



# ỨNG DỤNG CỦA PHƯƠNG PHÁP TÍCH CHẬP

---

LỌC ẢNH - FILTER



# Nhắc lại thao tác tính Tích chập

Nếu nhân tích chập chập có dạng:

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

$y-1$			
$y$			
$y+1$			
	$x-1$	$x$	$x+1$

Thì ảnh kết quả

$$g(x,y) = w_1 * f(x-1,y-1) + w_2 * f(x,y-1) + w_3 * f(x+1,y-1) + \\ w_4 * f(x-1,y) + w_5 * f(x,y) + w_6 * f(x+1,y) + \\ w_7 * f(x-1,y+1) + w_8 * f(x,y+1) + w_9 * f(x+1,y+1)$$

# 1. Bộ lọc làm trơn ảnh

- ❑ Bộ lọc làm trơn ảnh (Smoothing/Low Pass Filter)
- ❑ Khi tịnh tiến cửa sổ, pixel trung tâm được thay thế bằng trị trung bình của các pixel bên trong cửa sổ.
- ❑ Bộ lọc làm trơn ảnh được sử dụng để:
  - **Làm mờ ảnh (Blurring):** Thường được dùng trong bước tiền xử lý để loại bỏ các chi tiết nhỏ không mong muốn trước khi thực hiện rút trích các đối tượng liên quan (kích thước lớn), nối các đoạn gấp khúc trong đường cong.
  - **Giảm nhiễu**

# 1. Bộ lọc làm trơn ảnh



Raw data reconstruction with quantization error



Scan after application of smoothing filter

# 1. Bộ lọc làm trơn ảnh

Một số cửa sổ lọc

$$L_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$L_2 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$L_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$L_4 = \frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$L_5 = \frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$L_6 = \frac{1}{32} \begin{bmatrix} 1 & 3 & 1 \\ 3 & 16 & 3 \\ 1 & 3 & 1 \end{bmatrix}$$

# 1. Bộ lọc làm trơn ảnh

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

Mean Filter Kernel

IDRISI adds an extra row/column with the values of the adjacent cells for computation

Input Image

2	2	4	4	5	3	3	3	2
2	2	4	4	5	3	3	3	2
7	7	3	3	2	1	1	1	7
3	3	1	1	11	1	2	2	
3	3	4	4	4	3	2	2	
6	6	6	4	4	3	9		
6	6	6	6	6	3			
6	6	6	6	6				

$$\begin{aligned} &1/9 * 2 + 1/9 * 2 + 1/9 * 4 + \\ &1/9 * 2 + 1/9 * 2 + 1/9 * 4 + \\ &1/9 * 7 + 1/9 * 7 + 1/9 * 3 = 4 \end{aligned}$$

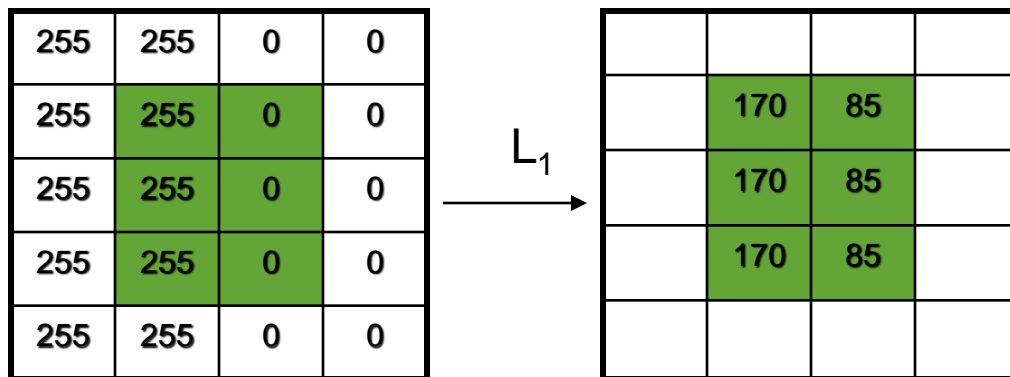
Output Image

4	4	4	3	3	2	
4	3	4	3	3	2	
4	3	4	3	3	2	
4	4	4	4	4	3	
5	5	5	4	4		
6	6	6	5	4		

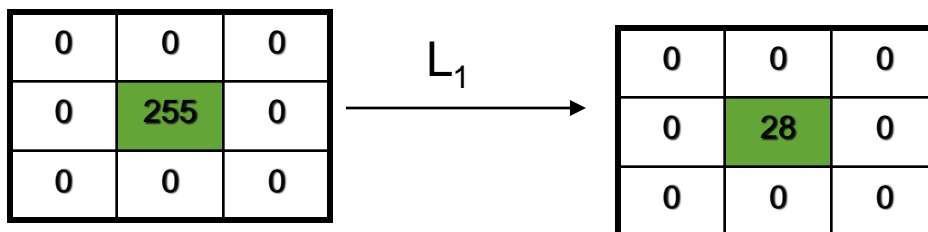
# 1. Bộ lọc làm trơn ảnh

Nhược điểm:

i. Làm mờ cạnh ảnh

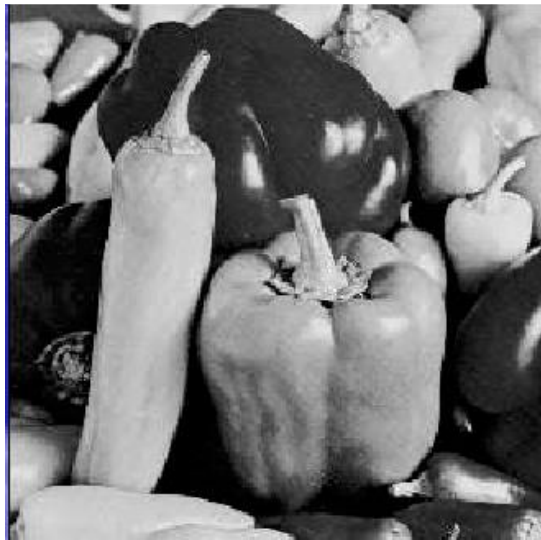


ii. Chỉ làm suy yếu chứ không thể loại những pixel nhiễu có giá trị rất lớn (hoặc rất bé) so với các pixel lân cận.

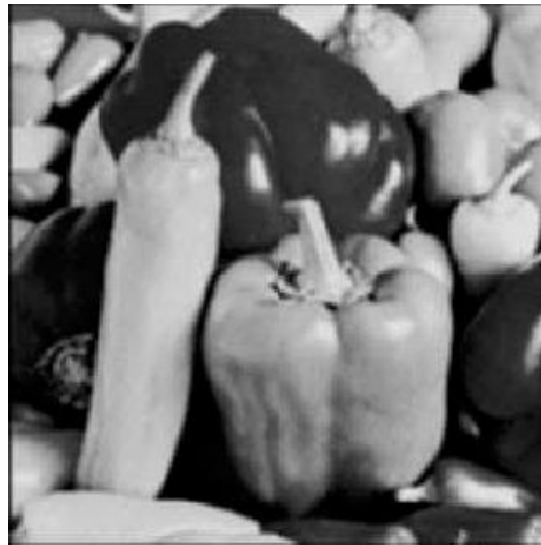


Bộ lọc này chỉ thích hợp với tín hiệu một chiều và tốt trong xử lý tín hiệu số

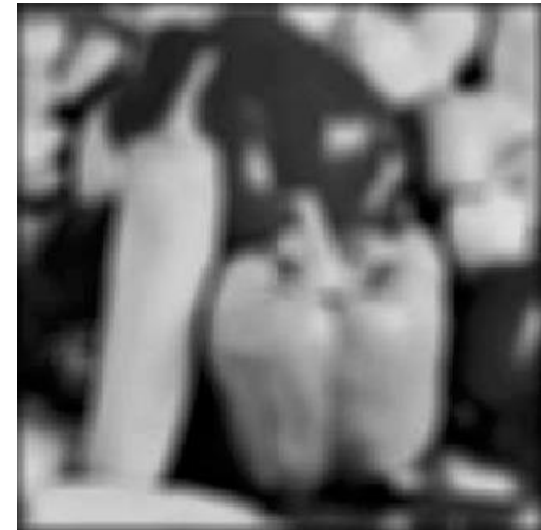
# 1. Bộ lọc làm trơn ảnh



Ảnh gốc



$L_1$  với  $N=3$



$L_1$  với  $N=11$

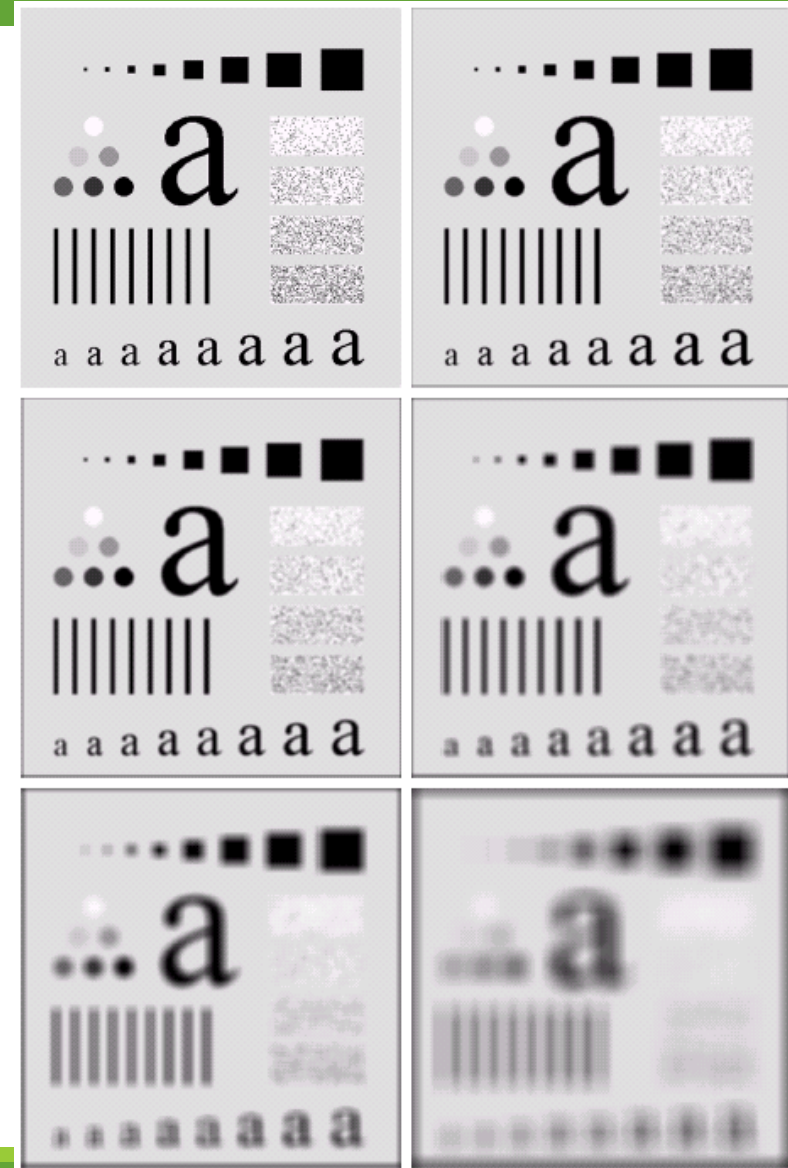


$L_1$  với  $N=5$

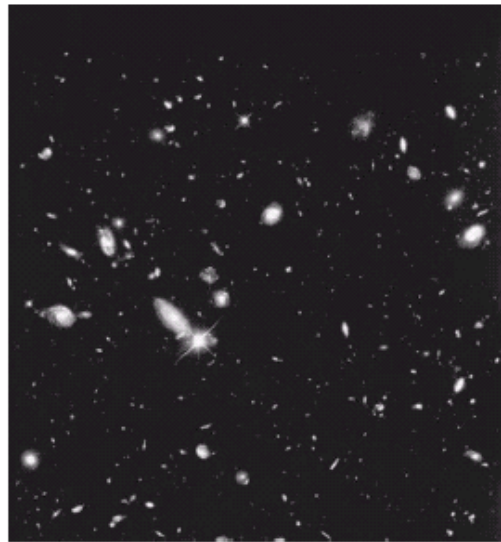


# 1. Bộ lọc làm trơn ảnh

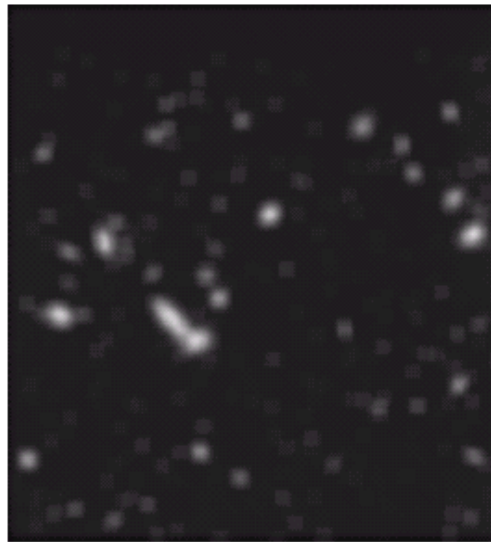
- The image at the top left is an original image of size 500\*500 pixels
- The subsequent images show the image after filtering with an averaging filter of increasing sizes
  - 3, 5, 9, 15 and 35
- Notice how detail begins to disappear



# 1. Bộ lọc làm trơn ảnh



**Original Image**

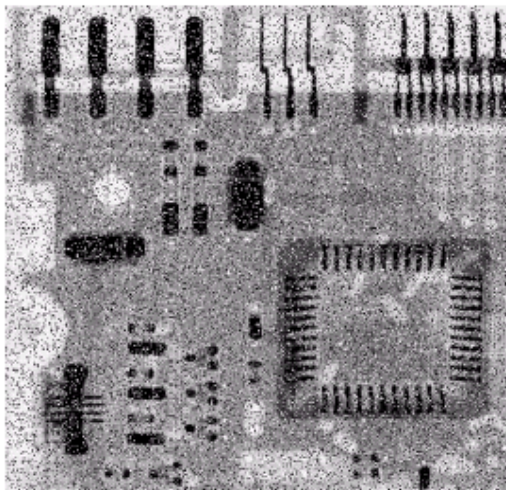


**Smoothed Image**

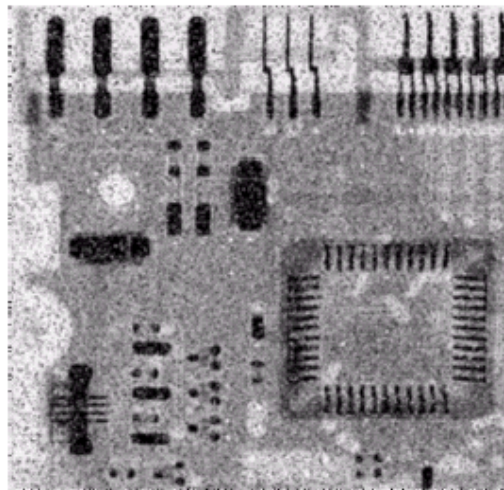


**Thresholded Image**

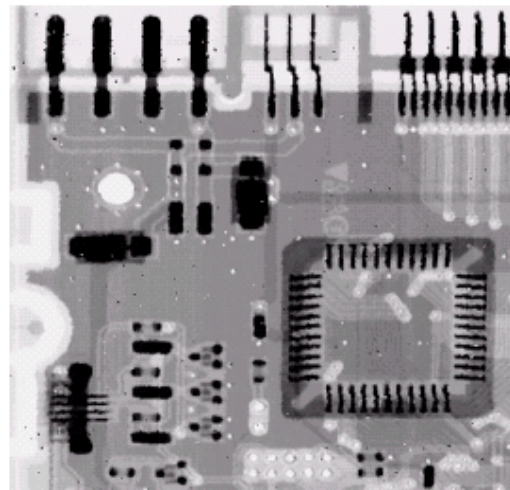
# 1. Bộ lọc làm trơn ảnh



**Original Image  
With Noise**

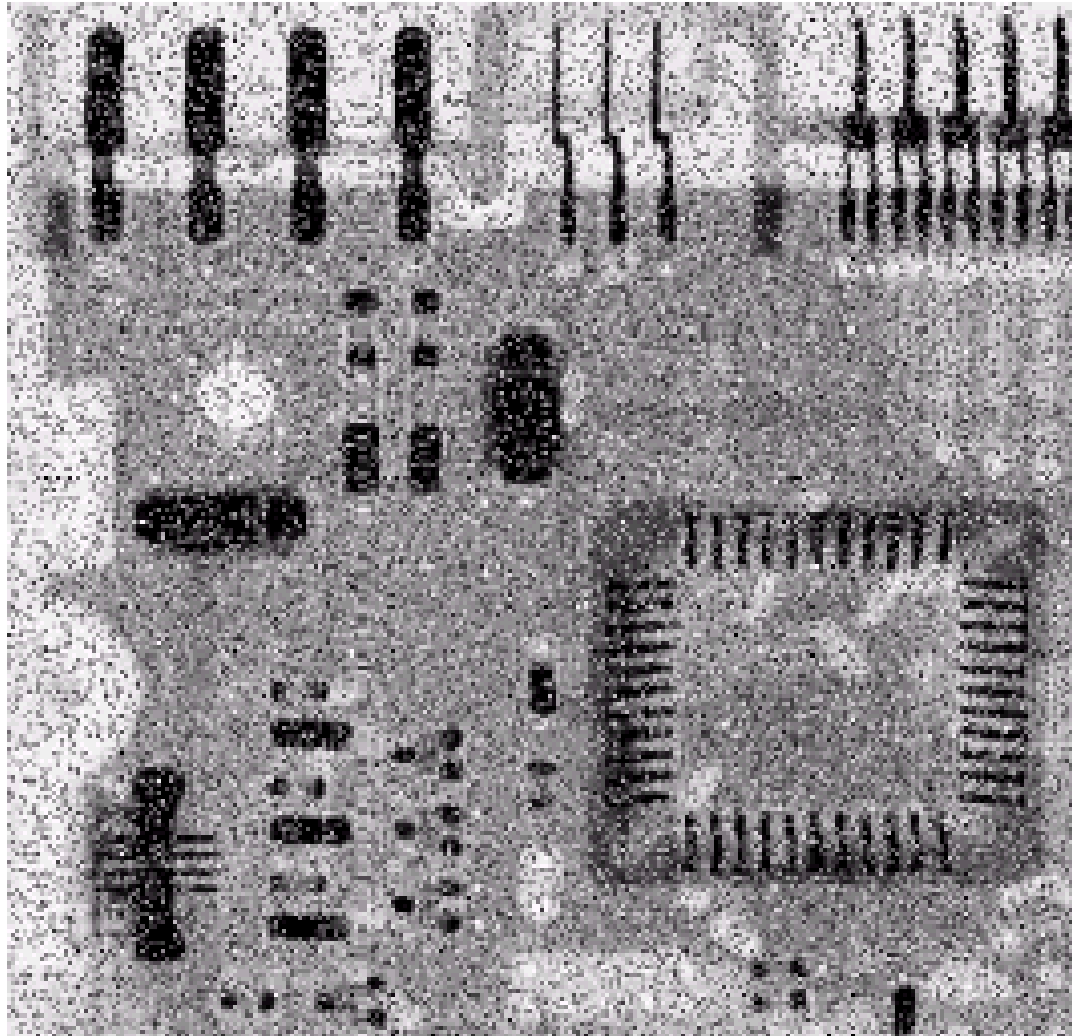


**Image After  
Averaging Filter**

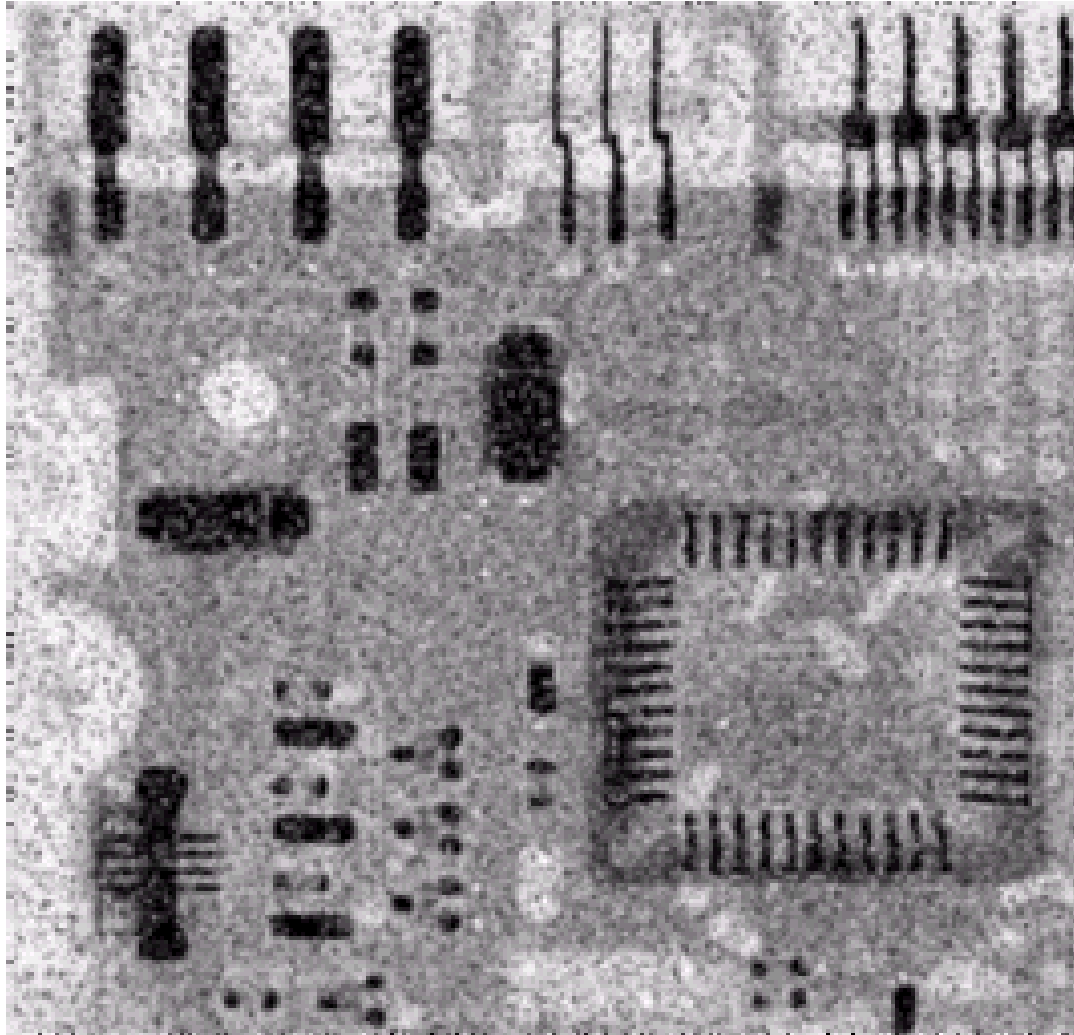


**Image After  
Median Filter**

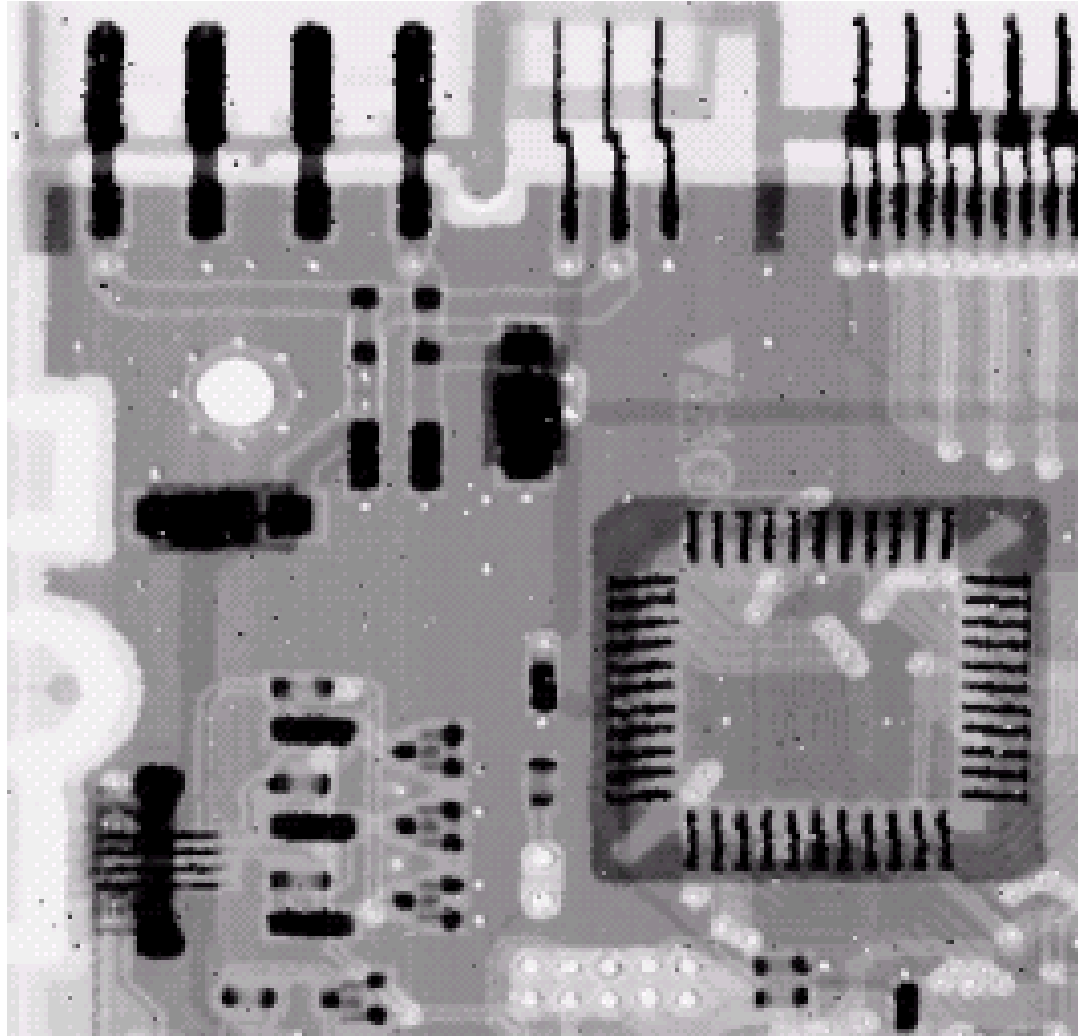
# 1. Bộ lọc làm trơn ảnh



# 1. Bộ lọc làm trơn ảnh



# 1. Bộ lọc làm trơn ảnh



## 2. Bộ lọc làm sắc ảnh

- ❑ Bộ lọc làm sắc ảnh (Sharpening/High Pass Filter)
- ❑ Dùng làm nổi bật các chi tiết ảnh.
- ❑ Nhân tích chập có hệ số dương ở tâm, các hệ số âm ở xung quanh.
- ❑ Thông thường, tổng các hệ số trong nhân bằng 1

## 2. Bộ lọc làm sắc ảnh

$$S_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

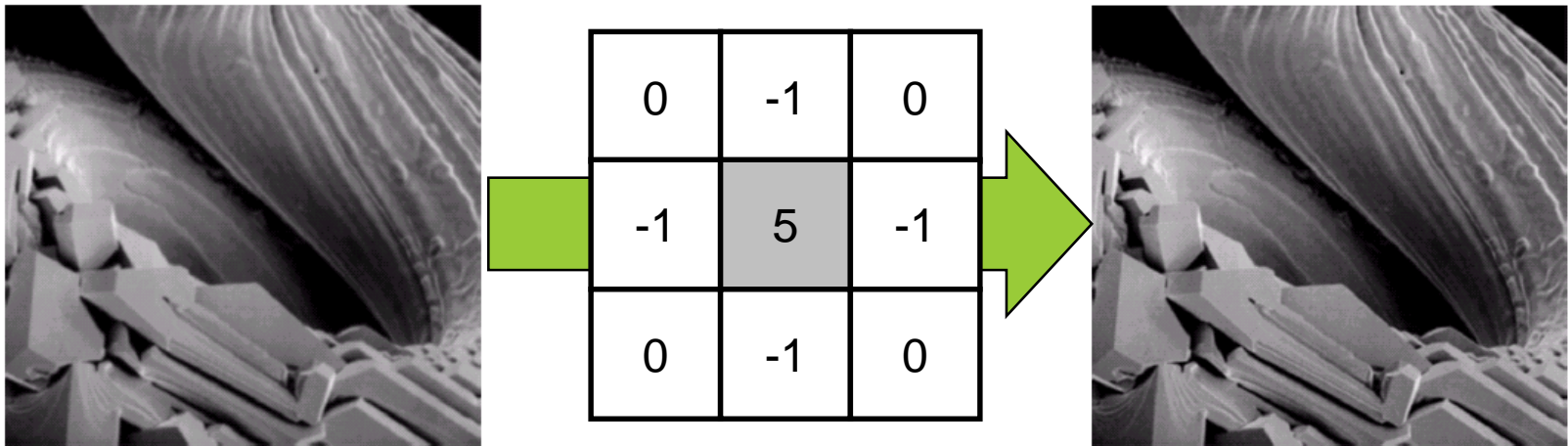
$$S_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

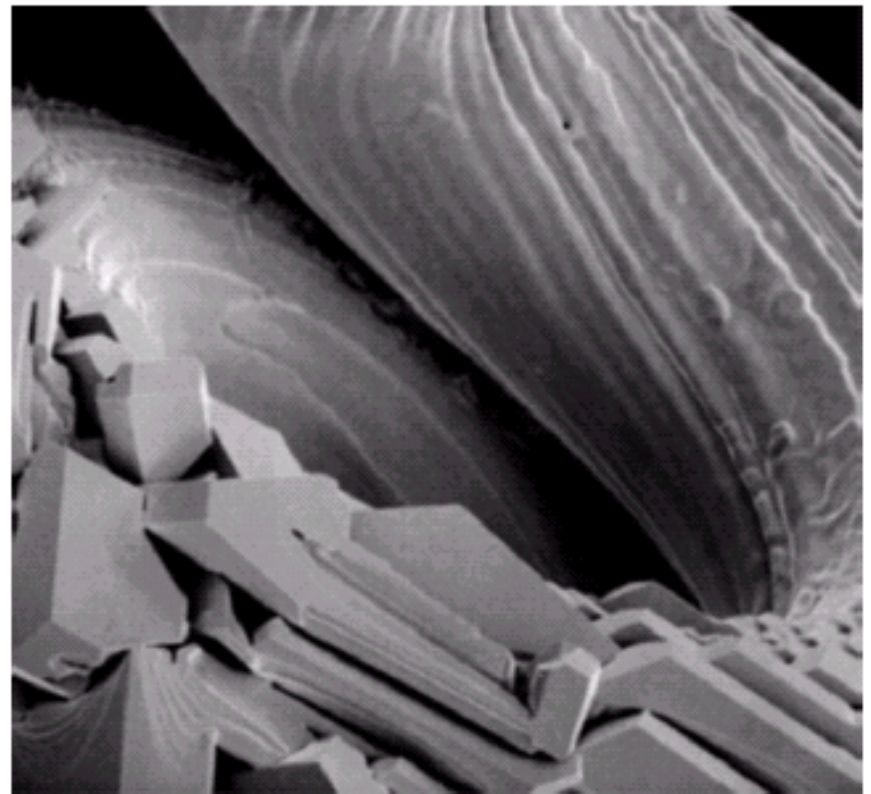
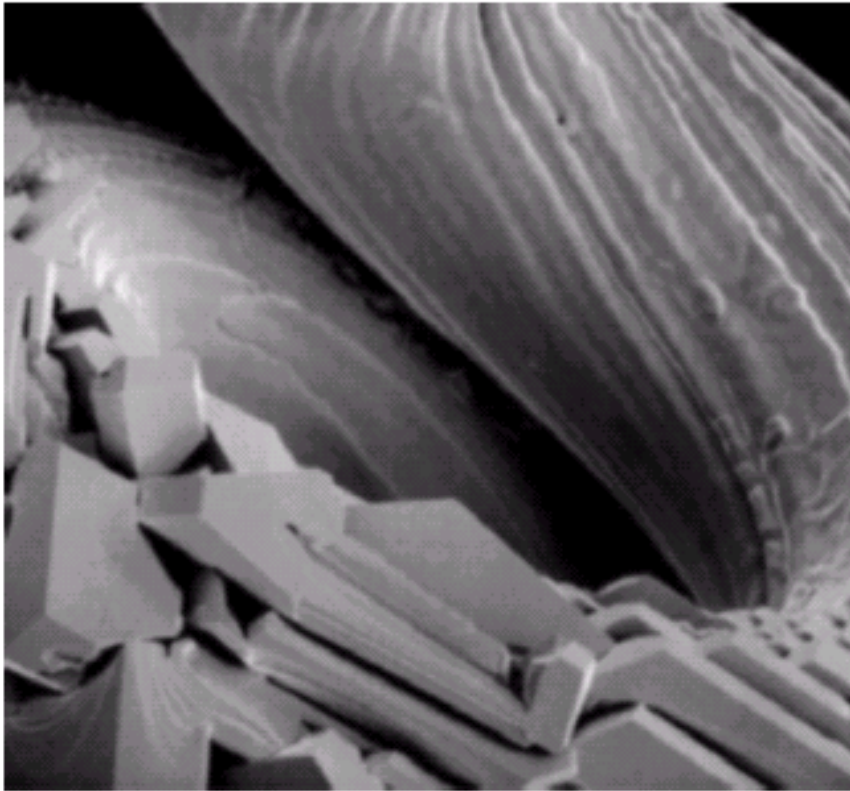




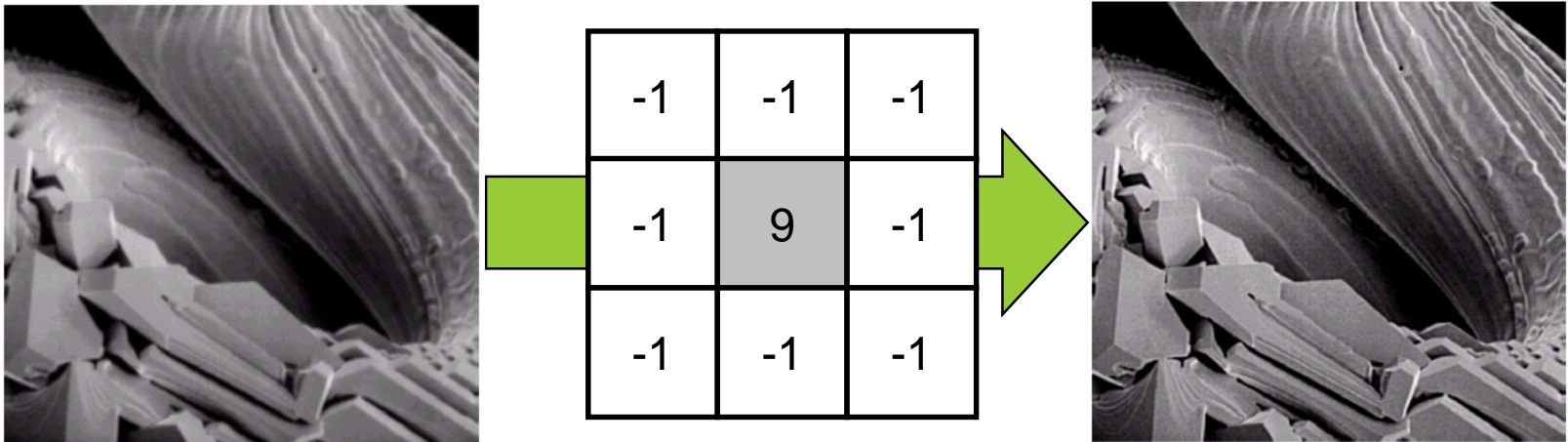
## 2. Bộ lọc làm sắc ảnh



## 2. Bộ lọc làm sắc ảnh



## 2. Bộ lọc làm sắc ảnh



## 2. Bộ lọc làm sắc ảnh

Unsharp Masking (High-boost filtering): được dùng để làm sắc ảnh. Nó lấy ảnh nguồn trừ đi ảnh đã được làm trơn:

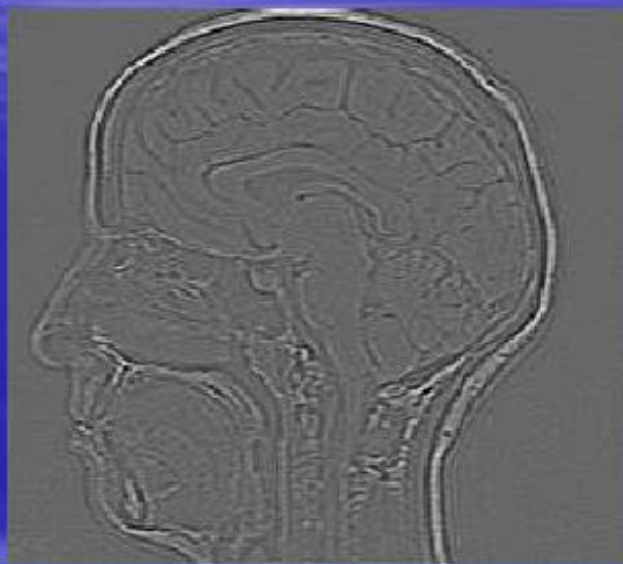
$$\text{Ảnh\_đích} = k_1(\text{Ảnh\_nguồn}) - k_2(\text{Ảnh\_đã\_được\_làm\_trơn})$$

- Trong đó  $k_1 + k_2 = 1$ . Thông thường thì  $k_1:k_2$  được lấy từ 1,5:1 đến 5:1.

## 2. Bộ lọc làm sắc ảnh



Original Image



Highpass filtering



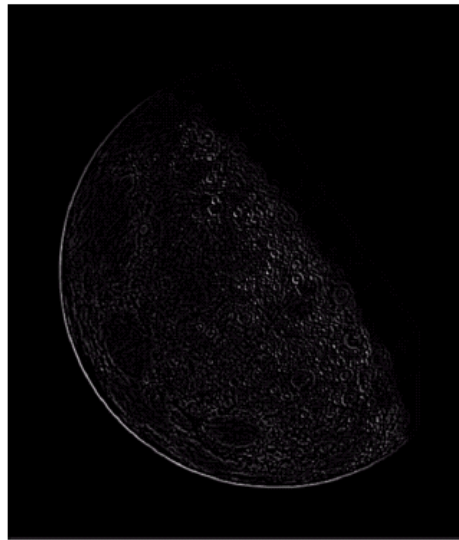
High-boost filtering

## 2. Bộ lọc làm sắc ảnh



Original  
Image

-



Laplacian  
Filtered Image

=



Sharpened  
Image



## 2. Bộ lọc làm sắc ảnh



# 3. Bộ lọc hướng

## Bộ lọc hướng (Directional Filter)

Bộ lọc này dựa trên 4 hướng chính:  $0^0$ ,  $45^0$ ,  $90^0$  và  $135^0$ . Hướng lọc là hướng có xác suất là cạnh ảnh lớn nhất/ bé nhất.

- Ta định nghĩa 2 hàm  $c(\Phi)$  và  $s(\Phi)$  như sau:

$\Phi$	$0^0$	$45^0$	$90^0$	$135^0$
$c(\Phi)$	1	1	0	-1
$s(\Phi)$	0	1	1	1

- Định nghĩa hàm lọc  $h(i,j,\emptyset)$ :
  - $h(c(\emptyset),s(\emptyset),\Phi)=0.5$
  - $h(c(\Phi),s(\Phi),\Phi)=h(-c(\Phi),-s(\Phi),\Phi)=0.25$



### 3. Bộ lọc hướng

Định nghĩa trên sẽ tương đương với định nghĩa của 4 nhân tích chập:

$$D_1 = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$0^\circ$

$$D_2 = \frac{1}{4} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$45^\circ$

$$D_3 = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$90^\circ$

$$D_4 = \frac{1}{4} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$135^\circ$

$$0 \begin{bmatrix} & & \\ & 0.5 & \\ & & \end{bmatrix} 0$$

# 3. Bộ lọc hướng

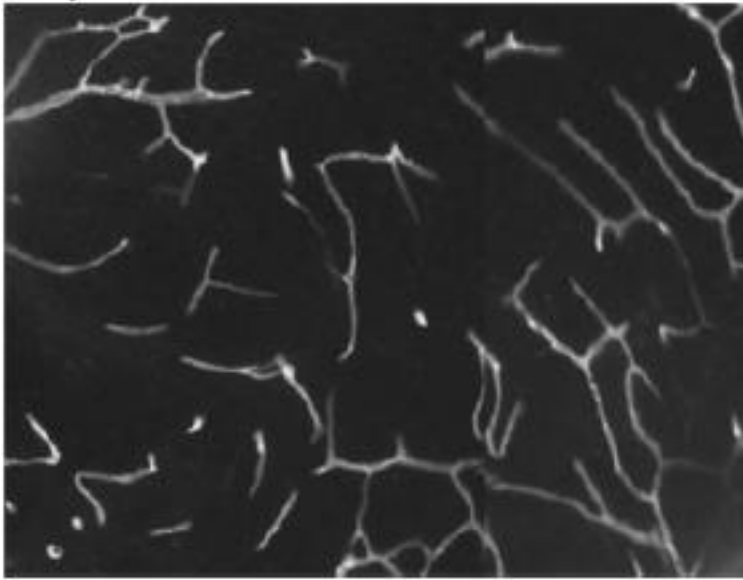
Tại từng pixel, các giá trị:

$$V(\Phi)=[f(x,y)-f(x+c(\Phi),y+s(\Phi))]^2 + \\ [f(x,y)-f(x-c(\Phi),y-s(\Phi))]^2$$

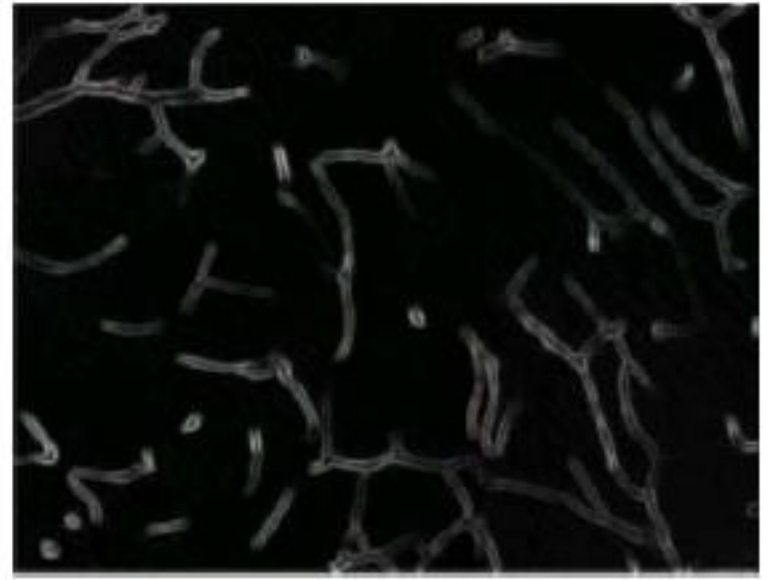
- trong đó:  $f(x,y)$ : mức xám tại pixel trung tâm
- $V(\Phi)$  được tính toán với các giá trị  $\Phi = 0^\circ, 45^\circ, 90^\circ$  và  $135^\circ$ . Với giá trị  $\Phi$  làm  $V(\Phi)$  nhỏ nhất, nhân tích chập tương ứng sẽ được áp dụng.

# 3. Bộ lọc hướng

Ví dụ:



Ảnh gốc (các mạch máu não của chuột)



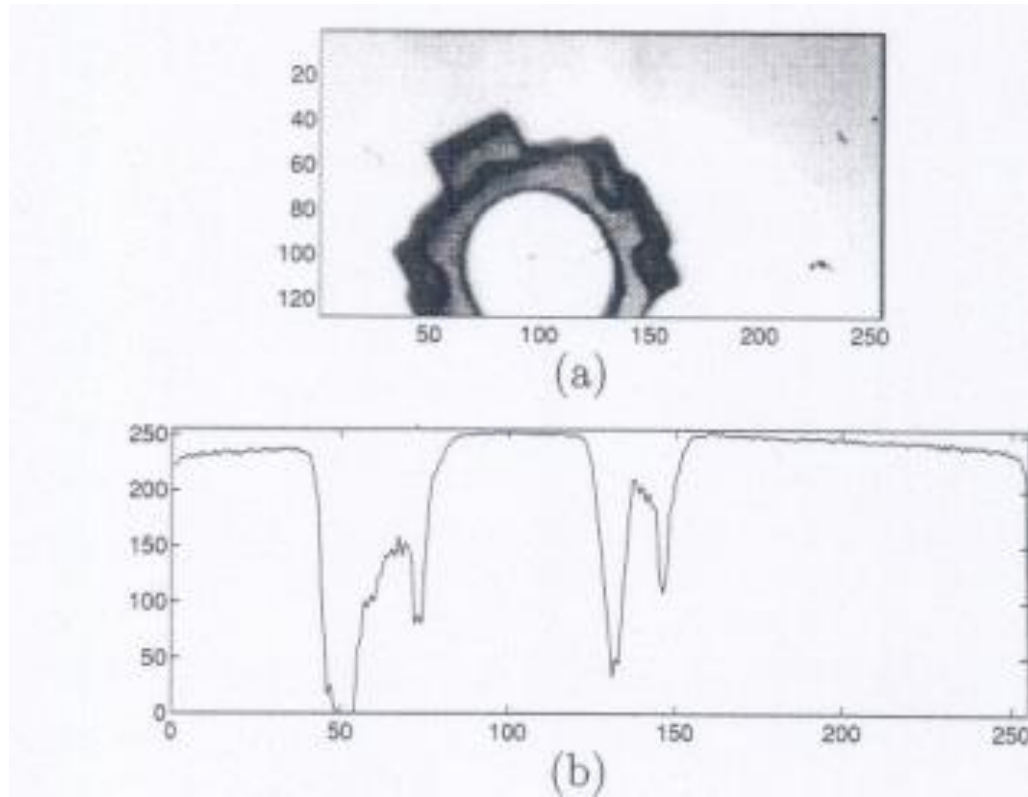
Ảnh sau khi thực hiện lọc hướng

# 4. Bộ lọc dò tìm cạnh ảnh

## Bộ lọc dò tìm cạnh ảnh (Edge Detection)

- ❑ Cạnh (đường viền) ảnh là một đặc trưng rất quan trọng do hệ thống thị giác của con người dựa trên cạnh để nhận thức được đối tượng.
- ❑ Việc dò tìm cạnh ảnh có thể dựa vào tốc độ thay đổi của các giá trị pixel, hay còn gọi là gradient của ảnh.
- ❑ Phương pháp dò cạnh có thể được chia làm 2 loại: phụ thuộc hướng và độc lập hướng.
- ❑ Phương pháp dò cạnh phụ thuộc hướng: sẽ dựa trên việc xác định gradient của ảnh theo các hướng x và y.

## 4. Bộ lọc dò tìm cạnh ảnh



# 4. Bộ lọc dò tìm cạnh ảnh

Gradient:

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix}$$

Độ lớn của gradient:

$$|\nabla f(x, y)| = \left[ \left( \frac{\partial f(x, y)}{\partial x} \right)^2 + \left( \frac{\partial f(x, y)}{\partial y} \right)^2 \right]^{1/2}$$

Thông thường, trong miền rời rạc độ lớn của gradient được tính xấp xỉ

# 4. Bộ lọc dò tìm cạnh ảnh

Xét vùng ảnh như sau:

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

Khi đó, ta có thể xấp xỉ:

$$|\nabla f(x, y)| \approx \left[ (z_5 - z_8)^2 + (z_5 - z_6)^2 \right]^{1/2}$$

Khi cài đặt, ta có thể dùng các nhân tương ứng:

$$h_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \text{ and } h_2 = \begin{bmatrix} 1 & -1 \end{bmatrix}$$

Hay ta có thể viết:  $|\nabla f(x, y)| \approx \left[ (f * h_1)^2 + (f * h_2)^2 \right]^{1/2}$

## 4. Bộ lọc dò tìm cạnh ảnh

Ngoài ra, ta có thể xấp xỉ:

$$|\nabla f(x, y)| \approx \left[ (z_5 - z_9)^2 + (z_6 - z_8)^2 \right]^{1/2}$$

Khi cài đặt, ta có thể dùng các nhân tương ứng:

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Hay ta có thể viết:

$$|\nabla f(x, y)| \approx \left[ (f * h_1)^2 + (f * h_2)^2 \right]^{1/2}$$



# 4. Bộ lọc dò tìm cạnh ảnh

Các nhân tích chập trên còn được gọi là toán tử **Roberts cross-gradient operators**



Ảnh gốc



$$h_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \text{ and } h_2 = \begin{bmatrix} 1 & -1 \end{bmatrix}$$



$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

# 4. Bộ lọc dò tìm cạnh ảnh

## Prewitt operation

Một xấp xỉ tốt hơn giá trị của gradient có thể được tính như sau:

$$|\nabla f(x, y)| \approx \left[ ((z_7 + z_8 + z_9) - (z_1 + z_2 + z_3))^2 + ((z_3 + z_6 + z_9) - (z_1 + z_4 + z_7))^2 \right]^{1/2}$$

Tương ứng với nhân tích chập sau:

$$h_1 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ and } h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

# 4. Bộ lọc dò tìm cạnh ảnh

## Prewitt operation



Ảnh gốc



Prewitt

# 4. Bộ lọc dò tìm cạnh ảnh (8)

## Phương pháp dò cạnh Sobel:

- Là phương pháp dò cạnh phụ thuộc hướng và đối xứng trục. Ảnh đích được xác định theo công thức sau:

$$D(x, y) = \sqrt{SH(x, y)^2 + SV(x, y)^2}$$

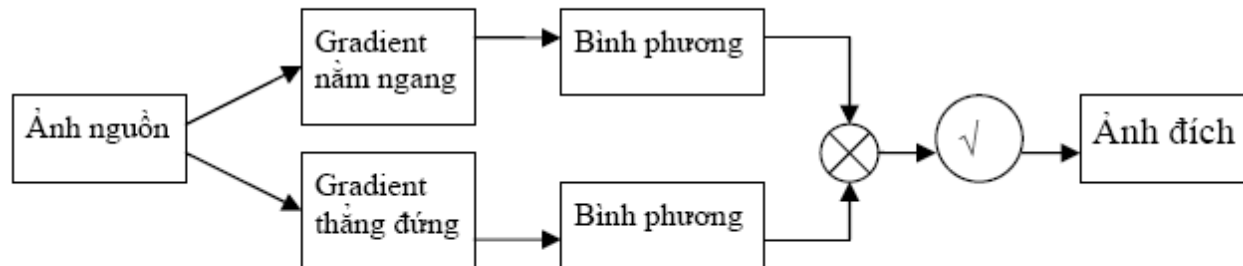
- Trong đó  $SH(x, y)$  và  $SV(x, y)$  là các ảnh gradient nằm ngang và thẳng đứng phát sinh từ việc tích chập ảnh nguồn với hai nhân sau:

H=

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

V=

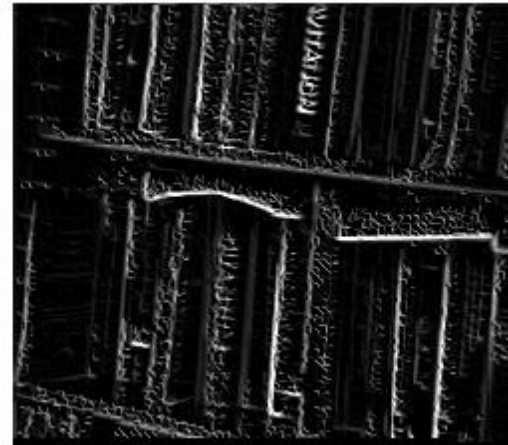
-1	0	1
-2	0	2
-1	0	1



Mô hình của phương pháp dò cạnh Sobel

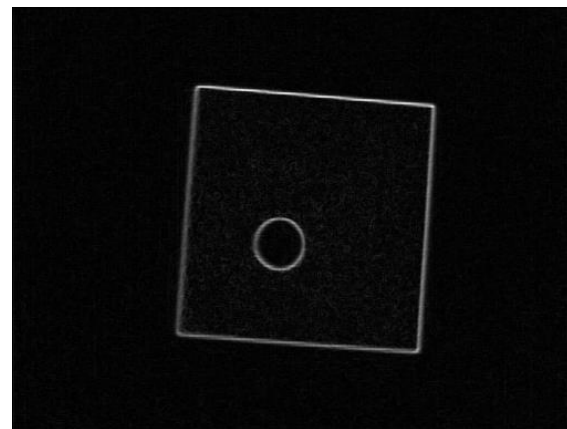
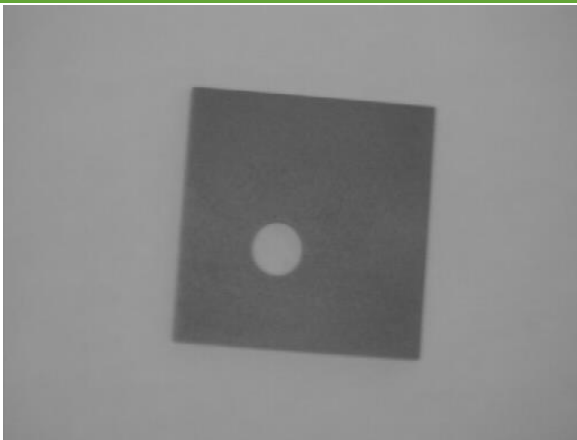
## 4. Bộ lọc dò tìm cạnh ảnh (9)

Ví dụ:



→ Hạn chế của phương pháp này: nhạy cảm đối với nhiễu (nếu nhiễu lớn thì nó sẽ lấy nhiễu làm cạnh).

## 4. Bộ lọc dò tìm cạnh ảnh (11)



Ảnh gốc

Sobel

# 4. Bộ lọc dò tìm cạnh ảnh



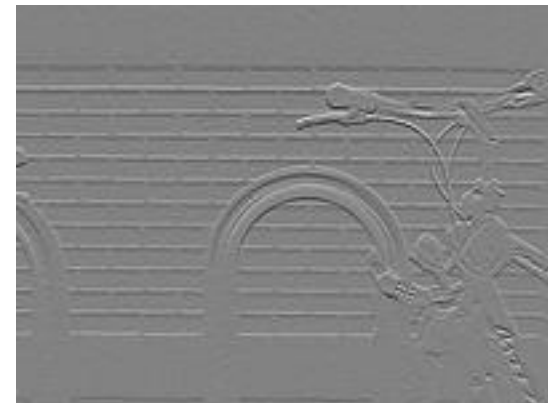
Grayscale image of a brick wall & a bike rack



Normalized sobel gradient image of bricks & bike rack



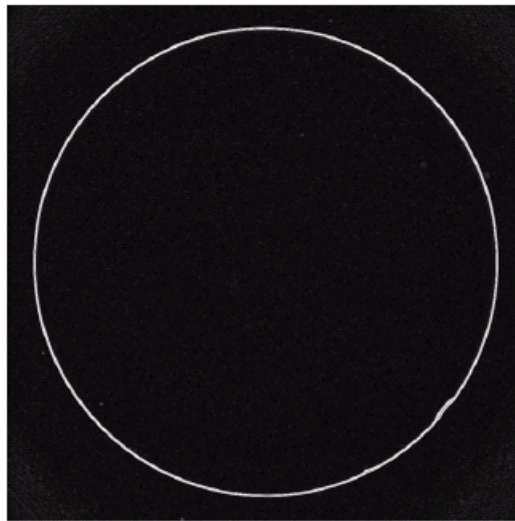
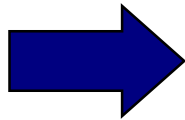
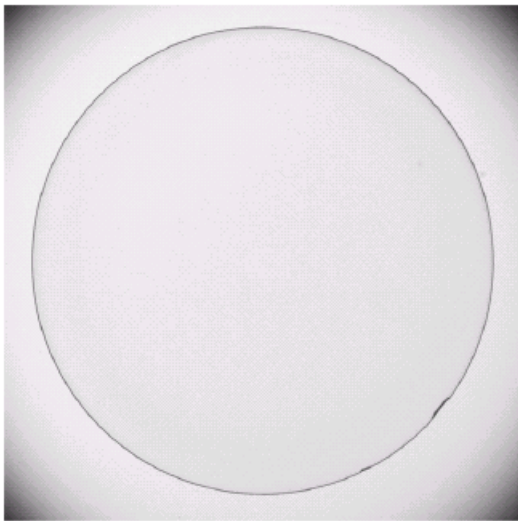
Normalized sobel x-gradient image



Normalized sobel y-gradient image

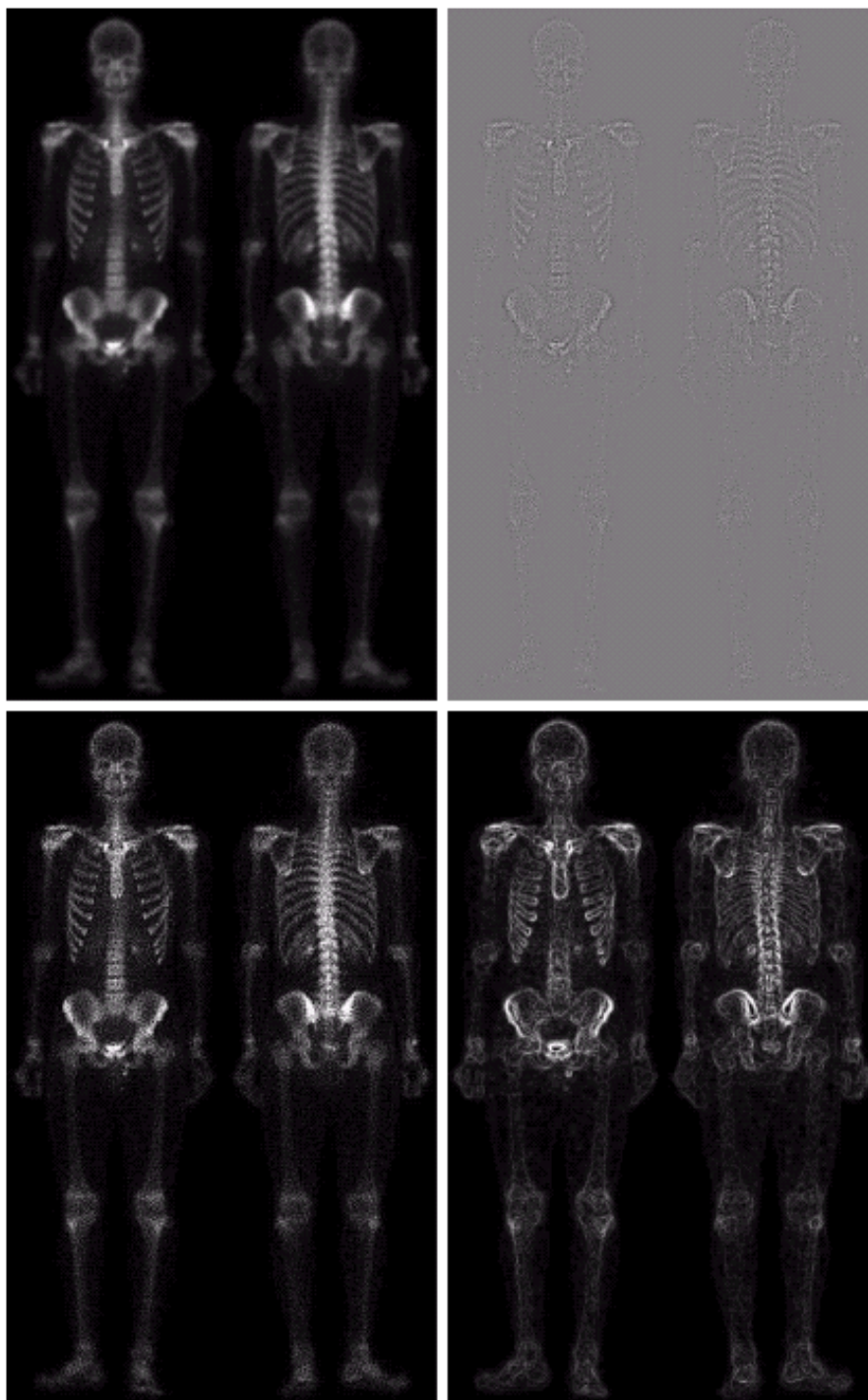


## 4. Bộ lọc dò tìm cạnh ảnh



**An image of a contact lens which is enhanced in order to make defects (at four and five o'clock in the image) more obvious**





a	b
c	d

**FIGURE 3.46**  
 (a) Image of whole body bone scan.  
 (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).

# Combining Spatial Enhancement Methods

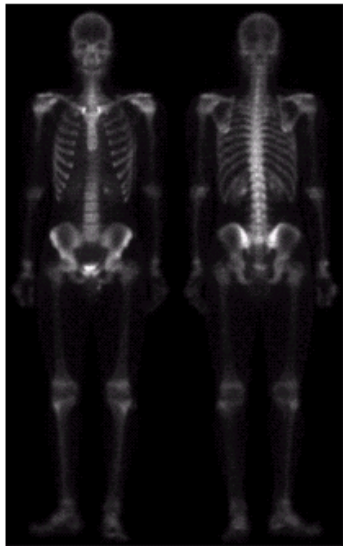
Successful image enhancement is typically not achieved using a single operation

Rather we combine a range of techniques in order to achieve a final result

This example will focus on enhancing the bone scan to the right



# Combining Spatial Enhancement Methods (cont...)



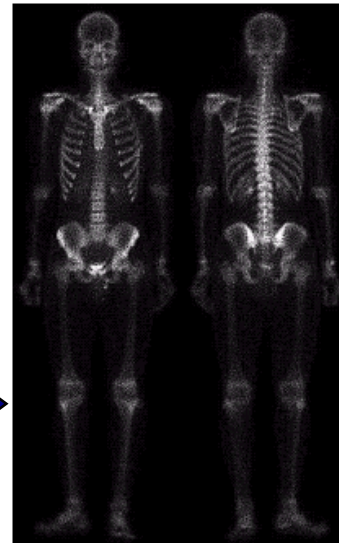
(a)

Laplacian filter of  
bone scan (a)



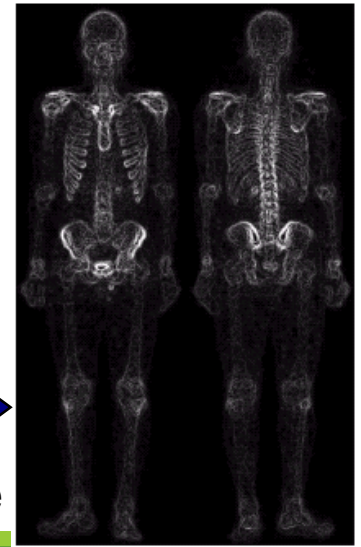
(b)

Sharpened version of  
bone scan achieved  
by subtracting (a)  
and (b)



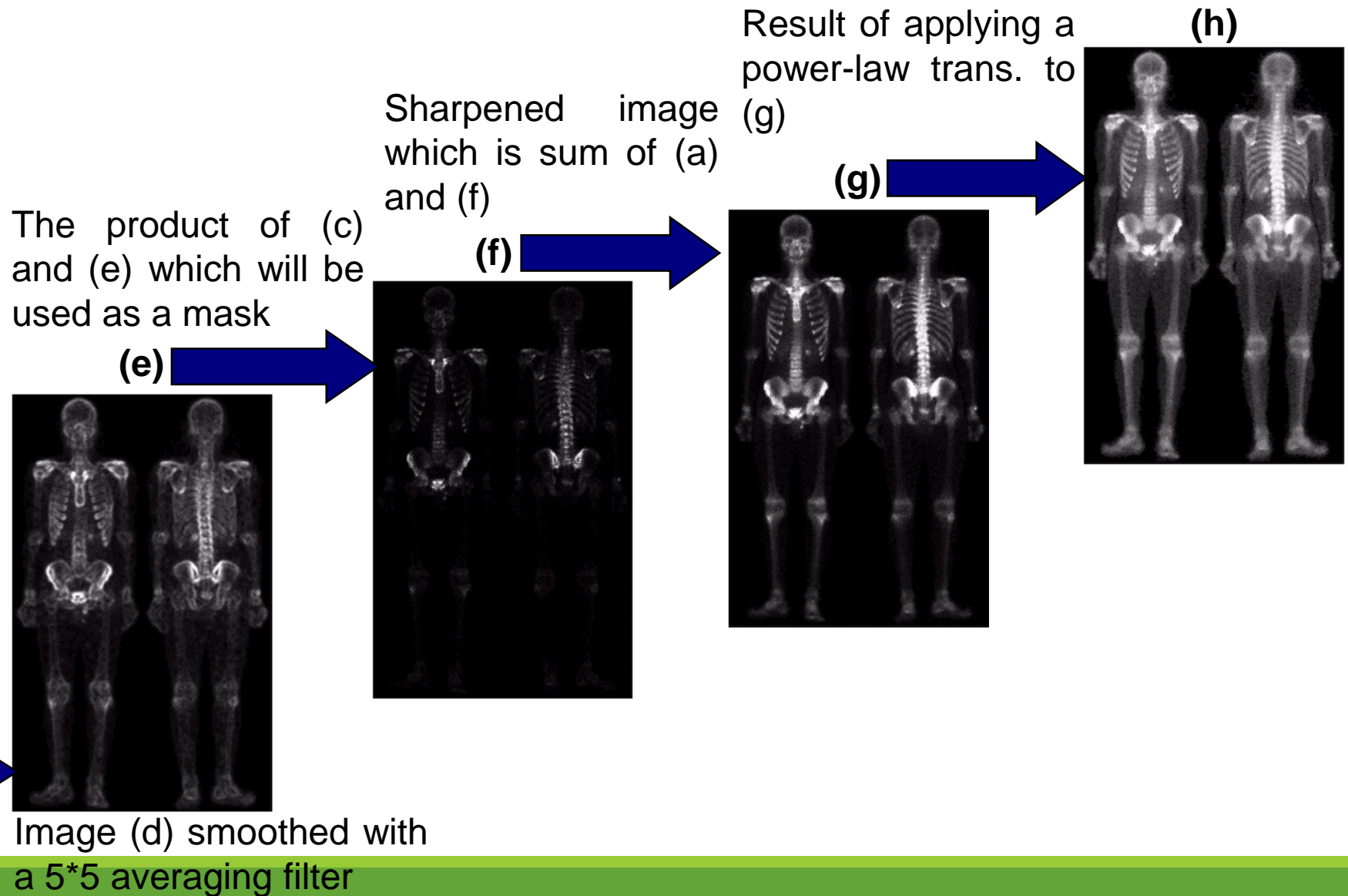
(c)

Sobel filter of bone  
scan (a)



(d)

# Combining Spatial Enhancement Methods (cont...)



# Combining Spatial Enhancement Methods (cont...)

Compare the original and final images



# Tài liệu tham khảo