Indoor SLAM using Kinect Sensor

H. Sankrit
UG Student
Department of Electronics & Telecommunication Engineering
Sinhgad Academy of Engineering India

Burhanuddin J. Panwala
UG Student
Department of Electronics & Telecommunication Engineering
Sinhgad Academy of Engineering India

Pulkit Mudgal
UG Student
Department of Electronics & Telecommunication Engineering
Sinhgad Academy of Engineering India

Chandrashekhar G. Patil
Associate Professor
Department of Electronics & Telecommunication Engineering
Sinhgad Academy of Engineering India

Abstract

The field of automation has gained a lot of importance in recent years due to its ability to reduce human efforts. Today a number of systems hitherto operated manually have been automated and are now operated by robotic systems. In various military and industrial applications, automaton can help avoid loss of unnecessary human lives. A single robotic system can accomplish a task which otherwise required the efforts of a number of humans. Such robotic systems can be used to accomplish trivial tasks that do not require human instinct. The "Simultaneous Localization and Mapping" (SLAM) system will allow a user to map an unknown environment using a robotic system. In this project SLAM has been implemented using a Kinect sensor (mounted on a moving platform) to generate a 2-D /3-D map of an unknown indoor environment. This map can be viewed on various devices using a corresponding GUI and will allow users to understand the internal layout of any structure before actually entering it.
Keywords: Automation, SLAM, Kinect, GUI, Layout

I. INTRODUCTION

The "Simultaneous Localization and Mapping" system is one which uses a robotic system that can map its environment while identifying its location in it simultaneously [2][3]. The system uses a Microsoft Kinect sensor to identify features of the robot environment [4]. The sensor is mounted on a moving platform which will be controlled by a remote control. The system will be implemented on "Robotic Operating System" (ROS) which is a platform compatible with a limited number of open source OS. ROS allows improved communication between different bot modules thus allowing the SLAM system to be integrated with any existing bot peripherals and modules. The robot will use feature extraction algorithms to generate a 2-D/3-D map of the bot environment. The map will be generated on a GUI on ROS called "Real-Time Appearance-Based Mapping" (RTAB-MAP) [8]. RTAB-MAP allows the users to view the scanned environment in different angles. In this project, the system has been designed to map an indoor environment, keeping in mind various military and industrial applications.

II. BACKGROUND

The concept of SLAM was first coined and developed by Hugh Durrant-Whyte and John J. Leonard, based on some earlier work by Smith and Cheeseman. Durrant-Whyte and Leonard had originally termed it as "SMAL" but it was later changed to "SLAM" to improve its aesthetic appeal [1]. SLAM is the problem of generating a map of an unknown environment using a mobile robot which can, at the same time, navigate the environment using the same map. The work done by Cheeseman and Whyte defined a statistical basis for describing relationships that exist between landmarks and for manipulating geometric uncertainty. An important part of this work was to prove that there must exist a high degree of correlation between estimates of locations of various landmarks within a map while the number of correlations grow with successive observations.

The conceptual break-through of SLAM came with the realisation that the combination of mapping and localisation could result in a single convergent estimation problem. It was seen that the correlations between landmarks was the most critical part of the problem and the more these correlations grew, the better the solution. All the findings concerning SLAM along with the new acronym, was first presented in a mobile robotics survey paper at the 1995 International Symposium on Robotics Research.

III. PROBLEM STATEMENT

The SLAM problem statement questions the possibility of a robot to autonomously map any given environment (indoor in this case) without any human interference. The participation of a human in the SLAM process, ideally, must be as limited as simply turning the bot on. The SLAM problem gives rise to the following questions:

1) How does an indoor environment with few recognizable objects in the images affect the performance of the algorithm?
2) What adjustments can be suggested to improve the result in such an environment?
3) How can the robot distinguish between two symmetrical rooms?
4) Can the bot automatically re-route its path to map areas behind obstacles?

However in this project the bot movement will be controlled manually while the mapping process is automated. SLAM will be performed using a Kinect sensor which is an RGB-D camera that captures the surroundings in video pace, capturing RGB images with an element of depth. Thus it becomes possible to get 3D maps with coloured textures. The ability to recognize similarities between images is a key factor for a visual SLAM algorithm to be successful otherwise the algorithm obsolete. For this purpose a feature extraction algorithm must be used (explained in later sections).

IV. METHODOLOGY

The SLAM system generates maps using the two following processes:

A. Training Process:

The training process flow-chart is as shown below:

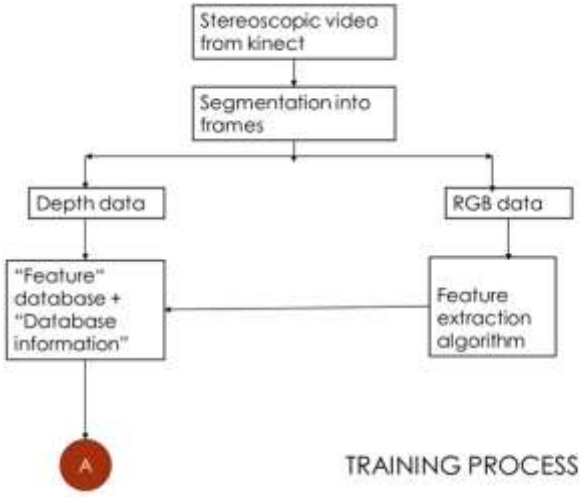

Fig. 1: Training Process Flowchart

In the training process the bot will first be driven into an unknown indoor environment. Then the Kinect sensor mounted on the bot will then take a stereoscopic video (RGB-D format) of the room. This video will be broken down into frames and the RGB and D (Depth) elements will be separated. A feature extraction algorithm is then applied to the RGB data and the various key points of the room are identified [1]. The feature extraction algorithms used will be explained in detail in the upcoming chapters. The extracted features will then be fused with the corresponding depth data and stored in a database.

B. Testing and matching process:

The testing process flowchart is as shown below:

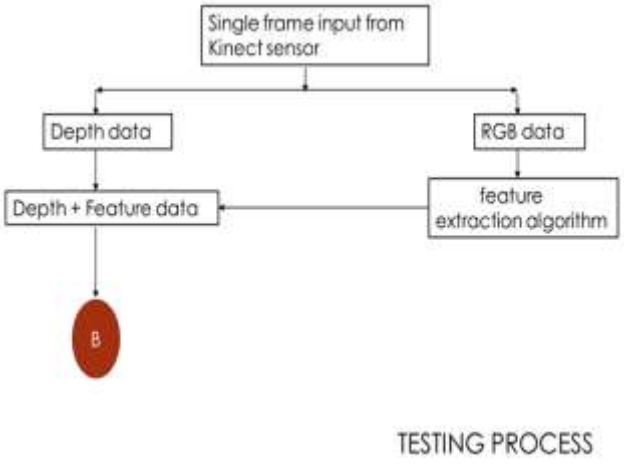

Fig. 2: Testing Process Flowchart

In the testing mode, the bot will take a single frame input from the Kinect and then split the RGB data and the D (Depth) data. The feature extraction algorithm will be applied to this new image and the key points will be identified. These key points will then be fused with the corresponding depth data and the final output will be compared with the database images.

If there is a match between the new image and an existing image, then the bot understands that it has already mapped the area and thus it can locate itself in the existing map by relating the new image with the wheel odometer data. This is called localization and is shown below.

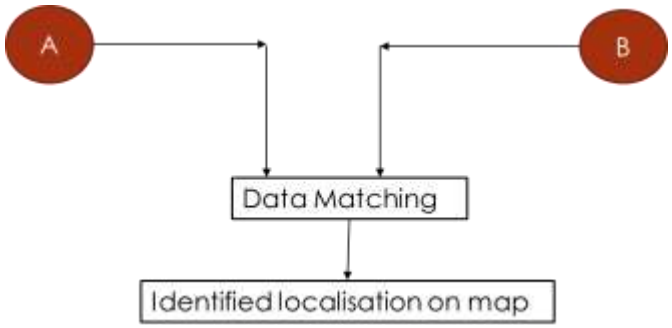

Fig. 3: Matching Process Flowchart

The matching process allows the robot to understand the extent of its mapping. This means that the bot can avoid unnecessary re-mapping of any particular region resulting in a more accurate map and better localization.

V. IMPLEMENTATION

The following are the elements used in the working of the SLAM system:

A. Kinect Sensor:

Kinect is a motion sensor which is a patented property of Microsoft[4][5]. Kinect is a RGB-D sensor which captures RGB data while using IR beam to record depth. The camera can capture features at video pace thus allowing the possibility for 3-D maps. Kinect sensor is a horizontal bar connected to a base with a motorized pivot. It is designed to be positioned lengthwise above or below any display. The device has various applications such as full body 3-D motion capture, facial recognition, gesture recognition, etc.

The different ranges of operation of a Prime Sense PS1080A model are shown below:

- Field of View (Horizontal, Vertical, Diagonal) = 58° H, 45° V, 70° D
- Spatial x/y resolution (@ 2 m distance from sensor) = 3 mm
- Depth z resolution (@ 2 m distance from sensor) = 1 cm
- Operation range = 0.8 m - 3.5 m

The Kinect sensor, as mentioned above, is used to capture images from the environment. These captured images along with their depth elements are stored in a database which is later used for localization.

B. Robotic Operating System (ROS):

Robot Operating System (ROS) is a platform which provides operating system-like functionality on a heterogeneous computer cluster. ROS provides various standard operating system services such as hardware abstraction, low-level device control, message-passing between processes, and package management. ROS is available on a limited number of open source platforms such as Ubuntu, Android etc.

Different ROS terminologies:-

1) Packages:-Software in ROS is organized in packages. A package might contain ROS nodes, a ROS-independent library, a dataset, configuration files, a third-party piece of software, or anything else that logically constitutes a useful module.
2) Stacks: - A stack is a collection of related packages. Packages in ROS are organized into ROS stacks.
3) Nodes: - One of the basic goals of ROS is to enable roboticists to design software as a collection of small, mostly independent programs called nodes that all run at the same time.
4) ROS master: - Nodes must be able to communicate with one another. The part of ROS that facilitates this communication is called the ROS master.

Essentially ROS allows different modules of the same bot to communicate with each other. Thus it is possible for different bot functionalities to establish synchronization with each other. This is possible by the use of packages within different nodes such that each node corresponds to functionality while being controlled by a common master node.

C. *Real Time Appearance Based Mapping (RTAB-MAP):*

RTAB-Map is a RGB-D Graph-Based SLAM approach based on an additive appearance-based loop closure detector [9]. Loop closure detection is essential for the mapping process as it determines map completion and also allows the bot to localize itself. The underlying structure of the map essentially is a graph which consists of nodes and links. The nodes save the odometry poses for each location in the map. The nodes also contain visual information like laser scans, RGB images, depth images and visual words used for loop closure detection [8][9]. The links store rigid geometrical transformations between nodes. There are two types of links: neighbour and loop closure. Neighbour links are added between the current and the previous nodes with their odometry transformation. Loop closure links are added when loop closure detection is found between the current node and one from the same or previous maps. The various operations required for mapping are:

1) Loop Closure Detection:

For global loop closure detection, the "Bag-of-Words" approach is used to form a visual dictionary by generating visual words using SURF (Speeded up Robust Features) and SIFT (Scale Invariant Feature Transform) [6][10][11]. In this approach, a Bayesian filter is used to evaluate loop closure hypotheses over all the previous images. When a loop closure hypothesis reaches some pre-defined threshold H, a loop closure is detected. Visual words are used to compute the likelihood required by the filter. Here the RGB image, from which the visual words are to be extracted, is combined with a depth image, i.e., for each 2D point in the RGB image, a corresponding 3D position can be computed using the calibration matrix and the depth information given by the depth image [9]. When a loop closure is detected, the rigid transformation between the matching images is computed by a RANSAC (Random Sample Consensus) approach using the 3D visual word correspondences. If a minimum of inliers are found, loop closure is accepted and a link with this transformation between the current node and the loop closure hypothesis node is added to the graph. RANSAC essentially provides a yes or no answer to determine whether loop closure has been completed.

2) Graph Optimization:

Generally in the field of robotics, the problem of minimizing non-linear errors is always present. These error functions can always be expressed as a graph. The overall goal in graph optimization is to find the configuration of parameters or state variables that maximally explain a set of measurements that are affected by Gaussian noise. TORO (Tree-based network Optimizer) is the graph optimization approach used in this project [7]. Here node poses and the link transformations are used as constraints. When loop closures are found, the errors introduced by odometry can then be propagated to all links, thus correcting the map. It is relatively straightforward to use TORO to create a tree from the map's graph when there is only one map: the TORO tree has therefore only one root.

3) Memory Management for Mapping:

For real time mapping any incoming data must be processed faster than the time required to acquire them. RTAB-Map employs a memory management approach is used to maintain a graph, which can be managed online using loop closure detection and graph optimization algorithms, thus making the metric SLAM approach independent of the size of the environment [7]. The memory is composed of a Short-Term Memory (STM), a Working Memory (WM) and a Long-Term Memory (LTM), as shown by Figure. The STM is the point of entry for new nodes to be added to the graph whenever new data is acquired, with a fixed size S. When the STM size reaches S nodes, the oldest node is moved to WM to be considered for loop closure detection. The WM size depends on a fixed time limit T.

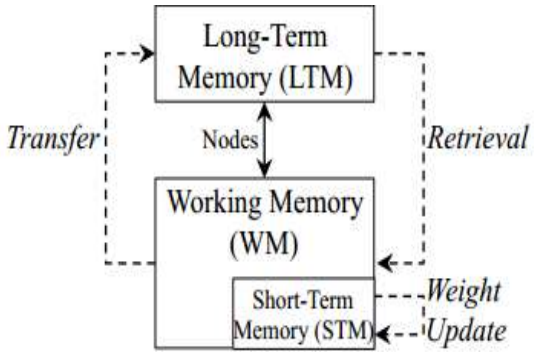

Fig. 4: Memory Management Model

When the time required to process the new data reaches T, some nodes of the graph are transferred from WM to LTM, thus keeping the WM size nearly constant. Whenever loop closure is detected, neighbours in LTM of the old node can be transferred back to WM (a process called Retrieval) for further loop closure detections. In other words, when a robot revisits an area which was previously forgotten, it can remember the area if a least one node of this area is still in WM.

VI. RESULTS

RTAB-Map generates two types of maps as per the input from the Kinect sensor. It generates a 3-D map of the bot environment as shown below:

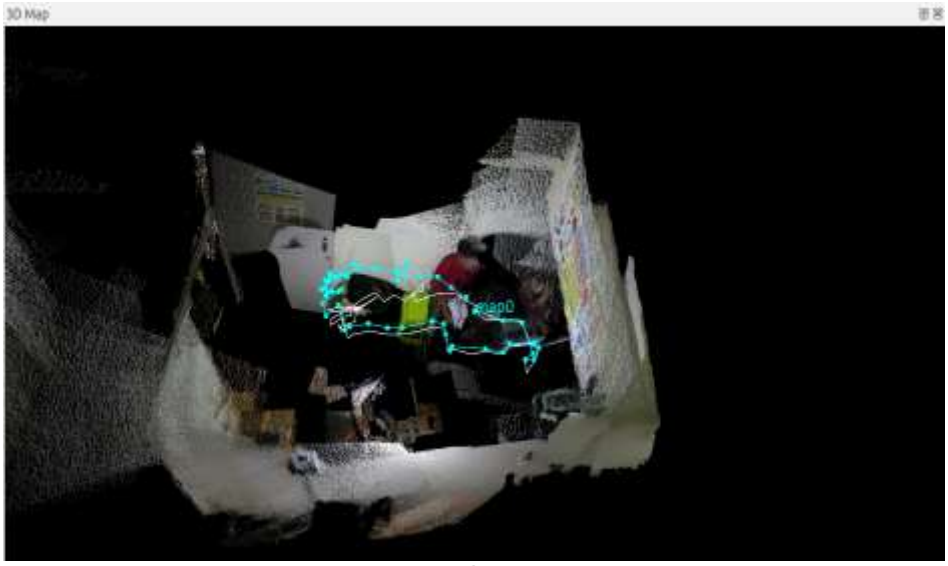

Fig. 5: 3D Map

While simultaneously a two dimensional map is generated using the visual odometry data from the Kinect. This is called as the graphical view of the environment and is the desired result of the project. This map essentially consists of the depth element only while RGB data is ignored as shown below:

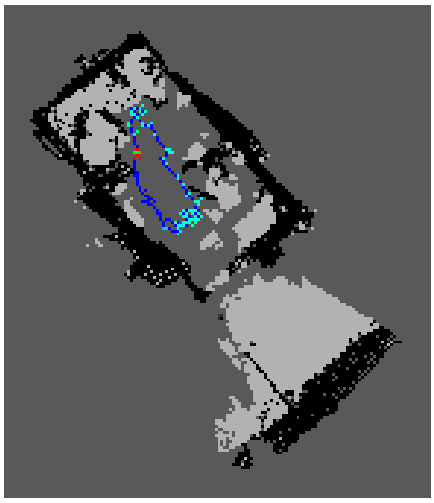

Fig. 6: 2D Map

VII. FUTURE SCOPE

SLAM can find multiple applications as the field of robotics grows. The SLAM problem can help in the development of household bots by training the bots in the house layout so that it can navigate around the house on its own.
Moreover the SLAM problem can be used for designing robots that can also find application in the field of industrial construction.

VIII. CONCLUSION

In this paper the SLAM problem was solved by applying the graphical approach for mapping. The resulting map can successfully describe the layout of the indoor bot environment while tracking the bot movements within the map (localization).

REFERENCES

[1] Søren Riisgaard and Morten Rufus Blas, "A Tutorial Approach to Simultaneous Localization and Mapping," Massachusetts Institute of Technology, 2004.
[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i," IEEE Robotics and Automation Magazine, vol. 13, no. 1, pp. 99–110, 2006.
[3] T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (slam): Part ii," IEEE Robotics and Automation Magazine, vol. 13, no. 3, pp. 108–117, 2006.
[4] Schindhelm, C.K. "Evaluating SLAM Approaches for Microsoft Kinect." In ICWMC 2012, The Eighth International Conference on Wireless and Mobile Communications, pp. 402-407. 2012.

[5] "Microsoft kinect," http://www.xbox.com/en-IN/Kinect.
[6] M. Labbe and F. Michaud, "Memory management for real-time appearance-based loop closure detection," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2011, pp. 1271–1276.
[7] M. Labbe and F. Michaud, "Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2014, pp. 2661–2666.
[8] M. Labbe and F. Michaud, "Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation," IEEE Transactions on Robotics, vol. 29, no. 3 , pp. 734 – 745, February 2013.
[9] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," IEEE Transactions on Robotics, vol. 24, no. 5, pp. 1027–1037, October 2008.
[10] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded Up Robust Features (SURF)," Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346–359, 2008.
[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.