Lecture notes for SSY150: Multimedia and video communications

# 2D Still Image Compression

(for lecture 4)

**Irene Y.H. Gu, Dept. of Signals and Systems,**
**Chalmers Univ. of Technology, Sweden**
**March 24, 26, 2009**

## Contents

1. First and second generation image compression techniques: differences
2. 1D DCT and IDCT transforms: revisit
3. 2D DCT and 2D IDCT: basics
4. Block-based 2D DCT transforms for image compression
5. Subband filters and wavelets:
   used for the first and 2nd generation compression
6. Tasks in laboratory-exercise 2

## 1. 2D Still Image Compression Techniques

**<u>1st generation</u> techniques:**

(a) compression is obtained in a transform domain by removing small value coefficients

(b) compression is done by using subband filters and variable bit allocations

**<u>2nd generation</u> techniques:**

Compression is achieved by exploiting visually important image properties in transform/subband filtering domains:

e.g. edges, textures (lowpass contents)

**example: wavelet transforms + using multi-scale edges**

## Select the type of transforms/subband filters

**Principles of selecting a "good" (suitable) transform**:

able to yield high energy compaction in the transform domain

e.g. Karhunen-Loeve transform (KLT)

using eigenvectors, eigenvalues of data autocorrelation matrix,

best energy compaction,

however, data dependency and high computations

DCT transform

is the 2nd best energy compaction if the data is stationary Markov sequence, the transform is independent of data

**Similar principles for choosing 'good' subband filters:**

orthogonal, energy compaction

# 2. 1D DCT Transforms (revisit)

**Forward DCT:**

$$\mathrm{f}(k) = \frac{w_k}{\sqrt{N}} \sum_{n=0}^{N-1} s(n) \cos\frac{(2n+1)\pi k}{2N}, \quad k = 0, \cdots N-1, \; w_k = \begin{cases} 1 & k = 0 \\ \sqrt{2} & k > 0 \end{cases}$$

Define a transform matrix $\mathbf{A} = [c_k(n)]$

$$c_k(\mathrm{n}) = \begin{cases} \frac{1}{\sqrt{N}} & k = 0 \\ \sqrt{\frac{2}{N}} \cos\frac{(2n+1)\pi k}{2N} & 0 < k \le N-1 \end{cases}$$

→ DCT (in the vector and matrix form) **f**=**As**

**Inverse DCT**: $\mathbf{s} = \mathbf{A}^{-1}\mathbf{f} = \mathbf{A}^T\mathbf{f}$

↑

(Since DCT is real and orthonomal $\Rightarrow \mathbf{A}^T = \mathbf{A}^{-1}, \mathbf{A} = \mathbf{A}^*$)

# 3. 2D DCT Transforms

**2D DCT transform:**

$$\mathrm{F}(k_1, k_2) = \frac{w_{k_1} w_{k_2}}{\sqrt{N_1 N_2}} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} S(n_1, n_2) \cos\frac{(2n_1+1)\pi k_1}{2N_1} \cos\frac{(2n_2+1)\pi k_2}{2N_2} \qquad (1)$$

where: $k_1 = 0, \cdots N_1 - 1, \; k_2 = 0, \cdots N_2 - 1, \quad w_{k_i} = \begin{cases} 1 & k_i = 0 \\ \sqrt{2} & k_i > 0 \end{cases}$

(1) is equivalent to: $F(k_1, k_2) = \sum_{n_1} \sum_{n_2} c_{k_1}(n_1) S(n_1, n_2) c_{k_2}(n_2)$

$$\begin{cases} \text{2D forward DCT:} & \mathbf{F} = \mathbf{A}\mathbf{S}\mathbf{A}^T \\ \text{2D Inverse DCT:} & \mathbf{S} = \mathbf{A}^T\mathbf{F}\mathbf{A} \end{cases}$$

where: $\mathbf{A}$ is an orthonormal matrix, $\mathbf{A}\mathbf{A}^T = \mathbf{A}^T\mathbf{A} = \mathbf{I}$

# 1D representation of 2D transforms

**2D forward and inverse transforms:**

$$\begin{cases} \mathbf{F} = \mathbf{A}\mathbf{S}\mathbf{A}^T \\ \mathbf{S} = \mathbf{A}^{*T}\mathbf{F}\mathbf{A}^* \end{cases}$$

$\Rightarrow$

column-scanned vector form

**corresponding 1D representation:**

$$\begin{cases} \mathbf{f} = (\mathbf{A} \otimes \mathbf{A})\mathbf{s} \\ \mathbf{s} = (\mathbf{A} \otimes \mathbf{A})^{*T}\mathbf{f} \end{cases}$$

If size of $\mathbf{A}$ : $N \times N \quad \Rightarrow$ size of $(\mathbf{A} \otimes \mathbf{A})$ : $N^2 \times N^2$

Kronecker product: $\mathbf{A} \otimes \mathbf{B} = \left[a_{m,n}\mathbf{B}\right] = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1N}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{N1}\mathbf{B} & \cdots & a_{NN}\mathbf{B} \end{bmatrix}$

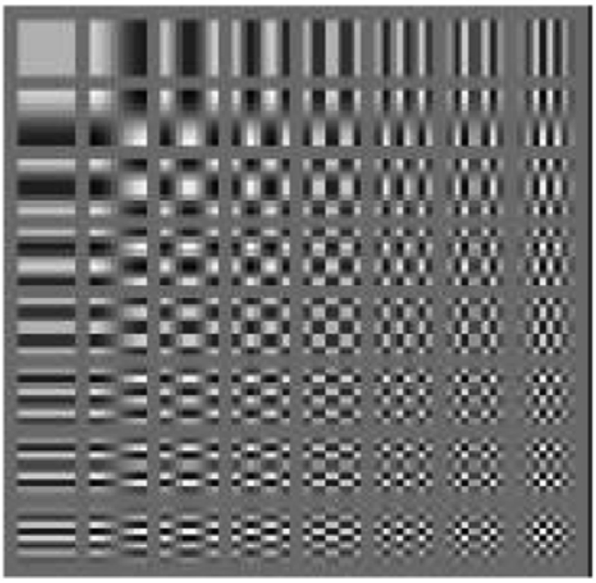
# Why select DCT transforms?

- **Energy compaction for highly correlated data:**
  The basis functions of 1D DCT = eigenvectors of the symmetric tri-diagonal matrix $R_A$ , which is close to the KLT of a 1st order stationary Markov sequence of length $N$

$$R_A = \begin{bmatrix} 1-\alpha & -\alpha & 0 & 0 & 0 \\ -\alpha & 1 & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & 1 & -\alpha \\ 0 & 0 & 0 & -\alpha & 1-\alpha \end{bmatrix}$$

- **2nd best as compared to the KL (Karhunen-Loeve) transform**

# 64 Basis images of a 8x8 DCT



# 4. Block-based 2D DCT transforms

**Why block-based image transformation?**

Most images are nonstationary!
however, approximate stationary within each small sized block

**Drawbacks of block-based transforms**

Block artifacts

**Possible way to overcome this problem**

- Overlapping blocks
- Subband filters

# Block-based transforms (cont'd)

**Divide image in blocks, apply the 2D transform to each image block**
=> *spatial-frequency* representation of 2D images

( as compared to time-frequency representation of 1D signals)
overlap blocks may be used as well. (similar to 1D signals)

**Example: block DCT transform of image**

- Divide image into 16 x 16 blocks
- Apply the DCT to each block of image
- Insert the transform coefficients into the corresponding block.

# Block-based transforms (cont'd)

## To obtain sub-images in the spatial-frequency domain:

For each DCT transformed block, extract one DCT coefficient from the corresponding position and insert it into a sub-image.

<u>e.g.</u>
*block size = 2x2*, image size of NxN
=> *4 subband filtered images* (size N/2 x N/2)

$$\begin{array}{|cccc|} \hline X_{11} & \Delta_{11} & X_{12} & \Delta_{12} \\ \nabla_{21} & \Box_{11} & \nabla_{22} & \Box_{12} \\ X_{21} & \Delta_{21} & X_{22} & \Delta_{22} \\ \nabla_{21} & \Box_{21} & \nabla_{22} & \Box_{22} \\ \hline \end{array} \Rightarrow \begin{array}{|cccc|} \hline X_{11} & X_{12} & \Delta_{11} & \Delta_{12} \\ X_{21} & X_{22} & \Delta_{21} & \Delta_{22} \\ \nabla_{11} & \nabla_{12} & \Box_{11} & \Box_{12} \\ \nabla_{21} & \nabla_{22} & \Box_{21} & \Box_{22} \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline F_{00} & F_{01} \\ \hline F_{10} & F_{11} \\ \hline \end{array}$$

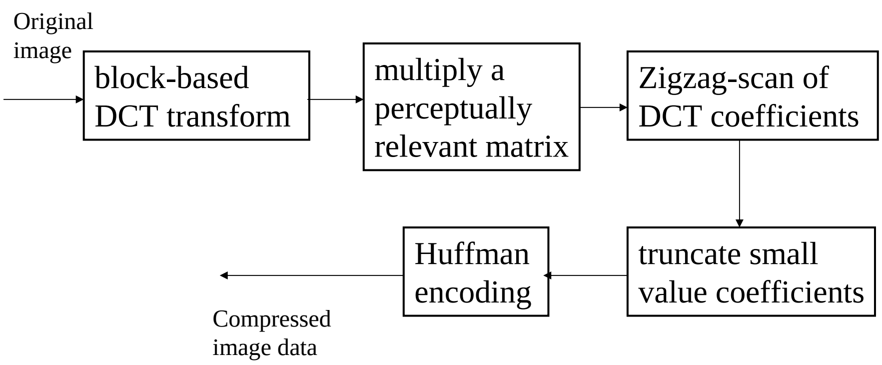
# Example: 2D image compression using block-based DCTs

Apply: DCTs, with non-overlapping 8x8 blocks

Original image

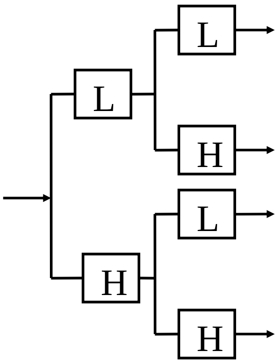
Reconstructed image
(retain 4.2% coefficients)

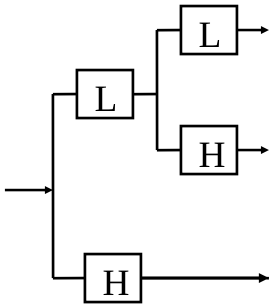
# JPEG still image compression standard



# 5. Subband filters and wavelets

1D subband filters:



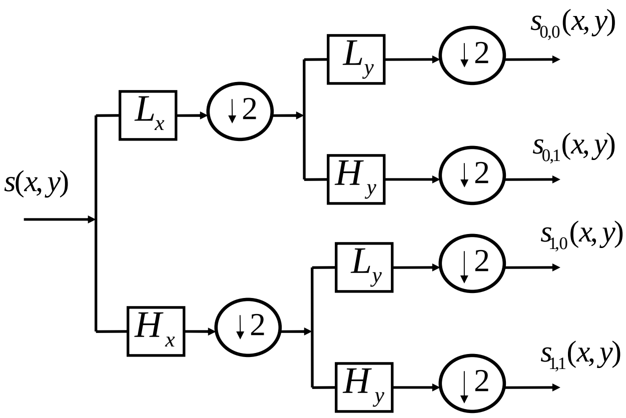




subband filters with equal bandwidth
(balanced tree-structure)

dyadic subband filters with
Octave bandwidths
(unbalanced tree-structure)

# 2D subband filters

For the 2D case, e.g., containing 2 bands in each direction:



# Wavelets

Wavelets: associated with a special type of subband filters $H(\omega)$,
satisfying certain conditions for the smoothing function $\phi(t)$
and wavelet function $\psi(t)$

- Decay of wavelet coefficients
- Smoothness (regularity) : the Number of zeros at $\pi$ is crucial
- Finite support

Special considerations for image processing:
e.g.: linear phase ( => choose bi-orthogonal wavelets)
orthogonal

# Subband image compression

## 1st generation methods:

- Assign low bit rates to high frequency bands (small coefficients)
- Use different bandwidths
  (different sensitivity in the Human Visual System)

## 2nd generation methods:

Encode perceptually important features, such as *edges*, *textures*
e.g. Wavelet-based compression:
use multi-scale edge curves in the sub-images
+ use textures in the lowest resolution sub-image

# Example:  wavelet for image decomposition

**Original image**

1 scale level

2 scale levels

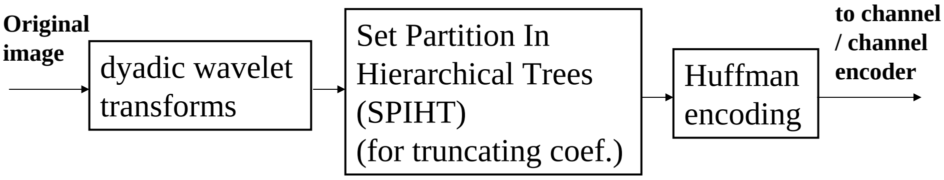
# Example: Wavelet for image compression

Select: bi-orthogonal spline wavelets

Original Image

Compressed image

# Wavelet-based 2D still image codec



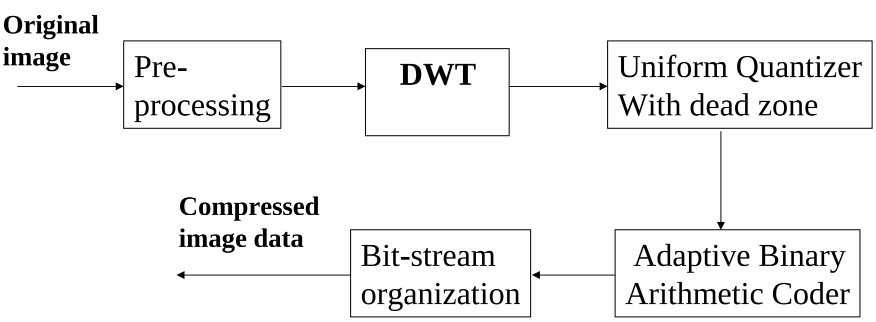

Advantages:

- suitable for progressive transmission of layered images
- high compression rate
- less block artifacts

JPEG 2000 standard uses discrete wavelet transform (instead of DCT)!

# Still image coding using JPEG 2000

(approved in 2002)

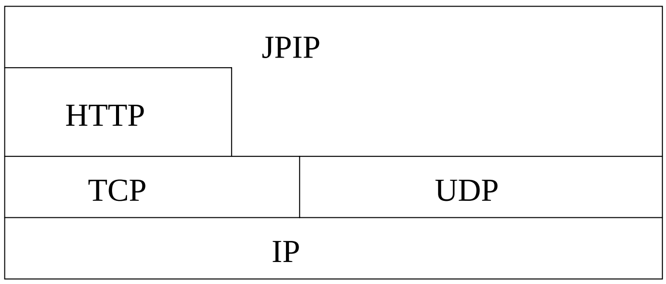



JPEG 2000 Building Blocks

# JPEG 2000 Part 9 - JPIP protocol

JPIP is a client-server protocol for JPEG 2000.

In JPIP (JPEG 2000 Interactive protocol), the client uses a view window to define image resolution, size, location, components, layers, etc.




JPIP protocol stack

# How to use the wavelet toolbox in Matlab?

- wavemenu (GUI with a pull down menu)
- wavedemo (demo examples)

# Laboratory exercise 2

- Task-1: 2D DCT for image compression
- Task-2: Block-based 2D DCT for image compression
- Task 3: 2D subband filters for image compression

# Tasks in Lab. 2

**1. Task-1: 2D DCT for an entire image compression**

( → *frequency domain* image representation)

* Learn the relation between a 1D transform and the corresponding 2D transform, assuming the (transform) kernel is separable:

$$h(n,m)=h(n)h(m)$$

* Learn how to achieve compression by setting small value DCT coefficients to zeros

**2. Task-2: Block-based 2D DCT for image compression**

(→ *spatial-frequency domain* image representation)

* Divide a nonstationary image into small (approx. stationary) blocks,
* Apply the DCT and compression to each image block.
* Apply compression to each block of DCT coefficients

# Tasks in Lab. 2

**3. Task-3: 2D wavelet (subband) filters for image compression**

- Use the GUI in Matlab wavelet toolbox ‘wavemenu’
  to perform this task
- Set the same compression ratio as that in the Task-2.

**4. Compare the performance of these two compression methods.**

• Subjective criterion: visual observation
• objective criterion: PSNR