

Xử lý ảnh và Thị giác máy tính



Chương 3. Khôi phục và tái tạo ảnh

- Low-pass filter
- High-pass filter



Khôi phục ảnh

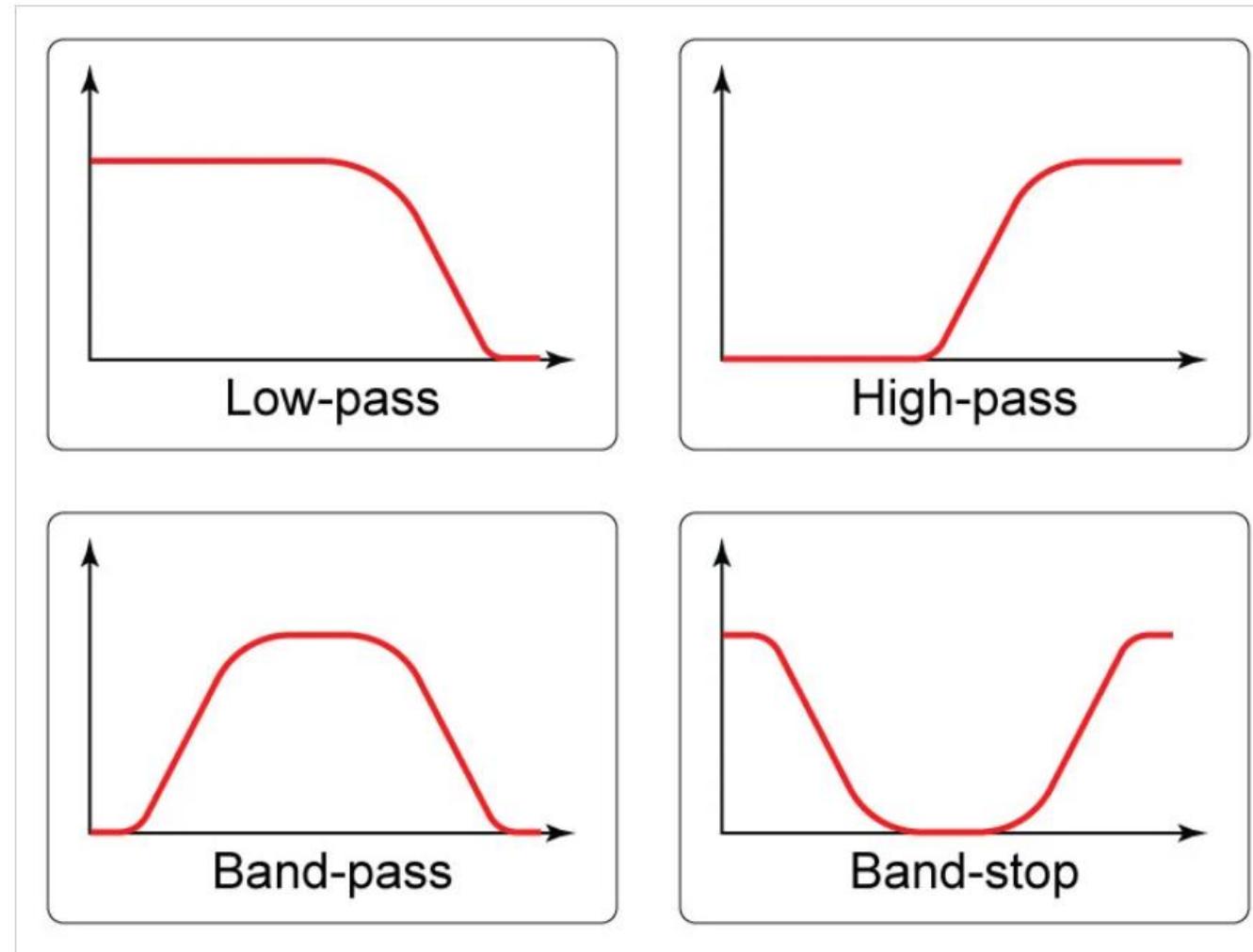
Toán tử trên miền
không gian

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

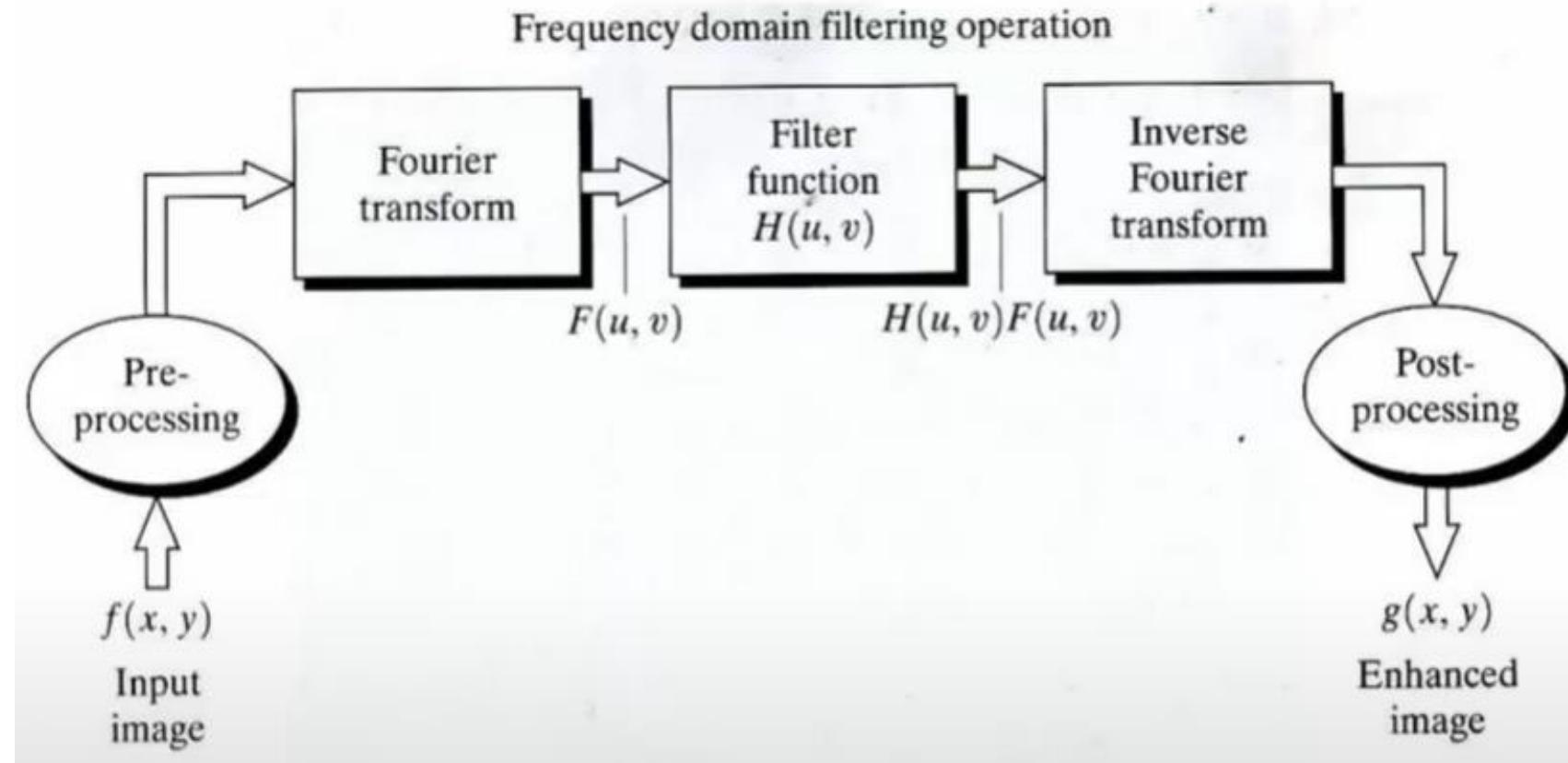
Toán tử trên miền
tần số

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

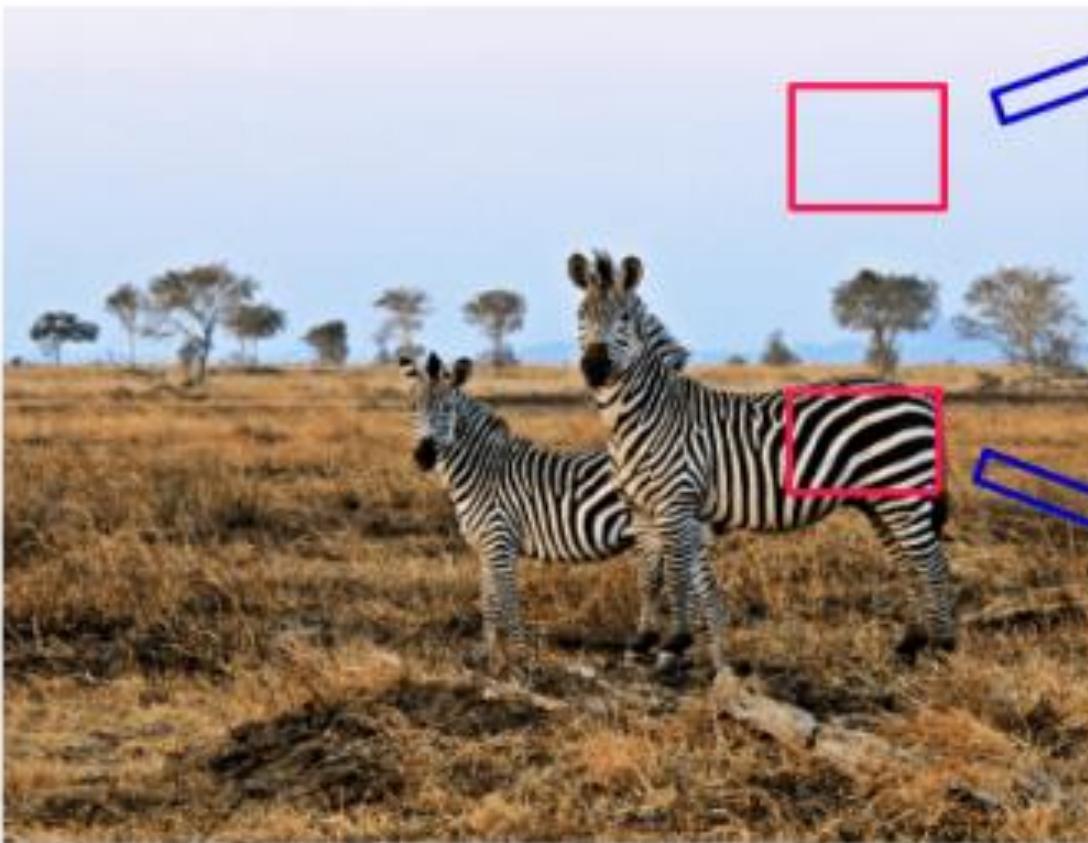
Các bộ lọc cơ bản trong miền tần số



Lọc trong miền tần số



Các thành phần tần số trong bức ảnh



Mức xám thay đổi
ít hoặc không
thay đổi

Low Frequency



Mức xám thay đổi
đột ngột và mạnh

High Frequency

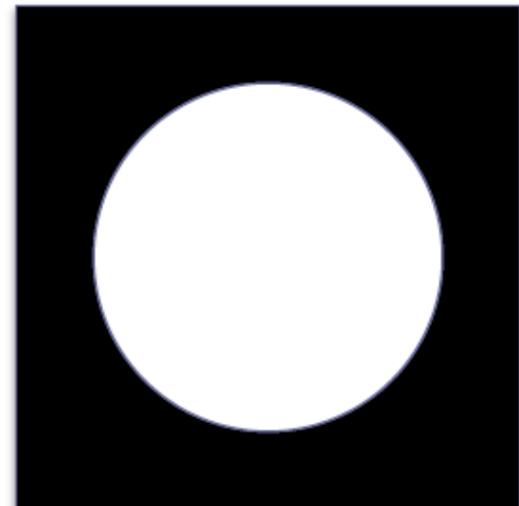
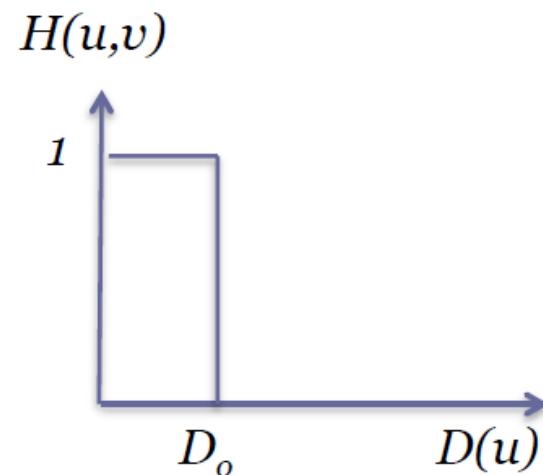
Lọc thông thấp lý tưởng (ideal low-pass filter)



- Cho thành phần tần số thấp đi qua
- Loại bỏ các thành phần tần số cao của biến đổi Fourier mà khoảng cách tới tâm là $D(u, v)$ lớn hơn ngưỡng cắt (tần số cắt) D_0

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

$$D(u, v) = \left[(u - M / 2)^2 + (v - N / 2)^2 \right]^{1/2}$$





Tính chất & Ứng dụng

- Tần số cắt càng lớn ảnh sẽ càng được giữ được nhiều thông tin
- Lọc nhiễu
- Làm mịn (smooth) ảnh
- Tương tự với bộ lọc trung bình trong miền không gian

Ví dụ lọc thông thấp lý tưởng

Original image



D0 =10



D0 =20



D0 =30



D0 =40

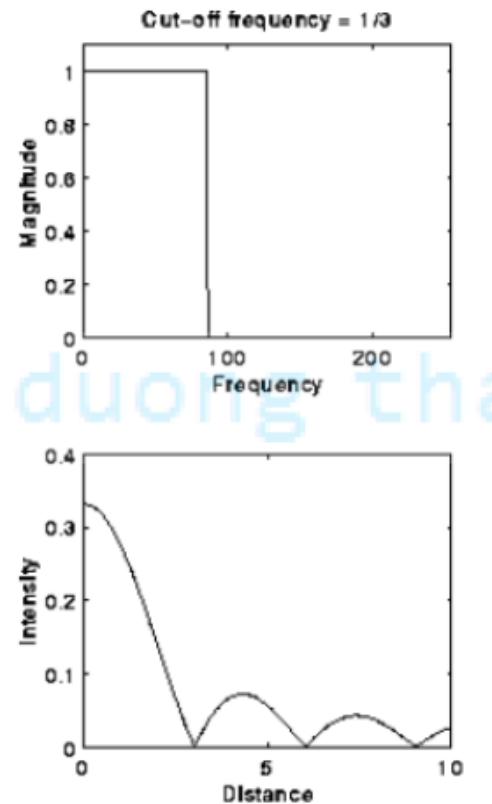


D0 =50



Nhận xét

- Đối với thông thấp lý tưởng, trong miền không gian sẽ tạo nên những gợn sóng → tạo ra hiệu ứng run ảnh (ringing)



đường thẳng

D0 = 20

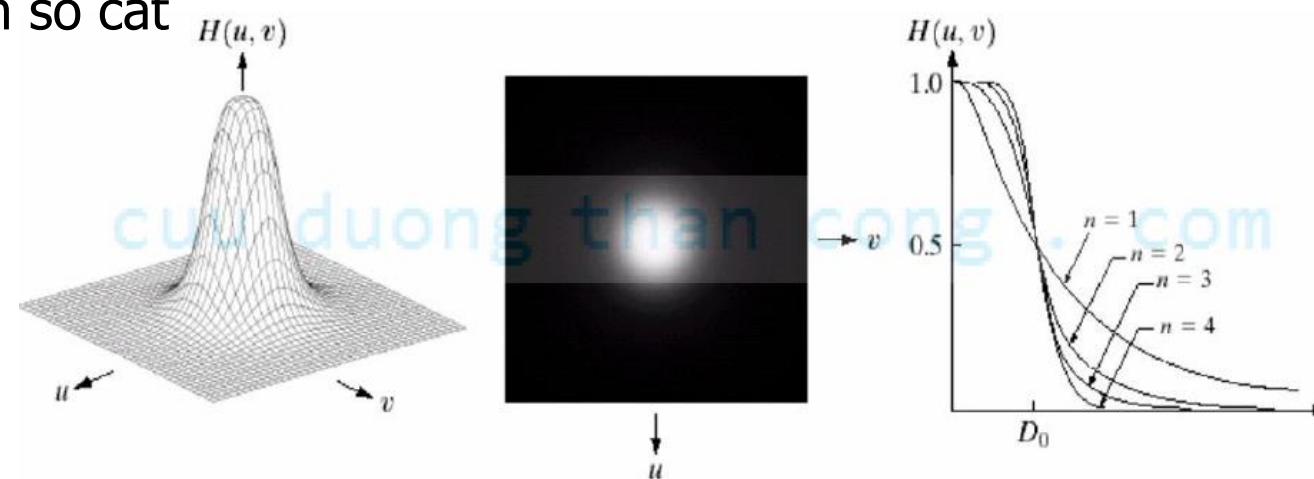


Lọc thông thấp kiểu Butterworth

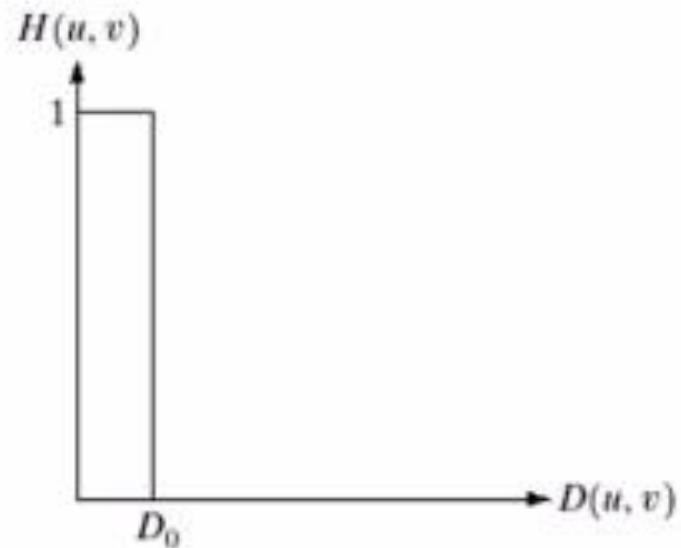
□ Công thức:

$$H(u, v) = \frac{1}{1 + (D(u, v) / D_0)^{2n}}$$

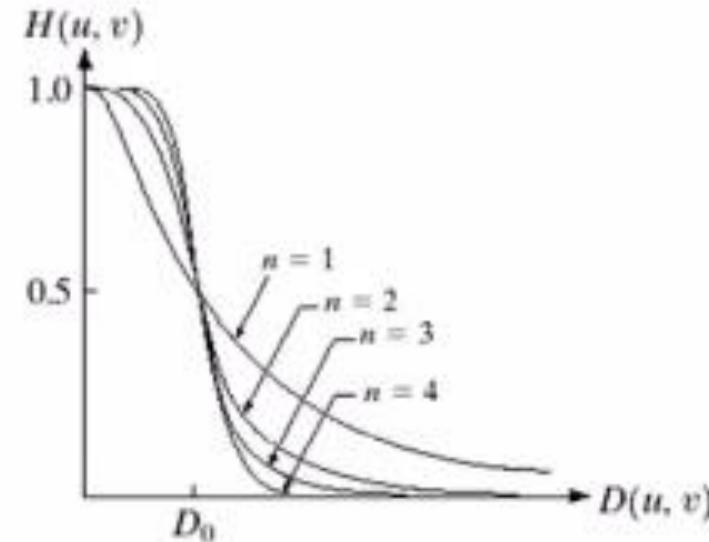
n là bậc, D_0 là tần số cắt



Lọc lý tưởng vs Butterworth



Ideal lowpass filter



Butterworth lowpass filter

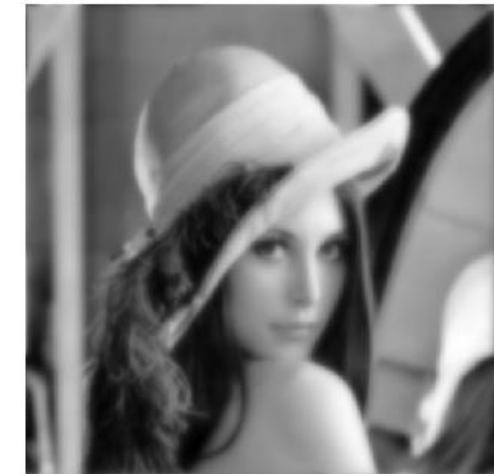
n (bậc) càng cao thì càng
tiến về bộ lọc lý tưởng

Butterworth low-pass filter $D_0 = 20$

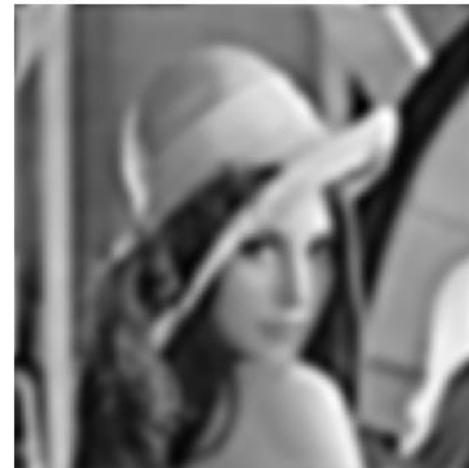
Original image



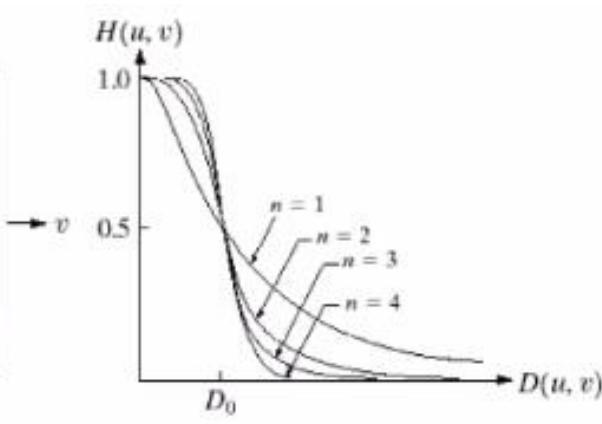
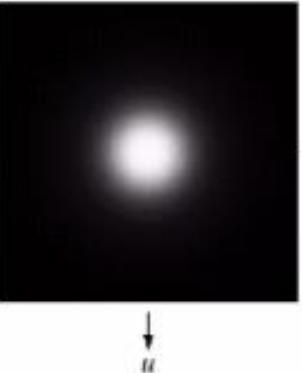
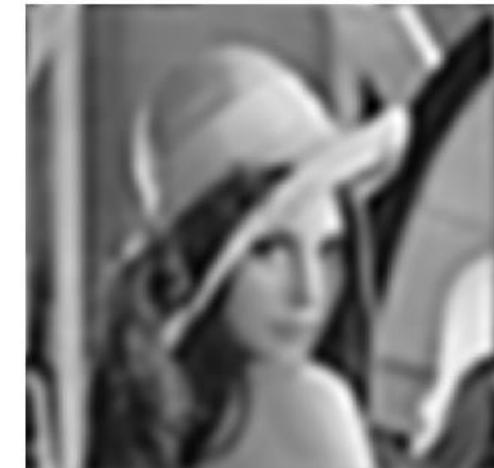
$n = 1$



$n = 3$



$n = 4$

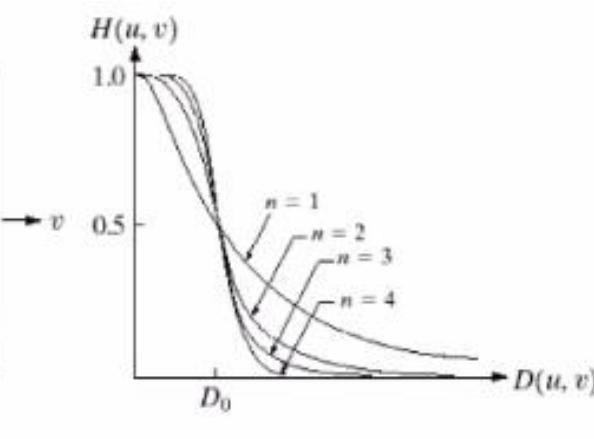
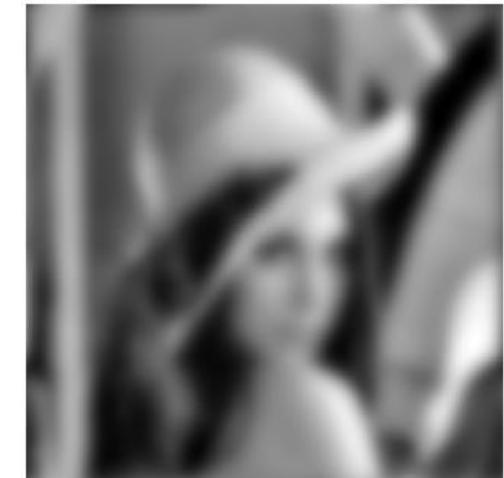


Ví dụ lọc thông thấp Butterworth n = 2

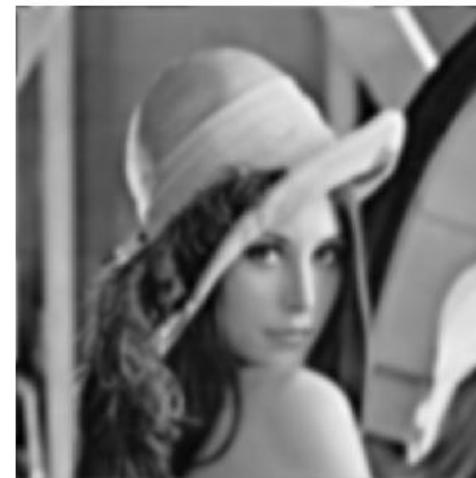
Original image



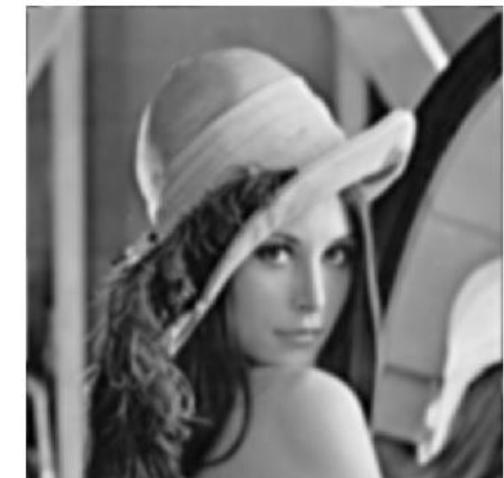
D0 =10



D0 =30



D0 =40

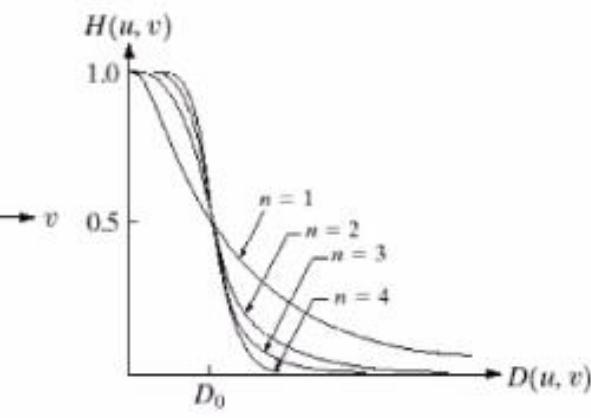
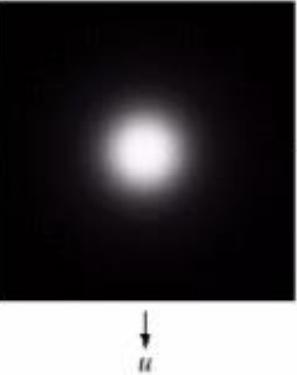
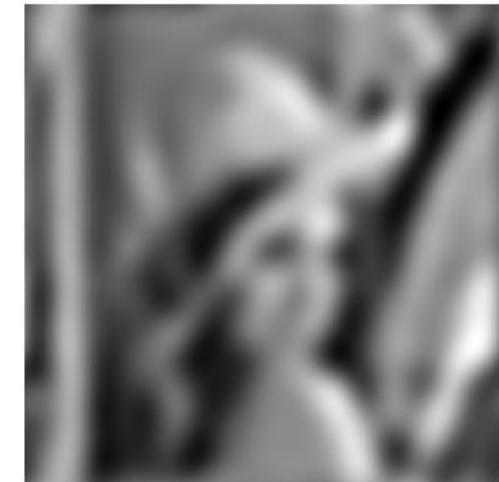


Ví dụ lọc thông thấp Butterworth n = 5

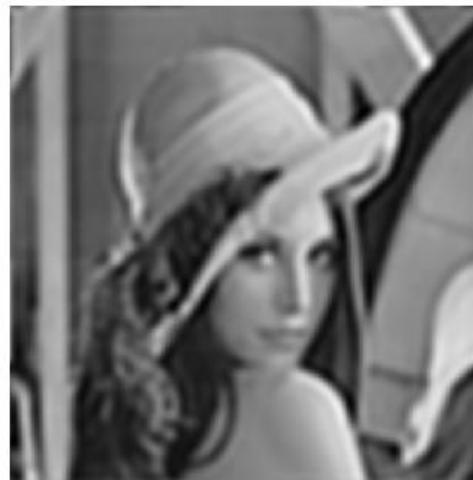
Original image



D₀ = 10



D₀ = 30

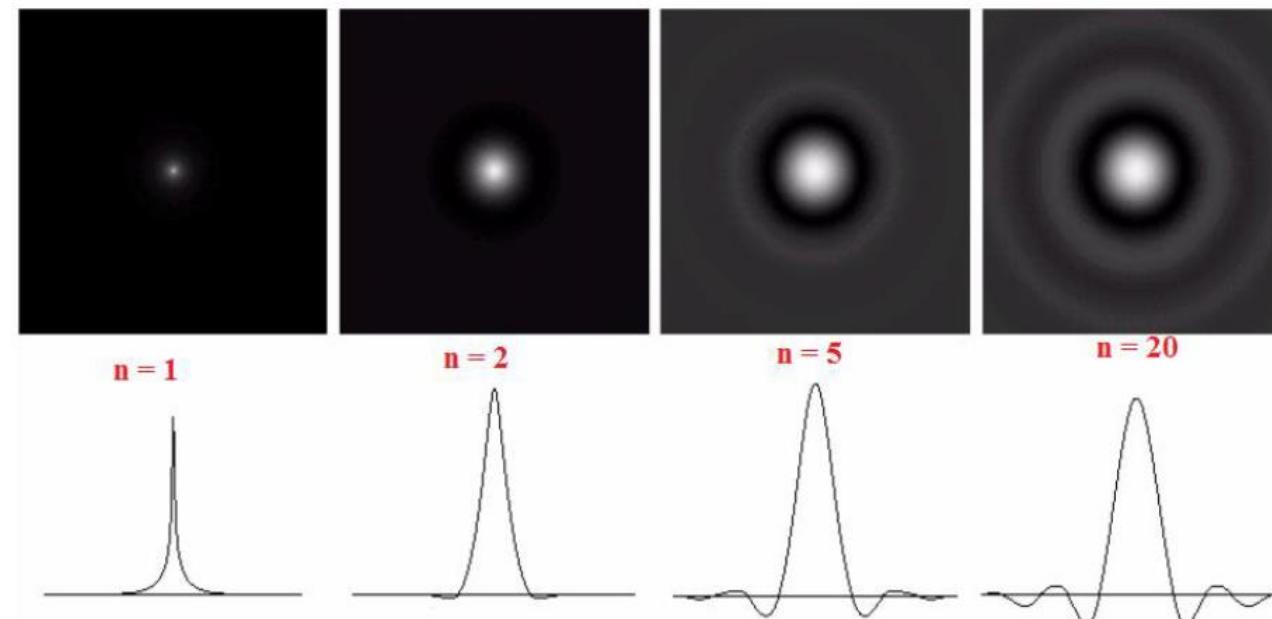


D₀ = 40



Bộ lọc Butterworth

- Giảm hiệu ứng run ảnh (ringing) so với bộ lọc lý tưởng
- Khi bậc của n nhỏ (từ bậc 1) hiệu ứng run ảnh không xuất hiện
- Khi bậc tăng dần hiện tượng run ảnh sẽ dần xuất hiện

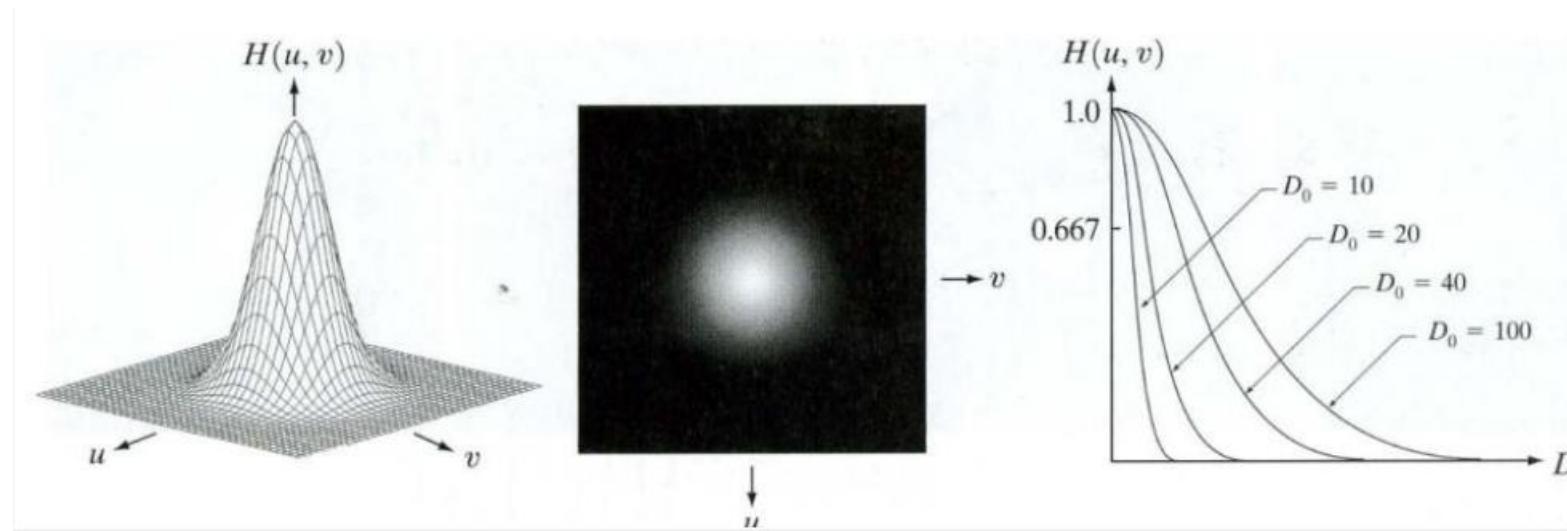


Lọc thông thấp Gaussian

□ Công thức:

$$H(u, v) = e^{-D^2(u, v)/2D_0^2}$$

Với D_0 là tần số cắt

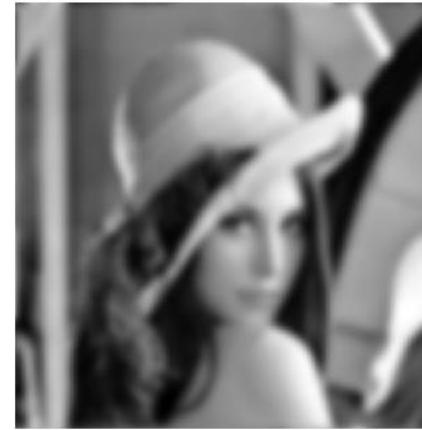


Ví dụ lọc thông thấp Gaussian

original image



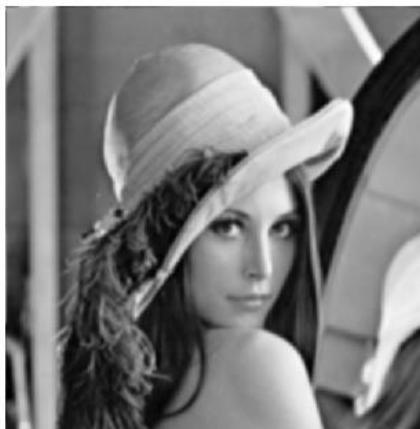
D0 =10



D0 =20



D0 =30



D0 =40



D0 =50



Ví dụ lọc thông thấp Gaussian

- Có thể lọc nhiễu Gauss hiệu quả, tương tự bộ lọc trung bình

original image



D0 =10



D0 =30

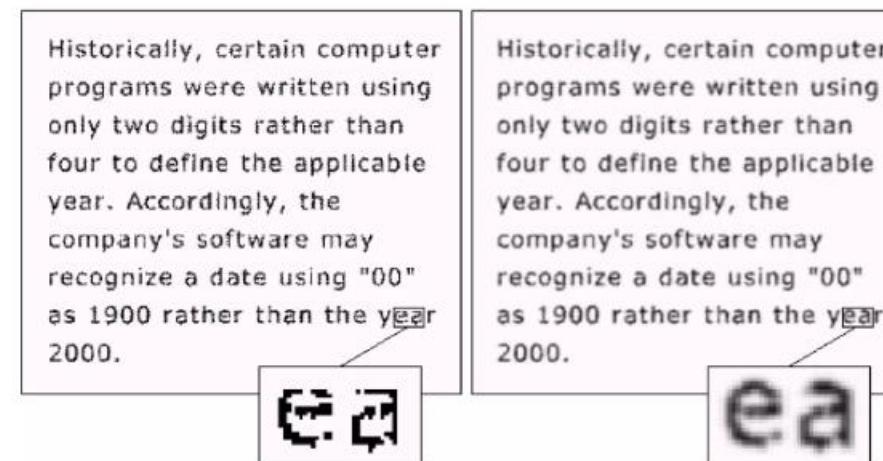


D0 =50



Ứng dụng

- Do biến đổi Fourier ngược cũng là hàm Gaussian nên sẽ không xảy ra hiện tượng run ảnh
- Làm mờ ảnh (xóa nếp nhăn trên mặt)
- Nối liền các đoạn đứt gãy chữ trong văn bản



Tóm lại

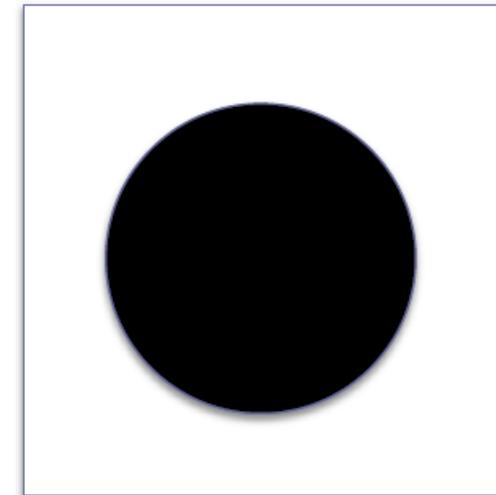
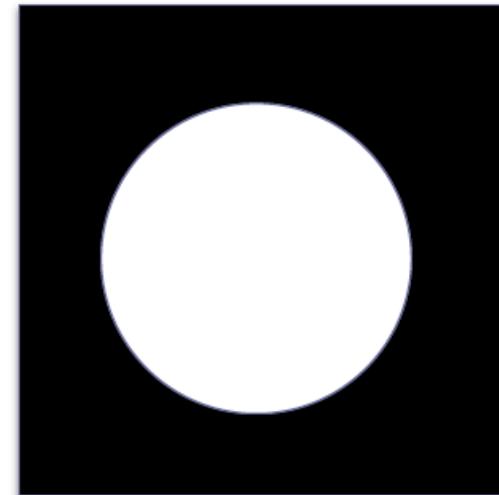
- Lọc thông thấp cho tần số thấp đi qua, tương đương với lọc trung bình trong miền không gian
- Làm mờ ảnh và lọc nhiễu

Bộ lọc	Công thức
Ideal lowpass filter (ILPF)	$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$
Butterworth lowpass filter (BLPF)	$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$
Gaussian lowpass filter (GLPF)	$H(u, v) = e^{-D(u,v)^2/2D_0^2}$

* D_0 : cut-off frequency, n : bậc của bộ lọc Butterworth

Lọc thông cao (high-pass filter)

- Làm suy yếu thành phần tần số thấp, cho thành phần tần số cao đi qua
- $H_{hp}(u, v) = 1 - H_{lp}(u, v)$



Lọc thông cao (high-pass filter)

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

- Lọc thông cao lý tưởng

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$$

- Lọc thông cao Butterworth

$$H(u, v) = \frac{1}{1 + (D_0 / D(u, v))^{2n}}$$

- Lọc thông cao Gauss

$$H(u, v) = 1 - e^{-D^2(u, v)/2D_0}$$

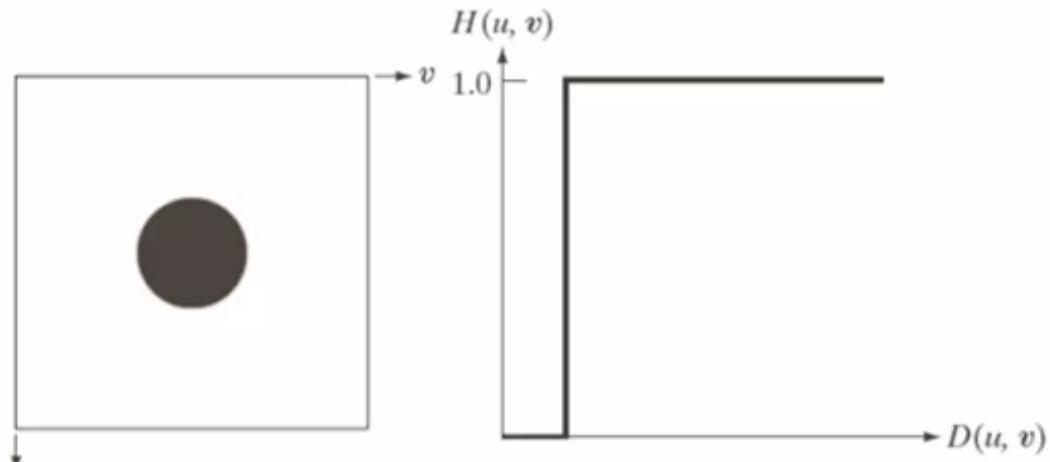
$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

$$H(u, v) = \frac{1}{1 + (D(u, v) / D_0)^{2n}}$$

$$H(u, v) = e^{-D^2(u, v)/2D_0^2}$$

Lọc thông cao lý tưởng

- D₀ càng cao phần
cạnh càng sắc nét



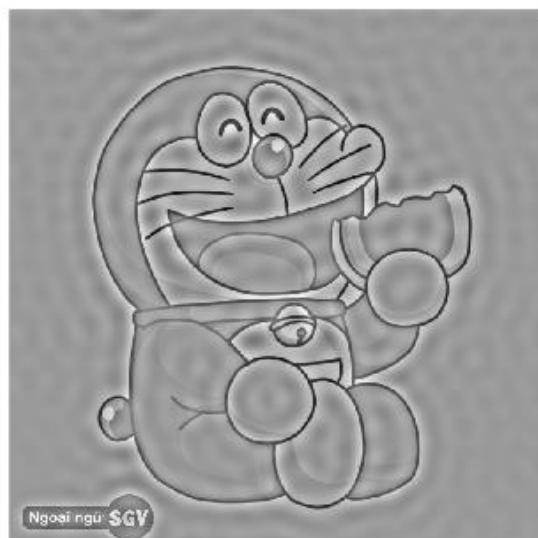
Original



D₀ =5



D₀ =20

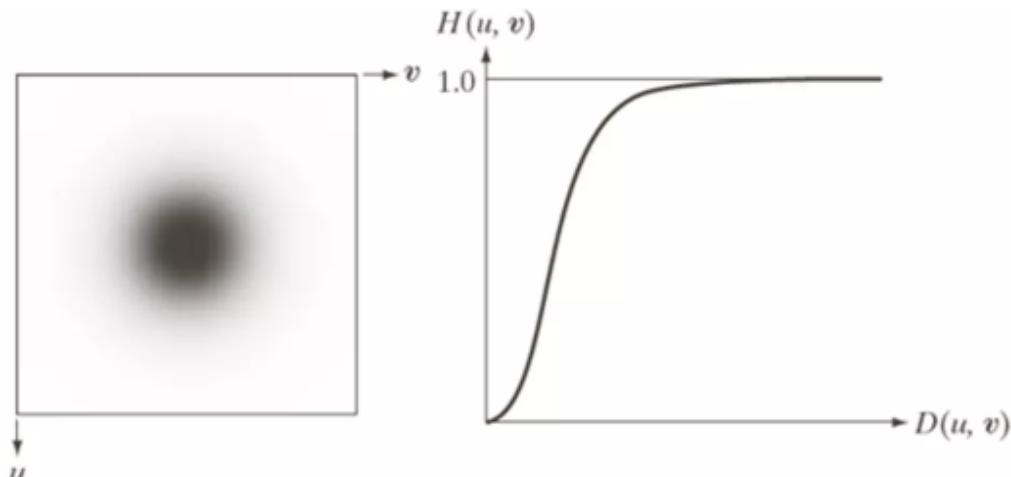


D₀ =35



Lọc thông cao Butterworth

Cố định bậc n = 2



Original



D0 =5



D0 =20



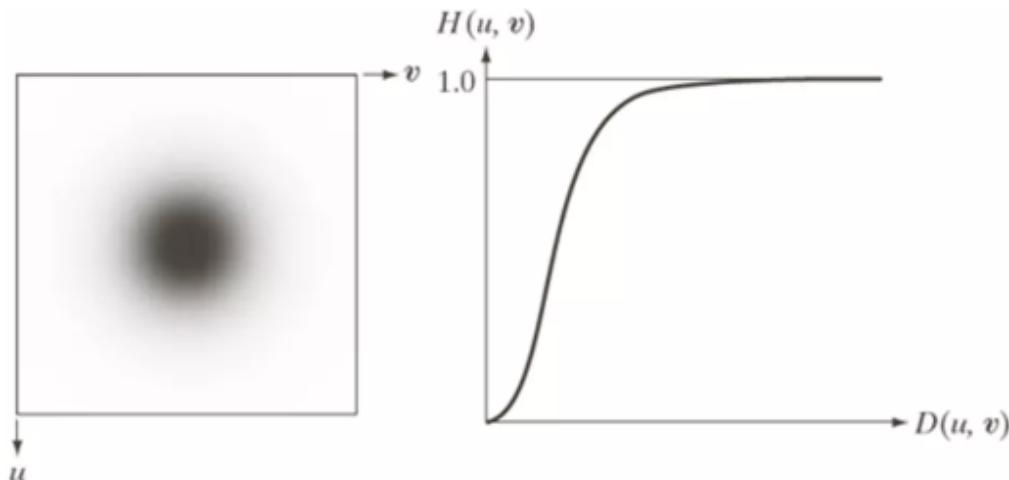
D0 =35



Lọc thông cao Butterworth

Cố định tần số cắt

$$D_0 = 20$$



Original



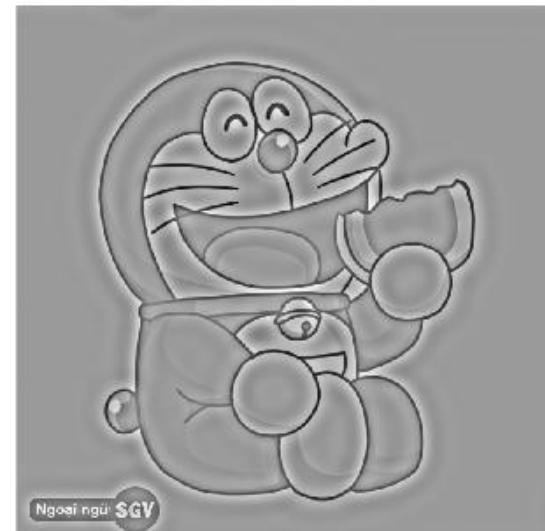
$n = 1$



$n = 3$

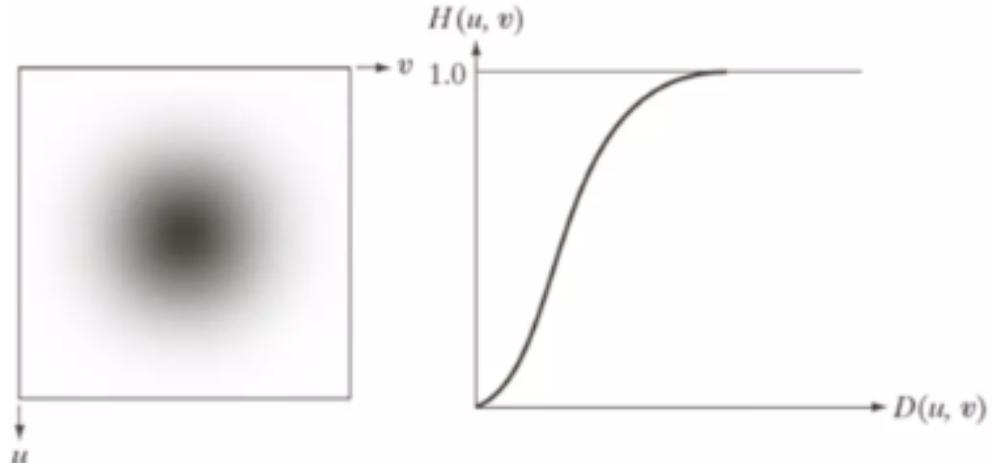


$n = 5$



Lọc thông cao Gaussian

- Hàm chuyển trong miền tần số là hàm dạng Gaussian



original image



D0 =5



D0 =20



D0 =35

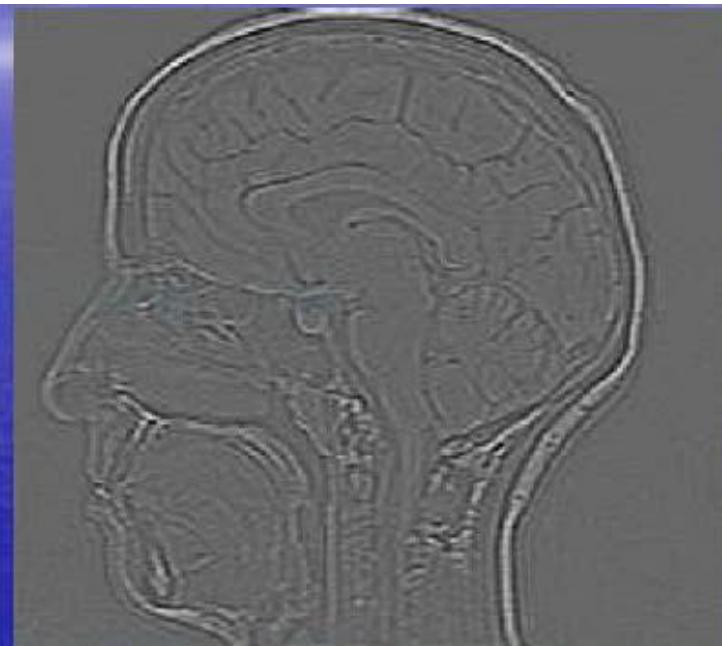


Lọc thông cao

- Lọc thông cao tương đương với lọc sắc nét trong miền không gian



Ảnh ban đầu



Lọc băng thông cao

Tóm lại

- Lọc thông cao cho tần số cao đi qua, tương tự với lọc sắc nét với toán tử Laplacian trong miền không gian.
- Làm nổi bật các cạnh các vật thể trong ảnh

Bộ lọc	Công thức
Ideal high filter (IHPF)	$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$
Butterworth highpass filter (BHPF)	$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$
Gaussian highpass filter (GHPF)	$H(u, v) = 1 - e^{-\frac{D(u,v)^2}{2D_0^2}}$

* D_0 : cut-off frequency, n : bậc của bộ lọc Butterworth



Khôi phục ảnh

Toán tử trên miền
không gian

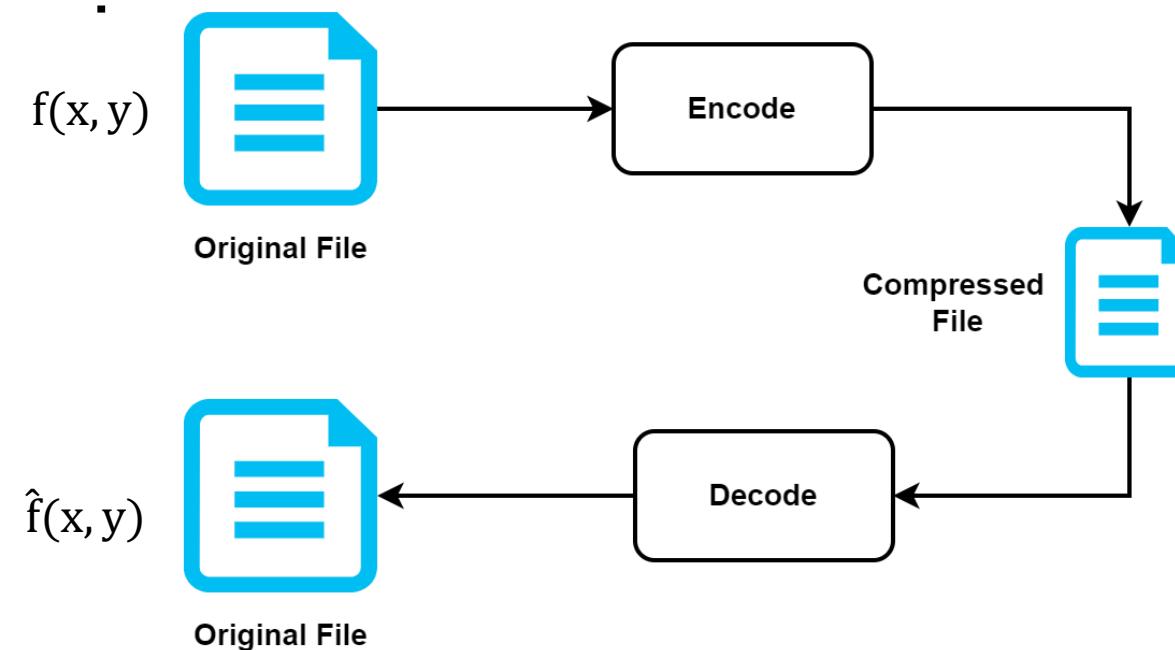
Toán tử trên miền
tần số

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

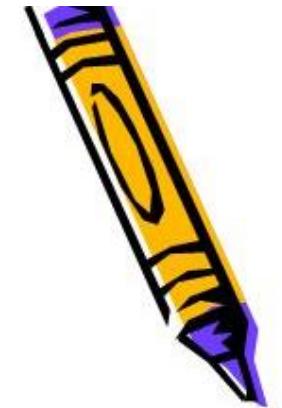
- Lọc thông thấp
- Lọc thông cao
- Lọc high-boost

Chương 4. Nén ảnh và phân đoạn ảnh

- ☐ Ảnh số cần nhiều không gian để lưu trữ
- ☐ Cần các kỹ thuật để giảm số lượng dữ liệu cần thiết để biểu diễn ảnh → giảm không gian cần lưu trữ và tăng tốc độ truyền dữ liệu

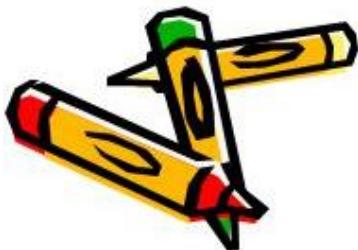


Data vs. Information



Data

- raw facts
- no context
- just numbers and text

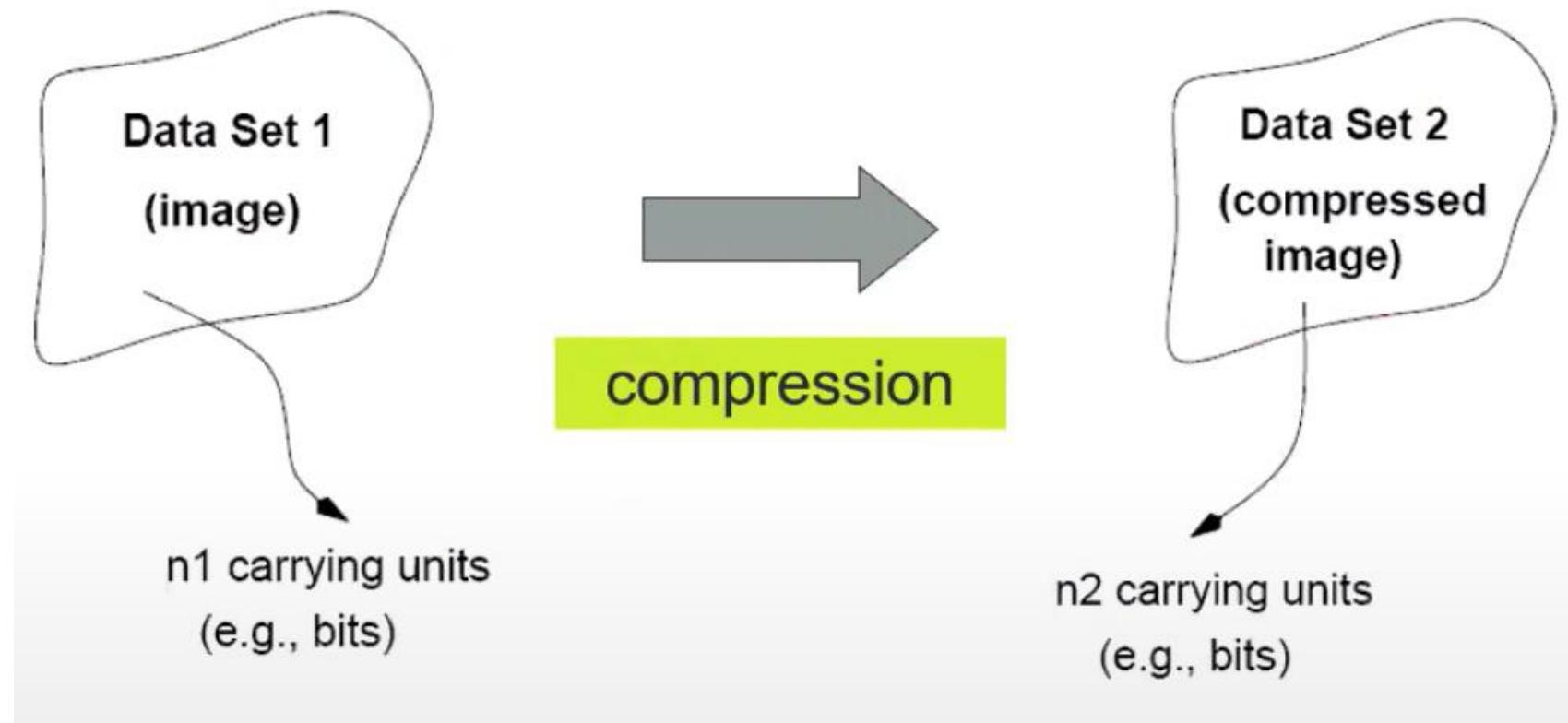


Information

- data with context
- processed data
- value-added to data
 - summarized
 - organized
 - analyzed

Nén dữ liệu: giảm lượng data trong khi giữ lại nhiều lượng thông tin nhất có thể

Tỉ lệ nén



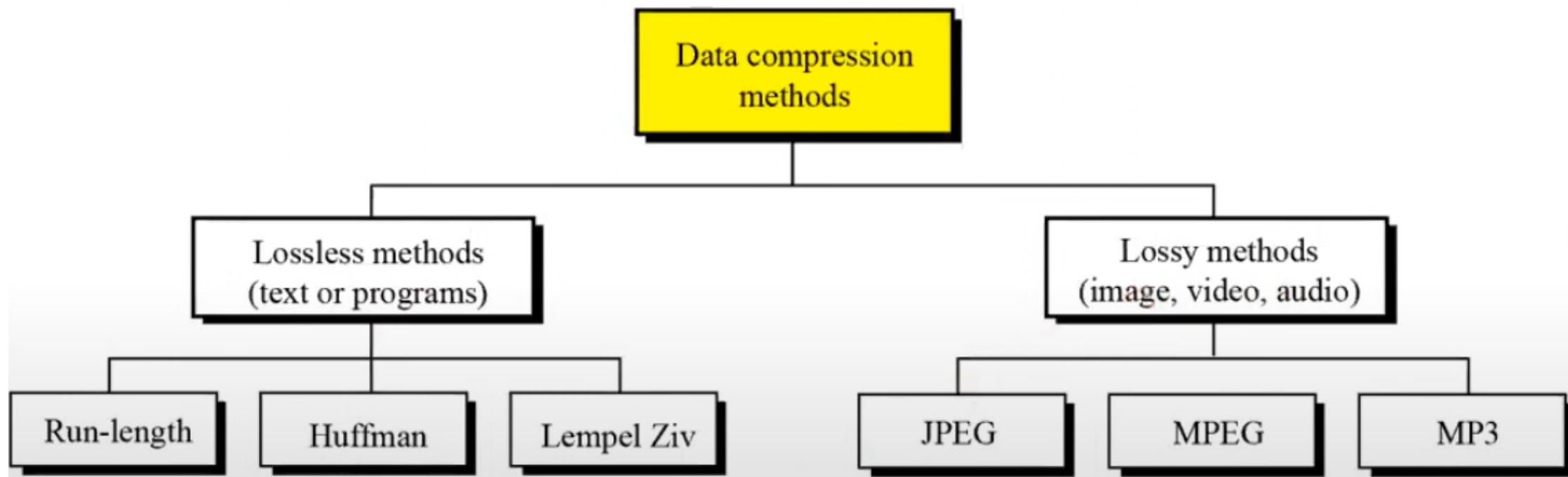
$$\text{Tỉ lệ nén: } C_R = \frac{n_1}{n_2}$$



Các loại nén

- Nén không mất thông tin (lossless)
 - Thông tin được bảo toàn
 - Tỉ lệ nén thấp
- Nén mất thông tin (lossy)
 - Thông tin không được bảo toàn – loại bỏ các dữ liệu dư thừa (redundant data)
 - Tỉ lệ nén cao

Các phương pháp nén dữ liệu





Nén dữ liệu không tổn thất

- Run-length coding (RLC)
- Huffman coding
- Lempel-Ziv



Thuyết trình

- Run-length coding (RLC)
- Huffman coding
- Lempel-Ziv

Giải thích ý nghĩa, ưu/khuyết điểm nếu có, giải tay, demo, tạo GUI