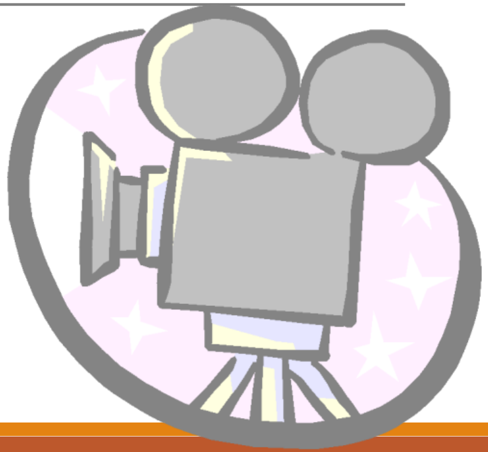
Image Processing

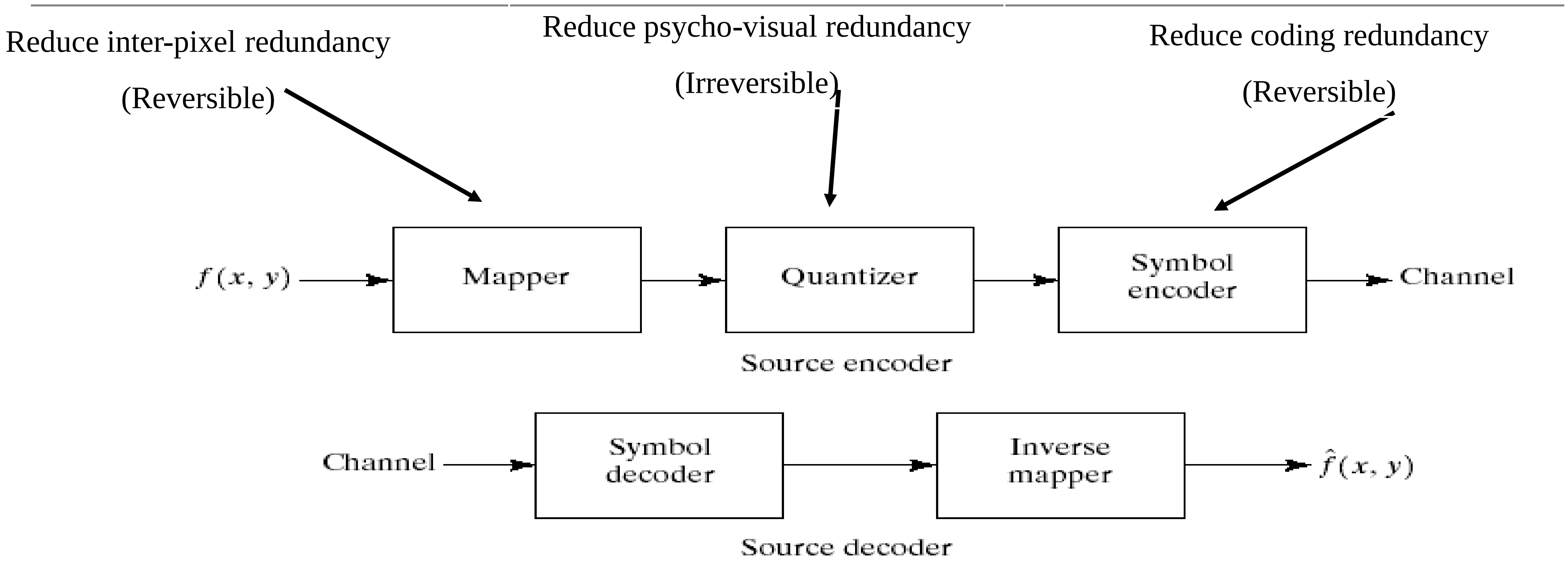
CS-317/CS-341



Outline

- Basic Transformations

Image Compression model



(a) Source encoder and (b) source decoder model.

Basic Transformation

Image Compression model : Transformation

Reduce inter-pixel redundancy

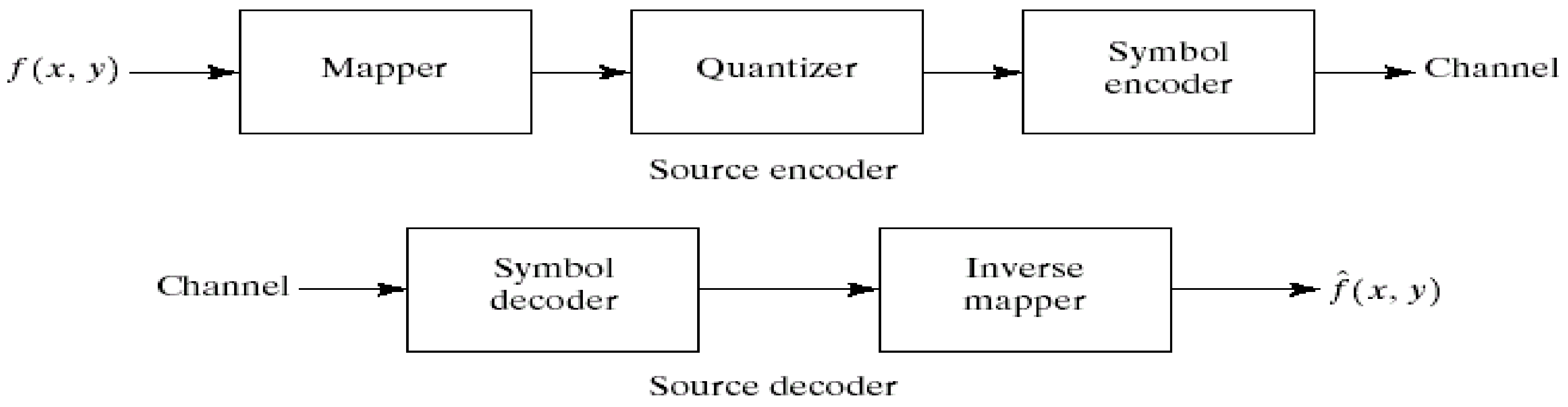
(Reversible)

Reduce psycho-visual redundancy

(Irreversible)

Reduce coding redundancy

(Reversible)

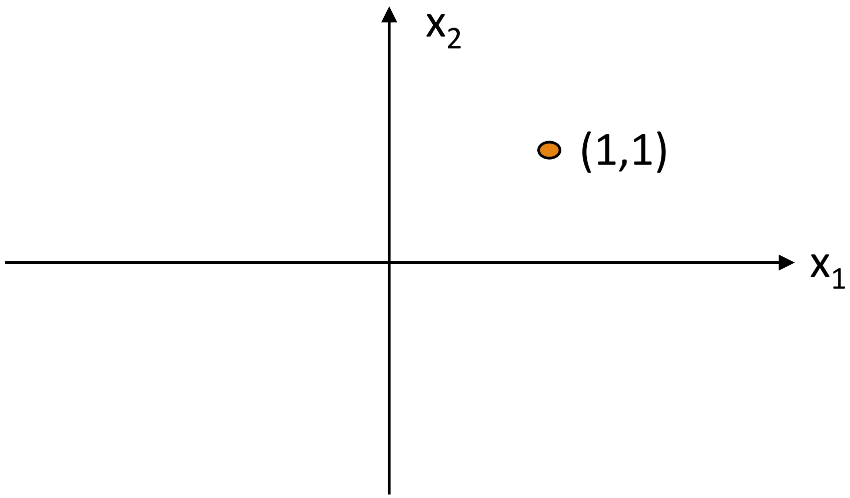


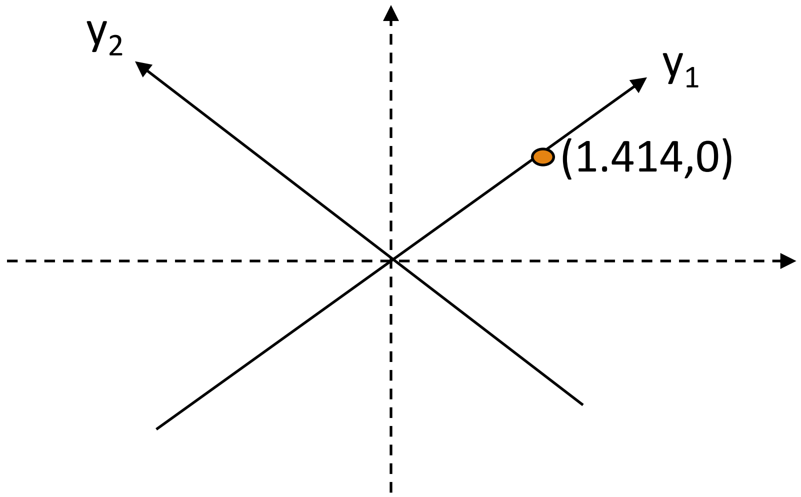
(a) Source encoder and (b) source decoder model.

Lossy Image Compression

- Lossy transform coding
  - Image Transforms (Discrete Cosine Transform)
  - Joint Photographic Expert Group (JPEG)

An Example of 1D Transform with Two Variables





$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \vec{y} = \mathbf{A}\vec{x}, \mathbf{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Transform matrix

Generalization into N Variables

forward transform $$\vec{y}_{N\times 1} = \mathbf{A}_{N\times N}\vec{x}_{N\times 1}$$

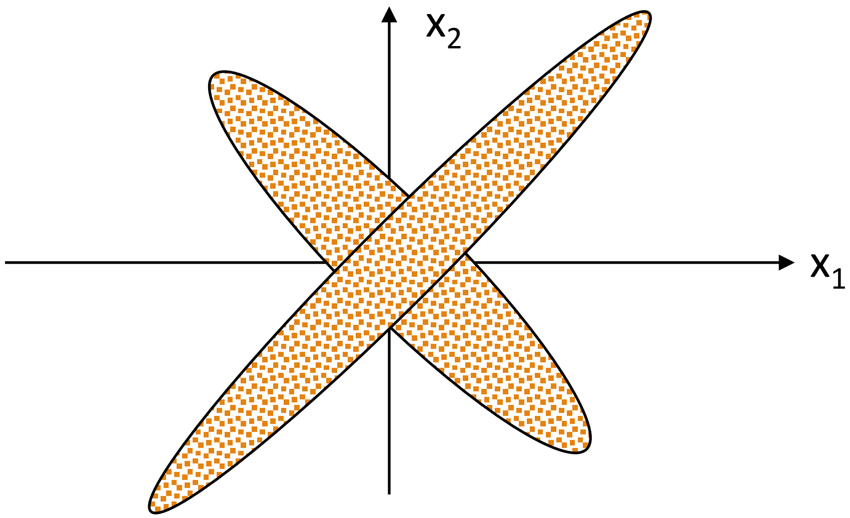
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & \cdots & a_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

$$\Updownarrow$$

$$\vec{y} = \sum_{i=1}^{N} x_i \vec{b}_i, \quad \vec{b}_i = [a_{i1}, \ldots, a_{iN}]^T$$

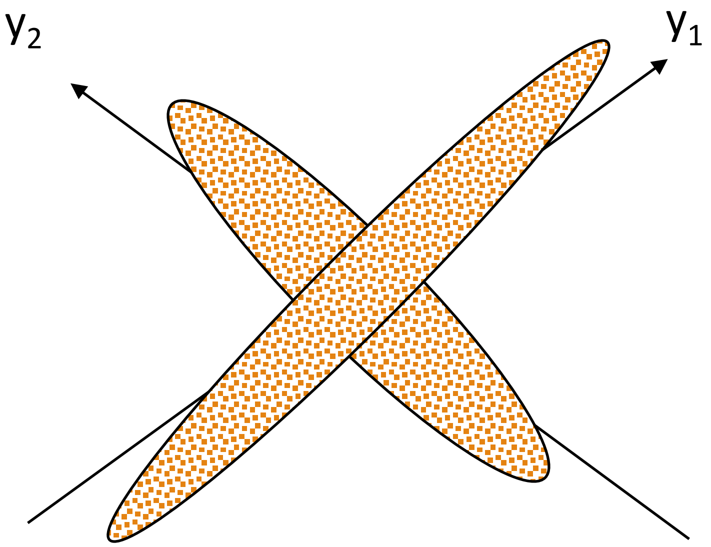
basis vectors (column vectors of transform matrix)

Decorrelating Property of Transform



x_1 and x_2 are highly correlated

$$p(x_1 x_2) \neq p(x_1)p(x_2)$$



y_1 and y_2 are less correlated

$$p(y_1 y_2) \approx p(y_1)p(y_2)$$

Transform=Change of Coordinates

Intuitively speaking, transform plays the role of facilitating the source modeling

- Due to the decorrelating property of transform, it is easier to model transform coefficients (Y) instead of pixel values (X)

An appropriate choice of transform (transform matrix A) depends on the source statistics P(X)

- We will only consider the class of transforms corresponding to unitary matrices

Unitary Matrix and 1D Unitary Transform

Definition

conjugate transpose

A matrix A is called unitary if $A^{-1} = A^{*T}$

When the transform matrix A is unitary, the defined transform $\vec{y} = A\vec{x}$ is called unitary transform

Example

$$\mathbf{A} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \mathbf{A}^{-1} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \mathbf{A}^T$$

For a real matrix **A**, it is unitary if $\mathbf{A}^{-1} = \mathbf{A}^T$

Inverse of Unitary Transform

For a unitary transform, its inverse is defined by

$$\vec{x} = \mathbf{A}^{-1}\vec{y} = \mathbf{A}^{*T}\vec{y}$$

Inverse Transform

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} a_{11}^* & \cdots & \cdots & a_{N1}^* \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{1N}^* & \cdots & \cdots & a_{NN}^* \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$\Updownarrow$

$$\vec{x} = \sum_{i=1}^{N} y_i \vec{b}_i, \quad \vec{b}_i = [a_{1i}^*, \ldots, a_{Ni}^*]^T$$

basis vectors corresponding to inverse transform

Properties of Unitary Transform

Energy compaction: only few transform coefficients have large magnitude

- Such property is related to the decorrelating role of unitary transform

Energy conservation: unitary transform preserves the 2-norm of input vectors

- Such property essentially comes from the fact that rotating coordinates does not affect Euclidean distance

Energy Compaction Example

Hadamard matrix

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, \vec{x} = \begin{bmatrix} 100 \\ 98 \\ 98 \\ 100 \end{bmatrix}$$

$\Downarrow$

$$\vec{y} = \mathbf{A}\vec{x} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 100 \\ 98 \\ 98 \\ 100 \end{bmatrix} = \begin{bmatrix} 198 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

significant

insignificant

Energy Conservation

$\vec{y} = \mathbf{A}\vec{x}$ A is unitary $\Rightarrow \|\vec{y}\|^2 = \|\vec{x}\|^2$

<u>Proof</u>

$$\|\vec{y}\|^2 = \sum_{i=1}^{N} |y_i|^2 = \vec{y}^{*T}\vec{y} = (\mathbf{A}\vec{x})^{*T}(\mathbf{A}\vec{x})$$

$$= \vec{x}^{*T}(\mathbf{A}^{*T}\mathbf{A})\vec{x} = \vec{x}^{*T}\vec{x} = \sum_{i=1}^{N} |x_i|^2 = \|\vec{x}\|^2$$

Numerical Example

$$\mathbf{A} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \vec{x} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$\Downarrow$$

$$\vec{y} = \mathbf{A}\vec{x} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 4 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 7 \\ 1 \end{bmatrix}$$

<u>Check:</u>

$$\| \vec{x} \|^2 = 3^2 + 4^2 = 25, \| \vec{y} \|^2 = \frac{7^2 + 1^2}{2} = 25$$

Implication of Energy Conservation

$$\vec{x} = [x_1,\ldots,x_N]^T \xrightarrow{\boxed{T}} \vec{y} = [y_1,\ldots,y_N]^T \xrightarrow{\boxed{Q}} \hat{\vec{y}} = [\hat{y}_1,\ldots,\hat{y}_N]^T \xrightarrow{\boxed{T^{-1}}} \hat{\vec{x}} = [\hat{x}_1,\ldots,\hat{x}_N]^T$$

$$\vec{y} = \mathbf{A}\vec{x} \qquad \hat{\vec{y}} = \mathbf{A}\hat{\vec{x}}$$

$$\vec{y} - \hat{\vec{y}} = \mathbf{A}(\vec{x} - \hat{\vec{x}})$$

A is unitary

$$\| \vec{y} - \hat{\vec{y}} \|^2 = \| \vec{x} - \hat{\vec{x}} \|^2$$

Summary of 1D Unitary Transform

Unitary matrix: $\mathbf{A}^{-1}=\mathbf{A}^{*T}$

Unitary transform: $\vec{y} = \mathbf{A}\vec{x}$ $\mathbf{A}$ unitary

Properties of 1D unitary transform

- Energy compaction: most of transform coefficients y_i are small
- Energy conservation: quantization can be directly performed to transform coefficients

$$\|\vec{y}\|^2 = \|\vec{x}\|^2 \Rightarrow \|\vec{y}-\hat{\vec{y}}\|^2 = \|\vec{x}-\hat{\vec{x}}\|^2$$

From 1D to 2D

Do N 1D transforms in parallel

$$\mathbf{Y}_{N\times N} = \mathbf{A}_{N\times N}\mathbf{X}_{N\times N}$$

$$\begin{bmatrix} y_{11} & \cdots & \cdots & y_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ y_{N1} & \cdots & \cdots & y_{NN} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & \cdots & a_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_{11} & \cdots & \cdots & x_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{N1} & \cdots & \cdots & x_{NN} \end{bmatrix}$$

$$[\vec{y}_1 \,|\, \dots \,|\, \vec{y}_i \,|\, \dots \,|\, \vec{y}_N] \qquad [\vec{x}_1 \,|\, \dots \,|\, \vec{x}_i \,|\, \dots \,|\, \vec{x}_N]$$

$$\vec{y}_i = \mathbf{A}\vec{x}_i, i = 1,2,\dots,N$$

$$\vec{x}_i = [x_{i1},\dots,x_{iN}]^T, \vec{y}_i = [y_{i1},\dots,y_{iN}]^T$$

Definition of 2D Transform

2D forward transform $$\mathbf{Y}_{N\times N} = \mathbf{A}_{N\times N}\mathbf{X}_{N\times N}\mathbf{A}^T_{N\times N}$$

$$\begin{bmatrix} y_{11} & \cdots & \cdots & y_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ y_{N1} & \cdots & \cdots & y_{NN} \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & \cdots & \cdots & a_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix}}_{\text{1D column transform}} \begin{bmatrix} x_{11} & \cdots & \cdots & x_{1N} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{N1} & \cdots & \cdots & x_{NN} \end{bmatrix} \underbrace{\begin{bmatrix} a_{11} & \cdots & \cdots & a_{N1} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{1N} & \cdots & \cdots & a_{NN} \end{bmatrix}}_{\text{1D row transform}}$$

2D Transform (Two Sequential 1D Transforms)

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T$$

$$\Updownarrow$$

column transform $\quad \mathbf{Y_1} = \mathbf{A}\mathbf{X}$ (left matrix multiplication first)

row transform $\quad \mathbf{Y} = \mathbf{Y}_1\mathbf{A}^T = (\mathbf{A}\mathbf{Y}_1^T)^T$

$$\Updownarrow$$

row transform $\quad \mathbf{Y}_2 = \mathbf{X}\mathbf{A}^T = (\mathbf{A}\mathbf{X}^T)^T$ (right matrix multiplication first)

column transform $\quad \mathbf{Y} = \mathbf{A}\mathbf{Y}_2$

<u>Conclusion:</u>

- 2D separable transform can be decomposed into two sequential
- The ordering of 1D transforms does not matter

From Basis Vectors to Basis Images

1D transform matrix $\mathbf{A}$ consists of basis vectors (column vectors)

$$\vec{y} = \sum_{i=1}^{N} x_i \vec{b}_i , \vec{b}_i = [a_{i1}, \ldots, a_{iN}]^T$$

2D transform corresponds to a collection of N-by-N basis images

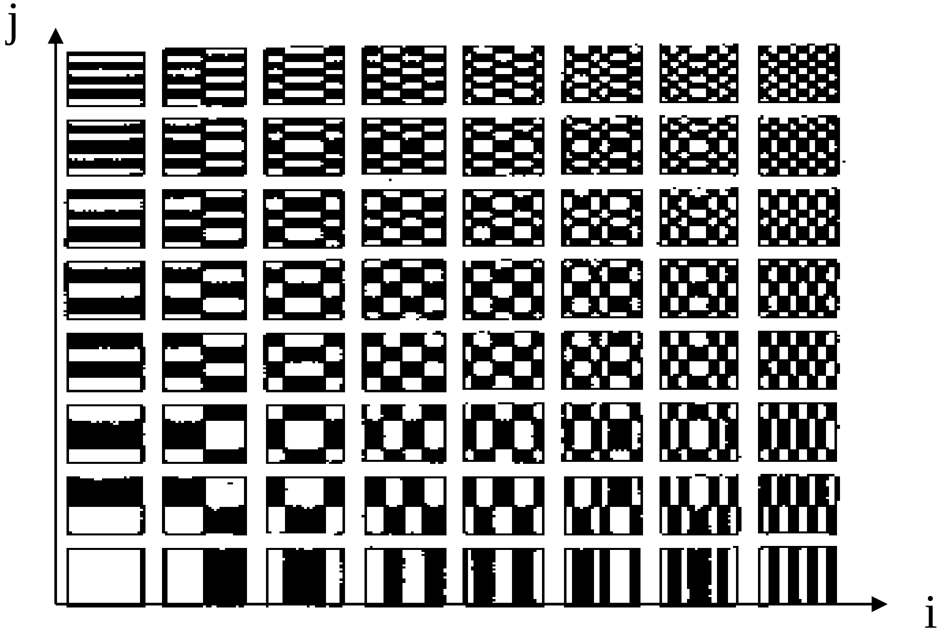
$$\mathbf{Y} = \sum_{i=1}^{N} \sum_{j=1}^{N} x_{ij} \mathbf{B}_{ij} , \mathbf{B}_{ij} = \vec{b}_i \vec{b}_j^{\,T} , \vec{b}_i = [a_{i1}, \ldots, a_{iN}]^T$$

basis image

Example of Basis Images

Hadamard matrix: $$\mathbf{A}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \mathbf{A}_{2n} = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{A}_n & \mathbf{A}_n \\ \mathbf{A}_n & -\mathbf{A}_n \end{bmatrix}$$

$$\mathbf{A} = \frac{1}{\sqrt{8}}\begin{array}{c} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \\ \begin{array}{cccccccc} \uparrow & & & & & & & \uparrow \\ \vec{b}_1 & & & & & & & \vec{b}_8 \end{array} \end{array},$$



B_{ij}

2D Unitary Transform

Suppose A is a unitary matrix,

forward transform $$\mathbf{Y}_{N\times N} = \mathbf{A}_{N\times N}\mathbf{X}_{N\times N}\mathbf{A}_{N\times N}^{T}$$

inverse transform $$\mathbf{X}_{N\times N} = \mathbf{A}_{N\times N}^{*T}\mathbf{Y}_{N\times N}\mathbf{A}_{N\times N}^{*}$$

Proof

Since **A** is a unitary matrix, we have $$\mathbf{A}^{-1} = \mathbf{A}^{*T}$$

$$\mathbf{A}^{*T}\mathbf{Y}\mathbf{A}^{*} = \mathbf{A}^{*T}(\mathbf{A}\mathbf{X}\mathbf{A}^{T})\ \mathbf{A}^{*} = \mathbf{I}\cdot\mathbf{X}\cdot\mathbf{I} = \mathbf{X}$$

Energy Compaction Property of 2D Unitary Transform

- Example

$$\mathbf{A} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 100 & 100 & 98 & 99 \\ 100 & 100 & 94 & 94 \\ 98 & 97 & 96 & 100 \\ 100 & 99 & 97 & 94 \end{bmatrix} \xrightarrow{\mathbf{Y} = \mathbf{AXA}^T} \mathbf{Y} = \begin{bmatrix} 391.5 & 0 & 5.5 & 1 \\ 2.5 & -2 & -4.5 & 2 \\ 1 & -0.5 & 2 & -0.5 \\ 2 & 1.5 & 0 & -1.5 \end{bmatrix}$$

A coefficient is called significant if its magnitude is above a pre-selected threshold *th*

insignificant coefficients (*th*=64)

Energy Conservation Property of 2D Unitary Transform

2-norm of a matrix $\mathbf{X}$

$$\|\mathbf{X}\|^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} | x_{ij} |^2$$

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T \quad A \text{ unitary} \quad \Rightarrow \quad \|\mathbf{Y}\|^2 = \|\mathbf{X}\|^2$$

<u>Example:</u>

$$\mathbf{A} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \xrightarrow{\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^T} \mathbf{Y} = \begin{bmatrix} 5 & -1 \\ -2 & 0 \end{bmatrix}$$

$$\|\mathbf{X}\|^2 = 1^2 + 2^2 + 3^2 + 4^2 = 30 = 5^2 + 2^2 + 1^2 + 0^2 = \| \mathbf{Y} \|^2$$

You are asked to prove such property in your homework

Implication of Energy Conservation

$$\mathbf{X} \longrightarrow \boxed{T} \longrightarrow \mathbf{Y} \longrightarrow \boxed{Q} \longrightarrow \hat{\mathbf{Y}} \longrightarrow \boxed{T^{-1}} \longrightarrow \hat{\mathbf{X}}$$

$$\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^{T} \qquad \hat{\mathbf{X}} = \mathbf{A}^{*T}\hat{\mathbf{Y}}\mathbf{A}^{*}$$

$$\mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{A}(\mathbf{X} - \hat{\mathbf{X}})\mathbf{A}^{T}$$

$$\Downarrow$$

$$\| \mathbf{Y} - \hat{\mathbf{Y}} \|^2 = \| \mathbf{X} - \hat{\mathbf{X}} \|^2$$

Similar to 1D case, quantization noise in the transform domain has the same energy as that in the spatial domain

Important 2D Unitary Transforms

Discrete Fourier Transform

- Widely used in non-coding applications (frequency-domain approaches)

Discrete Cosine Transform

- Used in JPEG standard

Hadamard Transform

- All entries are ± 1
- N=2: Haar Transform (simplest wavelet transform for multi-resolution analysis)

Transform Coding

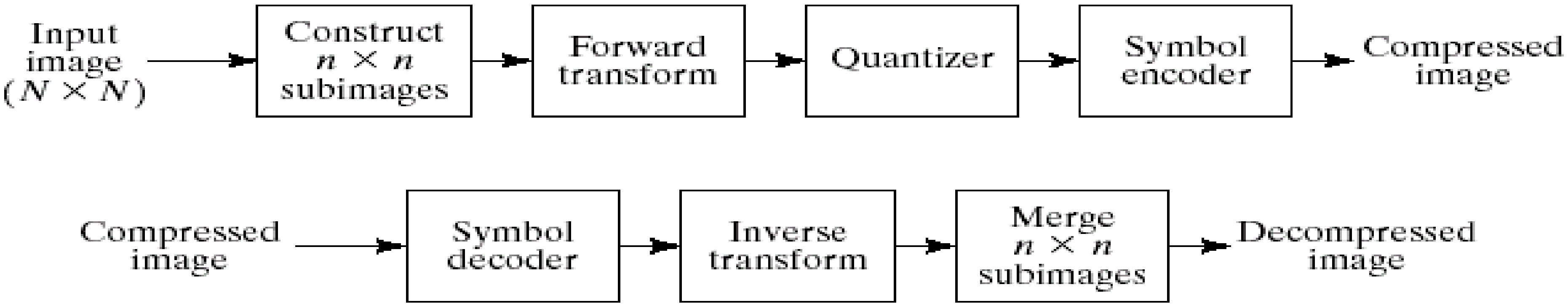
A reversible linear transform (such as Fourier Transform) is used to map the image into a set of transform coefficients

These coefficients are then quantized and coded.

The goal of transform coding is to decorrelate pixels and pack as much information into small number of transform coefficients.

Compression is achieved during quantization not during the transform step

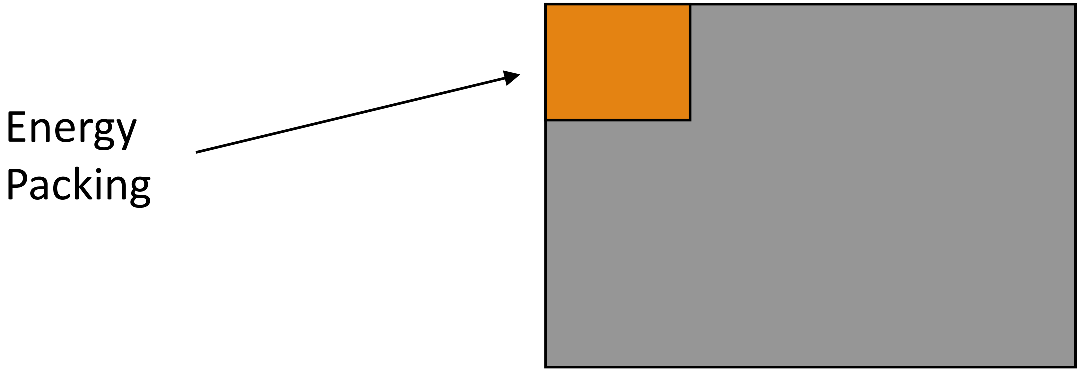
Transform Coding



2D Transforms

Energy packing

- 2D transforms pack most of the energy into small number of coefficients located at the upper left corner of the 2D array



2D Transforms

Consider an image *f(x,y)* of size *N x N*

Forward transform

$$T(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)g(x,y,u,v)$$

$$u,v = 0,1,2,\ldots, N-1.$$

g(x,y,u,v) is the forward transformation kernel or basis functions

2D Transforms

Inverse transform

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u,v) h(x,y,u,v)$$

$$x, y = 0, 1, 2, \ldots, N-1.$$

h(x,y,u,v) is the inverse transformation kernel or basis functions

Discrete Cosine Transformation (DCT)

DCT (1D)

Discrete cosine transform

$$
\begin{aligned}
C(u) &= \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \\
\alpha(k) &= \left\{ \begin{array}{ll} \sqrt{\frac{1}{N}} & \text{if } k = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise} \end{array} \right\}.
\end{aligned}
$$

The strength of the ‘u’ sinusoid is given by C(u)

- Project f onto the basis function
- All samples of f contribute the coefficient
- C(0) is the zero-frequency component – the average value!

DCT (1D)

Consider a digital image such that one row has the following samples

Index	0	1	2	3	4	5	6	7
Value	20	12	18	56	83	10	104	114

There are 8 samples so N=8

u is in [0, N-1] or [0, 7]

Must compute 8 DCT coefficients: C(0), C(1), …, C(7)

Start with C(0)

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

DCT (1D)

$$
\begin{aligned}
C(0) &= \sqrt{\tfrac{1}{8}} \sum_{x=0}^{7} f(x) \cos\left(\frac{(2x+1)\cdot 0\pi}{2\cdot 8}\right) \\
&= \sqrt{\tfrac{1}{8}} \sum_{x=0}^{7} f(x) \cos(0) \\
&= \sqrt{\tfrac{1}{8}} \cdot \{f(0)+f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)\} \\
&= .35 \cdot \{20+12+18+56+83+110+104+115\} \\
&= 183.14
\end{aligned}
$$

DCT (1D)

Repeating the computation for all u, we obtain the following coefficients

Spatial domain

f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	f(6)	f(7)
20	12	18	56	83	110	104	114

Frequency domain

$C(0)$	$C(1)$	$C(2)$	$C(3)$	$C(4)$	$C(5)$	$C(6)$	$C(7)$
183.1	-113.0	-4.2	22.1	10.6	-1.5	4.8	-8.7

DCT (2D)

The 2D DCT is given below where the definition for alpha is the same as before

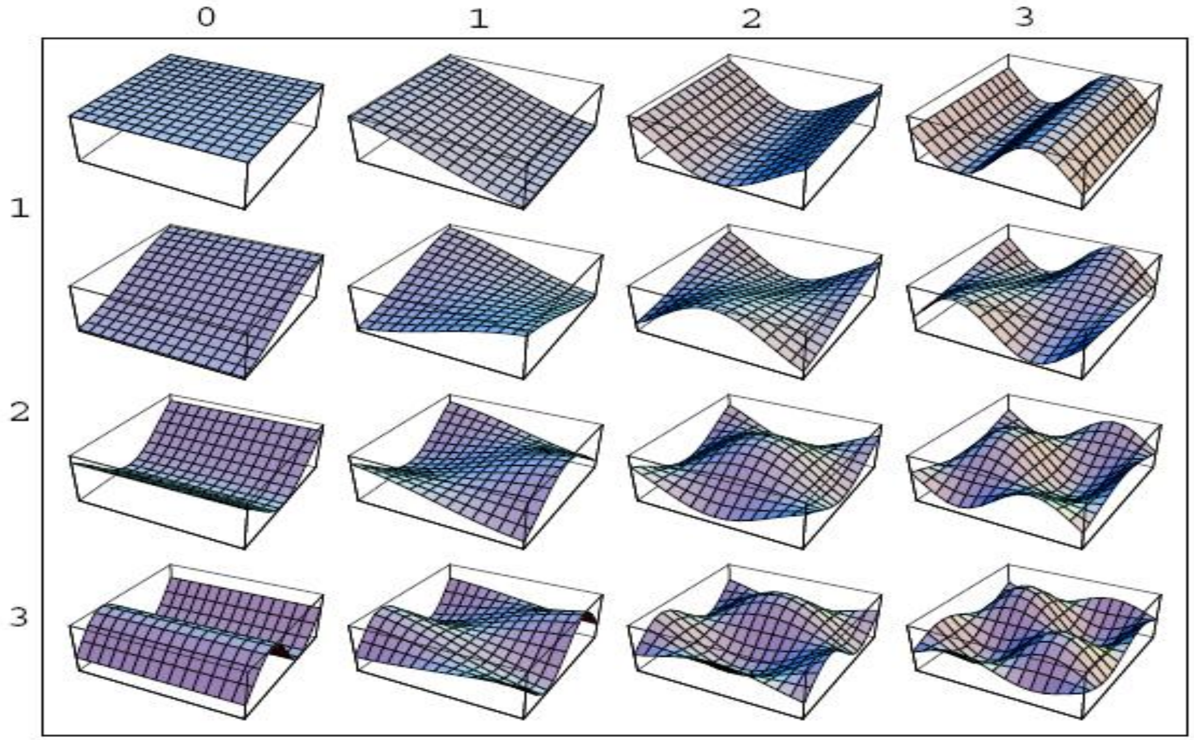
$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

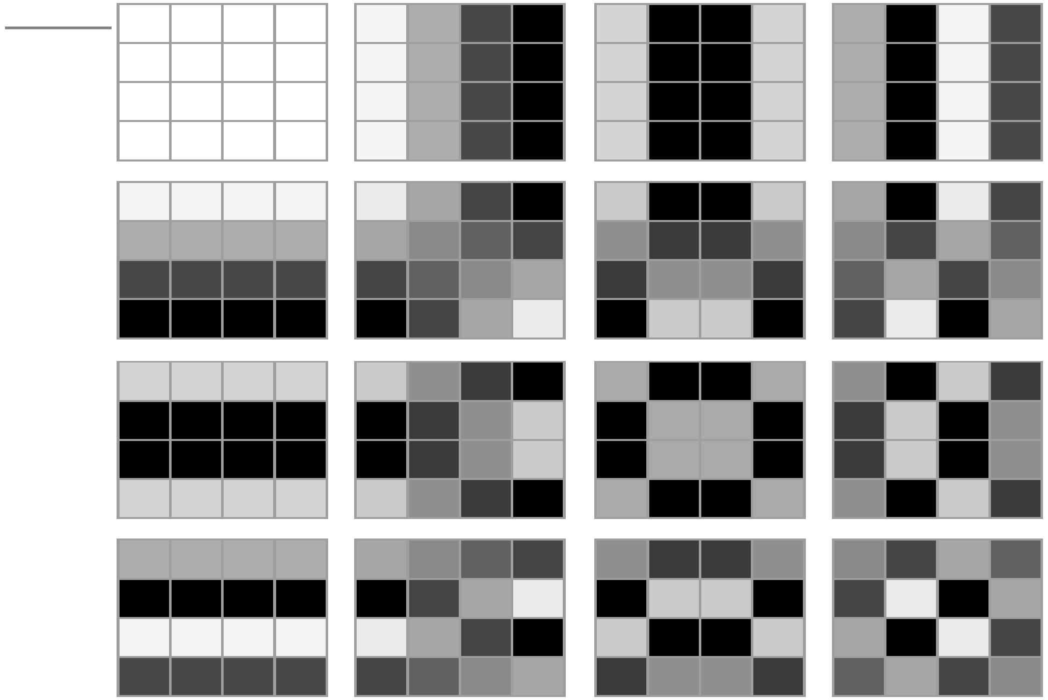
For an MxN image there are MxN coefficients

Each image sample contributes to each coefficient

Each (u,v) pair corresponds to a ‘pattern’ or ‘basis function’

DCT basis functions (patterns)





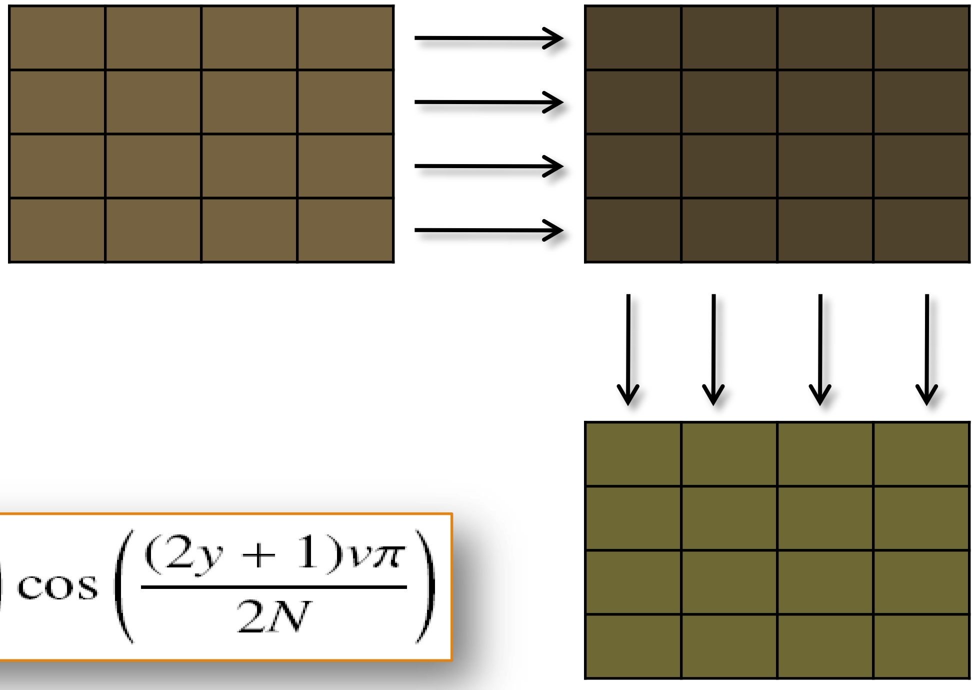
Basis functions

Basis patterns (imaged functions)

Separability

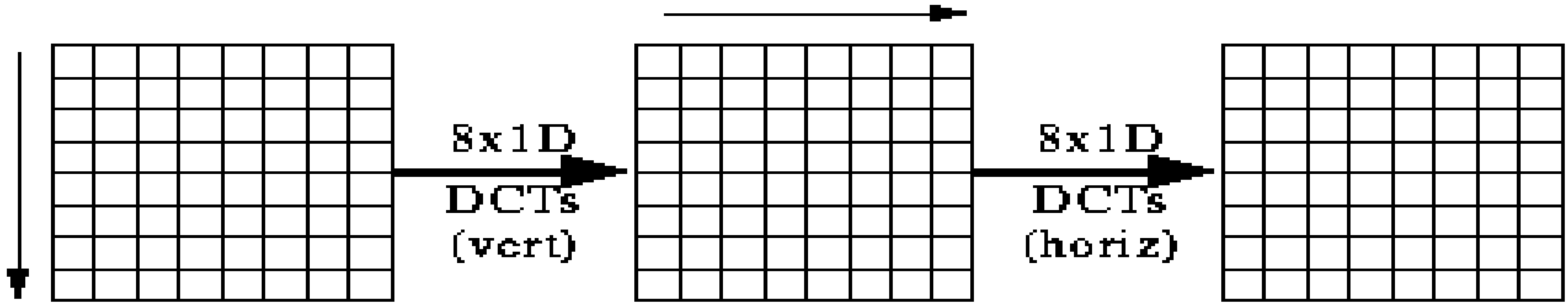
The DCT is separable

- The coefficients can be obtained by computing the 1D coefficients for each row
- Using the row-coefficients to compute the coefficients of each column (using the 1D forward transform)



$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

Separable



Invertability

The DCT is invertible

- Spatial samples can be recovered from the DCT coefficients

$$
\begin{aligned}
f(x) &= \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \\
f(x,y) &= \sum_{u=0}^{N-1} \sum_{v}^{N-1} \alpha(u)\alpha(v) C(u,v) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right)
\end{aligned}
$$

Summary of DCT

- The DCT provides energy compaction
  - Low frequency coefficients have larger magnitude (typically)
  - High frequency coefficients have smaller magnitude (typically)
  - Most information is compacted into the lower frequency coefficients (those coefficients at the ‘upper-left’)

- Compaction can be leveraged for compression
  - Use the DCT coefficients to store image data but discard a certain percentage of the high-frequency coefficients!
  - JPEG does this

Example: Energy Compaction

- Original Lena image

- 2D DCT

Example: Energy Compaction

(a)

(b)

75% coefficients are Discarded

Compressed

98% DC coefficients are discarded

Compressed

Suggested Readings

- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you