DIGITAL IMAGE PROCESSING

IMAGE COMPRESSION

CHANG SHU

2022-12-6

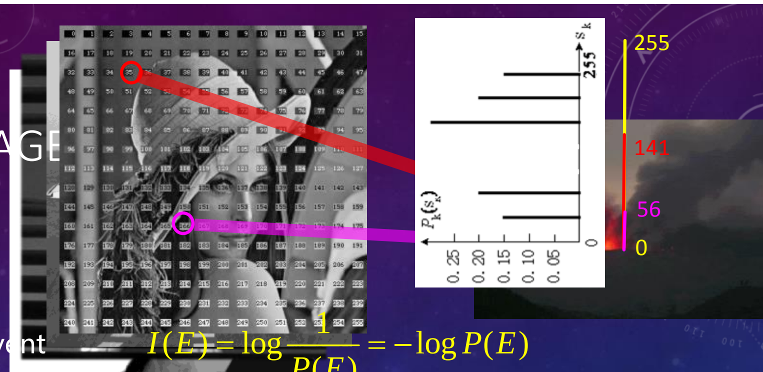
IMAGE COMPRESSION

Calculate the amount of data required to represent a two-hour high definition(HD) television movie using 1920×1080×3×8 bit pixel arrays, at rates of 30 fps(frames per second)

$$1920 \times 1080 \times 8 \times 3 \times 30 = 1.493 \text{ Gbit/s} = 186.624 \text{ MB/s}$$

$$186.624 \text{ MB/s} \times 3600s / h \times 2hrs = 1343.7\text{GB}$$

IMAGE



- fundamentals
  - Entropy of a single event $I(E) = \log \frac{1}{P(E)} = -\log P(E)$
  - Entropy of a zero-memory source $H = -\sum_i P(a_i) \log P(a_i)$
  - Entropy of a zero-memory intensity source(image)
  - Shannon's first theorem
  - Entropy of Markov(finite memory) source

IMAGE COMPRESSION

$$-\log_2 P(E) \sim (bit)$$

- fundamentals
  - Entropy of a single event $I(E) = \log \frac{1}{P(E)} = -\log P(E)$
  - Entropy of a zero-memory source $H = -\sum_i P(a_i) \log P(a_i)$ source symbol
  - Entropy of a zero-memory intensity source(image) $\tilde{H} = -\sum_{k=0}^{L-1} P_r(r_k) \log_2 P_r(r_k)$
  - Shannon's first theorem $H = \lim_{n \to \infty} \frac{L_{\text{avg},n}}{n}$
  - Entropy of Markov(finite memory) source

$$H_N = -\sum_{a_0} \sum_{a_1} \cdots \sum_{a_N} P(a_0, a_1, \cdots, a_{N-1}) \log_2 P(a_0 | a_1, \cdots, a_{N-1})$$

$$H_0 \geq H_1 \geq \cdots \geq H_{N-1} \geq H_N = H_{N+1} = H_{N+2} = \cdots = H_\infty$$

IMAGE COMPRESSION

- fundamentals
  - Entropy
  - average code length
  - Coding efficiency
  - Compression ratio
  - SNR

$$\tilde{H} = -\sum_{k=0}^{L-1} P_r(r_k)\log_2 P_r(r_k)$$

$$R(x) = \sum_i P(x_i)L_i$$

$$\frac{H(x)}{R(x)}$$

$$c = \frac{b_1}{b_2} = \frac{R_{before}(x)}{R_{after}(x)} = \frac{ceil(\log_2(L))}{R_{after}(x)}$$

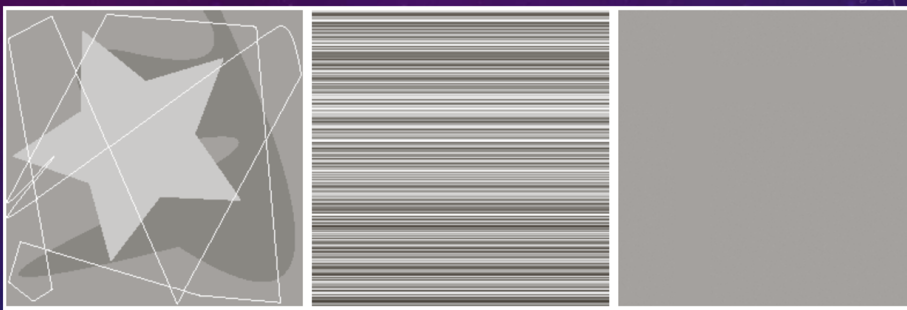
$$10\log\frac{\sigma_x^2}{\sigma_e^2}$$

IMAGE COMPRESSION

- Image data redundancy
  - Coding redundancy
  - Spatial and temporal redundancy
  - Irrelevant information

$$R(x) = \sum_i P(x_i) L_i$$

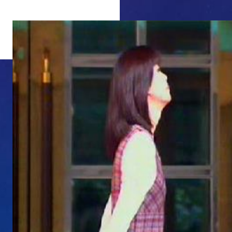
IMAGE COMPRESSION

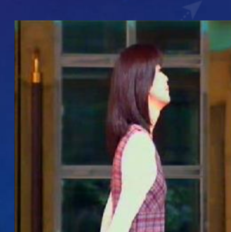


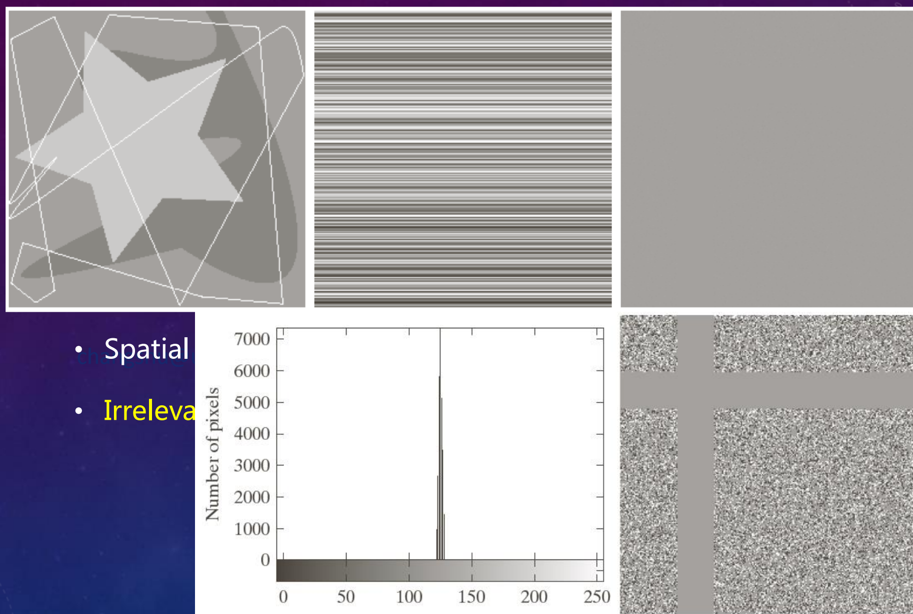
r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

IMAGE COMPRESSION

0.631	0.642	0.641	0.628	0.288	0.325	0.375	0.419
0.700	0.716	0.721	0.710	0.435	0.493	0.568	0.630
0.711	0.811	0.842	0.845	0.558	0.635	0.742	0.835







- Spatial
- Irreleva

IMAGE COMPRESSION

- Image compression models

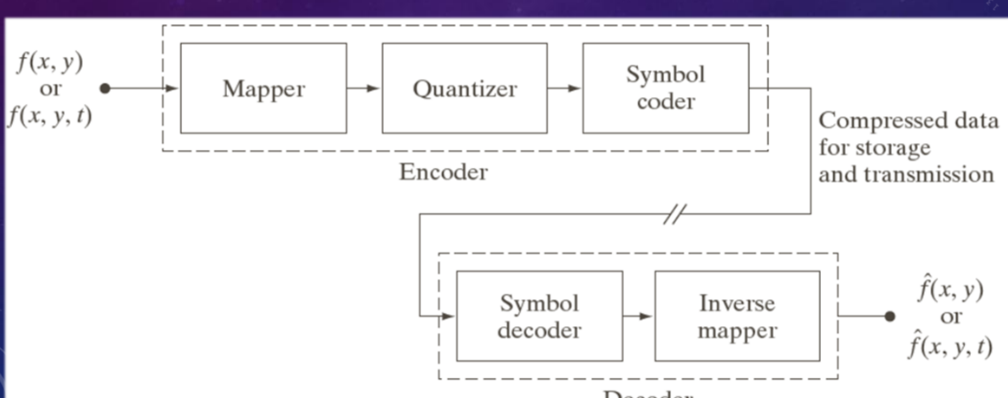


IMAGE COMPRESSION

Huffman's procedure:

(1) Create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into a single symbol that replaces them in the next source reduction

(2) Code each reduced source, starting with the smallest source and working back to the original source.

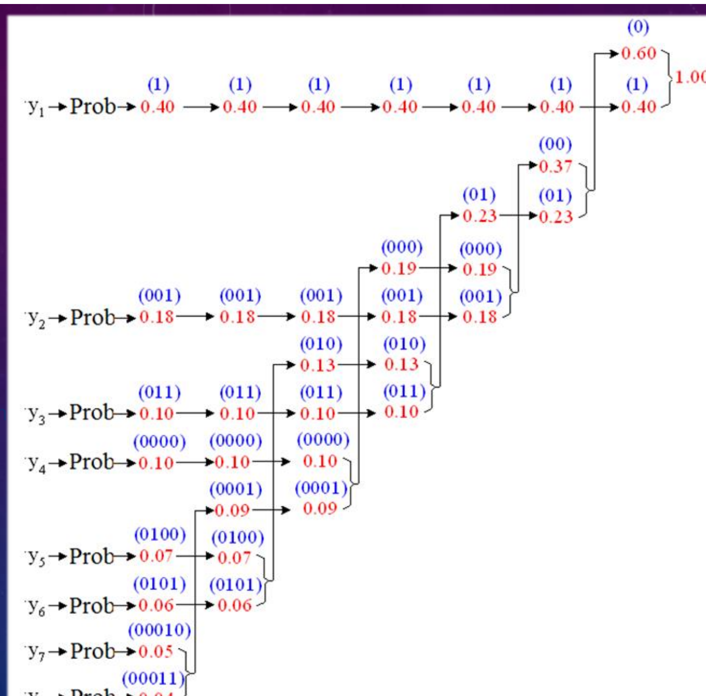


IMAGE COMPRESSION

Entropy of the intensity source：

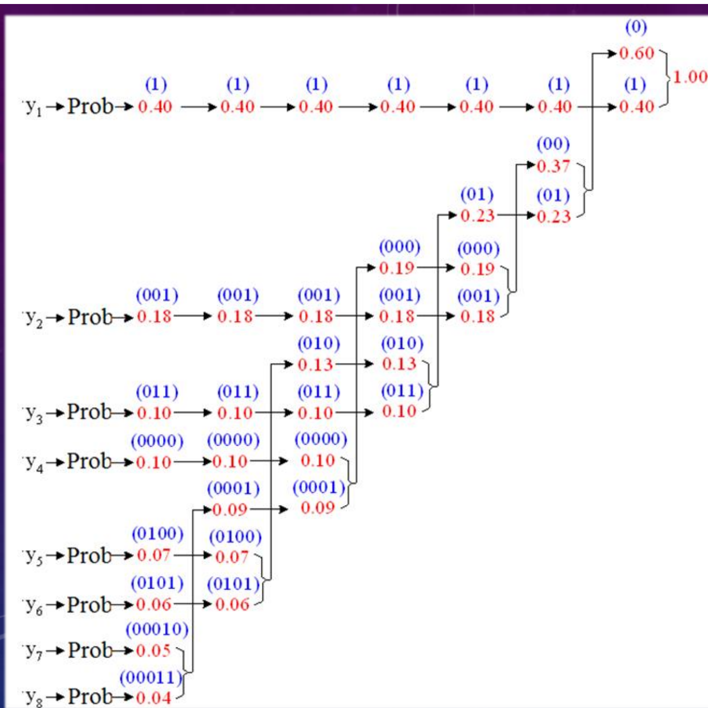
$$H = -\sum_{i=1}^{L} P_i \log_2 P_i = -(0.4\times\log_2 0.4 + 0.18\times\log_2 0.18 + 2\times 0.1\times\log_2 0.1 +$$
$$0.07\times\log_2 0.07 + 0.06\times\log_2 0.06 + 0.05\times\log_2 0.05 + 0.04\times\log_2 0.04) = 2.55$$

Average code length：

$$R = \sum_{i=1}^{L} P(a_i)\cdot l_i = 0.40\times 1 + 0.18\times 3 + 0.10\times 3 + 0.10\times 4$$
$$+ 0.07\times 4 + 0.06\times 4 + 0.05\times 5 + 0.04\times 5 = 2.61$$

Compression ratio： $C = 3/2.61 = 1.15$

Coding efficiency: $\eta = H/L = 2.55/2.61 = 97.8\%$



SION

Huffman decoding：

What is the code symbols for the encoded string 010100111011：

$y_6 y_2 y_1 y_1 y_3$

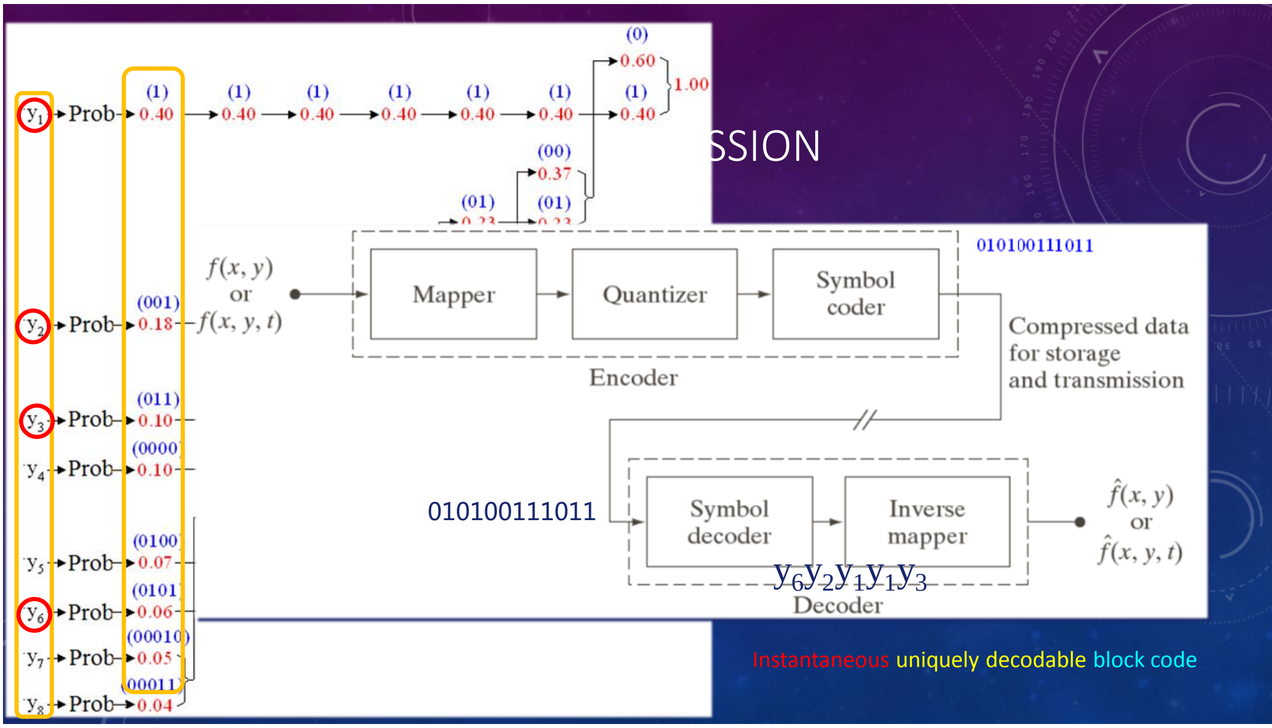


IMAGE COMPRESSION

- Huffman code is an instantaneous uniquely decodable block code
  - block code --- each source symbol is mapped into a fixed sequence of code symbols
  - instantaneous --- each code word in a string of code symbols can be decoded without referencing succeeding symbols
  - uniquely decodable --- any string of code symbols can be decoded in only one way

IMAGE COMPRESSION

- Arithmetic coding :

An entire sequence of source symbols is assigned a single arithmetic code word. The code word itself defines an interval of real numbers between 0 and 1.

As the number of symbols in the message increases, the interval used to represent it becomes smaller and the number of information units required to represent the interval becomes larger.

IMAGE COMPRESSION

$$\log_{10} x = \frac{\log_2 x}{\log_2 10}$$

$$H = -\sum_i P(a_i)\log_{10} P(a_i)$$

[0,1) [0.1 0.4) [0.1 0.13)

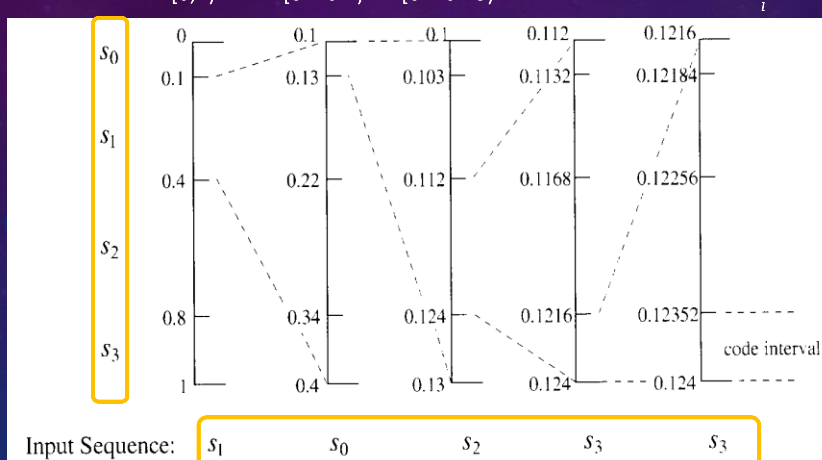


FIGURE 4 Arithmetic coding example.

IMAGE COMPRESSION

[0,1) [0.1 0.4) [0.1 0.13)

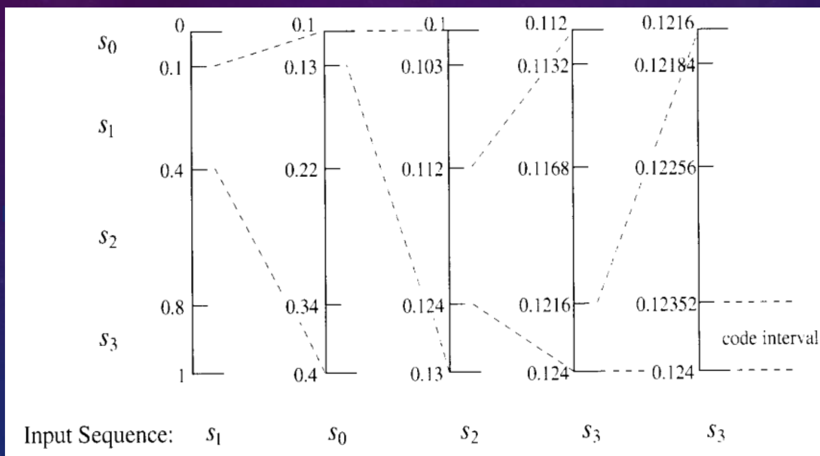


FIGURE 4 Arithmetic coding example.

Question: What's the input sequence for a given arithmetically-coded message 0.618 if S3 is the end-of-message indicator?

$S_2S_2S_1S_3$