COMP4901L Assignment7 Writeup

WONG Yuk Chun (ycwongal, 20419764)

CHEUNG Daniel (dcheungaa, 20423088)

Part 1 Theory Questions

Q1.1: Calculating the Jacobian

$$\text{Assuming}$$
$$\mathbf{W}(\mathbf{x};\mathbf{p}) = \begin{bmatrix}\mathbf{W}_x \\ \mathbf{W}_y \\ \mathbf{W}_1\end{bmatrix}\begin{bmatrix}u \\ v \\ 1\end{bmatrix} = \begin{bmatrix}1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u \\ v \\ 1\end{bmatrix}$$
$$\text{and,}$$
$$\mathbf{p} = [p_1 \quad p_2 \quad p_3 \quad p_4 \quad p_5 \quad p_6]^T$$

$$\begin{aligned}\frac{\partial \mathbf{W}}{\partial \mathbf{p}} &= \begin{bmatrix}\frac{\partial \mathbf{W}_u}{\partial \mathbf{p}_1} & \cdots & \frac{\partial \mathbf{W}_u}{\partial \mathbf{p}_6} \\ \frac{\partial \mathbf{W}_v}{\partial \mathbf{p}_1} & \cdots & \frac{\partial \mathbf{W}_v}{\partial \mathbf{p}_6}\end{bmatrix} \\ &= \begin{bmatrix}u & 0 & v & 0 & 1 & 0 \\ 0 & u & 0 & v & 0 & 1\end{bmatrix}\end{aligned}$$

$$L = \sum_{\mathbf{x}}[\mathbf{T}(x) - \mathbf{I}(\mathbf{W})]^2$$

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

Q1.2: Computational complexity

Initialization step

Evaluate $\nabla \mathbf{T}$, which the runtime is $O(n)$

Evaluate $\mathbf{J} = \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $\mathbf{p} = \mathbf{0}$,

To compute $\nabla \mathbf{T}\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$, since $\nabla \mathbf{T}$ is of dimension $(n \times 2)$ and $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is of dimension $(2 \times p)$, this multiplication will have computational cost $O(np)$.

To compute Hessian matrix $\mathbf{H} = \mathbf{J}^{\mathbf{T}}\mathbf{J}$, since $\mathbf{J}$ is of dimension $(n \times p)$, this multiplication will have computational cost $O(np^2)$

Thus the initialization cost will be $O(np + np^2) = O(np^2)$.

Incremental step

First compute $\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p}))$, which is $O(np)$.

Then compute $[\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) - \mathbf{T}]$, which is $O(n)$.

Then compute $\sum_{\mathbf{x}} (\nabla \mathbf{T}\frac{\partial \mathbf{W}}{\partial \mathbf{p}})^{\mathbf{T}}[\mathbf{I}(\mathbf{W}(\mathbf{x};\mathbf{p})) - \mathbf{T}(\mathbf{x})]$, which is $O(pn)$.

Then multiply previous result with inverse of Hessian to get $\Delta \mathbf{p}$, which is $O(p^3)$.

Then update the warp function, which is $O(p^2)$

Total $O(pn + p^3)$

Part 2 Lucas-Kanade Tracker

Q2.1

initial rects:

car: `[180 180 36 36]`

landing: `[552 126 20 20]`

Q2.3

initial rects:

car: `[126   102   208   177]`

landing: `[440    79   118    58]`

Q2.4

Small templates on spots or corners generally work best for Lucas-Kanade. Anything that would not exibit the barber-pole problem. The tracker tends to breakdown when large motions occur or when the brightness levels suddenly changed because there would be big different everywhere inside the tracker so it difficult for the Jacobian to tell which direction is the steepest

Part 3

Q3.1X Robust LK

initial rect:

car: `[158   144    92    66]`

landing: `[440    79   118    58]`

use `ec/ilk_demo.m` to run

Q3.2X Pyramid LK

initial rect:

car: `[188   175    85    34]`

landing: `[440    82    52    44]`

use `ec/plk_demo.m` to run

Best Videos

For the best outputs, the following methods are used for the clips in our `results/` folder.

- Car: Robust LK

- Landing: Pyramid LK