Making Spot® Your Travel Buddy

1. Introduction

1.1. "Who's Spot®, really?"

Boston Dynamics' Spot® is a marvel of bioinspired and biomimetic robotics. It can work in complex (although not very complex and cluttered) places along with humans. It is highly customisable: you can attach pretty much any kind of sensor to it and gather data all day round. It has already been used for making digital twins for maintenance, thermal imaging, tunnel surveying, leakage detection, crack detection etc.

One obvious application for a dog-like robot that is missing is as a 'companion'. Dogs are very loyal animals and respect their owner. Particularly for the elderly and lonely, it is used as a companion that not only understands and protects their owner but also executes their commands. E.g. bringing newspaper, attending the door etc.

1.2. The Problem

The first problem that springs up when we think of using a robot to be used as a companion, is that it has to accompany the owner wherever they go. E.g. if staying in a hotel, visiting friend's place, in a restaurant, and so on. For every new place, it has to figure out its position confidently and quickly. This is challenging in indoor environments due to following reasons:

a) there are no known landmarks
b) the place can be cluttered (or lots of objects)
c) no GPS data
d) frequent environment change (reordering, moving to an unfamiliar place etc.)
e) lack of large enough dataset for training

and many more. So it is a challenge worth addressing which I take up in this project.

2. Background

2.1. Early SLAM

Early implementation started as a feature-based pose tracking system that would make use of 3D geometry and epipolar constraints to learn camera parameters. Then some optimisation algorithms (like bundle adjustment) would reduce the error criteria.

This method required feature extraction that would consume a lot of computation and was generally known as the bottleneck for real-time applications.

2.2. Advanced SLAM

Using lasers, GPS, and LIDARs for generating point clouds of the environment led to very fast algorithms that have been used for outdoor navigation, most famously used in self-driving cars. In indoor environments, these sensors can easily map the surroundings, although localization is challenging because of reasons discussed in previous section.

New algorithms and new hardware have shown to improve indoor localization. ORB-SLAM [1] and KinectFusion [2] are some of the recent examples.

3. Link to code I developed

The repository of the project can be accessed at this link.

4. Dataset

4.1. TUM Freiberg Dataset

This is a large dataset containing RGB-D data and ground-truth data with the goal to establish a novel benchmark for the evaluation of visual odometry and visual SLAM systems. The dataset contains the colour and depth images of a Microsoft Kinect sensor along the ground-truth trajectory of the sensor. The data was recorded at full frame rate (30 Hz) and sensor resolution (640x480). The ground-truth trajectory was obtained from a high-accuracy motion-capture system with eight high-speed tracking cameras (100 Hz). Further, it provides the accelerometer data from the Kinect.

4.2. Visualisation

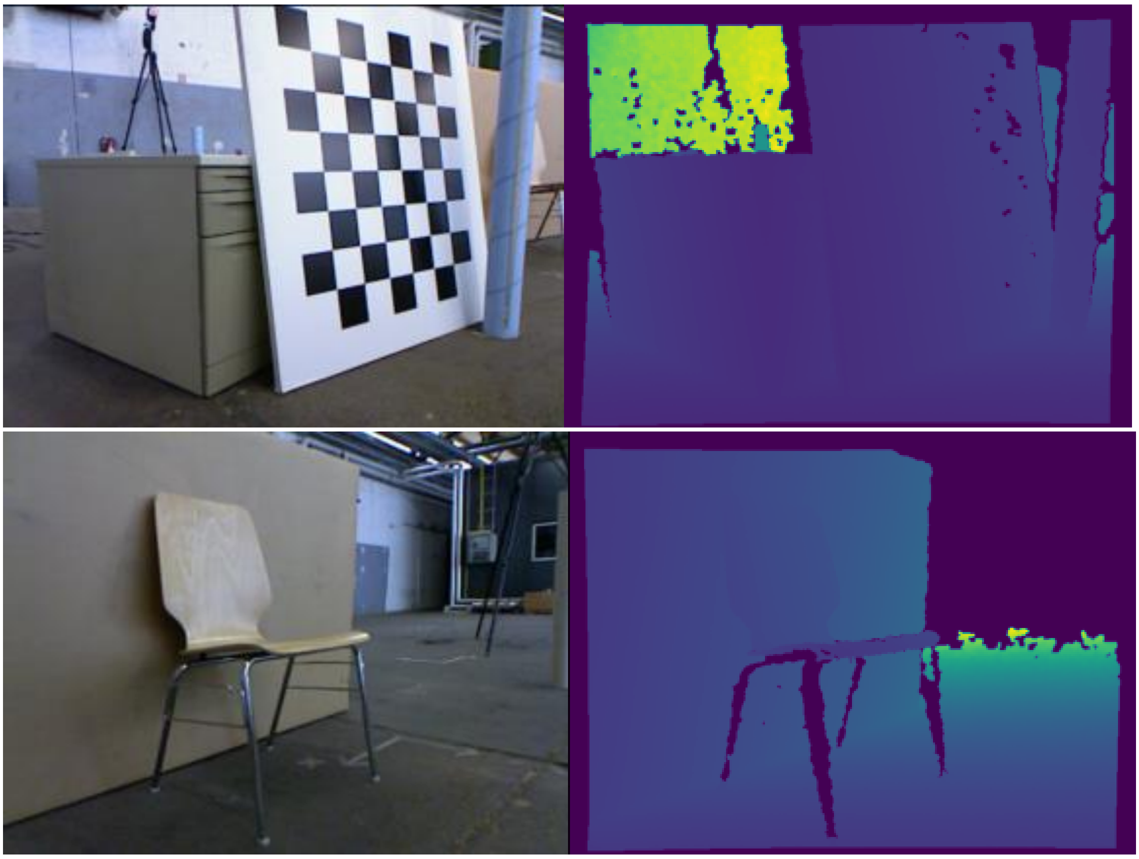

Figure 1 Collage showing sample images (left) and co-responding depth maps (right)

5. Experiments, relevance and important implementations

I decided to take the method discussed in [3] that has shown to break all previous benchmarks and make some modifications in order to understand the choice of architecture.

- Removing the optical flow inputs
- Changing the prediction from SE(3), depth pair to SO(3), displacement
- Simpler architecture and more easier to run
- Replace the final estimation model with CNN, RNN, and GRU

6. Results and significance. Did the project pan out the way you expected it or not? Why?

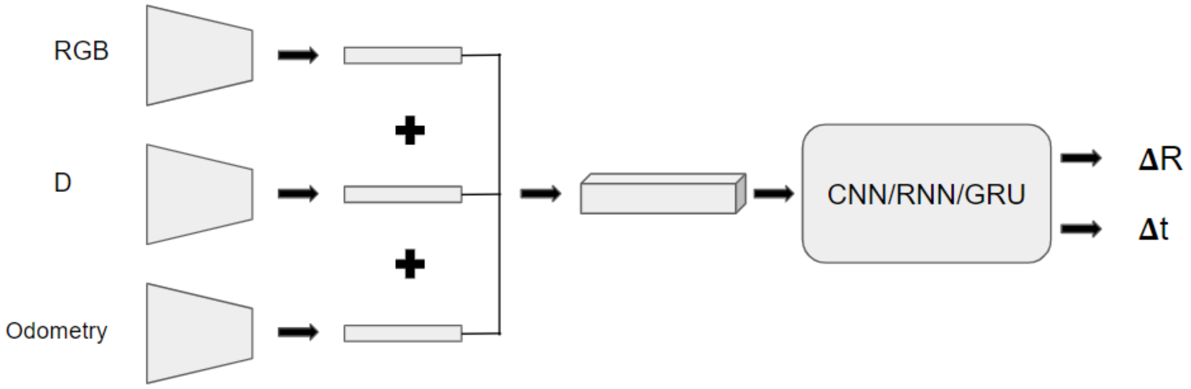

Figure 2 Suggested architecture

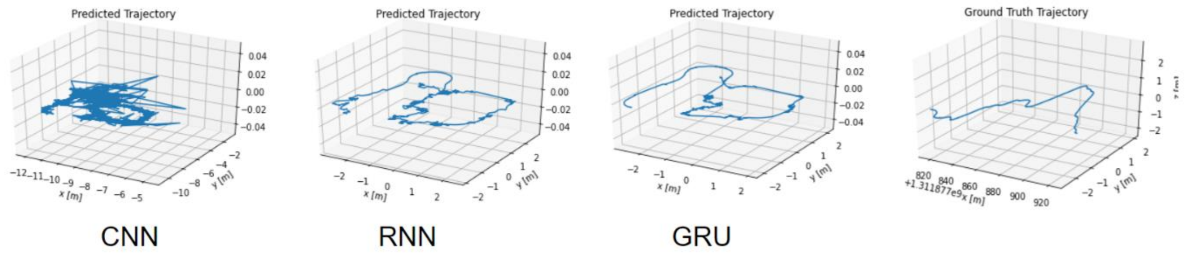

Figure 3 Results with different types of estimation networks

It can be seen that:

- The original implementation made use of ConvGRU i.e. CNN layers in place of MLP layers in the usual GRU. The current implementation uses a normal GRU.
- Simple CNN does not work as it gives a cluttered output without any clear sense of direction and step size
- Using a residual network helps in removing the clutter although it still has problem especially around sharp corners
- In the case of GRU, the time dependency improves the predictions
- In all 3 cases, the predictions do not match ground truth trajectory. The main reason is the removal of optical flow structure that is provided as input to the prediction network

7. Challenges faced in brief
   7.1. Extent of Work

The main obstacle I faced was about work distribution. Due to severe lack of support in the form of teammates, it became very difficult to keep up with other teams with 3 or more members. Another point I want to take up, is that, I also did not have any already going on research project to pivot as the class project. Nevertheless, I am confident that I have done at least twice as much work as a single person can possibly do given that I had 3 other courses (not on research track).

7.2. Industry Standards

It appears that the state-of-the-art algorithms in research and in industry is done in C or C++. Majority of recent work that is being in done in Python is concentrated towards mixing of machine learning with traditional implementations.

8. Summary and future work

- Optical flow helps in getting correct estimates
- The pose mapping on SE(3) puts more constraints on the outputs than SO(3)
- Depth estimation can be done using better models such as transformers
- Alternatives for optical flow can be searched to reduce computation

References

[1] Mur-Artal, R. and Tardós, J.D., 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, *33*(5), pp.1255-1262.

[2] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A. and Fitzgibbon, A., 2011, October. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 559-568).

[3] Teed, Z. and Deng, J., 2021. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in Neural Information Processing Systems*, *34*, pp.16558-16569.