**image\_detection\_loop():**

This will be the loop that is always running while the drones are idle. This is part of a python program that is using the process of image subtraction to detect intruders. An arbitrary number of 720p HD cameras *n* (between 3 and 6 probably) will be feeding frames into the program. The program compares matrices of pixels using a common image subtraction algorithm to detect a change in the scenery (similar to what is done in this paper **<https://ese.wustl.edu/ContentFiles/Research/UndergraduateResearch/CompletedProjects/WebPages/sp14/JonathanAllenSiwei/ESE498Report_Chiang_Chu_Su.pdf>***)*

**find\_position():**

We are attempting to use a python program that uses image processing as a means of detecting a location from a picture. This will be called after it is confirmed that an intruder is detected. The OpenCV library has been used to complete this task, as proven in this paper <http://sciencejournals.stmjournals.in/index.php/RRJoPHY/article/download/246/746>**.** Using light sources and keeping track of camera angles allows basic laws of triangles to be used to determine a position from pixels in an image.

**find\_drone():**

Find a drone that has a reasonable charge and if possible is close to the detected location of the intrusion. The assumption here is that the PlutoX drone api has a system call to check the battery levels of drones. If not, we will have to develop another way to check charging levels. Finding the closest drone to the disturbance should be no problem thanks to our use of UWB technology.

**feedback\_loop():**

The idea here is that once we have detected an intruder, estimated their location, and chosen a drone, we will use a closed loop system to guide the drone to the desired location. Using the UWB’s and the location tag on the drones, a C++ program will run to constantly to read the current position and then perform some actuation on the drone to move closer to the source. Using the PlutoX api, we should be able to develop a controller that communicates with the drone over wifi. For example, the drone’s location may 20 meters from the intruder in one direction and 15 meters in the other. The drone will be given a command to orient itself in the direction and to move towards it (pitch\_left(), thrust\_forward(), read\_position(), etc.).

**drone\_actuation():**

Once the drone is within a specified distance of the target, some action is performed. This could possibly be taking a picture of an intruder, or blinking a light and sounding an alarm. This will be open ended. After the drone has performed this action, it will fly back to its charging station using the previous feeback\_loop.