HawkEye

Design Review Presentation

Vedant Parekh, Siddesh Nageswaran, Alvin Shek
Advisors: Professor Savvides, Rashmi Anil

Introduction and Motivation

- What is Hawkeye?
  a. An automatic drone tracking system with live aerial footage
  b. Shoots aerial video that does not require human control, eliminating human error

- Use Cases:
  a. Useful for recreational filming, rescue missions etc.
  b. Imagine shooting exciting videos of sports events (your Turkey Bowl game), or having a hands-free vlogging experience!

HawkEye

Project Scope

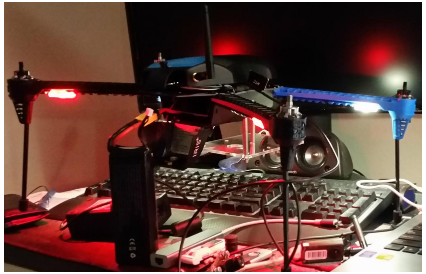
- We will operate under the assumptions:
  - The drone will operate in open field without obstacles
  - There will be WiFi access in test environment
  - There will be little to no wind
  - The drone will operate during daytime
  - The drone will be limited to tracking one person
  - The target being tracked may be amongst 3 other people, but not more
  - Flight height no more than 20 feet

Solution Approach– Hardware

- Drone Compute
  - IRIS 3D+ Drone
  - Arducam IMX477 MINI Camera for RPi
    - 20’ x 30’ field of view at 20’ elevation
  - RPi 3
    - Interfaces between camera / drone / TX1
  - 5100 mAh Battery
    - Powers drone by itself for 20 - 30 minutes
  - 5V Buck Converter (to power RPi)

- User Compute
  - Jetson TX1
    - 256 GPU cores allows relatively complex CV compute within 15 fps time requirement
  - 3S LiPo Battery
    - 11.1V (within 5 - 19V operating range of TX1)

- Wearable Display
  - Adafruit HDMI 5” Display
    - 4.8” x 3”
  - Buttons and Passives



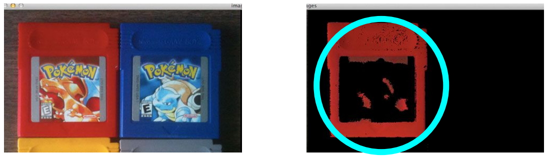
HawkEye

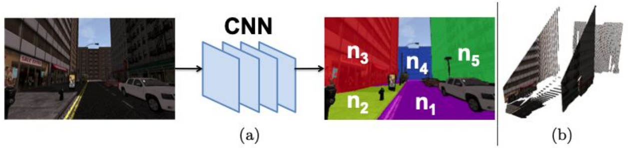
Solution Approach– Software

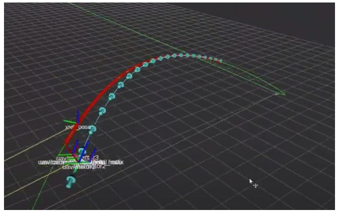
- Target Tracking (done on Jetson TX1)
    - Detect target with CV: HSV color filter + blob detection
    - Estimate velocity vector of target

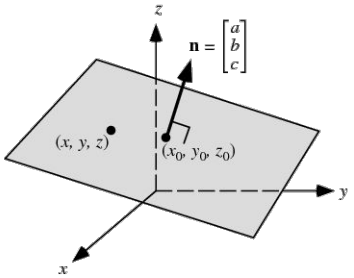
- Drone flight control (done on Jetson TX1)
    - Need to compute desired yaw and x, y and z positions relative to current drone position

- Communication between board and drone (handled on RPi)
    - Need to communicate the yaw and coordinates to the drone's flight controller and send the streamed video to the Jetson

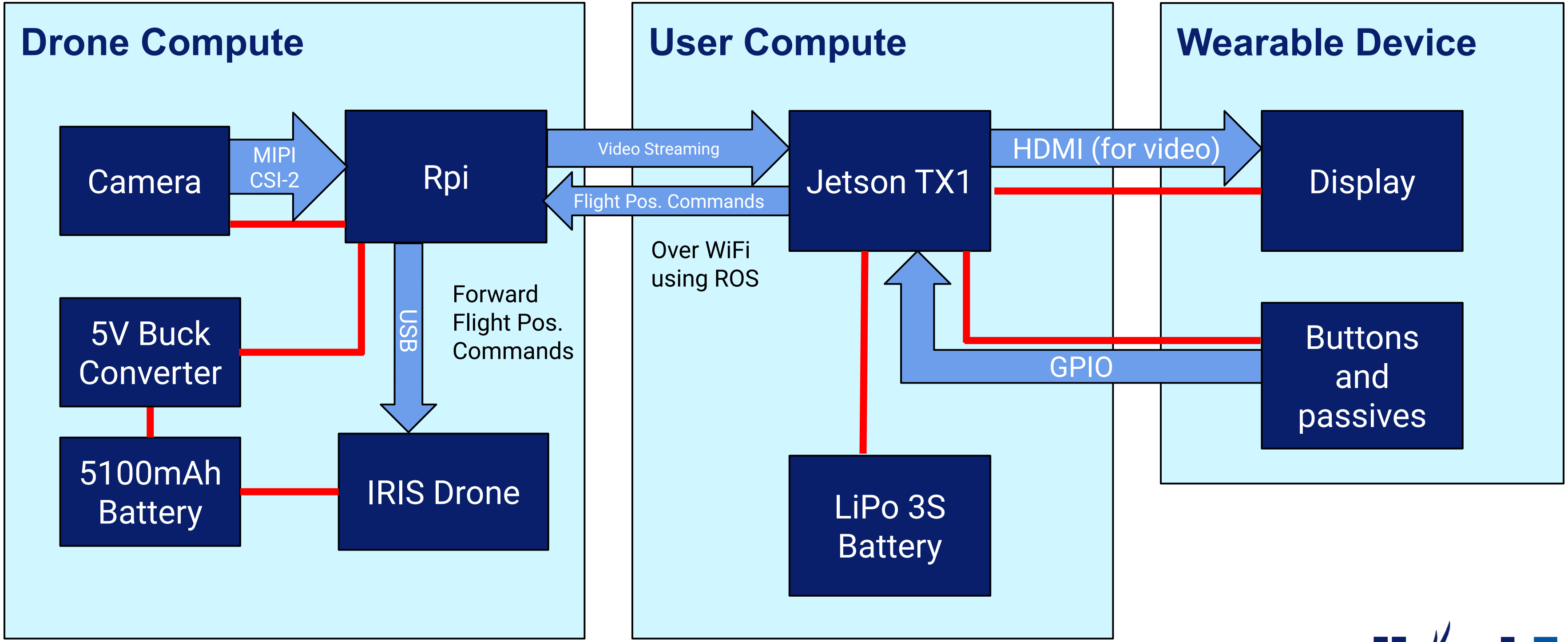








System Specification



Implementation Plan

Drone Compute

Camera
- **Purchase**
- **Design** housing to mount to drone

Rpi
- **Reuse**
- **Download** ROS

5100mAh Battery
- **Purchase**

IRIS Drone
- **Reuse**
- **Download** ROS

5V Buck converter
- **Reuse**

User Compute

Jetson TX1
- **Reuse**
- **Download** Pytorch, OpenCV and ROS
- **Design** case

LiPo 3S Battery
- **Purchase**

Wearable Device

Display
- **Purchase**
- **Design** housing for display / buttons

Buttons and passives
- **Reuse**
- **Prototype** on breadboard
- **Design** on through hole PCB

HawkEye

Testing: Target Detection

- **Measurements:**
  - Standard deviation in distance between true and predicted target position
- **Test scenarios:**
  - 5 different colored shirts, bright and shaded locations
- **Test setup:**
  - Target wears a specific colored shirt
  - Stationary camera on a high platform to record video for 1 minute
- **Test evaluation:**
  - Hand-label the center of target for each video frame in post-processing
  - Run target detection algorithm on each video frame to output predicted target position in pixel coordinates
  - Measure standard deviation across entire video

Testing: Drone Stability and Tracking

- **Measurements:**
  - Standard deviation in distance between target center and center of frame across 30 frame windows
- **Test scenarios:**
  - Stationary, forward motion, sideways motion, free-motion
  - Single color
- **Test setup:**
  - Simulation: Use a colored cube as target with nominal lighting conditions
  - Real world: Target wears specific colored shirt and follows specific motion
- **Test evaluation:**
  - Drone runs target detection and motion planning to follow target
  - Record video, split into windows of 30 frames
  - Hand-label target position at each frame, calculate measurements

Risk Mitigation

- Unable to detect target
  - More defined trackers (switch to brighter object)
  - Reduce height of drone

- Target identified but unable to track accurately
  - Modify target state estimation -> bias more towards pre-existing model of target motion rather than frame by frame data
  - Thresholding to discard sudden “jumps” in target position
  - Better camera

- WiFi drop
  - Land drone when WiFi stops working
  - Explore bluetooth

- Worst case
  - Drop out

HawkEye

Division of Labor

- Alvin:
  - Drone motion planning
  - Flight controller interface

- Vedant:
  - Target identification
  - Hardware communication protocols
  - Design circuitry for wearable

- Siddesh:
  - Target state prediction / estimation
  - Designing the housing for the TX1, display and drone

HawkEye

Updated Schedule

		Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
1	**Phase 1: Design**	■	■	■								
1.1	Make overall block diagram	Together										
1.2	Choose/finalize components	Together										
1.3	Set-Up Jetson TX1	Together										
1.4	Order components		Together									
1.5	Familiarize ourselves with drone API and simulator			Together								
1.6	Familiarize ourselves with Jetson TX1 API			Together								
1.7	Familiarize ourselves with wearable display communication protocol			Together								
1.8	Design Presentation		Together									
1.9	Design Report			Together								
2	**Phase 2: Pre-integration**				■	■	■					
2.1	Implement color filtering and blob detection				Vedant							
2.2	Test detection on static images					Vedant						
2.3	From target data across multiple frames, calculate velocity vector				Siddesh							
2.4	Use calculated velocity along with a model for target movement to accurately predict future movements					Siddesh						
2.5	Test detection on live video (long distance with smaller target since blob detection parameters need to be tuned)						Siddesh					
2.6	Successfully send sample motion commands to drone in simulation, then on physical drone				Alvin							
2.7	Design a general motion planning stack; design to easily integrate with target tracking					Alvin						
2.8	Test and debug the motion planning stack						Alvin					
2.9	Implement communication between camera and RPi Zero, film sample video					Vedant						
2.10	Implement video streaming over Wifi between RPi and TX1						Vedant					
2.11	Implement hardware protocol (ex. UART, I2C) to interface between TX1 and display						Vedant					
3	**Phase 3: Integration**							■	■			
3.1	Map target's motion in video to desired motion of drone							Alvin				
3.2	Design circuitry for the wearable device (ex. display, buttons, power supply etc.)							Vedant				
3.3	Hook up buttons to the TX1 and configure them to send start and stop instruction to RPi								Vedant			
3.4	Integrate RPi start stop signals received from nano with the drone flight controller							Siddesh				
3.5	Design safety fallback behavior and validate on simulation and on drone							Alvin				
3.6	Integrate the flight parameter data generated by TX1 with the flight controller								Alvin			
3.7	Create housing for Jetson TX1 and display so that they can easily be worn by the user								Siddesh			
3.8	Create chassis to house RPi and camera onto the drone and verify camera angle								Siddesh			
4	**Phase 4: Performance testing**									■	■	
4.1	Verify that video streaming performs with a stationary target									Together		
4.2	Verify drone can continuously track target during flight w/o streaming video ll									Together		
4.3	Verify that drone can match speed of target that is varying between speedsll									Together		
4.4	Verify live video from drone can be seen on wearable display for a moving targetll									Together		
4.5	Iterate and repeat tests ll									Together		
4.6	Slack										Together	
5	**Final Report**										■	■
5.1	Record demo video ll										Together	
5.2	Edit video ll										Together	
5.3	Final presentation ll											Together

Legend:
- Vedant (red)
- Alvin (yellow)
- Siddesh (blue)
- Together (green)

HawkEye