Homework 3 Lucas-Kanade Tracking

Yasser Corzo

October 2023

Q1.1

1. $\frac{\partial W(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}^T}$ is the jacobian. The math is as follows:
   $\mathbf{W} = (W_x(x,y), W_y(x,y))$ $\mathbf{p} = (p_1, p_2, ..., p_n)^T$
   Assuming $\mathbf{p}$ from 2.1, $n = 6$
   $\frac{\partial W(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}^T} = \frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial \mathbf{W}_x}{\partial p_1} & \frac{\partial \mathbf{W}_x}{\partial p_2} & \cdots & \frac{\partial \mathbf{W}_x}{\partial p_6} \\ \frac{\partial \mathbf{W}_y}{\partial p_1} & \frac{\partial \mathbf{W}_y}{\partial p_2} & \cdots & \frac{\partial \mathbf{W}_y}{\partial p_6} \end{bmatrix} = \begin{bmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{bmatrix}$

2. $\mathbf{b}$ is the error image and $\mathbf{A}$ is the image gradient * the jacobian.
   $\mathbf{\Delta p} = \text{argmin}_{\Delta p} = \sum_{x \in N} ||I_{t+1}(\mathbf{x}+\mathbf{\Delta p}) + I_t(\mathbf{x})||_2^2 = \sum_{x \in N} ||\frac{\partial I_{t+1}(\mathbf{x})}{\partial \mathbf{x}^T} \frac{\partial W(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}^T} \Delta \mathbf{p} - (I_{t+1}(\mathbf{x}) - I_t(\mathbf{x}))||_2^2 = \sum_{x \in N} ||\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} + I(\mathbf{W}(\mathbf{x};\mathbf{p}) - T(x)||_2^2 = \text{argmin}_{\Delta p} ||\mathbf{A}\Delta \mathbf{p} - \mathbf{b}||_2^2$

3. The conditions for $\mathbf{A}^T\mathbf{A}$ so that a unique solution for Δp can be found is that it must be square and of full rank.

Q1.2

Lucas Kanade function implemented in code.

Q1.3

Figure 1: Lucas-Kanade Tracking with One Single Template

Q1.4

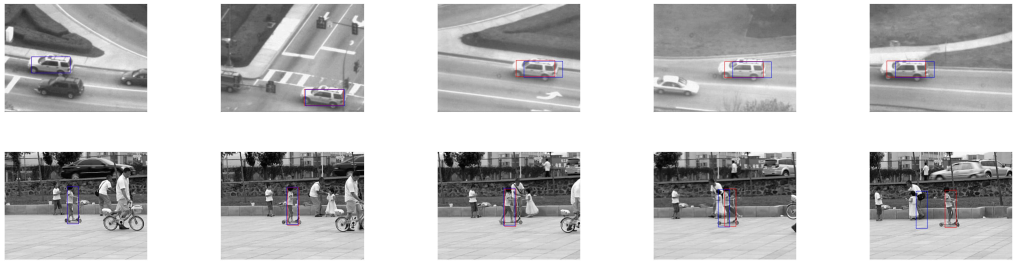

Figure 2: Lucas-Kanade Tracking with Template Correction

Q2.1

Lucas Kanade Affine function implemented in code.

Q2.2

SubtractDominantMotion function implemented in code.

Q2.3

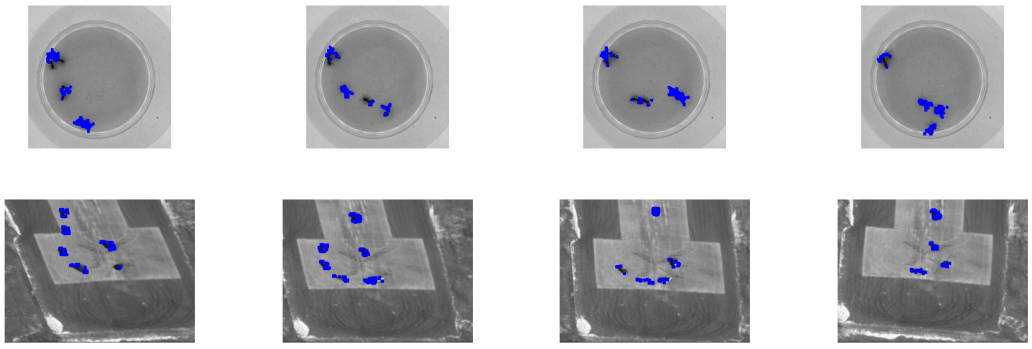

Figure 3: Lucas-Kanade Tracking with Motion Detection (ant sequence had tolerance of 0.08)

Q3.1

The inverse compositional approach is more computationally efficient than the classical approach because the jacobian, hessian, and gradient of the template is precomputed once before calculating the warp matrix, hence the runtime of the algorithm is around O(n). On the other hand, the classical approach involves calculating the Jacobian for every warped pixel (hence calculating the hessian and image gradient) during every iteration, hence the runtime of the algorithm is around $O(n^2)$.
The runtime of Aerial sequence on Lucas Kanade Affine was around 379 seconds. The runtime of Ant sequence on Lucas Kanade Affine was around 177 seconds. The runtime of Aerial sequence with inverse compositional extension of the Lucas-Kanade algorithm was around 103 seconds. The runtime of Ant sequence with inverse compositional extension of the Lucas-Kanade algorithm was 90 seconds. In the Aerial sequence case, there was a runtime improvement of around 72%, while for Ant sequence it was 49% improvement.

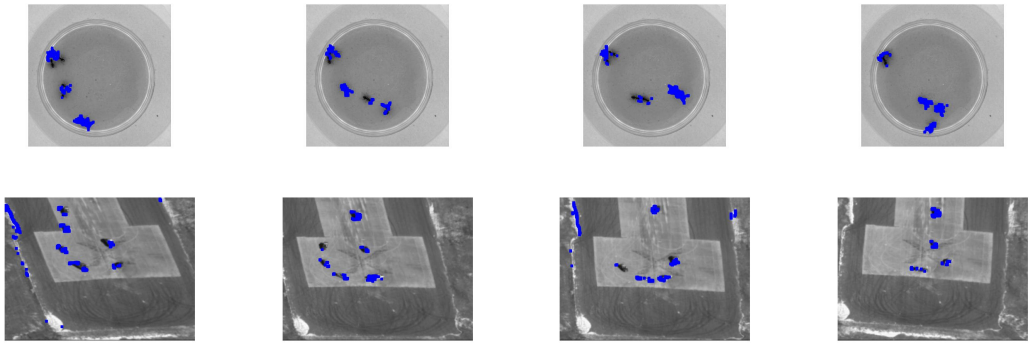

Figure 4: Lucas-Kanade Tracking with Motion Detection (ant sequence had tolerance of 0.08, aerial had tolerance of 0.28)