***Project Report on CSE 7000 Directed Individual Study***

Name: Swayamjit Saha

Net ID: ss4706

(Fall 2023)

*Course Topic: Self-Supervised Navigation System for AI Robotics System*

*Abstract:*

In structured environments, robots can navigate simply by following lane markings or pre-defined routes. However, in unstructured environments associated with construction, mining, agriculture, forestry, rural delivery and disaster sites, robots need to have a deeper comprehension of the semantics of a scene in order to understand the movability, affordances, and hazard-level of different objects. Unfortunately, most mobile robot systems use purely supervised learning which requires a time-consuming annotation process. To address this problem, this project aims to investigate self-supervised navigation algorithms with applications for field robots in unstructured environments. First, we will use stereo matching algorithms to create point clouds from input stereo images. Next, we will use SLAM algorithms for legged robots to estimate the robot pose and fuse the acquired point clouds. Finally, we will use self-supervised learning to automatically label traversable areas based on the trajectory travelled by the robot. Experiments will be carried out using a Unitree Go1 quadruped robot.

*Introduction:*

In contemporary times, robots are an integral part of the human life as they aid the functionalities often carried out by humans to make the life of humans easier. Robotic systems are driven by certain Artificial Intelligence (AI) algorithms. Based on the task the robot is performing, the algorithms are divided into sub-categories. For instance, perception algorithms are used to gather and process the sensory information from the various sensors that are aligned onto the robot and planning / navigation algorithms are used to probe the robotic system to move from point A to point B. Now, the robotic system should plan to navigate in a way so that it can successfully defend the obstacles both on-road and off-road and reach to the goal point without any significant amount of damages. In our DIS project study, we explore various Simultaneous Localization and Mapping algorithms for planning and navigation purposes to be carried out for the Unitree Go 1 quadruped robot.

*Experiment #1: VINS Fusion – An optimization-based multi-sensor state estimator*

VINS-Fusion is an optimization-based multi-sensor state estimator package, which achieves accurate self-localization for autonomous applications used in drones, cars, and AR/VR. I worked on the VINS-Fusion, which supports stereo cameras only [1]. The installation of the package involved installation of Ceres Solver [2] as a prerequisite. Unfortunately, I could not run the Ceres solver in my Linux machine as it prompted with certain errors during installation related with my GPU setting. The problem was I could not run the Cuda platform on my machine as I did not have the dedicated NVIDIA GPUs to run the same. Hence, we moved onto installing StereoVision package, another dedicated package for stereo matching discussed below.

*Experiment #2 : StereoVision - Producing 3D point clouds with a stereo camera in OpenCV*

Running this package [3], involves calibrating the stereo cameras firstly. Stereo camera is a type of camera which involves two or more lenses which capture the same image left and right so that it can be better visualized in three-dimensional space. The package involved using the library OpenCV. The author of this package did change the OpenCV’s stereoRectify function to get better point clouds. Point clouds are a collection of points in 3D space which is captured by a sensor. For instance, here the author tried to visualize the points in 3D space using a stereo camera setup. However, this package also did not function properly in my machine so I moved to Spherical Transformer for LiDAR-based 3D Recognition.

*Experiment #3 : Spherical Transformer (SphereFormer) for LiDAR-based 3D Recognition (CVPR 2023)*

SphereFormer [4] is a plug-and-play transformer module. It is used for generating point clouds in 3D space. SphereFormer involves implementation of PyTorch and implementation of CUDA environment. The reason why the package did not run on my machine is lack of NVIDIA GPUs to support the CUDA environment. Although, I tried to prune around and bypass the CUDA environment, I was unable to elicit meaningful results for the dataset [5].

*Experiment #4 : AANet StereoMatching ROS*

AANet [6] is a ROS package used for stereo matching and point cloud projection. Stereo matching involves matching the left (L) and right (R) pair of stereo images so that it can be used for point cloud generation. The step 2 of the project involves point cloud projection which involved creating 3D point clouds from the fused image produced after stereo matching step. The process involved activating Conda environment. I did work on the DrivingStereo dataset [7]. However, the installation did not work out as expected and no meaningful results could be elicited from this experiment. Hence, I moved to HDL Graph Slam method.

*Experiment #5 : HDL Graph SLAM*

hdl\_graph\_slam [9] is an open source ROS package for real-time 6DOF SLAM using a 3D LIDAR. It is based on 3D Graph SLAM with NDT scan matching-based odometry estimation and loop detection. It also supports several graph constraints, such as GPS, IMU acceleration (gravity vector), IMU orientation (magnetic sensor), and floor plane (detected in a point cloud). The HDL Graph SLAM requires the following libraries: OpenMP, PCL, g2o, and suitesparse. The following ROS packages are required: geodesy,nmea\_msgs, pcl\_ros, ndt\_omp, and fast\_gicp. We used this package to extract 3D point clouds and finally odometry information from the HILTI dataset [10]. This package did successfully run in my machine and I could successfully elicit the timestamps, position coordinates (x,y,z) and orientation coordinates (qx,qy,qz,qw). Until now, I have extracted the 6DOF file for hdl\_501\_filtered.bag file only. I plan to iterate the process and extract the 6DOF files for HILTI SLAM Challenge 2023 datasets. The 6DOF files thus generated could be fit to train on the machine learning prediction model for the robotic systems. The steps followed to achieve a 6DOF file are illustrated in [Fig. 2-7]. A snippt of the generated 6DOF file is illustrated in [Fig. 1].

A white background with black and white clouds

Description automatically generated with medium confidence

Fig. 1: Figure showing 6DOF file with the robot timestamps, position and orientation coordinates

A screenshot of a computer

Description automatically generated

Fig. 2: Roscore service running

A screen shot of a computer screen

Description automatically generated

Fig. 3: Roslaunch command running and executing the hdl\_graph\_slam501 launch file

A computer screen with white text

Description automatically generated

Fig. 4: Rosbag play command running

A line of colorful dots

Description automatically generated with medium confidence

Fig. 5: Rostopic echo /odom command running

A screenshot of a computer

Description automatically generated

Fig. 6: Rviz command running

A computer screen shot of a computer screen

Description automatically generated

Fig. 7: RVIZ window running

References:

1. <https://github.com/HKUST-Aerial-Robotics/VINS-Fusion>
2. <http://ceres-solver.org/installation.html>
3. <https://erget.wordpress.com/2014/04/27/producing-3d-point-clouds-with-a-stereo-camera-in-opencv/>
4. <https://github.com/dvlab-research/SphereFormer#spherical-transformer-for-lidar-based-3d-recognition-cvpr-2023>
5. <https://sites.google.com/inha.edu/diter/home?authuser=0>
6. <https://github.com/xmba15/aanet_stereo_matching_ros>
7. <https://drivingstereo-dataset.github.io/>
8. <https://arxiv.org/abs/2209.06522>
9. <https://github.com/koide3/hdl_graph_slam>
10. <https://hilti-challenge.com/dataset-2022.html>