ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

Image Filter

Lecturer: LeNga

Image filter

Three views of filtering:

- **Image filters in the spatial domain**
  - Filter is a **mathematical operation of a grid of numbers**S
  - moothing, sharpening, measuring texture
- **Image filters in the frequency domain**
  - Filtering is a **way to modify the frequencies of images**
  - Denoising, sampling, image compression
- **Templates and Image Pyramids**
  - Filtering is a **way to match a template to the image**
  - Detection, coarse-to-fine registration

Image filters

in the frequency domain

- Fourier series
- Fourier transform

Image filters
in the frequency domain

- **Fourier series**
- Fourier transform

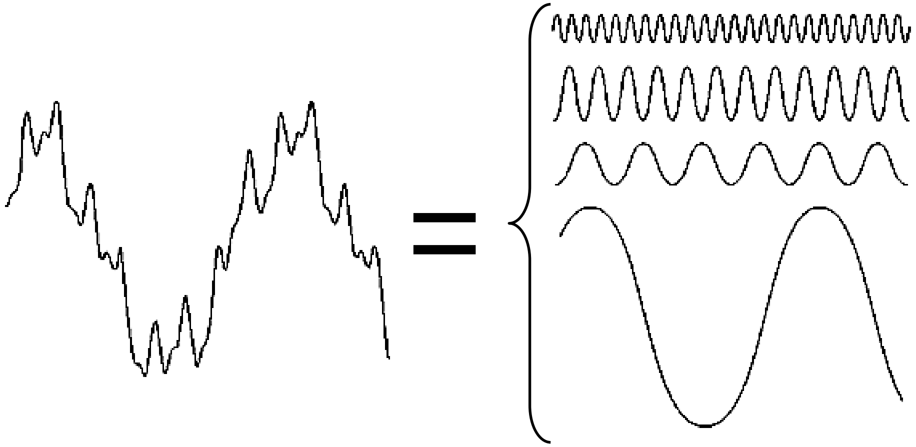
Jean Baptiste Joseph Fourier (1768-1830)

- Had crazy idea (1807):
  - Any periodic function can be rewritten as a weighted sum of sines and cosines of different frequencies.

- **Basic building block:** A sum of sines

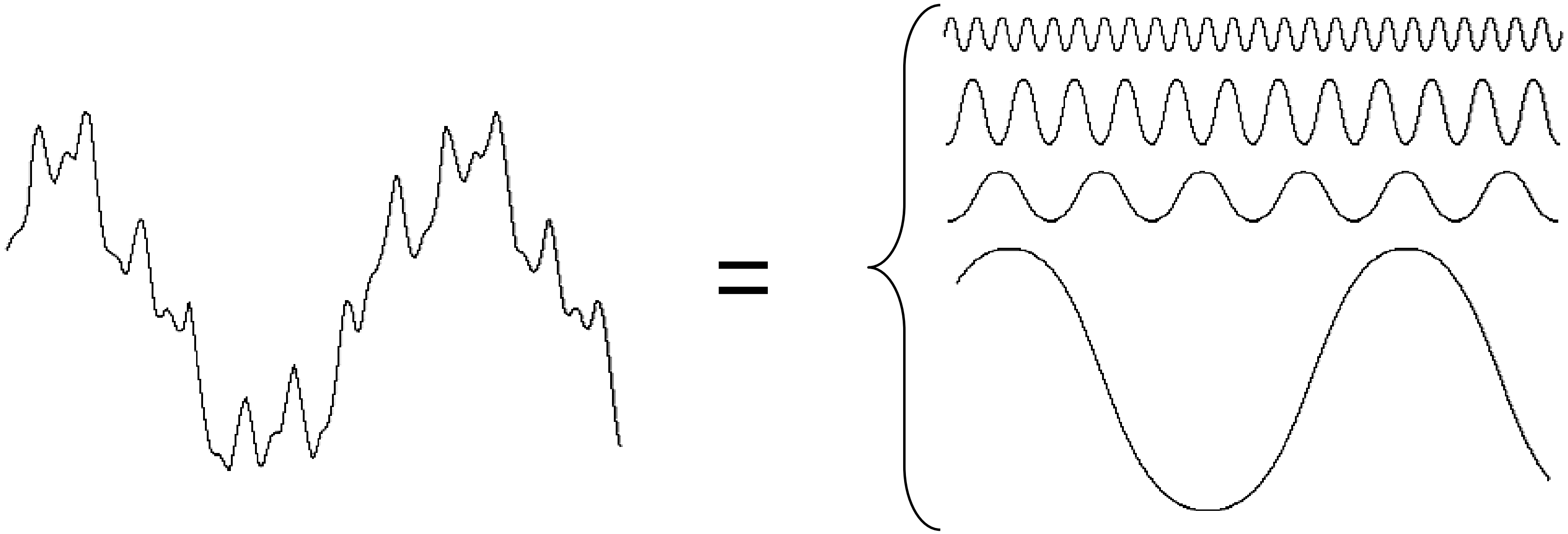
$$A\sin(\omega x+\phi)$$

amplitude
sin
angular frequency
variable
phase



Add enough of them to get any signal *f(x)* you want!

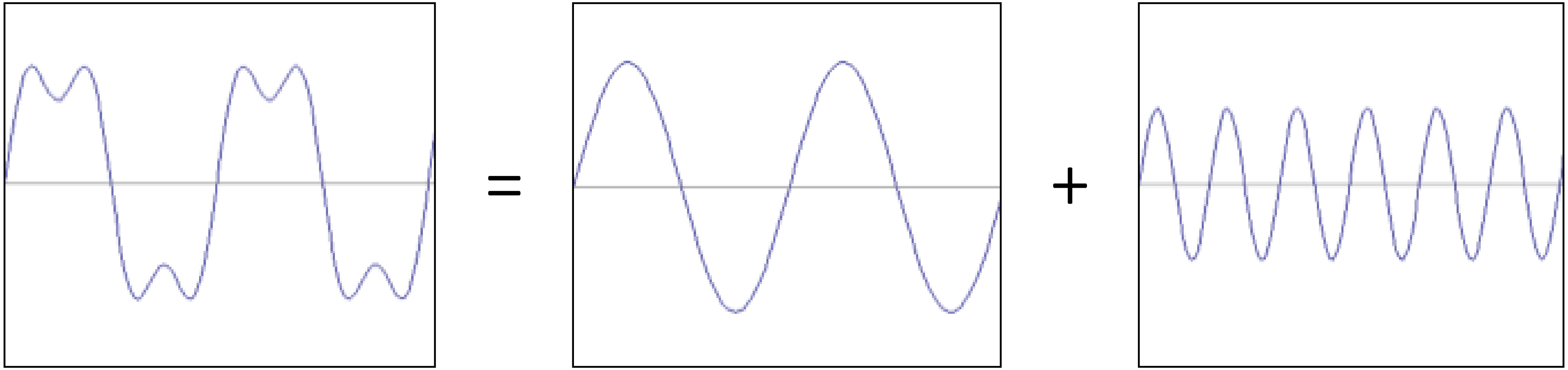
- Một hàm bất kỳ lặp lại có tính chu kỳ có thể biểu diễn dưới dạng tổng các hàm **sine** và **cosine** ở các tần số khác nhau ⇨**chuỗi Fourier**

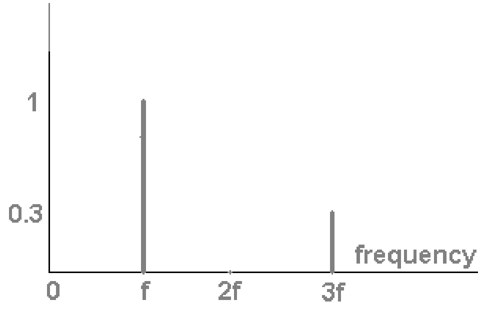


Frequency Spectra

Examples: How would you generate this function?

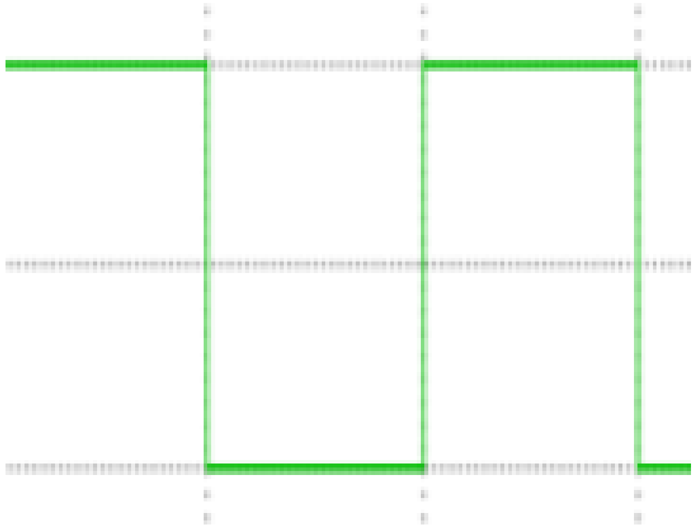
$$f(x) = \sin(2\pi x) + \frac{1}{3}\sin(2\pi 3x) \qquad \sin(2\pi x) \qquad \frac{1}{3}\sin(2\pi 3x)$$





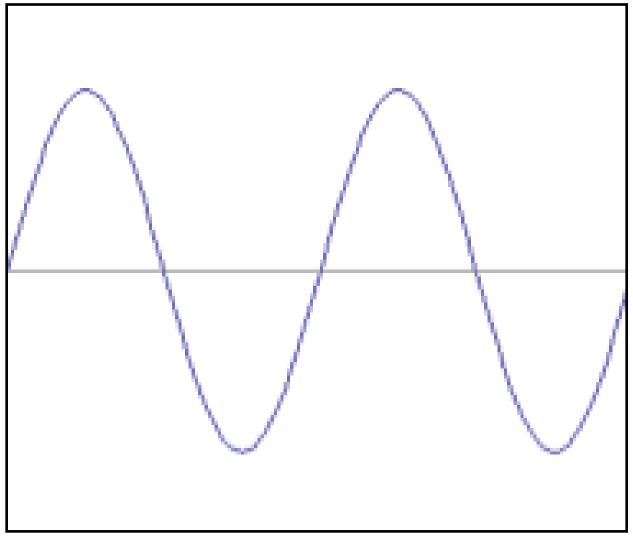
Frequency Spectra

Examples: How would you generate this function?

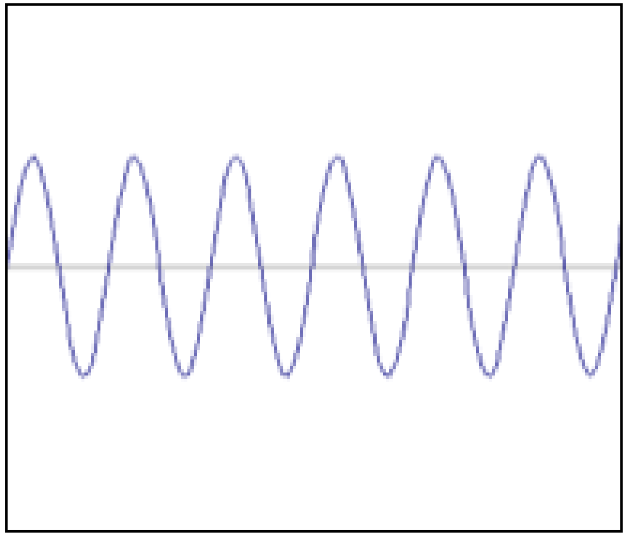


square wave

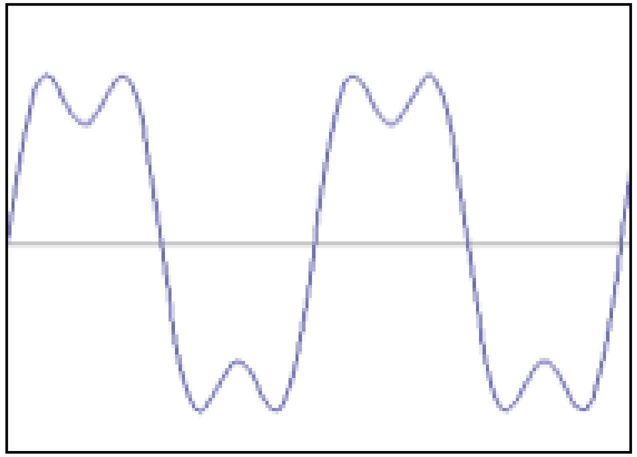
$\approx$



$+$

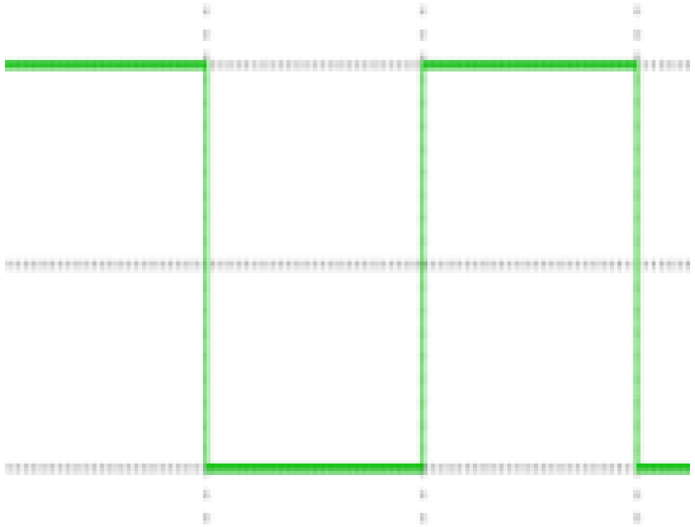


$=$



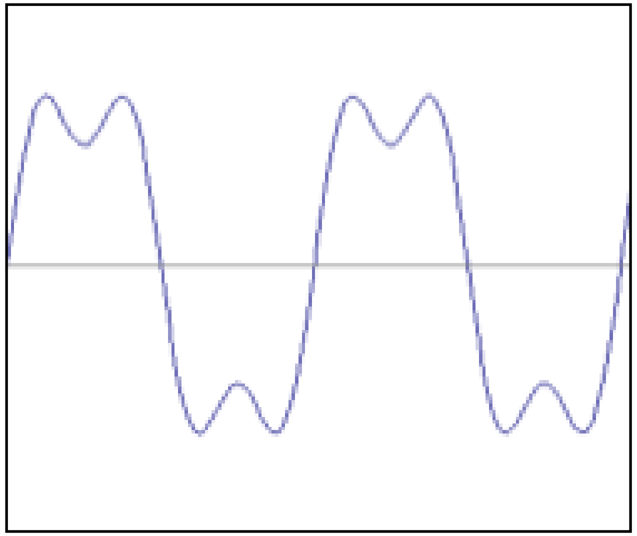
Frequency Spectra

Examples: How would you generate this function?

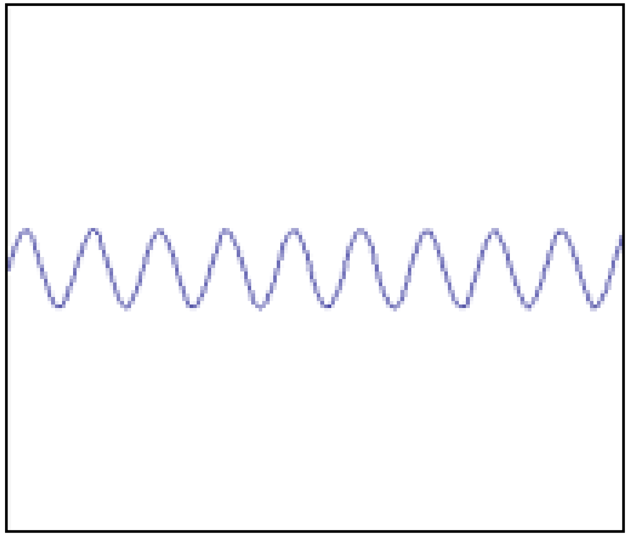


square wave

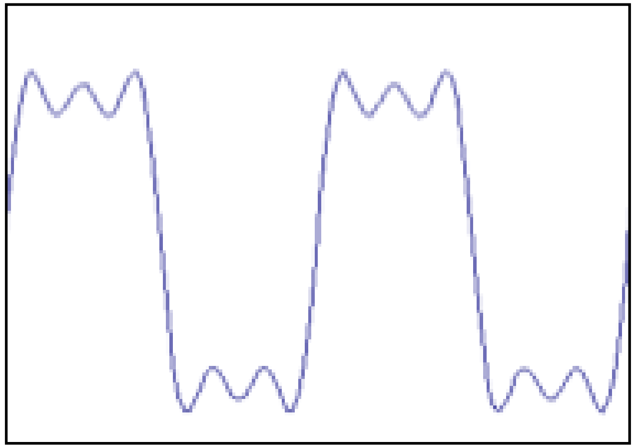
≈



+

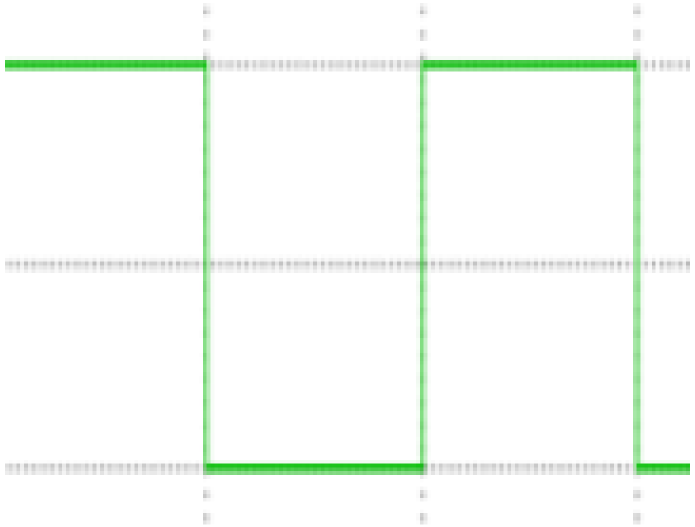


=



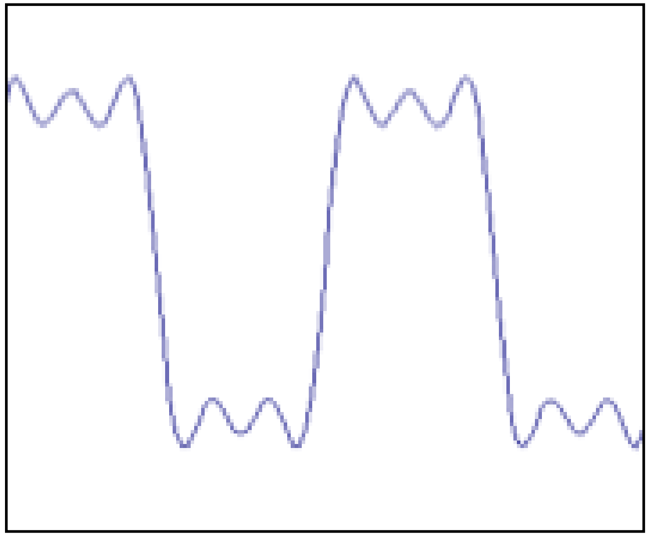
Frequency Spectra

Examples: How would you generate this function?

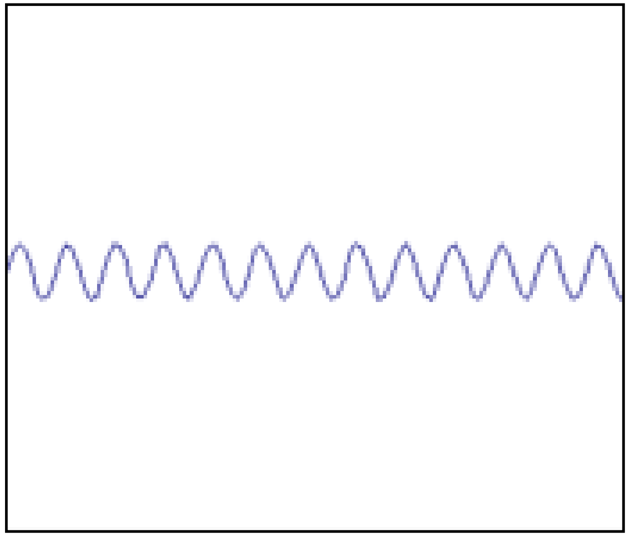


square wave

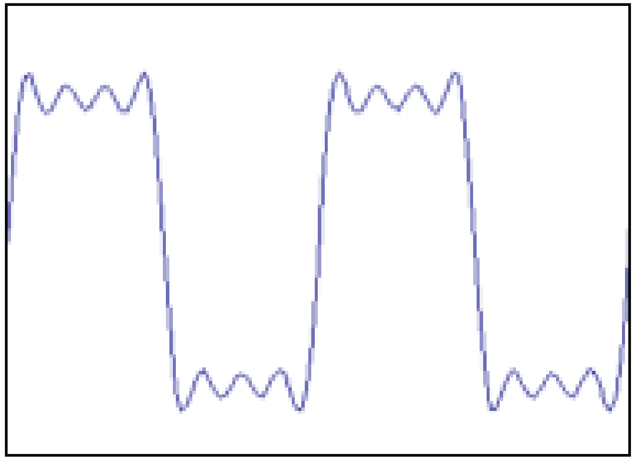
≈



+

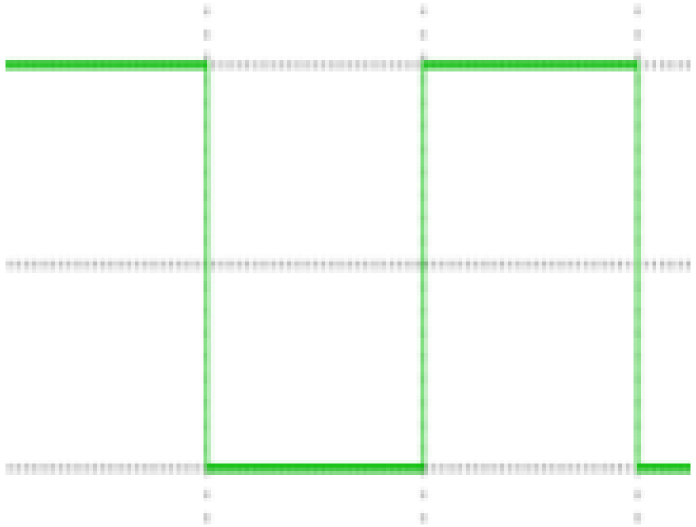


=

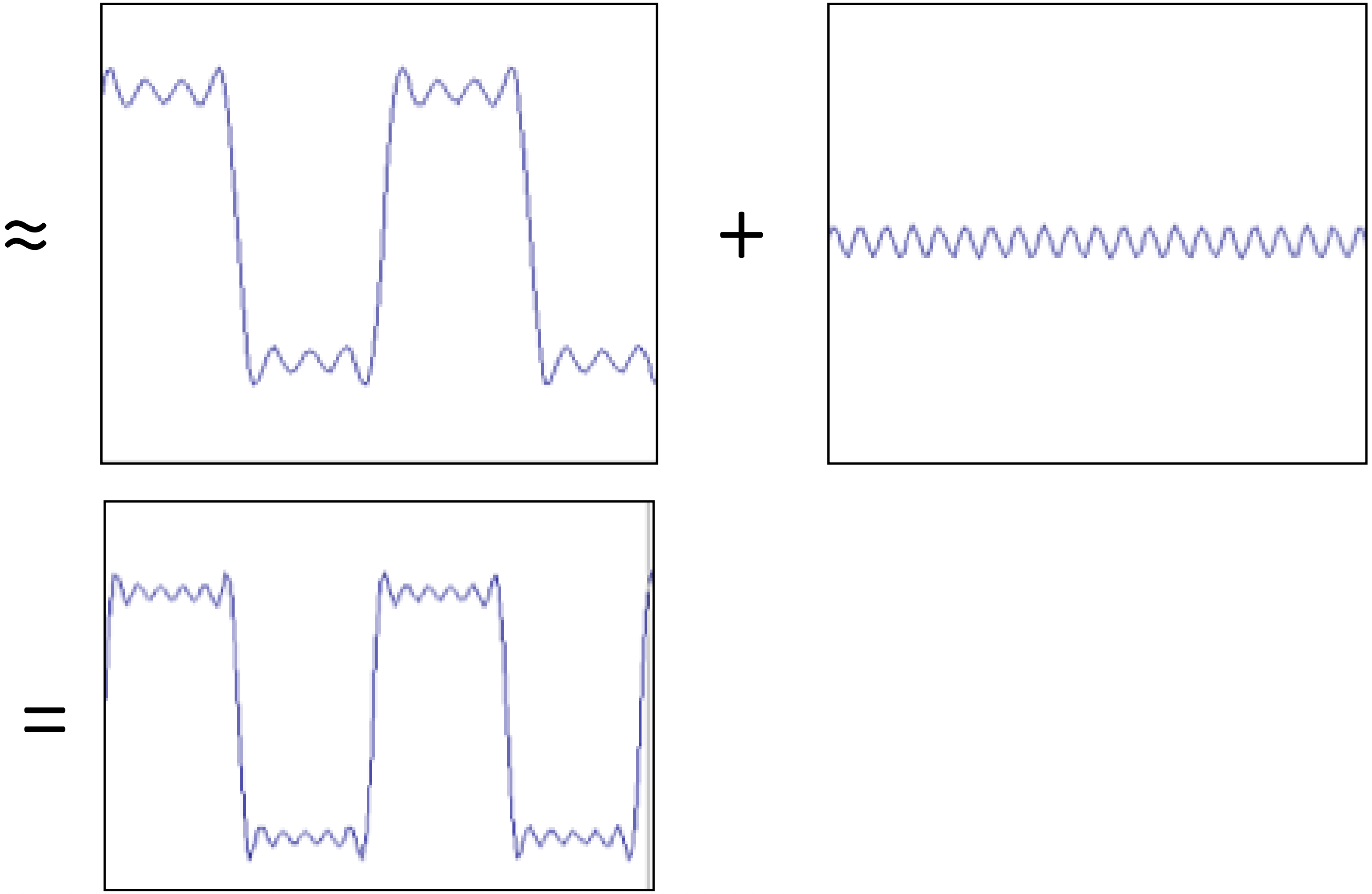


Frequency Spectra

Examples: How would you generate this function?

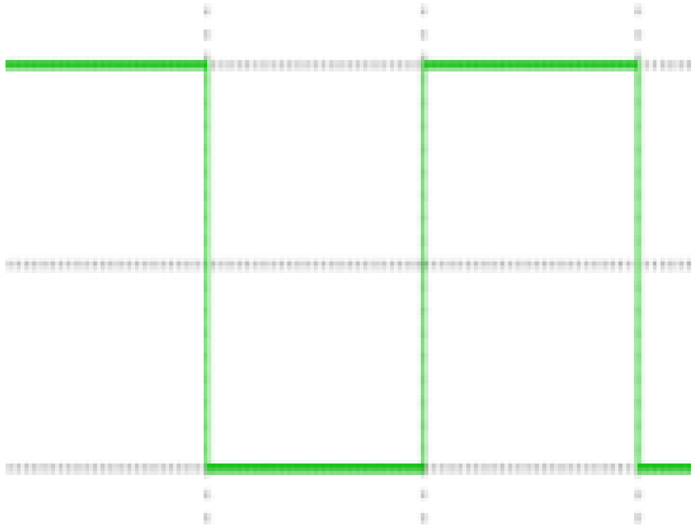


square wave

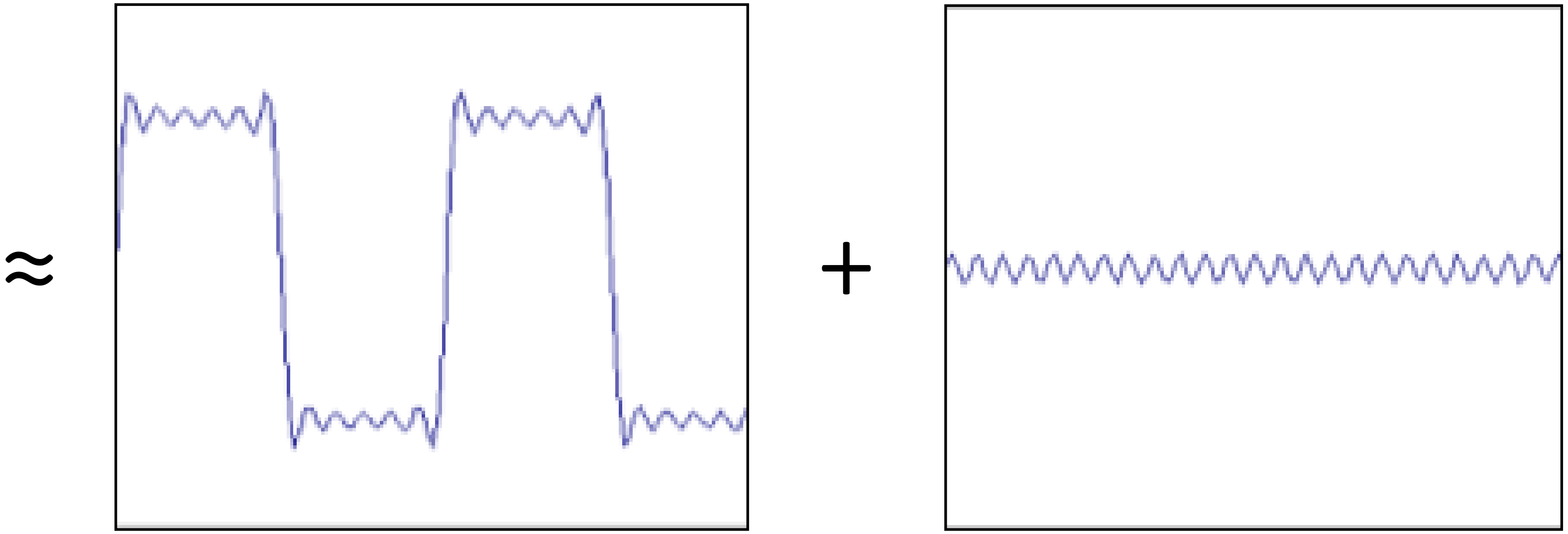


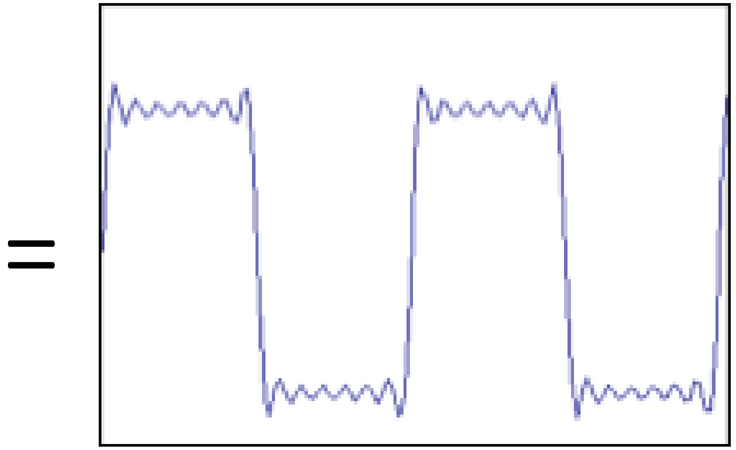
Frequency Spectra

Examples: How would you generate this function?



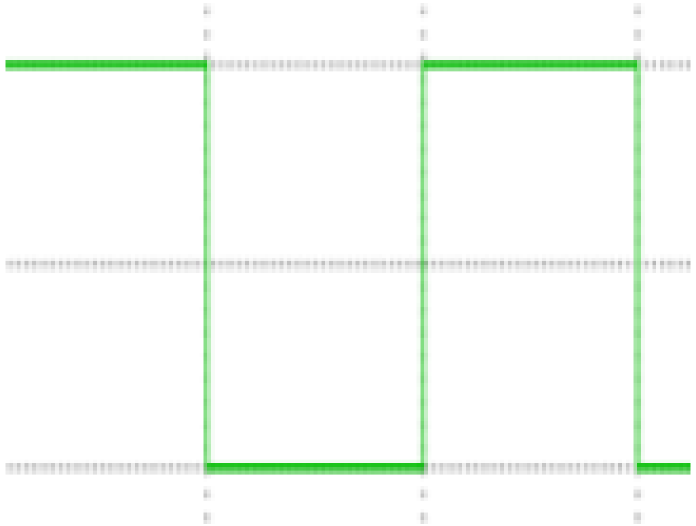
square wave





How would you express this mathematically?

Frequency Spectra



square wave

=

$$A \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi k x)$$

infinite sum of sine waves

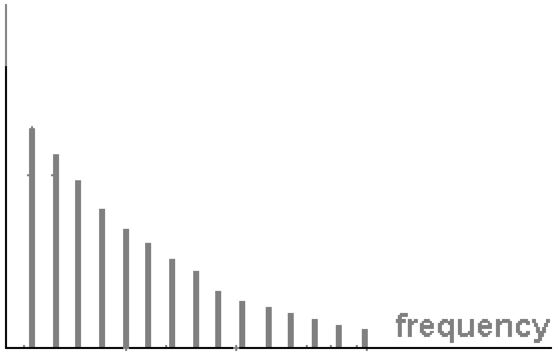


Image filters
in the frequency domain

- Fourier series

- **Fourier transform**

- **Biến đổi Fourier rời rạc** - Discrete Fourier Transform (DFT)
  - Biến đổi DFT của $f(x, y)$, với x = 0, 1, 2…M-1 và y = 0,1,2…N-1, được biểu diễn bởi $F(u, v)$:

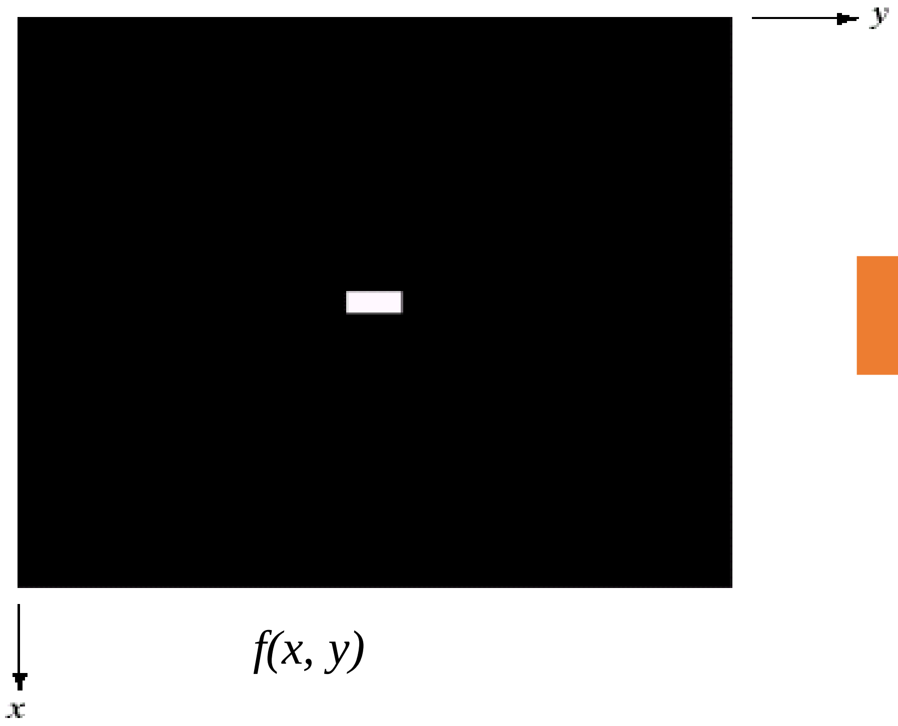
$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

Với u = 0, 1, 2…M-1 and v = 0, 1, 2…N-1.

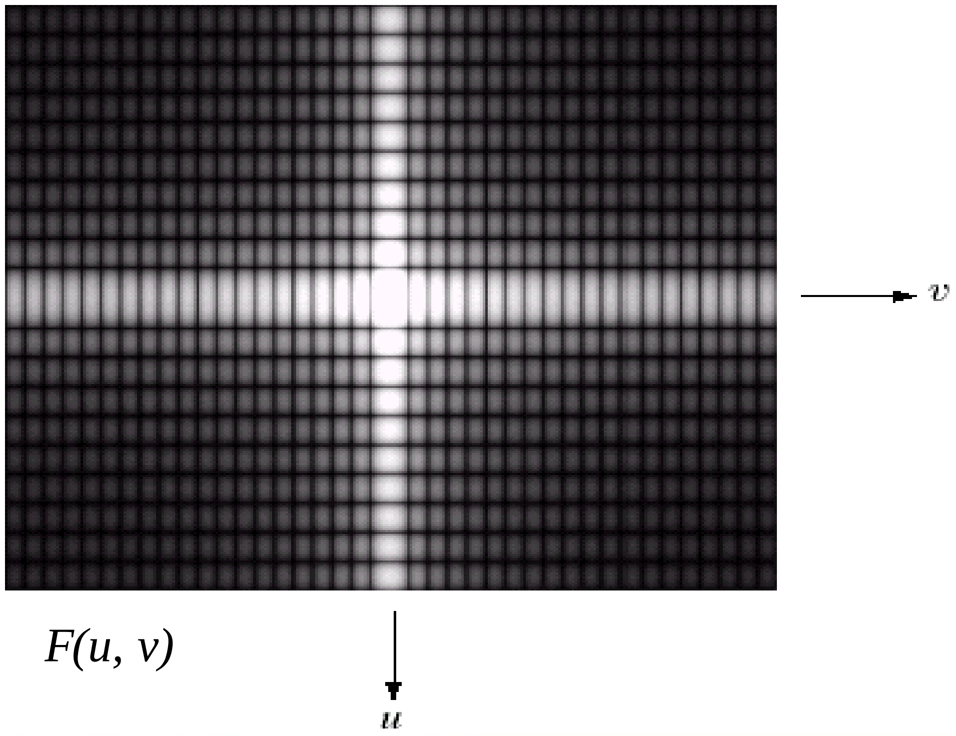
…Fourier transform

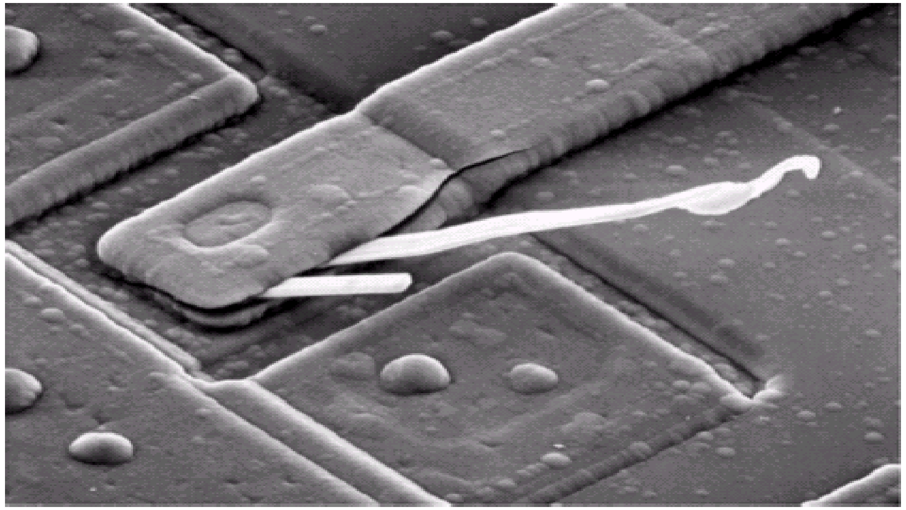
Biến đổi DFT của một ảnh được biểu diễn dưới dạng phổ của thành phần tần số.

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

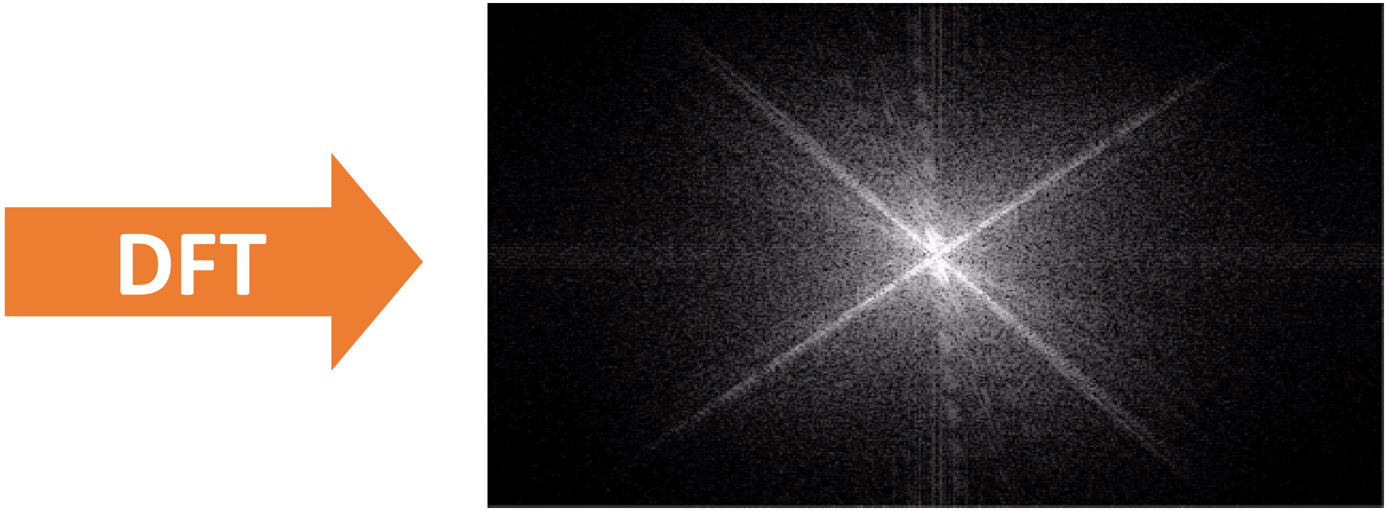


DFT





Ảnh của một mạch điện được phóng đại ~ 2500 lần



Phổ Fourier của ảnh

Biến đổi DFT ngược - Inverse DFT

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)e^{j2\pi(ux/M+vy/N)}$$

Với x = 0, 1, 2…M-1 và y = 0, 1, 2…N-1

The Convolution Theorem

- The Fourier transform of the convolution of two functions is the product of their Fourier transforms

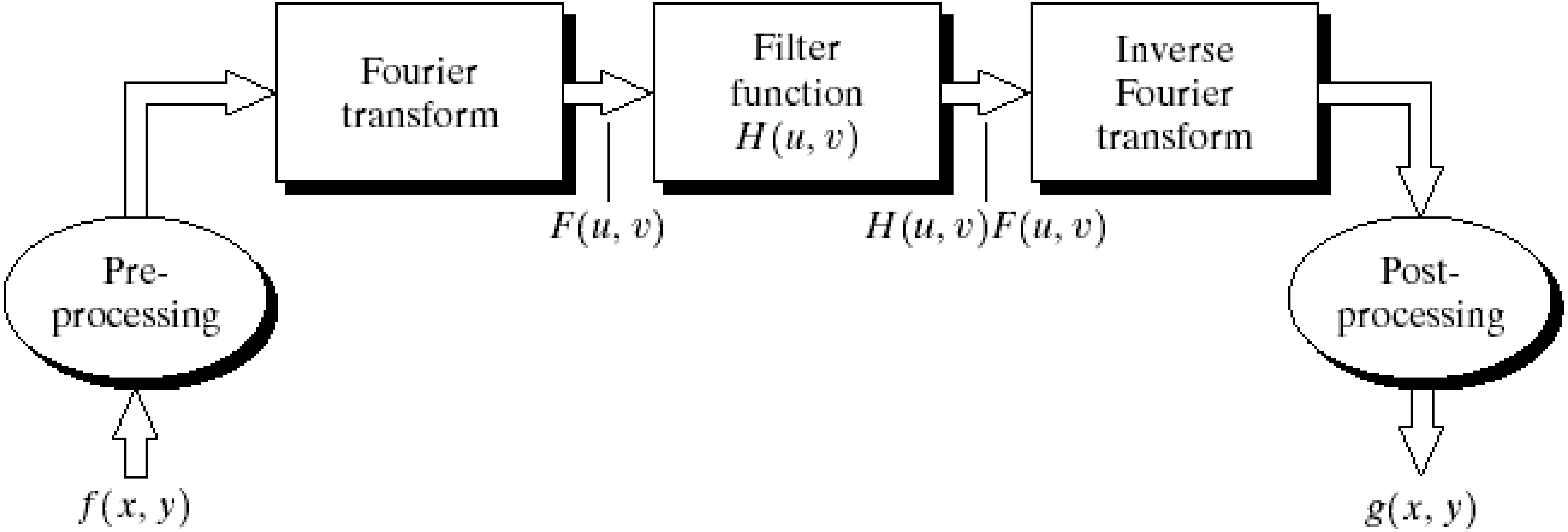
$$\mathrm{F}[g * h] = \mathrm{F}[g]\,\mathrm{F}[h]$$

- The inverse Fourier transform of the product of two Fourier transforms is the convolution of the two inverse Fourier transforms

$$\mathrm{F}^{-1}[gh] = \mathrm{F}^{-1}[g] * \mathrm{F}^{-1}[h]$$

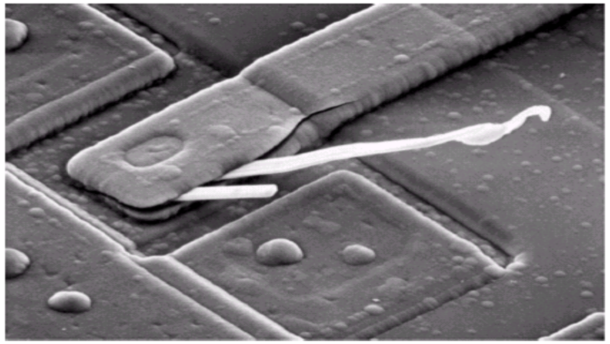
- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!

- **Biến đổi DFT và lọc ảnh trong miền tần số**:
  - Tính DFT $F(u,v)$ của ảnh
  - Nhân $F(u,v)$ với hàm lọc $H(u,v)$
  - Tính DFT ngược của kết quả

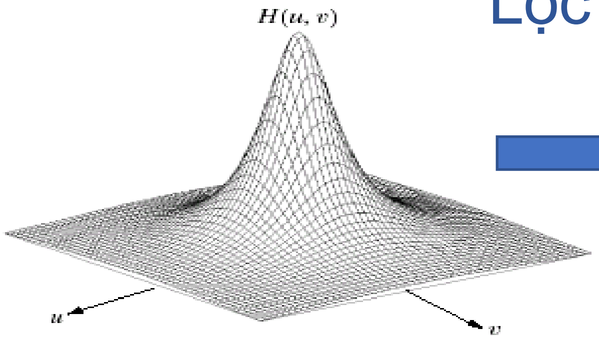


Lọc mịn ảnh (lọc thông thấp)

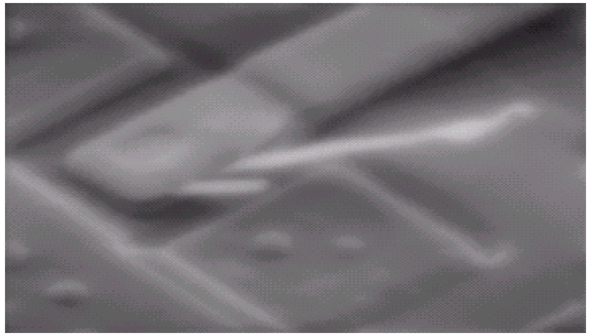
- Lọc ảnh trong miền tần số
  - Lọc thông thấp ⇨ Làm mịn ảnh
  - Lọc thông cao ⇨ Làm sắc nét ảnh

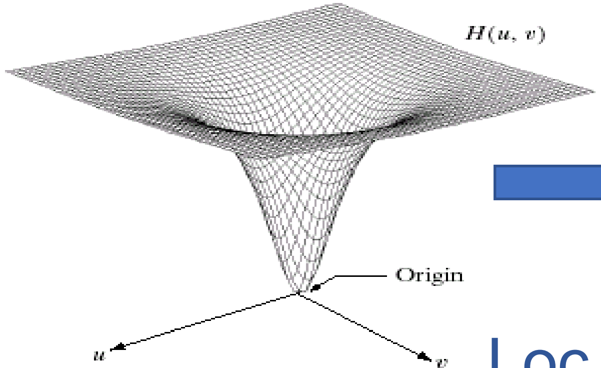


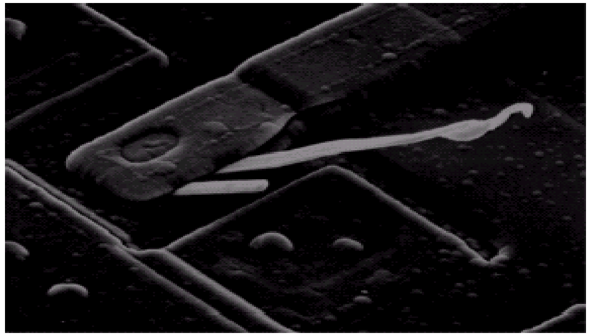
Ảnh gốc



Lọc thông thấp







Lọc thông cao

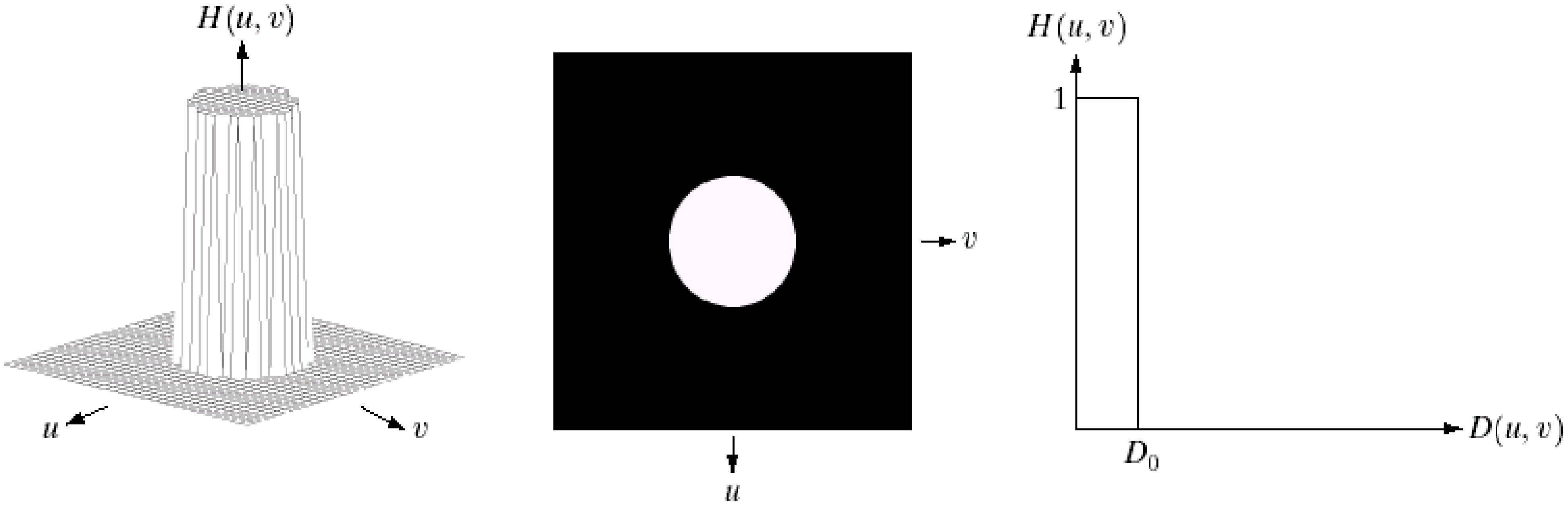
Lọc mịn ảnh (lọc thông thấp)

- Làm mịn ảnh trong miền tần số được thực hiện bằng cách loại bỏ thành phần tần số cao

  ⇨ chỉ cho qua các thành phần tần số thấp, loại bỏ thành phần tần số cao

- Mô hình cơ bản để lọc gồm: $G(u,v) = H(u,v) * F(u,v)$
- Trong đó:
  - $F(u,v)$ là biến đổi Fourier của ảnh cần lọc
  - $H(u,v)$ hàm chuyển đổi lọc

- Cắt bỏ thành phần tần số cao với khoảng cách D_0 so với biến đổi ban đầu



- Thay đổi khoảng cách sẽ làm thay đổi hành vi của bộ lọc

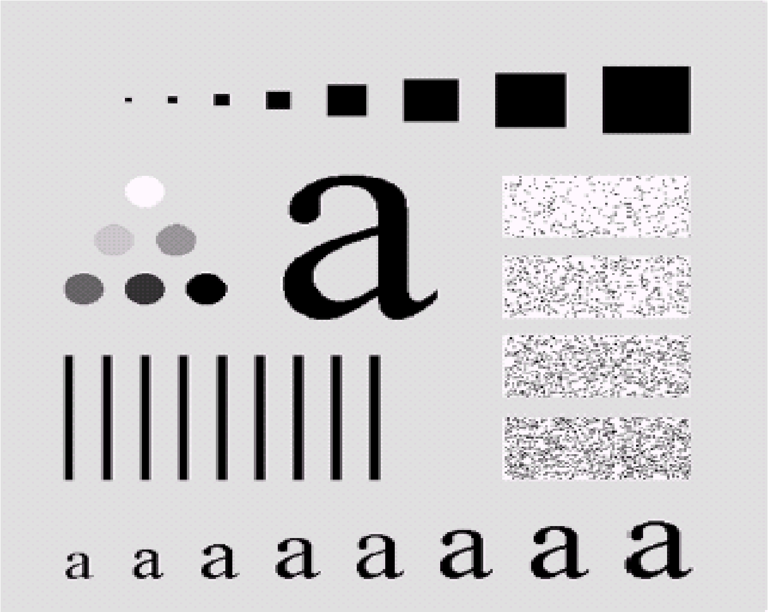
- Hàm chuyển đổi của bộ lọc thông thấp lý tưởng:

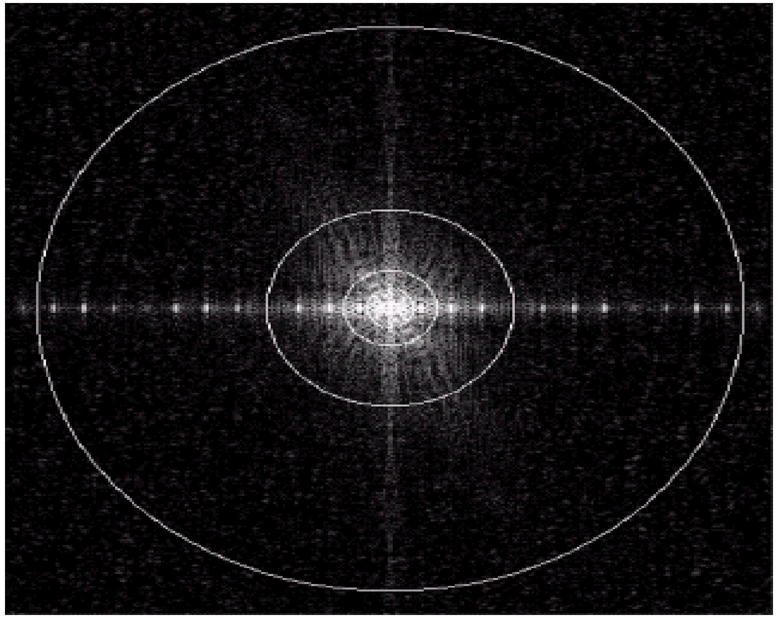
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

- Với hàm khoảng cách

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

Lọc mịn ảnh (lọc thông thấp)

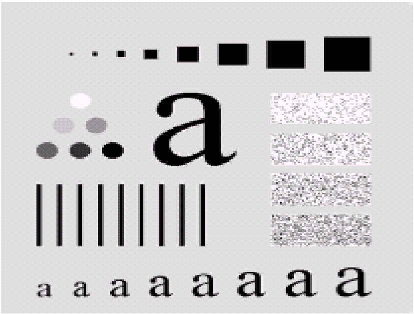


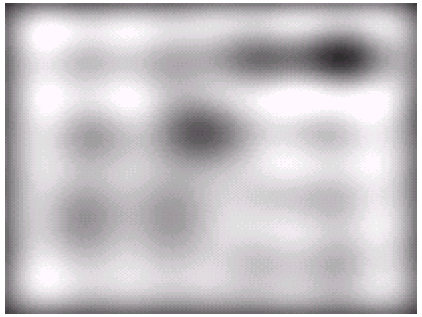


- Ảnh và phổ Fourier của nó
- Một chuỗi các bộ lọc thông thấp lý tưởng với bán kính 5, 15, 30, 80 và 230 chồng lên đỉnh nó

Lọc mịn ảnh (lọc thông thấp)

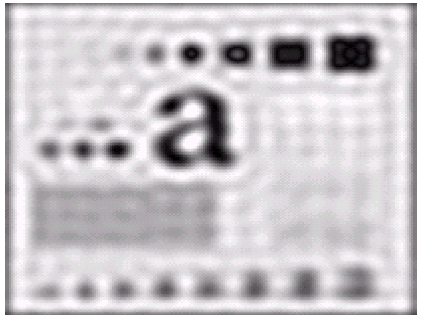
Ảnh gốc

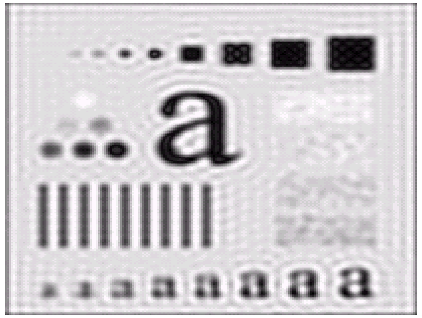




Kết quả lọc với bộ lọc thông thấp lý tưởng bán kính 5

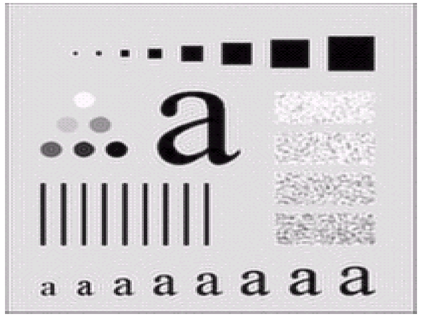
Kết quả lọc với bộ lọc thông thấp lý tưởng bán kính 15

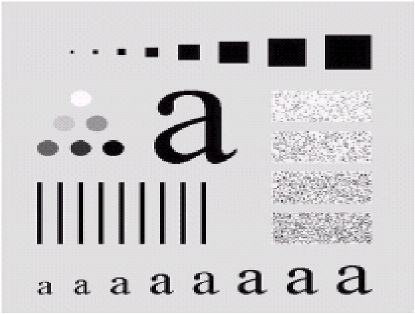




Kết quả lọc với bộ lọc thông thấp lý tưởng bán kính 30

Kết quả lọc với bộ lọc thông thấp lý tưởng bán kính 80



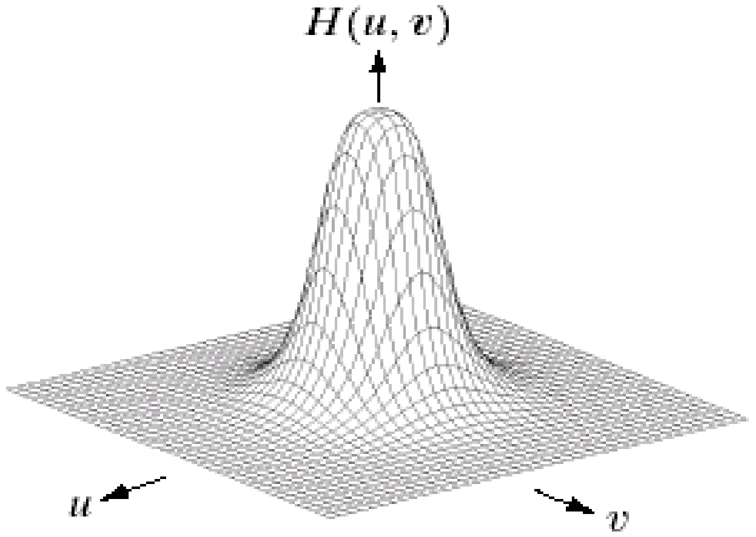


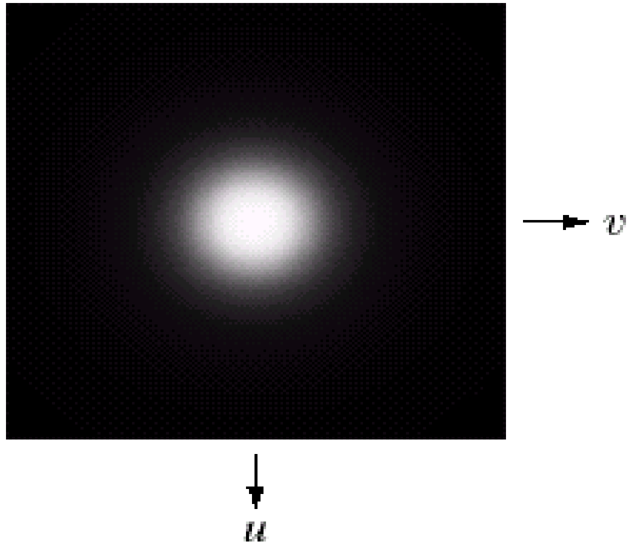
Kết quả lọc với bộ lọc thông thấp lý tưởng bán kính 230

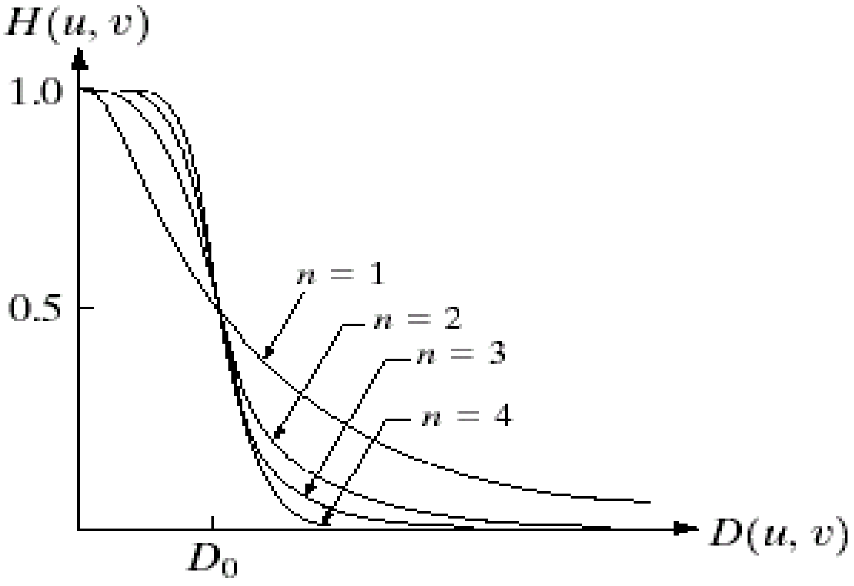
Lọc mịn ảnh (lọc thông thấp)

- **Lọc thông thấp Butterworth**: Hàm biến đổi của lọc thông thấp Butterworth bậc *n* với tần số cắt ở khoảng cách D_0 từ *tâm* được định nghĩa:

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







Lọc mịn ảnh (lọc thông thấp)

- **Lọc thông thấp Butterworth**:

Ảnh gốc

Kết quả lọc với bộ lọc Butterworth bậc 2 và bán kính ngưỡng 5

Kết quả lọc với bộ lọc Butterworth bậc 2 và bán kính ngưỡng 15

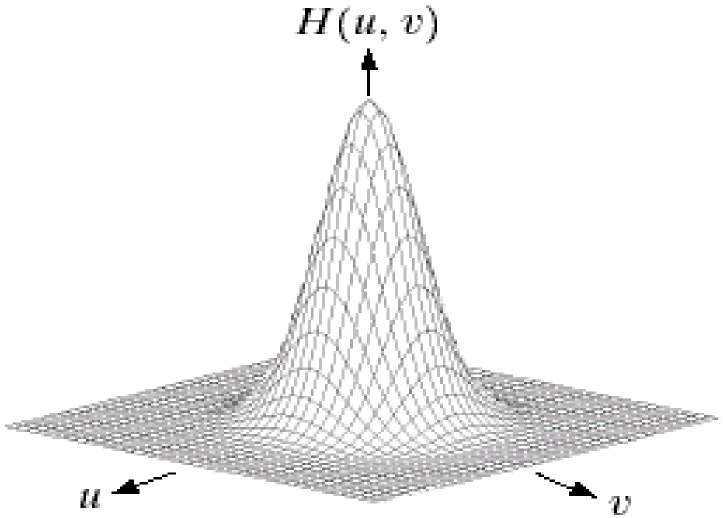
Kết quả lọc với bộ lọc Butterworth bậc 2 và bán kính ngưỡng 30

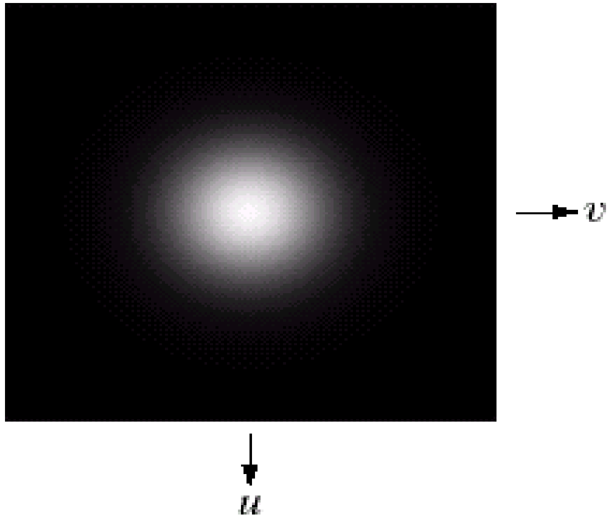
Kết quả lọc với bộ lọc Butterworth bậc 2 và bán kính ngưỡng 80

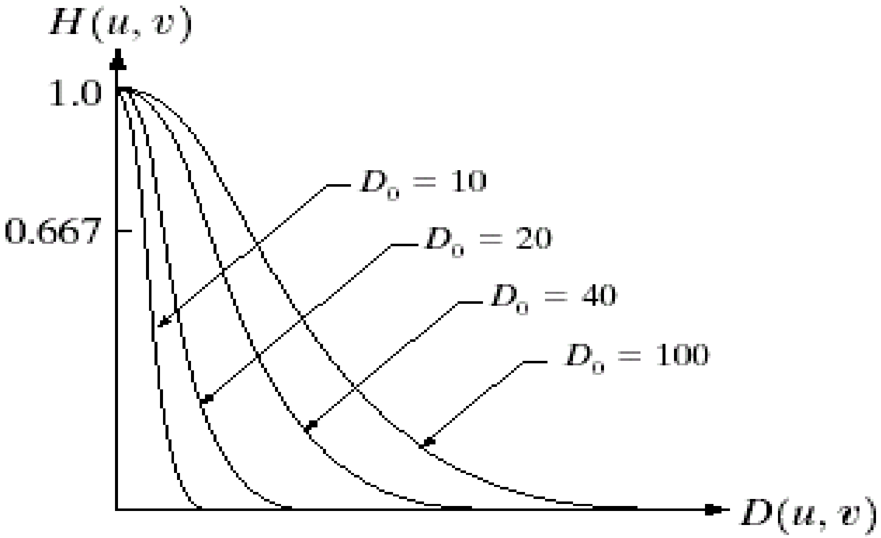
Kết quả lọc với bộ lọc Butterworth bậc 2 và bán kính ngưỡng 230

- **Lọc thông thấp Gaussian:** Hàm biến đổi của lọc thông thấp Gaussian được định nghĩa:

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



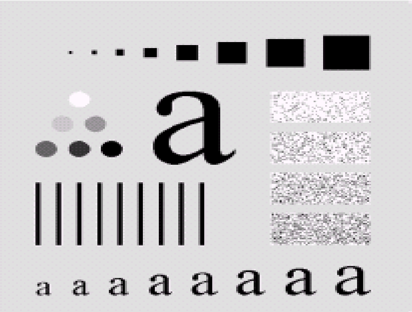


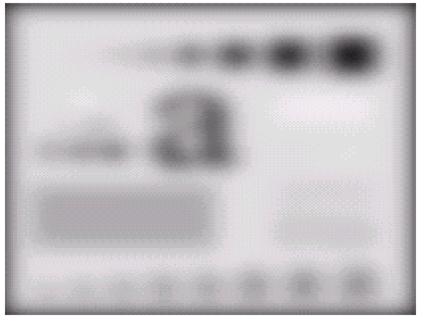


Lọc mịn ảnh (lọc thông thấp)

- **Lọc thông thấp Gaussian:**

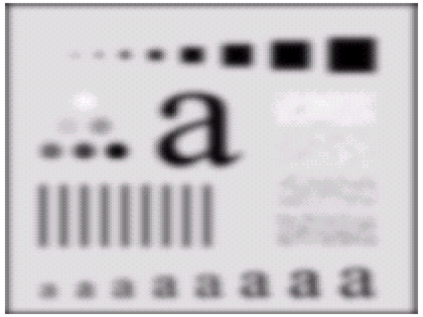
Ảnh gốc

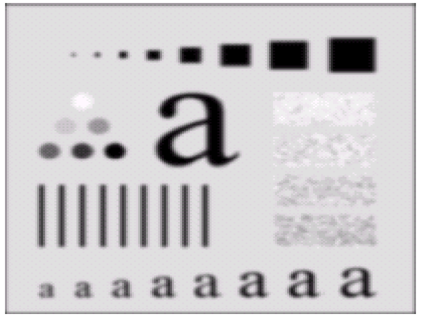




Kết quả lọc với bộ lọc Gaussian với bán kính ngưỡng 5

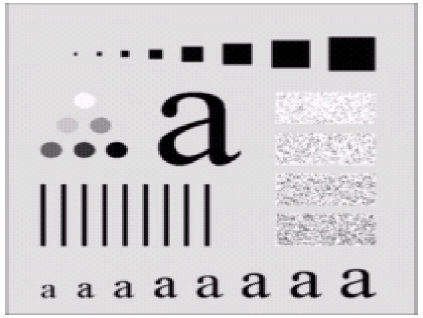
Kết quả lọc với bộ lọc Gaussian với bán kính ngưỡng 15

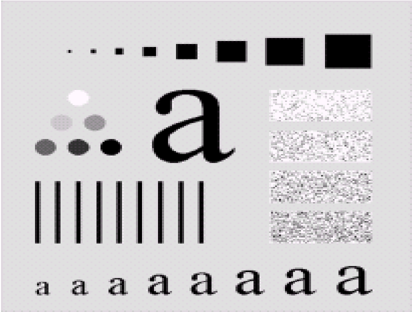




Kết quả lọc với bộ lọc Gaussian với bán kính ngưỡng 30

Kết quả lọc với bộ lọc Gaussian với bán kính ngưỡng 85



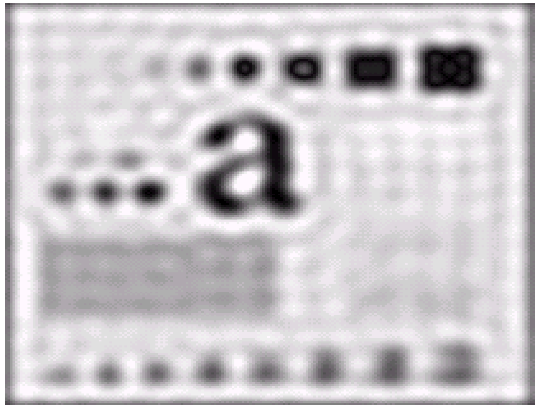


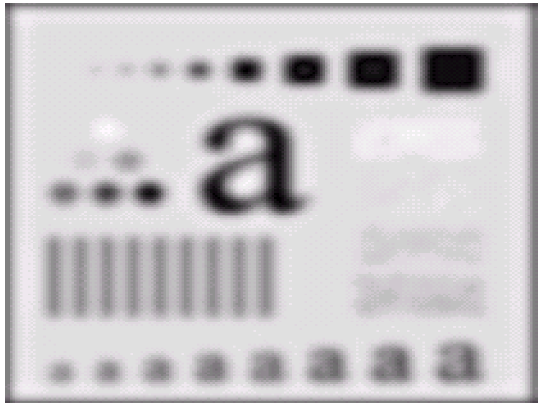
Kết quả lọc với bộ lọc Gaussian với bán kính ngưỡng 230

Lọc mịn ảnh (lọc thông thấp)

- **So sánh kết quả 3 bộ lọc thông thấp**

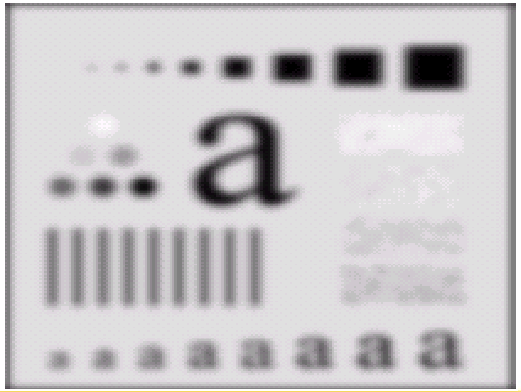
Kết quả lọc với bộ lọc thông thấp lý tưởng bán kính 15





Kết quả lọc với bộ lọc Butterworth bậc 2 và bán kính ngưỡng 15

Kết quả lọc với bộ lọc Gaussian với bán kính ngưỡng 15



Lọc sắc nét ảnh (lọc thông cao)

- Các chi tiết nét trong ảnh thường gắn với các thành phần tần số cao
- Lọc thông cao (High pass filters) – chỉ cho qua các thành phần tần số cao, loại bỏ thành phần tần số thấp
- Lọc thông cao chính là nghịch đảo của bộ lọc thông thấp:

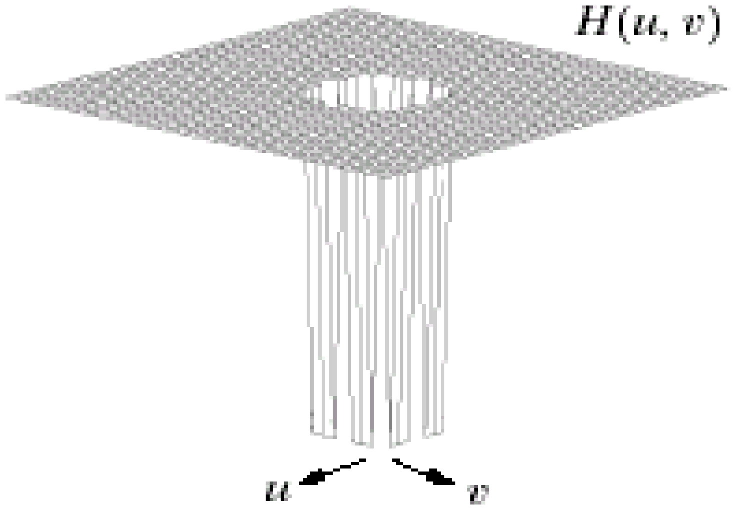
$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

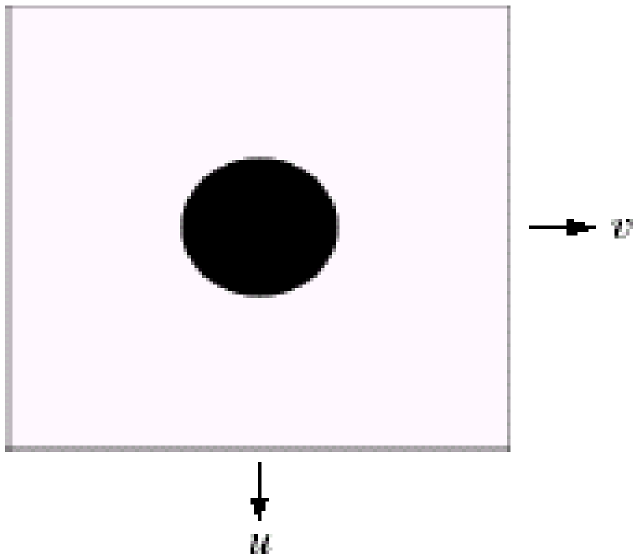
…Lọc sắc nét ảnh (lọc thông cao)

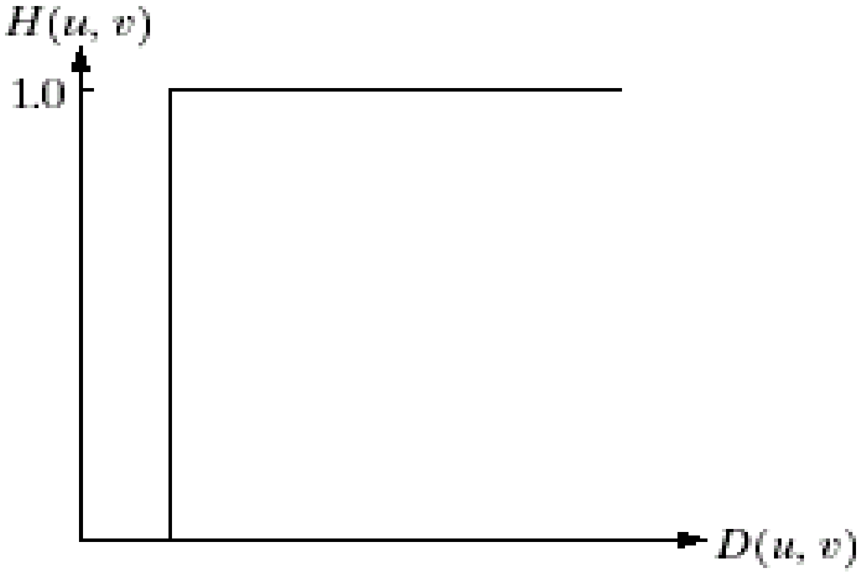
- **Lọc thông cao lý tưởng** như sau:

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

(D_0 là khoảng cách cắt)

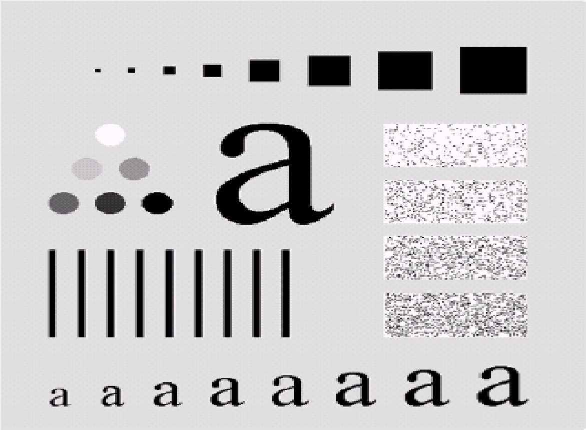




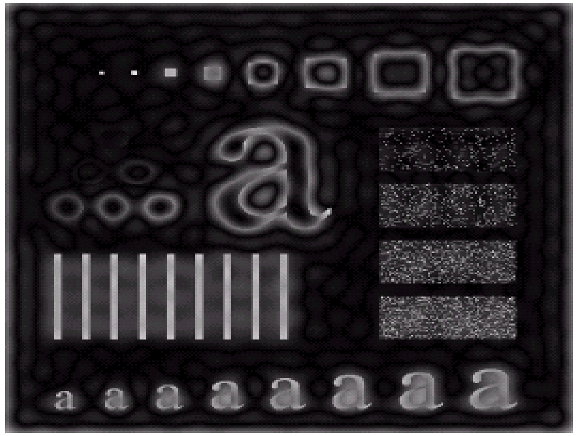


...Lọc sắc nét ảnh (lọc thông cao)

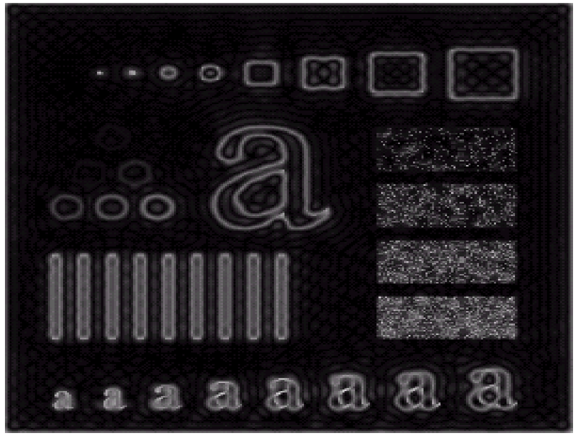
- **Lọc thông cao lý tưởng** (tiếp theo)



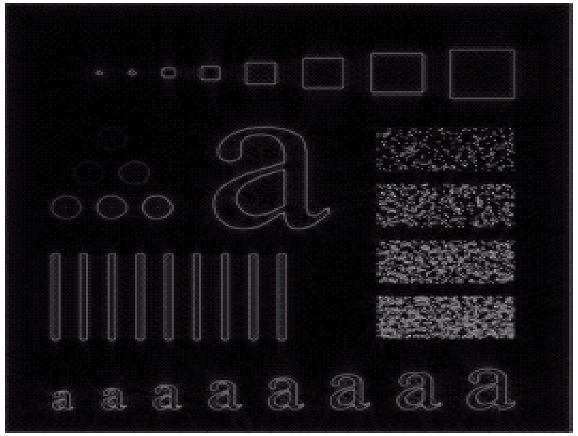
Ảnh gốc



Kết quả lọc thông cao lý tưởng với $D_0 = 15$



Kết quả lọc thông cao lý tưởng với $D_0 = 30$



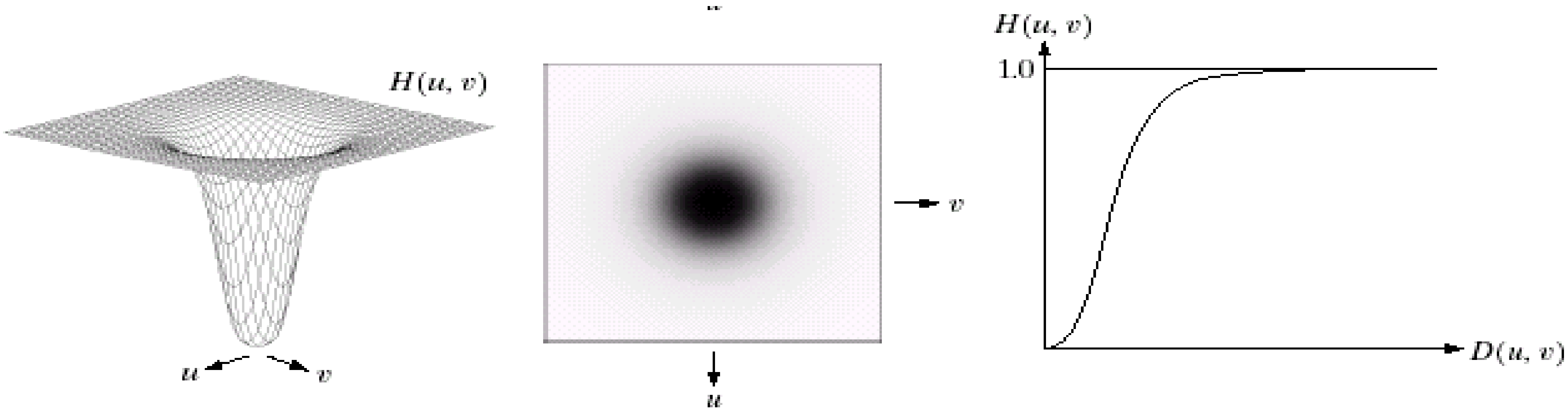
Kết quả lọc thông cao lý tưởng với $D_0 = 80$

…Lọc sắc nét ảnh (lọc thông cao)

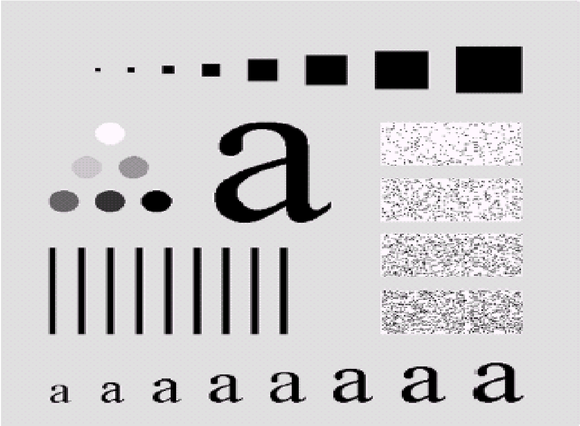
- **Lọc thông cao Butterworth** như sau:

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

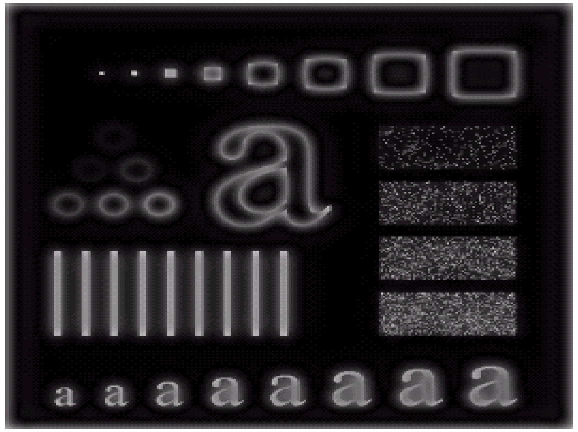
(n là bậc của bộ lọc và D_0 là khoảng cách cắt)



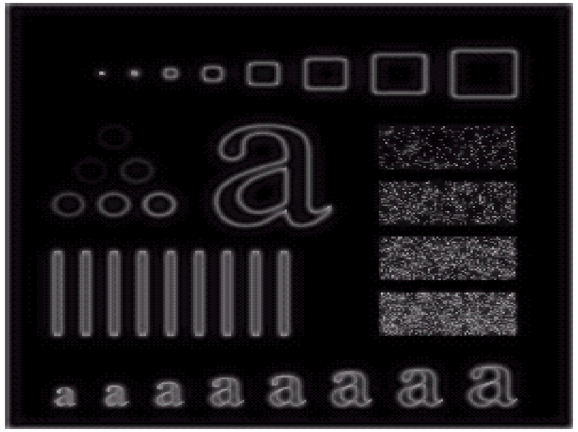
- **Lọc thông cao Butterworth** như sau:



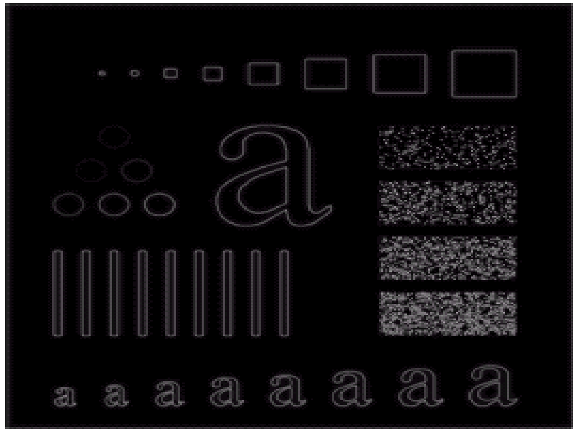
Ảnh gốc



Kết quả lọc thông cao Butterworth bậc 2 với $D_0 = 15$



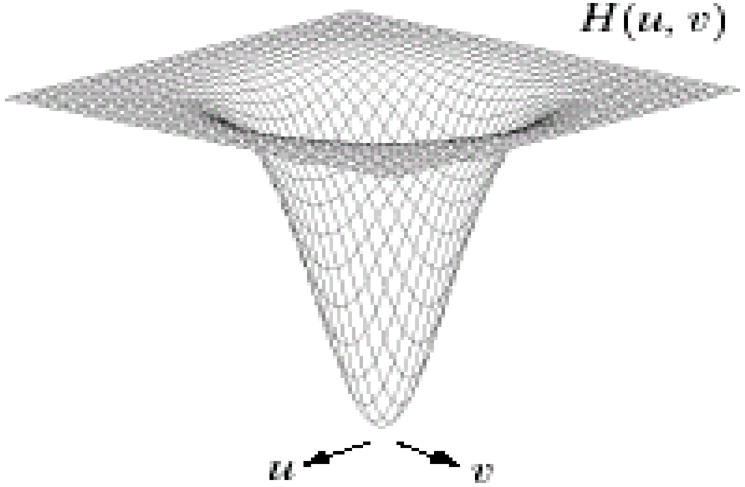
Kết quả lọc thông cao Butterworth bậc 2 với $D_0 = 30$

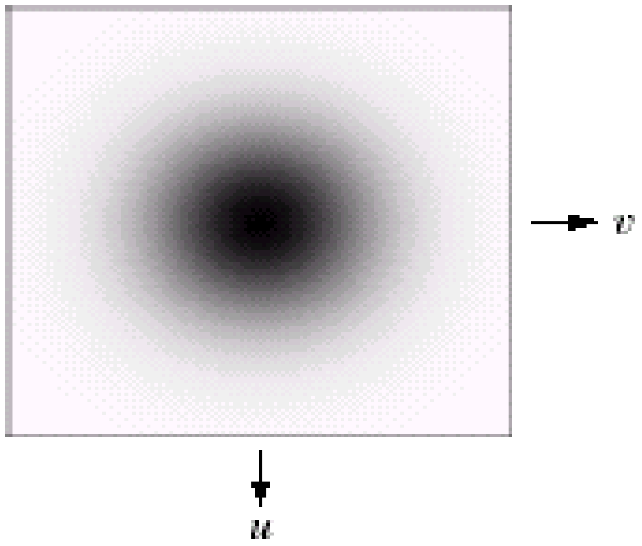


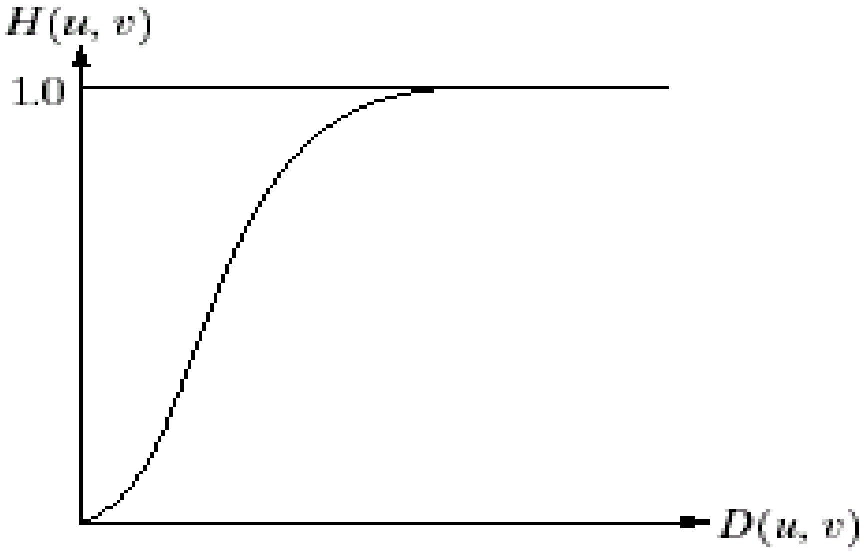
Kết quả lọc thông cao Butterworth bậc 2 với $D_0 = 80$

- **Lọc thông cao Gaussian** như sau:

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

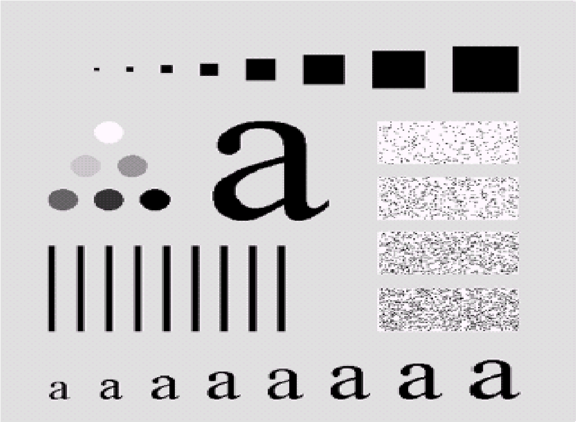




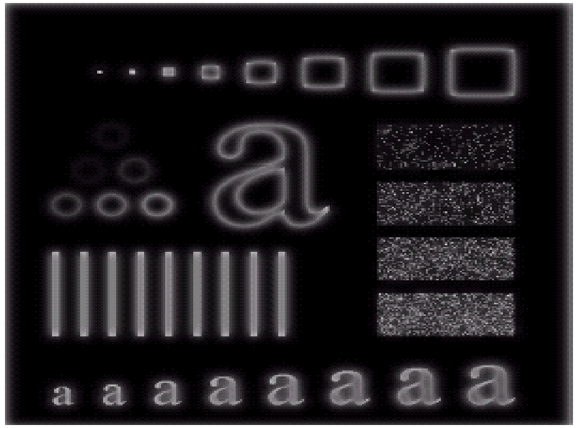


…Lọc sắc nét ảnh (lọc thông cao)

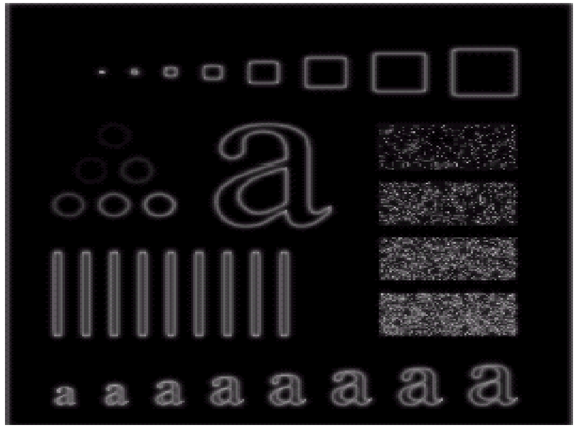
- **Lọc thông cao Gaussian**



Ảnh gốc



Kết quả lọc thông cao Gaussian với $D_0 = 15$

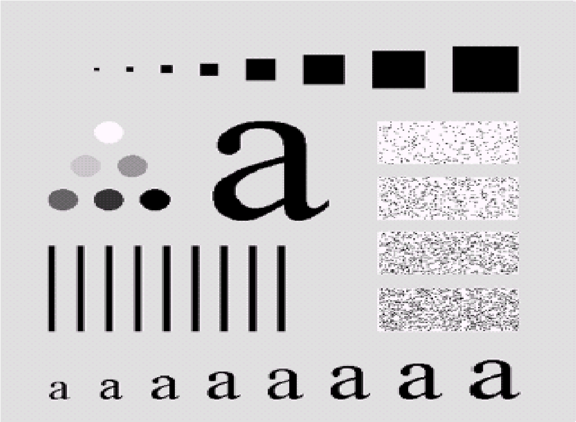


Kết quả lọc thông cao Gaussian với $D_0 = 30$

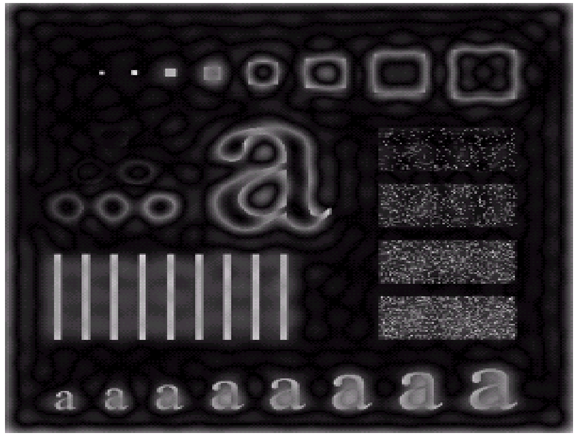
Kết quả lọc thông cao Gaussian với $D_0 = 80$

…Lọc sắc nét ảnh (lọc thông cao)

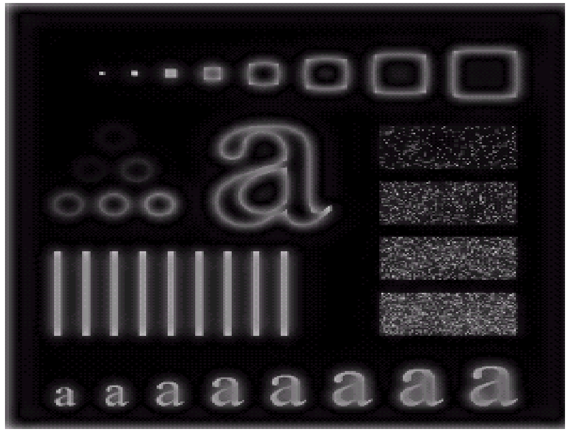
- **So sánh các lọc thông cao**



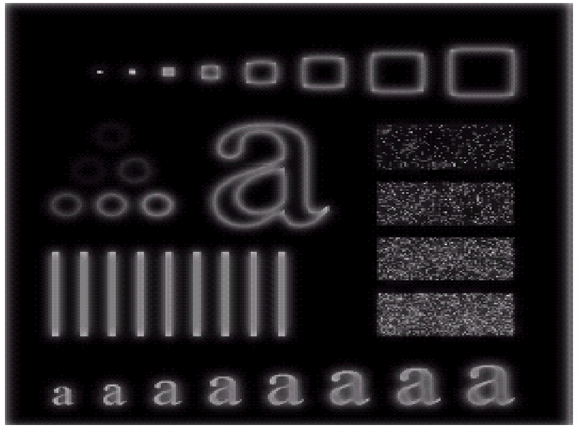
Ảnh gốc



Kết quả lọc thông cao lý tưởng với $D_0 = 15$

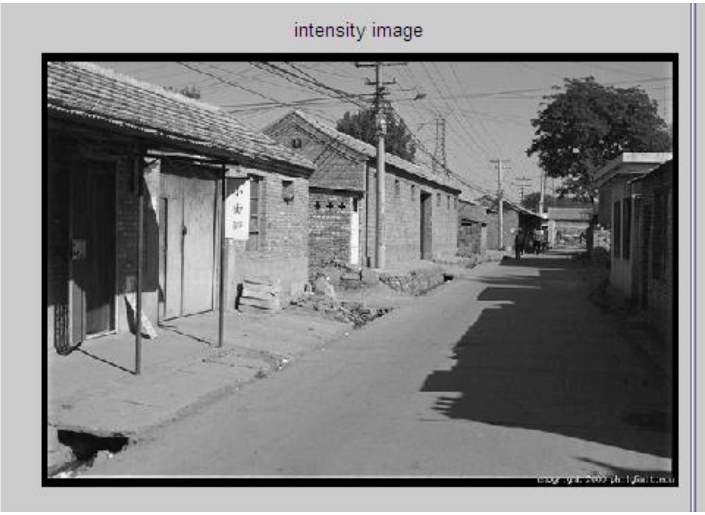


Kết quả lọc thông cao **Butterworth** bậc 2 với $D_0 = 15$



Kết quả lọc thông cao **Gaussian** với $D_0 = 15$

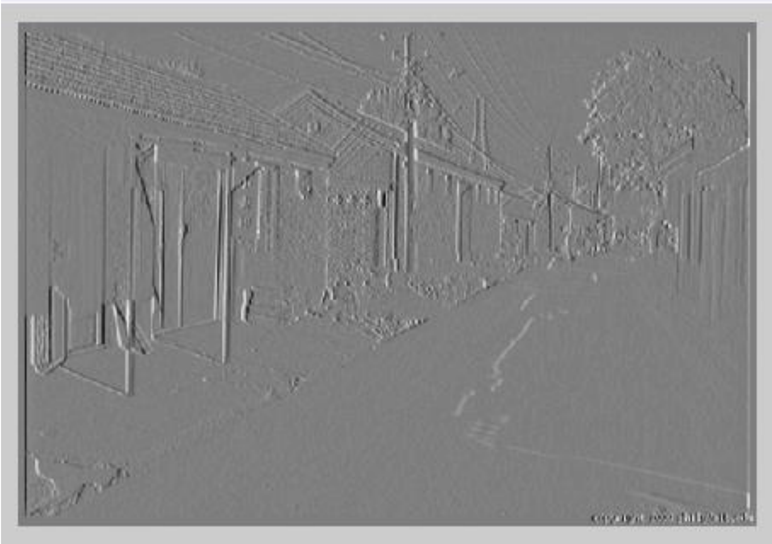
Filtering in spatial domain



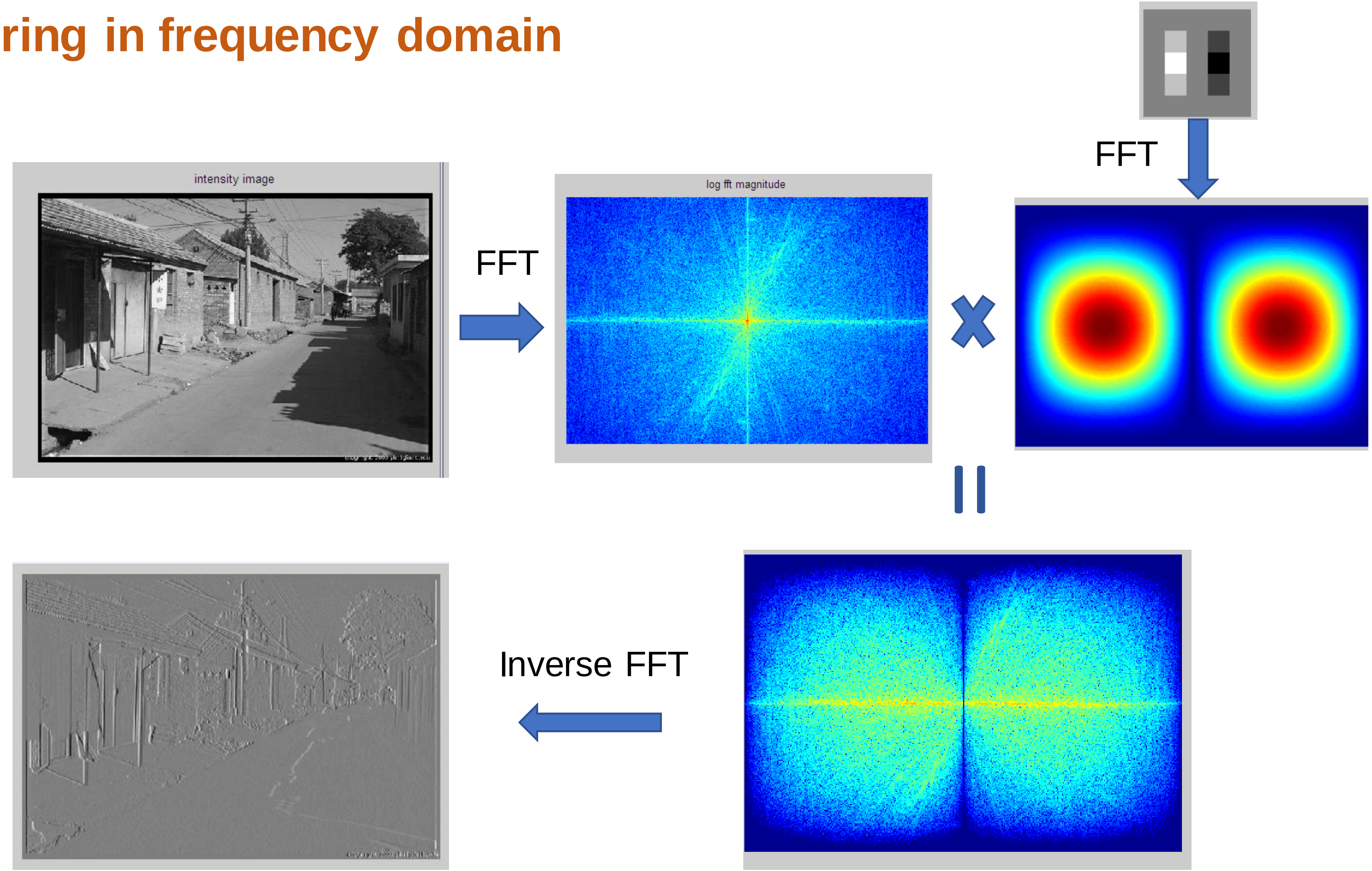
1	0	-1
2	0	-2
1	0	-1

*

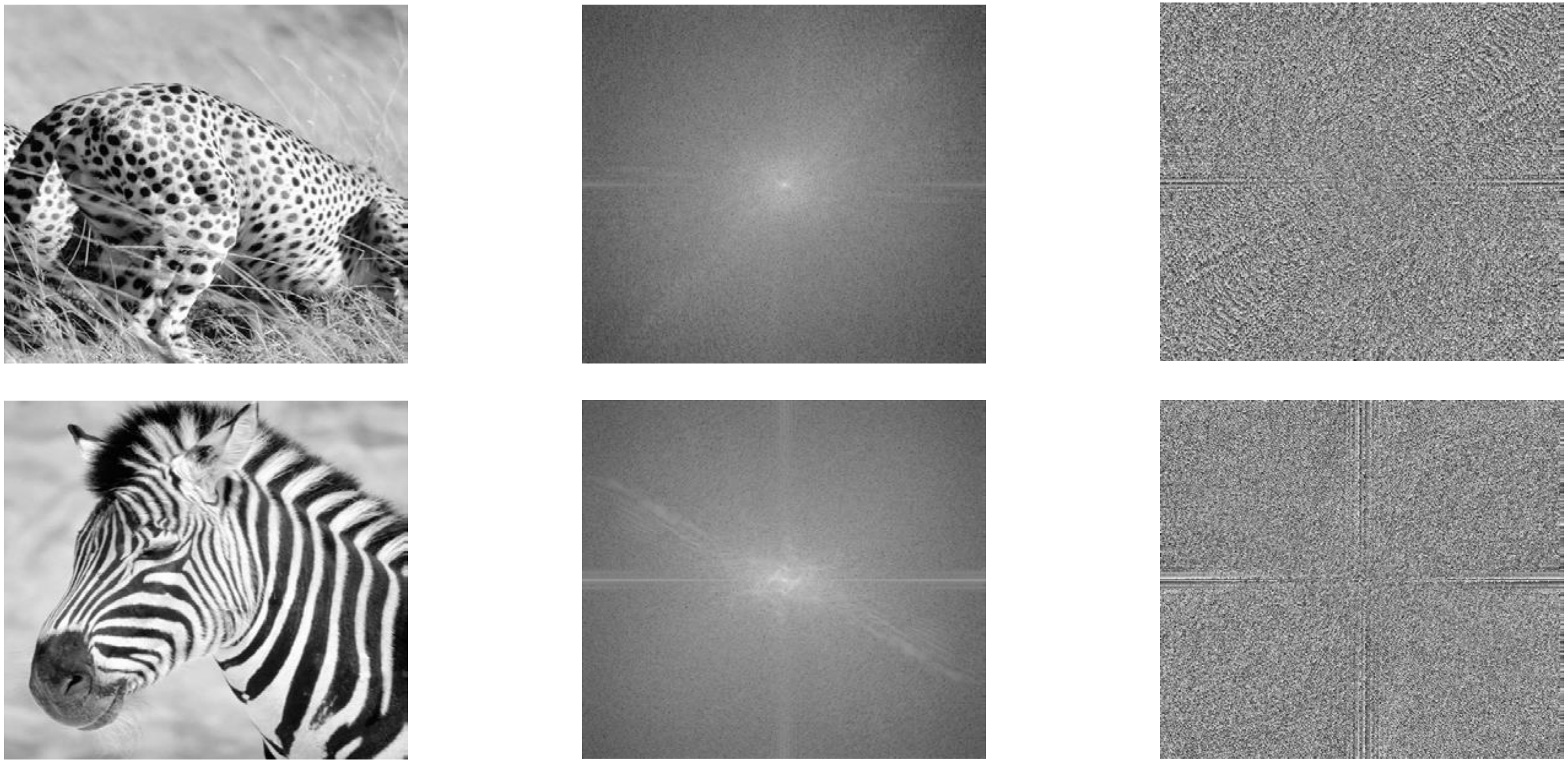
=



Filtering in frequency domain



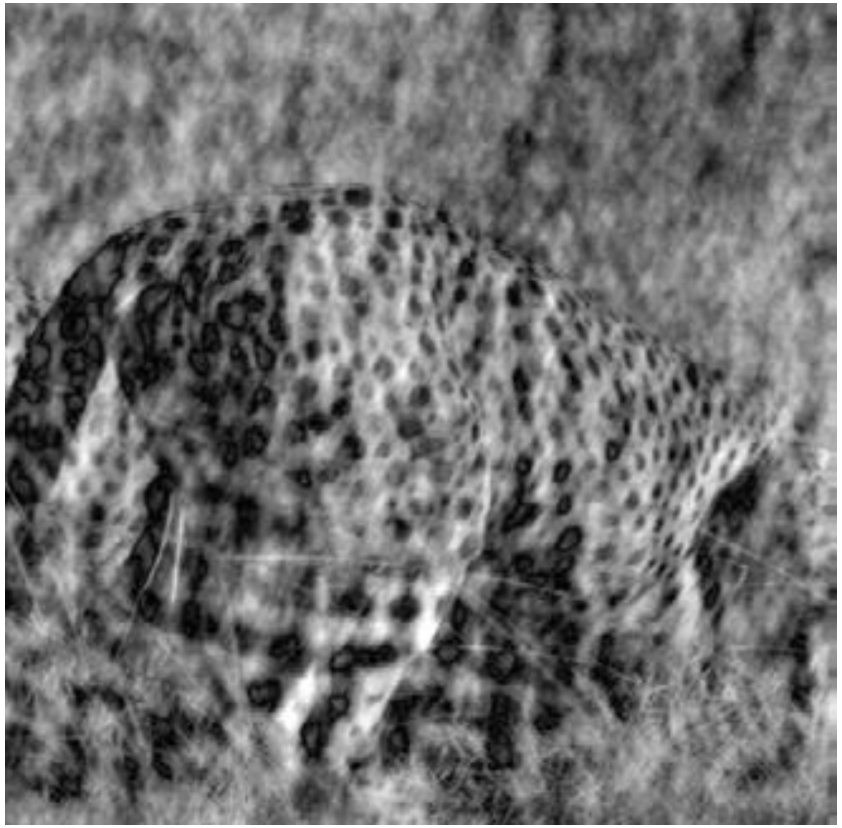
Fourier transforms of natural images



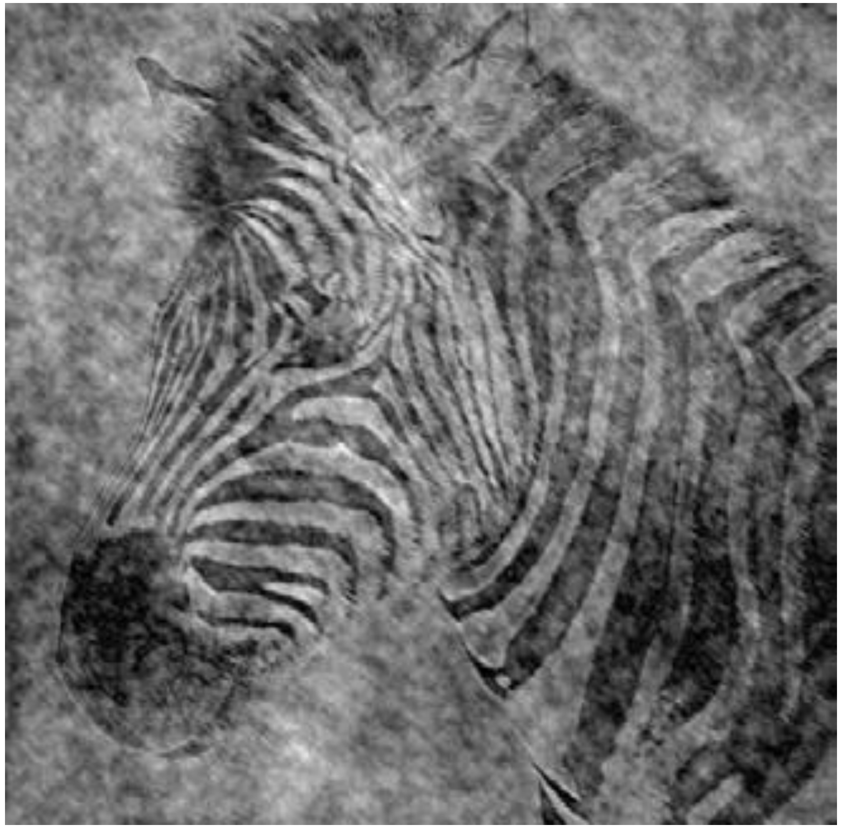
original amplitude phase

Fourier transforms of natural images

Image phase matters!



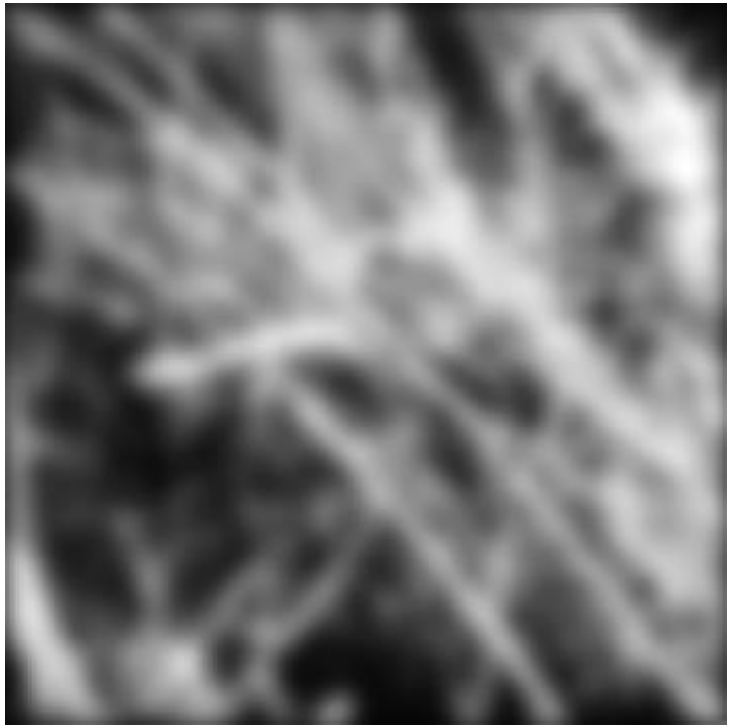
cheetah phase with zebra amplitude



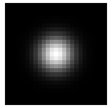
zebra phase with cheetah amplitude

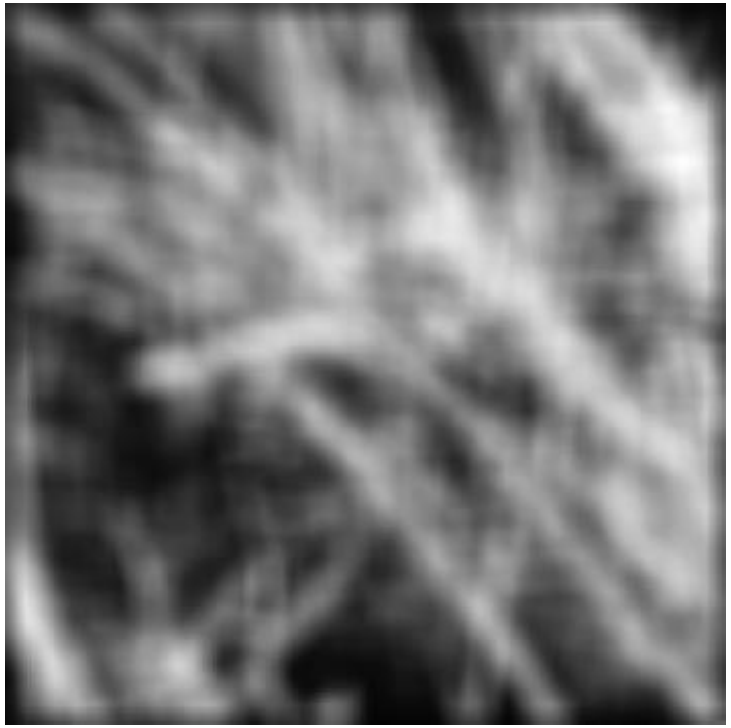
Revisiting blurring

Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

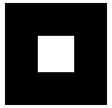


Gaussian filter

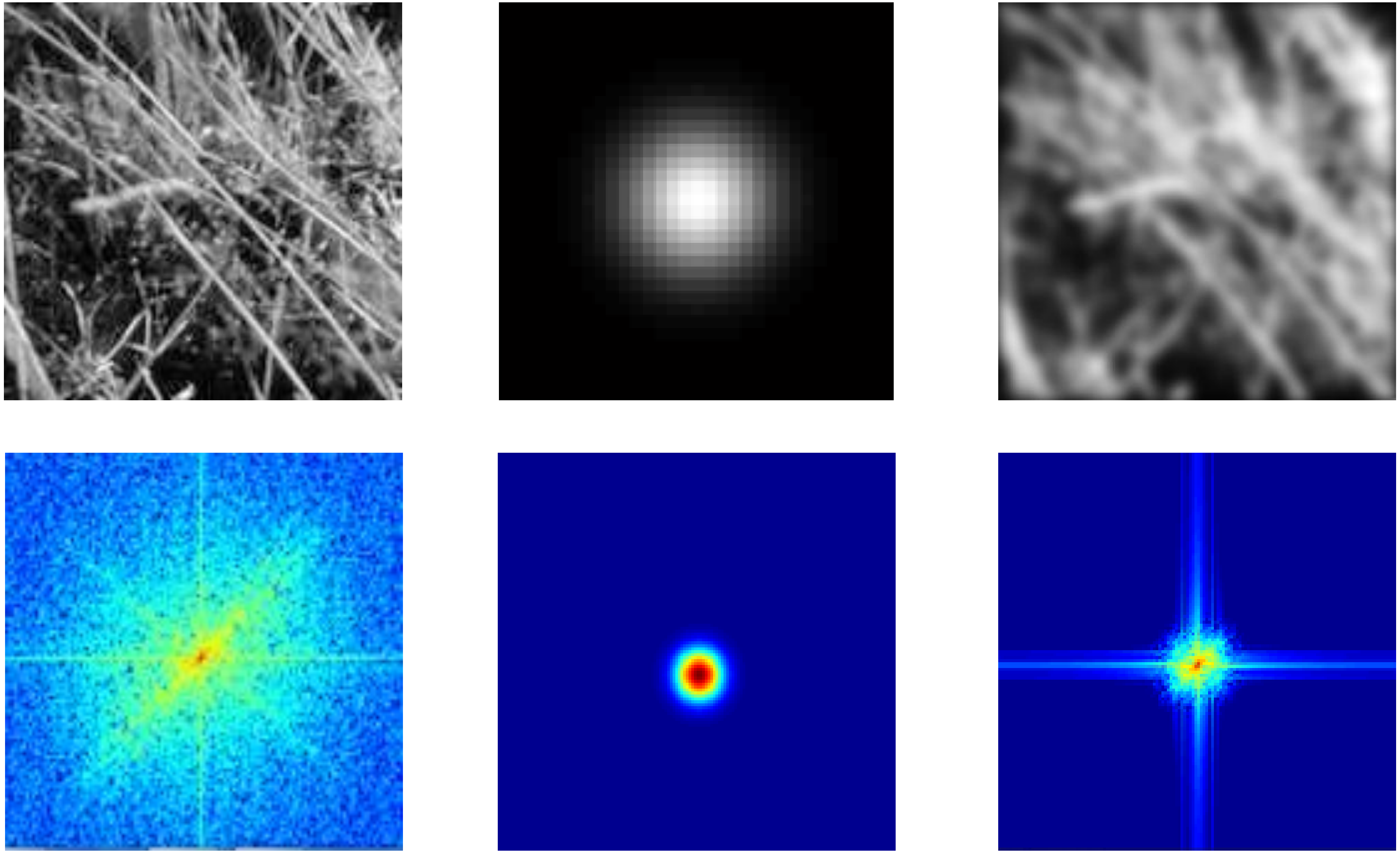




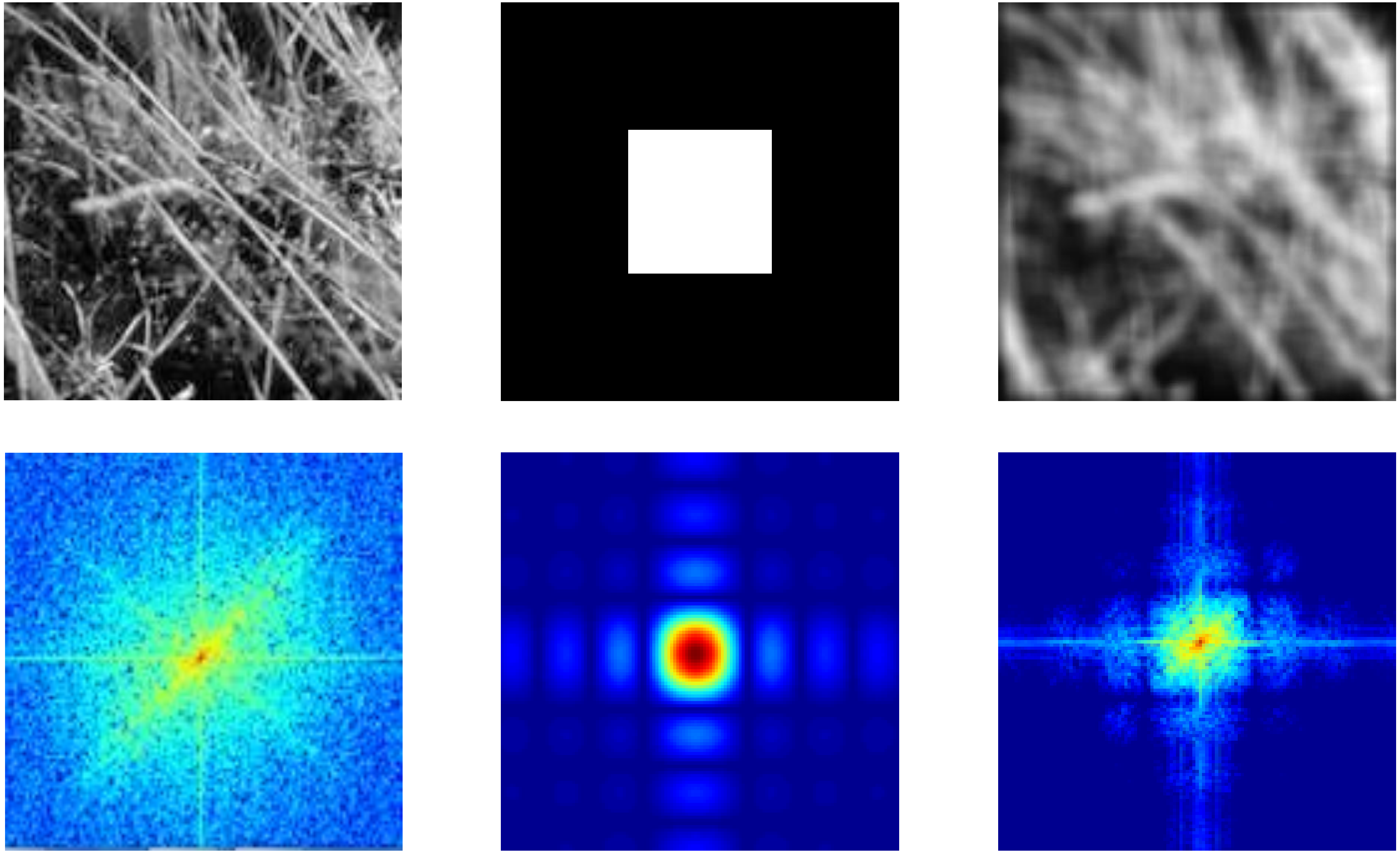
Box filter



Gaussian blur

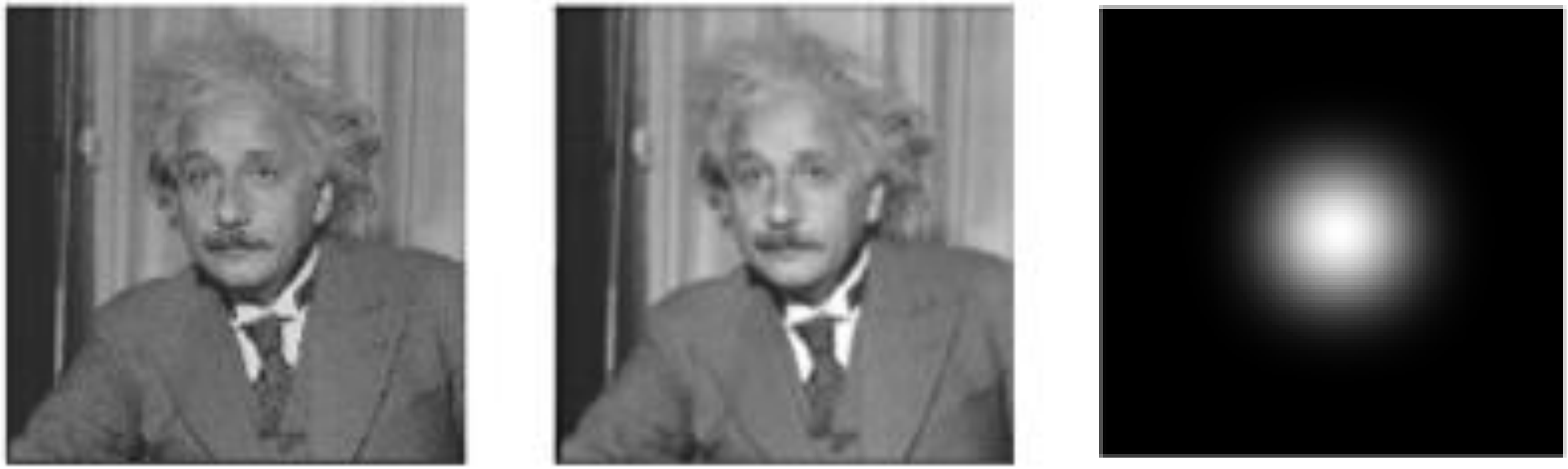


Box blur



More filtering examples

low-pass

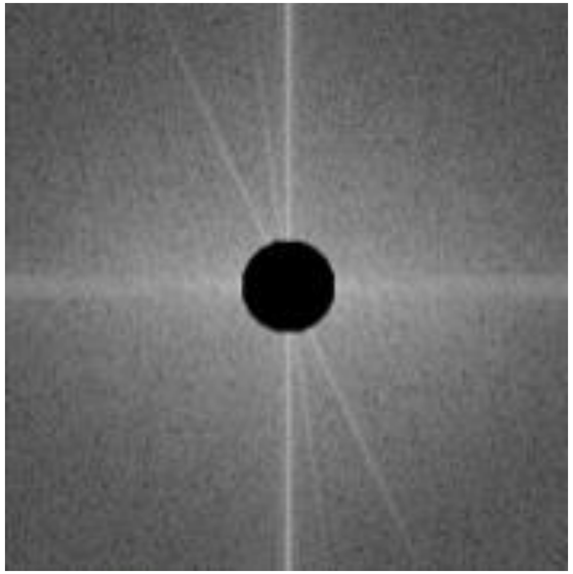


band-pass

filters shown in frequency-domain

More filtering examples

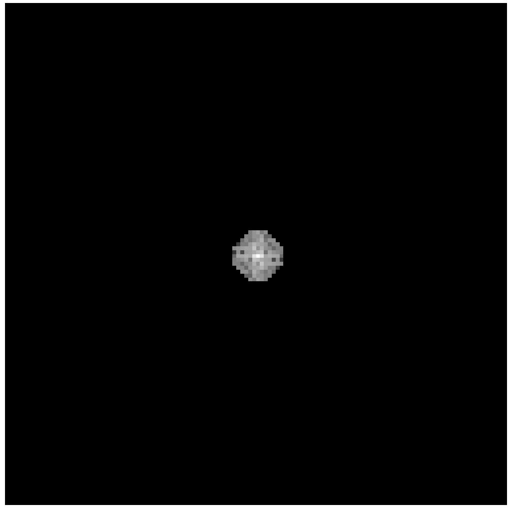
high-pass



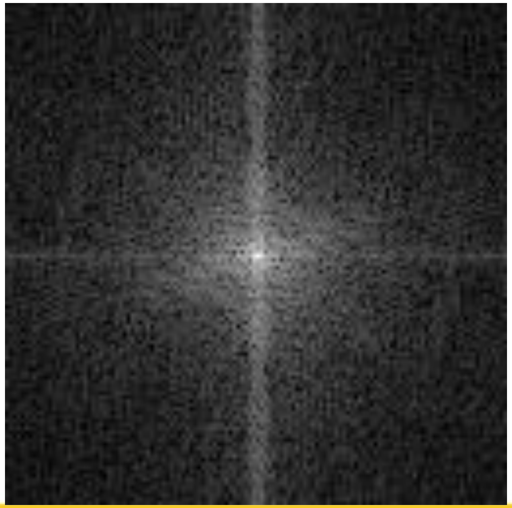
More filtering examples

original image

low-pass filter



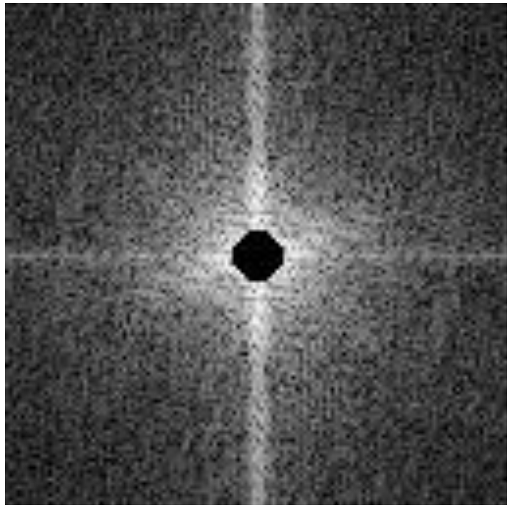
frequency magnitude



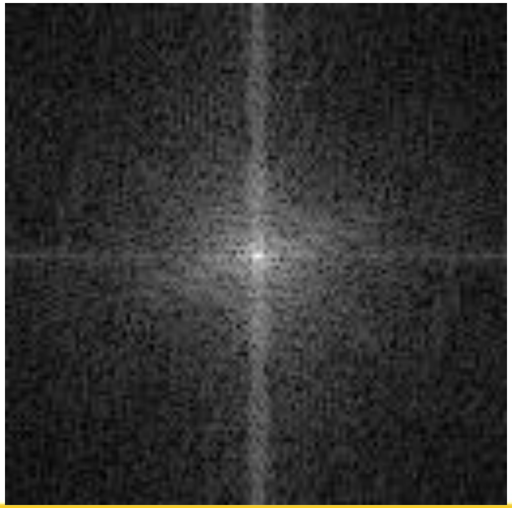
More filtering examples

original image

high-pass filter



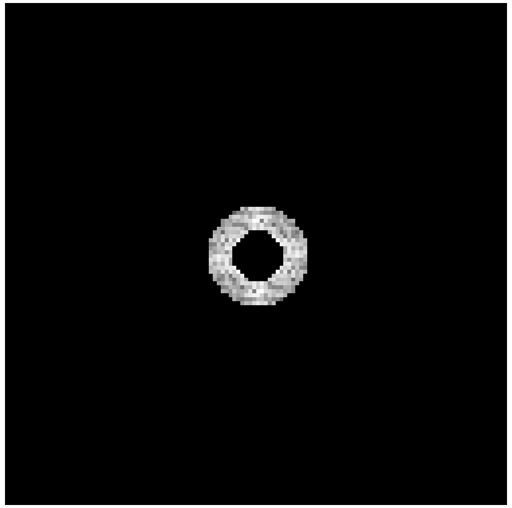
frequency magnitude

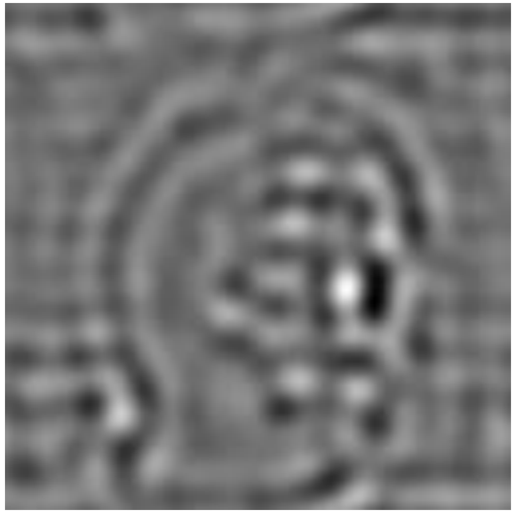


More filtering examples

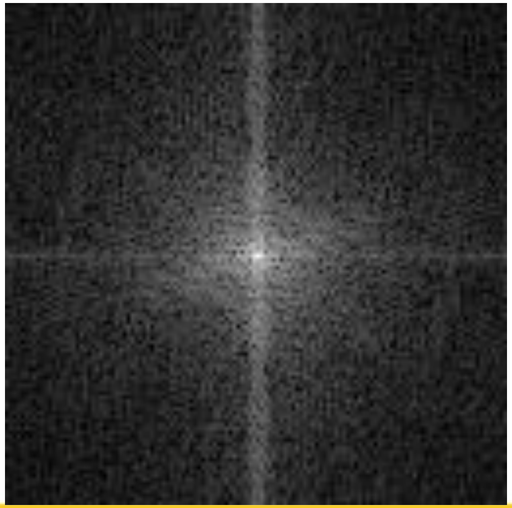
original image

band-pass filter





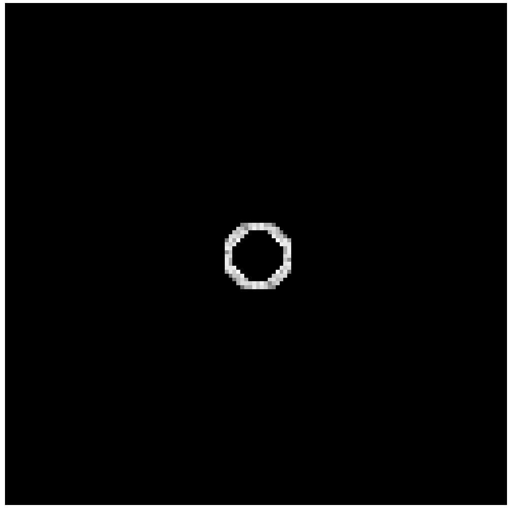
frequency magnitude

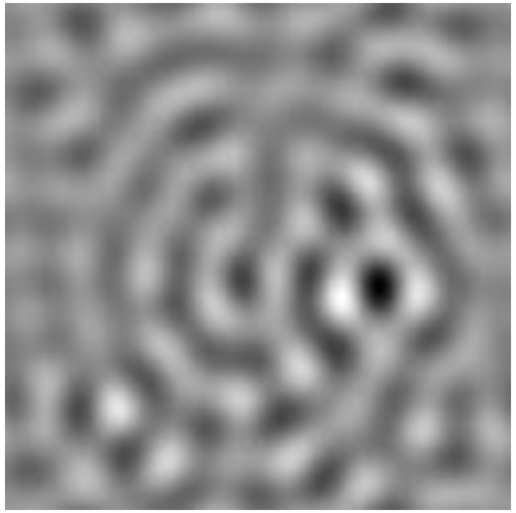


More filtering examples

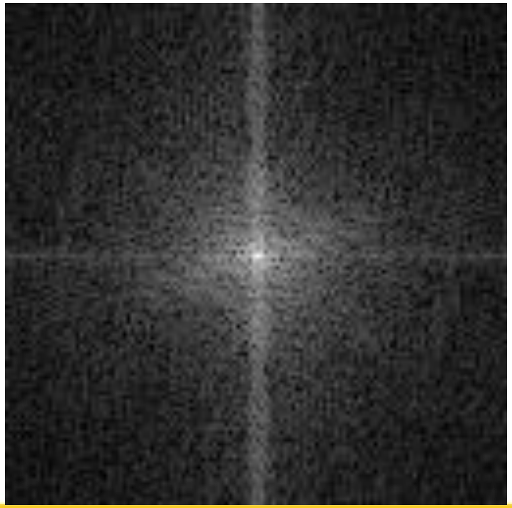
original image

band-pass filter





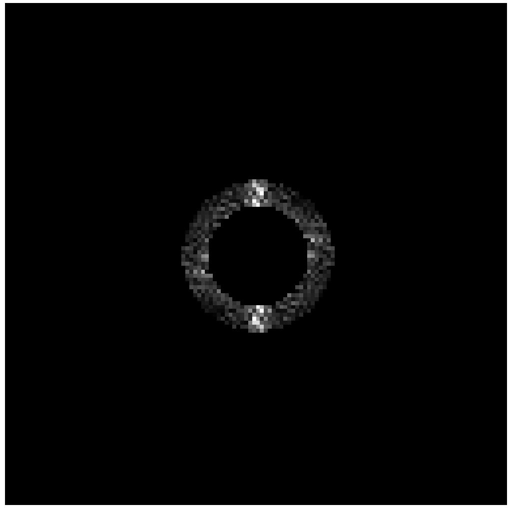
frequency magnitude



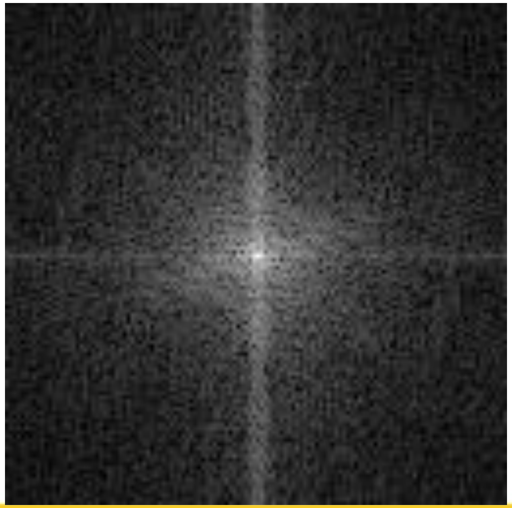
More filtering examples

original image

band-pass filter



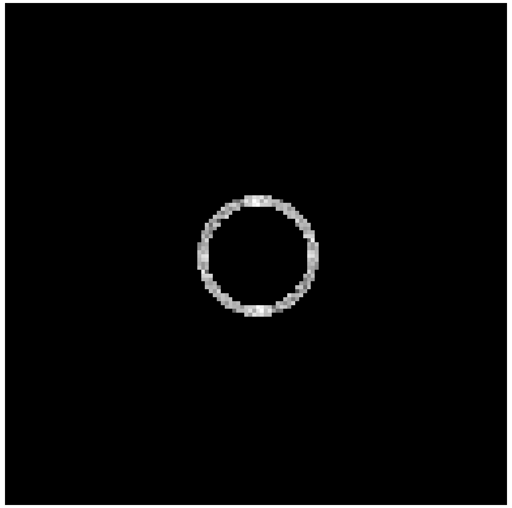
frequency magnitude

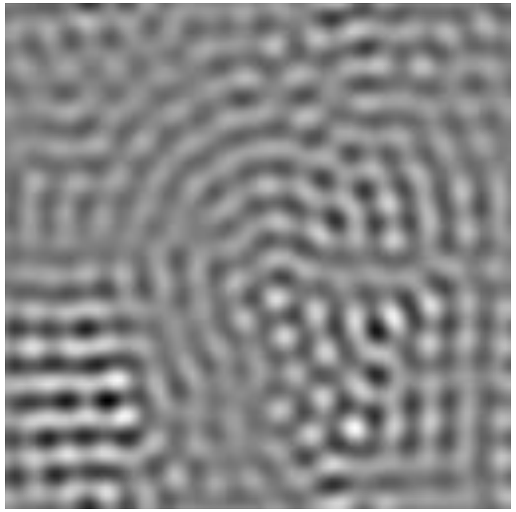


More filtering examples

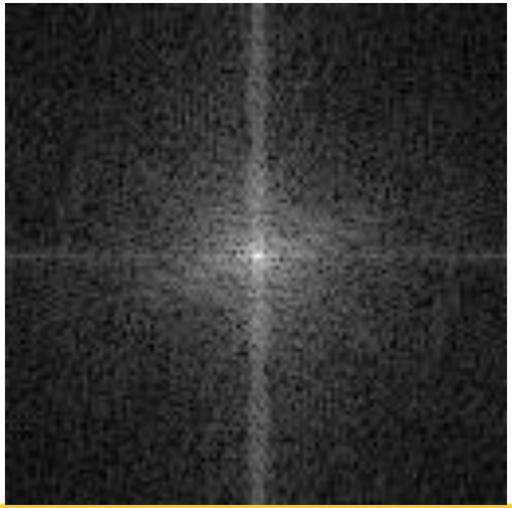
original image

band-pass filter





frequency magnitude

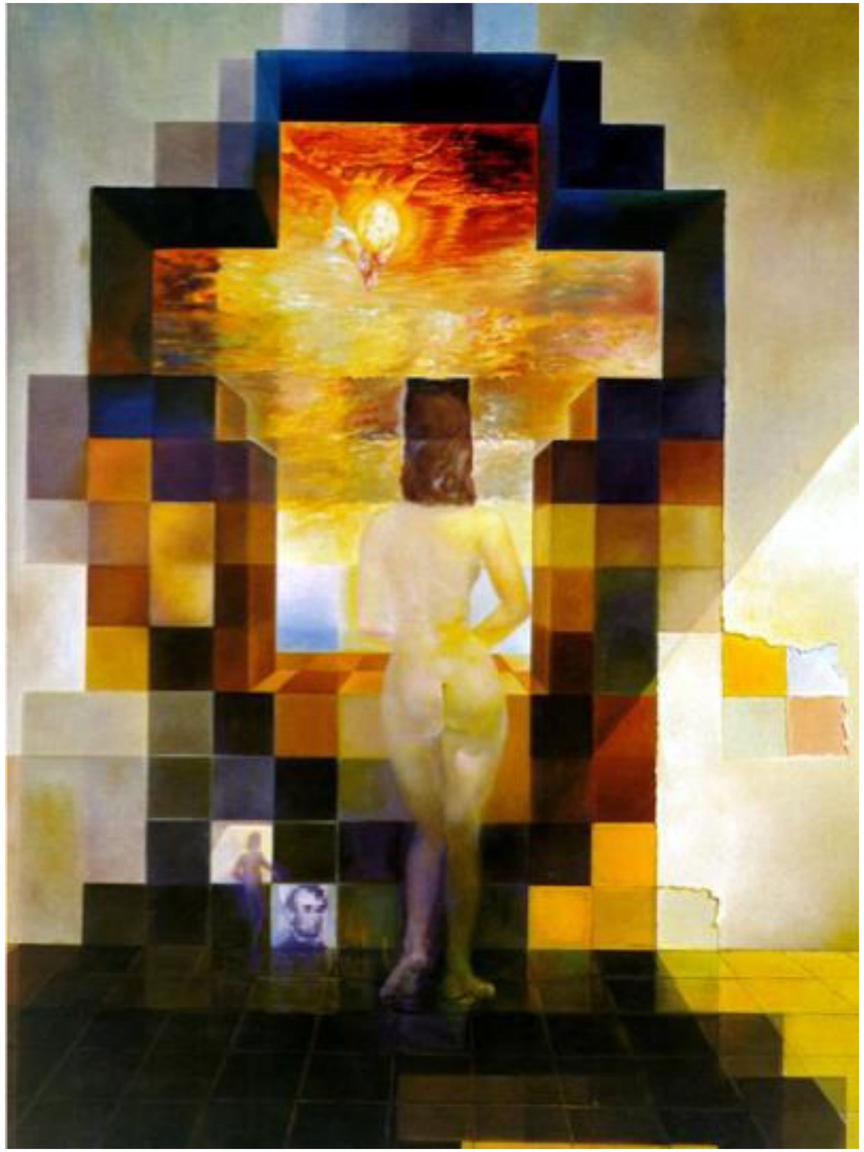


Frequency-domain filtering in human vision

"Hybrid image"

Aude Oliva and Philippe Schyns

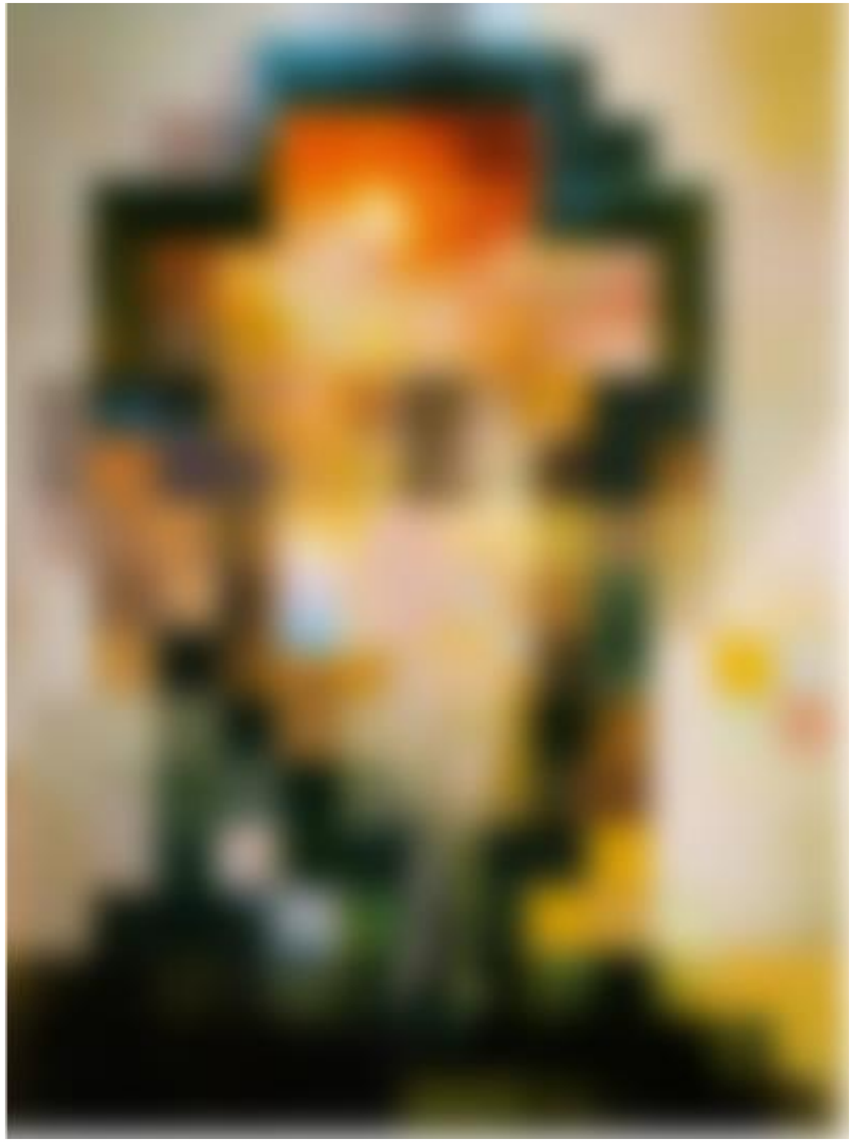
Frequency-domain filtering in human vision



Gala Contemplating the Mediterranean Sea Which at Twenty Meters Becomes the Portrait of Abraham Lincoln (Homage to Rothko)

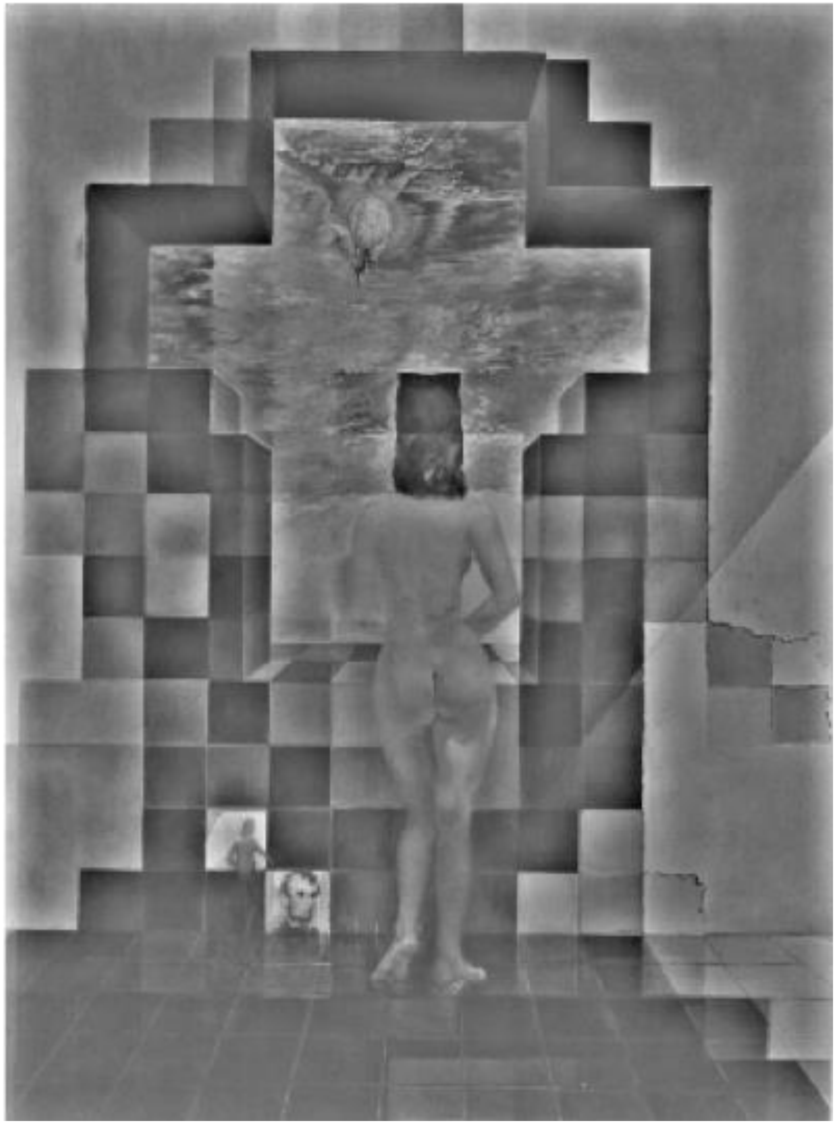
Salvador Dali, 1976

Frequency-domain filtering in human vision



Low-pass filtered version

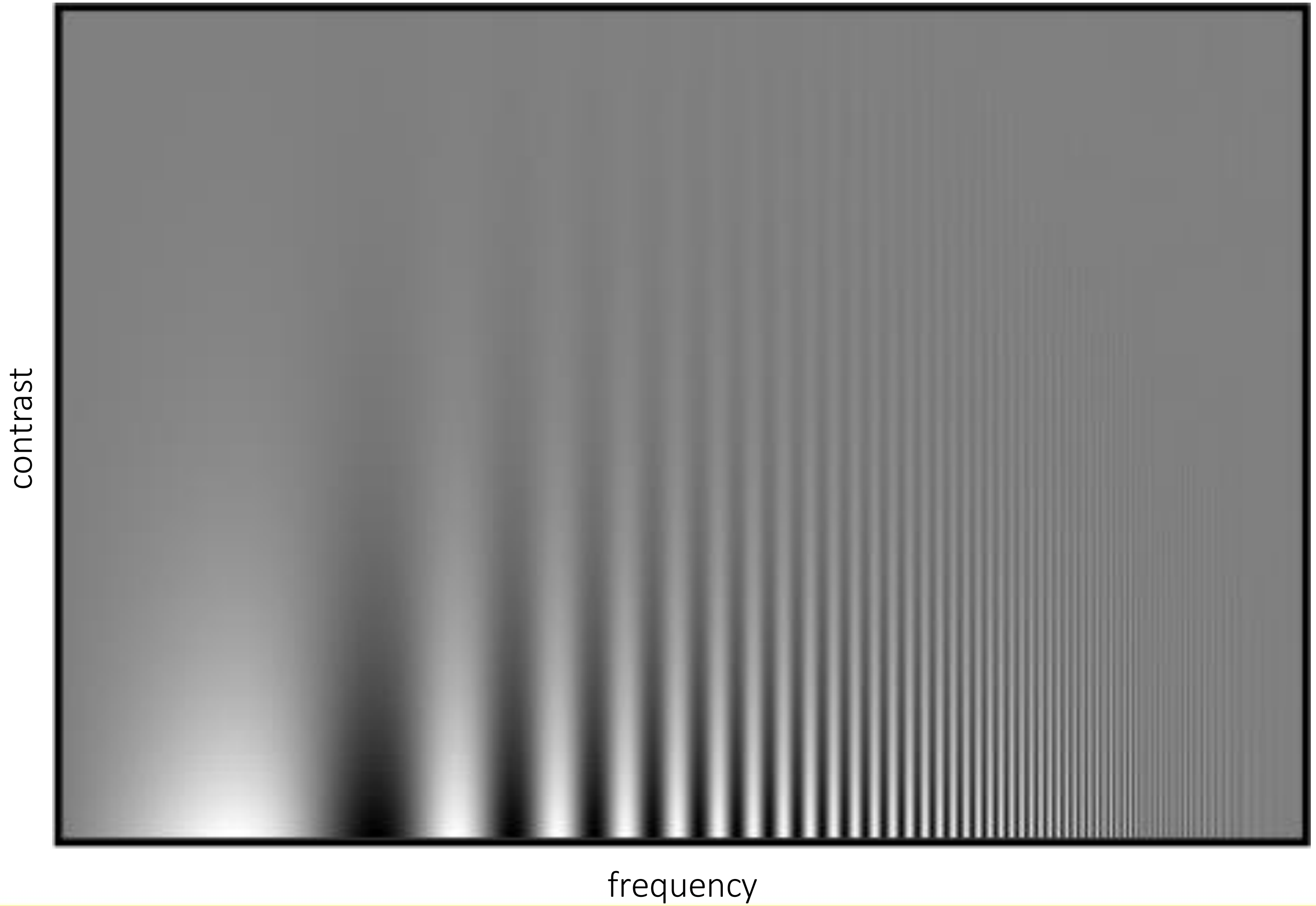
Frequency-domain filtering in human vision



High-pass filtered version

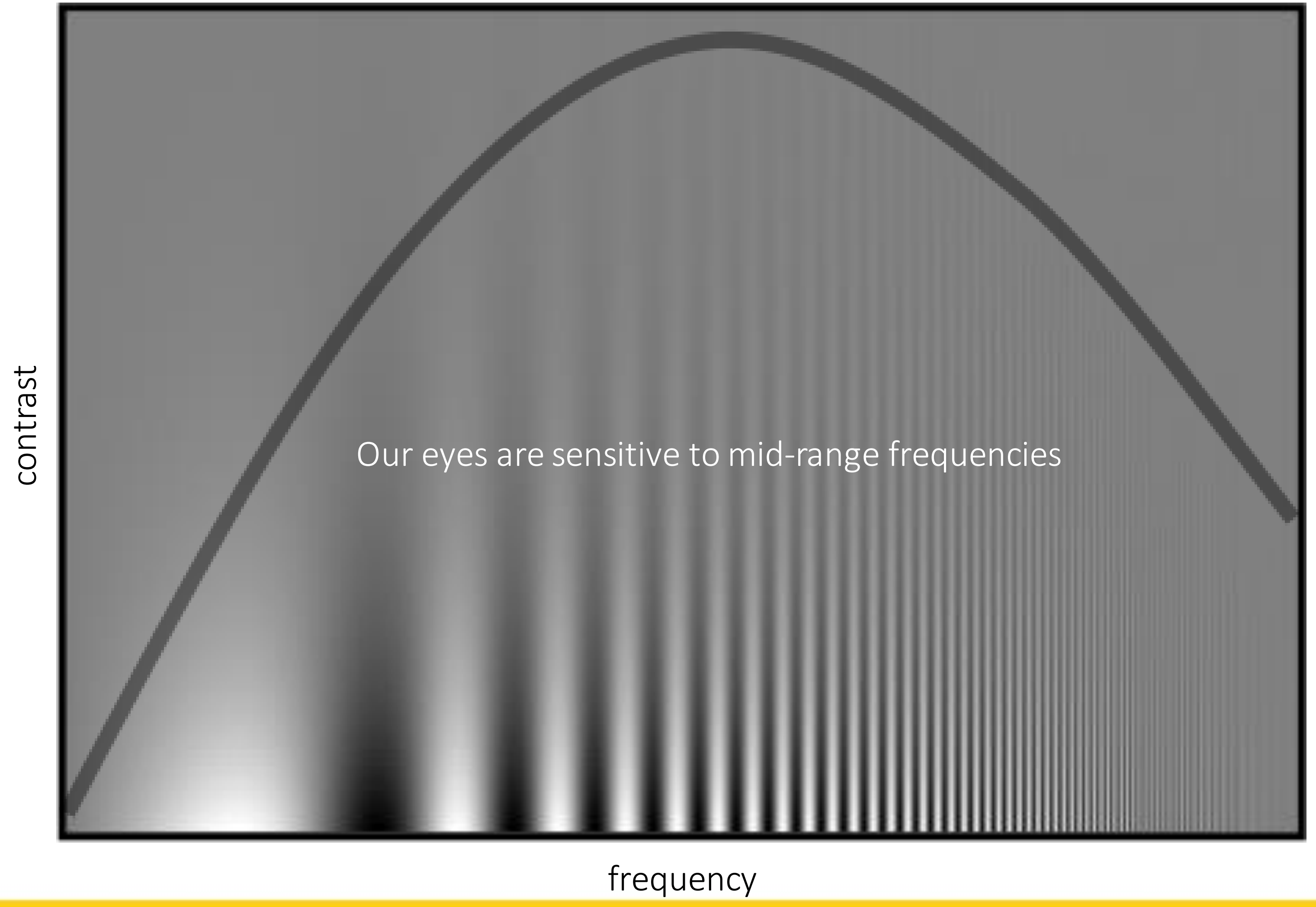
Variable frequency sensitivity

Experiment: Where do you see the stripes?



Variable frequency sensitivity

Campbell-Robson contrast sensitivity curve



- Early processing in humans filters for various orientations and scales of frequency
- Perceptual cues in the mid frequencies dominate perception

Image Filter

Thank You…!