

Weekly Status Report 6

Quadcopter Cameraman

sdmay19-42

October 10 – October 17

Aamid Ahabab (Lead Engineer) & Client

Zhengdao Wang (Team Advisor)

Alex Nicklaus (Lead Test Engineer)

Isaac Holtkamp (Web Manager)

Nate Allen (Report Manager)

Luke Rohl (Meeting Facilitator)

This week's accomplishments

Summary

On the hardware side of things, we were focused on research while waiting for parts to ship in. The team researched real time operating systems integrations and how we can set deadlines for tasks to be completed between the software and hardware. Aamid looked into accessing different modules on the hardware side, namely, the flight controller. A git repo was found that should be translatable to our flight controller so long as they are compatible.

On the software side, code was implemented for encoding faces to be later decoded and identified or recognized. The code was tested and proven to work for the face of 'Dwight Schrute.' Code was also implemented to recognize pedestrians within a frame and note their current locations, previous locations, and predicted next location.

- All
 - Design Document
- Luke
 - Research how to determine a specific user's face w/ amount confidence
 - Starting with Python face_recognition library
 - https://github.com/ageitgey/face_recognition/blob/master/examples/face_recognition_from_webcam_faster.py
 - Will use code there as guidance towards our end goal to upload photos of target and then have drone track them based on its confidence rating.
 - Project Plan v.2
 - Add comments from canvas to v.2 doc
 - Add definitions
- Nate
 - Researched methods for detection of people
 - Implemented a class for detecting people

- Video recorded field samples of people walking to test with the detection and tracking software
- Refactored the people detection code into a single class for easy imports into project
- Wrote logic to distinguish multiple tracked targets from each other and determine which movement is being made by each individual
- Wrote configurable settings to effect the algorithm and logic for tracking targets
- Tweaked the configuration until I found the most effective settings
- Wrote README information for setting up an environment to use our software
- Aamid
 - Researched resources for communication with flight controller and pi
 - Found Github Repository from project that uses a pi instead of a radio transmitter for data to flight controller
 - Uses C/C++
 - <https://github.com/rgw3d/PiQuadcopter>
- Alex
 - Researched Real Time Operating Systems to run on the Pi to handle flight operations
 - Setup and configuration of RTOS uses C
 - Continued work on real time task deadlining
 - Work is on paper but it takes in information from sensor data sheets to build possible workloads and tasks for operations
 - The sonar requires 20 ms to pull a single data point for drone's height
 - The camera tracking data is limited by the frame rate we set the camera to, so we'll get more data from 60 frames per second than 30 frames per second
 - However, how fast we can process a frame for the purposes of image recognition may serve as more of the bottle neck
- Isaac
 - Continued development of the Android app and communications

Planned to accomplish next week

- Luke
 - *Work with Isaach to implement target uploading to be encoded*
 - Created python module to handle communications from Android App
 - Deserialize byte array
 - Convert into JSON Object
 - Analyze command
- Nate
 - Work with Luke to understand his face encodings to see if we can use it to identify any object.
 - If so, then I will implement the use of encoding full bodies of people to identify when determining which movements are being done by which individuals (tracking movement of targets)
 - If not, then I will work on separating threads for video streaming and video analyzing
- Aamid

- Learn how to utilize the flight controller and access the information from the accelerometer and gyroscope
- Look through Github repo
- Alex
 - Continue research on RTOS
 - See if we can implement any other RTOS on the Pi
 - If we are limited to Raspbian see about creating some level of coding to schedule tasks to meet deadlines
 - If possible implement RTOS on Pi
 - Will need an .iso file to flash to the Pi's memory
 - May just start with getting Raspbian on the Pi and running code
 - Dive back into Embedded Systems to reproduce wave forms on the Pi to send to and receive from our sonar sensor
 - Create virtual schedule on Cheddar from theoretical timing diagrams
- Isaac
 - Begin basic communications with the Raspberry Pi and begin testing commands to the quadcopter.
 -

Roadblocks

- Raspberry Pi may not be able to run any OS other than Raspbian
- Walmart was out of reasonably priced drones for prototyping and domain elicitation purposes i.e. fly and most likely thrash

Hours Spend

Team member	Hours This Week	Hours Total
Nate Allen	10	40
Alex Nicklaus	3	26.5
Luke Rohl	4	31
Mir Ahbab	3	23
Isaac	3	24