①

ASSIGNMENT - 2

By
ASWINI PURNIMA SREEPADA
ID : 002647201

PART A :

Q1 :

Canny Edge Detection :

An edge can be defined as an image contour across which the image's brightness or hue changes abruptly in magnitude or in the rate of change in the magnitude. The mathematical discontinuities in an image are modeled as edges.

From the GSU Buildings, I am considering Adelhold building to find edges from the image patch $\{13, 30\}$ defined as :-

$$\begin{bmatrix} 179 & 173 & 174 & 181 & 167 \\ 104 & 96 & 92 & 110 & 108 \\ 126 & 108 & 87 & 112 & 134 \\ 141 & 129 & 102 & 121 & 137 \\ 134 & 152 & 110 & 126 & 167 \end{bmatrix}$$ 5x5 Patch.

The above patch is smoothened for abnormal values using the Gaussian filter :-

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{\left(\frac{-(x^2+y^2)}{2\sigma^2}\right)}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x^2)}{2\sigma^2}} \times \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-y^2}{2\sigma^2}}$$

Let us consider $x = 110$ and $y = 87$

$\sigma = 3.2 \quad \pi = 3.14$

$$= 4.976 \times e^{\left(\frac{-(110)^2+(87)^2}{2(3.2)^2}\right)}$$

$$= 4.976 \times e^{-906.68}$$

$$= 4.976 \times 0.012$$

$$= 0.0597124$$

We use the gradient of the gaussian function and find its derivatives:

$$\nabla(G * I)$$

The magnitude and Orientation are calculated using the above derivative

Magnitude: $|\nabla(G * I)|$

Orientation is given by $\bar{N} = \frac{\nabla(g * I)}{|\nabla(g * I)|}$

We consider the direction of the gradient to identify the pixels and perform Non-maximum suppression.

The gradient orientation is computed using the below formula:

Let the angle of the projection be 'θ', then:

$$\tan^2\theta = \frac{Y\text{coordinate}}{X\text{coordinate}}$$

$$\theta = \tan^{-2}\left(\frac{Y\text{coord}}{X\text{coord}}\right) = \tan^{-2}\left(\frac{87}{110}\right) = 0.447 \text{ radians}$$

$$= 25.6^{\circ}$$

From the above equations, we get the pixel values of detected edges as follows:

(800, 600) are the exact pixel coordinates where the edge was defined.

Refer to Asst2_PartB_Q3_Q4.APS.mlx for additional details.

Q2:

Harris Corner Detection:

This corner detection is based on the idea that if there are two windows situated in the same region, a small region around the feature shows a noticeable change in intensity

This is implemented as follows:

→ Calculating gradient in the direction of x and y axis.

→ Noting the number of frames per second.

→ Setting a maximum output window by applying a threshold on the number of dimensions to 3.

Let us define the change function as: $c(u,v)$ as the sum of square differences where $(u,v) = (x,y)$ coordinates of any pixel in the considered 5×5 patch window and the defined intensity of pixels.

$$c(u,v) = \sum_{x,y} w(x,y) \left[I(x+(u,y+v) - I(x,y) \right]^2$$

Let our Summation matrix be M

$$c(u,v) \approx [u \; v] \left[\sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right] \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum w(x,y) \left(\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right)$$

(8) Let the Eigen values of M be λ_1 and λ_2
Using these Eigen values, we find out if the detected ~~corner~~ region is a corner or a flat surface.

(1) If λ_1 and λ_2 are small, the region is flat
(2) If $\lambda_1 >> \lambda_2$, the region is an edge
(3) If λ_1, λ_2 are large and $\lambda_1 \approx \lambda_2$, the region is a corner

The considered region of interest is given by the patch {90, 30} of considered image.

$$\begin{bmatrix} 37 & 101 & 92 & 29 & 5 \\ 16 & 44 & 105 & 87 & 13 \\ 60 & 69 & 102 & 95 & 61 \\ 104 & 87 & 80 & 73 & 66 \\ 89 & 73 & 62 & 66 & 70 \end{bmatrix}_{5\times5 \text{ patch \# } \{20,3\}}$$

The Eigen values for the following patch are:

$\lambda_1 = 0.7395$ $\lambda_3 = 1.1306$
$\lambda_2 = 0.8174$ $\lambda_4 = 1.1375$

We observe that the values $\lambda_3 \approx \lambda_4$ implying that a corner was detected in the considered region.

The combination and plotting of the corresponding eigen vectors gives the corner at the below coordinates at (0.13, 0.94)

Please refer to Asst2_PartB_Q3_Q4_APS.mlx for additional details.