CyDrone

Design Document

Team sdmay20-47

Client: Ali Jannesari
Advisors: Ali Jannesari

Team Leader: Kenneth Lange,
Chief Hardware Developer: Alain Njipwo,
Chief Software Developer: Daniil Olshanskyi,
Chief Interface Developer: Luke Bell,
Chief Backend Developer: Max Medberry

E-Mail: sdmay20-47@iastate.edu
Website: https://sdmay20-47.sd.ece.iastate.edu

Revised: 10/06/2019 - v1

Executive Summary

Development Standards & Practices Used

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

Summary of Requirements

List all requirements as bullet points in brief.

Applicable Courses from Iowa State University Curriculum

List all Iowa State University courses whose contents were applicable to your project.

New Skills/Knowledge acquired that was not taught in courses

List all new skills/knowledge that your team acquired which was not part of your Iowa State curriculum in order to complete this project.

Table of Contents

List of figures/tables/symbols/definitions (This should be similar to the project plan)

1 Introduction

1.1 Acknowledgement

Huge thanks to Dr. Ali Jannesari for mentoring us on this project.

1.2 Problem and Project Statement

Nowadays, all aspects of human lives are moving towards automation. This has been true for as long as humanity has existed, but with the development of computers and growth of machine learning, more and more operations can be efficiently done by computers. Since we, humans, are receiving a big portion of information via our eyes, it's logical for us to try and teach our machines in a similar way; to see and process what they see, making decisions based on that.

This is why we decided to work with this problem. Our goal is to make a platform with high mobility (a drone), load it with exceptional computational powers (nvidia SoM from the Jetson family), and make it process a camera input with machine learning algorithms. As a starting point in our machine learning we want to teach the drone to: identify objects, follow or avoid them, and analyze them (including volumetric analysis). Further development will be done on creating more complex algorithms based on computer vision and machine learning, like adding the possibility of a master-slave system, where a fleet of drones are led by a single

We are hoping to build a drone-based computational system capable of solving complex tasks via machine-learning and computer-vision based algorithms. Object detection and tracking should serve as examples of how our system can handle complex machine-vision and machine-learning based operations while our architecture and implemented control functionalities will allow to use and expand the solution with ease.

1.3 Operational Environment

The operational environment on the software side is pretty simple: front end will be launched from the browser so anything supporting javascript, html5 and webgl is sufficient. The back end will be launched on the drone itself, so it will represent the server, the computer-vision analyzer and the drone controller at the same time. On the other hand, our hardware - the drone itself - should be able to withstand any environmental influences that a drone can encounter. Obviously, the balancing algorithms should be present to stabilize the machine mid-air and negate the windflow. In general we don't plan to use the drone during the rain so the water protection should be present, but only to cover from the occasional moist and small splashes. The protection should also cover the vulnerable electronics from small debris such as leaves and others that a drone can encounter in the air at normal conditions.

1.4 Requirements

Functional requirements:

- Incorporate information about old iterations of the project received from the client.
- Set up a simulation environment to test the software.
- Set up a website to see the drone/simulation camera stream
- Implement object detection and compute the volumetric analysis.

- The hardware will be Master-Slave oriented so that new hardware can be added and removed as desired in the future.
- Assemble the drone itself.

Non-functional requirements:

- All software should store logs of the past important information and crash reports.
- Software will have an architecture that is easy to understand and navigate.
- Code will be well documented, both via comments and a wiki explaining to future users what is being accessed by what and how.
- Code will be modular, so it is easy to fix, extend, and/or replace.
- Open source software will be used to make the process cheaper and more maintainable using the help of an open source community.

1.5 Intended Users and Uses

We intend the product to be used by big companies as well as small enthusiasts to rely tasks on the drones. The system should allow anyone to use the object detection and volumetric analysis functionality with the help of our platform. We also see other developers as our user, so it is sufficient that we make our program modular (with the help of ROS architecture approach) and expandable. Third-party developers will be able to use our system to perform their needed tasks as well as an opportunity to extend our hardware and software solutions to fit their needs. We shall provide basic instruments as well as examples of useful algorithms so the system has a good performance as well as a high level of flexibility.

1.6 Assumptions and Limitations

Assumptions:

- Client/end user is familiar with operating a drone
- The drone will be utilized for object volumetric calculations.
- The drone is used where it can be wirelessly connected to the network

Limitations:

- The hardware might struggle to accurately record parameters for thinner objects.
- Input Noise might impact some of recorded measurements.
- Limited to use ROS on the drone.

We are assuming that the client/end user will be familiar with the basic operations of a drone. We are also assuming that the drone will be operated as intended. To calculate the volume of any solid integral objects. We assume that the drone will be used in weather conditions that will not compromise the hardware's ability to accurately predict the size of objects.

The volume calculations will be limited to the hardware's capability to translate analog values to digital values. In an extreme case like a leaf where the length and width can be easily calculated the hardware will struggle to detect the dept. Cases like these should be considered and a possibility of a greater margin of error. The decrease in accuracy should be accounted for when the product is utilized.

1.7 Expected End Product and Deliverables

At the end of this project, our team should deliver a fully assembled drone that has unmanned automated vehicle (UAV) capabilities. Attached to the drone will be other hardware components that are needed to enable to drone to perform the required project function. There will be a GPS module to enable the drone to be tracked and relay it's position in real time. A camera to capture the data from the objects of interests. A GPU to directly process some of the data onboard.

There will be communication will the backend of the project. Hosted on a server, will be Artificial Intelligence portion of the software that will be able to communicate with the drone in real time during its flight. Also hosted remotely will be the algorithms necessary to calculate the volume of each object encountered.

For visuals, we will boundary test cases of objects that have been scanned by the drone as well as the obtained volume.We will manually do these same volumetric calculation. From both obtained results, we will do a percent error calculation to give an estimate of the expected accuracy of the drone.

We will include a video demonstrating the drone starting, taking flight and calculating the volume of a test object.

2. Specifications and Analysis

2.1 Proposed Design

This project was given to us by the client with an already-established approach to solving the problem. A previous senior design team was working on the same project, and they ran into a lot of issues out of their control that rendered them unable to fully complete the project. Having learned from those issues, our client did some research before giving us the project for how we should best go about implementing the solution to their problem. They identified multiple technologies for us to use: ROS as an industry-standard operating system for controlling robotic systems like drones, Gazebo as a suitable simulation environment for flying the drone, and Hector Quadrotor as a useful interface for sending the drone commands through the CyDrone web interface. Using these technologies will allow us to remotely operate the drone, both in simulations and the real world, and run machine learning algorithms on the web server that will control the simulated drone.

2.2 Design Analysis

So far, our work has been focused on: construction and calibration of the drone, setup of the Gazebo simulation environment, and operation of the CyDrone website. While doing all of this, we've been attempting to work with the results of a previous group's senior design project. That team had a lot of issues getting the different technologies to work together reliably, so after looking through their work and attempting to use it, we decided it would be more efficient if we reworked some aspects of their project, using their results to identify how we could implement the different components most effectively. So far, we've been able to do this successfully, making progress in areas where the other team was stuck. We'll keep doing this going forward then, and our next steps are to establish communications between the CyDrone website and the simulation environment and get the drone flying using commands sent through ROS.

2.3 Development Process

Our team is following a one-week Agile development process. We have a weekly meeting with our client where we present the work we've completed over the previous week and identify what tasks we will work on in the coming week. This process works well for our team and our client, as we meet frequently enough to continuously identify tasks to work on and are able to make solid progress on our project, but not so frequently that we can't find time in our schedules to meet or work on the project.

2.4 Design Plan

We want to create a modular, reusable system for current and future development of our volumetric analysis drone project. To do this, we will create a model system using Gazebo and ROS to simulate an environment for the autonomous drone with an image processing ability. This will be used to prevent crashes and allow for algorithms, specifically collision-avoidance machine learning algorithms, to learn from the flight patterns. On our real-world drone, we would like the user to be able to see from the drone's view. We plan on setting up a web-based interface that the user can see these images from. One of our core technical goals is to allow the drone to have an object detection functionality. This is what the volumetric analysis will be built upon. We plan on streaming the camera footage to our algorithm so that it can calculate the volume of an object while mid-flight.

3. Statement of Work

3.1 Previous Work And Literature

Include relevant background/literature review for the project

– If similar products exist in the market, describe what has already been done

– If you are following previous work, cite that and discuss the **advantages/shortcomings**

– Note that while you are not expected to "compete" with other existing products / research groups, you should be able to differentiate your project from what is available

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

3.2 Technology Considerations

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

3.3 Task Decomposition

In order to solve the problem at hand, it helps to decompose it into multiple tasks and to understand interdependence among tasks.

3.4 Possible Risks And Risk Management

Include any concerns or details that may slow or hinder your plan as it is now. These may include anything to do with costs, materials, equipment, knowledge of area, accuracy issues, etc.

3.5 Project Proposed Milestones and Evaluation Criteria

What are some key milestones in your proposed project? Consider developing task-wise milestones. What tests will your group perform to confirm it works?

3.6 Project Tracking Procedures

What will your group use to track progress throughout the course of this and next semester?

3.7 Expected Results and Validation

What is the desired outcome?

How will you confirm that your solutions work at a **High level**?

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

• A realistic, well-planned schedule is an essential component of every well-planned project

• Most scheduling errors occur as a result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

– Start with a Gantt chart showing the tasks (that you developed in 3.3) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.

– Annotate the Gantt chart with when each project deliverable will be delivered

• Completely compatible with an Agile development cycle if that's your thing

How would you plan for the project to be completed in two semesters? Represent with appropriate charts and tables or other means.

Make sure to include at least a couple paragraphs discussing the timeline and why it is being proposed. Include details that distinguish between design details for present project version and later stages of the project.

4.2 Feasibility Assessment

Realistic projection of what the project will be. State foreseen challenges of the project.

4.3 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be based on the projected effort required to perform the task correctly and not just "X" hours per week for the number of weeks that the task is active

4.4 Other Resource Requirements

Identify the other resources aside from financial, such as parts and materials that are required to conduct the project.

4.5 Financial Requirements

If relevant, include the total financial resources required to conduct the project.

5. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)
2. Define the individual items to be tested
3. Define, design, and develop the actual test cases
4. Determine the anticipated test results for each test case 5. Perform the actual tests
6. Evaluate the actual test results
7. Make the necessary changes to the product being tested 8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

5.1 Interface Specifications

– Discuss any hardware/software interfacing that you are working on for testing your project

5.2 Hardware and software

– Indicate any hardware and/or software used in the testing phase

– Provide brief, simple introductions for each to explain the usefulness of each

5.3 Functional Testing

Examples include unit, integration, system, acceptance testing

5.4 Non-Functional Testing

Testing for performance, security, usability, compatibility

5.5 Process

– Explain how each method indicated in Section 2 was tested

– Flow diagram of the process if applicable (should be for most projects)

5.6 Results

– List and explain any and all results obtained so far during the testing phase

- – Include failures and successes
- – Explain what you learned and how you are planning to change it as you progress with your project
- – If you are including figures, please include captions and cite it in the text

• This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place

-**Modeling and Simulation**: This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.

-List the **implementation Issues and Challenges**.

6. Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly reiterate your goals. Then, reiterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.