Journal Report 12

12/9/19-12/16/19

Austin Huang
Computer Systems Research Lab
Period 1, White

Daily Log

Monday December 9

Went back to apply to dirty passes. Start velocity was faster than other pass in horizontal direction. Could not keep up. Increased initial bounding box size.

Tuesday December 10

After thinking about it at home, realized it is better to increase background subtraction to entire frame at least for initial frames to better apply to every pass. Fixed some errors when tracking the dirty pass, like updating bbox with actual and not attempted. Added back in the synthesis function.

Thursday December 12

Programmed in the primary synthesis function to combine dxdy with background subtractor using dxdy as a check. Dirty pass now works and tracks through white player, and it was reasonably quick. Felt very proud. Went back to test on old pass with new synthesis function. Works. Went back to old computer to get back other game video to test other passes. For the initial pass that the ball went into the red track, pass still does not track.

Timeline

Date	Goal	Met
Dec 9	Finish implementing dxdy and combine it with background subtractor	was able to make code run faster by limiting background subtraction thresholded region with dxdy.
Dec 16	Make code more accurate by either adding another optical flow method or combining dxdy and background subtraction	Able to track one dirty pass through a white player
Dec 23	Diagnose exact reasons why I cannot track certain passes and also add method to detect end of pass	
Dec 30	Rest	
Jan 6	Rest	
WINTER GOAL	I will have an application that will reliably track any pass in reasonable time. (¡10 seconds per frame).	

Reflection

The dirty pass made me realize that it was essential to have the initial frames track the initial velocity of the ball. I don't think there is a way to ensure I have a bbox big enough to track the changing positions of the ball between frames, unless I scan the entire frame, which is why I had to revert to the old, slow, test-everything algorithm for the first few frames. As a result, in subsequent frames, after gauging the velocity, I was able to more reliably set a bbox to capture the ball between frames. I was able to track the dirty pass I was previously having trouble with. This was because of my new synthesis method, which checked the background subtractor CSRT tracker with the dxdy approximation. Essentially, the method checks whether the CSRT tracker bbox and the dxdy tracker bbox are close to each other. If they are, then the algorithm will trust the CSRT tracker and will set the actual dxdy bbox to the CSRT tracker. It means that the frame sequence was stable and more likely than not, the ball was tracker successfully. What I noticed before was when the ball got obscured or travelled a long distance between frames, the CSRT tracker would jump to an unreasonable location far away from the ball. To remedy this, the algorithm will trust the dxdy when this happens, expand the boundaries of the dxdy bbox approximation, and reset the tracker to that box. By doing this, hopefully the expanded bbox will contain the ball, which the newly reset tracker will pick up and continue to track. While this pass was able to successfully track a pass through an obstacle, one problem this now creates is a difficulty in detecting when a pass ends. Right now, I hard code in the ending of passes and have nothing actually coded in to detect. However, I think if I can use dxdy to predict the ending of a pass. Logically, the magnitude of dxdy in a given pass will at a maximum decrease twice: once at the apogee of the parabolic pass, and another at the end of the pass. I could probably use this fact to detect when the ending of a pass occurs. I still was not able to detect the example pass in the high school lacrosse game. After examining it further, I realized that the CSRT tracker could not track the ball even in the initial frames. This was most likely because of a problem with background subtraction and color detection. First the ball looks fainter than the other video, so there is probably not a lot of strong defining color separating it from the background. Next, the pass was at the top of the screen

where the ball was above a red track in frame. Because the frame average background subtraction color was green, my algorithm did not subtract the red track, so I think the CSRT tracker had a more difficult time identifying the ball it was supposed to track. When the background subtractor cannot initially detect the ball in the first frames, I have no other way to track, and right now as my algorithm is, the whole code just falls through. dxdy only starts happening after the initial frames when I get the initial velocity. I plan to think of possible solutions this week.