i - SLAM

Modular Autonomy using Consumer Technology

Aditya Bondada, Hardik Devrangadi, Harin Kumar Nallaguntla,
Prathima Ananda Kumar, Sriram Kodeeswaran, Thejaswini Goriparthi

1 Overview

We present a method for using an iPhone's sensors, specifically the LiDAR, Camera, and IMU data streams, to perform simultaneous localization and mapping (SLAM) using LiDAR, visual SLAM, and dead reckoning. By leveraging the capabilities of a commonly available consumer device - the iPhone, the proposed approach allows individuals to easily build their own autonomous robots. We demonstrate the effectiveness of the approach through experiments and discuss potential applications and future directions for this type of technology.

The advantage of using an iPhone with a LiDAR scanner as part of a mobile platform is the versatility and flexibility of the system. Because the iPhone can be easily attached to any mobile platform, it provides a modular and scalable solution that can be adapted to a wide range of applications and environments.

2 Background

Modular autonomy, the integration of interchangeable components to enable autonomous behavior in a system, can be achieved through various approaches. Specialized autonomous systems, tailored to specific tasks or applications, offer high reliability and efficiency, but may be costly and inflexible for other purposes. Alternatively, consumer technology such as smartphones can provide a cost-effective and adaptable solution, although it may not be as reliable or durable as specialized systems. Traditional sensors attached to a mobile platform also enable the use of high-quality sensors, but may incur higher costs and complexity in the integration process.

In contrast, the current project utilizes a smartphone equipped with LiDAR, IMU, and camera sensors to perform tasks such as SLAM, autonomous navigation, and mapping. By attaching the smartphone to a mobile platform such as an RC car or drone, modular autonomy is effectively solved through the use of consumer technology in a cost-effective and flexible manner. This approach has the potential to be applied in various contexts, including disaster response, reconnaissance, delivery robots, and mine exploration.

3 Objectives

The aim of the project is defined as three objectives:

1. Plot a trajectory of the LiDAR data recorded from an iPhone.
2. Plot a 3D and 2D trajectory of a video that is recorded from iPhone's camera with the PySLAM 2 algorithm.
3. Plot a trajectory of the path taken with the iPhone's sensors using the dead reckoning algorithm with MATLAB Mobile.

4 Sensors and Datasets

We have collected three datasets using an iPhone following the same trajectory and analyzed the data through different algorithms to obtain a mapping of the area in various forms.

Firstly, we have installed an application called "Record3D" on the iPhone that scans the area and gives LiDAR data. This application can stream live data in two ways: via USB and over Wi-Fi. RTAB-Map is also installed as an application on the Windows laptop that performs the mapping. The iPhone's camera is calibrated manually and the camera intrinsics are obtained. The LiDAR data obtained is saved as a sequence of images which is then processed by the RTAB-Map algorithm via USB live streaming.

Secondly, using an iPhone camera an RGB video is recorded and processed by the PySLAM 2 algorithm that detects visual landmarks to obtain the 2D and 3D trajectory.

Thirdly, utilizing the MATLAB application on iPhone consisting of motion sensors - Gyroscope, Magnetometer and Accelerometer, we performed dead-reckoning. The sensors on the iPhone are already calibrated off the shelf.

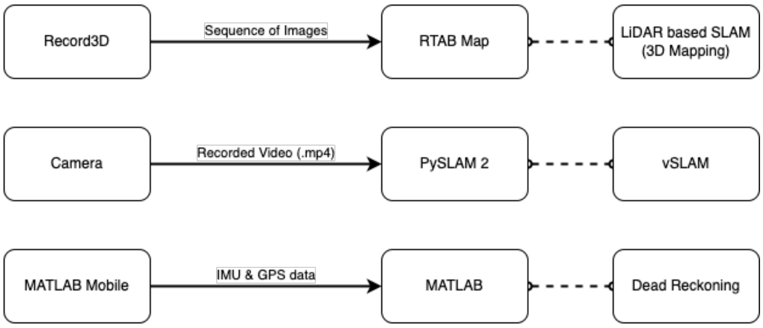


Figure 1: Different sources of datasets collected

The iPhone 14 Pro and iPhone 14 Pro Max are equipped with several sensors that can be used for a variety of applications. The LiDAR sensor is capable of accurately measuring the distance to nearby objects using lasers, and has a range of up to 5 meters and a scan rate of up to 60 fps. The Angle Random Walk Error error of Accelerometer and Gyroscope are 180 $\mu/\sqrt{hz}$ and $0.007°/s/\sqrt{hz}$ respectively [1]. The camera on the iPhone is a high-quality camera that is set to take photos and videos at 30 fps and 1080p. The GPS system on the iPhone can determine the device's location using signals from GPS satellites, which are upto 2m accurate (shown in MATLAB app).

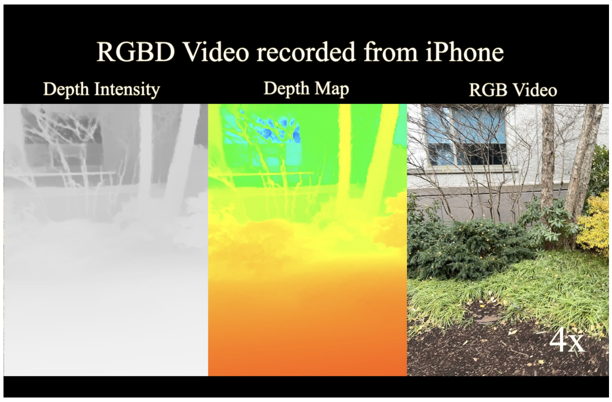


Figure 2: Processing of the LiDAR data + RGB video obtained from the iPhone

5 Algorithms

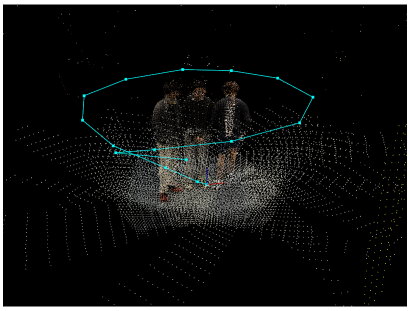
SLAM (Simultaneous Localization and Mapping) uses a combination of sensors and computer vision techniques to perceive the environment and estimate the pose (position and orientation) of the robot in real-time [2]. In our project, we have used RTAB-Map with LiDAR data, PySLAM with visual data and have performed dead reckoning with IMU data to obtain the path.

5.1 RTAB-Map

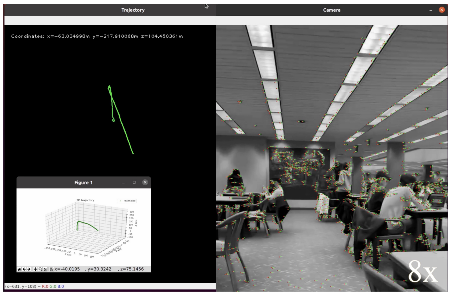
Real-Time Appearance-Based Mapping (RTAB-Map) is a LiDAR SLAM algorithm that can be used on mobile robots and autonomous vehicles to build a 2D or 3D map of the environment while simultaneously tracking its position and orientation. It has several advantages including real-time performance, loop closure detection [4], appearance-based mapping, compact and lightweight design. RTAB-Map uses LiDAR features in the environment to create the map, rather than relying on pre-defined landmarks or markers as seen in Figure 3a, making it suitable for use in a wide range of environments. It can also detect when the robot or device has returned to a previously visited location, allowing it to correct and improve the map by correcting for drift in real time, improving the accuracy of the map [5].

5.2 PySLAM 2

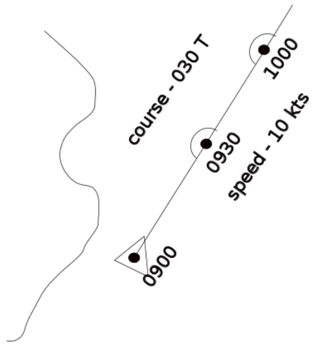
PySLAM is a Python library that provides tools for developing and testing Simultaneous Localization and Mapping (SLAM) algorithms [6]. It is built on top of the OpenCV library and can use a camera or a set of stereo images as input. PySLAM supports various types of SLAM algorithms, including monocular, stereo, and RGB-D [7]. It includes scripts for simple visual odometry (VO) and more advanced SLAM pipelines that include multiple frame feature tracking,



(a) Point Cloud and Camera Path/Poses generated from RTAB-Map



(b) Path generated from PySLAM using Feature Landmarks



(c) Dead Reckoning [3]

Figure 3: Paths generated by different algorithms using iPhone sensor data

point triangulation, keyframe management, and bundle adjustment. PySLAM has several advantages, including ease of use, modularity, compatibility with different platforms and environments, open source accessibility, and a supportive community. We used PySLAM in our project to gather data points in a video dataset and calculate the estimated path, 3D trajectory, and 2D trajectory of the dataset as seen in Figure 3b

5.3 Dead Reckoning

Dead Reckoning is a navigation technique that uses a combination of sensors and algorithms to estimate the position and orientation of a mobile device or vehicle. It is often used in situations where GPS signals are unavailable or unreliable. In dead reckoning, the mobile device uses sensors such as an accelerometer, gyroscope, and magnetometer to measure the linear and angular motion of the device. These measurements are combined with a known initial position and orientation to estimate the current position and orientation of the device. We use dead reckoning as PySLAM and RTAB-Map depends on the environment and the features detected, which means that the path obtained are heavily dependent on external factors. However, dead reckoning uses a different approach by relying on raw sensor data to obtain the path.

6 Analysis Approach

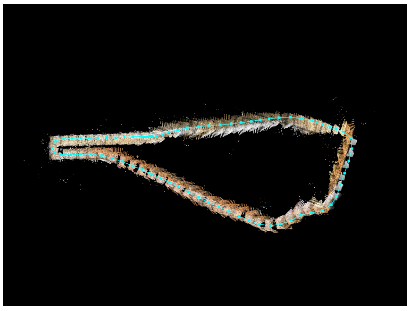
We took a quadrilateral path with four turns and compared the paths predicted from SLAM, vSLAM and Dead Reckoning. While we expected the predicted paths to match the shape of the quadrilateral path, it is unrealistic to expect perfect accuracy due to the presence of noise in sensor data, which can cause imperfections in the collected data and subsequently in the predicted path.

To evaluate the performance of the system and identify areas for improvement, we used a variety of approaches. First, the maps generated by RTAB-Map were visually inspected to ensure that they accurately represented the environment and captured relevant features. Second, the results of the visual SLAM algorithm were plotted by the PySLAM 2 algorithm that allowed us to assess the accuracy of the localization and the robustness of the algorithm to various environmental conditions. Third, the results of the dead reckoning algorithm were evaluated by comparing the estimated trajectory of the platform to the ground truth trajectory. This allowed us to assess the accuracy and robustness of the algorithm in cases where external sensors were unavailable or unreliable.

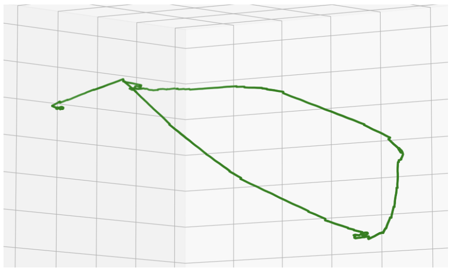
Finally, the results of all three approaches (RTAB-Map, visual SLAM, and dead reckoning) were compared and analyzed to identify the strengths and weaknesses of each approach and determine the most appropriate approach for the given task.

7 Final Results

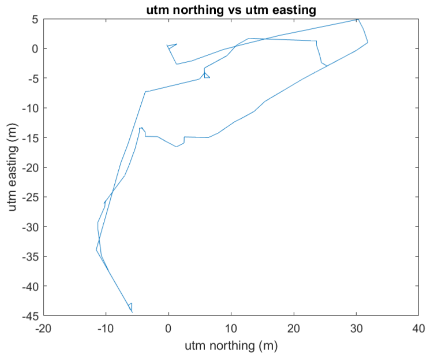
After the parameters were fine-tuned for each of the algorithms, a common dataset was collected over live-stream by following the same path for LiDAR-based SLAM, vSLAM, and Dead Reckoning. These algorithms were executed online. The final predicted paths show similarities, from which a final, most efficient and accurate path can be obtained. The path followed was outside the Snell Library at Northeastern University, so that GPS data could be used as the ground truth. The final predicted paths from SLAM, vSLAM, and Dead Reckoning are explained below.



(a) Image of the path generated from RTAB-Map



(b) Image of the path generated from PySLAM



(c) Path generated from Dead Reckoning

Figure 4: Paths generated by different algorithms using iPhone sensor data

In the case of path generated from RTAB-Map using the iPhone's LiDAR, as seen in Figure 4a, a loop-closure was detected, which led to a very accurate path and pose estimation of the iPhone. From our experiments, it was observed that LiDAR based SLAM worked exceptionally well with more complex environment, as this increases the number of characteristic features for the algorithm to process.

For the path generated based on vSLAM using the PySLAM algorithm, we obtained a path that showed similar resemblance to the other two, as seen in Figure 4b. Due to the requirement of collecting data outdoors (due to GPS), it was found that there were not many feature points or landmarks that could be used as visual indicators by the PySLAM algorithm to plot the most accurate path. Despite this limitation, it is observed that vSLAM performs well, and is able to recognize the turns and the overall characteristic path followed by the iPhone.

The predicted path obtained from Dead Reckoning is observed in Figure 4c. The loop in the path did not appear to close due to inherent noise in the IMU sensor. After filtering out the noise and outliers in the data, this path was obtained. However, with further filtering and with IMU characteristic data, this can be made more accurate.

8 Conclusion

In conclusion, the project aimed to develop a system for modular autonomy using consumer technology, specifically an iPhone equipped with LiDAR, IMU, camera, and GPS sensors. The system was able to generate a map using LiDAR data and RTAB-Map, perform visual SLAM using camera data and PySLAM, and perform dead reckoning using IMU data and MATLAB-Mobile. The predicted paths from these three approaches were compared and evaluated.

The results of this project, i - SLAM, demonstrate the feasibility of using an iPhone to enable autonomous behavior in a mobile platform, such as an RC car or drone. Potential applications of this system include reconnaissance, delivery robots, disaster response, and exploring mines. However, it is important to carefully consider the trade-offs between different approaches and choose the one that is most appropriate for the task at hand. As a potential improvement to the current system, it may be beneficial to combine multiple approaches to take advantage of the strengths of each approach and mitigate their weaknesses.

References

[1] [Online]. Available: https://www.bosch-sensortec.com/products/motion-sensors/imus/bmi160/

[2] [Online]. Available: https://www.mathworks.com/discovery/slam.html

[3] [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/e/ed/Dead-reckoning.svg

[4] M. Labbé 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, 2013.

[5] [Online]. Available: https://github.com/introlab/rtabmap

[6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[7] A. Fontán, J. Civera, and R. Triebel, "Information-driven direct rgb-d odometry," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4928–4936.