Data Compression Fundamentals

Prof. Dr. R. Logeswaran

Dean, School of Postgraduate Studies,
Asia Pacific University of Technology & Innovation (APU)
Technology Park Malaysia, Bukit Jalil
Kuala Lumpur

A · P · U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

IEEE (Malaysia)
Signal Processing Society

IEEE

@ UNITEN | *02 December 2016*

OVERVIEW

1. Introduction
2. Overview of Lossless Techniques
3. Overview of Lossy Techniques
4. Measuring Compression
5. Conclusion

1. INTRODUCTION

- 1.1 What is Data Compression
- 1.2 Motivation for Data Compression
- 1.3 Types of Data Compression
- 1.4 Lossy Compression
- 1.5 Lossless Compression
- 1.6 Concept of Models & Coding

1.1 WHAT IS DATA COMPRESSION

- Data Compression is the process of **reducing the number of bits** used to represent a certain amount of data through **removal of redundancy**.

- Highly correlated data generally have a lot of redundancy, wasting extra bits used for storage.

1.2 Motivation for Data Compression

- There are many benefits for compressing data.
- Among the most apparent of these is that :
  - **smaller size** - requires less physical storage, allows storage of more data at lower cost.
  - **increase in speed and efficiency -** dealing with a smaller amount of data in many operations such as in data backup.
  - **reduced transmission time and cost.**
  - **reduced resources requirements –** e.g. **memory** when **manipulating the data**.

1.2 Motivation for Data Compression (cont.)

- It must also be realised that the process of compressing (reducing) and decompressing (restoring) the data does **take up time and resources** as most of the better compression schemes store statistics and check the current input being examined with parts of data already compressed.

- These shortcomings are generally minute compared to the benefits of compression, making this field in high demand, especially with the ever increasing amount of data handled in modern times.

1.3 Types of Data Compression

- Minimised representation of data can be achieved either (Held, 1996):
  - **Logical-type compression**
    - generally applied to databases, where field sizes are reduced through encoding.
    - E.g., instead of allocating a large field size for long department names, a reference number could be used for each department.
  - **Physical-type compression**
    - achieved through the removal of redundancies in the data.

1.3 TYPES OF DATA COMPRESSION (CONT.)

- Most techniques are **physical compression** as it is a more general-purpose approach.
- Physical data compression techniques can be broadly classified into:
  - **Lossy** - Decompressed data is not an exact match of the source data (i.e., some loss of data during compression/decompression)
  - **Lossless** - Source data can be reconstituted exactly from the compressed data (i.e., no loss of data during compression / decompression)

1.4 LOSSY COMPRESSION

- Lossy compression techniques usually enable **better compression performance**, but are only applicable where the loss of data during compression and decompression has no significant effect.

- Examples of lossy compression includes compression where a certain amount of precision loss in **audio** and **video** is acceptable as the human visual and auditory faculties are not too sensitive to small losses and are able to compensate for the loss.

1.4 LOSSY COMPRESSION (CONT.)

- Some popular lossy techniques include certain types of :
  - Transformation coding (e.g. FFT, DCT), Layer coding (e.g. sub-band coding, subsampling, wavelets)
  - Vector Quantization.

- Common products of lossy encoding include JPEG, MPEG, H.261, DVI.

1.5 LOSSLESS COMPRESSION

- Lossless encoding attempts to minimise the size of the representation **without losing any information**.
- Decompression of losslessly compressed data will reproduce an exact match of the source data.
- Lossless compression is needed when the data needs to be precise and any loss in accuracy would be significant (e.g. when processing telemetry data from a satellite, or compressing binary files.

1.5 LOSSLESS COMPRESSION (CONT.)

- Examples of techniques for lossless compression include :
  - Statistical encoding (e.g. Huffman Coding),
  - Dictionary and substitution based coding (e.g. Lempel-Ziv string encoding and pattern substitution),
  - Suppression methods (e.g. run-length encoding).

1.6 Concept of Models & Coding

- **Data Compression = Modelling + Coding**

  to transform an input stream of symbols into output codes.

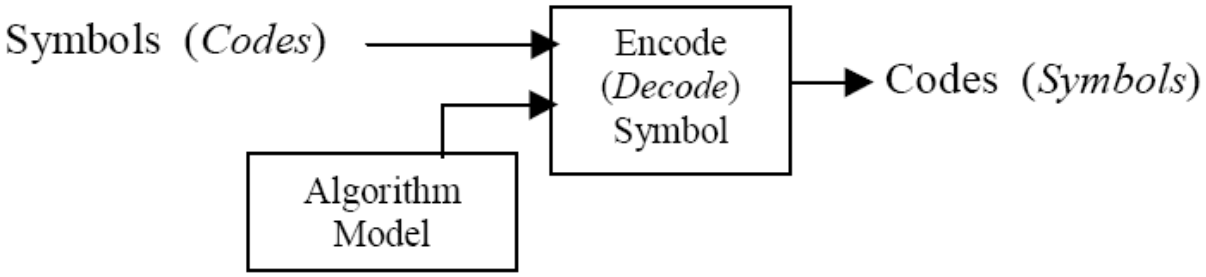
- A **model** is a collection of **data and rules**, and can be either **fixed or adaptive**.
  - Fixed models use pre-set information throughout the compression process, whereas adaptive models make adjustments to suit the pattern of the actual data at run-time.

1.6 Concept of Models & Coding (cont.)

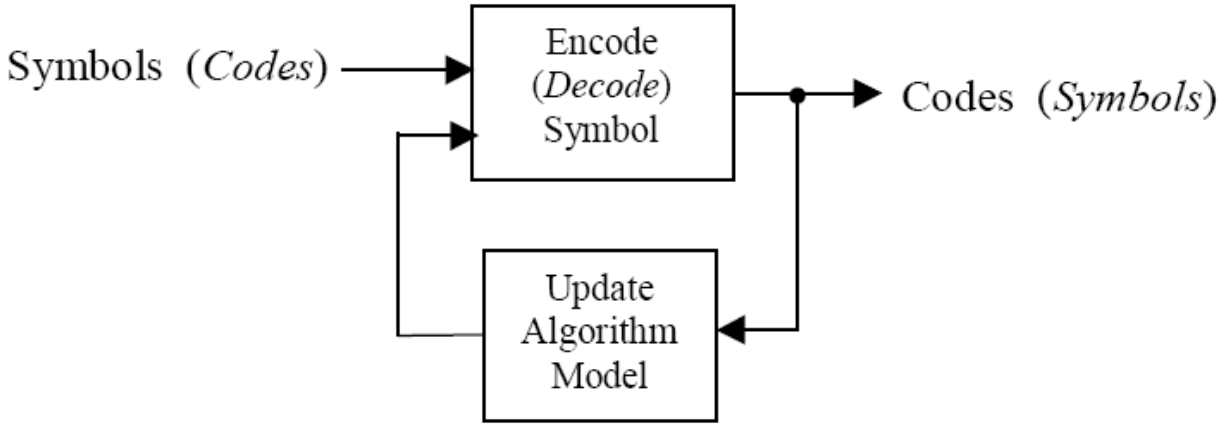
- A **coder** implements the algorithm that transforms the input into output, based on information provided by the model.

- As an example, Huffman coding uses probabilities supplied by a statistical model to transform each input symbol into its corresponding output code such that symbols with higher probabilities are encoded using a shorter bit sequence.

1.6 Concept of Models & Coding (cont.)



Fixed compression (*decompression*)



Adaptive compression (*decompression*)

1.6 Concept of Models & Coding (cont.)

- To illustrate fixed and adaptive models, lets take an example (see next slide):
  - The initial state denotes four input characters and the corresponding output Huffman codes.
  - Let the input stream be the character sequence X_2 X_4 X_2.
  - Using a fixed model, the output is the bit sequence 10 111 10.
  - Using the same (Huffman) coding scheme, but changing the model to an adaptive one, a shorter output stream of 10 111 0 is produced.

1.6 Concept of Models & Coding (cont.)

Input	Transmit	Adaptive compression table		
		Data	*Count*	*Code*
(Initial state)		X_1	0	0
		X_2	0	10
		X_3	0	110
		X_4	0	111
X_2	10	X_2	1	0
		X_1	0	10
		X_3	0	110
		X_4	0	111
X_4	111	X_2	1	0
		X_4	1	10
		X_1	0	110
		X_3	0	111
X_2	0	X_2	2	0
		X_3	1 4	10
		X_1	0	110
		X_3	0	111

Input:

X_2 X_4 X_2

Output (for model):

Fixed - 10 111 10

Adaptive - 10 111 0

1.6 Concept of Models & Coding (cont.)

- Thus, data compression relies on the combination of both, modelling and coding.

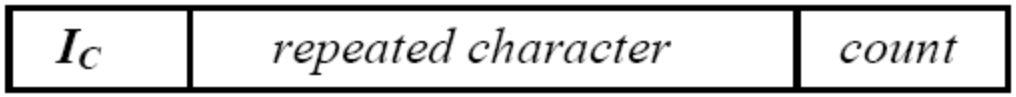
- Generally, adaptive models are capable of better compression performance than fixed models.

2. OVERVIEW OF LOSSLESS TECHNIQUES

- 2.1 Suppression Method
- 2.2 Substitution & Dictionary Methods
- 2.3 Bit Level Method
- 2.4 Relative Encoding
- 2.5 Statistical Method
- 2.6 Predictive Coding
- 2.7 Combination Method
- 2.8 Other Compression Techniques

2.1 Suppression Method - Runlength Coding

- Run-length encoding is a generalised suppression technique that compresses any repeating character sequence (Rubin, 1976).



- Example: the stream $********DaaaaaaaaaXYZ would be encoded as $$I_C$*8D$I_C$a9XYZ$ *where* I_C *is* followed by the repeated character (i.e. 'a') and the count).

- Note: there must be sequences of at least 4 repeated characters for compression.

2.2 Substitution & Dictionary Methods

- The substitution methods (e.g. **Pattern Substitution**) entail the use of different characters to replace the occurrence of a sequence or string of characters.
  - E.g., in a programming language such as BASIC, the keywords (e.g. END, GOTO, IF, PRINT etc.) could be easily compressed (into a form such as **$1**, **$2** etc.)
- Better substitution strategies employ the use of lookup tables or dictionaries (fixed or adaptive).
- This type of technique is one of the most popular for good compression performance.

2.3 Bit-Level Methods - Bit Mapping

- In the example below, the 8 byte input is encoded as a 4 byte code, where ∅ is the special character.

- The *10010001* is the one-byte bit map character and *X, Y* and *Z* are the other characters in the string (i.e. not the chosen character).

- The location of these characters in the original data can determined by 1's in the bit map and the order of occurrence in the compressed form.

2.4 Relative Encoding / Delta Coding

- Relative Encoding (Nelson & Gailly, 1996), also known as Delta Coding, is applicable only to data streams that vary very slightly from each other or run sequences that can be broken into patterns that are relative to each other (e.g. bit patterns of digital fax machines).

- Compression is achieved by transmitting the actual first input symbol, and then only sending subsequent differences

Original data

46	46	46.1	46	46.1	46.2

Relative encoded data

46	0	.1	-.1	.1	.1

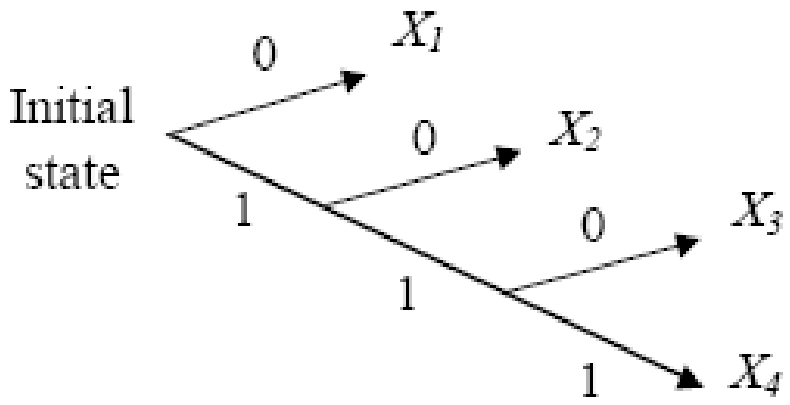
2.5 Statistical Methods - Huffman Coding

- Huffman Coding (Huffman, 1952) is a well known algorithm based on the information theory published by Shannon (Shannon, 1948) which uses the probabilities of occurrence to reduce the average code length used to represent the symbols of an alphabet.

- Compression is achieved by assigning shorter encoded characters to represent the most frequently occurring characters.

2.5 Statistical Methods - Huffman Coding (cont.)

Char	Probability & Code Development	Code
X_1	0.5625 (0) — 1.0	0
X_2	0.1875 (0) — 0.4375 (1) — 1.0	10
X_3	0.1875 (0) — 0.25 (1) — 0.4375 (1) — 1.0	110
X_4	0.0625 (1) — 0.25 (1) — 0.4375 (1) — 1.0	111



2.6 PREDICTIVE CODING

- A powerful technique used in lossy and lossless compression is to estimate the present value of a sample using its past values.
- The lossless implementation requires transmission / storage of the residues (difference) which are generally of much lesser magnitude and size than the original samples.
- To satisfy the lossless criterion, both the encoder and decoder need to simulate an identical prediction process.

2.6 PREDICTIVE CODING (CONT.)

- The order of the predictor (i.e. the number of past values used to predict the present value) determines the number of coefficient parameters as well as the number of initial input values to be transmitted.

- These coefficients may be fixed throughout the prediction process, or may be updated adaptively during the prediction process through the use of algorithms such as the normalised least mean squares (NLMS) and recursive least squares lattice (RLSL) (Haykin, 1991).

2.6 Predictive Coding - FIR Filter

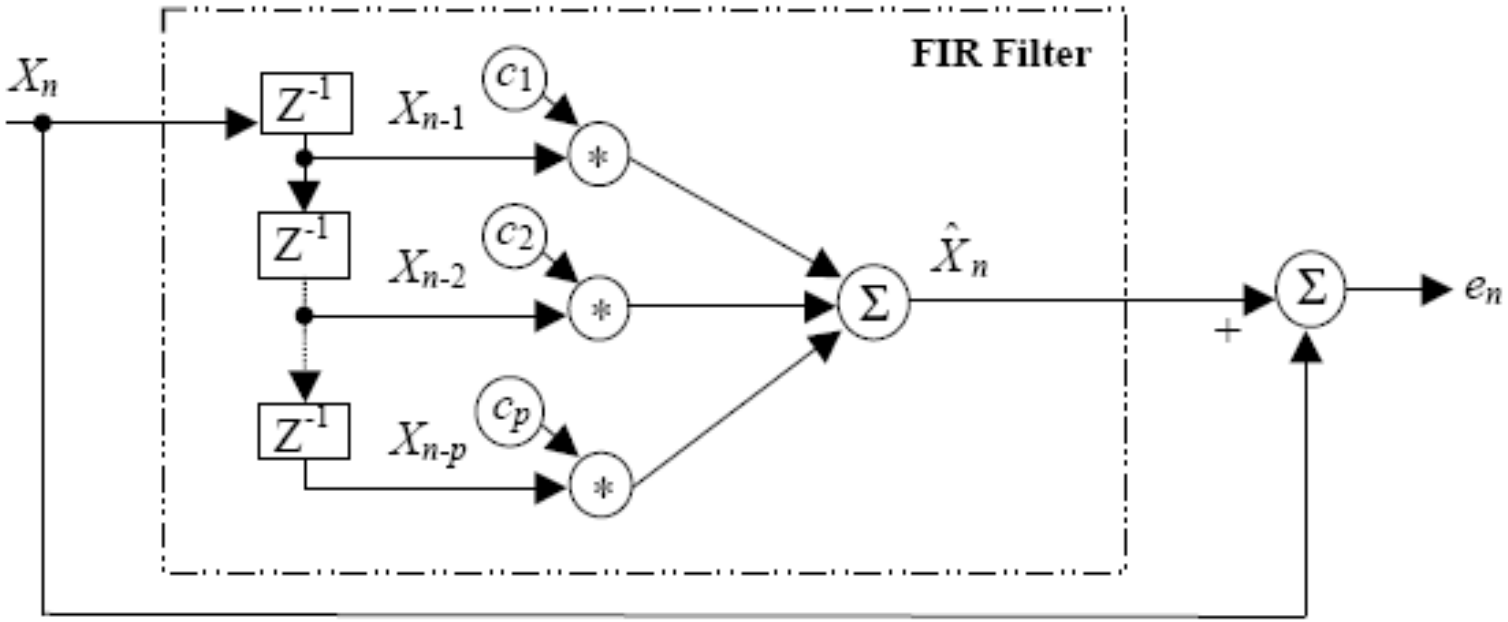
- The FIR filter uses delays on its inputs, such that it is the past values that are used at each iteration.

- The filter can be set up as a predictor by taking advantage of this characteristic.

- The structure of an p^{th} order FIR predictor is shown in the next slide. In the figure:
  - the input n^{th} iteration (X_n), is predicted using the product of the past values obtained through p delays (denoted by the *Z-1* symbols), with the corresponding p coefficients(denoted by c_j).

2.6 Predictive Coding - FIR Filter (cont.)

- the sum of these values produce the predicted value ($X\hat{}_n$).
- the n^{th} residue (e_n) is calculated by taking $X_n - X\hat{}_n$.
- The process is then repeated at each iteration until all the inputs are consumed.



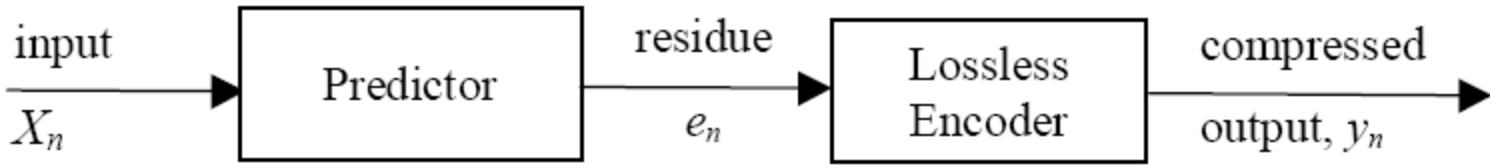
2.7 COMBINATION METHOD

- Multiple strategies can often be combined in order to provide better compression.

- These methods decorrelate the input to produce intermediate output codes which are then further decorrelate to produce the final output codes.

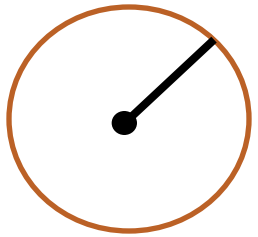
2.7 Combination Methods - Two-Stage Scheme

- This scheme usually uses a predictor-encoder combination
  - The first phase uses lossless prediction to reduce the token size.
  - The encoder in the second phase further reduces the tokens by decorrelating the tokens from the first phase.



2.8 Other Compression Techniques - Vector Graphics

- An alternative method of compression, especially for graphics is through the use of vector graphics.
- The basic parameters and formulae can be stored, instead of every bit of the image.
- For example, a circle can reconstructed from a formula, given the location of the centre point and the radius. Storing these parameters would suffice, instead of storing the location of each pixel that makes up the circle.



- The same method can be used for most of the common shapes. This method of compression is usually done by the graphics applications, rather than upon transmission.

3. OVERVIEW OF LOSSY TECHNIQUES

- 3.1 Lossy Compression
- 3.2 Examples of Lossy Compression Algorithms
- 3.3 JPEG
- 3.4 JPEG 2000
- 3.5 MPEG
- 3.6 MP3

3.1 LOSSY COMPRESSION

- Lossy compression is most commonly used to compress multimedia data (audio, video, still images), especially in applications such as streaming media and internet telephony.

- Slight loss of data and accuracy in multimedia data usually automatically compensated by the human mind.

- Audio can often be compressed at **10:1** with imperceptible loss of quality, and video can be compressed immensely (e.g. **300:1**) with little visible quality loss. Lossily compressed still images are often compressed to 1/10th their original size, as with audio, but the quality loss is more noticeable, especially on closer inspection.

Reference: http://en.wikipedia.org/wiki/Lossy_compression

3.1 LOSSY COMPRESSION (CONT.)

<a

b>

a) Original Image (lossless PNG, **60.1 kB** size) — raw image is **108.5 kB**

b) Low compression (**84% less** information than uncompressed PNG, **9.37 kB**)

<c

d>

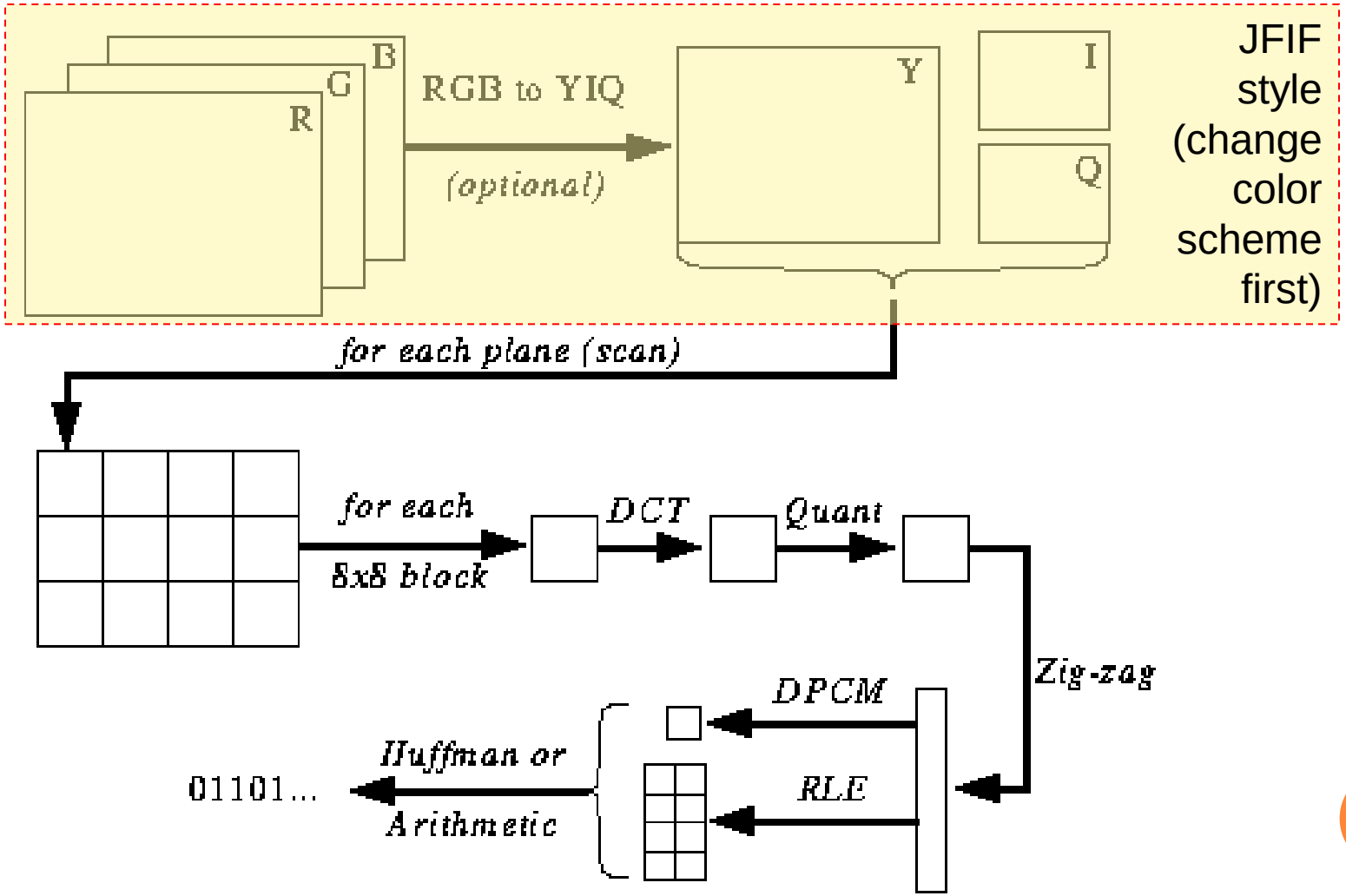
c) Medium compression (**92% less** information than uncompressed PNG, **4.82 kB**)

d) High compression (**98% less** information than uncompressed PNG, **1.14 kB**)

3.3 Joint Photographic Experts Group (JPEG)

- Standard approved in 1994
- Usage (Good performance)
  - best on photographs and paintings of realistic scenes with smooth variations of tone and color.
  - web usage in particular, where the bandwidth used by an image is important
- Not recommended for
  - files that will undergo multiple edits, as some image quality will usually be lost each time the image is decompressed and recompressed

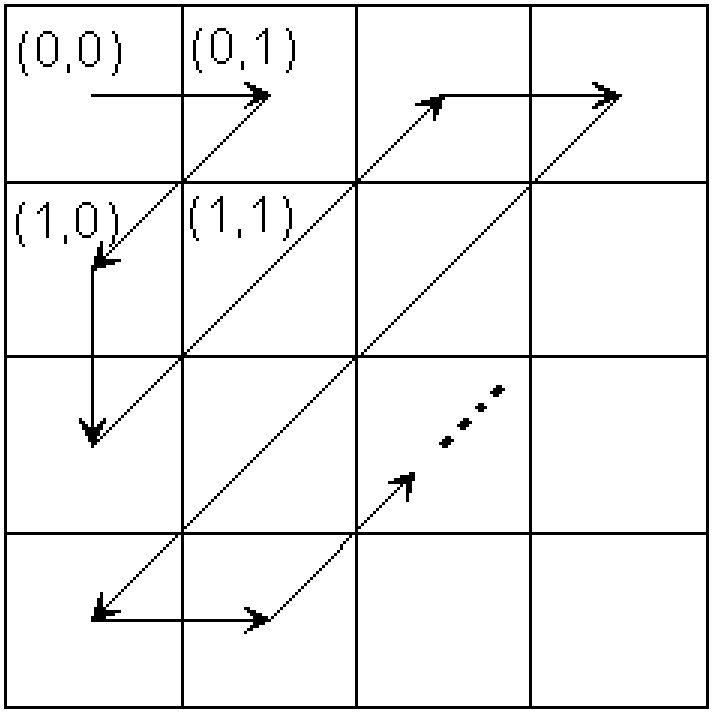
3.3 Joint Photographic Experts Group (JPEG) (cont.)



Reference: http://www.cs.cf.ac.uk/Dave/Multimedia/node234.html

3.3 Joint Photographic Experts Group (JPEG) (cont.)

DCT MATRIX:



The sequence continues for the entire 8 by 8 block.

Reference: http://cobweb.ecn.purdue.edu/~ace/jpeg-tut/jpegtut1.html

3.4 JPEG 2000

- Developed by the Joint Photographic Experts Group in 2000
- Filename extensions - .jp2, .jpx
- Advantages (compared to JPEG)
  - Allows more sophisticated progressive downloads with similar compression rates
  - JPEG 2000 becomes increasingly blurred with higher compression ratios rather than generating JPEG's "blocking and ringing" artifacts
  - Additional meta-data, e.g. lighting and exposure conditions, is encoded in XML. JPEG keep it in an application marker in the Exif format

Reference: http://www.cs.cf.ac.uk/Dave/Multimedia/node234.html

3.4 JPEG 2000 (CONT.)

- Disadvantages (compared to JPEG)
  - JPEG 2000 requires far greater decompression time
- Features
  - Superior compression performance
  - Multiple resolution representation
  - Progressive transmission
  - Lossless and lossy compression
  - Random code-stream access and processing / ROI
  - Error resilience / robust to noise
  - Flexible file format – colour space, metadata, interactivity
  - Side channel spatial information – supports transparency / alpha planes

3.5 Moving Pictures Expert Group (MPEG)

- MPEG Standards
  - **MPEG-1 :** Audio and video compression (1993). Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s (medium-bandwidth)
  - **MPEG-2 :** A video standard. Addition to MPEG-1. for high-bandwidth/broadband usage. Used in digital TVs, DVD-Videos and in SVCDs. Supports up to 5 audio channels.
  - **MPEG-4 :** designed specially for low-bandwidth (less than 1.5MBit/sec bitrate) video/audio encoding purposes. Best-known MPEG-4 video encoders are called DivX and XviD

Reference: http://www.afterdawn.com/glossary/terms/mpeg.cfm

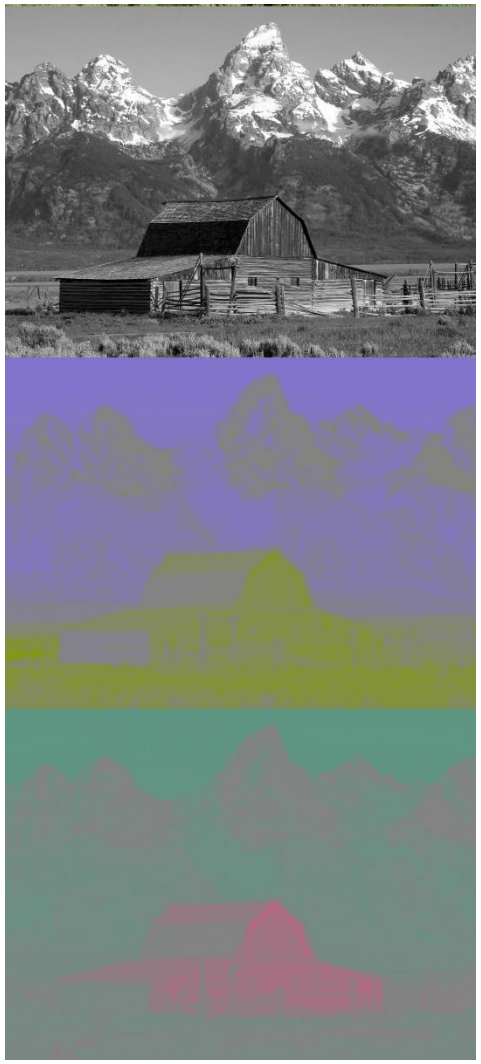
3.5 Moving Pictures Expert Group (MPEG) (cont.)

- **Algorithm:**
  1. **Color Space** of the original video is changed to YUV (Luminance and 2xChrominance) which is MPEG-1's native color space.
  2. Chrominance information is reduced from details for every pixel to details for groups of four pixels (**Color Space Sub-Sampling**).
  3. Full luminance and reduced chrominance information is re-stated as frequencies which are easier to compress (**Discrete Cosine Transform**).

Reference: http://www.bulents-byteme.com/digimed/MPEG_video_encoding.asp

3.5 Moving Pictures Expert Group (MPEG) (cont.)

Original RGB image



Y'
(Luma*)

Cb
(Chrominance blue)

Cr
(Chrominance red)

** Luma (Y') is the video-encoded luminance, a gamma compressed 'skewed' luminance used in displays*

Reference: http://wolfcrow.com/blog/understanding-luminance-and-chrominance/

3.5 Moving Pictures Expert Group (MPEG) (cont.)

4. Small frequency values for high frequencies that human eye is less sensitive to are set to zero for better compression (**Quantization**).

5. The reduced frequency information is scanned and only non-zero values are encoded (**Run-Length Encoding**).

6. The frame is encoded either as an independent Intra frame (**I-Frame encoding**) or a frame relative to other frames, i.e. Predicted and Bi-directional (**P- and B-Frame encoding**).

Reference: http://www.bulents-byteme.com/digimed/MPEG_video_encoding.asp

3.5 Moving Pictures Expert Group (MPEG) (cont.)

7. For relative frames, the current frame is analyzed by dividing its image into 16x16 pixel macro blocks to search them in past /future frames but in diff. places.

8. If a macro block is found in a future or past frame, only its new location instead of the full image data is encoded (**Motion Prediction and Compensation**).

9. If the light and/or color is slightly different for the same block in the current frame then these differences are also encoded (**Error Prediction**).

Reference: http://www.bulents-byteme.com/digimed/MPEG_video_encoding.asp

3.6 MP3

- MPEG-1 Audio Layer III (MP3)
- Common audio format for consumer audio storage
- De facto standard encoding for the transfer and playback of music on digital audio players
- MP3 file created using the mid-range bit rate setting of 128 kbit/s will result in a file that is typically about 1/10th the size of the CD file created from the original audio source.

3.6 MP3 (CONT.)

- CD-quality
  - Music is sampled 44,100 times per second. The samples are 2 bytes (16 bits) long, 689 kbit/s.
  - Separate samples are taken for the left and right speakers in a stereo system, so **1,378 kbit/s**.

- MP3
  - Goal to compress a **CD-quality** song by a factor of 10 to 14 without noticeably affecting the sound.
  - With MP3, a 32 MB song on a CD compresses down to about 3 MB.
  - Bitrate 8-320 kbit/s (generally **128 kbit/s**)

Reference: http://www.afterdawn.com/glossary/terms/mp3.cfm

3.6 MP3 (CONT.)

- Lossy compression - reduce accuracy of parts of sound deemed beyond the auditory resolution ability of most people (perceptual coding / psychoacoustic algorithms
  - **Leave 1 kHz to 4 kHz** audio frequencies untouched, (range the average human ear hears best)
  - **remove quieter** audio sounds playing at the same time as louder ones (not likely to be noticed)
  - converting stereo signals into **mono signals** (e.g. deep bass directionality not obvious)
  - file size is further reduced by optimizing the data for the most frequently occurring signals (**Huffman Coding**)

Reference: http://absoluteinsight.net/1678

4. MEASURING COMPRESSION

- 4.1 Introduction
- 4.2 Compression ratio (*CR)*
- 4.3 Mean Squared Error (MSE)
- 4.4 Signal to Noise Ratio (SNR)

4.1 INTRODUCTION

- In order to compare the performance of compression techniques, certain measurement parameters are used.
- For lossy compression, comparison can be done by calculating the Mean Squared Error (MSE) and Weighted MSE, Mean Opinion Score (for image compression) and other end-use measurements.
- For lossless compression, generally comparisons of physical size is used.

4.2 COMPRESSION RATIO

- This is one of the most popular measure of compression performance.

- The value of C_R , refers to the size ratio *original :compressed* (e.g. CR = 2:1 refers to data compressed to half of its original size).

$$C_R = \frac{\text{length of original data}}{\text{length of compressed data}}$$

4.3 Mean Squared Error (MSE)

- One of the most popular measurements for distortion, MSE is calculated as

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2$$

where

x_n = input data sequence,

y_n = reconstructed data sequence,

N = length of the data sequence.

4.4 Signal to Noise Ratio (SNR)

- Signal strength is measured in decibels (dB) using

$$SNR = 10 \log_{10} \frac{\sigma_x^2}{\sigma_d^2}$$

where

σ_x^2 = average square value of the original data sequence

σ_d^2 = MSE

4.5 Signal to Noise Ratio (SNR) (cont.)

- The Peak SNR (PSNR) is also a common measure, given as

$$PSNR = 10 \log_{10} \frac{x^2_{peak}}{\sigma_d^2}$$

where

x^2_{peak} = square value of the highest value of the original data sequence

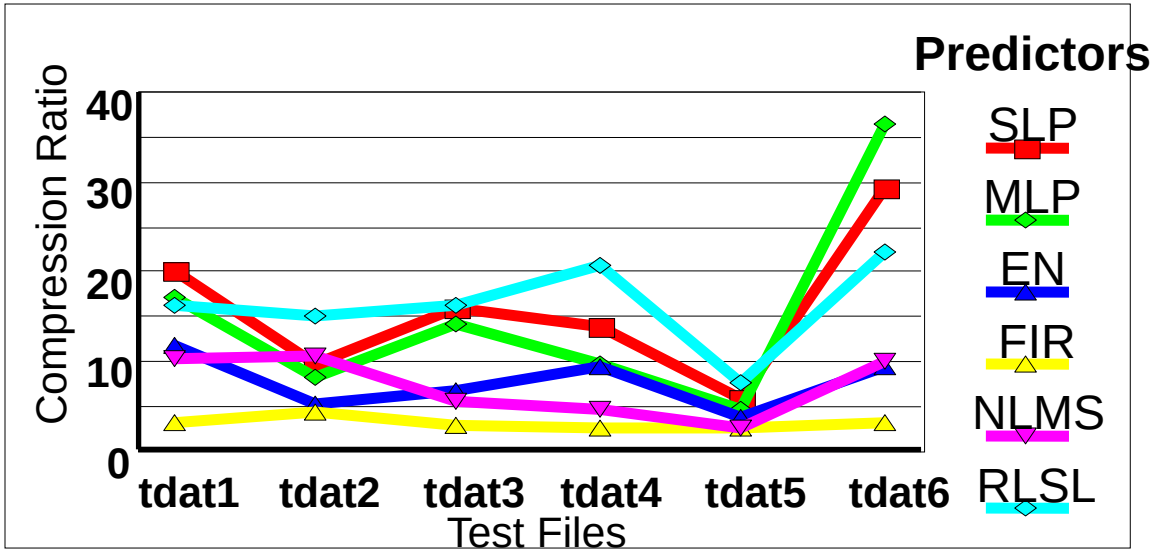
σ_d^2 = MSE

5. CONCLUSION

- By now, you should have :
  - gained a sound understanding of the basic principles of data compression,
  - the insight on understanding and enhancing compression methodologies to develop your own algorithms for a variety of data.

5. CONCLUSION (CONT.)

- Choice of compression algorithm must be suitable to the pattern of the data.
  - Different algorithms, even based on similar concepts, may differ in performance.

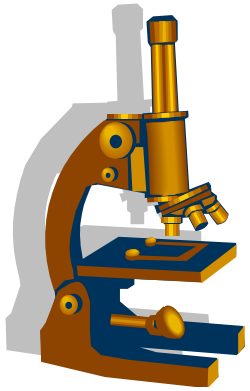


5. CONCLUSION (CONT.)

- Remember:
  - Innovate techniques at various parts of the compression chain in order to improve compression. Combination techniques often produce good compression results.
  - When dealing with complex data, attempt to identify the individual components/patterns and compress them separately.

Q&A

- HAPPY MAKING THINGS *smaller* !!!



Thank you . . .