Master in Computer Vision Barcelona

Module: Introduction to Human and Computer Vision

Lecture 5: Space-frequency representation (I)

Lecturer: Javier Ruiz Hidalgo

Me

- Javier Ruiz Hidalgo
  - Email: j.ruiz@upc.edu
  - Office: UPC, Campus Nord, D5-008
- Teaching experience
  - Basic signal processing
  - Image processing & computer vision
  - M1, M4 and M6 (Computer Vision Master)
- Research experience
  - Master on hierarchical image representations by UEA (UK)
  - PhD on video coding by UPC (Spain)
  - Interests in image & video coding, 3D analysis and super-resolution

Outline

- Introduction
  - Image model definition
  - Convolution
  - Correlation

- 2D Fourier Analysis
  - Definition, properties and basic transforms

- 2D Discrete Fourier Analysis
  - Definition, properties and basic transforms

Outline

- Introduction
  - Image model definition
  - Convolution
  - Correlation

- 2D Fourier Analysis
  - Definition, properties and basic transforms

- 2D Discrete Fourier Analysis
  - Definition, properties and basic transforms

Image model definition

- In the **space/frequency image model**, the image is understood as a **linear combination** of simpler functions; typically:
  - Impulse functions, as in the canonical representation.
  - Complex exponentials, as in the Fourier representation.

- One of the most natural operations are **Linear** and **Space-Invariant** operations. They linearly combine the pixel values in a given neighborhood of the pixel being processed (**impulse response, convolution mask, kernel**).

- Linear Space-Invariant (**LSI**) operators can be defined:
  - In the original (space) domain, through a **convolution**.
  - In the transformed (frequency) domain, through a **product**.

Canonical representation

- Unit Impulse

$$\delta[m,n] = \begin{cases} 1 & m = n = 0 \\ 0 & \text{other} \end{cases}$$

$$\delta[m,n] = \delta[m] \cdot \delta[n]$$

separable

- Linear decomposition
  - Every sequence can be expressed as a linear combination of displaced unit impulses

$$x[m,n] = \sum_{m'=-\infty}^{\infty} \sum_{n'=-\infty}^{\infty} x[m',n']\, \delta[m-m',n-n']$$

$$\begin{array}{ccc} 3 & 2 & 1 \\ 6 & 4 & 7 \\ \vdots & & \end{array} = 3 \cdot \begin{array}{cccc} 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \\ \vdots & & & \end{array} + 2 \cdot \begin{array}{cccc} 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & \\ \vdots & & & \end{array} + 1 \cdot \ldots$$

2D impulse response

- On a **linear** system:

$$y[m,n] = T\left\{x[m,n]\right\} = T\left\{\sum_{m'}\sum_{n'} x[m',n']\cdot\delta[m-m',n-n']\right\} =$$

$$= \sum_{m'}\sum_{n'} x[m',n']\cdot T\left\{\delta[m-m',n-n']\right\}$$

- If the system is **space-invariant**:

$$h[m,n] = T\left\{\delta[m,n]\right\}$$

Impulse response

$$T\left\{\delta[m-m',n-n']\right\} = h[m-m',n-n']$$

Image convolution (1)

- A Linear Space-Invariant operator can be defined by its **impulse response** $h[m, n]$:

$$x[m,n] \longrightarrow \boxed{h[m,n]} \longrightarrow y[m,n]$$

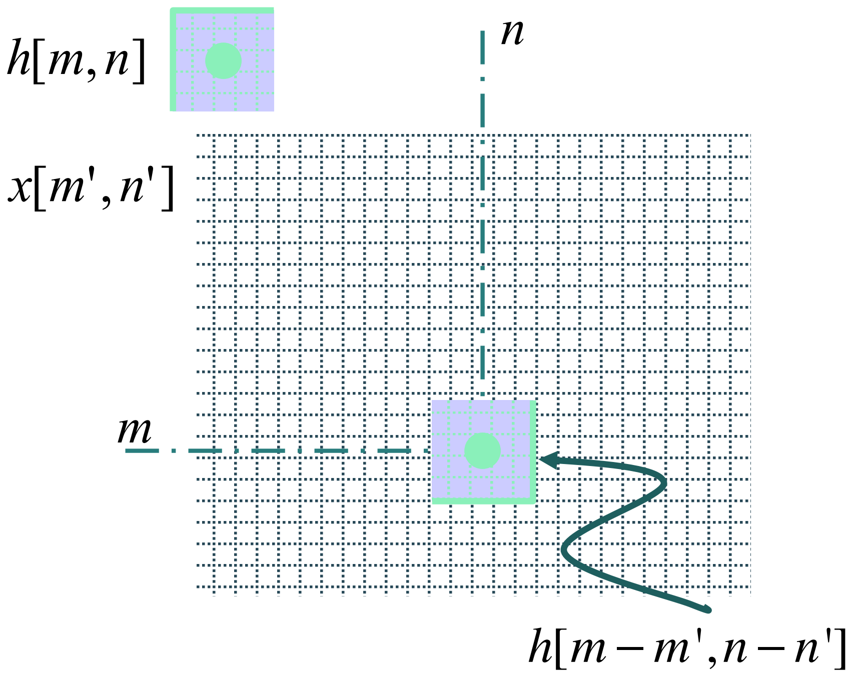
and the output is computed by the **convolution** equation:

$$y[m,n] = x[m,n] * h[m,n] = \sum_{m'} \sum_{n'} x[m',n']h[m-m',n-n']$$

Image convolution (2)

- For each pixel position (*m*, *n*):
  1. The **reflected version** of *h*[*m′*,*n′*] is shifted at position (*h*[*m-m′*,*n-n′*])
  2. the output *y*[*m*,*n*] is the sum of the pixel values included in the neighborhood defined by the filter and weighted by the co-located filter values (*h*[*m*,*n*] is the **filter impulse response**)

$$y[m,n] = \sum_{m'} \sum_{n'} x[m',n']h[m-m',n-n']$$



- A **similar interpretation** where the input signal is shifted is also possible

$$y[m,n] = \sum_{m'} \sum_{n'} x[m-m',n-n']h[m',n']$$

Image convolution (3)

- The result of the convolution of an image of size *M* x *N* and a filter with impulse response of size H_1 x H_2 is an image of size (M+H_1-1) x (N+H_2-1)
- Nevertheless, in almost all practical cases, the output image support is **finally restricted** to the original image size.

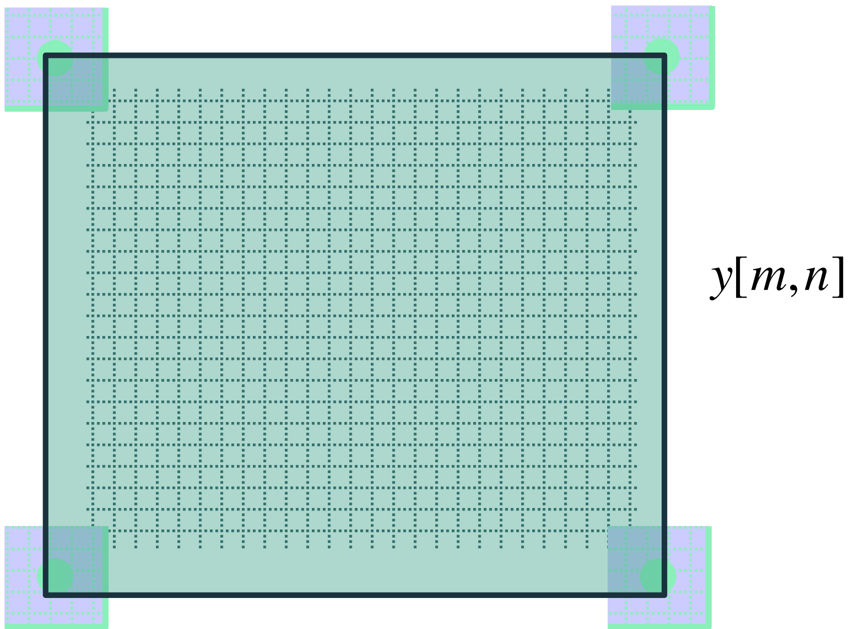
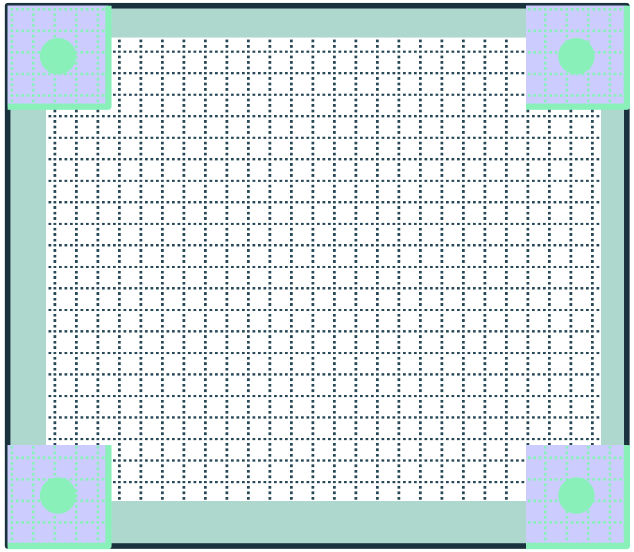


Image padding

- The computation of the output values close to the image border requires the use of pixel values outside the input image support.
- These values are commonly introduced by:
  - **Zero padding**: Including a frame of zeros in the signal to be processed (may introduce discontinuities in the signal).
  - **Mirroring**: Including pixel values that are a symmetric mirroring of the existing ones.
  - Implement a **space variant** filter to only use known pixel values.



Separability

- In some cases, the 2D impulse response can be defined as the product of two 1D impulse responses with different spatial indexes (**separability**)

$$h[m,n] = h_1[m] \cdot h_2[n]$$

- Separability allows for **faster implementations**

$$y[m,n] = \sum_{m'} \sum_{n'} x[m-m',n-n']h[m',n'] = \sum_{m'} \sum_{n'} x[m-m',n-n']h_1[m']h_2[n'] =$$
$$= \sum_{n'} h_2[n'] \sum_{m'} x[m-m',n-n']h_1[m'] = x[m,n] *^{1D}_{rows} h_1[m] *^{1D}_{columns} h_2[n]$$

Outline

- **Introduction**
  - Image model definition
  - Convolution
  - **Correlation**
- 2D Fourier Analysis
  - Definition, properties and basic transforms
- 2D Discrete Fourier Analysis
  - Definition, properties and basic transforms

Overview of correlation between 1D signals

- Distance between x[n] and y[n-m]:

$$D[m] = \sum_{n=-\infty}^{\infty} \left| x[n] - y[n-m] \right|^2 =$$

$$= \underbrace{\sum_{n=-\infty}^{\infty} \left| x[n] \right|^2}_{E_x} + \underbrace{\sum_{n=-\infty}^{\infty} \left| y[n-m] \right|^2}_{E_y} - \underbrace{\sum_{n=-\infty}^{\infty} x[n] y^*[n-m]}_{x[m] * y^*[-m]} - \underbrace{\sum_{n=-\infty}^{\infty} x^*[n] y[n-m]}_{x^*[m] * y[-m]}$$

- Cross-correlation between x[n] and y[n]

$$\boxed{r_{xy}[m] \equiv \sum_{n=-\infty}^{\infty} x[n] y^*[n-m] = x[m] * y^*[-m]}$$

therefore, $D[m] = E_x + E_y - (r_{xy}[m] + r_{xy}^*[m])$

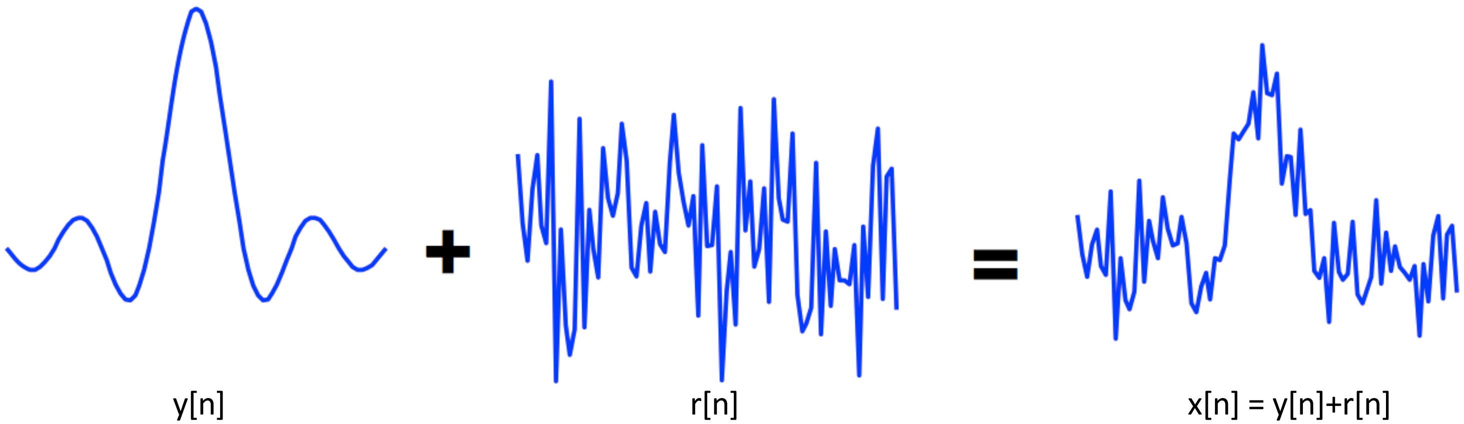
if E_x and E_y are constant: minimum distance $\Leftrightarrow$ maximum correlation

Matched filtering

- We can interpret the correlation as a filter with impulse response h[n]=y*[-n] (**matched filter**)

$$r_{xy}[m] \equiv x[m] * y^*[-m] = x[m] * h[m]$$

- Used in binary detection systems:
  - y[n] is a known signal
  - x[n] is the test signal (contaminated with Gaussian noise x[n] = y[n] + r[n])
  - The matched filter maximizes the SNR at m=0 ($r_{xy}[0]$)



Correlation of images

- Cross-correlation:

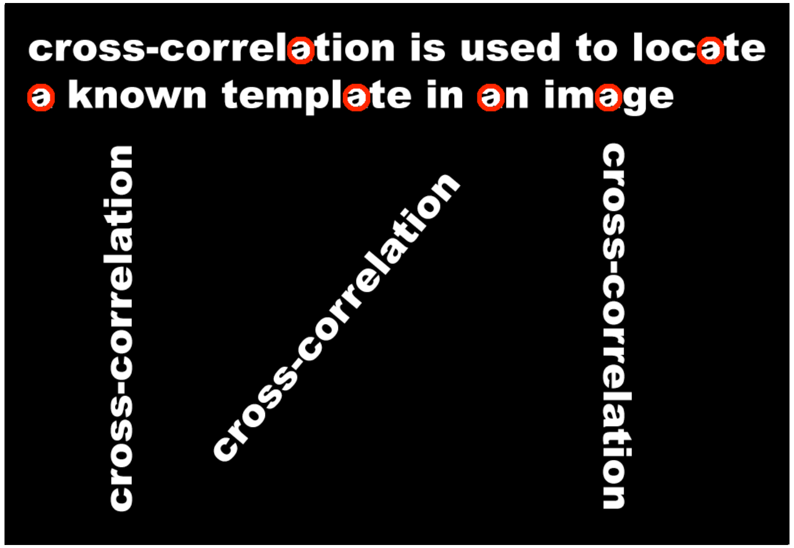
$$r_{xy}[m,n] = x[m,n] * y^*[-m,-n] = \sum_{m'=-\infty}^{\infty} \sum_{n'=\infty}^{\infty} x[m',n']\, y^*[m'-m,n'-n]$$

- Auto-correlation:

$$r_{xx}[m,n] = x[m,n] * x^*[-m,-n] = \sum_{m'=-\infty}^{\infty} \sum_{n'=\infty}^{\infty} x[m',n']\, x^*[m'-m,n'-n]$$

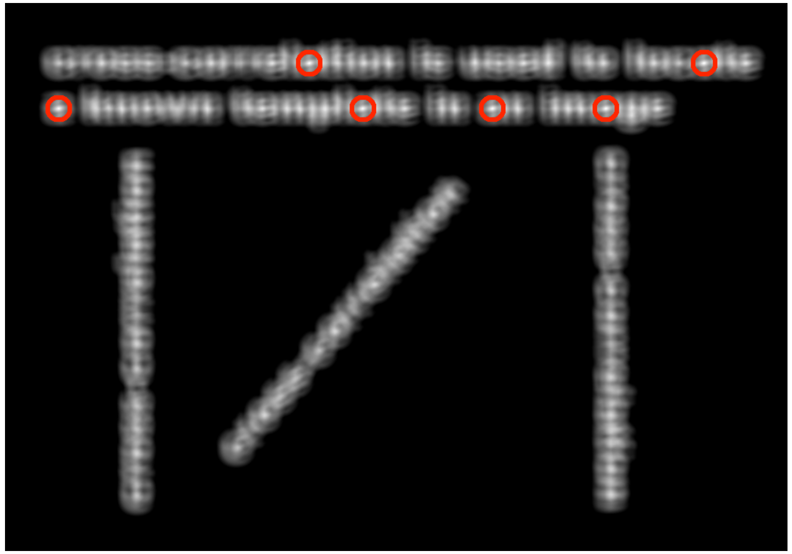
- Properties:
  - Energy: $E_x = r_x[0,0] = \sum_{n=-\infty}^{\infty} x[m,n]x^*[m-0,n-0] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |x[m,n]|^2$
  - Maximum: $|r_x[m,n]| \leq r_x[0,0] = E_x$
  - Symmetry: $r_x[m,n] = r_x^*[-m,-n]$

Application: Template matching (1)



x[m,n]

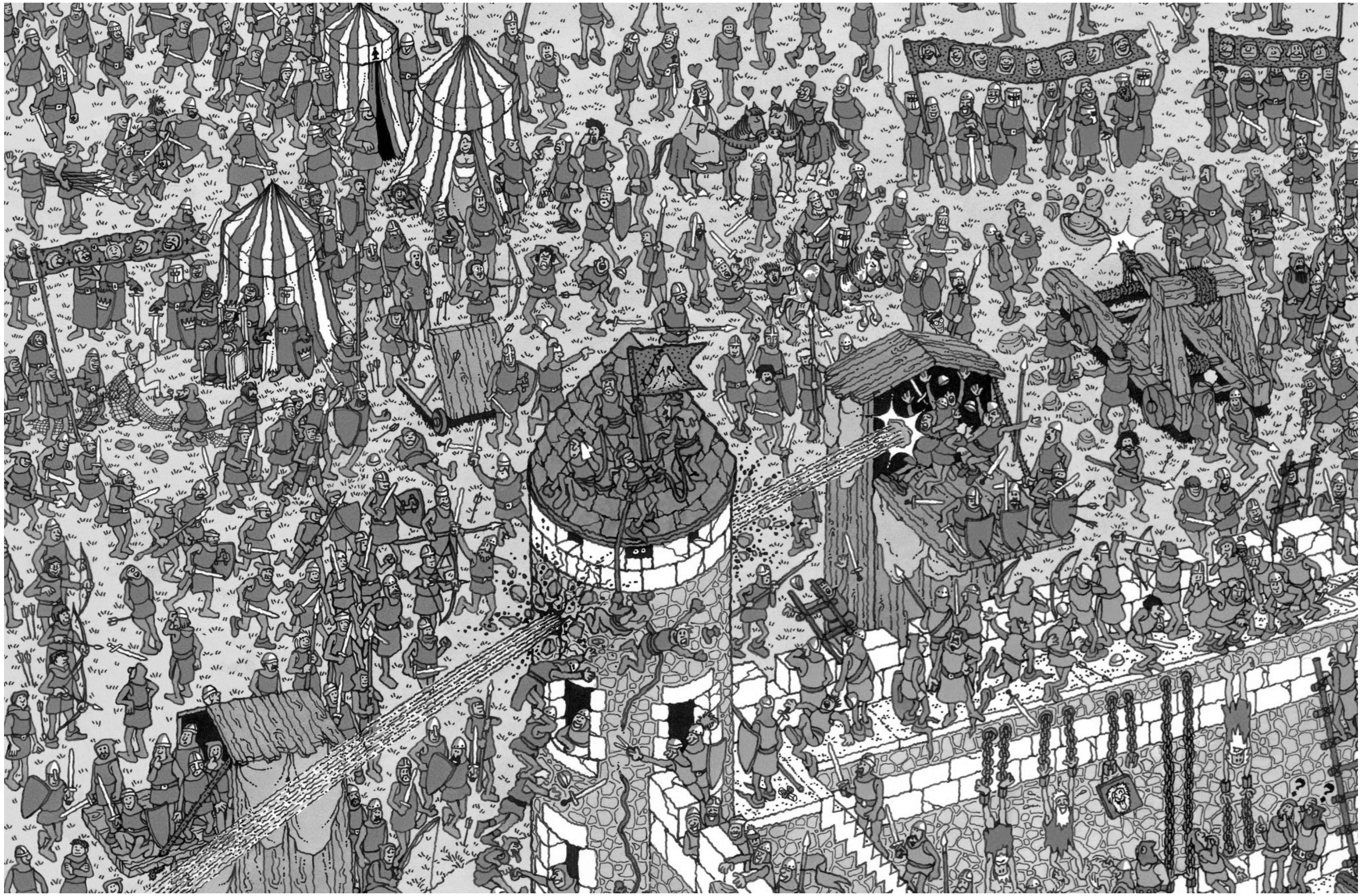
t[m,n]



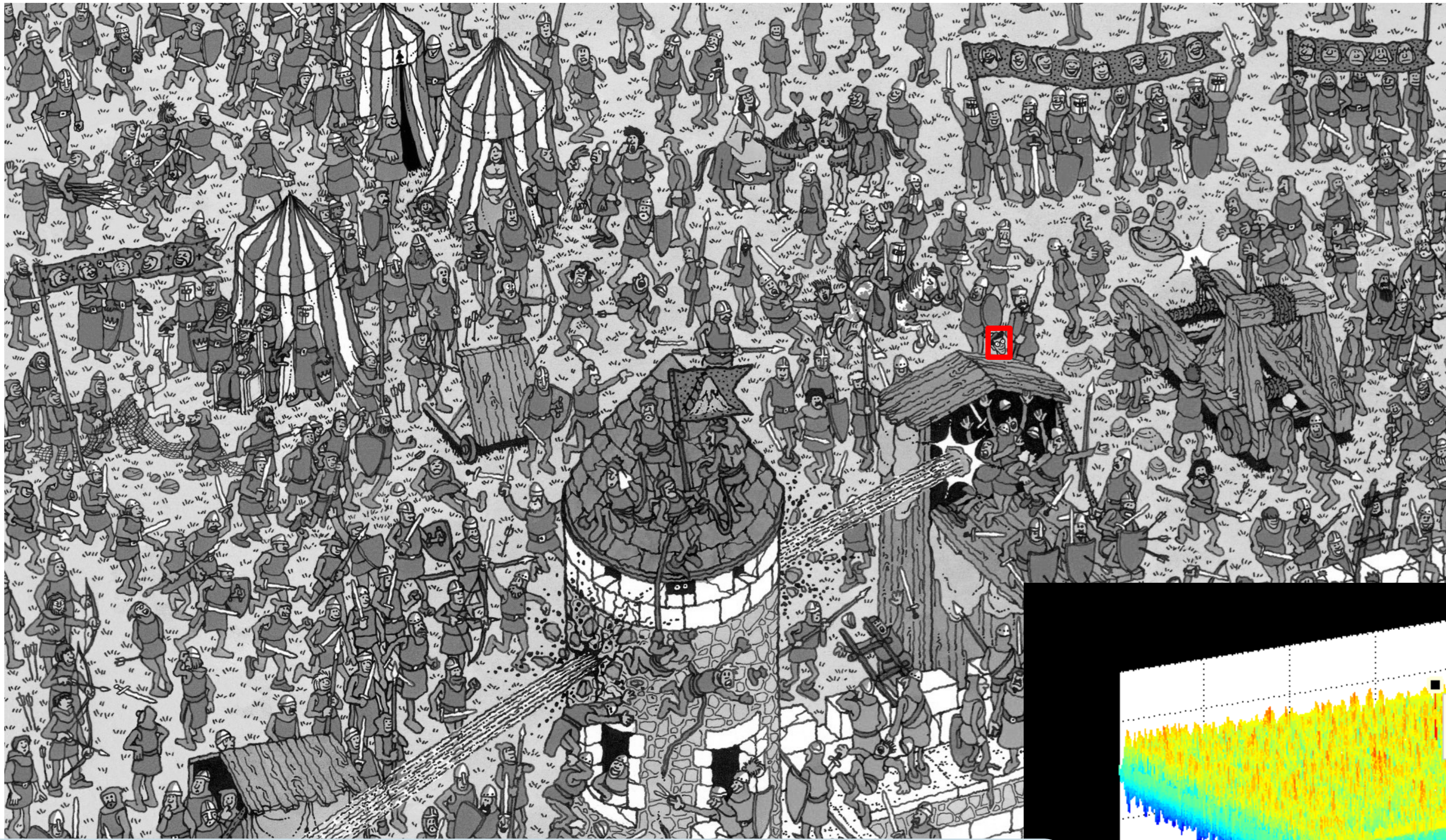
$r_{xt}[m,n]$

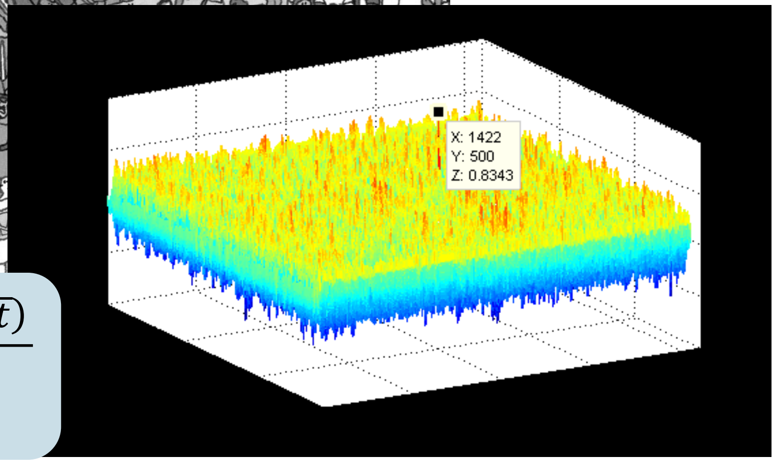
- Locate known objects or regions in image x[m,n] could be performed using cross-correlation with a **template** t[m,n]
- If there is a match the correlation will be maximum at the position where t[m,n] appears in x[m,n]
- There is a need of pre-processing (scale, align, etc.)

Application: Template matching (2)



Application: Template matching (3)





$$r_{xy}[m,n] = \frac{\sum_{m'=-\infty}^{\infty}\sum_{n'=-\infty}^{\infty}\left(x[m',n'] - \overline{x_{m,n}}\right)\left(t[m'-m,n'-n] - \bar{t}\right)}{\sqrt{E_{x[m',n']-\overline{x_{m,n}}}\ E_{t[m',n']-\bar{t}}}}$$

Application: Motion estimation (1)

- Motion estimation between two images (reference and actual)
- Block-Matching
  - Actual image is divided into blocks
  - Motion is estimated independently for each block
  - Translational motion
  - Luminance changes are assumed to be caused by motion (no illumination changes, noise, etc.)

Application: Motion estimation (2)

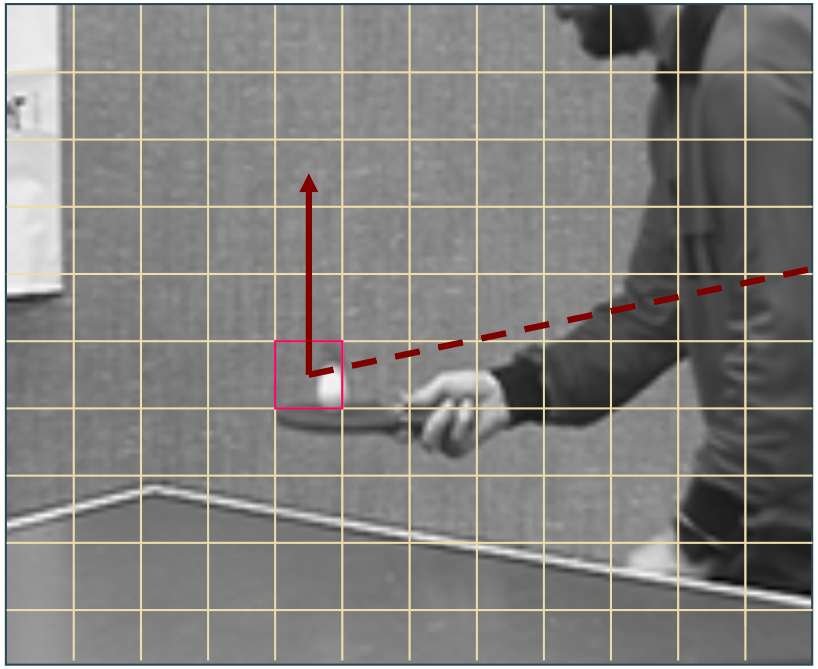
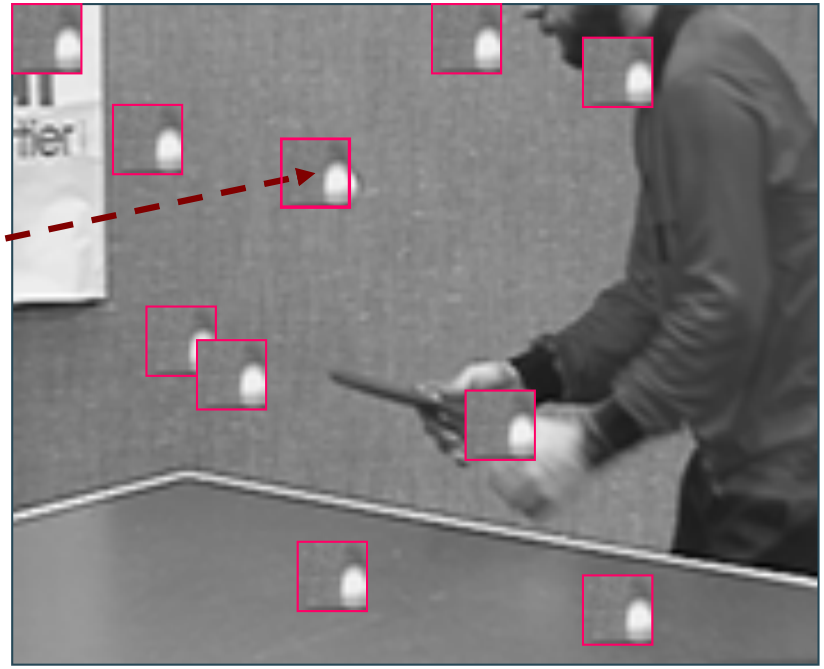


Image to estimate
(actual)

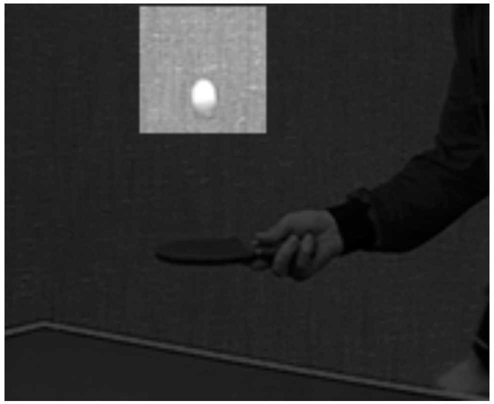


reference

Application: Motion estimation (3)

- Possible searching algorithm → **Correlation**
  - A search window is used to reduce the computational complexity (of doing a correlation between the block and the entire image for all blocks)
- The correlation maximum locates the most similar block
  - There is a need to **normalize the energy** of the blocks

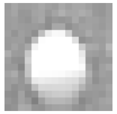
Image to estimate



reference

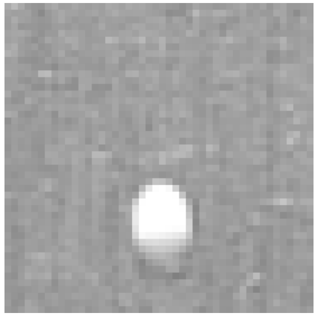
Application: Motion estimation (4)

b[m,n]



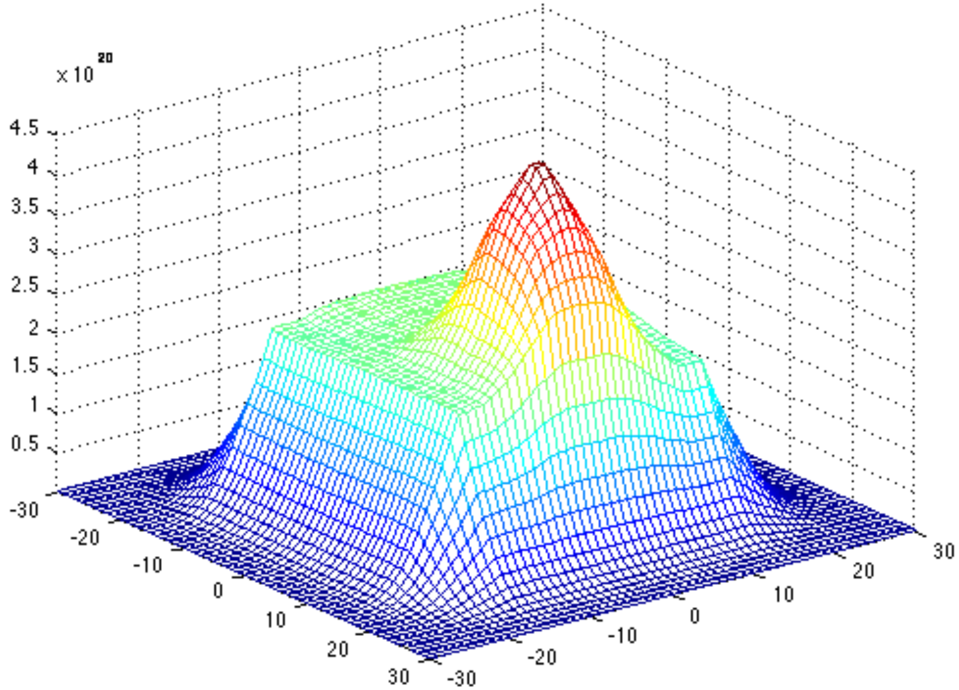
block (16x16)

a[m,n]



Window search (46x46)

$$r_{ab}[m,n] = a[m,n] * b^*[-m,-n]$$



Correlation (61x61)

Outline

- Introduction
  - Image model definition
  - Convolution
  - Correlation

- **2D Fourier Analysis**
  - **Definition, properties and basic transforms**

- 2D Discrete Fourier Analysis
  - Definition, properties and basic transforms

Space / Frequency image processing tools

- If LSI operators involve large impulse responses, they can be **efficiently implemented** thanks to the use of fast transforms:
  1. The input image is (fast) transformed (FFT).
  2. The operation is performed in the transformed domain by a product.
  3. The output image is obtained through an inverse FFT.

- Space/Frequency image processing tools allow (among others):
  - Convolution operations
  - Linear filter design
  - Analysis of sampling
  - Multi-resolution analysis:
    - Decomposition of the input image into multiple images containing the information at different 2D frequency bands

2D Fourier transform

- Definition:
  - The Fourier Transform (**FT**) of an image x[m,n] is defined as:

$$X\left(F_x,F_y\right)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}x[m,n]\,e^{-j\left(m2\pi F_x+n2\pi F_y\right)}\qquad -\frac{1}{2}\leq\left\{F_x,F_y\right\}<\frac{1}{2}$$

  - Whereas the Inverse Fourier Transform (**IFT**) is defined as:

$$x[m,n]=\int_{F_x=-1/2}^{1/2}\int_{F_y=-1/2}^{1/2}X\left(F_x,F_y\right)e^{j\left(m2\pi F_x+n2\pi F_y\right)}\,dF_x dF_y$$

F_x and F_y correspond to the horizontal and vertical frequencies respectively

FT properties (1)

- Usually samples in the spatial domain x[m,n] are real (or integer) whereas **samples in the frequency domain are complex**:

$$X(F_x,F_y) = X_R(F_x,F_y) + jX_I(F_x,F_y) = \left|X(F_x,F_y)\right| e^{j\theta_X(F_x,F_y)}$$

- It is a periodic signal (for all integer *k*):

$$X(F_x,F_y) = X(F_x \pm k, F_y) = X(F_x, F_y \pm k)$$

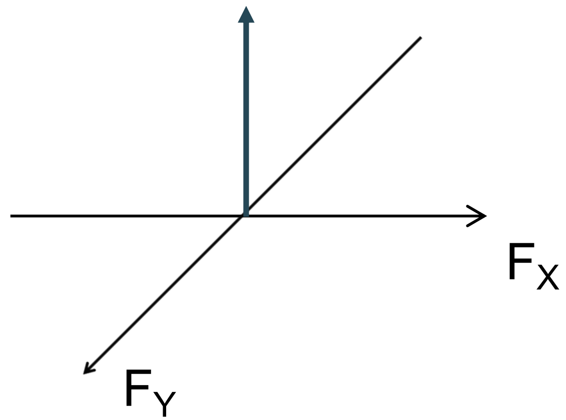
FT properties (2)

- Uniqueness
- Linearity
- Separability (compute it by rows and columns)
- Invariant against rotation
- Symmetry
- Convolution
  - Spatial convolution implies frequency product

$$x[m,n] * y[m,n] \leftrightarrow X\left(F_x, F_y\right) \cdot Y\left(F_x, F_y\right)$$

FT examples of infinite images (1)

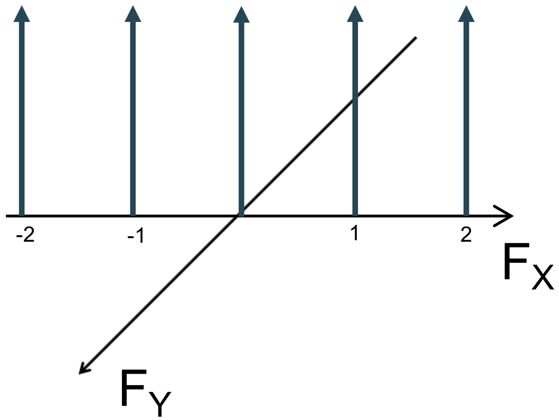
$$x[m,n] = 1 \overset{FT}{\longleftrightarrow} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(F_x - k, F_y - l)$$



$$\delta(F_x, F_y)$$

FT examples of infinite images (1)

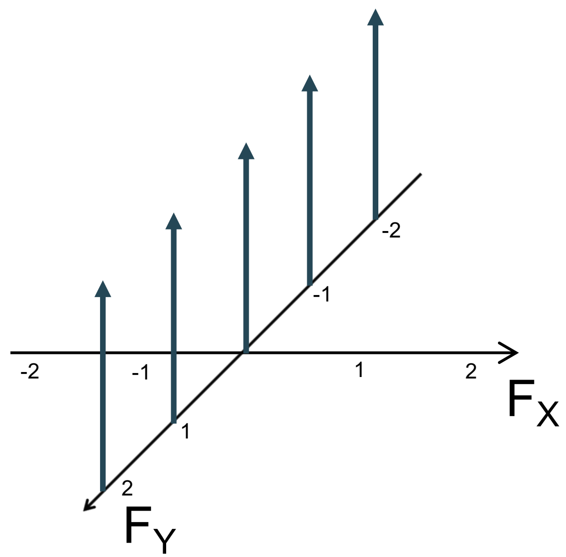
$$x[m,n] = 1 \overset{FT}{\longleftrightarrow} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(F_x - k, F_y - l)$$



$$\sum_{k=-\infty}^{\infty} \delta(F_x - k, F_y)$$

FT examples of infinite images (1)

$$x[m,n] = 1 \stackrel{FT}{\longleftrightarrow} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(F_x - k, F_y - l)$$

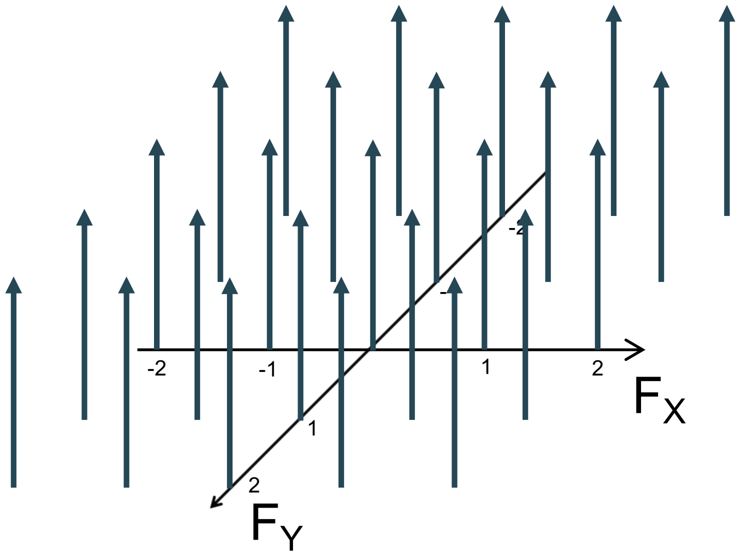


$$\sum_{l=-\infty}^{\infty} \delta(F_x, F_y - l)$$

FT examples of infinite images (1)

$$x[m,n] = 1 \stackrel{FT}{\longleftrightarrow} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(F_x - k, F_y - l)$$

FT



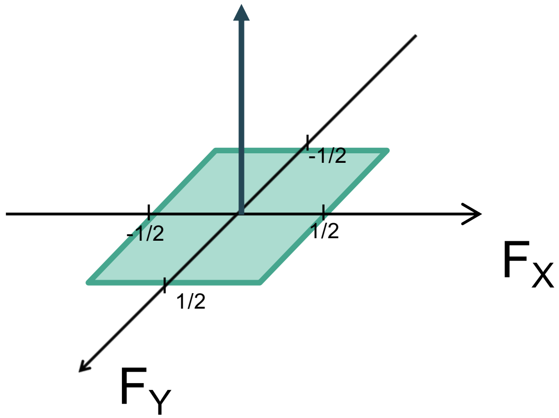
$$x[m,n] = 1$$

$$\sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(F_x - k, F_y - l)$$

FT examples of infinite images (1)

$$x[m,n] = 1 \overset{FT}{\longleftrightarrow} \delta(F_x, F_y) \qquad -\frac{1}{2} \leq \{F_x, F_y\} < \frac{1}{2}$$

FT



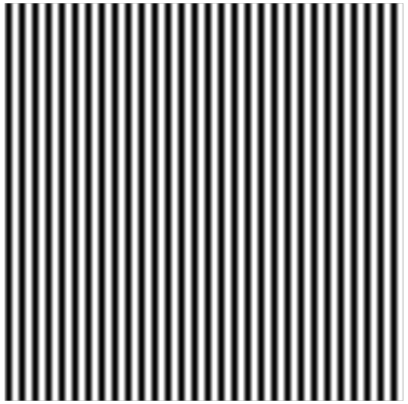
x[m,n] = 1

$$\delta(F_x, F_y) \qquad -\frac{1}{2} \leq \{F_x, F_y\} < \frac{1}{2}$$

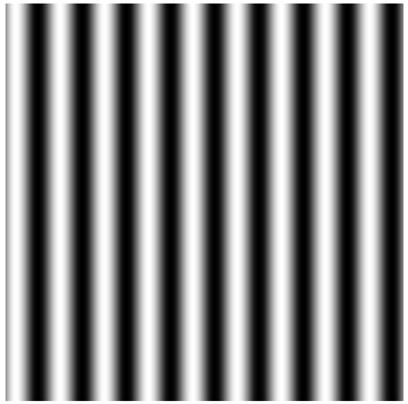
FT examples of infinite images (2)

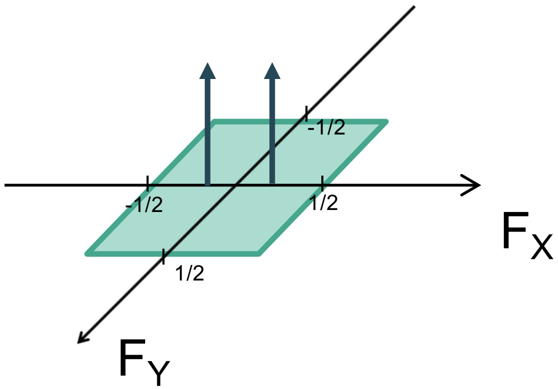
$$x[m,n] = cos(2\pi F_1 m) \overset{FT}{\longleftrightarrow} \frac{1}{2}\delta(F_x - F_1, F_y) + \frac{1}{2}\delta(F_x + F_1, F_y) \qquad -\frac{1}{2} \leq \{F_x, F_y\} < \frac{1}{2}$$

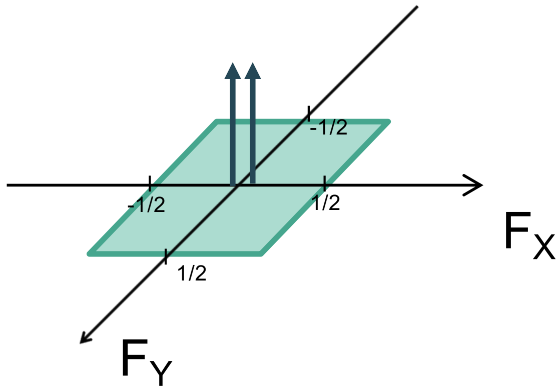
F_1= 0.1



F_1=0.03



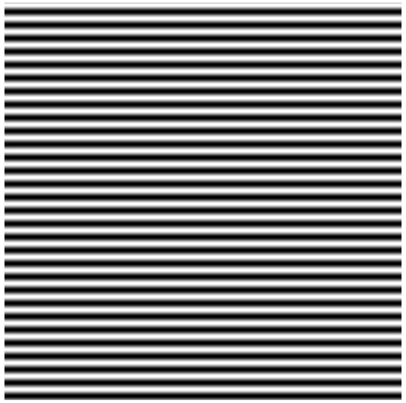




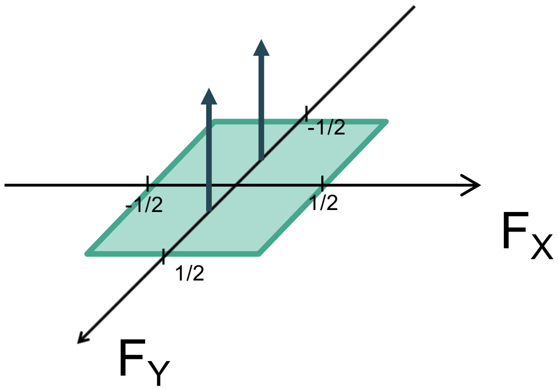
FT examples of infinite images (2)

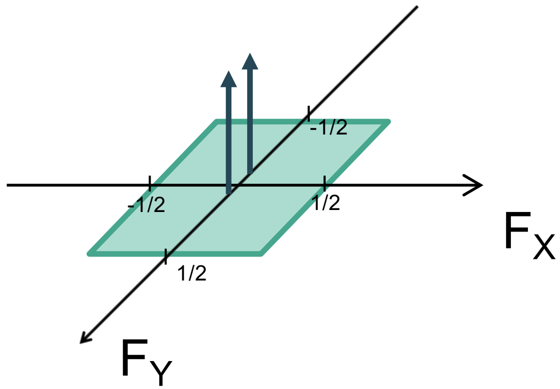
$$x[m,n] = cos(2\pi F_2 n) \stackrel{FT}{\longleftrightarrow} \frac{1}{2}\delta(F_x, F_y - F_2) + \frac{1}{2}\delta(F_x, F_y + F_2) \qquad -\frac{1}{2} \leq \{F_x, F_y\} < \frac{1}{2}$$

F_2= 0.1



F_2=0.03



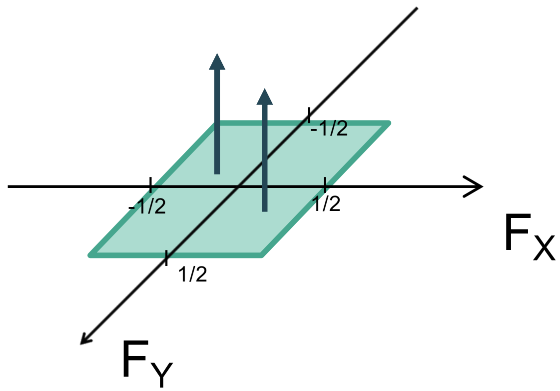


FT examples of infinite images (2)

$$x[m,n] = cos(2\pi F_1 m + 2\pi F_2 n) \overset{FT}{\longleftrightarrow} \frac{1}{2}\delta(F_x - F_1, F_y - F_2) + \frac{1}{2}\delta(F_x + F_1, F_y + F_2) \qquad -\frac{1}{2} \leq \{F_x, F_y\} < \frac{1}{2}$$

F_1=0.05, F_2= 0.03

$\overset{FT}{\longleftrightarrow}$



FT examples of infinite images (3)

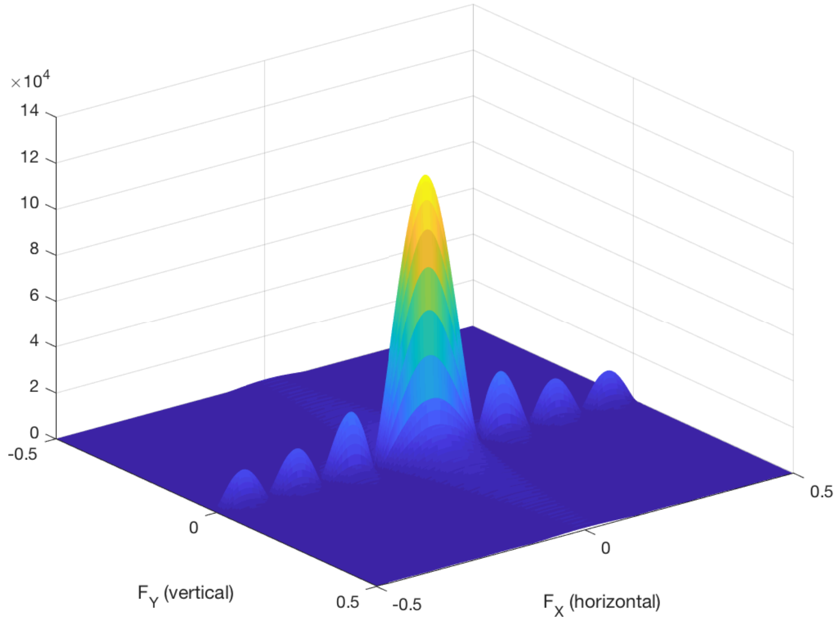
- FT of a rectangular pulse MxN (8x60)

$$x[m,n] = \Pi_{M\times N}[m,n] = \Pi_M[m]\cdot\Pi_N[n] \stackrel{FT}{\longleftrightarrow} X(F_x,F_y) = X_M(F_x)\cdot F_N(F_y)$$

separable

$$|X_M(F_x)| = \frac{1}{M}\frac{sin(M\pi F_x)}{sin(\pi F_x)}$$

x[m,n]

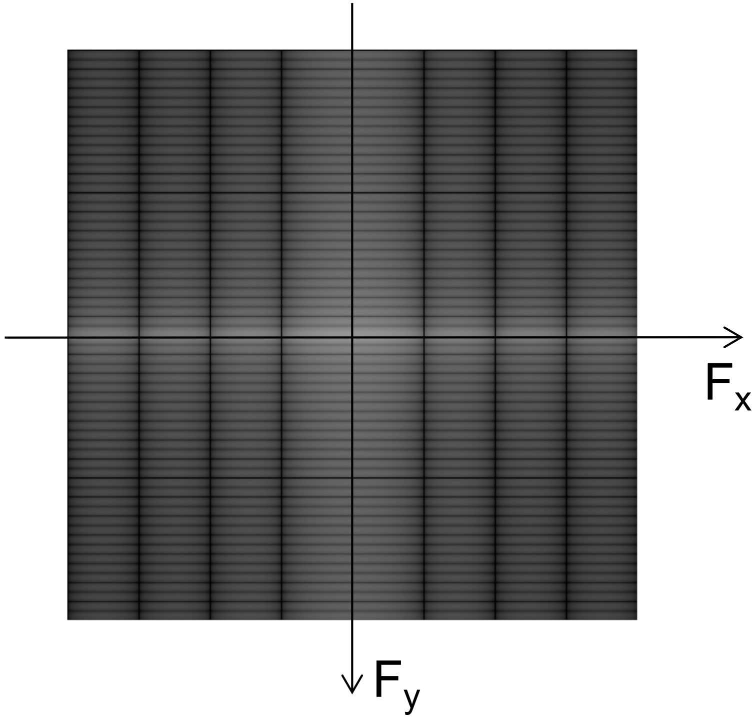


TF magnitude (modulus)

FT examples of infinite images (3)

- FT of a rectangular pulse (8x60)

image



TF magnitude (modulus)

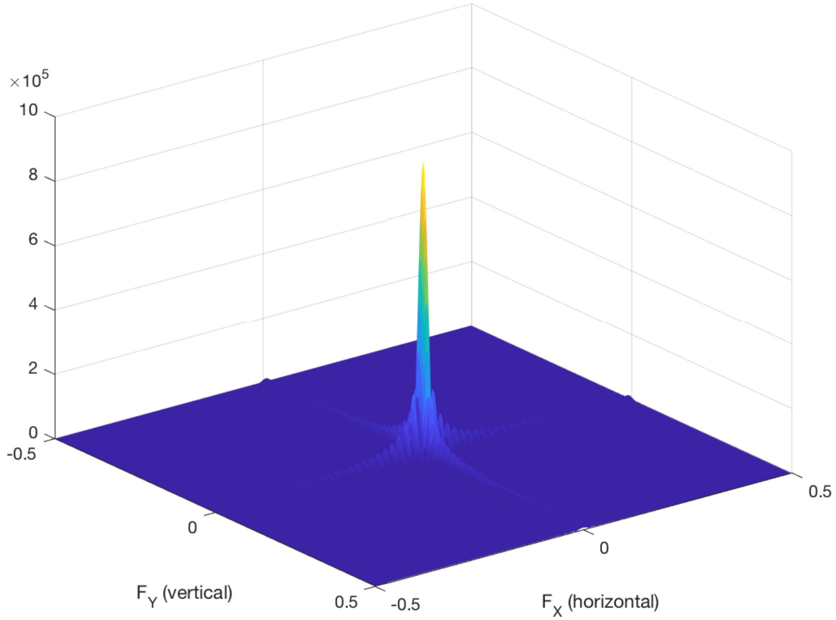
FT examples of infinite images (4)

- FT of a rectangular pulse MxN (60x60)

$$x[m,n] = \Pi_{M\times N}[m,n] = \Pi_M[m]\cdot\Pi_N[n] \overset{FT}{\longleftrightarrow} X(F_x,F_y) = X_M(F_x)\cdot F_N(F_y)$$

$$|X_M(F_x)| = \frac{1}{M}\frac{sin(M\pi F_x)}{sin(\pi F_x)}$$

x[m,n]

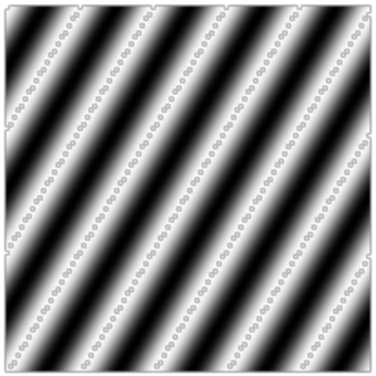


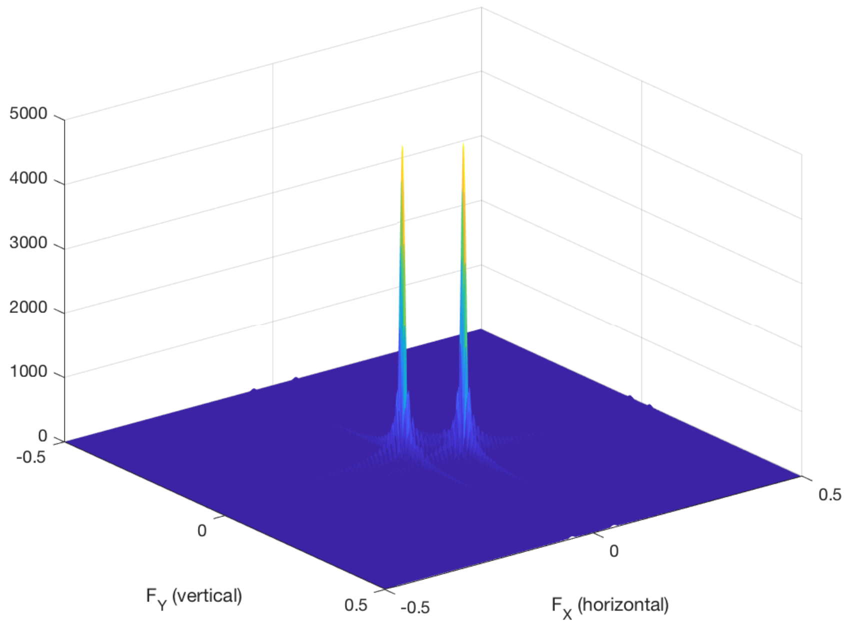
TF magnitude (modulus)

FT example of finite images

- Windowing effect (100x100)

F_A=0.05, F_B= 0.03





TF magnitude (modulus)

Outline

- Introduction
  - Image model definition
  - Convolution
  - Correlation

- 2D Fourier Analysis
  - Definition, properties and basic transforms

- **2D Discrete Fourier Analysis**
  - **Definition, properties and basic transforms**

2D Discrete Fourier analysis

- The **Discrete Fourier Transform** (**DFT**) of an image is defined as:

$$X[k,l] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m,n] e^{-j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)} \qquad 0 \leq \begin{Bmatrix} k \\ l \end{Bmatrix} < \begin{Bmatrix} M \\ N \end{Bmatrix}$$

  whereas the **Inverse Discrete Fourier Transform** (**IDFT**) is defined as:

$$x[m,n] = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} X[k,l] e^{j2\pi\left(\frac{k}{M}m + \frac{l}{N}n\right)} \qquad 0 \leq \begin{Bmatrix} k \\ l \end{Bmatrix} < \begin{Bmatrix} M \\ N \end{Bmatrix}$$

- Therefore, the DFT **transforms a 2D signal of MxN samples** in the original (space) domain into a set of **MxN samples in the transformed** (frequency) domain:
  - Usually, samples in the spatial domain are real (or integer) whereas **samples in the frequency domain are complex**.

DFT relations with FT

$$X[k,l] = X\left(F_x, F_y\right) * V\left(F_x, F_y\right)\Big|_{F_x=\frac{k}{M}, F_y=\frac{l}{N}}$$

where $V(F_x, F_y)$ represents the FT of the implicit rectangular window in the DFT of MxN samples

If MxN is greater than or equal to the size of the image then:

$$X[k,l] = X\left(F_x, F_y\right)\Big|_{F_x=\frac{k}{M}, F_y=\frac{l}{N}}$$

DFT properties (1)

- **Properties of DFT with respect to TF**:
  - Similar properties but, in the case of a DFT of length L, indexes should remain in the interval [0,L-1]. This leads to **circular** convolution, displacement or time-reversal. These can be expressed with the help of:

$$\tilde{t}[m,n] = \sum_{r=-\infty}^{\infty} \sum_{s=-\infty}^{\infty} t[m+rM, n+sN], \quad \text{for} \begin{cases} 0 \le m \le M-1 \\ 0 \le n \le N-1 \end{cases}$$

- **Convolution**:
  - Spatial convolution implies frequency product.

$$\tilde{t}[m,n] \leftrightarrow X[k,l] \cdot Y[k,l]$$
$$\text{if } t[m,n] = x[m,n] * y[m,n]$$

- **Convolution and windowing**:
  - Windowing in the spatial domain implies frequency convolution.

$$x[m,n] \cdot y[m,n] \leftrightarrow \frac{1}{MN} \tilde{T}[k,l]$$
$$\text{if } T[k,l] = X[k,l] * Y[k,l]$$

DFT properties (2)

- **Spatial shift or Translation**:
  - A spatial shift only affects the phase of the transformed signal.

$$\tilde{x}[m-m',n-n'] \leftrightarrow X[k,l]e^{-j2\pi\left(\frac{km'}{M}+\frac{\ln'}{N}\right)}$$

example

- **Frequency shift or Modulation:**
  - Multiplication of an image by a complex exponential implies a frequency shift.

$$x[m,n]e^{j2\pi\left(\frac{k'm}{M}+\frac{l'n}{N}\right)} \leftrightarrow \tilde{X}[k-k',l-l']$$

DFT properties (3)

- **Separability**
  - 1D algorithms can be used

$$DFT^{2D}_{image}[\cdot] = DFT^{1D}_{rows}\left[DFT^{1D}_{columns}[\cdot]\right]$$

$$\mathrm{DFT}^{1D}_{\mathrm{columns}}$$

$$X[k,l] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x[m,n]\, e^{-j2\pi\left(\frac{mk}{M}+\frac{nl}{N}\right)} = \sum_{m=0}^{M-1} e^{-j2\pi\frac{mk}{M}} \sum_{n=0}^{N-1} x[m,n]\, e^{-j2\pi\frac{nl}{N}} =$$

$$\mathrm{DFT}^{1D}_{rows}$$

$$= \sum_{m=0}^{M-1} e^{-j2\pi\frac{mk}{M}} V[m,l] \qquad 0 \le k \le M-1 \quad 0 \le l \le N-1$$

  - If the image is also a function of two 1D functions:

$$X[k,l] = DFT^{1D}_{n}\left[DFT^{1D}_{m}[x[m,n]]\right] = DFT^{1D}_{n}\left[DFT^{1D}_{m}[x_1[m]x_2[n]]\right] =$$
$$= DFT^{1D}_{n}\left[x_2[n]\right] DFT^{1D}_{m}\left[x_1[m]\right] = X_1[k]\cdot X_2[l]$$

DFT properties (4)

- **Rotation**:
  - A spatial rotation corresponds to the same frequency rotation

$$m = r\cos\theta \quad n = r\sin\theta$$

$$k = \omega\cos\phi \quad l = \omega\sin\phi$$

$$\boxed{x[r,\theta+\theta_0] \leftrightarrow X[\omega,\phi+\theta_0]}$$

- **Conjugate symmetry**:
  - If the **image is real**, its Fourier transform has the Hermitian symmetry
  - Useful when implementing filters

$$\boxed{X[k,l] = "X^*[-k,-l]"}$$

$$\begin{cases} X[0,0] = X^*[0,0] \\ X[0,l] = X^*[0,N-l], \quad \text{for } 1 \le l \le N-1 \\ X[k,0] = X^*[M-k,0], \quad \text{for } 1 \le k \le M-1 \\ X[k,l] = X^*[M-k,N-l], \quad \text{for } \begin{array}{c} 1 \le k \le M-1 \\ 1 \le l \le N-1 \end{array} \end{cases}$$

Other DFT issues (1)

- **Centered representation:**
  - Since images are generally positive signals, the highest value of the magnitude of their transform is at (*k,l*) = (0, 0). Given the symmetries of the DFT, this makes that the four corners of the transformed image contain the highest values of the magnitude. In order **to help visualizing**, the transformed image is represented centered at (*M*/2, *N*/2).
  - This can be seen as an example of the **modulation property**

$$\boxed{x[m,n]e^{j2\pi\left(\frac{k'm}{M}+\frac{l'n}{N}\right)} \leftrightarrow \tilde{X}[k-k',l-l']}$$

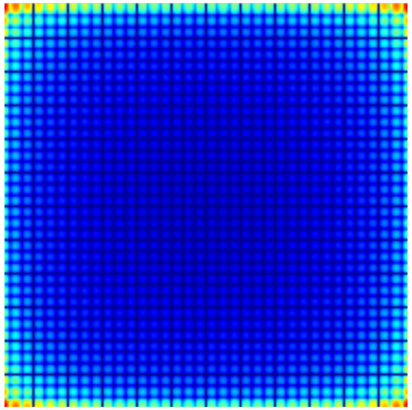
$$\text{For } k'=\frac{M}{2}, l'=\frac{N}{2}, \quad x[m,n]e^{j2\pi\left(\frac{m}{2}+\frac{n}{2}\right)} \leftrightarrow \tilde{X}[k-\frac{M}{2}, l-\frac{N}{2}]$$

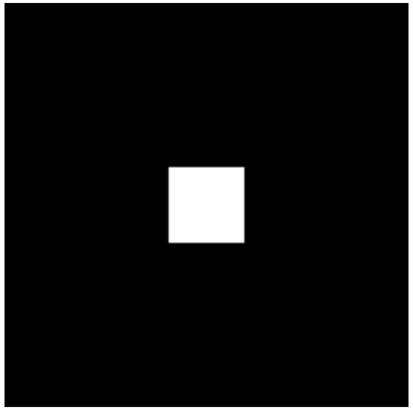
$$x[m,n](-1)^{m+n} \leftrightarrow \tilde{X}[k-\frac{M}{2}, l-\frac{N}{2}]$$

Other DFT issues (2)

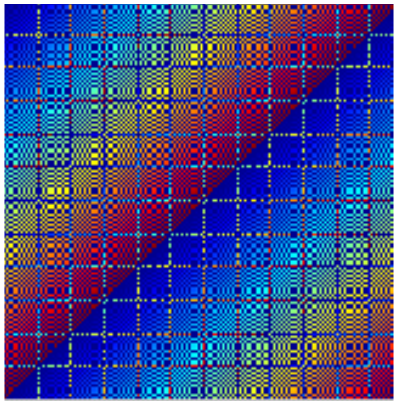
Non-centered representation

magnitude

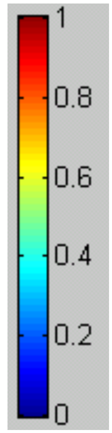




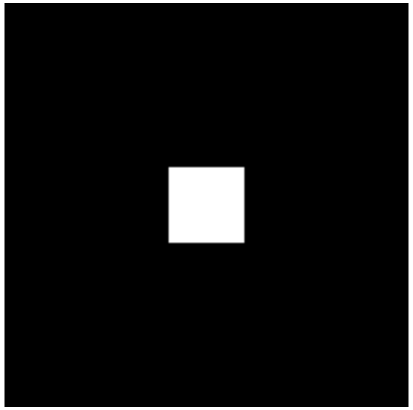
Input image



phase

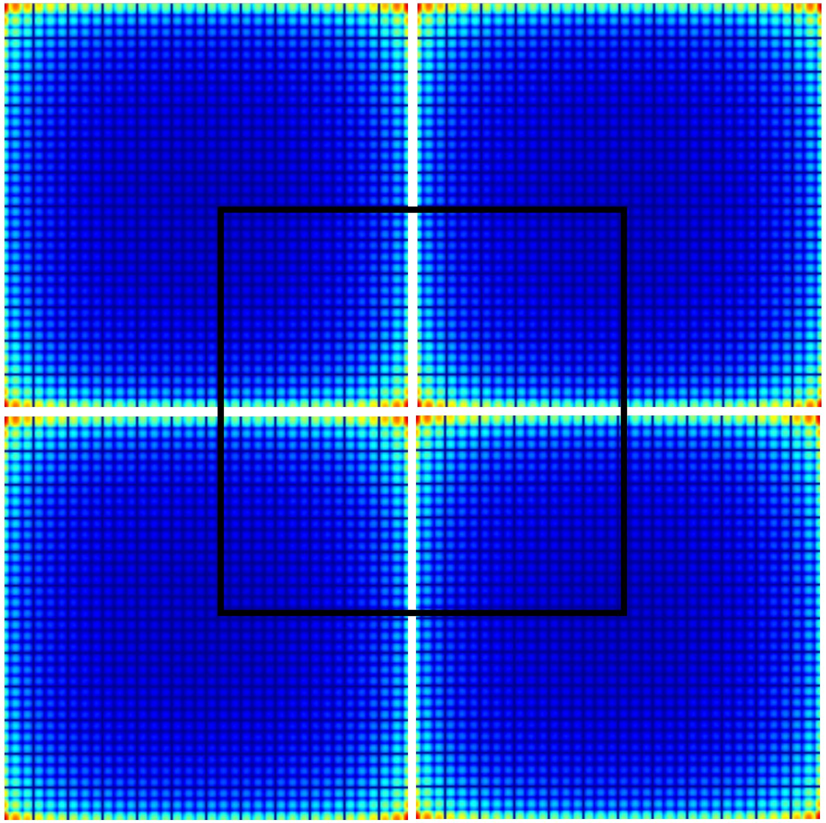


Other DFT issues (2)



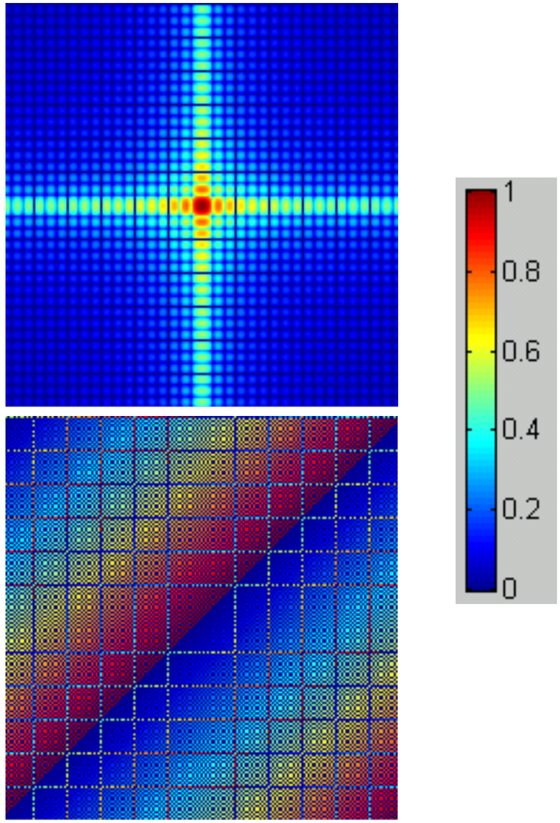
Input image

magnitude



Centered representation

magnitude

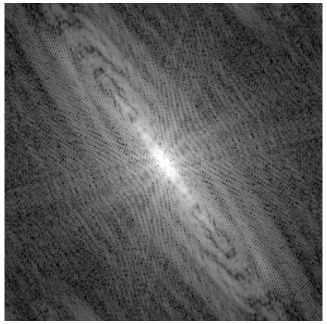


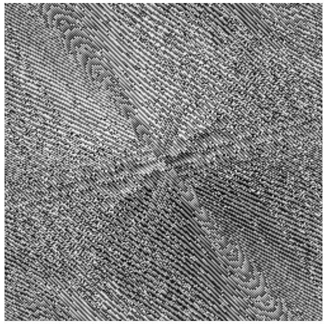
phase

Other DFT issues (3)

- **Magnitude and phase**

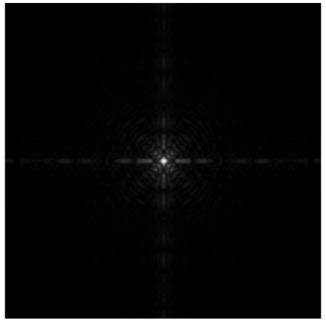
$$X[k,l] = \left|X[k,l]\right| e^{-j\varphi(k,l)}$$

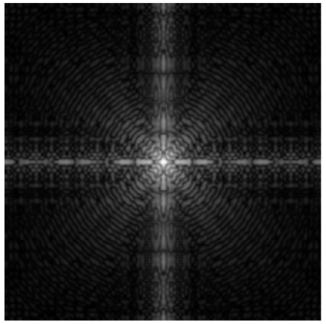




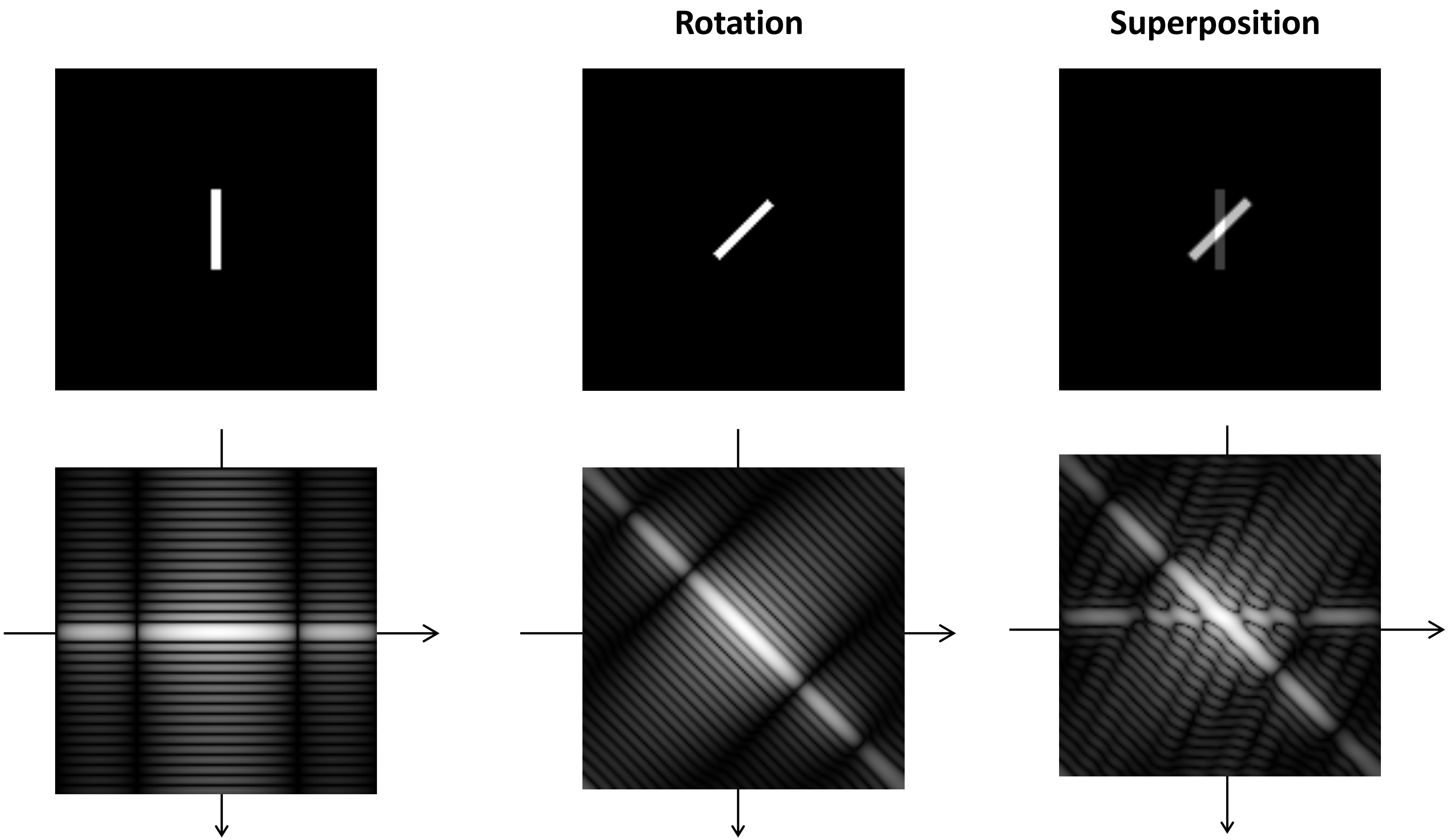
- **Logarithmic representation:**
  - To highlight the lowest values

$$\log\left(1 + \left|X[k,l]\right|\right)$$

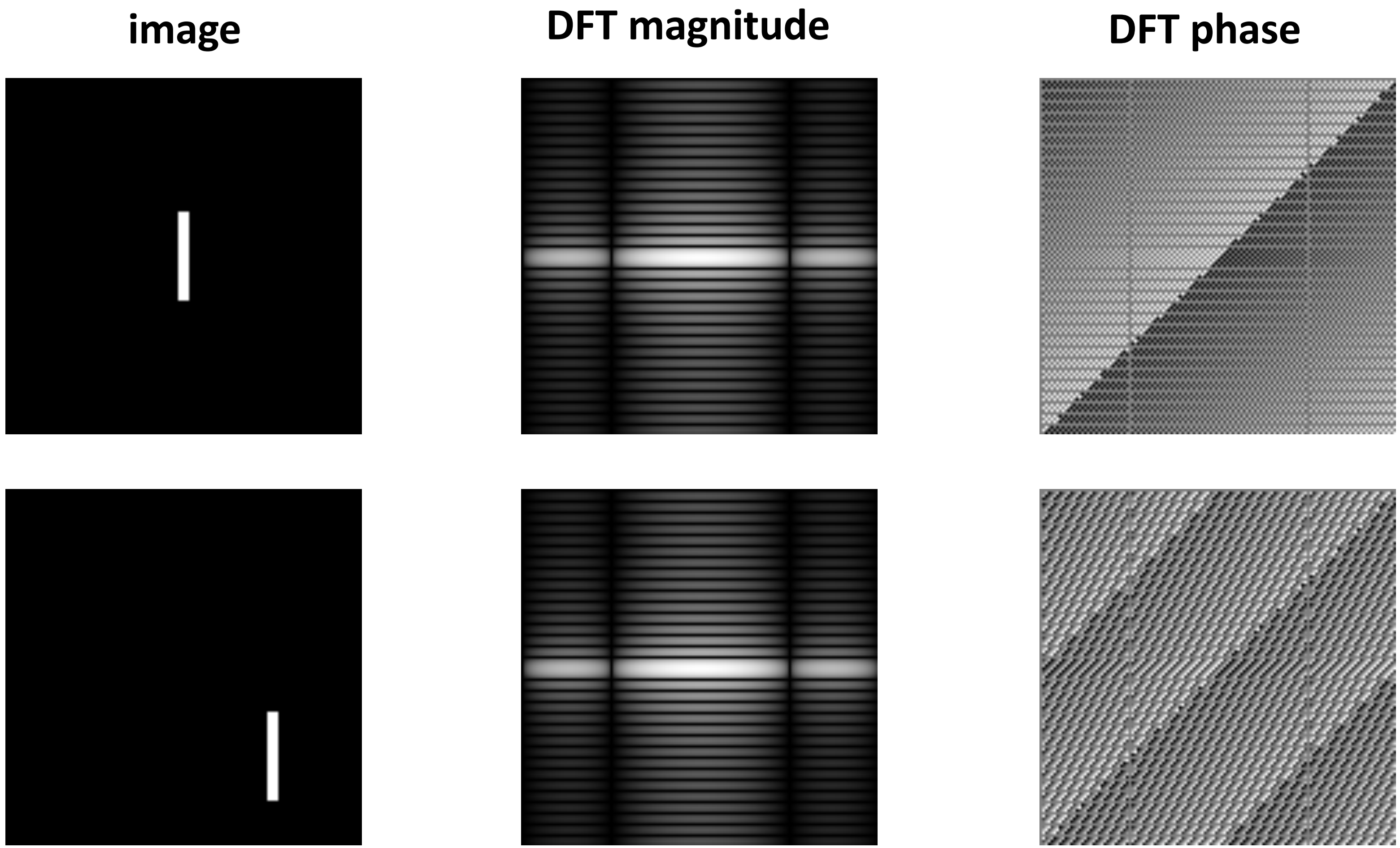




Basic DFT transforms (1)



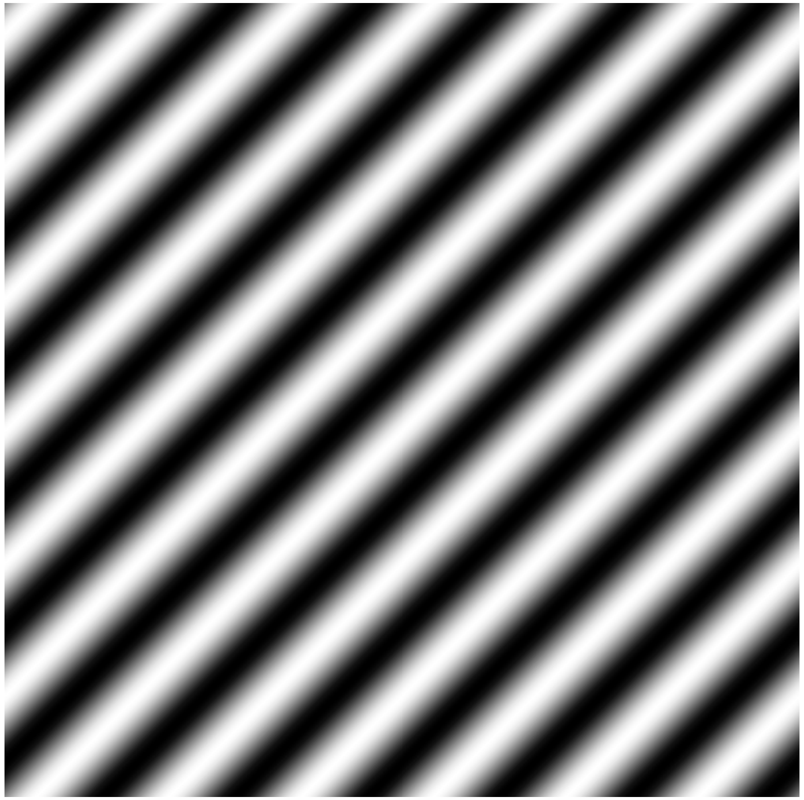
Basic DFT transforms (2)

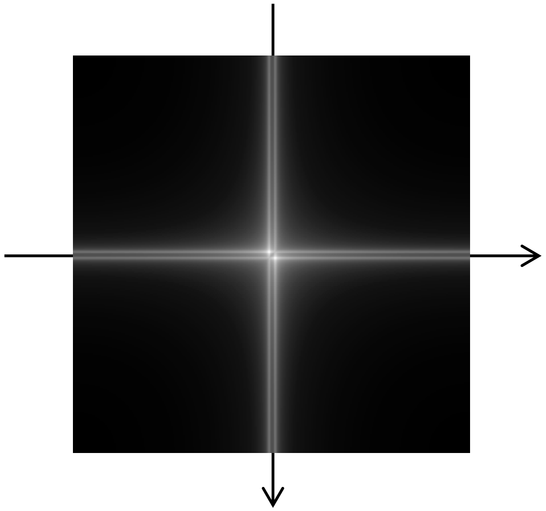


DFT windowing effect (1)

$$I[m,n] = \sin(2\pi F_1 m + 2\pi F_2 n)$$

F_1= 0.008, F_2= 0.008



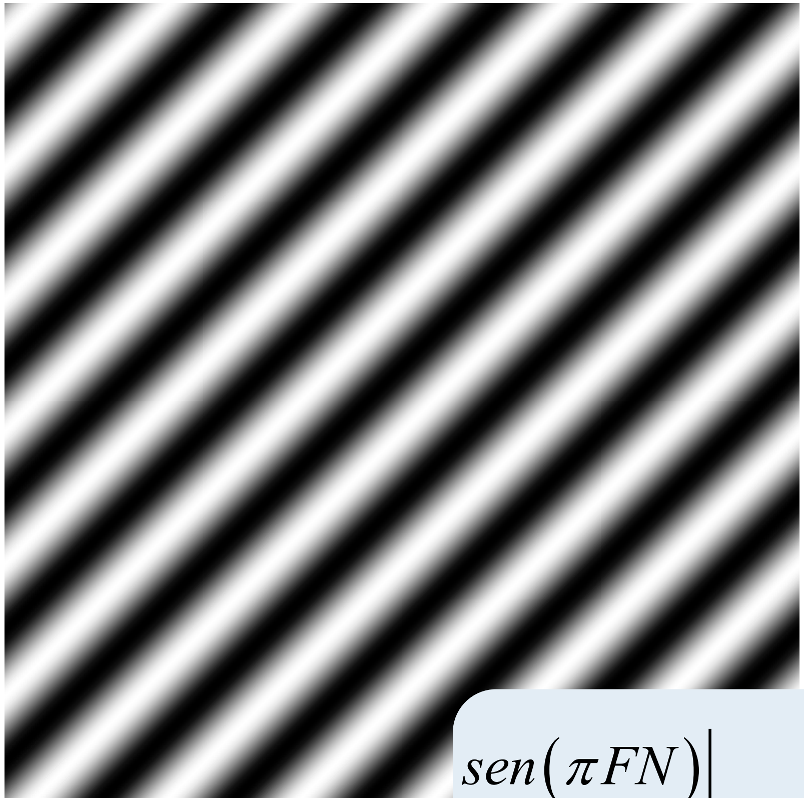


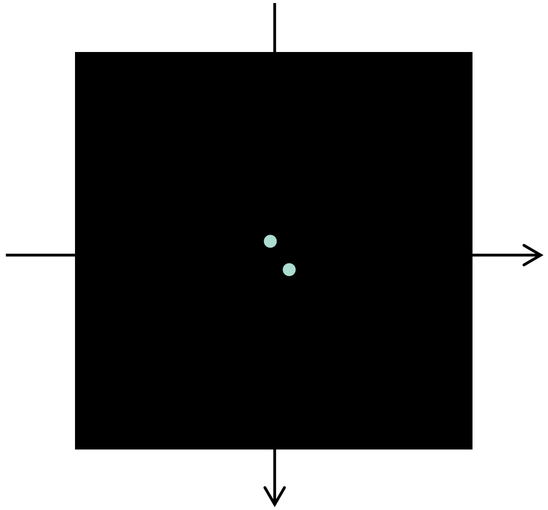
$|F_1| \neq k / M$, $|F_2| \neq l / N$

DFT windowing effect (2)

$$I[m,n] = \sin(2\pi F_1 m + 2\pi F_2 n)$$

F_1= 0.01, F_2= 0.01





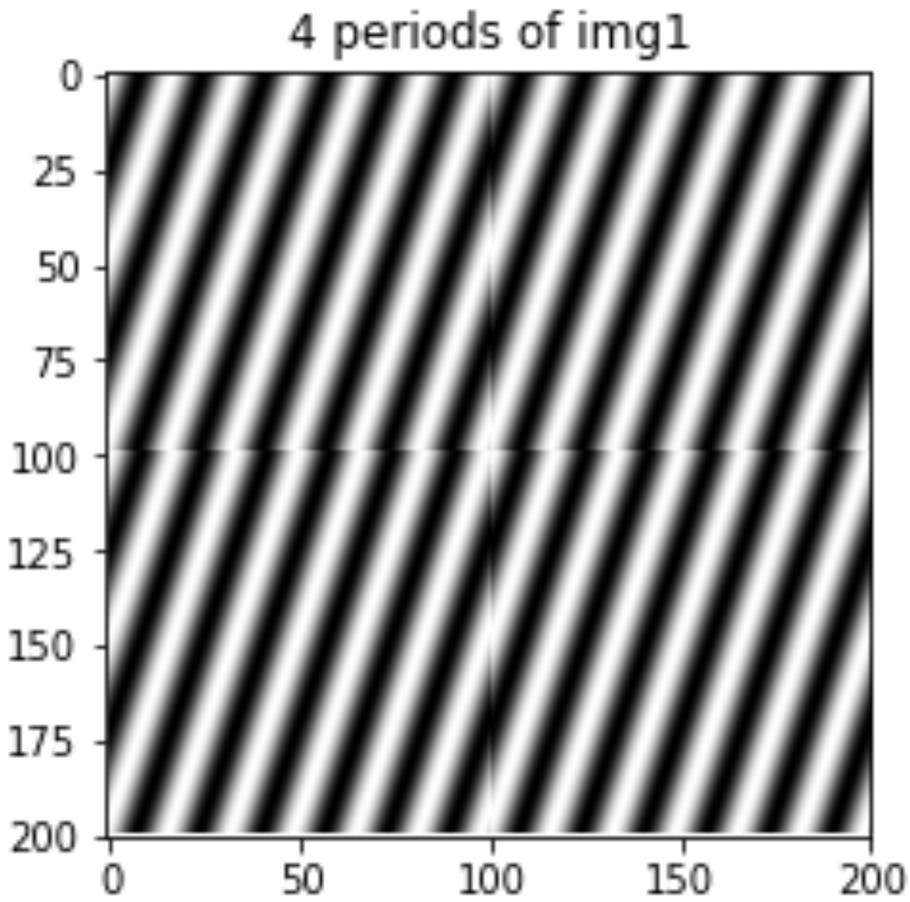
$$\left.\frac{sen(\pi F N)}{sen(\pi F)}\right|_{F=\frac{k}{N}} = N\,\delta[k]$$

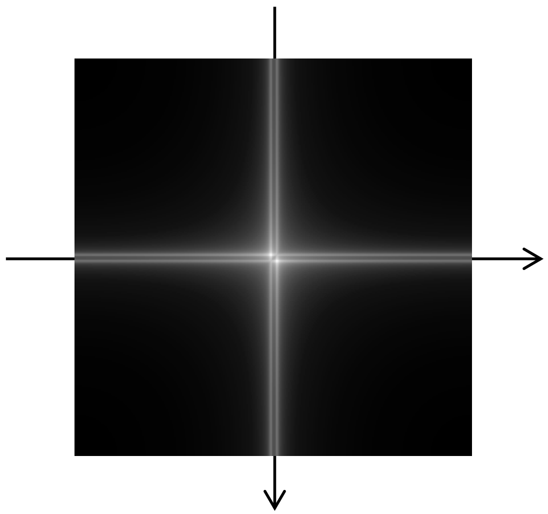
$|F_1| = k / M$, $|F_2| = l / N$

DFT windowing effect (3)

$$I[m,n] = \sin(2\pi F_1 m + 2\pi F_2 n)$$

F_1= 0.008, F_2= 0.008



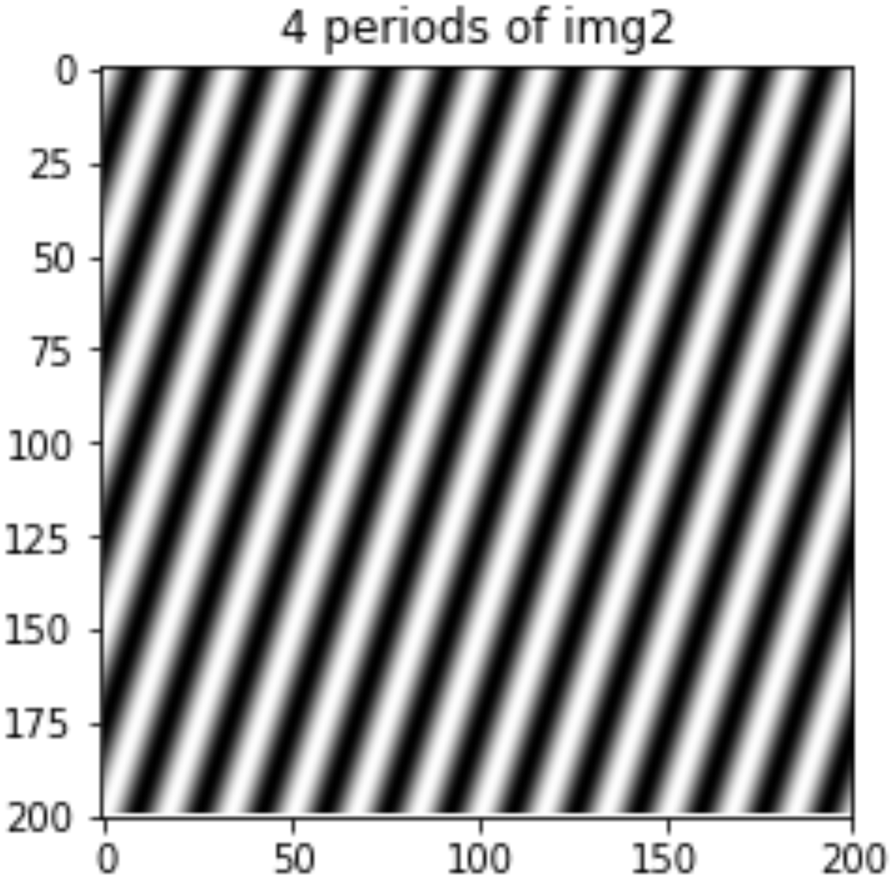


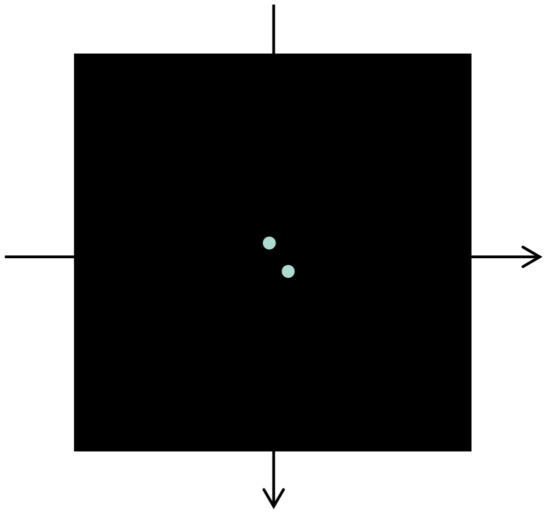
$|F_1| \neq k / M$, $|F_2| \neq l / N$

DFT windowing effect (4)

$$I[m,n] = \sin(2\pi F_1 m + 2\pi F_2 n)$$

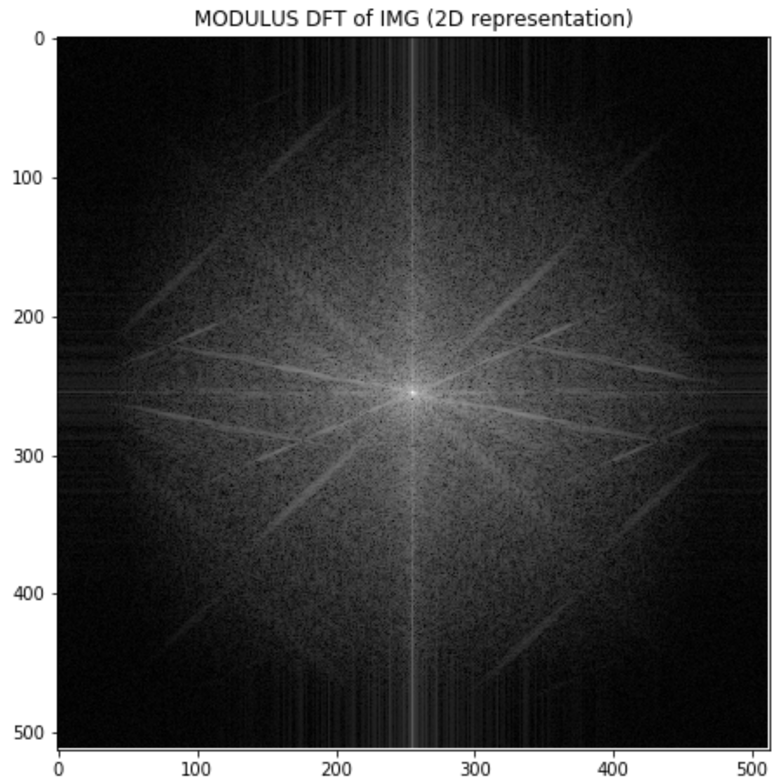
$F_1 = 0.01$, $F_2 = 0.01$



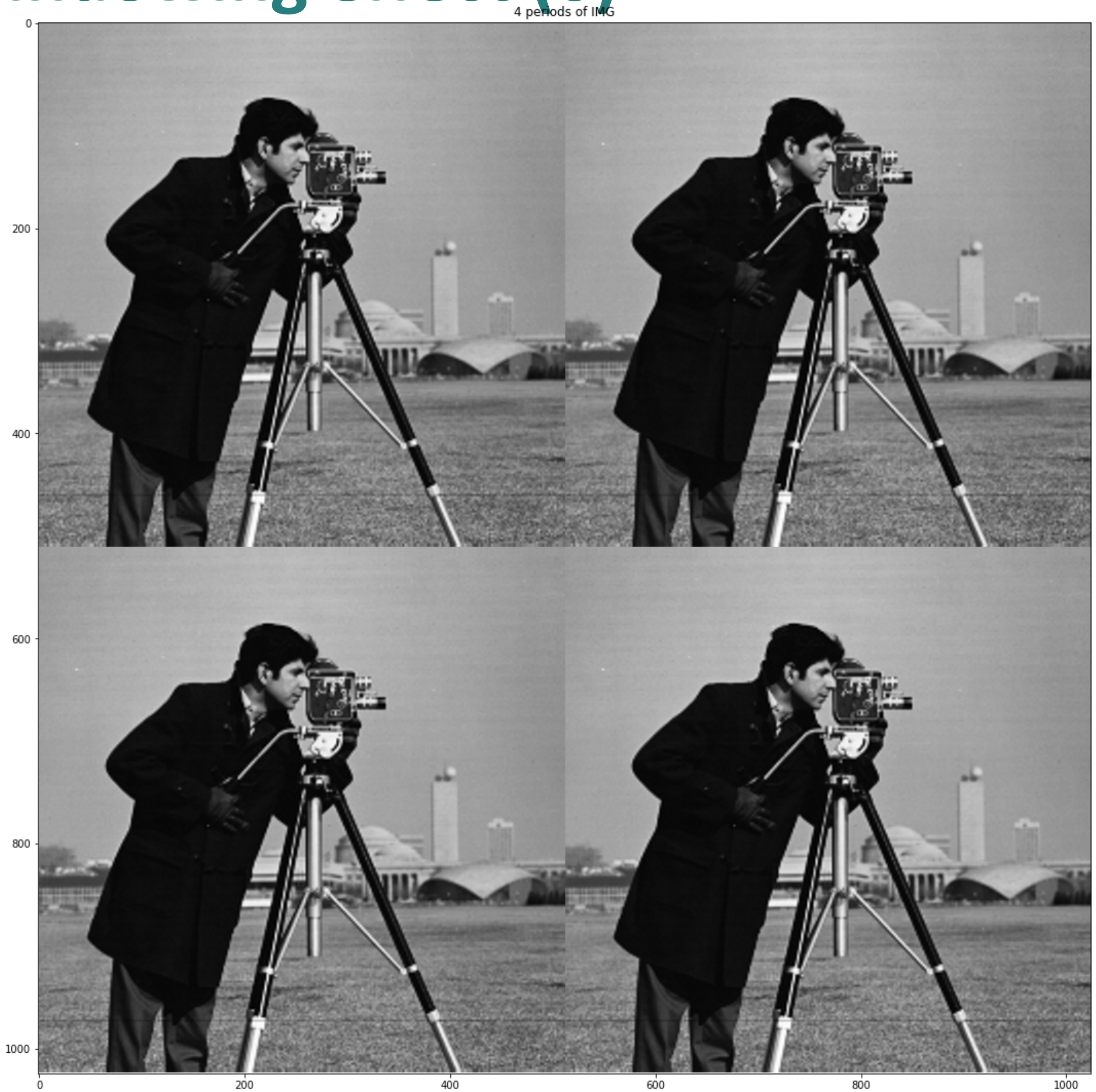


$|F_1| = k / M$, $|F_2| = l / N$

DFT windowing effect (5)



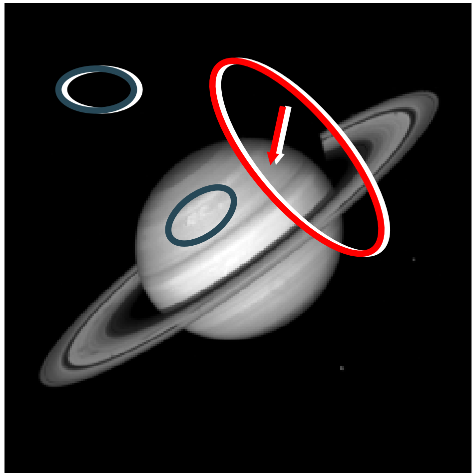
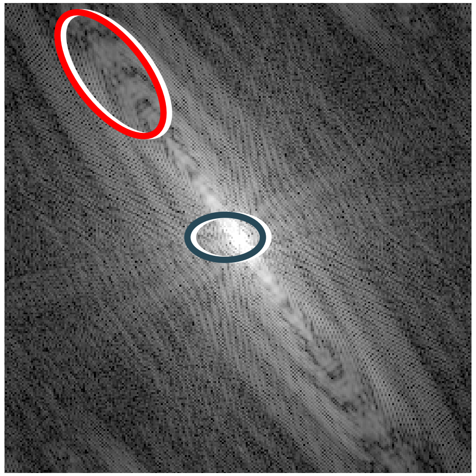
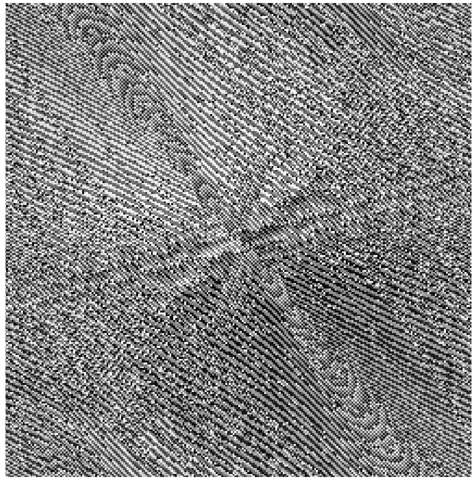
DFT windowing effect (6)



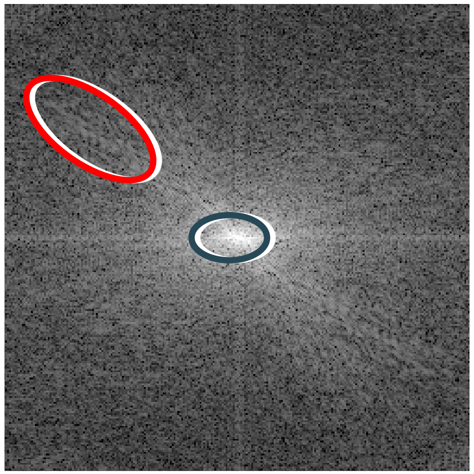
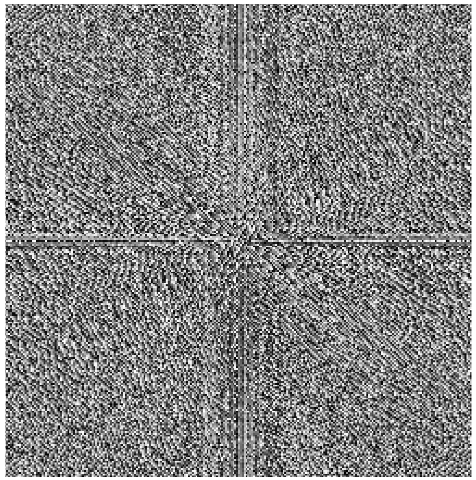
DFT transform examples (1)

High frequency

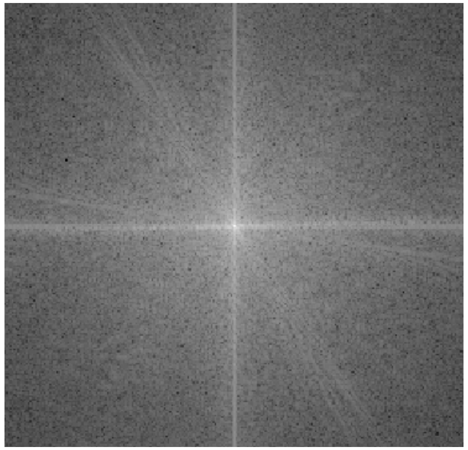
Low frequency

Signal ***DFT magnitude*** ***DFT phase***

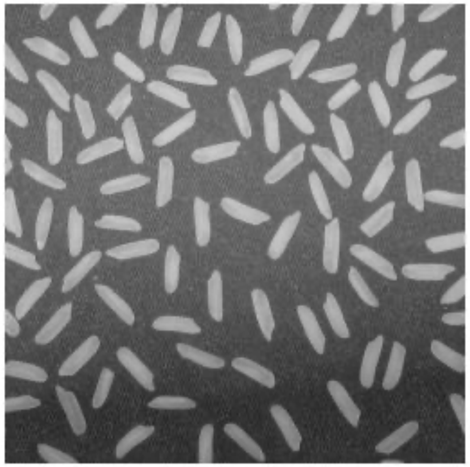




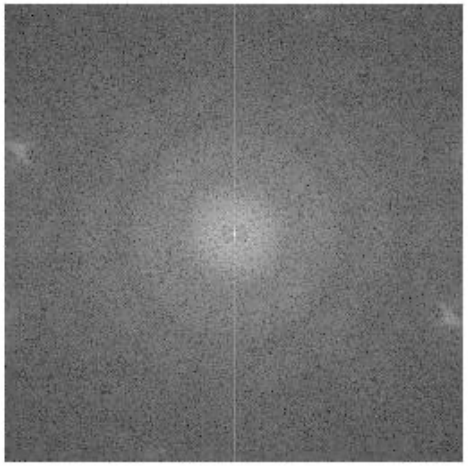
DFT transform examples (2)



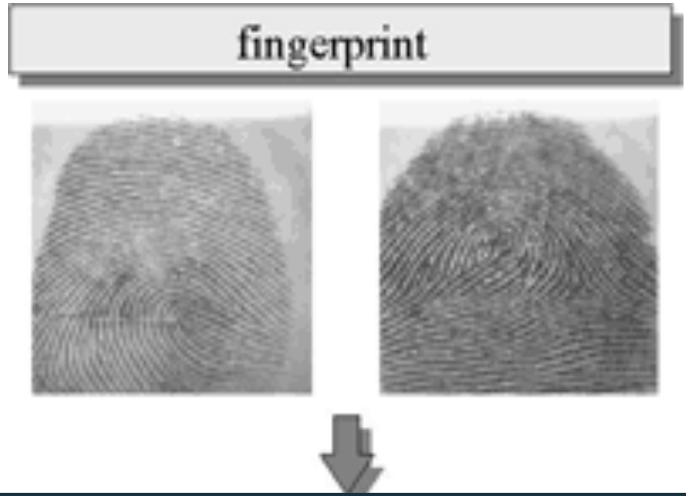
signal

DFT magnitude





DFT transform examples (3)

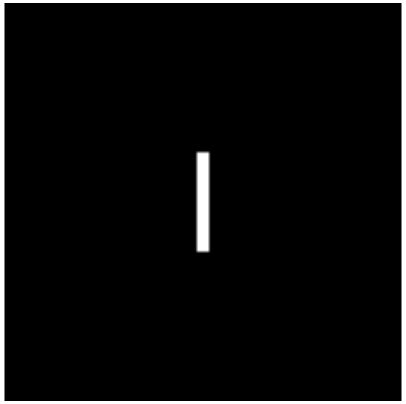
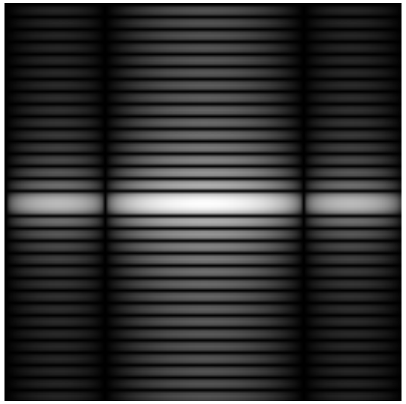
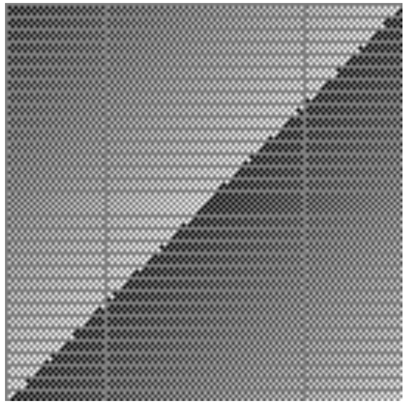


Ej. Gonzalez - Woods

Relevance of the phase in images (1)

- Associated to position of objects

image | DFT magnitude | DFT phase

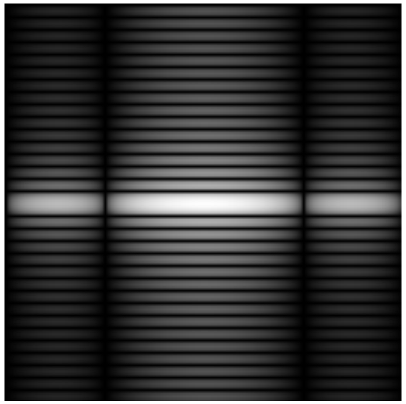
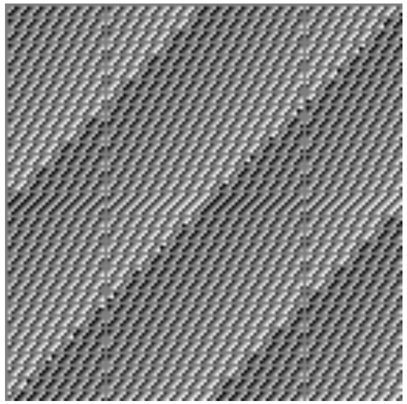




image | DFT magnitude | DFT phase

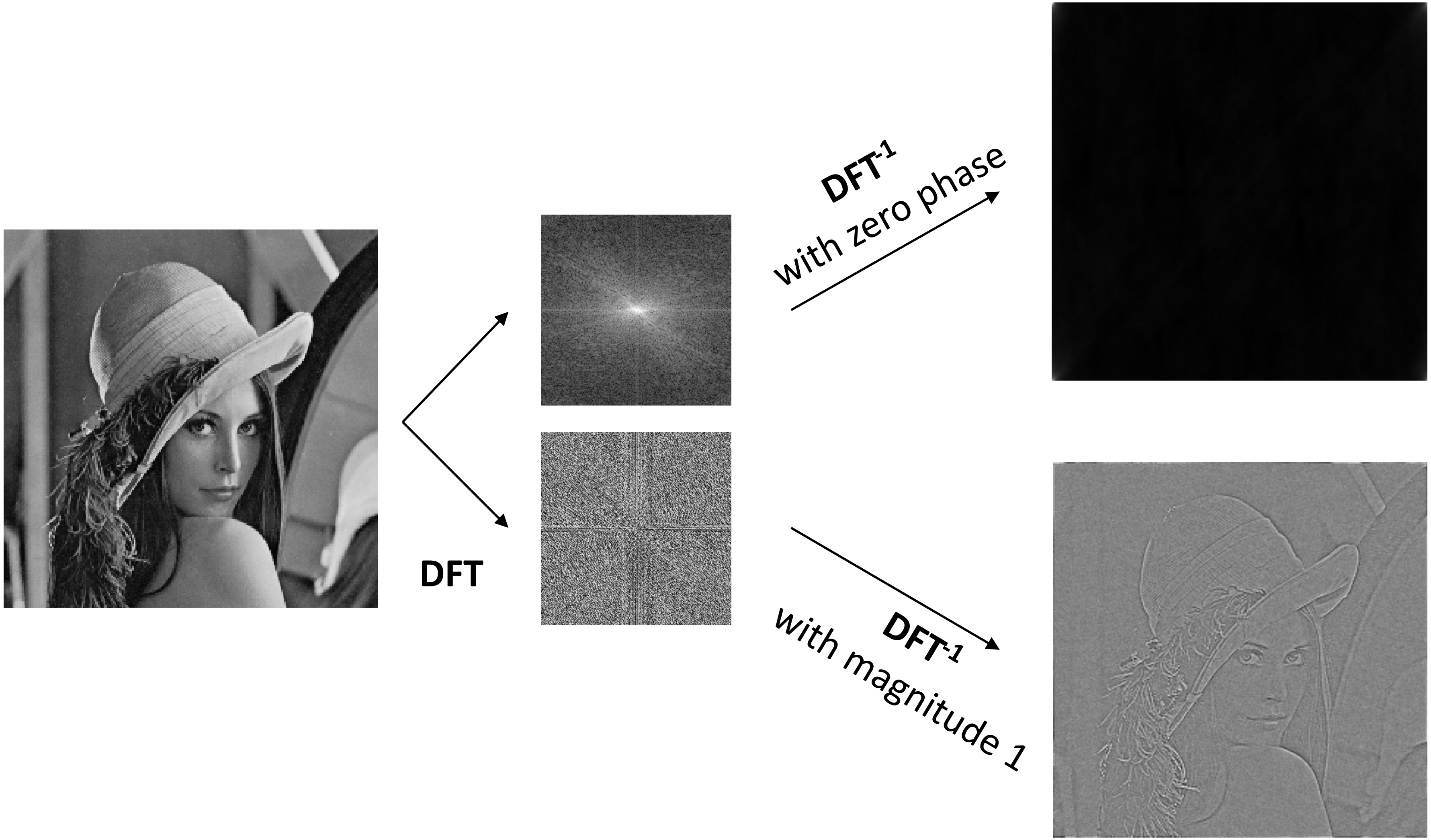
Relevance of the phase in images (2)

- Necessity of zero phase in processing

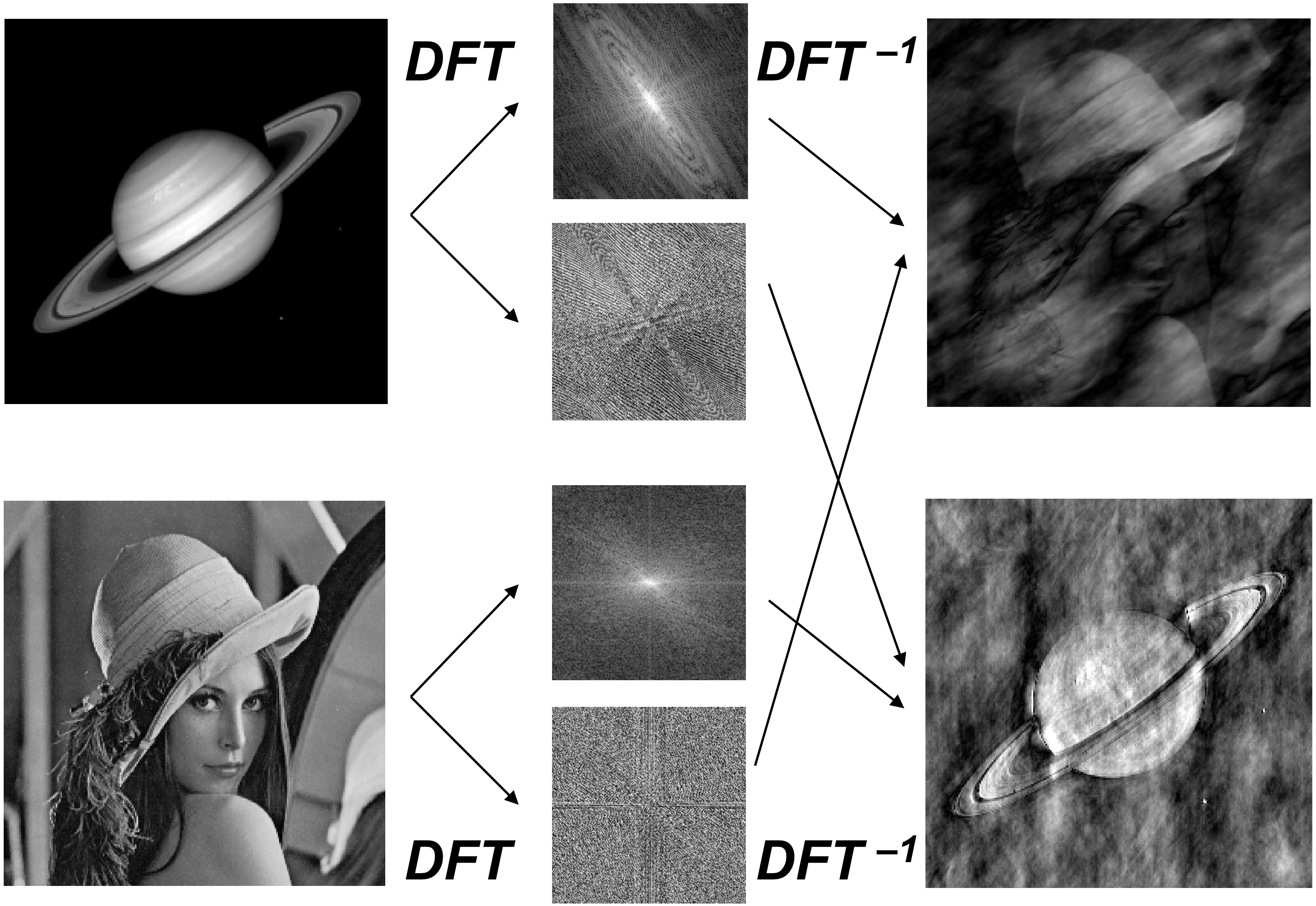
x[m,n]

$$DFT^{-1}\left\{ DFT\{x[m,n]\}\; e^{-j2\pi\frac{30}{M}m} e^{-j2\pi\frac{20}{M}n} \right\}$$

Relevance of the phase in images (3)



Relevance of the phase in images (4)



Summary

- Introduction
  - Image model definition
  - Convolution
  - Correlation

- 2D Fourier Analysis
  - Definition, properties and basic transforms

- 2D Discrete Fourier Analysis
  - Definition, properties and basic transforms