

Introduction

Data Compression

DATA COMPRESSION

- Data Compression is a reduction in the number of bits needed to represent data. Compressing data can save storage capacity, speed file transfer, and decrease costs for storage hardware and network bandwidth.

DATA COMPRESSION: REMOVE REDUNDANCY

DATA COMPRESSION

- Data Compression refers to the reducing the number of bits that need to be transmitted over communication channel.
- Data Compression reduces the number of bits sent

DATA COMPRESSION: REMOVE REDUNDANCY

DATA COMPRESSION

- Data Compression becomes particularly important when we send data with high size such as audio & video
- Even with very fast transmission speed of data we need to send data in short time. We need to Compress data for this purpose.

STORAGE ----- FAST TRANSMISSION

DATA COMPRESSION

- Virtually all form of data contain **redundancy**
i.e. it is the amount of wasted "space" used to transmit certain data.
- By making use of more efficient data representation methods, redundancy can be reduced.
- *The goal of data compression is to represent an information source (e.g. a data file, a speech signal, an image, or a video signal) as accurately as possible using the fewest number of bits.*

DATA COMPRESSION

- Data compression ratio is defined as the ratio between the uncompressed size and compressed size.

$$\text{Compression Ratio} = \frac{\text{Uncompressed Size}}{\text{Compressed Size}}$$

- Thus a representation that compresses a 10 MB file to 2MB has a compression ratio of $10/2 = 5$, often notated as an exploit ratio , 5:1 (read “five” to “one”) or as an implicit ratio 5/1

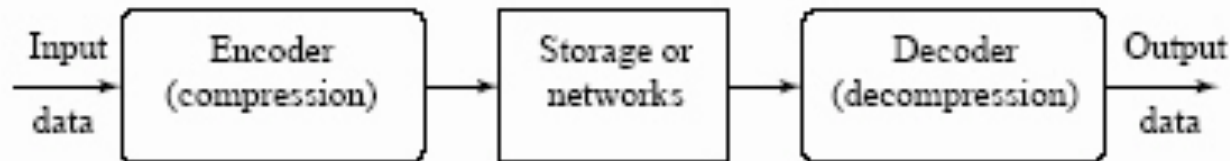
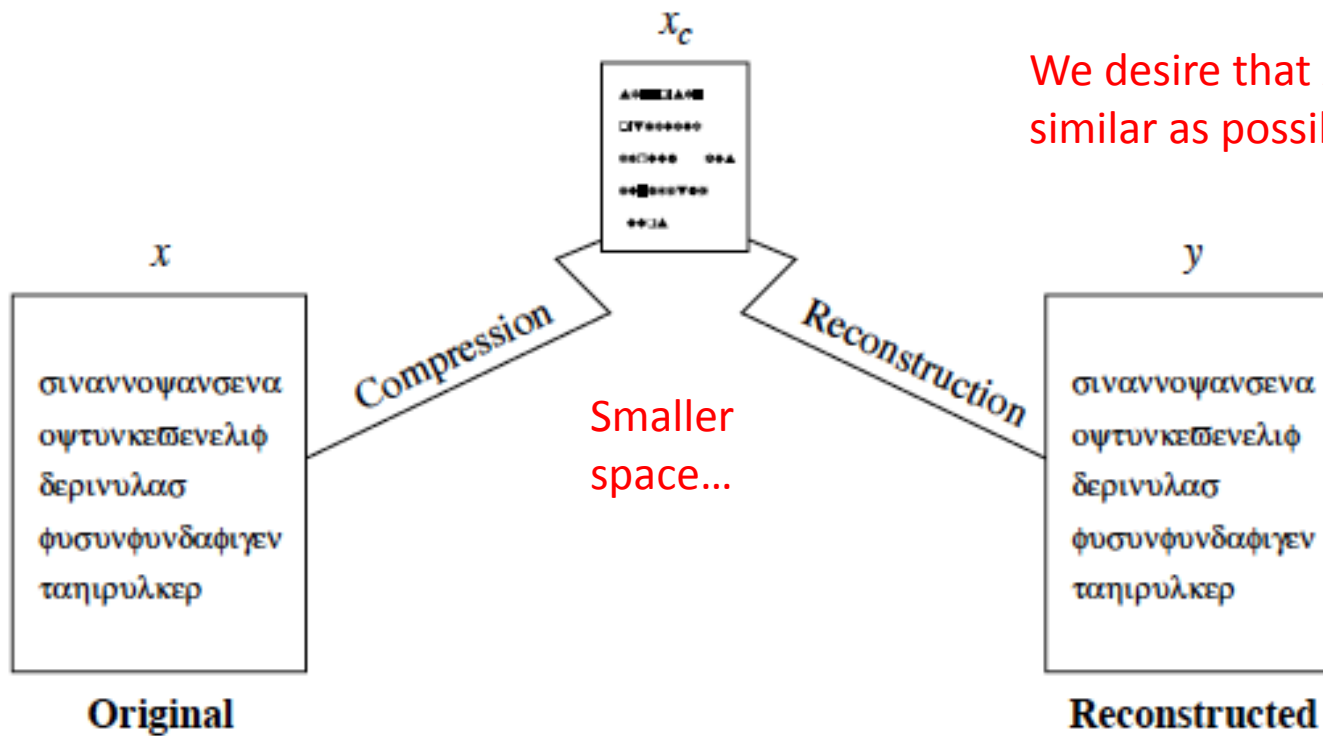
$$\text{compression ratio} = B_0/B_1$$

B_0 = number of bits before compression

B_1 = number of bits after compression

DATA COMPRESSION

- $$\text{Space Savings} = 1 - \frac{\text{Compressed Size}}{\text{Uncompressed Size}}$$
- Thus a representation that compresses a 10 MB file to 2 MB would yield a space saving of $1 - 2/10 = 0.8$ often notated as a percentage, 80 %

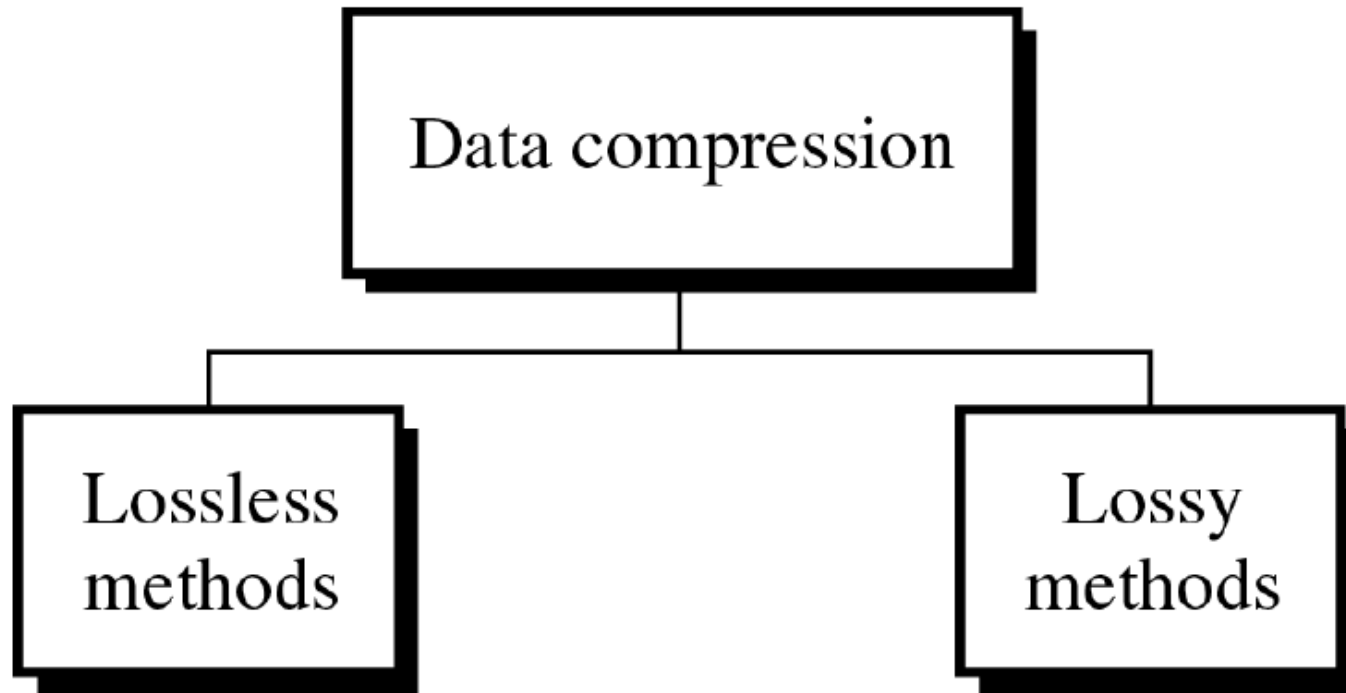


compression ratio = B_0/B_1

B_0 = number of bits before compression

B_1 = number of bits after compression

Data Compression Methods



Just to understand (some examples that you already know):

- **quantization:** lossy compression
- **Fourier sub-sampling satisfying Nyquist:** lossless compression
- **Fourier sub-sampling which does not satisfy Nyquist:** lossy compression



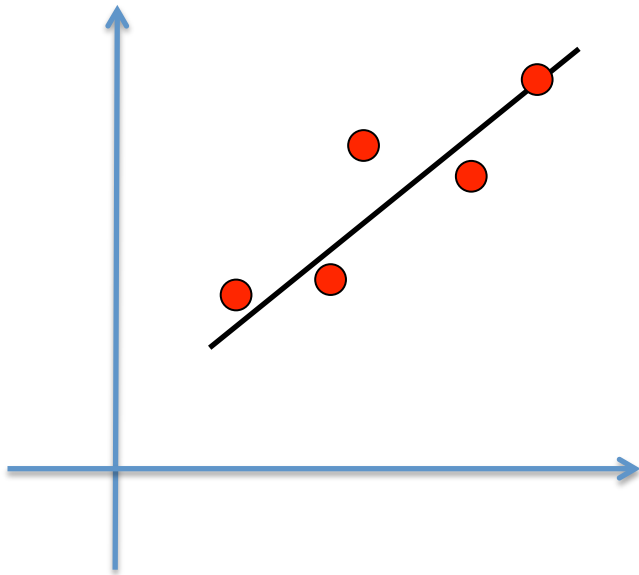
these are also considered compression techniques, and often are used jointly with other compression methods.

Just to understand (some examples that you already know):

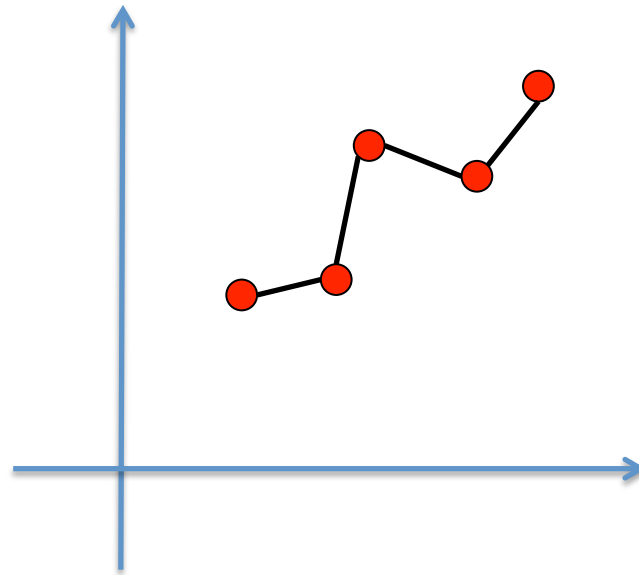
- **modeling:** lossy compression (regression), lossless compression (interpolation)



these are also considered compression techniques, and often are used jointly with other compression methods.



regression



interpolation

Modeling and Coding

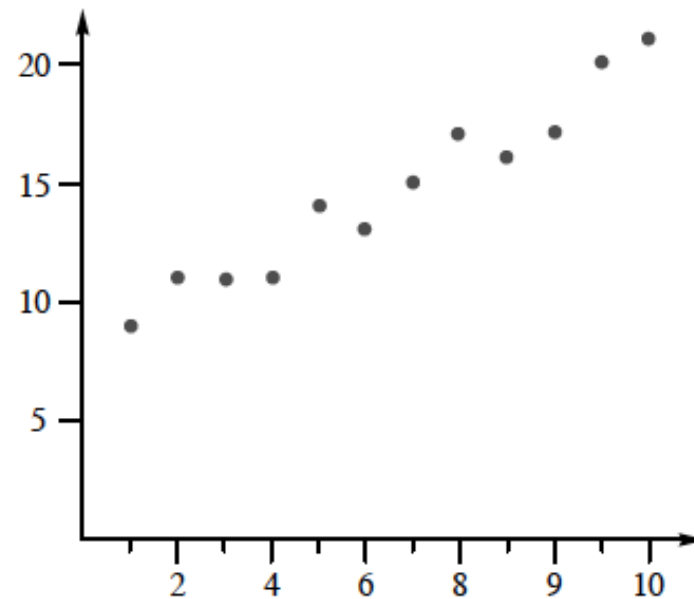
Example 1.2.1:

Consider the following sequence of numbers $\{x_1, x_2, x_3, \dots\}$:

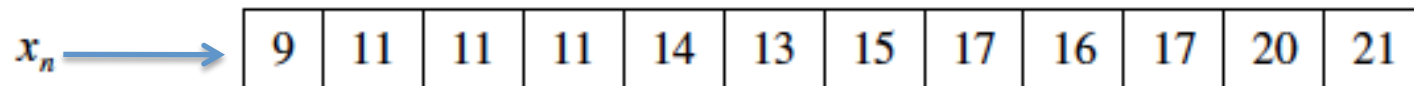
9	11	11	11	14	13	15	17	16	17	20	21
---	----	----	----	----	----	----	----	----	----	----	----

If we were to transmit or store the binary representations of these numbers, we would need to use 5 bits per sample. However, by exploiting the structure in the data, we can represent the sequence using fewer bits. If we plot these data as shown in Figure 1.2, we see that the data seem to fall on a straight line. A model for the data could therefore be a straight line given by the equation

$$\hat{x}_n = n + 8 \quad n = 1, 2, \dots$$



Modeling and Coding



$$\hat{x}_n = n + 8 \quad n = 1, 2, \dots$$

Thus, the structure in the data can be characterized by an equation. To make use of this structure, let's examine the difference between the data and the model. The difference (or residual) is given by the sequence

$$e_n = x_n - \hat{x}_n : 0 \ 1 \ 0 \ -1 \ 1 \ -1 \ 0 \ 1 \ -1 \ -1 \ 1 \ 1$$

The residual sequence consists of only three numbers $\{-1, 0, 1\}$. If we assign a code of 00 to -1 , a code of 01 to 0, and a code of 10 to 1, we need to use 2 bits to represent each element of the residual sequence. Therefore, we can obtain compression by transmitting or storing the parameters of the model and the residual sequence. The encoding can be exact if the required compression is to be lossless, or approximate if the compression can be lossy. ◆

Just to understand (some examples that you already know):

➤ **prediction: (related to modeling)**



these are also considered compression techniques, and often are used jointly with other compression methods.

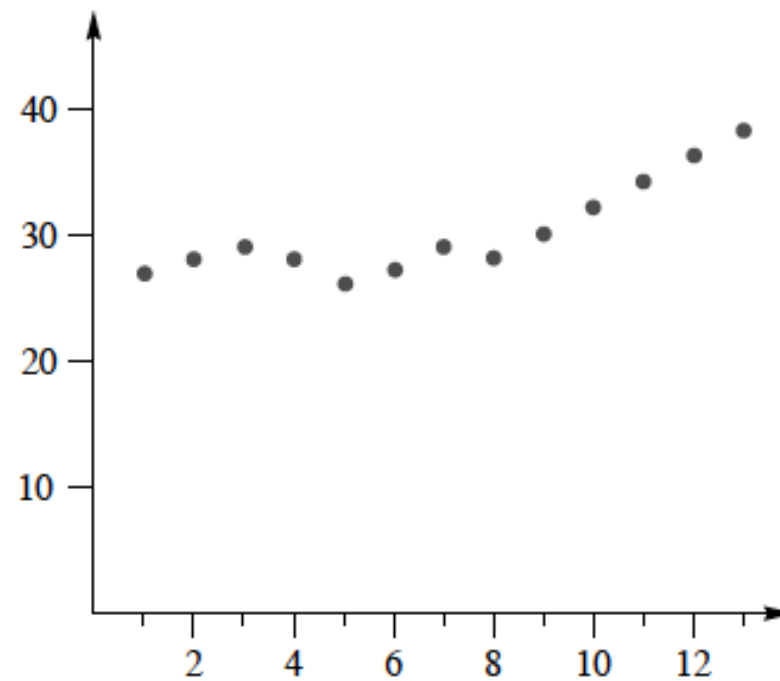
Modeling and Coding

Example 1.2.2:

Consider the following sequence of numbers:

27	28	29	28	26	27	29	28	30	32	34	36	38
----	----	----	----	----	----	----	----	----	----	----	----	----

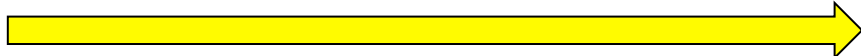
The sequence is plotted in Figure 1.3.



The sequence does not seem to follow a simple law as in the previous case. However, each value is close to the previous value. Suppose we send the first value, then in place of subsequent values we send the difference between it and the previous value. The sequence of transmitted values would be

27	1	1	-1	-2	1	2	-1	2	2	2	2	2
----	---	---	----	----	---	---	----	---	---	---	---	---

Like the previous example, the number of distinct values has been reduced. Fewer bits are required to represent each number and compression is achieved.

 The decoder adds each received value to the previous decoded value to obtain the reconstruction corresponding to the received value. Techniques that use the past values of a sequence to *predict* the current value and then encode the error in prediction, or residual, are called *predictive coding* schemes.

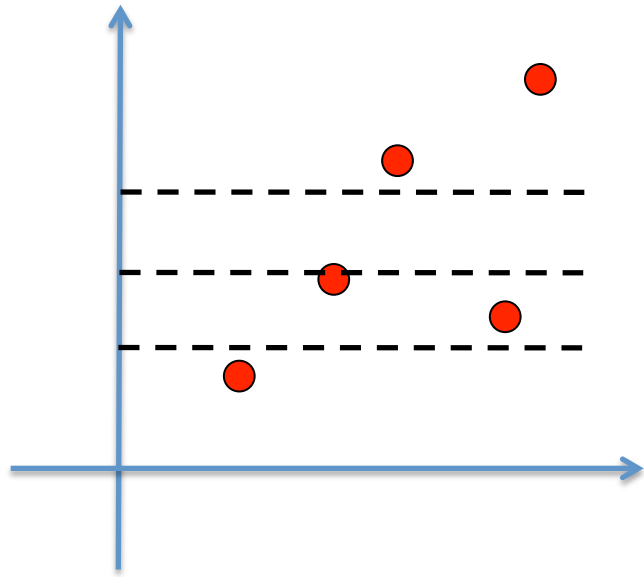
Just to understand (some examples that you already know):

- **quantization:** lossy compression (already mentioned in other slide)
- **clustering or smoothing:** lossy compression

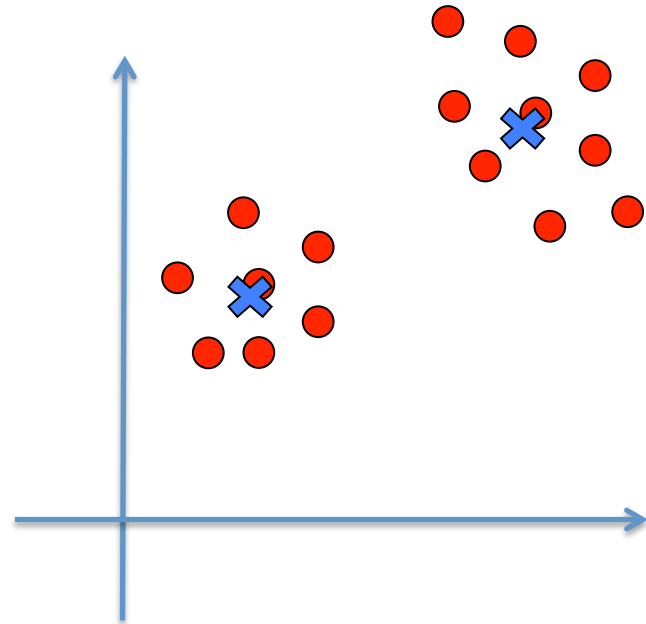


these are also considered compression techniques, and often are used jointly with other compression methods.

Especially for clustering, please think in removing redundancy



quantization



clustering

Statistical redundancy

Example

Suppose we have the following sequence:

abarayaranbarraybranbfarbfaarbfaaarbaway

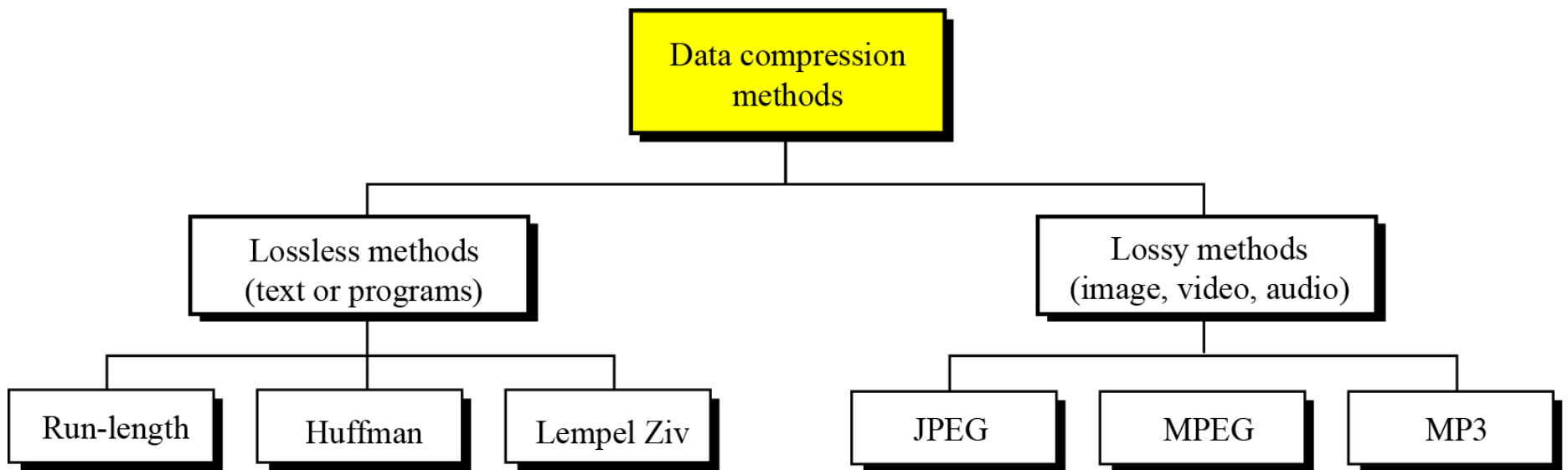
which is typical of all sequences generated by a source. Notice that the sequence is made up of eight different symbols. In order to represent eight symbols, we need to use 3 bits per symbol. Suppose instead we used the code shown in Table 1.1. Notice that we have assigned a codeword with only a single bit to the symbol that occurs most often, and correspondingly longer codewords to symbols that occur less often. If we substitute the codes for each symbol, we will use 106 bits to encode the entire sequence. As there are 41 symbols in the sequence, this works out to approximately 2.58 bits per symbol. This means we have obtained a compression ratio of 1.16:1. We will study how to use statistical redundancy of this sort in Chapters 3 and 4.

TABLE 1.1 **A code with codewords of varying length.**

<i>a</i>	1
<i>n</i>	001
<i>b</i>	01100
<i>f</i>	0100
<i>r</i>	0111
<i>w</i>	000
<i>y</i>	01101
<i>y</i>	0101



When dealing with text, along with statistical redundancy, we also see redundancy in the form of words that repeat often. We can take advantage of this form of redundancy by constructing a list of these words and then represent them by their position in the list. This type of compression scheme is called a *dictionary* compression scheme.



Lossless Compression

- The lossless compression refers to data compression techniques in which no data is lost.
- The PKZIP compression technology is an example of lossless compression.
- For most types of data, lossless compression techniques can reduce the space needed by only about 50%.
- For greater compression, one must use a *lossy compression*

Lossy Compression

- Lossy Compression refers to data compression techniques in which some amount of data is lost. Lossy compression technologies attempt to eliminate redundant or unnecessary information.
- lossy compression reduces a file by permanently eliminating certain information, especially redundant information.

*Several methods have been developed using lossy compression techniques. **Joint photographic experts group (JPEG)** is used to compress pictures and graphics. **Motion picture experts group (MPEG)** is used to compress video.*

Lossless and Lossy Compression Techniques

- Data compression techniques are broadly classified into lossless and lossy.
- Lossless techniques enable exact reconstruction of the original document from the compressed information.
 - Exploit redundancy in data
 - Applied to general data
 - Examples: Run-length, Huffman, LZ77, LZ78, and LZW
- Lossy compression - reduces a file by permanently eliminating certain redundant information
 - Exploit redundancy and human perception
 - Applied to audio, image, and video
 - Examples: JPEG and MPEG
- Lossy techniques usually achieve higher compression rates than lossless ones but the latter are more accurate.

Compression Utilities and Formats

- Compression tool examples:

- winzip, pkzip, compress, gzip

- General compression formats:

- .zip, .gz

- Common image compression formats:

JPEG, JPEG 2000, BMP, GIF, PCX, PNG, TGA, TIFF, WMP

- Common audio (sound) compression formats:

MPEG-1 Layer III (known as MP3), RealAudio (RA, RAM, RP), AU, Vorbis, WMA, AIFF, WAVE, G.729a

- Common video (sound and image) compression formats:

MPEG-1, MPEG-2, MPEG-4, DivX, Quicktime (MOV), RealVideo (RM), Windows Media Video (WMV), Video for Windows (AVI), Flash video (FLV)

Factors	Data compression	
	LOSSLESS COMPRESSION	LOSSY COMPRESSION
Definition	Lossless compression is a class of data compression algorithms that allow the original data to be perfectly reconstructed from the compressed data.	Lossy compression is the class of data encoding methods that uses inexact approximations to represent the content. These techniques are used to reduce the data size for storage, handling, and transmitting content[8]
Algorithm	RLW, LZW, Arithmetic encoding, Huffman coding, Shannon Fano coding	Transform coding, DCT, DWT, Fractal compression, RSSMS
USES	Text or programs, images and sound	Images, audio and video
IMAGES	RAW, BMP and PNG	JPEG and GUI are lossy image
Audio	WAV, FLAC AND ALAC	MP3, MP4 and OGG
Video	Few lossless video formats are in common consumer use, they would result in video files taking up a huge amount of space	Common Formats like H-264, MKV and WMV. H-264 can provides smaller files with higher qualities than previous generation of video codec because it has a “smaller” algorithm that’s better at choosing the data to throw out.
Advantages	It maintains quality. Conversion in any other format possible without loss of audio information.	It can make a multimedia file much smaller than its original size. It can reduce file sizes much more than lossless compression.
Dis Advantages	It does not reduce the file size as much as lossy compression. Lossless encoding technique cannot achieve high levels of compression.	Conversion to another format only with loss of audio information. It cannot be used in all types of files because it works by removing data. Text and data cannot be compressed because they do not have redundant information.