CS-E4850 Computer Vision
Exercise Round 8

Alex Herrero Pons

Spring Term Course 2020-2021

Exercise 1. Face tracking example using KLT tracker.
This exercise is based on the python face tracking demo `Exercise8.ipynb`.

Run the example as instructed below and answer the questions.

a) Run `Exercise8.ipynb` with input `frames=faceTracker('santi.avi')`.

Solution. Executing the program we strartede with 80 features and ended up with 51.

b) Run `Exercise8.ipynb` with a different input by changing the input to obama.avi: `frames=faceTracker('obama.avi')`.

Solution. Now when executing the program the features went from 80 to 3.

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

Solution. The main reason we noticed is rotation. When the image is rotated a high amount of the tracked features are dropped. Also the speed of the movement of the camera can be another main reason so that the program can't process the image that fast and it looses track of the points.

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

Solution. Not having rotation and fast movements would be the best way to avoid this problem. We could also try to improve the processing speed somehow so that it allows faster movements

Exercise 2. Kanade-Lucas-Tomasi (KLT) feature tracking (Pen & paper problem) Read Sections 2.1 and 2.2 from the paper by Baker and Matthews (`https://www.ri.cmu.edu/pub_files/pub3/baker_simon_2002_3/baker_simon_2002_3.pdf`). Show that the Equation (10) in the paper gives the same solution as the equations on slide 25 of Lecture 7, when the geometric warping $\mathbf{W}$ (between the current frame and the template window in the previous frame) is a translation.

Solution.
Having the equation shown in the Baker and Matthews paper,

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

$$H\Delta\mathbf{p} = \sum_{\mathbf{x}}\left[\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]\left[T(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p}))\right],$$

and knowing that

$$\Delta\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix},$$

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

We can obtain that

$$H = \sum_{\mathbf{x}}\left[\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^\top\left[\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right] =$$

$$= \sum_{\mathbf{x}}\frac{\partial\mathbf{W}}{\partial\mathbf{p}}^\top\nabla I^\top\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \frac{\partial I}{\partial\mathbf{x}} \\ \frac{\partial I}{\partial\mathbf{y}} \end{bmatrix}\begin{bmatrix} \frac{\partial I}{\partial\mathbf{x}} & \frac{\partial I}{\partial\mathbf{y}} \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} =$$

$$= \sum\begin{bmatrix} \frac{\partial I}{\partial\mathbf{x}}\frac{\partial I}{\partial\mathbf{c}} & \frac{\partial I}{\partial\mathbf{x}}\frac{\partial I}{\partial\mathbf{y}} \\ \frac{\partial I}{\partial\mathbf{y}}\frac{\partial I}{\partial\mathbf{x}} & \frac{\partial I}{\partial\mathbf{y}}\frac{\partial I}{\partial\mathbf{y}} \end{bmatrix} = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix},$$

Now we can express the first equation as:

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix}\Delta\mathbf{p} = \sum_{\mathbf{x}}\left[\nabla I\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]\left[T(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p}))\right]$$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \sum\begin{bmatrix} I_x \\ I_y \end{bmatrix}\left[T(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p}))\right].$$

Note that the template $T(x)$ is an extracted sub-region of the image at $t = 1$ and $I(x)$ is the image at $t = 2$. Hence,

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \sum\begin{bmatrix} I_x \\ I_y \end{bmatrix}\left[-I_t\right]$$

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

□