TEXAS A&M UNIVERSITY
Engineering

Aerial Pathfinding Reconnaissance
Final Presentation

Jason Gilman, Max Griffith, Mark Johnson, Dilanka Weerasinghe

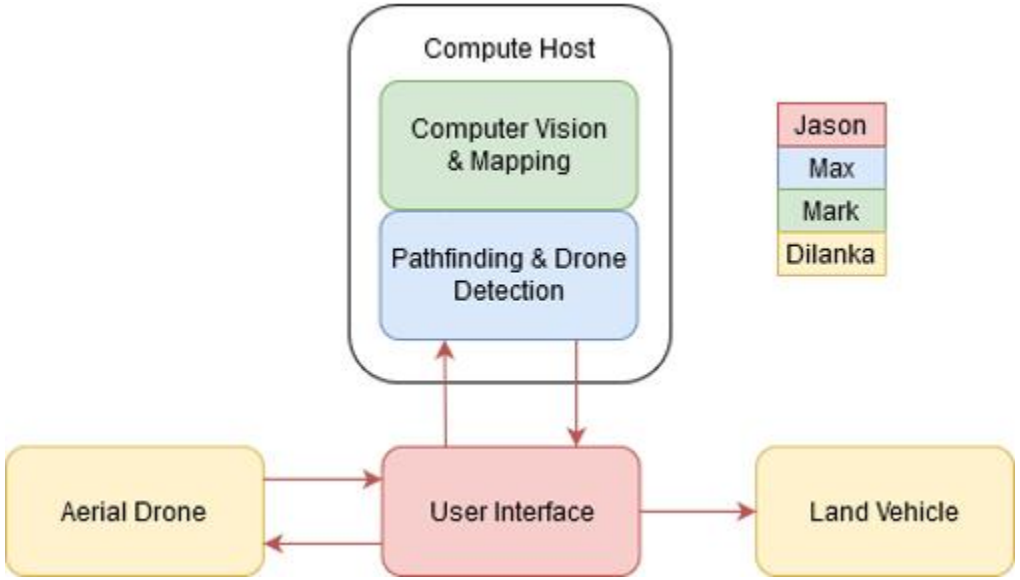
Overview

TEXAS A&M UNIVERSITY Engineering

Goal: Use aerial drone footage to find a safe path for a land vehicle.

Team Members and Responsibilities:

- Dilanka Weerasinghe *(Data Collection & Drone Control)*
  - Automated drone control and collection of aerial footage
- Mark Johnson *(Computer Vision & Mapping)*
  - Production of an obstacle map from drone videos
- Max Griffith *(Pathfinding & Drone Detection)*
  - Determination of vehicle starting location and route
- Jason Gilman *(Device Networking & UI)*
  - Creation of user interface and network interconnect



System Block Diagram

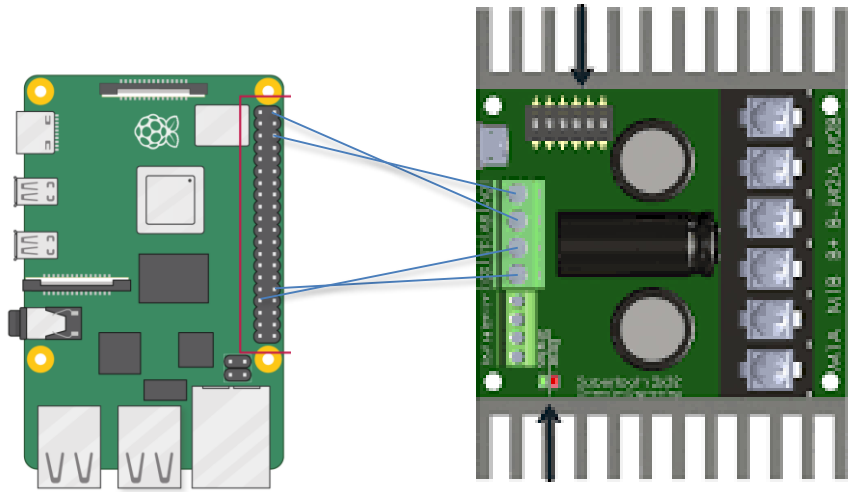
Data Collection & Drone Control

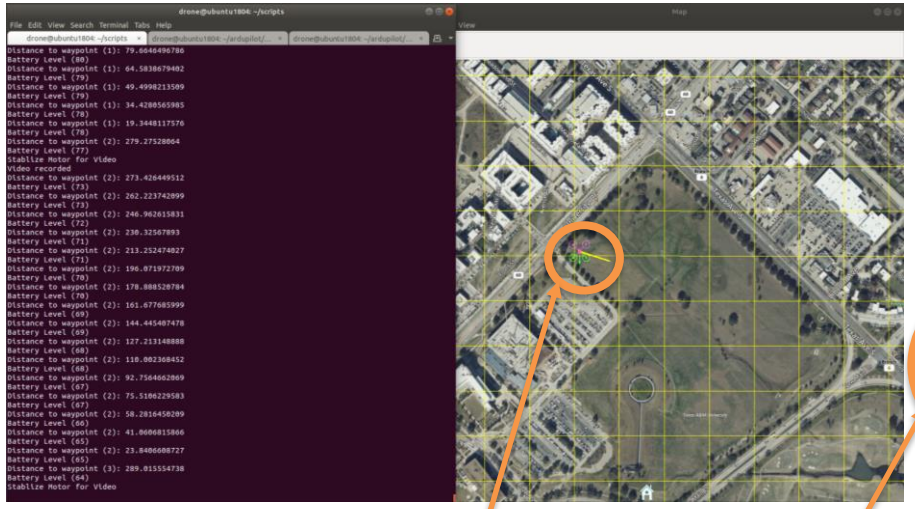
TEXAS A&M UNIVERSITY Engineering

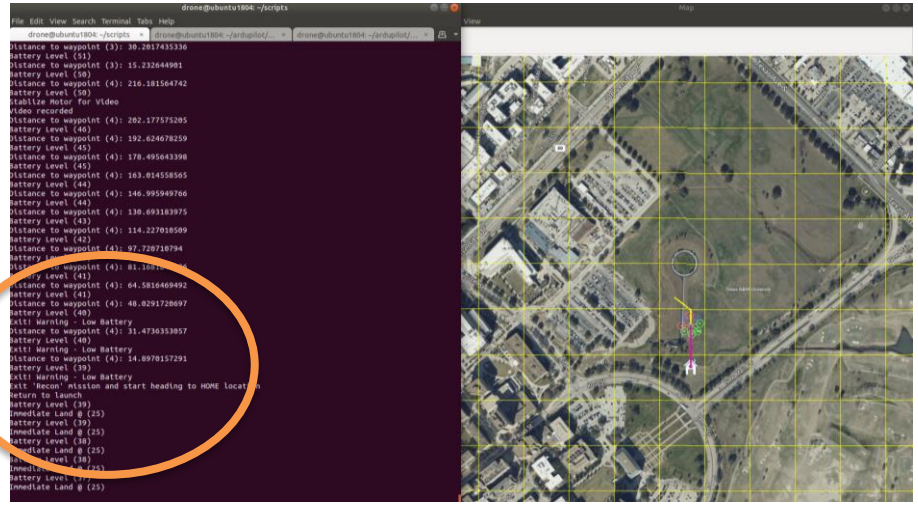
Accomplishments:

- Physical Electronics
  - PiCam records interval video and records to a datasheet with timestamp & GPS location
  - Avoidance system mounted to Aerial Drone with sensor hub
  - Land Rover complete with serial connection through Pi3. Not tested.

- Simulator
  - Drone reads in data file with altitude, camera angle yaw and waypoint locations
  - Drone performs safety check arms and takes flight.
  - Obstacle avoidance case for GUIDED directions
  - Loading commands to vehicle through serial/udp connection.
  - Flight distance, directionality and logging completed.



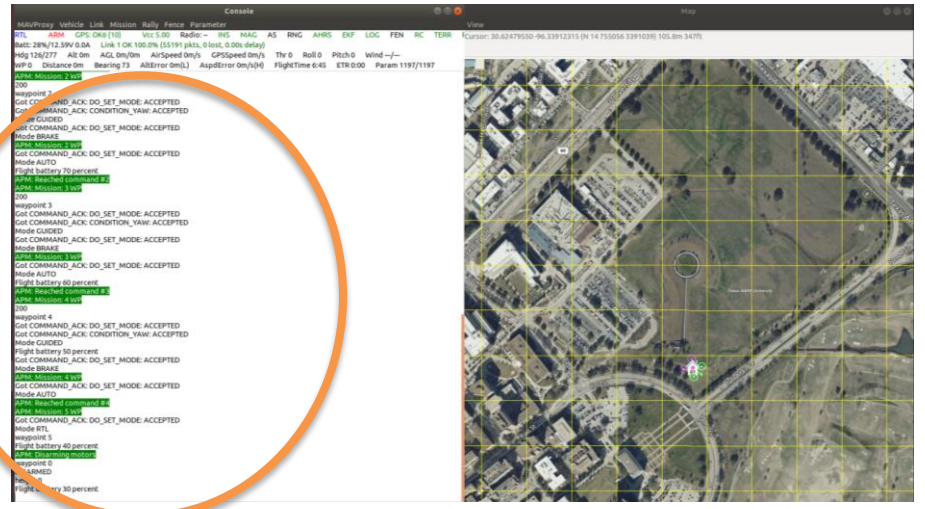




On data collection yaw set to 270˚, video last 5 seconds

On low battery drone logs warning and returns home. If battery is less than 25% land immediately.

Commands sent to PixHawk running ArduCopter firmware.



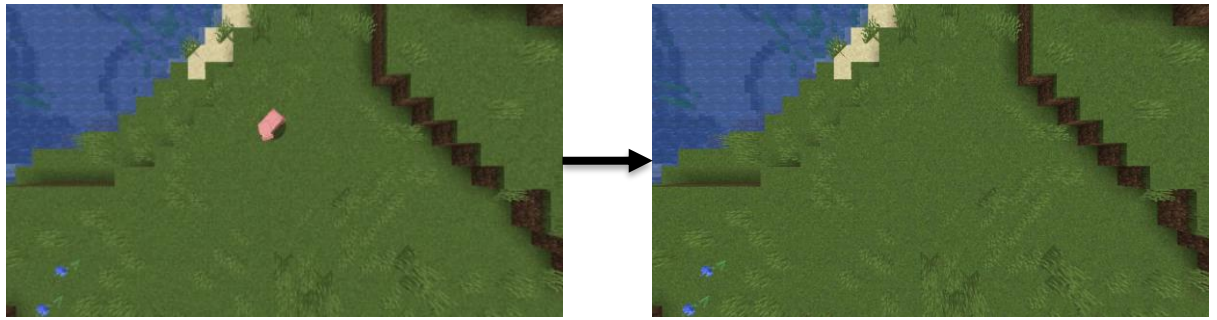
Computer Vision & Mapping

TEXAS A&M UNIVERSITY Engineering

All stages are complete:

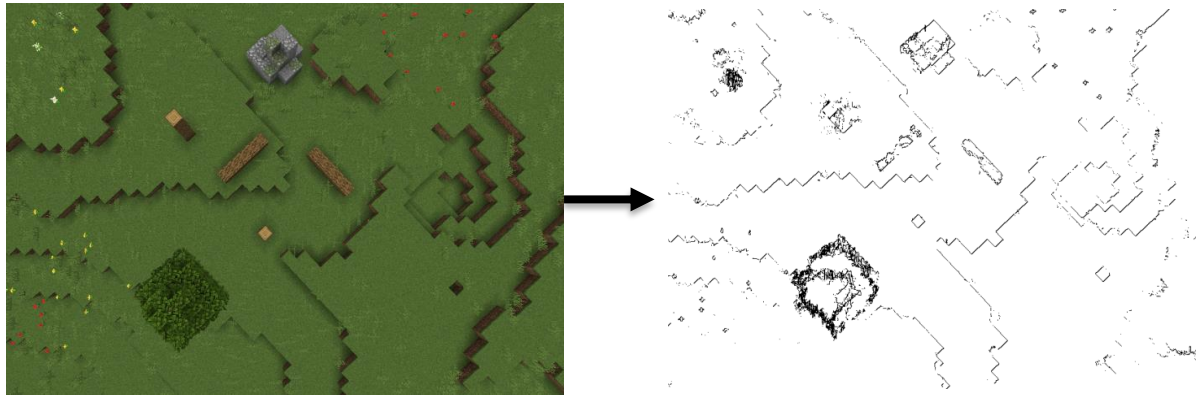
- Moving Object Removal
- Image Stitching
  - Poor result due to changing perspective.
  - Final product will not rely on this feature.
- Depth Map Inference
  - Parameters need tweaking at runtime.
- Depth Map Stitching
  - Custom algorithm developed for this project.
- Gradient Field Derivation
- Obstacle Delineation

Demo: Moving Object Removal



Input Video | Output Image

Demo: Obstacle Detection by Slope Analysis



Input Image (Left Eye View) | Output Obstacle Map

Computer Vision & Mapping
Full Pipeline Example

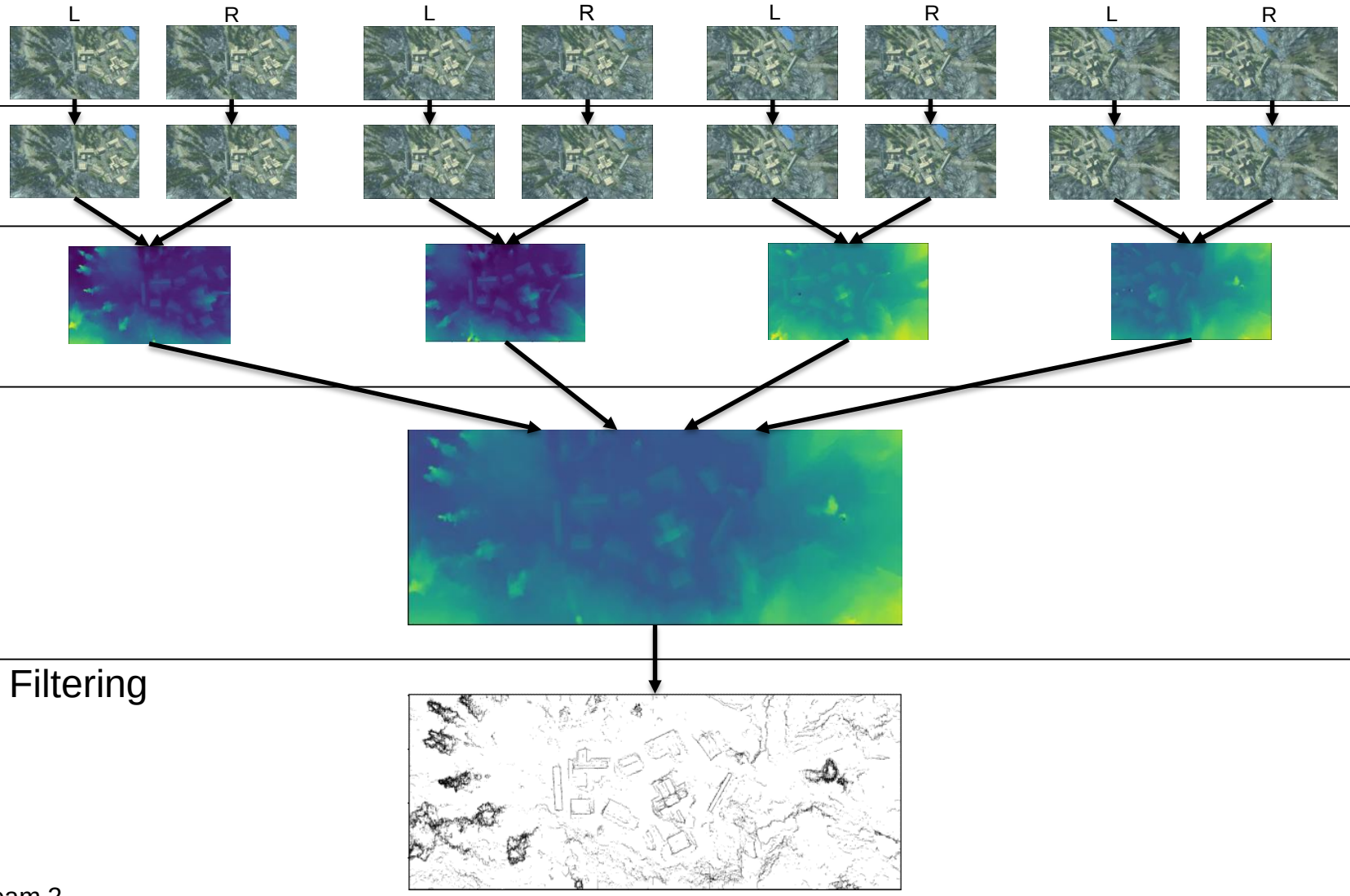
TEXAS A&M UNIVERSITY
Engineering

Stage 1.
Moving Object Removal

Stage 2.
Depth Inference

Stage 3.
Depth Map Merging

Stage 4.
Gradient Derivation & Slope Filtering



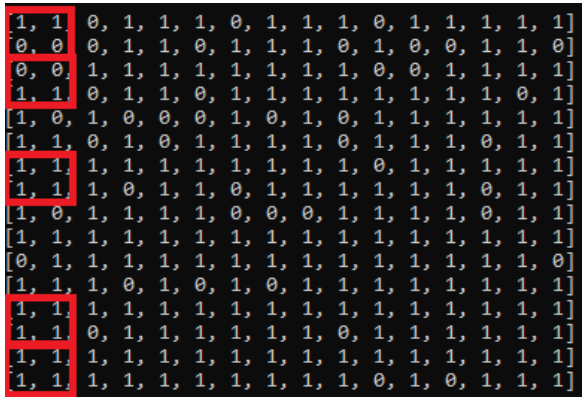
Pathfinding & Drone Detection

TEXAS A&M UNIVERSITY Engineering

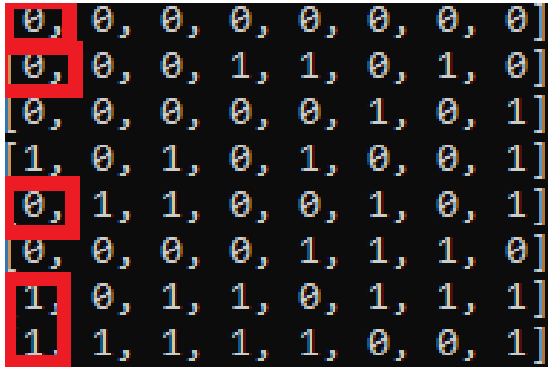
Stages Complete:

- Finding straightforward path
  - Working as intended
  - Paths not taken for final path are marked as invalid (0)
  - Greedy algorithm taking rightmost one
- Edge cases and error handling
  - Edge cases such as finding cycles have been fixed (infinite loops), improper start/ending positions
- Decrease Resolution from a map
  - Only works with evenly divisible pixel widths

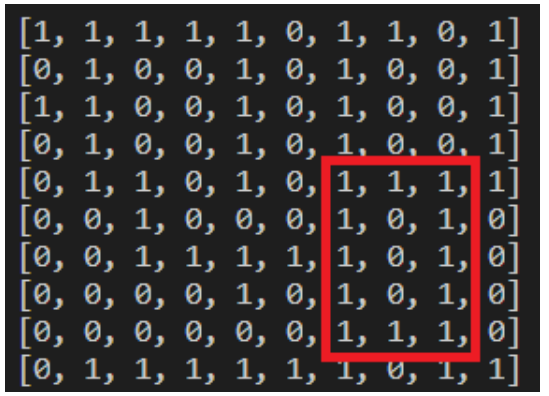
Random Generated Map



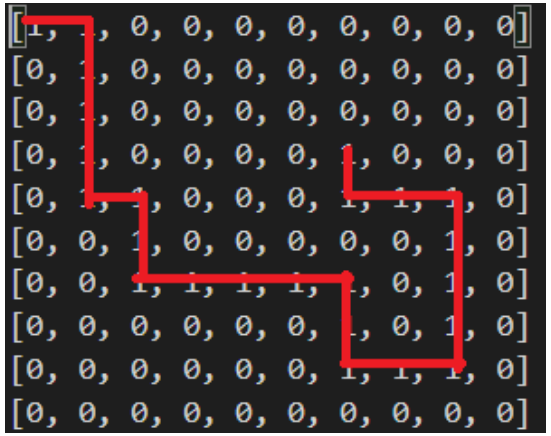
Map with decreased resolution



Cyclic Path:



Cyclic Path Solved*:



*The starting location is [0,0] and the ending location is [3,6]

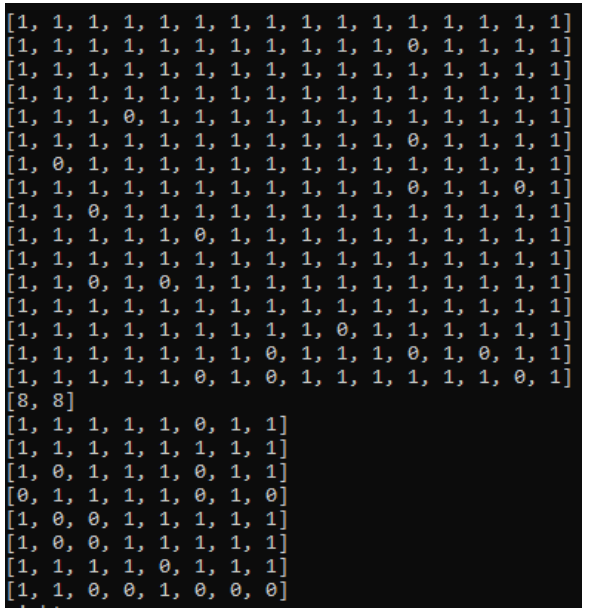
Pathfinding & Drone Detection

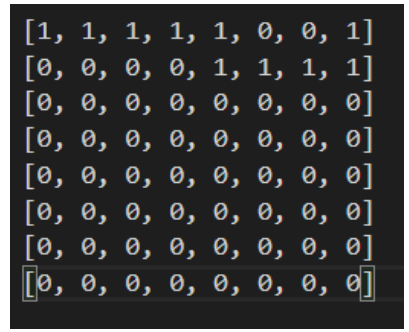
TEXAS A&M UNIVERSITY Engineering

Map Resolution decreased to output file

Example

- 16 by 16 matrix with 2x2 pixels
- Minimizes to 8x8
- Valid path created





Demo Plans:

- Show valid path with edge cases taken care of
- Show different resolutions of differently sized maps

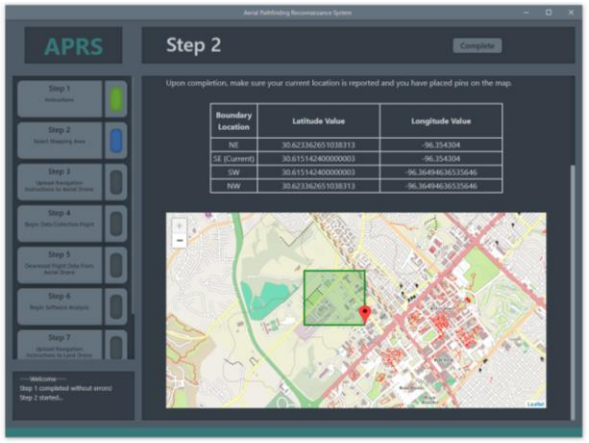
Device Networking & UI

TEXAS A&M UNIVERSITY
Engineering

Stages Complete:

- Desktop Application:
  - Mapping functionality based on user's current location
  - Generate navigation instructions for the aerial drone
  - Send navigation instructions, launch flight, and download data from aerial drone through HTTP methods on local wifi network
  - Error handling to prevent invalid operation
  - Full functionality offline

- Aerial Drone Server:
  - Raspberry Pi 3b+ acting as a wireless access point using TL-WN722N wireless module (range ~30 m)
  - Running a lightweight web-server to receive and process communications from the user



Mapping functionality: valid area

Error handling and GUI message

```
pi@raspberrypi:~ $ sudo systemctl status aprsAerialServer
● aprsAerialServer.service - aprs server
   Loaded: loaded (/etc/systemd/system/aprsAerialServer.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2020-11-08 04:59:28 GMT; 18h ago
 Main PID: 409 (bash)
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/aprsAerialServer.service
           ├─409 /bin/bash -c source /home/pi/Desktop/aprs/aprs-aerial-server/env/bin/activate;python /home/pi/Desktop/a
           └─420 python /home/pi/Desktop/aprs/aprs-aerial-server/aprsAerialServer.py

Nov 08 04:59:30 raspberrypi bash[409]:  * Environment: production
Nov 08 04:59:30 raspberrypi bash[409]:    WARNING: This is a development server. Do not use it in a production deploymen
Nov 08 04:59:30 raspberrypi bash[409]:    Use a production WSGI server instead.
Nov 08 04:59:30 raspberrypi bash[409]:  * Debug mode: off
Nov 08 04:59:30 raspberrypi bash[409]:  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
Nov 08 05:14:58 raspberrypi bash[409]: 10.0.0.20 - - [08/Nov/2020 05:14:58] "GET / HTTP/1.1" 200 -
Nov 08 23:34:37 raspberrypi bash[409]: 10.0.0.20 - - [08/Nov/2020 23:34:37] "POST /uploadInstructions HTTP/1.1" 200 -
Nov 08 23:35:24 raspberrypi bash[409]: 10.0.0.20 - - [08/Nov/2020 23:35:24] "POST /uploadInstructions HTTP/1.1" 200 -
Nov 08 23:36:26 raspberrypi bash[409]: 10.0.0.20 - - [08/Nov/2020 23:36:26] "POST /uploadInstructions HTTP/1.1" 200 -
Nov 08 23:36:47 raspberrypi bash[409]: 10.0.0.20 - - [08/Nov/2020 23:36:47] "POST /uploadInstructions HTTP/1.1" 200 -
```

Drone server showing communication with UI

Device Networking & UI

TEXAS A&M UNIVERSITY Engineering

Data Collected:

- Communication latency to back-end and drone servers
- Aerial drone navigation instructions

Demo Plans:

- Walk-through of desktop application
- Display offline/online options
- Demonstration of mapping utility
- Show automated communication to drone server
- Show error handling process

6725	200	png	TileLayer.js:158	58.8 kB	11 ms
6725	200	png	TileLayer.js:158	22.1 kB	318 ms
6725	200	png	TileLayer.js:158	60.7 kB	11 ms
6724	200	png	TileLayer.js:158	59.6 kB	326 ms
6724	200	png	TileLayer.js:158	54.2 kB	326 ms

Timing and image size when fetching map assets

10.0.0.1	200	fetch	step3.tsx:76	199 B	264 ms
10.0.0.1	200	fetch	step4.tsx:99	199 B	2.60 s

Timing when communicating with drone server

```
10
30.61510506006647,-96.35762462809495
30.6150303801994,-96.35762462805874
30.61495570033233,-96.35762462802253
30.614881020465262,-96.35762462798633
30.614806340598193,-96.35762462795013
30.614731660731124,-96.35762462791392
30.614656980864055,-96.35762462787771
30.614582300996986,-96.3576246278415
30.614507621129917,-96.35762462780531
30.614432941262848,-96.3576246277691
30.61435826139578,-96.35762462773289
30.61428358152871,-96.35762462769668
30.61420890166164,-96.35762462766047
30.614134221794572,-96.35762462762426
30.614059541927503,-96.35762462758805
30.613984862060434,-96.35762462755184
30.613910182193365,-96.35762462751563
```

Generated navigation instructions

Execution Plan

TEXAS A&M UNIVERSITY
Engineering

Milestone	Assigned To	Week 1 24-Aug-20	Week 2 31-Aug-20	Week 3 7-Sep-20	Week 4 14-Sep-20	Week 5 21-Sep-20	Week 6 28-Sep-20	Week 7 5-Oct-20	Week 8 12-Oct-20	Week 9 19-Oct-20	Week 10 26-Oct-20	Week 11 2-Nov-20	Week 12 9-Nov-20	Week 13 16-Nov-20	Week 14 23-Nov-20
Understand Project	All														
Initial planning/research	All														
Draft Conops Report	All														
Source Lab Materials (Drone/Sensors)	All														
Draft FSR Report	All														
Draft ICD Report	All														
Draft Exection Plan	All														
Draft Validation Plan	All														
Prepare Midterm Presentation	All														
Design Server-Side Architecture	Jason														
Remove Moving Obstacles from Data	Mark														
Increase familiarity with Python	Max														
Implement Sever-Side	Jason														
Map Environment from Data	Mark														
Create algorithm for valid accessible path	Max														
Research/Order Parts	Dilanka/Jason														
Create error handling for inaccessible path	Max														
Implement Client-Side/UI	Jason														
Estimate Depth on Path Sections	Mark														
Research Drone SDK / stm32 / Sabretooth	Dilanka														
Add flight mapping functionality	Jason														
Research software interaction with drone	Max														
Design Aerial/Land Electrical Schematics	Dilanka														
Program manual override of drone	Max														
Testing Electrical Off Drones	Dilanka														
Merge Depth Estimates with Map	Mark														
Build Air Drone	Dilanka														
Build Land Drone	Dilanka														
Prepare Status Update Presentation	All														
Implement Server on drone	Jason														
Test Flight of Air Drone	Dilanka														
Normalize Depth Field for Drone Altitude	Mark														
Program movement of drone	Max														
Test Travel of Land Drone	Dilanka														
Communicate with server on drone	Jason														
Derive Gradient Field From Elevation Map	Mark														
Program: Data Retreival of Air Drone	Dilanka														
Test Drone Simulation Programs	Dilanka														
Test Data Retreival From Drone	Dilanka														
Implement Obstacle Delineation	Mark														
Program drone path from map input	Max														
Package Software/Create Installer	Jason														
Prepare Final Presentaion	All														
Test and modify Air system	Dilanka														
Program: Autonomous Flight of Air Drone	Dilanka														
Draft Final Report	All														
Prepare Final Demo	All														

Legend
- Completed
- Deadline
- Planned Work Time
- In Progress
- Delayed

Validation Plan

TEXAS A&M UNIVERSITY Engineering

Legend	Complete	Incomplete

Data Collection and Drone Control	Aerial Drone flies from Point A to Point B	Drone follows waypoint from data file w/ interups	
	Drone stops at midpoint locations on the map and records video and GPS location	Drone Stablize, autofocus, record & save	
	Drone does not takeoff with failed connection/ prearm checks	Failure to lauch without udp / serial connection	
	Low Battery Warning & Exit	Drone RTL on < 40%; Drone Lands on < 30%	
	Add wp during flight and guided travel		
	Drone stops at user shutoff switch	Controller has switch that changes from AUTO to manual	
	Land Drone Functional with Directions	Land drone is built and programmed	
	Drone detects obstacles & Avoides	Dectection 0.05-3m, not operational	
	Electrical Systems Connected	No Pixhawk Power	
	System Built & Power On	No flight controller and battery	

Data Collection and Drone Control - Dilanka

Pathfinding and Drone Detection	Greedy Algorithm given a feasible path	Given a feasible path, if there is a valid path, it will be found 100% of the time	
	Edge cases such as Incorrect Start/End position causes program to end and output no map and error statements	If the start/end cases are not accurate, the program will not run and output a map of 0s	
	Handling Cyclic paths	If a cyclic path is found, the program will solve the cyclic path, retracing its steps once the same point is found twice	
	Changing Resolution of given map given pixel width	Works with 0 even division. 16 / 4, 8/4, 100/50 etc...	
	Updating Algorithm to handle weighted paths		

Pathfinding and Drone Detection - Max

Computer Vision and Mapping	The system produces depth maps with fewer than 1% of negative data points	Invalid points do not appear after the filtering stage. 0%	
	A change in elevation of <2% of drone altitude results in >10 disparity	Measured manually; can not be detected. 1.26%->12	
	The system can identify changes in elevation which account for <0.1% of the image area	Measured manually; object was 1.6% altitude with 15 disparity. 0.003%	
	Depth estimation occurs at a speed of >1MP/s for each image in a pair	Calculated automatically. 1.6 MP/s	
	Depth maps are merged to an accuracy of <0.5% of output map width	Measured manually; average over 14 trials. 0.26%	
	Image stitching correctly throws an exception when insufficient image data is provided	The error is caught from return value and raised as an exception	

Computer Vision and Mapping - Mark

Device Networking and UI	Fetch user's current location via geolocation api call (internet required) in under 5s	avg time to fetch = 145 ms size of data = 442 B	
	Fetch map images in under 5s (offline assets served by back-end)	avg time to fetch = 253ms avg size of data = 76.3 kB	
	Error handling and message provided in GUI to prevent crash and inform user	Error messages will be presented if the user attempts invalid operation	
	Create drone navigation instructions accurate within 1 Meter		
	Communicate with drone's server in under 10s	avg time to post data = 2.91 s avg size of data = 388.38 kB	
	Install software via encapsulated windows installer	create a single application containing all functionality and dependencies	

Device Networking and UI - Jason

TEXAS A&M UNIVERSITY
Engineering

Questions?