UAV-Based Visual SLAM Platform

Final Report

by

Deqiang Xu, Deqiang.Xu001@umb.edu, EE

Carlos J Orta, Carlos.Orta001@umb.edu, EE

Lucas Lomba, Lucas.Lomba001@umb.edu,CE

Technical Advisor: Dr. Honggang Zhang

May 27, 2020
UMass Boston Engineering
ENGIN 492 – Senior Design Project II

Table of Contents

1. Introduction 3

1.1 Project Synopsis 3

2. Background and Problem Statement 4

2.1 Existing Works 4

2.2 Global and Societal Context 7

2.3 Environmental and Public Welfare Impact 7

2.4 Economic Factors 7

2.5 Relevant Engineering Standards 8

3. Design and Methods 10

3.1 Specifications and Requirements 11

3.2 Design Goals 11

Goal 1: Functioning Big Drone 11

Goal 2: Real-time Visual SLAM system and Object Detection 12

Goal 3: Software programming 12

Goal 4: System integration and evaluation 13

3.3 Deliverables 13

3.4 Design 13

Goal 1: Functioning Big Drone 13

Goal 2: Real-time Visual SLAM system and Object detection 14

Goal 3: Software Programming 15

Goal 4: System integration and evaluation 16

3.5 Design Alternative 18

3.5.1 Hardware Combination Alternative 18

3.5.2 Edge Server 18

3.6 Milestones 18

3.7 Gantt Chart 20

4. Result 21

4.1 Drone and Flight Test 21

4.2 SLAM Result 22

4.3 Object Detection Result 23

4.4 Final Budge 24

5. Bibliography 25

1. Introduction

1.1 Project Synopsis

Simultaneous Localization And Mapping (SLAM) is a technology for constructing a map of an unknown environment while keeping track of the location of a moving robot based on the information collected from sensors. Visual SLAM mainly uses one or more cameras as the sensors to complete the SLAM tasks [1][2]. The resulting map can be in 2D or 3D based on the type of utilized sensors and the applied SLAM algorithm, and a map includes the sparse or dense information of explored spaces that is essential for providing localization and navigation services to robots or other agents.

Dr. Honggang Zhang in the Engineering Department, University of Massachusetts, Boston, would like to have an Unmanned Aerial Vehicle (UAV)-based real-time Visual Simultaneous Localization and Mapping (SLAM) platform for his laboratory. The system we proposed contains a big drone with an onboard camera for taking images of its surroundings and a computing platform to construct a map of the environment based on the SLAM algorithm. It can also provide navigation services for other smaller drones through its low power wireless communication modules. Our platform is the integration of flight control, SLAM algorithm, object detection algorithm, wireless communication modules, and power monitoring system, which features a high degree of hardware/software flexibility/modularity and system extendibility, and it is suitable for conducting further visual-orientated research on UAV.

2. Background and Problem Statement

Simultaneous Localization And Mapping (SLAM) is one of Dr. Zhang's research interests and his laboratory has conducted some research on classic SLAM algorithms such as the FastSLAM with UAV. He would like to have a UAV-based platform with a state-of-art SLAM algorithm that allows his students to work on the Visual SLAM aspect. The visual SLAM algorithms have been studied for a few decades and many of them have demonstrated their feasibility on robots. With the maturing of SLAM algorithms and some successful applications, the investigating direction of SLAM moves more toward sensor fusion, specialization for Embedded system, and semantic SLAM that takes advantage of the deep neural networks. These require a system with a high degree of system extendibility and modularity so that it can be flexible to investigate performance improvement with varied types of sensors and software.

2.1 Existing Works

Simultaneous Localization And Mapping

The mainstream visual SLAM algorithms have been proposed since the 2000s and many of them have reached satisfying estimations and reconstructing results. While the prevailing open source SLAM algorithms all have different characteristics from approaching methods, map density, to the camera types, and whether using Robotic Operating System (ROS). For example, the ORB-SLAM uses a feature-based method and creates a sparse map with a monocular camera. While the DTAM algorithm uses the direct method and creates a dense map with the RGB-D camera [1]. Since there is no universal answer for the SLAM algorithm, it is a must to choose an appropriate SLAM algorithm based on the client's requirements and interests. As Dr. Zhang has stereo cameras, we would mostly focus on the SLAM algorithms that could use this type of camera, such as the RTAP-MAP.

To the best of our knowledge, ORB-SLAM2 [3] suits our project the best among all those open source Visual SLAM algorithms. ORB-SLAM2 system builds upon the ORB-SLAM which is an ORB feature-based SLAM system with a monocular camera for sparse mapping of the surrounding environment in real-time. The performance of ORB-SLAM is degraded due to scale drift and potential failures when doing pure rotation. By applying a stereo camera or RGB-D camera, ORB-SLAM2 solves those issues and provides the relocation and map reuse services. It mainly composes three threads when operating the system: tracking of the camera movement and pose, local mapping, and loop closing for global map optimization [3].

UAV platform and the Wireless communication between Drones

The Crazyflie [4] is a lightweight, open-source hardware and software platform based on a small quadcopter. It is developed by Bitcraze, and it provides a broad and versatile platform that allows rapid development to be made on top of it. Research and development with the Crazyflie platform are possible since it provides various expansions for it, allowing it to increase the functionality and performance with minimal decrease in flight time. These expansions called Expansion decks can add wide functions from obstacle avoidance, known as OA Deck to pre-programmed flight paths through the Flow Deck. When adding the expansions, the increment in weight and power must be taken into consideration. It is possible to expand the Crazyflie to allow for heavier motors through the BigQuad Deck. The Crazyflie communicates through a 2.4 GHz radio USB dongle that allows for control implementation at the user's device. Since the Crazyflie has been heavily used in Dr. Zhang's laboratory and it could be extended to a bigger drone, we would build our big drone based on the Crazyflie system to reuse those hardware resources and take advantage of the previous design experiences.

Wireless communication has become a necessary part of various systems. With a wide range of choices, this system focuses on low power consumption as its key aspect. Bluetooth Low Energy (BLE) is well known for its low energy consumption, able to run for days to months on a single cell battery and its small size profile. It is a common technology that is well documented and seen throughout various industries. LoRa (Long Range) is another low power wireless communication that can transmit in a much longer range. BLE has a theoretical max range of ~350 meters within line-of-sigh, but a more operating range of closer to ~30 to 50 meters [5] while LoRa is able to reach ranges of ~10km [6], depending on area and blockage. Some students in Dr. Zhang's lab built a Bluetooth Deck with an NRF52832 SoC for the communication between the Crazyflies in their previous projects. To match their previous work, we would also use the Bluetooth Low Energy (BLE) protocol for the short-range and effective communication between drones. While considering the possible requirement of long-range communication in some practical applications, we would add the LoRa module to our system as well.

Power Management System

Lithium-ion batteries will be used to power the entire system as the lithium batteries have the highest energy densities of any other batteries in the current market [7]. Being able to power the system efficiently so that the visual SLAM algorithm can be done onboard while in flight, will be a challenge that should be considered. The battery management system is widely used in the car industry, which could provide information about the voltage level, state of charge, current flow, etc. It might also provide a battery protection mechanism. While in our system, we only need to provide the user the information about the current flow (power consumption) to each individual function block and the battery voltage level instead of all those features. A power management integrated circuit (PMIC) could be used to fulfill our requirements. A PMIC is a circuit that can monitor power consumption and distribution, and the efficiency of a battery can be doubled or tripled by managing how the power is directed to the different components in a system. [8]. With the information on power usage and battery level, our drone could perform a safe landing to avoid potential damages due to low battery.

All-in-one Solution

HydraOne, a platform for experimental research built in [9] provides us an example of an all-in-one solution. It is a robotics car with a computing platform, onboard sensors including cameras and a Lidar, and the actuation system, which utilizes the Robotic Operating System (ROS) [10] framework to efficiently manage all resources of their system. The Robot Operating System is a flexible framework for writing robot software. It collects robot-related tools, libraries, and conventions so that it can simplify the task of creating a complex and robust robot across a wide variety of robotic platforms [10]. It provides powerful developer tools and could integrate from drivers to state-of-the-art algorithms. In the HydraOne, each part of the system is treated as a ROS node, for instance, the camera part is treated as a node that would implement the functionality of a camera driver to stream the frames from the camera for the SLAM tasks. Every node can perform their own tasks and communicate with each other efficiently on ROS. Our project would require exactly such a framework to provide a well-modularized system that consists of the SLAM system, actuator unit, onboard sensors, and communication modules.

2.2 Global and Societal Context

Our open source UAV-based platform would make its contribution to the UAV and SLAM communities, allowing people who are interested in these areas around the world to use it for their research purpose. Our drone can also serve as a prototype of building expanded drone based on the Crazyflie system so that they can build an even bigger drone referencing our design and building experience. A bigger drone could be expanded to other areas such as agriculture, law enforcement, and infrastructure maintenance. Using UAV to take images of their crops, farmers can examine the images and decided which crops were ready to harvest. Drones within law enforcement could be used to investigate crime scenes and assist other agencies for disaster relief [11]. They can be equipped with thermal cameras to survey or detect when the vision field is constrained [12].

2.3 Environmental and Public Welfare Impact

Enhancing the aforementioned areas above can have a significant impact on public safety. Weeds can be a problem for harvesting crops, so drones can provide farmers with an advantage. Drones can improve weed detection which can save crops for harvesting [13]. Enhancing the work of law enforcement and infrastructure maintenance can greatly increase safety for all individuals who rely on those services. Building and bridges could be kept safe by quickly detecting any problems before they become critically damning to the structure. Avoiding disaster for any party involved. Drones have been shown in some cases to aid law enforcement in areas such as disaster relief and accident reconstruction [11]. The mapping capabilities of the drone system introduced in this proposal will provide the system with the potential to be implemented in these areas. Using the SLAM software, the drone could map out areas of crops for a farmer, or potentially provide law enforcement with aerial photography for scenarios of search and rescue or disaster relief.

2.4 Economic Factors

All the factors mentioned in the previous section lead to the fact that by using smarter technology in these fields, work can be completed more efficiently which not only benefits the outcome of the job but also reduces the cost. By finding the correct crops to harvest quicker, farmers spend less time and resources, reducing costs all around for the farmer and the consumer. Law enforcement can spend less time enforcing the law and that equates to fewer taxpayers’ money being used. And, just for the system itself, if our low-cost system were to be accessible to other agencies or schools, it would reduce the cost of building such a platform.

2.5 Relevant Engineering Standards

The state of Massachusetts does not have any strict laws against the operation of drones, apart from the town of Newton. UMass Boston is situated in the city of Dorchester which has no restriction on drones at the state level. However, federal law must be observed due to the proximity of the university to Boston Logan Airport. The National Airspace System (NAS) classifies the University of Massachusetts Boston as being located in Airspace Class B. Airspace class B, which is airspace surrounding a busy airport in the United State, will require the operator to ask for clearance from the Air Traffic Control(ATC) to operate the drone.

This drone system would need to meet Federal Aviation Administration (FAA) standards. Title 14 of the Code of Federal Regulations states that if a person wishes to operate an Unmanned Aircraft, the operator will need to acquire a license to operate a said drone. The drone itself will then have to be registered with the FAA in accords with part 107 of the FAA regulations. The FAA requires by part 107 as well that any aircraft in between .55 lbs. and 55lbs be registered as an aircraft [14].

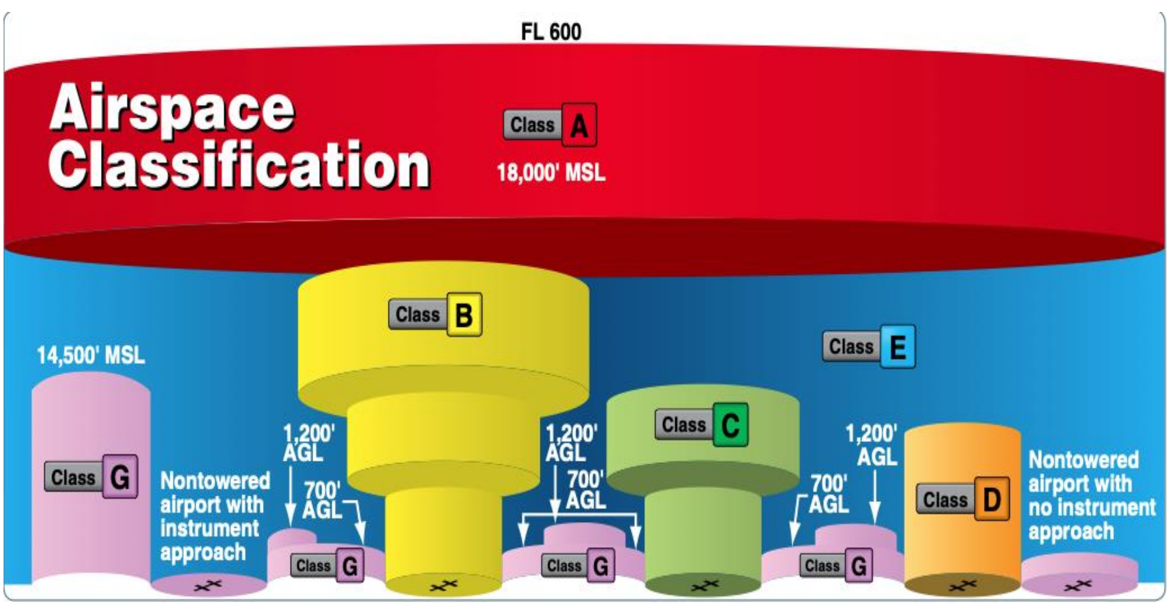


Figure 1. Airspace classification according to the NAS. [14]

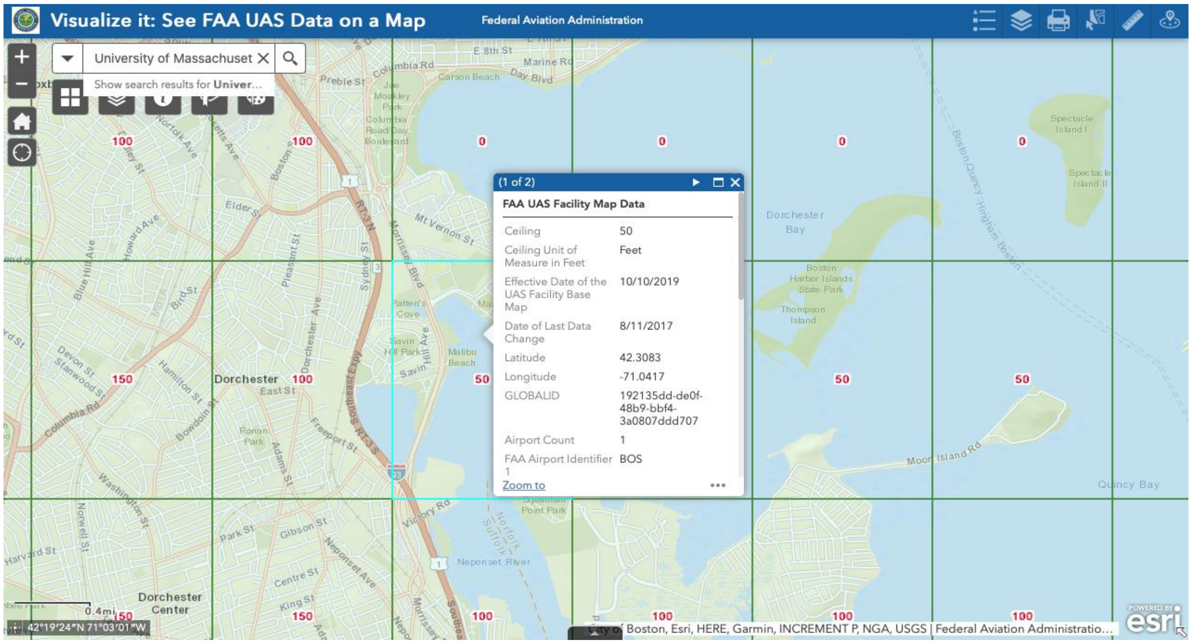


Figure 2. Map of airspace surrounding UMass Boston. [15]

3. Design and Methods

To finish this challenging embedded project, we have formed a team of three with specialized skills in both hardware and software.

Considering the modular design requirements on both the hardware and software, the system stability, and the requirement of the system extensibility, we have determined to choose the ROS framework to integrate all software and manage all hardware resources in our system. Regarding the computing platform of our system, the Nvidia Jetson Nano has been selected for our system for its balance of performance, cost, and power. The 40 General Purpose Input/Output (GPIO) pins and multiple ports enable the connections to multiple onboard hardware components through varied types of communication protocols. Its CPU and GPU combination provides sufficient computing power that allows it to perform real-time visual SLAM and object detection tasks while only consuming 5 watts. As the brain of our system, the Jetson Nano uses the incoming data from the stereo camera and other sensors to perform several computing and processing tasks in real-time, and issue motion commands to the Crazyflie 2.1 or the Crazyflie swam through its communication modules. In addition to the consideration of flight control, wireless communication and power monitoring, the hardware design of our system is shown in figure 3.

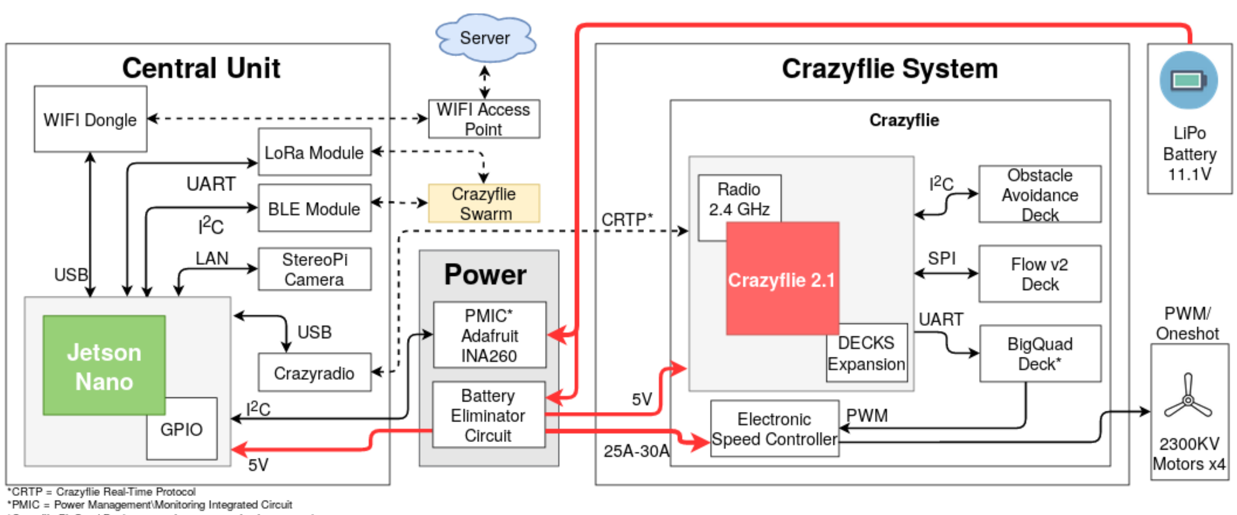


Figure 3. The hardware design of the system

3.1 Specifications and Requirements

Table 1. Requirements Matrix

Requirement	Note	Verification
Functioning Big Drone	• Manual/Autonomous Flight • BLE/LoRa communication	Running a test script with flying commands and flight controlling with game controllers.
Visual Tasks	• Camera Driver • Camera Calibration • Visual SLAM system • Object Detection	Feeding test datasets to the SLAM and object detection system
Software programming	• BLE/LoRa ROS driver • Crazyflie Controller • PMIC ROS driver	Monitoring errors when software running individually.
system Integration and evaluation	• Hardware and Software integration	checking whether each part runs without any system failure.

3.2 Design Goals

Goal 1: Functioning Big Drone

Our system required a drone with large load capacity to load all the hardware resources shown in figure 3. Building on top of the Crazyflie platform, we have chosen and assembled a drone with a bigger frame, higher power motors, Electronic Speed Control (ESC) for driving the motors, propellers with suitable size, and big battery. Custom development for the existing

BigQuad Deck and Flow Deck and from Bitcraze were made to achieve autonomous flying. The development of Bluetooth module and LoRa module were also required for the communication purpose between drones.

Goal 2: Real-time Visual SLAM system and Object Detection

Even though we have chosen the ORB-SLAM2 system and the darknet-based YOLO object detection algorithm for our system considering the real-time requirement, we have tested them with some datasets generated by our camera to ensure their feasibility to our system. Some modification of parameters and optimization have been made to reach the best performance.

Goal 3: Software programming

The ROS has been installed to the Jetson Nano board and figure 4 shows the ROS framework of our system. All the code pieces in our system should be well-separated into different ROS nodes based on their functionalities. Except the off-the-shelf ORB-SLAM2, Darknet, and camera driver packages, we have programmed other software including the Crazyflie ROS controller, BLE module ROS driver, LoRa ROS driver, and PMIC ROS driver. Each node would communicate to each other through the ROS internal communication protocol.

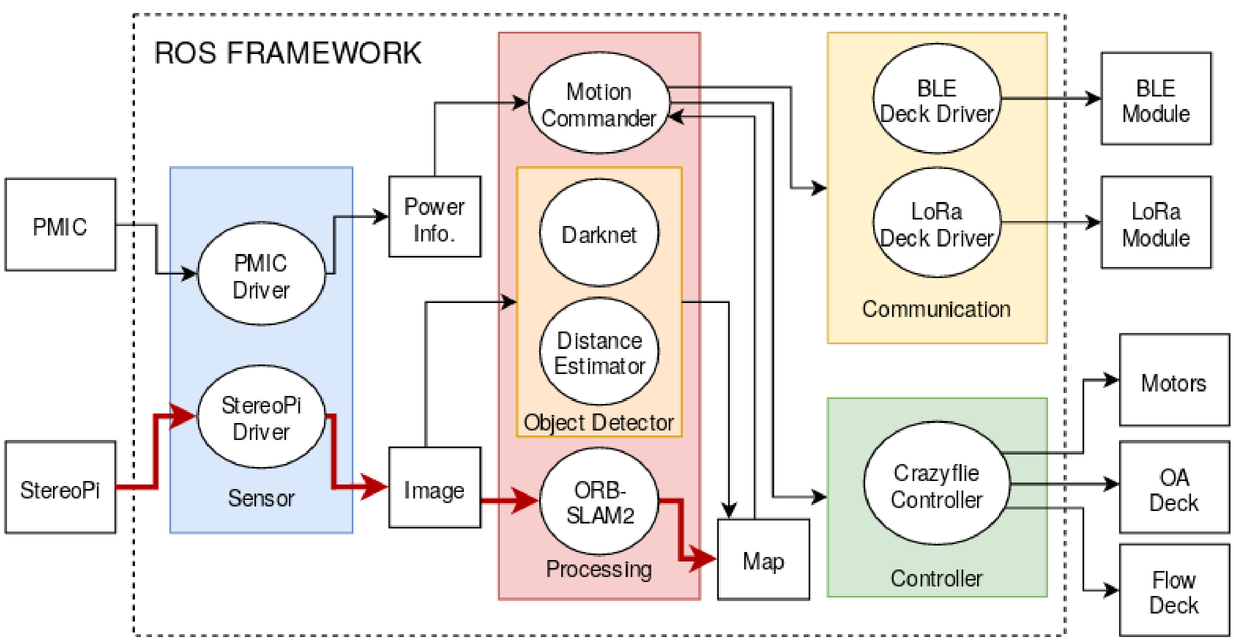


Figure 4. The ROS framework for our system

Goal 4: System integration and evaluation

We have integrated all hardware resources to the big drone and integrate all software to the ROS framework. With everything running together and performing test case tasks, the whole system should be tested for several times to clear for bugs and ensure system stability.

3.3 Deliverables

A fully functioning and well-modularized drone that is capable of autonomous/manual flight, Bluetooth/LoRa communication, and performing visual SLAM and object detection tasks.

3.4 Design

Goal 1: Functioning Big Drone

Task 1.1: The Building of the big drone based on Crazyflie system

Description: To have a functioning big drone, we have done multiple jobs, including the sourcing of appropriate metal frames, motors with enough power and their driving ESCs, assembling and soldering each part, motor testing, PID tuning, and finally the flight test. A light frame has been selected to reduce the overall weight of the drone to fulfill the regulation.

Challenges: The drone was essential to the projects as all other hardware built around it. We should have the drone ready as soon as possible.

Task 1.2: The building of custom PCBs

Description: With a working drone built in task 1.1, we have added the Flow deck and Obstacle Avoidance deck to realize autonomous flying. The Obstacle Avoidance deck is built for avoiding obstacles during the flight by giving the distance between the drone and the surrounding objects. While the Flow deck would give the information about the altitude of the drone above the ground level and how far the drone has moved. The Crazyflie's BigQuad Deck enables the control of the brushless motors using several GPIO pins. However, one of the IO ports conflicts with the Flow Deck's PMW3901 chip selection pin for its SPI bus. This restricts usage of both the BigQuad and the Flow Deck together on the Crazyflie. We have designed and built two custom PCB boards that allowed us to bypass these requirements as our only options were to risk destroying the decks during the physical modifications.

Challenges: The PCB design went through several iterations and it took some time waiting for manufacturing.

Task 1.3 The firmware programming of the LoRa module

Description: The LoRa module is for the long-range communication between drones. The module is responsible for receiving/sending data from/to the Jetson Nano though the UART port It should also build up the connection with another LoRa module on another drone and exchange data through the LoRa protocol. Thus, we have programmed the LoRa module firmware to realize those communication requirements.

Goal 2: Real-time Visual SLAM system and Object detection

Task 2.1: The testing of Camera Driver and Camera Calibration

Description: To stream the real-time video from the stereoPi camera, we have found an IP camera driver ROS package and validate it. The latency was considered to ensure the real-time performance. The camera calibration is an essential process to obtain the intrinsic parameters of the camera including focal lengths, principle points, distortion information etc. Those parameters are used when performing ORB-SLAM2 and generating disparity map tasks. The quality of the obtained intrinsic parameters directly influences the accuracy of the distance estimation result and the resulting map.

Challenges: Multiple camera calibration have been performed to achieve better results.

Task 2.2: The installation and testing of the ORB-SLAM2 System

Description: With an existing ORB-SLAM2 ROS implementation, we have built the package in our ROS space. The built package has been validated by some benchmarks and the videos generated by our stereoPi to ensure the compatibility of the whole system

Challenges: It required some experience on project building and some understandings on the SLAM system.

Task 2.3: The installation and testing of the YOLO algorithm

Description: Like the ORB-SLAM2 package, we have built the YOLO algorithm on our ROS space and performed some testing. For better performance, we have built the project with GPU support to explore the performance of the algorithm as the Nano has available GPU resources.

Challenges: Similar to the task 2.2.

Goal 3: Software Programming

Task 3.1: The programming of BLE/LoRa module ROS Driver

Description: With a BLE module and a LoRa module connecting to the Jetson Nano, we have built the interfaces for managing those modules and exchanging information. These two drivers were written in python, the programming languages supported by ROS. They could run as individual ROS nodes and data is encapsulated as ROS topics for communication between nodes.

Challenges: Finishing this task required the knowledge on ROS and we went through some ROS tutorials to understand how to program ROS nodes.

Task 3.2: The programming of PMIC ROS Driver

Description: The INA260 uses micropython code which is a version of python optimized to run on microcontrollers. This enables the PMIC to communicate via I2C protocol with the Jetson Nano. We have programmed the driver for the PMIC on ROS and it could get the real-time current draw from the battery to calculate the remaining energy, and alerts the drone to perform a safe landing when the battery is lower than a preset threshold.

Challenges: The INA260 library uses python3 which is not compatible with our version of ROS and we have found a way to get around it.

Task 3.3: The programming of Crazyflie ROS Controller

Description: The Crazyflie ROS controller is the interface to the onboard crazyflie system. We have programmed this controller and it could create a connection to the Crazyflie 2.1 via the Crazyflie dongle when launched, and continuously listens to the motion commands from the Jetson Nano which can direct the drone to lift, and maneuver through the air and safely land on the ground. It could also publish the sensor data taken from the sensors/decks on the crazyflie system, such as the Flow Deck, IMU, and the OA deck which tells the distance to the obstacles in four directions.

Challenges: It required the deep understand of some crazyflie libraries and how the crazfylie works.

Goal 4: System integration and evaluation

Task 4.1: The integration of all hardware resources to the big drone

Description: We have integrated all hardware resources to our big drone, considering the component placement requirement, such as the Flow Deck must be placed on the bottom of the drone without any blocking, the OA Deck must be in a place where nothing is blocking at its four directions, the stereoPi camera must be placed in front without any blocking.

Challenges: The space on drone was limited and it required to keep the overall weight balance of the drone.

Task 4.2: The integration of all software to the ROS framework

Description: After finishing the ROS software programming and as we discussed previously, all those are nodes on ROS, and we have interconnected each part with self-defined data structure and messages for the communication between ROS nodes.

Task 4.3: The experiments and system validation

Description: We have conducted several experiments to check our hardware and software separately. Debugging and modifying in hardware and software have been made when any issue was found. We should also conduct multiple experiments after integrating all the hardware and software to validate the whole system.

Challenges: Experiments were time-consuming but should be conducted as soon as possible. A system that could provide ground true measurements would be very helpful for verifying the map accuracy.

3.4.1 Estimated Budget for the overall project

Table 2: Estimated Overall Budget

No.	Equipment	Qty	Unit Price	Price (USD)	Note
1	Crazyflie 2.1	2	195.00	390	
2	Frame	1	28.99	28.99	
3	BigQuad Deck	1	7.00	7.00	
4	Flow Deck v2	1	45.00	45.00	
5	OA Deck	1	80.00	80.00	
6	BEC	2	6.90	13.80	Battery Eliminator Circuit
7	ESCs	4	10.56	42.24	Electronic Speed Control module
8	Emax RS 2205 Motors	8	7.25	58	Sets of x4
9	Propellers	4	14.99	14.99	
10	Battery	2	21.06	42.12	
11	PCB Development	N/A	N/A	100	Estimation based on OSH Park services
12	PCB Component	N/A	N/A	100	Estimation based on previous designs
13	Jetson Nano Development Kit	3	99.00	297.00	
14	Drone Registration	3	5.00	15.00	
15	PMIC Development Board	1	50.73	50.73	Power Management Integrated Circuit
	Total			1265.81	

3.5 Design Alternative

3.5.1 Hardware Combination Alternative

Two major components could be replaced for the alternative design. If the Crazyflie system fails to support us with stable flight control, Crazyflie would be replaced by the Arduino Uno. With an open source flight control software, the Uno would be the microcontroller for the control of the motors under the command of the Jetson Nano. In this case, extra work should be done in the hardware redesign of the Flow deck and Obstacle Avoidance Deck for the new flight control software. If the current stereo Pi camera would not provide enough pixel resolution for the Visual SLAM algorithm, the stereo Pi camera could be replaced with the ZED mini camera. The ZED Mini camera also provides a software development kit (SDK) for its development with the Jetson Nano development board. This would improve compatibility for the user.

For the Obstacle Avoidance (OA) Deck, instead of using the original laser sensor, stereo cameras can be put in four directions to detect the obstacle and even recognize the obstacle. For such a stereo camera-based OA, it should not be driven by the crazyflie anymore and it should be directly driven the Jetson Nano and an extra object detection node would be required on the ROS.

3.5.2 Edge Server

When the Jetson Nano is not capable of finishing the computing tasks especially if we decide to generate a higher density map with the ORB-SLAM, it can be upgraded to the Jetson TX2 board. Another option is to add an edge server to relieve the computing pressure like [16]. But it will cause a bigger latency if we migrate the Visual SLAM task to the edge server. If using the edge server, we would need to add a Wi-Fi module hardware and a Wi-Fi node in the ROS framework to handle the communication between our big drone and the edge server.

3.6 Milestones

The following table shows the arrangement for milestones and the distribution of tasks inside our group to reach the deliverables of this project.

Table 3: Milestone table for 2019 Fall and 2020 spring semester

Milestone (2019 Fall)	Date	Tasks
Milestone 1	11/14/19- 12/03/19	Drone Assembly (L & C) PMIC Researching (C) StereoPi testing (D) ROS studying (D&L&C)
Milestone 2	12/05/19- 12/19/19	Motor and ESC debugging (L&C) PMIC Studying (C) SLAM testing(D) ROS setting up (D) ROS studying (D&L&C)
Milestone (2020 Spring)		
Milestone 1	01/27/20- 02/11/20	Flight test without FlowDeck (L&C) Camera driver and calibration(D) PMIC design (C)
Milestone 2	02/12/20- 02/25/20	Draft PCB Design Draft Drone Accessories 3D printing (L&C) SLAM Algorithm live test (D) PMIC firmware design(C)
Milestone 3	02/26/20- 03/10/20	Crazyflie ROS Node Programming (D) OA, Flow, BLE Deck PCB (L&C) PMIC ROS driver (C&D) Flight test with FlowDeck(L&C)
Milestone 4	03/11/20- 03/24/20	Flight test with FlowDeck and Nano Part 1(L&C) Camera and SLAM integration to Drone system(D) PMIC integration to Drone(L&C)
Milestone 5	03/25/20- 04/07/20	Object detection testing (D) Flight test with FlowDeck and Nano Part 2(L&C)
Milestone 6	04/08/20- 04/21/20	Software integration and debugging (D) All Hardware testing and debugging(L&C) Technical writing (L&C&D)
Milestone 7	04/22/20- 05/05/20	System testing and debugging(L&C&D) Technical writing (L&C&D)

L: Lucas, C: Carlos, D: Deqiang

3.7 Gantt Chart

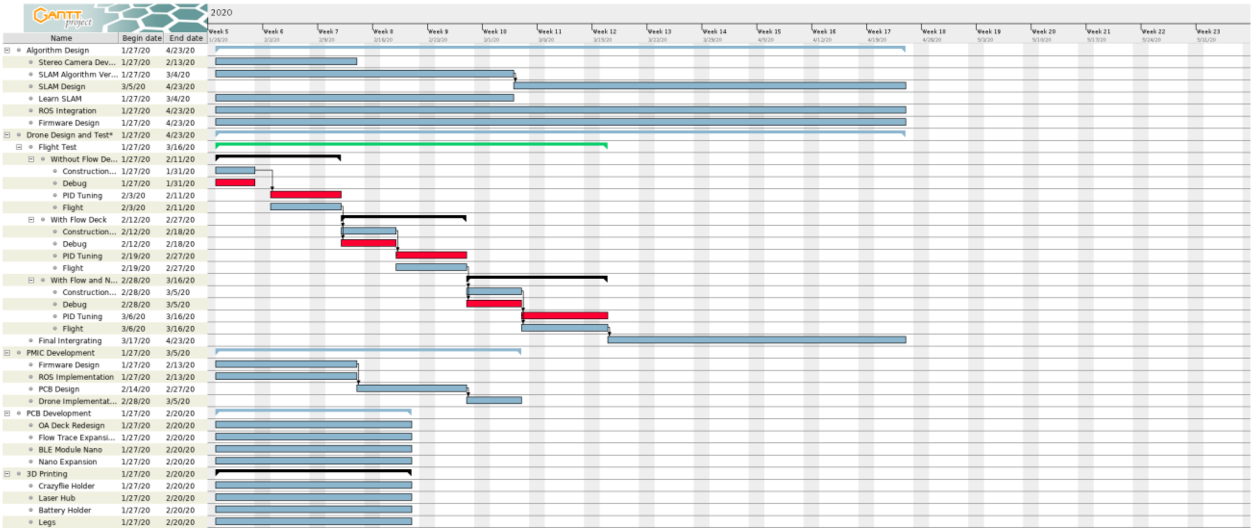


Figure 5: The proposed Gantt Chart for this project

4. Result

We have implemented, tested, and proven the feasibility of our proposed design, including the development of a large drone, the programming of drivers for flight controller and drone communication, and the integration of all ROS packages for performing real-time visual SLAM and object detection tasks.

4.1 Drone and Flight Test

We have successfully built the modular drone, as shown in Figure 6. The placement of the hardware components has been selected to maximize area usage and the protection of the onboard components. The Jetson Nano was placed on the top of the drone to allow its heat sink to emit heat without radiating to other components. The battery was fastened to the bottom of the drone as to protect the rest of the components if a malfunction were to occur. Four 3-D printed legs allow for an 80cm clearance for more space and to provide a stable platform for takeoff and landing. The stereoPi camera is no shown in the picture and it should be placed in the front top of the drone. We have evaluated our system in terms of a series of flight tests, and the maps generated from our environments.

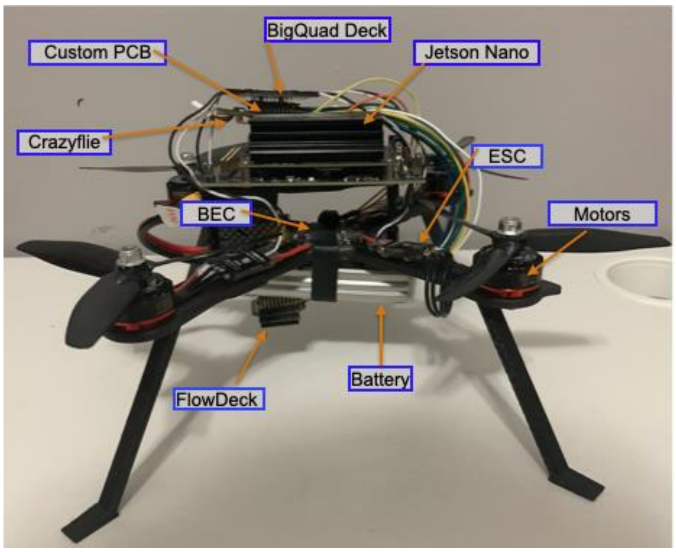


Figure 6: The Big Drone

We have performed two types of flight test, which at first the drone was controlled by the gamepad controller in the early stages of development to test the responsiveness of the motors and the flight stability and then its autonomous flight capabilities was tested. Without the sensor data provided by the Flow Deck, the drone is limited to the input commands of the pilot via the gamepad controller. In this mode of flight, the Big Quad Deck was the only enabled peripheral deck allowing the Crazyflie to use larger motors but with no autonomous functions of flight. In the test, the drone was taken off and flew through space stably.

Enabling the Flow Deck, the drone could move autonomously through space with commands from the Jetson Nano without manual control. The limitation to motor speed can be removed as it is taken care of by software and the whole flight time can increase as fewer error-prone movements might happen due to human input. In the test, the Flow Deck help stabilize the drone and the drone could maintain a constant distance from the floor with the help of laser senor on the Flow Deck when took off. The drone successfully took off and landed without any human interference.

4.2 SLAM Result

Figure 7 shows the generated map with our system after performing the visual SLAM task to a typical house consisting of a living room, a dining room, a study, and a kitchen. White dots are the landmarks extracted from the objects in each keyframe for motion tracking and scene recognition. The blue triangle represents the current pose of the camera and the red arrow indicates the place that the camera looked at.

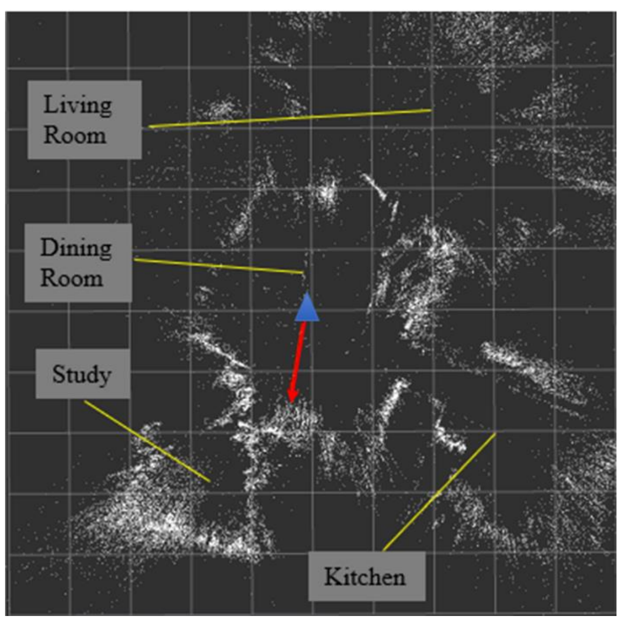
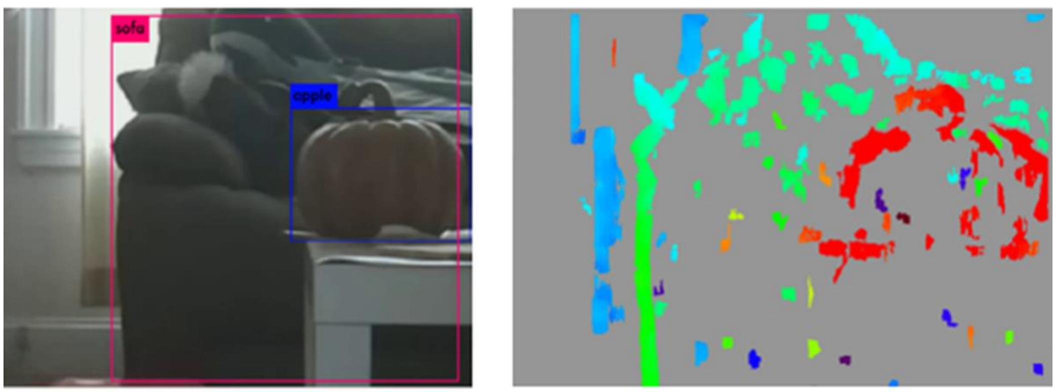


Figure 7: Generated Map by ORB-SLAM2

4.3 Object Detection Result

Figure 8 (a) shows the result of object detection in our system with a pre-trained model. A sofa was successfully detected in the image. Since there is not an object class called pumpkin in the trained model, the pumpkin was predicted as an apple. Each detected object was bounded in a rectangle box to show the edges of the object with a predicted name that has the highest confidence. Figure 8(b) shows the generated disparity image based on the left and right frame when performing image rectification. The disparity image is color-coded where deeper color represents a smaller distance while the lighter color represents a larger distance from the camera. The pumpkin was found closer and the sofa was found further away from the image.



(a) (b)

Figure 8: Object detection result

4.4 Final Budge

Table 4: Final spent budge on this project

Item	Number	Unit Price	Price
PMIC Board	1	$50.73	$50.73
Crazyflie 2.1	2	$195.00	$390.00
Crazyflie Dongle	3	$30.00	$90.00
Jetson Nano	1	$99.00	$99.00
Adafruit PMIC	1	$9.95	$9.95
PCB Boards			$10.00
New ESCs	4	30.33	121.32
New motors	4	14.99	59.96
Battery	1	57.99	57.99
Big Quad	2	7.00	14.00
Propellers	8	3.99	31.92
Total			$934.87

5. Bibliography

[1] Taketomi, Takafumi, et al. “Visual SLAM Algorithms: A Survey from 2010 to 2016.” IPSJ Transactions on Computer Vision and Applications, vol. 9, no. 1, Dec. 2017, p. 16. DOI.org (Crossref), doi:10.1186/s41074-017-0027-2.

[2] Cadena, Cesar, et al. “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age.” IEEE Transactions on Robotics, vol. 32, no. 6, Dec. 2016, pp. 1309–32. DOI.org (Crossref), doi:10.1109/TRO.2016.2624754.

[3] R. Mur-Artal and J. D. Tardos, “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras,” IEEE Transactions on Robotics,vol. 33, no. 5, pp. 1255–1262, Oct. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7946260

[4] Home | Bitcraze. https://www.bitcraze.io/. Accessed 27 May. 2020.

[5] M. Woolley, “Bluetooth 5 go faster. go further.” [On-line]. Available: https://www.bluetooth.com/wp-content/uploads/2019/03/Bluetooth_5-FINAL.pdf

[6] J. B.-V. M.-D. C. Ramon Sanchez-Iborra, Jesus Sanchez-Gomez and A. F. Skarmeta, “Performance evaluation oflora considering scenario conditions,” Mar. 2018

[7] “Lithium Ion Battery.” Clean Energy Institute, www.cei.washington.edu/education/

science-of-solar/battery-technology/.

[8] Li, Zheng. POWER MANAGEMENT INTEGRATED CIRCUITS DESIGN, FUNCTIONALITY ANALYSIS AND APPLICATIONS. University of Texas, 2005.

[9] Hydra. http://thecarlab.org/hydraone/. Accessed 27 May. 2020.

[10] ROS.Org | Powering the World’s Robots. https://www.ros.org/. Accessed 27 May. 2020.

[11] Sims, Ronald R., et al. Transforming Government Organizations: Fresh Ideas and Examples from the Field. IAP, 2016.

[12] Fan, Jin, and Mohamad Saadeghvaziri. *Applications of Drones in Infrastructures: Challenges and Opportunities*. New Jersey Institute of Technology, 2019.

[13] Shi, Yeyin, et al. “Unmanned Aerial Vehicles for High-Throughput Phenotyping and Agronomic Research.” PLOS ONE, vol. 11, no. 7, July 2016, p. e0159781. PLoS Journals, doi: 10.1371/journal.pone.0159781.

[14] Fact Sheet – Small Unmanned Aircraft Regulations (Part 107). https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=22615. Accessed 14 Nov. 2019.

[15] ArcGIS Web Application. https://faa.maps.arcgis.com/apps/webappviewer/index.html?id=9c2e4406710048e19806ebf6a06754ad. Accessed 14 Nov. 2019.

[16] Wang, Junjue, et al. "Bandwidth-efficient live video analytics for drones via edge computing." 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE, 2018.