Mid-Project Report Template

A Mid-Project Report Prepared for the
Final Project of the graduate Course
CS401: Intelligent Robots

May 25, 2018

Prepared by:

任涛
李可明
王森
何思阳
陈德缘
艾丹迪

1.1 Abstract

This document is to write the proposal for the CS401 final project. It includes six components: (1) Background and Motivation; (2) Hardware and Software; (3) Related work(4) Research Goals and Objectives; (5) System Introduction; (6) Expected Novelizes; (7) Staffing Plan; (8) Timeline; and (9) Reference.

2.1 Background and Motivation

In Robotic Mapping, there are several types of maps, metric map, topological map and semantic map. The algorithm we used is SLAM, which proposed by Smith and Cheeseman in 1988.In robotic mapping and navigation, simultaneous localization and mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it.There are various versions of SLAM, which are 2D, 3D and SLAM with semantic. The robotic mapping is widely used in manufacturing, rescue, national defense and other fields . The goal of our research is generating the 3D map by open-source tools and updating their performance by deep learning or machine learning methods such as Stacked Denoising Autoencoders.

2.2 Hardware and Software

Hardware platform:

1. *Running machine*

Servers from lab of Hisao Ishibuchi (if one of them has several free cores, there are 4 servers) For each server: 32-core CPU， 128G memory， 2T disk， 1 NVIDIA P4000GPU

2. *Turtlebot3*

“TurtleBot is a most popular ROS standard platform robot among developers and students teaching computer programming language using Logo.[1] The TurtleBot can run SLAM(simultaneous localization and mapping) algorithms so as to build a map and complete other features such as object recognition, tracing running and others.

3. *RGB-D*

Software platform:

1. OctorMap

OctorMap uses an octree to store the map. Each parent is a big cube and eight children of it represent the eight same-sized sub cubes in 3-demension space, when there is sparse there is no need to store a child and if we want a more accurate map we just need to increase the maximum depth of tree.

We choose it because of its flexible, full 3D model, updatable (add or remove element anytime we want) and compact.[2]

2. Stacked denoising auto-encoder (SDA)

The Stacked Denoising Autoencoder (SDA) is introduced in [3]. This would be chosen as one of the most important deep learning methods we would use.

3. RGB-D SLAM

4. *Tree-based Network Optimizer (TORO) [4]*

5. OpenCV

OpenCV (Open Source Computer Vision Library) which is with computational efficiency and focusing on real-time applications. [5]

6. *Input data*：

tum open dataset [6], Fab map dataset

2.3 Related Work

A pioneering work in SLAM is conducted by R.C. Smith and P. Cheeseman on the "representation and estimation of spatial uncertainty" in 1986. And in the early 1990s, the research group of Hugh F. Durrant-Whyte equip the vehicle with multiple servo-mounted sonar sensors to provide a way in which a subset of environment features can be precisely learned from the robot's initial location and subsequently tracked to provide precise positioning. This finding motivated the implementation of algorithms which are computationally tractable and approximate the solution. Now there are several ways to represent maps in robotic: grid map, appearance-based methods, etc.

SLAM will always use several different types of sensors, and the powers and limits of various sensor types have been a major driver of new algorithms. As for mass-market SLAM implementations, nowadays robot vacuum cleaners are good examples. What's more, self-driving cars by Google and others have now received licenses to drive on public roads in some US states.

2.4 Research Goals and Objectives

In robot mapping we will face a lot of complex problems, especially, positioning and mapping as two coupling, their estimate error will affect each other, and our subject itself is "robot mapping", we hope to:

1.Realize deep learning to optimize loopback detection

2.complete a relatively accurate self-learning mapping robot on the basis of existing datasets

2.5 System Introduction

The whole system is divided by four parts, the first part is the sensor information reading, that is to read image information from camera and perform preprocessing, the second part is visual odometer VO(also known as front end): Estimates the motion of the camera between adjacent images, as well as the appearance of a partial map. The next step is back-end optimization: it accepts camera poses measured by visual odometers at different times, as well as loopback detection information, optimize them to obtain globally consistent trajectories and maps. The last step is mapping, which is based on the estimated trajectory, a map corresponding to the mission requirements is established. Also, there is a loopback detection: it is used to determine if the robot has reached the previous position, if a loopback is detected, it will provide the information to the backend for processing.

The main purpose is that it concerns the object motion between adjacent images. The backend mainly refers to the problem of dealing with noise in the SLAM process. Consider: 1 How to estimate the state of the entire system from these noisy data. 2 How much uncertainty is estimated in this state(maximum a posteriori probability estimate) where the state includes the trajectory of the robot itself, as well as the map. As for the loopback detection, the information about the same point of A and B is told to the backend optimization algorithm, and the backend adjusts the trajectory and the map according to the information to match the loopback detection result. In this way, if there is sufficient and correct loop detection, the accumulated error can be eliminated, and a globally consistent trajectory and map can be obtained.

As for the mapping, which is the most important part. There are several parts of mapping. We use metric maps as it emphasize the precise representation of the positional relationship of objects in a map, usually classified by sparse and dense. Sparse maps are abstracted and do not need to express all objects. Choose a part of the road that is representative of the meaning. A map made up of road signs, not part of the road sign, can be ignored. While dense maps focus on modeling al objects that are seen, and dense maps are required for navigation. Dense maps are often composed of many small blocks at a certain resolution. For a two-digit metric map, there are many small squares, and three-dimensional is a small square. When querying a spatial location, the map can give information as to whethere the location can pass.

2.6 Expected Novelties

2.6.1 The feature selection of RGB-D cameras.

The traditional ICP algorithm has a large number of pairs of mismatched points, and the algorithm is easy to fall into local optimum. In traditional ICP algorithm, the spatial position of the feature points is calculated from the data provided by the depth camera. However, many visual points are based on the difference between the key points and the surrounding pixel points, so the acquired features tend to fall on the edge of the real object.

Depth cameras, based on the principles of structured light reflection or time-of-flight cameras, result in extremely unstable depth data measured at the edge of the object, creating difficulties for the RGB-D SLAM system.

In order to reduce the estimation error, a method of extracting feature points with reliable depth is proposed, which is called a plane feature point. It can be seen from the analysis that the points on the surface of the object can correctly reflect the structured light of the depth camera, so their positions have higher stability. However, they are not visually obvious compared to the points at the edges of the object. Traditional visual SLAM methods may not detect them.

Therefore, using the inherent characteristics of the depth camera, the Random Sample Consensus (RANSAC) model can be used to estimate the plane from the point cloud acquired by the depth camera, recover and enhance the image on the plane, and then extract features on these images.

2.6.2 Apply deep learning to loopback detection.

As the sensor changes, more features need to be designed to fit different types of sensors. And we hope to learn the feature description directly from the source image by learning the algorithm. That is, let the machine understand which features are in the image, and compare the similarities of the images according to the similarity of the features to complete the loopback detection.

Deep learning techniques offer possibilities for this approach, but in the field of SLAM research, it is far from being used in real-world applications. If we can express RGB-D data in a deep network, we can identify the image more easily.

2.7 Staffing Plan

任涛: Realize OctoMap

王森：Realize deep learning

李可明：Loopback detection

陈德缘：SDA

艾丹迪：Configuration environment

何思阳：Test

2.8 Timeline

3.19: proposal presentation

4.10 configuration environment

4.30 realize OctoMap

5.20 built figure optimization

5.30 final report

2.9 Reference

[1]. OctoMap: An efficient probabilistic 3D mapping framework based on octrees, Hornung, Armin and Wurm, Kai M and Bennewitz, Maren and Stachniss, Cyrill and Burgard, Wolfram, Autonomous Robots, 2013.

[2]. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems, Wurm, Kai M and Hornung, Armin and Bennewitz, Maren and Stachniss, Cyrill and Burgard, Wolfram, ICRA 2010.

[3]. 基于 ROS 轮式机器人自主融合探索建图与路径规划, 孙旭东, 06, 2017.

[4]. Gao, X., & Zhang, T. (2015). Loop closure detection for visual slam systems using deep neural networks. In The 34th Chinese control conference, (Hangzhou, Zhejiang Province), technical committee on control theory (TCCT) of Chinese Association of Automation (CAA). Accepted July 2015.

[5]. Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of rgb-d SLAM systems. In 2012 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 573–580), IEEE.

[6] emanual.robotis.com/docs/en/platform/turtlebot3/

[7] https://octomap.github.io/

[8] P. Vincent, H. Larochelle Y. Bengio and P.A. Manzagol, Extracting and Composing Robust Features with Denoising Autoencoders, Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML ‘08), pages 1096 - 1103, ACM, 2008.

[9] Grisetti G, Stachniss C, Grzonka S, et al. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. Robotics: Science and Systems, 2007

[10] https://opencv.org/

[11] Gao, X., & Zhang, T. (2015). Loop closure detection for visual slam systems using deep neural networks. In The 34th Chinese control conference, (Hangzhou, Zhejiang Province), technical committee on control theory (TCCT) of Chinese Association of Automation (CAA). Accepted July 2015.