Basic Concepts and Terminology

Applications of Image Processing

Image Processing

- Medical Application
- Industrial Application
- Consumer Electronics
- Military Application
- Security Application

What is an Image?

Discrete representation of data possessing both spatial(layout) and intensity (color) information.

What is a digital image?

Representation of a two-dimensional image using a finite number of points usually called picture elements or *pixels*.

$I(0,0)$
$I(0,N)$
Image pixel location at
(m,n) denoted $I(m,n)$
$I(M,0)$
$I(M,N)$

Image Color

Black/white

Greyscale

Black (min)

⋮

White (max)

RGB : Red, Green, Blue

HSV : Hue, Saturation, Value (Intensity)

Image Resolution and Quantization

Size of 2-D grid and data size stored for each individual image pixel.

Resolution of image source

- Spatial Resolution

  C x R e.g. 640 x 480, 800 x 600

- Temporal Resolution

  e.g. 25fps

- Bit Resolution

  e.g. 24 bit

Image Formats

JPEG	Joint Photographic Experts Group	Lossy compression
GIF	Graphics Interchange Format	Lossless compression Limited to 8bit color
BMP	Bit map picture	Basic format Lossless compression
PNG	Portable network graphics	Lossless compression
TIFF	Tagged Image file format	Very flexible Compressed/ Uncompressed

Image Data Types

- Binary Image
- Intensity or greyscale Image
- RGB or true-color Image
- Floating – Point Image

Scope of Image Processing

Low Level

Primitive operations

e.g. noise reduction

Mid Level

Extraction of attributes

e.g. edges and contours

High Level

Anlysis and interpretation

Example of some image processing operations.

1. Sharpening

original

sharper

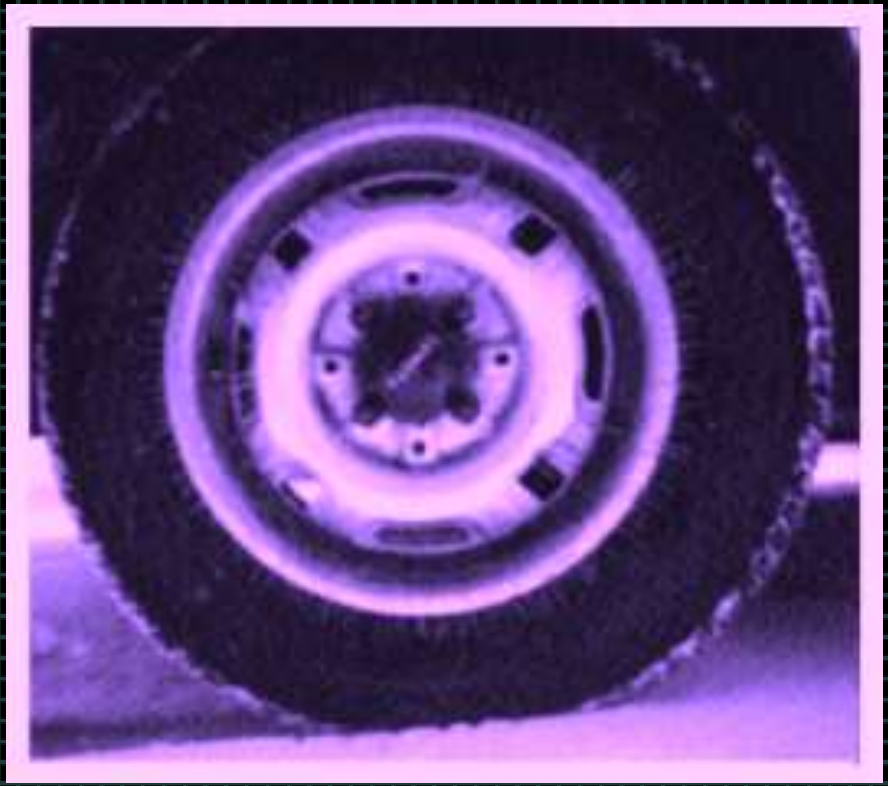
2. Noise Removal

original

noisy

3. De-blurring

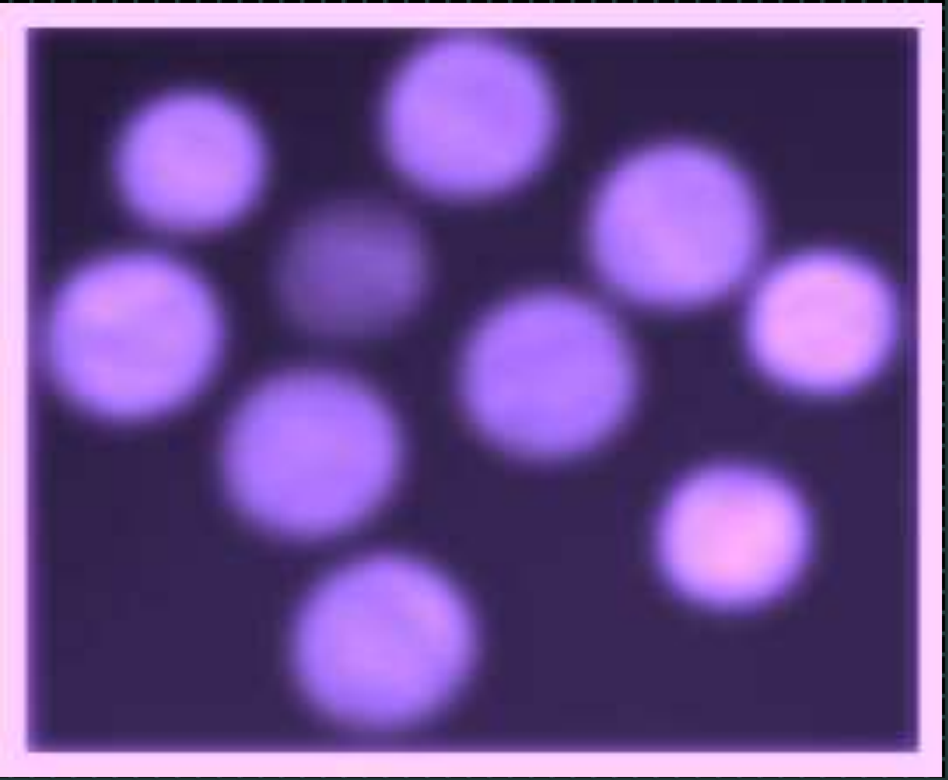
Blur image



De-blurred image

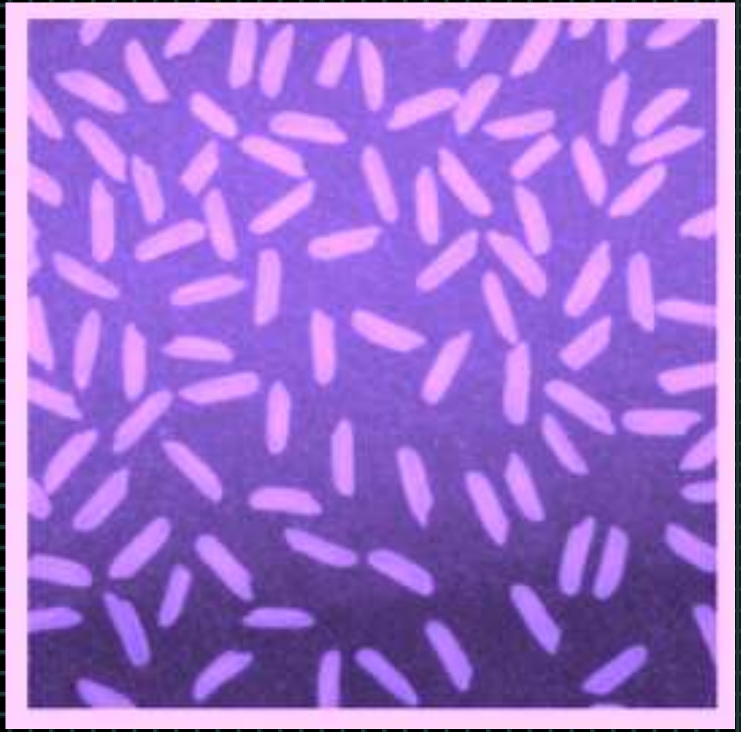
4. Blurring

original image

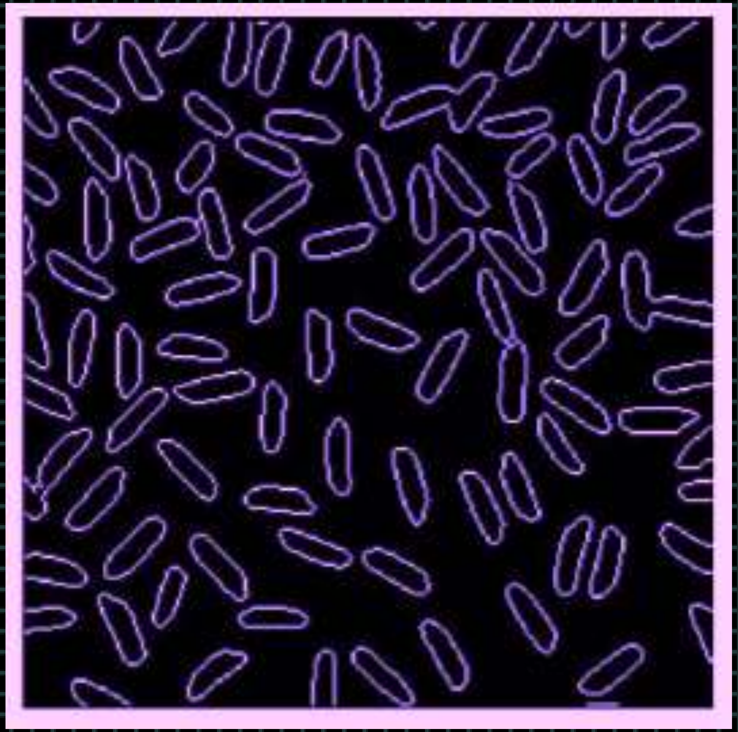


blurred image

4. Edge Extraction



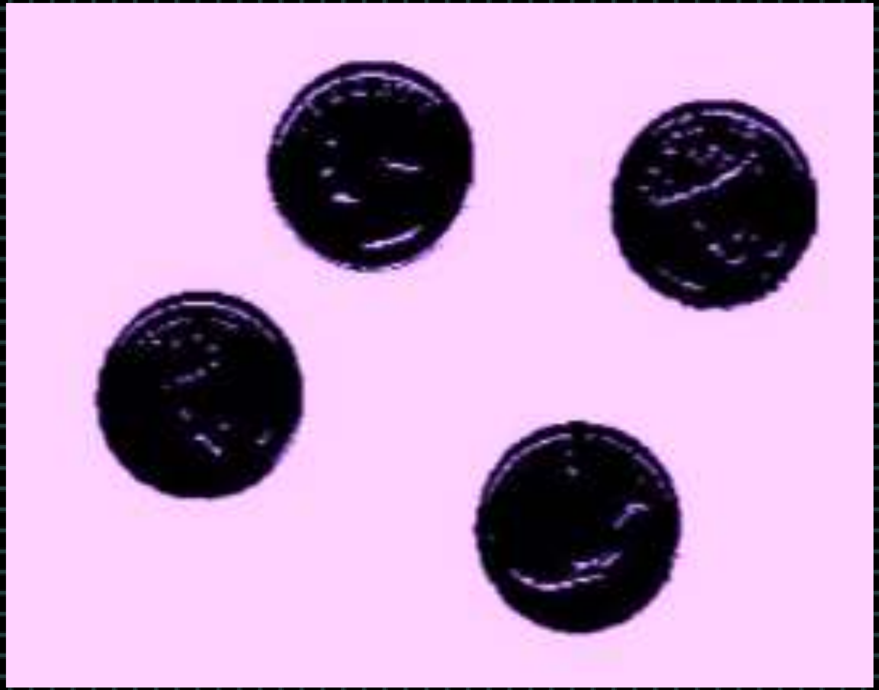
original image



After edge extraction

5. Binarization

original image



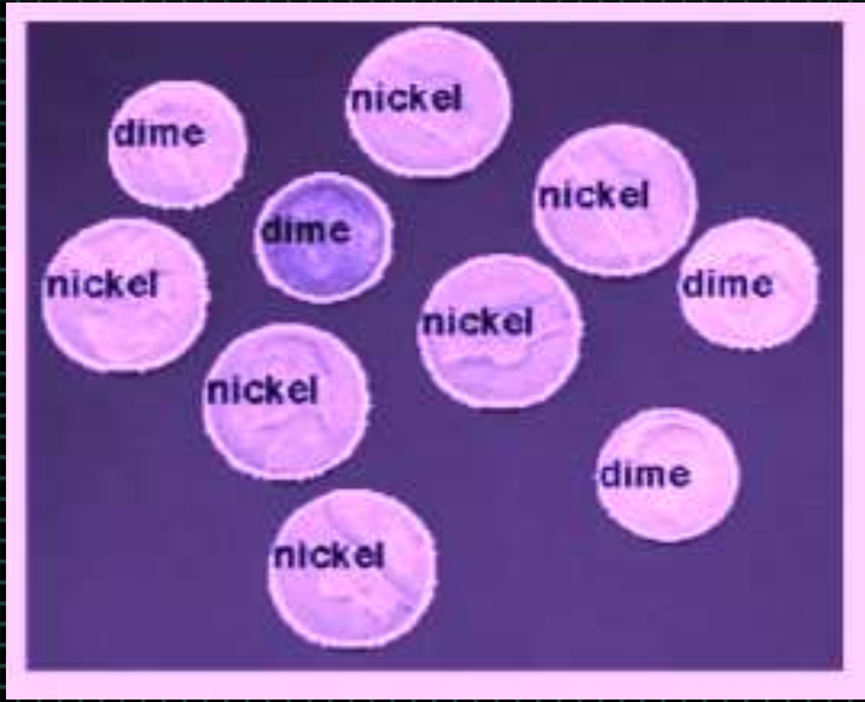
After binarization

7. Contrast Enhancement

original image

After contrast enhancement

6. Object Segmentation and Labeling



Some Basic Terms

Image Topology

Investigation of fundamental image properties using morphological operators

Neighborhood

Pixels surrounding a given pixel

Adjacency

Two pixels p and q are 4-adjacent if they are 4-neighbors of each other and 8-adjacent if they are 8-neighbors of one another.

Paths

A 4-path between two pixels p and q is a sequence of pixels starting with p and ending with q such that each pixel in the sequence is 4-adjacent to its predecessor in the sequence

Components

A set of pixels connected to each other

Connectivity

Existence path between two pixels

Overview of machine vision systems

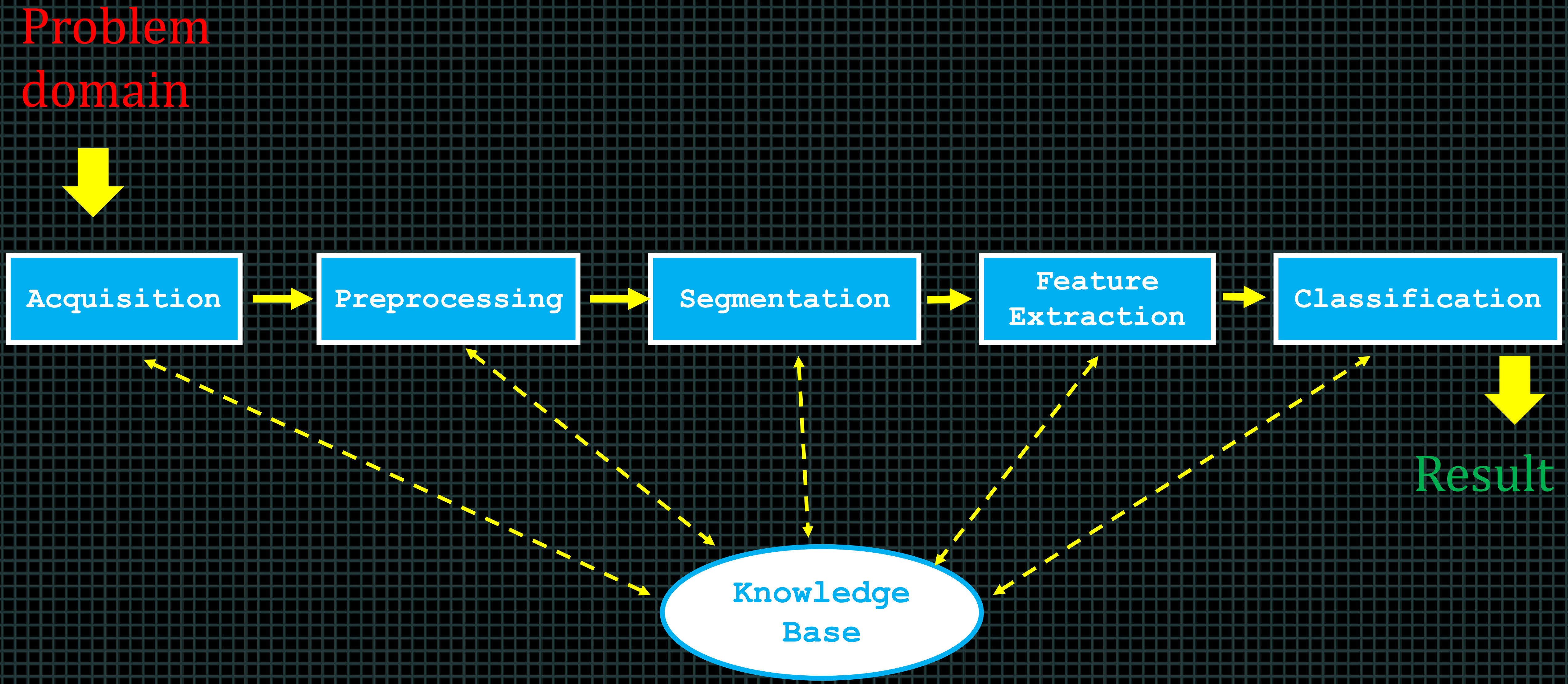

Problem domain
Acquisition
Preprocessing
Segmentation
Feature Extraction
Classification
Knowledge Base
Result

Image Formation

Understanding the formation of an image

$$\text{Image} = \text{PSF} * \text{Object function} + \text{Noise}$$

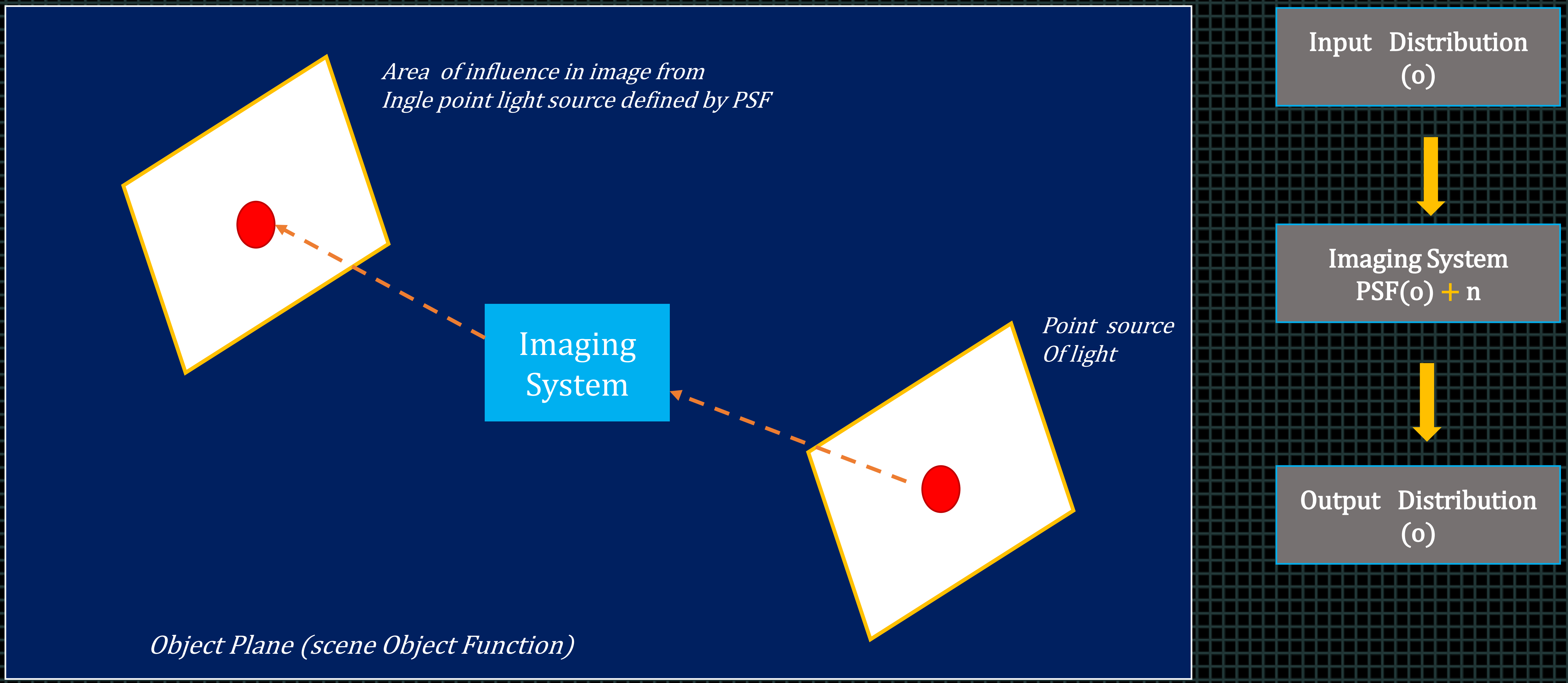
$$S = p * o + n$$

-The way information on the object function is spread .
-Characteristic of imaging device

Convolution

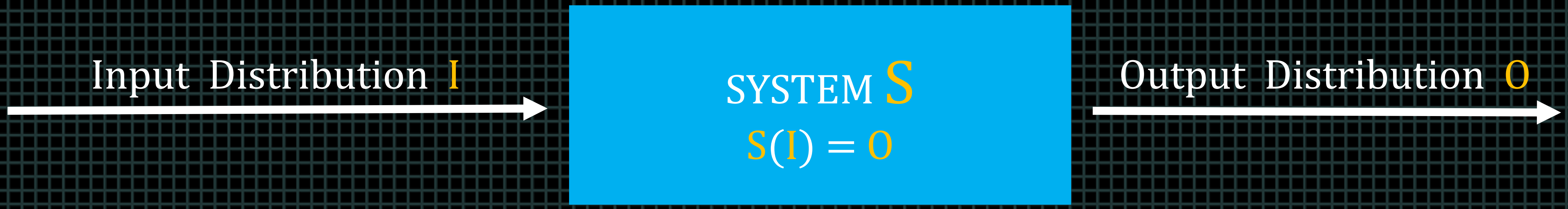
The way light is reflected from object to imaging instrument

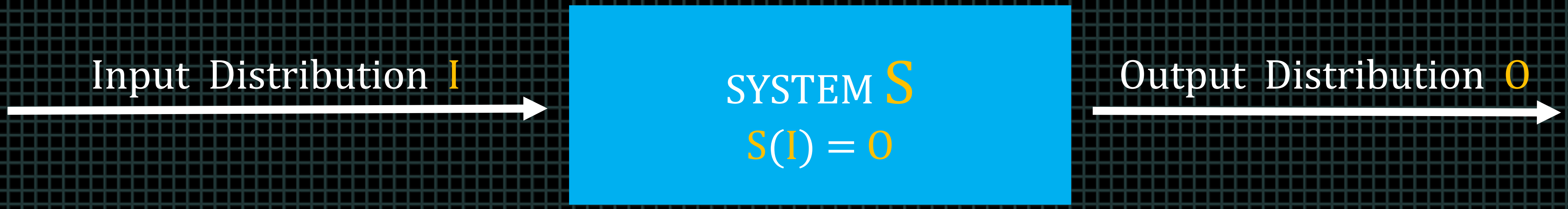
Unwanted external disturbances



Overview of image formation system

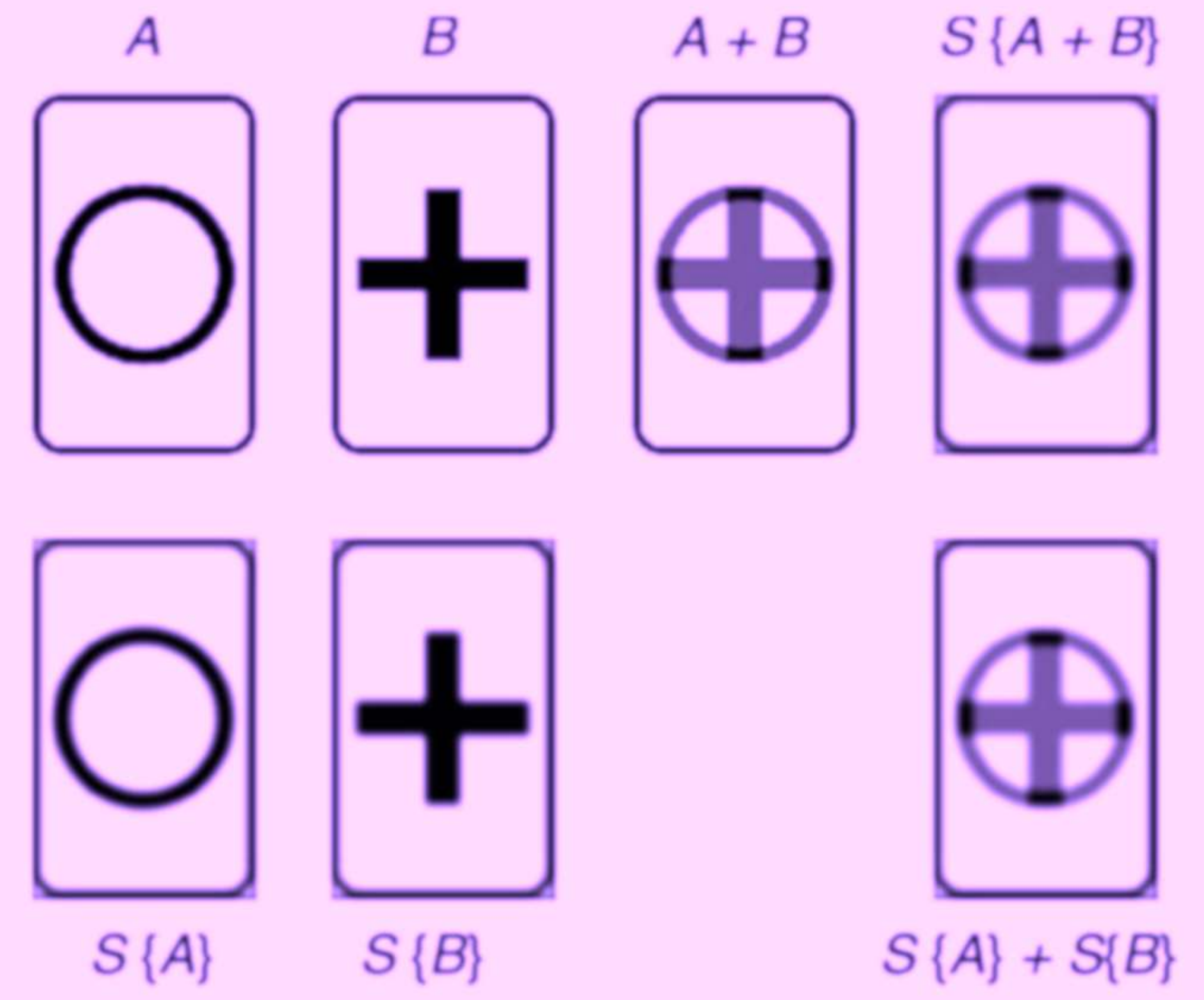
Understanding the mathematics of image formation


Input Distribution I
SYSTEM S
S(I) = O
Output Distribution O


Input Distribution I
SYSTEM S
S(I) = O
Output Distribution O

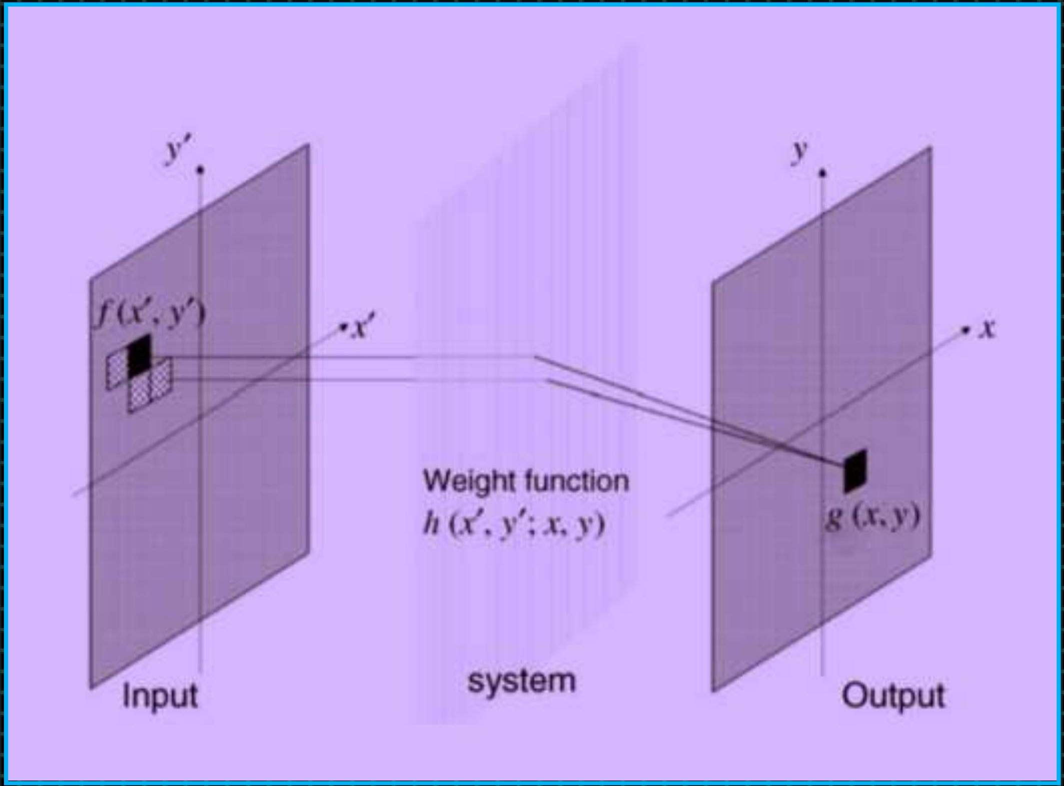
Linear imaging systems

$$S\{aX + bY\} = aS\{X\} + bS\{Y\}$$


A
B
A + B
S {A + B}
S {A}
S {B}
S {A} + S{B}

Linear superposition integral

$$g(x, y) = \iint f(x', y')h(x, y; x', y')dx'dy'$$



The Dirac delta or impulse function

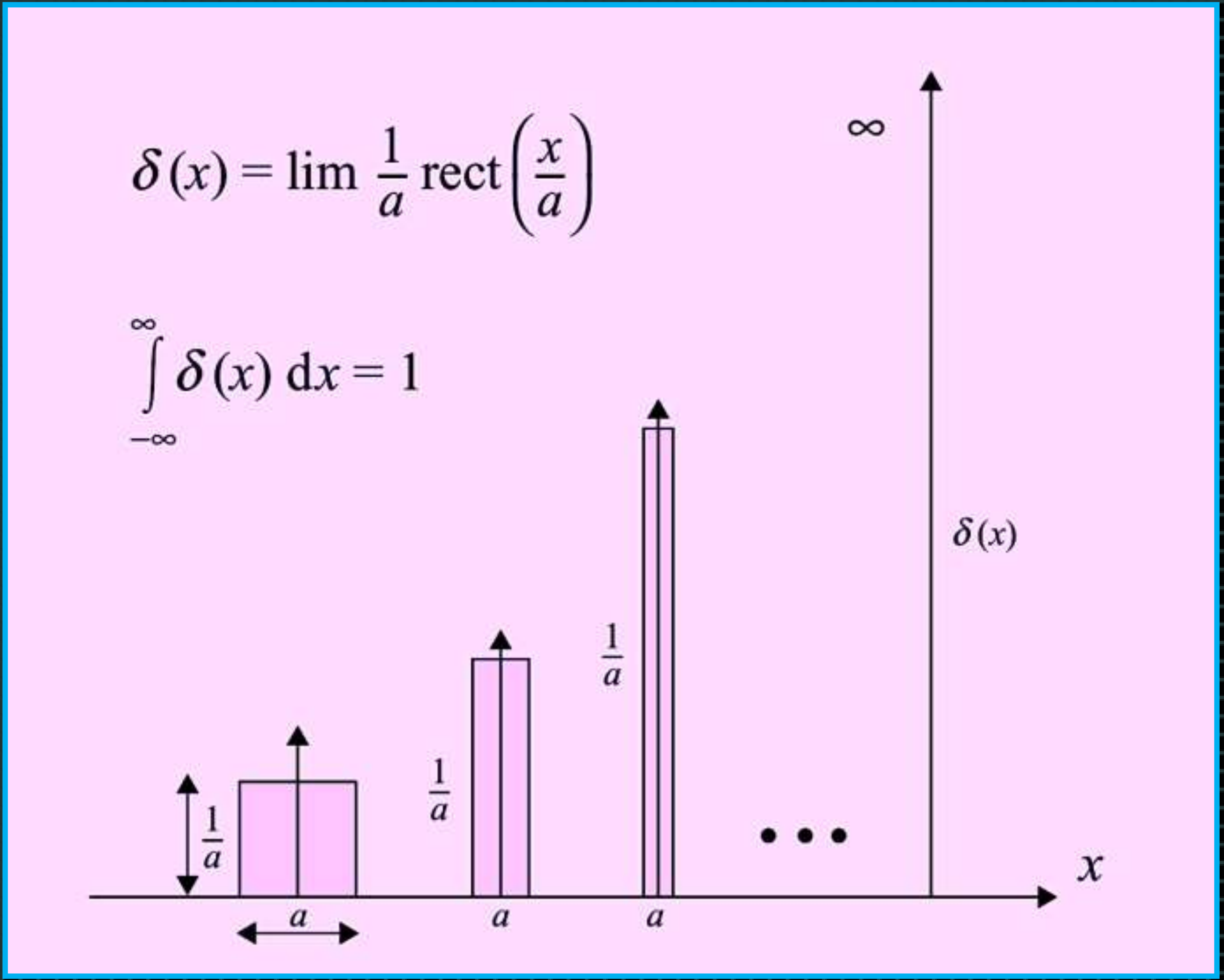
- Represents a bright intensity source
- Occupies infinitesimal region in space

1-D rectangle function :

$$rect\left(\frac{x}{a}\right) = 1 \ |x| < a/2$$
$$= 0 \ \text{otherwise}$$

$$\delta(x) = \lim_{a\to 0} \frac{1}{a} rect\left(\frac{x}{a}\right) \qquad \text{1-D}$$

$$\delta(x, y) = \lim_{a\to 0} \frac{1}{a} rect\left(\frac{x}{a}\right) rect\left(\frac{y}{a}\right) \qquad \text{2-D}$$



$$\delta(x) = \infty \qquad x = 0$$
$$\quad = 0 \qquad x \neq 0$$

$$\int_{\infty}^{\infty} \delta(x)\, dx = 1$$

$$\delta(x - x_0) = \infty \qquad x = x_0$$
$$\quad = 0 \qquad x \neq x_0$$

$$\delta(x, y) = \infty \qquad x = 0, y = 0$$
$$\quad = 0 \qquad otherwise$$

$$\iint_{\infty}^{\infty} \delta(x, y) dxdy = 1$$

$$\int_{\infty}^{\infty} f(x)\delta(x - x_0)\, dx = f(x_0) \qquad \text{1-D}$$

$$\iint_{\infty}^{\infty} f(x, y)\delta(x - x_0, y - y_0)dxdy = f(x_0, y_0) \qquad \text{2-D}$$

The Sifting theorem

- ***Singularity***

$$\delta(x) = \infty \qquad x = 0$$
$$= 0 \qquad x \neq 0$$

- ***Unit area***

$$\int_{\infty}^{\infty} \delta(x)\, dx = 1$$

- ***Shift property***

$$\int_{\infty}^{\infty} \boldsymbol{f(x)\delta(x - x_0)\, dx = f(x_0)}$$

Properties of the delta function

The Point – Spread Function (PSF)

The response of a system to an input distribution consisting of a very small intensity point.

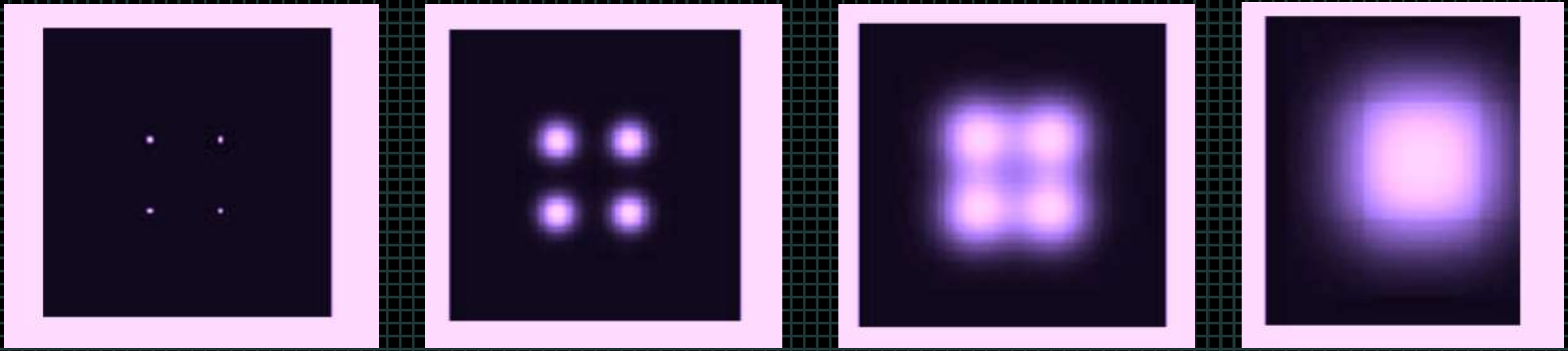
(Dirac delta function)

Input function : $f(x', y') = \delta(x' - x_0, y' - y_0)$

(Linear superposition integral)

$$g(x, y) = \iint \delta(x' - x_0, y' - y_0) h(x, y; x', y') dx' dy'$$

$$g(x, y) = h(x, y; x_0, y_0)$$



Effects of system PSF

Arithmetic and Logical Operations

Arithmetic Operations

$$X \; opn \; Y = Z$$

$opn : (+, -, X, /)$

X : 2D array

Y : another 2D array

First image

Second image

Result

Adding to images

Normalization

$$g = \frac{L_{max}}{f_{max} - fmin}(f - fmin)$$

f : current pixel in w

L_{max}: maximum possible intensity

g : corresponding pixel in Z

f_{max}: maximum pixel in W

W: intermediate result variable

Addition

$$\begin{array}{ccccc} X & & Y & & W \\ \begin{bmatrix} 200 & 100 & 100 \\ 0 & 10 & 50 \\ 50 & 250 & 120 \end{bmatrix} & + & \begin{bmatrix} 100 & 220 & 230 \\ 45 & 95 & 120 \\ 205 & 100 & 0 \end{bmatrix} & = & \begin{bmatrix} 300 & 320 & 330 \\ 45 & 105 & 170 \\ 255 & 350 & 120 \end{bmatrix} \end{array}$$

Normalizing [45,350] *range to [0,255]*

$$Z_a = \begin{bmatrix} 213 & 230 & 238 \\ 0 & 50 & 105 \\ 175 & 255 & 63 \end{bmatrix}$$

$$\boxed{g = \frac{L_{max}}{f_{max} - fmin}(f - fmin)}$$

Truncating all values above 255 *in W*

$$Z_b = \begin{bmatrix} 255 & 255 & 255 \\ 45 & 105 & 170 \\ 255 & 255 & 120 \end{bmatrix}$$

Subtraction

$$\begin{array}{c}X\\ \begin{bmatrix}200 & 100 & 100\\ 0 & 10 & 50\\ 50 & 250 & 120\end{bmatrix}\end{array} \quad \begin{array}{c}Y\\ \begin{bmatrix}100 & 220 & 230\\ 45 & 95 & 120\\ 205 & 100 & 0\end{bmatrix}\end{array} \quad = \quad \begin{array}{c}Z_a\\ \begin{bmatrix}100 & 0 & 0\\ 0 & 0 & 0\\ 0 & 150 & 120\end{bmatrix}\end{array}$$

With truncation

$$= X - Y$$

$$Z_b = \begin{bmatrix}0 & 120 & 130\\ 45 & 85 & 70\\ 155 & 0 & 0\end{bmatrix}$$

With truncation

$$= Y - X$$

$$Z_c = \begin{bmatrix}100 & 120 & 130\\ 45 & 85 & 70\\ 155 & 150 & 120\end{bmatrix} = |Y - X|$$

Multiplication

- Used to perform brightness adjustment
- Makes each pixel brighter or darker by multiplying its original value by a scalar factor
- Multiplication produces better subjective results than addition.

Logic Operations

- AND
- OR
- XOR
- NOT

__Conventions__

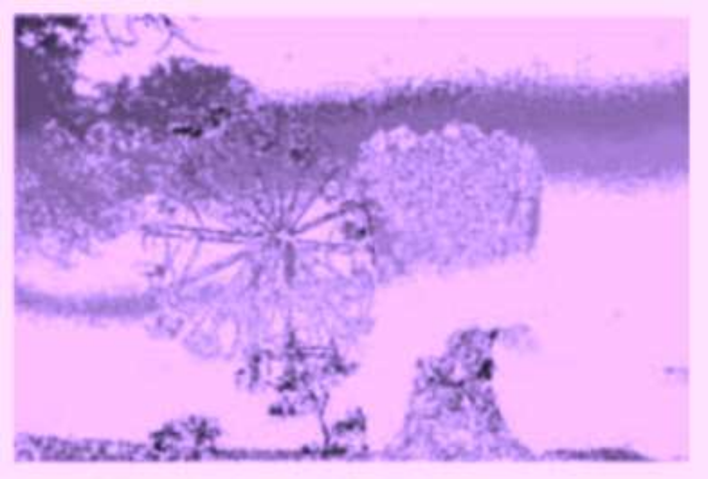
1 =	true	=	white pixel
0 =	false	=	black pixel

AND

$=$

OR

$=$



XOR

$=$

NOT

Linear Shift-Invariance & Convolution

$$g(x,y) = \iint f(x',y')h(x,y;x',y')dx'dy'$$

128 x 128

128^4 = 268.4 million

$$h(x,y;x',y') = h(x'',y'') = h(x-x',y-y')$$

The convolution integral

2-D

$$g(x,y) = \iint_{\infty}^{\infty} f(x',y')h(x-x',y-y')dx'dy'$$

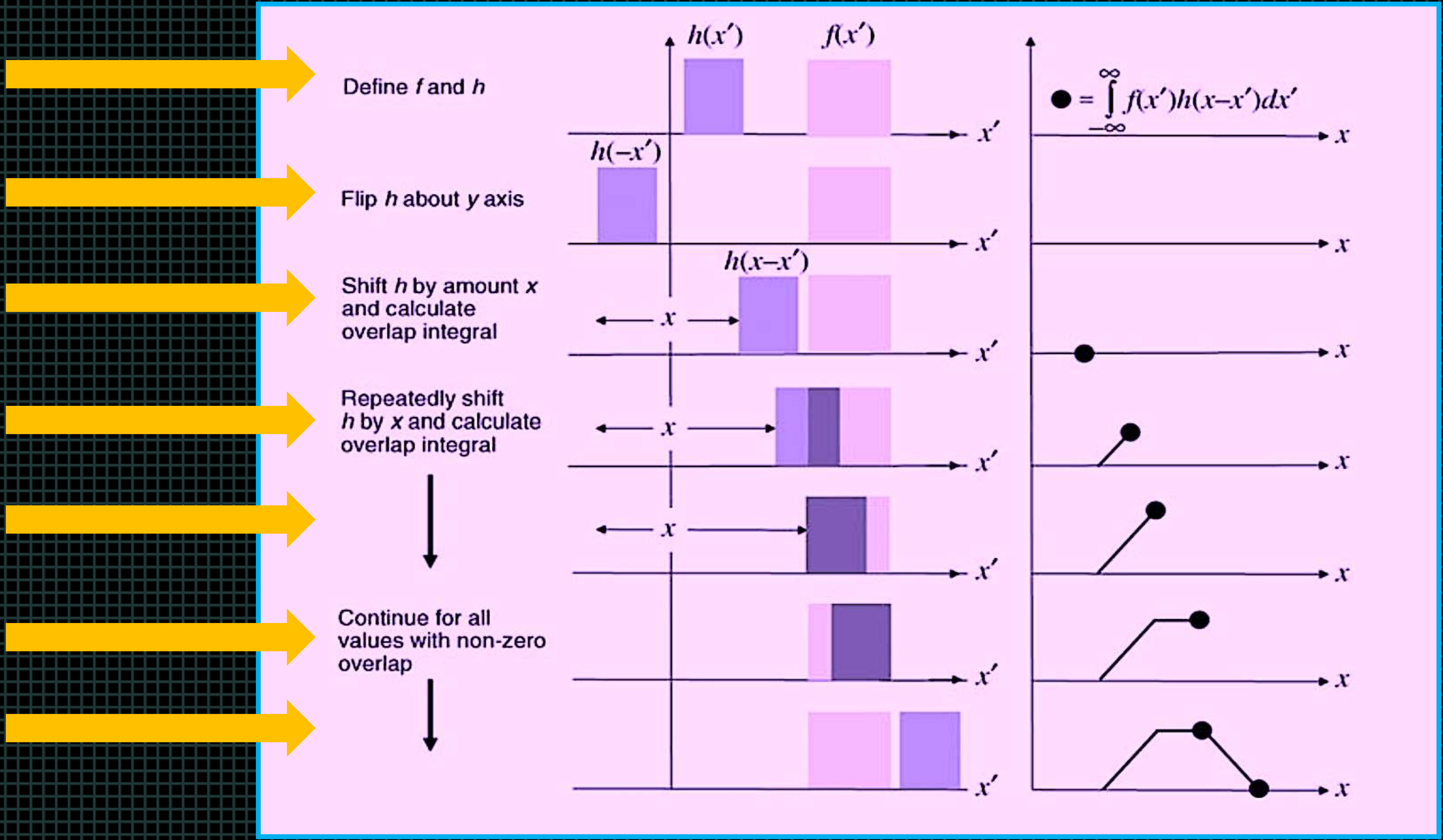
1-D

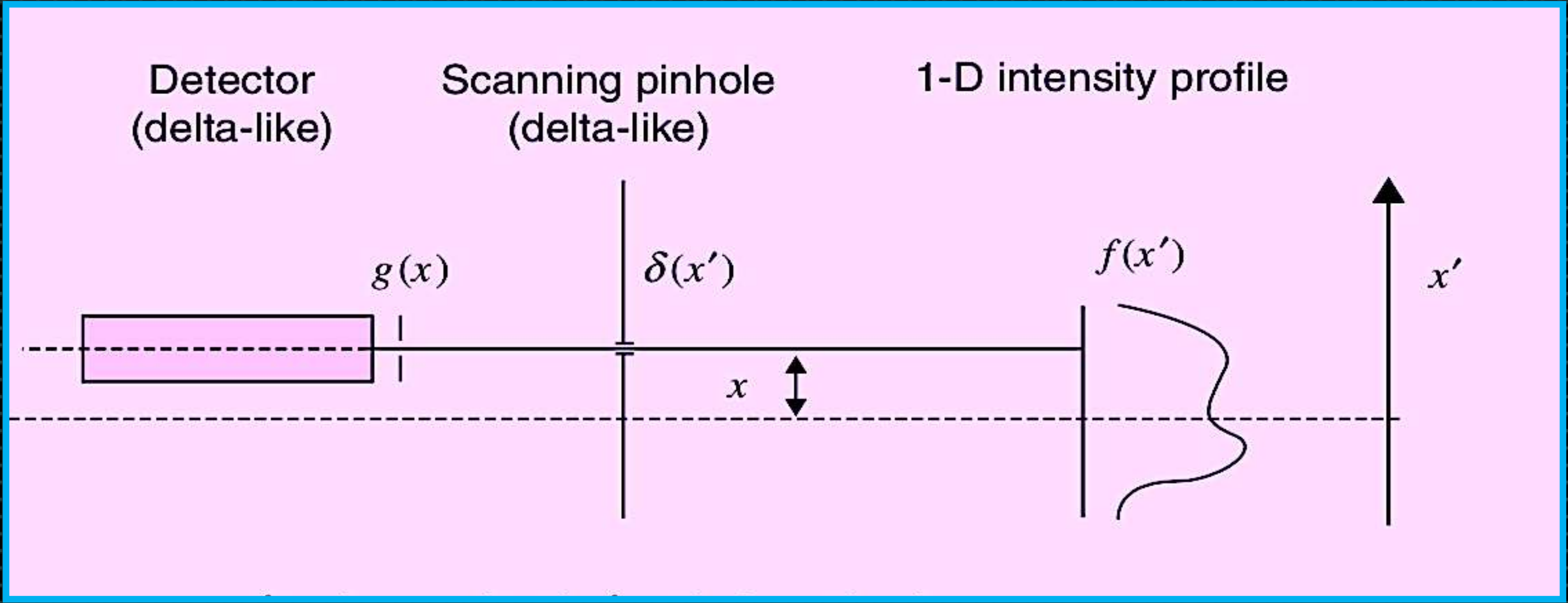
$$g(x) = \int_{\infty}^{\infty} f(x')h(x-x')\,dx'$$

$$g(x, y) = f(x, y) ** h(x, y)$$ 2-D

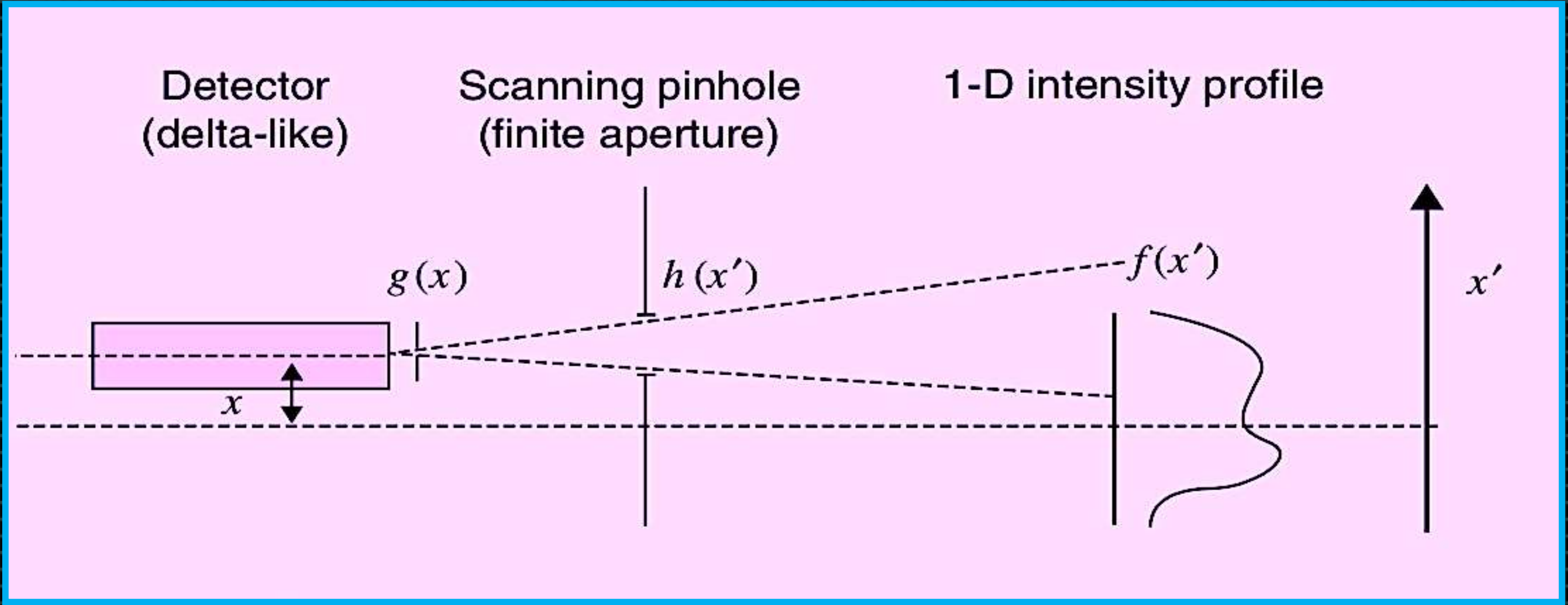
$$g(x) = f(x) * h(x)$$ 1-D

Calculation of a 1-D Convolution Integral

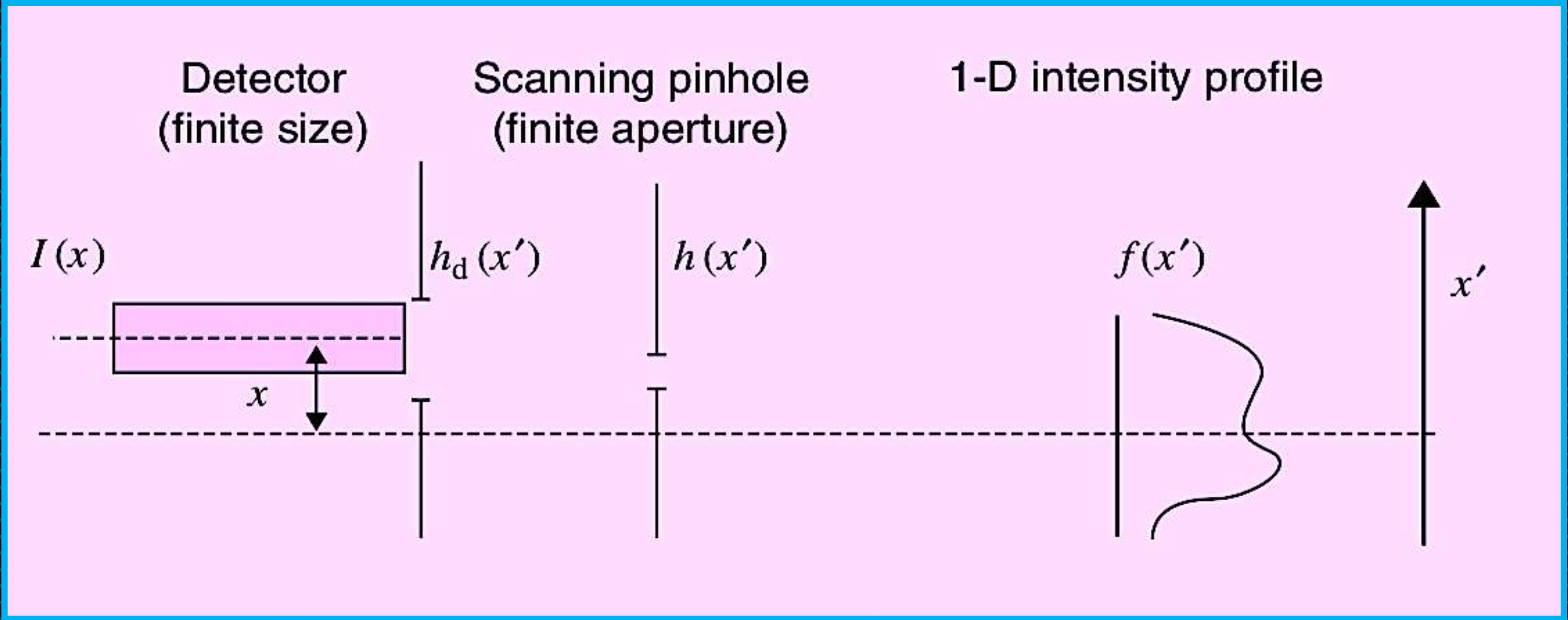

Define f and h
Flip h about y axis
Shift h by amount x and calculate overlap integral
Repeatedly shift h by x and calculate overlap integral
Continue for all values with non-zero overlap
$h(x')$
$f(x')$
$h(-x')$
$h(x-x')$
x'
x
$\bullet = \int_{-\infty}^{\infty} f(x')h(x-x')dx'$



$$g(x) = \int f(x')\, h(x, x')\, dx' = \int f(x')\, \delta(x - x')\, dx' = f(x)$$

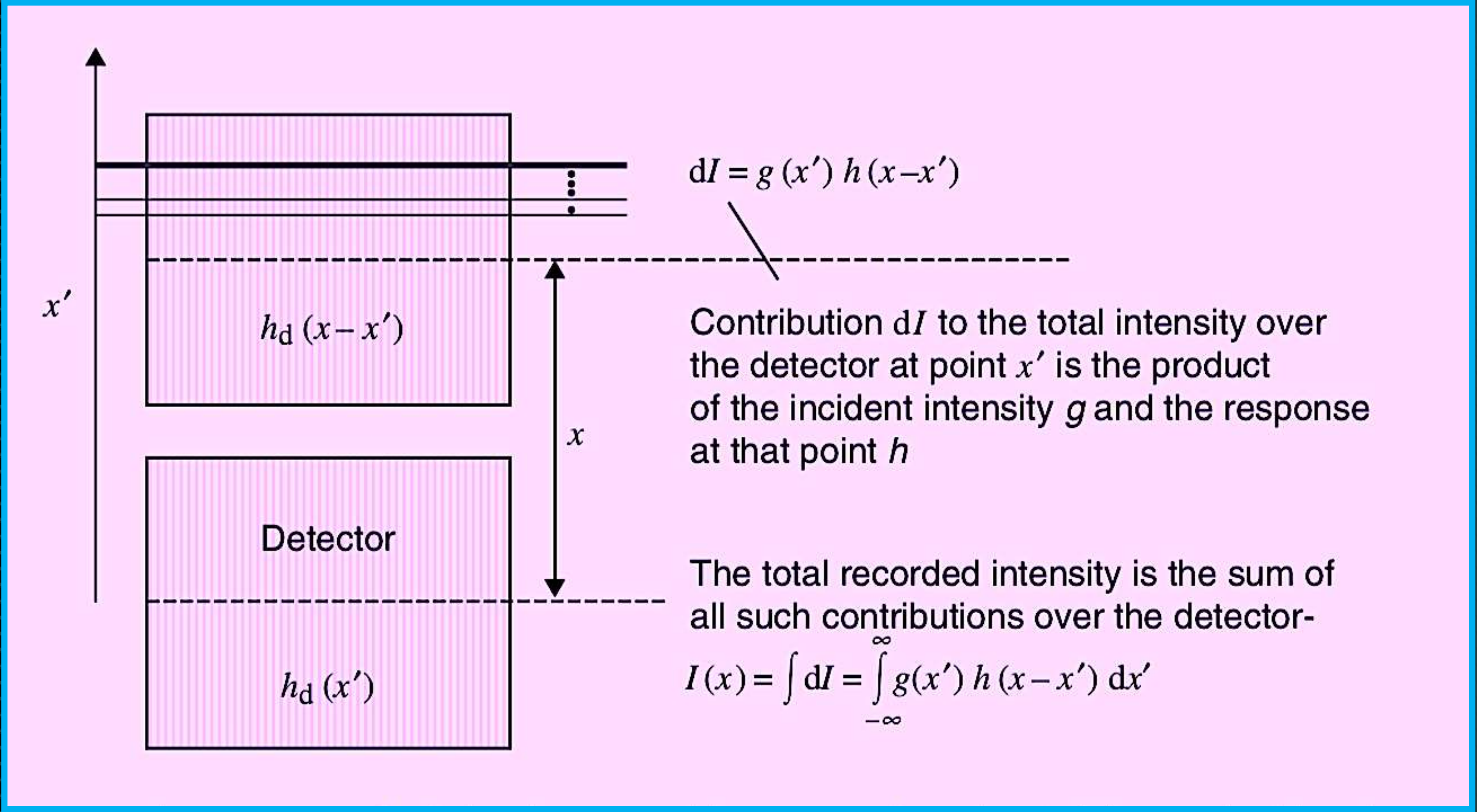


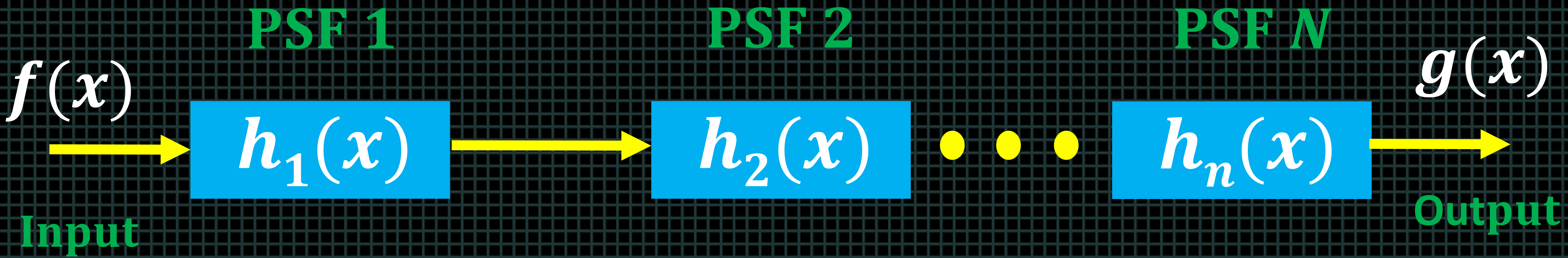
$$g(x) = \int f(x')\, h(x, x')\, dx' = \int f(x')\, h(x - x')\, dx'$$



$$I(x) = \int g(x')\, hd(x, x')\, dx' \quad \text{where} \quad g(x) = \int f(x')\delta(x - x')\, dx'$$

$$I(x) = f(x') * h(x,)\, hd(x\ )$$


x'
$h_d(x - x')$
Detector
$h_d(x')$
x
$dI = g(x')\,h(x - x')$
Contribution dI to the total intensity over the detector at point x' is the product of the incident intensity g and the response at that point h
The total recorded intensity is the sum of all such contributions over the detector-
$I(x) = \int dI = \int_{-\infty}^{\infty} g(x')\,h(x - x')\,dx'$



$$g(x) = f(x) * h_1(x) * h_2(x) \ldots * hn(x)$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

=

–1	–1	–1
–1	8	–1
–1	–1	–1

12	11	12	13	13	9
10	8	10	11	8	13
32	36	40	35	42	40
40	37	38	36	46	41
41	36	89	39	42	39
42	37	39	43	45	38

$$f_i = \sum_{1-j}^{9} w_k I_k(i)$$

$$= (-1x10) + (-1x11) + (-1x8) + (-1x40) + (8x35)$$

$$+ (-1x42) + (-1x38) + (-1x36) + (-1x46) = 14$$

$$g \; {}_{j} = \sum_{i} f i h j \quad i$$

$$g k l = \sum_{j} \sum_{i} f i j h_{k} \quad i, l \qquad j$$

Understanding the engineering of image formation

The Pixel

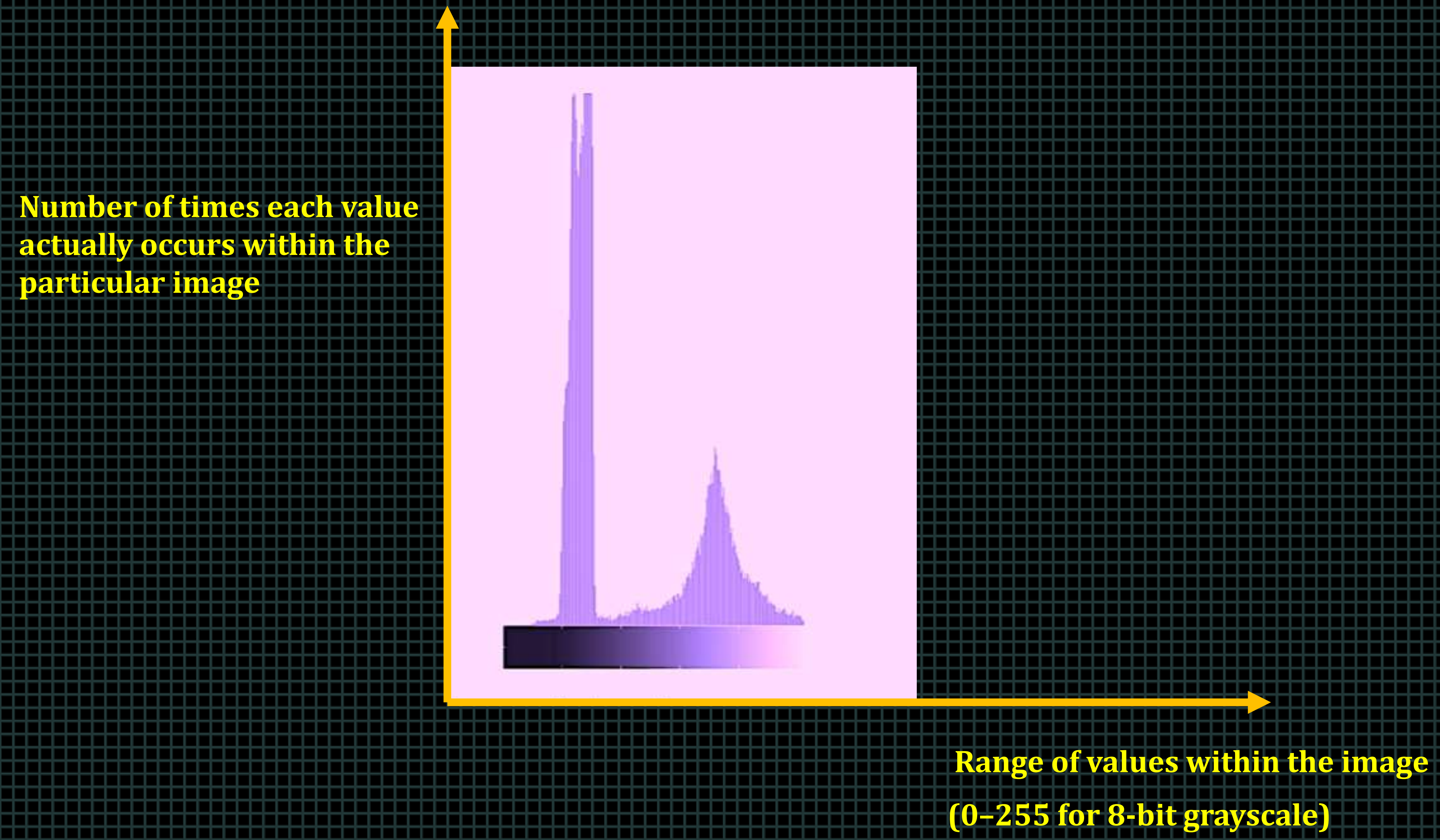
Pixel : Picture element

Histogram and Equalization

Introduction to Histograms

An image histogram is a plot of the relative frequency of occurrence of each of the permitted pixel values in the image against the values themselves.

A discrete probability density function which defines the likelihood of a given pixel value occurring within the image.


Number of times each value actually occurs within the particular image
Range of values within the image
(0–255 for 8-bit grayscale)

Gray-scale value : (0 -255)

```
initialize all histogram array entries to 0

for each pixel I(i, j) within the image I
    histogram(I(i,j)) = histogram(I(i,j)) + 1

end
```

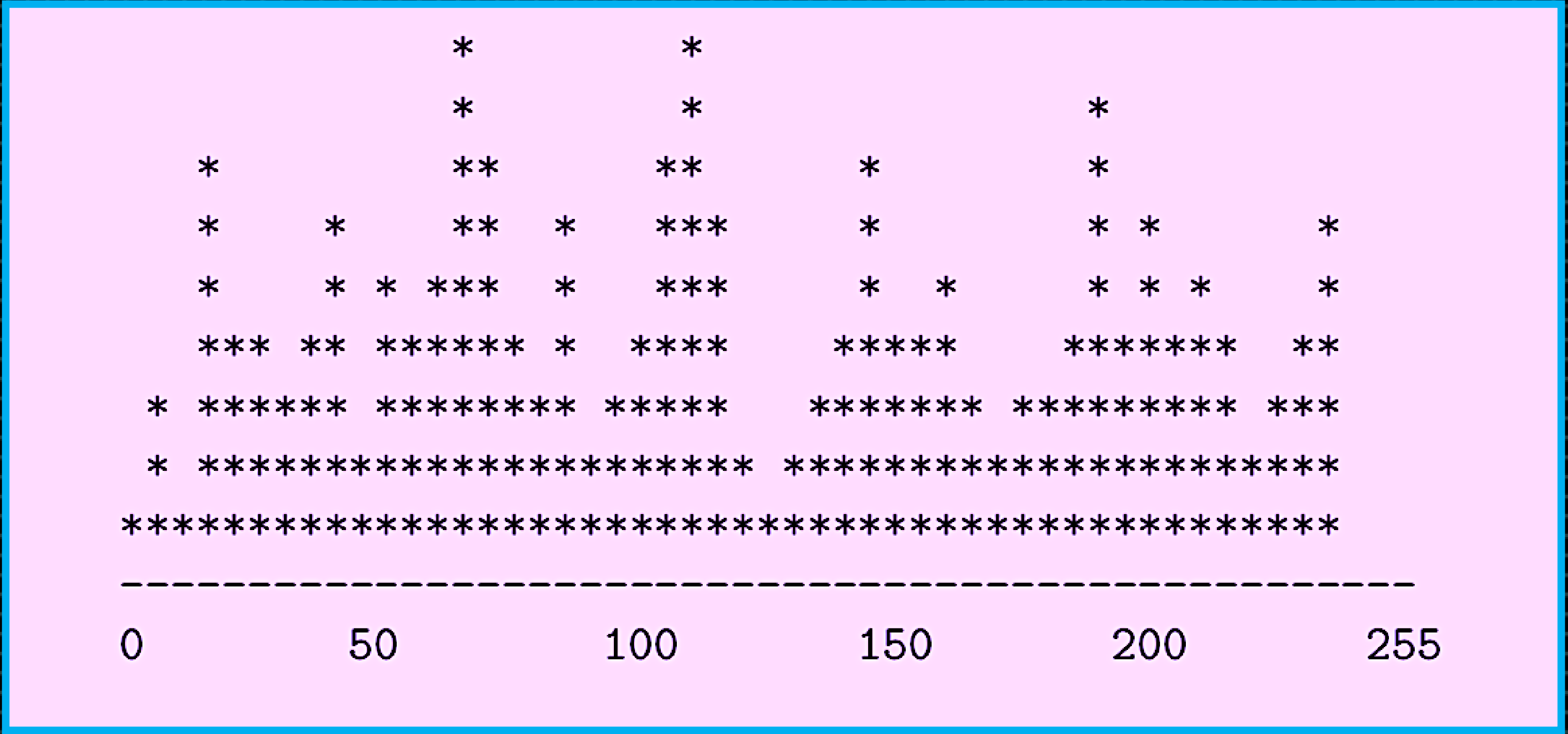
- Basic contrast present in the image

- Potential differences in the color distribution of the image foreground and background scene components.

Histogram

A histogram uses a bar graph to profile the occurrences of each gray level present in an image

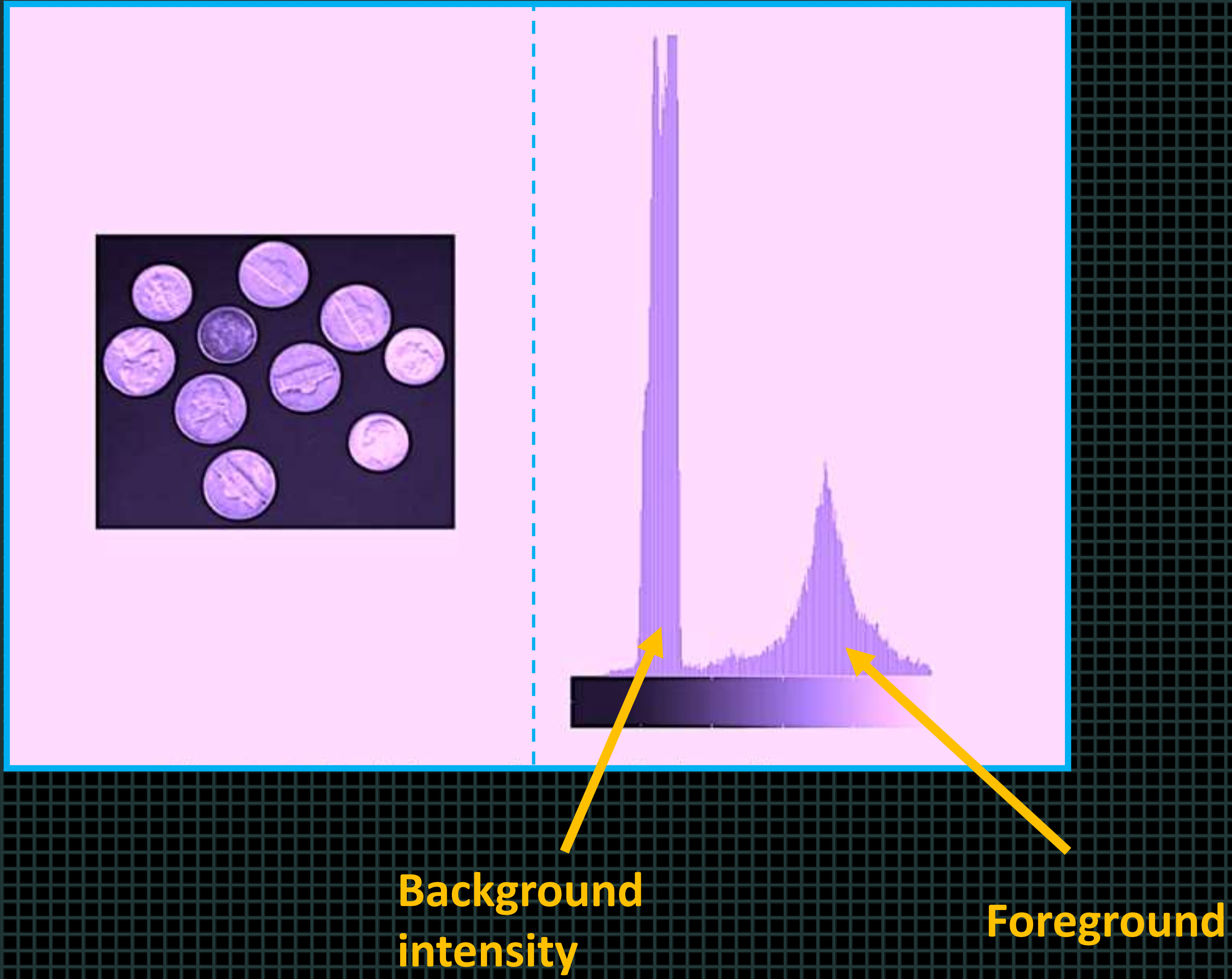
Histogram of a simple gray scale image

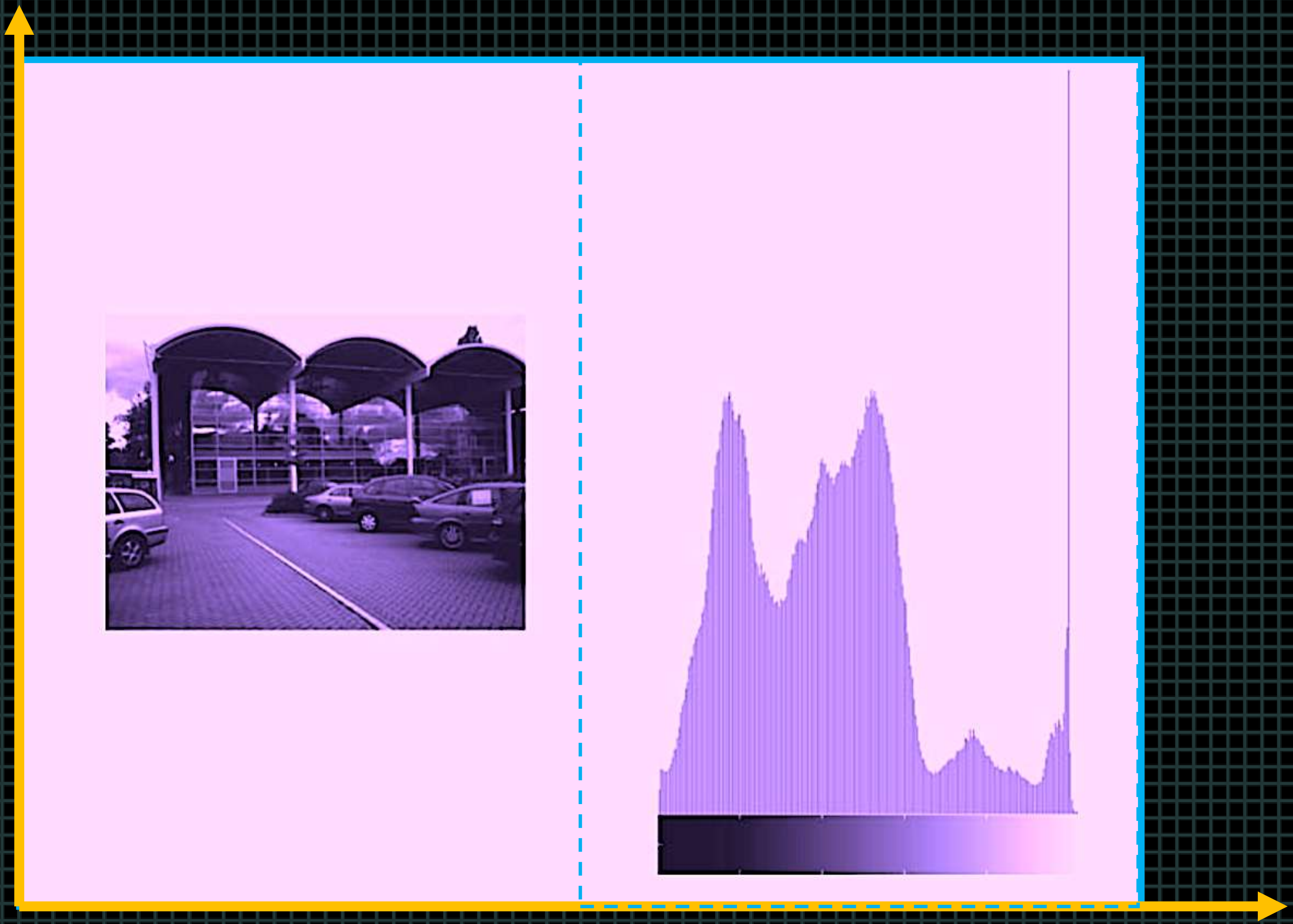


Simple Histogram

```
      **   ** ***
      **   ** ***
     ***   ** ***
     ***  *** ***
  *  *** **** ***
  * ********* ***
 ****************
-----------------------------------------------------
0        50        100       150       200       255
```

Histogram of a poorly scanned image


Background intensity
Foreground



Histogram of a more complex scene

Introduction to Histogram Equalization

Image with poor contrast

Equalization causes a histogram with bins (vertical lines)grouped closely together to “spread out” into a flat or equalized histogram.

Image with poor contrast

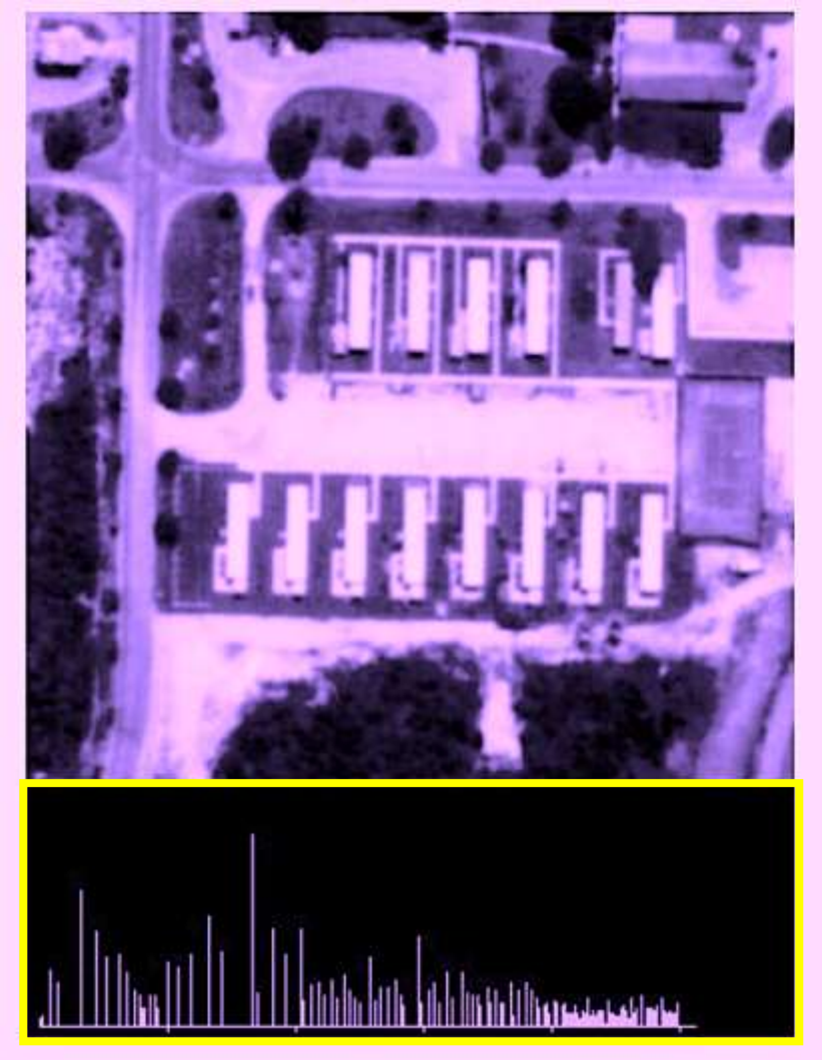
$$b(x,y) = f[c(x,y)]$$

C : image with poor histogram
b : new image with improved histogram
f : transformation function

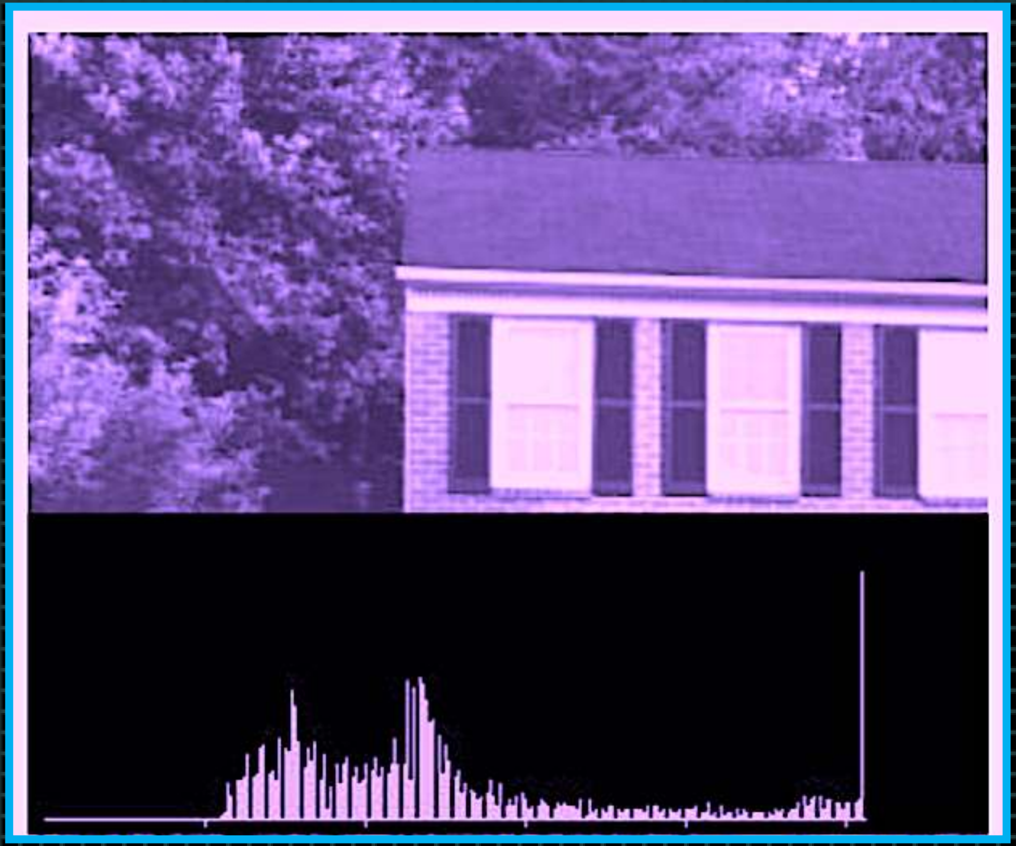
$$p1(a) = \frac{1}{Area_1} H_1(a)$$

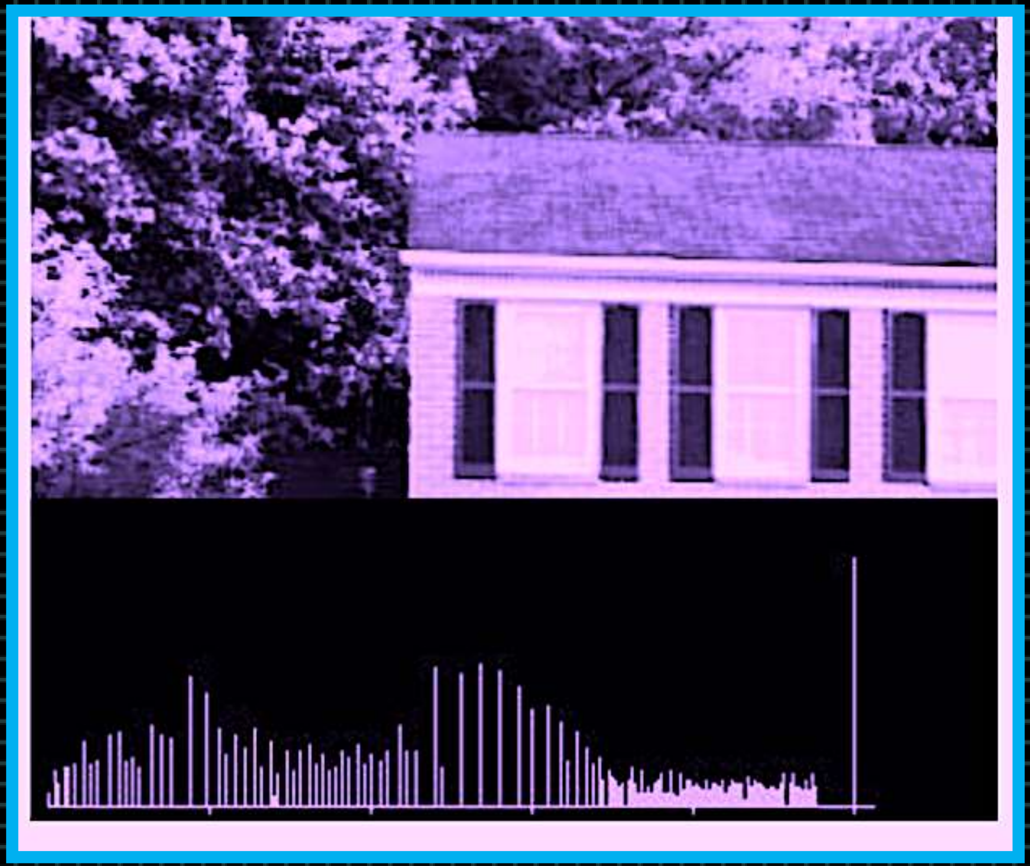
p1(a) : probability of finding a pixel with the value "a" in the image
$Area_1$: area or number of pixels in the image
$H_1(a)$: histogram of the image

Image with poor contrast

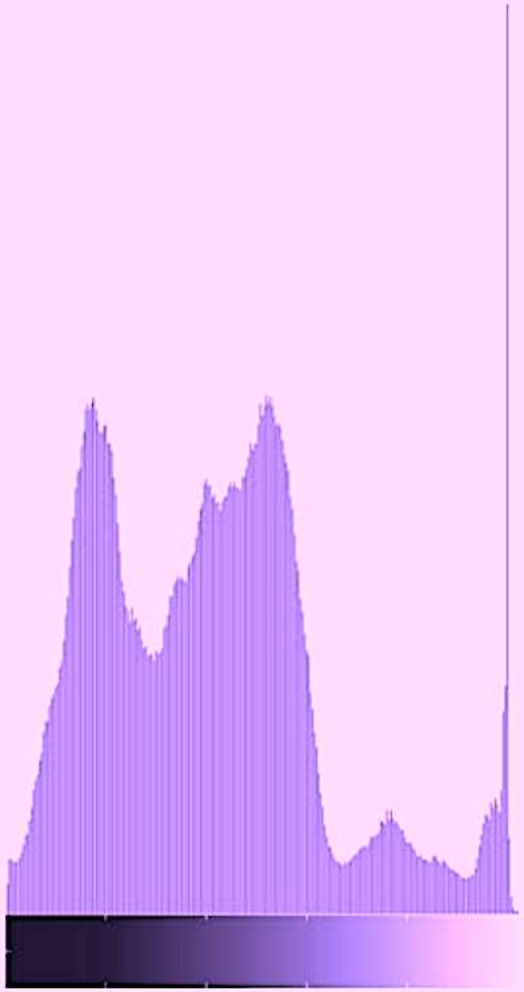


Equalized image





Histogram for threshold selection



Histogram of a more complex scene

Adaptive thresholding

- Different threshold at each pixel location .

$$t = mean + C$$

$$t = median + C$$

$$floor\left(\frac{max - min}{2}\right) + C$$

Image Enhancement Techniques

Enhancement by filtering

Spatial domain filtering

Filtering on the actual pixel rather than in the frequency domain

Image Filters

Linear Filters

Non Linear Filters

Connectivity

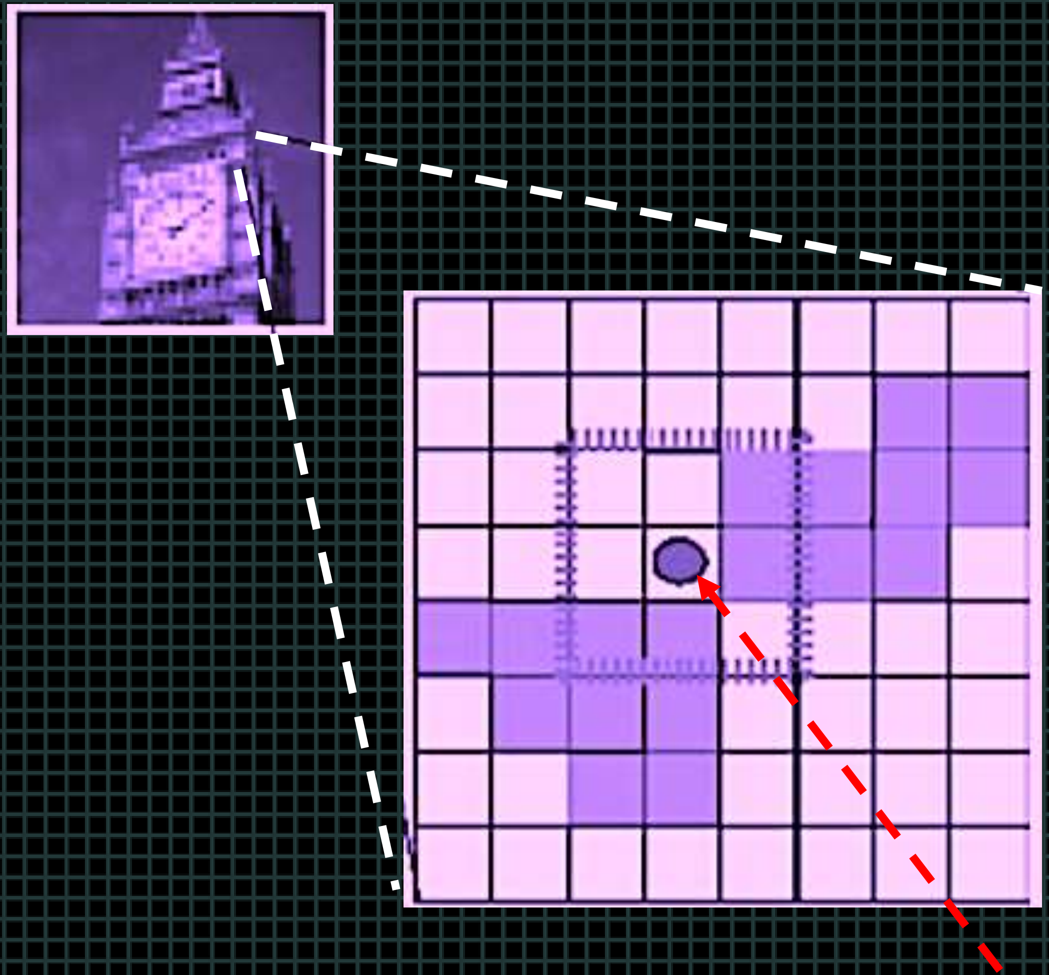
Deciding which pixels are connected to each other

E.g.

4 – connectivity : N, W, E, S

8 – connectivity : N, NW, W, NE, SE, E, SW, S

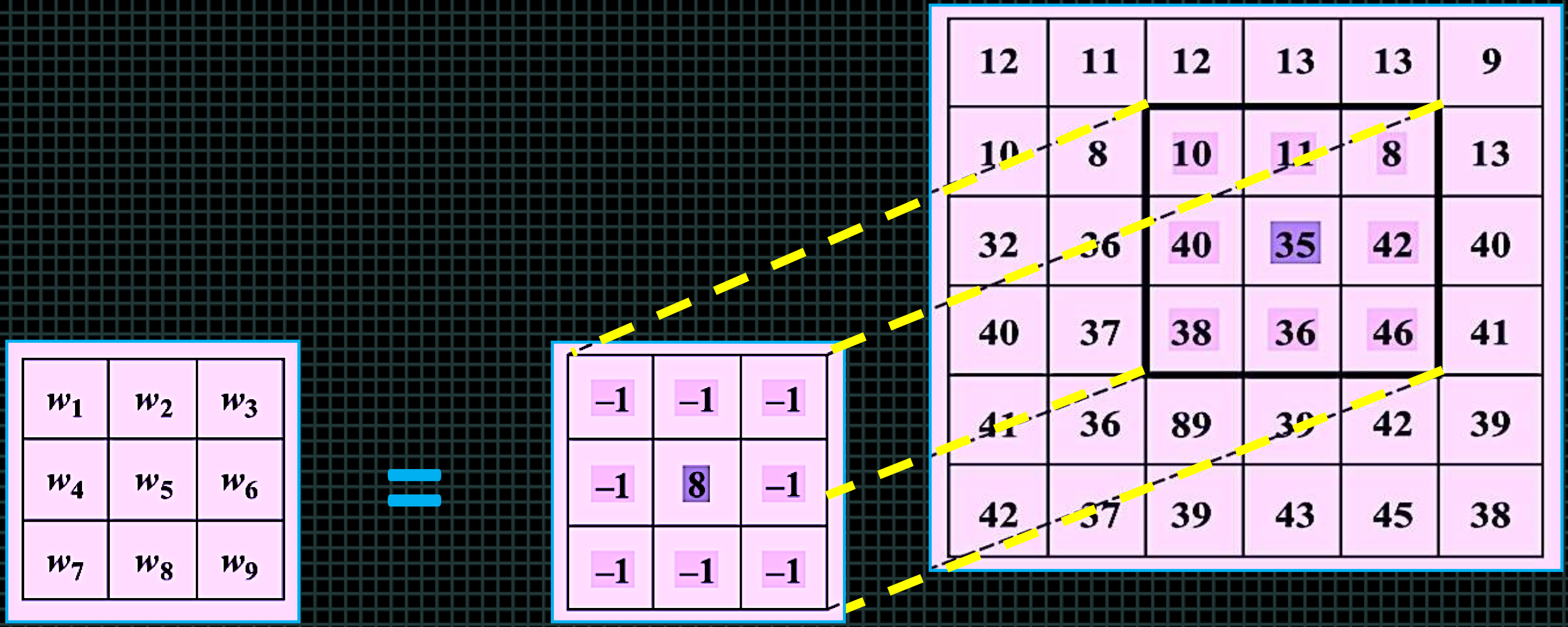
NW	N	NE
W	(i,j)	E
SW	S	SE



3 X 3 neighborhood centered at this point

The Filter Kernel

- Also known as the *mask*



$$f_i = \sum_{1-j}^{9} w_k I_k(i)$$

$$= (-1x10) + (-1x11) + (-1x8) + (-1x40) + (8x35)$$

$$+ (-1x42) + (-1x38) + (-1x36) + (-1x46) = 14$$

Filtering with an N X N = 3X3 kernel

Row and Column Indices

$$f(x,y) = \sum_{i=Imin}^{I_{max}} \cdot \sum_{j=Jmin}^{j_{max}} w(i,j)I(x+i,y+j)$$

$i = 0, j = 0$: *center pixel of the kernel)*

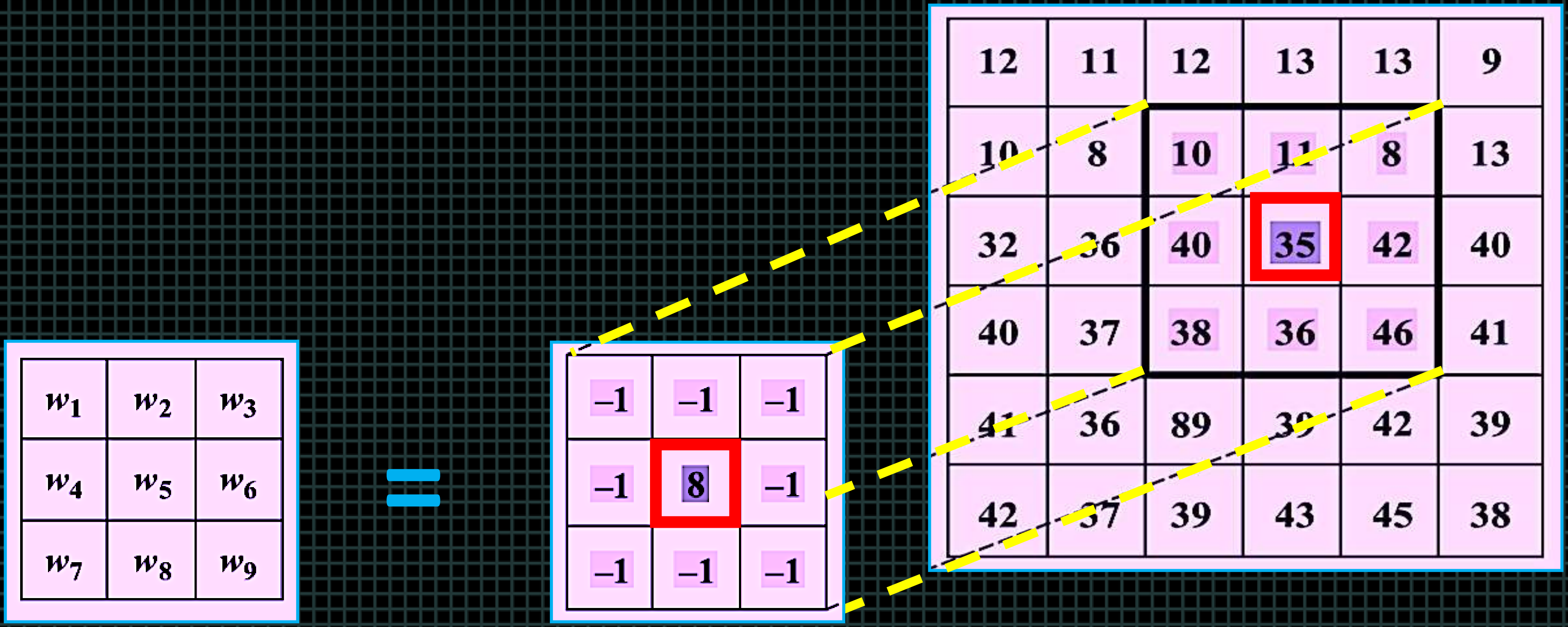
$(I_{max} - Imin + 1, Jmax - Jmin + 1)$: *size of kernel center pixel*

Linear Indices

$$f_i = \sum_{k=1}^{N} w_k I_k(i)$$

$I_{k(i)}$: *neighborhood pixel of the ith image pixel*

k: *a linear index running over the neighborhood*



$$f_i = \sum_{1-j}^{9} w_k I_k(i)$$

$$= (-1x10) + (-1x11) + (-1x8) + (-1x40) + (8x35)$$
$$+ (-1x42) + (-1x38) + (-1x36) + (-1x46) = 14$$

Filtering with an N X N = 3X3 kernel

Linear Filtering Steps

Define filter kernel

↓

Slide kernel over image

↓

Multiply pixels under the kernel by weights

↓

Copy resulting value to same location in new image

Geometric Operations

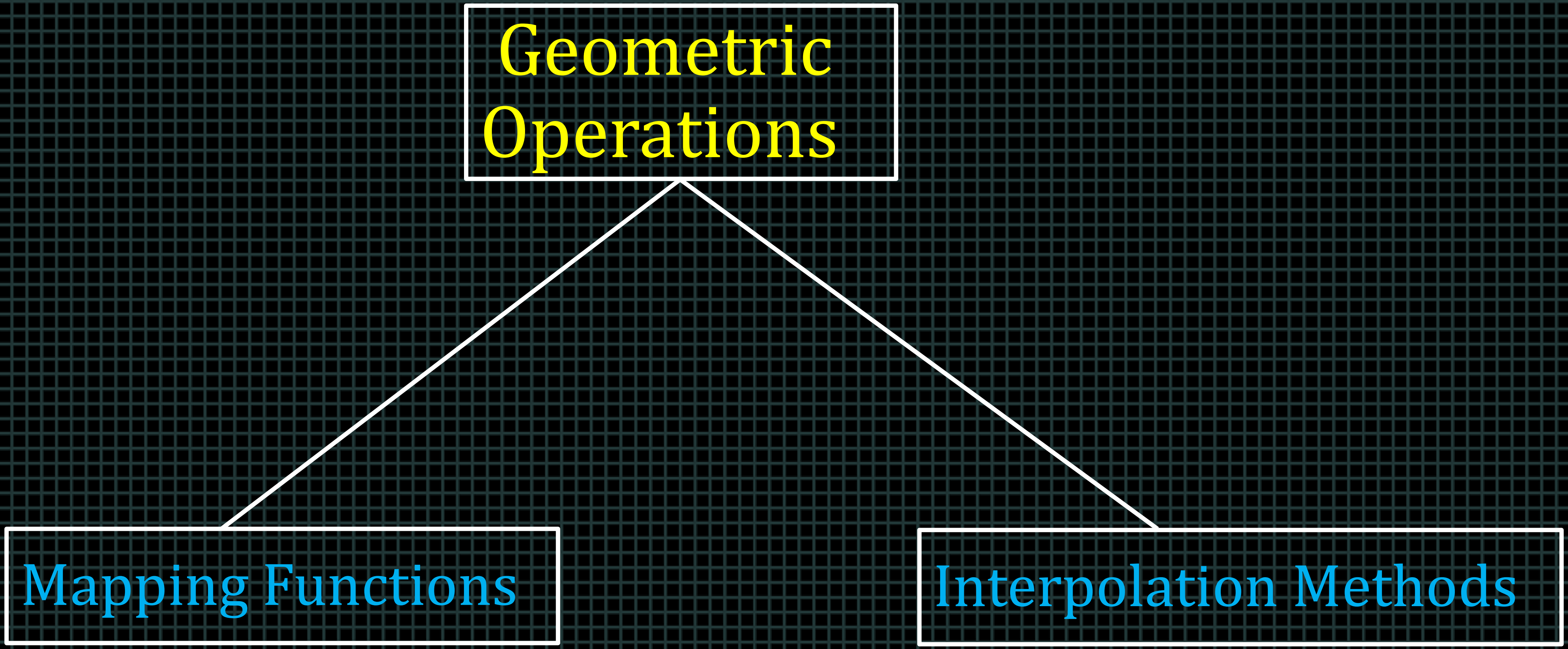
Overview of Geometric Operations

- Modify geometry of an image by repositioning pixels

Rotate | Flip | Crop | Resize

Common uses of geometric operations

- Correcting geometric distortions
- Creating special effects
- As part of image registration


Geometric Operations
Mapping Functions
Interpolation Methods

Mapping and Affine Transformation

$$f(x,y) \rightarrow g(x',y')$$

Mapping function:

$$(x',y') = T(x,y)$$

$$x' = T_x(x,y)$$

$$y' = T_y(x,y)$$

Where:

$T_x, T_y =$ *polynomials in x and y*

Affine transformation:

$$x' = a_0 x + a_1 y + a_2$$

$$y' = b_0 x + b_1 y + b_2$$

$$x' = a_0 x + a_1 y + a_2$$
$$y' = b_0 x + b_1 y + b_2$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

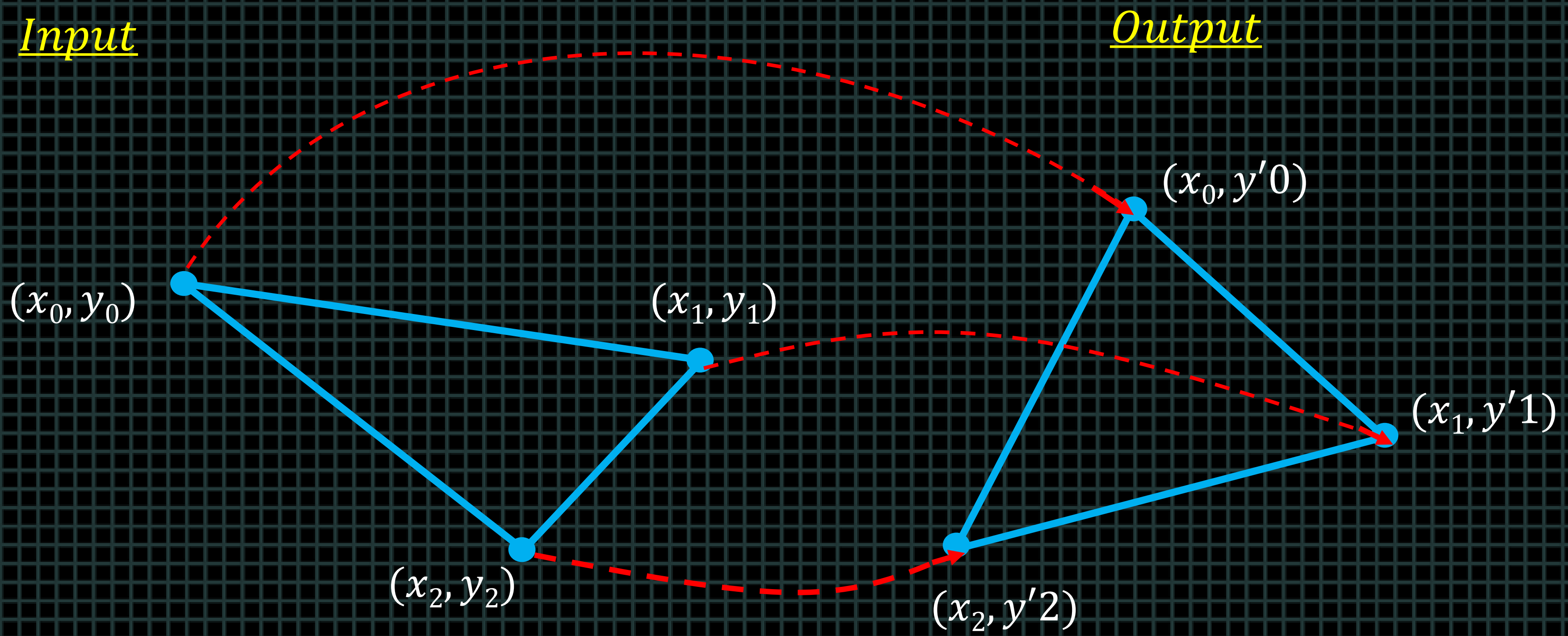
Summary of Affine Transformation Coefficients

Transformation	a_0	a_1	a_2	b_0	b_1	b_2
Translation by Δ_x, Δ_y	1	0	Δ_x	0	1	Δ_y
Scaling by a factor $[s_x, s_y]$	s_x	0	0	0	s_y	0
Counterclockwise rotation by angle θ	$\cos\theta$	$\sin\theta$	0	$-\sin\theta$	$\cos\theta$	0
Shear by a factor $[\mathrm{sh}_x, \mathrm{sh}_y]$	1	sh_y	0	sh_x	1	0

E.g. :

Generate the affine transformation matrix for :

- *rotation by 30°*
- *scaling by a factor 3.5 in both dimensions*
- *translation by[25,15] pixels*
- *Shear by a factor [2,3]*



Mapping one triangle onto another by an affine transformation.

$\cos 30^\circ = 0.866$ *and* $\sin 30^\circ = -0.500$

rotation by 30°

$$(a) = \begin{bmatrix} 0.866 & -0.500 & 0 \\ 0.500 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

scaling by a factor 3.5,3.5

$$(b) = \begin{bmatrix} 3.5 & 0 & 0 \\ 0 & 3.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformation	a_0	a_1	a_2	b_0	b_1	b_2
Translation by Δ_x, Δ_y	1	0	Δ_x	0	1	Δ_y
Scaling by a factor $[s_x, s_y]$	s_x	0	0	0	s_y	0
Counterclockwise rotation by angle θ	$\cos\theta$	$\sin\theta$	0	$-\sin\theta$	$\cos\theta$	0
Shear by a factor $[\mathrm{sh}_x, \mathrm{sh}_y]$	1	sh_y	0	sh_x	1	0

translation by[25,15] pixels

$$(c) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 25 & 15 & 1 \end{bmatrix}$$

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

Shear by a factor [2,3]

$$(d) = \begin{bmatrix} 1 & 3 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Gray - Level Transformations

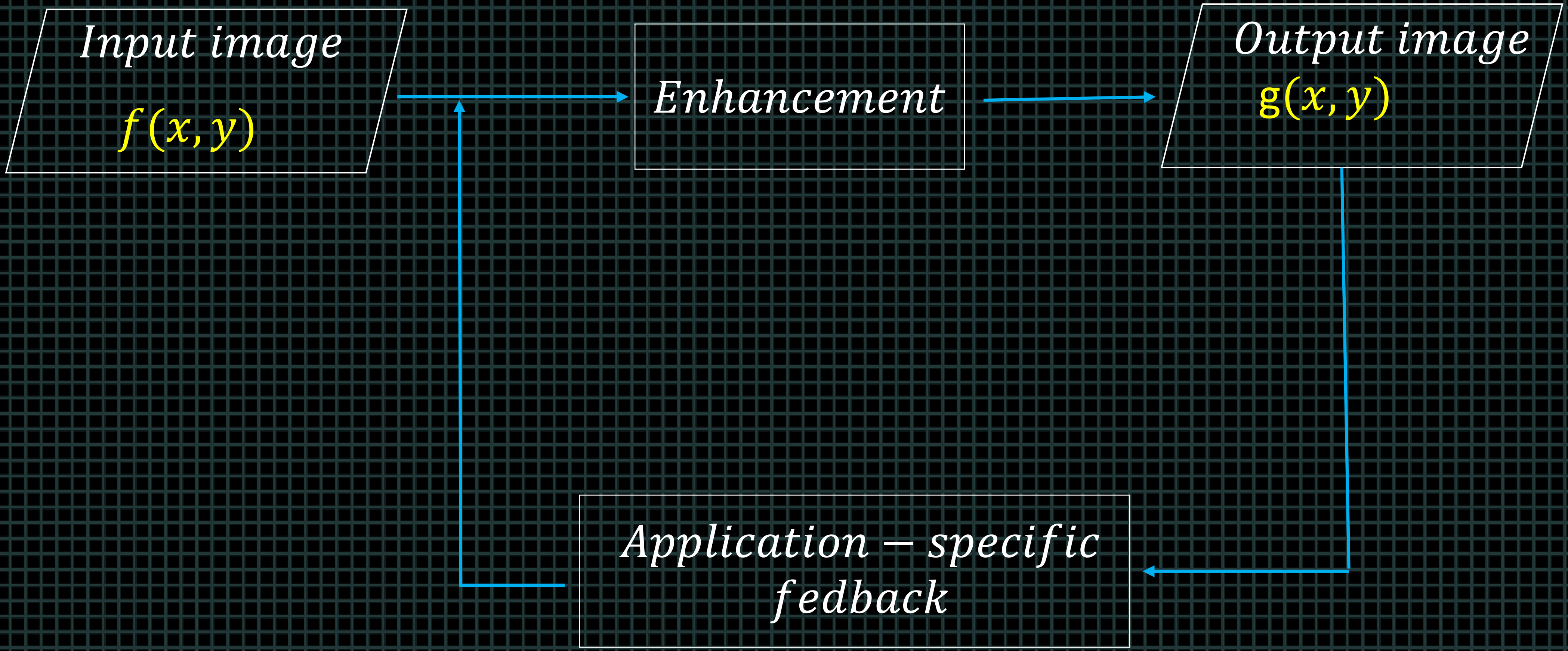


Image enhancement process

Overview of gray-level transformations

Spatial transformation :

$$g(x,y) = T[f(x,y)]$$

where :

$g(x,y)$: *processed image*
$f(x,y)$: *original image*
T: *operator on* $f(x,y)$

$$s = T[r]$$

where :

r : *original gray level of a pixel*

s: *resulting gray level of a pixel*

$$s = c \, . \, r + b$$

where : r: *origian pixel value*

s: *resulting pixel value*

c: *constant for controlling the contrast of output image*

Neighborhood Processing

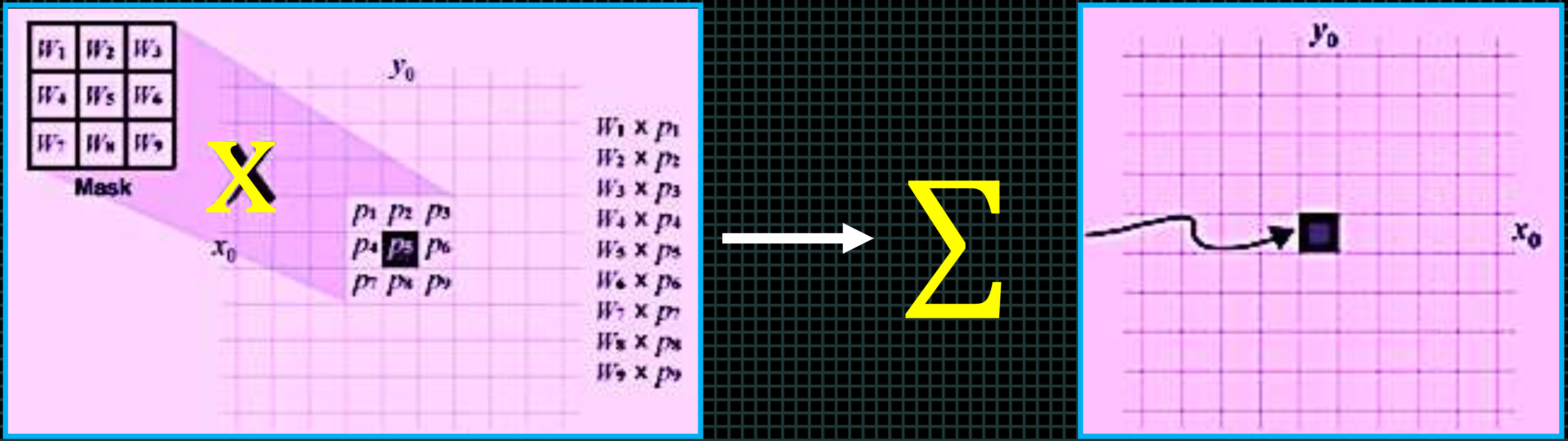
Steps :

- Define a reference point in the input image
- Perform reference point neighborhood operation in input image
- Apply result of operation to the pixel of same coordinates in the output image.
- Repeat the process for every pixel in the input image

Linear Filters

Nonlinear Filters

Convolution and Correlation



Neighborhood processing : linear filtering

Convolution

$$A * B = \sum_{j=-\infty}^{\infty} A(j) \bullet B(x-j)$$

Example :

Let : $A = \{0,1,2,3,1,0\}$ $\quad B = \{1,3,-1\}$

Solution

$$(0 \times (-1)) + (0 \times 3) + (1 \times 1) = 1$$

A		0	1	2	3	2	1	0
B	−1	3	1					
$A * B$		1						

Array B mirrored and center value aligned with 1st value of array A

$$\big(0 \text{ x } (-1)\big) + (1 \text{ x } 3) + (2 \text{ x } 1) = 5$$

A	0	1	2	3	2	1	0
B	−1	3	1				
$A * B$	1	5					

Array B shifted one position to the right

$$\big(1 \times (-1)\big) + (2 \times 3) + (3 \times 1) = 8$$

A	**0**	**1**	**2**	**3**	**2**	**1**	**0**
B	**−1**	**3**	**1**				
A * ***B***	**1**	**5**	**8**				

Array B shifted another position to the right

$$\big(1 \text{ x } (-1)\big) + (2 \text{ x } 3) + (3 \text{ x } 1) = 8$$

A	0	1	2	3	2	1	0
B		−1	3	1			
$A * B$	1	5	8				

Array B shifted another position to the right

$$\big(2 \times (-1)\big) + (3 \times 3) + (2 \times 1) = 9$$

A	**0**	**1**	**2**	**3**	**2**	**1**	**0**
B			$-$**1**	**3**	**1**		
$A * B$	**1**	**5**	**8**	**8**			

Array B shifted another position to the right

$$\left(3 \times (-1)\right) + (2 \times 3) + (1 \times 1) = 4$$

A	0	1	2	3	2	1	0
B				−1	3	1	
$A * B$	1	5	8	8	4		

Array B shifted another position to the right

$$\big(2 \text{ x } (-1)\big) + (1 \text{ x } 3) + (0 \text{ x } 1) = 1$$

A	0	1	2	3	2	1	0
B					-1	3	1
$A * B$	1	5	8	8	4	1	

Array B shifted another position to the right

$$\big(1 \text{ x } (-1)\big) + (0 \text{ x } 3) + (0 \text{ x } 1) = -1$$

A	0	1	2	3	2	1	0	
B						−1	3	1
$A * B$	1	5	8	8	4	1	−1	

Array B shifted another position to the right

$$A = \{0,1,2,3,1,0\}$$

$$B = \{1,3,-1\}$$

$$\boldsymbol{A} * \boldsymbol{B} = \{\mathbf{1} \quad \mathbf{5} \quad \mathbf{8} \quad \mathbf{8} \quad \mathbf{4} \quad \mathbf{1} \quad \mathbf{-1}\}$$

2-D Convolution and Correlation

$$g(x,y) = \sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(j,k) \bullet f(x-j, y-k)$$

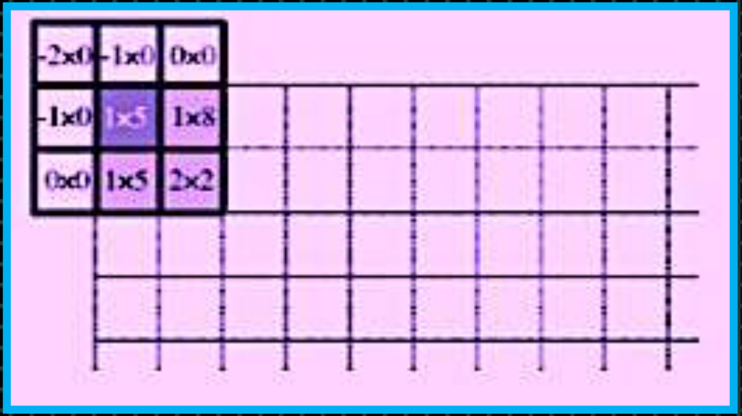
OR

$$g(x,y) = \sum_{k=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} h(j,k) \bullet f(x-j, y-k)$$

m_2 = half of mask's width
n_2 = half of mask's height

$$m_2 = [m/2] \qquad n_2 = [n/2]$$

$$A = \begin{bmatrix} 5 & 8 & 3 & 4 & 6 & 2 & 3 & 7 \\ 3 & 2 & 1 & 1 & 9 & 5 & 1 & 0 \\ 0 & 9 & 5 & 3 & 0 & 4 & 8 & 3 \\ 4 & 2 & 7 & 2 & 1 & 9 & 0 & 6 \\ 9 & 7 & 9 & 8 & 0 & 4 & 2 & 4 \\ 5 & 2 & 1 & 8 & 4 & 1 & 0 & 9 \\ 1 & 8 & 5 & 4 & 9 & 2 & 3 & 8 \\ 3 & 7 & 1 & 2 & 3 & 4 & 4 & 6 \end{bmatrix} \qquad B = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$



$(-2x0) + (-1x0) + (0x0) + (-1x0) + (1x5)$
$+ (1x8) + (0x0) + (1x5) + (2x2)$

$$A * B = \begin{bmatrix} 20 & 10 & 2 & 26 & 23 & 6 & 9 & 4 \\ 18 & 1 & -8 & 2 & 7 & 3 & 3 & -11 \\ 14 & 22 & 5 & -1 & 9 & -2 & 8 & -1 \\ 29 & 21 & 9 & -9 & 10 & 12 & -9 & -9 \\ 21 & 1 & 16 & -1 & -3 & -4 & 2 & 5 \\ 15 & -9 & -3 & 7 & -6 & 1 & 17 & 9 \\ 21 & 9 & 1 & 6 & -2 & -1 & 23 & 2 \\ 9 & -5 & -25 & -10 & -12 & -15 & -1 & -12 \end{bmatrix}$$

Correlation

$$A \odot B = \sum_{j=-\infty}^{\infty} A(j) \bullet B(x+j)$$

2-D Correlation

$$g(x,y) = \sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(j,k) \bullet f(x+j, y+k)$$

$$g(x,y) = \sum_{k=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} h(j,k) \bullet f(x+j, y+k)$$

m_2 = half of mask's width
n_2 = half of mask's height
$m_2 = [m/2]$

Image Smoothing

(Lowpass Filtering)

- Attenuates high frequency components
- Attenuates fine details in image
- Preserve coarse details in image and homogeneous areas

Mean Filter

- Known as *neighborhood averaging* or *spatial smoothing filter*.

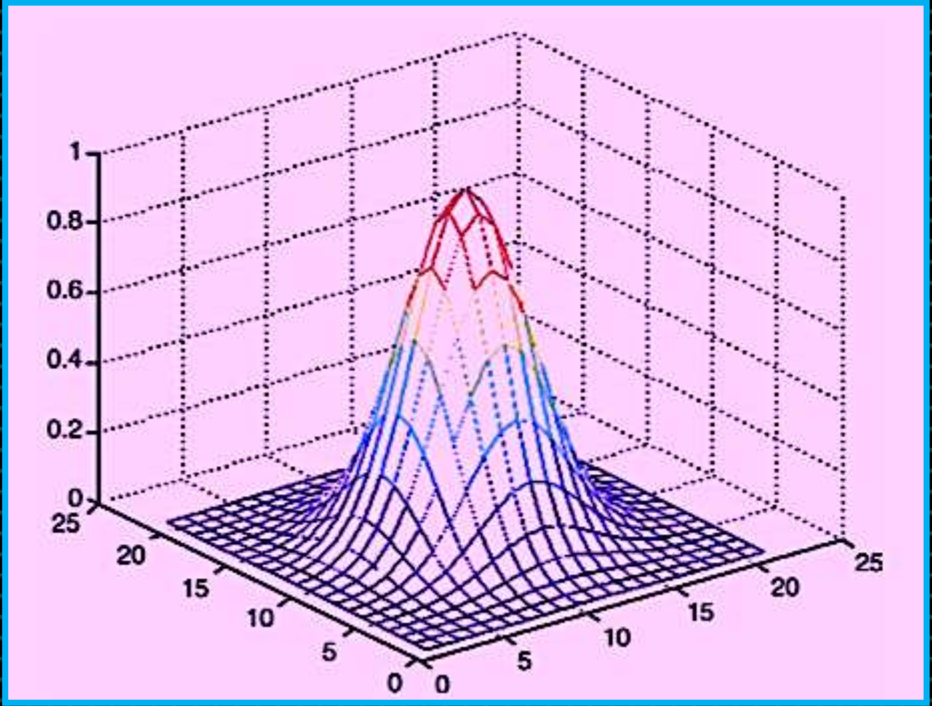
$$h(x,y) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Modified

$$h(x,y) = \begin{bmatrix} 0.075 & 0.125 & 0.075 \\ 0.125 & 0.2 & 0.125 \\ 0.075 & 0.125 & 0.075 \end{bmatrix}$$

Gaussian Blur Filter

$$h(x,y) = exp\left[\frac{(x^2+y^2)}{2\sigma^2}\right]$$



2D Gaussian function (with $\sigma = 3$)

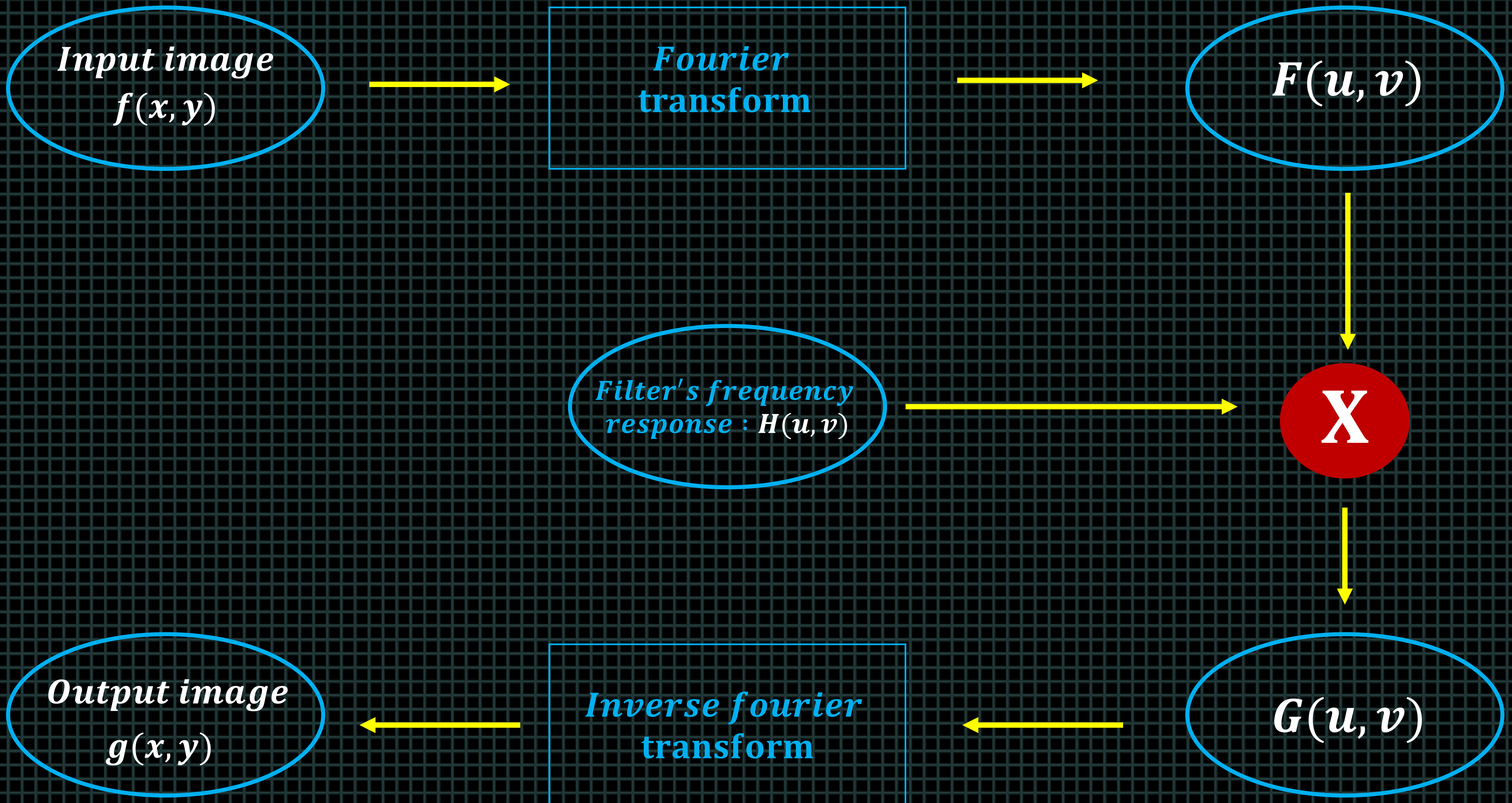
σ = controls the overall shape of the curve

Image Sharpening

(High-pass Filtering)

- Attenuates low frequency components
- Attenuates coarse details in image
- Preserve fine details in an image and homogeneous areas

Fourier Transform and Frequency Domain processing.

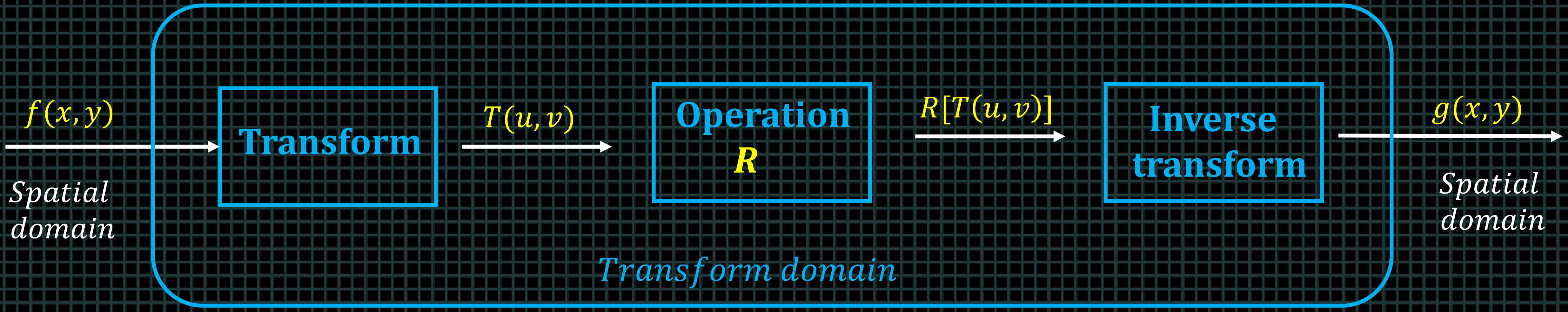

Input image
$f(x, y)$
Fourier transform
$F(u, v)$
Filter's frequency response : $H(u, v)$
X
$G(u, v)$
Inverse fourier transform
Output image
$g(x, y)$

$$g(x, y) = f(x, y) * h(x, y)$$

$$G(u, v) = F(u, v)\, H(u, v)$$

$where\ G, F, and\ H \rightarrow Fourier\ transform\ \ of\ g, f\ and\ h\ respectively$

Fourier Transform:



2D Forward Transform:

$$\boldsymbol{T(x,y)} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \boldsymbol{f(x,y)} \bullet \boldsymbol{r(x,y,u,v)}$$

where :

$$\left.\begin{aligned} u &= 0,1,2, \ldots M-1 \\ v &= 0,1,2, \ldots N-1 \end{aligned}\right\} \textit{transform variables}$$

$f(x,y)$: *input image*

$r(x,y,u,v)$: *forward transformation kernel*

2D Inverse Transform (Recovering original image)

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u,v) \bullet s(x,y,u,v)$$

$where:$

$$x = 0,1,2, \ldots M-1$$
$$y = 0,1,2, \ldots N-1$$

$s(x,y,u,v)$: $inverse\ transform\ kernel$

Transform pair:

$$T(x,y) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \bullet r(x,y,u,v)$$

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u,v) \bullet s(x,y,u,v)$$

original image

after applying lpf

Ideal Low-pass Filtering

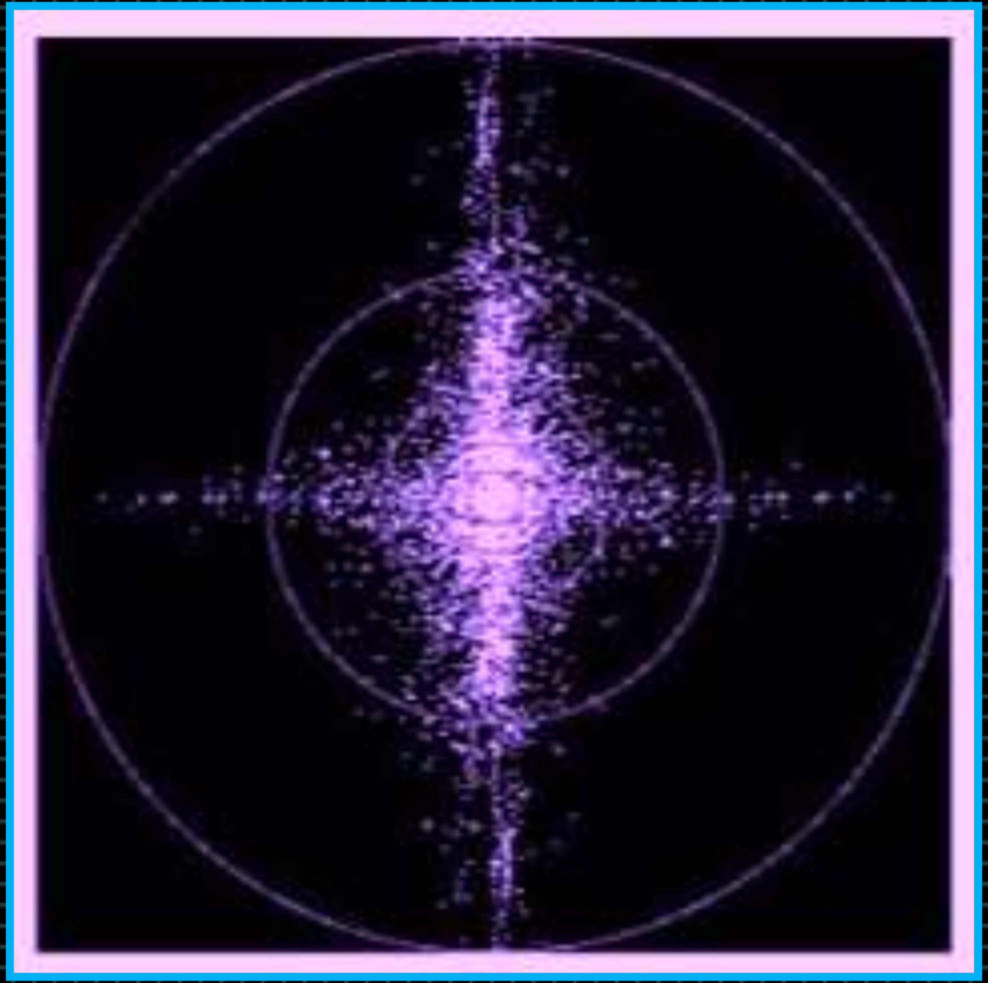
$$H_1(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

where :

$$D(u,v) = \sqrt{u^2 + v^2}$$

: *distance btw a point of coordinates and the origin of the 2d frequency plot*

original image



fourier spectrum

8 pixels

16 pixels

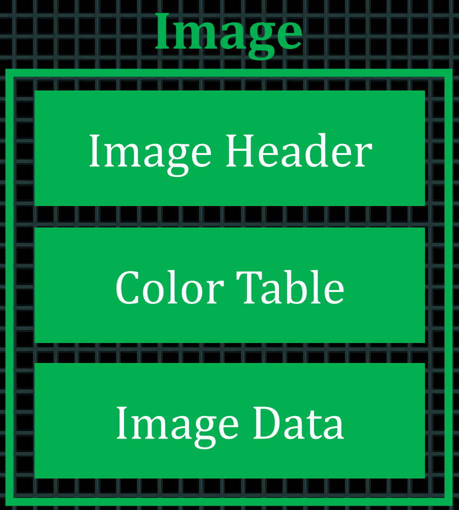
32 pixels

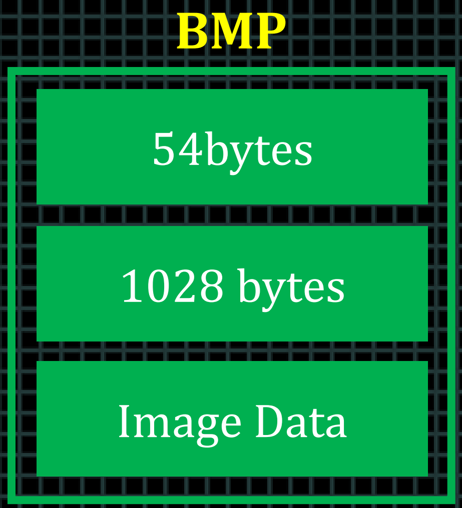
64 pixels

128 pixels

Projects

The bitmap Image





BMP (Windows) Header Format

offset	size	description
0	2	signature, must be 4D42 hex
2	4	size of BMP file in bytes (unreliable)
6	2	reserved, must be zero
8	2	reserved, must be zero
10	4	offset to start of image data in bytes
14	4	size of BITMAPINFOHEADER structure, must be 40
18	4	image width in pixels
22	4	image height in pixels
26	2	number of planes in the image, must be 1
28	2	number of bits per pixel (1, 4, 8, or 24)
30	4	compression type (0=none, 1=RLE-8, 2=RLE-4)
34	4	size of image data in bytes (including padding)
38	4	horizontal resolution in pixels per meter (unreliable)
42	4	vertical resolution in pixels per meter (unreliable)
46	4	number of colors in image, or zero
50	4	number of important colors, or zero

Operators

Operator

Function that acts on elements of a set to produce other elements of the same space.

Prewitt Operator

- Used to detect vertical and horizontal edges

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

Vertical Edges

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

Horizontal Edges

Sobel Operator

- Used to detect vertical and horizontal edges

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \qquad\qquad \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

Vertical Edges — Horizontal Edges

Kirsch Operator

- Also known as direction masks
- Edge detects in 8 compass directions

$$\begin{matrix} +5 & +5 & +5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{matrix}$$

N

$$\begin{matrix} +5 & +5 & -3 \\ +5 & 0 & -3 \\ -3 & -3 & -3 \end{matrix}$$

NW

$$\begin{matrix} +5 & -3 & -3 \\ +5 & 0 & -3 \\ +5 & -3 & -3 \end{matrix}$$

W

$$\begin{matrix} -3 & -3 & -3 \\ +5 & 0 & -3 \\ +5 & +5 & -3 \end{matrix}$$

SW

$$\begin{matrix} -3 & -3 & +5 \\ -3 & 0 & +5 \\ -3 & -3 & +5 \end{matrix}$$

E

$$\begin{matrix} -3 & +5 & +5 \\ -3 & 0 & +5 \\ -3 & -3 & -3 \end{matrix}$$

NE

$$\begin{matrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ +5 & +5 & +5 \end{matrix}$$

S

$$\begin{matrix} -3 & -3 & +5 \\ -3 & 0 & +5 \\ -3 & +5 & +5 \end{matrix}$$

SE

Robinson Operator

- Also known as direction masks
- Edge detects in 8 compass directions

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

N

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

NW

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

W

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

SW

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

E

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

NE

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

S

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

SE

Laplacian Operator

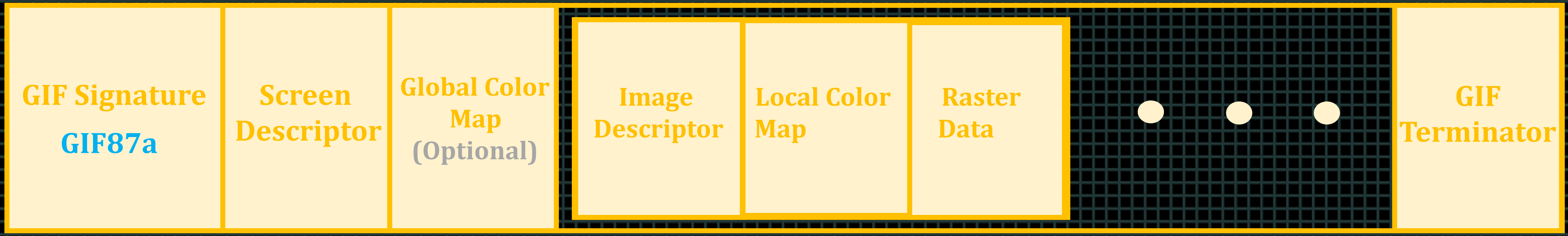
- 2^{nd} order derivative mask

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \qquad\qquad \begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

Negative

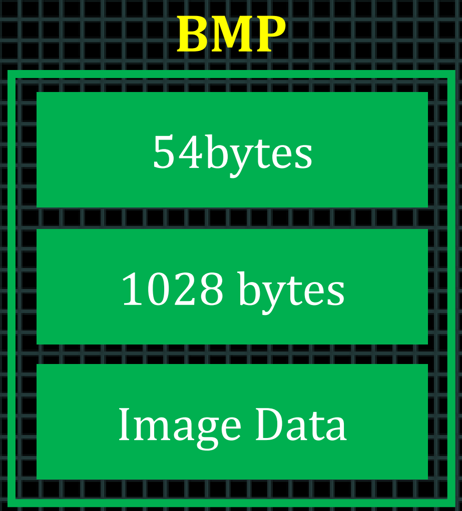
Positive

GIF


GIF Signature
GIF87a
Screen Descriptor
Global Color Map
(Optional)
Image Descriptor
Local Color Map
Raster Data
GIF Terminator

BMP

The bitmap Image



BMP (Windows) Header Format

offset	size	description
0	2	signature, must be 4D42 hex
2	4	size of BMP file in bytes (unreliable)
6	2	reserved, must be zero
8	2	reserved, must be zero
10	4	offset to start of image data in bytes
14	4	size of BITMAPINFOHEADER structure, must be 40
18	4	image width in pixels
22	4	image height in pixels
26	2	number of planes in the image, must be 1
28	2	number of bits per pixel (1, 4, 8, or 24)
30	4	compression type (0=none, 1=RLE-8, 2=RLE-4)
34	4	size of image data in bytes (including padding)
38	4	horizontal resolution in pixels per meter (unreliable)
42	4	vertical resolution in pixels per meter (unreliable)
46	4	number of colors in image, or zero
50	4	number of important colors, or zero

Filtering in the Frequency Domain

Simple Image Operations

Warping and Morphing

Basic Texture Operations

Manipulating Shapes

Edge Detection

Image Restoration

Morphological Processing

Image Compression and Coding

Feature Extraction and Representation

Feature Extraction and Representation

Image Classification

Image Data Basics

Signal Statistics and Noise

Signal :

How one parameter

Relates to another