16-720 : Computer Vision: Homework 3 (Spring 2020)
Lucas-Kanade Tracking

Han-Ting (Quentin) Cheng
Andrew id: hantingc

March 7, 2020

1 Lucas-Kanade Tracking

Q1.1

1. What is $\frac{\partial W(x;p)}{\partial p^T}$?
A: It is the jacobian of the warp function with respect to a change in p. In this case of pure translation, it is the 2x2 identity matrix.

$$\frac{\partial W(\mathbf{x};\mathbf{p})}{\partial \mathbf{p^T}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{1}$$

2. What is A and b?
A: Since the Jacobian is the identity matrix in this case, A can be simplified to the following Dx2 matrix where D is the number of datapoints. Matrix b is simply the stacked matrix of all the errors from each example.

$$\mathbf{A} = \begin{bmatrix} \frac{\partial I_{t+1}(\mathbf{x_1}+\mathbf{p})}{\partial(\mathbf{x_1}+\mathbf{p})^T} \frac{\partial W(\mathbf{x};\mathbf{p})}{\partial \mathbf{p^T}} \\ \vdots \\ \frac{\partial I_{t+1}(\mathbf{x_D}+\mathbf{p})}{\partial(\mathbf{x_D}+\mathbf{p})^T} \frac{\partial W(\mathbf{x};\mathbf{p})}{\partial \mathbf{p^T}} \end{bmatrix} = \begin{bmatrix} \frac{\partial I_{t+1}(\mathbf{x_1}+\mathbf{p})}{\partial(\mathbf{x_1}+\mathbf{p})^T} \\ \vdots \\ \frac{\partial I_{t+1}(\mathbf{x_D}+\mathbf{p})}{\partial(\mathbf{x_D}+\mathbf{p})^T} \end{bmatrix} = \begin{bmatrix} \frac{\partial I_{t+1}(\mathbf{x_1}+\mathbf{p})}{\partial x_1} & \frac{\partial I_{t+1}(\mathbf{x_1}+\mathbf{p})}{\partial y_1} \\ \vdots & \vdots \\ \frac{\partial I_{t+1}(\mathbf{x_D}+\mathbf{p})}{\partial x_D} & \frac{\partial I_{t+1}(\mathbf{x_D}+\mathbf{p})}{\partial y_D} \end{bmatrix} \tag{2}$$

$$\mathbf{b} = \begin{bmatrix} I_t(\mathbf{x_1}) - I_{t+1}(\mathbf{x_1}+\mathbf{p}) \\ \vdots \\ I_t(\mathbf{x_D}) - I_{t+1}(\mathbf{x_D}+\mathbf{p}) \end{bmatrix} \tag{3}$$

3. What conditions must $\mathbf{A^T A}$ meet so that a unique solution to $\Delta\mathbf{p}$ can be found?
A: $\mathbf{A^T A}$ needs to have rank equal to the number of variables in p, which in this case is 2 since it is only translation in x and y.

Q1.2

Included in LucasKanade.py

Q1.3

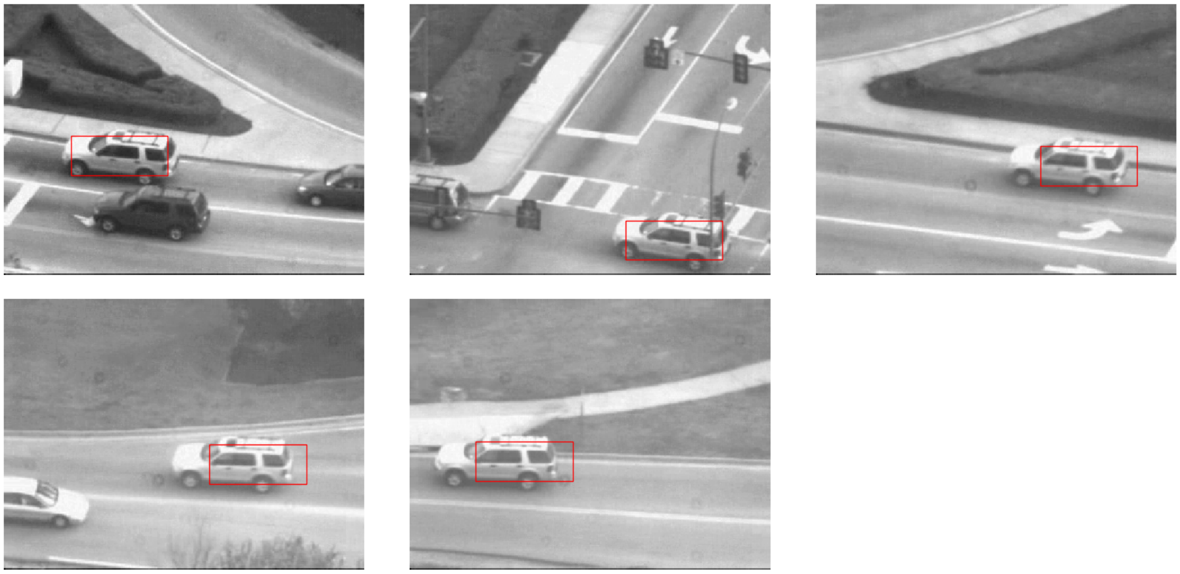

Figure 1: Car Sequence Tracking with Lucas-Kanade at frames 1, 100, 200, 300, 400

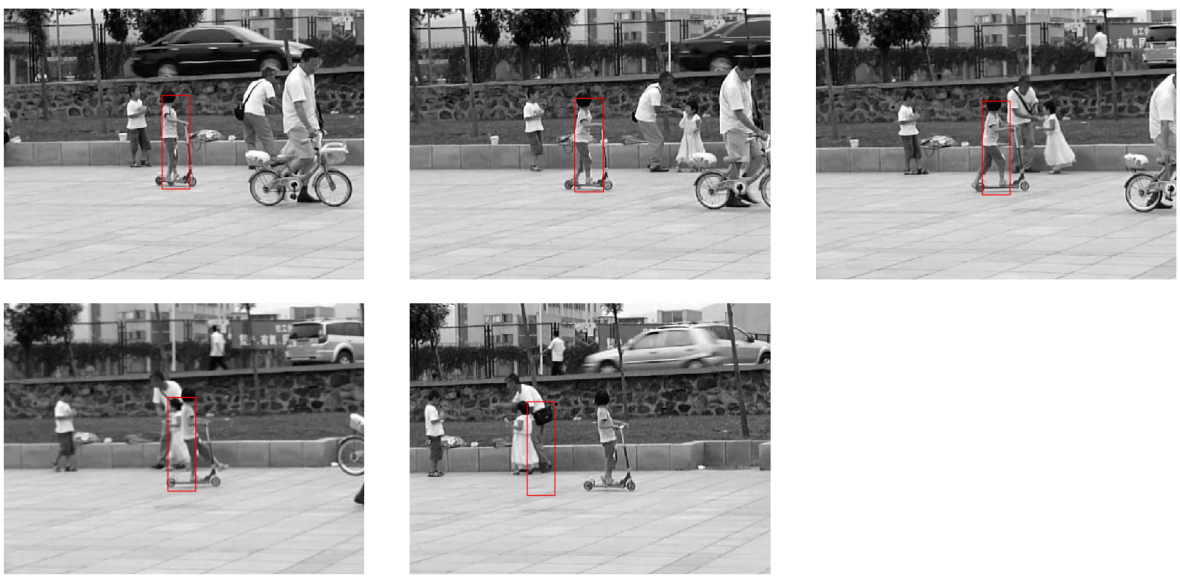

Figure 2: Girl Sequence Tracking with Lucas-Kanade at frames 1, 20, 40, 60, 80

Q1.4

Figure 3: Car Sequence with Template Correction at frames 1, 100, 200, 300, 400

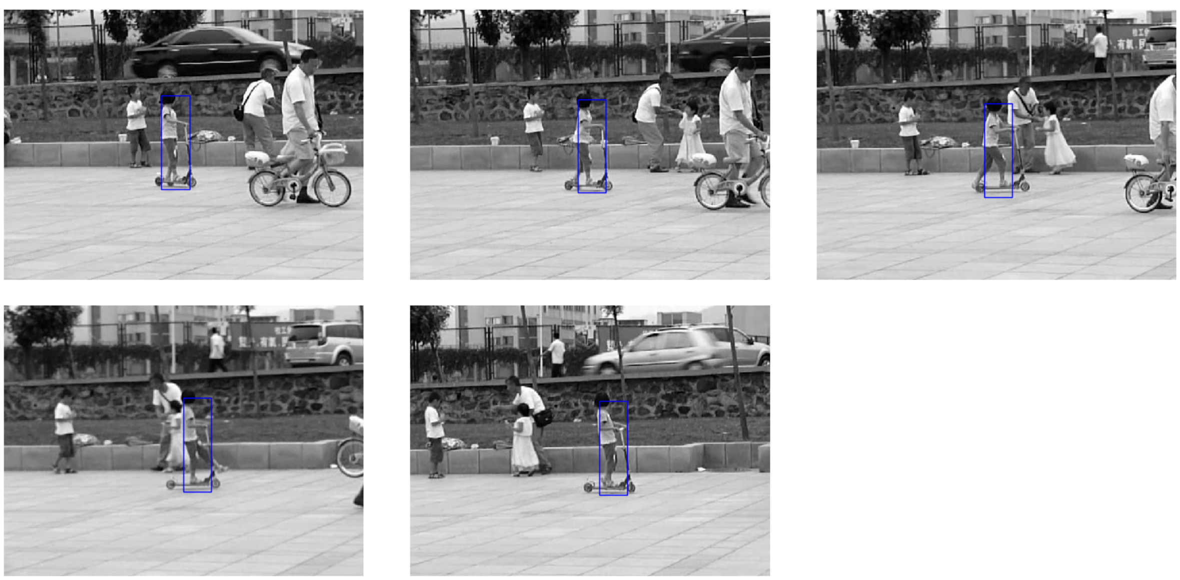

Figure 4: Girl Sequence with Template Correction at frames 1, 20, 40, 60, 80

2 Affine Motion Subtraction

Q2.3

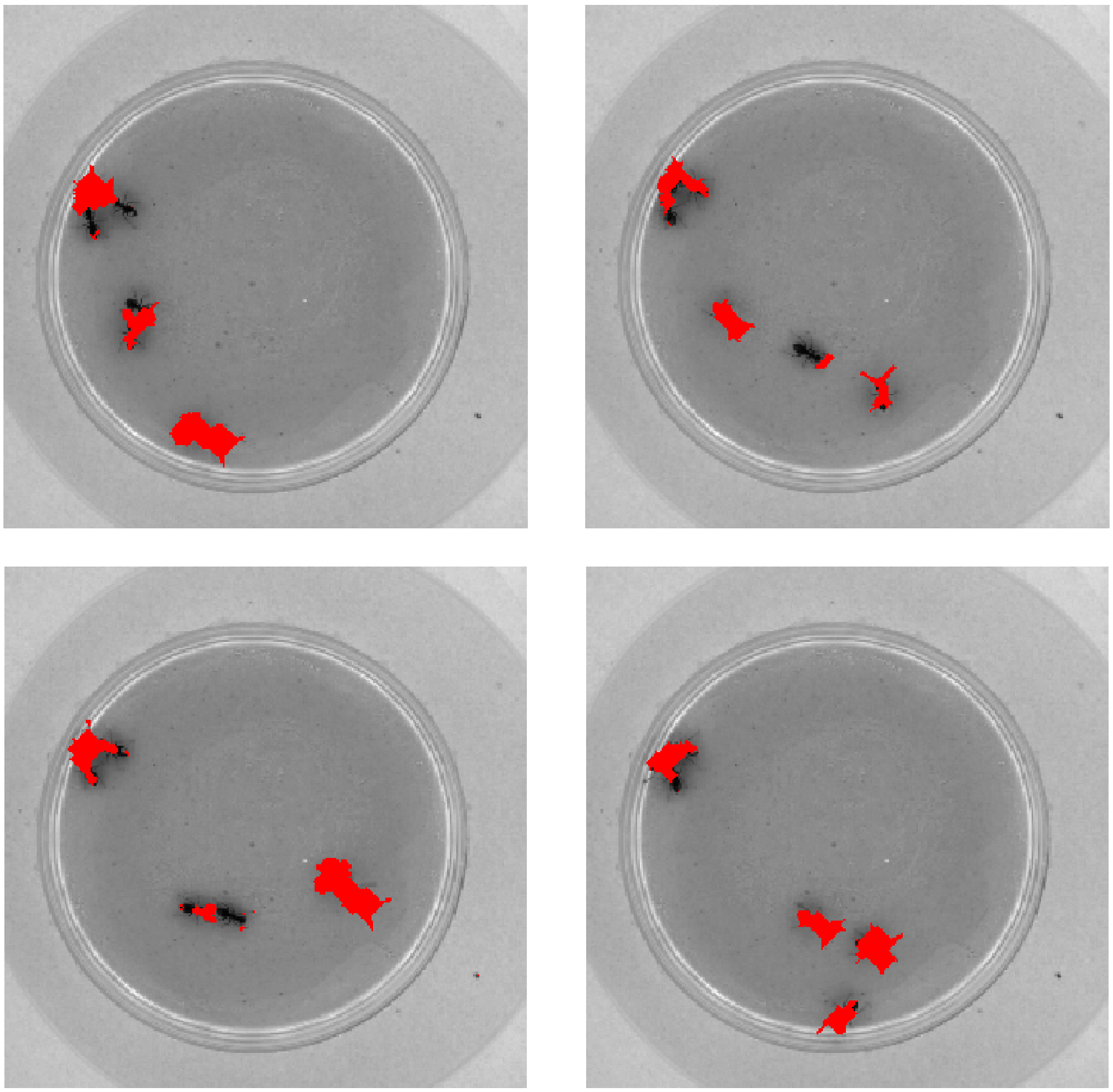

Figure 5: Ant Sequence Tracking with Lucas-Kanade-Affine at frames 30(top left), 60(top right), 90(bottom left), 120(bottom right)

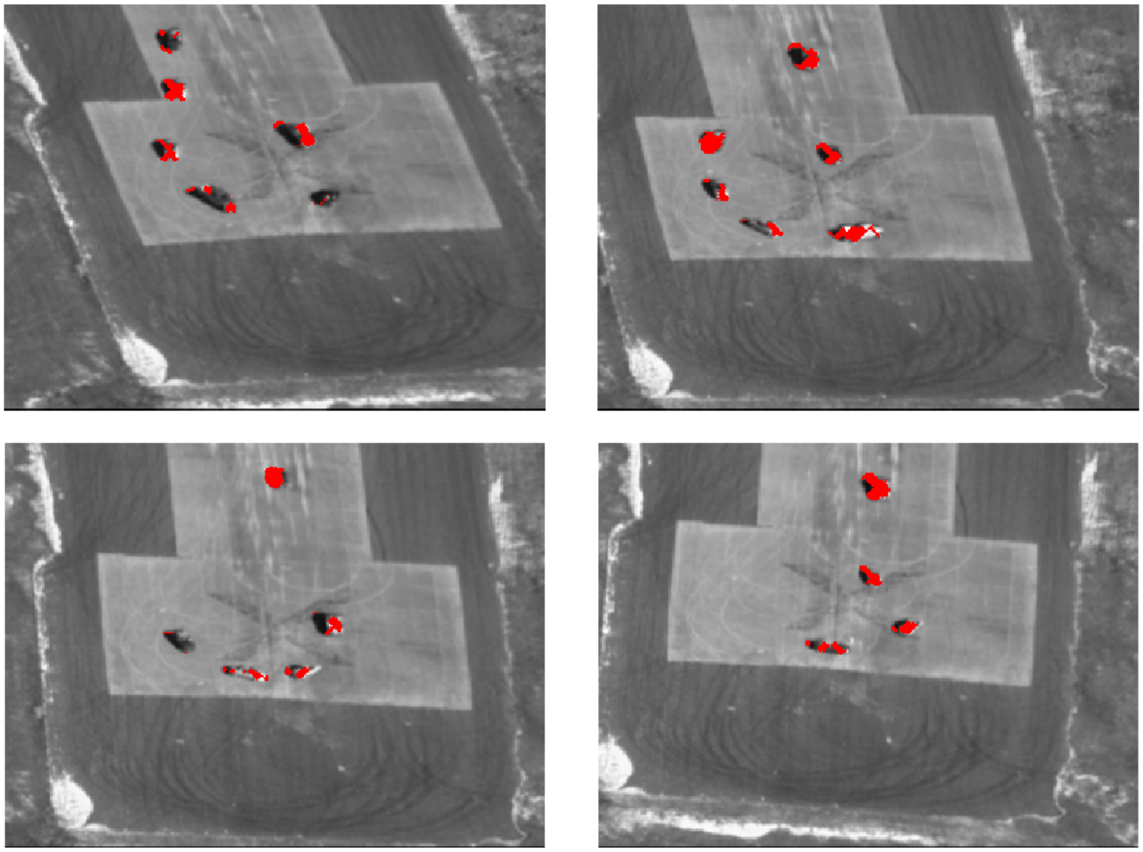

Figure 6: Aerial Sequence Tracking with Lucas-Kanade-Affine at frames 30(top left), 60(top right), 90(bottom left), 120(bottom right)

3 Efficient Tracking

Q3.1 Inverse Composition

The inverse compositional approach is more efficient because

- The Jacobian is now with respect to p=0 at every iteration so only needs to be calculated once for all iterations. Although in this case, since we have simple affine it turns out to be the same.
- The A matrix and Hessian are also calculated only once for all iterations, as we're moving the target back instead of going to the target every time. This could be done by inverting the roles of I and T cleverly, and updating warp matrix M differently.