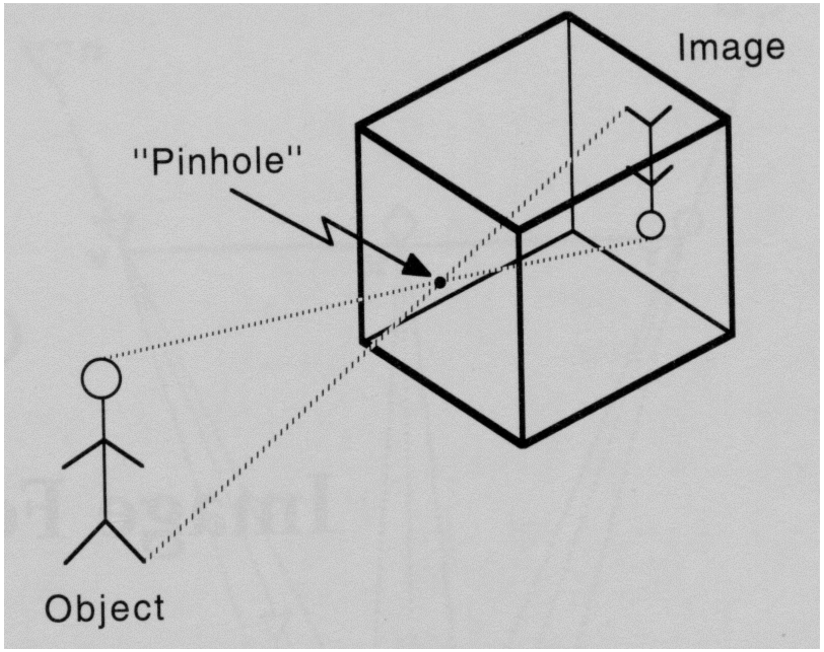
Image Processing

Image Processing

- **Image Fundamental**
- **Digital Image Processing**

Pinhole camera

- This is the simplest device to form an image of a 3D scene on a 2D surface.
- Straight rays of light pass through a “pinhole” and form an inverted image of the object on the image plane.

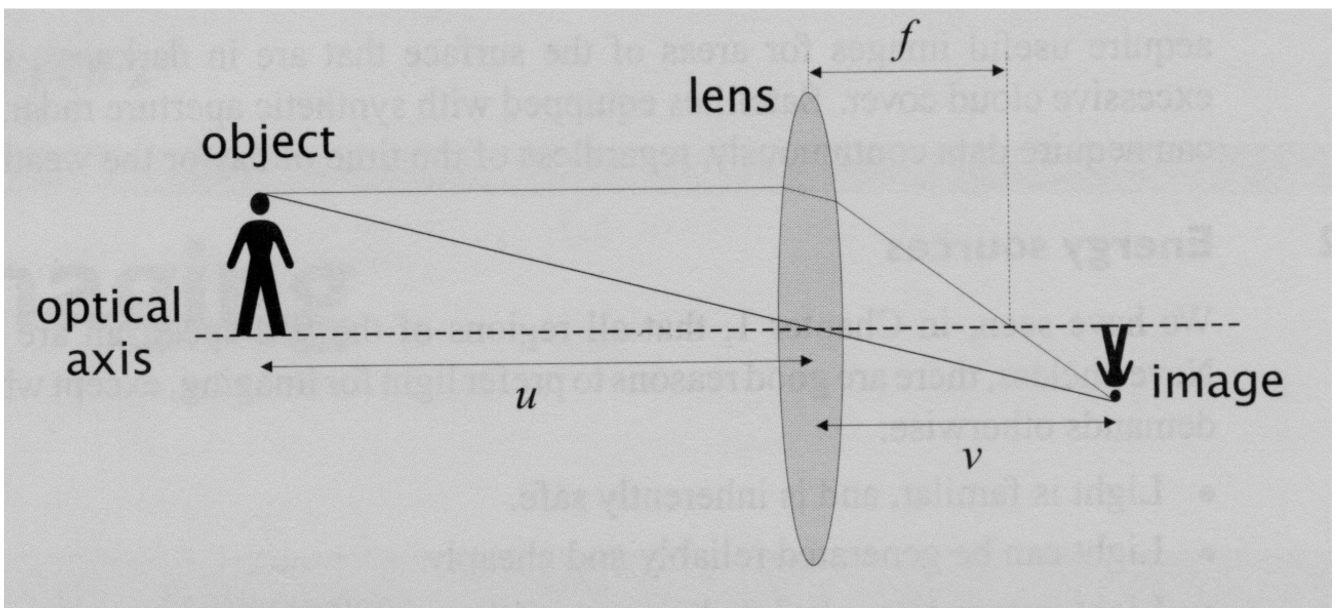


$$x = \frac{fX}{Z}$$

$$y = \frac{fY}{Z}$$

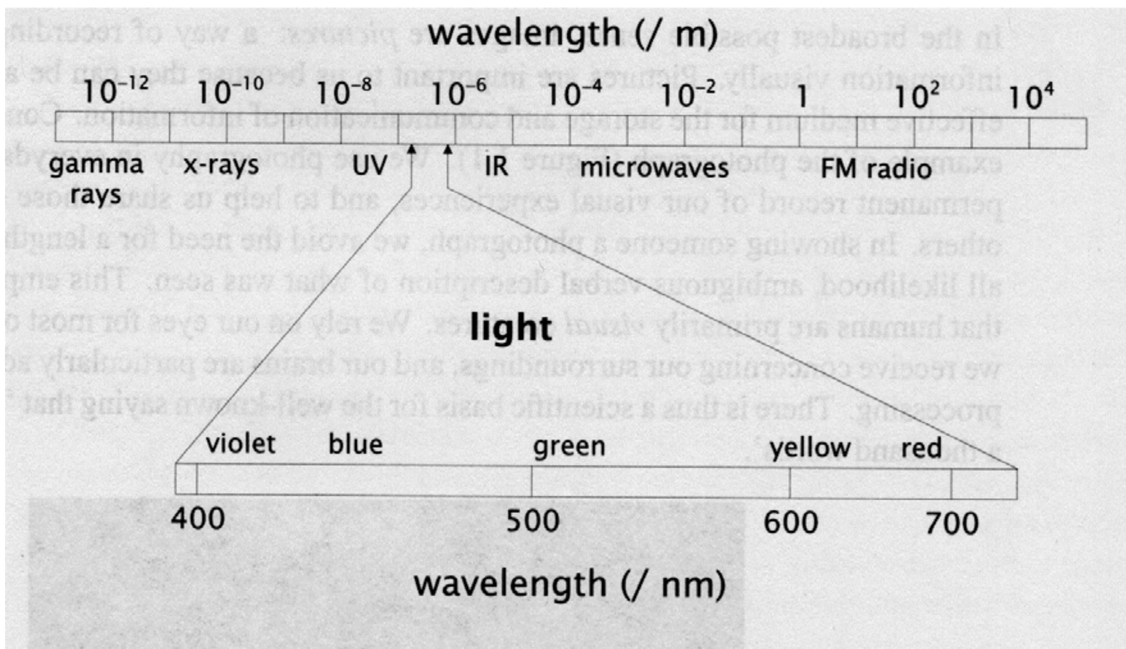
Camera optics

- In practice, the aperture must be larger to admit more light.
- Lenses are placed to in the aperture to **focus** the bundle of rays from each scene point onto the corresponding point in the image plane



What is light?

- The visible portion of the **electromagnetic** (EM) spectrum.
- It occurs between wavelengths of approximately 400 and 700 nanometers.



Short wavelengths

- Different wavelengths of radiation have different properties.
- The **x-ray** region of the spectrum, it carries sufficient energy to penetrate a significant volume or material.

Long wavelengths

- Copious quantities of **infrared** (IR) radiation are emitted from warm objects (e.g., locate people in total darkness).

Sonic images

- Produced by the reflection of sound waves off an object.
- High sound frequencies are used to improve resolution.

CCD (Charged-Coupled Device) cameras

- Tiny solid state cells convert light energy into electrical charge.
- The image plane acts as a digital memory that can be read row by row by a computer.

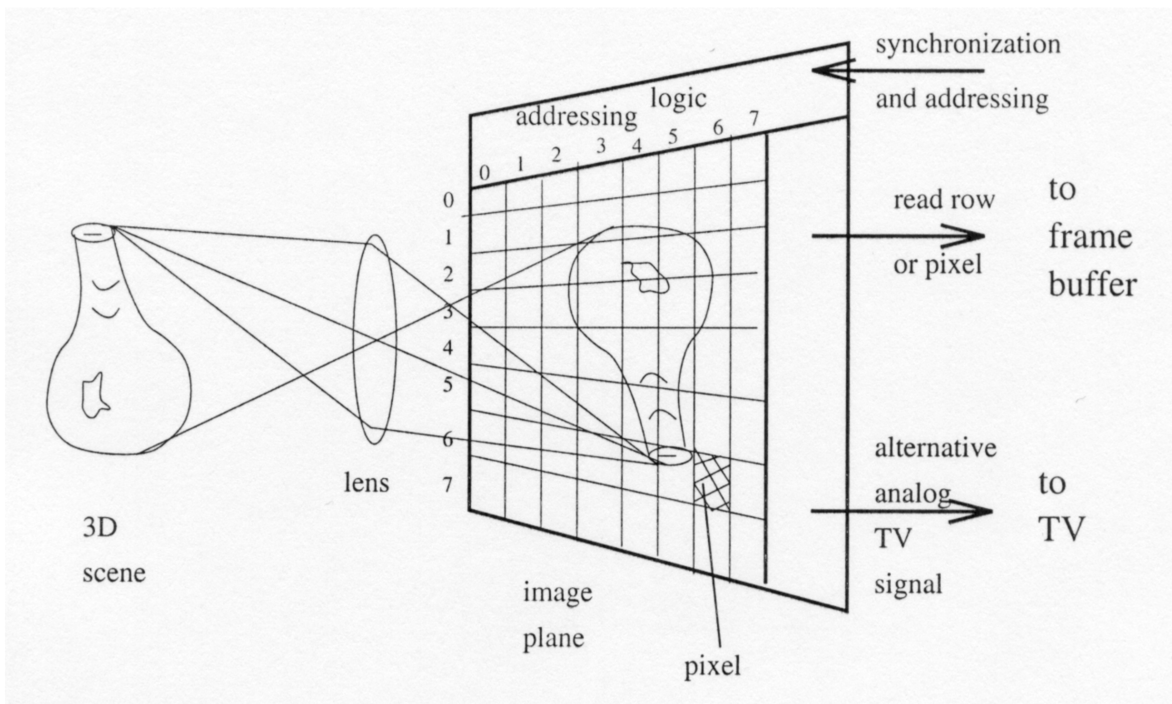
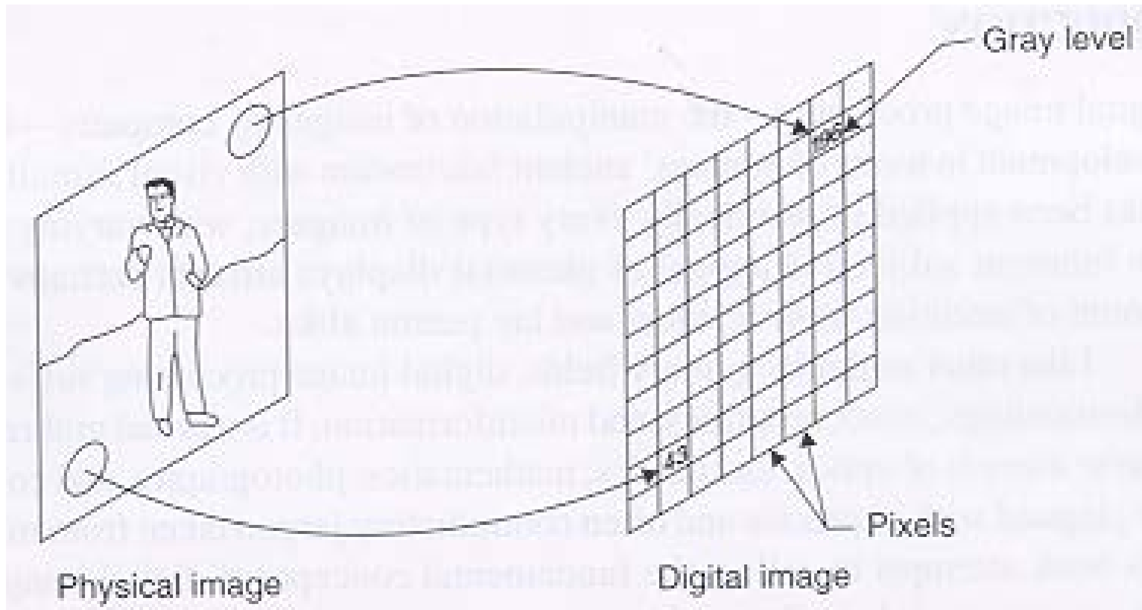
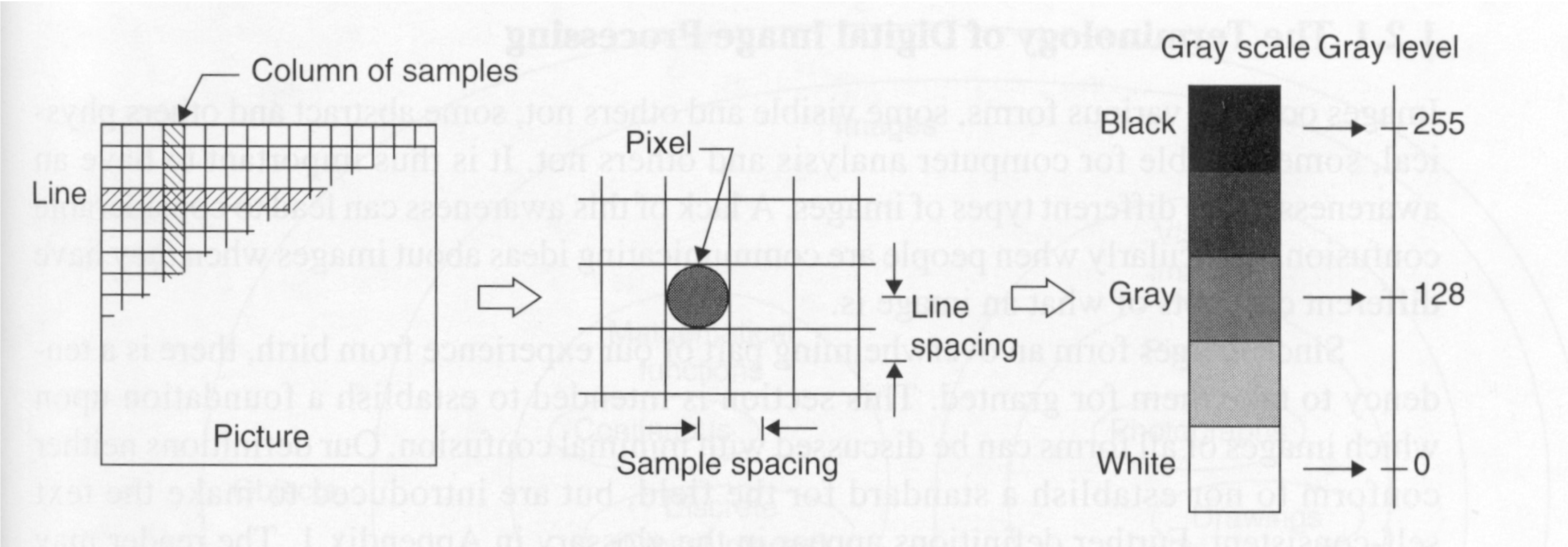


Image digitization



- **<u>Sampling</u>** means measuring the value of an image at a finite number of points.
- **<u>Quantization</u>** is the representation of the measured value at the sampled point by an integer.

Image digitization



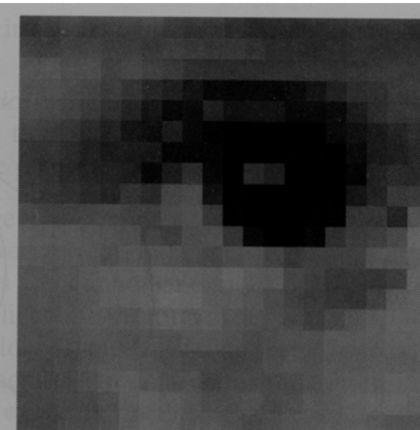
Digital image

- An image is represented by a rectangular array of integers.
- An integer represents the brightness or darkness of the image at that point.
- N: # of rows, M: # of columns

$$
\begin{array}{cccc}
f(0,0) & f(0,1) & \ldots & f(0,M-1) \\
f(1,0) & f(1,1) & \ldots & f(1,M-1) \\
\ldots & \ldots & \ldots & \ldots \\
f(N-1,0) & f(N-1,1) & \ldots & f(N-1,M-1)
\end{array}
$$

How are images represented in the computer?

Grey images
(0 to 255,
8 bits*1channel)



117	125	133	127	130	130	133	121	116	115	100	91	93	94	99	103	112	105	109	106
134	133	138	138	132	134	130	133	128	123	121	113	106	102	99	106	113	109	109	113
146	147	138	140	125	134	124	115	102	96	93	94	99	96	99	100	103	110	109	110
144	141	136	130	120	108	88	74	53	37	31	37	35	39	53	79	93	100	109	116
139	136	129	119	102	85	58	31	41	77	51	53	53	33	37	41	69	94	105	108
132	127	117	102	87	57	49	77	42	28	17	15	13	13	17	41	53	69	88	100
124	120	108	94	72	74	72	31	35	31	15	13	15	11	15	13	46	75	83	96
125	115	102	93	88	82	42	79	113	41	19	100	82	11	11	17	31	91	99	100
124	116	109	99	91	113	99	140	144	57	20	20	15	11	15	17	63	87	119	124
136	133	133	135	138	133	132	144	150	120	24	17	15	15	17	20	115	113	88	150
158	157	157	154	149	145	133	127	146	150	116	35	20	19	28	105	124	128	141	171
155	154	156	155	146	155	154	154	147	139	148	150	138	120	128	129	130	151	156	165
150	151	154	162	166	167	169	174	172	167	177	166	164	140	134	120	121	120	127	172
145	149	151	157	165	169	173	179	176	166	166	157	145	136	129	124	120	136	163	168
144	148	153	160	159	158	165	172	165	169	157	151	149	141	130	140	151	162	169	167
144	141	147	155	154	149	156	151	157	157	151	144	147	147	149	159	158	159	166	165
139	140	140	150	153	151	150	146	140	139	138	140	145	151	149	156	156	162	162	161
136	134	138	146	156	164	153	146	145	136	139	139	140	141	149	157	159	161	169	166
136	133	136	135	144	159	168	159	151	142	141	145	139	146	153	156	164	167	172	168
133	129	140	142	146	159	167	165	154	151	146	141	147	154	156	160	161	157	153	154

Color images (0 to 255, 8 bits*3channel)

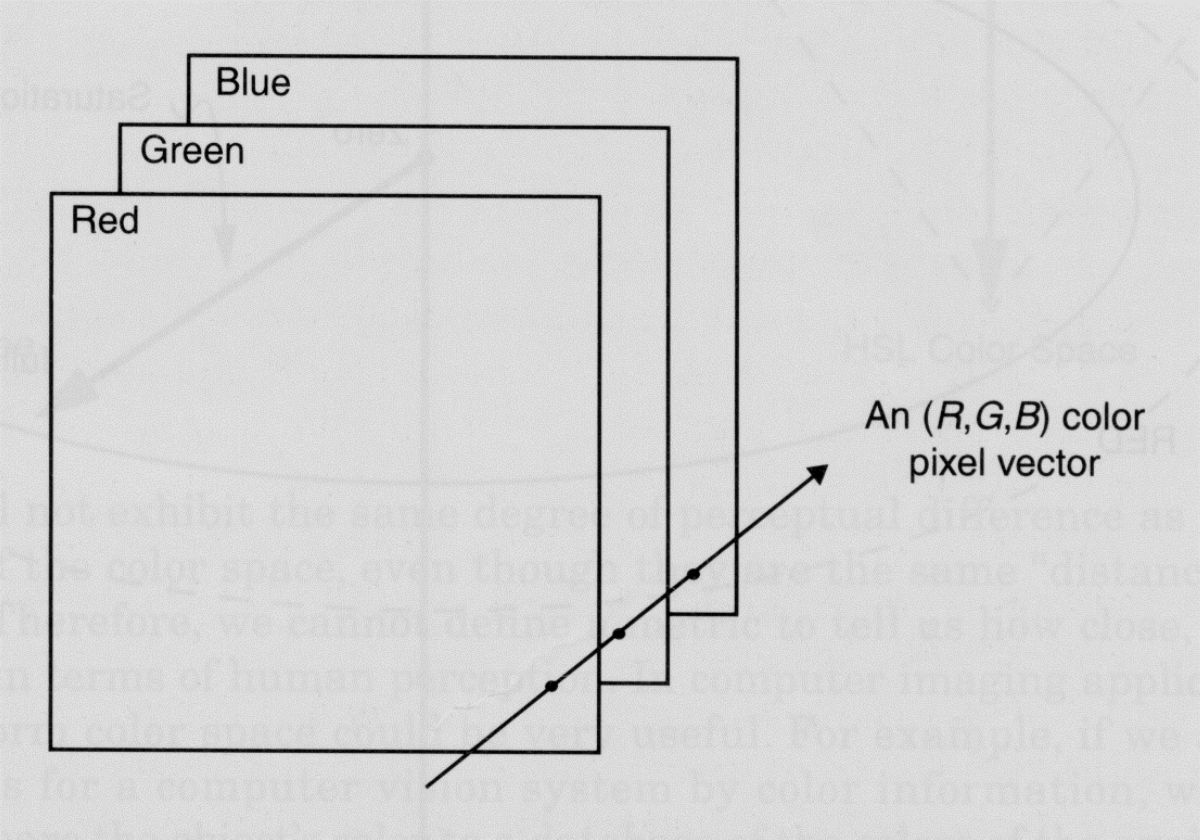
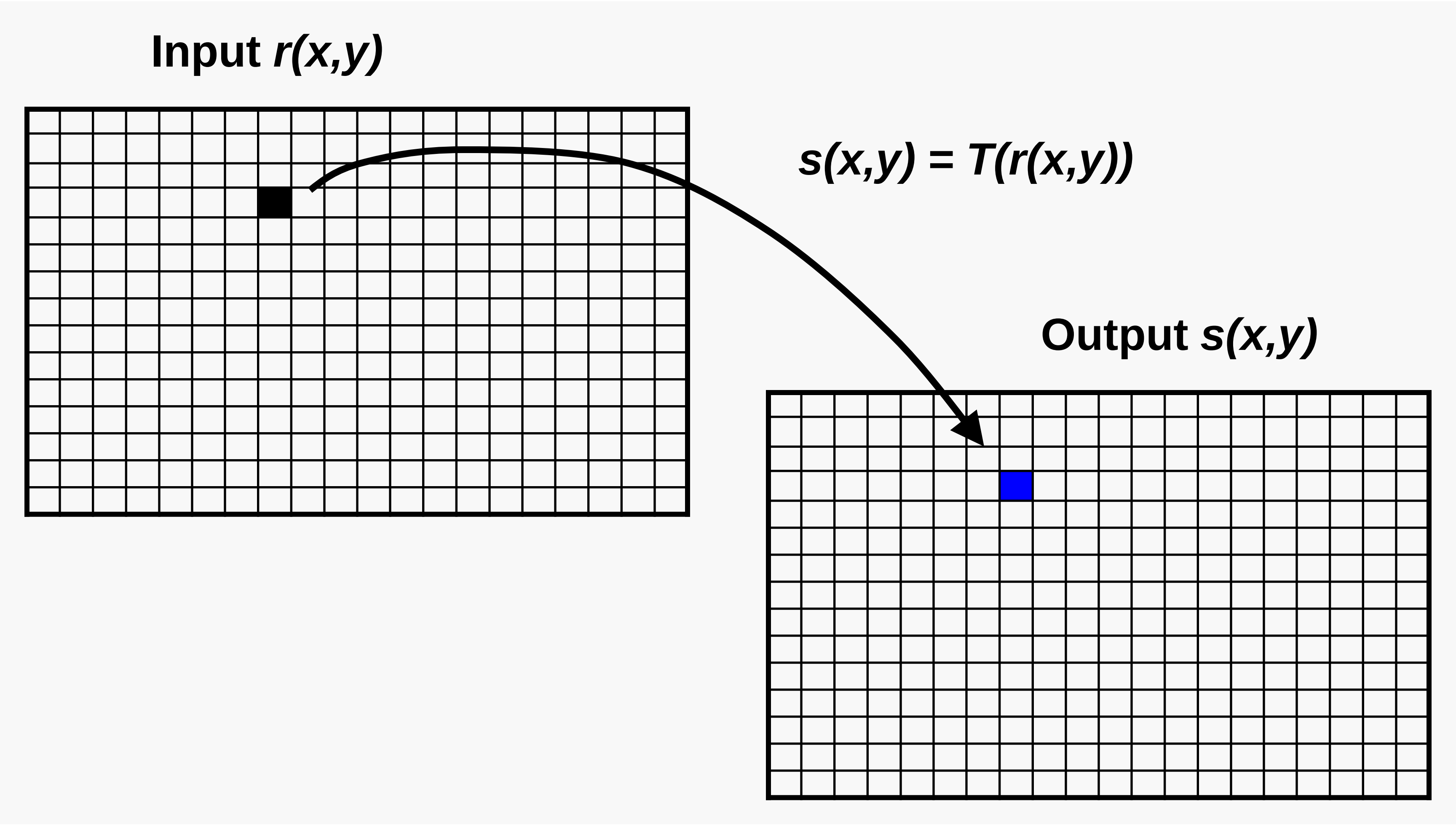


Image Processing

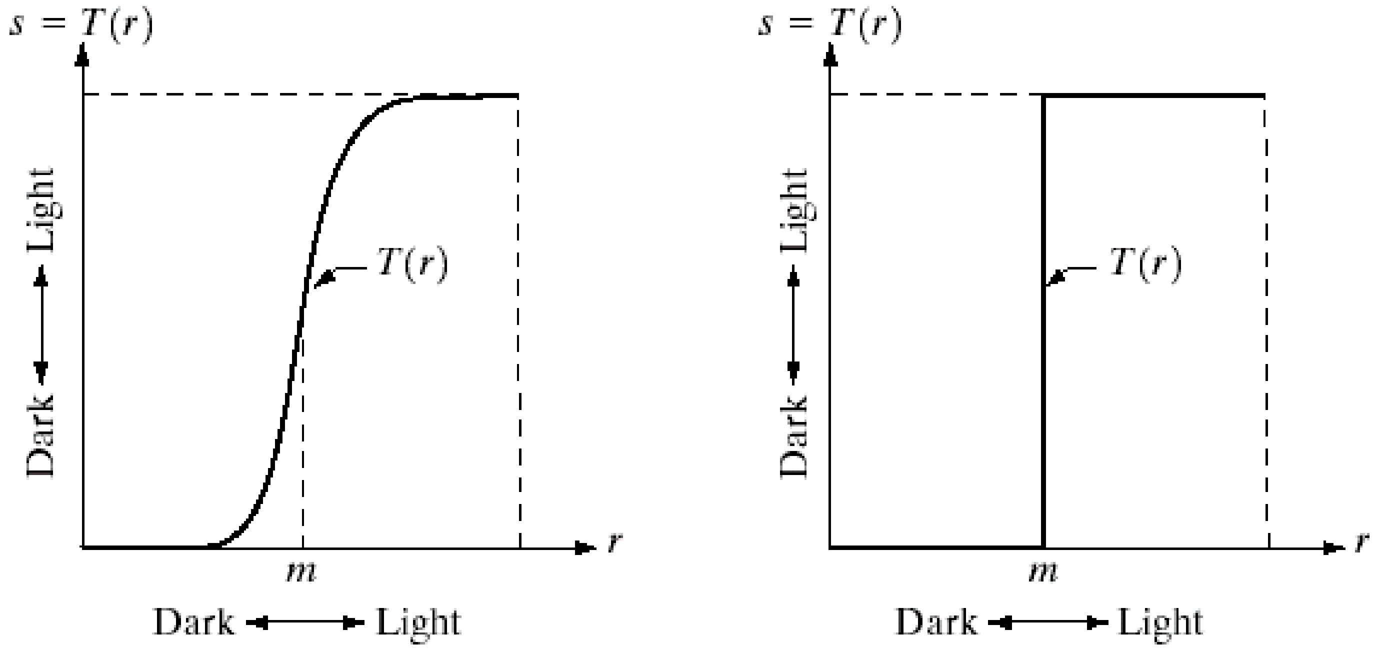
- **Image Fundamental**
- **Digital Image Processing**

Point Processing

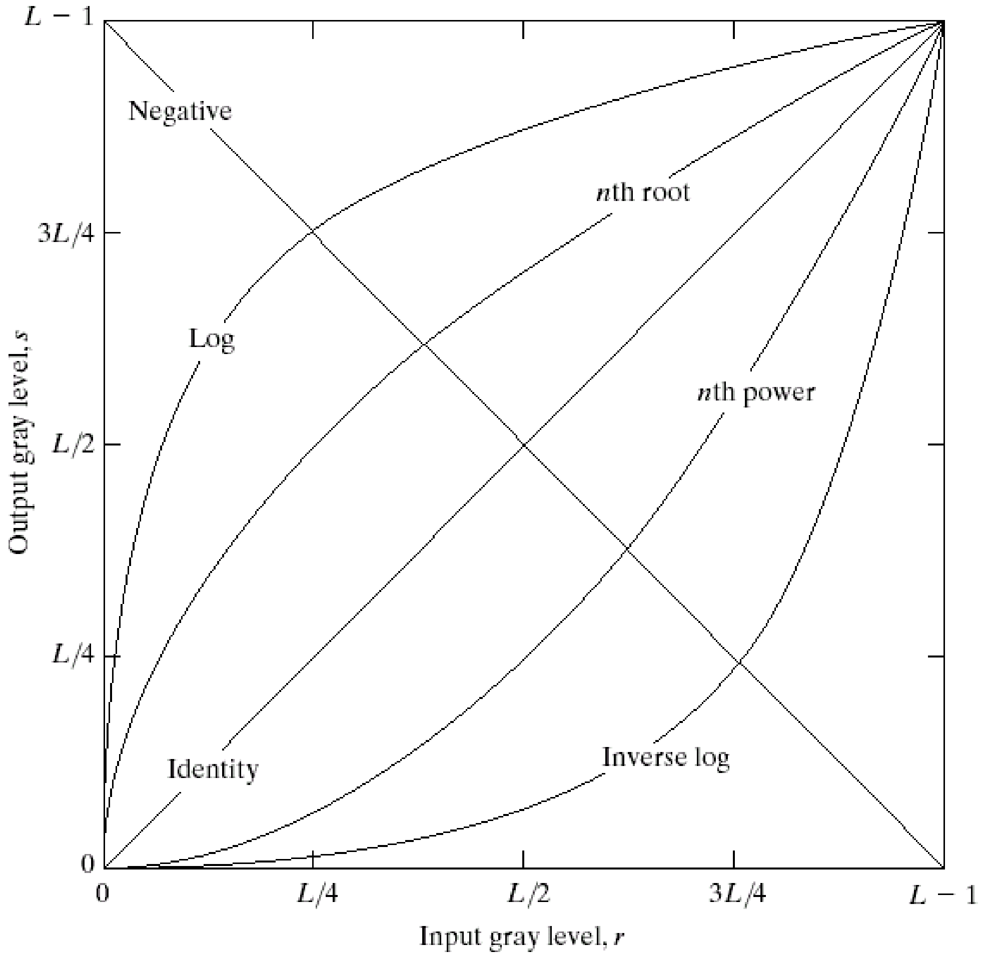


Contrast Enhancement

- Put a suitable curve to map *r* to *s* .
- Curve should be monotonically increasing.



Some Curves



Negative:

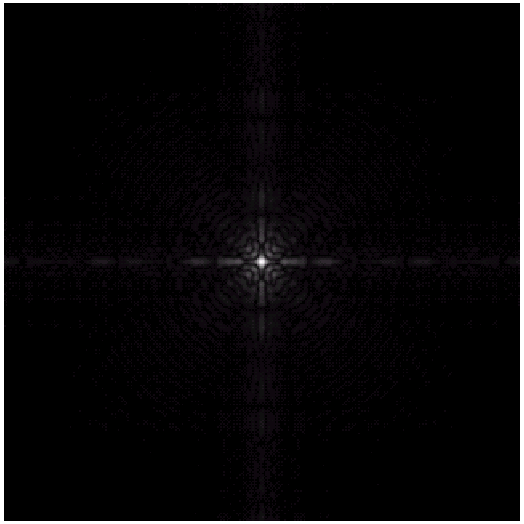
$$s = L - 1 - r$$

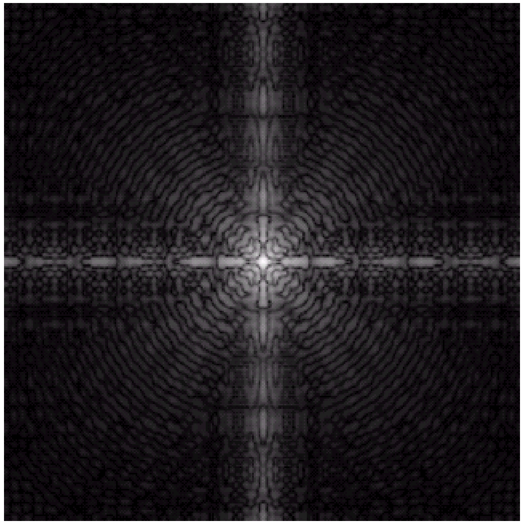
Log:

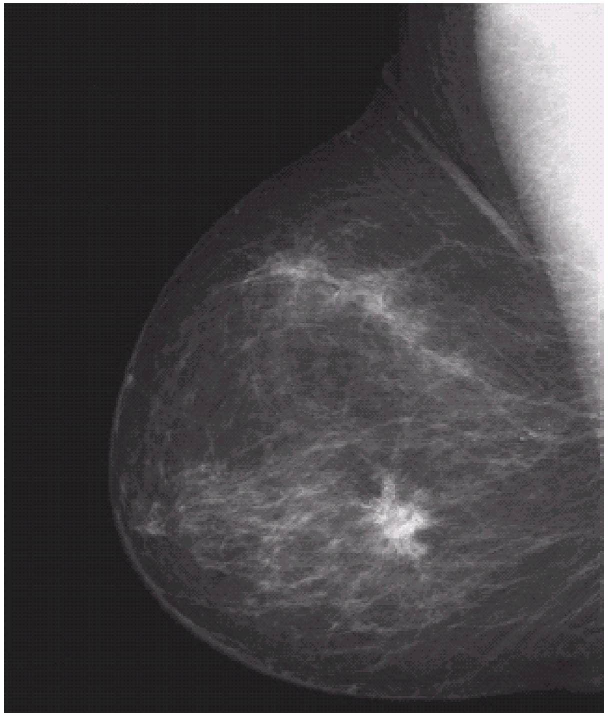
$$s = c \log(1 + r)$$

Examples

Fourier spectra:
log image





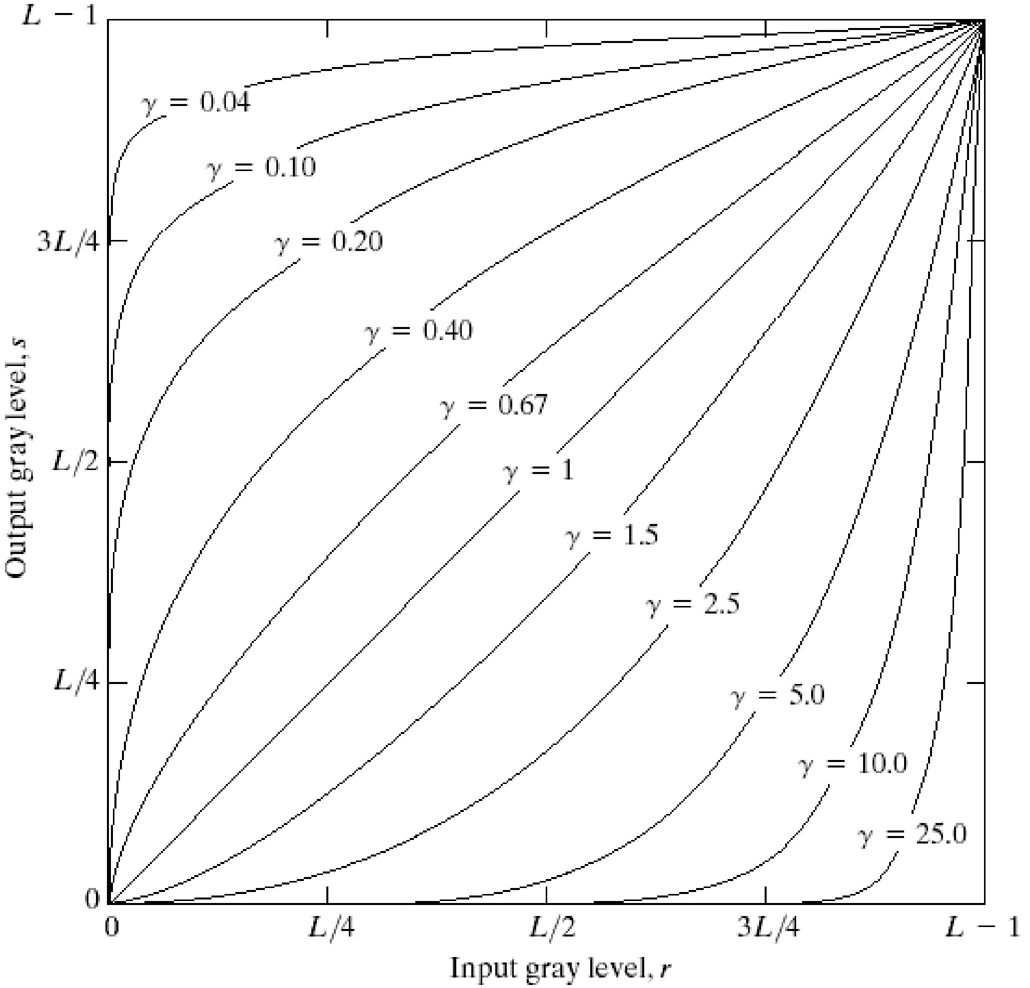


Mammogram:
negative image

Power Curve

Power:

$$s = c\, r^{\gamma}$$

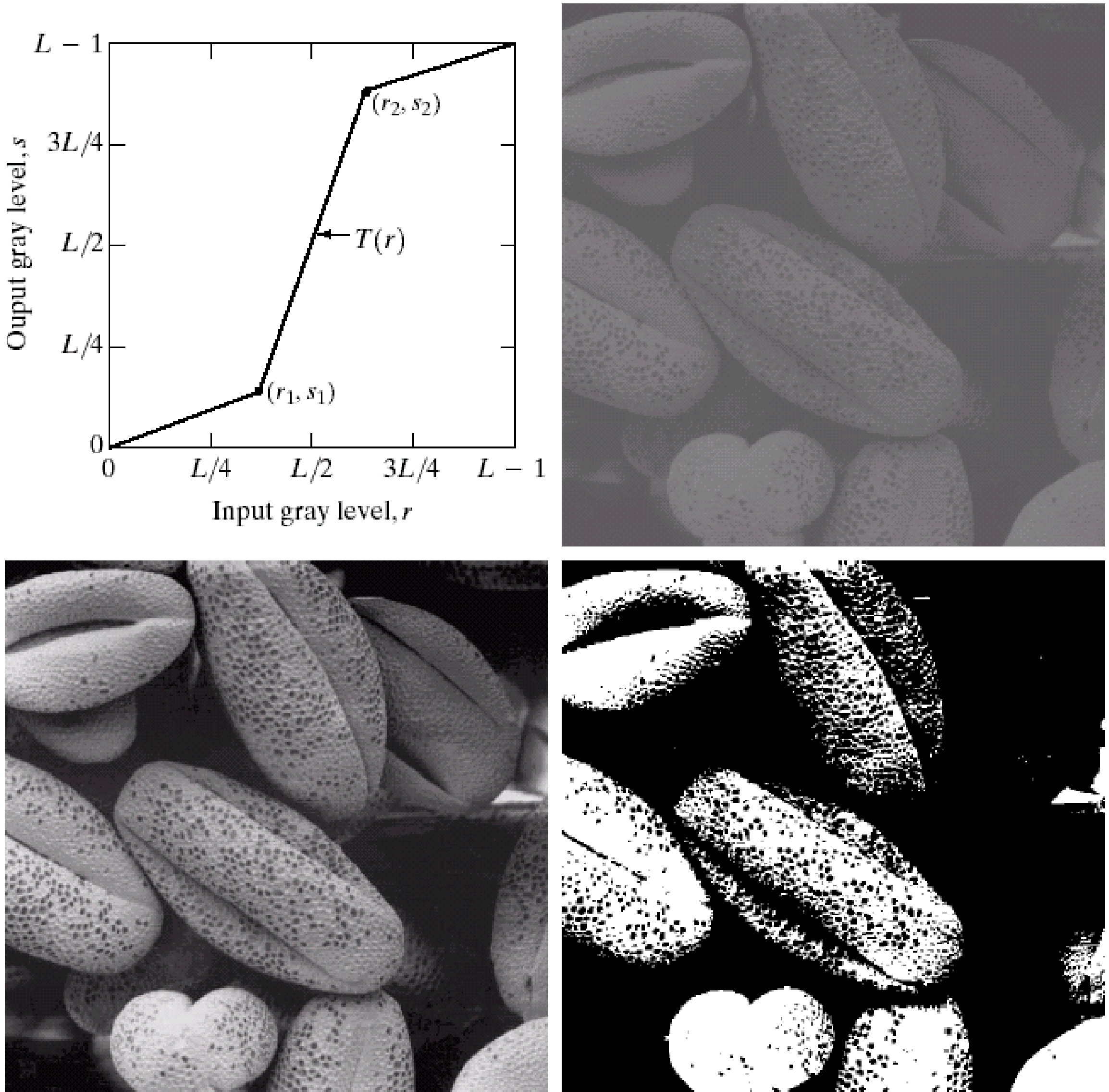


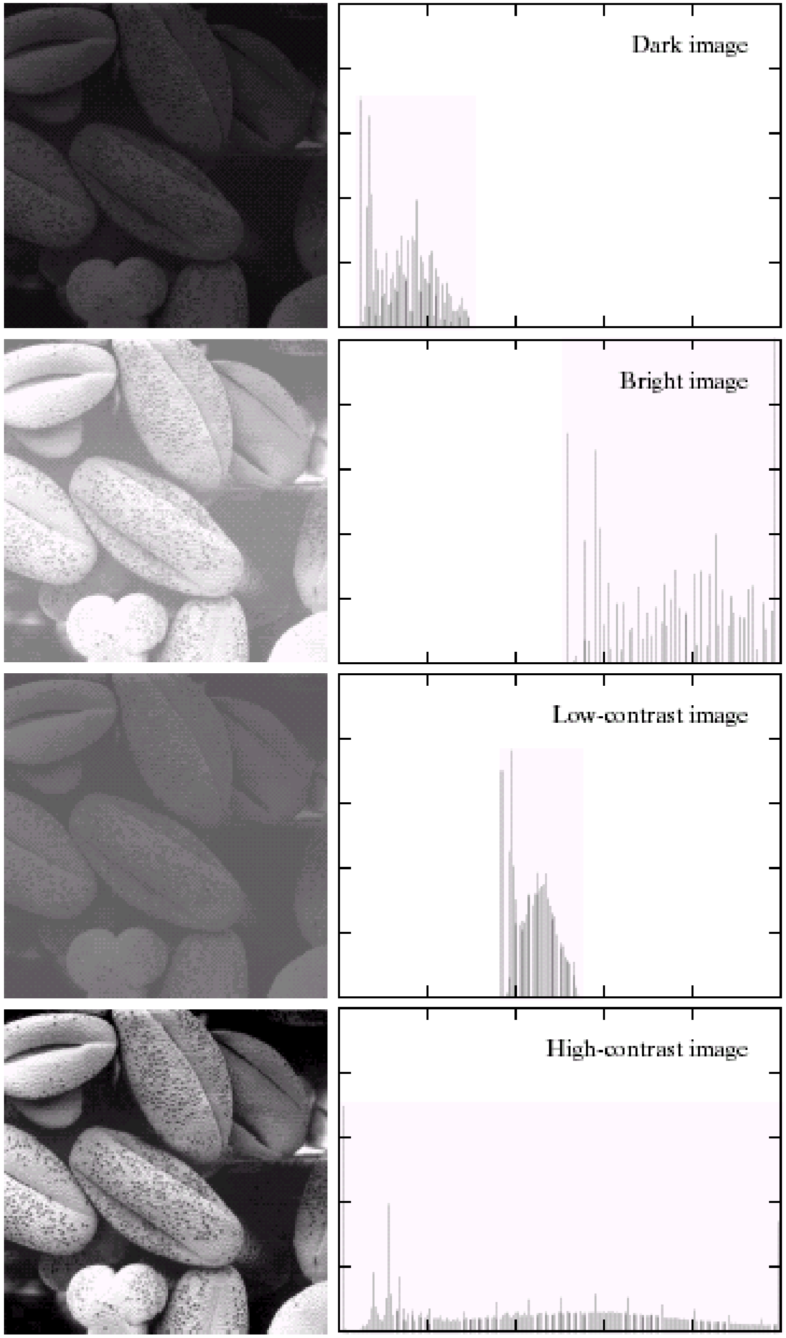
Examples

a b
c d

FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

Piecewise Linear Curves





Historgram

- Number of pixels with a particular value *v*.
  - for all *v*
- Becomes probability density function (pdf) when divided by total num. of pixels *n.*

Histogram Equalization

- Suppose input image has histogram:

$$p_r(w)\,, \quad w = 0, \cdots, 255$$

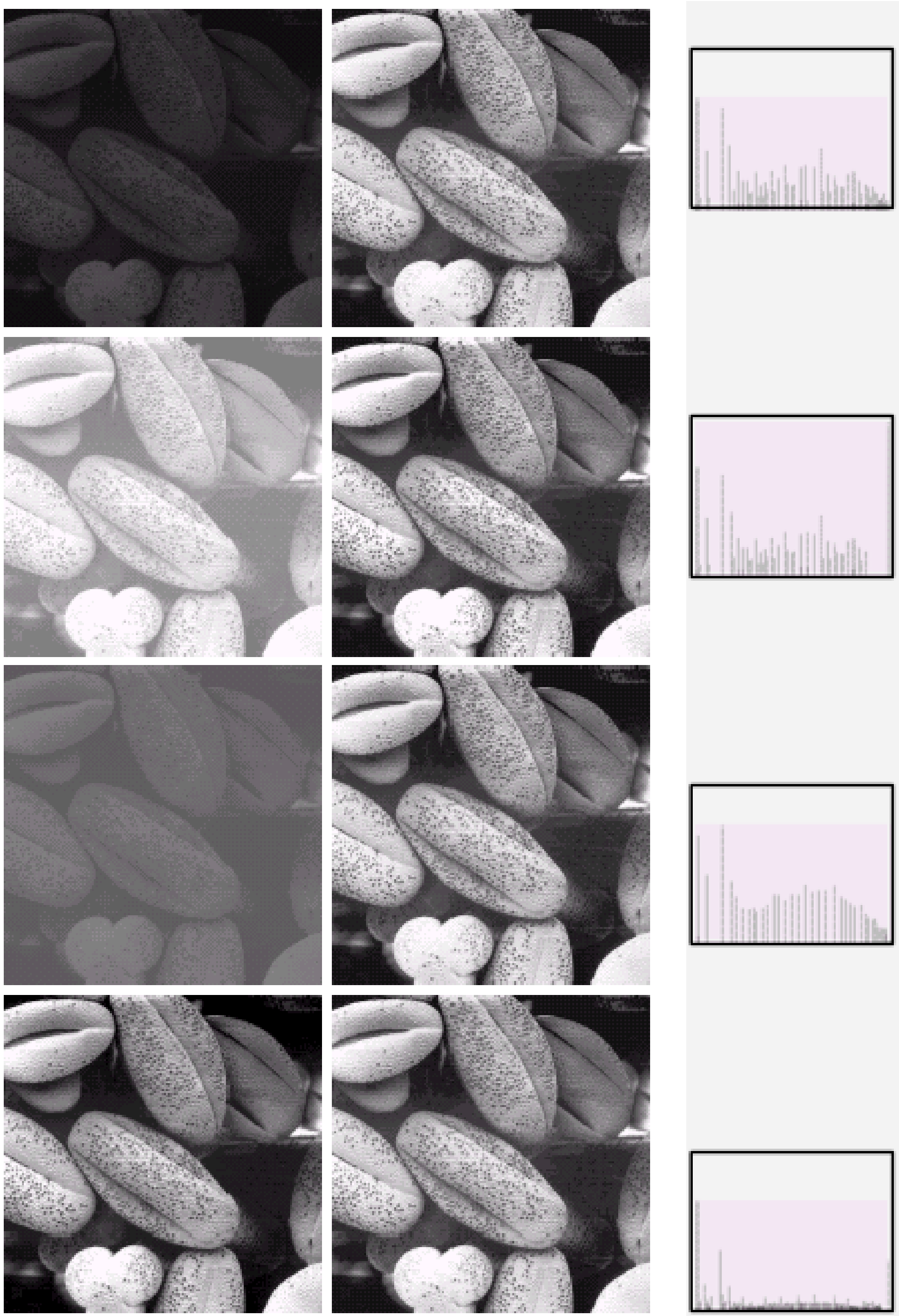
- Use the following curve:

$$s = T(r) = \int_0^r p_r(w)dw$$

- Then it can be shown that output image histogram is flat:

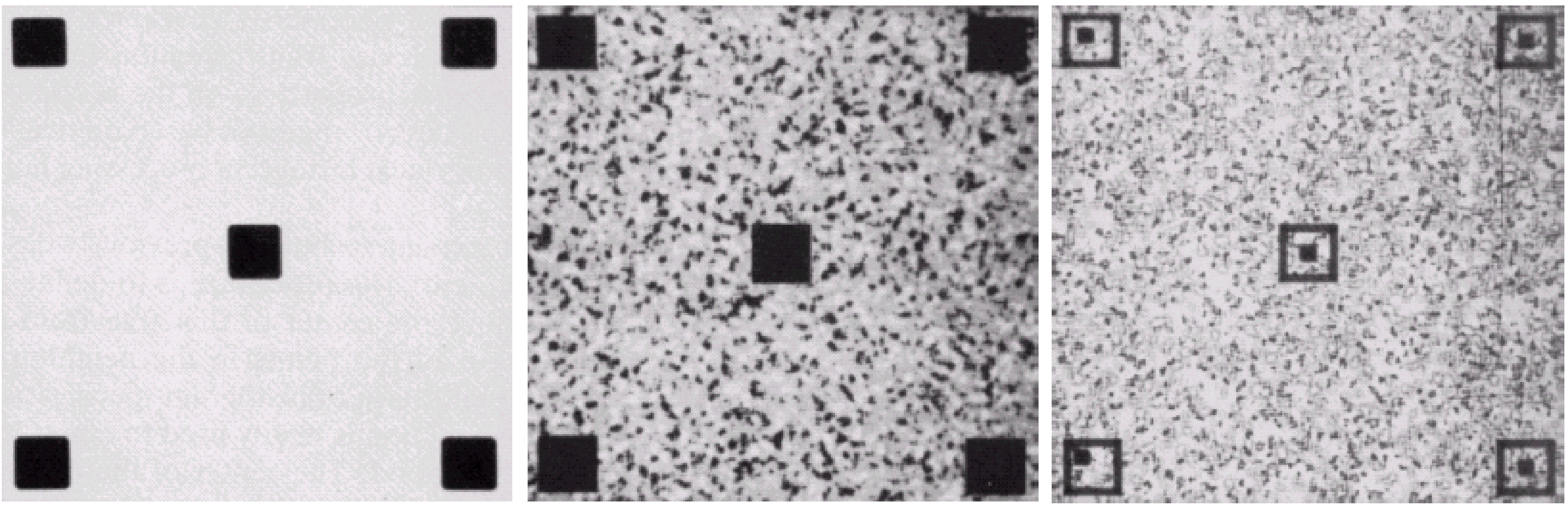
$$p_s(w) = 1, \quad w = 0, \cdots, 255$$

Example



- Histogram equalization results.
- Note that discrete histograms are not completely flat.

Local Enhancement

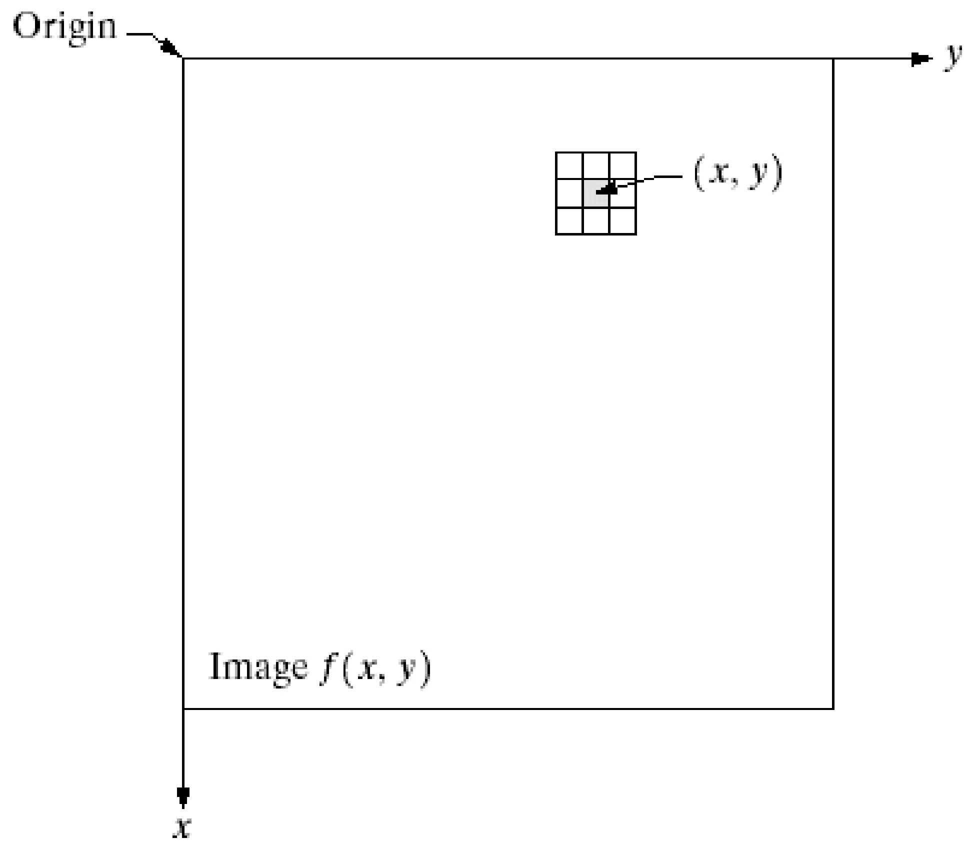


a b c

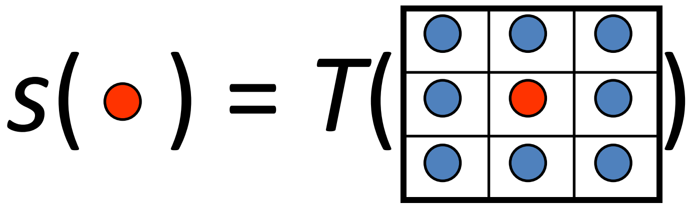
FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel.

Neighborhood Processing

- A 3 x 3 window *w* around (*x,y*)

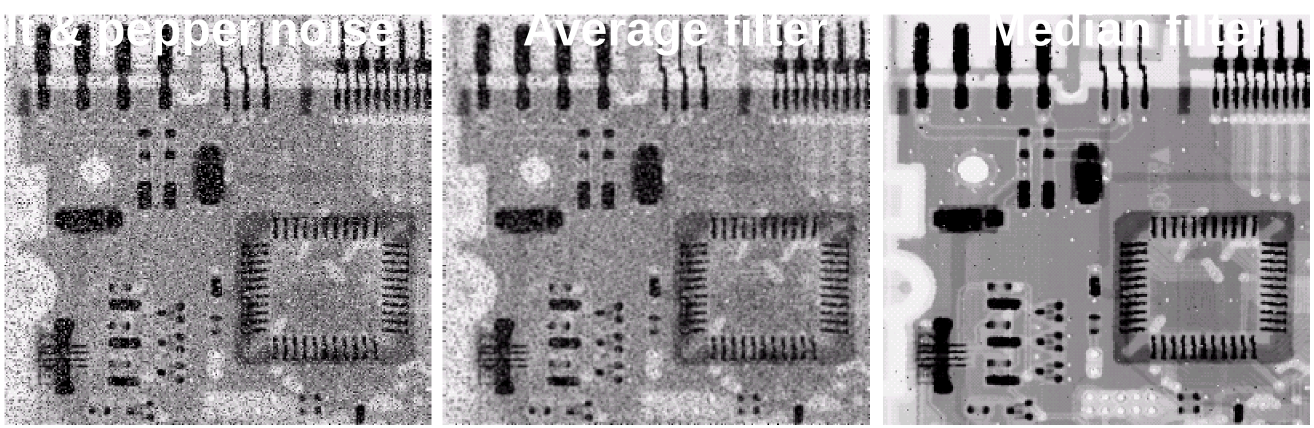


Neighborhood Processing



Examples:

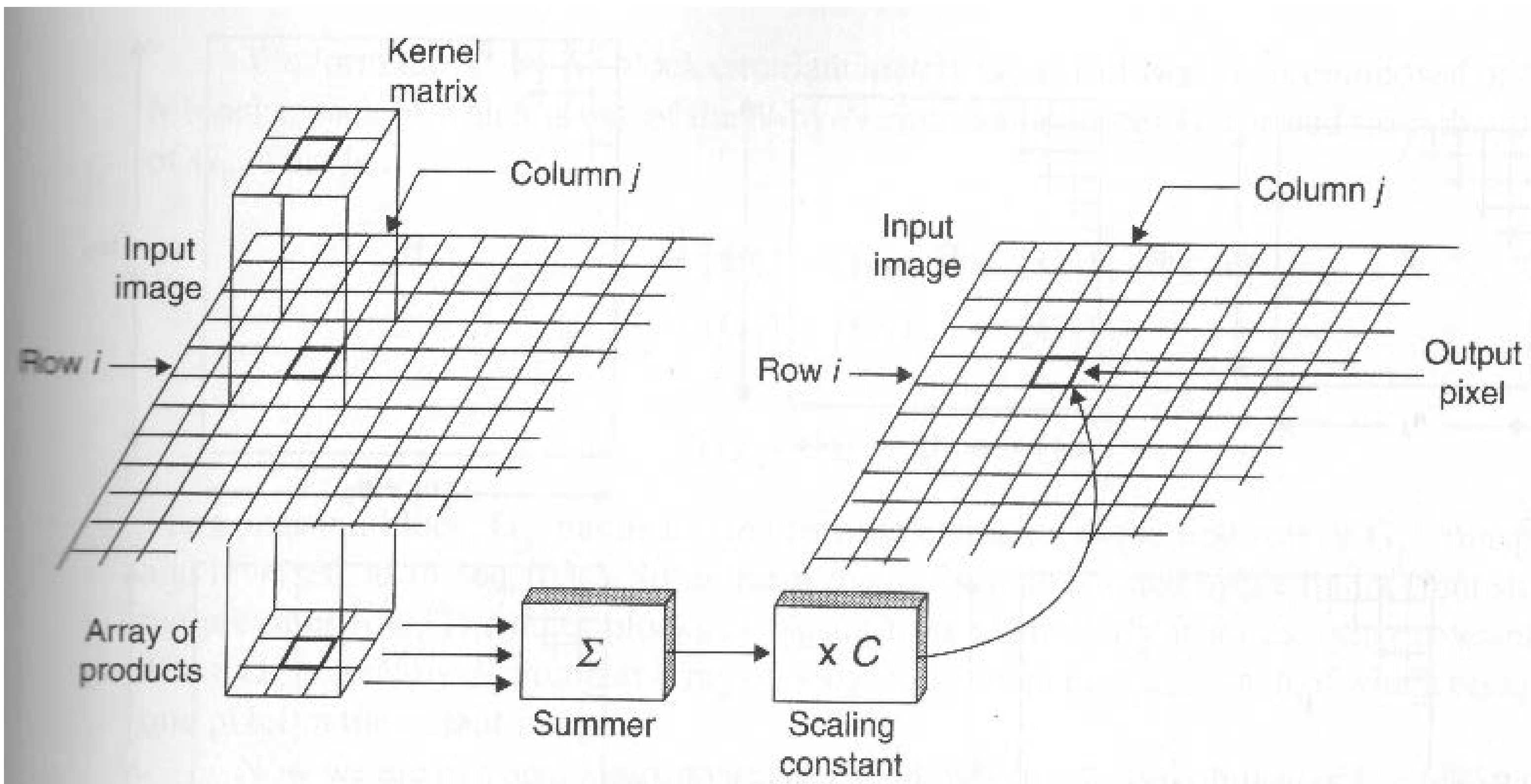
- T( ) = average
- T( ) = median (middle of sort listed list of values)
- T( ) = min, or max



Filtering

- Filter: a.k.a. mask, kernel

$$s(x,y) = C \sum_{(u,v)\in W} w(u,v)\, r(x+u, y+v)$$



Averaging as filtering

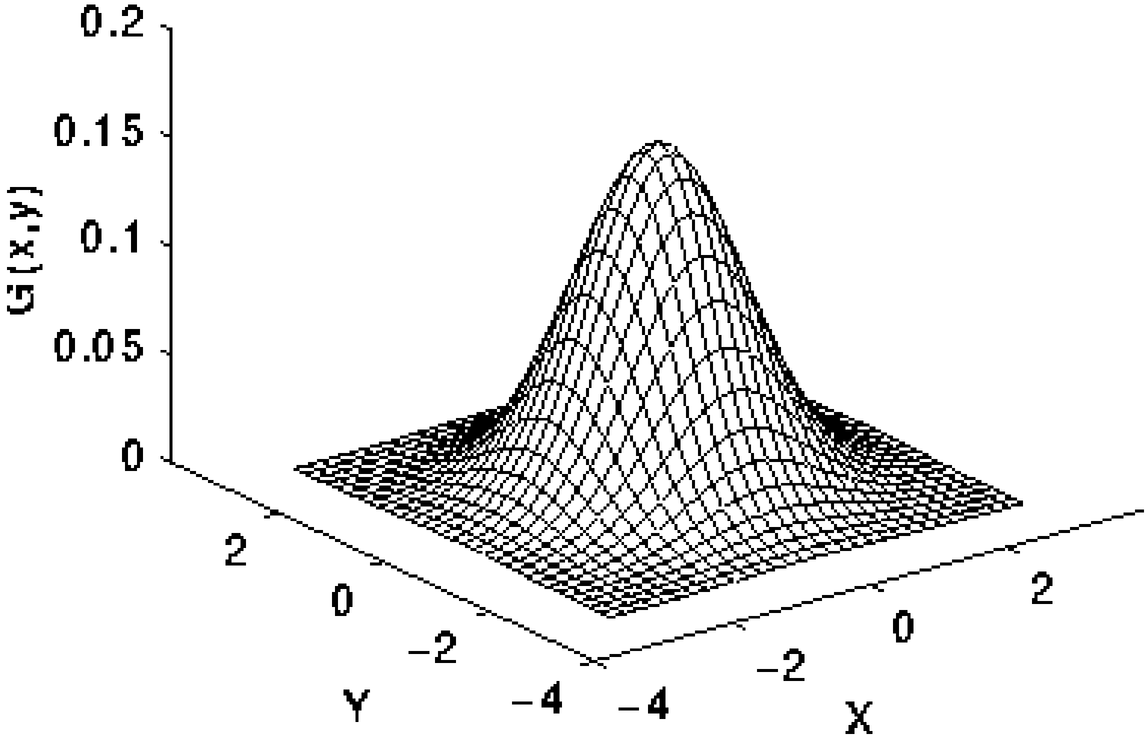
- 3x3 averaging as a mask:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

- 5x5 mask:

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Gaussian Mask



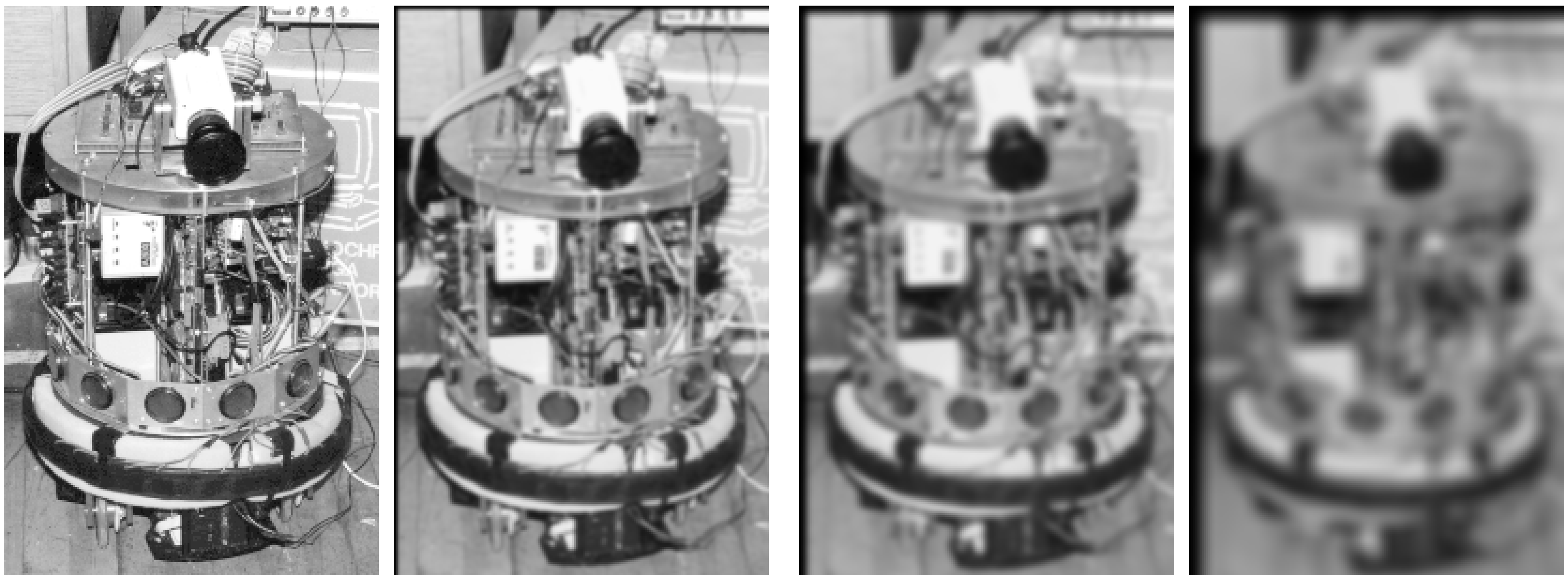
2D Gaussian function with mean = (0,0) and σ =1

Gaussian Mask

- Discrete approximation, $\sigma = 1.4$

$$\frac{1}{115}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Gaussian Blurring



original image

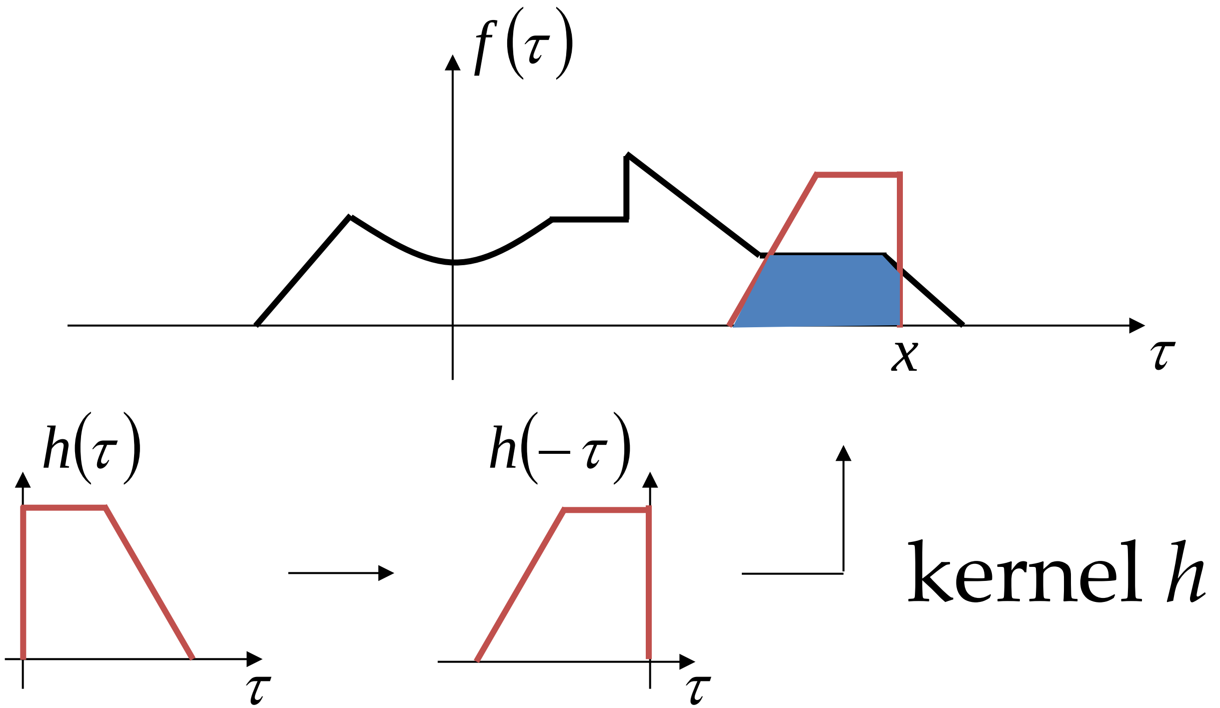
σ = 1.0 (5×5 kernel)

σ = 2.0 (9×9 kernel)

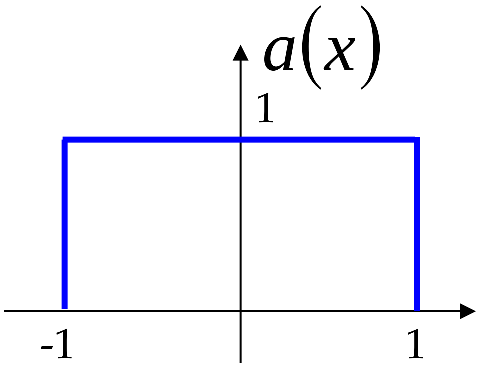
σ = 4.0 (15×15 kernel)

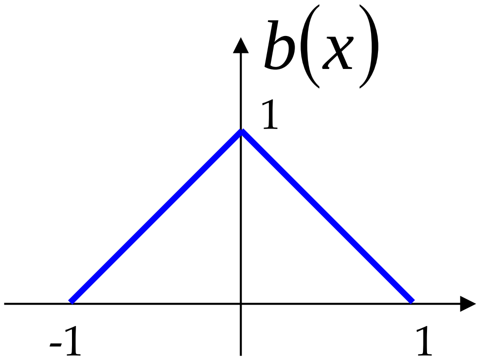
Convolution

$$g(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)d\tau \qquad g = f * h$$

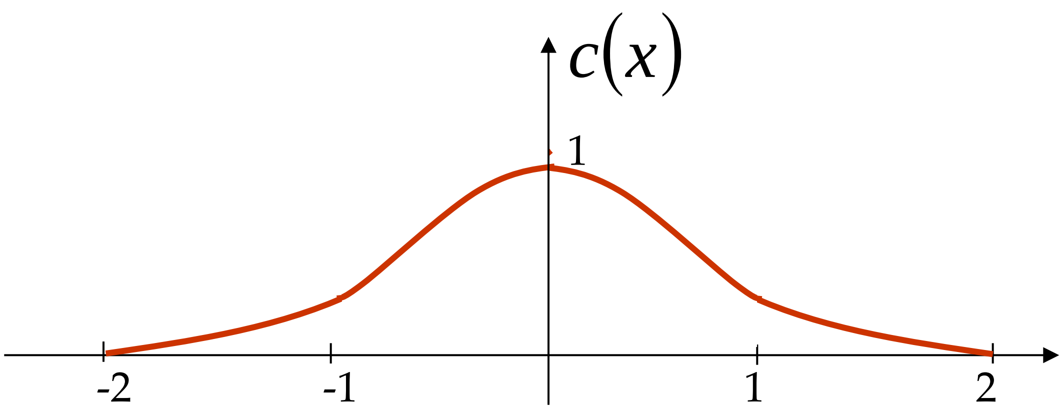


Convolution - Example



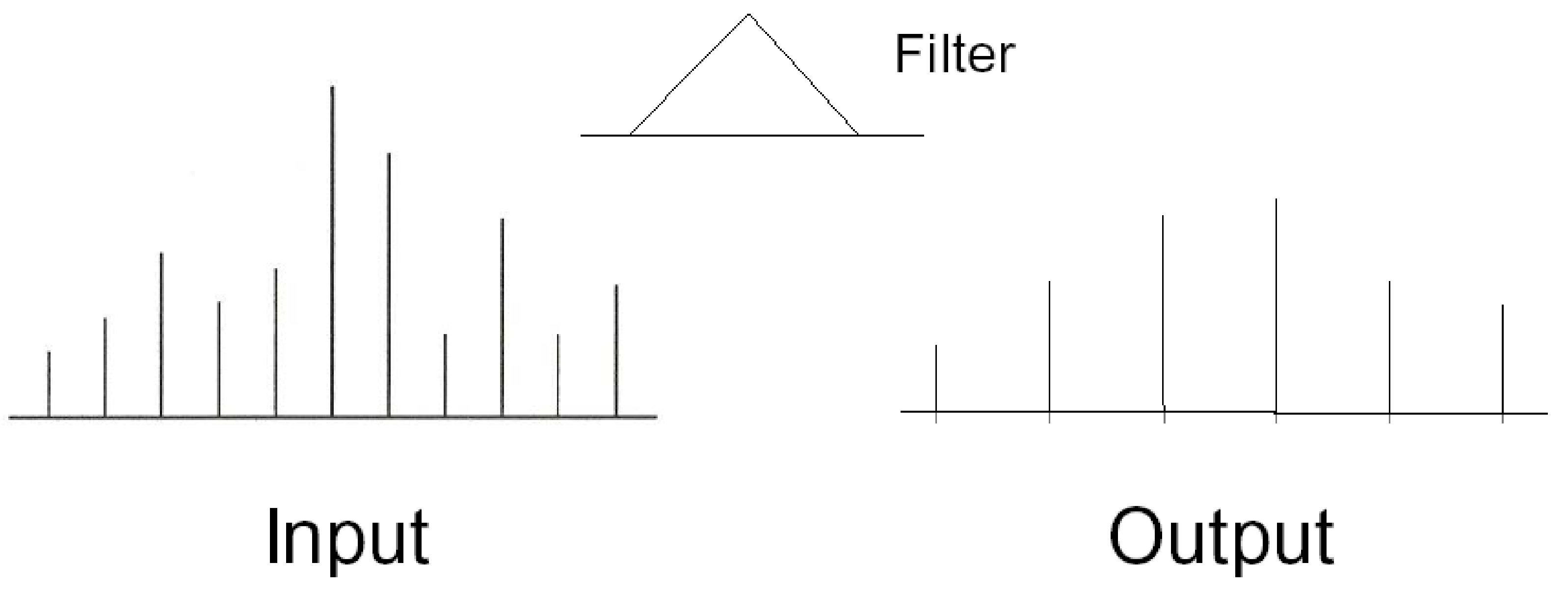


$$c = a * b$$



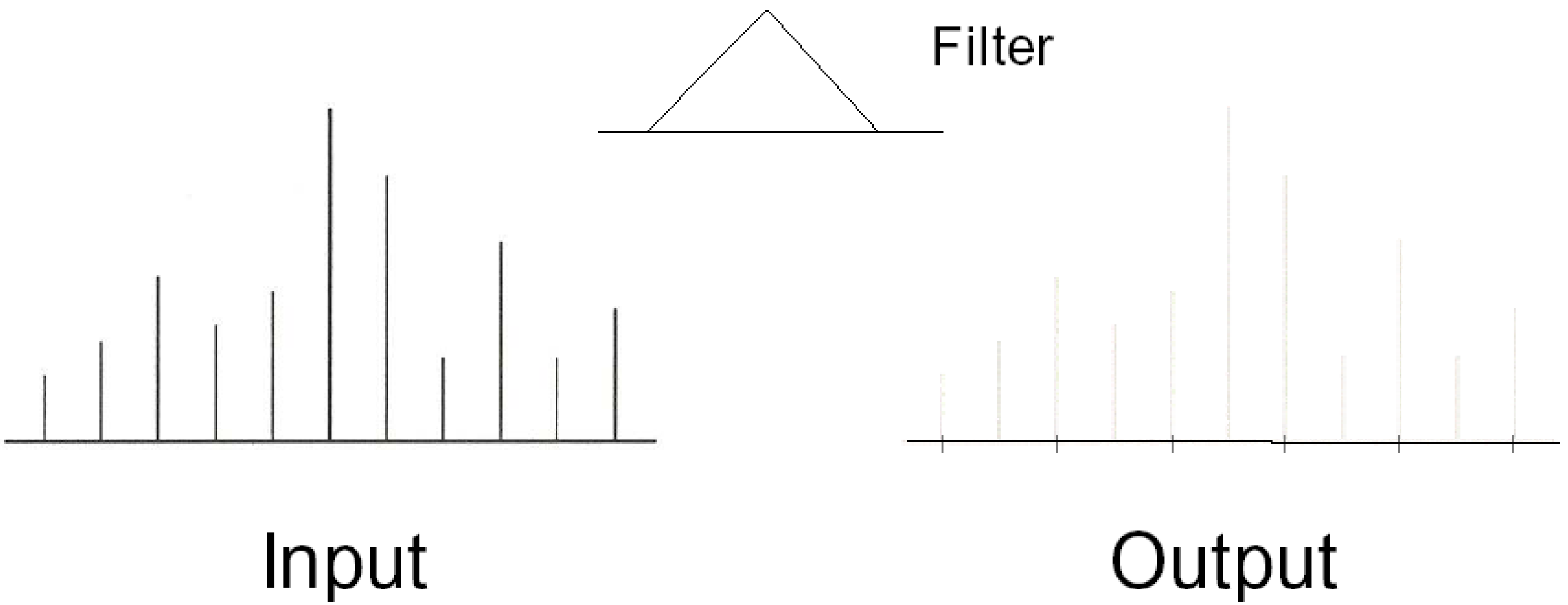
Convolution

- Spatial domain: output pixel is weighted sum of pixels in neighborhood of input image
  - Pattern of weights is the “filter”



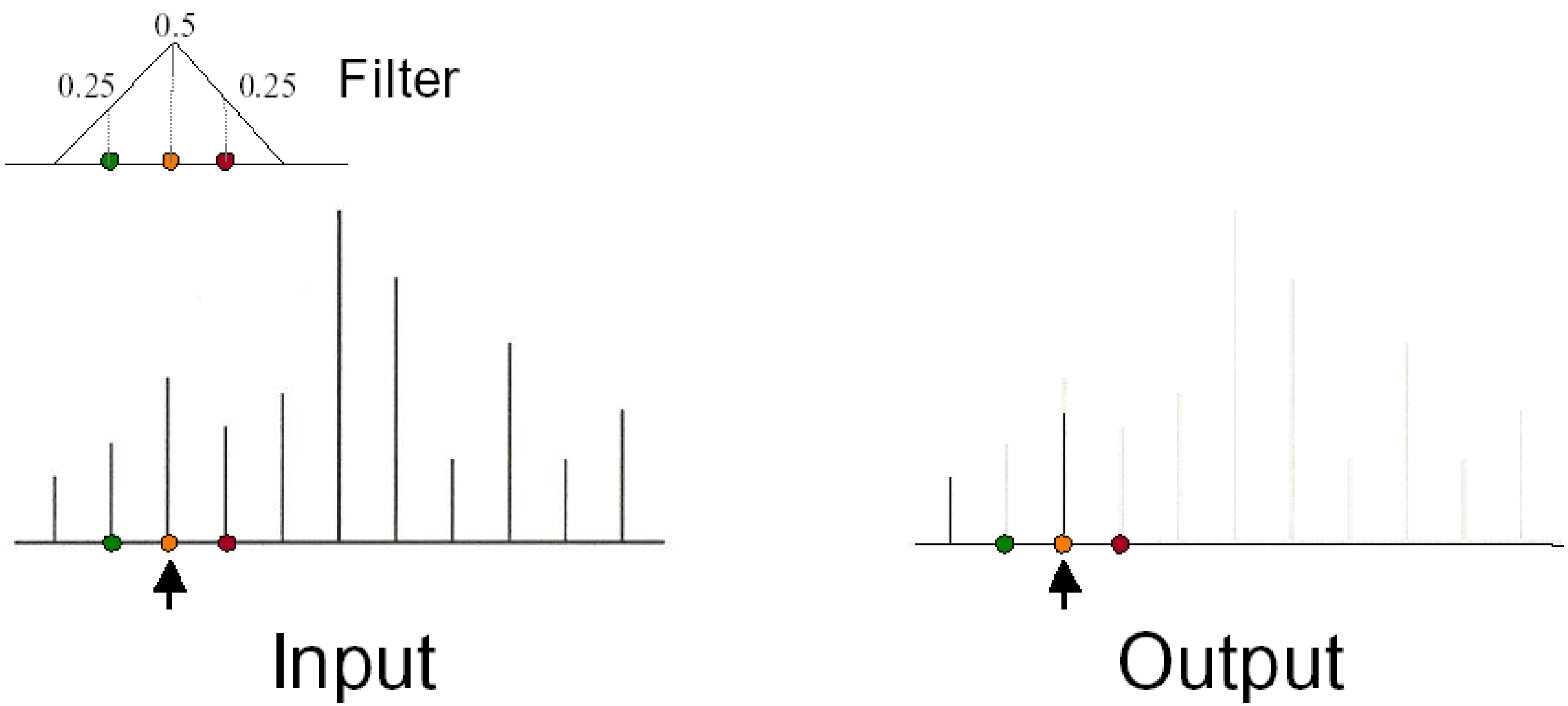
Convolution

- Example 1:



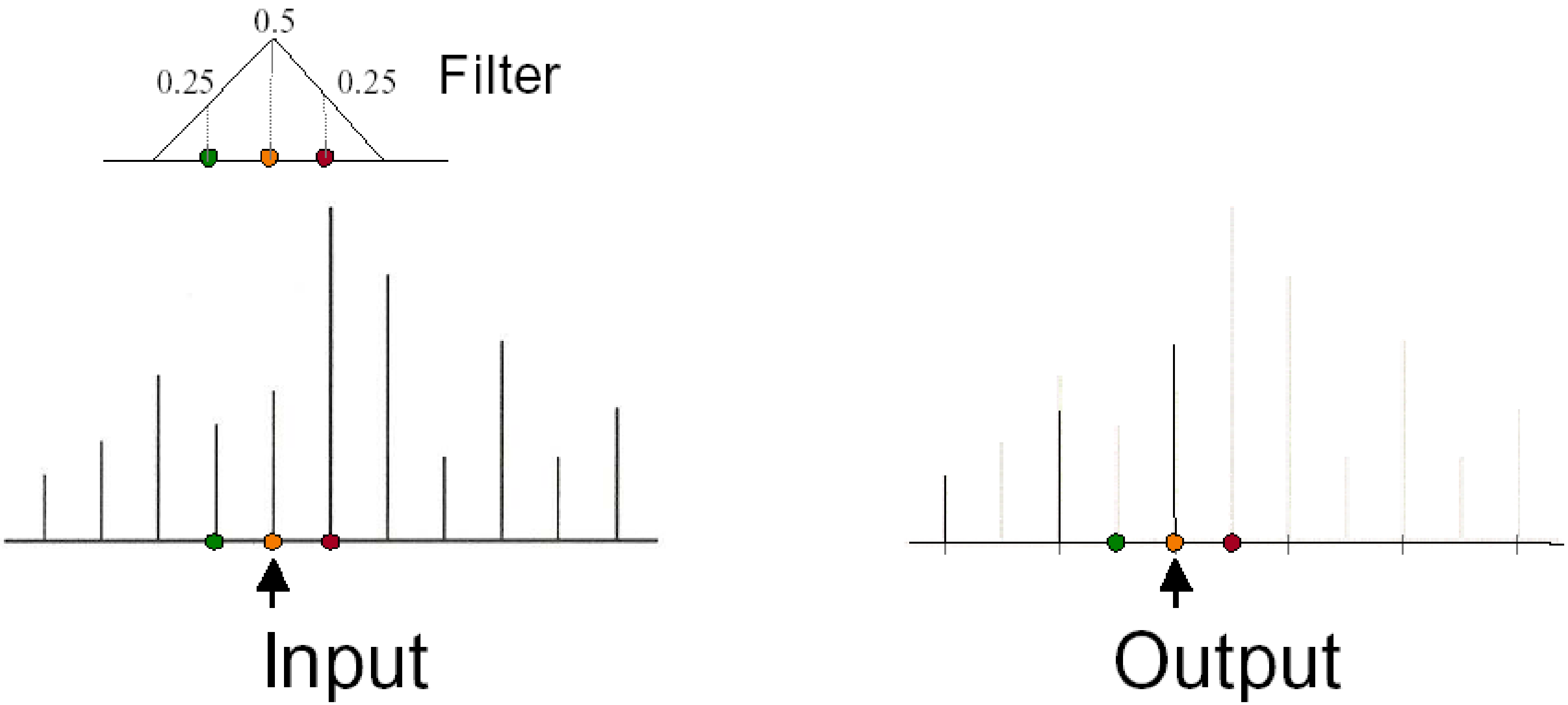
Convolution

- Example 1:



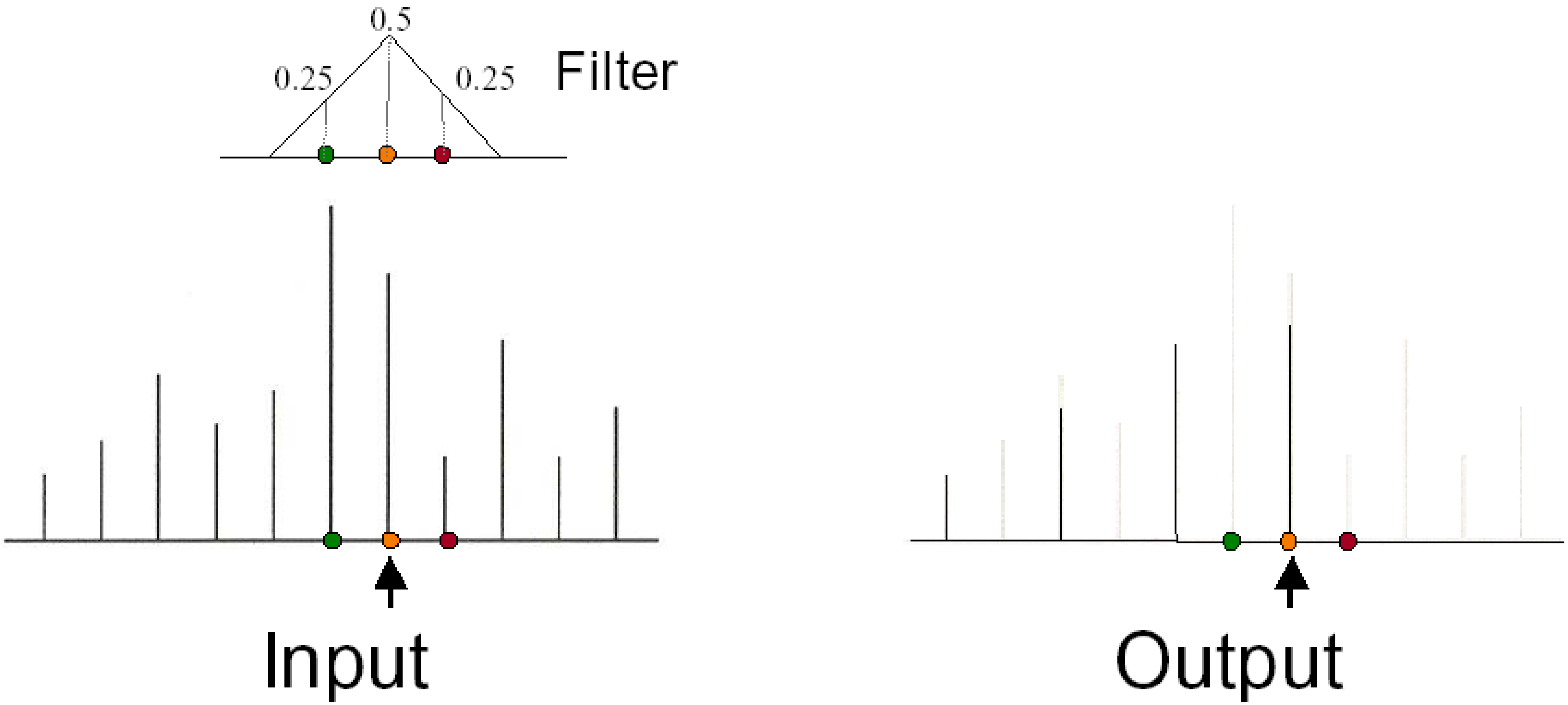
Convolution

- Example 1:



Convolution

- Example 1:



Discrete Convolution

Pixel gets sum of nearby pixels weighted by filter/mask

$$I_{new}(a,b) = \sum_{x=a-width}^{a+width} \sum_{y=b-width}^{b+width} f(x-a, y-b) I_{old}(x,y)$$

2	0	-7
5	4	9
1	-6	-2

Normalization

If you don't want overall brightness change, entries of filter must sum to 1. You may need to normalize by dividing

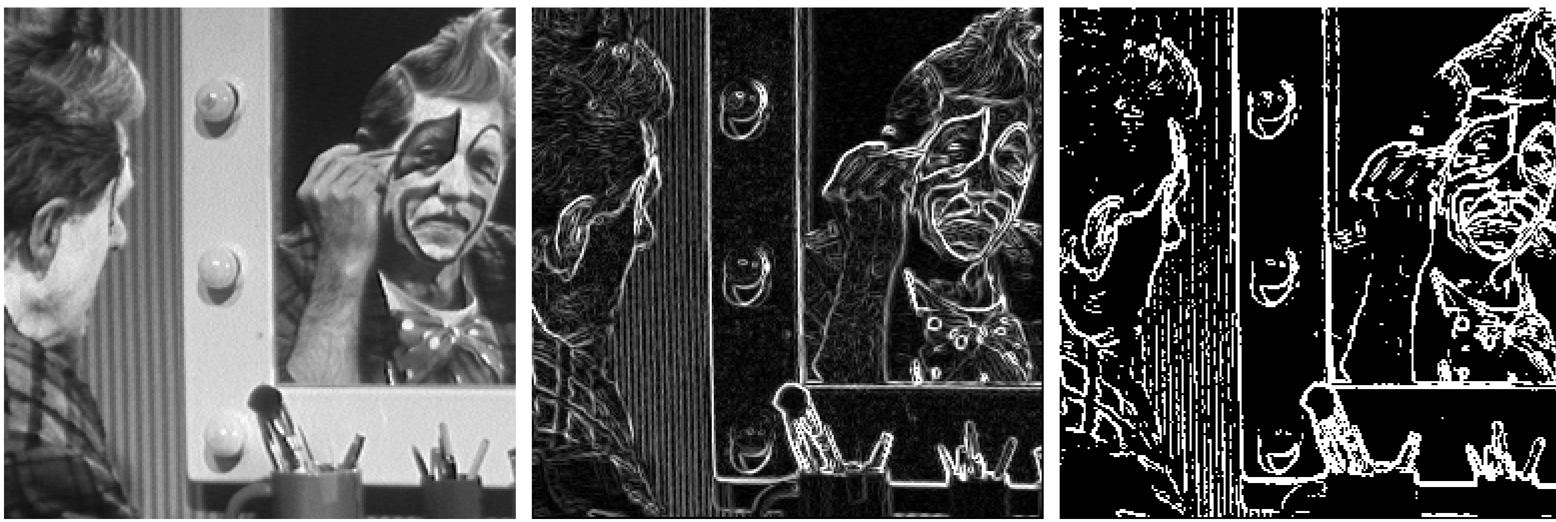
Edge Detection

+1	0
0	-1

Gx

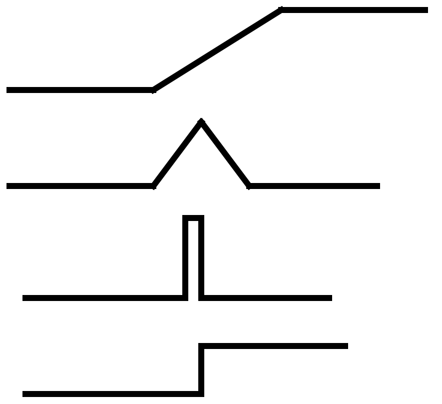
0	+1
-1	0

Gy



Edge Models

- Edges are places in the image with strong intensity contrast.
- Edges often occur at image locations representing object boundaries.
- Types of edges :
    - Ramp (1D profile)
    - Roof (1D profile)
    - Line (1D profile)
    - Step (1D profile)



The Step Edge Model

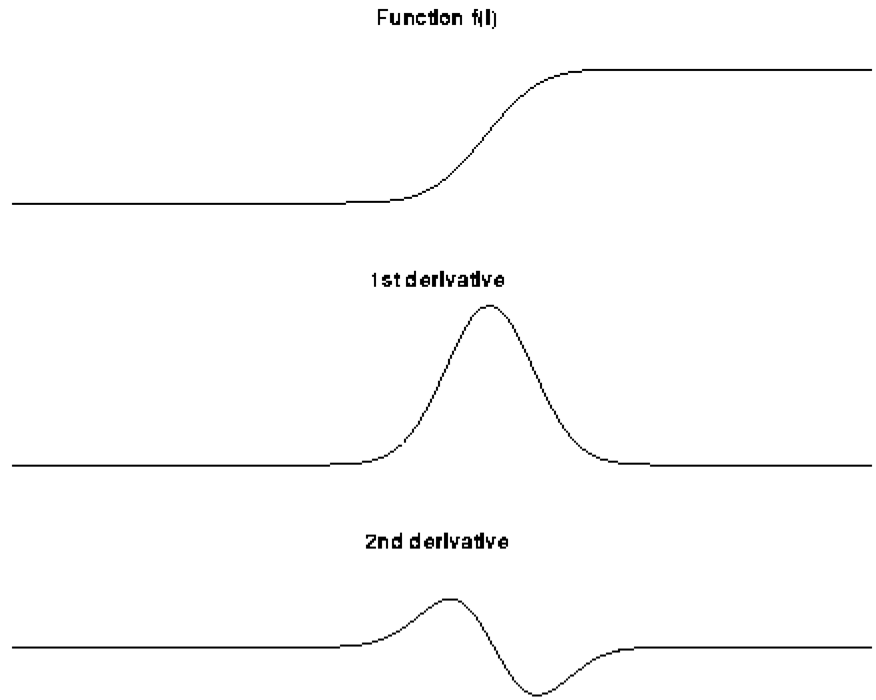
Step edge exist for artificially generated images.

Real images DO NOT have step edges because anti-aliasing filter used in the imaging system.

Real edge →

1st derivative →

2nd derivative →



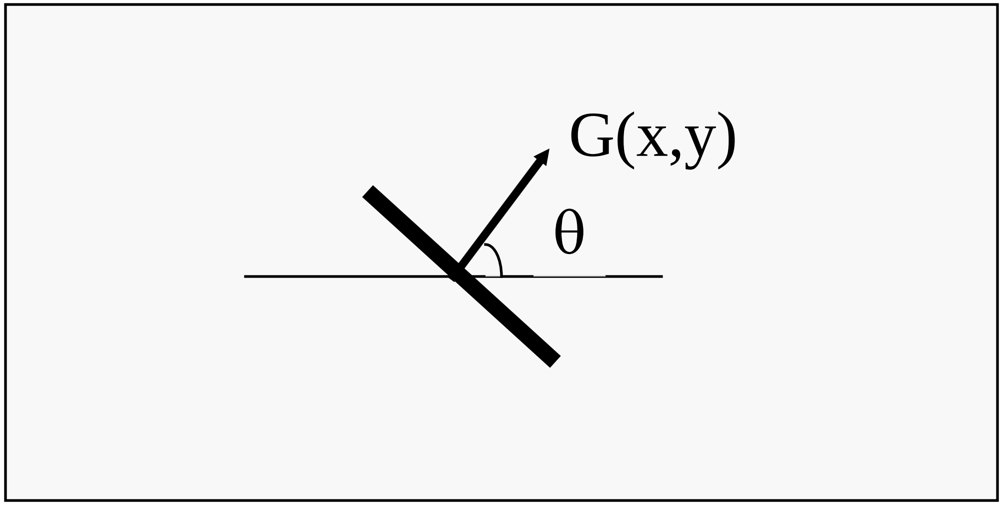
First derivative operator

Also known as Gradient operator.

For an edge in a 2D image,

$$G(x,y) = \frac{\partial F(x,y)}{\partial x}\cos(\theta) + \frac{\partial F(x,y)}{\partial y}\sin(\theta)$$

where G(x,y) is the gradient normal to the edge.



First derivative operator

Usually, we compute

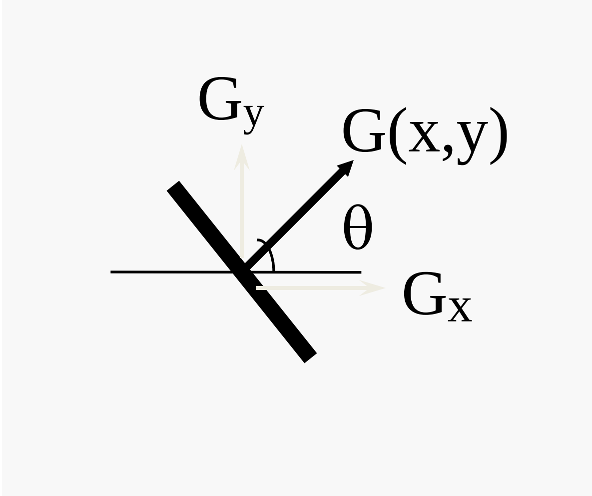
$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

where G_x is the horizontal gradient,

G_y is the vertical gradient.

$$|\nabla f| = \left[G_x^2 + G_y^2\right]^{1/2}$$

$$|\nabla f| \approx |G_x| + |G_y| \quad ; \theta(x, y) = \tan^{-1}\left(G_y / G_x\right)$$



Sobel's gradient operator

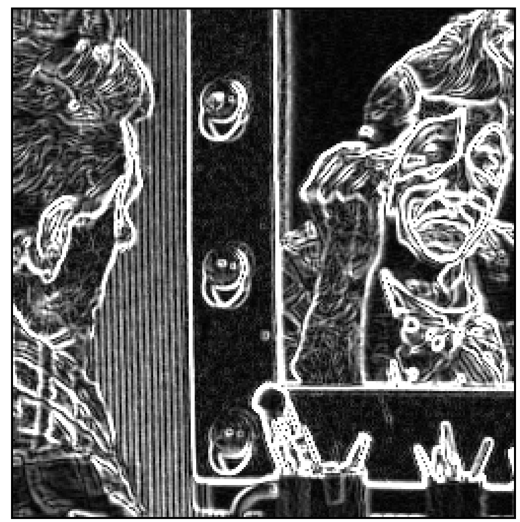
-1	0	+1
-2	0	+2
-1	0	+1

Gx

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

Gy

original



Convolved with Sobel's operator

Second derivative operator

Laplacian operator:

The Laplacian L(x,y) of an image at I(x,y) is:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Laplacian kernels:

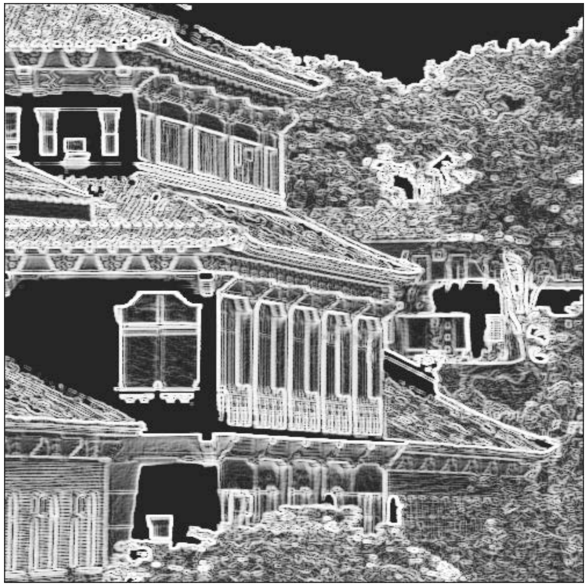
0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

−1	2	−1
2	−4	2
−1	2	−1

Laplacian operator usually produce closed contour original

Sobel output



Laplacian output

Second derivative operator

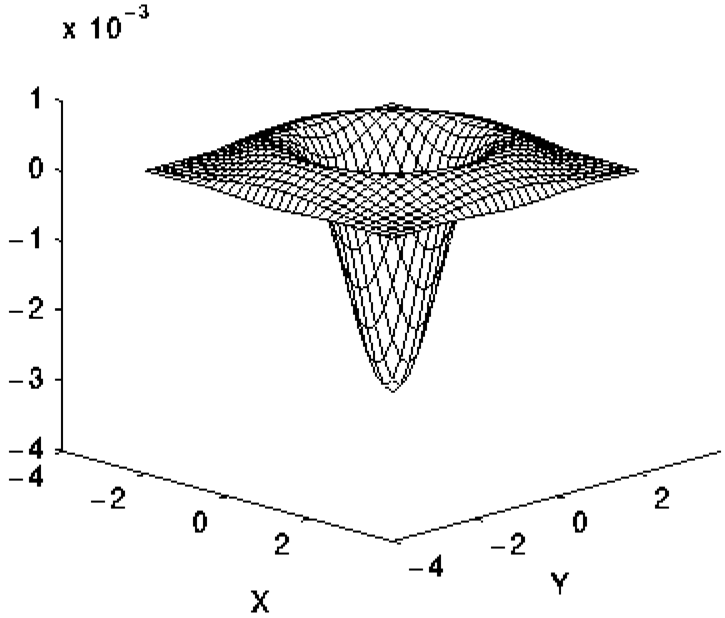
Laplacian of Gaussian (LoG) operator:

This is equivalent to Gaussian smoothing followed by Laplacian.

$$G(x,y) = \nabla^2[g(x,y) * I(x,y)]$$

$$= \nabla^2 g(x,y) * I(x,y)$$

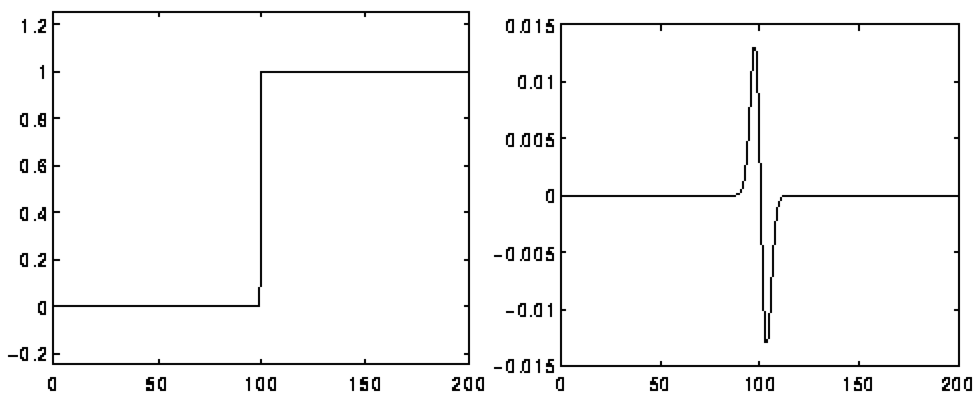
$$\nabla^2 g(x,y) = -\frac{1}{2\pi\sigma^4}\left(2 - \frac{x^2+y^2}{\sigma^2}\right)\exp\left[-\frac{x^2+y^2}{2\sigma^2}\right]$$



2D LoG function

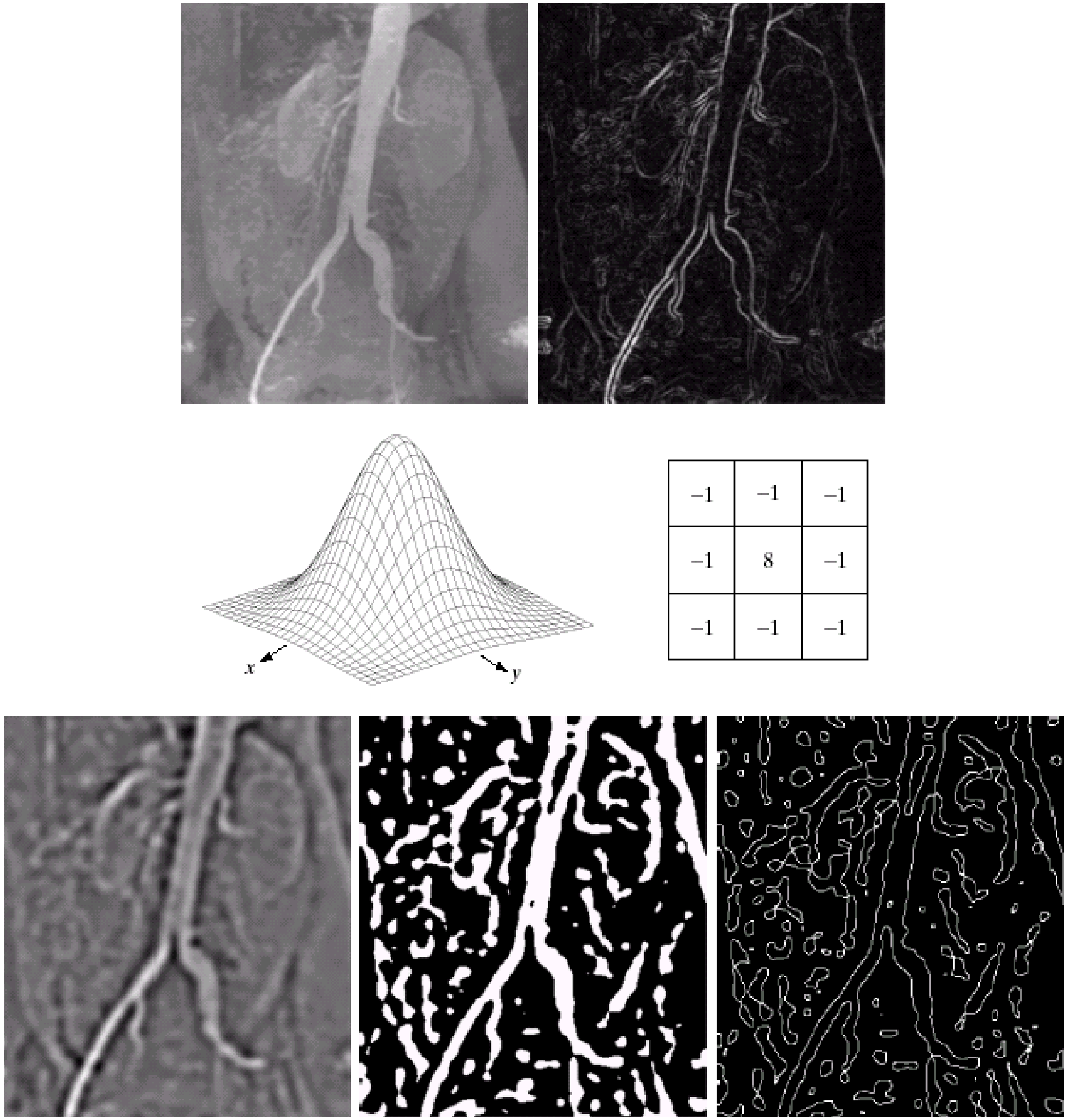
0	0	3	2	2	2	3	0	0
0	2	3	5	5	5	3	2	0
3	3	5	3	0	3	5	3	3
2	5	3	-12	-23	-12	3	5	2
2	5	0	-23	-40	-23	0	5	2
2	5	3	-12	-23	-12	3	5	2
3	3	5	3	0	3	5	3	3
0	2	3	5	5	5	3	2	0
0	0	3	2	2	2	3	0	0

2D LoG kernel



Response of the LoG to a step edge

Edge Detection



–1	–1	–1
–1	8	–1
–1	–1	–1

a b
c d
e f g

FIGURE 10.15 (a) Original image. (b) Sobel gradient (shown for comparison). (c) Spatial Gaussian smoothing function. (d) Laplacian mask. (e) LoG. (f) Thresholded LoG. (g) Zero crossings. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Summary

- Point processing computes an output pixel as a function of the corresponding input pixel.
- Neighborhood processing computes an output pixel as a function of the *neighborhood* around the input pixel.
- One type of neighborhood processing is called filtering, via a mask.
- Edge detection can be done using masks.
- Same technique can be used to detect corners, arcs, lines.