ENPM673 Project 4 Proposal
VISUAL SLAM

Team:

Aneesh Chodisetty
UID: 117041088
aneeshch@umd.edu

Bhargav Kumar Soothram
UID: 117041088
bsoothra@umd.edu

Koundinya Vinnakota
UID: 117041088
koundinyav@umd.edu

Madhu Narra Chittibabu
UID: 117041088
mnarrach@umd.edu

Date: 8 April 2022

1 Abstract

SLAM (simultaneous localization and mapping) is a method used for autonomous vehicles that allows map construction and localization of a vehicle in that map at the same time. Engineers can then use the map information to carry out tasks such as path planning and obstacle avoidance.

Visual SLAM (or vSLAM), as the name suggests, uses images acquired from cameras and other image sensors to estimate the map of a given space. It can be implemented at a relatively low cost using inexpensive cameras and since cameras provide a large volume of information, they can be used to detect landmarks (previously measured positions). Landmark detection can also be combined with graph-based optimization, achieving flexibility in SLAM implementation.

Monocular SLAM uses camera (one camera only) as the sensor and this makes depth estimation challenging. There exist a few solutions to this problem though, for example, detecting AR markers, checkerboards or other known objects in the image for localization.

The goal of this project is to implement a simple monocular SLAM using a Raspberry Pi camera and estimate the best path to a given goal based on the acquired map.

2 Implementation

The robot is moved around a given area where there are obstacles and its relative position can be determined by the use of AR markers or checkerboards. Then, ORB-SLAM is performed on the obtained set of images in order to estimate the depth information. If the images are taken with a single calibrated camera, then the 3-D structure and camera motion can only be recovered up to scale.

2.1 Monocular Visual SLAM

We intend to implement the ORB-SLAM, which is a feature-based vSLAM algorithm. Following are the steps involved, in brief:

- ORB-SLAM starts by initializing the map of 3-D points from two video frames. The 3-D points and relative camera pose are computed using triangulation based on 2-D ORB feature correspondences.
- Once a map is initialized, for each new frame, the camera pose is estimated by matching features in the current frame to features in the last key frame. The estimated camera pose is refined by tracking the local map.

- The current frame is used to create new 3-D map points if it is identified as a key frame. At this stage, bundle adjustment is used to minimize re-projection errors by adjusting the camera pose and 3-D points.
- Loops are detected for each key frame by comparing it against all previous key frames using the bag-of-features approach. Once a loop closure is detected, the pose graph is optimized to refine the camera poses of all the key frames.

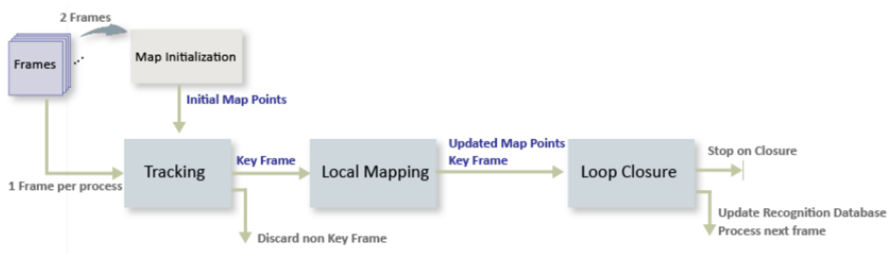

Figure 1: ORB-SLAM pipeline

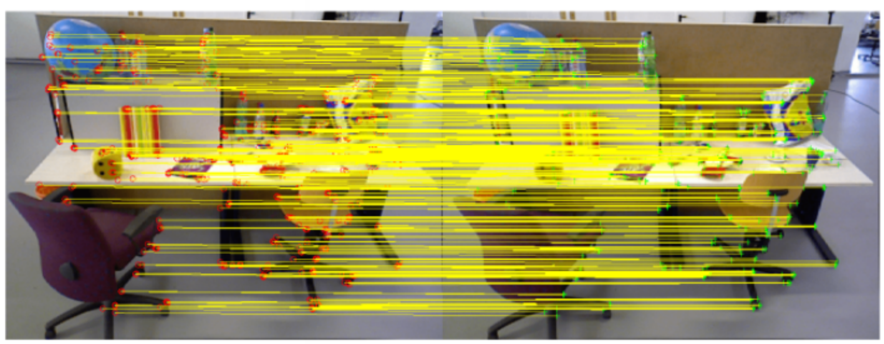

Figure 2: Feature Correspondences (Fundamental Matrix/Homography is computed after)

2.2 Hardware Implementation

Time permitting, we intend to run a Raspberry pi robot to capture the images, process the information and then reach the desired goal along the best path computed. This part needs further updating and suggestions on the implementation.

References

[1] https://www.mathworks.com/discovery/slam.html

[2] "ORB-SLAM: a Versatile and Accurate Monocular SLAM System", Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos

[3] https://www.youtube.com/watch?v=GnuQzP3gty4