Digital Image Processing (CSE/ECE 478)

Lecture # 20: Image Compression I

Avinash Sharma

Center for Visual Information Technology (CVIT),

IIIT Hyderabad

Motivation

- Consider a 2 hour, full HD video (resolution of 1920×1080)
- The storage space required per frame :$1920 \times 1080 \times 24$ bits = 6.22 MB
- Space required per second: $1920 \times 1080 \times 24 \times 30$ bits
- Space required for entire movie: $1920 \times 1080 \times 24 \times 30 \times 2 \times 60 \times 60$ bits = $1920 \times 1080 \times 3 \times 30 \times 2 \times 60 \times 60$ bytes = 1.34×10^{12} bytes= **1340 GB**
- To put it on a 25 GB blu ray disc: required compression factor = **53.6**

Redundancy

- Coding redundancy
- Spatial and Temporal redundancy
- Irrelevant Information (often perceptually irrelevant)

Formally, Redundancy can be measured as : $R = 1 - 1/C$

Compression is all about exploiting these redundancies!

Coding redundancy

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

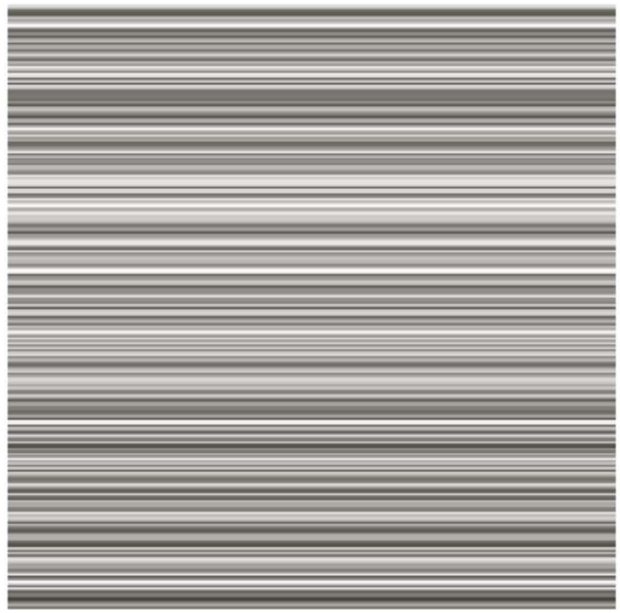
Average encoding length?

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p(r_k)$$

Code 2

$$L_{avg} = 0.25(2) + 0.47(1) + 0.25(3) + 0.03(3) = 1.81 \ bits$$

Spatial and temporal redundancy

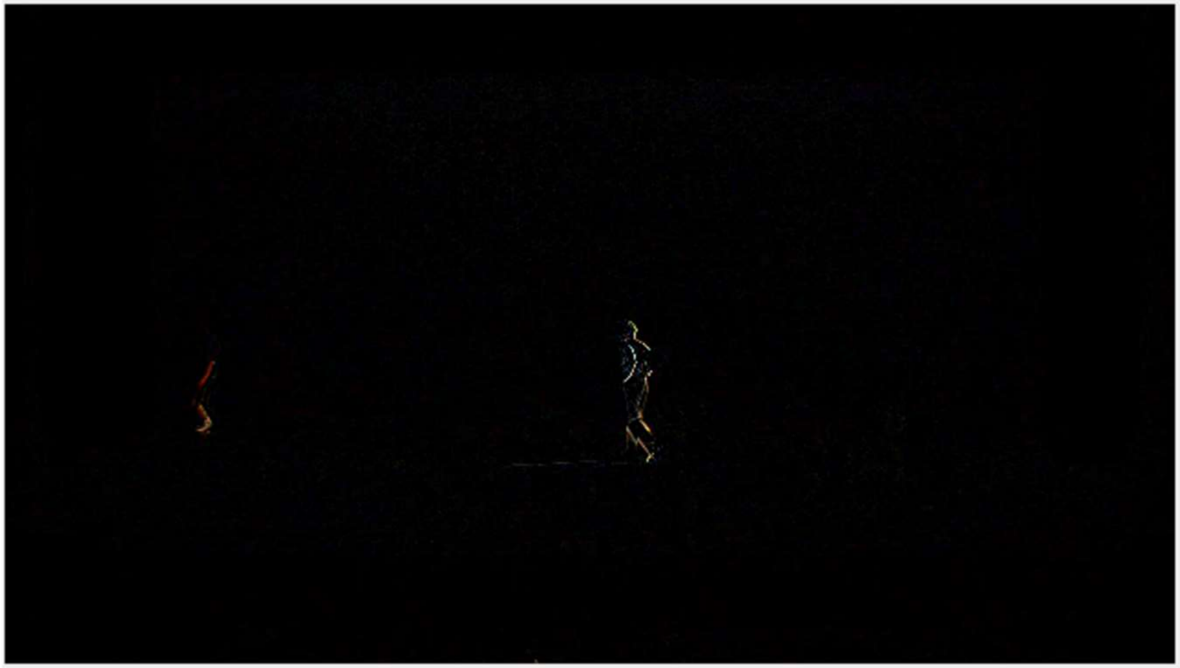


Spatial and temporal redundancy

frame t

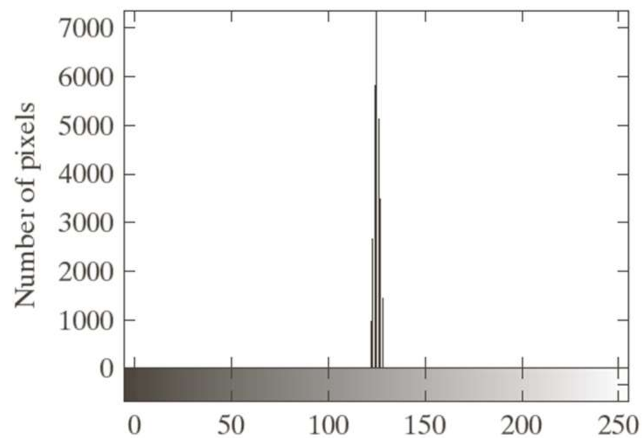
frame t+1

Spatial and temporal redundancy

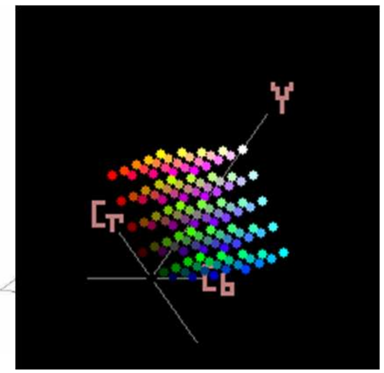


Irrelevant information or perceptual redundancy

- Not all visual information is perceived by eye/brain, so throw away those that are not



Irrelevant information or perceptual redundancy

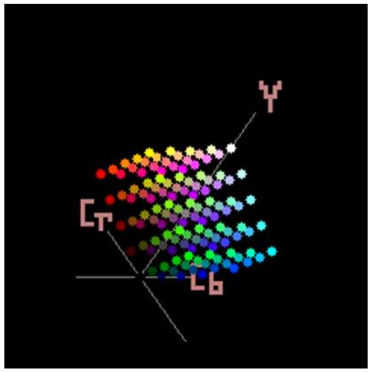


Y

Cb

Cr

Y-Cb-Cr Color Space



- Convert RGB to one luma and two chroma components

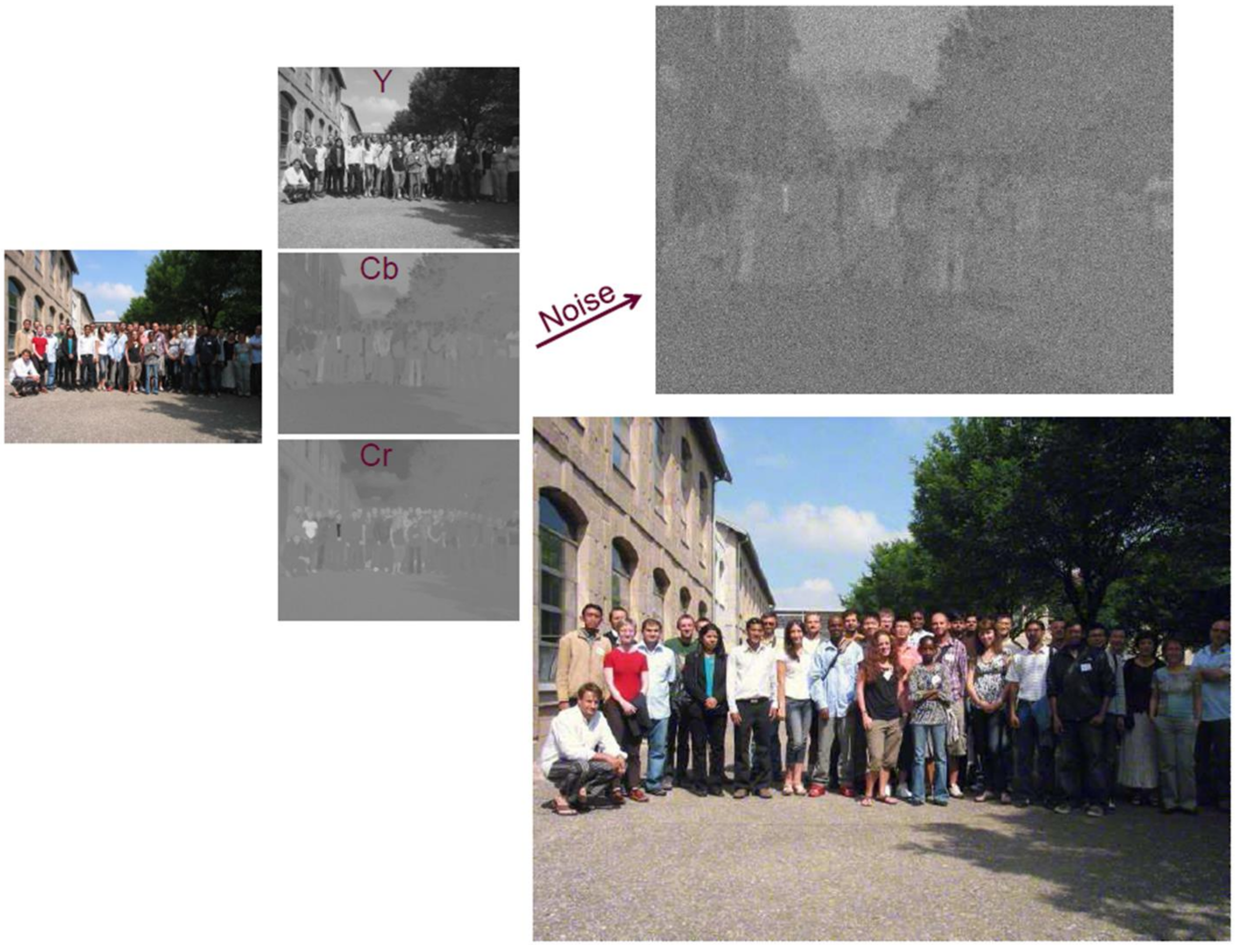
- Conversion from RGB:
  - Y=0.299(R-G) + G + 0.114(B-G)
  - Cb=0.564(B-Y)
  - Cr=0.713(R-Y)

- The Matrix form:

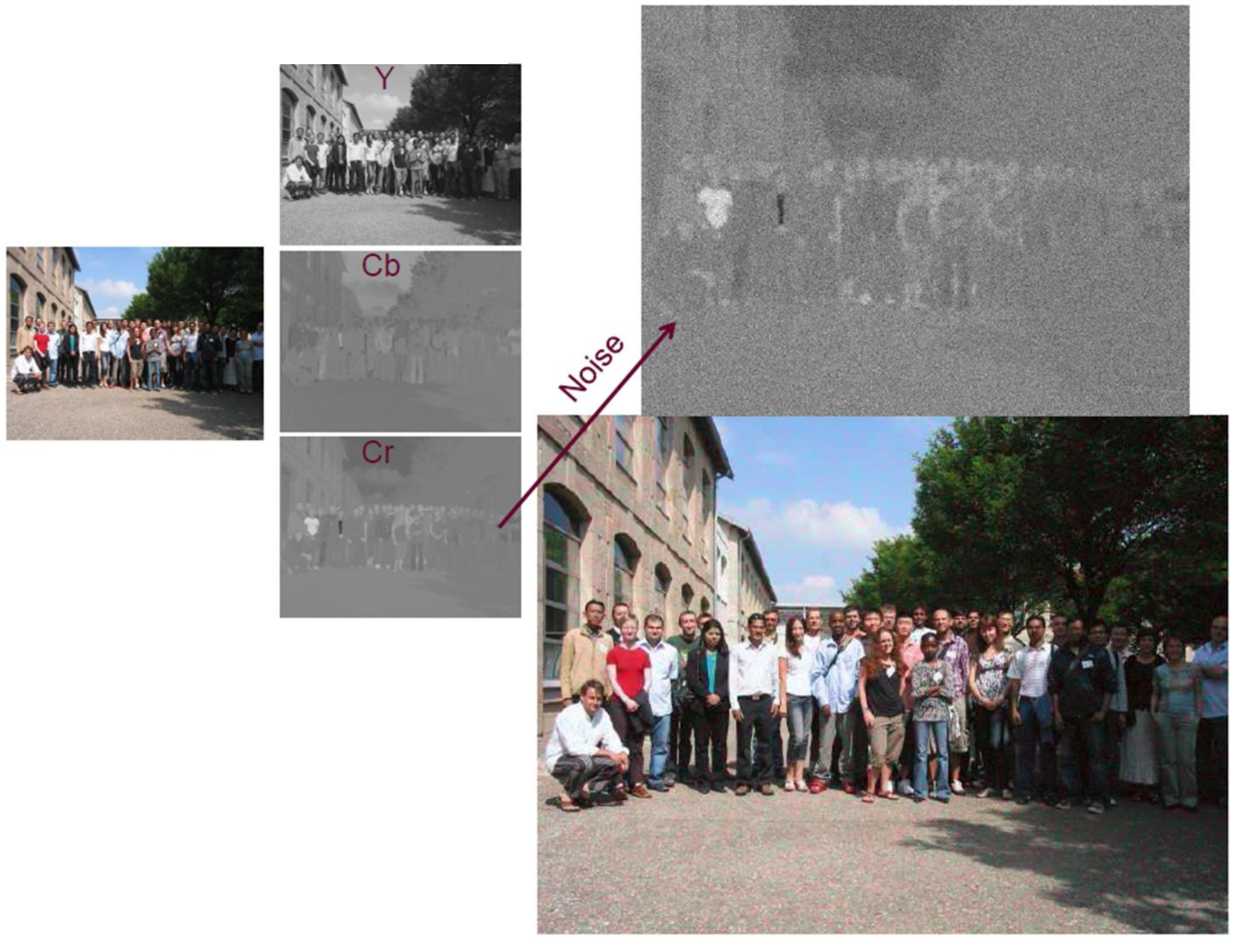
$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168636 & 0.232932 & -0.064296 \\ 0.499813 & -0.418531 & -0.081282 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Irrelevant information or perceptual redundancy

Irrelevant information or perceptual redundancy



Irrelevant information or perceptual redundancy



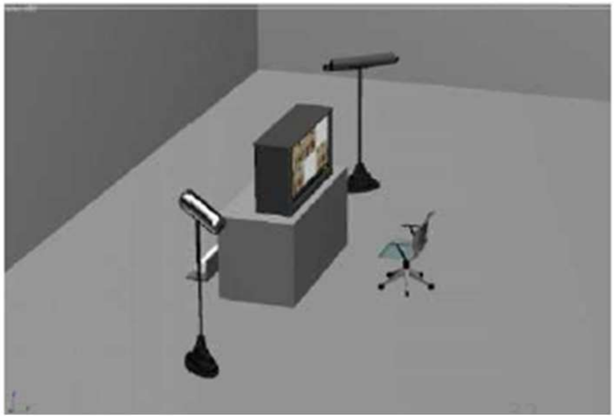
Compression types and evaluations

Two kinds:

1. Lossless
2. Lossy

Quality measurement: judged by human viewers

- Five scale system on the degree of impairment
  1. Impairment is not noticeable
  2. Impairment is just noticeable
  3. Impairment is definitely noticeable, but not objectionable
  4. Impairment is objectionable
  5. Impairment is extremely objectionable



Advantages: relies on HVS

Drawbacks: time, viewing conditions, viewers?

Quality measurement: Signal to noise ratio

$$e(x, y) = f(x, y) - g(x, y).$$

$$E_{\mathrm{ms}} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} e(x, y)^2$$

Fidelity Criteria

1 $$SNR_{ms} = 10 \log_{10} \left(\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(x, y)^2}{MN \cdot E_{\mathrm{ms}}} \right)$$

2 $$PSNR = 10 \log_{10} \left(\frac{255^2}{E_{\mathrm{ms}}} \right)$$

Information theory: Self energy

- Information is defined as knowledge, fact, and news
- It can be measured quantitatively

- The carriers of information are symbols. Consider a symbol with an occurrence probability p. The amount of information contained in the symbol is defined as:

$$I = \log_2 \frac{1}{p} \text{ bits} \quad \text{or} \quad I = -\log_2 p$$

Information theory: Entropy

- Consider a source that contains L possible symbols (events) $\{s, i = 0,1,2, \dots, L-1\}$
- With corresponding occurrence probabilities defined as $\{p_i, i = 0,1,2, \dots, L-1\}$

- **Entropy:** *"average information per source output"*

$$H = -\sum_{i=0}^{L-1} p_i \log_2 p_i$$

log (0.47) = -1.09
log(0.03) = -5.06

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

Entropy: $H = 1.6614\ bits/pixel$

Information theory: Shannon’s theorem

- Shannon’s lossless source coding theorem states that for a *discrete*, *memoryless*, *stationary* information source, the minimum bit rate required to encode a symbol on average is equal to the entropy of the source.

- In other words: we can’t do better than the entropy

- However, in practice, entropy of an image does not necessarily convey visual informative-ness of the image content.

- Entropy provides a lower bound on compression that can be achieved when coding statistically independent pixels but it breaks down when pixels of an image are correlated (often the case).

Validity of the code?

- Lets take an example

Symbol	Probability	Code1	Code2	Code3	Code4
s1	1/2	0	0	0	0
s2	1/4	0	1	10	01
s3	1/8	1	00	110	011
s4	1/8	10	11	111	0111

Image Compression: Overview

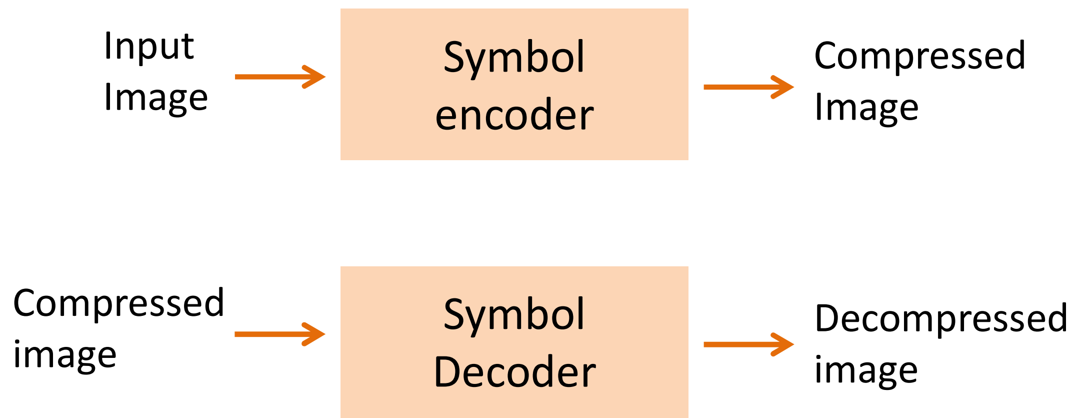


Image Compression: Overview

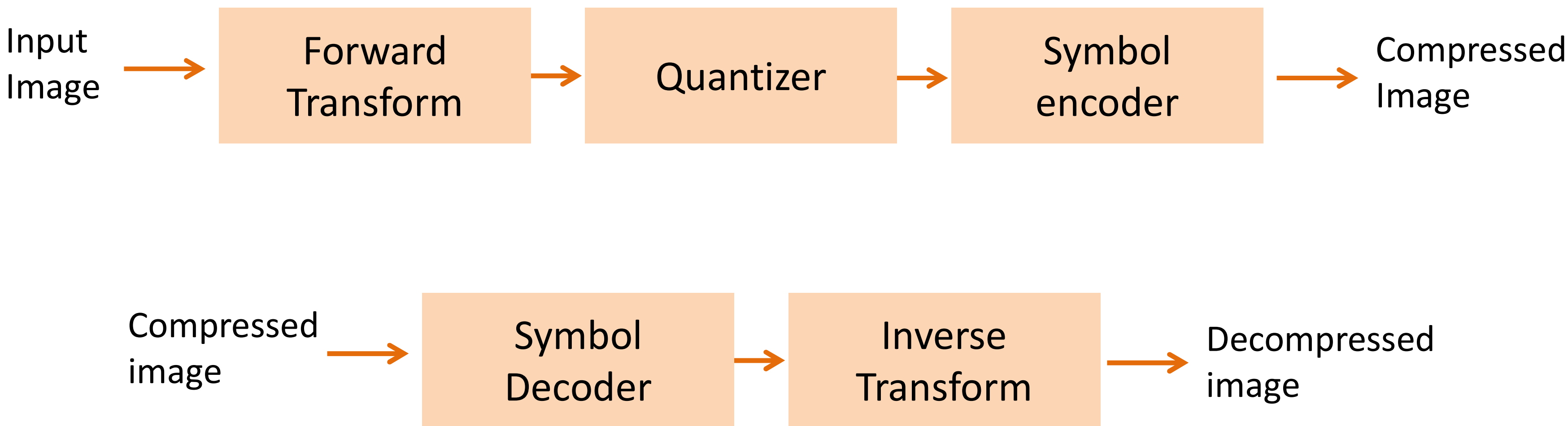


Image Compression: Overview

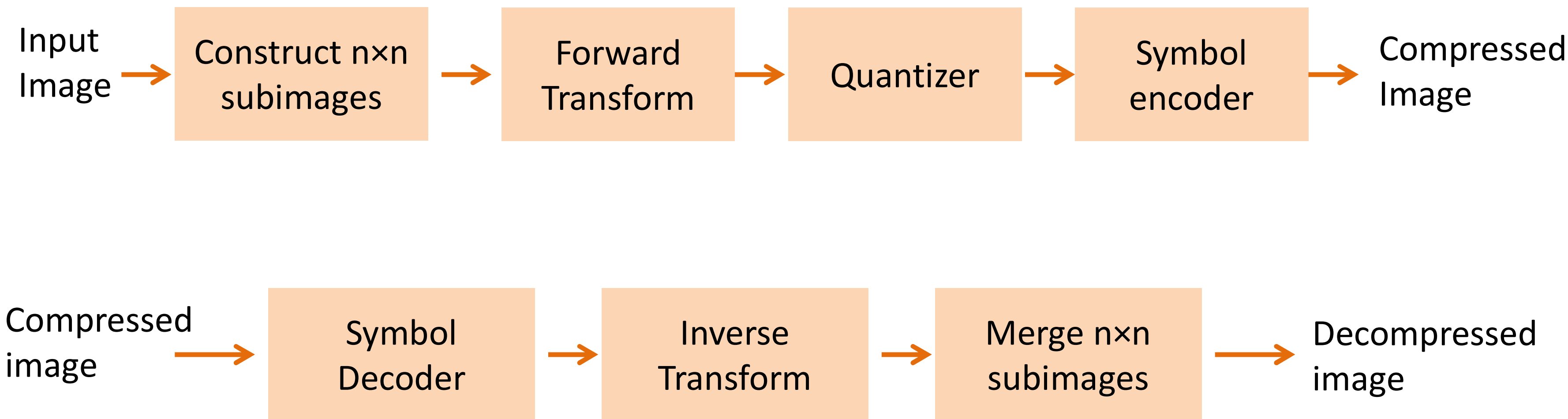
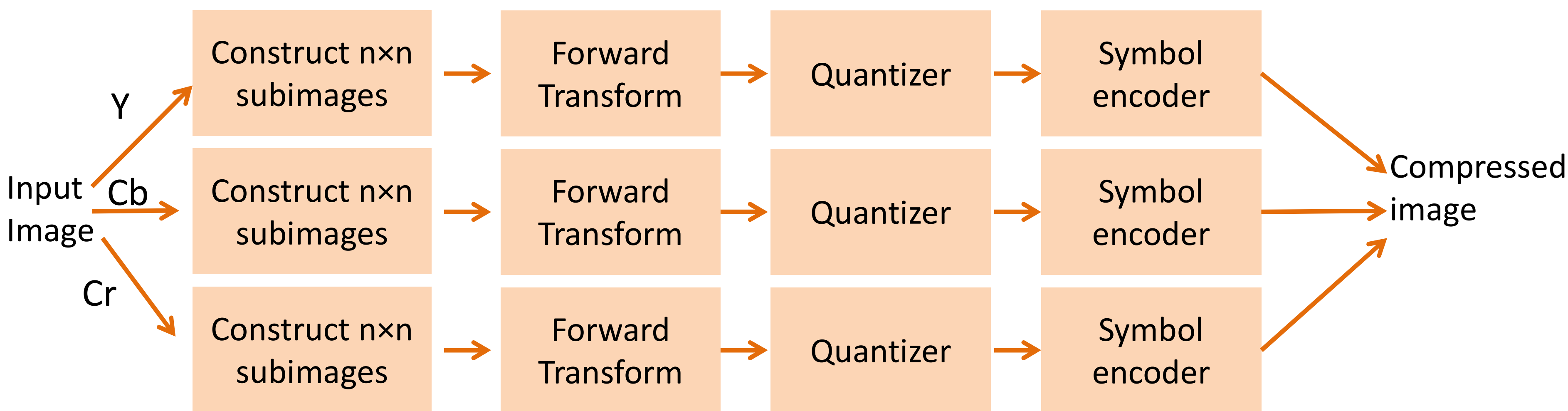
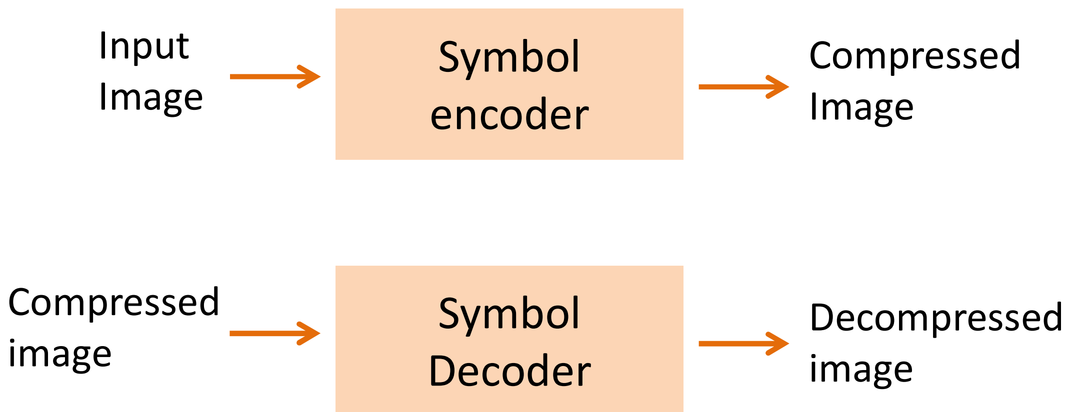


Image Compression: Overview



Lossless compression

Lets begin with simplest case: Lossless compression



Lossless compression : Huffman coding

- **Huffman coding** is a popular entropy encoding algorithm used for lossless data compression.
  - refers to the use of a variable length code table for encoding a source symbol.
  - the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol.

- Removes coding redundancy by yielding smallest possible number of code symbols per source symbol.

- Yields optimal **prefix-code** for a set of symbols and probabilities subject to the constraint that the symbols be coded one at a time.

Lossless compression : Huffman coding

- **Step 1: Source reduction**
  - Build a Huffman Tree

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

- **Step 2: Code assignment**

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Decode this stream : 010101110001011 $a_3 a_1 a_2 a_2 a_6 a_5$

Lossless compression : Run Length coding

- Quick example:

  0000000000000001111111111100000000111111 → 40 bits

- 15 0’s, 11 1’s , 8 0’s, 6 1’s

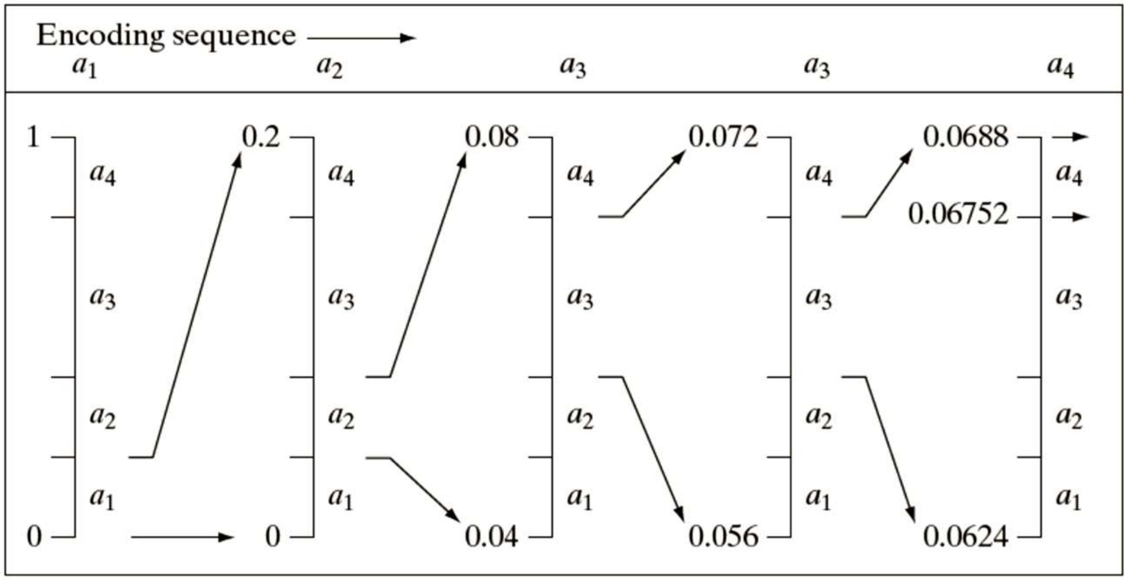
How many bits to store the count?

Give a scenario where run length coding will be extremely effective?

Lossless compression : Arithmetic coding

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

Input sequence: $a_1 a_2 a_3 a_3 a_4$



Final code: 0.068 (could be anything between the computed range)

3 decimal digits for 5 symbols = 3/5 digits per symbol

How many bits per symbol?

Lossless compression : Arithmetic coding

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

Another sequence: $a_1 a_1\ a_3$