AutoUmp

Timothy Adams, Matthew Barnes, Jason Camiel, Justin Marple
Faculty Advisor: Prof. Tilman Wolf

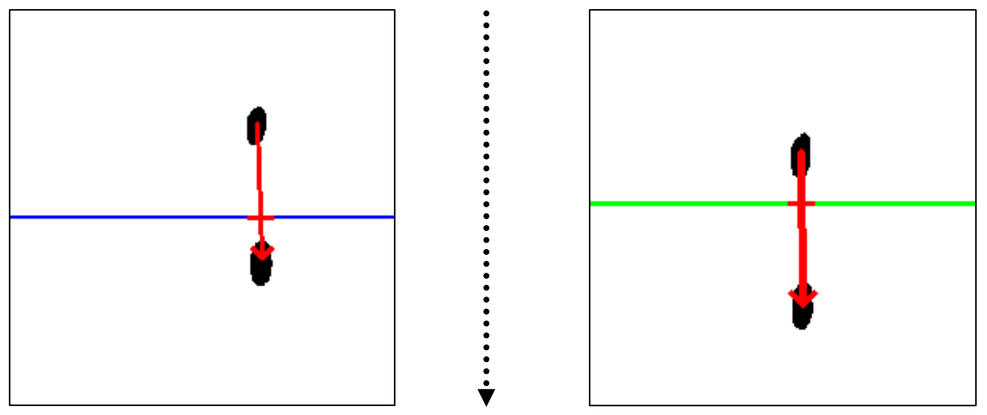
Abstract

Determining strikes and balls accurately is a core aspect of baseball. However, umpires are inaccurate or biased, and current technology solutions used in the MLB are prohibitively expensive. AutoUmp is a self-contained pitch calling solution installed in the home plate itself. It uses optical sensors and real-time image processing algorithms to detect strikes and balls.

System Overview

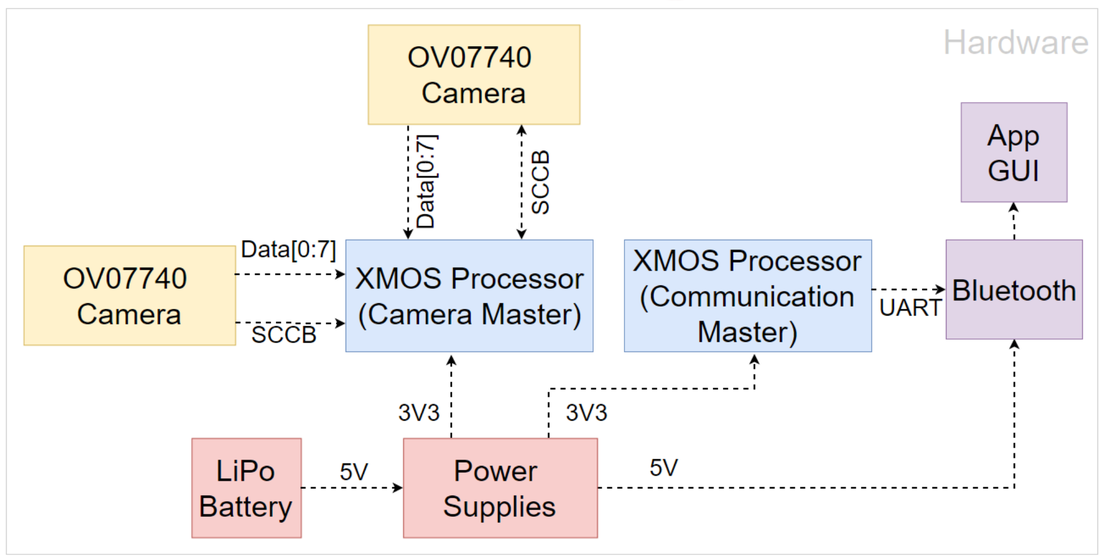
AutoUmp employs two high-speed cameras which detect a pitch as it moves across the plate. By tracking the trajectory of the pitch, we can interpolate the precise pixel location of the ball as it crosses the middle of the image, which represents a vector in the 2-D strike zone plane. By finding the intersection of the vectors obtained from both cameras, we are able to find the location of the pitch and determine whether it is a ball or a strike.

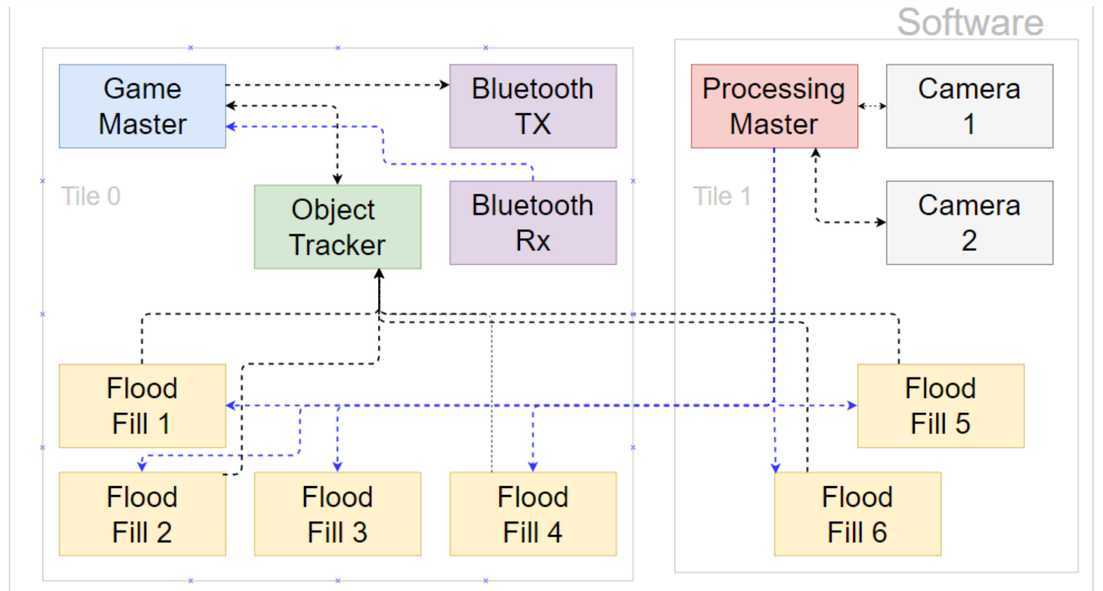
Problem

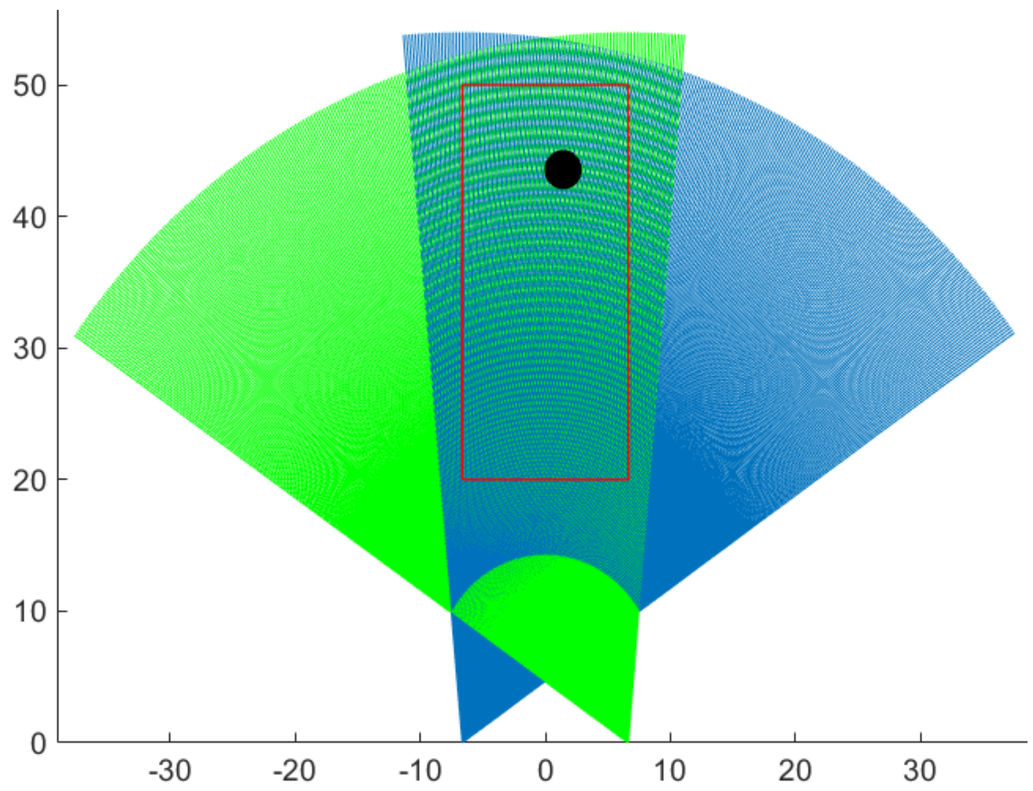


Finding the intersection of the ball with the strike zone plane in each camera

Block Diagram







The precise pixel location found in the previous step from both cameras can be used to find the location of the ball in the 2-D strike-zone plane.

Specifications

Specification	Goal	Actual
Accuracy of professional ump	85% accuracy in pitch classification	83% accuracy in pitch classification (n = 100)
Useable for batters up to heights of 6'6"	Detect 70 in up, 30 in to each side	Detect 60 in up, 20 in to each side
Pitch speeds of 70 mph	Operate at 60 frames per second	Operate at 60 frames per second
Real time use	<2 second delay	<1 second delay
Battery life > length of game	3 hour battery life	20 hour battery life (2.2W operation)
Robust, self-standing system	Self-enclosed, withstand impacts of normal play	Self-enclosed, withstand impacts of normal play
Enable varying heights	Control strike zone via app	Control strike zone via app

Results

	Actual Strike	Actual Ball
AutoUmp Calculated Strike	34	14
AutoUmp Calculated Ball	3	49

Pitch Calling Comparison Matrix				
	Little League Umpire	MLB Umpire	Pitch FX (MLB)	Our System (AutoUmp)
Accuracy	**~50%**	**85%**	**99.9%**	**83%**
Cost	**$**	**$$$**	**$$$$**	**$**

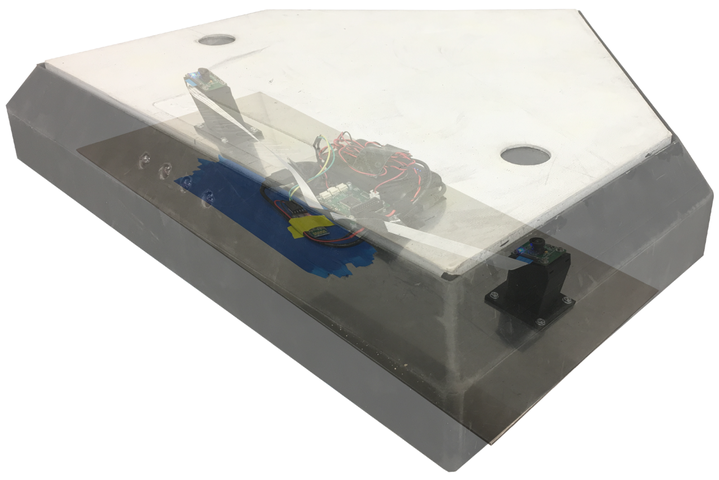
Acknowledgements

Our hearty thanks to Professor Wolf for his advice, support, and his many jokes made with us and at us. It's been an honor to be your SDP team. Thanks also to Professors Anderson and Moritz, our evaluators; Fran Caron for his help with our orders; Professor Hollot for his flexibility; and the ECE department for the funding.

Enclosure

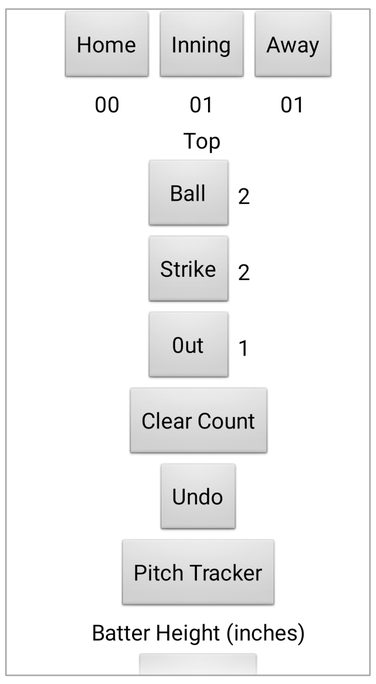
To perform effectively in a real world environment, the system requires an enclosure that will protect the electronics during normal gameplay, including being stepped on, slid into, or with a bat. Holes in the plate allow the camera to see through, while embedded sapphire crystals protect the lenses from shock, dirt and sand. The sapphire crystals were selected due to their ranking on the Mohs hardness scale (9), which exceeds that of quartz (7), a material typically found in sand. 3D printed mounts raise the cameras to just beneath the crystals and are angled 15 degrees inward to allow each camera to see the entire strike zone.

An aluminum backing fits snugly into the plate, and is secured by Velcro straps to ensure system stability throughout the game. These straps allow for the backing to be removed for battery charging.

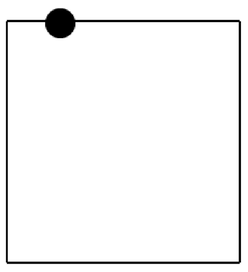


App/GUI

The app is the main interaction point between our system and the user. After connecting via Bluetooth to the system, the user enters a screen where a summary of the game is displayed. The pitch count will be automatically updated by the system as it detects balls or strikes. Should the user desire to make a change, they can use the undo button or increment any of the values through the buttons provided.



The app also provides a pitch tracker screen, where the user can view the location of the pitch, as well as the current pitch count. This screen is designed to mimic the view provided by the Pitch F/X technology used in the MLB and that is seen on TV.



Cost

Development

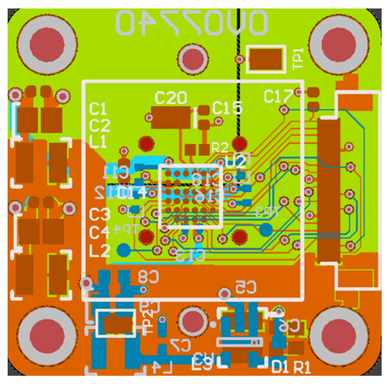
Item	Cost
Initial Prototype	$ 380.26
PCBs and PCB Parts	$ 478.28
Enclosure	$ 198.97
Misc.	$ 185.69
Total Cost	**$1,243.20**

Production

Item	Qty.	Unit Cost (Per 1000)	Total
Optical Lens	2	$ 1.50	$ 3.00
10000mAh Battery	1	$ 7.00	$ 7.00
Home plate	1	$ 61.99	$ 61.99
HC-05 Bluetooth Module	1	$ 2.60	$ 2.60
Sapphire Crystal	2	$ 5.00	$ 10.00
Image Sensor	2	$ 2.11	$ 4.22
Processor	1	$ 9.75	$ 9.75
PCB Parts	1	$ 11.47	$ 11.47
XMOS PCB	1	$ 3.99	$ 3.99
Image Sensor PCB	2	$ 3.86	$ 7.72
Camera Mounts	2	$ 4.07	$ 8.14
Total Cost			**$129.88**

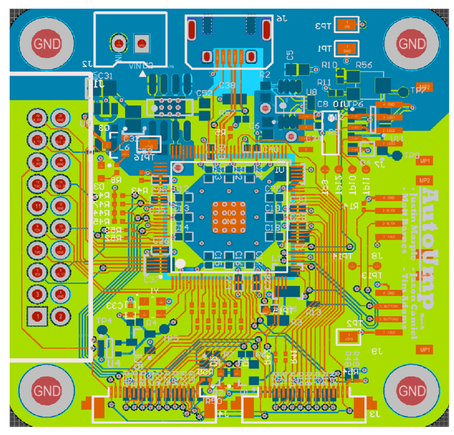
Hardware

To meet the real-time demands of our system, we chose a 2000MIPS, 16-core XMOS-XS2 processor, designed explicitly for inter-thread and inter-core communication. This choice enabled us to pipeline our algorithm and meet our throughput requirement.



Bottom layer of custom designed camera PCB

We designed a PCB for this processor, complete with power control, UART communication, and an implementation of the camera SCCB protocol. The camera boards themselves were custom designed, built for the Omnivision OV7740 image sensor, which was chosen for its ability to capture data at 60fps at a 320x240 resolution – a resolution high enough to meet our requirements but low enough to enable real-time image processing.



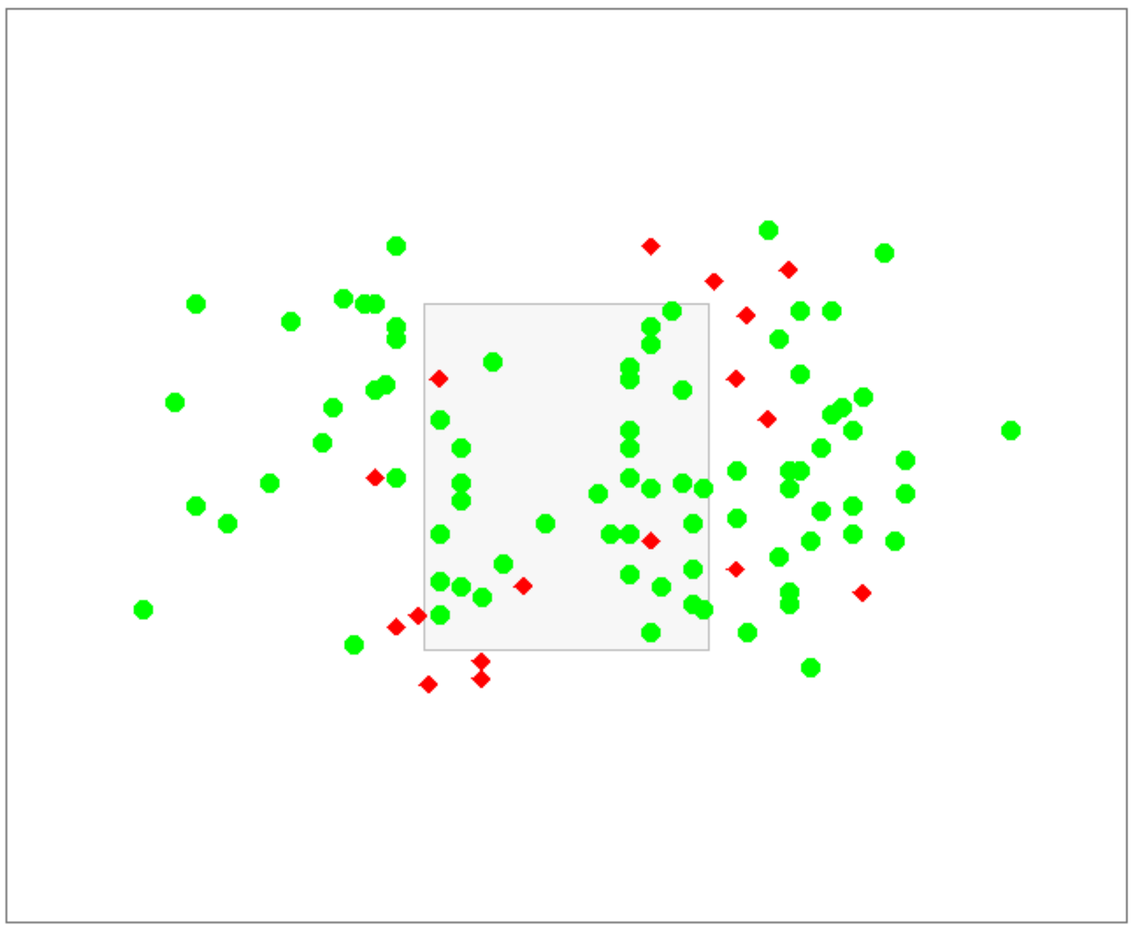
Top layer of custom designed processor PCB

Image Processing

We begin our image processing with background subtraction to detect motion, a process where the current frame is subtracted from the previous frame. The resulting image removes the background and sets objects in motion as white pixels. This step is implemented in assembly to run within the constraints necessary, and runs on the same thread that collects the raw camera data.

The next step, denoise and object detection, is parallelized across 6 cores, as it is by far the most computationally expensive step. Pixels are examined and removed if less than 3 of their 4-connected neighbors are white. Object detection then begins by finding connected sets of white pixels. The output is an array of objects, each modeled as rectangles.

These arrays are passed to an object tracker core, which unites the information from all 6 cores to track a pitch. When the ball passes the middle of the screen, the strike zone plane vector is calculated and a flag is set for that camera. When both flags are set, the double camera information is combined and the pitch is calculated. The result is then sent to the app via Bluetooth.



Result of 100 pitches. Blue means correct call was made; red indicates incorrect call