Intensity Transformations and Spatial Filtering

柯正雯

Image Enhancement

Input Image → **Enhancement Techniques** → "Better" Image, suitable for human or machine

Spatial Domain

Manipulate pixels directly in an image

Frequency Domain

Modify the Fourier Transform of an image

Intensity Transformations

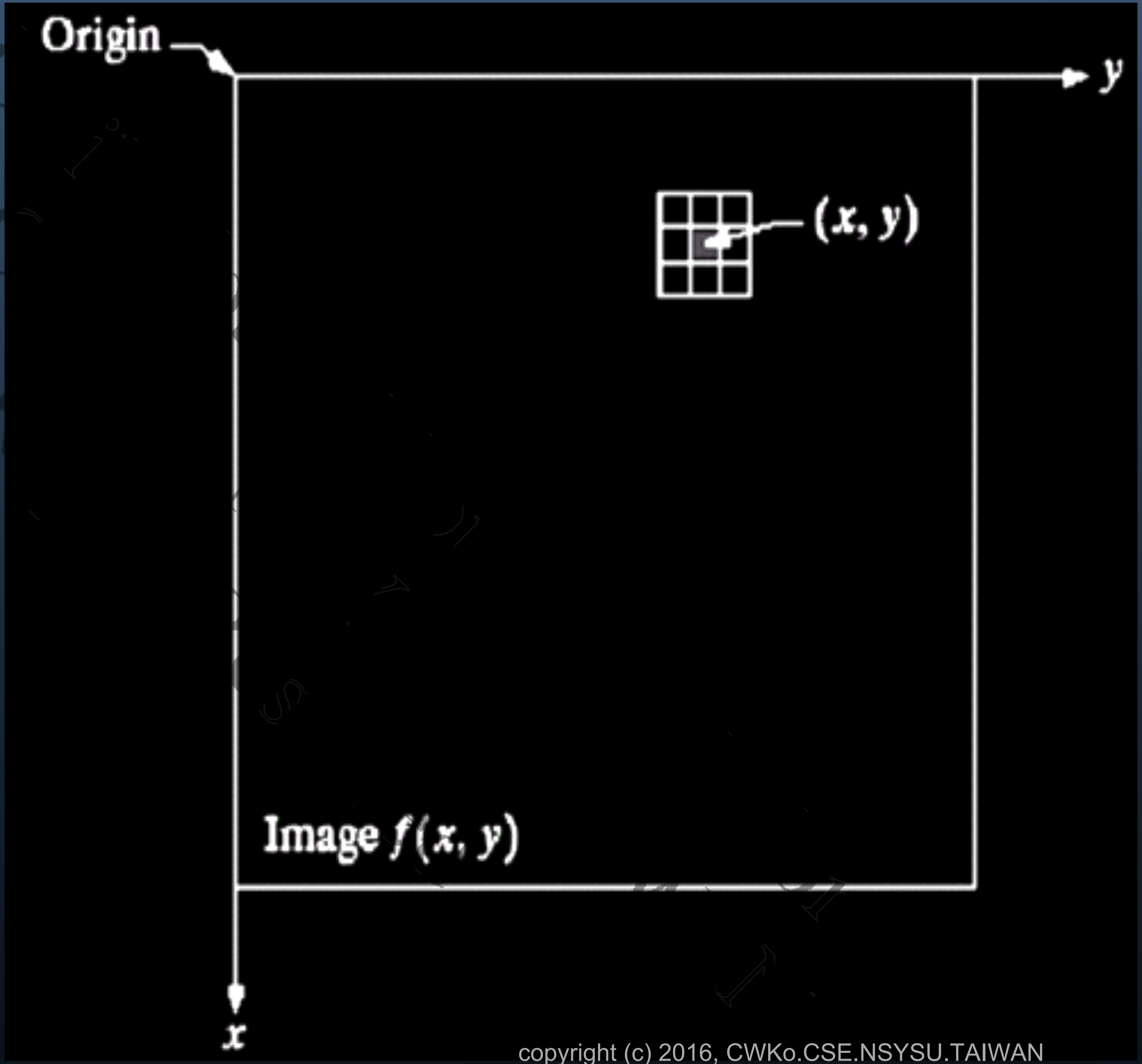
Intensity Transformations

f(x,y)

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

T: gray-level transformation function

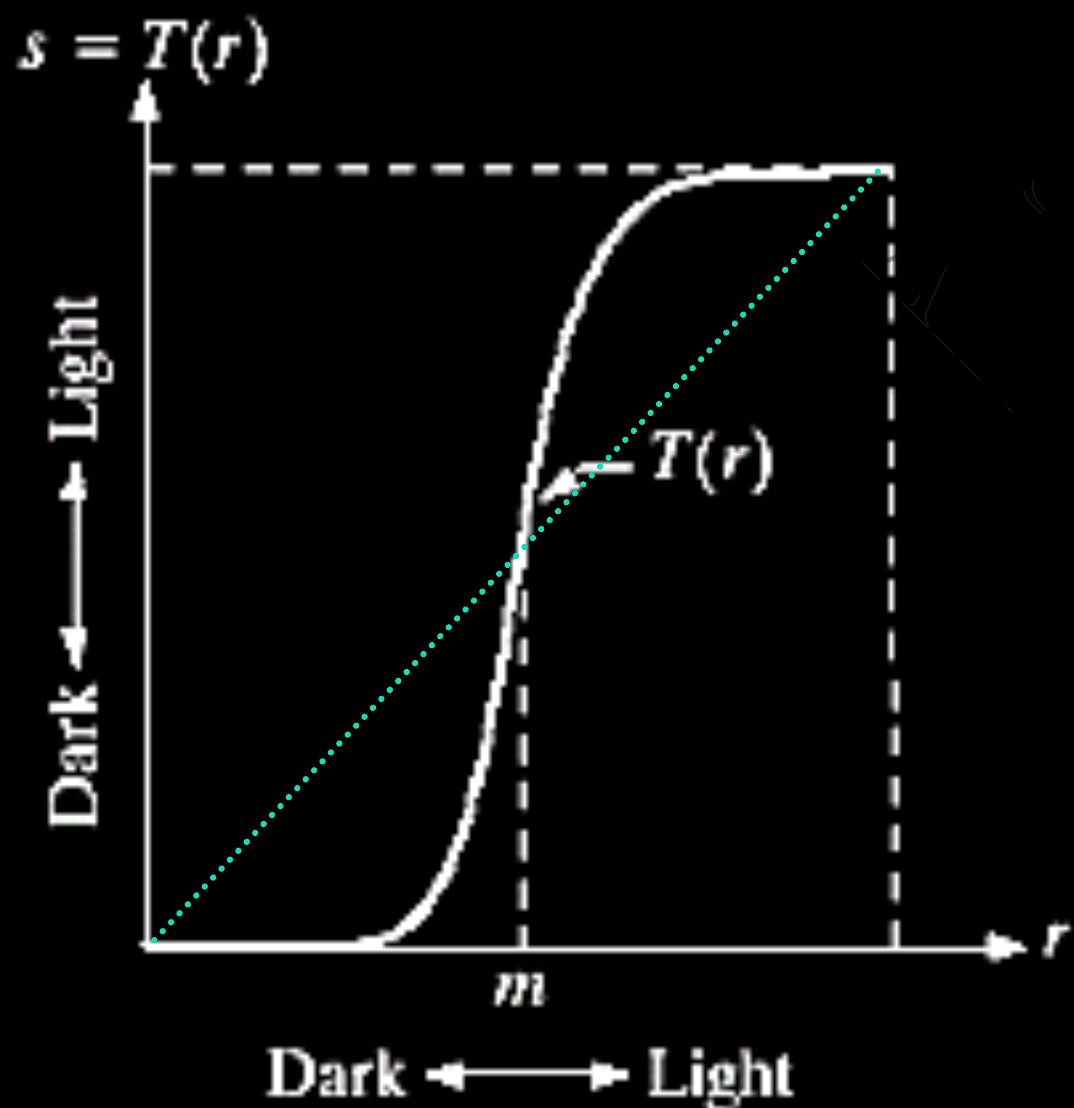
s=T(r)



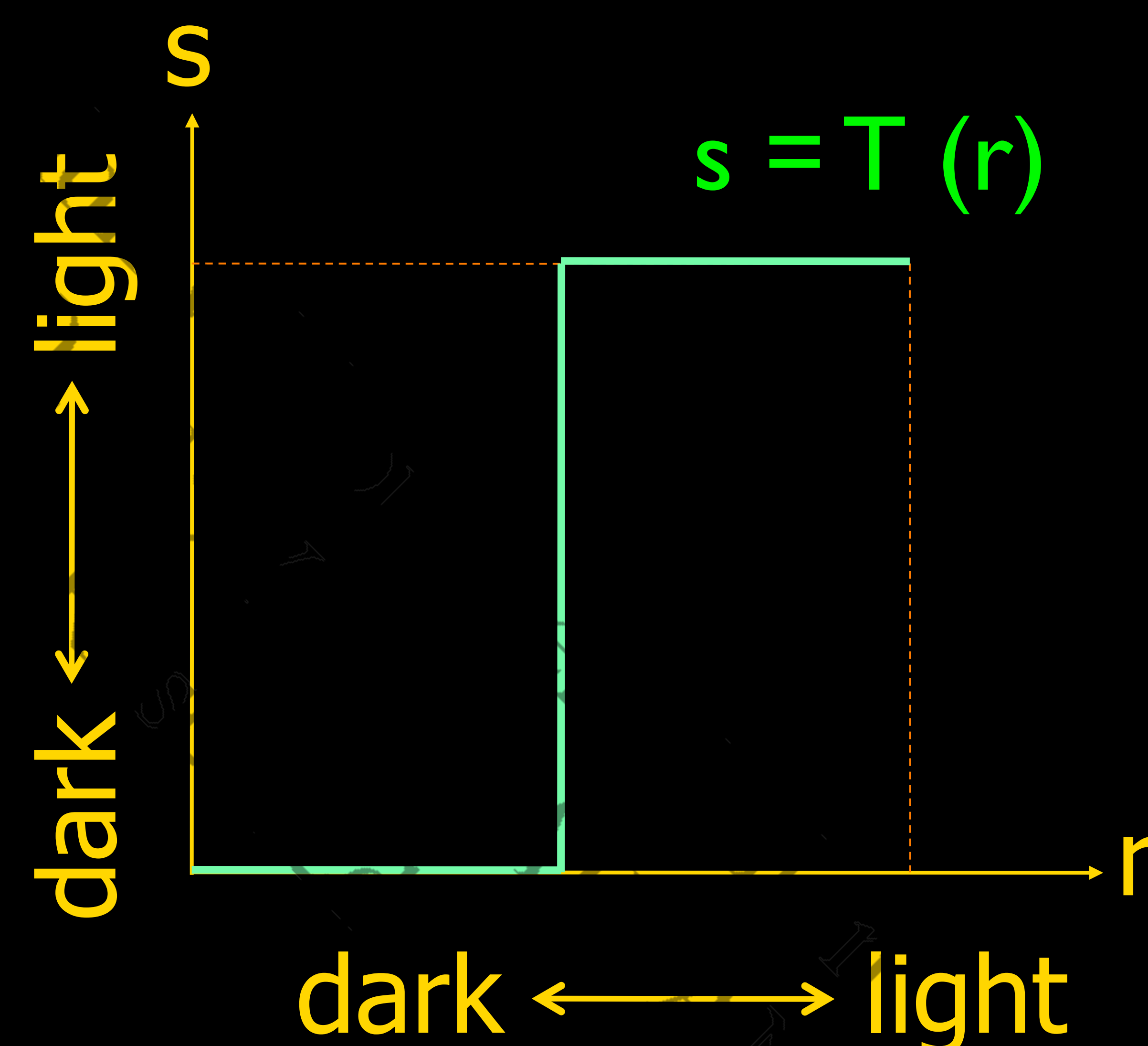
Intensity Transformation Functions

Gray-Level Transformation

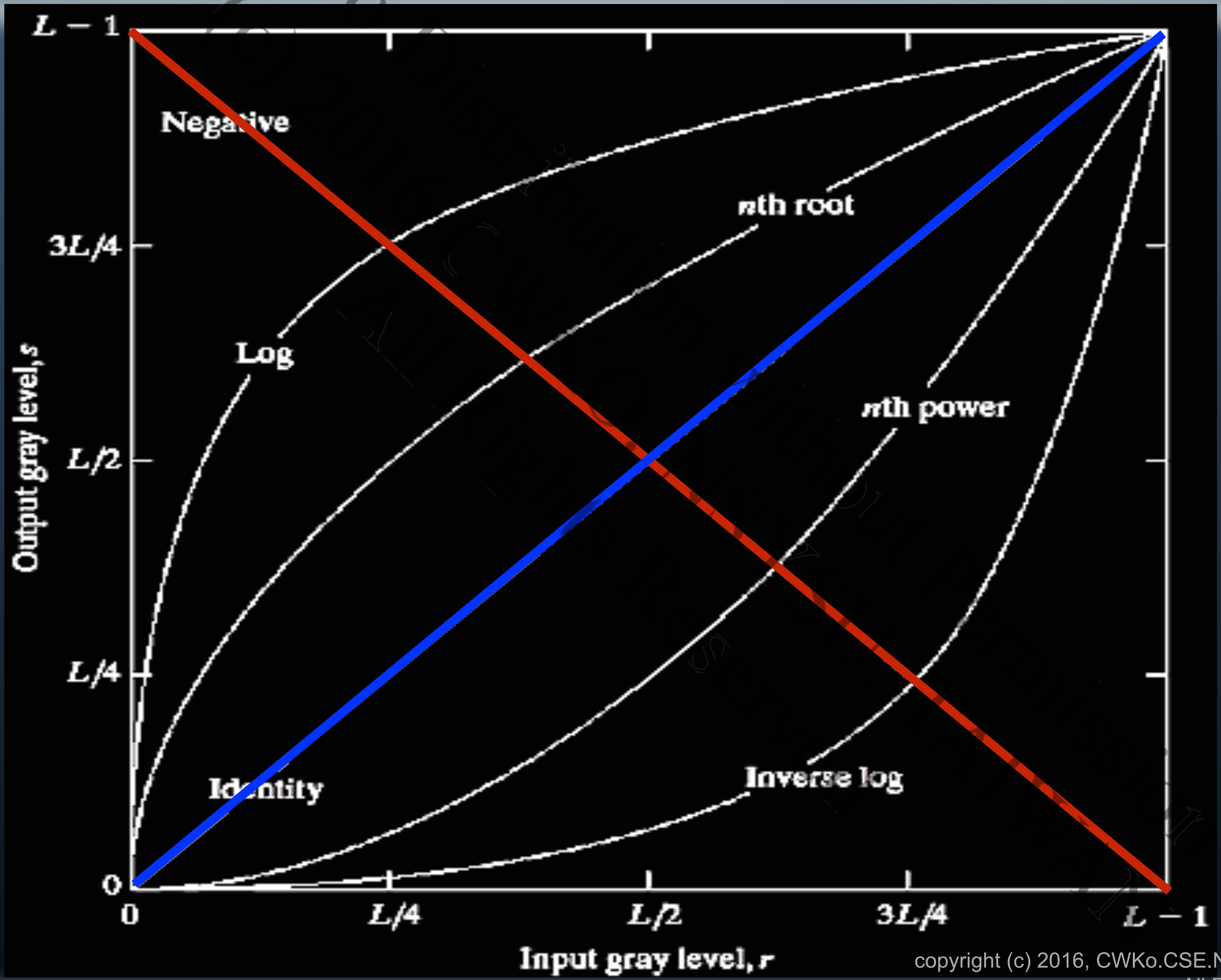
Example of Contrast Stretching



Thresholding

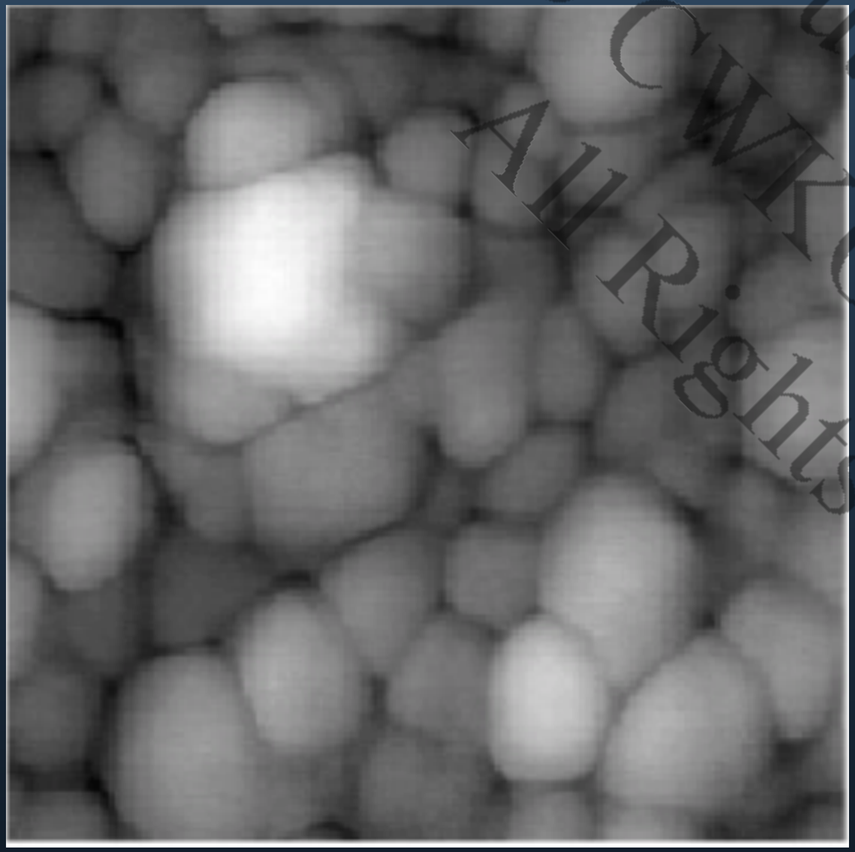


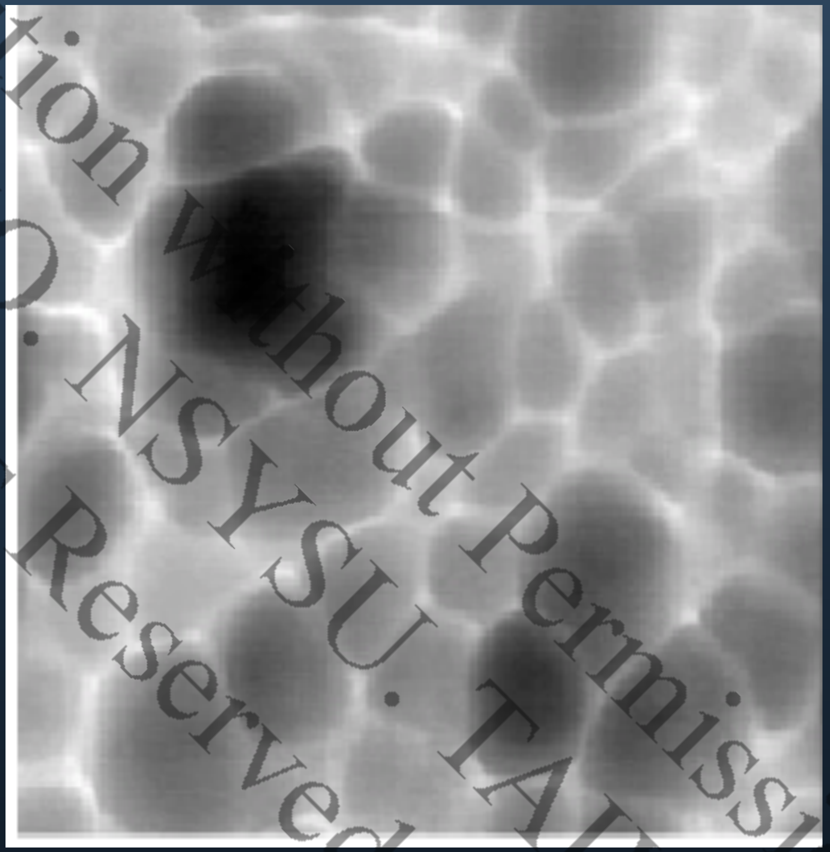
Gray Level Transformations



Negative Image

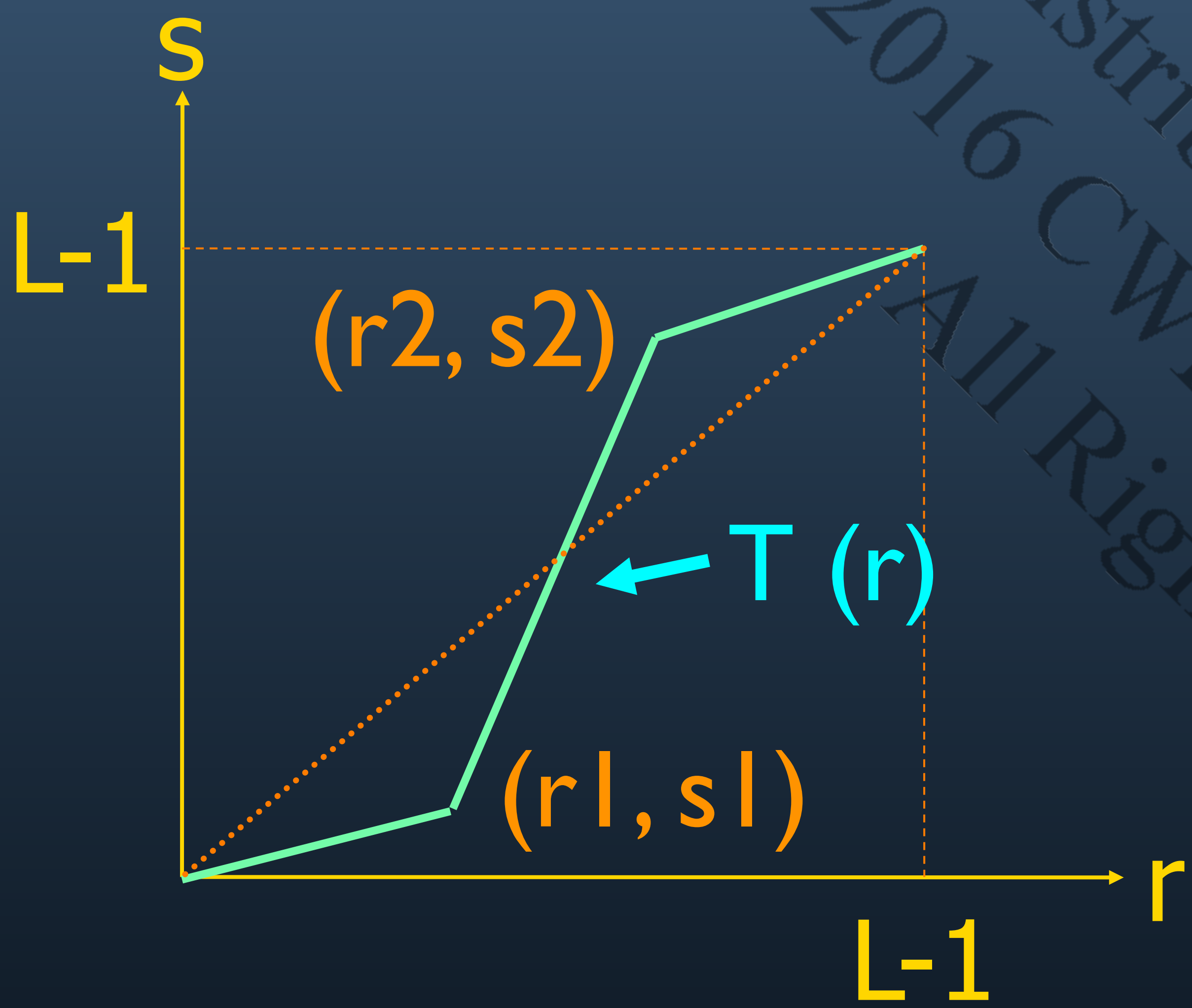
- Example:





g(x,y)=255-f(x,y)

Contrast Stretching



r1 = s1
r2 = s2

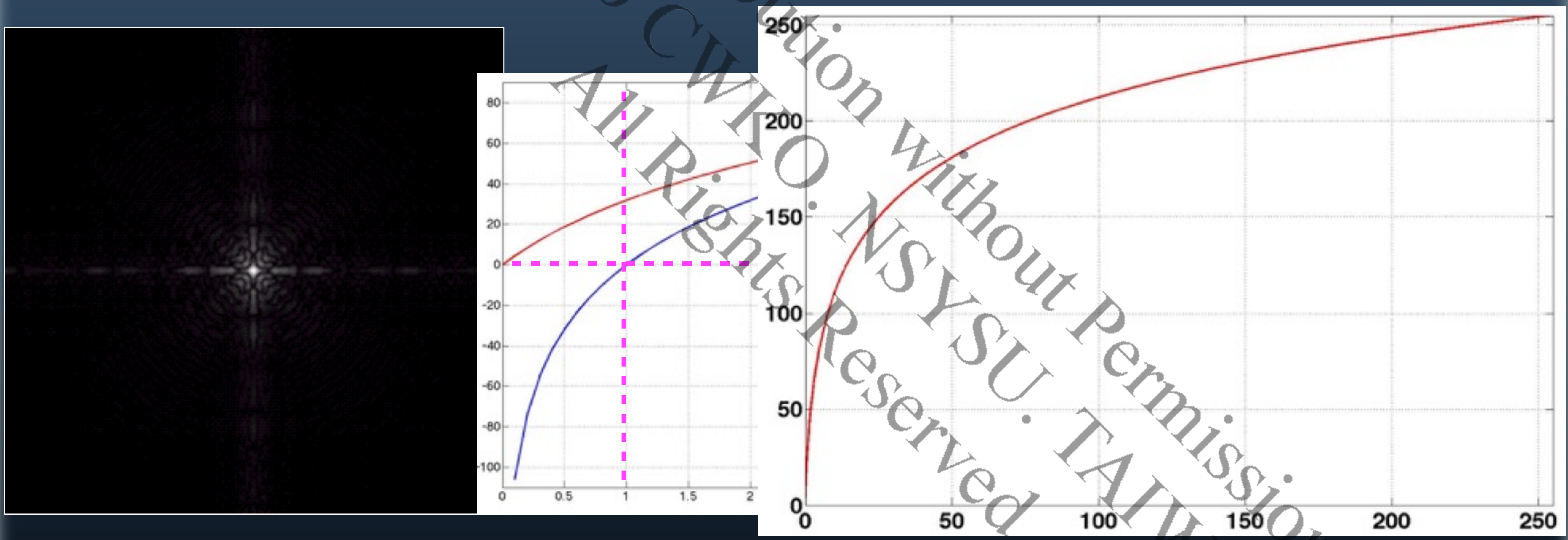
No change

r1 = r2
s1 = 0
s2 = L-1

thresholding

Log Transformation

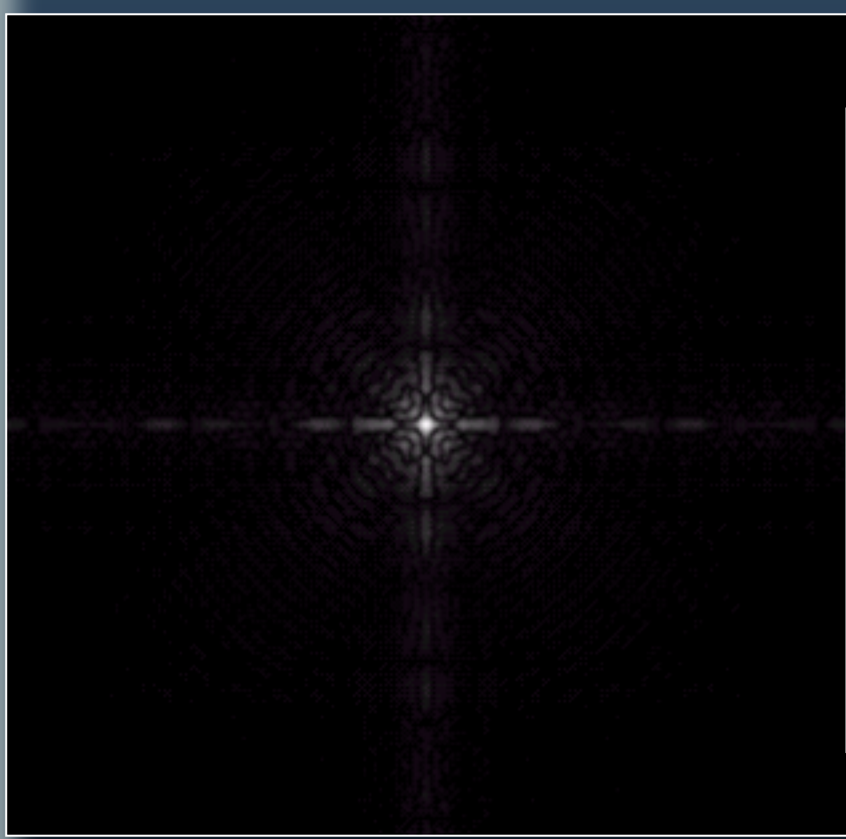
- $s = c \log(1 + |r|)$, c : scaling factor

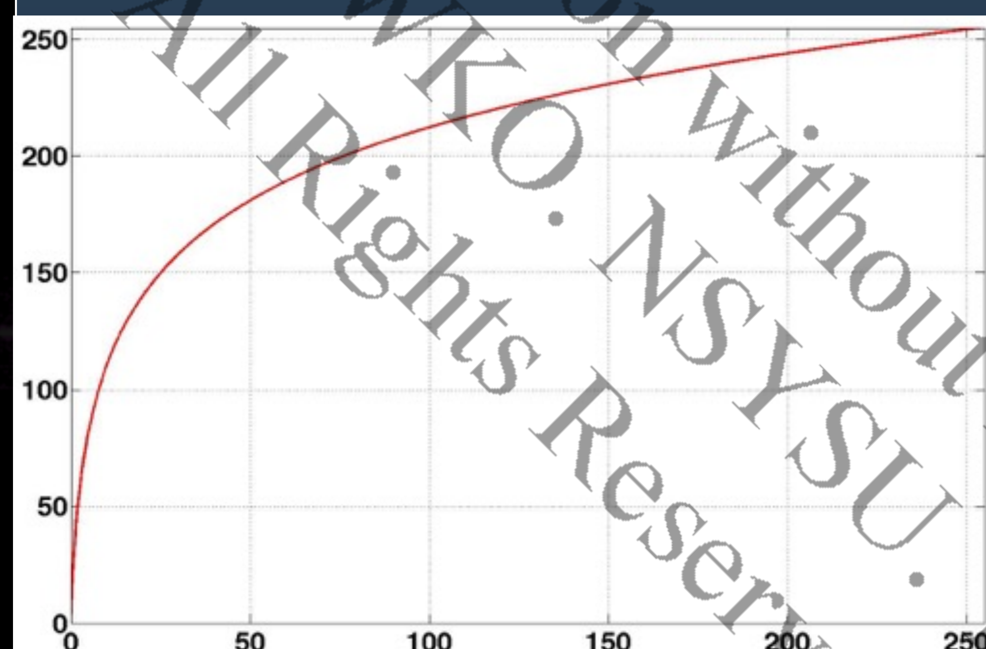


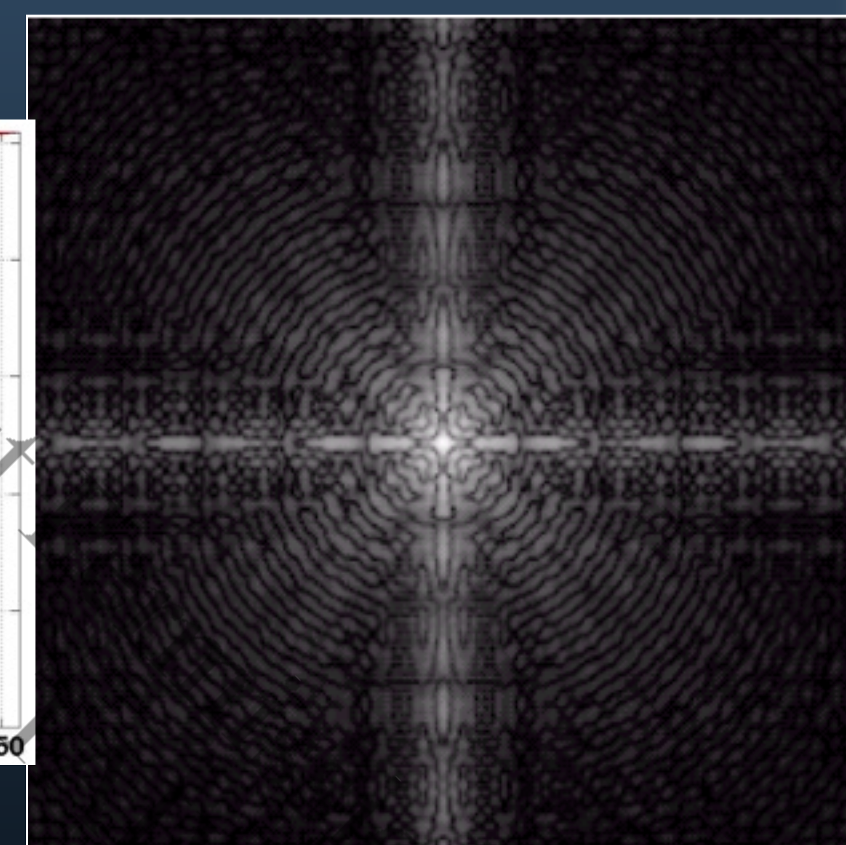
Example: display the Fourier spectrum

Log Transformation

- $s = c \log(1+|r|)$, c : scaling factor







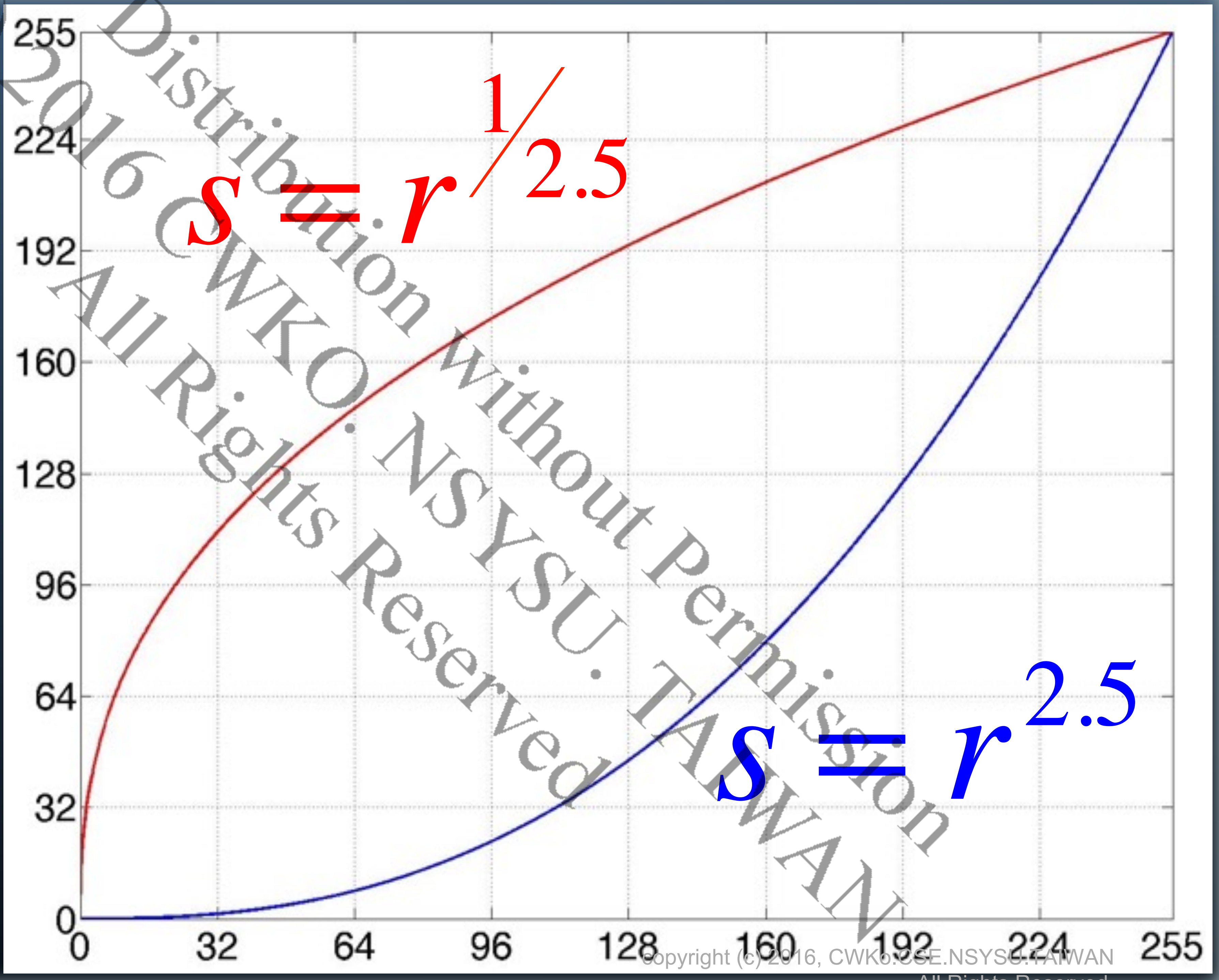
Example: display the Fourier spectrum

Power-Law Transformation

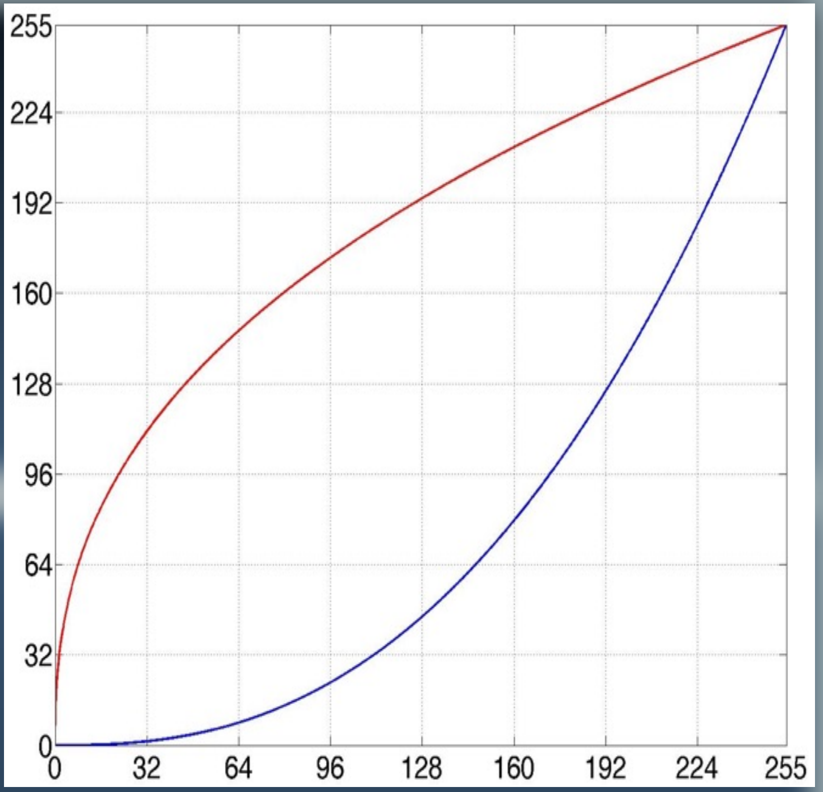
- $s = c\, r^{\gamma}$
- c, γ : positive constants
- CRT (Cathode Ray Tube) :
  intensity-to-voltage response follows a power function
  (typical value of gamma in the range 1.5-2.5)

Gamma Correction

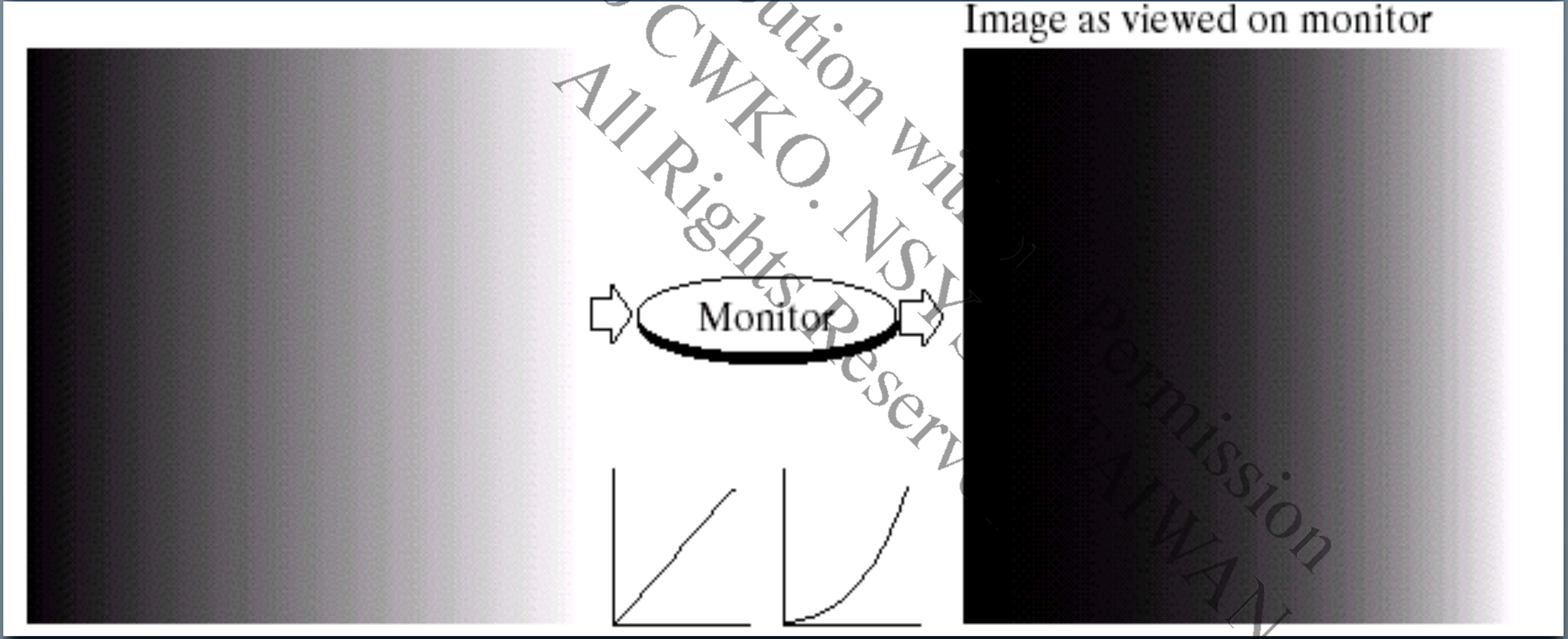
- $s = c\, r^{\gamma}$



Gamma Correction

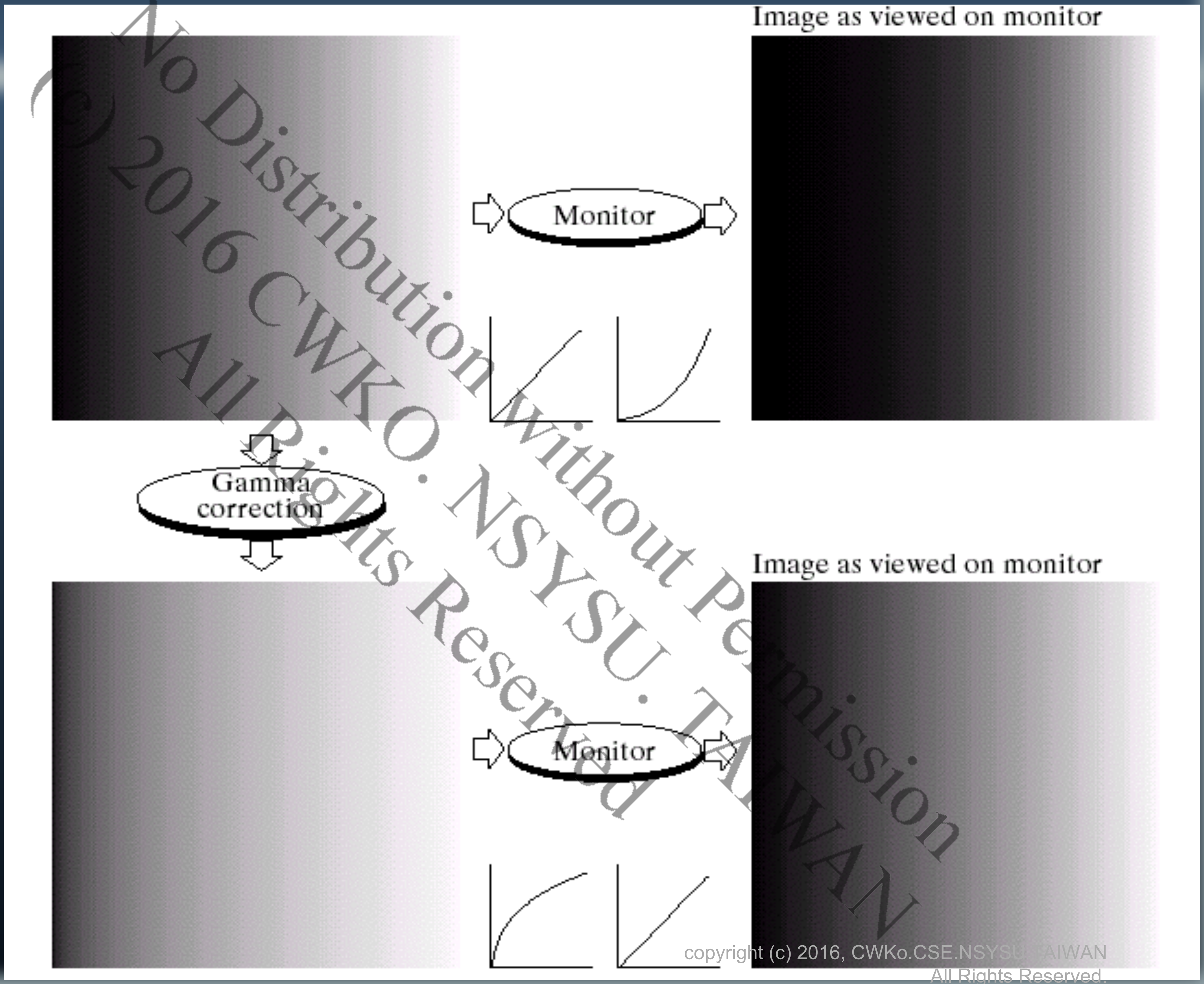


- without correction



Gamma Correction

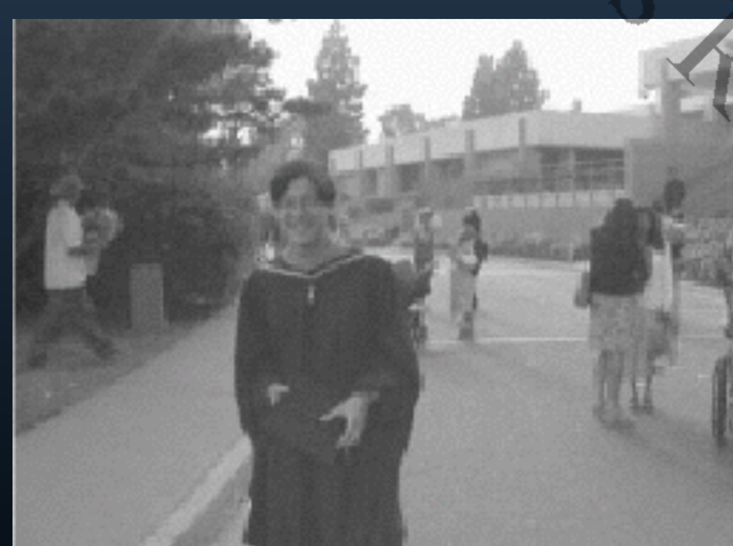
$$s = r^{1/2.5}$$



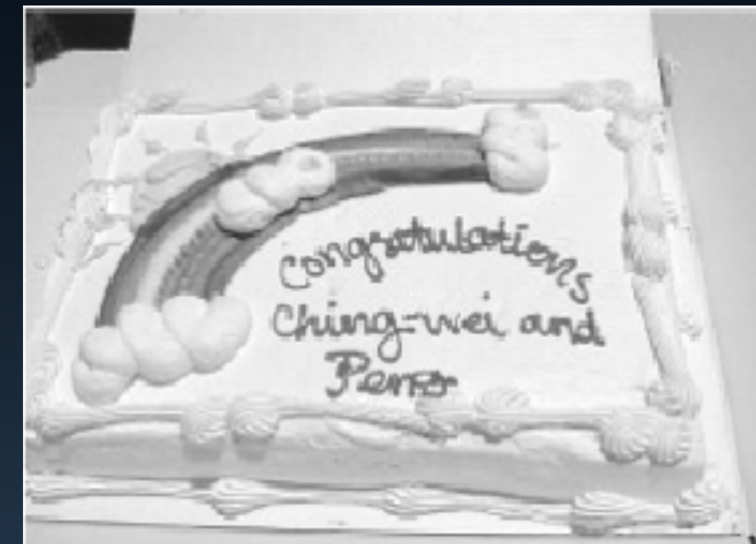
$$s = cr^{\gamma}$$

1

0.7



0.4

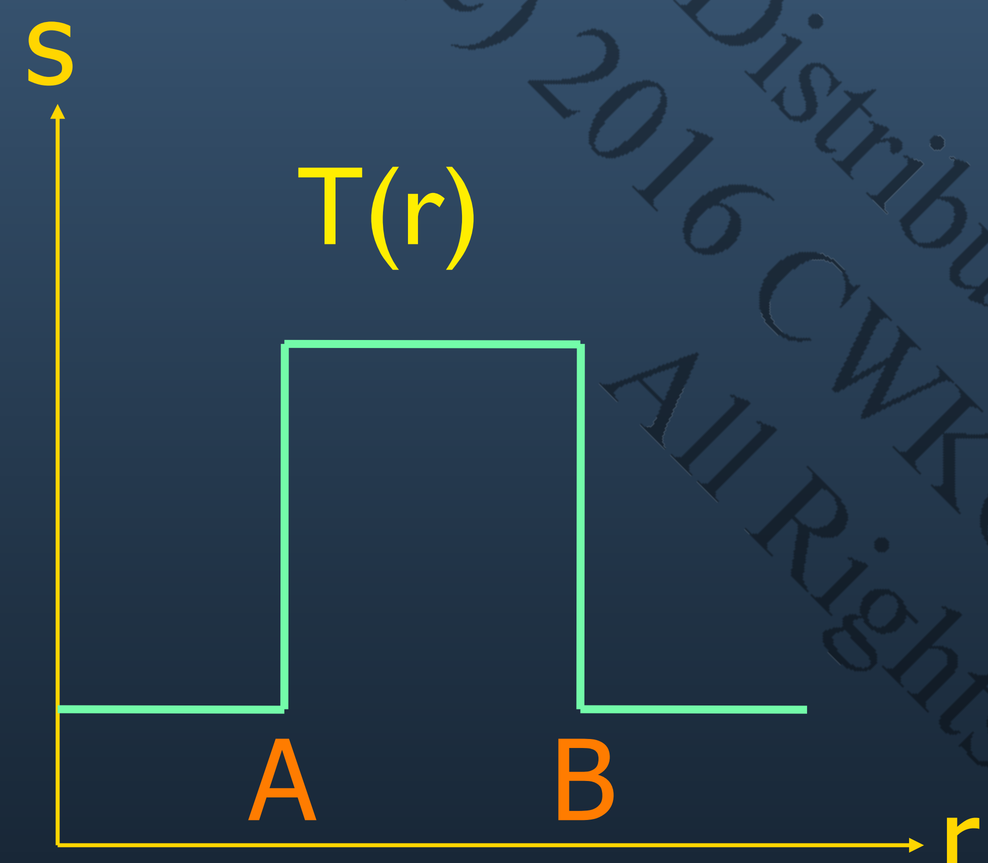


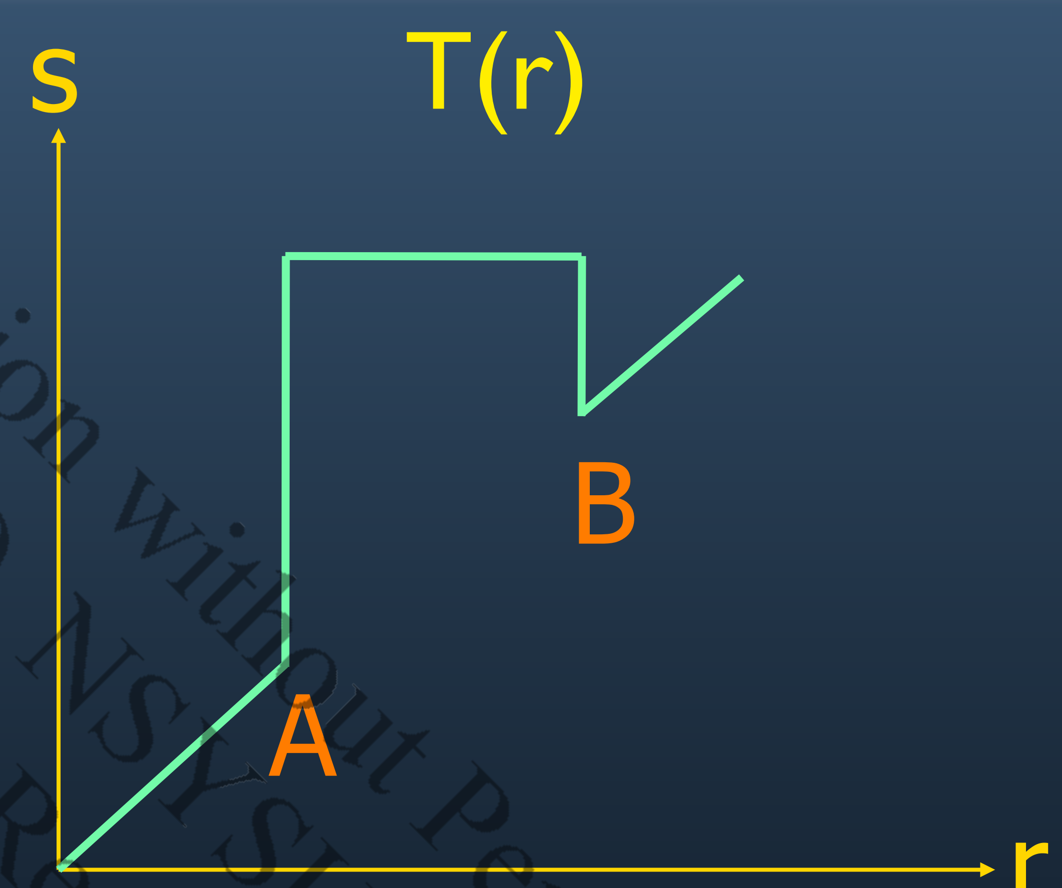
1

2

5

Gray Level Slicing





Highlights only the range [A, B]

Preserves other intensities

Bit Plane Slicing

Highlights contributions made by specific bits.

Bit Plane Slicing

One pixel of gray level 194 $\Longrightarrow$ 1 1 0 0 0 0 1 0

Example: MSB plane

- T(r) = 0, r in the range [0,127]
- T(r) = 255, r in the range [128, 255]

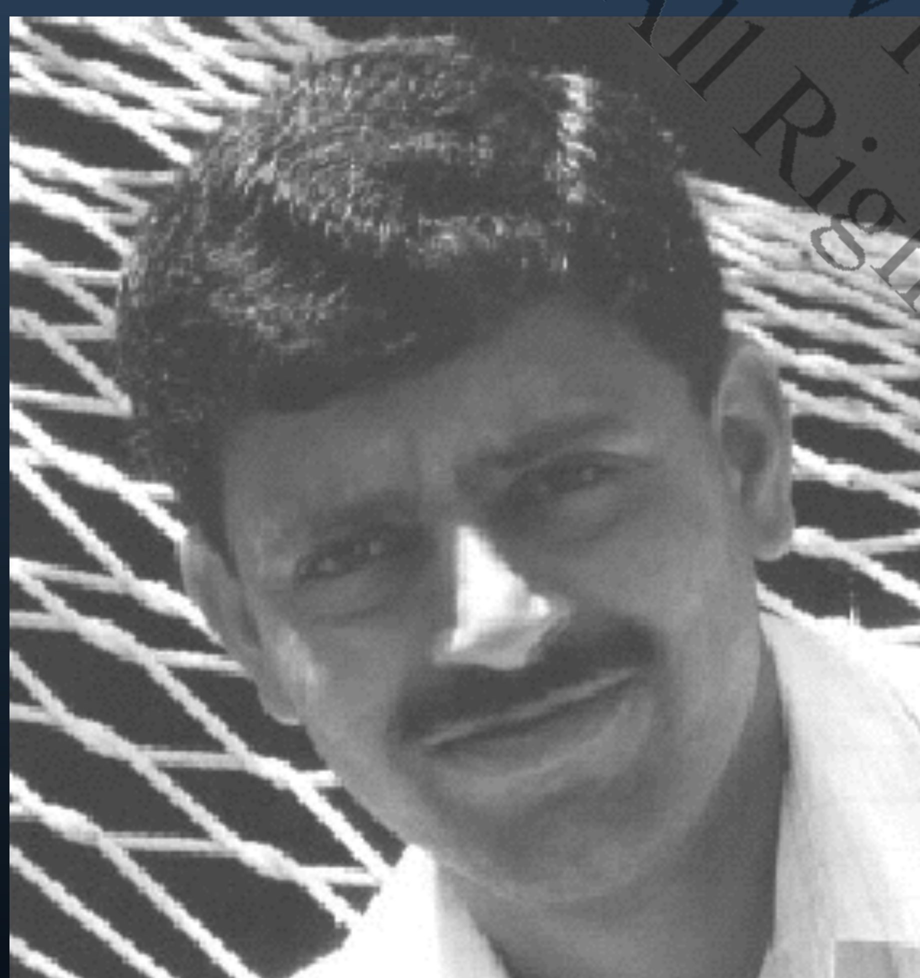
$\boxed{0}$ 1 1 1 1 1 1 1

$\boxed{1}$ 0 0 0 0 0 0 0

Most Significant Bit … Least Significant Bit

Example: MSB plane

- T(r) = 0, r in the range [0,127]
- T(r) = 255, r in the range [128, 255]



Bit-plane 7

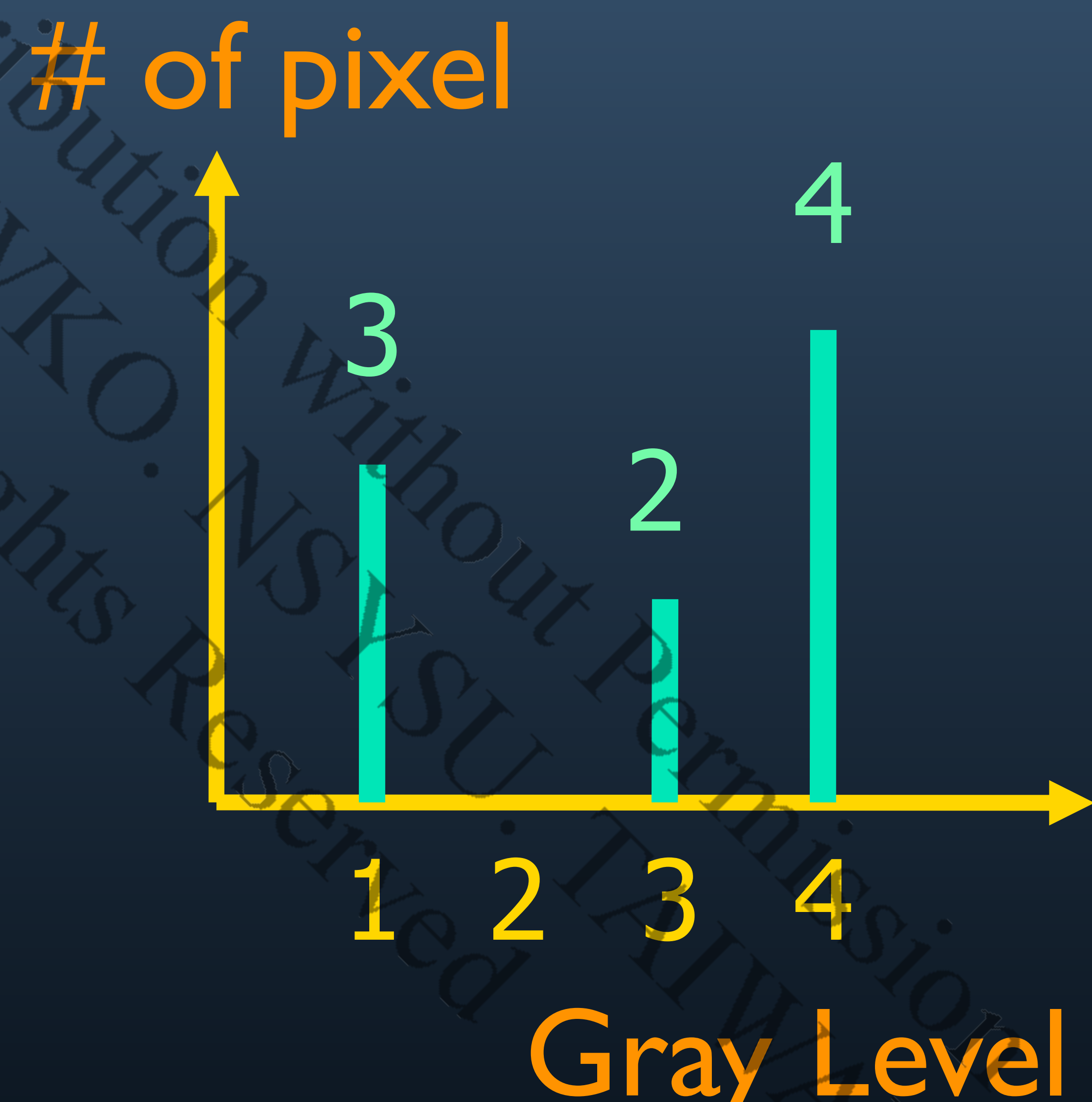
Histogram Processing

Histogram Processing

- Histogram Equalization
- Histogram Specification / Matching

Histogram

1	4	3
4	1	4
1	3	4



Normalized Histogram

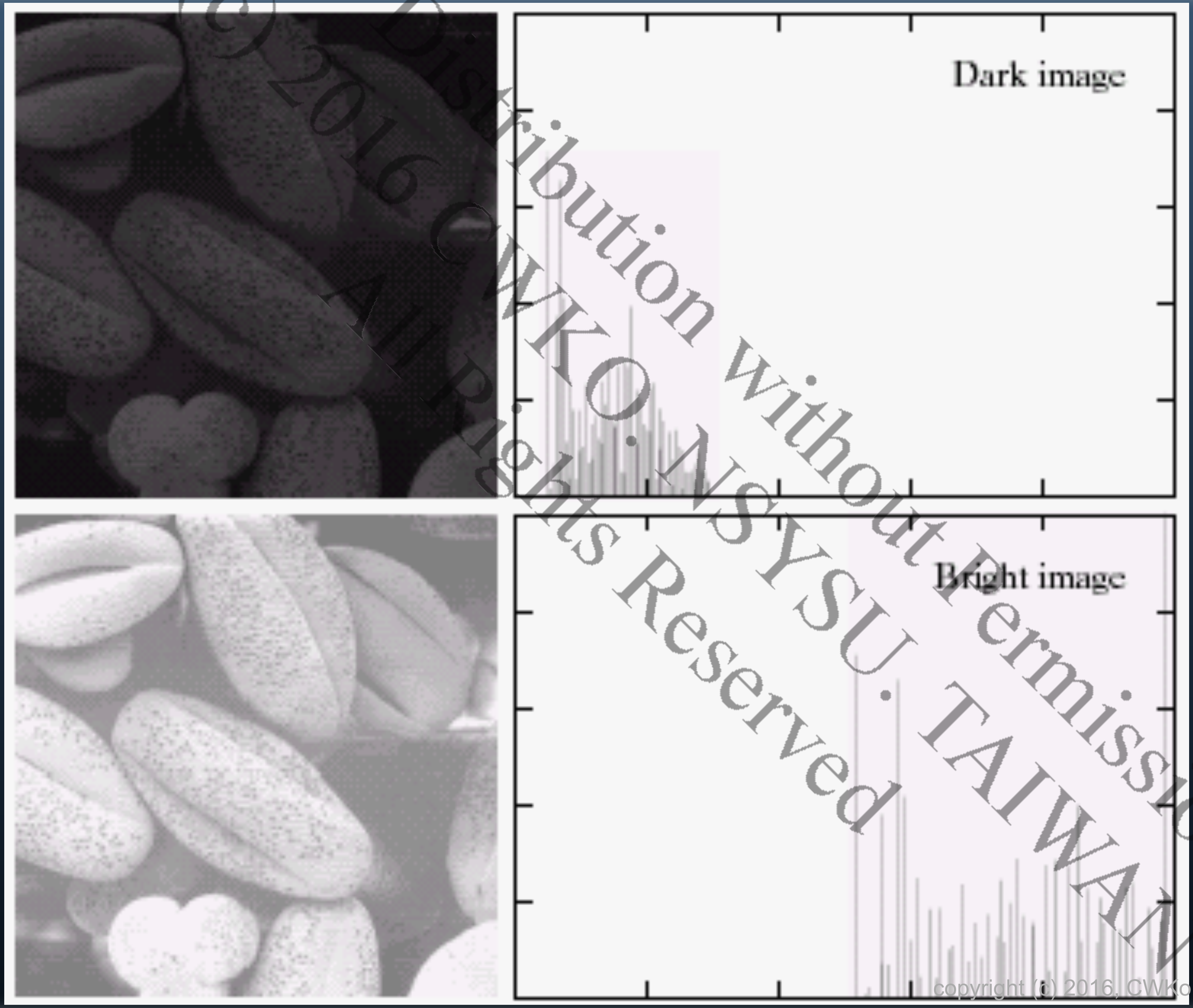
$$p(r_k) = \frac{n_k}{n}$$

$$r_k \in \{0,1,2,3,\ldots,L-1\}$$

n_k : # of pixel with gray level r_k

n : total # of pixel in the image

Histograms



Histogram Equalization

We are interested in obtaining a transformation function T( ) which transforms an arbitrary p.d.f. to an uniform distribution.

Histogram Equalization

r : Input gray level $\in [0, 1]$

s : Transformed gray level $\in [0, 1]$

$$\boxed{s = T(r)}$$

T : Transformation function

Histogram Equalization

(a) $T(r)$ is single valued and monotonically increasing in $0 \leq r \leq 1$

(b) $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$

- Inverse transformation: $T^{-1}(s) = r$
- $T^{-1}(s)$ also satisfies (a) and (b)
- The gray levels in the image can be viewed as random variables taking values in the range [0,1]

Histogram Equalization

- Let $p_r(r)$: p.d.f. of input level
  $p_s(s)$: p.d.f. of s
- From an elementary probability theory:

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

Histogram Equalization

We are interested in obtaining a transformation function T( ) which transforms an arbitrary p.d.f. to an **uniform** distribution.

Histogram Equalization

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

(Cumulative distribution function of r)

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right] = (L-1)p_r(r)$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = \frac{1}{L-1} \qquad 0 \le s \le L-1$$

Histogram Equalization

$$p_r(r_k) = \frac{n_k}{MN} \qquad k = 0,1,2,\ldots,L-1$$

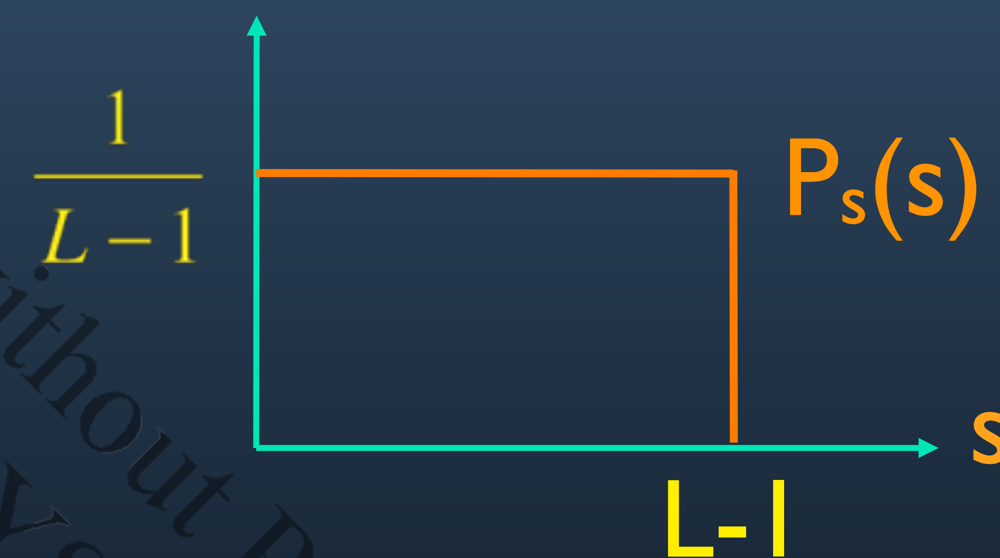
$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0,1,2,\ldots,L-1$$

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

Histogram Equalization

- When gray level is not normalized…



$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0,1,2,\ldots,L-1$$

Equalization -- example

r	0	1	2	3	4	5	6	7
n_k	1	7	21	35	35	21	7	1

128

↓

s	0	1	2	3	4	5	6	7
n_k	16	16	16	16	16	16	16	16

Equalization -- example

$$s_k = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0,1,2,\ldots,L-1$$

r	0	1	2	3	4	5	6	7
n_k	1	7	21	35	35	21	7	1

s_k = 8 x [1 8 29 64 99 120 127 128] / 128
= [0 0 1 4 6 7 7 8]
with r = [0 1 2 3 4 5 6 7]

s	0	1	2	3	4	5	6	7
n_k	8	21	0	0	35	0	35	29

Equalization

- Histogram equalization is very useful for low contrast images
- Images with detail hidden in dark region

Histogram Matching / Specification

- Uniform histogram is not always the best !
- To design the specific histogram

Matching / Specification

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

p_r → Original histogram

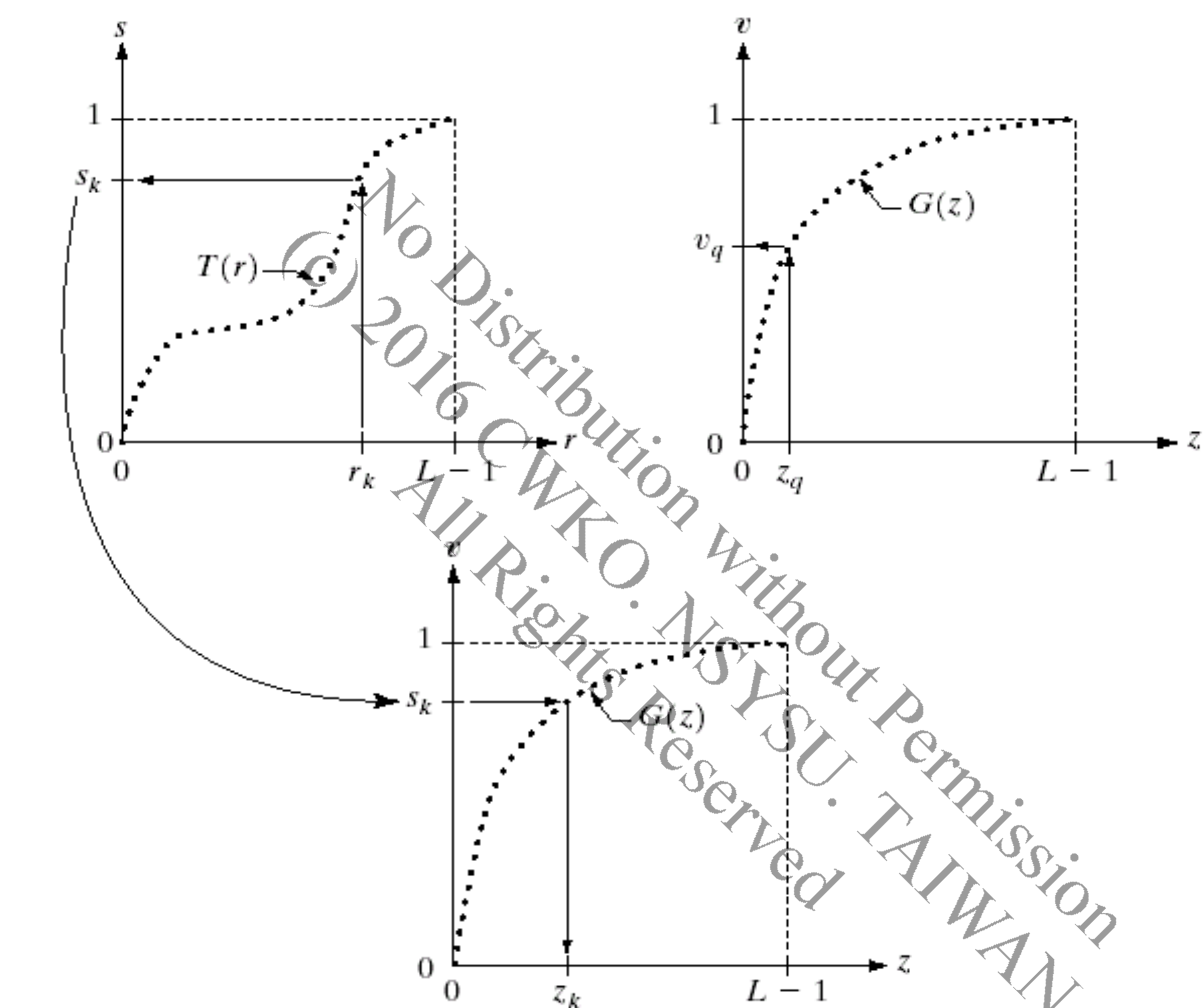
$p_z(z)$ → Desired histogram

Steps:

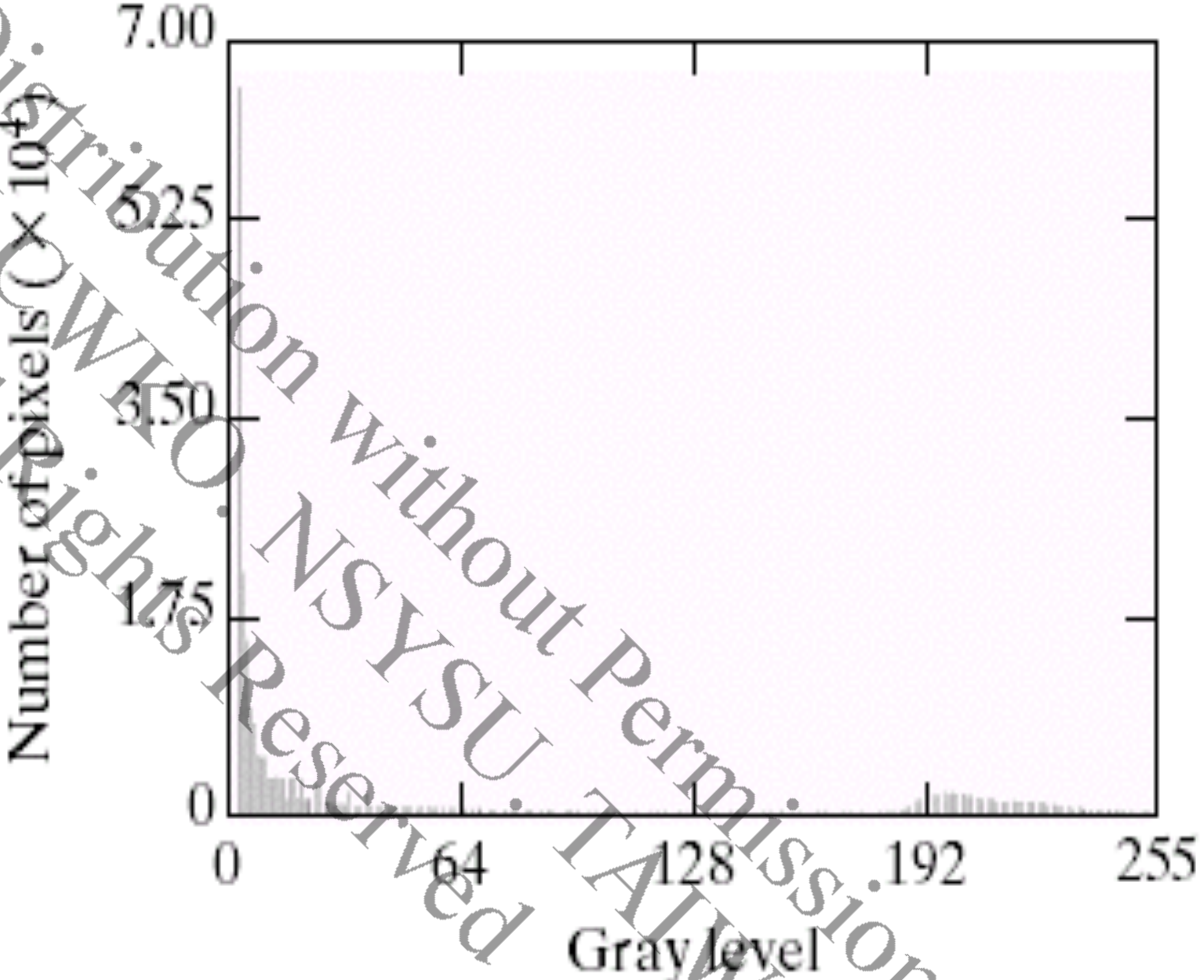
(1) Equalize the levels of original image

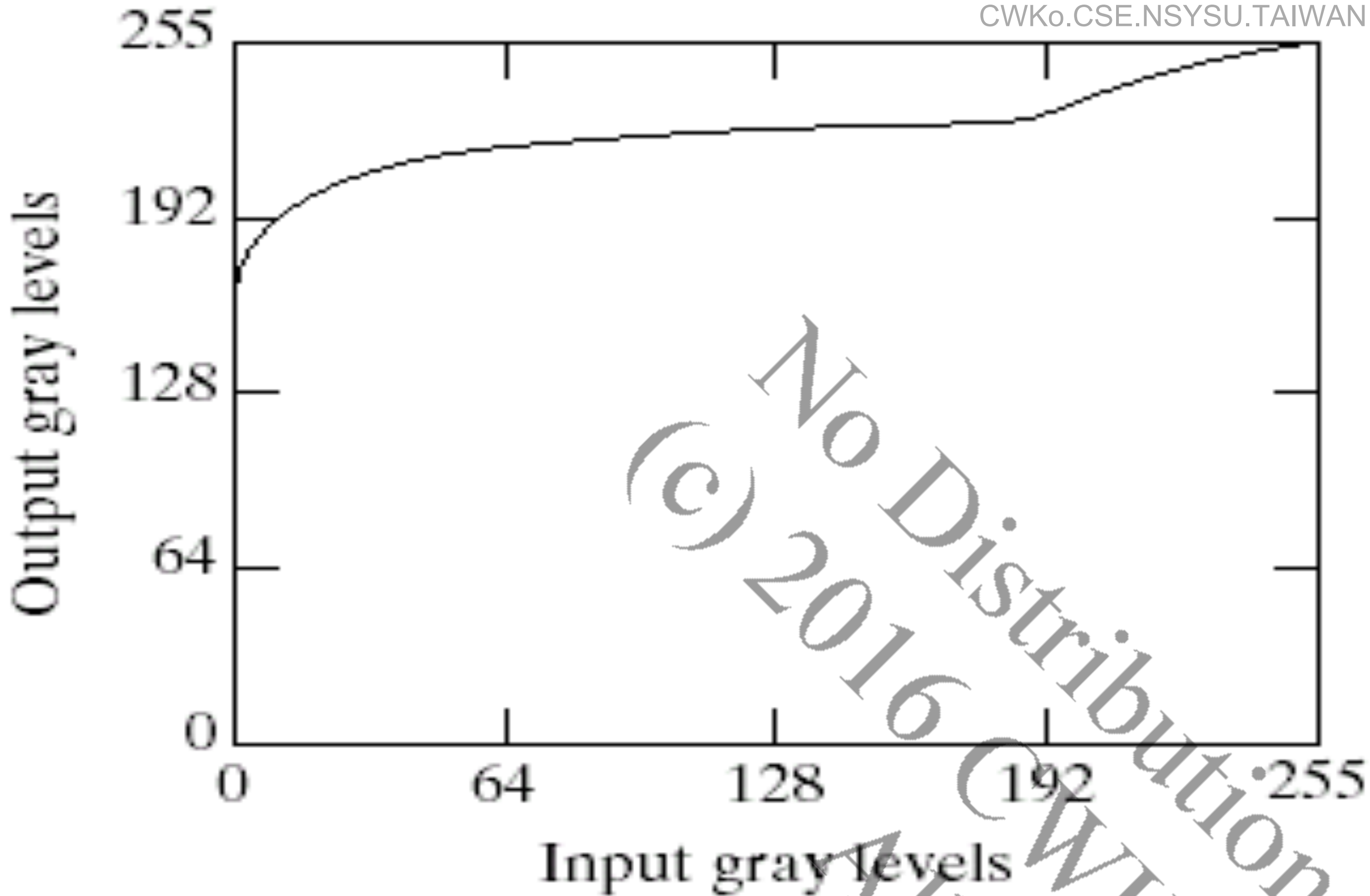
(2) Specify the desired $p_z(z)$ and obtain $G(z)$

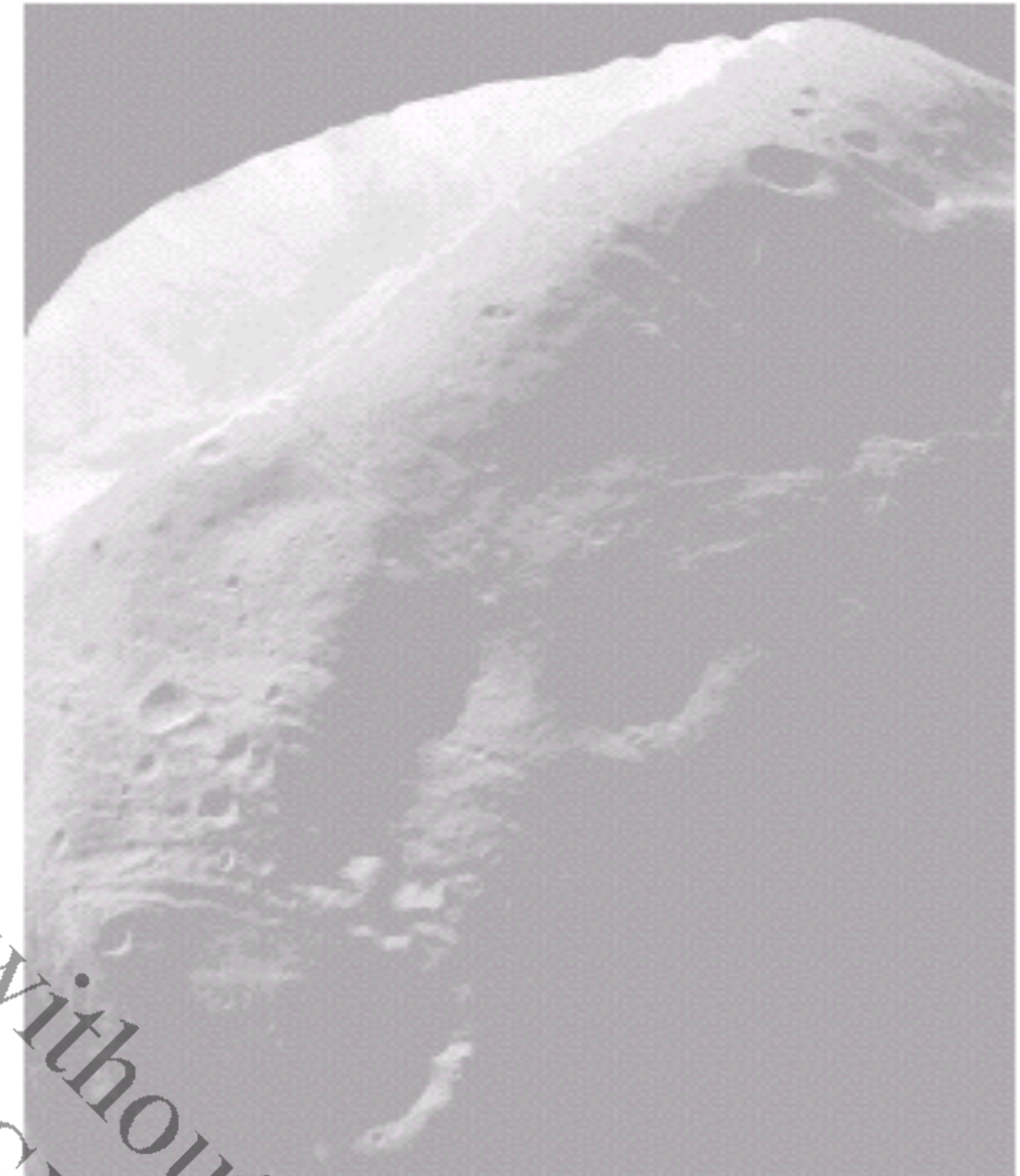
(3) Apply $z = G^{-1}(s)$ to the levels s obtained in step 1

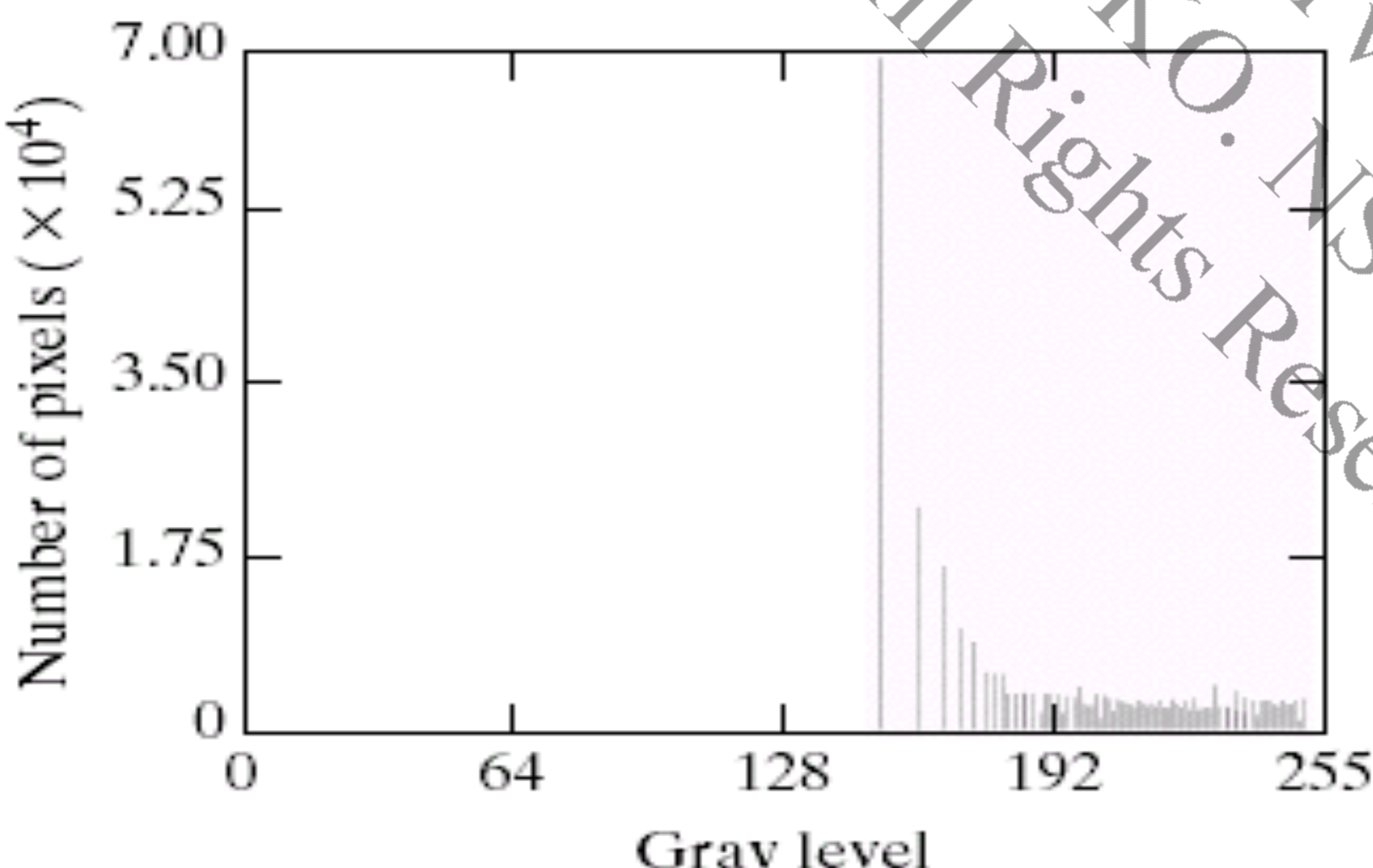


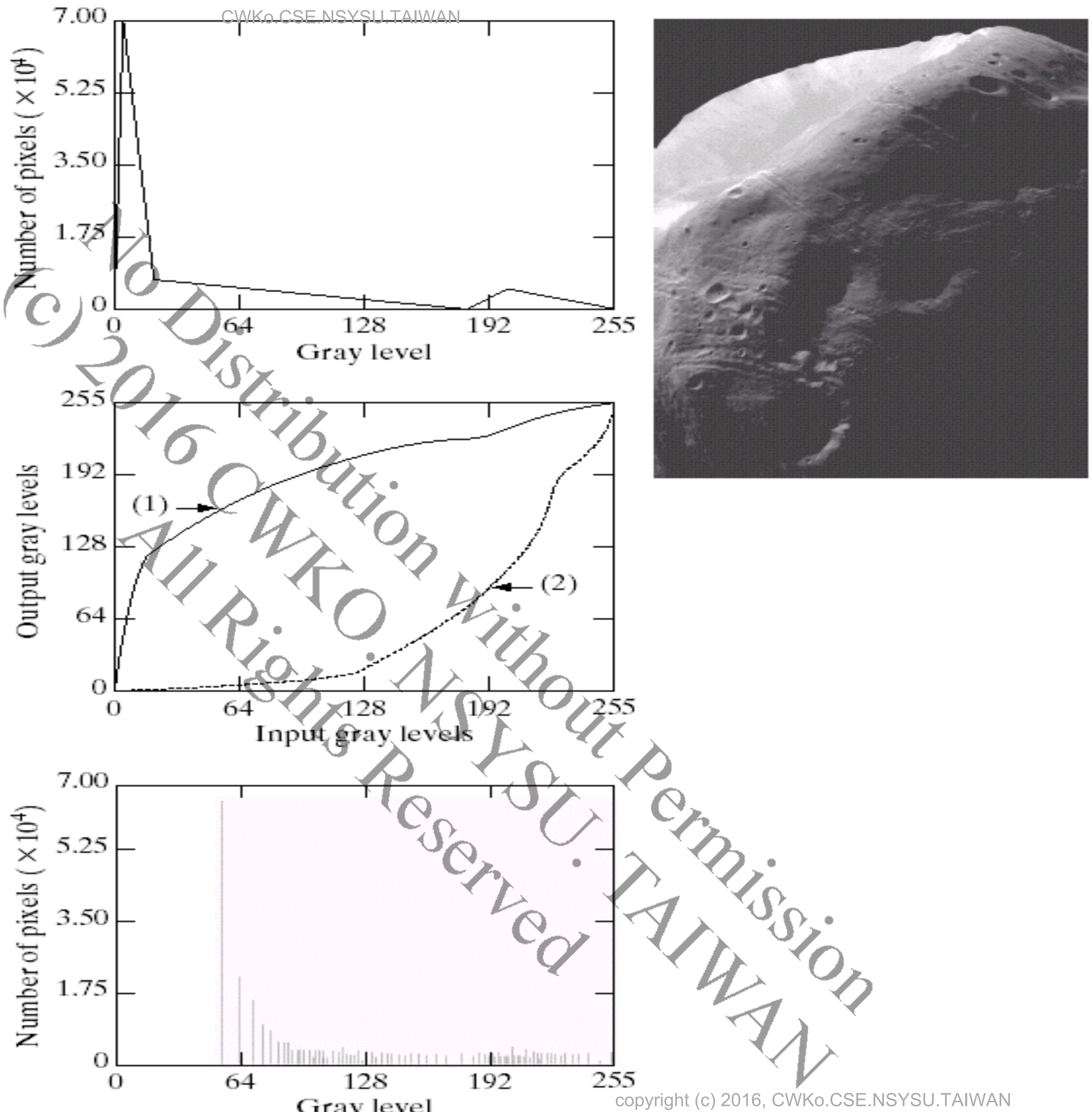
s
1
s_k
$T(r)$
0
0
r_k
$L-1$
r
v
1
$G(z)$
v_q
0
0
z_q
$L-1$
z
v
1
s_k
$G(z)$
0
0
z_k
$L-1$
z


7.00
5.25
3.50
1.75
0
Number of pixels ($\times 10^4$)
0
64
128
192
255
Gray level


255
192
128
64
0
Output gray levels
0
64
128
192
255
Input gray levels




7.00
5.25
3.50
1.75
0
Number of pixels (× 10^4)
0
64
128
192
255
Gray level



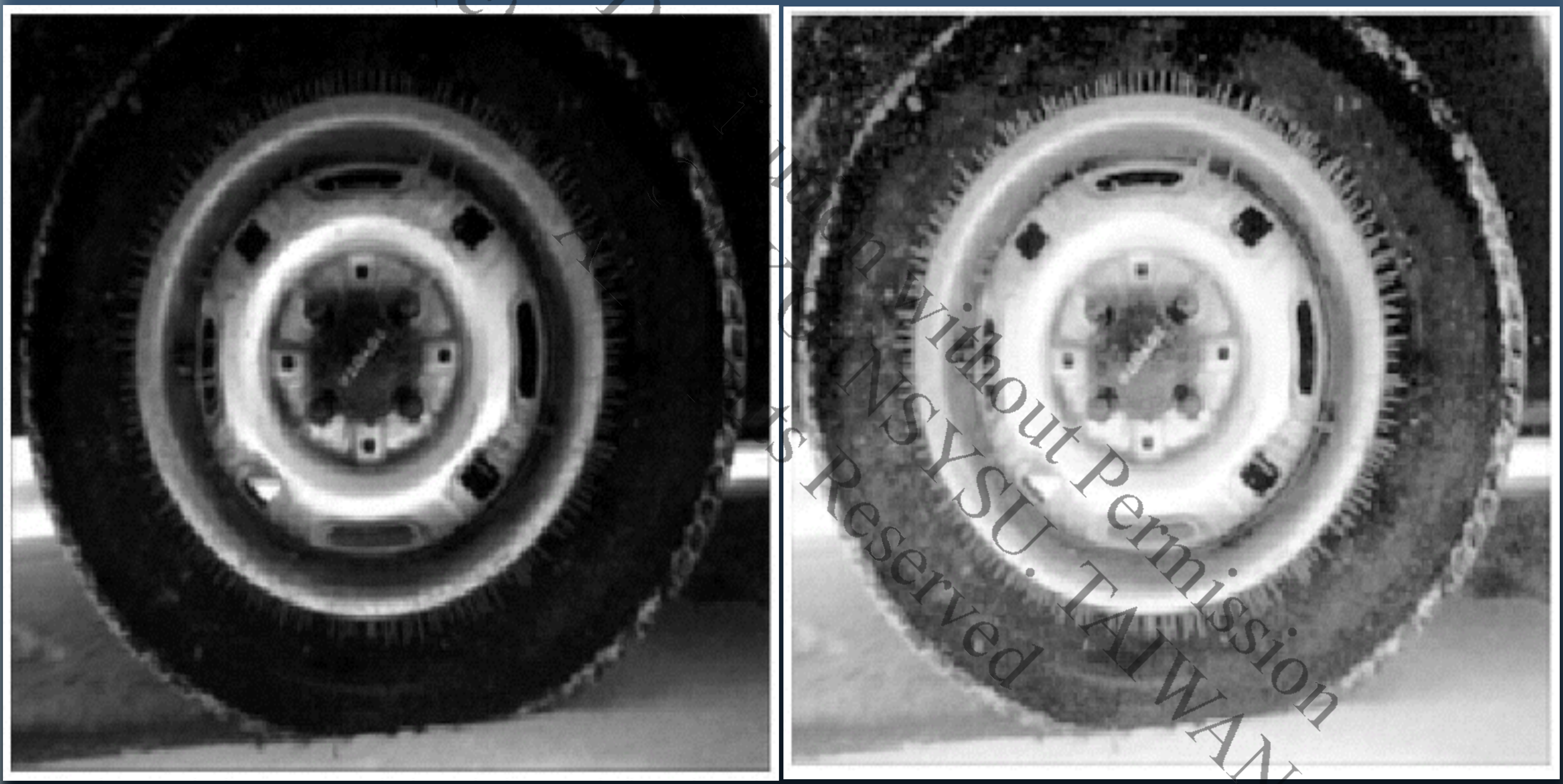
a	c
b	
d	

FIGURE 3.22
(a) Specified histogram.
(b) Curve (1) is from Eq. (3.3-14), using the histogram in (a); curve (2) was obtained using the iterative procedure in Eq. (3.3-17).
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

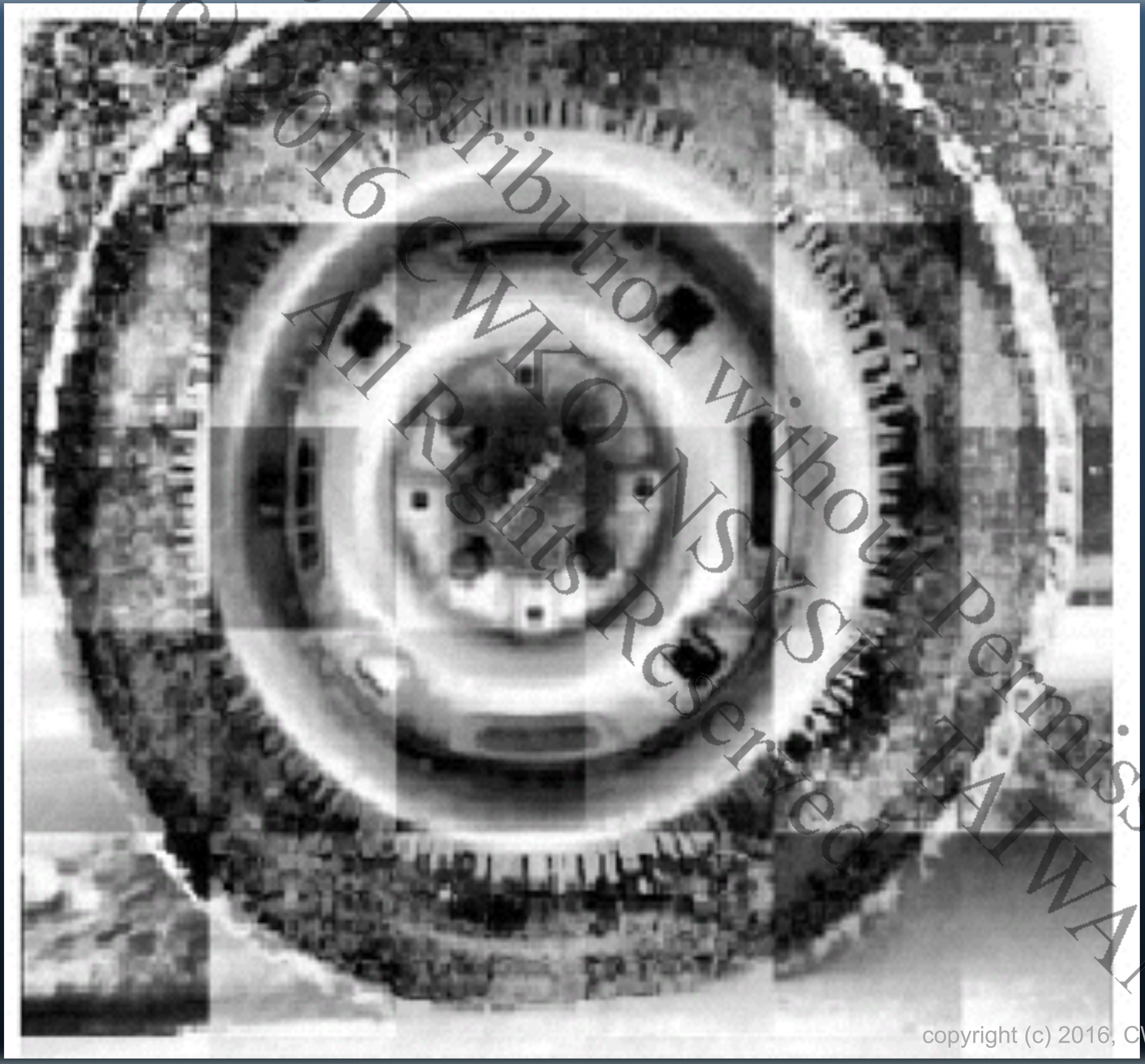
Image Arithmetic Operation

Local Enhancement, Subtraction & Average

Local Enhancement



Local Enhancement



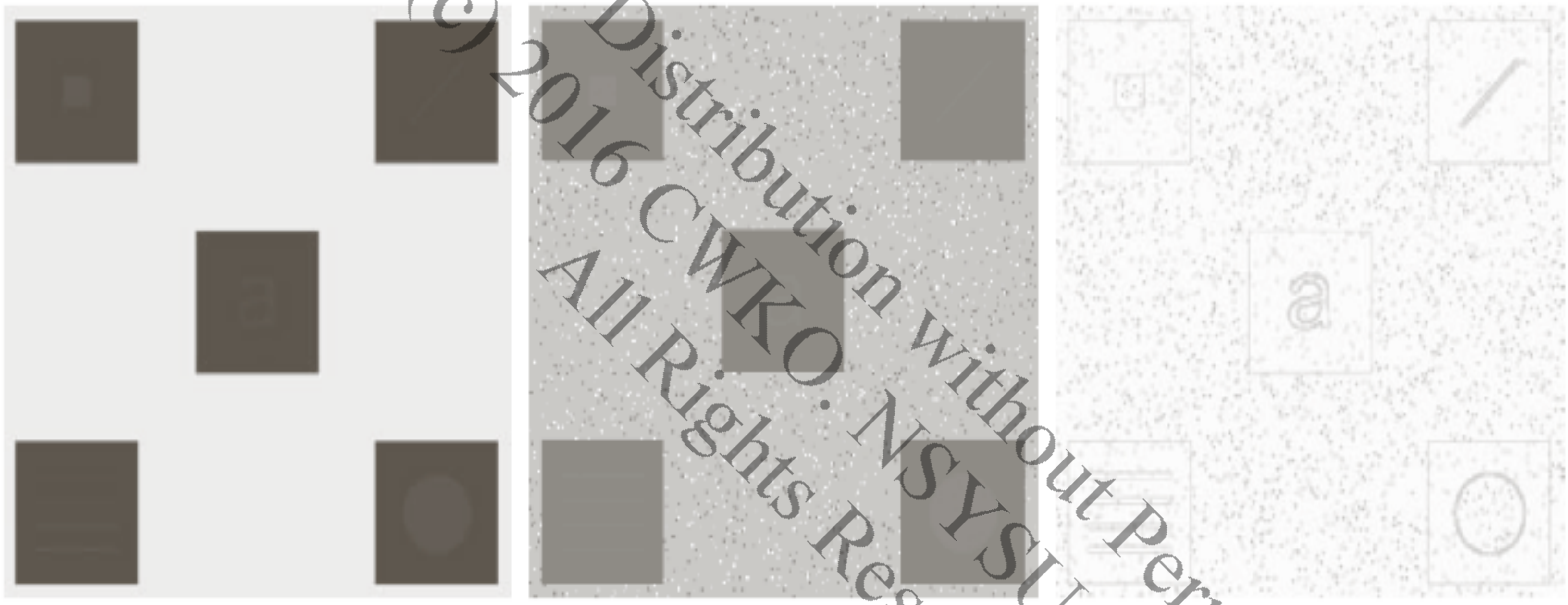


FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3 × 3.

Local Contrast Enhancement

$$g(x,y) = A(x,y)[f(x,y) - m(x,y)] + m(x,y)$$

$$A(x,y) = kM / \sigma(x,y) \quad , 0 < k < 1$$

M : Global Mean

$m(x,y)$, $\sigma(x,y)$: local Mean and standard dev.

Areas with low contrast

→ Larger gain A (x,y)

Local Contrast Enhancement Example

Image Subtraction

- Application in medical imaging –"mask mode radiography"
- H(x,y) is the mask, e.g., an X-ray image of part of a body
  f(x,y) : incoming image after injecting a contrast medium

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

Image Subtraction

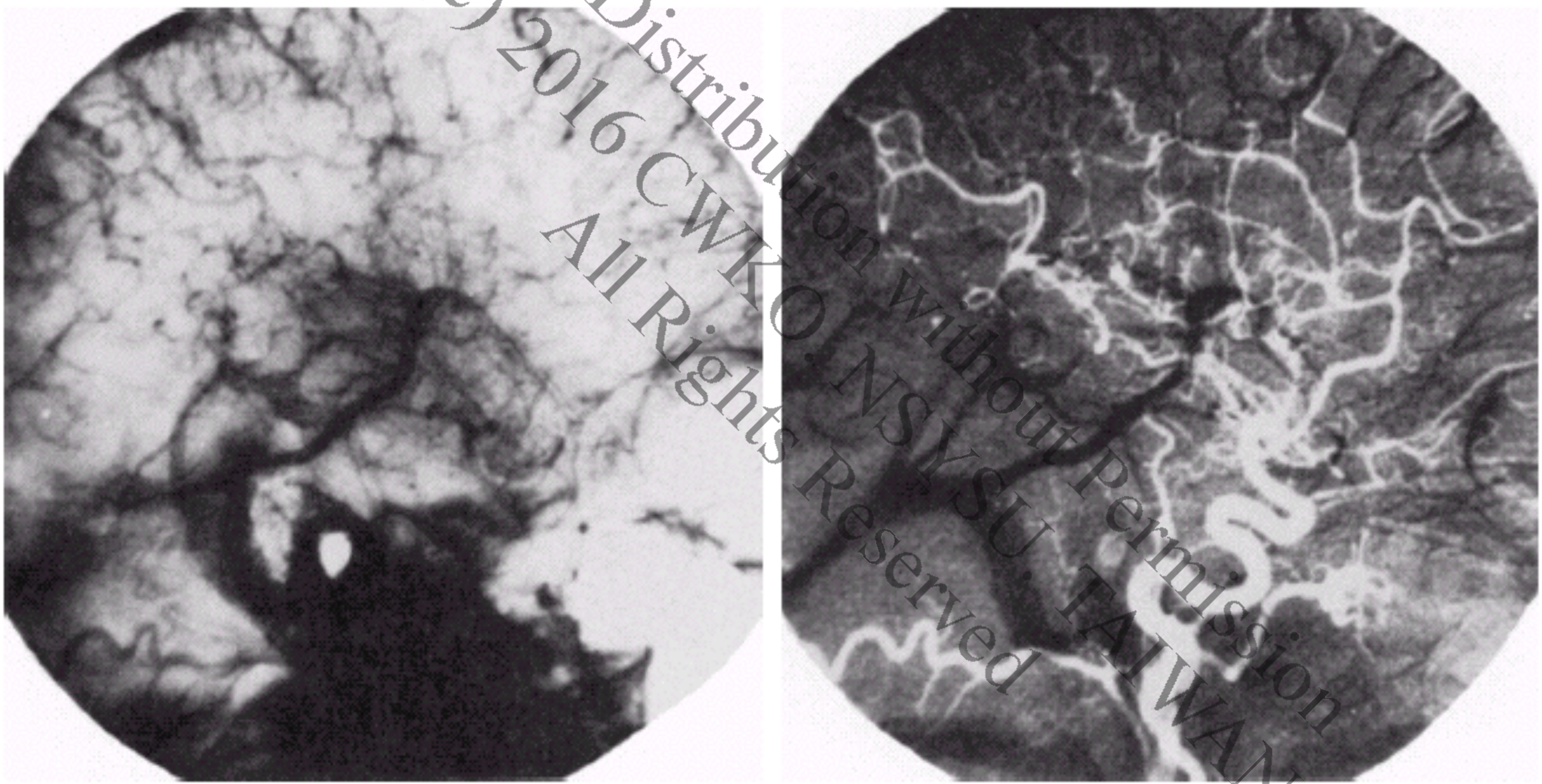


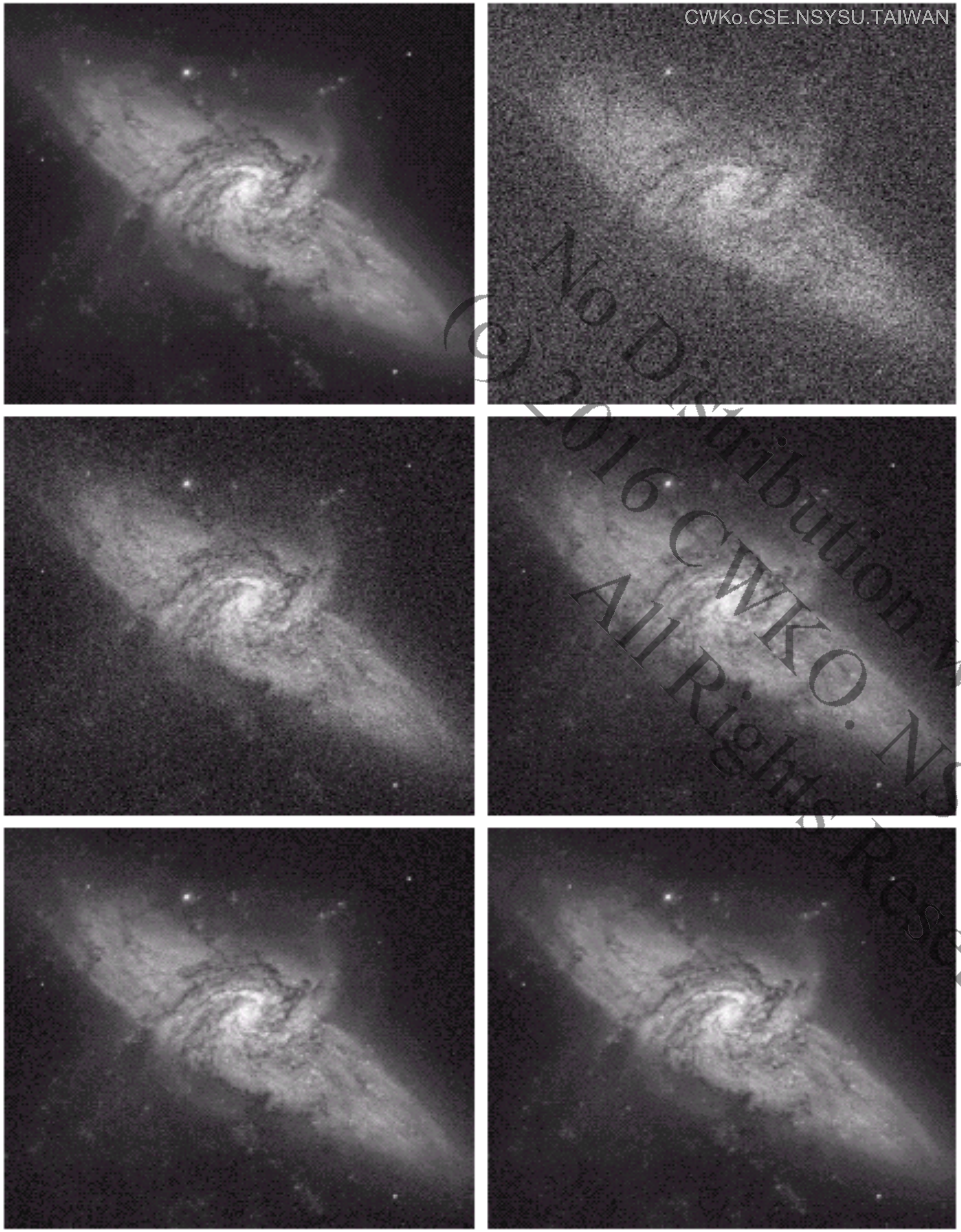
Image Average

$$g(x,y) = f(x,y) + \eta(x,y)$$

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{k} g_i(x,y)$$

$$E\{\bar{g}(x,y)\} = f(x,y)$$

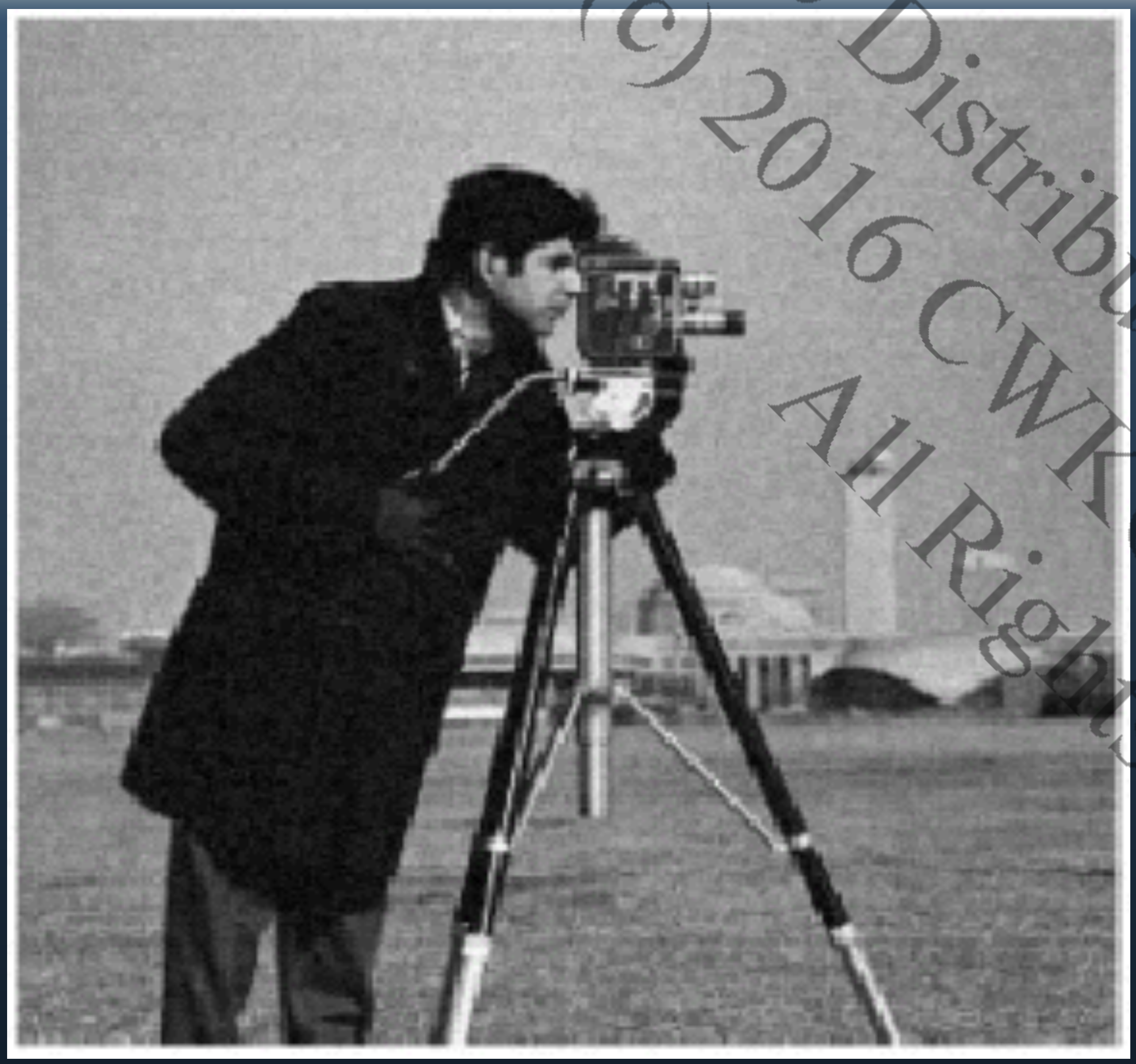
$\eta(x,y)$: noise, uncorrected, zero mean



a b
c d
e f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64,$ and 128 noisy images. (Original image courtesy of NASA.)

Image Average -- example



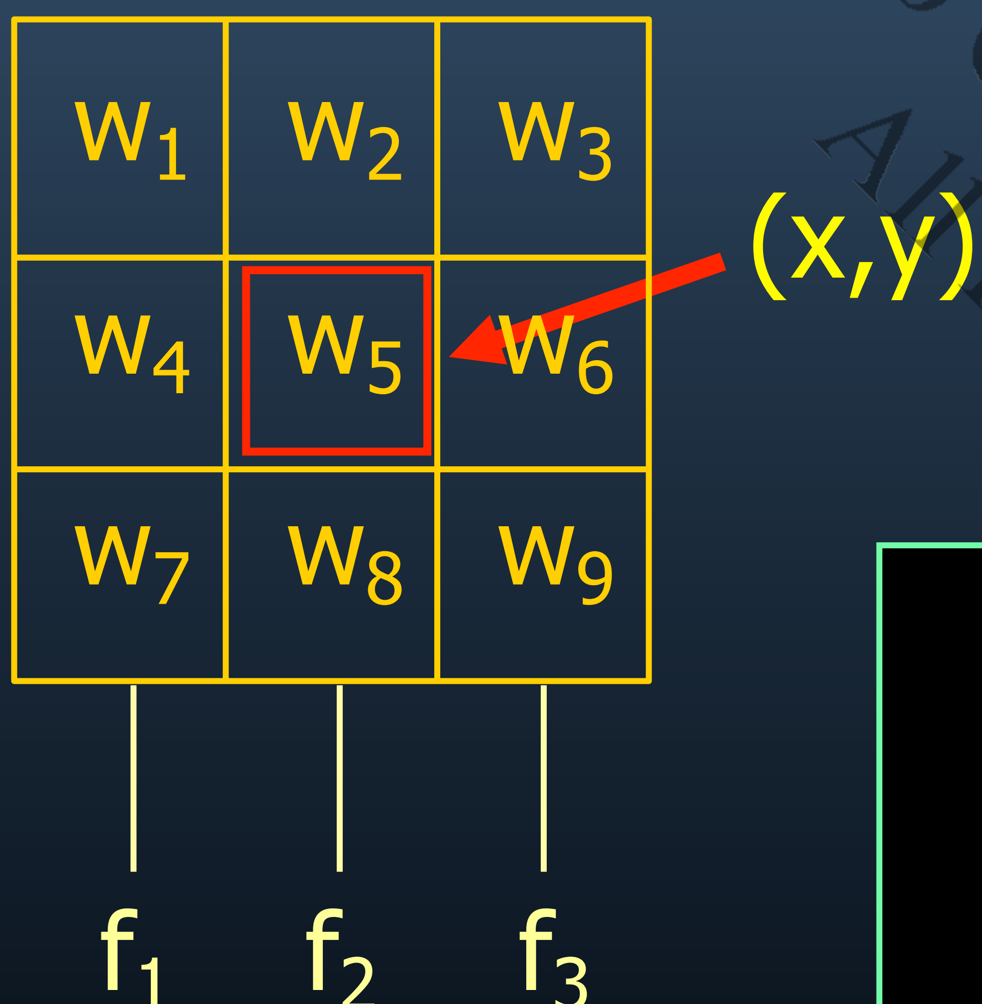
original image

averaged image
(10 samples)

Spatial Filtering

Spatial Filtering

- Smoothing (low-pass) filters



Replace f(x,y) with

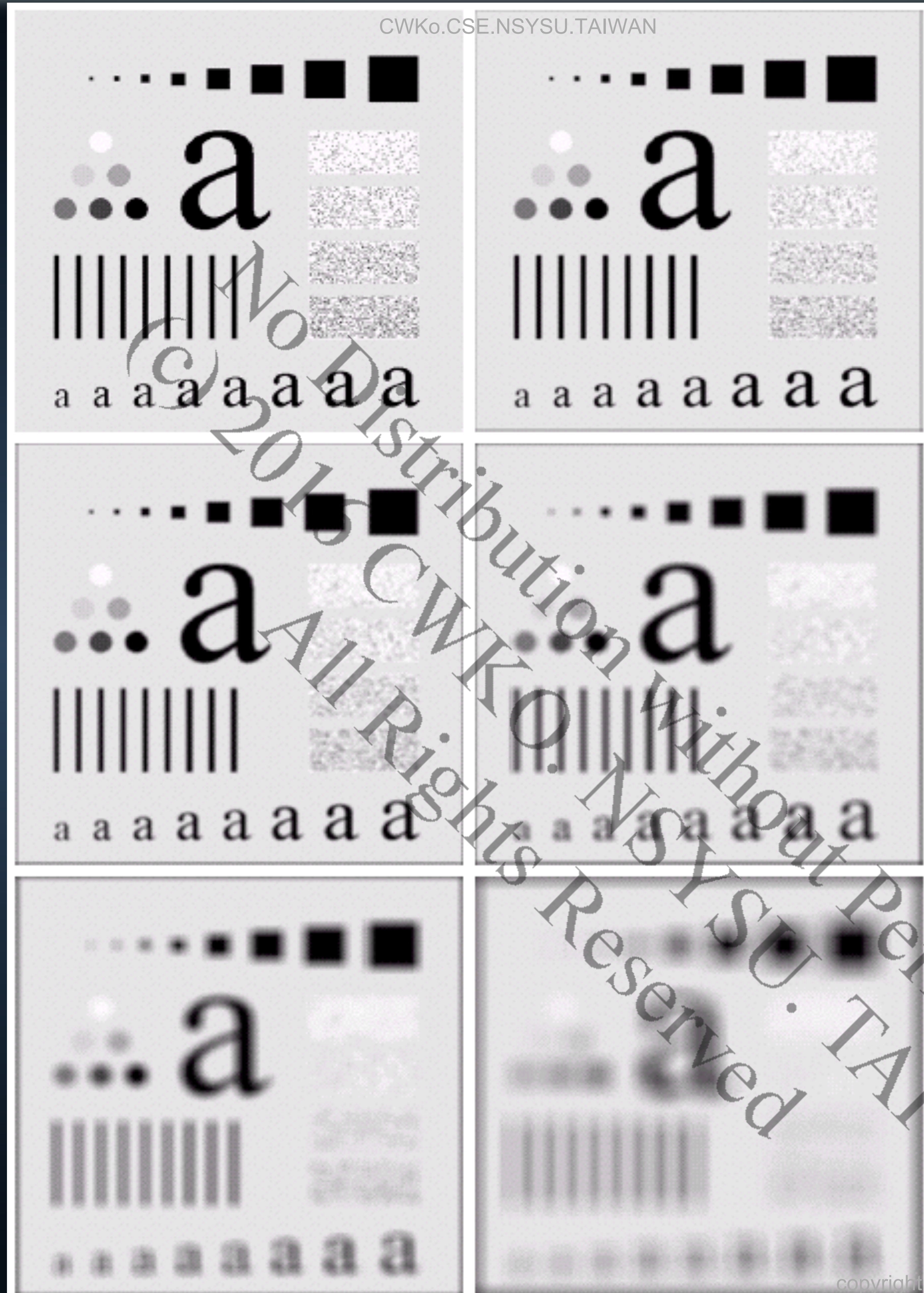
$$\hat{f}(x,y) = \sum_i w_i f_i$$

LPF: reduces additive noise

$\Longrightarrow$ blurs the image

$\Longrightarrow$ sharpness details are lost

(Example: Local averaging)



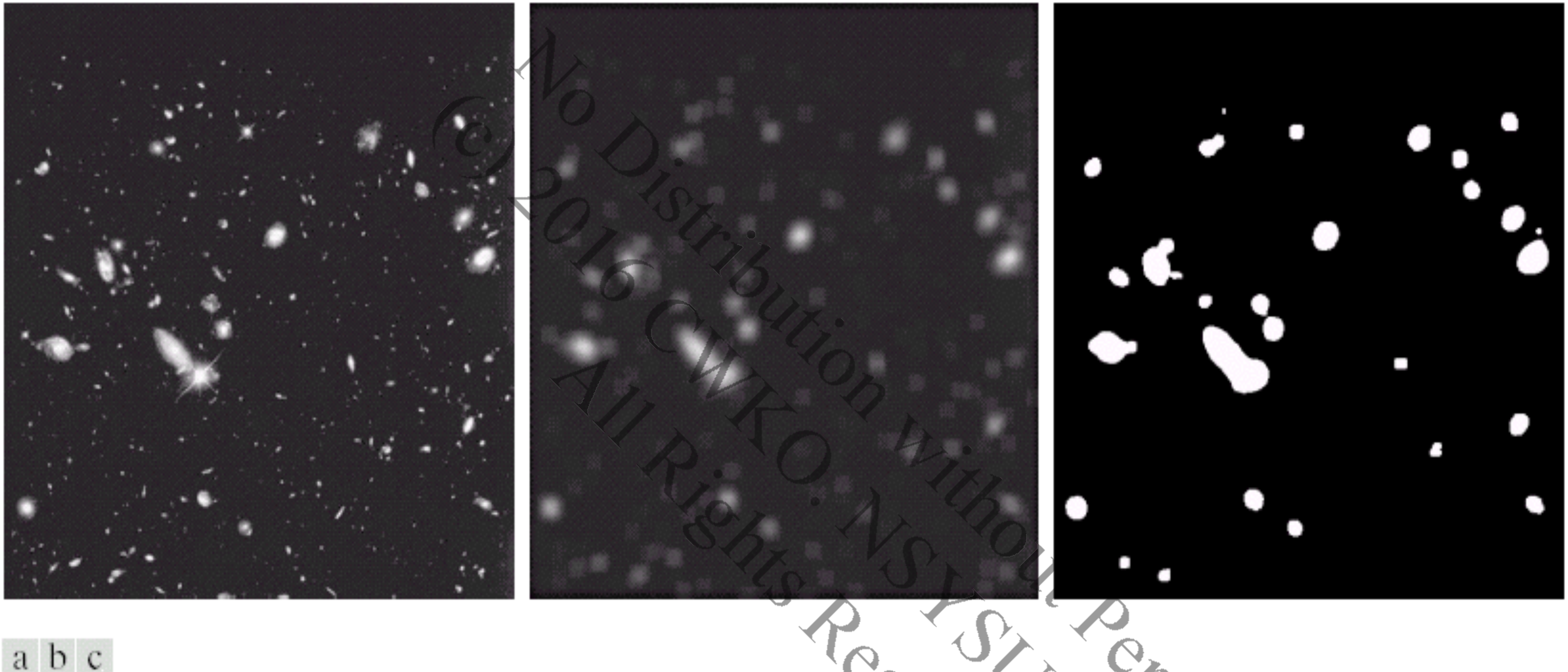


FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15×15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Median Filter

- Replace f (x,y) with median [f (x', y')]
- Example:

→ (10,15,20,20,$\boxed{20}$,20,20,25,100)

10	20	20
20	$\boxed{15}$	20
25	20	100

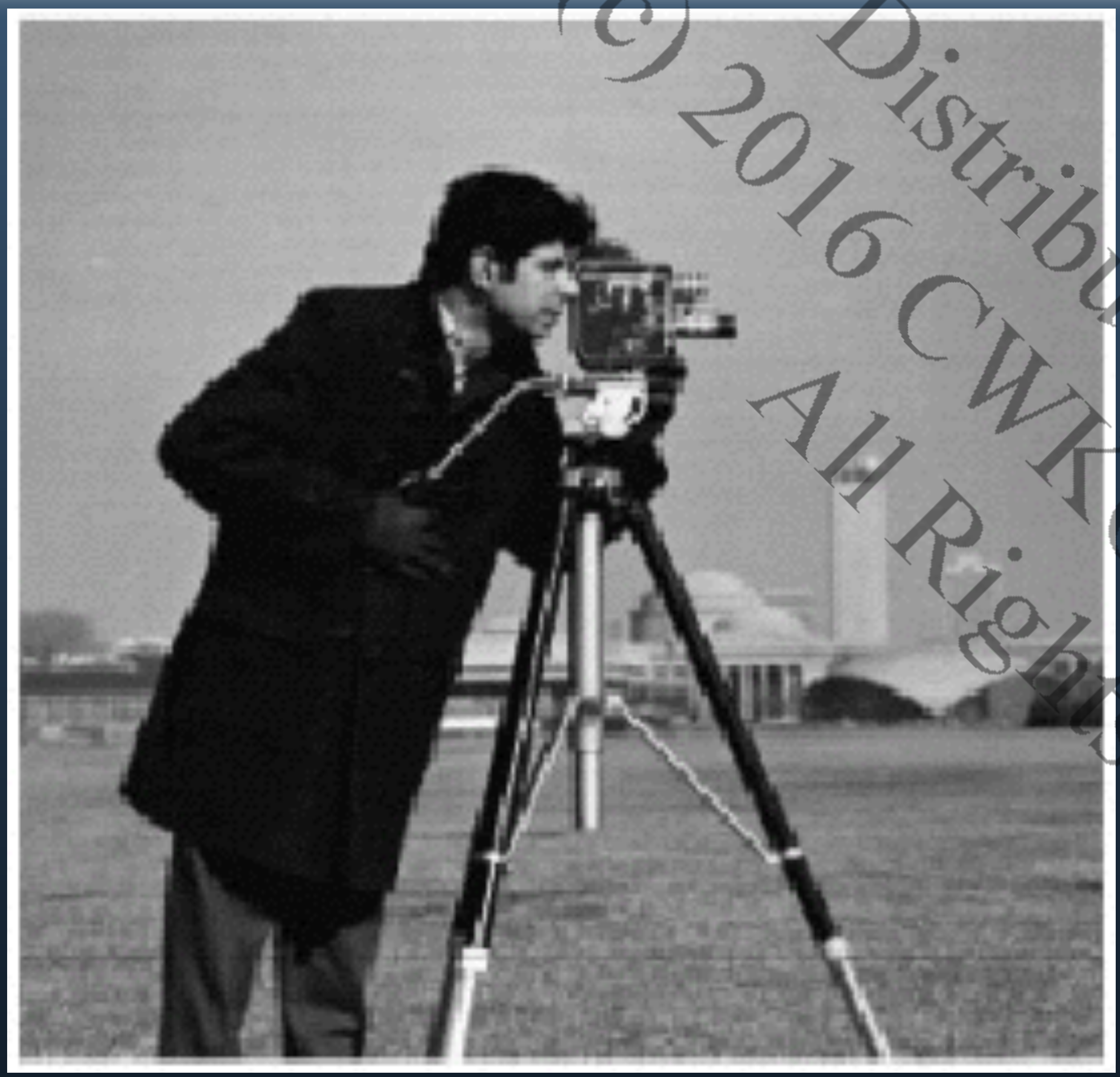
Median = 20

So replace "15" with "20"

Median Filter

- Useful in eliminating intensity spikes. (salt & pepper noise)
- Better at preserving edges.

Original and with Salt & Pepper Noise



% imnoise(image, “salt & pepper”);

Local Averaging

Median filtered

Sharpening Filter

- Enhance finer image details (such as edges)
- Detect region or object boundaries.

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

smoothing v.s. sharpening

Derivative

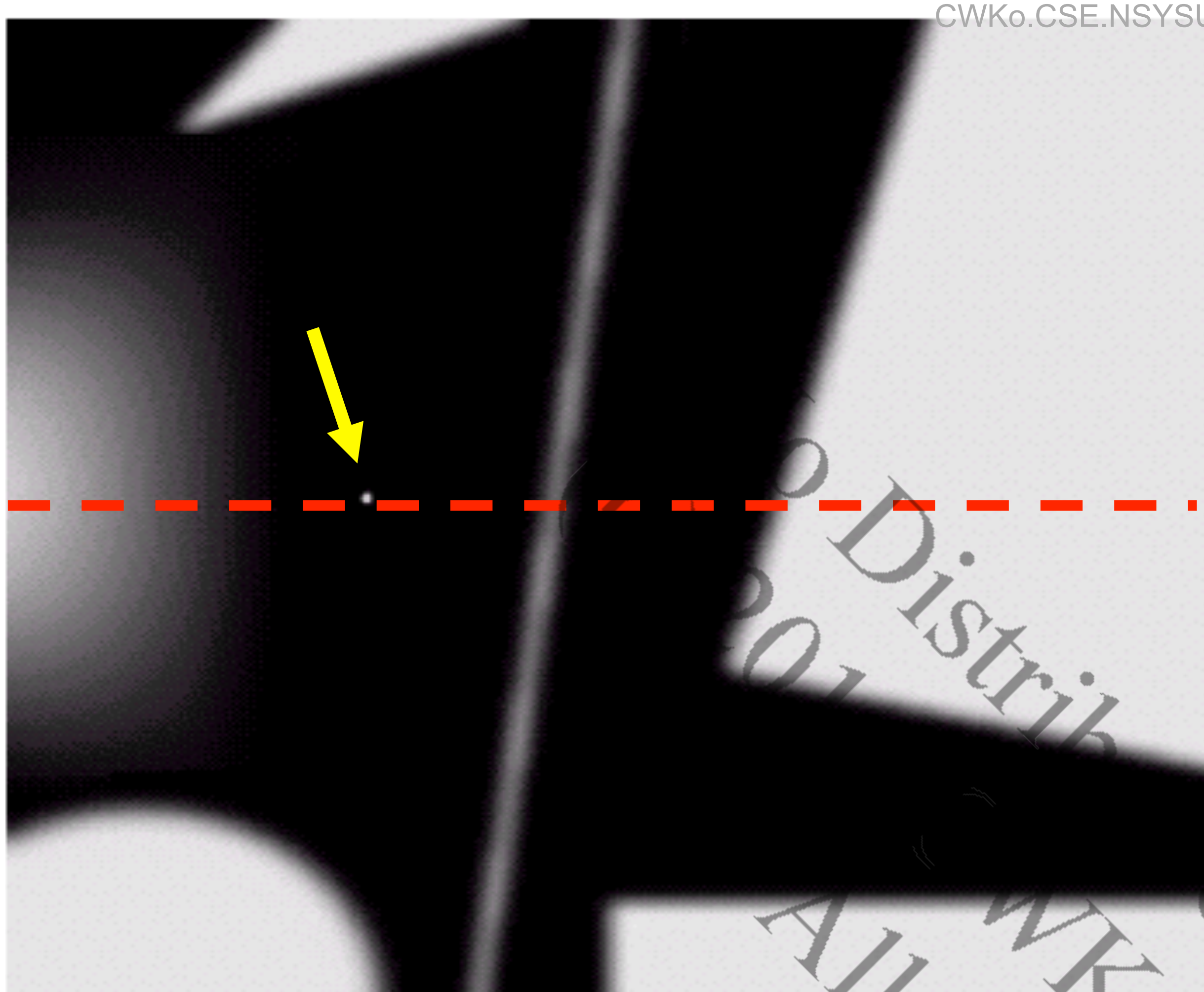
- First derivative

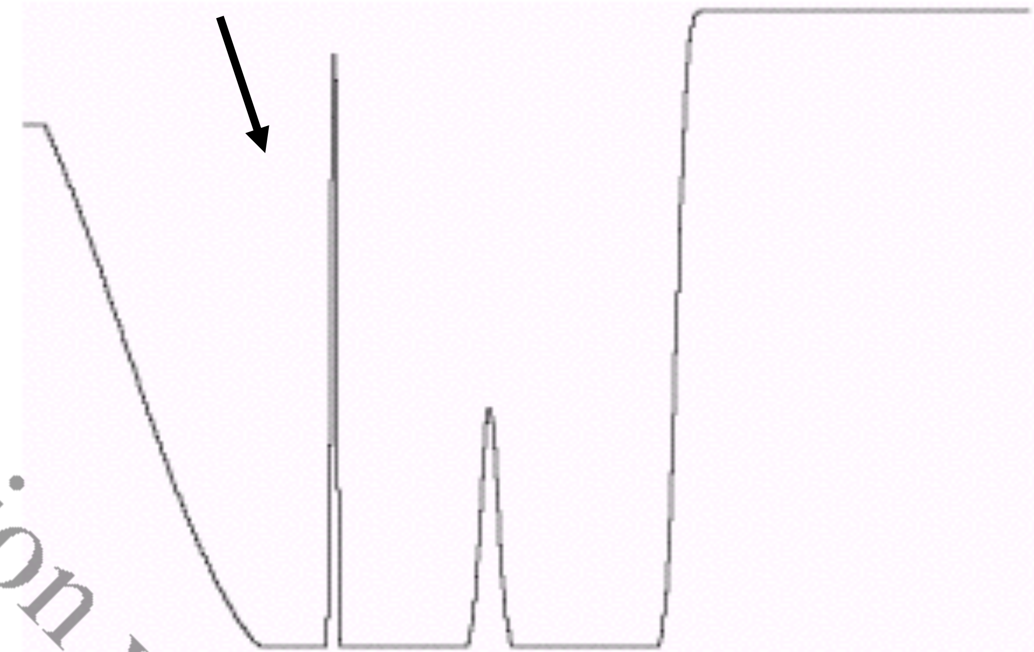
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- Second derivative

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

	f(x-1)	
	f(x)	
	f(x+1)	





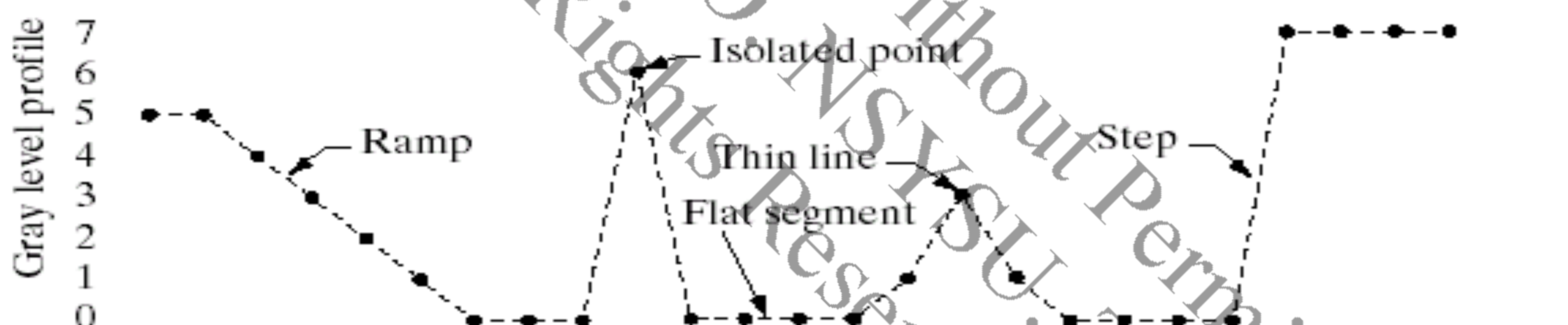


Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	•	•
First Derivative		−1	−1	−1	−1	−1	0	0	6	−6	0	0	0	1	2	−2	−1	0	0	0	7	0	0	0			
Second Derivative		−1	0	0	0	0	1	0	6	−12	6	0	0	1	1	−4	1	1	0	0	7	−7	0	0			

Laplacian Based Edge Detectors

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

$$= [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

0	1	0
1	-4	1
0	1	0

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Sharpening with the Laplacian

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) & \text{If the center coefficient of the Laplacian mask} < 0 \\ f(x,y)+\nabla^2 f(x,y) & \text{If the center coefficient of the Laplacian mask} > 0 \end{cases}$$

$$\frac{\partial^2 f}{\partial x^2}=f(x+1)+f(x-1)-2f(x)$$

Laplacian Based Edge Detectors

- Rotationally symmetric, linear operator
- Second derivatives

  $\Longrightarrow$ sensitive to noise
- Increase the contrast at the locations of gray-level discontinuities.

Unsharp Masking

$$g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)$$

- Subtract Low pass filtered version from the original
- Emphasizes high frequency information

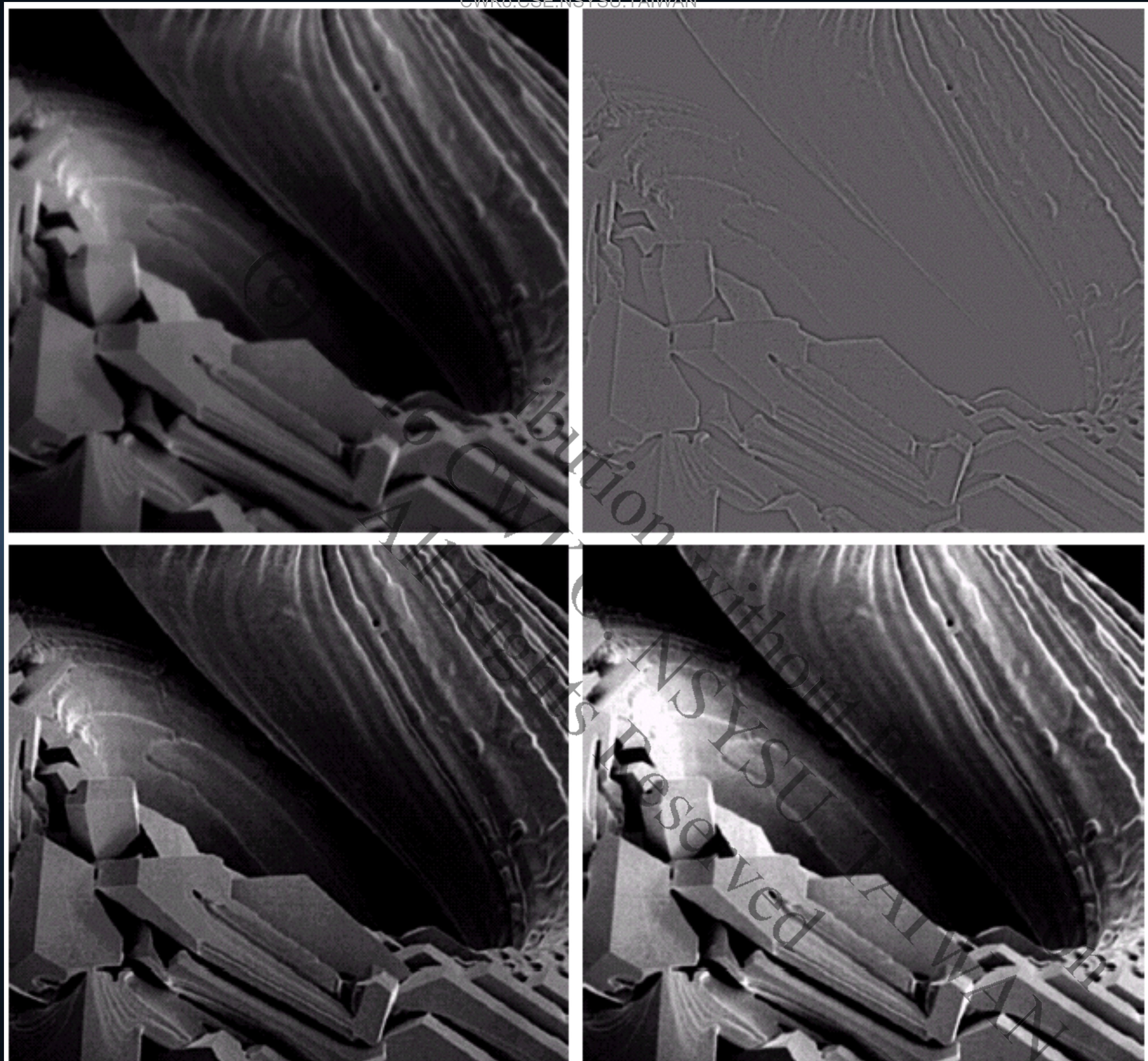
High-Boost Filtering

$$g(x,y) = f(x,y) + k * g_{mask}(x,y) \qquad k \geq 0$$

k=1 --> unsharp masking

k>1 --> highboost filtering

k<1 --> ?



The Gradient

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

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

$$\nabla f \approx |G_x| + |G_y|$$

The Gradient

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$G_x = (Z_8 - Z_5)$$

$$G_y = (Z_6 - Z_5)$$

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Roberts cross-gradient operators

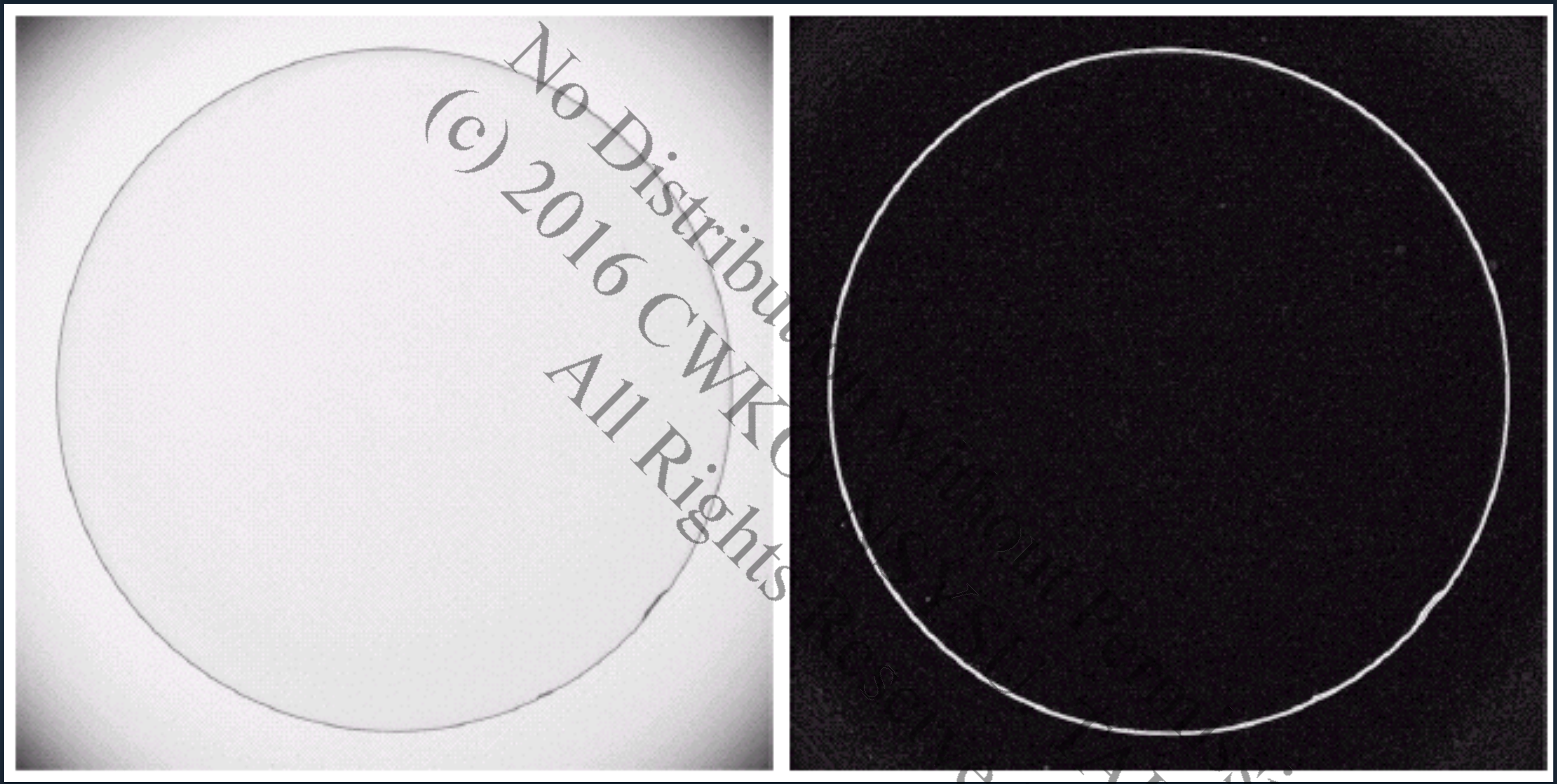
-1	0
0	1

0	-1
1	0

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

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

Sobel operators



Gradient Process

- Edge detection
- Constant or slowly varying shades are eliminated
- Automated inspection
- Chap 10 -- segmentation

Next Lecture

Fourier Transform