16-720 Homework 3: Lucas-Kanade Tracking

Austin Windham

October 24, 2023

1 Lucas-Kanade Tracking

Q 1.1

What is $\frac{\partial \mathcal{W}(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}^T}$?

From lecture we know that a warp function is given by the expression below:

$$\mathcal{W}(\mathbf{x};\mathbf{p}) = \begin{bmatrix} \mathcal{W}_x \\ \mathcal{W}_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & p_1 \\ 0 & 1 & p_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix} \tag{1}$$

Taking the derivatives then gives us the results below.

$$\frac{\partial \mathcal{W}(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}^T} = \begin{bmatrix} \frac{\partial \mathcal{W}_x(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}_1^T} & \frac{\partial \mathcal{W}_x(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}_2^T} \\ \frac{\partial \mathcal{W}_y(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}_1^T} & \frac{\partial \mathcal{W}_y(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}_2^T} \end{bmatrix} \tag{2}$$

$$\frac{\partial \mathcal{W}(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}^T} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3}$$

What is $\mathbf{A}$ and $\mathbf{b}$?

We are trying to find the equation in the minimizing function below, so if we expand it out by taking $\mathbf{x}' = \mathbf{x} + \mathbf{p}$, we get the result below after grouping the Δp terms together.

$$\arg\min_{\Delta p} \sum_{\mathbf{x}\in\mathbb{N}} \left\| \mathcal{I}_{t+1}\left(\mathbf{x}' + \Delta\mathbf{p}\right) - \mathcal{I}_t\left(\mathbf{x}\right) \right\|_2^2 \tag{4}$$

$$= \arg\min_{\Delta p} \sum_{\mathbf{x}\in\mathbb{N}} \left\| \mathcal{I}_{t+1}\left(\mathbf{x}'\right) + \frac{\partial \mathcal{I}_{t+1}\left(\mathbf{x}'\right)}{\partial {\mathbf{x}'}^T} \frac{\partial \mathcal{W}\left(\mathbf{x};\mathbf{p}\right)}{\partial \mathbf{p}^T} \Delta\mathbf{p} - \mathcal{I}_t\left(\mathbf{x}\right) \right\|_2^2 \tag{5}$$

$$= \arg\min_{\Delta p} \sum_{\mathbf{x}\in\mathbb{N}} \left\| \frac{\partial \mathcal{I}_{t+1}\left(\mathbf{x}'\right)}{\partial {\mathbf{x}'}^T} \frac{\partial \mathcal{W}\left(\mathbf{x};\mathbf{p}\right)}{\partial \mathbf{p}^T} \Delta\mathbf{p} - \left[\mathcal{I}_t\left(\mathbf{x}\right) - \mathcal{I}_{t+1}\left(\mathbf{x}'\right)\right] \right\|_2^2 \tag{6}$$

Now that the inside of the minimizing function is in the form $A\Delta p - b$, we can see what $\mathbf{A}$ and $\mathbf{b}$ are below.

$$\mathbf{A} = \frac{\partial \mathcal{I}_{t+1}\left(\mathbf{x}'\right)}{\partial {\mathbf{x}'}^T} \frac{\partial \mathcal{W}\left(\mathbf{x};\mathbf{p}\right)}{\partial \mathbf{p}^T} \tag{7}$$

$$= \begin{bmatrix} \frac{\partial \mathcal{I}_{t+1}\left(\mathbf{x}'_{\mathbf{1}}\right)}{\partial \mathbf{x}'_{\mathbf{1}}{}^T} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial \mathcal{I}_{t+1}\left(\mathbf{x}'_{\mathbf{D}}\right)}{\partial \mathbf{x}'_{\mathbf{D}}{}^T} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{W}(\mathbf{x_1};\mathbf{p})}{\partial \mathbf{p}^T} \\ \vdots \\ \frac{\partial \mathcal{W}(\mathbf{x_D};\mathbf{p})}{\partial \mathbf{p}^T} \end{bmatrix} \tag{8}$$

$$= \begin{bmatrix} \frac{\partial \mathcal{I}_{t+1}\left(\mathbf{x}'_{\mathbf{1}}\right)}{\partial \mathbf{x}'_{\mathbf{1}}{}^T} \frac{\partial \mathcal{W}(\mathbf{x_1};\mathbf{p})}{\partial \mathbf{p}^T} \\ \vdots \\ \frac{\partial \mathcal{I}_{t+1}\left(\mathbf{x}'_{\mathbf{D}}\right)}{\partial \mathbf{x}'_{\mathbf{D}}{}^T} \frac{\partial \mathcal{W}(\mathbf{x_D};\mathbf{p})}{\partial \mathbf{p}^T} \end{bmatrix} \tag{9}$$

$$\mathbf{b} = \mathcal{I}_t\left(\mathbf{x}\right) - \mathcal{I}_{t+1}\left(\mathbf{x}'\right) \tag{10}$$

$$= \begin{bmatrix} \mathcal{I}_t\left(\mathbf{x_1}\right) - \mathcal{I}_{t+1}\left(\mathbf{x}'_{\mathbf{1}}\right) \\ \vdots \\ \mathcal{I}_t\left(\mathbf{x_D}\right) - \mathcal{I}_{t+1}\left(\mathbf{x}'_{\mathbf{D}}\right) \end{bmatrix} \tag{11}$$

What conditions must $\mathbf{A^T A}$ meet so that a unique solution to $\Delta\mathbf{p}$ can be found ?

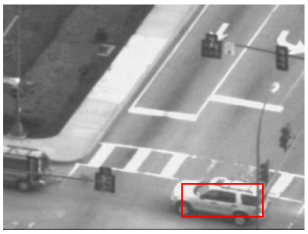
$\mathbf{A^T A}$ must have a full rank in order for $\Delta\mathbf{p}$ to have a unique solution. This can be determined by verifying that the determinant of $\mathbf{A^T A}$ is not equal to 0.

Q 1.2 Lucas Kanade script is included in code.

Q1.3

The images of the tracked car and tracked girl are below. As you can see, error accrues over time and the bounded region becomes more misaligned over time. The tracked girl error also increases after the girl crosses someone walking in the opposite direction. The default parameters were used.

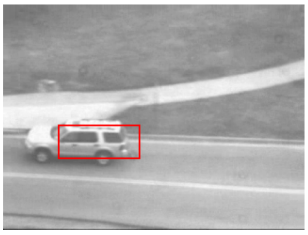
(a) Frame 1



(b) Frame 100

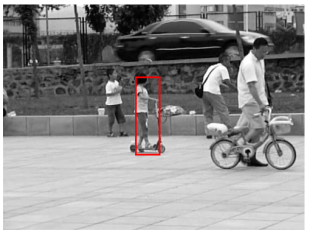
(c) Frame 200

(d) Frame 300

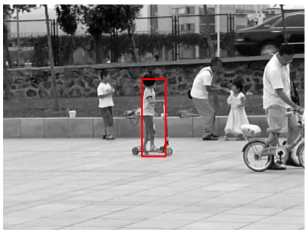


(e) Frame 400

Figure 1: Car Tracking without Correction

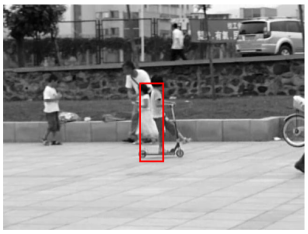


(a) Frame 1

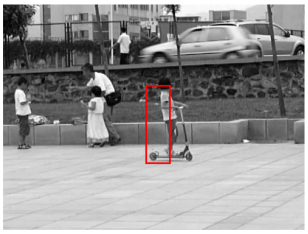


(b) Frame 20

(c) Frame 40



(d) Frame 60



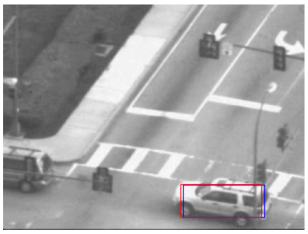
(e) Frame 80

Figure 2: Girl Tracking without Correction

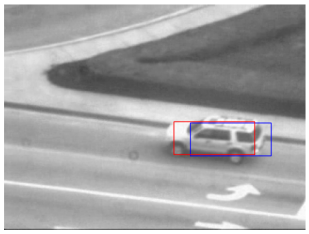
Q 1.4

The images of the tracked girl and tracked car with template corrections are below. The blue rectangles are the original tracker, and the red rectangles are the method with template correction. Default parameters were used, and as you can see the new tracker was able to eliminate much of the error.

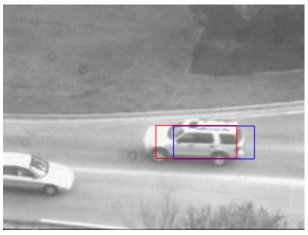
(a) Frame 1



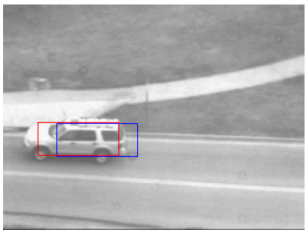
(b) Frame 100



(c) Frame 200

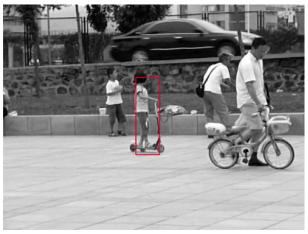


(d) Frame 300

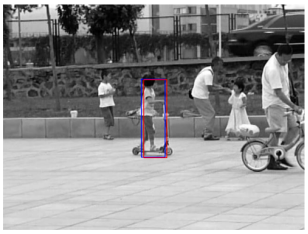


(e) Frame 400

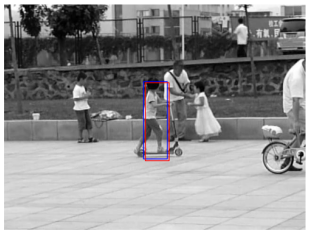
Figure 3: Car Tracking with Template Correction



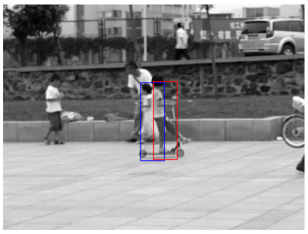
(a) Frame 1



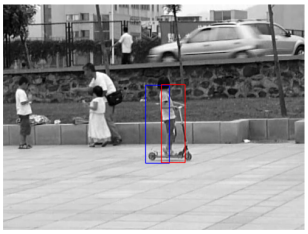
(b) Frame 20



(c) Frame 40



(d) Frame 60



(e) Frame 80

Figure 4: Girl Tracking with Template Correction

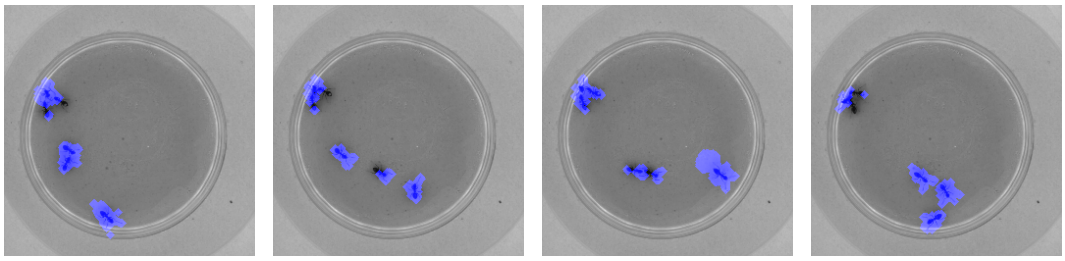
2 Affine Motion Subtraction

Q 2.1 Lucas Kanade Affine script included in the code.

Q 2.2 Subtract Dominant Motion script included in the code.

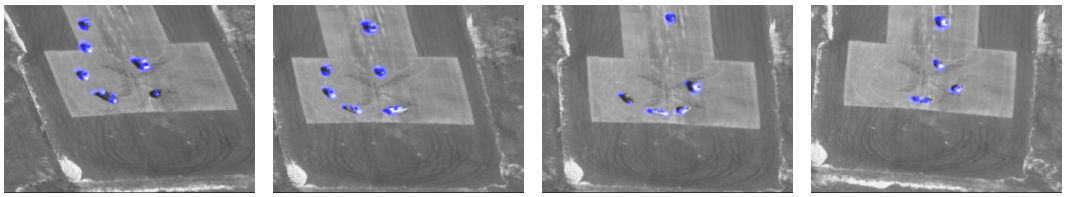
Q2.3

Results of the Lucas Kanade Affine motion detector are below. The aerial sequence uses default parameters, but the ant sequence uses a motion tolerance value of 0.05 since sometimes some of the ants barely move. In TestAntSequence.py, an additional binary dilation and erosion was applied to the mask to better show the tracking. The mask is overlaid on the gray images as blue pixels. As shown in the results, the tracker was able to effectivley track the individual ants and cars.



(a) Frame 30 (b) Frame 60 (c) Frame 90 (d) Frame 120

Figure 5: Ant Sequence Motion Detection with Regular Lucas Kanade Method



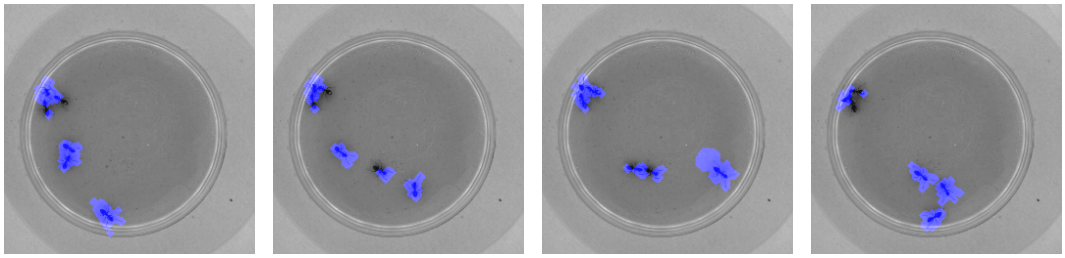
(a) Frame 30 (b) Frame 60 (c) Frame 90 (d) Frame 120

Figure 6: Aerial Sequence Motion Detection with Regular Lucas Kanade Method

3 Efficient Tracking

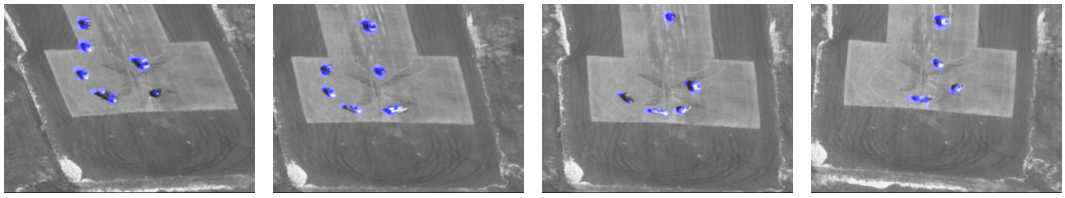
Q3.1

For the original Lucas Kanade affine method, the tests for the ant and aerial sequence lasted 49.3 and 115.3 seconds respectively. With the inverse composition method, the ant and aerial sequence lasted 29.1 and 67.6 seconds respectively, so the inverse composition method was more computationally efficient. This method is more computationally efficient because this method calculates an **A** matrix derived in Q 1.1 initially and then that matrix does not change. On the other hand, the regular Lucas Kanade method, calculates **A** every iteration. This saves time by not having to compute gradients each iteration to find **A**. The results for this method can also be seen below.



(a) Frame 30 (b) Frame 60 (c) Frame 90 (d) Frame 120

Figure 7: Ant Sequence Motion Detection with Inverse Composition Method



(a) Frame 30 (b) Frame 60 (c) Frame 90 (d) Frame 120

Figure 8: Aerial Sequence Motion Detection with Inverse Composition Method