Eunice Olorunshola

4/28/2022

CSC 8830 Computer Vision

Homework Assignment 2

1. Pick a region of interest in the image making sure there is an EDGE in that region. Pick a 5 x 5 image patch in that region that constitutes the edge. Perform the steps of CANNY EDGE DETECTION manually and note the pixels that correspond to the EDGE.

Notes: I used lecture 07_edge as reference to help me complete number 1

According to lecture07_edge the canny edge detector follows 5 steps: Noise reduction, Calculate the gradient, non-maximum suppression ,and Hysteresis Thresholding.

Image Coordinate:

$$I = \begin{bmatrix} 212 & 195 & 167 & 202 & 182 \\ 221 & 198 & 164 & 121 & 44 \\ 137 & 62 & 22 & 132 & 38 \\ 220 & 184 & 134 & 218 & 185 \\ 190 & 172 & 134 & 187 & 105 \end{bmatrix}$$

Noise Reduction

This step filters out any noise by using the Gaussian filter with a kernel of size 5 x5

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Convolution of the image by the Gaussian filter with a kernel of size 5 x 5 by multiplying the image coordinates matrix by the kernel size 5 x 5 to get the filtered matrix of 9 x 9

$$I = \begin{bmatrix} 212 & 195 & 167 & 202 & 182 \\ 221 & 198 & 164 & 121 & 44 \\ 137 & 62 & 22 & 132 & 38 \\ 220 & 184 & 134 & 218 & 185 \\ 190 & 172 & 134 & 187 & 105 \end{bmatrix} \times \ K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Conv2(I, K) =

$$\begin{bmatrix} 2.6667 & 7.7862 & 13.6730 & 18.2075 & 20.1950 & 17.5849 & 12.9057 & 7.1195 \\ 8.1132 & 24.9560 & 45.2327 & 58.6855 & 61.5031 & 51.4340 & 35.8616 & 18.0126 \\ 13.9497 & 43.8491 & 78.1258 & 100.7421 & 102.9811 & 83.8553 & 56.7547 & 28.2390 \\ 18.4969 & 56.9748 & 99.8239 & 127.2264 & 129.3333 & 105.3585 & 71.6918 & 35.3774 \\ 20.4591 & 61.5912 & 106.3585 & 135.3082 & 139.5283 & 115.5975 & 80.8050 & 40.6226 \\ 17.9245 & 54.8365 & 94.6981 & 121.3899 & 127.1321 & 108.8428 & 77.3836 & 39.5660 \\ 13.2327 & 41.0566 & 71.6541 & 93.5786 & 99.8428 & 86.6667 & 62.6981 & 32.3774 \\ 7.5472 & 22.9308 & 40.6792 & 53.4591 & 57.2767 & 49.1635 & 34.8679 & 18.0440 \\ 2.3899 & 6.9434 & 11.9874 & 11.9119 & 16.9560 & 14.0566 & 9.6918 & 4.9937 \end{bmatrix}$$

Patch the convolution of the image with a kernel of size 9 x 9 to a kernel size 5 x 5:

$$I_{smoothened} = \begin{bmatrix} 135.3082 & 139.5283 & 115.5975 & 80.8050 & 40.6226 \\ 121.3899 & 127.1321 & 108.8428 & 77.3836 & 39.5660 \\ 93.5786 & 99.8428 & 86.6667 & 62.6981 & 32.3774 \\ 53.4591 & 57.2767 & 49.1635 & 34.8679 & 18.0440 \\ 11.9119 & 16.9560 & 14.0566 & 9.6918 & 4.9937 \end{bmatrix}$$

Calculate the gradient

This step is to find the edge gradient and direction for each pixel. By using the Sobel operator that is based on 3 x3 filter that calculates the x and y component of the gradient.

The first step is to apply a pair of convolution masks in x and y directions :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Convolution masks in x direction multiply it with the convolution image that is patched with a kernel size 5 x 5 to get the convolution by image by the vertical filter :

$$I_{smoothened} = \begin{bmatrix} 135.3082 & 139.5283 & 115.5975 & 80.8050 & 40.6226 \\ 121.3899 & 127.1321 & 108.8428 & 77.3836 & 39.5660 \\ 93.5786 & 99.8428 & 86.6667 & 62.6981 & 32.3774 \\ 53.4591 & 57.2767 & 49.1635 & 34.8679 & 18.0440 \\ 11.9119 & 16.9560 & 14.0566 & 9.6918 & 4.9937 \end{bmatrix} \times G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\Downarrow$$

$$= \begin{bmatrix} 57.1926 & 84.8823 & 227.8140 & 294.5662 & 158.5019 \\ 166.3315 & 158.6948 & 227.0316 & 277.6913 & 152.2888 \\ 274.5378 & 269.0741 & 274.1264 & 267.0278 & 157.3424 \\ 328.4337 & 320.3291 & 294.9340 & 241.2712 & 151.6002 \\ 166.0472 & 157.3100 & 142.7275 & 114.4725 & 71.3715 \end{bmatrix}$$

Convolution masks in y direction multiply it with the convolution image that is patched with a kernel size 5 x 5 to get the convolution by image by the vertical filter :

$I_{smoothened}$ =

$$\begin{bmatrix} 135.3082 & 139.5283 & 115.5975 & 80.8050 & 40.6226 \\ 121.3899 & 127.1321 & 108.8428 & 77.3836 & 39.5660 \\ 93.5786 & 99.8428 & 86.6667 & 62.6981 & 32.3774 \\ 53.4591 & 57.2767 & 49.1635 & 34.8679 & 18.0440 \\ 11.9119 & 16.9560 & 14.0566 & 9.6918 & 4.9937 \end{bmatrix} \times G_y \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 71.2304 & 147.6135 & 172.6037 & 177.1485 & 177.6167 \\ 82.4103 & 109.0194 & 149.3749 & 164.6755 & 163.6608 \\ 85.3852 & 96.5442 & 122.2917 & 141.4989 & 137.1122 \\ 86.6934 & 92.3900 & 107.6096 & 121.3685 & 116.9328 \\ 83.4469 & 89.2211 & 108.0406 & 120.6215 & 115.6709 \end{bmatrix}$$

Find the gradient strength once x and y components of the gradients is calculated, find the magnitude of the vertical and horizontal filter:

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

$$G = \sqrt{{G_{57.1926}}^2 + {G_{71.2304}}^2} = 91$$

$$G = = \sqrt{{G_{84.8823}}^2 + {G_{147.6135}}^2} = 170$$

$$G = \sqrt{{G_{227.8140}}^2 + {G_{172.6037}}^2} = 285$$

$$G = \sqrt{{G_{294.5662}}^2 + {G_{177.1485}}^2} = 343$$

$$G = \sqrt{{G_{158.5019}}^2 + {G_{177.6167}}^2} = 238$$

$$G = \sqrt{G_{166.3315}{}^{2} + G_{82.4103}{}^{2}} = 185$$

$$G = \sqrt{G_{158.6948}{}^{2} + G_{109.0194}{}^{2}} = 192$$

$$G = \sqrt{G_{227.0316}{}^{2} + G_{149.3749}{}^{2}} = 271$$

$$G = \sqrt{G_{227.6913}{}^{2} + G_{164.6755}{}^{2}} = 281$$

$$G = \sqrt{G_{152.2888}{}^{2} + G_{163.6608}{}^{2}} = 224$$

$$G = \sqrt{G_{274.5378}{}^{2} + G_{85.3852}{}^{2}} = 287$$

$$G = \sqrt{G_{269.0741}{}^{2} + G_{96.5442}{}^{2}} = 285$$

$$G = \sqrt{G_{274.1264}{}^{2} + G_{122.2917}{}^{2}} = 300$$

$$G = \sqrt{G_{267.0278}{}^{2} + G_{141.4989}{}^{2}} = 302$$

$$G = \sqrt{G_{157.3424}{}^{2} + G_{137.1122}{}^{2}} = 208$$

$$G = \sqrt{G_{328.4337}{}^{2} + G_{86.6934}{}^{2}} = 339$$

$$G = \sqrt{{G_{320.3291}}^2 + {G_{92.3900}}^2} = 333$$

$$G = \sqrt{{G_{294.9340}}^2 + {G_{107.6096}}^2} = 313$$

$$G = \sqrt{{G_{241.2712}}^2 + {G_{121.3685}}^2} = 270$$

$$G = \sqrt{{G_{151.6002}}^2 + {G_{116.9328}}^2} = 191$$

$$G = \sqrt{{G_{166.0472}}^2 + {G_{83.4469}}^2} = 185$$

$$G = \sqrt{{G_{157.3100}}^2 + {G_{89.2211}}^2} = 180$$

$$G = \sqrt{{G_{142.7275}}^2 + {G_{108.0406}}^2} = 179$$

$$G = \sqrt{{G_{114.4725}}^2 + {G_{120.6215}}^2} = 166$$

$$= \sqrt{{G_{71.3715}}^2 + {G_{115.6709}}^2} = 135$$

$$G = \sqrt{{G_x}^2 + {G_y}^2} \quad = \quad \begin{bmatrix} 91 & 170 & 285 & 343 & 238 \\ 185 & 192 & 271 & 281 & 224 \\ 287 & 285 & 300 & 302 & 208 \\ 339 & 333 & 313 & 270 & 191 \\ 185 & 180 & 179 & 166 & 135 \end{bmatrix}$$

Calculate the gradient direction , which is always perpendicular to the edges and it is rounded to one of the four angles that represents the vertical , horizontal filter and the two diagonal directions.

The vertical and horizontal filter is calculated with the gradient direction :

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

= $\arctan\left(\frac{G_{71.2304}}{G_{57.1926}}\right)$ = 51

$\arctan\left(\frac{G_{147.6135}}{G_{84.8823}}\right)$ = 60

$\arctan\left(\frac{G_{172.6037}}{G_{115.5975}}\right)$ = 37

$\arctan\left(\frac{G_{177.1485}}{G_{294.5662}}\right)$ = 31

$\arctan\left(\frac{G_{177.6167}}{G_{158.5019}}\right)$ = 48

$\arctan\left(\frac{G_{82.4103}}{G_{166.3315}}\right)$ = 26

$\arctan\left(\frac{G_{109.0194}}{G_{158.6948}}\right)$ = = 35

$\arctan\left(\frac{G_{149.3749}}{G_{227.0316}}\right)$ = 33

$\arctan\left(\frac{G_{164.6755}}{G_{277.6913}}\right)$ = 31

$$\arctan\left(\frac{G_{163.6608}}{G_{152.2888}}\right) = 47$$

$$\arctan\left(\frac{G_{85.3852}}{G_{274.5378}}\right) = 17$$

$$\arctan\left(\frac{G_{96.5442}}{G_{269.0741}}\right) = 19$$

$$\arctan\left(\frac{G_{122.2917}}{G_{274.1264}}\right) = = 24$$

$$\arctan\left(\frac{G_{141.4989}}{G_{267.0278}}\right) = 28$$

$$\arctan\left(\frac{G_{137.1122}}{G_{157.3424}}\right) = 13$$

$$\arctan\left(\frac{G_{86.6934}}{G_{328.4337}}\right) = 15$$

$$\arctan\left(\frac{G_{92.3900}}{G_{320.3291}}\right) = 16$$

$$\arctan\left(\frac{G_{107.6096}}{G_{294.9340}}\right) = 20$$

$$\arctan\left(\frac{G_{121.3685}}{G_{241.2712}}\right) = 27$$

$$\arctan\left(\frac{G_{116.9328}}{G_{151.6002}}\right) = 37$$

$$\arctan\left(\frac{G_{83.4469}}{G_{166.0472}}\right) = 27$$

$$\arctan\left(\frac{G_{89.2211}}{G_{157.3100}}\right) = 30$$

$$\arctan\left(\frac{G_{108.0406}}{G_{142.7275}}\right) = 37$$

$$\arctan\left(\frac{G_{120.6215}}{G_{114.4725}}\right) = 47$$

$$\arctan\left(\frac{G_{115.6709}}{G_{71.3715}}\right) = 58$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) = \begin{bmatrix} 51 & 60 & 37 & 31 & 48 \\ 26 & 35 & 33 & 31 & 47 \\ 17 & 19 & 24 & 28 & 13 \\ 15 & 16 & 20 & 27 & 37 \\ 27 & 30 & 37 & 47 & 58 \end{bmatrix}$$

Non – maximum Suppression

This step is used to find the largest edge after calculating the gradient , this step is also used to suppress all the gradient values by setting them to 0 except the local maxima. Every pixel is checked if it is a local maximum in the direction of the gradient. And removes the pixels that are not considered to be part of an edge

At $angle\ 0\ ,45, 90, and\ 135$

$$= \begin{bmatrix} 3.7712 & 15.2560 & 33.7566 & 54.3481 & 70.5850 \\ 15.7046 & 57.7800 & 118.9907 & 182.4592 & 226.0160 \\ 34.7268 & 117.2569 & 210.2137 & 281.8186 & 318.7820 \\ 55.4896 & 177.6523 & 278.4849 & 302.3481 & 276.5146 \\ 71.6986 & 221.5360 & 320.5176 & 292.1199 & 177.8531 \end{bmatrix}$$

Hysteresis Thresholding

This is the last step of canny edge detection , which uses two thresholds (low and high)

- If a pixel gradient is above high than the pixel is declared as a strong edge pixel
- If a pixel gradient is below low, then the pixel is declared as a non-edge pixel

Thresholds values low = 0.075

Threshold value high – 0.175

Find the maximum of the gradient magnitude and declare it as an edge pixel :

Edge pixels:

$$\begin{bmatrix} 3.7712 & 9.0884 & 15.2560 & 22.6070 & 29.8492 \\ 3.3799 & 13.6252 & 19.8765 & 20.9372 & 32.1957 \\ 5.8843 & 12.0749 & 11.3698 & 10.6253 & 7.7373 \\ 12.5788 & 17.3407 & 19.0138 & 16.5867 & 11.7022 \\ 7.9610 & 14.1337 & 6.0974 & 16.0120 & 24.0813 \end{bmatrix}$$

2. Pick a region of interest in the image making sure there is a CORNER in that region. Pick a 5 x 5 image patch in that region that constitutes the corner. Perform the steps of HARRIS CORNER DETECTION manually and note the pixels that correspond to the CORNER.

Notes: I used lecture 07_part2_corner as reference to help me complete number 2

According to lecture07_part2_corner the Harris corner detector follows 5 steps: Compute Gaussian derivatives at each pixel, Compute second moment matrix M in a Gaussian window around each pixel, Compute corner response function R, Threshold R, and find local maxima of response function(non-maximum suppression)

Image Coordinates

$$I = \begin{bmatrix} 30 & 18 & 30 & 6 & 8 \\ 9 & 10 & 30 & 12 & 4 \\ 8 & 13 & 15 & 1 & 14 \\ 5 & 7 & 18 & 5 & 9 \\ 2 & 3 & 2 & 15 & 20 \end{bmatrix}$$

Filter the Image

Compute x and y Gaussian derivatives at each pixel

$I_x = G_\sigma^x \times I$

Gaussian derivative of x is multiplied by the image coordinates using Sobel filter:

$$G_\sigma^x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 30 & 18 & 30 & 6 & 8 \\ 9 & 10 & 30 & 12 & 4 \\ 8 & 13 & 15 & 1 & 14 \\ 5 & 7 & 18 & 5 & 9 \\ 2 & 3 & 2 & 15 & 20 \end{bmatrix} = \begin{bmatrix} 46 & 39 & -2 & -10 & 0 \\ 51 & 65 & 6 & -45 & -29 \\ 43 & 58 & -10 & -1 & -9 \\ 30 & 37 & 2 & 3 & 4 \\ 13 & 13 & 22 & 27 & -35 \end{bmatrix}$$

$$I_x = \begin{bmatrix} 46 & 39 & -2 & -10 & 0 \\ 51 & 65 & 6 & -45 & -29 \\ 43 & 58 & -10 & -1 & -9 \\ 30 & 37 & 2 & 3 & 4 \\ 13 & 13 & 22 & 27 & -35 \end{bmatrix}$$

Gaussian derivative of y is the transpose of x and is multiplied by the image coordinates :

$I_y = G_\sigma^y \times I$

$$G_\sigma^y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 30 & 18 & 30 & 6 & 8 \\ 9 & 10 & 30 & 12 & 4 \\ 8 & 13 & 15 & 1 & 14 \\ 5 & 7 & 18 & 5 & 9 \\ 2 & 3 & 2 & 15 & 20 \end{bmatrix} =$$

$$I_y = \begin{bmatrix} 28 & 59 & 82 & 58 & 20 \\ -49 & -47 & -40 & -19 & 7 \\ -11 & -22 & -34 & -21 & 3 \\ -22 & -39 & -22 & 21 & 26 \\ -17 & -37 & -48 & -37 & -23 \end{bmatrix}$$

Compute products of derivatives at every pixel(square the derivatives)

$I_{x^2} = I_x \times I_x$

$$= \begin{bmatrix} 46 & 39 & -2 & -10 & 0 \\ 51 & 65 & 6 & -45 & -29 \\ 43 & 58 & -10 & -1 & -9 \\ 30 & 37 & 2 & 3 & 4 \\ 13 & 13 & 22 & 27 & -35 \end{bmatrix} \times \begin{bmatrix} 46 & 39 & -2 & -10 & 0 \\ 51 & 65 & 6 & -45 & -29 \\ 43 & 58 & -10 & -1 & -9 \\ 30 & 37 & 2 & 3 & 4 \\ 13 & 13 & 22 & 27 & -35 \end{bmatrix}$$

$$= I_{x^2} = \begin{bmatrix} 3719 & 3843 & 142 & -2243 & -1153 \\ 4192 & 4520 & -500 & -4359 & -1104 \\ 4359 & 4713 & 162 & -3276 & -1281 \\ 3495 & 3854 & 236 & -1850 & -1219 \\ 2562 & 3172 & -884 & -1601 & 758 \end{bmatrix}$$

$I_{y^2} = I_y \times I_y$

$$\begin{bmatrix} 28 & 59 & 82 & 58 & 20 \\ -49 & -47 & -40 & -19 & 7 \\ -11 & -22 & -34 & -21 & 3 \\ -22 & -39 & -22 & 21 & 26 \\ -17 & -37 & -48 & -37 & -23 \end{bmatrix} \times \begin{bmatrix} 28 & 59 & 82 & 58 & 20 \\ -49 & -47 & -40 & -19 & 7 \\ -11 & -22 & -34 & -21 & 3 \\ -22 & -39 & -22 & 21 & 26 \\ -17 & -37 & -48 & -37 & -23 \end{bmatrix}$$

$$= I_{y^2} = \begin{bmatrix} -4625 & -5347 & -5088 & -741 & 2267 \\ 1670 & 490 & -696 & -1767 & -2084 \\ 1555 & 1631 & 1452 & -58 & -1091 \\ 143 & -1022 & -1606 & -784 & -761 \\ 3070 & 3716 & 3636 & 799 & -1176 \end{bmatrix}$$

$I_{xy} = I_x \times I_y$

$$I_x = \begin{bmatrix} 46 & 39 & -2 & -10 & 0 \\ 51 & 65 & 6 & -45 & -29 \\ 43 & 58 & -10 & -1 & -9 \\ 30 & 37 & 2 & 3 & 4 \\ 13 & 13 & 22 & 27 & -35 \end{bmatrix} \times \; I_y = \begin{bmatrix} 28 & 59 & 82 & 58 & 20 \\ -49 & -47 & -40 & -19 & 7 \\ -11 & -22 & -34 & -21 & 3 \\ -22 & -39 & -22 & 21 & 26 \\ -17 & -37 & -48 & -37 & -23 \end{bmatrix}$$

$$I_{xy} = \begin{bmatrix} -381 & 1315 & 2500 & 1759 & 927 \\ -340 & 2650 & 3760 & 1725 & 990 \\ -1353 & 403 & 2000 & 1914 & 1417 \\ -1129 & -278 & 654 & 910 & 851 \\ -514 & -86 & 884 & 1907 & 1924 \end{bmatrix}$$

Compute sums of the products of derivatives at every pixel

$S_{x^2} = G_{\sigma'} \times I_{x^2}$

$$G_{\sigma'} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \times I_{x^2} = \begin{bmatrix} 3719 & 3843 & 142 & -2243 & -1153 \\ 4192 & 4520 & -500 & -4359 & -1104 \\ 4359 & 4713 & 162 & -3276 & -1281 \\ 3495 & 3854 & 236 & -1850 & -1219 \\ 2562 & 3172 & -884 & -1601 & 758 \end{bmatrix}$$

S_{x^2} = -8955

$S_{y^2} = G_{\sigma'} \times I_{y^2}$

$$G_{\sigma'} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \times I_{y^2} = \begin{bmatrix} -4625 & -5347 & -5088 & -741 & 2267 \\ 1670 & 490 & -696 & -1767 & -2084 \\ 1555 & 1631 & 1452 & -58 & -1091 \\ 143 & -1022 & -1606 & -784 & -761 \\ 3070 & 3716 & 3636 & 799 & -1176 \end{bmatrix}$$

S_{y^2} = 411

$S_{xy} = G_{\sigma'} \times I_{xy}$

$$G_{\sigma'} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \times I_{xy} = \begin{bmatrix} -381 & 1315 & 2500 & 1759 & 927 \\ -340 & 2650 & 3760 & 1725 & 990 \\ -1353 & 403 & 2000 & 1914 & 1417 \\ -1129 & -278 & 654 & 910 & 851 \\ -514 & -86 & 884 & 1907 & 1924 \end{bmatrix}$$

S_{xy} = 12461

Compute second moment matrix M in a Gaussian window around each pixel

Now that the sum of the product of each pixel is computed , next is defining the matrix M which is the second moment matrix and is used to show how quickly the image changes and in which directions:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_{x^2} & I_x I_y \\ I_x I_y & I_{y^2} \end{bmatrix} = \text{Directions : } R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$M(x,y) = \begin{bmatrix} I_{x^2} & I_x I_y \\ I_x I_y & I_{y^2} \end{bmatrix}$$

$$M(x,y) = \begin{bmatrix} -8955 & 12461 \\ 12461 & 411 \end{bmatrix}$$

Compute corner response function R

Apply the formula: $R = \det(M) - k \times (trace(M))^2$ = $\lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$

This formula is used to determine if the current position has a corner or not

Breakdown of the formula:

$\det(M)$: is the determinant of the matrix of the 2 x 2 matrix

$$\det(M) = |M| = \begin{bmatrix} I_{x^2} & I_xI_y \\ I_xI_y & I_{y^2} \end{bmatrix} = \begin{vmatrix} a & c \\ c & b \end{vmatrix} = ab - c^2$$

k : is the constant or tunable parameter between the ranges (0.04 to 0.06)

$trace(M))^2$: trace of a square matrix is the total sum of the main diagonal and is used to calculate the trace of the matrix

$$trace(M) = \begin{bmatrix} I_{x^2} & I_xI_y \\ I_xI_y & I_{y^2} \end{bmatrix} = \begin{vmatrix} a & c \\ c & b \end{vmatrix} = a + b$$

$$R = \det(M) = |M| = \begin{bmatrix} 25150 & 2924 \\ 2924 & -15228 \end{bmatrix}$$

$R = \det(M) - k \times (trace(M))^2 =$ 30321.644, 10477.644

Threshold R

PART B: MATLAB Prototyping

3. Compare the outcome of problem (1) with MATLAB's Canny edge detection function

The outcome of problem 1 with MATLAB Canny edge detection function is that

Image coordinate step

- The image coordinates pixel was the same as what it shown for the canny edge detection function.

Noise reduction step

- The next step of noise reduction , I multiplied the gaussian filter of kernel size 5x5 by the image coordinates by hand , and the outcome on MATLAB was the same.
- After finding the values of the filtered image , the next step was to patch the image from a kernel size of 9 x 9 to a kernel size of 5 x 5. The outcome was different because when I did it on MATLAB than doing it manually it did not correctly patch the image to the correct kernel size .

Calculating the gradient step

- Calculating the gradient step required multiple parts , the first part was multiplying the patched image by the convolution masks in x direction to get the vertical filter. Doing this by hand and using the function both had the same results.
- The second part was multiplying the patched image by the convolution masks in y direction to get the horizontal filter. Doing this by hand and using the function both had the same results.
- The third part was using the gradient strength formula to find the magnitude of the vertical and horizontal filter. The outcome by hand and using canny edge detection function on MATLAB was different because the function needed a magnitude function to square the x and y direction.
- The last part was finding the gradient direction of the vertical and horizontal filter, both results was shown the same , because I had to use arctan to compute it by hand and canny edge also has an atan function.

Non maximum Suppression

- This step goes through all the points on the gradient matrix and finds the pixels with the maximum value in the edge directions.
- This step was challenging for me to do manually , because I was not correctly identifying the angles from the angle matrix for the edge direction
- Using the function on matlab made it easier because it was able to go through each pixel from left to right in the angle matrix and check if the

pixel in the same direction has a higher intensity then the current pixel. And if there is no high intensity detected it keeps the current pixel .

Hysteria Thresholding

- **This is the last step**