Journal Report 3
9/16/19-9/23/19
Austin Huang
Computer Systems Research Lab
Period 1, White

Daily Log

Monday September 9

I applied working algorithm of tracking a ball during a visible pass to old video. Changed parameters of initial bounding box to fit the pass, but the algorithm did not work. Returned to old NCAA video where I was able to successfully track a pass and located another frame where another pass was made.

Tuesday September 10

Scoured the VideoCapture API and found the .set() method. Changed parameters of initial bounding box to fit. Applied algorithm, doesn't work. Realized something either wrong with algorithm or video quality. Spent rest of class researching. Realized my background subtraction did not work as intended. Looked at NCAA video, turns out it is 30 fps.

Thursday September 12

Went back to original pass that did work and tried to find reasons why pass does not work. Did research on ways to implement dynamic camera background subtraction. Do not completely understand this new algorithm.

Timeline

Date	Goal	Met
Sept 9	Locate and mark a lacrosse ball when it is visible in frame	Was able to mark and track ball, need to address locating it.
Sept 16	Locate and mark a lacrosse ball when it is visible in frame	Refer to above.
Sept 23	Locate the lacrosse ball when it is visible in frame and fix tracking a pass issues	to be determined.
Sept 30	Identify and track all players on the field	
Oct 7	Identify and track all players on the field	

Reflection

I don't know how I did not discover this before, but the VideoCapture.set() method allows the video to faster advance frames to start the video on a designated frame I want. This makes running and debugging easier then advancing frames manually with read and while loops, which is what I had been doing for frames that were 5000 into the video before.

Sometime in between this class and previous class, my windows computer bricked. I logged in and the screen froze and keyboard kept on blinking. Luckily I was able to somehow un-freeze (thaw?) the computer by restarting it approximately 10 times and plugging in the charger. This is a sign that I should keep code stored on cloud or on usb in future.

I initially thought that the background subtraction method in conjunction with the tracker was enough to track the ball. I used the built in BackgroundSubtractorMOG2 found in OpenCV and the CSRTtracker. However, I soon realized that in my code I only used the backgroundSubtractor once in the initial frame and never actually updated it. This meant that my algorithm didn't use the background subtractor at all. The only reason I thought that I needed the background subtractor was that it advanced a frame on the initial set up that pushed the ball just into the correct spot for the bounding box to encapsulate it.

I got rid of the background subtractor and adjusted the frames to test this hypothesis, and sure enough, the pass worked. Now, I needed to answer the question why it worked for this particular pass and not for others. The answer is most likely that the pass that worked was a small handoff between players. The speed at which the ball was moving was small between frames and likely more defined as a circle.

Research into the background subtractor, which I still believe may be key to tracking the ball, if implemented correctly, proved complicated. Turns out the backgroundsubtractor for opencv was intended only for static backgrounds and the way it works is to differentiate moving objects from static. However in game footage, the camera is moving, so both the background and foreground are moving elements of the video. I found a previous project that implemented a dynamic camera background subtractor, but the algorithm they used was complex. It involved scanning the frame for key lines or shapes and somehow determining its distance from the camera. Will probably either need to spend more time researching this dynamic camera background subtractor or move on to another way to tackle my problem.