Wireless Communication Systems
@CS.NCTU

Lecture 5: Compression

Instructor: Kate Ching-Ju Lin (林靖茹)

Chap. 7-8 of "Fundamentals of Multimedia"
Some reference from http://media.ee.ntu.edu.tw/courses/dvt/15F/

Outline

- **Concepts of data compression**
- Lossless Compression
- Lossy Compression
- Quantization

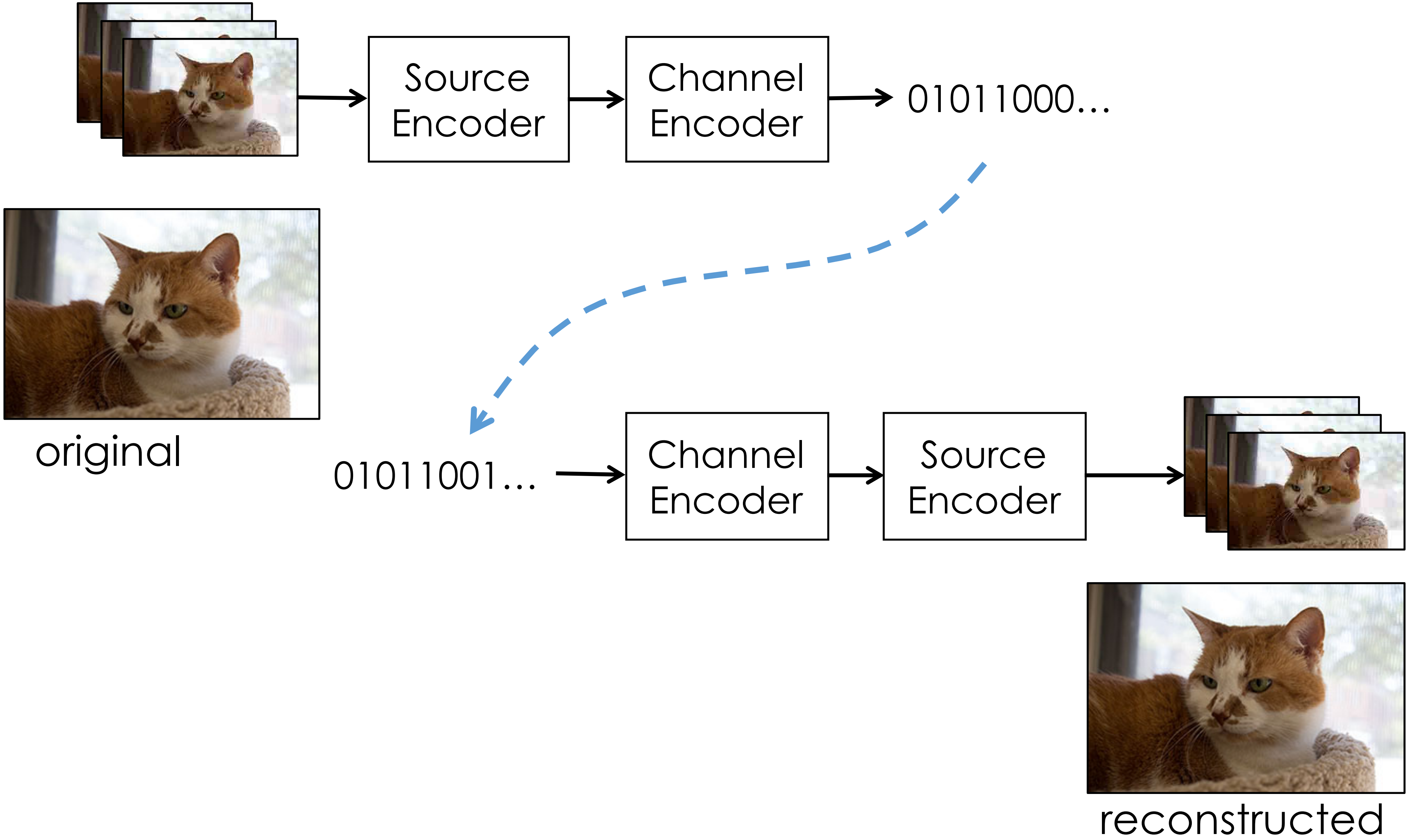
Why compression?

- Audio, image, and video require huge storage and network bandwidth if not compressed

Application	uncompressed	compressed
Audio conference	64kbps	16-64kbps
Video conference	30.41Mbps	64-768kbps
Digital video on CD-ROM (30fps)	60.83Mbps	1.5-4Mbps
HDTV (59.94fps)	1.33Gbps	20Mbps

Remove redundancy!

Compression Concepts



Compression Concepts

- **Source Coding**
  - Also known as data compression
  - The objective is to reduce the size of messages
  - Achieved by removing redundancy
  - Entropy encoding: minimize the size of messages according to a probability model
- **Channel Coding**
  - Also known as error correction
    - Repetition codes, parity codes, Reed-Solomon codes, etc.
  - Ensure the decoder can still recover the original data even with errors and (or) losses
  - Should consider the probability of errors happening during transmission (e.g., random loss or burst loss)

Considerations for Compression

- Lossless vs. Lossy
- Quality vs. bit-rate
- Variable bit rate (VBR) vs. constant bit rate (CBR)
- Robustness
  - Combat noisy channels
- Complexity
  - Encoding and decoding efficiency

Compression Performance

- Compression ratio = $\frac{\text{size}_{\text{before}}}{\text{size}_{\text{after}}}$
- Signal quality
  - Signal-to-noise ratio

    $$\text{SNR} = 10\log_{10}(\frac{\sigma_s^2}{\sigma_n^2})$$

  - Peak-Signal-to-noise ratio

    $$\text{PSNR} = 10\log_{10}(\frac{\sigma_{\text{peak}}^2}{\sigma_n^2})$$

  - Mean Opinion Score (MOS)
    - very annoying, annoying, slightly annoying, perceptible but not annoying, imperceptible
- Goal:
  - **Higher signal quality with higher compression ratio**

Mean square error

$$\sigma_n^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$$

Compression Technologies

- **Statistical redundancy**
  - Lossless compression
  - Also known as entropy coding
  - Build on the probabilistic characteristics of signals

- **Perceptual redundancy**
  - Lossy compression
  - Lead to irreversible distortion
  - Complex and depends on context or applications

Information Theory

- Consider an information source with alphabet $S = \{s_1, s_2, \ldots, s_n\}$, the self-information contained in s_i is defined as

$$i(s_i) = \log_2 \frac{1}{p_i}$$

  where p_i is teh probability that symbol s_i in S will occur

- Key idea of variable length coding
  - Frequent symbols → represented by less bits
  - Infrequent symbols → represented by more bits

Low probability p_i → Large amount of information
High probability p_i → Small amount of information

Information Theory - Entropy

- Entropy η of an information source
  - **Expected** self-information of the whole source

$$\eta = H(S) = \sum_{i=1}^{n} p_i * i(s_i) = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i}$$

$$= -\sum_{i=1}^{n} p_i \log_2 p_i$$

  - Measure the disorder of a system → **more entropy, more disorder**
    - Greater entropy when the distribution is flat
    - Smaller entropy when the distribution is more peaked
  - **Shannon's theory:** best lossless compression generates an average number of bits equal to entropy

Claude Elwood Shannon, "A mathematical theory of communication," Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, Jul. and Oct. 1948

Properties of Compression

- **Unique decodable**
  - Encode: $y = f(x)$
  - Decode: $x = f^{-1}(y)$ → there exists only a single solution
  - A code is <u>not</u> unique decodable if $f(x_i) = f(x_j) = y$ for some $x_i \neq x_j$

x: symbol
y: codeword

- **Instantaneous code**
  - Also called prefix-free code or prefix code
  - Any codeword cannot be the prefix of any other codeword, i.e., y_i not the prefix of y_j for all $y_i \neq y_j$
  - Why good?
    - When a message is sent, the recipient can decode the message unambiguously from the beginning

Properties – Examples

- **Non-unique decodable**

$s_1 = 0$
$s_2 = 01$
$s_3 = 11$
$s_4 = 00$

0011 could be $s_4 s_3$ or $s_1 s_1 s_3$

- **Non-Instantaneous code**

$s_1 = 0$
$s_2 = 01$
$s_3 = 011$
$s_4 = 11$

Coded sequence: 0111111 .... 11111 (last "11" underlined and labelled s_4)

→ Decode until receiving all bits

Outline

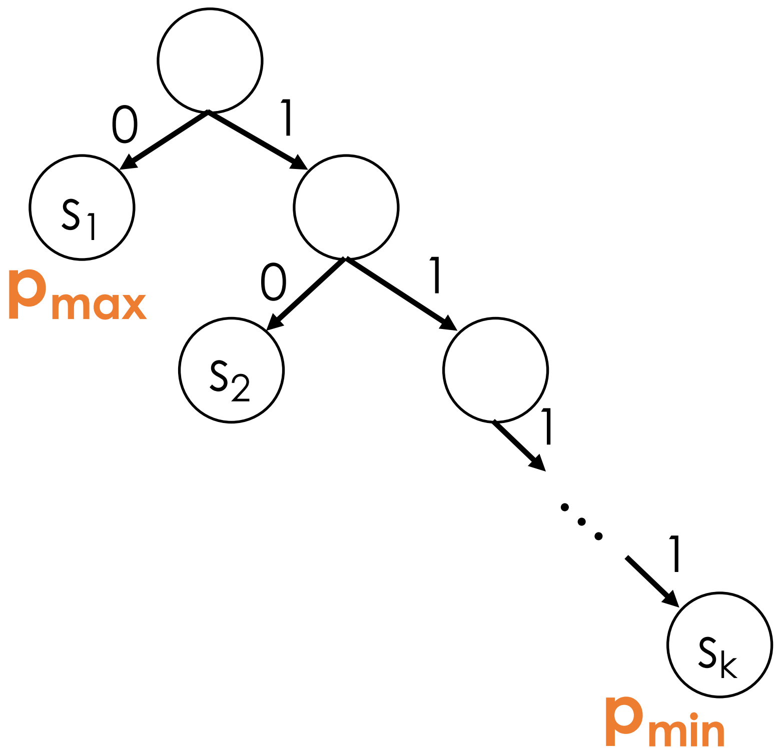
- Concepts of data compression
- **Lossless Compression**
- Lossy Compression
- Quantization

Lossless Compression

- Commonly known as entropy coding
- Algorithms
  - Huffman coding
  - Adaptive Huffman coding
  - Arithmetic coding
  - Run-length coding
  - Golomb and Rice coding
  - DPCM

Huffman Coding

- Proposed by David A. Huffman in 1952
- Adopted in many applications, such as fax machines, JPEG and MPEG
- Bottom-up manner: build a **binary coding tree**
  - left branches are coded 0
  - right branches are coded 1
- High-level idea
  - Each leaf node is a symbol
  - Each path is a codeword
  - Less frequent symbol → longer codeword path



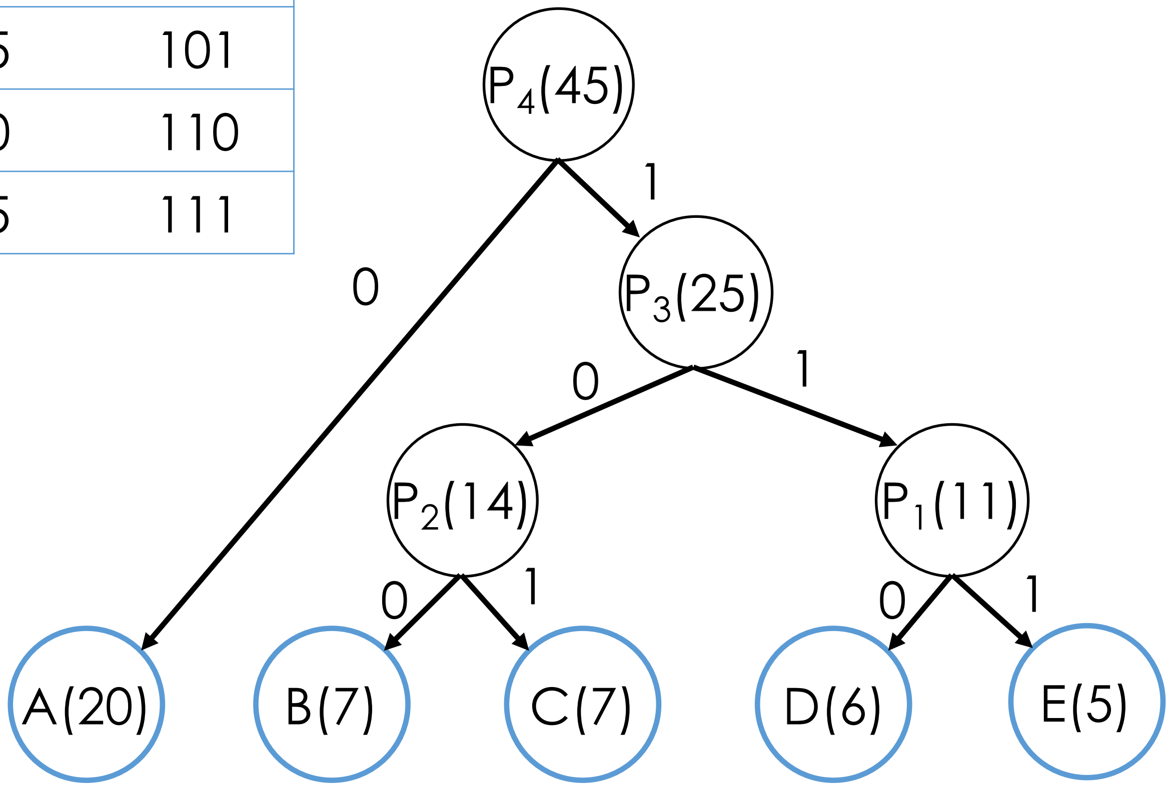
Huffman Coding

- Algorithm
  1. Sort all symbols according to their probabilities
  2. Repeat until only one symbol left
     a) Pick the two symbols with the **smallest** probabilities
     b) Add the two symbols as childe nodes
     c) Remove the two symbols from the list
     d) Assign the sum of the children's probabilities to the parent
     e) Insert the parent node to the list

Huffman Coding – Example

Symbol	Count	Probability	Code
A	15	0.375	0
B	7	0.175	100
C	7	0.175	101
D	6	0.150	110
E	5	0.125	111

1. {A, B, C, D, E}
2. {A, B, C, P_1}
3. {A, P_2, P_1}
4. {A, P_3}
5. {P_4}



Huffman Coding – Pro and Cons

- Pros
  - Unique decodable
  - Prefix code
  - Optimality: average codeword length of a message approaches its entropy
    → shown $\eta \leq E[L] \leq \eta+1$

- Cons
  - Every code has an integer bit length
  - Why inefficient?
    - If a symbol occurs very frequently
      $\log_2(1/p)$ close to 0 → but still need one bit

Arithmetic Coding

- Usually outperform Huffman coding
- Encode the whole message as one unit
- High-level idea
  - Each message is represented by an interval [a,b), 0 ≤ a,b ≤ 1
  - Longer message → shorter interval
    → more bits to represent a smaller real number
  - Shorter message → longer interval
    → less bits to represent a greater real number
  - Example

Symbol	low	high	range
	0	1.0	1.0
C	0.3	0.5	0.2
A	0.30	0.34	0.04
E	0.322	0.334	0.012
E	0.3286	0.3322	0.0036
$	0.33184	0.33220	0.00036

Arithmetic Coding – Encoding

- Maintain a probability table
  - Frequent symbol → larger range
  - Need a terminator symbol $
- Algorithm:
  - Initialize low = 0, high = 1, range = 1
  - Repeat for each symbol
    - low = low + range * $range_{min}$(symbol)
    - high = low + range * $range_{max}$(symbol)
    - Range = high - low

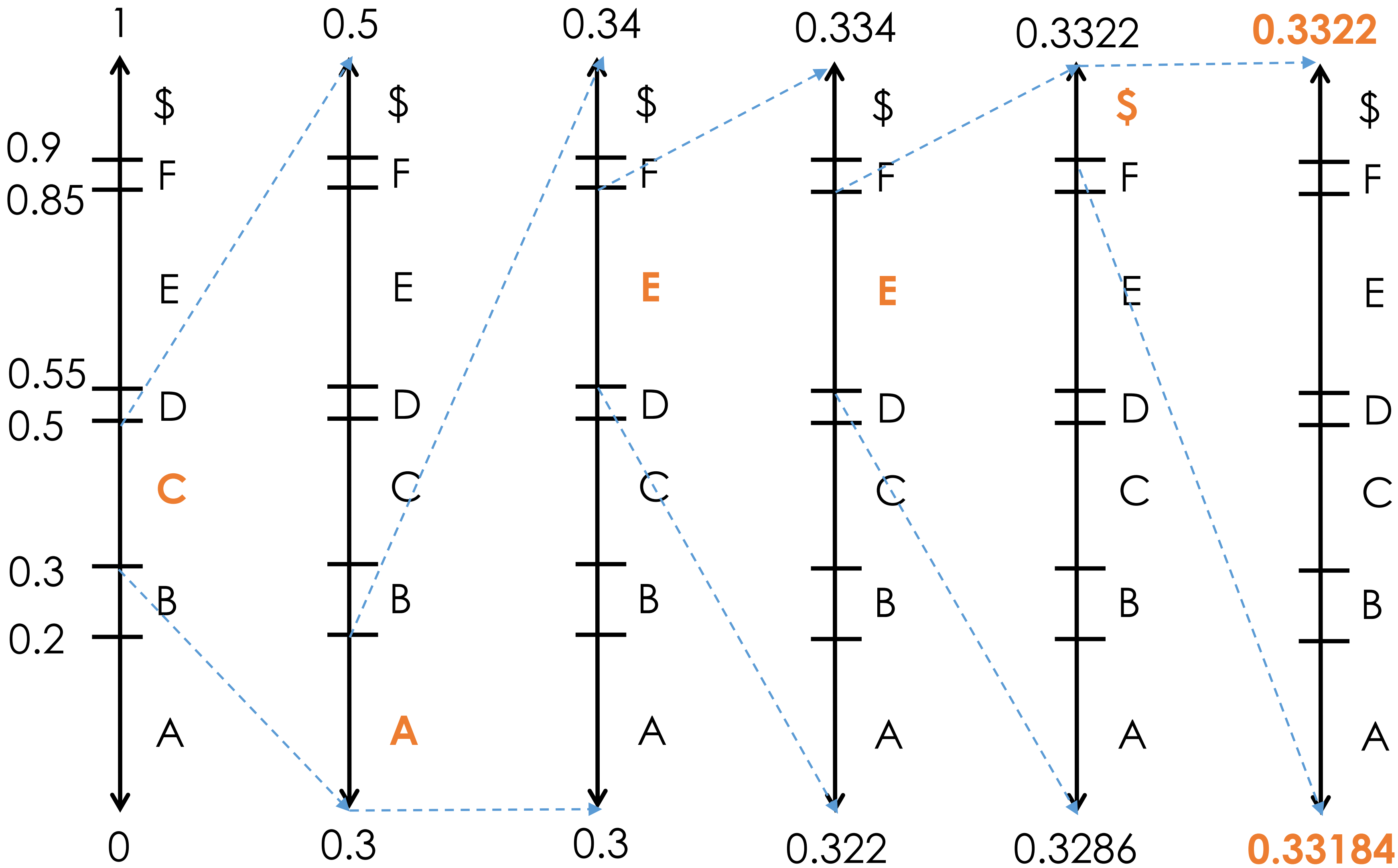
Sym	probability	range
A	0.2	[0, 0.2)
B	0.1	[0.2, 0.3)
C	0.2	[0.3, 0.5)
D	0.05	[0.5, 0.55)
E	0.3	[0.55, 0.85)
F	0.05	[0.85, 0.9)
$	0.1	[0.9, 1)

Encode a message CAEE$

Symbol	low	high	range
	0	1.0	1.0
C	0.3	0.5	0.2
A	0.30	0.34	0.04
E	0.322	0.334	0.012
E	0.3286	0.3322	0.0036
$	0.33184	0.33220	0.00036

Arithmetic Coding – Encoding

- Illustration



Arithmetic Coding – Decoding

- Algorithm
  - while not $
    1. Find a symbol s so that $range_{min}(s) \leq value \leq range_{max}(s)$
    2. Output s
    3. $low = range_{min}(s)$
    4. $high = range_{max}(s)$
    5. range =high – low
    6. value = (value – low) / range

Arithmetic Coding – Properties

- When the intervals shrink, we need very high-precision number for encoding
  - Might not be feasible
- Need a special terminator symbol $
  - Need to protect $ in noisy channels

Run-Length Coding

- Input sequence:

  0,0,-3,5,0,-2,0,0,0,0,2,-4,0,0,0,1

- Run-length sequence:

  (2,-3)(0,5)(1,-2)(4,2)(0,-4)(3,1)

Number of zeros   next non-zero value

- Many variations
- Reduce the number of samples to code
- Implementation is simple

Outline

- Concepts of data compression
- Lossless Compression
- **Lossy Compression**
- Quantization

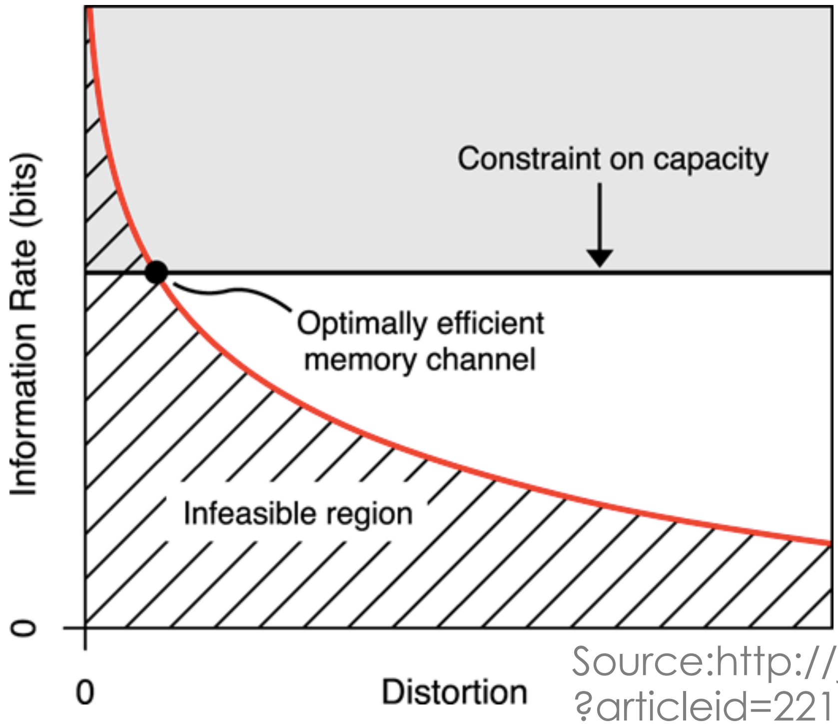
Compression Technologies

- Statistical redundancy
  - Lossless compression
  - Also known as entropy coding
  - Build on the probabilistic characteristics of signals

- **Perceptual redundancy**
  - Lossy compression
  - Lead to irreversible distortion
  - Complex and depends on context or applications

Rate-Distortion Function

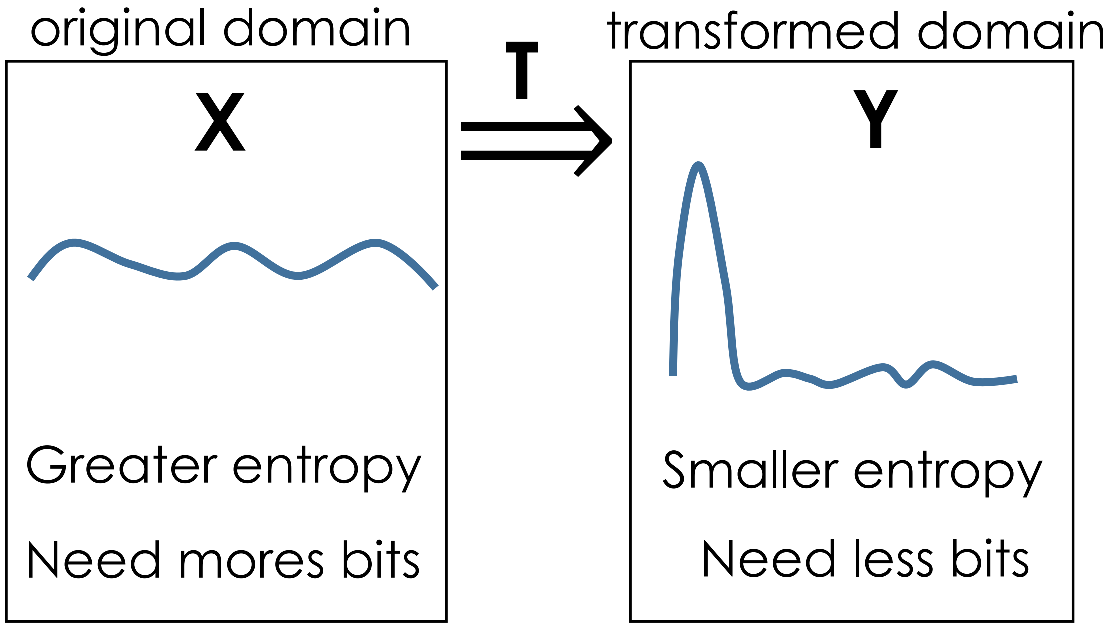
- Numerical measure for signal quality
  - SNR
  - PSNR
- How to evaluate the tradeoff between compression ratio and signal quality?
  - Rate-distortion function (D = 0 means lossless0



Source:http://jov.arvojournals.org/article.aspx?articleid=2213283

Transform Coding

- Remove spatial redundancy
  - Spatial image data are transformed in to a different representation: **transformed domain**
  - Make the image data easier to be compressed
  - Transformation (T) itself does not compress data
    - Compression is from quantization!



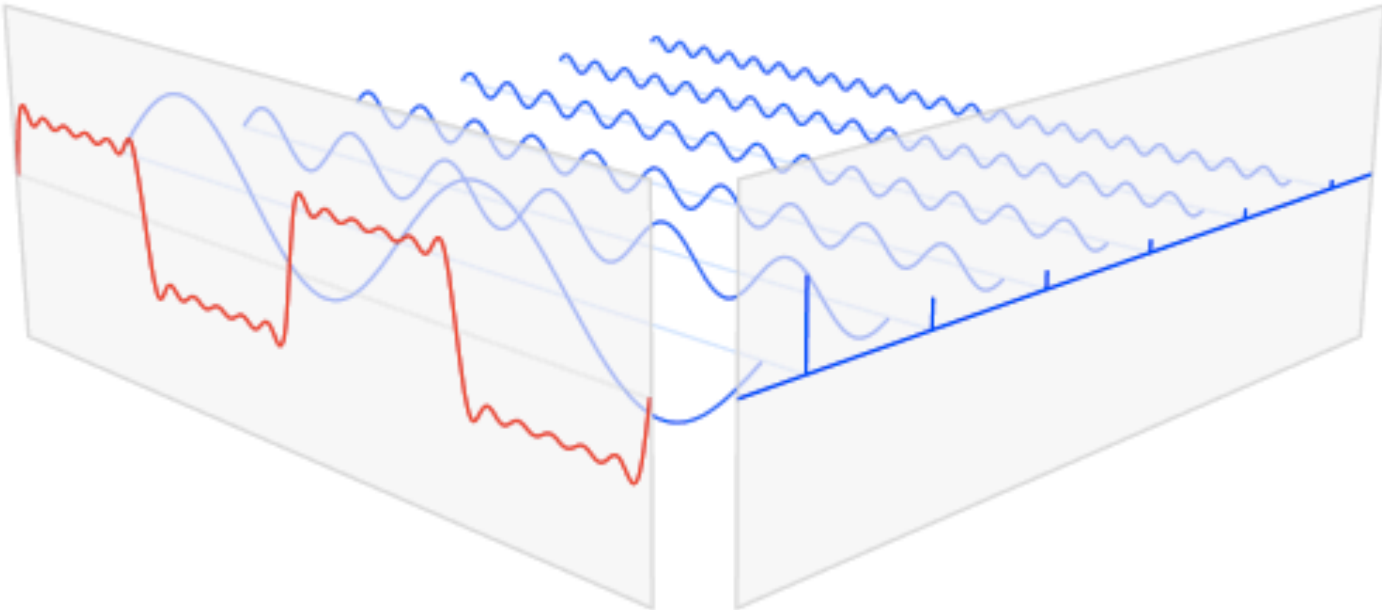
Fourier Analysis

- Fourier showed that any periodic signal can be decomposed into **an infinite sum of sinusoidal waveforms**

$$f(x) = \sum_{u=0}^{\infty} F(u)\cos(uwx)$$

  - *nwx* is the frequency component of the sinusoidal wave
  - *F(u)* is the coefficient (weight) of a wave, cos(nwx)
- Why useful?
  - Most of natural signals consists of only a **few dominant** frequency components
  - Due to this **sparsity,** it is easier to compress the signals after transformation
  - **Dropping weak components** → **distortion is small** and hardly be detected by human eyes

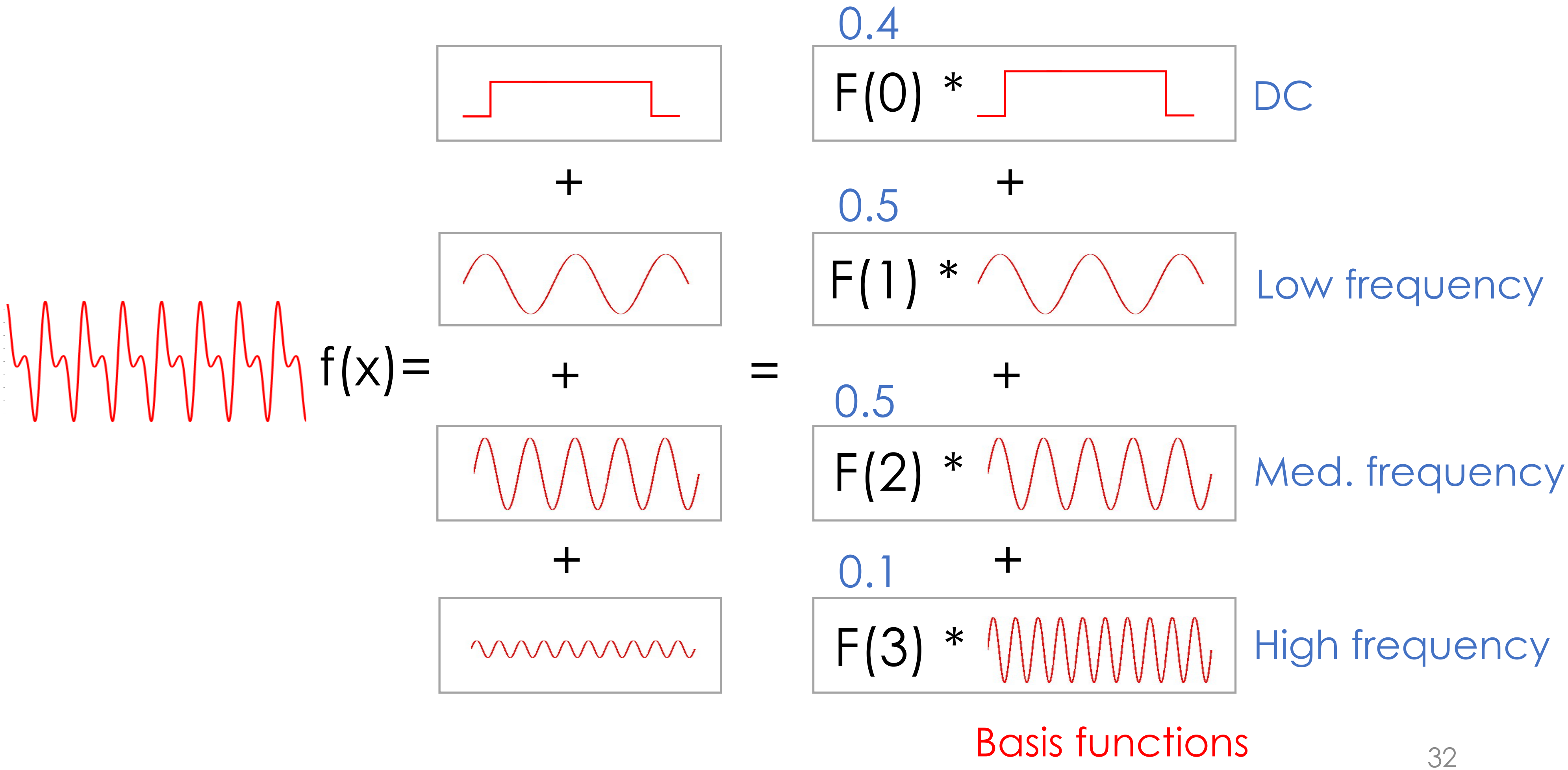
FFT Example



http://68.media.tumblr.com/8ab71becbff0e242d0bf8d
b5b57438ab/tumblr_mio8mkwT1i1s5nl47o1_500.gif

Fourier Analysis – Example

- Coefficient F(n) is the **amplitude** of its corresponding frequency component



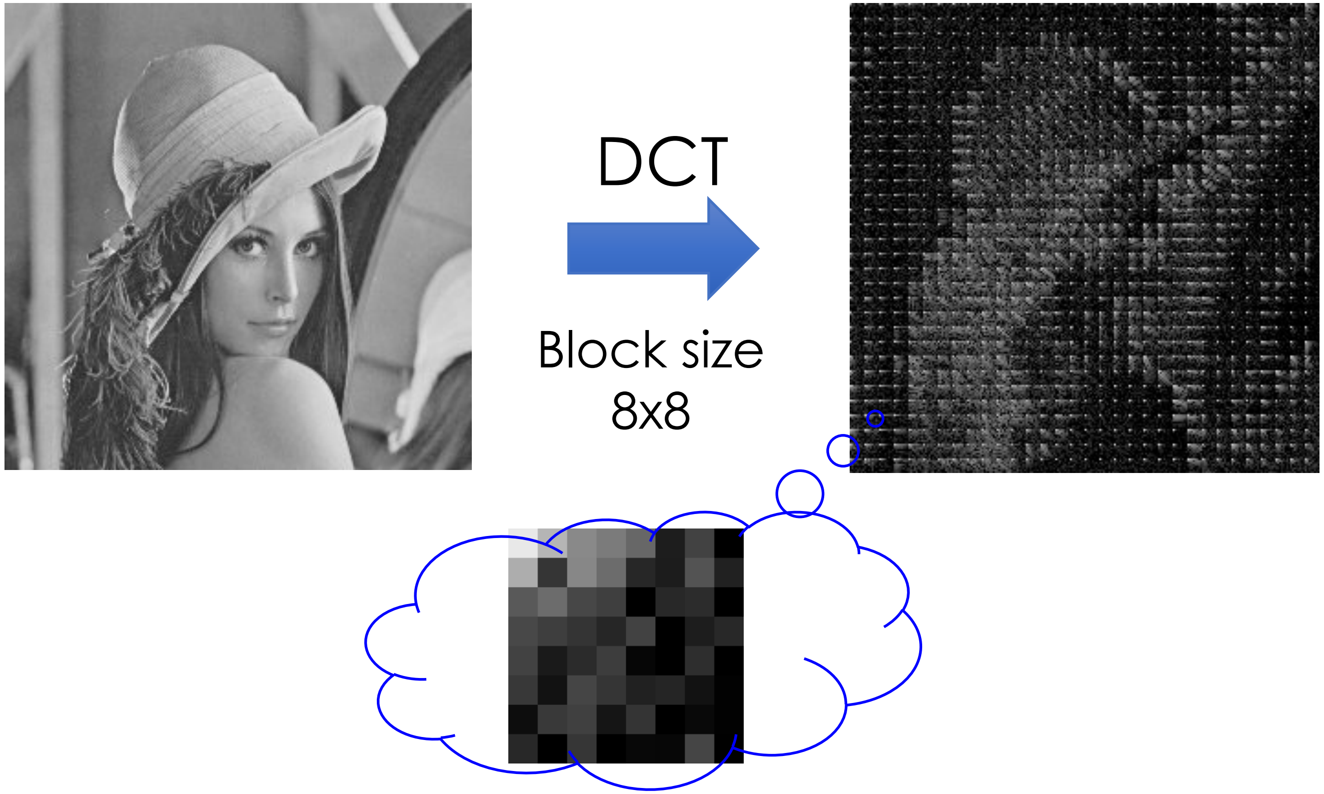
Transformation Technologies

- **Discrete cosine transform (DCT)**
  - Usually applied to small blocks
  - JPEG, H26x, MPEG-x

- Discrete wavelet transform (DWT)
  - Usually applied to large images
  - JPEG 2000, MPEG-4 still texture

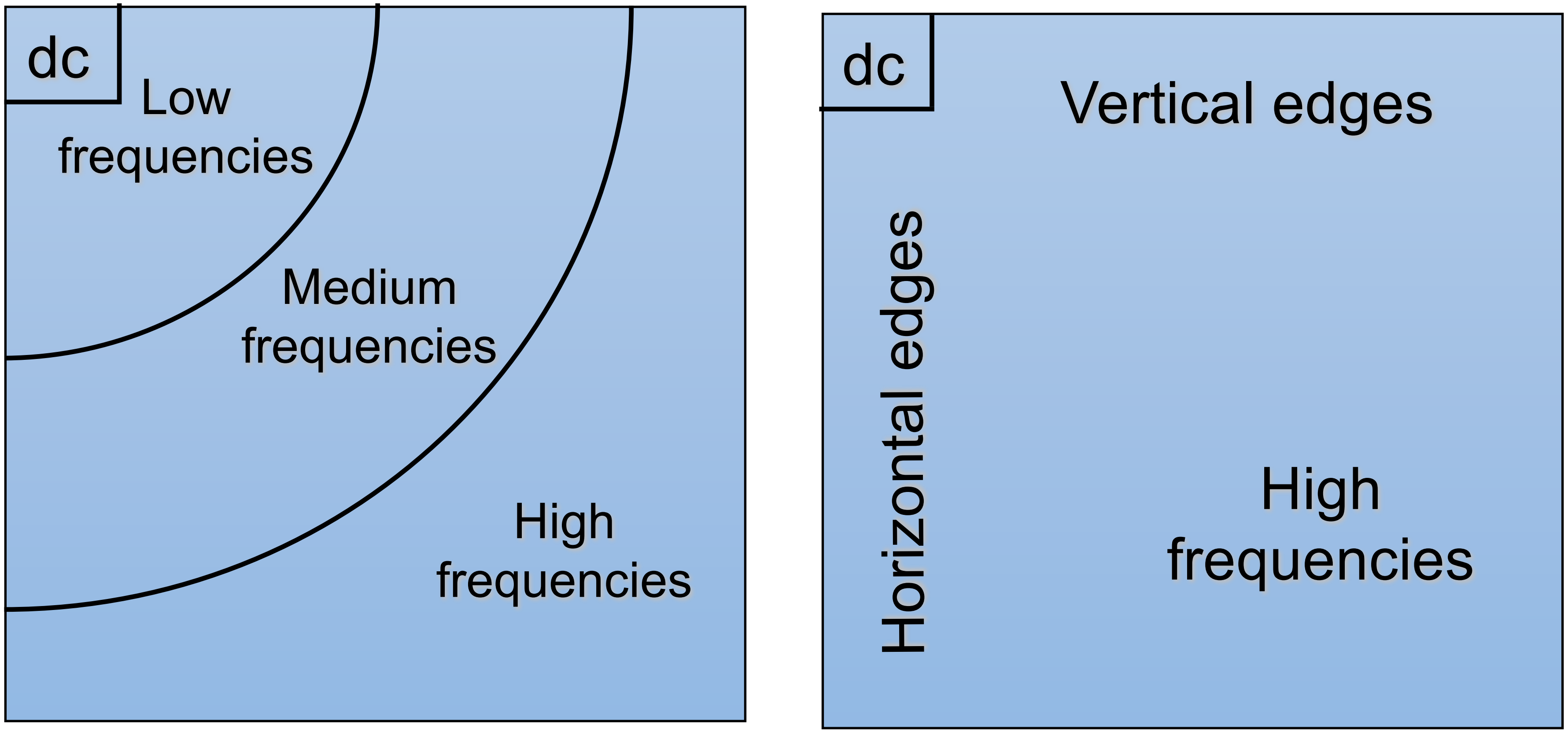
Discrete Cosine Transform (DCT)

- Image have discrete points → DCT
  - Idea is similar to Fourier analysis



- Usually only the DC component (left-top) has a large amplitude
- High-frequency AC components are close to zero

2D-DCT Coefficients



Discrete Cosine Transform (DCT)

- Original signals is the linear combination of DC and different ACs basis functions
  - (DC: frequency = 0, AC_i: frequency $i\pi$)

Time domain DCT **Frequency domain**

f(i) $\Longrightarrow$ F(u)

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

$$, u = 0, 1, \cdots, N-1$$

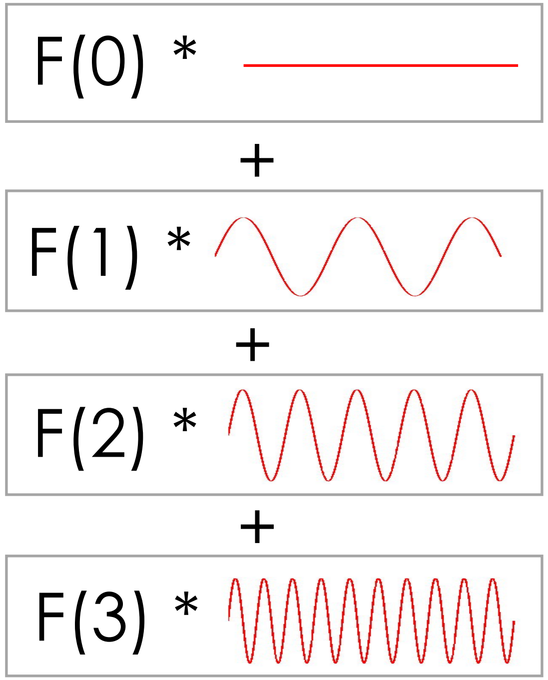
Time domain IDCT **Frequency domain**

F(u) $\Longrightarrow$ f(i)

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

$$, x = 0, 1, \cdots, N-1$$

f(x)=



Basis function

$$C(0) = \sqrt{\frac{1}{N}},$$

$$C(u) = 1, \forall u \neq 0$$

2D-DCT

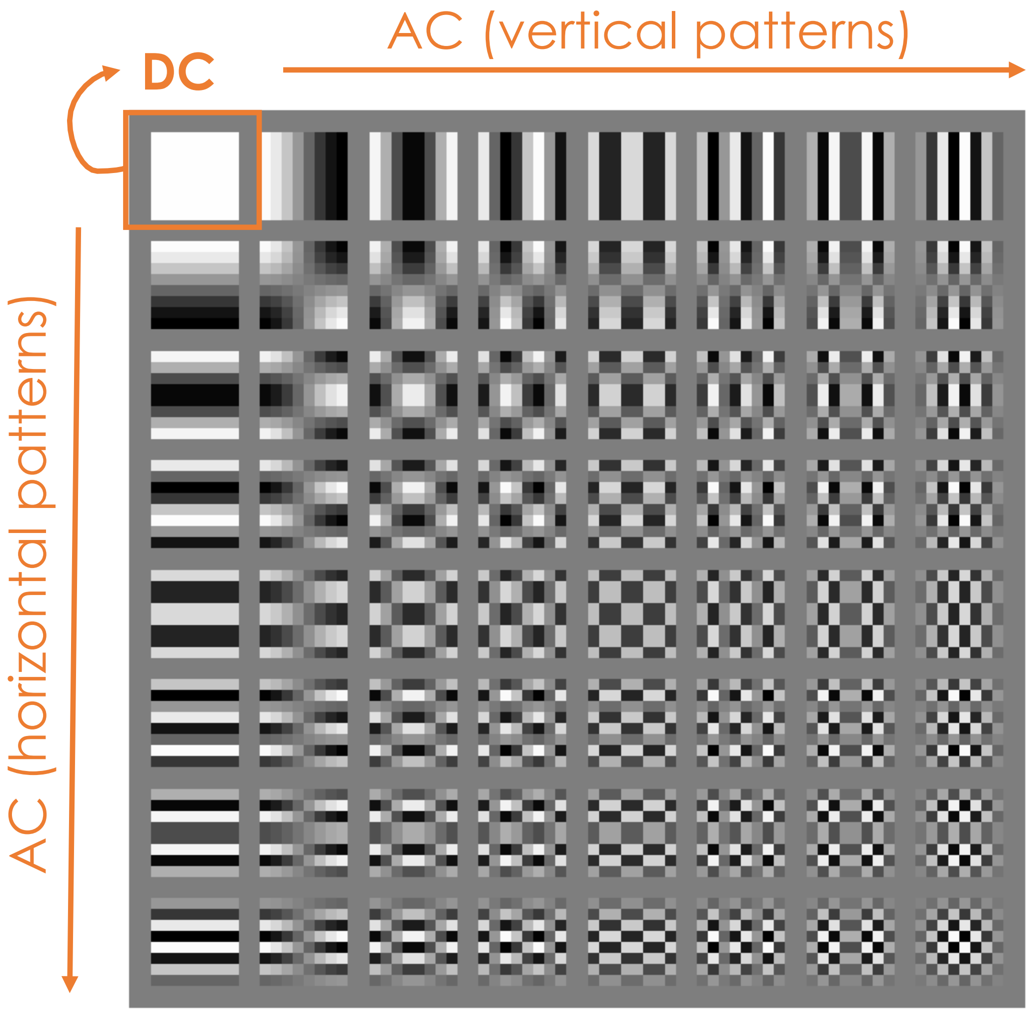
- Two-dimensional DCT
- Represent each block of image pixels as a weighted sum of 2D cosine functions (basis functions)

$$F(u,v) = \frac{2}{N} C(u) C(v) \sum_{i=0}^{N-1} f(x,y) \cos \frac{2\pi(2x+1)u}{4N} \cos \frac{2\pi(2y+1)v}{4N}$$

$$C(u), C(v) = \begin{cases} \sqrt{1/2}, u, v = 0 \\ 1, \text{otherwise} \end{cases}$$

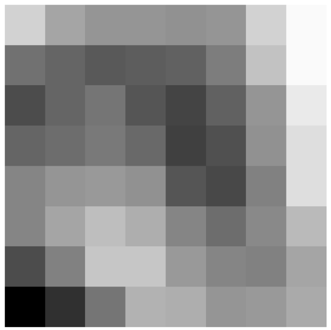
- Block size: N x N
- f(x,y): pixel value
- F(u,v): DCT coefficients

2D 8x8 DCT Basis Functions



An Example of Energy Distribution

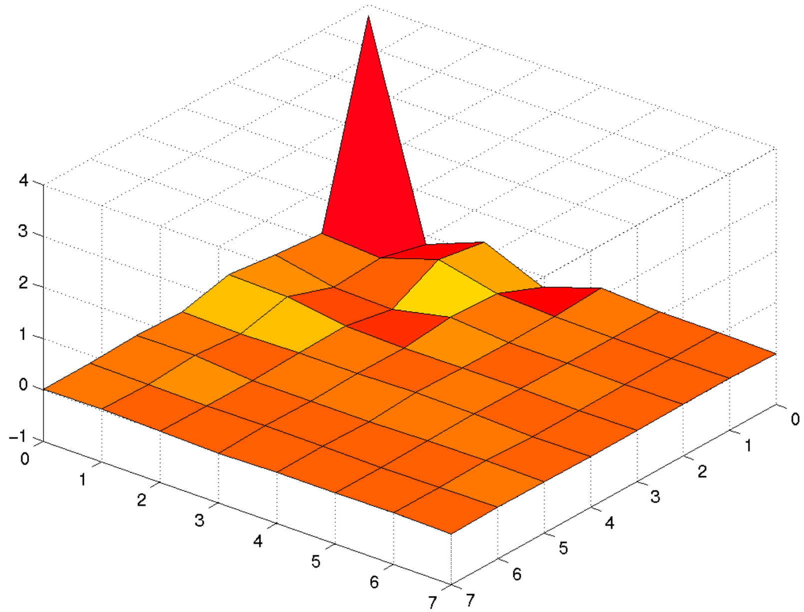
- In frequency domain, bias distribution → small entropy → need only a few bits for compression
- Peak in the DC component
- Nearly 0 in high frequency components
  - Compress them!



time-domain pixel values

DCT

⟹



frequency-domain response

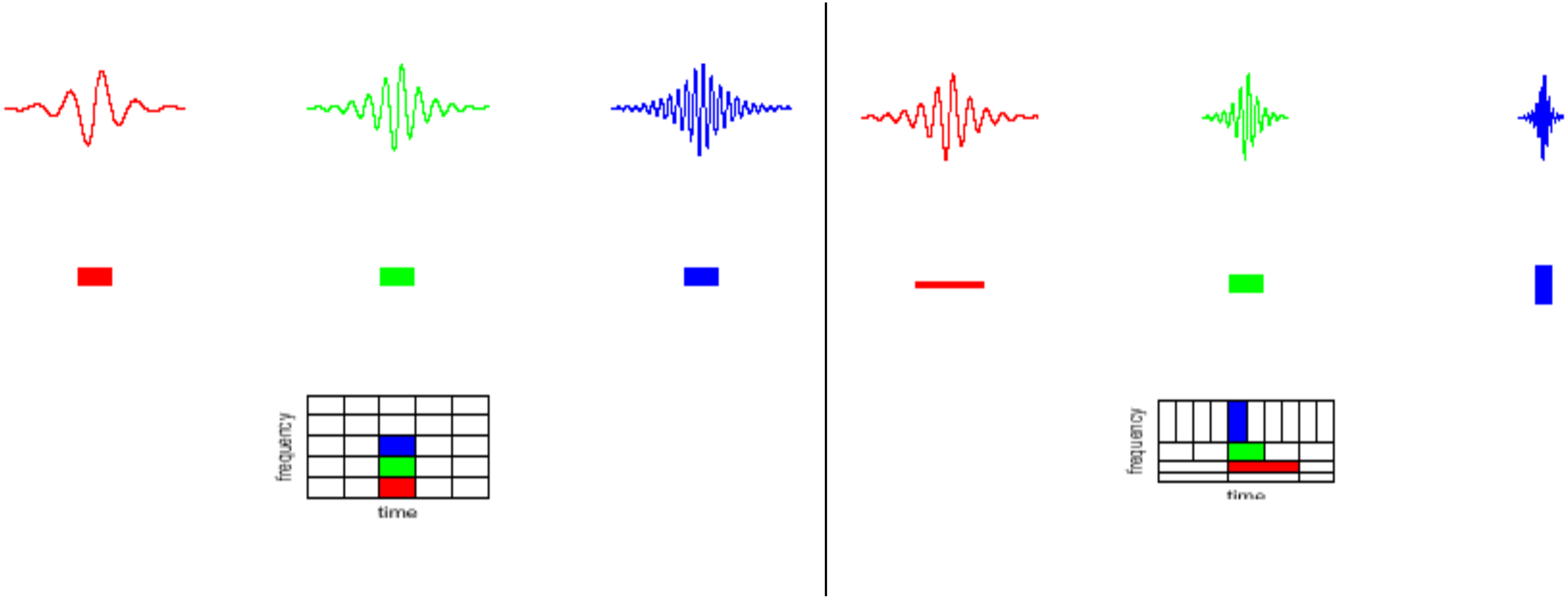
Transformation Technologies

- Discrete cosine transform (DCT)
  - Usually applied to small blocks
  - JPEG, H26x, MPEG-x

- **Discrete wavelet transform (DWT)**
  - Usually applied to large images
  - JPEG 2000, MPEG-4 still texture

Wavelet Transformation Concept

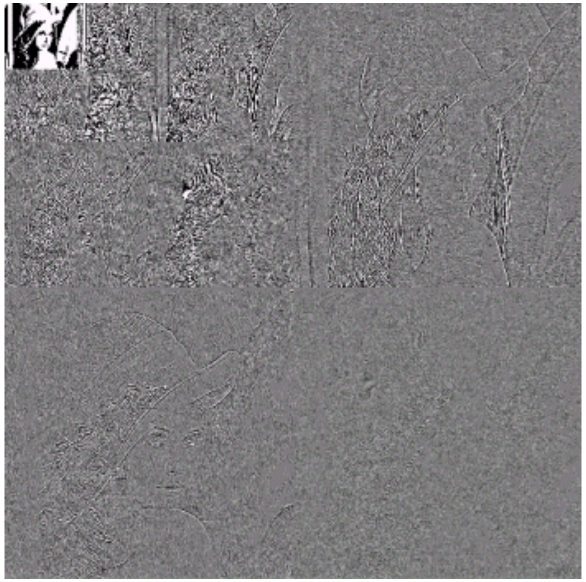
- Consider both variable frequency and time resolutions



- DCT (or DFT) only has the same time and frequency resolution

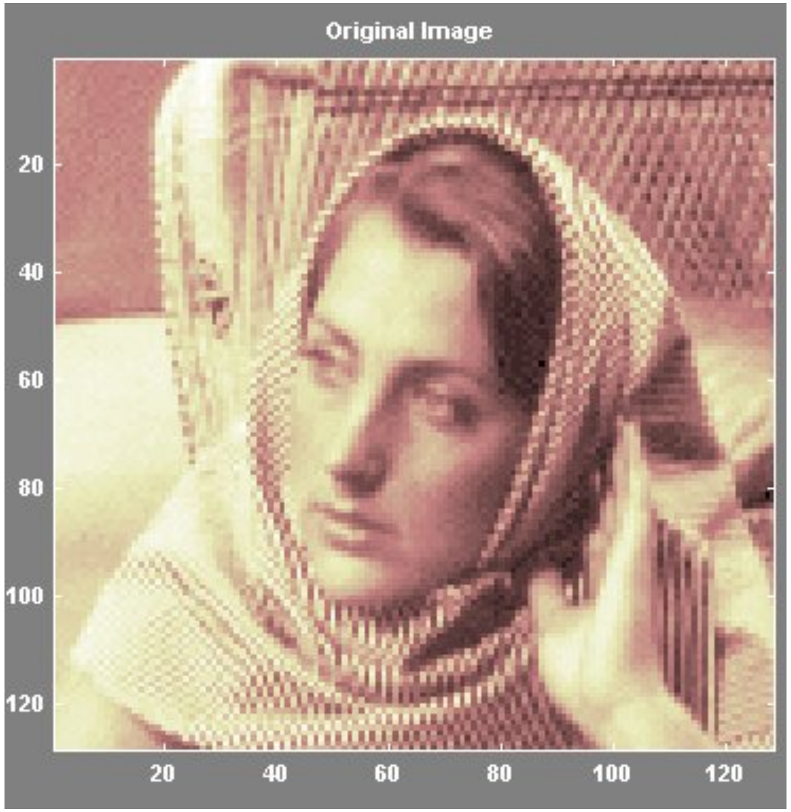
Wavelet Transformation Concept

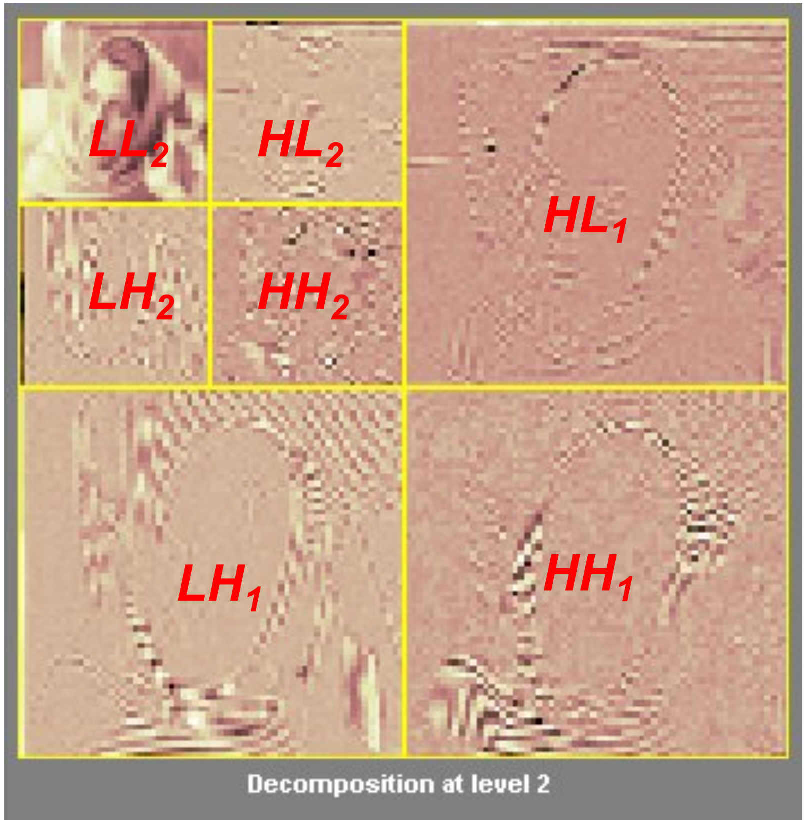
- Dynamic decomposition
  - Lower frequency subbands have finer frequency resolution and coarser time resolution
  - Suitable for natural images



2D-DWT

- High-level idea:
  - LL block: subsampled values (taking average)
  - Others: difference



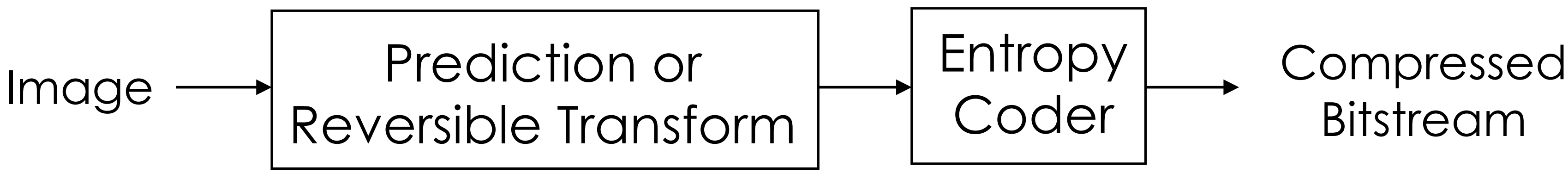


Outline

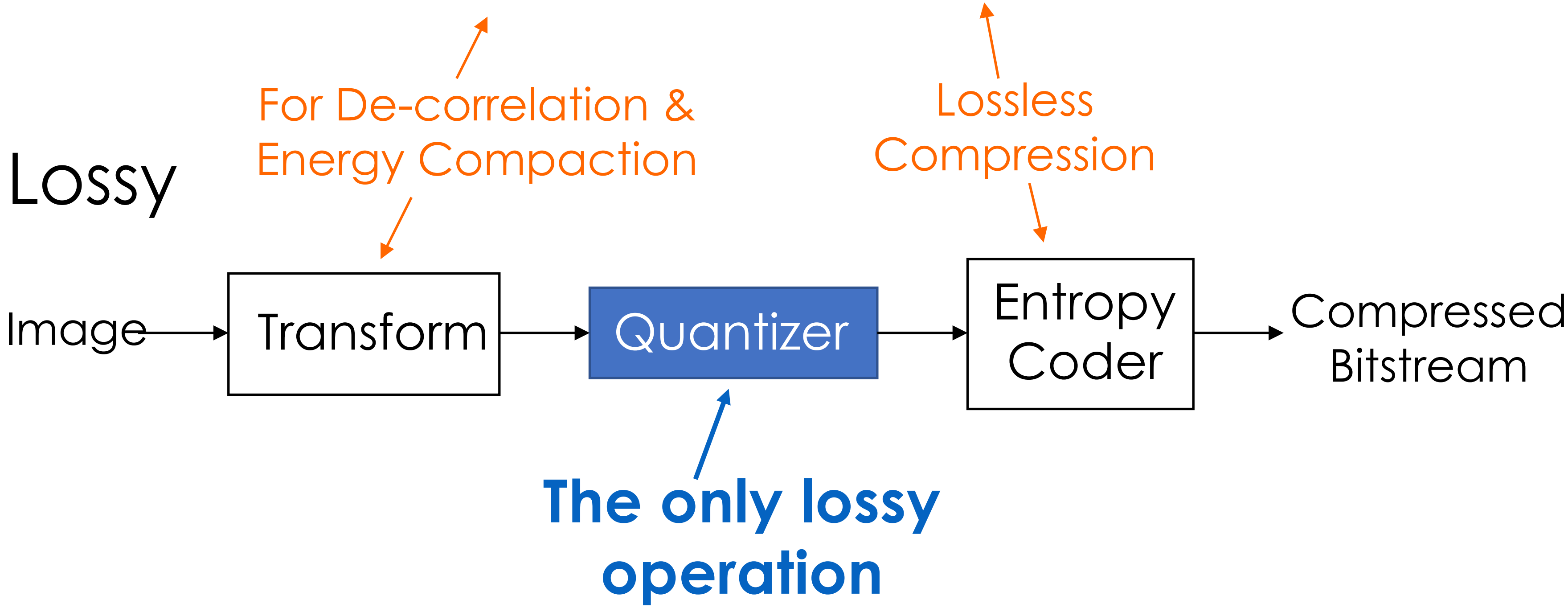
- Concepts of data compression
- Lossless Compression
- Lossy Compression
- **Quantization**

Image Compression Framework

- Lossless:



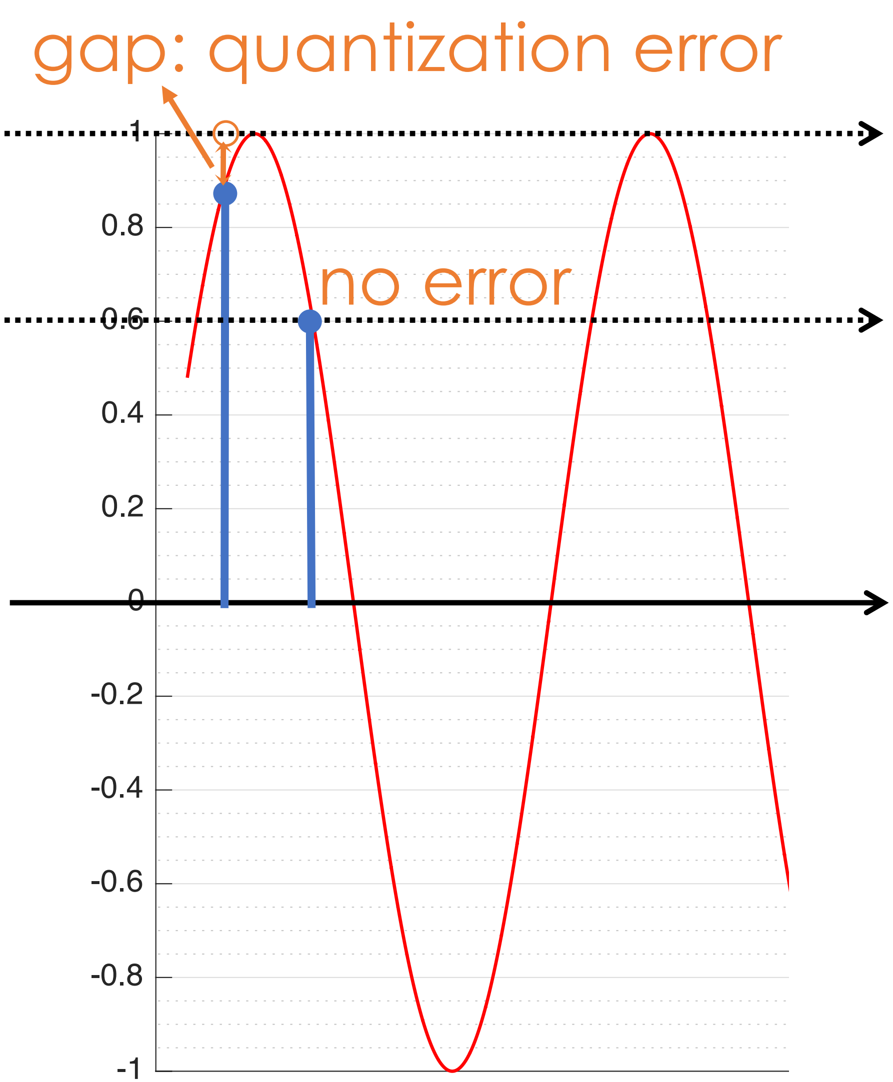
- Lossy



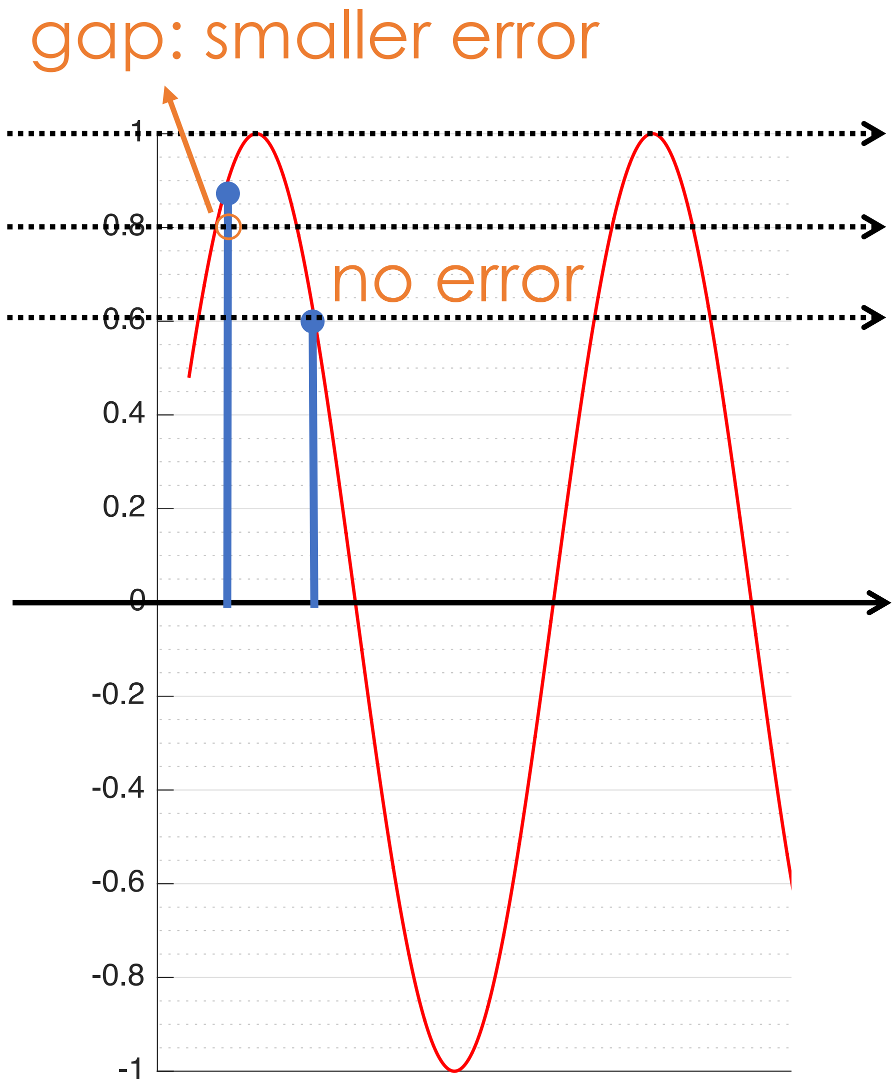
Quantization

- Why need quantization?
  - *Digital image*
  - n bits to represent a digital sample → max number of different levels is $m = 2^n$
- Real values are rounded to the nearest level
- Some information (precision) could be lost
  - Hopefully, we want those non-important parts are lost
- Usually the only **lossy** operation that removes perceptual irrelevancy
- Cannot be recovered
  - More bits → Smaller quantization errors

Quantization Error

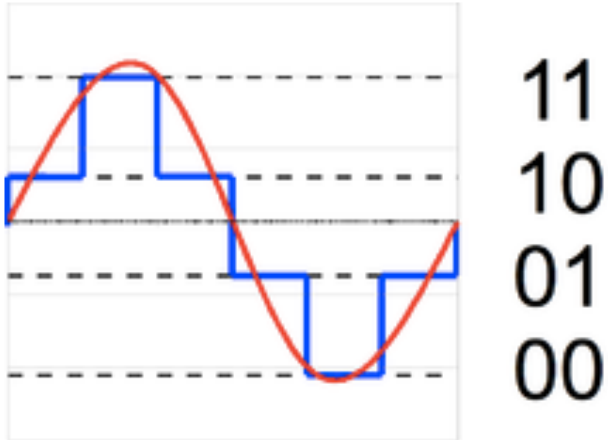


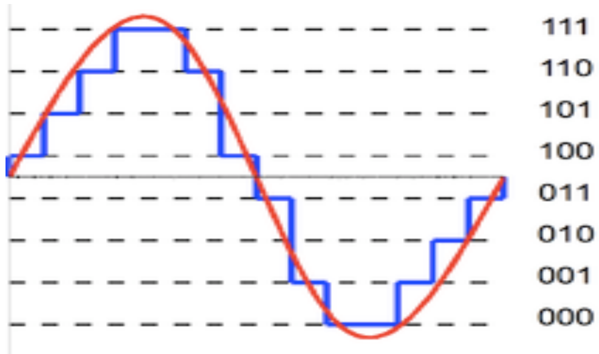
Coarser resolution



Finer resolution

Quantization Resolution





Four levels
Larger errors

Eight levels
Smaller errors

Source: wikipedia

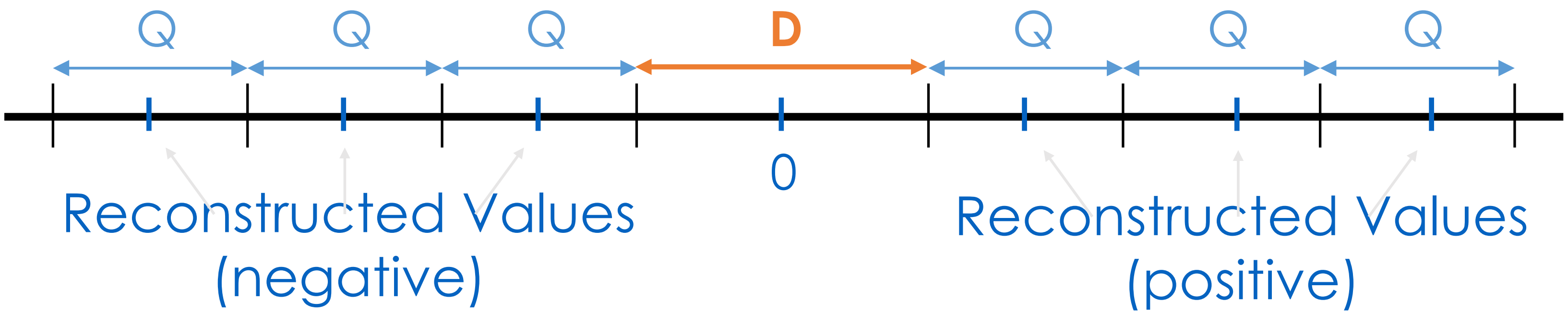
Quantization Methods

- Quantization distribution
  - Uniform quantization
  - Optimized non-uniform quantization

- Number of dimensions
  - Scalar quantization (one-dimensional input)
  - Vector quantization (multi-dimensional input)
    - Partition a vector into groups, each of which is represented by the same number of points (e.g., k-means clustering)

Uniform Quantization with Dead-Zone

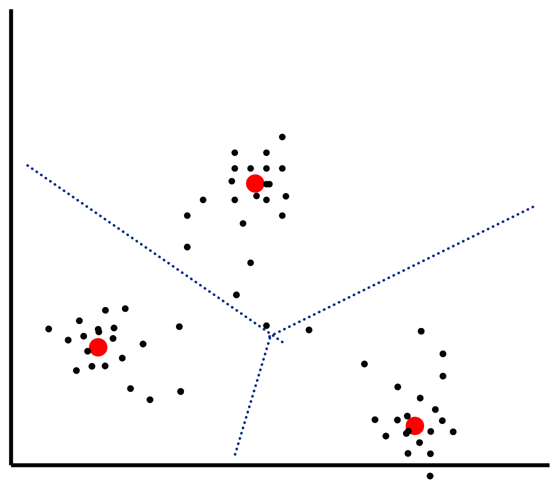
- Q: quantizer step size
- D: dead-zone size



- D is usually greater than or equal to Q
  - Larger D increases the number of signal values quantized to 0
  - The 0 index usually needs fewest bits to encode → adjusting D to get better rate-distortion optimization
- If D = Q → degenerate to normal uniform quantization

Optimized Non-uniform Quantization

- Optimization Criterion: **Expectation of Distortion**
  - Sometimes with a constraint of maximal bit-rate (very complex)
- Designed according to a mathematical model or real (training) data
- Example of quantizer from training data



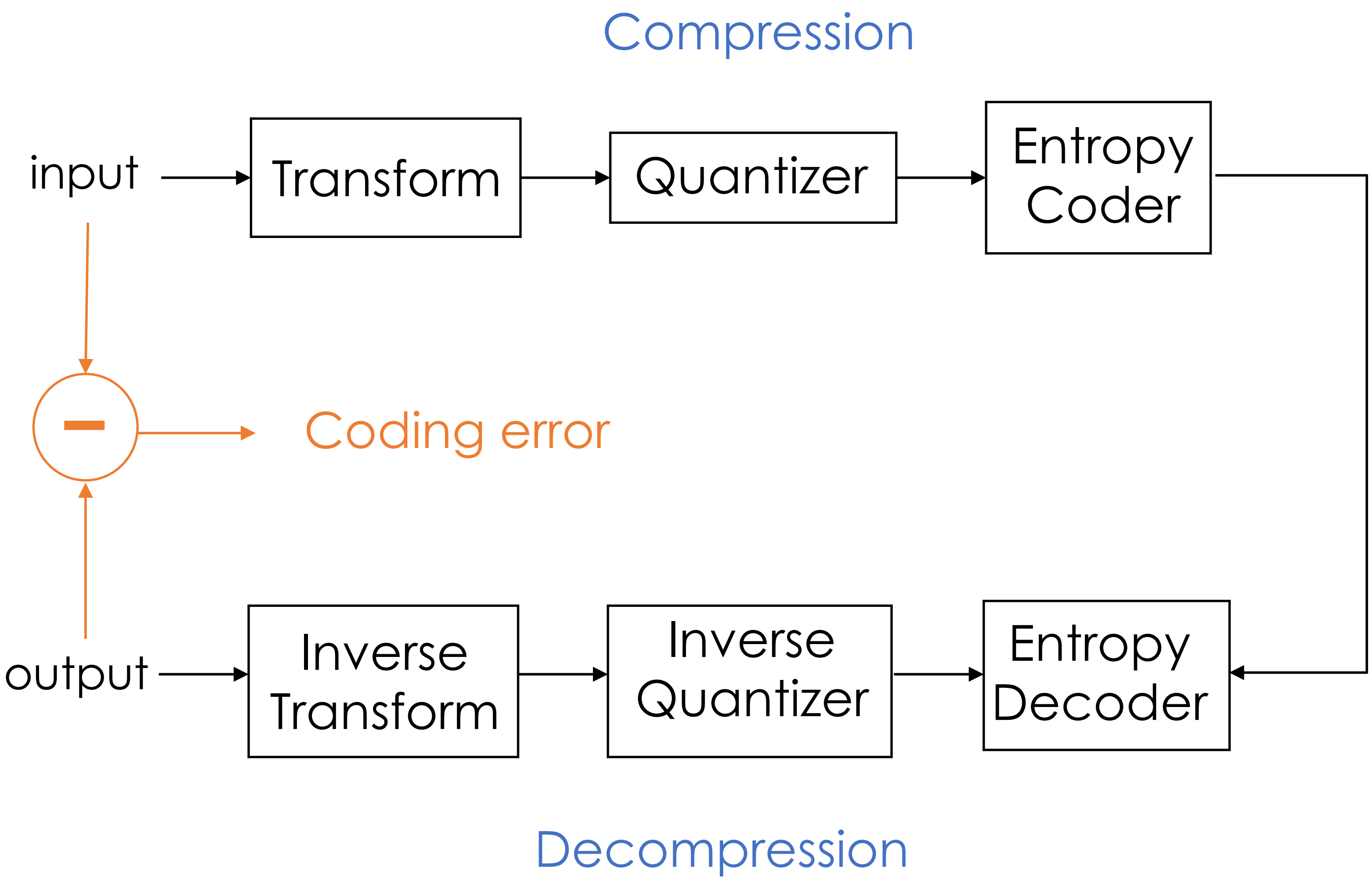
● : Reconstructed value point, usually the centroid

SQNR

- Signal-to-Quantization-Noise Ratio

$$\frac{\text{maximum amplitude } 2^{n-1}}{\text{maximum quantization noise, e.g., } 1/2}$$

$$\Rightarrow 20\log_{10}\left(\frac{2^{n-1}}{1/2}\right)$$


Compression
input
Transform
Quantizer
Entropy Coder
Coding error
output
Inverse Transform
Inverse Quantizer
Entropy Decoder
Decompression

Summary

- **Concepts of data compression**
  - Trade off between rate and distortion
  - Statistical and perceptual redundancy
- **Lossless Compression**
  - Entropy coding
  - Bounded by the information amount of a source
- **Lossy Compression**
  - Transform coding
  - Coding in the domain with a higher entropy
  - Easier to compact the insignificant copmonents
- **Quantization**
  - Unavoidable losses in digital signals