# 16-720 Homework 3

Chendi Lin

October 24, 2018

**Problem 1.1.**

- What is $\frac{\delta \mathcal{W}(\boldsymbol{x};\boldsymbol{p})}{\delta \boldsymbol{p}^T}$?
- What is $\boldsymbol{A}$ and $\boldsymbol{b}$?
- What conditions must $\boldsymbol{A^T A}$ meet so that a unique solution to $\Delta \boldsymbol{p}$ can be found?

**Solution .** (a) Since for a pure translation warp function

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

$$= \begin{bmatrix} x + 0 + p_1 \\ 0x + y + p_2 \end{bmatrix} \tag{2}$$

$$= \begin{bmatrix} \mathcal{W}_x(\boldsymbol{x};\boldsymbol{p}) \\ \mathcal{W}_y(\boldsymbol{x};\boldsymbol{p}) \end{bmatrix} \tag{3}$$

$$\frac{\delta \mathcal{W}(\boldsymbol{x};\boldsymbol{p})}{\delta \boldsymbol{p}^T} = \begin{bmatrix} \frac{\delta \mathcal{W}_x(\boldsymbol{x};\boldsymbol{p})}{\delta p_1} & \frac{\delta \mathcal{W}_x(\boldsymbol{x};\boldsymbol{p})}{\delta p_2} \\ \frac{\delta \mathcal{W}_y(\boldsymbol{x};\boldsymbol{p})}{\delta p_1} & \frac{\delta \mathcal{W}_y(\boldsymbol{x};\boldsymbol{p})}{\delta p_2} \end{bmatrix} \tag{4}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5}$$

(b) Since we are trying to obtain

$$arg \min_{\Delta \boldsymbol{p}} \sum ||\mathcal{I}_{t+1}(\boldsymbol{x; p + \Delta p}) - \mathcal{I}_t(\boldsymbol{x})||_2^2 \tag{6}$$

$$= arg \min_{\Delta \boldsymbol{p}} \sum ||\mathcal{I}_{t+1}(\boldsymbol{x'}) + \frac{\delta \mathcal{I}_{t+1}(\boldsymbol{x'})}{\delta \boldsymbol{x'}^T} \frac{\delta \mathcal{W}(\boldsymbol{x;p})}{\delta \boldsymbol{p}^T} \Delta \boldsymbol{p} - \mathcal{I}_t(\boldsymbol{x})||_2^2 \tag{7}$$

$$= arg \min_{\Delta \boldsymbol{p}} \sum ||\frac{\delta \mathcal{I}_{t+1}(\boldsymbol{x'})}{\delta \boldsymbol{x'}^T} \frac{\delta \mathcal{W}(\boldsymbol{x;p})}{\delta \boldsymbol{p}^T} \Delta \boldsymbol{p} - (\mathcal{I}_t(\boldsymbol{x}) - \mathcal{I}_{t+1}(\boldsymbol{x'}))||_2^2 \tag{8}$$

$$\tag{9}$$

where $\boldsymbol{x'} = \mathcal{W}(\boldsymbol{x};\boldsymbol{p}) = \boldsymbol{x} + \boldsymbol{p}$. So we have

$$\boldsymbol{A} = \begin{bmatrix} \frac{\delta \mathcal{I}_{t+1}(\boldsymbol{x'_1})}{\delta \boldsymbol{x'_1}^T} & \cdots & \mathbf{0}^T \\ \vdots & \ddots & \vdots \\ \mathbf{0}^T & \cdots & \frac{\delta \mathcal{I}_{t+1}(\boldsymbol{x'_N})}{\delta \boldsymbol{x'_N}^T} \end{bmatrix} \begin{bmatrix} \frac{\delta \mathcal{W}(\boldsymbol{x_1};\boldsymbol{p})}{\delta \boldsymbol{p}^T} \\ \vdots \\ \frac{\delta \mathcal{W}(\boldsymbol{x_N};\boldsymbol{p})}{\delta \boldsymbol{p}^T} \end{bmatrix} \tag{10}$$

$$\boldsymbol{b} = \begin{bmatrix} \mathcal{I}_t(\boldsymbol{x_1}) - \mathcal{I}_{t+1}(\boldsymbol{x'_1}) \\ \vdots \\ \mathcal{I}_t(\boldsymbol{x_N}) - \mathcal{I}_{t+1}(\boldsymbol{x'_N}) \end{bmatrix} \tag{11}$$

(c) To guarantee a unique solution, we want $\boldsymbol{A}$ to span all the dimensions, meaning that $\boldsymbol{A^T A}$ needs to be full rank. In another word, $det(\boldsymbol{A^T A}) \neq 0$.

**Problem 1.3 Test Car Sequence.**

Figure 1: Lucas-Kanade Tracking with One Single Template

The tracking results are shown in yellow rectangles.

**Problem 1.4 Car Tracking with Template Correction.**

Figure 2: Lucas-Kanade Tracking with Template Correction

For here, the yellow rectangles indicate the tracking results with template correction, and the green rectangles indicate the results from Q1.3. It is clear to see from the selected frames shown above that, the performance was improved a lot.

**Problem 2.1.** Express $\boldsymbol{w}$ as a function of $\mathcal{I}_{t+1}$, $\mathcal{I}_t$, and $\{\mathcal{B}_k\}_{k=1}^{K}$, given Equation 6. Note that since the $\mathcal{B}_k$s are orthobases, they are orthogonal to each other

**Solution .** Firstly, we flatten the equation

$$\mathcal{I}_{t+1}(x) = \mathcal{I}_t(x) + \sum_{k=1}^{K} \omega_k \mathcal{B}_k(x) \tag{12}$$

We can combine all the basis vector together to construct matrix $\boldsymbol{B}$ so that each column $i$ in $\boldsymbol{B}$ is the flattened basis $\mathcal{B}_i$. The equation above can be re-written as

$$\mathcal{I}_{t+1}(x) = \mathcal{I}_t(x) + \boldsymbol{B}\boldsymbol{w} \tag{13}$$

$$\mathcal{I}_{t+1}(x) - \mathcal{I}_t(x) = \boldsymbol{B}\boldsymbol{w} \tag{14}$$

$$\boldsymbol{B}^T(\mathcal{I}_{t+1}(x) - \mathcal{I}_t(x)) = \boldsymbol{B}^T\boldsymbol{B}\boldsymbol{w} \tag{15}$$

Since the columns in $\boldsymbol{B}$ are orthogonal to each other, $\boldsymbol{B^T B}$ is a $k \times k$ orthogonal matrix with each entry corresponding to the square of the norm of the basis. Thus,

$$\boldsymbol{w} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T(\mathcal{I}_{t+1}(x) - \mathcal{I}_t(x)) \tag{16}$$

$$= \frac{\boldsymbol{B}^T(\mathcal{I}_{t+1}(x) - \mathcal{I}_t(x))}{\boldsymbol{B}^T\boldsymbol{B}} \tag{17}$$

where the division means that, since $\boldsymbol{B^T B}$ is a diagonal matrix, the inversion of it can simply be the inversion of each entry in $\boldsymbol{B^T B}$.

If $\boldsymbol{B}$ is orthonormal, $\boldsymbol{B^T B = I}$. Then, $\boldsymbol{w} = \boldsymbol{B}^T(\mathcal{I}_{t+1}(x) - \mathcal{I}_t(x))$.

**Problem 2.2 Tracking.** Reformulating the problem, we have

$$arg \min_{\Delta p, w} ||A\Delta p - b - Bw||_2^2 \tag{18}$$

$$=arg \min_{\Delta p} ||B^{\perp}(A\Delta p - b)||_2^2 \tag{19}$$

$$=arg \min_{\Delta p} ||(A\Delta p - b) - BB^T(A\Delta p - b)||_2^2 \tag{20}$$

$$=arg \min_{\Delta p} ||(A - BB^T A)\Delta p - (b - BB^T b)||_2^2 \tag{21}$$

$$\tag{22}$$

**Problem 2.3 Test Sylv Sequence.**

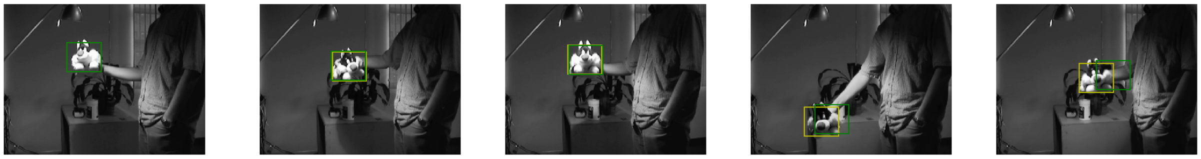

Figure 3: Lucas-Kanade Tracking with Appearance Basis

Lucas-Kanade Tracking with Appearance Basis is compared with Lucas-Kanade Tracking with Template Correction. In the figures shown above, the yellow rectangles indicate the tracking results with appearance basis, and the green rectangles indicate the results with Template Correction. It is obvious that, with appearance basis, the tracking results were greatly improved.

**Problem 3.3 Moving object Detection.**

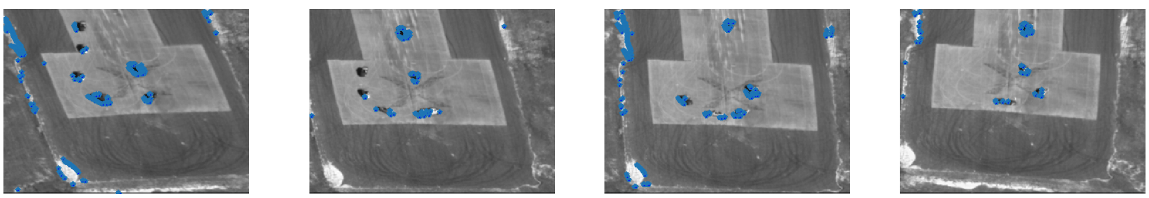

Figure 4: Moving Object Detection using Affine Motion Subtraction

The moving objects were marked by blue dots. To remove the falsely detected moving objects on the edge while capturing the actual targets nicely, by some engineering tuning, the threshold for dominant motion estimation was chosen to be 0.2. Function "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.

**Problem 4.1 Inverse Composition.**

**Solution .** $A$ is a $D\times 6$ matrix, which can be very large, especially when regarding the whole image as a template. Using classical approach, we need to update $A$ and $b$ in every iteration until $\Delta p$ converges. 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 $\Delta p$ converges, which saves a huge amount of computational costs.

**Problem 4.2.** Solve a linear least-square discriminant

**Solution .** Solving

$$
\begin{aligned}
&arg\min_g \frac{1}{2}||y - X^T g||_2^2 + \frac{\lambda}{2}||g||_2^2 && (23)\\
=&arg\min_g \frac{1}{2}(y - X^T g)^T(y - X^T g) + \frac{\lambda}{2}g^T g && (24)\\
=&arg\min_g \frac{1}{2}(y^T y - g^T X y - y^T X^T g + g^T X X^T g) + \frac{\lambda}{2}g^T g && (25)\\
=&arg\min_g \frac{1}{2}(y^T y - 2g^T X y + g^T X X^T g) + \frac{\lambda}{2}g^T g && (26)\\
& && (27)
\end{aligned}
$$

We can take the derivative of the equation above with respect to $g$ and set it to zero, so we have

$$
\begin{aligned}
-Xy + XX^T g + \lambda g &= 0 && (28)\\
XX^T g + \lambda g &= Xy && (29)\\
(S + \lambda I)g &= Xy && (30)\\
g &= (S + \lambda I)^{-1}(Xy) && (31)
\end{aligned}
$$

**Problem 4.3 Response Visualization.**

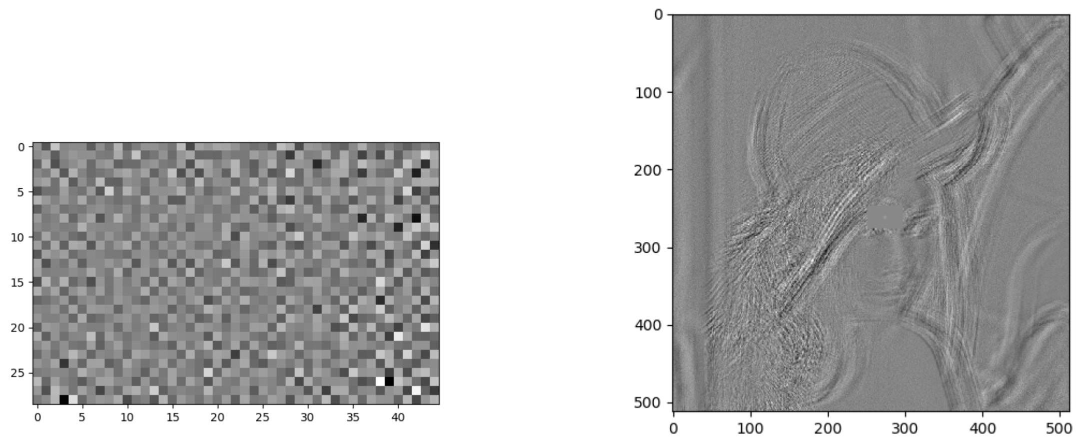


Figure 5: Visualization of filter and the Correlation Response with $\lambda = 0$

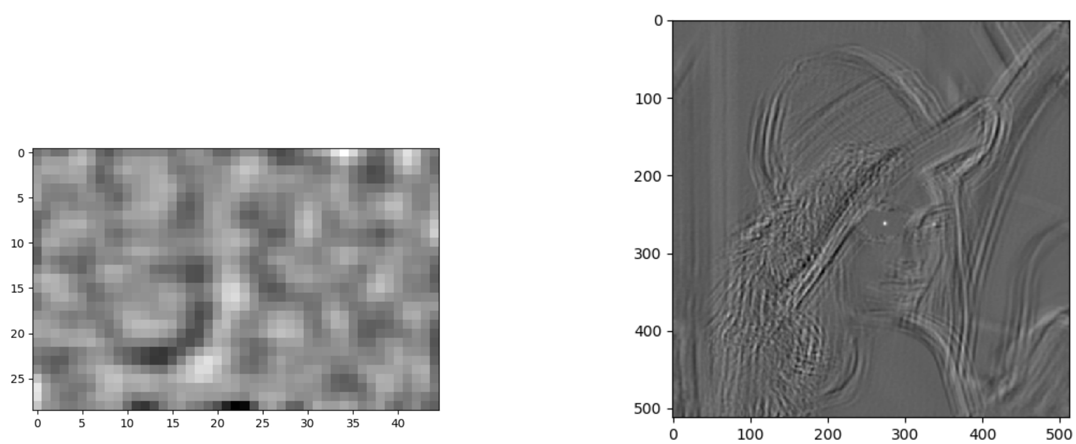


Figure 6: Visualization of filter and the Correlation Response with $\lambda = 1$

The figures above are the visualizations of the resultant linear discriminant weight vectors and their correlation responses for different penalty value. From the results we can see than the case when $\lambda = 1$ worked better than that of $\lambda = 0$. $\lambda = 1$ added a constraint on the norm of the weight vector, i.e., the trust region. With the trust region, we can see a clear peak point in the response (the white point). However, when $\lambda$, meaning that no trust region was applied to the filter, barely any information can be extracted from the response.

**Problem 4.4 Convolution Response.**

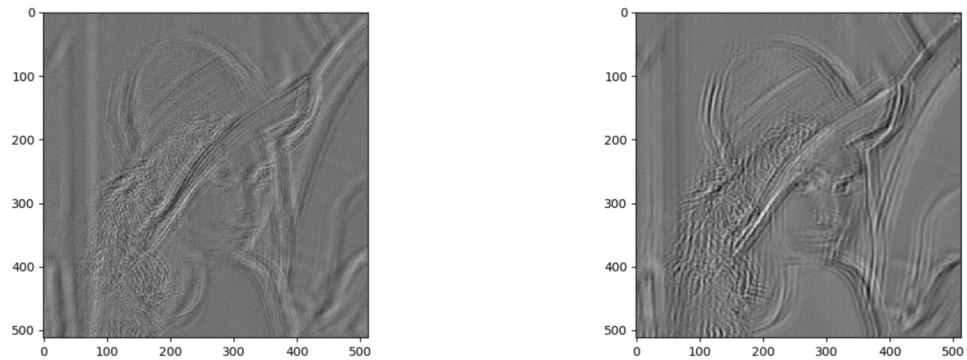


Figure 7: Visualization of the Convolution Response with $\lambda = 0$ and $\lambda = 1$

The convolution responses are shown in the figures above. They are different from those obtained by "correlate" because comparing to "correlate", "convolve" flipped the filter and then applied the correlation process.

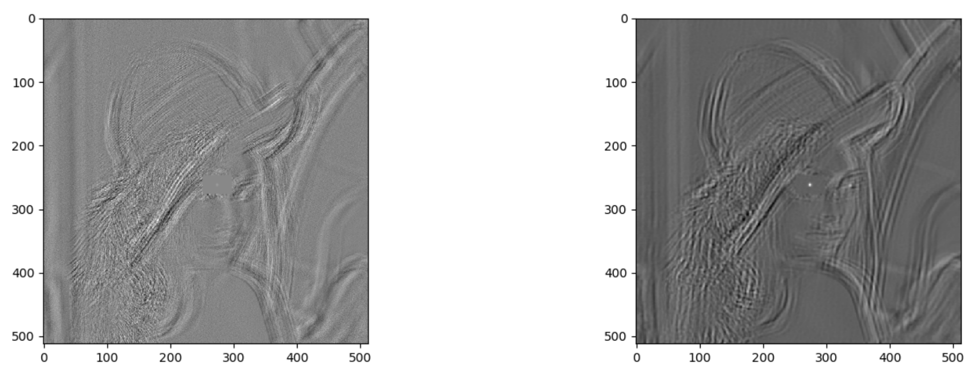


Figure 8: Visualization of the Convolution Response with Index Rearrangement and $\lambda = 0$ and $\lambda = 1$

Here, we firstly flipped the filter $g$ by numpy indexing operations ($g = g[::-1]$), and then applied the function "scipy.ndimage.convolve". The results look the same as those obtained from "scipy.ndimage.correlate", which verified the method.