Lecture-3 notes, SSY150: Multimedia and video communications

2D Image Transforms, Subband Filters, and Compression

Irene Gu

Chalmers University of Technology, Sweden

April 2, 2020

1

Contents

2

1. 2D Still Image Compression Techniques

1st generation 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

2nd generation techniques

Compression is achieved by exploiting visually important image properties, e.g. edges, textures (lowpass contents), in a transform/subband filtering domain

Example: in wavelet transform domain, using multi-scale edges

3

How to select: a 'best' transform / subband filter

Principles of selecting a "best" (suitable) transformation:

one that results in high energy compaction in the transform domain

e.g. Karhunen-Loeve transform (KLT)

use eigenvectors, eigenvalues of data autocorrelation matrix,

- best energy compaction,
- however, data dependency and high computations

DCT transform

- 2nd best transform in terms of energy compaction if data is stationary Markov sequence,
- transform matrix is independent of data

Similar principles for selecting a 'best' subband filter

orthogonal, energy compaction

4

2. 1D DCT Transform: Revisit

Forward DCT:

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

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

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

→ **Forward DCT** (in the vector and matrix form)

$$\mathbf{f=As}$$

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

↑

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

3. 2D DCT Transforms

2D DCT transform:

$$F(k_1,k_2)=\frac{w_{k_1}w_{k_2}}{\sqrt{N_1N_2}}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}S(n_1,n_2)\cos\frac{(2n_1+1)\pi k_1}{2N_1}\cos\frac{(2n_2+1)\pi k_2}{2N_2}\qquad(1)$$

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

2D forward DCT: $\mathbf{F}=\mathbf{ASA}^T$

2D Inverse DCT: $\mathbf{S}=\mathbf{A}^T\mathbf{FA}$

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

2D transforms: with 1D representation

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:

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

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

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

7

Why choose DCT transform?

- **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

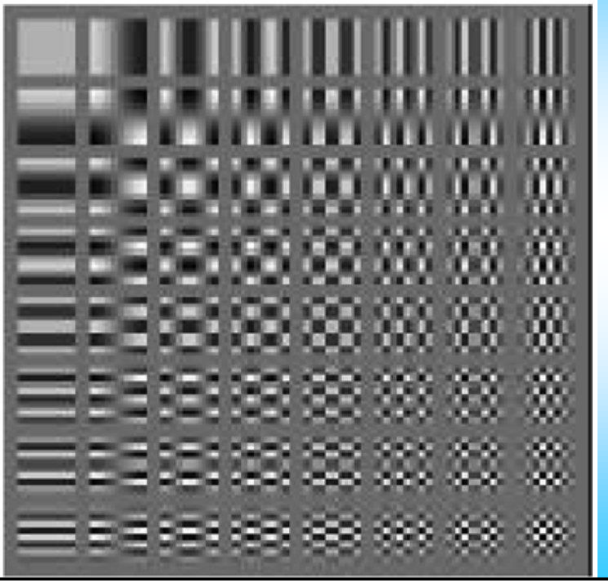
$$R_A = \begin{bmatrix} 1-\alpha & -\alpha & 0 & 0 & 0 \\ -\alpha & 1 & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & 1 & -\alpha \\ 0 & 0 & 0 & -\alpha & 1-\alpha \end{bmatrix}$$

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

8

64 Basis images of a 8x8 DCT

If a 2D transform kernel is separable, a basis image = outer product of 2 basis vectors!



9

4. Block-based 2D DCT transform

Why block-based image transform?

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

Drawback of block-based transform

Block artifacts

Possible way to overcome this problem

- Use blocks with overlaps
- Use subband filters

10

Block-based transform (cont'd)

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

Compare to 1D SP concept: time-frequency representation of 1D signals

Overlapped blocks may be used (similar to 1D signal processing case)

Example: block DCT transform of image

- Divide image into blocks (size M x M pixels)
- Apply the DCT to each block of image
- Insert DCT transform coefficients into a corresponding block.

Block-based transform (cont'd)

Equivalent to generating sub-images in spatial-frequency domain:

For each DCT transformed block, extract DCT coefficients
from the corresponding positions, and insert them into sub-images.

<u>e.g.</u>
block size = 4x4
can be converted to: 4 subband filtered images (each of size 2 x 2)

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

Example: 2D image compression using block-based DCT

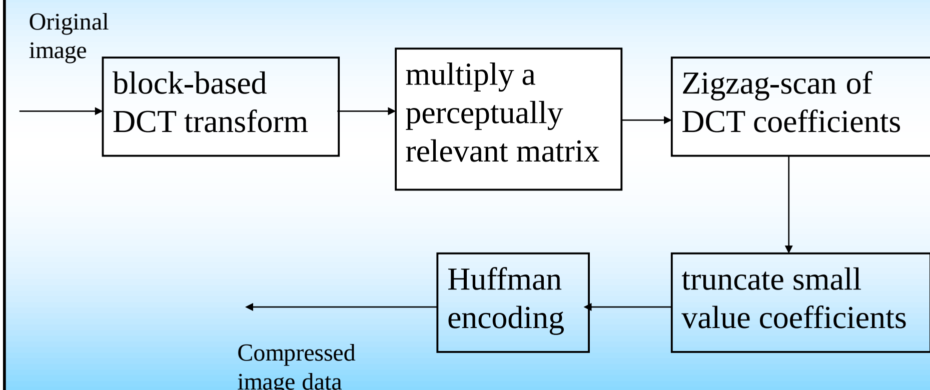
Apply DCT, with non-overlapping 8x8 blocks

Original image

Reconstructed image
(retain 4.2% coefficients)

13

Still image compression standard: JPEG

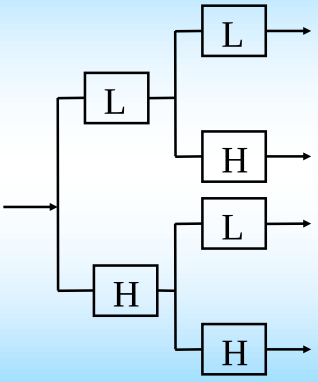


JPEG pipeline

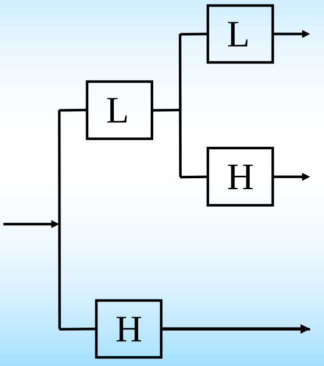
14

5. Subband filters and wavelets

1D subband filters



with equal bandwidth
(balanced tree-structure)

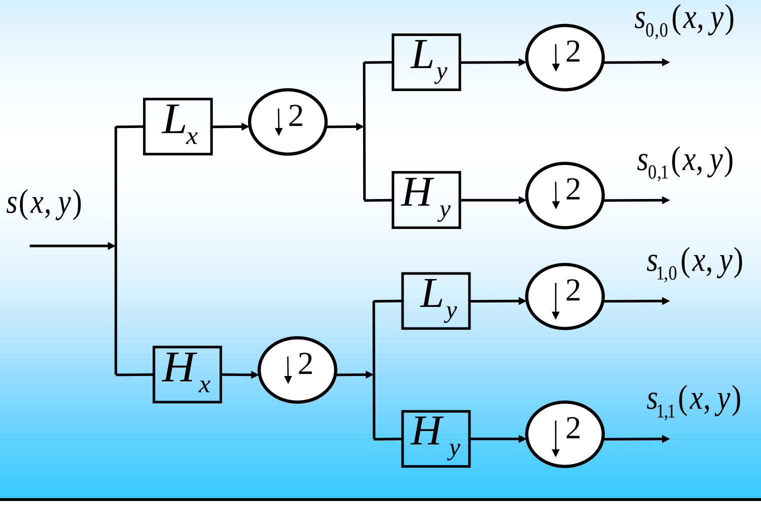


with octave bandwidths
(unbalanced tree-structure)

15

2D subband filters

Corresponding case to a 2-band 1D subband filter,
a 2D subband filter has 4 bands (2 bands (L/H) in each direction) !



16

Wavelets

Wavelets: a special type of subband filters $H(\omega)$ fulfil certain special conditions, consisting of:
- a smoothing function $\phi(t)$
- a wavelet function $\psi(t)$

Satisfying certain conditions
- Decay of wavelet coefficients
- Smoothness (regularity) : the number of zeros at π is crucial
- Finite support

Consideration when choosing wavelet kernels for image processing
e.g.: linear phase,
orthogonal
=> choose bi-orthogonal wavelets

17

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

18

Example: Wavelet for image decomposition

Original image

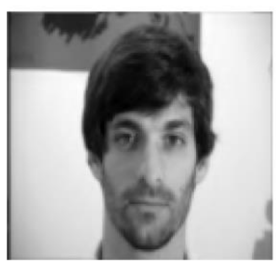
1 scale level

2 scale levels

Example: Wavelet for image compression

Select: bi-orthogonal spline wavelets

Original image

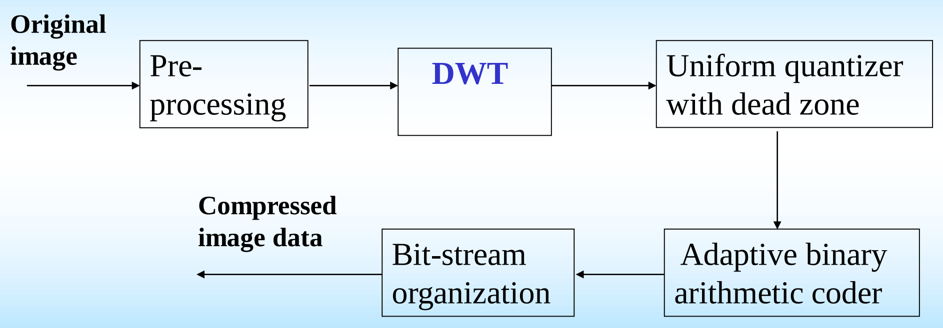


Compressed image

Use Wavelet Toolbox in Matlab

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

Still image compression standard: JPEG 2000



JPEG 2000 pipeline

21

6. Objective Image Quality Measures

PSNR (Peak signal-to-noise ratio):

$$PSNR = 10\ \log_{10} \frac{(Max_{\mathrm{I}})^2}{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I(i,j) - I_{ideal}(i,j)\right)^2} = 10\ \log_{10}\frac{(Max_{\mathrm{I}})^2}{MSE} \quad (1)$$

where $Max_{\mathrm{I}} = 2^B - 1$ (for $B = 8$ bits, $Max_{\mathrm{I}} = 255$)

SSIM (Structural Similarity):

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)\,(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2)$$

22

How SSIM Equation (2) is derived ?

Define: overall similarity measure between 2 images x and y:

$$S(\mathbf{x},\mathbf{y}) = f\left(l(\mathbf{x},\mathbf{y}), c(\mathbf{x},\mathbf{y}), s(\mathbf{x},\mathbf{y})\right)$$

For comparing the luminance:

$$l(\mathbf{x},\mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

C_1 is included to avoid instability when $\mu_x^2 + \mu_y^2$ is very close to zero

$$C_1 = (K_1 L)^2 \qquad K_1 \ll 1$$

For comparing the contrast:

$$c(\mathbf{x},\mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

where: $C_2 = (K_2 L)^2$, and $K_2 \ll 1$

For comparing the structure:

$$s(\mathbf{x},\mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

Noting:
μ_x and σ_x^2 are computed in a small 2D window, where pixels within the window are weighted by the Gaussian shape

23

How SSIM Equation (2) is derived ? (cont'd)

- set the structural similarity measure as:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = [l(\mathbf{x},\mathbf{y})]^{\alpha} \cdot [c(\mathbf{x},\mathbf{y})]^{\beta} \cdot [s(\mathbf{x},\mathbf{y})]^{\gamma}$$

where $\alpha > 0, \beta > 0$ and $\gamma > 0$, used to adjust the relative importance of the three components.

- set $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$ =>

$$\boxed{\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}}$$

Algorithm:

- For each pixel, compute $SSIM(\mathbf{X}_{i,j}, \mathbf{Y}_{i,j})$ using a small 2D window (e.g. 8 x 8)
- Compute the **Mean SSIM** by averaging SSIM over all pixels in the images **X,Y**

$$MSSIM(\mathbf{X},\mathbf{Y}) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} SSIM(\mathbf{X}_{i,j}, \mathbf{Y}_{i,j})$$

24