CS120: Computer Networks

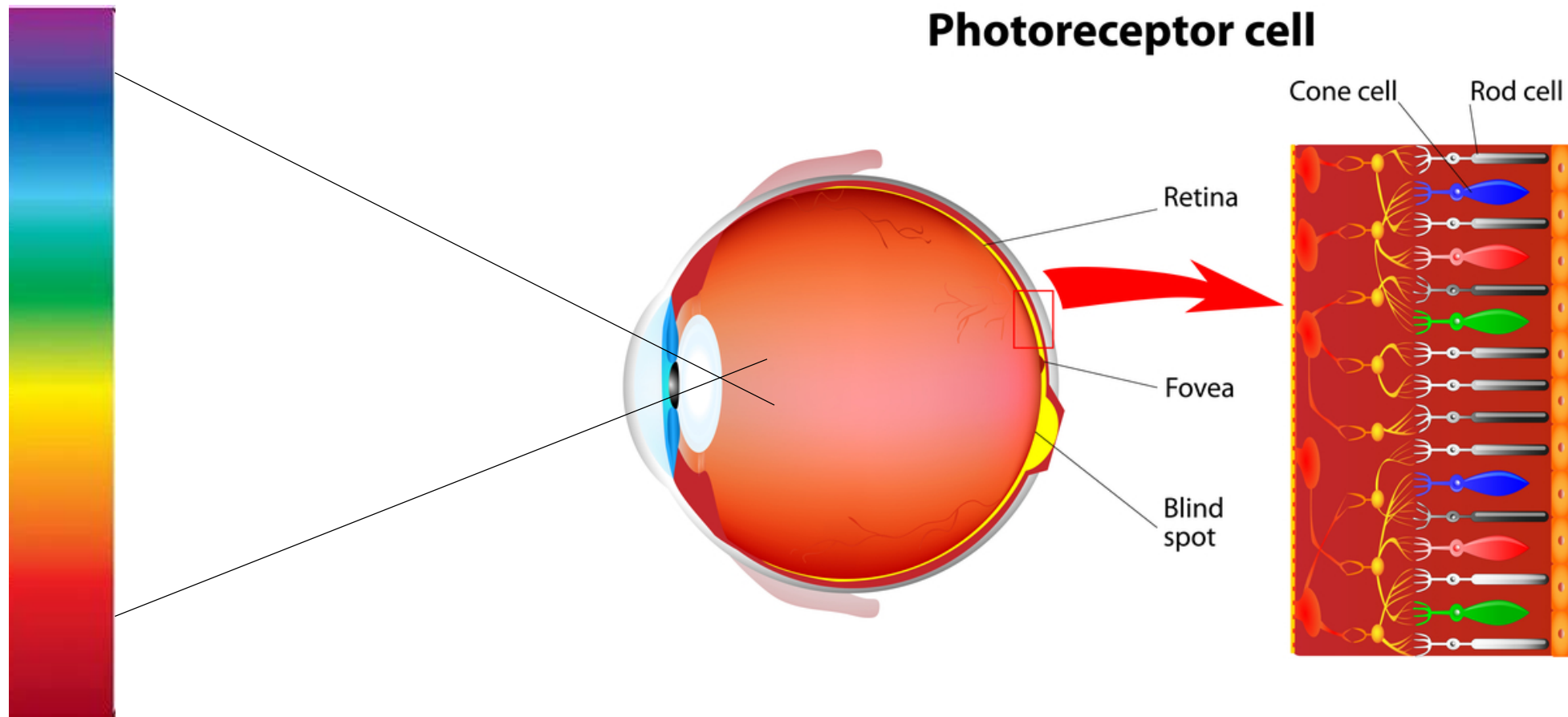**Lecture 23. End-to-End Data 2**

Zhice Yang

End-to-End Data

- Data Presentation
- Data Compression
    - Lossless Compression
    - ➢Multimedia Compression
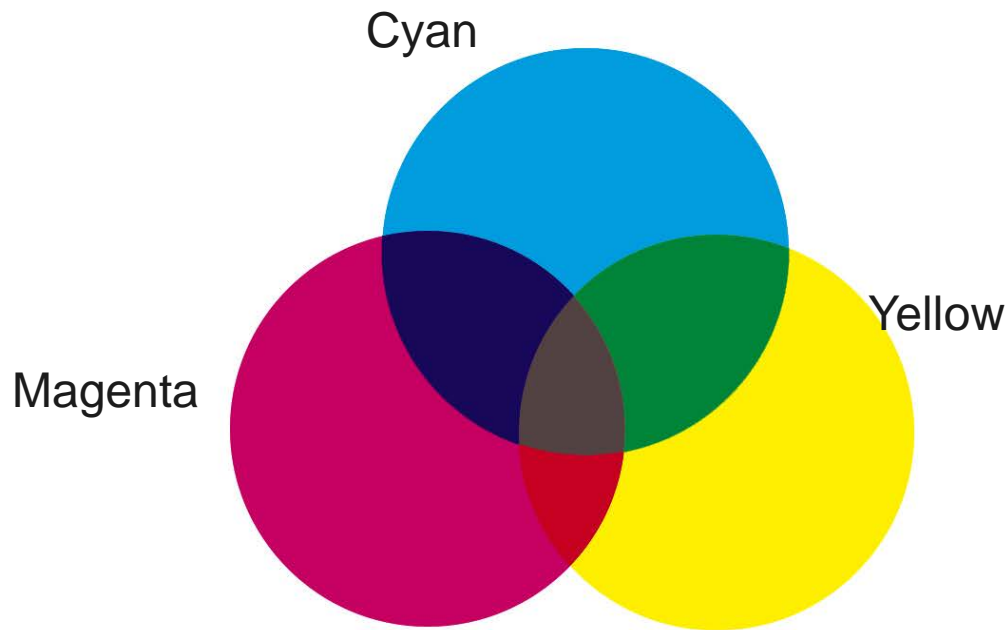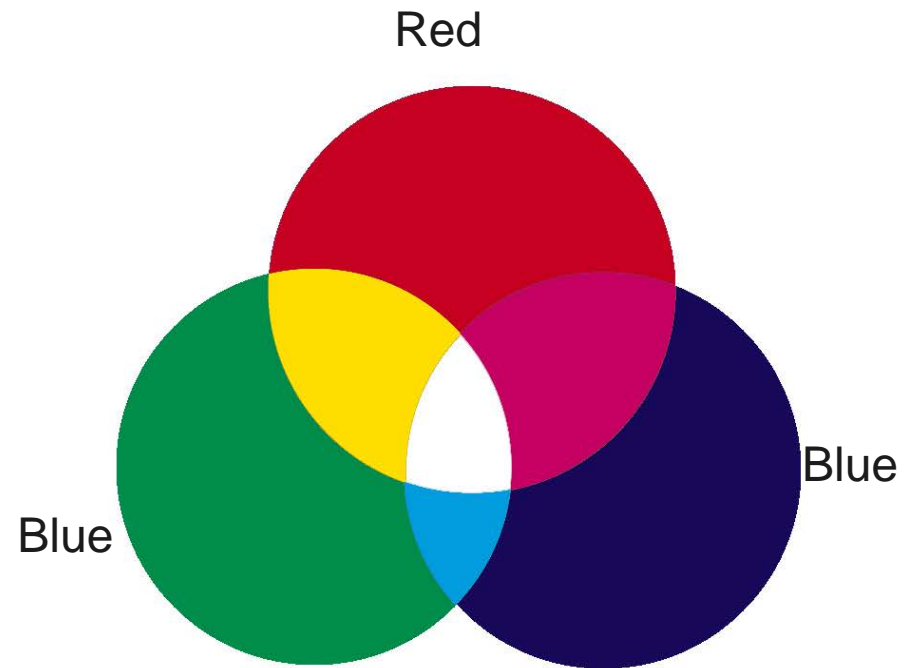
Color and Display

- eyes spectrum

Photoreceptor cell

Cone cell

Rod cell

Retina

Fovea

Blind spot

Color and Display

- Color Model

Cyan

Magenta

Yellow

Subtractive color (CMYK)
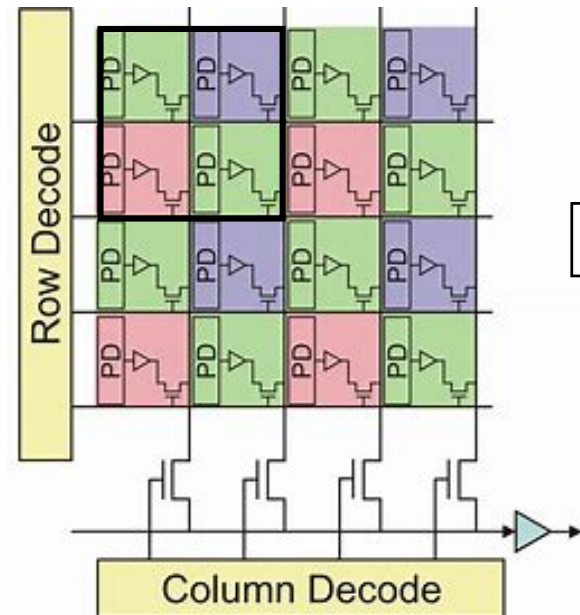
for printing

Red

Blue

Blue

Additive Color (RGB)

for display

Color and Display

- Imaging and Display

RGBs

RGBs

CMOS

Display

5

Color and Display

- Digital Image
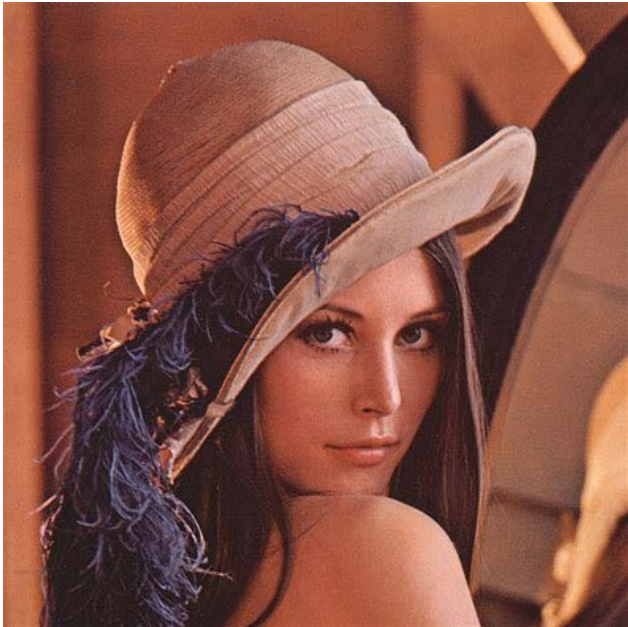
$$= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a & a & \cdots & a \\ \vdots & \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & & & \\ \vdots & \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}_{n \text{ x } m} \end{pmatrix} \end{pmatrix}$$

Size = 1080*1920*8*3 = 50Mb !

6

GIF – Image Compression

- Filename Extension: `.gif`
- Simple Lossy Compression
- 3*8 bit => 256 colors

JPEG – Image Compression

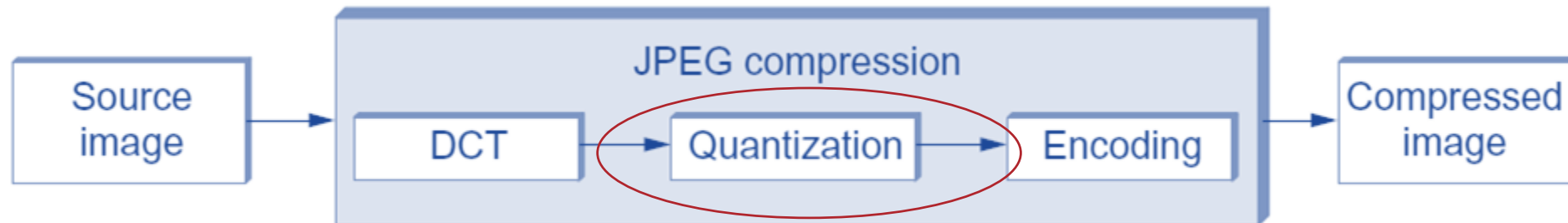- Filename Extension: `.jpg, .jpeg`
- Joint Photographic Experts Group
- Intuition
  - Human eyes are sensitive to intensity change, but less sensitive to chromatic changes
  - Human eyes are sensitive to low frequency change, but less sensitive to high frequency changes

JPEG Compression Flow

Source image → **JPEG compression**: DCT → Quantization → Encoding → Compressed image

Information Loss

JPEG Compression: Splitting
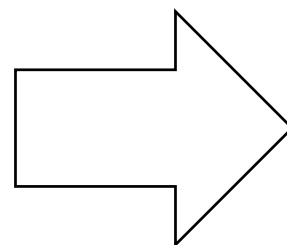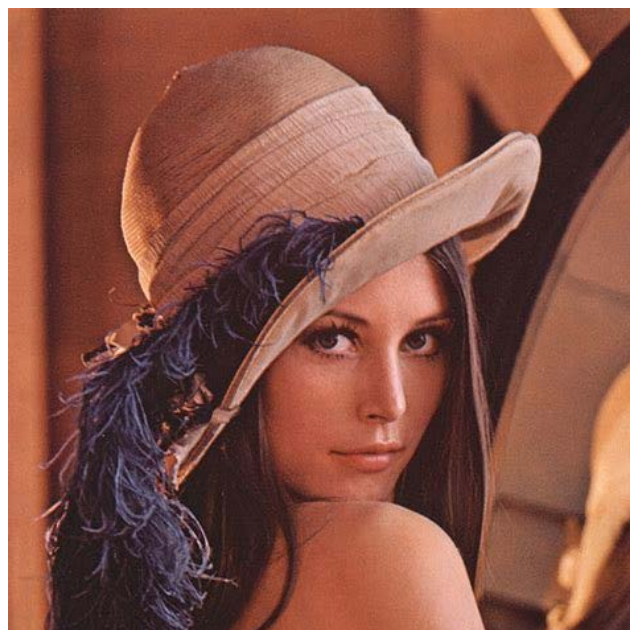
8 pixels * 8 pixels

JPEG Compression: RGB -> YUV

- YUV Space
  - Y -> luminance
    - Sensitive
  - U, V -> chrominance
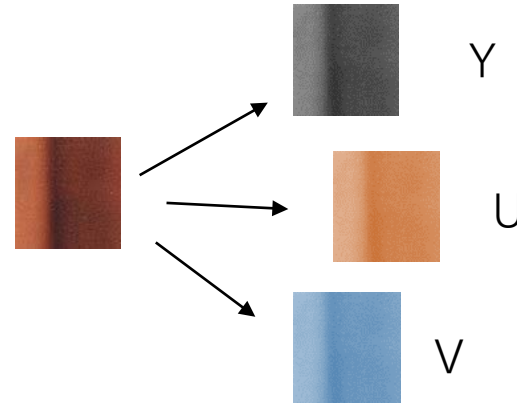
$$Y = 0.299R + 0.587G + 0.114B$$

$$U = (B - Y) \times 0.565$$

$$V = (R - Y) \times 0.713$$

JPEG Compression: Subsampling UV

- YUV Space
  - Y -> luminance
    - Sensitive
  - U, V -> chrominance
    - Less sensitive
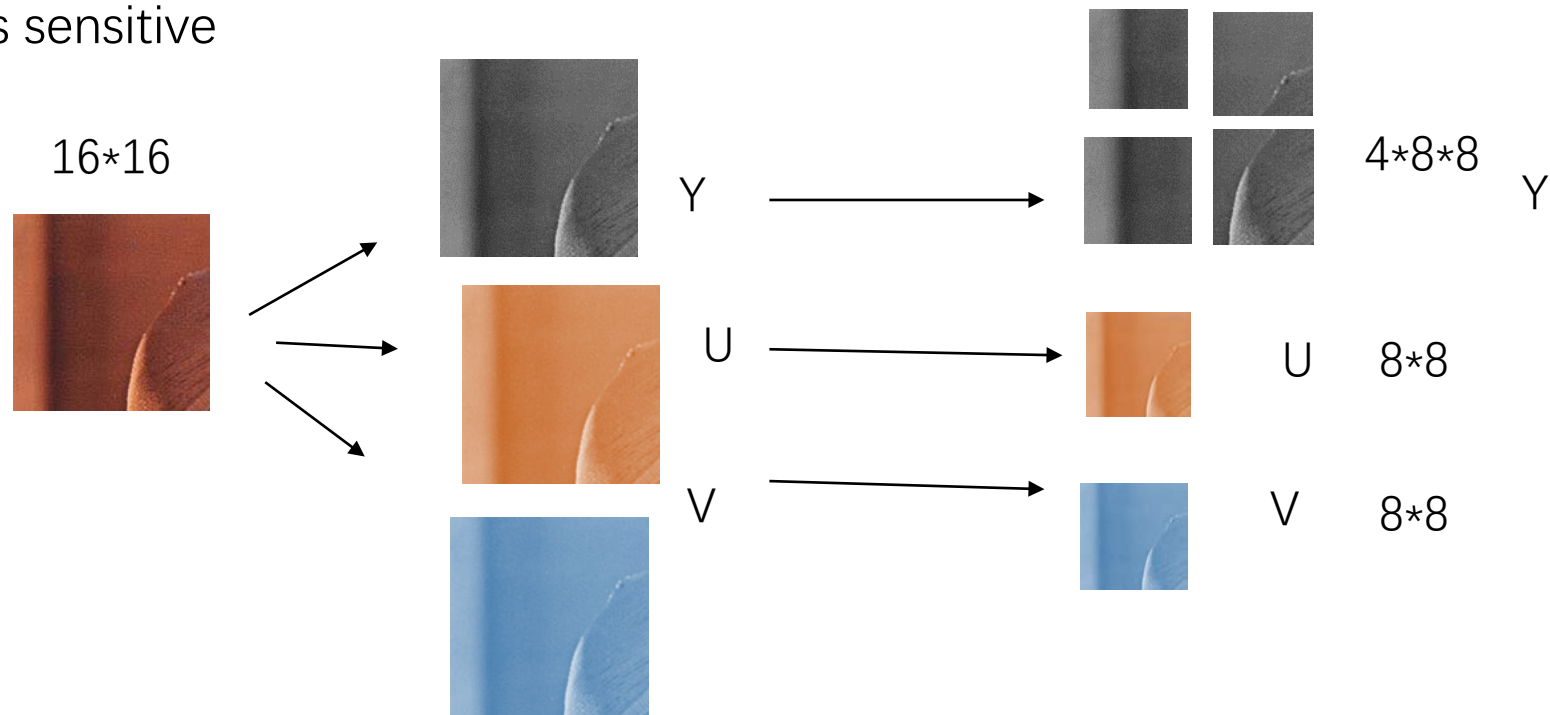
16*16

Y

U

V

4*8*8   Y

U   8*8

V   8*8

JPEG Compression: DCT

- Discrete Cosine Transform

FFT Spectrum ? → Complex Number

JPEG Compression: DCT

- Discrete Cosine Transform

FFT Spectrum

Real Number
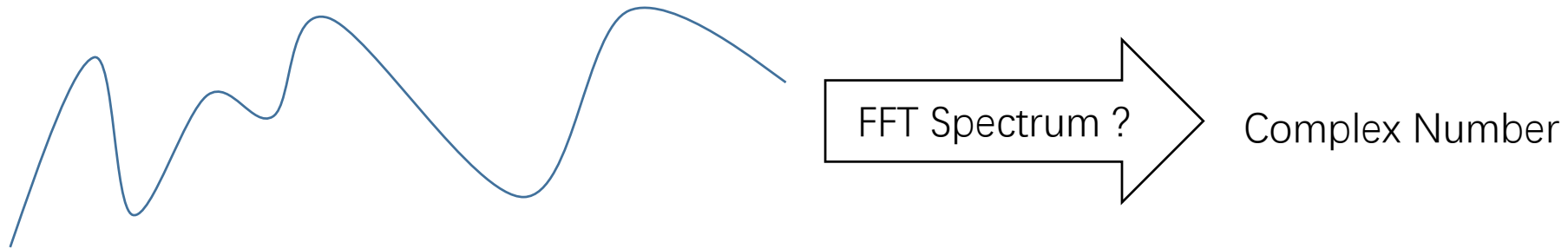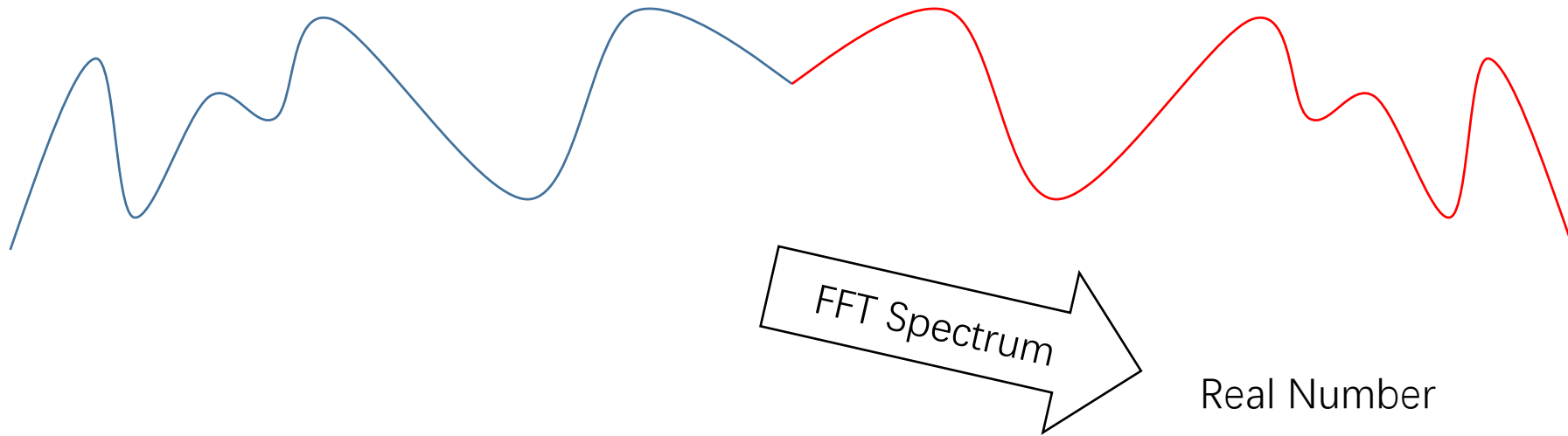
JPEG Compression: DCT

DCT
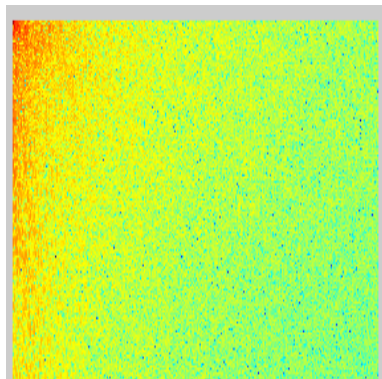
Sensitive to Low frequency
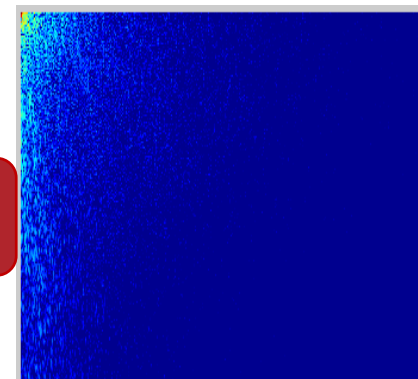
DCT

JPEG Compression: Quantization
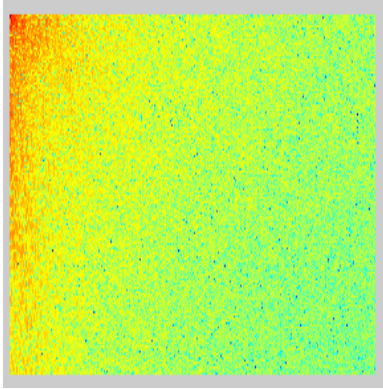
Round (DCT(i,j)/Quantum(i,j))
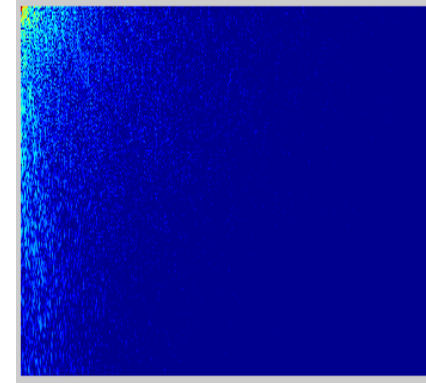
$$Quantum = \begin{bmatrix} 3 & 5 & 7 & 9 & 11 & 13 & 15 & 17 \\ 5 & 7 & 9 & 11 & 13 & 15 & 17 & 19 \\ 7 & 9 & 11 & 13 & 15 & 17 & 19 & 21 \\ 9 & 11 & 13 & 15 & 17 & 19 & 21 & 23 \\ 11 & 13 & 15 & 17 & 19 & 21 & 23 & 25 \\ 13 & 15 & 17 & 19 & 21 & 23 & 25 & 27 \\ 15 & 17 & 19 & 21 & 23 & 25 & 27 & 29 \\ 17 & 19 & 21 & 23 & 25 & 27 & 29 & 31 \end{bmatrix}$$

Determine the How Much Information is dropped

JPEG Compression: Quantization

- Quantization is a lossy process
  - Recovered DCT(i,j) = QuantizedValue(i,j)*Quantum(i,j)
  - Rounding in Quantization is lossy

JPEG Compression: Zig-Zag

8*8

1*64

...
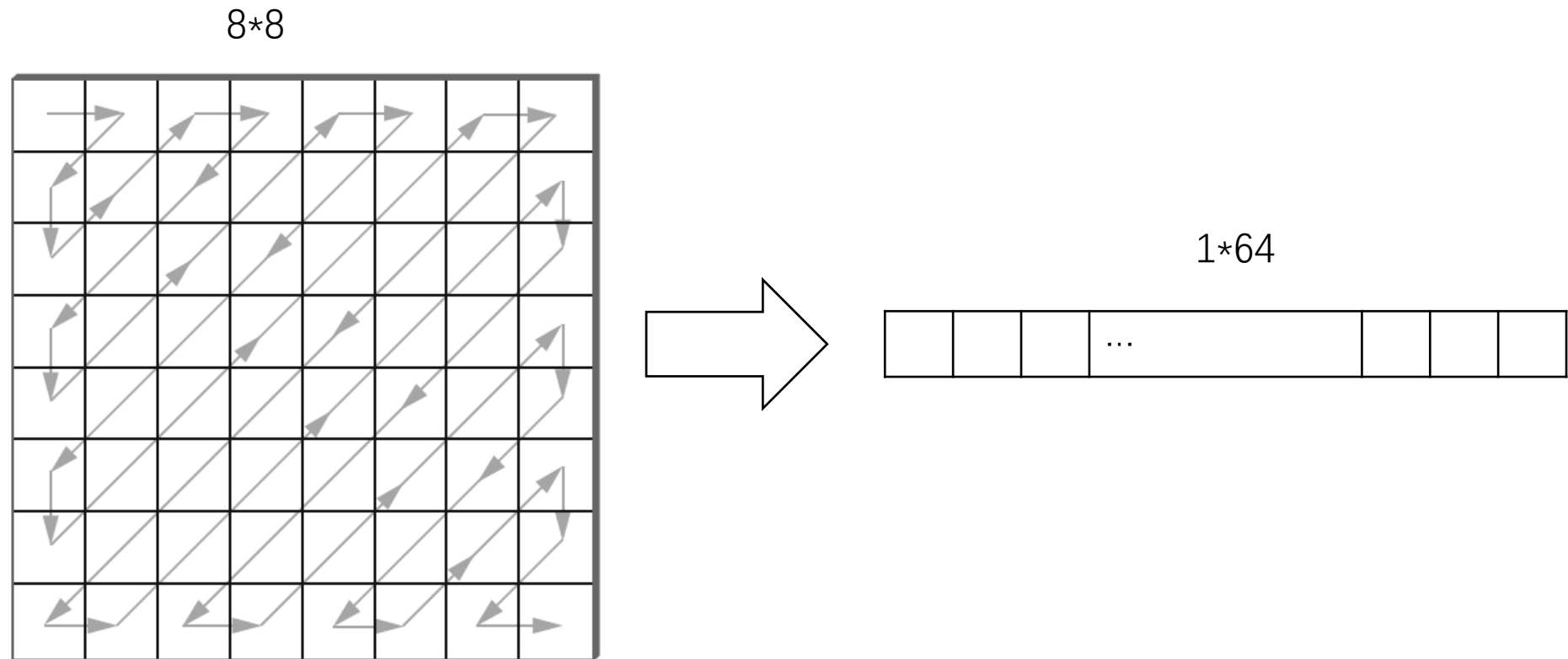
JPEG Compression: DC Component

- DC Components are large and normally non-zero
- Nearby DC Components are closed
- Differential Pulse Code Modulation (DPCM)
  - 14, 12, 13, 12, 15 => 14, -2, 1, -1, 3

JPEG Compression: DC Component

- DC Component can be expressed in integer
  - eg. in ones complement
    - 3 => 0011
    - -3 => 0000
    - 4 => 100
    - -4 => 011
- Problem
  - If expressing integer in fix-length bits
    - padding zeros waste space
  - If expressing integer in dynamic length bits
    - how to split the bit stream ?

JPEG Compression

- DC Component can be expressed as (size, amplitude)
  - Size: number of bits to express amplitude
  - Amplitude: DPCM value in ones complement
  - Examples:
    - 0 => (0,-)
    - 1 => (1,1)
    - -1 => (1,0) bitwise inverse for negative value
    - 2 => (2, 10)
    - -2 => (2, 01)
    - 3 => (2,11)
    - -3 => (2,00)

JPEG Compression: Huffman Coding

- DC Component can be expressed as (size, amplitude)
  - Size: number of bits to express amplitude, Huffman coded
    - The coding table is include in the JPEG file
  - Amplitude: DPCM value in ones complement
  - Examples:
    - 0 => (0,-) => 0
    - 1 => (1,1) => 101 1
    - -1 => (1,0) => 101 0
    - 2 => (2, 10) => 011 10
    - -2 => (2, 01) => 011 01
    - 3 => (2,11) => 011 11
    - -3 => (2,00) => 011 00

Length	Code	Size
3 bits	000	04
	001	05
	010	03
	011	02
	100	06
	101	01
	110	00 (End of Block)
4 bits	1110	07
5 bits	1111 0	08
6 bits	1111 10	09
7 bits	1111 110	0A
8 bits	1111 1110	0B

table ref: https://www.impulseadventure.com/photo/jpeg-huffman-coding.html

JPEG Compression: AC Component

- AC Components are small and normally zero
- ➤ Run Length Encoding (RLE)
  - 0000020000100000210000 => (5,2)(4,1)(4,2)(0,1)(0,0)

JPEG Compression: Huffman Coding

- AC Component (length, value) can be expressed as (length, size, amplitude)
  - Amplitude: AC value in ones complement
  - Size: number of bits to express amplitude, Huffman coded
  - Length: number of skipped zeros
- Size and length are expressed in 8 bits together, Huffman coded
  - length/size
- Final result: <length/size | amplitude>
  - (5,2)(4,1)(4,2)
  - =>(5/2,2)(4/1,1)(4/2,2)
  - => 111110 10 010 1 101 10

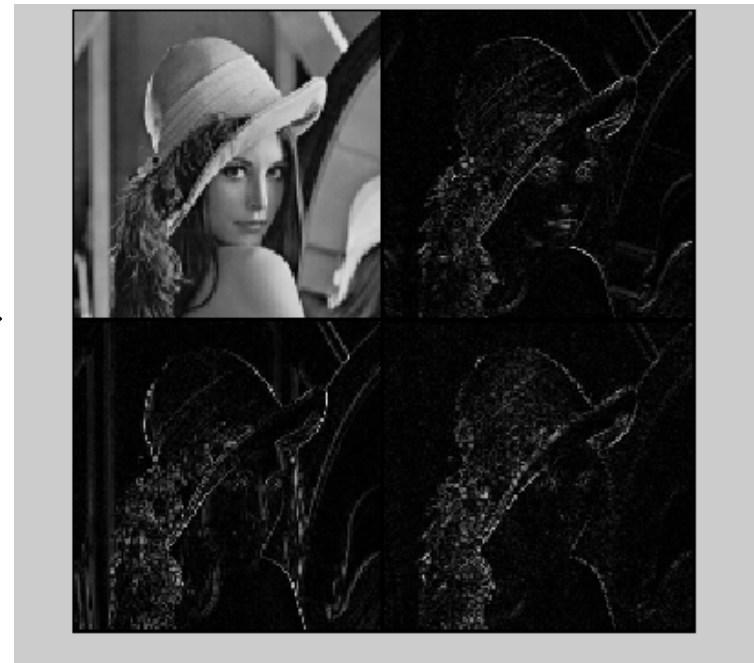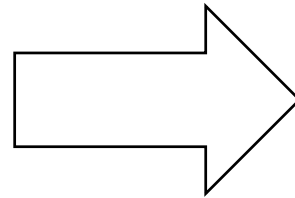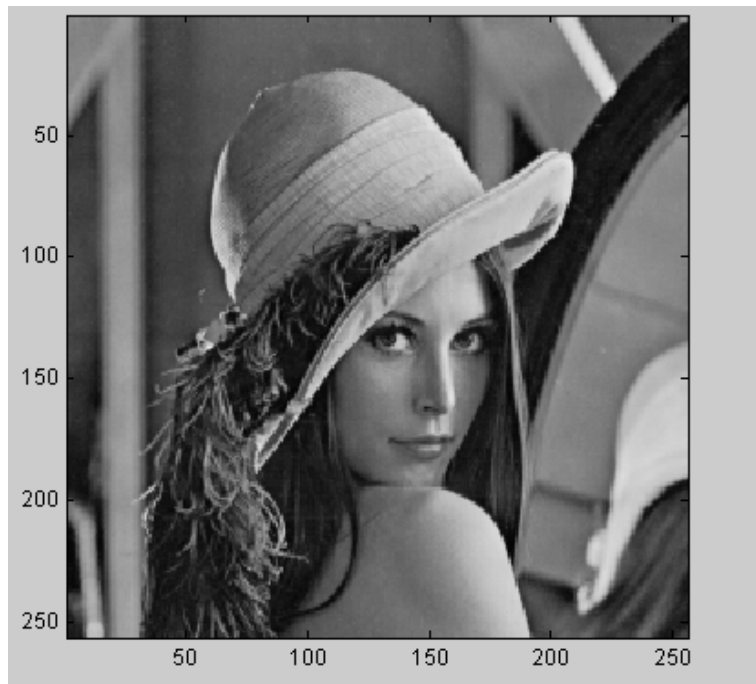| Code	Length/Size
000	5/0
001	5/1
1111 10	5/2
011	5/3
100	5/4
1111 1110	5/5
...	...
110	5/A
1110	4/0
010	4/1
101	4/2
...	...
1111 0	4/A

JPEG Problem

- Compression Granularity is in Unit of 8*8

JPEG 2000

- Wavelet Transform

MPEG – Video Compression

- Filename Extension: MPEG-4 `.mp4`
- Moving Pictures Experts Group
- Intuition
  - Adjacent frames are similar and changes are due to foreground motion

Background

foreground

MPEG Compression: I Frame and P Frame

- I (intra) Frame
  - Independent frames
  - Coded without reference to other frames (JPEC Compressed)

- P (predictive) Frame
  - Not Independent frames
  - Predicted from a past frame (I or P)

I Frame	P Frame	P Frame	I Frame

Forward prediction
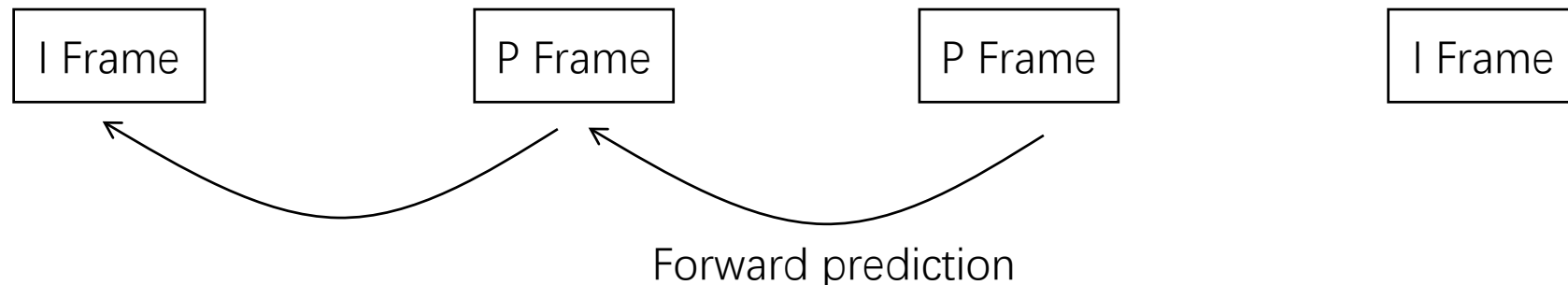
MPEG Compression: Forward Prediction
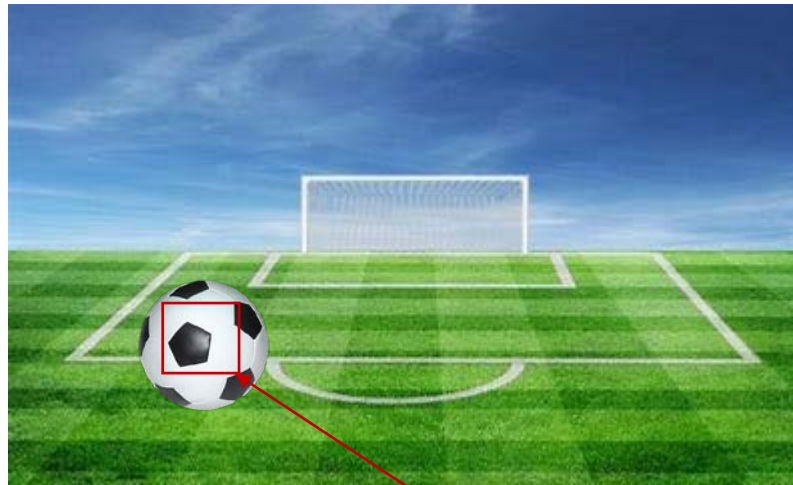
current location vector
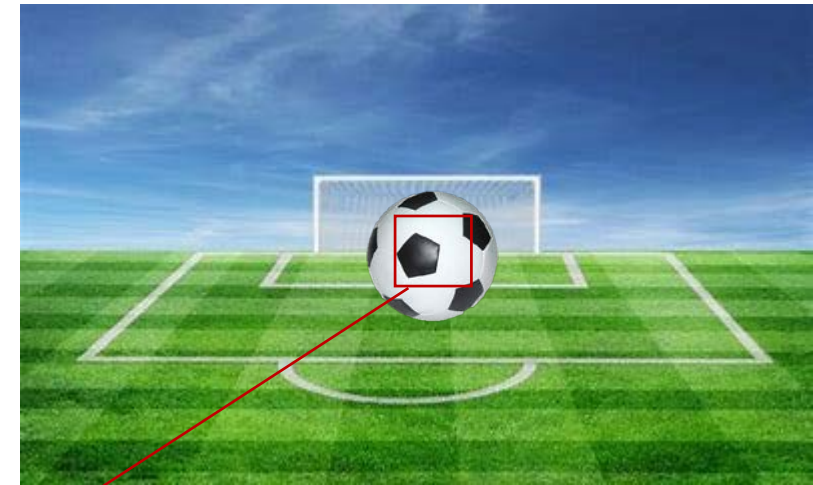
future motion vector

16*16

Difference is encoded similar to JPEG

MPEG Compression: B Frame

- B (Bidirectional) Frame
  - Not Independent frames
  - Reason: enhance forward prediction
  - Coded with reference to both previous and future frames (I or P)

Bidirectional prediction

| I Frame | B Frame | P Frame | B Frame | P Frame | B Frame | I Frame |

Forward prediction

MPEG over a Network

- A Video Stream

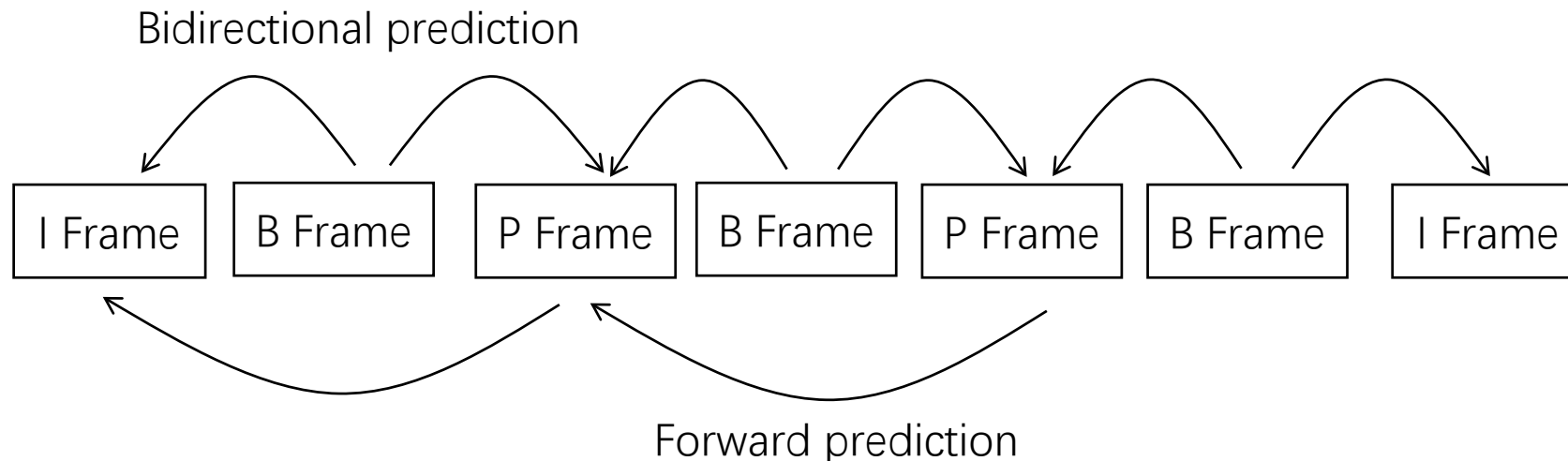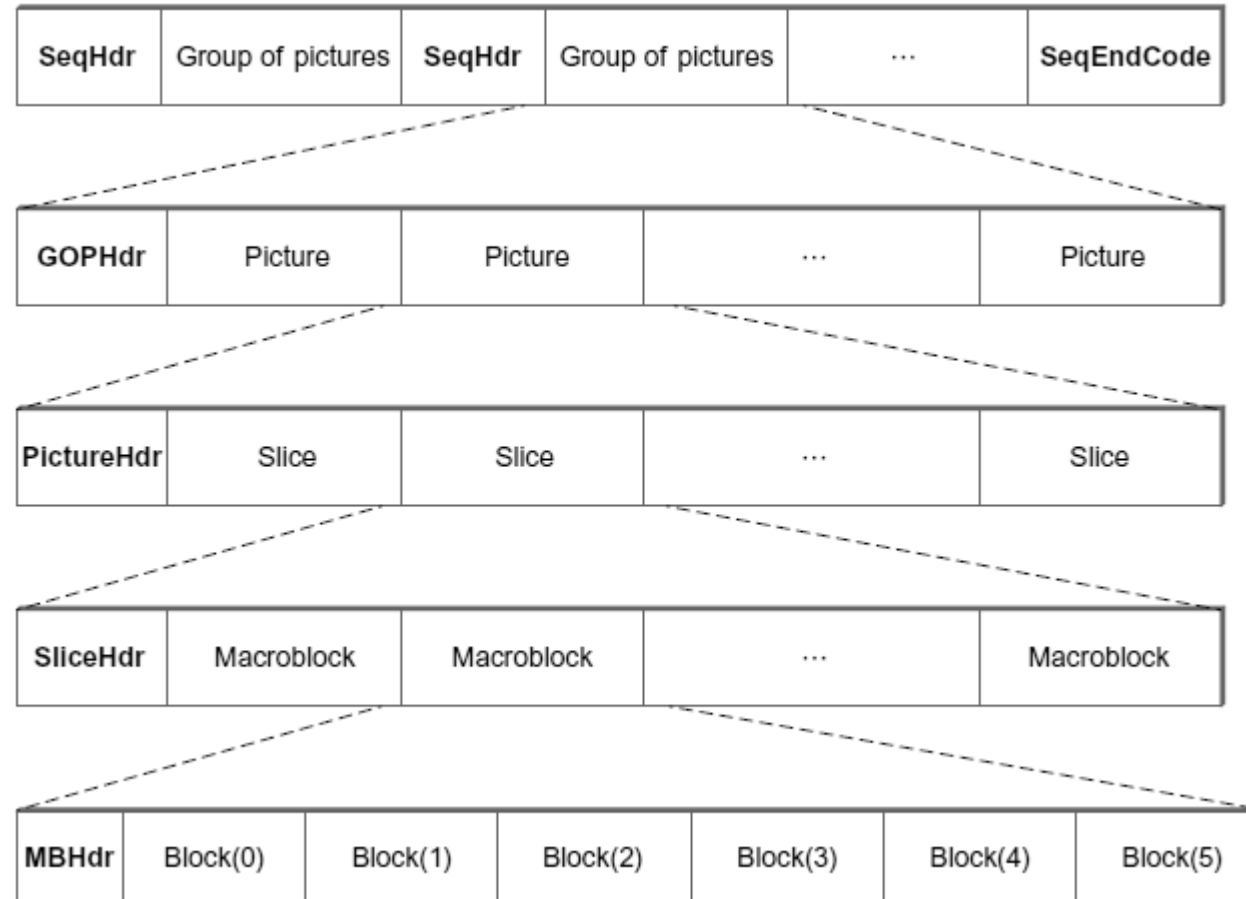| SeqHdr	Group of pictures	SeqHdr	Group of pictures	...	SeqEndCode

GOPHdr	Picture	Picture	...	Picture

PictureHdr	Slice	Slice	...	Slice

SliceHdr	Macroblock	Macroblock	...	Macroblock

MBHdr	Block(0)	Block(1)	Block(2)	Block(3)	Block(4)	Block(5)

MPEG over a Network

- Frame Sequence
  - Target Seq: IBBBBPBBBBI
  - Transmitting Seq: IPBBBBIBBBB
    - Large Delay
  - For Interactive Videos
    - Only use I and P frames or pure I frames

MP3 – Audio Compression

- Filename Extension: `.mp3`
- A part of MPEG
  - MP3 is introduced in MPEG-1 to encode audio
- Intuition
  - Human ear are less sensitive to high frequency sound
  - Divide audio signal into subbands
  - Compressing subband by allocating different numbers of bits

Reference

- Textbook 7.2