Friday January 29th, 2021

NSWC Crane AISUM Prize Challenge

Abstract

Blue Wave AI Labs was founded in 2016 with a mission to utilize recent advances in data sciences to solve problems that were previously unsolvable. Since its inception, Blue Wave has combined the resources of academia with the experience of industry experts to use Artificial Intelligence techniques to provide solutions to formerly intractable problems.

Blue Wave consists of team of unusually talented scientists and engineers, including Aeronautical engineering, with deep experience in applying AI and Machine Learning. We use a unique blend of science, engineering, AI and mathematical expertise to solve real-world problems at a world-class level. Our focus is in defense, and nuclear energy.

Blue Wave has participated in the DARPA Alpha Dogfight challenge to develop an autonomous fighter and have been awarded a DARPA Gamebreaker contract. We partner with a hypersonic design company, *Hysonic Technologies*, to develop an AI-powered hypersonic shape design system. One member of our staff is an expert in aircraft control systems and is involved in the design of a small unmanned aircraft for fighting forest fires using AI-developed algorithms.

Glossary

CNN	Convolutional Neural Network – A type of neural network used to process images.
GPS	Global Positioning System – A satellite-based navigation system. Not allowed in AISUM
IMU	Inertial Measurement Unit - a sensor package including 3-axis accelerometers and 3-axis gyroscopes. Used by the flight controller to update the position and orientation of the vehicle.
PID	Proportional Integral /Derivative - shorthand for the controller that converts desired changes in drone position and orientation into speed commands for the motors.
SLAM	Simultaneous Localization and Mapping – general name for the problem of using only local measurements to correct for IMU drift and to build a map of the surroundings.
YOLO	You Only Look Once – A state-of-the art neural network used for object recognition.

Friday January 29th, 2021

AISUM Challenge White Paper – Blue Wave A.I. Labs

Challenge Description

The AISUM Challenge seeks to develop algorithms to improve the maneuverability and reconnaissance abilities of drones operating in confined spaces. AISUM has given several examples of scenarios that the drones may have to perform. We can summarize these scenarios by listing the capabilities that the drones must demonstrate; navigation, mapping, collision avoidance, object recognition, color detection, and friend/foe determination. The drone will come with standard hardware including a flight controller (containing inertial navigation sensors and the PID flight controls), an onboard processing unit, a main camera, and optical avoidance sensors. The drone has the capacity to carry 0.5 lb. of additional sensor/computation payload. As the drones will be working in confined, interior spaces, GPS signals will not be available for navigation and mapping.

Technical Approach

Navigation, Mapping, and Collision Avoidance

Drones typically navigate using a combination of an inertial measurement unit and GPS receivers. The inertial measurement unit consists of three-axis accelerometers and three-axis rate gyroscopes which measure the accelerations and angular velocities about the drone's body axes. These measurements can be combined with the dynamics equations to predict the spatial displacement and orientation of the drone. Periodic GPS measurements are used to refine these predictions and correct from drift and bias in the sensors.

As the drone will be operating in an indoor environment, GPS reception will be poor to nonexistent. Instead, we introduce a local, relative method of navigation referred to as Simultaneous Localization and Mapping (SLAM). SLAM methods use measurements of the local environment to correct the drone location and to build up a map the local environment. At each cycle, the drone measures its surroundings and identifies a number of landmarks. If the landmark was seen previously, its current position is compared to its expected position based on the motion of the drone; the difference in these numbers is used to correct the drone state and the landmark position. If the measured landmark is new, its location is saved to be used in later updates. Blue Wave will develop a set of algorithms to use these landmarks to create a map of the space explored by the drone.

A number of different sensors and software algorithms can be used in conjunction with SLAM. Single (monocular) and dual (binocular) camera can be used with SLAM. Cameras can directly measure the azimuth and elevation angles of the landmarks relative to the drones while range would be determined by taking two images and computing parallax. Dual-camera systems can use images taken simultaneously by both cameras while single-camera systems must move the drone slightly to get a second image, introducing slight uncertainty. Recent work has shown that a convolutional neural network (CNN) can be trained to determine the depth of pixels in an image [1]. Blue Wave will train an appropriately complex CNN to perform map the depth of each object in the picture. Lidar is a popular sensor choice due to its precision and ability to directly measure the distance to an object. Additionally, Lidar data can also be utilized for key object detection and collision avoidance since it is unaffected by poor lighting conditions and occlusions (where vision-based systems may struggle). We have chosen to use the drone's onboard camera together with the Extended Kalman Filter-based algorithm for navigation and mapping.

The navigation algorithm will use the map data along with the position history of the drone to determine which areas the drone has not previously explored. The navigation algorithm will generate waypoints in those areas that are passed to the flight controller where the "ArduPilot" software will adjust the motors to move the drone. The optical avoidance sensors are small sensors that point in directions not in the main camera's field of view. When the sensors detect that the drone is drifting close to an obstacle, an interrupt in the navigation algorithm will be triggered, moving the drone away from the obstacle. The navigation algorithm will include provisions for basic navigation and collision avoidance. It will also include the ability to detect staircases, and maneuver between floors. Blue Wave will use **Reinforcement Learning with an A3C policy** as intelligent reward function which aligns the drone with the nearest wall and the ceiling to train the agent to develop navigation algorithms. The agent will be trained within the provided simulation environment.

Object Detection and Identification

We will build and train a convolutional neural network (CNN) to detect the objects of interest. We will train the network on a large number of sample images to attain a high accuracy while being mindful of real-time impact. The current state of the art in object recognition is the YOLO (You Only Look Once) algorithm [2], or the pared down, but faster "tiny YOLO" [3]. YOLO has some notable flaws including problems detecting close objects and small objects. We will use YOLO as a starting point to develop a CNN tailored to use on the drone. In addition to a label for the object, the CNN would be designed to output bounding boxes indicating the location of the objects in the cameras field of view. These bounding boxes can be used to crop the full image so that the detected objects can be sent to another algorithm to determine certain properties such as color. Color can be determined by converting the cropped image from red-green-blue representation to hue-saturation-value representation. The dominant hue can be determined and compared with hue ranges for given colors/ Images of humans would be sent to a second algorithm to determine friend or foe. The object detection CNN can be combined with the depth-finding CNN to save resources.

If possible, we will run object detection onboard the drone. Having the location and identities of objects of interest can be used to improve the map that the drone builds. The CNN can also be used to help identify the open door through which the drone will exit. We can also imagine a situation where the drone is performing reconnaissance in a building feeding map and object data in real-time to ground assets following close behind. In such a case, we would want the object detection and classification algorithms to be performed onboard the drone. If the onboard processor does not have enough computational power to run the object detection algorithms in real-time alongside the navigation ang mapping algorithms, we have identified possible alternatives. The simplest alternative would be to store 1-2 images per second to onboard memory as the drone navigates the facility. Upon exiting the facility, these images will be transmitted to a base station for detection and classification. The second alternative is to use the 0.5 lbs. payload to include additional computational power. Several companies produce integrated circuits specifically designed to solve neural networks quickly and efficiently. We have identified the Intel Neural Compute stick as the potential secondary processor. We note that the drone's onboard processing unit may already include neural processing circuits (such as the Nvidia Jetson).

Data Flow

It is useful to consider the data flow from the sensors and through the different computational units for the scenarios in which we fly all three pieces of optional hardware. Here we assume the lidar is part of the payload, and a neural compute unit is available, whether onboard the supplied processor or as part of the optional payload. At each cycle, the IMU, the lidar, and the optical avoidances sensors take measurements, while the main camera will snap a picture. If no lidar is available, the main camera will take a second image within 0.5 seconds. The outputs from the optical avoidance sensors are passed to the processor to be used by the navigation algorithm to ensure that the drone does not collide with other objects. The IMU data is passed to the flight controller to estimate the change in position and orientation since the last time step. The estimated position and the lidar measurements (or two pictures) are passed to the compute unit and into the SLAM algorithm. The data flow for a drone with lidar and real-time object detection is shown in Figure 1.

The SLAM algorithm begins by identifying a number of landmarks present in the lidar data or images. Some of these landmarks are new, while many of them have been identified previously. The algorithm then finds the location of the prior landmarks relative to the drone. This location is combined with the updated estimate of the drone location to estimate the location of the landmark with respect to the position of the drone at the initial time (point [0, 0, 0]). The newly computed positions of these landmarks are compared to the locations stored in memory, and an Extended Kalman Filter is used to correct for drift in the IMU and reduce uncertainty in the landmark locations. The positions of the new landmarks are computed, and the locations of all landmarks are used to build a map of the environment. Finally, the corrected drone position is sent back to the flight controller.

The images recorded by the main camera are passed to the compute unit. The image is processed using the CNN, which outputs bounding boxes and labels for various objects in the image. If an object is determined to be of interest, the bounding box can be used to crop the large image, and the cropped image can be sent to another algorithm to determine key properties (like color) or sent to more complicated classification algorithms to determine friend or foe. The identities, properties, and locations are then sent back to the compute unit to be stored in memory and marked on the local area map.

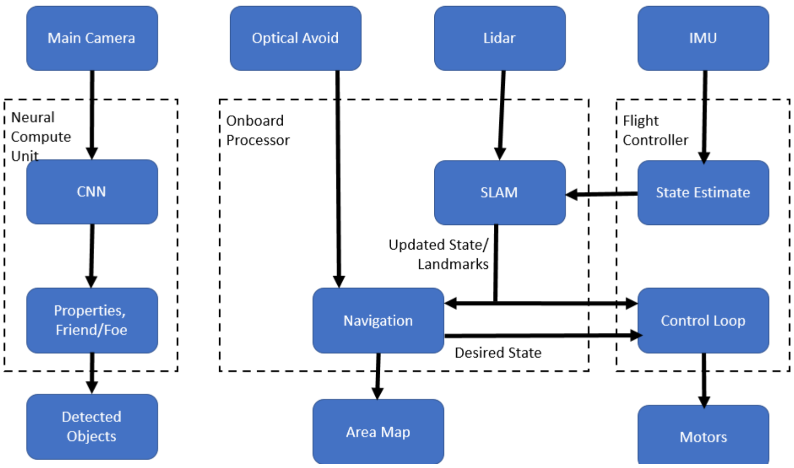

Figure 1: Flowchart of data flow through the various pieces of hardware and software algorithms for a drone with lidar and real-time object detection.

Commented [AP1]: We can have two types of shape in this diagram to identify what is a process and what is an output:
1) Output like Objects, State Estimate could be parallelogram shaped
2) Processes like YOLO, Navigation can remain as they are

Commented [J2R1]: I like that idea

Commented [J3R1]:

Development Path

The first task in developing the drone is to design and code the software. Starting from the system architecture, summarized in Figure 1, the individual modules will be designed and coded. At the same time, the data layout will be designed, taking care to manage stored images, environment maps and program data carefully. Consideration will be given to efficient program execution with the aim of executing critical navigation and collision avoidance functions in real-time, as these will be limiting factors in drone speed. We will develop software to interface with our sensors and software for our main algorithms: navigation, mapping, object detection, object classification. Once all the required programs are created, we can begin to simulate the drone using the AIRSIM environment and UE Unreal Engine Version 4. We will begin by testing our software for robustness in the simulated environment. We would then emulate the flight controller hardware to determine if the flight controller alone is powerful enough to complete the desired tasks. Finally, we would attempt full hardware-in-the-loop simulations to test the real hardware followed by "all-up" tests of the full drone system in a laboratory setting.

Future Development and Optional Hardware

While we intend to complete the challenge using the stock drone hardware, we identified additional components that could be included as payload. A small lidar system can be included to improve the accuracy of mapping and navigation, to measure ranges directly instead of using parallax like the monocular SLAM algorithm. If the stock processing unit is not powerful enough to perform real-time object detection in addition to the other necessary algorithms, we can also include an Intel Neural Compute Stick as part of our payload to accelerate the CNN computations. Table 1 shows the relevant specifications of our option payload components. As the table shows, we could fly both components cost-effectively, with limited impact on battery life, and with plenty of mass left over for mounting hardware.

Table 1: Specifications of the optional payload hardware.

Component	Mass	Power	Cost
Lidar	0.71 oz	0.5 W	$280
Intel Compute Stick	0.5 oz	3 W	$70
Total	1.21 oz	3.5 W	$350

References

[1] K. Tateno, F. Tombari, I. Laina, and N. Navab, “CNN-SLAM: Real-time Dense Monocular SLAM with Learned Depth Prediction,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[2] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *https://arxiv.org/abs/1506.02640* , 2015.

[3] Wyder, Philippe Martin, et al. "Autonomous drone hunter operating by deep learning and all-onboard computations in GPS-denied environments" *PloS one* 14.11 (2019): e0225092.