



# Xử lý ảnh – Image processing

## Chương 4: Lọc ảnh trong miền tần số (frequency domain based filters)

# Nội dung

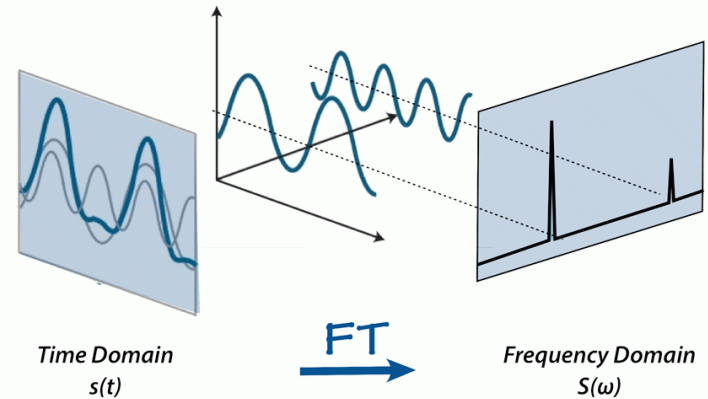
- ▶ Tổng quan
- ▶ Biến đổi Fourier
- ▶ Các bước lọc ảnh trong miền tần số
- ▶ Các bộ lọc thông thấp
  - ▶ Lọc trung bình
  - ▶ Lọc trung vị
  - ▶ Lọc Gauss
- ▶ Các bộ lọc thông cao
  - ▶ Làm sắc nét tuyến tính
  - ▶ Một số nhân lọc chuyên dụng
  - ▶ Đạo hàm của ảnh

# Tổng quan

- ▶ Có rất nhiều các nhiễu trong ảnh mà ta không thể lọc được trong miền không gian. Đối với các nhân có kích thước lớn => việc lọc trong miền không gian sẽ mất thời gian trong khi để đạt được cùng mục đích, một phép lọc trong miền tần số sẽ nhanh hơn rất nhiều (biến đổi Fourier rời rạc). Biến đổi Fourier cung cấp một công cụ để phân tích hình ảnh hiệu quả vì nó cho phép trích xuất được các đặc điểm mà mắt người và các công cụ khác không thực hiện được (motion magnification, sound from motion ...).
- ▶ Việc lọc ảnh trong miền tần số khác với lọc ảnh trong miền không gian (tường minh hơn): bức ảnh ban đầu được biểu diễn trong miền tần số, sau đó sẽ được biểu diễn trong miền tần số thông qua biến đổi Fourier xuôi, sau đó phép lọc trong miền tần số sẽ được thực hiện, cuối cùng là biến đổi Fourier ngược sẽ được thực hiện để chuyển kết quả phép lọc trong miền tần số thành ảnh kết quả trong miền không gian.

# Biến đổi Fourier (Fourier Transformation)

- ▶ Biến đổi Fourier dựa trên lý thuyết Fourier về biểu diễn tín hiệu: tất cả các tín hiệu tuần hoàn nào cũng có thể biểu diễn dưới dạng tổng có trọng số của các hàm sin/cosin có tần số (frequency) và biên độ (amplitude) (chuỗi Fourier).
- ▶ Fourier là một nhà khoa học người Pháp, lý thuyết biến đổi Fourier được nghĩ ra khi ông còn rất trẻ - 1 sinh viên cao học, ban đầu được viết dưới dạng 1 bài luận ngắn - memoir - khó hiểu tới nỗi 2 người thầy rất nổi tiếng thế giới là Cô si và Laplace không hiểu được. Sau khi hiểu ra thì hai người này đều thống nhất là không công bố sớm vì nó quá quan trọng cho sự phát triển của Khoa học kỹ thuật nói chung.

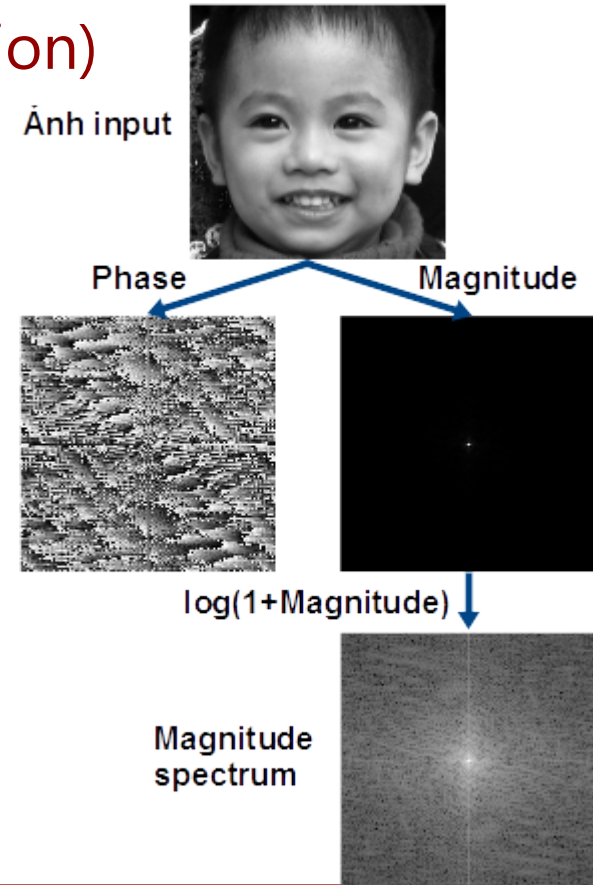


## Biến đổi Fourier (Fourier Transformation)

- ▶ Trong biến đổi Fourier gốc, tín hiệu là liên tục, còn đối với ảnh số thì tín hiệu là rời rạc và ta có biến đổi Fourier rời rạc (Fourier Transform – FT là sự tổng quát của Discrete Fourier Transform – DFT). Một DFT của một tín hiệu rời rạc sẽ gồm 2 phần: phần cường độ (Magnitude) và thành phần pha (góc) gọi là Phase.
- ▶ Một biến đổi Fourier (thuận – Forward) sẽ thực hiện chuyển đổi từ một tín hiệu ở miền không gian sang miền tần số gồm có 2 thành phần: cường độ (Magnitude) và góc/pha (Phase). Trong đó thành phần Phase là quan trọng hơn (nó chứa nhiều thông tin quan trọng hơn). Biến đổi Fourier ngược (Inverse) sẽ thực hiện chuyển từ một biểu diễn trong miền tần số (Magnitude và Phase) thành một tín hiệu trong miền không gian. Nếu FT (Fourier Transform) áp dụng cho các tín hiệu rời rạc thì ta sẽ biến đổi FT rời rạc (Discrete FT – DFT).

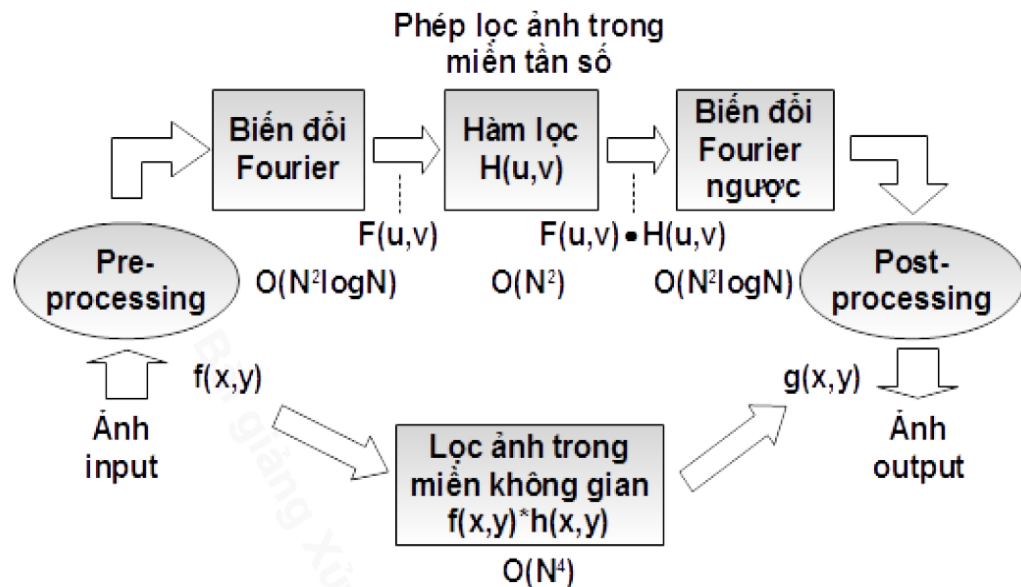
# Biến đổi Fourier (Fourier Transformation)

- ▶ Trong số các cài đặt của FT, cài đặt tối ưu (áp dụng khi kích thước dữ liệu là  $N = 2^k$ ) sẽ có độ phức tạp là  $O(N) = N \cdot \log(N)$  và gọi là Fast FT (FFT). Các công cụ xử lý tín hiệu và ảnh đều cài đặt các hàm thực hiện các biến đổi Fourier, Matlab (fft và ifft), OpenCV (dft và idft), FFTW, MKL của Intel.
- ▶ Thao tác thực hiện biến đổi Fourier đã được trang bị sẵn trong các công cụ lập trình (OpenCV, Matlab, GPU, MKL của Intel, FFTW) cài đặt thuật toán biến đổi Fourier nhanh (FFT –  $O(N) = N \log(N)$ ).



# Các bước của 1 bộ lọc trong miền tần số

- ▶ Tiền xử lý (preprocessing): lọc nhiễu
- ▶ Áp dụng DFT để biểu diễn ảnh trong miền tần số
- ▶ Thực hiện phép nhân từng phần tử của biểu diễn trong miền tần số với 1 nhân có kích thước bằng ảnh gốc.
- ▶ Áp dụng DFT ngược để nhận được 1 ảnh trong miền không gian.
- ▶ Bước hậu xử lý (postprocessing).



## Các bước của 1 bộ lọc trong miền tần số

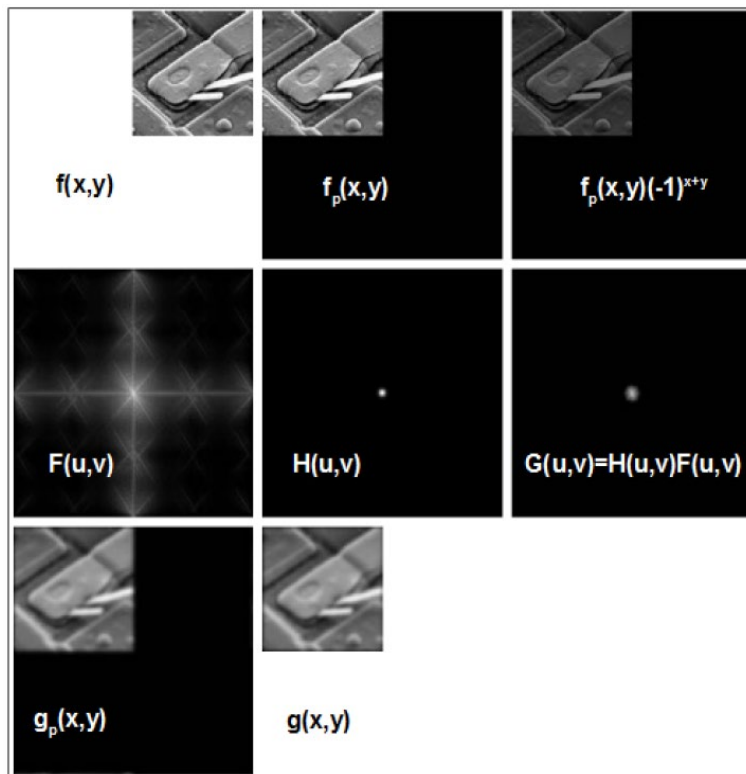
- ▶ Các bước thực hiện của một bộ lọc trong miền tần số:
- ▶ Input: ảnh đầu vào  $f(x, y)$  kích thước  $M \times N$  và  $H(u, v)$  là một bộ lọc trong miền tần số.
- ▶ Output: ảnh  $g(x, y)$  với chất lượng ảnh đã được tăng cường.
- ▶ Bước 1: Chọn  $P = 2M$ ,  $Q = 2N$ .
- ▶ Bước 2: Sinh một ảnh  $f_p(x, y)$  có kích thước là  $P \times Q$  bằng cách thêm các số 0 vào ảnh gốc (nhằm bên trái phía trên của  $f_p(x, y)$ ).
- ▶ Bước 3: Nhân  $f_p(x, y)$  với các giá trị  $(-1)^{x+y}$  để đảm bảo tính đối xứng của biến đổi DFT ở bước sau.



## Các bước của 1 bộ lọc trong miền tần số

- ▶ Bước 4: Tính  $F(u, v)$  là kết quả của biến đổi DFT với ảnh ở bước 3:
- ▶  $F(u, v) = R(u, v) + j I(u, v)$ .
- ▶  $R(u, v)$  là thành phần thực còn  $I(u, v)$  là thành phần ảo trong biến đổi DFT từ  $f_p(x, y)$ .
- ▶ Bước 5: Sinh một bộ lọc đối xứng  $H(u, v)$  trong miền tần số có kích thước là  $P \times Q$ , đối xứng tại tâm  $(P/2, Q/2)$ . Tính tích  $G(u, v)$ :
- ▶  $G(u, v) = H(u, v)F(u, v) = H(u, v)R(u, v) + j H(u, v)I(u, v)$
- ▶ Bước 6: Thực hiện biến đổi Fourier ngược và lấy phần thực theo công thức sau:
- ▶  $g_p(x, y) = \{\text{real}[\text{IDFT}[G(u, v)]]\}(-1)^{x+y}$
- ▶ Bước 7: lấy ảnh kết quả cuối cùng là phần phía trên bên trái có kích thước  $M \times N$  của  $g_p(x, y)$ .

## Các bước của 1 bộ lọc trong miền tần số

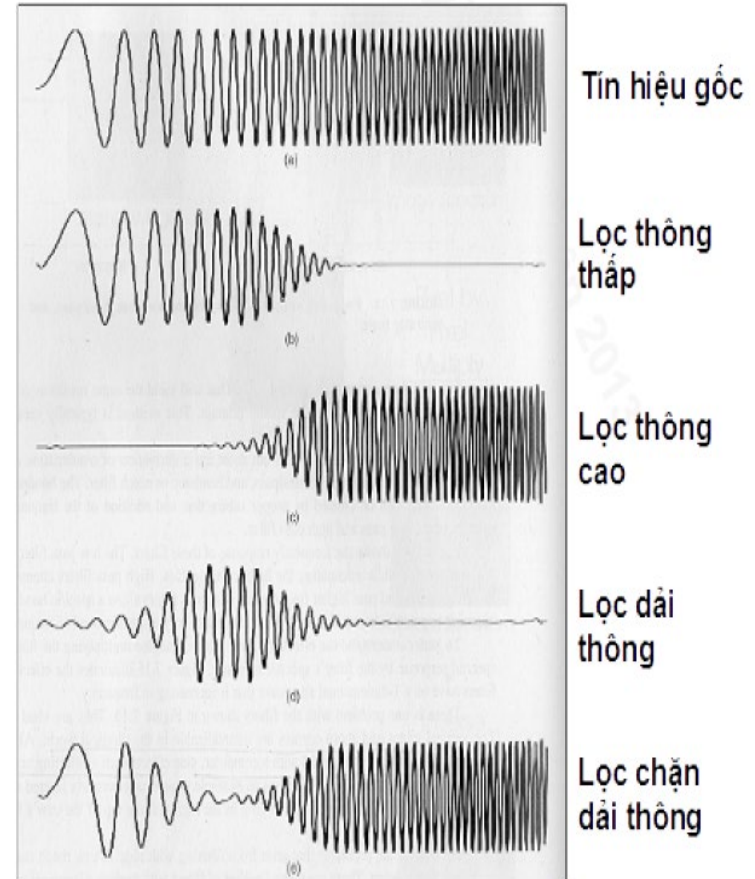


## Sinh bộ lọc $H(u, v)$

- ▶ Bộ lọc  $H(u, v)$  sẽ được sinh từ bộ lọc không gian  $h(x, y)$  như sau:
- ▶ Bước 1: Sinh  $h_p(x, y)$  bằng cách thêm vào các số 0.
- ▶ Bước 2: Nhân  $h_p(x, y)$  với  $(-1)^{x+y}$ .
- ▶ Bước 3: Tính DFT của kết quả thu được.

# Các phép lọc tần số phổ biến

- ▶ Lọc thông thấp (lowpass filter): chỉ cho đi qua các tín hiệu tần số thấp
- ▶ Lọc thông cao (highpass filter): chỉ cho đi qua các tín hiệu có tần số cao.
- ▶ Lọc dải thông (bandpass filter): chỉ cho đi qua các tín hiệu trong một khoảng cho trước.
- ▶ Lọc chặn dải thông (band reject filter): chặn các tín hiệu trong một khoảng cho trước.



## Các bộ lọc thông thấp (Lowpass filter)

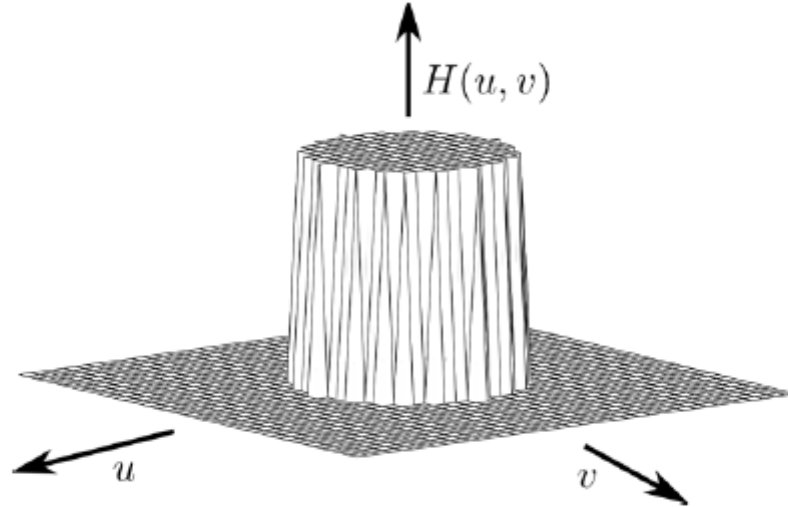
- ▶ Các bộ lọc thông thấp chỉ cho qua các tần số thấp và sẽ làm mờ ảnh, tác dụng là loại bỏ nhiễu. Có 3 bộ lọc thông thấp: Ideal, Butterworth, Gaussian.
- ▶ Bộ lọc Ideal Lowpass filter: hàm lọc  $H(u, v)$  được tính như sau:

$$H_{LP}(u, v) = \begin{cases} 1 & \text{nếu } D(u, v) \leq D_0 \\ 0 & \text{nếu } D(u, v) > D_0 \end{cases} \text{ trong đó } D(u, v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

- ▶ Với  $D_0$  là giá trị cắt tần số (cut-off frequency) hoặc bán kính cắt tần số. Đối với bộ lọc thông thấp, giá trị này càng giảm thì các tần số cao sẽ bị cắt càng nhiều.

## Lọc thông thấp lý tưởng ILF (Ideal Lowpass filter)

- ▶ Các tần số bị loại bỏ nằm trong một hình tròn nên mỗi bộ lọc thông thấp



Hình 4.20: Tần số bị loại bỏ trong lọc thông thấp ILPF

## Lọc thông thấp lý tưởng ILF (Ideal Lowpass filter)

- Để tạo bộ lọc thông thấp cho một ảnh có kích thước rows x cols ta sử dụng đoạn mã sau:

```
int m = getOptimalDFTSize(rows );
int n = getOptimalDFTSize(cols); // on the border add zero
values
Point centre = Point(m / 2, n / 2);
Mat tmp = Mat::zeros(m, n, CV_32F);
circle(tmp, cvPoint(tmp.cols / 2, tmp.rows / 2), D0,
cvScalarAll(1.0), -1 , 0 ,0);
```

# Lọc thông thấp lý tưởng ILF (Ideal Lowpass filter)

```
Mat DFTFilter(const Mat & inputIm, const Mat & dft_Filter)
{
    Mat padded, result; //expand input image to optimal size
    // create 2D filter
    Mat filter;
    Mat toMerge[] = {dft_Filter, dft_Filter};
    merge(toMerge, 2, filter);
    int m = getOptimalDFTSize( inputIm.rows );
    int n = getOptimalDFTSize( inputIm.cols );
    // on the border add zero values
    copyMakeBorder(inputIm, padded, 0, m - inputIm.rows, 0, n - inputIm.cols, BORDER_CONSTANT,
Scalar::all(0));
    Mat planes[] = {Mat_<float>(padded), Mat::zeros(padded.size(), CV_32F)};
    Mat complexI;
    merge(planes, 2, complexI); // Add to the expanded another plane with zeros
    dft(complexI, complexI); // this way the result may fit in the source matrix

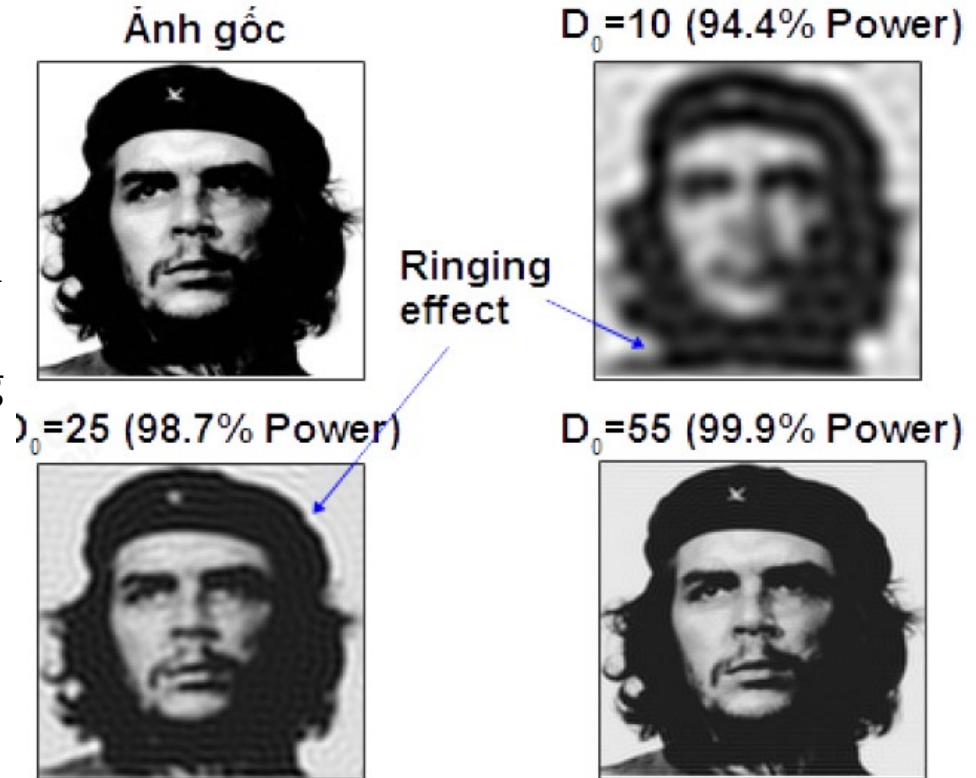
    // apply filter
    FFT_0::shiftDFT(complexI);
    mulSpectrums(complexI, filter, complexI, 0);
    FFT_0::shiftDFT(complexI);
    // do inverse DFT on filtered image
    idft(complexI, complexI, DFT_SCALE);
    //idft(complexI, complexI);
    // split into planes and extract plane 0 as output image
    split(complexI, planes);
    //result = result(Rect(0, 0, result.cols & -2, result.rows & -2));
    result = planes[0](Rect(0, 0, inputIm.cols, inputIm.rows));
    normalize(result, result, 0, 1, CV_MINMAX);

    return result;
}
```



## Lọc thông thấp lý tưởng ILF (Ideal Lowpass filter)

- ▶ Hiện tượng ringing effect (hiệu ứng vòng)
- ▶ Là hiện tượng xuất hiện trong ảnh kết quả của phép lọc khi các tín hiệu gần với các tần số bị loại bỏ xuất hiện. Hiện tượng này là do sự không liên tục của các phần tử của bộ lọc Ideal sinh ra. Để khắc phục hiện tượng này, ta có thể sử dụng các bộ lọc Butterworth và Gaussian.



## Lọc thông thấp Butterworth BLF (Butterworth Lowpass filter)

- ▶ Bộ lọc thông thấp Butterworth: hàm lọc  $H(u, v)$  sẽ được tính như sau:

$$H_{LP}(u, v) = \frac{1}{1 + \left[ \frac{D(u, v)}{D_0} \right]^{2n}} = \frac{1}{1 + \left[ \frac{(u - M/2)^2 + (v - N/2)^2}{D_0^2} \right]^n}$$

- ▶ Với  $D_0$  là bán kính cắt tần số giống như trong bộ lọc Ideal. Công thức trên làm trơn quá trình chuyển đổi giữa các tần số thấp và tần số cao trong quá trình lọc nên giảm được hiện tượng ringing effect. Số mũ  $n$  được gọi là bậc của bộ lọc (thường có giá trị bằng 2).

## Lọc thông thấp Butterworth BLF (Butterworth Lowpass filter)

- ▶ Bộ lọc thông thấp Butterworth: hàm lọc  $H(u, v)$  sẽ được tính như sau:

$$H_{LP}(u, v) = \frac{1}{1 + \left[ \frac{D(u, v)}{D_0} \right]^{2n}} = \frac{1}{1 + \left[ \frac{(u - M/2)^2 + (v - N/2)^2}{D_0^2} \right]^n}$$

- ▶ Với  $D_0$  là bán kính cắt tần số giống như trong bộ lọc Ideal. Công thức trên làm trơn quá trình chuyển đổi giữa các tần số thấp và tần số cao trong quá trình lọc nên giảm được hiện tượng ringing effect. Số mũ  $n$  được gọi là bậc của bộ lọc (thường có giá trị bằng 2).

## Lọc thông thấp Gaussian GLF (Gaussian Lowpass filter)

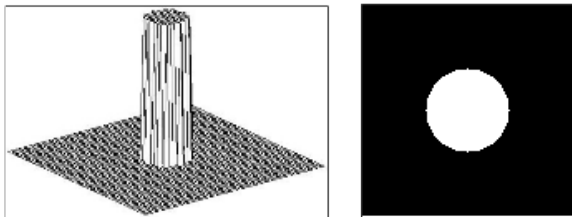
- ▶ Bộ lọc thông thấp Gaussian: hàm lọc  $H(u, v)$  sẽ được tính như sau:

$$H_{LP}(u, v) = e^{-\frac{D(u, v)^2}{2\sigma^2}}, \text{ với } \sigma = D_0 \text{ ta có } H_{LP}(u, v) = e^{-\frac{D(u, v)^2}{2D_0^2}}$$

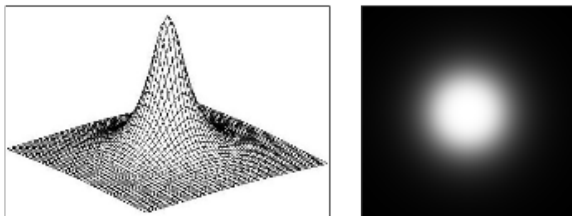
- ▶ So với hai bộ lọc Ideal và Butterworth thì bộ lọc Gaussian triệt để loại bỏ được hiện tượng ringing effect.

# So sánh các nhân lọc thông thấp

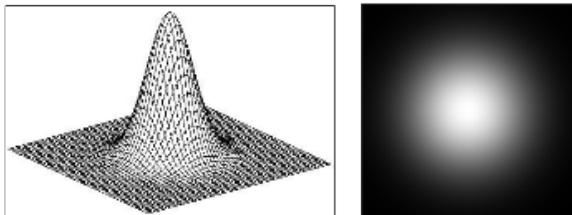
Ideal Lowpass Filter



Butterworth Lowpass Filter



Gaussian Lowpass Filter



# Ứng dụng của Lọc thông thấp

- ▶ Ứng dụng của các bộ lọc thông thấp:
  - ▶ làm mờ ảnh
  - ▶ lọc nhiễu
  - ▶ khôi phục các nét đứt trong ảnh hoặc loại bỏ các chi tiết không cần thiết.
- ▶ Bộ lọc thông thấp Gaussian là hay được dùng nhất.

## Lọc thông cao (Highpass filter)

- ▶ Các tần số cao của ảnh thường tương ứng với các đường (edges) và các chi tiết (fined details) của ảnh, do đó việc lọc thông cao (loại bỏ các chi tiết tần số thấp) sẽ làm cho ảnh trở nên sắc nét hơn và làm giảm độ mờ của ảnh.
- ▶ Các bước lọc không thay đổi so với các bộ lọc thông thấp, chỉ có hàm lọc  $H(u, v)$  là thay đổi:

$$H_{HP}(u, v) = 1 - H_{LP}(u, v).$$

## Lọc thông cao (Highpass filter)

- ▶ Hàm lọc thông cao cho bộ lọc Ideal:

$$H_{HP}(u, v) = \begin{cases} 0 & \text{nếu } D(u, v) \leq D_0 \\ 1 & \text{nếu } D(u, v) > D_0 \end{cases} \text{ trong đó } D(u, v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

- ▶ Hàm lọc thông cao Butterworth:

$$H_{HP}(u, v) = \frac{1}{1 + \left[ \frac{D_0}{D(u, v)} \right]^{2n}} = \frac{1}{1 + \left[ \frac{D_0^2}{(u - M/2)^2 + (v - N/2)^2} \right]^n}$$

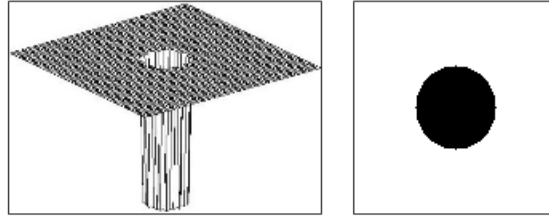
- ▶ Hàm lọc thông cao Gaussian:

$$H_{HP}(u, v) = 1 - e^{-\frac{D(u, v)^2}{2D_0^2}}$$

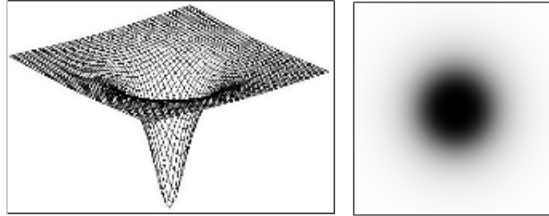


# Nhân lọc thông cao $H_{HP}(u,v)$

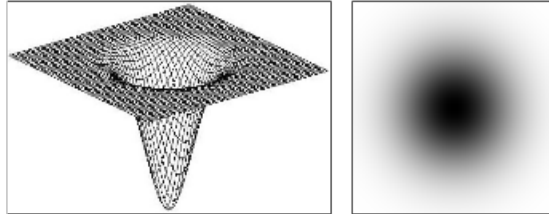
Ideal Highpass Filter



Butterworth Highpass Filter



Gaussian Highpass Filter



## Lọc thông cao $H_{HP}(u,v)$

- ▶ Ứng dụng của lọc thông cao là để làm nổi bật các chi tiết của ảnh, ví dụ như trong các ảnh chụp y học (chụp X-quang).
- ▶ Tương tự như lọc thông thấp, trong lọc thông cao cũng có hiện tượng ringing effect và cách giải quyết tương tự.

# Bài tập

- ▶ Bài 1: Viết chương trình đọc 1 ảnh, thực hiện biến đổi Fourier và hiển thị các thành phần cường độ (Magnitude) và pha (Phase) tương ứng.
- ▶ Bài 2: Viết chương trình đọc 1 file video/camera, thực hiện biến đổi Fourier với từng frame và hiển thị kết quả. Cho nhận xét.
- ▶ Bài 3: Viết chương trình sinh 1 bộ lọc  $H(u, v)$  của các phép lọc tương ứng: Ideal, Butterworth, Gaussian.
- ▶ Bài 4: Viết chương trình thực hiện 1 phép lọc thông thấp với 1 ảnh.
- ▶ Bài 5: Viết chương trình thực hiện 1 phép lọc thông cao với 1 ảnh.

# Question?