

# Xử lý ảnh và Thị giác máy tính

# Chương 3. Khôi phục và tái tạo ảnh

- Mô hình nhiễu
- Khôi phục ảnh khi có nhiễu
- Các phương pháp lọc

# 1. Nhiễu trong ảnh

- Do các hạn chế vật lý vốn có của các thiết bị ghi khác nhau, hình ảnh có xu hướng bị nhiễu ngẫu nhiên trong quá trình thu nhận hình ảnh
- Nhiễu có thể hiểu là hiện tượng méo tín hiệu

# 1. Nhiễu trong ảnh

- Ảnh thường bị biến dạng do nhiễu ngẫu nhiên. Nhiễu xuất hiện trong quá trình thu nhận ảnh hoặc truyền tin
- Các yếu tố môi trường, ví dụ điều kiện ánh sáng yếu, nhiệt độ của thiết bị cảm biến cũng ảnh hưởng đến sự xuất hiện của nhiễu
- Nhiễu có thể là nhiễu trắng, nhiễu Gauss, nhiễu xung hoặc nhiễu muối tiêu

# Ví dụ



## 2. Mô hình nhiễu

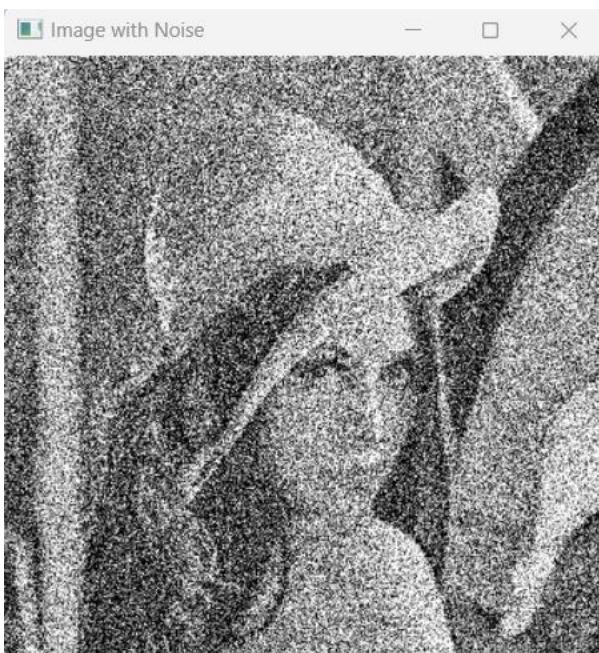
$$f(x, y) = g(x, y) + n(x, y)$$

Trong đó

- $f(x, y)$  là ảnh nhiễu
- $g(x, y)$  là ảnh gốc
- $n(x, y)$  là ma trận nhiễu

## 2. Mô hình nhiễu

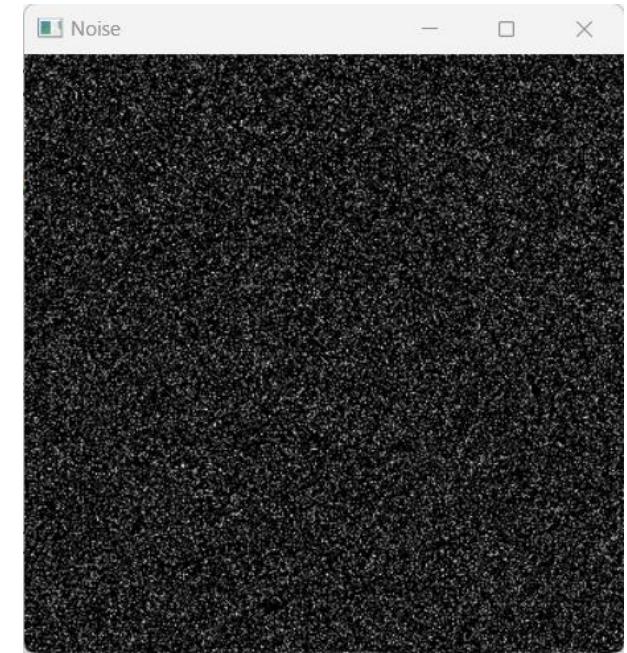
$$f(x, y) = g(x, y) + n(x, y)$$



=



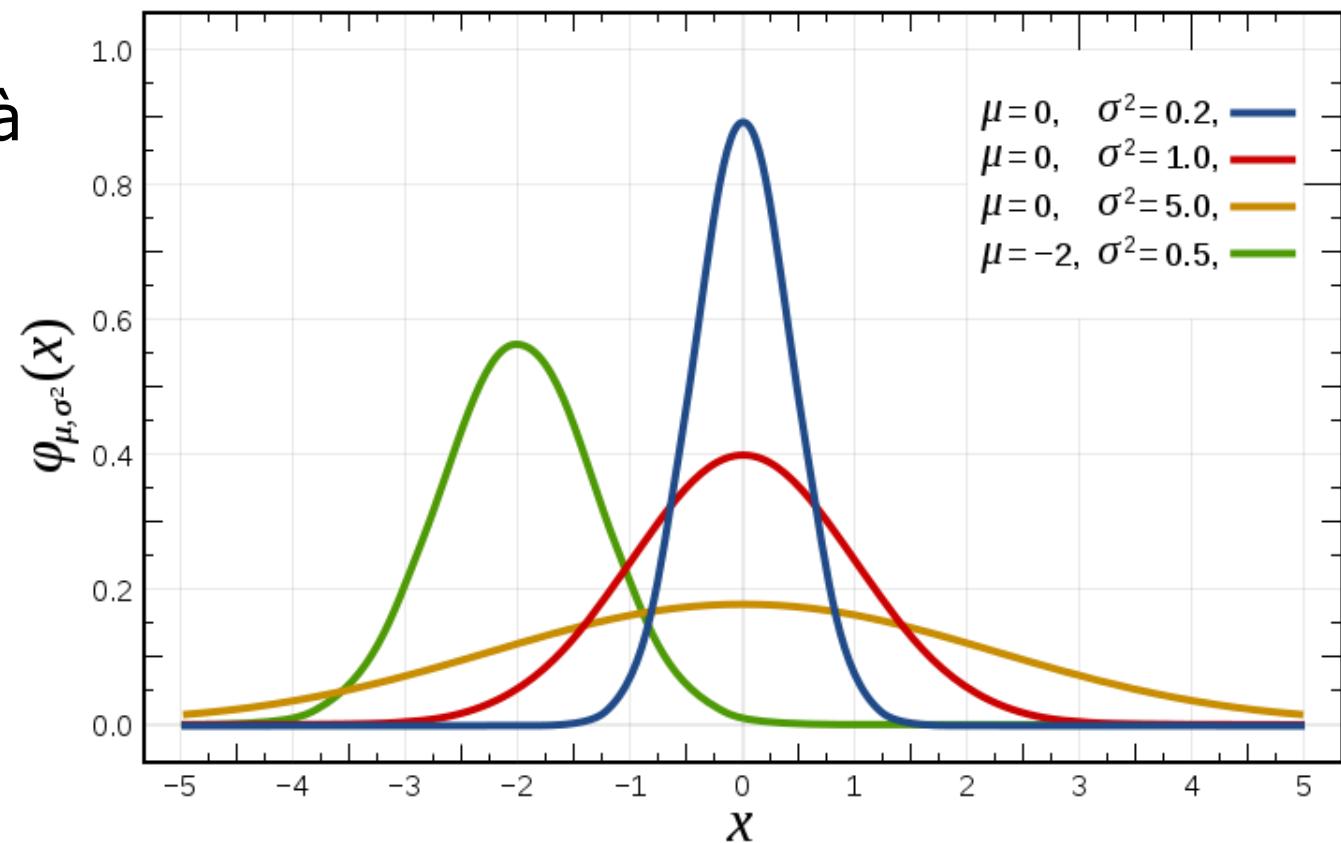
+



## 2.1 Nhiễu Gauss

- Mật độ phân bố xác suất của nhiễu là hàm Gauss, được đặc trưng bởi giá trị trung bình  $\mu$  và phương sai  $\sigma^2$

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

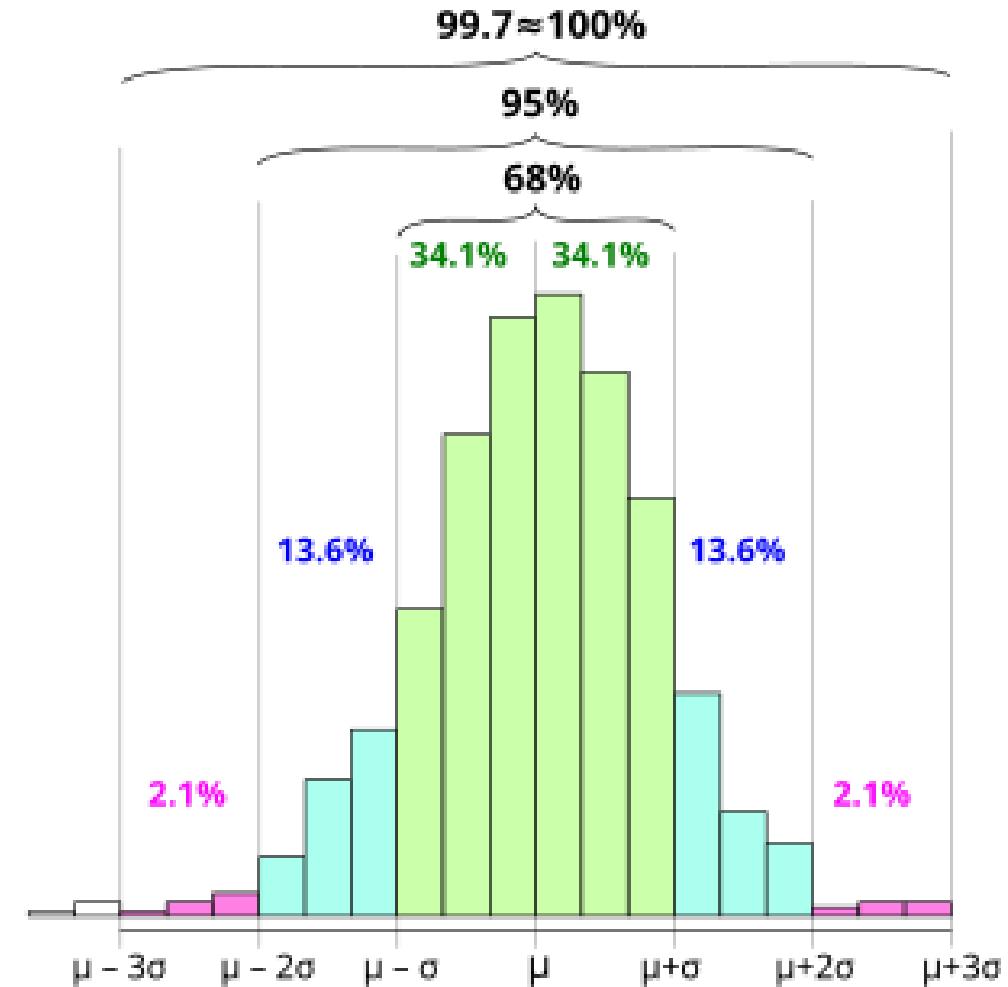


# Phân phối chuẩn Gaussian

$\Pr(\mu - 1\sigma \leq X \leq \mu + 1\sigma) \approx 68.27\%$

$\Pr(\mu - 2\sigma \leq X \leq \mu + 2\sigma) \approx 95.45\%$

$\Pr(\mu - 3\sigma \leq X \leq \mu + 3\sigma) \approx 99.73\%$



## 2. Mô hình nhiễu Gauss

$$f(x, y) = g(x, y) + n(x, y)$$

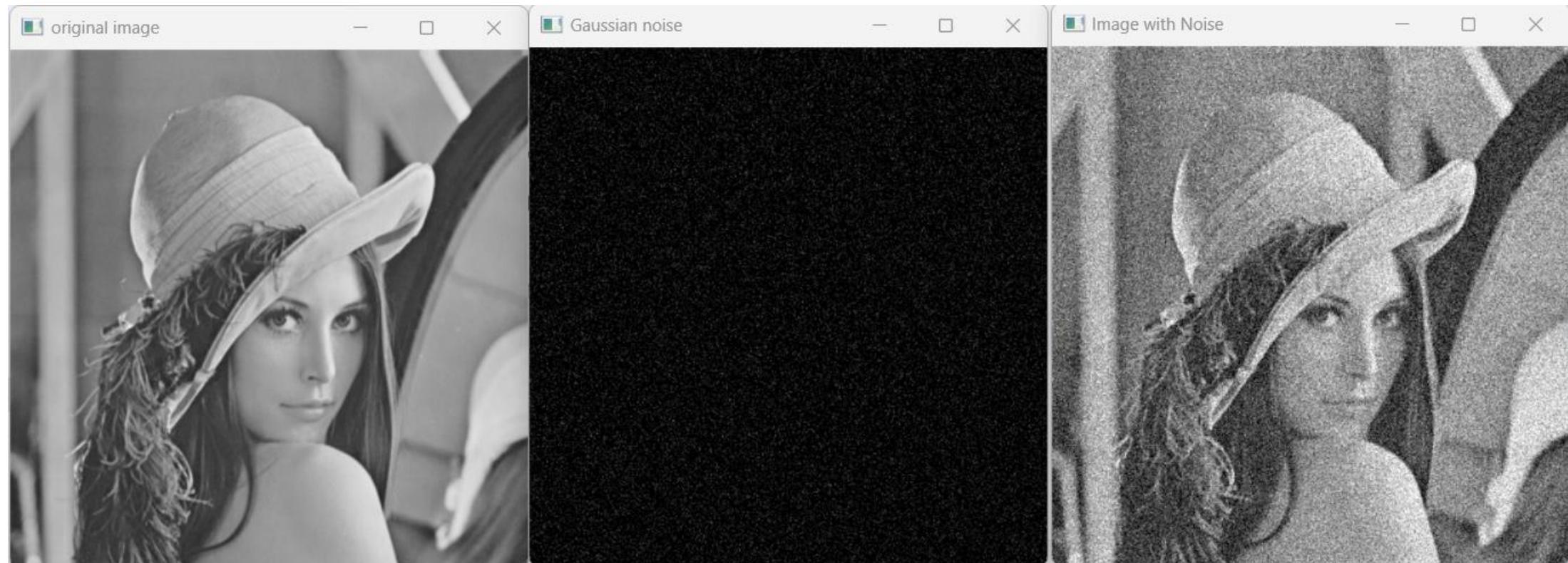
Trong đó

- $f(x, y)$  là ảnh nhiễu
- $g(x, y)$  là ảnh gốc
- $n(x, y)$  là ma trận nhiễu gauss (theo phân phối chuẩn)

# Ví dụ

```
n = np.random.normal(loc=mean, scale=sigma, size=(x,y))
```

```
f = g + n
```





## 2.2 Nhiễu trắng

- Nhiễu trắng là nhiễu có phổ năng lượng không đổi
- Nếu nhiễu có phổ năng lượng nhiều hơn ở một vài tần số, nhiễu được gọi là nhiễu màu

# Ví dụ

Original Image



White noise



## 2.3 Nhiễu Poisson

- Trong quá trình thu nhận, nếu số lượng lớn hạt photon tập trung vào một điểm, chúng sẽ tạo ra nhiễu tại điểm đó
- Nhiễu được đặc trưng bởi hàm mật độ phân bố xác suất Poisson, nên được gọi là nhiễu Poisson

# Ví dụ

Original Image



Poisson noise





## 2.4 Nhiễu muối tiêu (salt & pepper)

- Nhiễu xung đặc trưng bởi một điểm ảnh có giá trị mức xám khác biệt lớn so với những điểm lân cận
- Xung của nhiễu có thể là âm (đen) hoặc dương (trắng)

# Ví dụ nhiễu muỗi tiêú



# Ví dụ chỉ có nhiễu muỗi



# Ví dụ chỉ có nhiễu tiêu



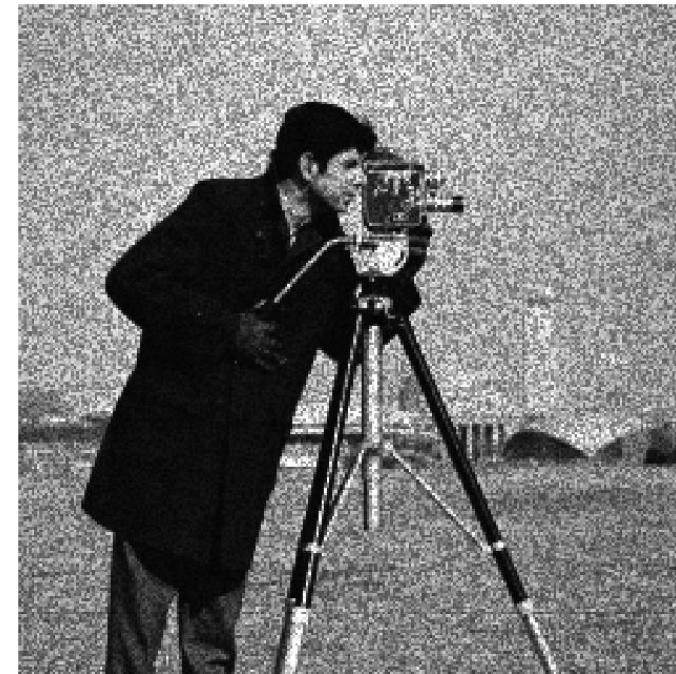
## 2.5 Nhiễu đốm (Speckle Noise)

- Speckle noise là nhiễu xuất hiện trong quá trình thu nhận hình ảnh, dưới dạng các “đốm”

Original Image

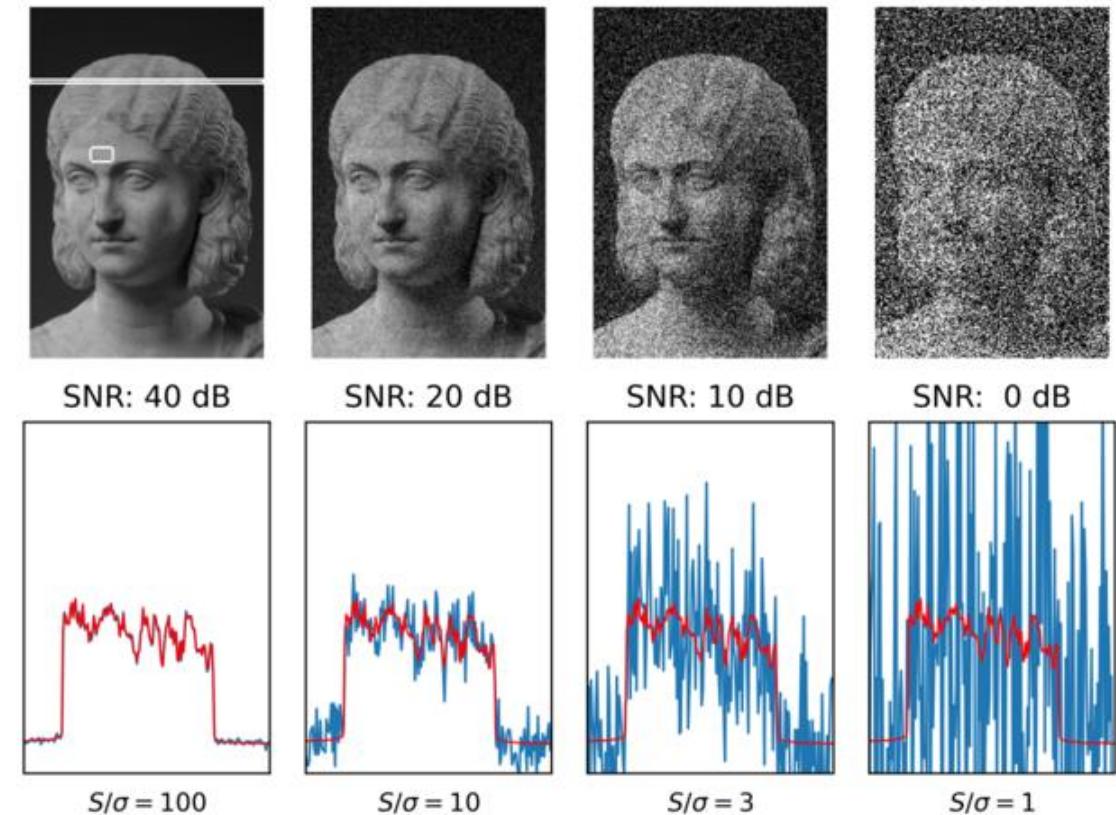


Image with speckle noise



### 3. Các phương pháp khử nhiễu

- Mục đích hướng đến của các phương pháp khử nhiễu
  - Giảm nhiễu trong hình ảnh tự nhiên
  - Giảm thiểu việc mất các tính năng gốc và cải thiện độ nhiễu tín hiệu (SNR – Signal Noise ratio)





## Khôi phục ảnh

Toán tử trên miền  
không gian

Toán tử trên miền  
tần số

- Lọc sắc nét
- Lọc trung bình
- Lọc trung vị
- Lọc cực đại
- Lọc cực tiểu

- Lọc thông thấp
- Lọc thông cao
- Lọc high-boost

# THANG ĐIỂM CHUNG



8.5-10

Làm được giao diện GUI

7-8.5

Demo code



5-7

Giải thích phương pháp, cách  
tính toán

2-5

Lý thuyết:  
Định nghĩa  
Ý nghĩa để làm gì?  
...

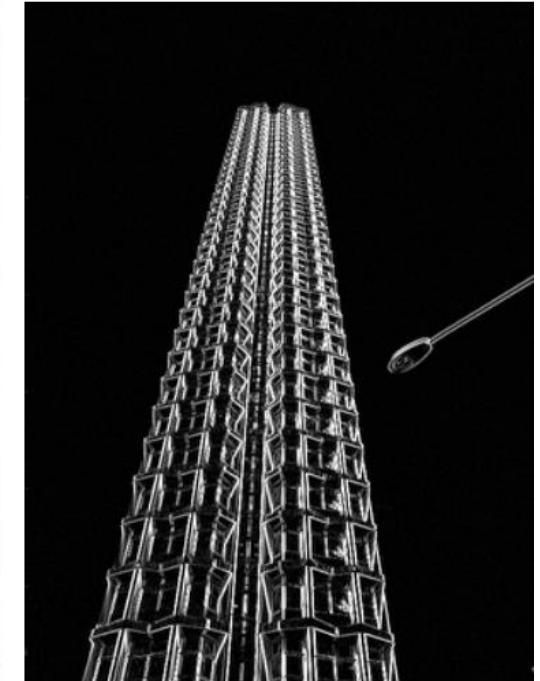


0-2

Slides trình bày, code

# Lọc sắc nét

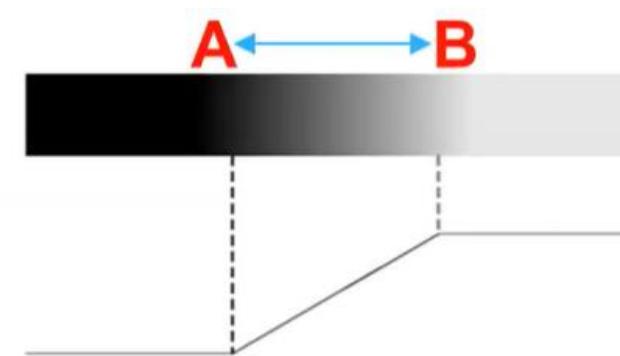
- Lọc sắc nét làm **nổi bật các biên** của ảnh
- Biên ảnh là tập hợp các pixel mà tại đó **cường độ xám thay đổi đột ngột**
- Lọc sắc nét = kỹ thuật là **nổi biên** ảnh



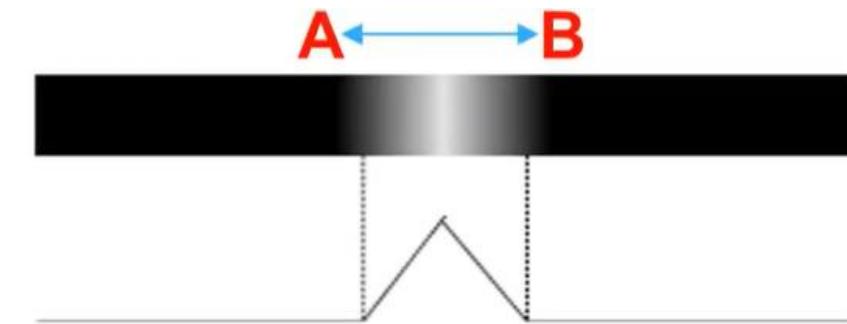
# Các loại biên



biên nhảy vọt (skip)



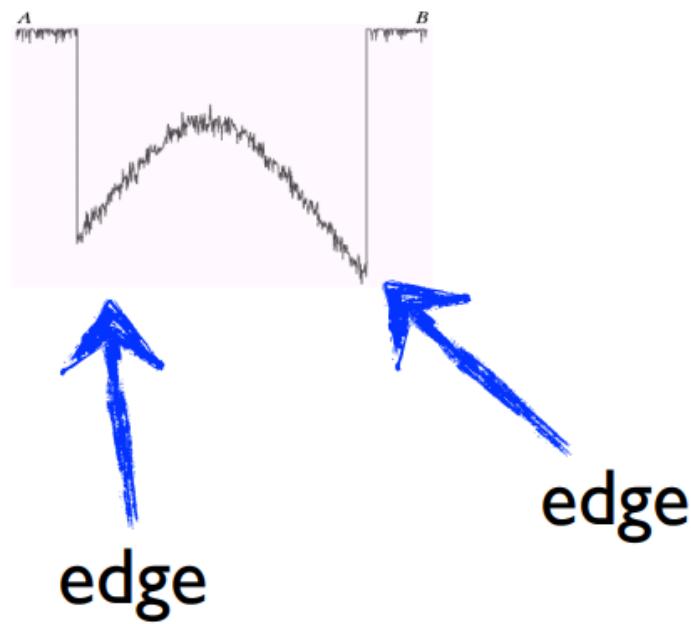
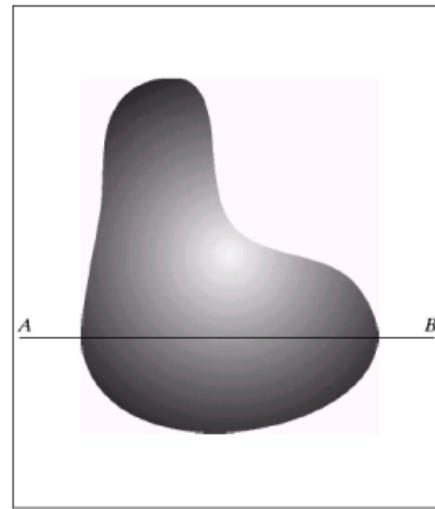
biên thoai thoả (Ramp)



biên kiểu mái nhà (Roof)



# Nhận dạng biên

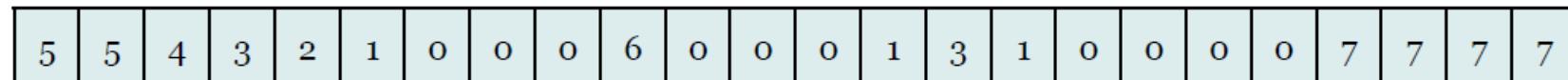


*How would you detect an edge?*

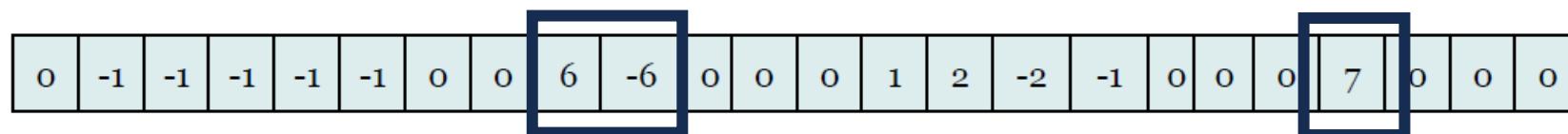
*What kinds of filter would you use?*

# Đạo hàm bậc một và đạo hàm bậc 2

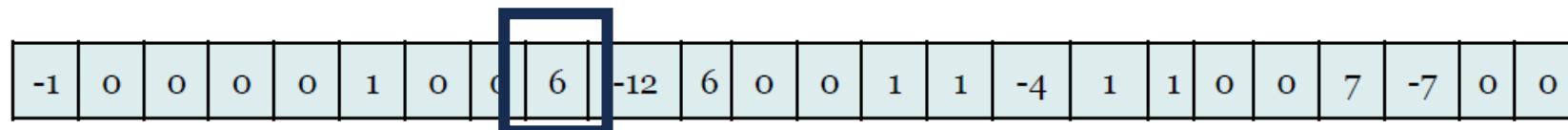
Đạo hàm bậc nhất	Đạo hàm bậc hai
$\frac{\partial f}{\partial x} = f(x+1) - f(x)$	$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$



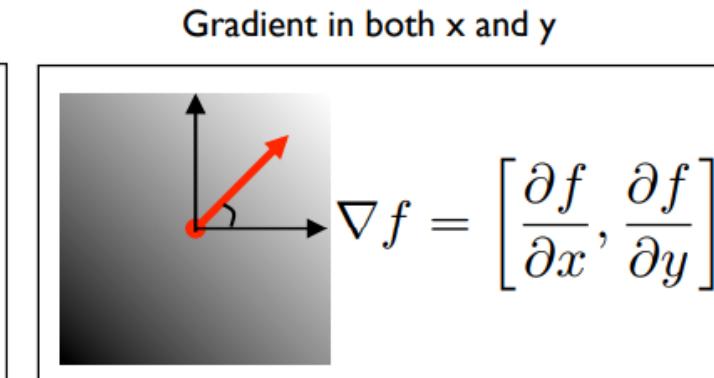
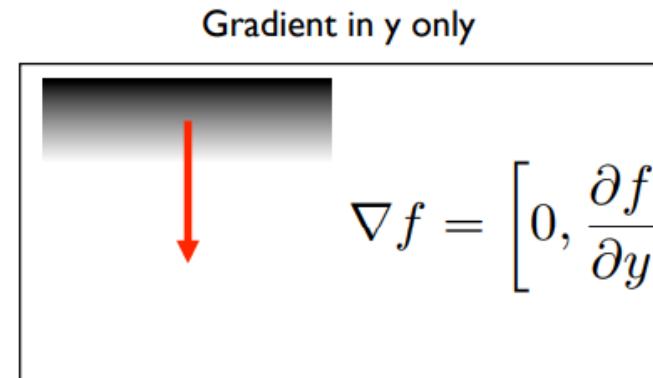
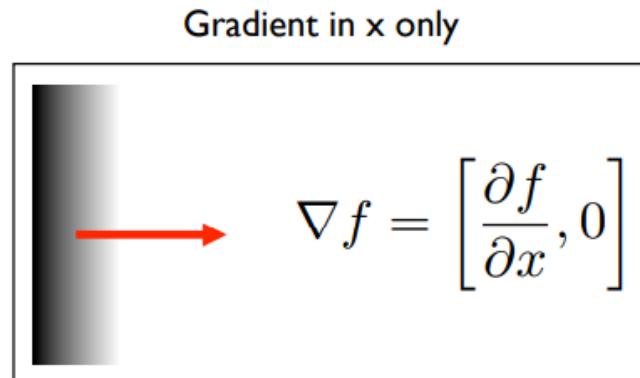
Đạo hàm bậc nhất



Đạo hàm bậc hai



# Toán tử Gradient



□ Các toán tử Gradient phổ biến

Sobel

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

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

Prewitt

1	0	-1
1	0	-1
1	0	-1

1	1	1
0	0	0
-1	-1	-1

# Toán tử Laplacian (mặt nạ Laplacian)

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

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

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

# Toán tử Laplacian (mặt nạ Laplacian)

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{If center coefficient is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{If center coefficient is positive} \end{cases}$$

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

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





# Demo & GUI

- Gradient gồm toán tử Sobel\_X và Sobel\_Y
- Toán tử Laplacian
- Phép chập



# Tạo GUI để xử lý ảnh

- Tạo nút nhấn browse để load ảnh đã bị nhiễu (hoặc tạo nhiễu bằng code) hoặc ảnh gốc chưa xử lý
- Áp các bộ lọc trung bình, bộ lọc trung vị, bộ lọc cực đại, cực tiểu (...), xuất ảnh sau khi xử lý để so sánh
- Show các ảnh sau khi đã áp bộ lọc

# Hướng dẫn dùng Tkinter để tạo giao diện

