Q1.1:

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial \mathbf{W_u}}{\partial \mathrm{p}_1} & \cdots & \frac{\partial \mathbf{W_u}}{\partial \mathrm{p}_6} \\ \frac{\partial \mathbf{W_v}}{\partial \mathrm{p}_1} & \cdots & \frac{\partial \mathbf{W_v}}{\partial \mathrm{p}_6} \end{bmatrix} = \begin{bmatrix} u & 0 & v & 0 & 1 & 0 \\ 0 & u & 0 & v & 0 & 1 \end{bmatrix}$$

$$\mathbf{L} = \sum[\mathbf{T}(\mathbf{x}) - \mathbf{I}(\mathbf{W})]^2 \Rightarrow \frac{\partial \mathbf{L}}{\partial \mathbf{I}(\mathbf{W})} = -2\sum[\mathbf{T}(\mathbf{x}) - \mathbf{I}(\mathbf{W})]$$

$$\frac{\partial \mathbf{I}(\mathbf{W})}{\partial \mathbf{W}} = \begin{bmatrix} \frac{\partial \mathbf{I}}{\partial \mathbf{u}} & \frac{\partial \mathbf{I}}{\partial \mathbf{v}} \end{bmatrix}$$

Thus:

$$J = \frac{\partial \mathbf{L}}{\partial \mathbf{p}} = \frac{\partial \mathbf{L}}{\partial \mathbf{I}(\mathbf{W})} \frac{\partial \mathbf{I}(\mathbf{W})}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{p}} = -2\sum[\boldsymbol{T}(\boldsymbol{x}) - \boldsymbol{I}(\boldsymbol{W})] \begin{bmatrix} \frac{\partial \mathbf{I}}{\partial \mathbf{u}} & \frac{\partial \mathbf{I}}{\partial \mathbf{v}} \end{bmatrix} \begin{bmatrix} u & 0 & v & 0 & 1 & 0 \\ 0 & u & 0 & v & 0 & 1 \end{bmatrix}$$

Q1.2:

Inverse Compositional method:

Initialization:

Jacobian: $O(np)$

Hessian: $O(np) + O(np^2)$

Inverse Hessian: $O(p^3)$

Sum: $O(np^2+p^3)$

Iteration:

Warp: $O(np)$

Error: $O(n)$

deltaP: $O(np) + O(p^3)$

Update warp: $O(p^2)$

Sum: $O(np + p^3)$

Lucas-Kanade: for each iteration

Hessian: $O(np) + O(np^2)$

Inverse Hessian: $O(p^3)$

And some other steps...

Sum: at least $O(np^2+p^3)$

Therefore, comparing these two methods, the former run faster because the computational cost of an iteration is lower, and the initial computational cost is only charged once.

Q2.4:

```
x = 170;
y = 150;
w = 100;
h = 80;
tracker = [x y w h];
```

For LK tracker, the template I used is small with the size match the last frame of the sequence and placed to track the number plate because it includes corners. Moreover, when the scene's illumination changed, the light of the number plate is still quite clear, thus allow the tracker to track it. If we set the template to be bigger, the tracker might be distracted by the background and break down, since there is not a lot of corners that the track can follow comparing with the background.

For MB tracker, I used the same template, but the tracker breaks down when the illumination change. The reason should be the MB is not particularly robust, and both trackers tend to break down when illumination change fast or there is fast movement.

Q2.5: For this sequence, using MB tracker give us a better result:

Note the comment sections in lk_demo.m and mb_demo.m. Use them to check my implementation.

```
x = 425;
y = 80;
w = 150;
h = 60;
tracker = [x y w h];
```

Q3.1: Test directly in lk_demo.m. Change the initial video as well as the tracker's method in iteration to test. Since the initial tracker is robust enough, using this tracker is not actually necessary in car sequence case. The running time is also much longer.

Q3.2: Test directly in lk_demo.m. Change the initial video as well as the tracker's method in iteration to test. As expected, this method gives us the good result with a quicker running time.