International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 3, May-Jun-2019, ISSN (Online): 2395-566X

Simultaneous Localization and Mapping on a Quadcopter

Preeti Keshava Lakmapur Rakshith B H Rakshitha S R Shiva Prasad K S
Asst. Prof. Pushpalatha S
Dept. of Information Science and Engineering
Dr. Ambedkar Institute of Technology
Bangalore,Karnataka, India

***Abstract*-Simultaneous Localization and Mapping (SLAM), in robotics, is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of a robot's location within it. In an outdoor environment, GPS is used to address such issues to track the robot's location in a map. But in cases like indoor environment or when the GPS is unavailable, SLAM comes into picture. The paper aims to implement Visual-SLAM using Kinect camera on a quadcopter. The Visual-SLAM algorithm has been implemented on x86_64 architecture-based intel core i7 processor on a laptop with Ubuntu 12.04 running as the host OS. The algorithm uses Kinect camera that has both RGB camera and an Infrared Depth camera to obtain the pictures and depth information of the frame. In addition to this OpenCV and several ROS packages like Octomap Server, eigen, cv bridge etc. has been used to support its functionality. The front-end of the algorithm takes care of the observations being made by the robot while the back-end builds a 3D map using these observations and localizes the robot within it. This paper hence provides a 3D map of an unknown environment using kinect camera on a quadcopter (Jatayu).**

***Keywords* – 3D Mapping, Kinect Camera, Quadcopter, Drone.**

I. INTRODUCTION

In today's world, the usage of drones for Surveillance has increased rapidly. The next generation of remotely piloted aircraft could swarm enemy defenses, serve as wingmen for pilots, attack targets with lasers, or work as mobile weather radar, estimation of wildlife, can help detect life during natural calamities, map the area and plan for attacks and detection of intruders at the border. The Jatayu uses the technique of Simultaneous Local isation and Mapping (SLAM) for its movement and also to map the surroundings whereas having aThermal imaging camera on board which transmits the live images/video of the area, gives it the ability to deploy it at nights.

Fig.1 Quadcopter with on-board Kinect camera

1. System Overview

The project aims at delivering a quadcopter model which can build a 3D map of any given environment using Microsoft Kinect and also navigate in outdoors using GPS. The emphasis lies on object recognition using techniques based on detection of points of interest on objects or surfaces. This is achieved by extracting features such as corners and edges. These features are made invariant to image location, rotation or scale to track the points of interests. Some of the methods used to build the model are Harris Corner to detect corner, SIFT, SURF etc.

It is based on two important aspects of the features- one is the detection of the key point and the second is its descriptor. The descriptor identifies a region with a strong variation of intensity like the edges and corners. Measuring of the main orientations of the surrounding points gives us the detector. The model is further extended to work in the dark as the depth camera is always in function. The external surroundings are visible but mapping doesn't happen. Hence these features give Jatayu the ability to be deployed at nights as well and in any unknown environments.

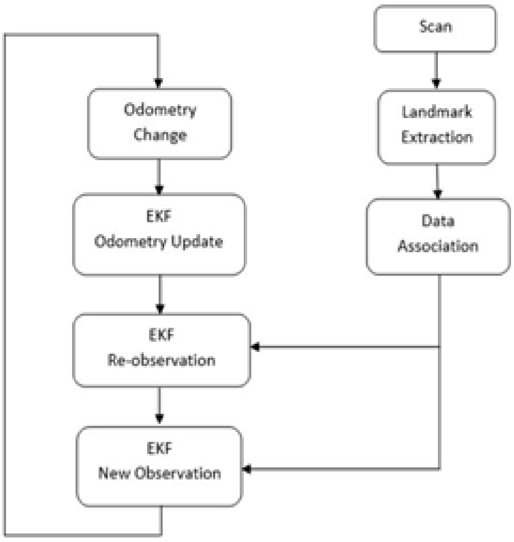

Fig. 2 Overview of Slam Process.

II.LITERATURE SURVEY

Fioraio and Konolige recently presented a system that uses bundle adjustment to align the dense point clouds of the Kinect directly however without further exploiting the RGB images. Most similar to their approach is the work by Henry et al. Their approach uses sparse key point matches between consecutive colour images as an initialization to ICP. In their experiments, they found however that often the computationally expensive ICP step was not necessary. Therefore, they improved the algorithm so that ICP was only used if few (or none) key point matches could be established. Henry et al. post-process the resulting point cloud into a surf-el representation, while here a volumetric voxel representation is created that can directly be used for robot localization, path planning and navigation.

III. HARDWARE DESCRIPTION

1. Microsoft Kinect

Kinect is a line of motion sensing input devices by Microsoft for Xbox360 and Xbox One video game consoles and Windows PCs. Based around a webcam-style add-on peripheral, it enables users to control and interact with their console/computer without the need for a game controller, through a natural user interface using gestures and commands. But here, kinect is being used to obtain the 3D image with the depth perception and thereby to access the 3D point cloud data for mapping. The device features an RGB camera, depth sensor and multi-array microphone running proprietary software which provide full-body 3D motion capture, facial recognition and voice recognition capabilities

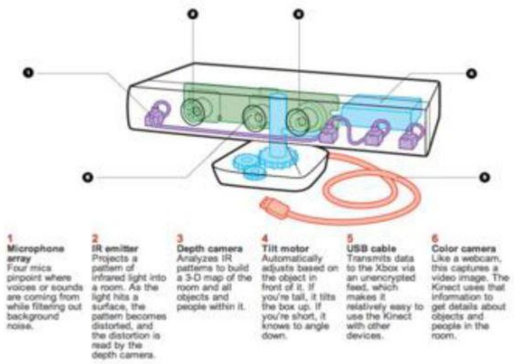

Fig. 3 Kinect Camera.

2. Lipo Battery

Lithium polymer batteries, more commonly known as LiPo, have high energy density, high discharge rate and light weight which make them a great candidate or RC applications. A lithium polymer battery is a rechargeable battery of lithium-
ion technologyusinga polymer electrolyte instead of a liquid electrolyte. High conductivity semisolid (gel) polymers form this electrolyte. These batteries provide higher specific energy than other lithium battery types and are used in applications where weight is a critical feature, like mobile devices and aircraft. The capacity of a LiPo battery is measured in mAh (milli-amp hours). “mAh” is basically an indication of how much current you can draw from the battery for an hour until it’s empty. 4200 mAH lipo battery is used here.

Fig. 4 Lipo Battery

3. Electronic Speed Controller

An electronic speed control or ESC is an electronic circuit that controls and regulates the speed of an electric motor. It may also provide reversing of the motor and dynamic braking. Miniature electronic speed controls are used in electrically powered radio controlled models. Full-size electric vehicles also have systems to control the speed of their drive motors.

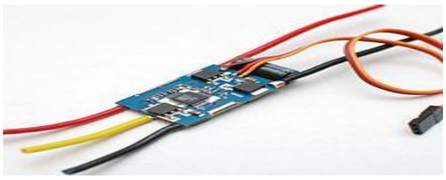

Fig. 5 Electronic Speed Controller.

4. Brushless Motors

A brushless DC electric motor also known as electronically commutated motor and synchronous DC motors, are synchronous motors powered by DC electricity via an inverter or switching power supply which produces an AC electric current to drive each phase of the motor via a closed loop controller. The controller provides pulses of current to the motor windings that control the speed and torque of the motor.

Fig. 6 Brushless Motor.

5. Pixhawk

It is an open-source autopilot system oriented toward inexpensive autonomous aircraft. Low cost and availability enable hobbyist use in small remotely piloted aircraft. The project started in 2009 and is being further developed and used at Computer Vision and Geometry Lab of ETH Zurich (Swiss Federal Institute of Technology) and supported by the Autonomous Systems Lab and the Automatic Control Laboratory. Several vendors are currently producing PX4 autopilots and accessories. An autopilot allows a remotely piloted aircraft to be flown out of sight. The Pixhawk autopilot module runs a very efficient real-time operating system (RTOS), which provides a POSIX-style environment (i.e. printf(), pthreads, /dev/ttyS1, open(), write(), poll(), ioctl(), etc). The software can be updated with an USB bootloader. PIXHAWK is the industry standard autopilot, designed by the open hardware development team in collaboration with 3DRobotics.

Fig. 7 PIXHAWK

6. Ublox Neo M8n Gps Module

Ublox NEO-M8N High Precision GPS Module with Built in Compass for APM and PIXHAWK FC is the new Ublox Neo-M8N GPS module that includes an HMC5883L digital compass. This module has a high level of sensitivity and features active circuitry for the ceramic patch antenna. It also comes enclosed in the plastic case to protect the module from the elements. This module outputs precise position updates at 10Hz and also has a rechargeable backup battery for warm starts. The Ublox NEO-M8N is configured to run at a baud rate of 38400 and can be used with Pixhawk and APM. Additionally, it can be connected the same module to all versions of PIXHAWK Flight Controller just after swapping its 5-pin connector with an extra 6-pin connector provided in Package.

FIG. 8 Ublox Neo M8n Gps Module

7. Telemetry

Telemetry is used to retrieve flight information of the drone on a computer or on a radio control in order to follow several parameters of your aircraft on the ground. One of the really nice features of modern-day remote-control radios is the fact that they can send signals to an RC multirotor to control it. Plus, they can receive signals back from the same aircraft, allowing to monitor important information about that aircraft. This feature is referred as telemetry. This is used to monitor information from a variety of sensors that is installed on the aircraft. This allows the pilot to keep track of the aircraft, monitor external things around the quad, and have real-time information on key components of the multirotor, such as battery levels.

Fig. 9 Telemetry.

8. Carbon Fiber Frame

Carbon fiber is a composite material, being made up of many layers of interwoven carbon fibers that have been rigidly cemented within a binding matrix of epoxy. The popularity of carbon fiber as a frame material is due to its low weight and high strength. Apart from being durable, this carbon fiber drone frames comes with much space .Carbon fibers are fibers about 5–10 micrometers in diameter and composed mostly of carbon atoms. Carbon fibers have several advantages including high stiffness, high tensile strength, low weight, high chemical resistance, high temperature tolerance and low thermal expansion. These properties have made carbon fiber very popular in aerospace, civil engineering, military, and motorsports, along with other competition sports.

Fig. 10Carbon Fiber Frame

IV. SOFTWARE DESCRIPTION

1.Ubuntu Operating System

Ubuntu is a complete Linux operating system, freely available with both community and professional support. The Ubuntu community is built on the ideas enshrined in the Ubuntu Manifesto: that software should be available free of charge, that software tools should be usable by people in their local language and despite any disabilities, and that people should have the freedom to customize and alter their software in whatever way they see fit.

2.OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl, Ch, Haskell and Ruby have been developed to encourage adoption by a wider audience.Since version 3.4, OpenCV.js is a JavaScript binding for selected subset of OpenCV functions for the web platform. All of the new developments and algorithms in OpenCV are now developed in the C++ interface.

3. Ros Hydro

Robot Operating System (ROS or ros) is robotics middleware .Although ROS is not an operating system, it provides services designed for a heterogeneous computer clustersuch as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. Running sets of ROS-based processes are represented in a graph architecture where processing takes place in nodes that may receive, post and multiplex sensor data, control, state, planning, actuator, and other messages. Despite the importance of reactivity and low latency in robot control, ROS itself is not a real-time OS (RTOS). It is possible, however, to integrate ROS with real-time code.

3.1Ros Hydro Medusa

ROS Hydro Medusa is the seventh ROS distribution release and was released on September 4th 2013. In this release they have focused on converting many of the packages in ROS over to the new catkin build system while also fixing and improving core ROS components. Also, there were many improvements in tools like rviz and rqt as well.

3.2 G2o Library

g2o is an open-source C++ framework for optimizing graph-based nonlinear error functions. g2o has been designed to be easily extensible to a wide range of problems and a new problem typically can be specified in a few lines of code. The current implementation provides solutions to several variants of SLAM and BA.

3.3 Qt LIBRARY

Qt is a free and open-source widget toolkit for creating graphical user interfaces as well as cross-platform applications that run on various software and hardware platforms such as Linux, Windows, mac OS, Android or embedded systems with little or no change in the underlying codebase while still being a native application with native capabilities and speed. QT is used for developing graphical user interfaces (GUIs) and multi-platform applications that run on all major desktop platforms and most mobile or embedded platforms. Most GUI programs created with Qt have a native-looking interface, in which case Qt is classified as a widget toolkit. Also non-GUI programs can be developed, such as command-line tools and consoles for servers. An example of such a non-GUI program using Qt is the Catalyst web framework.

3.4 CMAKE

C Make is an extensible, open-source system that manages the build process in an operating system and in a compiler-independent manner. Unlike many cross-platform systems, C Make is designed to be used in conjunction with the native build environment. Simple configuration files placed in each source directory (called CMakeLists.txt files) are used to generate standard build files which are used in the usual way. CMake can generate a native build environment that will compile source code, create libraries, generate wrappers and build executables in arbitrary combinations. CMake supports

International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 3, May-Jun-2019, ISSN (Online): 2395-566X

in-place and out-of-place builds, and can therefore support multiple builds from a single source tree. CMake also supports static and dynamic library builds. Another nice feature of CMake is that it generates a cache file that is designed to be used with a graphical editor.

3.5 Q MAKE

Q make is a utility that automates the generation of make files. Make files are used by the program make to build executable programs from source code therefore qmake is a make-make file tool, or make for short.`Qmake` is a tool that helps simplify the build process for development project across different platforms. `Qmake` automates the generation of Make files so that only a few lines of information are needed to create each Make file. `qmake` can be used for any software project, whether it is written in Qt or not.`Qmake` generates a Make file based on the information in a project file. Project files are created by the developer, and are usually simple, but more sophisticated project files can be created for complex projects. `Q make` contains additional features to support development with Qt, automatically including build rules for moc and uic. `Q make` can also generate projects for Microsoft Visual studio without requiring the developer to change the project file.

V.IMPLEMENTATION

1. Frontend

The Frontend is responsible for establishing spatial relations from the sensor data. The system computes pair wise relations between camera images by matching of visual features. OpenCV is used for detection, description and matching of various feature types, namely SURF, SIFT and ORB. Since SIFT features are computationally much more demanding than SURF and ORB, we make use of a GPU based implementation of SIFT.

2. Backend

The pair wise transformations between sensor poses, as computed by the frontend, form the edges of a pose graph. Due to estimation errors, the edges form no consistent trajectory. To create a consistent trajectory, optimize the pose graph using the g2o framework. The g2o framework is an extensible graph optimizer that can be applied to a wide range of problems. It performs a minimization of a non-linear error function that can be represented as a graph.

3. Mapping

The system computes a globally consistent trajectory. Using this trajectory, the original data can be used to construct a representation of the environment by projecting all point measurements into a global 3D coordinate system. This however, is highly redundant and requires large computational and memory resources. To overcome this, use 3D occupancy grid maps to represent the environment. Octo Map which is an octree-based mapping framework is used.

VI. RESULT

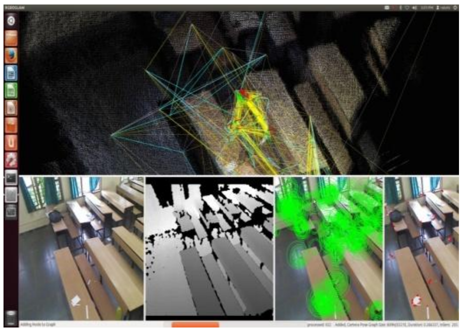

Fig. 10 The system takes RGB and Depth images as input from kinect camera which is mounted on Jatayu and uses octo map to generate 3D maps

1. Features

- Probabilistic estimation of the state of the camera in motion.
- Probabilistic estimation of the 3D map obtained.
- To help in efficient processing.
- Obtaining a sparse map of high-quality features.

2. Applications

- At home: vacuum cleaner, lawn mower.
- Air: surveillance with unmanned air vehicles
- Military Applications (Aerial/ Area mapping and planning attacks).
- Mapping unknown buildings and places where humans cannot reach.
- Forest Combing for terrorists and Smugglers using Thermal imaging.
- Can be used at the borders to keep an eye on intruders.
- Detect life in areas affected by natural calamities.
- Detection of fire and wild life in forest areas and agriculture applications.

VII. CONCLUSION

The system successfully implements 3D mapping on a Quadcopter. The whole system is designed keeping in mind the applications of the final product. The system captures the images of the environment using Kinect camera. The captured images in the form of RGB and depth images are used for further processing. The prominent visual features such as corners from the incoming color images are extracted that is Feature Extraction. These features are matched against features from previous images and using the eigen transformation the new position of the robot is evaluated, that is Motion Tracking using SIFT. Using this new localized point, the pixels are projected in RGB frame with its corresponding depth frame values to obtain the Point Cloud Data (PCD) or the 3D map. Once the process of obtaining the frame is stopped (or paused), the HOGMAN algorithm starts optimizing the pose graph hierarchically to reduce any errors caused during the active mapping. The system is designed to carry out missions in both indoor and outdoor environments. Hence GPS is used for outdoor navigation and SLAM for indoor navigation.

REFERENCES

[1]. Philipp Robbel, David Demirdjian, Cynthia Breazeal. "Simultaneous Localization and Mapping with People". RSS Workshop on Advanced Reasoning with Depth Cameras, June 2011

[2]. Pratik Agarwal, Gian Diego Tipaldi, Luciano Spinello, Cyrill Stachniss, and Wolfram Burgard. "Robust map optimization using dynamic covariance scaling." Proceedings - IEEE International Conference on Robotics and Automation, pages 62–69, 2013.

[3]. Andrew J. Davison."Real-Time Simultaneous Localisation and Mapping with a Single Camera." University of Oxford, OX1 3PJ, UK. Pages 1403 - 1410 vol.2 2003

[4]. Chee Sing Lee."A Review of Submapping SLAM techniques."A Thesis Submitted for the Degree of MSc Erasmus Mundus in Vision and Robotics (VIBOT) 2010.

[5]. Soren Riisgaard and Morten Rufus Blas. "SLAM for Dummies." A Tutorial Approach to Simultaneous Localization and Mapping.

[6]. Viet Nguyen, AhadHarati, Roland Siegwart. "A Lightweight SLAM algorithm using Orthogonal Planes for Indoor Mobile Robotics." Swiss Federal Institute of Technology (ETH Zurich).

[7]. NiklasKarlsson, Enrico Di Bernardo, Jim Ostrowski, Luis Goncalves, Paolo Pirjanian, and Mario E. Munich. "The vSLAM Algorithm for Robust Localization and Mapping. "Published in Proc. of Int. Conf. on Robotics and Automation (ICRA) 2005.