Homework 4
M1522.001800 Computer Vision (2019 Spring)

2014-10469 InSeoung Han
Date: May 6 Monday

1 Theory Questions [30 pts]

1.1 Question : Calculating the Jacobian

$$W(x;p) = \begin{bmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} (1+p_1)u + p_3 v + p_5 \\ p_2 x + (1+p_4)v + p_6 \end{bmatrix}$$

$$\frac{\partial W}{\partial p} = \begin{bmatrix} u & 0 & v & 0 & 1 & 0 \\ 0 & u & 0 & v & 0 & 1 \end{bmatrix}$$

1.2 Question : Calculating the Gradient

$$\nabla T(x) = \begin{bmatrix} \frac{\partial T(x)}{\partial u} & \frac{\partial T(x)}{\partial v} \end{bmatrix}$$

and from before question,

$$\frac{\partial W}{\partial p} = \begin{bmatrix} u & 0 & v & 0 & 1 & 0 \\ 0 & u & 0 & v & 0 & 1 \end{bmatrix}$$

$$G(x) = \nabla T(x) \frac{\partial W}{\partial p} = \begin{bmatrix} \frac{\partial T(x)}{\partial u}u & \frac{\partial T(x)}{\partial v}u & \frac{\partial T(x)}{\partial u}v & \frac{\partial T(x)}{\partial v}v & \frac{\partial T(x)}{\partial u} & \frac{\partial T(x)}{\partial v} \end{bmatrix}$$

1.3 Question : Computational complexity of initialization

First try to get ∇T by convolving T(x) and 3x3 sobel operator. convolution processs takes O(MNmn) for $M \times N$ matrix and $m \times n$ matrix, but the size of sobel operator is constant and the size of T(x) is n, actually it takes just O(n).

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure 1: 3x3 sobel operator

$$G = \nabla T \frac{\partial W}{\partial p}$$

After getting ∇T, multiplying Jacobian Matrix $\frac{\partial W}{\partial p}$ by T(x) for all points x, gathering it up, we can get G. matrix multiplication takes $O(M \times N \times R)$ for $M \times R$ matrix and $R \times N$ matrix. So it takes $O(np)$ to get G for our situation because ∇T is $n \times 2$ matrix and $\frac{\partial W}{\partial p}$ is $2 \times p$ matrix.

After we get G which is $n \times p$ matrix, we can compute H by using it. Process to get H which means $G^T G$ takes $O(np^2)$ to multiply G^T by G.

1.4 Question : Computational complexity of iteration

Given $G^T(p \times n)$, $G(n \times p)$, $H^{-1}(p \times p)$, we just need 5 steps step1) Warp I with $W(x;p)$ to $I(W(x;p))$
this step is just doing multiply W matrix with all points in T, which normally takes $O(np)$

step2) Compute error image $I(W(x;p)) - T(x)$
this step is trivial, which takes O(n) to subtract T(x) to I(W(x;p)).

step3) Compute $\Sigma_x[\nabla T \frac{\partial W}{\partial p}]^T[I(W(x;p)) - T(x)]$
we already know $[\nabla T \frac{\partial W}{\partial p}]^T$ and $[I(W(x;p)) - T(x)]$. So multiply two terms for all points takes O(np).

step4) Compute Δp

$$\Delta p = H^{-1} G^T [I_{warped} - T]$$

we know every term because of previous steps and pre-computation steps. So, just multiplying those terms takes $O(np^2)$.

step5) Update the warp $W(x;p) \Leftarrow W(x;p) \circ W(x;\Delta p)^{-1}$. normally this process takes $O(p^2)$

Therefore, total cost is $O(np + p^2)$.

1.5 Question : Practical Issue I

use sobel operator.

1.6 Question : Practical Issue II

use Harris Corner. In this starting point, moving any wrong side make error huge. So, we can more easily find global optima.