EECS 556 – Image Processing– W 09

2D SPACE SIGNALS/SYSTEMS

- DFT & FFT
- Lecture notes [Chapter DFT]

2D Discrete-Signal Fourier Series

If g[n,m] is periodic with period (N,M):

$$\tilde{x}[n,m] = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} \tilde{X}[k,l] \boxed{\mathrm{e}^{\imath 2\pi (kn/N + lm/M)}}$$

Orthonormal basis
Fourier basis

with

$$\tilde{X}[k,l] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \tilde{x}[n,m] \, \mathrm{e}^{-\imath 2\pi (kn/N + lm/M)}$$

Properties of DFS

- **linearity**
- **Linearity**
- **Shift**
- **Average value**
- **Symmetry properties**
- **Duality**

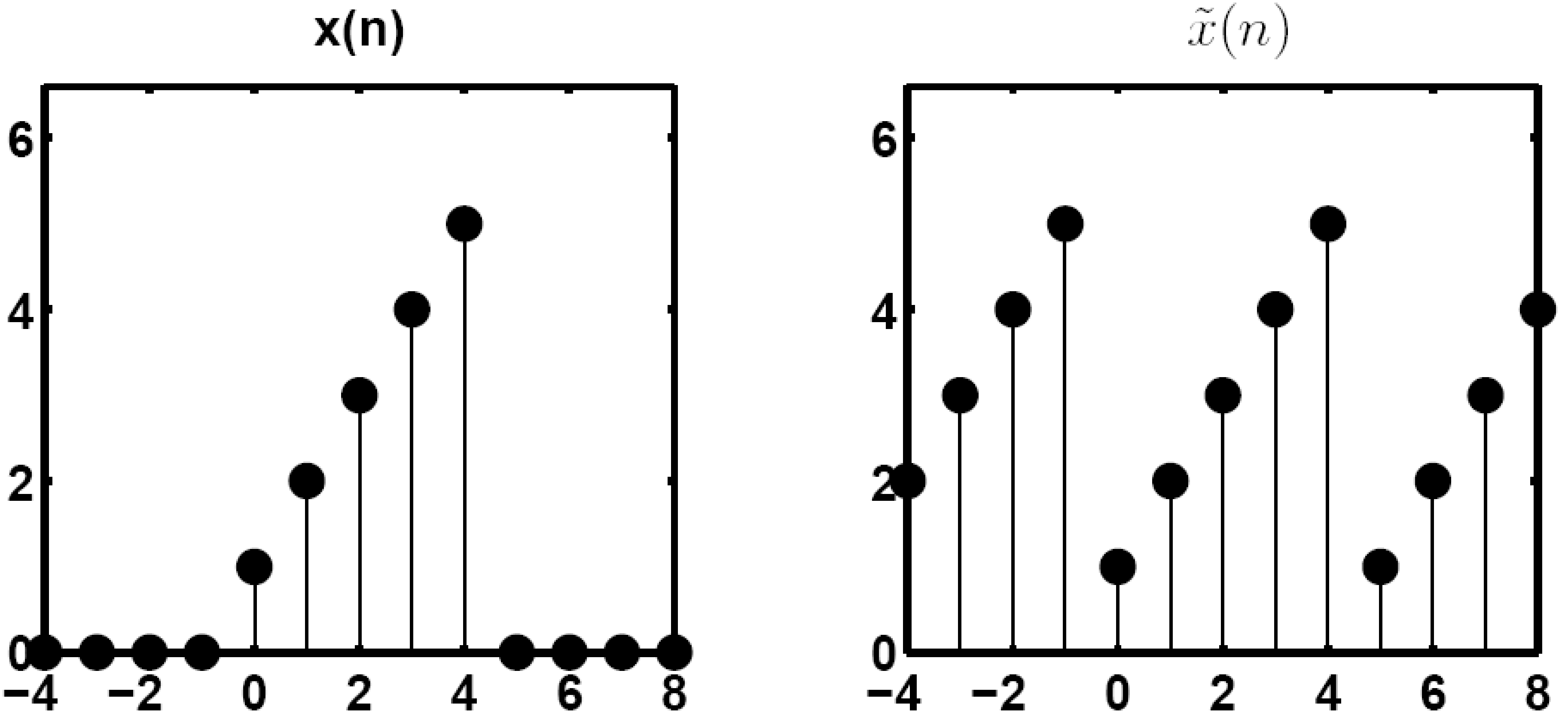
- For details see lecture notes on 2DFT

Circular Convolution

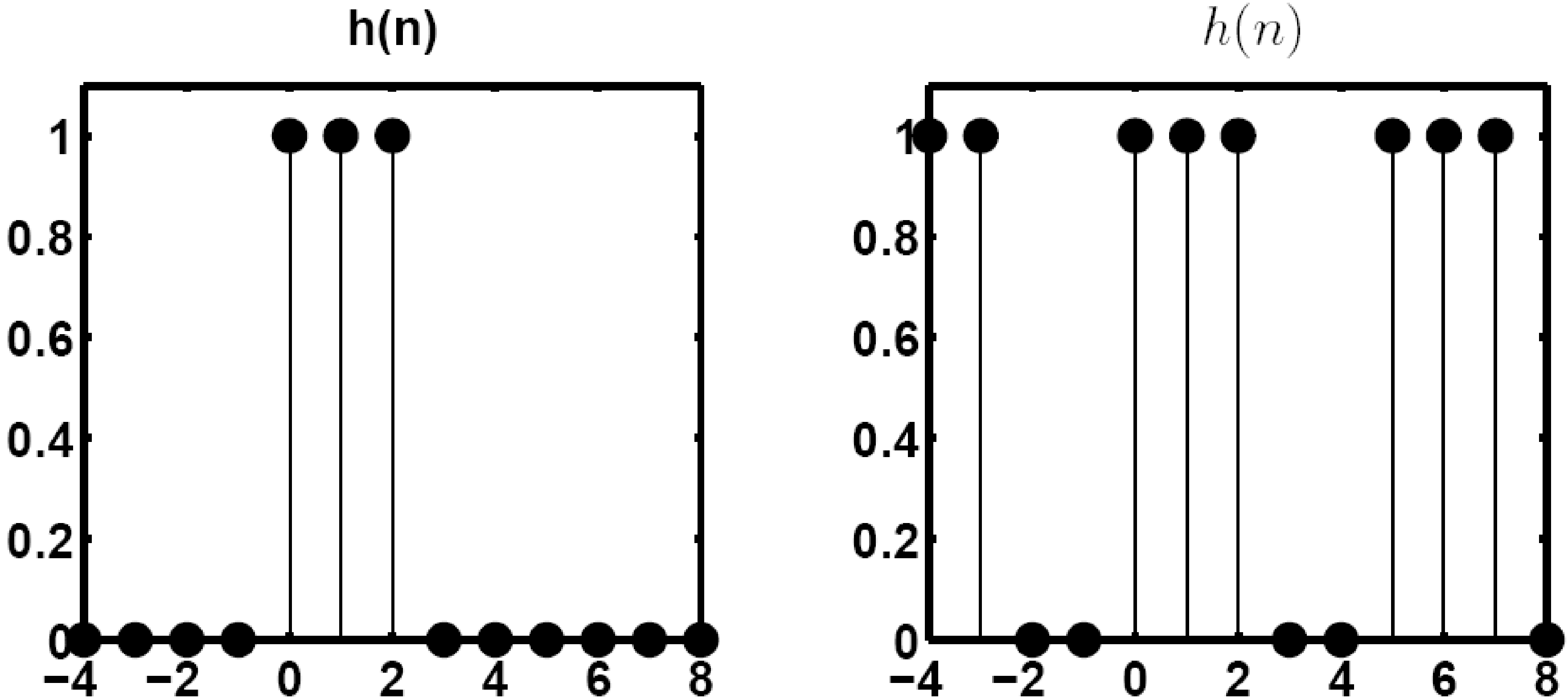
$$\boxed{\tilde{x}[n,m] \bigotimes \tilde{h}[n,m]} \stackrel{\text{DFS}}{\longleftrightarrow} \tilde{X}[k,l]\,\tilde{H}[k,l]$$

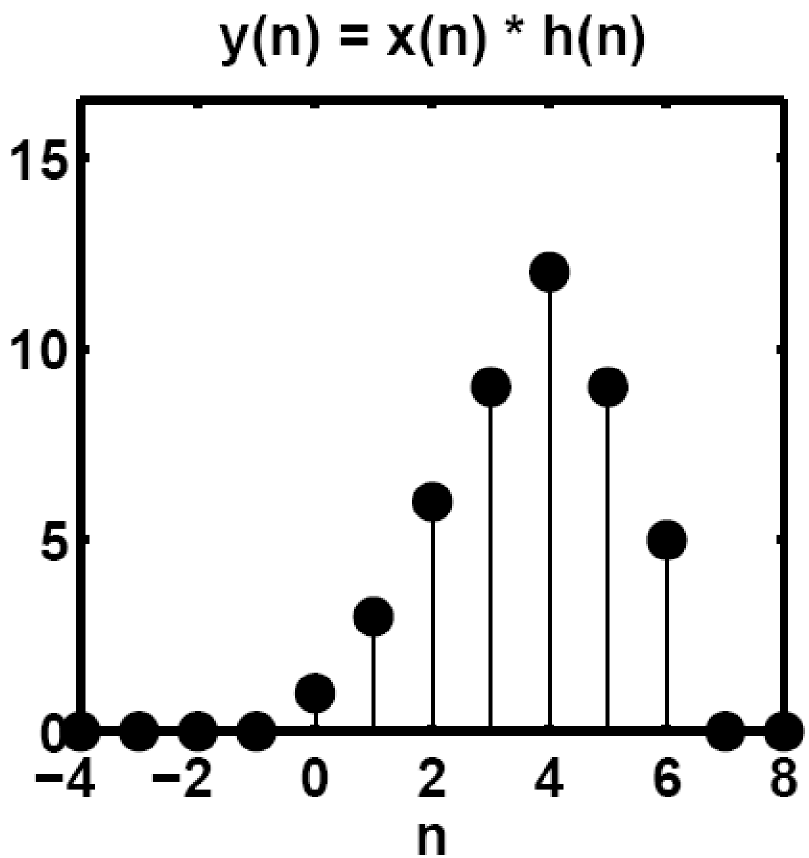
$$\boxed{\tilde{y}[n,m] = \sum_{n'=0}^{N-1} \sum_{m'=0}^{M-1} \tilde{x}[n',m']\,\tilde{h}[n-n',m-m']}$$

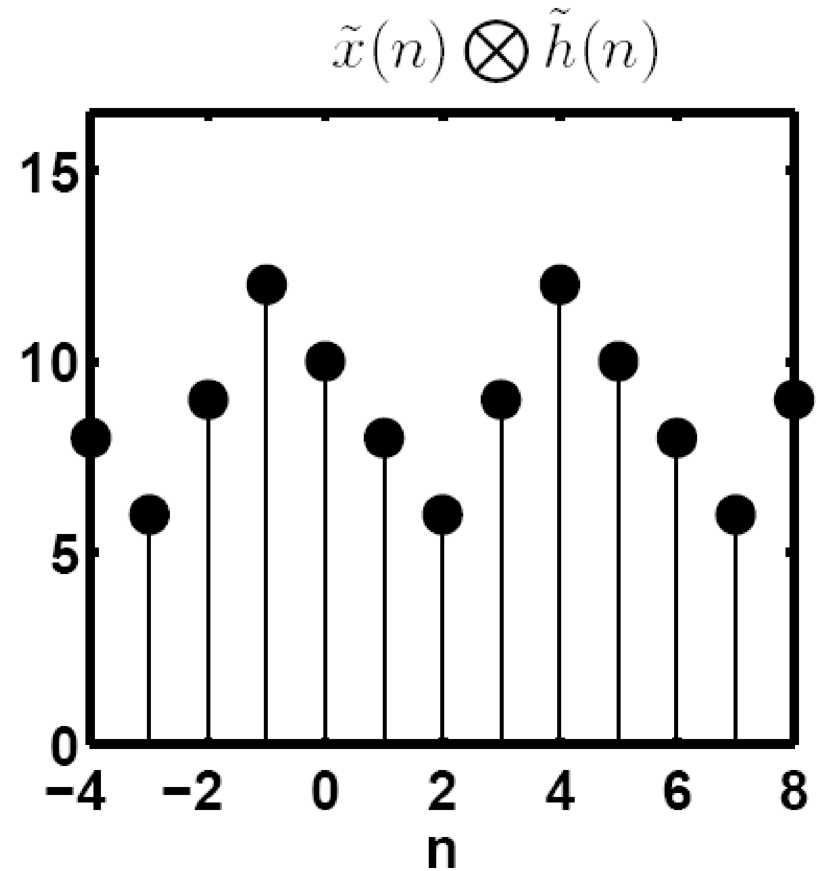
Note: the summation is finite as opposed to linear convolution



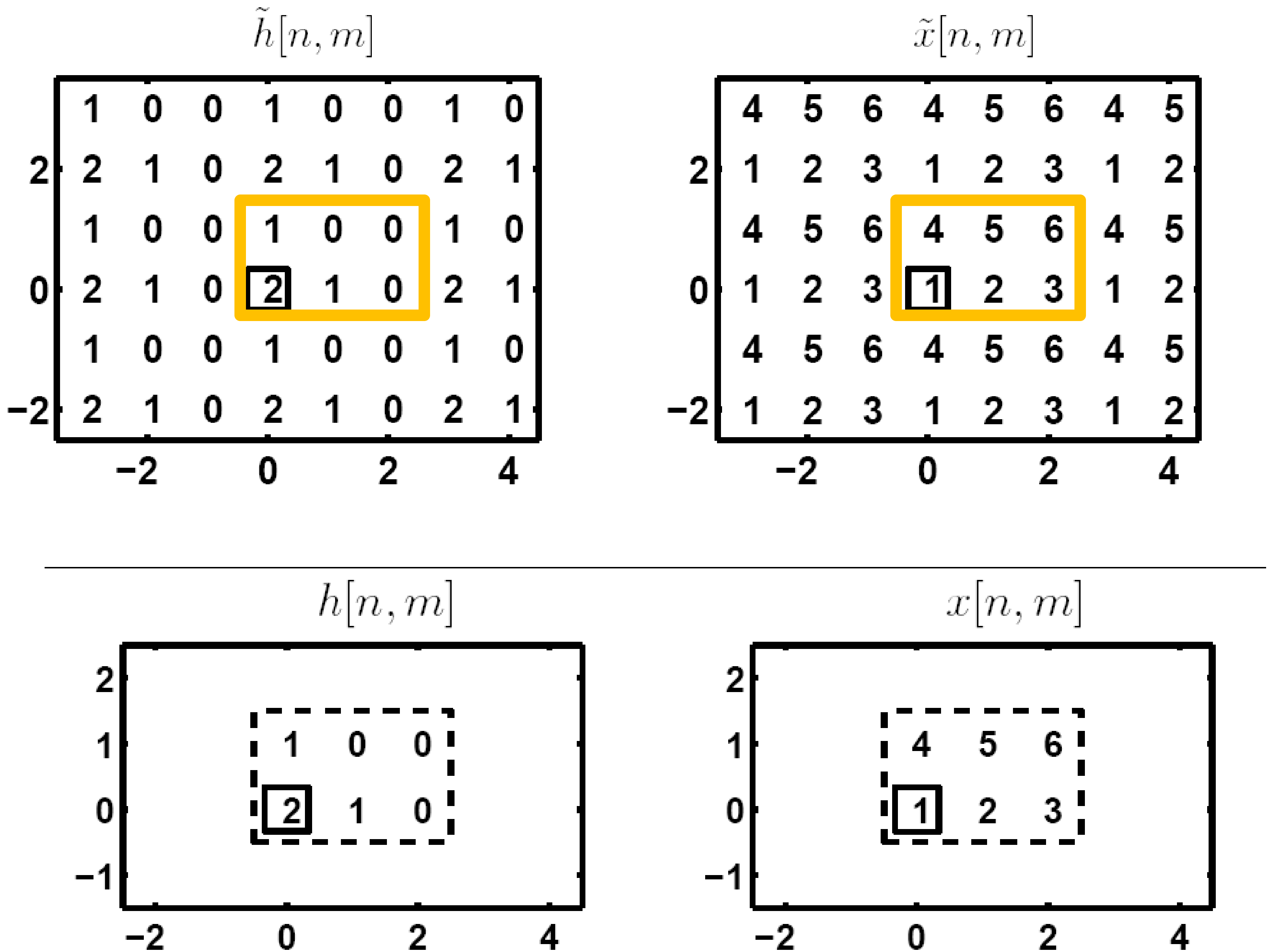
x(n)
$\tilde{x}(n)$
6
4
2
0
−4
−2
0
2
4
6
8

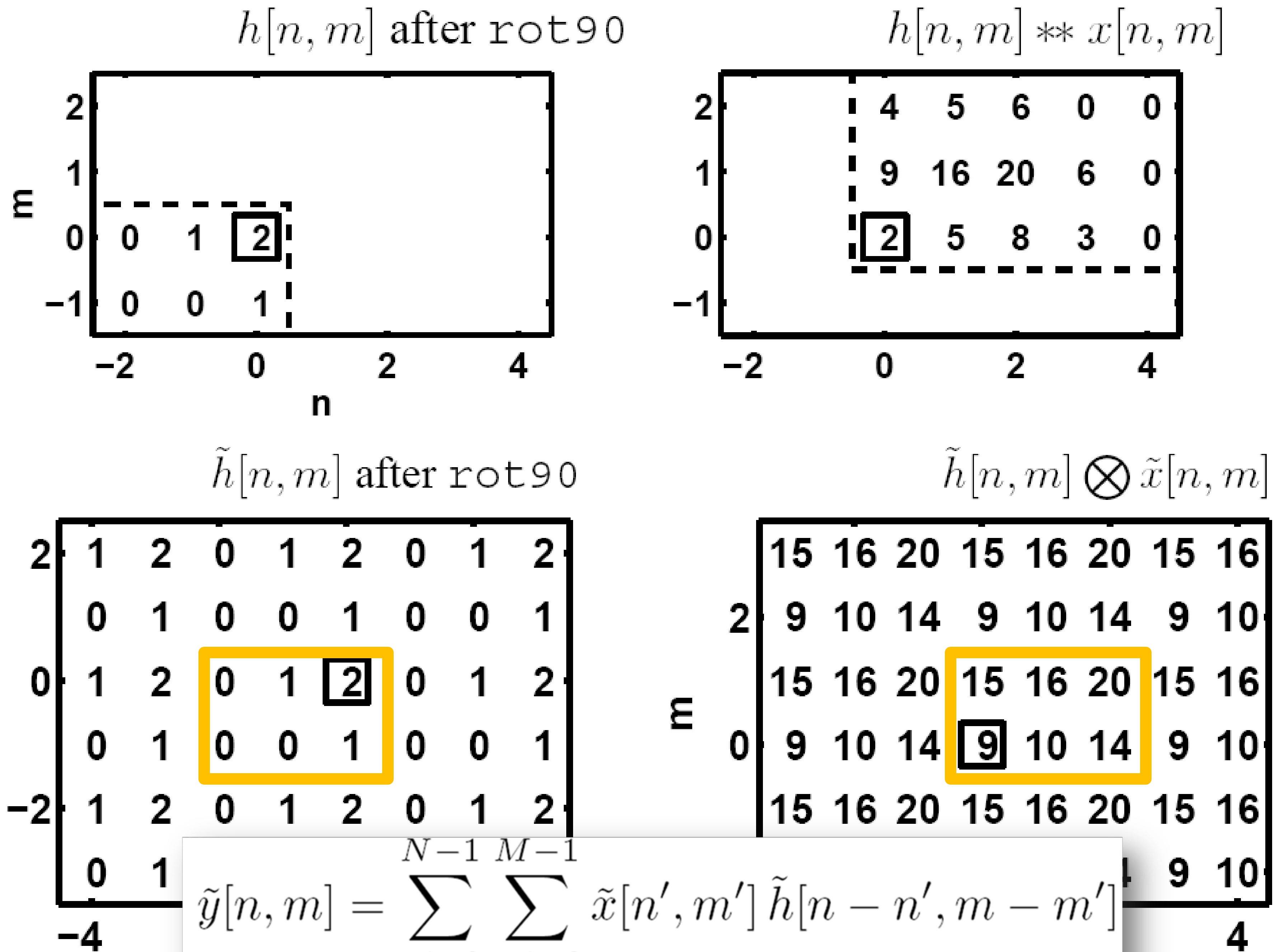

h(n)
$\tilde{h}(n)$
1
0.8
0.6
0.4
0.2
0
−4
−2
0
2
4
6
8





- The linear convolution of a 5-point sequence with a 3-point sequence yields a 7-point sequence.
- Periodic convolution of two 5-periodic sequences is still 5-periodic.


$\tilde{h}[n,m]$
$\tilde{x}[n,m]$
1 0 0 1 0 0 1 0
2 1 0 2 1 0 2 1
1 0 0 1 0 0 1 0
2 1 0 2 1 0 2 1
1 0 0 1 0 0 1 0
2 1 0 2 1 0 2 1
4 5 6 4 5 6 4 5
1 2 3 1 2 3 1 2
4 5 6 4 5 6 4 5
1 2 3 1 2 3 1 2
4 5 6 4 5 6 4 5
1 2 3 1 2 3 1 2
$h[n,m]$
$x[n,m]$
1 0 0
2 1 0
4 5 6
1 2 3



$$\tilde{y}[n,m] = \sum_{n'=0}^{N-1} \sum_{m'=0}^{M-1} \tilde{x}[n',m']\,\tilde{h}[n-n',m-m']$$

Discrete Fourier transform (DFT)

- What is the DFT of a $x[n,m]$ with $N \times M$ finite-support signal ?

- Let's explore that using DFS

- First let's look at relationship between periodic and finite-support signals

Finite support signals

• Given a 2D DS signal x[n,m], we can form periodic signals from x[n,m] in two distinct ways:

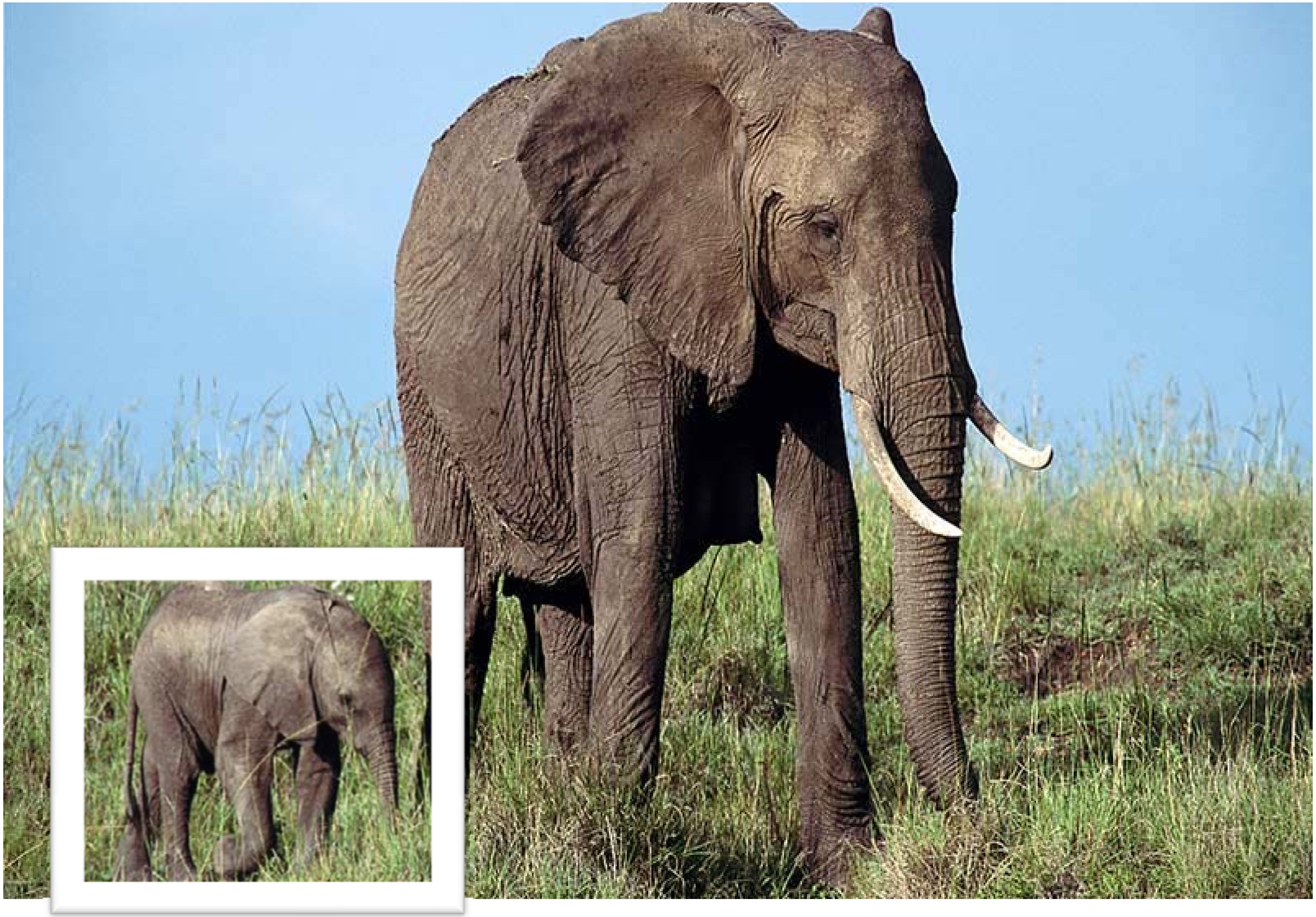
1. (N,M)-**point circular extension**

$$\tilde{x}[n,m] \triangleq x[n \bmod N,\ m \bmod M]$$

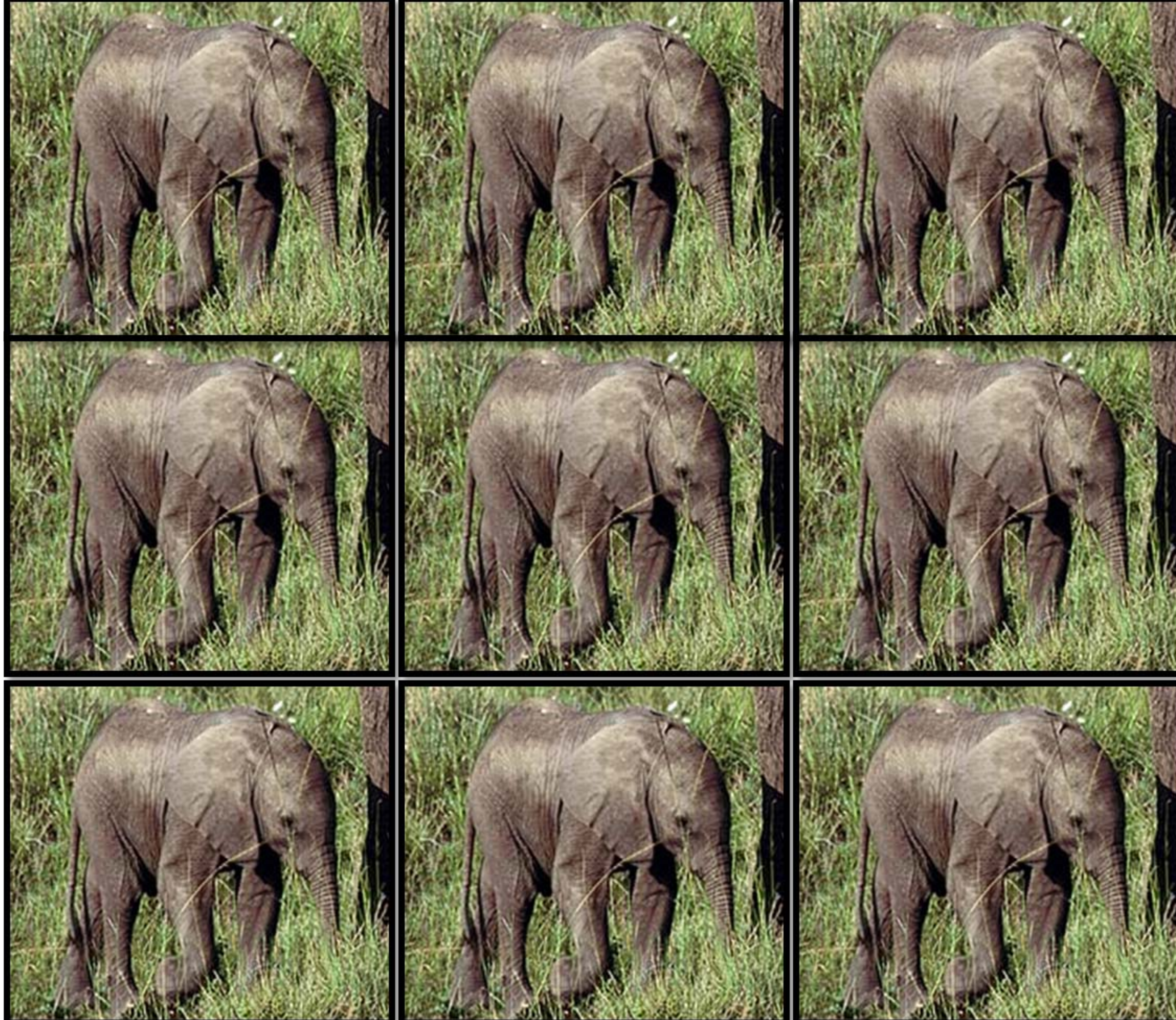
38 mod 12 = 2

2 mod 12 = 2

•circular extension depends only on the values of x[n,m] for n = 0, . . . ,N − 1 and m = 0, . . . ,M − 1



x[n,m]



Finite support signals

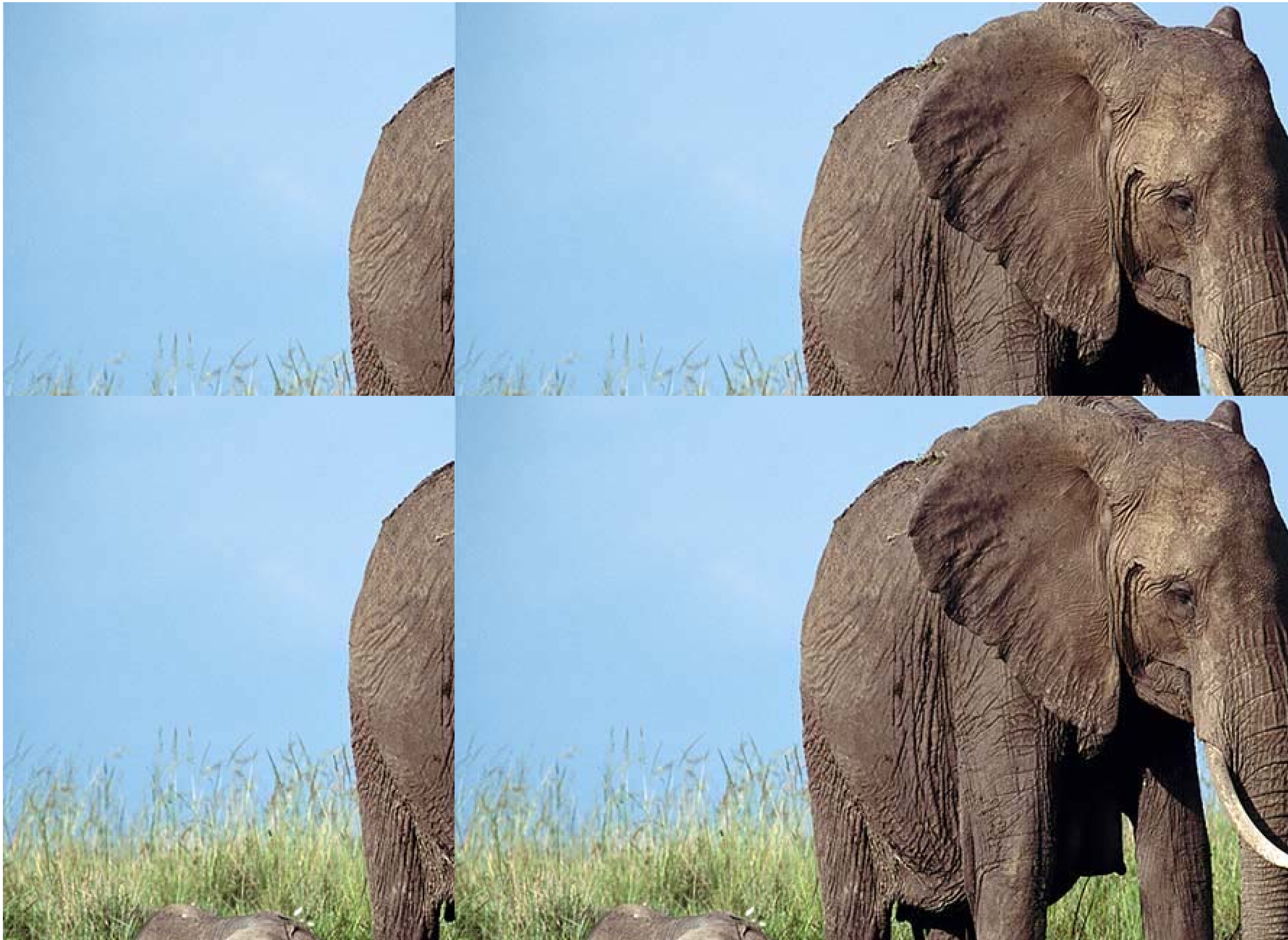
• Given a 2D DS signal x[n,m], we can form periodic signals from x[n,m] in two distinct ways:

2. (N,M)-**point periodic superposition**

$$x_{\mathrm{ps}}[n,m] = x[[n,m]]_{(N,M)}$$

$$= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x[n-kN, m-lM]$$

This periodic signal depends on all of the values of the original signal x[n,m].



Finite support signals

1. (N,M)-**point circular extension** $\tilde{x}[n,m]$
2. (N,M)-**point periodic superposition** $x_{\text{ps}}[n,m]$

What if x[n,m] is a N × M finite-support signal <or>
x[n,m] nonzero only for n = 0, . . . ,N − 1 and m = 0, . . . ,M − 1

$$x_{\text{ps}}[n,m] \;=\; \tilde{x}[n,m]$$

Recover original signal:

rectangular window in the space domain

$$x[n,m] = \tilde{x}[n,m]\boxed{R_{NM}[n,m]}$$

Discrete Fourier transform (DFT)

- What is the DFT of a $x[n,m]$ with $N \times M$ finite-support signal ?

- Let's explore that using DFS

- First let's look at relationship between periodic and finite-support signals

Discrete Fourier transform (DFT)

- what is the DFS of the periodic extension signal $\tilde{x}[n,m]$?

$$\tilde{x}[n,m] = \frac{1}{NM}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1}\tilde{X}[k,l]\,\mathrm{e}^{\imath 2\pi(kn/N+lm/M)}$$

$$\tilde{X}[k,l] = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\boxed{\tilde{x}[n,m]}\,\mathrm{e}^{-\imath 2\pi(kn/N+lm/M)}$$

$$= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\boxed{x[n,m]}\,\mathrm{e}^{-\imath 2\pi(kn/N+lm/M)}$$

Non zero for all k,l in Z

Discrete Fourier transform (DFT)

$$\tilde{X}[k,l] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x[n,m]\, \mathrm{e}^{-\imath 2\pi(kn/N + lm/M)}$$

Truncate this periodic set of coefficients to form a finite-extent set:

$$\boxed{X[k,l]} \triangleq \tilde{X}[k,l]\, R_{NM}[k,l]$$

$$= \begin{cases} \tilde{X}[k,l], & k = 0,\ldots,N-1,\ l = 0,\ldots,M-1 \\ 0, & \text{otherwise.} \end{cases}$$

Discrete Fourier Transform (DFT) of x[n,m]

Discrete Fourier transform (DFT)

we can recover x[n,m] from X[k, l]:

$$x[n,m] = \boxed{\tilde{x}[n,m]} R_{NM}[n,m]$$

DFS

$$= \boxed{\frac{1}{NM}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1}\tilde{X}[k,l]\,\mathrm{e}^{\imath 2\pi(kn/N+lm/M)}} R_{NM}[n,m]$$

$$= \begin{cases} \frac{1}{NM}\sum_{k=0}^{N-1}\sum_{l=0}^{M-1} X[k,l]\,\mathrm{e}^{\imath 2\pi(kn/N+lm/M)} & n=0,\ldots,N-1, m=0,\ldots,M-1 \\ 0, & \text{otherwise.} \end{cases}$$

Discrete Fourier transform (DFT)

To RECAP: for a N,M finite support signal x[n,m]:

$$X[k,l] \triangleq = \begin{cases} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x[n,m]\, \mathrm{e}^{-\imath 2\pi (kn/N + lm/M)} & k = 0,\dots,N-1, l = 0,\dots,M-1 \\ 0, & \text{otherwise.} \end{cases}$$

Undefined (?)

$$x[n,m] = \begin{cases} \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} X[k,l]\, \mathrm{e}^{\imath 2\pi (kn/N + lm/M)} \\ ?, \end{cases}$$

Why DFT is important?

- **General goal:** perform efficient linear convolution
- Perform convolution as product of DFTs
- **Pros:** DFT can be implemented using the FFT (fast fourier transform)
    - FFT is very efficient (fast!)

- **Cons:** DFT perform circular convolution
    - Compensate the wrap-around effect
- **Cons:** Online-memory storage
    - Use the overlap-add method or overlap-save method

Relationship between DFT and DSFT

For a finite-support signal x[n,m], relate DFT to the DSFT as follows:

$$X[k,l] \stackrel{\text{DFT}}{=} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x[n,m]\, \mathrm{e}^{-\imath 2\pi (kn/N + lm/M)}$$

$$\stackrel{\text{Due to finite support}}{=} \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} x[n,m]\, \mathrm{e}^{-\imath 2\pi (kn/N + lm/M)}$$

$$\stackrel{\text{DSFT}}{=} X(\omega_{\mathrm{X}}, \omega_{\mathrm{Y}})\big|_{\omega_{\mathrm{X}} = 2\pi k/N,\ \omega_{\mathrm{Y}} = 2\pi l/M}$$

DFT Properties

- **Linearity:** if x[n,m] and y[n,m] have the same period:

$$\alpha\, x[n,m] + \beta\, y[n,m] \stackrel{\mathrm{DFT}}{\longleftrightarrow} \alpha\, X[k,l] + \beta\, Y[k,l]$$

- **Separability:**

$$x[n,m] = x_1[n]\, x_2[m] \stackrel{\mathrm{DFT}}{\longleftrightarrow} X_1[k]\, X_2[l]$$

- **Parseval's theorem:**

$$\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x[n,m]\, y^*[n,m] = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} X[k,l]\, Y^*[k,l]$$

DFT Properties

complex conjugate

$$x^*[n,m] \stackrel{\text{DFT}}{\longleftrightarrow} X^*[-k \bmod N,\ -l \bmod M]$$

$$\tilde{x}^*[n,m] \stackrel{\text{DFS}}{\longleftrightarrow} \tilde{X}^*[-k,-l]$$

DFT Properties

Circular shift of n_o

1D

$$\{x[0], \ldots, x[N-1]\} \implies$$

$$\{x[n_0], \ldots, x[N-1], x[0], \ldots, x[n_0-1]\},$$

$$x[(n-n_0) \bmod N]$$

What's the DST of a circular shifted signal?

$$x[(n-n_0) \bmod N, (m-m_0) \bmod M]$$

$$\stackrel{\text{DFT}}{\longleftrightarrow} e^{-\imath 2\pi (kn_0/N + lm_0/M)} X[k,l]$$

DFT Properties

Circular frequency shift

$$\mathrm{e}^{\imath 2\pi (k_0 n/N + l_0 m/M)}\, x[n,m]$$

$$\overset{\mathrm{DFT}}{\longleftrightarrow} X[(k-k_0) \bmod N,\, (l-l_0) \bmod M]$$

DFT Properties

Circular frequency shift

- circular space reversal of x[n,m] is x[−n mod N, −m mod M]

$$x[-n \bmod 4, -m \bmod 3] =$$

$$x[n,m] = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \underline{9} & 10 & 11 & 12 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 8 & 7 & 6 \\ 1 & 4 & 3 & 2 \\ \underline{9} & 12 & 11 & 10 \end{bmatrix}$$

x[n,m] is **circularly even** if x[−n mod N, −m mod M] = x[n,m],

DFT Properties

Symmetry properties

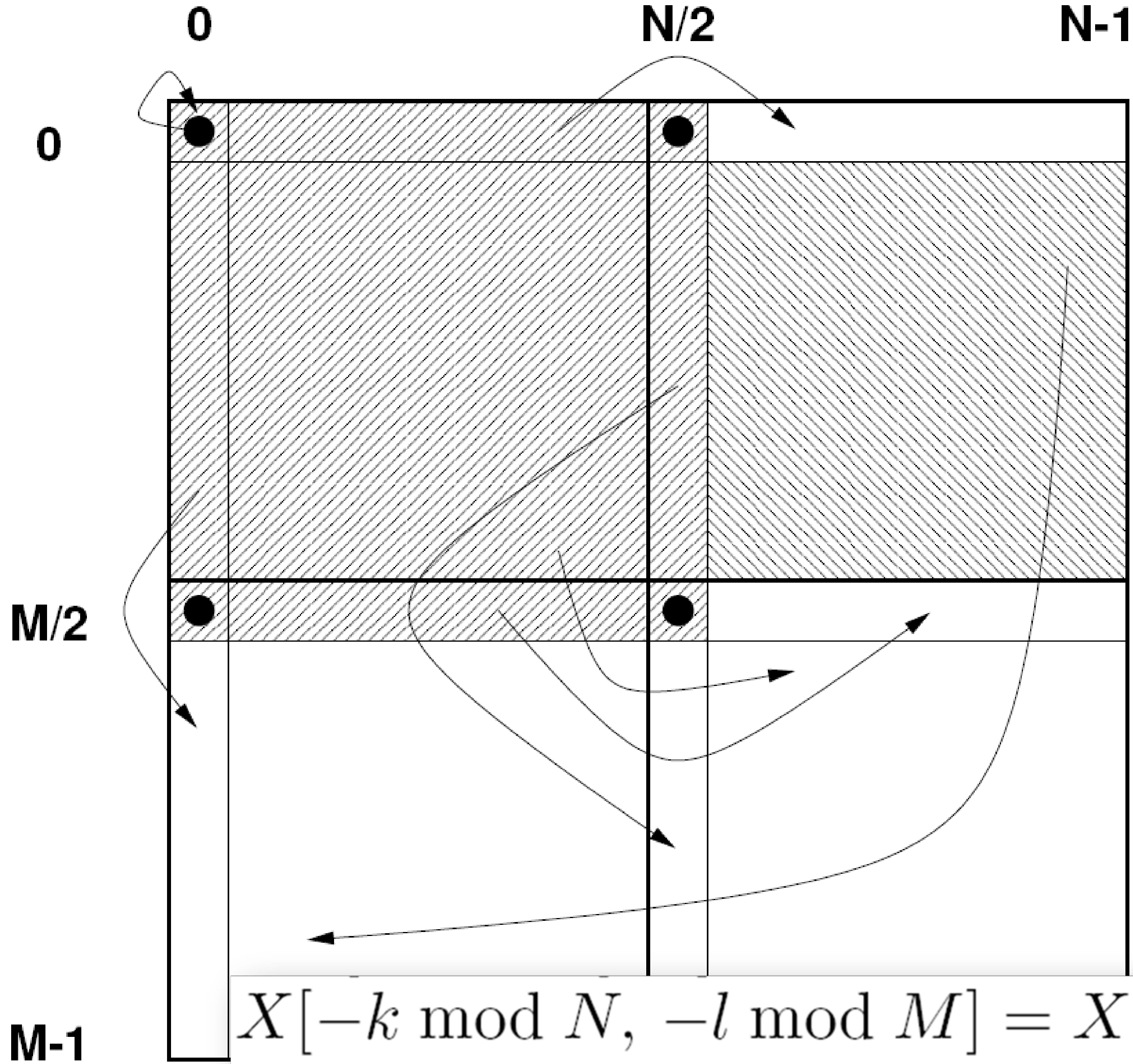
$$x^*[n,m] \xleftrightarrow{\text{DFT}} X^*[-k \bmod N,\ -l \bmod M]$$

•If x[n,m] is **circularly even,** then X[k, l] is also **circularly even:**

$$x[-n \bmod N,\ -m \bmod M] = x[n,m]$$

•If x[n,m] is real, then X[k, l] is Hermitian circularly symmetric:

$$X[-k \bmod N,\ -l \bmod M] = X^*[k,l]$$



$$X[-k \bmod N,\ -l \bmod M] = X^*[k,l]$$

DFT Properties: Convolution

- h[n,m] and x[n,m] are both finite-support signals
- DFT coefficients H[k, l] and X[k, l]

modified **circular convolution**

??

$$\boxed{\boxed{h[n,m] \bigotimes x[n,m]} \stackrel{\mathrm{DFT}}{\longleftrightarrow} H[k,l]\, X[k,l],}$$

y[n,m]   DFT → Y [k, l] = H[k, l]X[k, l]

$$y[n,m] = \boxed{\tilde{y}[n,m]}\, R_{NM}[n,m] \qquad \boxed{(\tilde{h}[n,m] \bigotimes \tilde{x}[n,m])}$$

$$\tilde{y}[n,m] \stackrel{\mathrm{DFS}}{\longleftrightarrow} \tilde{Y}[k,l] = H[k,l]\, X[k,l] = \tilde{H}[k,l]\, \tilde{X}[k,l]$$

DFT Properties: Convolution

- h[n,m] and x[n,m] are both finite-support signals
- DFT coefficients H[k, l] and X[k, l]

modified **circular convolution**

$$\boxed{h[n,m] \bigotimes x[n,m] \overset{\text{DFT}}{\longleftrightarrow} H[k,l]\,X[k,l],}$$

$$= \begin{cases} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} \tilde{h}[k,l]\,\tilde{x}[n-k,m-l] \\ 0, \end{cases} =$$

$$\begin{cases} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} h[k,l]\,x[(n-k) \bmod N,\,(m-l) \bmod M] & n=0,\dots,N-1, m=0,\dots,M-1 \\ 0, & \text{otherwise.} \end{cases}$$

DFT Properties: Convolution

- h[n,m] and x[n,m] are both finite-support signals
- DFT coefficients H[k, l] and X[k, l]

modified **circular convolution**

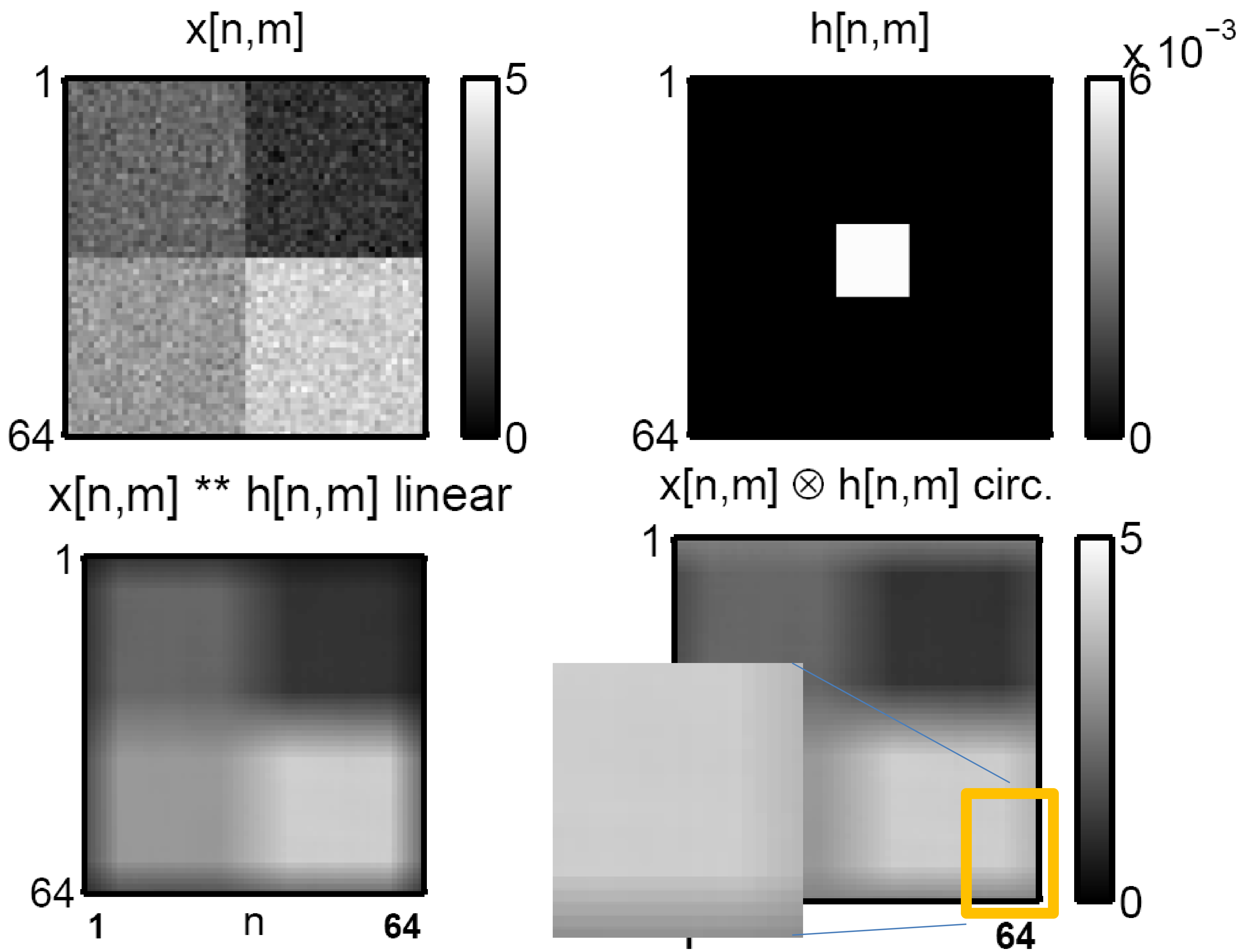
$$\boxed{h[n,m] \bigotimes x[n,m] \overset{\text{DFT}}{\longleftrightarrow} H[k,l]\, X[k,l],}$$

$$= \begin{cases} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} \tilde{h}[k,l]\, \tilde{x}[n-k, m-l] \\ 0, \end{cases}$$

- Form their periodic extension,
- Perform circular convolution
- Truncate the result

DFT Properties: Convolution

- The effect of circular convolution is a wrap around effect:
  - signal values from one border of an image can “wrap around” to affect values on the other side (after filtering via a DFT)


x[n,m]
h[n,m]
x 10^−3
x[n,m] ** h[n,m] linear
x[n,m] ⊗ h[n,m] circ.
1
64
n
5
0
6

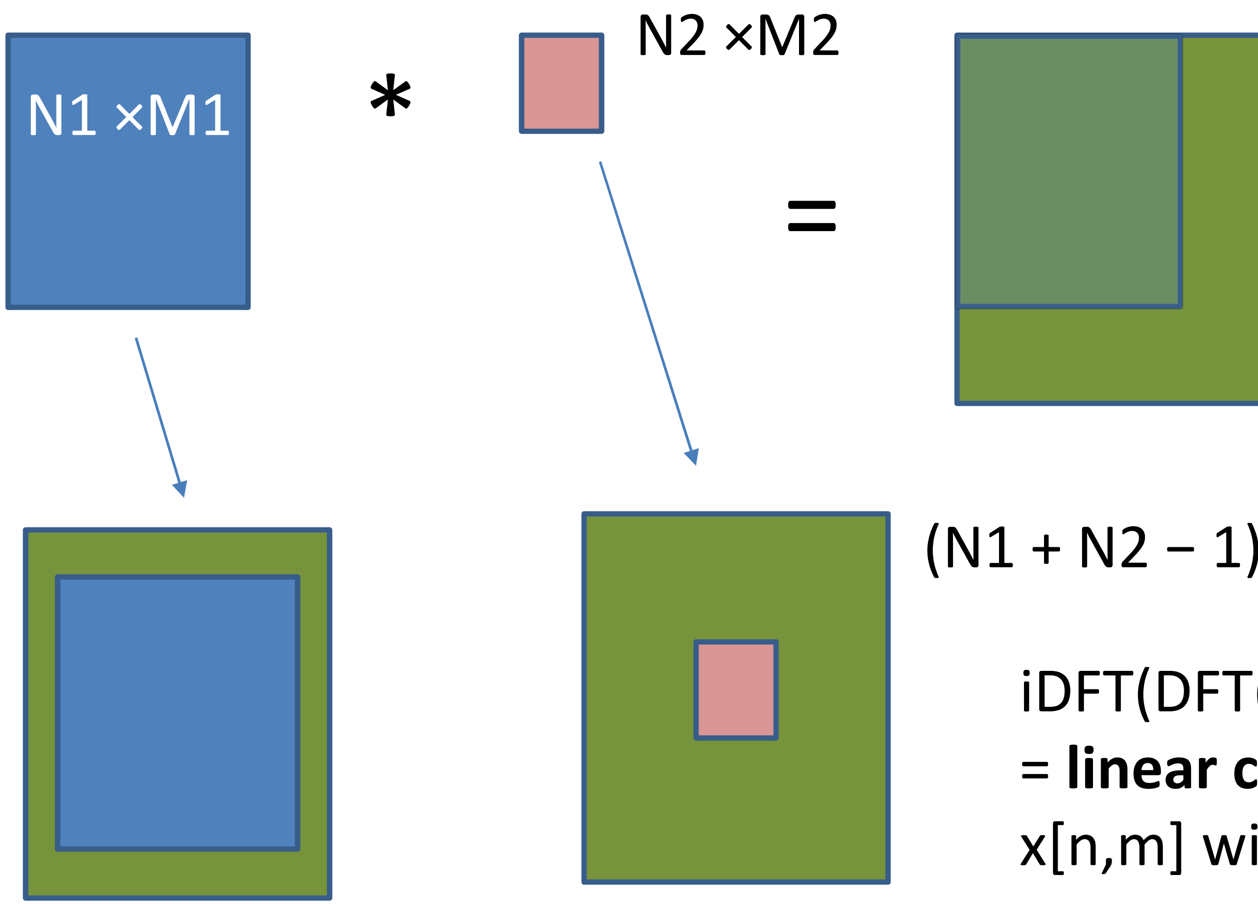
DFT Properties: Convolution

- The effect of circular convolution is a wrap around effect:
  - signal values from one border of an image can “wrap around” to affect values on the other side (after filtering via a DFT)

- Undesirable

- Possible solutions:
  - Accept the wrap around, and ignore the edges of the image
  - Zero padding
  - Use the overlap-add method or overlap-save method

Zero padding



iDFT(DFT(x) .* DFT(h))
= **linear convolution of**
x[n,m] with h[n,m].

Why DFT is important?

- **General goal:** perform efficient linear convolution
- Perform convolution as product of DFTs
- **Pros:** DFT can be implemented using the FFT (fast fourier transform)
  - FFT is very efficient (fast!)
- **Cons:** DFT perform circular convolution
  - Compensate the wrap-around effect
- **Cons:** Online-memory storage
  - Use the overlap-add method or overlap-save method

Fast Fourier transforms (FFT)

- Brute-force evaluation of the 2D DFT would require O((NM)2) flops

$$X[k,l] \triangleq$$

$$= \begin{cases} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x[n,m]\, \mathrm{e}^{-\imath 2\pi (kn/N + lm/M)} & k = 0,\ldots,N-1, l = 0,\ldots,M-1 \\ 0, & \text{otherwise.} \end{cases}$$

- DFT is a **separable operation**

  → we can reduce greatly the computation

Fast Fourier transforms (FFT)

- DFT is a **separable operation:**

$$X[k,l] \;=\; \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x[n,m]\,\mathrm{e}^{-\imath 2\pi(kn/N+lm/M)}$$

$$=\; \sum_{n=0}^{N-1} \mathrm{e}^{-\imath 2\pi kn/N}\left[\sum_{m=0}^{M-1} x[n,m]\,\mathrm{e}^{-\imath 2\pi lm/M}\right]$$

- Apply the 1D DFT to each column of the image, and then apply the 1D DFT to each row of the result.

- Use the **fast Fourier transform (1-D FFT)** for these 1D DFTs!

Fast Fourier transforms (FFT)

- FFT computational efficiency:

- **inner set of 1D FFTs** : N **O**(M logM)
- **outer set of 1D FFTs**: M **O**(N logN)

•**Total: O**(MN logMN) flops

- A critical property of FFT is that $N = 2^k$ with k =integer

If x = 512x512 image →
saving is a factor of 15000 relative to the brute-force 2D DFT!

Matlab: fft2 = fft(fft(x).').'

FFT & Efficiency

- **General goal:** perform efficient linear convolution
- Perform convolution as product of DFTs
- **Pros:** DFT can be implemented using the FFT (fast fourier transform)
  - FFT is very efficient (fast!)

- **Cons:** DFT perform circular convolution
  - Compensate the wrap-around effect
- **Cons:** Online-memory storage
  - Use the overlap-add method or overlap-save method

FFT & Efficiency

- Suppose we wish to convolve a **256 × 256** image with a **17 × 17** filter.
- The result will be **272×272**.
- The smallest prime factors of 272 is **2**.
- So one could pad to a **512 × 512 image**
- Note: only 28% of the final image would be the part we care about - the rest would be zero in exact arithmetic.

- Handling to a 512 × 512 image requires **much memory**

→ Use overlap-add method

Overlap-add method

- Decompose image into small-sized blocks:

$$x[n,m] = \sum_k x_k[n,m]$$

$$y[n,m] \quad = \quad h[n,m] ** x[n,m]$$

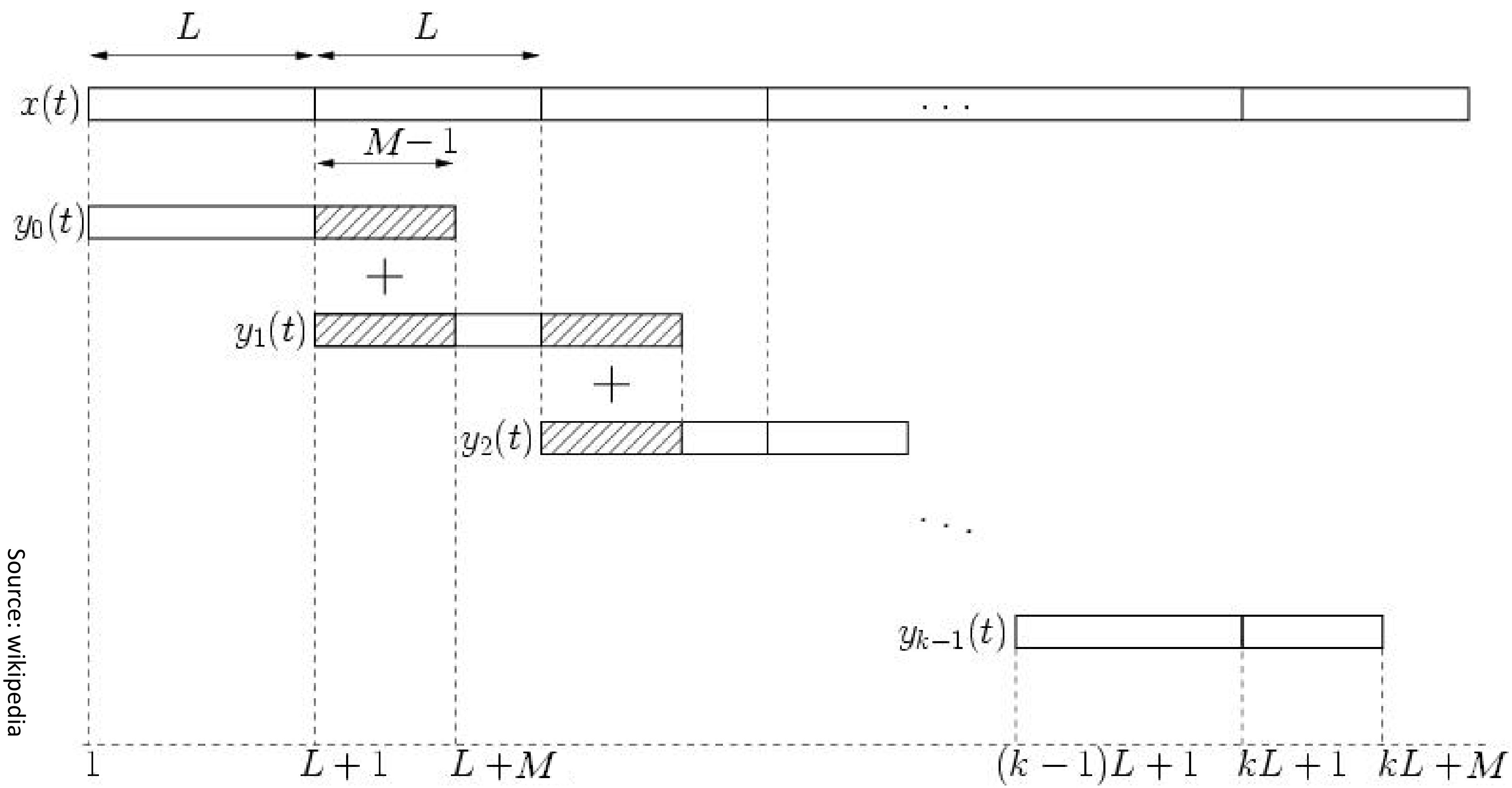
$$= h[n,m] ** \sum_k x_k[n,m]$$

Convolution is linear

$$= \quad \sum_k h[n,m] ** x_k[n,m] \triangleq \sum_k y_k[n,m]$$

$$\text{where } y_k[n,m] \triangleq h[n,m] ** x_k[n,m]$$

Overlap-add method



Source: wikipedia

- •Perform several limited-memory FFTS and sum up results
- •We must be careful when we sum things up

Matrix representation of 1-D DFT

The DFT is a linear transformation of the sequence x[0], . . . , x[N − 1] to the coefficients X[0], . . . ,X[N − 1].

$$\boldsymbol{X} = \boldsymbol{W}\boldsymbol{x}$$

$$\boldsymbol{x} = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \text{ and } \boldsymbol{X} = \begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix}$$

$$\boldsymbol{W}_{kn} = W_N^{kn} \text{ where } W_N \triangleq \mathrm{e}^{-\imath 2\pi/N} \qquad X[k] = \sum_{n=0}^{N-1} x[n]\,\mathrm{e}^{-\imath 2\pi nk/N}$$

Coefficients of W =N × N matrix orthogonal matrix

Matrix representation of 1-D DFT

The DFT is a linear transformation of the sequence x[0], . . . , x[N − 1] to the coefficients X[0], . . . ,X[N − 1].

$$\boldsymbol{x} = \boldsymbol{W}^{-1}\boldsymbol{X} = \frac{1}{N}\boldsymbol{W}'\boldsymbol{X}$$

Since W =N × N matrix orthogonal matrix

$$\boldsymbol{W}'\boldsymbol{W} = N\boldsymbol{I} = \boldsymbol{W}\boldsymbol{W}'$$

$$\boldsymbol{W}^{-1} = \frac{1}{N}\boldsymbol{W}'$$

Matrix representation of 2-D DFT

$$\boldsymbol{X} = (\boldsymbol{W}_M \otimes \boldsymbol{W}_N)\boldsymbol{x}$$

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{bmatrix}$$

$\boldsymbol{A} \otimes \boldsymbol{B}$ denotes the **Kronecker product**

$$\boldsymbol{x} = [x[0,0]\ \ x[1,0]\ \ \dots\ \ x[N-1,0]\ \ x[0,1]\ \ \dots\ \ x[N-1,1]\ \ \dots$$

$$\dots\ \ x[0,M-1]\ \ \dots\ \ x[N-1,M-1]]^T$$

Why a Matrix representation is useful?

- x[n] includes both a deterministic signal component and an additive random noise component,

$$x[n] = \mu[n] + \varepsilon[n]$$

$\varepsilon[n]$ is an uncorrelated sequence of random variables all having the same variance σ^2.

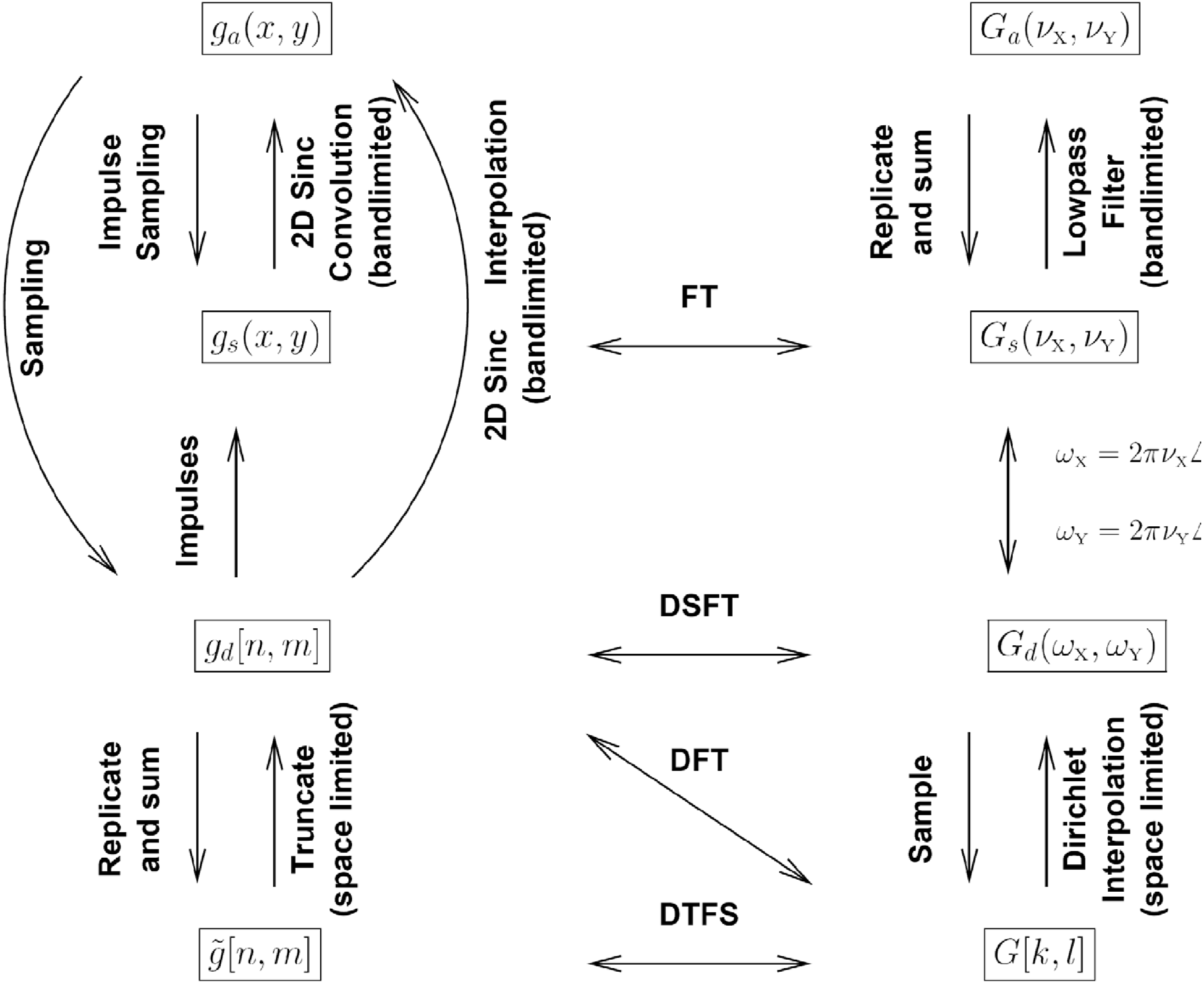
$$\mathsf{Cov}\{\boldsymbol{x}\} = \sigma^2 \boldsymbol{I}$$

- Let’s take N-point DFT of x[n]. What’s Cov (X)?

$$\mathsf{Cov}\{\boldsymbol{X}\} = \mathsf{Cov}\{\boldsymbol{W}\boldsymbol{x}\} = \boldsymbol{W}\,\mathsf{Cov}\{\boldsymbol{x}\}\,\boldsymbol{W}'$$

$$= \boldsymbol{W}\sigma^2\boldsymbol{I}\boldsymbol{W}' = \sigma^2\boldsymbol{W}\boldsymbol{W}' = \sigma^2 N\boldsymbol{I}$$

signal values are uncorrelated (w ithsame variance) then the DFT coefficients are also uncorrelated (with same variance)



$g_a(x, y)$
$G_a(\nu_X, \nu_Y)$
Sampling
Impulse Sampling
2D Sinc Convolution (bandlimited)
2D Sinc Interpolation (bandlimited)
Replicate and sum
Lowpass Filter (bandlimited)
FT
$g_s(x, y)$
$G_s(\nu_X, \nu_Y)$
Impulses
$\omega_X = 2\pi\nu_X$
$\omega_Y = 2\pi\nu_Y$
DSFT
$g_d[n, m]$
$G_d(\omega_X, \omega_Y)$
Replicate and sum
Truncate (space limited)
DFT
Sample
Dirichlet Interpolation (space limited)
DTFS
$\tilde{g}[n, m]$
$G[k, l]$

- We skipped the Discrete cosine transforms (DCT)
    - Useful in image compression

- We'll get back to that if time allows

- **Next lecture:** Image enhancement