

# Xử lý ảnh

Hoàng Văn Hiệp

Bộ môn Kỹ thuật máy tính

Viện Công nghệ thông tin và Truyền thông

Email: [hiephv@soict.hut.edu.vn](mailto:hiephv@soict.hut.edu.vn)

# Nội dung

- ❑ Chương 1. Giới thiệu chung
- ❑ Chương 2. Thu nhận & số hóa ảnh
- ❑ Chương 3. Cải thiện & phục hồi ảnh
- ❑ Chương 4. Phát hiện tách biên, phân vùng ảnh
- ❑ Chương 5. Trích chọn các đặc trưng trong ảnh
- ❑ Chương 6. Nén ảnh
- ❑ Chương 7. Lập trình xử lý ảnh bằng Matlab và C

# Chương 3. Cải thiện và phục hồi ảnh

- ❑ Cải thiện ảnh
- ❑ Phục hồi ảnh

# Cải thiện ảnh

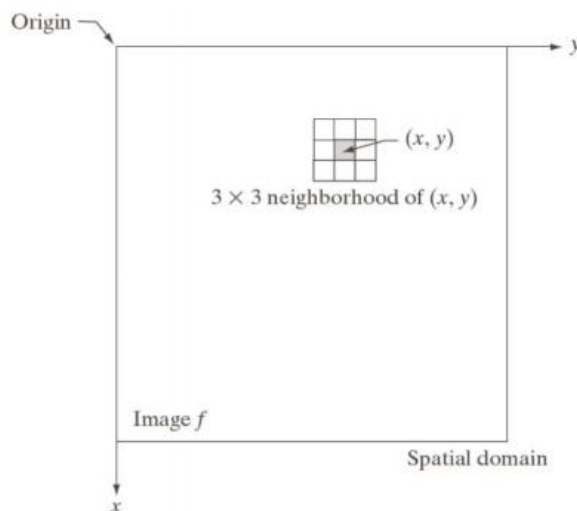
- ❑ Xử lý ảnh để đầu ra “tốt” hơn đầu vào cho mục đích nhất định
  - Do đó: Cải thiện ảnh rất phụ thuộc vào từng ứng dụng cụ thể
- ❑ Phương pháp cải thiện ảnh
  - Xử lý trên miền không gian
    - Xử lý trên điểm ảnh
    - Xử lý mặt nạ
  - Xử lý trên miền tần số
    - Các phép lọc
  - Xử lý trên màu sắc

# Xử lý trên miền không gian

## □ Spatial Domain process

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

- Trong đó:  $f(x, y)$  ảnh gốc
- $g(x, y)$  ảnh sau biến đổi
- $T$ : phép biến đổi ảnh



# Xử lý trên miền không gian

## □ Nếu xét cửa sổ lân cận: $1 \times 1$

- Phép xử lý trên điểm ảnh
- Giá trị đầu ra tại một điểm ảnh chỉ phụ thuộc điểm đó, không phụ thuộc vào các điểm khác

## □ Nếu xét cửa sổ lân cận $w \times w$

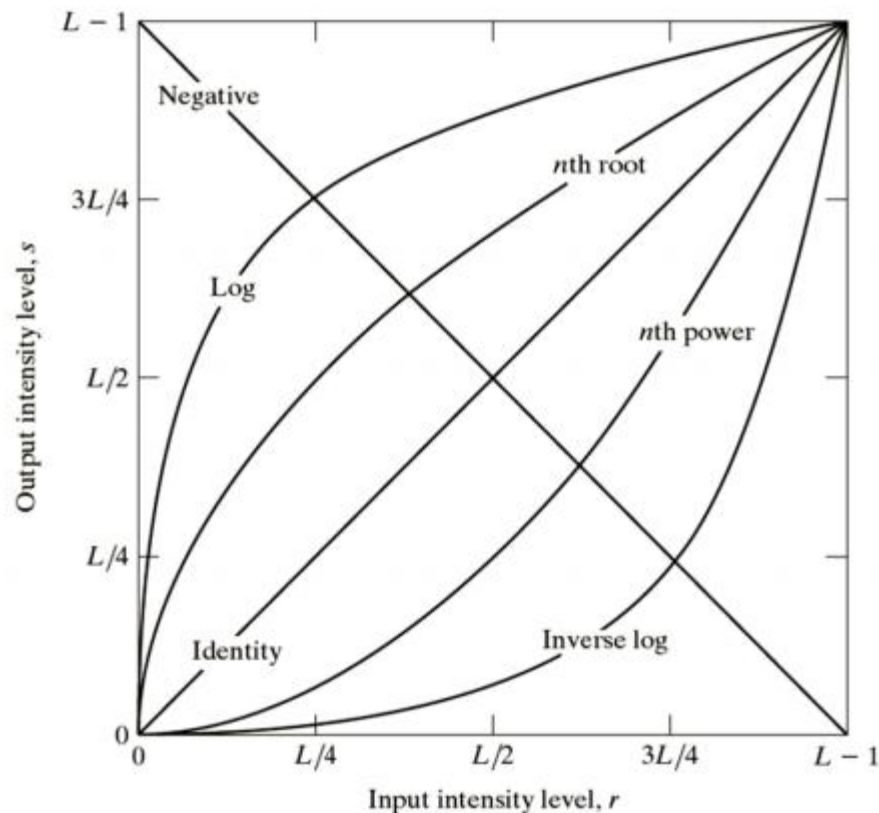
- Cửa sổ lân cận còn gọi là: mặt nạ (mask), nhân (kernel), Cửa sổ (window), bộ lọc (filter), template
- Giá trị đầu ra tại một điểm phụ thuộc vào các điểm lân cận của nó

# Các phép biến đổi ảnh dựa trên điểm ảnh

- ❑ Phép biến đổi âm bản ảnh
- ❑ Biến đổi dùng hàm logarit
- ❑ Biến đổi dùng hàm mũ
- ❑ Biến đổi dựa trên histogram
- ❑ Biến đổi dựa trên các phép số học/logic

# Một số phép xử lý cơ bản dựa trên điểm ảnh

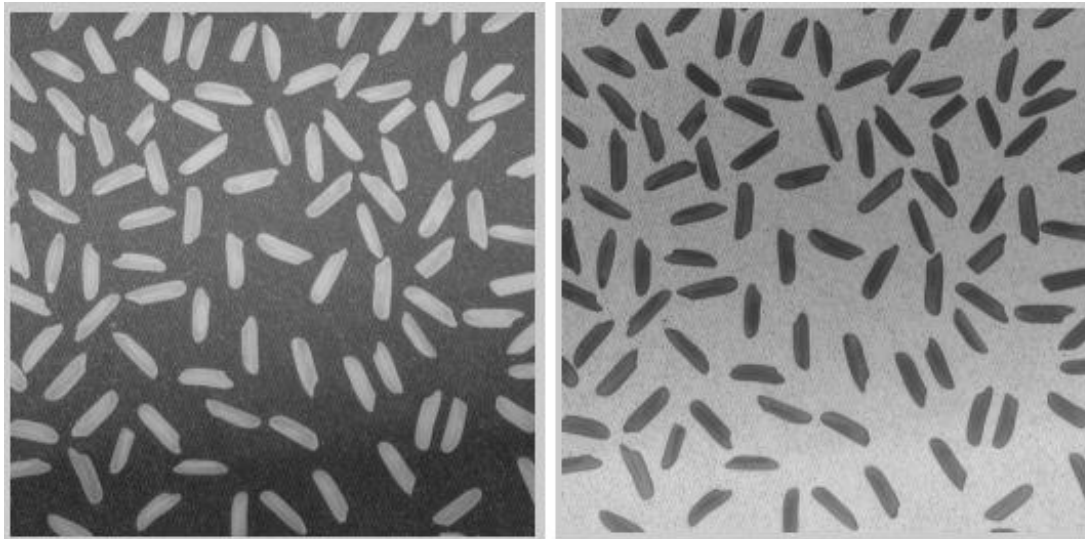
□ Một số hàm biến đổi





# Phép biến đổi âm bản ảnh

□  $s = L - 1 - r$

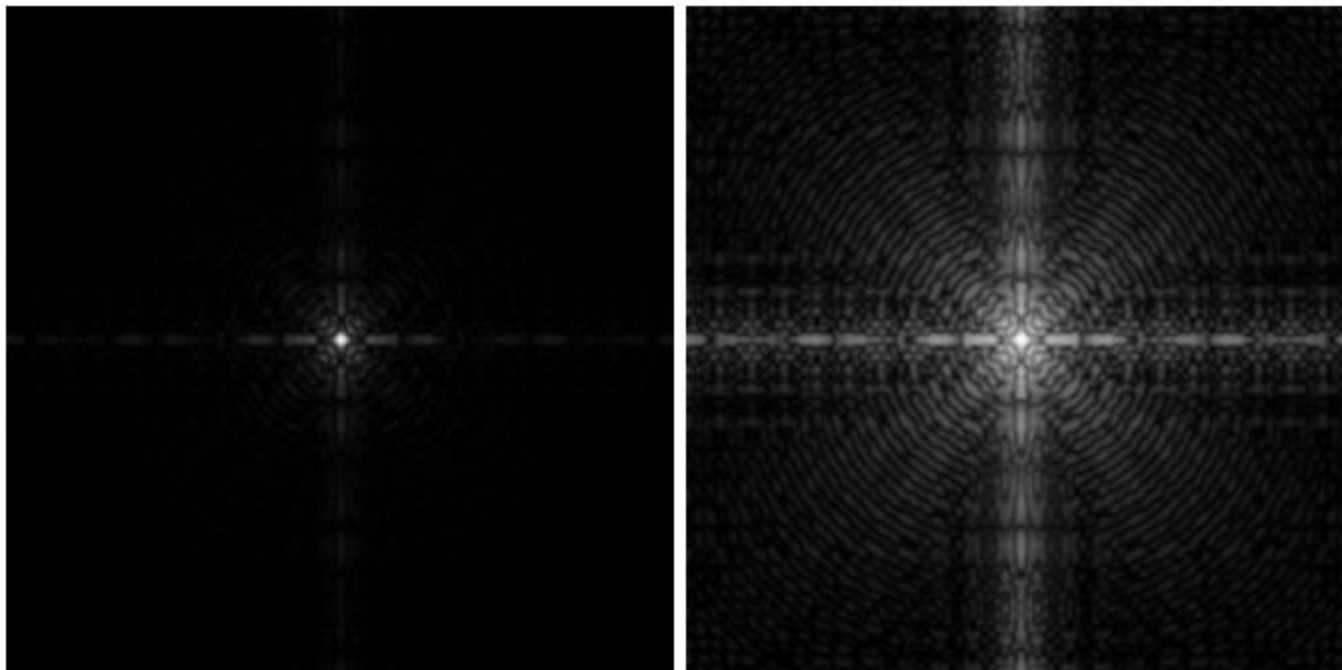


Matlab code:

```
I = imread('rice.png');  
J = 255 - I; imshow(J)
```

# Phép biến đổi log

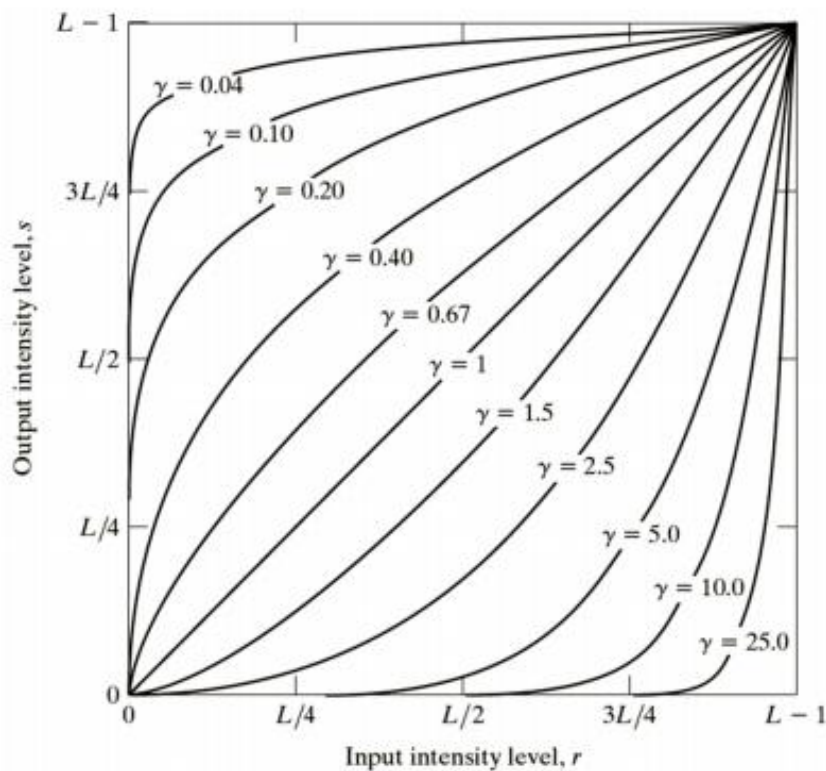
- ❑ Công thức  $s = c \log(1 + r)$
- ❑ Tác dụng: Kéo giãn các giá trị ở vùng tối, thu hẹp các giá trị ở vùng sáng



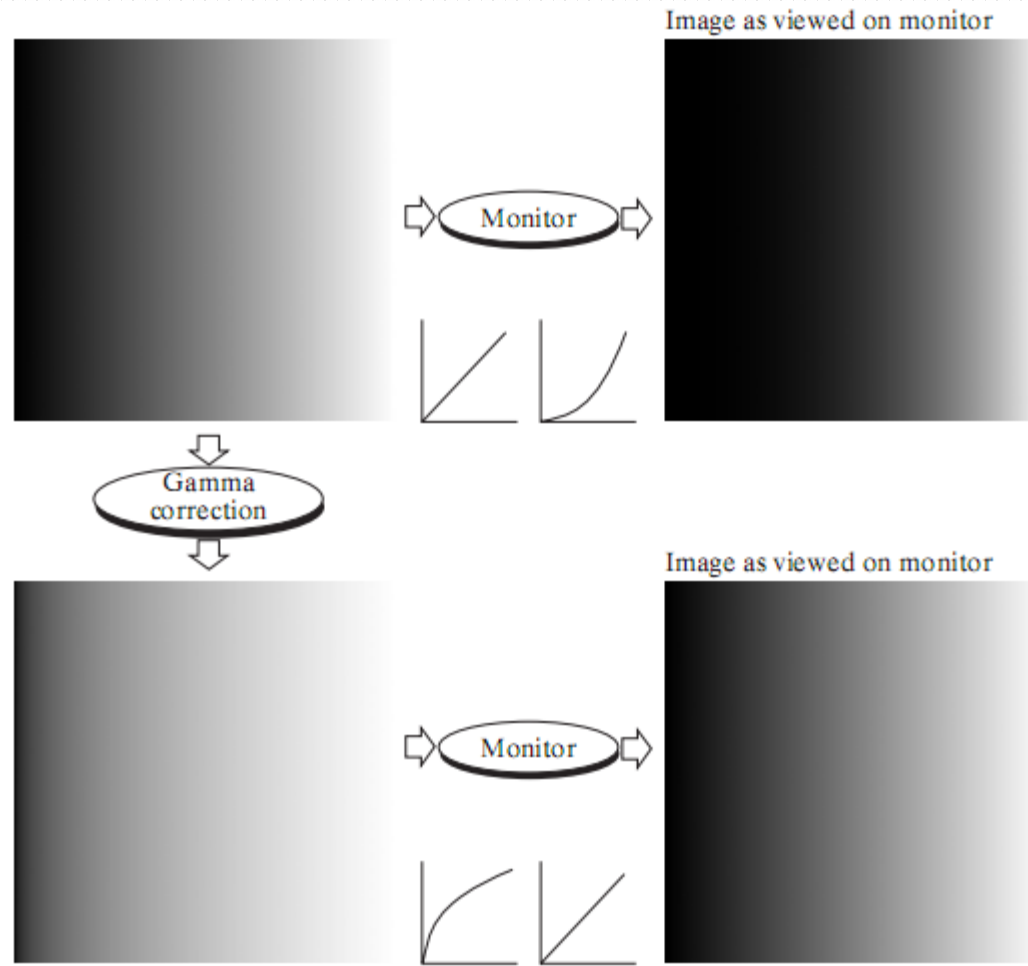
# Phép biến đổi hàm mũ

□ Công thức tổng quát

$$s = c(r + \varepsilon)^\gamma$$



# Gama correction



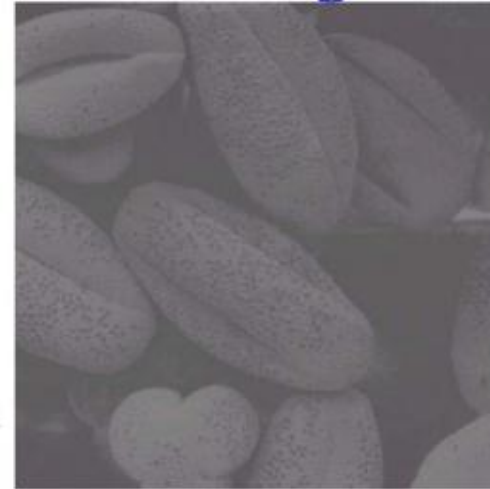
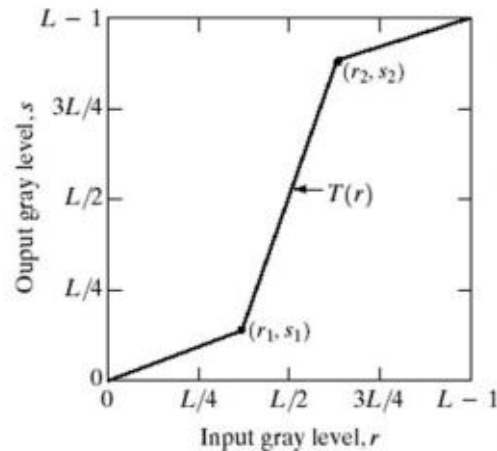
# Phép biến đổi hàm mũ

□  $c = 1; \gamma_1 = 3; \gamma_2 = 4; \gamma_3 = 5;$



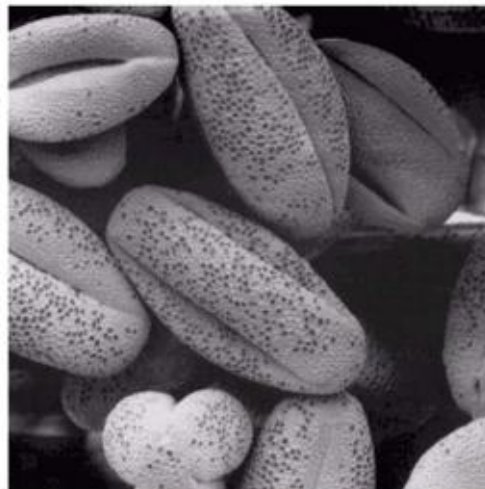
# Phép biến đổi tuyến tính từng khúc

Original



a b  
c d

**FIGURE 3.10**  
Contrast stretching.  
(a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

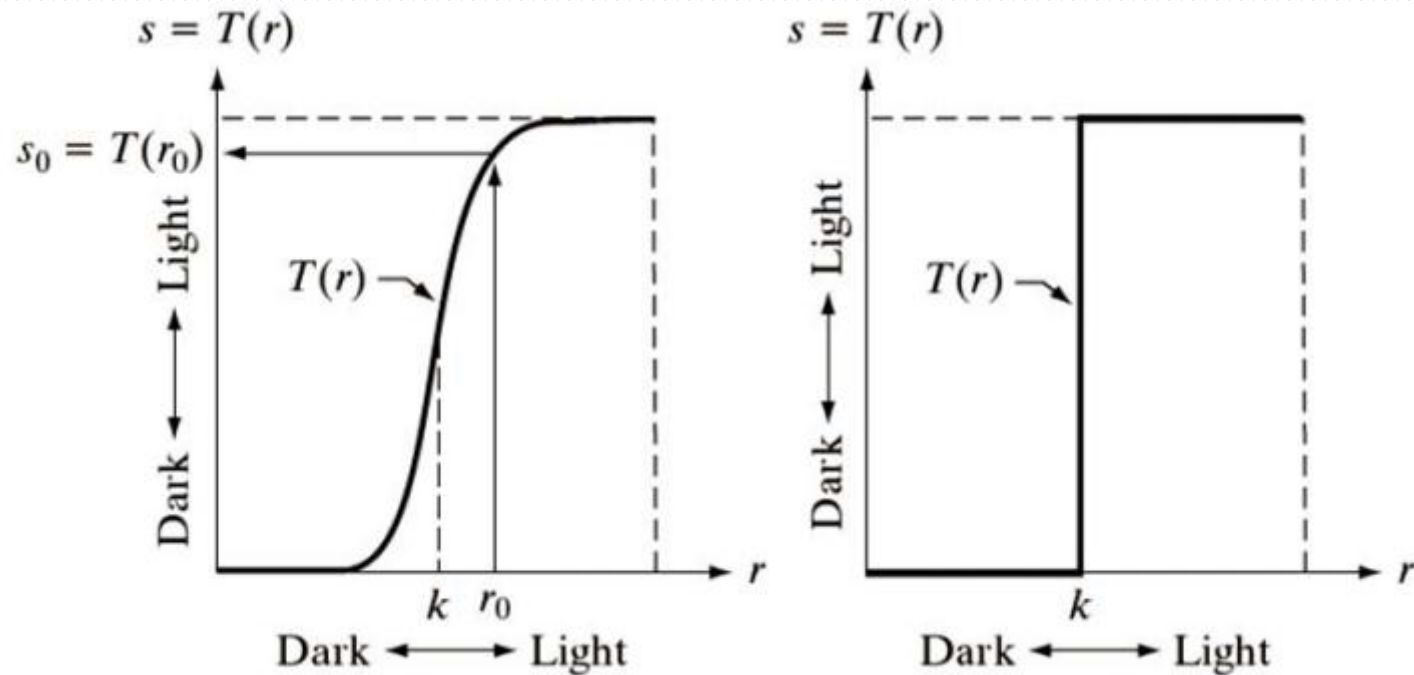


C. S.



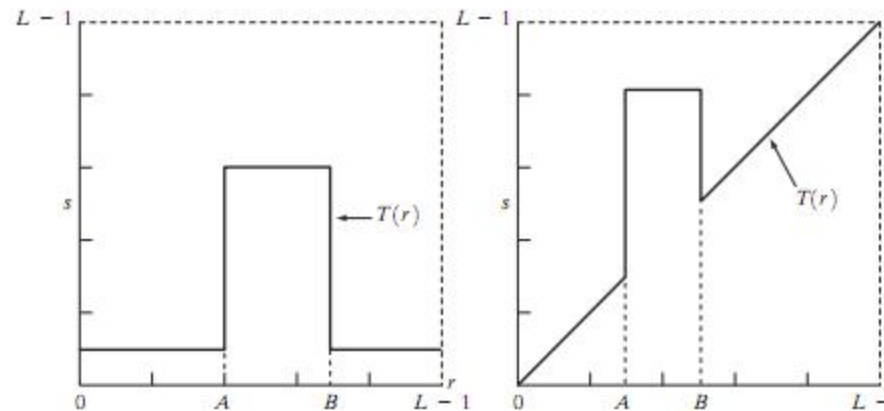
THR.

# Biến đổi tăng độ tương phản

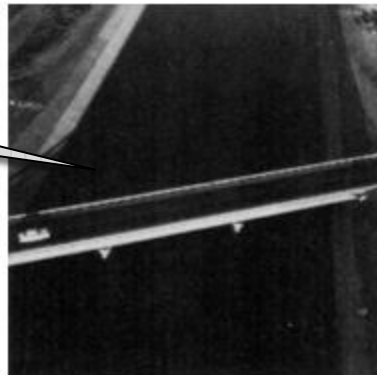


# Gray level slicing

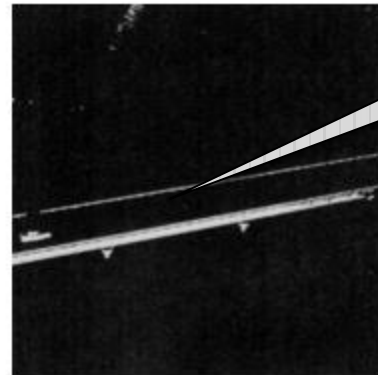
- Tăng cường mức xám ở một dải cố định  $[A, B]$



Ảnh  
gốc



Ảnh biến đổi qua  
hàm đầu

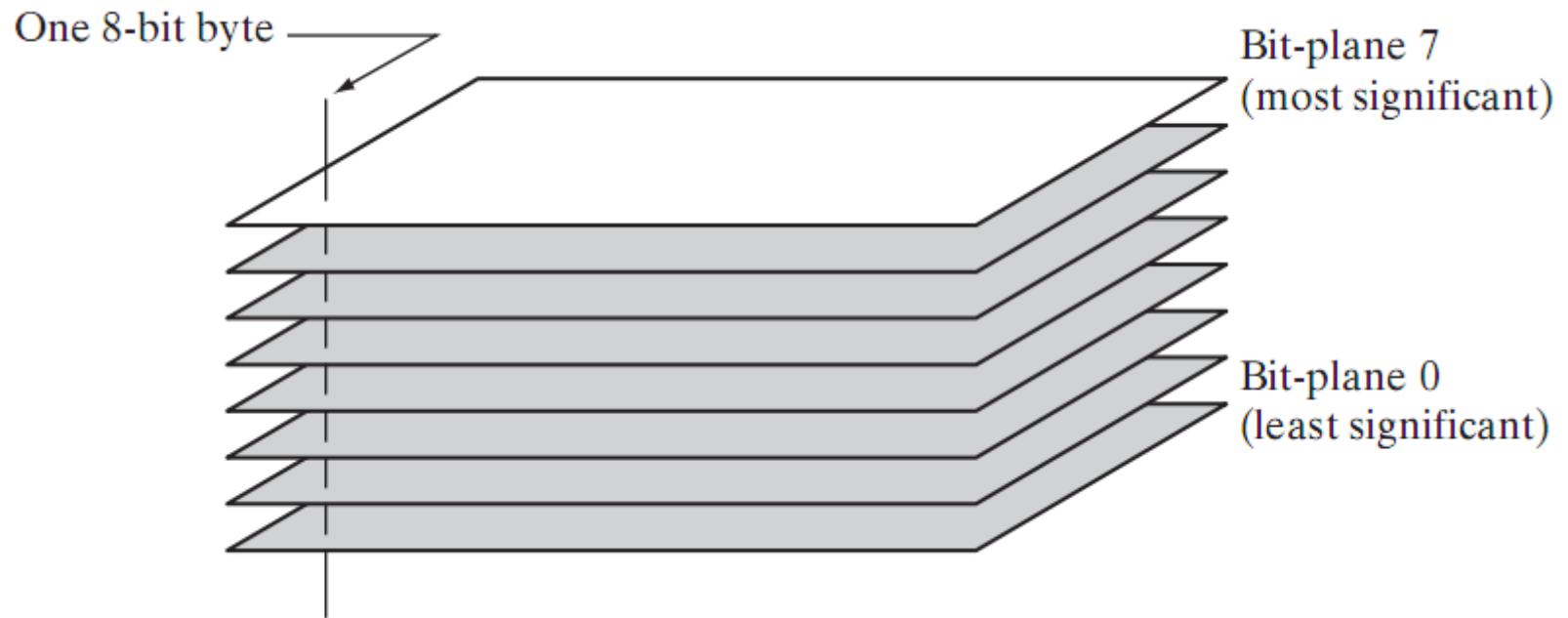




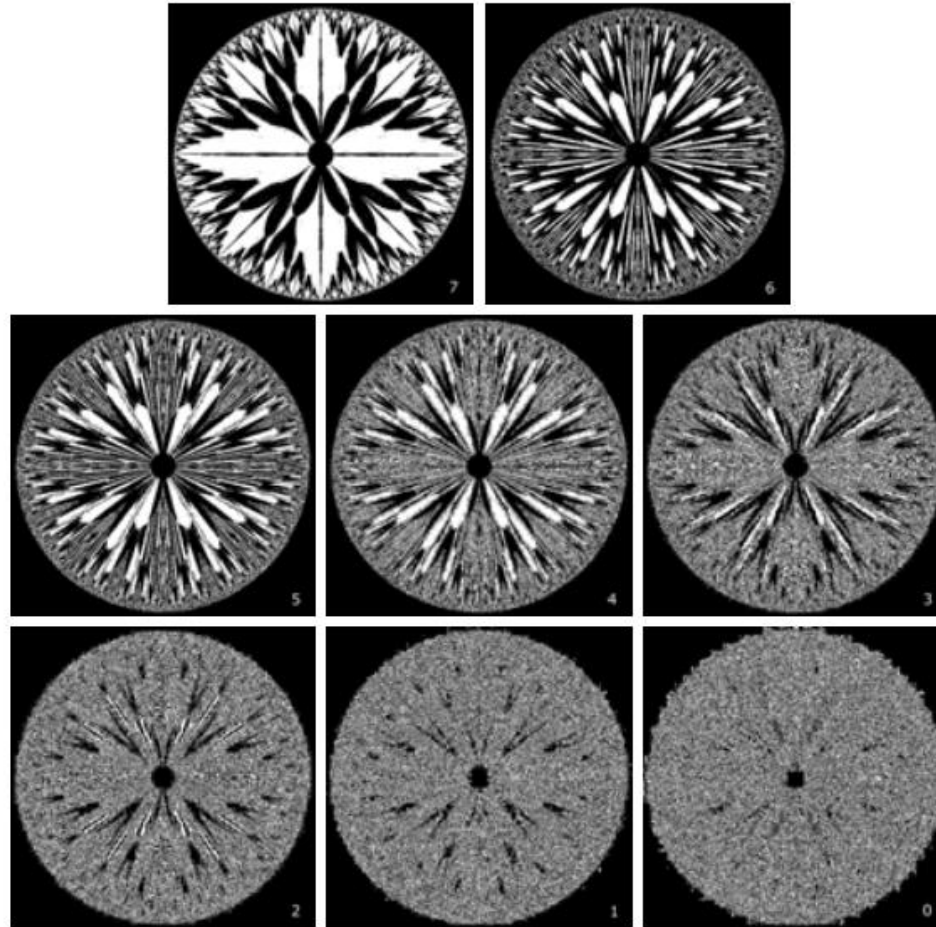
# Bit-plan slicing

- ❑ Với ảnh 8bit: mỗi pixel được biểu diễn bằng 8 bit
- ❑ Tưởng tượng mỗi ảnh là tổng hợp của 8 mặt phẳng 1 bit (1bit - plan): từ plan 0 đến plan 7
  - Plan 0: chứa tất cả các bit thấp nhất trong các byte pixel trong ảnh
  - ...
  - Plan 7: chứa tất cả các bit cao nhất trong các byte pixel trong ảnh

# Bit-plan slicing



# Bit-plan slicing



# Một số phép xử lý dựa trên điểm ảnh

- Bài tập: Cài đặt các phép biến đổi dựa trên điểm ảnh trên bằng Matlab

# Phép biến đổi dựa trên histogram

## □ Histogram là gì?

- Histogram của ảnh đa mức xám:  $[0 \text{ } L-1]$  là hàm rời rạc:

- $h(r_k) = n_k$

- Với  $r_k$  là thành phần mức xám thứ  $k$
  - $n_k$ : số lượng pixel có mức xám là  $r_k$

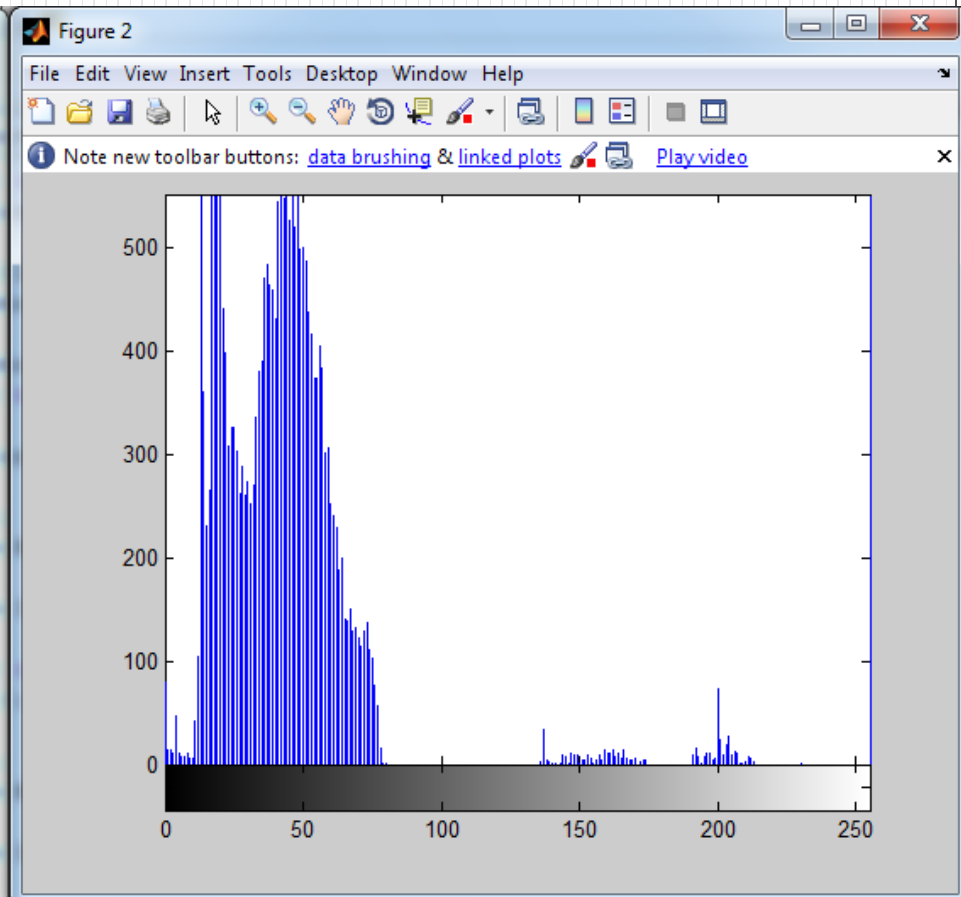
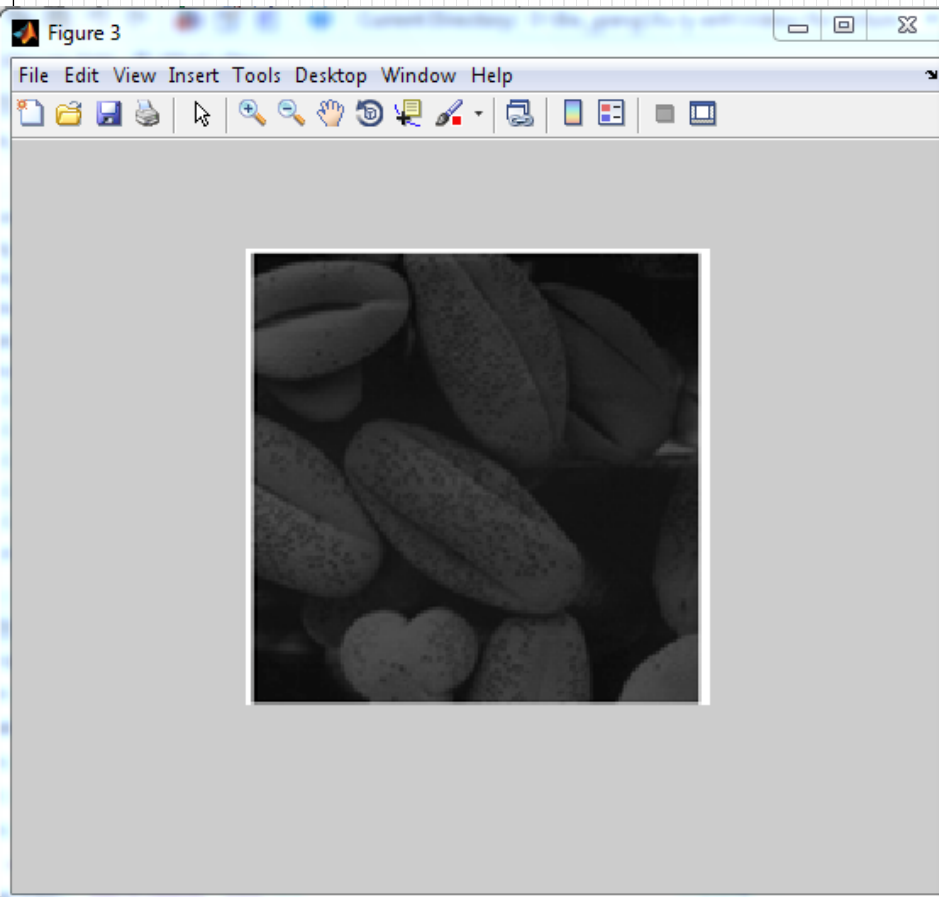
- Dạng chuẩn hóa:

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

Với  $n$ : tổng số pixel trong ảnh

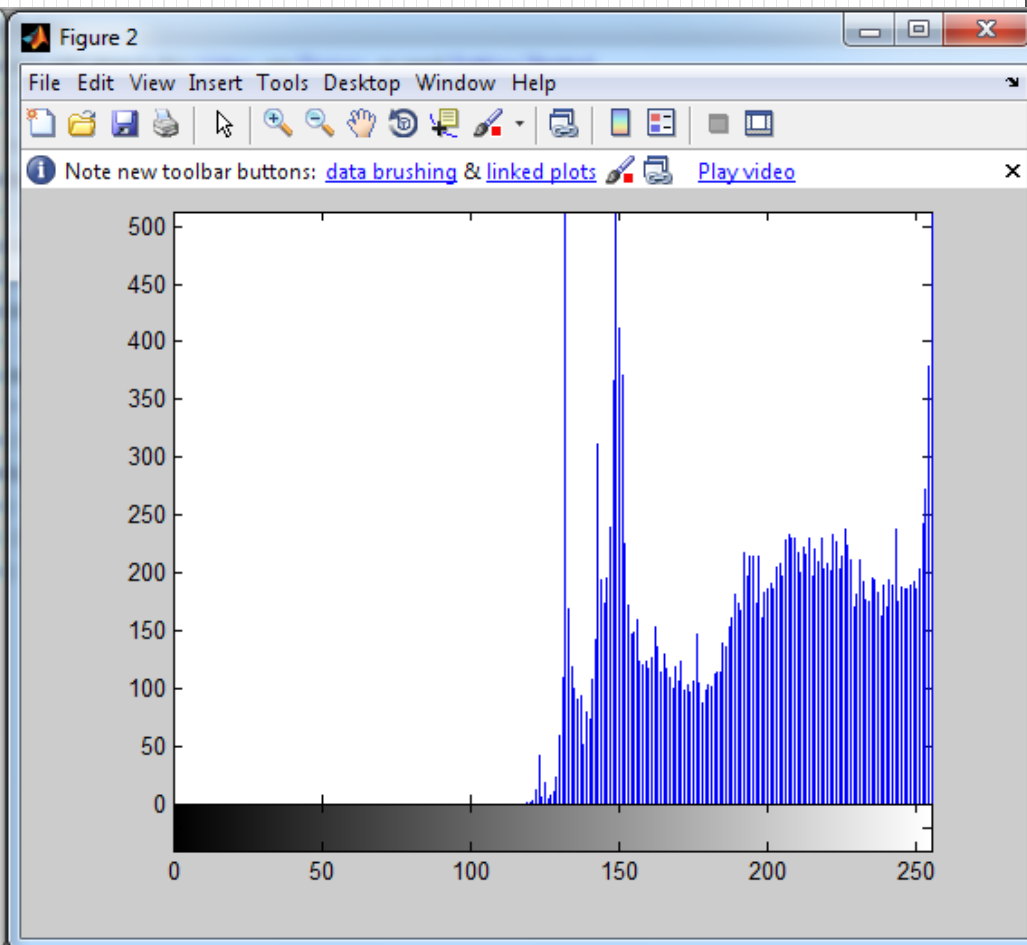
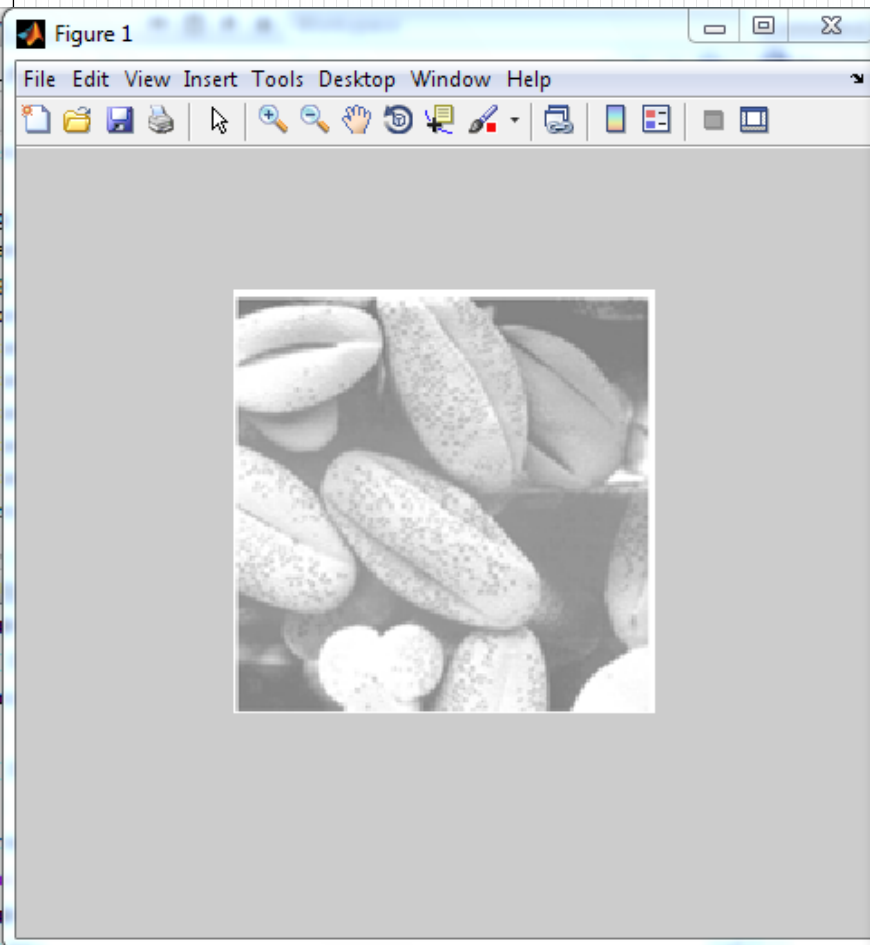
# Histogram

□ Ảnh tối



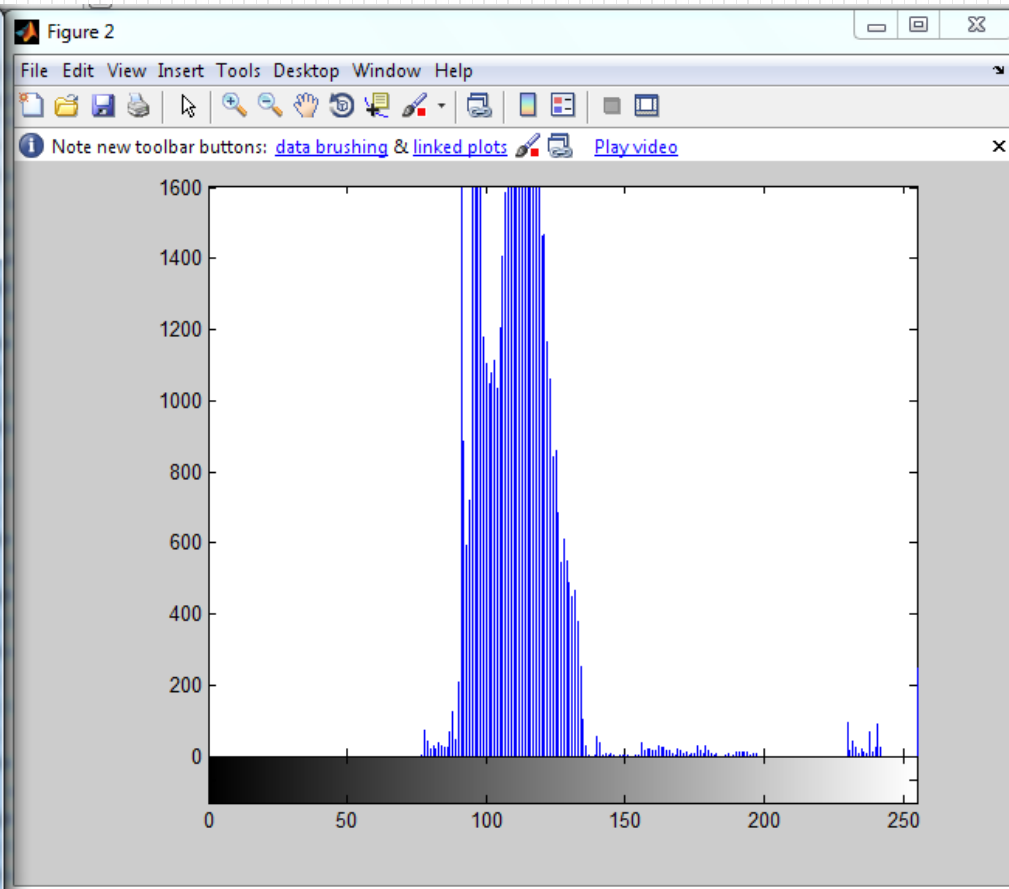
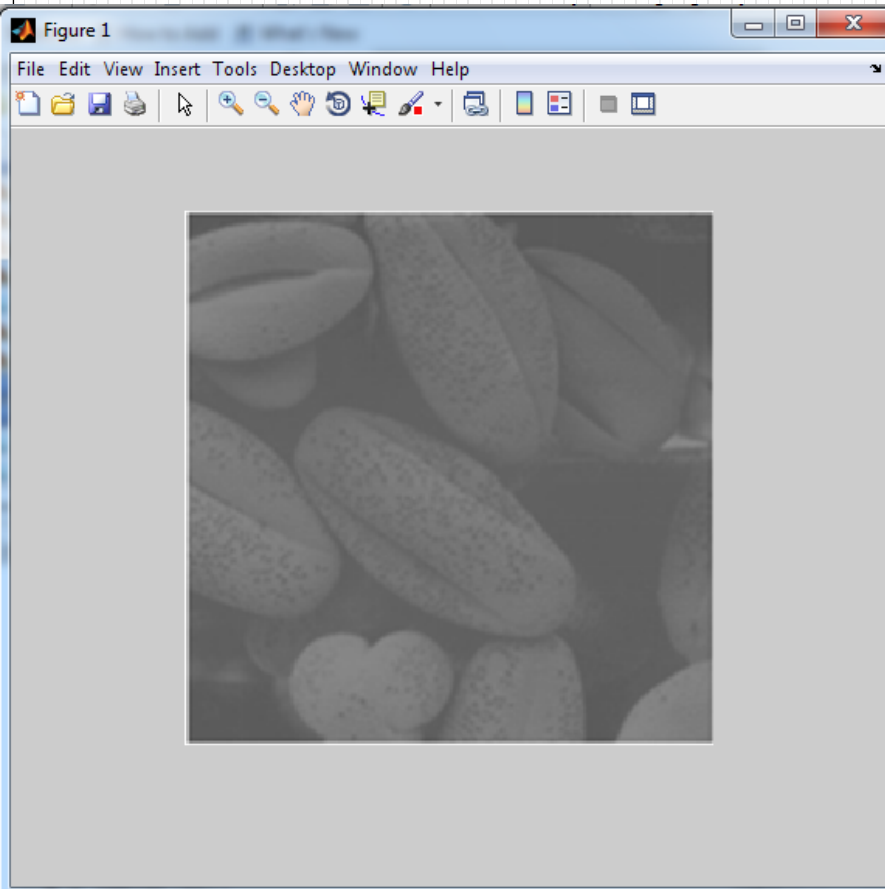
# Histogram

□ Ảnh sáng



# Histogram

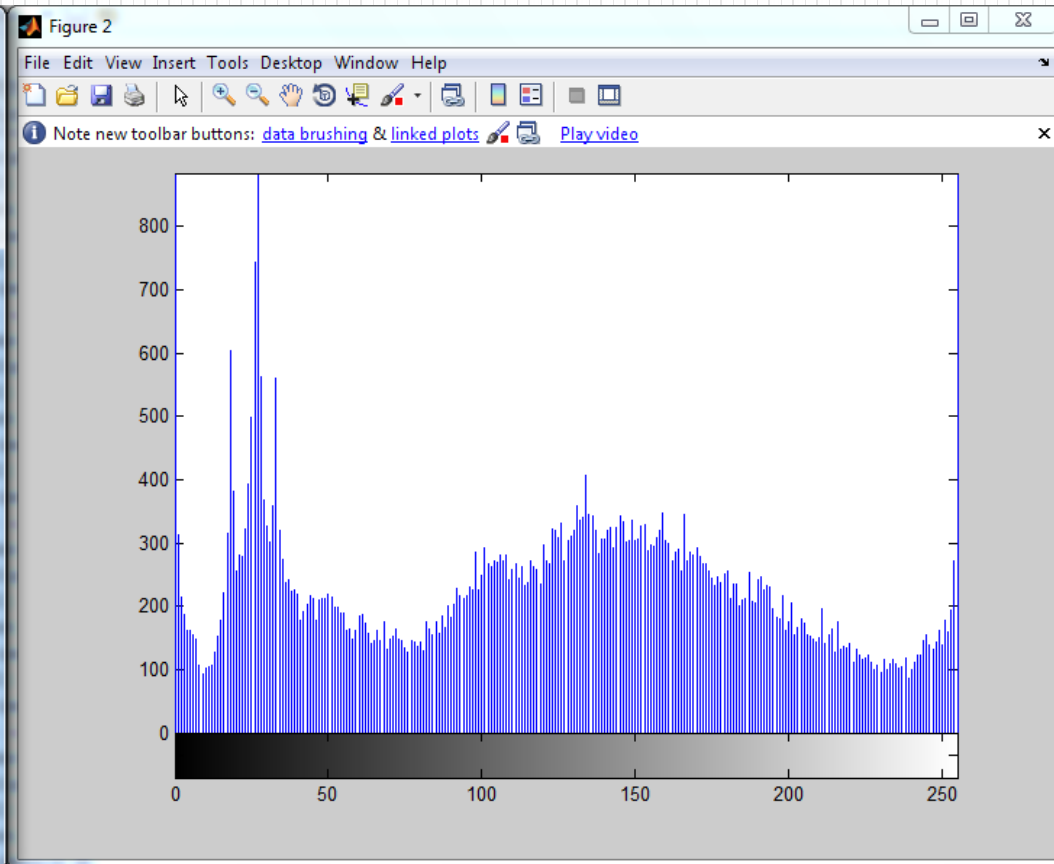
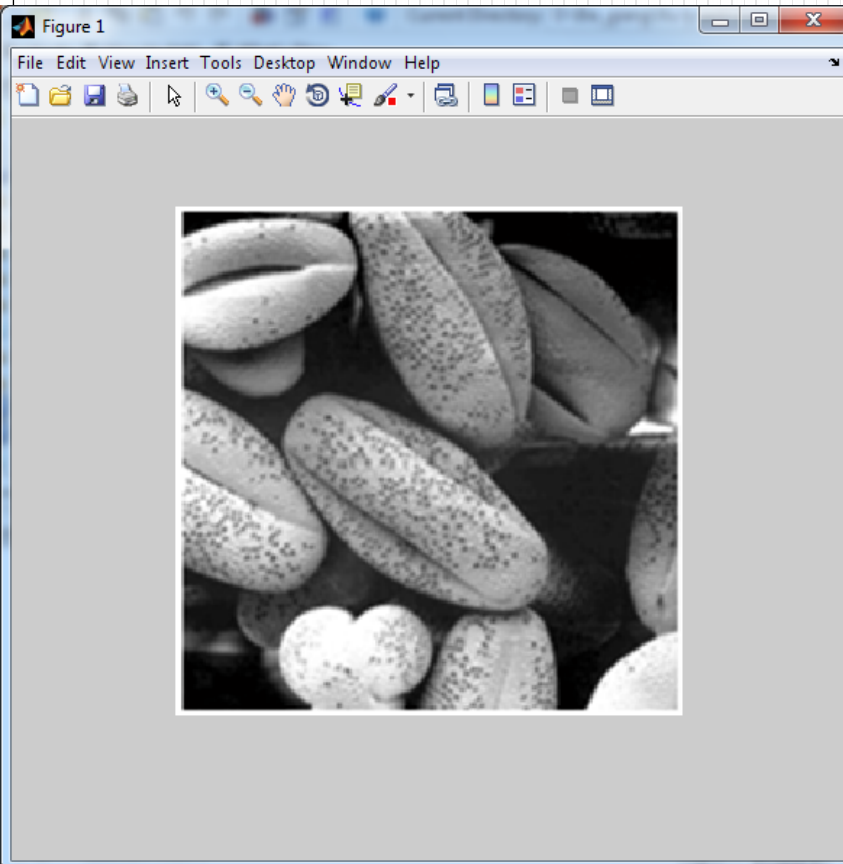
□ Ảnh độ tương phản thấp





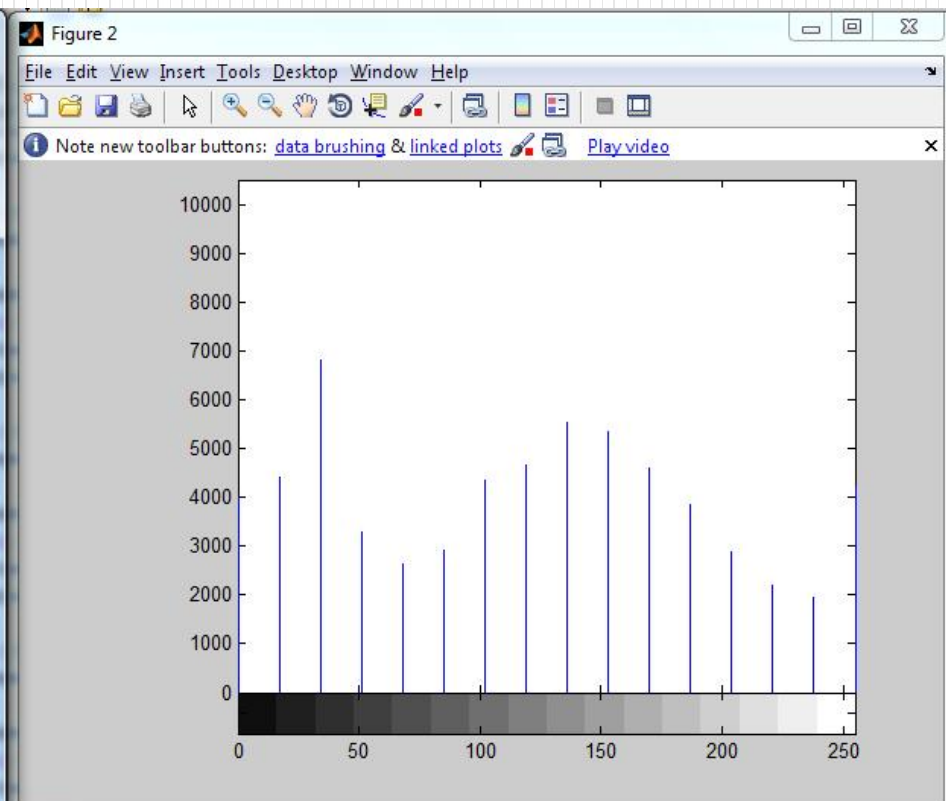
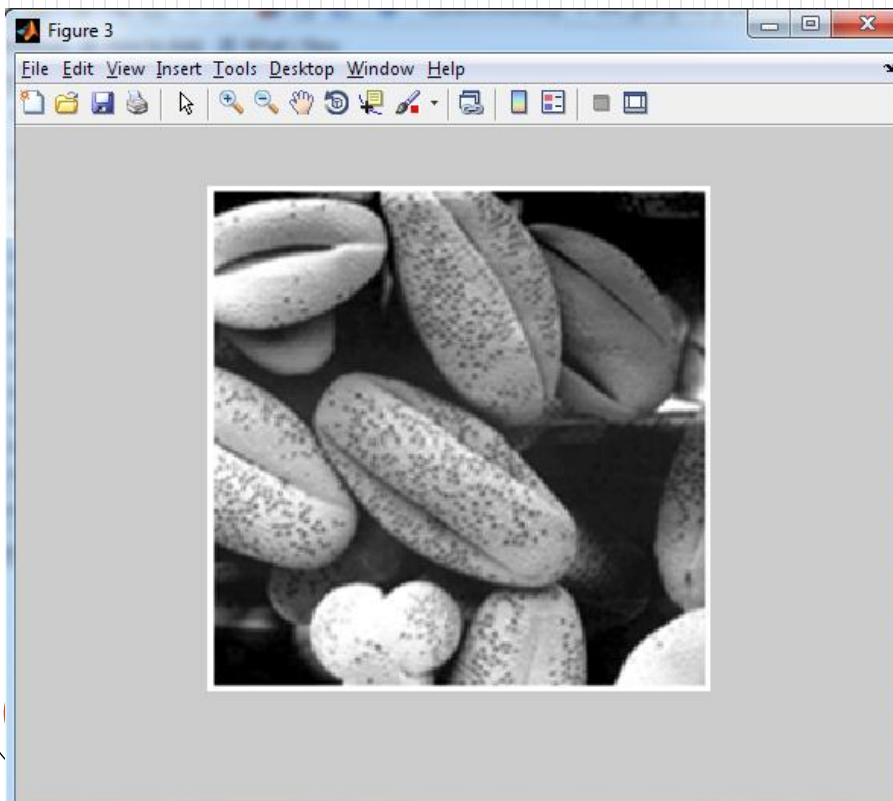
# Histogram

□ Ảnh độ tương phản cao



# Histogram

- ❑ Nhận xét?
- ❑ Khái niệm: số bins
  - Mặc định trong ảnh đa mức xám: 256 bins



# Phép cân bằng histogram

□ Histogram equalization: tác dụng?

□ Trước hết xét trên miền liên tục

- $r$ : biến ngẫu nhiên thể hiện các giá trị cấp xám trong ảnh ban đầu

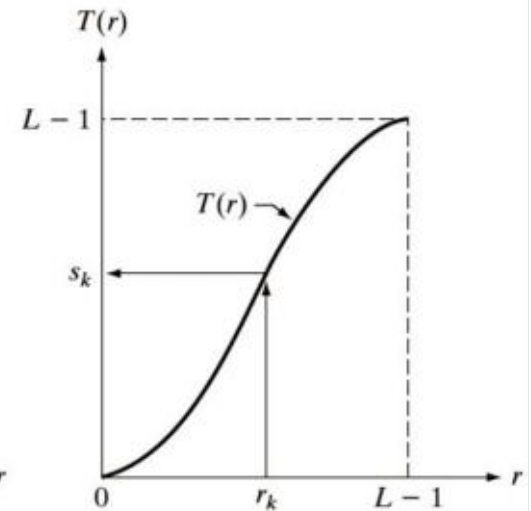
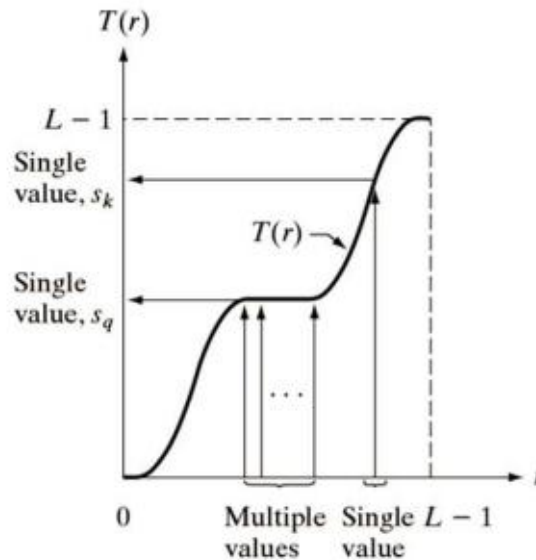
$$0 \leq r \leq 1$$

- $s$ : biến ngẫu nhiên thể hiện giá trị cấp xám trong ảnh biến đổi
- Cần tìm phép biến đổi:

$$s = T[r]$$

# Histogram equalization

$$s = T(r) : \begin{cases} T(0) = 0 \\ T(1) = 1 \\ T(r) \geq 0 \\ T'(r) \geq 0 \end{cases}$$



# Histogram equalization

□ Theo lý thuyết xác suất nếu tồn tại phép biến đổi ngược từ  $s \rightarrow r$  thì:

$$\begin{cases} s = T(r) \\ r = T^{-1}(s) \end{cases} \Rightarrow P_s(s) = P_r(r) \left| \frac{dr}{ds} \right|$$

□ Nếu chọn:

$$s = T(r) = (L-1) \int_0^r p_r(w) dw \Rightarrow \frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1) p_r(r)$$

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

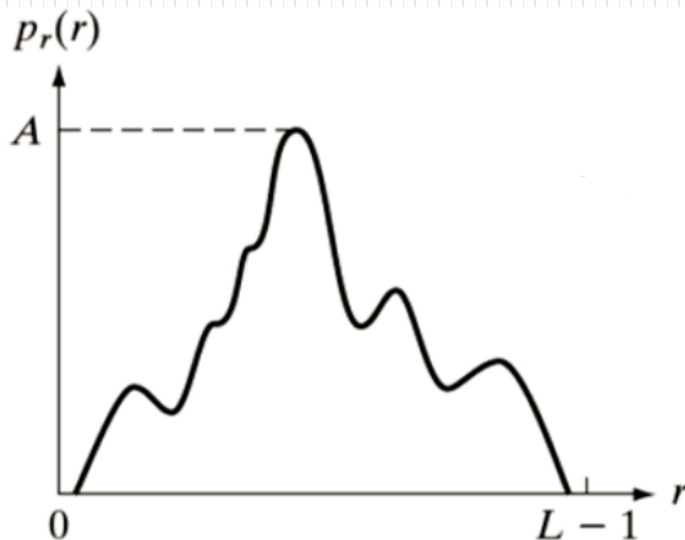
$\therefore$  Uniform Distribution  $[0, L-1]$

# Histogram equalization

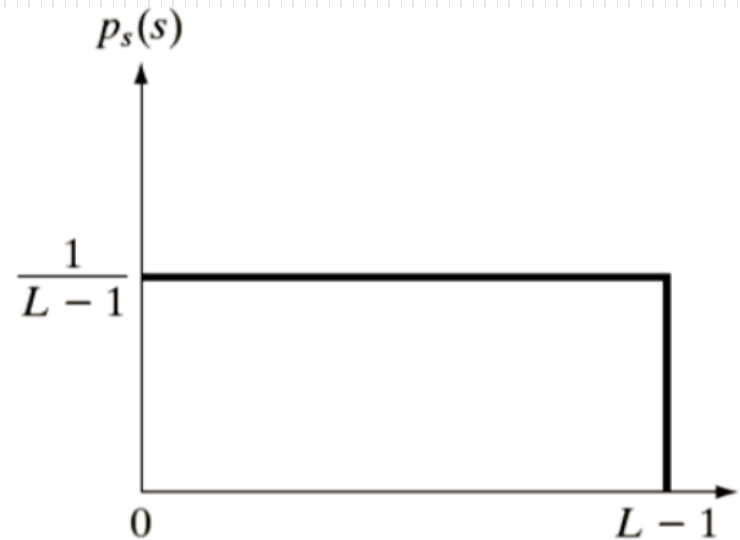
$$s = T(r) = (L-1) \int_0^r p_r(w) dw \Rightarrow \frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1) p_r(r)$$

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

$\therefore$  Uniform Distribution  $[0, L-1]$



$\rightarrow$



# Histogram equalization

□ Trên miền rời rạc (áp dụng cho ảnh số)

$$p_r(r_k) = \frac{n_k}{MN} = \frac{n_k}{n}, \quad k = 0, 1, 2, \dots, L-1$$

$$S_k = T(r_k) = (L-1) \sum_{j=0}^k p_r(r_j) = \frac{L-1}{MN} \sum_{j=0}^k n_j, \quad k = 0, 1, \dots, L-1$$

$$\hat{S}_k^1 = \lfloor S_k + 0.5 \rfloor = \text{round}(S_k)$$

$$\hat{S}_k^2 = \left\lfloor \frac{S_k - S_k^{\min}}{L-1 - S_k^{\min}} (L-1) + 0.5 \right\rfloor$$

# Histogram equalization

- Ảnh mới nhận được bằng cách ánh xạ mỗi pixel tại cấp xám  $r_k$  trong ảnh ban đầu với pixel tương ứng tại mức xám  $s_k$

$r_k$	$n_k$	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

Phân bố cường độ  
sáng của ảnh 3 bit  
dữ liệu: size 64 x 64



$s_k$	$\hat{s}_k$
1.33	1
3.08	3
4.55	5
5.67	6
6.23	6
6.56	7
6.86	7
7.00	7

$$S_k = 7 \sum_{j=0}^k p_r(r_j)$$



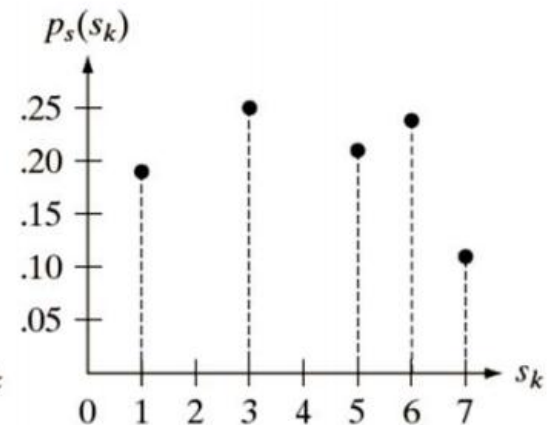
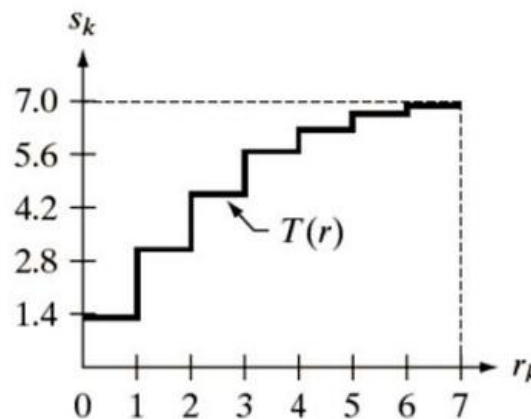
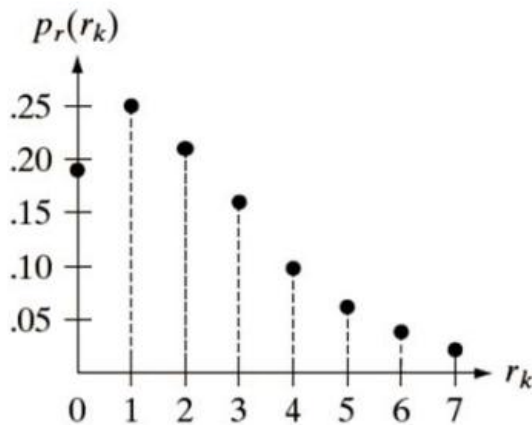
$r_k$	$n_k$	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

**Phân bố cường độ  
sáng của ảnh 3 bit  
dữ liệu: size 64 x 64**

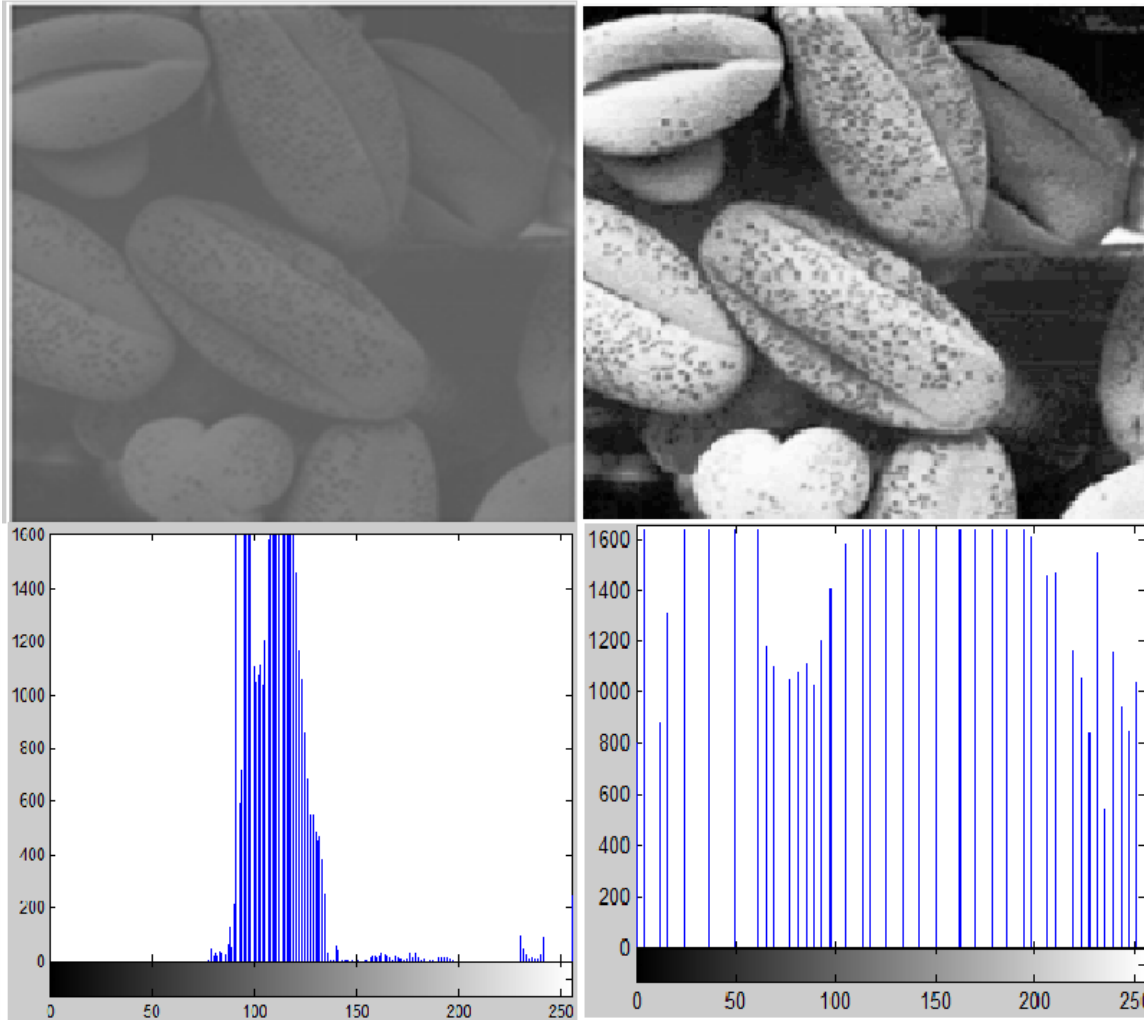


$S_k$	$\hat{S}_k$
1.33	1
3.08	3
4.55	5
5.67	6
6.23	6
6.56	7
6.86	7
7.00	7

$$S_k = 7 \sum_{j=0}^k p_r(r_j)$$



# Histogram equalization

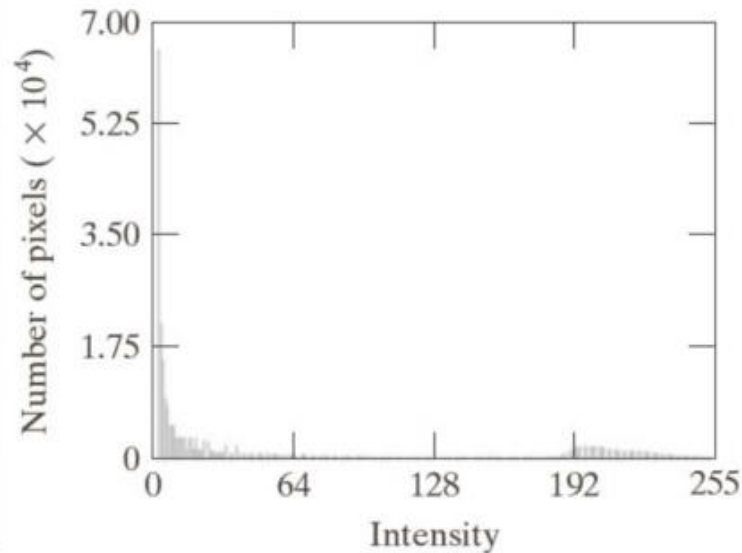


# Histogram equalization

## □ Thảo luận

- So sánh: Cân bằng histogram và phép biến đổi tăng độ tương phản bằng tuyến tính từng khúc
- Cân bằng histogram trường hợp nào cũng tốt?

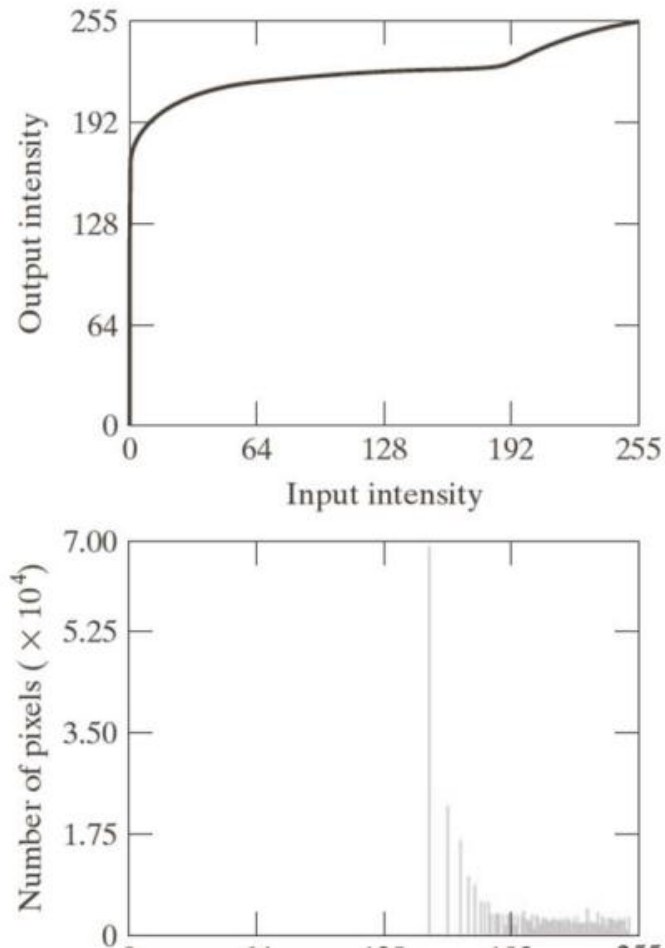
# Histogram equalization



a b

**FIGURE 3.23**  
(a) Image of the Mars moon Phobos taken by NASA's *Mars Global Surveyor*.  
(b) Histogram. (Original image courtesy of NASA.)

# Histogram equalization



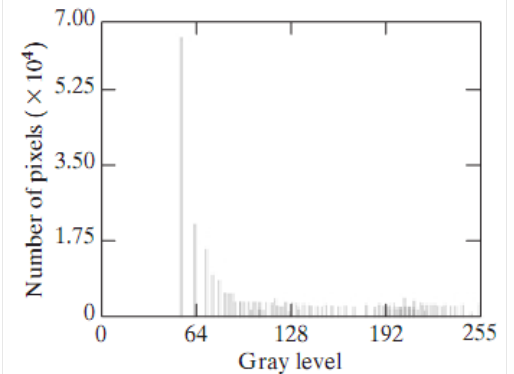
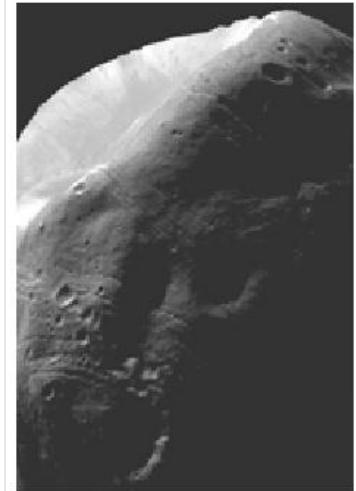
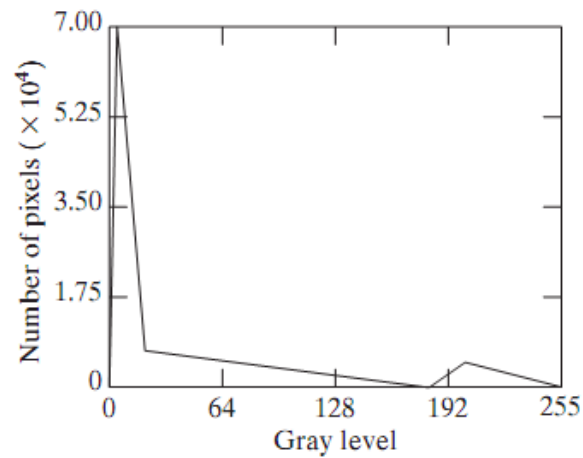
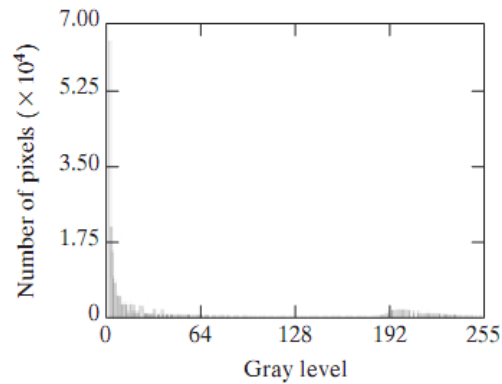
a b  
c

**FIGURE 3.24**  
(a) Transformation function for histogram equalization.  
(b) Histogram-equalized image (note the washed-out appearance).  
(c) Histogram of (b).

# Histogram matching/specification

- ❑ Nhiều trường hợp: histogram phân bố đều không cho kết quả ảnh tốt nhất
  - ❑ → Chỉ định histogram có hình dạng tốt
  - ❑ → Biến đổi ảnh theo hình dạng histogram cho trước
- Bài toán histogram matching hay histogram specification

# Histogram matching



# Histogram matching

□ Ý tưởng trên miền liên tục

$$p_r(r) \xrightarrow{?} p_z(z)$$

$$\left. \begin{aligned} s = T(r) &= (L-1) \int_0^r p_r(w) dw \\ G(z) &= (L-1) \int_0^z p_z(t) dt = \mathbf{s} \end{aligned} \right\} \Rightarrow z = G^{-1}[T(r)] = G^{-1}[s]$$



# Histogram matching

## □ Áp dụng trên miền rời rạc (ảnh số)

- Từ  $r_k$  tính  $s_k$ 
$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j)$$
$$= \sum_{j=0}^k \frac{n_j}{n} \quad k = 0, 1, 2, \dots, L - 1$$

- Từ  $z_k$  tính  $v_k$ 
$$v_k = G(z_k) = \sum_{i=0}^k p_z(z_i) = s_k \quad k = 0, 1, 2, \dots, L - 1$$

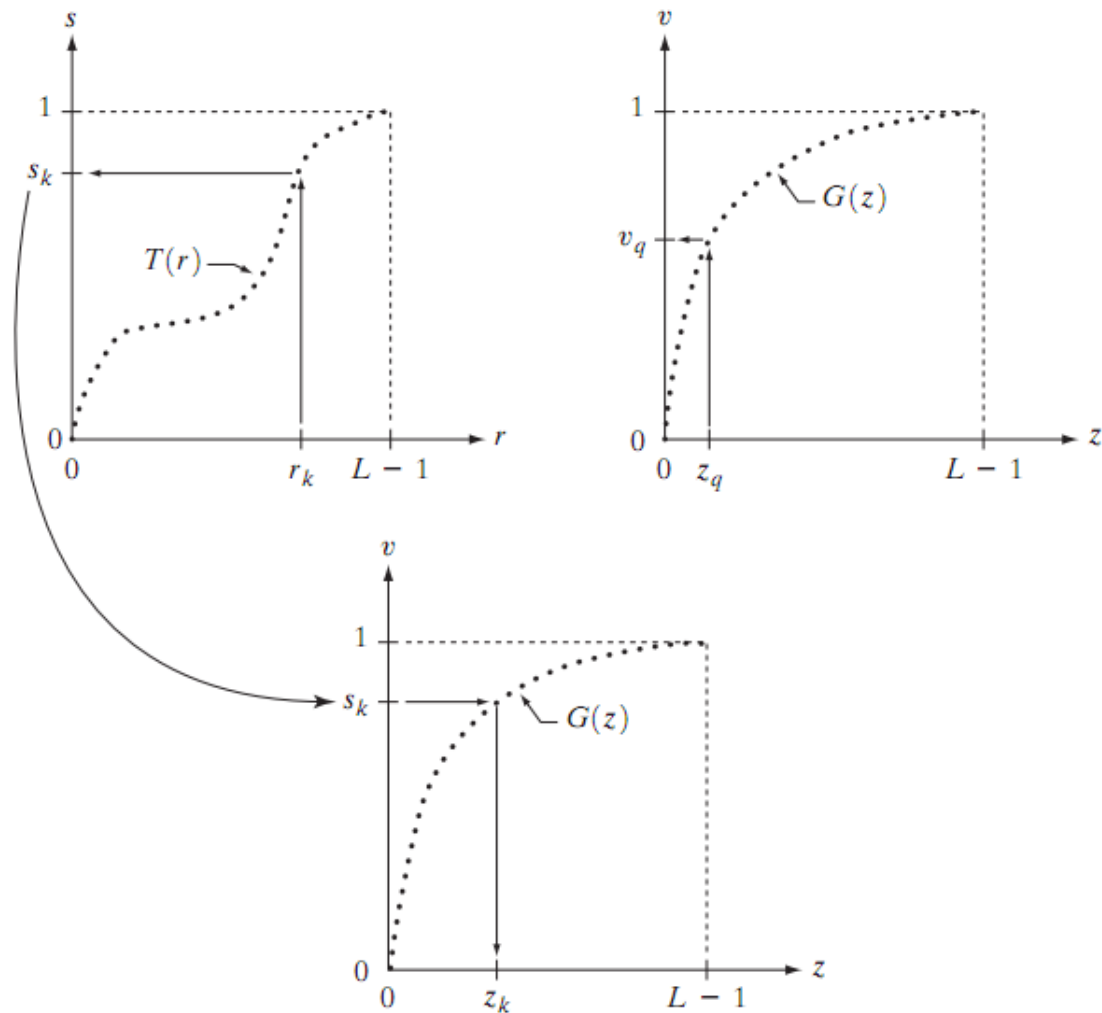
- Từ đó tính  $z_k$ 
$$z_k = G^{-1}[T(r_k)] \quad k = 0, 1, 2, \dots, L - 1$$
$$z_k = G^{-1}(s_k) \quad k = 0, 1, 2, \dots, L - 1.$$

# Histogram matching

a b  
c

**FIGURE 3.19**

(a) Graphical interpretation of mapping from  $r_k$  to  $s_k$  via  $T(r)$ .  
(b) Mapping of  $z_q$  to its corresponding value  $v_q$  via  $G(z)$ .  
(c) Inverse mapping from  $s_k$  to its corresponding value of  $z_k$ .



# Histogram matching

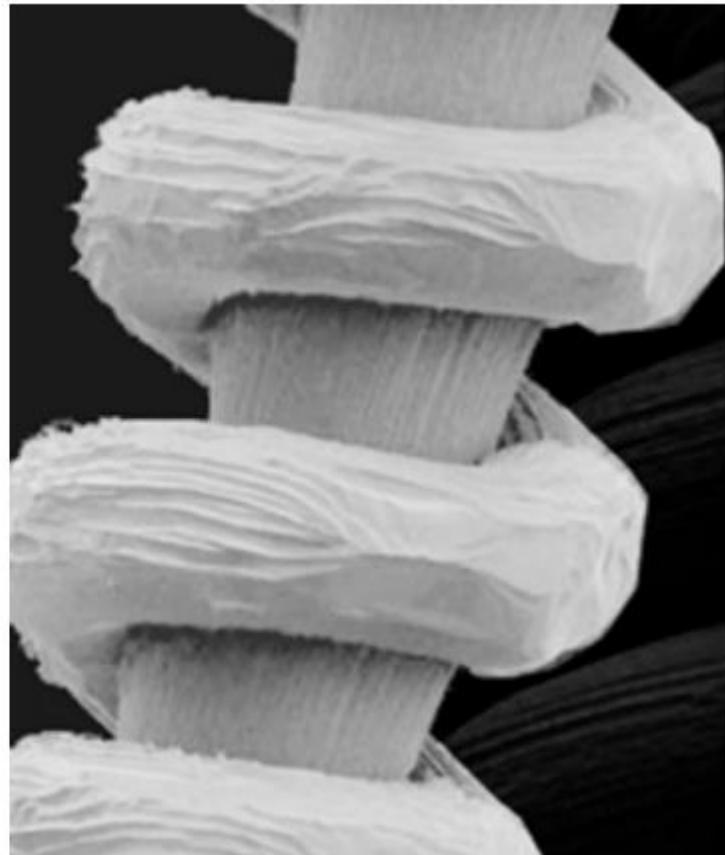
## □ Ứng dụng

- Hiệu chỉnh các ảnh chụp cùng một cảnh, nhưng được chụp bởi các camera, sensor khác nhau

# Cải thiện ảnh dựa trên các con số thống kê của histogram (histogram statistic)

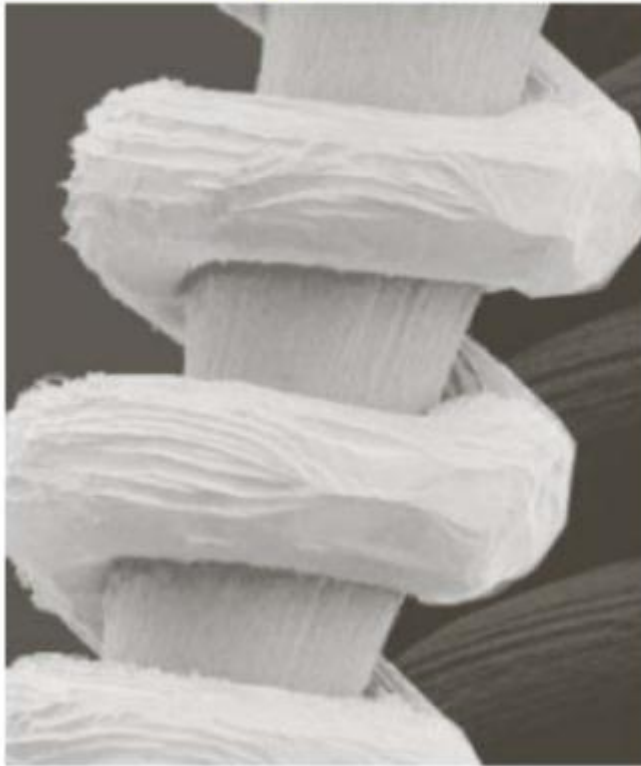
❑ Đặt vấn đề: Làm nổi vùng ảnh bên phải?

**FIGURE 3.24** SEM image of a tungsten filament and support, magnified approximately 130 $\times$ . (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene).

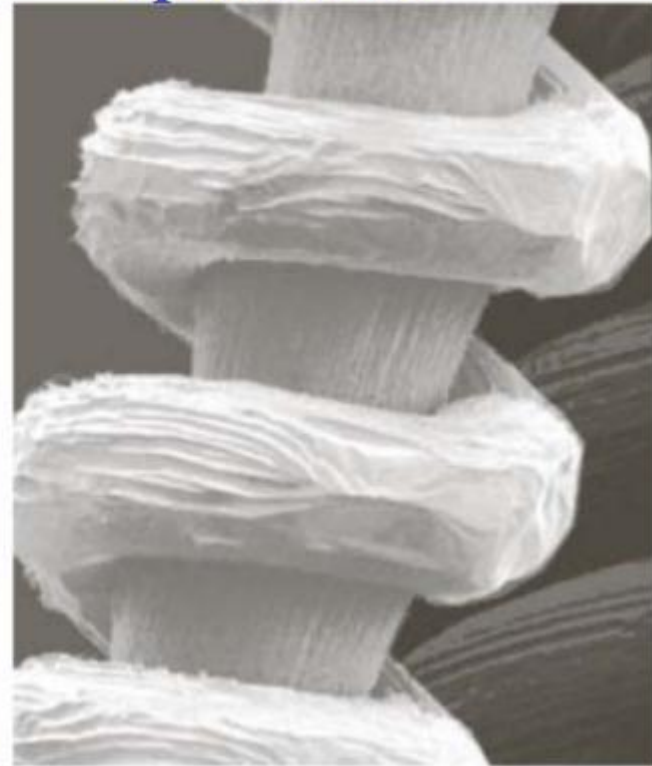


# Histogram statistic

Ảnh gốc



Ảnh sau histogram equalization



❑ Kết quả chưa tốt

# Histogram statistic

□ Tính toán các thống kê toàn cục

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i) \approx \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x, y) - m]^n$$

$$m = \sum_{i=0}^{L-1} r_i p(r_i) \approx \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N f(x, y)$$

□ Các thống kê cục bộ:  $S_{xy}$ : mặt nạ tâm (x,y)

$$m_{S_{xy}} = \sum_{(s,t) \in S_{xy}} r_{s,t} p(r_{s,t})$$

$$\sigma_{S_{xy}}^2 = \sum_{(s,t) \in S_{xy}} [r_{s,t} - m_{S_{xy}}]^2 p(r_{s,t}).$$

# Histogram statistic

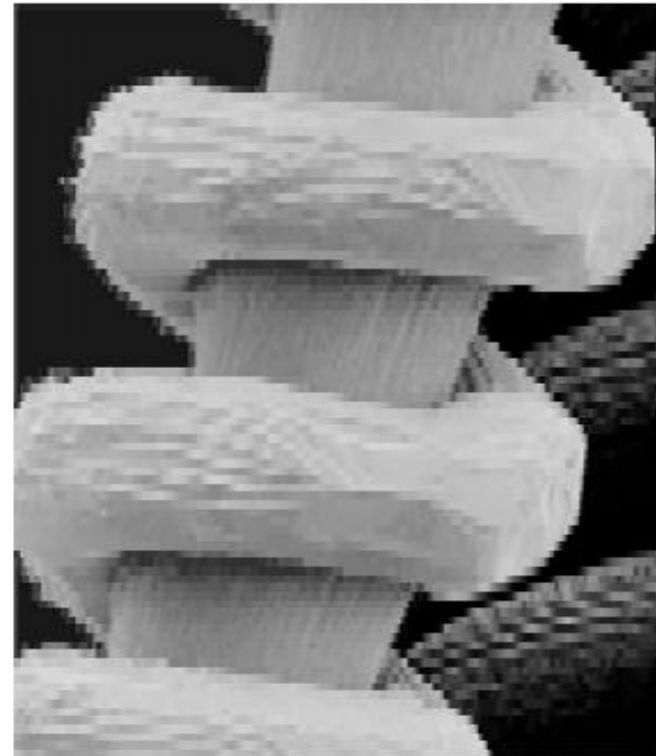
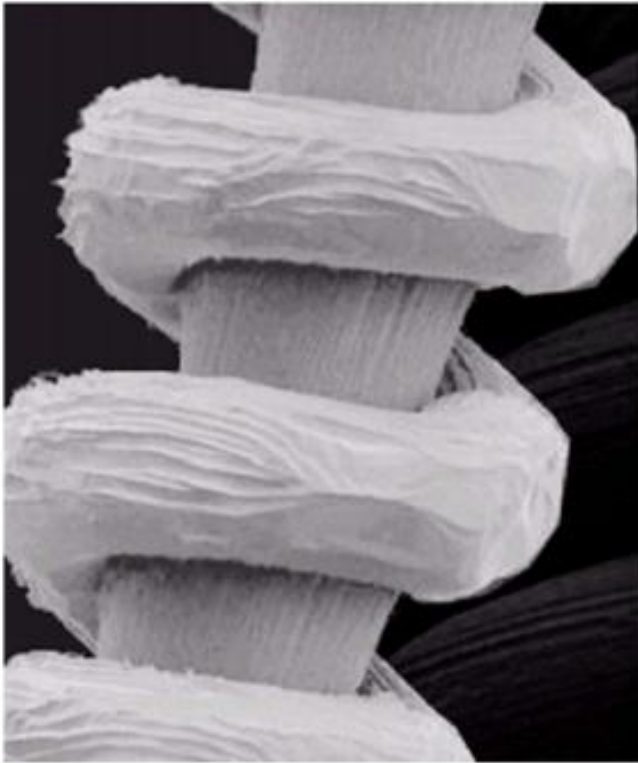
- ❑ Vấn đề: Làm sao để tăng độ tương phản trong một vùng ảnh mà không làm ảnh hưởng đến các vùng khác
- ❑ → Chỉ làm sáng, làm tối vùng cần thiết, các vùng còn lại giữ nguyên

$$g(x, y) = \begin{cases} E \cdot f(x, y) & m_s(x, y) \leq k_0 m_G \text{ and } k_1 \sigma_G \leq \sigma_s(x, y) \leq k_2 \sigma_G \\ f(x, y) & O.W \end{cases}$$

- Với  $k_0, k_1, k_2$  là các hằng số

# Histogram statistic

$$E = 4.0, k_0 = 0.4, k_1 = 0.02, k_2 = 0.4$$



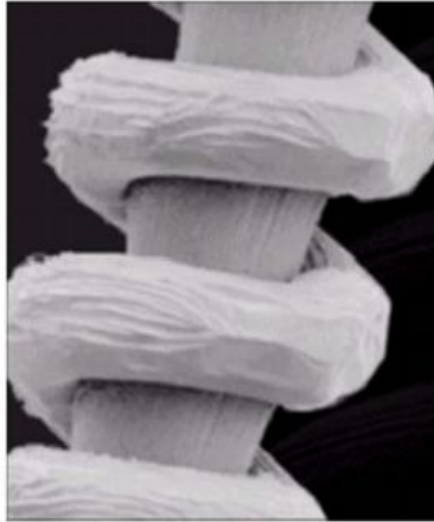


# Histogram statistic

Original image



Local Mean



Local Var



E or one



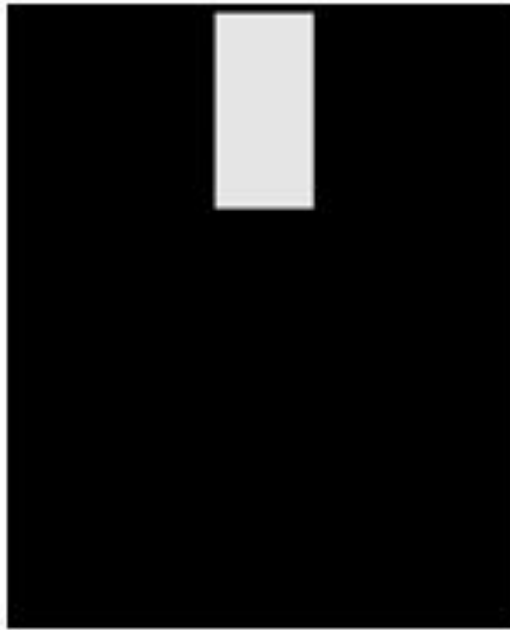
# Các phép biến đổi ảnh dựa trên các phép toán số học/logic

- ❑ Phép AND ảnh
- ❑ Phép OR ảnh
- ❑ Phép trừ ảnh
- ❑ Phép cộng ảnh

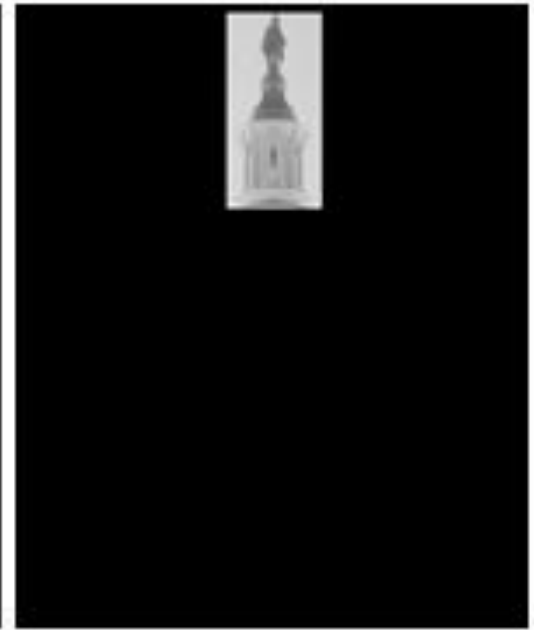
# Phép AND ảnh



Ảnh gốc



Mặt nạ AND

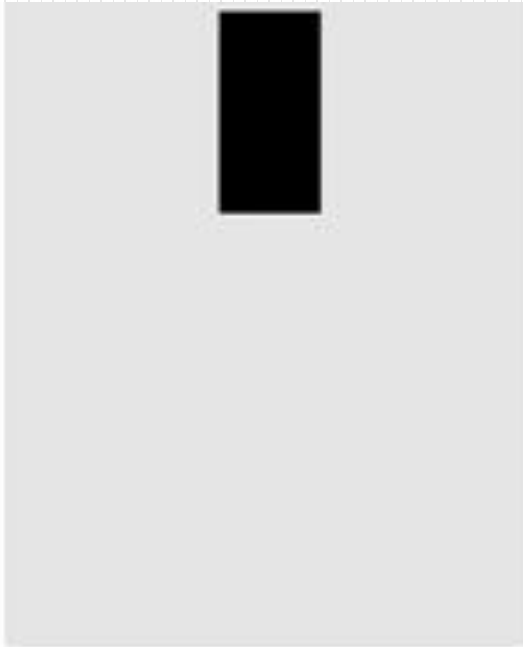


Phép AND ảnh

# Phép OR ảnh



Ảnh gốc



Mặt nạ OR



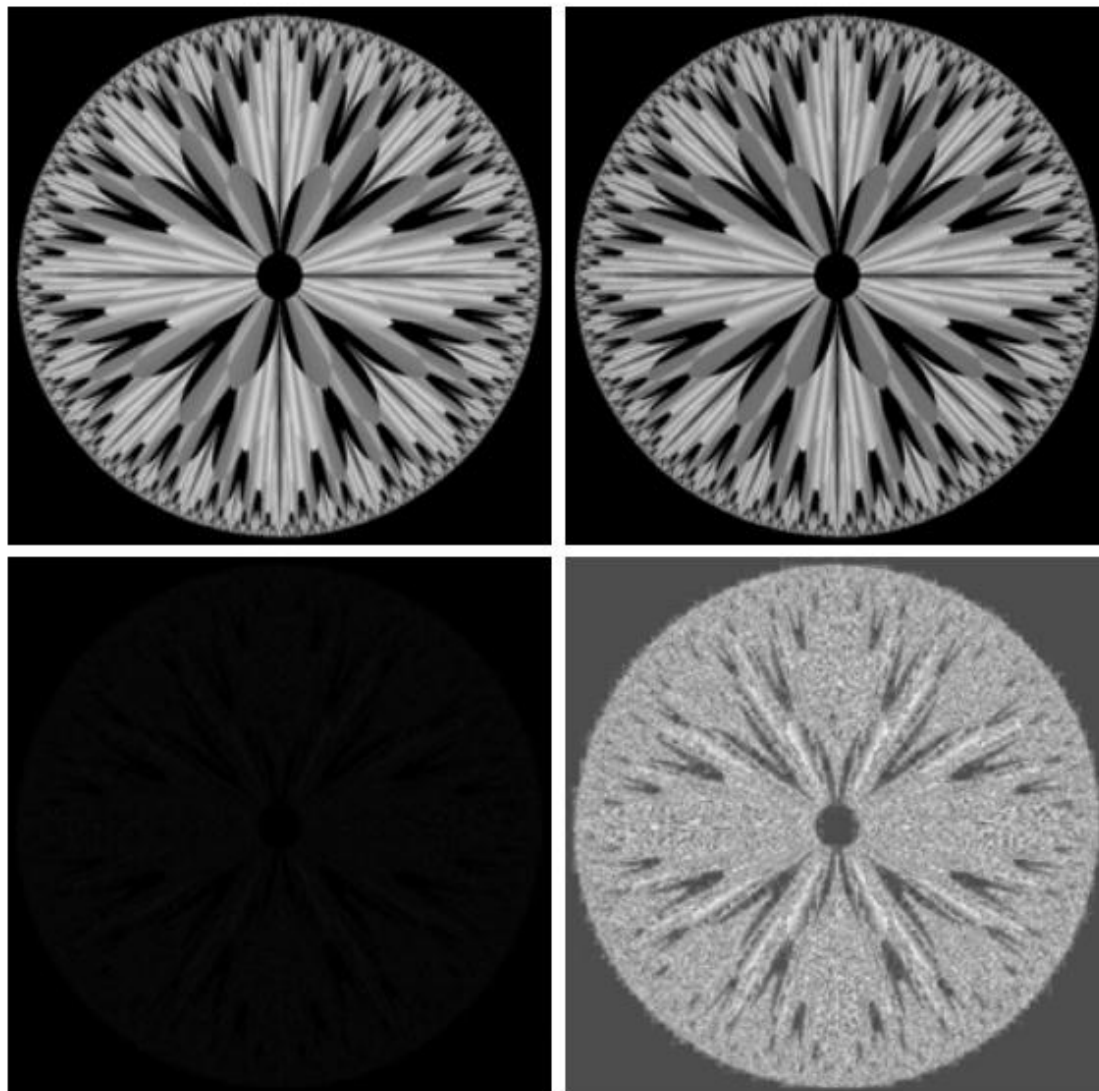
Phép OR ảnh

# Phép trừ ảnh

a b  
c d

**FIGURE 3.28**

(a) Original fractal image.  
(b) Result of setting the four lower-order bit planes to zero.  
(c) Difference between (a) and (b).  
(d) Histogram-equalized difference image. (Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA ).



# Trung bình ảnh

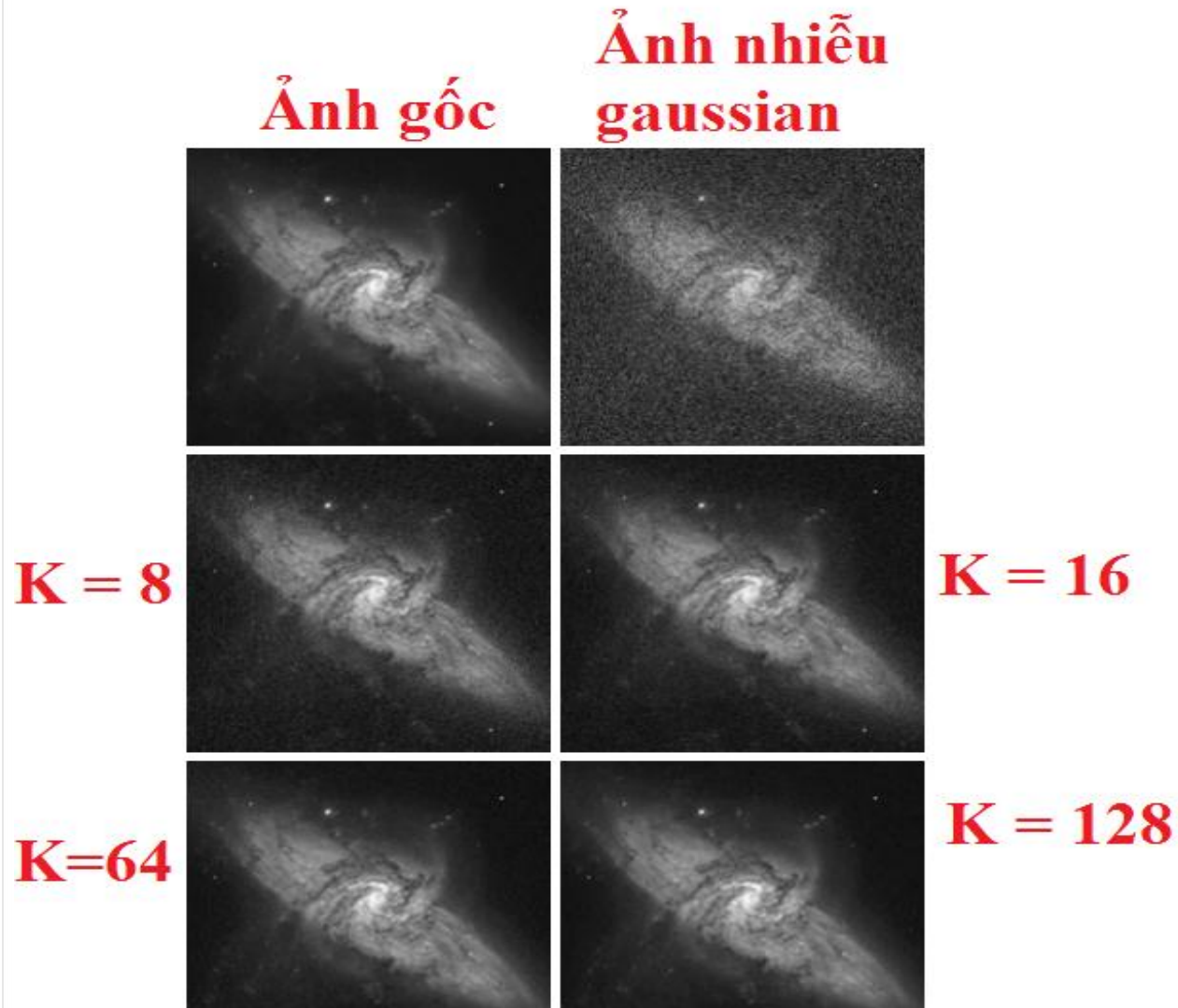
- $g(x,y)$  là ảnh nhiễu thu được bởi ảnh gốc  $f(x,y)$  và nhiễu  $\eta(x,y)$

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

- Với tập ảnh nhiễu  $\{g(x,y)\}$

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

# Phép trung bình ảnh



# Bài tập

□ Cho ảnh như sau: 8 bit – 8 x 8

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

- 1) Tính và vẽ histogram trong các trường hợp: 8 bins, 16 bins, 32 bins
- 2) Nhận xét về độ tương phản của ảnh trên
- 3) Cân bằng histogram cho ảnh trên,
  - Vẽ histogram sau khi cân bằng
  - Tính toán lại các giá trị điểm ảnh ứng với histogram mới



# Xử lý trên miền không gian

## □ Nếu xét cửa sổ lân cận: $1 \times 1$

- Phép xử lý trên điểm ảnh
- Giá trị đầu ra tại một điểm ảnh chỉ phụ thuộc điểm đó, không phụ thuộc vào các điểm khác

## □ Nếu xét cửa sổ lân cận $w \times w$

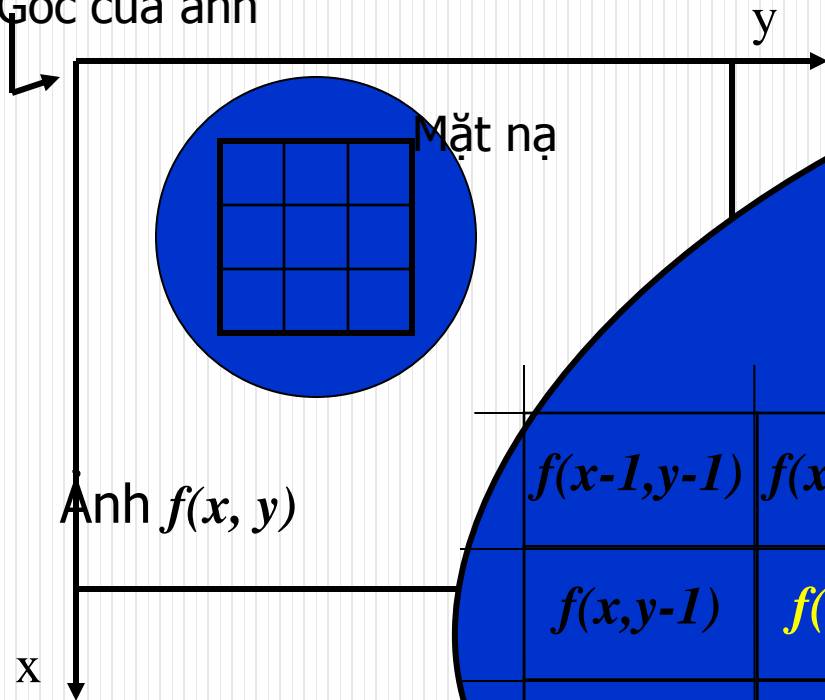
- Cửa sổ lân cận còn gọi là: mặt nạ (mask), nhân (kernel), Cửa sổ (window), bộ lọc (filter), template
- Giá trị đầu ra tại một điểm phụ thuộc vào các điểm lân cận của nó

# Biến đổi ảnh trên miền không gian

- ❑ Các phép lọc thường được liên tưởng đến lọc trên miền tần số
- ❑ Spatial filtering: lọc trên miền không gian – kết quả các phép lọc tác động trực tiếp đến các pixel trong ảnh
  - Các phép lọc làm trơn ảnh
  - Các phép lọc thống kê thứ tự
  - Các phép lọc tăng cường độ nét, cải thiện biên

# CƠ CHẾ CỦA LỌC KHÔNG GIAN

Gốc của ảnh



Mặt nạ

Ảnh  $f(x, y)$

x

y

$w(-1,-1)$   $w(-1,0)$   $w(-1,1)$

$w(0,-1)$   **$w(0,0)$**   $w(0,1)$

$w(1,-1)$   $w(1,0)$   $w(1,1)$

$f(x-1,y-1)$   $f(x-1,y)$   $f(x-1,y+1)$

$f(x,y-1)$   **$f(x,y)$**   $f(x,y+1)$

$f(x+1,y-1)$   $f(x+1,y)$   $f(x+1,y+1)$

$$R = w(-1,-1)f(x-1, y-1) + w(-1,0)f(x-1, y) + \dots \\ + w(0,0)f(x, y) + \dots + w(1,0)f(x+1, y) + w(1,1)f(x+1, y+1)$$

# Cơ chế của lọc không gian

□ Mặt nạ kích thước:  $m \times n$

- Thường:  $m = 2 \times a + 1$ ,  $n = 2 \times b + 1$
- $\rightarrow$  kích thước mặt nạ thường lẻ:  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ,  $9 \times 9 \dots$

- $$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

# Chú ý phân biệt: correlation với convolution

Spatial Correlation (☆) and Convolution (★)

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x-s, y-t)$$

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t)$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

# Ví dụ bộ lọc không gian 3 x 3

- ❑ Mặt nạ 3×3 trong hình trên đáp ứng tại điểm  $(x, y)$  của ảnh được tính bởi công thức

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 = \sum_{i=1}^9 w_i z_i$$

# XỬ LÝ TRƯỜNG HỢP ĐẶC BIỆT

- ❑ Trường hợp thực thi các phép toán lân cận khi tâm của bộ lọc nằm trên biên của ảnh?

$f(0,0)$	$f(0,1)$	$f(0,2)$	$f(0,3)$	$f(0,4)$
$f(1,0)$	$f(1,1)$	$f(1,2)$	$f(1,3)$	$f(1,4)$
$f(2,0)$	$f(2,1)$	$f(2,2)$	$f(2,3)$	$f(2,4)$
$f(3,0)$	$f(3,1)$	$f(3,2)$	$f(3,3)$	$f(3,4)$
$f(4,0)$	$f(4,1)$	$f(4,2)$	$f(4,3)$	$f(4,4)$

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

Mặt nạ

# XỬ LÝ TRƯỜNG HỢP ĐẶC BIỆT

- ❑ Khi tâm mặt nạ di chuyển gần đến biên của ảnh thì một hoặc một số dòng/cột của mặt nạ sẽ nằm ngoài ảnh.
- ❑ Ở ví dụ dưới các hệ số  $w_1$ ,  $w_4$ ,  $w_7$  nằm ngoài ảnh.

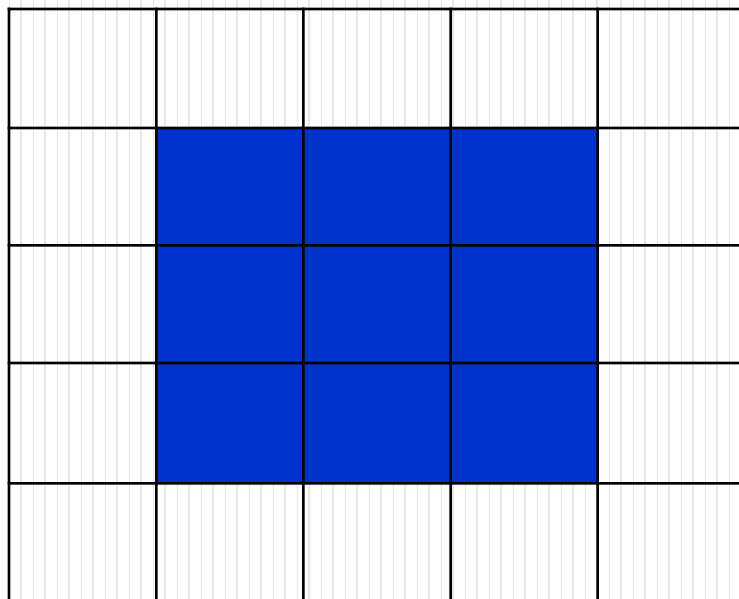
$w_1$	$w_2$	$w_3$			
$w_4$	$w_5$	$w_6$			
$w_7$	$w_8$	$w_9$			



# XỬ LÝ TRƯỜNG HỢP ĐẶC BIỆT

❑ **Cách 1:** Giả sử mặt nạ có kích thước  $n \times n$ .

- Cho vị trí tâm của mặt nạ không được nhỏ hơn  $(n-1)/2$  điểm ảnh kể từ biên → Ảnh sau khi lọc có kích thước nhỏ hơn ảnh gốc, nhưng tất cả các điểm ảnh đều được xử lý.



$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

# XỬ LÝ TRƯỜNG HỢP ĐẶC BIỆT

- ❑ **Cách 2:** Yêu cầu ảnh kết quả có kích thước bằng ảnh gốc
  - Đưa thêm các dòng đệm và cột đệm mang giá trị 0 vào quanh biên của ảnh.

0	0	0	0	0	0	0
0						0
0						0
0						0
0						0
0						0
0	0	0	0	0	0	0

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

# Bộ lọc làm trơn (smooth filtering)

## □ Tác dụng

- Làm mờ ảnh

- Áp dụng trong các phép tiền xử lý ảnh
- Loại bỏ các đối tượng nhỏ trong ảnh

- Khử nhiễu

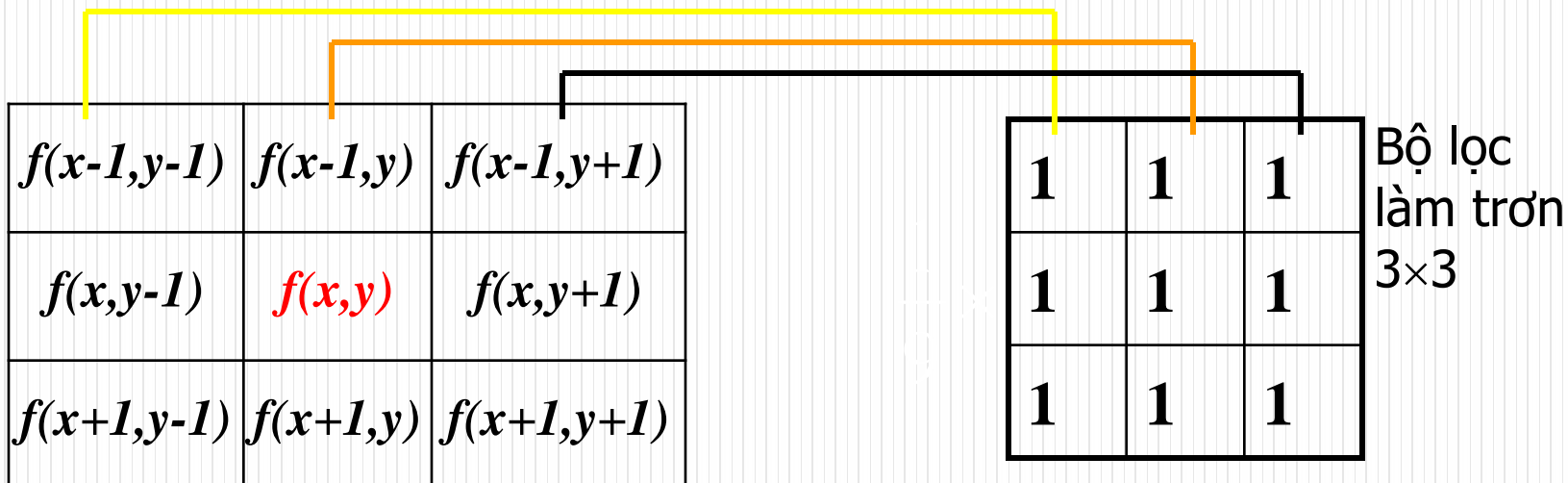
## □ Bộ lọc làm trơn đơn giản nhất là bộ lọc trung bình (average filtering)

# Bộ lọc làm trơn ảnh

- Ý tưởng bộ lọc trung bình: thay thế giá trị tại mỗi pixel bằng trung bình các giá trị pixel trong mặt nạ lân cận
  - → loại bỏ những pixel biến đổi lớn so với lân cận (nhiều)
  - → những pixel nằm trên biên cũng có sự biến đổi lớn so với lân cận → làm mờ

# Bộ lọc làm trơn ảnh

- Với bộ lọc không gian kích thước  $3 \times 3$ , thì cách sắp xếp đơn giản nhất là cho các hệ số bằng  $1/9$ .



$$g(x, y) = \frac{1}{9} [f(x-1, y-1) + f(x-1, y) + f(x-1, y+1) + f(x, y-1) + f(x, y) + f(x, y+1) + f(x+1, y-1) + f(x+1, y) + f(x+1, y+1)]$$

# Bộ lọc làm trơn ảnh

- ❑ Bộ lọc với các hệ số khác nhau (bộ lọc có trọng số)
- ❑ Ta thấy hệ số trung tâm của bộ lọc có giá trị lớn hơn so với các hệ số khác → điểm ảnh ứng với tâm bộ lọc có tầm quan trọng hơn các điểm ảnh khác.

$$g(x, y) = \frac{1}{16} [f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1) + 2f(x, y-1) + 4f(x, y) + 2f(x, y+1) + f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)]$$

1	2	1
2	4	2
1	2	1

Bộ lọc làm trơn 3×3

# Bộ lọc làm trơn ảnh

- Công thức tổng quát của bộ lọc trung bình

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

# Bộ lọc làm trơn ảnh

□ Ví dụ: Lọc trung bình

0	0	0	0	0	0	0	0	0
0	10	15	15	45	25	14	23	0
0	12	14	255	12	25	12	45	0
0	25	56	25	12	45	255	12	0
0	14	48	98	51	12	15	20	0
0	12	32	36	34	25	26	24	0
0	12	14	5	7	54	12	51	0
0	14	56	25	14	20	47	12	0
0	0	0	0	0	0	0	0	0

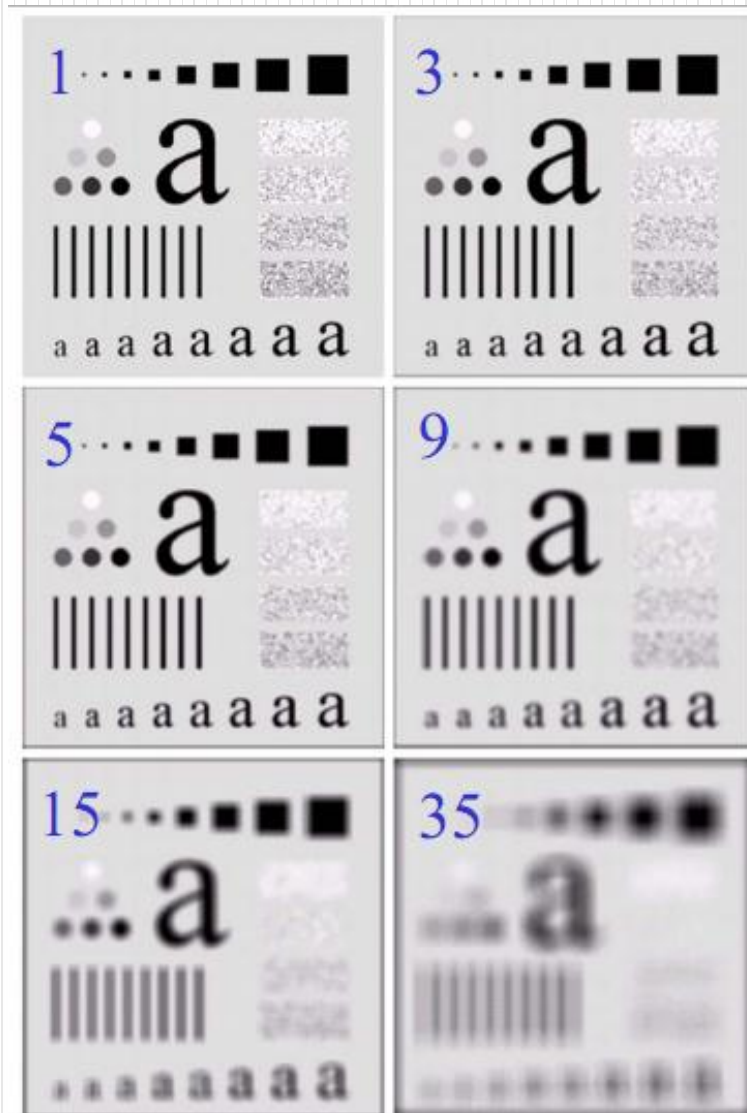
1	1	1
1	1	1
1	1	1

=RC	36	40	42	15	16	10
15	47	50	51	49	51	40
19	61	63	59	49	49	40
21	38	44	38	53	48	39
15	30	36	36	26	27	16
16	23	25	24	27	30	19
11	14	13	14	17	22	14



# Bộ lọc làm trơn ảnh

- Minh họa lọc trung bình với các kích thước mặt nạ khác nhau

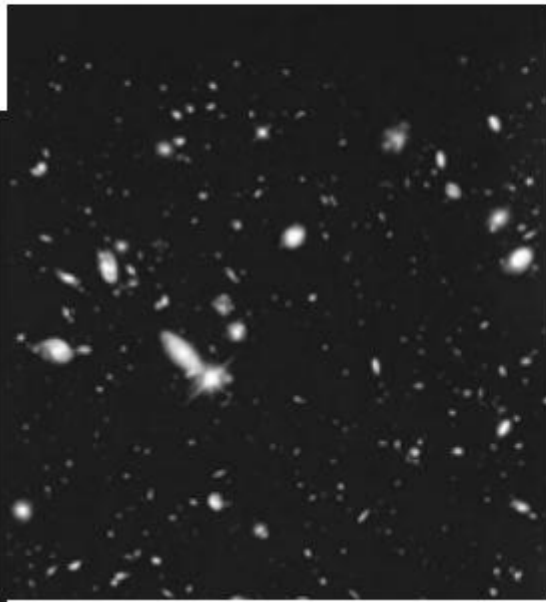


# Bộ lọc làm trơn ảnh

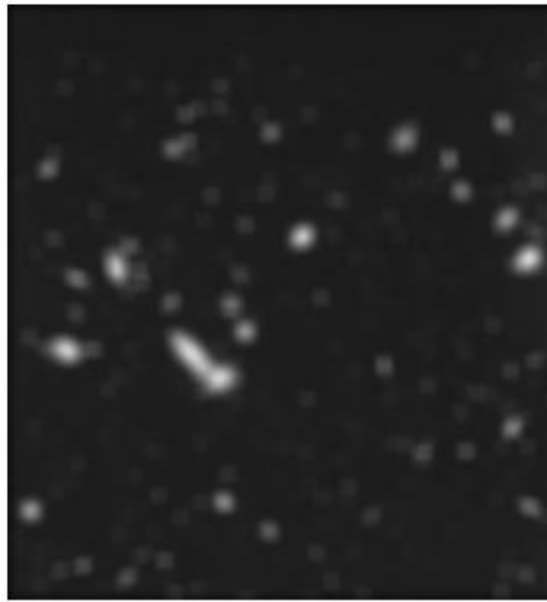
## □ Ví dụ lọc làm mờ ảnh

- Tác dụng: loại bỏ các đối tượng nhỏ

Ảnh gốc



Ảnh blurring



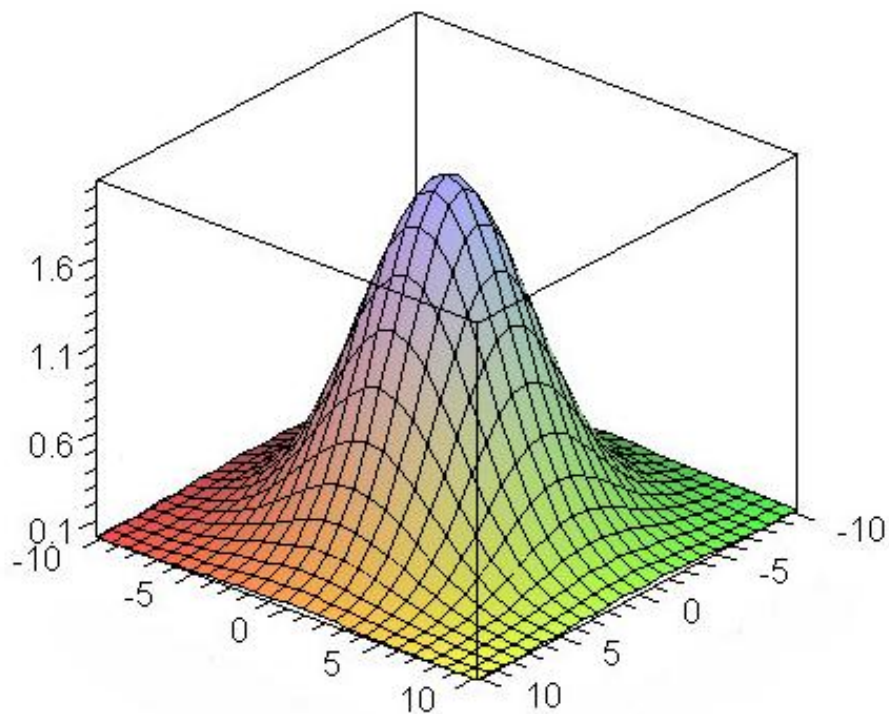
Thresholding của  
ảnh sau làm mờ



# Bộ lọc làm trơn ảnh

## □ Bộ lọc Gaussian

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



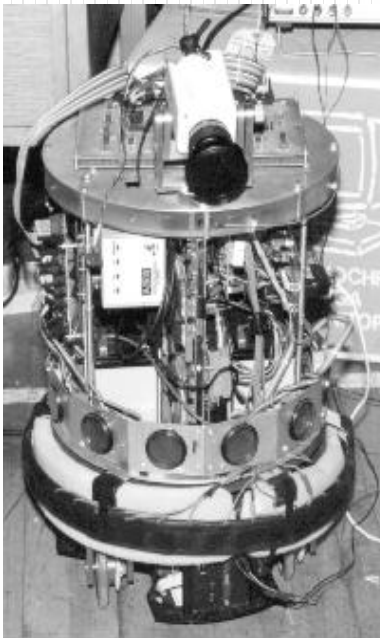
# Bộ lọc làm trơn ảnh

□ Mặt nạ gaussian với  $\sigma = 1.0$

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

# Bộ lọc làm trơn ảnh



**sigma = 1.0 (and  
kernel size 5×5).**



**sigma = 2.0 (and  
kernel size 9×9).**



**(sigma = 4.0 and  
kernel size 15×15)**

# Lọc thống kê thứ tự (order-statistic filtering)

## □ Ý tưởng:

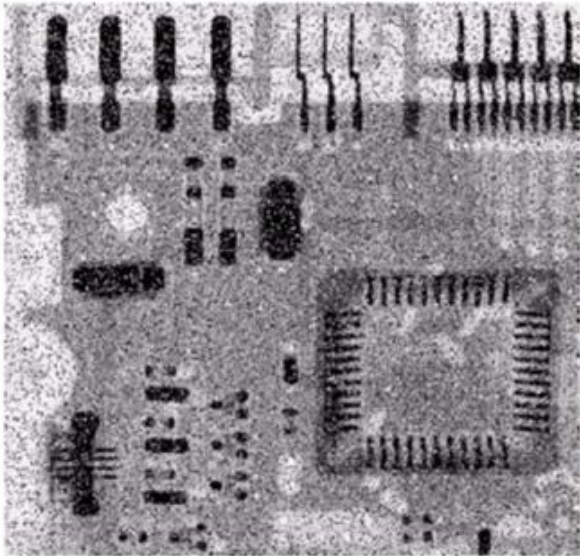
- Các pixel trong mặt nạ lân cận được sắp xếp theo thứ tự (ranking)
- Thay thế pixel trung tâm bằng giá trị của một pixel chỉ định trong danh sách
  - → các loại lọc thống kê thứ tự
    - Lọc trung vị
    - Lọc cực đại (max)
    - Lọc cực tiểu (min)

# Lọc trung vị

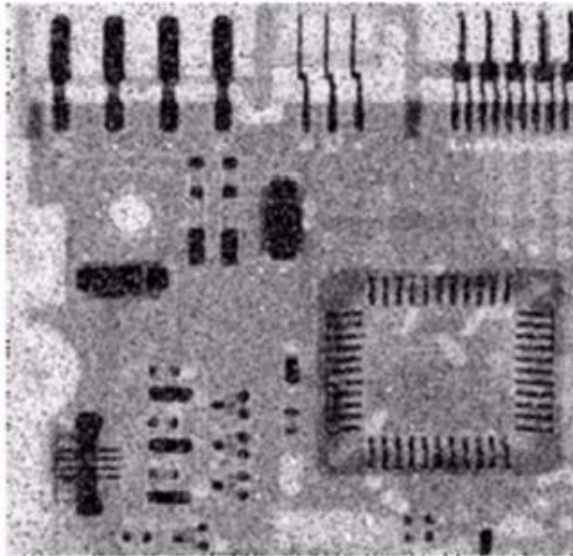
- ❑ Được sử dụng phổ biến:
  - Khử nhiễu ngẫu nhiên tốt
  - Ít bị nhòe hơn so với phép lọc trung bình
- ❑ Cài đặt
  - Sắp xếp các giá trị pixel trong cửa sổ lân cận
  - Thay thế giá trị pixel bằng giá trị giữa của danh sách
- ❑ Ví dụ: Cho dãy
  - $\{x_n\} = \{3, 4, 6, 29, 4, 30, 40, 30, 5\}$
  - $\{x_n\} = \{3, 4, 4, 5, 6, 29, 30, 30, 40\}$ .
  - $\rightarrow$  trung vị = 6

# Lọc trung vị

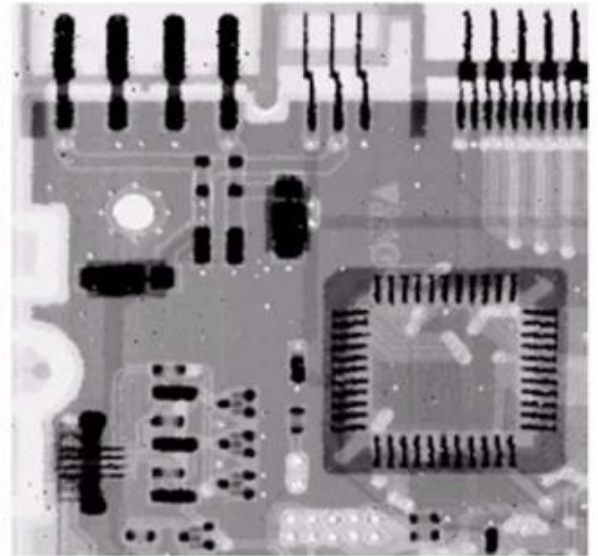
Salt-Pepper Noise



$3 \times 3$  Averaging



$3 \times 3$  Median





# Các phép lọc tăng cường độ nét, cải thiện biên (sharpening filter)

- ❑ Mục đích của lọc tăng cường độ nét là để làm nổi bật một số chi tiết trong ảnh các điểm ảnh có sự biến đổi mức xám lớn so với lân cận
- ❑ Lọc làm nổi bật đường biên của ảnh
- ❑ Cơ sở của các phép lọc tăng cường độ nét và cải thiện biên là các phép đạo hàm cấp 1 và cấp 2

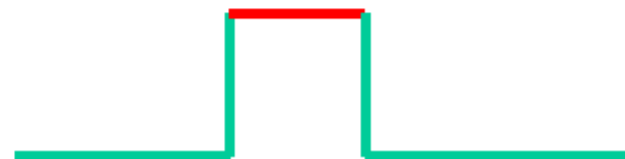
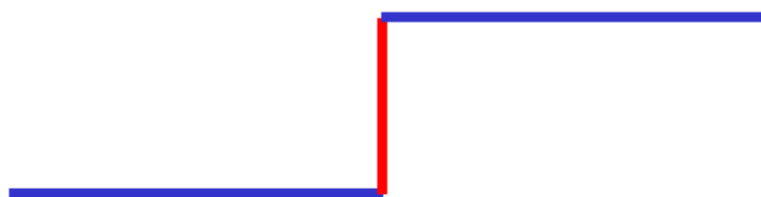
# Đạo hàm các cấp áp dụng cho ảnh số

$$\frac{\partial f}{\partial x} \approx \begin{cases} f(x+1, y) - f(x, y) \\ f(x, y) - f(x-1, y) \\ 0.5(f(x+1, y) - f(x-1, y)) \end{cases}$$

$$\frac{\partial^2 f}{\partial x^2} \approx f(x+1, y) - 2f(x, y) + f(x-1, y)$$

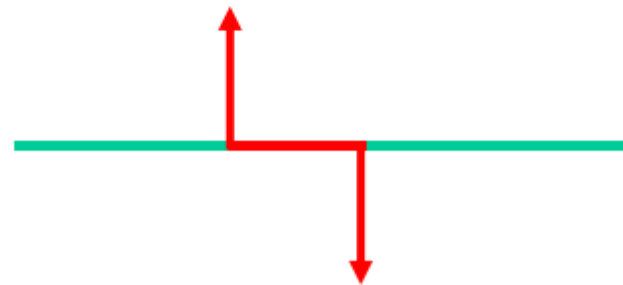
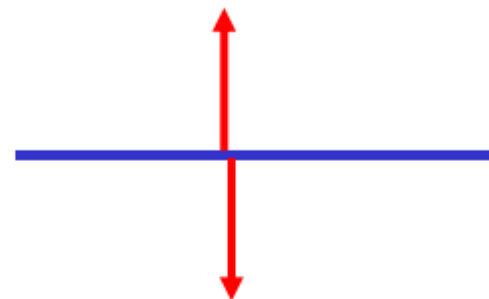
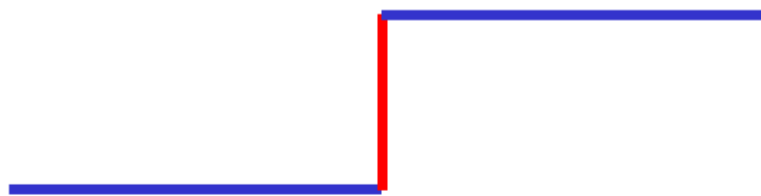
# Đạo hàm cấp 1

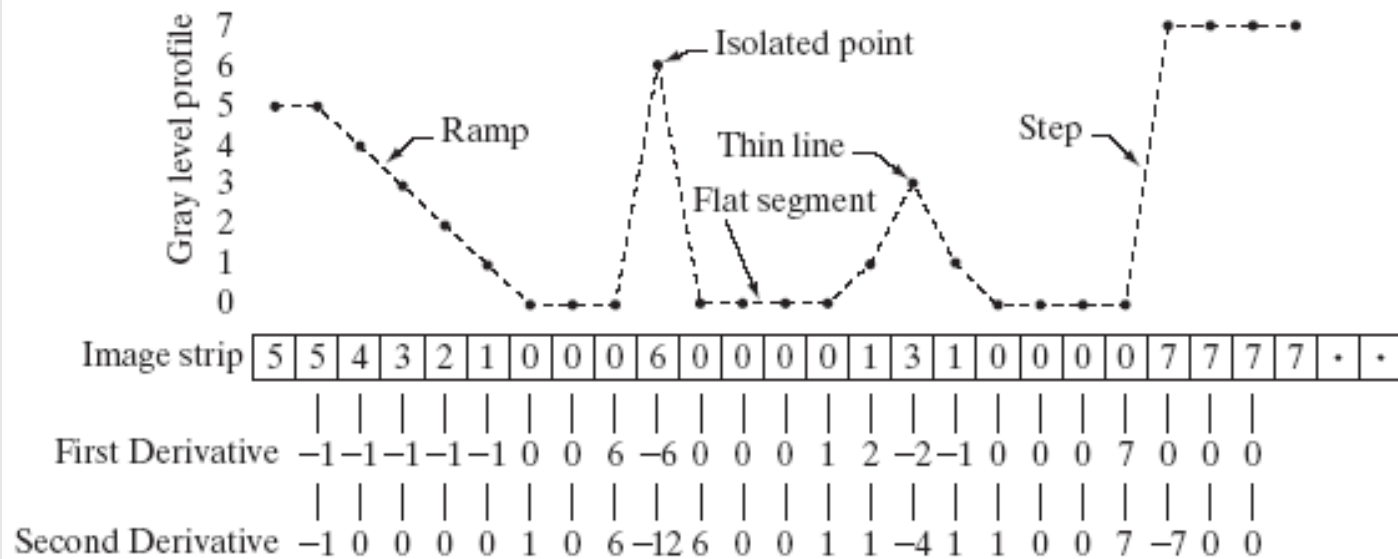
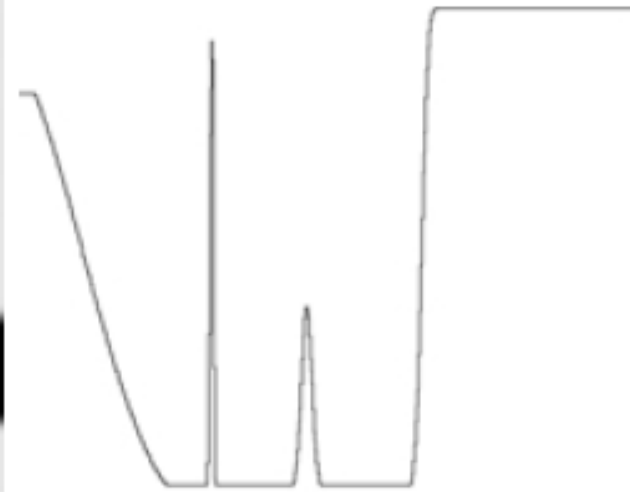
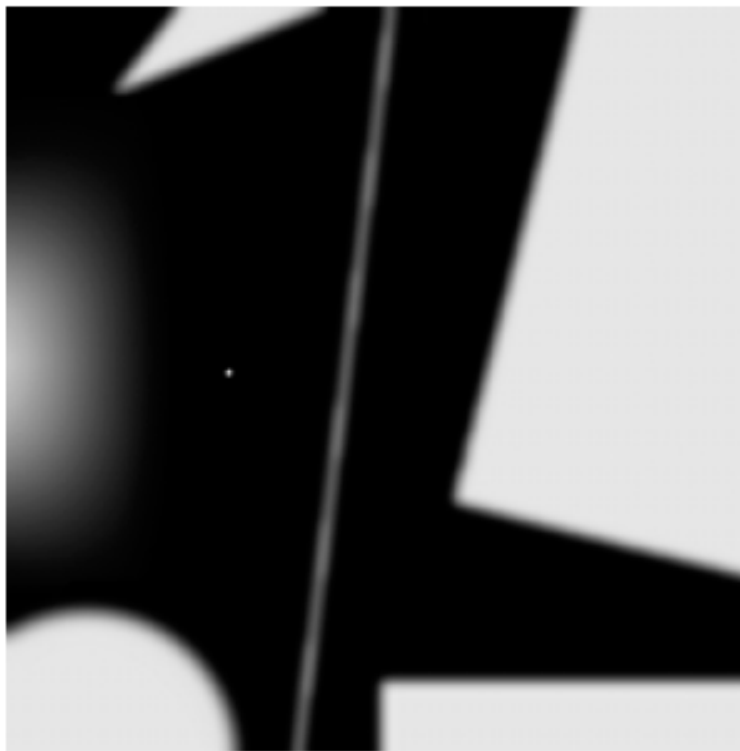
- Bằng 0 tại những vùng không biến đổi
- Khác 0 tại điểm bắt đầu đường “dốc”
- Khác 0 trên vùng “dốc”



# Đạo hàm cấp 2

- Bằng 0 tại những vùng không biến đổi
- Khác 0 tại những điểm bắt đầu/kết thúc vùng “dốc”
- Bằng 0 tại những điểm trên vùng “dốc”





# Đạo hàm các cấp

## □ Nhận xét:

### ■ Đạo hàm cấp 1

- Cho biên dày hơn
- Nhạy với các bước nhảy mức xám

### ■ Đạo hàm bậc 2

- Đáp ứng mạnh với các chi tiết, các điểm độc lập
- Đáp ứng gấp đôi tại các bước nhảy mức xám

# SỬ DỤNG ĐẠO HÀM BẬC HAI TOÁN TỬ LAPLACIAN

- Toán tử đạo hàm đẳng hướng là toán tử Laplacian đối với ảnh  $f(x, y)$  được định nghĩa như sau:

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

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

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

# SỬ DỤNG ĐẠO HÀM BẬC HAI TOÁN TỬ LAPLACIAN

- Theo công thức trên ta có thể xây dựng mặt nạ lọc L như sau:

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

0	1	0
1	-4	1
0	1	0

- Mặt nạ này bất biến với phép quay 90°.



# SỬ DỤNG ĐẠO HÀM BẬC HAI TOÁN TỬ LAPLACIAN

- Theo công thức trên ta có thể xây dựng mặt nạ lọc L như sau:

$$\nabla^2 f = 4f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)]$$

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

- Mặt nạ này bất biến với phép quay 90°.

# SỬ DỤNG ĐẠO HÀM BẬC HAI TOÁN TỬ LAPLACIAN

- Chúng ta cũng có thể xem xét đạo hàm bậc hai theo đường chéo, khi đó mặt nạ lọc L thu được là:

$$\nabla^2 f = [f(x+1, y+1) + f(x+1, y) + f(x+1, y-1) + f(x-1, y+1) + f(x-1, y) + f(x-1, y-1) + f(x, y+1) + f(x, y-1)] - 8f(x, y)$$

1	1	1
1	-8	1
1	1	1

- Mặt nạ này bất biến với phép quay 45°.

# SỬ DỤNG ĐẠO HÀM BẬC HAI TOÁN TỬ LAPLACIAN

- Chúng ta cũng có thể xem xét đạo hàm bậc hai theo đường chéo, khi đó mặt nạ lọc L thu được là:

$$\nabla^2 f = 8f(x, y) - [f(x+1, y+1) + f(x+1, y) + f(x+1, y-1) + f(x-1, y+1) + f(x-1, y) + f(x-1, y-1) + f(x, y+1) + f(x, y-1)]$$

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

- Mặt nạ này bất biến với phép quay 45°.

# Cải thiện ảnh với toán tử Laplacian

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & -sign \\ f(x, y) + \nabla^2 f(x, y) & +sign \end{cases}$$

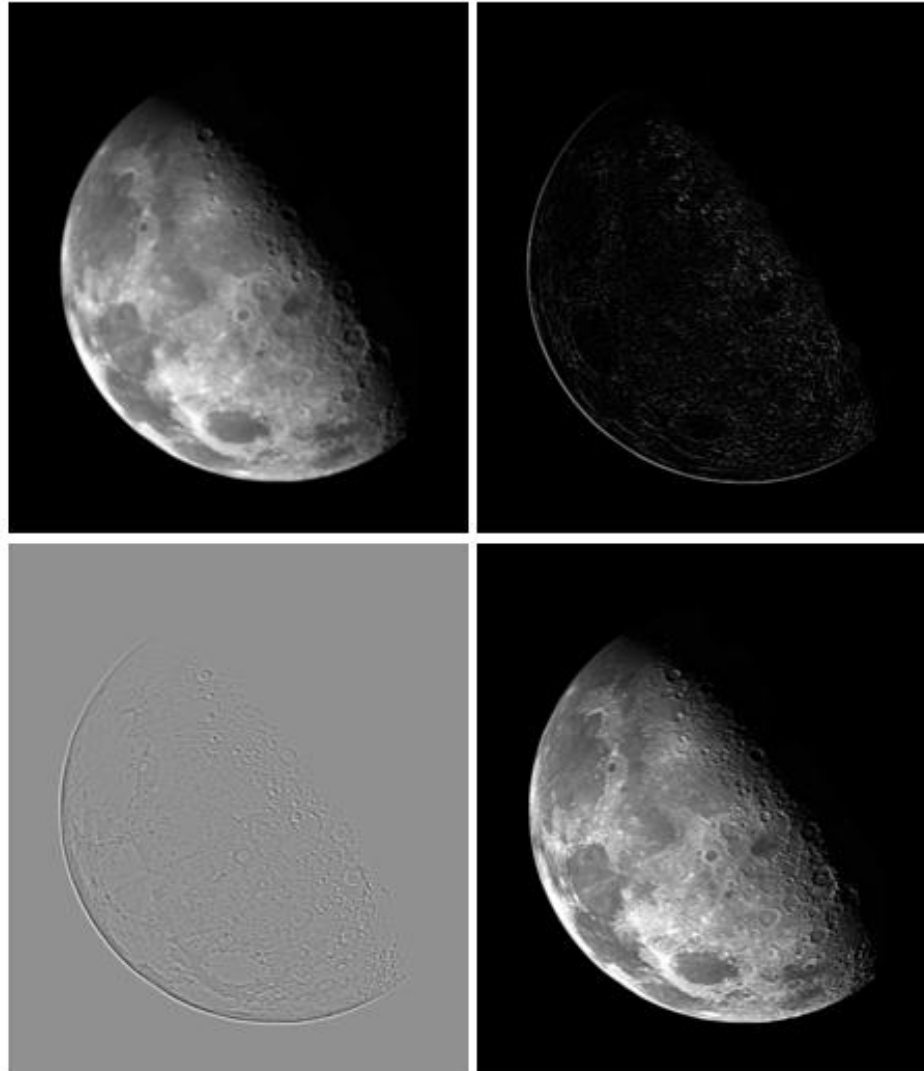
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & +5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad 90^\circ \text{ isotropic} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & +9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad 45^\circ \text{ isotropic}$$

# Cải thiện ảnh với toán tử laplacian

a b  
c d

**FIGURE 3.40**

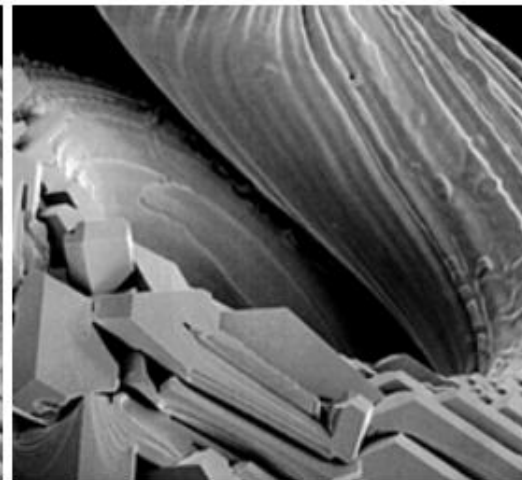
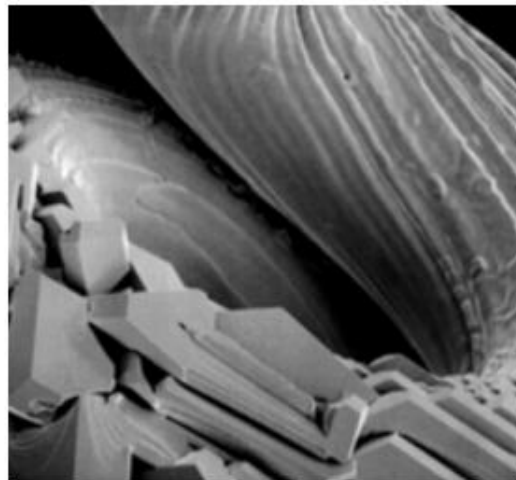
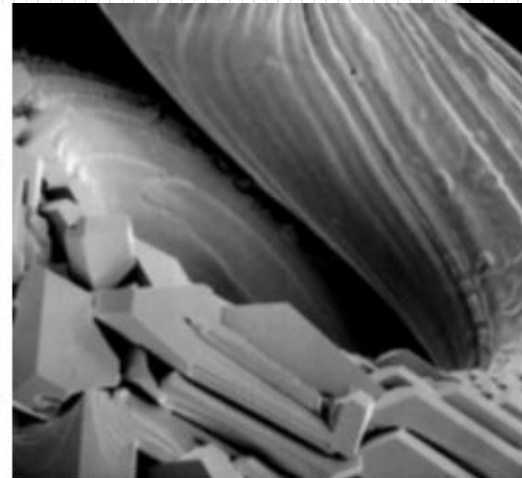
(a) Image of the North Pole of the moon.  
(b) Laplacian-filtered image.  
(c) Laplacian image scaled for display purposes.  
(d) Image enhanced by using Eq. (3.7-5).  
(Original image courtesy of NASA.)



# Cải thiện ảnh với toán tử laplacian

0	-1	0
-1	5	-1
0	-1	0

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



a b c  
d e

**FIGURE 3.41** (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

# CHÚ Ý

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) \\ f(x, y) + \nabla^2 f(x, y) \end{cases}$$

0	-1	0
-1	5	-1
0	-1	0

=

0	0	0
0	1	0
0	0	0

+

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

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

=

0	0	0
0	1	0
0	0	0

+

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

# SỬ DỤNG ĐẠO HÀM BẬC NHẤT TOÁN TỬ GRADIENT

- ❑ Đạo hàm bậc nhất trong xử lý ảnh là được thực hiện bằng cách sử dụng độ lớn của gradient.
- ❑ Với hàm  $f(x, y)$ , gradient của  $f$  tại tọa độ  $(x, y)$  được định nghĩa là một vectơ cột hai chiều:

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



# SỬ DỤNG ĐẠO HÀM BẬC NHẤT TOÁN TỬ GRADIENT

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

Độ lớn của vector được cho bởi:

$$\begin{aligned} \nabla f &= \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} \\ &= \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \end{aligned}$$

Xấp xỉ

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

# SỬ DỤNG ĐẠO HÀM BẬC NHẤT TOÁN TỬ GRADIENT

$$\begin{aligned}\nabla f &\approx |G_x| + |G_y| = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right| \\ &= |f(x+1, y) - f(x, y)| + |f(x, y+1) - f(x, y)|\end{aligned}$$

Như vậy:

$$G_x = f(x+1, y) - f(x, y)$$

$$G_y = f(x, y+1) - f(x, y)$$

# SỬ DỤNG ĐẠO HÀM BẬC NHẤT TOÁN TỬ GRADIENT

- Trong vùng  $3 \times 3$ , sử dụng ký hiệu  $z_5$  của mặt nạ tương ứng với điểm ảnh  $f(x, y)$ , điểm  $z_1$  của mặt nạ tương ứng với điểm ảnh  $f(x-1, y-1)$ , ...
- Xấp xỉ đơn giản nhất đối với đạo hàm bậc nhất thỏa mãn điều kiện:

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

$$G_x = f(x+1, y) - f(x, y)$$

$$G_y = f(x, y+1) - f(x, y)$$


$$G_x = (z_6 - z_5) \text{ và } G_y = (z_8 - z_5).$$

# SỬ DỤNG ĐẠO HÀM BẬC NHẤT TOÁN TỬ GRADIENT

$$G_x = (z_8 - z_5) \text{ và } G_y = (z_6 - z_5)$$

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

□ Khi đó:

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_8 - z_5)^2 + (z_6 - z_5)^2]^{1/2}$$

$$\nabla f \approx |z_8 - z_5| + |z_6 - z_5|$$

# SỬ DỤNG ĐẠO HÀM BẬC NHẤT TOÁN TỬ GRADIENT

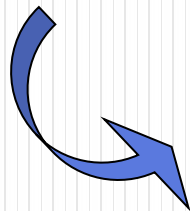
□ Hai toán tử khác do Roberts đề nghị như sau:

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

$$G_x = (z_9 - z_5) \text{ và } G_y = (z_8 - z_6)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$



-1	0	0	-1
0	1	1	0

# SỬ DỤNG ĐẠO HÀM BẬC NHẤT TOÁN TỬ GRADIENT

□ Hai toán tử khác do Sobel đề nghị như sau:

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

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


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

# CHÚ Ý

- Tổng tất cả các hệ số trong mặt nạ bằng 0. Điều này nhằm làm cho đáp ứng tại những vùng cấp xám không thay đổi có giá trị bằng 0.

