Comparison of widely used SLAM algorithms in ROS environment.

Prasanna Reddy Pulakurthi
Rochester Institute of Technology
1 Lomb Memorial Dr, Rochester, NY 14623
pp4405@rit.edu

Guoyu Lu
Rochester Institute of Technology
1 Lomb Memorial Dr, Rochester, NY 14623
luguoyu@cis.rit.edu

Abstract

The aim of this document is to provide a comprehensive review on the topic SLAM which is an acronym for Simultaneous Localization and Mapping. The following paper presents with different methods for implementing solution for SLAM. Initially the basic framework for SLAM is provided followed by different types of SLAM based on the types of sensors used. An in-depth literature review is done on the visual SLAM since it is the most popular method. Three of the most popular slam methods are discussed along with advantages and disadvantages of each. All these three methods are simulated in ROS environment and the results are presented. Finally a complete summery of the comparison of different methods of SLAM are presented and conclusion are drawn.

1. Introduction

The Simultaneous Localization and Mapping (SLAM) has been receiving a lot of attention in the recent robotics literature. The concept of SLAM works by building a map of an unknown environment by a robot while simultaneously navigating the environment using the map. This concept was first developed by J.J. Leonard and H.F. Durrant-Whyte [1]. SLAM is a fundamental technology with applications in many fields ranging from robotics, computer vision to augmented reality.

The process of SLAM is achieved in several steps. Firstly since the odometry (provides the robot's position) of the robot can be prone to error, we need special sensors such as lasers scanners. With the help of these sensors the position of the robot can be corrected. This is done by first extracting the features from the environment and re-observing when the robot moves. With the help of an Extended Kalman Filter a more accurate position of the robot can be updated. These extracted features are commonly called landmarks.

The section that follow are structured as follows. Section 2 contains literature review of different methods such as vSLAM, MonoSLAM, PATM, DTAM, RGB-D based SLAM, ORB-SLAM, Lidar Based SLAM approaches, GraphSLAM, FastSLAM, Gmapping, Octomap and RTAB SLAM. Section 3 describes the Robot, Environment, Sensors and the setup. Section 4 contains the resulting maps from various SLAM algorithms. Section 5 provides a complete summery of different types of SLAM methods and comparison of various SLAM techniques. Table 1 shows the different methods of SLAM along with the reference number.

Method	Reference
MonoSLAM	[9,10]
PATM	[11,12,13]
DTAM	[14,15,16]
LSD-SLAM	[17,18,17,20,21]
RGB-D	[22,23,24,25,26]
ORB-SLAM	[27,28]
VO	[30 to 37]
SFM	[38,39]
Graph SLAM	[47]
Fast SLAM	[48]
Hector Mapping	[49]
RTAB Mapping	[50,51]
Octomap	[52]

Table 1. Different methods of SLAM.

2. Literature review

SLAM systems are required to provide the orientation and the position of the robot by creating a map of the unknown environment while navigating the environment using the map. The SLAM systems make use of a set optical sensors.

The basic framework of SLAM comprises of initialization, tracking and mapping, re-localization and global map optimization. Initially, a certain co-ordinate system is defined for camera pose estimation. Once an initial map is created, mapping and tracking are performed continuously to find the camera pose. The technique of feature matching is used to build the 2D-3D correspondence between the image and the map. Then by making use of this point correspondences the camera pose can be estimated as presented in [2, 3].

The map is generated and represented by a 3D structure of the environment. In case tracking fails due to fast camera motion re-localization is performed. The map generally accumulates errors with each step in the process. We can refine the map by performing a global optimization. One solution for refining the map can be using loop closure techniques. Pose-graph optimization is a popular method to reduce the accumulated error [4, 5]. In visual SLAM the error is mainly from the re-projection error. Re-projection error is the difference between the perceived location and the actual location of each set of points. Bundle adjustment can be used to minimize the re-projection error [6].

The popular options for the sensors are visual SLAM (Camera Sensors) and Lidar-based SLAM (Light Detection and Ranging). The following section expand on the literature in the topics of different types of SLAM.

2.1. vSLAM

In recent years, SLAM making use of only camera data has been actively discussed in the literature. SLAM making use of a camera is referred to as visual SLAM (vSLAM). In this method a set of landmarks are tracked through successive frames and triangulate the 3D position. vSLAM has made into several popular fields including computer vision, Augmented reality [7] and unmanned autonomous vehicles (UAV) in the field of robotics [8]. It is important to note that most of the vSLAM algorithms operated on the assumption that the camera intrinsic parameters are calibrated. Therefore, using translation and rotation of the camera the camera pose can be determined in the global co-ordinate system. The main aspects of vSLAM can be categorized as Feature-Based, Direct and RGB-D based SLAM approaches.

2.1.1 MonoSLAM and PATM

Feature-Based method mainly has two broad categories namely filter-based and Bundle Adjustment based methods. Firstly the popular filter-based vSLAM algorithm was developed and named MonoSLAM by Davison et al. [9, 10]. In this method the camera motion and the 3D structure are simultaneously estimated using Extended Kalman Filter(EKF). MonoSLAM first makes use of known objects in the scene to create and initial map, then with the help of a extended kalman filter the camera motion and 3D positions of the feature points are estimated. The drawback so this method is the high computational cost with increasing environment size.

This problem of high computation cost is solved by splitting the tracking and mapping computational cost between different threads. This Parallel Tracking and Mapping (PTAM) method was introduce in [11]. By making use of this method BA can be incorporated into real-time vSLAM. In PTAM, five point algorithm [12] is used to get the initial map. Then feature matching is used to estimate the camera pose. Then by making use of triangulation the 3D positions of feature points can be estimated and optimized by BA. Finally using randomized tree-based searching the tracking process can be recovered [13]. One of the significant contributions to PTAM is to use keyframes. In the newer versions PTAM also make use of re-localization algorithms are seen in [13].

2.1.2 DTAM and LSD SLAM

Direct Methods make use the input image without using any features. A full direct method called DTAM was proposed by Newcombe et al. [14]. DTAM works by performing tacking by comparing input image with a synthetic image which is generated by a reconstructed map which is initialized using a stereo measurement. The synthetic view generated image is used to estimate the camera motion. Then using multi-baseline stereo the Depth information of each pixel is found and optimized. There are several methods of DTAM [14, 15, 16] which are optimised for fast real-time modeling.

Another popular method in direct methods is the LSD-SLAM. LSD-SLAM derives the idea from semi-dense VO [17]. In LSD-SLAM, reconstruction targets are limited to certain areas, as compared to DTAM that reconstructs the complete areas. The texture less areas are ignored as it is difficult to accurately estimate the depth. In the mapping step, a set of random values are assigned as initial depth values, and them, are optimized based on photo-metric consistency.

LSD-SLAM works as follows. Initially random depth values are given to each pixel. The camera motion is estimated from the synthetic view generation and the map. Only high-intensity gradient areas are considered for reconstruction. Finally a geometrically consistent map is generated by using pose-graph optimization.

With the help of a CPU these semi-dense approaches [17, 18] can be processed in real-time. They also can be optimized to mobile phones [19]. LSD-SLAM can also be extended to omni-directional and stereo camera setup [20, 21].

2.1.3 RGB-D based SLAM approaches

Visual SLAM can be performed using monocular camera, which is very cheap. However because this lacks the depth information the scale of the map and the estimated trajectory is unknown. It is hard to produce an initial map using from the very first frame. By making use of a Stereo camera or and RGB-D camera these problems of a monocular camera can be overcome. This allows for a more reliable visual SLAM solutions.

RGB-D approaches, In the recent times RGB-D cameras [22] such as Microsoft Kinect [23] have become popular because of low cost and portable size. RGB-D camera sensors provide Depth information along with the Red, Green and Blue channels. The main use of RGB-D is that a 3D structure of the environment along with the texture information can be obtained. A point to be highlighted is that most of the depth cameras are mainly developed for indoor use as they have a limited range of depth measurement at one to four meters.

Popular methods as presented in [24, 25] find the camera motion by tracking feature points between two successive frames. For the tracking, the complete RGB image is used. Using these tacked features a translation matrix is computed and refined by using an iterative closest points (ICP) algorithm [26]. Once these are computed a 3D structure of the environment can be constructed by combining multiple depth maps.

2.1.4 ORB-SLAM

Raul Mur-Artal and Juan D. Tardos present ORB-SLAM2 [27] which is a complete SLAM system for monocular, stereo and RGB-D cameras. This method includes map reuse, loop closing and relocalization capabilities and works in real-time. This method is evaluated on 29 popular public sequence and achieves state of the art performance. This method is built their previous work on monocular ORB-SLAM method [28]. ORB-SLAM2 consists of three main parallel threads. Firstly, Tracking to localize the camera by feature matching on every frame with the local map. Then the re-projection error is minimized by applying motion-only BA. Second, the local map is optimized by local BA. Finally, loop closing is performed to correct the accumulated drift. This is followed by full BA optimization to compute optimal motion and structure soluiton.

The important contribution of ORB-SLAM2 are as follows. This is the first open source SLAM system for monocular, stereo and RGB-D cameras. Using Bundle Adjustment it achieves better performance than state-of-the-art methods which are based on Iterative closest point or depth error minimization methods. This method also produces better than state-of-the-art performance in the case of stereo SLAM. This is achieved by making use of close and far stereo points. Even with the mapping disabled a lightweight localization mode is present to effectively reuse the map.

2.1.5 Fast Loop Closure

Closing the loop is very popular in may SLAM techniques as it reduces the localization drift. The main of closing the loop is to look for repetitive scenes from previous data to reset localization. To solve loop closure problem several algorithms are proposed in the literature which make use of visual feature that achieves accurate results, but with a high computational cost. However these methods ignore the patterns in an image such as the shape and texture information which is generally unique in the scene. Han Wang proposed Fast Loop Closure technique [29] which use this information by compressing the image into a binary image. This method greatly reduces the computational cost by making use of binary image without compromising on recall rate.

Fast Loop Closure consists of three parts. Firstly, binary content construction which consists of precise loop closure detection and fast image retrieval. In the first step objects and salient regions are extracted and compressed into a binary image. Then most of the unmatched pairs are removed by fast image retrieval which performs binary image indexing. Finally precise loop closure step removes any false positives. The main contributions are as follows. This method is fast as it makes use of binary operation for loop closure detection. This method can be easily implanted in exiting SLAM system easily.

2.1.6 Visaul Odometry and Structure from Motion

Visual Odometry (VO) is a camera based odometry technique which is an active field of research in the literature [30, 31]. Visual odometry is very similar to vSLAM according to survey papers [32, 33]. The main relation between the two is that vSLAM is a combination of VO and global map optimization. Initial publications on the VO technique was seen in [34, 35] and then vSLAM was proposed in [36, 37] by adding the global optimization to VO. Another popular technique that is very similar to vSLAM is Structure from Motion (SfM). Structure from Motion works by estimating the camera motion and environments 3D motion [38]. Am improved version is presented in [39] to facilitate running this algorithm online. Looking closely vSLAM and real-time SfM are very similar. In conclusion vSLAM, VO and SfM are very similar techniques.

2.1.7 Problems with vSLAM

In real world-situations, vSLAM faces some problems. This section presents these problems as follows.

In a few application such as augmented reality the sensor devices experience purely rotational motion. This is a problem in monocular vSLAM are disparities cannot be observed during pure rotation motion. Several projection models are used to solve this probelm [40, 41]. A point to be noted is that this purely rotational motion is not an issue in RGB-D vSLAM. This is because of the pre-existing depth map is used for tracking and mapping. Where as monocular camera-based vSLAM cannot map during pure rotation motion.

To achieve accurate estimation in vSLAM map initialization plays a very important role. Baseline should be wide to obtain an accurate initial map. This might not be the case in practical scenarios. Mulloni et al. proposed an easy initialization method [40] that makes use of 2D/3D guides for map initialization. Several other methods are proposed in the literature to solve this problem. The idea is to use reference objects and estimate the initial camera pose by racking these objects. Also methods [43, 44] make use of 3D objects whose shape is used to refine the map.

Estimating intrinsic camera parameters is important as vSLAM algorithms work on the assumption that the camera intrinsic parameters are known. Therefore camera calibration has to be performed prior to using vSLAM. Also the intrinsic camera parameters are fixed during estimation process in vSLAM. In literature, there are several methods to find intrinsic camera parameters during vSLAM [45]. The intrinsic parameters are made to gradually converge during vSLAM estimation.

In some vSLAM application using monocular vSLAM we observe scale ambiguity. To overcome this problem several sensors such as accelerometer, magnetic sensors and gyros can be used [46].

Rolling shutter distortion in vSLAM. Generally vSLAM algorithms assume a global shutter. However for accurate camera pose estimation, shutter type has to be considered. To make the cameras cost effective many RGB-D cameras employ rolling shutter. The drawback of a rolling shutter is that each row of the image is taken by a different camera pose. Therefore, it is difficult to estimate the camera pose directly. In the literature, interpolation based approach are used to estimate the rolling shutter and hence accurately estimate the camera pose.

2.2. Lidar based SLAM

A Lidar-based SLAM system uses a laser sensor to map the environment similar to visaul SLAM, but it has a higher accuracy in one dimension. A Lidar measures the position and the distance to an objected by emitting a pules of light and measuring the reflected pulse. Because of the high speed of light, precise laser performance is necessary to accurately track the distance to an object. Lidar SLAM is both a fast and an accurate approaches. However the drawbacks of using Lidar is that it doesn't provide a dense mapping hence loosing the texture information.

2.3. Graph SLAM

Sebastian Thrun and Michael Montemerlo. introduced GraphSLAM [47]. This method is transforms the SLAM posterior into a graphical network. This graphical network represents a log-likelihood of the data which is then reduced to a lower dimensional problem using variable elimination techniques. GraphSLAM extracts a set of soft constraints from the data and represents it by a sparse graph. The map and the robot path is determined by solving these constraints which are generally nonlinear. These nonlinear constraints can be linearized using optimization techniques and the resulting least square problem can be solved. This method can be applied to large-scale mapping problems and can handle large number of features.

GraphSLAM has a number of limitations. It relies on a good initial estimate of the map. As the number of steps increase a degrade in performance can be observed in the accuracy of the odometry-based initial guess and lead to increase in data-association errors. It also assumes independent Gaussian noise, however in the real world noise need not be Gaussian not independent. Another limitation is in the matrix inversion as it can be slow and better optimization methods such as conjugate gradient must be used. It also assumes a static environment and a dynamic environment as seen in the real world would degrades its performance.

2.4. Fast SLAM

Several of popular SLAM techniques cannot handle very large number of landmarks as present in the real-world. For instance SLAM methods that make use of kalman filters see a quadratic increase in time with increase in landmark numbers. Michael Montemerlo and Sebastian Thrun proposed FastSLAM [48] to overcome this problem. Their method recursively estimates the full posterior distribution over robot pose and landmarks. This method scales logarithmically with the number of landmarks which is a huge improvement in terms of computational cost.

2.5. Gmapping

Gmapping build on Rao-Blackwellized particle filters for solving SLAM problem. It presents and adaptive Rao-Blackwellized technique to reduce the number of particles [49]. Gmapping is widely used to create a 2D map with the help of laser scanners.

2.6. 3D mapping - RTAB-mapping and Octomap

Many robotic Applications such as flying robots require a three dimensional models. The following two methods produce 3D map reconstructions of the unknown environment.

RTAB stands for Real Time Appearance Based mapping(RTAB-mapping) which is a widely used graph based SLAM approach that can build three dimensional dense map. An optimized version of RTAB-map is presented in [50] which makes use of multiple strategies to allow for real time loop closure for large scale and long term SLAM. A Semantic-RTAB-map technique was proposed in [51] making use of RTAB-map and deep learning techniques to implement a semantic SLAM system.

Octomap uses a three dimensions occupancy grid mapping approach. This mapping approach is based on the concept of octress and makes use of a probabilistic occupancy estimation. Each cube in an octree represented as a cubic volume, each of these cubes can be subdivided into 8 symmetric octress. A three dimensional representation of an object at different resolutions can be created using this concept. Octomap uses this concept along with probabilistic sensor fusion, multi-resolution queries and octree map compression to build its maps. Octomap shows efficient update on the representation, consistent data modelling and minimum memory requirement [52].

3. Experimental setup and project description

For this project we will draw comparison on three popular SLAM techniques namely Gmapping, Octomap and Real-time Appearance-based mapping.

The entire project was simulated using Robot operating system. Robot Operating System (ROS) is a frame work consisting of tool and libraries for writing robot software which aims to model complex robot behavior.

The robot was designed using Solid-works and simulated in Gazebo. Gazebo is a three dimensional simulator, while ROS serves as the interface for the robot. Two robots were used in the project, one was built from scratch and contained an Intel real-sense d435 which a RGB-D sensor as shown in 1.

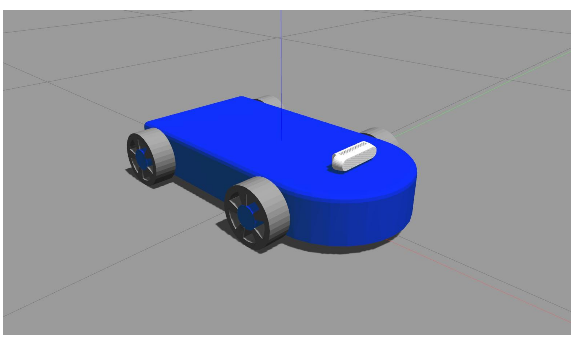

Figure 1. Robot built from starch.

The other robot used is the Rosbot which contains RPLIDAR A2 whcih is a lasaer scanner and rgbd sensor called Astra for visal and depth information. In addition it also contains an IMO, an antenna and DC motor with encodes. Rosbot in the Gazebo environment is shown in Figure 2.

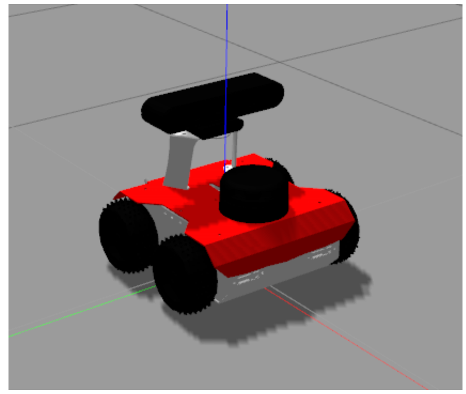

Figure 2. Rosbot Robot.

The main sensors used in the project are the RGB-D sensors for visual and depth information and laser scanner. The RGB visual output is shown in Figure 3 and the point could using the depth information is shown in Figure 4. The bots have various sensors as different sensors are used in different algorithms. RVIZ was used for visualizing the sensor data which is a three dimensional visualization tool for ROS.

Figure 3. RGB sensor data.

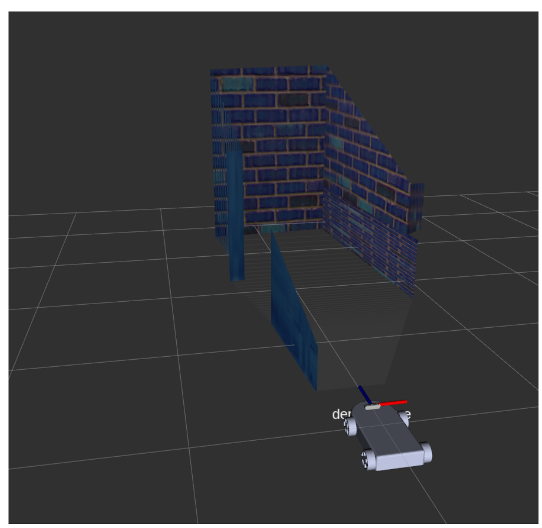

Figure 4. Point could sensor data.

A simulated indoor environment was build with the help of Gazebo. The environment is a closed room with a door opening a few windows and wooden partition for the robot to navigate and map. The simulated indoor environment is shown in Figure 5.

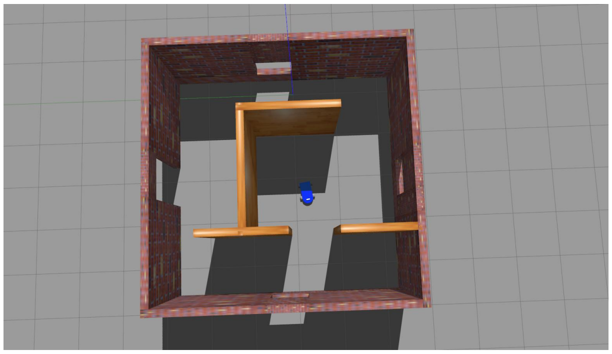

Figure 5. Environment.

4. Results

The robot was manually navigated in the simulated environment and three different algorithms were used to perform mapping.

Gmapping makes use of a laser scanner by taking the raw laser data along with odometry and is optimized for long range laser scan. Gmapping makes use of Roa-Blackwellized particle filter. It efficiently reduces the number of particles to reduce the computational speed and also selectively re-sampling operation to account for particle depletion. The final mapped output of the Gmapping techniques is shown in Figure 6.

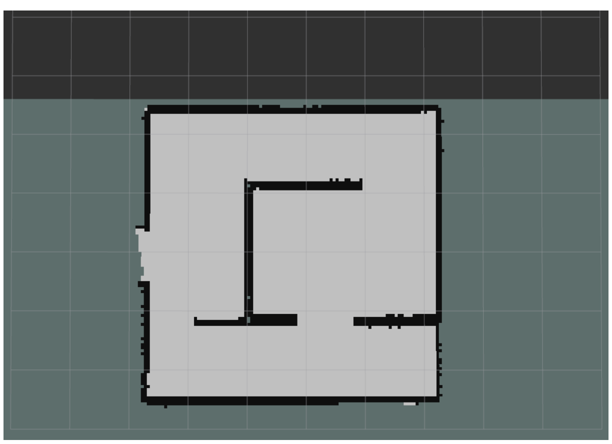

Figure 6. Map generated using Gmapping technique.

The following methods produce a three dimensional map which is useful for various robotics applications.

The results of an efficient three dimensional occupancy grid mapping called Octomap is shown in Figure 7.

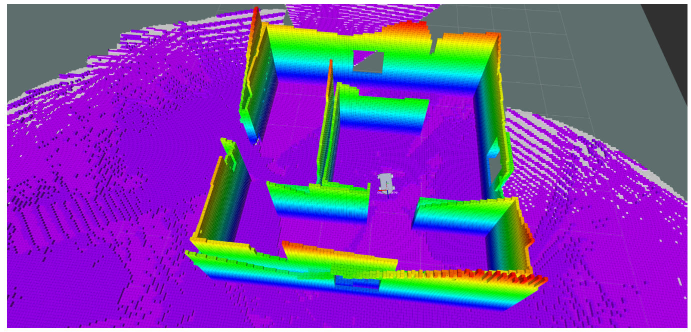

Figure 7. Map generated using Octomap technique.

Real-Time Appearance-Based Mapping (RTAB Maping) output is shown in Figure 8. We observe a three dimensional reconstruction of the map using visual and depth information from the RGB-D sensor. The path traced by the robot is seen in the blue trace. The method takes a long time for computation when compared to gmapping.

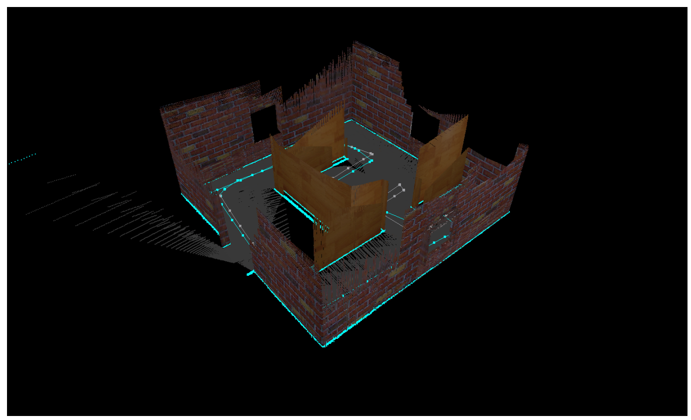

Figure 8. Map generated using RTAB technique.

5. Conclusions

Firstly the basic framework of a SLAM is introduced and consists of five main categories namely initialization, tracking and mapping, re-localization and global map optimization. Then the most popular SLAM method namely vSLAM is presented. There are three main categories in vSLAM, which are Feature-Based, Direct and RGB-D based SLAM approaches. Feature-Based method consisted of MonoSLAM with had a drawback of using high computation with increasing environment size. To overcome this PTAM was introduced.

Second, Direct method consisted of two main methods DTAM and LSD-SLAM. Though monocular cameras are very cheap they lack depth information. RGB-D based methods make use of this depth information to produce more accurate results. ORB-SLAM was discussed which provided state-of-the-art performance for monocular, stereo and RGB-D camera settings. Visual Odometry and Structure from Motion are very similar to vSLAM. The problems with slam were discussed, the important ones being, purely rotation motion, map initialization, Estimating intrinsic camera parameters, scale ambiguity and rolling shutter distortion and presented solution in the literature to address these problems. Lidar based SLAM was discussed along with other methods such as Graph SLAM and Fast SLAM.

After analysing the above results of all the three maps of the same indoor environment. The advantages and disadvantages of Gmapping, Octomap and RTAB-map are discussed as follows.

The advantages of Gmapping is that its computational inexpensive. The implementation of Gmapp is fairly straight forward and its easy to program in ROS environment. Gmapping the best 2D mapping algorithm algorithm for Mobile robots applications. The disadvantages of Gmapping compared to other two are as follows. Gmapping makes of laser sensor which is costly. It cannot detect glass as the laser passes though a transparent glass. On the other hand RGB-D equipment are inexpensive when compared to laser and a three dimensional maps can be created because of the dense texture information available in RGB-D sensors.

After comparing the above two 3D maps, we could find unique advantages and disadvantage between Octomap and RTAB-map. The advantages of both the methods compared to Gmapping is that they enhance the performance of a multi robot system. Octomap ideal for detection obstacles to help collision navigation and to share navigation information. Though octree data structure allows for a lightweight mathematical solution, it cannot create a descriptive map of its environment which is need for an intelligent robot navigation in the real world. RTAB is ideal for a scenario that requires to build descriptive map with detailed texture information. The disadvantage of both these methods are that they are computationally expensive and not so easy to code and implement as Gmapping.

References

[1] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," in IEEE Transactions on Robotics and Automation, vol. 7, no. 3, pp. 376-382, June 1991.

[2] Klette R, Koschan A, Schluns K (1998) Computer vision: three-dimensional data from images. 1st edn

[3] Nister D (2004) A minimal solution to the generalised 3-point pose problem. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Vol. 1. pp 560–5671

[4] Grisetti G, Kümmerle R, Stachniss C, Burgard W (2010) A tutorial on graph-based slam. Intell Transp Syst Mag IEEE 2(4):31–43

[5] Kümmerle R, Grisetti G, Strasdat H, Konolige K, Burgard W (2011) g2o: A general framework for graph optimization. In: Proceedings of International Conference on Robotics and Automation. pp 3607–3613

[6] Bundle adjustment a modern synthesis. In: Triggs B, McLauchlan PF, Hartley RI, Fitzgibbon AW (eds) (2000) Vision algorithms: theory and practice. pp 298–372

[7] Billinghurst M, Clark A, Lee G (2015) A survey of augmented reality. Found Trends Human-Computer Interact 8(2-3):73–272

[8] Engel J, Sturm J, Cremers D (2012) Camera-based navigation of a low-cost quadrocopter. In: Proceedings of International Conference on Intelligent Robots and Systems. pp 2815–2821

[9] Davison AJ (2003) Real-time simultaneous localisation and mapping with a single camera. In: Proceedings of International Conference on Computer Vision. pp 1403–1410

[10] Davison AJ, Reid ID, Molton ND, Stasse O (2007) Monoslam: real-time single camera SLAM. Pattern Anal Mach Intell IEEE Trans 29(6):1052–1067

[11] Klein G, Murray DW (2007) Parallel tracking and mapping for small AR workspaces. In: Proceedngs of International Symposium on Mixed and Augmented Reality. pp 225–234

[12] Nistér D (2004) An efficient solution to the five-point relative pose problem. Pattern Anal Mach Intell IEEE Trans 26(6):756–770

[13] Williams B, Klein G, Reid I (2007) Real-time SLAM relocalisation. In: Proceedings of International Conference on Computer Vision. pp 1–8

[14] Newcombe RA, Lovegrove SJ, Davison AJ (2011) DTAM: dense tracking and mapping in real-time. In: Proceedings of International Conference on Computer Vision. pp 2320–2327

[15] Ondruska P, Kohli P, Izadi S (2015) MobileFusion: real-time volumetric surface reconstruction and dense tracking on mobile phones. IEEE Trans Vis Comput Graph 21(11):1251–1258

[16] Stühmer J, Gumhold S, Cremers D (2010) Real-time dense geometry from a handheld camera. In: Goesele M, Roth S, Kuijper A, Schiele B, Schindler K (eds). Springer, Berlin, Heidelberg, pp 11–20

[17] Engel J, Sturm J, Cremers D (2013) Semi-dense visual odometry for a monocular camera. In: Proceedings of International Conference on Computer Vision. pp 1449–1456

[18] Engel J, Schöps T, Cremers D (2014) LSD-SLAM: large-scale direct monocular SLAM. In: Proceedings of European Conference on Computer Vision. pp 834–849

[19] Schöps T, Engel J, Cremers D (2014) Semi-dense visual odometry for AR on a smartphone. In: Proceedngs of International Symposium on Mixed and Augmented Reality. pp 145–150

[20] Caruso D, Engel J, Cremers D (2015) Large-scale direct SLAM for omnidirectional cameras. In: Proceedings of International Conference on Intelligent Robots and Systems

[21] Engel J, Stueckler J, Cremers D (2015) Large-scale direct SLAM with stereo cameras. In: Proceedings of International Conference on Intelligent Robots and Systems

[22] Geng J (2011) Structured-light 3d surface imaging: a tutorial. Adv Opt Photon 3(2):128–160

[23] Zhang Z (2012) Microsoft kinect sensor and its effect. MultiMedia IEEE 19(2):4–10

[24] Henry P, Krainin M, Herbst E, Ren X, Fox D (2012) RGB-D mapping: using Kinect-style depth cameras for dense 3D modeling of indoor environments. Int J Robot Res 31(5):647–663

[25] Endres F, Hess J, Engelhard N, Sturm J, Cremers D, Burgard W (2012) An evaluation of the RGB-D SLAM system. In: Proceedings of International Conference on Robotics and Automation. pp 1691–1696

[26] Besl PJ, McKay ND (1992) A method for registration of 3-d shapes. IEEE Trans Pattern Anal Mach Intell 14(2):239–256

[27] Mur-Artal, Raul, and Juan D. Tardos. "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras." IEEE Transactions on Robotics 33.5 (2017): 1255–1262. Crossref. Web.

[28] Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, 2015.

[29] Wang, Han Li, Juncheng Ran, Maopeng Xie, Lihua. (2020). Fast Loop Closure Detection via Binary Content.

[30] Nistér D, Naroditsky O, Bergen J (2004) Visual odometry. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Vol. 1. p 652

[31] Yousif K, Bab-Hadiashar A, Hoseinnezhad R (2015) An overview to visual odometry and visual slam: applications to mobile robotics. Intell Ind Syst 1(4):289–311

[32] Scaramuzza D, Fraundorfer F (2011) Visual odometry [tutorial]. Robot Autom Mag IEEE 18(4):80–92

[33] Fraundorfer F, Scaramuzza D (2012) Visual odometry: Part ii: matching, robustness, optimization, and applications. Robot Autom Mag IEEE 19(2):78–90

[34] Engel J, Sturm J, Cremers D (2013) Semi-dense visual odometry for a monocular camera. In: Proceedings of International Conference on Computer Vision. pp 1449–1456

[35] Kerl C, Sturm J, Cremers D (2013) Robust odometry estimation for RGB-D cameras. In: Proceedings of International Conference on Robotics and Automation. pp 3748–3754

[36] Engel J, Schöps T, Cremers D (2014) LSD-SLAM: large-scale direct monocular SLAM. In: Proceedings of European Conference on Computer Vision. pp 834–849

[37] Kerl C, Sturm J, Cremers D (2013) Dense visual SLAM for RGB-D cameras. In: Proceedings of International Conference on Intelligent Robots and Systems. pp 2100–2106

[38] Agarwal S, Furukawa Y, Snavely N, Simon I, Curless B, Seitz SM, Szeliski R (2011) Building rome in a day. Commun ACM 54(10):105–112

[39] Civera J, Grasa OG, Davison AJ, Montiel J (2010) 1-point ransac for extended kalman filtering: application to real-time structure from motion and visual odometry. J Field Robot 27(5):609–631

[40] Gauglitz S, Sweeney C, Ventura J, Turk M, Höllerer T (2012) Live tracking and mapping from both general and rotation-only camera motion. In: Proceedngs of International Symposium on Mixed and Augmented Reality. pp 13–22

[41] Pirchheim C, Schmalstieg D, Reitmayr G (2013) Handling pure camera rotation in keyframe-based SLAM. In: Proceedngs of International Symposium on Mixed and Augmented Reality. pp 229–238

[42] Mulloni A, Ramachandran M, Reitmayr G, Wagner D, Grasset R, Diaz S (2013) User friendly SLAM initialization. In: Proceedngs of International Symposium on Mixed and Augmented Reality. pp 153–162

[43] Bleser G, Wuest H, Stricker D (2006) Online camera pose estimation in partially known and dynamic scenes. In: Proceedings of International Symposium on Mixed and Augmented Reality. pp 56–65

[44] Tamaazousti M, Gay-Bellile V, Collette SN, Bourgeois S, Dhome M (2011) Nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp 3073–3080

[45] Civera J, Bueno DR, Davison A, Montiel JMM (2009) Camera self-calibration for sequential bayesian structure from motion. In: Proceedings of International Conference on Robotics and Automation. pp 403–408

[46] Mustaniemi J, Kannala J, Särkkä S, Matas J, Heikkilä J (2016) Inertial-based scale estimation for structure from motion on mobile devices. CoRR. abs/1611.09498

[47] Sebastian Thrun and Michael Montemerlo. 2006. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. Int. J. Rob. Res. 25, 5–6 (May 2006), 403–429. DOI:https://doi.org/10.1177/0278364906065387

[48] Montemerlo, Michael Thrun, Sebastian Koller, Daphne Wegbreit, Ben. (2002). FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. Proceedings of the National Conference on Artificial Intelligence.

[49] G. Grisetti, C. Stachniss and W. Burgard, ”Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters,” in IEEE Transactions on Robotics, vol. 23, no. 1, pp. 34-46, Feb. 2007.

[50] Das, Sagarnil. “Simultaneous Localization and Mapping (SLAM) using RTAB-MAP.” ArXiv abs/1809.02989 (2018): n. pag.

[51] Mingyuan Mao, Hewei Zhang, Simeng Li, Baochang Zhang. SEMANTIC-RTAB-MAP (SRM): A semantic SLAM system with CNNs on depth images. Mathematical Foundations of Computing, 2019, 2 (1): 29-41. doi: 10.3934/mfc.2019003

[52] Hornung, A., Wurm, K.M., Bennewitz, M. et al. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Auton Robot 34, 189–206 (2013). https://doi.org/10.1007/s10514-012-9321-0