SwimView 2.0

An autonomous camera tracking system for SwimView

Harrison Burch
Demetris Coleman
Avion Foreman
Jungihn Kim
Dustin Spencer
William Stewart
Nicholas Thompson

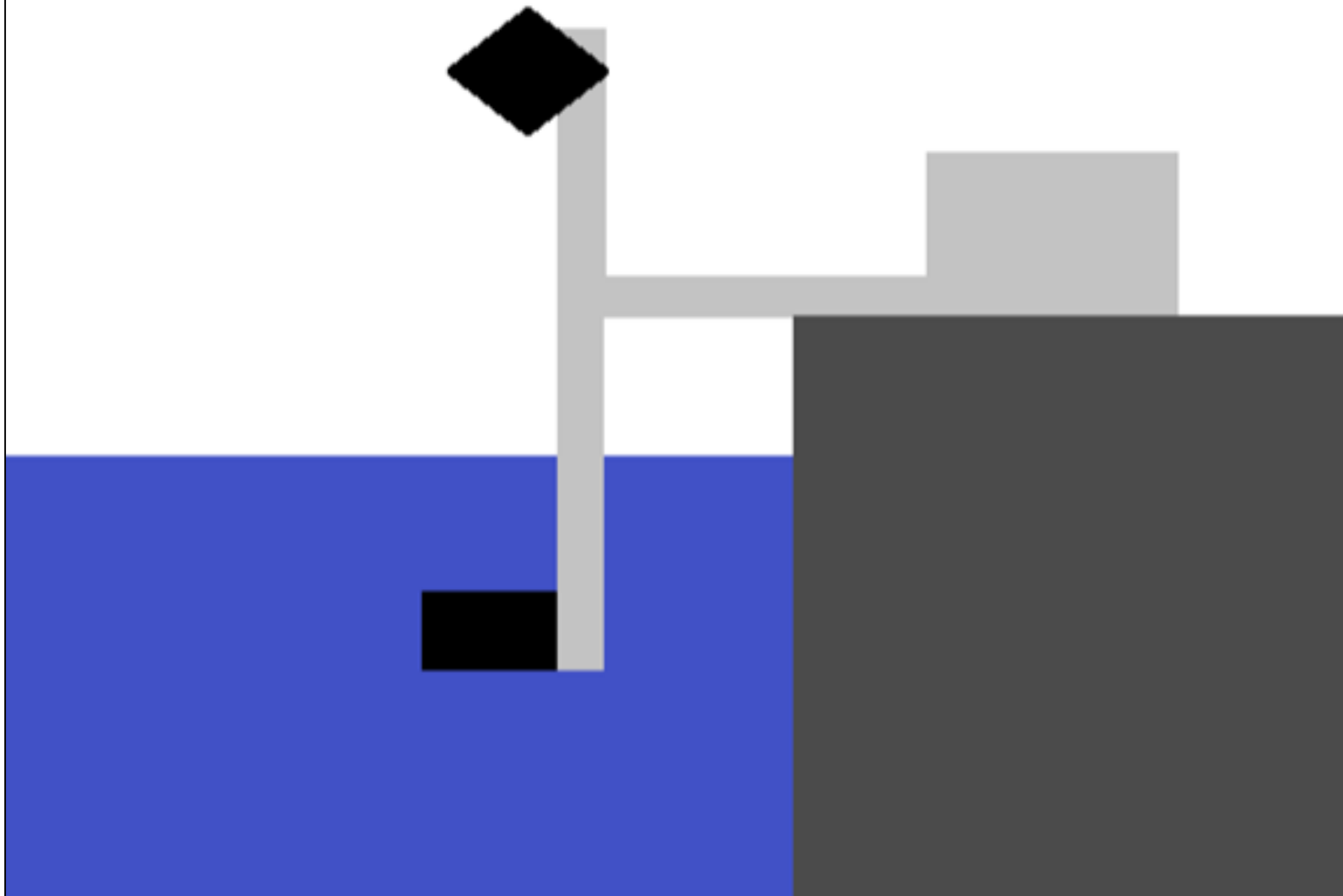
Review of Existing System

- Remote controlled camera system
- Motorized cart
- Marshall CV200-MB camera
- Xbox controller
- Live-stream and Recording capability

Objectives

- Make system completely autonomous
  - Develop camera tracking system
  - Develop autonomous controls system
  - Hardware Integration

•Fix slipping of cable/pulley drive system



Team Responsibilities

- Software
  - Data acquisition
  - Image processing
  - Controls
- Hardware
  - Mounting new arm/camera
  - Integration of new microcontroller

Team Member	Primary Responsibility	Secondary Responsibility
Harrison Burch	Controls	Image Processing
Demetris Coleman	Image Processing	Controls
Avion Foreman	Controls	Hardware Integration
Jungihn Kim	Hardware Integration	Image Processing
Dustin Spencer	Image Processing	Controls
William Stewart	Controls	Project Management
Nick Thompson	Project Management	Hardware Integration
Dr. Robert Dean	Technical Advisor	
Dr. Wendi Weimer	Kinesiology Department Liaison	

Image Tracking Hardware

- Raspberry Pi 3
  - 1.2 GHz Quad-Core ARM
  - 1GB DDR2 RAM
  - 5V, 2.5A
- Raspberry Picamera
  - 8 Megapixels
  - 1080p/720p
  - 640 x 480
- PVC arm

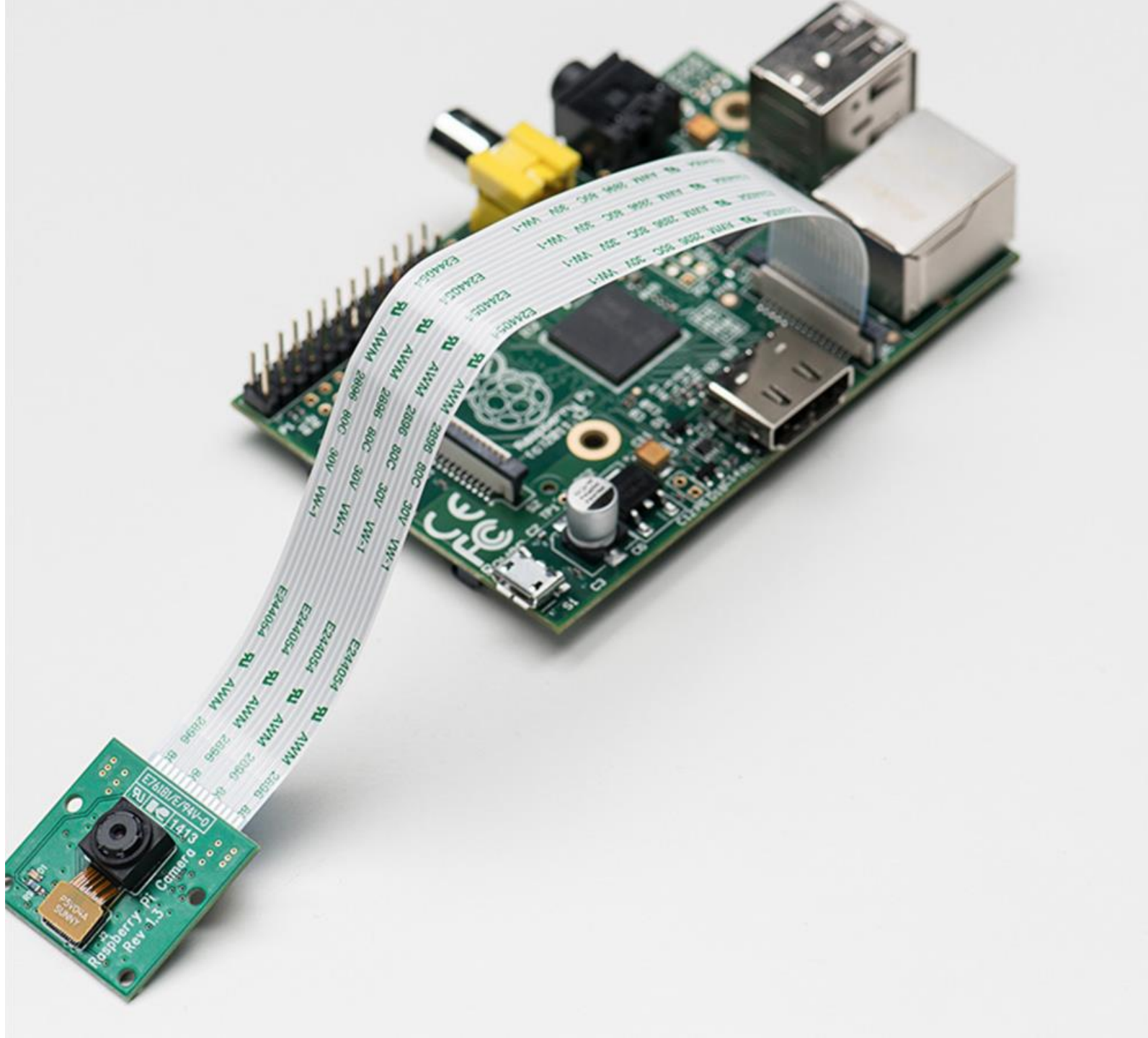


Image Tracking Hardware

- Arduino Uno
  - 16 MHz Core
  - 5V Supply
  - 14 Digital I/O Pins
  - 6 PWM Outputs
  - 32 KB Flash Memory

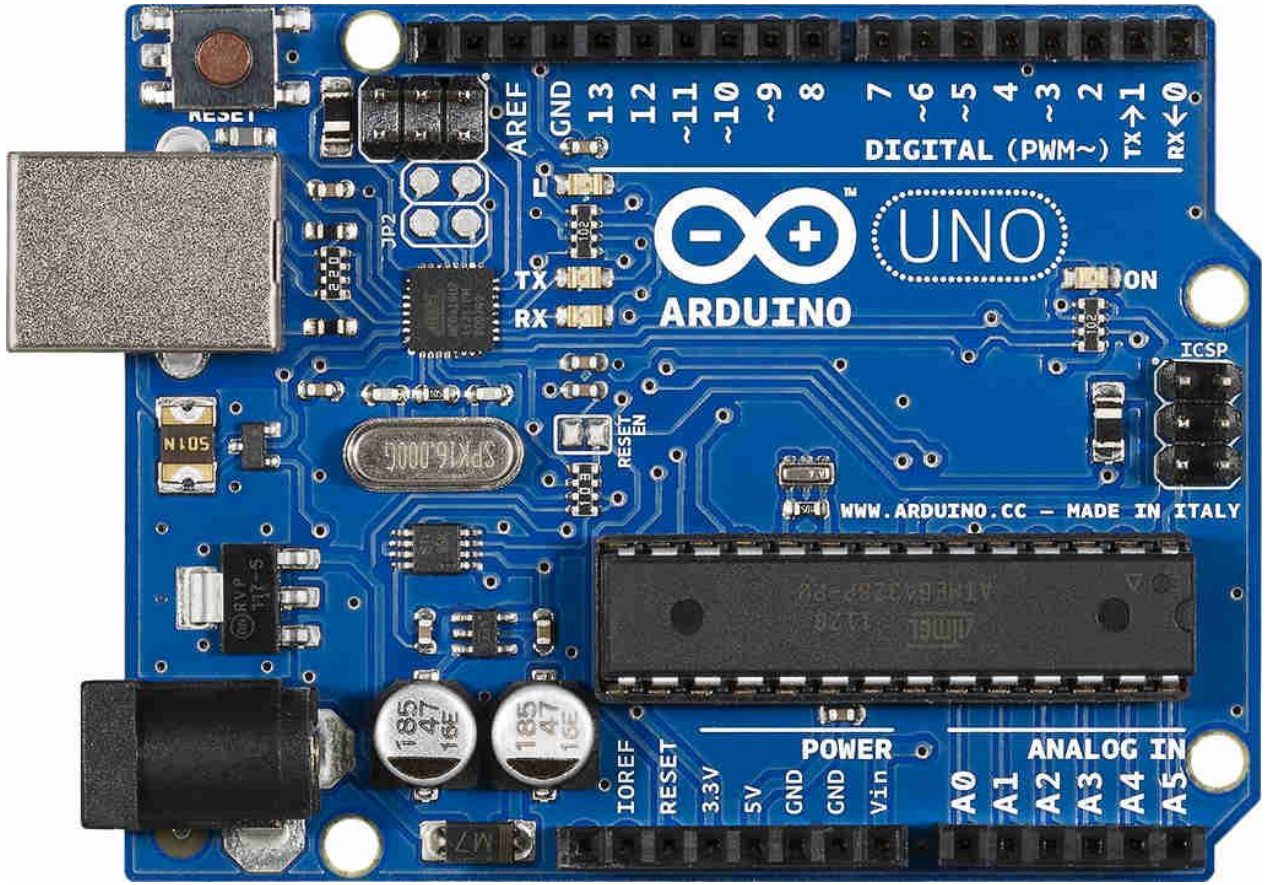


Image Tracking Software

- Python
- OpenCV
- Roborealm

OpenCV vs Roborealm

- Prebuilt functions
- Compatible with pi
- Uses existing hardware
- Simpler control system
- No power issues

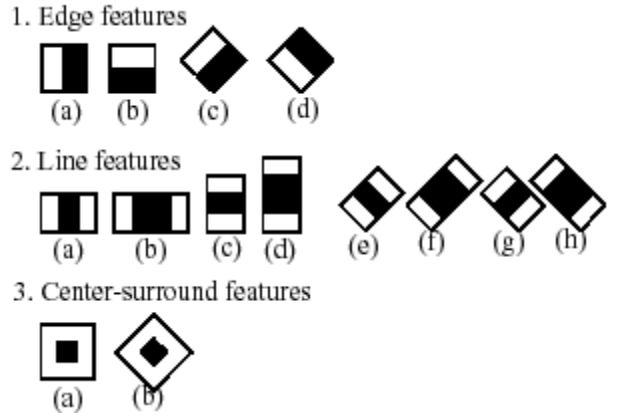
- Prebuilt filters
- Incompatible with pi
- New hardware
- More complex control system
- Power issues

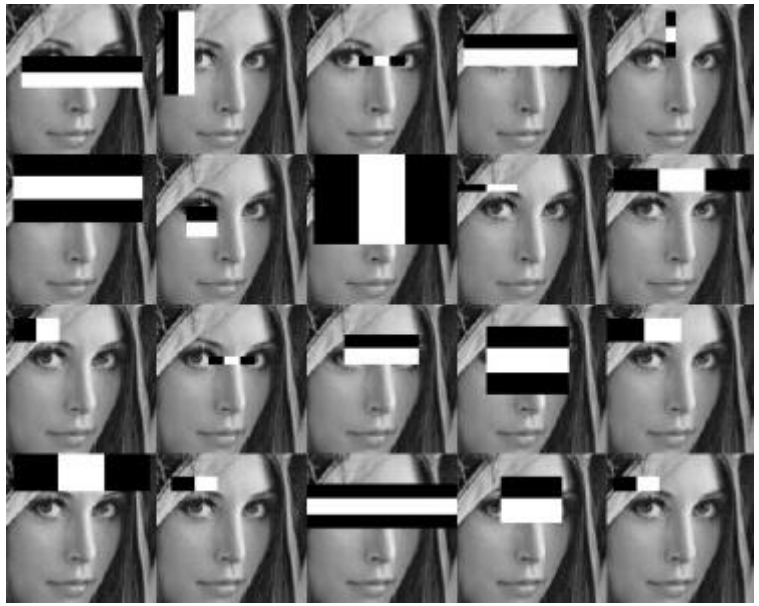
Data Acquisition

- Recording of swimmers
- Different markers and color schemes
- Pictures for training classifiers
- Video for testing tracking algorithm and control system
- Concerns: marker location, depth of swimmer

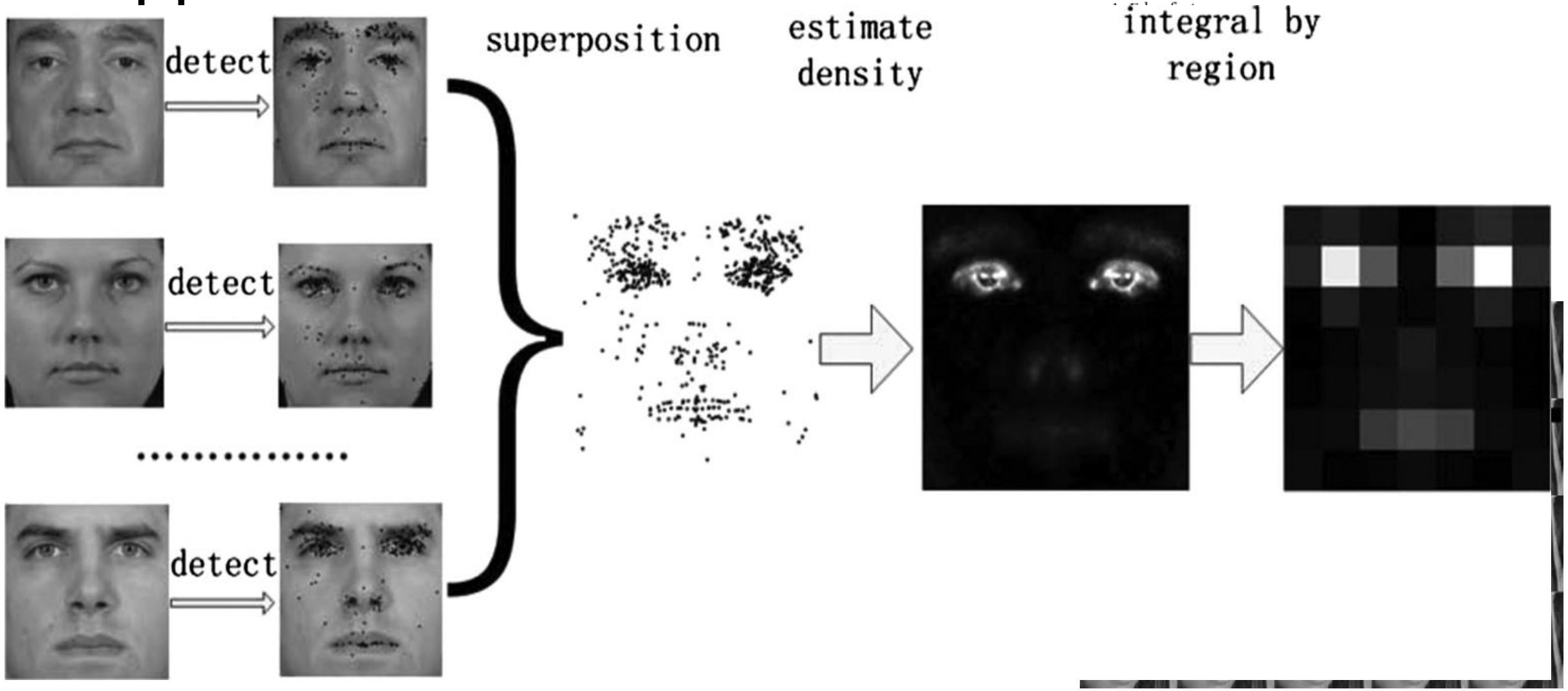
Approach 1: Cascade Classifiers

- Haar Classifiers vs Local Binary Patterns
  - Swimmer’s body
  - Simple shape
  - Complicated Shape



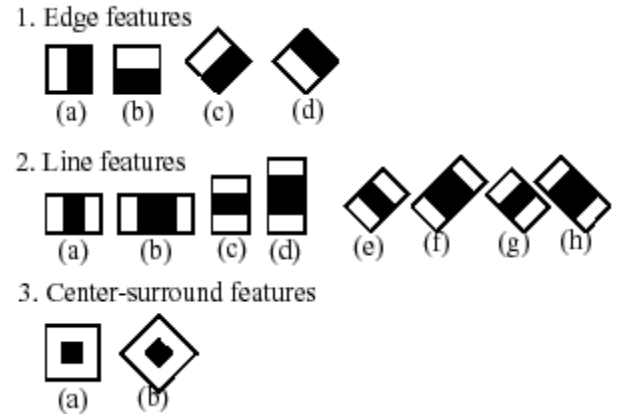


Approach 1: Cascade Classifiers

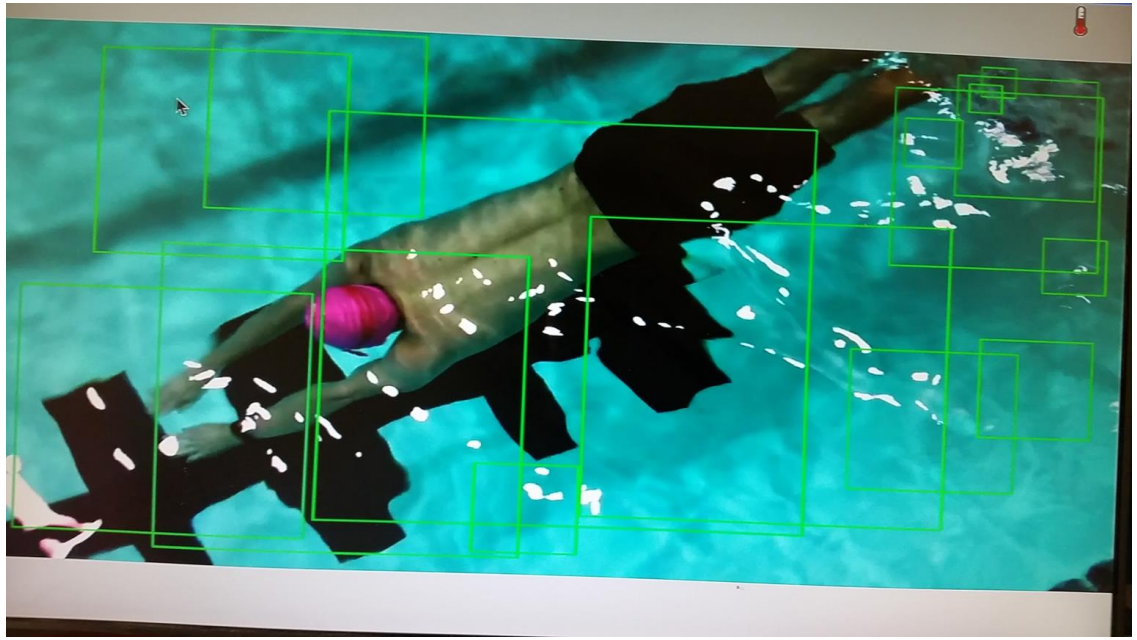


Approach 1: Cascade Classifiers

- Haar Classifiers vs Local Binary Patterns
  - Swimmer's body
  - Simple shape
  - Complicated Shape



Haar Classifier Results



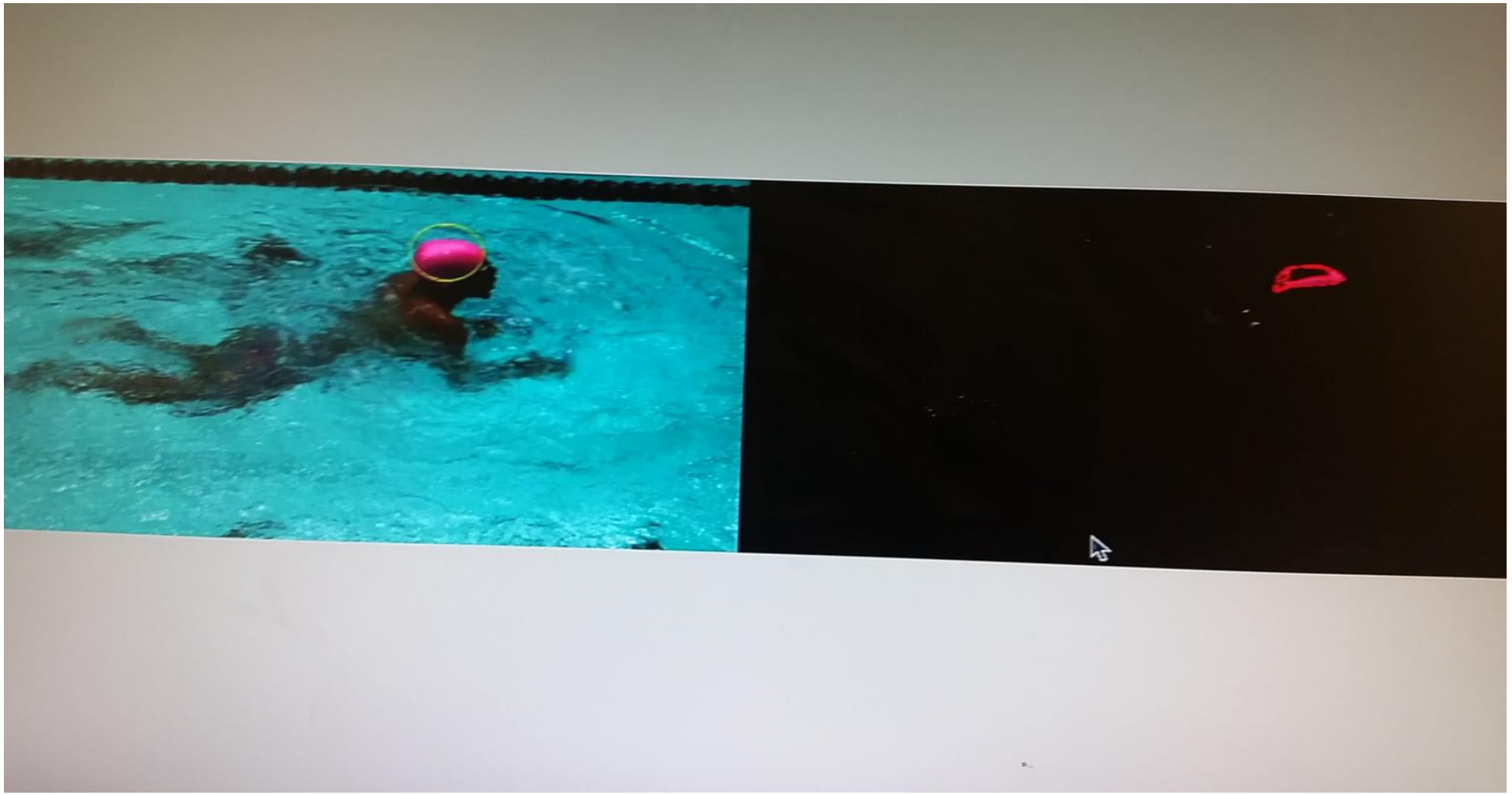
Haar Classifier Results

Approach 2: Color Tracking

- Blob Detection
  - group of connected pixels in an image that share some common property
- Contour
  - a curve joining all the continuous points along the boundary with the same color or intensity

Blob Color Tracking Results

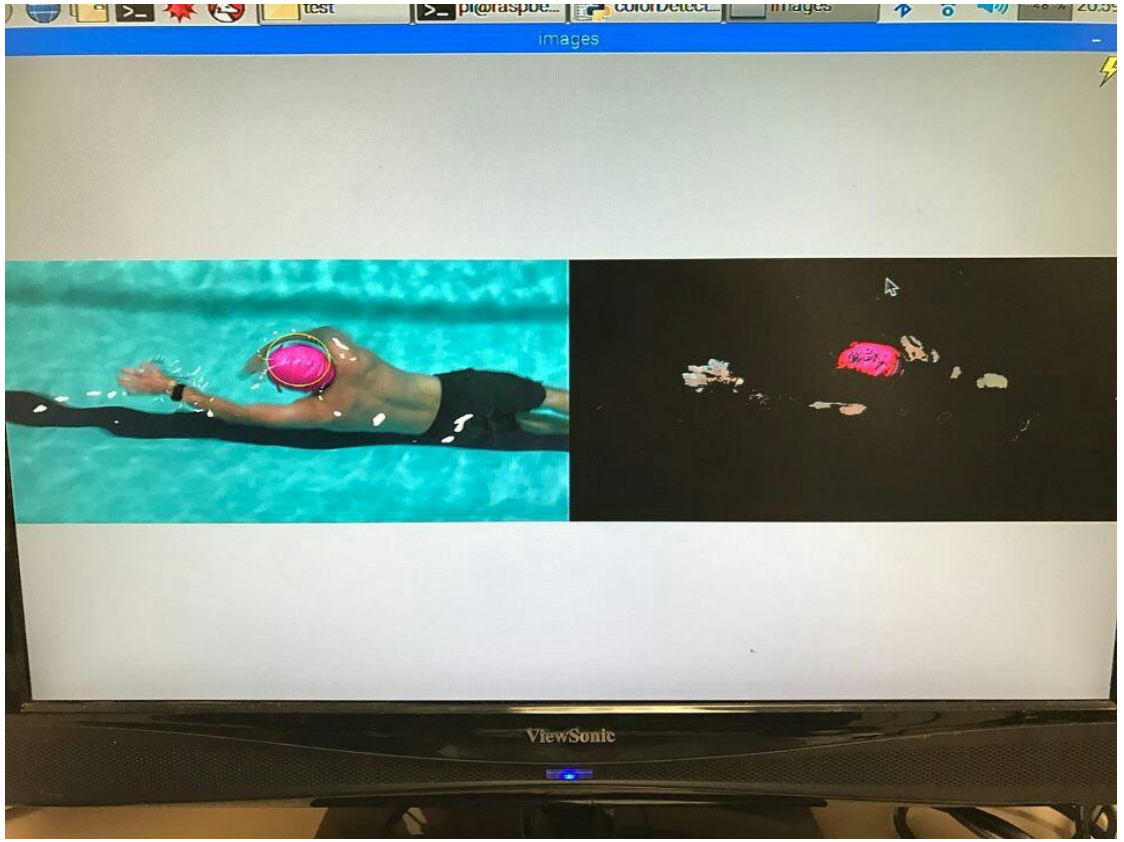
Contour Color Tracking Results



Practical Limitations

New Tracking Method

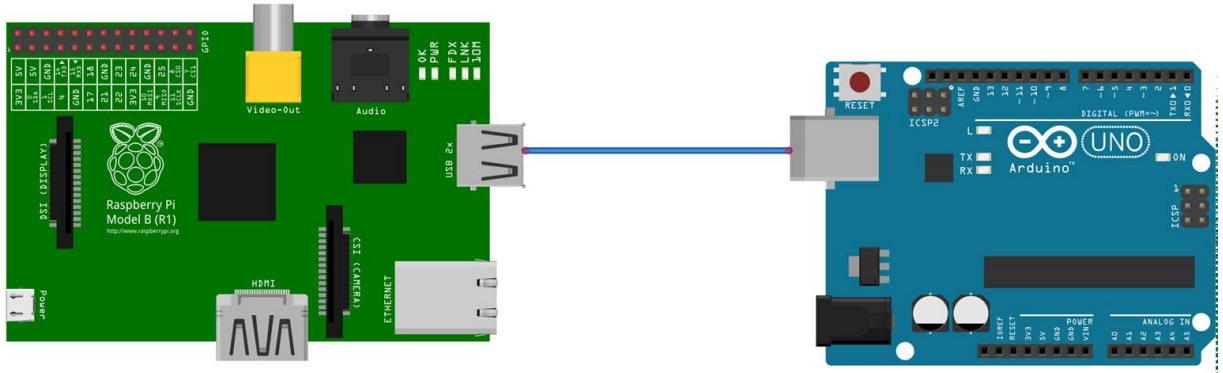
- Swimmer Identification
  - Wear colored band around waist/torso
  - OpenCV detects this color, and calculates its center of gravity (CoG)
  - The x-axis value of the CoG is sent to the motor control system



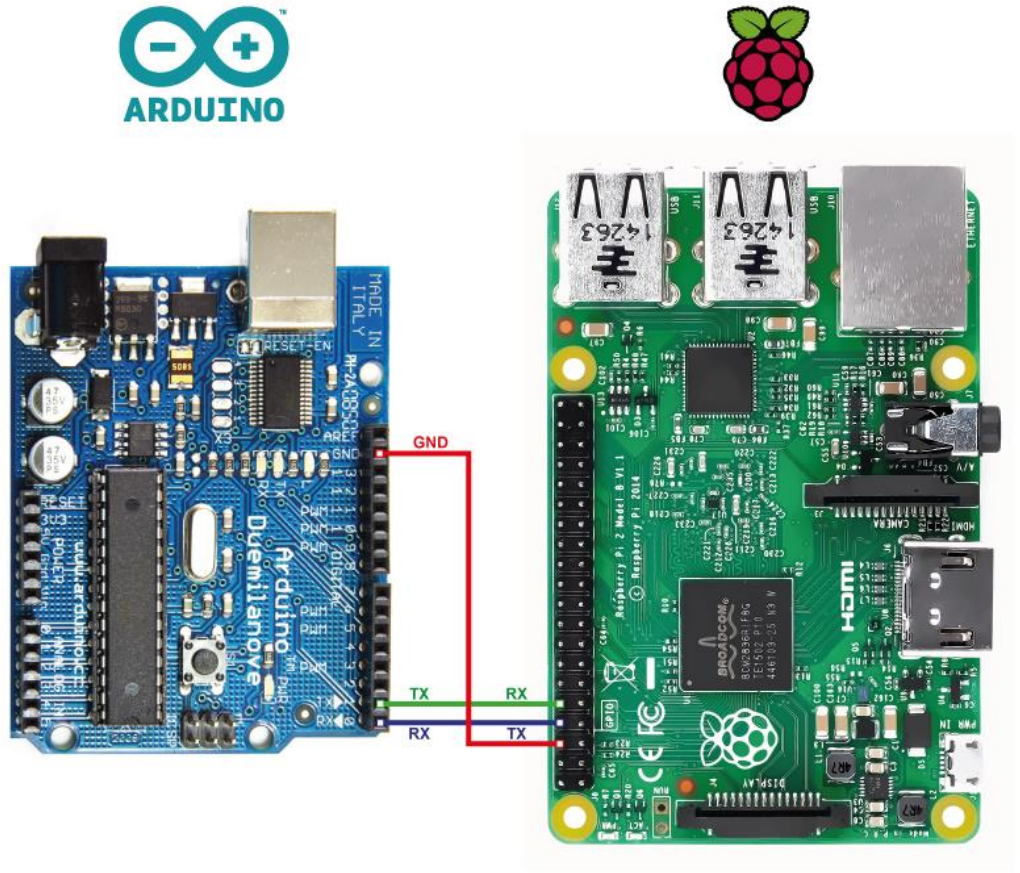
Algorithm Development (Control System)

Possible Method 1

- Raspberry Pi to Arduino
  - RPi generates integer 0-255 based on location of swimmer
  - This value is sent to Arduino via serial communications once per frame
- Arduino to Sabertooth
  - Arduino controls operation mode
  - Uses serial data from RPi to create PWM signal
  - The PWM signal is sent to Sabertooth via low-pass filter



Algorithm Development (Control System)



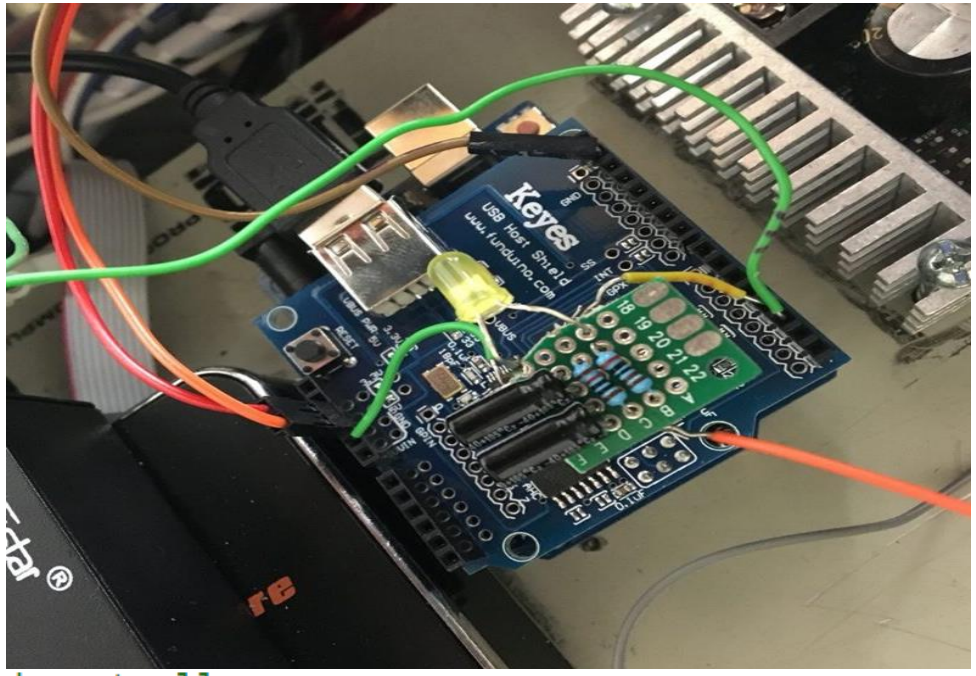
Possible Method 2

- Raspberry Pi to Arduino
  - Detect location of swimmer
  - Determine speed and direction
  - Set GPIOs based on these parameters (6 pins used)

- Arduino to Sabertooth
  - Arduino monitors GPIOs
  - Uses status of GPIOs to create PWM signal
  - The PWM signal is sent to Sabertooth via low-pass filter

Control System

- Manual Control
  - Xbox Controller to Arduino
    - Radio Communication
  - Arduino to Sabertooth
    - Analog Signal
  - Implement Proportional control




```
  //read controller
  if (Xbox.getAnalogHat(RightHatX, i) > 7500) {
    curLvl += increment;
    change = true;
  } else if (Xbox.getAnalogHat(RightHatX, i) < -7500) {
    curLvl -= increment;
    change = true;
  }
}
//filter output to be inbounds before write
if (curLvl < minLvl) { curLvl = minLvl; }
if (curLvl > maxLvl) { curLvl = maxLvl; }
```

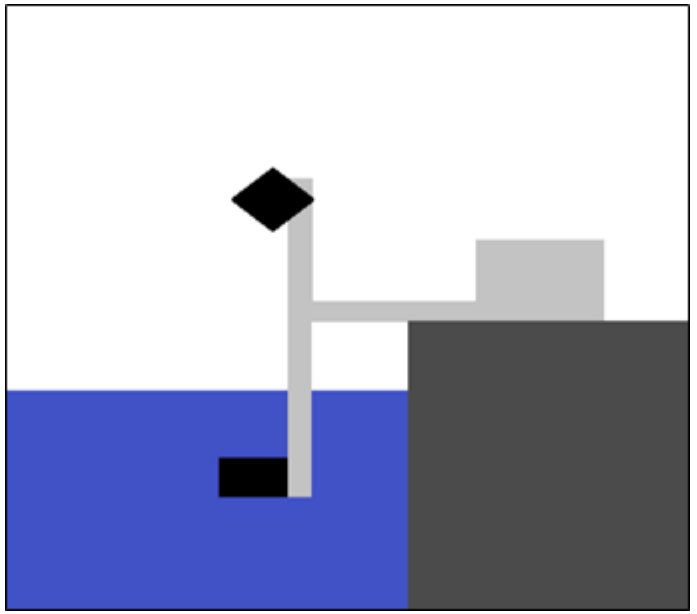
```
curLvl = Midpoint + (Xbox.getAnalogHat(RightHatX, i)/analogStickMax)*bandWidth;// move porpotional to position of analog stick
```

Hardware Integration

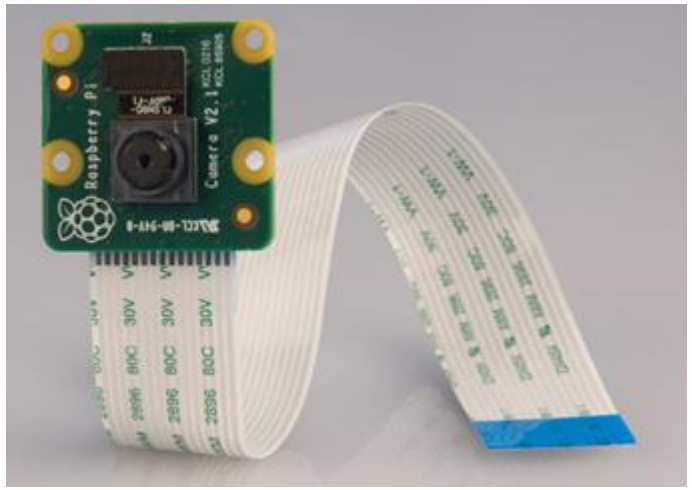
- Power (Raspberry Pi 3)
  - 5V supply
  - 2.5A current draw
    - 5V Power Supply connected to Xbox receiver.
- Slippage
  - Athletic Tape around wheel

Hardware Integration (Raspberry Picamera)

- Mounting
  - Above water
  - PVC pipe onto previous camera mount
  - 45 degree angle towards water

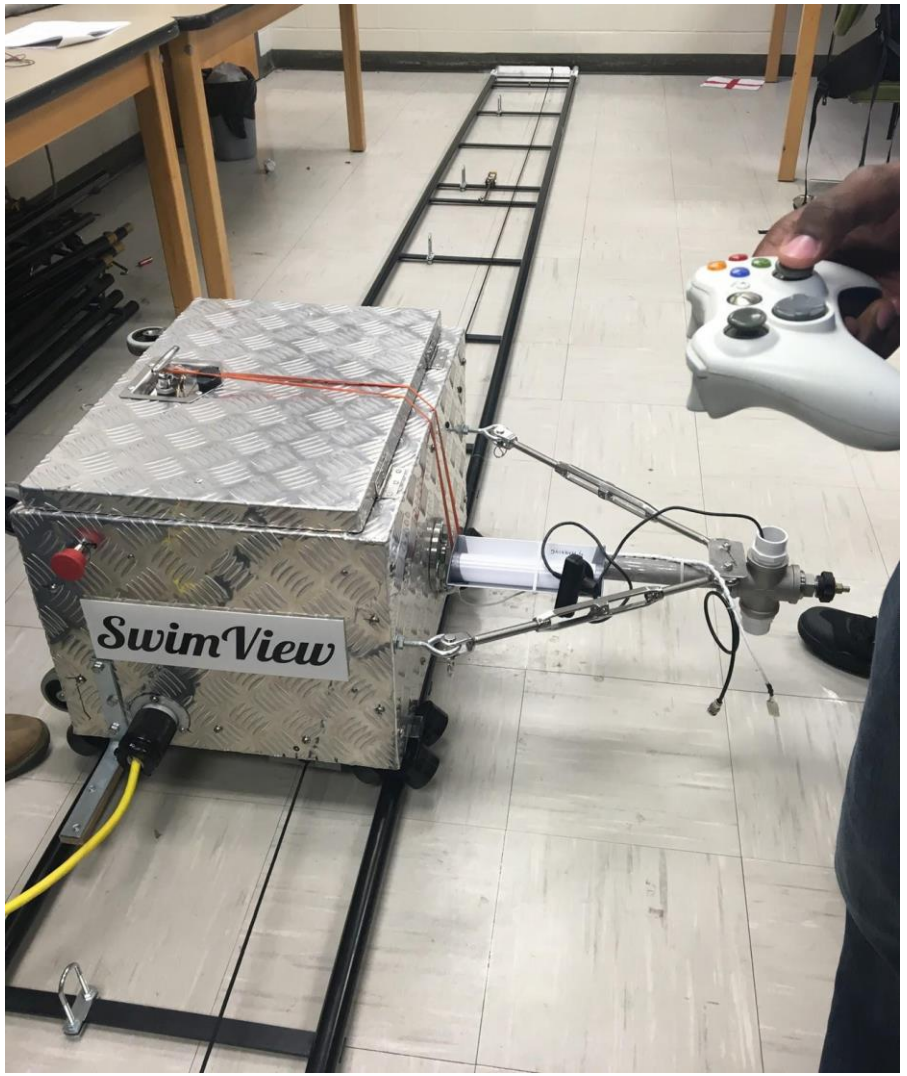


- Electrical Connection
  - 2m Ribbon cable to Pi 3



Complete System Testing

- Software/Hardware
  - Interaction between the tracking algorithm and control system
  - Test indoor with face and color detection
- Functionality
  - Cart speed/direction
  - Stability of pulley system



Complete System Testing (on-site)

- Usability
  - Manual controller to autonomous control mode
  - Camera angle
  - Video recording and live-stream features

Updated Budget

Items Already Purchased	
Item	Price
Raspberry PiCamera	$29.99
Picamera Case	$8.49
Picamera Cable	$11.96
USB-to-Micro USB Cable	$6.95
	$57.39

Remaining Items	
Rubber	$10.00
PVC Pipe	$50.00
Raspberry Pi Camera Lens	$30.00
Colored Athletic Tape	$20.00
Overhead	$58.54
	$168.54

Updated Timeline

Table 1: Updated Timeline

Phase	Week No.	Wednesday Date	Objective
Phase Two	10	22-Mar	Tracking Algorithm Development
	11	29-Mar	Controls Testing
	12	5-Apr	Hardware Integration
	13	12-Apr	System Testing
	14	19-Apr	System Testing
	15	26-Apr	Presentation Development
	16	28-Apr	Senior Design Fair

Conclusion

- On track
- Getting ready to test controls
- Tweak the code