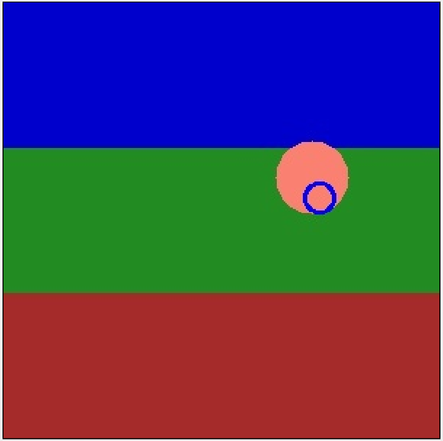
Computer Vision Spring 2020 Problem Set #5

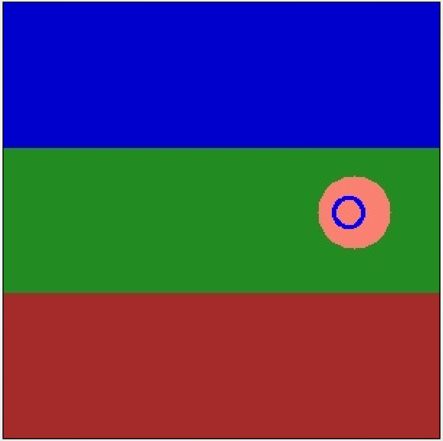
Andrew Parmar
aparmar32@gatech.edu

1b: KF Tracking a circle



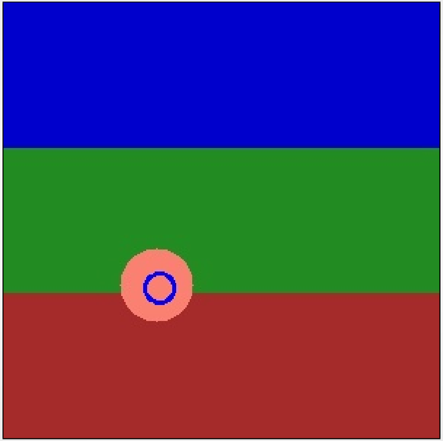
ps5-1-b-1

1b: KF Tracking a circle (cont.)



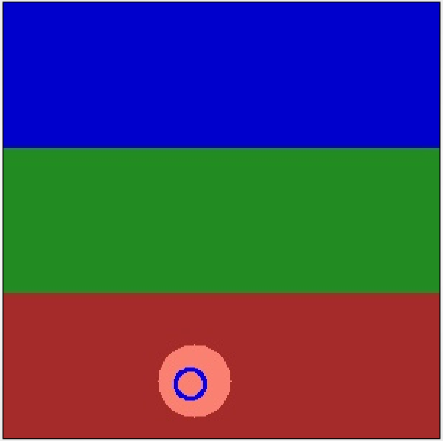
ps5-1-b-2

1b: KF Tracking a circle (cont.)



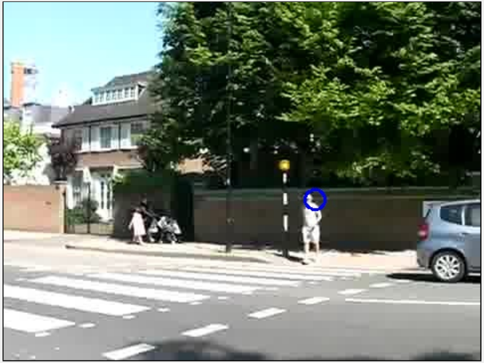
ps5-1-b-3

1b: KF Tracking a circle (cont.)



ps5-1-b-4

1c: KF Tracking pedestrians



ps5-1-c-1

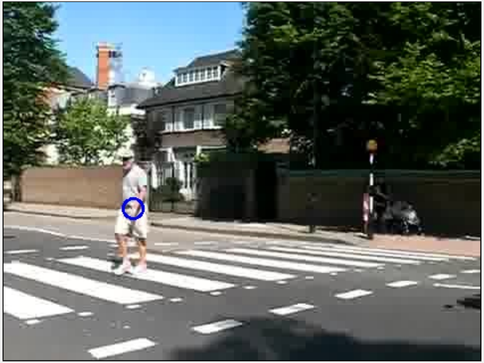
1c: KF Tracking pedestrians

ps5-1-c-2

1c: KF Tracking pedestrians

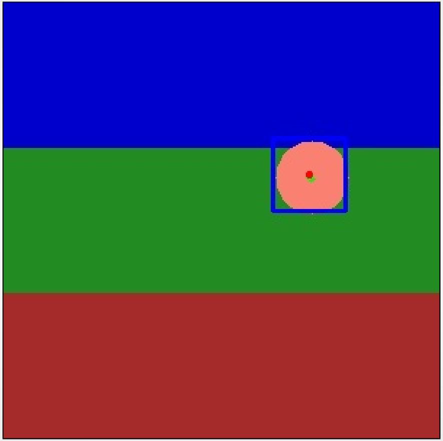
ps5-1-c-3

1c: KF Tracking pedestrians



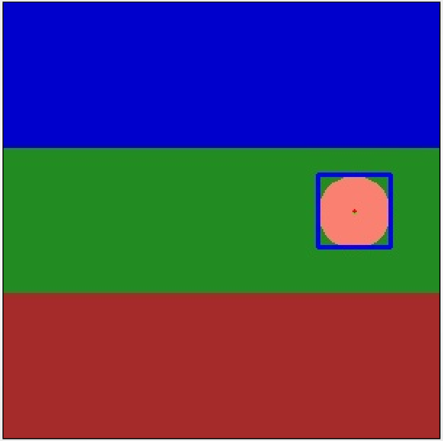
ps5-1-c-4

2a: PF Tracking a circle



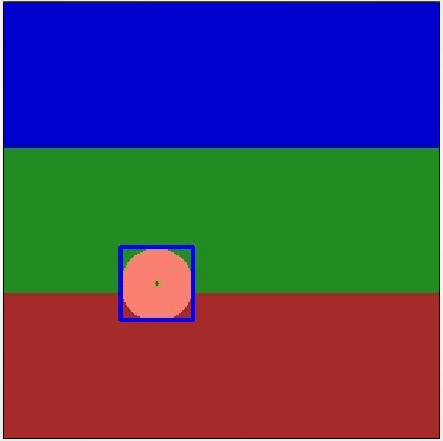
ps5-2-a-1

2a: PF Tracking a circle (cont.)



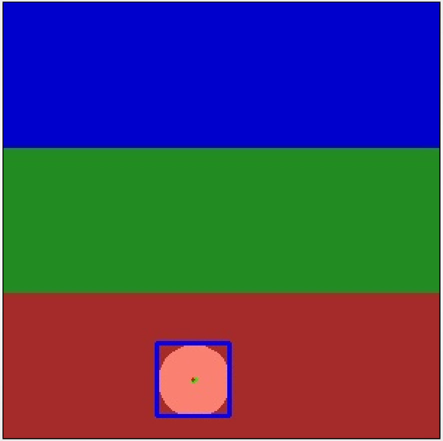
ps5-2-a-2

2a: PF Tracking a circle (cont.)



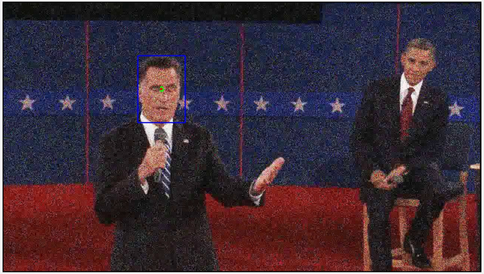
ps5-2-a-3

2a: PF Tracking a circle (cont.)



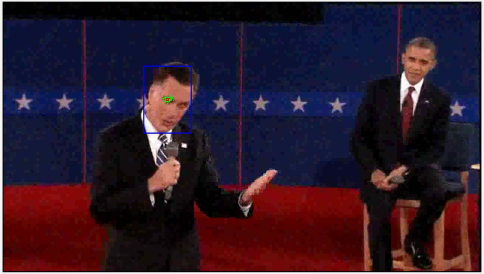
ps5-2-a-4

2b: PF Tracking noisy video



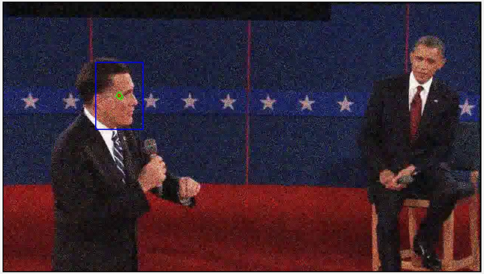
ps5-2-b-1

2b: PF Tracking noisy video (cont.)



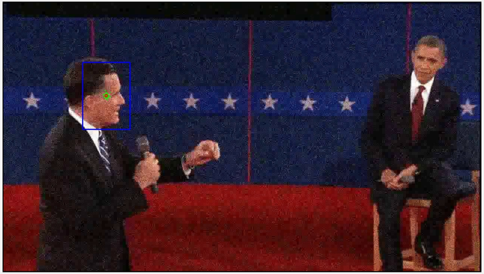
ps5-2-b-2

2b: PF Tracking noisy video (cont.)



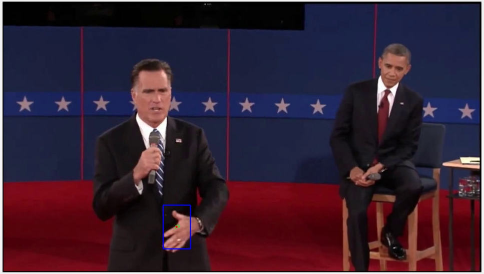
ps5-2-b-3

2b: PF Tracking noisy video (cont.)



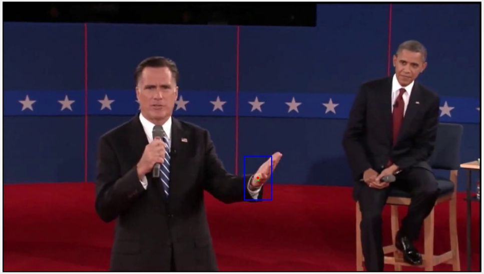
ps5-2-b-4

3a: PF Changes in Appearance



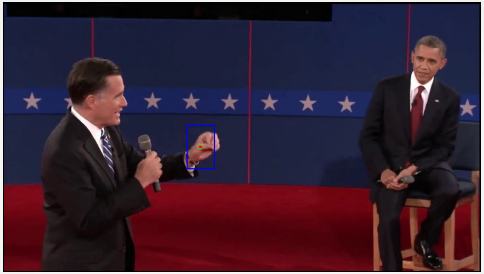
ps5-3-a-1

3a: PF Changes in Appearance (cont)



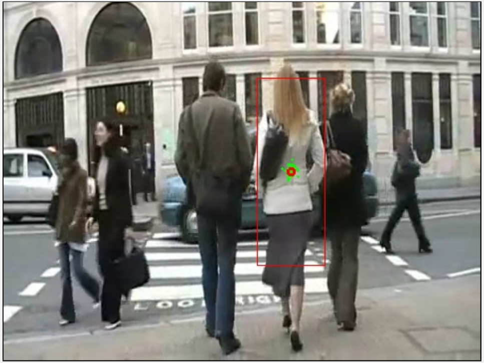
ps5-3-a-2

3a: PF Changes in Appearance (cont)



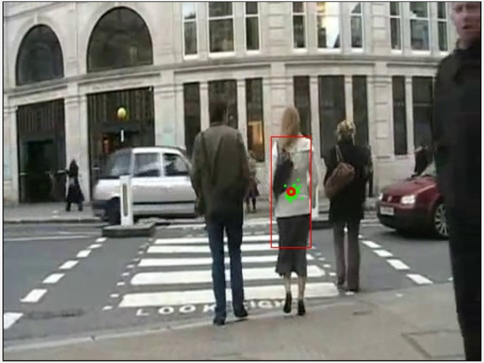
ps5-3-a-3

4a: PF Occlusions



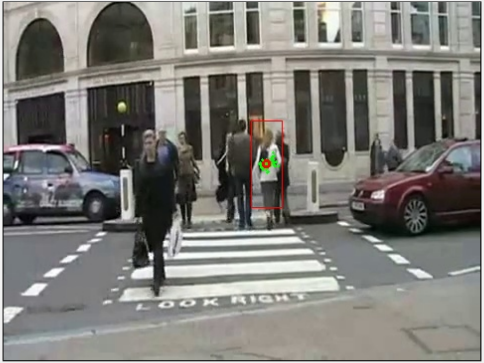
ps5-4-a-1

4a: PF Occlusions (cont.)



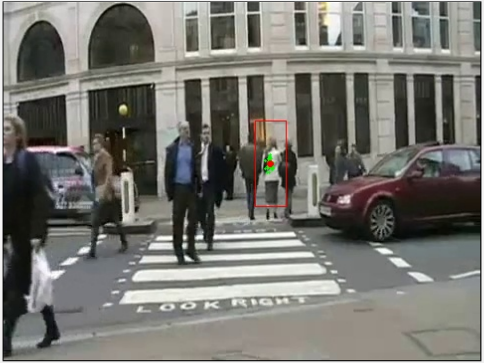
ps5-4-a-2

4a: PF Occlusions (cont.)



ps5-4-a-3

4a: PF Occlusions (cont.)



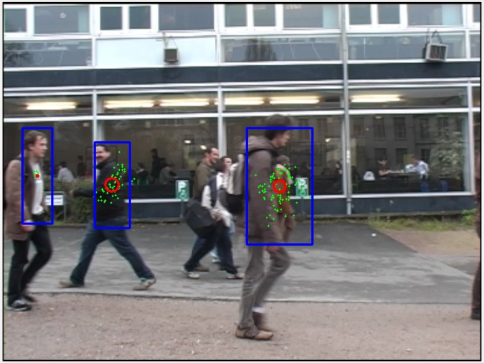
ps5-4-a-4

4: Text response

Describe what you did. How did you modify the Particle Filter class to continue tracking after occlusions?

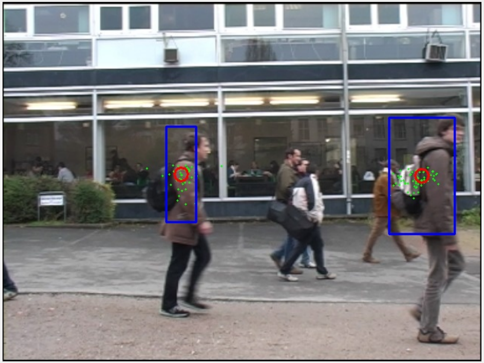
To start off, I noticed that the target being tracked moves slower in the x, y plane than in the z (scale) direction. I used a low sigma value for the dynamics to mimic this planar velocity.
I also noticed that during occlusions, the similarity score dropped significantly. I used the similarity score as my occlusion metric. A mean score below 0.00001 was considered an occlusion, and re-weighting and resampling was omitted. This would continue until the occlusion scenario had passed., at which point the mean would rise above the occlusion threshold and resampling would be resumed. The dynamics sigma was large enough to match the target again and resume tracking within a few frames.

5: Tracking multiple targets



ps5-5-a-1

5: Tracking multiple targets (cont.)



ps5-5-a-2

5: Tracking multiple targets (cont.)

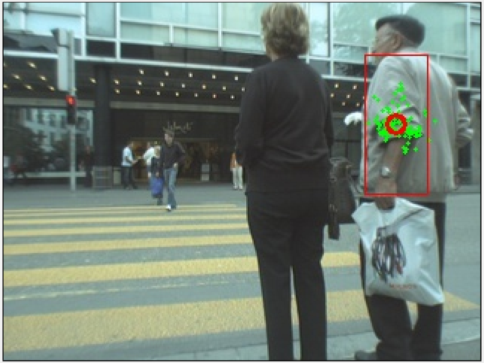
ps5-5-a-3

5: Text response

Describe what you did. How different it was to use a KF vs PF? Which one worked best and why? Include details about any modifications you had to apply to handle multiple targets.

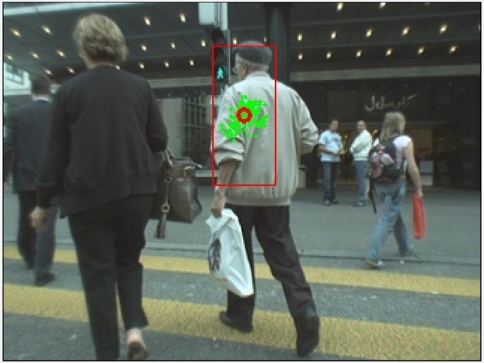
I ended up using the Appearance-Particle-Filter for this problem. The trick was to use a template that had enough unique features with the least amount of change between frames. The torso and face sections worked quite well for this, a backpack was a plus. Including the legs resulted in the template changing too much from between frames which caused the particle filter to jump around a lot. I used a low alpha value to take into consideration the effect of the changing background, while keeping the main features of the person we were tracking relatively consistent. I used a larger sigma value for the dynamics as the targets were moving pretty quickly. This also helped during occlusions, as the particles would disperse, but the large dynamics helped in regaining the target pretty quickly, so I didn't need to do anything special to deal with occlusions.
I cloned and modified run_particle_filter into run_particle_filter_multi_target that could take a list of targets. This was needed to initialize and run one particle filter for each of the targets.
I also tried the KalmanFilter but ran into issues with the underlying cv2.matchTemplate in run_kalman_filter that provided measurement values. Investigating what measurements were being returned showed large errors as match template returned coordinates for the wrong person in the frame. I'm sure with more tuning KalmanFilter would also have given reasonable results, specially since we were dealing with tracking constant velocity targets - however, particle filtering seemed more appealing as it required less effort to implement.

6: Challenge Problem



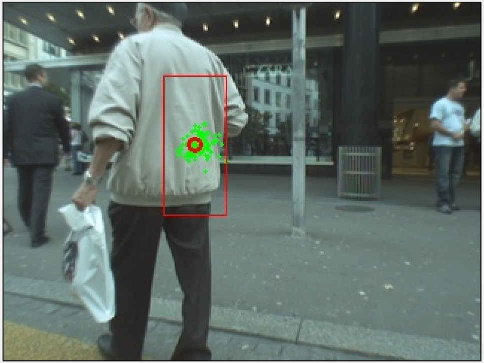
ps5-6-a-1

6: Challenge Problem (cont.)



ps5-6-a-2

6: Challenge Problem (cont.)



ps5-6-a-3

6: Challenge Problem Text response

Describe what you did. Did this task present any additional challenges compared to the previous sections? Include details about any modifications you had to apply.

The target moves closer and further to the camera throughout the frame sequence. This prompted me to use MDParticleFilter so that scale could be incorporated. I used some of the tricks learned in part 5, mainly to use a smaller template zone, specifically this target's torso area. The camera also stays to the the left of the target for the most part so I was able to get away with using a fixed template rather than using an alpha value for changing appearance.

I didn't need to modify run_particle_filter for this part, it worked as is.