16-720A Computer Vision: Homework 3 Lucas-Kanade Tracking

Radhika Mohanan(rmohanan)

March 7, 2020

Q1.1

Q1.1a

For a pure translation warp function

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

Q1.1b

Using first-order Taylor expansion we can linearize the objective function locally and on rearranging the equation to minimize we get as below.

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

$$where \;\; \mathcal{I}_{t+1}\left(\mathbf{x}' + \Delta\mathbf{p}\right) \approx \mathcal{I}_{t+1}\left(\mathbf{x}'\right) + \frac{\partial\mathcal{I}_{t+1}(\mathbf{x}')}{\partial\mathbf{x}'^T}\frac{\partial\mathcal{W}(\mathbf{x};\mathbf{p})}{\partial\mathbf{p}^T}\Delta\mathbf{p}$$

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

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

$$\arg\min_{\Delta\mathbf{p}} \|\mathbf{A}\Delta\mathbf{p} - \mathbf{b}\|_2^2$$

On comparing the last two above forms we get that:

$$\begin{aligned}\mathbf{A} &= \sum_{\mathbf{x}\in\mathbb{N}} \frac{\partial\mathcal{I}_{t+1}(\mathbf{x}')}{\partial\mathbf{x}'^T}\frac{\partial\mathcal{W}(\mathbf{x};\mathbf{p})}{\partial\mathbf{p}^T} \\ \mathbf{b} &= \sum_{\mathbf{x}\in\mathbb{N}} \mathcal{I}_t(\mathbf{x}) - \mathcal{I}_{t+1}\left(\mathbf{x}'\right)\end{aligned} \tag{3}$$

A represents the steepest descent images and **b** represents the error image.

Q1.1c

For minimizing $\Delta\mathbf{p}$ in equation 2, we take its first derivative and equate it to zero and we get the following:

$$\begin{gathered}
\textstyle\sum_{\mathbf{x}\in\mathbb{N}} 2\mathbf{A}^\top(\mathbf{A}\Delta\mathbf{p} - \mathbf{b}) = 0 \\
\textstyle\sum_{\mathbf{x}\in\mathbb{N}} \mathbf{A}^\top\mathbf{A}\Delta\mathbf{p} = \sum_{\mathbf{x}\in\mathbb{N}} \mathbf{A}^\top\mathbf{b} \\
\Delta\mathbf{p} = \textstyle\sum_{\mathbf{x}\in\mathbb{N}} (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}
\end{gathered} \tag{4}$$

In order to solve for $\Delta\mathbf{p}$, $\mathbf{A}^T\mathbf{A}$ must be invertible or non-singular matrix or must have a non-zero determinant to obtain a unique solution for $\Delta\mathbf{p}$.

1.3 Car and Girl Sequence Tracking

1. Test Car Sequence

Figure 1: Frame1

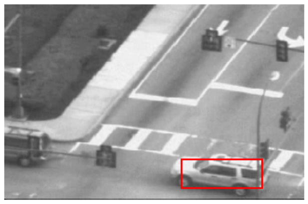

Figure 2: Frame100

Figure 3: Frame200

Figure 4: Frame300

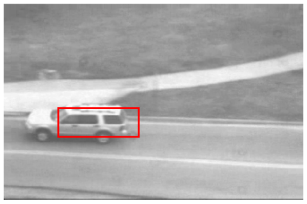

Figure 5: Frame400

2. Test Girl Sequence

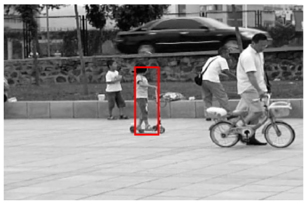

Figure 6: Frame1

Figure 7: Frame20

Figure 8: Frame40

Figure 9: Frame60

Figure 10: Frame80

1.4 Car and Girl Sequence Tracking with Template Drift Correction

In the below images, the red rectangles indicate the tracking results with template correction, and the blue rectangles indicate the results from Q1.3.Since the template is updated every frame and registered to the initial template (drift correction), the object is tracked correctly till the end and improves the tracking robustness compared to the original Lucas-Kanade.

1. Test Car Sequence(corrected)

Figure 11: Frame1

Figure 12: Frame100

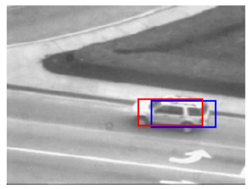

Figure 13: Frame200

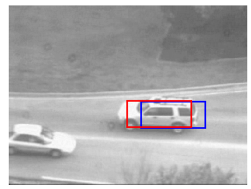

Figure 14: Frame300

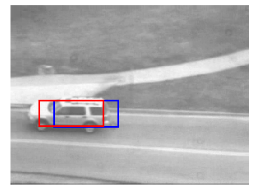

Figure 15: Frame400

2. Test Girl Sequence (corrected)

Figure 16: Frame1

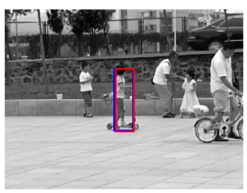

Figure 17: Frame20

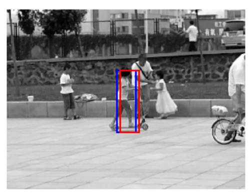

Figure 18: Frame40

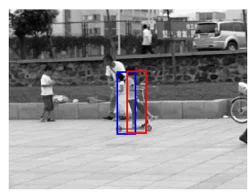

Figure 19: Frame60

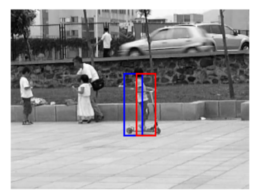

Figure 20: Frame80

2.3

In the images given below the moving objects were marked by blue dots. To remove the falsely detected moving objects on the edge while capturing the actual targets tuning is done where the tolerance for dominant motion estimation for aerial sequence was chosen to be 0.2 with default threshold and for ant sequence was chosen to be 0.02 with a threshold of 0.01. Functions "scipy.ndimage.morphology.binary erosion" was used to remove the edge-like points, and function "scipy.ndimage.morphology.binary dilation" was applied to expand the cover range of correctly detected moving objects.
Test Aerial Sequence

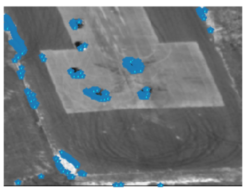

Figure 21: Frame30

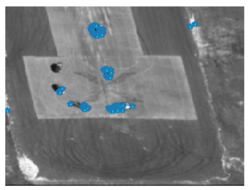

Figure 22: Frame60

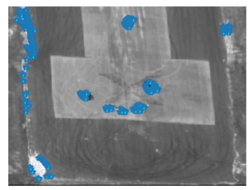

Figure 23: Frame90

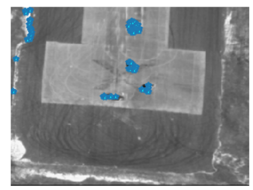

Figure 24: Frame120

Test Ant Sequence

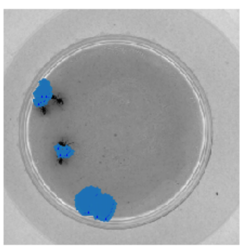

Figure 25: Frame30

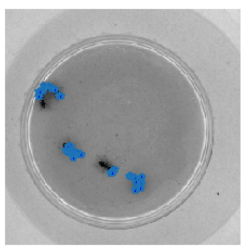

Figure 26: Frame60

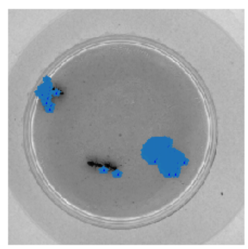

Figure 27: Frame90

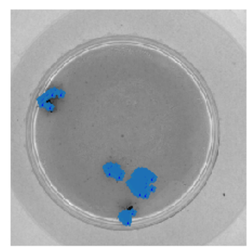

Figure 28: Frame120

3.1

In the classical approach we need to update A and b in every iteration until $\Delta\mathbf{p}$ converges.A being a very large matrix D X 6 matrix, it requires more time for the convergence of $\Delta\mathbf{p}$. However, with inverse compositional approach, A$'$ and $({A'}^T A)^{-1}{A'}^T$ can be precomputed only once, and then it can be multiplied to updated b until $\Delta\mathbf{p}$ converges, which saves a huge amount of computational time and costs.
Test Aerial Sequence

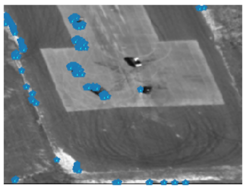

Figure 29: Frame30

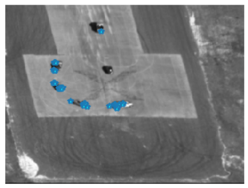

Figure 30: Frame60

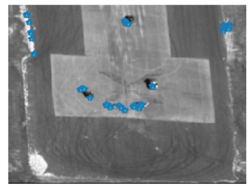

Figure 31: Frame90

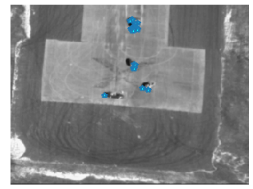

Figure 32: Frame120

Test Ant Sequence

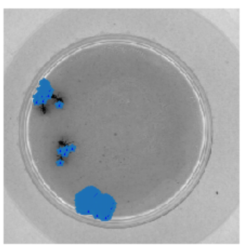

Figure 33: Frame30

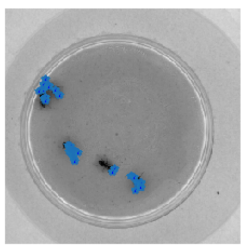

Figure 34: Frame60

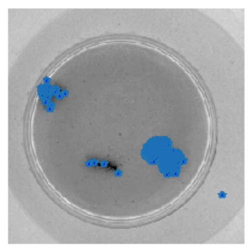

Figure 35: Frame90

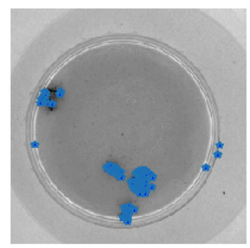

Figure 36: Frame120