

# UAV SEARCH & RESCUE

## SENIOR CAPSTONE

---

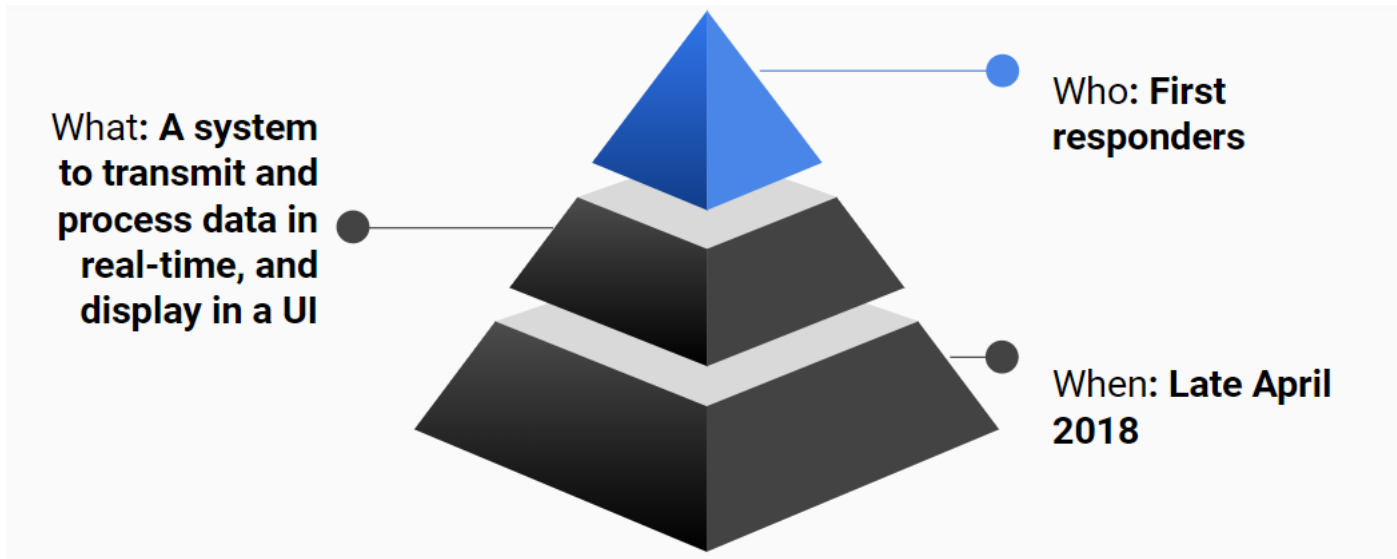
Kevin Dorosh, Richard Preston, Ryan  
Stocking, Elyse Cooper  
Sponsor – Prof. Karen Panetta

# Drones!

- What are we using them for?
  - Ok, let me give you a (terrifying) scenario..
- Disaster scenarios
  - Dirty bomb
  - Natural disaster
  - Nuclear power plant failure
    - e.g., Chernobyl
    - e.g., Stuxnet
      - On related note, Russia hacked a US power plant July of this year!



# The Project



- Why
  - Faster response times and fewer lives lost



# Customer Requirements

- Compatible with 3DR Solo Drone
- Supports picture, thermal, and radiation sensing
- Accurate GPS location
- Real time transmission of data
- Intuitive UI for data visualization



# Hasn't this been done before?

- This project has actually been in Panetta's lab for years
  - Many variations have seen different levels of success
- The U.S. military/government already has this tech, but it isn't open-sourced for use by low-ranking first-responders
  - Open source tools have come far
    - We are contributing toward OpenDroneMap and other tools as we leverage them



# How it works

- Fairly simple!



## Inputs

Thermal camera images

GPS coordinates

Radiation counts per second

## Outputs

Visualization of data on UI

- Slight pivot after stakeholder interview with Luke Harwath
  - Director of Communications for Help.NGO – Global Dirt since 2011
    - First deployment to Japan to deliver dosimeters (detect how much radiation someone has been exposed to)
    - Radiation can have 50km radius
    - Used radioisotope identifier coupled with GPS to map radiation
    - Radiation varies greatly with height
    - Fixed wing would be better for speed – unfortunately that's not what we have



# The pieces

- Software
  - Ardupilot
  - MissionPlanner
  - WebODM
    - This is what I am building off of
- Hardware
  - Black box attachment
    - Thermal camera
    - Geiger Mueller tube
    - Standard images & GPS
- WebODM
  - Database
  - Nodes for image processing
  - Front End



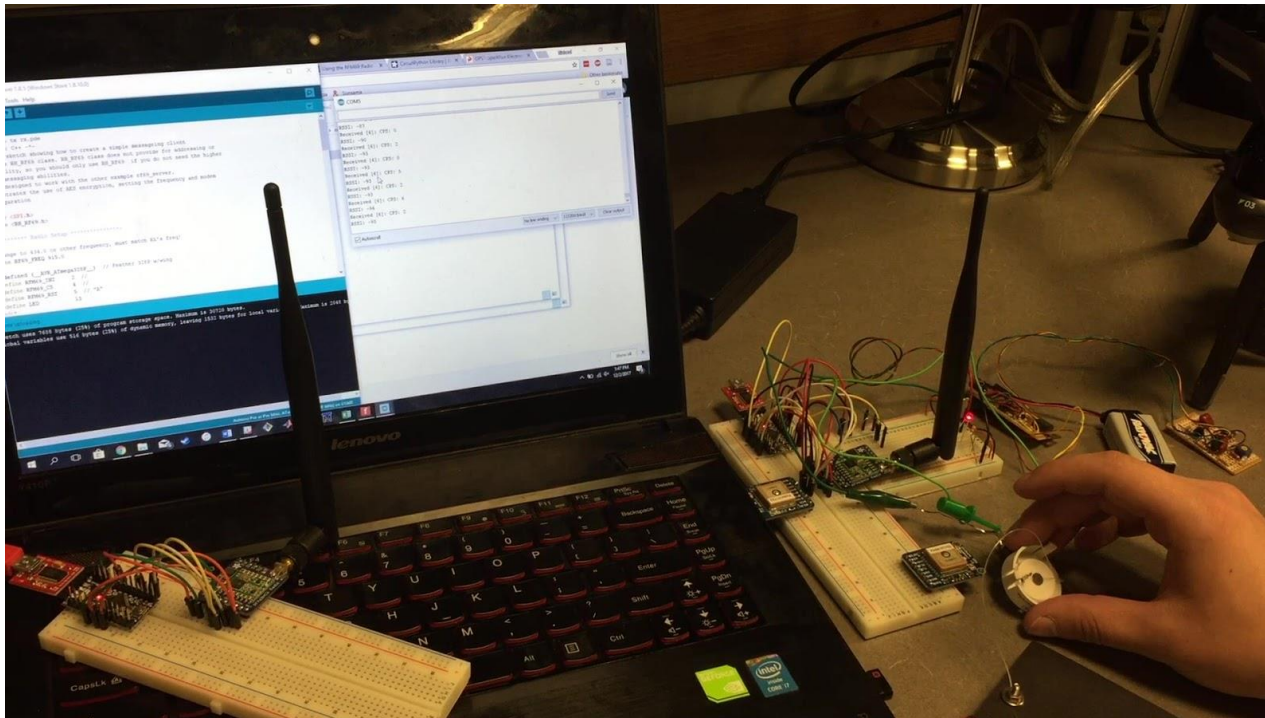
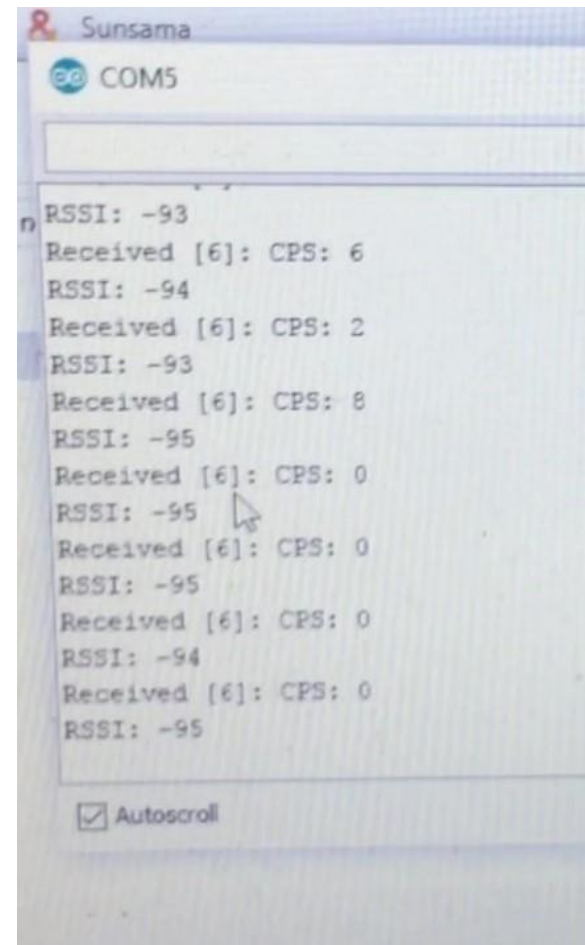
MissionPlanner (above)



Geiger Muller Tube (above)

# What are the hardware people doing?

- Hardware proof of concept
  - Measured alpha particles from a smoke detector
  - Transmitted to receiver and displayed Counts Per Second (CPS) on Arduino serial monitor





# What am I doing?

- Learning A LOT of technologies I've never used before to leverage WebODM
  - NodeJS (and npm)
  - Docker
  - Django
  - React
  - Babel
- Learning drone/imaging terminology
  - Dense Point Cloud
  - 3D Mesh
  - Textured Mesh
  - LAS format point cloud
  - PLY format point cloud
  - etc.
- Learned how to fly the drone
- Began modifications to WebODM
  - Small UI changes so far



Hey look. It's me. A drone at Tufts!

# The Image Processing



DJI\_0176.JPG



DJI\_0177.JPG



DJI\_0178.JPG



DJI\_0179.JPG



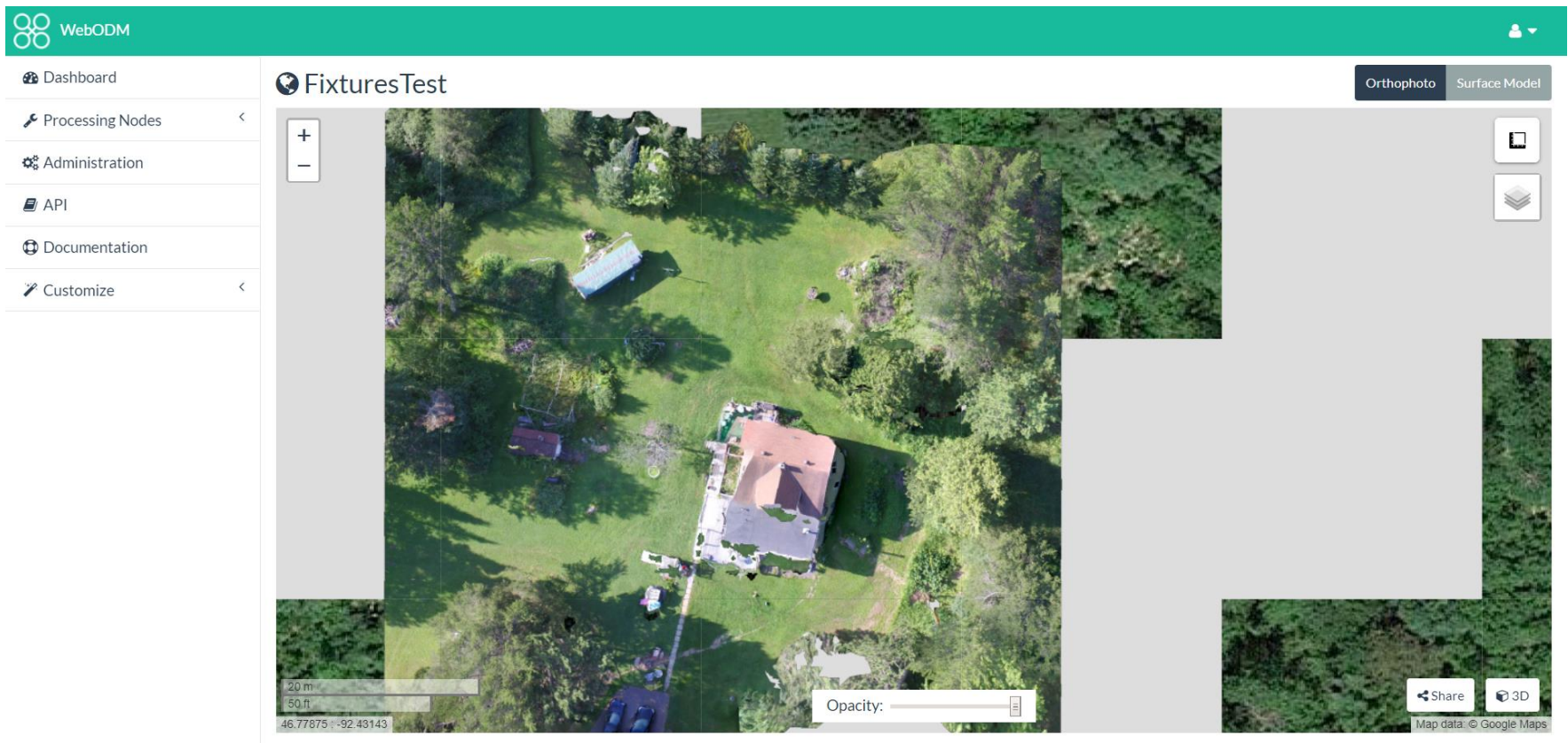
DJI\_0180.JPG

- Becomes:



# WebODM

- Baseline map overlay/display



# For next semester

- We don't want or need to use the browser
  - We don't even expect to have an internet connection!
  - Thus I may package the UI into cross-platform desktop applications using Electron
- Django comes with a lot of user authentication and support for multiple nodes for image processing
  - We need a less robust solution that is simpler and runs quicker
  - We need to support image stitching with gaps (i.e., doesn't have 60% overlap)
- Front End
  - Support UI tooltips and overlays for thermal and radiation data
- Back End
  - Read data in from serial port rather than taking user uploaded files
    - Important since this needs to run in REAL TIME
- Testing
  - Homeland security

# Challenges

- I am building top-down and they are building bottom up
  - We are at risk of not being able to “plug and play” when the time comes
    - We have designed an API we will (attempt) to adhere to that should minimize this risk
- Unknown bottlenecks
  - We need to know how quickly we can send and process data with our prototype hardware
- Scaling
  - I am a one-man UI/image processing team!



# Questions?