

Tăng cường ảnh: lọc không gian

- Tăng cường ảnh trong miền không gian được mô tả như:

Ảnh tăng
cường

$$g(m,n) = T(f(m,n))$$

Biến đổi

Ảnh ban đầu

- Biến đổi T có thể tuyến tính hay phi tuyến.

Xác định T

- Nếu T tuyển tính và thay đổi bất biến (linear and shift invariant - LSI), và có đặc tính PSS (point-spread sequence) $h(m,n)$, thì

$$g(m,n) = h(m,n)^* f(m,n)$$

$$= \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h(m-k, n-l) f(k, l)$$

$$= \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f(m-k, n-l) h(k, l)$$

Xác định T

- Trong thực tế, để giảm tính toán thì

$$h(k,l) = 0 \text{ for } (k,l) \notin \Delta$$

với Δ là một tập nhỏ (tập láng giềng). Δ còn được gọi là xác định của h .

- Trong miền tần số thì được mô tả như:

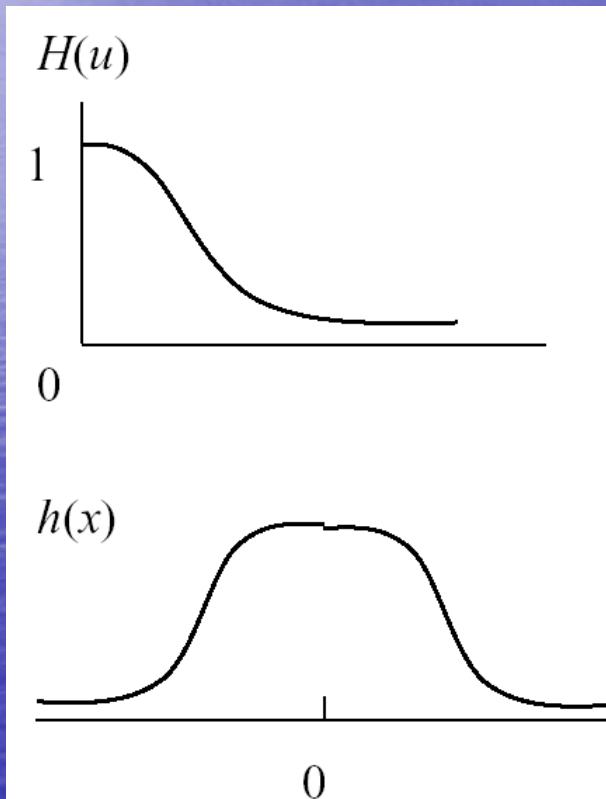
$$G(u,v) = H_e(u,v) F_e(u,v)$$

- Có nhiều phép toán LSI dùng trong miền tần số như "phép lọc".
- Lọc theo một giới hạn cũng được xem như phép toán.

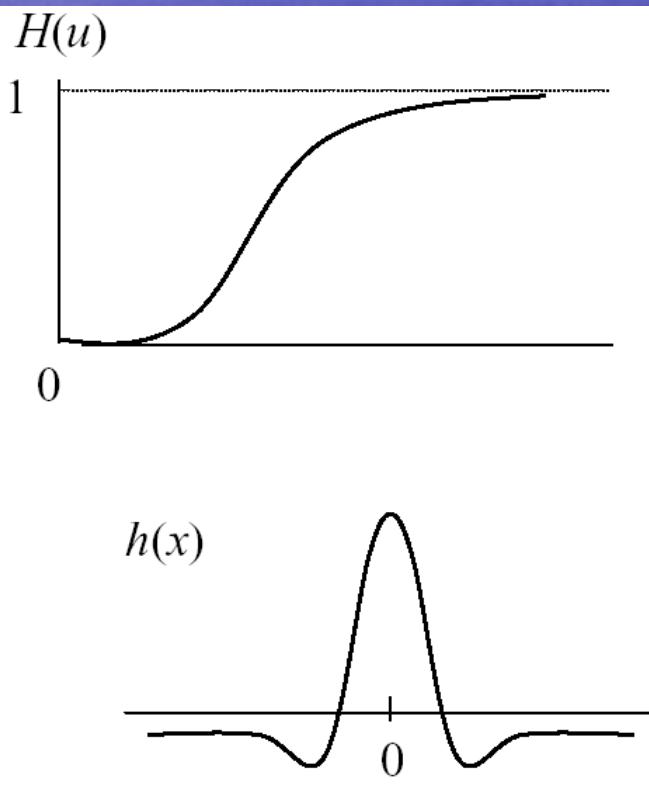
Xác định T

- Ví dụ (trường hợp 1-D):

Lọc băng thông thấp



Lọc băng thông cao



- Nếu $h(m, n)$ là một mảng
nặng 3x3

$h =$

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

thì

$$\begin{aligned}
 g(m, n) = & w_1 f(m-1, n-1) + w_2 f(m-1, n) + w_3 f(m-1, n+1) \\
 & + w_4 f(m, n-1) + w_5 f(m, n) + w_6 f(m, n+1) \\
 & + w_7 f(m+1, n-1) + w_8 f(m+1, n) + w_9 f(m+1, n+1)
 \end{aligned}$$

- Giá trị $g(\tilde{m}, \tilde{n})$ được tính bằng cách trượt mặt nạ qua mỗi pixel của ảnh $f(m, n)$.
- Cẩn thận đặc biệt khi các pixel nằm trên biên của ảnh $f(m, n)$. Để giải quyết, chúng ta có thể chọn:
 - Mặt nạ được cắt ngắn tại biên (Biên tự do)
 - Mở rộng thêm dòng/cột tại biên (Hiệu chỉnh biên).
 - Biên được “bao bọc xung quanh” (Chu kỳ biên).
- Trong MATLAB dùng lệnh `filter2` dựa trên lệnh `conv2`.

Các lọc làm trơn ảnh

- Làm trơn ảnh thuộc phép biến đổi ảnh-ảnh, phép biến đổi này làm khác biệt giữa các pixel không nhiều.
- Phép lọc trơn này dùng cho:
 - **Làm mờ ảnh (Blurring)**: Đây là bước tiền xử lý nhằm loại bỏ bớt các chi tiết nhỏ (không cần thiết) trước khi trích đổi tượng cản thiết (lớn), hay làm tăng chi tiết ảnh.
 - **Giảm nhiễu (Noise reduction)**: Giảm bớt tác động của nhiễu.

Làm trơn ảnh bằng mặt nạ trung bình (lọc không gian băng thấp)

- Có thể áp dụng mặt nạ trung bình (**averaging mask**) để làm trơn.
- Một mặt nạ trung bình là mặt nạ có tổng các trọng băng 1. Được tính dựa trên các láng giềng, nên mặt nạ còn có tên trung bình láng giềng (**neighborhood averaging**).
- Vài loại 3x3:

$$\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{32} \begin{bmatrix} 1 & 3 & 1 \\ 3 & 16 & 3 \\ 1 & 3 & 1 \end{bmatrix} \quad \frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Phép toán này tương đương lọc băng thông thấp.

Ví dụ làm mờ ảnh



Ảnh ban đầu

$$\frac{1}{N^2} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{N \times N}$$

Mặt nạ trung bình

Ví dụ làm mờ

$N = 3$



$N = 7$



$N = 11$



$N = 21$



Ví dụ giảm nhiễu



Ảnh ban đầu

$$\frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Ví dụ giảm nhiễu



Nhiễu Zero-mean Gauss, Variance = 0.01

Ví dụ giảm nhiễu



Nhiễu Zero-mean Gauss, Variance = 0.05

Lọc Median

- Nếu nhiễu thuộc dạng phân bố Gauss thì lọc trung bình rất hiệu quả.

$$p_{\text{noise}}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-x^2}{2\sigma^2}\right)$$

- Lọc trung bình là loại lọc làm mờ cạnh và chi tiết.
- Lọc median làm cạnh có chất lượng tốt và được dùng khi nhiễu thấy tách biệt. **Ví dụ:** nhiễu muối tiêu.
- Lọc median là một **lọc phi tuyến** dùng mặt nạ. Mỗi pixel được thay thế bằng median của các pixel lảng giêng.

Lọc median

- Giả sử $A = \{a_1, a_2, \dots, a_k\}$ là các giá trị pixel láng giềng với $a_1 \leq a_2 \leq \dots \leq a_k$.Ì

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

- Chú ý: median của một tập có thứ tự là “giá trị tâm”.
- Ví dụ:
 - nếu $A = \{0, 1, 2, 4, 6, 6, 10, 12, 15\}$ thì $\text{median}(A) = 6$.

Ví dụ giảm nhiễu

Nhiễu Gauss $s = 0.2$



Nhiễu muối tiêu prob. = 0.2



Ví dụ giảm nhiễu

Dùng lọc trung bình 3×3



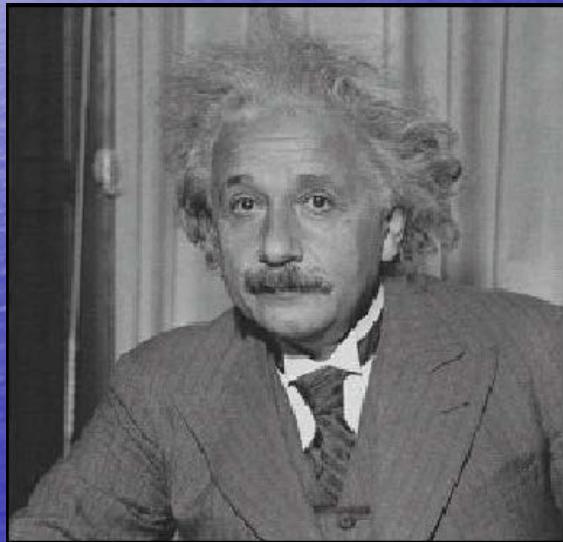
Ví dụ giảm nhiễu

Dùng lọc median 3x3

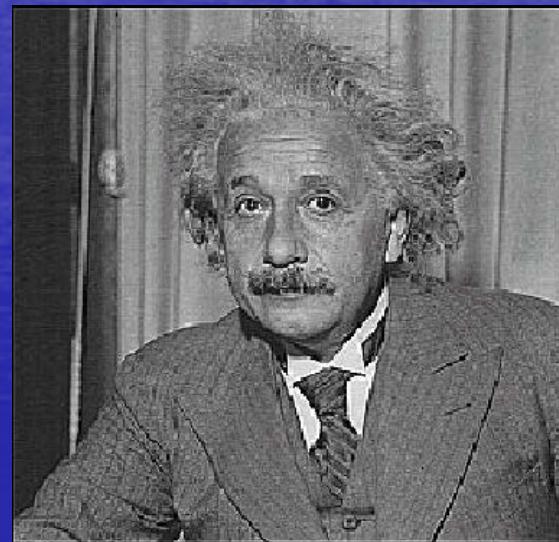


Làm ảnh sắc nét

- Tăng cường độ sắc nét bằng phương pháp làm mờ.



Trước



Sau khi làm sắc nét

Lọc không gian băng thông cao cơ bản

- Dùng phép toán LSI, được cài đặt bởi mặt nạ trung bình, gồm các giá trị dương và âm.
- Mặt nạ có tên là mặt nạ tạo sắc nét, quan tâm những nơi mức xám thay đổi đột ngột trong ảnh.
- Mặt nạ nên có giá trị dương ở tâm và giá trị âm ở xung quanh, các giá trị này có tổng bằng zero.

Lọc không gian bằng thông cao cơ bản

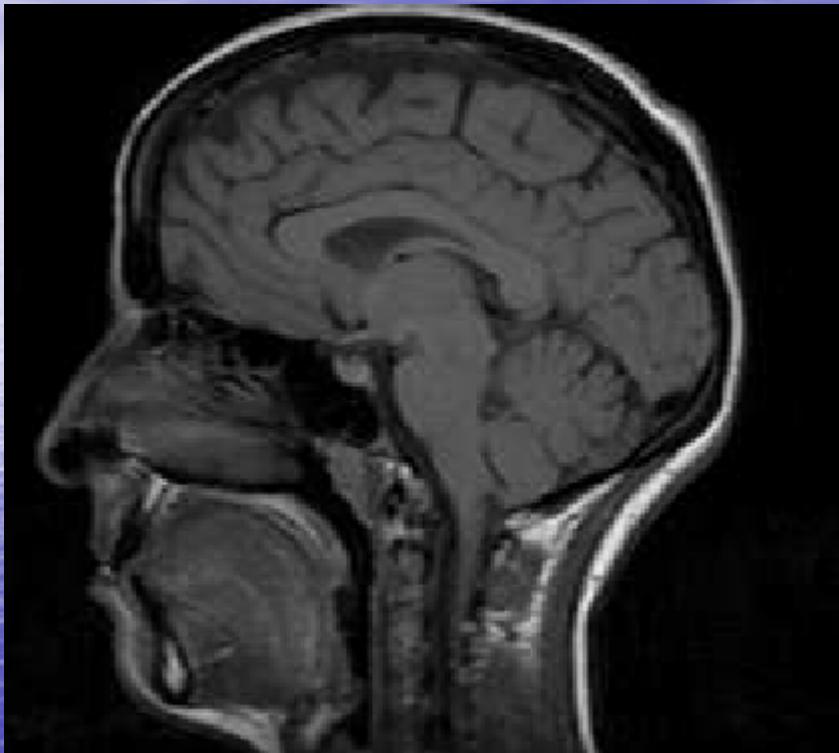
- Ví dụ:

$$\frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

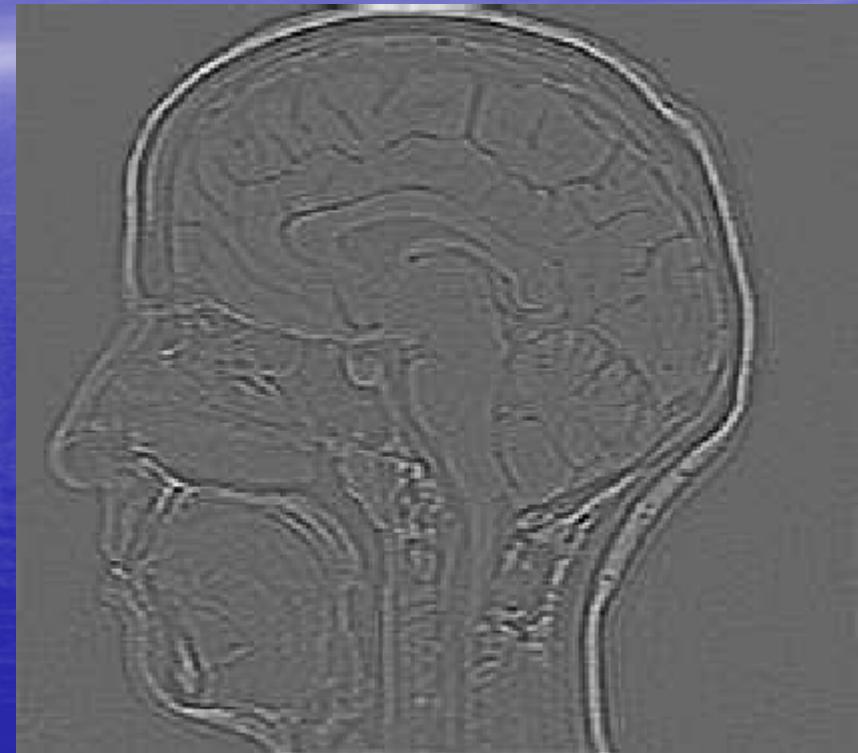
- Mặt nạ tương đương **lọc băng thông cao**.
- Sự khác biệt của f và g sau khi lọc băng thông cao có thể xem như:

$$g(m,n) = f(m,n) - lowpass(f(m,n))$$

Ví dụ



Ảnh ban đầu



Lọc băng thông cao

Lọc High-boost

- Đây là một lọc dùng phép trừ băng thông thấp từ mở rộng của f

$$g(m,n) = Af(m,n) - \text{lowpass}(f(m,n))$$

Đây còn gọi là mặt nạ không sắc nét (unsharp masking).

Lọc High-boost

- Quan sát thấy

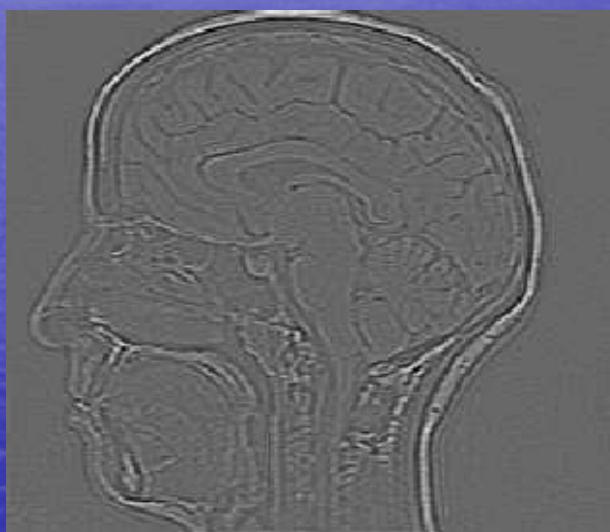
$$\begin{aligned}g(m,n) &= Af(m,n) - \text{lowpass}(f(m,n)) \\&= (A-1)f(m,n) + f(m,n) - \text{lowpass}(f(m,n)) \\&= (A-1)f(m,n) + \text{hipass}(f(m,n))\end{aligned}$$

- Kết quả là giống ảnh ban đầu nhưng các cạnh sẽ nổi bật hơn.

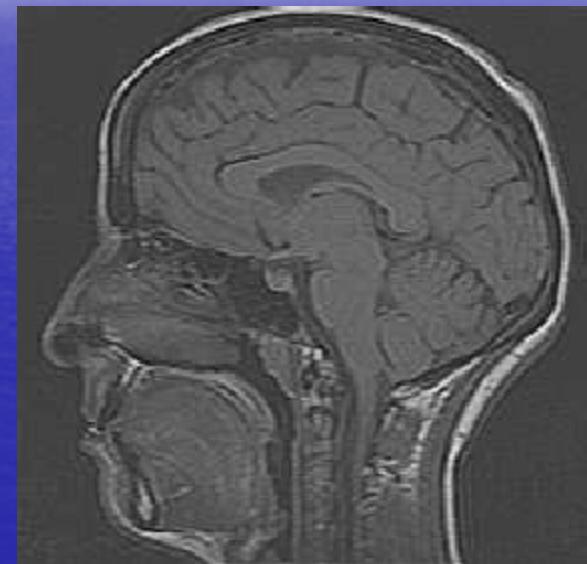
Ví dụ



Ảnh ban đầu



Lọc băng thông cao



Lọc High-boost

Lọc theo đạo hàm

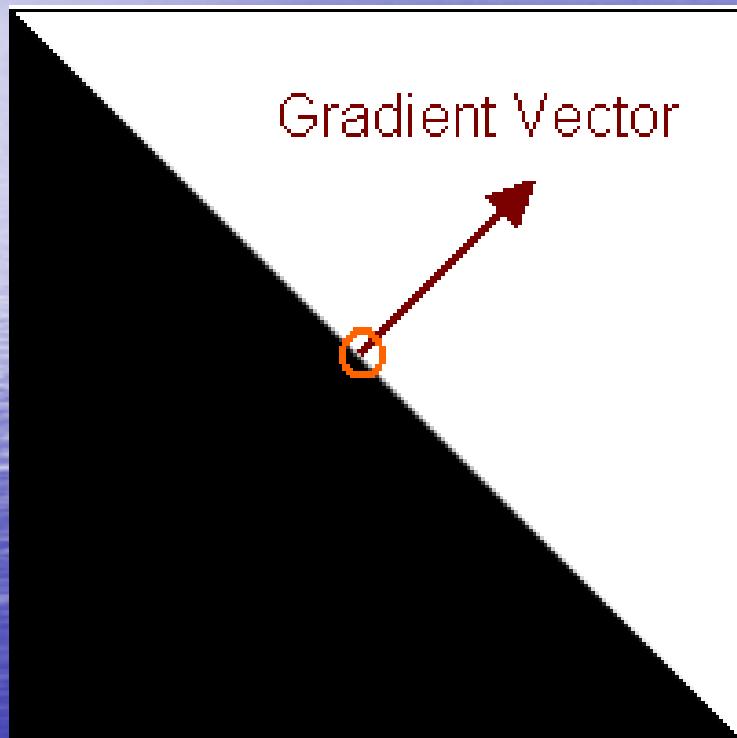
- Trung bình quan tâm làm mờ các chi tiết trong ảnh bằng cách xem xét quan hệ trung bình hay tích hợp xung quanh.
- Một cách tự nhiên, đạo hàm cho ta cái nhìn về sự thay đổi độ dốc.
- Phép toán đạo hàm thông dụng là gradient:

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

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

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

Cạnh và gradient



1. Độ đậm của cạnh bằng độ lớn của gradient.
2. Hướng của cạnh bằng góc của gradient.

Lọc theo đạo hàm

- Độ lớn của gradient:

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

- Thường xấp xỉ rời rạc hóa độ lớn của gradient.

- Xem xét vùng ảnh:

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Chúng ta có thể xấp xỉ

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

Lọc theo đạo hàm

- Có thể cài đặt theo mặt nạ: $h_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ và $h_2 = [1 \quad -1]$
- Theo đó:

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

- Một lựa chọn khác, chúng ta có thể xấp xỉ:

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

- Sẽ cài đặt theo $h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ và $h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

- Thì

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

Lọc theo đạo hàm

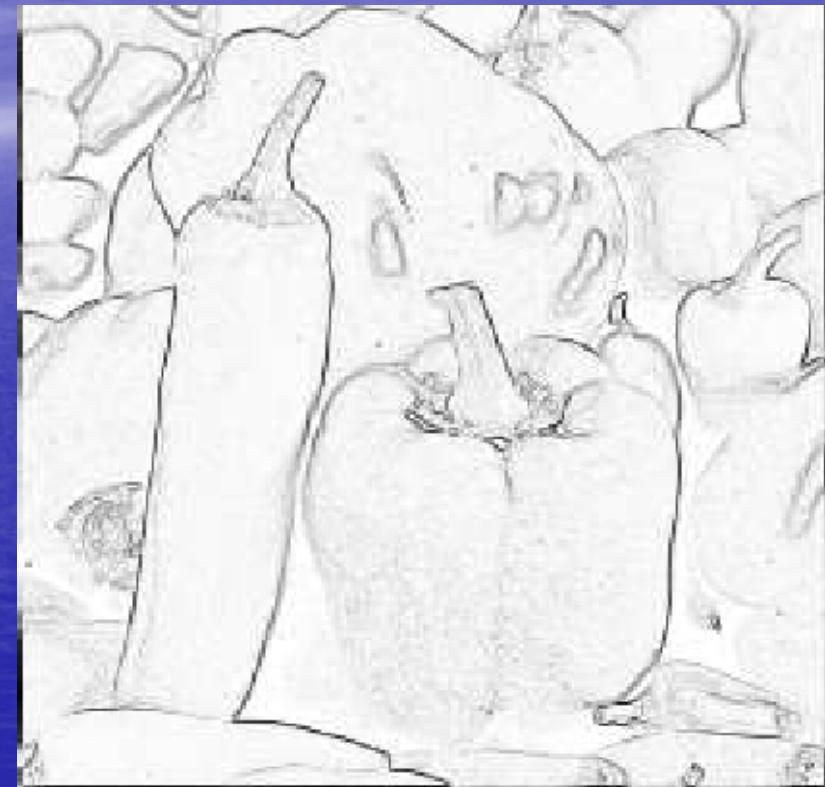
- Mặt nạ sau cùng được gọi là phép toán gradient chéo Robert (**Roberts cross-gradient operator**).
- Phép toán Robert và phép toán Prewitt/Sobel (trình bày sau) được dùng để xác định cạnh, nên đôi khi còn được gọi xác định cạnh.

Ví dụ: phép toán gradient-chéo Robert



Ảnh ban đầu

Ví dụ



$$h_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \text{ và } h_2 = [1 \quad -1]$$

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

Phép toán Prewitt

- Xấp xỉ gradient tốt hơn:

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

- Có thể được cài đặt như mặt nạ:

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

- Với công thức:

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

Phép toán Sobel

- Một xấp xỉ khác:

$$|\nabla f| = [[(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)]^2 + [(z_2 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)]^2]^{1/2}$$

$$h_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ và } h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- Đây được gọi là **phép toán Sobel**.

Ví dụ



Prewitt



Sobel

LINEAR SPATIAL FILTERING

- ```
>> f=imread('fish.jpg'); %load in checkerboard figure
```
- ```
% g=imfilter(f,w,filtering_mode, boundary_options, size_options)
```
- ```
% f is the input image , w is the filter mask
```
- ```
% Filtering mode:
```

 - % 'corr' filtering is done using correlation
 - % 'conv' filtering is done using convolution -- flips mask 180 degrees
- ```
% Boundary options:
```

  - % P without quotes (default) - pad image with zeros
  - % 'replicate' - extend image by replicating border pixels
  - % 'symmetric' - extend image by mirroring it across its border
  - % 'circular' - extend image by repeating it (one period of a periodic function)
- ```
% Size options:
```

 - % 'full' - output is the same size as the padded image
 - % 'same' - output is the same size as the input
- ```
>> w=ones(9); % create a 9x9 filter (not normalized)
```
- ```
>> gd=imfilter(f,w); % filter using default values
```
- ```
>> imshow(gd, []) % [] causes MATLAB to display using low and high
```
- ```
- % gray levels of input image.
```
- ```
- % Good for low dynamic range
```
- ```
>> gr=imfilter(f,w,'replicate');
```

 - % pad using replication
- ```
>> figure, imshow(gr, []) %
```
- ```
>> gs=imfilter(f,w,'symmetric');
```

 - % pad using symmetry
- ```
>> figure, imshow(gs, []) % show this figure in a new window
```

- LINEAR SPATIAL FILTERING
- >> f=imread('fig3.15(a).jpg'); %load in checkerboard figure
- >> w=ones(9); % create a 9x9 filter (not normalized)
  - % f is of type double in [0,1] by default
- >> f8=im2uint8(f); % converts image to uint8, i.e., integers in range [0,255]
- >> g8r=imfilter(f8,w,'replicate'); % pad using replication
  - % imfilter creates an output of same data class as input, i.e., uint(8)
- >> imshow(g8r, [ ]) % clipping caused data loss since filter was not normalized

- ```
>> f=imread('fish.jpg'); %load in checkerboard figure
```
- ```
>> w=fspecial('type', parameters); % create filter mask
```
- % filter types:
  - % 'average', default is 3x3
  - % 'gaussian', default is 3x3 and sigma=0.5
  - % 'laplacian', default alpha=0.5
  - % 'prewitt', vertical gradient, default is 3x3. Get horizontal by wh=w'
  - % 'sobel', vertical gradient, default is 3x3
  - % 'unsharp', default is 3x3 with alpha=0.2
- ```
>> f=imread('Moon.jpg'); %load in lunar north pole image
```
- ```
>> w4=fspecial('laplacian',0) % creates 3x3 laplacian, alpha=0 [0:1]
```
- ```
>> w8=[1 1 1;1 -8 1;1 1 1] % create a Laplacian that fspecial can't
```
- ```
>> f=im2double(f); % output same as input unit8 so
```
- ```
 % negative values are truncated.
```
- ```
 % Convert to double to keep negative values.
```
- ```
>> g4=f-imfilter(f,w4,'replicate'); % filter using default values
```
- ```
>> g8=f-imfilter(f,w8,'replicate'); % filter using default values
```
- ```
>> imshow(f) % display original image
```
- ```
>> imshow(g4) % display g4 processed image
```
- ```
>> imshow(g8) % display g8 processed image
```