Sharif University of Technology

EE Department

Project Signal and System

Student Name: Mohammad Ghafourian

Student ID: 99106493

QUESTION 1:

2D Convolution

SHOW WITH EXAMPLE

We have this two matrix:

25	100	75	49	130
50	80	0	70	100
5	10	20	30	0
60	50	12	24	32
37	53	55	21	90
140	17	0	23	222

x

1	0	1
0	1	0
0	0	1

h

STEP 1 MATRIX INVERSION

1	0	1
0	1	0
0	0	1

h

0	0	1
0	1	0
1	0	1

h'

1	0	0
0	1	0
1	0	1

h''

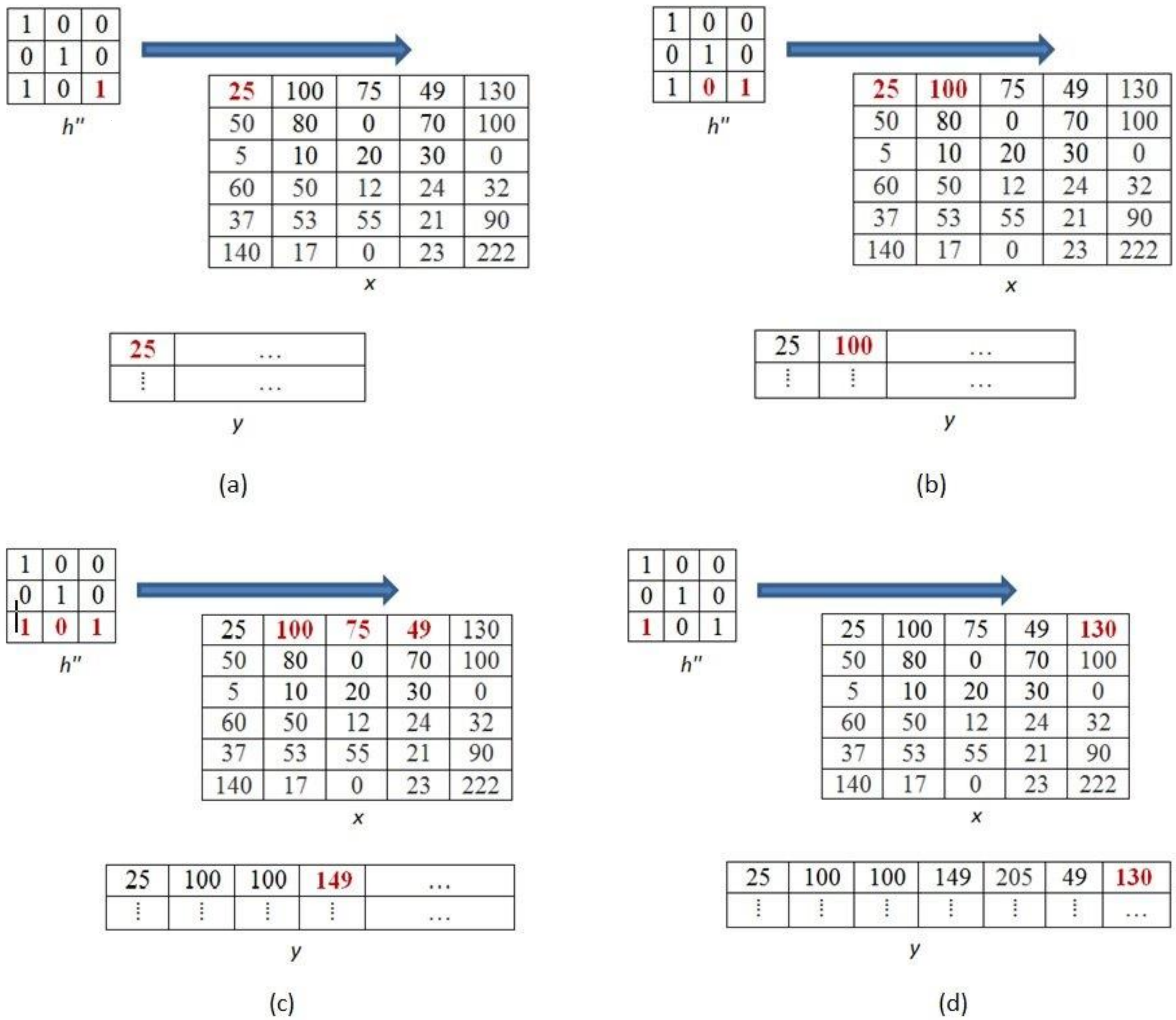
So our matrix is

$$\begin{matrix} 0 & 1 \\ 2 & 3 \end{matrix}$$

Will be

$$\begin{bmatrix} 3 & 2 \\ 1 & 0 \end{bmatrix}$$

So the conv in example will be



At the end

25	100	75	49	130
50	80	0	70	100
5	10	20	30	0
60	50	12	24	32
37	53	55	21	90
140	17	0	**23**	**222**

x

1	**0**	0
0	1	0
1	0	1

h''

25	100	100	149	205	49	130
50	105	150	225	149	200	100
5	60	130	140	165	179	130
60	55	132	174	74	94	132
37	113	147	96	189	83	90
140	54	253	145	255	137	254
0	140	54	53	78	243	90
0	0	140	17	0	**23**	…

y

$$y\,(8, 6) = 23 \times 1 + 222 \times 0 = 23 + 0 = 23$$

25	100	100	149	205	49	130
50	105	150	225	149	200	100
5	60	130	140	165	179	130
60	55	132	174	74	94	132
37	113	147	96	189	83	90
140	54	253	145	255	137	254
0	140	54	53	78	243	90
0	0	140	17	0	23	255

y

$$y\,[i, j] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h\,[m, n] \cdot x\,[i - m, j - n]$$

So we have this two matrix

$$\begin{matrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{matrix} \text{ and } \begin{matrix} 3 & 2 \\ 1 & 0 \end{matrix}$$

The answer without padding

So the firs row will be

$$\begin{matrix} 5 & 11 \end{matrix}$$

2nd row

$$\begin{matrix} 23 & 29 \end{matrix}$$

Second conv

So we have this two matrix

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \text{ and } \begin{matrix} 4 & 3 \\ 2 & 1 \end{matrix}$$

So the firs row will be

$$\begin{matrix} 23 & 33 \end{matrix}$$

2nd row

$$\begin{matrix} 53 & 63 \end{matrix}$$

Final ans is

$$\begin{matrix} 5 & 11 \\ 23 & 29 \end{matrix} + \begin{matrix} 23 & 33 \\ 53 & 63 \end{matrix} = \begin{matrix} 28 & 44 \\ 76 & 92 \end{matrix}$$

QUESTION 2

The 'kernel' for smoothing, defines the shape of the function that is used to take the average of the neighboring points. A Gaussian kernel is a kernel with the shape of a Gaussian (normal distribution) curve. Here is a standard Gaussian, with a mean of 0 and a σ (=population standard deviation) of 1.

We use imfilter order and the first input is the image and the second input is the kernel and this code smooth the image

So we need a Gaussian 3×3 kernel with matlb with this code

And it give us the with sigma=5

```
4 -     h = fspecial('gaussian',3,5)
```

Command Window

```
h =

    0.1096    0.1118    0.1096
    0.1118    0.1141    0.1118
    0.1096    0.1118    0.1096

fx >>
```

And the result is

orginal

filtered

```
clc;
clear;
close all;
h = fspecial('gaussian',3,5)
I = imread('image reference.jpg');
subplot(1,2,1)
imshow(I);
title('orginal')
MotionBlur = imfilter(I,h,'replicate');
subplot(1,2,2)
imshow(MotionBlur);
title('filtered')
```

Edge detecting

HORIZONTAL and VERTICAL:

we need this two kernels

X – Direction Kernel

-1	0	1
-2	0	2
-1	0	1

Y – Direction Kernel

-1	-2	-1
0	0	0
1	2	1

For example we have an image for vertical edge for example

100	100	200	200
100	100	200	200
100	100	200	200
100	100	200	200

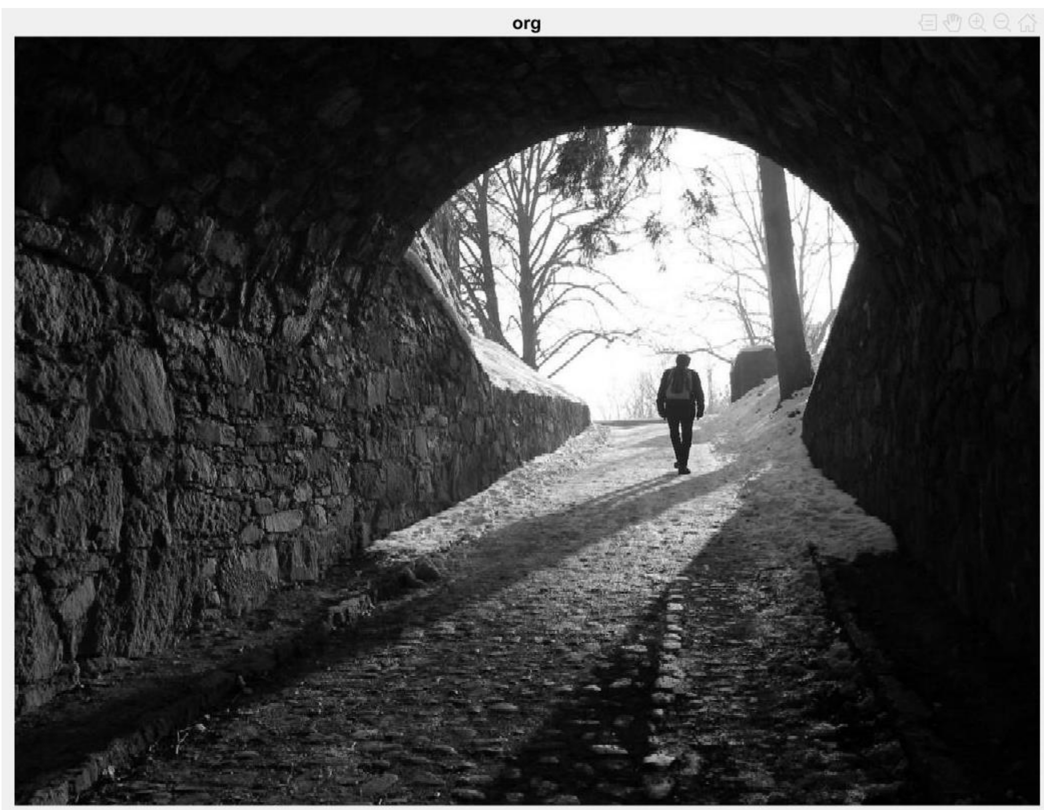
In red pixel if we convolve we have 400 that means white but if all of that area were 100 the conv answer will be zero and its show the edge

This was just example we use this two kernels

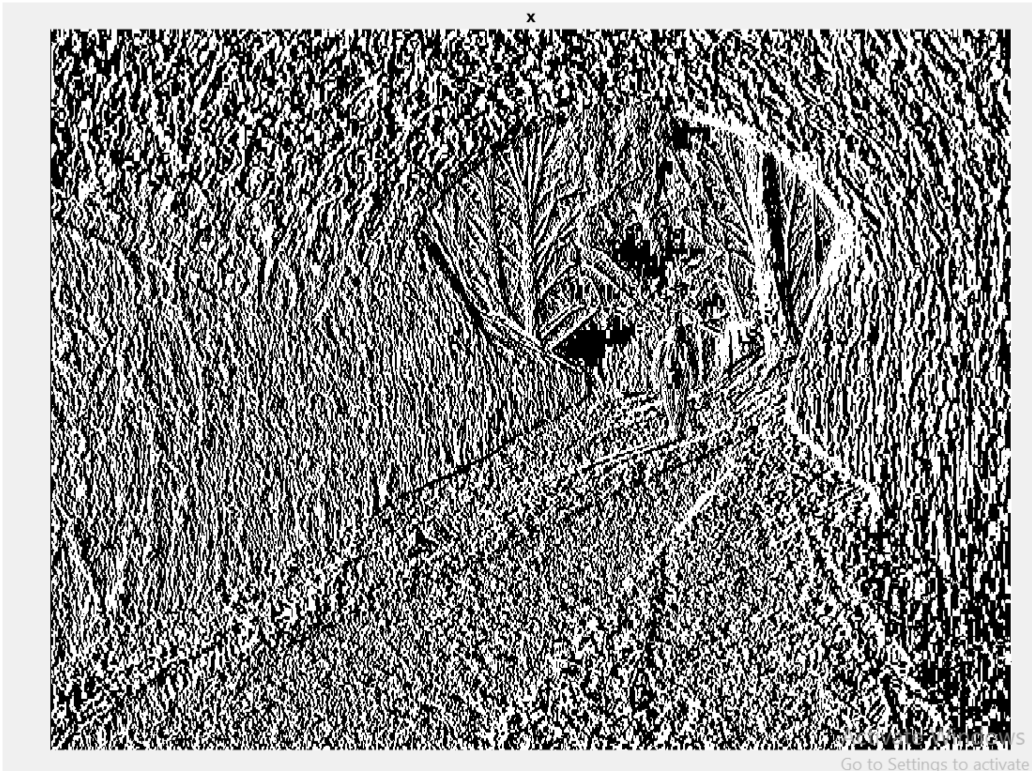
$$\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \;,\; \begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$$

So the answer and the code:

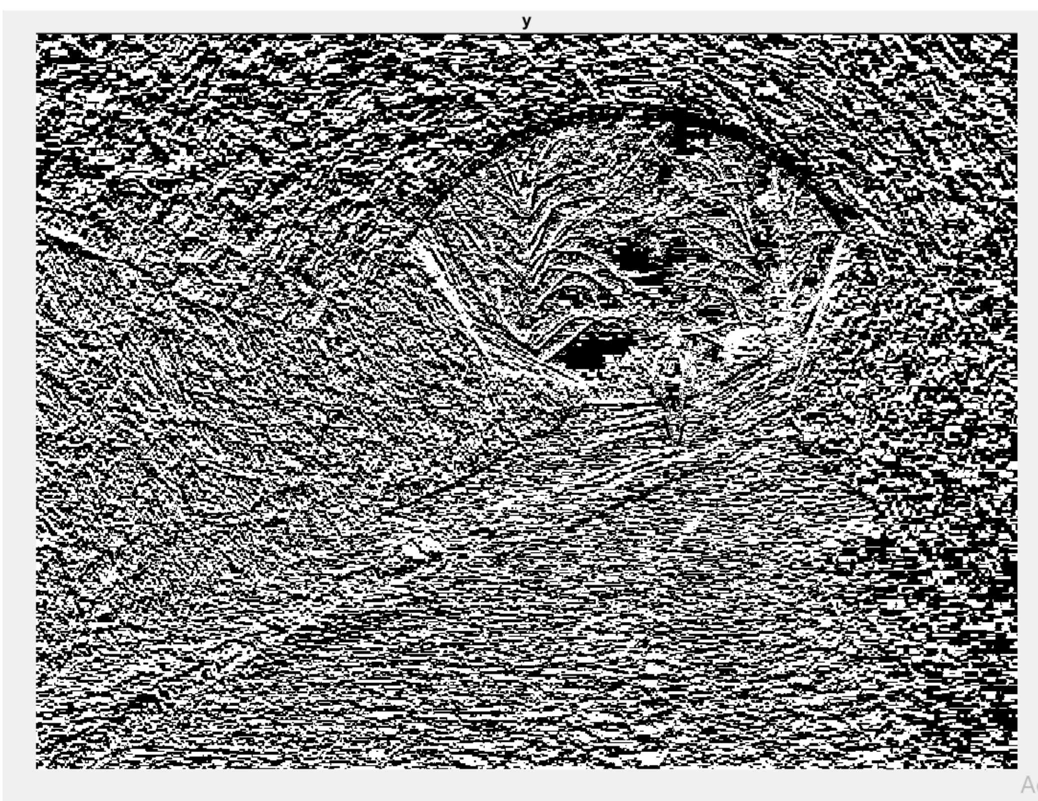
```
clc;
clear;
close all;
I = imread('image reference.jpg');
hx = [-1 0 1; -1 0 1; -1 0 1];
hy = [-1 -1 -1; 0 0 0; 1 1 1];
figure
imshow(I)
title('org')
xkernel = conv2(I,hx,'same');
figure
imshow(xkernel);
title('x')
ykernel = conv2(I,hy,'same');
figure
imshow(ykernel);
title('y')
```



Vertical



Horizontal



QUESTION 3

The Sobel operator, sometimes called the Sobel–Feldman operator or Sobel filter, is **used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasizing edges**.

We need this two kernel for sobel

approximations respectively, the computations are as follows:[2]

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

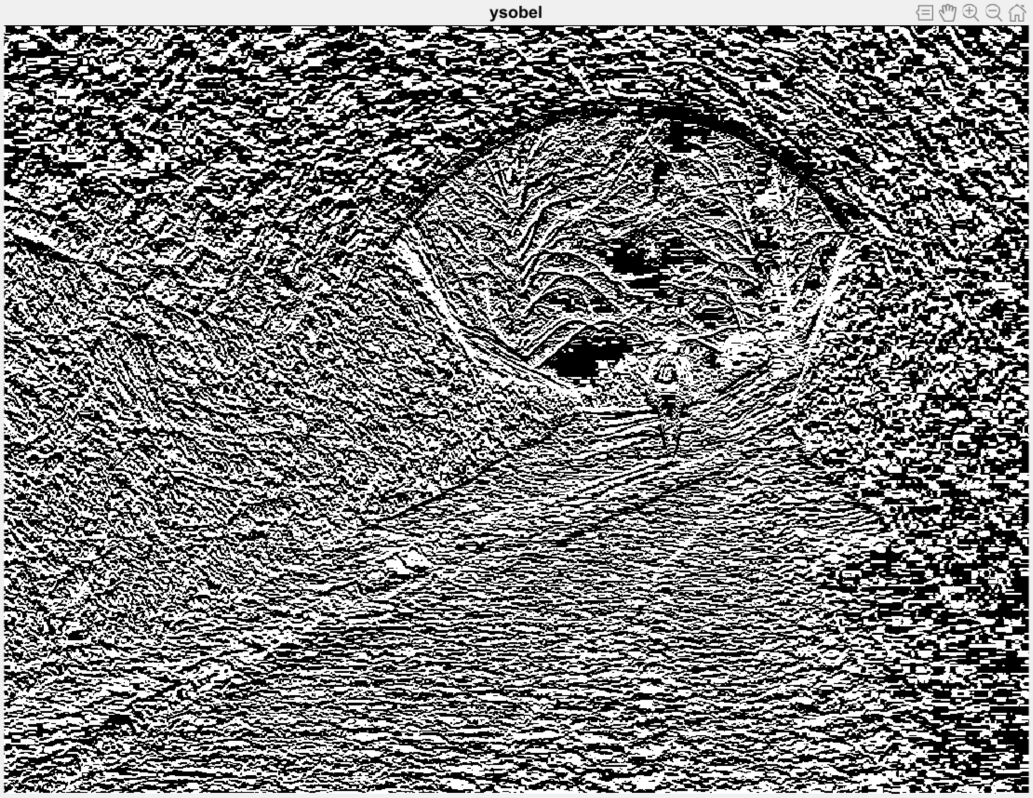
$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

like question number 2 we make our new image with the priveus code and then we find G;

```
clear;
close all;
hx = [-1 0 1; -2 0 2; -1 0 1];
hy = [-1 -2 -1; 0 0 0; 1 2 1];
I = imread('image reference.jpg');
MotionBlur1 = conv2(I,hx,'same');
figure
imshow(I)
title('org')
figure
imshow(MotionBlur1);
title('xsobel')
MotionBlur2 = conv2(I,hy,'same');
figure
imshow(MotionBlur2);
title('ysobel')
figure
final = sqrt(MotionBlur1.^2 + MotionBlur2.^2);
imshow(final,[])
title('final')
```

org

ysobel



Result: