

# **Vision Based Simultaneous Localisation and Mapping for Mobile Robots**

Muhammad Naveed

Université de Bourgogne

Supervisors:  
Dr. David Fofi,  
Dr. Samia Ainouz

A Thesis Submitted for the Degree of  
Erasmus Mundus Masters in Vision and Robotics (VIBOT)  
· 2008 ·



## Abstract

The ability of a robot to localise itself and simultaneously build a map of its environment (Simultaneous Localisation and Mapping (SLAM) or Concurrent Mapping and Localisation (CML)) is a fundamental characteristic required for its autonomous operation. Classically laser and sonar sensors have been used for performing SLAM but during the last decade a significant amount of research has been carried out on SLAM using vision sensors. Vision Sensors/Cameras are low-cost, light and compact, easily available, offer passive sensing, have low power consumption and provide rich information about the environment enabling the detection of stable features. These features make cameras very attractive to be used for SLAM.

Different types of imaging systems have been used to carry out SLAM including single cameras, stereo camera pairs, multiple camera rigs and catadioptric sensors. A single acquisition from a single camera only provides direction of the observed features while one acquisition from a stereo pair can provide 3D location of the observed features. Catadioptric sensors offer a 360° field of view but with non-uniform spatial resolution while multiple camera rigs can provide large fields of view with uniform spatial resolution. Similarly different types of features are extracted from the environment including point features, edge/line features and some SLAM approaches exist which do not explicitly extract any features from the environment. As a SLAM system starts, landmarks for SLAM can be initialised in an undelayed manner as in case of a stereo or multiple camera rig, or by using artificial targets at the start. In contrast, the landmarks are initialised with some delay when a single camera is used to perform SLAM without the use of any artificial target because multiple acquisitions from a single camera are required to compute 3D location of the observed features. Different algorithms have been used to perform SLAM including Extended Kalman Filtering, Particle Filtering, biologically inspired techniques like RatSLAM, and others like Local Bundle Adjustment. Similarly, vision based SLAM has been carried out with and without using the wheel odometry information. Each of the above described approaches to solve different issues involved in performing SLAM have their own pros and cons and one approach can be more suitable for a specific application than others.

A framework for development of a synthetic dataset is presented which can permit the implementation and hence testing and evaluation of different vision based SLAM techniques. Implementation of Extended Kalman Filtering based SLAM on the developed data set is also presented.



# Contents

<b>Acknowledgments</b>	vii
<b>1 Introduction</b>	1
<b>2 Problem Definition</b>	3
2.1 Motivation and Need	3
2.2 Problem Step 1: Study of State-of-the-Art Techniques	3
2.3 Problem Step 2: Development of a Dataset	4
2.4 Problem Step 3: Implementation of Visual SLAM Techniques	4
2.5 Problem Definition in a Nutshell	4
2.6 Aims of the Thesis	5
2.7 Tasks/Objectives	5
<b>3 State-of-the-Art</b>	7
3.1 Imaging Systems	7
3.1.1 Single Camera	8
3.1.2 Stereo Camera Pair	8
3.1.3 Multiple Camera Rigs	9
3.1.4 Catadioptric Sensors	9
3.2 Features from Environment	10
3.2.1 Point Features	10
3.2.2 Line/Edge Features	12
3.2.3 Featureless Approaches	12
3.3 Initialisation of Landmarks	13
3.3.1 Undelayed Approaches	14
3.3.2 Delayed Approaches	14
3.4 SLAM Techniques	15
3.4.1 Extended Kalman Filtering (EKF)	15
3.4.2 Particle Filtering	17

3.4.3	RatSLAM .....	18
3.4.4	Local Bundle Adjustment .....	19
3.5	Use of Wheel Odometry .....	21
3.6	Discussion .....	21
3.6.1	Monocular and Stereo Imaging Systems .....	21
3.6.2	Interest Point Detectors .....	22
3.7	Summarising Table .....	23
<b>4</b>	<b>Methodology</b> .....	<b>25</b>
4.1	Requirement of a Visual SLAM Dataset .....	25
4.1.1	Significance of the Dataset .....	25
4.1.2	Basic Characteristics of the Dataset .....	25
4.1.3	Feature Types Required to be Present in the Dataset .....	26
4.1.4	Flexibility in terms of Feature Noise .....	26
4.2	Development of the Dataset .....	26
4.2.1	Epipolar Geometry Toolbox (EGT) .....	26
4.2.2	Steps Involved in Dataset Development .....	27
4.3	Implementation of Vision Based EKF SLAM .....	30
4.3.1	Aims .....	30
4.3.2	Assumptions .....	30
4.3.3	Sensor Noise .....	31
4.3.4	Steps Involved in EKF Based SLAM Implementation .....	31
4.4	Problems Faced .....	34
<b>5</b>	<b>Results</b> .....	<b>35</b>
5.1	Dataset .....	35
5.1.1	Results .....	35
5.1.2	Comments .....	37
5.2	EKF Based SLAM Implementation .....	37
5.2.1	Sensor Noise Variances .....	37
5.2.2	Results .....	38
5.2.3	Comments .....	40
<b>6</b>	<b>Conclusions</b> .....	<b>41</b>
6.1	Conclusion .....	41

6.2 Significance of the Work .....	42
6.3 Future Works .....	42
<b>Bibliography</b>	<b>43</b>

# List of Figures

3.1. Robot mounted with a stereo camera pair used by [9] . . . . .	8
3.2. Eight camera rig used by [11] . . . . .	9
3.3. Single camera double mirror catadioptric sensor (left) and an acquired image (right), used by [14]. . . . .	10
3.4. Stereo panoramic images by [14] . . . . .	10
3.5. Acquired image (left), greyscale image and selected sub-window (middle), and the corresponding image array (right), by [4] . . . . .	13
3.6. Artificial target with four known features, used by [7] . . . . .	14
3.7. RatSLAM structure by [21] . . . . .	18
3.8. Local bundle adjustment on addition on camera $C_i$ , by [2] . . . . .	20
3.9. Evolution of robot pose uncertainty for (left) stereo and (right) monocular approaches, by [9] . . . . .	21
3.10. Repeatability of detected features with changes in viewpoint (left) and scale (right), by [16] . . . . .	23
4.1. Two views of the 3D structure (all units in mm) . . . . .	27
4.2. Three views of ground truth stereo camera pair trajectory (green), (all units in mm) . . . . .	28
4.3. Structure of State Covariance Matrix $P$ . . . . .	33
5.1. Robot trajectory top view (all units in mm) . . . . .	35
5.2. Dataset images for time instants 146-147 for the two cameras . . . . .	36

5.3. Ground truth trajectory of the robot in 3D environment (all units in mm) . . . . .	37
5.4. Two views of ground truth (green) and estimated (blue) robot trajectories (all units in mm). . . . .	38
5.5. Ground truth (green), estimated (blue) and 3xSigma bounds (red) for x, y and z coordinates of robot position for first 25 time stamps . . . . .	39

# List of Tables

3.1. Classification of state-of-the-art Visual SLAM . . . . . 24

5.1. Overall mean absolute estimation errors . . . . . 39

# Acknowledgments

I am very thankful to my supervisor Dr. David Fofi for his ideas, technical guidance, encouragement, always listening to me about all my problems and issues, and his precious advice which helped me a great deal. I am also very thankful to my co-supervisor Dr. Samia Ainouz for all her advice, ideas, and helping me on technical and general issues throughout the course of the thesis.

I'd like to thank my friends Bushra, Arsalan, Saleh and Wajahat for their care, chit-chats, dinners and the technical help they provided whenever I wanted. Thanks to my friends Sang, Arun, Kaikai, Indy, Alonso, Yousuf and Masoud for their company and support. I would also like to thank Thomas, Nathalie, Mathias, Iqbal, Thierry, Nicholas, Benjamin, Rindra, Pierre-Emanuele, Ionut and Jerome for their company in coffee breaks and all their help in the lab.

I'm grateful to Prof. Bernard Lamalle, Fabrice Marieaudeau, my supervisor David Fofi, Joaquim Salvi, Joan Marti, Xevi Cufi, Robert Marti, Judith Bell, Yvan Petillot, Emanuele Trucco, David Lane, Katia Labert, Alice Meriaudeau, Valérie Torrès, Sandra, Montse Vila, Lola, Adriana Grosu and all other professors and staff members involved in the idea and realisation of VIBOT masters course for providing me this excellent opportunity and helping me all the way along the two years of my masters studies. Special thanks to European Commission for providing me the funding and opportunity to study in Europe through their Erasmus Mundus program.

Finally I'd like to thank my mum, dad, sister Saadia, brothers Usman and Salman and other family members for always helping and supporting me and all the care and love.



# Chapter 1

## Introduction

Simultaneous Localisation and Mapping (SLAM) is the problem of a robot being autonomously able to build a map of an unknown environment and simultaneously localise itself in the environment. This ability makes a robot truly autonomous.

While performing SLAM, the robot observes the environment around it and detects the position of some features in the environment. Some of the detected features serve as landmarks for the SLAM process. The estimation of positions of these landmarks constitutes the *mapping* part of SLAM process. As the robot moves, it again observes the landmarks. Currently observed landmarks are then matched with the previously known landmarks and discrepancy between the expected and currently measured positions of landmarks is used to adjust the estimate of robot position, this is the *localisation* part of SLAM.

Classically, laser and sonar sensors were used for the perception of environment and thus performing SLAM. However, the situation is changing rapidly and during last decade a considerable amount of research has been carried out on SLAM using vision sensors. Visual Sensors/Cameras provide rich information about the environment enabling the detection of stable features. Furthermore, cameras are low-cost, light and compact, easily available, offer passive sensing and have low power consumption. All these features make cameras very attractive to be used for SLAM.

This thesis presents a study of different vision based simultaneous localisation and mapping techniques and an implementation of one of the techniques on a synthetic dataset. The thesis is structured as follows:

- **Chapter 2** defines the problem of the study and formulates the aims and objectives of the thesis.

- **Chapter 3** provides current state-of-the-art of visual SLAM with pros and cons of different techniques used to solve different issues involved in the problem of visual SLAM.
- **Chapter 4** gives the methodology followed to realise the objectives of the thesis study.
- **Chapter 5** presents the results for experiments conducted during the thesis.
- The thesis concludes in **Chapter 6** stating the significance of thesis work and possible future works.

# Chapter 2

## Problem Definition

### 2.1. Motivation and Need

Different approaches exist in the field of Vision based Simultaneous Localisation and Mapping (SLAM) in order to solve different steps involved in performing SLAM. For instance, many different types of imaging systems can be used to carry out visual SLAM ranging from single camera to stereo catadioptric sensors. Similarly different types of features can be extracted from the environment ranging from point features to edge and planar features and different approaches exist to find the correspondences between the extracted features. Several mathematical techniques exist to solve the SLAM problem ranging from Extended Kalman Filtering and Particle Filtering to biologically inspired techniques like RatSLAM. These multiple possibilities for solving different issues in the problem of vision based SLAM have their own pros and cons, and therefore, a technique can be more suitable for a specific applications than others.

### 2.2. Problem Step 1: Study of State-of-the-Art Techniques

Existence of the above mentioned variety of approaches available to solve the issues involved in visual SLAM creates the necessity of a thorough and comprehensive study of state-of-the-art visual SLAM approaches. This state-of-the-art study can permit the identification of advantages and disadvantages of each approach over the others. The study can also serve as the basis for further research in the field of visual SLAM in an efficient and application specific way.

### **2.3. Problem Step 2: Development of a Dataset**

After a comprehensive state-of-the-art study, the subsequent requirement is the availability of a dataset which can be used to implement, test and evaluate different visual SLAM approaches. This dataset should consist of images taken along a trajectory in a 3D environment along with ground truth data i.e. robot's trajectory with orientation and velocity information, and camera intrinsic parameters etc. This dataset should be complete and relatively simpler in order to permit rapid implementation and testing of algorithms.

Despite the fact that a significant amount of research has been carried out on visual SLAM in last decade, no suitable dataset is freely available that can be used to test different visual SLAM implementations including monocular and stereo camera approaches. This leads to the problem of design and development of one such dataset.

### **2.4. Problem Step 3: Implementation of Visual SLAM Techniques**

After a comprehensive study of existing vision based SLAM techniques and development of a dataset, subsequent problem is the implementation of one or more vision based SLAM approaches. This implementation will lead to better understanding of merits and limitations of the approaches, and hence their suitability for a specific application.

### **2.5. Problem Definition in a Nutshell**

In a nutshell, the problem of the thesis can be defined as (i) study and compilation of a comprehensive state-of-the-art of current visual SLAM techniques, (ii) development of a complete and relatively simpler dataset which can be used for fast implementation, testing and evaluation of different vision based SLAM techniques and (iii) implementation of a visual SLAM technique on the above mentioned dataset in order to study its merits and limitations.

The above problem definition leads to the formal definition of the aims and objectives of the thesis as follows.

## **2.6. Aims of the Thesis**

The aim of the thesis project is first to study and compile a survey of state-of-the-art visual SLAM techniques. A dataset is to be developed in order to enable the implementation and evaluation of different visual SLAM techniques. Then, a visual SLAM technique is to be implemented on the developed dataset in order for the technique to be critically evaluated.

## **2.7. Tasks/Objectives**

Following is the list of tasks/objectives for the thesis project:

1. A comprehensive study of current visual SLAM techniques.
2. Compilation of current state-of-the-art on visual SLAM.
3. Development of a complete dataset to permit the implementation, testing and evaluation of visual SLAM techniques.
4. Implementation of a visual SLAM technique on the above mentioned dataset.



# Chapter 3

## State-of-the-Art

A significant amount of research has been carried out on vision based SLAM during the last decade. A number of steps are involved in vision based SLAM process. This includes perception of environment using a vision sensor, extraction of features from the perceived data, selection of some features and their initialisation as landmarks which are then used during the SLAM process, implementation of an estimation algorithm etc. This chapter gives a classification of state-of-the-art vision based SLAM techniques.

The classification is based on the following criteria: (i) *imaging systems* used for performing SLAM which include single cameras, stereo pairs, multiple camera rigs and catadioptric sensors, (ii) *features extracted* from the environment in order to perform SLAM which include point features and line/edge features, (iii) *initialisation of landmarks* which can either be undelayed as in the case of stereo or multiple camera rigs (because they can provide direction and depth of the detected features) or delayed as in the case of a single camera (because single acquisition can only provide direction of the detected features), (iv) *SLAM techniques* used which include Extended Kalman Filtering, Particle Filtering, biologically inspired techniques like RatSLAM, and other techniques like Local Bundle Adjustment, and (v) *use of wheel odometry information*. A comparison between some of the above mentioned approaches is given in the Discussion Section.

### 3.1. Imaging Systems

Different imaging systems have been used for visual SLAM including single cameras, stereo camera pairs, multiple camera rigs and catadioptric sensors. Each of these imaging systems have some advantages and disadvantages and a specific type of imaging system might be more suitable for a particular application. The pros and cons of each of the above mentioned imaging

systems have been discussed below with pointers to some of the published implementations.

### 3.1.1. Single Camera

Single Camera SLAM is also referred to as Bearing-only SLAM as a single image provides only the direction of features present in robot's environment and does not provide the depth information. To get the 3D location of a feature, multiple images from different viewpoints are required. Some visual SLAM implementations using single cameras are [1], [2], [3], [4], [5] and [6]. Wide-angle cameras (above 90° field of view) have also been used for visual SLAM as in [7] and [8], as these cameras enable the tracking of features over wider motion ranges. Most of the single camera SLAM implementations mentioned above are for indoor environments. [2] and [9] show implementations of single camera visual SLAM in outdoor environment.

### 3.1.2. Stereo Camera Pair

A stereo camera pair can provide 3D location of the observed features in the environment (using triangulation); this makes a stereo pair readily usable for visual SLAM. However the feature matching problem in case of stereo is slightly more complicated than in the case of single camera [9]. The reason for this increase in complexity is that in case of stereo pair, the features have to be matched between the two images from the stereo pair, and then between consecutive acquisitions in time [9].

[9] gives an implementation of visual SLAM using stereo pair for ground and aerial robots. Fig. 3.1 shows the robot mounted with a stereo camera pair used by [9]. [10] also shows an implementation of visual SLAM using stereo pair.



Figure 3.1. Robot mounted with a stereo camera pair used by [9].

### 3.1.3. Multiple Camera Rigs

Multiple camera rigs have also been used for visual SLAM. One advantage is that the use of multiple cameras increase the field of view and enables the tracking of features over wider robot motion. Another advantage is that the spatial resolution over the field of view of a multiple camera rig is uniform unlike the catadioptric sensors which also offer a large field of view.

Using multiple camera rigs can also provide better constraints for reconstruction of the environment compared to single cameras or stereo pairs [11]. Moreover if a multiple camera rig has the cameras directed both frontwards and backwards, the robot can easily be driven in both the directions while performing SLAM [11]. One disadvantage of using multiple camera rigs is the high computational cost.

[11] have used an eight camera rig which offers 360° field of view to carry out visual SLAM, without making any assumptions like regular spacing between the cameras or overlapping between the views. Fig 3.2 shows the eight camera rig used by [11].



Figure 3.2. Eight camera rig used by [11].

### 3.1.4. Catadioptric Sensors

Catadioptric sensors are attractive for application in visual SLAM because they offer a wide field of view. Catadioptric sensors have been used in different formations for visual SLAM. [12] shows implementation using a single catadioptric sensor mounted on a ground robot. [13] have used two catadioptric sensors as a stereo pair.

[14] use a catadioptric sensor consisting of a single camera and two fixed conic mirrors as shown in Fig 3.3 (left). This type of sensor provides two views of the scene in a single image. The advantage of using this type of sensors (instead of two catadioptric sensors as a stereo pair)

is that corresponding points in two views of the scene lie on radial lines in the image reducing the complexity of stereo matching process. An image obtained by this kind of sensor is shown in Fig 3.3 (right) and Fig 3.4 shows two stereo panoramic images obtained during single acquisition by the sensor.

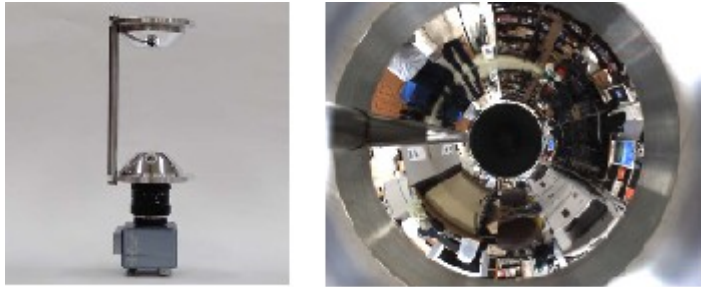


Figure 3.3. Single camera double mirror catadioptric sensor (left) and an acquired image (right), used by [14].



Figure 3.4. Stereo panoramic images by [14].

## 3.2. Features from Environment

In order to carry out SLAM using vision, features from the environment have to be extracted that can be used as landmarks for SLAM. These features have to be stable and observable from different view points and angles. Many types of environment features have been used for SLAM using vision. There are also some SLAM approaches which do not extract any specific features from the environment. Pros and cons of some important feature types that can be extracted from the environment in order to perform SLAM are given below, along with a description of featureless SLAM approaches.

### 3.2.1. Point Features

Point features are the most commonly used features for visual SLAM. Harris corner detector has widely been used in the recent years for interest point detection as in [2], [11], [12] and [9]. [15]

suggest that Harris corner detector is the most suitable interest point detector for visual SLAM.

To apply Harris corner detector to an image, the image gradients  $I_x$  and  $I_y$  in  $x$  and  $y$  directions respectively are calculated for each pixel. Then a matrix  $C$  is calculated at each pixel using an image patch around the pixel:

$$C(x, y) = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{pmatrix} \quad (\text{Eq 3.1})$$

Let  $\lambda_1$  and  $\lambda_2$  be the Eigen values of matrix  $C$ , an auto-correlation function  $R$  is defined as:

$$R = \lambda_1 \lambda_2 - k (\lambda_1 + \lambda_2)^2 \quad (\text{Eq 3.2})$$

where  $k$  is an empirical constant. Sharply peaked values of  $R$  represent corners in the image under consideration. [16].

[1] use Harris-Laplace detector for the detection of interest points. Harris-Laplace detector uses a scale adapted Harris function to detect interest points [16]. The detection operator of Shi and Tomasi has been used for interest point detection in [7] and [8].

Scale-Invariant Feature Transform (SIFT) has also been used for interest point detection for visual SLAM as in [3], [14] and [17]. SIFT features are invariant to scaling, translation and rotation and partially invariant to illumination changes and 3D projection [18].

In order to extract SIFT features from an image, a difference of Gaussian function is applied to the image in scale-space. This can be done by building an image pyramid with resampling between each level. Maxima and minima of the difference of Gaussian function are selected as key locations. These maxima and minima are detected first by comparing each pixel by its eight neighbours at the same level of the pyramid and then by comparing it with the closet pixels at the neighboring levels. Image gradients and orientations are then computed at each key location at each level of the smoothed images. For a level  $A$ , the image gradient  $M_{ij}$  and orientation  $R_{ij}$  at pixel  $A_{ij}$  are computed as follows:

$$M_{i,j} = \sqrt{(A_{i,j} - A_{i+1,j})^2 + (A_{i,j} - A_{i,j+1})^2} \quad (Eq3.3)$$

$$R_{i,j} = \tan^{-1}(A_{i,j} - A_{i+1,j}, A_{i,j+1} - A_{i,j}) \quad (Eq3.4)$$

Gradient magnitudes are then thresholded at a value of 0.1 times the maximum possible gradient value. A canonical orientation is then assigned to each of the key locations. This orientation is the peak value in a histogram of local image orientations. Local image descriptors are then calculated based on location, scale and orientation information for the key location. [18].

As the robot moves, it has to (i) associate the newly observed features with the previously seen features and (ii) identify new features in order to create new landmarks when required. This data association is a crucial stage in SLAM using any type of sensors because wrong data association can lead to extremely erroneous localisation and mapping. Once the interest points have been detected using an interest point detector, the data association can be performed using different methods. [7] and [8] use normalised cross-correlation between planar patches around interest points for data association. [10] and [12] use local groups of interest points to perform robust data association.

### 3.2.2. Line/Edge Features

Line/Edge features exist in abundance in structured environments. This type of features is more useful for mapping than point features as they also provide some geometrical information about the environment. Moreover, these features are invariant to lighting [6] and significant viewpoint changes [5]. [5] use Plücker coordinates to represent line features in the environment and use them for carrying out SLAM. [6] use Canny's detector to extract edges from the images and use portions of these edges as features which they name *edgelets*. The advantage of using edgelets over using complete edges is that the complete edges can at times be partially occluded or broken into multiple edges in the image [6].

### 3.2.3. Featureless Approaches

SLAM using vision has also been carried out without explicitly extracting any features from the environment. [4] have presented a technique in which they take 640x480 image, convert it into greyscale, take a 300x160 pixels sub-window at centre of the image and compute an *image*

array by summing the pixel values in each column of the sub-window. Fig 3.5 shows an acquired image, the image converted into greyscale and the selected sub-window at the centre of the image, and the corresponding image array.



Figure 3.5. Acquired image (left), greyscale image and selected sub-window (middle), and the corresponding image array (right), by [4].

By comparing the shift in image arrays of consecutive frames, the rotation and speed of the moving camera is extracted as follows:

Rotation  $\Delta\theta$ :

$$f(s) = \frac{1}{w - |s|} \sum_{n=1}^{w - |s|} |I_{n + \max(s, 0)}^{k+1} - I_{n - \min(s, 0)}^k| \quad (\text{Eq 3.5})$$

$$\Delta\theta = \alpha(\arg \min f(s)) \quad (\text{Eq 3.6})$$

where  $I$  represents the image array values for images  $k$  and  $k+1$ ,  $w$  is the image width,  $s$  is the shift between two image arrays and  $\alpha$  is an empirically determined gain constant.

Speed  $v$ :

$$v = \frac{1}{w - |s_m|} \sum_{n=1}^{w - |s_m|} |I_{n + \max(s_m, 0)}^{k+1} - I_{n - \min(s_m, 0)}^k| \quad (\text{Eq 3.7})$$

where  $s_m$  is the image array shift corresponding to the best rotation match. [4].

### 3.3. Initialisation of Landmarks

To carry out SLAM, some of the extracted features from environment are used as landmarks. In

order to be used as landmarks, both direction and depth information of the feature has to be estimated. In general two types of landmark initialisation approaches exist.

### 3.3.1. Undelayed Approaches

To get direction and depth information of a feature right at initialisation of the SLAM system, one way is to use a stereo pair. [9] show an implementation of SLAM using stereo vision for ground and aerial robots. Another way of getting an undelayed initialisation of landmarks is to use an artificial target of known appearance for the SLAM system to initialise. [7] use a solid rectangle printed on a paper as initialisation target for the SLAM system. Fig 3.6 shows the matching of four known features of an artificial target in order to get undelayed initialisation of landmarks in [7]. The featureless approach of [4] explained in sub-section 3.2.3 is also a special case in the category of undelayed approaches.



Figure 3.6. Artificial target with four known features, used by [7].

### 3.3.2. Delayed Approaches

When a single camera is used to carry out SLAM without the aid of any artificial target at start-up, determining the depth of features detected in the first acquired frame is not possible, using the first frame alone. In this case the camera is moved to slightly different view points and corresponding features are matched in different frames. This enables estimating the depth of some features which are then used as landmarks to initialise the SLAM system. [1], [2] and [3] give implementations of delayed landmark initialisation using multiple frames from single camera.

### 3.4. SLAM Techniques

Different SLAM techniques exist and have been implemented using visual sensors. These include the well known *Extended Kalman Filtering*, *Particle Filtering* and some other techniques. Four SLAM techniques are explained below including *Extended Kalman Filtering*, *Particle Filtering*, *RatSLAM* and *Local Bundle Adjustment*.

#### 3.4.1. Extended Kalman Filtering (EKF)

As the robot performs SLAM, at a time instant  $k$ , let  $x_k$  be the vector representing the current robot state (position and orientation),  $u_k$  be the control input applied at time  $k-1$  to move the robot to state  $x_k$  (this can also be the wheel odometry readings during the interval  $(k-1, k]$ ). Let  $m$  be the set representing the locations of all landmarks and  $z_k$  be the set of landmark observations at time instant  $k$ .

In EKF, the motion model (model that gives the robot state  $x_k$  from state  $x_{k-1}$  and control input  $u_k$ ) is described as:

$$x_k = f(x_{k-1}, u_k) + w_k \quad (\text{Eq 3.8})$$

where function  $f$  models the robot kinematics and  $w_k$  accounts for the un-modelled kinematics and noise and is considered to be zero mean uncorrelated Gaussian noise with covariance  $Q_k$ . Similarly the observation model (the model for observation of landmarks  $m$  from robot at state  $x_k$ ) is described as:

$$z_k = h(x_k, m) + v_k \quad (\text{Eq 3.9})$$

where function  $h$  describes the observation geometry and  $v_k$  accounts for the observation errors and is zero mean uncorrelated Gaussian noise with covariance  $R_k$ . [19].

EKF is performed in two steps, first the robot motion is predicted using control input and then updated using the landmarks observation. For both the steps, the sets representing the mean of robot state ( $x_k$ ) and landmark locations ( $m_k$ ) are calculated, and a matrix  $P$  representing covariances within and between robot state and landmark locations is also calculated.  $P$  is structured as:

$$P_{k|k} = \begin{bmatrix} P_{xx} & P_{xm} \\ P_{xm} & P_{mm} \end{bmatrix}_{k|k} \quad (Eq 3.10)$$

At time instant  $k$  the prediction step is performed as follows:

$$\bar{x}_{k|k-1} = f(\bar{x}_{k-1|k-1}, u_k) \quad (Eq 3.11)$$

$$\text{and } P_{xx, k|k-1} = \nabla f P_{xx, k-1|k-1} \nabla f^T + Q_k \quad (Eq 3.12)$$

where  $\nabla f$  represents the Jacobian of  $f$  calculated at the estimate  $\bar{x}_{k-1|k-1}$ . The update step is performed as follows:

$$\begin{bmatrix} \bar{x}_{k|k} \\ \bar{m}_k \end{bmatrix} = \begin{bmatrix} \bar{x}_{k|k-1} \\ m_{k-1}^- \end{bmatrix} + W_k [z_k - h(\bar{x}_{k|k-1}, m_{k-1}^-)] \quad (Eq 3.13)$$

$$\text{and } P_{k|k} = P_{k|k-1} - W_k S_k W_k^T \quad (Eq 3.14)$$

where:

$$S_k = \nabla h P_{k|k-1} \nabla h^T + R_k \quad (Eq 3.15)$$

$$W_k = P_{k|k-1} \nabla h^T S_k^{-1} \quad (Eq 3.16)$$

$\nabla h$  is the Jacobian of  $h$  calculated at  $\bar{x}_{k|k-1}$  and  $\bar{m}_{k-1}$ . [19].

$[z_k - h(\bar{x}_{k|k-1}, \bar{m}_{k-1})]$  is called the *innovation* and it represents the difference between observation and prediction,  $S_k$  is the *innovation covariance* and  $W_k$  is the Kalman Gain.

By far EKF is the most used SLAM technique. One problem with EKF techniques is that their computational cost grows quadratically with the number of landmarks. Secondly they use linearized models of non-linear motion and observation models. Some implementations of EKF for visual SLAM can be found in [1], [7], [5], [8], [12], [9] and [13].

### 3.4.2. Particle Filtering

Particle Filters have also been successfully employed in visual SLAM. The FastSLAM algorithm introduced by [20] used particle filter to estimate robot pose and EKF for estimating landmark locations.

The FastSLAM algorithm is based on the fact that in SLAM problem if robot's pose is known, the individual landmark measurements are independent. In other words, if the robot poses are known, the estimation of landmark locations can be decoupled into independent estimation problems for each of the landmarks. [20].

This leads to the factorisation of the combined SLAM state as follows:

$$\begin{aligned} P(X_{0:k}, m | Z_{0:k}, U_{0:k}, x_0) \\ = P(m | X_{0:k}, Z_{0:k}) P(X_{0:k} | Z_{0:k}, U_{0:k}, x_0) \end{aligned} \quad (Eq 3.17)$$

where  $X_{0:k}$ ,  $Z_{0:k}$  and  $U_{0:k}$  represent the sets of all robot states, observations and control inputs respectively from time 0 to  $k$ . [19].

At the time instant  $k$ , the joint distribution is represented by the set:

$$\{w_k^{(i)}, X_{0:k}^{(i)}, P(m | X_{0:k}^{(i)}, Z_{0:k})\}_i^N \quad (Eq 3.18)$$

where  $N$  is the total number of particles,  $w_k^{(i)}$  is the importance weight given to the  $i$ th particle and

$$P(m | X_{0:k}^{(i)}, Z_{0:k}) = \prod_j^M P(m_j | X_{0:k}^{(i)}, Z_{0:k}) \quad (Eq 3.19)$$

where  $M$  is the total number of landmarks. [19].

For robot state at time instant  $k$ , each particle calculates the new robot state using the state at time  $k-1$  and control input  $u_k$ . This creates a temporary set of particles. This set is sampled by giving different weights to different particle, which in turn gives the final set of particles representing the robot state at time instant  $k$ . For each particle, each landmark is updated using a

separate Kalman filter. [20].

[3], [6] and [17] perform visual SLAM using techniques based on particle filtering.

### 3.4.3. RatSLAM

RatSLAM is a SLAM technique based on the model of rodent hippocampus. Rodents have *place fields* which are patterns of neural activity that correspond to locations in space, and are modulated by rodent motion and visual sensing. This technique uses a competitive attractor network as the approximation of the rodent hippocampus. Activity packets in the attractor network represent pose hypotheses. The attractor network is called *pose cells*. Wheel odometry information is used to inject activity in pose cells and thus shifting the activity packets, this is the process of “path integration”. Visual sensing information is converted into *local view* representation. If the current visual scene is familiar, it also injects activity into the pose cells that are linked to current scene. [21]. The RatSLAM structure is shown in Fig. 3.7.

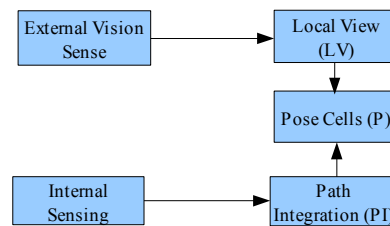


Figure 3.7. RatSLAM structure by [21].

A pose cell unit excites the units close to it and inhibits the units that are far from it. This process leads to the dominance of an activity packet. When activity is injected close to the dominating activity packet, the activity tends to move the packet towards itself. In contrast, the activity injected far from the dominating packet creates a new packet which competes with the dominating packet and can eventually become the dominating packet. [21].

During the path integration process, when the robot translates, the pose cell activity is shifted in  $x,y$  plane and magnitude of this shift depends on translational velocity of the robot. Similarly, if the robot rotates, the activity is shifted in  $\theta$  direction and the magnitude of shift depends on the rotational velocity of the robot. [21].

The visual sensing information is encoded in the *local view cells*. Association between the active local view cells and highly active pose cells is strengthened by changing the weight of connections between them. This is done using Hebbian learning, and is expressed as follows:

$$\beta_{(ijk)(lmn)}^{t+1} = \beta_{(ijk)(lmn)}^t + \eta (P_{lmn} V_{ijk}) \quad (\text{Eq 3.20})$$

where  $\beta$  is the strength of connection between local view cell and pose cell,  $\eta$  is the learning rate,  $V$  and  $P$  are the activation levels of the local view and pose cells respectively, and  $ijk$  and  $lmn$  represent the spaces in which local view and pose cells are represented respectively. [21].

This SLAM technique has been used in the featureless implementation of [4].

#### 3.4.4. Local Bundle Adjustment

SLAM is similar to the problem of *Structure from Motion (SFM)* where the movement of a camera and the positions of the observed points are estimated. There are two types of SFM algorithms. The methods that fall into the first type perform computationally expensive global bundle adjustment optimisation, an example is [22]. Because of high computation time these algorithms are off-line and are not feasible for real-time applications in SLAM. Other type of methods are fast as they do not perform a global optimisation, and hence are suitable for on-line applications. One such method has been described in [23] for monocular and stereo camera cases. Problem with these methods is that they accumulate error with time [2].

[2] have proposed a method which uses *fast and local* bundle adjustment in order to carry out SLAM in real-time using a single camera. When the SLAM system initialises, three acquired frames are used to set up the global coordinate system. The system uses Harris corners as interest points and the points are matched between frames by computing Zero Normalised Cross Correlation in the regions of interest. As new frames are acquired during the SLAM process, some frames are selected as *key-frames*. In order to understand the camera pose estimation process, consider at a point in time when we have already calculated the camera poses  $C_l$  to  $C_{i-1}$  which correspond to the key-frames  $I_l$  to  $I_{i-1}$ . A set of points and their projections in the corresponding images are also known. Now a new frame  $I$  is acquired and we have to estimate the corresponding camera pose  $C$ . Frame  $I$  is matched with last key-frame  $I_{i-1}$  to find a set of points whose projections on the cameras  $C_{i-2}$ ,  $C_{i-1}$  and  $C$  are known and whose 3D coordinates

have already been computed earlier. Now the pose  $C$  is estimated using a pose estimation algorithm and RANSAC which is then refined using a fast LM optimization. [2].

As the new frames are acquired, if the number of matching points between the current frame and last key-frame is less than a threshold, preceding frame (the frame acquired before the current frame) is selected as a new key-frame. Similarly when the uncertainty of estimated position is very high, a new key-frame is added to the system. As the new key-frame  $I_i$  is added, new points which are observed only in the current three key-frames are reconstructed using triangulation. At the addition of the new key-frame, a local bundle adjustment is carried out by Levenberg-Marquardt minimisation of the cost function  $f^i(C^i, P^i)$  where  $C^i$  and  $P^i$  are the camera poses and 3D points selected for the current optimisation stage (see Fig. 3.8). For this stage  $n$  last cameras and  $N$  last frames are used where  $N$  is greater than or equal to  $n$ .  $C^i$  represents the set  $\{C_{i-n+1} \dots C_i\}$  and  $P^i$  is the set of all 3D points projected on cameras in set  $C^i$ . The function  $f^i$  is given by:

$$f^i(C^i, P^i) = \sum_{C_i \in \{C_{i-N+1} \dots C_i\}} \sum_{P_j \in P^i} (\varepsilon_{ij}^2) \quad (Eq 3.21)$$

$$\text{with } \varepsilon_{ij}^2 = d^2(p_{ij}, K_i p_j) \quad (Eq 3.22)$$

where  $\varepsilon_{ij}^2$  is the squared Euclidean distance between the estimated projection of point  $p_j$  on camera  $C_i$  and the corresponding measured position.  $K_i$  is the  $i$ th projection matrix consisting of corresponding extrinsic and intrinsic parameters. [2].

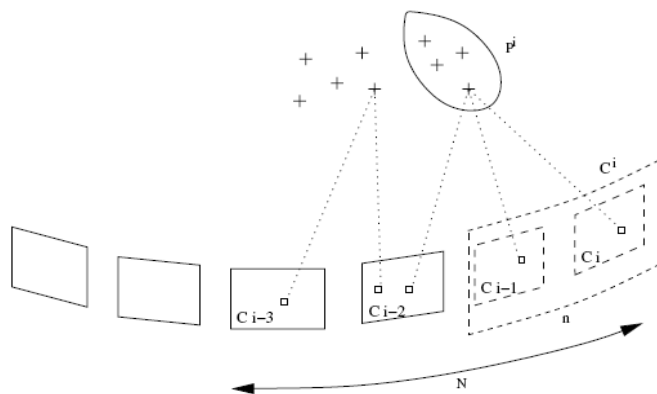


Figure 3.8. Local bundle adjustment on addition of camera  $C_i$ , by [2].

[2] have found that  $n = 3$  or  $4$  and  $6 \leq N \leq 11$  are sufficient values for the above local bundle adjustment.

### 3.5. Use of Wheel Odometry

Classically SLAM is carried out using both the wheels odometry information and data from other sensors sensing the environment. Process uncertainties account for inaccuracies in odometry data and other noise. [7] suggest that even if no wheel odometry is available, the whole camera motion can be modelled as process uncertainty or noise. Many visual SLAM implementations do not use the odometry data to carry out SLAM including [2], [4], [6], [7], [8] and [17], whereas many implementations use wheel odometry along with other sensors to carry out SLAM as in [1], [3], [5] and [11].

### 3.6. Discussion

#### 3.6.1. Monocular and Stereo Imaging Systems

[9] have experimented and compared monocular and stereo approaches to visual SLAM for ground and aerial robots. Fig. 3.9 taken from [9] shows the evolution of robot pose for stereo and monocular approaches with camera(s) mounted frontwards and sidewards on a ground robot that takes three loops on a 6m diameter circular trajectory.

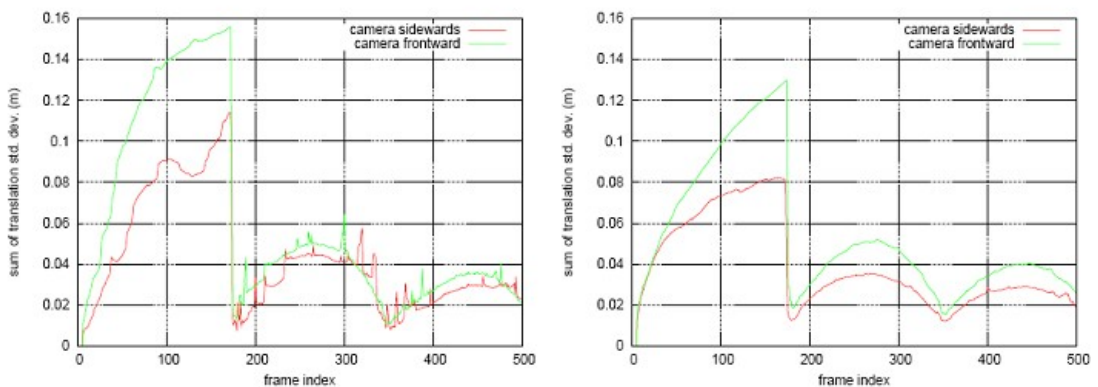


Figure 3.9. Evolution of robot pose uncertainty for (left) stereo and (right) monocular approaches, by [9].

From Fig. 3.9 it can be observed that the uncertainty significantly drops at the first loop closure. It can also be observed that the uncertainties are lower for cameras mounted sideways. This is because of the fact that in sideways case the features are tracked on more frames. Moreover, in sideways case for monocular camera, the base-line between the two consecutive frames is greater which results in fast initialisation of landmarks. Here monocular SLAM approach seems to slightly outperform the stereo approach in terms of uncertainty in robot pose. This is because in case of stereo, the feature matching is more complicated because they are to be matched between two images from the stereo pair and also between two consecutive acquisitions in time. This results in fewer matches in some frames in the stereo case. In contrast, the matching problem is relatively less complicated in monocular case because the features are only to be matched between consecutive acquisitions in time. [9]. In terms of consistency of robot pose estimate, [9] suggest that bearing-only SLAM (monocular approach) does not perform as good as stereo SLAM.

### 3.6.2. Interest Point Detectors

[16] have studied the suitability of different interest point detectors for the application in visual SLAM. They have experimentally compared the robustness of five interest point detectors i.e. Harris, Harris-Laplace, SUSAN (Smallest Univalued Segment Assimilating Nucleus), SIFT (Scale Invariant Feature Transform) and SURF (Speeded Up Robust Features), against changes in viewpoint and scale. In their experiment, [16] acquired 12 image sequences each containing 21 images with a viewpoint change of  $2.5^\circ$  between every two consecutive frames; and to study the robustness against change in scale they acquired 14 image sequences each containing 12 images where camera moved 0.1m between every two consecutive acquisitions.

Fig. 3.10 (left) shows the repeatability of the features detected in the first frame, in the proceeding frames as the viewpoint changes gradually. Harris corner detector outperforms other interest points detectors as it is able to maintain 30% of the initially detected features till the last frame with  $50^\circ$  change in viewpoint. Similarly Fig. 3.10 (right) shows repeatability of features with change in scale. In this case, Harris corner detector also outperforms other interest point detectors.

### 3.7. Summarising Table

Table 3.1 gives a classification of state-of-the-art visual SLAM techniques.

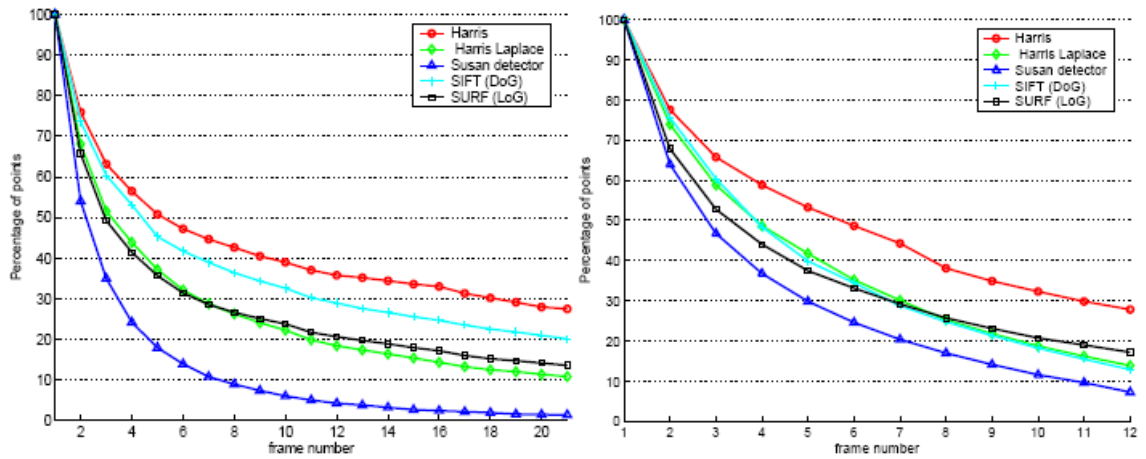


Figure 3.10. Repeatability of detected features with changes in viewpoint (left) and scale (right), by [16].



# Chapter 4

## Methodology

This chapter describes the methodology followed in order to achieve the project's tasks of designing and developing a dataset for visual SLAM and implementation of a visual SLAM technique on this dataset (tasks 3 and 4 as defined in chapter 2: Problem Definition). First two tasks of the thesis project i.e. a comprehensive study of current visual SLAM techniques and compilation of a state-of-the-art on visual SLAM techniques has already been presented in the previous chapter (chapter 3).

### 4.1. Requirement of a Visual SLAM Dataset

#### 4.1.1. Significance of the Dataset

Availability of a dataset which can permit the implement of different visual SLAM techniques is a basic and significant issue in carrying out research on visual SLAM. By implementing different techniques on the same dataset, the techniques can be critically evaluated and their pros and cons can be comprehended and formulated in an effective way.

#### 4.1.2. Basic Characteristics of the Dataset

The dataset should consist of the images taken by a stereo pair (fixed on a supposed robot) while moving along a trajectory in a three dimensional space. In addition to the images, the dataset should contain the information about ground truth position, orientation and velocity of the robot as it moved along the trajectory while grabbing the images, and information about the camera parameters. Dataset consisting of the images from a stereo pair enables the implementation of both single camera and stereo-vision SLAM approaches, so it is important that the dataset should be developed as stereo pair images instead of single camera images taken

along the trajectory of the robot.

#### **4.1.3. Feature Types Required to be Present in the Dataset**

In order to permit the study of visual SLAM techniques based on different feature types i.e. point, line and planar features etc. the dataset should be constructed for an environment containing all of these feature types. Similarly, corresponding images in the dataset should contain all of these feature types.

#### **4.1.4. Flexibility in terms of Feature Noise**

The images in the dataset should not contain any significant noise in terms of image quality or pixel position of the imaged structures in the environment. This has two advantages: (i) the absence of noise permits quicker implementation and hence evaluation of any visual SLAM technique and (ii) it gives the flexibility of adding different amounts of noise after extracting the features from the images. In this way different visual SLAM techniques can be tested and their performance for different noise values can be evaluated using a single visual SLAM dataset.

## **4.2. Development of the Dataset**

### **4.2.1. Epipolar Geometry Toolbox (EGT)**

The Epipolar Geometry Toolbox developed for Matlab by [24] provides a framework for creation and visualisation of 3D environments. It also allows creation of one or more cameras, and imaging of environment with the defined cameras.

The toolbox provides functions to define a world coordinate system, define point and line features, define cameras with desired intrinsic and extrinsic parameters, and to image the defined points and lines using the defined cameras. These functions provide the basic functionality required to build a synthetic dataset that can be used for the implementation of visual SLAM.

### 4.2.2. Steps Involved in Dataset Development

Different steps involved in the development of a dataset for visual SLAM are described below:

1. **Defining a World Coordinate Frame:** A world Coordinate Frame was defined using the EGT function “ $f_{3Dwf}$ ”.
2. **Defining 3D Structures:** Two 3D structures (simple urban outdoor structures) were defined in the world coordinate frame by defining points and lines in world coordinate frame and EGT function “ $f_{scenept}$ ” helped to visualise the 3D world.
3. **Defining Features for SLAM:** Point and Line features were defined on the above mentioned 3D structures. These point and line features were the corners and edges of windows and doors superimposed on the above mentioned 3D structures. Fig 4.1 shows different views of the 3D world created using above steps.

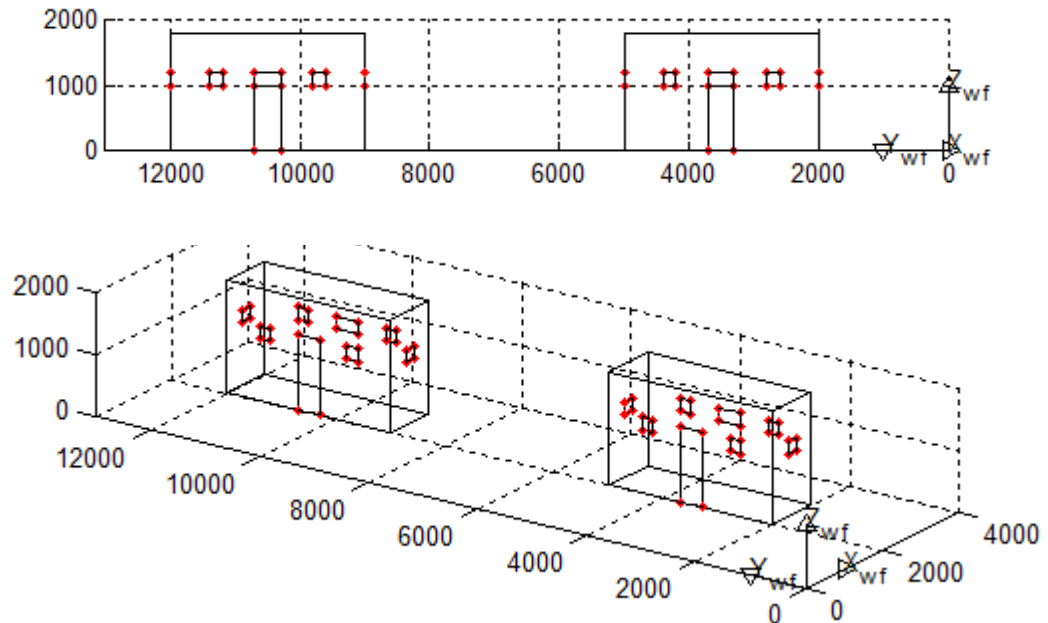


Figure 4.1. Two views of the 3D structure (all units in mm).

4. **Stereo Camera Pair:** Two cameras forming a stereo pair were defined. The stereo pair baseline was set as 50mm and the camera intrinsic parameters were set similar to those

of a real low cost webcam. EGT function “f\_3Dcamera” helped visualising the cameras in 3D world.

5. **Robot Trajectory:** A robot trajectory was defined to be traversed by the stereo camera pair. Cameras moved in three dimensions with changing orientations around the 3D structures. The trajectory was a looped path containing both straight segments and turns. Fig 4.2. shows the trajectory followed by the stereo camera pair around the 3D structure.

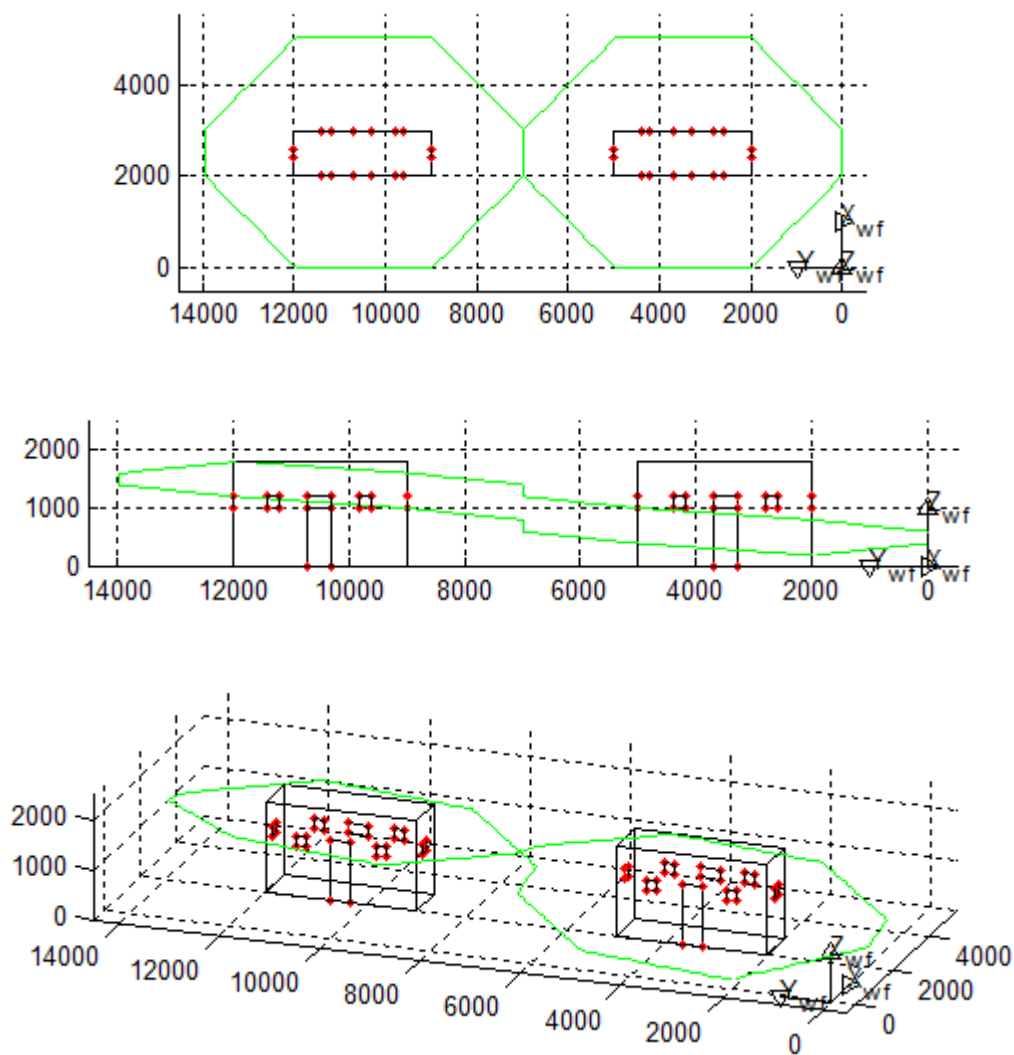


Figure 4.2. Three views of ground truth stereo camera pair trajectory (green), (all units in mm).

6. **Image Acquisition:** Images were acquired as the stereo camera pair moved along the above defined trajectory by utilising the EGT function “`f_perspproj`”. The number of acquired images per camera was set to 300 but it can be changed if desired and the new dataset can be constructed again by running the Matlab Code.
7. Three `.mat` files were also generated along with the images taken along robot trajectory. Two of these `.mat` files contain the ground truth data for robot's motion, and the third one contains the intrinsic camera parameters for the two cameras. The structure of the three files is described as follows.
  - a. *GroundTruth.mat*: This matrix/file contains the ground truth data for robot positions and velocity and landmark positions for all time instants. Its number of columns equals the number of time stamps (which was set to 300). First three rows give the ground truth values of x, y and z coordinates of robot's position with respect to the world coordinate frame, for all time instants. Next three rows give the x, y and z components of robot's velocity with respect to the world coordinate system, for all time instants. Similarly, the rows following the first six rows, give the ground truth x, y and z coordinates of all the landmark positions with respect to the world coordinate frame.
  - b. *RotGT.mat*: This matrix/file gives the ground truth for robots orientation as it moves along the defined trajectory. The file is a row vector having a length equal to the number of time stamps (i.e. 300 for the implementation). Each element of the vector represents a value of the robot rotation about z-axis of the world coordinate frame.

It is worth mentioning that at the instant when the two cameras (fixed on the assumed robot) are defined in EGT, the camera coordinate systems are aligned to the world coordinate system defined in EGT. Camera coordinate systems are given a rotation of  $\pi/2$  radians about y-axis of the world coordinate system, followed by a rotation of  $-\pi/2$  radians about x-axis of the world coordinate system. This is done in order to acquire the correct images of the defined 3D environment. Therefore at any time instant, the value of rotation angle about z-axis given in `RotGT.mat` can be

used to find the current rotation matrices for the camera pair by pre-multiplying the z-rotation matrix by the initial camera rotation matrices formed using only the above mentioned rotations about y and x axes.

- c. *InSicsIn2.mat*: This is a 3x6 matrix/file. Its first and second 3x3 portions contain the intrinsic camera matrices for the two cameras in the stereo camera pair.

Some resulting images from the dataset generated by following the above described procedure are presented in the next chapter (Chapter 5: Results).

### **4.3. Implementation of Vision Based EKF SLAM**

#### **4.3.1. Aims**

After developing the dataset for visual SLAM, the Extended Kalman Filtering (described in section 3.4.1) based visual SLAM was implemented on the dataset. Main aim of the implementation was to evaluate the performance of EKF SLAM when it is performed using vision as the only sensor detecting landmarks in the environment, along with the information from other sensors measuring robot position and velocity (odometry). Following part of this chapter explains the steps involved in the implementation and other details. The SLAM toolbox by [25] is an excellent tool to understand the basics in the area of Extended Kalman Filtering based SLAM and served as a tool for SLAM implementation for current thesis.

#### **4.3.2. Assumptions**

In order to accelerate and simplify the implementation process, the feature extraction and matching problems were assumed to have been already solved. This was possible as the dataset was synthetically developed. Incorporating dataset development framework within the SLAM implementation framework enabled the issue of feature extraction and matching considered to be solved.

### 4.3.3. Sensor Noise

Once the values from sensors were obtained, the desired amount of noise could be added to them. This flexibility permitted to investigate the performance of EKF based visual SLAM for different values of noise in the information from cameras (i.e. the sensors being used to measure landmark positions) and other sensors which measured robot position and velocity.

### 4.3.4. Steps Involved in EKF Based SLAM Implementation

Different steps involved in the implementation of EKF based visual SLAM are described below:

1. **Setting up 3D Environment:** As explained earlier that the dataset development framework was incorporated in the SLAM implementation framework, the first step in the implementation was setting up the 3D world with two building structures, features, stereo camera pair and a robot trajectory (as described in section 4.2.2, steps 1 - 5).
2. **Noise Variance:** Values of noise variance for the data from sensors measuring landmark positions and robot position and velocity were defined. Noise in all the sensors was considered to be Gaussian with zero mean. Moreover the variance of noise accounting for the inaccuracies in system modeling and disturbances was also defined.
3. **Models and Matrix Definitions:** Implementation of SLAM requires some models and matrices to be defined depending on the robot/system dynamics, available measuring sensors and their accuracy, and landmark observation geometry, as explained in section 3.4.1. These include the state matrix  $X$ , motion model  $F$ , control input matrix  $U$  and noise covariance matrix  $Q$  (see Eq 3.8), measurement model  $H$  and noise covariance matrix  $R$  (see Eq 3.9), and similarly the state covariance matrix  $P$  (see Eq 3.10). The definitions of above mentioned models and matrices is further explained as follows.
  - a. *State Matrix X:* The state matrix  $X$  was defined as a column vector containing the  $x$ ,  $y$  and  $z$  coordinates of robot position as the first three elements, and  $x$ ,  $y$  and  $z$  components of robot velocity in the next three elements. After these six elements, every three elements represent the  $x$ ,  $y$  and  $z$  coordinates of each of the landmark positions.

- b.** *Motion Model  $F$* : The motion model  $F$  was defined as a square matrix having size corresponding to the length of state matrix  $X$ , and with diagonal elements as 1. Non-diagonal elements of the matrix were zero with the exception of elements (1,4), (2,5) and (3,6) as they account for the displacement of the robot in x, y and z directions caused by the x, y and z components of the current robot velocity.
- c.** *Control Input Matrix  $U$* : The control input matrix was defined as a column vector with the length equal to the length of state matrix  $X$ . All the elements of  $U$  were set to zero as there is no explicit external input into the system.
- d.** *Noise Covariance Matrix  $Q$* : Noise covariance matrix  $Q$  caters for inaccuracies in system modeling and other disturbances.  $Q$  was defined as a square matrix of size corresponding to the length of state matrix  $X$ , and having all zeros except the diagonal elements whose values were set to the process noise value defined in step 2.
- e.** *Measurement Model  $H$* : The measurement model  $H$  is defined as an identity matrix of size corresponding to the length of state matrix  $X$ . This represents that the measured robot position, velocity and landmark positions are represented in the state matrix as they are measured in 3D world coordinate system.
- f.** *Noise Covariance Matrix  $R$* : Similar to  $Q$ , the matrix  $R$  represents the covariance in measurement noise for all the sensors.  $R$  was setup as diagonal matrix with size corresponding to the length of state matrix. Diagonal elements were given the corresponding values of the noise variances defined in step 2, i.e. first three diagonal elements have the values equal to the variance of position measurement sensor noise, next three diagonal elements have the values equal to the velocity measurement sensor noise variance, and rest of the elements have the values equal to the variance of noise in the measurement of 3D landmark positions.
- g.** *State Covariance Matrix  $P$* : State covariance matrix  $P$  represents the covariance between different elements in the state. It is a square matrix with size corresponding to the length of state matrix  $X$  and is generally initialised at least with some larger diagonal values. For the current SLAM implementation the matrix was initialised

with some nonzero non-diagonal elements in addition to the nonzero diagonal elements, as shown and explained as follows. The matrix shown is 9x9 and represents the presence of only one landmark in the state, actual matrix dimensions are three times more rows and columns per additional landmark. Empty places in the matrix represent zeros and have been left blank for clarity.

p+v			v			r		
	p+v			v			r	
		p+v			v			r
v			v					
	v			v				
		v			v			
r						r+p		
	r						r+p	
		r						r+p

Figure 4.3. Structure of State Covariance Matrix  $P$ .

The first three diagonal entries have the values equal to the sum of robot position and velocity measurement noise variances (shown  $p+v$  in the above figure), because the velocity measurement sensors also affect the position estimate as the velocity information is considered to estimate the new position at a time instant. Next three diagonal elements represent the covariance in the velocity measurement sensor. Last three diagonal elements have the values equal to the sum of robot position and landmark position measurement variances (shown  $r+p$  in the above figure). The non-diagonal values of “v” represent the covariance between robot position and velocity. Similarly, non-diagonal values of “r” represent the covariance between robot position and landmark positions.

- 4. Prediction:** At this step, the state at time instant  $k$  is predicted using the function matrix  $F$ , state estimate at time instant  $k-1$  and the control input  $U_k$ . In other words, this is the implementation of Eq 3.11 described in chapter 3.

Similarly, the new values of covariance matrix  $P_k$  are computed using the function

matrix  $F$  and covariance matrix at time instant  $k-1$ . This is the implementation of Eq 3.12.

5. **Observation:** At this step, the landmarks are observed using the camera pair. In the implementation, currently viewable landmarks in the environment are detected. As explained in section 4.3.2, the feature extraction and matching problem has been taken for granted as solved, therefore the pixel positions of the visible landmarks are back projected into 3D to get the 3D position of the visible landmarks. Once the positions are computed, desired amount of noise is added to the positions. Similarly, the current position and velocity of the robot are measured and desired amount of noise is added.
6. **Update:** Once the measurements have been made, first of all the *innovation* vector is calculated, i.e. the difference between measured and predicted (in previous step) values of the state. The innovation term for unobserved landmarks is considered zero.

The innovation covariance matrix  $S$  is calculated using matrices  $H$ ,  $P_k$  and  $R$ . Then, Kalman gain  $W$  is calculated using covariance matrix  $P_k$ , matrix  $H$  and the innovation covariance  $S$ . This is the implementation of Eq 3.15 and Eq 3.16.

Kalman gain  $W$  and calculated *innovation* are then used to update the state matrix for time instant  $k$ . This is the implementation of Eq 3.13.

**Note:** The steps 4, 5 and 6 are continuously repeated in a loop for all the time instants  $k$ .

Results of the above explained EKF based visual SLAM implementation are presented in the next chapter (Chapter 5: Results).

#### 4.4. Problems Faced

A problem faced during the thesis study was the limitation of time span for the study. Because of time limitations we could not implement the EKF based visual SLAM for 6 DOF. Instead it was done only for 3 DOF as explained in the previous section.

# Chapter 5

## Results

This chapter presents the results for the dataset development and Extended Kalman Filtering based visual SLAM implementation as described in the previous chapter. First the results from the developed dataset have been presented, followed by the SLAM implementation results.

### 5.1. Dataset

The developed dataset consists of the images taken from a stereo pair as the cameras (fixed on an assumed robot) move along a trajectory, the ground truth robot position, velocity and orientation in 3D as it moves along the trajectory, landmark positions in 3D, and intrinsic parameters for both cameras of the stereo pair. The structure of the .mat files containing the ground truth data and intrinsic camera parameters has already been described in the previous chapter in section 4.2.2 (statement 7).

#### 5.1.1. Results

As the number of time instants was set to 300, the dataset consists of 600 images i.e. 300 per camera. The images have a resolution of 640 x 480 pixels. Three images acquired per camera are presented below in Fig 5.2. The images belong to the portion of robot trajectory marked as a small blue coloured spot (time instants 146-148) in Fig 5.1 (top view of the 3D environment).

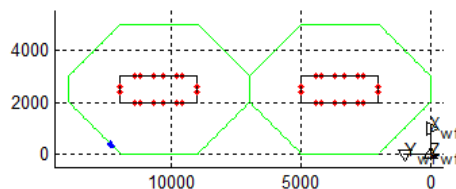


Figure 5.1. Robot trajectory top view (all units in mm).

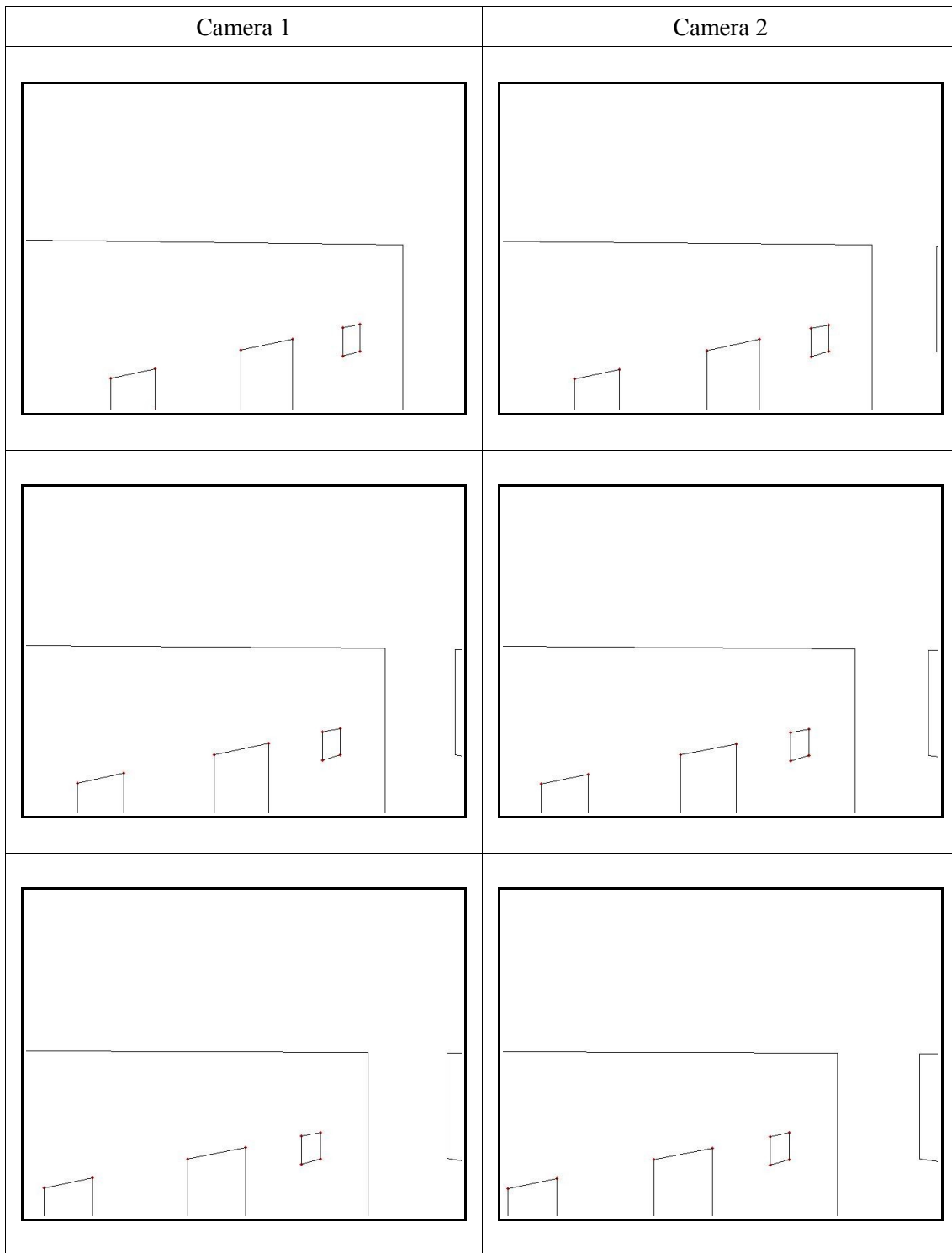


Figure 5.2. Dataset images for time instants 146-148 for the two cameras.

### 5.1.2. Comments

The images belong to the portion of robot trajectory marked in blue colour in Fig 5.1. At this time the robot is moving rightwards and downwards (with reference to environment shown in Fig 5.1). From this portion of the trajectory, at the 146th time instant, camera-1 could only see one building while camera-2 has started to see the other (farther) building as it is slightly ahead of camera-1 (the stereo pair baseline was set to 50mm). Similarly, it can be seen in the images taken at time instants 147 and 148 that camera-2 can see scene more toward right in the scene as compared to camera-1.

## 5.2. EKF Based SLAM Implementation

This section presents the results for the Extended Kalman Filtering based implementation of visual SLAM using the developed dataset, as explained in the previous chapter.

Below is a figure showing 3D environment with ground truth robot trajectory. Robot traveled a total distance of 38.789m along the shown trajectory. As the total number of time stamps was set to 300, the distance traveled by robot per time stamp is 0.129m.

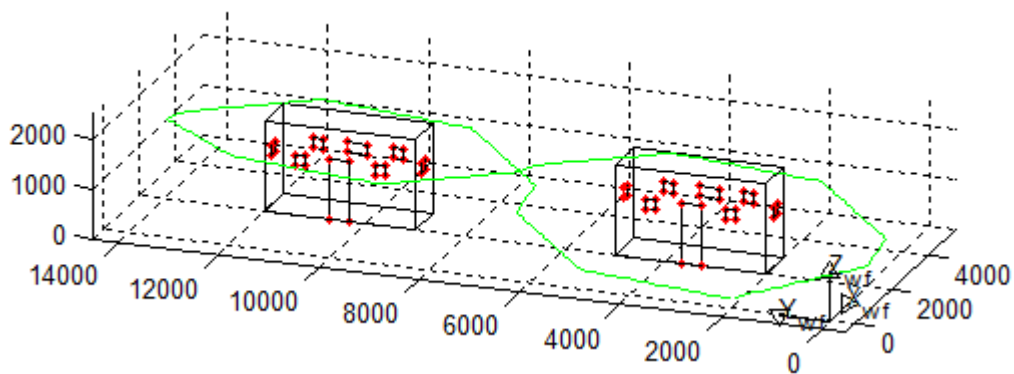


Figure 5.3. Ground truth trajectory of the robot in 3D environment (all units in mm).

### 5.2.1. Sensor Noise Variances

Considering the robot's displacement of about 129mm (0.129m) between every two consecutive time stamps, it was considered reasonable to test the SLAM algorithm using the robot position

noise variance of  $50^2$  (i.e. standard deviation of 50mm or 0.05m) for each of x, y and z position coordinates. The velocity measurement and landmark position measurement processes was considered less noise and the corresponding noise covariances were set to  $10^2$  (i.e. standard deviation of 10mm or 0.01m).

### 5.2.2. Results

Following figure shows some views of the ground truth (green) and estimated (blue) trajectories in the 3D environment.

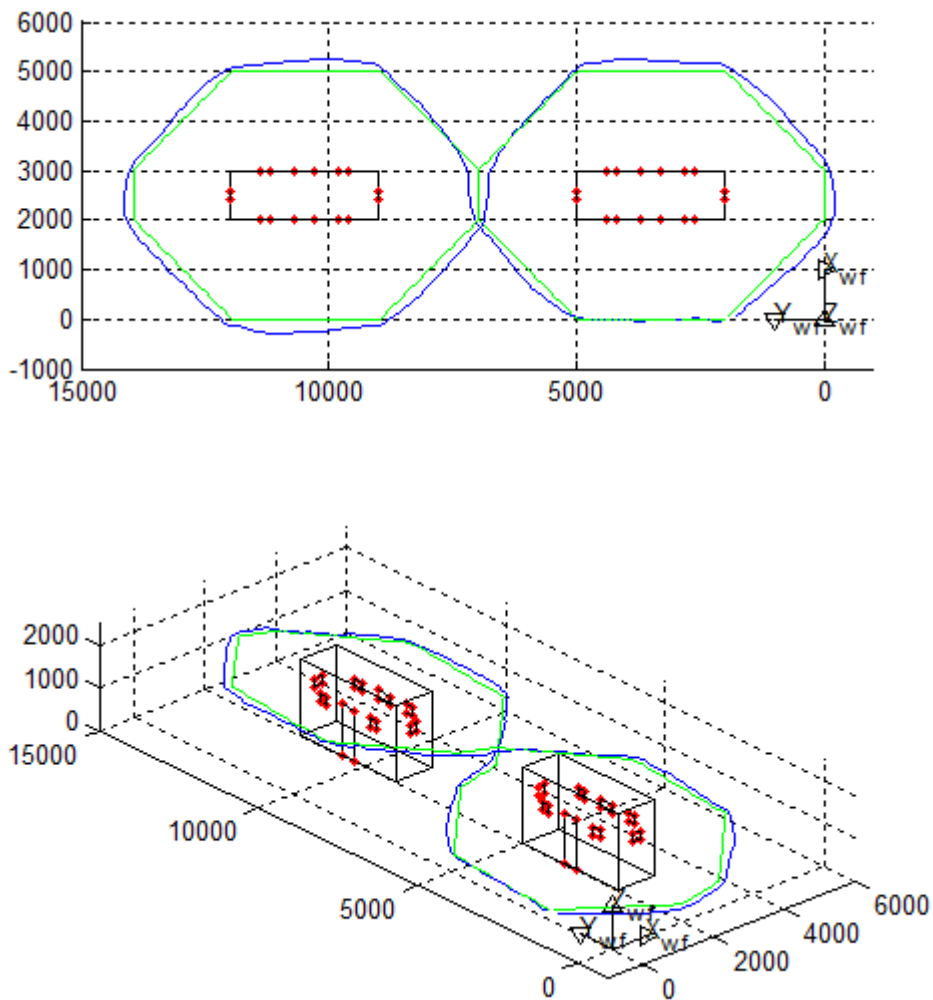


Figure 5.4. Two views of ground truth (green) and estimated (blue) robot trajectories (all units in mm).

Fig 5.5 shows the ground truth and estimated x, y and z coordinates of robot position and the computed  $3\sigma$  boundaries for the first 25 time stamps. Table 5.1 shows the values of mean absolute errors for x, y and z components of robot position and velocity for the complete robot trajectory.

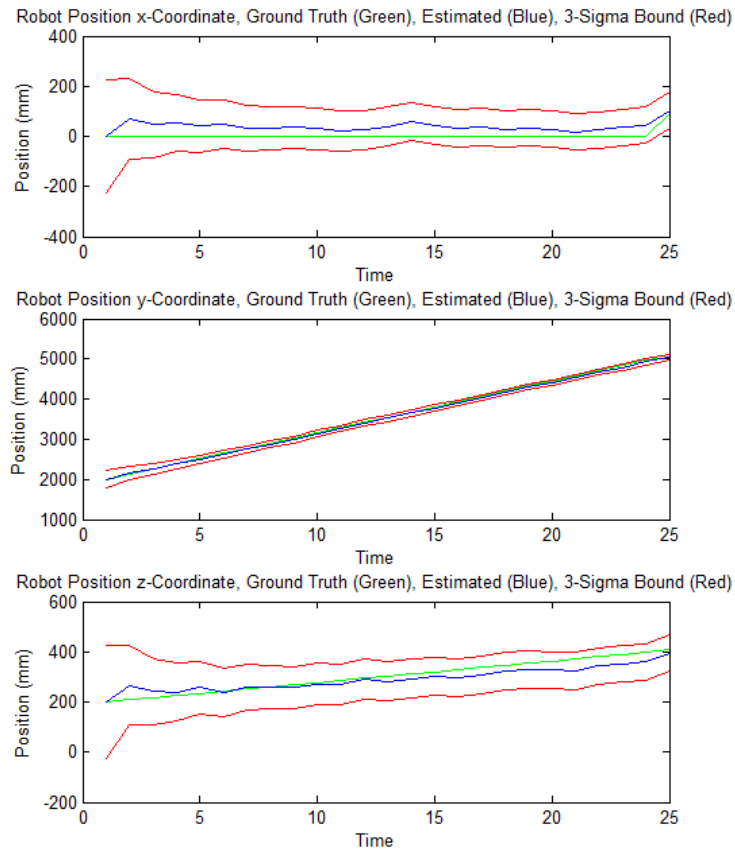


Figure 5.5. Ground truth (green), estimated (blue) and  $3\sigma$  bounds (red) for x, y and z coordinates of robot position for first 25 time stamps.

Table 5.1. Overall mean absolute estimation errors.

Position x	Position y	Position z	Velocity x	Velocity y	Velocity z
110.6mm	102.7mm	18.7mm	21.8mm	23.3mm	3.8mm

### 5.2.3. Comments

From Fig 5.4 it is evident that EKF based visual SLAM algorithm performs well even with moderate values of noise added to the robot position and velocity, and landmark measurements.

Fig 5.5 provides further in-depth into the SLAM process and shows that the ground truth position values lie within  $3\sigma$  bounds of the estimated position values for the time interval shown in the figure. The figure also shows how the uncertainty in the estimated robot position coordinates decreases as the SLAM process progresses in time.

Table 5.1 shows that the overall mean absolute error for the x,y and z components of robot velocity is significantly lower than the error in robot position coordinates. This is because of the fact that position measurements were set as five times more noisy than velocity measurements and described earlier in section 5.2.1. The table also shows that the error for z components of both robot's position and velocity is significantly lower than the error in x and y components. This is because of the fact that while moving along the ground truth trajectory, robot moves very smoothly in z direction as compared to x and y directions as can be clearly seen in figures 5.3 and 5.4.

# Chapter 6

## Conclusions

This thesis presents a state-of-the-art on vision based Simultaneous Localisation and Mapping (Visual SLAM). Development of a synthetic dataset to permit the implementation and testing of visual SLAM techniques has also been presented along with the implementation of Extended Kalman Filtering based visual SLAM on the dataset.

### 6.1. Conclusions

The work presents the techniques that are currently being used to solve different issues involved in the problem of visual SLAM including different types of imaging systems, types of features extracted from the environment, types of landmark initialisations, SLAM techniques and use of wheel odometry information. The study leads to the conclusion that multiple possibilities exist to solve different issues involved in visual SLAM and each possible technique has its own pros and cons and might be more suitable for a particular application than others.

Unavailability of a dataset that can serve as an environment for testing and evaluation of different visual SLAM techniques lead towards the development of a synthetic dataset which can be used for implementation, testing and evaluation of visual SLAM techniques.

Implementation of Extended Kalman Filtering based visual SLAM on the developed dataset has proved the technique to be efficient and successful even in the presence of significant noise in robot position and velocity, and landmark position sensing.

## **6.2. Significance of the Work**

State-of-the-art on visual SLAM studied and compiled during the thesis is a significant contribution which can serve as a basis for future research in the area. This will lead to the study and evaluation of different visual SLAM techniques and their effective and application specific implementation into the real systems including not only robots but also augmented reality applications.

The dataset developed during the study can serve as a basis for the implementation and hence better understanding and evaluation of different visual SLAM techniques, as it did for Extended Kalman Filtering based visual SLAM during the current study.

## **6.3. Future Works**

The work presented in this thesis provides a foundation for further research in the area of visual SLAM. Possible future research strategy is to study and implement different visual SLAM techniques on the developed synthetic dataset. Then a real dataset can be acquired using imaging hardware. Depending on a desired application, the best suited techniques can be implemented and evaluated using the real dataset. This can lead to further improvements of existing visual SLAM techniques and their application on real world robotic systems including autonomous ground, aerial and underwater robots.

# Bibliography

- [1] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman, “A Framework for Vision Based Bearing Only 3D SLAM”, in *International Conference on Robotics and Automation*, Orlando, FL, 2006.
- [2] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, “Monocular Vision Based SLAM for Mobile Robots”, in *International Conference on Pattern Recognition*, Hong Kong, 2006.
- [3] L. Goncalves, E. Di Bernardo, D. Benson, M. Svedman, J. Ostrowski, N. Karlsson, and P. Pirjanian, “A visual Front-end for Simultaneous Localization and Mapping”, in *International Conference on Robotics and Automation*, Barcelona, Spain. 2005.
- [4] M. Milford and G. Wyeth, “Featureless Vehicle-Based Visual SLAM with a Consumer Camera”, in *Australasian Conference on Robotics and Automation*, Brisbane, Australia, 2007.
- [5] T. Lemaire and S. Lacroix, “Monocular-vision based SLAM using line segments”, in *International Conference on Robotics and Automation*, Roma, Italy, 2007.
- [6] E. Eade and T. Drummond, “Edge Landmarks in Monocular SLAM”, in *British Machine Vision Conference*, Edinburgh, UK, 2006.
- [7] A.J. Davison, I.D. Reid, N.D. Molton and O. Stasse, MonoSLAM: Real-Time Single Camera SLAM, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, 2007, pp 1052-1067.
- [8] A.J. Davison, Y.G. Cid, and N. Kita, “Real-Time 3D SLAM with Wide-Angle Vision”, in *IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, 2004.

- 
- [9] T. Lemaire, C. Berger, I. Jung and S. Lacroix, Vision-Based SLAM: Stereo and Monocular Approaches, *International Journal of Computer Vision*, vol. 74, no. 3, 2007, pp 343-364.
- [10] C. Berger and S. Lacroix, "Using Planar Facets for Stereovision SLAM", HAL – CCSD e-articles, available at <http://hal.archives-ouvertes.fr/hal-00174889/en/>, 2007.
- [11] M. Kaess and F. Dellaert, Visual SLAM with a Multi-Camera Rig, *Technical Report GIT-GVU-06-06*, Georgia Institute of Technology, 2006.
- [12] T. Lemaire and S. Lacroix, SLAM with panoramic vision, *Journal of Field Robotics*, vol. 24, no. 1-2, 2007, pp. 91-111.
- [13] J. Kim and M.J. Chung, "SLAM with Omni-directional Stereo Vision Sensor", in *International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, 2003.
- [14] J. Kim, K. Yoon, J. Kim and I. Kweon, "Visual SLAM by Single-Camera Catadioptric Stereo", in *SICE-ICASE International Joint Conference*, Busan, Korea, 2006.
- [15] M. Ballesta, A. Gil, Ó.M. Mozos, and Ó. Reinoso, "Local Descriptors for Visual SLAM", in *Workshop on Robotics and Mathematics*, Coimbra, Portugal, 2007.
- [16] Ó.M. Mozos, A. Gil, M.Ballesta, and Ó. Reinoso. "Interest Point Detectors for Visual SLAM", in *Conference of the Spanish Association for Artificial Intelligence*, Salamanca, Spain, 2007.
- [17] P. Elinas, R. Sim, and J.J. Little, " $\sigma$ -SLAM: Stereo Vision SLAM Using the Rao-Blackwellised Particle Filter and a Novel Mixture Proposal Distribution", in *International Conference on Robotics and Automation*, Orlando, FL, 2006.
- [18] D.G. Lowe, "Object Recognition from Local Scale-Invariant Features", in *International Conference on Computer Vision*, Corfu, Greece, 1999.

- 
- [19] H. Durrant-Whyte, T. Bailey. Simultaneous Localization and Mapping: Part I, *IEEE Robotics & Automation Magazine*, 2006, pp 99-108.
- [20] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem”, in *National Conference on Artificial Intelligence*, Edmonton, Canada, 2002.
- [21] M. J. Milford, G.F. Wyeth and D. Prasser, “RatSLAM: A Hippocampal Model for Simultaneous Localization and Mapping”, in *International Conference on Robotics and Automation*, New Orleans, LA, 2004.
- [22] H-Y. Shum, Q. Ke and Z. Zhang, “Efficient Bundle Adjustment with Virtual Key Frames: A Hierarchical Approach to Multi-frame Structure from Motion”, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Ft. Collins, CO, 1999.
- [23] D. Nistér, O. Naroditsky and J. Bergen, “Visual Odometry” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004.
- [24] G. L. Mariottini and D. Parattichizzo, EGT for Multiple View Geometry and Visual Servoing, *IEEE Robotics & Automation Magazine*, 2005, pp 26-39.
- [25] J. Salvi, “Simultaneous Localization and Mapping toolbox”, available at <http://eia.udg.es/~qsalvi/Slam.zip>