y_r$ denotes location of the robot's rear axle centre with respect to the global coordinate frame $(O_G)$ and $\phi_r$ is the robot heading with respect to the $x$-axis of $O_G$.

The robot process model is given by,

$$\mathbf{x}_r(k) = \mathbf{f}(\mathbf{x}_r(k-1), \mathbf{u}(k)) + \mathbf{v}(k) \tag{2.2}$$

where $\mathbf{v}(k)$ is the zero mean additive Gaussian process noise with covariance $\mathbf{Q}(k)$. The control input $\mathbf{u}(k)$ is also assumed to have a Gaussian error distribution with zero mean and covariance $\mathbf{U}(k)$. The motion model for the robot considered in the simulations and experiments in the thesis could be approximated by,

$$\mathbf{x}_r(k) = \begin{bmatrix} x_r(k) \\ y_r(k) \\ \phi_r(k) \end{bmatrix} = \begin{bmatrix} x_r(k-1) + \Delta T V(k) \cos(\phi_r(k-1)) \\ y_r(k-1) + \Delta T V(k) \sin(\phi_r(k-1)) \\ \phi_r(k-1) + \Delta T \omega(k) \end{bmatrix} + \mathbf{v}(k) \tag{2.3}$$

where ΔT is the time step, $V(k)$ is the instantaneous velocity and $\omega(k)$ is the instantaneous turn-rate ( $\mathbf{u}(k) = [V(k), \omega(k)]^T$ ).

The position of the $i^{th}$ point feature in space is defined by the vector,

$$\mathbf{x}_{fi} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \tag{2.4}$$

and the map vector is defined by,

$$\mathbf{x}_m = \begin{bmatrix} \mathbf{x}_{f1} \\ \vdots \\ \mathbf{x}_{fn} \end{bmatrix} \tag{2.5}$$

The robot makes relative observations to the features in 3-D space,

$$\mathbf{z}(k) = \begin{bmatrix} z_x(k) \\ z_y(k) \\ z_z(k) \end{bmatrix} \tag{2.6}$$

where $z_x(k), z_y(k)$ and $z_z(k)$ are the Cartesian coordinates of the observed features with respect to the robot coordinate frame $O_R$. Let the observation model be,

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{w}(k) \tag{2.7}$$

with observation noise $\mathbf{w}(k)$ assumed to be Gaussian distributed, zero mean with covariance $\mathbf{R}(k)$ and $\mathbf{x}(k)$ being the augmented state vector containing both map and robot state vectors.

The observation model, as would be discussed in detail later has a significant impact on the performance of the SLAM algorithm. Therefore, at this point it suffices to note that the actual stereo observation model is highly nonlinear.

### 2.1.2 Data Association

Data association refers to the process of matching observations against the corresponding features that already exist in the map being built. Correct data association is of paramount importance to the consistent performance of any SLAM algorithm. When traditional sensors like laser range finders are used, statistical methods such as the nearest neighbour techniques [26, 27] have been used in order to establish correspondence between observations and features in the map. However, feature congestion and growing uncertainty in map estimates tend to make such methods susceptible to failure. Ambiguous associations are generally discarded resulting in a loss of valuable information. To alleviate these issues in data association, several robust methods have evolved recently including the joint compatibility test [28] , robust techniques in multiple frame tracking [29, 30], delayed association techniques [31] and multiple hypothesis techniques [32].

Computer vision, on the other hand offers information rich observations, which could be used to augment data association. For instance, the Scale Invariant Feature Transform (SIFT) [33] has been used in several SLAM implementations for data association [34, 35]. This thesis uses image registration techniques for data association based on the well established KLT feature tracker [36, 37] and a recently developed robust feature detector/descriptor method, Speed Up Robust Features (SURF) [38].

### 2.1.3 Loop Closure

Loop closure is the ability of a robot to recognise when it has returned to a previously visited location after an arbitrary traverse in the environment. This is an important aspect of a SLAM implementation where large information gains are made

by re-observing well established features in its state vector, which bounds the error estimates from growing incessantly. Loop closure is essentially a form of data association where the correspondence between features in the map and observations occur after considerable temporal lapse between the observations and the feature initialisations. Therefore, from a traditional sensor perspective, loop closure resorts to techniques used for general data association, such as nearest neighbour [26, 27] which lacks the robustness. Recent interest has grown in developing loop closure methods by augmenting traditional techniques with image registration techniques [39-41] and has seen implementations with considerable robustness.

This thesis explicitly uses image registration techniques for loop closure. However, it does not dictate a particular algorithm to be used. A suitable algorithm could be selected considering the application and the environment in which the algorithm is to be used. For instance, due to its practical utility, recently proposed SURF [38] features are used in this thesis for experimental analysis of the SLAM algorithms.

## 2.2 Estimation Theoretic Approach to SLAM

SLAM could be posed as a multivariate parameter estimation problem where a set of unknown state variables corresponding to the robot pose and map $(\mathbf{x})$ are estimated via the observations $(\mathbf{z})$ made of the environment through sensors on-board the robot. A general solution to the problem is to obtain the maximum a posteriori (*MAP*) estimate,

$$\begin{aligned} \mathbf{x}^{*} &\triangleq \underset{\mathbf{x}}{\operatorname{argmax}} \quad p(\mathbf{x} \mid \mathbf{z}) \\ &= \underset{\mathbf{x}}{\operatorname{argmax}} \quad p(\mathbf{z} \mid \mathbf{x}) p(\mathbf{x}) \end{aligned} \tag{2.8}$$

where $p(\mathbf{z} \mid \mathbf{x})$ is the observation likelihood and $p(\mathbf{x})$ is the prior.

The choice of estimator is problem dependent. Recursive and batch techniques form the core of the solution presented in the thesis. Generally, recursive methods are well

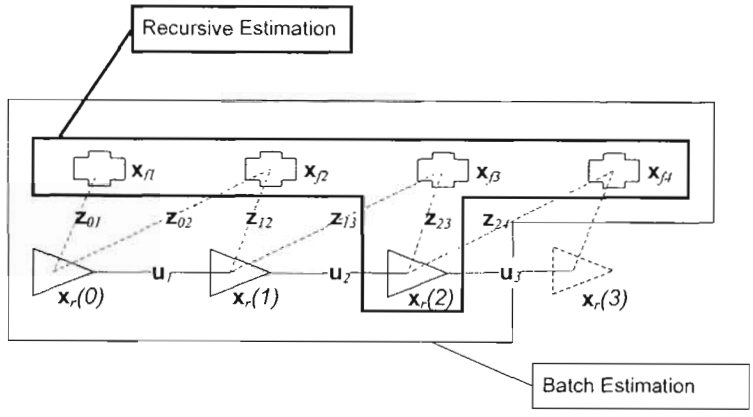



**Figure 2.2: Pictorial representation of the principal estimation techniques used in the thesis for solving the SLAM problem. Figure also illustrates the conditional independence between the observations and the states.**

suited for online estimation problems and batch methods are more suited for post analyses problems. Figure 2.2 depicts the two techniques from a SLAM perspective.

Assume that during the robots motion through the environment, from the initial pose at $\mathbf{x}_r(0)$, it has reached the current position $\mathbf{x}_r(2)$. Then, recursive estimators attempt to estimate the current robot pose $(\mathbf{x}_r(2))$ and the map $(\mathbf{x}_{f1},...,\mathbf{x}_{f4})$. It uses only the current observations $(\mathbf{z}_{23},\mathbf{z}_{24})$ and the odometry $(\mathbf{u}_2)$ in the estimation process. All the information gathered so far is summarised in the prior, which is the mean and covariance of the state vector in case of EKF. EKF is known to provide the best available recursive solution to (2.8). Batch techniques generally attempt to estimate the entire robot pose $(\mathbf{x}_r(0),...,\mathbf{x}_r(2))$ along with the map. This requires the entire observation history $(\mathbf{z}_{01},...,\mathbf{z}_{24})$ and odometry data $(\mathbf{u}_1,\mathbf{u}_2)$ to be available for the estimator.

Batch methods have been well studied for decades in computer vision and photogrammetry research. Particularly the problem of Structure from Motion (SFM)[12] in computer vision and bundle adjustment in photogrammetry extensively uses batch optimisation techniques. The problem refers in general to the scene and

the motion reconstruction of a monocular camera. The problem is more complex than SLAM, in that, depth information to features is not available and generally no prior information is assumed regarding the camera motion. This lack of constraints causes the SFM to be solvable only up to a scale factor. SLAM problem, on the other hand, could be constrained through additional information, such as the motion of the robot sensed through odometry and by the consideration of nonholonomic motion of most robotic platforms.

Lu and Milios [42] presented an early adaptation of batch least squares optimisation in a SLAM scenario. Deans [43] has demonstrated the applicability of batch methods in SLAM using bearing only sensors. Dellart *et al* [10], have developed factorisation techniques that are exact and have provided valuable insight into improving performance of batch algorithms through heuristic mechanisms. Frese, in a recent paper [44] discuss at length, both, recursive and batch estimation methods in the context of SLAM.

Figure 2.2 also reveals a unique characteristic of the SLAM problem in relation to the observation dependencies with respect to the state space. For instance, the observation $\mathbf{z}_{23}$ made at feature #3 relies only on the current robot pose $\mathbf{x}_r(2)$ and the feature location $\mathbf{x}_{f3}$. Similar conditional independence could be seen with control input $\mathbf{u}_i$, which only relates the current robot pose $x_r(i)$ and the previous robot pose $x_r(i-1)$. In fact, Dellart *et al* [10] use this character of the SLAM problem exclusively in the development of fast batch solutions.

From the common starting point (2.8), it is now possible to look at both recursive and batch estimators in general and in specific terms in relation to the SLAM problem.

### 2.2.1 Recursive State Estimation: The Extended Kalman Filter

The EKF is inherently a recursive algorithm comprising of three stages. Namely, prediction, observation and update. In case of SLAM, the augmented state $\mathbf{x}(k)$ at

the time step $k$, consists of the state of the robot, denoted by $\mathbf{x}_r(k) = [x_r, y_r, \phi_r]^T$ and the state of the map, denoted by $\mathbf{x}_m$ such that,

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_r(k) \\ \mathbf{x}_m \end{bmatrix} \tag{2.9}$$

The EKF then proceeds to estimate the current state $\hat{\mathbf{x}}(k)$ and the corresponding state covariance matrix,

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{rm} \\ \mathbf{P}_{rm}^T & \mathbf{P}_{mm} \end{bmatrix} \tag{2.10}$$

where $\mathbf{P}_{rr}$ is the covariance corresponding to the robot pose estimate, $\mathbf{P}_{mm}$ is the covariance of the map estimate and $\mathbf{P}_{rm}$ is the correlation matrix.

In the prediction stage, an estimate of the robot pose is generated through the robot process model (2.2),

$$\hat{\mathbf{x}}_r(k \mid k-1) = \mathbf{f}(\mathbf{x}_r(k-1), \mathbf{u}(k)) \tag{2.11}$$

Assuming a static environment, the predicted state of the map is given by,

$$\hat{\mathbf{x}}_m(k \mid k-1) = \hat{\mathbf{x}}_m(k-1) \tag{2.12}$$

The state covariance matrix is estimated as,

$$\hat{\mathbf{P}}(k \mid k-1) = \mathbf{F}_{\mathbf{x}}(k)\, \hat{\mathbf{P}}(k-1 \mid k-1)\mathbf{F}_{\mathbf{x}}^T(k) + \mathbf{Q}(k) \tag{2.13}$$

where, $\mathbf{F}_{\mathbf{x}}(k)$ is the Jacobian of $\mathbf{f}(\cdot)$ with respect to the state evaluated at $\mathbf{x}(k-1 \mid k-1)$.

During the second phase of the algorithm, observations $\mathbf{z}(k)$ are made to map features in the environment. These observations are then used in the recursive algorithm to update the state estimate. Thus,

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{K}(k)[\mathbf{z}(k) - \mathbf{h}(k \mid k-1)] \tag{2.14}$$

$$\hat{\mathbf{P}}(k \mid k) = \hat{\mathbf{P}}(k \mid k-1) - \mathbf{K}(k)\mathbf{S}(k)\mathbf{K}^{T}(k) \tag{2.15}$$

the Kalman gain is given by,

$$\mathbf{K}(k) = \hat{\mathbf{P}}(k \mid k-1)\mathbf{H}_{x}^{T}(k)\mathbf{S}^{-1}(k) \tag{2.16}$$

where,

$$\mathbf{S}(k) = \mathbf{H}_{\mathbf{x}}(k)\hat{\mathbf{P}}(k \mid k-1)\mathbf{H}_{\mathbf{x}}^{T}(k) + \mathbf{R}(k) \tag{2.17}$$

is the innovation covariance matrix and $\mathbf{H}_{\mathbf{x}}(k)$ is the Jacobian of $\mathbf{h}(\bullet)$ with respect to $\mathbf{x}$ evaluated at $\hat{\mathbf{x}}(k \mid k-1)$.

### 2.2.2 Nonlinear Batch Optimisation

Let the state variable $\mathbf{x}$ contain all the robot posses $\mathbf{x}_{r}(j), j = 1,\ldots,k$ and the map features $\mathbf{x}_{m}$. By taking the negative log of (2.8),

$$\mathbf{x}^{*} = \underset{\mathbf{x}}{\operatorname{argmin}} \quad -\log p(\mathbf{z} \mid \mathbf{x})p(\mathbf{x}) \tag{2.18}$$

The conditional independence in observations allows the factorization of the observation likelihood $p(\mathbf{z} \mid \mathbf{x})$ thus,

$$\mathbf{x}^{*} = \underset{\mathbf{x}}{\operatorname{argmin}} \quad -\log p(\mathbf{x})\prod_{i=1}^{z} p(\mathbf{z}_{i} \mid \mathbf{x}) \tag{2.19}$$

where $\mathbf{z}_{i}, i = 1,\ldots,z$ are the observations. Assuming a Gaussian observation model, the observation likelihood is,

$$p(\mathbf{z}_{i} \mid \mathbf{x}) = ke^{-\frac{1}{2}(\mathbf{z}_{i}-\mathbf{h}_{i}(\mathbf{x}))^{T}\mathbf{R}_{i}^{-1}(\mathbf{z}_{i}-\mathbf{h}_{i}(\mathbf{x}))} \tag{2.20}$$

where $\mathbf{h}_i(\mathbf{x})$ is the observation function and $\mathbf{R}_i$ is the corresponding observation covariance. The optimisation process requires an initial starting point $\mathbf{x}_0$ which is closer to the true value for meaningful convergence. Assuming a uniform prior over the states, the term $p(\mathbf{x})$ is removed. Let $\boldsymbol{\varepsilon}_i = (\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}))$, then the cost function to minimise is,

$$\mathbf{F}(\mathbf{x}) = \sum_{i=1}^{Z} \boldsymbol{\varepsilon}_i^T \mathbf{R}_i^{-1} \boldsymbol{\varepsilon}_i \tag{2.21}$$

Then the MAP estimate becomes,

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{Z} \varepsilon_i^T \mathbf{R}_i^{-1} \varepsilon_i \tag{2.22}$$

Equation (2.22) represents the nonlinear least-square problem to be solved. The term $\boldsymbol{\varepsilon}_i$ is the observation innovation, and $\mathbf{R}_i$ weights the contribution made by each observation to the estimation process. In general, the problem is solved through an iterative sequence,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k \tag{2.23}$$

where $\mathbf{x}_k$ represents the current estimate of the minimum at $k^{th}$ iteration. For a stable iterative process, $\Delta\mathbf{x}_k$ requires satisfying the condition,

$$\mathbf{F}_{k+1}(\mathbf{x}) < \mathbf{F}_k(\mathbf{x}) \tag{2.24}$$

When the observation function has continuous first and second derivatives, there exist gradient methods that efficiently compute $\Delta\mathbf{x}$ that satisfy the above condition including the Newton's method [45] which solves the problem in a single iteration. However, when the exact gradients are unknown, i.e. an exact linearization point could not be estimated, Gauss-Newton iterations could be used. In such quasi-Newton methods, the starting point in solving the nonlinear least squares problem of

(2.22) is the Newton's method, where the basic iteration is given by the following equation relating the first and second derivatives of the objective function,

$$(\mathbf{J}_k^T\mathbf{R}^{-1}\mathbf{J}_k + \mathbf{S}_k)\Delta\mathbf{x}_k = -\mathbf{J}_k^T\mathbf{R}^{-1}\boldsymbol{\varepsilon}_k \tag{2.25}$$

where $\mathbf{J}_k$ is the first derivative term (Jacobian) of $\mathbf{h}(\mathbf{x})$ and $\mathbf{S}_k$ represents the second derivative terms. The term $(\mathbf{J}_k^T\mathbf{R}^{-1}\mathbf{J}_k + \mathbf{S}_k)$ is the Hessian matrix. Since $\mathbf{x} \to \mathbf{x}^*, \mathbf{S}(\mathbf{x}) \to 0$, this could be further simplified by ignoring the term $\mathbf{S}_k$ in calculating the Hessian. Methods that ignore the effects of $\mathbf{S}_k$ are referred to as small residual algorithms.

Then the corresponding derivative terms could be assembled into (as in [10, 43]),

$$\begin{aligned}\mathbf{A} &= \sum \mathbf{J}_i^T\mathbf{R}_i^{-1}\mathbf{J}_i = \mathbf{J}^T\mathbf{R}^{-1}\mathbf{J} \\ \mathbf{b} &= \sum \mathbf{J}_i^T\mathbf{R}_i^{-1}\boldsymbol{\varepsilon}_i = \mathbf{J}^T\mathbf{R}^{-1}\boldsymbol{\varepsilon}\end{aligned} \tag{2.26}$$

The following can now be solved repeatedly for $\Delta\mathbf{x}$, until an acceptable lower bound is reached,

$$\mathbf{A}_k\Delta\mathbf{x}_k = -\mathbf{b}_k \tag{2.27}$$

This method relies on a 'good' initial guess $\hat{\mathbf{x}}_0$ for possible global convergence. In general, $\mathbf{J}$ is not rank deficient, however, limitations in numerical precision could cause $\mathbf{A}$ to become singular. A slight modification to (2.27) provides a means of dealing with this problem.

Consider,

$$(\mathbf{A}_k + \mu_k\mathbf{I})\Delta\mathbf{x}_k = -\mathbf{b}_k \tag{2.28}$$

Where $\mu_k \geq 0$ is a scalar and $\mathbf{I}$ is the unit matrix of the same order as $\mathbf{A}_k$. By having a sufficiently large value for $\mu_k$, the matrix $(\mathbf{A}_k + \mu_k\mathbf{I})$ becomes positive definite. In order to achieve similar asymptotic rate of convergence to that of the

Gauss-Newton method, it requires that as $\mathbf{x}_k \to \mathbf{x}^*$, $\mu_k \to 0$ . Levenberg-Marquardt method provides a strategy for selecting $\mu_k$ [46].

The Hessian $\mathbf{A}_k$ is the Fisher Information matrix [43]. Therefore, the covariance of the MAP estimate is given by the inverse of $\mathbf{A}_k$.

$$\mathbf{P}_k = \mathbf{A}_k^{-1} \tag{2.29}$$

A direct inversion of the Information matrix is costly. However, various factorisation methods exist that enable computation of $\mathbf{P}_k$ without inverting $\mathbf{A}_k$. Particularly, in the context of SLAM, the sparse nature of the matrix $\mathbf{A}$ could be exploited to gain considerable computational advantages [10]. Even though, this thesis is not concerned with improving the computational aspects of the batch method, chapter 4 briefly describes possible directions to be looked at in improving the performance of proposed solutions.

### 2.2.3 Cholesky Factorisation

The computational efficiency could be increased in the batch solution by realising the Cholesky Factorisation of the Hessian matrix $\mathbf{A}_k$. Since the Hessian is positive definite and is symmetric it can be factorised as $\mathbf{C}_k \mathbf{C}_k^T = \mathbf{A}_k$.

Then the following dual could be solved in $O((N+M)^2)$ instead of solving (4.39) directly. $N, M$ are the number of features and number of robot poses respectively.

$$\begin{aligned} \mathbf{C}_k \mathbf{a}_k &= \mathbf{b}_k \\ \mathbf{C}_k^T \Delta\mathbf{x}_k &= \mathbf{a}_k \end{aligned} \tag{2.30}$$

The Cholesky factorisation is computed in $O((N+M)^3)$ for a dense matrix. In the work described in this thesis the *chol* function available with Matlab was used with considerable improvement in computational time. Further improvements have been suggested in the literature in solving similar problems by considering the structure of

the SLAM problem and the sparse nature of the Hessian. These methods generally require a reordering of the Hessian [10].

## 2.3 Filter Consistency

Generally, the filter consistency refers to how well the filter estimates represent the true underlying distributions of the system. The Gaussian representation of the posterior in the previous algorithms leads to the SLAM state estimates being approximated by a mean and a variance.

Bailey [47] argues that this approximation could eventually contribute to accumulated error resulting in filter divergence. There is considerable evidence being put forth suggesting filter inconsistency due to the linearization of the system functions. In fact, Castellanos *et al* [48] indicate that issues of consistency due to linearization appear well before the computational problems become apparent due to the growing map. This suggests that issues related to consistency should be resolved prior to large scale deployments of SLAM. Julier [49] proves that the map always becomes inconsistent for a stationary robot, and through simulations shows similar results when the robot is moving. Huang [50] has recently shed valuable theoretical insight in to the effects of linearization has on the EKF with specific commentary on claims made by previous authors on filter consistency. In particular, this work substantiate the claim that, estimator inconsistency could be a fatal problem when the error in robot orientation is large [47].

As will be shown later, the observation model of the vision sensor used in this thesis has considerable influence towards filter inconsistency. Therefore, it is essential that the algorithms developed be tested systematically for filter inconsistency. Normalised Estimation Error Squared (NEES) [51] is a commonly used measure for consistency analysis when the true values $\mathbf{x}(k)$ of the states are known. NEES is defined by,

$$\varepsilon(k) = (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k))^T \mathbf{P}(k \mid k)^{-1} (\mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k)) \qquad (2.31)$$

Under the hypothesis $H_0$ that filter is consistent and the assumption that the process models are linear and the noise sources are distributed according to Gaussian models, $\varepsilon(k)$ is chi-square distributed with $n_x$ degrees of freedom. Where $n_x$ is the dimension of the state.

Then the test is whether

$$E[\varepsilon(k)] = n_x \tag{2.32}$$

could be accepted.

For a set of $N$ independent simulations, the average NEES could be estimated as,

$$\bar{\varepsilon}(k) = \frac{1}{N}\sum_{i=1}^{N}\varepsilon_i(k) \tag{2.33}$$

Then under $H_0$, $N\bar{\varepsilon}(k)$ will have a chi-square distribution with $Nn_x$ degrees of freedom. The hypothesis that the state errors are consistent with the filter estimated covariances is accepted if

$$\bar{\varepsilon}(k) \in [r_1, r_2] \tag{2.34}$$

where the acceptance interval is determined on a statistical basis. For example, in a simulation experiment consisting of 50 trials, for an estimate of the robot pose consisting of 3-dimensions, a value of $r_2 = 3.72$ represents the upper 95% confidence bound and $r_1 = 2.36$ is the lower bound. When the estimated value is higher than the upper bound, the filter is said to be optimistic, when it is lower than the lower bound the filter is behaving conservatively.

## 2.4 Simulations

Simulations form a valuable tool in analyses and validation of various hypotheses put forward in the thesis. Specially, when addressing the issues of consistency, the absolute knowledge of feature and robot locations is essential to assess the behaviour

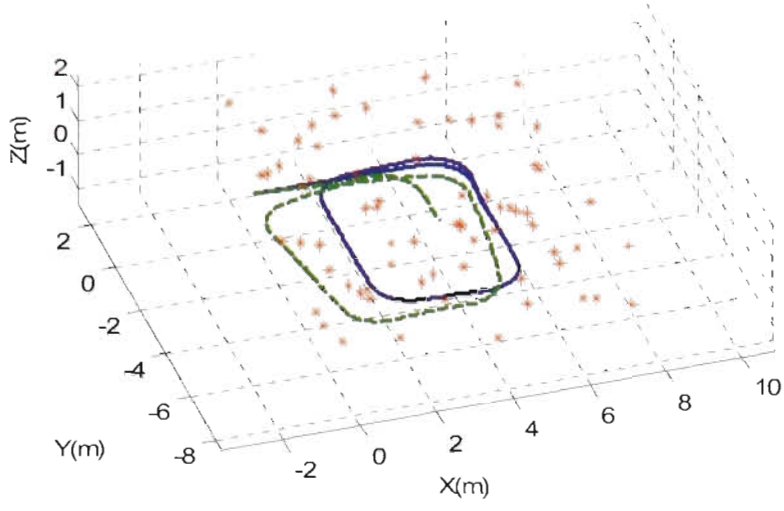



**Figure 2.3: Simulated environment for 3D SLAM experiments. Solid line indicates the true path of the robot, dashed line shows the path obtained using odometry and asterisks indicate features.**

of the filter. Simulated data therefore, allows for comparison between filter estimates and the true states of the system. Real experiments presented in Chapter 6 on the other hand rely on state estimates generated by an alternate SLAM algorithm that uses a high precision laser range finder as the primary sensor to establish 'ground truth'. However, the laser based SLAM algorithm only provides an estimate of the robot poses and the locations of vision features is difficult to establish. Therefore, computer simulations are used in this thesis to examine the consistency of proposed algorithms whereas experiments are used to demonstrate the robustness of these algorithms in dealing with practical issues such as data association and spurious observations.

### 2.4.1 Simulation Environment

The simulated environment and the robot path were chosen to emulate one of the real experiments presented in chapter 6 (Figure 2.3). Table 2.1 shows the noise parameters used in the simulations. They correspond to empirical values obtained through numerous experiments conducted using a real robot. The robot trajectory contains four sharp $90^o$ turns preceded by fairly straight segments of traverse. Features are randomly distributed in 3-D space. The sensor on-board the robot observes features constrained within a cone that emulates the field-of-view of the real sensor (Appendix A.1 and chapter 3) and has a maximum range of 10m. Data association between observations and corresponding features are assumed to be known explicitly. The next section illustrates a general example using the simulated environment with an observation model that assumes Cartesian observations corrupted by Gaussian noise.

**Table 2.1: Simulation filter parameters based on empirical observations**

| | | |
|---|---|---|
| Sampling period | $\Delta T$ | $0.25s$ |
| Robot control noise covariance matrix | **U** | $\begin{bmatrix} 0.05^2(m/s)^2 & 0 \\ 0 & 0.08^2(rad/s)^2 \end{bmatrix}$ |

### 2.4.2 An Example

Since stereo vision involves a highly nonlinear projection, the noise parameters and the mechanism used to simulate vision observations warrants special attention. This will be discussed further in the following chapters. In the example presented next, attention is focused on establishing the fact that the stochastic methods discussed earlier produce consistent estimates in the simulation environment of the previous section when a generic Gaussian observation model is used. Note that the observation model developed below is different from the actual observation model of a stereo camera.

Consider a hypothetical sensor that can sense features in 3D in Cartesian reference frame (as in Figure 2.1). Then the observation model is,

$$\mathbf{z}(k) = \begin{bmatrix} z_x(k) \\ z_y(k) \\ z_z(k) \end{bmatrix} = \begin{bmatrix} a \\ b \\ z_{fi}(k) \end{bmatrix} + \mathbf{w}(k) \tag{2.35}$$

where

$$a = (x_{fi}(k) - x_r(k))\cos(\phi_r(k)) + (y_{fi}(k) - y_r(k))\sin(\phi_r(k))$$
$$b = -(x_{fi}(k) - x_r(k))\sin(\phi_r(k)) + (y_{fi}(k) - y_r(k))\cos(\phi_r(k))$$

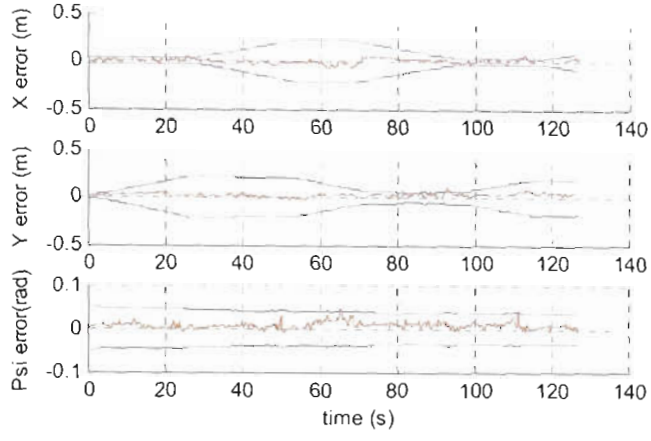
Figure 2.4a shows the error in the estimate of robot pose along with the $\pm 2\sigma$ error bounds for a SLAM algorithm that uses the above observation model in an EKF. In this simulation, loop closure occurs at approximately $95sec$ into the execution, reducing both error bounds corresponding to the robot position. Interestingly, the information gain is gradual as indicated by the gradual decrease in the covariances and the characteristic jump in robot uncertainty at loop closure is not visible in the results. Figure 2.4b shows the average NEES for 50 trials indicating that the estimator is consistent.

Figure 2.5 shows comparative results from the nonlinear batch optimisation algorithm. The implementation uses the hypothetical observation model presented earlier and does not assume any knowledge about the robot motion. A technique is developed later in the chapter for constraining the optimisation through odometry information and nonholonomic motion of the robot.

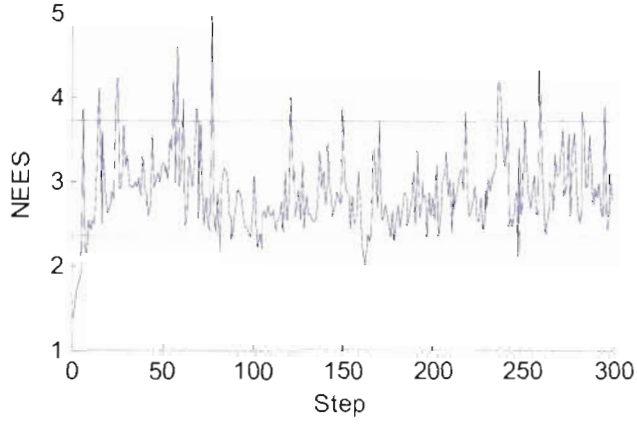
## 2.5 Discussion

This chapter provided background information and reviewed some of the estimation theoretic methods available for solving the Simultaneous Localisation and Mapping problem. Particular attention was paid to the Extended Kalman Filter and nonlinear batch optimisation algorithms. Notion of filter consistency and a tool for measuring it

was also discussed. Primary motivation for the work presented in the thesis stems from the inconsistencies noted during simulations that use stereo observation models. This has prompted an exhaustive empirical study of a stereo sensor (discussed in Chapter 3) and the consequent estimation techniques in chapter 4. Chapter 5 presents a novel approach to the SLAM problem using a Multi-Map framework. Multi-Map representation helps to manage the complexity of SLAM solution while maintaining consistency of the algorithm. Interestingly, the solution emulates general tendencies of navigating humans.

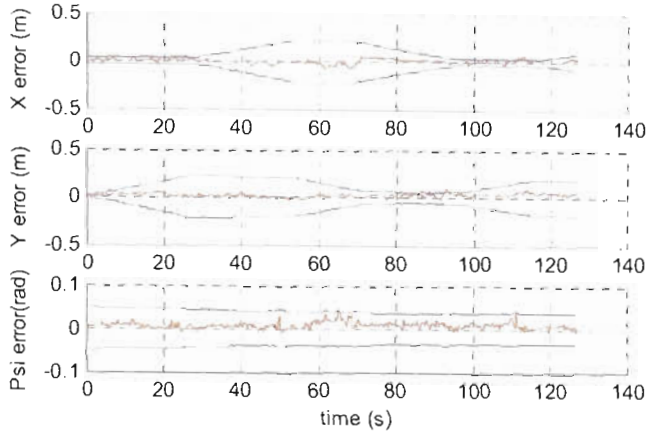



(a)

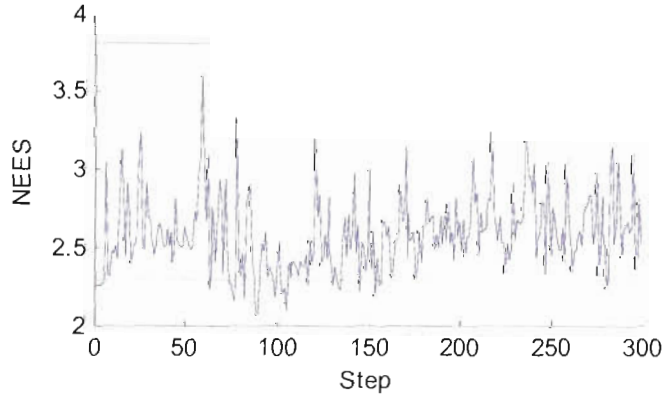



(b)

**Figure 2.4: Estimator results using EKF and a Cartesian observation model.** **(a) Robot pose errors with estimated 2-sigma bounds. (b) Average NEES for 50 trials with the 95% acceptance interval**




(a)




(b)

**Figure 2.5: Estimator results using batch optimisation and a Cartesian observation model.** **(a) Robot pose errors with estimated 2-sigma bounds. (b) Average NEES for 50 trials with the 95% acceptance interval**

# Chapter 3

# The Vision System

## 3.1 Introduction

Generally, more precise the measurements from the sensors used in SLAM, more tractable and practical the solution is. Thus the underlying characteristics of the sensor play an important role in determining the scale and the practical use of the SLAM algorithm. Sensors such as laser range finders have proven to be very precise in nature and have shown to work well in large environments for extended periods of time [52-54]. However vision based sensing is yet to prove its application in similar environments. In vision, successful implementations of SLAM to date have used either large baseline stereo cameras [55, 56] or camera configurations with more than two cameras providing refined observations [34]. And on the other end of the spectrum, single camera bearing only [57-59] methods are vigorously studied. In this, research is mostly concentrated in developing novel initialisation algorithms [11, 60-62] that can handle the absence of range information in the observations.

A concerted study on the use of short baseline ($< 0.1m$) stereo cameras has largely been overlooked by the robotics navigation research community. The motivation for pursuing such a sensor in the application of SLAM is twofold. Firstly, a camera with

a small foot print is ideal for compact robotic systems that require higher degree of agility and speed. An example application of such requisites is the 'RoboCup rescue' [63]. Secondly, additional information provided by a second camera improves the tractability of SLAM solutions compared to bearing-only SLAM with a single camera. In this regard, low cost short baseline stereo vision is an appealing alternative to conventional sensors providing ideal balance between size, performance and cost.

Principal aim of this chapter is to assess the performance of a short baseline binocular stereo camera equipped with wide angle lenses in the context of SLAM. A complete view of the sensor encompassing not only the characteristics of stereo vision, but also of the image processing techniques that extracts representative information of the environment are examined in detail.

## 3.2 Stereo Vision in SLAM

Stereopsis or stereoscopic vision is the process of perceiving depth or distances to objects in the environment. As a strand of computer vision research stereo vision algorithms have advanced noticeably in the past few decades to a point where products are available as 'off the shelf' devices. Since the thesis concentrates more on development of consistent and fast SLAM algorithms based on stereo vision, it is not in the scope of this thesis to develop custom stereo rigs or stereo algorithms. Hence, a commercially available stereo rig and stereo algorithms are employed in the experiments discussed in this thesis.

A higher level component list of the vision sensor can be summarised as follows along with interactions amongst the components outlined in Figure 3.1.

- Camera pair for acquiring images of the scene.
- Calibration information – that provides the intrinsic and extrinsic parameters of the camera pair.

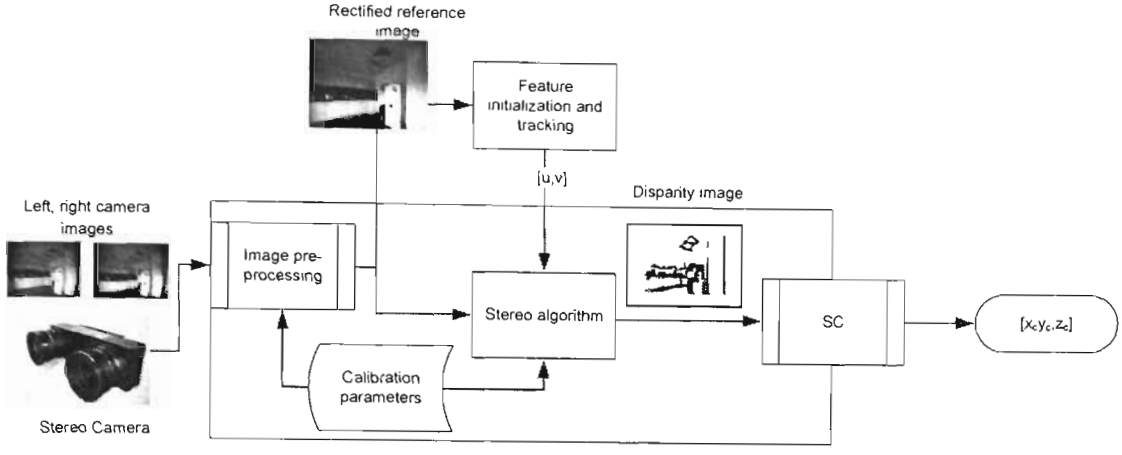



**Figure 3.1: The vision system for a SLAM implementation. Shaded area indicates the image processing section of the vision system that has been adapted from a commercially available stereo vision package. Further details are presented in Appendix A.**

- Feature selection algorithm – for extracting distinctive points of interest in the scene.
- Feature tracking algorithm – for associating respective features between consecutive images
- Stereo algorithm – for generating dense depth maps of the scene.
- Signal Conditioning (SC) – for filtering noisy input signals.

Algorithmic requirements for depth recovery are represented by the shaded area in Figure 3.1, whilst, feature initialisation and tracking component represents the interest operator that extract information necessary for representing the environment. As in many other signal processing systems smoothing and signal conditioning algorithms are also required. The experiments described in this thesis uses the *Videre Design* camera model *STH-MDCS/C* and the associated stereo algorithm, the Small Vision System (SVS) [64]. A short review of the stereo camera rig and the stereo correspondence algorithm are presented in Appendix A. Even though the algorithm incorporated into SVS provides various techniques to reduce spurious data,

it is observed during implementations that further refinements are required. Several additional techniques developed for this purpose are discussed in the next section.

An interest operator based on the image gradients [36] forms the feature selection algorithm. A tracking algorithm that is optimal by design [37, 65-67] provides fast data association of features between successive images. These algorithms are collectively known as the Kanade-Lucas-Tomasi (KLT) algorithm. The reasoning behind this selection is primarily the speed and the control over feature selection space. Especially KLT provides better ranking of features, so a well informed decision could be made on discriminative feature selection. Secondly, the algorithm provides fast feature tracking between consecutive images. Compared to alternate methods that require large databases of descriptors (such as SIFT [33, 68, 69], SURF [38] and several other methods [70]) to be compared against each other for tracking, KLT requires only a minimalist search window to be correlated.

### 3.2.1 Camera Model

The standard model for a calibrated parallel camera pair (Figure 3.2) yields following triangulation equations,

$$\begin{aligned} z_c &= Bf / d_i \\ x_c &= u_i z_c / f \\ y_c &= v_i z_c / f \end{aligned} \tag{3.1}$$

where image coordinates $(u_i, v_i)$, disparity $(d_i)$ are considered to be the principal observations, $(u = 0\ , v = 0)$ is the image centre in the image plane, $B$ *(baseline)* is the offset between the two cameras in the stereo rig, $(f)$ is the focal length and the observation vector in camera centric coordinates is $\mathbf{z}_c = [x_c, y_c, z_c]^T$. The global coordinate frame $(O_G)$ is selected to be at the initial robot pose. (i.e. $O_G \equiv O_R$ at time $k = 0$). A coordinate transform is necessary to map the camera-centric

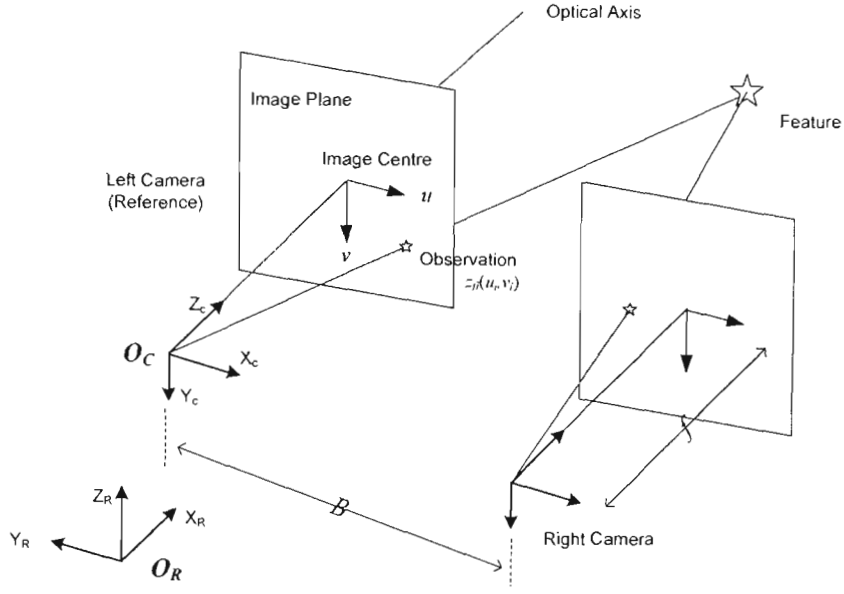



**Figure 3.2: Stereo camera model. A coordinate transformation maps features between camera-centric ($O_C$) and robot-centric ($O_R$) frames.**

coordinate frame ($O_C$) to the robot-centric coordinate frame ($O_R$). All the derivations henceforth are with respect to the robot centric coordinate frame unless specified and $O_C \equiv O_R$ is assumed for clarity, although in practice the camera coordinate frame may be offset with respect to robot coordinate frame by $(X_{offset}, Y_{offset}, Z_{offset})$. The location of the $i^{th}$ feature with respect to $O_R$ is defined as $[x_{fi}, y_{fi}, z_{fi}]$.

Origins of the camera, robot and global coordinate frames are assumed to lie on the same horizontal plane. Therefore, the triangulation equations with respect to the robots frame of reference ($O_R$) is defined by,

$$\mathbf{z}_R = \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} = \frac{B}{d} \begin{bmatrix} f \\ -u \\ -v \end{bmatrix} \tag{3.2}$$

In order to utilise (3.2) in a SLAM scenario, it is necessary to derive an error model for the triangulation. Matthies [71] describes a Gaussian error model for the projective observation and Murray [72] recently extended its interpretation to account for the differences in pixel accuracy in left and right cameras . As in [71] assuming $\mathbf{z} = [u, v, d]^T$ to be Gaussian distributed random vector, the corresponding covariance matrix describing the errors of the measurements $u$,$v$ and $d$ could be defined as,

$$\mathbf{R}_{uvd} = \begin{bmatrix} \sigma_d^2 & 0 & 0 \\ 0 & \sigma_u^2 & 0 \\ 0 & 0 & \sigma_v^2 \end{bmatrix} \tag{3.3}$$

where $\sigma_u, \sigma_v, \sigma_d$ are the standard deviations of the Gaussian error distributions of $u$,$v$ and $d$ respectively. The Jacobian of the projected observation $(\mathbf{z}_c)$ with respect to $\mathbf{z}$ is,

$$\mathbf{J} = \begin{bmatrix} \frac{-fB}{d^2} & 0 & 0 \\ \frac{Bu}{d^2} & \frac{-B}{d} & 0 \\ \frac{Bv}{d^2} & 0 & \frac{-B}{d} \end{bmatrix} \tag{3.4}$$

Then the propagated error covariance matrix of the feature location in camera centric coordinates corresponding to the projected observation $\mathbf{z}_c$ is,

$$\mathbf{R}_{xyz} = \mathbf{J}\,\mathbf{R}_{uvd}\,\mathbf{J}^T = \frac{B^2}{d^2}\begin{bmatrix} \frac{f^2\sigma_d^2}{d^2} & \frac{-uf\sigma_d^2}{d^2} & \frac{-vf\sigma_d^2}{d^2} \\ \frac{-uf\sigma_d^2}{d^2} & \sigma_u^2 + \frac{u^2\sigma_d^2}{d^2} & \frac{uv\sigma_d^2}{d^2} \\ \frac{-vf\sigma_d^2}{d^2} & \frac{uv\sigma_d^2}{d^2} & \sigma_v^2 + \frac{v^2\sigma_d^2}{d^2} \end{bmatrix} \tag{3.5}$$

It has been demonstrated that [73, 74], the linear approximation made to derive (3.5) results in introducing a bias to the estimated feature location $[x_{fi}, y_{fi}, z_{fi}]$. It has also been shown that the effect of this bias is considerable when observing features at larger depths ($\approx 1 pixel$). Conversely the bias manifests at relatively smaller depths when using cameras with shorter baseline. This effect is discussed from the sensor perspective in section 3.4 and its effects in the context of SLAM are discussed in detail in chapter 4.

### 3.2.2 Interest Operator

Computer vision literature abounds with different interest operators that are suited for a multitude of tasks [36, 75-77]. The requirement in the case of SLAM is extracting physically stable features. Shi and Tomasi [36] describes an interest operator based on the intensity gradient.

$$\nabla = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} \tag{3.6}$$

Features are extracted by analysing the nature of the gradient matrix described by (3.6) in a fixed size window. For a good feature, it is argued in [36] that this matrix must be well conditioned and above noise level. This is analysed by the two eigenvalues of the matrix $\nabla$ where the ideal scenario being two large eigenvalues representing a highly textured area. A single high eigenvalue represents a unidirectional texture pattern. Thus a selection criterion can be written as,

$$\min(\lambda_1, \lambda_2) \geq \lambda_{threshold} \tag{3.7}$$

Where $\lambda_{threshold}$ is selected empirically to yield stable textured features in the environment.

**Figure 3.3: An example of a KLT tracking sequence (top left shows the set of initialised features). Notice the virtual border (dashed rectangle) within which the features are initialised and tracked. A border is required to maintain the correlation window within the image frame. A negative effect of this can be seen in the bottom right image where three of the obvious features being lost due to the border.**

### 3.2.3 Feature tracker

A robust tracking algorithm essentially can solve the data association problem present in SLAM related implementations. There is a rich literature comprising of a variety of tracking algorithms and variants. However, a compromise between performance and computational overhead needs to be established. The algorithm described in a series of papers by Kanade, Lucas and Tomasi [37, 65, 67] (KLT algorithm) provides an efficient tracker which can be implemented in real-time, is used in the thesis. Tracking is achieved by minimising the dissimilarity between the two windows in consecutive images. Brief introduction to the tracking algorithm is given in Appendix A.4.

Figure 3.3 shows an example image sequence where the first frame of the sequence initialises a set of features that are then tracked in the consecutive frames by the KLT algorithm.

### 3.2.4 Additional Signal Conditioning

Stereo algorithms rely on image correlation techniques for estimating the pixel disparity between images from the camera pair, and such algorithms are susceptible

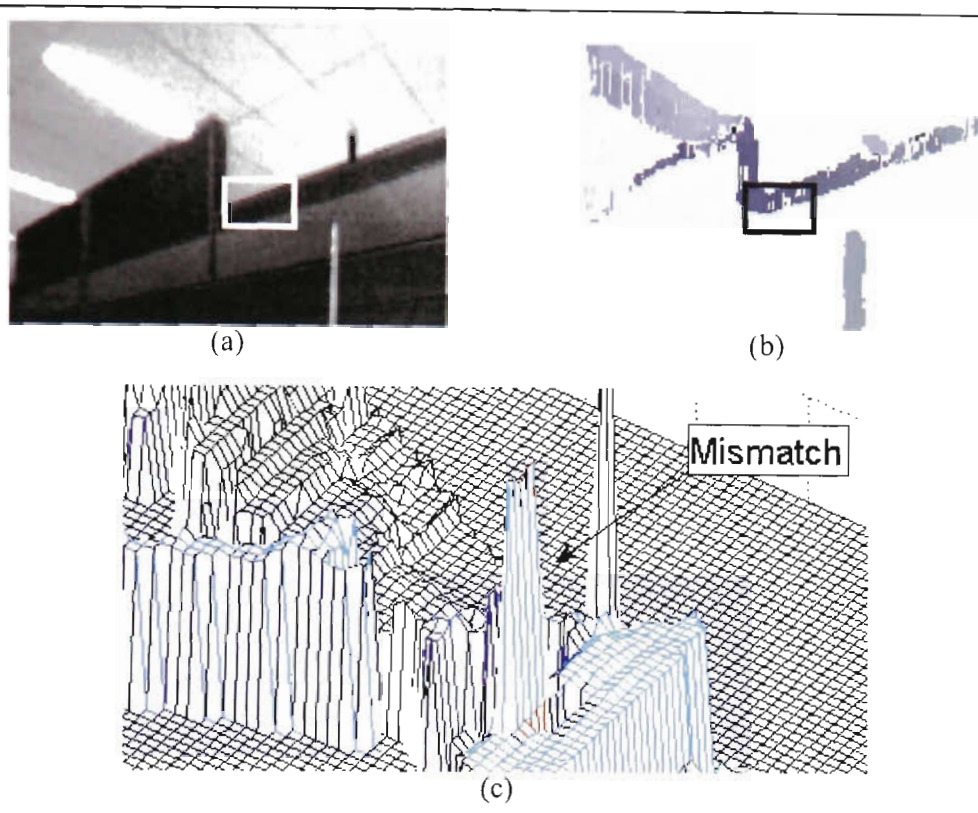



**Figure 3.4: Various aspects of stereo processing (a) A rectified image showing several edge and planar profiles. (b) Corresponding disparity image (c) Close-up view of the depth profile of the indicated area. The profile corresponds to a planar section of the partition board. The two spikes indicate depth discontinuities that are not present in reality and they should be treated as spurious data and should be removed from the signal.**

to occasional mismatches. Filtering schemes implemented by the correspondence algorithm used within SVS (Appendix A.3) removes most of the errors in a depth map at the cost of increased computation and loss of information. However, even after implementation of such signal conditioning mechanisms occasional errors do exist (see Figure 3.4c), and has the potential to destabilise EKF implementations.

The lack of texture in the environment and occlusions contribute to such mismatches during stereo processing. It is possible to device a mechanism to filter such spurious observations in this particular environment by noticing that most of the structures are planar surfaces. Hence, it is possible to hypothesise that any spurious features would be represented by large discontinuities in the disparity continuum.

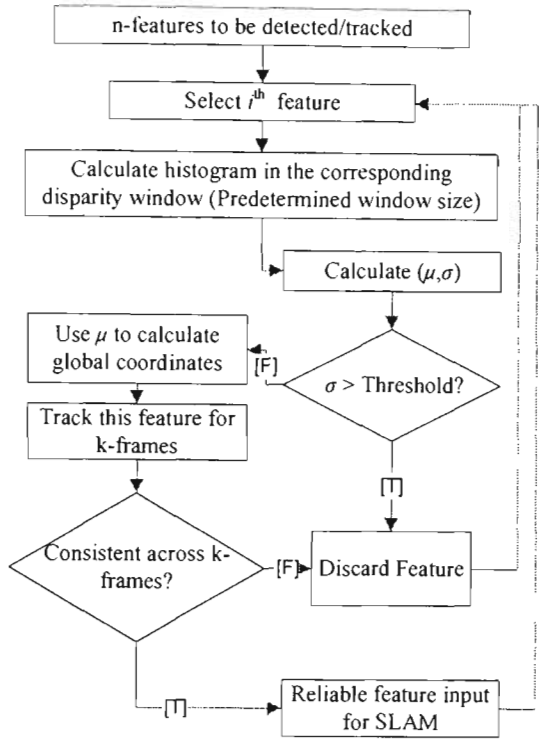



**Figure 3.5: Feature detection mechanism in a general SLAM implementation along with spurious depth filtering.**

This could be further illustrated by observing that the two spikes in Figure 3.4c are sudden discontinuities in an otherwise relatively smooth disparity profile that corresponds to the planar surface of the office partition in Figure 3.4a.

Therefore, a filtering algorithm could be devised on the assumption that for a given feature, the neighbourhood is planar. The algorithm considers a small patch of the disparity image around the corresponding feature location and calculates the depth histogram for this small patch and determines the mean and the standard deviation. The resulting standard deviation ($\sigma$) is compared against a predetermined threshold ($\sigma_{threshold}$) to eliminate features with spurious disparity values. A general observation in the application of this algorithm is that the calculated mean ($\mu$) better represents the disparity at the selected point. A 3x3 patch, which is the smallest possible support region, was found to be adequate in the applications of the filter. Figure 3.5 presents

a schematic of the overall application of the Signal Conditioning algorithm during the feature initialisation phase of a SLAM algorithm. In this, additional measures are taken to establish reliable features through delayed initialisation of the features. A delay of $k$=2 frames was found to be a reasonable compromise.

## 3.3 An Empirical Study of Sensor Characteristics

Characteristics of a sensor dictate the limits of its application. A rich literature is readily available on quantitative error analyses of stereo vision, based on static cameras [71, 78]. However, it was decided to study the resulting sensor characteristics of the particular combination of hardware and algorithms related to the vision system in a robotics navigation scenario, which is potentially different from the characteristics of a static camera. In the following sections, a discussion of an empirical study of the vision system (section 3.1) is given based on two representative experiments. This study concentrates specifically on the characterisation of noise performance of a short baseline camera in respect to SLAM.

The triplet $\mathbf{z} = [u, v, d]^T$ forms the principal observation $\mathbf{z}$ of the sensor. Assuming that the errors in the observations to be additive, $\mathbf{z}$ can be written as,

$$\mathbf{z} = \mathbf{z}_{true} + \mathbf{v}(\zeta, \mathbf{z}_{true})$$

Where $\mathbf{z}_{true}$ is the true observation and $\mathbf{v}$ is the additive noise component dependent on the sensor characteristics $\xi$ and on the true state itself as will be shown empirically later. Modelling and understanding the behaviour of $\mathbf{v}$ is the subject of discussion in following sections. The empirical analysis is based on the two mapping experiments discussed in the following section.

### 3.3.1 Mapping Experiment

For the experiment discussed below and in the remainder of the thesis, an ActiveMedia Robotics' PioneerDX was used as the mobile robotic platform (Figure

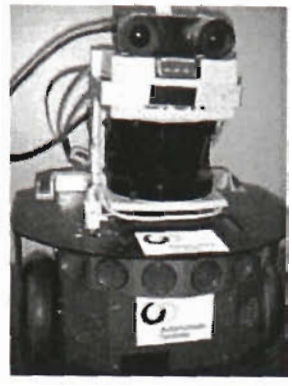


**Figure 3.6: Pioneer mobile robot platform with sensor payload**

3.6). The platform contains an odometry sensor, a frontal array of sonar sensors and a top mounted SICK LMS laser range finder. The stereo camera was mounted on top of the laser as depicted in Figure 3.6. All the experiments were conducted at the robotics laboratory at University of Technology Sydney.

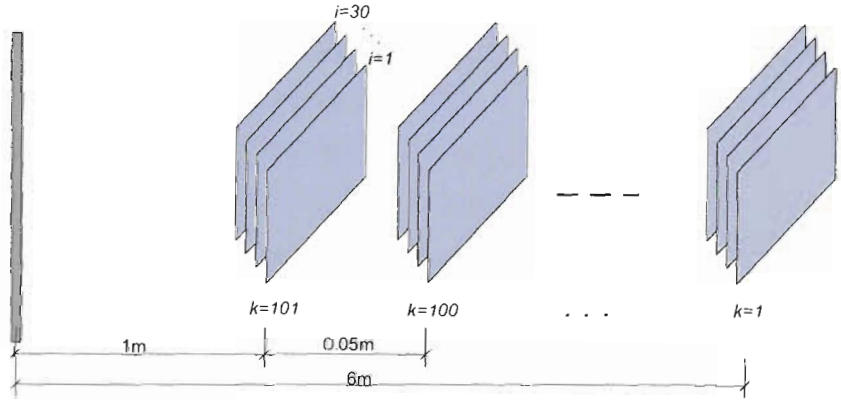
The robot platform, mounted with the stereo camera was moved on a controlled path, while observing a set of artificial features laid on a large vertical planar surface (see Figure 3.7). The artificial features were designed in such a way that the KLT feature detector ranks them higher in the order. Therefore, they act as stable and physically identifiable features with known locations. The 2D locations of these features with respect to each other were measured manually during the experimentation.

A SICK laser range finder was used to maintain parallel alignment between the camera and the surface, and to measure the nominal distance between the robot and the surface. Robot was moved in 0.05m increments from a depth of 6m to 1m. At each stop point 30 images were collected in a total of 101 stop points. A total of 3030 images were collected for the analyses described later.

### 3.3.2 Uncertainty in Disparity

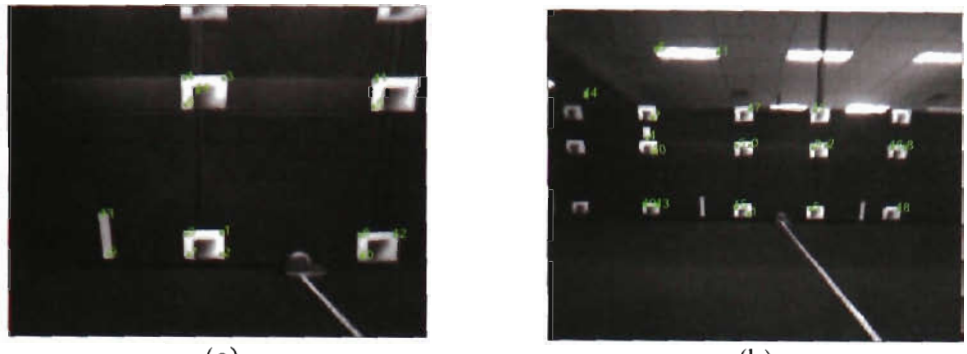
Data from above experiment could be used in validating the general assumption of disparity observations being corrupted by Gaussian process noise [55, 71, 74]. In

(a)




(b)

**Figure 3.7: Mapping experiment for sensor characterisation. (a) Environment prepared for the experiment. White sheets marked with a black shade pattern were laid in a regular grid on the vertical planar surface. (b) The robot was moved in a stop-go manner from a depth of 6m from the planar surface. Experiment was terminated when the distance between the robot and the surface was 1m ($k$=1,...,101). At each stop point 30 images were acquired ($i$=1,...,30).**



(a) (b)

**Figure 3.8: Rectified images overlaid with KLT features at (a) 1.0 m (b) 2.8 m**

fact, the data could be used to assess the behaviour of disparity between a range of depths. This additional information provides a unique insight into the distribution of disparity error as would be perceived by a stereo sensor on a moving robotic platform.

To begin the analysis, consider the sequence of 30 images collected at the last stop point (i.e. $k$=101 in Figure 3.7a) in the experiment. In the first image of the sequence ($i$=1), *20* KLT features were initialised so that most of the artificial features in the image get selected as KLT features (see Figure 3.8a). Then, the KLT tracker was used to track these *20* features through the remaining set of images ($i$=2...30). Of the *20* features selected/tracked by KLT, 9 features were manually selected later for further analysis. Selecting a subset of 9 features was done in order to affirm that they correspond to the artificial features on the planar surface and also, that they are widely dispersed so as to give the widest possible representation of disparity behaviour in the image frame. Then for these manually selected features, corresponding disparity was calculated in each of the 30 images. This data represents observing 9 static features repeatedly for 30 times from a depth of 1m.

In Figure 3.9a, frequency histograms of the observed disparity of 4 randomly selected features from this subset are plotted. Given the fact that these observations were made from a depth of 1.0m, the mean disparity is expected to be 14.2pixel.

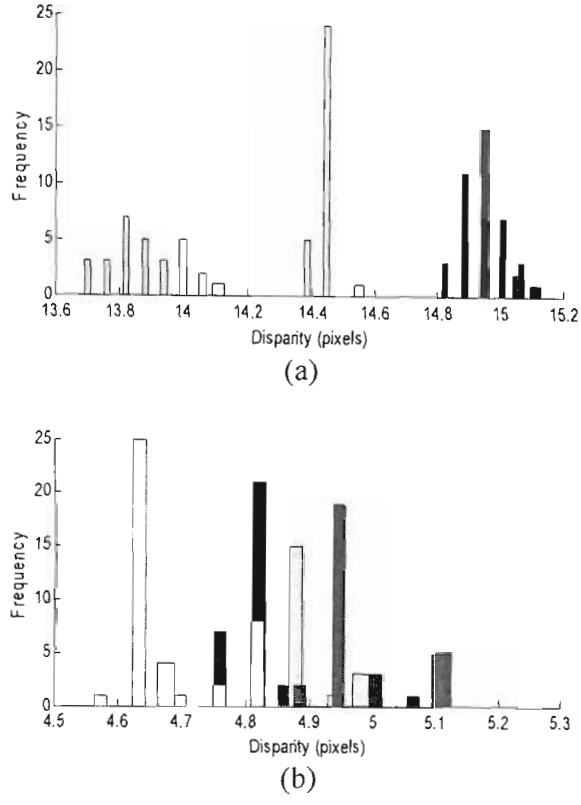
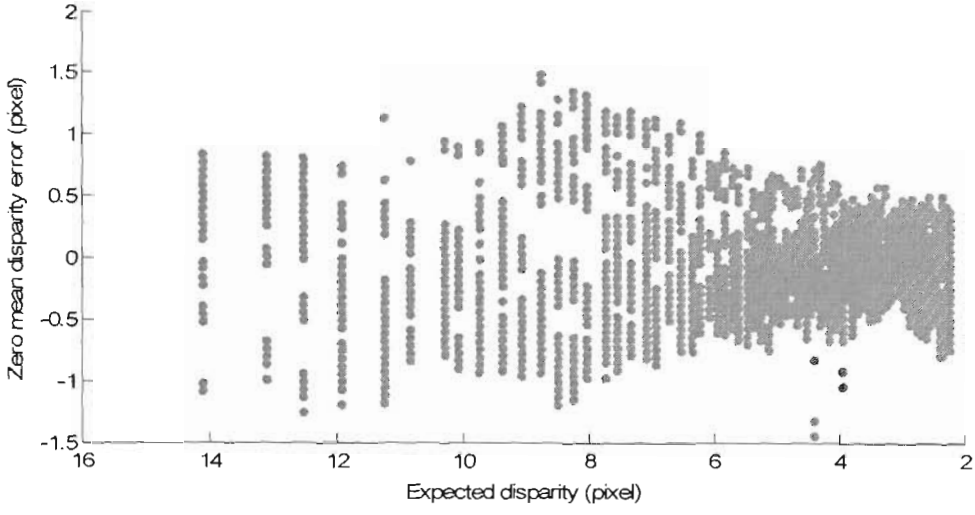



**Figure 3.9: Disparity distribution at several depths of four randomly selected artificial features. (a) depth = *1.0 m*, expected disparity = *14.2 pixel* (b) depth = *2.8 m*, *d* = *4.8 pixel***

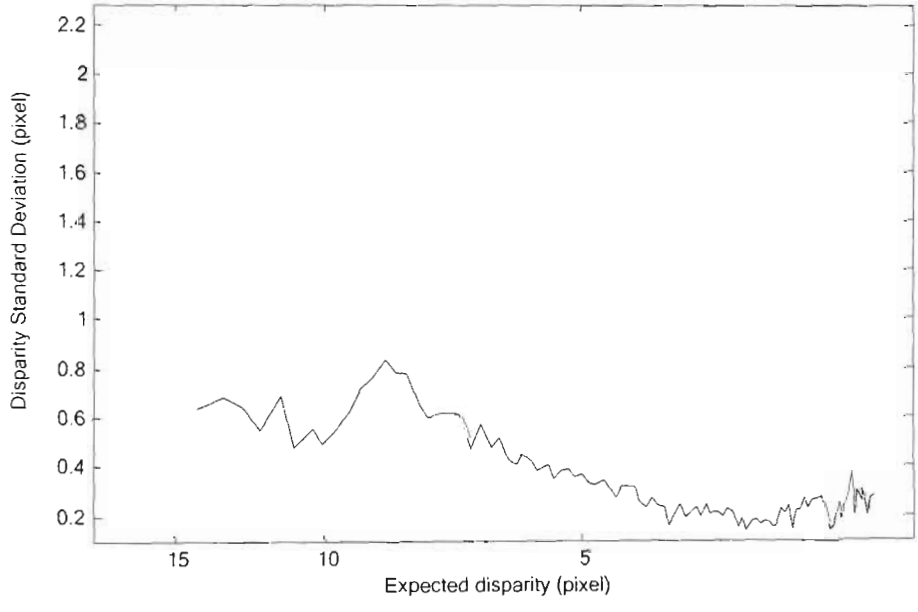
Figure 3.8b and 3.10b shows similar information for the image sequence taken at a depth of 2.8m, where expected disparity is 4.8pixel.

In each of the image sequences analysed, disparity observation histograms indicate several localised distributions within a small range of the expected disparity (Figure 3.9). These distributions are a result of a complex combination of local biases introduced in the image rectification process, distortions induced by lenses, stereo correlation mismatches and hardware misalignments between the robot and the camera that are not explicitly measurable.

Therefore, it is appropriate to analyse the data with respect to true mean at each stop point. For this, disparity observations were re-centred about zero by subtracting the




(a)




(b)

**Figure 3.10: Variation in disparity error with depth. (a) Disparity error distribution at various depths (b) Disparity error standard deviation (log scale).**

estimated mean. This approximates the true error distribution of disparity observations at various depths.

Figure 3.10a shows the frequency histogram of the disparity error plotted against the true disparity at each depth. Figure 3.10b shows the corresponding standard deviation. This approximates the expected $\sigma_d$ at various depths.

The behaviour of $\sigma_d$ warrants a special discussion, Figure 3.10b shows that the standard deviation of disparity distribution varies with the observed depth. This is contrary to the use of a constant $\sigma_d$ value to approximate the disparity standard deviation used in most of the literature. Similar observation was made by Jung [55], where he observes that the variation in disparity standard deviation is strongly correlated with the shape of the similarity score curve[1] around its peak. He then uses an empirically generated curve to estimate the disparity standard deviation. However, this data was not available from the stereo algorithm used in the experimental work of this thesis. Also, it is difficult to conclusively establish a correlation between the disparity standard deviation and the depth at which the observations were made.

Therefore, it was decided to arrive at an approximate value for the disparity standard deviation by analysing the complete data set ($k$=1,...,101) encompassing disparity observations at varying depths. It is argued that this model would better approximate the behaviour of observed disparity within the range of depths considered as opposed to the general practice of estimating disparity standard deviation by observing a set of static features from single camera location [74].

Figure 3.11 shows the overall distribution of error in observing disparity for the complete data set. For the remainder of the thesis, this distribution is assumed to approximate the true distribution that would be expected from disparity observations.

[1] Function describing the intensity difference between two windows in the left and right images.

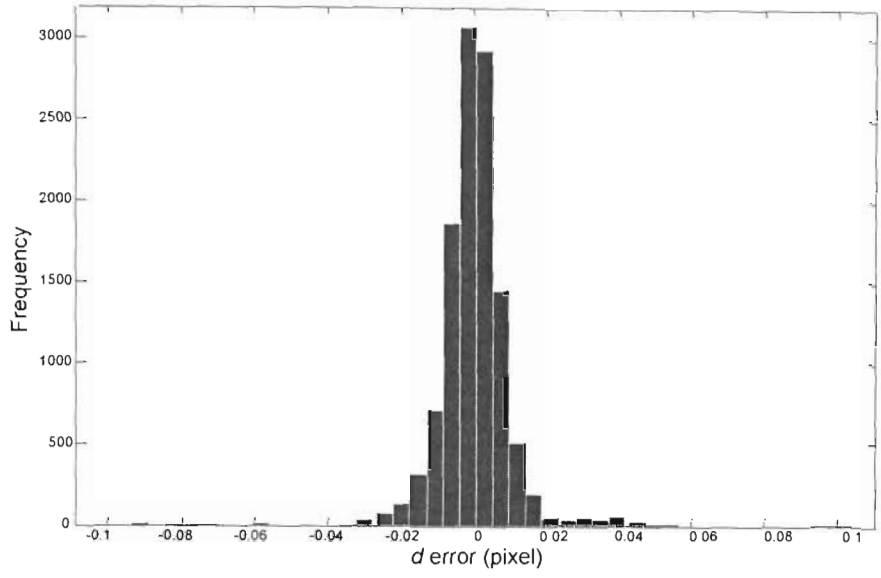



**Figure 3.11: Disparity error distribution with standard deviation $\sigma_d = 0.65$**

Through the numerous experiments that were conducted during the course of the thesis, it was found that the estimated value of $\sigma_d (= 0.65)$ was an adequate approximation for the disparity standard deviation to be used in various SLAM algorithms. By comparison, the estimated value of standard deviation is considerably higher than the disparity standard deviation seen in Figure 3.10b, reducing the risk of a SLAM algorithm becoming inconsistent due overly optimistic observations.

### 3.3.3 Behaviour of Features in the Image Plane

In order to model the errors in $u$ and $v$ for SLAM, a dynamic camera error model is considered. This includes a study of error behaviour in $u$ and $v$, in the presence of local bias due to distortion effects and misalignments as well as KLT tracking errors.

#### Tracking Uncertainty

To establish the contribution from the tracker to the overall uncertainty in the feature location, data from mapping experiment could be used. Consider the sequence of 30 images collected at the last stop point (i.e. $k$=101) in Figure 3.7a. In the first image of

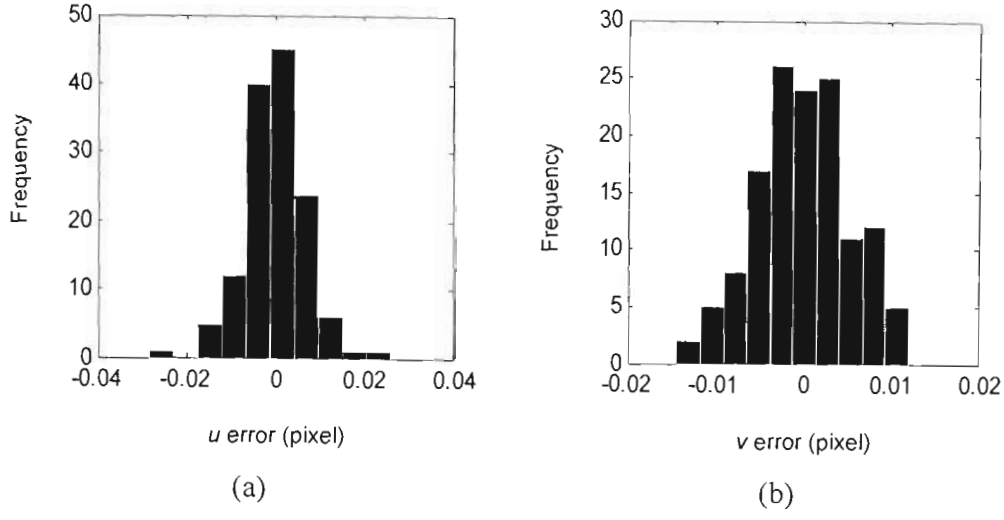



**Figure 3.12: Distribution of KLT tracking errors. (a) Distribution in u ($\sigma_u^{KLT}$ = 0.0068pixel) (b) Distribution in v ($\sigma_v^{KLT}$ =0.0053pixel)**

the sequence ($i$=1), KLT feature detector was used to initialise 30 features that adequately represent the entire image frame. Then, the KLT tracker was used to track these features through the remaining set of images ($i$=2...30).

This procedure was then repeated for remainder of the data set (k=1,...,100). The tracking errors can be analysed by ignoring the first frame and subtracting the remaining $u,v$ values by the respective means. Figure 3.12 presents the error

histograms obtained. This shows that the Probability Density Function (*pdf*) of $u$ and $v$ could be approximated by a Gaussian distribution. Most importantly, it could be observed that the standard deviations of these distributions are in the order of $10^{-3}$ pixel and thus the contribution from KLT tracker to the overall error (in $u$ & $v$) in the image plane is insignificant. However, exceptions occur when the KLT pick-up spurious features. Figure 3.13 illustrates two commonly encountered instances of such features that are in the experimental environment.

In Figure 3.13a, feature marked '1' (initialised in the left most image) is physically nonexistent. Apparent intersection is due to the two perpendicular walls that in

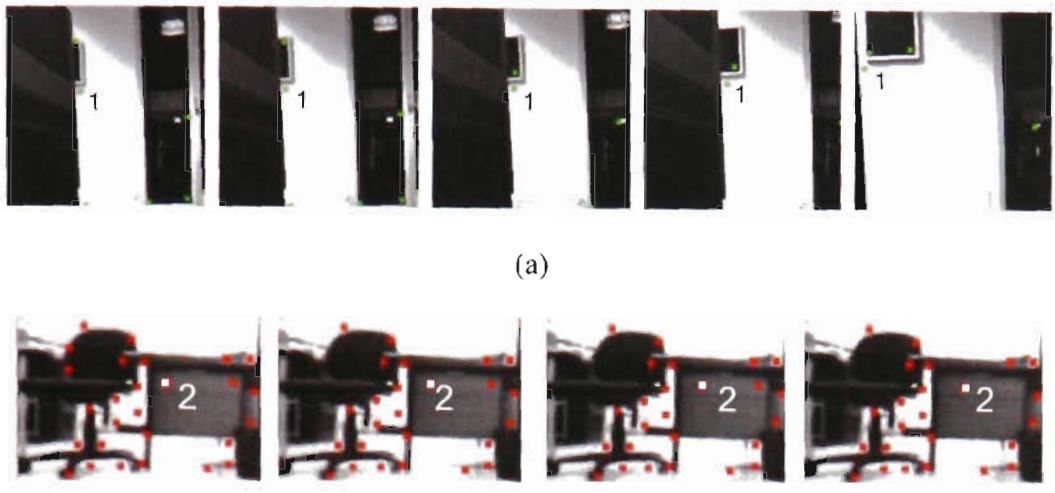



**Figure 3.13: Various tracking errors. (a) Due to occlusions (b) Due to variation in illumination**

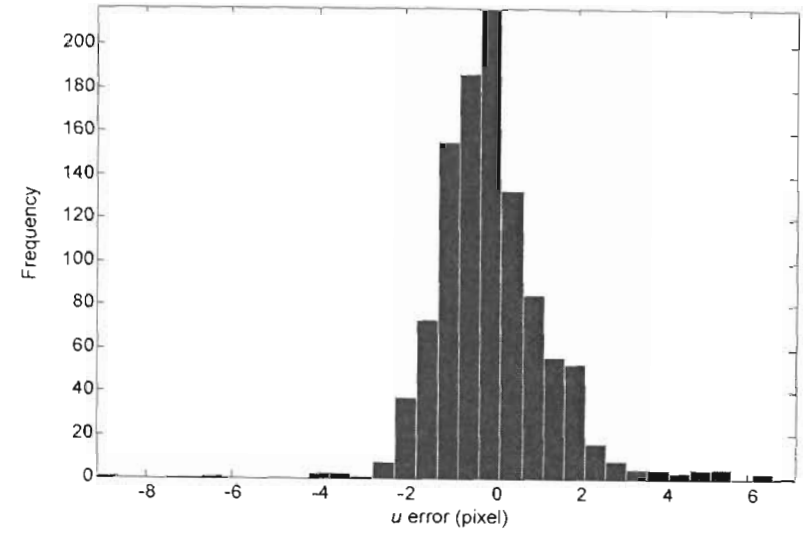
reality are separated by a $1m$ pathway between them. As the robot moves closer to the wall, the feature appears to be drifting to the left.

Figure 3.13b shows similar feature behaviour where the effect is due to change in image intensity. In this sequence a stationary camera is exposed to a static scene while varying the intensity of the illumination of the source. The camera auto gain continuously adjusts for the changing illumination. Feature marked '2' drifts to the right from its initialised position (left most image) as the gain changes. Similar effect was observed when the robot moves in the artificially illuminated test environment, for instance when the camera moves from a brightly lit area to a shaded area.

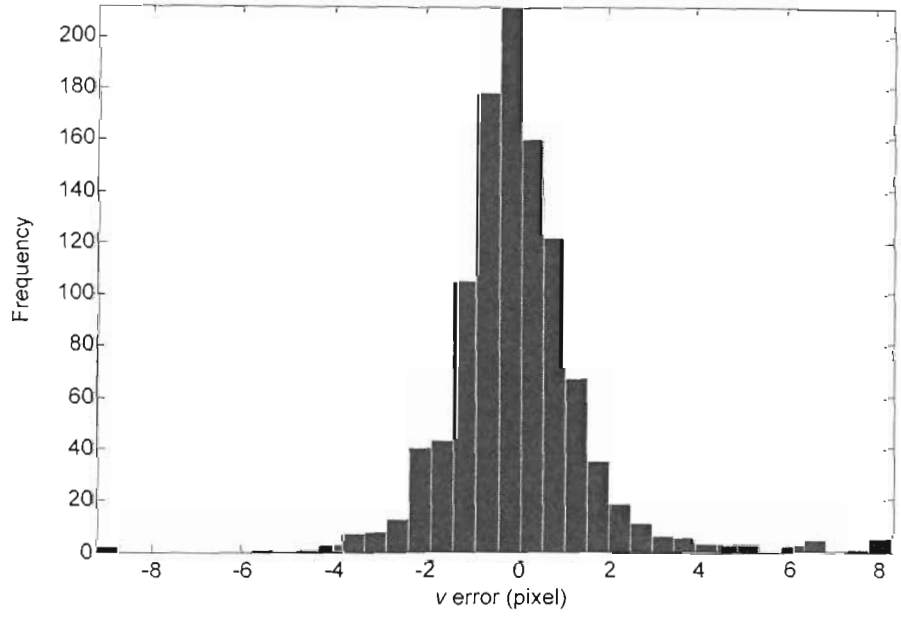
A mechanism is developed in the Chapter 6 that considerably reduces the effect of such features using a novel data validation algorithm.

### Overall Uncertainty

Not only tracking errors, but also stereo matching errors contribute to the error distributions in $u$ and $v$. Data from the mapping experiment were used again to estimate the overall error parameters. This analysis was done by considering only a single image ($i$=1) at each stop point, $k$(=1,...,101). Expectation is to capture the




(a)




(b)

**Figure 3.14: Error distributions in image feature location (a) in u with standard deviation $\sigma_u = 1.34$ (b) in v with standard deviation $\sigma_v = 1.5$**

variance in $u$ and $v$ as the camera moves (as opposed to static camera) through the scene. $n(=16)$ features were selected at $k=1$ using the KLT feature detector. These were then tracked from $k=2,\ldots, 101$ using the KLT tracker. While tracking a set of features at fixed locations in space will map to varying $u,v$ coordinates in the image plane, it essentially captures the overall behaviour of $u,v$ in the entire image plane at varying scales.

Figure 3.14 shows the error histograms of cumulated data for both $u$ and $v$ components of the observation. Qualitatively these results resemble Gaussian distributions. But it is possible to observe various artefacts appearing in the tails of the distributions indicating that a considerable amount of spurious data is present in the observations for the reasons discussed earlier. However, the standard deviations estimated using this data was found to be adequate in later SLAM implementations.

### 3.3.4 A Summary of Parameters

It is now possible to summarise all the empirical parameters used throughout the thesis. These parameters, shown in Table 3.1 were arrived at by the results discussed previously and through numerous trials during the course of the thesis. The slightly higher value of $\sigma_v$ compared to $\sigma_u$ could be attributed to small undulations that are present on the carpeted floor of the experimental environment affecting the tilt angle of the camera from pose to pose.

**Table 3.1: Summary of parameters**

| | | |
|---|---|---|
| KLT window size | | 7 x 7 (pixel$^2$) |
| KLT minimum Eigen threshold | $\lambda_{threshold}$ | 1 |
| Patch size for additional signal conditioning (SC) | | 3 x 3 (pixel$^2$) |
| Threshold standard deviation for SC | $\sigma_{threshold}$ | 1 (pixel) |
| Disparity standard deviation | $\sigma_d$ | 0.65 (pixel) |
| standard deviation in the image plane observation in $u$-direction | $\sigma_u$ | 1.34 (pixel) |
| standard deviation in the image plane observation in $v$-direction | $\sigma_v$ | 1.5 (pixel) |

## 3.4 Effects of the Non-linearity in the Projective Transformation

The projective mapping of stereo observations in the image plane to the Cartesian coordinate frame are related by the nonlinear equations (3.1). Many authors consider the probability density function of the projected observation to be Gaussian [71, 79]. However, Sibley *et al* [74] shows that when observing features at a depth corresponding to disparities near the order of a pixel (i.e near the maximum observable depth), a bias is introduced in the depth observation through the triangulation and the linearized error propagation (3.5) causes the filter to diverge. This section presents an empirical study of this bias in relation to the sensor used.

### 3.4.1 Bias

This bias could be studied by projecting the disparity data analysed in Section 3.3.2 through the following triangulation equation,

$$x_R = Bf / d$$

Results are shown in Figure 3.15. The graph clearly shows that as the observed depth increases, the distribution of disparity error tends to skew towards one side resulting in an overestimation of the mean.

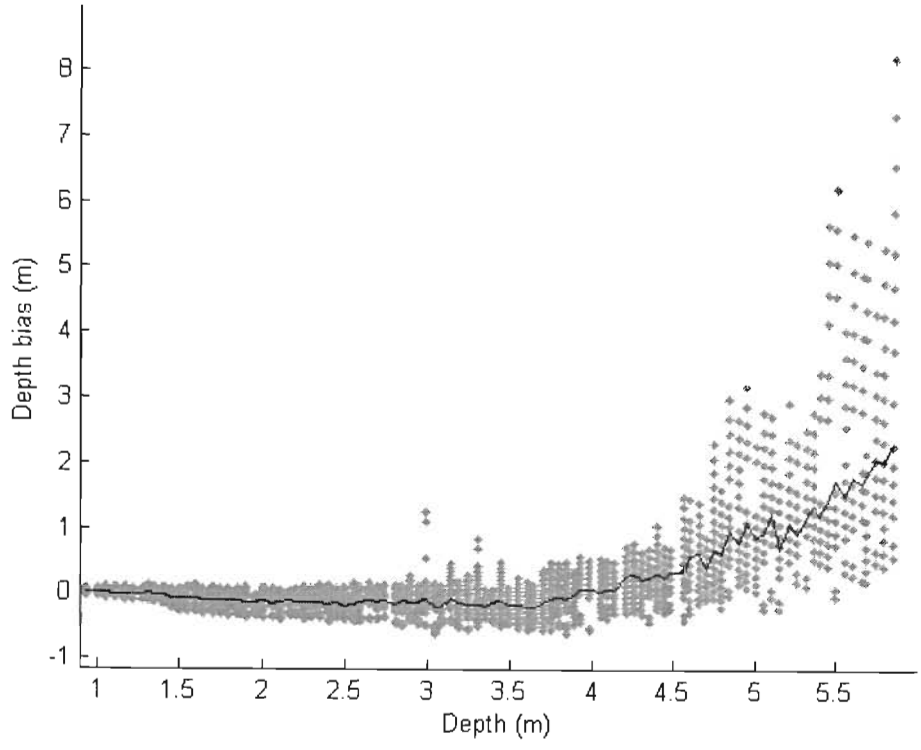



**Figure 3.15: Variation in projective errors with changing depth. The graph correlates with the data presented in Figure 3.10a. The bias in the observed depth increases with increasing depth.**

### 3.4.2 Simulated Example

Consider observing a single feature at 10m in the x-direction and 1m in the y-direction using the stereo vision sensor. In the simulation, the corresponding observations in the image $(d,u)$ space are drawn from Gaussian *pdfs* with standard deviations of $\sigma_d$ and $\sigma_u$ respectively. These observations are projected into the Cartesian frame through triangulation described by (3.2). Note that only 2-dimensions are used for clarity. Figure 3.16 shows the projected observations in the Cartesian frame and appears to be skewed with a tail towards one side. The figure also shows the true covariance ellipse of the corresponding *pdf* of the projected observations. Same figure also depicts the covariance of the projected observations estimated through linearised error propagation, which appears to be optimistic

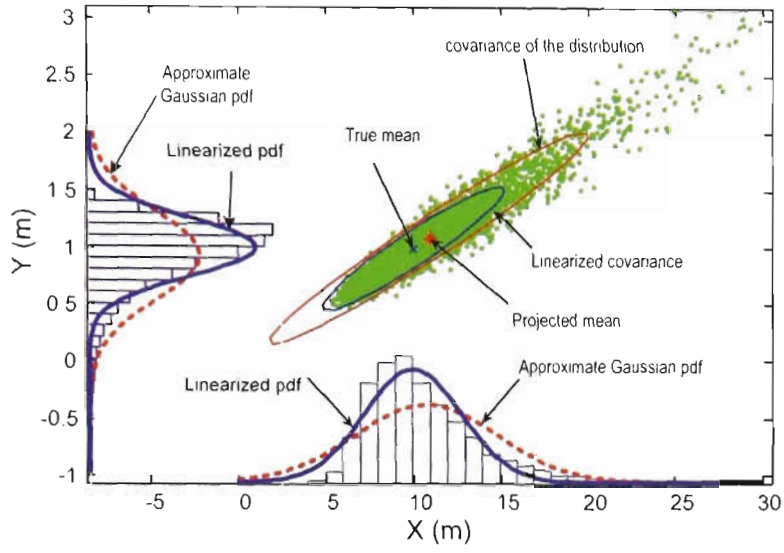



**Figure 3.16: An illustrative example of errors in projective mapping.**

compared to the true error covariance. Also, the true mean of the *pdf* is smaller than the mean obtained through linearization, indicating the presence of a bias.

In [74], an analytical approach has been taken in order to analyse this bias. Assuming that the disparity distribution is $\aleph(\bar{d},\sigma_d^2)$, then the true distribution of the projected depth is,

$$f(x_R)=\frac{fB}{\sqrt{2\pi}\sigma_d z^2}\exp\left(\frac{-(fB/x_R-\bar{d})^2}{2\sigma_d^2}\right) \tag{3.8}$$

Using equation (3.8), the *pdf* of the depth could be analysed. However, the mean of this function can not be obtained analytically and requires simulation. Figure 3.17 shows an example depth *pdf* of a feature observed from a depth of 10m and the skewed nature is clearly visible.

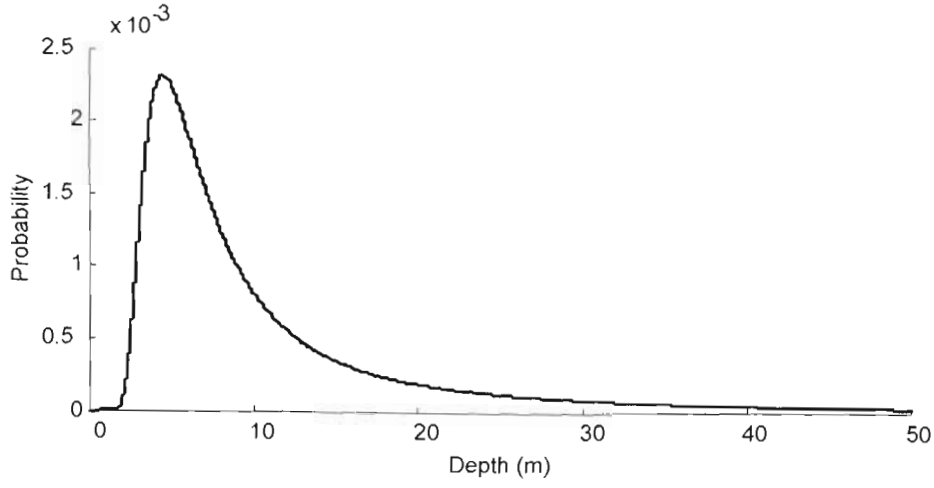



**Figure 3.17: The pdf of depth for a feature at 10m from the origin.**

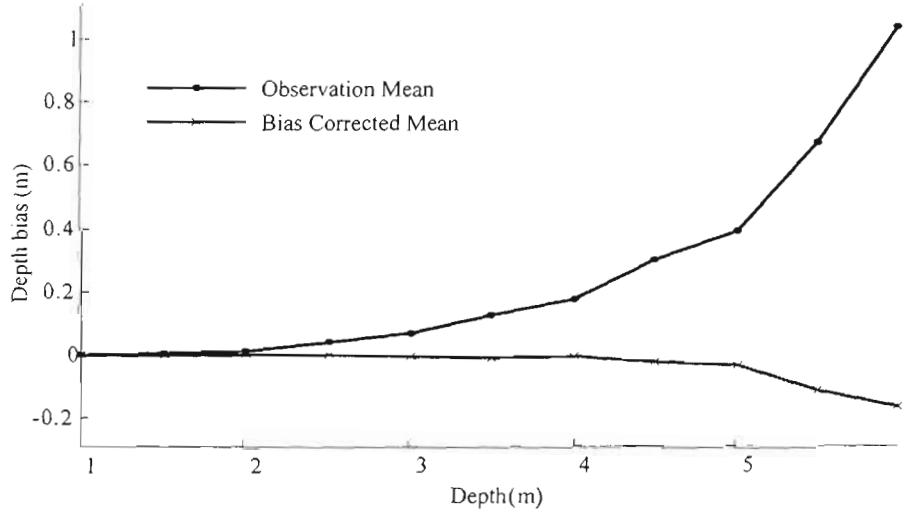



**Figure 3.18: Simulated example of bias in observed depth and application of the bias correction for several depths.** $f$ **= 150 pixels,** $B$ **= 0.088 m and** $\sigma_d$ **= 0.65 pixels**

Sibley *et al* [74] conclude that the mean is always over estimated and proposes a correction to the bias,

$$x_R(\hat{d}) \approx \tilde{x}_R(\hat{d}) - \frac{1}{2}\sigma_{\hat{d}} \left.\frac{\partial^2 x_R}{\partial d^2}\right|_{\hat{d}} \tag{3.9}$$

Figure 3.18 shows the bias in the depth along with the effects of the correction (3.9) applied to a set of simulated stereo data. The depth was generated using (3.2) and realistic noise parameters from Table 3.1 were used for spawning the disparity observations.

It is worthwhile comparing the bias seen in Figure 3.18 using simulated data with the results seen using real data in Figure 3.15. The biases appear to follow similar trends. However, due to the fact that this bias correction requires exact knowledge of $\sigma_d$, which is difficult to estimate accurately, the correction proposed by Sibley *et al* was not applied in this work[1].

Baseline of the camera also affects the behaviour of the bias. Figure 3.19 shows several *pdfs* of projected observations estimated using (3.8) for baselines ranging from 0.088m to 0.88m. The observed depth was 3.0m. This clearly shows that the *pdf* of the projected observation tends to deviate from the Gaussian behaviour as the baseline is decreased. Importantly, this indicates that the non-linear effects of the sensor model manifest at short depth for short baseline cameras.

The deviation from Gaussian distribution of the projected observations is aggravated when the standard deviation of the disparity observations is higher. A comparison of projected error distributions with varying $\sigma_d$ and baselines are shown in Figure 3.20. The dashed ellipse indicates the estimated mean of the projected observation while the solid line shows the covariance estimated through linearised error propagation. Figure 3.20b corresponds to a stereo camera with a baseline of 0.5m. This figure

[1] During EKF based SLAM experiments using real stereo data, the improvement in filter estimates with the bias correction applied was not quantifiable against results obtained without this correction.

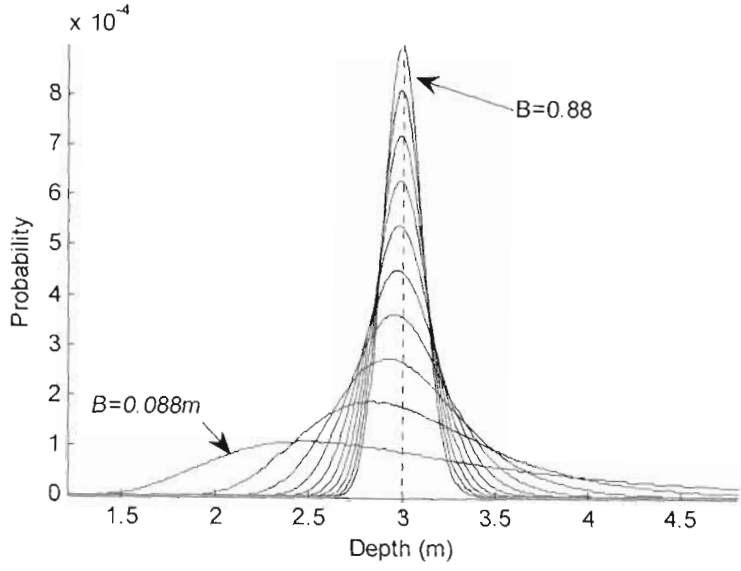
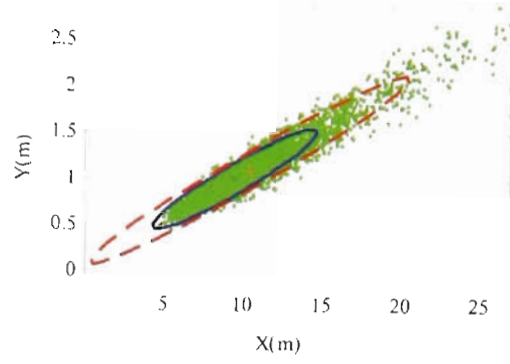



**Figure 3.19: Probability density functions of depth for varying baselines. $f$=150 pixels, $B$ =0.088 – 0.88 m and $\sigma_d$ =0.65 pixels. Measured depth=3.0 m.**

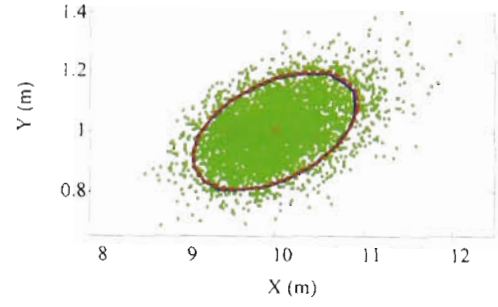
shows that both the mean and covariance derived from linearization are in agreement. Also, the observations appear to follow a Gaussian distribution around the true mean. In Figure 3.20c, the baseline corresponds to the camera used in the work of this thesis, but the disparity standard deviation has been reduced to that presented in [74] and the covariances appear to be similar.

These simulations demonstrate that the short baseline contributes negatively to the severity of the bias. Hence, the effects of bias in projective mapping appear at shorter depth. Also, with the increase in disparity standard deviation, the covariance estimated through linear error propagation tends to become more optimistic.

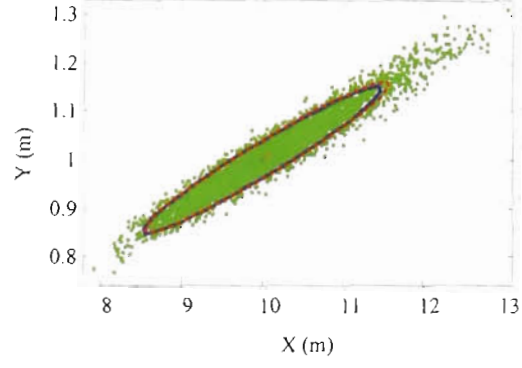
An empirical compromise to alleviate the effects of short baseline has been suggested based on the ratio (*baseline/depth*) > (1/30) [55] where any depth measurements beyond this limit are discarded. In the sensor considered, this ratio translates to observing ranges less than 2.6m. This is far from acceptable given that for a successful implementation of SLAM features need to be observed for prolong periods of time, especially from a distance to improve the heading errors. Therefore,




(a)




(b)




(c)

**Figure 3.20: Effects of baseline and disparity standard deviation on the projected mapping. Feature is placed at (10m,1m). Note that the figures are not to the same scale for clarity. (a) Baseline ($B$) = 0.088m, $\sigma_d$ = 0.65, (b) $B$ = 0.5m, $\sigma_d$ = 0.65, (c) $B$ = 0.088m, $\sigma_d$ = 0.3**

alternative approaches that dampen the sensor effects will be studied in the next chapter.

## 3.5 Discussion

This chapter presented a characterisation of the sensor to be used in the remainder of the thesis. The underlying theme is that of a sensor capable of not only perceiving depth but also extracting natural features from the surroundings in order to successfully localise itself and map the environment simultaneously.

The general character of the sensor could be summarised as having the primary observation mode to be in the image plane $(u,v,d)$ with Gaussian observation noise. However, it was shown through empirical evidence and with reference to recent analytical results, that the projected observations in 3D do not necessarily follow a Gaussian error distribution as generally accepted. This non-Gaussian nature mainly stems from the short baseline of the stereo system; hence it is not dependent on the feature selection/tracking algorithm. Also it was shown that the projected observations carry a statistical bias and the sensor is prone to produce considerable amount of spurious data even after the implementation of several filtering algorithms.

As will be shown in the next chapter, sporadic nature of the errors in disparity, statistical bias in the projective mapping and nonlinear nature of the sensor model contribute to inconsistent performance of classical SLAM implementations. In fact, this challenging nature of the sensor is the motivation for later developments presented in the thesis

# Chapter 4

# Stereo Vision based Simultaneous Localisation and Mapping

## 4.1 Introduction

The stereo vision sensor detects, tracks and provides the 3D coordinates of natural features in the environment. These are the key requirements of a sensor to be used in solving the SLAM problem. Salient characteristics of this sensor have been presented in Chapter 3. This chapter begins with a study on the consistency of state estimates using traditional solutions to the SLAM problem with the stereo vision sensor. Later, it addresses three key issues relating to practical implementation of the 3D SLAM algorithm.

Section 4.2 begins by formulating an EKF solution to the SLAM problem using the stereo vision system. A Monte Carlo experiment is then presented to discuss the effects of nonlinear nature of the associated observation equations on the consistency of the SLAM algorithm, highlighting the relationship between filter inconsistency and the length of the baseline of the stereo camera. Section 4.3 investigates several alternative strategies in order to avoid the inconsistency seen with the EKF

implementation in section 4.2. However, simulation results indicate that these approaches are ineffective and that an EKF still yields inconsistent state estimates. As a solution, section 4.4 poses SLAM as a nonlinear batch optimisation problem. A novel formulation is presented for constraining the optimisation through odometry. A scheme for initialising the batch algorithm is also presented with simulation results showing consistent state estimates.

## 4.2 Recursive State Estimation Using the Triangulated Observation Model

The EKF has been successfully applied in SLAM implementations for recursive estimation of robot and map states, in widely differing sensor domains [27, 80, 81]. A straightforward implementation of the EKF SLAM algorithm is investigated in this section using the stereo sensor presented in Chapter 3.

### 4.2.1 Observation Model

The traditional SLAM observation model consists of range and bearing measurements to features in the 2D [18, 24, 81]. A natural extension into 3D is to use range, bearing and elevation in the observation model [82]. However, from a computer vision perspective, a natural way of representing the observations is found in the Cartesian coordinate frame [83]. The model presented in section 3.2 represents a 3D observation in the Cartesian frame obtained through triangulation. With the *x-axis* being in the forward looking direction of the robot, the expected observation is related to the features and the robot location through,

$$Z(k+1) = \begin{bmatrix} z_x(k+1) \\ z_y(k+1) \\ z_z(k+1) \end{bmatrix} = \begin{bmatrix} x \\ y \\ z_{fi}(k+1) \end{bmatrix} \tag{4.1}$$

where,

$$x = (x_{fi}(k+1) - x_r(k+1))\cos(\varphi_r(k)) + (y_{fi}(k+1) - y_r(k+1))\sin(\varphi_r(k))$$
$$y = -(x_{fi}(k+1) - x_r(k+1))\sin(\varphi_r(k)) + (y_{fi}(k+1) - y_r(k+1))\cos(\varphi_r(k))$$

### 4.2.2 Simulation

This section presents results of the application of the EKF (section 2.2.1) using the previous sensor model. Simulations are carried out using the environment presented in section 2.4.1. However, the features are now assumed to be observed using the stereo camera. Therefore, observation vector $\mathbf{z}(k) = [z_x, z_y, z_z]^T$ is now generated through stereo triangulation: $\mathbb{R}^3 \rightarrow \mathbb{R}^3$,

$$\begin{bmatrix} z_x \\ z_y \\ z_z \end{bmatrix} = \frac{B}{z_d} \begin{bmatrix} f \\ -z_u \\ -z_v \end{bmatrix} \tag{4.2}$$

The observations in the image plane $(z_u, z_v, z_d)$ are drawn from Gaussian *pdfs* with parameters that emulate experimental values in Table 3.1. As developed in section 3.2.1, the associated covariance matrix of the observation is,

$$\mathbf{R}_{xyz} = \frac{B^2}{d^2} \begin{bmatrix} \frac{f^2\,\sigma_d^2}{z_d^2} & \frac{-z_u f\,\sigma_d^2}{z_d^2} & \frac{-z_v f\,\sigma_d^2}{z_d^2} \\ \frac{-z_u f\,\sigma_d^2}{z_d^2} & \sigma_u^2 + \frac{z_u^2 \sigma_d^2}{z_d^2} & \frac{z_u z_v\,\sigma_d^2}{z_d^2} \\ \frac{-z_v f\,\sigma_d^2}{z_d^2} & \frac{z_u z_v\,\sigma_d^2}{z_d^2} & \sigma_v^2 + \frac{z_v^2 \sigma_d^2}{z_d^2} \end{bmatrix} \tag{4.3}$$

The values of the camera baseline ($B \approx 0.09m$) and focal length ($f = 150 pixel$) correspond to the experimental camera rig (Appendix A.1). Data association was assumed to be known completely. The noise parameters used in the filter were selected to exactly match the noise injected to observation and control models.

Figure 4.1 shows the estimator errors and the 95% confidence bounds for the robot pose indicating that the filter is inconsistent. Figure 4.2 shows the average NEES results of the pose estimate (see section 2.4) for a set of 50 simulations, confirming the fact that the estimator is inconsistent.

It is known that linearization errors can cause an EKF based SLAM estimates to produce optimistic results [84]. Several other work including Bailey *et al* [47] and, Julier and Uhlman [49] have also shown that the SLAM estimator output become inconsistent due to the inherently nonlinear nature of the SLAM problem. However, observing inconsistency in this simulation environment which covers an area of approximately 8x8m is unexpected.

To examine the effect of the baseline of the camera, which was shown in Chapter 3 to have a significant impact in the linearization errors, a separate set of simulated experiments were carried out by changing the sensor baseline from 0.2m to 0.5m, while keeping other parameters unchanged. The above baseline range was selected considering that some of the successful stereo vision based SLAM implementations up-to-date have considered stereo cameras with similar (in [56] $B = 0.336m$) or larger (in [85] $B = 2.2m$) baselines. Average NEES of robot pose for a set of 50 simulations with the varying baselines are shown in Figure 4.3. As can be seen from the figure, the filter becomes consistent as the baseline reaches 0.5m. These results indicate that the nonlinearity of the observation model has a significant impact on the performance of the SLAM estimators based on stereo vision and the inconsistency is correlated to the length of the camera baseline.

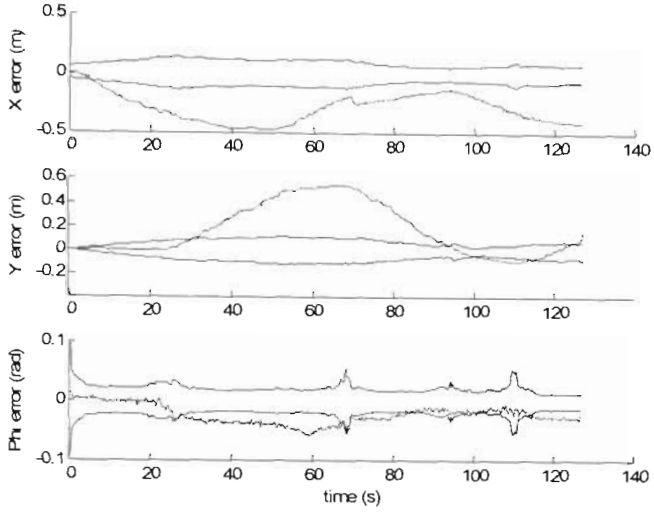



**Figure 4.1: Robot pose errors and the estimated $2\sigma$ error bounds. For each of the three dimensions the error covariance is optimistic indicating inconsistency.**

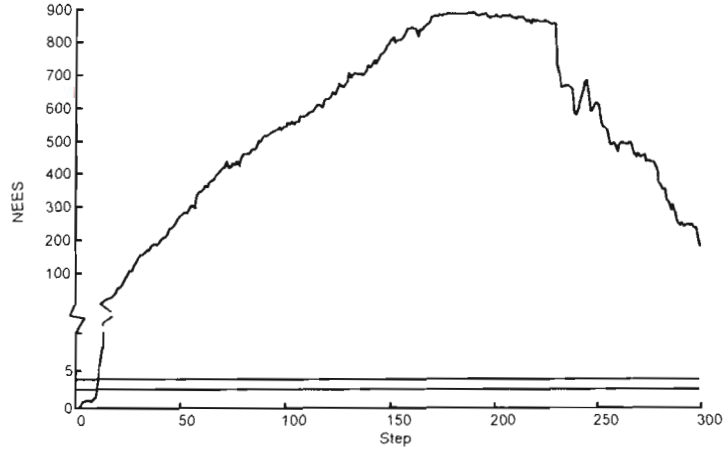
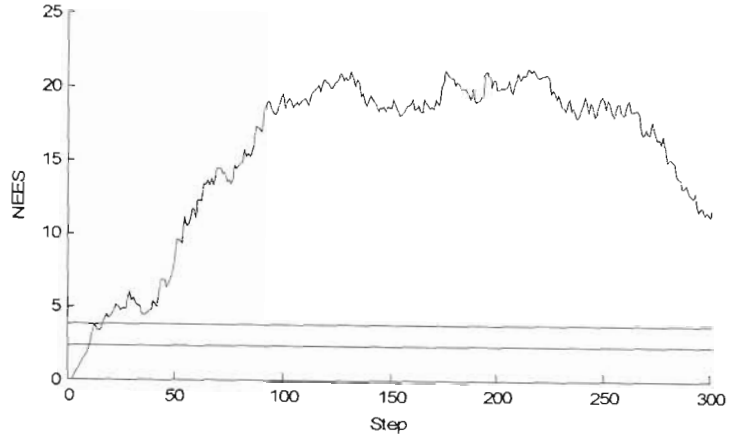
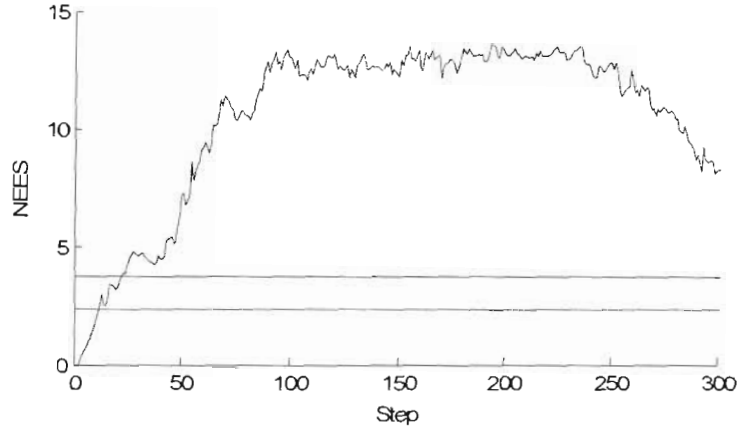



**Figure 4.2: Average NEES of the robot pose for a set of 50 runs using traditional EKF SLAM. The average results are well over the 95% confidence bounds indicating an overly optimistic filter.**

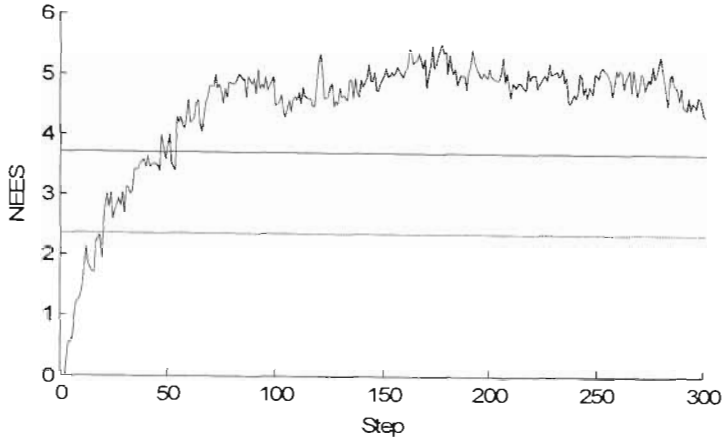



(a)

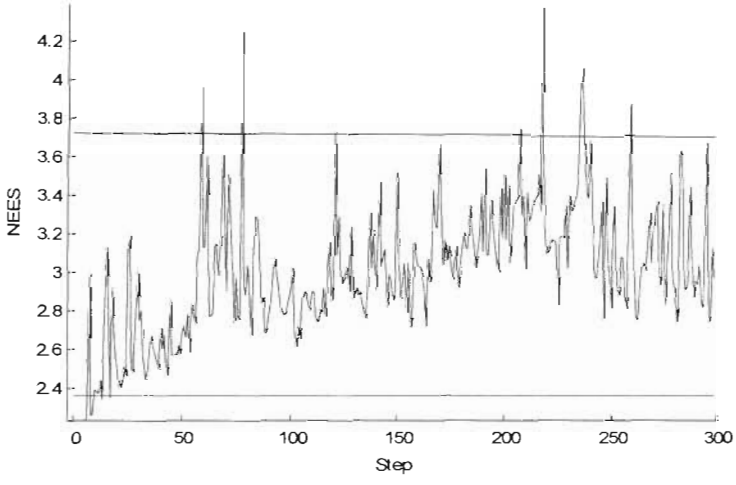



(b)

**Figure 4.3: Average NEES estimates of the robot pose using 50 runs of traditional EKF based SLAM for varying lengths of baseline (a) stereo camera with a baseline of 0.2m was used. (b) baseline of 0.3m was used. (continued next page...)**




(c)




(d)

**Figure 4.3 (continued): Average NEES of the robot pose for a 50 runs using a traditional EKF based SLAM solution where, (c) stereo camera with a baseline of 0.4m was used (d) stereo camera with a baseline of 0.5m was used.**

## 4.3 Alternative Approaches

This section discusses several alternative estimation approaches that have been known to perform better in the presence of highly nonlinear system models.

### 4.3.1 An Alternative Observation Model

Bar-Shalom *et al*, [86, 87] has observed that in tracking using sonar and radar, coordinate conversions from polar to Cartesian tend to create inconsistent filter performance. As part of the strategy in improving filter consistency it is suggested that the observation model to be formulated in its original polar coordinates.

Thus, an alternative approach to the classical observation model presented in Section 3.2.1 is to define the observation model in the image coordinate frame by its location $(u, v)$ and its disparity $(d)$. The new observation model is,

$$\mathbf{z}(k+1) = \begin{bmatrix} z_u(k+1) \\ z_v(k+1) \\ z_d(k+1) \end{bmatrix} = \frac{f}{x} \begin{bmatrix} y \\ z_{fi}(k+1) \\ B \end{bmatrix} \tag{4.4}$$

where

$$x = (x_{fi}(k+1) - x_r(k+1))\cos(\phi(k)) + (y_{fi}(k+1) - y_r(k+1))\sin(\phi(k))$$
$$y = -(x_{fi}(k+1) - x_r(k+1))\sin(\phi(k)) + (y_{fi}(k+1) - y_r(k+1))\cos(\phi(k))$$

The observation noise covariance matrix is now diagonal (section 3.3),

$$\mathbf{R}(k+1) = \begin{bmatrix} \sigma_u^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_d^2 \end{bmatrix} \tag{4.5}$$

Even though the new observation model has better Gaussian characteristics, it should be noted that in the application of EKF, the nonlinear transform in (4.2) and linearization is still required during feature initialisation and prediction phases of the SLAM algorithm.

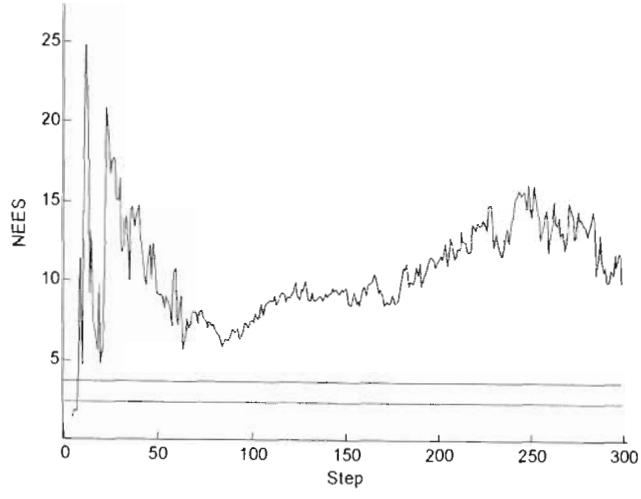



**Figure 4.4: Average NEES of the robot pose using EKF and the alternative observation model.**

Figure 4.4 shows the average NEES of the robot pose with the new observation model using the data set from the simulations presented in Section 4.2.2. As can be seen from the figure, that while the average NEES has significantly reduced compared to the NEES estimates obtained using the previous observation model, the estimates are still inconsistent. This compels to consider alternative filtering mechanisms that are known to perform better over EKF when the observation models are highly nonlinear.

### 4.3.2 Unscented Kalman Filter

The Sigma point filter or the Unscented Kalman Filter (UKF) [6, 9, 88] performs the nonlinear transform discussed previously through a deterministic weighted sampling scheme which captures the posterior mean and covariance to an accuracy of third order compared to the first order approximation of a Taylor series expansion used in EKF. Interestingly, both EKF and UKF are of the same order of complexity, hence UKF could also be implemented in real time. This section presents results of the application of the UKF with the observation model given by (4.4) to the SLAM

problem. The Unscented Transformation (UT) forms the technique for addressing the issue of linearized approximation of the EKF. The UT is discussed in brief next.

Assume that, it is required to propagate a random variable $\mathbf{x}$ of dimension $n$ through the nonlinear function,

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) \tag{4.6}$$

then, if the mean of $\mathbf{x}$ is $\bar{\mathbf{x}}$ and the covariance is $\mathbf{P}_x$, in order to calculate the statistics of $\mathbf{y}$, a set of $2n+1$ sigma vectors $\chi_i$ are formed according to,

$$\begin{aligned}
\chi_0 &= \bar{\mathbf{x}} \\
\chi_i &= \bar{\mathbf{x}} + \left(\sqrt{(n+\lambda)\mathbf{P}_x}\right)_i \quad i = 1,\ldots,n \\
\chi_i &= \bar{\mathbf{x}} - \left(\sqrt{(n+\lambda)\mathbf{P}_x}\right)_{i-n} \quad i = n+1,\ldots,2n
\end{aligned} \tag{4.7}$$

with the corresponding weights,

$$\begin{aligned}
W_0^m &= \lambda/(n+\lambda) \\
W_0^c &= \lambda/(n+\lambda) + (1-\alpha^2+\beta) \\
W_i^m &= W_i^c = 1/2(n+\lambda) \quad i = 1,\ldots,2n
\end{aligned} \tag{4.8}$$

where $\lambda = \alpha^2(n+k) - n$ is a scaling parameter. $\alpha$ (set to $1e-3$) determines the spread of the sigma points around the mean $\bar{\mathbf{x}}$. $k$ is a secondary scaling parameter (set to zero in most cases). For a Gaussian distribution $\beta = 2$ is optimal. The term $\left(\sqrt{(n+\lambda)\mathbf{P}_x}\right)_i$ represents the $i^{\text{th}}$ row of the matrix square root. These sigma vectors are then propagated through the nonlinear function (4.6) such that,

$$\mathbf{Y}_i = \mathbf{g}(\chi_i) \quad i = 0,\ldots,2n \tag{4.9}$$

Then the mean of $\mathbf{y}$ is found by the weighted sample average of the posterior means of the sigma points.

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2n} W_i^m \mathbf{Y}_i \tag{4.10}$$

and the covariance of the distribution of $\mathbf{y}$ is found by,

$$\mathbf{P}_{\mathbf{y}} \approx \sum_{i=0}^{2n} W_i^c \{\mathbf{Y}_i - \bar{\mathbf{y}}\}\{\mathbf{Y}_i - \bar{\mathbf{y}}\}^T \tag{4.11}$$

The Unscented Transform could be used in order to propagate the prior statistics through the nonlinear observation and process models to calculate the resulting mean and covariance of the respective distributions. The UKF follows the recursive pattern of the EKF algorithm with the state at time $k$ represented by a random variable $\mathbf{x}(k) \sim p(\mathbf{x}(k))$ with the estimated mean and covariance,

$$\begin{aligned} \hat{\mathbf{x}}(k) &= E\{\mathbf{x}(k)\} \\ \hat{\mathbf{P}}(k) &= E\{(\mathbf{x}(k) - \hat{\mathbf{x}}(k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k))^T\} \end{aligned} \tag{4.12}$$

Then the sigma points at time $k-1$ are,

$$\boldsymbol{\chi}(k-1) = \left[\hat{\mathbf{x}}(k-1) \quad \hat{\mathbf{x}}(k-1) \pm \sqrt{(n+\lambda)\mathbf{P}(k-1)}\right] \tag{4.13}$$

The prediction,

$$\begin{aligned} \boldsymbol{\chi}^*(k \mid k-1) &= \mathbf{f}(\boldsymbol{\chi}(k-1), \mathbf{u}(k)) \\ \hat{\mathbf{x}}^-(k) &= \sum_{i=0}^{n} W_i^m \boldsymbol{\chi}_i^*(k \mid k-1) \\ \mathbf{P}^-(k) &= \sum_{i=0}^{2n} W_i^c [\boldsymbol{\chi}_i^*(k \mid k-1) - \hat{\mathbf{x}}^-(k)][\boldsymbol{\chi}_i^*(k \mid k-1) - \hat{\mathbf{x}}^-(k)]^T \end{aligned} \tag{4.14}$$

Observation prediction,

$$\begin{aligned}\boldsymbol{\chi}(k \mid k-1) &= \left[\hat{\mathbf{x}}^{-}(k) \quad \hat{\mathbf{x}}^{-}(k) \pm \sqrt{(n+\lambda)\mathbf{P}^{-}(k)}\right] \\ \mathbf{Z}(k \mid k-1) &= \mathbf{h}[\boldsymbol{\chi}(k \mid k-1)] \\ \hat{\mathbf{z}}^{-}(k) &= \sum_{i=0}^{2n} W_i^m \mathbf{Z}_i(k \mid k-1)\end{aligned} \tag{4.15}$$

and the update,

$$\begin{aligned}\mathbf{S}(k) &= \sum_{i=0}^{2n} W_i^c [\mathbf{Z}_i(k \mid k-1) - \hat{\mathbf{z}}^{-}(k)][\mathbf{Z}_i(k \mid k-1) - \hat{\mathbf{z}}^{-}(k)]^T + \mathbf{R}(k) \\ \mathbf{P}^{x,z}(k) &= \sum_{i=0}^{2n} W_i^c [\boldsymbol{\chi}_i(k \mid k-1) - \hat{\mathbf{x}}^{-}(k)][\boldsymbol{\chi}_i(k \mid k-1) - \hat{\mathbf{x}}^{-}(k)]^T \\ \mathbf{K}(k) &= \mathbf{P}^{x,z}(k)\mathbf{S}^{-1}(k) \\ \hat{\mathbf{x}}(k) &= \hat{\mathbf{x}}^{-}(k)\mathbf{K}(k)(\mathbf{z}(k) - \hat{\mathbf{z}}^{-}(k)) \\ \mathbf{P}(k) &= \mathbf{P}^{-}(k) - \mathbf{K}(k)\mathbf{S}(k)\mathbf{K}^T(k)\end{aligned} \tag{4.16}$$

Having defined the UKF, the previous set of data could be used to evaluate the estimator consistency with respect to the observation model given in (4.4). Figure 4.5 shows the average NEES for the 50 trials. As can be seen, compared to the results shown in Figure 4.4, a reduction in the filter bias is achieved with the UKF. However, estimates still remain inconsistent.

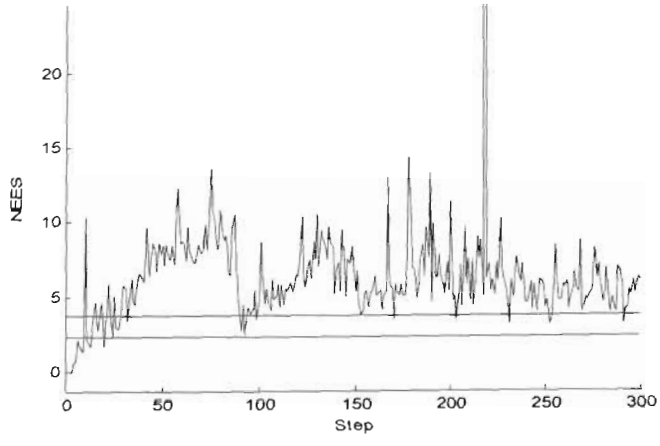



**Figure 4.5: Average NEES of the robot pose for the Unscented Filter implementation.**

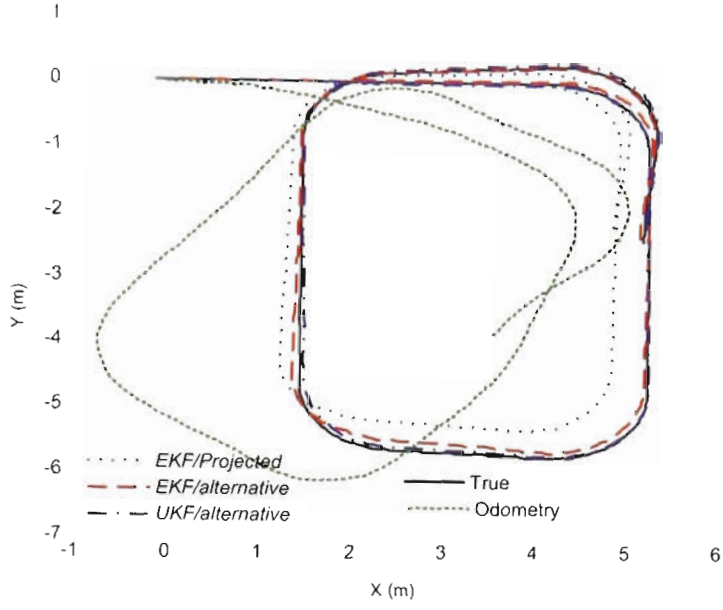
### 4.3.3 A Comparison of Estimator Error

Table 4.1 summarises the average sum squared errors of the pose estimates from the previous examples. The lowest error amongst them is recorded for the UKF which is an improvement of 67% over the EKF implementation.

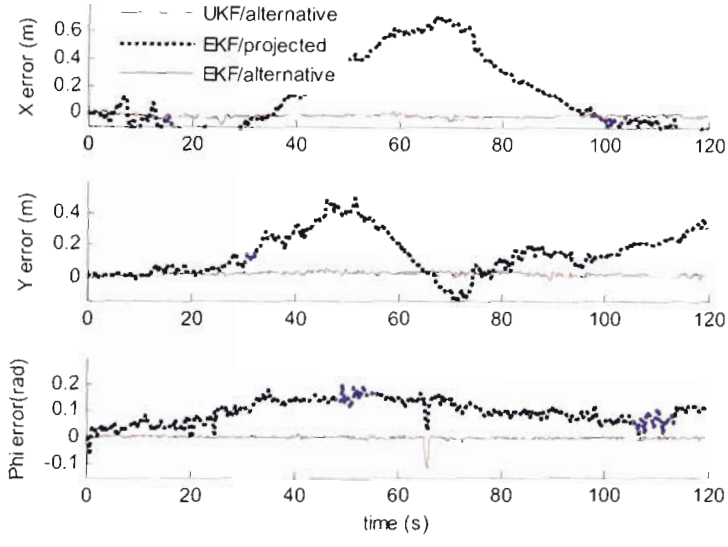
Figure 4.6a shows the estimated robot trajectory for different observation model/filter pair. This provides a visual perspective of scale of errors involved with the simulated robot trajectory. In Figure 4.6b the error in estimated robot poses for the same observation model/filter pair are plotted. It is clear that when the observation model assumes that of (4.2) the path estimate carries the largest error. Even though the estimates are inconsistent, it is interesting to note that the absolute error is reasonably small, especially for the alternative approaches presented in section 4.3. This fact is exploited in developments presented later.

**Table 4.1: A comparison of sum squared errors in the robot position for the different formulations. The Unscented Filter combined with the alternative observation model yields the smallest error**

| Filter/Obs. Model | EKF/projected | EKF/alternative | UKF/alternative |
|---|---|---|---|
| Sum Squared Error | 4.8220 | 3.4752 | 1.6032 |




(a)




(b)

**Figure 4.6: A comparison of estimator errors. (a) Robot path estimates using EKF with the Cartesian observation model (EKF/projected), EKF with the alternative observation model (EKF/alternative) and UKF with the alternative observation model (UKF/alternative). (b) Robot pose estimate errors.**

### 4.3.4 Conclusions

A cross section of recursive estimation techniques has been discussed in this section in relation to solving the SLAM problem using stereo vision. A small scale simulated environment was used to illustrate that all the recursive methods examined produce inconsistent state estimates. This was attributed to the highly nonlinear nature of the observation model. Two alternative observation models were presented along with a combination of recursive estimators, namely the Extended Kalman Filter and the Unscented Kalman Filter. The later, generally known to better handle highly nonlinear models, however did not produce consistent state estimates. Even though the filter estimates were inconsistent, the absolute errors were small and visually appeared to follow the true estimates with reasonable accuracy. This fact has significant empirical importance in the later developments presented in the thesis.

## 4.4 Estimation Through Non-linear Batch Optimisation

Nonlinear batch optimisation techniques generally preserve the entire history of the states to be estimated. The large number of observations used in a single optimisation cycle could potentially improve the linearization due to more accurate estimates being available for use in the linearization process.

This section formulates the general nonlinear batch optimisation method discussed in chapter 2 in a SLAM context. Odometry measurements and nonholonomic process model are used as constraints during the optimisation process. It is shown that this approach yields consistent state estimates.

Recall the maximum a posteriori (MAP) estimate defined in chapter 2,

$$\mathbf{x}^{*} \triangleq \underset{\mathbf{x}}{\operatorname{argmax}} \quad p(\mathbf{z} \mid \mathbf{x}) p(\mathbf{x}) \tag{4.17}$$

In order to formulate SLAM as an optimisation problem, the associated states and the observation models need to be defined.

### 4.4.1 System States

Let the robot pose at the $i^{th}$ time step be,

$$\mathbf{x}_{ri} = \begin{bmatrix} x_{ri} \\ y_{ri} \\ \phi_{ri} \end{bmatrix} \tag{4.18}$$

Each feature $\mathbf{x}_{fi}$ can be defined by a point in 3-D as,

$$\mathbf{x}_{fi}(k) = \begin{bmatrix} x_{fi}(k) \\ y_{fi}(k) \\ z_{fi}(k) \end{bmatrix} \tag{4.19}$$

then the augmented state vector is given by,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_m \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{r0} & \mathbf{x}_{r1} & \cdots & \mathbf{x}_m & \mathbf{x}_{f0} & \mathbf{x}_{f1} & \cdots & \mathbf{x}_{fM} \end{bmatrix}^T \tag{4.20}$$

where $\mathbf{x}_r \triangleq \{\mathbf{x}_{ri}\}$ describes the entire robot trajectory $(1 \ldots N)$ and $\mathbf{x}_m \triangleq \{\mathbf{x}_{fi}\}$ is the map with $(1 \ldots M)$ features.

### 4.4.2 Observations to Features

Assuming a Gaussian observation model,

$$p(\mathbf{z}_i \mid \mathbf{x}) = \eta e^{-\frac{1}{2}\boldsymbol{\varepsilon}_i^T \mathbf{R}_i^{-1} \boldsymbol{\varepsilon}_i} \tag{4.21}$$

where,

$$\boldsymbol{\varepsilon}_i = (\mathbf{z}_i - \mathbf{h}_i(\mathbf{x})) \quad i = 1, \ldots, z \tag{4.22}$$

is the innovation of the stereo vision observation and $\mathbf{h}_i(\mathbf{x})$ is the predicted observation to the feature. The particular observation model $\mathbf{h}_i(\mathbf{x})$ to be used in this section was developed in section 4. 3.1 given by (4.4),

$$\hat{\mathbf{z}}_i = \begin{bmatrix} z_u(k+1) \\ z_v(k+1) \\ z_d(k+1) \end{bmatrix} = \frac{f}{x} \begin{bmatrix} y \\ z_{fi}(k+1) \\ B \end{bmatrix} \tag{4.23}$$

where,

$$x = (x_{fi}(k+1) - x_r(k+1))\cos(\phi(k)) + (y_{fi}(k+1) - y_r(k+1))\sin(\phi(k))$$
$$y = -(x_{fi}(k+1) - x_r(k+1))\sin(\phi(k)) + (y_{fi}(k+1) - y_r(k+1))\cos(\phi(k))$$

with the associated covariance,

$$\mathbf{R}_i = \begin{bmatrix} \sigma_u^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_d^2 \end{bmatrix} \tag{4.24}$$

### 4.4.3 Dealing with Constraints

The state vector is constrained through additional information such as the nonholonomic nature of robot motion and odometry data that are generally available in the context of SLAM.

The momentary control inputs $\mathbf{u}_i$ at time $i$ is related to the robot pose at time $i-1$ and $i$ by,

$$\mathbf{u}_i = \begin{bmatrix} u_{ri} \\ \omega_{ri} \end{bmatrix} = \begin{bmatrix} \sqrt{(x_{ri} - x_{ri-1})^2 + (y_{ri} - y_{ri-1})^2} / \Delta t_i \\ (\phi_{ri} - \phi_{ri-1}) / \Delta t_i \end{bmatrix} \tag{4.25}$$

The covariance matrix of odometry information $\mathbf{u}_i, i = 1,\ldots,n$ is given by,

$$\mathbf{U} = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix} \tag{4.26}$$

It is important to note that equation (4.25) is valid for any point mass and does not capture the complete motion of a robot with nonholonomic constraints. Thus the robot poses at $i$ and $i-1$ are also related by the equality,

$$c(\mathbf{x}) = \cos(\phi_{ri-1})(y_{ri} - y_{ri-1}) - \sin((\phi_{ri-1})(x_{ri} - x_{ri-1}) = 0 \qquad (4.27)$$

Equation (4.27) represents a deterministic observation characterised by the unit impulse. For numerical stability a narrow Gaussian is used ($\sigma_\delta \approx 1e-3$) to model the distribution of $c(\mathbf{x})$.

The complete relationship between the state vector $\mathbf{x}$ and the odometry measurements can now be written as,

$$\mathbf{u}_i = \begin{bmatrix} u_{ri} \\ \omega_{ri} \\ 0 \end{bmatrix} = \begin{bmatrix} \sqrt{(x_{ri} - x_{ri-1})^2 + (y_{ri} - y_{ri-1})^2} / \Delta t_i \\ (\phi_{ri} - \phi_{ri-1}) / \Delta t_i \\ \cos(\phi_{ri-1})(y_{ri} - y_{ri-1}) - \sin((\phi_{ri-1})(x_{ri} - x_{ri-1}) \end{bmatrix} \qquad (4.28)$$

with the new augmented covariance matrix,

$$\mathbf{U} = \begin{bmatrix} \sigma_v^2 & 0 & 0 \\ 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & \sigma_\delta^2 \end{bmatrix} \qquad (4.29)$$

Since the odometry measurements represents a Gaussian distribution,

$$p(\mathbf{u}_i \mid \mathbf{x}) = \eta e^{-\frac{1}{2}\mu_i^T \mathbf{U}^{-1} \mu_i} \qquad (4.30)$$

where $\boldsymbol{\mu}_i = \mathbf{u}_i - \mathbf{g}_i(\mathbf{x})$ and $\mathbf{g}_i(\mathbf{x})$ is the predicted odometry measurement.

### 4.4.4 SLAM as an Optimisation Problem

The formulation begins by noticing that the observation likelihood now contains both observations $(\mathbf{z})$ to features as well as odometry measurements $(\mathbf{u})$. However, these two types of measurements are conditionally independent. Therefore,

$$\begin{aligned} \mathbf{x}^* &\triangleq \underset{\mathbf{x}}{\operatorname{argmax}} \quad p(\mathbf{z} \mid \mathbf{x}) p(\mathbf{x}) \\ &= \underset{\mathbf{x}}{\operatorname{argmax}} \quad p(\mathbf{z}, \mathbf{u} \mid \mathbf{x}) p(\mathbf{x}) \end{aligned} \qquad (4.31)$$

The same conditional independence holds between odometry measurements and observations to features. Therefore, the constrained observation model is,

$$p(\mathbf{z}_i, \mathbf{u}_i \mid \mathbf{x}) = p(\mathbf{z}_i \mid \mathbf{x})\, p(\mathbf{u}_i \mid \mathbf{x}) \tag{4.32}$$

Where $\mathbf{x}$ contains both map and the entire set of poses of the robot. Substituting the observation likelihood functions from the previous section,

$$p(\mathbf{z}_i, \mathbf{u}_i \mid \mathbf{x}) = \eta e^{-\frac{1}{2}(\varepsilon_i^T \mathbf{R}_i^{-1} \varepsilon_i)}\, e^{-\frac{1}{2}(\mu_i^T \mathbf{U}^{-1} \mu_i)} \tag{4.33}$$

and assuming a uniform prior over the state variables, the cost function ($F(\mathbf{x})$) to minimise in order to solve the SLAM problem is derived by taking the negative of the log of the new likelihood function,

$$\begin{aligned} F(\mathbf{x}) &= \frac{1}{2}\sum_{i=1}^{Z} \varepsilon_i^T \mathbf{R}_i^{-1} \varepsilon_i + \frac{1}{2}\sum_{i=1}^{n} \mu_i^T \mathbf{U}_i^{-1} \mu_i \\ &= \frac{1}{2}(\varepsilon^T \mathbf{R}^{-1} \varepsilon + \mu^T \mathbf{U}^{-1} \mu) \end{aligned} \tag{4.34}$$

Then the new MAP estimate is,

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}(\varepsilon^T \mathbf{R}^{-1} \varepsilon + \mu^T \mathbf{U}^{-1} \mu) \tag{4.35}$$

Again the problem is solved through an iterative sequence,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k \tag{4.36}$$

In order to minimise the cost function, the algorithm starts at $k = 0$ with an initial estimate of the state vector $\hat{\mathbf{x}}_0$, and proceeds to calculate the approximations to the derivatives. The first derivative,

$$\nabla F(\mathbf{x}) = \mathbf{b} = -\mathbf{J}_h^T \mathbf{R}^{-1} \varepsilon - \mathbf{J}_g^T \mathbf{U}^{-1} \mu \tag{4.37}$$

and the Hessian matrix,

$$\nabla^2 F(\mathbf{x}) \approx \mathbf{A} = \mathbf{J}_h^T \mathbf{R}^{-1} \mathbf{J}_h + \mathbf{J}_g^T \mathbf{U}^{-1} \mathbf{J}_g \tag{4.38}$$

Then at the $k^{th}$ iteration,

$$\mathbf{A}_k \Delta \mathbf{x}_k = \mathbf{b}_k \tag{4.39}$$

and an improved estimate of the states are realised through (4.36). As shown in section 2.2, Levenberg-Marquardt iterations continue until an acceptable minimum is reached. A good indication of a minimum is when the rate of change in residual is minimal.

### 4.4.5 Initial Parameter Estimation for Batch Optimisation

The small residual assumption requires that an initial 'guess' close to the true solution of the state vector be available in order to avoid converging to an incorrect local minimum. In reality, finding a good initial guess is non trivial and simplistic techniques such as integrating odometry would not normally produce a sufficiently accurate starting point. Figure 4.7 shows simulation results when the batch optimisation algorithm was initialised with the robot odometry estimates. The map was initialised with true feature locations corrupted by small uniformly distributed pseudo-random numbers to minimise the effects due to error in initial guess of the map features. Figure shows that the batch estimates appear to be inconsistent.

As noted in section 4.3.3 that, even though EKF/UKF estimates tend to be inconsistent, the errors in pose estimates are generally small. Therefore, the EKF or UKF estimates of the states could potentially be used as a reasonable starting point for the batch optimisation method. In fact, it was realised that the EKF[1] estimates of the pose suffice as an initial guess in the Levenberg-Marquardt iterations.

[1] EKF was preferred over UKF for practical reasons, including the speed of execution which is slightly higher compared to UKF as noted in [19] by a constant factor.

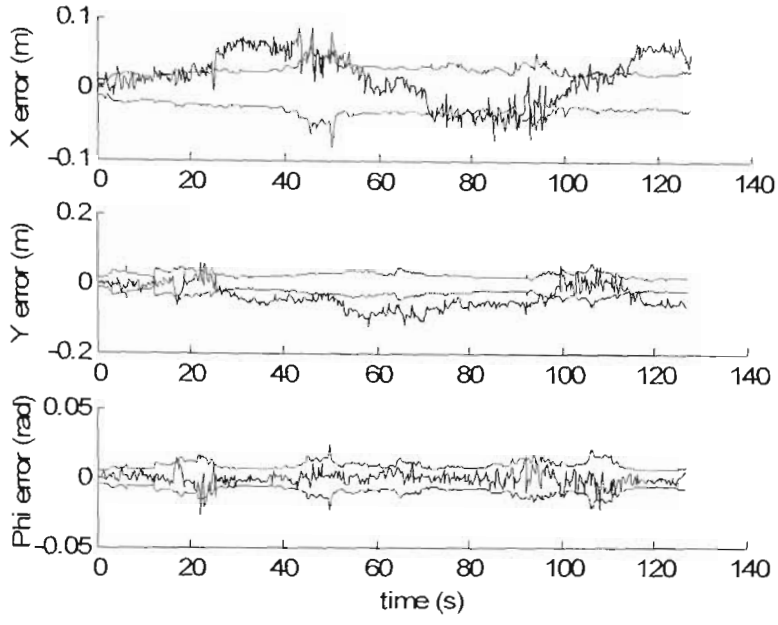



**Figure 4.7: Robot pose error estimates from the batch algorithm with $2\sigma$ confidence bounds where odometry data has been used for the initial guess. The results indicate that the batch estimates are inconsistent.**

Thus, if the complete history of the robot poses estimated by an EKF is,

$$\hat{\mathbf{x}}_r^{EKF} \triangleq \{\hat{\mathbf{x}}(t)\}; \quad t = 1,\ldots,N \tag{4.40}$$

Then the batch optimisation starts with an initial estimate of the states,

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} \hat{\mathbf{x}}_r^{EKF} \\ \hat{\mathbf{x}}_m^{EKF} \end{bmatrix} \tag{4.41}$$

where $\hat{\mathbf{x}}_m^{EKF}$ is the map estimate from the EKF. The simulations presented in the next section show that (4.41) is in fact a reasonable initial guess for the batch optimisation algorithm.

### 4.4.6 Simulation Results: Consistent State Estimation with Batch Optimisation

Having developed the necessary techniques for solving SLAM using batch optimisation, it is now possible to test the new algorithm for consistency using simulations. The simulations presented below uses the simulated data from the previous section which was shown to produce inconsistent state estimates when used in the recursive estimators discussed. Also, the EKF state estimates are used as the initial 'guess' in the Levenberg-Marquardt iterative process. Figure 4.8 shows the robot state estimate error results using the batch optimisation algorithm. As can be seen from the figure, the errors are well bounded within the 95% confidence interval.

Figure 4.9 shows the final map estimates of both the batch algorithm and the EKF used for the initialisation of the batch algorithm. The figure shows that the batch estimates of the map feature locations are consistent whereas the EKF based feature location estimates are inconsistent.

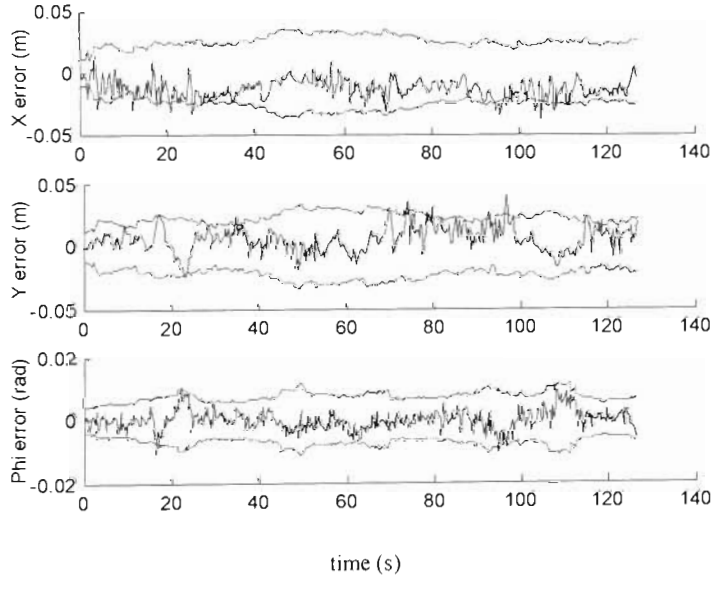
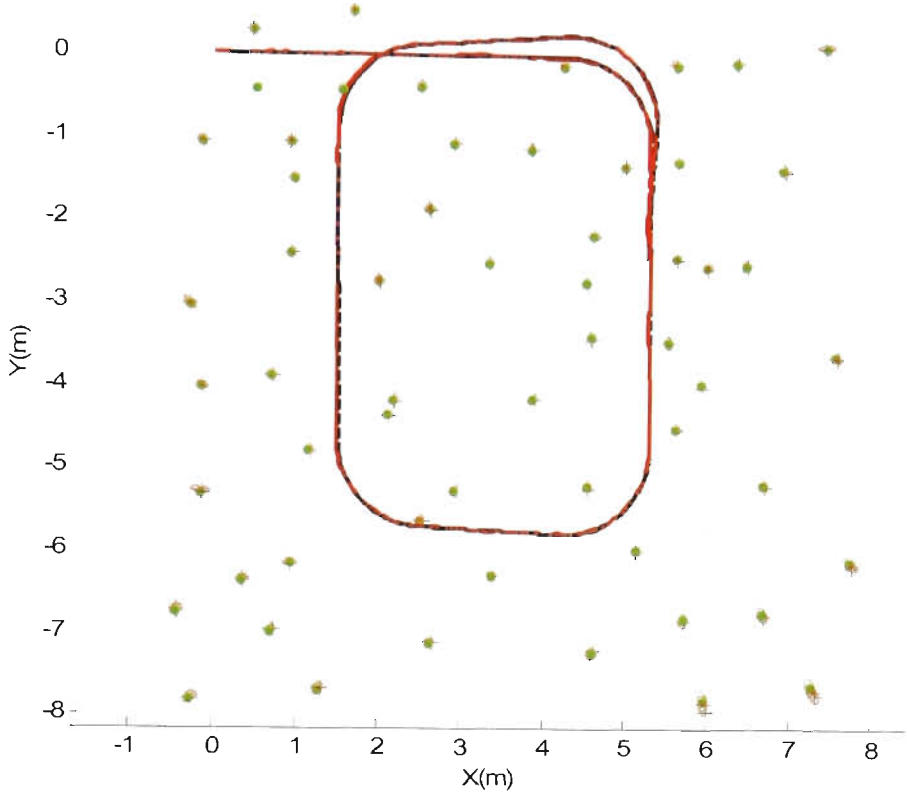
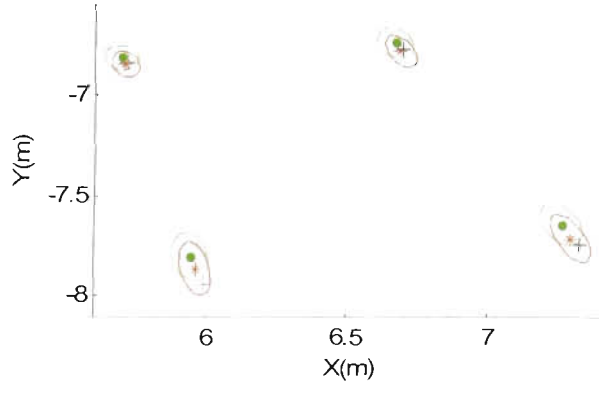



**Figure 4.8: Robot pose error estimate with $2\sigma$ confidence bounds from the batch algorithm.**




(a)




(b)

**Figure 4.9: The final map estimate. The batch estimates '*' and EKF feature estimates '.' and the true feature locations '+'. (a) Complete map with the batch estimated robot path (solid line) and the true robot trajectory (dashed line). (b) Enlarge portion of the map.**

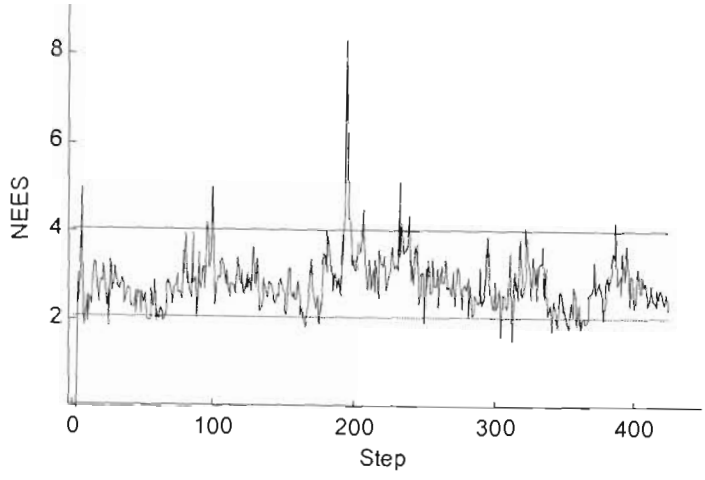



**Figure 4.10: Average NEES estimates from the batch algorithm**

Finally, figure 4.10 shows the average NEES estimates of the 50 simulation runs using the batch optimisation algorithm.

## 4.5 Discussion

The chapter has presented an insight into the nature of EKF SLAM as applied to a short baseline stereo vision sensor. The analysis began with an implementation of an EKF using the 'standard' stereo observation model. The average NEES results indicate that this combination of filter and observation model yields highly optimistic state estimates. Further it was shown that when the baseline of the camera is higher, the filter estimates become consistent. This particular result prompted the derivation of an alternative observation model that was shown in chapter 3 to be well approximated by Gaussian distribution. Even though the use of the new observation model in an EKF reduced the bias in the average NEES compared to the previous implementation, estimates were still inconsistent. The new observation model was then implemented in an Unscented Kalman Filter. The Unscented Transform is well known to capture the distribution statistics of a nonlinear transform. However, for the particular stereo vision sensor, UKF also failed to generate consistent state estimates. These results particularly insightful, in that, it highlight the fact that the stereo vision

sensor considered has a highly nonlinear observation model resulting in inconsistent state estimates when used in conventional recursive estimators. Especially noticeable is the fact that the effects of the nonlinearity of the observation model become apparent in relatively small loops of robot navigation.

This inconsistency in recursive filter estimates prompts the formulation of a new batch algorithm for SLAM that was presented later in the chapter. The new algorithm uses odometric and nonholonomic information to constrain the optimisation algorithm. It was also shown that the use of odometric information in the initial estimate of the batch solution tends to produce inconsistent state estimates, while use of the EKF results tend to be an adequate initial guess for a successful convergence with consistent estimates. The main drawbacks of the batch optimisation are the relatively higher computational cost compared to the recursive methods discussed previously and the requirement to delay the estimation process until all the observations corresponding to the entire robots trajectory are available to the estimator. Several practical issues including the above shortcomings are addressed in the next chapter through a novel Multi Map representation of the SLAM problem.

# Chapter 5

# Multi Map SLAM

## 5.1 Introduction

This chapter presents a new Multi Map (MM) SLAM framework for consistent mapping of large environments using visual sensing. The framework was inspired by the observations of human navigation patterns. Analogous to the human behaviour, where striking visual cues are used to localise within the global context, a set of map features are used in Multi Map framework for the localisation of the robot within the global coordinate frame. Features corresponding to this high level map rely on visually salient features that can be recognised using high dimensional descriptors that are scale and affine invariant. This map referred to as the Global Map (GM) is sparse and the features could only be observed at certain points in the robot trajectory. A second low level map is used for navigating the robot between the features of the high level map. The relatively dense Local Map (LM) is periodically marginalised from the state vector of the estimator in order to reduce the overall computational cost of the SLAM algorithm.

Section 5.2 summarises the key features of the Multi Map framework. Section 5.3 extends the batch optimisation algorithm developed in Chapter 4 to the MM

framework. Section 5.4 describes the estimation process using the generalised Multi Map framework. Developments in this section relies on the Variable State Dimension Filter [12]. Section 5.4 summarises the work presented in this chapter.

## 5.2 Multi Map Approach

Formally, the proposed method consists of two independent sets of maps of which, one aids local navigation between Points of Interests (PoI) called the Local Map (LM) while a second, a Global Map (GM) is used to describe such points of interest that could potentially aid loop closure. The GM consists of clusters of 3D features that are view invariant and are described by high dimensional descriptors. The descriptive nature of these features provides the necessary data association information. Once a PoI is reached, a batch optimisation is executed combining both the GM and the LM corresponding to the current segment of the robot path bounded by the two latest PoIs. The current LM is then marginalised from the estimation process.

The algorithm is developed in two stages. In the first stage, the batch optimisation technique developed in Chapter 4 is extended to accommodate the two independent maps in the estimation process. In the second stage, algorithm is generalised for the case of arbitrary navigation realising the full potential of the MM approach.

### 5.2.1 Point of Interest (PoI)

A Point of Interest could either be a real navigation start/end point for a given navigation task or virtual points selected along the navigation path. An example for the former would be the start and end points, when a robot needs to traverse between two adjacent rooms where the navigation path is reasonably simplistic. If the task is more involved and the robot is expected to travel an extended period of time then virtual PoIs need to be defined along the trajectory. Scheduling of PoI in the latter case is problem dependent and requires an insight into the environment in which the robot operates. The primary requirement of a PoI is to provide with strategic

locations where a loop closure could be anticipated. In the simplest case when a robot is known to return to its origin after a short run as depicted in Figure 5.1 the origin becomes the only PoI. For a more complicated navigation task such as the rectangular path in the simulations presented in the previous chapter a PoI could be scheduled after each right angle turn.

A secondary task of a PoI is to act as a switching point that switches from EKF to the batch optimisation in order to improve the consistency of the estimates. This necessitates PoIs to be scheduled at regular intervals during the trajectory even if a loop closure is not anticipated.

### 5.2.2 Local Map (LM)

Key requirement of LM features are that they can be efficiently tracked between consecutive robot poses without incurring heavy computational burden on tracking or initialisation. Since LM features are not required to aid in loop closure, marginalising LM features from the map prior to subsequent batch optimisations does not result in loss of information. This is one of the key advantages the MM framework has. KLT based features are used as LM features in the experimental work in this thesis.

### 5.2.3 Global Map (GM)

Once a PoI is reached, a set of view invariant features are selected to describe the PoI independent from the LM features. As mentioned previously the requirement is to select a set of features that are considerably invariant to view point changes and uniquely described by high dimensional descriptors thereby enabling them to be viewed and associated within a large view and scale change.

There is a wide selection of such feature detector-descriptor combinations among which SIFT has gained much consideration [33]. A comparison of available state of the art descriptors could be found in [70, 89] along with a comparison of feature detectors in [77]. However in this work a more efficient implementation called the

Speed Up Robust Features (SURF) [38] is used. A detailed description of practical issues pertaining to feature selection is discussed in the next chapter. For the development of Multi Map framework this chapter considers that a set of independent features are observed in the vicinity of each PoI and are globally associated (data association is known even when these features are observed after an intermediate absence). These features form the Global Map.

## 5.3 Batch Optimisation with Multiple Maps

Figure 5.1 depicts a simple navigation scenario. The robot starts at the origin and completes a small loop by returning to the origin after a temporal lapse. The problem is similar to the simulations discussed previously. However, the approach to the problem is now from a practical perspective. Recall that in Chapter 3, the vision sensor is only capable of tracking features between consecutive images. Therefore, detecting a loop closure using KLT features is not possible. However these features provide valuable information about the motion of the robot along the path. Thus, these features form the Local Map. In the human navigation analogy presented in Chapter 1, these features emulate the task of lane markings, etc.

In order to realise loop closure a sparse set of unique features are used. The requirement is to select a set of features that are considerably invariant to view changes and uniquely describable by high dimensional descriptors thereby enabling them to be viewed and associated within a large view variance. In the analogy of human navigation, these features represent key visual cues. In the example, these features are marked as 'GM-node 1'.

A unique PoI is at the origin of the reference frame in this particular example. Once the robot completes the journey, it sees the GM features again and realises the loop closure. The algorithm that extends the batch optimisation to accommodate the two maps is described next.

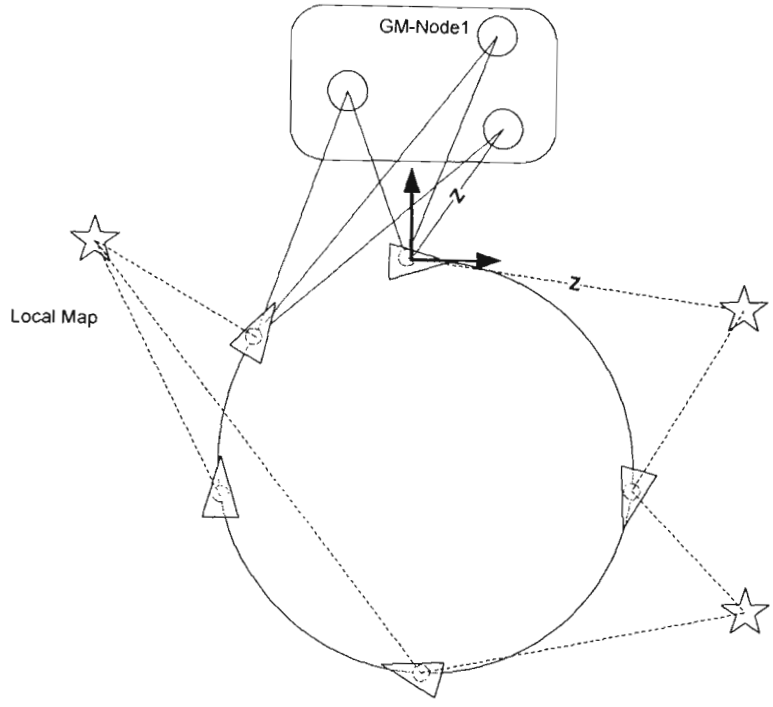



**Figure 5.1: Simplest form of the MM approach with a single node of Global Map. GM-Node1 provides Loop Closure information.**

### 5.3.1 Multi Map Filter (MMF)

The first extension to the solution presented in Chapter 4 occurs when the state vector is extended to accommodate the GM features,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_{LM} \\ \mathbf{x}_{GM} \end{bmatrix} \tag{5.1}$$

As opposed to the state vector in (4.20), the new augmented state vector contains two independent maps. The LM ( $\mathbf{x}_{LM}$ ) contributes to the frame-to-frame optimisation whilst the GM ( $\mathbf{x}_{GM}$ ) contributes to the loop closure.

The initialisation of batch optimisation is provided by an EKF,

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} \hat{\mathbf{x}}_r^{EKF} \\ \hat{\mathbf{x}}_{LM}^{EKF} \\ \hat{\mathbf{x}}_{GM,0} \end{bmatrix} \tag{5.2}$$

Since the Global Map features need to be integrated in the optimisation algorithm an initial estimate of the GM states ( $\hat{\mathbf{x}}_{GM,0}$ ) is required. Given that the EKF already has an improved robot pose estimate, a straight forward strategy is to use the GM feature observations with the EKF robot pose estimates ( $\hat{\mathbf{x}}_r^{EKF}$ ) to initialise the map. Once the state and the initial estimates are defined the minimisation could proceed as discussed in the previous chapter.

### 5.3.2 An Example

To illustrate the Multi Map filter presented thus far, it is apt to reconsider the simulated example presented in previous chapter. Figure 5.2 depicts the LM and the robot trajectory. During this simulation features are tracked only when they are consecutively visible. Thus when a previously seen feature appears again after an intermediate lapse of visibility, it is initialised as a new feature. This can be seen in the figure as clutters of points congregated around locations where single features used to be in previous simulations. Essentially this debilitates the detection of loop closure. For instance, in Figure 5.4 (left) plots the robot errors and the 95% confidence bounds for the EKF that is used in the initialisation of the MMF. What is noticeable now apart from the inconsistency is the lack of collapse in the pose covariance estimates as seen in the case of filtering with loop closure.

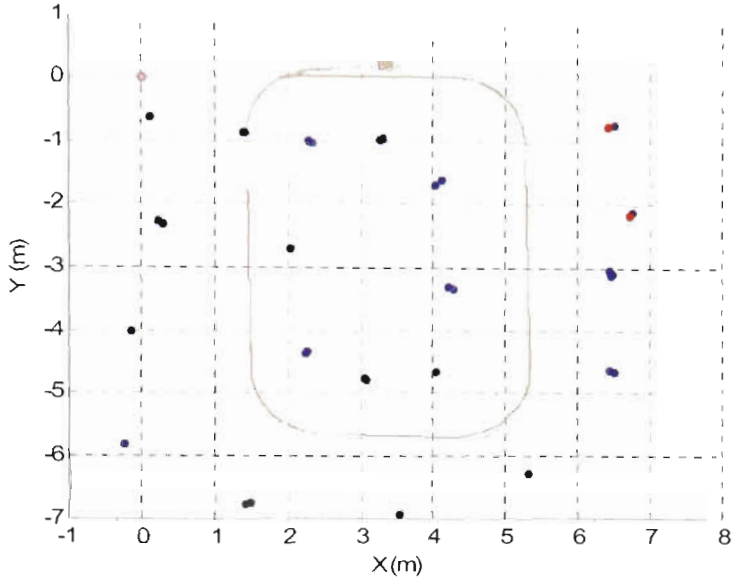



**Figure 5.2: The Local Map (LM) and the robot path. Multiple instances of the same feature could be seen initialised due to lack of loop closure information in the LM features.**

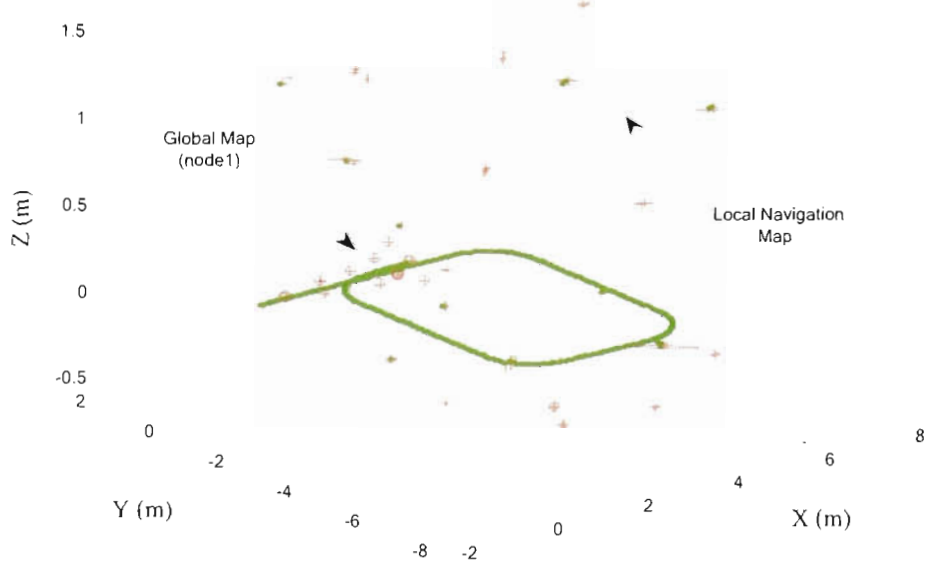



**Figure 5.3: Multi Map Filter results showing the LM and the GM.**

In order to simulate the invariant features, a second set of features are generated closer to the origin as depicted in Figure 5.3 marked GM-node 1. During this simulation robot can observe these features at the beginning of its traverse and again at the end of its navigation. Data association between the two sets of observations are now assumed to be known in order to simulate the descriptor based data association.

As the robot reaches the end of its journey the Multi Map filter is invoked. Figure 5.4 (right) shows the results from the MMF (full batch optimisation) with both maps integrated. This clearly shows the reduction in pose uncertainty as would be expected by the information gain due to the loop closure. Further, the filter now shows consistent results

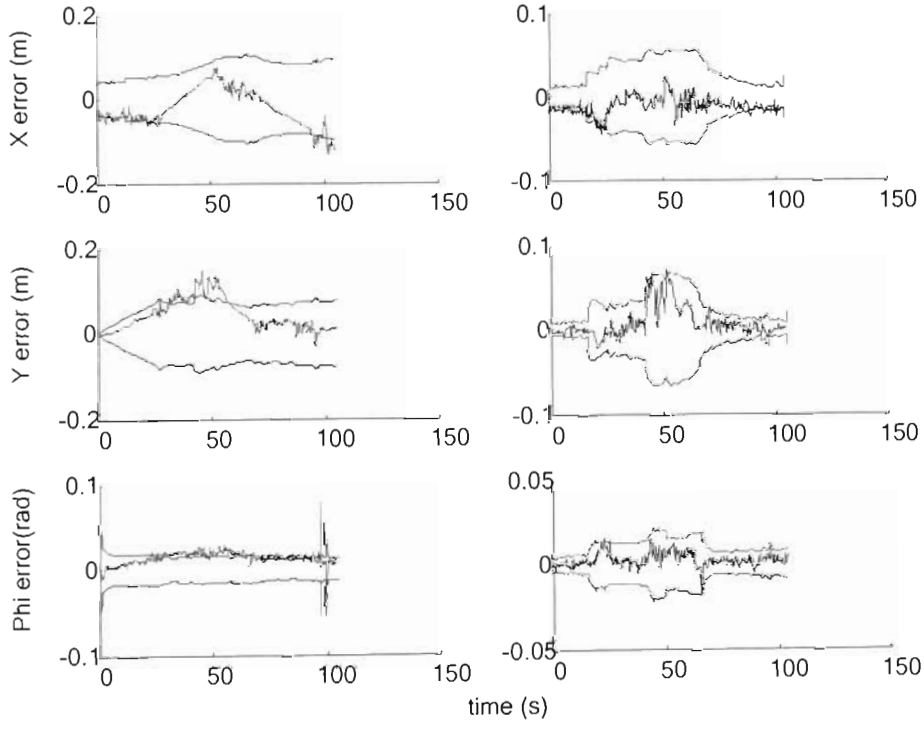



**Figure 5.4 MMF robot pose estimate errors with 95% confidence bounds. Results from the EKF (left) are used in the initialisation of the MMF (right). Note the difference in *y-axis* scales.**

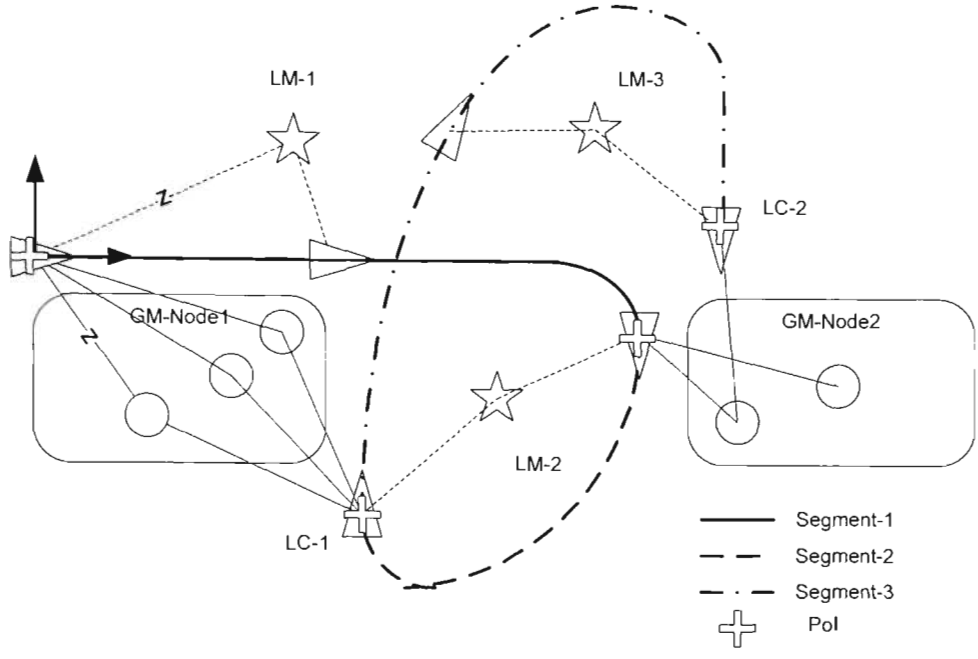



**Figure 5.5: A simplified illustration of the Generalised Multi Map approach. The robot starting from the origin takes a slightly complicated route. Apart from the origin, PoIs are created when the robot makes a 'sharp' turn. The first Loop Closure (LC-1) occurs when the features observed at the first PoI are observed again from the third PoI. A second Loop Closure (LC-2) occurs when some of the features observed at second PoI are re-observed from 4$^{th}$ PoI.**

## 5.4 Generalised Multi Map Smoothing (MMS)

In this section the MM framework is extended from single loop navigation to the generalised arbitrary navigation. Several key innovations are presented, including techniques based on the Variable State Dimension Filter (VSDF) [12, 90, 91] algorithm to improve the efficiency and consistency of the MMF.

Figure 5.5 shows a simplified navigation scenario which could be used to appreciate the Multi Map Smoothing (MMS) algorithm. The first innovation stems from the regular placement of PoIs along the navigation path. To illustrate this further, consider the simulation environment shown in Figure 5.6. The robot path is similar to the example in Figure 5.5. Assume that the MMF is to be used in this simulation (i.e. single PoI at the origin). Since the robot is not returning to its origin, the GM-node1 features cannot be used in closing this larger loop. Effects of this could be illustrated

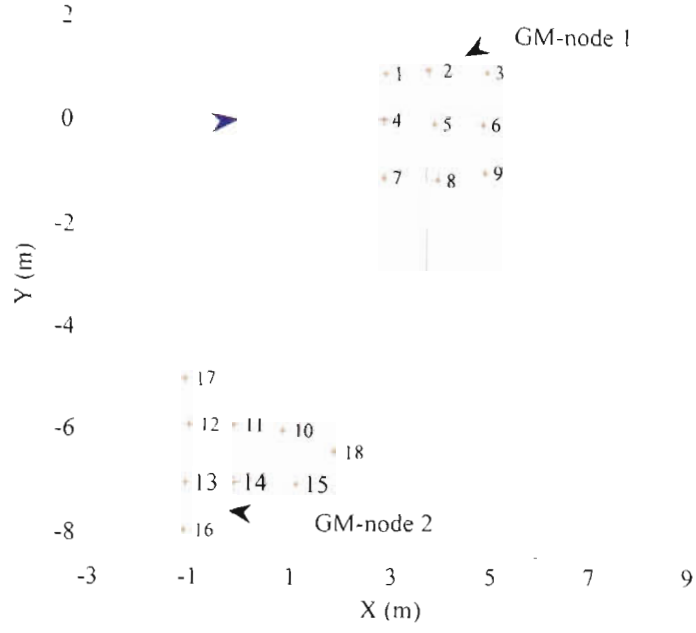



**Figure 5.6: Feature layout and robot trajectory for the Multi Map Smoothing simulation. 'o'- represents the LM features while '*'- are the GM features.**

by executing a MMF in this simulation environment. Figure 5.7 shows pose estimate errors from both the EKF used in the initialisation as well as the MMF. It is clearly apparent from the figure that a loop closure is not realised. However, what is more important is the inconsistency in the estimates of the MMF. This in fact is due to the growing error in the EKF that is used to initialise the MMF. This fact was verified by initialising the MMF with state estimates that are generated from the true values corrupted by a Gaussian distribution with 0.05m standard deviation, which produces consistent MMF estimates.

Thus, PoIs are required at regular intervals of the robot path,

- To maintain the consistency of the MMF estimates by executing the MMF at shorter intervals. Thereby increasing the accuracy of the initial estimates of the consecutive optimisation cycles
- To anticipate loop closures at locations other than the origin.

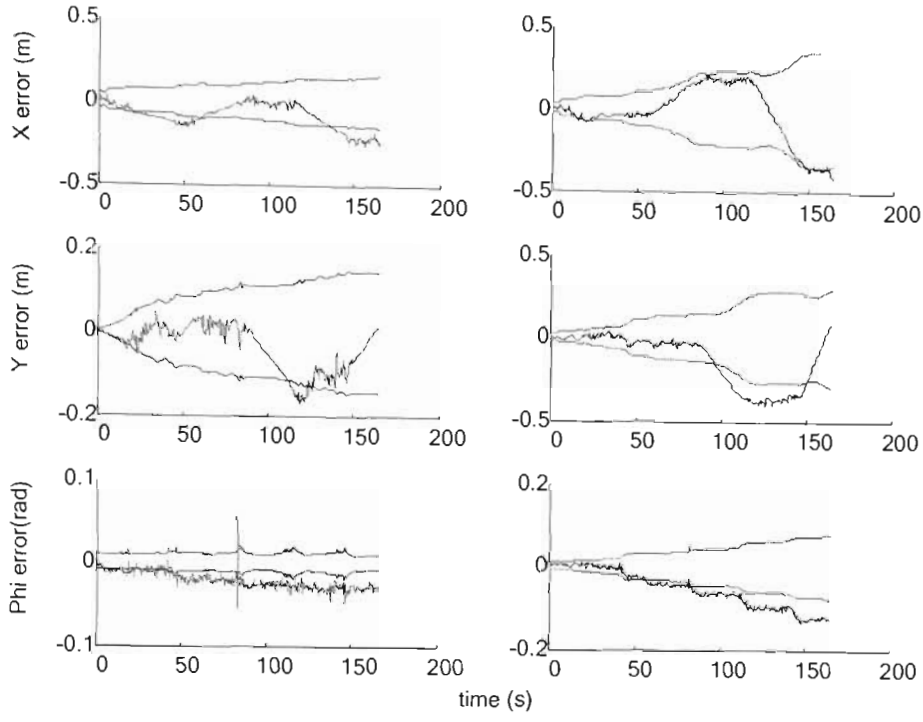



**Figure 5.7: The robot pose error estimate for the entire loop in Figure 5.6 without the loop closure along with the $2\sigma$ bounds. Left chart shows the results from the EKF. In the right are the results from the MMF algorithm starting with the initial state estimates provided by the EKF.**

The second innovation improves the efficiency of the MMF algorithm by marginalising LM features at appropriate intervals. Techniques based on the VSDF algorithm are used to maintain the consistency of the estimates while features are marginalised.

### 5.4.1 The Variable State Dimension Filter

The Variable State Dimension Filter (VSDF) [12, 90, 91] is similar to the EKF, however the VSDF formulation also takes the advantages of the batch optimisation techniques as well. The technique allows for solving the general nonlinear optimisation problem involving a global state and a variable number of local states. The formulation is ideally suited to the MM framework in that the GM represents the

global state of the problem whilst LM represents the variable local states which allows for marginalisation as the robot evolves in the environment. The marginalisation process will be discussed later in the chapter. Once a state is removed from the state vector, related observations also need to be removed from the optimisation problem. This is also dealt within the VSDF framework.

Consider the general batch optimisation algorithm, the cost function to minimise is,

$$\mathbf{F}(\mathbf{x}) = \sum_{i=1}^{Z} \boldsymbol{\varepsilon}_i^T \mathbf{R}_i^{-1} \boldsymbol{\varepsilon}_i \tag{5.3}$$

Then the batch optimisation proceeds by calculating the associated derivatives,

$$\begin{aligned} \mathbf{A} &= \sum_{i=1}^{Z} \mathbf{J}_i^T \mathbf{R}_i^{-1} \mathbf{J}_i \\ \mathbf{b} &= \sum_{i=1}^{Z} \mathbf{J}_i^T \mathbf{R}_i^{-1} \boldsymbol{\varepsilon}_i \end{aligned} \tag{5.4}$$

where $\boldsymbol{\varepsilon}_i = \mathbf{z}_i - \mathbf{h}_i(\mathbf{x})$ and iteratively solving,

$$\begin{aligned} \mathbf{A}\Delta\mathbf{x} &= -\mathbf{b} \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \Delta\mathbf{x} \end{aligned} \tag{5.5}$$

starting with an initial estimate of the states $\hat{\mathbf{x}}_0$.

Assume that the first observation ($i=1$) is to be removed from the cost function. Then it is required to assess its contribution to the derivatives and thus to the cost function. This amounts to approximating the contribution from the observation at the current state estimate by linearising $\mathbf{h}_1(\bullet)$,

$$\mathbf{h}_1(\mathbf{x}) \approx \mathbf{h}_1(\mathbf{x}_0) + \mathbf{J}_1(\mathbf{x} - \mathbf{x}_0) \tag{5.6}$$

and incorporating it in a Gaussian prior $(\mathbf{x} - \hat{\mathbf{x}}_0)\mathbf{A}_0(\mathbf{x} - \hat{\mathbf{x}}_0)^T$ such that the new cost function contains both linearized measurements and nonlinear measurements [43].

$$\mathbf{F}(\mathbf{x}) = (\mathbf{x}-\mathbf{x}_0)\mathbf{A}_1(\mathbf{x}-\mathbf{x}_0)^T + \sum_{i=2}^{Z} \boldsymbol{\varepsilon}_i^T \mathbf{R}_i^{-1} \boldsymbol{\varepsilon}_i \tag{5.7}$$

where,

$$\begin{aligned} (\mathbf{x}-\hat{\mathbf{x}}_0)\mathbf{A}_1(\mathbf{x}-\hat{\mathbf{x}}_0)^T &\approx \boldsymbol{\varepsilon}_1^T \mathbf{R}_1^{-1} \boldsymbol{\varepsilon}_1 \\ &\approx (\mathbf{z}_1 - \mathbf{J}_1(\mathbf{x}-\hat{\mathbf{x}}_0))^T \mathbf{R}_1^{-1} (\mathbf{z}_1 - \mathbf{J}_1(\mathbf{x}-\hat{\mathbf{x}}_0)) \end{aligned} \tag{5.8}$$

The contribution from the first observation to the Hessian ( $\mathbf{A}_1$ ) and the Jacobian ( $\mathbf{b}_1$ ) are,

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{J}_1^T \mathbf{R}_1^{-1} \mathbf{J}_1 \\ \mathbf{b}_1 &= \mathbf{J}_1^T \mathbf{R}_1^{-1} (\mathbf{z}_1 - \mathbf{h}_1(\mathbf{x}_0)) \end{aligned} \tag{5.9}$$

Then the new Hessian and the Jacobian of the new cost function are,

$$\begin{aligned} \mathbf{A} &\approx \mathbf{A}_1 + \sum_{i=2}^{Z} \mathbf{J}_i^T \mathbf{R}_i^{-1} \mathbf{J}_i \\ \mathbf{b} &\approx \mathbf{b}_1 + \mathbf{A}_1(\mathbf{x}-\hat{\mathbf{x}}_0) + \sum_{i=2}^{Z} \mathbf{J}_i^T \mathbf{R}_i^{-1} \boldsymbol{\varepsilon}_i \end{aligned} \tag{5.10}$$

This technique could be applied to an arbitrary number of observation removals, and in an iteration of the Levenberg-Marquardt minimisation, the general form of the cost function, its Hessian and the Jacobian could thus be written as,

$$\begin{aligned} \mathbf{F}(\mathbf{x}) &= (\mathbf{x}-\mathbf{x}_0)\mathbf{A}_0(\mathbf{x}-\mathbf{x}_0)^T + \sum \boldsymbol{\varepsilon}_i^T \mathbf{R}_i^{-1} \boldsymbol{\varepsilon}_i \\ \mathbf{A} &= \mathbf{A}_0 + \sum \mathbf{J}_i^T \mathbf{R}_i^{-1} \mathbf{J}_i \\ \mathbf{b} &= \mathbf{b}_0 + \mathbf{A}_0(\mathbf{x}-\hat{\mathbf{x}}_0) + \sum \mathbf{J}_i^T \mathbf{R}_i^{-1} \boldsymbol{\varepsilon}_i \end{aligned} \tag{5.11}$$

$\mathbf{A}_0$ and $\mathbf{b}_0$ captures the contribution of the linearized observations.

### 5.4.2 The Estimation Process

The algorithm could be outlined as follows (*with reference to Figure 5.5*).

1. The first PoI is created at the origin where the robot initiates its navigation as in the case of the MMF.

- GM-Node 1 in Figure 5.5

2. Following PoIs are scheduled heuristically considering the anticipated navigation pattern and knowledge of the environment in which the robot operates. As described previously PoIs are marked with a set of view invariant features described by high dimensional descriptors (*GM-node 1,2*) enabling detection of a Loop Closure.

- For instance in Figure 5.5 PoIs are placed after each turn of the robot.

3. Once a PoI is reached, algorithm sets out to execute the MMF utilising the LM and GM information currently available. This is similar to the previously described MMF.

- In the example given in Figure 5.5, at LC-1, the MMF has GM-Node 1, 2 and LM-2 features in the augmented map state vector.

4. VSDF techniques are used to marginalise the features from the state vector that do not contribute to the current execution of the MMF. The Variable State Dimension Filter [12] provides a means to preserve the 'contributions' to the estimator from the observations made to these marginalised features by linearising these measurements and incorporating them into a Gaussian prior within the batch optimisation paradigm.

- In the previous example, LM features belonging to LM-1 needs to be marginalised from the state vector prior to the execution of MMF at LC-1 in order to estimate the states corresponding to the segment-2.

5. This sequence is recursively applied at each PoI.

The MMS is essentially a recursive smoother with dynamically varying state dimension. The Filter could be summarised as consisting of following components.

Assuming that the robot is at the $p^{th}$ PoI, ($p>2$) the current LM contains only a partial set of the LM features that are observed from the PoI at $p$-$1$. Thus the new cost function to minimise contains a prior reflecting the linearized control input data as well as the nonlinear measurement data,

$$F(\mathbf{x}) = (\hat{\mathbf{x}}_0 - \mathbf{x})^T \mathbf{A}_0 (\hat{\mathbf{x}}_0 - \mathbf{x}) + (\mathbf{d} - f(\mathbf{x}))^T \mathbf{Q}^{-1} (\mathbf{d} - f(\mathbf{x})) + (\mathbf{z} - h(\mathbf{x}))^T \mathbf{R}^{-1} (\mathbf{z} - h(\mathbf{x})) \quad (5.12)$$

Also it now requires augmenting the state vector with the new LM features, its corresponding observations, robot poses encompassing the current LM and any newly initialised GM features. Thus the new state change from the previous PoI,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_r(p-1) \\ \mathbf{x}_{LM}(p-1) \\ \mathbf{x}_{GM}(p-1) \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_r(p) \\ \mathbf{x}_{LM}(p) \\ \mathbf{x}_{GM}(p-1) \\ \mathbf{x}_{GM}(p) \end{bmatrix} \quad (5.13)$$

Where ($\mathbf{x}_r(p)$) represents the limited pose history attributed to the $p^{th}$ LM and ($\mathbf{x}_{LM}(p)$) contains the LM features belonging to the $p^{th}$ LM. $\mathbf{x}_{GM}(p-1)$ contains GM features that are already initialised during previous PoIs beginning from the origin and ($\mathbf{x}_{GM}(p)$) are the newly observed GM features yet to be estimated. Thus it is clear that the only component of this dynamic state vector that contributes to monotonic increase in the dimension is the observation of new GM features. Rest of the state changes with each PoI but does not contribute to continuous increase of the overall dimensionality of the state vector.

Having defined the state vector and the cost function, Levenberg-Marquardt optimisation could be performed by defining the derivatives with respect to the current state vector,

$$\begin{aligned} \mathbf{A} &= \mathbf{A}_0 + \mathbf{J}_h^T \mathbf{R}^{-1} \mathbf{J}_h + \mathbf{J}_f^T \mathbf{Q}^{-1} \mathbf{J}_f \\ \mathbf{b} &= \mathbf{b}_0 + \mathbf{A}_0 (\hat{\mathbf{x}}_0 - \hat{\mathbf{x}}^-) + \mathbf{J}_h^T \mathbf{R}^{-1} \boldsymbol{\varepsilon}_x + \mathbf{J}_f^T \mathbf{Q}^{-1} \boldsymbol{\varepsilon}_y \end{aligned} \quad (5.14)$$

Once optimisation has converged LM features belonging to the current segment of the robot path are marginalised, corresponding observations, robot poses and any GM observations related to the removed robot poses are also removed. This smoothing cycle is applied recursively at each new PoI. Thus the consistency of the state estimate is preserved at regular intervals of the robots trajectory, while maintaining an efficient algorithm for arbitrary navigation.

### 5.4.3 Marginalising LM from the MM

The first requirement in the MMS approach is to remove the LM features from the state once they are used to improve the pose estimates of the relevant potion of the navigation. Consider the state vector in (5.1) which is the starting point for the Multi Map optimisation at a given ($p^{th}$) PoI,

$$\mathbf{x}^{MMF} = \begin{bmatrix} \mathbf{x}_{r,p} \\ \mathbf{x}_m^{MMF} \end{bmatrix} \tag{5.15}$$

where $\mathbf{x}_m^{MMF} = [\mathbf{x}_{LM}, \mathbf{x}_{GM}]^T$ is the combined multi map at the current PoI.

It is required to remove ($\mathbf{x}_{LM}$) from ($\mathbf{x}_m^{MMF}$). This could be achieved by straight forward removal of the corresponding map elements from the overall state vector. Thus,

$$\mathbf{x}^{MMF} = \begin{bmatrix} \mathbf{x}_{r,p} \\ \mathbf{x}_{GM} \end{bmatrix} \tag{5.16}$$

However the optimisation algorithm now contains a prior inverse covariance term which should also be marginalised accordingly. Assume that $\mathbf{A}_0$ is reordered to reflect the states to be removed ($\mathbf{A}_{LM,*}$) and retained ($\mathbf{A}_{R,*}$) such that,

$$\mathbf{A}_0 = \begin{bmatrix} \mathbf{A}_{LM,LM} & \mathbf{A}_{LM,R} \\ \mathbf{A}_{LM,R}^T & \mathbf{A}_{R,R} \end{bmatrix} \tag{5.17}$$

Then using the matrix inversion lemma, the new prior after marginalisation of LM could be calculated as,

$$\mathbf{A}_0 \leftarrow \mathbf{A}_{R,R} - \mathbf{A}_{LM,R}^T \mathbf{A}_{LM,LM}^{-1} \mathbf{A}_{LM,R} \tag{5.18}$$

This technique lends itself to the removal of poses prior to the current PoI as well.

### 5.4.4 Removing Observations Associated with Marginalised Features

As the MM approach extends beyond the first loop closure it becomes necessary to remove observations corresponding to the previous LM features, since they are marginalised as described in the previous section. As a consequence the pose estimates belonging to these parts of the journey are no longer as accurate as would be when improved local navigation information was available through the LM. The technique described in section 5.4.1 could be used to linearise such observations. The technique also lends itself to be used in linearising any odometry measurements corresponding to marginalised robot poses in the state vector.

### 5.4.5 A Note on Information Loss

The unique character of the LM features results in the marginalisation exclusive of information loss. This could be illustrated from an Information Filter perspective [14] noting that the Hessian $\mathbf{A}_0$ as the current information matrix. Then each new observation 'adds' information to the matrix,

$$\mathbf{A} = \mathbf{A}_0 + \mathbf{J}_i^T \mathbf{R}_i^{-1} \mathbf{J}_i \tag{5.19}$$

In a general SLAM scenario, if a feature is marginalised, the potential information gain from a future observation to this feature ($\mathbf{J}_i^T \mathbf{R}_i^{-1} \mathbf{J}_i$) is lost, contributing to information loss. However, in the MMS framework, LM features are inherently incapable of long term data association. Therefore, even if the LM features were retained in the state vector, they do not contribute to future information gains owing

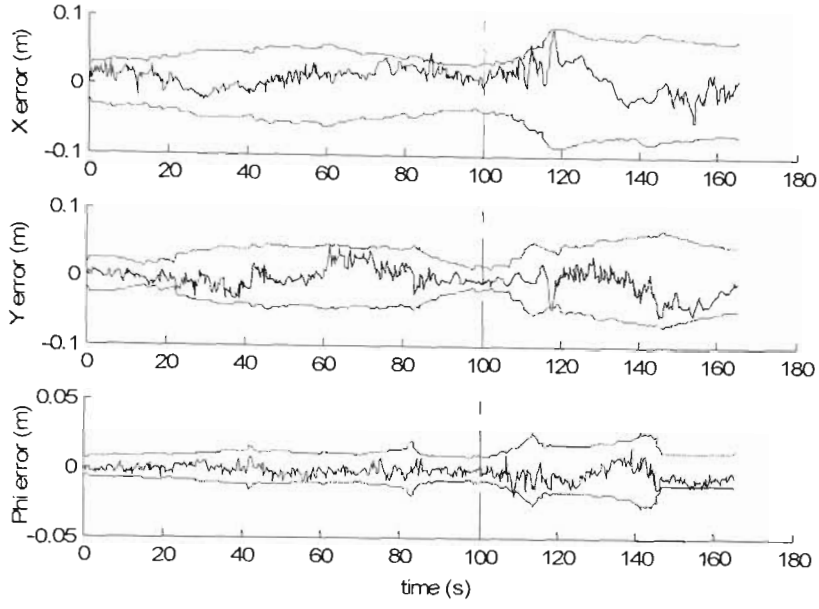
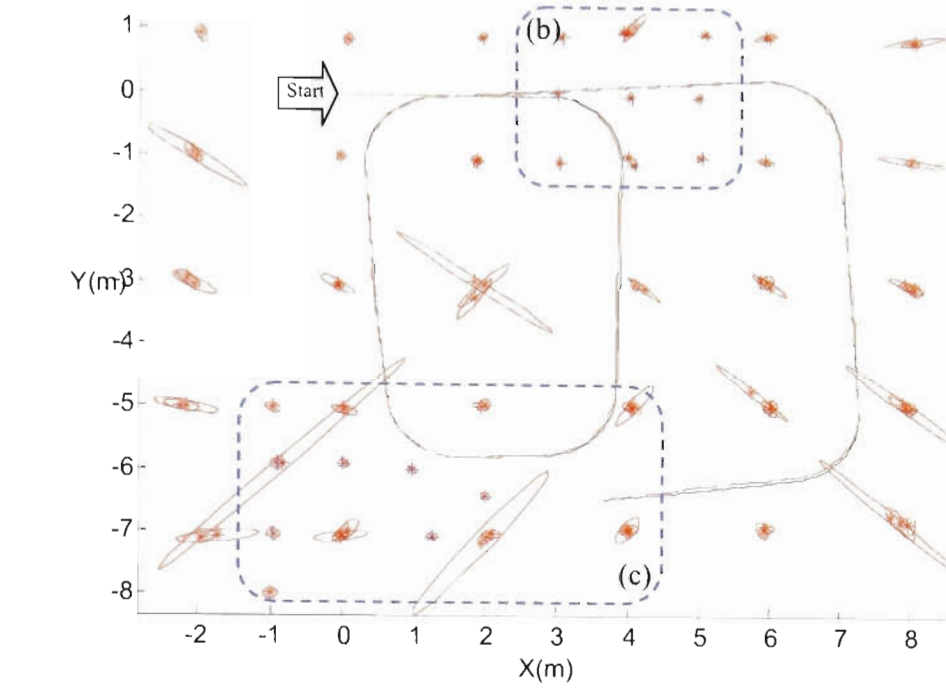



**Figure 5.8: Robot pose estimates from the MM Smoothing. The dashed line indicates the first loop closure corresponding to the first node of the GM.**

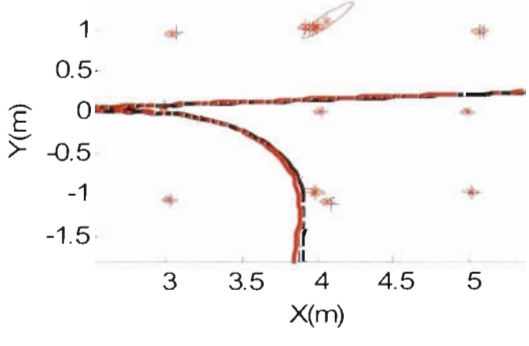
to the inability of re-observing these features. Conversely, marginalisation of these LM features therefore does not contribute to any information loss.

### 5.4.6 Simulation Results

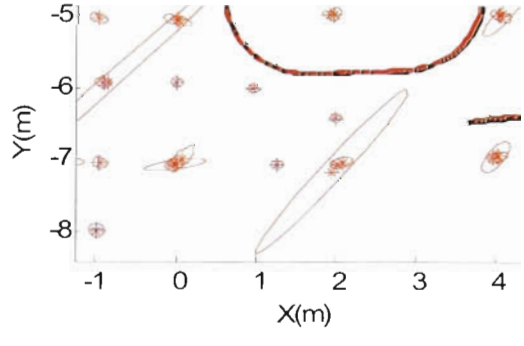
Figure 5.8 shows the complete pose estimate error using the MMS approach in the environment shown in Figure 5.6. It clearly shows the multiple loop closures occurring at the end of the small loop and at the end of large loop. The dashed lines indicate where the first PoI is established. The pose estimates are now consistent. Figure 5.9 shows the map estimates from the MMS algorithm. There are two nodes of the GM of which the first node initialised at the beginning of the robots trajectory aids the small loop closure and the second node first initialised during the third leg of the smaller loop aids the final loop closure. Finally in Figure 5.10, the average NEES estimates for 50 trials of the simulation experiment are presented. The estimates are within the $2\sigma$ confidence bounds.




(a)




(b)




(c)

**Figure 5.9: Map estimate from the MMS. (a) The complete map comprising of the two nodes of GM ('+') and the LM ('*'). (b) Close up view near the first node of the GM. '+' indicates the true locations of the GM features. (c) Close up view near the second node of the GM which aids the final loop closure.**

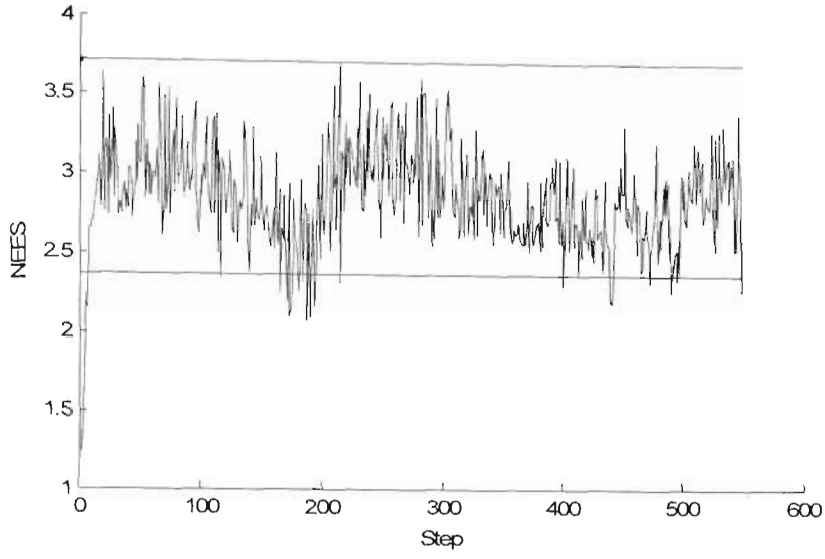



**Figure 5.10: Average NEES results for MMS**

## 5.5 Summary

This chapter presented the Multi Map approach for SLAM. The MM approach was developed to specifically accommodate a set of features that serves the purpose of arbitrary data association (as opposed to successive data association - tracking). The novel approach constitutes of two maps that are momentarily correlated to produce a consistent estimate of the robot path and map features. The GM principally aids the loop closure and is the only map that grows in dimensionality with time. The LM aids the navigation between points of interest (PoI) in the robots trajectory. These features are marginalised from the state vector at regular intervals corresponding to the PoIs. It was shown that the particular nature of these LM features results in a 'loss less' marginalisation of them from an information filter perspective. Thus the MM approach has the potential to be deployed in mapping of large environments. Finally, simulation results were presented with consistent results. The next chapter presents experimental results of the various algorithms developed thus far with particular insights relating to practical implementation issues.

# Chapter 6

# Experimental Results

Robotics is essentially a strand of applied science. Therefore, any algorithm put forth needs to be validated in a relevant practical context. This thesis focuses on the problem of Simultaneous Localisation and Mapping using a short baseline stereo vision system as the primary sensor, in an indoor setting. Some characteristics pertaining to the vision sensor has been studied in chapter 3 through experimental evaluations. Based on these evaluations, novel techniques to improve the consistency and performance of the SLAM algorithm have been developed in Chapter 4 and 5. This chapter presents representative real-world experimental results to illustrate the practical effectiveness of these solutions.

Section 6.1 discuss several issues related to the experimental environment and provides implementation details. Section 6.2 presents some of the initial results using the EKF algorithm in order to illustrate practical aspects of SLAM implementations with stereo vision. A novel algorithm for validating stereo vision data is also presented. Section 6.3 shows results using the batch optimisation algorithm with manual data association. Section 6.4 shows results from the implementation of MMS algorithm in a large environment. Section 6.5 concludes the chapter.

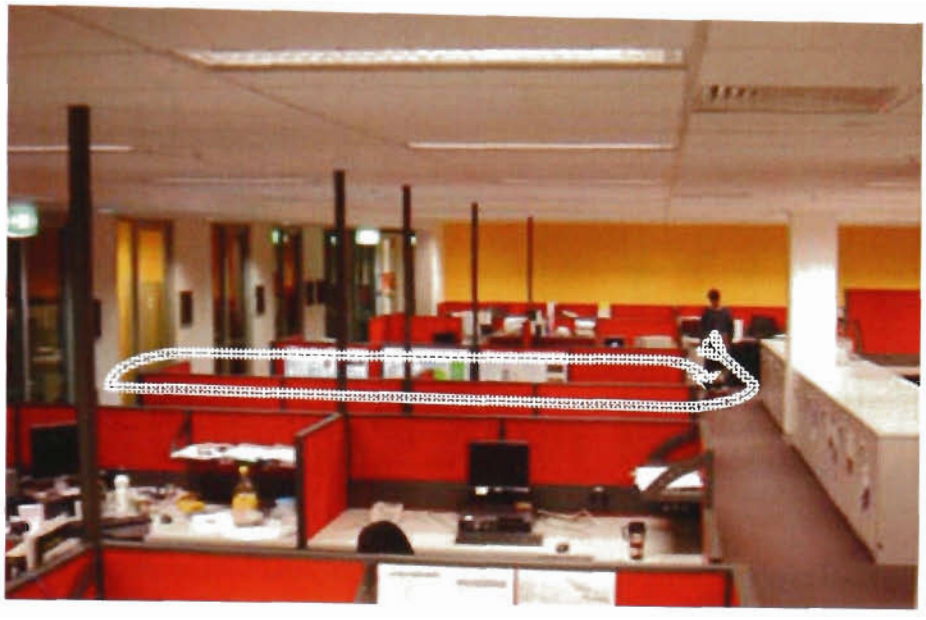


**Figure 6.1: The robotics lab at University of Technology Sydney. Dashed line indicates a small loop that the robot has made in one of the experiments. Aisle to the right is approximately 1m wide.**

## 6.1 Introduction

### 6.1.1 Robotic Platform

Throughout the experiments, an Active Media Robotics Pioneer 2 was used as the robotic platform (see section 3.3.1 for details).

### 6.1.2 Environment

All the experiments presented in the thesis were conducted in the indoor robotics laboratory at University of Technology, Sydney (Figure 6.1). The environment represents a typical indoor office space illuminated with regular fluorescent lighting. Most of the experiments were carried near the workstations indicated by the dashed line in Figure 6.1. The particular environment poses several challenges to a vision based SLAM algorithm.

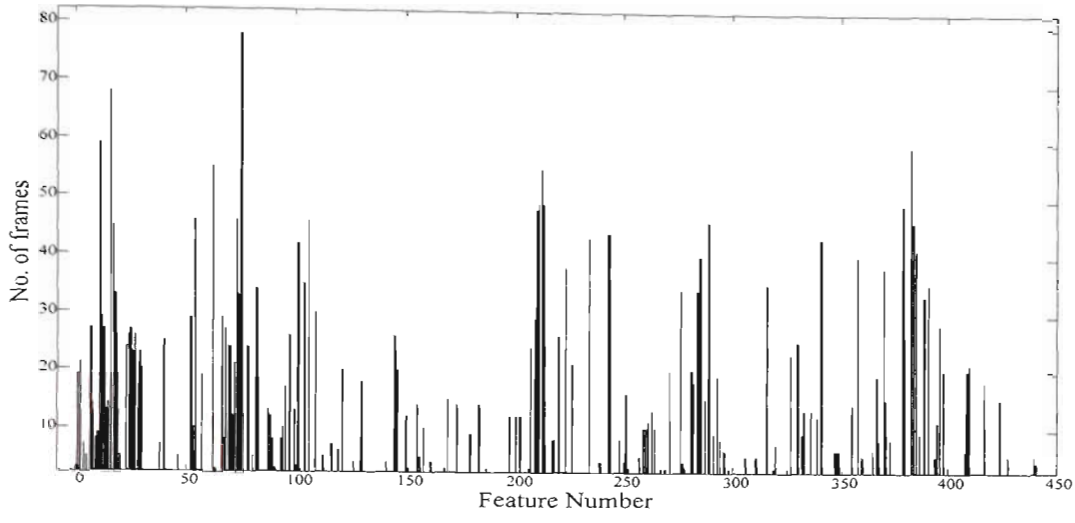



**Figure 6.2: Feature survival histogram of 450 KLT selected features for a small loop around the area indicated in Figure 6.1.**

### 6.1.3 Practical Challenges due to the Environment Structure

As can be seen from Figure 6.1, the passageways are considerably narrow (<2 m) limiting the effective field of view of the camera. As the robot moves, this results in features rapidly moving out of the camera field of view, invariably reducing the information available to the estimator. Also, when previously seen features become unobservable new features need to be initialised in order to maintain the tractability of the filter. This results in a rapidly growing map.

The feature survival histogram shown in Figure 6.2 of a SLAM run in the experimental environment provides an experimental verification of this issue. The robot speed was set at approximately 0.2 $ms^{-1}$ and camera frame rate was 4Hz. In the trial shown, only 7% of the total features remained visible after 40 frames which is about 10s in duration and about 2m of traverse. Marginalisation of the LM features help arrest the incessant growth of map.

Another issue found to be problematic in the environment was its lack of texture. As can be seen from the Figure 6.1 the built environment is mostly monotonous and lacks texture which is needed for reliable feature selection. Also, the structure of the

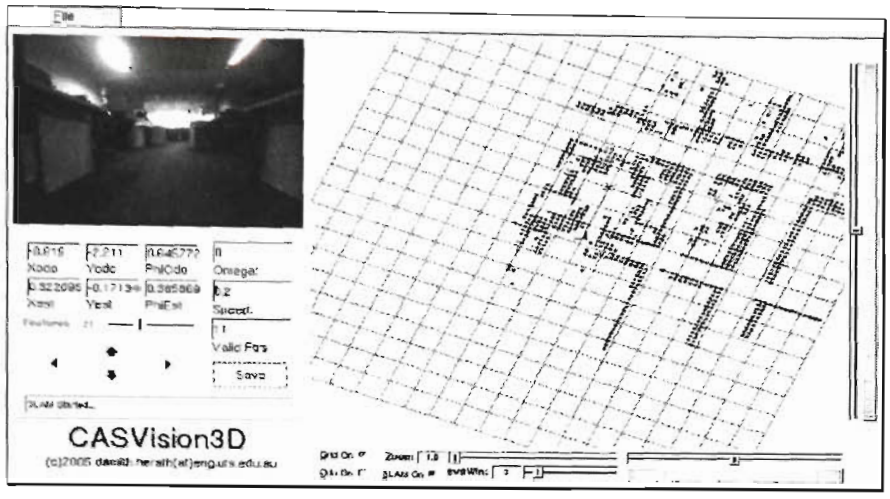



**Figure 6.3: The CASVision3D interface used for controlling the robot, image capture, feature initialisation and tracking. Interface also provides real-time visualisation of the map being built.**

environment is such that, there is increased potential for feature occlusions. This has been illustrated in section 3.3.3 and a solution is proposed in 6.2 in the form of a data validation algorithm.

### 6.1.4 Ground Truth

Knowledge of ground truth is necessary to assess the filter performance. Even though, devices such as GPS are capable of providing accurate position estimates, they are ineffective in an indoor environment. Therefore, a separate feature based SLAM solution using a laser range finder was used to generate 'ground truth'. For this purpose artificial beacons (laser reflectors) were strategically laid out in the experimental environment.

### 6.1.5 CASVision3D

A real time C++ front end was developed for the robot control, image capture, image rectification, stereo processing, feature registration, feature tracking and visualisation purposes for the the experimental work presented in this thesis. Figure 6.3 shows a snapshot of the user interface of CASVision3D after the completion of a small loop.

The interface allows the user to change the number of KLT features initialised/tracked at a given time as well as the speed and turn rate of the robot. It also provides visualisation of current reference image with KLT features overlaid on top and a map of the environment based on laser scans and pose estimates of the robot.

All the SLAM algorithms, including the EKF, batch and MMS were implemented in Matlab and the related results presented in this chapter have been achieved through off-line analysis of stored data that has been collected using the CASVIsion3D interface.

## 6.2 Initial Results

In Chapter 4, it was shown that, even though recursive algorithms such as EKF and UKF tend to produce inconsistent state estimates using the vision sensor, they are considerably accurate to be used as initial estimates in the batch algorithms. Through simulation it was also shown that simple integration of odometric data is not suitable as an initial guess.

However owing to the spurious nature of real stereo vision observations, it was realised that the estimates from a straight forward implementation of such an algorithm are not reliable to be used in the batch optimisation process. This resulted in the novel data validation algorithm discussed below, and later several results from an EKF using real data are presented for comparison.

### 6.2.1 Data Validation Algorithm

Several sources of spurious data in vision sensor were discussed in Chapter 3. It is highly likely that some spurious observations could eventually reach through the filtering stage defying conventional validation gate techniques. This section develops a robust mechanism that rejects such outliers [92].

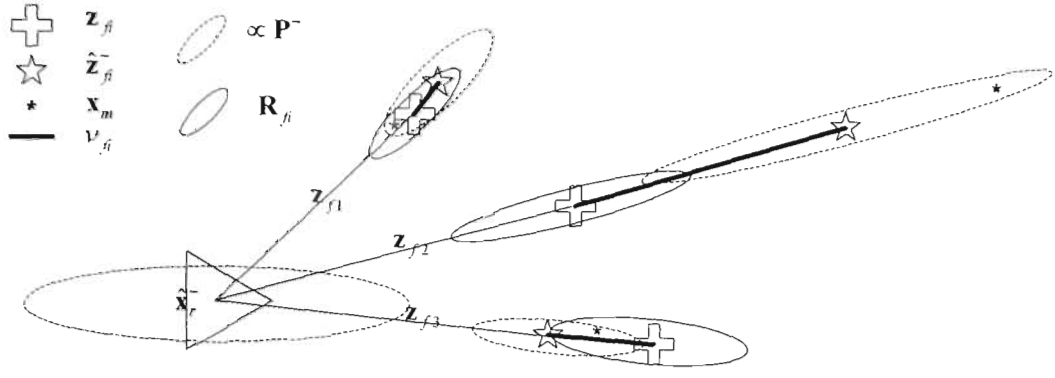



**Figure 6.4: Robot and the map prior to an update where one of the observations to the features is spurious.**

Inspiration for the algorithm stems from the robust Random Sampling estimator RanSaC (Random Sample Consensus) [93] used in the photogrammetry community. The principal idea behind RanSaC is that given a data set that supposedly subscribe to a given model, a random minimal subset of data points are selected recursively to initialise the model parameters. Then the remaining data are tested for conformity with the initialised model. The largest data set that achieves a consensus to the initialised model is selected as the putative solution.

A similar technique is adapted in solving the problem of identifying spurious features in a SLAM scenario. Algorithm could be described with an illustrative example.

**Problem**

In Figure 6.4 a robot observes 3 features at time $k$ of which the observation to the second feature is spurious. Data association is assumed to be known (vision based sensor). Thus the problem is to validate the data generated by the sensor. It is important to note that due to the nature of the stereo vision sensor model, the observation uncertainty correlates to the depth of the observation,

$$\mathbf{R}_{fi} \propto \mathbf{z}_{fi}^{2} \tag{5.20}$$

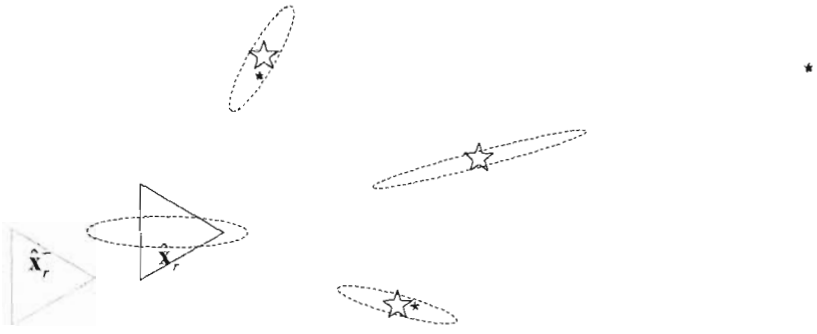


**Figure 6.5: Inconsistent state estimation due to spurious observations.**

### Nearest Neighbour Solution

In the illustration, application of Nearest Neighbour (NN) techniques for validating the associations would result in all the observations being accepted as valid. Thus, the spurious observation will also be used in the final update resulting in the scenario shown in Figure 6.5. In the illustration, the true location of feature 2 lies further away from the robot, however the observation made to this feature indicates a relatively nearer location and due to the effect of (5.20) results in an overly optimistic uncertainty. Since the prior uncertainty of this particular feature is larger, it results in the observation being accepted in a NN criterion. The spurious observation affects the update drastically resulting in inconsistent state estimates.

### Data Validation Algorithm

The novel data validation algorithm relies on the fact that observations at a given time step correlates with the current robot pose. Thus in the SLAM context, the model to be initialised is the current state of the system,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_r(k) \\ \mathbf{x}_m \end{bmatrix} \tag{5.21}$$

and the initialisation of the model is through a virtual update of the state using a random observation (datum) from the group of observations. The resulting scenario is depicted in Figure 6.6 where feature 1 is used in the virtual update. Since, the error

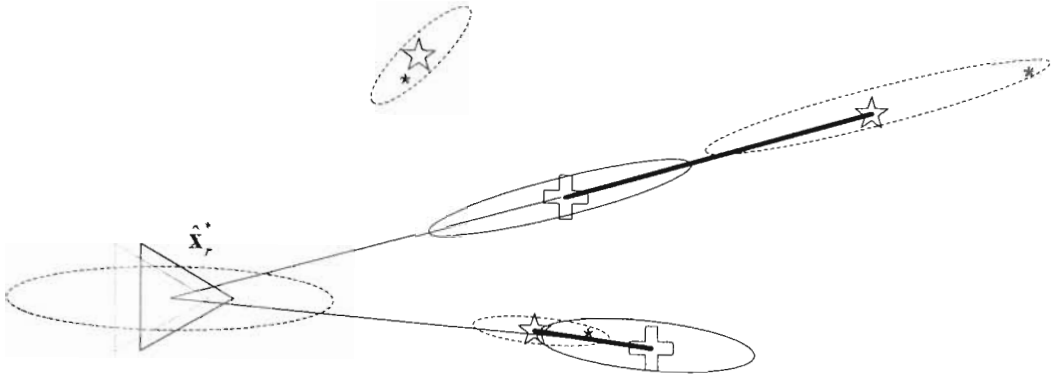


**Figure 6.6: Map and the robot after the virtual update.**

in the estimates always decrease [4] monotonically with each update, the virtual update results in an improved estimate of the states. A consensus set of observations could now be searched for, from the observation space using the improved estimates of the states. The search is terminated when the maximum set of observations is found. This putative set of observations could now be used in the proper update.

### 6.2.2 A comparison of Estimator Error

EKF results play an important role in the implementation of the batch algorithms during the initialisation. As noted earlier an EKF implementation is preferred over an UKF for this purpose due to the slight performance gain [19] of the former.

Figure 6.7 shows SLAM results obtained using an implementation of the EKF in a small loop. As can be seen, the triangulated observation (see Section 4.2.1) model yields poor results. The alternative observation model (see Section 4.3.1) performs better, however the error is still larger compared to the simulation results presented in section 4.3.4 of a comparable loop. This was localised to stem from spurious observations in the experiment. The data validation algorithm developed in the previous section alleviates most of these spurious observations resulting in a better estimate of states and the batch algorithms use the results from this combination of algorithms during the initialisation in the experiments presented later in this chapter.

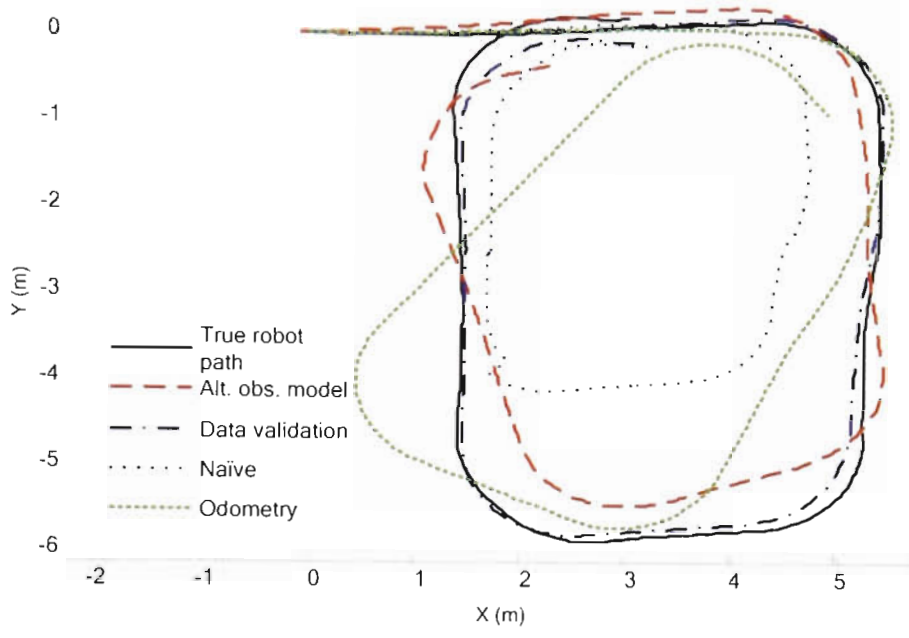



**Figure 6.7: Robot path estimates for real data. Ground Truth is obtained through a laser based EKF SLAM algorithm where estimator errors generally remain under *5cm* for a loop of this size. The alternative observation model performs better compared to a naïve implementation of EKF using the triangulated observation model. Considerably better results could be achieved when the data validation algorithm is implimented.**

## 6.3 Nonlinear Batch Optimisation

Figure 6.8 plots the batch optimisation results for a SLAM run similar to the experiment shown in section 6.2 along with the EKF estimates used for the initialisation. Data association was provided manually by identifying several features that were common in the first frame and the last frame of the data set. As can be seen from the figure, the map estimates from the vision EKF are comparatively optimistic to that of the former. Figure 6.11 gives a close-up view of the robot path estimate. As can be seen, estimation agrees closely with the 'ground truth'. The algorithm required six Levenberg-Marquardt iterations to reach the solution.

Figure 6.10 shows a map of the environment generated using the pose estimates of the batch algorithm and laser scans. The thickness of the walls is a general indication of the error distribution in the robot path estimates. Therefore, quantitatively the algorithm provides sensible estimation of the robot path. Figure 6.11 shows the environment map generated using the 'ground truth' and laser scans for comparison.

Figure 6.13 shows the robot pose error relative to 'ground truth' given by the laser EKF. The outer (dashed) line is the compounded error estimate $\left(2\sqrt{\sigma_{laser}^2+\sigma_{vision}^2}\right)$.

The $2\sigma$ error bounds from the batch algorithm as well as from the vision EKF are shown in the graph for comparison. As expected, vision EKF produced optimistic results. Apart from the error in $y$ estimate, both $x$ direction and heading errors appear to be consistently estimated relative to the compounded error bound. It is possible that the off shoot in y direction to be attributed to any accumulated errors in the laser EKF prior to the loop closure.

Figure 6.12 shows the sparse nature of the Hessian matrix. As discussed in the previous chapters, the structure of the SLAM problem causes the Hessian matrix to be sparse. Therefore, this could be exploited in improving the efficiency of the batch algorithm [10, 43].

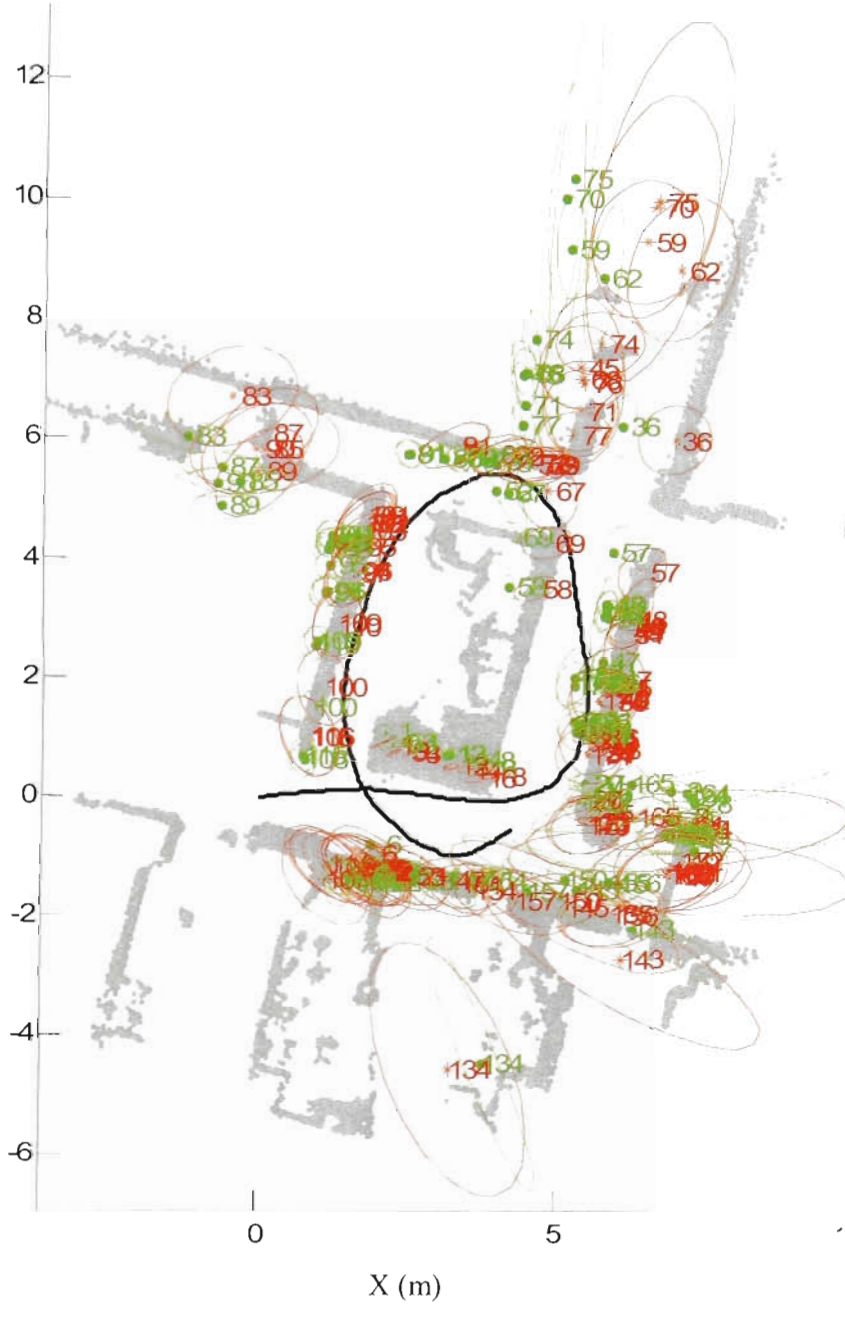



**Figure 6.8: SLAM results using the batch optimisation algorithm with experimental data from a small loop. * indicates map estimates from the batch algorithm. '.' Indicates the vision EKF map estimates. Green and red ellipses represent the covariance estimates from EKF and batch algorithms respectively.**

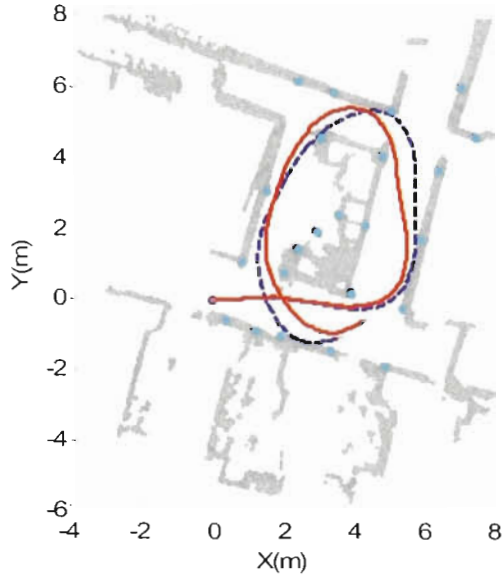



**Figure 6.9: Map of the environment generated using laser scans and robot path estimate from the laser based SLAM algorithm which is used as the ground truth (solid line). '.' Indicates the locations of laser reflectors. Dashed line shows the odometry.**

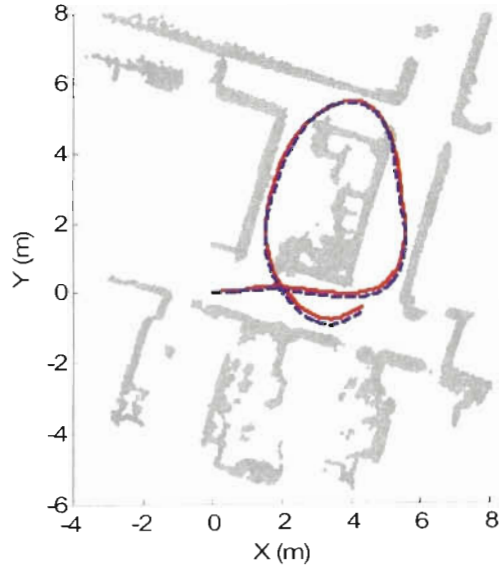



**Figure 6.10: Map of the environment generated using laser scans and robot path estimate (solid line) from the batch algorithm. Dashed line shows 'ground truth'.**

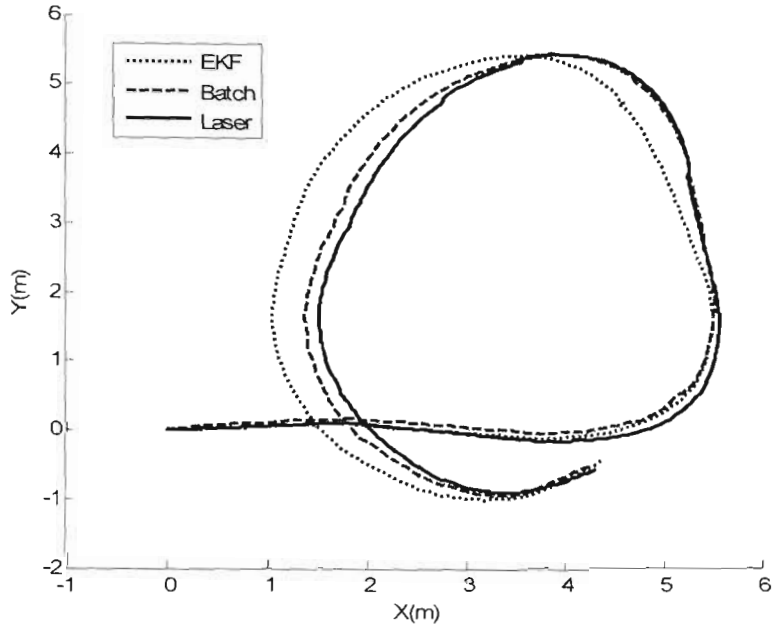



**Figure 6.11: Robot path estimates. Batch estimate agrees well with 'ground truth'.**

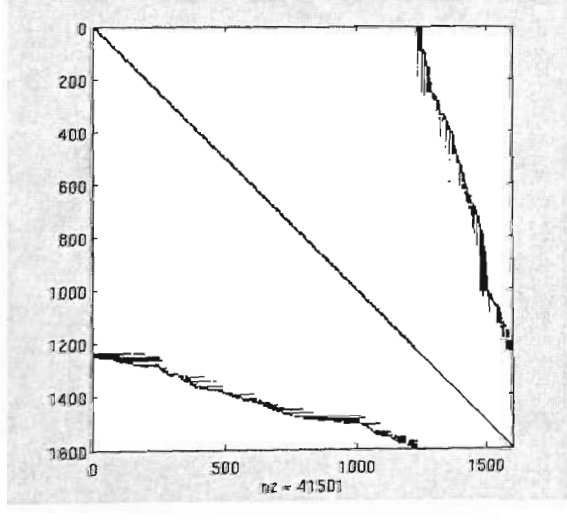



**Figure 6.12: Sparseness of the Hessian. Top (412*3) block corresponds to the robot poses.**

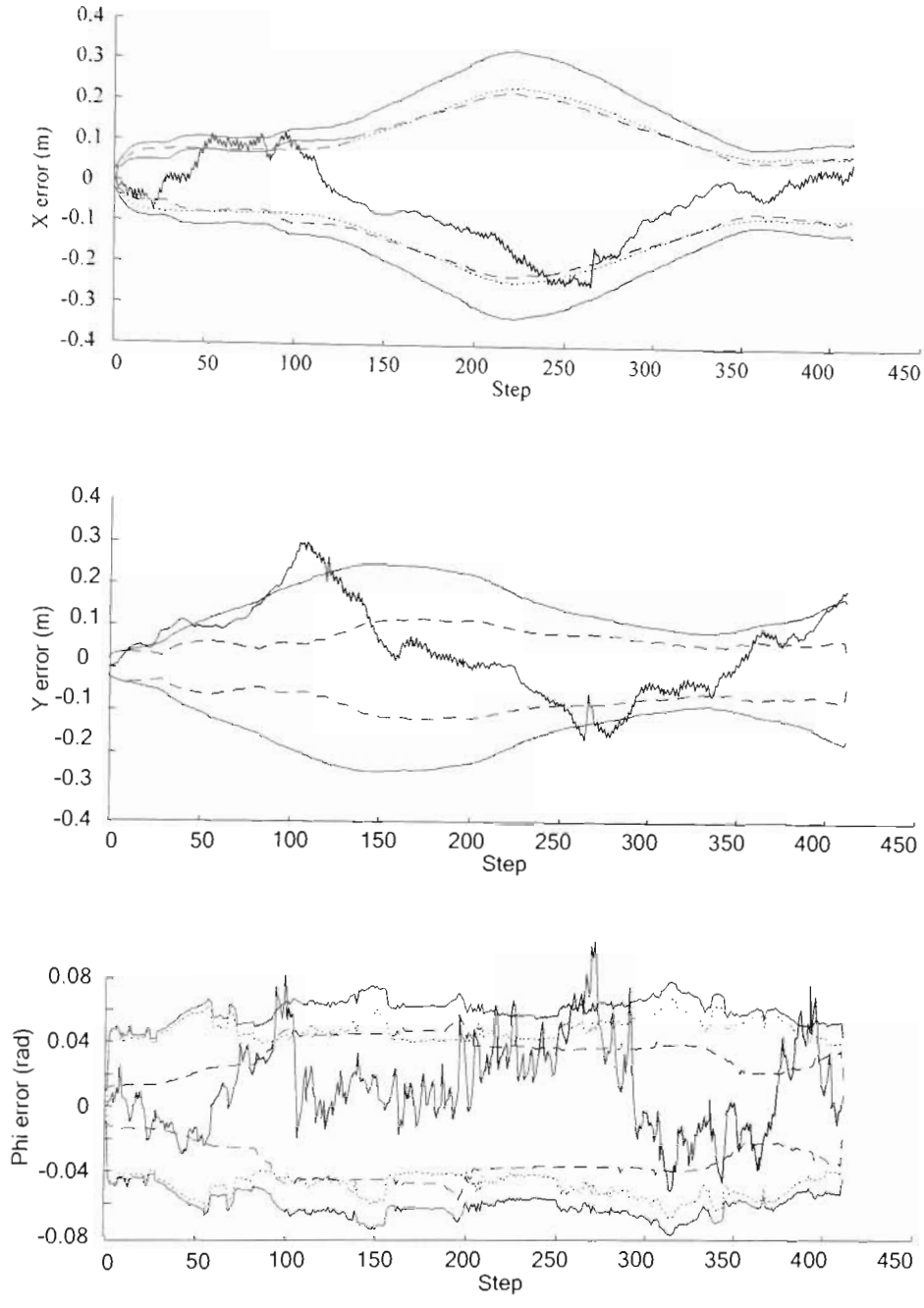



**Figure 6.13: Comparative robot $2\sigma$ bounds (solid lines indicate the combined estimate of laser and batch algorithm, dotted lines indicate estimates from the batch algorithm and the dashed lines are from the vision EKF) and the error in batch SLAM robot pose estimation relative to the laser based EKF (ground truth).**

## 6.4 Multi Map SLAM

An implementation of the Multi Map frame work is presented in this section. A larger navigation horizon was chosen for the experiment in order to assess the stability of the algorithm. Figure 6.14 shows the complete LM with the EKF estimates used for initialisation of the MM algorithm superimposed for comparison. Since the algorithm was executed off-line only two nodes were used for the GM (Figure 6.15). This results in two LMs. The first one corresponds to the smaller inner loop and the second consists of the path that extends to a second outer loop. First loop closure occurs when the robot visits near the origin of the system. The first segment of the robot trajectory consists of the first 700 poses. At the $700^{th}$ pose MM optimisation was carried out and the corresponding LM was marginalised. Then the second segment begins and it ends when the robot stops after traversing the outer loop. The second segment constitutes 560 robot poses. At the end of the robot trajectory a MM optimisation is performed again.

Figure 6.16 shows a close-up of the complete path estimate and the GM. Considerable irregularities in the path estimate near the right corner are due to a person moving in front of the camera. However, the estimates recover from the errors when the second loop is closed later in the trajectory.

Most important to notice is the fact that the novel MM approach enables highly compressed Map representations of considerably larger environments. In this example, only 15 map features were required to represent a large loop. If a more traditional approach was to be used the map would constitute of approximately 450 features.

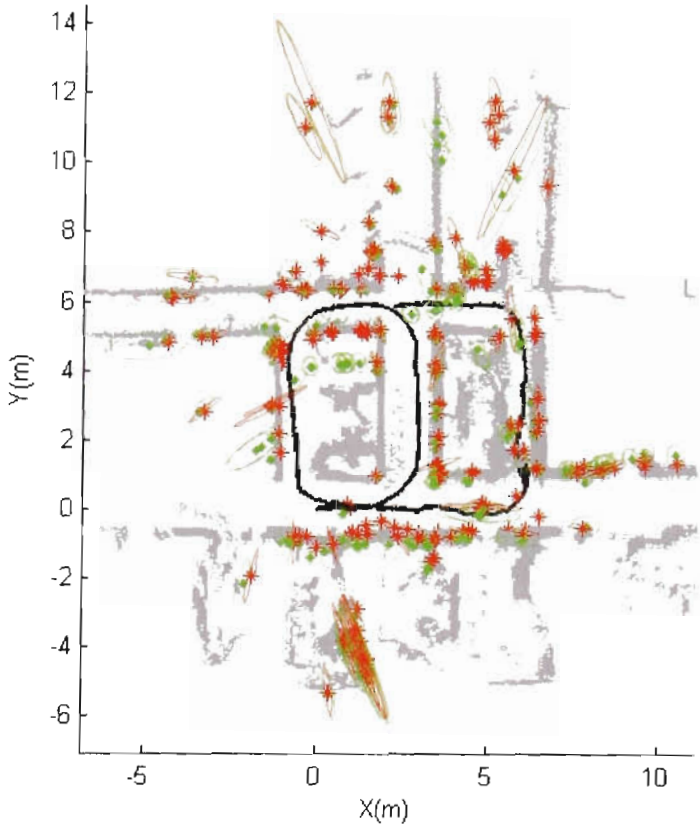



**Figure 6.14: Multi Map estimates of the LM and robot path using real data from a large loop. Green ellipses are the estimates of LM features from the EKF used for initialisation.**

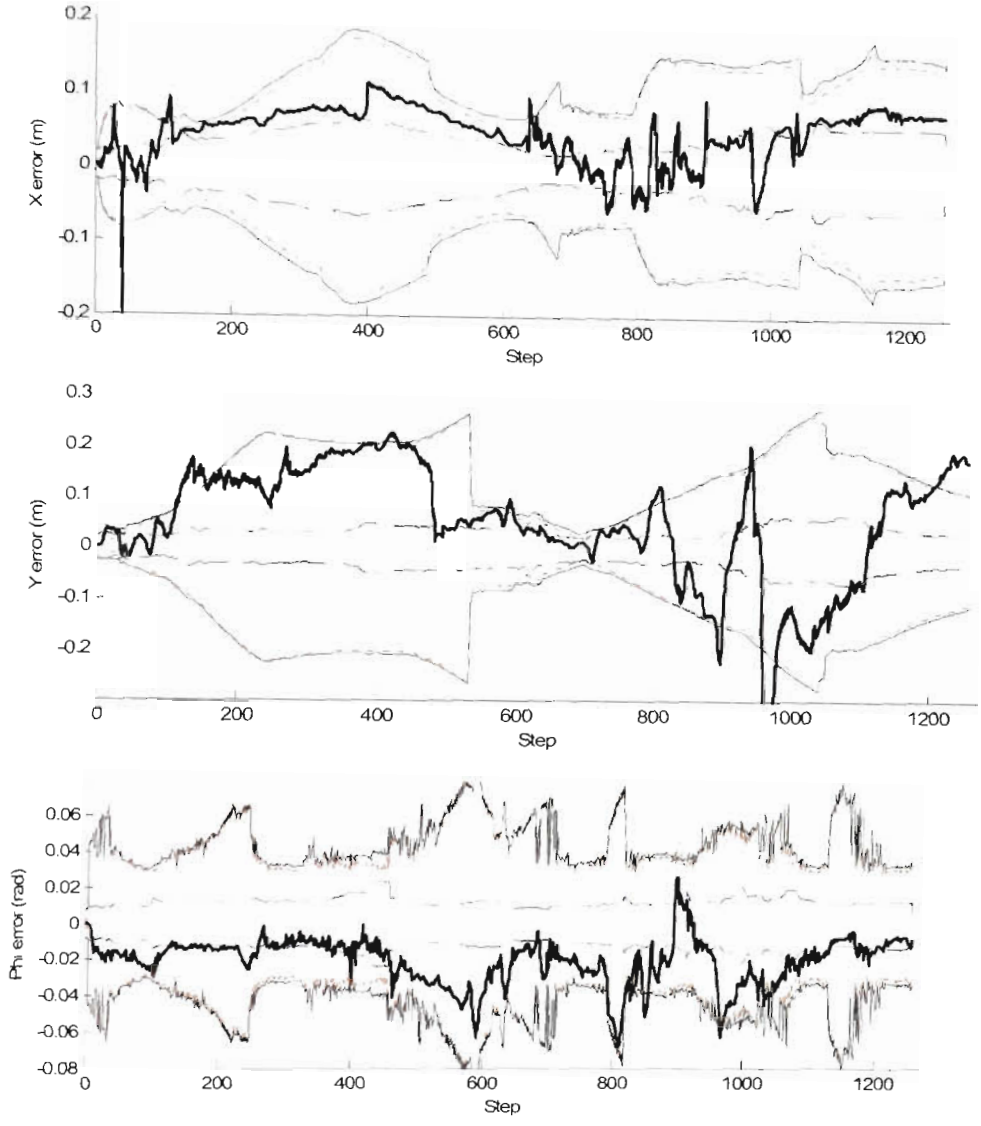



**Figure 6.15: Error in MM SLAM robot pose estimation relative to the laser based EKF (ground truth) and comparative $2\sigma$ bounds (solid lines indicate the combined estimate of laser and MMS algorithms, dotted lines indicate estimates from the MMS algorithm and the dashed lines are from the vision EKF).**

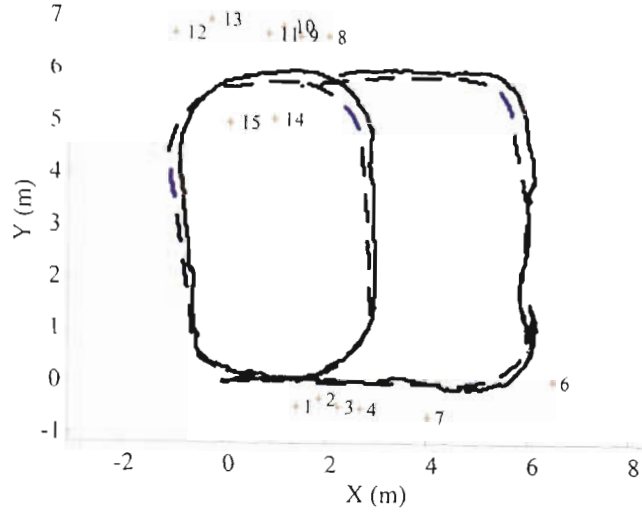



**Figure 6.16: MMS estimate of the Global Map and the robot path. Dashed line shows the 'ground truth'. GM consists of two nodes with features 1-7 belonging to the first node and 8-15 belonging to the second.**

### Selecting GM Features

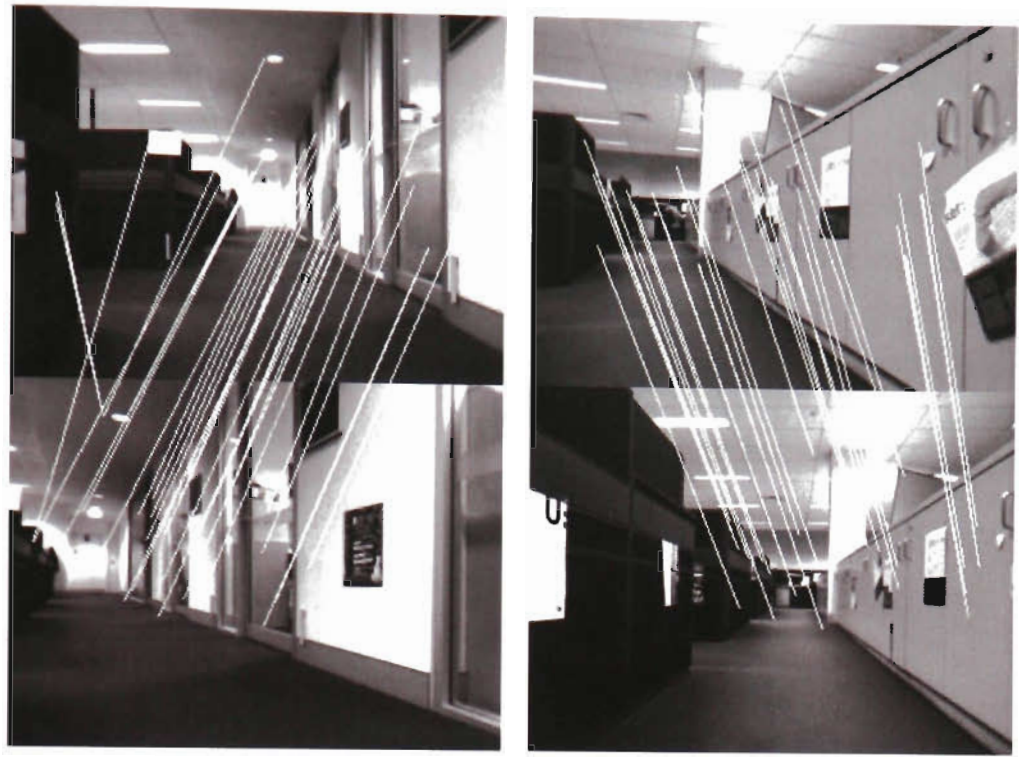
The GM features were selected using the SURF [38] algorithm and an implementation in C++ is available on the web[1]. Figure 6.17 shows some of the features selected by the SURF algorithm.

As can be seen from the images, SURF tends to pick-up large number of potential features in each image. In the environment presented SURF generally pick-up 120 ~ 150 features per image. Since the algorithm was executed off-line, it was possible to select a minimal set of features that were observable in consecutive images (locally observable) during the run as well as during the loop closure (globally observable). Figure 6.18 shows two examples of feature correspondence made through the SURF descriptor for potential loop closure detection. However, in an online version of the algorithm it might require additional marginalisation of GM features that appear not to be contributing the loop closure.

[1] http://www.vision.ee.ethz.ch/~surf/

**Figure 6.17: Typical SURF features selected in the experiment environment.**

The SURF matching criteria tend to pick-up potentially large number of mismatches between images, especially when the difference in the image viewpoints is considerably larger (see Figure 6.18). In order remove such mismatches, a batch matching criterion was used such that during the loop closure, potential features were matched across a sequence of images used for the initialisation of GM features as well as for the loop closure (local observability). Putative matches were then cross matched between the two sequences (global observability). The combination of local observability and global observability check removed all the wrong correspondences in the experiment presented. However, if higher robustness is desired, alternate descriptors such as SIFT [33, 77] could be used.



(a) (b)

**Figure 6.18: SURF correspondence. (a) GM features corresponding to potential first node. (b) GM features corresponding to potential second node**

## 6.5 Summary

This chapter presented several experimental implementations of the algorithms put forth in the thesis. In the absence of absolute estimates of the states, a laser based EKF SLAM algorithm was used to compare estimator errors. An algorithm for detecting spurious stereo vision observations was also discussed. All the experiments were carried in an office space where the regular nature of the built environment poses considerable challenges to the successful implementation of a SLAM algorithm. Some of these issues were discussed in detail in section 6.2.

Section 6.3 presented experimental results using the batch algorithm developed in Chapter 4. A small loop was used to illustrate that the algorithm is capable of

producing sensible estimates. In fact it was shown that, EKF based results are unreliable in the particular sensor domain and a viable alternative could be found in the batch algorithm developed in this thesis.

Section 6.4 presented results of the MMS approach applied to a larger SLAM problem. SURFs were used as GM features. In this experiment the robot has completed several loops necessitating multiple loop closures. The algorithm was shown to produce consistent results against the laser based ground truth measure. Most importantly, only 15 GM features were required to represent the large environment.

# Chapter 7

# Conclusions

This thesis has investigated a specific instance of the SLAM problem where a small baseline stereo vision system was used as the primary sensor. The choice of sensor was justified in terms of cost, compactness, wide applicability and tractability of SLAM algorithms compared to single camera methods. The work presents several insights into the nature of the problems faced in the process of developing SLAM algorithms using the sensor. Section 7.1 summarises these problems and the solutions developed. Section 7.2 provides an outlook towards future research.

## 7.1 Summary of Contributions

The main contributions result from the identification of the filter consistency issue related to the stereo vision sensor employed, the use of alternate stochastic estimation techniques and the subsequent development of various algorithms for addressing the consistency issue as well as the development of a novel framework for efficient SLAM implementation. These contributions are discussed in summary below.

### 7.1.1 Sensor Characterisation

Chapter 3 presented the stereo vision sensor in the context of SLAM. The sensor unifies several key techniques that are needed to realise a stereo vision based SLAM algorithm. An extensive empirical study was then conducted to asses the characteristics of the sensor. The challenge was to understand the sensor in the SLAM environment. This is considerably different from studying stereo vision algorithms or tracking algorithms from a computer vision perspective, in that the primary requirement is to characterise the complete sensor behaviour on a dynamic platform. Issues pertaining to the unique combination of hardware and software were identified through the experiments presented in chapter 3. One of the key findings is that although, in general it is possible to assume Gaussian error distributions in the observations, there is high susceptibility of observations being corrupted by spurious data.

Two sources were identified that contributes to generation of spurious data. Firstly, there is considerable noise generated in the stereo correspondence algorithm itself. Even though the commercially available algorithm used in the work employs several noise suppression mechanisms, it was observed that depending on environmental factors it generates large number of outliers. In light of this, a signal conditioning algorithm was developed for improving the reliability of the stereo correspondence algorithm, with particular attention given to the nature of the experimental environment.

Secondly, the tracking algorithm was also found to contribute to the problem. Most of the errors were identified to stem from occlusions and intensity variations. These errors are more subtle compared to the errors in stereo correspondence. Hence they are harder to detect through general techniques such as statistical gating methods [4]. Nevertheless, in the long run such spurious data are catastrophic in a SLAM implementation. A novel algorithm based on the RanSaC [93] algorithm was

developed in chapter 4 that detects such spurious feature resulting in considerable improvements in SLAM estimates.

Chapter 3 also looked at the nonlinear nature of the observation model. Particular attention was paid to the sensor bias as reported by several authors including Sibley *et al* [74]. Both experimental and simulation studies were carried out to understand the effects of the observation model. The findings indicated that the small baseline stereo system yields biased depth estimates even at short depth (<3m) and the general approximations made in error propagation could result in inaccurate error estimates being made of the observations. Effects of which were made apparent in the beginning of chapter 4.

### 7.1.2 Consistent Stochastic Estimation

Chapter 4 provided an insight into the behaviour of recursive estimators such as the EKF in solving SLAM problem using the stereo vision sensor described in chapter 3. Particular attention was paid to the inconsistent filter performance due to the nonlinear nature of the observation model. Various potential techniques were discussed as alternative approaches to realising consistency in state estimates of the standard EKF implementations. It was demonstrated that conventional recursive filtering mechanisms known to perform well using traditional sensors were incapable of producing consistent estimates in the particular vision sensor domain.

A batch method inspired by the techniques used in Structure from Motion estimation in the computer vision research was developed as an alternative to the recursive estimation. A novel formulation that integrates odometric information to the batch estimation was also proposed. It was shown through empirical evidence that the new batch estimation algorithm produces consistent state estimates.

Through empirical studies it was shown that, even though recursive algorithms produced inconsistent error estimates, the mean errors remained sufficiently low to be used as an initial estimate in the batch estimation process.

### 7.1.3 Multi Map SLAM

Chapter 5 developed a novel Multi-Map representation for the SLAM problem. Inspiration for the framework stems from observations made at human navigation patterns. The framework relies on a two tier map representation. By allocating individual maps to the two fundamental tasks in the SLAM framework (i.e. observing new features and re-observing old features) considerable computational gains are achieved and the real-time performance capabilities of the recursive algorithm is retained even in large exploration tasks. By integrating batch optimisation techniques at key points, consistency of the state estimates is also preserved.

The first map, called the Local Map (LM) is a volatile map that aids navigation between local points of interest (PoI). A recursive estimator uses LM features to provide an initial estimate of robots path to a batch optimisation algorithm called the Multi Map algorithm. The second map referred to as the Global Map (GM) contains descriptive features [38] anticipated to aid SLAM loop closure. Both maps were used in the Multi Map optimisation for consistent state estimation at key locations in the environment marked by PoI. Once an optimal minimum is found, features belonging to the current LM were marginalised from the state vector and were not used in future optimisation cycles, thereby considerably reducing the computational cost. This technique also alleviates another common problem in vision based SLAM applications. In comparable environments, vision sensors tend to generate large number of natural features. For instance, SURF [38] algorithm would pick-up 120-150 features compared to the 3-4 features picked-up by a laser range finder. This could potentially lead to very large maps resulting in inefficient estimators. On the other hand, a large number of features are required to achieve robustness due to the limited field of view of the sensor. Therefore, the marginalisation technique in MM framework lends well to the algorithm efficiency in the context in which it was developed.

Since the features in the LM are not effective in loop closure the marginalisation does not contribute to loss of information. This is due to the implementation modality of the LM features which are based on the KLT algorithm.

### 7.1.4 Experimental Validation

In chapter 6, techniques developed in the thesis were demonstrated using real data collected in an indoor office like environment. Various aspects of the implementation were discussed with particular attention to the practicality of the techniques developed.

## 7.2 Future Research

The observations by Faugeras, a leading researcher in the field of computer vision as quoted at the beginning of chapter 1 summarises the current state of the art in computer vision, and stereo vision is no exception. In fact, our own experience with stereo vision suggests that further research is necessary to achieve comparable robustness of SLAM algorithms to those implemented with traditional sensors. A reciprocating approach between stereo vision and robotics research is envisioned to be the best path forward. Some of the outstanding issues that could be addressed in future work are discussed below.

### 7.2.1 Real-time Implementation

Any robotic system that potentially utilise the service of a SLAM algorithm would require the algorithm to be implementable in real-time. However, 'real-time' is a relative term that depends on the application domain and the available resources. Therefore, it is more appropriate to study such algorithms in terms of the computational complexity.

In the algorithms presented in this thesis, apart from the obvious computational gains realised through the compact GM representation, attention has not been paid towards

computational aspects of the algorithms put forth in this thesis. Further investigations could be directed towards possible computational improvements.

As noted in [43], photogrammetry research has developed techniques to speed-up the general batch optimisation process by utilising the sparse nature of the structure from motion problem. Similar techniques have been considered lately from a SLAM perspective. Notable are the contributions made by Dellaert et al [10]. However, in the current MMS implementation, marginalisation could potentially introduce large non zero blocks in parts of the Hessian matrix where previously were zeros, and as the MMS evolves, the entire matrix could get filled in, which results in the algorithm becoming computationally intractable. Re-ordering techniques [10] and sub-optimal partial elimination techniques [12] could be looked at in developing more efficient MM schemes.

### 7.2.2 Point of Interest

The scheduling of a PoI at present is purely empirical. For instance, the examples presented in previous chapter imply that certain amount information is known *a priori* of the environment in order to place the PoI at the locations indicated. In a realistic navigation scenario, such inferences are generally not plausible. As such, a more principled approach is needed in the placement of PoI. Several parameters could be considered in developing such an approach.

For instance, MM framework uses EKF estimates as the initial guess. However, after a certain time, the potential growth in the EKF estimator error could result in the batch optimisation failing to achieve acceptable local minima as shown in section 5.4.4. Therefore, the range of robot in each segment of the trajectory is bounded by the accuracy of the initial estimate, a problem that could be studied empirically for developing guidelines in placement of PoIs.

A proactive approach could be realised in the placement of PoIs by anticipating potential loop closures. For instance active loop-closing [94] techniques and

maximally informative path planning techniques [95], where the optimisation strategy could be augmented to reward detection of GM clusters for potential loop closure, could be used to proactively direct the robots evolution and the placement of PoIs. It is worth noting that in the selection of suitable PoIs, the emphasis is on identifying points in the robot's motion space that are regularly visited rather than finding visually distinctive areas, since the feature selection criteria has already imposed the necessary distinctiveness for reliable correspondence.

### 7.2.3 Information Loss

The general argument put forth in Chapter 5 that the marginalisation of LM features from the state vector does not constitute information loss needs to be further studied in the context, that the approximations made during the linearization of the observations could potentially contribute to considerable information loss. Effects of these approximations in the long run are yet to be studied and could indicate potential limitations of consistency in the MM framework.

## 7.3 Conclusions

This thesis has presented several tools to be used in the deployment of a SLAM system using stereo vision in an indoor environment. The collection of algorithms addresses issues relating to consistency and efficiency of the implementations. However, there is still scope for further improvements to be made to these algorithms to fulfil the ultimate goal of realising fully autonomous robots exploring unknown environments engaged in lengthy missions with considerable robustness.

# Appendix A

# Stereo Camera and Related Algorithms

## A.1 Stereo Head

The Videre Design camera model STH-MDCS/C [64] (see Figure A.1) has an approximate baseline of 0.9 m and the lenses used have an effective field of view of 84.9° x 68.9° . Performance and Specification parameters of the camera and the lenses are summarised in Table A.1 and A.2 respectively.


**Figure A.1: Videre Design STH-MDCS/C colour stereo camera**

**Table A.1: Stereo head specifications**

| | |
|---|---|
| Model | STH-MDCS/-C Stereo Head - *Videre Design* |
| Imagers | ½" format CMOS 1280x960 active area Progressive scan Colour or monochrome |
| Digital Camera Specification | Version 1.30 |
| Formats | 1280x960, 640x480 8 bit monochrome or Bayer colour pattern |
| Frame Rates | 3.75, 7.5, 15, 30 Hz Max 7.5 Hz at 1280x960 |
| Exposure | 1 line time to full frame |
| Gain | 0 – 22 dB |
| Sensitivity | 2.5 V/lux-sec |
| S/N | > 55 dB, no gain |
| Power | < 1 W |
| Synchronisation | Internal: pixel-locked External: 60 us |
| Lens | 6.0 mm F 1.4 C mount |
| Size | 1.5" high x 5" long x 1" deep (excluding lenses) |
| Stereo Baseline | 9 cm |
| SVS software | Linux kernel 2.4 MSW 98SE, ME, 2000 and XP |

**Table A.2: Wide angel lens specifications**

| | |
|---|---|
| Model / Maker | H35CSWI / International Space Optics (Rainbowcctv) |
| Focal Length: | 3.5mm |
| Maximum Relative Aperture: | 1:1.6 |
| Iris: | F1.6-Close |
| Mount: | CS-mount |
| Minimum Object Distance: | 0.1m (from front vertex) |
| Angular Field of View: | 1/2": 84.9° x 68.9° |
| Optical Back Focal Distance: | 11.45mm (in air) |
| Operation: | Focus: Manual<br>Iris: Manual |
| Operation Temp: | -10 ~ +50°C (+14 ~ 122°F) |
| Filter Size: | 43mm P0.75 |
| Weight: | Approx. 90g (3.1oz) |

## A.2 Calibration

Automated calibration routines are provided with the hardware for camera intrinsic and extrinsic parameter estimations. Tsai [96] and Faugeras [83] has in-depth discussions on similar calibration procedures. The calibrations are based on a planar test pattern with known geometric relationships. Use of the wide angle cameras necessitates the estimation and application of radial distortion parameters for the camera. Once these parameters are estimated the raw camera images can be warped in to the standard position which represents the ideal pin hole camera model. Figure A.2 shows a sample image from the left and right camera along with the corresponding ideal pinhole camera representations.

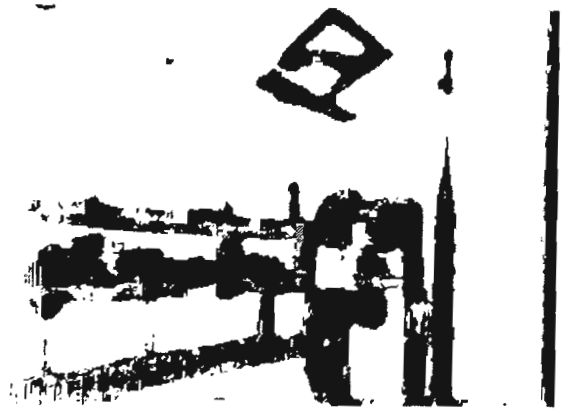
Although the supplied calibration software provides an optimal set of parameters for the camera it does not guarantee optimailty of individual parameters since these are estimated in a 'bundled' optimisation method. Especially given the wide-angle nature and close proximity at which the calibration pattern needs to be presented to the camera it was observed that minor improvements to some of these parameters was possible through additional calibration procedures. A discussion of this is presented in section 3.6.3.

## A.3 Stereo correlation Algorithm

A software API is provided with the stereo head for generating dense stereo range images called the 'Small Vision System (SVS)'[64]. The API provides access to a fast Area Correlation method from which a disparity image can be derived (Figure A.2c). This image represents disparity between left and right image patches at the best match for each pixel in the left (reference) image. For a more thorough treatment of the subject on stereo correlation reader is referred to the works by Faugeras [83] and a recent survey of the state of the art in stereo algorithms by Scharstein and Szeliski [97].

(a)

(b)



(c)

**Figure A.2: Example images from the stereo process (a) Raw Left and right images from the camera (Notice the radial distortion) (b) Rectified images (c) Disparity image**

Depending on the image composition it is possible to generate mismatches occasionally. SVS algorithm provides texture confidence thresholding and a Left-

Right Check as post filtering processes for removing such high frequency noise components from the disparity image.

## A.4 KLT Tracking Algorithm

Given two images $I,J$ assuming small inter-frame displacements, image motion can be described by suitably moving every point in the current frame to achieve the next frame,

$$I(x,y,t+\tau) = I(x-\xi(x,y,t,\tau), y-\eta(x,y,t,\tau)) \qquad (A.1)$$

displacement of a point at x is given by $\delta = (\xi,\eta)$ and using the affine motion model,

$$\boldsymbol{\delta} = \mathbf{D}\mathbf{x} + \mathbf{d} \qquad (A.2)$$

where, $\mathbf{D} = \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix}$ is the deformation matrix and $\mathbf{d}$ is the displacement vector.

A point at x in the first image $I$ moves to point $\mathbf{A}\mathbf{x}+\mathbf{d}$ in the second image $J$.

$$J(\mathbf{A}\mathbf{x}+\mathbf{d}) = I(\mathbf{x}) \qquad (A.3)$$

Where, $\mathbf{A} = 1+\mathbf{D}$ .In order to find the above motion parameters $\mathbf{A}$ and $\mathbf{D}$ minimise dissimilarity,

$$\varepsilon = \iint_w [J(\mathbf{A}\mathbf{x}+\mathbf{d}) - I(\mathbf{x})]^2 w(\mathbf{x})\, d\mathbf{x} \qquad (A.4)$$

where $w(\mathbf{x})$ is a weighting function, which is set to 1 in the implementation.

# Bibliography

[1] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge, UK: Cambridge University Press, 2000.

[2] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Second Annual Conference on Uncertainty in Artificial Intelligence (UAI '86)*, University of Pennsylvania, Philadelphia, PA, USA, 1986 pp. 435-461

[3] R. Smith, M. Self, and P. Cheeseman, "A Stochastic Map For Uncertain Spatial Relationships," in *Fourth International Symposium on Robotics Research*, 1987.

[4] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," *IEEE Transactions On Robotics and Automation,* vol. 17, pp. 229-241, June 2001.

[5] S. Majumder, H. Durrant-Whyte, S. Thrun, and M. d. Battista, "An Approximate Bayesian Method for Simultaneous Localization and Mapping," *submitted to IEEE Trans. Robotics and Automation.*

[6] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE,* vol. 92, pp. 401-422, 2004.

[7] R. Martinez-Cantin and J. A. Castellanos, "Unscented SLAM for large-scale outdoor environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, 2005, pp. 3427-3432.

[8] J. Andrade-Cetto, T. Vidal-Calleja, and A. Sanfeliu, "Unscented Transformation of Vehicle States in SLAM," in *IEEE International Conference on Robotics and Automation (ICRA 2005)* 2005, pp. 323-328.

[9] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *14 th American Control Conference*, Seattle, WA, 1995

[10] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing," *International Journal of Robotics Research,* vol. 25 December 2006

[11] Lemaire, T., Lacroix, S., Sola, and J., "A practical 3D bearing-only SLAM algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, 2005, pp. 2449-2454.

[12] P. F. McLauchlan, "The Variable State Dimension Filter applied to Surface-Based Structure from Motion," School of Electrical Engineering, Information Technology and Mathematics,University of Surrey, Guildford CVSSP Technical Report VSSP-TR-4/99, 1999.

[13] K. Jae-Hean and C. M. Jin, "Absolute Stereo SFM without Stereo Correspondence for Vision Based SLAM," 2005, pp. 3360-3365.

[14] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Y. Ng, "Simultaneous Mapping and Localization With Sparse Extended Information Filters: Theory and Initial Results," Computer Science Department, Carnegie Mellon University, Pittsburgh, PA CMU-CS-02-112, 2002.

[15] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended

Information Filters " *The International Journal of Robotics Research,* vol. 23, pp. 693-716, 2004.

[16] G. Sibley, G. Sukhatme, and L. Matthies, "The Iterated Sigma Point Filter with Applications to Long Range Stereo," in *Robotics: Science and Systems II*, Cambridge, USA, 2006.

[17] A. Henrique, P. Selvatici, A. Helena, and R. A. Costa, "Fast loopy belief propagation for topological Sam," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, 2007, pp. 664-669.

[18] S. Thrun, "Robotic Mapping: A Survey," School of Computer Science, Carnegie Mellon University, Pittsburgh CMU-CS-02-111, February 2002.

[19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics* Cambridge, Mass.: MIT Press, 2005.

[20] G. Dissanayake, H. Durrant-Whyte, and T. Bailey, "A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem," in *IEEE International Conference on Robotics and Automation (ICRA '00)*, 2000 pp. 1009 - 1014.

[21] G. Dissanayake, S. B. Williams, H. Durrant-Whyte, and T. Bailey, "Map Management for Efficient Simultaneous Localization and Mapping (SLAM)," *Autonomous Robots,* vol. 12, pp. 267-286, 2002.

[22] S. B. Williams, "Efficient Solutions to Autonomous Mapping and Navigation Problems," in *Department of Mechanical and Mechatronic Engineering*. vol. PhD Sydney: The University of Sydney, 2001.

[23] Williams, S.B, Dissanayake, G., Durrant-Whyte, and H., "An efficient approach to the simultaneous localisation and mapping problem," in *IEEE International Conference on Robotics and Automation (ICRA '02)*, 2000, pp. 406 - 411

[24] T. Bailey, "Mobile Robot Localisation and Mapping in Extensive Outdoor Environments." vol. PhD Sydney: The University of Sydney, 2002.

[25] Z. Zivkovic, B. Bakker, and B. Krose, "Hierarchical map building using visual landmarks and geometric constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, 2005, pp. 2480-2485.

[26] P. M. Newman, "On the Structure and Solution of the Simultaneous Localisation and Map Building Problem," in *Australian Centre for Field Robotics*. vol. PhD Sydney: The University of Sydney, 1999, p. 178.

[27] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '91)*, 1991, pp. 1442 - 1447.

[28] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation,* vol. 17, pp. 890 - 897 December 2001.

[29] V. Lepetit, A. Shahrokni, and P. Fua, "Robust data association for online application," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003, pp. I-281 - I-288.

[30] L. D. L. Perera, W. S. Wijesoma, and M. D. Adams, "Data association in dynamic environments using a sliding window of temporal measurement frames," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, 2005, pp. 753-758.

[31] J. J. Leonard, P. M. Newman, R. J. Rikoski, J. Neira, and J. D.Tardós, "Towards Robust Data Association and Feature Modeling for Concurrent Mapping and Localization," in *The Tenth International Symposium of Robotics Research (ISRR'2001)*, Lorne, Victoria, Australia, 2001, pp. 7 - 20.

[32] Nieto, J., Guivant, J., Nebot, E., Thrun, and S., "Real time data association for FastSLAM," in *IEEE International Conference on Robotics and Automation (ICRA '03)* 2003, pp. 412 - 418.

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision,* vol. 60, pp. 91-110, 2004.

[34] S. Se, D. Lowe, and J. Little, "Mobile Robot Localization And Mapping with Uncertainty using Scale-Invariant Visual Landmarks," *International Journal of Robotic Research,* vol. 21, August 2002.

[35] A. Gil, O. Reinoso, O. M. Mozos, C. Stachniss, and W. Burgard, "Improving Data Association in Vision-based SLAM," 2006, pp. 2076-2081.

[36] J. Shi and C. Tomasi, "Good Features toTrack," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94)* Seattle, 1994, pp. 593 - 600

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

[38] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," in *Ninth European Conference on Computer Vision*, 2006.

[39] P. Newman and K. Ho, "SLAM- Loop Closing with Visually Salient Feature," in *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[40] C. Chen and H. Wang, "Appearance-based topological Bayesian inference for loop-closing detection in a cross-country environment," *The International Journal of Robotics Research,* vol. 25, pp. 953-984, October 2006.

[41] M. Cummins and P. Newman, "Probabilistic Appearance Based Navigation and Loop Closing," in *IEEE International Conference on Robotics and Automation (ICRA '07)*, Roma, Italy, 2007.

[42] F. Lu and E. Milios, "Globally Consistent Range Scan Alignment for Environment Mapping," *Autonomous Robots,* vol. 4, pp. 333-349, October 1997.

[43] M. Deans, "Bearings-Only Localization and Mapping," in *Robotics Institute.* vol. PhD Pittsburgh, PA: Carnegie Mellon University, 2005.

[44] U. Frese, "A Discussion of Simultaneous Localization and Mapping " *Autonomous Robots* vol. 20, pp. 25-42, January 2006.

[45] D. Simon, *Optimal state estimation : Kalman, H. [infinity] and nonlinear approaches*. New Jersey: John Wiley, 2006.

[46] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*. New York: Cambridge University Press, 1986.

[47] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM Algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, Beijing, China, 2006, pp. 3562-3568.

[48] J. A. Castellanos, J. Neira, and J. D. Tardos, "Limits to the consistency of EKF-based SLAM," in *5th IFAC Symp. on Intelligent Autonomous Vehicles (IAV'04)*, Lisbon, Portugal, 2004.

[49] S. J. Julier and J. K. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *IEEE International Conference on Robotics and Automation (ICRA 2001)*, 2001, pp. 4238-4243.

[50] S. Huang and G. Dissanayake, "Convergence Analysis for Extended Kalman Filter based SLAM," in *IEEE International Conference on Robotics and Automation (ICRA 2006)*, Orlando, Florida, 2006.

[51] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Somerset, New Jersey: Wiley InterScience, 2001.

[52] C.-C. Wang, "Simultaneous Localization, Mapping and Moving Object Tracking," in *Robotics Institute*. vol. Doctor of Philosophy Pittsburgh, PA 15213: Carnegie Mellon University, 2004, p. 164.

[53] J. Guivant, J. Nieto, F. Masson, and E. Nebot, "Navigation and Mapping in Large Unstructured Environments," *International Journal of Robotics Research*, vol. 23, pp. 449-472, September 2003.

[54] J. E. Guivant, "Efficient Simultaneous Localization and Mapping in Large Environments," in *Department of Mechanical and Mechatronic Engineering*. vol. PhD Sydney: University of Sydney, 2002, p. 227.

[55] I. K. Jung, "Simultaneous localization and mapping in 3D environments with stereovision," in *LAAS*. vol. PhD Toulouse: Institut National Polytechnique, 2004, p. 118.

[56] A. J. Davison, "Mobile Robot Navigation Using Active Vision ". vol. D.Phil (Ph.D): University of Oxford, 1998.

[57] N. M. Kwok and G. Dissanayake, "An efficient multiple hypothesis filter for bearing-only SLAM," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)* 2004, pp. 736 - 741.

[58] N. M. Kwok and G. Dissanayake, "Bearing-only SLAM in Indoor Environments Using a Modified Particle Filter," in *Australasian Conference on Robotics & Automation*, Brisbane, 2003.

[59] A. J. Davison, Y. G. Cid, and N. Kita, "Real-Time 3D Slam with Wide-Angle Vision," in *IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, 2004.

[60] N. M. Kwok, G. Dissanayake, and Q. P. Ha, "Bearing-only SLAM Using a SPRT Based Gaussian Sum Filter," in *IEEE International Conference on Robotics and Automation (ICRA 2005)*, 2005, pp. 1109-1114.

[61] J. Sola, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, 2005, pp. 2499-2504.

[62] J. M. M. Montiel, J. Civera, and A. J. Davison, "Unified Inverse Depth Parametrization for Monocular SLAM," in *Robotics: Science and Systems Conference (RSS'06)*, 2006.

[63] A. Davids, "Urban search and rescue robots: from tragedy to technology," *IEEE Intelligent Systems*, vol. 17, pp. 81-83, 2002.

[64] K. Konolige, "Small Vision Systems: Hardware and Implementation," in *Eighth International Symposium on Robotics Research*, 1997

[65] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 920 - 932 September 1994.

[66] T. Kanade, M. Okutomi, T. Nakahara, p, January, and 1992., "A Multiple-baseline Stereo Method," in *DARPA Image Understanding Workshop*, 1992.

[67] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *International Joint Conference on Artificial Intelligence (IJCAI '81)*, Vancouver, BC, Canada,, 1981, pp. 674-679.

[68] D. G. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, Corfu, Greece, 1999, pp. 1150-1157.

[69] D. G. Lowe, "Local feature view clustering for 3D object recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001, pp. 682-688.

[70] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Monona Terrace Convention Center, Madison, Wisconsin, 2003.

[71] L. Matthies and S. S., "Error modeling in stereo navigation," *IEEE Journal of Robotics and Automation,* vol. 3, pp. 239 - 248 June 1987.

[72] D. R. Murray, "Patchlets : a method of interpreting correlation stereo 3D data," in *Department of Computer Science*. vol. Doctor of Philosophy: The University of British Columbia, 2003.

[73] D. C. Herath, K. R. S. Kodagoda, and G. Dissanayake, "Modeling Errors in Small Baseline Stereo for SLAM," in *The 9 th International Conference on Control, Automation, Robotics and Vision (ICARCV 2006)*, Singapore, 2006.

[74] G. Sibley, L. Matthies, and G. Sukhatme, "Bias Reduction and Filter Convergence for Long Range Stereo," in *12th International Symposium of Robotics Research (ISRR 2005)*, San Francisco, CA, USA, 2005.

[75] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988, pp. 147-151.

[76] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector," in *7th European Conference on Computer Vision (ECCV 2002)*, Copenhagen, Denmark, 2002, pp. 128-142.

[77] K. Mikolajczyk and C. Schmid, "Scale & Affine Invariant Interest Point Detectors," *International Journal of Computer Vision,* vol. 60, pp. 63 - 86, October 2004.

[78] R. Mohan, G. Medioni, and R. Nevatia, "Stereo error detection, correction, and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 113-120, 1989.

[79] D. Murray and J. J. Little, "Patchlets: Representing Stereo Vision Data with Surface Elements " in *Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05)* 2005, pp. 192-199.

[80] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 865 - 880 July 2002.

[81] S. Clark and G. Dissanayake, "Simultaneous localisation and map building using millimetre wave radar to extract natural features," in *IEEE International Conference on Robotics and Automation*, 1999, pp. 1316 - 1321.

[82] S. Takezawa, D. C. Herath, and G. Dissanayake, "SLAM in Indoor Environments with StereoVision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Sendai International Center, Sendai, Japan, 2004.

[83] O. Faugeras, *Three-Dimensional Computer Vision : A Geometric Viewpoint*. Cambridge: The MIT Press, 1993.

[84] S. Huang and G. Dissanayake, "Convergence and Consistency Analysis for Extended Kalman Filter based SLAM," *IEEE Transactions on Robotics*, pp. 1036-1049, October 2007.

[85] I.-K. Jung and S. Lacroix, "High resolution terrain mapping using low altitude aerial stereo imagery " in *Ninth IEEE International Conference on Computer Vision (ICCV'03)*, 2003, p. 946.

[86] D. Lerro and Y. Bar-Shalom, "Tracking with debiased consistent converted measurements versus EKF," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 29, pp. 1015-1022, 1993.

[87] L. Mo, X. Song, Y. Zhou, K. S. Zhong, and Y. Bar-Shalom, "Unbiased converted measurements for tracking," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 34, pp. 1023-1027, 1998.

[88] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control,* vol. 45, pp. 477-482, 2000.

[89] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 1615- 1630, October 2005.

[90] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle Adjustment A Modern Synthesis " in *Vision Algorithms: Theory and Practice*, W. Triggs, A. Zisserman, and R. Szeliski, Eds.: Springer Verlag, 2000, pp. 298-375.

[91] P. F. McLauchlan and D. W. Murray, "Active camera calibration for a head-eye platform using the variable state-dimension filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 18, pp. 15-22, 1996.

[92] D. C. Herath, S. Kodagoda, and G. Dissanayake, "Simultaneous Localisation and Mapping: A Stereo Vision Based Approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, Beijing, China, 2006, pp. 922-927.

[93] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM,* vol. 24, pp. 381 - 395, June 1981.

[94] C. Stachniss, D. Hahnel, and W. Burgard, "Exploration with active loop-closing for FastSLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, 2004, pp. 1505-1510.

[95] S. Huang, N. M. Kwok, G. Dissanayake, Q. P. Ha, and G. Fang, "Multi-Step Look-Ahead Trajectory Planning in SLAM: Possibility and Necessity," in *IEEE International Conference on Robotics and Automation (ICRA 2005)*, 2005, pp. 1091-1096.

[96] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation,* vol. 3, pp. 323 - 344, August 1987

[97] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal of Computer Vision,* vol. 47, pp. 7-42, April 2002.