

## VISION-ENHANCED LOCALIZATION FOR COOPERATIVE ROBOTICS

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

---

Sreekanth Boga

Certificate of Approval:

---

Sanjeev Baskiyar  
Associate Professor  
Computer Science And Software Engineering

---

Thaddeus A. Roppel, Chair  
Associate Professor  
Electrical and Computer Engineering

---

David A. Umphress  
Associate Professor  
Computer Science And Software Engineering

---

George T. Flowers  
Dean  
Graduate School

VISION-ENHANCED LOCALIZATION FOR COOPERATIVE ROBOTICS

Sreekanth Boga

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama

May 14, 2010

VISION-ENHANCED LOCALIZATION FOR COOPERATIVE ROBOTICS

Sreekanth Boga

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

---

Signature of Author

---

Date of Graduation

## VITA

Sreekanth Boga, son of Venkata Subbaiah Boga and Padmavathi Boga was born on Febraury 16th, 1985, in Kadapa district, India. He received the Bachelor of Technology degree in Computer Science Engineering from Vignan Institute of Technology, Visakhapatnam in 2006. He worked in Infosys Technologies Limited, Bangalore from May 2006 to December 2007. He joined the Masters program in the Department of Computer Science Engineering at Auburn University in January 2007. His current area of research deals with Simultaneous Localization And Mapping problem.

## THESIS ABSTRACT

### VISION-ENHANCED LOCALIZATION FOR COOPERATIVE ROBOTICS

Sreekanth Boga

Master of Science, May 14, 2010  
(B.Tech.,Jawaharlal Nehru Technological University, 2006)

59 Typed Pages

Directed by Thaddeus A. Roppel

Simultaneous Localization And Mapping - SLAM, is an extremely challenging open problem. SLAM involves a mobile robot building a spatial map of its environment and finding the ego-position in the partially built map. The problem is analogous to the chicken-egg situation; since, an accurate map cannot be built without knowing the ego-position, while, the position of self cannot be determined until it has a very accurate map.

As the real world is far from being ideal, a probabilistic approach is devised to model the SLAM system. This thesis aims to handle the problem with a stereovision system. There are 2 aspects of the solution; one deals with processing the images gathered from the environment and the other deals with estimating the ego-position of the robot from the images gathered. The first aspect is fulfilled by using a SIFT (Scale Invariant Feature Transforms) algorithm and the second aspect is managed by the Rao-Blackwellized particle filtering algorithm. The results provided by the

Matlab simulation environment showed that the SLAM system could converge well to the real world values.

## ACKNOWLEDGMENTS

First and foremost, I would express my sincere gratitude to my advisor Associate Professor Dr. Thaddeus A. Roppel without whose guidance and support this thesis would not be possible. I really thank him for understanding my situation and encouraging me through my research. I thank Dr. Sanjeev Baskiyar for co-advising me and directing me through the proper channels in my research course. I gratefully acknowledge the facilities and financial support provided by Dr. Prathima Agrawal. I also thank Dr. David Umphress for being a part of my advisory committee and catering time during his busy schedules.

My thanks goes to my senior Chris Wilson for introducing me to the robotics environment; also, I thank my friends Yogesh Kondareddy, Nirmal Andrews, Nida Bano, Santosh Kulkarni and Pratap Simha for discussions and valuable suggestions on my research work. I thank my other friends who were supporting me externally when their presence is needed.

I am grateful to my parents for their consistent support and encouragement during my study.

Style manual or journal used Journal of Approximation Theory (together with the style known as “aums”). Bibliography follows van Leunen’s *A Handbook for Scholars*.

Computer software used The document preparation package T<sub>E</sub>X (specifically L<sup>A</sup>T<sub>E</sub>X) together with the departmental style-file `aums.sty`.



## TABLE OF CONTENTS

LIST OF FIGURES	xi
1 INTRODUCTION	1
2 LITERATURE SURVEY	4
2.1 SLAM History . . . . .	4
2.2 SLAM Sensing Systems . . . . .	5
2.2.1 Sonar and Laser . . . . .	5
3 HARDWARE	9
3.1 Experimental Field . . . . .	9
3.2 Robotic Hardware . . . . .	10
3.2.1 Chassis . . . . .	11
4 SLAM AND SIFT	16
4.1 SLAM process and model . . . . .	16
4.1.1 SLAM in General . . . . .	16
4.1.2 SLAM Mathematical Model . . . . .	20
4.2 Bayesian Filtering Technique . . . . .	23
4.3 Scale Invariant Feature Transforms - SIFT . . . . .	24
4.3.1 Scale-space extrema detection . . . . .	25
4.3.2 Keypoint localization . . . . .	25
4.3.3 Orientation assignment . . . . .	26
4.3.4 Keypoint descriptor . . . . .	26
4.4 Rao-Blackwell Particle Filter - RBPF . . . . .	27
4.4.1 Reintroducing SLAM . . . . .	27
4.4.2 Factoring the SLAM Posterior - $Bel(x_t)$ . . . . .	28
4.4.3 RBPF Data Association . . . . .	30
5 EXPERIMENT AND RESULTS	35
5.1 The Setup . . . . .	35
5.1.1 Modified Hardware Testbed . . . . .	35
5.1.2 Stereo Imaging . . . . .	36
5.2 Results . . . . .	39

6 CONCLUSION	43
BIBLIOGRAPHY	44

## LIST OF FIGURES

3.1	Image of the Experimental Field. . . . .	10
3.2	Current Distance measurement sensor. . . . .	13
3.3	Robot mounted with optical mouse and batteries. . . . .	14
3.4	Gumstix (bottom) and Wifistix (top). . . . .	15
4.1	Mobile Robot operating scenario. . . . .	17
4.2	Accrued Uncertainty in robot pose and landmark. . . . .	18
4.3	SLAM illustration graphically. . . . .	19
4.4	SLAM process . . . . .	20
4.5	Robot observing the Range and Bearing of a Feature. . . . .	22
4.6	Online operation of Bayesian Rule . . . . .	24
4.7	SLAM Problem for realizing the factored solution. . . . .	29
4.8	Particles in RBPF. . . . .	30
4.9	Re-sampling to approximate target distribution. . . . .	33
5.1	Modified CRR testbed. . . . .	36
5.2	Simulating stereo imaging. Right mounted and left mounted on chassis. . . . .	37
5.3	Extracting $[x, y, z]$ from left and right images. . . . .	38
5.4	SIFT feature matches between left and right images at $(20,0)$ . . . . .	40
5.5	Final Output of RBPF simulation. . . . .	41
5.6	Plot of path estimated by the robot. . . . .	42
5.7	Feature 21 estimate by 100 particles. . . . .	42

## CHAPTER 1

### INTRODUCTION

Vision based systems have gained significant importance in the field of mobile robotics during the last decade due to the low cost and rich information that cameras provide as compared to the traditional robotic sensors like sonar's or laser scanners. Applications of Vision-based systems widely range from object recognition [1]-[3], obstacle avoidance [4]-[6], navigation [7][8], to global localization [9][10] and, more recently, in SLAM [11]-[13].

The term SLAM is an acronym for Simultaneous Localization and Mapping. The term was originally coined by Hugh Durrant-Whyte and John J. Leonard [14]. SLAM has been one of the most challenging open problems in building truly autonomous robots and in fact the solution to the SLAM problem is in many ways a "Holy Grail" to the autonomous vehicle research community. The ability to place an autonomous mobile robot at an unknown location in an unknown environment and then have it build a map, using only relative observations of the environment, and then to use this map simultaneously to navigate would indeed make such a robot truly autonomous. In short the problem involves a mobile robot attempting to recover a spatial map of its environment, while simultaneously estimating its own pose relative to the partially built map. The problem is explained in detail in the coming chapters.

This thesis work addresses enhancing the localization of the mobile robots in Cooperative Robotics Testbed [16][17] employing vision as observatory sensors. The

mobile robots work in groups based on a cooperative distributed protocol called SARA - Search And Rescue Algorithm. SARA incorporates the concepts of a relatively new field of cooperative robotics [18][19]. Cooperative robotics deals with the study of multiple autonomous agents “working together” to perform the task that is too difficult or impossible for one agent to perform acting alone. SARA accomplishes its task by tackling several sub-problems like mutual communication among robots, path planning, object recognition and target tracking. All of these sub problems are inadvertently dependent on the task of the robot localizing itself in the hardware environment.

Incorporating vision-based localization into the SARA protocol would span a new direction to the existing cooperative robotics test bed. The SARA is a protocol that has passed through several phases of modifications from the base version by incorporating certain new features, removing the defective pieces of the system and also enhancing the performance of the existing parts. Currently, the primary focus would be to perform simulations on the vision induced mobile robot. Based on the results of simulation the robots will be attached with a vision sensor system to work for the SARA algorithm and the system protocol can be upgraded to the next version.

Vision based localized systems revolve around two primarily algorithms; firstly, for feature extraction from the images of the environment and secondly, for estimating the position of the mobile robot involving the usage of filtering algorithms. For feature extraction an algorithm called SIFT (Scale Invariant Feature Transforms) and for pose estimation Rao Blackwellized particle filters have been employed. The presence of inherent visual noise in real life scenarios as compared to ideal environments demands the use of probabilistic approaches for modeling robotic movements along with the

position of the landmark in the hardware test bed. The algorithms used in this work are very efficient in performance in their respective domains.

In Chapter 2 an overview of the literature in the domain of machine learning, pertaining to the foundations of the visual-SLAM's key algorithms SIFT and Rao-Blackwellized particle filtering is provided. Chapter 3 encompasses a complete description of the hardware that is used for performing the simulations follows. This chapter gives a detailed explanation of the vision sensor along with the type of images being studied. It also explains about the robot's hardware and the hardware test bed environment on which the simulation models are drawn. Chapter 4 introduces the problem in a mathematical perspective and also provides the notations that are being used through the document. This chapter describes the reasons to choose the aforementioned feature extraction and position estimation algorithms for dealing with the problem of visual-SLAM, also a sufficient amount of description and pseudo-code of the algorithms is provided for better comprehension. The simulation methods and the results obtained will be presented in Chapter 5. Chapter 6 discusses the obtained results. It compares the robots with and without the usage of vision sensors. The directions for future work and a brief conclusion are drafted in Chapter 7. An appendix with the reference to the source code used for simulations is also given.

## CHAPTER 2

### LITERATURE SURVEY

There are huge libraries of literature concerning Machine Learning. Machine Learning is in its budding stages of research and researchers all over the globe are contributing heavily due to the potential advantages involved in machine learning. This chapter discusses the existing research work in the domain of simultaneous localization and mapping. It starts with a brief history of SLAM. The history is followed by a discussion of other methods based on sensing systems that have been employed to handle the problem of SLAM. This section also includes and covers literature regarding the application of SIFT and Rao-Blackwell Particle Filter algorithms.

#### 2.1 SLAM History

The term SLAM is as stated an acronym for Simultaneous Localization And Mapping. It was originally developed by Hugh Durrant-Whyte and John J. Leonard [28] based on earlier work by Smith, Self and Cheeseman [29]. Durrant-Whyte and Leonard originally termed it SMAL but it was later changed to SLAM to give a better impact. They were the first people who have employed the probabilistic approach for modeling the uncertainty in nature. Prior to them scientists employed to represent spatial uncertainty in typical robotic applications [31][32] by numerically computing min-max bounds on the errors. In [33] the authors have proposed a probabilistic

model using a scalar representation of positional uncertainty instead of a multivariate representation of position and orientation.

## **2.2 SLAM Sensing Systems**

The design of a mobile robot relies mostly on the mode of its interaction with the environment. One of the design parameters would be the kind of information and amount of information to gather from the environment. Depending on the environment the design may also fuse multiple sensors for extracting the data. Pertaining to the problem of SLAM, the sensors can be narrowed to infrared, sonar, laser and vision. The literature of sensing systems along with the techniques that were applied for data management is described.

### **2.2.1 Sonar and Laser**

Sonar has been used in some early applications. Although sonar has been found to be very useful in underwater scenarios [35], it fails badly in terrestrial and aerial applications. SLAM experiments conducted in [33] employed as many as 16 sonar sensors. It proved to be useful only in a structured and noiseless environment. Considering the noise in the environment the results were shown to be poor. Elfes [34] developed a sonar sensor model, which simulates the inherent defects of sonar sensor like that of poor range precision, wide beam angles, multiple reflects, and detection sensitivity. Wouter [36] has performed simulations to analyze the kinematic model of the robot based on the speed of the laser. A laser based tabletop localization system is designed for the study. A battery powered 650 nm laser beam rotating in a horizontal plane is employed. The laser beam triggers phototransistors placed in four corners of



a flat table. A base station collects the voltage signals from these 4 phototransistors. The values are transmitted back to the robot, which can then determine the robot position using simple mathematics [36]. The robot uses these 4 beams reflected back to the robot are being used to determine the robot's position. This non-linear localization system has a mean Euclidean error of 0.02 m at a rotational laser velocity of 35 m/s and a robot velocity of 1 m/s. The largest errors occurred at the corners of a table and are about four times the amount of the smallest error. It was found that, as the speed of the robot increases the accuracy keeps declining.

## **Vision Sensors**

Vision systems have gained tremendous popularity in the recent past. Computing power has been growing steadily as per Moore's Law. As a result, the limitations that restricted sophisticated image processing are being removed. A lot of work in the technical literature has addressed robot localization and SLAM using visual sensors, it spans vision systems like omni directional, monocular, stereo, and trinocular cameras.

### *Omnidirectional Cameras*

In [43] an omnidirectional camera is used to estimate the distance of closest color transition in the environment, mimicking the performance of laser rangefinders. These measurements are filtered by particle filter to determine the robot's pose in a previously constructed map. Tamini et al. [44] presented an omnidirectional camera-based global localization approach for mobile robots using a modified version of SIFT. Modified SIFT generates fewer number of features and hence needs less processing.

The trinocular camera used in [41] addresses the SLAM problem by tracking SIFT visual features in static environments [52]. The motion of the self is computed from the robot odometry as an initial estimation, and a least-squares method, which finds the camera movement with the best alignment between the observed and the predicted image coordinates of the 3D landmarks in the map. The spatial uncertainty in the landmarks is modeled by a Kalman filter. In [49] a trinocular SLAM system, which uses 3D line segments as the elements of the map (instead of point features) has been proposed. The robot pose is approximated with a particle filter and uncertainty in the 3D segments of the map is modeled with a Gaussian distribution that is updated over time with an EKF. Since these approaches employ the information provided by three images at each time step, the matching process is now more robust, which tradeoffs with the computation needed to process 3 images.

In [38][51] SIFT features are extracted from stereo images and the corresponding 3D points in space are computed. These features are considered as the landmarks for the RBPF filtering. They computed the RBPF particle weights from the landmarks innovation; innovation being the distance between predicted (using pose of the particle mapped to corresponding landmark) and observed landmarks. The matching process occurs between the reduced version of 36 dimensional SIFT descriptors of 3 dimensional landmark. An iterative Levenberg-Marquardt nonlinear optimization algorithm addresses the motion model.

Similarly, another RBPF approach which also employs SIFT features is presented in [50] as a vision-based solution to SLAM. This time, the motion model is based on

the robot odometry and the observation model relies on the Mahalanobis distance between the positions of the observed and mapped landmarks.

### *Monocular Cameras*

Monocular vision systems employ one camera and hence are very much useful in cost effective scenarios. Vision-based localization in [45] used just one camera to obtain a visual map of the ceiling. It uses brightness as a metric and employs a particle filter to estimate the pose of the robot. The work presented in [46], combines an image retrieval system based on invariant features with the particle filter-based localization. All of these approach address only global localization and do not deal with SLAM.

In [40] an RBPF-based method for monocular SLAM is experimented. The experiment extracts SIFT features from the images and filters using an Unscented Kalman Filter (UKF) for the robot localization and to update the observed landmarks. SLAM is tackled in [39][42][37] using a single camera, which they named as MonoSLAM. The proposed method that happens to be in real time, extracts reduced number of salient image features through employing Shi and Tomasi's operator. MonoSLAM is limited by the scale factor, which is resolved by initializing the system viewing at a known size pattern. The work in [48] adds an Inertial Measurement Unit (IMU) to an implementation of the inverse depth-based monocular SLAM, reporting an improved accuracy in the estimation of the scale factor of the map. The SLAM approach proposed in [47] introduces the inverse depth parameterization for the undelayed initialization of features.

## CHAPTER 3

### HARDWARE

The existing robotic agent's hardware is equipped with IR sensors mounted on a turret driven by a servomotor, for distance measurement. Although these IR sensors are cost effective, the time to scan the environment is very long and a considerable amount of battery power is required to drive the servomotors. The simulations presented in the thesis are performed by making use of existing robot hardware, by just mounting a camera on the robot's chassis. The camera has a battery of its own and does not use the battery from the chassis.

The following sections will give a brief introduction to the hardware employed in the cooperative robotics testbed, for better comprehension of the environment in which the simulations are being performed. Basically, there are two key parts to the hardware used in evaluating the effect of vision-based localization algorithms: the experimental field and the robots themselves. This chapter will be divided up in the same way. The experimental field will be described first, followed by the robotic hardware.

#### **3.1 Experimental Field**

This section describes the environment employed by simulations of the thesis. The simulation field is housed in an 8 ft. x 16 ft. field designed as a scale-model of an indoor environment consisting of a hallway with three rooms on either side.

The field is constructed by using four 4 x 8 foot x 3/4" medium density fiberboard (MDF) panels for the floors. Each board is attached to the others using straight metal brackets and 3/8" screws. The walls are 12" tall and are also constructed from MDF. The walls are secured to each other and to the floorboards using A66 angle brackets. In order to ensure a smooth floor, pieces of poster board are put across every seam and attached using 1" masking tape. The field is shown in Figure 3.1.



Figure 3.1: Image of the Experimental Field.

### **3.2 Robotic Hardware**

This section provides a brief overview of the existing robot hardware. This will be divided into five subsections each describing one specific aspect: Chassis, Drive System, Sensors, Control System, and Batteries.

### 3.2.1 Chassis

The robot's skeleton is provided by BudgetRobotic's ScooterBot II chassis kit. This kit consists of a 7" diameter round chassis assembly designed for differential drive steering. It consists of the base platform with voids cut out for casing drive wheels and an upper platform for housing the computer and interface board. A third platform was used to serve as a base in mounting the rotating sensor turret or a vision sensor system.

### Drive System

As this is a laboratory grade experiment in an indoor environment, a wheeled design is chosen for robot's drive system, since, the wheeled vehicles perform better on smooth level terrain and can usually carry more payload and travel at faster speeds than a legged vehicle.

The differential drive system provided by the ScooterBot II chassis kit is used as the wheeling system. The kit uses GW Servo S03NXF STD continuous rotation servos, which allows speed control, based on servo timing parameters. These motors weigh 59.6 grams and produce 35 oz-in of torque. They can rotate at a maximum speed of  $60^\circ$  in 0.15 seconds at 4.8 Volts. The wheels used have a diameter of 2.5 inches, giving the robot a maximum speed of 22.16 cm/s. A linearized velocity equation can be determined with servo-timing signal as input and wheel linear velocity as the output:

$$v(t) = \begin{cases} 44.32 \times (t - 1.5) & 1 \geq t \geq 2 \\ -22.16 & t \leq 1 \\ 22.16 & t \geq 2 \end{cases}$$

## Sensors

The SARA protocol employs primarily 2 kinds of sensors: distance and odometric. The distance sensors are employed for observing the world around the robot and building a plot of the environment. The odometric sensors would aid in determining the robot's position in the world. The following subsections give a description about the distance and odometric sensors.

### Distance Sensors

The existing SARA algorithm uses IR sensors as the primary distance sensor. The IR sensors are attached to the turret as shown in figure 3.2.

The idea was to rotate the two sensors, which are facing away from each other at 180° angle so that they could provide 360° coverage. Though this design is very economical, one of the major disadvantages of infrared sensors is the size required to provide good resolution. As the resolution goes up the size of the IR sensor would also increase. The better the resolution the better is the calibration process. More over the infrared sources are vulnerable to the temperature differences in the environment

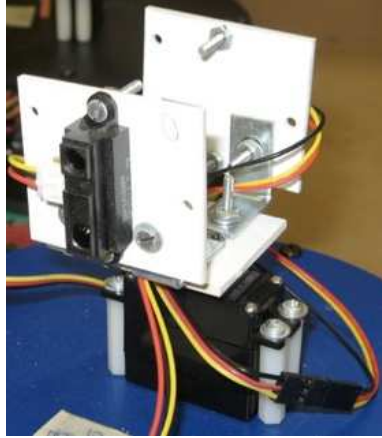


Figure 3.2: Current Distance measurement sensor.

(hot surfaces also generate invisible infra red radiations), which arises a possibility to detect incorrect readings.

In order to circumvent these drawbacks, a vision sensor system that is not vulnerable to the external factors is introduced. The vision sensor is passive and is not affected by the external noise factors. The experiment in the laboratory is conducted using a Panasonic SDR-H40 camera that observes JPG images with a width of 640 pixels and length of 360 pixels. The size of each size would fall around 58 Kilobytes enabling faster processing time. The camera has a focal length of 75.6 mm with the focal point at  $320 \times 280$  pixels (in the image coordinates).

#### *Odometry Sensors*

The current robotic hardware employs a floor encoder system using optical mice. The optical mice have the advantage of converting movements of the surface to  $\Delta x$  and  $\Delta y$ . The mice have a varying resolution from 1 count per mm to 8 count per mm. Due to the inexact nature of the mounting, a calibration procedure from Bonarini et al in [21] is followed to mitigate any systemic errors. This kind of system's main



advantage is that it does not suffer measurement errors from the slippage of the wheel and can be very accurate. An image of the optical mice mounted to the bottom of the robot is as shown in the figure 3.3.

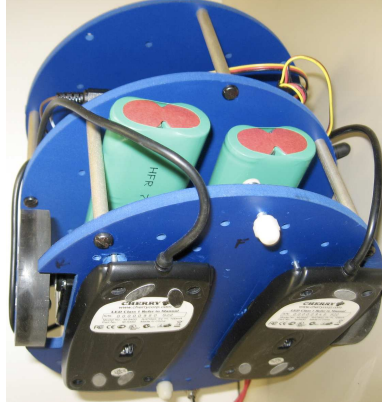


Figure 3.3: Robot mounted with optical mouse and batteries.

## Control System

The brain's role of the robotic agent is taken up the Gumstix platform running embedded Linux inside it. The gumstix platform has an 802.11b wireless card expansion module called Wifistix to communicate with the other robot agents, and has a microcontroller based expansion module called Robostix to monitor and control the peripherals.

The *Gumstix processor* is a 400 MHz Intel X-Scale ARM based chip with 64MB of RAM and 16MB of flash. Some of the limitations that eventually led to perform the simulation are the lack of a floating-point unit and inability to handle a large amount of image processing.

The *Robostix module* contains an AVR ATmega128 micro-controller designed to drive Input Output subsystems and serve as interface between the Gumstix and

hardware peripherals. The commands from the gumstix are converted into electrical impulses to perform the tasks like driving motors, polling the mice or reading the distance sensors and sending the data back to the gumstix.

The *Wifistix* is a Marvell DRCM81 module from Wistron NeWeb Corporation with the Marvell 88W8385 chipset [22]. The chipset supports 802.11b/g both infrastructure and ad-hoc connection modes. An image of the Gumstix along with the attached Wifistix is as shown in the following figure 3.4.

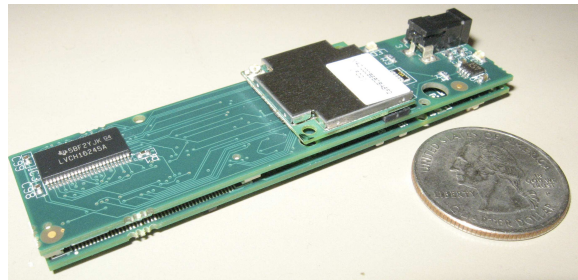


Figure 3.4: Gumstix (bottom) and Wifistix (top).

The *workstation* running Ubuntu Linux operating system (version 8.04 LTS x86-64) plays a major role in performing the heavy image processing and particle filtering. The gumstix platform gathers the entire image and odometric data and exports this data to the Matlab software running on the workstation.

## **Batteries**

The robot hardware is powered by two 7.2V, 2800 mAh batteries. One battery powers the electronics; the sensors and the microcontroller stack, while the other powers six servos. The camera holds its own power supply for the simulations. An image of the batteries mounted on the robot can be found in Figure 3.3.

## CHAPTER 4

### SLAM AND SIFT

This chapter covers the primary algorithms utilized to implement and analyze the effects of introducing vision sensor system in to the existing robotic agent's hardware. Firstly, a mathematical model of SLAM problem is presented followed by that for robot and the landmarks is proposed. Secondly, an explanation of basic Bayesian technique that forms the foundation for the other robust data filtering algorithms is presented. Thirdly, SIFT (*Scale Invariant Feature Transforms*) [23], is presented as an algorithm for dealing with online feature extraction from stereovision system. Finally, an explanation of central concept of SLAM process - the RBPF (*Rao-Blackwell particle filtering*) method for estimating the robot's position is outlined.

#### 4.1 SLAM process and model

##### 4.1.1 SLAM in General

In general SLAM is the problem of building a spatial map of the unknown environment while simultaneously tracking its own position using the partially built map. In an implementation view, the goal of the slam process is to use the observed environment to update the position of the robot since the odometry of the robot is definitely prone to errors.

Figure 4.1 shows a general operating environment of a mobile robot performing Simultaneous Localization And Mapping. A mobile robot (shown with a triangle) in a 3 Dimensional plane with a fixed Global Reference Frame scans the environment for scale invariant landmarks (depicted with coned shapes) and uses these landmarks to update the robot's pose.

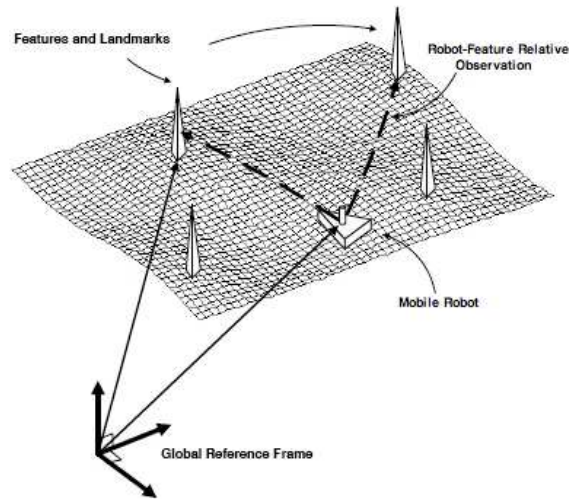


Figure 4.1: Mobile Robot operating scenario.

As this process goes on forever the uncertainty of measurements goes up. Figure 4.2 shows the uncertainty accrued over robot path and estimate of absolute location for one landmark.

Figure 4.3 illustrates the SLAM process with a good detail for a better comprehension:

- 1  $\Rightarrow$  The robot initially measures using its sensors the location of landmarks.
- 2  $\Rightarrow$  The robot moves, and thinks it is in the position indicated by dark triangle given by odometry.
- 3  $\Rightarrow$  The robot once again measures the location of the landmarks using its sensors

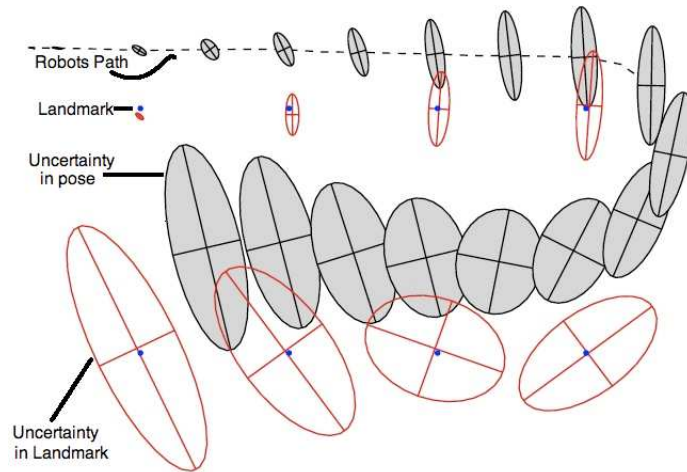


Figure 4.2: Accrued Uncertainty in robot pose and landmark.

but finds out they don't match with where the robot thinks they should be (given the robots location). Thus the robot is not where it thinks it is.

4  $\Rightarrow$  As the robot believes more its sensors than its odometry it now uses the information gained about where the landmarks actually are to determine where it is (the location the robot originally thought it was at is illustrated by the dashed triangle).

5  $\Rightarrow$  In actual the robot location is here. Since, the sensors are not perfect, the robot will not its exact pose. However this estimate is better than relying on odometry alone. The dotted triangle (triangle beneath the dark triangle) represents where it thinks it is; the dashed triangle where odometry told it was, and the last triangle where it actually is.

Figure 4.4 is block diagram showing the elements of SLAM employed for this thesis. SLAM is a recursive process. Since a robot doesn't know its location in an unknown environment; it assumes itself to be in some random place given by samples (a.k.a particles) of a probability distribution function. Referring to the Figure 4.4, when the odometry changes because the robot moves the uncertainty pertaining to

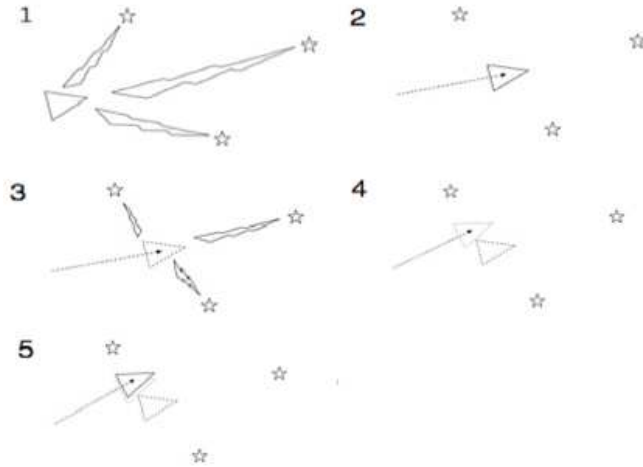


Figure 4.3: SLAM illustration graphically.

the robot's new position is updated in the RBPF filter using Odometry update. Landmarks are then extracted from the environment from the robots new position.

The robot then attempts to associate these landmarks to observations of landmarks it previously has seen. Re observed landmarks are then used to update the robots position in the RBPF. Landmarks that have not previously been seen are added to the RBPF as new observations so they can be re-observed later.

The final step is an optional RBPF Sampling Importance Re-sampling (SIR), which needs to be performed to prevent the depletion of particles [25]. An SIR is performed based on the number of samples present in the current RBPF filter. All these steps will be explained in the succeeding sections. It should be noted that at any point in these steps the RBPF would have an estimate of the robots current position.

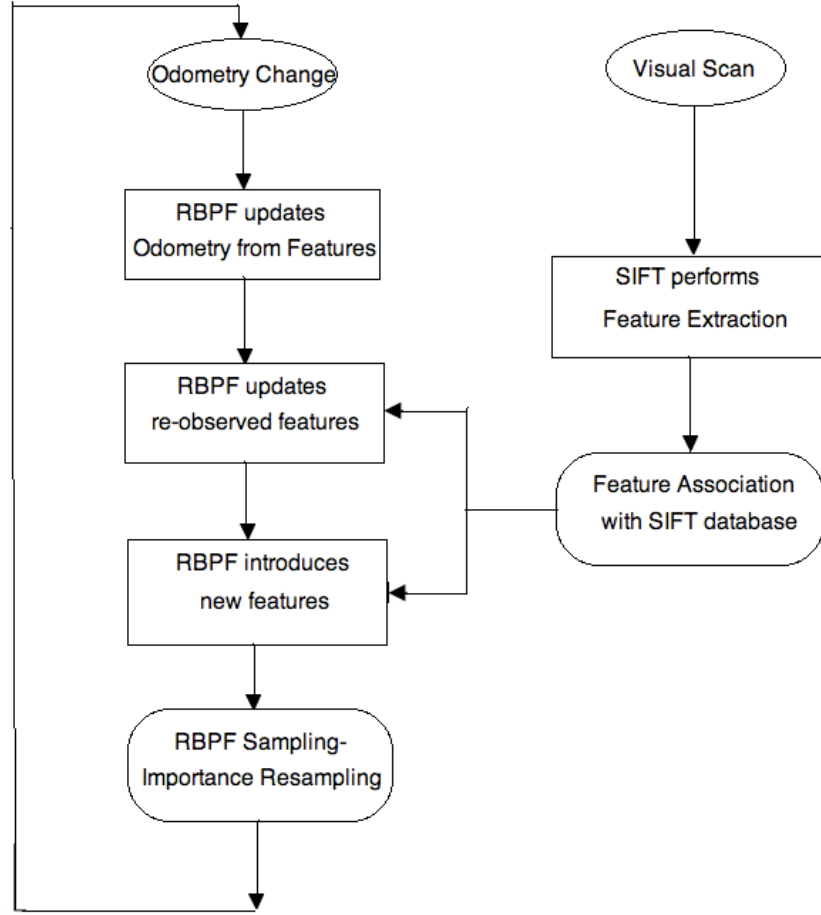


Figure 4.4: SLAM process

#### 4.1.2 SLAM Mathematical Model

Let  $s_t$ ,  $u_t$ , and  $z_t$  be the momentary robot pose, the action, and the observation at time step  $t$ , respectively, and let  $m$  be the map of the environment. The goal of SLAM is to recover the map from sensor measurements

$$p(s_t, m | z^t, u^t) \quad (4.1)$$

where,

$$z^t = \{z_1, z_2, \dots, z_n\}, u^t = \{u_1, u_2, \dots, u_n\}, s^t = \{s_1, s_2, \dots, s_n\}$$

and  $m = \{m_1, m_2, \dots, m_n\}$

In this experiment,  $u_t$  represents the robot pose change between time steps  $t - 1$  and  $t$ , which, in our case, is unknown and will be estimated by means of the optical mice odometry. The map ' $m$ ' is a set of ' $n$ ' landmarks, with each landmark being represented by  $m_j$  ( $j = 1$  to  $n$ ). Also, please note that the superscripted  $^t$  indicates it is a set of values from 1 to  $t$ , where as subscripted  $_t$  indicates, it is a value acquired at the ' $t^{th}$ ' moment.

SLAM process entails 2 sources of information to solve the problem. The models of these sources serve as inputs to the RBPF filter.

## Measurement Model

At the heart of the RBPF is a generative model of sensor measurements, that is, a probabilistic law that specifies the process according to which measurements are generated. This model will be referred to as measurement model and is of the following form:

$$p(z_t | s_t, u_t) = g(m_j, s_t) + \xi_t \tag{4.2}$$



The measurement model is conditioned on the robot pose  $s_t$ , the landmark's map identity  $m_j (j = 1 \text{ to } n)$ , and the specific feature  $z_t$  that is being observed. The model is governed by a deterministic function 'g' distorted by a noise element. The noise at time 't' is modeled by the random variable  $\xi_t$ , which is assumed to be a normal distribution with zero mean and covariance  $R_t$ .

The measurement function is non-linear in nature and involves in calculating the range and bearing of the observing landmark. They are easily evaluated employing simple trigonometry. Figure 4.5 illustrates the same:

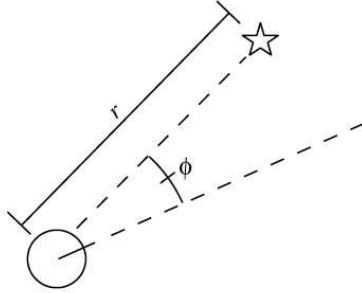


Figure 4.5: Robot observing the Range and Bearing of a Feature.

### Kinematic Motion Model

A second important source of information for solving the SLAM problem is the robotic controls. The parameter  $u_t$  denotes all the motor commands carried out in  $[t - 1, t)$ . This probabilistic law governing the evolution of robot's poses is called the Kinematic motion model and is of the following mathematical form:

$$p(s_t | s_{t-1}, u_t) = h(u_t | s_{t-1}) + \delta_t \quad (4.3)$$

## 4.2 Bayesian Filtering Technique

The concrete foundation for today's most popular data filtering algorithms is Bayesian estimation. Bayesian estimation is rooted on the Bayes' Theorem. Thomas Bayes first introduced the concept 250 years ago. According to the Bayesian rule, if we know a prior belief about a random variable  $x$ , and generated an observation  $y$ , with a known observation model then the prior belief of the random variable can be updated to reduce the deviation from the actual value. Mathematically, it is represented as:

$$p(x|y) \propto p(x) \times p(y|x) \tag{4.4}$$

This technique can be employed for online operation by iteratively executing prediction and update steps. In the *prediction step*, the system state (comprising the robot pose) is propagated in time according to some given transition model (the motion model), leading to the prior distribution of the system state. In the *update step*, this prior is refined according to a given observation model, obtaining the posterior probability density, which already includes the information available up to some given time step (this is a Markov Process). Figure 4.6 is an illustration of the Bayesian estimation:

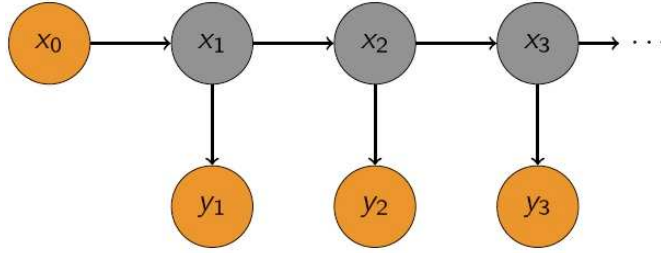


Figure 4.6: Online operation of Bayesian Rule

### 4.3 Scale Invariant Feature Transforms - SIFT

Measurement model of the SLAM deals with a whole lot of image processing. The RBPF data filter employs the features output from the images to update the predicted robots pose. We employ SIFT as an image processing algorithm to generate features from the image frames scanned from the environment. SIFT was developed by Lowe [23] for generating image features. The features are invariant to image translation, scaling, rotation, and partially invariant to illumination changes and 3D projections. These properties make these features most appropriate for a robust SLAM. It is because, as the mobile robots move around in an environment, landmarks are observed over time, but from different angles, distances or under different illumination. At each time frame, SIFT features are extracted from the left and right images to extract the range and bearing of the landmark in scene. These extracted parameters are compared to the existing map to correct the position of the corresponding landmarks along with the robots position. Stereo matching employed in this work is discussed in the Chapter 5. The following subsections provide a brief outline of SIFT algorithm.

### 4.3.1 Scale-space extrema detection

The first stage of keypoint detection is to identify locations and scales that can be repeatedly assigned under differing views of the same object. Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space. In an outline, finding the interest points that are extrema (maximum or minimum) with respect to both scale and space involves:

1. Creating several versions of the original image with an increasing Gaussian blurring applied (an image pyramid - a type of scale-space).
2. Subtract an image from its more-blurred neighbor image - Difference of Gaussians (DoG).
3. Mark any points that are maximums or minimums in their  $3 \times 3 \times 3 = 27$  neighborhood (9 pixels in the less-blurred image, 9 pixels in its own image and 9 pixels in the more-blurred image)

### 4.3.2 Keypoint localization

Scale-space extrema detection produces too many keypoint candidates, some of which are unstable. This step performs a detailed fit to the nearby data for accurate location, scale, and ratio of principal curvatures. This information helps to reject all the keypoints that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. The output keypoints that are invariant to a wide scale space are then processed through the following 3 stages for accurate localization.

1. Interpolation of nearby data for accurate position
2. Discarding low-contrast keypoints
3. Eliminating edge responses

### **4.3.3 Orientation assignment**

In this step, each keypoint is assigned one or more orientations based on local image gradient directions. This is the key step in achieving invariance to rotation as the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation.

### **4.3.4 Keypoint descriptor**

Previous steps found keypoint locations at particular scales and assigned orientations to them. This ensured invariance to image location, scale and rotation. This step computes a descriptor vector for each keypoint such that the descriptor is highly distinctive and partially invariant to the remaining variations such as illumination, 3D viewpoint, etc. These descriptors aid in performing the object recognition, as well as for searching through a large database of descriptors.

A brief outline of SIFT algorithm has been described; a complete discussion is found in [23]. The following section describes the operation of the Rao-Blackwell particle filter.

## 4.4 Rao-Blackwell Particle Filter - RBPF

The meat of the SLAM process lies in the filtering algorithm. The estimation of the robot's pose and also the process of approximating the landmark positions will be carried out by the filtering algorithm. This thesis works with the RBPF filter due to its inherent advantages of scalability and the ability to distinguish between ambiguities when they arise. Following discussion provides a derivation of general Bayes' filter problem towards the Rao-Blackwell filter and then a pseudo-code for RBPF filtering algorithm is also given.

### 4.4.1 Reintroducing SLAM

From equation (4.1) the SLAM problem is formally given by:

$$p(s_t, m | z^t, u^t)$$

The notation is same the one used in defining the problem. Employing the preliminary Bayesian estimation technique the solution for obtaining the robot and map estimates would take the following recursive form:

$$\begin{aligned} p(s^t, m | z^t, u^t) &= Bel(s_t, m) \\ &= \eta p(z_t | s_t, u_t) \int p(s_t | u_t, s_{t-1}) p(s_{t-1}, m | z^{t-1}, u^{t-1}) ds_{t-1} \\ &= \eta p(z_t | s_t, u_t) \int p(s_t | u_t, s_{t-1}) Bel(s_{t-1}, m) ds_{t-1} \end{aligned} \quad (4.5)$$

Where  $\eta$  is normalizing constant and it is independent of any of the variables.  $\eta$  is for ensuring that the values sum to 1. One might consider that the posterior (equation 4.5) captures all relevant information, for robotic SLAM. However, there are other, better distributions that can be estimated in SLAM. The RBPF algorithm, estimates the posterior over robot paths, not just momentary poses, along with the map. Here is the new version of SLAM with RBPF path estimation:

$$p(s^t, m|z^t, u^t)$$

At a first glance, estimating the entire path posterior might appear to be demanding greater number of CPU cycles. Since, with the increase in the path length, the sampling space would also increase. Such a thing would be difficult to implement in real time situations. But, some specific types of filters calculate posteriors over robot paths as efficiently as for momentary poses. The true motivation behind this approach comes from the observation that the posterior or  $Bel(x_t)$  can be factorized into a product of smaller terms. The final form of the filter after derivations [27] is

$$p(s^t, m|z^t, u^t) = p(s^t|z^t, u^t) \prod_{n=1}^N p(m_n|s^t, z^t) \quad (4.6)$$

#### 4.4.2 Factoring the SLAM Posterior - $Bel(x_t)$

Murphy [53] has observed a key insight in the posterior or  $Bel(x_t)$ , which states that the posterior can be written in the factored form given by the following product form:

$$p(s^t, m | z^t, u^t) = p(s^t | z^t, u^t) \prod_{n=1}^N p(m_n | s^t, z^t) \quad (4.7)$$

This factorization states that the calculation of the posterior over paths and maps can be decomposed into  $N + 1$  recursive estimates; one estimate over robot paths, and  $N$  separate estimators over feature locations conditioned on path estimate. This property is a trivial property of SLAM. Consider, the following explanation in figure 4.7:

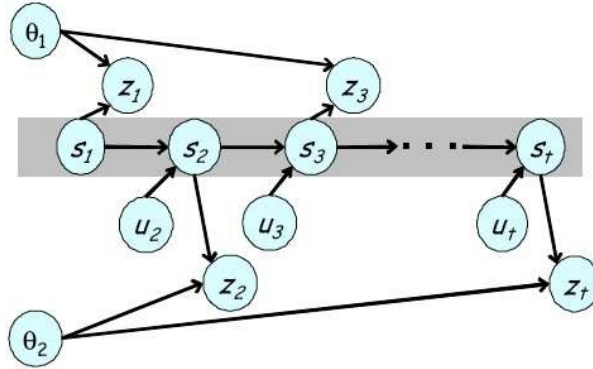


Figure 4.7: SLAM Problem for realizing the factored solution.

The SLAM problem involves a robot moving from pose  $s_1$  influenced under a sequence of controls,  $u_1, u_2, u_3 \dots u_t$ . As the robot moves, it measures landmarks. At time  $t = 1$ , it observes landmark  $\theta_1$ . The measurement is denoted  $z_1$  (range and bearing). At time  $t = 2$ , it observes the other landmark,  $\theta_2$ , and at time  $t = 3$ , it observes  $\theta_1$  again. The SLAM problem is concerned with estimating the locations of the landmarks and the robot's path from the controls  $u_t$  and the measurements  $z_t$ . The gray shading illustrates a conditional independence relation. As the picture



illustrates, each measurement is a function of position of corresponding feature and the robot pose at time of measurement. Knowledge of robot path is d-separates the individual feature estimation. In other words it says that the knowledge of one feature in no way helps in knowing the locations of the other features. A mathematical derivation of equation 4.7 is given in [27]. The following section describes the process of data association in the RBPF algorithm.

### 4.4.3 RBPF Data Association

RBPF exploits the factored representation by maintaining  $N + 1$  filters, one for each factor in equation 4.7. The robot pose (first term in equation 4.7) is calculated using a particle filter [54]. This particle filter's computations time is independent of length of the path. The remaining  $N$  (conditional) posteriors over feature locations (second term in equation 4.7) are calculated by extended Kalman filters (EKFs). The individual EKFs estimate a single landmark pose conditioned on the robot paths. This mode of operation makes EKF low dimensional. Hence, each particle possesses its own set of EKFs. In total there are  $NM$  EKFs, one for each feature in the map and one for each landmark.

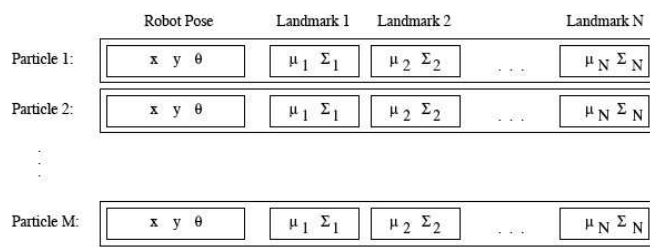


Figure 4.8: Particles in RBPF.

Here  $M$  being the number of particles employed for the SLAM process and  $N$  number of features extracted from the SIFT algorithm. Figure 4.8 illustrates the structure of  $M$  particles. In equation form a particle is given by:

$$S_t^{[m]} = \langle s^{t,[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]} \dots \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \rangle$$

The  $[m]$  represents the particle index,  $\mu_{n,t}^{[m]}$  and  $\Sigma_{n,t}^{[m]}$  are mean and covariance of the landmark respectively.  $S^{t,[m]}$  is the path estimate and  $S_t^{[m]}$  is the  $m^{th}$  particle. Calculating the posterior or  $Bel(S_t, m)$  at time  $t$  from the one at time  $t - 1$  involves generating a new particle set  $S_t$  from  $S_{t-1}$  (particle set one time step earlier). This new particle set incorporates a new control  $u_t$  and a measurement  $z_t$ . The steps needed to these updates are listed below:

### **Kinematic Motion Model realized by extending the path posterior**

RBPF uses the robot control  $u_t$  for sampling new robot pose  $S_t$  for each particle in  $S_{t-1}$ . For example, consider the situation of the  $m^{th}$  particle in  $S_{t-1}$ , denoted by  $S_{t-1}^{[m]}$ . RBPF samples the pose  $S_t$  in accordance with the  $m^{th}$  particle, by drawing a sample according to the kinematic motion model (with respect to  $m^{th}$  particle):  $S_t^{[m]} \approx p(S_t | S_{t-1}^{[m]}, u_t)$ . Here  $S_{t-1}^{[m]}$  is the estimate for the robot location at time  $t - 1$ , inside the  $m^{th}$  particle. The resulting sample  $S_t^{[m]}$  is then added to a temporary set of particles, along with the path of previous poses,  $S^{t-1,[m]}$ . The operation requires constant time per particle independent of map size  $N$ .

## Measurement Model to update the landmarks

Next, RBPF updates the posterior of the landmark estimates, represented by the mean  $\mu_{n,t-1}^{[m]}$  and the covariance  $\Sigma_{n,t-1}^{[m]}$ . The updated values are then added to the temporary particle set, along with the new pose. The updating process depends on landmark's observable at time  $t$ . If the landmark is not observed then the posterior of the landmark remains unchanged. If the landmark is re-observed then the status of the landmark is updated with the following simple equation:

$$p(m_j|s^t, z^t) = \eta p(z_t|s_t, m_j)p(m_j|s^{t-1}, z^{t-1})$$

In effect the following is the transition:

$$\langle \mu_{n,t}^{[m]}, \Sigma_{n,t}^{[m]} \rangle = \langle \mu_{n,t-1}^{[m]}, \Sigma_{n,t-1}^{[m]} \rangle$$

Since, this update process is performed by an EKF, RBPF also employs the same linearization process as EKF to extract the values from the measurement model. In specific, it uses a Taylor expansion to denote the function  $g$  since the function  $g$  is Gaussian. The new mean and covariance are obtained using standard EKF update [55].

## RBPF Re-sampling

As a final step RBPF draws  $M$  particles (with replacement), from the constructed temporary particles set. These particles then form the new set of particles at time  $t - S_t$ . The need to resample arises because the particles in the temporary set are not distributed as per the desired posterior. RBPF while extending the posterior

generates poses  $S_t$  by catering to the most recent control  $u_t$ . It does not consider the measurement  $Z_t$ .

Re-sampling is a common technique in particle filtering to correct such mismatches. This scenario is depicted in the figure 4.9. From the figure, the dashed line symbolizes the *proposal distribution*, which is the distribution at which particles are generated, and the solid line is the target distribution [41]. In RBPF, the proposal distribution is independent on  $Z_t$ , but the target distribution is dependent on  $Z_t$ . By weighing particles as shown in the bottom of this figure, and re-sampling according to those weights, the resulting particle set approximates the target distribution. The weight of each sample used in the re-sampling step is called the *importance factor*.

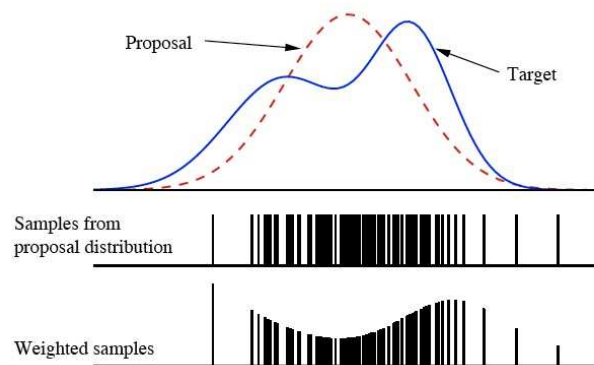


Figure 4.9: Re-sampling to approximate target distribution.

The importance factor is given by the following equation [27] with  $Q_t^{[m]}$  being the covariance:

$$w_t^{[m]} \approx \eta \left| 2\pi Q_t^{[m]} \right|^{-\frac{1}{2}} e^{-\frac{1}{2} (z_t - \hat{z}_t^{[m]})^T Q_t^{[m]-1} (z_t - \hat{z}_t^{[m]})}$$

From the derivation we note that the execution time of the update does not depend on the total path length  $t$ . Since, only the most recent pose is used in the

process of generating a new particle at time  $t$ . As a result, the past poses can safely be discarded. This has the pleasing consequence that neither the time requirements, nor the memory requirements of RBPF depend on time.

## CHAPTER 5

### EXPERIMENT AND RESULTS

The experiment setup consists of 3 major components: Matlab software on which the simulations are performed, the modified hardware test bed in the cooperative robotics laboratory and the robot mounted with Panasonic camera. SIFT and RBPF filtering algorithms are written in Matlab. Odometry data and images for the experiment are collected manually from the test bed. The first section details the experimental setup and the rest of the chapter includes the results obtained from the experiments.

#### **5.1 The Setup**

This section first details the set up of the hardware test bed, followed by the set up process employed for mapping the stereo images to the real world coordinates of the landmark.

##### **5.1.1 Modified Hardware Testbed**

The existing cooperative robotics test bed is as shown in the figure 3.1. This experiment fixes the world coordinate system  $(0,0,0)$  at the position of the robot shown in figure 5.1. The x-axis and y-axis are along the blue lines as shown in the figure 5.1. The landmarks are depicted with the black tape marks on the testbed.

Imaging this test bed with the camera and applying SIFT extracted very few features from the environment, which are not sufficient for SLAM. For the RBPF algorithm the more the number of features, the better will be the estimate of the robot's location. The number of observable features is increased by sticking a black tape in known locations. The modified test bed is shown in 5.1:

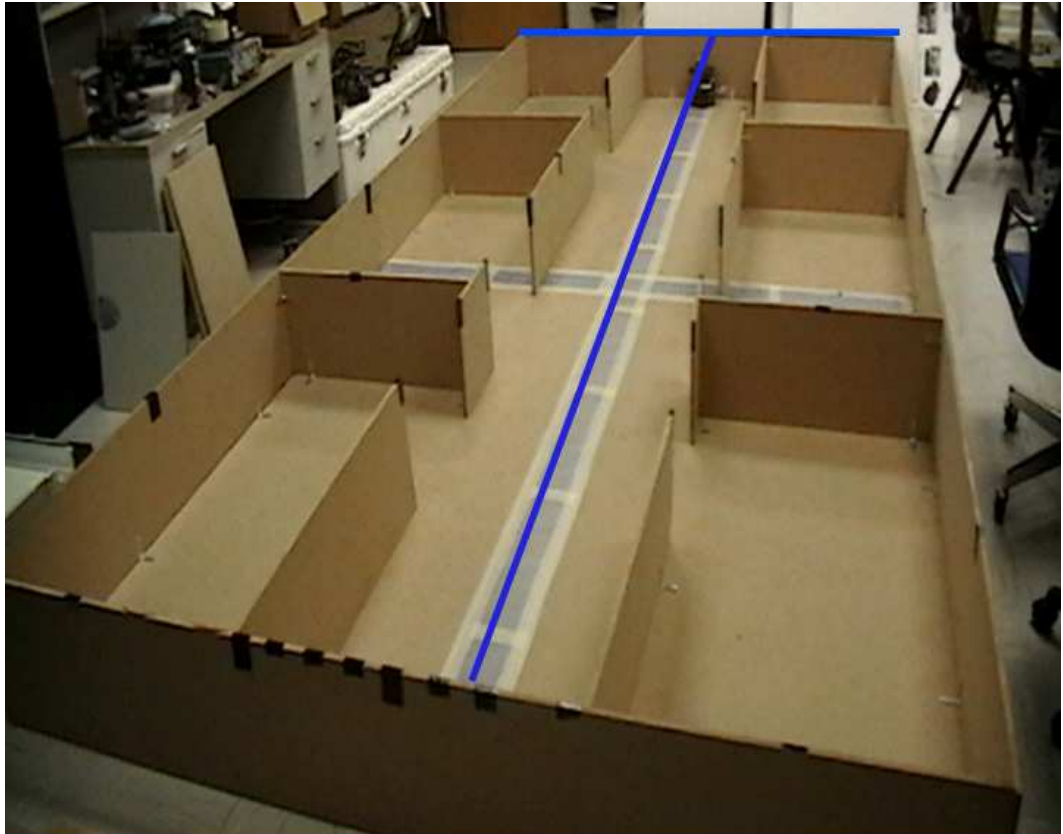


Figure 5.1: Modified CRR testbed.

### 5.1.2 Stereo Imaging

A stereo image involves a scene being observed with 2 monocular cameras separated by a baseline distance. Using these 2 images the coordinates of the feature can be constructed. In the experiment a Panasonic monocular camera is first mounted

on the left side of the chassis to acquire the reference images (left images) and then it is mounted on the right side of the chassis to acquire the right image. The Matlab SIFT software matches the features and extracts the coordinates of the feature in the global frame. The cameras are mounted as shown in the figure 5.2.



Figure 5.2: Simulating stereo imaging. Right mounted and left mounted on chassis.

The baseline is the distance between the optical centers of the right and left cameras. This set up of cameras are mounted with a baseline distance of 12 centimeters to form a stereovision system. The following is a brief explanation for extracting the landmark locations from the left and right images obtained through the cameras. The discussion refers to figure 5.3.

The left and right cameras obtain images of the same scenario but with a different angle of view. The angle between the left and right camera is called as the parallax



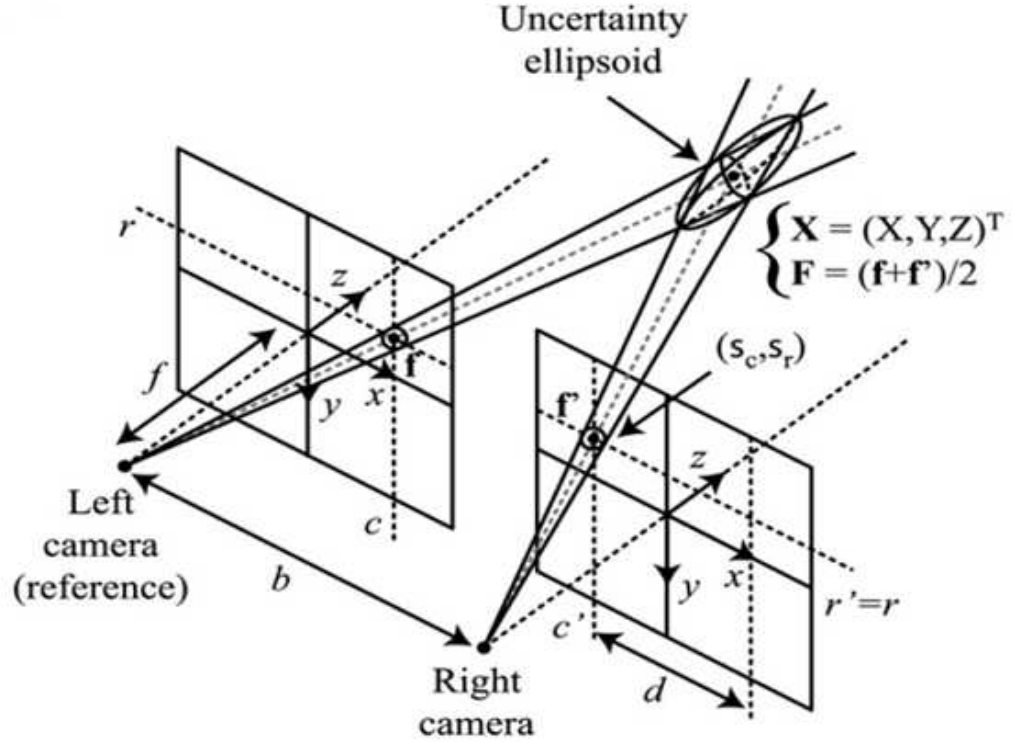


Figure 5.3: Extracting  $[x, y, z]$  from left and right images.

angle. This angle of view called the parallax angle greatly helps in estimating the  $z$  coordinate (perspective or depth) of the world coordinate in the robots algorithms. The left and right images when processed with SIFT provide us with the scale invariant features of the images. The features acquired from the left image are matched to that of the features of the image obtained from the right camera using the corresponding sift descriptors. Using these matched points and the camera configuration the most likely 3D coordinates of the landmark are estimated by projecting them back to the space. The uncertainty in 3D landmark due to image quantization and feature detection process is also considered.

The 3D coordinates of a point can be calculated using the following three equations:

$$X = (c - c_0)\frac{b}{d}; \quad Y = (r - r_0)\frac{b}{d}; \quad Z = f\frac{b}{d}$$

where  $(r, c)$  are the coordinates of interest point in the left image (reference image) and,  $(r_0, c_0)$  are the coordinates of principal point (focal point) in the reference image; and  $b, d, f$  are stereovision parameters namely baseline, disparity and the focal length respectively. The disparity is the distance between the points seen in the left image to right image. The uncertainty covariance matrix for  $r, c$  and  $d$  is also calculated [56]. The SIFT descriptors for the matched point is given by  $F$  which is the average of corresponding descriptors of features from left and right images. The data gathered from these images is stored in a table. RBPF algorithm is fed with the data from this table.

## 5.2 Results

The experiment is conducted with the data acquired from the robot. Firstly, the world coordinates are obtained by matching the right and left images at each time step and mapping them back to the real world coordinates (process described in section 5.1.2). These real world coordinates are stored in a database to be fed to the RBPF online filtering algorithm. Figure 5.4 is an output from the SIFT matching algorithm. The figure shows matching between left and right images taken at  $(20,0)$  coordinate. The green lines connect the matched features. There are few incorrect matches between the stereo images. This incorrect matches account to the manual data collection process employed for the experiment. RBPF being a probabilistic model the incorrect matches are also accounted.



Figure 5.4: SIFT feature matches between left and right images at  $(20,0)$ .

Figure 5.5 is the final output after the simulation has run. The green star is the actual position of the landmark in the global map. The red dot is a robots estimate of the landmark position in the map. The cyan lines connect the robot to the visible landmarks in the map. The blue line starting from  $(0,0)$  is an estimate of path taken by robot. The line with red dots is the real path of the robot.

When the robot reaches the end of the simulation, the robot would have constructed a map of the entire environment and also evaluated the complete path of the environment. Figure 5.6 is a plot of the robot's path estimate.

The spread of the robot's path estimation is along the actual robot path taken along line  $(x_t, 0)$ .  $x_t$  is  $x$  coordinate of the robot at time ' $t$ '. The picture shows that with the progress of time the effective estimate of the path does not exceed a limit of -4 centimeters.

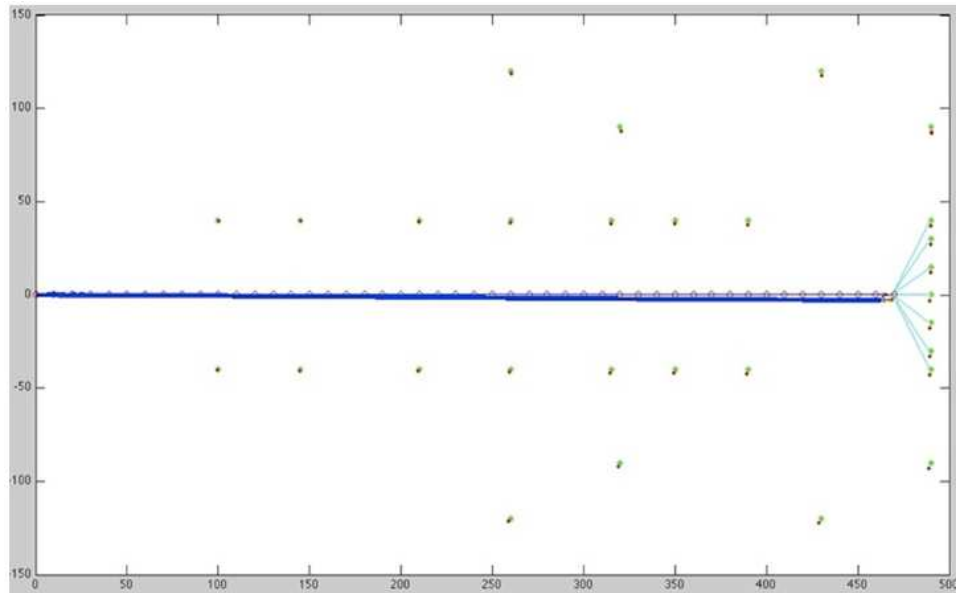


Figure 5.5: Final Output of RBPF simulation.

Figure 5.7 is a plot of the estimates of feature 21's location (feature 21 is at (480,0) coordinates) in the real world. The plot shows the estimates approximated by 100 RBPF particles.

The Extended Kalman Filter running inside the particles had estimated the particle position to an accuracy of 99%. The estimate with lowest covariance is chosen to be the actual estimate for that time step. The best estimate of the feature captured in the map is the estimate acquired before the robot loses the view of the feature.

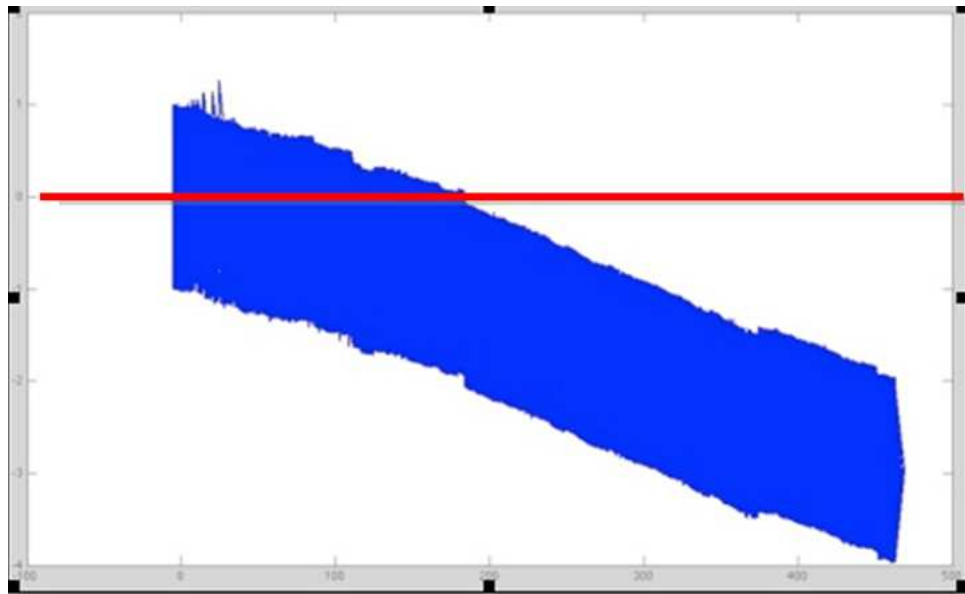


Figure 5.6: Plot of path estimated by the robot.

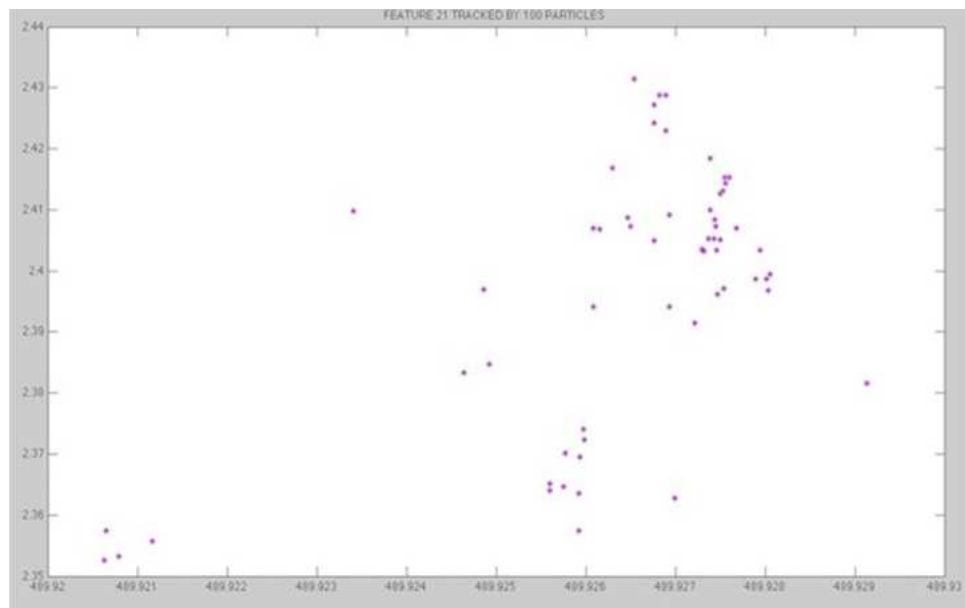


Figure 5.7: Feature 21 estimate by 100 particles.

## CHAPTER 6

### CONCLUSION

This thesis described a partially simulated experiment to improve the localization in the cooperative robotics testbed by employing a stereovision system. The simulation tests have shown that the error in the estimated robot pose is low. The simulation also shows that the error propagation is linearized with time also, the uncertainty in a landmark's location goes down with subsequent observations of the landmark. The existing issue of localization in the hardware test bed has been addressed to a large extent.

#### **Future Work**

As part of the direction to the future work, this simulation environment can be easily programmed on to the real world robot with C and C++ languages. Data used in this thesis is for a straight-line propagation of the robot, which can be extended to gather random data from the environment. With the installation of the new powerful hardware that can also cater for the floating point operations, the SIFT and Rao-Blackwell particle filters can be implemented to run effectively. Utilizing a lighter version of the SIFT would definitely increase the performance of the system; a study for using lighter versions of the algorithm can be made. The Rao-Blackwell Particle filter (part of the fast-SLAM 1.0) used in this version suffers from few ambiguous situations; using the later versions would circumvent these situations.

## BIBLIOGRAPHY

- [1] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (4) (2002) 509-522.
- [2] A.C. Berg, T.L. Berg, J. Malik, Shape matching and object recognition using low distortion correspondences, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, vol. 1, 2005.
- [3] D.G. Lowe, Object recognition from local scale-invariant features, in: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999.
- [4] K. Hosoda, K. Sakamoto, M. Asada, Trajectory generation for obstacle avoidance of uncalibrated stereovisual servoing without 3D reconstruction, in: *Proceedings of IEEE/RSJ International Conference on the Intelligent Robots and Systems, IROS, Human Robot Interaction and Cooperative Robots*, vol. 1, 1995.
- [5] J. Michels, A. Saxena, A.Y. Ng, High speed obstacle avoidance using monocular vision and reinforcement learning, in: *Proceedings of the 22nd International Conference on Machine Learning*, vol. 2, ACM Press New York, NY, USA, 2005, pp. 593-600.
- [6] K. Sabe, M. Fukuchi, J.S. Gutmann, T. Ohashi, K. Kawamoto, T. Yoshigahara, Obstacle avoidance and path planning for humanoid robots using stereo vision, in: *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, vol. 1, 2004.
- [7] D. Murray, J.J. Little, Using real-time stereo vision for mobile robot navigation, *Autonomous Robots* 8 (2) (2000) 161-171.
- [8] I. Ohya, A. Kosaka, A. Kak, Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing, *IEEE Transactions on Robotics and Automation* 14 (6) (1998) 969-978.
- [9] J. Kosecka, F. Li, Vision based topological Markov localization, in: *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, vol. 2, 2004.

- [10] I. Ulrich, I.R. Nourbakhsh, Appearance-Based Place Recognition for Topological Localization, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, 2000, pp. 1023-1029.
- [11] A.J. Davison, I. Reid, N. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 1052-1067.
- [12] L.M Paz, P. Pinies, J.D. Tardos, J. Neira, Large-scale 6-dof slam with stereo-inhand, IEEE Transactions on Robotics 24 (5) (2008) 946-957.
- [13] R. Sim, P. Elinas, M. Griffin, J.J. Little, Vision-based SLAM using the Rao-Blackwellised particle filter, IJCAI Workshop on Reasoning with Uncertainty in Robotics, RUR, 2005.
- [14] Leonard, Durrant-Whyte: Mobile robot localization by tracking geometric beacons:
- [15] S. Thrun, W. Burgard, D. Fox, Probabilistic robotics, The MIT Press, 2005
- [16] C. Wilson “Hardware Testbed for Collaborative Robotics Using Wireless Communication”, Master’s thesis, Auburn University, 2009
- [17] A. Ray, “Cooperative Robotics using Wireless Communication,” Master’s thesis, Auburn University, 2005
- [18] J. Liu and J. Wu. Multi-Agent Robotic Systems. CRC Press: London, 2001.
- [19] L. E. Parker, “Current state of the art in distributed autonomous mobile robotics,” in Distributed Autonomous Robotic Systems 4, Springer-Verlag Tokyo, 2000, pp 3-12.
- [20] <http://www.habmigern2003.info/Sensor-technology.html>
- [21] A. Bonarini, M. Matteucci, and M. Restelli, “A kinematic-independent dead reckoning sensor for indoor mobile robotics,” Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, vol. 4, pp. 3750-3755 vol.4, Sept.-2 Oct. 2004.
- [22] Gumstix, “Gumstix Wiki - Wifi,” [http://docwiki.gumstix.org/Frequently asked questions/Wifi](http://docwiki.gumstix.org/Frequently%20asked%20questions/Wifi). [Online]. Available: [http://docwiki.gumstix.org/Frequently asked questions/Wifi](http://docwiki.gumstix.org/Frequently%20asked%20questions/Wifi).
- [23] D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” Int’l J. Computer Vision, vol. 2, no. 60, pp. 91-110, 2004.
- [24] S. Thrun, W. Burgard, D. Fox, Probabilistic robotics, The MIT Press, 2005.



- [25] G. Casella and C. P. Robert, "Rao-Blackwellisation of sampling schemes," *Biometrika*, vol. 83, no. 1, pp. 81-94, 1996.
- [26] Nixon, Aguado. "Feature Extraction and Image Processing (2002)" ISBN 0750650788
- [27] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *AAAI*, 2002.
- [28] Smith, Self, Cheesman: Estimating uncertain spatial relationships in robotics
- [29] Leonard, Durrant-Whyte: Mobile robot localization by tracking geometric beacons:
- [30] Brooks, R. A. 1982. Symbolic Error Analysis and Robot Planning. *Int. J. Robotics Res.* 1(4): 29-68
- [31] Taylor, R. H. 1976. A Synthesis of Manipulator Control Programs from Task-Level Specifications. AIM-282. Stanford, Calif.: Stanford University Artificial Intelligence Laboratory.
- [32] Chatila, R. and Laumond, J-P. 1985. Position Referencing and Consistent World Modeling for Mobile Robots. *Proc. IEEE Int. Conf. Robotics and Automation*. St. Louis: IEEE, pp. 138-145.
- [33] Leonardo, Alessandro, Geraldo "Simultaneous Localization And Map Building by a mobile robot using sonar sensors".
- [34] A. Elfes, "Sonar-based real-world mapping and navigation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 3, pp. 249-265, June 1987.
- [35] P.Kimball, S.Rock "Sonar-Based Iceberg-Relative AUV Localization", August, 2009.
- [36] C. Wouter "Kinematics of laser localization", Wageningen University, The Netherland.
- [37] A.J. Davison, I. Reid, N. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (6) (2007) 1052-1067
- [38] R. Sim, P. Elinas, M. Griffin, J.J. Little, Vision-based SLAM using the Rao-Blackwellised particle filter, *IJCAI Workshop on Reasoning with Uncertainty in Robotics, RUR*, 2005.
- [39] A.J. Davison, Real-time simultaneous localization and mapping with a single camera, in: *Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2003, pp. 1403-1410.

- [40] M. Li, B. Hong, Z. Cai, R. Luo, Novel Rao-Blackwellized particle filter for mobile robot SLAM using monocular vision, *International Journal of Intelligent Technology* 1 (1) (2006) 63-69.
- [41] S. Se, D. Lowe, J. Little, Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks, *The International Journal of Robotics Research* 21 (8) (2002) 735.
- [42] A.J. Davison, Y.G. Cid, N. Kita, Real-time 3D SLAM with wide-angle vision, in: *Proc. IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, 2004.
- [43] E. Menegatti, A. Pretto, A. Scarpa, E. Pagello, Omnidirectional vision scan matching for robot localization in dynamic environments, *IEEE Transactions on Robotics* 22 (3) (2006) 523-535.
- [44] H. Tamimi, H. Andreasson, A. Treptow, T. Duckett, A. Zell, Localization of mobile robots with omnidirectional vision using Particle Filter and iterative SIFT, *Robotics and Autonomous Systems* 54 (2006) 758-765.
- [45] F. Dellaert, W. Burgard, D. Fox, S. Thrun, Using the CONDENSATION algorithm for robust, vision-based mobile robot localization, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2 (1999) 588-594.
- [46] J. Wolf, W. Burgard, H. Burkhardt, Robust vision-based localization for mobile robots using an image retrieval system based on invariant features, in: *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, 2002.
- [47] J. Civera, A.J. Davison, J. Montiel, Inverse depth parametrization for monocular slam, *IEEE Transactions on Robotics* 24 (5) (2008) 932-945
- [48] P. Pinies, T. Lupton, S. Sukkarieh, J.D. Tards, Inertial aiding of inverse depth SLAM using a monocular camera, in: *IEEE International Conference on Robotics and Automation*, 2007, pp. 2797-2802.
- [49] M.N. Dailey, M. Parnichkun, Landmark-based simultaneous localization and mapping with stereo vision, in: *Proceedings of the Asian Conference on Industrial Automation and Robotics*, 2005.
- [50] A. Gil, . Reinoso, O. Martinez Mozos, C. Stachniss, W. Burgard, Improving data association in Vision-based SLAM, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2076-2081.

- [51] R. Sim, P. Elinas, M. Griffin, A. Shyr, J.J. Little, Design and analysis of a framework for real-time vision-based SLAM using Rao-Blackwellised particle filters, in: Proceedings of the 3rd Canadian Conference on Computer and Robot Vision, 2006, pp. 21-29.
- [52] D. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* 60 (2) (2004) 91-110.
- [53] K. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 1999.
- [54] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods In Practice*. Springer Verlag, New York, 2001.
- [55] P. Maybeck. *Stochastic Models, Estimation, and Control, Volume 1*. Academic Press, Inc, 1979.
- [56] F. Moreno, J. Blanco, J. Gonzalez, Stereo vision-specific models for particle filter-based SLAM. *Robotics and Autonomous Systems*, volume 57, issue 1, pp 64-74, 2008.