# Computer Vision
# Fall 2023
# Problem Set #5

Trevor Davidson
tdavidson30@gatech.edu

# 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?**

The primary change to the original Particle Filter methodology was removing the requirement to resample at each iteration. To determine if resampling was necessary, I used the MSE from the best particle and compared it against the MSE from the previous frame using a scale factor. If the best particle's error was too high relative to the previous frame, this indicates the target has been occluded, and resampling is ignored. By continuing to add noise to the particles, this allows the particles to spread out more and potentially reacquire the correct match.

The other major change I made was to add the tracking template scale to the particles (with a modified sigmaDyn parameter to account for the difference in pixel/percentage). This allowed the scale to adjust as the woman walked from the image foreground to the background, improving the algorithm's handling of occlusions. Accordingly, to keep the window from getting too small during occlusion, I added a minimum scale parameter to keep the template from shrinking too far and did not adjust template size when the system detected an occlusion.

# 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.

To track multiple targets, I created multiple kalman filters. Templates were initialized from the first frame the individual was fully visible (to best match the images in the instructions) and the Kalman filter was then updated at each frame the person was in frame. I used the matching algorithm from ps5.utils.runKalmanFilter() to get the measurements from each scene.
As visible in the images above, the Kalman filter performed well when the template individuals were present and unobscured. However, the base KF struggled when individuals became obstructed behind other people (such as the tracked individual in the black jacket). Although I did not implement a particle filter for this assignment, a PF would likely have similarly struggled during the occlusion. With proper dynamics model tuning, a particle filter would likely have been more effective than a base Kalman model.

In order to overcome the occlusion, both the KF and PF would have needed to estimate the x-velocity of the individuals and ignore the measurements when the error is above a certain threshold.

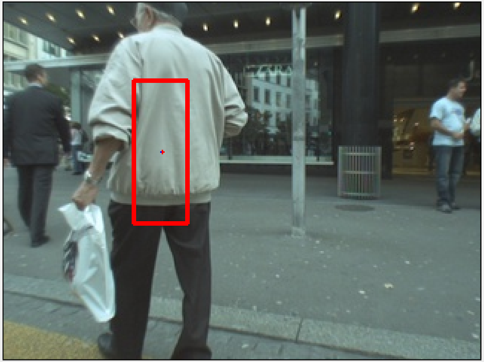
# 6: Detect Pedestrians from a moving camera

**ps5-6-a-1**

# 6: Detect Pedestrians from a moving camera (cont.)

**ps5-6-a-2**

# 6: Detect Pedestrians from a moving camera (cont.)



**ps5-6-a-3**

# 6: Detect Pedestrians from a moving camera

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.

This assignment added multiple layers to the previous images. This challenge is encountered immediately, when the person of interest changes orientation: in frame 1 he stands sideways relative to the camera, before turning to walk away from the camera. This makes the original tracking template much less effective.
The movement of the camera also makes tracking the man increasingly difficult, as he rapidly becomes much larger than the image picture, resulting in parts of the template becoming unusable.
The natural, uncontrolled scene also resulted in frequent lighting changes, adjusting sensor values and sometimes resulting in rapid changes in error.
Finally, the rapid movement early in the video to the right means particles can be left behind and begin tracking other targets. This required significant tuning of the dynamics parameters, particularly the template scaling parameter.
All of these factors are also on top of frequent occlusion.
Using the MDParticleFilter class as a base, I made multiple modifications to track the man. In particular, I adjusted the dynamics model to separate x and y particle spreading. I adjusted this because the man's y position does not change much, but he makes rapid shifts to the left and right.
I also adjusted the minimum scale parameter and attempted to control the template size direction by adjusting the dynamics distribution to skew towards increasing the template size.