Search and Aid

Project Proposal | SS24 ECE Capstone

Team C7: **Nina Guo**, Ronit Shah, David Wu

Electrical & Computer ENGINEERING

Use Case

- A search and aid **drone** application that can detect and locate people needing help to **autonomously** drop aid packages
- Manually delivering aid to people in disasters or dangerous remote areas is **inefficient**, **expensive**, and **error-prone**
- ECE Areas: Software Systems, Signals and Systems

Source: https://drones.wfp.org/index.php/activities

Requirement #1

Accurately identify humans in a flat landscape

Motivation

Rescue agencies should be aware of the location of a person to effectively and promptly handle extraction. Accuracy is also needed for package drops

Sub-requirements

- Identification of humans with low computation time (<5 sec) using a distributed server
- Low latency communication between the drone, distributed server, and web-application
- Processing of varying camera angles to improve accuracy of location detection

Requirement #2

Autonomously comb a landscape

Motivation

Humans needing aid must be found in a large landscape by a drone if the location of a human is unknown

Sub-requirements

- Customized autonomous “drone controller” controller to guide the drone
- Varying camera angles of a location to gather detailed information
- Cost effective drone (<$300)

Requirement #3

Report findings through a web application

Motivation

Reporting information like the location of the drone, the location of people, and more through an easy-to-use web application

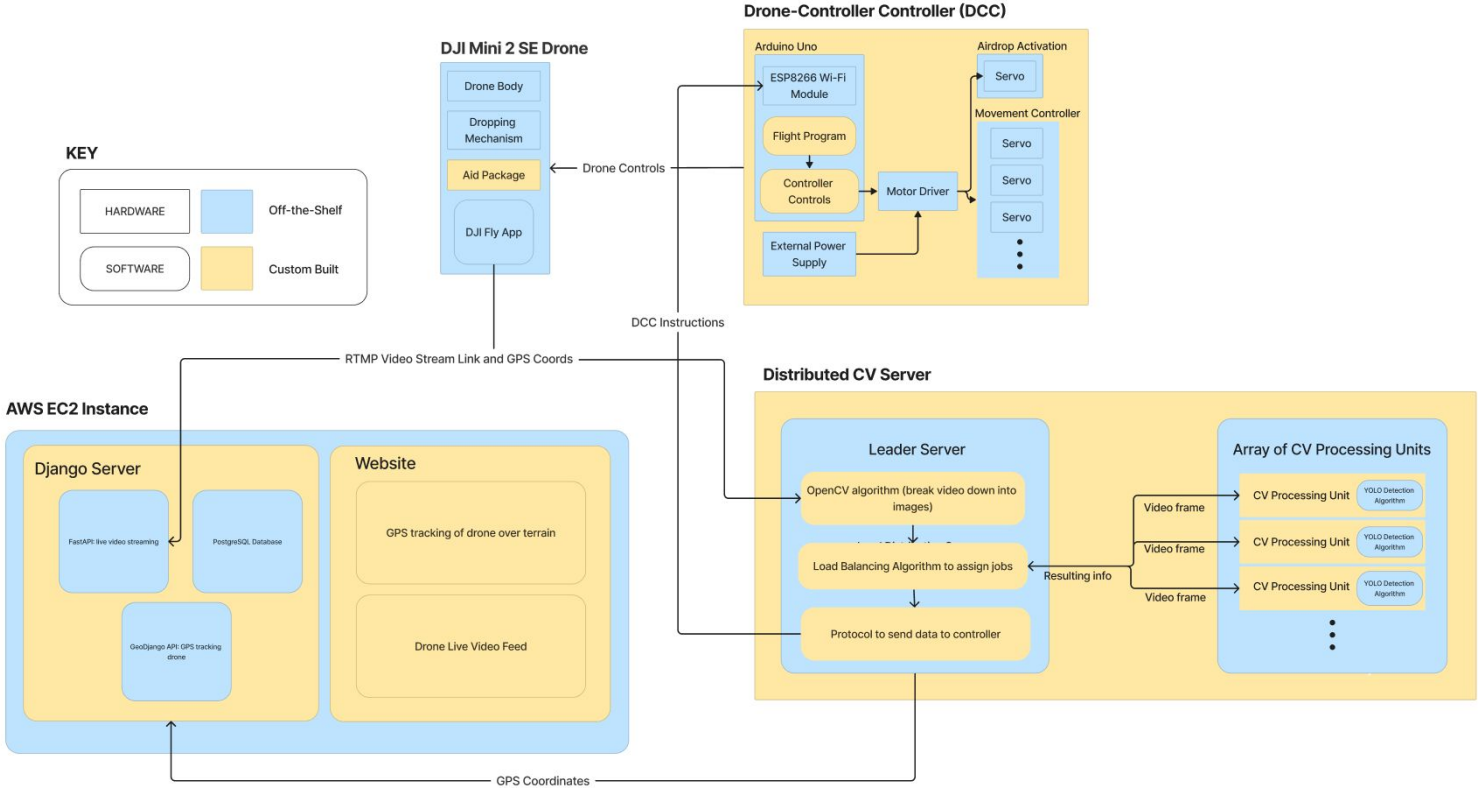
Sub-requirements

- Web application deployment via Amazon EC2 to allow users in all geographical locations to connect with the website
- Map interface to visualize locations of people and the drone

Technical Challenges

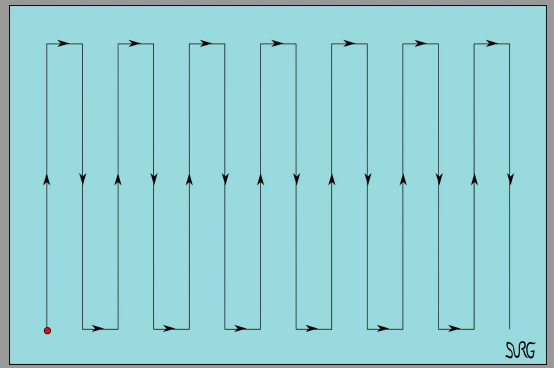
- Create a **controller** setup that uses motors to control the drone to autonomously comb through a predefined area and drop off an aid package
- Incorporate **computer vision** in a **distributed server** to identify people within a video stream received from the drone and processes frames concurrently
- Design a **website** that allows for monitoring of the drone's status at any point, including its position and live footage
- Maintain **low latency** communication between components
- Automatic/manual dropping **mechanism**

Solution Approach



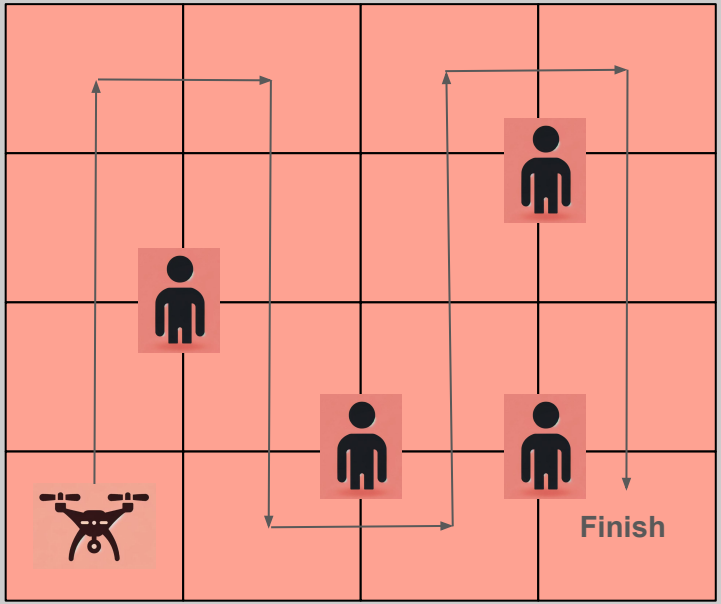
Solution Approach - Flight Program

- Ascend to pre-specified height
- Perform initial search in a creeping line search pattern
- Upon receiving coordinates of found person, determine which direction to fly given the current drone's coordinates
  - Maintain a moving average of the person's coordinates to account for imprecision
- Upon reaching coordinates, descend lower and release package, then ascend to original height
- Upon receiving coordinates of original start point, return to home



Source: https://en.m.wikipedia.org/wiki/File:Creeping_line_search_pattern.png

Flight Control & Waypoint Testing



- **Algorithm Tests:** to test human identification within a video stream, the speedup achieved through load balancing, and website unit tests

- **Scenario Tests:** to test the drone controller, detection of various “humans” within the area, and integration of all components

Testing, Verification, and Metrics

Requirements	Testing	Metrics
Accurately identify humans in a flat landscape	Unit test images of people in different flat environments, body parts hidden, multiple people	Top-1 Accuracy: > 80% Top-5 Accuracy: > 90%
Autonomous flight control	General DCC checkpoint tests	Can fly in pre-specified pattern, airdrop successfully, and correctly navigate to person if found
Low latency	Time taken to send, process, and receive flight and video data	Latency of detection, data routing, and result processing: < 5s
Calculation for person's GPS location	Comparison tests between person's actual coordinates vs calculated coordinates using altitude and scraping video frame data	Offset in person location and GPS calculation: ± 3 feet
Travel to waypoint	Field tests to send drone to specific GPS coordinates	Drone can accurately fly and hover over within 1 foot radius of coordinates

Tasks and Division of Labor

- **Development of the Drone Controller (DCC):** David
- **Wifi Module & Communication with Servers:** David
- **Load Balancing Algorithm & Distributed Server:** Ronit
- **Computer Vision Algorithm:** Ronit & Nina
- **Live Video Streaming & GPS Geolocation:** Nina
- **Web Application & Django Server:** Nina
- **Hardware & Software Integration:** All members
- **Scenario Design & Testing:** All members

Schedule

task/milestone	description	person/people with primary responsibility	start date	finish date	Wed 2/7	Wed 2/14	Sun 2/18	Wed 2/21	Wed 2/28	Fri 3/1	Wed 3/6	Wed 3/13	Wed 3/20	Wed 3/27	Wed 4/3	Wed 4/10	Wed 4/17	Sun 4/21
Obtain materials	Perform research on correct materials to buy (motors, Arduino components, etc.), and purchase them	David	2/7	2/14	x	x												
Load Balancing Algorithm (1/2)	Perform research on optimal load balancing algorithms and implement in Go	Ronit	2/7	2/14	x	x												
General Website Setup	Create base template for Django website, do front/backend setup of site, and research hosting live video streaming	Nina	2/7	2/14	x	x												
Design Presentation Slides	Design Presentation Slides	All	2/14	2/18		x	x											
Begin coding of flight program logic	Inital coding of logic (set up interface and general inputs/outputs of the Arduino)	David	2/14	2/21		x	x	x										
Load Balancing Algorithm (2/2)	Implement the load balancing algorithm in Go, and finish unit testing. Research fetching the video stream from the DCC	Ronit	2/14	2/21		x	x	x										
Implement Live Video Stream and GPS Tracking	Create working displays of live video feed and gps tracking of drone, extra features to improve UX	Nina	2/14	2/21		x	x	x										
Examine parts, begin assembly	Examine parts, build simple Arduino wifi receiver to allow for testing with CV servers	David	2/21	2/28				x	x									
Extract Real Time Data + Calculation Algo	Implement routing to send flight data to website and cv server, GPS calculation of human location	Nina	2/21	2/28				x	x									
Computer Vision Algorithm on the Distributed Server	Implement the Computer Vision Algorithm on the CV processing units. Integrate with video frames and unit tests	Ronit	2/21	2/28				x	x									
Perform DCC assembly - hardware motor building, spring break	Develop mechanism for being able to move the switches, develop DCC box, fit components into the entire box	David	2/28	3/13					x	x	x	x						
Deploy website, UI/UX, + Fully integrate	Deploy code on EC2, and implement a better UI/UX, Fully integrate and test all components are connected efficiently	Nina & Ronit	2/28	3/13					x	x	x	x						
WiFi chip integration with DS	Implement protocol to send data to the DCC. Test this out with David	Ronit & David	2/28	3/13					x	x	x	x						
Test initial creeping search	Test autonomy and creeping search, airdrop package and accuracy involving GPS coords	All	3/13	3/20								x	x					
Test entirety of DCC	Making sure everything is integrated and working with server and website	All	3/20	3/27									x	x				
Slack	Slack time	All	3/27	4/21										x	x	x	x	x
Final Presentation Slides	Final Presentation Slides	All	4/17	4/21													x	x