Drone Autopilot System for Vehicular Network Using VLC

Sabit Shaiholla, Asset Bekov, Supervisor: Professor Refik C. Kizilirmak

Department of Electrical and Computing Engineering, School of Engineering, Nazarbayev University, Astana, Kazakhstan, 2018

Abstract

This project delegates the autopilot system for drones with object recognition using OpenCV for vehicular network with visible light communication (VLC). The main aim of this project is to model the autopilot system based on computer vision for drone and simulate the model on VLC-based vehicular network which uses drone as a base station. The underlying objective is to model the autopilot system to follow the vehicles using on-board camera and to calculate the distances and finally find the spot on which bit rate for vehicular network is the highest.

System Overview

In the depiction of current state of vehicular networks the increase of safety and convenience of road-users is directly related to reliability and authenticity of data transferred through vehicular network [1].

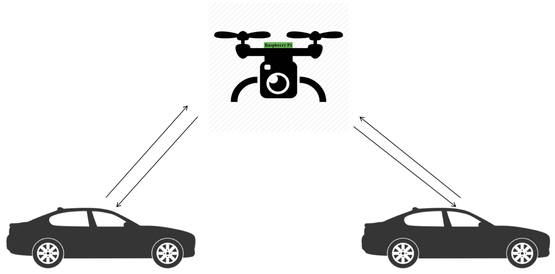

Figure 1: Overview of the system

- Using camera and OpenCV library, the needed vehicle cluster will be detected
- Next, computer algorithm will calculate the distance between drone's current location and location of vehicles
- Consequently, based on predefined algorithm the optimum location for data transfer will be calculated and drone will go to that position

System Block Diagram

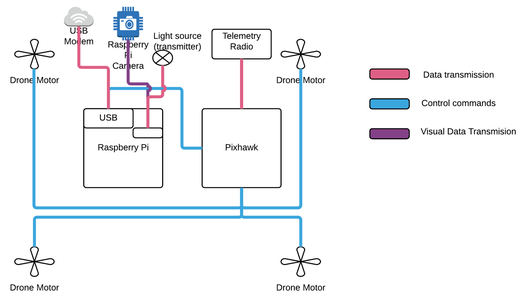

Figure 2: Block diagram of the system

1. **Pixhawk** - flight controller connected directly to drone motors to control drone directions
2. **Raspberry Pi** - single board computer performing autopilot and OpenCV algorithms
3. **Camera** - Raspberry Pi camera for visual data reception and transmission
4. **Telemetry** - transmission of telemetry of the drone to third party applications
5. **USB modem** - to establish connection between on-board computer and ground station
6. **Light source transmitter** - is used for VLC network

Motivation

Usage of drone in VLC network between vehicles allows secure exchange of data as private network can be established between fixed number of cars with the help of UAV. The implementation of this project on industrial level will potentially increase the security of vehicular networks and will service the inquiries of data on road traffic.

Software Implementation

- **Python** - high-level open source language, supports vast array of libraries, comes pre-installed on Raspbian OS
- **Dronekit** - application programming interface which allows to create apps that run on an on-board computer and communicate with the Pixhawk flight controller using a low-latency link
- **OpenCV** - open-source vision and communication library that allows for image processing and manipulation through the use of Python
- **MAVLink** - protocol for communicating with small unmanned vehicle, mostly used for communication between a ground stations and drone, and in the inter-communication of the subsystem of the drone
- **Ground Control Station** - software used to track the drone's states using telemetry

Simulation

- Mission planner software allow to simulate the commands without actual hardware
- Simulation of the UAV going to specific points and following specific commands

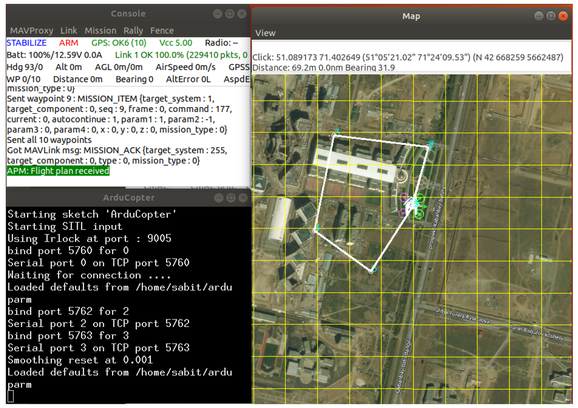

Figure 3: Simulation of the drone

Autopilot System

```
vehicle = connect(connection_string, wait_ready=True)
vehicle.mode = VehicleMode("GUIDED")
vehicle.armed = True
print("Going towards destination point ...")
point = LocationGlobalRelative(55.361354, 71.165218, 20)
vehicle.simple_goto(point, groundspeed=10)
print("Returning to Launch")
vehicle.mode = VehicleMode("RTL")
vehicle.close()
```

Figure 4: Main codes for drone commands

```
cv2.VideoCapture(0)
bcapWebcam.isopened()
capWebcam.read()
gray = cv2.cvtColor(imgOriginal, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (7, 7), 0)
edged = cv2.Canny(blurred, 50, 150)
(cnts, hierarchy) = cv2.findContours(edged.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
```

Figure 5: Main codes for OpenCV object recognition

Future Work

- System implementation on the hardware level with the software developed
- Connection of OpenCV with the autopilot algorithm
- Implement the algorithm calculating the spot with the highest bit rate based on the distance between the current location of drone and location of vehicle

Reference List

[1] Bryan, Parno, and Adrian, Perrig. "Challenges in Securing Vehicular Networks." Workshop on Hot Topics in Networks (HotNets-IV), 2005, 1-6.

[2] Alsalam, Bilal Hazim Younus and Morton, Kye and Campbell, Duncan and Gonzalez, Felipe, "Autonomous UAV with vision based on-board decision making for remote sensing and precision agriculture", Aerospace Conference, 1-12, 2017 IEEE

[3] Ward, Sean and Hensler, Jordon and Alsalam, Bilal and Gonzalez, Luis Felipe, "Autonomous UAVs wildlife detection using thermal imaging, predictive navigation and computer vision", Proceedings of the 2016 IEEE Aerospace Conference, 1-8, IEEE, 2016