Image Processing

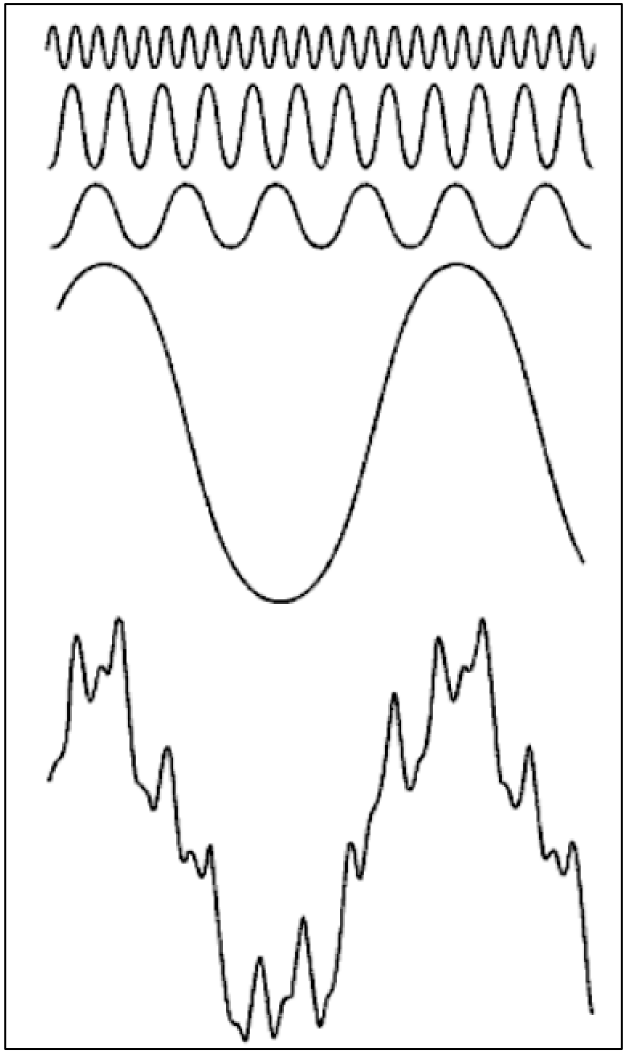
Traitement d’images

Yuliya Tarabalka
yuliya.tarabalka@hyperinet.eu
yuliya.tarabalka@gipsa-lab.grenoble-inp.fr
Tel. 04 76 82 62 68

2D Fourier Transform

Fourier series & Fourier transform

- **Fourier series:**
  - Any function that periodically repeats itself can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient
  - It does not matter how complicated the function is, as long as it is periodic and meets some mild mathematical conditions



The function at the bottom is the sum of the four functions above it

Fourier series & Fourier transform

- **Fourier transform:**
  - Non-periodic functions (whose area under the curve is finite) can be expressed as the integral of sines and/or cosines multiplied by a weighing function

$$h(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\omega)[\cos(\omega t) + j\sin(\omega t)]d\omega$$

  - Its utility is greater than the Fourier series in most practical problems

- **Important property:**
  - A function, expressed in either Fourier series or Fourier transform can be reconstructed (recovered) completely via an inverse process, with no loss of information

Fourier transform and its inverse

- **1D: FT** $$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$ $$j = \sqrt{-1}$$

  FT^{-1} $$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

- **2D: FT** $$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-j2\pi(ux+vy)} dx dy$$

  FT^{-1} $$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) e^{j2\pi(ux+vy)} du dv$$

u and *v* are the ***frequency variables*** (determine the frequencies of the components of the transform)

Fourier transform and its inverse

- **2D DFT of a function f(x,y) of size M x N**

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

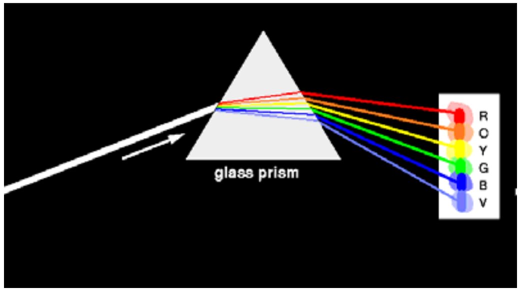
$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M + vy/N)}$$

- **Important property of the DFT:**
  - The discrete Fourier transform and its inverse always exist.
  - Thus, for digital image processing, existence of either the discrete transform or its inverse is not an issue.

Fourier transform

- **Analogy between the FT and a glass prism:**

  - Glass prism: separates light into various color components, each depending on its wavelength (or frequency) content



  - Fourier transform = “mathematical prism”: separates a function into various components, each depending on its frequency content

Discrete Fourier transform (TF = *fft2()*)

- **Magnitude (spectrum) of the FT:**

abs(TF)

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$

- **Phase angle (phase spectrum):**

angle(TF)

$$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

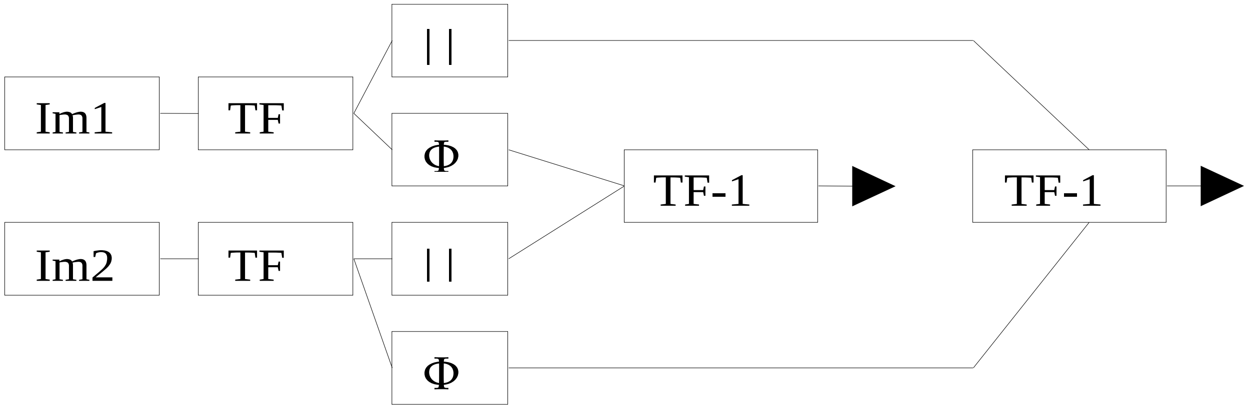
- **Power spectrum (spectral density):**

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

R(u, v) and *I(u, v)* are real and imaginary parts of *F(u, v)*, respectively

Discrete Fourier transform

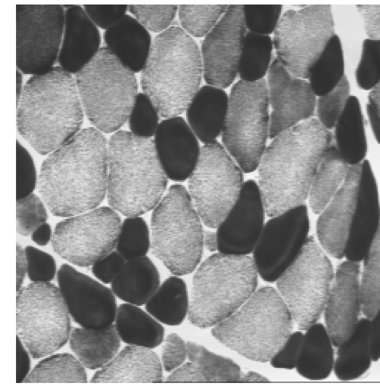
- **The phase angle of the FT of an image is more difficult to interpret than its magnitude, but it provides a lot of information**
- **We exchange magnitudes and phase angles between two images**



Discrete Fourier transform

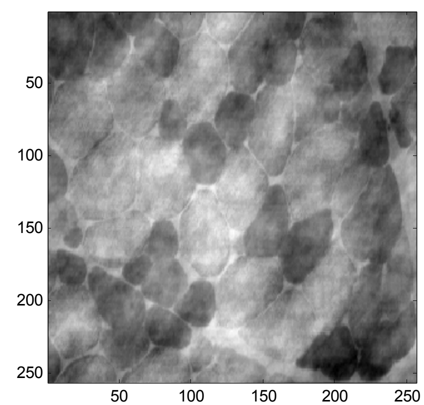
- **We exchange magnitudes and phase angles between two images**
- **In each image, we recognize information provided by the phase angle**

woman

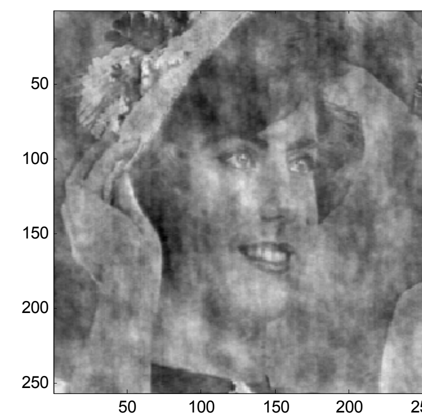


muscle

- **Another example: TF(sin) = Dirac**
  - ➢ Magnitude gives info about the sinus frequency
  - ➢ Phase angle gives its localization (dephasing)



|woman|+Φ(muscle)



|muscle|+Φ(woman)

Some important properties of 2D DFT

- **Centering the spectrum by**

$$\Im\left[f(x,y)(-1)^{x+y}\right] = F(u-M/2, v-N/2)$$

where $\Im$ **denotes the FT of the argument**

- **"Direct current" (DC) component of the spectrum is equal to the average gray level of the image:**

$$F(0,0) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$$

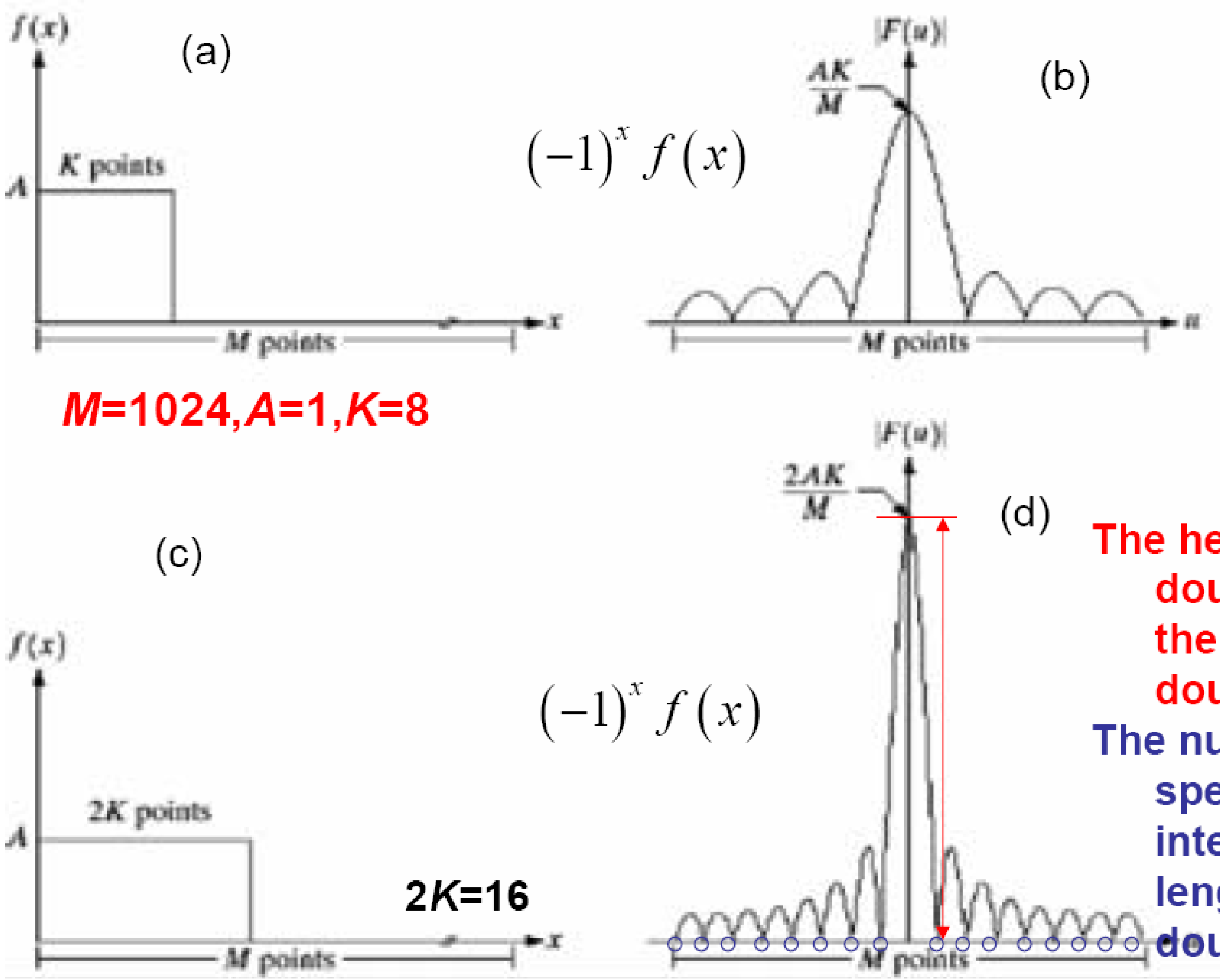
- **Conjugate symmetry: If f(x, y) is real,**

$$F(u,v) = F^*(-u,-v) \qquad |F(u,v)| = |F(-u,-v)|$$

- **Relations between samples in the spatial and frequency domains**

$$\Delta u = \frac{1}{M\Delta x} \qquad \Delta v = \frac{1}{N\Delta y}$$

Examples



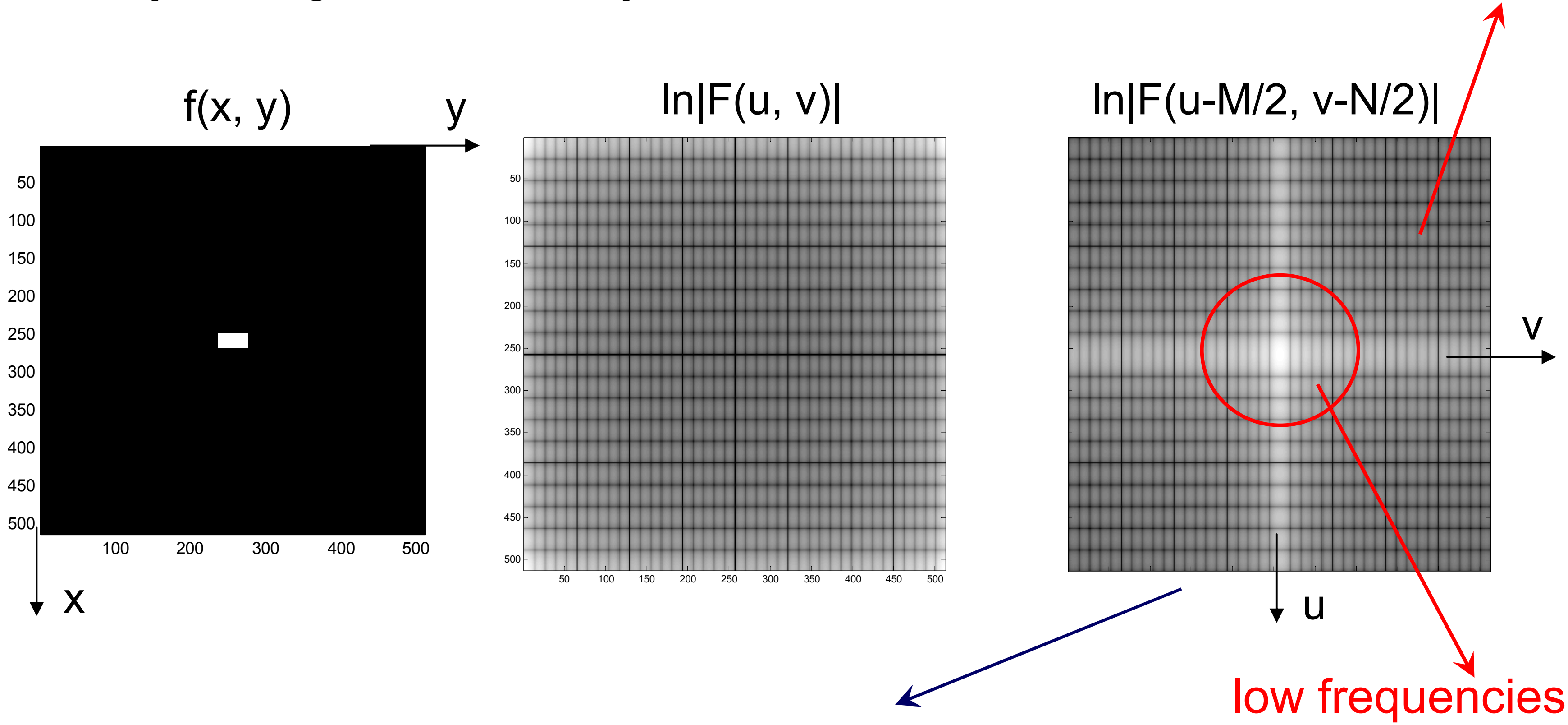
(a) a discrete function of *M* points, and (b) its Fourier spectrum. (c) a discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.

The height of the spectrum doubled as the area under the curve in the x-domain doubled

The number of zeros in the spectrum in the same interval doubled as the length of the function doubled.

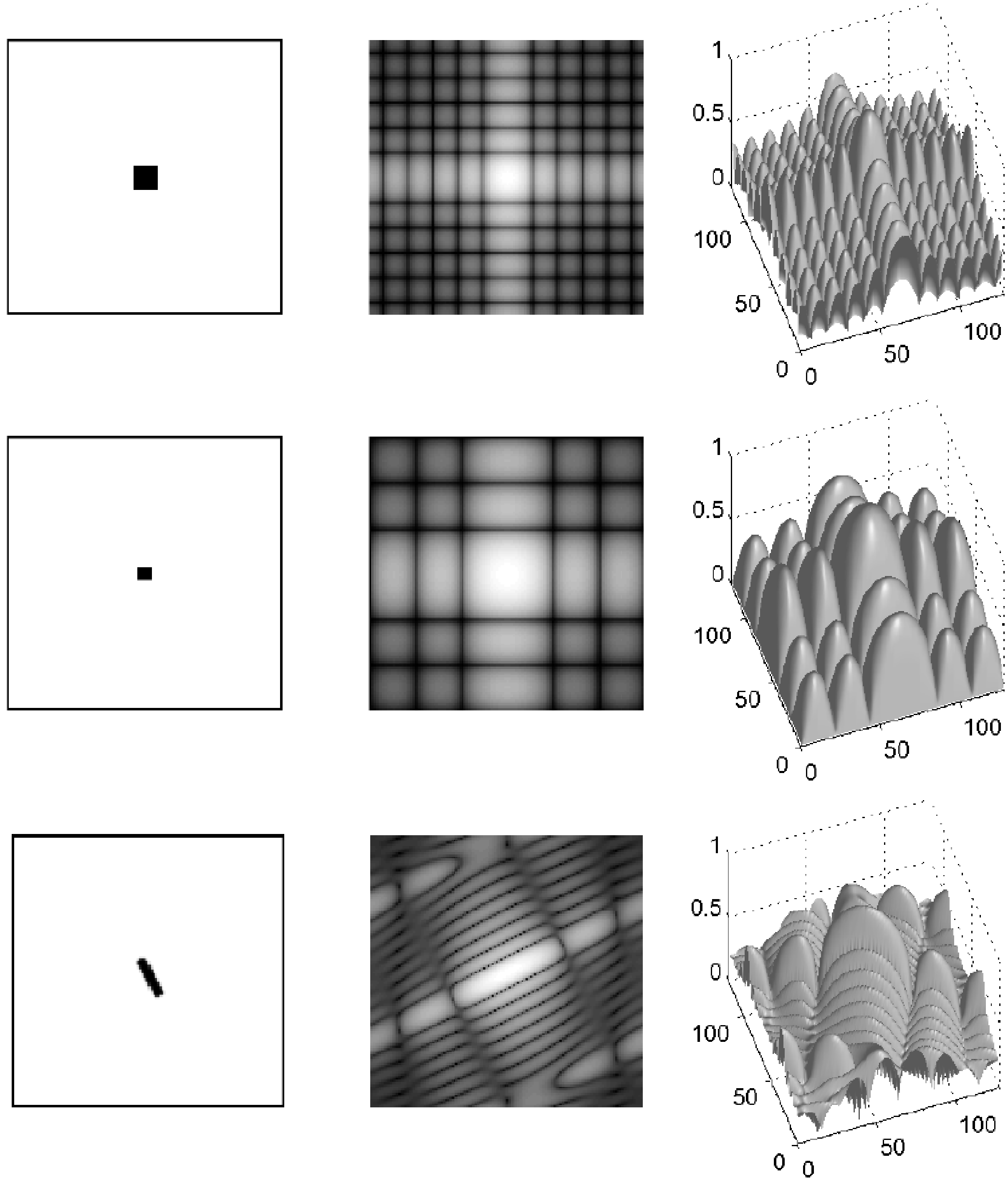
Example of centering 2D DFT

- The original image f(x, y) is the product of two rectangular functions: f(x, y) = f1(x)*f2(y)
- TF(rectangular function) = sinc

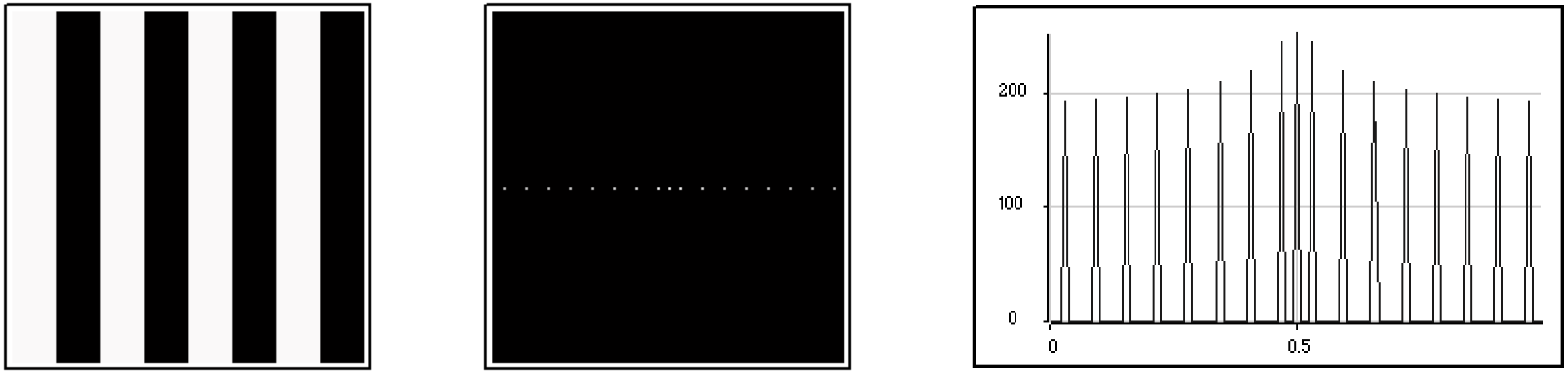


The form we often use for visualization

Examples

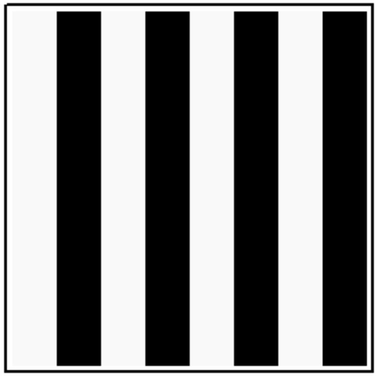
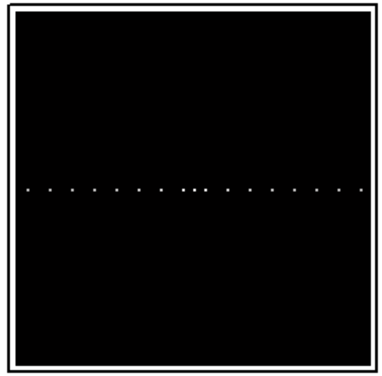
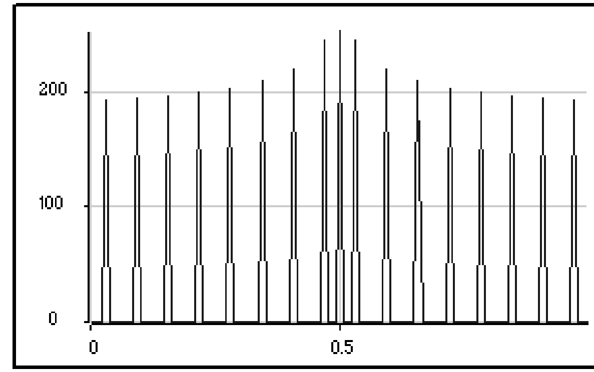


- **Since the image is periodic, the spectrum is discrete**

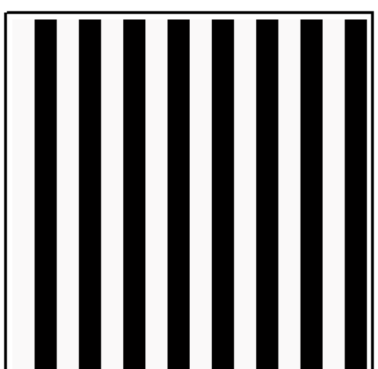
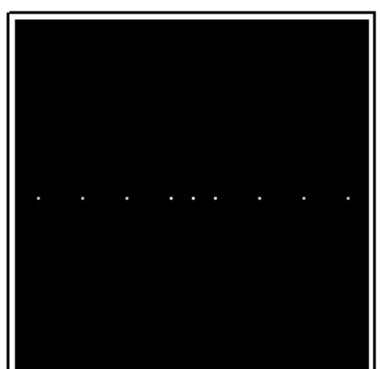
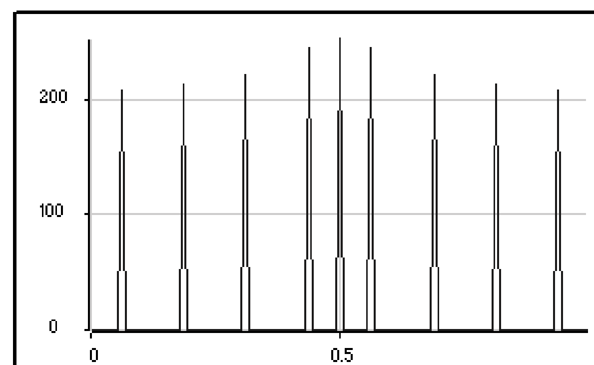


- **Translation: the last stripe is thinner. The spectrum is continuous**

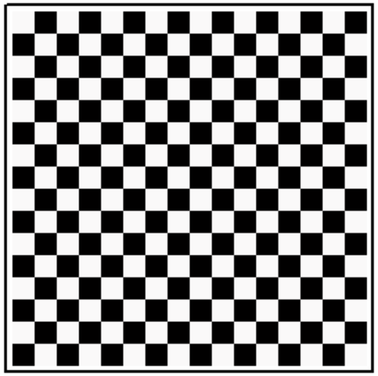
- **Since the image is periodic, the spectrum is discrete**

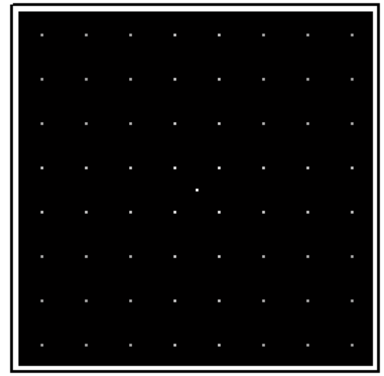




- **Duality: if the period of the signal increases, the distance between the frequencies decreases, and *vice versa***

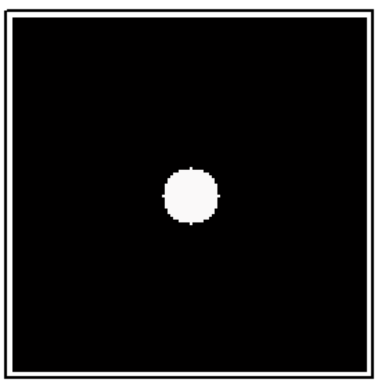




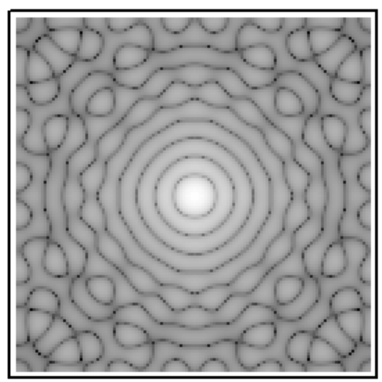
- **The 2D periodicity of the image induces the 2D periodicity of the FT**

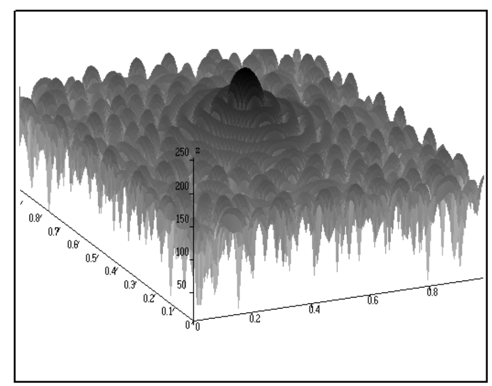




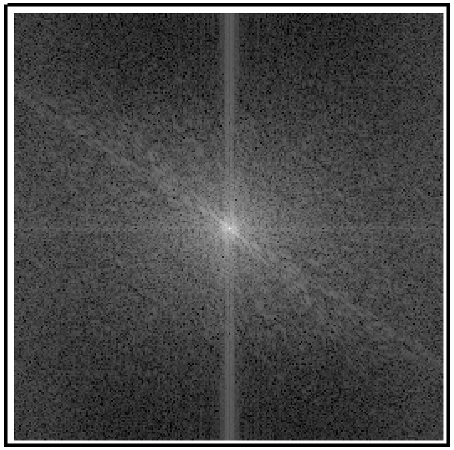
- **Disk & aliasing**





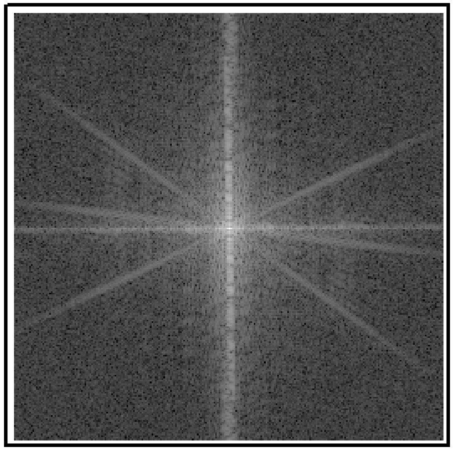


- **Interpretation of the TF of real images**



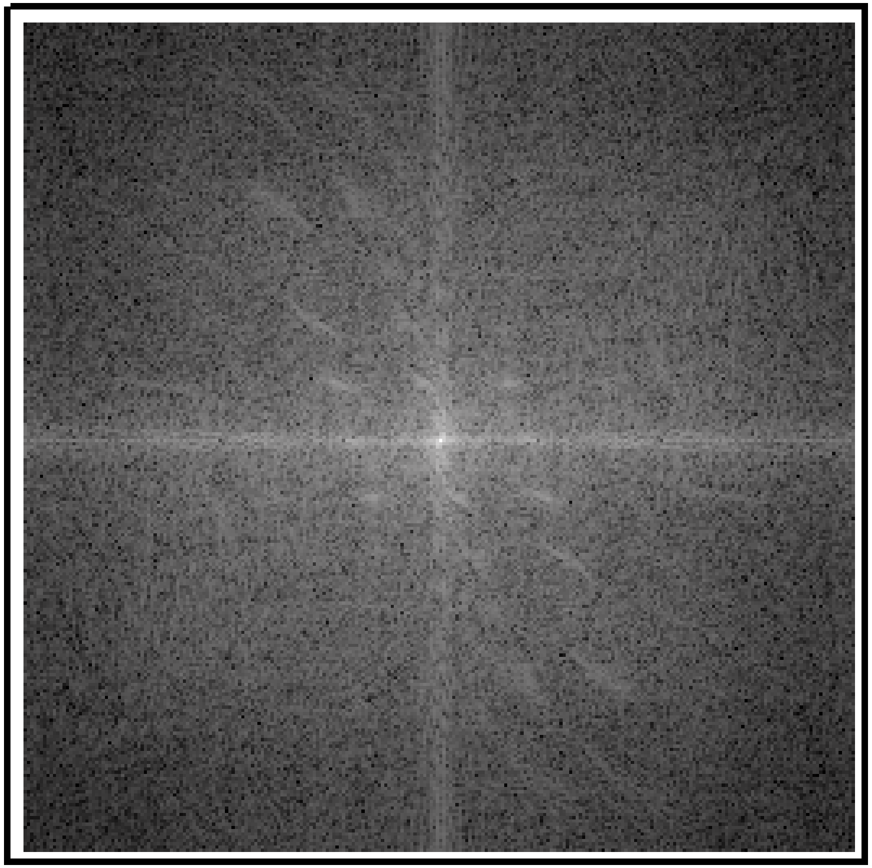
- ➢ Most of the information is concentrated in the low frequences and 3 lines
- ➢ Lines represent either discontinuities or edges, they are perpendicular to the discontinuity they present

- ➢ Horizontal and vertical lines are the result of the border effect
- ➢ Diagonal line is the result of the discontinuities induced by the hat



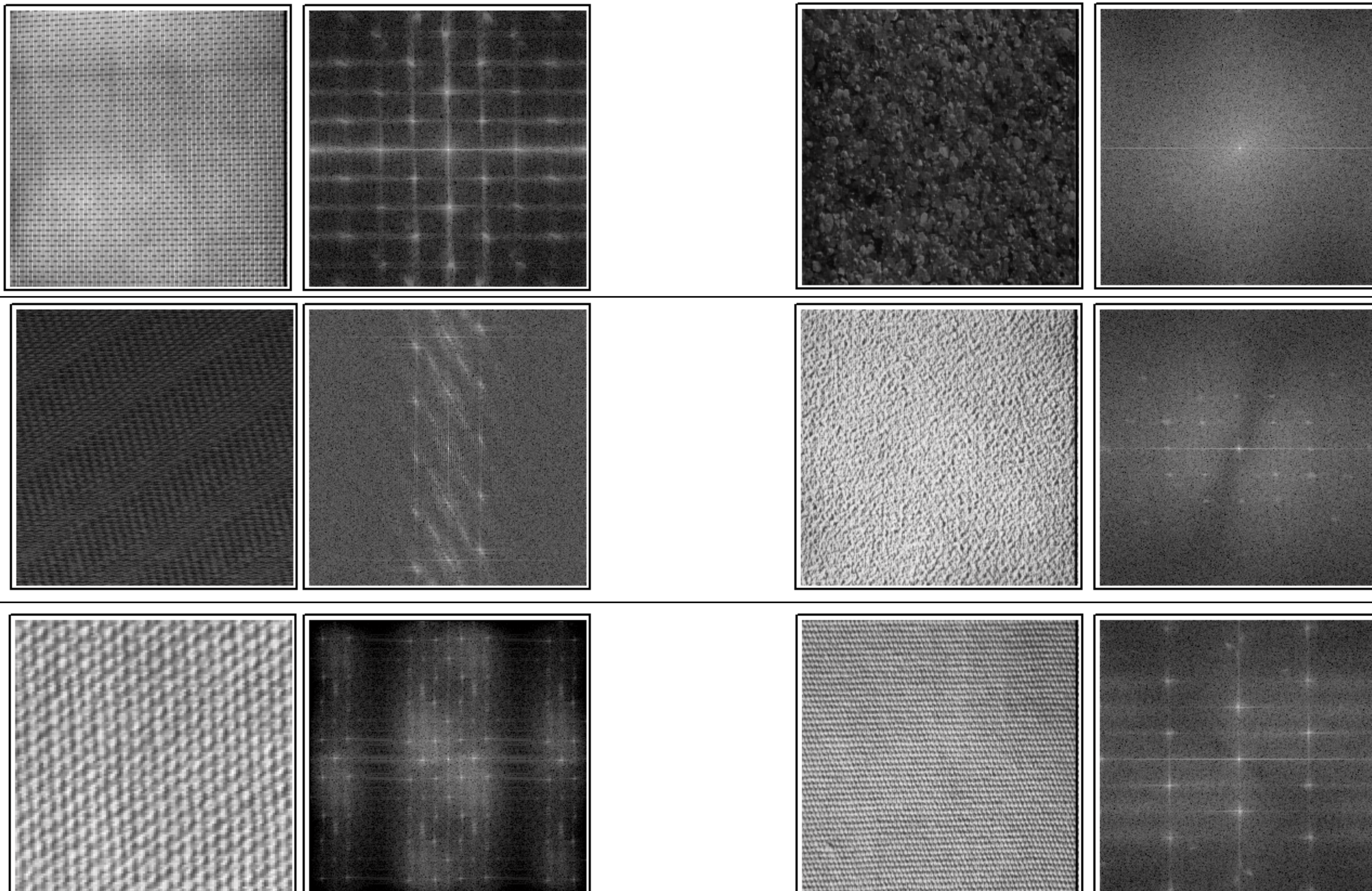
• Interpretation of the TF of real images

➢ Small lines correspond to the windows and flags



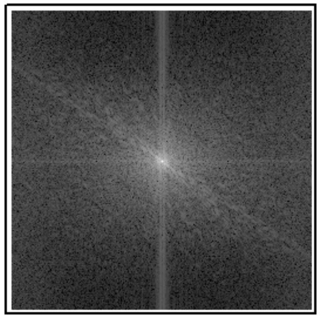
Textures = periodic patterns in an image

- **The Fourier spectrum is well suited for describing the directionality of textures**

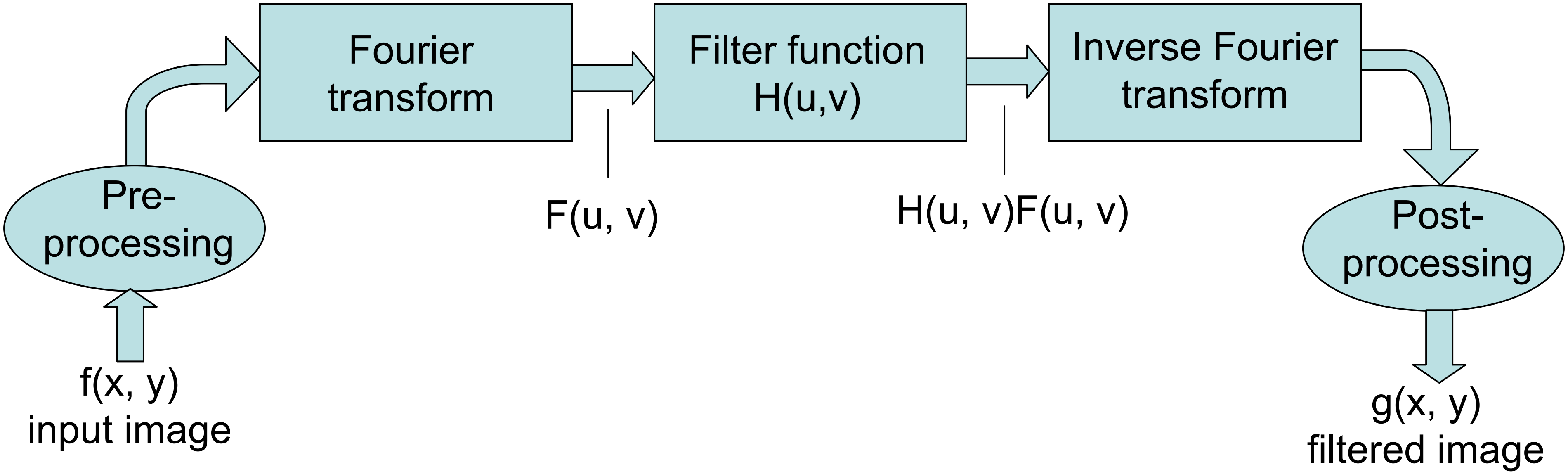


Filtering in the frequency domain

- **Basic properties of the frequency domain**
  - Each term of F(u, v) contains *all* values of f(x, y), modified by the values of the exponential terms
  - The slowest varying frequency component (u = v = 0) = the average gray level in an image
  - Low frequencies correspond to the slowly varying components of an image
  - High frequencies correspond to fast gray level changes in the image (edges, noise…)



Filtering in the frequency domain



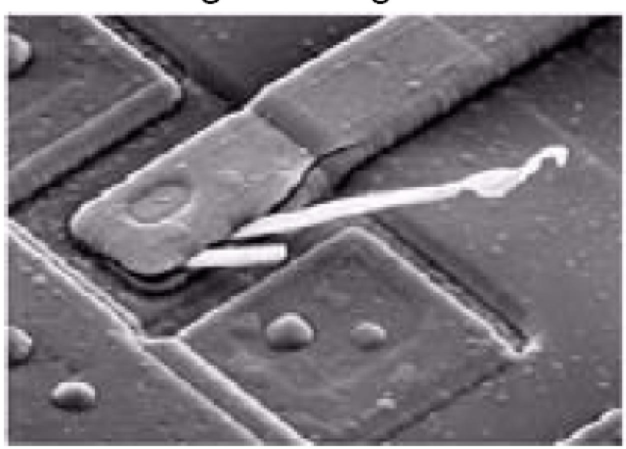
- ***Filter H(u, v)*** **supresses certain frequences in the transform while leaving others unchanged**
- **G(u, v) = H(u, v)F(u, v)**

Example of filtering

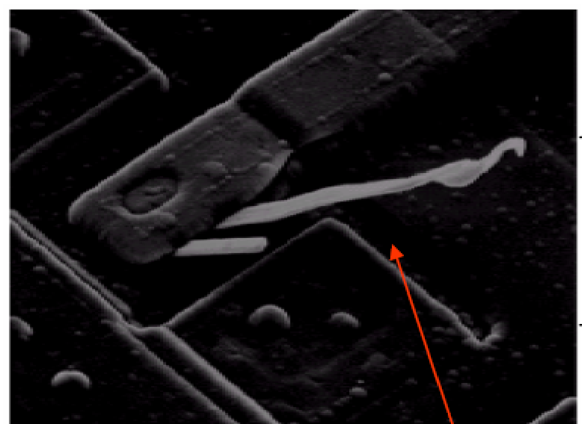
- **Notch filter** $H(u,v)=\begin{cases}0, & if\ (u,v)=(M/2,N/2)\\ 1, & otherwise\end{cases}$

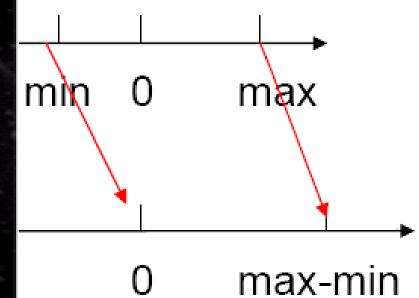
  - Frequency domain: Set the DC component to zero and leave all other frequency components unchanged
  - Spatial domain: Force the average value to zero

Original image



Result of filtering the image with a notch filter that set DC to 0.





Edges stand out

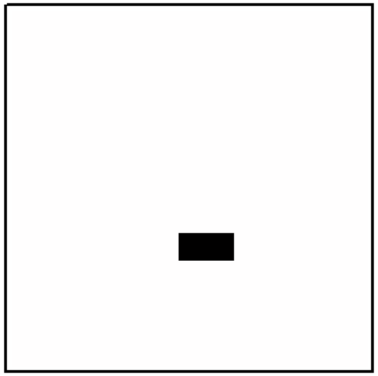
Correspondence between filtering in the spatial and frequency domains

- **Convolution theorem:**

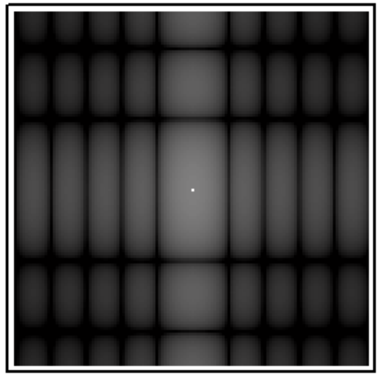
$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v)$$
$$f(x,y)h(x,y) \Leftrightarrow F(u,v) * H(u,v)$$

- **Given a filter in the frequency domain, we can obtain the corresponding filter in the spatial domain by taking the inverse FT of the former. The reverse is also true**
- **Filtering is often more intuitive in the frequency domain**
- **Whenever possible, it is preferable to filter in the spatial domain using small filter masks**

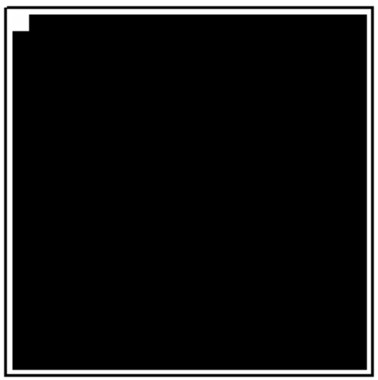
Correspondence between filtering in the spatial and frequency domains



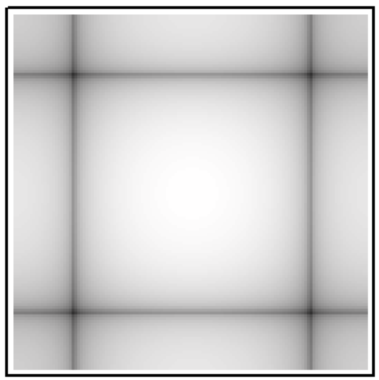
Original image



FT of the image

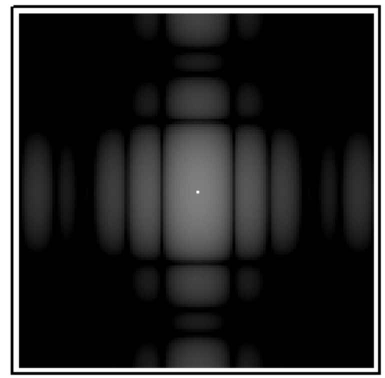


Mask

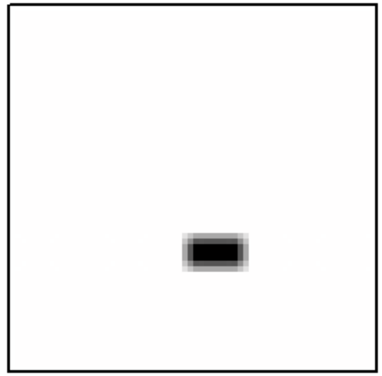


FT of the mask

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



FT(image)FT(mask)



(image)*(mask)

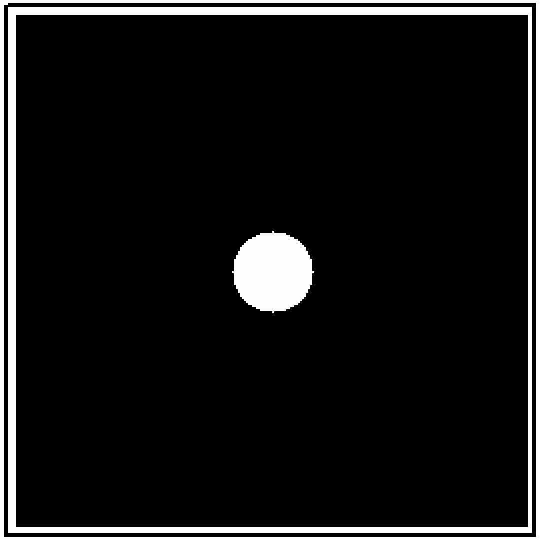
||

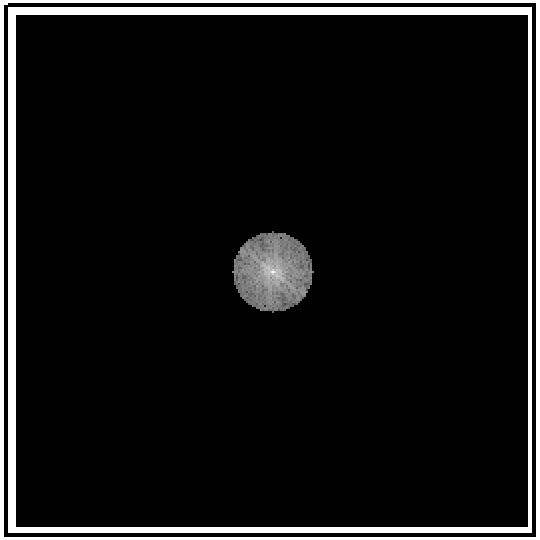
$FT^{-1}(FT(image)FT(mask))$

Ideal lowpass filter (smoothing)

- **The ideal lowpass filter cuts off all high-frequency components of the FT that are at a distance greater than a specified distance *D_0* (*cutoff frequency*) from the origin of the (centered) transform**

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



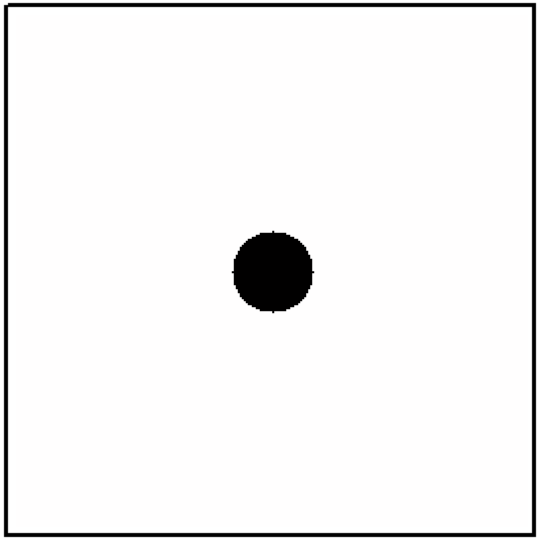


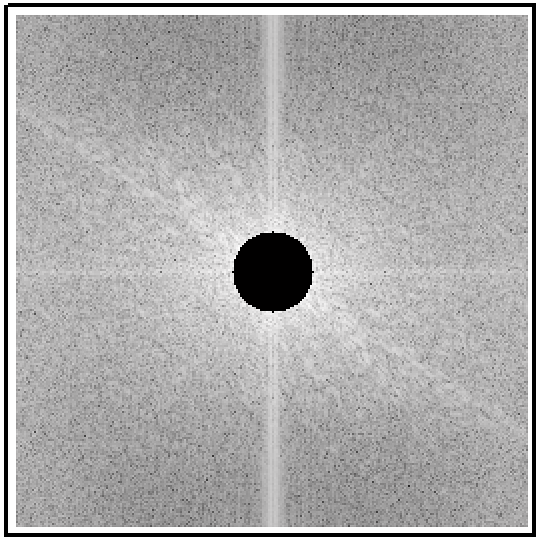
Blurring and ringing effects

Ideal highpass filter (sharpening)

- **The ideal highpass filter**

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

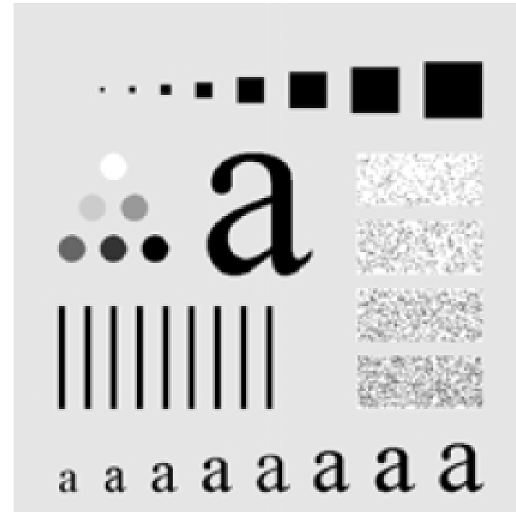




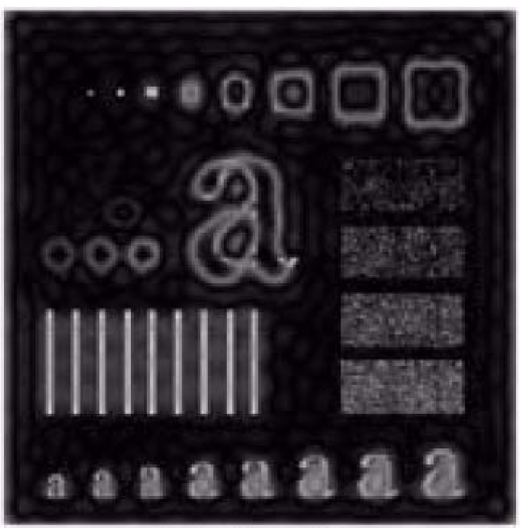
Only the edges appear

Ideal highpass filter (sharpening)

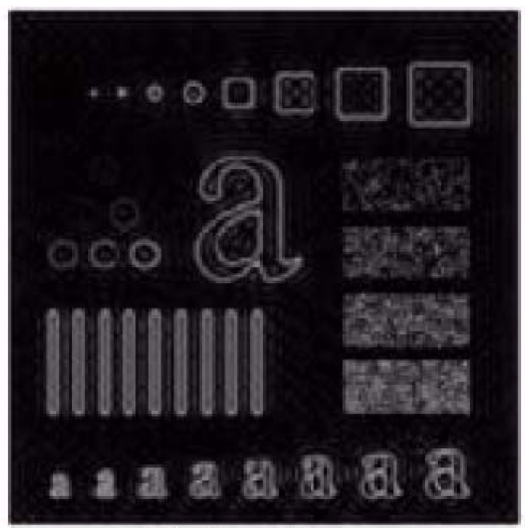
- **The ideal highpass filter**



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



(a) D_0=15



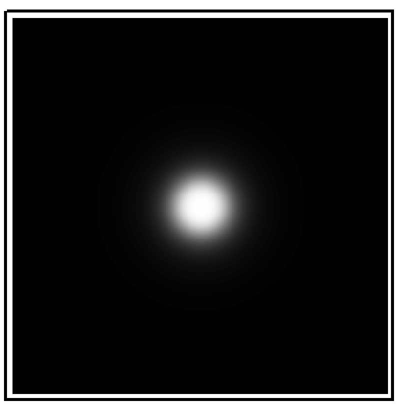
(b) D_0=30

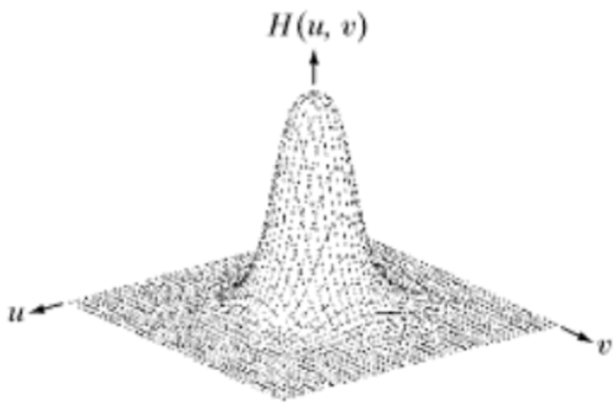
(c) D_0=80

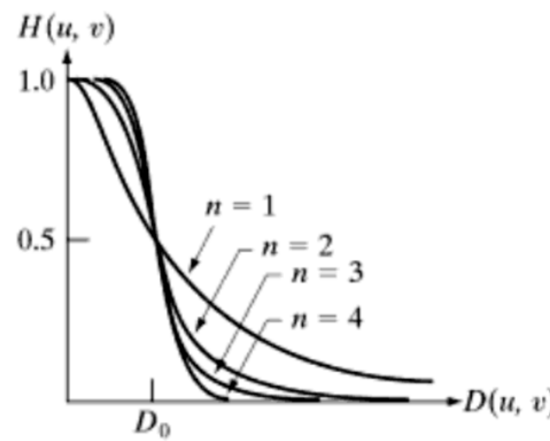
Butterworth lowpass filter

- **Helps to cope with the ringing effect**
- **The transfer function of a Butterworth lowpass filter of order *n*, and with cutoff frequency at a distance D_0 from the origin:**

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





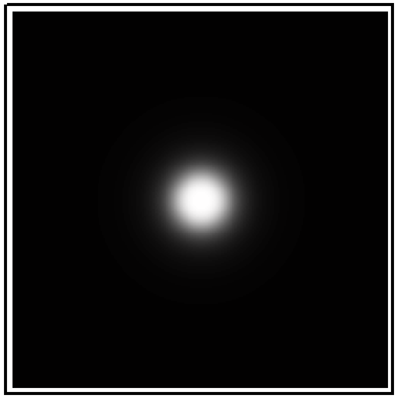


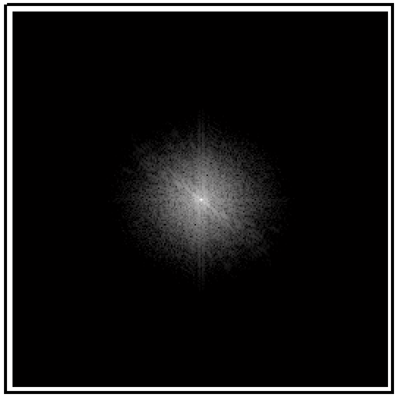
Butterworth lowpass filter

- **Helps to cope with the ringing effect**
- **The transfer function of a Butterworth lowpass filter of order *n*, and with cutoff frequency at a distance D_0 from the origin:**

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

- ***n* = 2 is a good compromise between effective lowpass filtering and acceptable ringing effects**





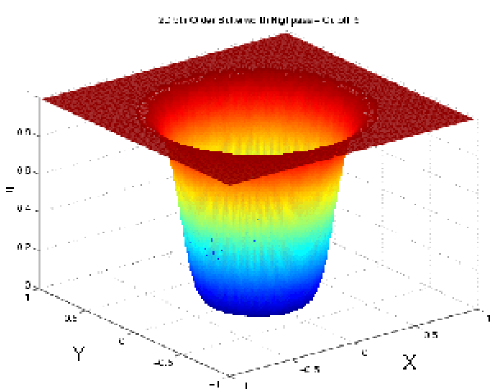
The essential part of the image is preserved (compression)

Butterworth highpass filter

- **The transfer function of a Butterworth highpass filter (BHPF) of order *n*, and with cutoff frequency at a distance D_0 from the origin**

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

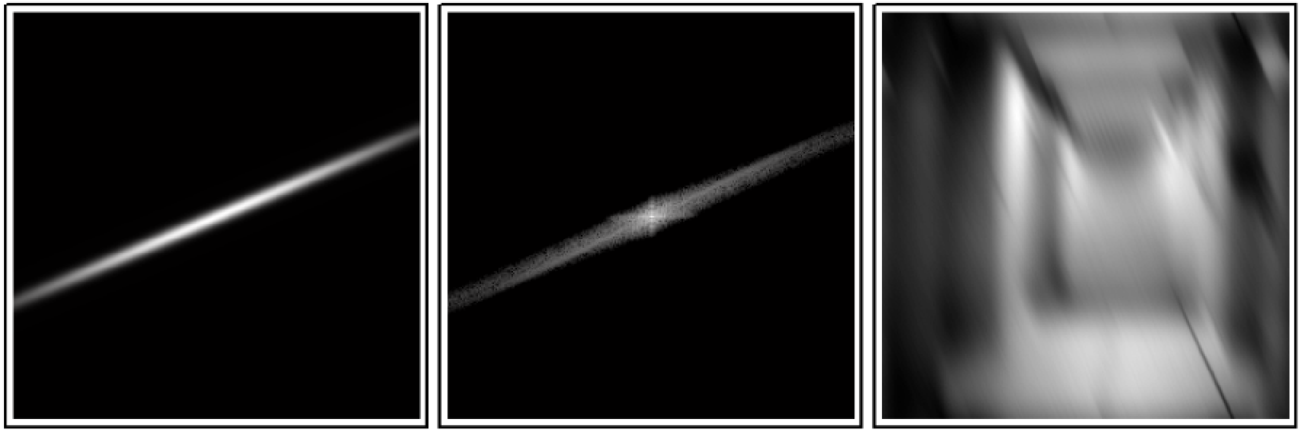
- **BHPFs behave smoother than ideal filters**




n=5, cutoff frequency = .5

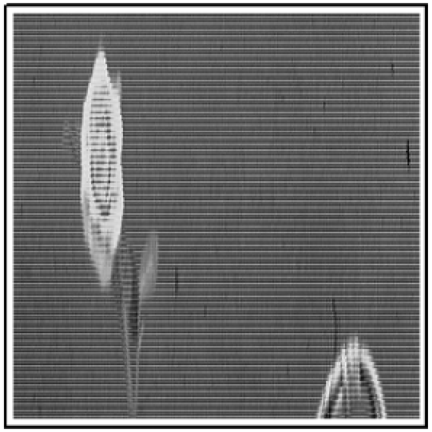
Gabor filtering

- **The Gabor filter is an ellipse with the Gaussian decrease**

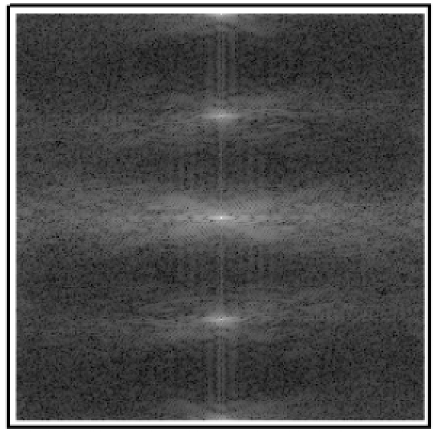


Gabor filtering

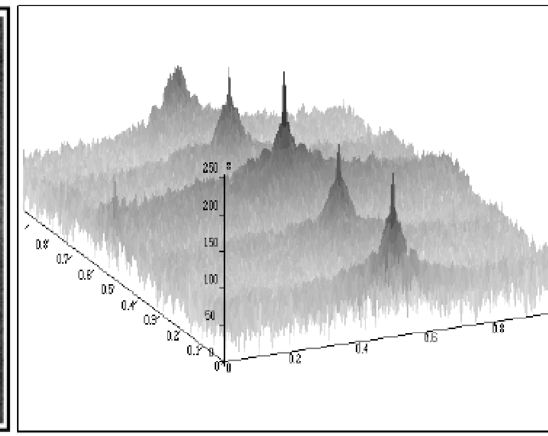
- **The Gabor filter is an ellipse with the Gaussian decrease**



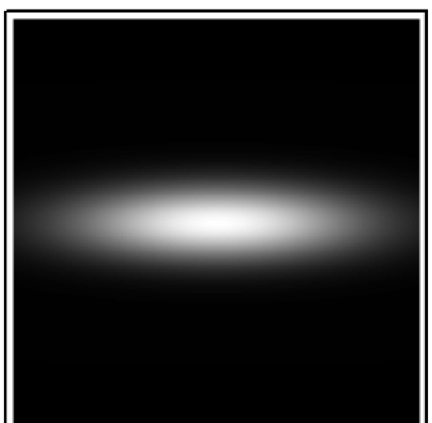
Original image



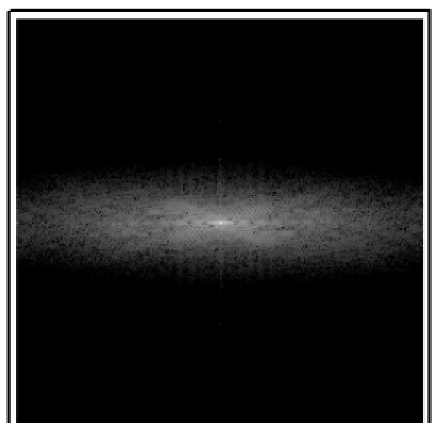
2D FT



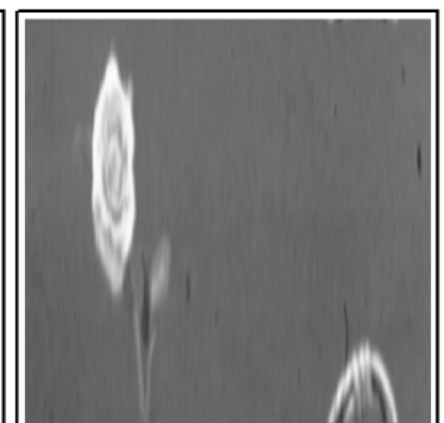
3D FT



Filtre



FT(final image)



Final image)