1 Lucas-Kanada Tracking

Q1.1

$$
\begin{aligned}
W(\mathbf{x}:\mathbf{p}) &= \begin{bmatrix} 1 & 0 & p_1 \\ 0 & 1 & p_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix} \\
&= \begin{bmatrix} W_x(\mathbf{x}:\mathbf{p}) \\ W_y(\mathbf{y}:\mathbf{p}) \end{bmatrix} \\
\frac{\partial W(\mathbf{x}:\mathbf{p})}{\partial p^T} &= \begin{bmatrix} \frac{\partial W_x(\mathbf{x}:\mathbf{p})}{\partial p_1} & \frac{\partial W_x(\mathbf{x}:\mathbf{p})}{\partial p_2} \\ \frac{\partial W_y(\mathbf{x}:\mathbf{p})}{\partial p_1} & \frac{\partial W_y(\mathbf{x}:\mathbf{p})}{\partial p_2} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned}
$$

Q1.2

$$
\begin{aligned}
& arg \min_{\Delta \mathbf{p}} \sum ||I_{t+1}(x; p + \Delta \mathbf{p}) - I_t(x)||_2^2 \\
&= arg \min_{\Delta \mathbf{p}} \sum ||I_{t+1}(x') + \frac{\partial I_{t+1}(x')}{\partial x'^T} \frac{\partial W(\mathbf{x}:\mathbf{p})}{\partial p^T} \Delta \mathbf{p} - I_t(x)||_2^2 \\
&= arg \min_{\Delta \mathbf{p}} \sum || \frac{\partial I_{t+1}(x')}{\partial x'^T} \frac{\partial W(\mathbf{x}:\mathbf{p})}{\partial p^T} \Delta \mathbf{p} - (I_t(x) - I_{t+1}(x'))||_2^2 \\
&= arg \min_{\Delta \mathbf{p}} \sum ||\mathbf{A} \Delta \mathbf{p} - \mathbf{b}||_2^2
\end{aligned}
$$

$$
\mathbf{x}' = w(\mathbf{x}; \mathbf{p}) = \mathbf{x} + \mathbf{p}
$$

$$
A = \begin{bmatrix}
\frac{\partial I_{t+1}(x')}{\partial x'^T} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\
\mathbf{0} & \frac{\partial I_{t+1}(x')}{\partial x'^T} & \ddots & \ddots & \vdots \\
\vdots & \ddots & \frac{\partial I_{t+1}(x')}{\partial x'^T} & \ddots & \vdots \\
\vdots & \ddots & \ddots & \frac{\partial I_{t+1}(x')}{\partial x'^T} & \mathbf{0} \\
\mathbf{0} & \cdots & \cdots & \mathbf{0} & \frac{\partial I_{t+1}(x')}{\partial x'^T}
\end{bmatrix}
$$

$$
b = \begin{bmatrix} I_t(x_1) - I_{t+1}(x_1') \\ \vdots \\ I_t(x_N) - I_{t+1}(x_N') \end{bmatrix}
$$

To make sure that a unique solution of Δp can be found, the $A^T A$ must to be a full rank matrix, which mean $det(A^T A) \neq 0$.

Q1.3
The tracking performance on the car sequence tracking is shown in figure 1.

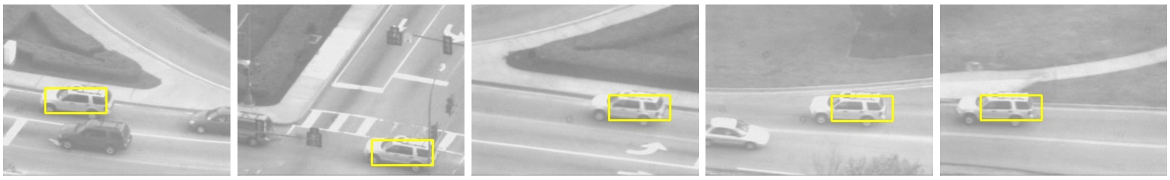

Figure 1: Lucas-Kanade Tracking with One Single Template(Frame 1, 100, 200, 300, 400).

Q1.4
The tracking performance on the car sequence tracking is shown in figure 2.

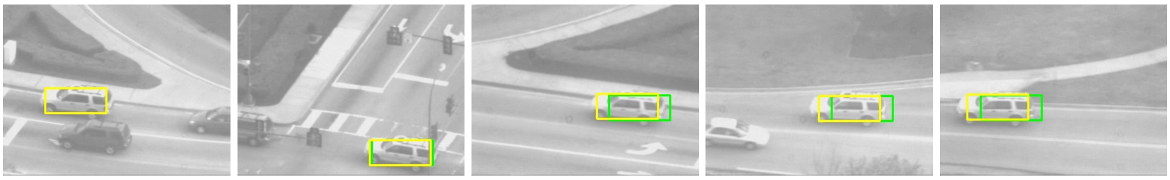

Figure 2: Lucas-Kanade Tracking with Template Correction(Frame 1, 100, 200, 300, 400).

2 Lucas-Kanade Tracking with Appearance Basis

Q2.1

$$
\begin{aligned}
I_{t+1}(x) &= I_t(x) + \sum_{k=1}^{K} w_k \\
B_k(x) I_{t+1}(x) &= I_t(x) + \mathbf{Bw} \\
I_{t+1}(x) - I_t(x) &= \mathbf{Bw} \\
\mathbf{B}^T(I_{t+1}(x) - I_t(x)) &= \mathbf{B}^T\mathbf{Bw} \\
\mathbf{w} &= \frac{\mathbf{B}^T(I_{t+1}(x) - I_t(x))}{\mathbf{B}^T\mathbf{B}} \\
\mathbf{w} &= \frac{\mathbf{B}^T}{\mathbf{B}^T\mathbf{B}}(I_{t+1}(x) - I_t(x)) \\
\mathbf{w} &= \begin{bmatrix} \frac{1}{||B_1(x)||} B_1(x)^T \\ \vdots \\ \frac{1}{||B_K(x)||} B_K(x)^T \end{bmatrix} (I_{t+1}(x) - I_t(x))
\end{aligned}
$$

Q2.2
Implemented `LucasKanadeBasis(It, It1, rect, bases, p0 = np.zeros(2))`.

Q2.3
The tracking performance on the car sequence tracking is shown in figure 3.

Figure 3: Lucas-Kanade Tracking with Appearance Basis(Frame 1, 200, 300, 350, 400).

3 Affine Motion Subtraction

Q3.1
Implemented `LucasKanadeAffine(It, It1)`.

Q3.2
Implemented `SubtractDominantMotion(image1, image2)`.

Q3.3

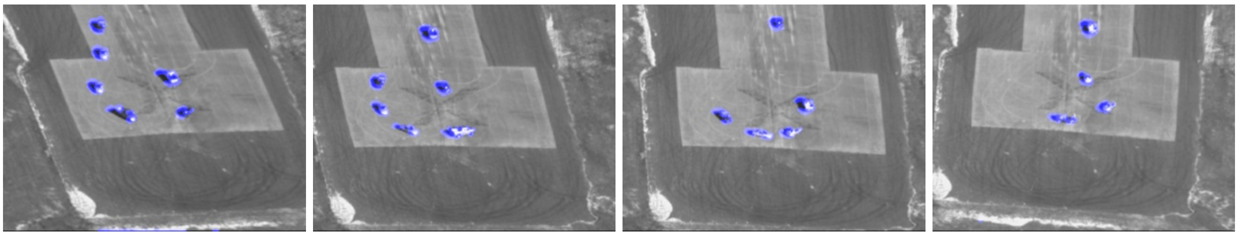

Figure 4: Lucas-Kanade Tracking with Appearance Basis(Frame 30, 60, 90, 120).

4 Efficient Tracking

Q4.1
The main computational bottle neck of the original algorithm is the cost in re-evaluating the Hessian in every iteration. The new idea is that we find a Hessian that were constant and it could be recomputed and re-used. Although the Hessian is a function of p in both formulations and it is hard to be approximated out by others method and remain same performance. The way it used is switching the role of the image and the template, and this will performed using compositional approach. We only need to compute the A matrix once, and update the p only using b.

In the testing for both approach, we can see the inverse composition finish the demo with 30.326 seconds and the traditional one needs 49.515 seconds, which is matching with our thought.

Q4.2

$$
\begin{aligned}
&arg \min_g \frac{1}{2}||y - X^T g||_2^2 + \frac{\lambda}{2}||g||_2^2 \\
&= arg \min_g \frac{1}{2}(y - X^T g)^T (y - X^T g) + \frac{\lambda}{2} g^T g \\
&= arg \min_g \frac{1}{2}(y^T y - y^T X^T g - g^T X y + g^T X X^T g) + \frac{\lambda}{2} g^T g \\
&= arg \min_g \frac{1}{2}(y^T y - 2 g^T X y + g^T X X^T g) + \frac{\lambda}{2} g^T g
\end{aligned}
$$

$$
\begin{aligned}
0 &= -Xy + XX^T g + \lambda g \\
Xy &= XX^T g + \lambda g \\
Xy &= (S + \lambda I) g \\
g &= (S + \lambda I)^{-1} (Xy)
\end{aligned}
$$

Q4.3

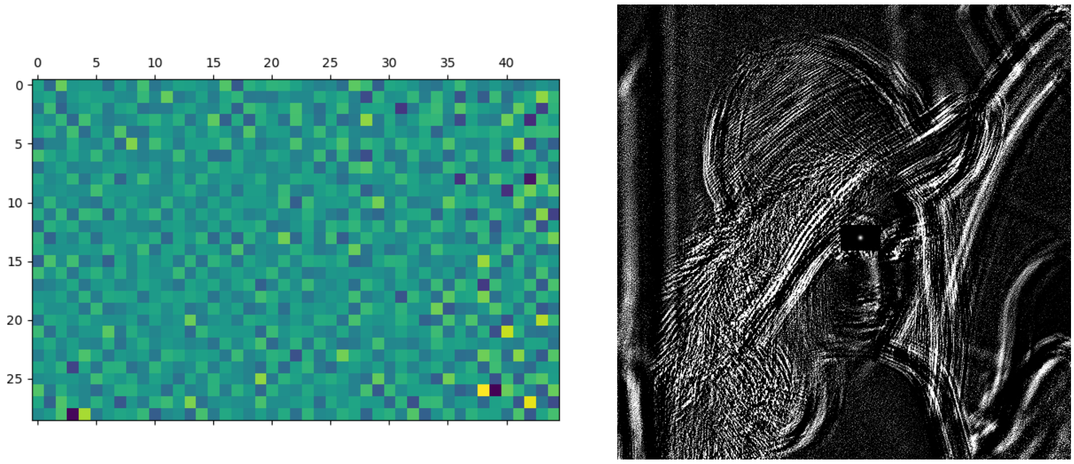

Figure 5: Visualization with correlation response $\lambda = 0$.

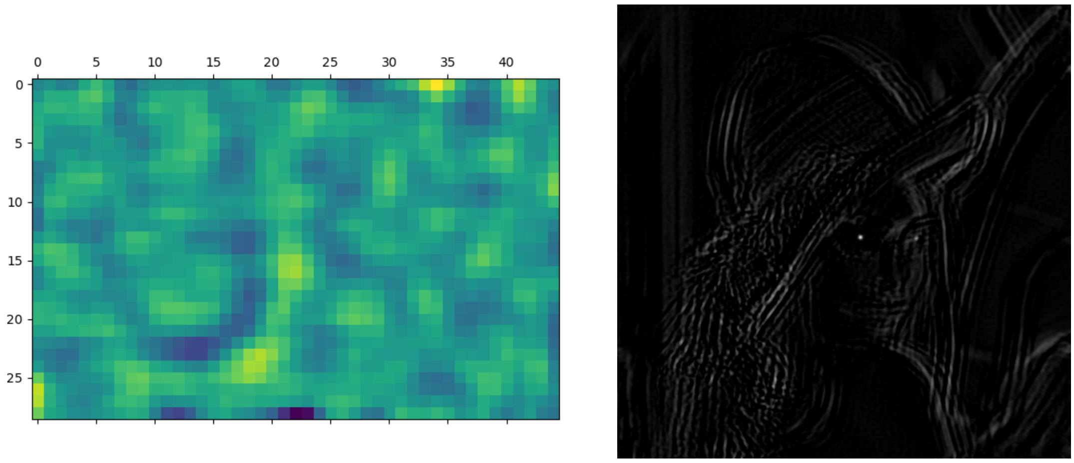

Figure 6: Visualization with correlation response $\lambda = 1$.

As the result shown in figure 5 and 6, we can see the image with $\lambda = 1$ is better, and the one with $\lambda = 0$ is more like some noise graph. With the information brings from the trust region, it is sufficient to use it and obtain the target position which is the white slot in the eyes.

Q4.4

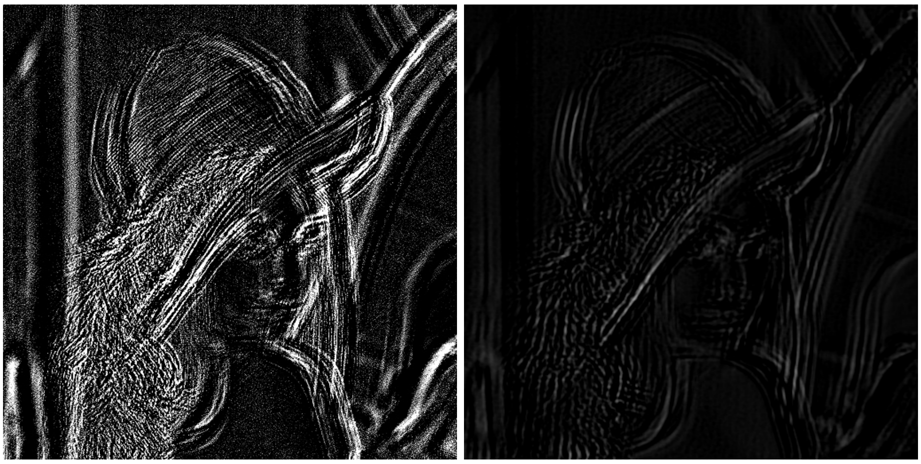

Figure 7: Visualization of convolution response.

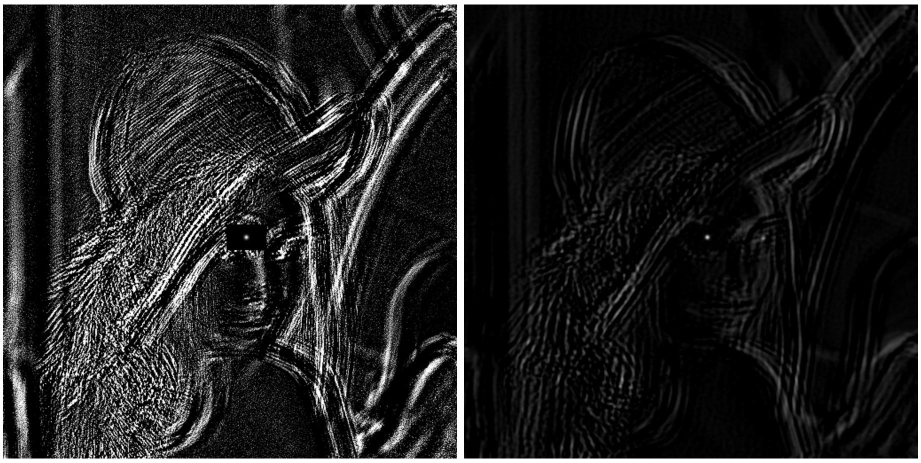

Figure 8: Visualization of convolution response with index rearrangement.

Since the `convolve` flipped the filter and then applied the correlation process, the output response must be different and it won't show the a spacial value for the target point, as shown in figure 7. However, we can first flipped the filter by changing the index of the array before applying the `convolve` function. As shown in figure 8, the result is the same as the one from **Q4.3**.