भारतीय प्रौद्योगिकी संस्थान जम्मू
INDIAN INSTITUTE OF TECHNOLOGY JAMMU
विद्यायनं सर्वधन प्रधानम्

D.R.D.O. SASE's UAV Fleet Challenge

MID-EVALUATION

Institute- IIT Jammu

Team Members:-

1. **Kshitij Garg(Leader)**
2. **Kshitij Aggarwal**
3. **Chinmay Kurade**
4. **Ajay Kushwaha**
5. **Kshitij Singh**
6. **Aman Patkar**
7. **Tushar Goyal**

1. Introduction

Nowadays, UAVs are attracting a lot of attention from industrial and research fields. They are suited to a large number of applications, thus making them of interest for commercial and research purposes.A challenging upcoming scenario for UAVs is the use of swarms, or Flying Ad-Hoc Networks (FANETs): several drones, from tens to hundreds, jointly used in order to execute a given task. The joint use of multiple UAVs poses several challenges that must be met. Several advantages can be brought by the use of multiple UAVs in those scenarios: for instance, it is likely that the overall cost of acquisition and maintenance of several small UAVs is lower than the overall cost of a single large UAV needed for the same task. Furthermore, fault-tolerance is inherently provided by the use of swarms, because a single drone can be removed with a limited impact on the overall formation. Swarms can also provide scalability, i.e., adding or removing drones from a swarm, in order to better adapt to changing conditions or to simply replace one or more UAVs experiencing issues or battery depletion.

What do we need?

1. OpenMV- M7 Machine Vision system
2. RaspberryPi Zero
3. USB to Serial (FTDI) adapter and cables
4. DroneKit Python

OpenMV

The OpenMV Cam M7 is powered by the 216 MHz ARM Cortex M7 processor, enabling 640x480 grayscale images/video (up to 320x240 for RGB565 still) with a camera on board that you program in Python. They make it easy to run machine vision algorithms on what the OpenMV Cam sees so you can track colors, detect faces, and more in seconds and then control I/O pins in the real-world.

Raspberry Pi Zero

We are using the basic Pi Zero on this system, we don't need additional connectivity except for the USB to Serial adapter. The RPI Zero is running with the standard Graphical User Raspian OS with the Python DroneKit installed. We need to get the console disabled in order to access the serial ports, generally, it is done by editing the /boot/cmdline.txt file and enabling the serial port.

2. Image understanding and recognition

Many applications of computer vision entered everyone's life, e.g. face detection software embedded in digital cameras or the Optical character recognition the definition (OCR) software for standard scanners;

The objective of enabling a UAV to perform very complex tasks such as object detection, recognition, and analysis in an unknown environment requires very fast and robust algorithms and there are still no standard approaches in the literature. Moreover, fast and robust methods of image analysis are intended to be automatic and specifically designed for the collaborative setting of a swarm of
UAVs.

Even if the collaborative setting poses a number of issues (e.g. regarding the information sharing and processing), it could be seen as a strength,
if a data fusion step is properly implemented on the different data flows: this kind of processing may efficiently increase in quantity and quality of the information extracted by the diverse sensors hosted by the UAVs.

The **object detection** steps are as follows:

1. Capture raw image from the drone's camera
2. Converts image from red, green, blue (RGB) representation to hue, saturation, and value (HSV) representation using the equation below:
   V=max(R,G,B)

S= V - min(R,G,B) , if V != 0 else 0

S= 60(G-B)/S , if V=R
120+60(B-R)/S , if V=G
240+60(R-G)/S , if V=B

3. Filter every pixel with threshold:
   p(x,y) = 1, if Hmin<=H<=Hmax and Smin<=S<=Smax and Vmin<=V<=Vmax
   0, else
   Where Hmin, Hmax, Smin, Smax, Vmin, and Vmax are threshold values that have to be calibrated before
4. Compute the size of the object detected
   size= summation(x=0->width, y=0->height) of p(x,y)
   Detected object is well known as blob. Blob size will be used as a fitness function. If the size of the object bigger than the threshold, then the UAV will track the object.

Object tracking Algorithm:-

We use a proportional integral differential (PID) control as a tracking algorithm. The main idea of object tracking is to maintain the object in the center of UAVs view, by adjusting its position. Suppose, we have object position in the image (X, Y), the center of the image (Xc, Yc), drone's position in the real-world coordinate system(Xr, Yr), then we apply PID control algorithm with steps below

1. Compute error of object position based on the image center
2. Update current position (Xr(t),Yr(t)) based on previous position (Xr(t-1),Yr(t-1)), and last 2 previous error Ex(t-1), Ey(t-1), Ex(t-2), Ey(t-2)
3. Update previous position data and last 2 previous error data for the next iteration

PID control is also used for point to point movement, where the robot is commanded to go to any position target. As correction and feedback, we use the current robot position calculation by odometry.

Particle Swarm Optimization for Object Localization

PSO is used to build swarm robots intelligent. The main principle of the PSO algorithm is each agent moves to find the target based on individual (local) perception and community (global) perception. In PSO, the local base is the location where an agent has maximum fitness function among its history, and the global base is the location with the maximum fitness function of all agents in their history.PSO algorithm needs

information from every drone used as a swarm member. Therefore, the odometry algorithm is required so that the location of robots can be known every time. PixHawk has embedded odometry algorithms for these purposes.

How does it work?

Initially, quadcopter receives signal to start the mission, then the main process of detecting and tracking are performed. The quadcopter enters main of the behavior which is a loop activity. In this looping process there are two parallel processes running simultaneously. First, quadcopters calculate their positions using odometry. Second, quadcopters detect target presence based on their camera sensors. Using a Companion Computer -Raspberry Pi Zero - and Drone Kit Python, we are switching to GUIDED mode so that we are controlling an ArduPilot based Flight Controller - Pixracer - to rotate the copter slowly around while it receives on the serial port the coordinates of a detected object from the OpenMV camera x-y plane. During the search, it keeps track of the 'best' object it saw so after the copter has rotate 360 degrees it sends commands to point the vehicle in the direction of that object and then sends 3D velocity requests, up to 15 times per second to guide the copter towards the top of the target.Object is detected and tracked using the algorithm mentioned above. Upon reaching the target, quadcopter’s bottom camera is used to calculate the dimension of the detected object using pixel per metric ratio. Once it matches the dimension of expected block size,it sends the coordinates of the detected object, the OpenMv camera system is losing track of the object and the python script resume control back to LOITER, making it ready to repeat the mission once we switch back to guided. If all quadcopters find and track all the targets, then mission is accomplished, and all quadcopters will be landed. If quadcopter find unexpected condition (e.g. does not detect target), quadcopter updates local based and global based, by the mean update information from its history (local base),and other quadcopters history (global based). These two components are used as input to determine the next position of this quadcopter using PSO algorithm.