Image Processing and Computer Vision

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

IMAGE COMPRESSION

- **Definition:** Compression means storing data in a format that requires less space than usual.
- Data compression is particularly useful in communications because it enables devices to transmit the same amount of data in fewer bits.
- The bandwidth of a digital communication link can be effectively increased by compressing data at the sending end and decompressing data at the receiving end.
- There are a variety of data compression techniques, but only a few have been standardized.

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

IMAGE COMPRESSION

Image processing is the application of 2D signal processing methods to images

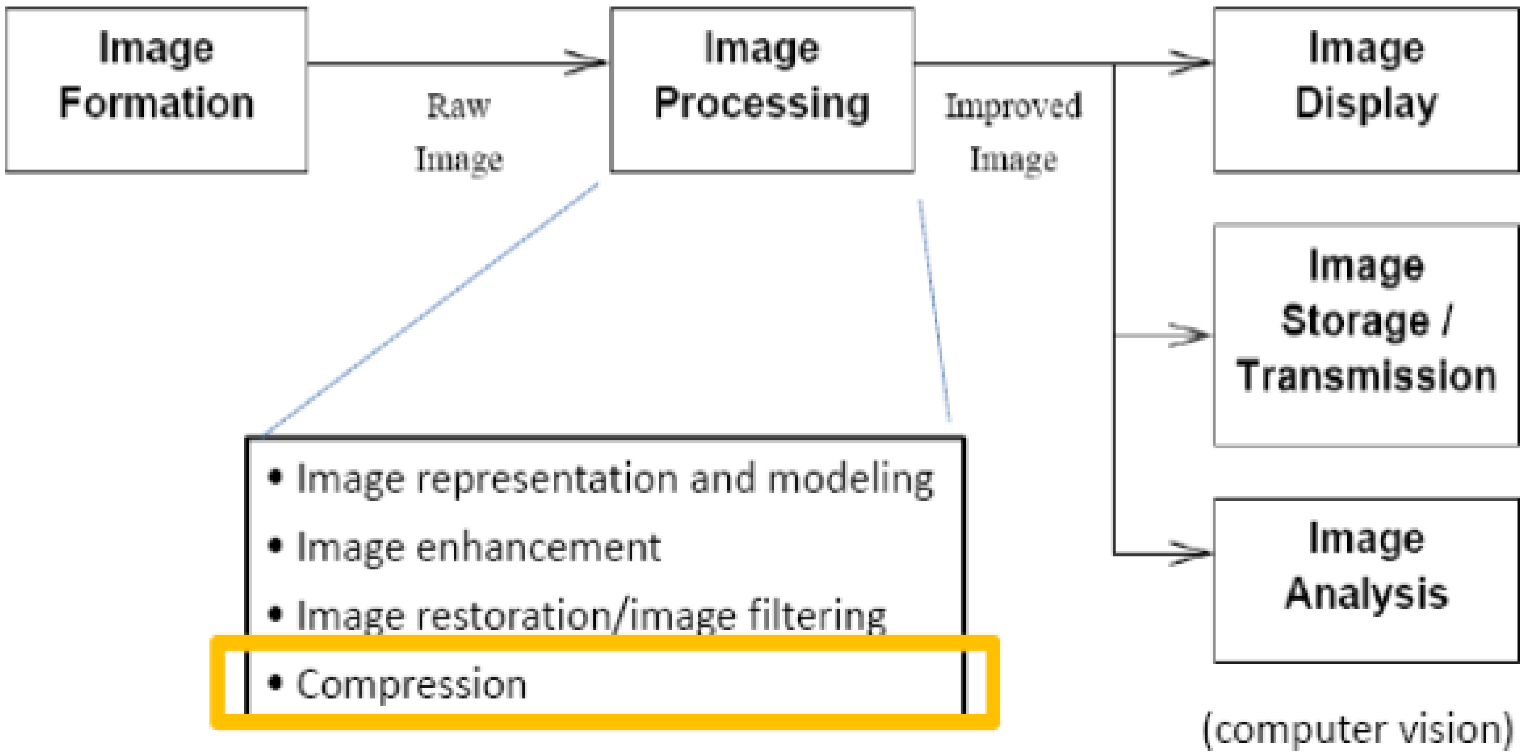


IMAGE COMPRESSION

- Image and video data compression refers to a process in which the amount of data used to represent image and video is reduced to meet a bit rate requirement (below or at most equal to the maximum available bit rate).

➢ The quality of the reconstructed image or video satisfies a requirement for a certain application.

➢ The complexity of computation involved is affordable for the application.

- **The basis of compression process is removal of redundant data.**

WHAT IS THE NEED FOR COMPRESSION

- In terms of storage, the capacity of a storage can be effectively increased.
- In terms of communications, the bandwidth of a digital communication link can be effectively increased .
- At any given time, the ability of the internet to data is fixed. Thus, if data can effectively be compressed wherever possible, significant improvement of data throughput can be achieved.
- Many files can be combined into one compressed documents making sending easier.

NEED FOR COMPRESSION

- To minimize the Storage Space
- To enable higher rate of Data transfer
- There is still a need to develop newer and faster algorithms adapted to image data

Types of Data Compression

- There are two main types of data compression : Lossy and Lossless.
- In Lossy data compression the message can never be recovered exactly as it was before it was compressed.
- In a Lossless data compression file the original message can be exactly decoded.
- Lossless compression is ideal for text.
- Huffman coding is type of lossless data compression.

TYPES OF COMPRESSION

- There are two types of compression

❖ **Lossless compression-** It can recover the exact original data after compression.

➢ It is used mainly for compressing database records, spreadsheet or word processing files, medical imaging, where exact replication of the original is essential.

❖ **Lossy compression-** It will result in a certain loss of accuracy for a substantial increase in compression.

➢ Lossy compression is more effective when used to compress graphic images and digitized voice where losses outside visual or aural perception can be tolerated.

TYPES OF COMPRESSION: LOSSLESS

Basic Idea:

- Convert each gray-value (symbol) into a binary sequence (codeword)
- Codewords may have different lengths for different gray-values.

Exploit statistical redundancy

- Assign shorter binary sequences to more common gray levels, and longer codewords to less common ones.
- How do we make this assignment quantitatively? Use the notion of Entropy.

TYPES OF COMPRESSION: LOSSLESS

❖ Perfectly reproduce the image at the decoder(MSE=0, PSNR=Inf)

❖ Reduction in bit rate is usually limited.

❖ Can not guarantee a fixed rate.

❖ Applications: –Fax Machines–Medical Images

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

LOSSLESS COMPRESSION METHODS

- Run length Encoding (RLE)
- Huffman coding
- Arithmetic coding
- LZ and LWZ coding
- Predictive coding
- Variable length coding etc

TYPES OF COMPRESSION: LOSSY

- ❑ Idea: some deviation of decompressed image from original is often acceptable.
- ❑ Human visual system might not perceive loss or tolerate it. Thus, omit irrelevant details that humans cannot perceive.
- ❑ No need to represent more than the visible resolution in:
  - ❖ Space
  - ❖ Time
  - ❖ Brightness
  - ❖ color

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

HOW TO ACHIEVE COMPRESSION

- Minimizing the Redundancy in the Image.
- Types of Redundancy-Mainly two types

❖ Statistical Redundancy- Interpixel Redundancy & Coding

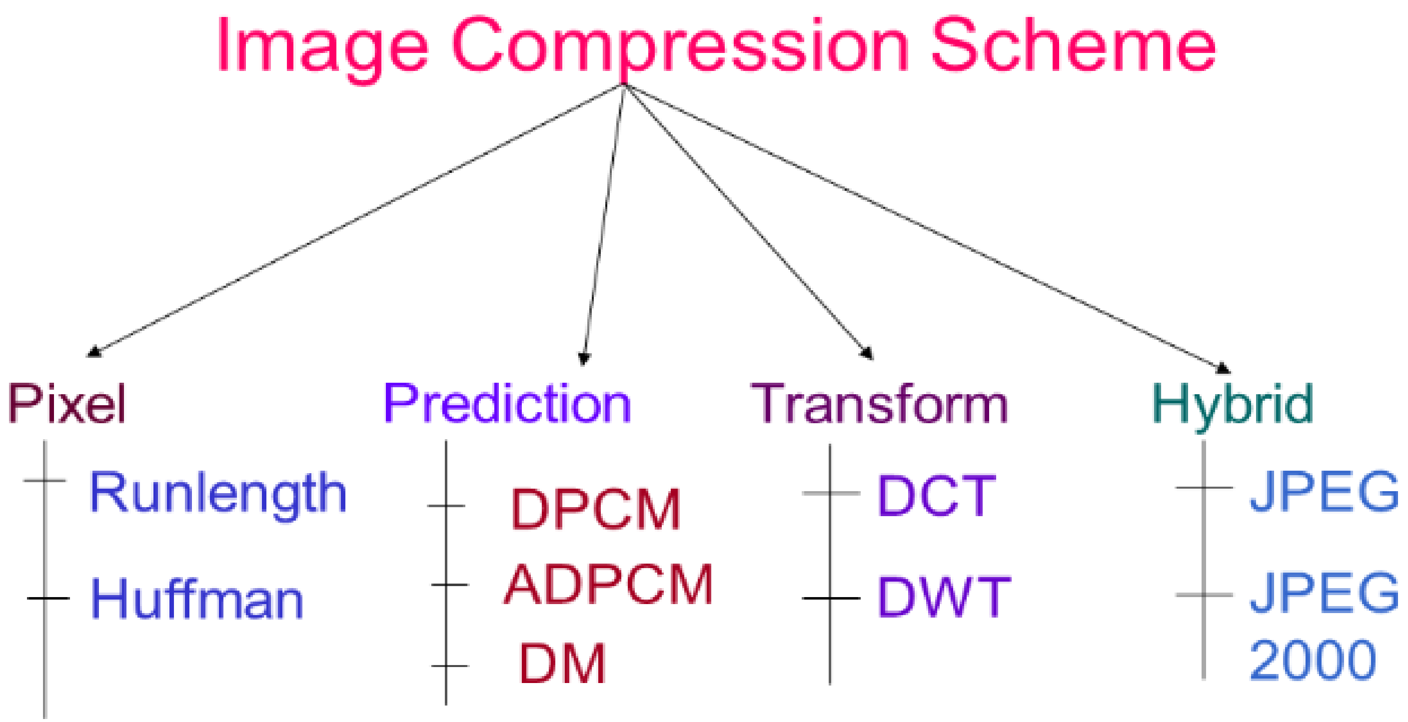
- Redundancy.

➢ Interpixel Redundancy-Spatial Redundancy & Temporal Redundancy.

❖ Psychovisual Redundancy.

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

IMAGE COMPRESSION TECHNIQUES



Compression Algorithms

- Huffman Coding
- Run Length Encoding
- Shift Codes
- Arithmetic Codes
- Block Truncation Codes
- Transform codes
- Vector Quantization

Huffman Coding

- Huffman coding is a popular compression technique that assigns variable length codes (VLC) to symbols, so that the most frequently occurring symbols have the shortest codes.
- On decompression the symbols are reassigned their original fixed length codes.
- The idea is to use short bit strings to represent the most frequently used characters
- and to use longer bit strings to represent less frequently used characters.
- That is, the most common characters, usually space, e, and t are assigned the shortest codes.
- In this way the total number of bits required to transmit the data can be considerably less than the number required if the fixed length ASCII representation is used.
- A Huffman code is a binary tree with branches assigned the value 0 or 1.
-

Huffman Algorithm

- To each character, associate a binary tree consisting of just one node.
- To each tree, assign the character's frequency, which is called the tree's weight.
- Look for the two lightest-weight trees. If there are more than two, choose among them randomly.
- Merge the two into a single tree with a new root node whose left and right sub trees are the two we chose.
- Assign the sum of weights of the merged trees as the weight of the new tree.
- Repeat the previous step until just one tree is left.

Advantages & Disadvantages

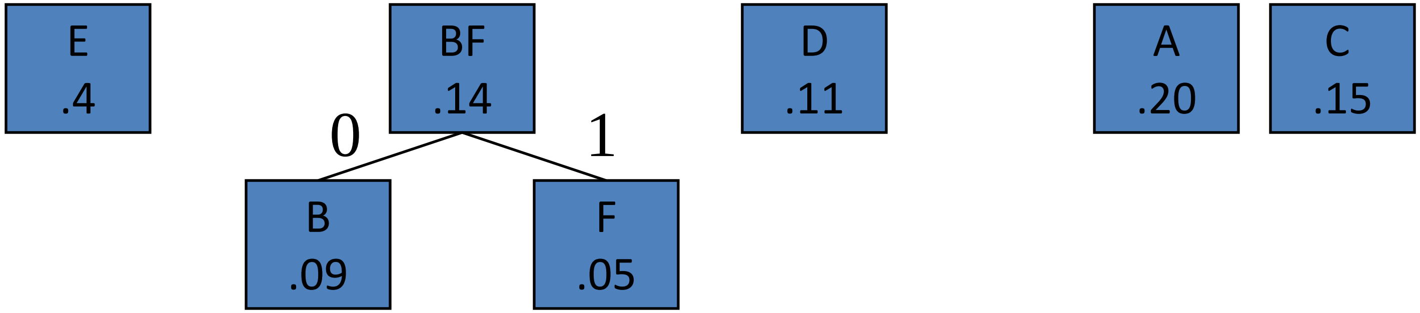
- The problem with Huffman coding is that it uses an integral number of bits in each code.
- If the entropy of a given character is 2.5 bits,the Huffman code for that character must be either 2 or 3 bits , not 2.5.
- Though Huffman coding is inefficient due to using an integral number of bits per code , it is relatively easy to implement and very efficient for coding and decoding.
- It provides the best approximation for coding symbols when using fixed width codes.

Huffman Coding Example

- Character frequencies
  - A: 20% (.20)
  - B: 9% (.09)
  - C: 15%
  - D: 11%
  - E: 40%
  - F: 5%

- No other characters in the document

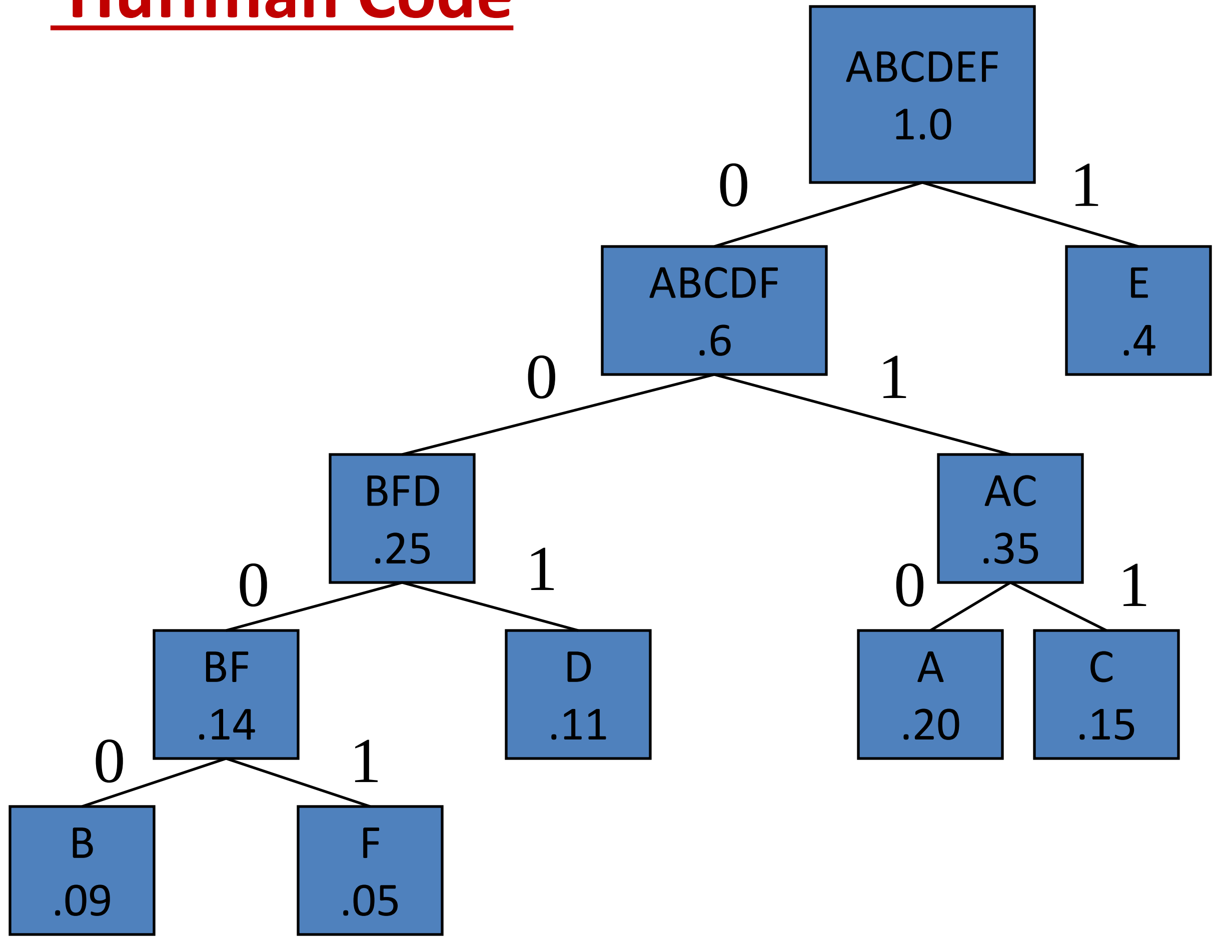
Huffman Code



Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

Huffman Code

- Codes
  - A: 010
  - B: 0000
  - C: 011
  - D: 001
  - E: 1
  - F: 0001



- Note
  - None are prefixes of another

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

Huffman Coding

- TENNESSEE

```
        9
      0/   \1
     5     e(4)
   0/   \1
 s(2)   3
      0/   \1
   t(1)   n(2)
```

- ENCODING
- E : 1
- S : 00
- T : 010
- N : 011

Average code length = (1*4 + 2*2 + 3*2 + 3*1) / 9 = 1.89

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

Average Code Length

Average code length =

$\Sigma_{i=0,n}$ (length*frequency)/ $\Sigma_{i=0,n}$ frequency

= { 1(4) + 2(2) + 3(2) + 3(1) } /(4+2+2+1)

= 17 / 9 = 1.89

ENTROPY

Entropy is a measure of information content:the more probable the message, the lower its information content, the lower its entropy

Entropy = $-\Sigma_{i=1,n} (p_i \log_2 p_i)$

(p - probability of the symbol)

$$= - (0.44 * \log_2 0.44 + 0.22 * \log_2 0.22 + 0.22 * \log_2 0.22 + 0.11 * \log_2 0.11)$$

$$= - (0.44 * \log 0.44 + 2(0.22 * \log 0.22 + 0.11 * \log 0.11) / \log 2$$

$$= 1.8367$$

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune