

CS462 Image Processing

Chapter 4

Image Filter

By Dr. Paween Khoenkaw

Computer Science MJU



Image Filter



Original Image
Image Courtesy of MIT



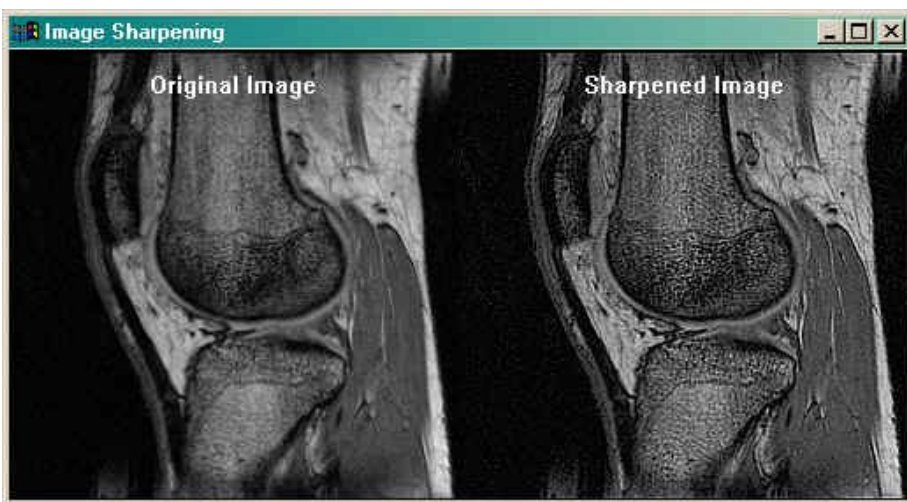
Motion Blurred Image



Blurred Image



Sharpened Image



Spatial convolution

$$\mathbf{F}(x) * \mathbf{G}(x) = \int_{-\infty}^{\infty} \mathbf{F}(\tau) \cdot \mathbf{G}(x - \tau) d\tau$$

Where

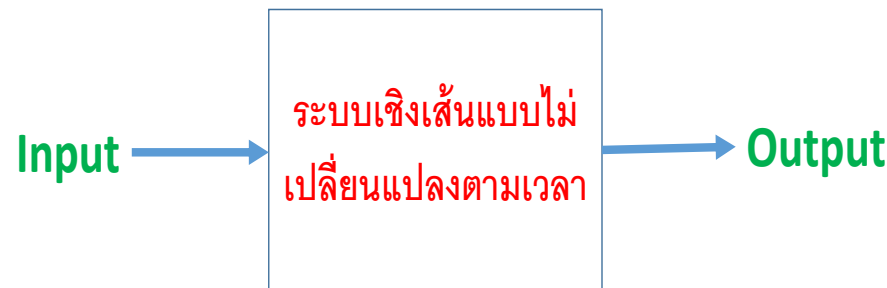
* means convolution

$\mathbf{F}$ is image

$\mathbf{G}$ is convolution kernel

ระบบเชิงเส้นที่ไม่เปลี่ยนแปลงตามเวลา

ระบบเชิงเส้นที่ไม่เปลี่ยนแปลงตามเวลา (Linear time-invariant: LTI) คือระบบการประมวลผลที่มีสัญญาณขาเข้า (input) และสัญญาณขาออก (output) มีคุณสมบัติ 2 ประการคือ เป็นเชิงเส้น และไม่เปลี่ยนแปลงตามเวลา



เป็นเชิงเส้น

หากระบบ $x_1(t)$ ทำให้เกิด $y_1(t)$ และ $x_2(t)$ ทำให้เกิด $y_2(t)$ แล้ว

$a_1x_1(t) + a_2x_2(t)$ จะได้ $a_1y_1(t) + a_2y_2(t)$; เมื่อ a_1 และ a_2 เป็นจำนวนจริง

ไม่เปลี่ยนแปลงตามเวลา

หากระบบ $x_1(t)$ ทำให้เกิด $y_1(t)$ แล้ว $x_1(t - T)$ ทำให้เกิด $y_1(t - T)$ ด้วยเช่นกัน

โดยที่ T คือเวลาที่ถูกหน่วง

Spatial convolution

F

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

G

0.25	0.25
0.25	0.25

$$\mathbf{I}(x) = \mathbf{F}(x) * \mathbf{G}(x)$$

Spatial convolution

F

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

G

0.25	0.25
0.25	0.25

I

Spatial convolution

F

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

G

0.25	0.25
0.25	0.25

I

x							

$F(\tau)$

1	2
9	10

G

0.25	0.25
0.25	0.25

•

$$I(1,1) = (1 \times 0.25) + (2 \times 0.25) + (9 \times 0.25) + (10 \times 0.25)$$

$$I(1,1) = 5.5$$

Spatial convolution

F

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

G

0.25	0.25
0.25	0.25

I

5.5							

$F(\tau)$

1	2
9	10

G

0.25	0.25
0.25	0.25

•

$$I(1,1) = (1 \times 0.25) + (2 \times 0.25) + (9 \times 0.25) + (10 \times 0.25)$$

$$I(1,1) = 5.5$$

Spatial convolution

F

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

G

0.25	0.25
0.25	0.25

I

5.5	x						

$F(\tau)$

2	3
10	11

G

0.25	0.25
0.25	0.25

•

$$I(1,1) = (2 \times 0.25) + (3 \times 0.25) + (10 \times 0.25) + (11 \times 0.25)$$

$$I(1,1) = 6.5$$

Spatial convolution

F

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

G

0.25	0.25
0.25	0.25

I

5.5	6.5						

$F(\tau)$

2	3
10	11

G

0.25	0.25
0.25	0.25

•

$$I(1,1) = (2 \times 0.25) + (3 \times 0.25) + (10 \times 0.25) + (11 \times 0.25)$$

$$I(1,1) = 6.5$$

Spatial convolution

F

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

G

0.25	0.25
0.25	0.25

I

5.5	6.5	7.5	8.5	9.5	10.5	11.5	

Spatial convolution

F

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

G

0.25	0.25
0.25	0.25

I

5.5	6.5	7.5	8.5	9.5	10.5	11.5	
x							

Spatial convolution

F

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

G

0.25	0.25
0.25	0.25

I

5.5	6.5	7.5	8.5	9.5	10.5	11.5	
13.5	14.5	15.5	16.5	17.5	18.5	19.5	

Spatial convolution

F

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

G

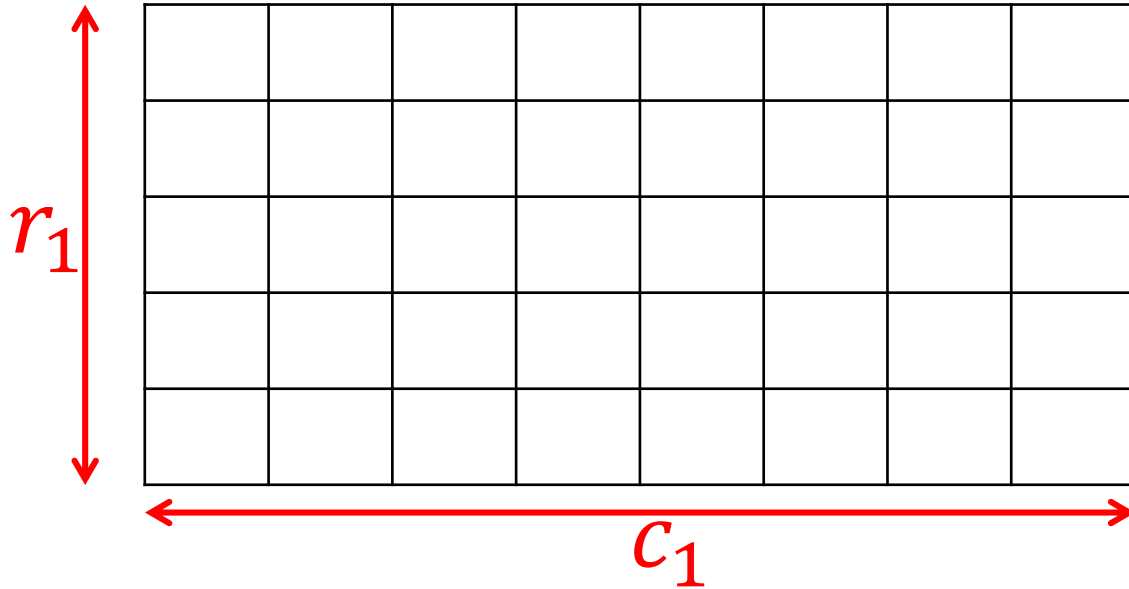
0.25	0.25
0.25	0.25

I

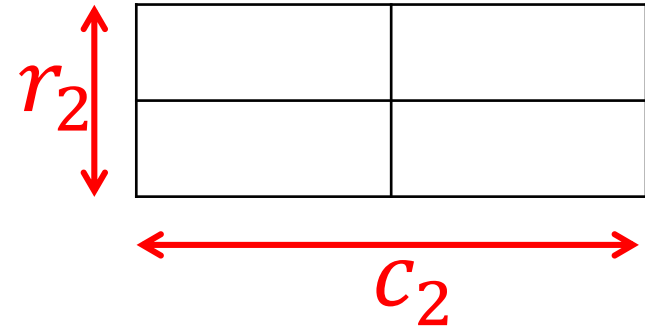
5.5	6.5	7.5	8.5	9.5	10.5	11.5	
13.5	14.5	15.5	16.5	17.5	18.5	19.5	
21.5	22.5	23.5	24.5	25.5	26.5	27.5	
29.5	30.5	31.5	32.5	33.5	34.5	35.5	
							

Spatial convolution

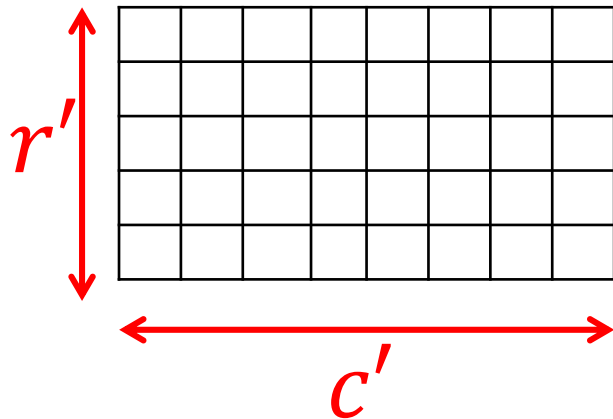
F



G



The final size



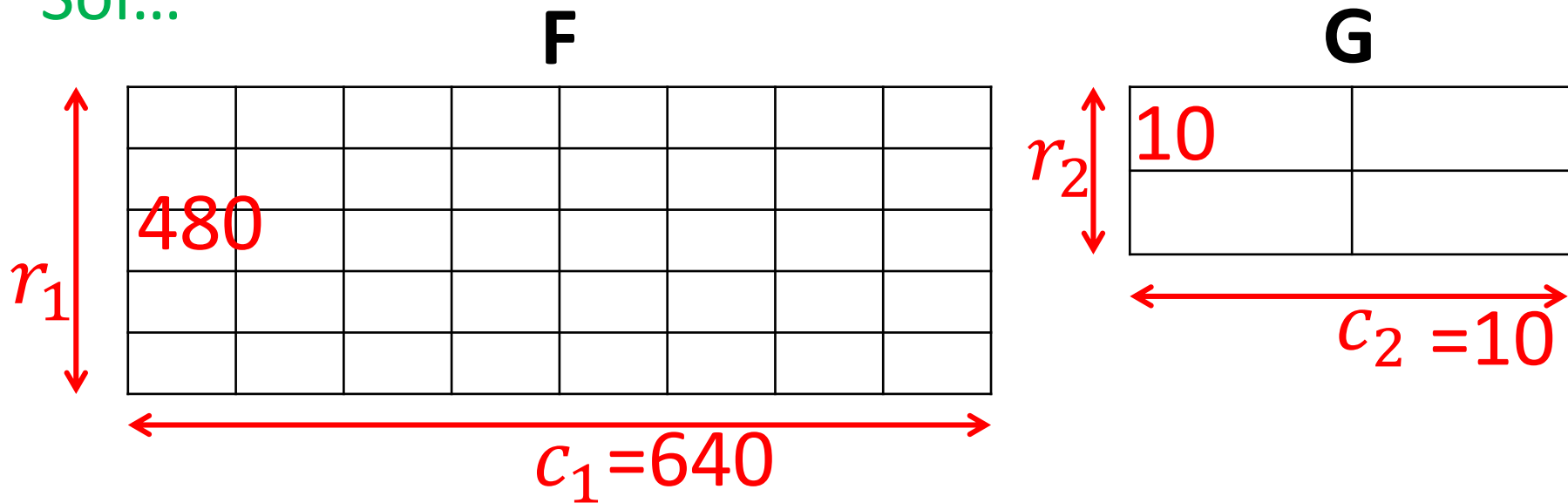
$$r' = r_1 - (r_2 - 1)$$

$$c' = c_1 - (c_2 - 1)$$

Question !

The original width and height of image I are 640 and 480 pixels, please determine the final dimension of this image when convolution with kernel size 10x10?

Sol...



$$r' = r_1 - (r_2 - 1)$$

$$r' = 480 - (10 - 1)$$

$$r' = 471$$

$$c' = c_1 - (c_2 - 1)$$

$$c' = 640 - (10 - 1)$$

$$c' = 631$$

Spatial convolution (Zero pad)

F

1	2	3	4	5	6	7	8	0
9	10	11	12	13	14	15	16	0
17	18	19	20	21	22	23	24	0
25	26	27	28	29	30	31	32	0
33	34	35	36	37	38	39	40	0
0	0	0	0	0	0	0	0	0

G

0.25	0.25
0.25	0.25

I

5.5	6.5	7.5	8.5	9.5	10.5	11.5	6
13.5	14.5	15.5	16.5	17.5	18.5	19.5	10
21.5	22.5	23.5	24.5	25.5	26.5	27.5	14
29.5	30.5	31.5	32.5	33.5	34.5	35.5	18
16.7	17.2	17.7	18.2	18.7	19.2	19.7	10

Spatial convolution (**Replicate**)

F

1	2	3	4	5	6	7	8	8
9	10	11	12	13	14	15	16	16
17	18	19	20	21	22	23	24	24
25	26	27	28	29	30	31	32	32
33	34	35	36	37	38	39	40	40
33	34	35	36	37	38	39	40	40

G

0.25	0.25
0.25	0.25

I

5.5	6.5	7.5	8.5	9.5	10.5	11.5	12
13.5	14.5	15.5	16.5	17.5	18.5	19.5	20
21.5	22.5	23.5	24.5	25.5	26.5	27.5	28
29.5	30.5	31.5	32.5	33.5	34.5	35.5	36
33.5	34.5	35.5	36.5	37.5	38.8	39.5	40

Question !

207	24	40	36	167
230	71	247	107	9
32	139	244	233	216
232	244	123	202	238
161	246	204	244	173

I

Image **I** is represented as **double precision**.

Please determine the **$I * M$** ?

1 0 1

0 -4 0

1 0 1

M

Sol..

239

-556

239

276

-352

-315

-335

370

29

$I * M$

Quiz : คำนวณ $F * G$ โดยไม่ต้อง pad

F

1	2	3	4
9	10	11	12

G

0.5	0.5
-----	-----

$$I = F * G$$

I

1.5	2.5	3.5
9.5	10.5	11.5

Quiz : คำนวณ $F * G$ โดยไม่ต้อง pad

F

1	2	3	4
9	10	11	12

G

0.5
0.5

$$I = F * G$$

I

5	6	7	8
---	---	---	---



Image Filter

Frequency Filter

- Low-pass filter
- High-pass filter
- Band-pass filter
- Band-stop filter

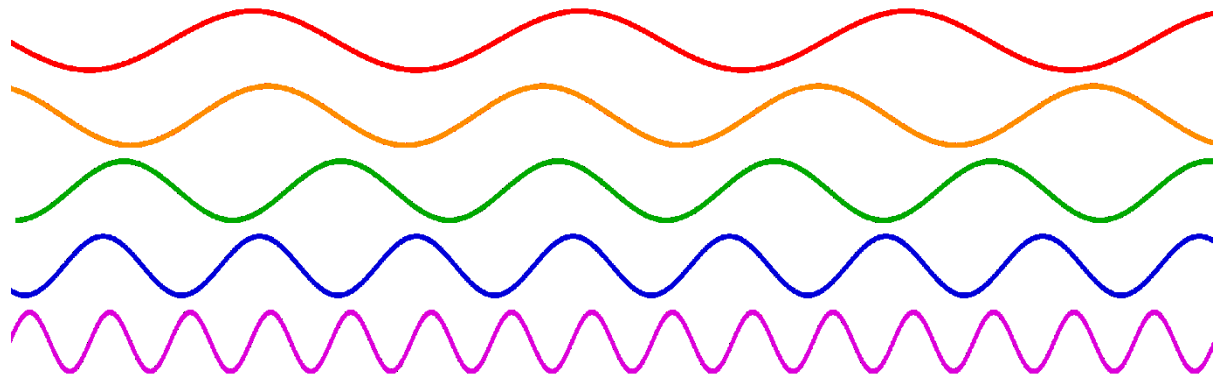
Frequency

Frequency is the number of occurrences of a repeating event per unit time.

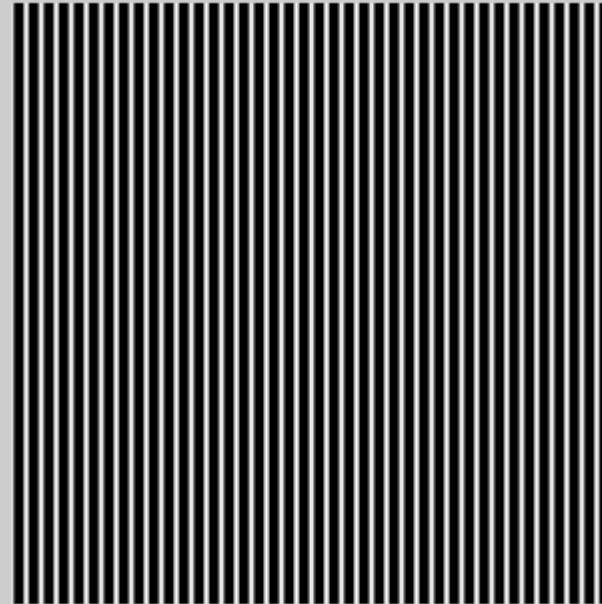
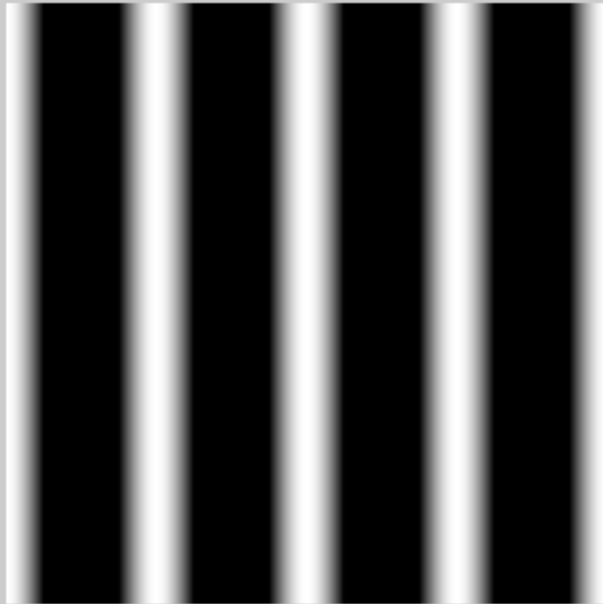
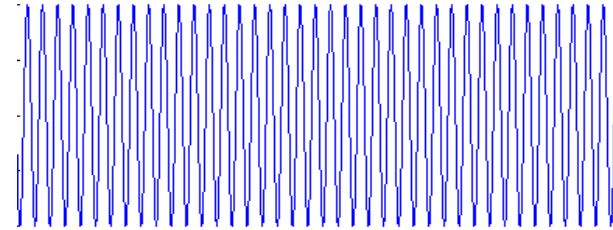
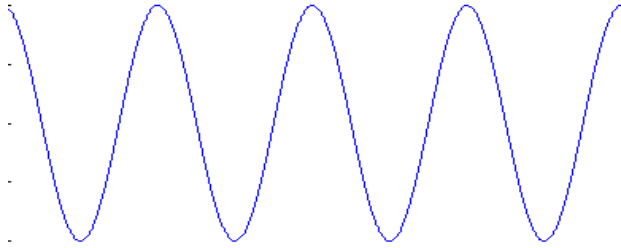
● $f = 0.5 \text{ Hz}$
 $T = 2.0 \text{ s}$

● $f = 1.0 \text{ Hz}$
 $T = 1.0 \text{ s}$

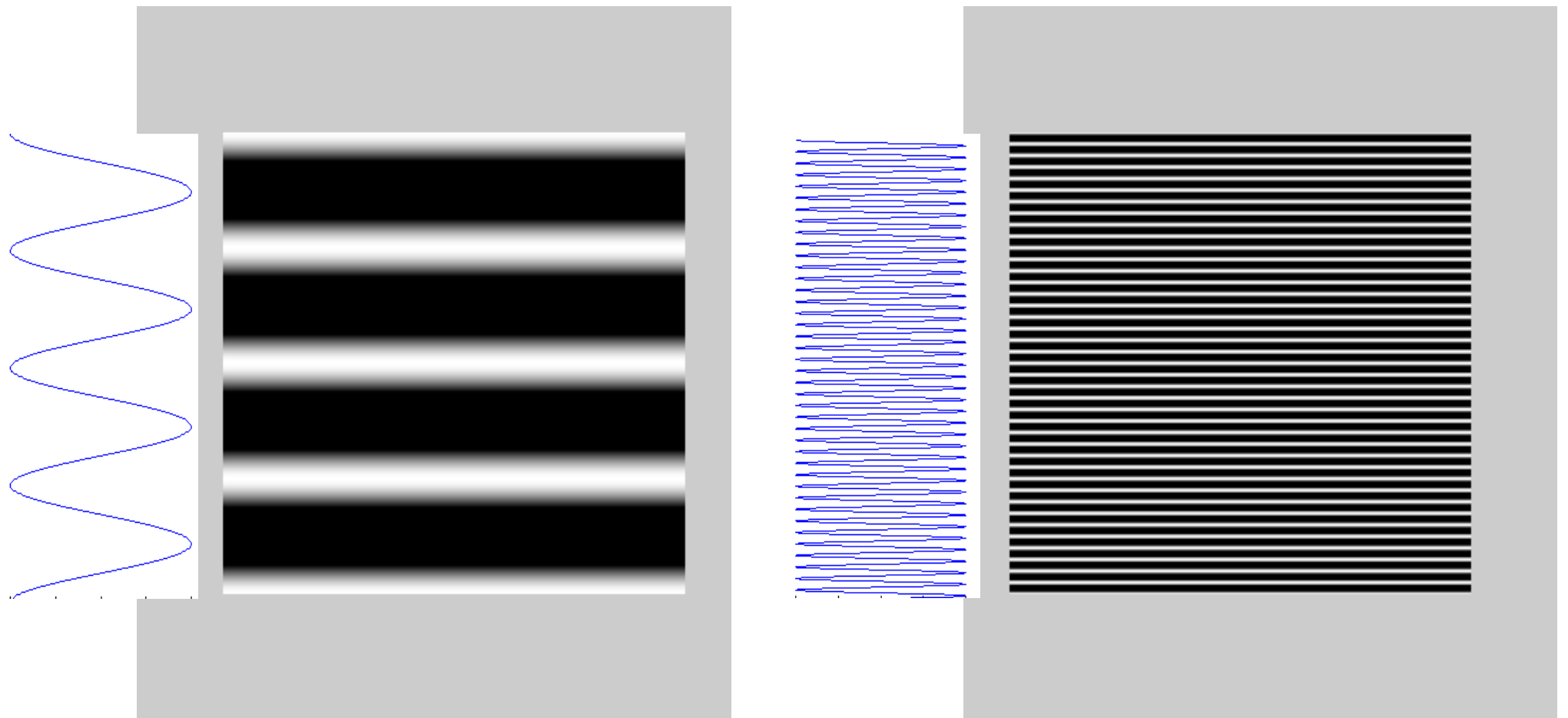
● $f = 2.0 \text{ Hz}$
 $T = 0.5 \text{ s}$



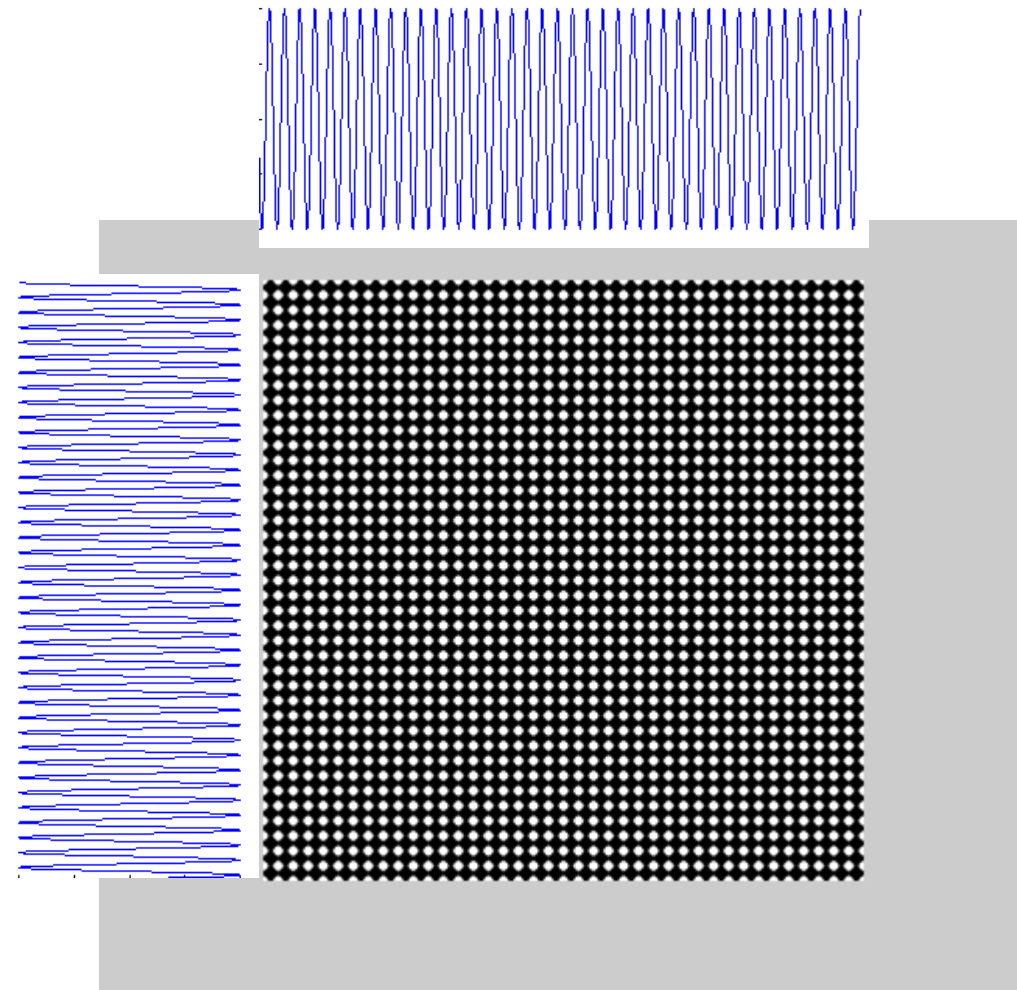
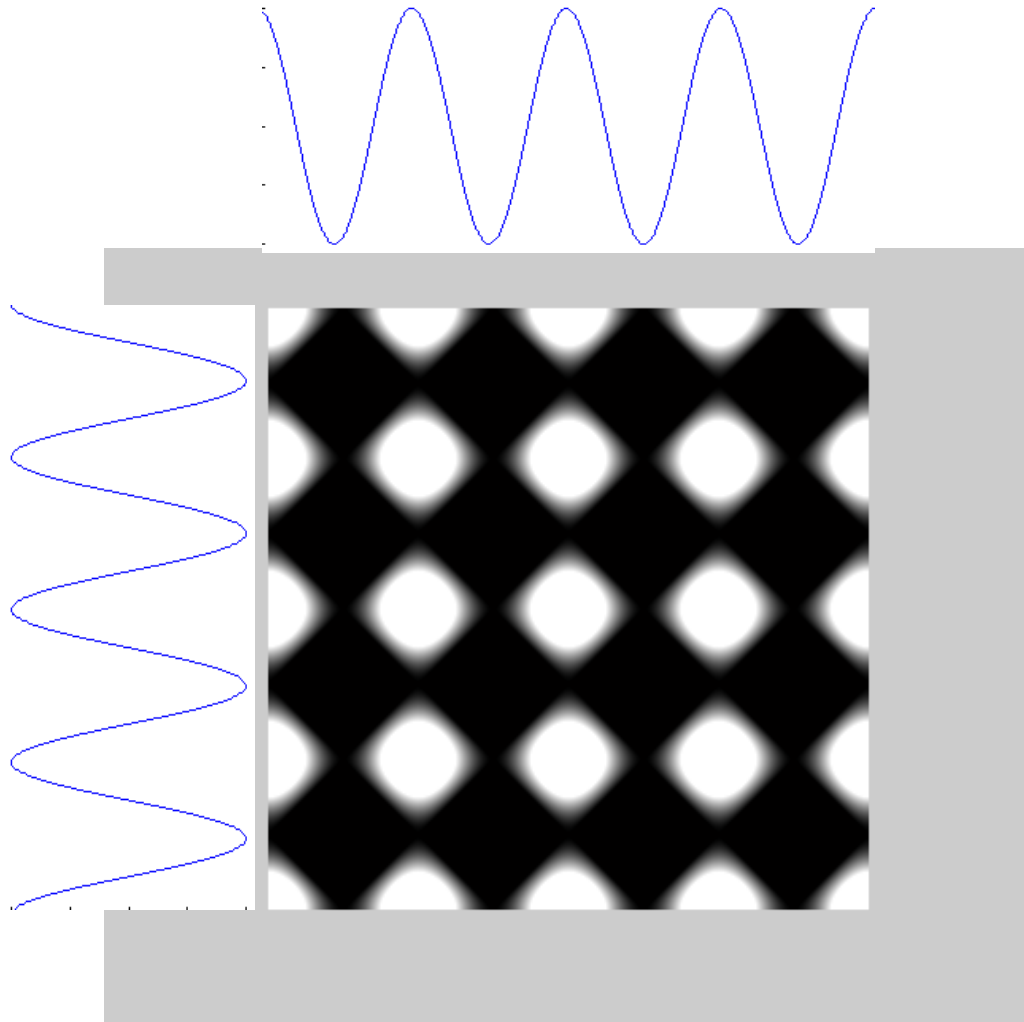
Frequency



Frequency



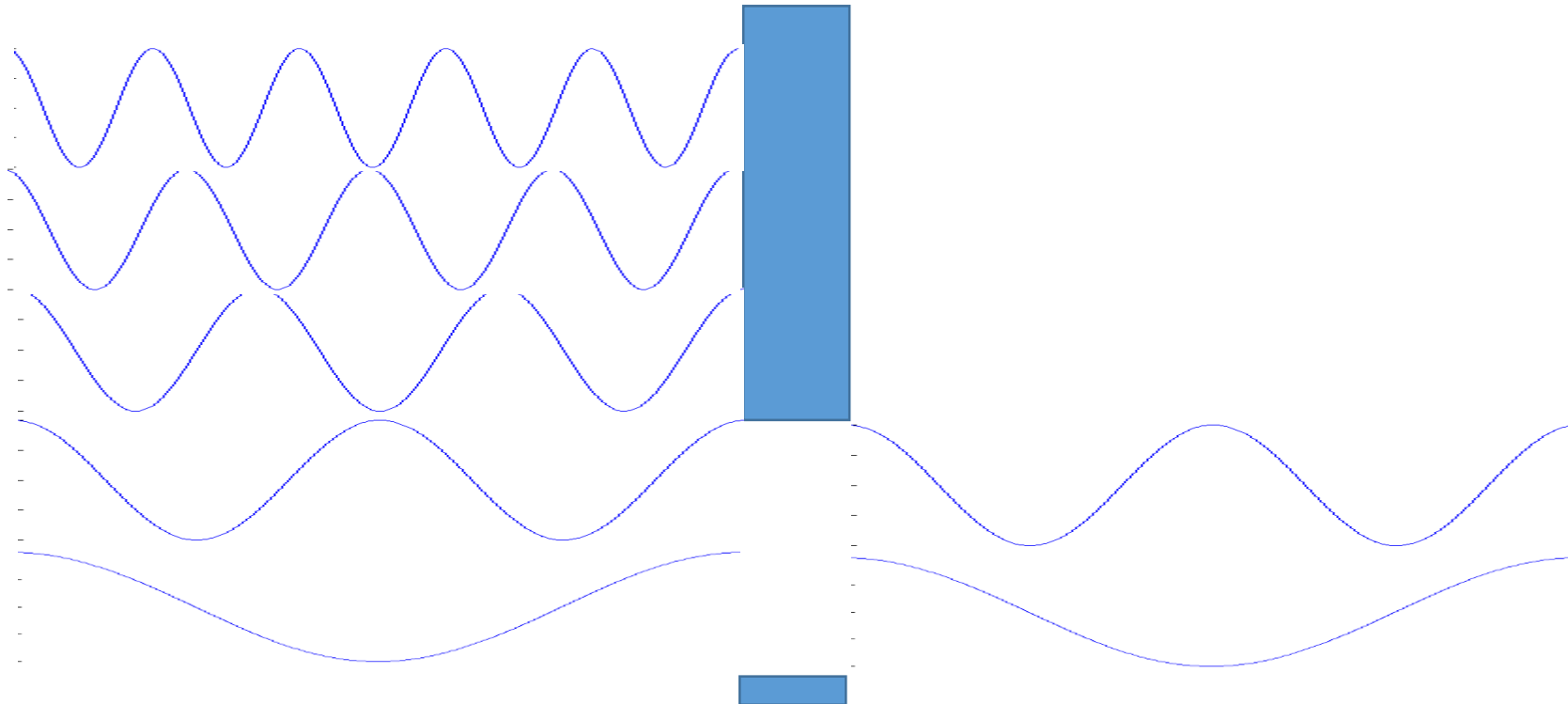
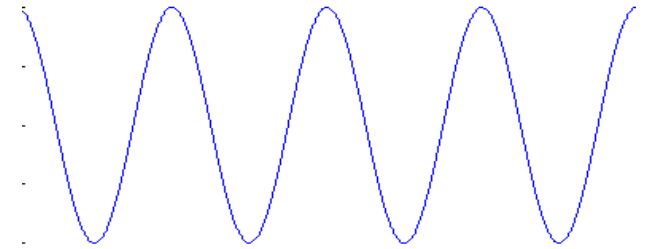
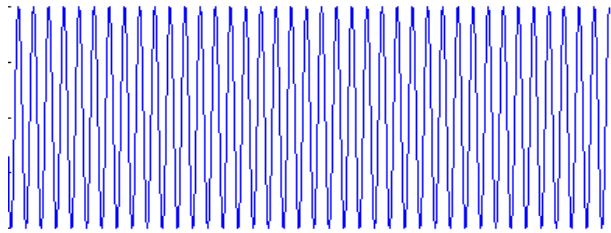
Frequency



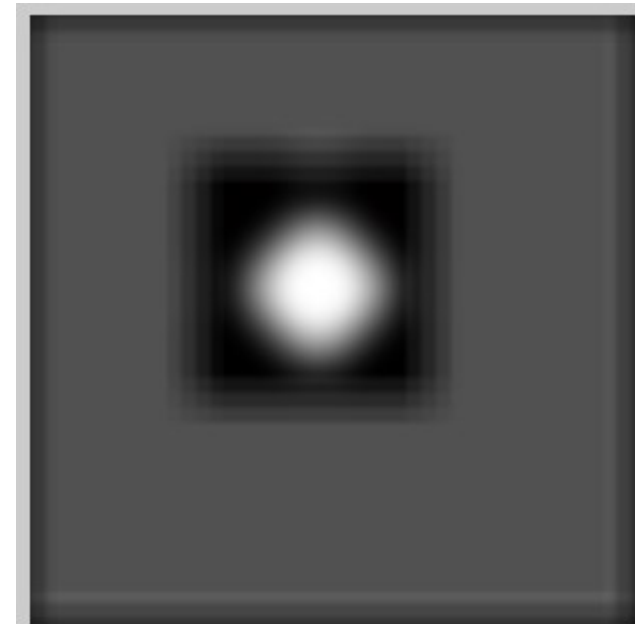
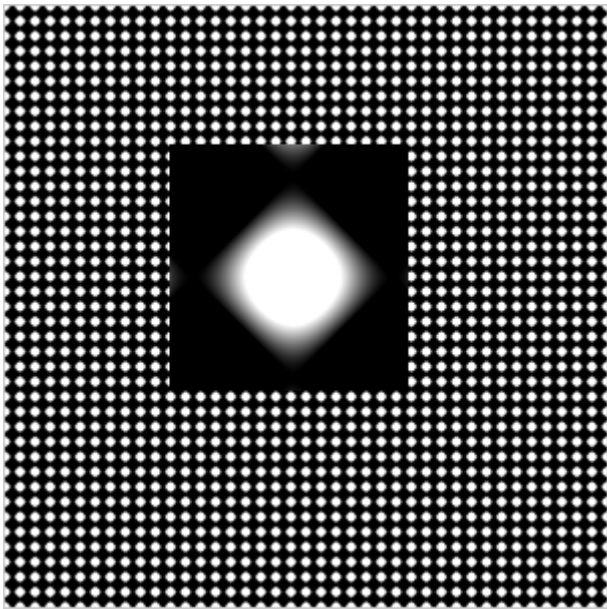
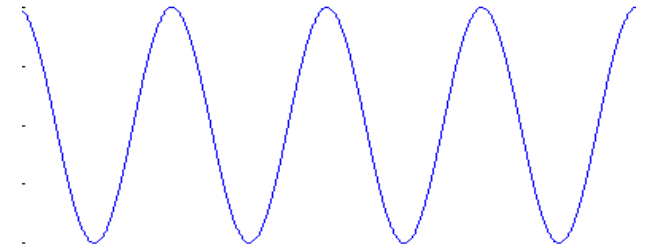
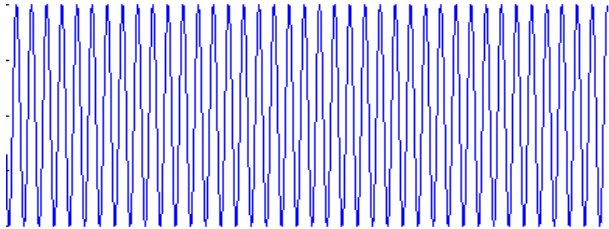
Frequency



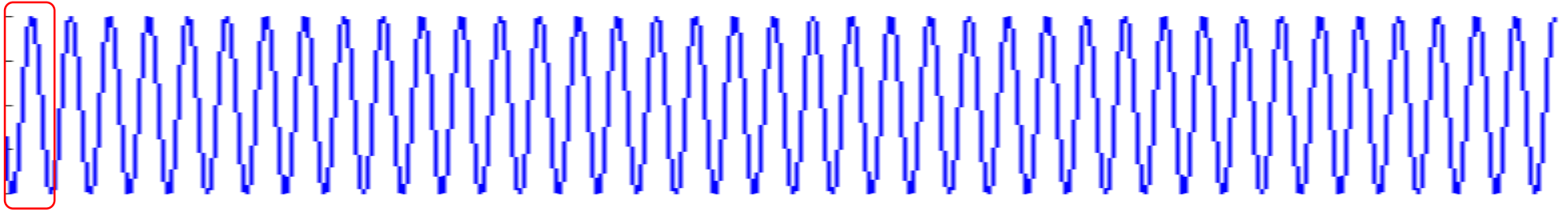
Low-pass filter



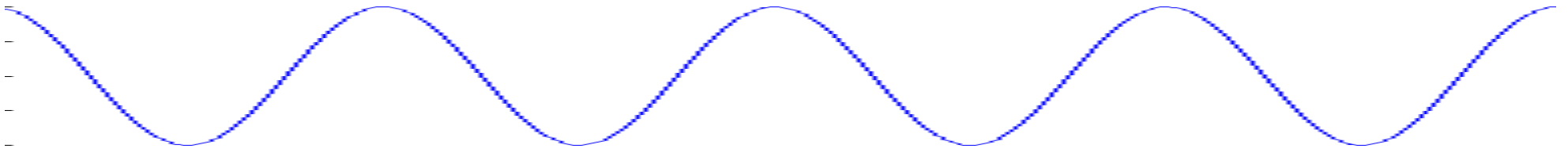
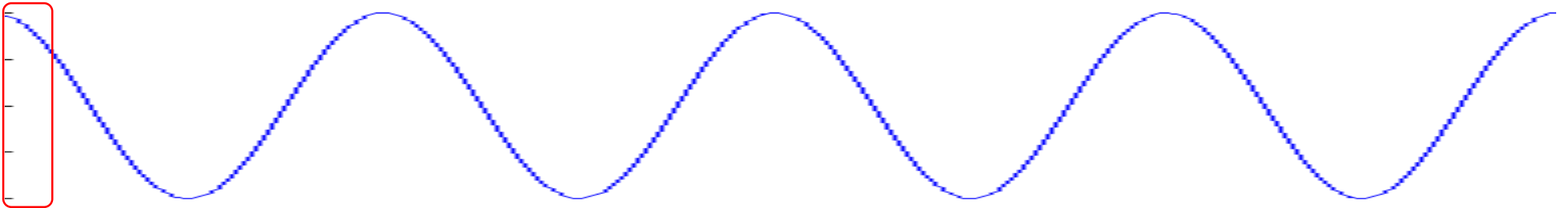
Low-pass filter



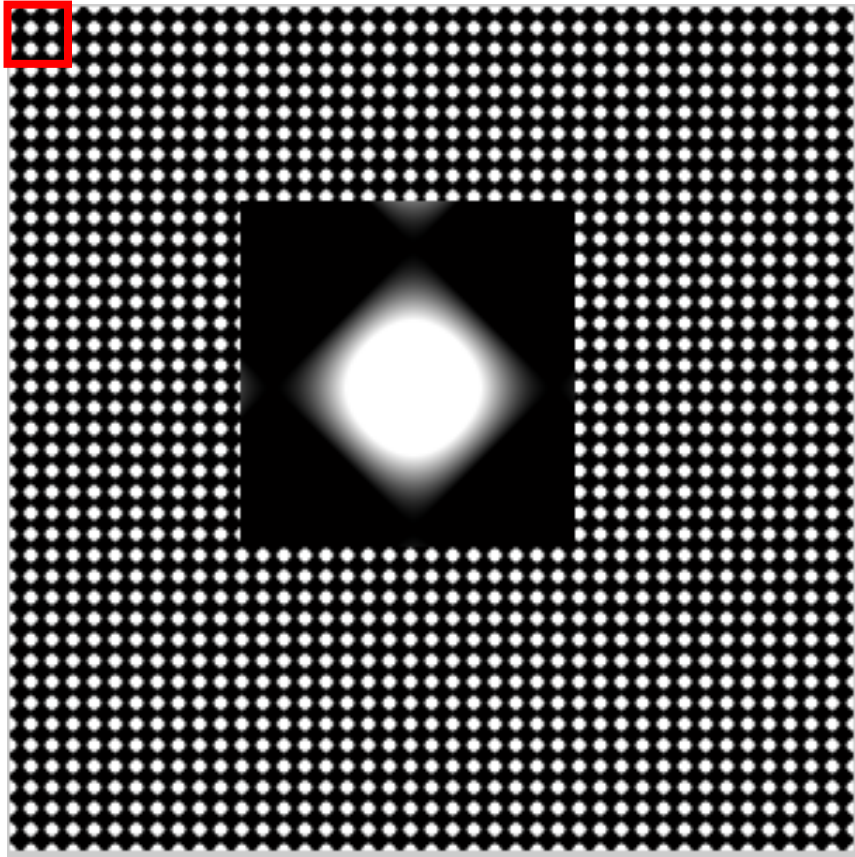
Low-pass filter



AVERAGE



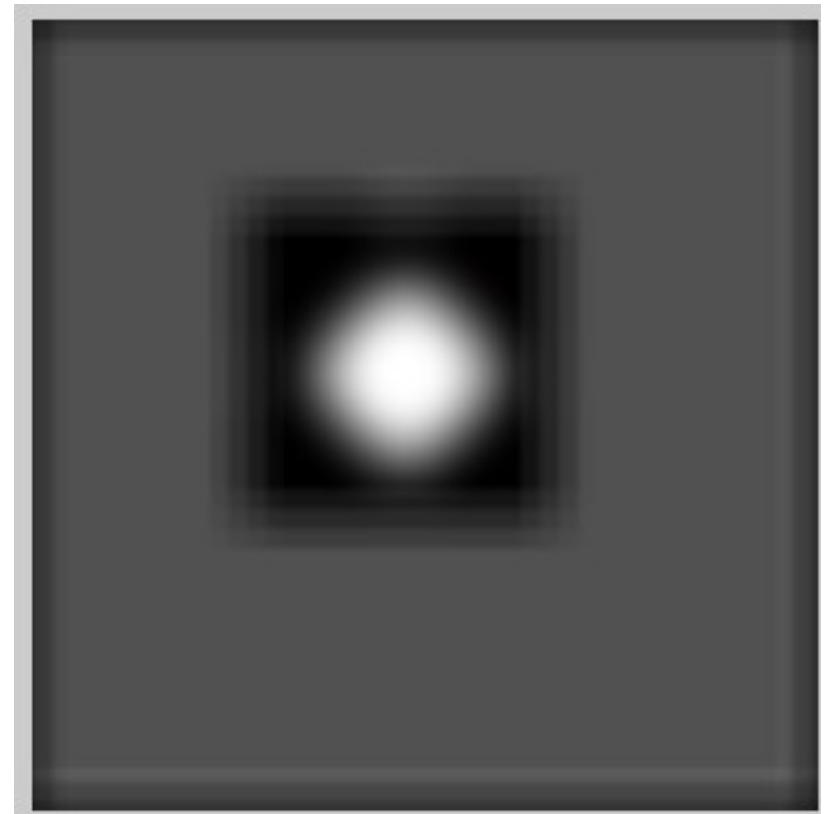
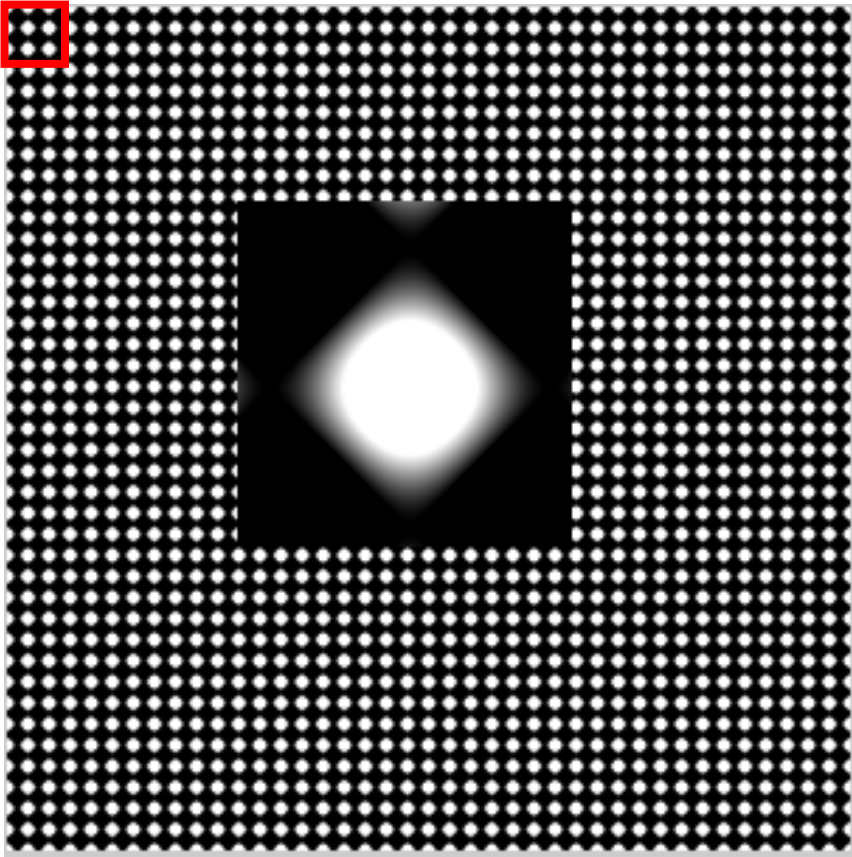
Low-pass filter



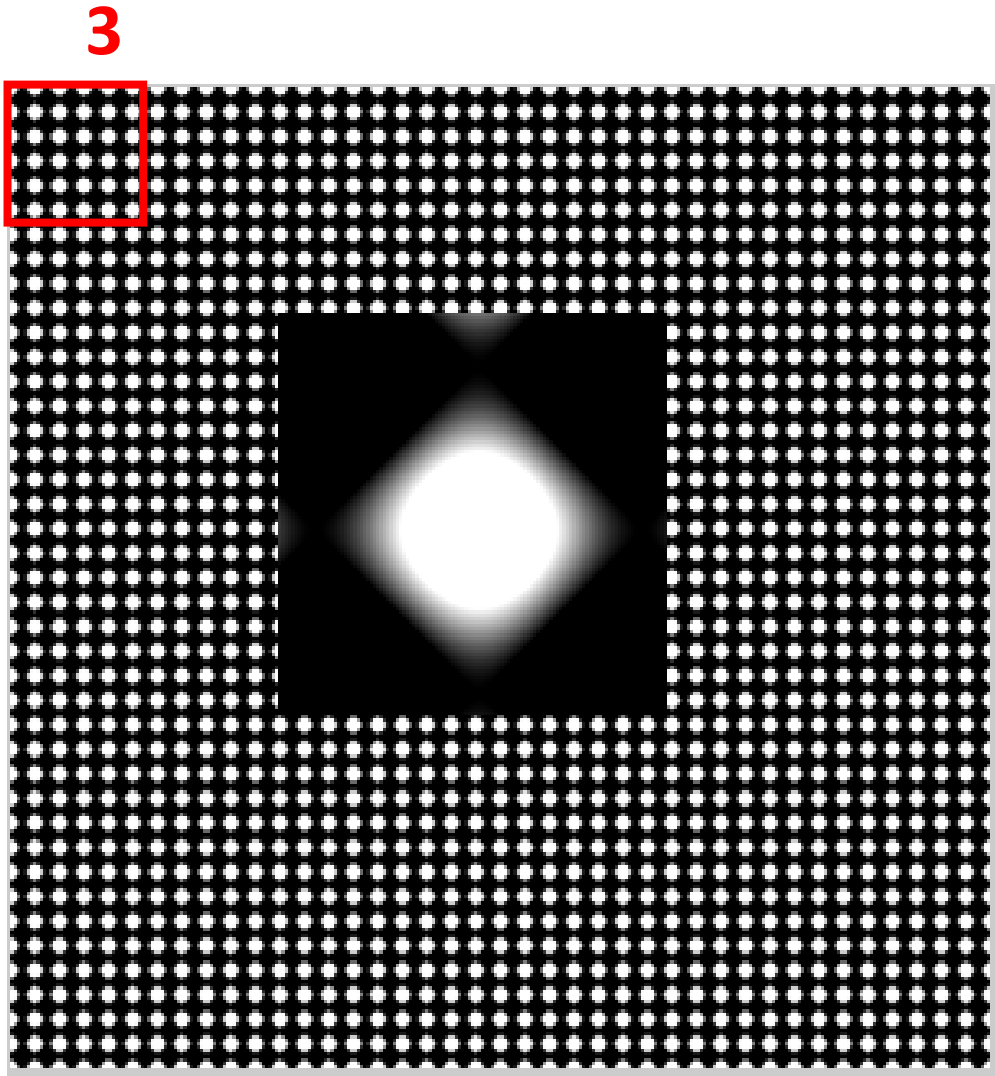
Arithmetic mean

$$k = \sum_X \sum_Y I(x, y) / (|X| \times |Y|)$$

Low-pass filter



Low-pass filter



*

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Average Kernel

Low-pass filter

Average Kernel

$1/4$	$1/4$
$1/4$	$1/4$

2x2

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

3x3

$1/16$	$1/16$	$1/16$	$1/16$
$1/16$	$1/16$	$1/16$	$1/16$
$1/16$	$1/16$	$1/16$	$1/16$
$1/16$	$1/16$	$1/16$	$1/16$

4x4

Low-pass filter

Average Kernel



5x5



10x10



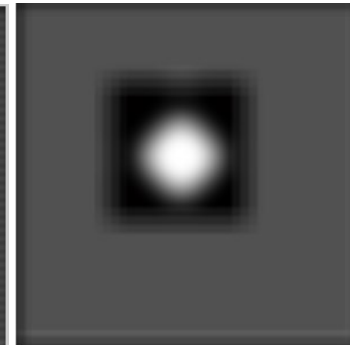
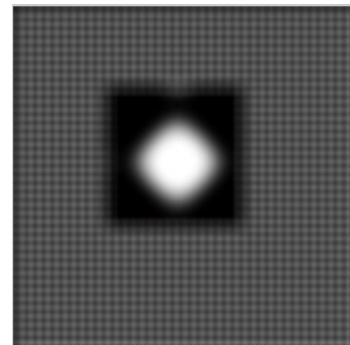
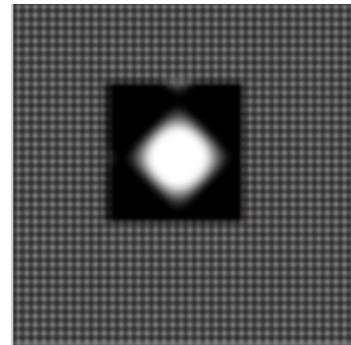
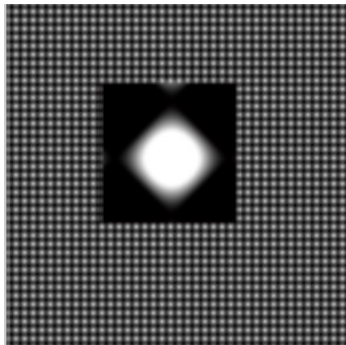
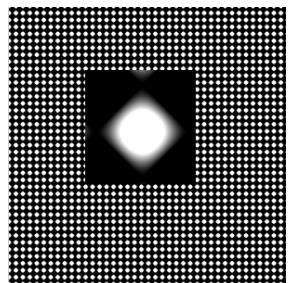
20x20



30x30



40x40



Low-pass filter



**salt & pepper
noise**



2x2



3x3



4x4



5x5



speckle



2x2



3x3



4x4



5x5

Low-pass filter

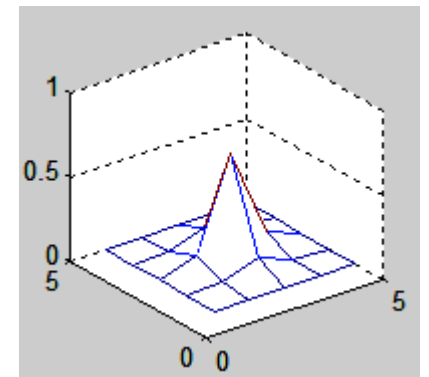
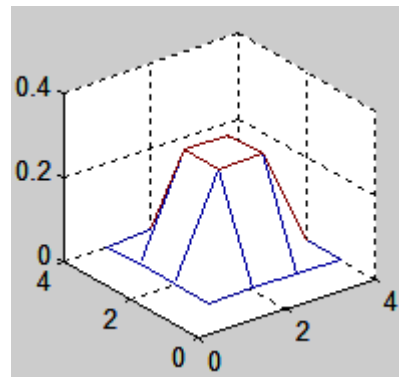
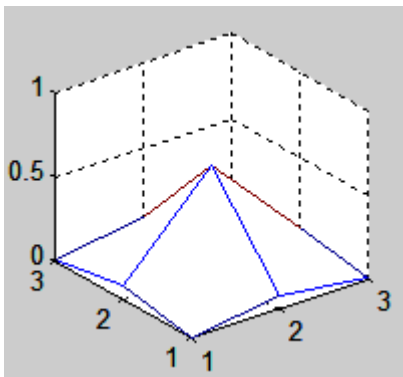
Gaussian Kernel

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

0.0113	0.0838	0.0113
0.0838	0.6193	0.0838
0.0113	0.0838	0.0113

0.0001	0.0044	0.0044	0.0001
0.0044	0.2411	0.2411	0.0044
0.0044	0.2411	0.2411	0.0044
0.0001	0.0044	0.0044	0.0001

0.0000	0.0000	0.0002	0.0000	0.0000
0.0000	0.0113	0.0837	0.0113	0.0000
0.0002	0.0837	0.6187	0.0837	0.0002
0.0000	0.0113	0.0837	0.0113	0.0000
0.0000	0.0000	0.0002	0.0000	0.0000



Low-pass filter



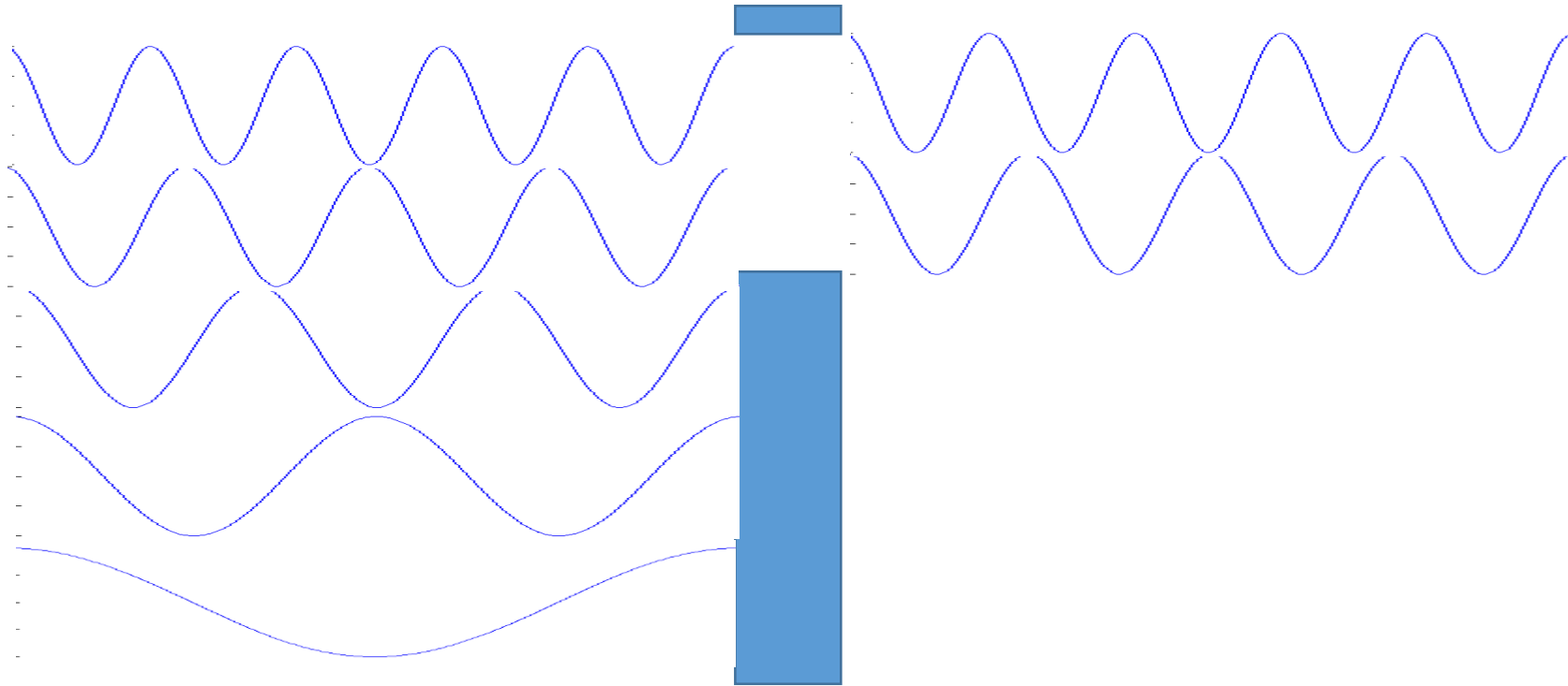
Gaussian 5x5, $\sigma=1.2$



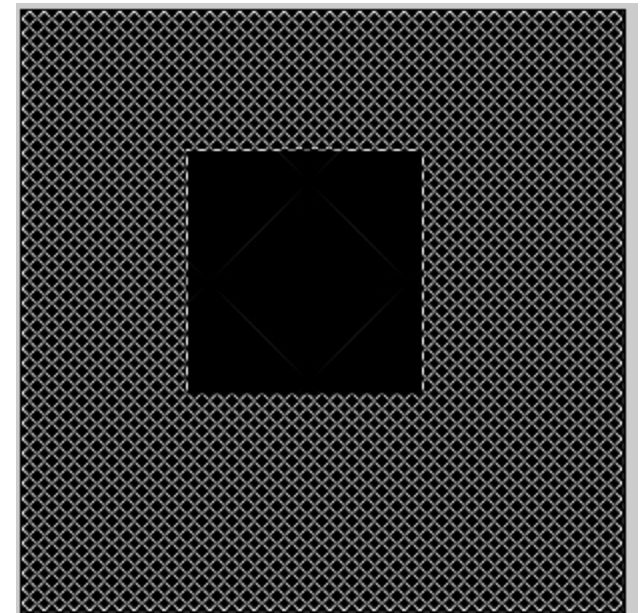
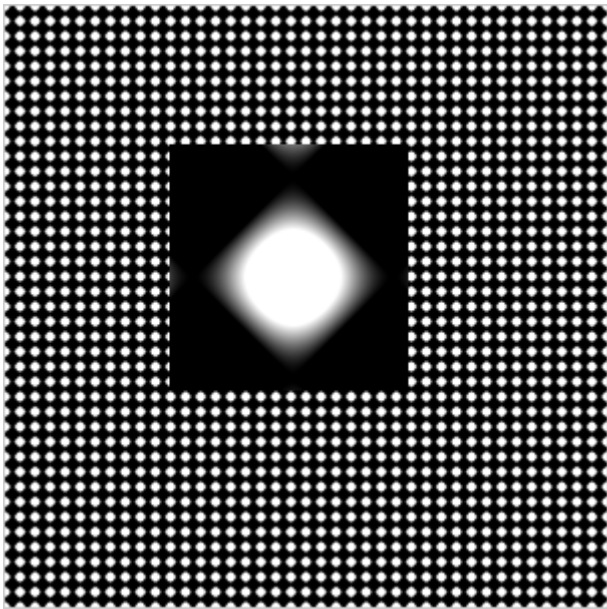
Average 5x5



High-pass filter



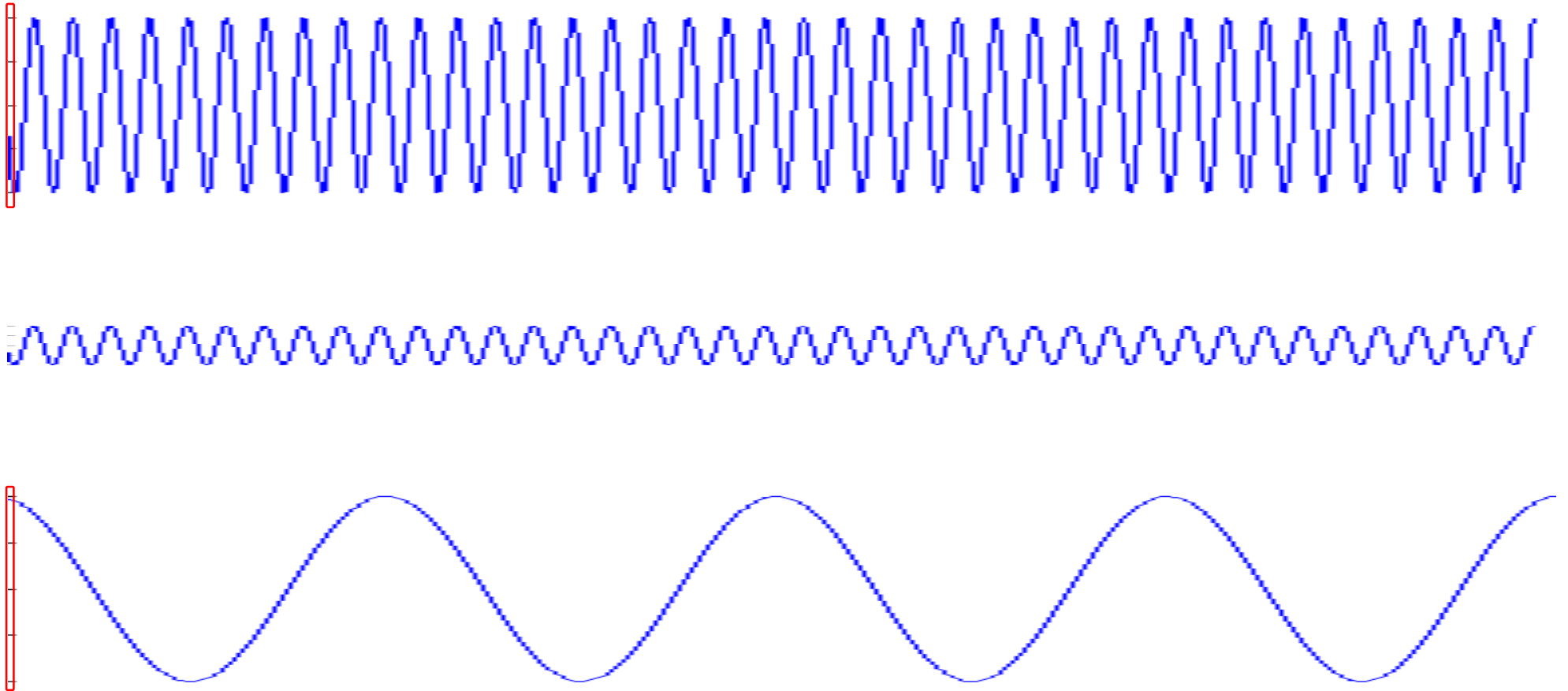
High-pass filter



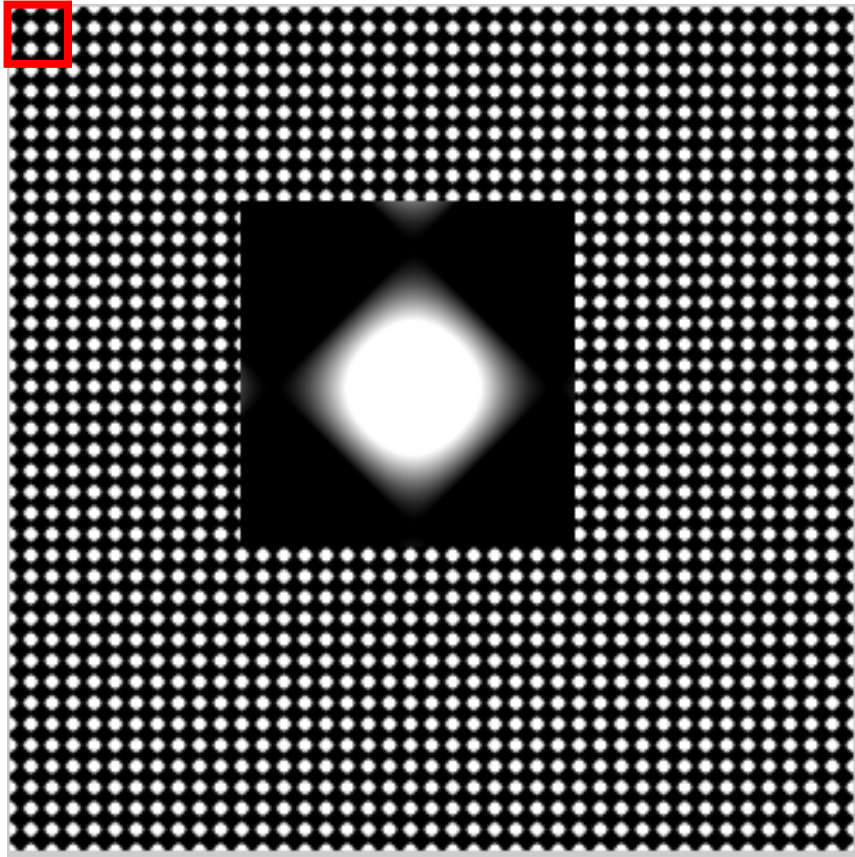
High-pass filter

Derivative

$$\frac{\partial(y)}{\partial(x)}$$



High-pass filter



2D gradient

$$k = \frac{\partial I}{\partial x} + \frac{\partial I}{\partial y}$$

High-pass filter

Laplacian Kernel

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

$$\nabla^2 = \frac{4}{\alpha + 1} \begin{bmatrix} \frac{\alpha}{4} & \frac{1 - \alpha}{4} & \frac{\alpha}{4} \\ \frac{1 - \alpha}{4} & -1 & \frac{1 - \alpha}{4} \\ \frac{\alpha}{4} & \frac{1 - \alpha}{4} & \frac{\alpha}{4} \end{bmatrix}$$

$$\mathbf{0} \leq \alpha \leq \mathbf{1}$$

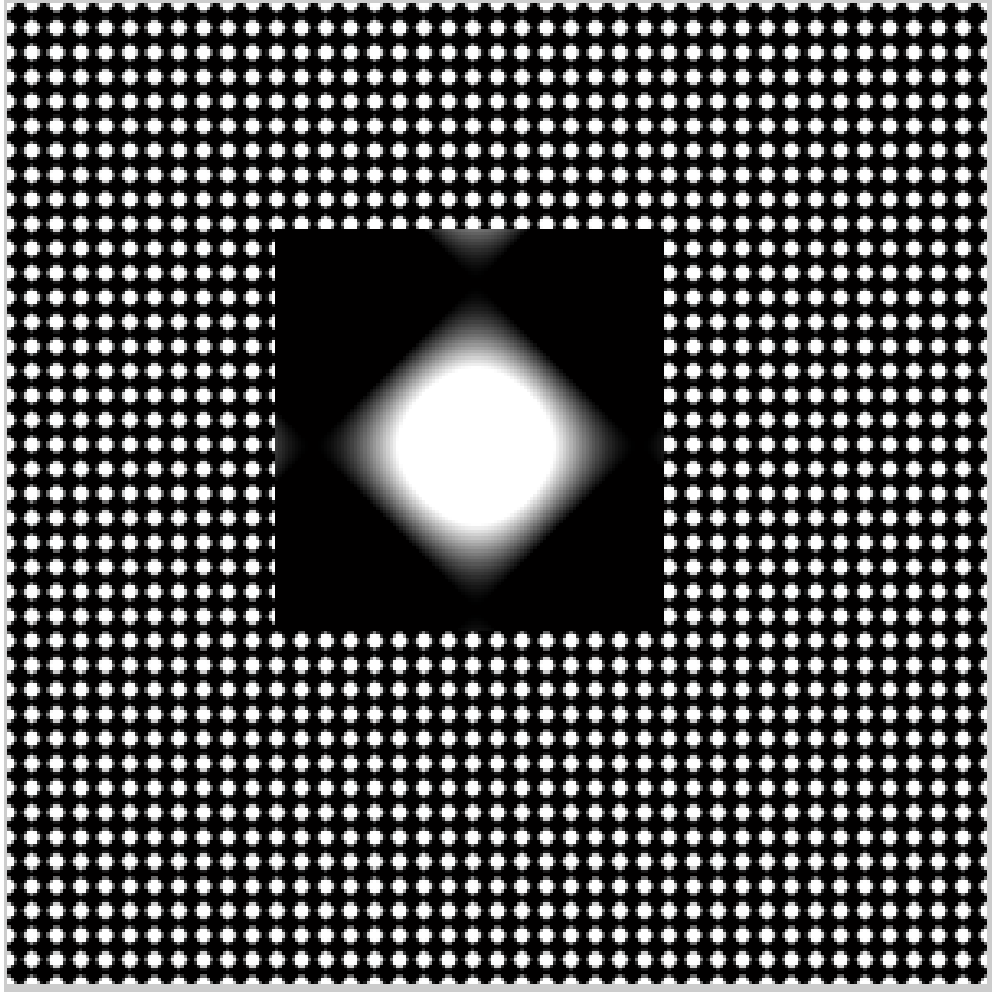
High-pass filter

Laplacian Kernel

$$\alpha = 0, \quad \nabla^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

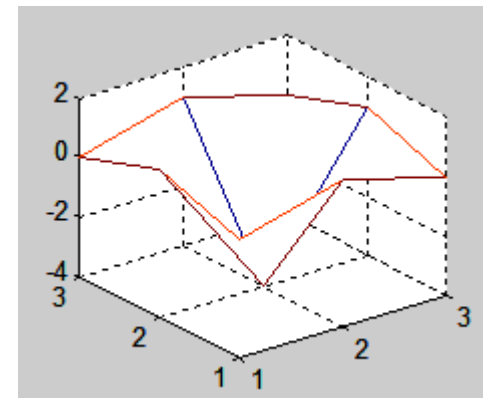
$$\alpha = 1, \quad \nabla^2 = \begin{bmatrix} .5 & 0 & .5 \\ 0 & -2 & 0 \\ .5 & 0 & .5 \end{bmatrix}$$

High-pass filter

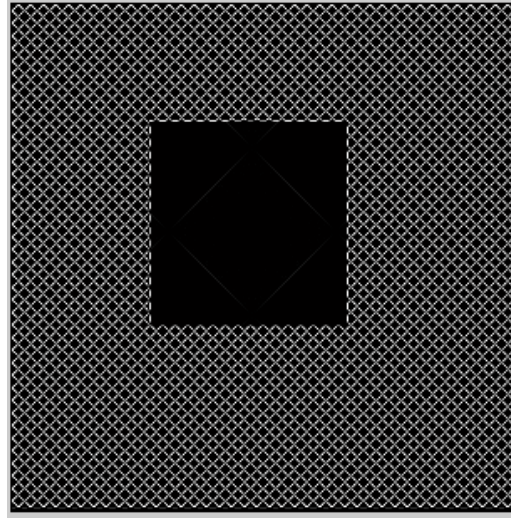
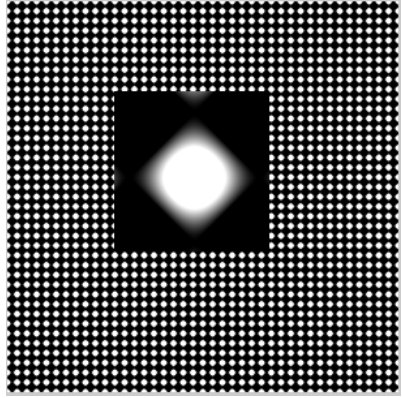


*

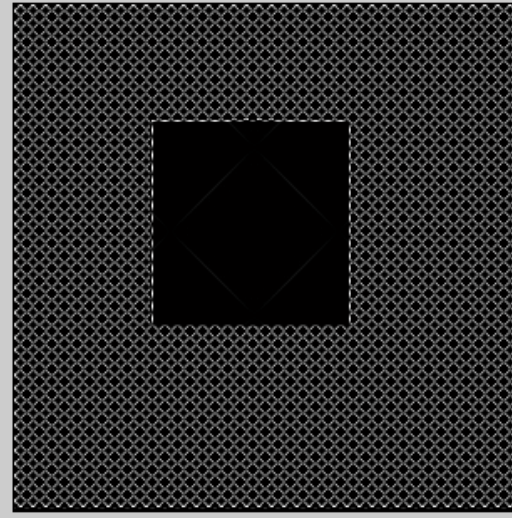
0	1	0
1	-4	1
0	1	0



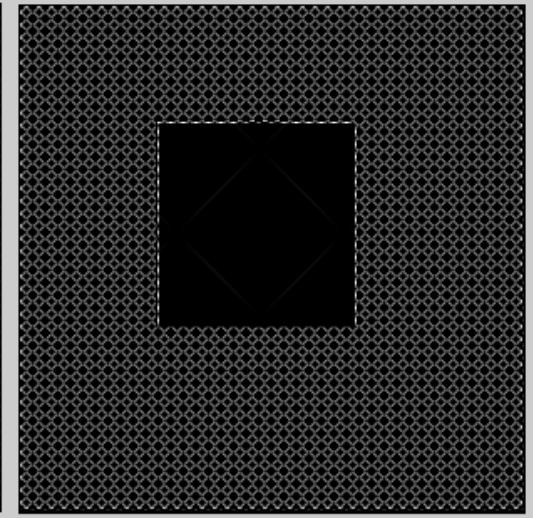
High-pass filter



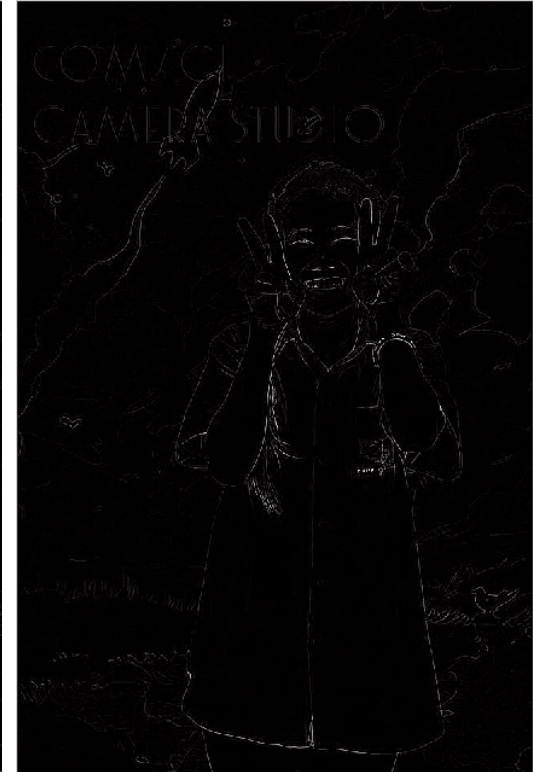
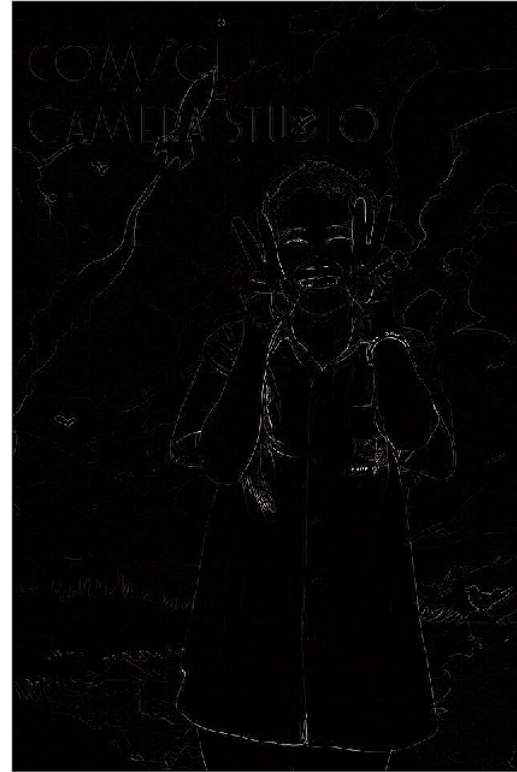
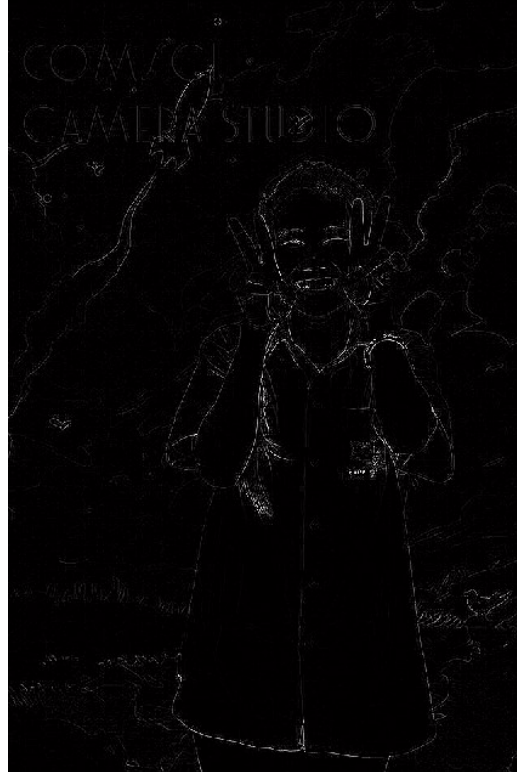
$\alpha = 0$



$\alpha = 0.5$



$\alpha = 1$



High-pass filter

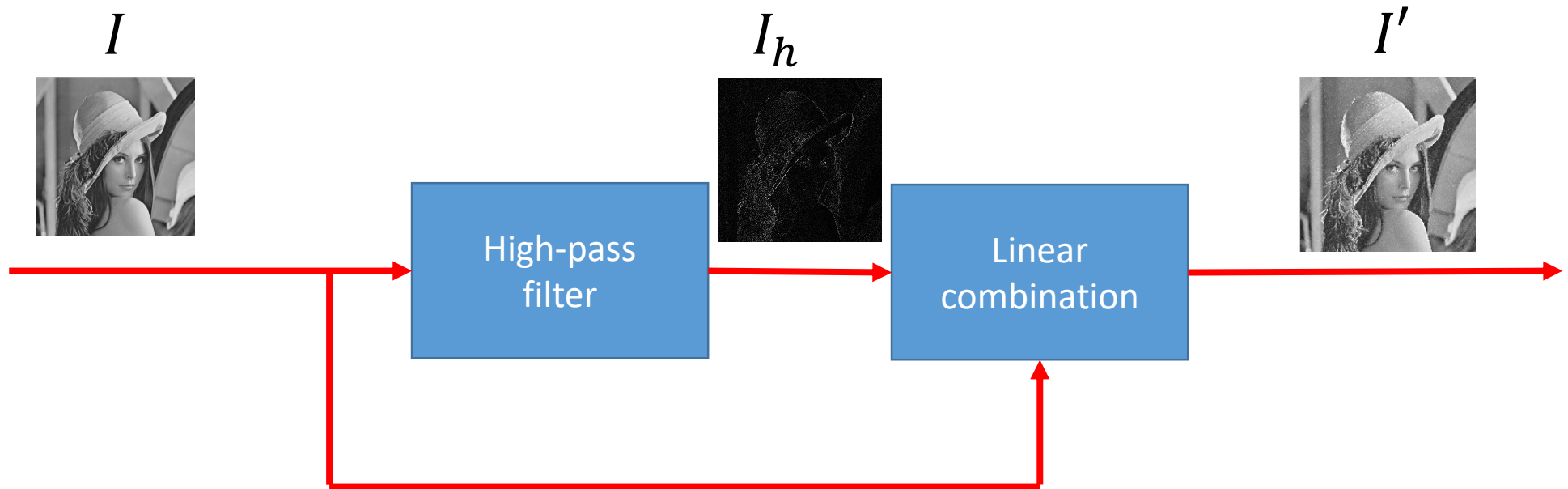


Image Sharpening Algorithm

$$I' = I + wI_h$$

High-pass filter



$\alpha = 0, w = 0.4$



$\alpha = 0, w = 0.6$



$\alpha = 0, w = 10$



High-pass filter



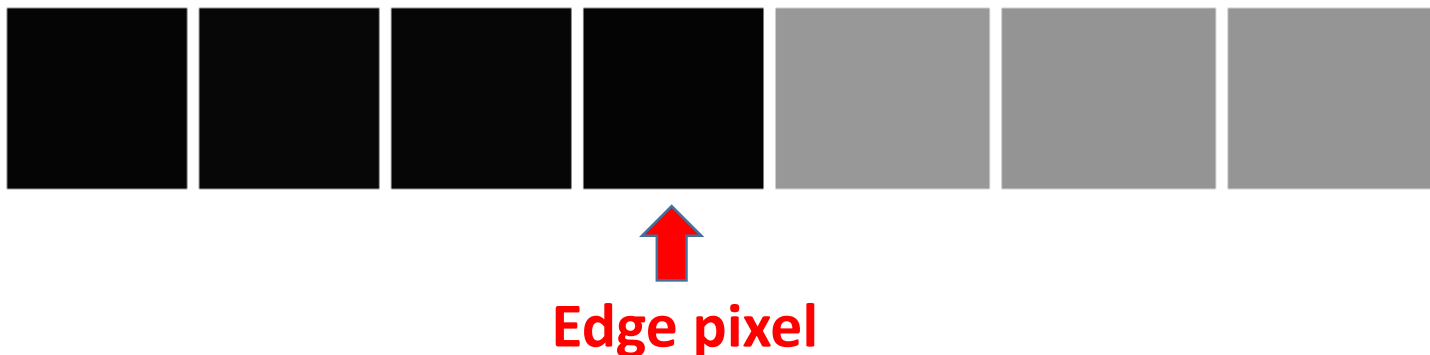
-1	-1	-1
-1	9	-1
-1	-1	-1



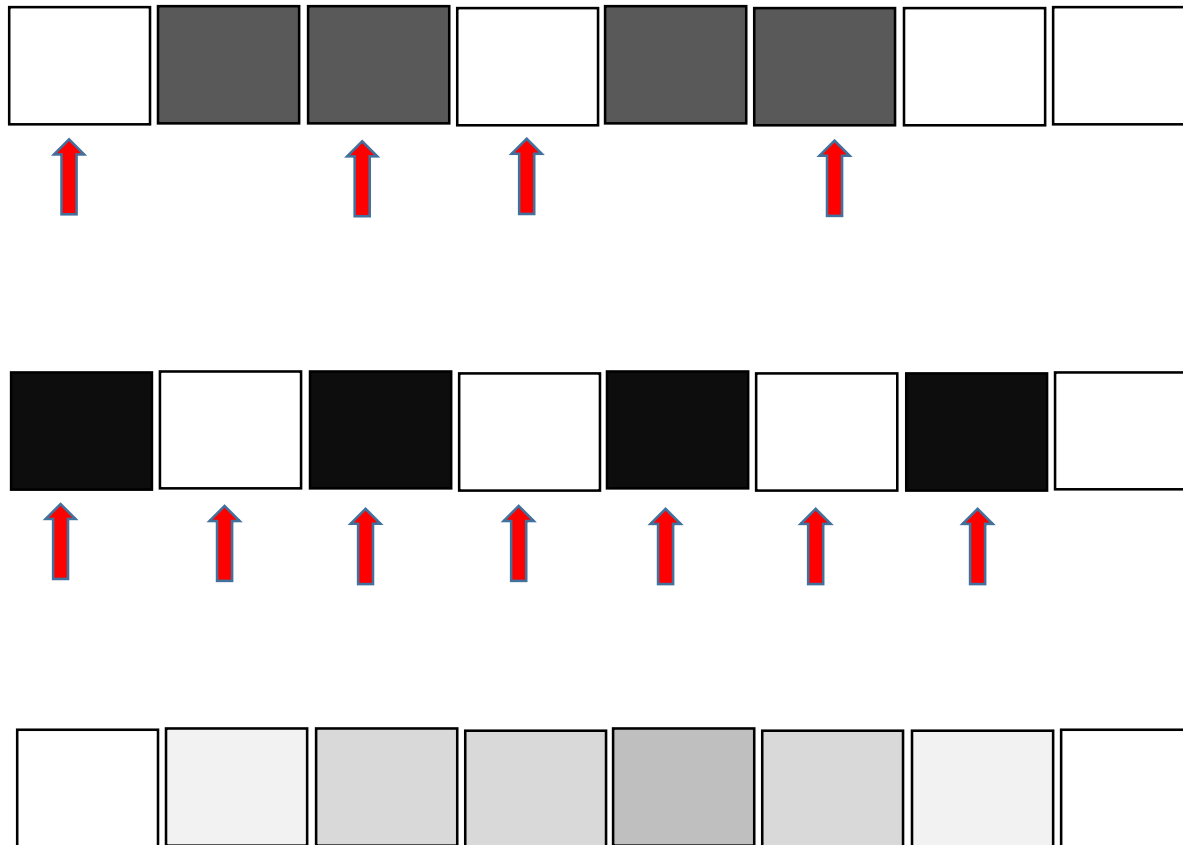
$$w = 0.4$$

Edge detection

Edge detection is the name for a set of mathematical methods which aim at identifying **points** in a digital image at which the image brightness changes sharply or, more formally, has **discontinuities**.



Edge detection



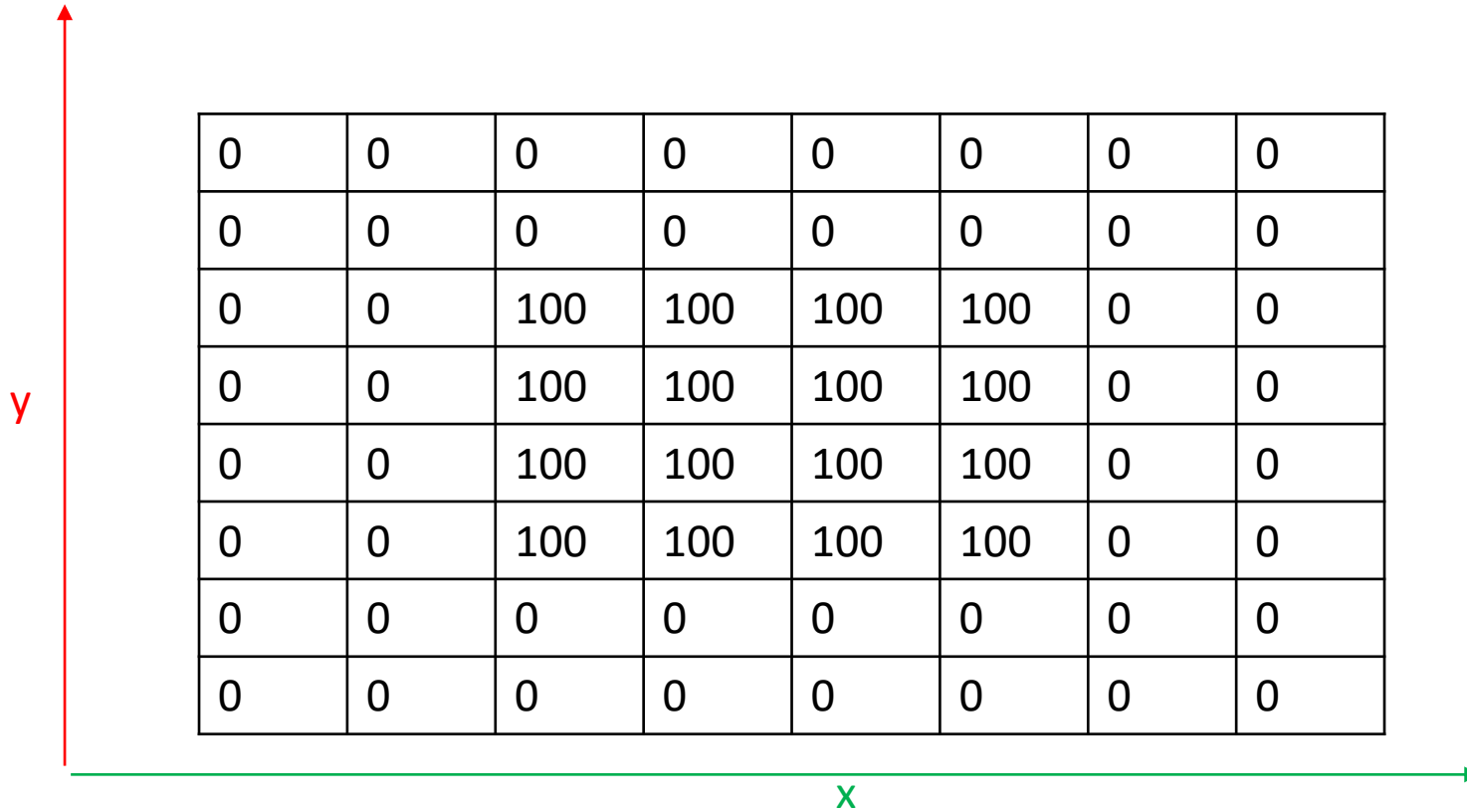
$$Edge_i = I_{i-1} - I_i$$

Edge detection

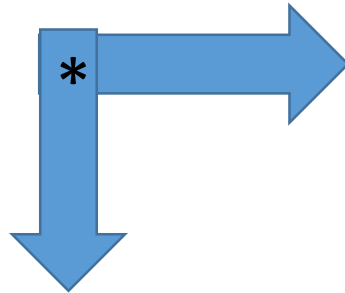
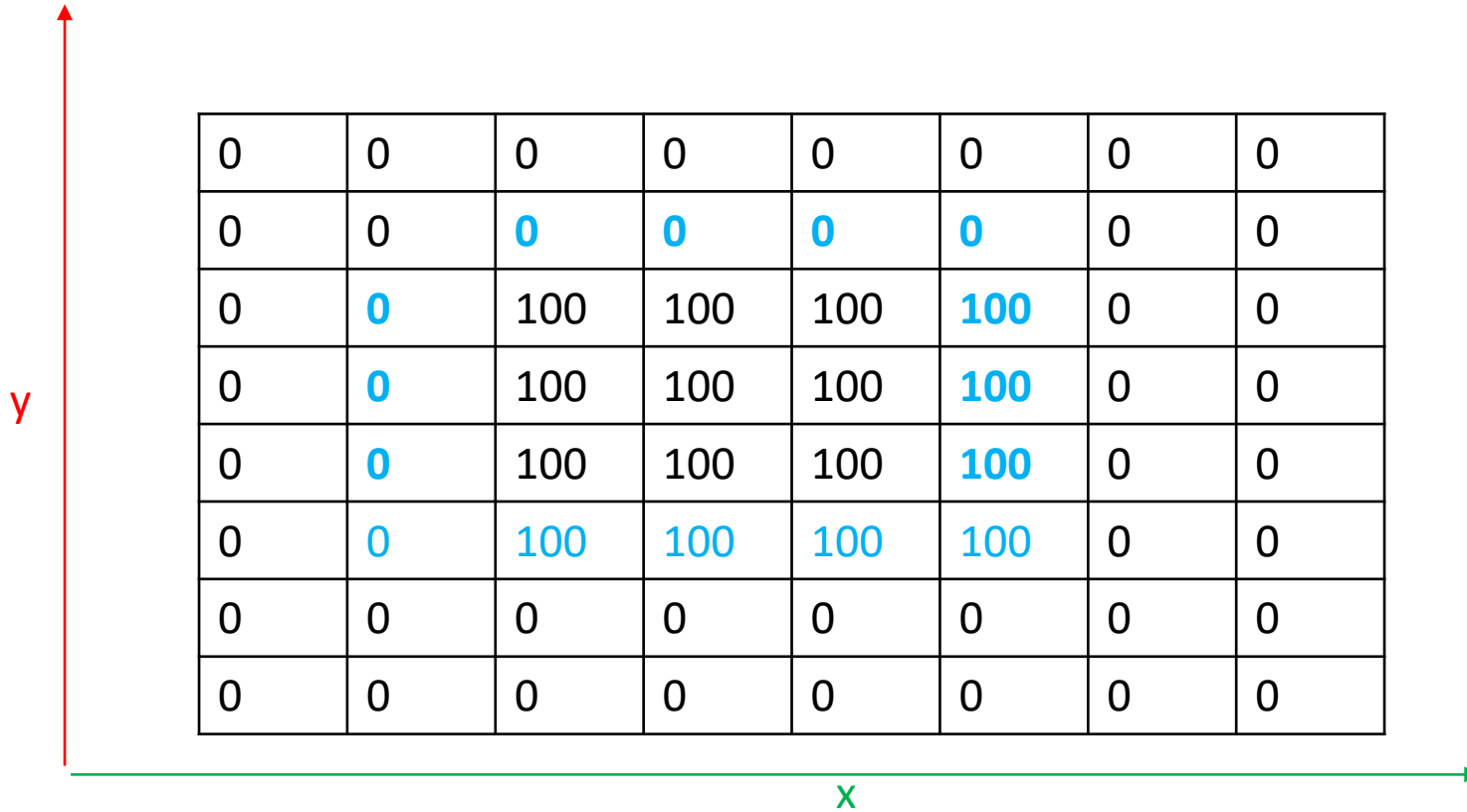


Gradient Magnitude Edge Detection Method

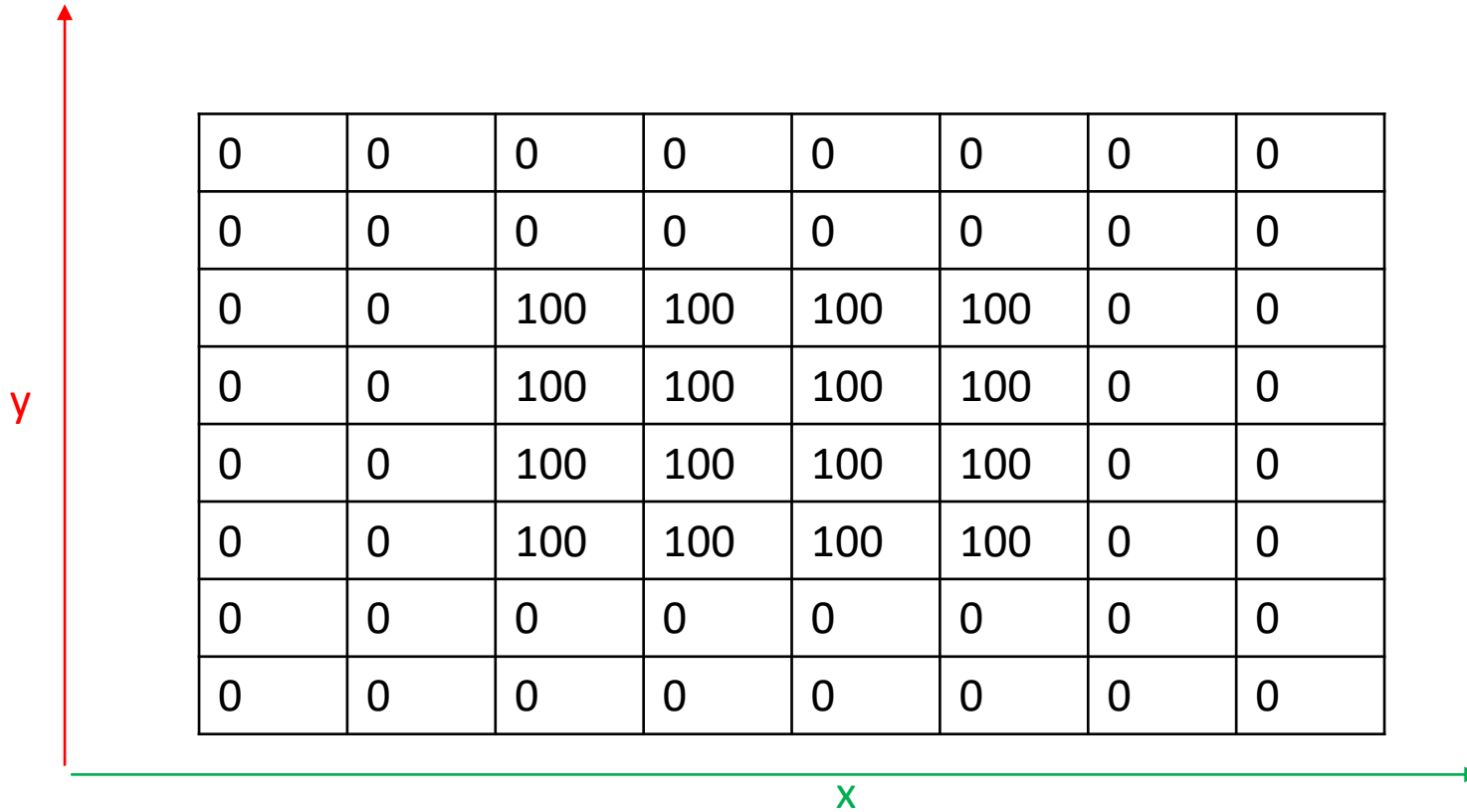
Edge detection(Gradient Magnitude)



Edge detection(Gradient Magnitude)



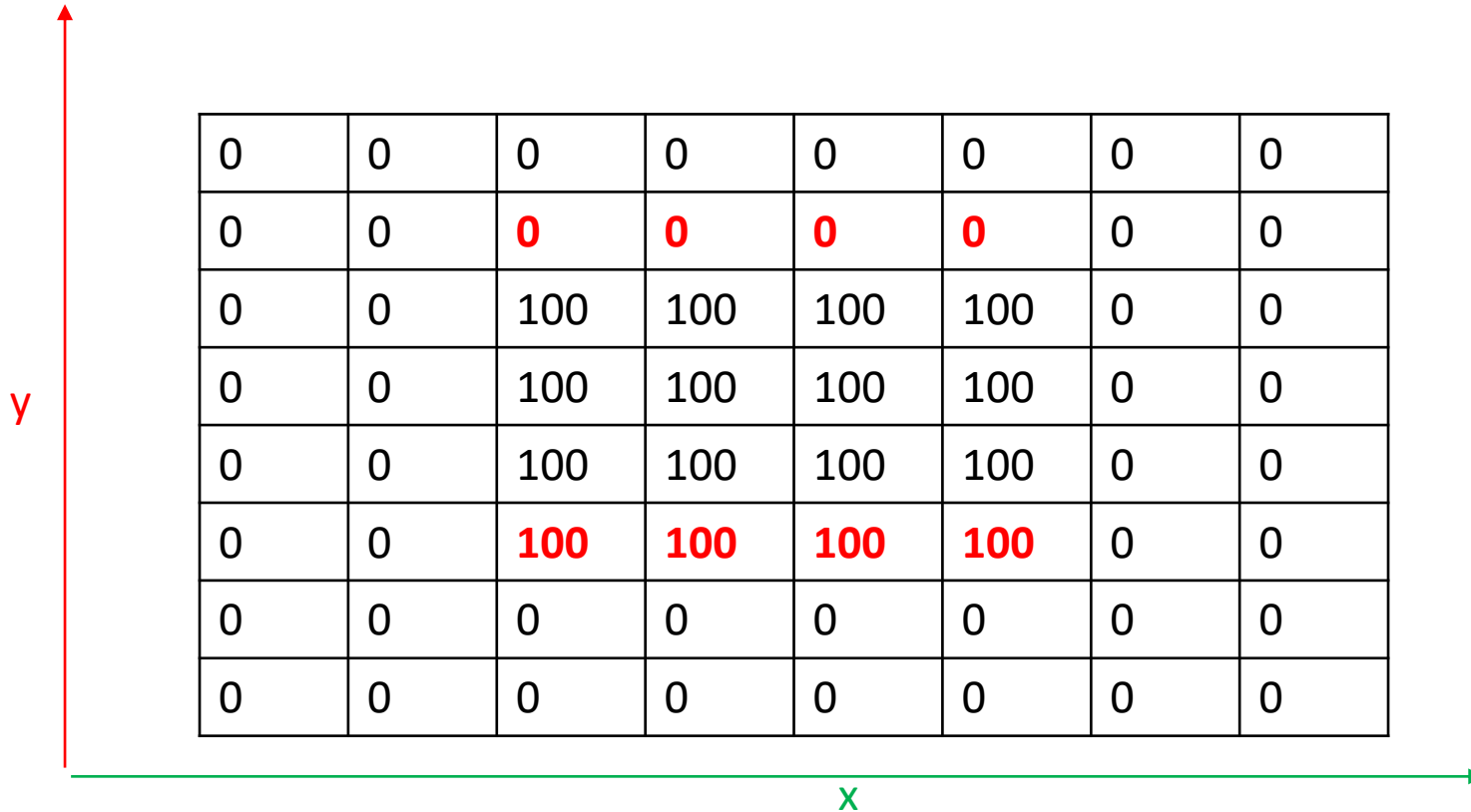
Edge detection(Gradient Magnitude)



∇y

Horizontal Edge

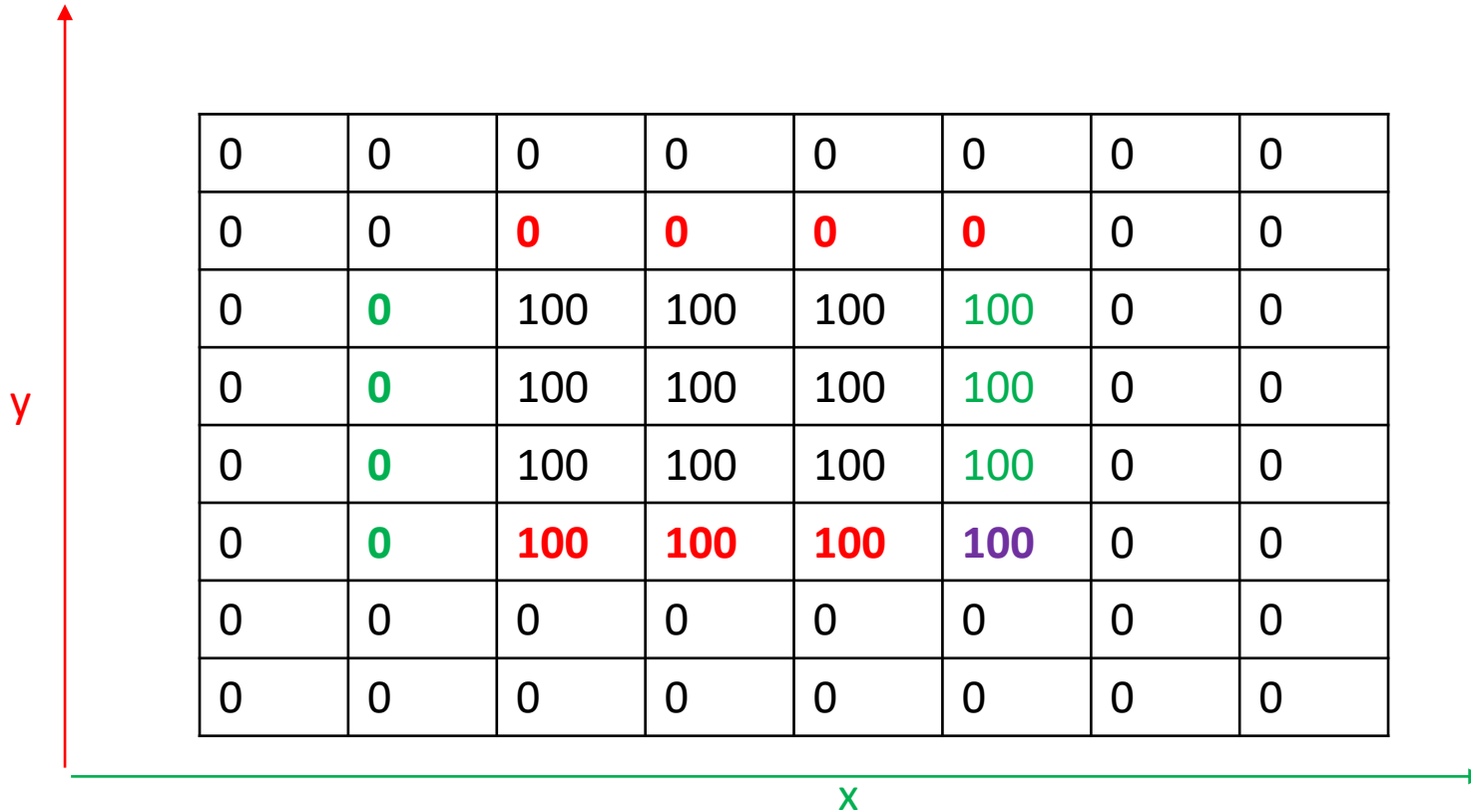
Edge detection(Gradient Magnitude)



∇y

Horizontal Edge

Edge detection(Gradient Magnitude)



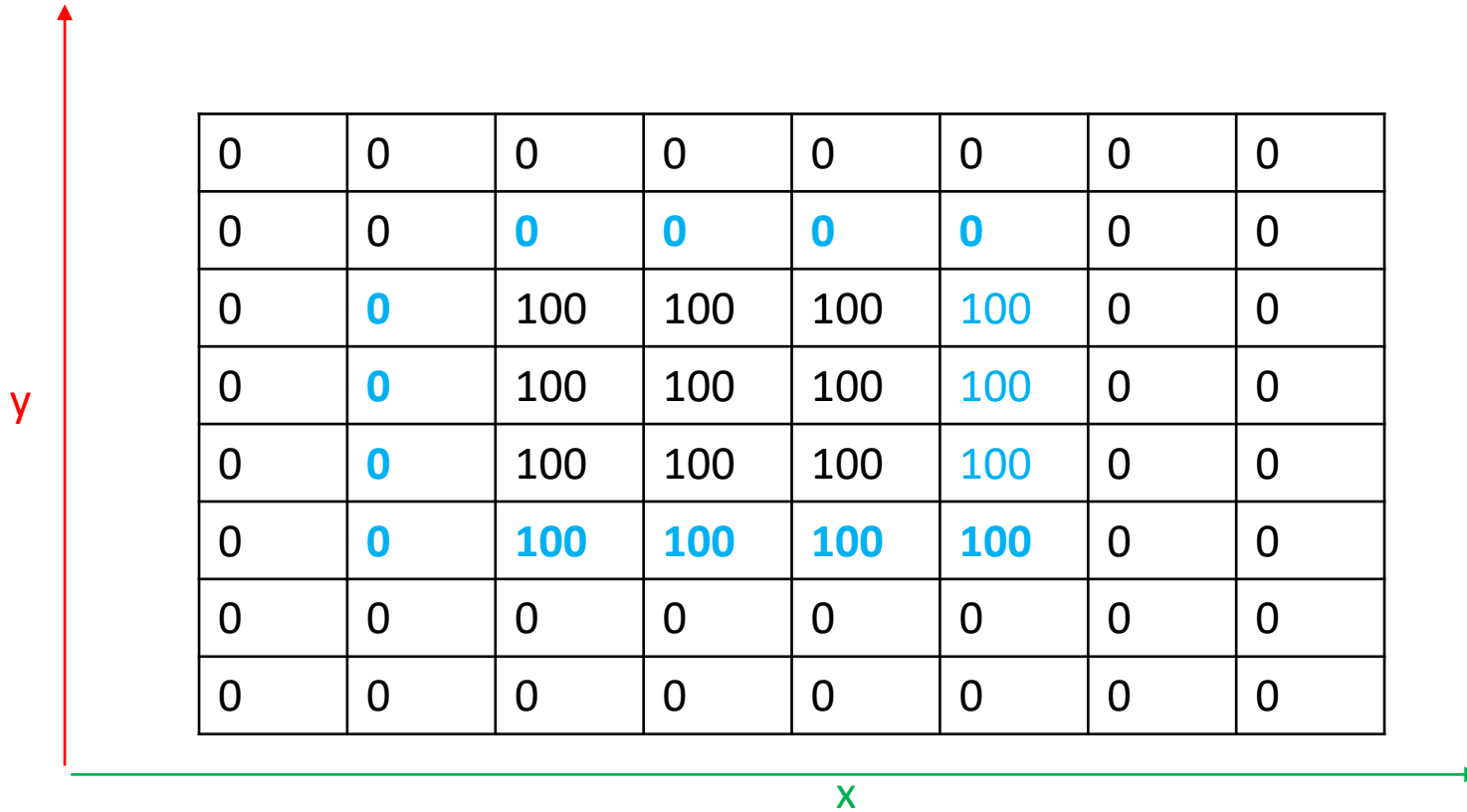
∇y

Horizontal Edge

∇x

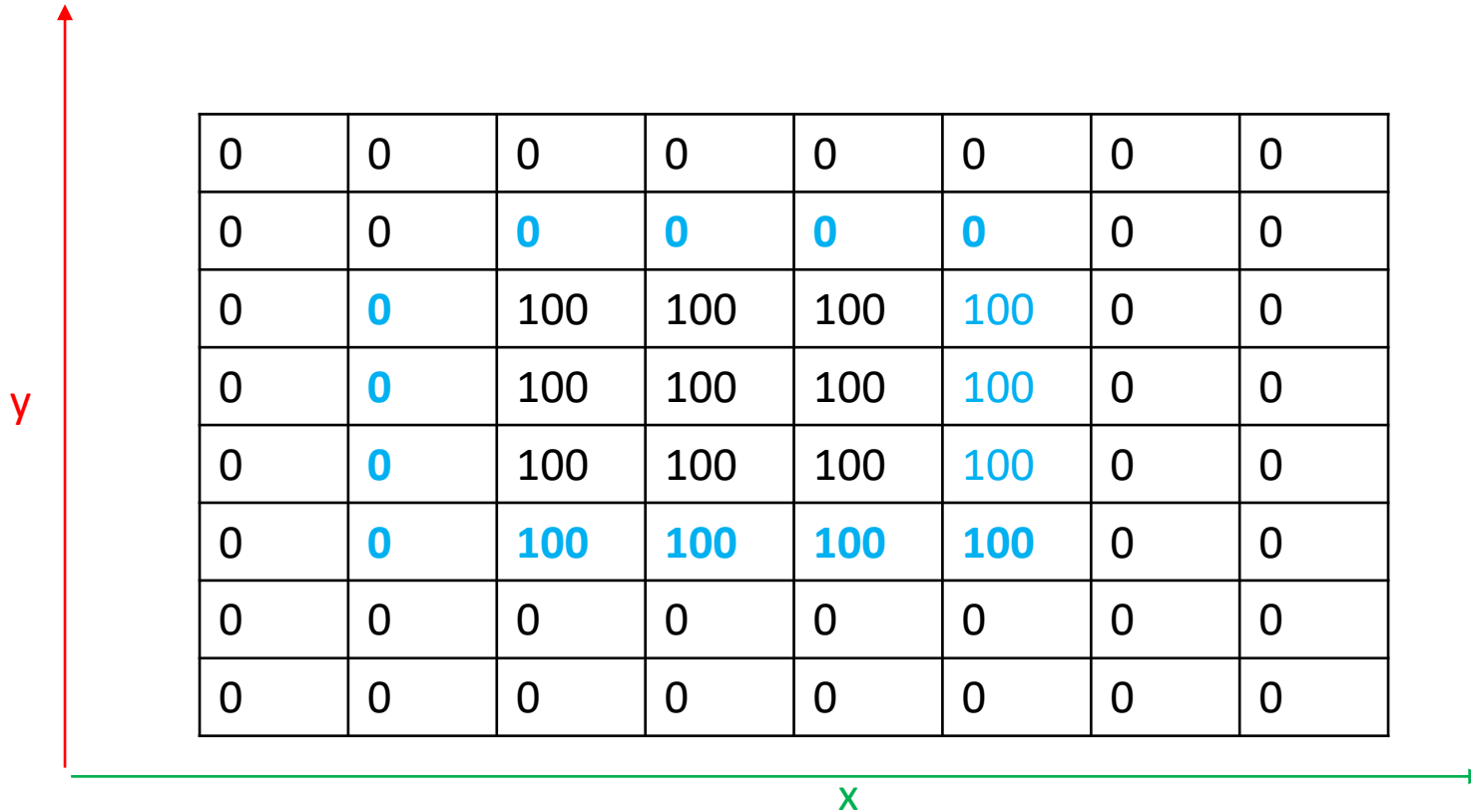
Vertical Edge

Edge detection(Gradient Magnitude)



$$\nabla y + \nabla x$$

Edge detection(Gradient Magnitude)



$$\nabla y + \nabla x$$

-1
1

-1	1
----	---

Edge detection(Gradient Magnitude)

I

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	100	100	100	100	0	0
0	0	100	100	100	100	0	0
0	0	100	100	100	100	0	0
0	0	100	100	100	100	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$I * [-1 \quad 1]$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	-100	0	0	0	100	0
0	-100	0	0	0	100	0
0	-100	0	0	0	100	0
0	-100	0	0	0	100	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Edge detection(Gradient Magnitude)

I

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	100	100	100	100	0	0
0	0	100	100	100	100	0	0
0	0	100	100	100	100	0	0
0	0	100	100	100	100	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$I * \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

0	0	0	0	0	0	0	0
0	0	-100	-100	-100	-100	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	100	100	100	100	0	0
0	0	0	0	0	0	0	0

Edge detection(Gradient Magnitude)

I

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	100	100	100	100	0	0
0	0	100	100	100	100	0	0
0	0	100	100	100	100	0	0
0	0	100	100	100	100	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

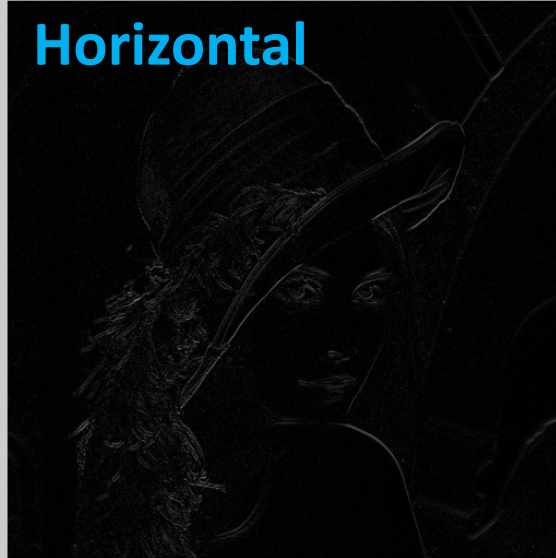
$$abs(I * \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}) + abs(I * \begin{bmatrix} -1 & 1 \end{bmatrix})$$

0	0	0	0	0	0	0	0
0	0	100	100	100	100	0	0
0	100	0	0	0	100	0	0
0	100	0	0	0	100	0	0
0	100	0	0	0	100	0	0
0	100	100	100	100	200	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

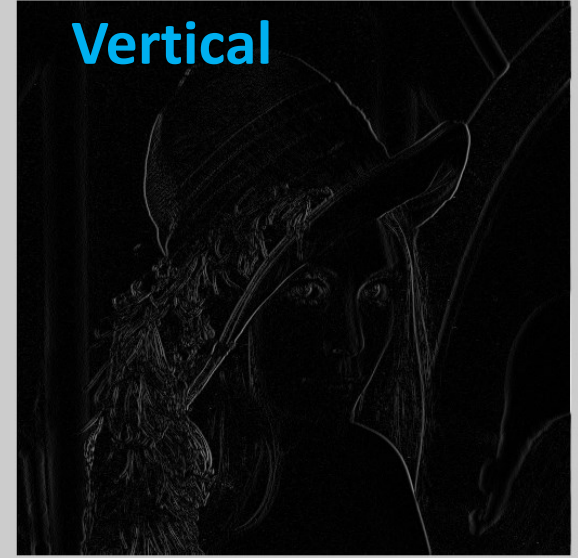
Edge detection(Gradient Magnitude)



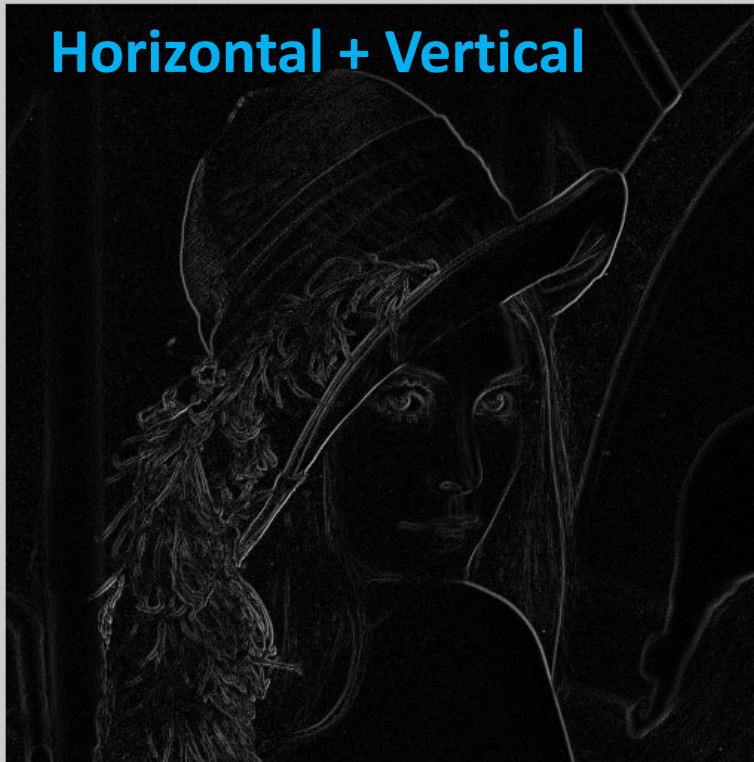
Horizontal



Vertical



Horizontal + Vertical



Edge detection(Prewitt)

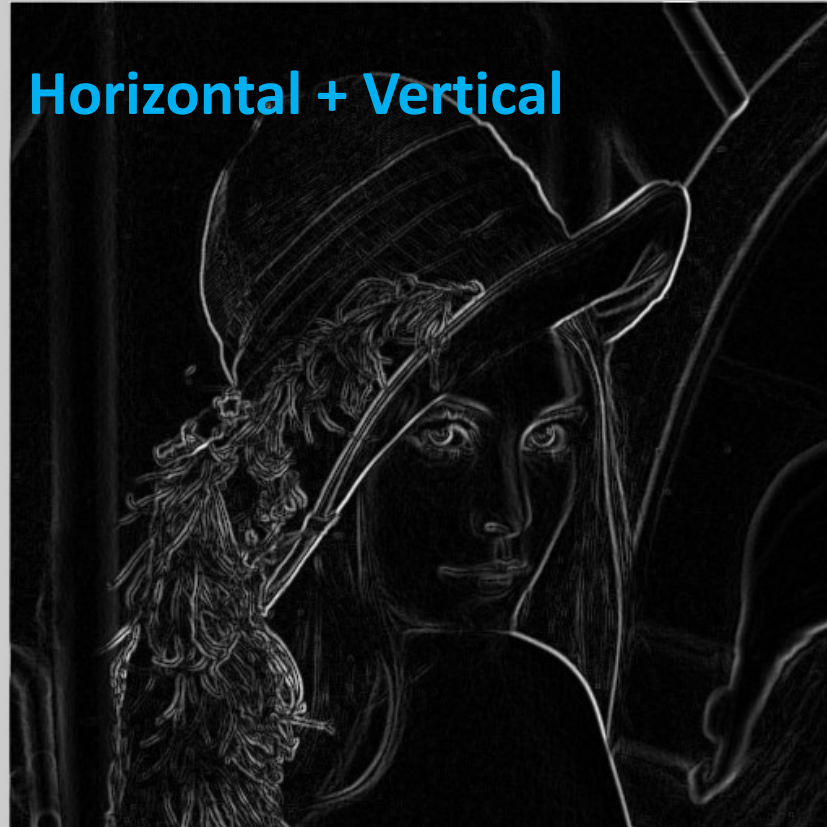
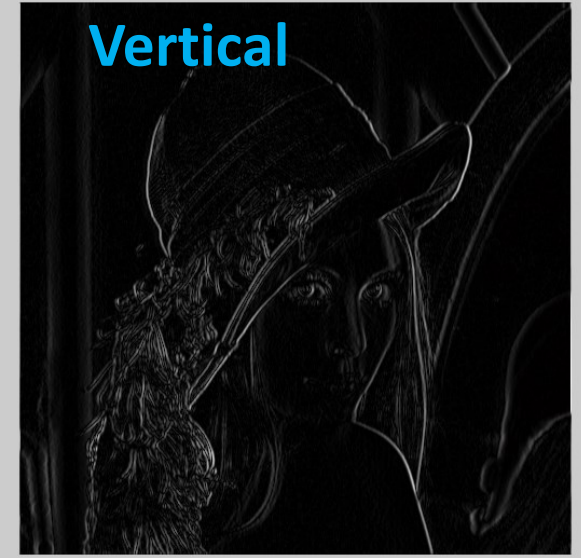
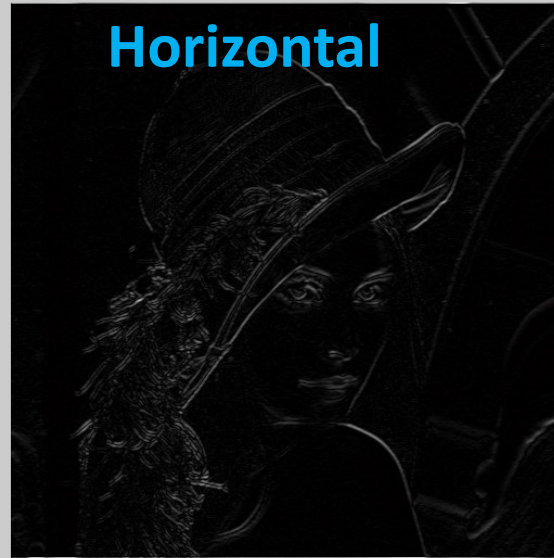
1	1	1
0	0	0
-1	-1	-1

Prewitt Horizontal

1	0	-1
1	0	-1
1	0	-1

Prewitt Vertical

Edge detection(Prewitt)



Edge detection(Sobel)

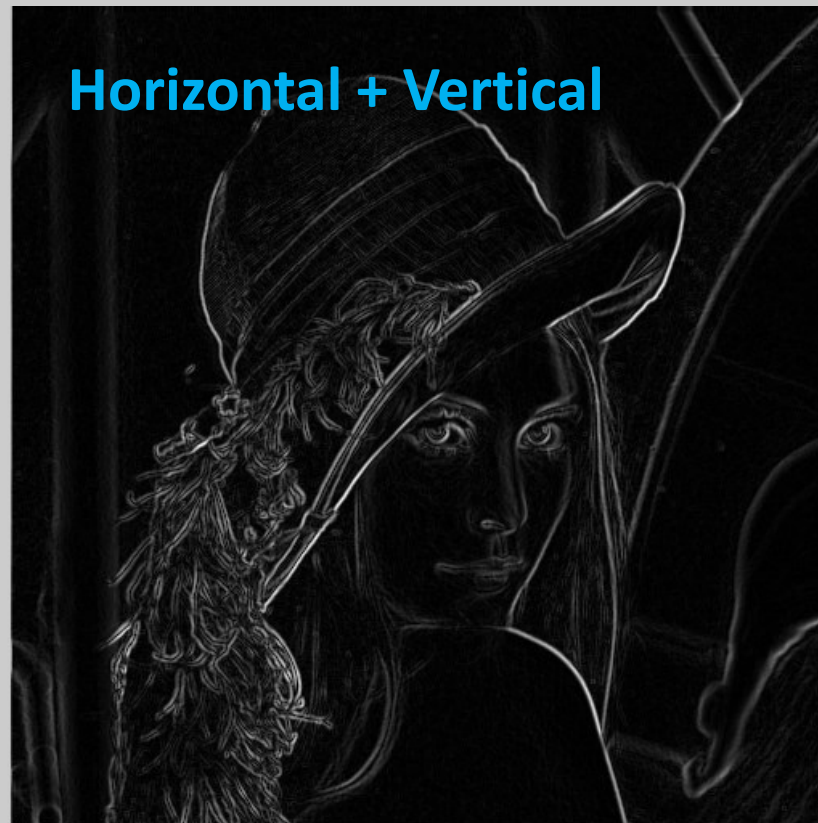
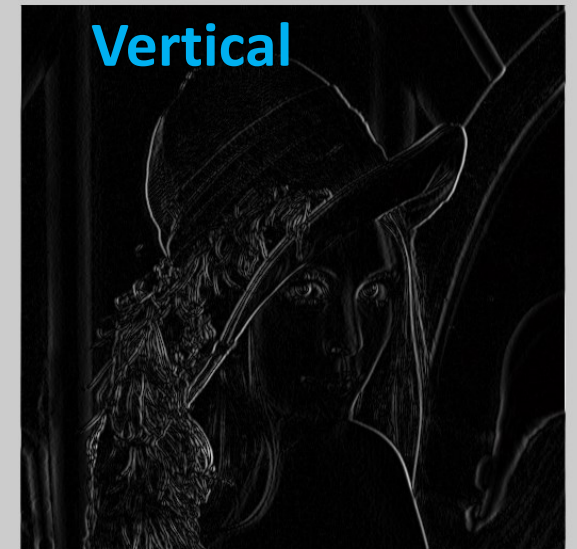
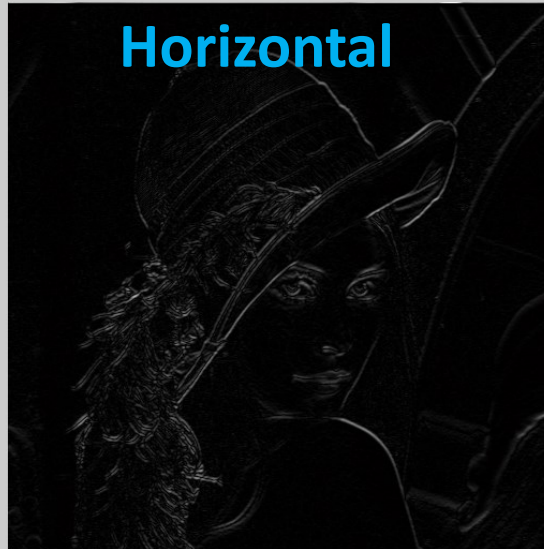
1	2	1
0	0	0
-1	-2	-1

Sobel Horizontal

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

Sobel Vertical

Edge detection(Sobel)



Edge detection(Robert)

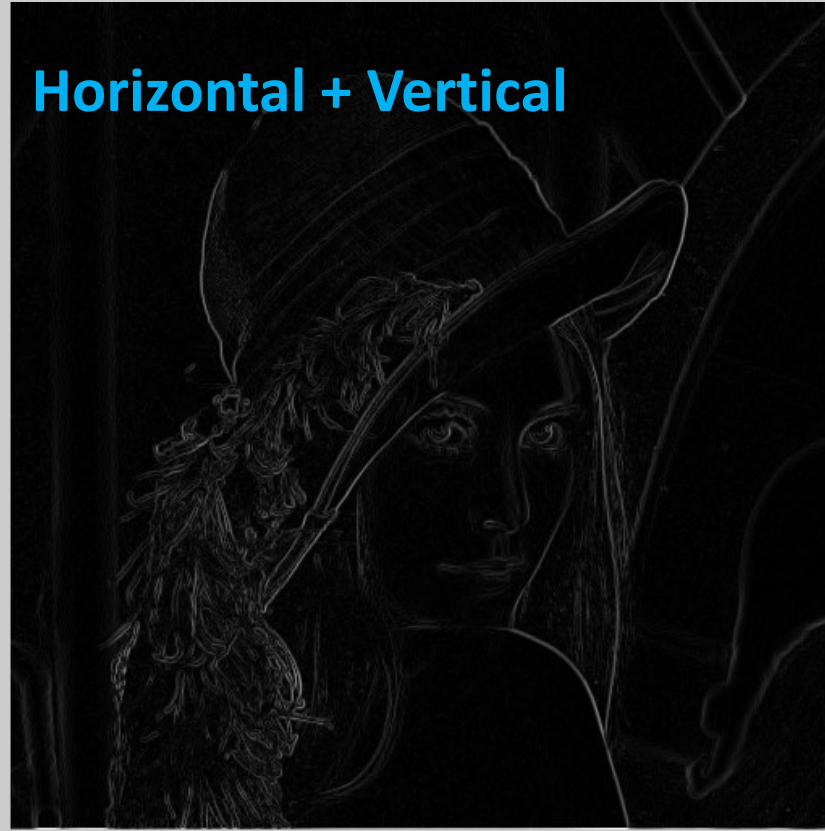
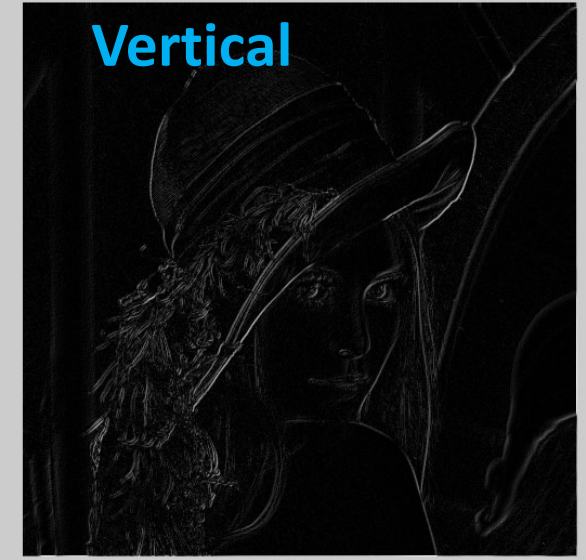
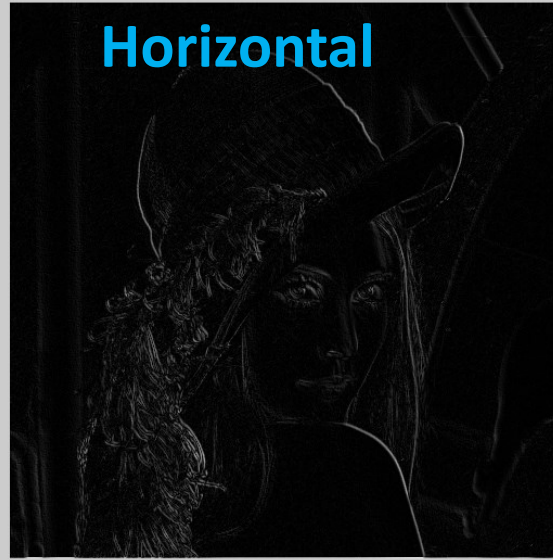
0	1
-1	0

Robert Horizontal

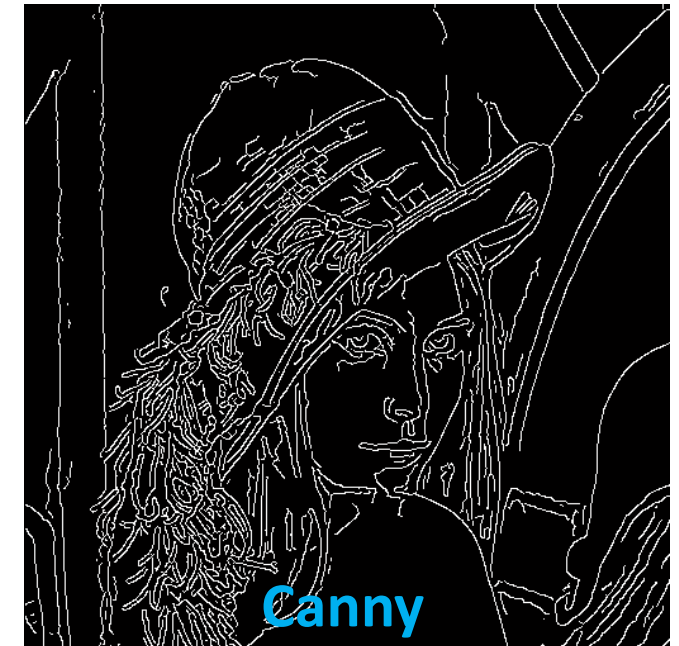
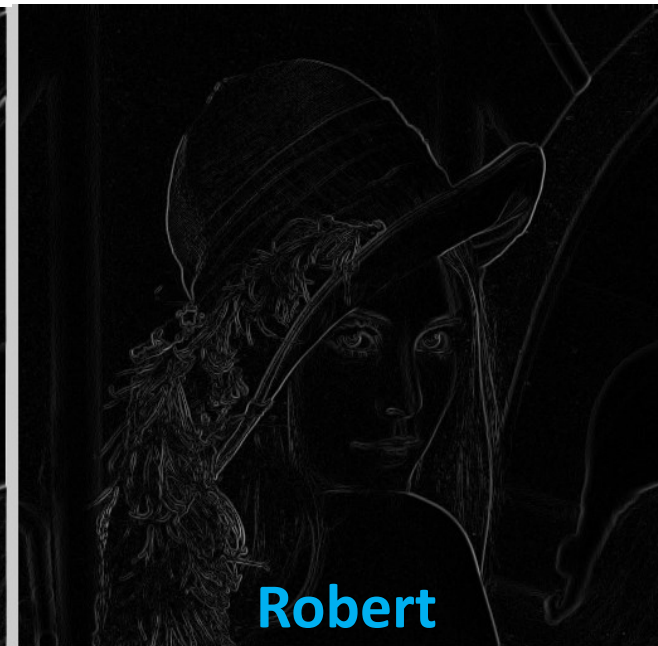
1	0
0	-1

Robert Vertical

Edge detection(Robert)



Edge detection

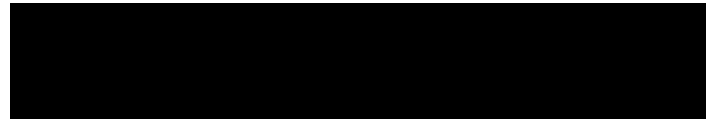


Thresholding

Thresholding is the simplest method of [image segmentation](#). From a [grayscale](#) image, thresholding can be used to create [binary images](#)



$$I_{binary}(j) = \begin{cases} 1 & \text{where } I(j) \geq \theta \\ 0 & \text{where } I(j) < \theta \end{cases}$$



Thresholding

Thresholding



Original Image



Binary Image



$\theta = 0.1$



$\theta = 0.2$



$\theta = 0.3$

Question !

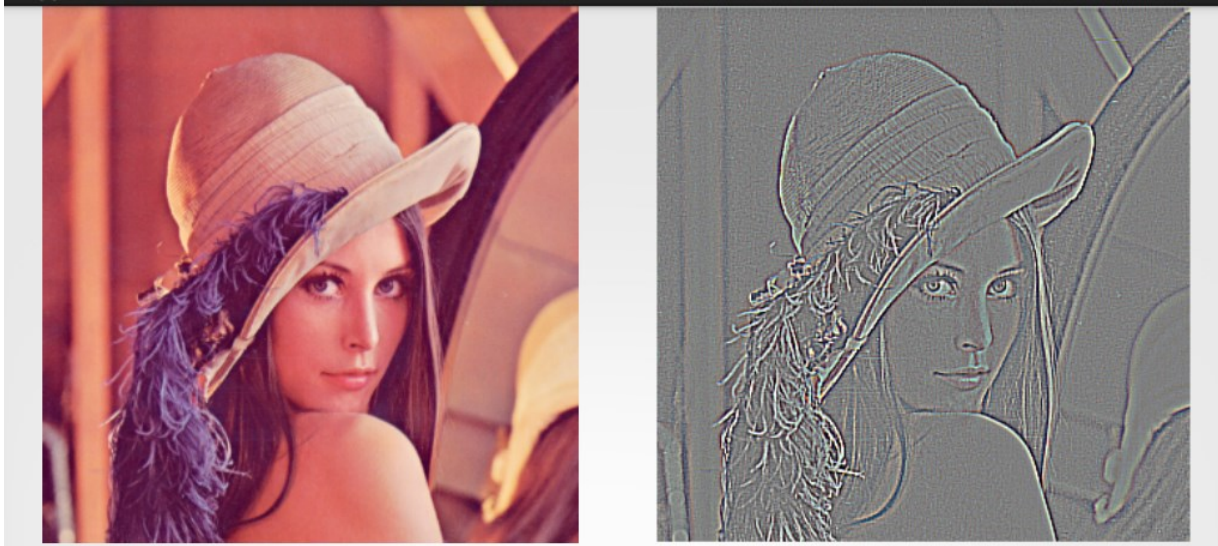
164	239	53	50	79
97	223	77	58	235
207	140	120	44	110
136	159	59	58	47
89	150	215	111	231

I

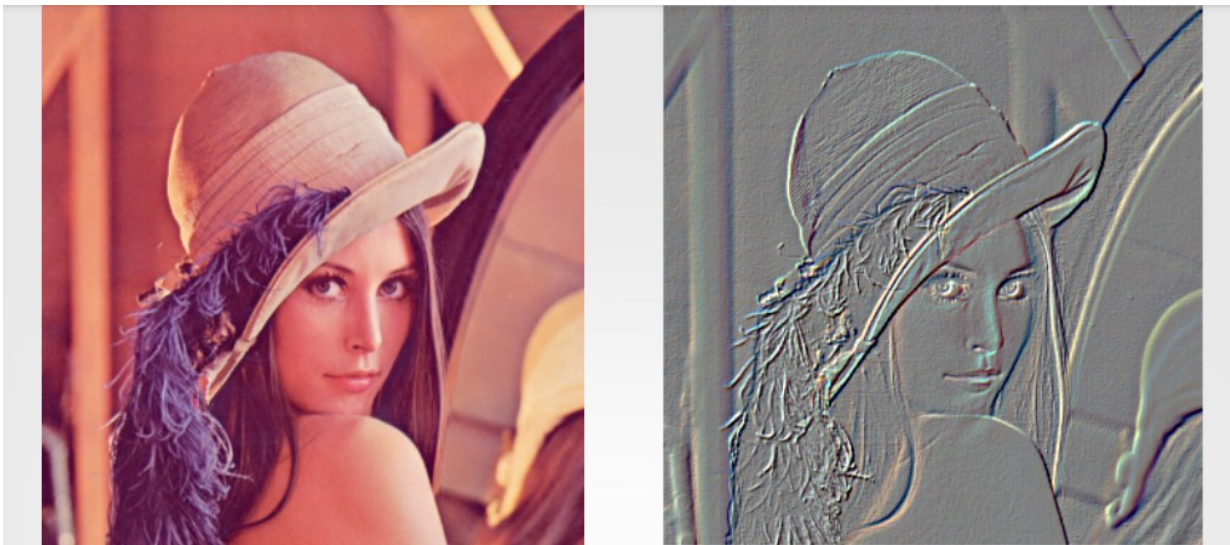
Please convert uint8 image *I* to
binary image using threshold
 $\theta = 127$

Sol..

1	1	0	0	0
0	1	0	0	1
1	1	0	0	0
1	1	0	0	0
0	1	1	0	1



Emboss



Engrave

Matlab image processing function

Kernel Generator:

```
H=fspecial('kernel type',param1,param2);
```

2D Convolution:

```
Im=conv2(im,H);
```

Image filter:

```
Im=imfilter(im,H);
```

Threshold:

```
Im=im2bw(im,threshold);
```

Image filter kernel

[https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))

