Lucas-Kanade Tracking

16-720 Computer Vision - Homework 3

Q. 1.1

1. $\frac{\partial W(x;p)}{\partial p^T}$ is the Jacobian Matrix.

The Jacobian for pure translation is, $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

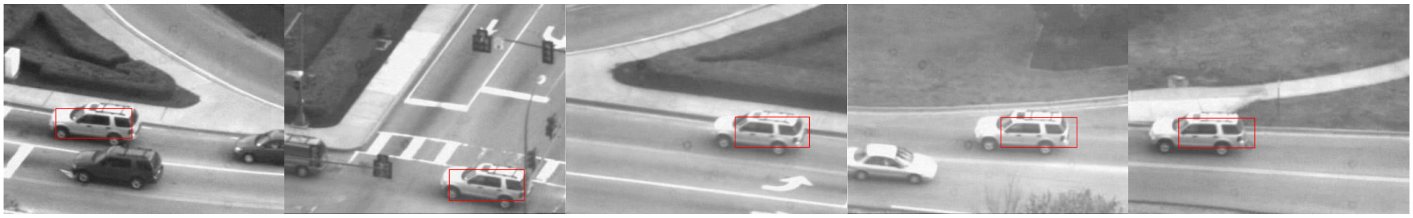
2. Matrix $\mathbf{A}$ is the Steepest Descent. It is given by $\mathbf{A} = \nabla I * \frac{\partial W(x;p)}{\partial p^T}$

where ∇I is the gradient of the image and $\frac{\partial W(x;p)}{\partial p^T}$ is the Jacobian

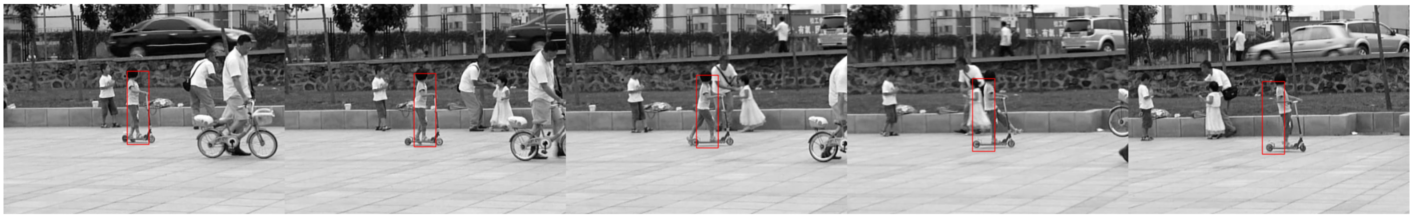
3. $\mathbf{A}^T\mathbf{A}$ represents the Hessian Matrix. ∂p is the inverse of the Hessian Matrix and hence to find a unique solution to ∂p, it must be invertibe.

Q. 1.3 Lucas Kanade Tracking

The Lucas-Kanade (LK) tracker run on the frames of ’carseq.npy’. The images below show the frames with the tracked rectangle for the frames 1, 100, 200, 300 and 400.

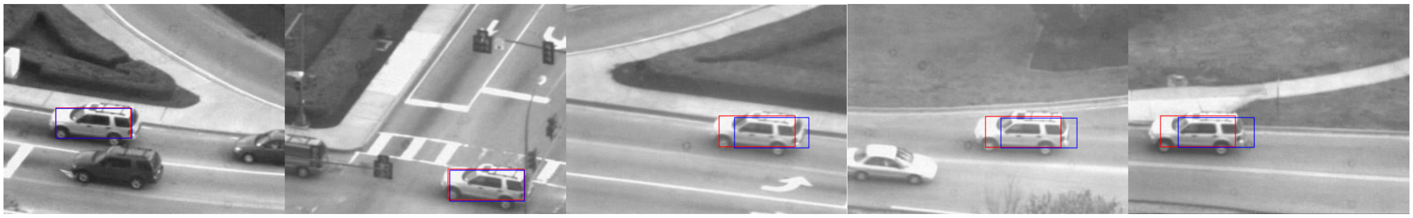


The below images show the frames 1, 20, 40, 60 and 80 of the sequence ’girlseq.npy’ with the tracked rectangle using the Lucas-kanade(LK) tracker.

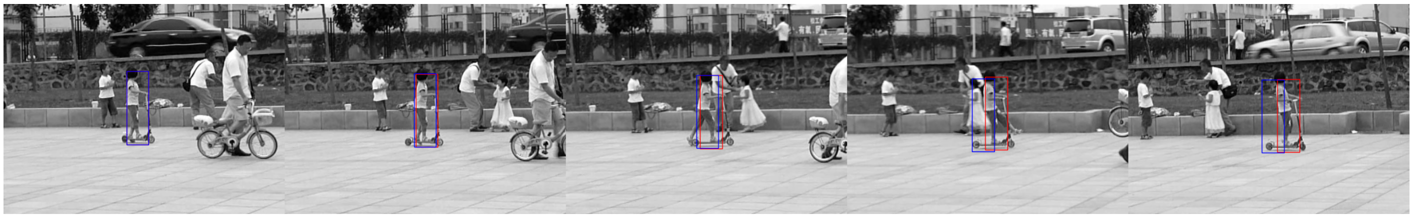


Q. 1.4 Lucas Kanade Tracking with Template Correction

Template correction for the Lucas Kanade tracker is performed and the results for the frames 1, 100, 200, 300 and 400 for 'carseq.npy' are shown below. The blue rectangle shows the tracked rectangle without template correction from Q. 1.3. The red rectangle is the tracker with template correction. The drift from the previous question has been removed. This is distinctly visible in frames 200, 300 and 400.

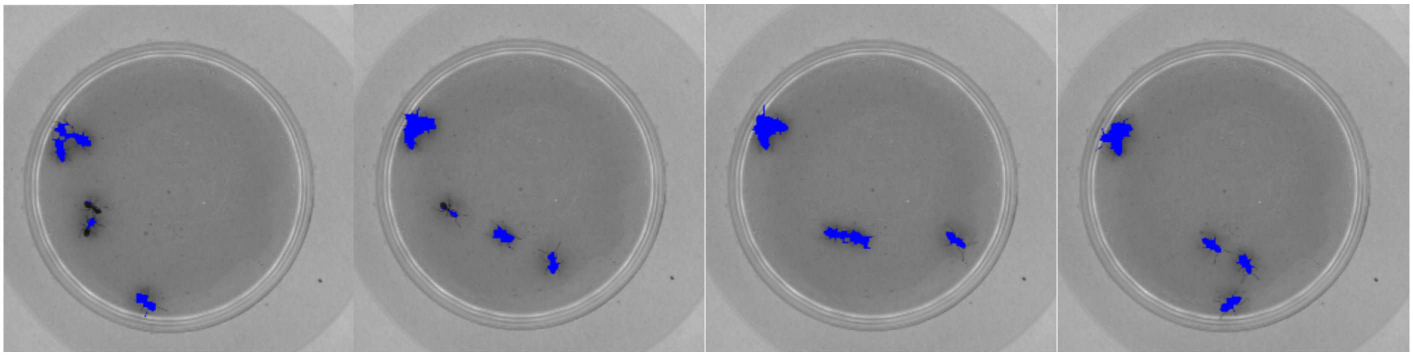


The results for the Lucas Kanade tracker with template drift correction for the sequence 'girlseq.npy' is shown below.

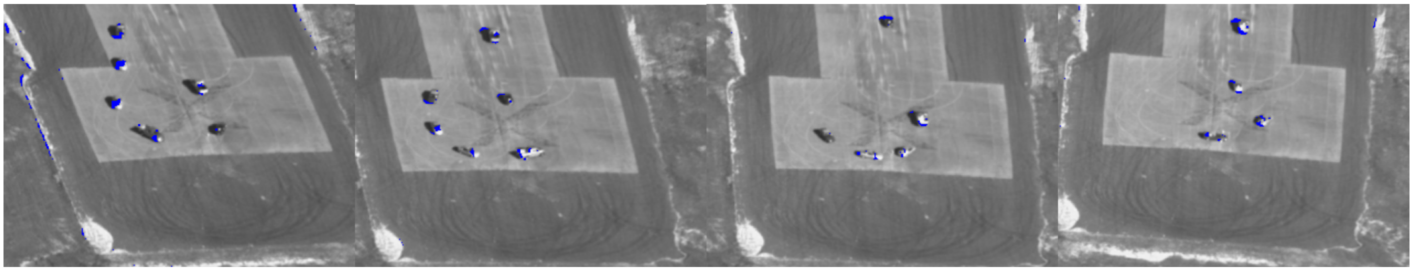


Q. 2.3 Affine Motion Subtraction

Motion detection using Lucas Kanade and Dominant Motion calculation is done for the frames for the sequence 'antseq.npy'. The images are the frames 30, 60, 90 and 120 with the blue patches indicating the motion in that frame.

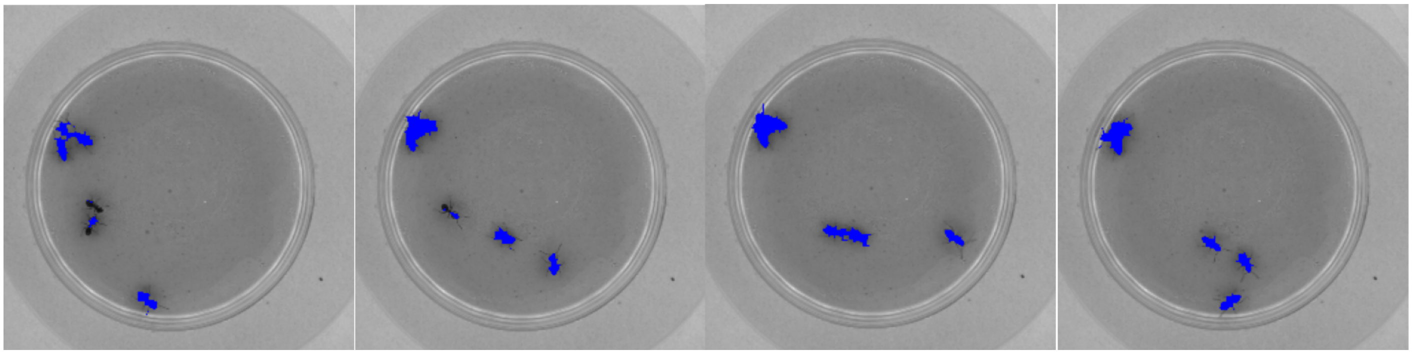


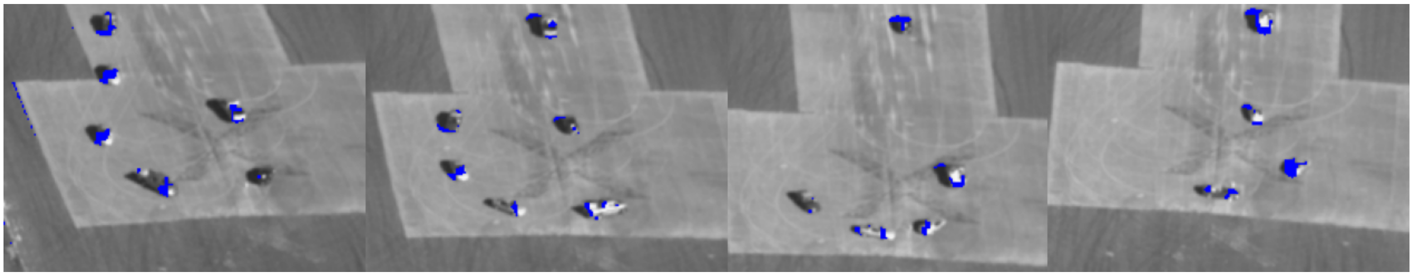
Motion detection for the sequence 'aerialseq.py'. The frames 30, 60, 90 and 120 are shown below.



Q. 3.1 Inverse Composition Affine

Motion tracking has been implemented using Inverse Composition. The results for the 'antseq.npy' and 'aerialseq.npy' have been shown below.





The algorithm is computationally faster as the Hessian Matrix calculation is done for the template image which can be performed once irrespective of the number of iterations the tracking algorithm runs. The Hessian Matrix calculation involves the gradient, Jacobian and steepest descent. Since all these calculations are performed once, the algorithm is faster.