1 Lucas-Kanade Tracking

1.1

- $\frac{\delta \mathcal{W}(x;\mathbf{p})}{\delta \mathbf{p}^T}$ is the Jacobian of the warp. It defines how the warp is changed with a change in each of the parameters of the vector $\mathbf{p}$.
where,
$$\mathbf{W}(x;p) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix} \tag{1}$$
if $\mathbf{p} = [0,0]^T$ then,
$$\frac{\delta \mathcal{W}(x;\mathbf{p})}{\delta \mathbf{p}^T} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2}$$
- Comparing equations 4 and 5 from the homework document, we get,
$$\mathbf{A} = \frac{\delta \mathcal{I}(\mathbf{x}')}{\delta \mathbf{x}'^T} \frac{\delta \mathcal{W}(x;\mathbf{p})}{\delta \mathbf{p}^T} \tag{3}$$
where,
$$\frac{\delta \mathcal{I}(\mathbf{x}')}{\delta \mathbf{x}'^T} = \begin{bmatrix} \frac{\delta \mathcal{I}_{t+1}(\mathbf{x}'_1)}{\delta \mathbf{x}'^T_1} & \frac{\delta \mathcal{I}_{t+1}(\mathbf{x}'_2)}{\delta \mathbf{x}'^T_2} & \ldots \mathbf{0}^T \\ \vdots & \ddots & \vdots \\ \mathbf{0}^T & \ldots & \frac{\delta \mathcal{I}_{t+1}(\mathbf{x}'_N)}{\delta \mathbf{x}'^T_N} \end{bmatrix} \tag{4}$$
$$\mathbf{b} = \begin{bmatrix} \mathcal{I}_{t+1}(\mathbf{x}'_1) - \mathcal{I}_t(\mathbf{x}'_1) \\ \vdots \\ \mathcal{I}_{t+1}(\mathbf{x}'_N) - \mathcal{I}_t(\mathbf{x}'_N) \end{bmatrix} \tag{5}$$
- $A^T A$ must be invertible so that a unique solution can be found for $\Delta\mathbf{p}$.

1.3: Implement Lucas-Kanade

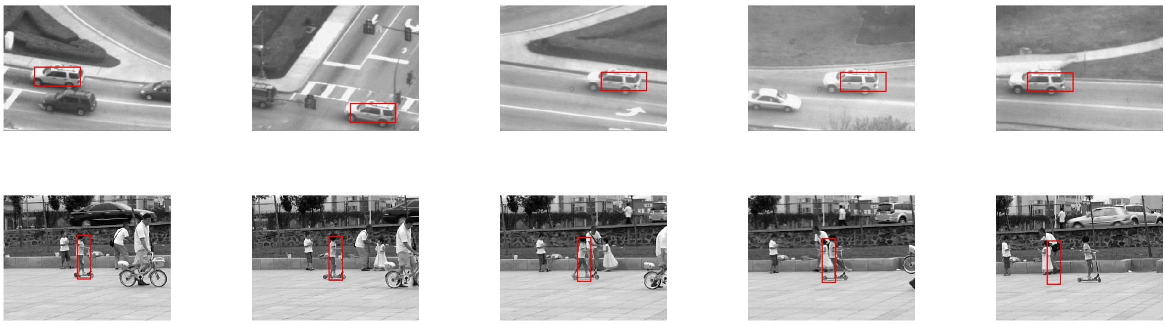

Figure 1: Lucas-Kanade Tracking without template correction. The figures for the car are from frames 1,100,200,300,400 (left to right). The figures for the girl are from frames 1,20,40,60,80 (left to right).

Parameter Tuning The parameters were tuned for both the videos. For the car sequence, it was noted that the error in tracking is initially induced when other objects(noise) are introduced in the video. In the car sequence such objects tend to be the poles, traffic signals, and other vehicles. Upon tuning, it was determined that best the most optimal values for detecting the car were as follows:

- **Threshold for Lucas-Kanade**: 0.01
- **Maximum Iterations**: 1000

For both the cases it can be concluded that error accumulates over time as template correction is not applied to the problem. For the girl sequence, it was noted that the error in tracking is initially induced when another girl enters the patch of the template being tracked. Upon tuning, it was determined that best the most optimal values for detecting the car were as follows:

- **Threshold for Lucas-Kanade**: 0.001
- **Maximum Iterations**: 1000

Overall, it was noticed that increasing the resolution of the threshold increased the execution time for Lucas-Kanade without bolstering tracking noticeably. Such behavior could be attributed to the lack of template correction in the currently implemented algorithm. A similar trend was noted for the hyperparameter for maximum iterations. For both the cases the error accumulated over time as no template correction has been applied.

1.4: Template Correction

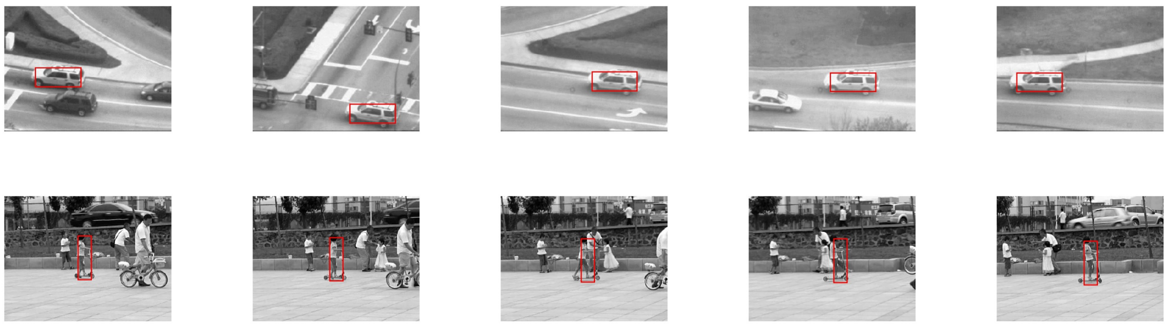

Figure 2: Lucas-Kanade Tracking with template correction. The figures for the car are from frames 1,100,200,300,400 (left to right). The figures for the girl are from frames 1,20,40,60,80 (left to right).

Parameter Tuning The parameters were tuned for both the videos. For the car sequence, it was noted that the error in tracking is minimized compared to figure 1. Upon tuning, it was determined that best the most optimal values for detecting the car were as follows:

- **Template Threshold**: 1
- **Threshold for Lucas-Kanade**: 0.01
- **Maximum Iterations**: 200

For both the cases it can be concluded that error does not accumulate over time anymore as template correction is applied that compensates for any significant drift by updating or not updating the template for the next iteration depending on the proximity of the convergence of the two gradient descents. For the girl sequence, it was noted that the error in tracking is significantly reduced compared to 1. Even when another girl enters the patch of the template being tracked, the tracker stays put on the main girl herself. Upon tuning, it was determined that best the most optimal values for detecting the car were as follows:

- **Template Threshold**: 1
- **Threshold for Lucas-Kanade**: 0.01
- **Maximum Iterations**: 1000

Overall, it was noticed that increasing the resolution of the threshold slightly, increased the execution time for Lucas-Kanade without bolstering tracking noticeably A similar trend was noted for the hyperparameter for maximum iterations. For both the cases the error accumulated over time was nearly eliminated as template correction has been applied effectively.

2 Affine Motion Subtraction

2.2: Moving Object Detection

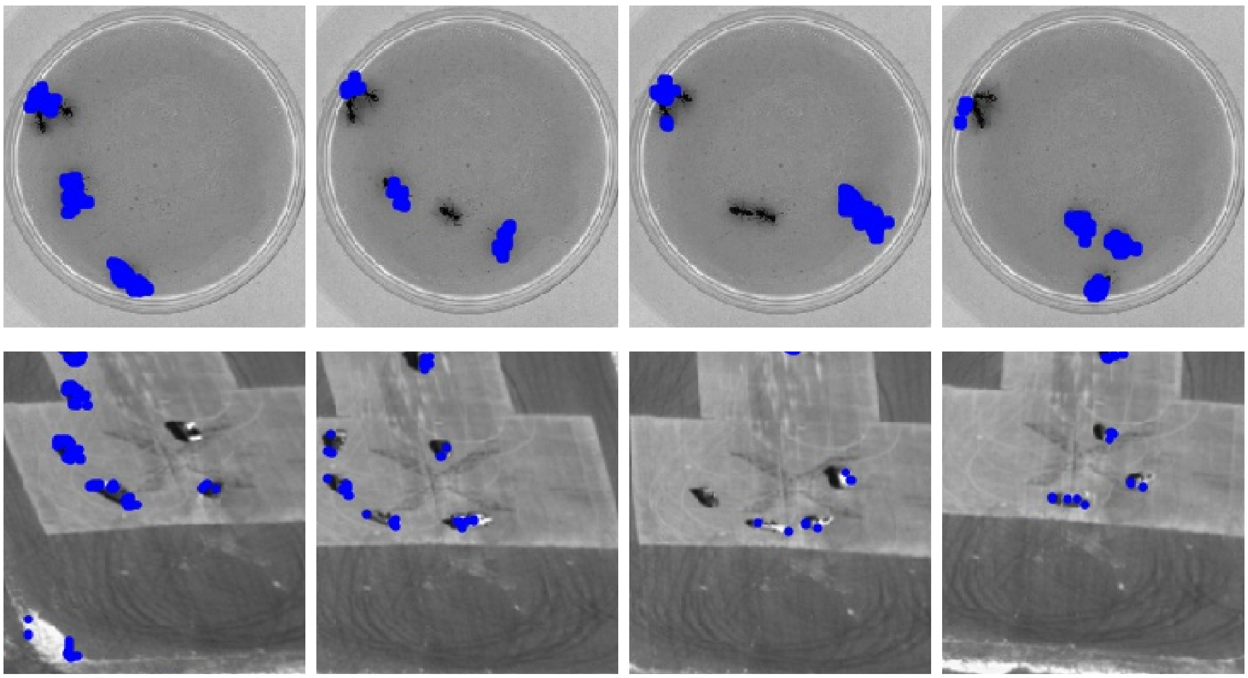

Figure 3: Lucas-Kanade Tracking with Affine transformation. The figures for the ants and cars are from frames 30, 60, 90, 120 (left to right).

3 Efficient Tracking: Inverse Composition

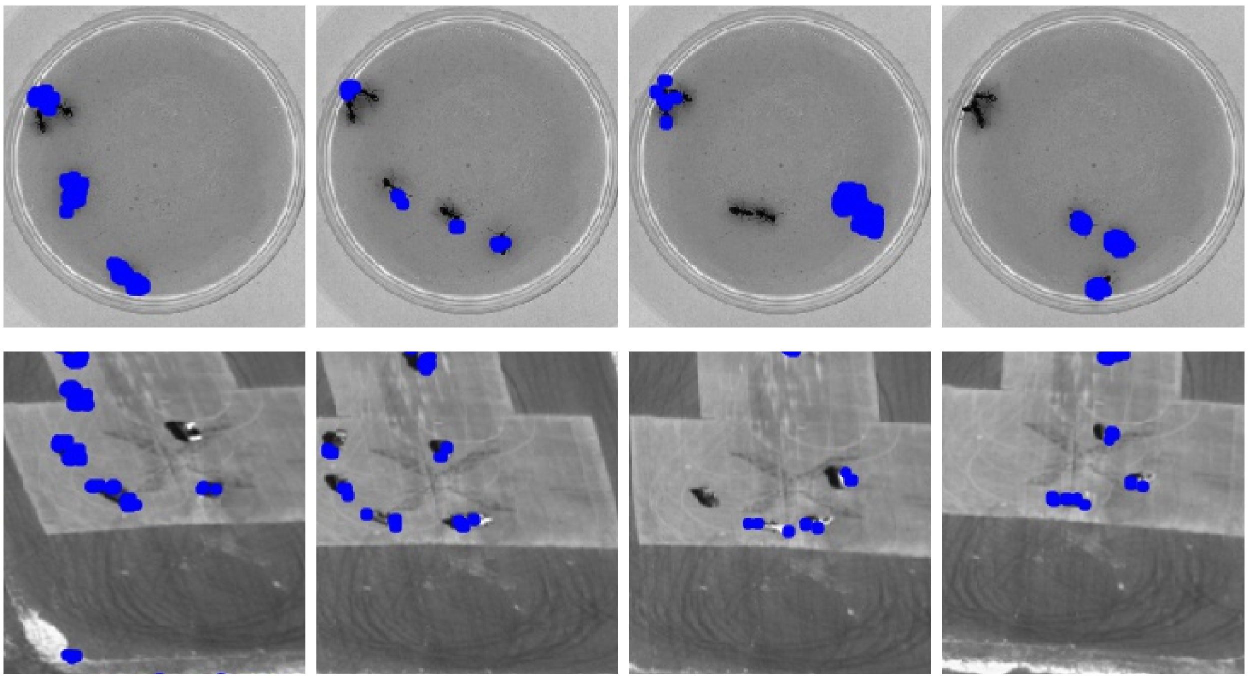

Figure 4: Lucas-Kanade Tracking with Inverse Composition. The figures for the ants and cars are from frames 30, 60, 90, 120 (left to right).

3.1: Inverse Composition

The inverse compositional method is computationally more efficient than the classical approach because unlike the conventional approach, inverse composition method does not recompute Hessain for each iteration but instead pre-computes a Hessian and treats it as a constant from there on. This sophisticated approach is the reason for the low computational cost of the inverse computation method.