

XỬ LÝ ẢNH SỐ

CHƯƠNG 3. NÂNG CAO CHẤT LƯỢNG ẢNH

Trong phần này chúng ta sẽ tìm hiểu về các kỹ thuật tăng cường ảnh trong miền không gian:

- Thế nào là tăng cường ảnh?
- Các kiểu tăng cường ảnh
- Cải thiện ảnh sử dụng toán tử điểm ảnh
- Cải thiện ảnh sử dụng toán tử láng giềng
- Khôi phục ảnh

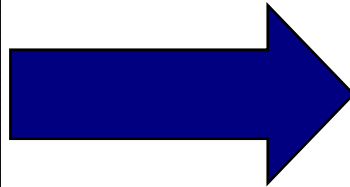
Thế nào là tăng cường ảnh?

- ✓ Tăng cường ảnh là quá trình làm cho bức ảnh trở nên hữu ích hơn.
- ✓ Lý do để thực hiện tăng cường ảnh bao gồm:
 - Làm nổi bật những chi tiết cần quan tâm trong bức ảnh
 - Loại bỏ nhiễu khỏi các bức ảnh
 - Làm cho bức ảnh hấp dẫn hơn về mặt trực quan

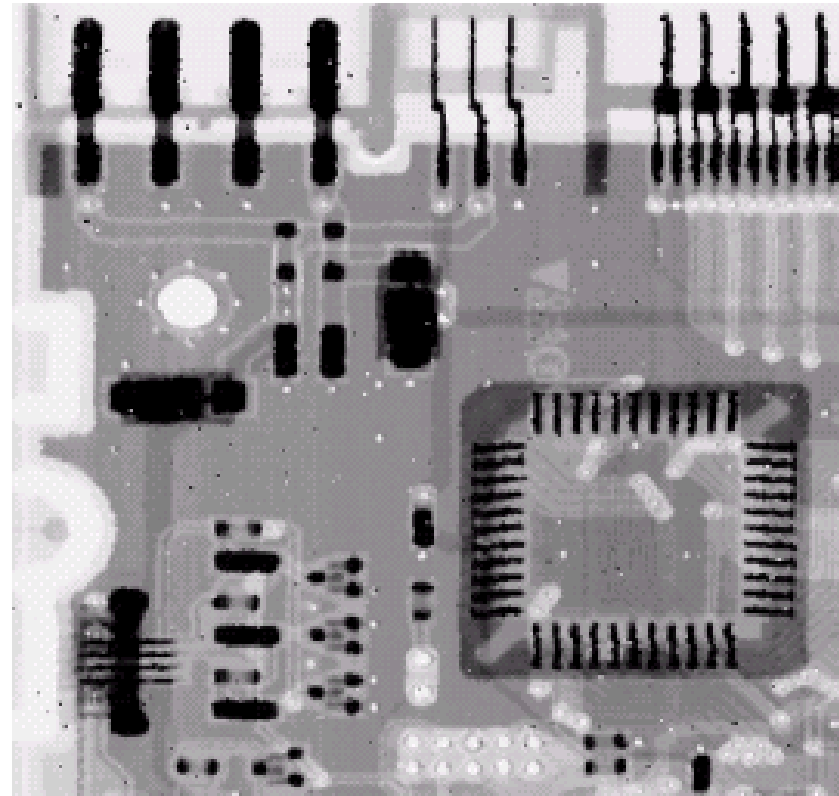
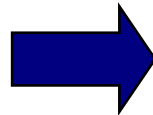
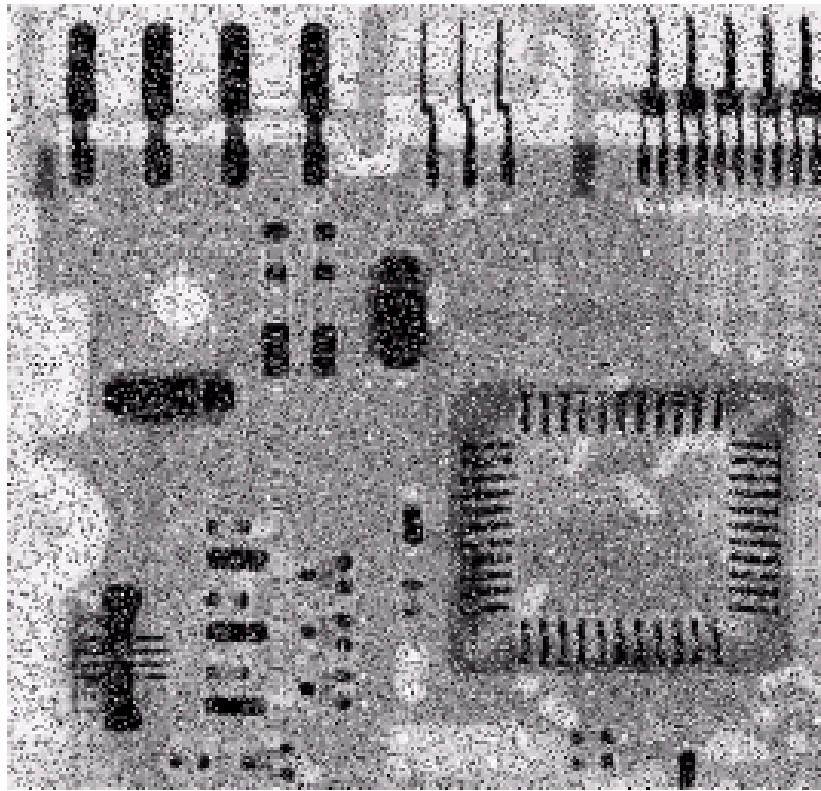
Ví dụ về tăng cường ảnh (1)



Ví dụ về tăng cường ảnh (2)



Ví dụ về tăng cường ảnh (3)

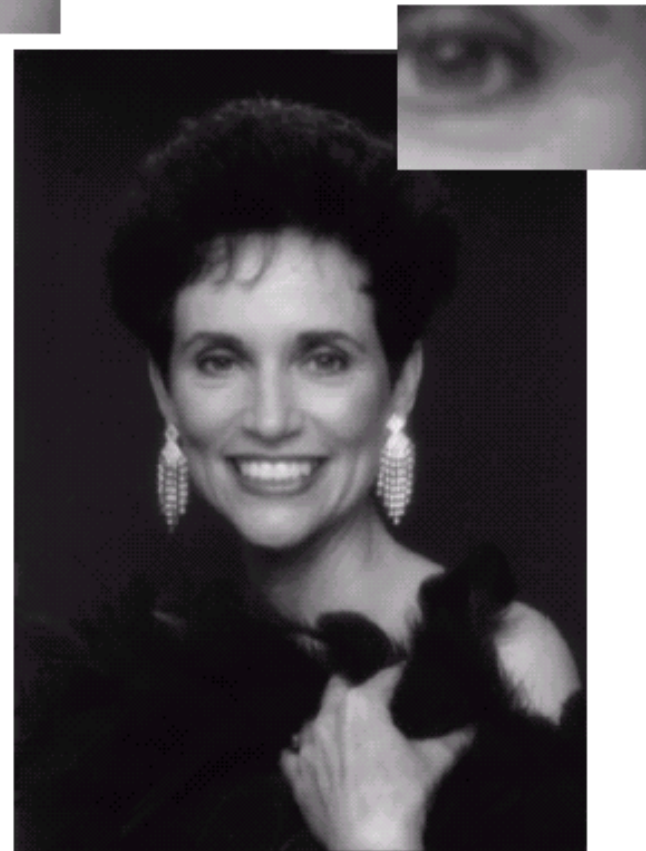


Ví dụ về tăng cường ảnh (4)

Higher attraction



Reduction in fine skin lines in the magnified sections



Miền không gian và miền tần số

Các kỹ thuật tăng cường ảnh có thể chia ra 2 nhánh:

- ❖ Các kỹ thuật tăng cường ảnh trong miền không gian
 - Thao tác trực tiếp với các điểm ảnh
- ❖ Các kỹ thuật tăng cường ảnh trong miền tần số
 - Đưa ảnh về miền tần số bằng phép biến đổi fourier hoặc wavelet sau đó xử lý trên miền tần số

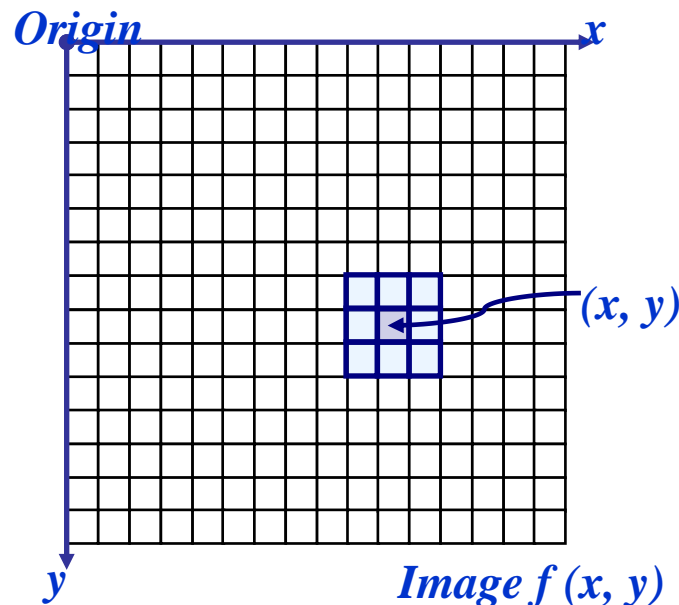
Chương này sẽ tập trung vào các kỹ thuật xử lý ảnh trong miền **không gian**

CẢI THIỆN ẢNH SỬ DỤNG TOÁN TỬ ĐIỂM

Toán tử điểm

Xử lý ảnh trên miền không gian

- ❖ Phép toán xử lý ảnh có dạng: $g(x,y) = T[f(x,y)]$
 - ✓ $f(x,y)$ là ảnh đầu vào
 - ✓ $g(x,y)$ là ảnh đầu ra
 - ✓ T là một số toán tử tác động lên các điểm ảnh xung quanh (x,y)



Xử lý điểm ảnh

Xử lý ảnh trên miền không gian

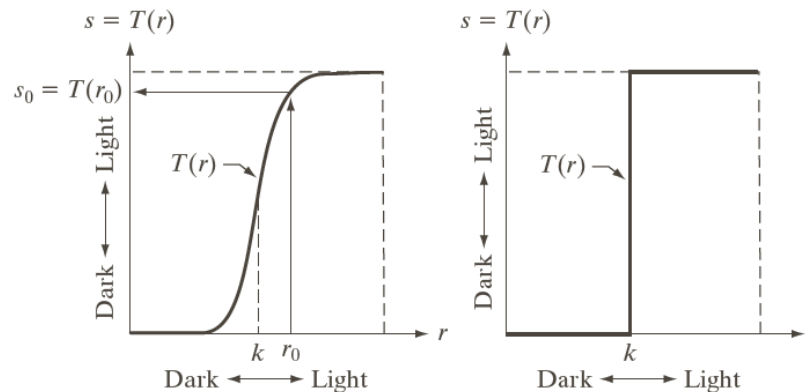
$$g(x,y) = T[f(x,y)]$$

- ✓ Nếu T chỉ tác động lên 1 điểm ảnh tại (x,y) thì T là *hàm biến đổi mức xám hay toán tử xử lý điểm ảnh*
- ✓ Các toán tử xử lý điểm ảnh có dạng:

$$s = T(r)$$

r : mức xám của ảnh đầu vào tại (x,y)

s : mức xám của ảnh đầu ra tại (x,y)



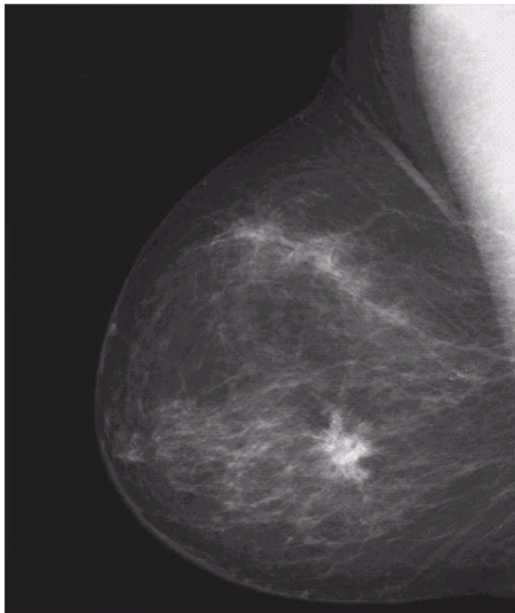
MỘT SỐ TOÁN TỬ ĐIỂM ẢNH

- Biến đổi âm bản
- Phân ngưỡng
- Một số hàm biến đổi mức xám cơ bản: tuyến tính, logarithm, hàm mũ.

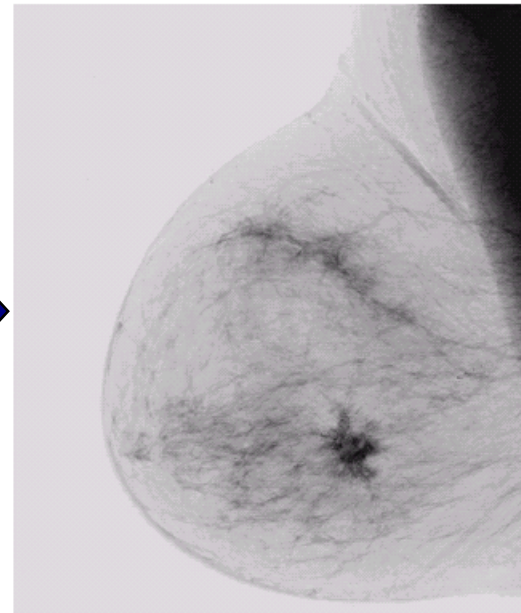
Phép biến đổi âm bản

- ❖ Phép biến đổi âm bản (negative image) sử dụng để tăng cường các chi tiết trắng hoặc xám trong nền tối
- ❖ Phép biến đổi: $s = T(r) = Intensive_{max} - r$

Ảnh gốc



$$s = 1.0 - r$$

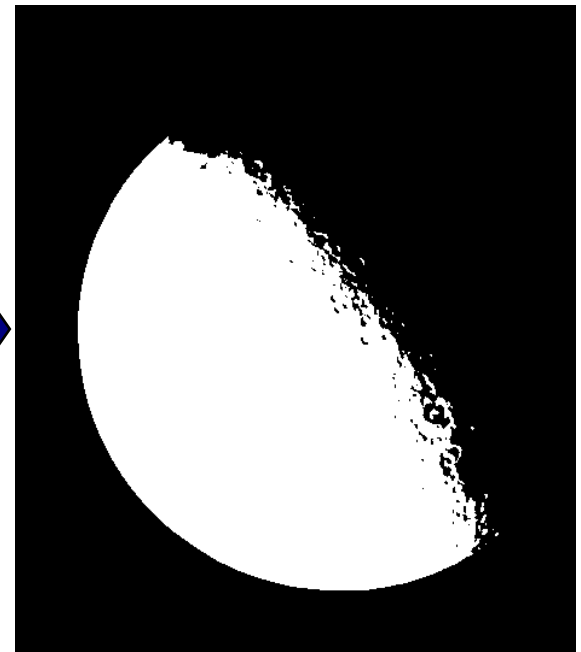
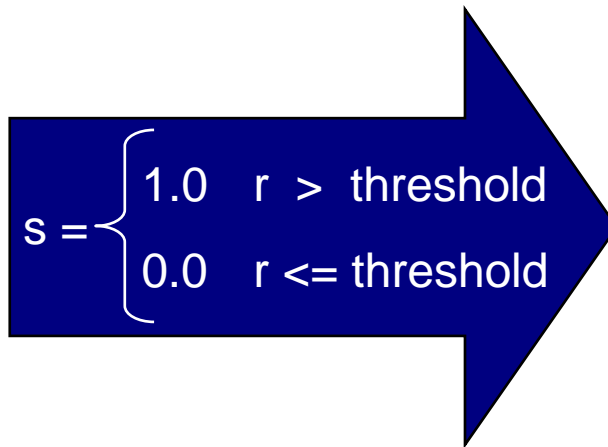


Ảnh âm bản

Phân ngưỡng (Thresholding)

❖ Biến đổi phân ngưỡng được sử dụng cho việc làm nổi bật các đối tượng cần quan tâm trên nền của ảnh

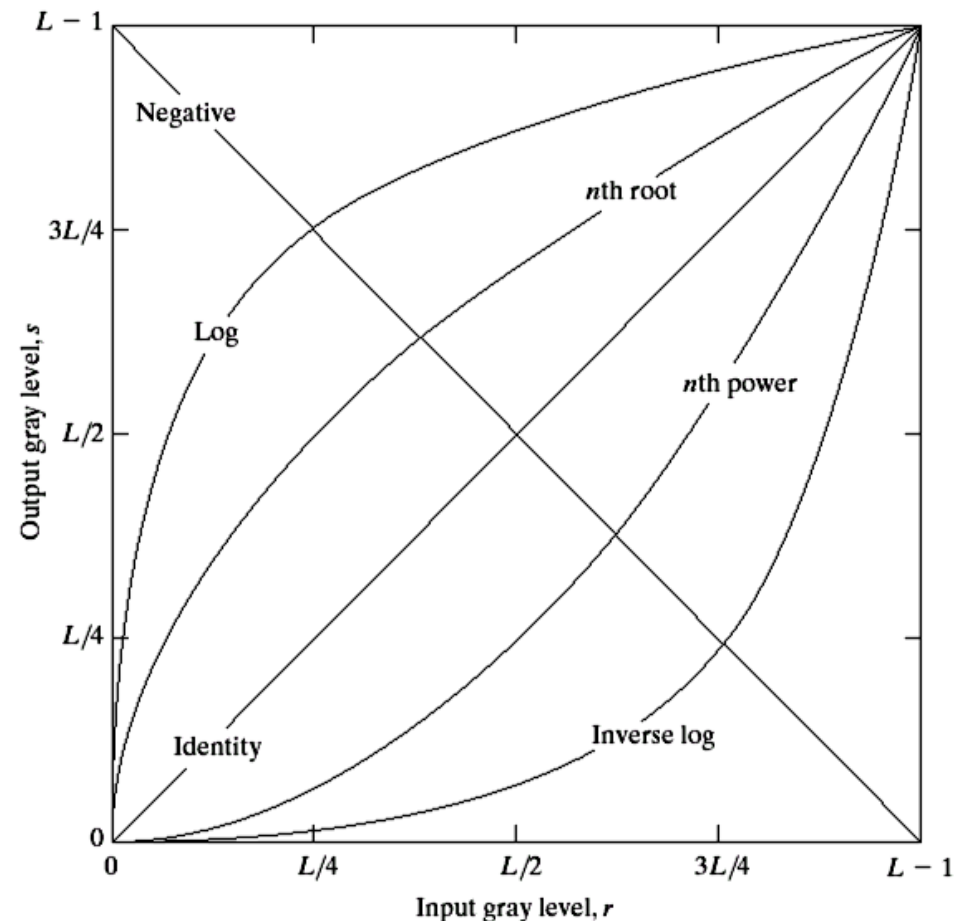
❖ Phép biến đổi: $s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r \leq threshold \end{cases}$



Các hàm cơ bản biến đổi mức xám

Có 3 hàm biến đổi mức xám thông dụng nhất:

- Biến đổi tuyến tính
 - Negative/Identity
- Biến đổi logarith
 - Log/Inverse log
- Biến đổi hàm mũ
 - n^{th} power/ n^{th} root



Biến đổi logarith

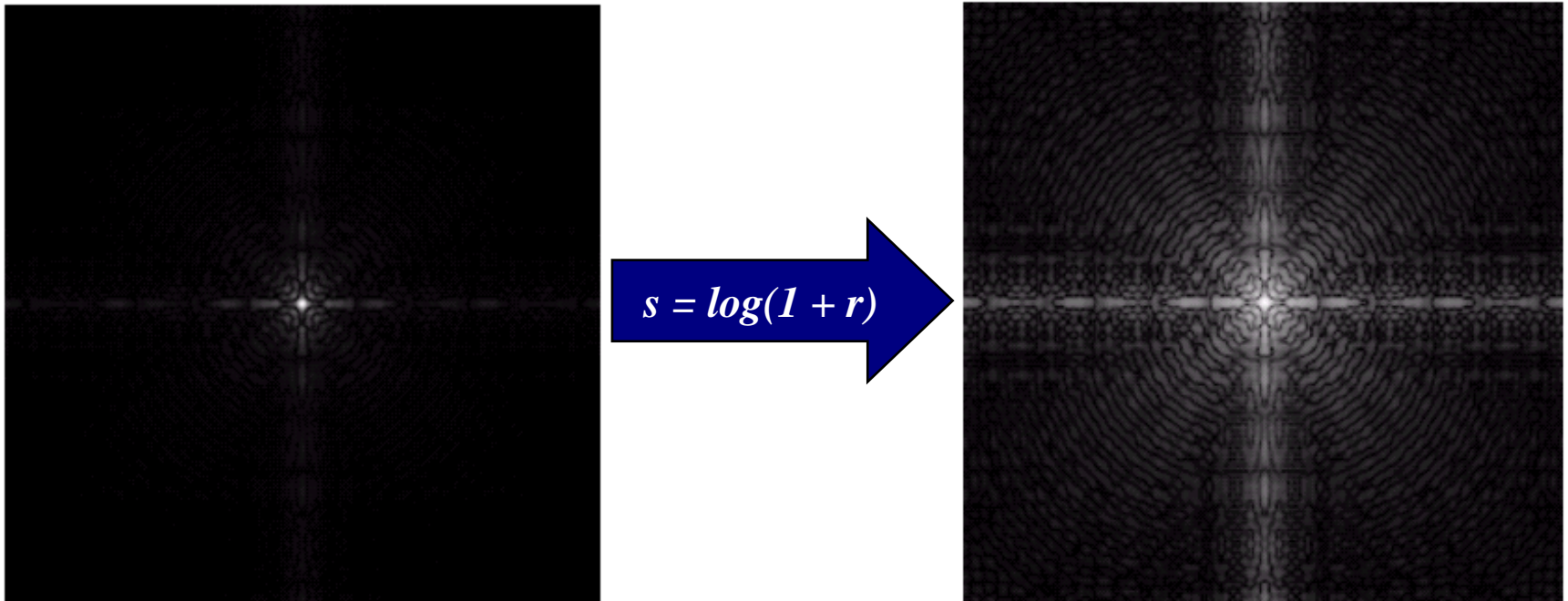
- ❖ Dạng tổng quát của biến đổi log:

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

- ❖ Biến đổi log ánh xạ một dải hẹp mức xám thấp của ảnh đầu vào đầu vào sang một dải rộng hơn cho ảnh đầu ra
- ❖ Biến đổi ngược của hàm log thực hiện điều ngược lại

Biến đổi logarith (tiếp...)

- ❖ Các hàm log Log được sử dụng khi giá trị mức xám của ảnh đầu vào nằm trong 1 dải rất rộng.
- ❖ Biến đổi Fourier của ảnh được đưa qua biến đổi log để hiển thị các chi tiết rõ ràng hơn

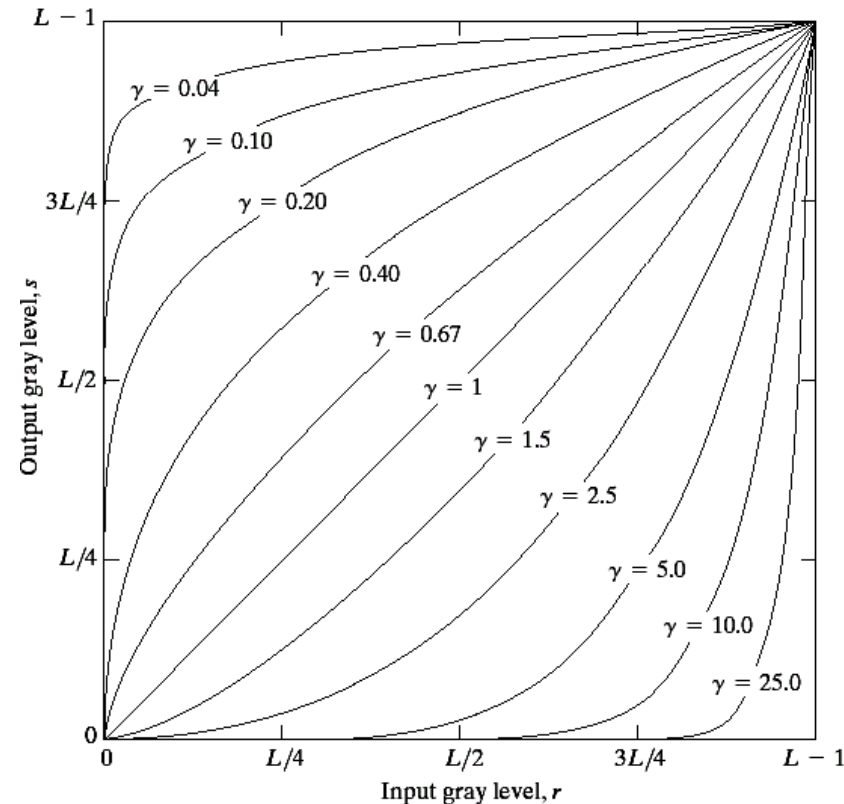


Biến đổi hàm mũ (1)

❖ Biến đổi hàm mũ có dạng:

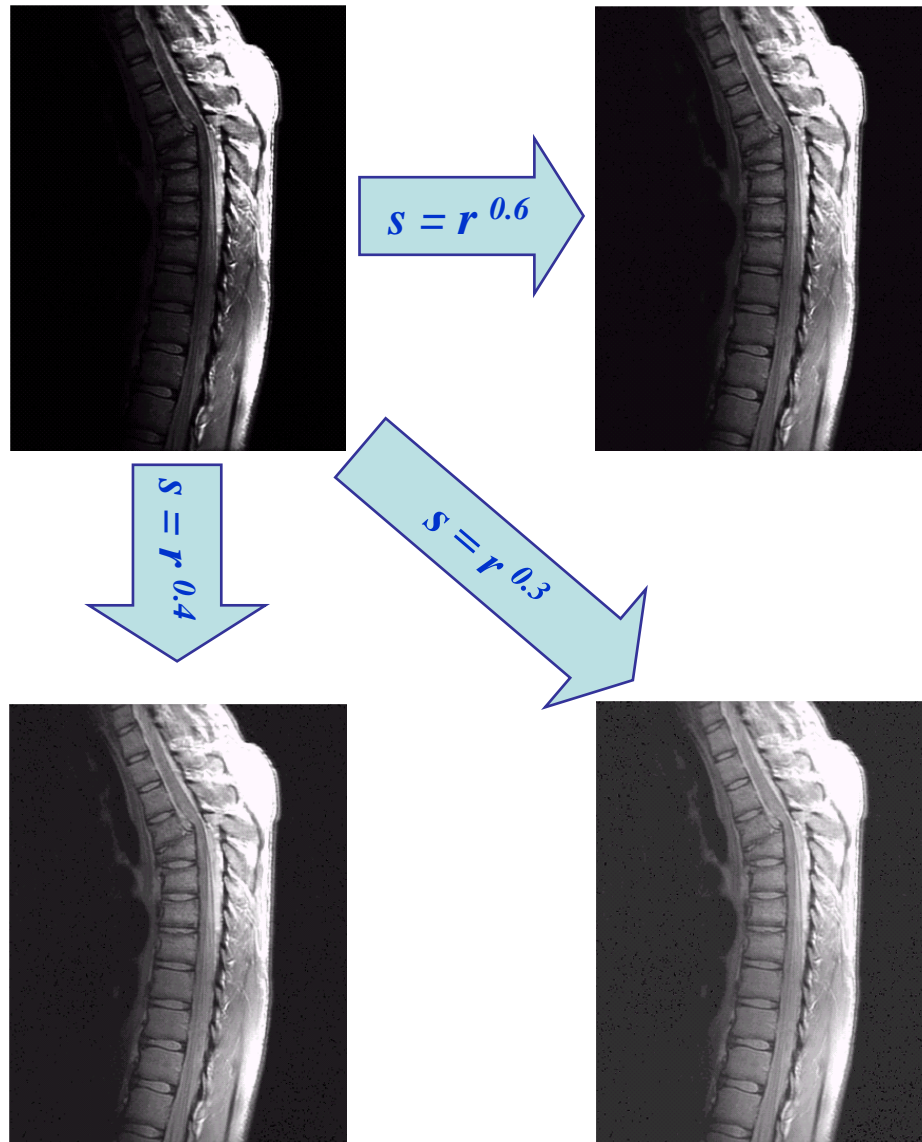
$$s = c * r^\gamma$$

❖ Biến đổi hàm mũ ánh xạ một dải hẹp mức xám thấp của ảnh đầu vào đầu vào sang một dải rộng hơn cho ảnh đầu ra và ngược lại

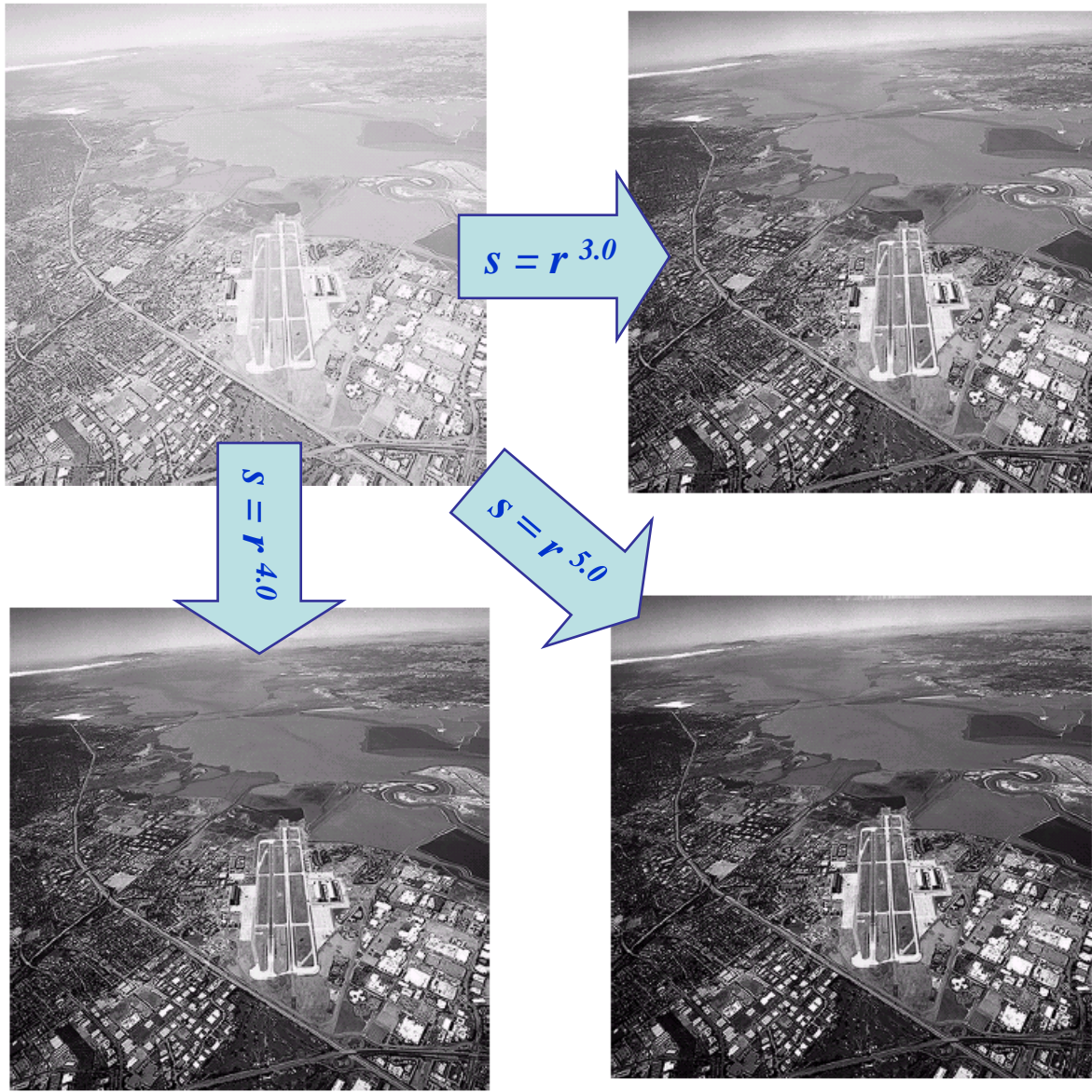


Khác với biến đổi log, có thể thay đổi giá trị γ của biến đổi hàm mũ để tạo ra các đồ thị biến đổi khác nhau.

Biến đổi hàm mũ (2)

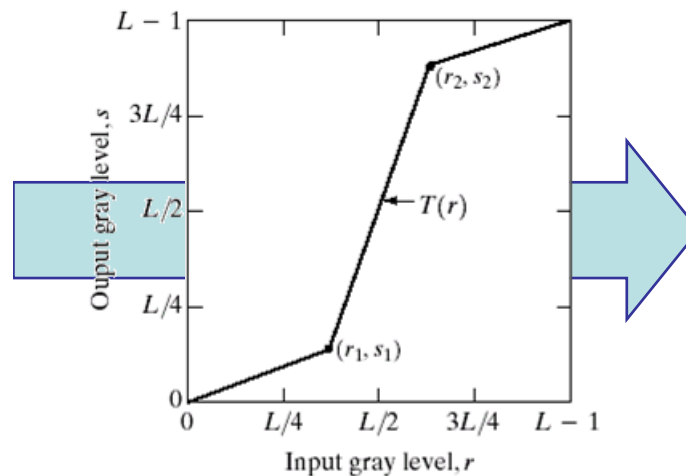
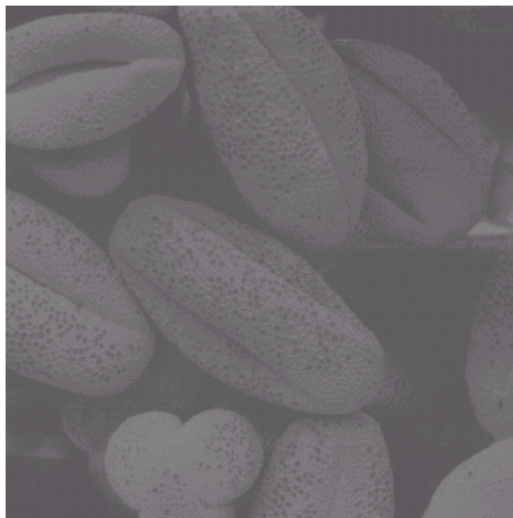


Biến đổi hàm mũ (3)



Tăng độ tương phản

- ❖ Thay vì sử dụng 1 hàm toán học, ta có thể sử dụng nhiều hàm toán học tại các phân đoạn khác nhau để biến đổi mức xám



LỰC ĐỒ XÁM VÀ CÂN BẰNG LỰC ĐỒ XÁM

Histogram (Lược đồ mức xám)

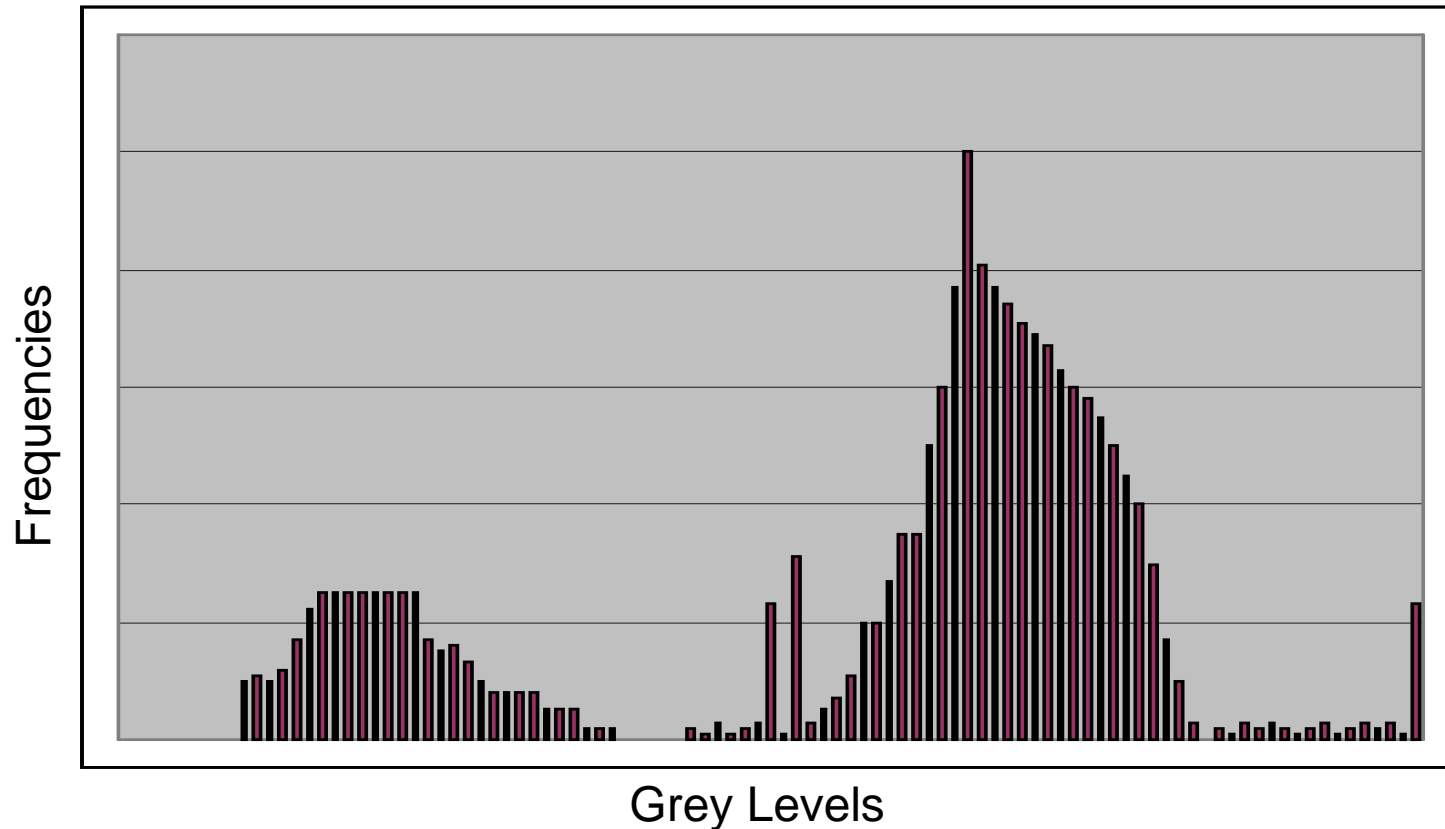
- Lược đồ xám của ảnh có mức xám trong khoảng $[0, L - 1]$ là hàm rời rạc có dạng như sau:

$$p(r_k) = \frac{n_k}{n}, \text{ với } \begin{cases} n_k \text{ là số lượng pixel có giá trị xám } r_k \\ n \text{ là tổng số pixel của ảnh} \end{cases}$$

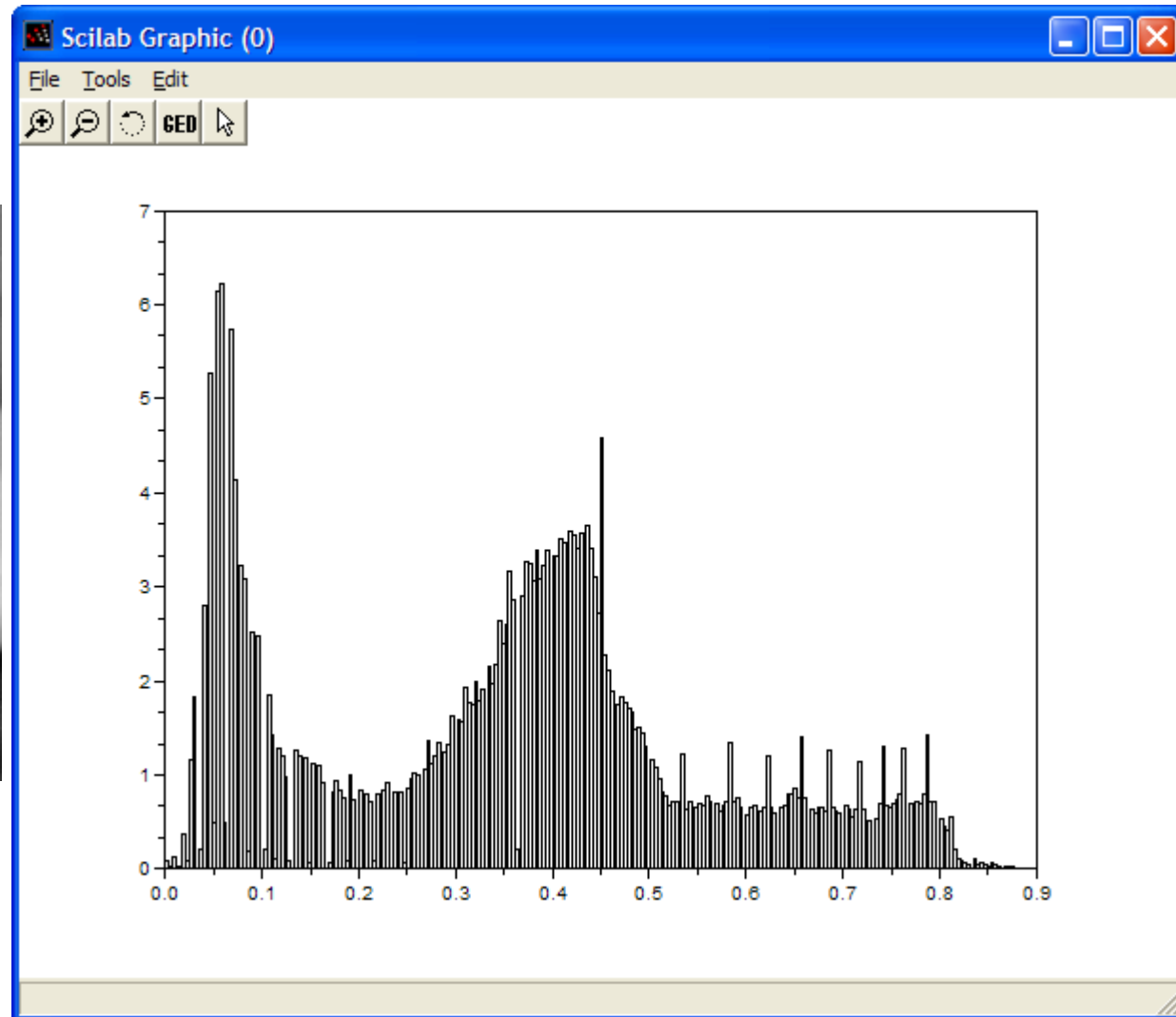
- Hàm $p(r_k)$ mô tả tỷ lệ giữa tổng số pixel có mức xám r_k và tổng số pixel của ảnh
- Histogram mô tả tổng thể phân bố mức xám của ảnh

Lược đồ xám của ảnh

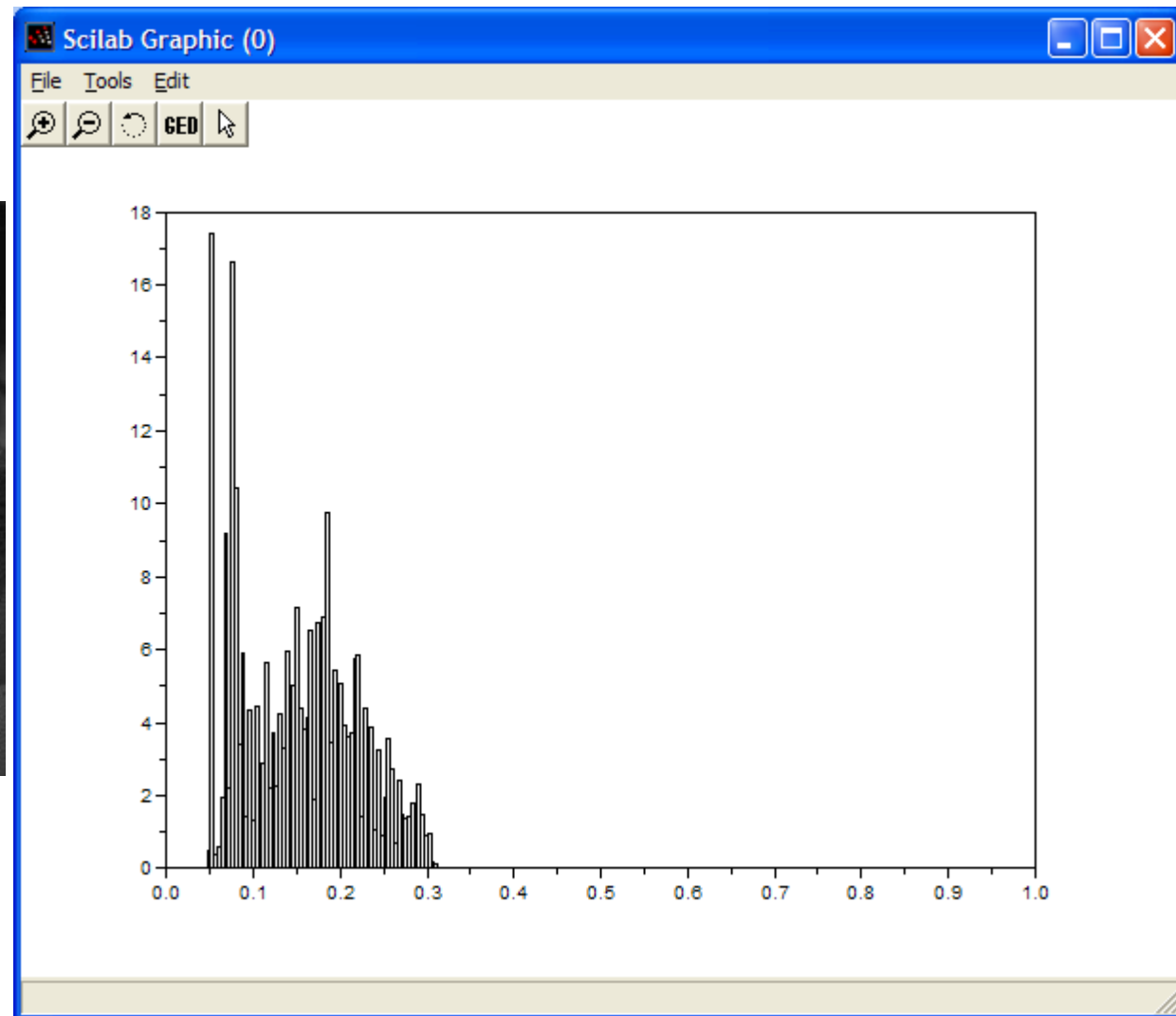
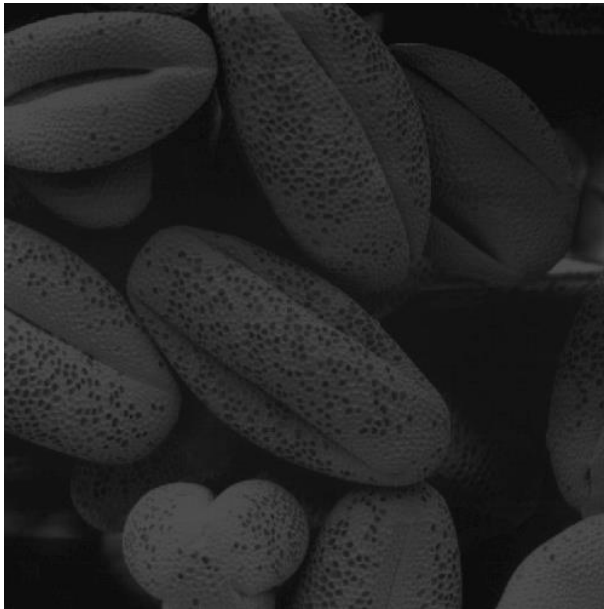
Ví dụ về biểu diễn histogram:



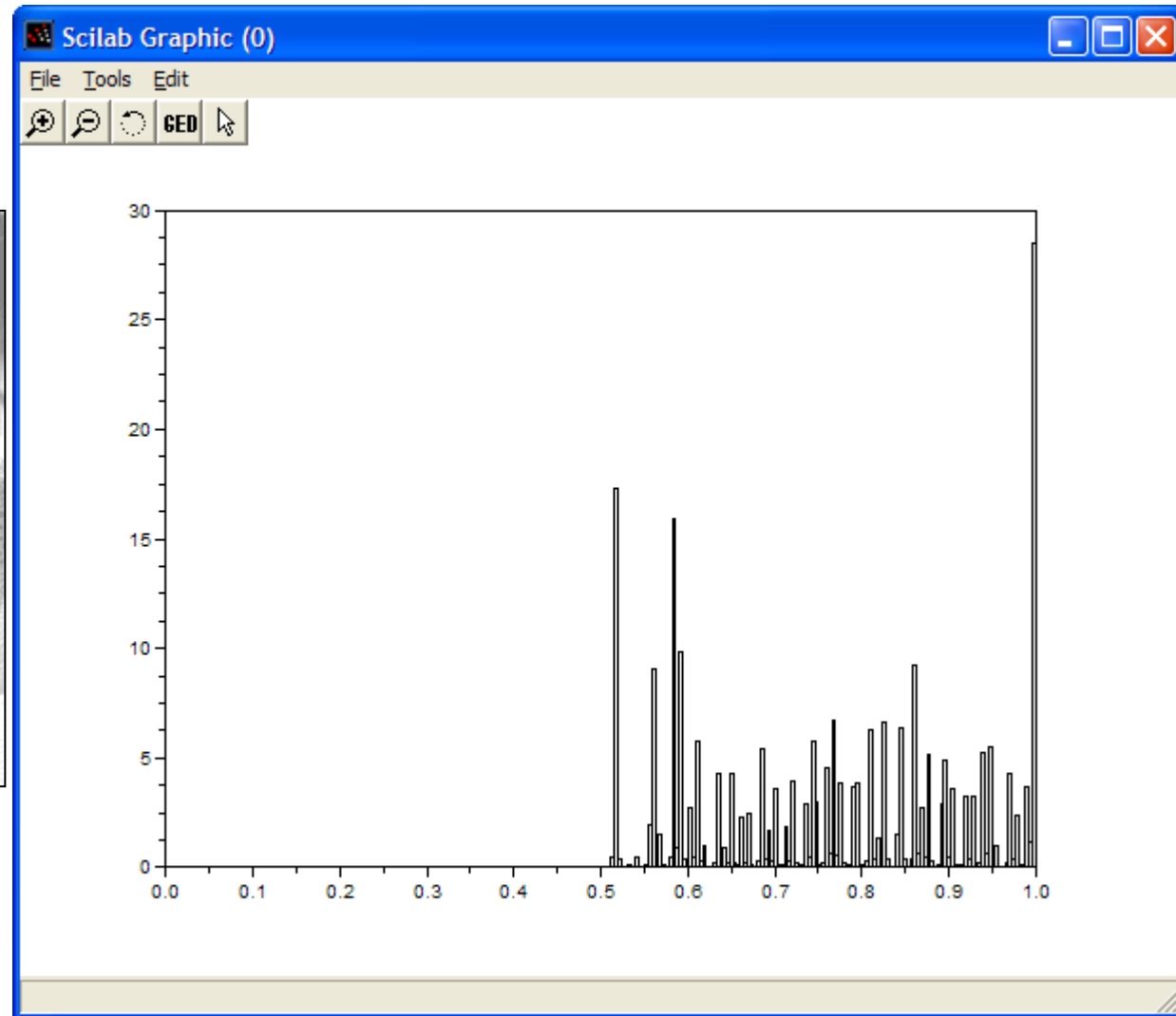
Ví dụ lược đồ xám (cont...)



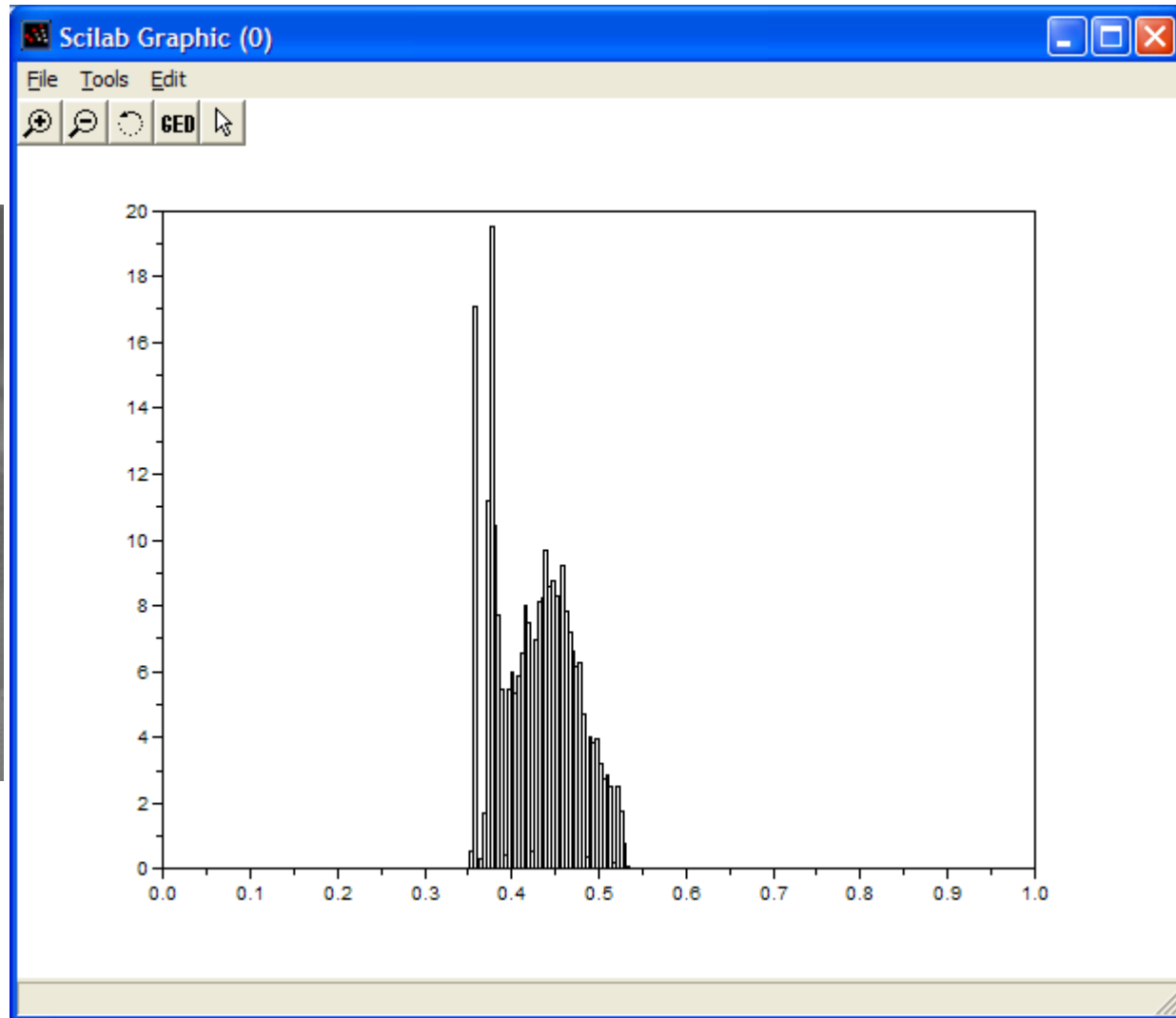
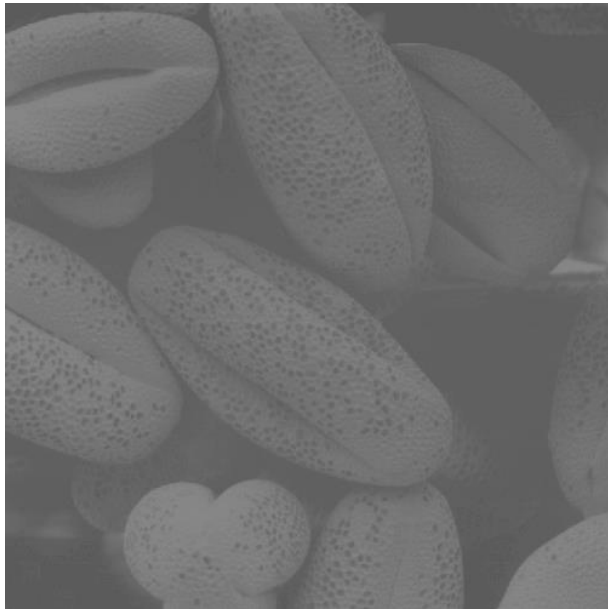
Ví dụ lược đồ xám (cont...)



Ví dụ lược đồ xám (cont...)



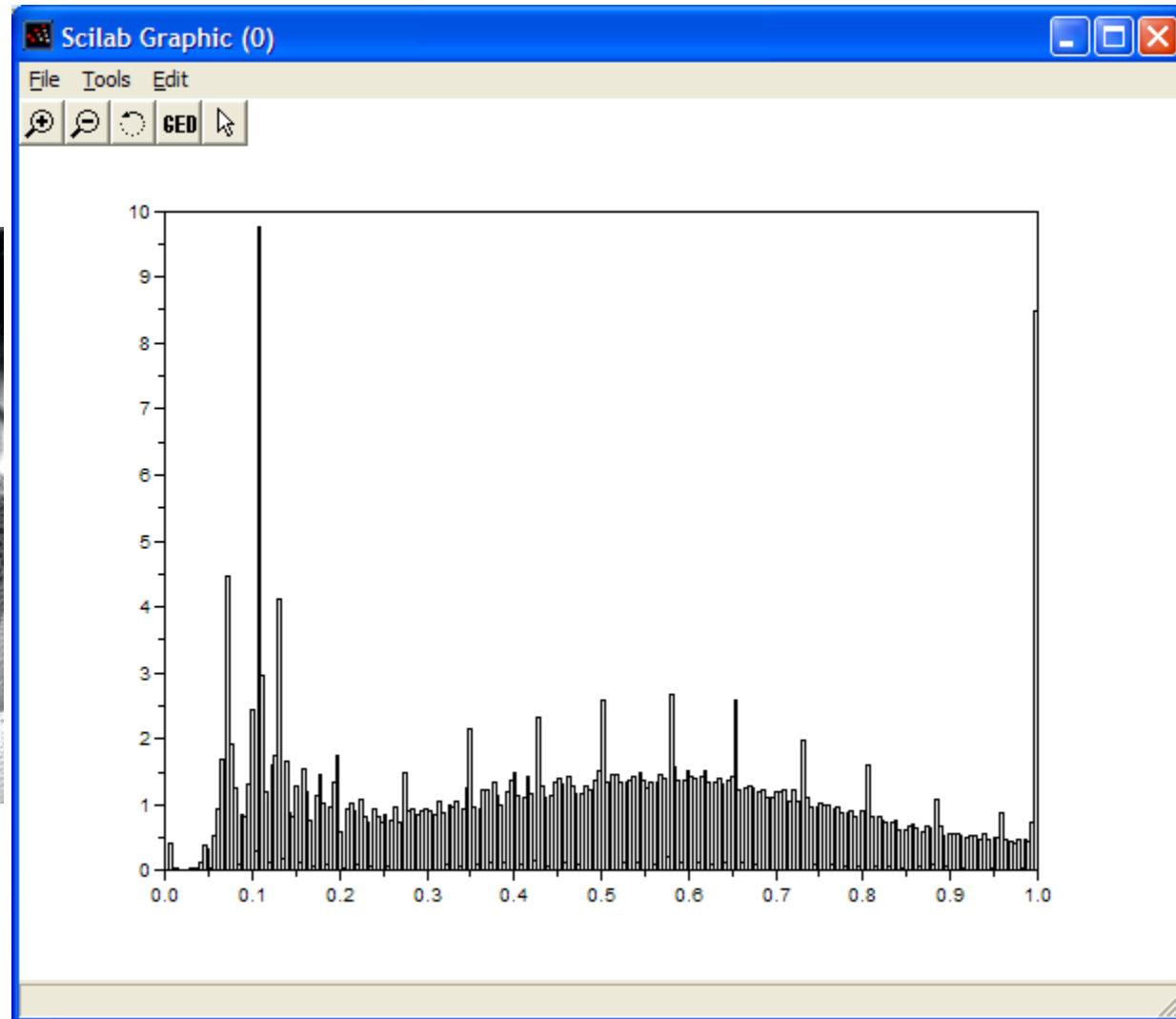
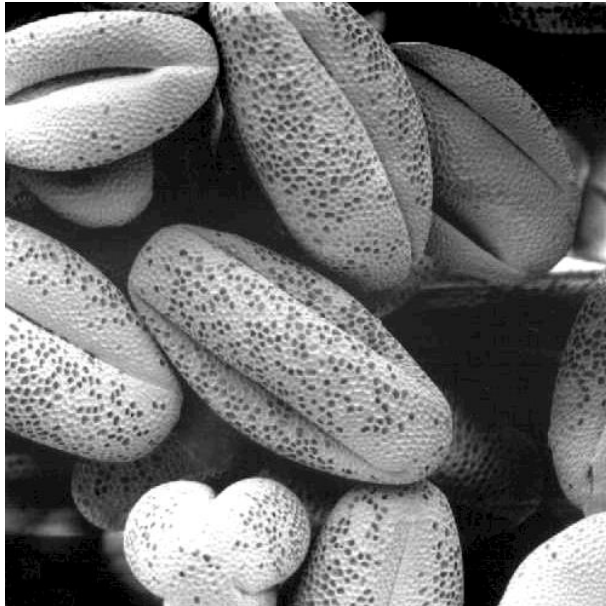
Ví dụ lược đồ xám (cont...)



Ví dụ lược đồ xám (cont...)

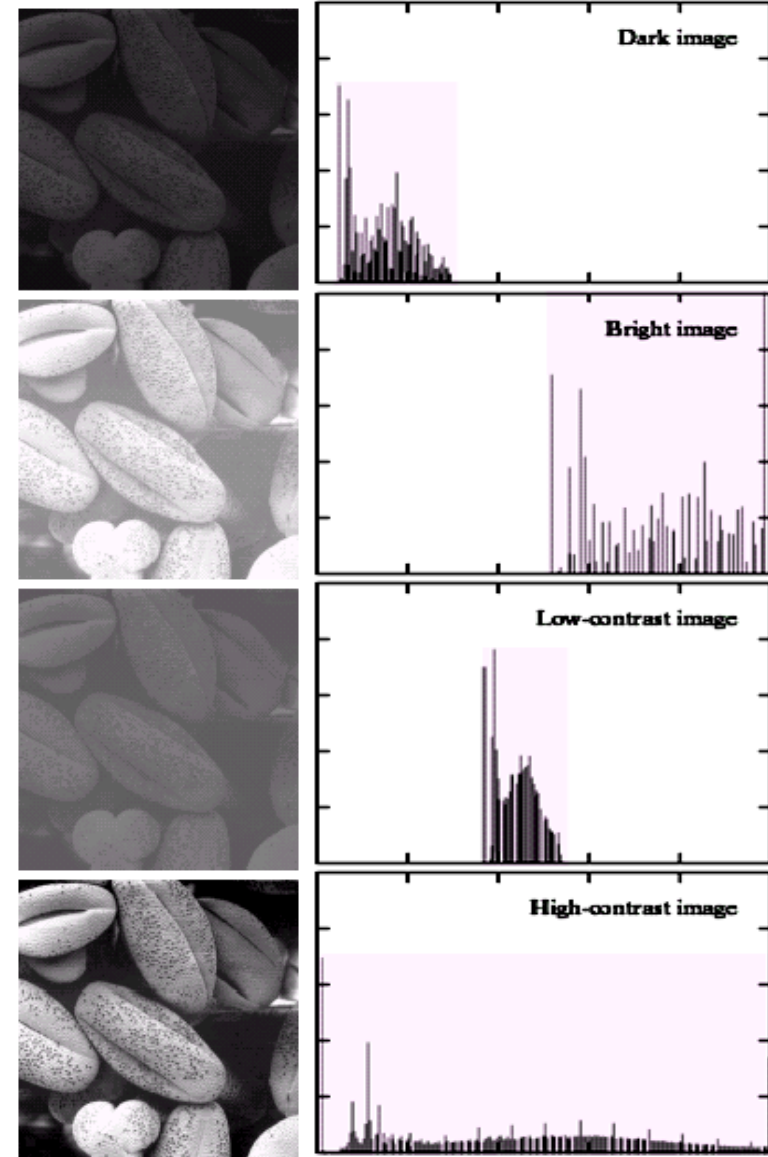


Ví dụ lược đồ xám (cont...)



Ví dụ lược đồ xám (cont...)

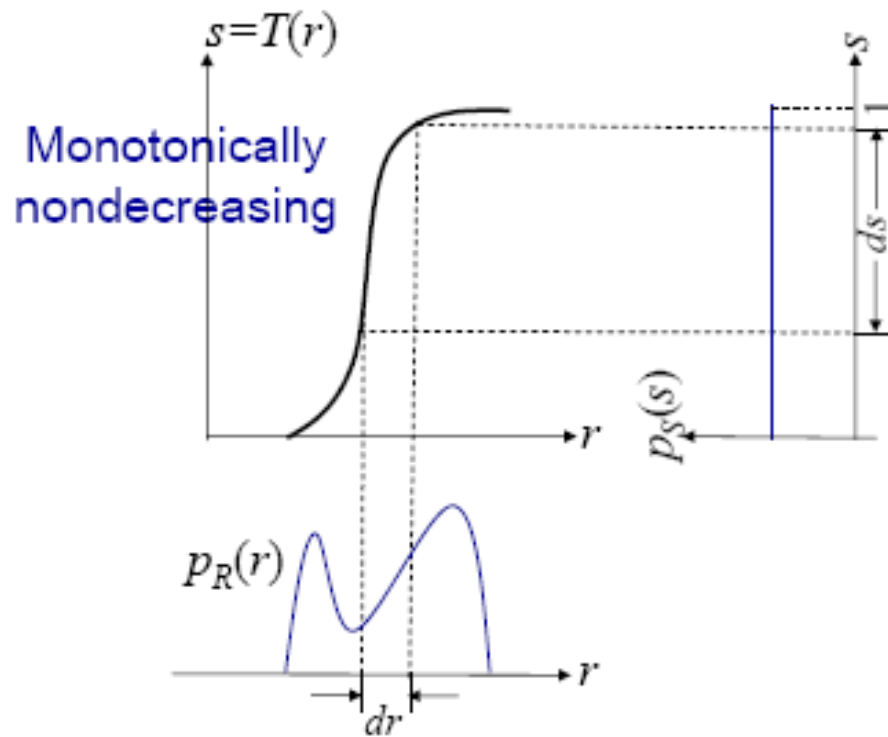
Bức ảnh có độ tương phản cao thì phân bố xác suất mức xám đều và đồng nhất



Cân bằng lược đồ xám

❖ Việc phải làm là tìm hàm biến đổi $s = T(r)$

$$p_R(r) \Rightarrow \boxed{T} \Rightarrow \text{uniform } p_S(s)$$



Cân bằng lược đồ xám

- r_k : mức xám đầu vào
- s_k : mức xám đầu ra
- k : dải cường độ sáng
- n_j : tần suất của cường độ j
- n : tổng số điểm ảnh
- L' : số mức xám ở ảnh đầu ra
(thông thường bằng số mức xám đầu vào)

$$\begin{aligned} s_k &= T(r_k) \\ &= (L' - 1) \sum_{j=0}^k p_r(r_j) \\ &= (L' - 1) \sum_{j=0}^k \frac{n_j}{n} \end{aligned}$$

Ví dụ 1

- Xét bức ảnh có kích thước 64×64 ở 8 mức xám (0, 1, ..., 7). Chuẩn hóa giá trị mức xám như trong bảng sau. Hãy thực hiện quá trình cân bằng mức xám cho ảnh.

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

Ví dụ

$$s_0 = T(r_0) = \sum_{i=0}^0 p_{\text{in}}(r_i) = p_{\text{in}}(r_0) = 0.19 \rightarrow 1/7$$

$$s_1 = T(r_1) = \sum_{i=0}^1 p_{\text{in}}(r_i) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) = 0.44 \rightarrow 3/7$$

$$s_2 = T(r_2) = \sum_{i=0}^2 p_{\text{in}}(r_i) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + p_{\text{in}}(r_2) = 0.65 \rightarrow 5/7$$

$$s_3 = T(r_3) = \sum_{i=0}^3 p_{\text{in}}(r_i) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_3) = 0.81 \rightarrow 6/7$$

$$s_4 = T(r_4) = \sum_{i=0}^4 p_{\text{in}}(r_i) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_4) = 0.89 \rightarrow 6/7$$

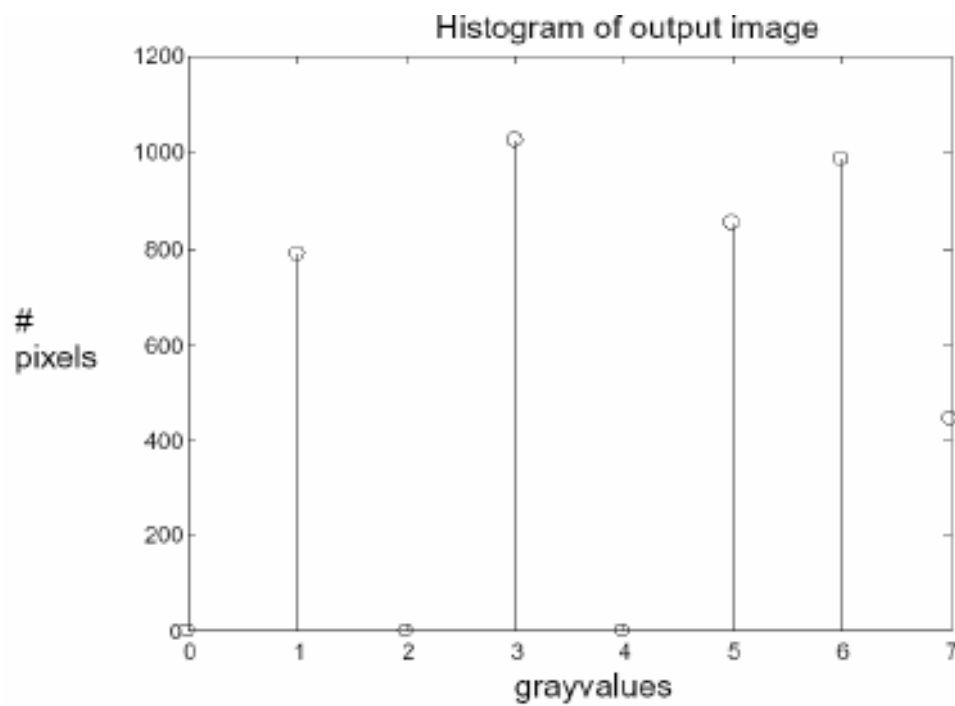
$$s_5 = T(r_5) = \sum_{i=0}^5 p_{\text{in}}(r_i) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_5) = 0.95 \rightarrow 1$$

$$s_6 = T(r_6) = \sum_{i=0}^6 p_{\text{in}}(r_i) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_6) = 0.98 \rightarrow 1$$

$$s_7 = T(r_7) = \sum_{i=0}^7 p_{\text{in}}(r_i) = p_{\text{in}}(r_0) + p_{\text{in}}(r_1) + \cdots + p_{\text{in}}(r_7) = 1.00 \rightarrow 1$$

Ví dụ

k	s_k	n_k	$p(s_k)=n_k/n$
0	1/7	790	0.19
1	3/7	1023	0.25
2	5/7	850	0.21
3	6/7	985	0.24
4	1	448	0.11



Ví dụ 2

- Cho bức ảnh I. Hãy vẽ lược đồ xám và thực hiện cân bằng lược đồ xám. Tìm ảnh I' sau khi đã cân bằng lược đồ xám.

$$I = \begin{pmatrix} 1 & 2 & 0 & 4 \\ 1 & 0 & 0 & 7 \\ 2 & 2 & 1 & 0 \\ 4 & 1 & 2 & 1 \\ 2 & 0 & 1 & 1 \end{pmatrix}$$

- Cho bức ảnh I . Hãy vẽ lược đồ xám và thực hiện cân bằng lược đồ xám. Tìm ảnh I' sau khi đã cân bằng lược đồ xám.

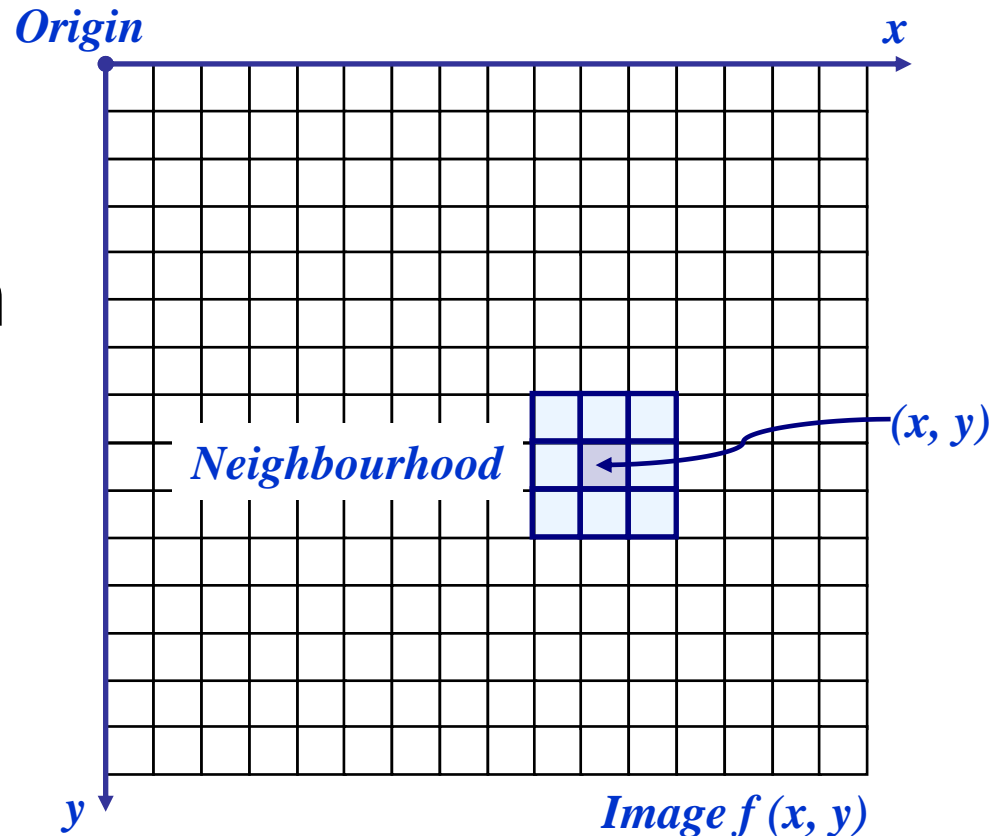
$$I = \begin{bmatrix} 0 & 5 & 0 & 5 & 5 \\ 1 & 7 & 6 & 6 & 7 \\ 5 & 5 & 6 & 2 & 7 \\ 6 & 6 & 7 & 5 & 6 \\ 0 & 1 & 5 & 0 & 2 \end{bmatrix}$$

CẢI THIỆN ẢNH SỬ DỤNG CỬA SỔ

Toán tử láng giềng (neighbourhood)

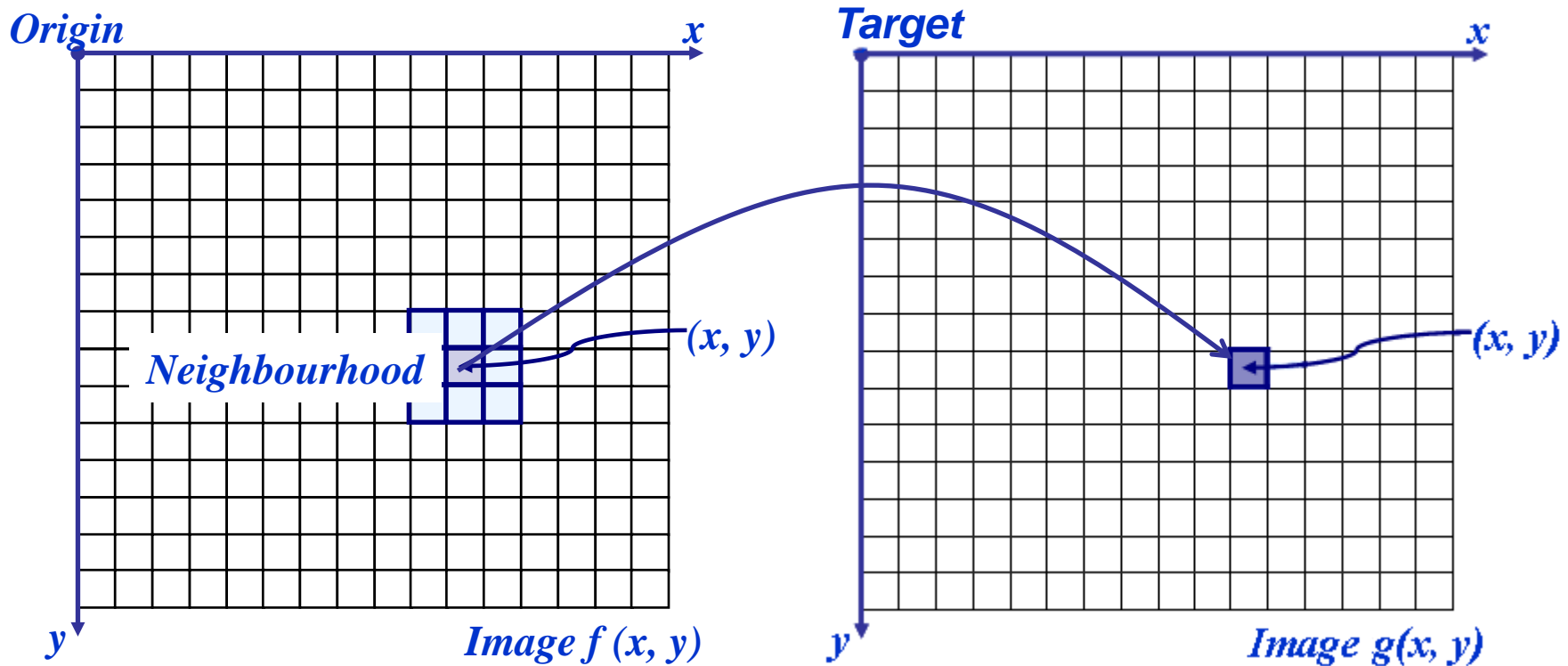
- Toán tử láng giềng hoạt động trên một tập các pixel xung quanh pixel tâm.

- Neighbourhoods (láng giềng, lân cận) hầu hết có dạng hình chữ nhật quanh pixel tâm.



Toán tử láng giềng

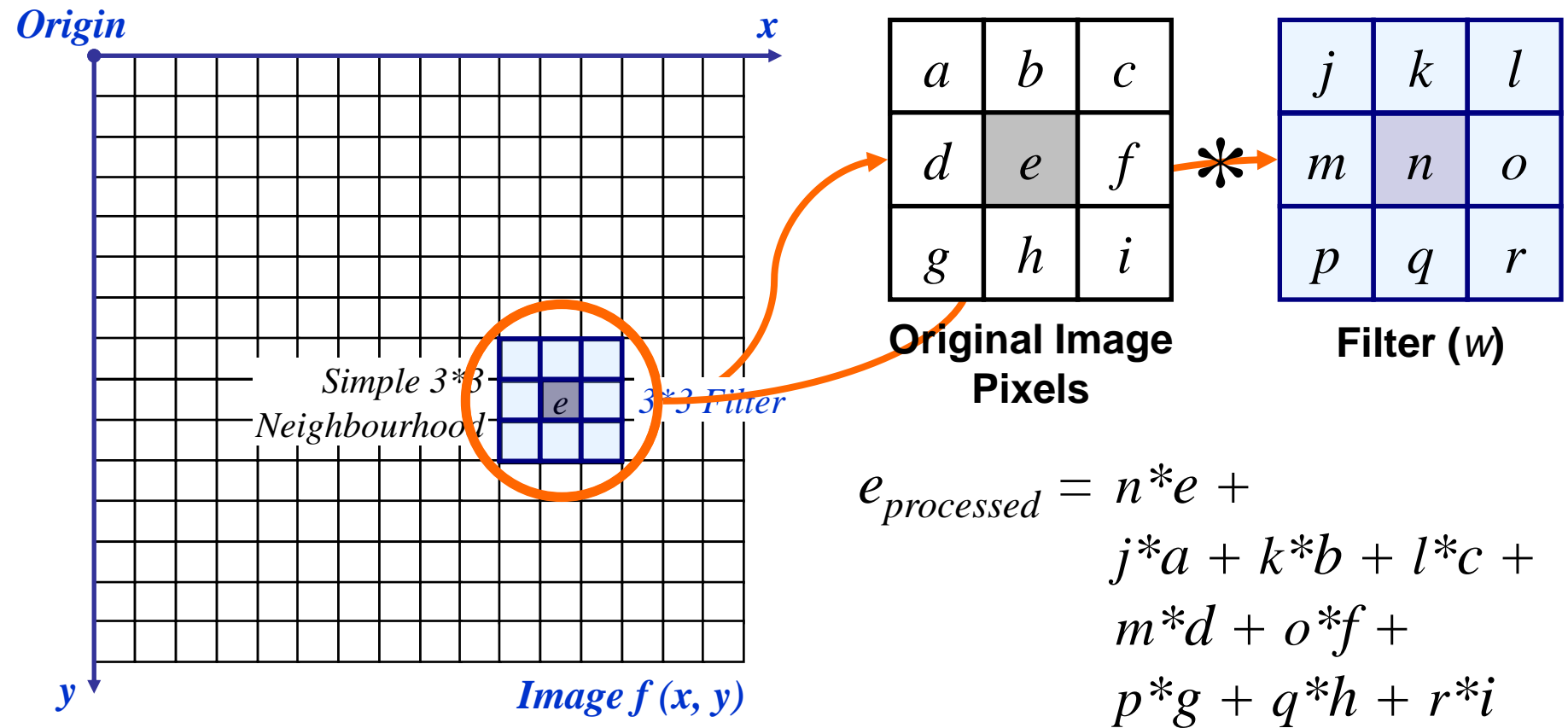
- Đối với mỗi pixel trong ảnh gốc, sau xử lý đầu ra là pixel ở cùng vị trí đó trong ảnh đích.



Một số toán tử láng giềng đơn giản

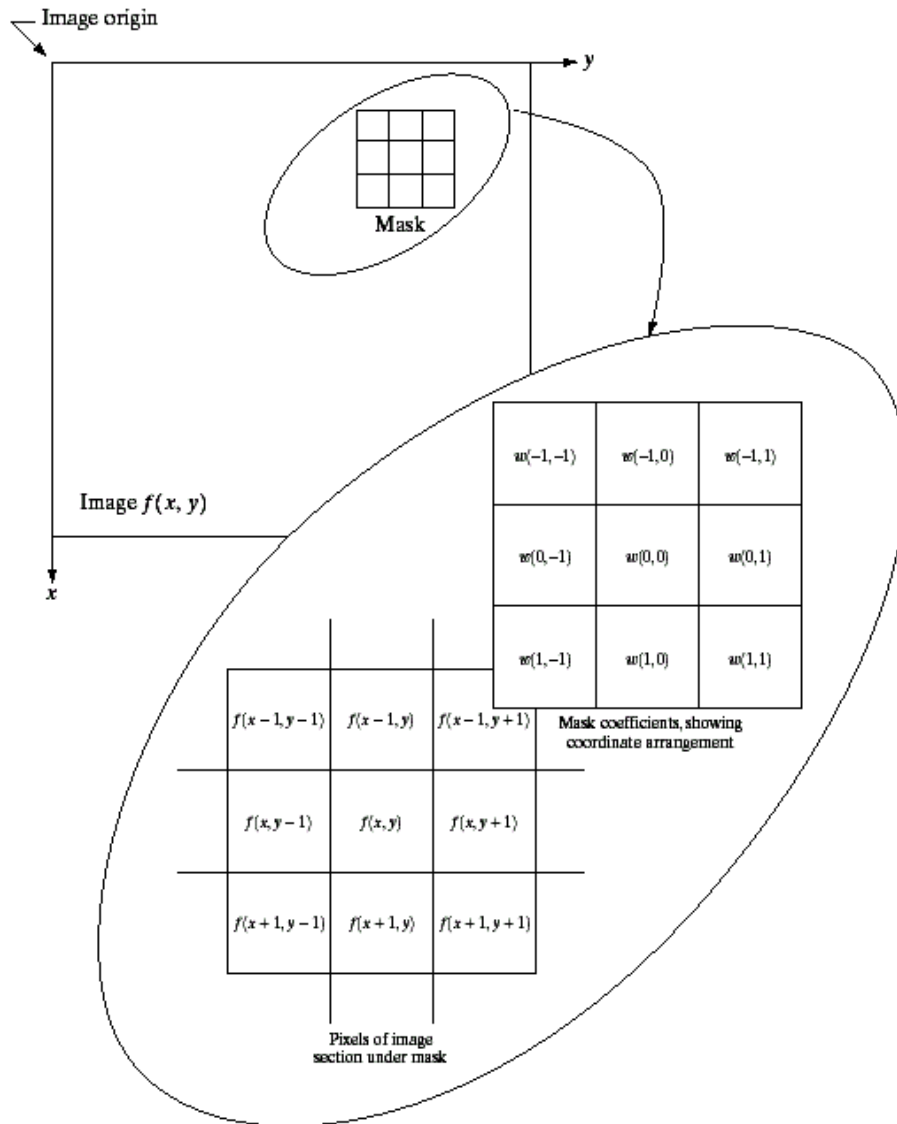
- **Min:** Thiết lập giá trị pixel trong ảnh mới là giá trị min của neighbourhood.
- **Max:** Thiết lập giá trị pixel trong ảnh mới là max của neighbourhood.

Quá trình lọc không gian



Quá trình trên được lặp lại đối với mỗi pixel trong ảnh gốc để tạo ra ảnh mới.

Lọc không gian: Công thức



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

Ví dụ



Original image

Image with color values placed over it

Image with 3x3 kernel placed over it

Output image

164	188	164
178	201	197
174	168	181

Color values



0	1	0
1	1	1
0	1	0

Kernel

Divided by the sum of the kernel

$932 \div 5 = \text{new pixel color}$

Các bộ lọc không gian làm mịn ảnh

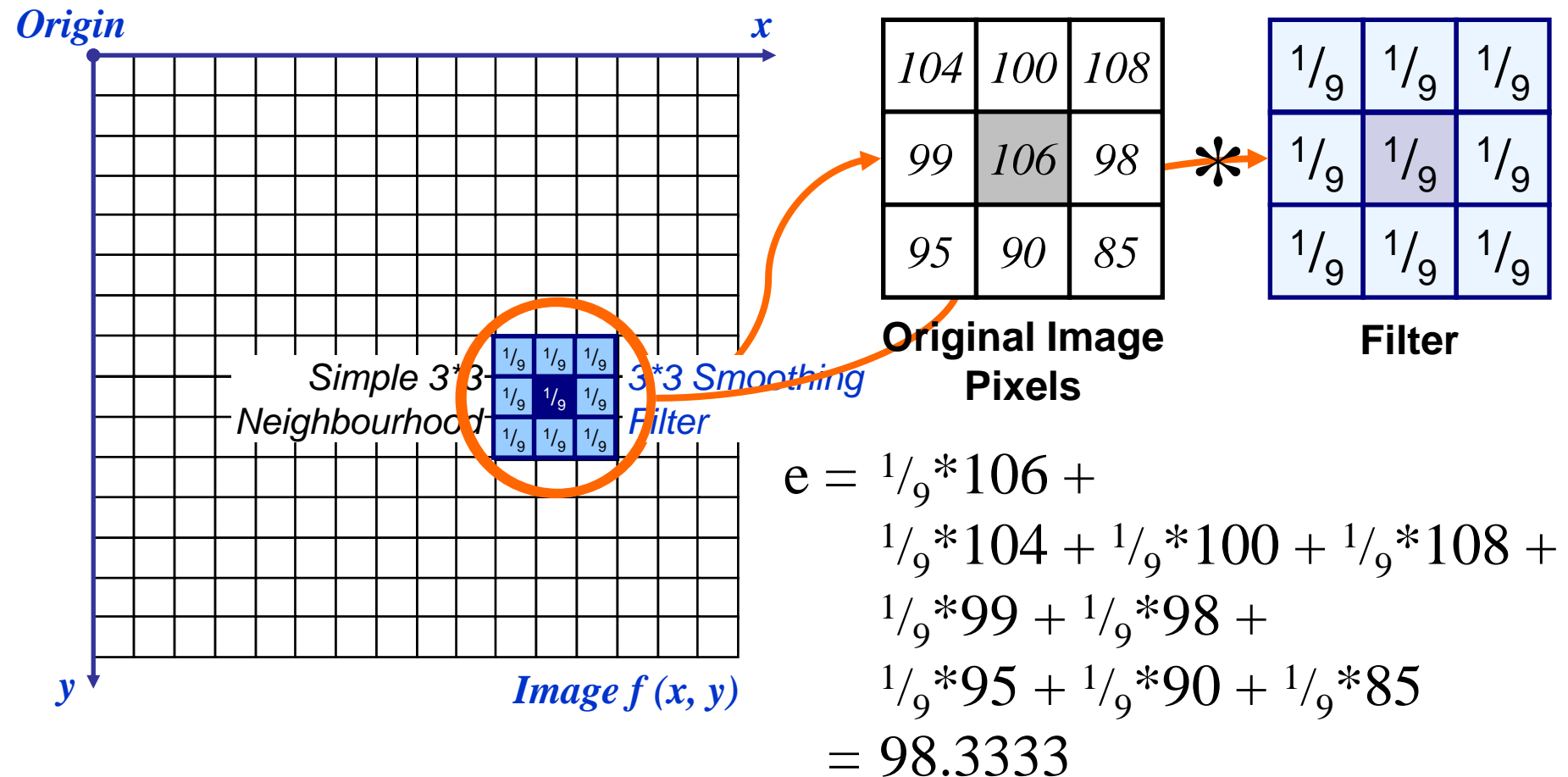
❖ Một trong số các toán tử lọc không gian đơn giản nhất là toán tử làm mịn ảnh.

- Đơn giản là lấy trung bình của tất cả các pixel trong vùng lân cận xung quanh giá trị tâm.
- Đặc biệt hữu ích trong việc loại bỏ nhiễu trong ảnh.

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

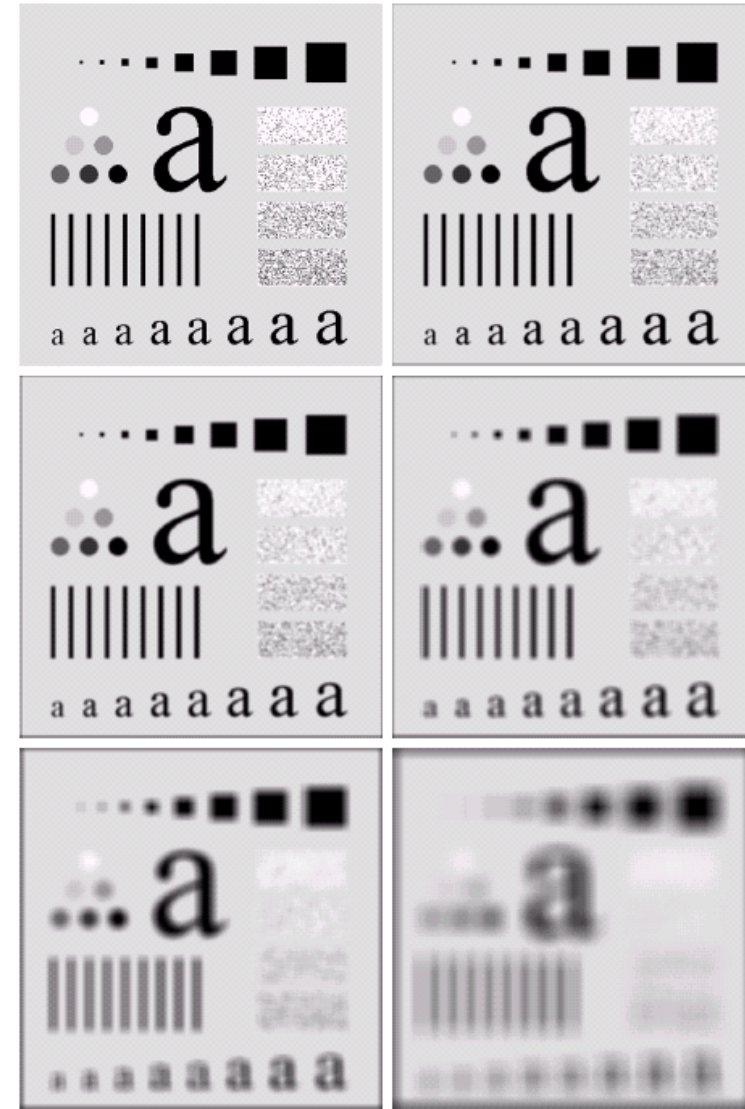
Bộ lọc
trung bình
đơn giản

Bộ lọc không gian làm mịn ảnh



Ví dụ làm mịn ảnh

- Ảnh phía trên bên trái là ảnh gốc kích thước 500*500
- Các ảnh sau là ảnh sau khi lọc bằng bộ lọc trung bình với kích thước tăng dần 3, 5, 9, 15 và 35
- Chú ý các chi tiết dần biến mất.



Bộ lọc trung bình có trọng số

➤ Có thể tạo ra bộ lọc làm mịn ảnh hiệu quả hơn bằng cách gán trọng số cho các pixel khác nhau của lân cận trong hàm trung bình.

- Các pixel gần pixel tâm có vai trò quan trọng hơn.
- Các bộ lọc này thường được gọi là trung bình có trọng số (*weighted averaging*)

$1/16$	$2/16$	$1/16$
$2/16$	$4/16$	$2/16$
$1/16$	$2/16$	$1/16$

Bộ lọc trung
bình có trọng số

Bộ lọc trung bình k giá trị gần nhất

- ❖ Cho ảnh I , pixel P , cửa sổ $W(P)$, ngưỡng θ , giá trị k .
- ❖ Thuật toán:
 - ❖ Bước 1: Tìm k giá trị nằm trong cửa sổ $W(P)$ gần với giá trị pixel P nhất.
 - ❖ Bước 2: Tính trung bình của k giá trị này:

$$AV_k(P) = (\sum_{i=1}^k P_i) / k$$

- ❖ Bước 3: Gán giá trị
$$I(P) = \begin{cases} I(P) & |I(P) - AV_k(P)| \leq \theta \\ AV_k(P) & |I(P) - AV_k(P)| > \theta \end{cases}$$

Ví dụ

- Cho ảnh I như sau:
- Cửa sổ $W(3 \times 3)$,
- Ngưỡng $\theta=2$, giá trị $k=3$

$$I = \begin{pmatrix} 1 & 2 & 3 & 2 \\ 4 & 16 & 2 & 1 \\ 4 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix} \quad I_{kq} = \begin{pmatrix} 1 & 2 & 3 & 2 \\ 4 & 8 & 2 & 1 \\ 4 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix}$$

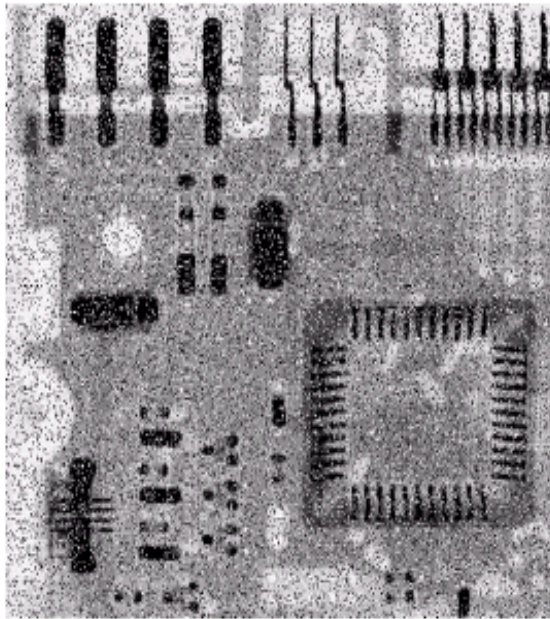
Bộ lọc trung vị (Median filter)

- ❖ Lọc median (lọc trung vị) được dùng khi nhiễu có các mức xám tách biệt. Ví dụ: nhiễu muối tiêu
- ❖ Lọc median là **lọc phi tuyến**. Mỗi pixel được thay thế bằng giá trị giữa trong các pixel láng giềng
- ❖ Giả sử $A = \{a_1, a_2, \dots, a_k\}$ là các giá trị pixel láng giềng với $a_1 \leq a_2 \leq \dots \leq a_k$, thì

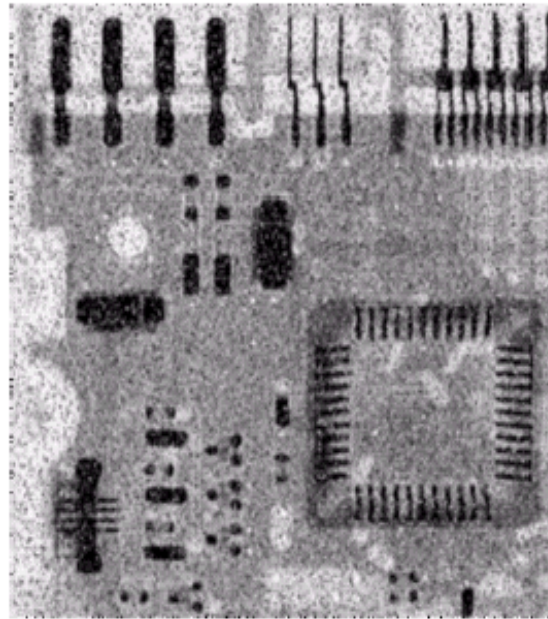
$$\text{median}(A) = \begin{cases} a_{k/2} & \text{với } k \text{ chẵn} \\ a_{(k+1)/2} & \text{với } k \text{ lẻ} \end{cases}$$

- Ví dụ: $A = \{0, 1, 2, \mathbf{6}, 8, 10, 13\}$ thì $\text{median}(A)=6$

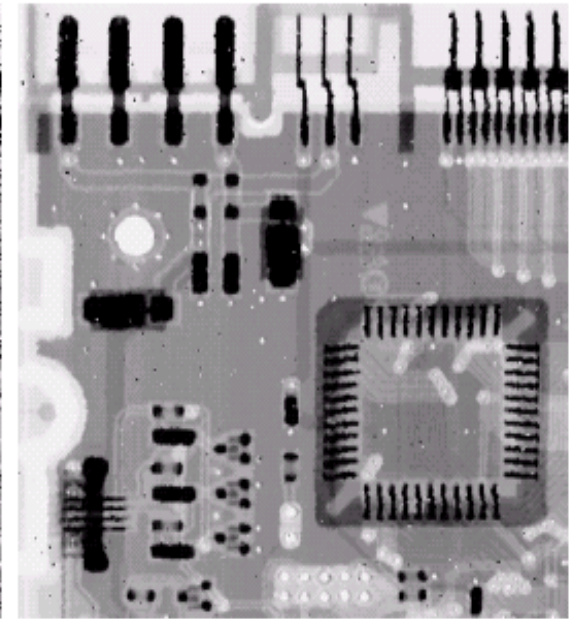
Ví dụ về bộ lọc trung bình và bộ lọc trung vị



Ảnh gốc bị nhiễu



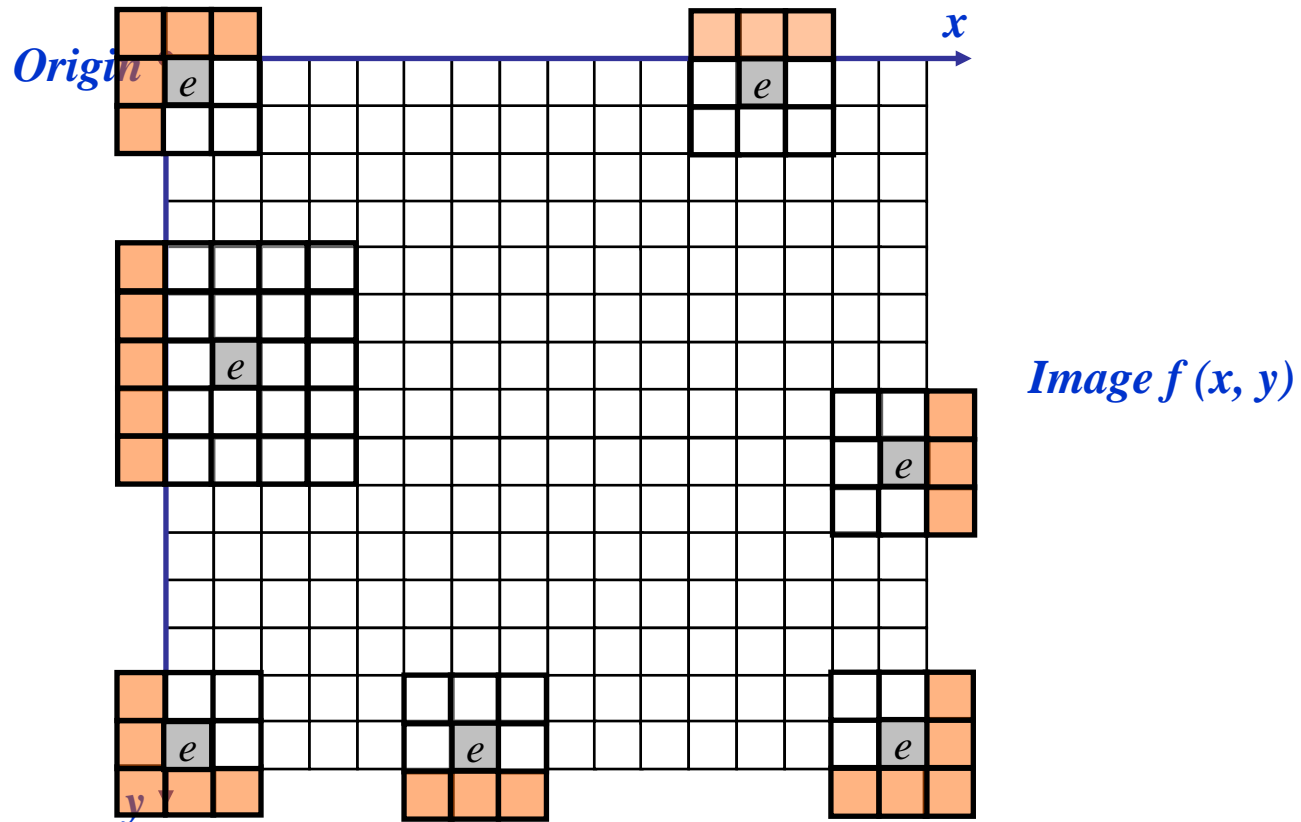
**Ảnh sau khi
lọc trung bình**



**Ảnh sau khi
lọc trung vị**

Xử lý biên ảnh

Tại biên của bức ảnh sẽ gặp trường hợp thiếu pixel ảnh để tạo thành neighbourhood.



Xử lý biên ảnh (tiếp)

Có một số cách tiếp cận để giải quyết vấn đề thiếu pixel ở biên:

- Loại bỏ pixel bị thiếu.
 - Chỉ làm việc với một số bộ lọc.
 - Làm thay đổi kích thước ảnh.
- Chèn thêm pixel vào ảnh
 - Thông thường thêm vào hoặc toàn pixel đen hoặc toàn pixel trắng.
 - Làm thay đổi tính chất ảnh.
- Lặp lại các pixels ở rìa ảnh.

Ví dụ

Cho bức ảnh bị nhiễu sau. Dùng bộ lọc trung bình và trung vị kích thước 3×3 để loại bỏ nhiễu trong ảnh.

2	7	3	0
3	1	6	3
0	1	3	5
3	6	7	1

KHÔI PHỤC ẢNH

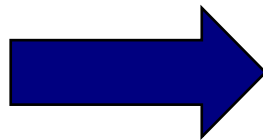
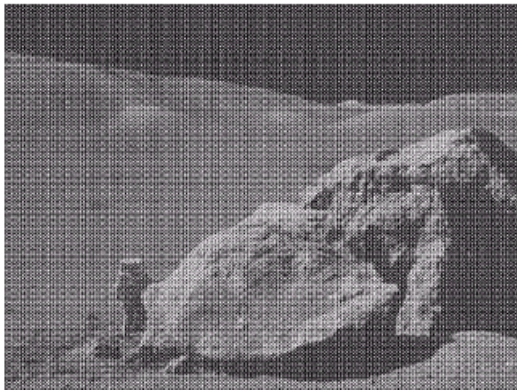
Nội dung

- Thế nào là khôi phục ảnh?
- Nhiễu và ảnh
- Các mô hình nhiễu
- Loại bỏ nhiễu sử dụng bộ lọc không gian
- Nhiễu tuần hoàn và loại bỏ nhiễu sử dụng bộ lọc miền tần số.

Thế nào là khôi phục ảnh?

Khôi phục ảnh nỗ lực tái tạo lại bức ảnh đã bị xuống cấp về chất lượng.

- Xác định quá trình làm bức ảnh xuống cấp và nỗ lực làm ngược lại.
- Tương tự như tăng cường ảnh nhưng có mục đích hơn.



Nhiều và ảnh

Nguồn nhiễu trong ảnh số thường bắt nguồn từ quá trình thu nhận ảnh (số hóa) và truyền ảnh.

- Sensor thu nhận ảnh có thể bị ảnh hưởng bởi các điều kiện xung quanh.
- Ảnh cũng có thể bị can nhiễu trong quá trình truyền dẫn.



Mô hình nhiễu

- Ta có thể mô hình hóa ảnh nhiễu như sau:

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

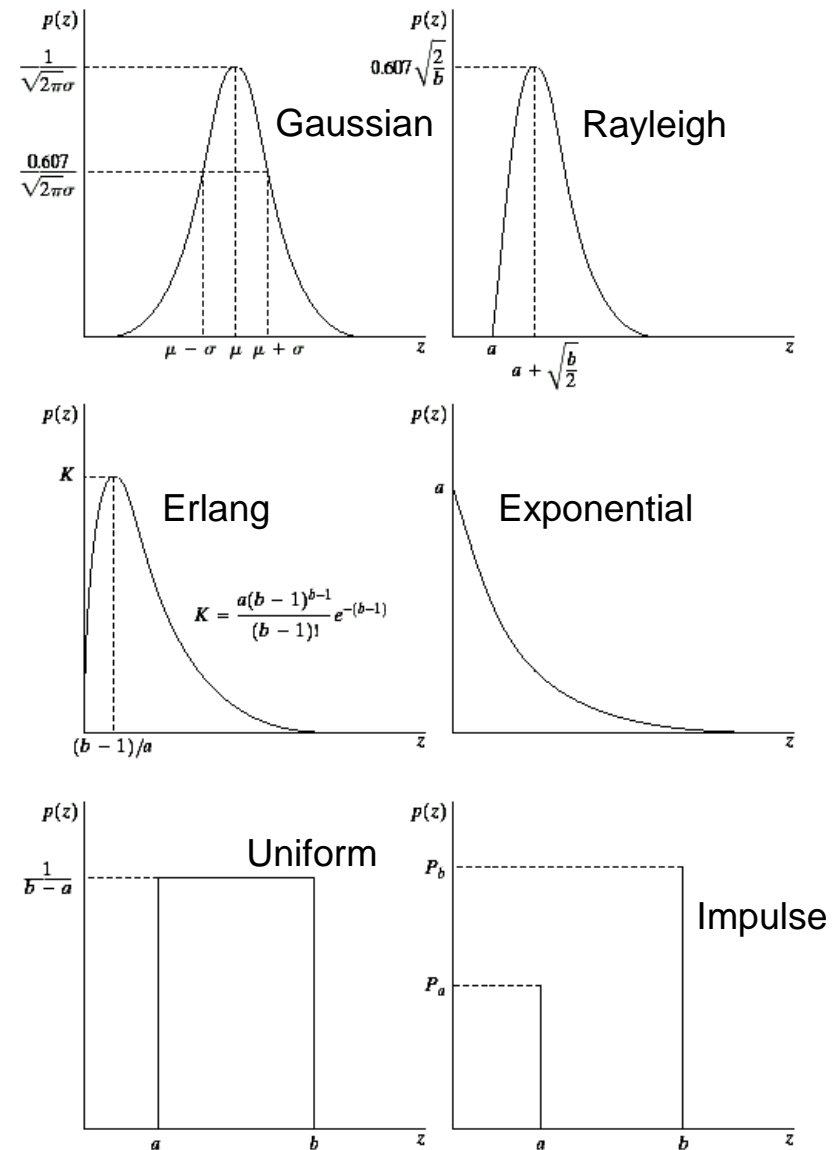
ở đó $f(x, y)$ là pixel ảnh gốc, $\eta(x, y)$ là nhiễu, $g(x, y)$ là pixel bị nhiễu.

- Nếu ta có thể ước đoán mô hình nhiễu của bức ảnh thì có thể đưa ra cách khôi phục bức ảnh.

Các mô hình nhiễu

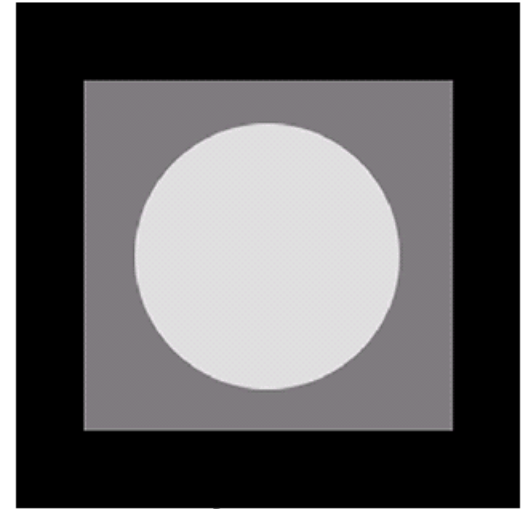
Có nhiều mô hình nhiễu khác nhau sử dụng cho ảnh: $\eta(x, y)$

- Gaussian
 - Mô hình thông dụng nhất
- Rayleigh
- Erlang
- Exponential (hàm mũ)
- Uniform (đồng nhất)
- Impulse (hàm xung)
 - *Salt and pepper* noise

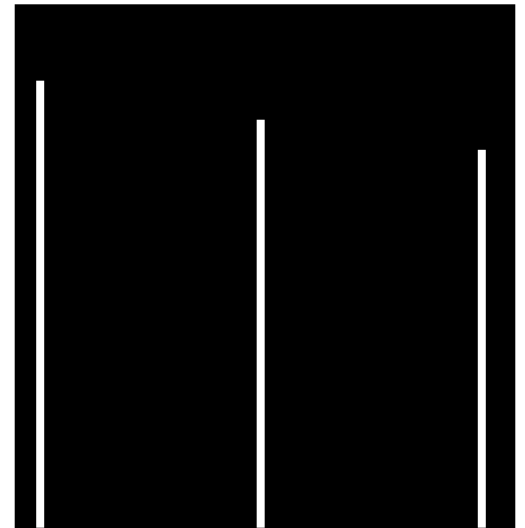


Ví dụ về nhiễu

- Ảnh bên phải là ảnh gốc
- Các slide tiếp theo mô tả kết quả của quá trình thêm nhiễu vào ảnh với các mô hình khác nhau.

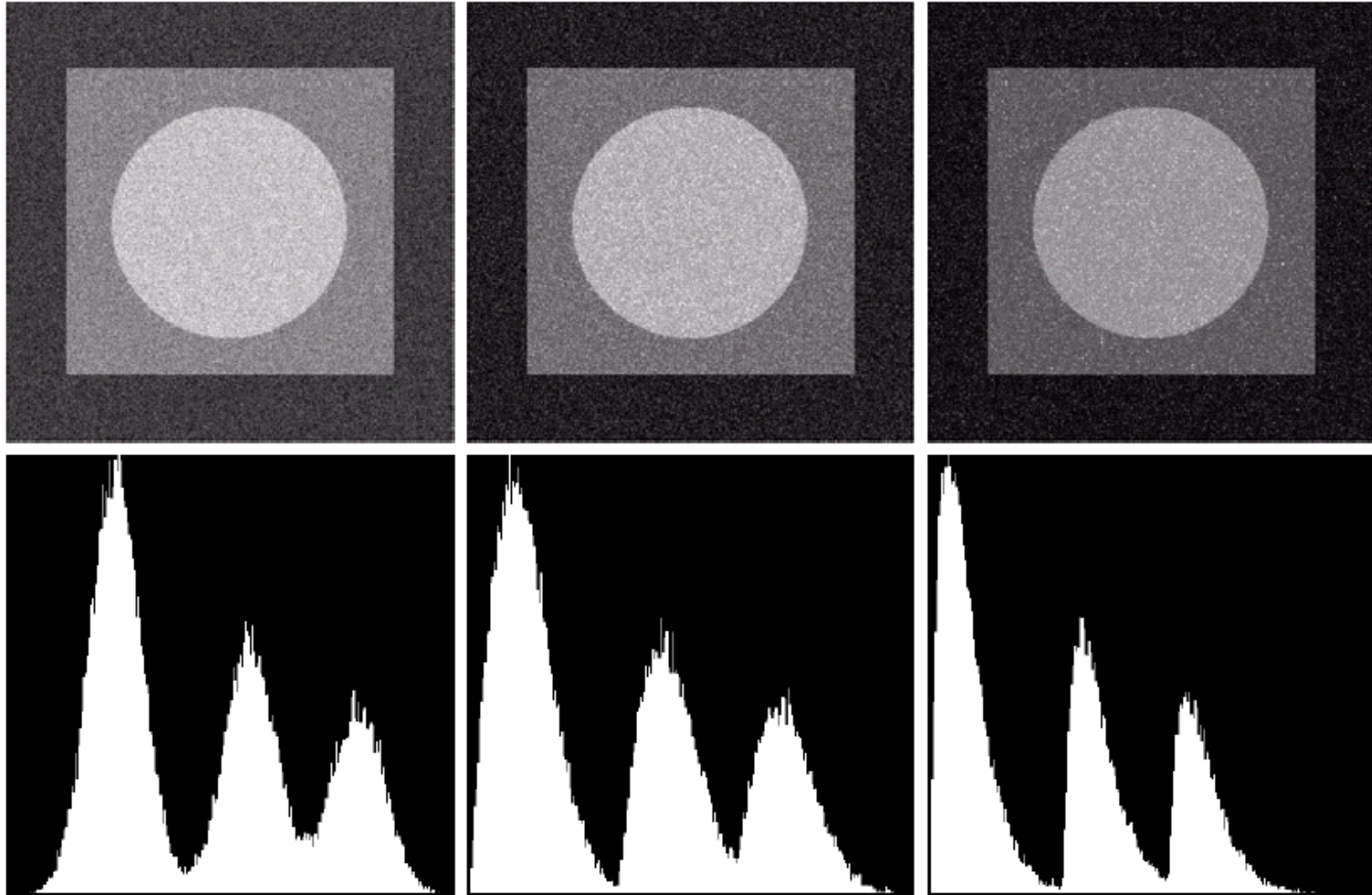


Ảnh gốc



Histogram 63

Ví dụ về nhiễu (tiếp)

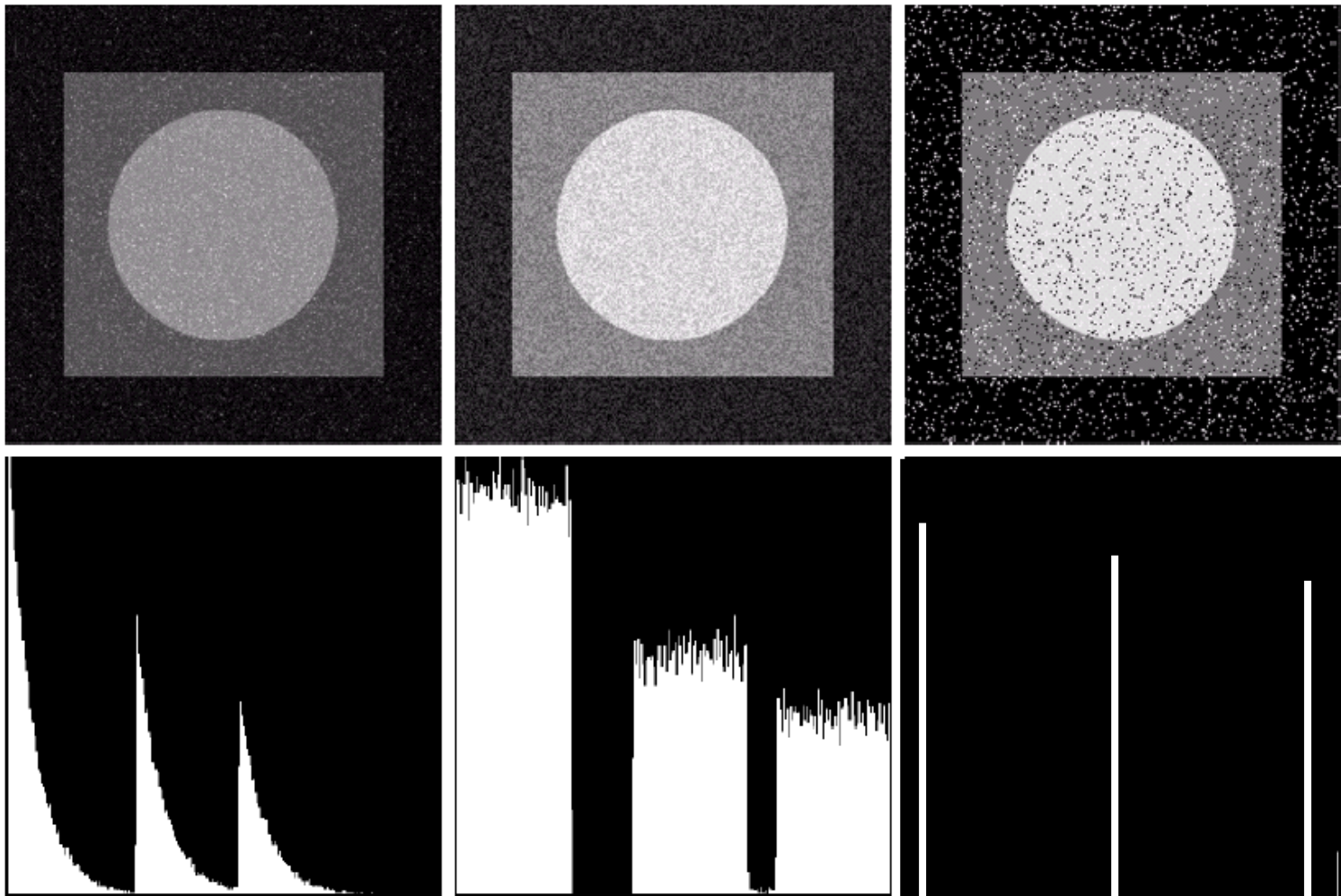


Gaussian

Rayleigh

Erlang

Ví dụ về nhiễu (tiếp)



Exponential

Uniform

Impulse

Lọc để loại bỏ nhiễu

- Ta có thể sử dụng các loại bộ lọc không gian khác nhau để loại bỏ các loại nhiễu khác nhau.
- Bộ lọc trung bình số học (*arithmetic mean filter*) là bộ lọc rất đơn giản và được tính như sau:

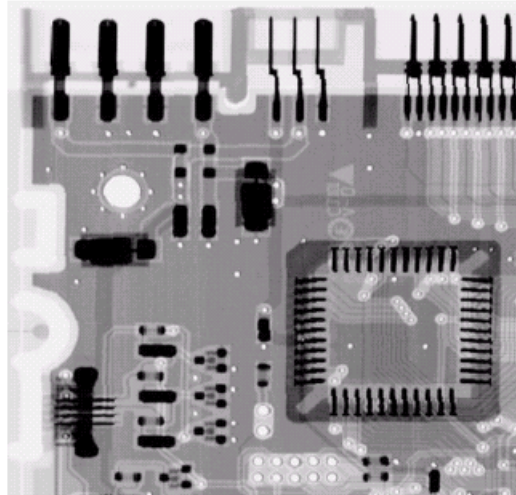
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

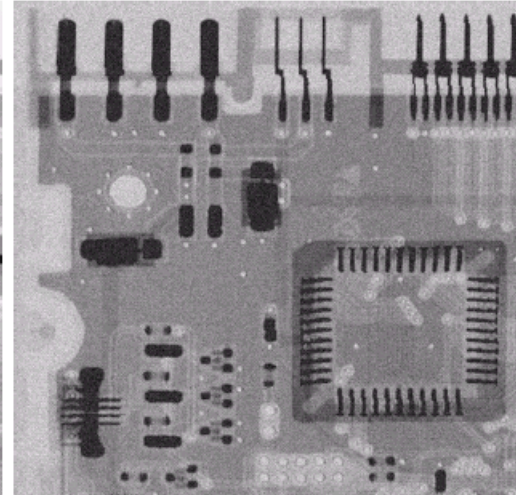
- Bộ lọc này được thực hiện như bộ lọc làm mịn ảnh đơn giản.
- Làm mờ ảnh để loại bỏ nhiễu.

Ví dụ về loại bỏ nhiễu

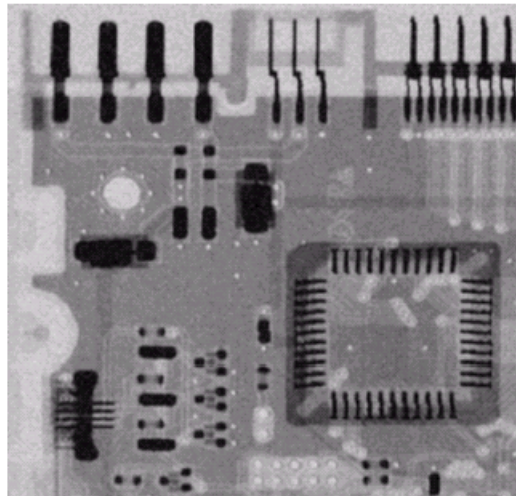
Ảnh gốc



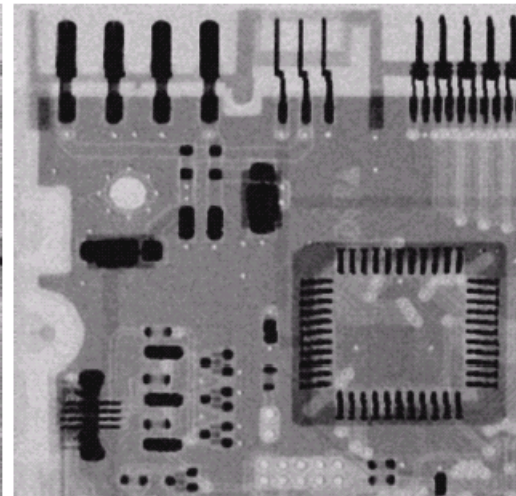
Ảnh bị nhiễu
Gaussian



Sau khi lọc
bằng bộ lọc
trung bình
số học 3*3



Sau khi lọc
bằng bộ lọc
trung bình
hình học 3*3



Một số bộ lọc không gian khác

- Bộ lọc trung vị (Median filter)
- Bộ lọc max và min (Max and min filter)
- Bộ lọc điểm giữa (Midpoint filter)

Bộ lọc trung vị

$$\hat{f}(x, y) = \underset{(s,t) \in S_{xy}}{\text{median}}\{g(s, t)\}$$

- Loại bỏ nhiễu tốt mà không có hiệu ứng làm mờ ảnh như các bộ lọc làm mịn khác.
- Đặc biệt hiệu quả với nhiễu muối tiêu.

Bộ lọc Max và Min

Bộ lọc Max:

$$\hat{f}(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\}$$

Bộ lọc Min:

$$\hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\}$$

- Bộ lọc Max hiệu quả với nhiễu tiêu và bộ lọc min hiệu quả với nhiễu muối.

Midpoint Filter

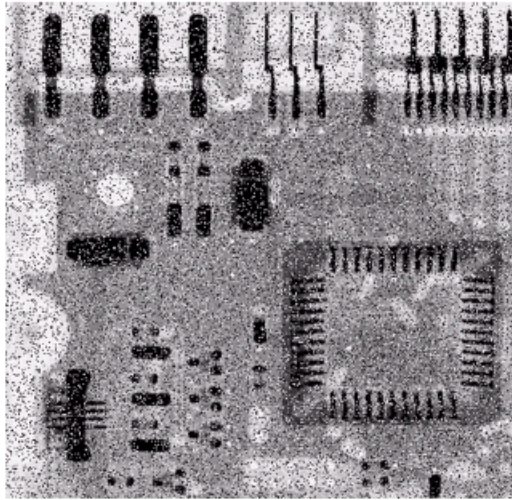
Midpoint Filter:

$$\hat{f}(x, y) = \frac{1}{2} \left[\max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\} \right]$$

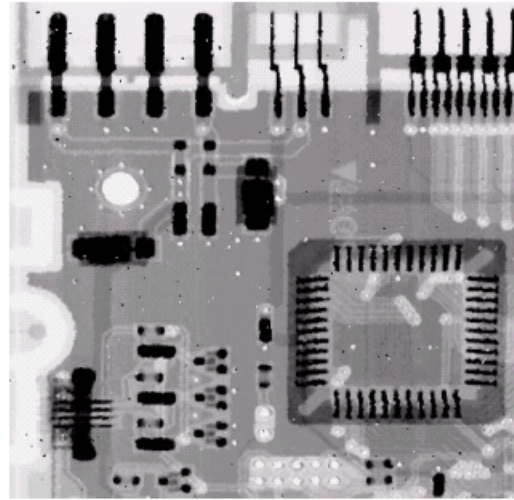
- Hiệu quả với nhiễu Gaussian ngẫu nhiên và nhiễu đồng nhất.

Ví dụ về loại bỏ nhiễu

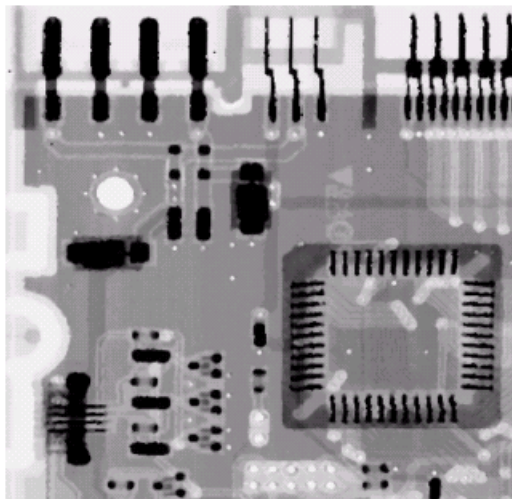
Image
Corrupted
By Salt And
Pepper Noise



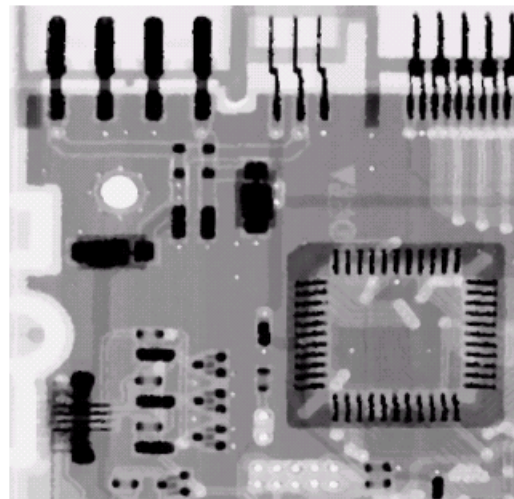
Result of 1
Pass With A
3*3 Median
Filter



Result of 2
Passes With
A 3*3 Median
Filter



Result of 3
Passes With
A 3*3 Median
Filter



Ví dụ về loại bỏ nhiễu (tiếp)

Image
Corrupted
By Pepper
Noise

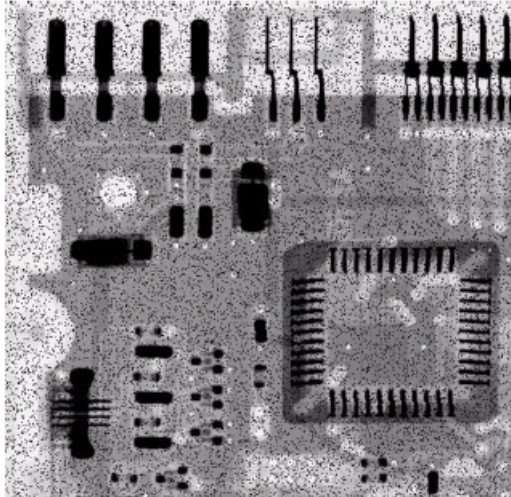
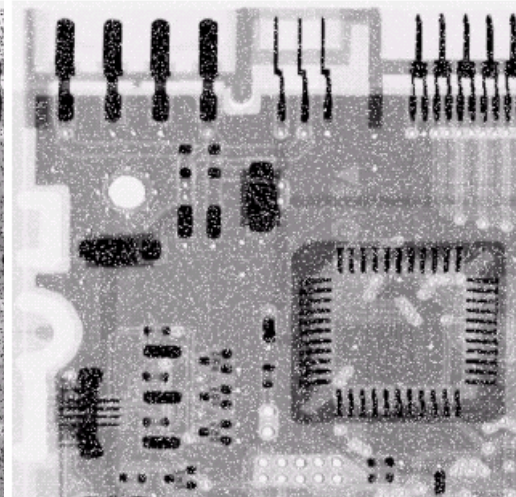
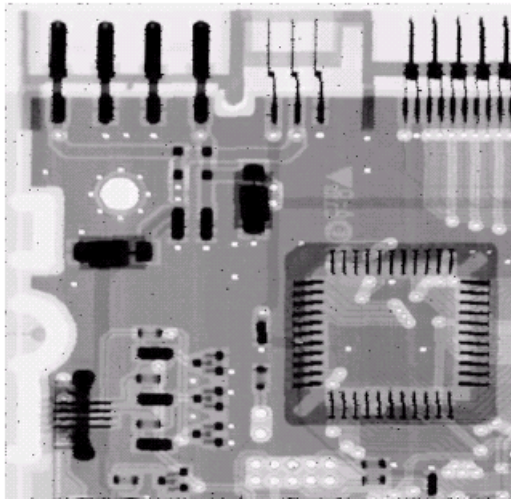


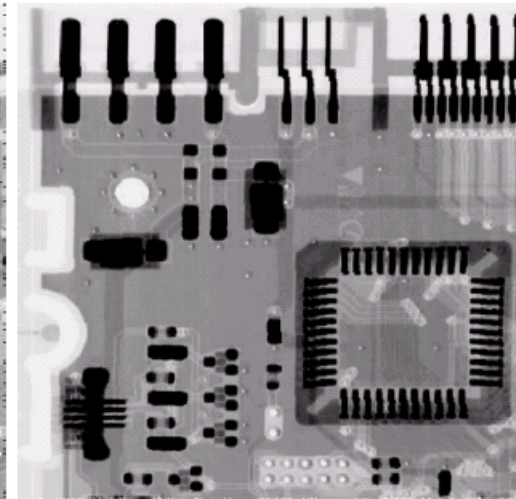
Image
Corrupted
By Salt
Noise



Result Of
Filtering
Above
With A 3*3
Max Filter



Result Of
Filtering
Above
With A 3*3
Min Filter



Ví dụ về loại bỏ nhiễu (tiếp)

Image
Corrupted
By Uniform
Noise

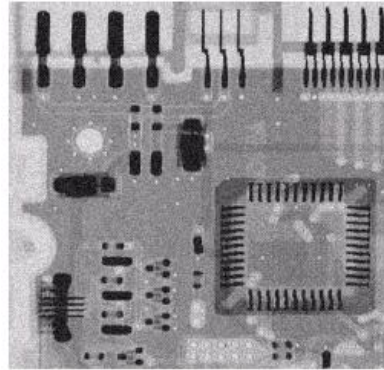
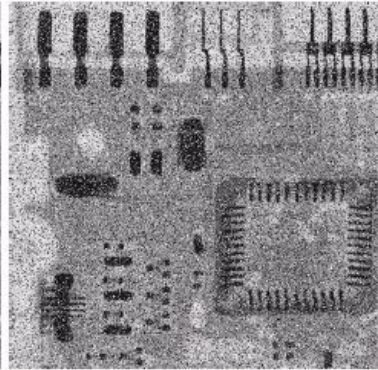
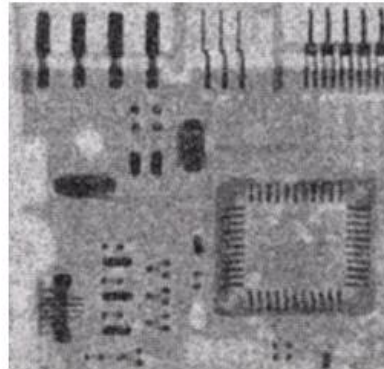


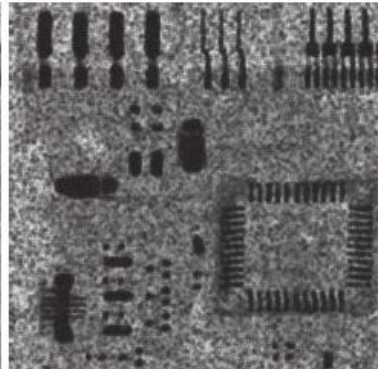
Image Further
Corrupted
By Salt and
Pepper Noise



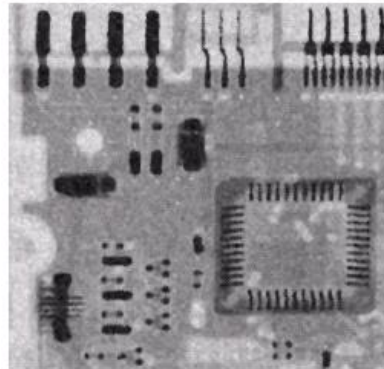
Filtered By
 5×5 Arithmetic
Mean Filter



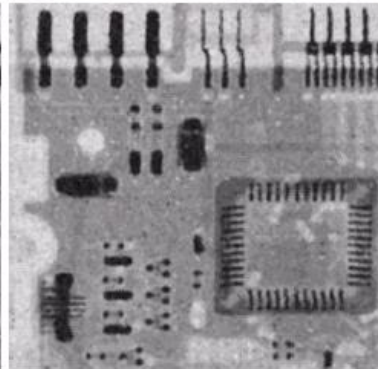
Filtered By
 5×5 Geometric
Mean Filter



Filtered By
 5×5 Median
Filter

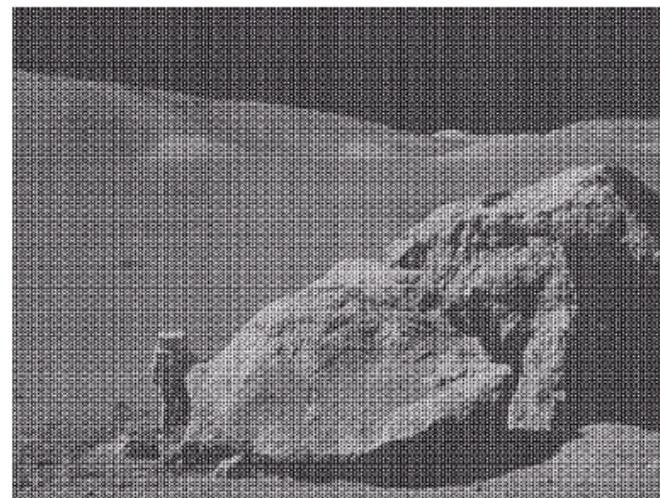


Filtered By
 5×5 Alpha-Trimmed
Mean Filter



Nhiều tuần hoàn

- Thường xuất hiện do can nhiễu điện hoặc điện từ.
- Thường tạo ra các mẫu nhiễu điển hình trong bức ảnh.
- Để loại bỏ nhiễu này thì sử dụng các kỹ thuật miền tần số là hiệu quả nhất.



Kết luận

- ❖ Khôi phục ảnh là quá trình xử lý có mục đích hơn so với tăng cường ảnh.
- ❖ Các kỹ thuật miền không gian đặc biệt hữu ích đối với việc loại bỏ nhiễu ngẫu nhiên.
- ❖ Các kỹ thuật miền tần số đặc biệt hữu ích đối với nhiễu tuần hoàn.