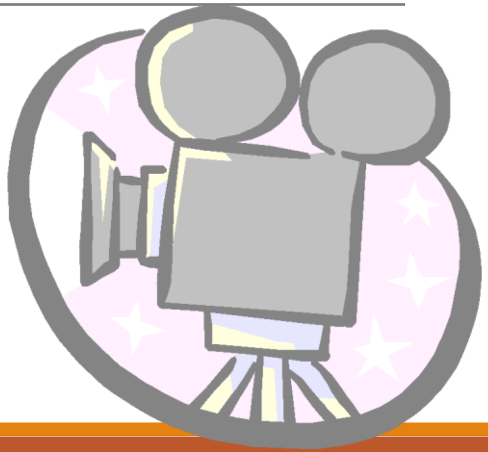
Image Processing

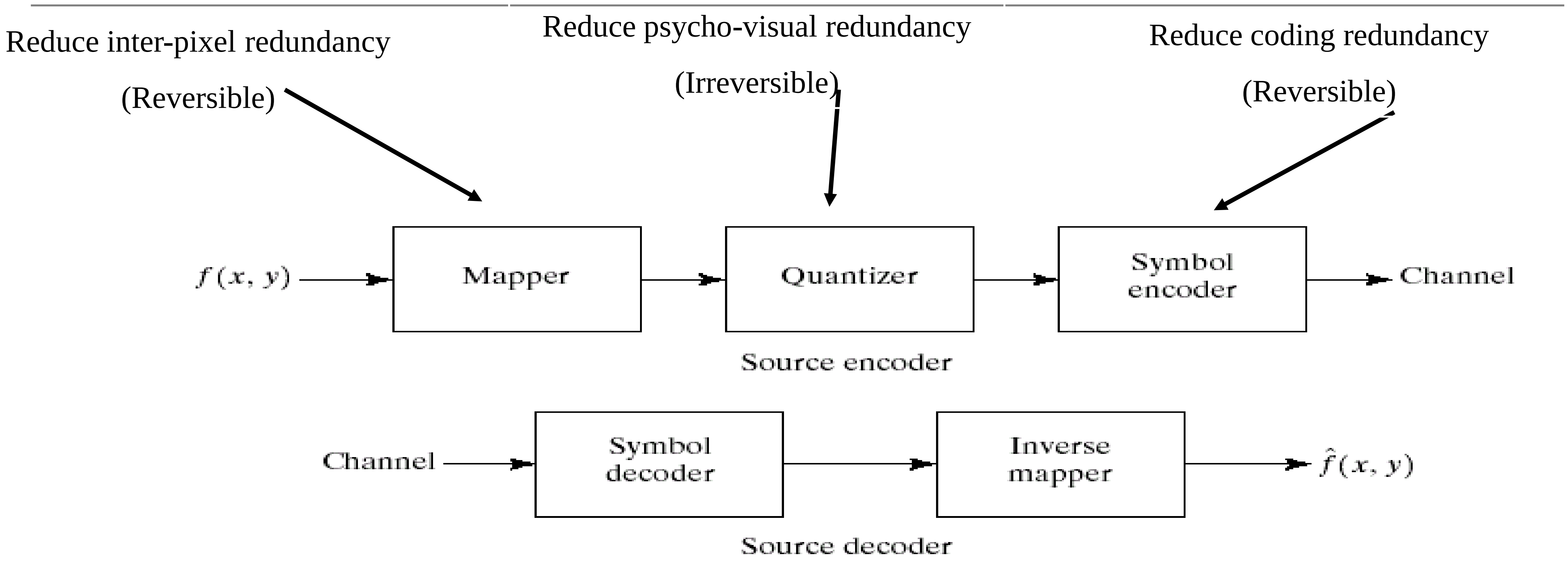
CS-317/CS-341



Outline

- Lossless Compression
  - Run length coding
- Lossy image compression techniques

Image Compression model



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

Run-length encoding

- Run-length encoding is probably the simplest method of compression.

- It can be used to compress data made of any combination of symbols.

- It does not need to know the frequency of occurrence of symbols and can be very efficient if data is represented as 0s and 1s.

- The general idea behind this method is to replace consecutive repeating occurrences of a symbol by one occurrence of the symbol followed by the number of occurrences.

    replace runs of symbols (possibly of length one) with pairs of (run-length, symbol)
    For images, the maximum run-length is the size of a row

- The method can be even more efficient if the data uses only two symbols (for example 0 and 1) in its bit pattern and one symbol is more frequent than the other.

Run-length coding (RLC) (addresses interpixel redundancy)

Reduce the size of a repeating string of symbols (i.e., runs):

1 1 1 1 0 0 0 0 0 0 1 → (1,5) (0, 6) (1, 1)

a a a b b b b b b c c → (a,3) (b, 6) (c, 2)

No. of vector=3, maximum length=6, so 3 bits in binary are required, no. of bits per pixel=1

Size=no. of vcetors* (bit requirement for each vector+no. of bits per pixel)

3*(3+1)=12

Original size= 12*1=12

Run-length coding (RLC) (addresses interpixel redundancy)

0	0	0	0	0
0	0	0	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Reduce the size of a repeating string of symbols (i.e., runs):

Horizontal RLC

No. of vectors : (0,5) ;(0, 3), (1, 2);(1,5);(1,5);(1,5)

No. of vector=6, maximum length=5, so 3 bits in binary are required, no. of bits per pixel=1

Size=no. of vcetors* (bit requirement for each vector+no. of bits per pixel)

6*(3+1)=24

Original size= 5*5*1=25, CR=25/24=1.042:1

Run-length coding (RLC) (addresses interpixel redundancy)

0	0	0	0	0
0	0	0	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Reduce the size of a repeating string of symbols (i.e., runs):

Vertical RLC

No. of vectors : (0,2) (1,3); (0, 2),(1,3);(0,2),(1,3));(0,1),(1,4);(0,1), (1,4)

No. of vector=10, maximum length=4, so 3 bits in binary are required, no. of bits per pixel=1

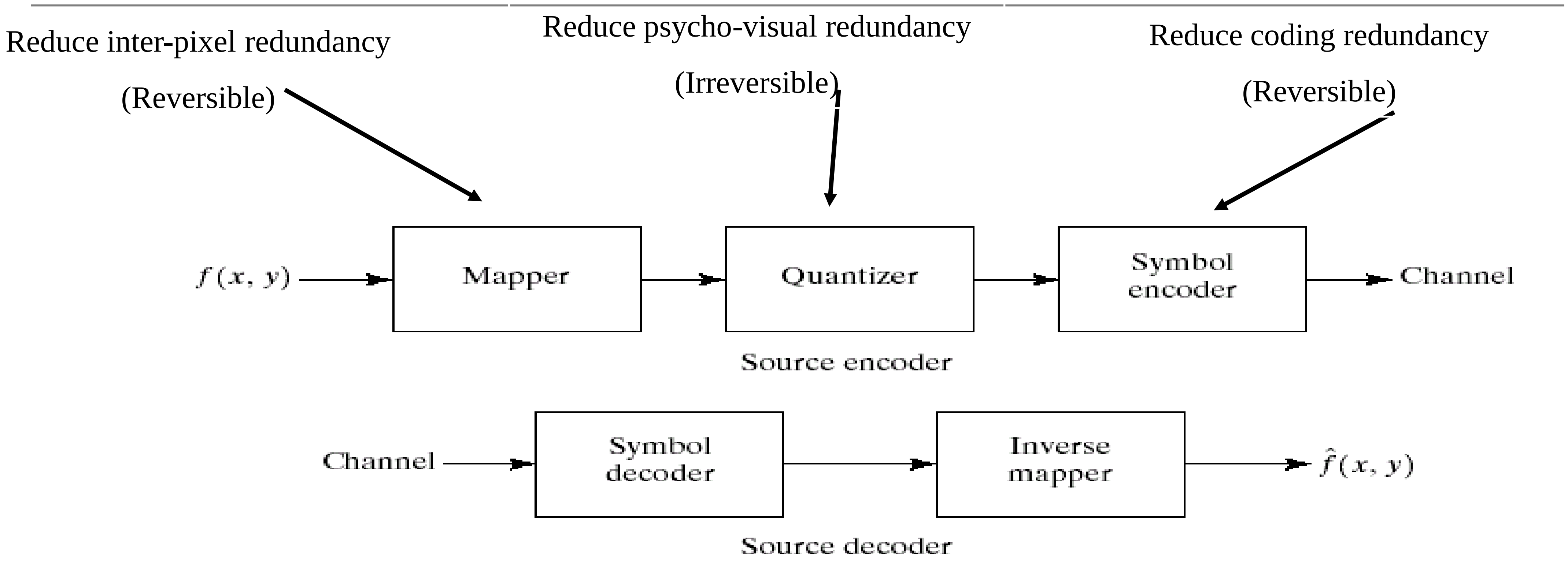
Size=no. of vcetors* (bit requirement for each vector+no. of bits per pixel)

10*(3+1)=40

Original size= 5*5*1=25, CR=25/40=0.625:1

Lossy Image Compression

Image Compression model



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

Why Lossy?

In most applications related to consumer electronics, lossless compression is not necessary

- What we care is the subjective quality of the decoded image, not those intensity values

With the relaxation, it is possible to achieve a higher compression ratio (CR)

- For photographic images, CR is usually below 2 for lossless, but can reach over 10 for lossy

A Simple Experiment

Bit-plane representation

$$A = a_0 + a_1 2 + a_2 2^2 + \ldots \ldots + a_7 2^7$$

Least Significant Bit
(LSB)

Most Significant Bit
(MSB)

<u>Example</u>

$$A = 129 \rightarrow a_0 a_1 a_2 \ldots a_7 = 10000001$$

$$a_0 a_1 a_2 \ldots a_7 = 00110011 \rightarrow A = 4 + 8 + 64 + 128 = 204$$

A Simple Experiment (Con't)

- How will the reduction of gray-level resolution affect the image quality?
  - Test 1: make all pixels even numbers (i.e., knock down a_0 to be zero)
  - Test 2: make all pixels multiples of 4 (i.e., knock down a_0,a_1 to be zeros)
  - Test 3: make all pixels multiples of 4 (i.e., knock down a_0,a_1,a_2 to be zeros)

Experiment Results

original

Test 1

Test 2

Test 3

How to Measure Image Quality?

Subjective
- Evaluated by human observers
- Do not require the original copy as a reference
- Reliable, accurate yet impractical

Objective
- Easy to operate (automatic)
- Often requires the original copy as the reference (measures fidelity rather than quality)

Objective Quality Measures

Mean Square Error (MSE)

original decoded

$$MSE = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}[X(i,j) - Y(i,j)]^2$$

Peak Signal-to-Noise-Ratio (PSNR)

$$PSNR = 10\log_{10}\frac{255^2}{MSE}(dB)$$

<u>Question</u>:
Can you think of a counter-example to prove objective measure is not consistent with subjective evaluation?

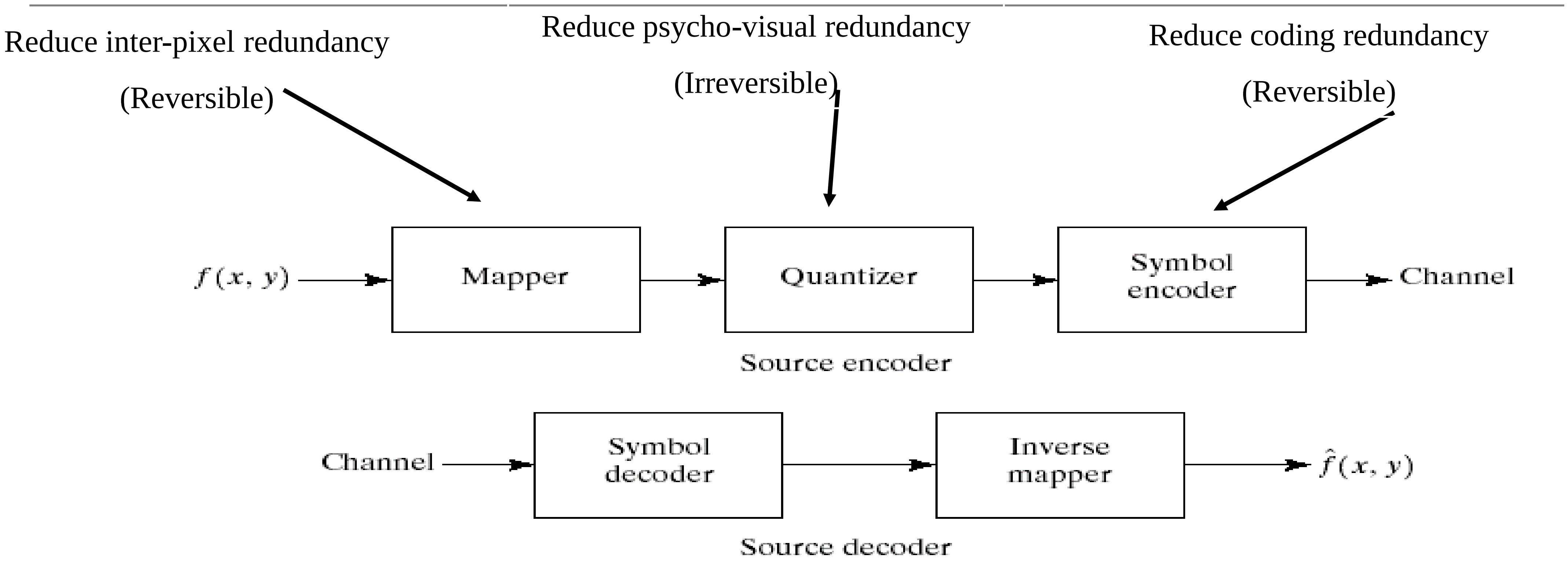
Results

Shifted (MSE=337.8)

Original *cameraman* image

By shifting the last row of the image to become the first row, we affect little on subjective quality but the measured MSE is large

Image Compression model



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

Lossy Image Compression

Quantization basics

- Uniform Quantization

What is Quantization?

In image compression

- To limit the possible values of a pixel value or a transform coefficient to a discrete set of values by information theoretic rules

Examples

Unlike entropy, we encounter it everyday (so it is not a monster)

Continuous to discrete

- a quarter of milk, two gallons of gas, normal temperature is 98.6F, my height is 5 foot 9 inches

Discrete to discrete

- Round your tax return to integers
- The mileage of my car is about 55K.

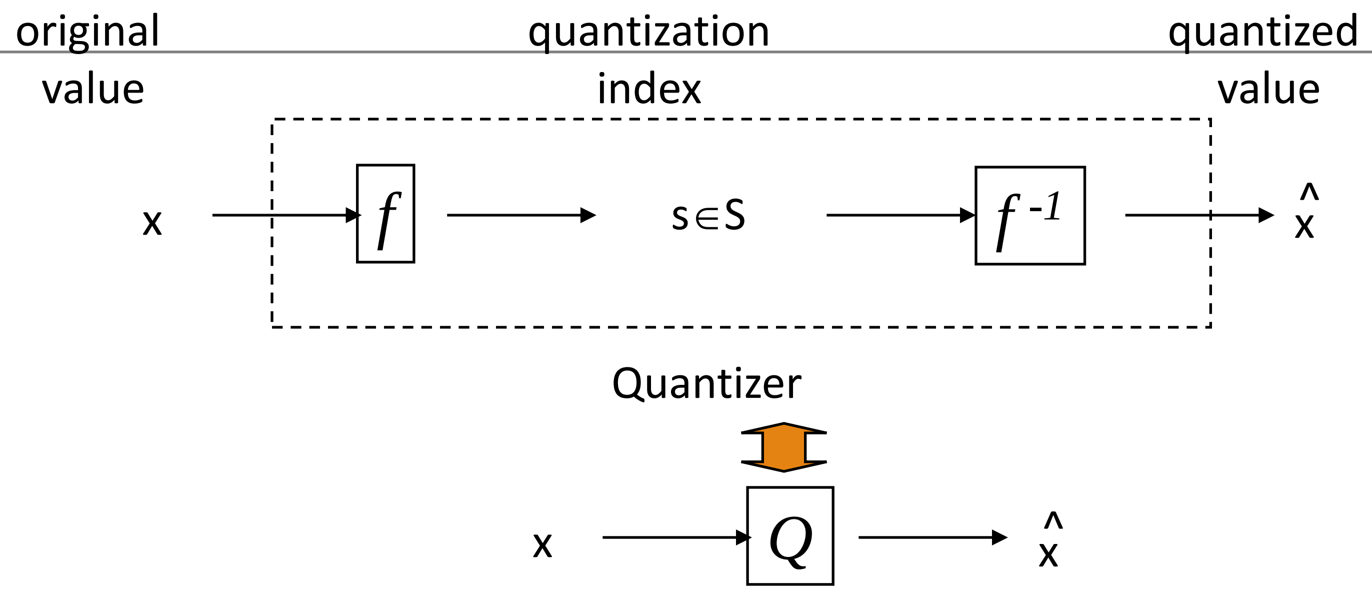
Scalar vs. Vector Quantization

We only consider the scalar quantization (SQ) in this course

- Even for a sequence of values, we will process (quantize) each sample independently

Vector quantization (VQ) is the extension of SQ into high-dimensional space

Definition of (Scalar) Quantization



*f finds the **closest** (in terms of Euclidean distance) approximation of x from a **codebook** C (a collection of codewords) and assign its index to s;*
f^{-1} operates like a table look-up to return the corresponding codeword

Numerical Example-3

$$Q(x) = 8 + \left\lfloor \frac{x}{16} \right\rfloor \cdot 16,\ x \in [0,255]$$

$$x = \begin{matrix} 225 & 222 & 235 & 228 \\ 220 & 206 & 209 & 44 \\ 49 & 56 & 64 & 42 \\ 128 & 106 & 94 & 27 \end{matrix} \longrightarrow \boxed{Q} \longrightarrow \hat{x} = \begin{matrix} 232 & 216 & 232 & 232 \\ 216 & 200 & 216 & 40 \\ 56 & 56 & 72 & 40 \\ 136 & 104 & 88 & 24 \end{matrix}$$

Notes

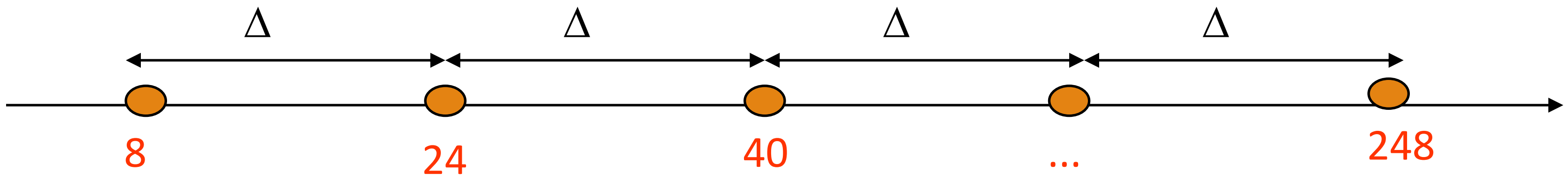
- For scalar quantization, each sample is quantized independently
- Quantization is irreversible

Uniform Quantization (UQ) for Uniform Distribution

Uniform Quantization

A scalar quantization is called uniform quantization (UQ) if all its codewords are uniformly distributed (equally-distanced)

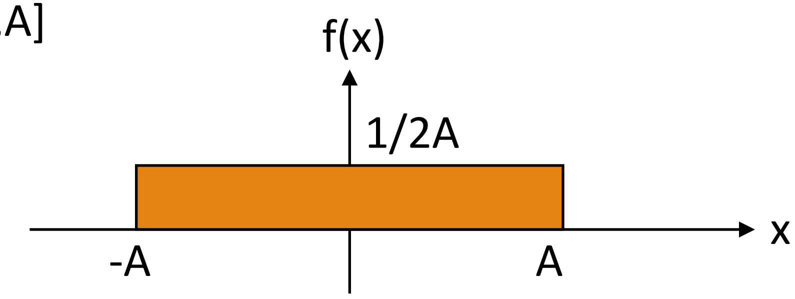
Example (quantization stepsize Δ=16)



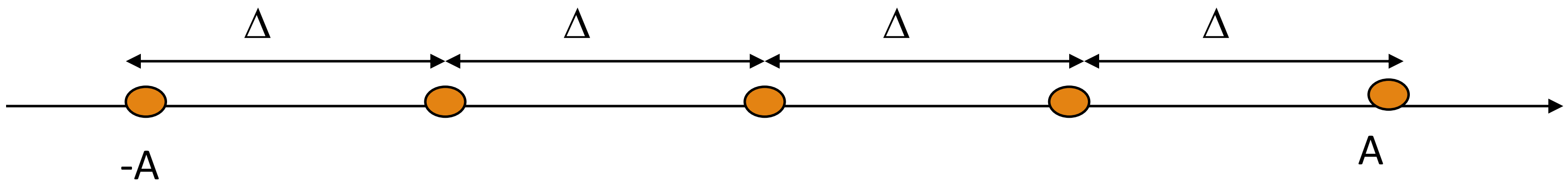
Uniform Distribution

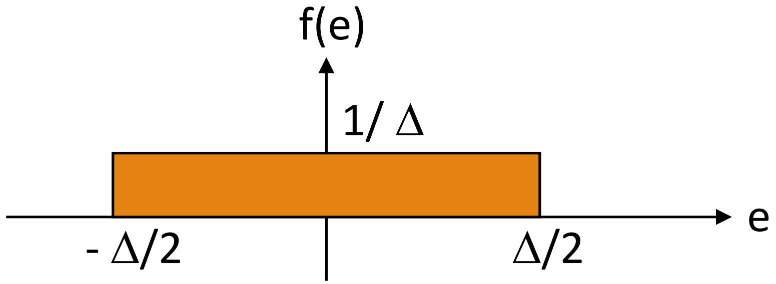
denoted by U[-A,A]

$$f(x) = \begin{cases} \frac{1}{2A} & x \in [-A, A] \\ 0 & else \end{cases}$$



Quantization Noise of UQ





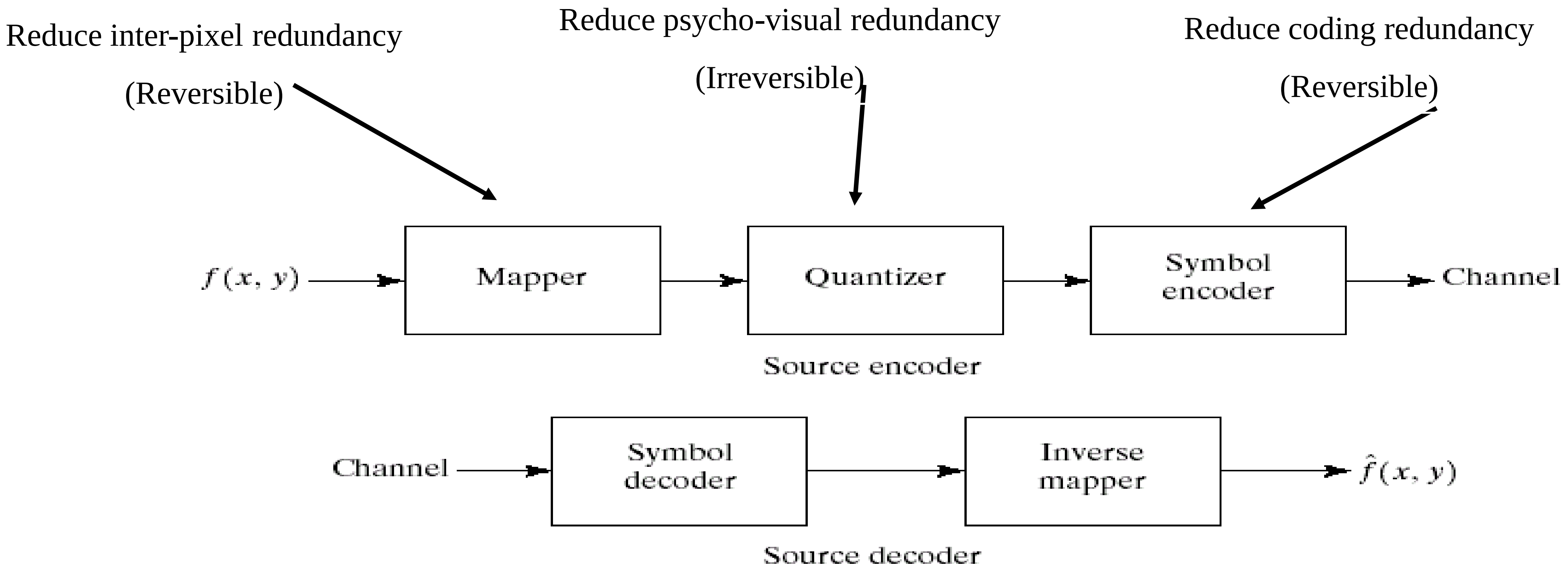
Quantization noise of UQ on uniform distribution is also uniformly distributed

<u>Recall</u> Variance of U[- Δ/2, Δ/2] is

$$\sigma^2 = \frac{1}{12}\Delta^2$$

Basic Transformation

Image Compression model : Transformation

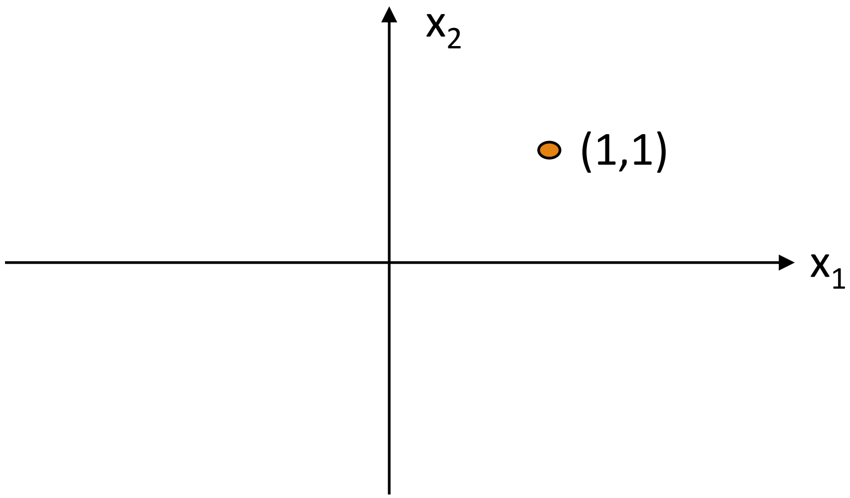


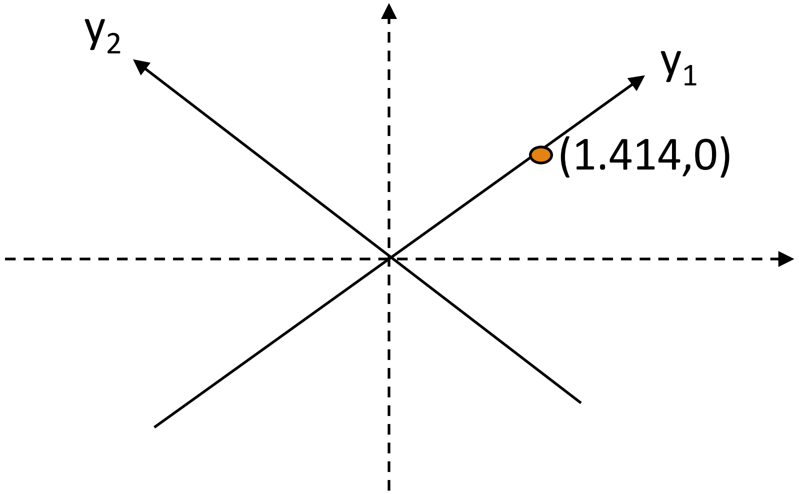
(a) Source encoder and (b) source decoder model.

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

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

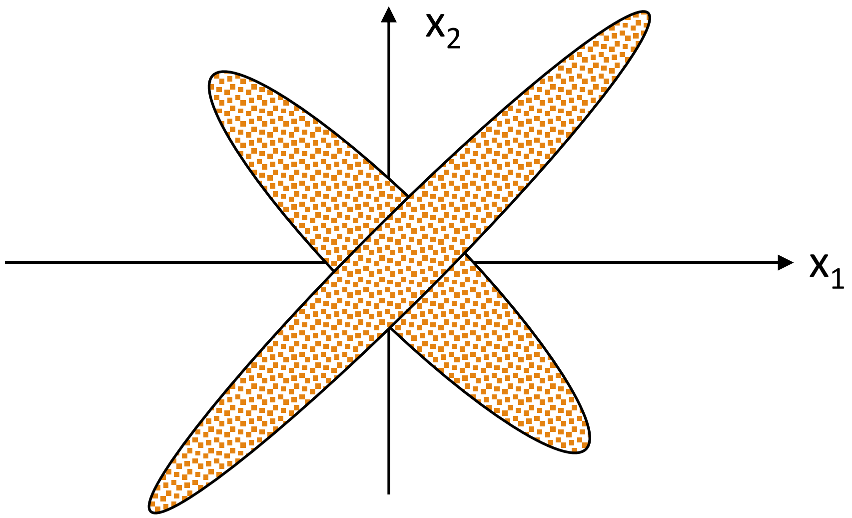
$$\Updownarrow$$

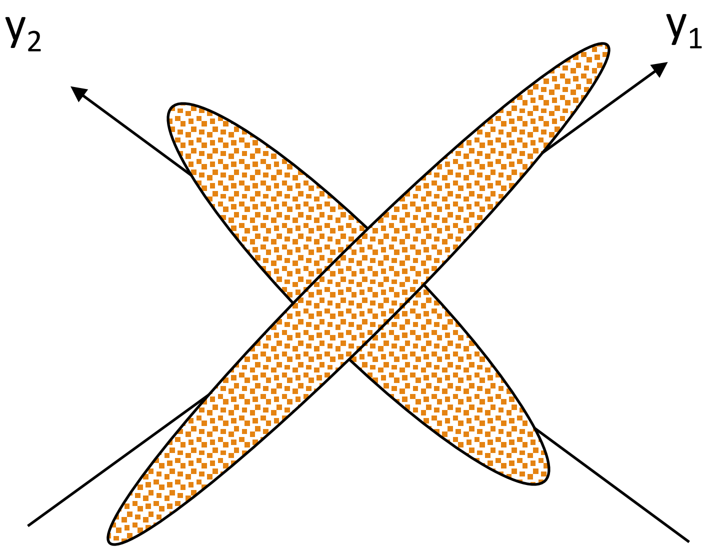
$$\vec{y} = \sum_{i=1}^{N} x_i \vec{b}_i, \quad \vec{b}_i = [a_{i1}, \ldots, a_{iN}]^T$$

$\downarrow$

basis vectors (column vectors of transform matrix)

Decorrelating Property of Transform





x_1 and x_2 are highly correlated

$$p(x_1x_2) \neq p(x_1)p(x_2)$$

y_1 and y_2 are less correlated

$$p(y_1y_2) \approx p(y_1)p(y_2)$$

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 $\mathbf{A}$, 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

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

$$\Updownarrow$$

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

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

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

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

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

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

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

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

$\Downarrow$ **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

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

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

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

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

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

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 $\mathbf{Y}_1 = \mathbf{A}\mathbf{X}$ (left matrix multiplication first)

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

$$\Updownarrow$$

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

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

<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

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

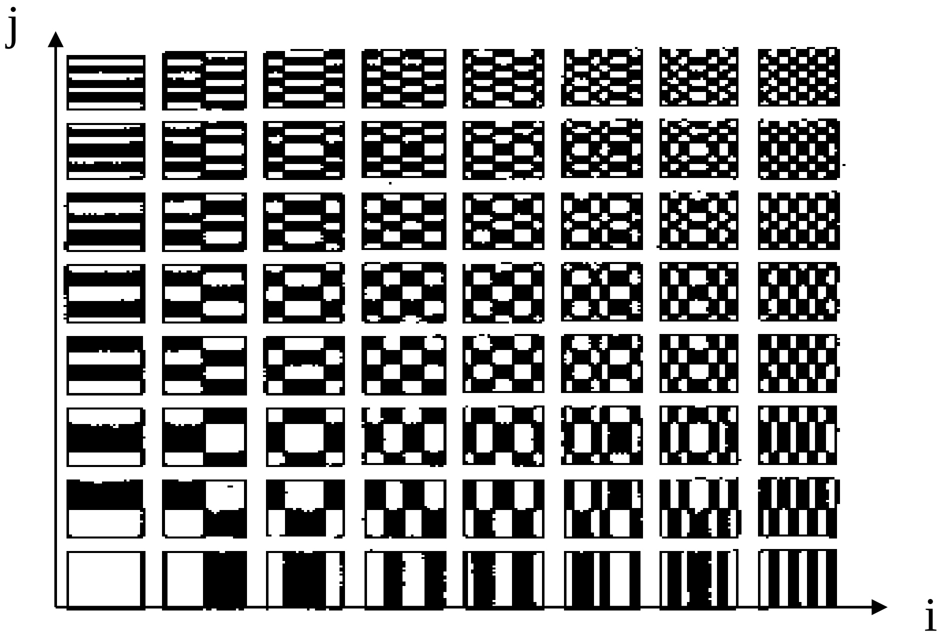
Example of Basis Images

Hadamard matrix:

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

$$\mathbf{A} = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix},$$

$\vec{b}_1$ $\vec{b}_8$



B_{ij}

2D Unitary Transform

Suppose A is a unitary matrix,

<u>forward transform</u>

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

<u>inverse transform</u>

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

<u>Proof</u>

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

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

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

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

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

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

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

$$\Downarrow$$

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

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)

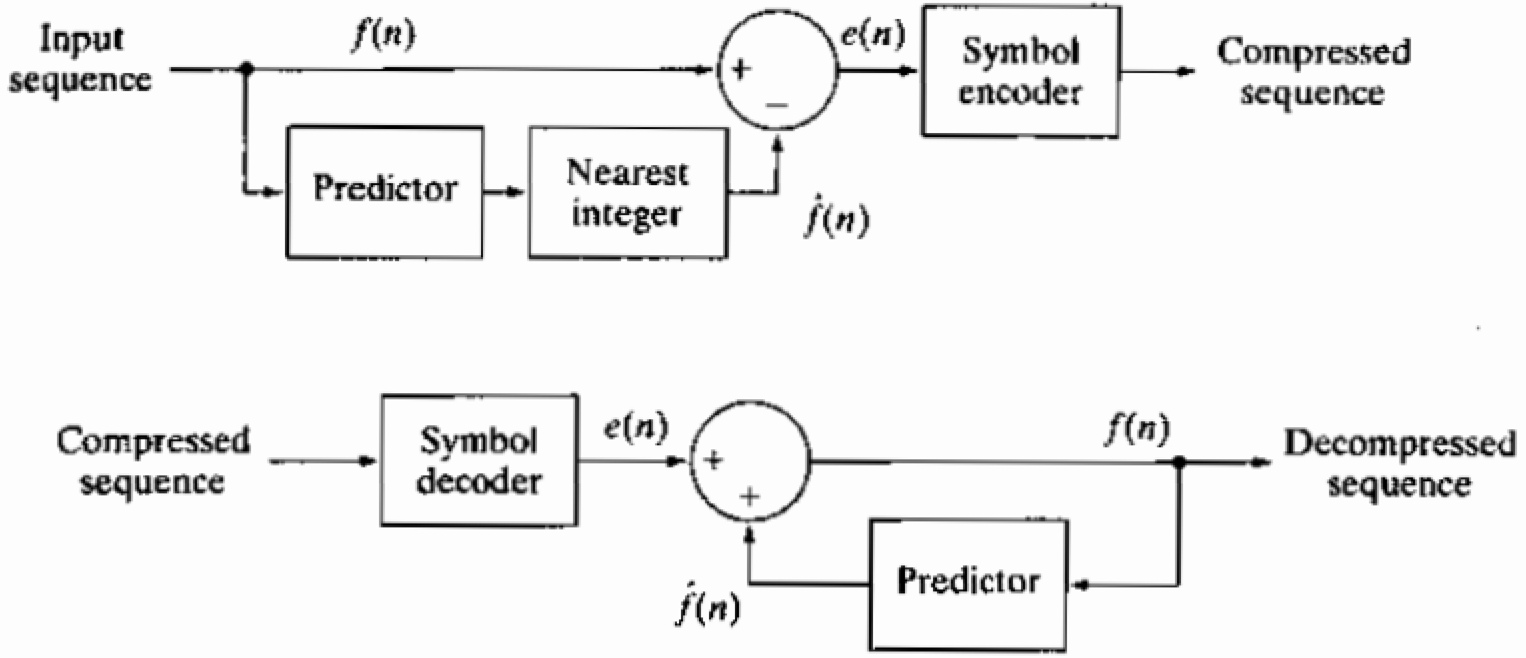
Discrete Cosine Transform

- Used in JPEG standard

Hadamard Transform

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

Predictive Coding



a
b

FIGURE 8.33 A lossless predictive coding model: (a) encoder; (b) decoder.

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