Computer Vision AI – Practice 2
Harris Corner Detector

April 3, 2014

This assignment is a preparatory work for those who did not take the IMS-AI course in FALL 2013. The students will practice some basic computer vision tasks in this assignment. Students are not expected to submit any report for this practice.

1 Harris Corner Detector

In this section, a derivation of the Harris Corner Detector is presented:

Given a shift $(\Delta x, \Delta y)$ at a point (x, y), the auto-correlation function is defined as,

$$c(x,y) = \sum_{(u,v)\in W(x,y)} w(u,v)(I(u,v) - I(u+\Delta x, v+\Delta y))^2 \tag{1}$$

where $W(x,y)$ is a window centered at point (x,y) and $w(u,v)$ is a Gaussian function. For simplicity, from now on, $\sum\limits_{(u,v)\in W(x,y)}$ will be referred to as $\sum\limits_{W}$ Approximating the shifted function by the first-order Taylor expansion we get:

$$\begin{aligned} I(u+\Delta x, v+\Delta y) &\approx I(u,v) + I_x(u,v)\Delta x + I_y(u,v)\Delta y & (2) \\ &= I(u,v) + [I_x(u,v)\ I_y(u,v)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} & (3) \end{aligned}$$

where I_x and I_y are partial derivatives of $I(x,y)$. Therefore, the auto-correlation function can be written as:

$$\begin{aligned} c(x,y) &= \sum_W (I(u,v) - I(u+\Delta x, v+\Delta y))^2 & (4) \\ &\approx \sum_W ([I_x(u,v)\ I_y(u,v)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix})^2 & (5) \\ &= [\Delta x \Delta y] Q(x,y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} & (6) \end{aligned}$$

where $Q(x, y)$ is given by:

$$Q(x,y) = \sum_W \begin{bmatrix} I_x(x,y)^2 & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y(x,y)^2 \end{bmatrix} \quad (7)$$

$$= \begin{bmatrix} \sum\limits_W I_x(x,y)^2 & \sum\limits_W I_x(x,y)I_y(x,y) \\ \sum\limits_W I_x(x,y)I_y(x,y) & \sum\limits_W I_y(x,y)^2 \end{bmatrix} \quad (8)$$

$$= \begin{bmatrix} A & B \\ B & C \end{bmatrix} \quad (9)$$

The 'cornerness' $H(x, y)$ is defined by the two eigenvalues of $Q(x, y)$, λ_1 and λ_2:

$$H = \lambda_1\lambda_2 - 0.04(\lambda_1 + \lambda_2)^2 \quad (10)$$

$$= det(Q) - 0.04(trace(Q))^2 \quad (11)$$

$$= (AC - B^2) - 0.04(A + C)^2 \quad (12)$$

In this section, you are going to implement the last equation to measure H and use it to get the corners in an image. For that purpose, you need to compute the elements of Q, i.e. A, B and C. To do that, you need to know that I_x is the smoothed derivative of the image, it can be obtained by convolving the first order Gaussian derivative, Gd , with the image I along the x-direction. Later, $A = \sum\limits_W I_x(x, y)^2$, can be obtained by squaring I_x then convolving it with a Gaussian, G. Similarly, B and C can be obtained. For example, to get C, you need to convolve the image with Gd along the y-direction (to obtain I_y), raise it to the square, then convolve it with G. At this point, you can compute H. The corners points are the local maxima of H. Therefore, in your function you should check for every point in H, if it is greater than all its neighbours (in an $n \times n$ window centered around this point) and if it is greater than the user-defined threshold, then it is a corner point. Your function should return the H matrix, the rows of the detected corner points r, and the columns of those points c - so the first corner is given by $(r(1), c(1))$. Your function should also plot three figures, the computed image derivatives I_x and I_y, and the original image with the corner points plotted on it.