EE/SE 492: sdmay24-08
Path Planning of Aerial Robots with Reinforcement Learning
Bi-Weekly Report 8
2/12/24 - 2/23/24
Client: Amir Niaraki
Advisor: Ali Janessari

Team:
Akash Setti
Andrew Sailer
Cody Draper
Jesse Gillingham

Past Week Accomplishments:

- Training drone to search field sparsely populated with rewards
    - Ran more training experiments while changing parameters trying to get new behavior
    - Ran into issues with the environment
- Improvements to Complete Coverage - Akash and Cody
    - Changed the image processor to handle .tif files
    - Changed the image processor to handle very large images
        - Resizes the image, records points, resizes the points to the original image size, and applies a mask to the original image.

Pending Issues:

- Reevaluate cost and wind function, make sure math still makes sense and drone behaves as expected
- Resolve environment issues ASAP
- Optimization of Complete Coverage algorithm to follow state-of-the-art examples in industry by changing search pathing based on searched polygon.
- Implement parallel training of learning model to decrease time overhead of training
- Debug image processor
- Integrate image processor into complete coverage

Individual Contributions:

Member	Contribution	2-Week Hours	Cumulative Hours
Akash Setti	Tried to implement CC on specified image	8	53
Andrew Sailer	Ran training experiments for sparse rewards scenario	10	85
Cody Draper	Rewrote image processor to handle large .tif files	15	65

Jesse Gillingham	Ran a Training experiment	8	68

Plans for Next Week:

- Reinforcement Learning
    - Resolve issues with environment and retrain for previous scenario
    - Balance detection rewards and movement costs
    - Change observation space and reward function according to research paper linked by client
    - Output list of locations when demoing (testing)
    - Explore options to run training simulations in parallel to decrease time overhead
        - Explore ways to average model outputs for different training data in parallel
- Complete Coverage and Environment
    - Debug image processor
    - Mask the large .tif file the client provided for us
    - Integrate image processor into complete coverage
    - Alter complete coverage to accept arguments
        - “-p” flag to process image
        - “Image” argument to signify cc to not randomly generate a world but use the one provided.