CS-E4950 Computer Vision
Exercise Round 8

Rui Qu
rui.qu@aalto.fi

August 19, 2020

Exercise 1

c) What could be the main reasons why most of the features are not tracked very long in case b) above?

Because the tracking algorithm will tracks liners and eliminate outliers from the last frame by using RANSCA. Key points will be missed when the image is rotated or the camera moves at a fast speed.

d) How could one try to avoid the problem of gradually losing the features? Suggest one or more improvements.

We try to avoid large movements over a short period of time so the system can learn new features and track them after a period of time. Another try is to keep outliers rather than eliminating them for further tracking, but the problem is points always exist even if you move your face out of the camera.

e)

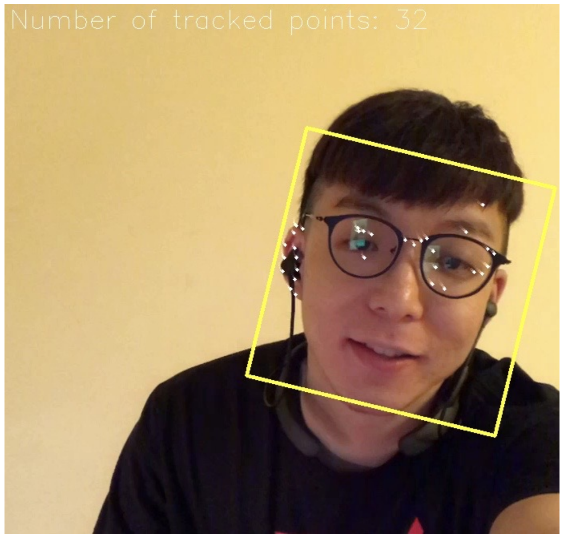

Figure 1: title

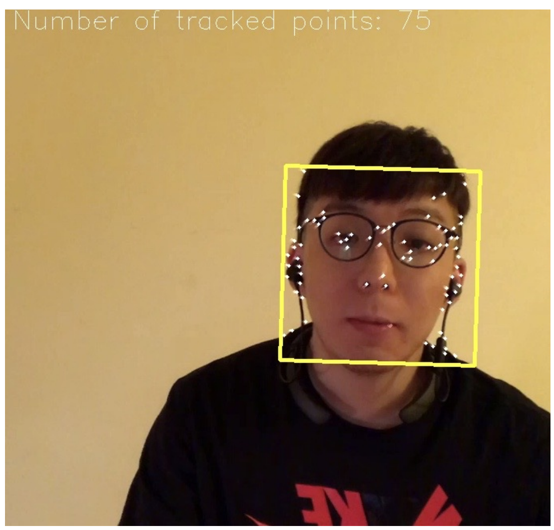

Figure 2: title

Exercise 2

Equation10 in the paper:

$$\Delta p = H^{-1} \sum_{x} \left[\nabla I \frac{\delta W}{\delta p} \right]^{\mathsf{T}} \left[T(x) - I(W(x;p)) \right] \tag{1}$$

where,

$$
\begin{aligned}
\frac{\delta W}{\delta p} &= \begin{bmatrix} \frac{\delta W_x}{\delta u} & \frac{\delta W_x}{\delta v} \\ \frac{\delta W_y}{\delta u} & \frac{\delta W_y}{\delta v} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
\Delta P &= \begin{bmatrix} u \\ v \end{bmatrix}
\end{aligned} \tag{2}
$$

the Hessian matrix:

$$
\begin{aligned}
H &= \sum_x \left[\nabla I \tfrac{\delta W}{\delta p}\right]^{\mathsf{T}} \left[\nabla I \tfrac{\delta W}{\delta p}\right] \\
&= \sum_x \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\delta I}{\delta x} \\ \frac{\delta I}{\delta y} \end{bmatrix} \begin{bmatrix} \frac{\delta I}{\delta x} & \frac{\delta I}{\delta y} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix}
\end{aligned} \tag{3}
$$

Thus, Equation10 in the paper could be re-written:

$$
\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix}^{-1} \sum_x \begin{bmatrix} I_x \\ I_y \end{bmatrix} \left[T(x) - I(W(x;p))\right] \tag{4}
$$

which is the same as the equation in the slides:

$$
\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_y \\ \sum I_y I_t \end{bmatrix} \tag{5}
$$

Algorithm 1 Framework of ensemble learning for our system.

Input: The set of positive samples for current batch, P_n; The set of unlabelled samples for current batch, U_n; Ensemble of classifiers on former batches, E_{n-1};

Output: Ensemble of classifiers on the current batch, E_n;

1: Extracting the set of reliable negative and/or positive samples T_n from U_n with help of P_n;

2: Training ensemble of classifiers E on $T_n \cup P_n$, with help of data in former batches;

3: $E_n = E_{n-1} cup E$;

4: Classifying samples in $U_n - T_n$ by E_n;

5: Deleting some weak classifiers in E_n so as to keep the capacity of E_n;

6: **return** E_n;