Vision Inspired Simultaneous Localization and Mapping: The Algorithm

By

Shiva Karthik Reddy Koyya

A Research Paper Submitted

in

Partial Fulfillment

of the

Requirements for the Degree of

MASTER OF SCIENCE

in

Electrical Engineering

Approved by:

PROF____

(Dr. Ferat Sahin, Research Advisor)

PROF____

(Dr. Sohail A. Dianat, Department Head)

DEPARTMENT OF ELECTRICAL AND MICROELECTRONIC ENGINEERING

KATE GLEASON COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

AUGUST 2016

Acknowledgments

Completion of this project was not possible without the able guidance of my advisior Dr. Feart Sahin.He motivated me and belived in me which kept me going during difficult stages of the project.

I would like to extend my gratitude to Mathew Haywood under whos guidance and I did the the initial part of my project.

Shitij Kumar you acted as a helping hand clearing my doubts and giving me ideas when ever i came up to you. Your help and guidance is much appreciated.

Lastly I would like to thank my faimly and friends for thier support though out the project.

Abstract

Vision Inspired Simultaneous Localization and Mapping: The Algorithm

Shiva Karthik Reddy Koyya

Supervising Professor: Dr. Ferat Sahin

Vision inspired Simultaneous Localization and Mapping algorithm is a improvement of ORB SLAM which utilizes an improved Oriented Rotated Brief features for fast features matching. The project works in real time in a static indoor environment on an Adept Mobile robot (Amigobot) equipped with Xtion Pro Live as Sensor. The major goal of this project was to make an algorithm form the scratch capable of Localizing the Robot with high accuracy and in the process creating a sparse 3D map. The major contribution of this project is in increasing the feature (Landmark in context of Simultaneous localization and mapping) matching speed by applying median filter on images before using Oriented Rotated Brief (ORB) which not only gives more distinctive feature point but improves the matching speed when using RANSAC for feature matching, which overall effects the real-time performance of the algorithm and the localization accuracy of the robot. The use of median filter improves the data correspondence between the views which has effect on the convergence rate of the system to correct values. Rao Blackwellised Particle filter is used as backend for the state estimation of the SLAM problem.

Contents

Acknowledgments **2**

Abstract **3**

1 Introduction **1**

2 Literature Survey **3**

2.1 Simultaneous Localization and Mapping 3

2.1.1 Kalman Filter SLAM 5

2.1.2 Rao Blackwellized Particle Filter SLAM 7

2.2 3D sensors in context of SLAM 9

2.3 Visual SLAM 10

3 Proposed Method and Experiments **13**

3.1 FastSLAM Backend 13

3.2 Image Processing 15

3.3 ROS and Robot Model 18

4 Results and Discussions **19**

5 Conclusion **23**

6 Future Work **24**

Bibliography **25**

List of Tables

4.1 The Results for ORB and Median + ORB are listed along with the number of features extracted from each image, number of good correspondence made after RANSAC and time taken for making those correspondence 19
4.2 Localization accuracy of ORB and ORB + median are compared, when the robot is moving in straight line. The table shows the correspondence made in each step in terms of the number of matches found / number of good correspondence made. 22
4.3 Timing chart for the time taken by ORB and ORB + median particle filters(pf) to run. . . . 22

List of Figures

2.1 The figure shows the data association problem due to uncertainty in the pose of the robot, i.e. if the robot is the as shown in left its assumed that the measurements are from two left most landmarks and when the robot pose is assumed as shown in the right the measurements are assumed from the rightmost landmarks, which leads to uncertainty of measurement. 5

2.2 The figure is graphical model of the SLAM problem where X, U, Z, m have their usual meaning In FastSLAM the poses of the robot are known exactly so the observations become conditionally independent and since observations are independent so the map states are also independent. In particle Filter each particle has its belief of the robot pose and map of the environment. 8

2.3 the figure shows the basic pinhole camera model, which describes the relationship between the world coordinate and the image plane. 11

3.1 Outline of the proposed algorithm. The detailed explanation of each module is presented in further in the chapter 14

3.2 The figure depicts the general flow of the Image processing part of the SLAM 16

3.3 1000 ORB features extracted using Open CV in an indoor office Environment 16

3.4 The image on the left is the normal image and one on the right is the same image after median blur. The effect may not be so obvious but if we look closely along the chair we can see that the left image has some noise along the edges which is smoothened after Blurring. 17

3.5 Robot model with all frames; Rviz visualization. 18

4.1 The scene on which ORB and Median+ ORB methods were tested for speed of matching and number of good correspondence. 20

4.2 The scene on which ORB and Median+ ORB methods were tested for their localization accuracies. 21

Chapter 1

Introduction

Simultaneous Localization and Mapping (SLAM) is a challenging problem and has been studied extensively in the literature of Mobile robotics / Mobile Autonomous systems. The problem is similar to a chicken and egg problem which deals with building a map of an unknown environment and determining the robot pose in that environment simultaneously. The author in [1] presents a two part paper with an extensive study on the SLAM, approaches taken to solve the problem and issue that still needs to be addressed. SLAM is essentially a state estimation problem where the state variables are pose of the robot and location of landmarks in the environment. Introduction of probabilistic models in robotics gave a way to solve these estimation problems by use of recursive Bayesian model. There are a lot of methods proposed in the literature for the state estimation Kalman Filter family, Particle filters and graph based SLAM are some of the most studied. Though the SLAM problem has been solved to a great extend issues like building a rich dense 3D map, Computational Complexity, speed, good data associations are still some areas to be worked upon. Further the problems of dealing with highly dynamic and fast environment are still an area to work upon. Mapping the observed environment is done using various sensors but all have some limitations. The 3 most used sensors for visualization of the environment are Laser Range Scanner, Camera's monocular or 3D sensors and Sonar sensors. The level of detail of mapping needed for a specific application along with other environmental factors like lighting, weather determines the use of a particular sensor. For example 3D sensors are cheap but the range of operations is limited and cannot work in dark environments whereas the range scanners are expensive but more reliable with longer range. Another aspect of the SLAM which has come into picture is the hardware complexity required for creating a rich 3D maps, researchers have used GPU pipelines to achieve good reconstruction of the environment but such a system can be expensive. In recent years Digital and 3D cameras are being investigated to solve the SLAM problem due cost effectiveness but still the high resolution images obtained pose a challenge when it comes to realtime implementation due the large processing time required.

The project proposes an improved Oriented Rotated Brief (ORB) feature based SLAM algorithm with Rao Blackwellised Particle Filter for state estimation to be implemented on an Adept Mobile Robot (Amigobot) equipped with a RGB and Depth sensor: Xtion Pro Live ;developed by Asus. The robot is equipped with 39000 ticks/wheel resolution encoders on both the wheels which give a precise odometry estimate. The algorithm proposes use of Median filter on the RGB images before extraction of ORB features which results in a faster feature matching process with less false positive. The main focus of the project was to come up with an algorithm which is able to localize the robot realtime while creating a Sparse Feature map with just enough features/Landmarks so as to achieve precise localization. The problem of Data Association is addressed via use of a Locality sensitive Hashing algorithm for initial correspondence and Random Sample Consensus for final correspondence of features/ Landmarks, using a locality sensitive hashing make tree like database of management of features observed which makes the search for nearest neighbors much efficient. In rest of the report the words Landmark and features are used interchangeably depending upon the context being discussed i.e. SLAM or Image Processing and both essentially mean the same.

Chapter 2

Literature Survey

2.1 Simultaneous Localization and Mapping

Ability of a mobile robot to move around in an unknown environment autonomously is the SLAM problem. There are lot of aspects which needs to be considered while addressing SLAM on any platform, to name a few : Hardware available, Sensor types, sensor accuracy, type of environment(Static or dynamic), on a mobile robot or a air drone, what kind of map you need for the application(Grid map or feature map), real time or offline. Lot of research is going in all these areas since past couple of decades. A major breakthrough came in late 90's with introduction of probabilistic approach towards the SLAM problem. SLAM is basically two steps in recursion namely measurement and update. The robot equipped with a particular sensor observes its environment and based on its observations has a belief of its position in the environment. As it moves forward if it re-observes a landmark in the environment its updates its belief about its position and landmarks position. Since there is an uncertainty associated with the observations made by the sensor as well as the robot motion so to capture these uncertainties probability came into picture and since the process is recursive the research came up with a Bayesian recursive model to address the SLAM problem. Equation 2.2 is the mathematical equivalent of the SLAM.

In accordance with popular SLAM literature [2] [1] , s represents the robot pose at time t, m represents the map so the SLAM problem so the state estimation can be written as equation below.

$$x_t = (s_t, m_t) \tag{2.1}$$

Now let u denote the control signals from time t-1 to t and z represent the current observation then the SLAM probabilistic model can be represented as:

$$p(s_t, m_t | z_t, u_t) = p(x_t | z_t, u_t) \tag{2.2}$$

now if we apply Bayes rule on equation 2.2 we get [2]

$$p(x_t|z_t, u_t) = \eta p(z_t|x_t) \int p(x_t|u_t, x_t{-}1) p(x_t{-}1|z_t{-}1, u_t{-}1) dx_t{-}1) = \eta p(z_t|x_t) \int p(x_t|u_t, x_t{-}1) Bel(x_t{-}1) dx_t{-}1 \tag{2.3}$$

Equation 2.3 is the equation we are trying to estimate over time. In literature there have been various methods proposed to solve these equations most of them can be put into 3 basic paradigms

1. Kalman filter [3] [4]
2. Particle filter [5][6][7]
3. Graph based SLAM [8]

Apart from the approach taken to solve the state estimation problem prominent research is going in field of mapping techniques author in [9] provides a comparison between 2D and 3D mapping of indoor environment in context of SLAM. Laser scanners [10], monocular camera [8], RGBD camera [11][12][13] have been successfully used to solve the SLAM, Localization and mapping problems. SLAM with unknown data association has to be considered for a accurate solution to the SLAM problem.[5][6][14] have presented the SLAM with unknown data association. Figure 2.1 below shows the data association problem. Some SLAM utilize visual odometry [3] [12] while others rely on robot odometry [15].

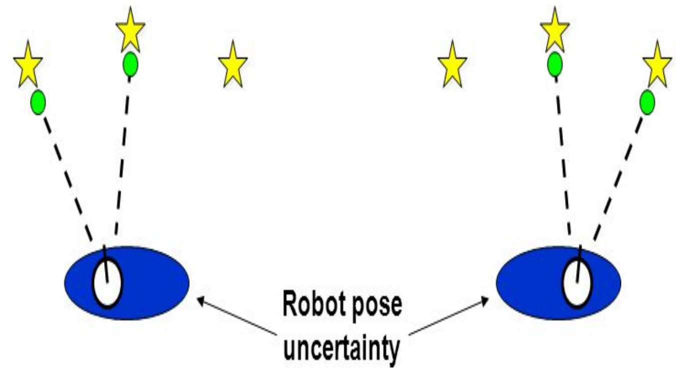

Figure 2.1: The figure shows the data association problem due to uncertainty in the pose of the robot, i.e. if the robot is the as shown in left its assumed that the measurements are from two left most landmarks and when the robot pose is assumed as shown in the right the measurements are assumed from the rightmost landmarks, which leads to uncertainty of measurement.

2.1.1 Kalman Filter SLAM

Kalman filters family was one of the first approaches taken to solve the state estimation problem in SLAM. The Extended Kalman filter (EKF) SLAM is popular among researchers, the extended Kalman filter makes the assumptions that the robot motion uncertainty as well as landmarks position uncertainty in the map can be approximated using a Gaussian model. In basic EKF-SLAM the vehicle motion is described in form [1]

$$P(x_t|x_t - 1, u_t) = f(x_t - 1, u_t) + w_t \tag{2.4}$$

Where f(.) models vehicle kinematics and w is additive , zero mean uncorrelated Gaussian motion noise with covariance Q. Equation 2.4 is called the motion model , motion model captures the uncertainty associated with the robot motion. Similarly the observation model which captures the uncertainty associated with the sensor is of form equation 2.5

$$P(z_t|x_t, m) = h(x_t, m) + v_t \tag{2.5}$$

Where h(.) denotes the geometry of the observation and v is the additive Gaussian noise with covariance R. Now with the motion model and observation model EKF can be used to compute the mean and covariance of the joint posterior robot pose and landmark locations in the map given by equation 2.2.

$$mean \triangleq \begin{bmatrix} \widehat{x_t|t} \\ \widehat{m_t} \end{bmatrix} = E \begin{bmatrix} x_k|Z_0 : t \\ m|Z_0 : t \end{bmatrix}$$

$$covarince \triangleq P_t|t = \begin{bmatrix} P_{xx} & P_{xm} \\ P_{xm}^T & P_{mm} \end{bmatrix} = E \begin{bmatrix} x_k - \widehat{x_k} \\ m - \widehat{m_k} \end{bmatrix} \begin{bmatrix} x_k - \widehat{x_k} \\ m - \widehat{m_k} \end{bmatrix}^T$$

where P_{xx} is the correlation between x, y and orientation of the robot pose. P_{xm} is the correlation between the robot pose and landmark location. $P_{mm}$is the correspondence between different landmarks. Now the time update is given by:

$$\widehat{x_{t|t-1}} = f(\widehat{x_{t-1|t-1}}, u_t) \tag{2.6}$$

$$P_{xx,t|t-1} = \nabla f P_{xx,t-1|t-1} \nabla f^T + Q_t \tag{2.7}$$

where ∇f is the Jacobian of f evaluated at the estimate $\widehat{x_{t-1|t-1}}$. Similarly the observation update is given by equations:

$$\begin{bmatrix} \widehat{x}_{t|t} \\ \widehat{m}_t \end{bmatrix} = \begin{bmatrix} \widehat{x}_{t|t} & \widehat{m}_{t-1} \end{bmatrix} + w_t \begin{bmatrix} z_k - h(\widehat{x}_{t|t-1}, \widehat{m}_{k-1}) \end{bmatrix} \tag{2.8}$$

$$P_{t|t} = P_{t|t-1} - W_t S_t W_t^T \tag{2.9}$$

where $S_t = \nabla h P_{t|t-1} \nabla h^T + R_t$ and $W_t = P_{k|k-1} \nabla h^T S_t^{-}1$ and where ∇h is the Jacobian of h evaluated at $\widehat{x}_{t|t-1}$ and $\widehat{m}_{t-1}$

Extended Kalman Filters and other filters from Kalman Filter Family like Sparse Extended Information filter(SEIF) have been successfully used to solve SLAM. But the Kalman filters have certain drawbacks:

1. Computational Effort: In EKF-SLAM at every observation update step update of all landmarks and joint covariance matrix is required, which is a quadratic operation with the number of landmarks. so as the number of landmarks observed increases so does the computational time. Though this drawback can be overcome by using a SEIF but the result is not as accurate as EKF.

2. Linear Assumption: The Kalman Filter family based filters employs a Gaussian liner model for nonlinear

motion and observation model. Such an assumption of linearity leads to inconsistency in highly nonlinear systems.

3. Data Association: EKF-SLAM is fragile to incorrect association of observation and Landmarks. Once a incorrect association is made at a particular time there is no way to correct the wrong association in the future which results in poor convergence to the real world environment.

2.1.2 Rao Blackwellized Particle Filter SLAM

Roa Blackwellized particle Filter popularly known as FastSLAM [6] marked the conceptual shift in design of recursive probabilistic SLAM. FastSLAM is based on Monte Carlo particle filter and is effective in representing the nonlinear behavior of the robot motion. In FastSLAM the robot pose is estimated by particles using sampling principle where each robot has its own belief about the robot pose and the landmarks are represented as 2x2 or 3x3 Kalman Filters. Each particle maintains its own map. FastSLAM exploits the conditional dependence between the robot pose and mapping whereby the location of different landmarks become independent of each other. It is better explained graphically in figure 2.2. Conditional independence implies that equation 2.2 which is the joint posterior can be factored into robot pose and individual landmark posterior[7][6]. Mathematically is given by:

$$p(s_t, m|z_t, u_t) = p(s_t|z_t, u_t) \prod p(m_i|s_t, z_t, u_t) \tag{2.10}$$

The mathematical formulation of Particle filter SLAM is as follows:

Path Estimate: The robot path is estimated using particles similar to Monte Carlo Localization. At every time instance a set of particles s_t represent the robot path posterior $p(s_t|z_t, u_t)$. So particle belief at any time instance can be calculated incrementally for previous pose of particles and applying the control command u_t

$${s_t}^k \approx p(s_t|u_t, {s_{t-1}}^k) \tag{2.11}$$

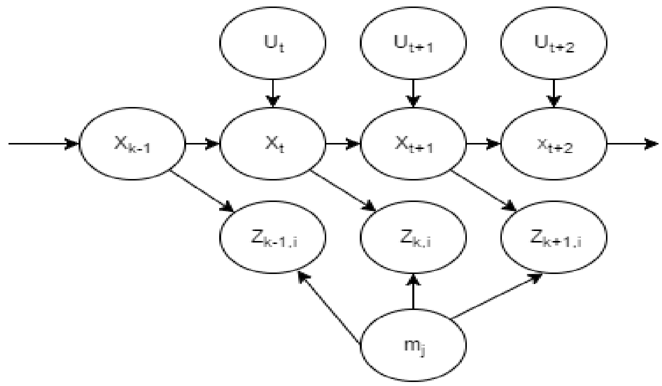

Figure 2.2: The figure is graphical model of the SLAM problem where X, U, Z, m have their usual meaning In FastSLAM the poses of the robot are known exactly so the observations become conditionally independent and since observations are independent so the map states are also independent. In particle Filter each particle has its belief of the robot pose and map of the environment.

where the superscript k represents the k-th particle set. This is applied per particle basis. the resulting distribution is called proposal distribution of particle filtering. Now after generating K particles a new particle set is obtained from the temporary set. Now each ${s_t}^k$ is drawn with replacement from the proposal distribution based upon so called importance factor ${w_t}^k$ given by:

$$ {w_t}^k = targetdistribution/proposaldistribution = p({s_t}^k|z_t, u_t)/p({s_t}^k|z_{t-1}, u_t) \tag{2.12} $$

Landmarks in FastSLAM: The landmarks estimate $p(m_i|s_t, z_t, u_t)$ is done using Kalman filters. For all landmarks each particle maintains full posterior over the particles full path and each landmark location and is represented by :

$$ S_t = ({s_t}^k, {\mu_1}^k, {\Sigma_1}^k, ..., {\mu_M}^k, {\Sigma_M}^k) \tag{2.13} $$

where μ and Σ are the mean and covariance of i-th landmark associated with each particle of the particle set S_t. The posterior over the location of i-th landmark depends on if the landmark is observed or not. If a particular landmark is observed the second part of equation 2.10 becomes:

$$ p(m_{i=n_t}|s_t, z_t, u_t, n_t) \propto p(z_t|m_i, s_t, n_t)p(m_i|s_{t-1}, z_{t-1}, u_{t-1}, n_{t-1}) \tag{2.14} $$

where the subscript n_t indicates landmark n at time t. If the landmark is not observed the Gaussian is left unchanged:

$$p(m_{i \neq n_t} | s_t, z_t, u_t, n_t) = p(m_i | s_{t-1}, z_{t-1}, u_{t-1}, n_{t-1}) \tag{2.15}$$

In the FastSLAM algorithm the update equation is realized using EKF where the filter uses a linearized version of observation model $p(z_t|s_t, m)$ the difference between FastSLAM and EKF-SLAM is the fact that the update step in FastSLAM is a constant time operation in the number of landmarks unlike EKF-SLAM where the complexity increases quadratically with the number of landmarks.

Another aspect of FastSLAM which makes it better than EKF SLAM is that the data association is done on per particle basis which ensures we end up with the right estimate of the map more often than not. The particles which make correct data association will receive a higher weight because they explain the measurement at a particular time instance better. Similarly particles with poor association will be given less weight and will be replaced by particles with higher weight during the resampling step. This is mathematically written as:

$${n_t}^k = argmaxp(z_t | {s_t}^k, m_{n_t}, n_t) \tag{2.16}$$

the per particle based data association has two advantages:

1. the Uncertainty in the robot motion doesn't affect the current data association given that we have sufficient number of particles estimating the pose of the robot.

2.A any given point in time in past some particles are bound to make a wrong association but the past association will not affect the future state as the particles will be replaced with particles with good association , which is unlike EKF-SLAM where the wrong data association incorporated at any time cannot be removed.

2.2 3D sensors in context of SLAM

Use of stereo vision in field of SLAM [4] gave a new cheaper solution towards sensing the environment in mobile robot trying to achieve autonomous exploration of environment. The Only problem with stereo vision is that a particular landmark had to be observed multiple times or from different viewpoints so as to estimate its depth and to be utilized in any SLAM algorithm. The introduction of micosoft kinect in 2009 gave a better solution towards problems like SLAM, Mapping, 3D reconstruction, pose estimation etc. 3D sensors like kinect and Xtion use structured light approach to obtain depth information unlike stereo vision which rely on disparity image. The 3D sensors project a light pattern on the 2D image and use triangulation to compute the depth with the known base distance between the RGB camera and the IR camera. The base distance between the sensors limits the range and accuracy of the sensors. The Xtion sensor used in this experiment has an operational range between 0.8-3.5 meters with a field of view of 58 degrees horizontal and 45 degrees vertical. The RGB and Depth images of 480x640 resolution are available at 30fps. The High frame rate and reasonable accuracy of 3D sensors made them popular among researchers for real time SLAM with mobile robots as well as aerial robots[3] [16][12] . [11] provides a comparative study between use of stereo vision and 3D cameras for mobile robot localization.

2.3 Visual SLAM

The use of camera's and 3D sensors in SLAM simplified economy constraint associated with the LIDARS but also opened a new area of research in areas like images features to be chosen as natural landmarks, the computational complexity associated with Dense 3D point cloud RGB images,mapping, feature matching techniques etc. Literature has suggested the use of SIFT and SURF features to be used as natural landmarks for the SLAM system due to their invariance to image translation , scaling and rotation.In this experiment we use ORB features due to their advantage when compared to SIFT and SURF in terms of matching speed.The feature matching is an important aspect of any vision inspired system. Literature suggested use of algorithms like RANSAC [17] and Joint Compatibility Branch and Bound (JCBB)[16] for outliers removal to improve the feature matching. The main idea of these feature based landmarks is that once unique points in image are identified these can be projected into world/map coordinates using the pin-hole model camera model.

Any visual sensors like camera can be described by its internal or intrinsic parameters which include focal length along x and y axis of image plane , lens distortion. Once a camera is calibrated the intrinsic parameters of the camera don't change unless the lens properties are changed like increasing the zoom etc. The external or extrinsic parameters describe the camera's position and orientation with respect to world frame. The pinhole camera model describes relationship between 3D point in world space and its corresponding 2D project into image plane as shown in figure 2.3.The Basic pinhole camera model equations are given below.[18]

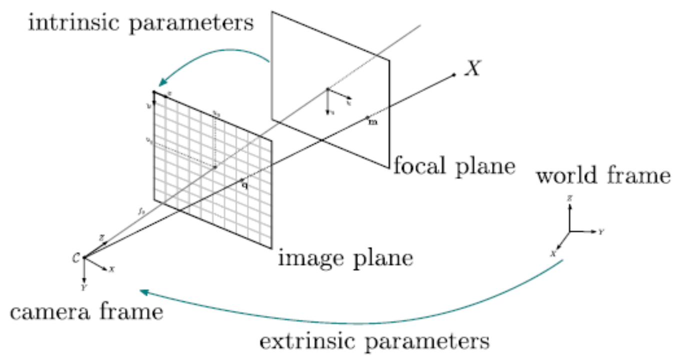

Figure 2.3: the figure shows the basic pinhole camera model, which describes the relationship between the world coordinate and the image plane.

Lets consider the optical axis being collinear to the z axis of the camera and optical center being located at origin of 3D coordinate system as shown in figure 2.3 then

$$u = Xf_x/Z, v = Yf_y/Z \tag{2.17}$$

where u and v are the coordinates of the 3D point in Image plane and f_x and f_y are focal length along x and y axis respectively.

The the above formulation gives the position of a 3D point into image coordinates as :

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & O_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.18}$$

where O_x O_y are the offset of the lens in x and y direction and λ is the homogeneous scaling factor.

The 3D sensors being equipped with as offset between the the RGB and IR have an additional static transform between the depth image and the RGB image which means the depth information given by the depth sensor has to be overlaid over the RGB image to get the correct depth information of a particular point in the RGB image. equation 2.19 shows the static rotation and translation between IR optical Frame and RGB optical frame and equation 2.20 shows the static translation and rotation between RGB optical frame and Base fare of the camera.

$$R = \begin{bmatrix} 9.9977e^{-}01 & 1.7292e^{-}03 & -2.1225e^{-}02 \\ -2.0032e^{-}03 & 9.9941e^{-}01 & -1.2893e^{-}02 \\ 2.1201e^{-}02 & 1.2933e^{-}02 & 9.9969e^{-}01 \end{bmatrix} T = \begin{bmatrix} 2.1354e^{-}02 & 2.5073e^{-}03 & -1.2922e^{-}01 \end{bmatrix} \tag{2.19}$$

$$T = \begin{bmatrix} -0.0220 & 0.2649 & -0.0550 \end{bmatrix} Quaternion = \begin{bmatrix} 0.4999 & -0.4999 & 0.4999 & 0.5000 \end{bmatrix} \tag{2.20}$$

Chapter 3

Proposed Method and Experiments

The proposed algorithm is built in python using OpenCV for image processing and ROS for controlling the robot, interacting with the Xtion camera. Figure.3.1 below shows the general layout of the SLAM algorithm proposed. We use Robot odometry provided by the amigobot for the motion model.Amigobot is equipped with position encoder with 39000/ticks wheel resolution which provides a good reliable estimate of the robot pose.The RGBD canmera provides with the landmark locations . Both the odomerty along with the landmark estimate are fed to the RBPF backend which refines the map and pose of the robot. The system map is updated every other time step. The rest of the chapter discusses about the RBPF backend in context of 3D landmark position. Image processing aspect which includes feature extraction, feature matching, improvement proposed in the ORB matching and the frame transformations, Robot model, ROS is discussed in later part of the chapter.

3.1 FastSLAM Backend

FastSLAM discussed in chapter 2 is extended to incorporate the visual features obtained from the 3D camera , the proposed algorithm makes certain changes to the structure to take into account the 3D landmarks and incorporate them into the our FastSLAM system.

$$p(s^t, \theta \,|\, u^t, z^t) \tag{3.1}$$

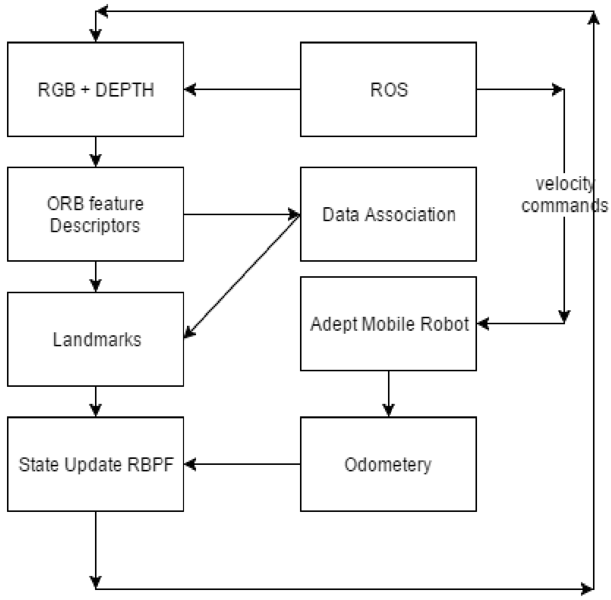

Figure 3.1: Outline of the proposed algorithm. The detailed explanation of each module is presented in further in the chapter

(where s is the state of the robot at time t,theta is the map, u is the control command at time t and z is the observation at time t.)into seprate path and landmark distribution.

$$p(s^t \left| u^t, z^t) \prod_{1}^{N} p(\theta | s^t, u^t, z^t) \right. \tag{3.2}$$

In our experiment the robot state s is represented as 3-element vector

$$s = (x, y, \phi) \tag{3.3}$$

where x and y are robot 2d position and ϕ is the heading direction of the robot relative to the origin of the map.

the landmarks in the map are represented as

$$\Theta = (\theta_1, ...\theta_N) \tag{3.4}$$

Eash landmark is represented by a triplet

$$\theta = (\mu, \sigma, d) \tag{3.5}$$

where μ is 3D landmark position, σ is the landmark error covariance and d is the landmark descriptor. An observation i.e. a landmark as observed from the current robot position is described as a triplet

$$z = (p, R, d) \tag{3.6}$$

where p is the position, R is the error covariance of the sensor, d is the landmark descriptor. The motion command is associated with the motion velocity, the direction of motion and time of motion. Taking into account all these modification of the structure of representation of 3d landmarks the exact update equation for the fast SLAM can be derived from similar to one shown in chapter 2.

3.2 Image Processing

Making sense of images to find suitable landmarks is the essence of the SLAM problem. Accuracy of the sensor in its measurements of the landmarks affects the confidence of the robot about certainty of its position. In context of Vision systems extracting unique and repeatable features which can be matched efficiently in overlapping images is essential. A lot of research has been done with regards to what kind of features can be used which gives unique features. SLAM literature has used SIFT, SURF, ORB and other similar feature detectors. In this experiment we use ORB features with slight modification to improve the feature matching speed. ORB features use FAST (Feature from Accelerated Segment Test) algorithm for feature point detection with modification so as to provide scale and viewpoint invariance. After obtaining feature points with associated direction descriptors are built using BRIEF (Binary Robust Independent Elementary Features) algorithm. BRIEF generate descriptors around feature points by binary coding method. The BRIEF descriptor is simpler and storage space is smaller than its counterparts SIFT and SURF.The Figure 3.2 below describes the flow of the image processing part of the SLAM problem and Figure 3.3 shows ORB features extracted from a indoor view.The rest of the section discusses the ORB modification used and the Data association using FLANN based matcher and RANSAC algorithm.

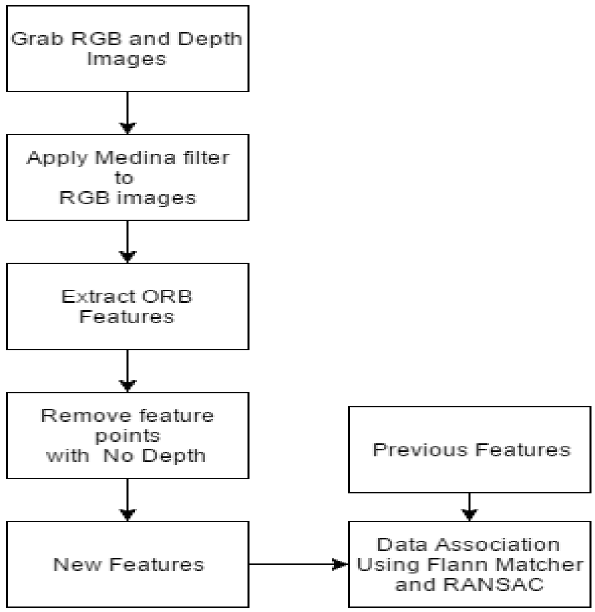

Figure 3.2: The figure depicts the general flow of the Image processing part of the SLAM

Median Filtering

In our algorithm we use a median filter on the images obtained in each frame before extracting the ORB features. The use of median filter removes the noise and gets higher precision matching points. The basic

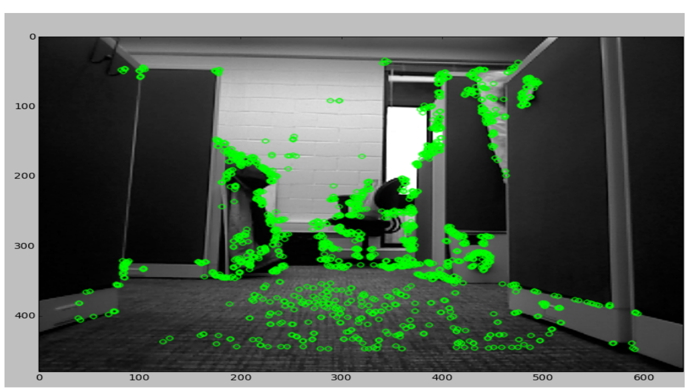

Figure 3.3: 1000 ORB features extracted using Open CV in an indoor office Environment

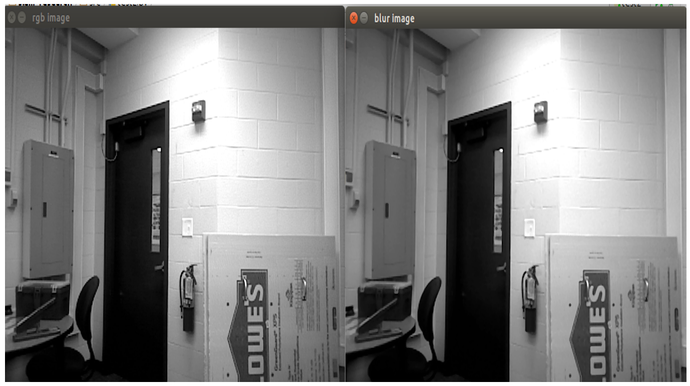

Figure 3.4: The image on the left is the normal image and one on the right is the same image after median blur. The effect may not be so obvious but if we look closely along the chair we can see that the left image has some noise along the edges which is smoothened after Blurring.

principle of Median filter is the value of point in the digital image is set to a mid-value, making the surrounding pixel values close to the real value and thus eliminating isolated noise. Figure 3.4 shows an image before and after Median filtering. Median filtering is particularly effective in removing salt and pepper noise and at the same time could keep the edge sharpness; it will remove texture in the same area, such as the trees in background. The use of median filters affects the performance of RANSAC algorithm and provides more correct matching points. The improvement in performance with use median filters in increasing the matching accuracy as well as the speed of matching is discussed in Results section.

Data Association

Two Factors contribute to uncertainty in the SLAM posterior : Measurement noise and motion noise. Each leads to different type of data ambiguity. In other words uncertainty of landmark position during measurement leads to confusion between nearby landmarks.Figure 2.1 shows one such condition where due to pose uncertainty there is confusion in terms of the measurement obtained corresponds to which landmark.

In our experiment the data association between landmarks is done in two steps. First an initial correspondence is made using nearest neighbor estimate between the current observations and previous Landmarks Descriptors. In case of a large map the observations has to be compared to all the landmarks which can be time consuming , therefore the Locality Sensitive Hashing (LSH) algorithm[19] was used for nearest neighbor search.The LSH algorithm shows good performance with high dimensional ORB descriptors hence suitable for several thousands of landmarks. Once the initial correspondence was established the correspondence was further refined using RANSAC algorithm [17]. RANSAC algorithm in an iterative manner tries to find the best association taking into account inlier's and outliers. Both RANSAC and LSH algorithms were implemented using Open CV.

3.3 ROS and Robot Model

ROS is an open source framework which provides libraries and tools for creating software applications for robot and provides hardware abstraction.ROS provides its SLAM package [20],ROS has been successfully been used with kinect for indoor mapping [21] and localization. In our experiment we use ROS to grab images frames and control the robot. ROS Aria package is used to control and give commands to the Amigobot. All the frame transformations for the entire SLAM system was taken care by creating a Universal Robot Descriptive Format(URDF)file for the robot mounted with the Xtion camera and then used ROS TF package to publish all transformations.URDF file is a XML based file which describes the position, properties in terms of collision properties , shape and other constrains in terms of motion of each sub frame of the robot system and the transformation between them. for example the wheel can move by 360 degrees but the camera is stationary Figure 3.6 below show robot model with all associated frames.

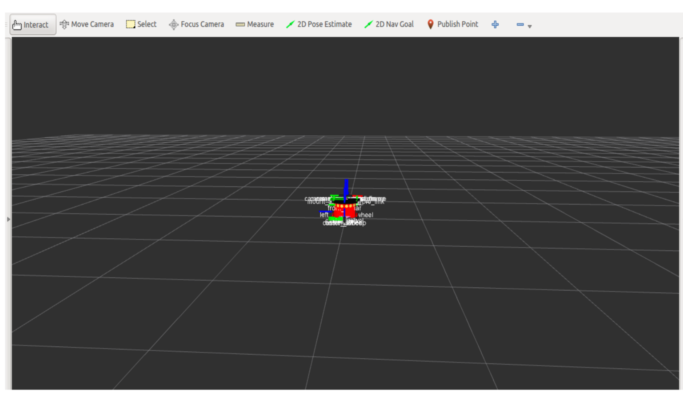

Figure 3.5: Robot model with all frames; Rviz visualization.

Chapter 4

Results and Discussions

The results of the experiment are discussed in two parts first we discuss how does use of median filter effect the performance of ORB descriptors in terms of its matching speed. Then we look at the SLAM results. The entire experiment was performed using Python, Open CV and ROS on a hp notebook with Intel core i3 processor. The table 4.1 shows the results of ORB and Median blur + ORB methods with different number of features extracted from two partially overlapping scene (Figure 4.1). It can been seen from the table that Median + ORB is not only faster than ORB but given better correspondences (more matches). In a particular case where we extracted 500 features from the scenes ORB couldn't find any matches but ORB + median found 12 correct matches. Better correspondence between scenes in SLAM context is very important because greater the landmarks re-observed better is the localization accuracy which in turn effects the mapping accuracy. As mentioned earlier we are more concerned about the localization accuracy of the system while creating just a sparse map of the environment.

Method	Img1 and Img2	Correspondence	Time(sec)
ORB	1000,1000	25	0.09978
Median+ORB	1000,1000	32	0.08804
ORB	5000,1000	26	0.11261
Median+ORB	5000,1000	34	0.09400
ORB	500,500	0	0.13461
Median+ORB	500,500	34	0.09400

Table 4.1: The Results for ORB and Median + ORB are listed along with the number of features extracted from each image, number of good correspondence made after RANSAC and time taken for making those correspondence

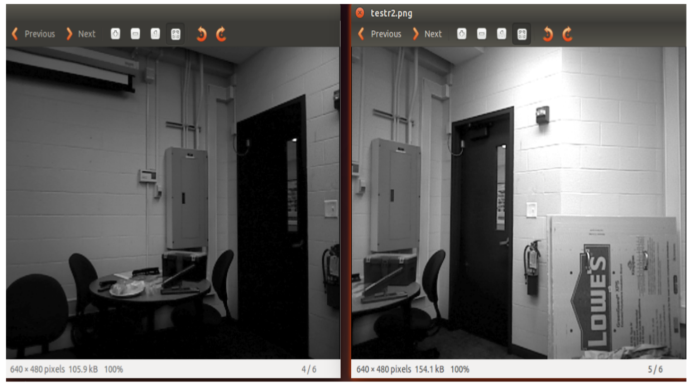

Figure 4.1: The scene on which ORB and Median+ ORB methods were tested for speed of matching and number of good correspondence.

In early stages of our experiment, results were computed with different number of particle set size and different number of Landmarks been extracted from each view. In the final run of the experiment we extracted 300 landmarks per view and used 20 particles for estimation of the pose of the robot. These final values were considered with keeping in mind the real-time performance and the localization accuracy desired. The localization results for a particular office setup is below discussed which shows ORB + median filters features outperform the ORB features when it comes to convergence rate of the predicted value to the real world environment. Table 4.2 below shows the localization accuracies in both for both methods when tested in the same environment. In the particle filter algorithms,there are different ways to initiate the particles when the systems starts, in our experiment we initiate all the particles with their x coordinate, y coordinate and orientation as zero (0,0,0). This reasonable assumption because whenever our system initiates we assume the robot is a (0, 0, 0) and start mapping from there. In literature when the exact size of the mapping area is known particles are be distributed uniformly in the area but since in our case the map grows dynamically we use the (0, 0, 0) initialization. When computing the localization accuracy of the system at each time stamp we average over the pose of all the particles and compare it against the current pose of the robot. The table 4.2 discusses the case where the robot is moving in a straight line path and trying to map the environment as shown in figure 4.2. The mapping results have not been discussed as part of this research.

the table 4.2 shows the accuracies of both the methods at each time stamp when tested in the same

Figure 4.2: The scene on which ORB and Median+ ORB methods were tested for their localization accuracies.

environment. It can be seen that the Median + ORB converges to real world with an error of 0.06m in 7 time stamps which corresponds to approximately 7 meters of robot movement. We can see that in time stamp 3 Median + ORB makes better correspondences as compared to ORB and converges to ground truth better.

Table 4.3 shows the timing performance of both the methods , i.e the time taken for the particle filter backend to run at each time stamp. We can observe from table 4.3 that the average time taken by the particle filter back end to run for the ORB + median filter is less than that or normal ORB which also emphasizes on the fact that updating observed landmarks is less time consuming that initializing new landmarks.Since ORB + median gives better good correspondances therefore less landmarks have to be initialized.

TimeStamp	ORB	Median+ORB	Corr ORB	Corr Median+ORB
1	0.61	0.71	0	0
2	0.67	0.70	200/32	212/35
3	0.54	0.505	193/179	213/199
4	0.47	0.30	235/22	200/199
5	0.38	0.17	240/225	204/202
6	0.25	0.10	250/28	239/12
7	0.14	0.06	273/40	246/32

Table 4.2: Localization accuracy of ORB and ORB + median are compared, when the robot is moving in straight line. The table shows the correspondence made in each step in terms of the number of matches found / number of good correspondence made.

TimeStamp	ORB pf timing(sec)	Median+ORB pf timing(sec)
1	1.12	1.25
2	0.35	0.31
3	0.49	0.56
4	0.24	0.04
5	0.65	0.05
6	0.10	0.20
7	0.71	0.54
	max = 0.71	max = 0.56
	avg = 0.52	avg = 0.42

Table 4.3: Timing chart for the time taken by ORB and ORB + median particle filters(pf) to run.

Chapter 5

Conclusion

In this project we introduced a Vision Inspired Simultaneous localization and mapping algorithm with improved ORB features for better correspondence and better matching speed. The project worked in realtime on a Amigobot equipped with Xtion Pro live sensor and utilizing Particle Filters as backend for the state Estimation.The project was successful as the localization aspect of SLAM problem was successfully addressed. Use of Median filter on the images before extraction of ORB features helped in improving correspondence between two consecutive view which improved the overall performance of the system in terms of timing as well as localization.The results showed a localization accuracy with an error less than 0.06 meters over a run of 10 meters in an indoor static office like environment with pronounced corners and edges.

The experimental results showed that the performance of the systems increases with increasing the number of particles in the system estimating the robot pose and maintaing their own maps , but the increase in the number of particles also effected the realtime performance of the system as the particle filter run time increased with the number of particles increased.

Chapter 6

Future Work

In this particular implementation of SLAM algorithm a lot of aspects of the problems were relaxed, first creating a rich visual map of the environment, secondly the environment in which the system was tested was static and well-structured environment. Aspects like loop closure, kidnapped robot problem, environments with not or similar features were not addressed in the scope of the work. As an extension of the work, GPU pipeline system can be employed for creating a rich visual maps of the environment with high detail. A SLAM system working in dynamic environment where the landmarks can change with time is still an active topic of research and coming up with algorithms to tackle this particular aspect has to be worked upon. Tackling featureless environment and loop closure will require a sophisticated algorithm for feature database management which can make decisions about which feature to track and the once to be remove over a long period of time. One observation with particle filters is that for unknown environment choosing a large number of particles effect the real-time performance of the system and choosing less particles effects the accuracy of the system as there are sufficient samples of the pose of the robot ,which enforces us to look at other alternatives like graph SLAM which have showed better results than particle Filter is some cases.

Bibliography

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE Robotics Automation Magazine*, vol. 13, pp. 99–110, June 2006.

[2] R. Sim, P. Elinas, M. Griffin, J. J. Little, *et al.*, "Vision-based slam using the rao-blackwellised particle filter," in *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, vol. 14, pp. 9–16, 2005.

[3] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *International Symposium on Robotics Research (ISRR)*, vol. 2, 2011.

[4] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *The international Journal of robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[5] M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 1, pp. 695–701 vol.1, 2002.

[6] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 2, pp. 1985–1991 vol.2, Sept 2003.

[7] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Aaai/iaai*, pp. 593–598, 2002.

[8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, pp. 1147–1163, Oct 2015.

[9] W. R. Abdulmajeed and R. Z. Mansoor, "Comparison between 2d and 3d mapping for indoor environments," in *International Journal of Engineering Research and Technology*, vol. 2, ESRSA Publications, 2013.

[10] H. Strasdat, C. Stachniss, M. Bennewitz, and W. Burgard, "Visual bearing-only simultaneous localization and mapping with improved feature matching," in *Autonome Mobile Systeme 2007*, pp. 15–21, Springer, 2007.

[11] L. Sabattini, A. Levratti, F. Venturi, E. Amplo, C. Fantuzzi, and C. Secchi, "Experimental comparison of 3d vision sensors for mobile robot localization for industrial application: Stereo-camera and rgb-d sensor," in *Control Automation Robotics Vision (ICARCV), 2012 12th International Conference on*, pp. 823–828, Dec 2012.

[12] H. Dong, N. Figueroa, and A. E. Saddik, "Towards consistent reconstructions of indoor spaces based on 6d rgb-d odometry and kinectfusion," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1796–1803, Sept 2014.

[13] O. Miksik, Y. Amar, V. Vineet, P. Prez, and P. H. S. Torr, "Incremental dense multi-modal 3d scene reconstruction," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 908–915, Sept 2015.

[14] A. Prozorov, A. Tyukin, I. Lebedev, and A. Priorov, "Self-localization of mobile robot in unknown environment," in *Open Innovations Association (FRUCT), 2015 17TH Conference of*, pp. 173–178, IEEE, 2015.

[15] F. Dellaert and A. W. Stroupe, "Linear 2d localization and mapping for single and multiple robot scenarios," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1, pp. 688–694, IEEE, 2002.

[16] J. Hartmann, D. Forouher, M. Litza, J. H. Kluessendorff, and E. Maehle, “Real-time visual slam using fastslam and the microsoft kinect camera,” in *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pp. 1–6, May 2012.

[17] R. Cupec, E. K. Nyarko, and R. Grbic, “Detection of dominant planar surfaces in disparity images based on random sampling,” *Tehnicki Vjesnik*, vol. 18, no. 3, pp. 301–308, 2011.

[18] P. Sturm, “Pinhole camera model,” in *Computer Vision*, pp. 610–613, Springer, 2014.

[19] Y. Fan, J. Xing, and W. Hu, “Load-balanced locality-sensitive hashing: A new method for efficient near duplicate image detection,” in *Image Processing (ICIP), 2015 IEEE International Conference on*, pp. 53–57, Sept 2015.

[20] B. L. E. A. Balasuriya, B. A. H. Chathuranga, B. H. M. D. Jayasundara, N. R. A. C. Napagoda, S. P. Kumarawadu, D. P. Chandima, and A. G. B. P. Jayasekara, “Outdoor robot navigation using gmapping based slam algorithm,” in *2016 Moratuwa Engineering Research Conference (MERCon)*, pp. 403–408, April 2016.

[21] H. I. M. A. Omara and K. S. M. Sahari, “Indoor mapping using kinect and ros,” in *Agents, Multi-Agent Systems and Robotics (ISAMSR), 2015 International Symposium on*, pp. 110–116, Aug 2015.