A Generalized Odometry for Implementation of Simultaneous Localization and Mapping for Mobile Robots

Kethavath Raj Kumar

Intelligent Systems and Robotics Division, Centre for Artificial Intelligence and Robotics, India

Keywords: SLAM, Odometry, LiDAR, IMU, Particle Filter, Mobile Robot.

Abstract: This paper proposes a novel method for calculation of generalized Odometry using velocities from Light Detection and Ranging (LiDAR) and Inertial Measurement Unit (IMU), discounting velocities from motor encoder values. Further, the estimated velocities are used for the calculation of Odometry using rigid body Newtonian equations. The generalized Odometry and laser scans are used for implementation of the particle filter Simultaneous Localization and Mapping (SLAM) algorithm. This method overcomes errors due to slippages in mobile robots. The outputs of SLAM maps are experimentally validated in both straight and curved trajectories with reference to ground truth maps. SLAM results obtained from the proposed method Odometry is better than the only LiDAR, IMU and Encoder Odometry in an indoor environment for autonomous navigation of mobile robots.

1 INTRODUCTION

Mobile robots are used in various applications such as surveillance, military, mine detection, search and rescue operations etc. Simultaneous Localization and Mapping (SLAM) is used for navigating the mobile robot autonomously in indoor environments. Indoor SLAM systems are based on vision and LiDAR sensors. However, visual SLAM methods are sensitive to illumination conditions (Mur-Artal et al., 2015; Engel et al., 2014). LiDAR SLAM methods overcome drawback of former methods, which can be considered for mobile robots. These algorithms are based on scan-to-scan and scan-to-map matching techniques.

A scan-to-scan matching algorithms takes laser scan as input to compute the SLAM (Sokolov et al., 2017; Olson, 2015). However, these techniques quickly accumulate error over a time period. These accumulated error from scan-to-scan match techniques are corrected through scan-to-map algorithm (Olson, 2009). But this algorithm does not provide loop closure detection. Two other common approaches for solving accumulation errors are graph based and particle filters SLAM algorithms. Graph based SLAM algorithm takes input as laser scans and Odometry (Kohlbrecher et al., 2011). However, this method expects accurate Odometry like GPS-INS, which is not available for indoor mobile robot navigation. Particle filters are most popular among bayes filters (Konolige et al., 2010). It uses a laser scans and Odometry measurements for computation of SLAM. This method performs well even with approximate Odometry in indoor environments (Thrun et al., 2005). However, this method fails if no Odometry is available.

A novel method is proposed for calculation of Odometry using LiDAR and IMU velocities. This is a generalized Odometry because it is independent of the mobile robot kinematics. Further, the generalized Odometry and laser scans are used for the implementation of the particle filter SLAM algorithm on mobile robot in an indoor environment.

In this paper, Section-II describes the related work, Section-III describes the proposed method for calculation of Odometry, Section-IV describes the test platform, Section-V describes the SLAM implementation, Section-VI describes the results and discussion, and Section-VII describes the conclusion.

2 RELATED WORK

M. Sokolov et al. have presented ROS-based visual and LiDAR Odometry analysis for crawler-type robot in indoor small scale with minimal turnings (Sokolov et al., 2017). They found that the trajectory generated from LiDAR Odometry was closer to ground truth as compared to visual Odometry in spite of doing
3 PROPOSED METHOD

For calculation of generalized Odometry a novel method is proposed as shown in Figure 2.

3.1 Estimation of Velocities from LiDAR and IMU

3.1.1 The Linear Velocity Estimated from LiDAR

For estimation of linear velocity from LiDAR, the pose estimation method is adopted from Stefan Kohlbrecher et al. (Kohlbrecher et al., 2011). The pose vector $p_G$ is represented as

$$\begin{bmatrix} t \\ x_g \\ y_g \\ \theta_g \end{bmatrix}$$

The linear velocities are estimated from above pose estimation is represented as

$$v_x = \frac{dx_g}{dt}$$
$$v_y = \frac{dy_g}{dt}$$

where $v_x$ and $v_y$ are the longitudinal and lateral velocities. The magnitude of the velocity is given in equation 4, is assumed to be along the heading of the vehicle

$$V = \sqrt{v_x^2 + v_y^2}$$

3.1.2 The Angular Velocity from IMU

The IMU sensor is used for calculation of angular velocity from gyroscope sensor. IMU is a combination of accelerometer, gyroscope and magnetometer. It provides the raw accelerations, angular rates and fused orientations.

3.2 Odometry Calculation using Rigid Body Newtonian Equations

Given a velocities (linear and angular) of a rigid body the pose is calculated using Newtonian equations defined as

$$\Delta x = V \Delta t \cos \theta$$
$$\Delta y = V \Delta t \sin \theta$$
$$\Delta \theta = w \Delta t$$
where \( v \) and \( w \) represent the linear and angular velocities of the rigid body and \( \Delta t \) is the sampling interval. \( \Delta x, \Delta y \) and \( \Delta \theta \) are the distance travelled in the last sampling interval.

The pose vector \( p \) of the mobile robot defined as

\[
p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}
\]  

(8)

where \( x \) and \( y \) represent the position and \( \theta \) is the orientation of the mobile robot. Considering the incremental motion of the robot \( \Delta x, \Delta y \) and \( \Delta \theta \), the updated pose \( p' \) of the robot can be represented as

\[
p' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix}
\]  

(9)

4 TEST PLATFORM

For implementation of the particle filter SLAM algorithm using generalized Odometry, the tracked mobile robot is considered for experiments as shown in Figure 1. The tracked robot is integrated with IMU and 2D-LiDAR sensors. The robot consists of embedded pc (IPC2 intel core i7 with 16 GB RAM and 1.4 GHz dual core) with operating system Ubuntu 16.04 Linux distribution along with the ROS Kinetic framework.

Figure 1: The tracked mobile robot used for the experiments.

5 SLAM IMPLEMENTATION

5.1 Particle Filter SLAM

Particle filter based ROS slam gmapping algorithm developed by Giorgio Grisetti et al. (Grisetti et al., 2005; Grisetti et al., 2007) is used for implementation of SLAM algorithm.

The generalized Odometry and laser scan measurements are provided as input to the particle filter based SLAM implementation as shown in block diagram Figure 2.

5.2 Particle Filter Formulation

Given a set of observations from LiDAR and measurements from Odometry defined as

\[
Z_{1:t} = Z_1, Z_2, \ldots, Z_t
\]  

(10)

\[
U_{2:t} = U_1, U_2, \ldots, U_t
\]  

(11)

Calculating the posterior \( P \) with \( M \) grid map using equations and the trajectory of the robot defined as

\[
P \left( X_{1:t}, Z_{1:t}, U_{2:t} \right)
\]

(12)

\[
X_{1:t} = X_1, X_2, \ldots, X_t
\]  

(13)

The Rao-Blackwellized particle filter for SLAM makes use of the following factorization

\[
P \left( X_{1:t}, Z_{1:t}, U_{2:t} \right) = P \left( X_{1:t}, Z_{1:t} \right) P \left( Z_{1:t} | U_{2:t} \right)
\]

(14)

This factorization allows us to estimate the trajectory of the robot and then computes the map given a trajectory. Since the map strongly depends on the pose estimation of the robot, this approach offers an efficient computation.

6 RESULTS AND DISCUSSION

To conduct our SLAM experiments we considered the following maps as ground truth as shown in Figures 3,4,5.

6.1 SLAM using only LiDAR Odometry

In this experiment, the main objective was to verify only LiDAR based SLAM algorithm to create complete corridor of the indoor environment as shown in Figure 3.

Only LiDAR based SLAM algorithm fails to create the corridor, due to failure of loop closure in LiDAR Odometry as it created two corridors instead of one as shown in Figure 6.
6.2 SLAM using Encoder Odometry

In this experiment, the main objective was to verify the SLAM algorithm using Encoder Odometry from tracked robot with respect to the ground truth map as shown in Figure 3. Encoder Odometry in a tracked robot is not accurate because of the amount of slippage in turns which causes the failure of SLAM as shown in Figure 7.

From this experiment we conclude that for tracked robot, encoder Odometry failed to create the map.

6.3 SLAM using IMU Odometry

In this experiment, the main objective was to verify the SLAM algorithm using IMU Odometry from tracked robot with respect to the ground truth map as shown in Figure 5. For calculation of IMU Odometry, the accelerations are integrated twice for calculation of positions, which causes drift in the position over a time period. The created map using IMU Odometry fails to create corridor as shown in Figure 8.

6.4 SLAM using Generalized Odometry

To verify the particle filter SLAM using generalized Odometry multiple experiments are carried out in an indoor environment using tracked robot.

Experiment-1:

In the first experiment the robot is moved along a path composed of a turn and straight trajectory. The main objective is to verify the particle filter SLAM in a predefined path using generalized Odometry with reference to the ground truth map as shown in Figure 5.

The tracked robot is placed in a corridor initially the robot is turned and travelled in a straight trajectory as shown in rviz output Figure 9 to compute the SLAM in a corridor. The red color indicates the pose from generalized Odometry and green color is the...
Experiment-2:

In the second experiment, the main objective is to verify the particle filter SLAM algorithm in different curved and straight trajectories using generalized Odometry with reference to ground truth map as shown in Figure 4.

Initially, the tracked robot is placed in a foyer of 4 pillars connected with two corridors and two rooms. The robot travelled through different straight trajectories and curved trajectories like on the spot, turn of 180 degree and two turns of 90 degree rotations to compute the SLAM as shown in rviz output Figure 11.

After travelling through different straight and curved trajectories, the robot has created the map as shown in Figure 12.

Experiment-3:

In the third experiment, the main objective was to create a complete corridor map as shown in Figure 3 using generalized Odometry.

The tracked robot created an complete corridor map using generalized Odometry which was completely matching with ground truth map as shown in rviz Figure 13. The output of the trajectory is shown in red color. The output of the map is shown in Figure 14.

7 CONCLUSION

The generalized Odometry using LiDAR and IMU was considered instead of only LiDAR, Encoder and IMU Odometry for implementation of the particle fil-
The SLAM algorithm along with laser scans on tracked mobile robot in an indoor environment. The output SLAM maps using generalized Odometry in all the three cases was matching with respect to ground truth maps. These maps can be used for the autonomous navigation in indoor environments. This method overcomes errors due to slippages, because motor encoder velocities are discounted for calculation of Odometry. Since this method is independent of mobile robot kinematics, it can be experimented on other mobile robots also.

ACKNOWLEDGEMENTS

The author thanks the Director, CAIR for granting permission to publish the results of this research.

REFERENCES


