Image Transforms summary

Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 3.4.1

Artificial Intelligence & Information Analysis Lab

Contents

Artificial Intelligence & Information Analysis Lab

2D Discrete Space Fourier Transform

2D discrete space Fourier transform is defined as:

$$X(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x(n_1, n_2) \exp(-i\omega_1 n_1 - i\omega_2 n_2),$$

$$x(n_1, n_2) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} X(\omega_1, \omega_2) \exp(i\omega_1 n_1 + i\omega_2 n_2)\, d\omega_1 d\omega_2 .$$

2D Discrete Space Fourier Transform

2D discrete space Fourier transform properties.

- ***2D linear convolution***:

$$x(n_1, n_2) ** h(n_1, n_2) \leftrightarrow X(\omega_1, \omega_2) \cdot H(\omega_1, \omega_2),$$

Introduction

Image transforms are represented by transform matrices $\mathbf{A}$:

$$\mathbf{X} = \mathbf{A}\mathbf{x},$$

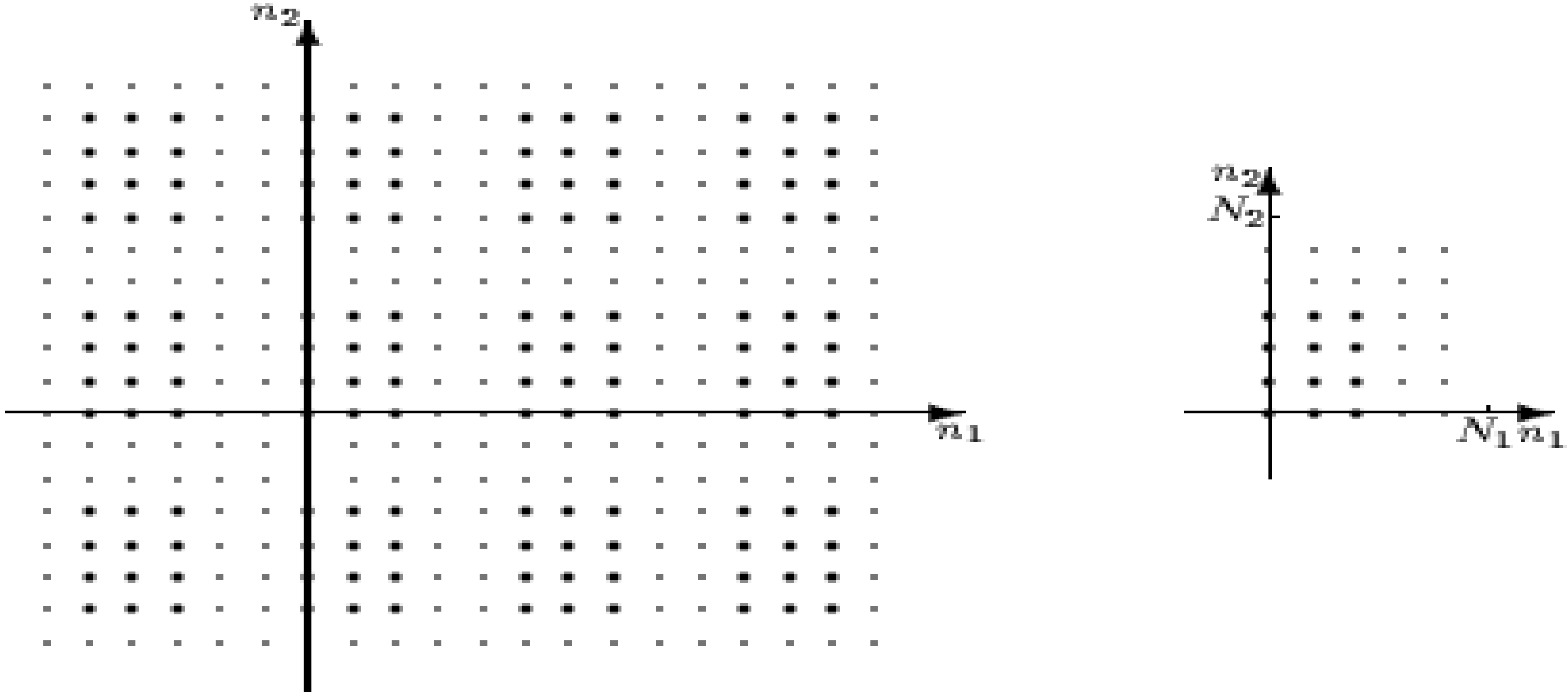
- $\mathbf{x}$ and $\mathbf{X}$ are the original and transformed image respectively.

In most cases the transform matrices are ***unitary***:

$$\mathbf{A}^{-1} = \mathbf{A}^{*T}.$$

- The columns of $\mathbf{A}^{*T}$ are the ***basis vectors*** of the transform.
- In image transforms, they correspond to ***basis images***.
- The most popular image transforms are:
  - Discrete Fourier Transform (DFT).
  - Discrete Cosine Transform (DCT).

2D Discrete Fourier Transform



Rectangularly periodic sequence and its fundamental period $N_1 = 5, N_2 = 6$.

2D Discrete Fourier Transform

- Another commonly used form of the DFT is:

$$X(k_1,k_2)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}x(n_1,n_2)W_{N_1}^{n_1k_1}W_{N_2}^{n_2k_2}$$

$$x(n_1,n_2)=\frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}X(k_1,k_2)W_{N_1}^{-n_1k_1}W_{N_2}^{-n_2k_2},$$

where:

$$W_{N_i}=\exp\left(-i\frac{2\pi}{N_i}\right),\qquad i=1,2.$$

Row-Column FFT algorithm

- Transform magnitude is given by:

$$X_M(k_1,k_2)=\sqrt{X_R(k_1,k_2)^2+X_I(k_1,k_2)^2}\,.$$

- The transform phase is given by:

$$\phi(k_1,k_2)=\tan^{-1}\left(\frac{X_I(k_1,k_2)}{X_R(k_1,k_2)}\right).$$

- The magnitude of the transform provides useful information about the frequency content of an image

2D Discrete Fourier Transform

2D cyclic signal shift is defined by:

$$y(n_1,n_2) = x\left(((n_1-m_1))_{N_1}, ((n_2-m_2))_{N_2}\right),$$

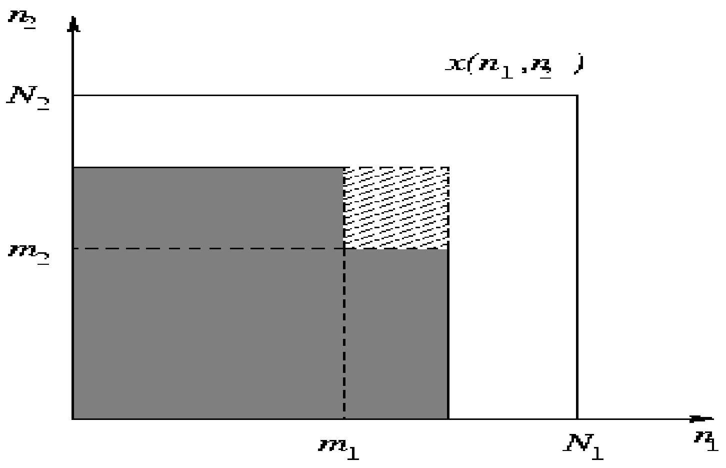
$$((n))_N \triangleq n \bmod N.$$

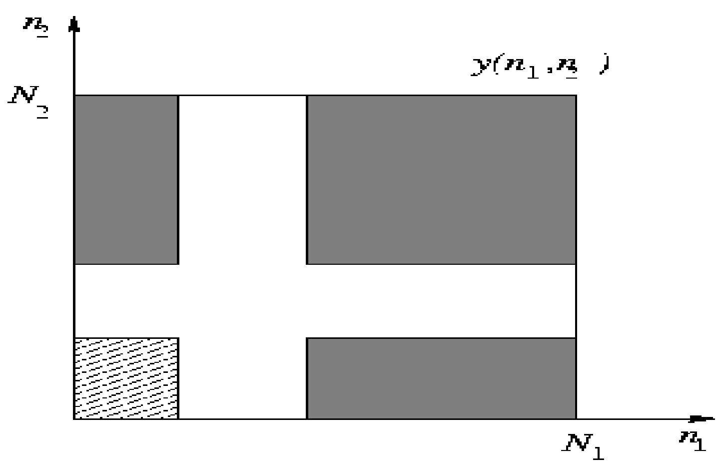
2D cyclic convolution of two signals can be computed by means of the cyclic shift of one of the two signals:

$$y(n_1,n_2) \triangleq x(n_1,n_2) \circledast\circledast h(n_1,n_2) =$$

$$= \sum_{m_1=0}^{N_1} \sum_{m_2}^{N_2-1} x(m_1,m_2) h\left(((n_1-m_1))_{N_1}, ((n_2-m_2))_{N_2}\right).$$

2D Discrete Fourier Transform





Cyclic shift of a 2D sequence.

2D Discrete Fourier Transform

- ***Cyclic convolution***:

$$x(n_1,n_2) \circledast\circledast\, h(n_1,n_2) \leftrightarrow X(k_1,k_2)\cdot H(k_1,k_2),$$

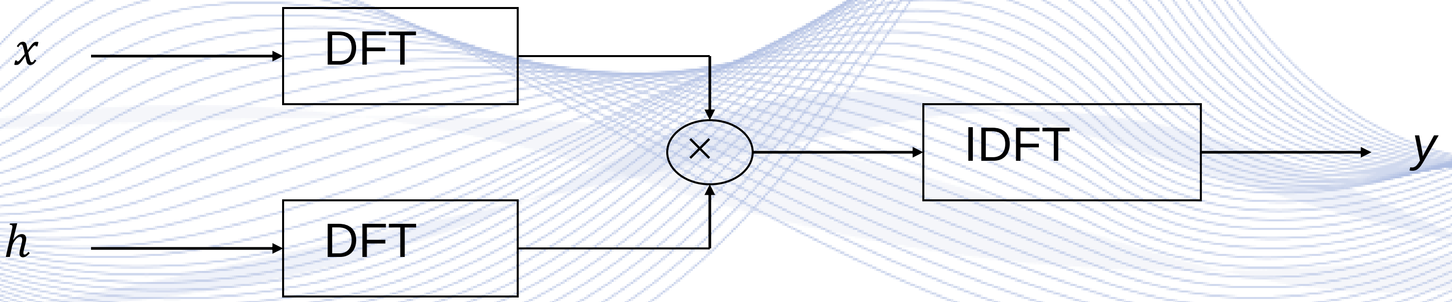
- Signal multiplication:

$$x(n_1,n_2)h(n_1,n_2) \leftrightarrow \frac{1}{N_1N_2}X(k_1,k_2) \circledast\circledast\, H(k_1,k_2).$$

2D Discrete Fourier Transform

3. Calculate the DFTs of the new sequences $x_p(n_1, n_2)$ and $h_p(n_1, n_2)$.
4. Calculate the DFT $Y_p(k_1, k_2)$, as the product of $X_p(k_1, k_2)$ and $H_p(k_1, k_2)$.
5. Calculate $y_p(n_1, n_2)$ by using the inverse DFT. The result of the linear convolution is:

$$y(n_1, n_2) = y_p(n_1, n_2), \qquad (n_1, n_2) \in \mathcal{R}_{L_1 L_2}$$



Convolution calculation using DFTs.

2D Discrete Fourier Transform

- 2D DFTs are calculated fast through ***2D Fast Fourier Transform*** (***FFT***) algorithms.

- Typically, 2D DFT length is chosen to be a power of 2:

$$L_i = 2^{l_i} \geq N_i + M_i - 1, \qquad i = 1, 2.$$

- Both zero padding, the use of power of 2 2D DFT lengths and complex number representations and calculations lead to excessive memory needs.

Row-Column FFT algorithm

- The number of complex multiplications for RCFFT is:

$$C = N_1 \frac{N_2}{2} \log_2 N_2 + N_2 \frac{N_1}{2} \log_2 N_1 = \frac{N_1 N_2}{2} \log_2 (N_1 N_2).$$

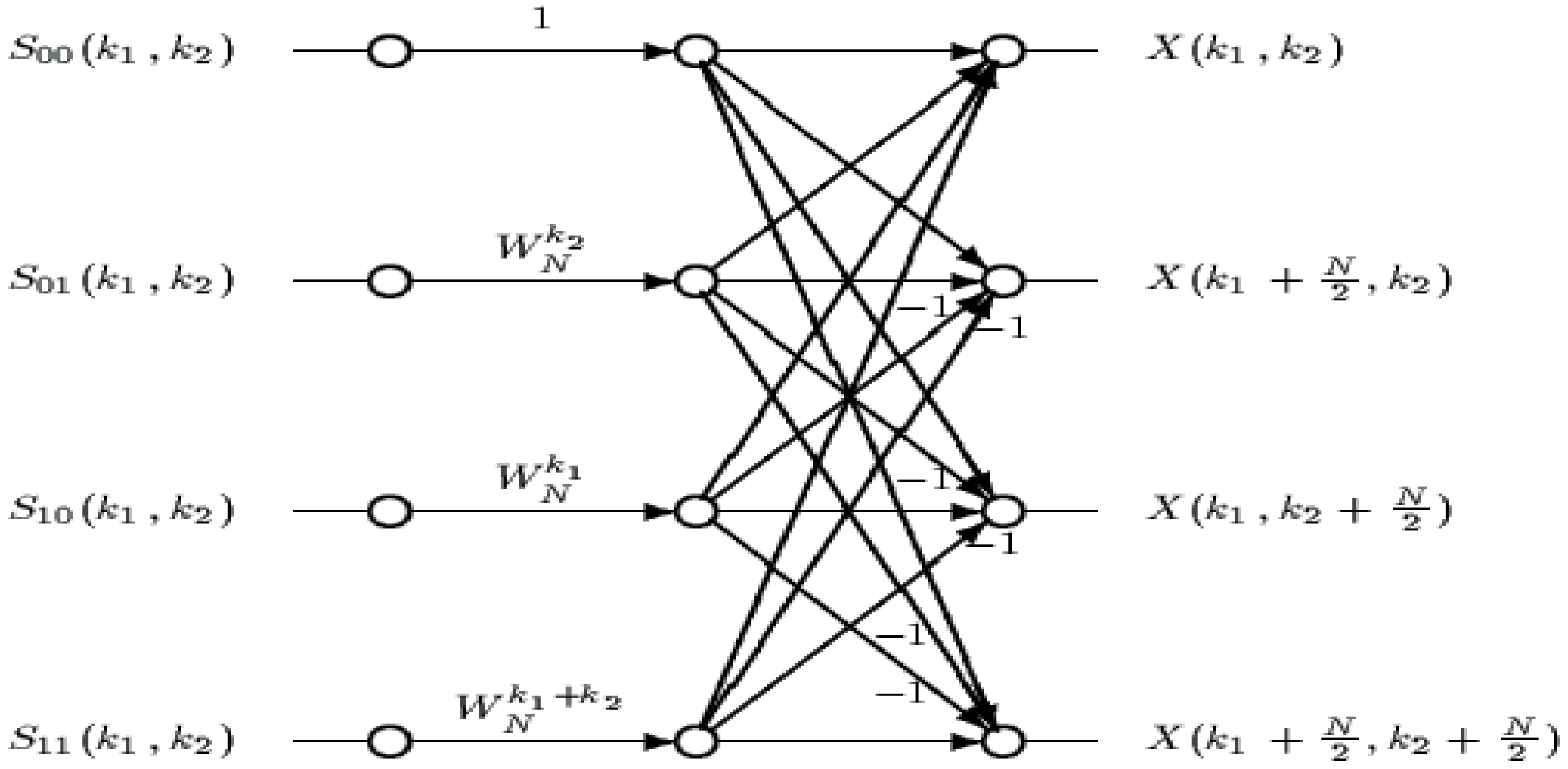
- If radix-2 FFT is used then the number of complex additions for RCFFT is:

$$A = N_1 N_2 \log_2 (N_1 N_2).$$

- The computational complexity is of the order :

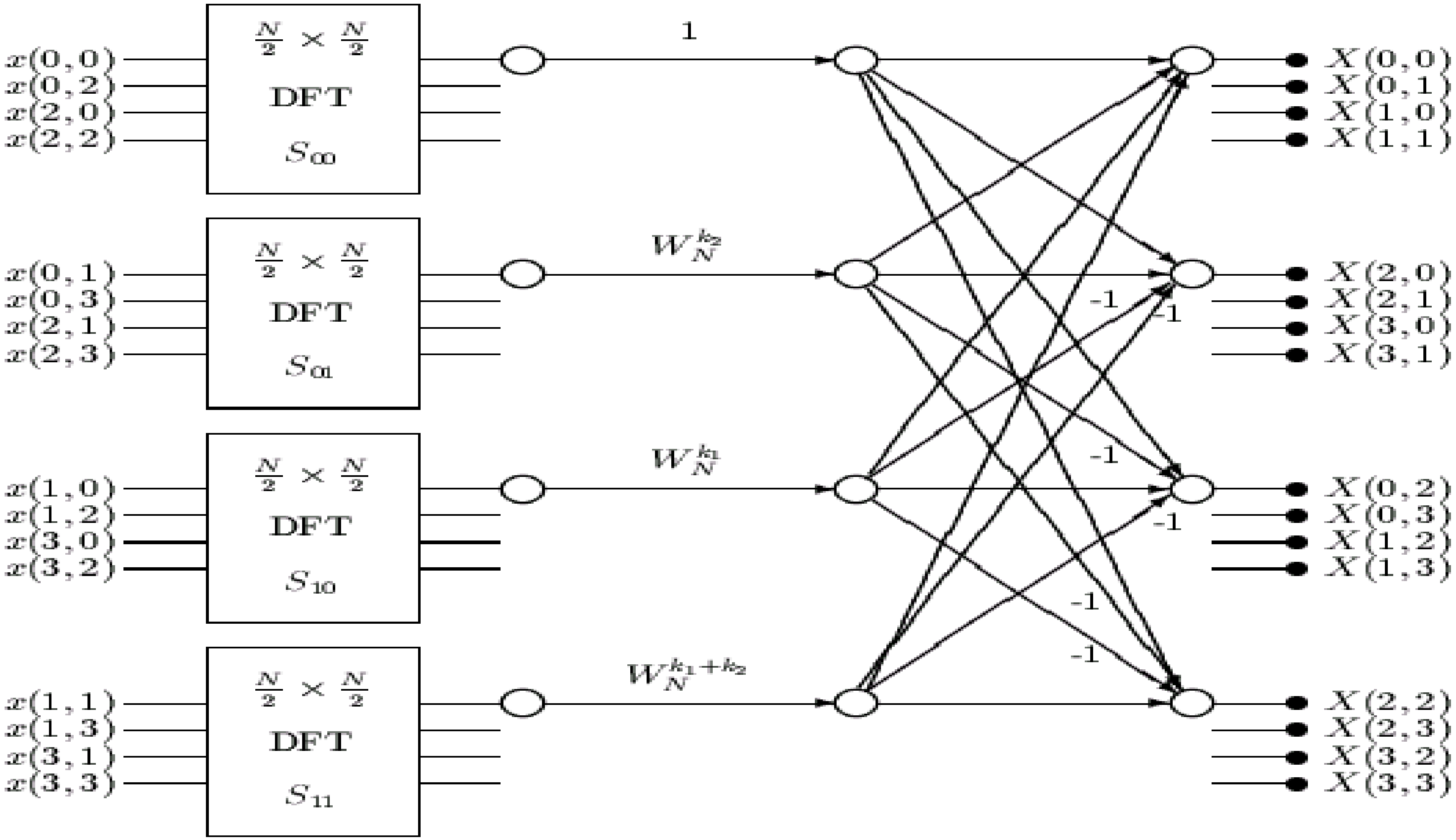
$$O(kN^2 \log_2 N)$$

Vector-radix FFT algorithm



Radix 2×2 butterfly.

Vector-radix FFT algorithm



Flow diagram of a 4×4 vector-radix FFT.

2D FFT algorithms

- The number of complex multiplications of RCFFT and VRFFT for square $N \times N$ images are given by:

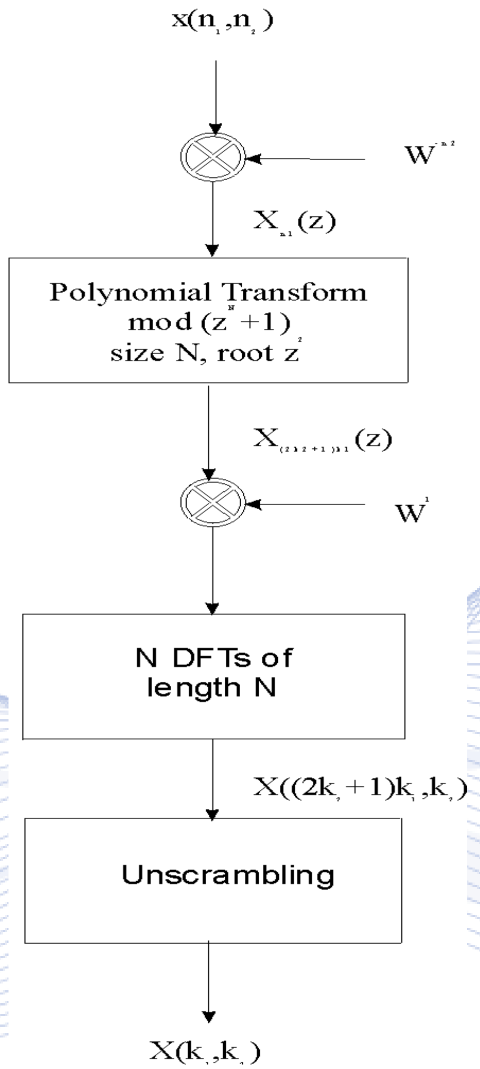
$$C = N^2 \log_2 N.$$

$$C = \frac{3N^2}{4} \log_2 N.$$

respectively.

- Both have computational complexity order $O(kN^2 \log_2 N)$, which is much less than direct 2D DFT computational complexity order $O(kN^4)$.

Polynomial Transform FFT



Polynomial Transform FFT.

2D DFT memory problems

x_{00}	x_{01}	x_{02}	x_{03}	x_{04}	x_{05}	x_{06}	x_{07}
x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}
x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}
x_{30}	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}
x_{40}	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}
x_{50}	x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}
x_{60}	x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}
x_{70}	x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}

→

x_{00}	x_{01}	x_{02}	x_{03}	x_{40}	x_{41}	x_{42}	x_{43}
x_{10}	x_{11}	x_{12}	x_{13}	x_{50}	x_{51}	x_{52}	x_{53}
x_{20}	x_{21}	x_{22}	x_{23}	x_{60}	x_{61}	x_{62}	x_{63}
x_{30}	x_{31}	x_{32}	x_{33}	x_{70}	x_{71}	x_{72}	x_{73}
x_{04}	x_{05}	x_{06}	x_{07}	x_{44}	x_{45}	x_{46}	x_{47}
x_{14}	x_{15}	x_{16}	x_{17}	x_{54}	x_{55}	x_{56}	x_{57}
x_{24}	x_{25}	x_{26}	x_{27}	x_{64}	x_{65}	x_{66}	x_{67}
x_{34}	x_{35}	x_{36}	x_{37}	x_{74}	x_{75}	x_{76}	x_{77}

→

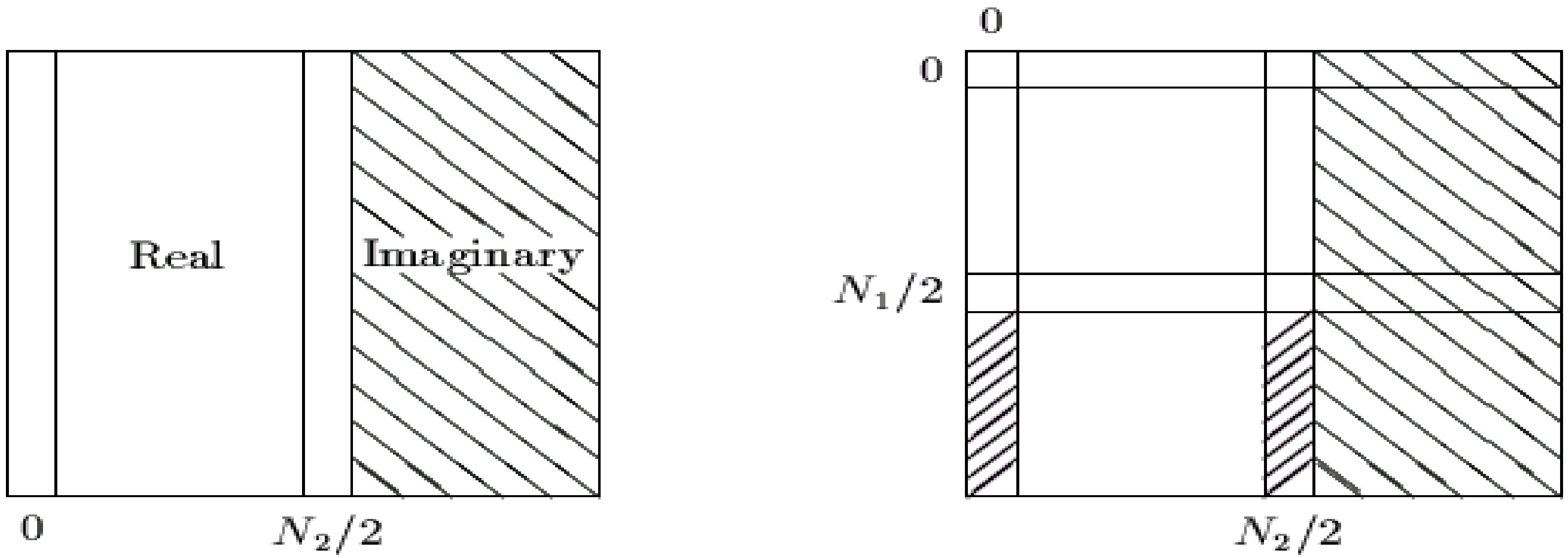
x_{00}	x_{01}	x_{20}	x_{21}	x_{40}	x_{41}	x_{60}	x_{61}
x_{10}	x_{11}	x_{30}	x_{31}	x_{50}	x_{51}	x_{70}	x_{71}
x_{02}	x_{03}	x_{22}	x_{23}	x_{42}	x_{43}	x_{62}	x_{63}
x_{12}	x_{13}	x_{32}	x_{33}	x_{52}	x_{53}	x_{72}	x_{73}
x_{04}	x_{05}	x_{24}	x_{25}	x_{44}	x_{45}	x_{64}	x_{65}
x_{14}	x_{15}	x_{34}	x_{35}	x_{54}	x_{55}	x_{74}	x_{75}
x_{06}	x_{07}	x_{26}	x_{27}	x_{46}	x_{47}	x_{66}	x_{67}
x_{16}	x_{17}	x_{36}	x_{37}	x_{56}	x_{57}	x_{76}	x_{77}

→

x_{00}	x_{10}	x_{20}	x_{30}	x_{40}	x_{50}	x_{60}	x_{70}
x_{01}	x_{11}	x_{21}	x_{31}	x_{41}	x_{51}	x_{61}	x_{71}
x_{02}	x_{12}	x_{22}	x_{32}	x_{42}	x_{52}	x_{62}	x_{72}
x_{03}	x_{13}	x_{23}	x_{33}	x_{43}	x_{53}	x_{63}	x_{73}
x_{04}	x_{14}	x_{24}	x_{34}	x_{44}	x_{54}	x_{64}	x_{74}
x_{05}	x_{15}	x_{25}	x_{35}	x_{45}	x_{55}	x_{65}	x_{75}
x_{06}	x_{16}	x_{26}	x_{36}	x_{46}	x_{56}	x_{66}	x_{76}
x_{07}	x_{17}	x_{27}	x_{37}	x_{47}	x_{57}	x_{67}	x_{77}

Three stages in the transposition of an 8×8 matrix.

2D DFT memory problems



(a) (b)

In-place storage of the 2D DFT of a real-valued signal. (a) Storage after row transform. (b) Storage after column transform. The cross-hatched areas denote the storage places for the imaginary part of the transform.

2D Power Spectrum estimation

a) Image LENNA; b) periodogram of LENNA.

Discrete Cosine Transform

VML

- DCT is used in the JPEG and MPEG standards.
- Forward DCT transform:

$$C(0) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n)$$

$$C(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos \frac{(2n+1)k\pi}{2N}.$$

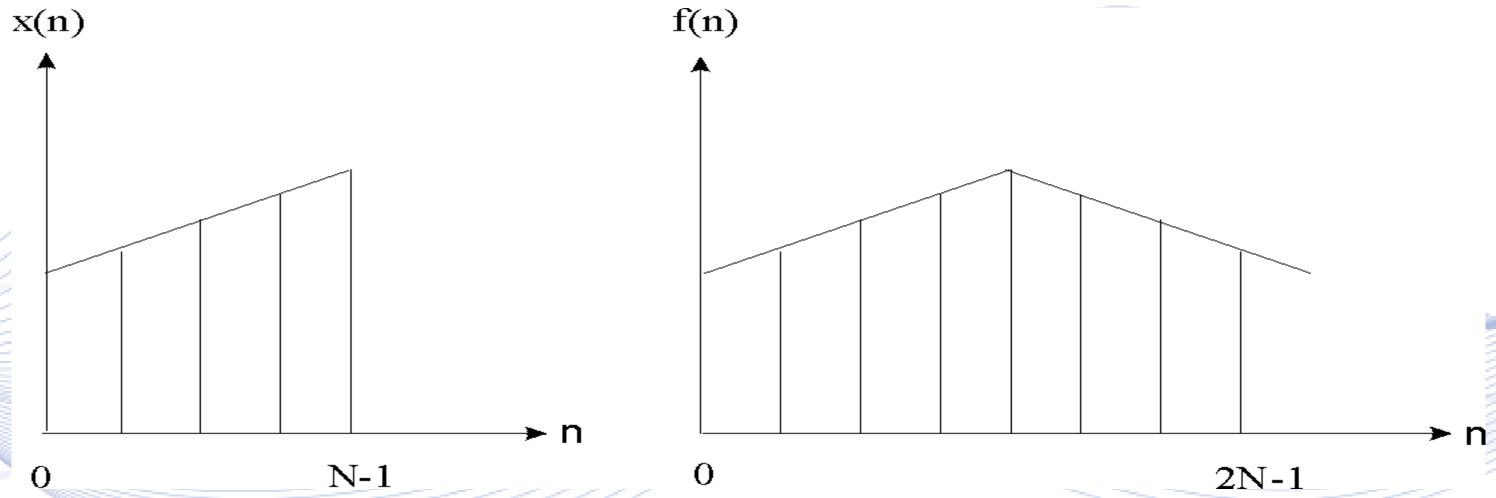
- Inverse DCT:

$$x(n) = \frac{1}{\sqrt{N}} C(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} C(k) \cos \frac{(2n+1)k\pi}{2N}.$$

Discrete Cosine Transform

- Let $x(n)$ be a signal. An ***even sequence*** $f(n)$ can be produced by this signal as:

$$f(n) = \begin{cases} x(n) & 0 \leq n \leq N-1 \\ x(2N-1-n) & N \leq n \leq 2N-1. \end{cases}$$



Creation of an even sequence.

2D Discrete Cosine Transform

2D $N_1 \times N_2$ DCT is defined as:

$$C(k_1,k_2)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} 4x(n_1,n_2)\cos\frac{(2n_1+1)k_1\pi}{2N_1}\cos\frac{(2n_2+1)k_2\pi}{2N_2},$$

for $0 \le k_1 \le N_1 - 1, 0 \le k_2 \le N_2 - 1.$

$$x(n_1,n_2)=\frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} w_1(k_1)w_2(k_2)C(k_1,k_2)\cos\frac{(2n_1+1)k_1\pi}{2N_1}\cos\frac{(2n_2+1)k_2\pi}{2N_2},$$

where:

$$w_1(k_1)=\begin{cases}1/2 & k_1=0\\ 1 & 1\le k_1\le N_1-1\end{cases} \quad w_2(k_2)=\begin{cases}1/2 & k_2=0\\ 1 & 1\le k_2\le N_2-1\end{cases}.$$

2D Discrete Cosine Transform

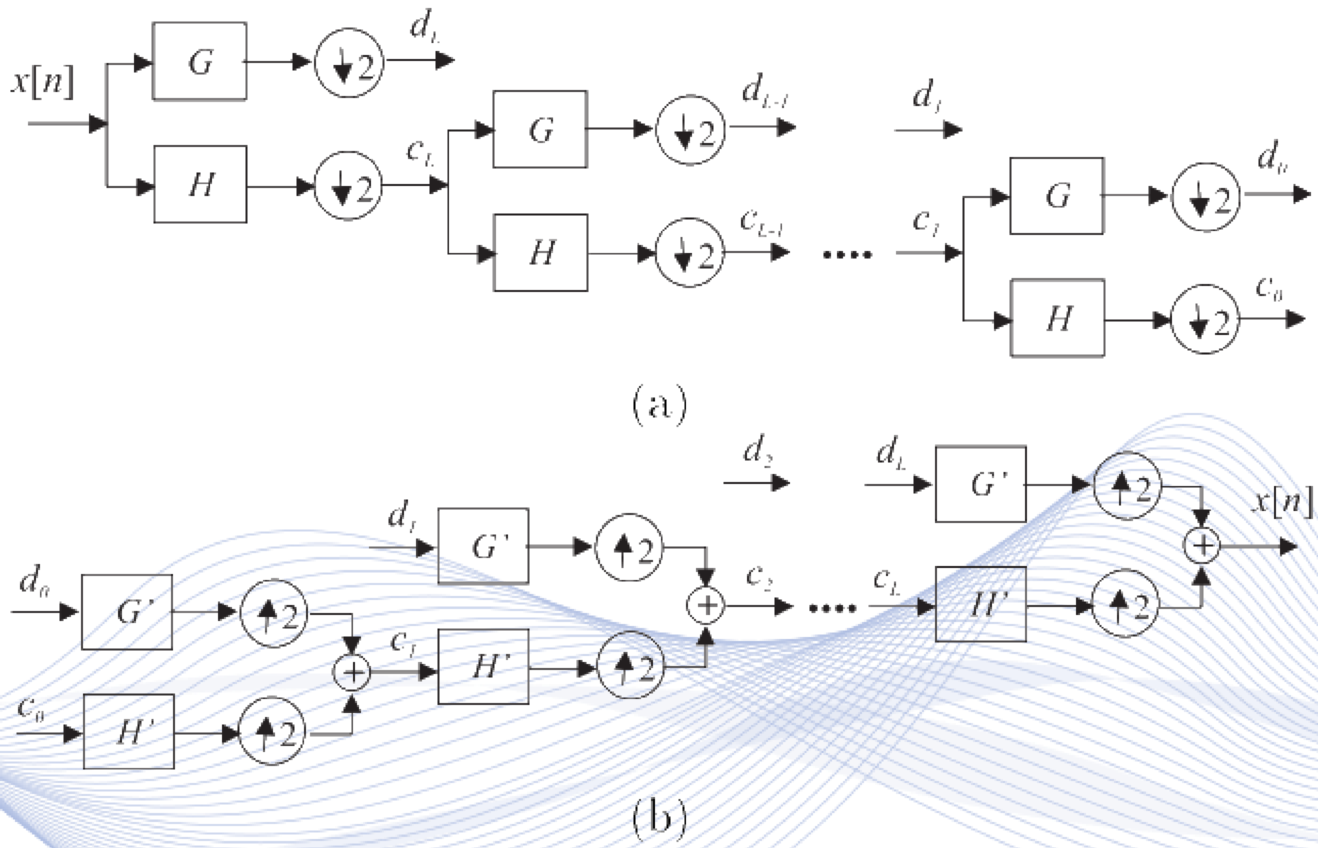
a) Image LENNA; b) Energy concentration in low DCT frequencies.

Discrete Wavelet Transform

Discrete Wavelet Transform (***DWT***) is a fast, linear, invertible, orthogonal image transform.

- It breaks down the signal (image) into a ***wavelet*** representation (sub-images) that represent it as a function of space and scale.
- It analyses non-stationary images well, because of its parallel localization ability in both space and scale.

Discrete Wavelet Transform



a) Multi-level wavelet decomposition; b) Reconstruction of a 1-dimensional signal.

Discrete Wavelet Transform

Wavelet image transform [2DWAV] .

Artificial Intelligence &
Information Analysis Lab

Discrete Wavelet Transform

- DWT can be used in image compression for loseless or lossy compression.
- DWT offers good spatial and frequency resolution, which is used for producing good quality for an image.
- JPEG2000 is a DWT based compression standard.

Bibliography

[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[PIT2017] I. Pitas, “Digital video processing and analysis” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television” , Createspace/Amazon, 2013.
[NIK2000] N. Nikolaidis and I. Pitas, “3D Image Processing Algorithms”, J. Wiley, 2000.
[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr