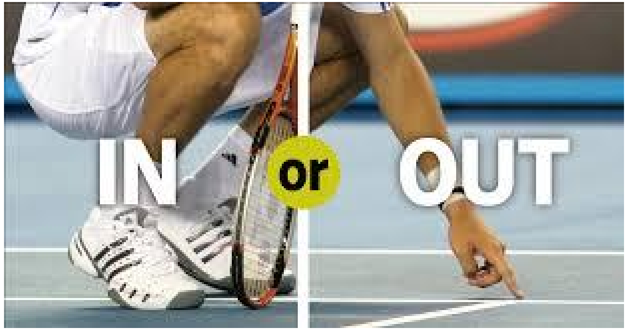
# Automated Tennis - Image Processing and Launcher

Group 14
Michael Rathbun
Aviel Yashar
Khoa Hoang
Kyle Willnow

# Motivation

- Make tennis as convenient as bowling
  - Appeal to the lazy and luxurious
  - Increase play time


- Remove human error
  - NCAA tennis
  - High school tennis
  - Recreational tennis




# Goals and Objectives

- Provide ball retrieval
- Increase play time
- Supply tennis balls to players upon request
- Provide unbiased scoring and line calls
- Achieve maximum accuracy for camera system despite budget constraints

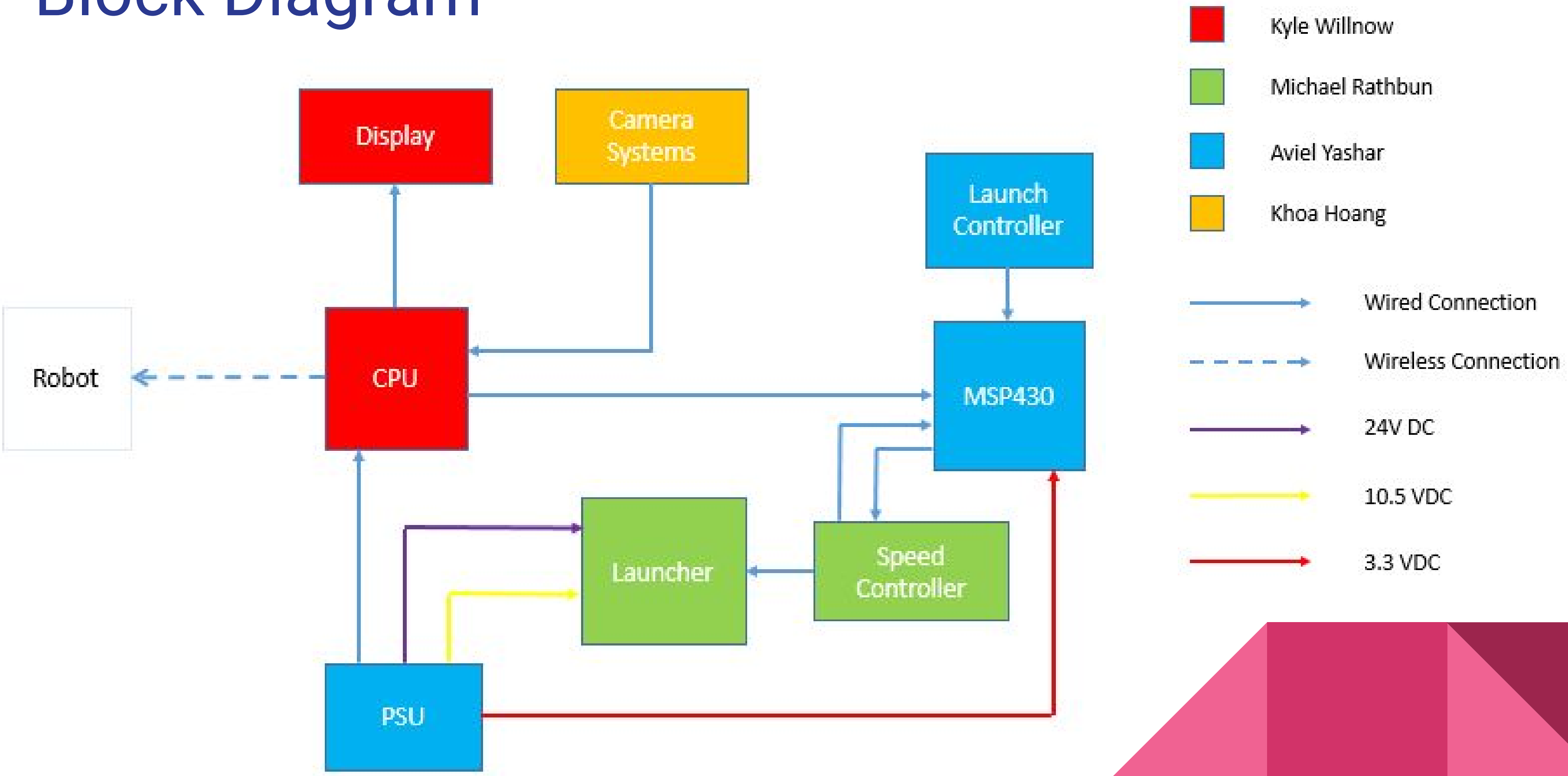
# Specifications

- Camera System
  - Locate ball in 2D in more than 80% of the frames that contain a ball
  - Determine 3D ball location in more than 50% of the frames examined
  - Perform a line call correctly in more than 70% of cases
  - Command launcher correctly 80% of cases

# Specifications

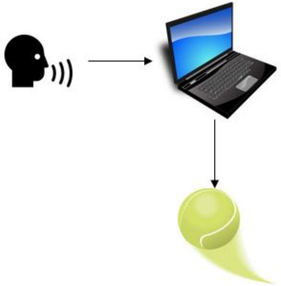
- Tennis Ball Launcher
  - Launch ball in direction of player (∓15 degrees)
  - Consistency in landing (within 1 foot)
  - Initiate launch upon request (within 15 seconds)
  - Safety feature: ball should not launch if player is within 6 feet of the front
  - Store up to three balls

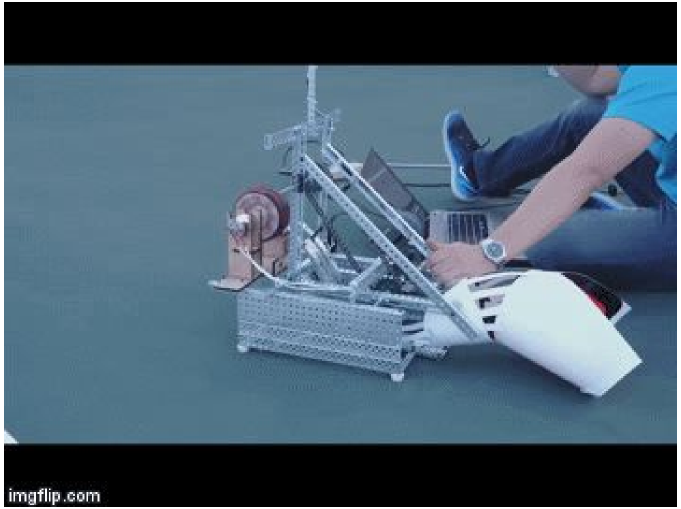
# Block Diagram



# Launcher

- Houses PSU and controller PCB
- Loads ball into belt, waits to be fired
- Communicates with CPU through voice command to launch ball
- Structurally sound to take vibration and avoid damage if hit by tennis ball





# Power Supply

# MCU Matrix

| MCU | Non-Volatile Memory (KB) | RAM (KB) | GPIO Pins | UART | $V_{in}$ (V) | $V_{CC}$ (V) | Price |
|---|---|---|---|---|---|---|---|
| **ATmega328P** | 32 | 2 | 14 | Yes | 7-12 | 5 | $3.70 |
| **MSP430G2553** | 16 | 24 | 16 | Yes | 1.8-3.6 | 3.3 | $1.00 |
| **MSP430FR2111** | 4 | 1 | 12 | Yes | 1.8-3.6 | 3.3 | $0.66 |

# Chosen MCU - MSP430G2553

- UART communication
  - Voice command
- Contains needed amount of I/O pins
  - Supports integrated functions & future functions that may be desired
- Inexpensive
- Familiar with integration
- Adjusts launch distance and controls ball loading and firing

# Speed Controller

- Implements load control
  - As ball is taken from retriever, belt servo is turned on
  - Once it is “loaded”,  belt servo is turned off
- Changes launching distance
  - Voice command chooses high or low launch distance
  - Input change causes duty cycle of output PWM to change, thus adjusting speed of launch motor

# Pressure Switch Matrix

| Product | Price | Vcc | Range of Effectiveness (g) |
|---|---|---|---|
| **Limit Switch** | Free | Any | >11 |
| **Resistive Pad** | $7-$8 | 5V | 50-1000 |

# Limit Switch

- Tennis ball has mass of about 50 g
- Initiates signal to stop ball before firing
- Ball holds this position until player initializes ball return
- Once initialized, limit switch is overridden and ball moves forward

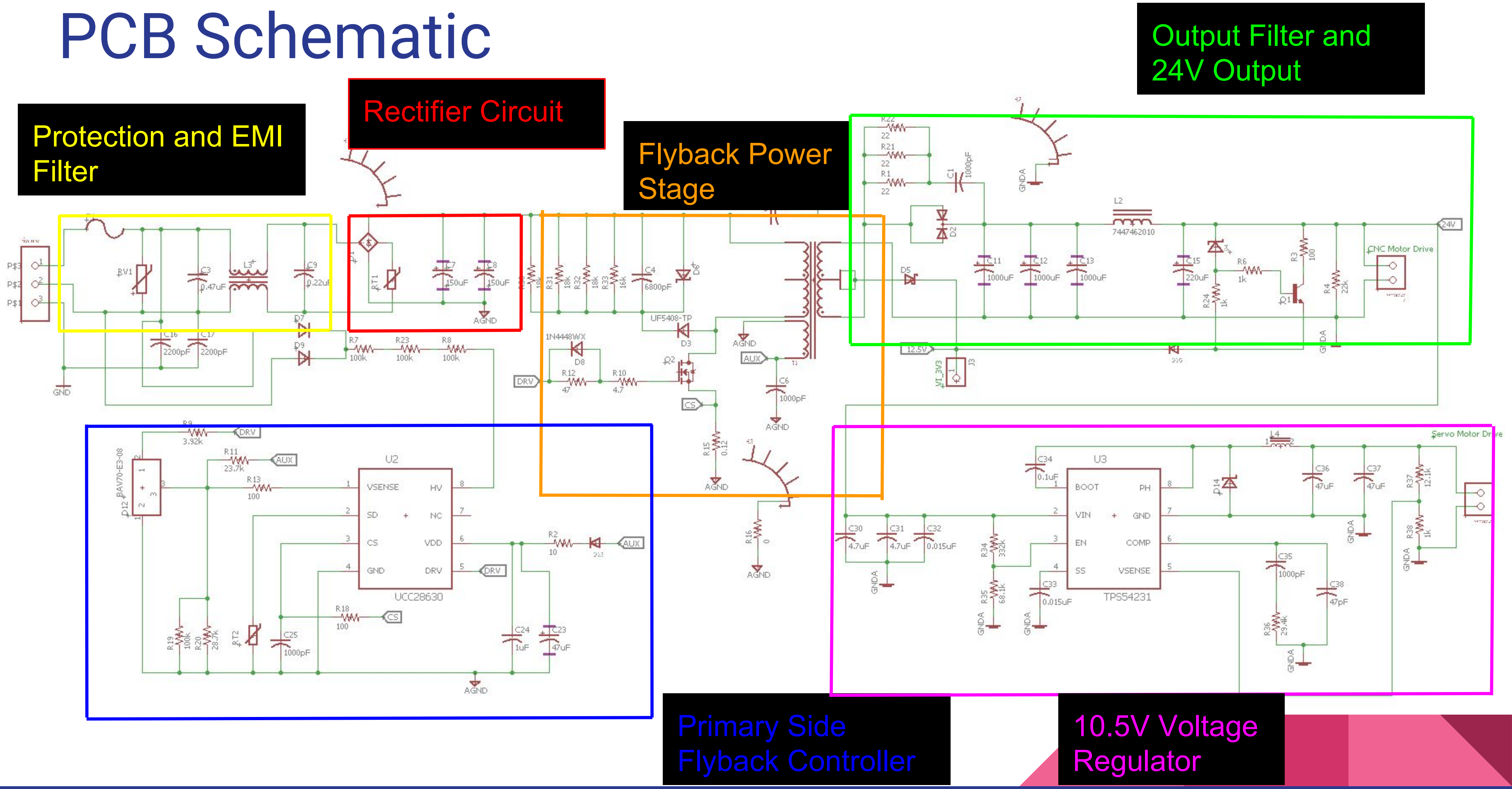
# Ultrasonic Sensor Matrix

| Product | Supply Voltage | Communication | Cost | Sense Distance |
|---|---|---|---|---|
| Vex Ultrasonic | 5 V | N/A | Free ($30) | 3 cm-3 m |
| Optical Distance (MR) | 5 V | N/A | $26.95 | 0.5 cm-8 m |
| Ultrasonic + Optical (MR) | 5 V | I2C | $44.95 | 1 cm-2.55 m |
| HC-SR04 Ultrasonic | 5 V | TTL | $2.50 | 2 cm-4 m |

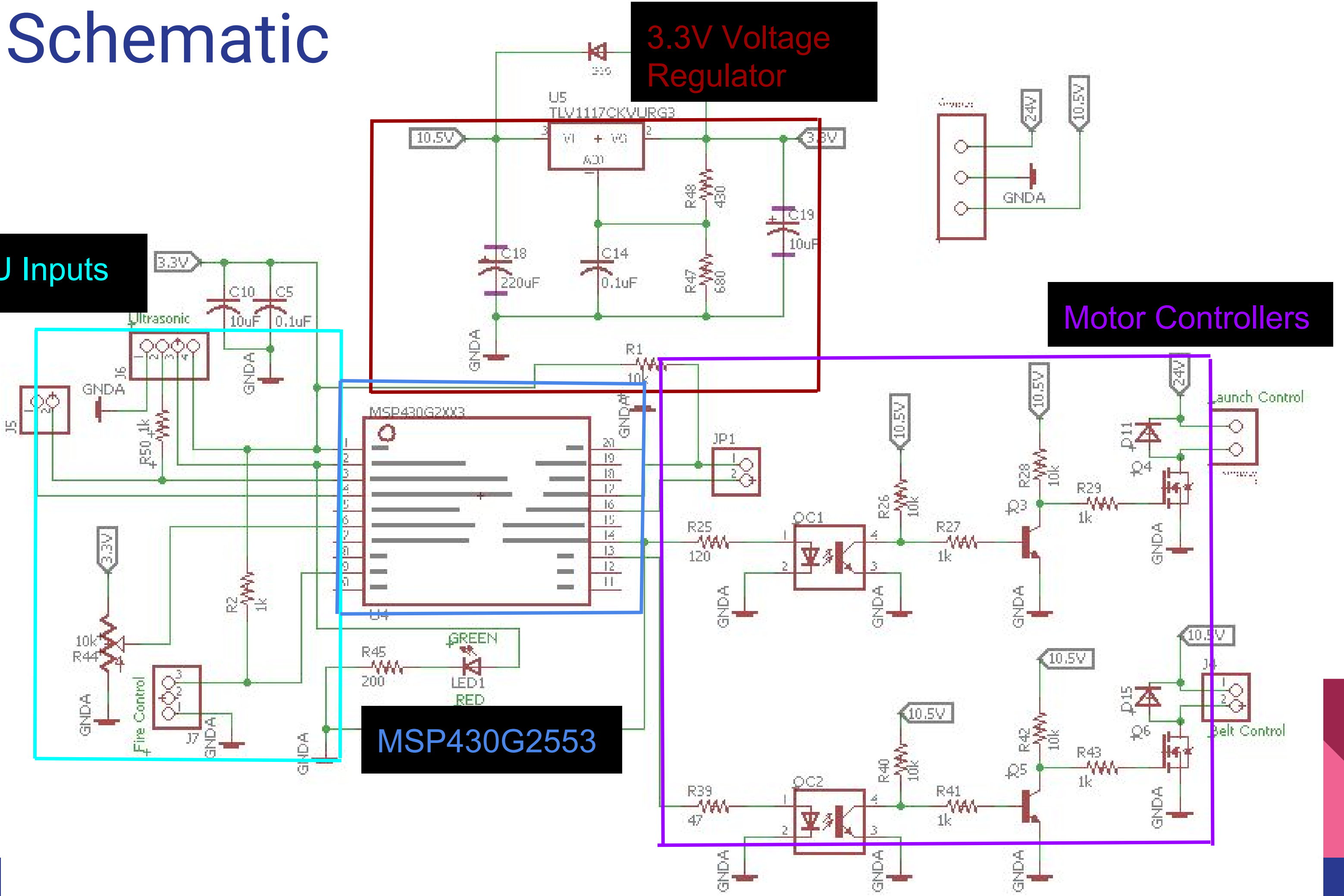
# Ultrasonic Sensor

- Safety feature preventing harm to passerby
- HC-SR04 provides needed specifications at low cost
- Threshold for object detection
  - Withholds servo from moving to launch ball

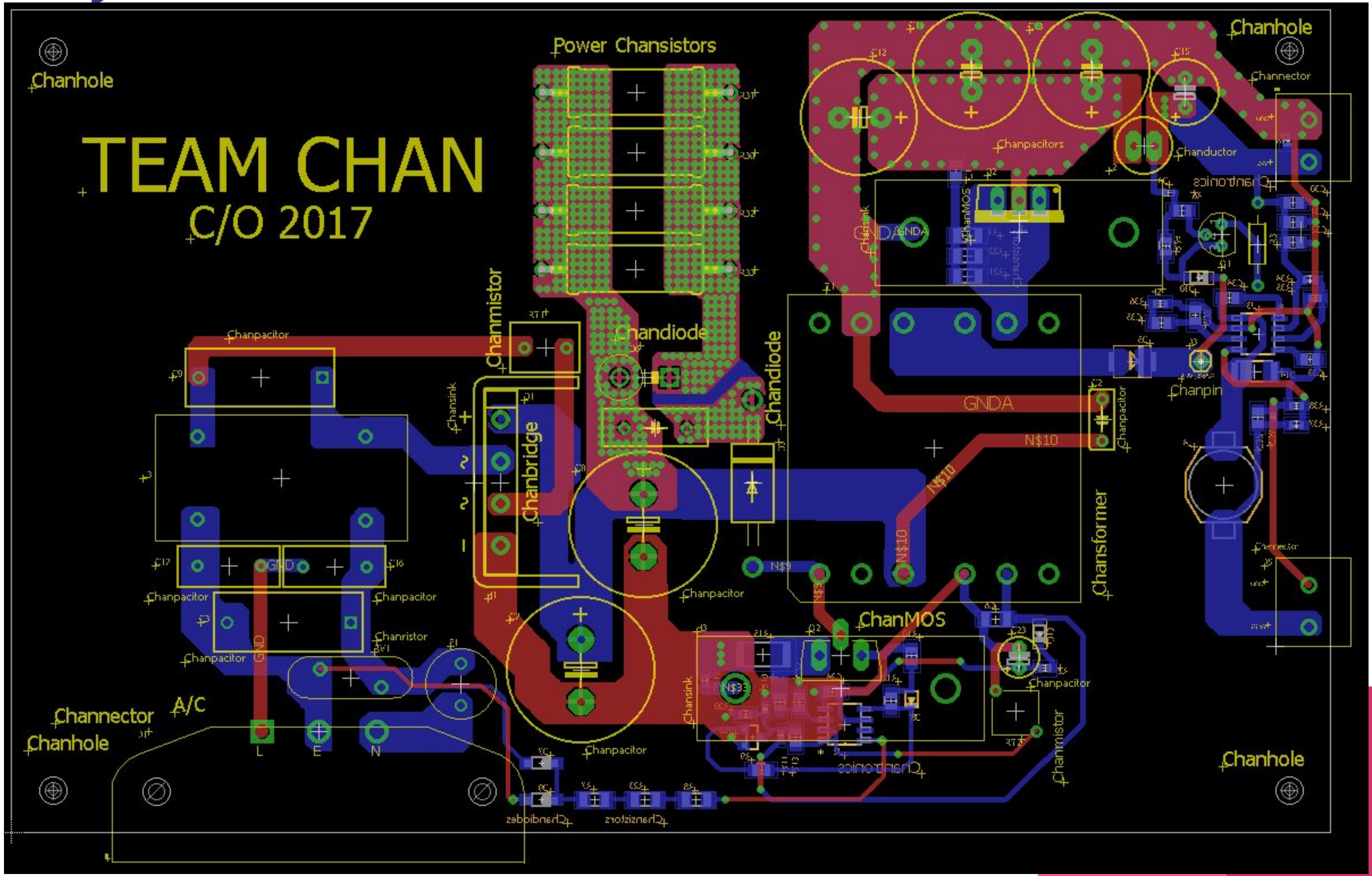
# PCB Schematic



# PCB Schematic

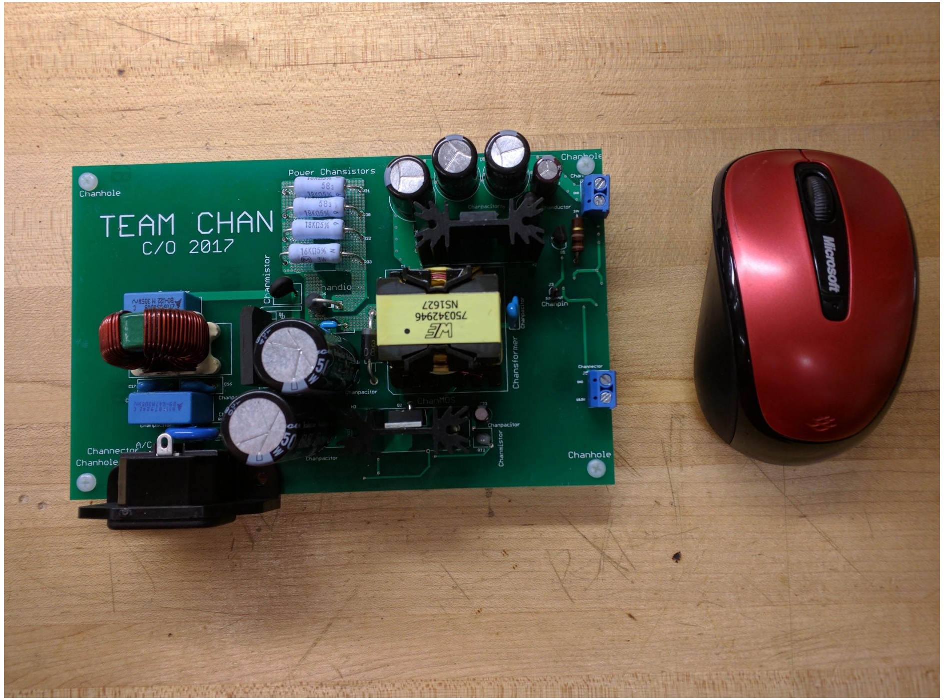


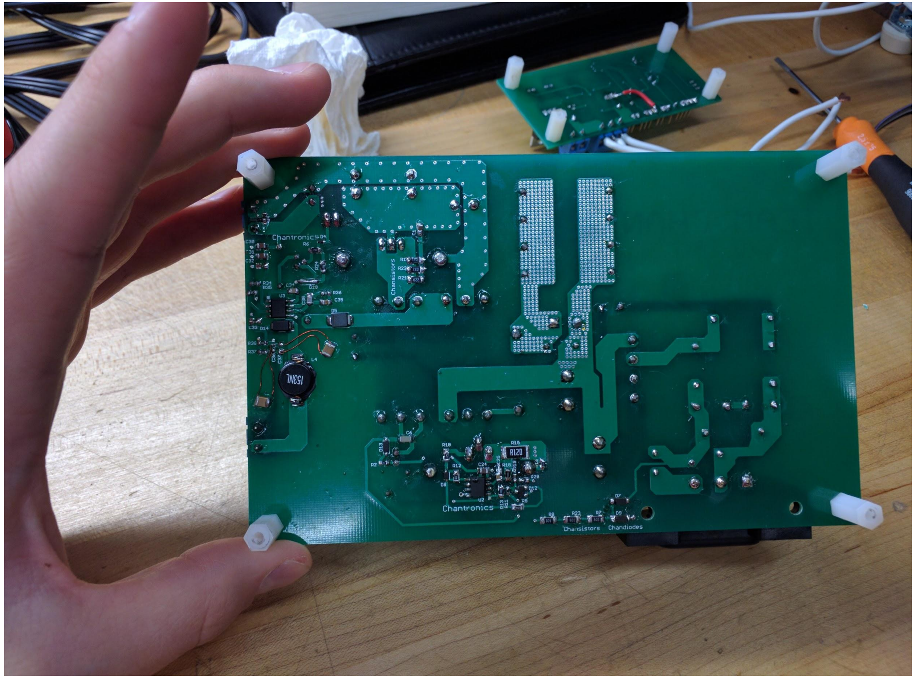
# PCB Layout



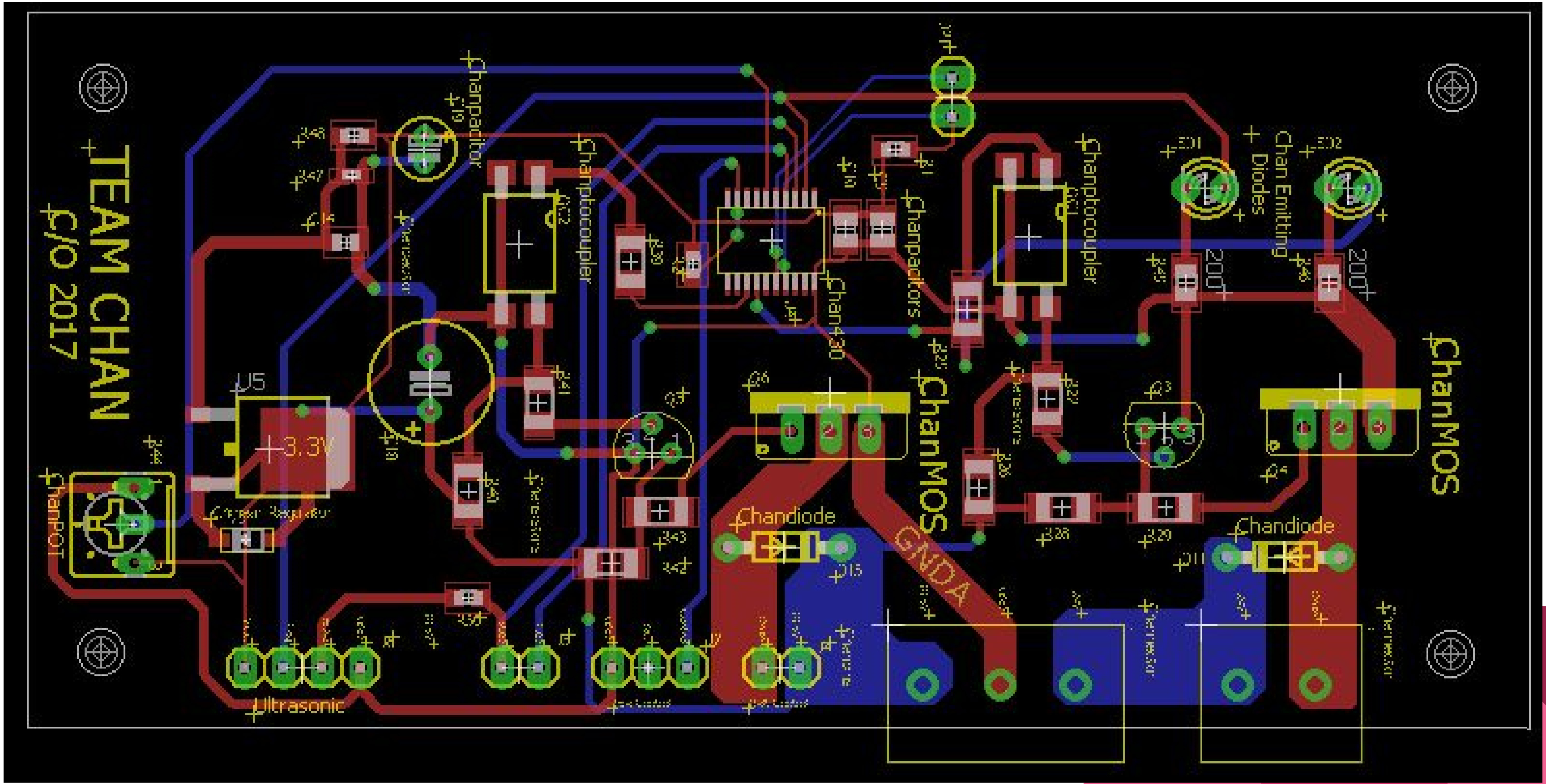
TEAM CHAN
C/O 2017
Power Chansistors
Chanhole
Chanpacitors
Chanductor
Channector
Chandiode
Chanmistor
Chanbridge
Chansink
Chanpacitor
Chanpin
GNDA
N$10
Chansformer
ChanMOS
Chanristor
Channector
A/C
L
E
N
GND
Chanhole

# PCB Layout

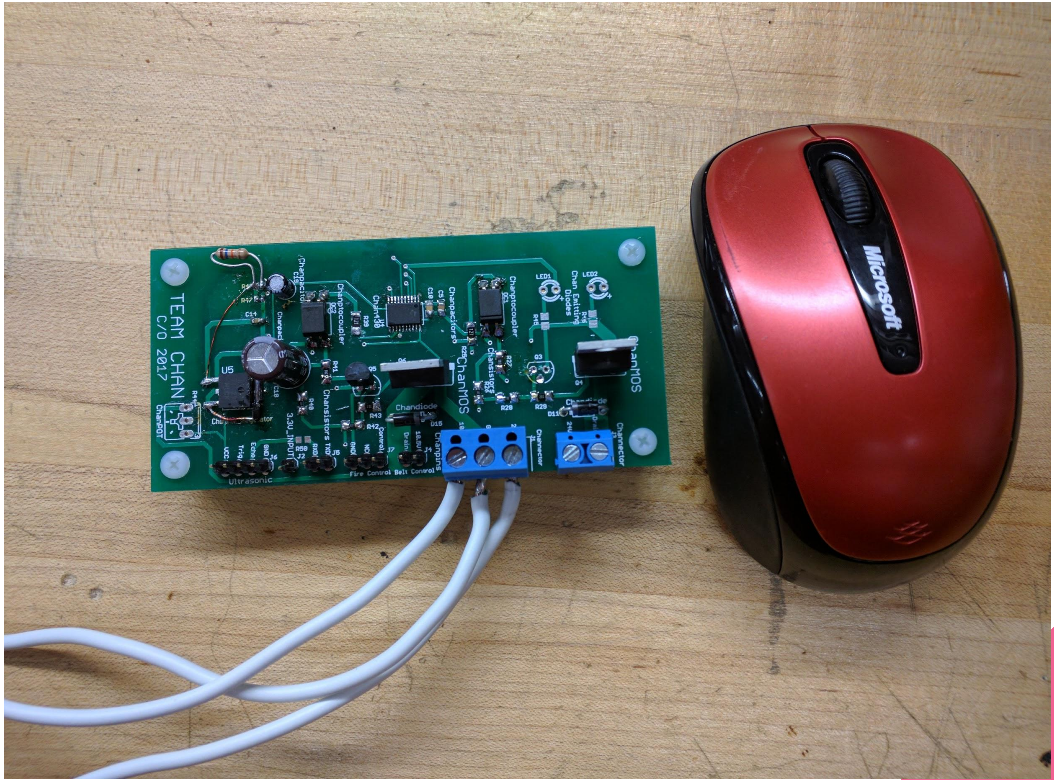

TEAM CHAN
C/O 2017
Power Chansistors
Chanhole
Chanhole
Chanhole
Chanhole
Channector A/C
Chansformer
750342946
NS1627
Microsoft


Chantronics
Chantronics
Chansistors
Chandiodes

# PCB Layout


TEAM CHAN
C/O 2017
U5
3.3V
ChanPOT
Ultrasonic
Chanpacitor
Chanptocoupler
Chan430
Chanpacitors
Chan Emitting Diodes
ChanMOS
ChanMOS
Chandiode
Chandiode
GNDA

# PCB Layout



# Problems

- Structural integrity and functionality
  - Vibrations are induced by unbalanced wheel
  - Two wheels are needed to increase launching capacity
- Large power supply is needed
  - Launch motor requires large amperage, mostly caused from instabilities during ramping
  - two boards needed to separate large voltage from small voltage
- Noisy voltage rails
  - 200 mV noise causes intermittent MCU functionality and zero ultrasonic sensor functionality
- Signals sent from ultrasonic sensor may not bounce back directly to receiver
  - Results in false negative

# Solutions

- Structural
  - Wheel mount laser cut to reduce vibrations
  - Carpet needed to prevent device from rotating as vibration occurs
- Power supply
  - Motor limited to ~90% as not to activate overcurrent protection
- Clean power supply is used
  - MSP430 launchpad for ultrasonic sensor
  - 10.5V rail is regulated to 3.3V instead of center tap from transformer
- Ultrasonic sensor signal reflection
  - Sensor elevated to increase chance of orthogonal reflection

# Voice Recognition

- Runs as separate lightweight process in Python alongside image processing software
  - Uses PocketSphinx and PySerial
- Respond to voice commands from player to trigger launcher to launch ball
  - 2 levels - “far”, “short”
  - Can be susceptible to external noise
- Use serial port to communicate with launcher via UART
  - Send one character to set speed, then another to launch

# Image Capture

- Need to capture live video footage of tennis match for tennis ball tracking
- Singles court size: 27 feet x 39 feet
  - Need multiple views to capture such a large area, otherwise examine only one section
- Tennis serve upwards of 55 m/s for professionals
  - Need cameras with high FPS to achieve clear, non blurred footage
  - Even at 60 FPS, a 50 m/s serve travels 0.8 meters in only one frame
- Need 3D location of tennis ball relative to camera
  - Use pairs of cameras for stereoscopic vision

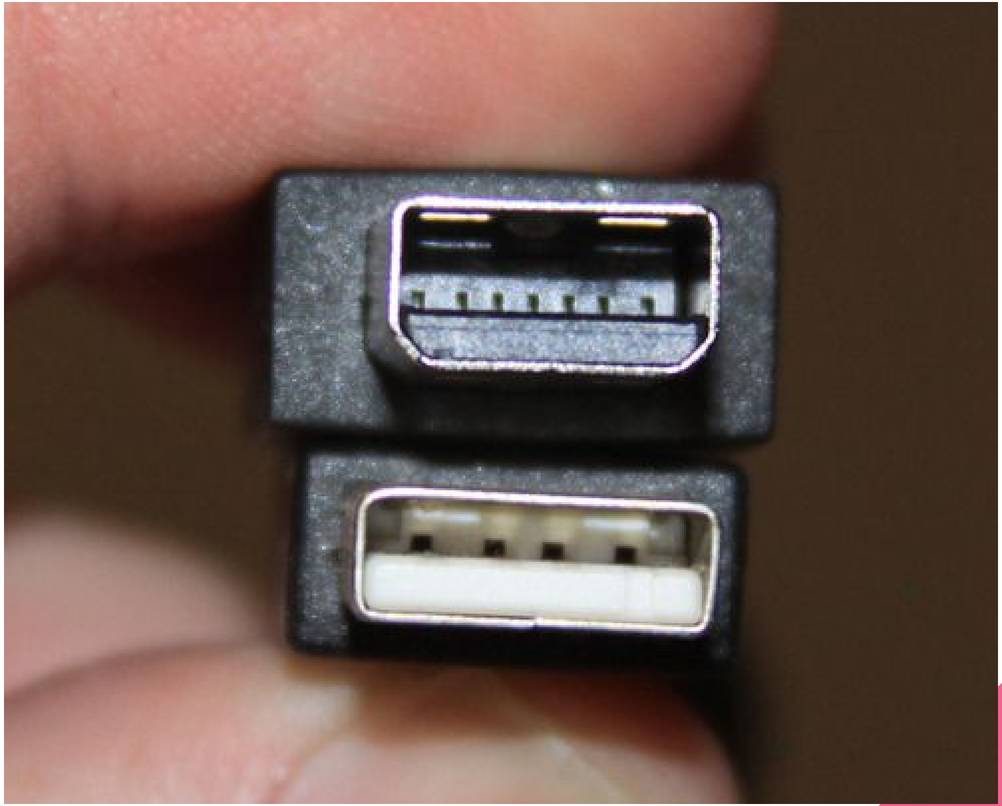
# Camera

| Camera | Playstation 4 Eye | Playstation 3 Eye |
|---|---|---|
| **Resolution @ FPS** | 1280x800@60<br>640x400@120<br>320x192@240 | 640x480@60<br>320x240@120 |
| **Cost** | $40.99 | $4.98 |
| **Notes** | Stereoscopic, High FPS, Synchronized | Cheap, High FPS |

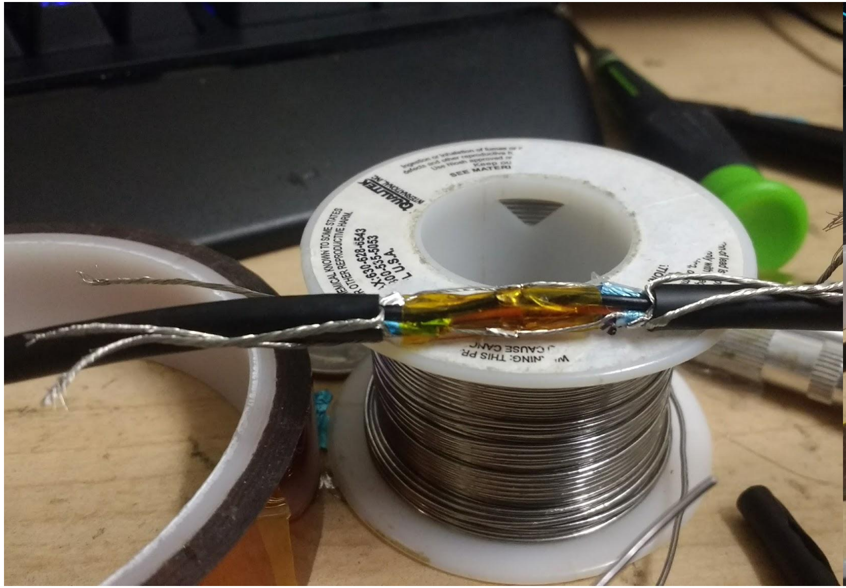
# Camera Selection

- Chose Playstation 4 Eye
  - Already stereoscopic, and synchronized internally
  - High FPS per dollar
  - Unfortunately not directly PC compatible

# Playstation 4 Eye - USB3

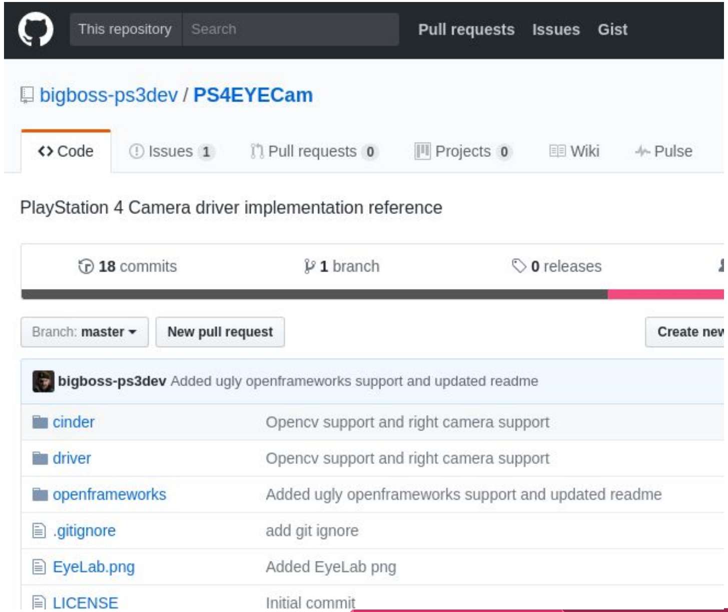


# Playstation 4 Eye - USB3

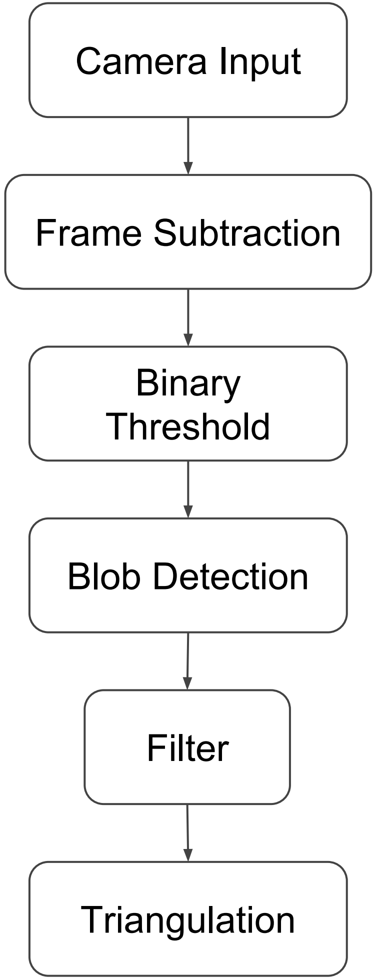


# Playstation 4 Eye - Driver

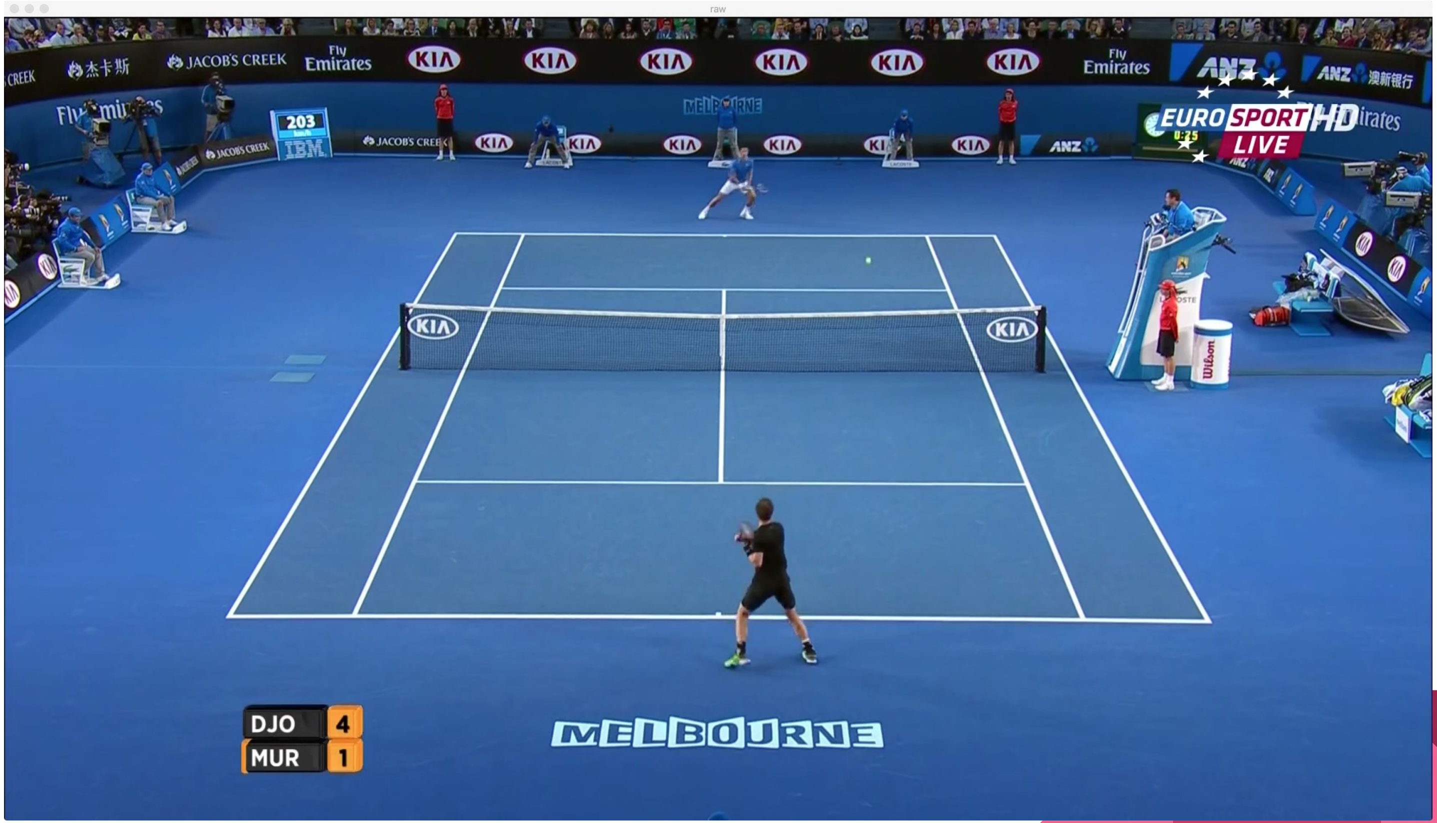
- The PS4 Eye had been reversed and they already wrote a driver for the camera.
- Problem: It only support OSX Maverick
- We able to port it to Linux and integrated into our program.



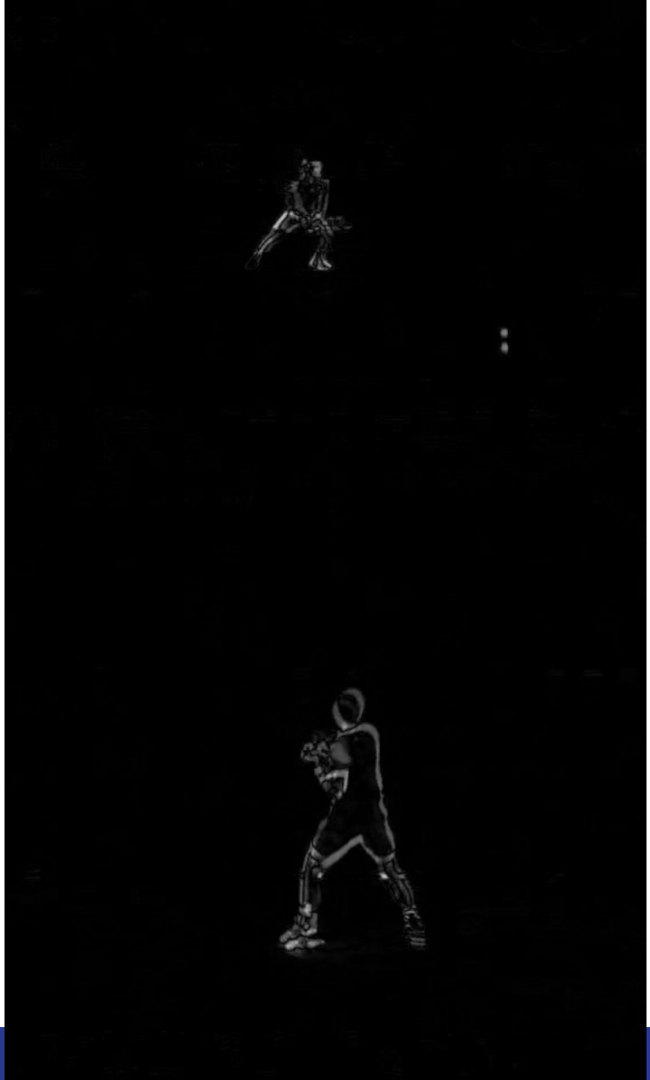
# System


Camera Input
Frame Subtraction
Binary Threshold
Blob Detection
Filter
Triangulation

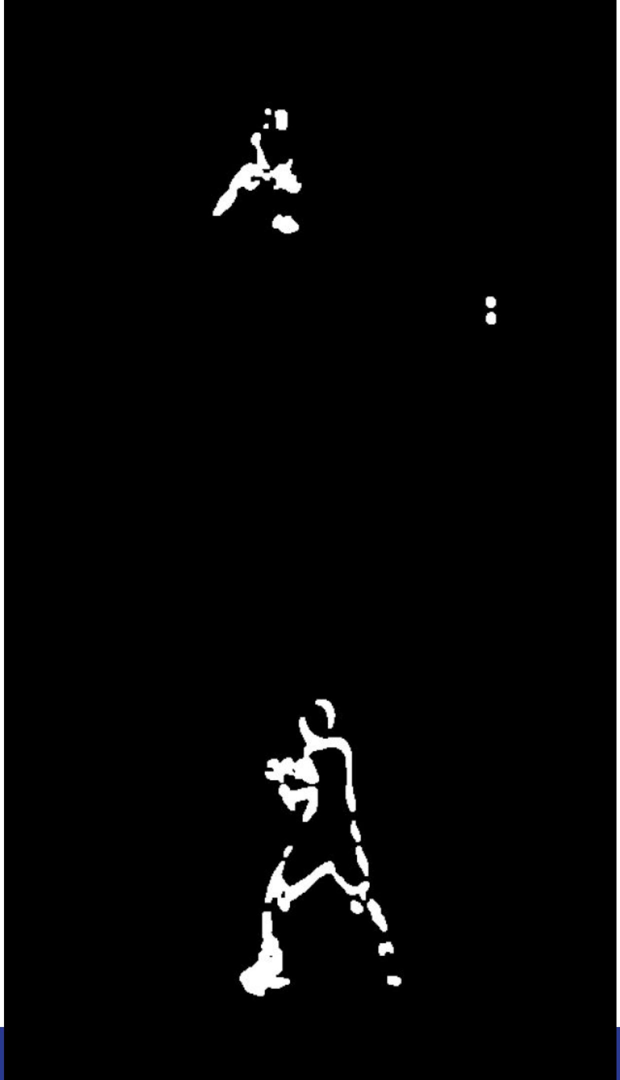
# Input


raw
KIA
JACOB'S CREEK
Fly Emirates
203
IBM
EUROSPORT HD
LIVE
DJO 4
MUR 1
MELBOURNE

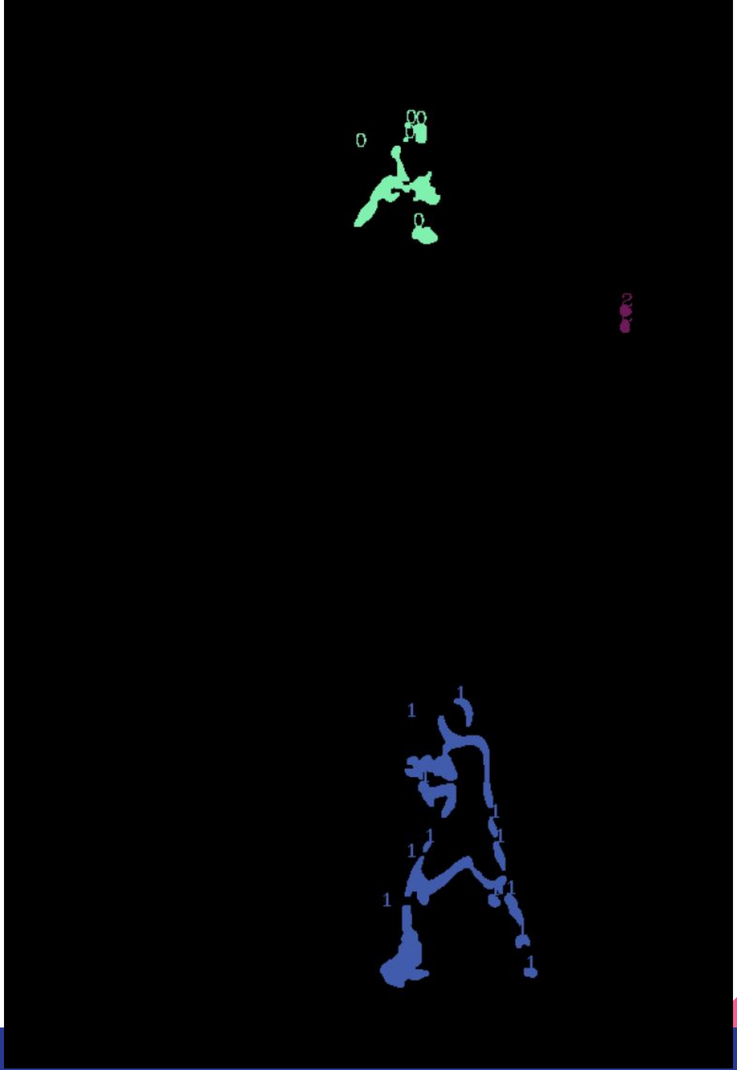
# Subtraction



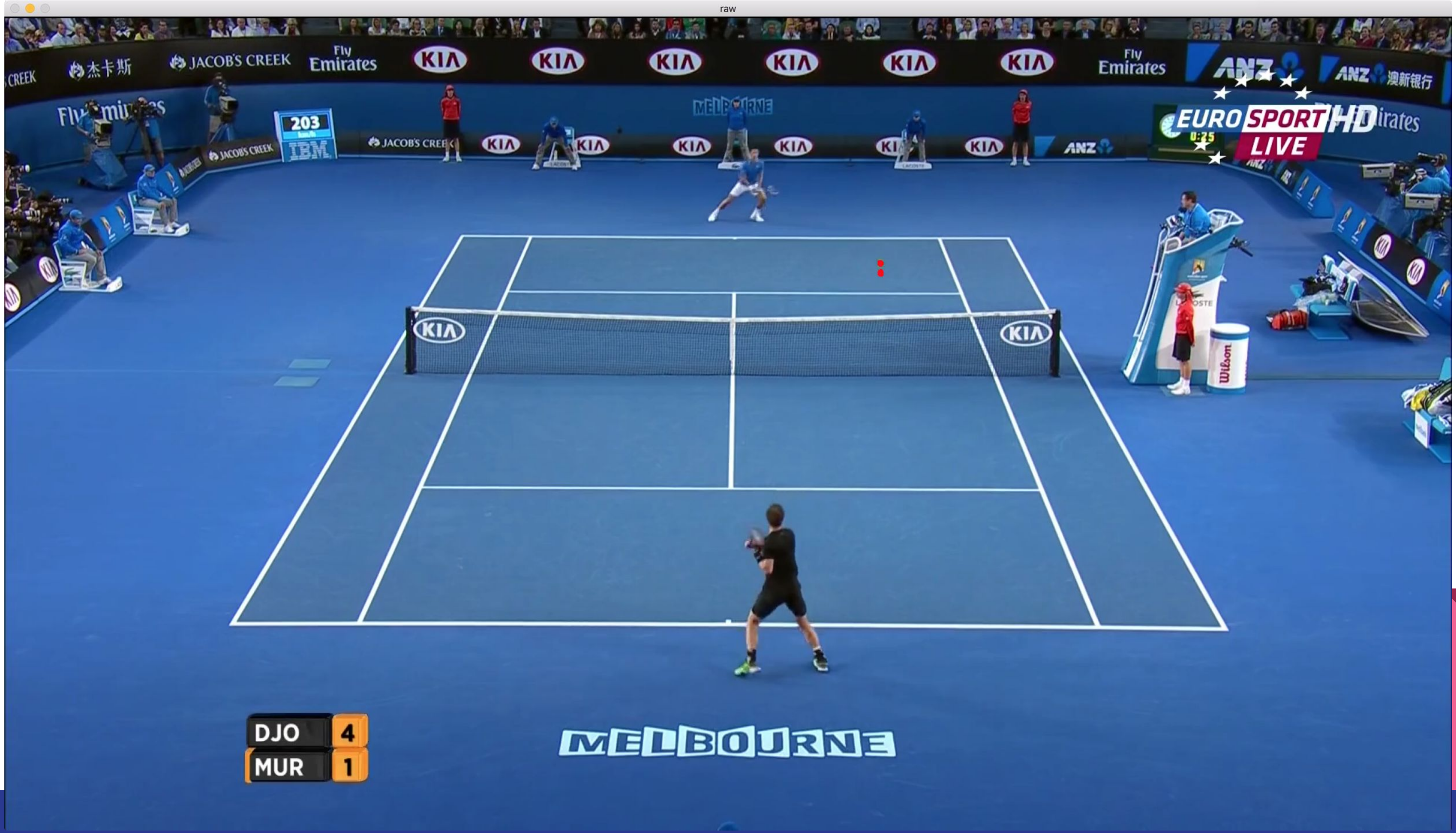
# Threshold



# Blob Detection



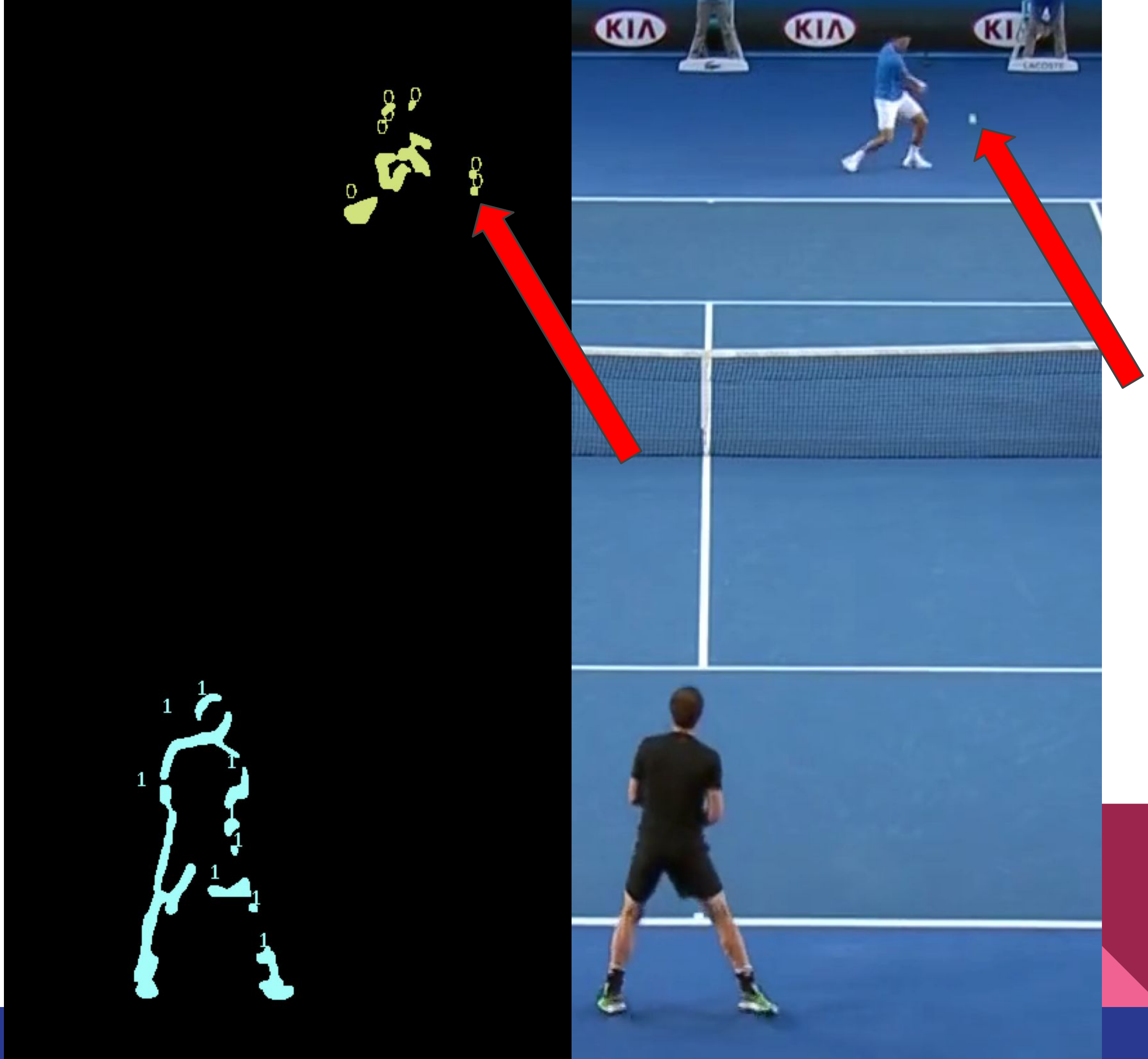
# Filtering


raw
JACOB'S CREEK
Fly Emirates
KIA
ANZ
203
IBM
EUROSPORT HD
LIVE
DJO 4
MUR 1
MELBOURNE

# Filtering

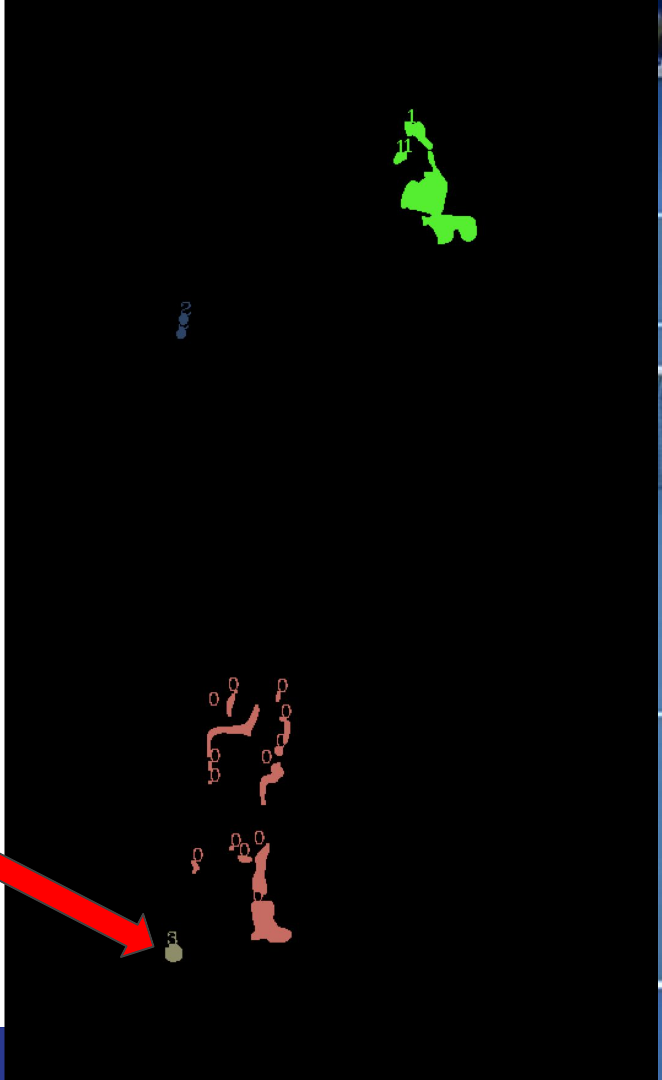
# Filtering Problems

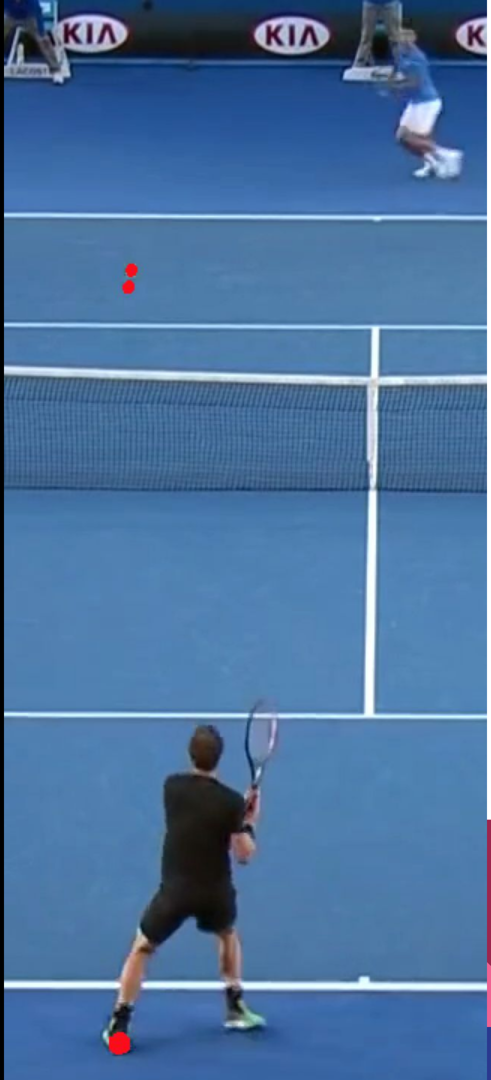
- False negatives



# Filtering Problems

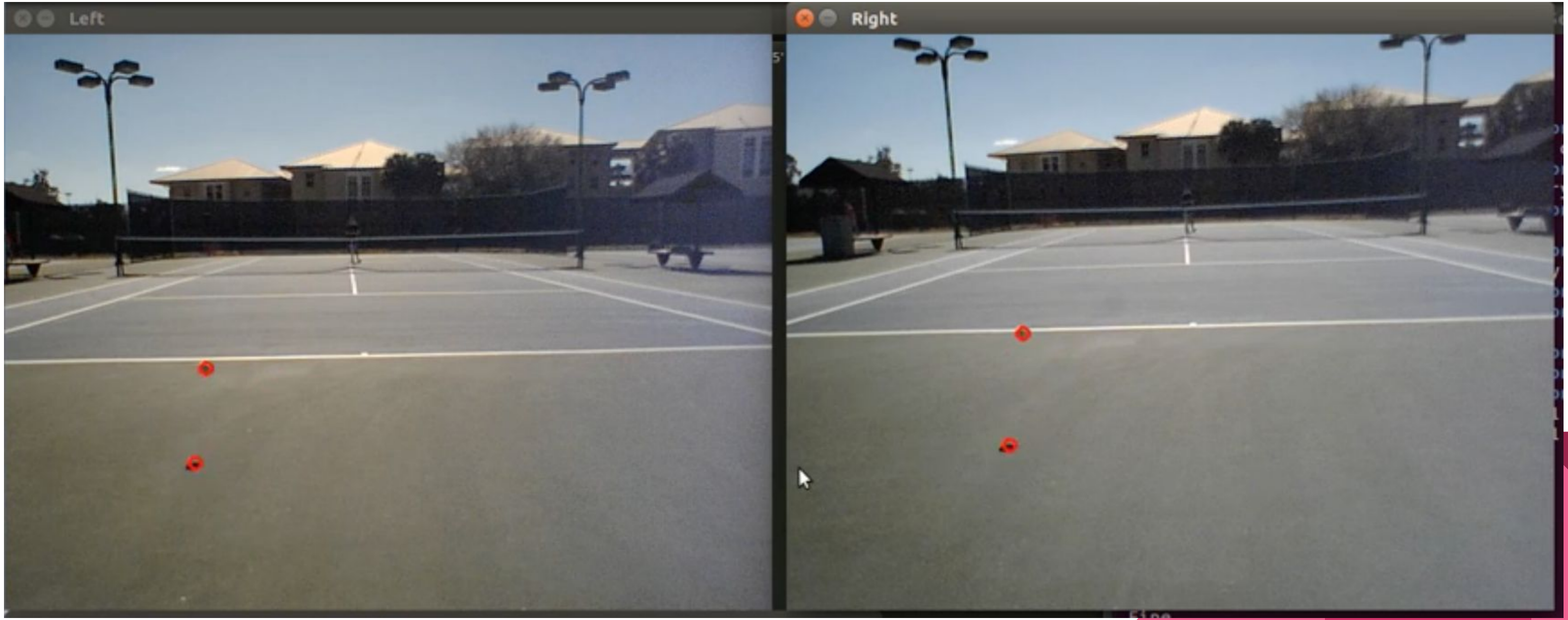
- False positives





# Filtering Problems

- Shadow - Resolution: Choose highest point in image

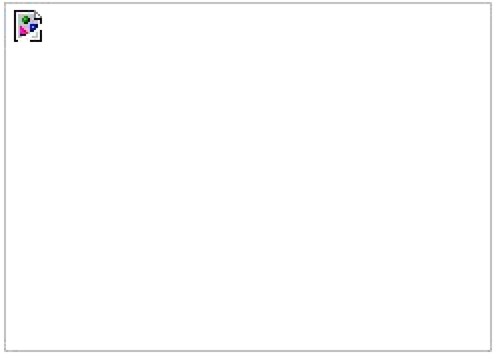


# Camera Setup

- Every court and camera are different, so some initialization must be done
  - Camera Calibration - Find distortion parameters of individual cameras, then (because of stereoscopic vision) find location of one camera w.r.t. the other
  - Measurement - Get physical distances of camera setup to the court

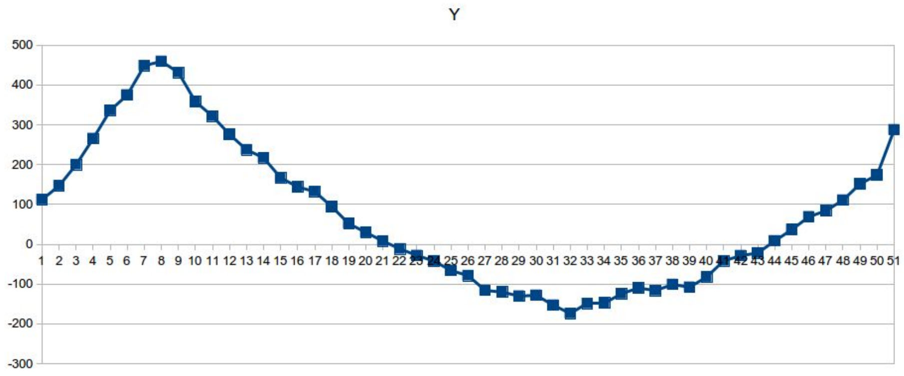
# Triangulation

- From location and filtering stage, receive pairs of left/right points representing ball locations
- Use OpenCV API for triangulation, get 3D point relative to camera
  - 3D point must be made relative to court
- Compare 3D point to court measurements to determine a court location (service boxes, in / out)



# Bounce Detection

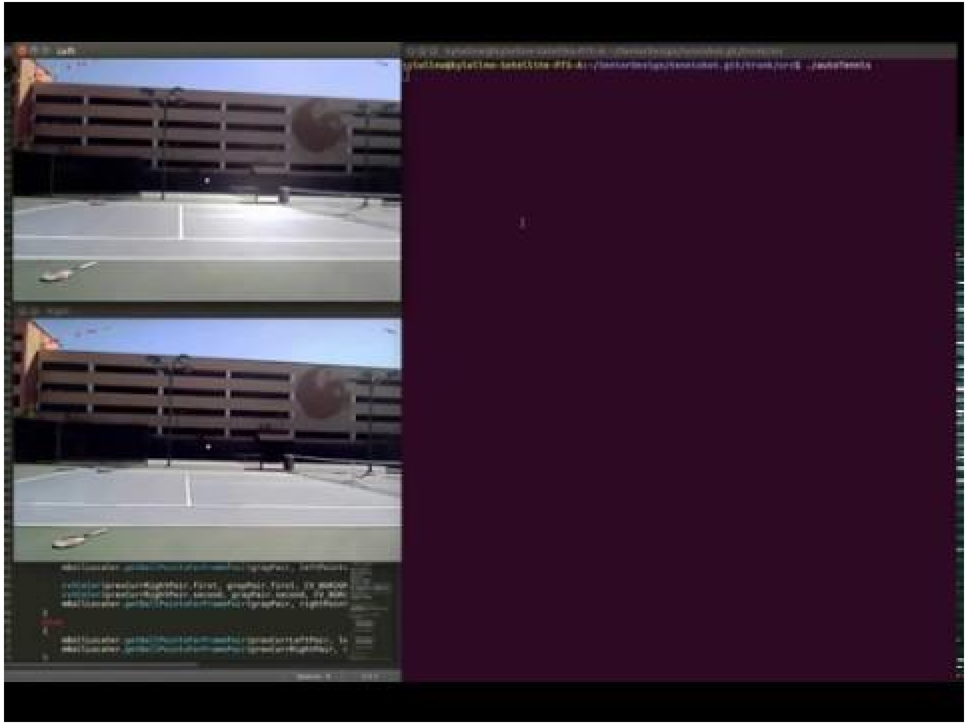
- Y coordinate of ball indicates altitude, greater is lower altitude
- Get point of interest and compare neighboring values
  - If any values larger than point, the point is not a bounce
- Can refine after detection (i.e. “bounce” above or behind the camera)



# Decision Making - Robot Interface

- Ball Retrieval Robot has only local vision, needs assistance from camera system for ball location
  - Guide robot to general ball location, robot can locate ball from that point
- Subdivide court into numbered grid
  - Numbering system agreed upon between robot and camera system
- At the end of the rally, find ball and send location to robot using WiFi and UDP

# Image Processing System



# Administrative Content

# Work Distribution

- Aviel - Electronics and Launcher Design
  - Design and construction of launcher functions (launching, loading, stabilization)
  - PCB building and testing
- Michael - Electronics and Control
  - PCB design
  - Control circuitry and software implementation and testing
- Khoa - Image Processing
  - Raw image processing (ball detection, filtering)
  - PS4 Eye driver port
  - Voice detection
- Kyle - Image Processing
  - Refined image processing (ball location, triangulation, bounce detection)
  - Retriever robot communication

# Budget

| Component | Unit Cost | Quantity | Total Cost |
|---|---|---|---|
| PSEye | $40.99 | 1 | $40.99 |
| Power Supply Components | $71.78 | 1 | $71.78 |
| Power supply PCB | $40.31 | 1 | $40.31 |
| Speed Controller Components | $22.47 | 1 | $22.47 |
| Speed Controller PCB | $29.94 | 1 | $29.94 |
| | | | **Total: 205.49** |

# Financing

- Don Harper
- Self funding

# Automated Tennis - Questions?