Path Planning of Aerial Robots with Reinforcement Learning

Design Document

Team Name
sdmay24-08

Client
Amir Niaraki

Adviser
Ali Jannesari

Team
Andrew Sailer, Akash Setti, Cody Draper, Jesse Gillingham

Email | Website
sdmay24-08@iastate.edu | https://sdmay24-08.sd.ece.iastate.edu/

Revised
12/03/2023

Executive Summary

Project in Brief

The goal of this project is to train a drone using reinforced learning to traverse a variety of environments such that maximum anomalies are found and energy usage is minimized. This will be accomplished by training models in a simulated environment using Python.

Development Standards & Practices Used

List all standard circuit, hardware, software practices used in this project. List all the Engineering standards that apply to this project that were considered.

- Python
- OpenAI
- StableBaselines3

Summary of Requirements

Simulate drone search of a polygon. Return metrics of interest including battery usage, time taken, anomalies found.

Applicable Courses from Iowa State University Curriculum

CprE 288, EE 526, EE 322, ComS 474, DS 303, ComS 309, EE 285, ComS 327

New Skills/Knowledge Acquired During Project

- Understanding of Reinforcement Learning
- Complete Coverage Algorithms
- Git for collaborative coding projects
- Collaboration/communication on coding projects

Table of Contents

1 Preamble

1.1 Team Members

- Jesse Gillingham
- Andrew Sailer
- Akash Setti
- Cody Draper

1.2 Required Skill Sets for Project

- Programming
- Problem Solving
- Making Design Compromises

1.3 Skill Sets of Team

- Jesse Gillingham: Hardware, Embedded Systems,
- Andrew Sailer: Hardware, Control Systems, Some Programming.
- Akash Setti: Software Development, Data Science, Machine Learning
- Cody Draper: Software Development, Embedded System, Machine Learning

1.4 Project Management Style

We are using a hybrid approach to our project management style. We are incorporating pieces from both the Agile and Waterfall approaches. Each week we meet with our client to discuss what we accomplished last week and how we should move forward. Then, during the week we start working on the new features we discussed. This is what we have found to be the best approach to managing our project.

1.5 Initial Project Management Roles

- Jesse Gillingham: Artificial Intelligence research
- Andrew Sailer: Take notes on in person meetings
- Akash Setti: Record online meetings
- Cody Draper: Coding Support

1.6 Problem Statement

Train a drone via reinforced learning to traverse a variety of environments such that maximum area is covered while minimizing energy usage. The drone completes a desired objective over an assigned area.

1.7 Requirements & Constraints

1.7.1 Functional Requirements

- Codebase should contain a variety of reinforcement learning (RL) algorithms
- Codebase should contain complete coverage (non-RL) algorithm
- All algorithms will return at least: time taken, energy used, rewards found.
- Script to mask parts of an image and remove all rewards. The image should then be saved and available for use by all algorithms.
- Method of comparing algorithm performance.

1.7.2 Resource Requirements

- Personal computers for all members
- IDE (VSCode)
- Git
- Python and libraries
    - Gymnasium
    - Numpy
    - Pandas
    - Math
    - Random
    - Os
    - Stablebaselines3
    - Cv2 (opencv-python)
    - Threading
    - Sys
    - Time
- GPU's for simulations and training

1.7.3 Environmental Requirements

Simulated search algorithm should attempt to optimize for energy usage. Only applies to a physical drone, which is not in the scope of this project.

1.7.4 Performance Requirements

Algorithms should maximize the number of anomalies found, and minimize energy usage, and time taken for a complete search.

1.7.5 Legal Requirements

Testing with a physical drone will conform with US UAV regulations. The drone will not trespass on private property, or search legally unauthorized areas. Only applies to a physical drone, which is not in the scope of this project.

1.7.6 Maintainability Requirements

Code will be commented and formatted according to []. Code will be written in a modular fashion, allowing for easy expansion in the future. Code will conform to OpenAI's Gymnasium API, allowing for easy addition of new reinforcement learning algorithms.

1.8 Engineering Standards

PEP 8 Style Guide [14]
All Python source code shall follow the PEP 8 Style Guide for Python code for consistency and readability.

Gymnasium Documentation [3]
OpenAI's Gymnasium API for reinforcement learning. Describes the interface for making custom reinforcement learning environments.

Pytorch Documentation [2]
Pytorch API for designing custom deep learning models. Will be used to interface with Stable Baselines 3.

Stable Baselines 3 Documentation [4]
Provides additional reinforcement learning algorithms via Pytorch.

IEEE 1936.1-2021 - Standard for Drone Application Framework [5]
IEEE's general requirements for safe operation of drones in various applications.

Standards and regulations surrounding drones and their applications.
49 USC 44809: Exception for limited recreational operations of unmanned aircraft [6]

US Law describing requirements for legally operating a drone (pilot's license), and regulations surrounding the operation of drones.

1.9 Intended Users and Uses

In a commercial setting, the algorithms from this project could be used by local governments or companies interested in observing/monitoring vast areas. This might include municipalities monitoring a town for roads in disrepair, or a cooperative searching for damaged power lines after a storm.

Larger governmental agencies might use them for search-and-rescue missions after a natural disaster, or even for military surveillance for when satellites aren't an option.

2 Project Plan

2.1 Task Decomposition

A discussion with the client lead to the initial list of tasks:

- Verify Drone Environment
- Object Detection
- Research Reinforcement Learning Algorithms
- Reinforcement Learning Experimentation
- Implementation on Physical Drone

After completion of the first task, verifying the drone environment, the client asked us to instead work on improving the complete coverage algorithm, which is intended to provide a baseline performance score for reinforcement learning. The task proved very challenging and time consuming, and resulted in a major detour from our original schedule (see 2.4.1).

A further discussion with the client led to a revised task list seen in sections 2.1.1 to 2.1.5. This included removing any application of the simulation to a physical drone, due to underestimating the complexity and time requirements of the simulation portion of the project.

2.1.1 Verify Performance of Existing Drone Simulation Environment

Before any experiments with drone search algorithms can be performed, the drone environment simulation needs to be verified. The client requested the following subsystems in particular: reward collection, world padding, and frame resizing of the drone.

2.1.2 Script to Mask Images

An additional task from the client involved making a Python script to mask parts of a generated world effectively removing the rewards from them and shrinking the searchable area.

2.1.3 Improve Upon Existing Complete Coverage Algorithm

In order to validate any sort of performance increase from reinforcement learning, we need to represent the performance of existing work fairly. Research and input from our client suggests current solutions for using drones to search areas involve a complete coverage algorithm.

The existing code from the client allows the drone to search a rectangular area. The client requested that the complete coverage algorithm be able to search a polygon (specified by the image masking script) in an optimal/efficient manner.

2.1.4 Reinforcement Learning Research

Learn the fundamentals of reinforcement learning. Research related applications of reinforcement learning, as well as state of the art of drone search algorithms. Identify relevant performance metrics for comparison. Research the most optimal models/strategies that provide the best results in simulation.

2.1.5 Experiment with Different Reinforcement Learning Algorithms

This is the most significant part of the project. Implement reinforcement learning algorithms and compare their performance to the complete coverage algorithm. The results from this section serve as the findings for this project.

2.1.6 Compiling Final Results

Decisions on best way to compare algorithms. Advantages and disadvantages of certain algorithms. Verifying that chosen metrics fairly represent algorithms. Make sure the algorithms run successfully, and compare them to see what is most optimal in simulation.

2.2 Project Management/Tracking Procedures

We are using Git to update our code to add the new features we decide on with our client every week. We are tracking our progress with the Git commits so everyone understands where we are at on implementing the feature we are working on.

We also use Google Drive to host our shared documents and notes so we all have access to the most updated version of all our documentations. This allows us all to work on a document at the same time and increases our productivity.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

- Functional Complete Coverage algorithm which runs optimally (to some degree) on polygon areas.
- Functional image editor
- List of reinforcement learning algorithms to be implemented
- First reinforcement learning algorithm implementation
- All algorithms run successfully
- Final results from comparisons

2.4 Project Timeline/Schedule

2.4.1 Initial 1st Semester Timeline

Task \ Date	10/20	10/27	11/3	11/10	11/17	11/24	12/1	12/8	12/15	2nd Sem.
Verify Drone Environment										
Object Detection										
Research RL Algorithms										
RL Experimentation										
Implement on Physical Drone										

Figure 1. Initial 1st Semester Timeline

2.4.2 Actual 1st Semester Timeline

Task \ Date	10/20	10/27	11/3	11/10	11/17	11/24	12/1	12/8	12/15	2nd Sem.
Verify Drone Environment										
Improve Complete Coverage Algorithm (Andrew & Cody)										
Image Editor (Cody)										
Research RL Algorithms (Jesse & Akash)										

Figure 2. Actual 1st Semester Timeline

2.4.3 Initial 2nd Semester Timeline

Task \ Date	1/19	1/26	2/2	2/9	2/16	2/23	3/1	3/8	3/15	3/22
Research RL Algorithms	■	■	■	■	■	■	■	■		
RL Experimentation			■	■	■	■	■	■	■	■
Compilation of Final Results										

Task \ Date	3/29	4/5	4/12	4/19	4/26	5/3
Research RL Algorithms						
RL Experimentation	■	■	■			
Compilation of Final Results		■	■	■	■	■

Figure 3. Initial 2nd Semester Timeline

2.5 Risks and Risk Management/Mitigation

2.5.1 Verify Drone Environments

This task should not require a lot of hours, unless something is wrong with the drone environment. Should any major bugs be discovered, no other simulation tasks can continue before these issues are resolved.

2.5.2 Improve Complete Coverage Algorithm

The scope of this could be made very large and consume all of the team’s time until the end of next semester. Design considerations must be made to limit the complexity of this task. See section 3.6.1.

2.5.3 Image Editor

Much like the Complete Coverage Algorithm task, this task’s complexity must be constrained to only do what we need. See section 3.6.2.

2.5.4 Reinforcement Learning Research

The algorithms and models can be quite extensive, so it is important to maintain a strong foundation and a good understanding so the model can run smoothly. As overcomplicating and misunderstanding could potentially hurt the overall progression of the project.

2.6 Personnel Effort Requirements

Task	**Estimated Hours**	**Andrew**	**Jesse**	**Cody**	**Akash**
Verify Drone Environment	20	7	7	7	7
Improve Complete Coverage Algorithm	60	20	10	15	10
Image Editor	20	5	0	10	0
Research RL Algorithms	20	3	15	3	10
Class Deliverables	40	10	10	10	10
Sum	**160**	**45**	**42**	**45**	**37**

Figure 4. Personal Effort Requirements

3 Design

3.1 Design Content

The project requires the design, test, and comparison of multiple algorithms for searching a designated area for target objects. The base/control algorithm is a complete coverage algorithm to exhaustively search the area for anomalies. Other algorithms to be implemented will be different forms of reinforcement learning algorithms to (hypothetically) more efficiently search the designated area.

3.2 Design Complexity

We will be experimenting with different state of the art reinforcement learning algorithms which are exceptionally difficult to understand and implement. To accomplish this, OpenAI's Gymnasium environment class will be extended and customized to most accurately simulate the drone's interactions with its environment.

Another goal of the project is to simulate the current state of the art in industry by implementing a complete coverage algorithm, which does not use reinforcement learning. Hypothetically, using reinforcement learning should result in efficiency improvements greater than the current state of the art.

Many design compromises have been made in almost all aspects of the project in the interests of reducing complexity, limited time, and our team's programming ability. See section 3.6 for more details.

3.3 Modern Engineering Tools

- Github - Industry standard code hosting platform for version control and collaboration
- VSCode - Industry IDE used to edit code
- Discord - Messaging/chatroom application
- Google Drive - Cloud based file storage, allows for collaboration of word processing documents, presentation, and spreadsheets
- Python - Popular and versatile scripting language
- Microsoft Teams - Messaging/chatroom application

3.4 Design Context

Area	**Description**	**Examples**
Public health, safety, and welfare	Our project affects the general well-being of the agricultural industry. Our project affects the general well-being of the people affected by dangerous situations. Our project affects the general well-being of a population suffering from food shortages due to crop decay.	Reduce the exposure to and propagation of plant based diseases in crops. Reduce the time to finding targets in search and rescue missions by automating searching. Increase crop yields by culling infected crops of infection when found.
Global, cultural, and social	Raises ethical concerns over whether employing this technology is net-beneficial to society.	Surveillance jobs normally performed by humans could be replaced with robots. May result in areas being involuntarily surveyed, if used improperly. Could be considered a privacy violation.

Environmental	Is useful for projects spanning large areas which may apply to environmental remediation projects.	Reforestation, chemical clean up, waste management…
Economic	Can replace menial surveillance jobs with automated drones, saving companies money on wages.	Routine surveillance of large-scale infrastructure such as roads, powerlines, buildings…

Figure 5. Design Context

3.5 Prior Work/Solutions

The client demonstrated existing software which plans an optimal complete coverage path for a drone based on an outlined image. Our project intends to go a step further and optimize for the number of anomalies found instead of speed of area coverage. There is sufficient research on finding optimal complete coverage paths for different types of robots (roombas, drones, submarines…). These will be useful baselines to compare our final product against. [8, 10, 11, 12]

3.6 Design Decisions

3.6.1 Complete Coverage Simplification

The original complete coverage algorithm from the client ran perfectly over the entirety of a rectangular image, but when tasked with searching a polygon of area, many more factors were introduced.

The most optimal way to search a polygon would be to follow the longest edge of the shape to reduce the number of turns made by the drone, saving time and energy. Since the unmasked polygon (to be searched) is described by an array of 2D points, it is difficult to start on the longest edge, follow it in a lawn-mower fashion, and guarantee the entire area of the polygon will be explored. Numerous edge-cases can be trivially thought of.

To address this, the team came-up with a much simpler approach which improves complete coverage's performance on most masked shapes, and guarantees that the entire area will be explored, arguably the most important part of the complete coverage algorithm. It works by simply reusing the existing lawn-mower approach from the client, but resizing the rectangle such that it searches the largest rectangle described by the array of points. This works decently well for most convex shapes, but is rather inefficient for concave shapes.

In both figures, the red box depicts the smallest rectangle containing the unmasked image, and the red arrows represent the future path of the drone. In Figure 6., the drone covers a shape rather efficiently, and in Figure 7., the drone has an inefficient path, exhibiting a poor use case for our solution.

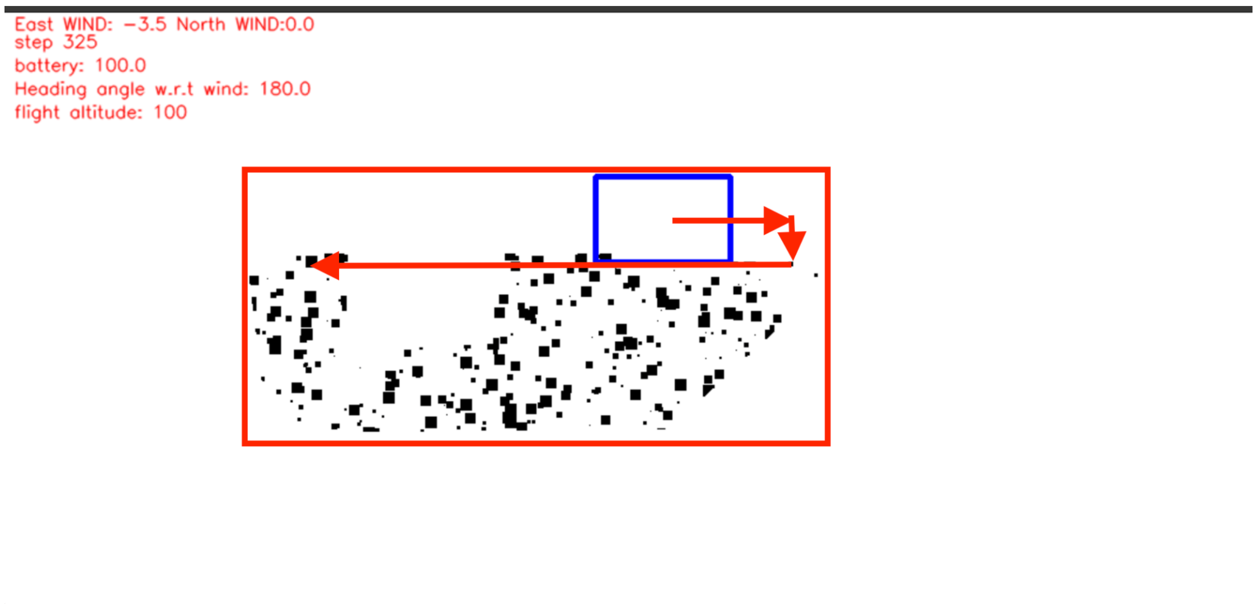

Figure 6. Efficient Complete Coverage Algorithm

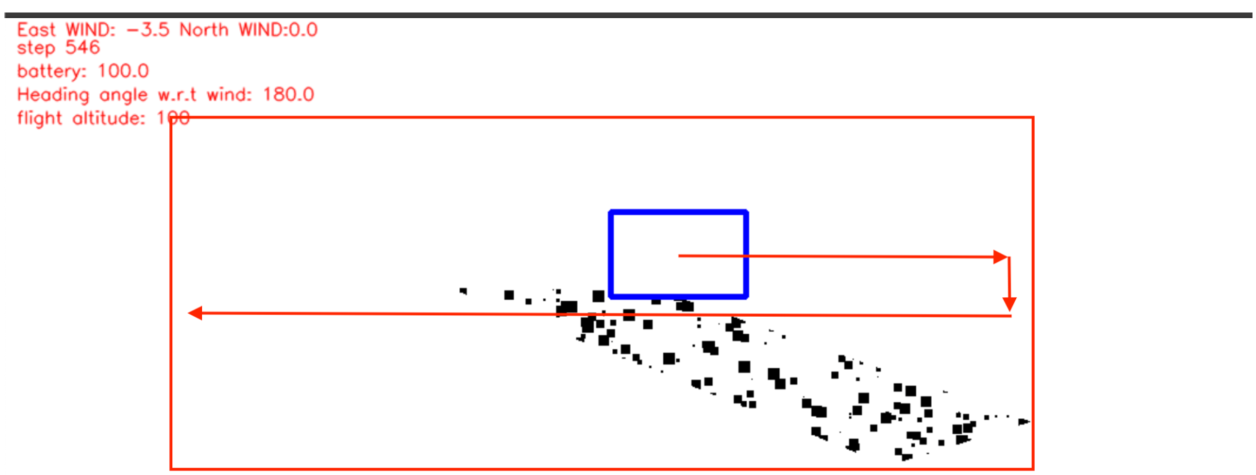

Figure 7. Inefficient Complete Coverage Algorithm

3.6.2 Image Editor Simplification

The image masking script works as expected, but does have a few key limitations. The biggest being that only one area can be outlined, instead of multiple. Not only does this simplify the image masking script itself, but also the search algorithms to be run on the masked image. Storing data about which areas are to be searched is much more difficult when there's multiple separate areas being outlined.

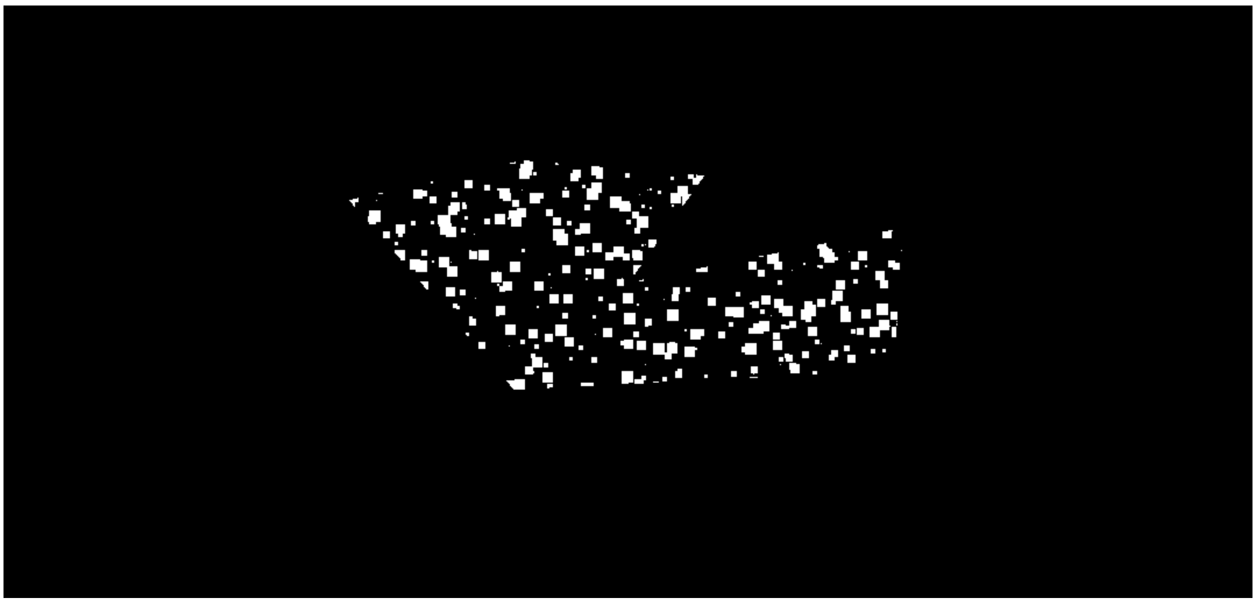

Figure 8. Masked Image

3.6.3 Scope of Project Change

Due to unforeseen complexity and time constraints, the scope of the project has been reduced to finalizing the simulations. This means there will not be an implementation on a physical drone.

3.7 Proposed Design and Design Iterations

A complete coverage algorithm which runs in our simulation with different variables to change such as world size, drone camera FOV (field of view), drone positioning, and image overlap.

Eventually, other standard and state of the art reinforcement learning algorithms will be implemented, with their relative performance to the complete coverage algorithm compared using a few important metrics such as: battery used, time taken, anomalies found…

3.7.1 Design 0 - Client's Existing Code

When initialized, the drone environment pulls information from a configuration file: world area described in a 3d plane as x, y, and z, a wind speed, a battery life, and a number of rewards. The environment runs a world generator function that sets up the world with an area of x by y and a height of z (start and init function). It then populates the ground with a predefined number of seeds.

The code enters into a loop using functions defined by OpenAI Gymnasium and tailored to our project by us to search out the rewards in a complete coverage algorithm. The algorithm functions by starting in an initial location and sweeping right to left with some predefined overlap searching for rewards (reset is called at the end of each step, step being one movement of the drone). When it has found a reward, it will remove it from the map and continue on until it has reached the end of possible movement in the world or has run out of battery life (stop function). The wind speed increases battery life usage if the drone moves against the wind.

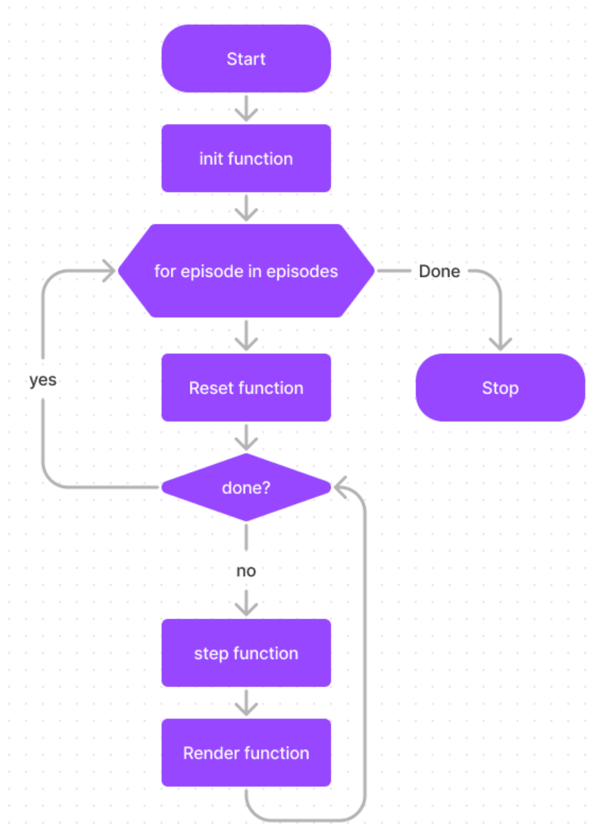

Figure 9. Simulation Flowchart

The current design does not use RL. It has a complete coverage algorithm which will serve as the baseline beat with reinforcement learning algorithms in the future. The user is only able to search for objects by having the drone cover a specified area. This works, but is far from optimal.

3.7.2 Design 1 - Improved Complete Coverage

This design has an improved complete coverage algorithm, called “Complete Coverage Polygon Algorithm” in Figure 10. which can run over masked images in a semi-optimal manner. The polygon shapes come from an image masking script, called “Image Editor Script” in Figure 10.

With these two files, the design is now more versatile than before and has the potential to be much more useful in practice.

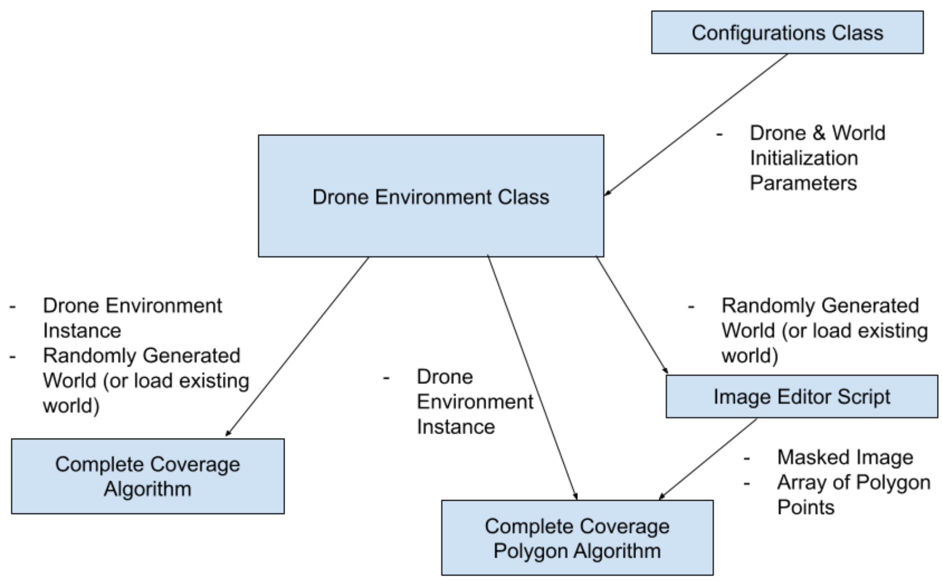

Figure 10. Design 1 File Hierarchy

3.7.3 Design 2 - Reinforcement Learning (Future)

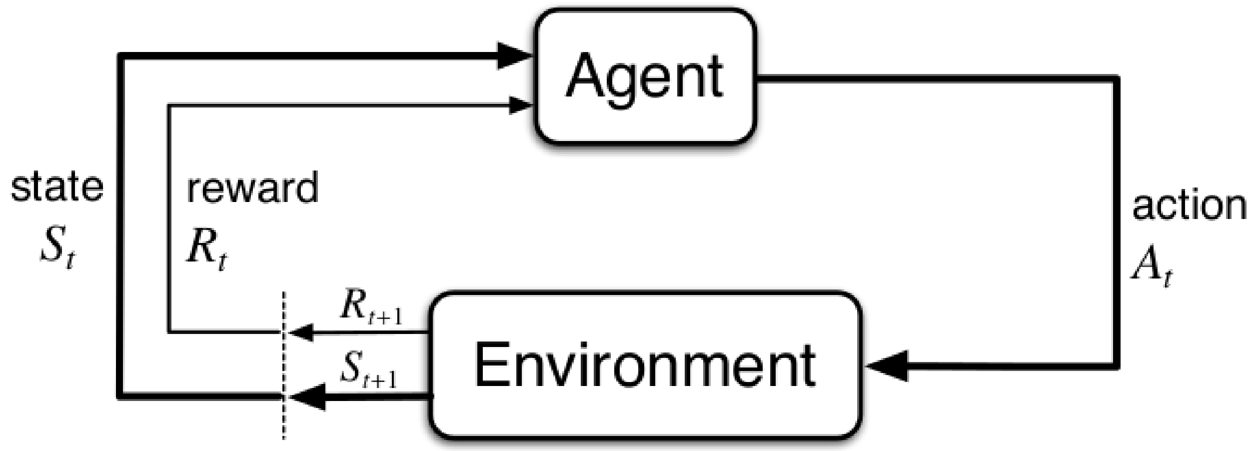

Figure 11. Reinforcement Learning Diagram Overview [7]

This is reinforcement learning from a high-level. In our case, the agent is the drone, the environment is the world it flies in and the ground/objects it sees. The drone interacts with its environment by taking actions which result in a new state and a reward (or penalty). Our drone flies to a new location, and makes an observation of the world (this represents a state). If the drone sees an anomaly in the world, it receives a reward. Everytime the drone moves and uses energy without finding a reward it suffers a slight penalty instead. Using this information, the drone learns which action to take based on the state it's in to receive rewards.

3.8 Technology Considerations

TBD

3.9 Design Analysis

TBD

4 Testing

4.1 Unit Testing

Systems can be isolated and will be tested using the assert function from the "unittest" library. The program will be broken down and tested to ensure the model runs effectively and the requirements are properly met. An additional testing script file will be written to perform these tests. Systems to be tested are:

A. Battery
    - Simulation ends when battery < 0
    - Initialized as a 100
B. Move Cost
    - Equations for calculating cost from drone speed, wind speed, wind angle, drone drag coefficients
C. Movement
    - Drone adheres to overlap percentage
    - Drone's location does not exceed world boundaries
D. Frame Capture
    - Correct resolution
    - Scaled correctly based on height and FoV from camera
E. Reward Collection
    - Does not double count rewards
    - Upscaled rewards not counted more than once
F. World Padding
    - Sufficient as to not cause simulation error
G. World Rendering
    - Rendered window is readable and displays useful information
H. Map Saving/Loading
    - Worlds can be saved and selectively loaded as parameter from command line
    - Saved files are compatible with all algorithms and the image masking script
I. Drone Environment Initialization
    - Drone and world parameters are loaded from a configuration file, are initialized by hardcoded values or overwritten in the environment file.

4.2 Interface Testing

Readme file which describes how to install necessary python packages and how to run each algorithm from the command line. All commands should be tested on Windows, MacOS, and Linux to ensure all packages are compatible and work as intended.

The generated window displaying the simulation should properly show the area being searched, known rewards, unknown rewards, and the drone's search area. This interface can be tested by running a simulation and verifying that the visuals are being properly displayed.

The image masking interface should properly display the world and allow for easy interaction with the mouse. The interface will allow the user to choose the parameters of the searched area by cropping outside of the defined points placed by mouse interaction. The interface will allow for as many points placed as desired and process all points on the press of the enter key. This will be tested in the same manner as the simulation window.

4.3 Integration Testing

The most critical piece of integration is the drone environment's interaction with OpenAI's Gymnasium API. If this is not done properly RL algorithms from OpenAI will not function properly, if at all. The integration of the image masking script will also need to be integrated and tested.

The first challenge will be to get an imported algorithm to run properly. To do this, OpenAI's Gymnasium API will need to be extended correctly by the drone environment class. Once this is complete, the unit tests from section 4.1 will be used to ensure expected functionality of methods in the new algorithm.

The image masking script needs to be able to load saved worlds from a .npy file, edit them as a .png, and then export the results back into a .npy format. To test this, the file can be loaded by an algorithm to verify if file conversion was performed properly.

4.4 System Testing

All systems need to pass all unit tests and integration tests, and be able to run without errors in different scenarios such as: square or polygon shaped world, many/few rewards, and large/small area. Each algorithm should output relevant metrics, formatted for readability, in the terminal and as a file.

4.5 Regression Testing

Additional algorithms that do not require any changes to the drone environment parent class, should not prompt regression testing. In any case where the drone environment class needs to be changed significantly, regression testing should be performed.

Regression testing entails the retesting of unit tests from section 4.1 and integration tests from section 4.3.

4.6 Acceptance Testing

After the team conducts testing as listed in sections 4.1 - 4.5, testing can be performed with the client to ensure requirements are being met. Additionally, all testing will be uploaded to the shared Git repository owned by the client to be viewed and run at their leisure.

4.7 Results

What are the results of your testing? How do they ensure compliance with the requirements? Include figures and tables to explain your testing process better. A summary narrative concluding that your design is as intended is useful.

5 Implementation

Next semester, our team will begin implementing RL Algorithms from OpenAI Gymnasium. After the arduous process of troubleshooting and integrating, the RL algorithms will be compared to the complete coverage algorithm on the basis of energy efficiency, number of anomalies found, and time.

6 Professionalism

6.1 Areas of Responsibility

Contextualizing Professionalism Paper	**IEEE Code of Ethics [13]**
Work Competence	"...maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations…" IEEE encourages its members to continually develop one's skills and to provide the highest quality service possible. It also states that one should know their own limits, and to not take on projects exceeding their capabilities.
Financial Responsibility	Mention of providing products or services at reasonable cost is absent from IEEE's Code of Ethics

Communication Honesty	"...disclose promptly factors that might endanger the public or the environment..." "...seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others…" Working ethically means that all parties should be aware of aspects of a project negatively affecting parties. Provide constructive feedback and gracefully accept criticism.
Health, Safety, Well-Being	"...hold paramount the safety, health, and welfare of the public…" Services for public use should prioritize public users' safety, health and well-being.
Property Ownership	"...seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others…" [] Do not use another's ideas as one's own and give credit where it is due.
Sustainability	"...strive to comply with ethical design and sustainable development practices…"

	Where possible, make environmentally conscious decisions to promote sustainability.
Social Responsibility	"...hold paramount the safety, health, and welfare of the public…" Engineers ought to aim to better society and improve the lives of people through their work.

Figure 12. Comparison of Contextualizing Professionalism and IEEE Code of Ethics

6.2 Project Specific Professional Responsibility Areas

Contextualizing Professionalism Paper	**Application to Project/Our Team's Performance**
Work Competence	Our team does its best to decide on the best possible solutions for problems, then implement them in alignment with guidelines outlined in this document. **Fair**
Financial Responsibility	Our team's work does not contribute to the cost of the project and is not financially compensated. The team may be recognized in a final paper. **Not applicable**
Communication Honesty	Communicating honestly with our client is important since we are effectively assisting with his research. Each week we honestly discuss our team's accomplishments and shortcomings, intending to promote honest communication between both parties. **Good**
Health, Safety, Well-Being	Application of this research could be used for maintaining infrastructure or

	saving lives in search and rescue missions. It could also be used for indirect human harm in military applications. There are applications for this technology to be used for societal good and bad. As junior researchers, we feel there isn't much we can do to influence how this technology will be used. **Not Applicable.**
Property Ownership	When using or citing ideas from sources, our team credits the author with its use. **Excellent**
Sustainability	One goal of the project is to find a better way to perform aerial searches with drones to minimize energy usage. Our team has done a poor job of this so far, due to the fact that we have not fully tested the battery usage portion of the simulation. **Poor**
Social Responsibility	There are concerns that the drone may capture images of an area without the property owner's consent. We have begun to investigate this through the use of the image masking script. Although the complete coverage algorithm still takes pictures of the masked image currently (as of December 2023). **Fair**

Figure 13. Contextualizing Professionalism Applied to the Team

6.3 Most Applicable Professional Responsibility Area

The area our team has the chance to act most professionally in is Work Competence. Our team's main contribution is to our client's research [11], which means high quality work should be prioritized. Honest Communication is note-worthy as well, seeing as we are working closely with our client to accomplish tasks.

7 Closing Material

7.1 Discussion

N/A

7.2 Conclusion

N/A

7.3 References

[1] Python, "The Python Standard Library — Python 3.8.1 documentation," *Python.org*, 2020. https://docs.python.org/3/library/index.html

[2] "torch — PyTorch 1.12 documentation," *pytorch.org*. https://pytorch.org/docs/stable/torch.html

[3] "Gymnasium Documentation," *gymnasium.farama.org*. https://gymnasium.farama.org/index.html

[4] "Stable-Baselines3 Docs - Reliable Reinforcement Learning Implementations — Stable Baselines3 1.2.0a2 documentation," *stable-baselines3.readthedocs.io*. https://stable-baselines3.readthedocs.io/en/master/

[5] "IEEE SA - IEEE 1936.1-2021," *IEEE Standards Association*. https://standards.ieee.org/ieee/1936.1/7455/

[6] "[USC02] 49 USC 44809: Exception for limited recreational operations of unmanned aircraft," *House.gov*, 2018. https://uscode.house.gov/view.xhtml?req=granuleid:USC-prelim-title49-section 44809&num=0&edition=prelim

[7] *Oreilly.com*, 2023. https://www.oreilly.com/radar/wp-content/uploads/sites/3/2019/06/image3-5f8cbb1fb6fb9132fef76b13b8687bfc.png (accessed Dec. 03, 2023).

[8] T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira Jr. , "Survey on Coverage Path Planning with Unmanned Aerial Vehicles," *Drones*, vol. 3, no. 1, p. 4, Mar. 2019, doi: https://doi.org/10.3390/drones3010004.

[9] "Overlap & Flight Pattern | Drone Data Processing," *Aerotas: Drone Data Processing for Surveyors*. https://www.aerotas.com/overlap-flight-pattern

[10] E. Galceran and M. Carreras, "A survey on coverage path planning for Robotics," A survey on coverage path planning for robotics, https://www.sciencedirect.com/science/article/abs/pii/S092188901300167X (accessed Oct. 19, 2023).

[11] A. Niaraki, J. Roghair, and A. Jannesari, "Visual exploration and energy-aware path planning via reinforcement learning," arXiv.org, https://arxiv.org/abs/1909.12217 (accessed Oct. 19, 2023).

[12] J. Shi and M. Zhou, "A data-driven intermittent online coverage path planning method for AUV-based bathymetric mapping," MDPI, https://www.mdpi.com/2076-3417/10/19/6688 (accessed Oct. 19, 2023).

[13] IEEE, "IEEE Code of Ethics," ieee.org, Jun. 2020. https://www.ieee.org/about/corporate/governance/p7-8.html

[14] G. van Rossum, B. Warsaw, and N. Coghlan, "PEP 8 – Style Guide for Python Code | peps.python.org," peps.python.org, Jul. 05, 2001. https://peps.python.org/pep-0008/

7.4 Appendices

N/A

7.5 Team Contract

Team Name: RLPIIUAV

Team Members:

1). Jesse Gillingham

2). Andrew Sailer

3). Akash Setti

4). Cody Draper

Team Procedures:

1. Meetings: Thursdays 7pm, Face-to-Face in SICTR 4201 if possible. Weekly.
2. Communication: Discord and Email.
3. Decisions: Majority Vote or Compromise.
4. Record Keeping: Andrew Sailer will take meeting notes and post them in #meeting-minutes channel in discord. Other relevant documents will be posted in #resources.

Participation Expectations:

1. Expected weekly meetings with mandatory attendance, along with day to day communication through the discord involving the project.
2. Each member will be assigned work for the period of time set in the meeting. Each team member is to finish their section of work in that assigned time. If the team member cannot finish in the allotted time they must inform the rest of the team.
3. Each team member is expected to be very communicative regarding their work
4. Decisions will be posted in Discord for the entire team, and will be respected afterward.

Leadership:

1. Leadership roles for each team member:

We all hold equal responsibility for keeping each other accountable.

2. Strategies for supporting and guiding the work of all team members:

Agile Project Management - discuss what we worked on in weekly meetings, problems we're having, and what we plan to work on in the future.

3. Strategies for recognizing the contributions of all team members:

GitLab will be involved in the project. Based on the commits, the team will be able to see the work for the specific team member.

Collaboration and Inclusion:

1. Jesse Gillingham: Hardware, Embedded Systems,

Andrew Sailer: Hardware, Control Systems, Some Programming.

Akash Setti: Software Development, Data Science, Machine Learning

Cody Draper: Software Development, Embedded System, Machine Learning

2. Strategies for encouraging contributions and ideas from all team members: We will have group meetings and discussions at our major points.
3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

Goal-Setting, Planning, and Execution:

1. Team goals for this semester: Get familiarized with the code base. Set up the development environment and train RL models. Experiment with different RL Algorithms using stable baseline3.
2. Strategies for planning and assigning individual and team work: Assign work to team members during team meetings.
3. Strategies for keeping on task: Use Gitlab issues and milestones to keep track of timelines and maintain a steady flow of work.

Consequences for Not Adhering to Team Contract:

1. How will you handle infractions of any of the obligations of this team contract?

Any infractions will result in a team vote on the specific consequences.

2. What will your team do if the infractions continue?

We will inform the person who determines the grades.

A. I participated in formulating the standards, roles, and procedures as stated in this contract.
B. I understand that I am obligated to abide by these terms and conditions.
C. I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

1). Jesse Gillingham Date: 9/7/23

2). Andrew Sailer Date: 9/7/23

3). Akash Setti Date: 9/7/23

4). Cody Draper Date: 9/7/23