16-820 Advanced Computer Vision Assignment 2

Li-Wei Yang
liweiy@andrew.cmu.edu
Section A

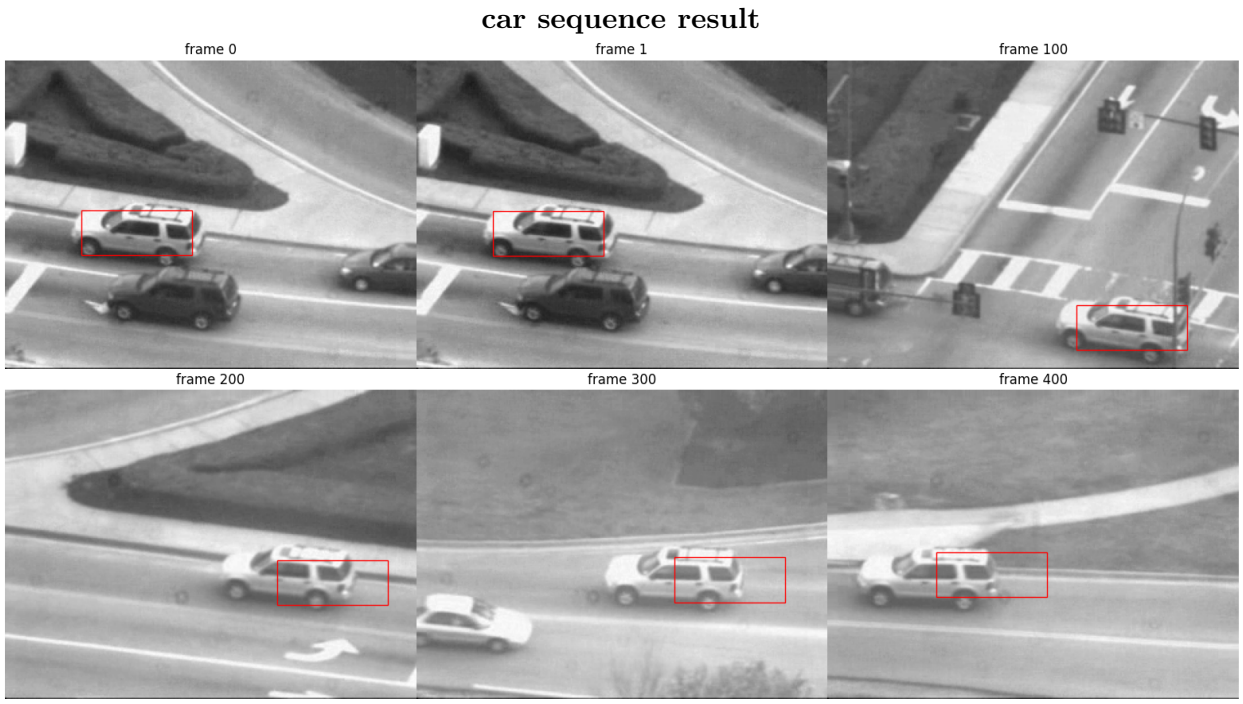
Q1.1

---

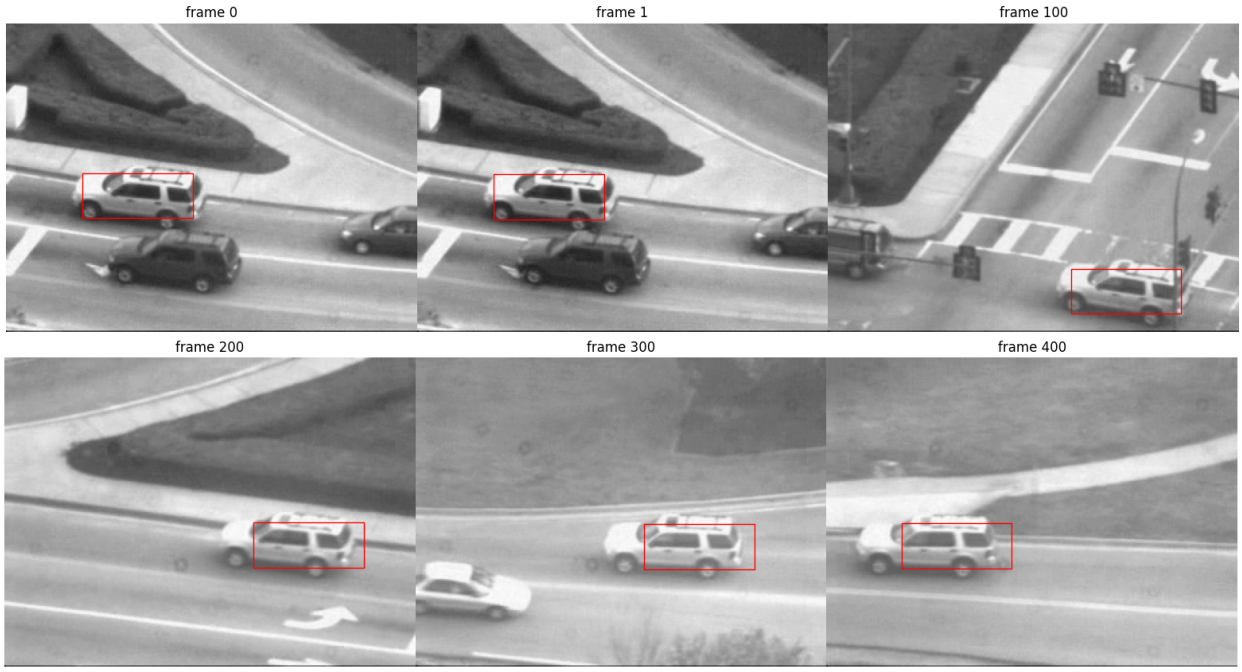
1. $\frac{\partial W(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}^T}$ is the Jacobian of warped image coordinates w.r.t warp parameters $\mathbf{p}$.
2. $\mathbf{A} = \sum_{x,y} [I_x, I_y]$ is the image gradient patch in x direction and y direction. $b = \sum_{x,y} D(x,y) = I_{t+1}(x,y) - I_t(x,y)$ is the error between the template patch and the current image patch.
3. $\mathbf{A}^T\mathbf{A}$ must be invertible so that unique solution to $\Delta\mathbf{p}$ can be found.

Q1.2

Q1.3



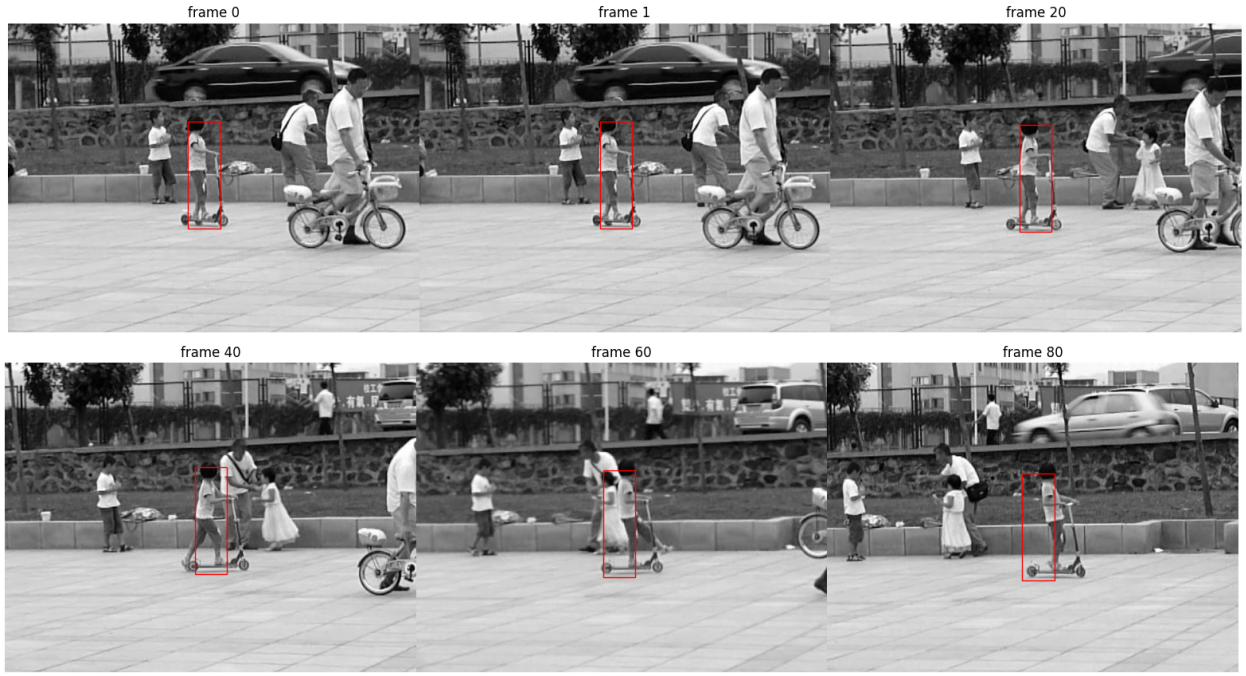
num_iters = 1e4, threshold = 0.8



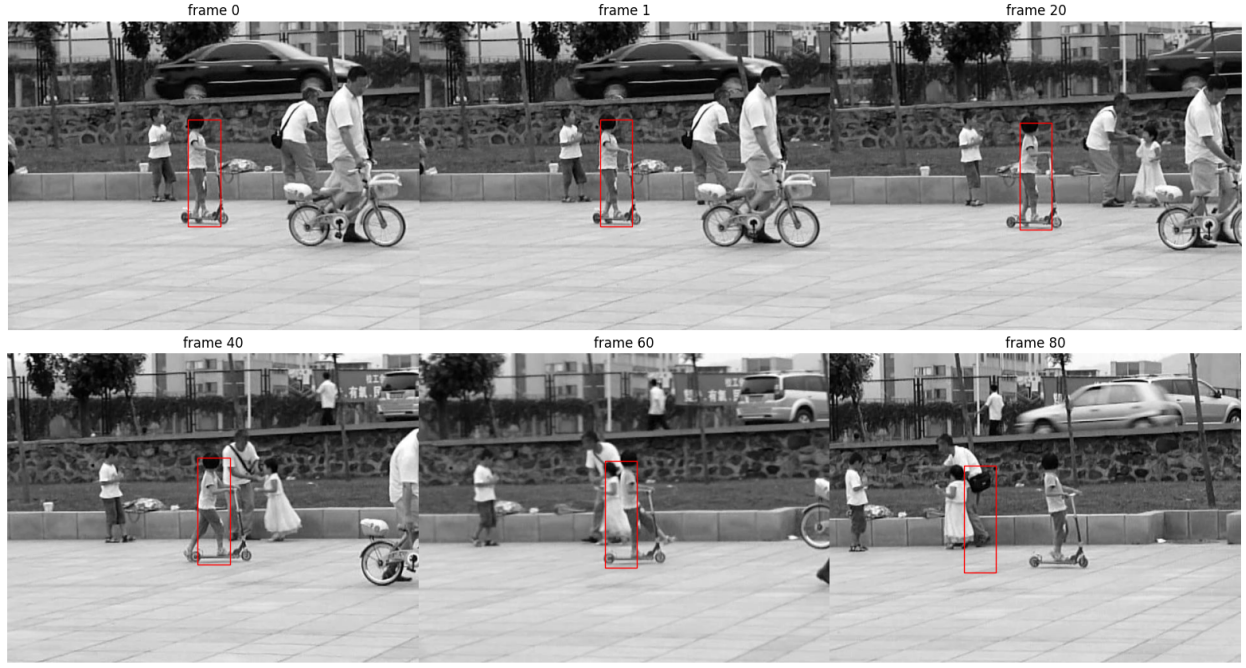
num_iters = 1e4, threshold = 1e-2

From the two settings, we can see that the threshold is crucial while tracking. Decreasing the threshold might increase accuracy if the object moves on a clean background.

girl sequence result



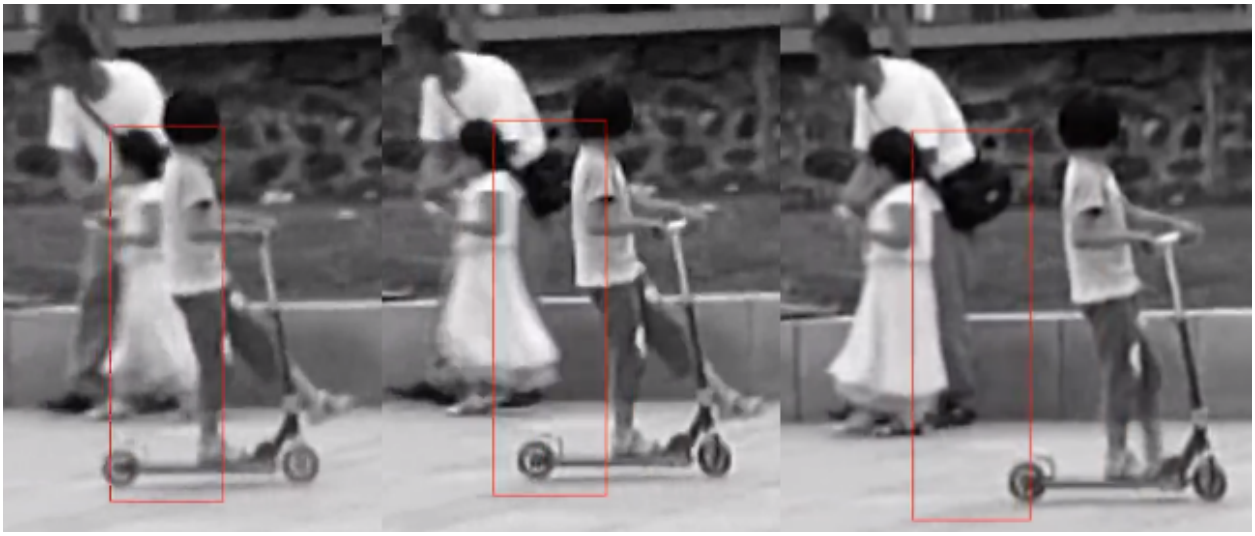
num_iters = 1e4, threshold = 1e-2



num_iters = 1e5, threshold = 1e-5

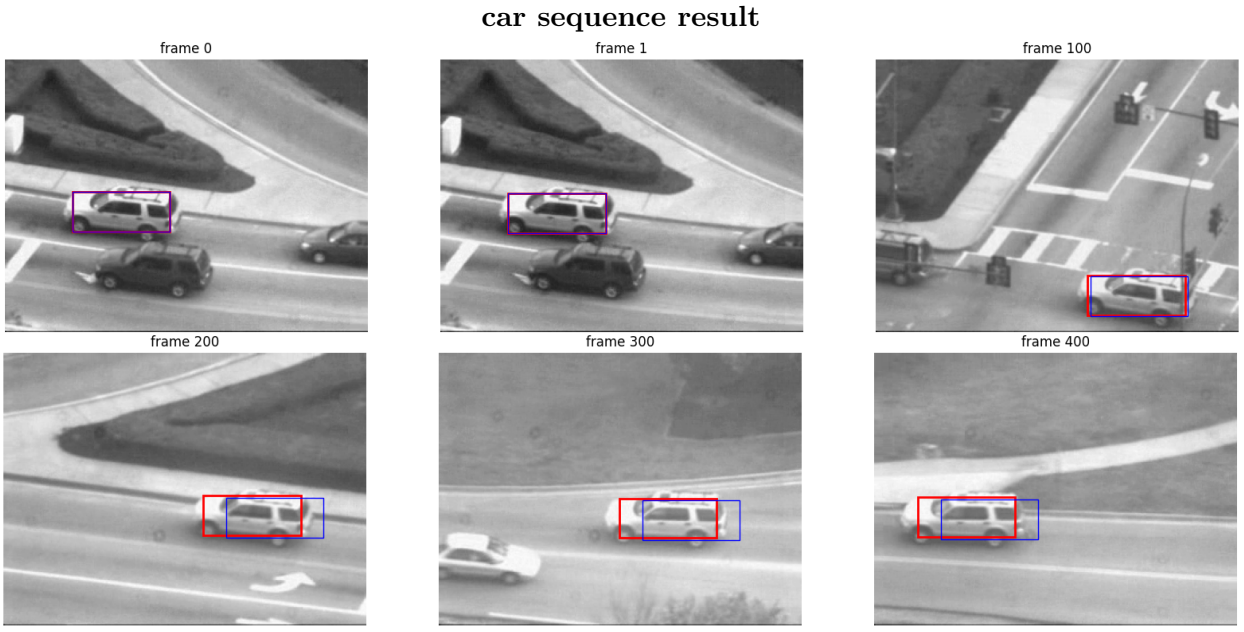
From the two settings, we could see that decreasing the threshold would not necessarily increase

accuracy if the background is cluttered. From playing the tracking video, I found the tracking failed because if the threshold is too small, we allow the rect to update more to reach minimal error. If there is another object takes up more space in the tracking box, the pattern we are tracking will change to that object. From the figure below, we can see the tracking object first changes to the other girl, then the men's pants and bag.

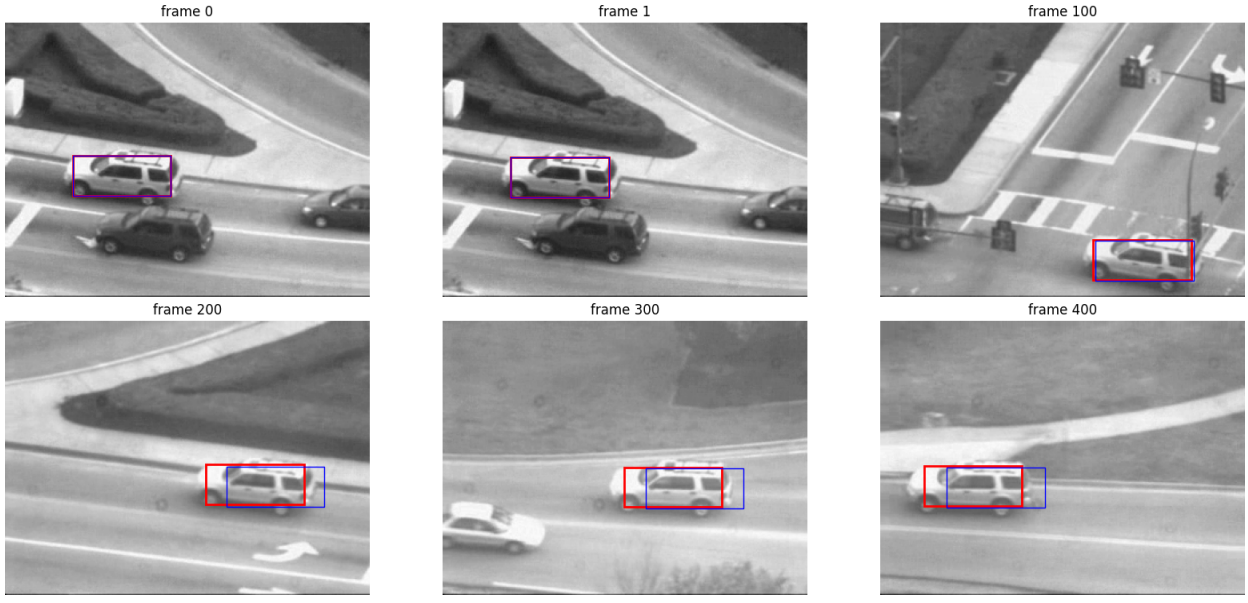


Q1.4

rect in red is the result with template correction, while rect in blue is without template correction.



num_iters = 1e4, threshold = 1e-2, template_threshold = 5

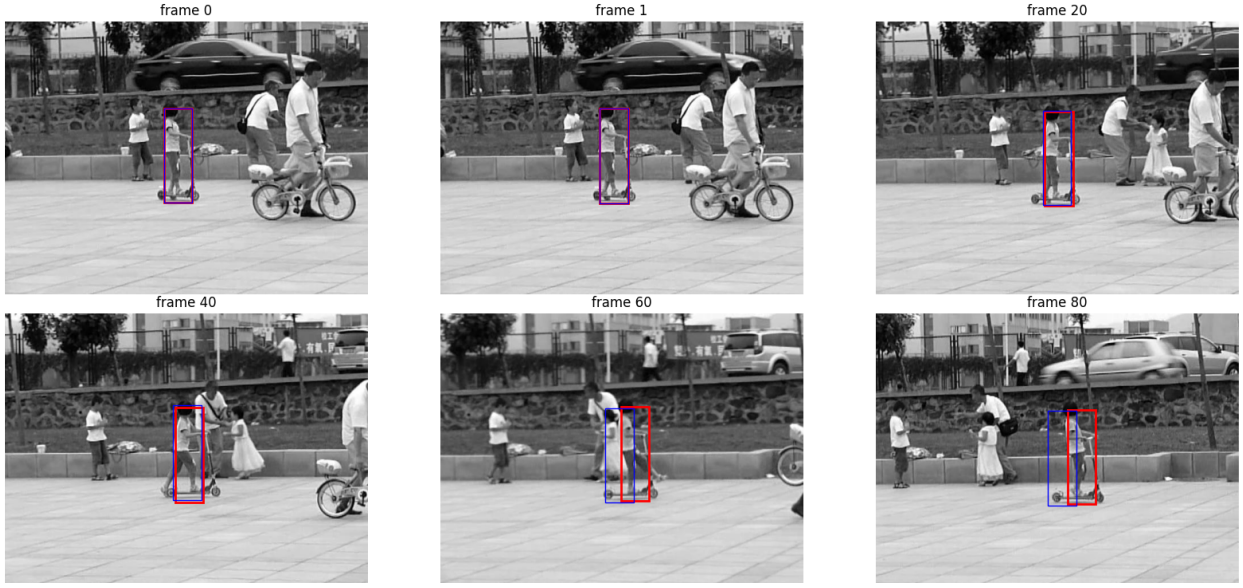


num_iters = 1e4, threshold = 1e-2, template_threshold = 0.5

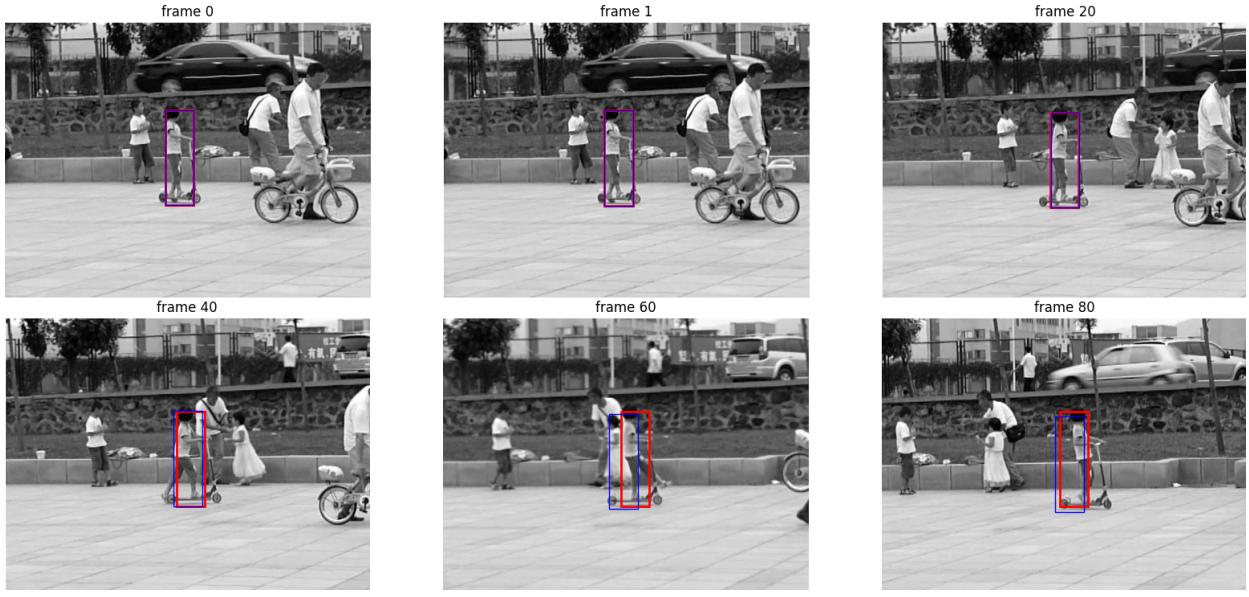
From the above two settings, we cannot see the difference, but through the print function, I found that with a smaller threshold, when the car is partially blocked by the street light, we did not

update the template. After the car passes the street light, we can track it again. Thus, template correction method may be useful when objects are temporarily blocked and then reappear.

girl sequence result



num_iters = 1e4, threshold = 1e-2, template_threshold = 5



num_iters = 1e4, threshold = 1e-2, template_threshold = 0.5

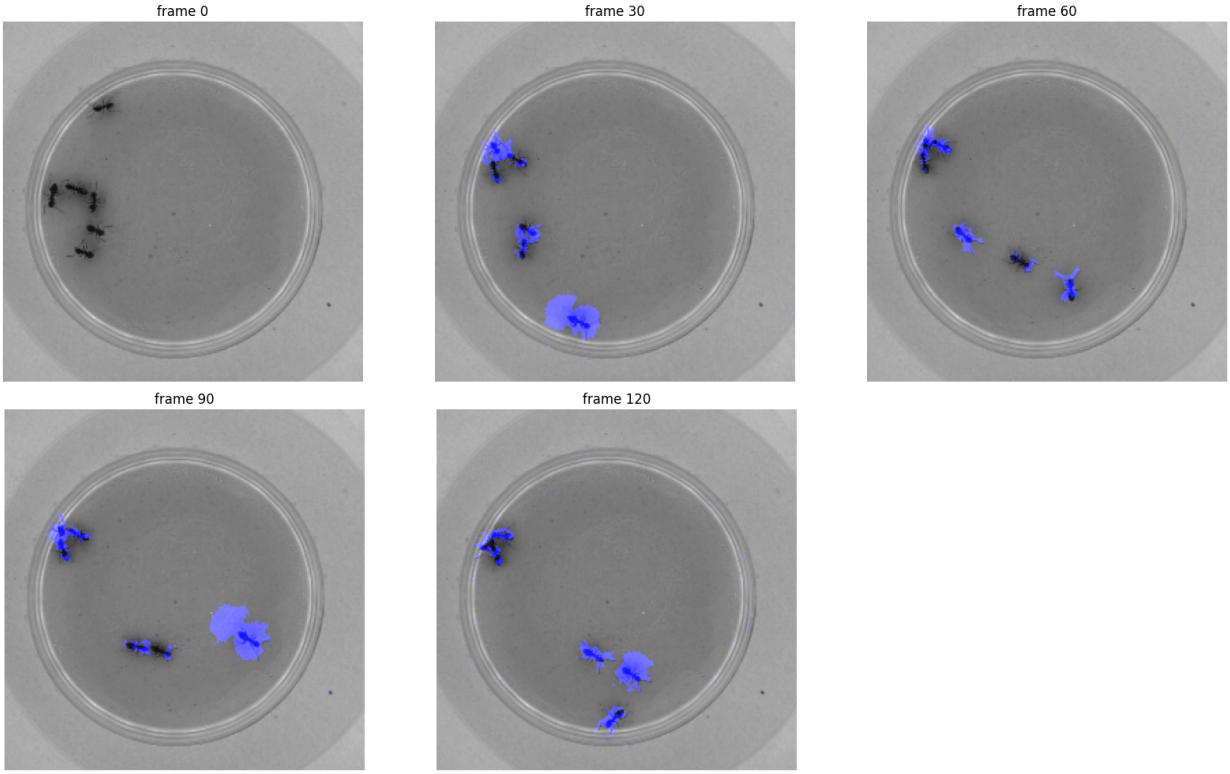
From the above two settings, we can see that with a smaller threshold, when the girl is passing through the crowd, we did not update the template. Thus, when the girl passes and changes her pose, we cannot track her again. In conclusion, template correction is more useful when the tracking object does not change its own shape or color, like a car.

Q2.1

Q2.2

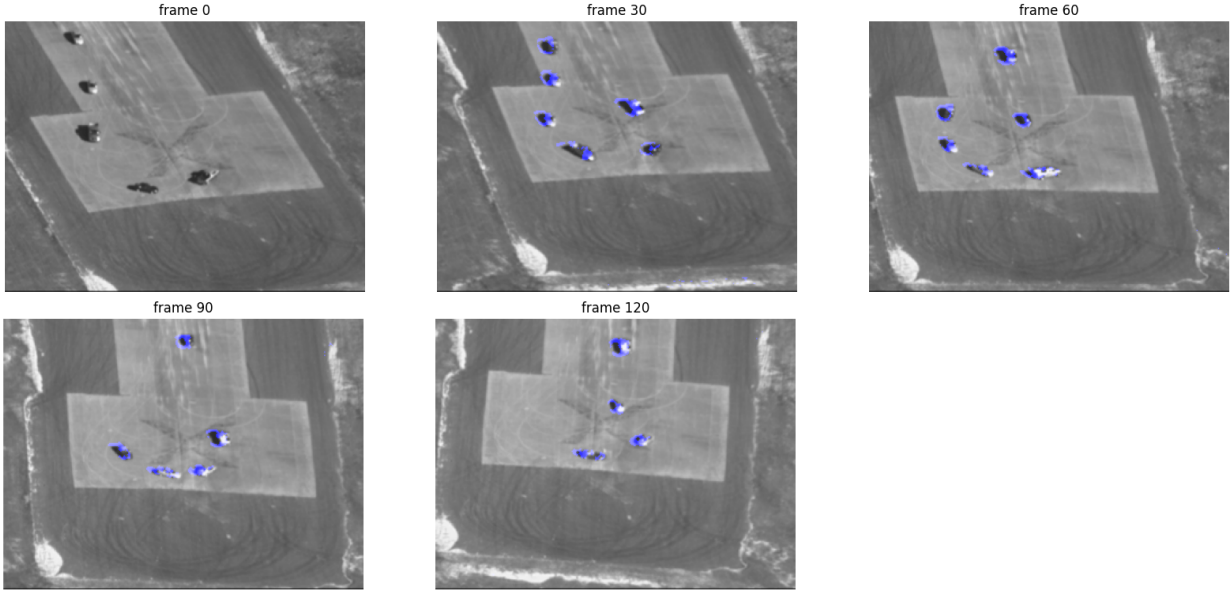
Q2.3

ant sequence result



num_iters = 1000, threshold = 0.01, tolerance = 5

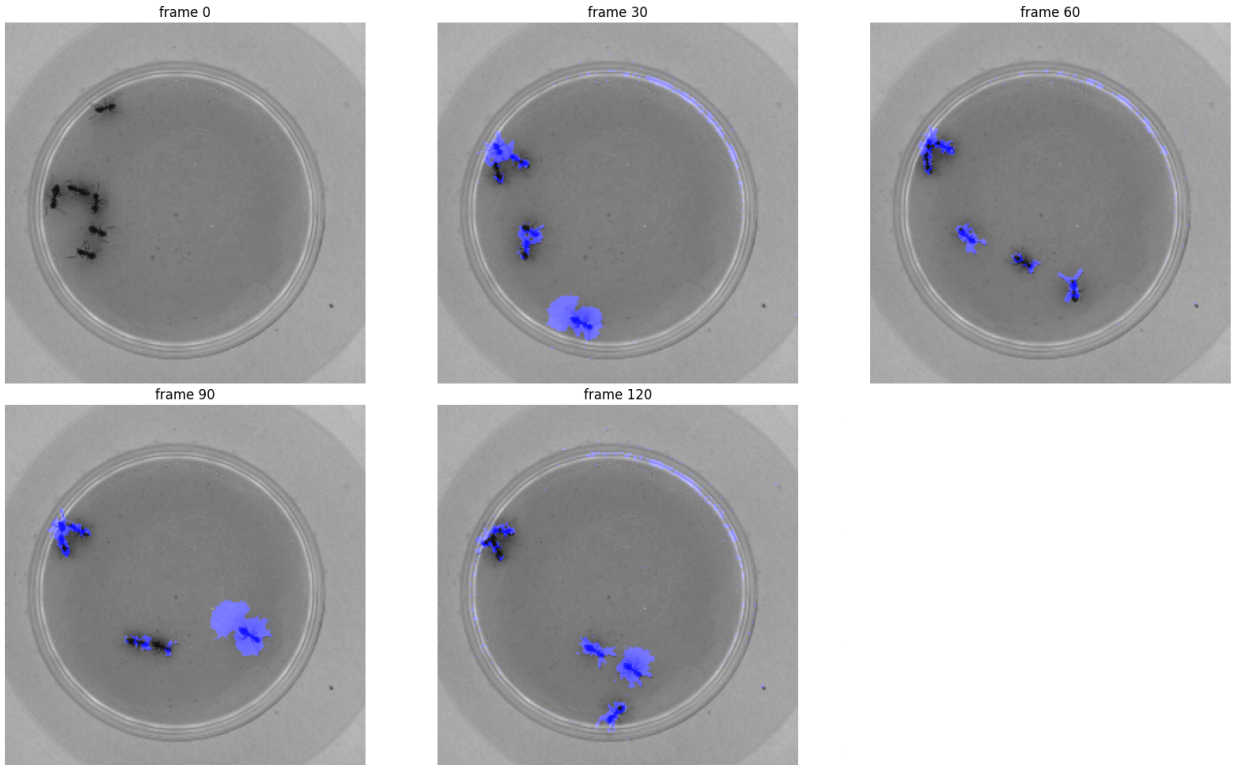
aerial sequence result



num_iters = 1000, threshold = 0.01, tolerance = 20
aerial sequence result

Although not visible in the picture, there is a thin spurious detection along the white edge in the aerial sequence. We could set another threshold for the density of detection, and If a certain area's detection density is above the threshold, then consider a valid detection to remove spurious detection.

Q3.1



num_iters = 1000, threshold = 0.01, tolerance = 5
ant sequence result with inverse composition

We could see some false detection along the edge of Petri dish. It may result from error propagation. Because we warp the image twice, the error is magnified. It could also be removed by implementing the method I mentioned in Q2.3

Q3.2

The Hessian matrix in the inverse composition method is pre-computed, so we don't have to re-compute the Hessian matrix in every iteration. Thus, inverse composition is more efficient.