# Chương 2: Cải thiện ảnh

## Bài 3:

### **Thang mức sáng**

2^n(n là số bit mà bức ảnh có).

* VD: Ảnh 4 bit = 2^4 = 16 ([0,15]) mức xám

### **Histogram**

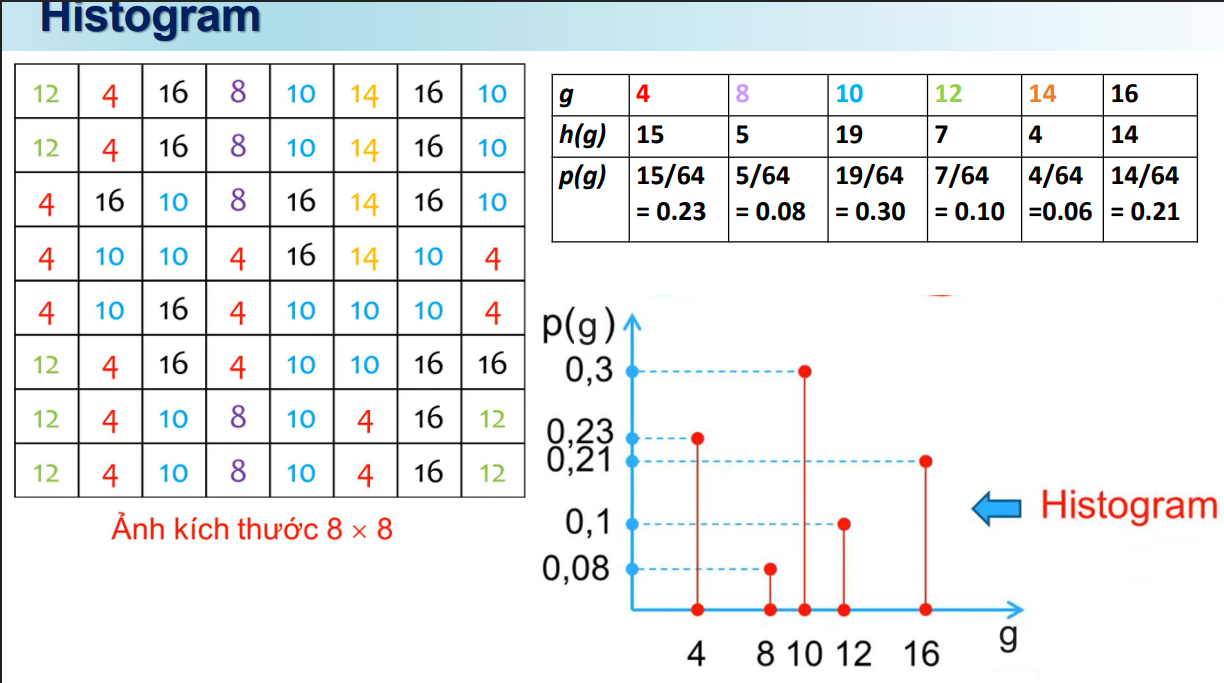
Là biểu đồ tần suất thống kê số lần xuất hiện các mức sáng trong ảnh

g : Cường độ mức xám.

H(g) : Số lượng điểm ảnh tương ứng với mức xám.

H(g) = h(g) = n(g)

p(g) : Biểu đồ chuẩn hóa. = h(g)/(M\*N)



### Contrast stretching

Có L: Tổng số lượng các mức xám (16,32,64,..256):

### Histrogram Equalization

g : Cường độ mức xám, g ⊂ [0, L-1]

Hàm cân bằng: s = T(g)

Các bước làm:

B1: Tính p(g) cho từng mức xám như trên histogram tính Biểu đồ chuẩn hóa.

B2: Tính hàm phân phối tích lũy:

Ck= Ck-1+ Pk

Sk = round((L-1)\*(Ck))

**VD: 5 bit => L = 32**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **g** | 4 | 8 | 10 | 12 | 14 | 16 |
| **h(g)** | 15 | 5 | 19 | 7 | 4 | 14 |
| **p(g)** | 15/64=0.23 | 5/64=0.08 | 19/64=0.3 | 7/64=0.11 | 4/64=0.06 | 14/64=0.22 |
| **Ck(g)** | 0.23+0=0.23 | 0.08+0.23=0.31 | 0.3+0.31=0.61 | 0.61+0.11=0.72 | 0.72+0.06=0.78 | 0.77+0.21=1 |
| **Sk** | 0 | 0.08/0.77\*31=3 | 15 | 20 | 22 | 31 |

### Histogram Stretching(Final)

Trong đó Cmax = 1, Cmin > 0

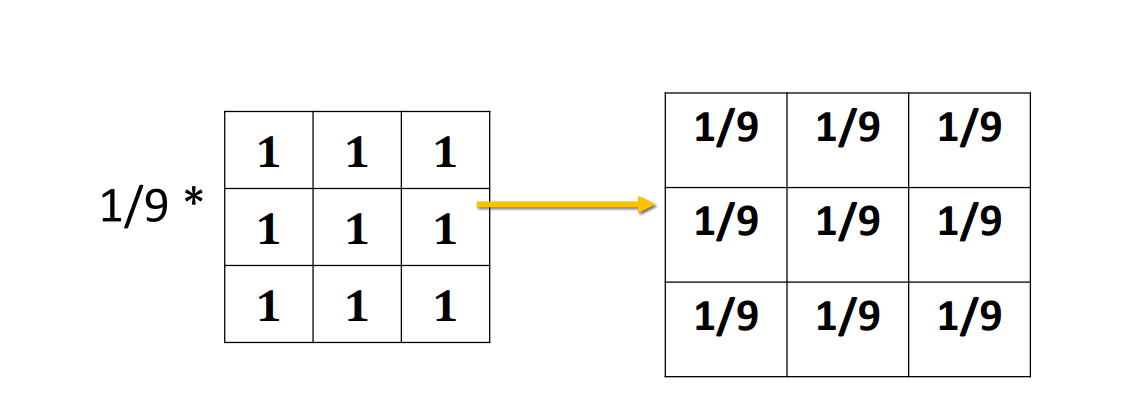
## Bài 4: Filtering

### Lọc mịn: Làm mờ ảnh

#### Lọc trung bình

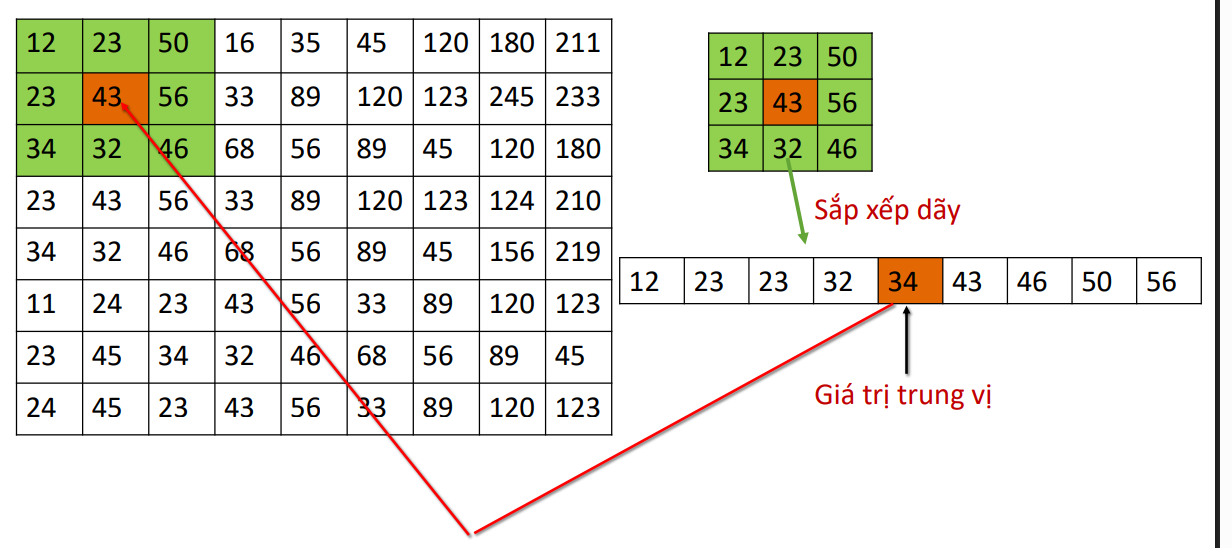
Lấy các điểm ứng với kernel để chập cộng lại chia kernel size

Kernel 3\*3 = 9



#### Lọc trung vị

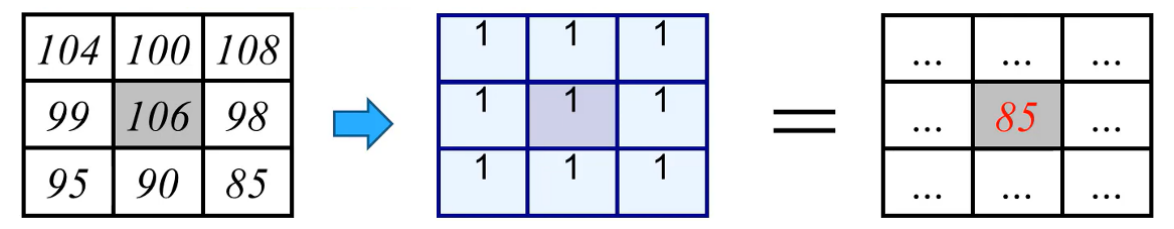
Lấy các điểm ứng với kernel để chập và sắp xếp theo thự tự tăng dần hoặc giảm dần lấy ra phần tử ở giữa nếu kernel chẵn thì lấy phần từ ở vị trí kernel size/2 hoặc (kernel size/2) +1



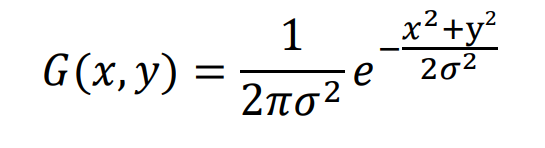
#### Lọc min or max

Lấy giá trị nhỏ nhất hoặc lớn nhất trong các điểm trong kernel được áp vào tâm anchor point

**VD:** min



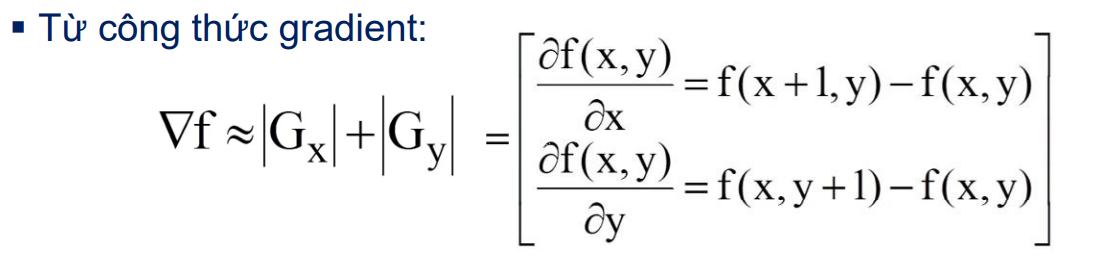
#### Lọc Gauss

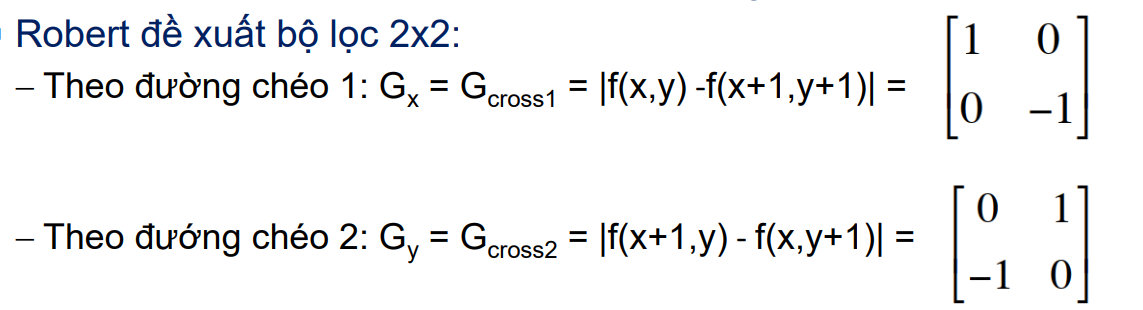
**CT: **

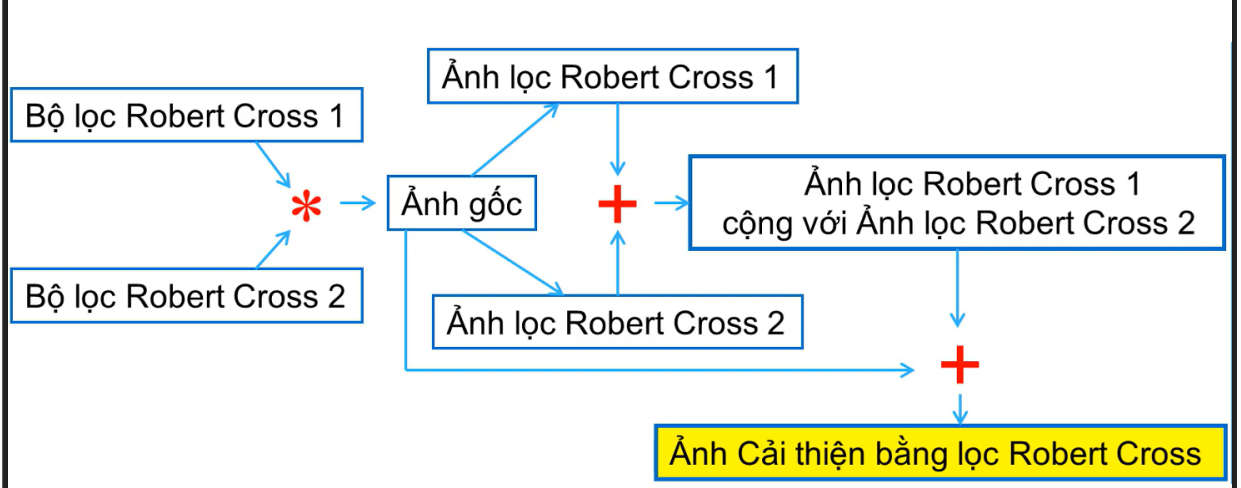
### Lọc sắc nét: Là làm nổi bật chi tiết ảnh

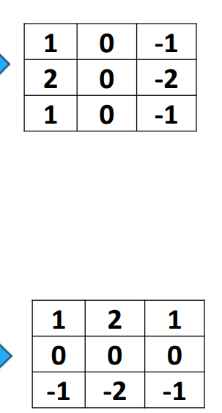
#### Đạo hàm bậc 1

* **Robert cross gradient:**

****

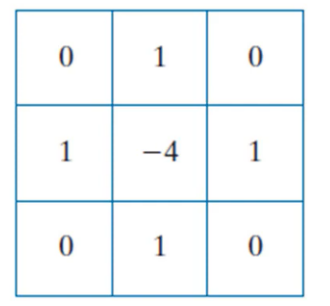
****

****

**Bộ lọc Sobel:** Công thức cái thiện ảnh tương tự robert cross

#### Đạo hàm bậc 2

* **Laplacian:**

****

Còn nhiều biến thể khác

### Unsharp masking

Lấy ảnh gốc trừ đi ảnh được làm mờ từ ảnh gốc: gmask(x,y)= f(x,y)-fLP(x,y)

### High-Boost Filter

g(x,y) = f(x,y) + kgmask(x,y)

Trong đó g(x,y) là ảnh sắc nét thu được bằng highboot filter

gmask là ảnh sắc nét thu được từ unsharp masking

fLP ảnh được làm mờ từ ảnh gốc

## Bài 5: Lọc ảnh trong miền tần số

### Xử lý (lọc) ảnh trong miền tần số

**B1:** f(x,y) là ảnh gốc (đầu vào) cần xử lý

− Biến đổi ảnh sang miền tần số dùng DFT

− Sắp xếp lại (hoán chuyển 4 phần tư ảnh q1-q4, q2-q3) để chuyển tần số thấp về tâm ảnh ta thu được F(u,v)

− Để hiển thị phổ (Spectrum) sử dụng (log(1+ |F(u,v)|)

**B2:** Thiết kế bộ lọc

− Bộ lọc (mask) có thể là hình tròn, hình chữ nhật, đường thẳng, v,v

− Nếu bộ lọc thiết kế trong miền không gian h(x,y) thì cần biến đổi sang miền tần số ta thu được H(u,v)

− Nếu bộ lọc thiết kế trong miền tần số thì bọ lọc đó chính là H(u,v)

**B3:** Lọc ảnh bằng phép nhân ta thu được ảnh lọc trong miền tần số G(u,v) = F(u,v). H(u,v)

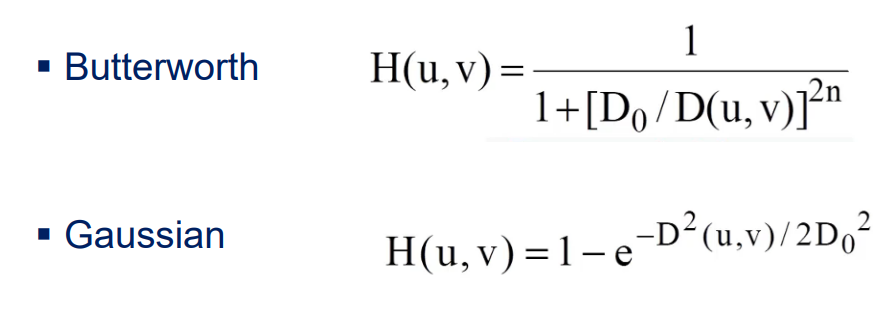
**B4:**

− Sắp xếp lại (hoán chuyển lại 4 phần tư ảnh q1-q4, q2-q3)

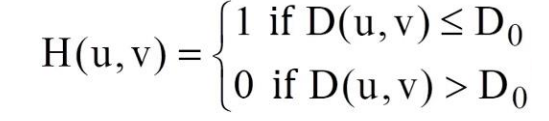
− Biến đổi ngược dùng IDFT

**B5**: Lấy độ lớn (Amplitude) hay biên độ (Magnitude) ta thu được ảnh đã xử lý trong miền không gian g(x,y)

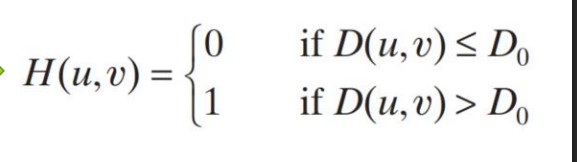
### Lọc thông thấp, cao:



#### Ct làm mịn ảnh(Lọc thông thấp)

****

#### Ct làm sắc nét (Lọc thông cao)

****

# Chương 3: Khôi phục ảnh

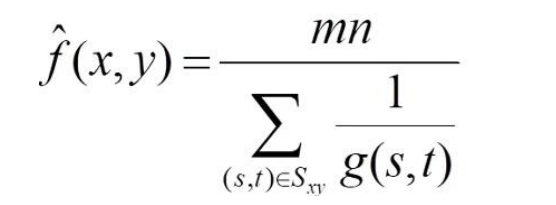
## Bài 6:

### Lọc trung bình (mean):

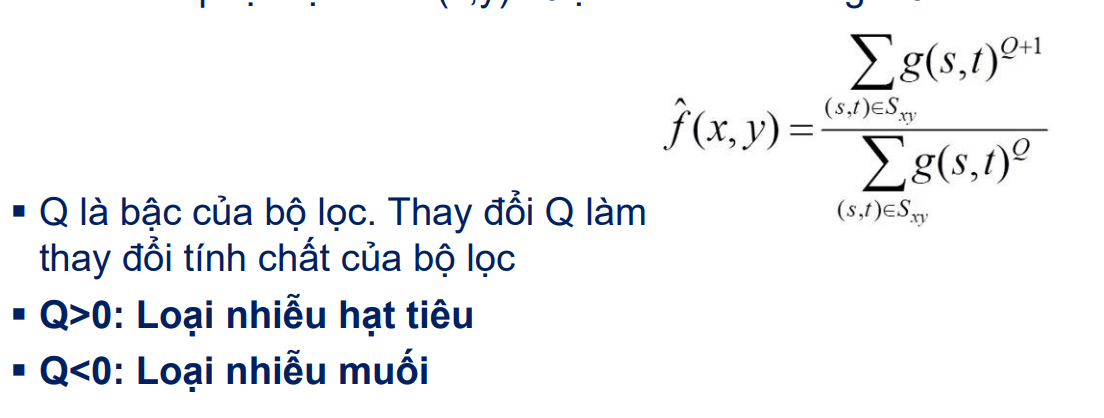
Làm mịn đơn giản, làm mờ ảnh để loại nhiễu.

### Lọc trung bình Harmonic:

Làm việc tốt với nhiễu muối và nhiễu Gaussian, không tốt với nhiễu hạt tiêu.

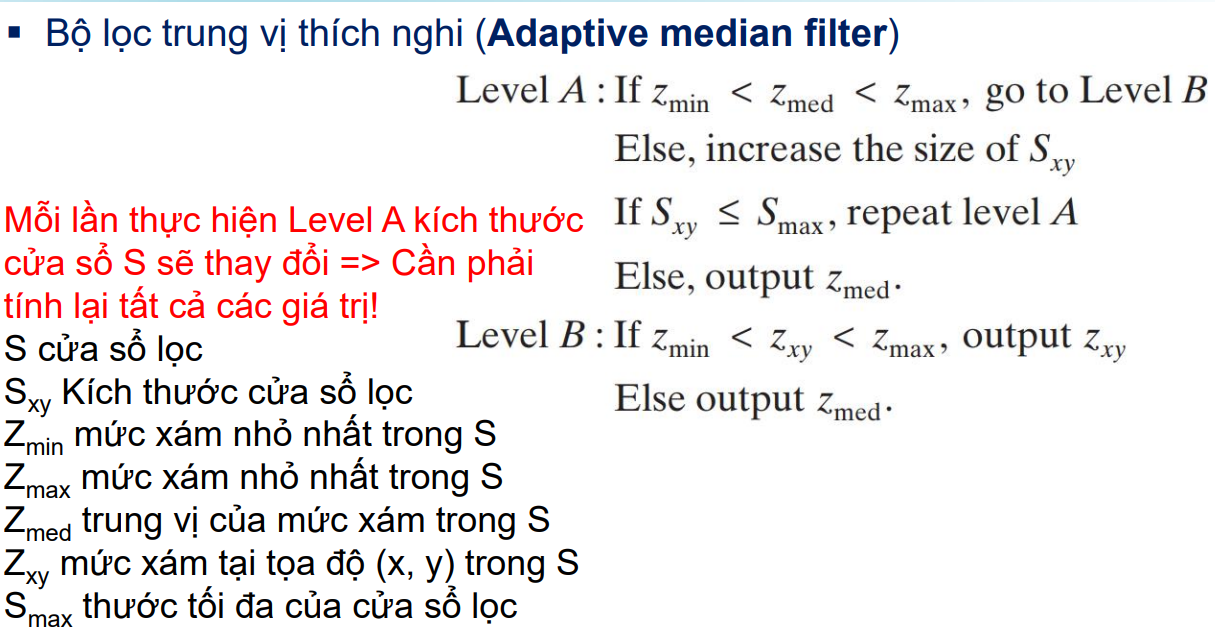


### Lọc trung bình ContraHarmonic:

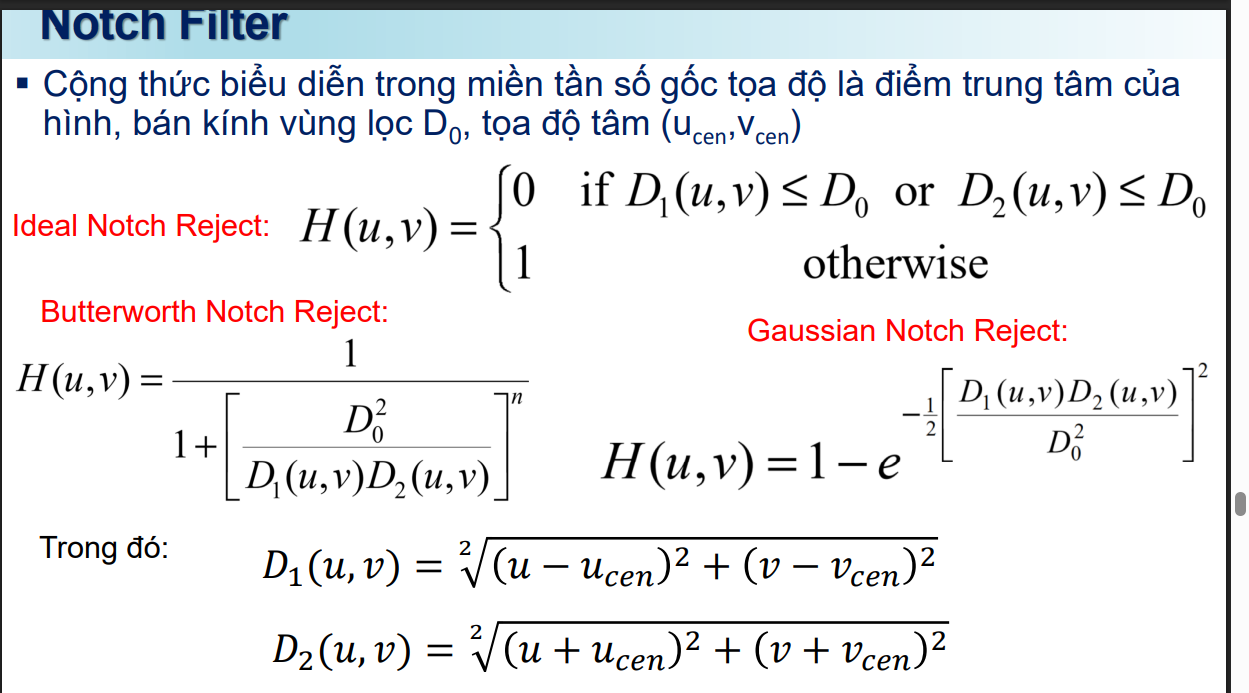


* Lọc trung vị (median): Hiệu quả cho lọc nhiễu muối và hạt tiêu.
* Lọc min: Hiểu quả cho lọc nhiễu muối
* Lọc max: Hiểu quả cho lọc nhiễu hạt tiêu

### Bộ lọc thích nghi



### Notch filter



# Chương 4 (Bài 7): Phép biến đổi

## Biến đổi hình thái học

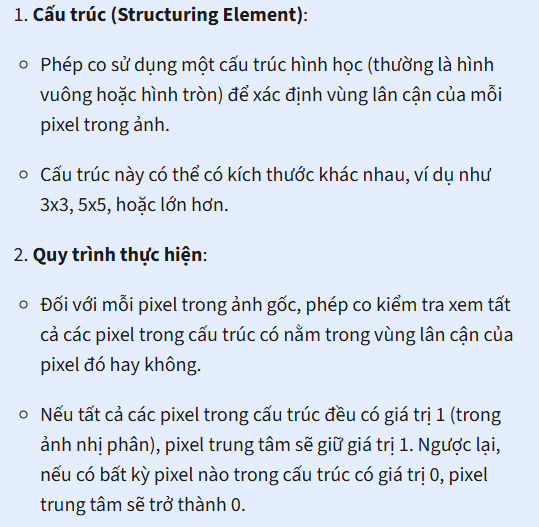
### Phép dãn

**Nguyên tắc hoạt động**:

Trong ảnh nhị phân, tại một điểm cần xét, giá trị đầu ra sẽ là 1 nếu có ít nhất một điểm trong cấu trúc (Structuring Element) nằm trong vùng lân cận của điểm đó.

Nếu không có điểm nào trong cấu trúc, giá trị đầu ra sẽ là 0.

### Phép co

****

# Chương 5: Phân vùng ảnh

## Bài 8: kỹ thuật cắt ngưỡng

### Thuật toán tìm Ngưỡng toàn cục cơ bản

**B1:** Chọn giá trị ngưỡng T0 (Thường là trung bình mức xám của ảnh)

**B2**: Sử dụng T0 để phận đoạn ảnh, tạo thành 2 nhóm điểm ảnh:

* Nhóm G1 chứa các pixel có mức xám >= T­0
* Nhóm G2 chứa các pixel có mức xám < T­0

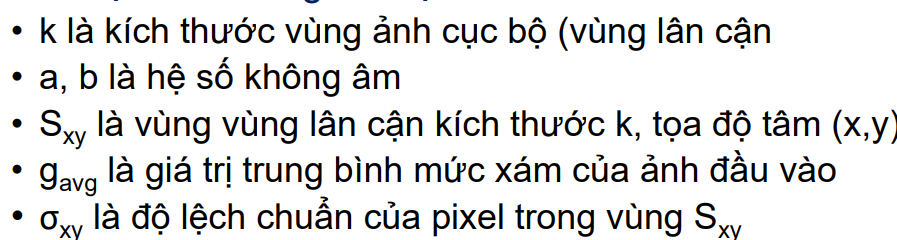
**B3**: Tính mức xám trung bình nhóm G1 là m1 và G2 là m2

**B4**: Tính ngưỡng mới: T = (m1+m2)/2

**B5**: Tính deltaT = |T - T0|

**B6**: Coi T0 = T lặp lại B2 đến B5 cho đến khi deltaT nhỏ hơn ngưỡng thres cho trước

### Thuật toán Adaptive Local thresholding



B1: Tính gavg toàn ảnh(mức xám trung bình)

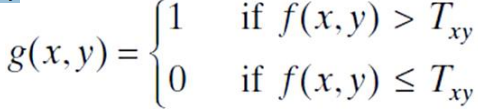
B2: Với mỗi điẻm xy của ảnh đầu vào:

B3: Xác định vùng lân cận Sxy

B4: Tính độ lệch chuẩn vùng lân cận σxy

B5: Tính ngưỡng Txy = a \* σxy + b\*gxy­

B6: Ảnh phân đoạn được tính bằng:

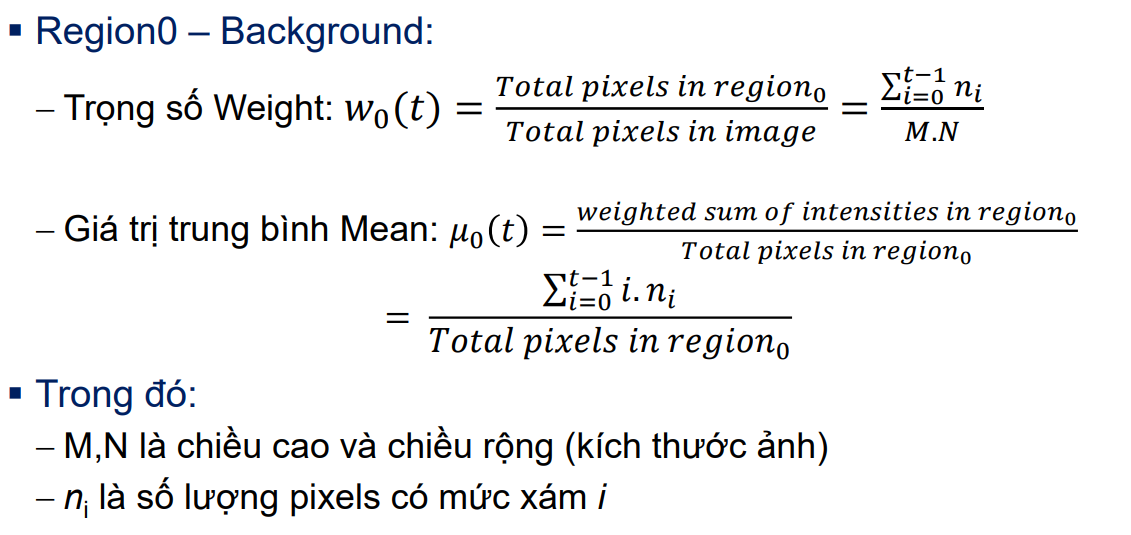


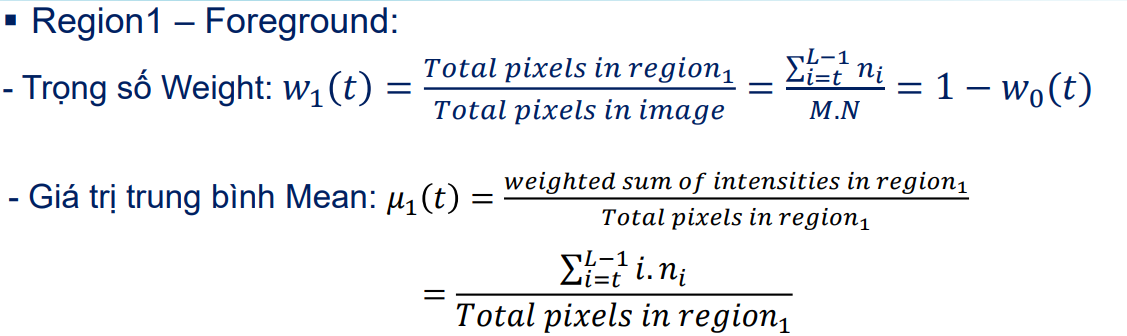
### Thuật toán tìm Ngưỡng Otsu

**Với ngưỡng T ban đầu 0 <= T <= L-1 sẽ được chia thành 2 vùng:**

* Vùng 0 (Background): là vùng chứa tất cả các pixel có giá trị xám nhỏ hơn T
* Vùng 1 (Foreground): là vùng chứa tất cả các pixel có giá trị xám [T,..L-1]

#### Tính trọng số weight ω(t) và giá trị trung bình Mean μ(t) cho mỗi vùng





#### Tìm phương sai:

****

**Lặp lại đến hết ngưỡng L-1 lấy T có phương sai lớn nhất**

# Chương 6 (Bài 10): Kỹ thuật trích rút đặc trưng

## **HoG**

Thuật toán này sẽ tạo ra các bộ mô tả đặc trưng (feature descriptor) nhằm mục đích phát hiện vật thể (object detection).

− Từ một bức ảnh, ta sẽ lấy ra 2 ma trận quan trọng giúp lưu thông tin ảnh đó là độ lớn gradient (gradient magnitute) và phương của gradient (gradient orientation).

− Kết hợp 2 thông tin này vào một biểu đồ phân phối histogram, trong đó độ lớn gradient được đếm theo các nhóm bins của phương gradient. Cuối cùng ta sẽ thu được véc tơ đặc trưng HoG đại diện cho histogram.

**Các bước xử lý**

**B1:** Tiền xử lý(chuẩn hóa kích thước, chuyển sang ảnh xám,..)

**B2:** Tính gradient

**B3:** Tính vector đặc trưng cho mỗi ô (cells) – thường kích thước là 8\*8 (Bài tập chỉ làm đến B3)

**B4:** Chuẩn hóa blocks – 1 block = 4 cells

**B5:** Tính toán đặc trưng HoG

VD: cho ma trận 8\*8 sử dụng padding giữ kích thước output = input

F(x,y) =

[ 52, 55, 61, 59, 79, 61, 76, 61]

[ 62, 59, 55, 104, 94, 85, 59, 71]

[ 63, 65, 66, 113, 144, 104, 63, 72]

[ 64, 70, 70, 126, 154, 109, 71, 69]

[ 67, 73, 68, 106, 122, 88, 68, 68]

[ 68, 79, 60, 70, 77, 66, 58, 75]

[ 69, 85, 64, 58, 55, 61, 65, 83]

[ 70, 87, 69, 68, 65, 73, 78, 90]

**B1:**

=> Zero padding

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[ 0, 52, 55, 61, 59, 79, 61, 76, 61, 0]

[ 0, 62, 59, 55, 104, 94, 85, 59, 71, 0]

[ 0, 63, 65, 66, 113, 144, 104, 63, 72, 0]

[ 0, 64, 70, 70, 126, 154, 109, 71, 69, 0]

[ 0, 67, 73, 68, 106, 122, 88, 68, 68, 0]

[ 0, 68, 79, 60, 70, 77, 66, 58, 75, 0]

[ 0, 69, 85, 64, 58, 55, 61, 65, 83, 0]

[ 0, 70, 87, 69, 68, 65, 73, 78, 90, 0]

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

**B2:**

G(x,y) = (Gx2+Gy2)\*\*1/2 VD: Mag(0,0)

Gx = F(x+1,y) – F(x-1,y) => Gx0 = 55 - 0 = 55

Gy = F(x,y+1) – F(y-1) Gy0 = 62 - 0 = 62

Φ = arctan(Gy/Gx) G(0,0)  = Mag(0,0) = ((55)2 + (62)2)1/2 = 82.88

Φ = arctan(62/55) = 48.42 (deg)

Tính đến hết bảng

**B3: Tính Histogram có 9 bins**

**=> 1 bin = 180/9 = 20**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mag |  |  | 47.99 | 34.89 |  |  |  |  |  |
| Bins | 0 | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 |
| Feature | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Mag(0,0) = 82.88

Orient = 48.42 => nằm giữa 40 và 60

Mag40 = (60 – 48.42)/20 \* 82.88 =47.99

Mag60 = 82.88 – 47.99 = 34.89

Cứ tiép tục tính hét cộng tích lũy