# Autonomous Robotics Competition Club (ARCC)

Rachel Axten, Wen-Yu Chien, Jacob Crouse, Oliver Dunbabin, Venkatakrishnan Iyer, Kalki Sharma, and Vidullan Surendran*
Pennsylvania State University, University Park, Pennsylvania

ABSTRACT

In this paper, the technical approach to achieve the objectives for the indoor and outdoor missions of the International Micro Air Vehicle (IMAV) competition are presented. The ARCC team plans to utilize a single drone for the indoor competition and multiple drones to achieve the outdoor mission objectives. A scaled up version of the indoor drone would be utilized for the outdoor competitions to increase endurance and carry a larger sensor payload. A fully autonomous and primarily vision focused approach was taken for both missions with all computations being performed on-board excluding the 2D mapping which is performed off-board for the outdoor mission, whereas the indoor mission leverages off-board computing to reduce vehicle mass. Computer vision tasks are performed using a combination of the ZBar library for QR codes, semantic segmentation using the UNet architecture, object recognition using an RCNN, and classical image processing techniques such as ORB features and Hough transforms.

## 1 INTRODUCTION

The Autonomous Robotics Competition Club (ARCC) is a student run organization formed in the spring of 2018 and advised by Dr. Eric Johnson. The club consists of graduate and undergraduate students from the Pennsylvania State University. The club was founded with the purpose of participating in autonomous aerial vehicle competitions. The club consists primarily of students studying in the fields of acoustics, aerospace engineering, mechanical engineering, and electrical engineering but is open to anyone interested in such activities. A detailed description of the organisation and prior work can be found at the the club's website: https://sites.psu.edu/arcc/.

In this paper the previous work of Penn State's ARCC team, the technical approach for the indoor and outdoor competitions, and hardware and software specific elements are outlined.

## 2 PREVIOUS WORK

Thus far, the ARCC organisation has participated in the Vertical Flight Society's 2019 Mico-Aerial Vehicle Challenge (videos related to the competition are available on the club's website). The competition required the vehicle to pickup a package, navigate a course avoiding obstacles while flying over certain zones, and finally drop off the payload at a specified location while staying within the arena boundaries. The problems addressed were localisation, image recognition, and decision making. Initially an optical flow device paired with a Lidar was used to aid in localisation, but the mono-chromatic competition floor was featureless and thus we adopted a vision based approach using the Intel RealSense. A downward-facing camera was used to detect targets marked by April Tags. The images were transferred to a ground station via WiFi for processing. For obstacle detection a sonar sensor was used. The total weight of the vehicle was 500 grams. For package retrieval a pick-up mechanism was 3D printed doubling as the legs of the vehicle. To pick up the package the vehicle would hover above the package, lower itself, and close its servo actuated legs. The vehicle followed a pre-programmed flight path which was fine-tuned during run time through visual identification of landmarks reactionary behaviour to the world state. For example, if the vehicle was set to go straight but the sonar detected an obstacle, it would perform an obstacle avoidance maneuver prior to continuing on its path. An image of the vehicle is shown in Fig. 1.

Figure 1: Vehicle used for the Vertical Flight Society's competition with the optical flow (later replaced by RealSense).

## 3 HARDWARE

Since the competition is split into an indoor and outdoor portion separate drones have been built for the two missions.

This section outlines the hardware components chosen for each mission along with an estimate for the component and total weight.

Due to the increased image processing requirements of the outdoor competition, in addition to the Aaeon compute board, the outdoor drones will be outfitted with a much more capable Jetson Nano board, which is equipped with a GPU allowing us to speed up CNN inference. The specific components are outlined in Table 1. Since the outdoor vehicle is larger, the frame, power train, gripper, and battery are sized to reflect this. We chose to reuse the rest of the sensor payload from the indoor vehicle on the outdoor one. The indoor vehicle currently weighs about 470 grams whereas the outdoor vehicle weighs about 931 grams.

The open source px4 flight stack was chosen for this competition and thus the Pixhawk flight controller was utilized. The device has an on-board magnetometer and external GPS module, but these are disabled/removed on the indoor flight vehicle. The GPS sensor is removed to reduce weight and the magnetometer was disabled due to the potential of magnetic interference indoors. The flight controller is used to stabilise the vehicle whereas the Intel Realsense, Lidar, and camera make up the perceptual system.

## 4 INDOOR MISSION

The indoor mission requires the drone to navigate a warehouse environment autonomously. The mission profile involves recognition of key features (QR codes, flags, landing pad, etc), avoidance of obstacles (e.g. shelving), package retrieval and delivery. In order to accomplish these tasks autonomously several problems need to be addressed and solved. These are the problems of localisation, image recognition, control, path planning, communication, and package delivery. Prior to addressing the problems an overview of the hardware components and how they interact is presented in the next section.

### *4.1 Architectural Overview*

The control architecture of the vehicle is comprised of three modules as shown in Fig. 2. The first module is the manual control. The manual control sends commands directly to the fight controller on the vehicle via a human controlled transmitter operating in the 2.4GHz band. This link allows us to take over manual control or disable the vehicle in case of emergencies. The second module is the vehicle. The vehicle is comprised of on-board computer and micro controller, 5GHz WiFi data link, flight controller, Intel realsense, and a camera used to identify mission elements using computer vision. The inertial sensors and motors are connected to the flight controller which forwards telemetry data to the on-board computer. The camera feed is sent to the on-board computer which transmits it over the data link to a ground station for processing. The on-board computer receives control commands from the ground station to be forwarded to the flight controller which executes the commands.

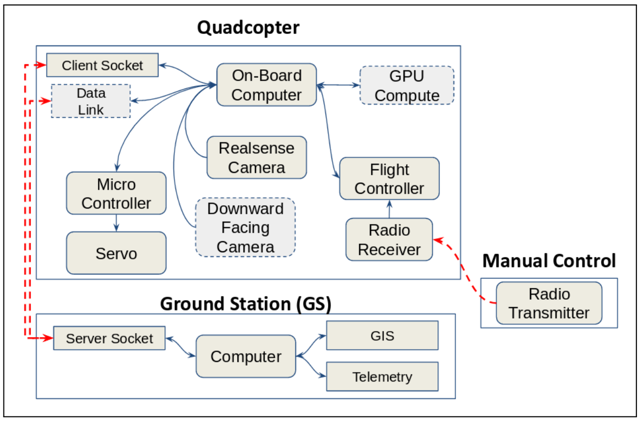


Figure 2: Overview of sub systems and how they are connected. Red links represent wireless communication. Grey dashed boxes are specific to the outdoor vehicle only.

In the event of a loss of connection with the ground station, the on-board computer is able to maintain static stability of the craft while it tries to reconnect to the ground station. The micro-controller is used to actuate the pickup mechanism and read/write to any sensor payloads that may be used.

### *4.2 Localisation*

Localisation will be performed visually using the RealSense T265 camera. Since the camera does not provide a map, we will be post processing camera images to generate a map. This map could help us correct deviations in local position estimates if the ground truth locations of the static environment are known or estimated during flight using stereo vision/distance sensor.

In order to effectively command the Micro Aerial Vehicle (MAV), its current position in inertial space must be known within some degree of accuracy. Inside a warehouse environment, the drone is likely denied a GPS signal, and hence must look towards other sensors to provide the pose measurements required to estimate the vehicle's state. To this end, visual Simultaneous Localisation and Mapping (V-SLAM) is used, in the form of the RealSense T265, a camera-CPU suite capable of performing V-SLAM.

SLAM algorithms estimate vehicle pose by identifying features in a camera image plane, and tracking it frame-to-frame. The movement of the feature in the image between frames, coupled with knowledge of camera movement provided through accelerometer and gyroscope sensors, allow for the estimation of the location of that feature point in 3D inertial space with-respect-to the camera. Simultaneously, the accelerometers and gyroscopes can be used to estimate the camera's pose via some filtering technique such as a Kalman Filter in this case. This estimate is subject to accelerometer bias and gyroscopic drift. The feature point state estimates are then used to correct for these sensor drifts. The result is an accurate estimate of both the camera and feature point

| **Component** | **Quantity** | **Weight - Indoor** (g) | **Weight - Outdoor** (g) | **Description** |
|---|---|---|---|---|
| Frame | 1 | 80 | 150 | Composite |
| ESC | 4 | 10 | 10 | 4 in 1 |
| Motor | 4 | 16 | 16 | Emax motors |
| Flight Controller | 1 | 10 | 70 | Outdoor incl. GPS |
| Onboard Computer | 1 | 60 | N/A | Aaeon Up |
| GPU Compute | 1 | N/A | 130 | Jetson Nano |
| Intel RealSense | 1 | 60 | N/A | T265 |
| Lidar | 1 | 12 | N/A | Range sensor |
| Camera | 1 | N/A | 15 | Downward facing |
| Propeller | 4 | 5 | 8 | APC |
| Payload Gripper | 1 | 40 | 30 | 3D printed |
| Battery | 1 | 120 | 300 | LiPo |
| Miscellaneous | N/A | 50 | 100 | Wires, connectors,... |
| Total | | 470g Approx. | 931g Approx. | |

Table 1: Vehicle components and weight. N/A represents components not applicable to that version of the vehicle.

state. A LIDAR distance sensor is used for accurate altitude measurement and as is common for UAV applications, an Extended Kalman Filter is used to combine the measurements from these sensors, with vehicle state equations, to yield an estimate of the vehicle state (position and pose).

### *4.3 Image Recognition*

QR code recognition will be done using the ZBar library. This library requires a cropped image containing the QR code in order to recognise it. We trained a Regional Convolutional Neural Network (R-CNN) using the inception v3 backbone on synthetically generated images containing various QR codes superimposed on a wide variety of indoor backgrounds. It is hard to quantify accuracy as our synthetic data set is not standardised, but we obtained an accuracy of approximately 92% on the held out test set. The bounding box generated was then cropped out and recognised using ZBar. A similar strategy was employed to detect the boxes and the flag.

To circumvent the issue of being unable to predict the effectiveness of the RCNN in the actual competition environment and since we have ample computing power due to offboard processing, we validate our results by performing semantic segmentation using the UNET architecture [2]. The goal of a segmentation problem is to section the pixels of an image into the classes that the network has been trained on. The data for these classes would be obtained from an image search on the internet thereby augmenting our dataset with labelled data that represents the classes we are interested in such as the shelves, mailboxes and other objects of interest. The segmented pixels would then be extracted and then cross-validated against the CNN classifier to confirm that the correct class has been detected. A weighted average of the the R-CNN and UNET prediction would be considered as the final result.

For the boxes on the shelves we had to train a regional network as the architecture we used for semantic segmentation does not segment multiple objects of the same class, only providing a blob that contains all the objects. Since there are many boxes on the shelf, we are training a faster-RCNN [3] to draw bounding rectangles around them. The boxes are nondescript and of various shapes with the identifying factor that they contain QR codes. The training dataset for this particular case was generated from images of cardboard boxes that were super imposed with a QR code and warped, rotated, and skewed using classical image processing techniques.

In the case of the flag, during take off we hope to detect the flag pole and store the image of the flag as a ground truth image. When we encounter the image during the drop-off phase, we would compare it to the ground truth image using ORB [4] features. A backup plan is to use transfer learning to quickly train a CNN in the 2 training days preceding the competition when we are made aware of the flags that would be used as these would only be a subset of all the flags in the world.

Finally, to detect the landing zone which has an 'H' on it, we will reuse code from our previous competitions where we use probabilistic Hough transforms to identify the 3 lines which are oriented in a known geometric shape.

### *4.4 Controls*

Control of the MAV must be robust enough to handle the coupled dynamics of the MAV and payload package slung load; to this end, control is provided by the PX4 firmware running atop the Pixracer flight controller. As the lion's share of vehicle control is being provided by the PX4 firmware, only a brief treatment of the controller software will be presented. The PX4 firmware estimates the state of the vehicle (from accelerometer, gyroscope, LIDAR, V-SLAM, and GPS sensors) via an Extended Kalman Filter (EKF). It then compares the

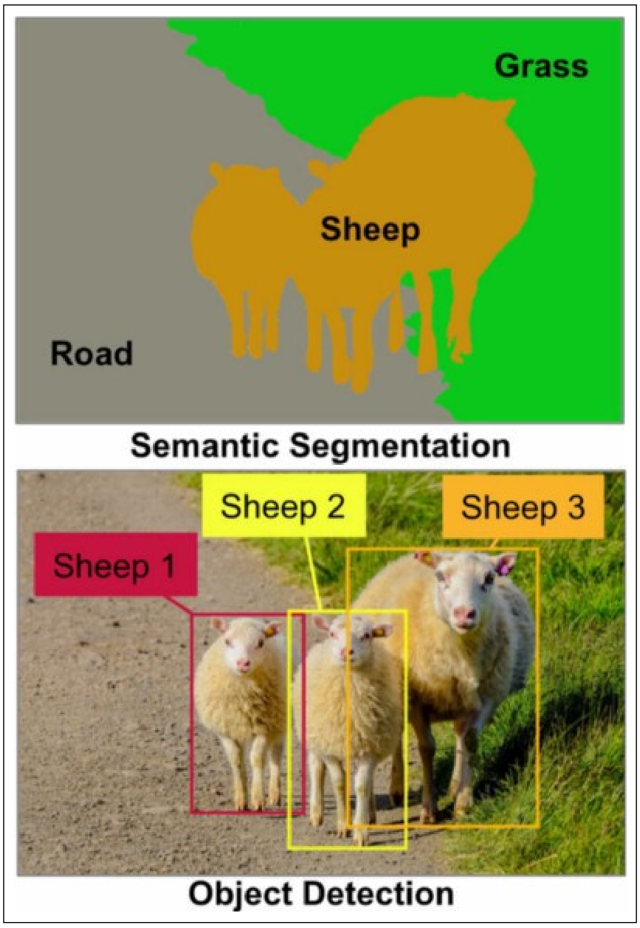


Figure 3: Illustrates the difference between image segmentation and object recognition, the two approaches adopted in our algorithms. [1]

vehicles current state estimate (position, velocity, pose, and pose rates) to that which is commanded by trajectory planning. Error between current and desired state, prompts the controller to send signals to the four motors, eliciting collective, lateral and longitudinal cyclic (governing pitch, roll, and yaw of the vehicle - $\phi, \theta, \psi$ - and body angular rates p, q, r) response of the vehicle.

The magnitude to which this state error influences the motor inputs is regulated via a simple proportional-integral-derivative (PID) controller.

$$U(s) = (K_p + \frac{1}{s}K_i + K_d s)E(s) \tag{1}$$

Here, $E(s)$ is the tracking error, in the Laplace domain, $K_p$ gain minimises tracking error, where the higher the gain value the faster the response; $K_d$ gain dampens the response of the vehicle, reducing overshoot; $K_i$ gain aids in reducing steady-state error.

PID controllers are used to control the vehicle's rate, attitude, velocity, and position command. Despite the dynamics of the system changing after the package pickup, the PX4 controllers are assumed robust enough (with large enough margins), as to observe minimal controller performance degradation. These gains are tuned for our particular flight needs, and will likely be slightly different for indoor and outdoor missions.

### *4.5 Path Planning*

A feature point in the camera image plane is defined as a point which may correspond to an obstacle in the environment. The process of tracking a feature point involves identifying a point of interest within the camera image frame and then tracking those points between frames (correlating a measured feature point with its counterpoint in a database of tracked features). The first step is to identify which objects within a frame are of interest. Two methods being utilised aboard our vehicle are classical image processing techniques (Harris Corner and Canny Edge detection algorithms)[5] as well as more contemporary/deep-learning techniques (training CNNs to identify objects likely to be encountered by the drone). Once these features are found within the image plane, these measurements need to be corresponded to those features seen in prior frames (stored in a database) in such a way that the same features are tracked between frames. To this end, the statistical Z-test is employed; a method which, when presented a new image of measured features/objects, can match measurements with objects stored in the database, with highest probability[6]. It is necessary to store these observed features in a database (and subsequently estimate their state - i.e. position), so that we can track their relative location, even in their absence from camera frames. To this end, the vehicle's state (i.e. pose), and the feature point's movement within the image plane, is used to estimate the object's location[7]. This is done using an EKF, whereby the vehicle's state equations, the feature's state equations, and the entire suite of vehicle sensors are fused to provide an accurate estimate of its location relative to the vehicle[8]. These estimates can then be used to avoid (and in some cases track towards - c.f. using image recognition to identify points of interest) objects within field of view of vehicle.

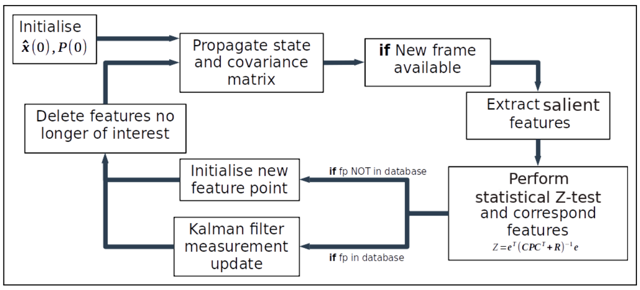


Figure 4: The feature point (fp) tracking EKF process. N.B: this flowchart is invariant of the image processing method.

### *4.6 Payload Delivery*

For the indoor vehicle, an alternative, weight-saving approach for the pickup mechanism will be considered since the package weighs 25 grams for vehicles weighing under 500 grams. The proposed mechanism will be a small hook like device which can be extended to secure the loop on the package. The hook will be mounted on a pivot such that it minimises visual interference during flight, and allows the vehicle to land on its 'belly' eliminating the need for landing gear. A common 3.5 gram micro servo actuates the hook the device and is capable of moving 60 degrees in 0.05 seconds. The servo arm is approximately 15cm and exploits the structural strength of the 3D printed plastic in tension to lift the package. This minimises the load on the servo compared to a gripper design which uses servo torque to hold the package. This allowed us to reduce weight and power consumption.

### *4.7 Communication*

This section outlines how the vehicle and ground station communicate with one another. Two types of data formats are sent and received by both the vehicle and the ground station. The data formats are Mavlink messages and video. The vehicle transmits telemetry data to the ground station, and the ground station logs the information along with displaying the current state for purposes of debugging and testing. The ground station sends commands to the vehicle via user input during testing. Both the telemetry and commands are sent as Mavlink messages. The other data format is that of video. Video is transmitted from the vehicle to the ground station, where the ground station displays the feed and uses it to feed the object detectors.

The communication protocol used for Mavlink data transmission is TCP/IP and for video is UDP/IP. Bo protocols utilize WiFi at 5GHz for data transmission, but the TCP and UDP have differing structures. The benefit of TCP is that it performs rigorous checking to ensure the data transmitted is received and in the format which it was sent. While UDP streams data are connection less meaning there is no acknowledgement that a package has been successfully transmitted. Therefore TCP is better suited for sending telemetry and commands where the transmission of a corrupted message or missing a message all together can prove fatal, and UDP is adequate for video transmission where missing a frame is not detrimental to the vehicle. When setting up socket communication a server and client must be specified. In this case the ground station serves as the server which waits for a client, the vehicle, to connect.

Apart from the vehicle and the ground station, a transmitter is also able to communicate with the vehicle in order to take over manual control. The transmitter operates at a frequency of 2.4GHz. In order to minimize interference and increase available bandwidth, Wi-Fi communication will utilize the 5GHz band.

## 5 OUTDOOR MISSION

For the outdoor competition, we plan to use a GPS module to obtain the current position of the drone in addition to fusion of vision data. As time is a crucial factor, three drones would be utilized to simultaneously deliver the three packages. As the location of the houses and post boxes are not provided, a random grid search would be performed to locate these targets. Similar to the indoor approach, we will leverage R-CNNs to identify the targets, namely the houses, post boxes, the crashed drones, and the missing parcels. The system will be equipped with a Raspberry Pi camera for use with mapping and target detection. From pixel coordinates, aircraft attitude and altitude, along with the GPS information, the pinhole camera model would be used to estimate the location of the detected objects in the world coordinate system.

Finally, the task of generating a 2D map of the area would be done using a Geographic Information System library, in our case GeoPython as our off-board computational stack is coded using python 3. Features in the images would be identified and tagged with global positioning information to stitch them together to generate a 2D map of the area. The process of feature tracking and determining the latitude and longitude in the images is automated such that the on-board computer tags each image taken with the current GPS location at capture so that the stitching of the images can be completed in the 20 minutes duration provided at the end of the mission. Further, vision would be used to detect the balloons in the takeoff and landing area but since a quad-rotor affords vertical take off and landing, with careful flight planning we do not anticipate an encounter with the obstacles.

In contrast to the indoor mission, the outdoor vehicle will communicate with the ground station using a 2.4GHz wireless broadcast link with a 900MHz control link for emergency manual control. These frequencies provide adequate range and bandwidth for video and control signals respectively. This would allow for the transmitting of camera data from the vehicle to the ground station for 2D map generation. The vehicles would still be autonomous with all processing done off-board, so loss of link would only restrict its ability to map the environment and send telemetry data.

The decision to perform all computation on-board the vehicle unlike the indoor competition is driven by the risk of connection loss on the 2.4GHz band at the allowed 25mW power limit at maximum operational range. As the vehicle all up weight is not as critical in the outdoor mission, we chose to include a more powerful compute board on-board for robustness and safety reasons.

### *5.1 Architectural Overview*

The proposed approach to the outdoor competition is to use two identical 500 gram quadrotors similar to the indoor competition mainly for package drop off with a larger (approximately 1Kg) vehicle that identifies drop off locations, and maps the environment. This is to optimize the weights of the vehicles to maximize the weight multipliers while creating a vehicle which will be powerful enough to lift the avionics as well as a post box package with a reasonable endurance.

The on-board avionics system for the two smaller vehicles will be the same as the indoor vehicle with the addition of a GPS module. The larger vehicle will use the Pixhawk 2.1 flight controller which supports autonomous waypoint tracking, as well as compatibility with all other avionics peripherals. To localize the system in the global coordinate frame, the team will use the Here GNSS GPS system, which is compatible with the Pixhawk and has been found to be accurate up to 3.7 meters of the desired destination [9]. Additionally for communication, the team will utilize 915MHz transmitters with a common ground station. All antennas will be circular polarized (2-5 dBi gain) to ensure uniform converge regardless of azimuth angle.

The vehicles have been designed to optimize the scoring system of the competition. The weight has been selected as an ideal point for a high weight scale factor, as well as allowing sufficient vehicle weight to support the proposed avionics set. Additionally, the system is optimized for target detection over mapping because the scoring system of the outdoor competition heavily favors detecting the points of interest over creating a high quality map.

### *5.2 Localization*

The plan to localize each identical system is using the Here GNSS GPS system which is compatible with the Pixhawk 2.1 flight controller.

### *5.3 Object Detection*

A Regional Convolutional Neural Network (R-CNN) is trained with synthetic databases generated to identify the house, mailboxes, lost packages and crashed drones similar to the indoor mission. As the mailboxes and lost packages are of unique colors, color identification schemes can also be used to identify these objects. However, owing to the ground color being a possible mixture of yellow and green, identification of the yellow mailbox using color detection schemes may yield false positives. Further, as the same network architecture can be trained to identify different objects using different databases, it is more convenient to use the same deep learning approach for detection of all objects. Transfer learning approach will be used to train the RCNN at a lower computational cost.

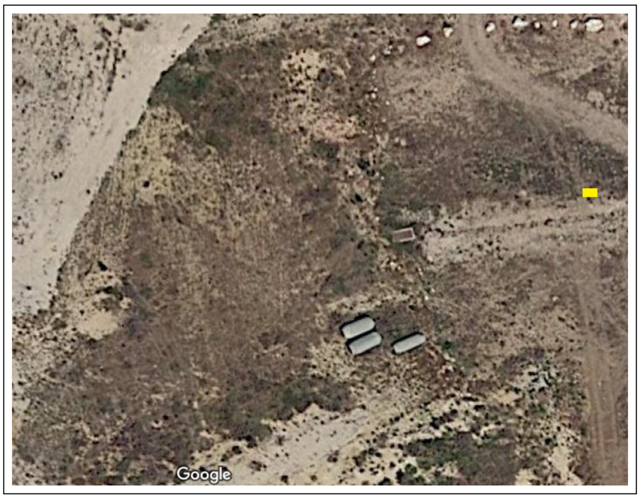


Figure 5: Example of an image for training the neural network. The superimposed mailbox can be seen in yellow.

| Characteristic | Value |
|---|---|
| f | 1.14 mm |
| $X_0$ | 0.507 mm |
| $Y_0$ | 0.395 mm |
| $k_1$ | -0.013 |
| $k_2$ | 0.1764 |
| $k_3$ | -0.6391 |
| $p_1$ | -0.0032 |
| $p_2$ | -0.0072 |

Table 2: Raspberry Pi camera parameters.

The CNN is fed images from a Raspberry Pi camera mounted on-board for inference. Since the multirotors will be flying at a height of 25 $[m]$, it is possible to determine the ground dimensions of the image since the internal parameters of the camera is known. The internal parameters of the camera is not available from the manufacturer and hence the same needs to be determined through camera calibration. A similar approach is adopted by Piras et al [10] where the camera parameters have been determined using camera calibration. The camera internal parameter as determined in the cited paper have been used. The parameters are listed in table 2.

In order to train the neural network, a database of images is generated. These images are generated by superimposing a colored box over an image of the outdoor competition's terrain in a random location. This colored box is scaled to be of the correct size relative to the predetermined height of flight of 25 $[m]$. An example of one of the training images can be seen in Figure 5. From this example we observe that the size of the objects are much smaller than the overall image which prompted us to use an R-CNN over other architectures such as YOLO [11] as emperical tests have confirmed that it has superior performance for smaller objects with careful tuning of anchor size and aspect ratio hyper parameters.

### *5.4 2D Mapping*

Two approaches are proposed to generate the 2D map of the area. In the first approach, multiple images with sufficient overlap as explained in the Path Planning section are obtained. Image stitching is then performed based on keypoints generated and using the RANSAC algorithm. This approach is currently implemented in the ground station. Figure

Figure 6: 2D mapping using image stitching.

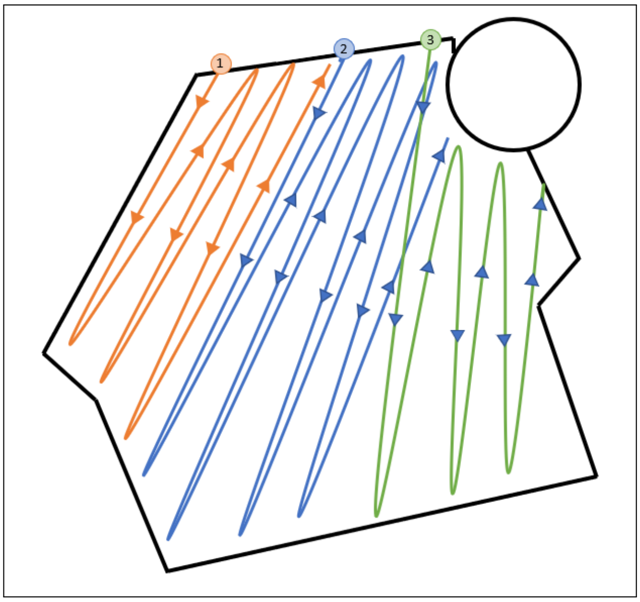


Figure 7: Predetermined path of each outdoor quadrotor.

6 illustrates the same based on multiple overlapping images obtained of an area from an elevation.

While the image stitching approach might yield acceptable results, there exists a possibility of artifacts in the stitched image owing to 3D objects. Thus, the alternative approach proposed is to generate orthomosaics. The orthomosaics are then geo-referenced using tools such as Open Drone Map, etc. This will generate the required 2D map of the operational area of the outdoor mission.

### *5.5 Path Planning*

The path design for the outdoor competition will take advantage of all three quadrotors at the same time. The two smaller vehicles will transmit their image data to the ground station for off-board identification of targets whereas the larger drone will perform this on-board. The common ground station ensures all vehicles will be kept in sync regarding recognised targets. Using this approach, the area will be mapped in the shortest time possible.

The default path for the vehicles will imitate the "mowing of a lawn" by starting at the starting location and sweeping back and forth within the competition area, moving in one direction until the boundary of the area is reached, and then shifting some distance horizontally while rotating 180° to move back toward the starting area. This behavior can be seen in the Figure 7 In case of a failure in Wi-Fi link at range, the two smaller drones will be unable to perform this task. In this scenario, the larger drone will be commanded to complete this behaviour slowing down the search, but still retaining the capability of using the smaller drones to deliver packages using GPS as this is not dependent on a high throughput Wi-Fi link.

In order to ensure that the entire area is mapped, sufficient overlap between images need to exist so that the image stitching can be performed correctly. Based on the geometry as detailed in Kraus [12], the percentage overlap in the forward direction is set to 60% and 30% in the sides. This will ensure that there are sufficient DoG and Harris keypoints available to ensure effective image stitching. The homography matrix for the matched vectors is determined using the RANSAC algorithm. With the desired percentage overlap, images in the forward direction need to be captured at every 40 m and the distance between two parallel paths should not be greater than 30 m.

### *5.6 Payload Delivery*

Time is a crucial factor in scoring of the outdoor phase of the competition. Each of the three multirotors will be loaded with a single package. Once one of the multirotors locates a post box, this information will be sent to the ground station and relayed to the multirotor with the corresponding package.

After the multirotor is aware of the location of the correct postbox, it will use the onboard Pixhawk 2.1 flight controller along with the Global Positioning System (GPS) location of the post box to navigate to the post box and deliver the package. Given the competition's requirement of landing the package within a 5 meter radius of the post box and the accuracy of the Here GNSS GPS module being reported to be within 3.7 meters we did not employ any additional vision correction. As the competition does not require the package to be picked up by the drone autonomously, the payload mechanism used is a mainly a package drop mechanism as shown in figure 8

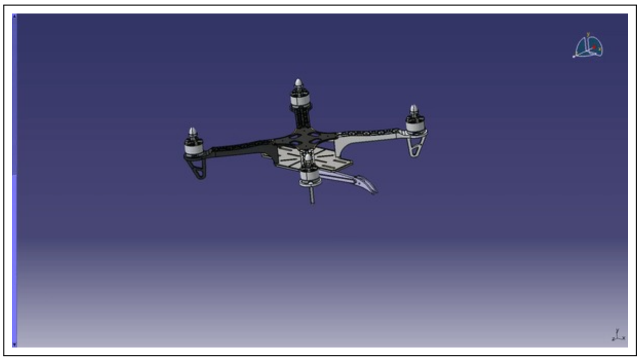

Figure 8: Package Drop Mechanism.

As can be seen in figure 8, the drop mechanism is in the retracted position. This ensures that the drop mechanism does not obstruct landing of the drone. The mechanism will be extended after takeoff and the package can be attached to the hook when the vehicle is hovering. The package is dropped at a desired location by retracting the mechanism.

### 5.7 Communication

The method of communication is similar to the indoor portion with the major difference being the transmission frequencies. Due to the range required of the outdoor vehicle, the WiFi operates at 2.4GHz as opposed to 5GHz. The manual control transmitter is operated at 915MHz to ensure it does not interfere with the WiFi communication.

## 6 CONCLUSION

The details presented in the paper is a brief insight into the technical approach that the team has envisioned to successfully complete the IMAV 2019 challenge. With many modules already implemented and ready, the team is on track to fly, test and validate all modules well in advance of the actual competition. With the feature tracking methodology, the warehouse mission can be successfully completed considering the fact that the real sense module provides localization with tolerances of within 3cms observed during testing. Finally, the deep learning framework proposed will enable real time target detection and identification owing to CNN inference being computationally efficient compared to classical vision processing.

## ACKNOWLEDGEMENTS

The team is thankful to the Department of Aerospace Engineering, Pennsylvania State University for providing necessary support and infrastructure. The team would also like to thank the Pennsylvania State University for the resources in terms of funding. Finally, the team is indebted to Dr. Eric Johnson who, as the academic advisor of the club, has been a constant source of technical guidance and support.

## REFERENCES

[1] Ravindra Parmar. Detection and Segmentation through ConvNets,*https://towardsdatascience.com/detection-and-segmentation-through-convnets-47aa42de27ea*, Sep 2018.

[2] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. *CoRR*, abs/1807.10165, 2018.

[3] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR*, abs/1506.01497, 2015.

[4] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society.

[5] Daniel Magree, John Mooney, and Eric Johnson. Monocular Visual Mapping for Obstacle Avoidance on UAVs. *Journal of Intelligent Robotic Systems*, 74(1-2):17–26, 04 2014.

[6] Girish Chowdhary, Eric N. Johnson, Daniel Magree, Allen Wu, and Andy Shein. GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. *Journal of Field Robotics*, 30(3):415–438, 2013.

[7] J. Civera, A. J. Davison, and J. Montiel. Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, 2008. Date revised - 2009-07-01; Last updated - 2011-11-14; SubjectsTermNotLitGenreText - Parametrization; Inverse; Standards; Representations; Localization; Uncertainty; Mapping; Position (location); Robotics.

[8] Autonomous Flight in GPS-Denied Environments Using Monocular Vision and Inertial Sensors. *Journal of Aerospace Information Systems*, 10(4):172–186, 2013.

[9] Titan Elite. Pixhawk compatible gnss receiver comparison,*https://files.zubax.com/products/com.zubax.gnss/gnss-receiver-performance-report-april-20184.pdf*, 2018.

[10] Macro Piras, Nives Grasso, and Ansar Abdul Jabbar. UAV Photogrammetric Solution Using A Raspberry Pi Camera Module and Smart Devices: Test and Results.

[11] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[12] K. Kraus, I. Harley, and S. Kyle. *Photogrammetry: Geometry from Images and Laser Scans*. Number v. 1 in De Gruyter textbook. Bod Third Party Titles, 2007.