اللهم صل وسلم وبارك على سيدنا محمد وعلى
آله وصحبة وسلم تسليماً كثيراًطيباً مباركاً فيه

File Organization

Dr \ Mohammed Ahmed Mahfouz

Doctor of Information Systems
Thebes Higher Institute for Management and Information Technology

Organizing Files for Performance

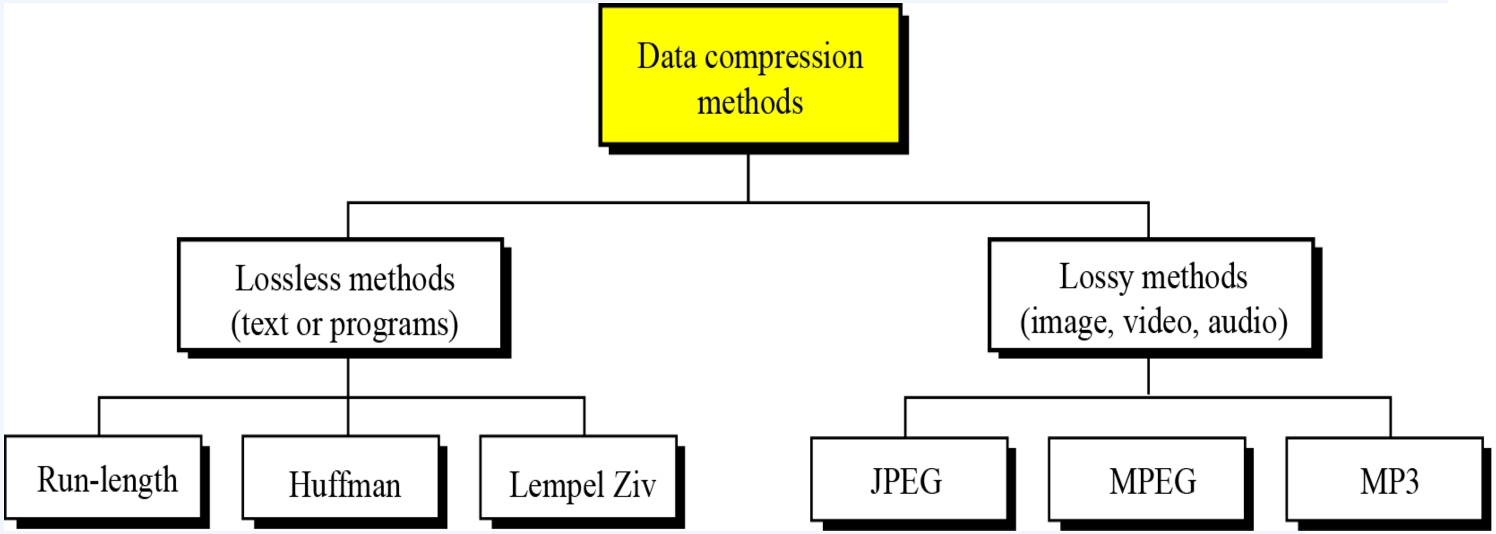
Lecture No. 9

Contents

Data Compression Methods

- Data compression is about storing and sending a smaller number of bits.

- There're two major categories for methods to compress data: lossless and lossy methods



Lossy Compression Methods

Lossy Compression Methods

What is Lossy Compression?

- It is techniques where **some data is permanently removed** to reduce file size.
- The removed data is typically something humans **cannot easily detect** (visual or auditory details).

Why can we accept this loss?

- The human eye cannot notice very small changes in images.
- The human ear cannot detect certain frequencies or quiet sounds masked by louder ones.

Advantages

- Requires **less storage**.
- Faster to process.
- Suitable for **media files**.

Lossy Compression Methods

- Used for compressing images and video files (our eyes cannot distinguish subtle changes, so lossy data is acceptable).
- These methods are cheaper, less time and space.

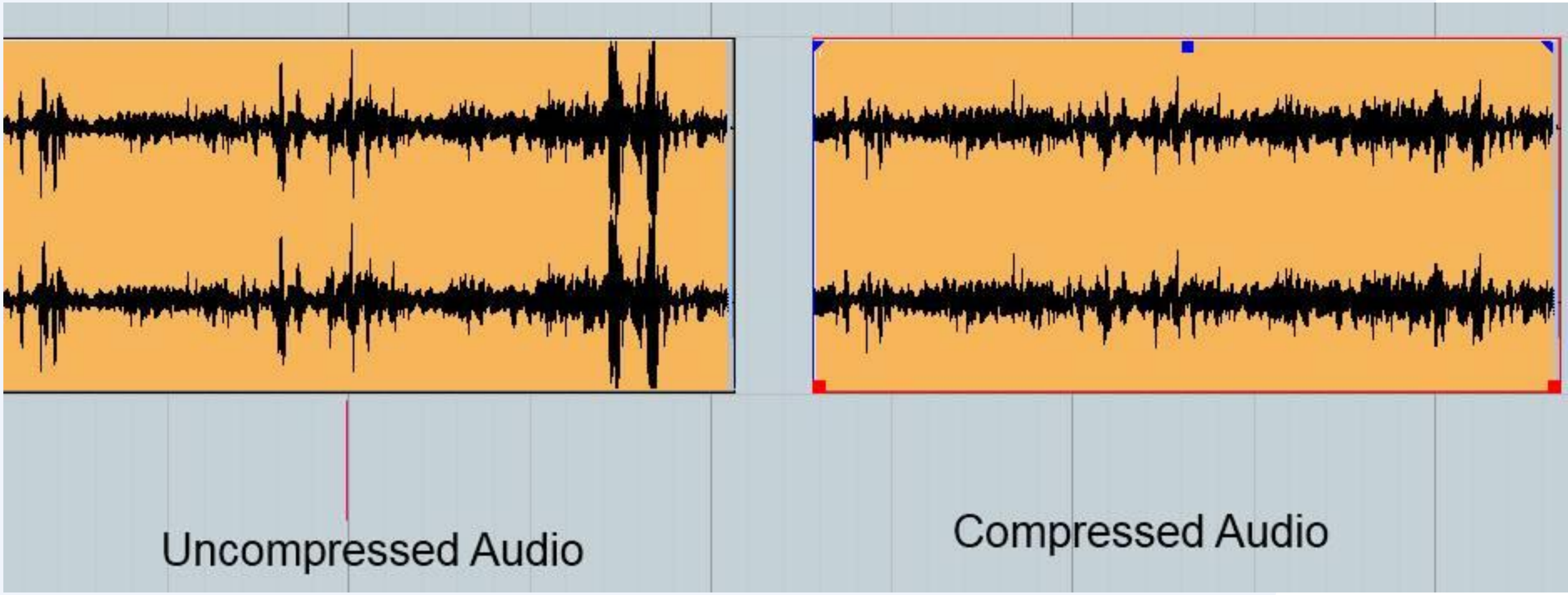
- Several methods:
  - JPEG (Joint Photographic Experts Group): compress pictures and graphics
  - MPEG (Moving Picture Experts Group): compress video
  - MP3 (MPEG-1 Audio Layer 3) : compress audio

Lossy Compression Methods

Compressed by 90%- 5.93 KB

Uncompressed- 57.5 KB

Lossy Compression Methods



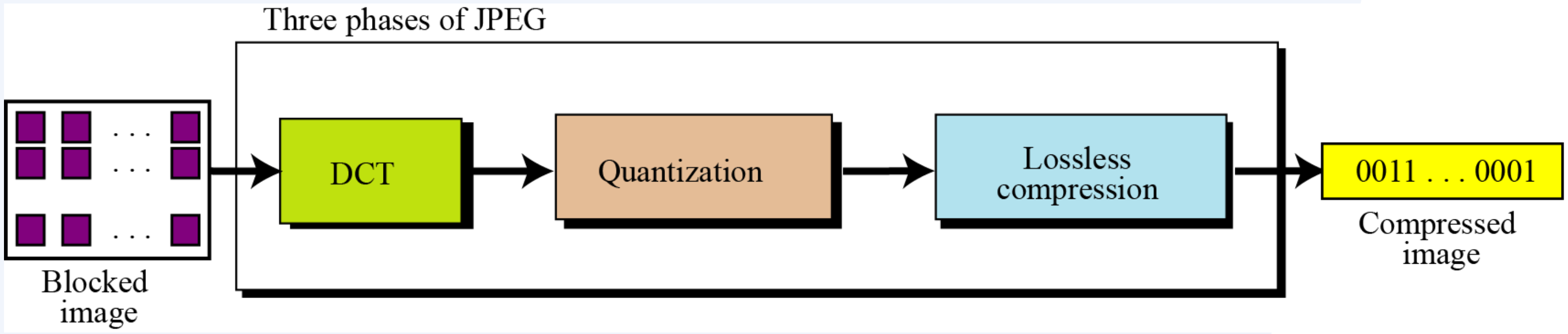
JPEG Encoding

JPEG Encoding

- Used to compress pictures and graphics.

Step 1: In JPEG, a grayscale picture is divided into 8x8 pixel blocks to decrease the number of calculations.

- **Basic idea:**
  1. Change the picture into a linear (vector) sets of numbers that reveals the redundancies.
  2. The redundancies is then removed by one of lossless compression methods.



JPEG Encoding- DCT

Step 2: **DCT**: Discrete Concise Transform

1) Definition of DCT-II (The Formula Used in JPEG)

The most commonly used form is the **Type-II Discrete Cosine Transform (DCT-II)** with a normalization factor α(k)

1-D Formula (N points):

$$X_k = \alpha(k) \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi(2n+1)k}{2N}\right), \quad k = 0, \ldots, N-1$$

- where the normalization factor is: $\alpha(k) = \begin{cases} \frac{1}{\sqrt{N}}, & k = 0 \\ \sqrt{\frac{2}{N}}, & k > 0 \end{cases}$

Inverse Transform (IDCT):

$$x_n = \sum_{k=0}^{N-1} \alpha(k) X_k \cos\left(\frac{\pi(2n+1)k}{2N}\right), \quad n = 0, \ldots, N-1$$

JPEG Encoding- DCT

2) 2-D Formula (For Images, Used in JPEG on 8×8 Blocks)

The **2-D DCT** is **separable**, so for an N×N image f(x,y):

$$X_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left(\frac{\pi(2x+1)u}{2N}\right) \cos\left(\frac{\pi(2y+1)v}{2N}\right)$$

Inverse 2-D DCT:

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) X_{u,v} \cos\left(\frac{\pi(2x+1)u}{2N}\right) \cos\left(\frac{\pi(2y+1)v}{2N}\right)$$

with the same definition of α(·) as above.

- **In JPEG:** This formula is applied to each 8×8 block (i.e., N=8).

JPEG Encoding- DCT

- **DCT**: Discrete Concise Transform
- DCT transforms the 64 values in 8x8 pixel block in a way that the relative relationships between pixels are kept but the redundancies are revealed.
- Example:

A gradient grayscale

8×8

Block

20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90
20	30	40	50	60	70	80	90

P (x, y)

400	–146	0	–31	–1	3	–1	–8
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
0	0	0	0	0	0	0	0

T (m, n)

JPEG Encoding- DCT

Transform the block using DCT

DCT converts the 64 pixel values (spatial domain) into frequency values.

Low frequencies = smooth areas

High frequencies = fine details/noise

Goal:

Keep the important (low-frequency) values, reduce the unimportant (high-frequency) values.

Quantization & Compression

Step 3: **Quantization:** This is the **lossy step**.

Each DCT value is divided by a number (from the quantization table), and the decimal part is dropped.

- After T table is created, the values are quantized to reduce the number of bits needed for encoding.
- Quantization divides the number of bits by a constant, then drops the fraction. This is done to optimize the number of bits and the number of 0s for each particular application.

DCT Value	Divide By	Result
400	10	40
5	3	1
2	3	0

- Most high-frequency values become **0**, which is important for compression.

Quantization & Compression

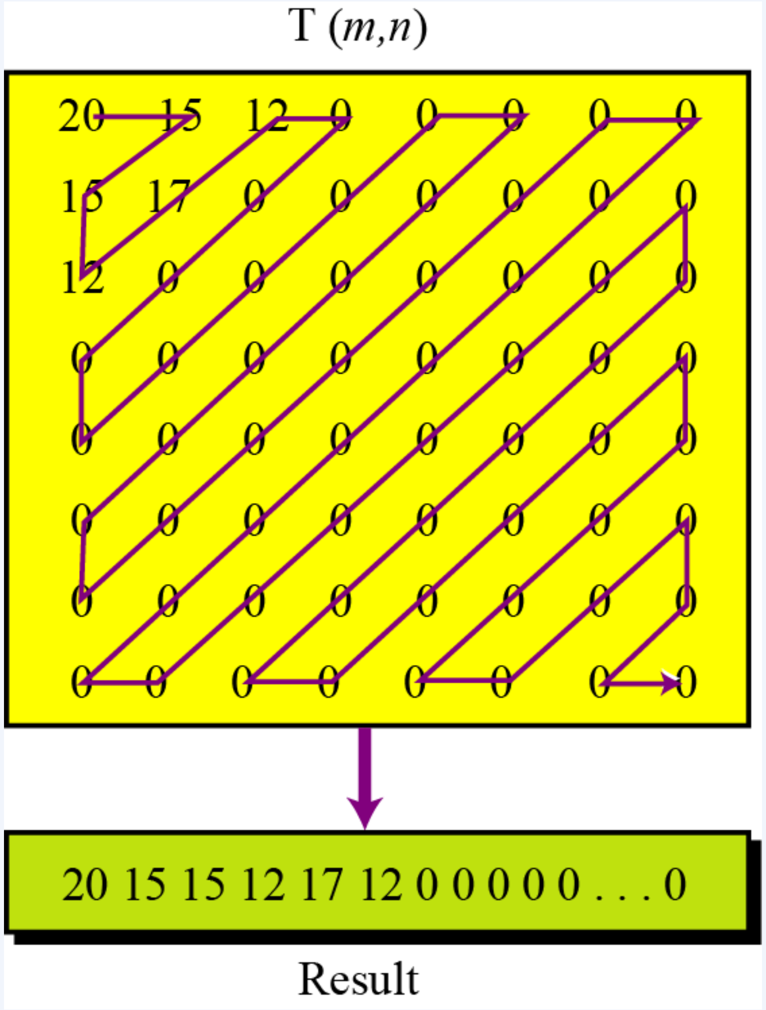
Step 4: **Compression:**

- The 8×8 block is read in a zigzag pattern to group all zeros together.
- Quantized values are read from the table and redundant 0s are removed.
- To cluster the 0s together, the table is read diagonally in an zigzag fashion. The reason is if the table doesn't have fine changes, the bottom right corner of the table is all 0s.

Step 5: **Lossless stage (RLE or Huffman)**

- JPEG applies **Run-Length Encoding (RLE)** to compress long sequences of zeros.
- Then uses **Huffman Coding.**

JPEG Encoding



MPEG Encoding

MPEG Encoding

- Used to compress video.

A video is simply a **sequence of images** (frames).

Basic idea:

- Each video is a rapid sequence of a set of frames.
- Each frame is a spatial combination of pixels, or a picture.
- Compressing video =

  spatially compressing each frame

  +

  temporally compressing a set of frames.

MPEG Encoding

- **1. Spatial Compression (Inside each frame)**

  Each frame is compressed like a JPEG image.
- **2. Temporal Compression (Between frames)**

  This is the major saving.
- **Key idea:**

  Most frames are **very similar** to each other.
- Example:
  A person talking → only the **lips move**, the rest of the frame stays almost identical.
- MPEG compares frames:
  - If a region is unchanged, it is **not stored again**.
  - Only the **difference** is stored.

MPEG Encoding

Example (conceptual):

- Frame 1:

  Person sitting (mouth closed)

- Frame 2:

  Same image (mouth slightly open)

- Instead of storing the whole frame again, MPEG stores:

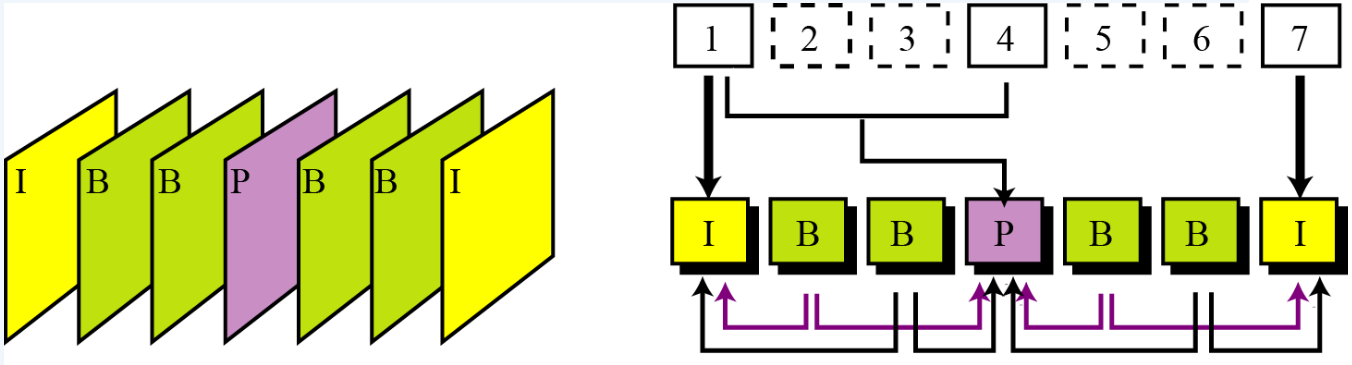
  Only the pixels around the mouth that changed

  This creates huge savings.

MPEG Encoding

- **Spatial Compression**
  - Each frame is spatially compressed by JPEG.

- **Temporal Compression**
  - Redundant frames are removed.
  - For example, in a static scene in which someone is talking, most frames are the same except for the segment around the speaker's lips, which changes from one frame to the next.



a. Frames

b. Frame construction

Audio Encoding

Audio Compression

- Used for speech or music
  - Speech: compress a 64 kHz digitized signal
  - Music: compress a 1.411 MHz signal

- Two categories of techniques:
  - Predictive encoding
  - Perceptual encoding

Audio Compression

1. Predictive Encoding (speech)

Stores only the **difference between samples**, not the full waveform.

Example:

If samples are: 100, 101, 102, 103, 103, 104

Differences are: +1, +1, +1, 0, +1

These numbers are smaller → easier to compress.

- Used in standards like:
  - GSM (13 kbps)
  - G.729 (8 kbps)
  - G.723.3 (6.4 kbps)

Audio Compression

2. Perceptual Encoding (MP3 for music)

MP3 removes parts of the sound humans **cannot hear**.

Examples:

- Very high frequencies
- Very quiet sounds hidden behind louder sounds (psychoacoustic model)
- This drastically reduces file size while keeping sound quality acceptable.

Example:

- Original CD audio: **1411 kbps**
- MP3 file: **128 kbps** or even **64 kbps**

Audio Encoding

Predictive Encoding

- Only the differences between samples are encoded, not the whole sample values.
- Several standards: GSM (13 kbps), G.729 (8 kbps), and G.723.3 (6.4 or 5.3 kbps)

Perceptual Encoding: MP3

- CD-quality audio needs at least 1.411 Mbps and cannot be sent over the Internet without compression.
- MP3 (MPEG audio layer 3) uses perceptual encoding technique to compress audio.



Thank You !