Vision Based Simultaneous Localization and Mapping

Shweta Gupta[1], Javed Ashraf[2]

[1]Research Scholar , [2]Asst. Professor
Deptt. of Electronics & Communication Engineering, Al-falah School of Engineering. & Technology, India.

shwegu88@gmail.com , jashraf.jmi@gmail.com

Abstract

This paper gives an introduction to the Simultaneous Localization and Mapping (SLAM) methods and recent advances in computational methods of SLAM problem for large scale & complex environment. SLAM addresses the problem of a robot navigating through an unknown environment. It is a process by which a mobile robot can build a map of its near-by environment and at the same time use this map to compute its own location. There are number of approaches to the SLAM problem. Different algorithms have been used to perform SLAM including Extended Kalman Filtering, Particle Filtering, Local Bundle Adjustment etc. There are still many practical issues to overcome, especially in more complex outdoor environments. This present paper elucidates variants of the SLAM problem and proposes a taxonomy for the same.

Keywords: *SLAM, Localization, mapping.*

1. Introduction

The autonomous, multi-purpose robot is an enduring dream of both science and the general public. These robots might take over the day to day drudgery of domestic labor, clean up hazardous waste or assist the disabled among many possible tasks. To achieve complete autonomy in environments previously unknown to the robot a robot must be able to create a map of its environment while maintaining its location within the map. This process in the robotics is known as simultaneous localization and mapping (SLAM). It is a difficult problem to solve because noise in the estimate of the robot's pose leads to noise in the estimate of the map and vice versa. A solution to the SLAM problem has been seen as a "holy grail" for the mobile robotics community as it would provide the means to make a robot truly autonomous.

A robot must use sensors to measure its environment and collecting the information as part of the SLAM process. Mapping is the problem of integrating the information gathered by and depicting that information as a given representation. It can be described by the first characteristic question, What does the world look like? In contrast to this, localization is the problem of estimating the location of the robot relative to a map; in other words, the robot has to answer the second characteristic question, *Where am I?* While performing SLAM, the robot observes the environment around it and detects the position of some features in the environment. These features serve as landmarks for the SLAM process. Estimating the positions of these landmarks constitutes the *mapping* part of SLAM process. As the robot moves, it again observes the landmarks. Currently observed landmarks are then matched with the previously known landmarks and discrepancy between the expected and currently measured positions of landmarks is used to adjust the estimate of robot position, this is the *localization* part of SLAM.

At present we have robust methods for mapping the environments that are static, structured, and of limited size. mapping unstructured, dynamic or large scale environment remains largely an open research problem. SLAM has also been implemented in a number of different domains from indoor robots to outdoor, underwater, and airborne systems.

In general, SLAM algorithms must address the following parameters

• Sensors, i.e., sonar, laser or vision, wide or narrow field of view
• Map representation, i.e., occupancy grid, 2D or 3D, natural or specialized landmarks
• Robot dynamics
• Environment dynamics, i.e., indoor or outdoor, static or dynamic
• Framework for combining over time the incoming sensor measurements and robot control signals.

Each of these choices has advantages and disadvantages as well as direct implications on the applicability of the algorithms based on them. The sensors may include wheel encoders, LIDAR sensors, acoustic range sensors or cameras. Cameras are an attractive option because of their low cost, low power use and passive nature. When cameras are used as the primary sensors the SLAM process is known as visual SLAM (VSLAM).

2. Background

Visual simultaneous localization and mapping has been a topic of interest for more than a decade. The genesis of the probabilistic SLAM problem occurred at the 1986 IEEE

Robotics and Automation Conference held in San Francisco, California. A number of researchers tried to apply estimation-theoretical methods to mapping and localization problems. Work by Smith and Cheesman and Durrant-Whyte [10] developed a statistical basis for describing relationships between landmarks and manipulating geometric uncertainty. Early work done by Azarbayejani and Pentland [2] included the concept of using an extended Kalman filter framework to estimate the camera's motion, scene structure and the camera's focal length simultaneously. But this technique does not operate in real time. Davison was the first to present a real time VSLAM system using a monocular camera in [7] which was able to operate in small, office scale environments in real time, completing loops of this limited scale. However, as the size of the environment grows, the complexity of the extended Kalman filter makes it too slow for real time applications. Eade and Drummond presented the first real time particle filter approach to monocular VSLAM in [6]. But that was able to operate in small, office scale environments in real time. Particle filter approach can theoretically complete loops in environments of any size. With a particle filter the hope is that at least one particle in the distribution contains a correct representation of the map with the loop completed. For this to be the case over large and complex environments, the number of particles must be too large for real-time processing. This limitation is probably the reason why the particle filter has not been in use in many recent VSLAM systems. Vision researchers have recently focused on extending VSLAM systems to larger, more complex environments than a desktop or a single room as well. For that purpose, a method is needed that can re-detect previously visited areas. Correlations between landmarks play an important role in SLAM, the more these correlations grew, the better the solution. At this time, work focused on improving computational efficiency and addressing issues in data association, or loop closure.

Interest in SLAM has grown exponentially in recent years, and workshops continue to be held at both ICRA and IROS.

3. SLAM MODEL

In our framework, robot starts at an unknown location and captures the images. The camera used can be either stereo vision camera or single camera. . For a single camera, the camera image synchronization and mechanical equipment for maintaining the relative positions of the cameras is not needed. Hence it makes the problem simpler. The information used are sequences of relative observations captured by the mobile robot. As the robot moves through the environment and captures the image, the SIFT algorithm will provide distinguishable and identifiable features. These extracted features will not be permanent features because whenever the robot switches its position to another location, the robot will lose track of the features that it has located. Hence, the features are to be extracted again. The features that are extracted from the SIFT[12] algorithm gives the key points, scale, orientation of the features. The term "robot pose" or "pose" is synonymously used to refer to the way the robot is positioned. Fig1 represents SLAM flow module.

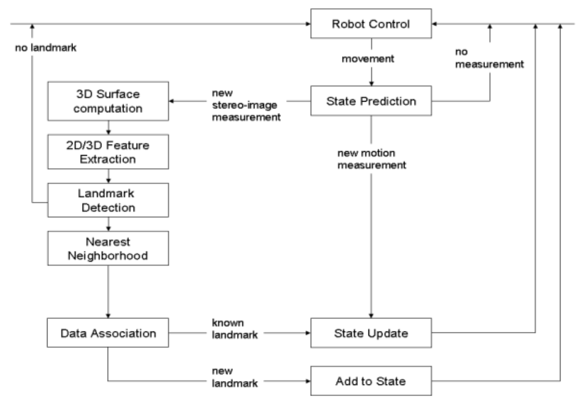

FIG. 1 SLAM FLOW DIAGRAM

ALGORITHM STEPS:

Most existing approaches to visual odometry are based on the following stages:

1. Acquire input images: using either single cameras, stereo cameras, or omnidirectional cameras.
2. Image correction: apply image processing techniques for lens distortion removal, etc
3. Feature detection: define interest operators, and match features across frames and construct optical flow field.
 a) Use correlation to establish correspondence of two images, and no long term feature tracking.
 b) Feature extraction and correlation
 c) Construct optical flow field.
4. Check flow field vectors for potential tracking errors and remove outliers.
5. Estimation of the camera motion from the optical flow.
 a) Choice 1: Kalman filter for state estimate distribution maintenance.
 b) Choice 2: find the geometric and 3D properties of the features that minimize a cost function based on the re-projection error between two adjacent images. This can be done by mathematical minimization or random sampling.

4. SLAM Solutions

SLAM solution involve finding an appropriate representation for the observation model and motion model. Two approaches are generally used – most common representation with additive Gaussian noise, generates the use of EKF technique, and another alternative representation describes vehicle motion as a set of asmples of non-Gaussian probability distribution called as particle filter approach.

A. EKF SLAM

This algorithm allows the robot to navigate in indoor environments using odometry and landmarks. The lines are identified using the Hough Transform. The prediction phase is done using geometric model of the robot. The updated phase uses the parameters of the line detected by Hough Transform in the Kalman equations without any modifications shown in Fig. 2

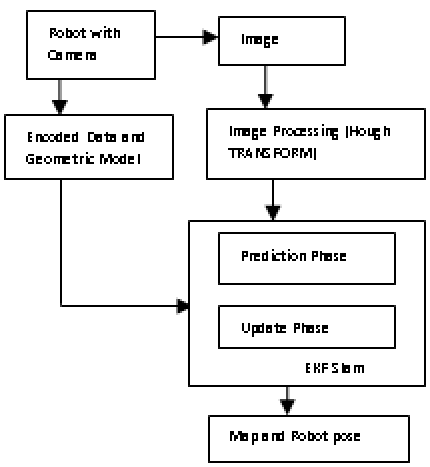

FIG. 2 EKF based proposed system

EKF[8] calculates the best estimate of the state vector in two phases: a) the prediction phase that predicts the current state based on the previous state and on the applied input signals; b) and the update phase to correct the predicted state by verifying its compatibility with the actual sensor measurements. In SLAM it's also required to include the landmark coordinates in the state vector. A Kalman filter is composed of following steps: Prediction and Update.

4.A1. Process Model

The state of the system consists initially of the position y and velocity v of the vehicle. Once a landmark is observed, the state is augmented with the position y_k of the new landmark.

Hence, the state of the system at instant t is defined by the following equation,

$y(t) = [y, v, y_1 \,..yn\,]$

4.A2. Update Phase

The estimate of the state vector and its corresponding covariance matrix are then updated as follows,

$$\hat{\hat{y}}(t+1)|t+1) = \hat{\hat{y}}(t+1|t) + W(t+1)v(t+1)$$

where W(t+1) is known as optimal Kalman gain at time t+1.

This Kalman Filter has a drawback that it cannot be implemented in large scale environment. Then another technique was designed.

B. PARTICLE FILTER

In terms of theoretical frameworks, the Extended Kalman Filter (EKF) has been the most common approach since its application by Smith. By maintaining a covariance matrix which encompasses all landmarks this method allows the EKF to develop pose and landmark estimates incrementally. However as the number of landmarks grows this matrix quickly becomes difficult to expand and update efficiently. It is necessary to update all elements for any new observation and this leads to a complexity of $O(N^2)$, where N is the number of landmarks. Another approach is the use of Particle Filters to approximate the posterior distribution over robot poses and maps. PFs can handle outliers better than the EKF but scale poorly with respect to the dimensionality of the state. Rao- Blackwellized particle filter[13] approach is generally to estimate a posterior of the path of the robot, in which each particle has associated with it an entire map. The distributions of landmarks are also represented by particle sets, where separate particles are used to represent the robot and the landmarks. This increases the computational load but the method is still applicable in real-time. The key advantage of this method is that the full posterior over robot poses and maps can be nonlinearly approximated at every point in time by particles. Our practical implementation also shows it can avoid rapid convergence of the particles to the maximum likelihood state.

When particle filters are applied in robot SLAM, the state variable *t x* contains two quantities that influence sensor measurements over time: the map (namely landmark position in our research) and robot's pose in the environment. Therefore, if *m* represents *K* landmark positions and *s* the robot's pose, equation (1) can be expressed as follows:

$$p(s_t\,, m \mid , z^t\,, u^t, n^t\,) = \eta p\,(z_t \mid s_t\,, m\,, n_t\,)$$

$$\int p(st|ut, st-1)p(st-1), m|zexp(t-1), uexp(t-1), nexp(t-1)dst-1$$

Particle Filter Path Estimation

First of all a plain particle filter is employed for estimating the path posterior *p*(*st* | *zt* ,*ut* , *nt*). The path posterior is denoted by S_t and each particle

$s_t^{[i]}$ € *S* represents an estimate of the robot's path:

Following [Montemerlo *et al.*, 2002] the particle set is calculated incrementally using the set St_{-1} , the control u_t and the measurement z_t

Particle Filter Landmark Location Estimation

A particle filter is next employed in estimating landmark location. Since this estimate is conditioned on the robot pose, the landmark particle filters are attached to individual pose particles in $t\ S$. The full posterior over paths and landmark positions can be represented by the sample set

$S_t = \{s^{t,[i]}, m1^{[i],[j]}, m2^{[i][j]},\ldots mk^{[i][j]}\}$

i= 1,2,…..M

j= 1,2, ….N.

$mk^{[i][j]}$ is landmark's pose

4. B1 Algorithm Process

Our algorithm for this approach proceeds as follows:

1. Initialize *r N* particles representing robot pose with normally distributed random numbers around the start position, each particle being a 3 by 1 state vector consisting of a position and orientation. Initialize *M* sets of *l N* particles representing *M* landmarks, each particle being a 2 by 1 state vector initially set to random positions within the environment.
2. Apply the motion model to each of the particles created in step 1.
3. For each particle representing robot pose,

a. Predict the observation for each particle in the particle sets representing landmark position; calculate the likelihood (weights) of particles from the measured value

b. Select (re-sample) the particles that best explain the observation according to their likelihoods.

4. For all particles representing robot pose, calculate the weights from measured values and estimated landmark position.
5. Re-sample the particles representing robot pose that best explain the observation according to their

4 likelihoods.

6. Go to step 2.

5. Conclusion

This paper has presented an introduction to SLAM process and different techniques to implement SLAM process. In this paper we have discussed various approaches like KALMAN Filter, Particle Filter that estimates camera's motion, scene structure and camera's focal length simultaneously. But both of these techniques have certain drawbacks that they can only map small, office scale environment in real time. But as the environment complexity increases, these techniques cannot be implemented. further modification can be done by combining both of these techniques. Generally Kalman filtering is used for localisation and Particle filter is used for mapping.

References

[1] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardos. Mapping large loops with a single hand-held camera. In Robotics: Science and Systems, June 2007.

[2] A. Azarbayejani and A. Pentland. Recursive estimation on motion , structure and focal length.

[3] Wenyan Hu, Tom Downs, GordonWyeth, Michael Milford and David Prasser, A Modified Particle Filter for Simultaneous Robot Localization and Landmark Tracking in an Indoor Environment

[4] Brian Clipp1, Christopher Zach1;Jongwoo Lim2, Jan-Michael Frahm1 and Marc Pollefeys Adaptive, Real-Time Visual Simultaneous Localization and Mapping

[5] Wenyan Hu, Tom Downs, GordonWyeth, Michael Milford and David Prasser. A Modified Particle Filter for Simultaneous Robot Localization and Landmark Tracking in an Indoor Environment

[6] E. Eade and T. Drummond. Scalable Monocular Slam

[7] Andrew J. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera

[8] Andre M. Santana, Adelardo A.D Mederios. SLAM based on Odometry and Extended Kalman Filter.

[9] Robert Sim, Pantelis Elinas, Matt Griffin, and James J. Little. Vision-based SLAM using the Rao-Blackwellised Particle Filter

[10] H.F. Durrant-Whyte. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. IEEE Trans. Robotics and Automation.

[11] Paul Smith, Ian Reid and Andrew Davison. Real-Time Monocular SLAMwith Straight Lines

[12] D. Lowe. Distinctive image features from scale-invariant keypoints.IJCV,60(2):91-110,Nov. 2004

[13] Robert Sim, Pantelis Elinas, Matt Griffin, and James J. Little. Vision-based SLAM using the Rao-Blackwellised Particle Filter.