UMass Amherst

Comprehensive Design Review

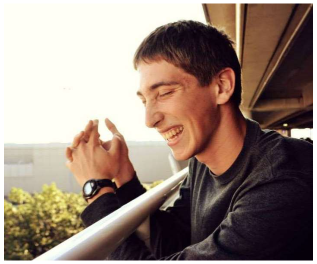
Team 1 (AutoUmp)

March 9th, 2017

Department of Electrical and Computer Engineering

Advisor: Wolf

The Team



Timothy Adams
CSE

Justin Marple
CSE

Jason Camiel
EE

Matt Barnes
EE

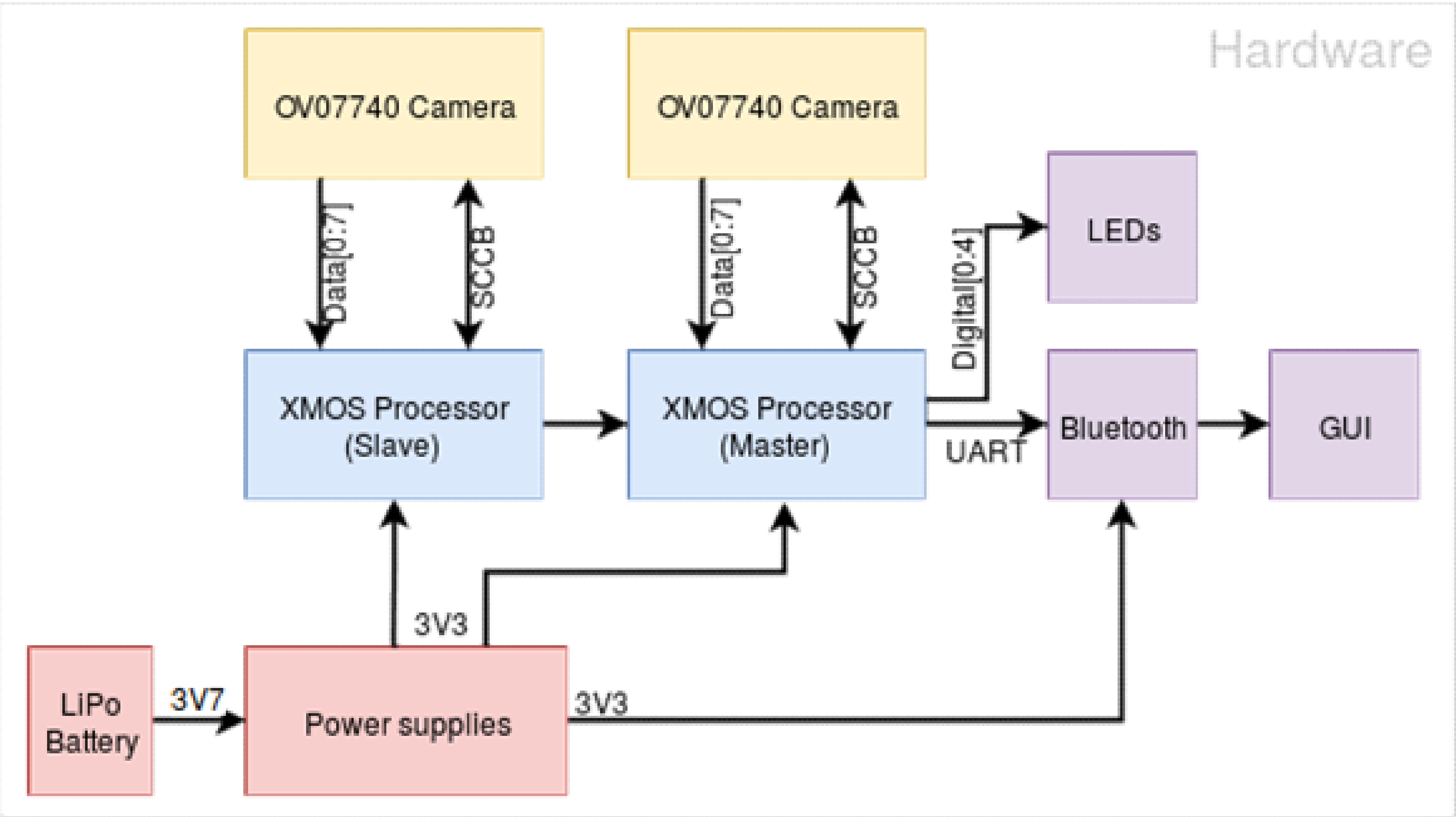
The Problem

- Determining strikes and balls without a professional umpire is difficult and inaccurate
- Current technology solutions are expensive and require extensive set-up
- Incorrectly called pitches lead to angry players

The Solution

- Use a stereo camera and image processing system built into the home plate to determine the baseball's location
- Challenges:
  - Time restrictions for bulk of image processing (16.2ms)
  - High accuracy required for successful use (~95%)
  - Ball detection vs. object detection (e.g. bat)

Block Diagram - Hardware



CDR Deliverables

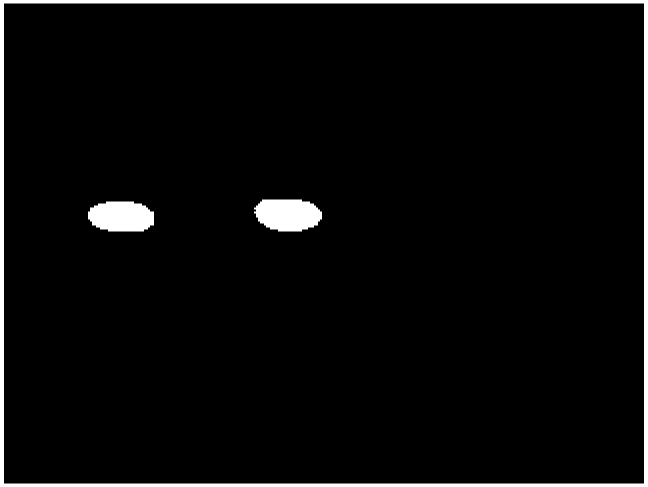
- Demonstrate complete system functionality
  - Implement ball detection, strike zone intersection, and baseball height functions in C (Tim)
  - Functioning PCBs (Justin)
  - Capable of detecting balls/strikes with hardware at 60fps
- Implement full app functionality (Jason)
  - 2-way Bluetooth communication established
    - App receives ball/strike information including XY position of ball
    - Plate receives batter height
- First iteration of enclosure built (Matt)

Background Subtraction and Denoise

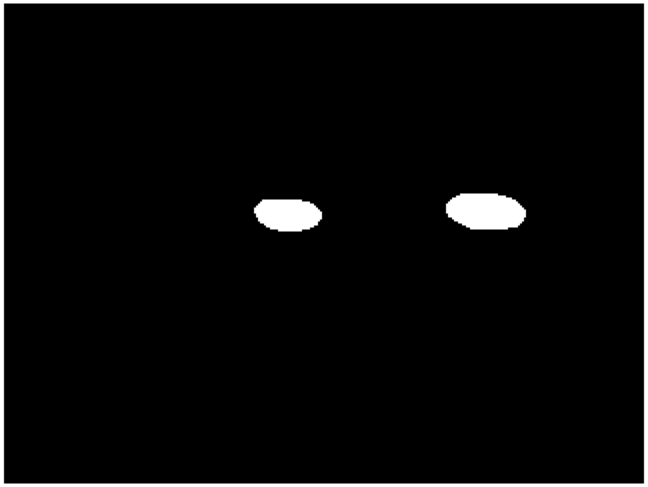
UMassAmherst

Background Subtraction and Denoise

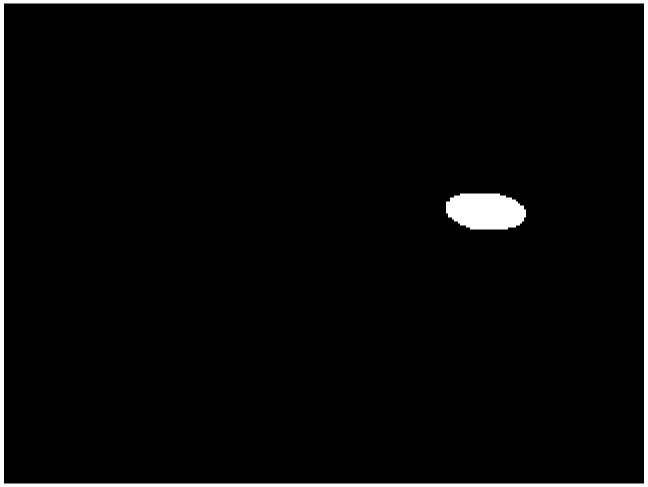
Background Subtraction and Denoise



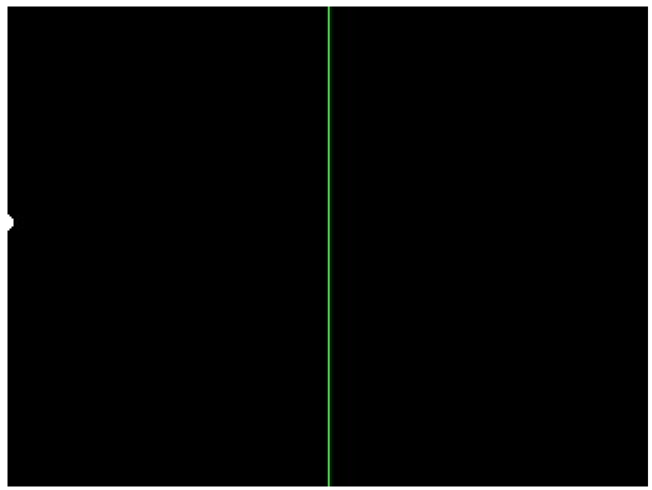
Background Subtraction and Denoise



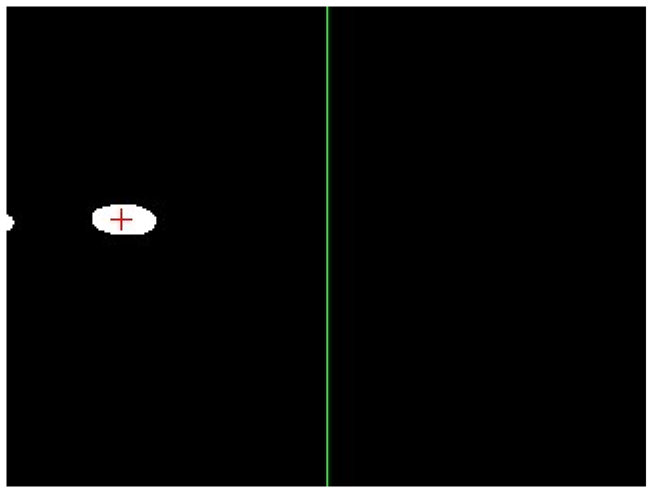
Background Subtraction and Denoise



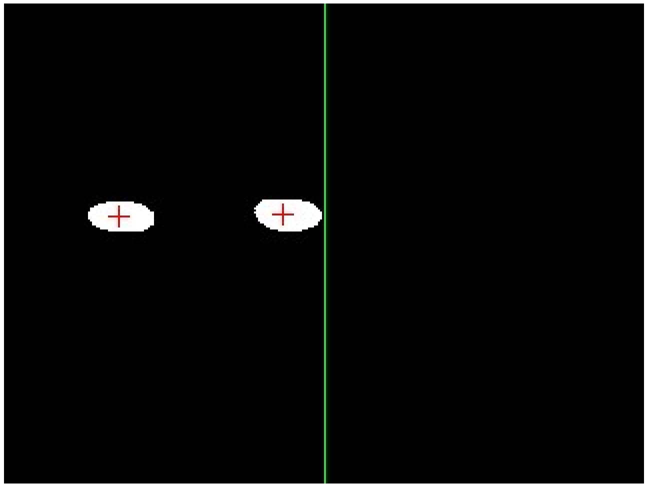
Object Detection and Find Intersection



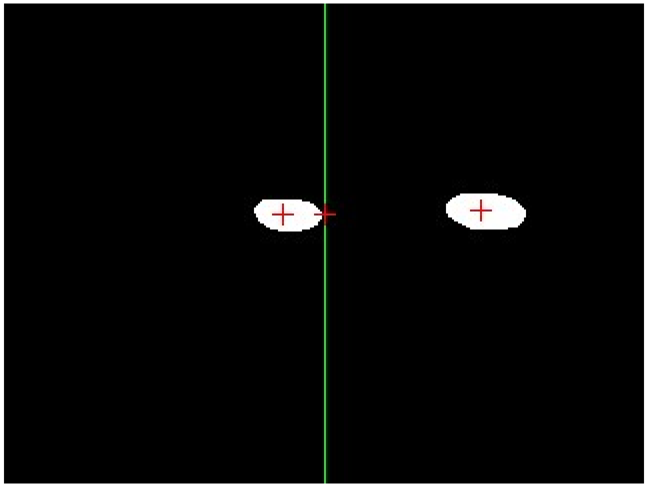
Object Detection and Find Intersection



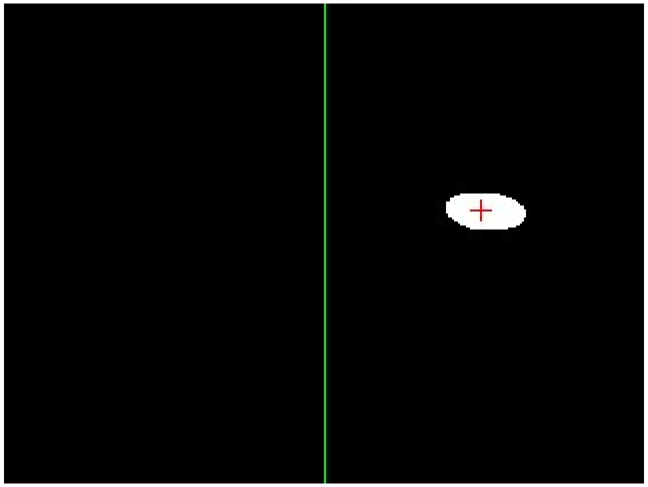
Object Detection and Find Intersection



Object Detection and Find Intersection



Object Detection and Find Intersection



Processing Results

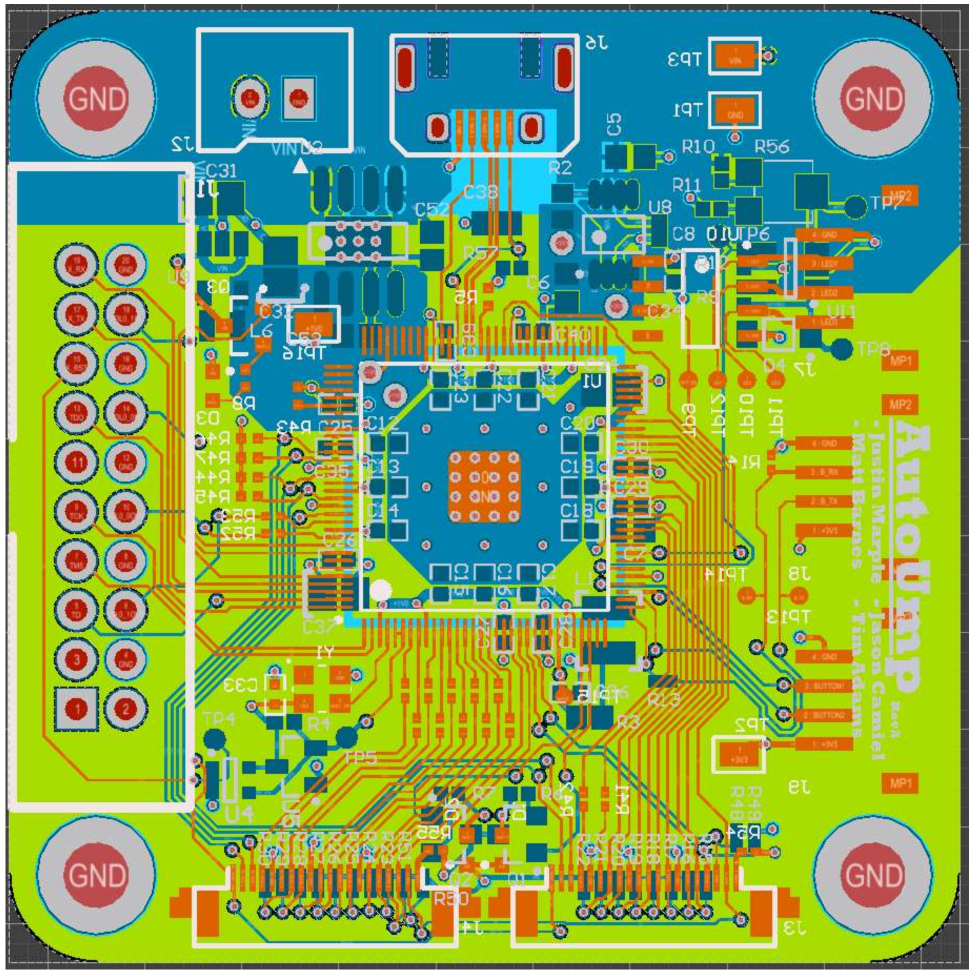
(Units in Inches)	Recorded		Calculated, Matlab		Calculated, C Code	
Pitch	**Width**	**Height**	**Width**	**Height**	**Width**	**Height**
1	**0**	**24**	-0.1	23.99	-0.191	19.07
3	**5.5**	**33.75**	5.4	28.2	5.72	23.09
4	**-3.5**	**31.25**	-2.2	27.8	-2.187	20.85

PCB's

- XMOS PCB
  - 49mmx49mm 4 layer board
  - 2000MIPS with 16 cores
  - 256kB of RAM, 2MB of FLASH

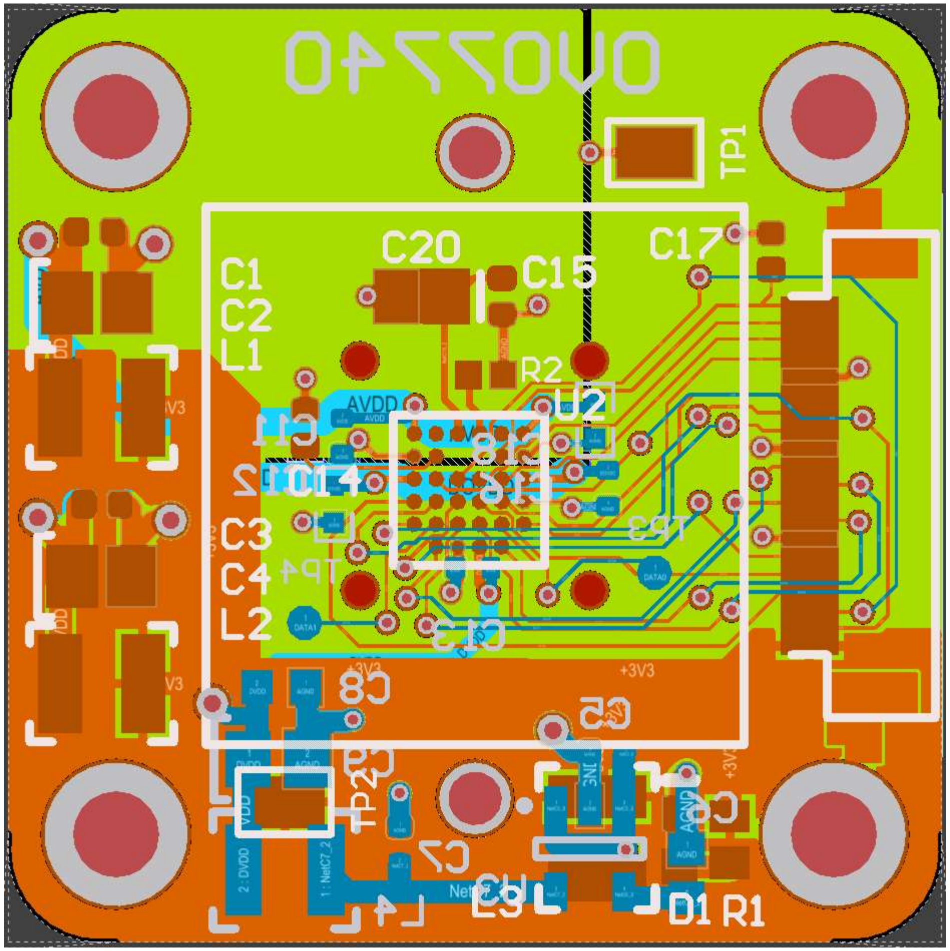
- Image Sensors PCB
  - 26x26mm 4 layer board
  - 0.65mm BGA pads requiring 4mil trace/spacing
  - Stencil was built to aid in the soldering of the BGA

PCBs



UMassAmherst

PCBs



Low level implementation

- Steps:
  - Outputs a 25Mhz CLK signal to cameras
  - Retrieves a 50Mhz Pixel-CLK signal back from cameras
  - Retrieves Horizontal and Vertical sync lines
  - Reads 8-bit data bus on falling edge of Pixel-CLK and saves into 32-bit buffer
  - 32-bit buffer is split into only the “Y” components of “YUYV” format
  - Run background subtraction for every pixel

Low level implementation – difficulties

- The XMOS board has horizontal and vertical sync signals are all on the same port
  - Each camera needs their own core. Very difficult to share a port event over two cores.
- A master thread is used to look for changes on the sync port
  - XMOS “channels” are used to communicate when a sync signal changes. Port counters are used to ensure that each core samples their respective camera at the correct time.

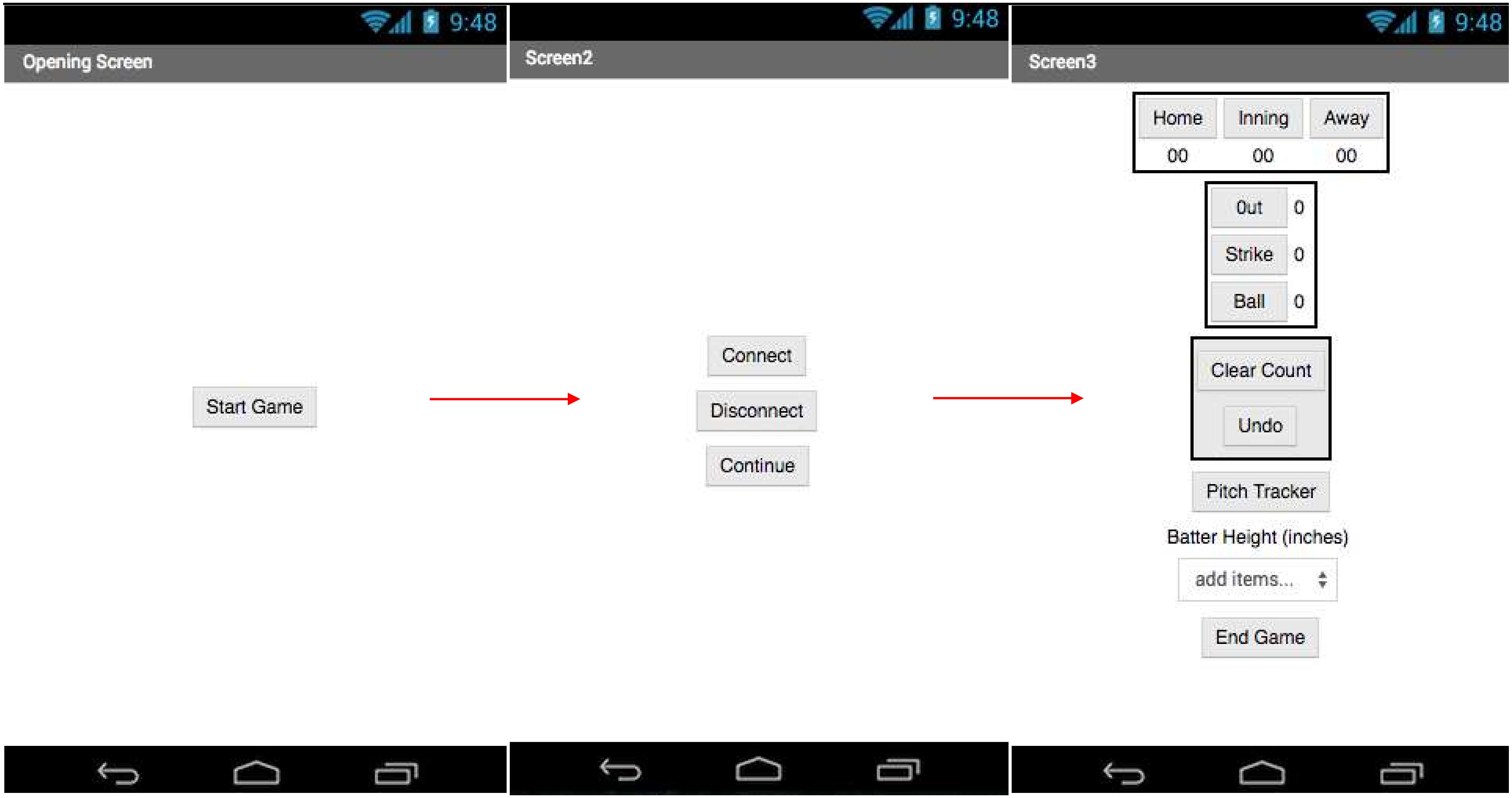
Low level implementation – difficulties

- Sync Frame Input pin is used to ensure the PCLK, data-bus, and sync signals are perfectly synchronized
  - Knowing one HREF/VSYNC of one camera means we'll know the HREF/VSYNC of the other
  - Critical for running the two cameras together reliably

Two way Bluetooth communication

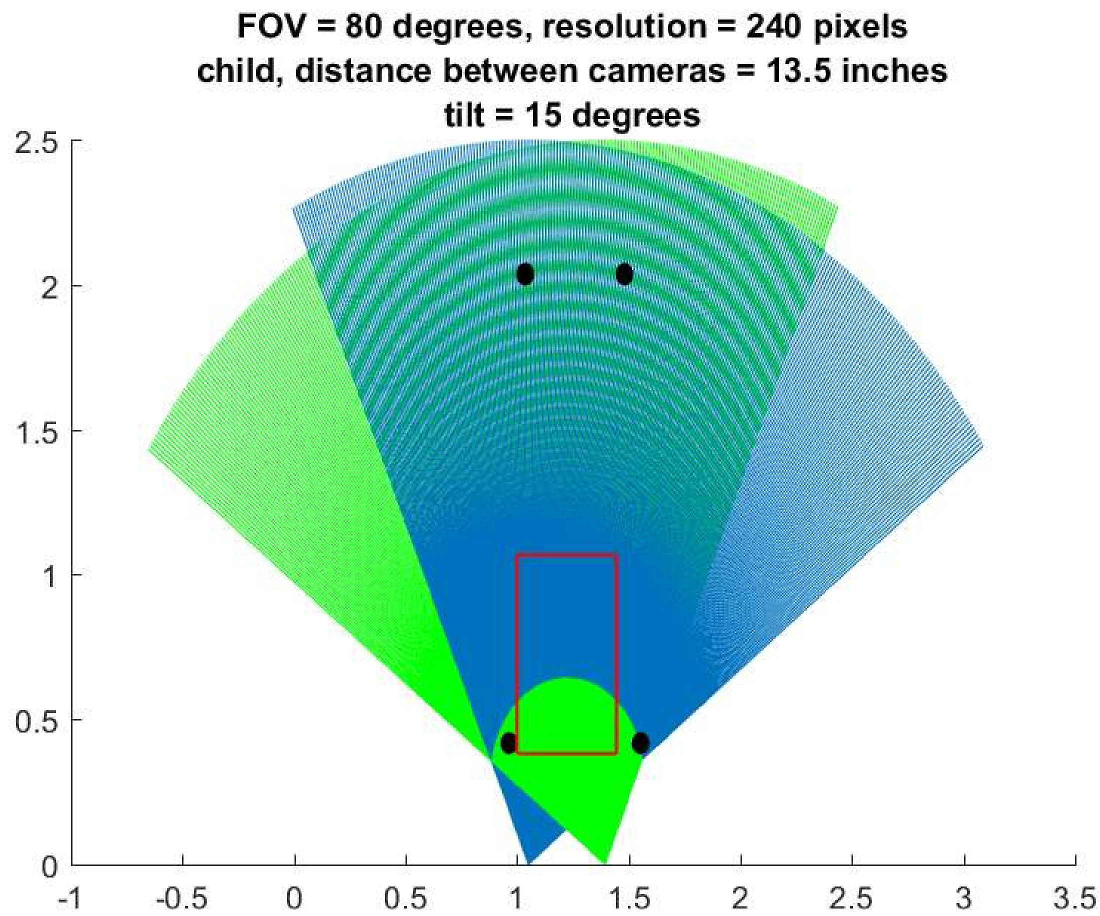
- Batter height is set per batter within in the app; this data is promptly sent to XMOS
- The XMOS drives the app with updates on ball count, strike count, and location
- With each set of updated data the app display is changed automatically
- In event of error, user may interact with the app to make adjustment or undo; updated data is sent to the XMOS

UI


9:48
Opening Screen
Start Game
9:48
Screen2
Connect
Disconnect
Continue
9:48
Screen3
Home
Inning
Away
00
00
00
Out
0
Strike
0
Ball
0
Clear Count
Undo
Pitch Tracker
Batter Height (inches)
add items...
End Game

Enclosure

- Cameras are placed 13.5 inches apart, tilted 15 degrees to have coverage of the strike zone and surrounding area



Enclosure

- The camera boards are placed on 3D printed mounts that tilt the cameras the desired 15 degrees
- The mounts are attached to a single surface to prevent them from rotating relatively to each other
- Aluminum sheet (3mm thick) is cut to fit inside outer edge of plate to prevent injuries
- Two support ridges are removed from plate to make room for cameras

Enclosure

- 30mm holes are cut into the plate slightly less than 13.5 inches apart
  - This prevents the “home plate” from blocking the camera’s field of view
- 34mm sapphire watch crystals will be inserted into a ridge cut 2mm below the top of the hole
- The cameras will be mounted so that the lenses are 4mm from the bottom of the lens

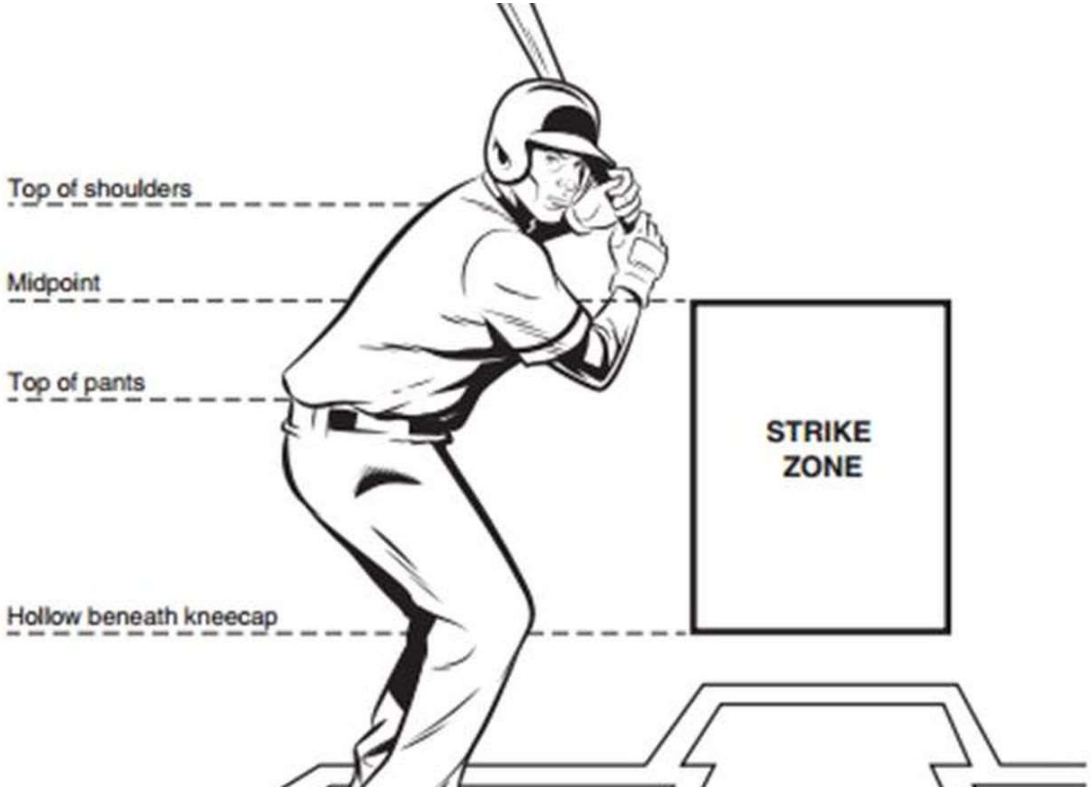
FPR Goals

- Code optimized and fully running on XMOS
  - Implement filtering algorithm to consistently isolate balls
  - Object detection and pitch determination each < ~16ms
- Detection accurate, sent to app
  - Primary issue, we believe: distortion. Calculate distortion coefficients.
  - Report pitch location and determination to app
- Complete enclosure
  - Insert sapphire crystals
  - Update camera mounts/fix to aluminum to create mount that can be secured underneath the plate

FPR Goals

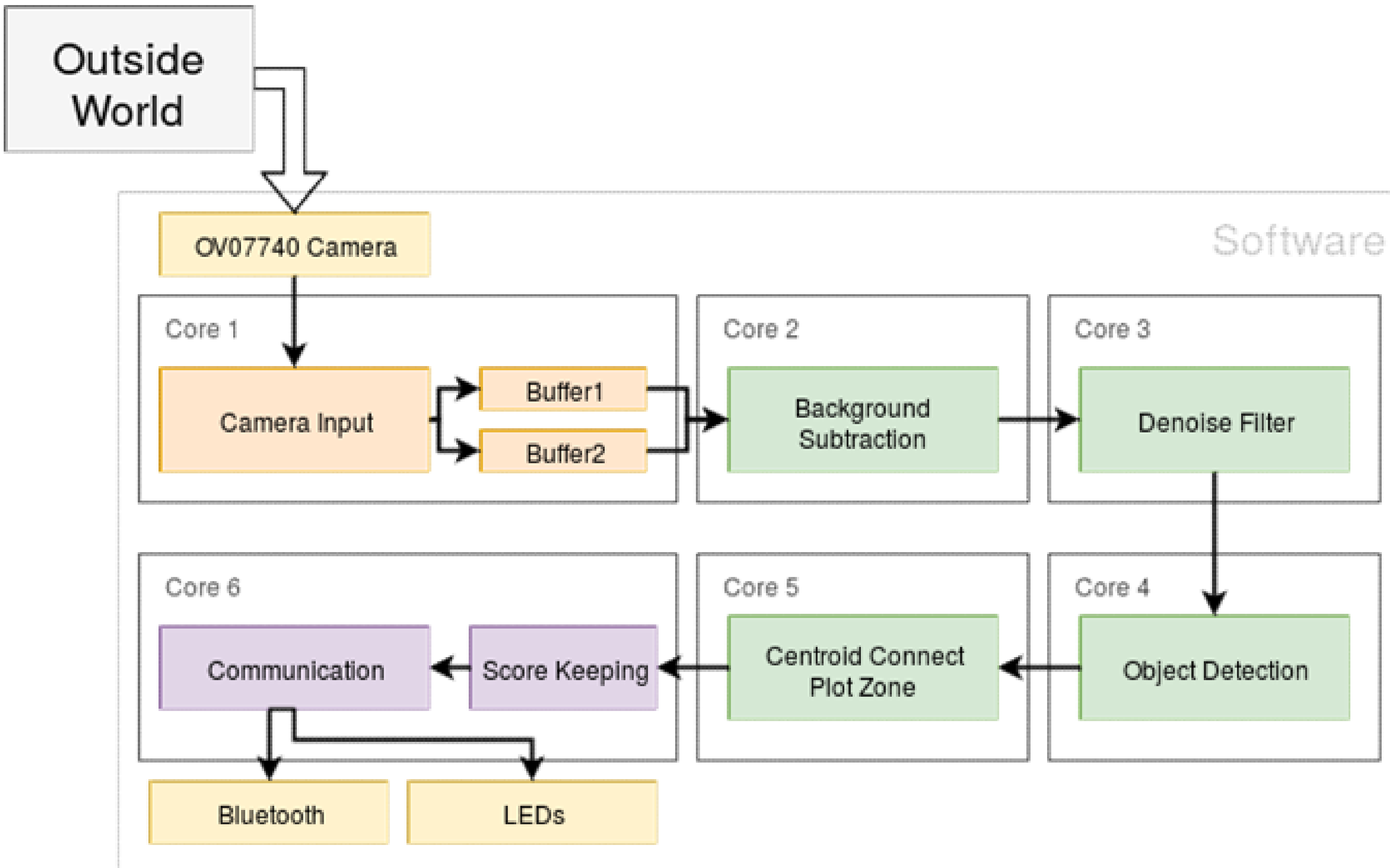
- Code optimized and fully running on XMOS
  - Implement filtering algorithm to consistently isolate balls
  - Object detection and pitch determination each < ~16ms
- Detection accurate, sent to app
  - Primary issue, we believe: distortion. Calculate distortion coefficients.
  - Report pitch location and determination to app
- Complete enclosure
  - Insert sapphire crystals
  - Update camera mounts/fix to aluminum to create mount that can be secured underneath the plate

Strike Detection Concept



Locate point in 3D space where ball intersects with the "Strike Zone" plane

Block Diagram - Software

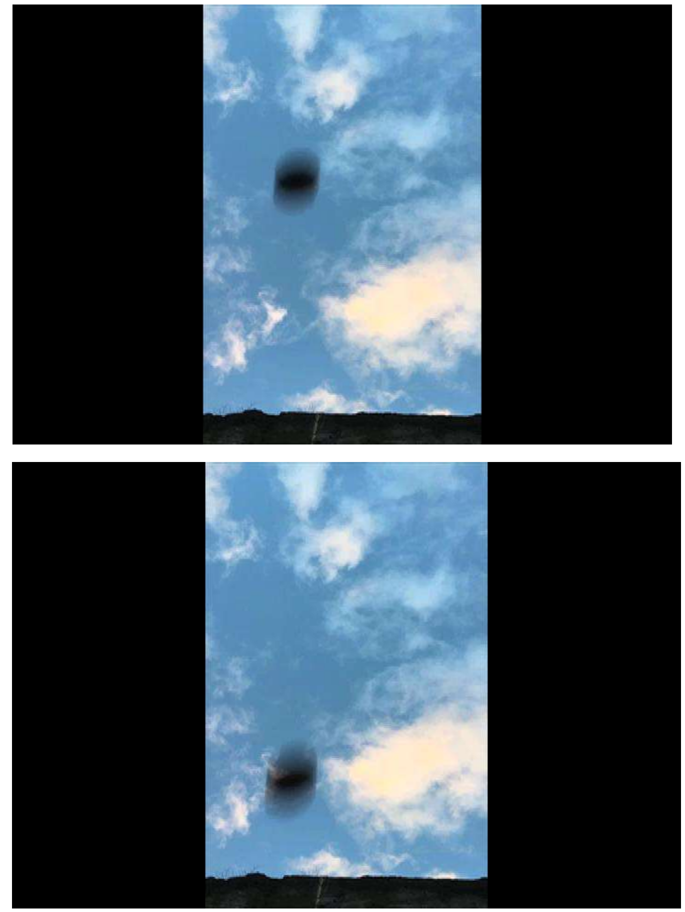


The Algorithm – Input

Left:

t

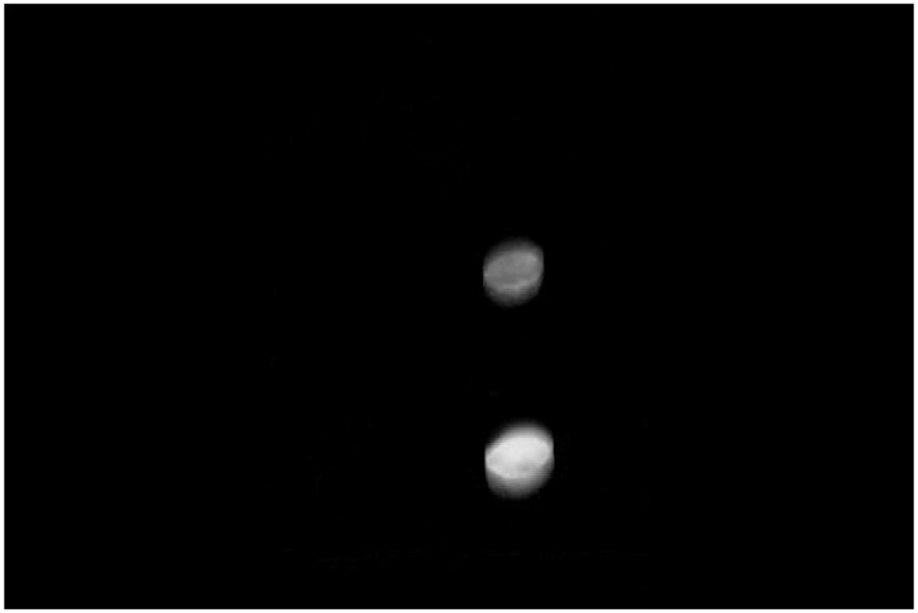
Right:



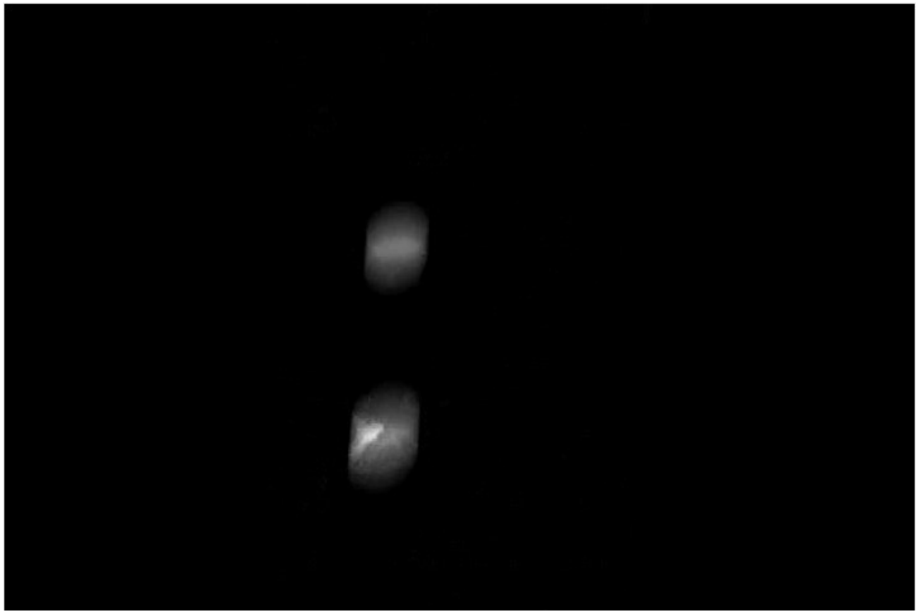
UMassAmherst

The Algorithm – Background Subtraction

Left:

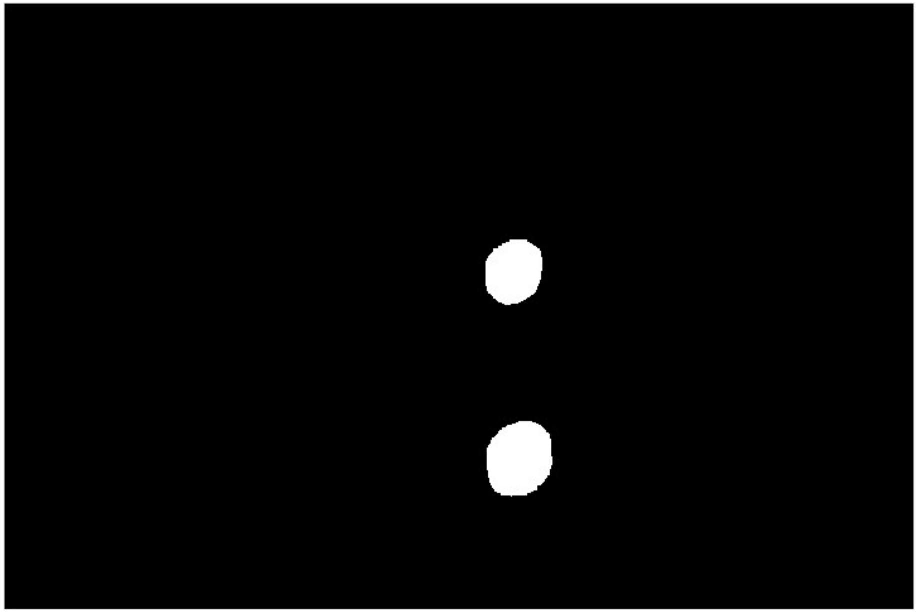


Right:

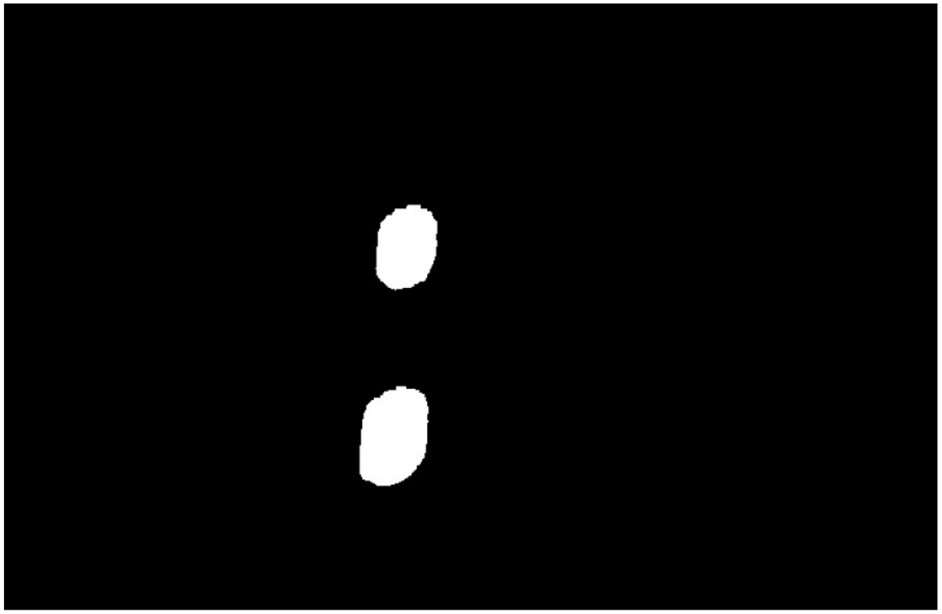


The Algorithm – Filter and Enhance

Left:

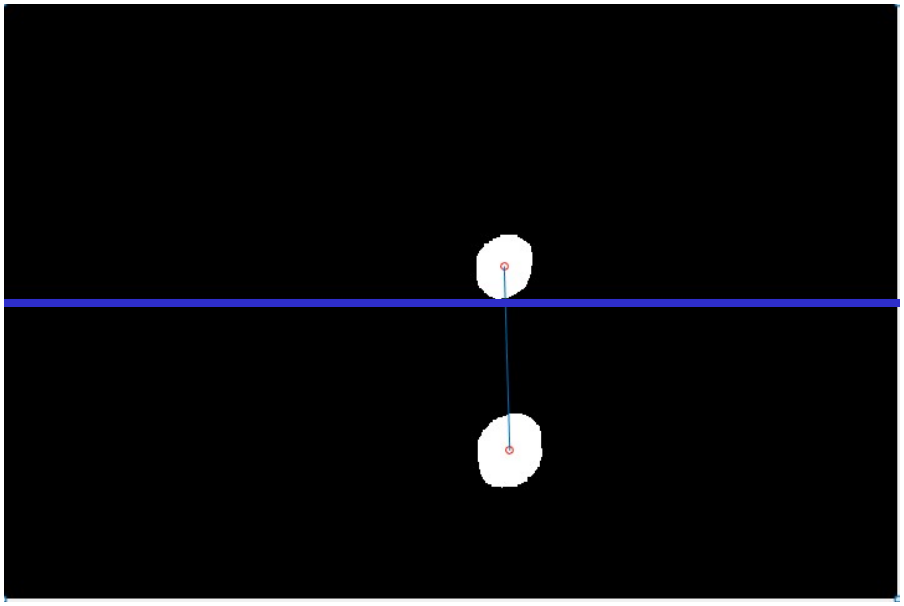


Right:

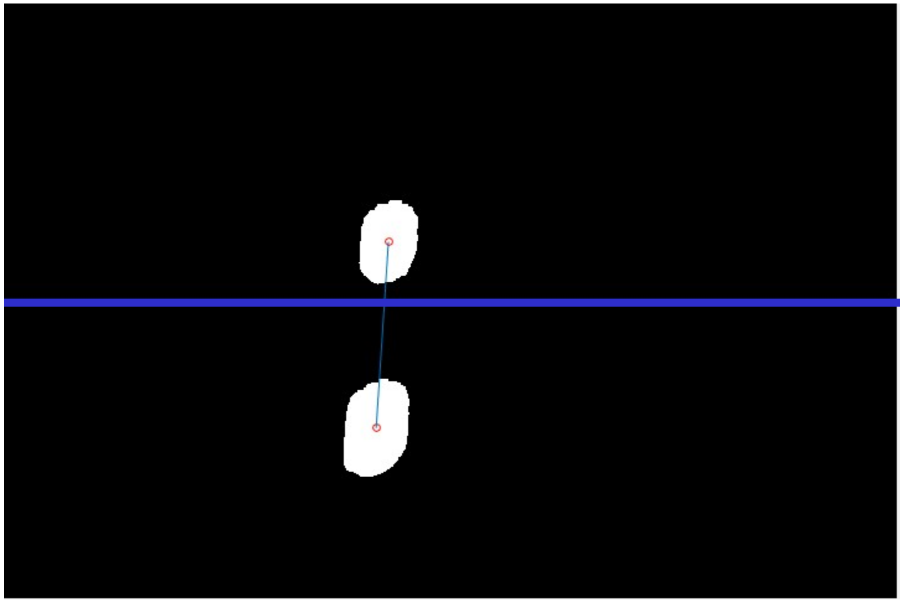


The Algorithm – Intersection with Strike Zone

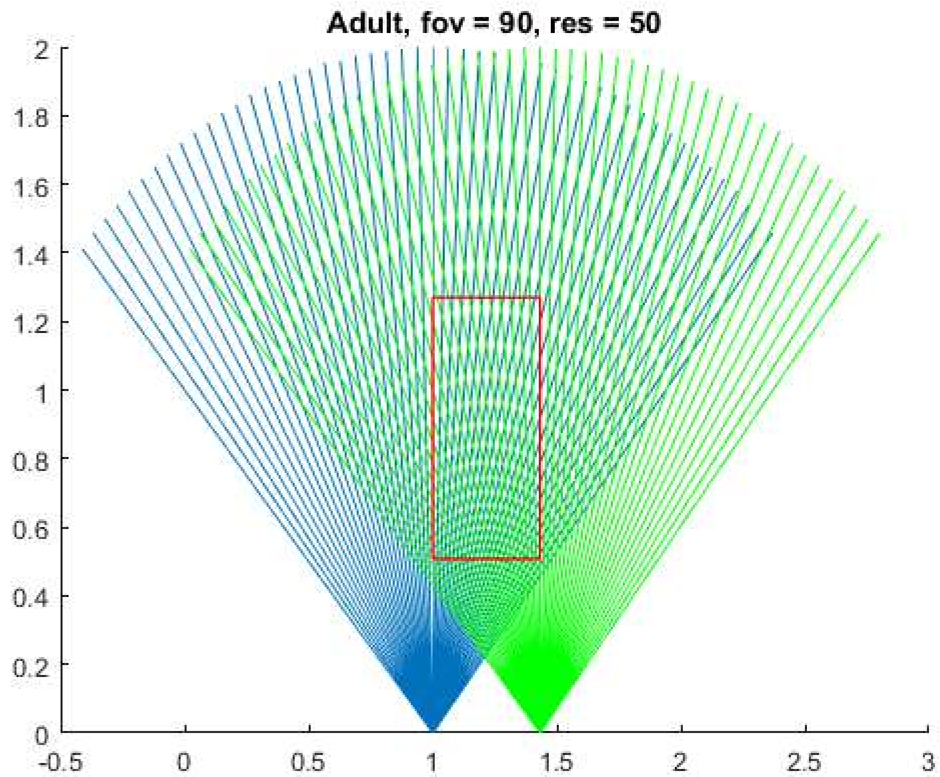
Left:



Right:

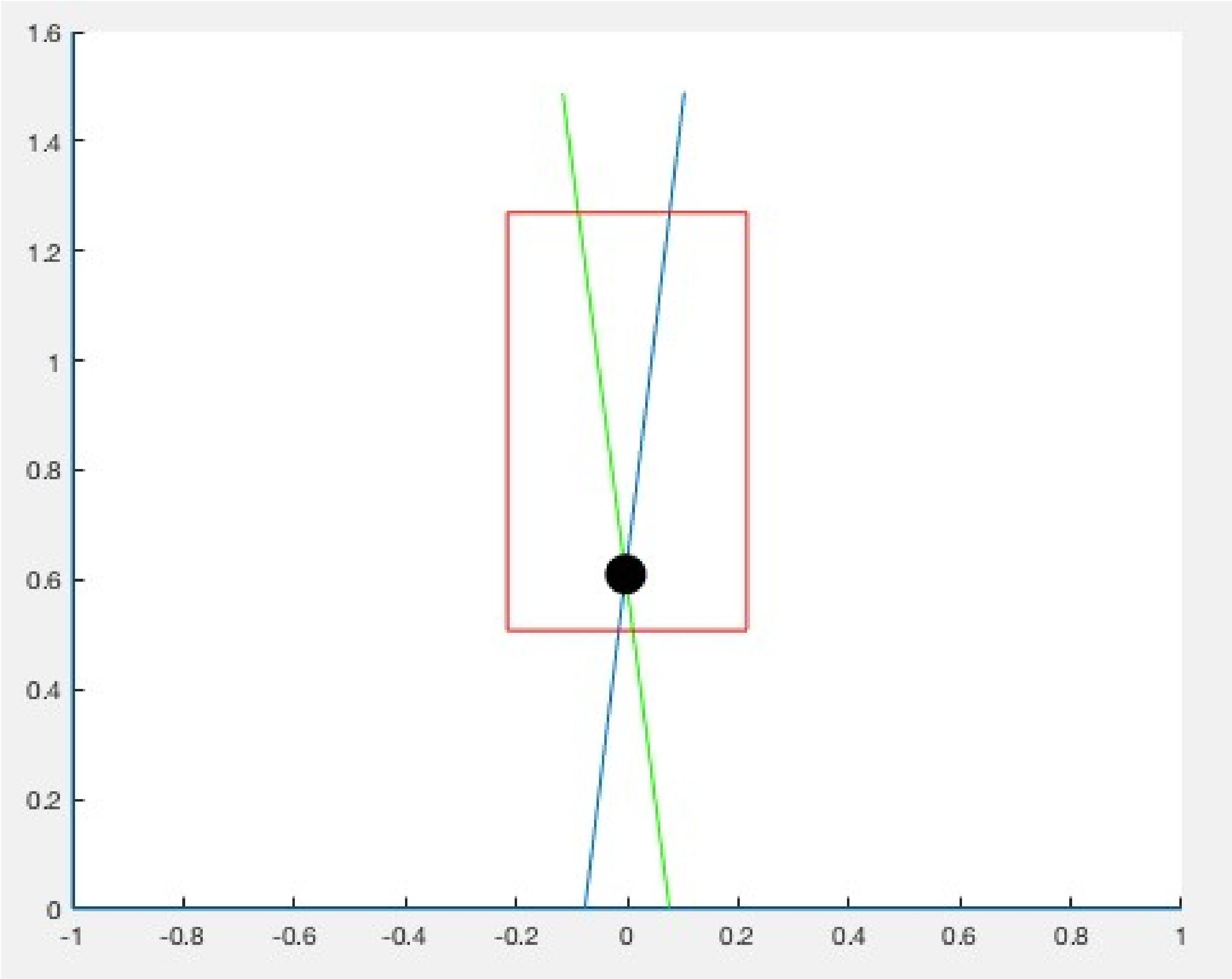


The Algorithm – Find Centroid Pixel

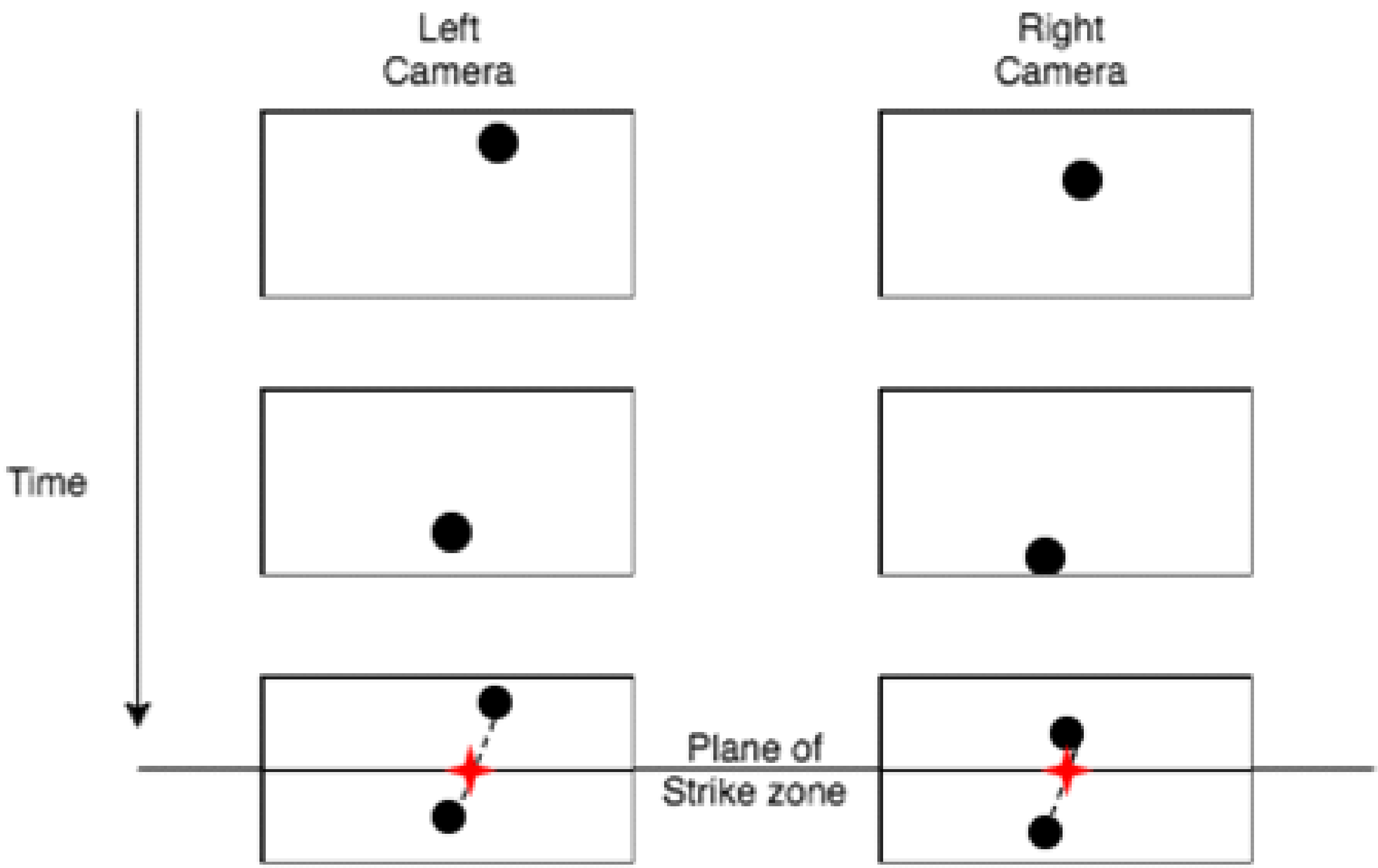


UMassAmherst

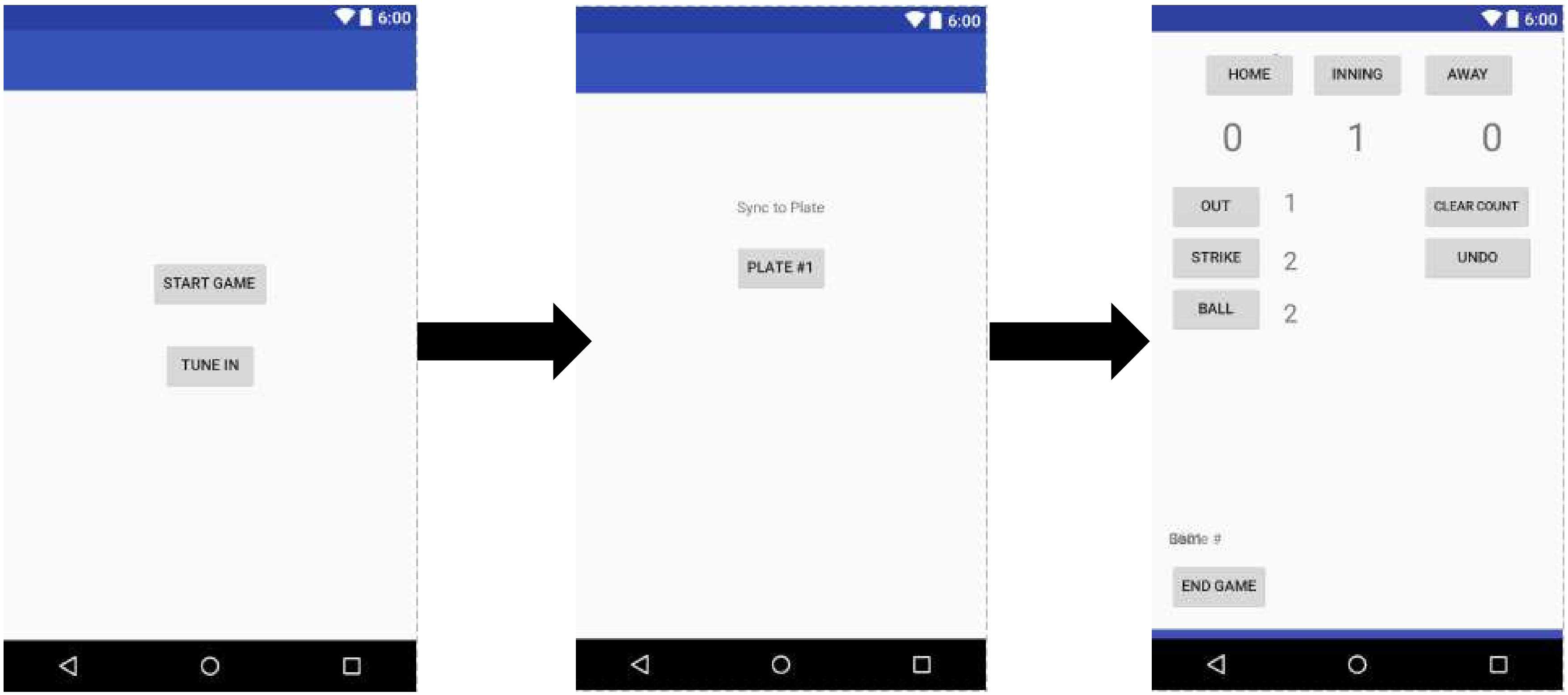
The Algorithm – Determine Ball Location



A Word on Synchronization

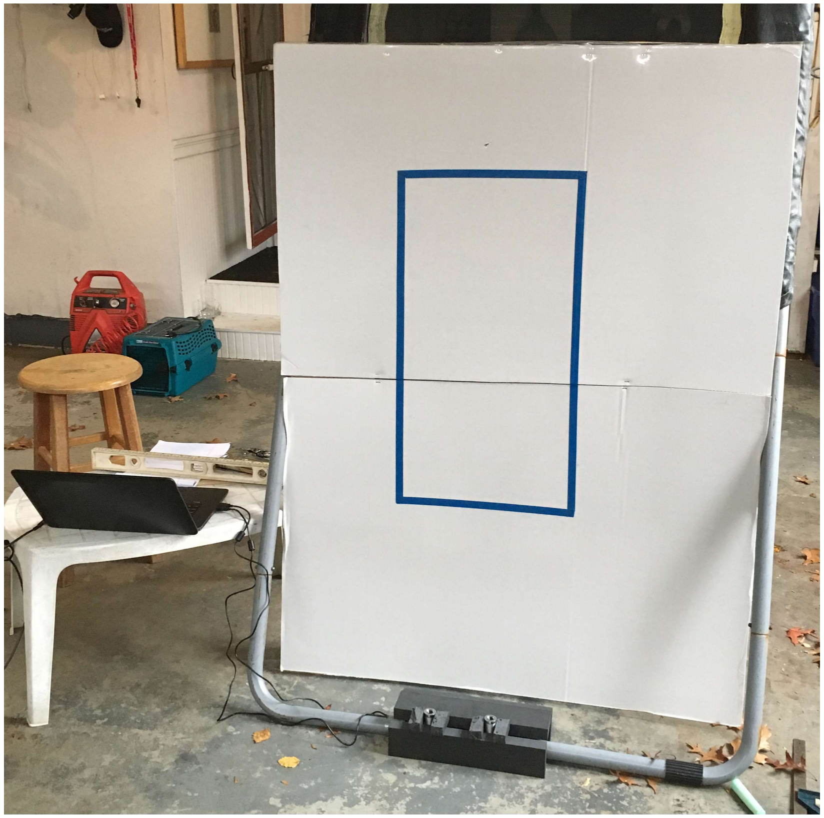


Android App UI



Indoor Garage testing

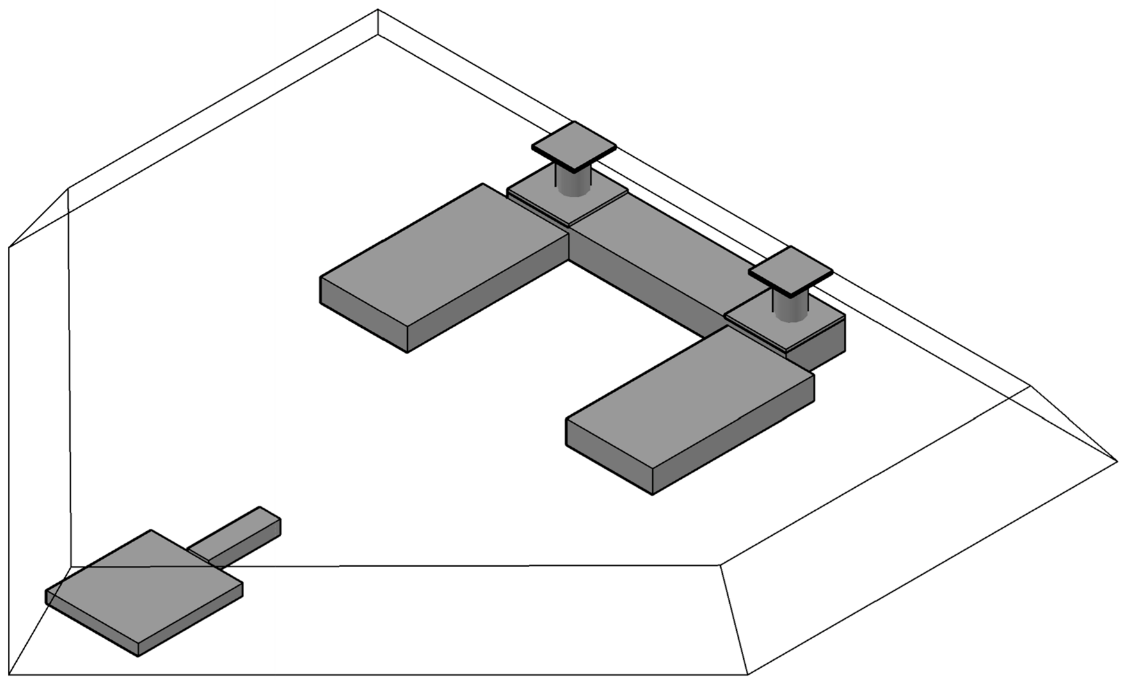
- Choose aperture and exposure time for outdoor use
- Cameras will need to be securely fastened to the plate in addition to each other
- Frame rate of 60fps required for accurate testing

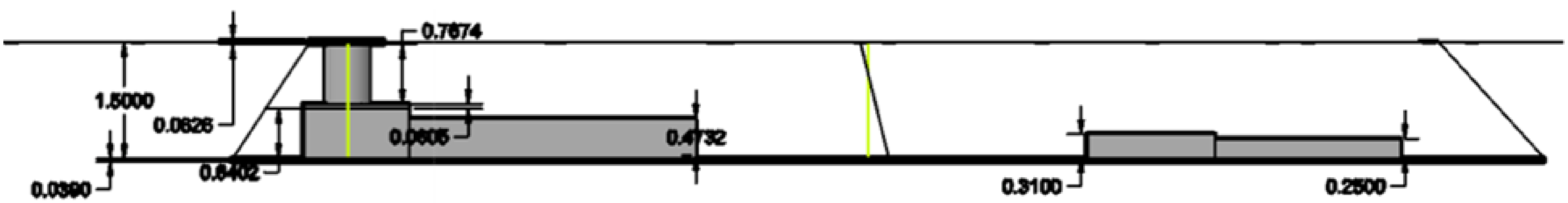


Wall Testing

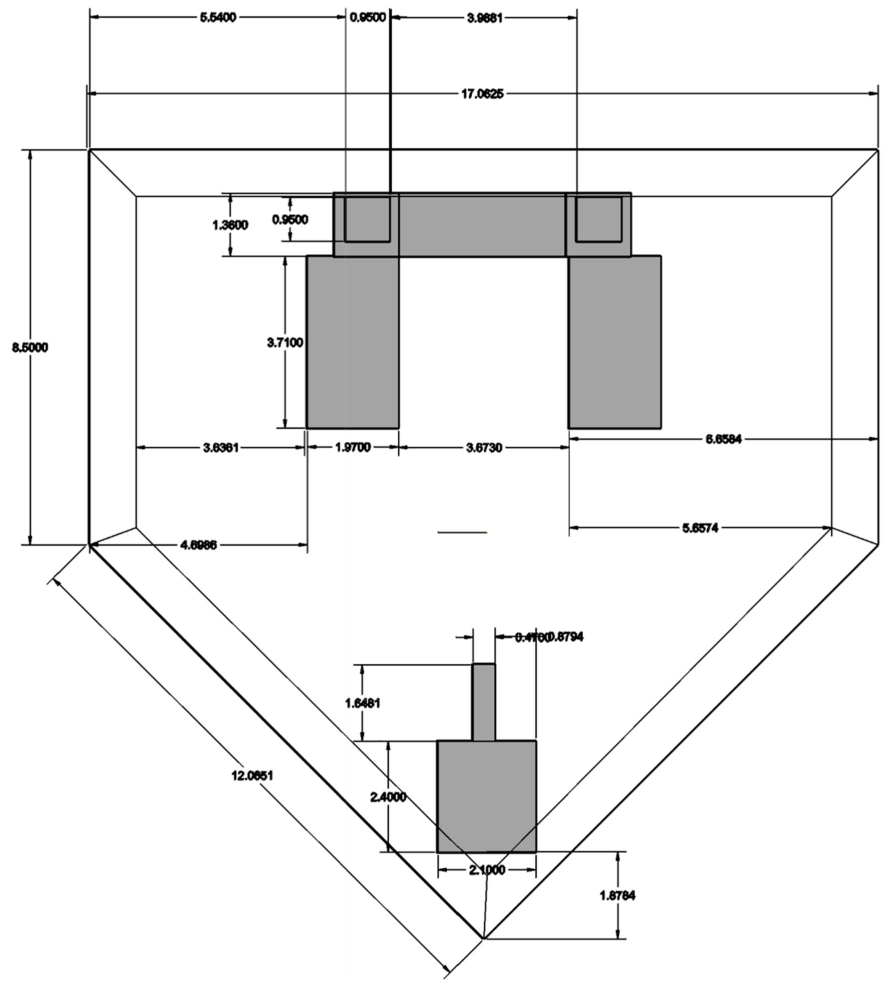
- Using two iPhones spread 6 inches apart and defined axis on a cement wall.
- Marked and measured pitches using chalk on the wall
- The data collected is used to perform validation of the location algorithm
- Sources of error that will affect perfect validation include
  - Camera movement/angle orientation
  - Ball arc between camera and mark on wall a distance of approximately 18 inches.
  - Human error in the marking of the position on wall

Enclosure





Enclosure Top View



Power Requirement

- Power consumption for processors
  - 3.7V*0.45A = 1.65W
- 4 hour game time: 6.75Wh
- Charge: 1824mAh
- Result: Choose a 2000mAh battery

Required Parts List

- OV7740 Camera Chip (x2)
  - 60fps
  - SCCB Interface
- XEF216-512-TQ128 (x2)
  - 2MB flash
  - 512kB RAM
  - 16 cores
- OV7440 Lens (x2)
- OV7440 Mount (x2)
- 2000mAh battery

CDR Deliverables

- Demonstrate complete system functionality
  - Implement ball detection, strike zone intersection, and baseball height functions in C (Tim)
  - Functioning PCBs (Justin)
  - Capable of detecting balls/strikes with hardware at 60fps
- Implement full app functionality (Jason)
  - Pair to plate, complete viewer and umpire views
  - 2-way Bluetooth communication established
    - App receives ball/strike information
    - Plate receives batter height
- First iteration of enclosure built (Matt)

Timeline

- Complete orders for final system
- Design and send out order for PCB

Dec 21^{st}

- Implement full algorithm in C
- Achieve image collection on PCB system
- Receive second round of PCB if needed

Jan 21^{st}

- Bluetooth communication established
- App functionality completed
- Implement full algorithm in XC

Feb 21^{st}

- Enclosure completed

Feb 28^{th}

Demo – Image Collection and Object Detection

- Image size: 176x144
- Frames per second: 30 -> ~33.3 ms/frame
- Limited memory space: can only store two images at a time