Olin College of Engineering

Olin Aerial Robotics Team

Control of a Multirotor Swarm Through Guided Autonomy

International Aerial Robotics Competition

Maximillian Schommer, Eric Miller, Paul Nadan, Lydia Zuehsow
Franklin W. Olin College of Engineering

11th Annual Symposium on Indoor Flight Issues

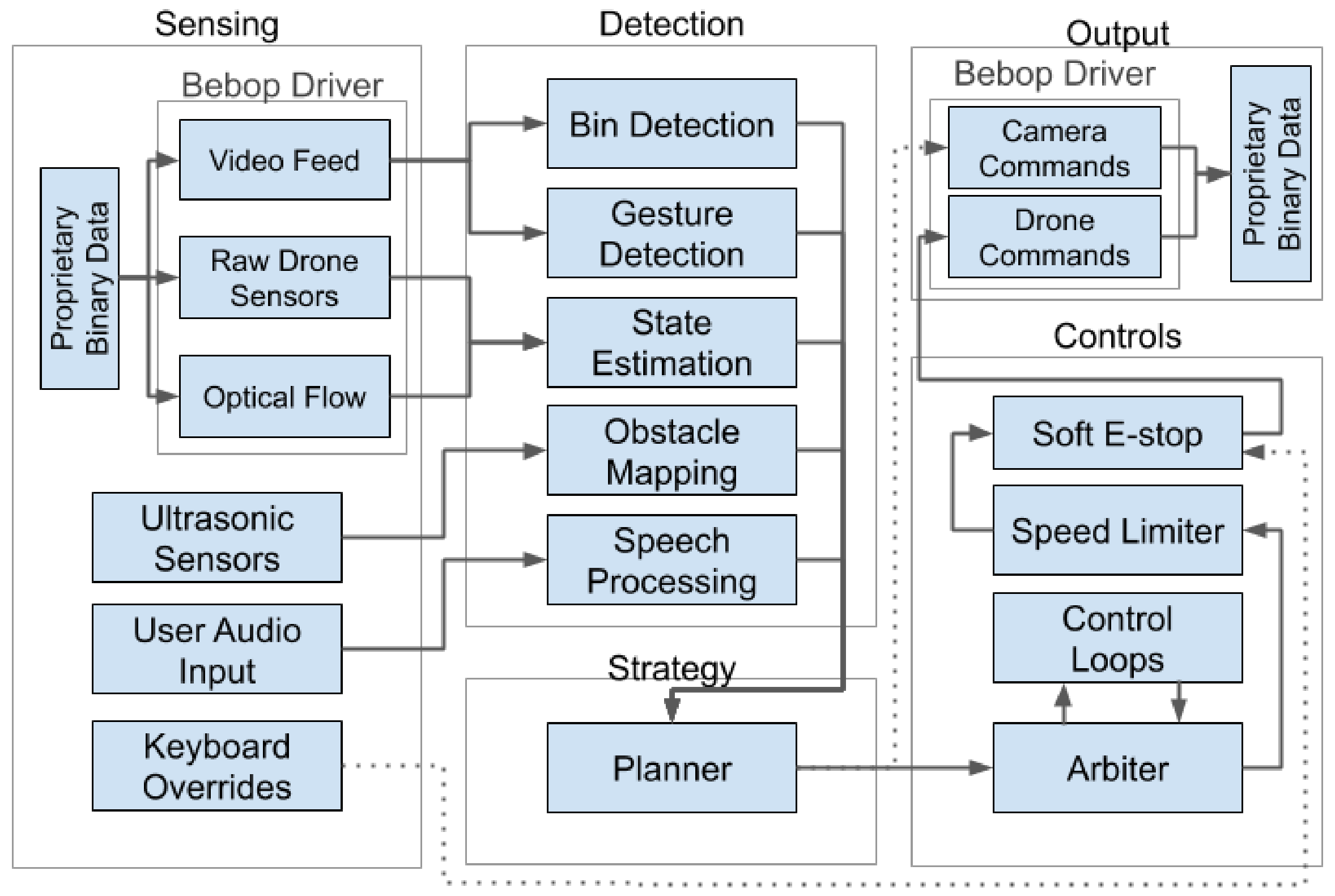
Project Objective

Create an integrated, intelligent robotic system that can…

- Avoid static and mobile obstacles
- Navigate indoors without the aid of GPS
- Simultaneously support four cooperative air vehicles led by a human via voice or gesture
- Locate known objects in unknown geography
- Identify and decipher a quartered QR code

System Overview

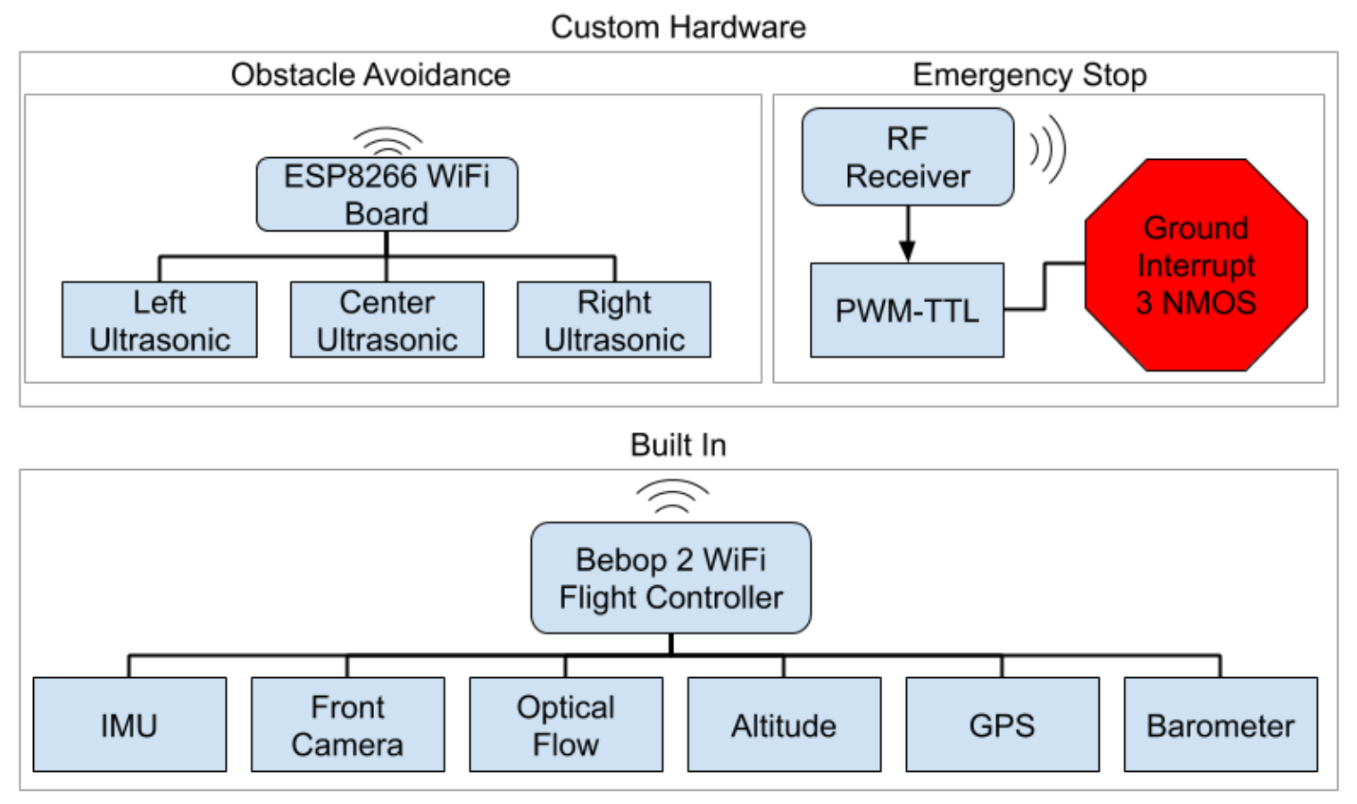
- All processes are run on a single computer
- The main execution thread processes operator voice commands, and dictates the chosen behaviors
- Each vehicle has an assigned thread, which is responsible for locally autonomous behaviors



Overall System Architecture

Platform

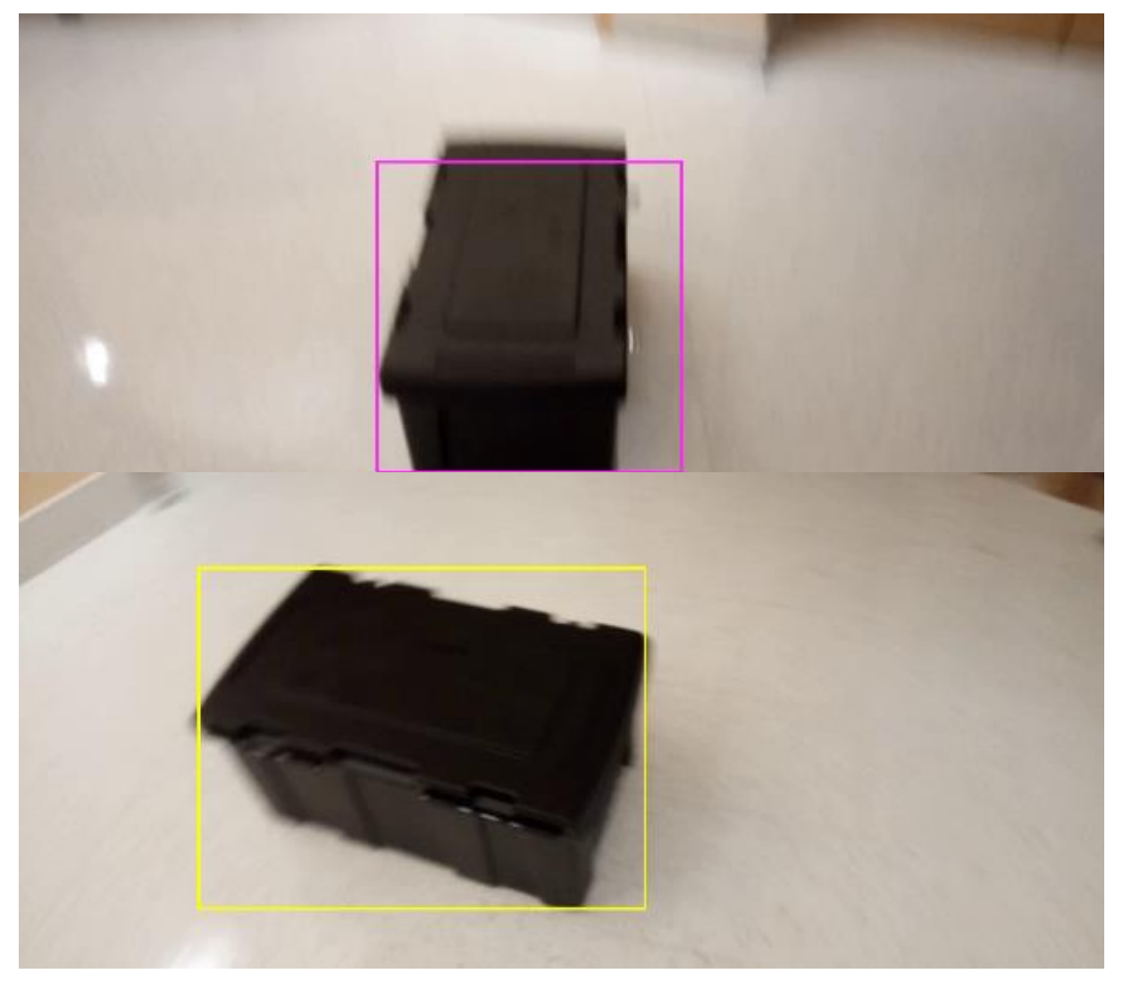
Parrot Bebop 2 Drone



Sensory output configuration

- Four Parrot Bebop 2 quadrotor drones
- Downward camera for optical flow & odometry
- Forward gimballed camera for object identification & tracking
- Custom E-STOP & obstacle avoidance package
- Sensor data streams via WiFi

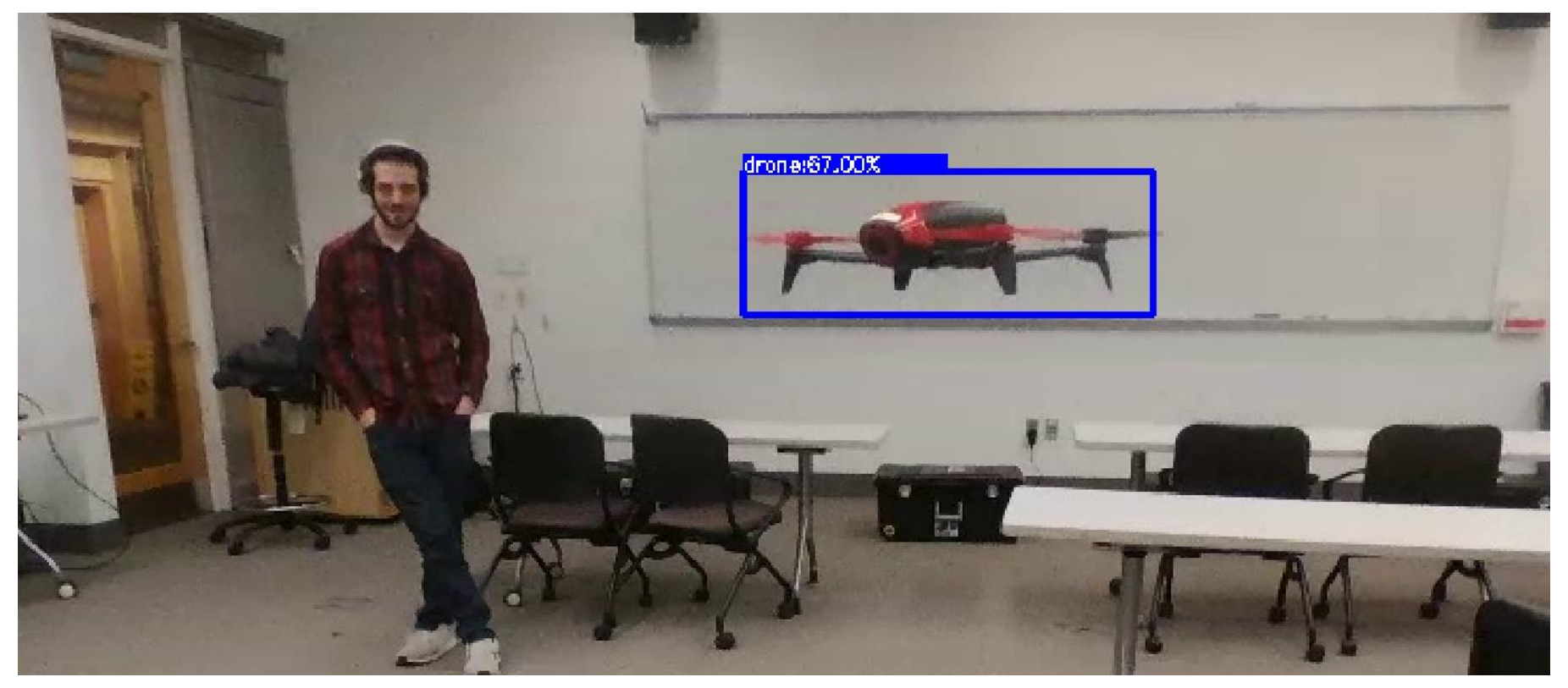
Target Identification



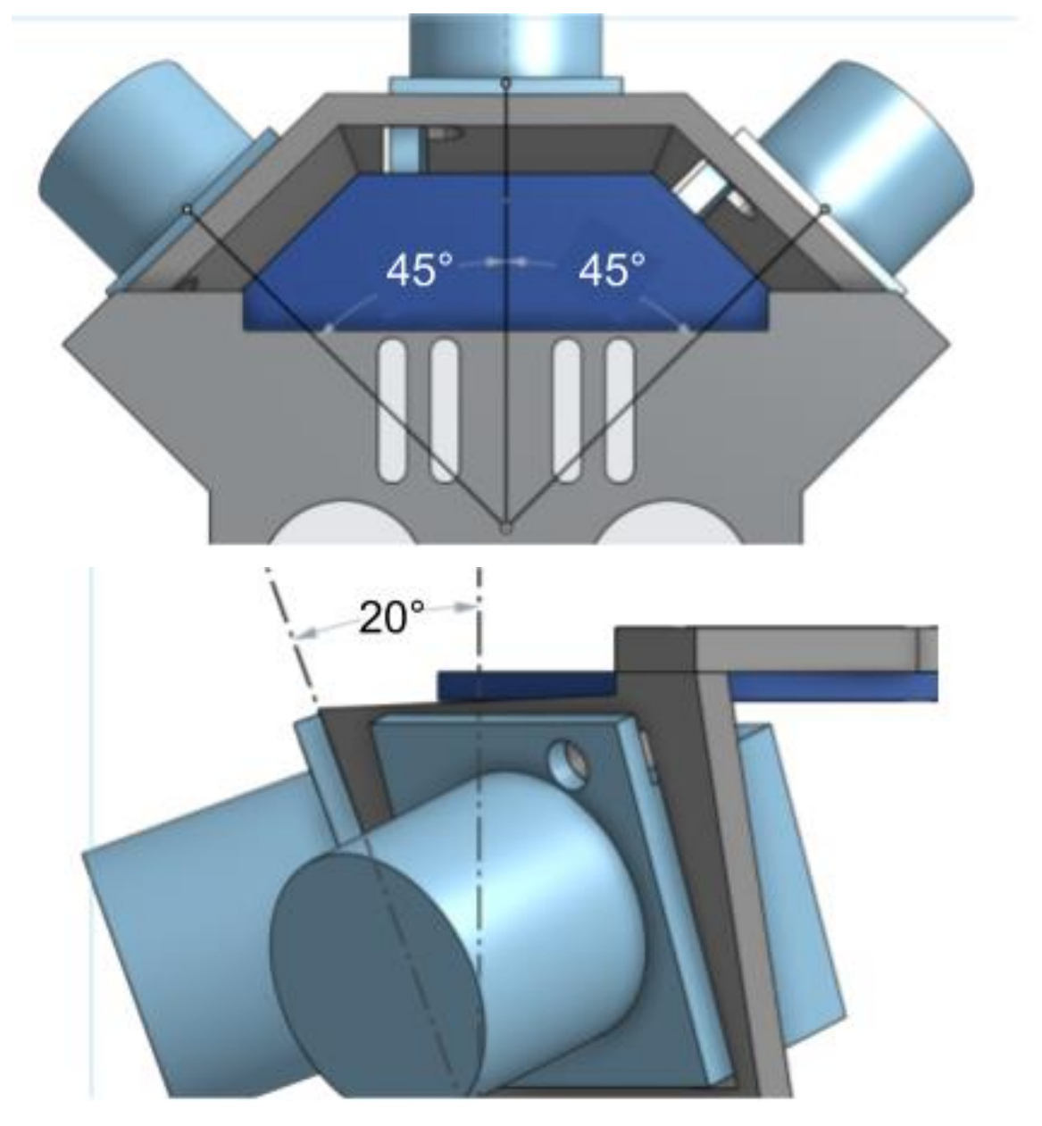
Bin detection result

- Bins are initially detected by a Harr Cascade classifier, after which a Distractor-aware Siamese Network is used for continued tracking
- Training data for the classifier was generated by applying several automated labeling tools to the original video footage, including GrabCut, adaptive thresholding, color segmentation, and edge detection
- QR quadrants are then located by color thresholding the camera image and searching for white rectangular contours
- The region inside the contour is rotated, cropped, and thresholded to produce a clean QR quadrant
- After all 4 QR quadrants have been located, they are stitched together in each possible permutation and checked for a valid code

- Human and drone detection are both carried out by a single deep neural-network based object detector implemented with the TensorFlow object detection API
- The final network architecture employs a variant of the Single-Shot Multibox Detector
- Only drones with high confidence values are initially selected, followed by a second inference performed on cropped regions of the image from lower-confidence detections of the previous step



Drone detection result



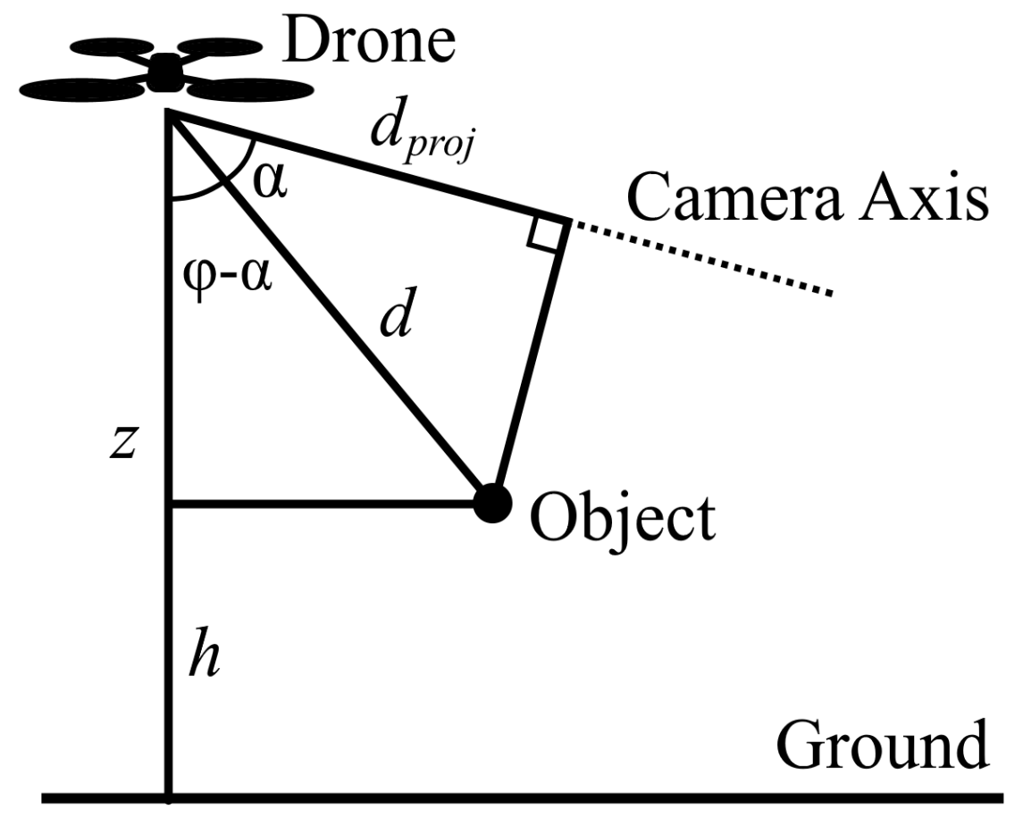
Rangefinder Mount Design

Threat Avoidance

- A trio of front-facing rangefinders provide general obstacle detection
- The rangefinders are spaced 45 degrees apart, ensuring over 135 degrees of coverage with some overlap
- Objects detected by the rangefinders are transformed into a list of obstacle locations in global coordinates shared among the entire swarm
- A potential-field gradient approach is used to simultaneously navigate each drone away from obstacles and towards its desired position

Human-Machine Interface

- A pointing gesture directs a drone to move a set distance in the given direction when accompanied by the proper voice command
- The human operator's arm is found via color detection, aided by distinctive colored patches worn on the human's wrist and shoulder
- The locations of the colored patches are transformed into points in 3D space, then the vector between them is projected onto a 2D plane



Pointing Gesture Geometry

Target	Action	Parameters	Desired Behavior
Swarm, Alexa, Google, Siri, Clippy	North/East/South/West	[number] m/cm/in/ft	Move in a cardinal direction
	Forward		Move in the direction drone faces
	Follow		Move in the direction of a gesture
	Jump/Duck		Increase or decrease altitude
	Look	North/East/South/West	Turn to face a cardinal direction
	Turn	Left/Right [number]	Rotate by a given angle (yaw)
	Takeoff/Land	N/A	Takeoff or land
	Stop	N/A	Hover in place
	Picture	N/A	Take an image of a QR code

Voice Command Interface

- Voice recognition is carried out by the CMUSphinx engine, implemented via the SpeechRecognition library
- The command syntax is specified by a Java Speech Grammar Format
- A wireless microphone is worn by the human operator to detect voice commands

Flight Control

Odometry	We primarily use the existing onboard odometry, but cross compare across all platforms to improve accuracy
Arbiter	The arbiter makes low-level behaviors available as discrete Rostopic commands, facilitating usage by higher-level behaviors
Soft E-STOP	Power to motors is stopped in software via the Wifi channel
Hard E-STOP	The main ground line of the battery is interrupted by 3 parallel power NMOS, which can be triggered via a radio signal to physically cut power

Risk Reduction

EMI/RFI Solutions

- Bebop 2 drones are internally protected from EMI
- We ran additional communications on WiFi to prevent interference
- We used a frequency hopping RC controller and receiver to eliminate any ESTOP interference between vehicles

Shock/Vibration Solutions

- Bebop 2 rotor chassis is isolated from the sensor suite by vibration dampening rubber
- We designed flexible, finger-safe propeller guards, which prevent damage to the vehicle upon a crash
- We set a maximum speed on all drones, to prevent hazardous flight or high-speed impacts

Simulation and Physical Testing

- We developed multiple simulations with differing levels of complexity to test our software without costly and dangerous crashes
- We verified that prop guards do not shatter on collisions, and live tested the majority of our command and perception code

Conclusion

- We have created a drone swarm to aid a human operator through guided autonomy
- The swarm is capable of autonomous flight, perceiving and avoiding threats, identifying and station keeping over vision targets, and responding to voice and gesture commands
- The system has performed well in both simulation and physical testing

Acknowledgments

We would like to thank all of the generous donors who made this whole endeavor possible. We are also grateful to Olin College of Engineering for the use of their space and resources.