INT 307
Multimedia Security System

Image Representation and Compression

Sichen.Liu@xjtlu.edu.cn

XJTLU | SCHOOL OF FILM AND TV ARTS

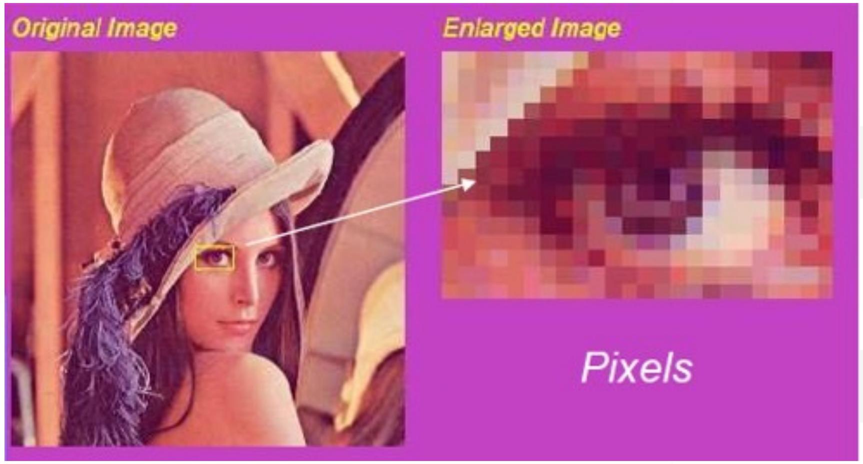
Xi'an Jiaotong-Liverpool University
西交利物浦大学

Aims

- Understand how people perceive image and video
- Master how images are presented and compressed in computer systems
- Understand the technology about information encoding

Image / Video Representation

- Images are composited by pixels
  - Each pixel has a color
  - The density of pixel is known as ppi
  - Human visual system has a resolution of approximate 300 ppi
- Video is composited by frames
  - Each frame presents the image at a moment
  - The rate of frame known as fps
  - Human visual system approximates 60 fps



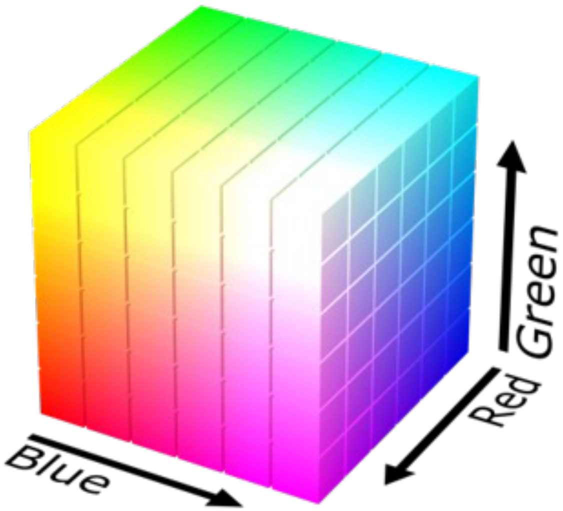
Color Space

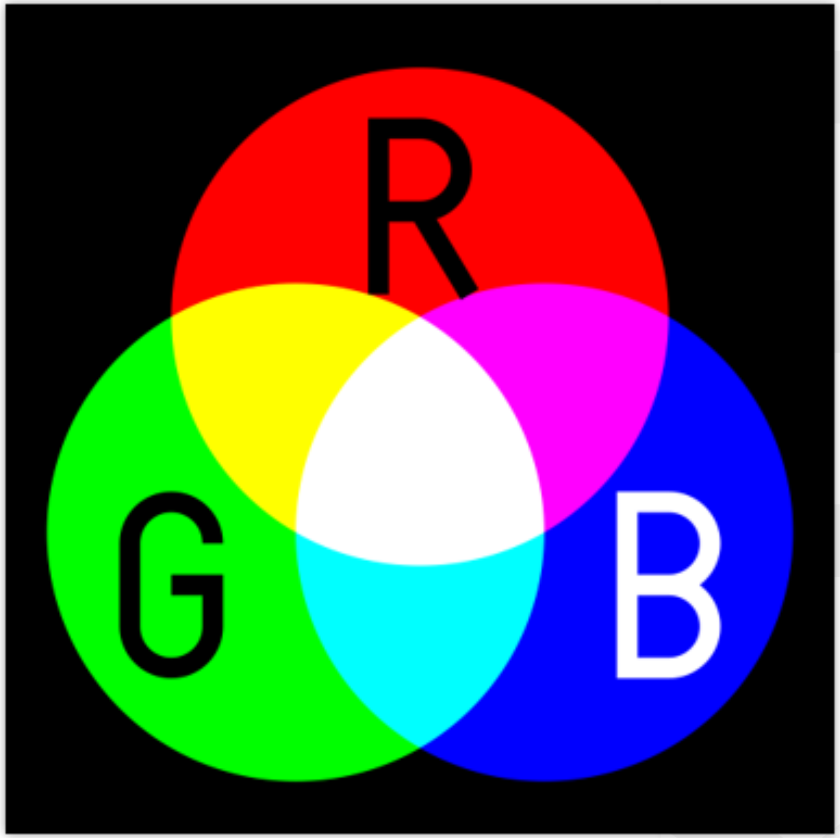
Usually, a vector is used to represent the color of a pixel, where each element in the vector represents a component

- RGB → Screen display (Red, Green and Blue)
- CMY(K) → Printer: Cyan, Magenta and Yellow (with Black)
- YUV → MPEG / JPEG system: separate light and color

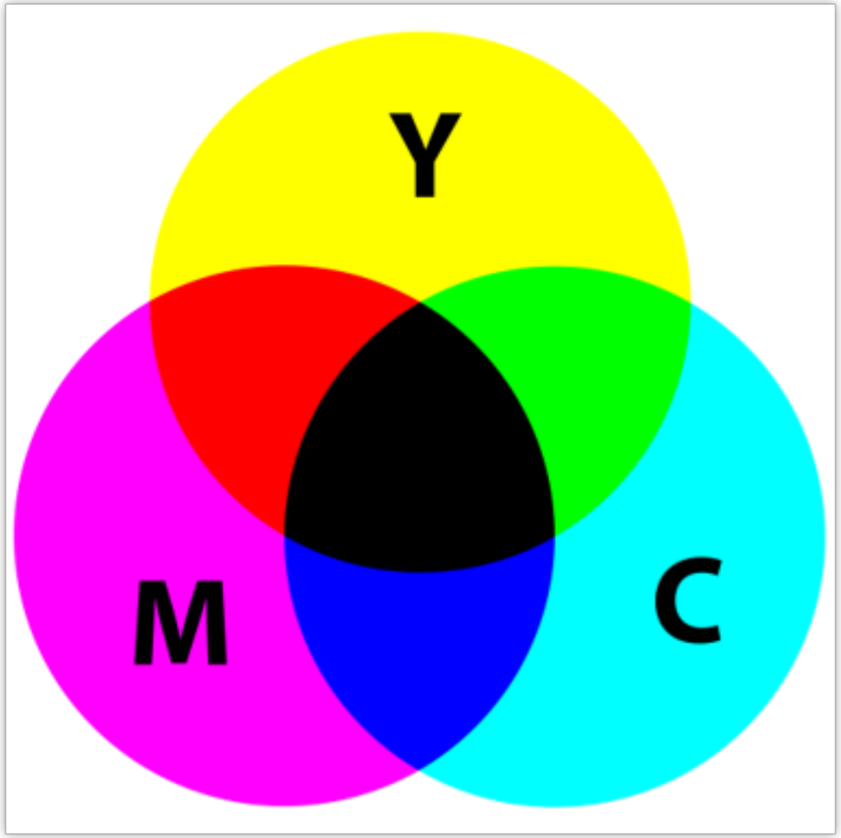
RGB System

Additive Colour System





CMY(K) System



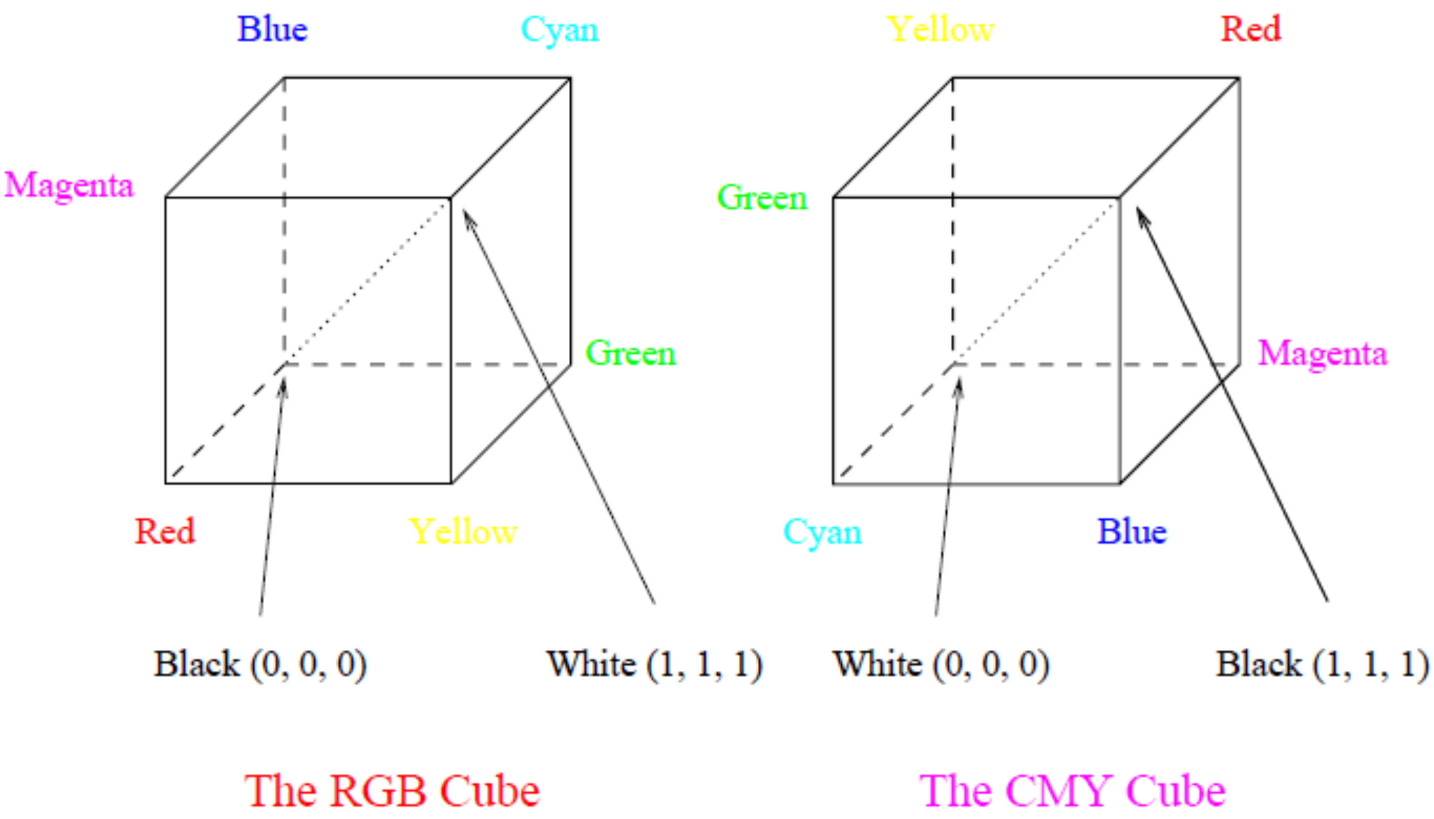
Subtractive color system

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

CMY(K) System

Subtractive color system

- No perfect Black → K component for black
- Printer system

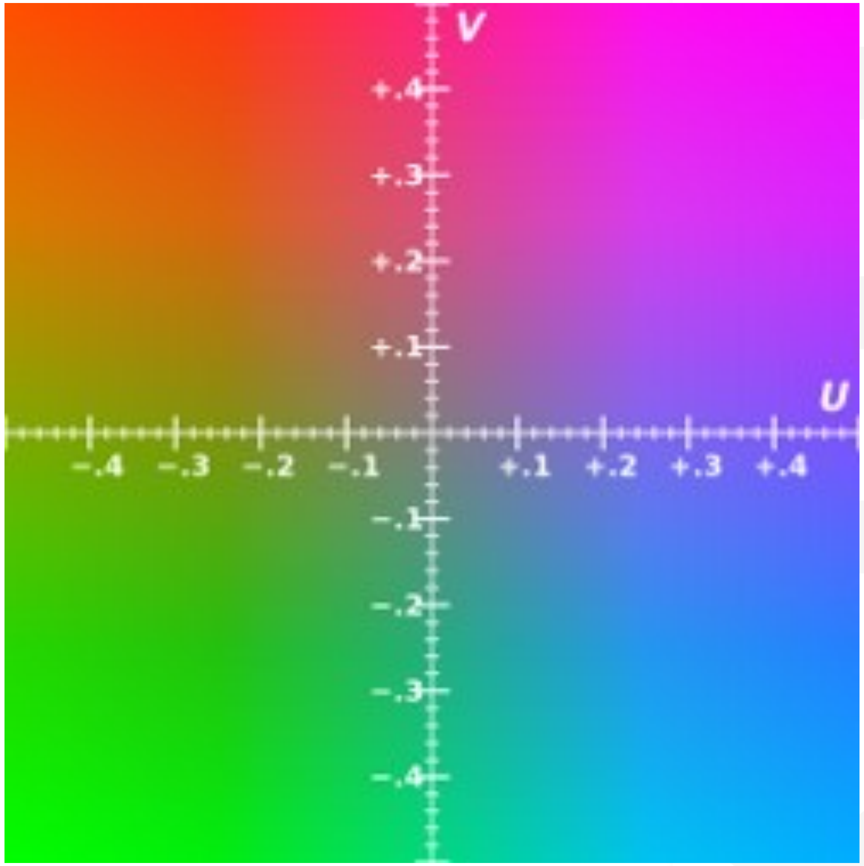


YUV System

- Separate Lightness and Color
- Easier compression as human visual system is more sensitive for light

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

When Y = 0.5, the YUV system seems like



Calculation Question

- For an image RGB system is used with a depth of 8 bits for each component, how many bits needed to represent a picture whose resolution is 1920×1080?

- With the same configuration of the previous question as a frame of a piece of video, how many bits needed to record a one-hour video whose resolution is 1920×1080 with a frame rate of 60 fps?

Facts on Multimedia

- Fact 1: A film frame = 480 x 260 pixels = 374,400 bytes = 2,995,200 bits =599,040 alphabets = 124,025 English words
- Fact 2: A 2 hour 10 mins movie = 187,200 frames = 70,087,680,000 bytes= 23,217,480,000 English words = 7,067 years of words a person can talk
- Fact 3: 128 hours of video = 13,648,457 frames = 31,131,584,804,571 bytes= storage space of 31 1-TB hard disks at my laptop

Why Needs Compression?

- Demand
  - VHS (Video Home System): 352 x 240 x 30 x 24 = 60.8 Mb / s
  - DVD (Digital Versatile Disc): 720 x 480 x 30 x 24 = 248.8 Mb / s
  - HDTV (High Definition Television): 1280 x 720 x 60 x 24 = 1327.2 Mb / s
- Supply
  - Telephone: < 56 Kb/s
  - ISDN: 64 - 144 Kb/s
  - T1: 1.5 Mb/s
  - Ethernet: 10 - 100 Mb/s
  - 802.11b: 1 - 10 Mb/s
  - 802.11g: 50 - 100 Mb/s
  - CD-ROM: 1.2 Mb/s

Compression

As a result, we need to compress the media

- Image Compression
- Video Compression
  - Intra-frame compression
  - Inter-frame compression

Compression ratio

- If the compression and decompression processes induce no information loss, then the compression scheme is **lossless**; otherwise, it is **lossy**.

- **Compression ratio**:

$$compression\,ratio = \frac{B_0}{B_1}$$

- B_0 – number of bits before compression
- B_1 – number of bits after compression

Information Measurement

- Information Measure: Consider a symbol x with an occurrence probability p, its information content

$$I = \log \frac{1}{p(x)} = -\log p(x)$$

  - The smaller the probability, the more info. the symbol contains
  - The occurrence probability somewhat related to the uncertainty of the symbol

Information Entropy

- The Entropy is defined as the average information content per symbol of the source. The Entropy, H, can be expressed as follows

$$H = -\sum_{i=1}^{m} p_i \log_2 p_i \quad \text{bits}$$

- From this definition, the entropy of an information source is a function of occurrence probabilities.
- The entropy reaches the Max. when all symbols in the set are equally probable

Information Content

- Consider the two blocks of binary data shown below which contains the most information?

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

$$P_0 = \frac{63}{64}$$

$$P_1 = \frac{1}{64}$$

$$H = -\frac{63}{64}\log_2\frac{63}{64} - \frac{1}{64}\log_2\frac{1}{64}$$

$$= 0.116 \text{ bits/pixel}$$

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

$$P_0 = \frac{32}{64} = \frac{1}{2}$$

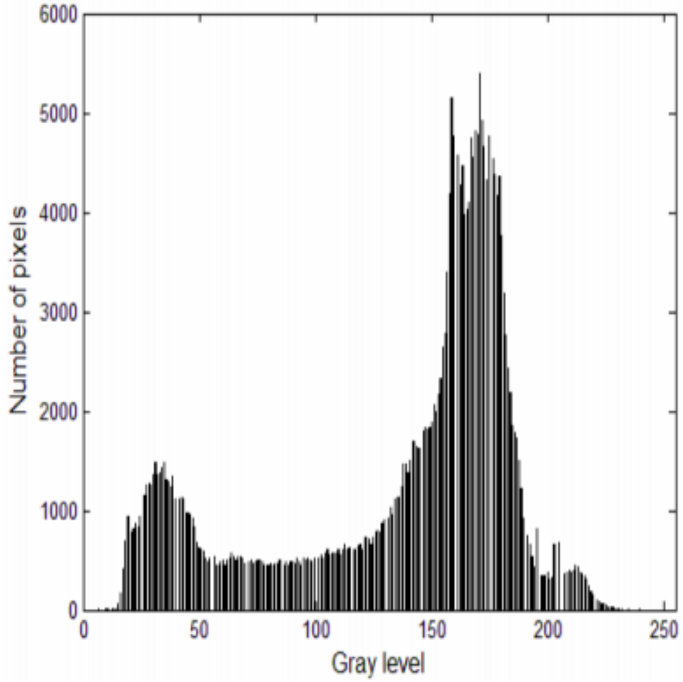
$$P_1 = \frac{32}{64} = \frac{1}{2}$$

$$H = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}$$

$$= 1.0 \text{ bits/pixel}$$

Entropy Coding

- Image Histogram: entropy = 7.63 bits/pixel



Entropy Coding

- Fixed length coding

Symbol	Probability	Codeword	Codeword Length
A	0.75	00	2
B	0.125	01	2
C	0.0625	10	2
D	0.0625	11	2

- Average bits/symbol
- Average bits/symbol = 0.75*2 + 0.125*2 + 0.0625*2 + 0.0625*2 =2.0 bits/pixel

Entropy Coding

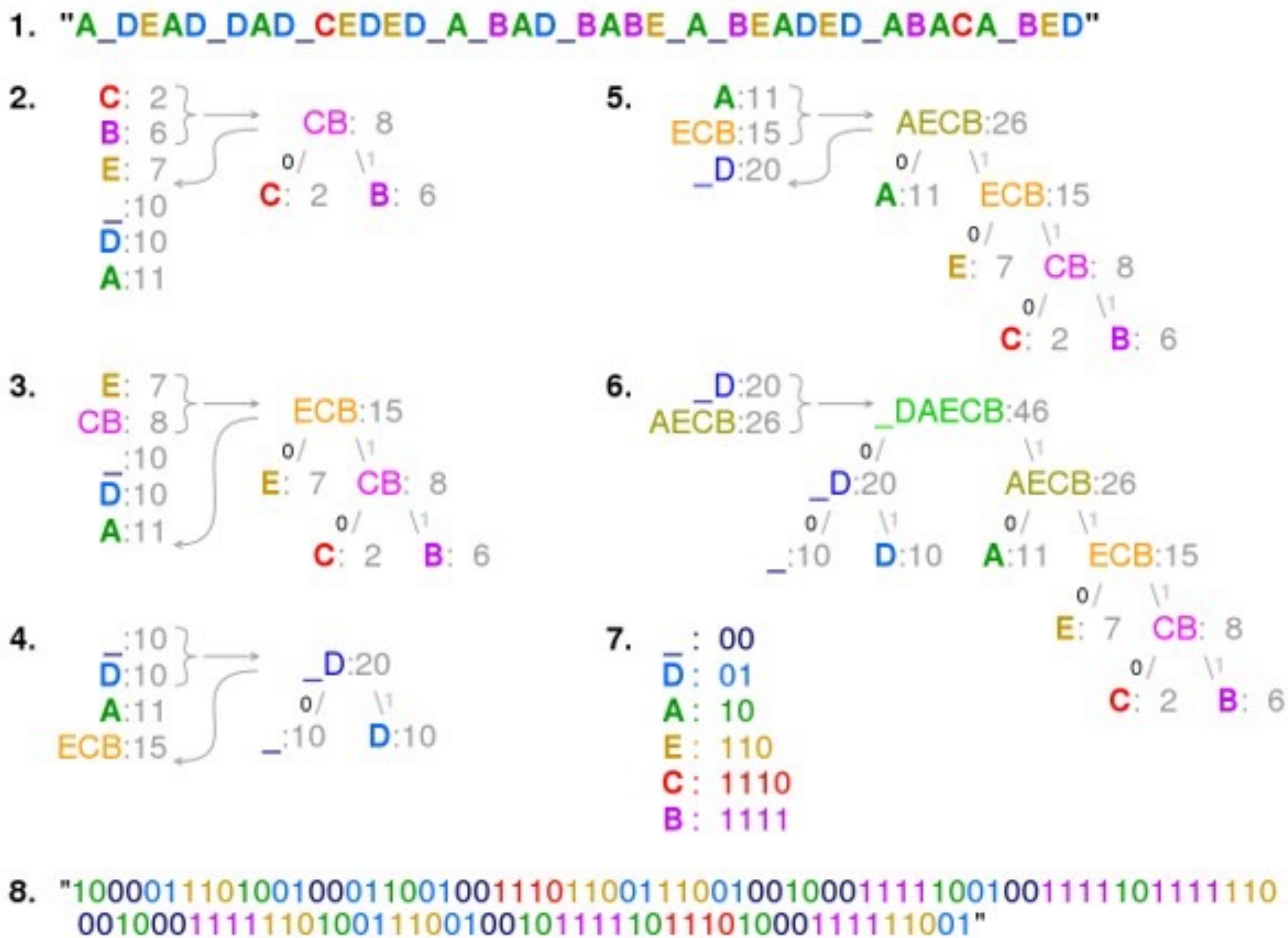
- Variable Length Coding

Symbol	Probability	Codeword	Codeword Length
A	0.75	0	1
B	0.125	10	2
C	0.0625	110	3
D	0.0625	111	3

- Average bits/symbol
- Average bits/symbol = 0.75*1 + 0.125*2 + 0.0625*3 + 0.0625*3 = 1.375 bits/pixel

Huffman Codingwords

- If the symbol probabilities are known, Huffman codewords can be automatically generated
  - Reorder in decreasing order of probability at each step
  - Merge the two lowest probability symbols at each step

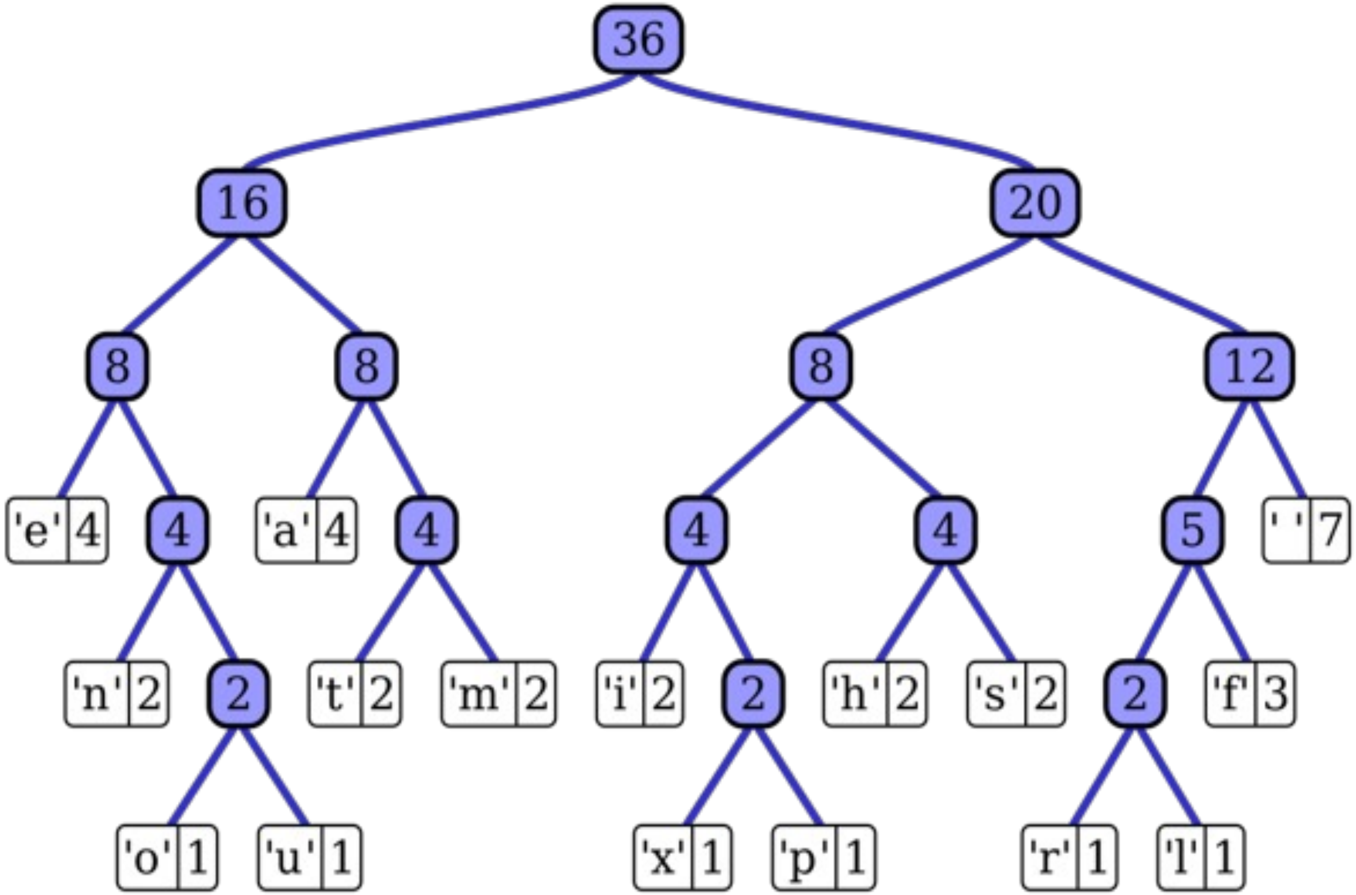


Huffman Codingwords

- Huffman codewords have to be an integer number of bits long
- The symbol probabilities must be known in the decoder
  size. If not, they must be generated and transmitted to the decoder with the Huffman coded data
- A larger of number of symbols results in a large codebook
- Dynamic Huffman coding scheme exists where the code words are adaptively adjusted during encoding and decoding, but it is complex for implementation

Exercise

Please use Huffman coding to code “this is an example of a huffman tree”



Arithmetic Coding

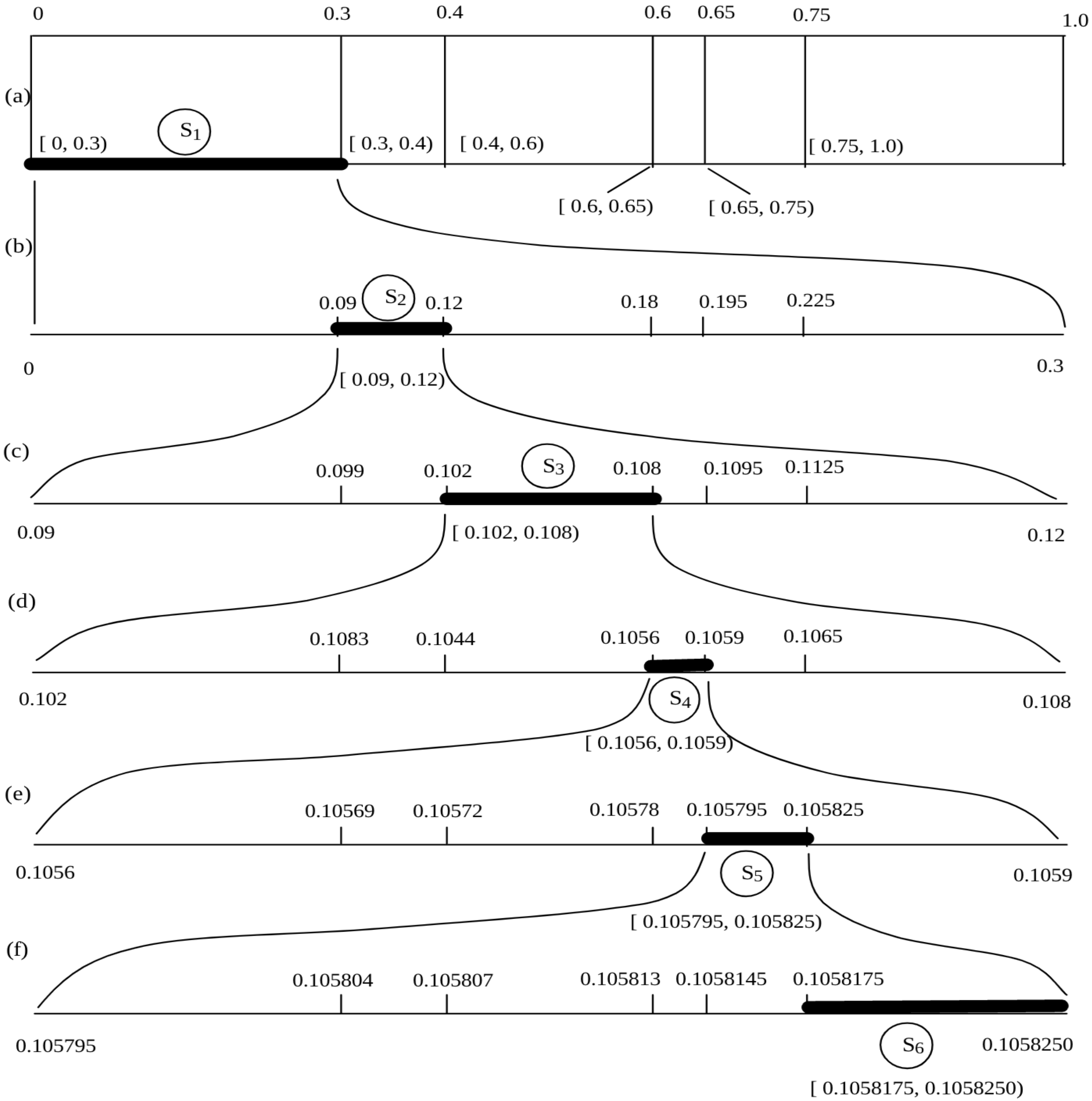
- It overcomes limitation of Huffman coding: non-integer length coding, and probability distribution can be derived in real-time
- It operates by replacing a stream of input symbols with a single floating point output number.
- Consider the following symbols with probabilities (CP is the cumulative probability):

Source symbol	Occurrence probability	Associated subintervals	CP
S_1	0.3	[0, 0.3)	0
S_2	0.1	[0.3, 0.4)	0.3
S_3	0.2	[0.4, 0.6)	0.4
S_4	0.05	[0.6, 0.65)	0.6
S_5	0.1	[0.65, 0.75)	0.65
S_6	0.25	[0.75, 1.0)	0.75

Arithmetic Coding

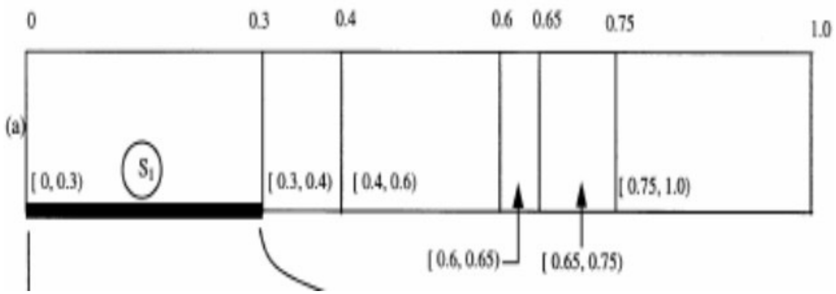
- Suppose we wish to encode the string: S1; S2; S3; S4; S5; S6;
- We start with the interval $[L, H)$ and set to [0, 1) for 6 symbols.
- Since the first symbol is S1, we pick up its subinterval[L, H) = [0.0, 0.3), and any real symbols can be considered as disjoint to be divided in the same way.
- To encode the S2, we use the same procedure as used in above to divide the interval [0, 0.3) into six sub-intervals. We pick up the S2 subinterval [0.09, 0.12)

Arithmetic Coding



Arithmetic Decoding

- For encoding, the input is a source symbol string and the output is a subinterval (called the final subinterval). For our case, it is [0.1058175, 0.1058250)
- Decoding sort of reverses what encoding has done
- The decoder knows the encoding procedure and therefore has the information contained in the following figure



- It compares the lower end point of the final subinterval 0.1058175 with all the end points in the above figure **0** < **0.1058175** < **0.3**

Run-length Coding

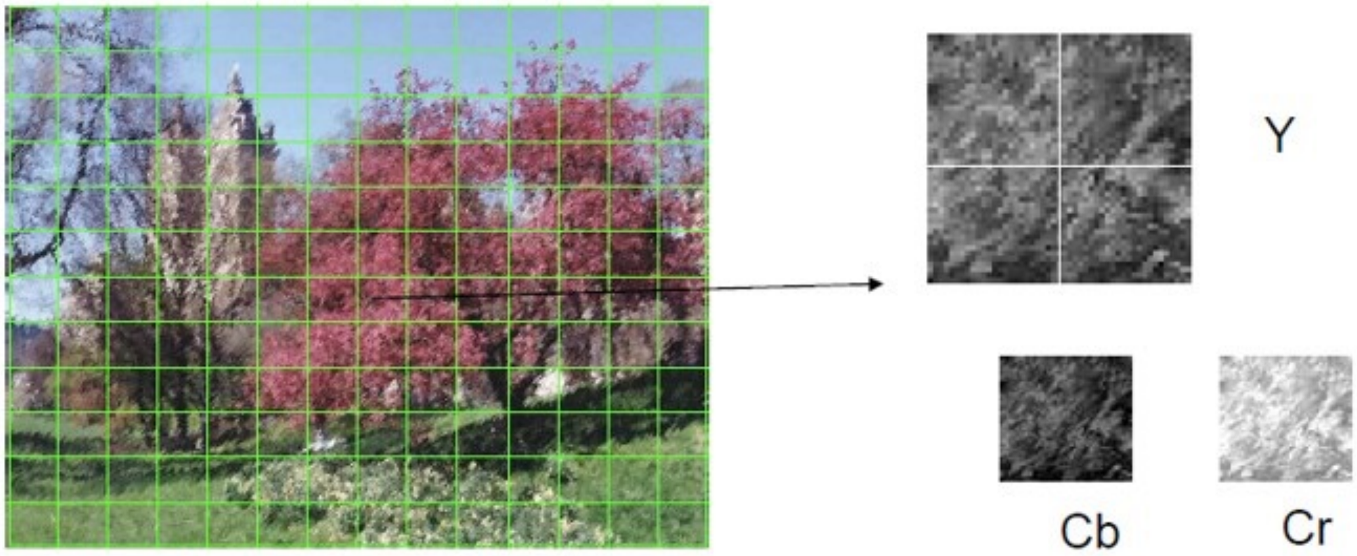
- In run-length coding, a run of consecutive identical symbols is combined together and represented by a single codeword.
- The various run-lengths are then represented by a variable length (e.g. Huffman) codeword.
- Use an escape code.

JPEG

- Divide Images into 8×8 (luminance) or 16×16 (chrominance) blocks
- Perform DCT on image blocks
- Apply Quantization
- Perform Zigzag ordering and Run-length encoding
- Entropy Encoding (Huffman Encoding)

Macroblocks

- An image is divided into 8×8 blocks for the luminance components
- An image is divided into 16×16 blocks for the chrominance components
- Chrominance blocks are down-sampled to 8×8 blocks (called 4:2:2 format)
- Zero-Padding on boundary blocks



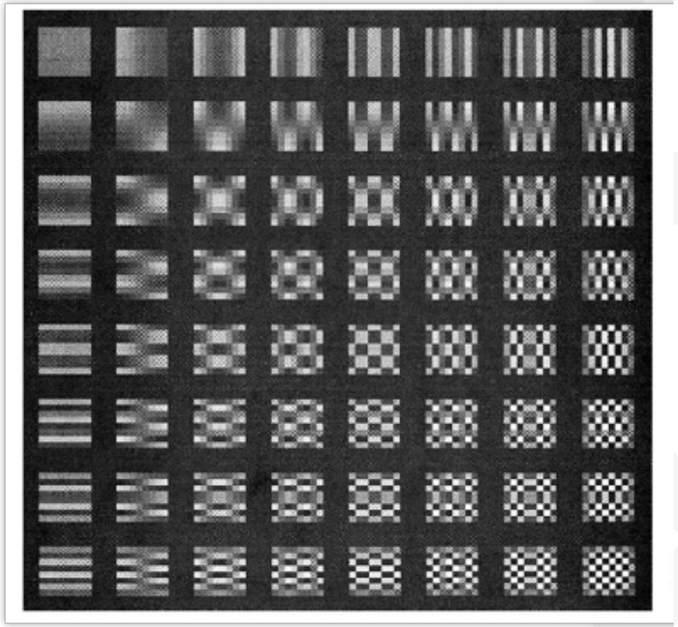
Discrete Frequency-Domain Analysis

$$F(\mu,\nu) = \frac{C(\mu)}{2}\frac{C(\nu)}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} f(x,y)\cos[(2x+1)\mu\pi/16]\cos[(2x+1)\nu\pi/16]$$

$$C(\mu) = \begin{cases} \frac{1}{\sqrt{2}}, if \mu = 0 \\ 1, if \mu > 0 \end{cases}$$

- Frequency of image
- Rapid changes → High frequency
- Minor changes → Low frequency
- Usually the image is low frequency (redundancy in spatial)

2D-DCT bases



Quantization

- Human eye is good at seeing small difference in brightness over a relative area
- Not good at the brightness variations of high frequency
- Reduce the amount of information in the high frequency component
- Divide by a constant and round to the nearest integer
- Effect: high frequency → zero
- Compression ratio: larger element value → greater compression.

Quantization

- Luminance Quantization Table and Chrominance Quantization Table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

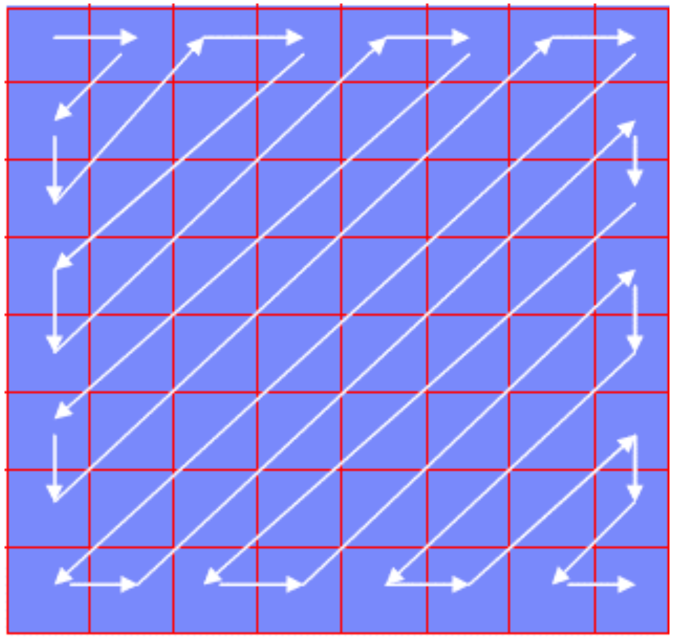
(a)

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(b)

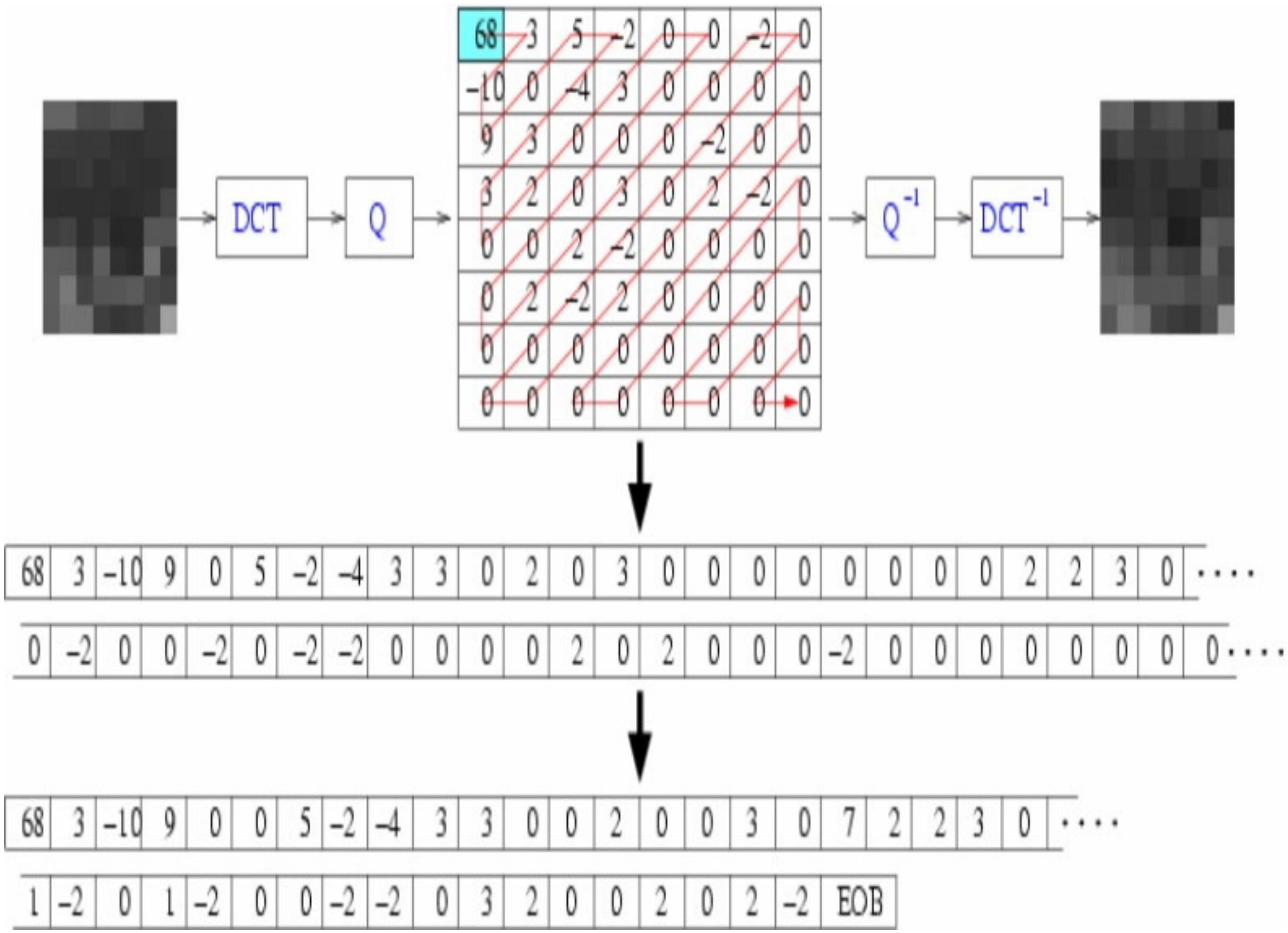
Zig-Zag Order

An order to sort the 2D DCT coefficients to 1D signals



Lossless Compression

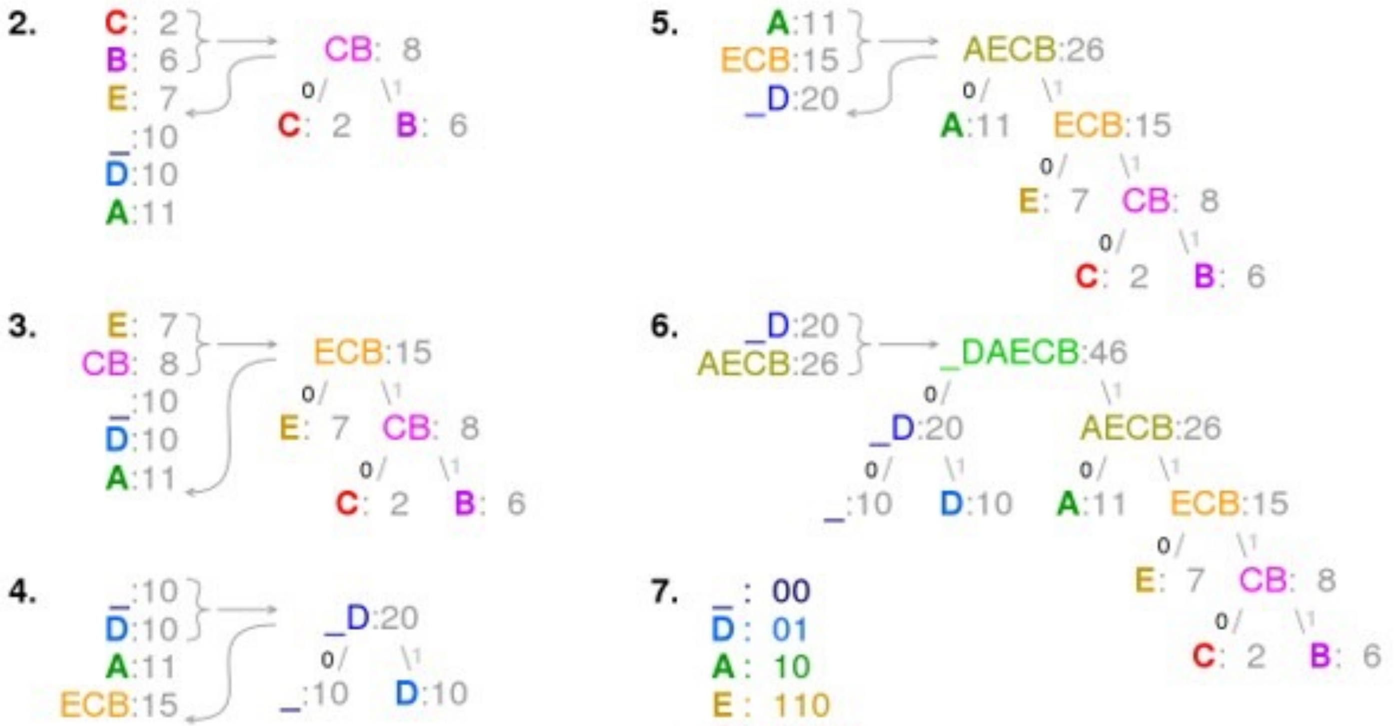
Run Length Coding



Lossless Compression

Huffman Coding

1. "A_DEAD_DAD_CEDED_A_BAD_BABE_A_BEADED_ABACA_BED"



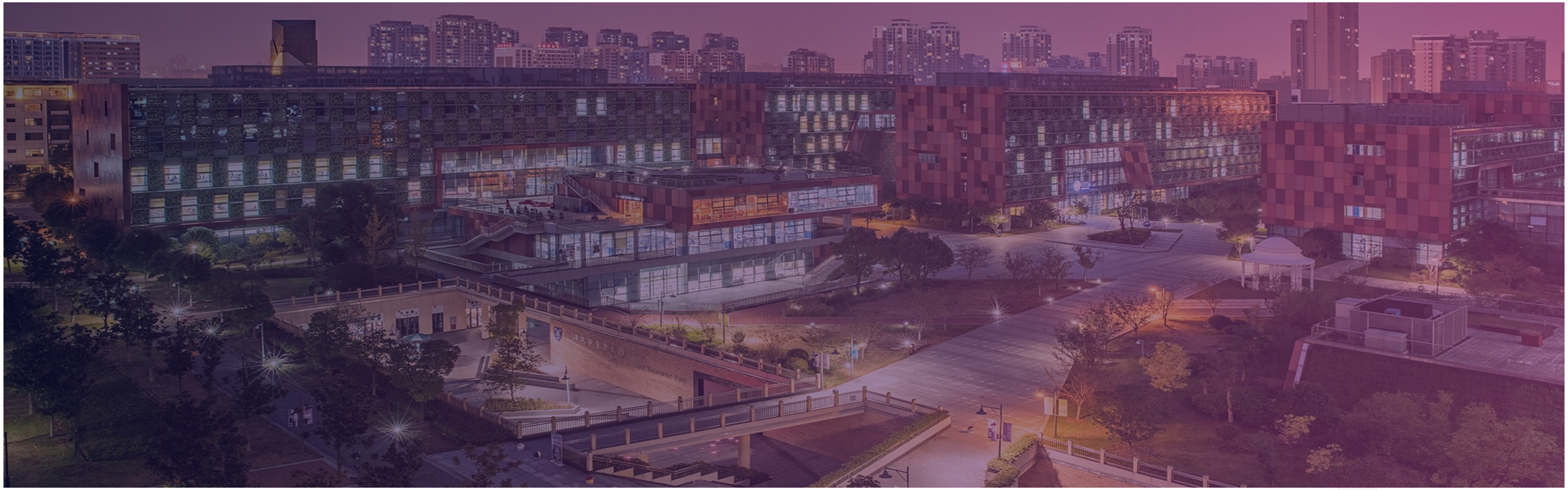
7.
_ : 00
D : 01
A : 10
E : 110
C : 1110
B : 1111

8. "1000011101001000110010011101100111001001000111110010011111011111110
0010001111110100111001001011111011101000111111001"

JPEG

As an example, the bottom picture shows our compressed image.

This right image is a mere 1.5% of the original size.



THANK YOU

VISIT US

WWW.XJTLU.EDU.CN

FOLLOW US

@XJTLU

Xi'an Jiaotong-Liverpool University
西交利物浦大学

XJTLU | SCHOOL OF FILM AND TV ARTS