Autonomous Navigation of a Quadrotor in an Indoor Environment using Deep Learning

Shanmuga Perumal Harikumar,Master of Science in Robotics Engineering,WPI

***Abstract*—The purpose of this project is to develop an autonomous quadrotor system that could navigate an indoor environment without running into obstacles. This documents outlines the project in which the author wishes to train a deep neural network to generate motion commands based on the input image. The sensor data is directly mapped to the control input to the system by the deep learning network. An initial problem statement is formulated. The final project goals are enumerated. Previous works in the field relating to the project undertaken are explained in brief. The various hardware and software components to be used in the project are listed,along with a short description of each. Finally the proposed project logistics is outlined.**

***Keywords*—Deep learning, Autonomous Navigation, Convolutional Neural Networks, Quadrotor Indoor Navigation**

I. Introduction

The purpose of this project is to make a quadrotor learn navigate through an indoor environment autonomously using deep learning. In other words, the author hopes to map sensor input directly to motion commands.The quadrotor must learn to detect and avoid obstacles in its path. The application of such a learning system is immense. The quadrotor market is currently largely focused on outdoor quadrotors that rely heavily on GPS for positioning and way-points navigation. There is however a dire need for quadrotors that can operate in an indoor environment. There are several problems that need to be addressed in order for a quadrotor to successfully fly indoors. The problem of navigating through a cluttered indoor space such as a warehouse or an office is yet to be solved. This project aims at using deep learning to solve this problem.

Convolutional Neural Networks(CNN) have been used in object recognition with high success rates.Before CNNs object recognition was done by using hand crafted features which required domain experts and required special features to be developed for each specific application which was laborious. The development of Deep Neural Networks(DNN) led to the process of classification becoming more general.

A. Problem Statement

In this project the author hopes to train a deep neural network to detect obstacles and navigate in an indoor environment avoiding the detected obstacles. The inputs to the network will be image and depth information from a RGBD camera(Intel Real Sense) along with IMU data as labels. The learned network parameters will be able to get a RGBD image and give out IMU data as output. This velocity data will then be used to control the quadrotor.

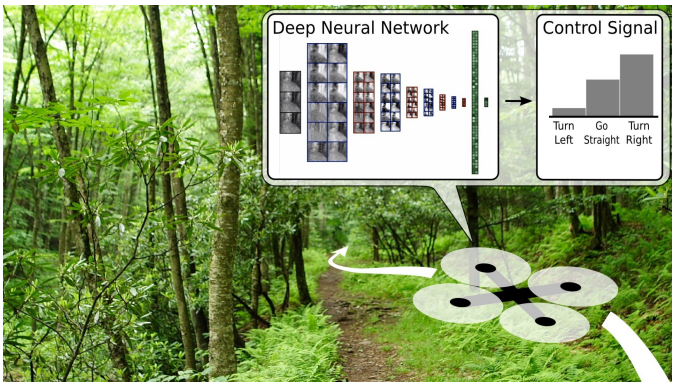

Fig. 1. The picture showing the motion commands that is generated by the network from the input image

B. Project Goals

The project goals are as follows

- Create a data set that includes RGBD camera data also IMU data large enough to train a deep neural network
- Use the generated dataset to train the deep neural network to perform autonomous navigation
- Test the network with test dataset
- Use the network to actually navigate a quadrotor in an indoor environment

II. Previous Work

There have been several previous attempts at using Deep Neural Networks for navigation of a robot.[1] used DNN to navigate a forest trail autonomously. The training data was obtained by using three cameras mounted on th helmet of a hiker. One of the cameras was pointing straight,one towards the left and one towards the right. The hiker was made to follow the trail while always turning towards the direction the trail was heading in.The cameras pointing towards the left always had the trail towards its right and the camera pointing towards right always had the trail to its left.The images were also flipped and the dataset was augmented.The problem was treated as a classification problem. The final network took an image as a input and gave the direction towards which it thought the trail was in as output. The final output was something like a probability. The difference between the probability of turn left and turn right commands was used as the yaw command and the probability of going straight was used as the velocity.

[2] uses a somewhat similar approach to control a car using an input image. The input from the sensor is mapped to control

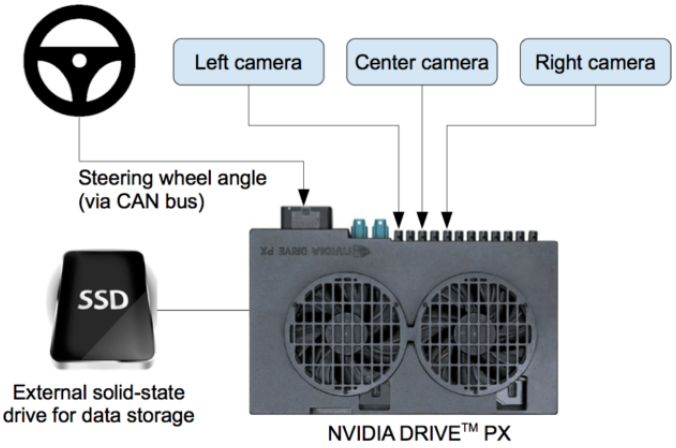

Fig. 2. System Architecture of the Autonomous car

commands for the vehicles to follow. The data acqisition setup has three cameras ,one pointing straight and the other two at a fixed angular offset on either sides. The dataset was further augmented by distorting the input images by applying fixed transformations. The data was obtained from various roads under various weather conditions,to ensure good learning and generality. The learned parameters were first tested on the simulator and subsequently on a real car.The authors claim that the car was autonomous 98% of the time. This shows that it is feasible to map the sensor inputs directly to control outputs

III. Hardware and Software

This section explains about the various hardware and software components to be used in the project.

A. Theano

Theano is an opensource python Library that aids with numerical computation. It is closely linked with Numpy. The library is optimized to run in CPU or GPU ,making it ideal for deep learning applications.

B. Intel Realsense R200

Real sense R200 is a RGBD sensor from Intel that is both small and light weight making it one of the first choices to be mounted on a drone where weight is a constraint.It has an IR camera with a resolution of 640x480 capable of recording at 60 fps. It also holds a 1080p RGB camera capable of recording at 30 fps.

C. DJI Matrice 100 with Guidance sensor package

Matrice 100 is a developer quadrotor from DJI that has a flight controller ,GPS , propulsion management systems already installed. It is a very stable platform that has very robust controllers able to compensate of any kind of external disturbances. It also has a guidance sensor package attached to it. Guidance package consists of the guidance core computer ,five stereo pairs and five ultrasonic sensors, one pointing in each direction and one pointing down, and also an IMU.

Fig. 3. DJI Matrice 100 Developer drone with guidance installed

D. Robot Operating System(ROS)

Robot Operating System(ROS) is a software framework for development of software for robots specifically. It is an opensource implementation and has a very active community. Using ROS simplifies communication between the various parts of the system and help visualize the entire system better. There are hundreds of packages available in ROS ranging from perception to motor drivers.

E. DJI Onboard SDK and Guidance SDK ROS

DJI has a Software Development Kit for both Guidance sensor package and Onboard Controller. Both these packages have ROS implementations which make them much easier to use.The guidance package from DJI publishes the monochrome images from the stereo pair ,the obstacle distance information and IMU data which can now be used by other ROS packages. The Onboard SDK lets us control the aircraft. Since all the roll pitch and yaw controllers are already built in,the developer can focus on the specific implementations instead of flying the aircraft stably

IV. Logistics

A. Current Progress

- Conduct literature review on various approaches to navigation using machine learning
- Get the hardware ready

B. Expected Goals

- Record a dataset of RGBD data from the real sense and the IMU
- Decide on the structure of the Deep Network
- Learn Theano and start the coding process
- Train the network with the data obtained
- Test the Network with the test set
- Test the Network on unknown indoor environment
- Make the test dataset opensource

C. Reach Goals

- Implement the algorithm on the actual drone and fly the drone autonomously

REFERENCES

[1] Alessandro Giusti,Jerome Guzzi,Dan C. Cirezan,Fang Lin HE, *A Machine learning Approach to Visual Perception of Forest Trails for Mobile Robots*, IEEE Robotics and Automation Letters,Nov 2016

[2] Mariusz Bojarski,David Del Testa, Daniel Dworakowski Bernhard Firner, *End to End Learning for Self Driving Cars*, Nvidia Corporation

[3] DJI website, *www.DJI.com*,DJI 2016

[4] Deep learning, *http://deeplearning.net/software/theano/*

[5] Intel real sense, *https://software.intel.com/en-us/realsense/devkit*