- Flash (SWF) (H.W)
- Video

If an animation is designed to be played back on a TV, either from broadcast signal or from DVD, it must be saved as video. The codec, frame rate , frame size and so on all be determined by the standards of the medium.

If animation is to be delivered over the internet or in some medium intended to be played back on a computer, any of the three main format could be used, with no constraints on their parameters.

Data Compression

Data compression is the process of converting an input data stream (the source stream or the original raw data) into another data stream (the output, or the compressed, stream) that has a smaller size.

Data compression has come of age in the last 20 years; it is popular for two reasons:

1. People like to accumulate data and hate to throw anything away. No matter how big a storage device one has, sooner or later it is going to overflow.
2. People hate to wait a long time for data transfers. When sitting at the computer, waiting for a Web page to come in or for a file to download, we naturally feel that anything longer than a few seconds is a long time to wait.

Data Compression: It is a key solution for **space / time** data transmission

There are many known methods for data compression. They are based on different ***ideas***, are suitable for different types of data, and produce different results, but they are all based on the same principle, namely they compress data by ***removing redundancy*** from the original data in the source file.

In typical English text, for example, the letter **E** appears very often, while **Z** is rare. This is called ***alphabetic redundancy*** and suggests assigning variable size codes to the letters, with E getting the shortest code and Z getting the longest one .

The principle of compressing by removing redundancy also answers the following question: "**Why is it that an already compressed file cannot be compressed further?**"

- The answer, of course, is that such a file has little or ***no redundancy***, so there is nothing to remove. An example of such a

file is random text. In such text, each letter occurs with ***equal probability***, so assigning them fixed-size codes does not add any redundancy. When such a file is compressed, there is no redundancy to remove.

- Another answer is that if it were possible to compress an already compressed file, then successive compressions would reduce the size of the file until it becomes ***a single byte, or even a single bit***. This, of course, is ridiculous since a single byte cannot contain the information present in an arbitrarily large file.

In order to compress a data file, the compression algorithm has to ***examine the data***, ***find redundancies in it***, and ***try to remove them***. Since the redundancies in data depend on the type of data (text, images, sound, etc.), any given compression method has to be developed for a specific type of data and performs best on this type. **There is no such thing as a universal, efficient data compression algorithm.**

There are many compression methods, some suitable for text and others for graphical data (still images or movies).

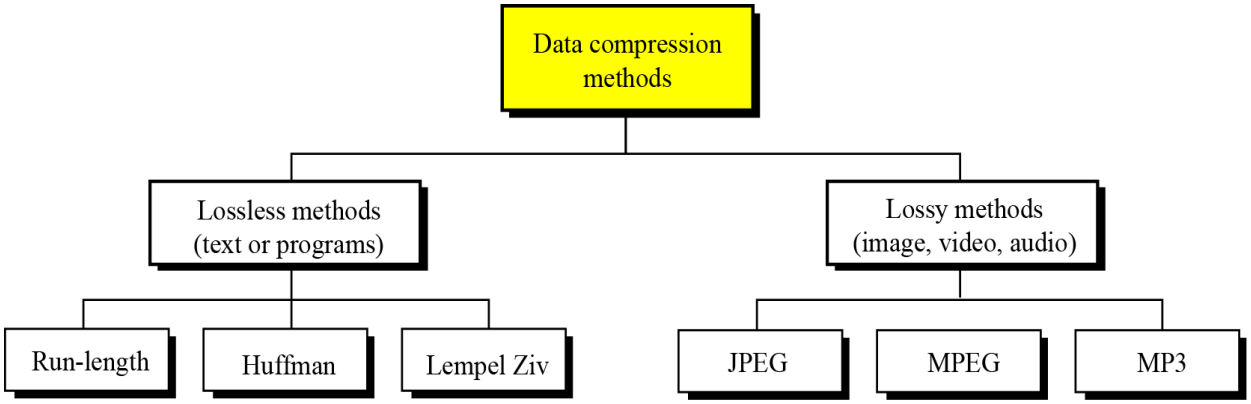
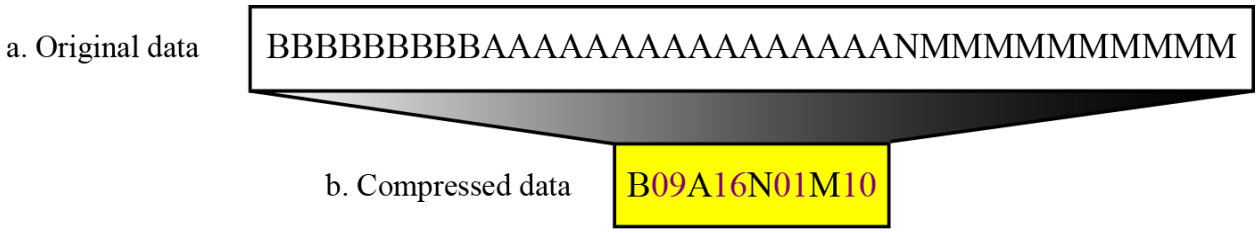


Figure_1: Data compression methods



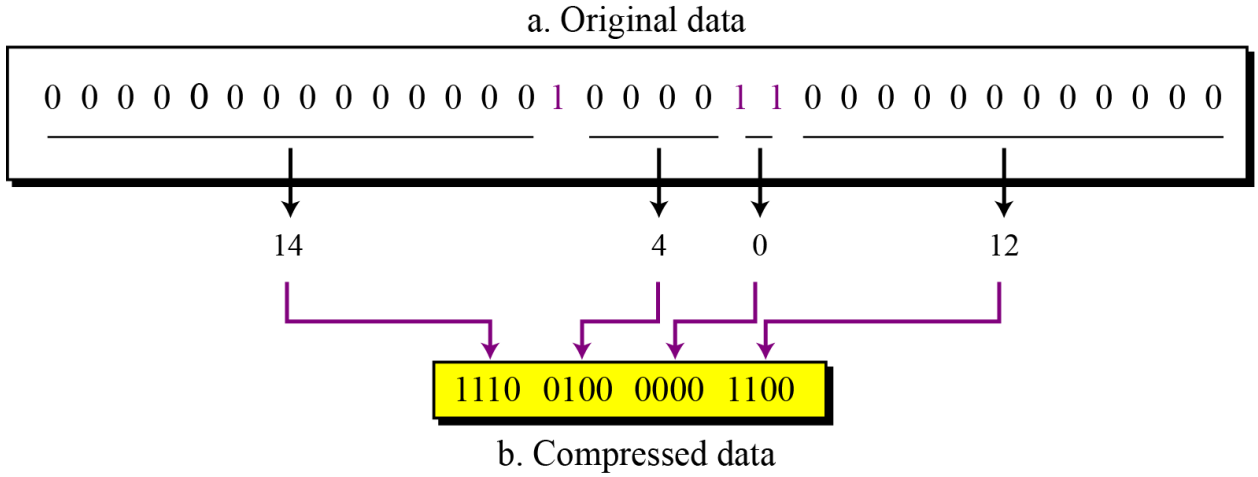


Figure _2 : Run Length Encoding

The Run-length Encoding

This encoding method is frequently applied to images (or pixels in a scan line). It is a small compression component used in JPEG compression.

In this instance, sequences of image elements $X_1, X_2, \ldots, X_n$ are mapped to pairs $(c_1, l_1), (c_2, l_2), \ldots, (c_n, l_n)$ where c_i represent image intensity or colour and l_i the length of the *i*th run of pixels (Not dissimilar to zero length compression above).

For example:

Original Sequence:

11112223333311112222

can be encoded as:

(1,4),(2,3),(3,6),(1,4),(2,4)

The savings are dependent on the data. In the worst case (Random Noise) encoding is more heavy than original file: 2*integer rather 1* integer if data is represented as integers.

Before delving into the details, we discuss important ***data compression terms***:-

1. The ***compressor or encoder*** is the program that compresses the raw data in the input stream and creates an output stream with compressed (low-redundancy) data. The ***decompressor or decoder*** converts in the opposite direction. Note that the term encoding is very general and has wide meaning, but since we discuss only data compression, we use the name encoder to mean data compressor. The term codec is sometimes used to describe both the encoder and decoder. Similarly, the term ***companding*** is short for ***"compressing/expanding."***

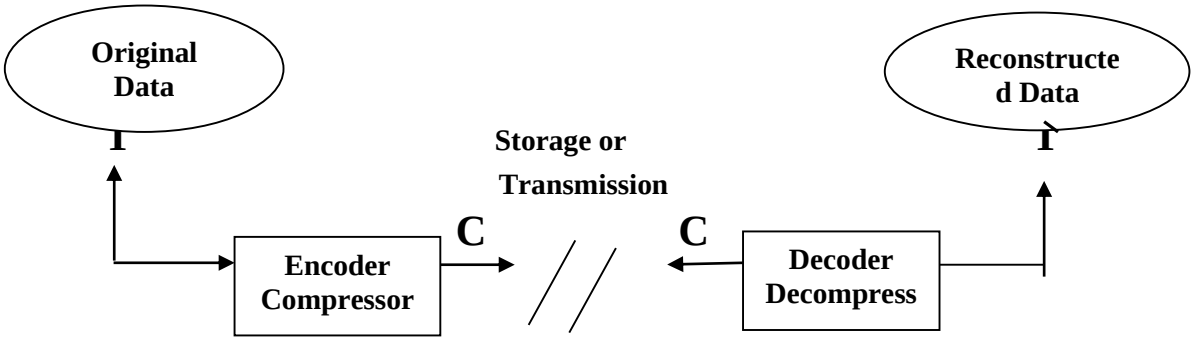


Figure _3 : Encoder /Decoder System (CODEC)

2. The term ***"stream"*** is used instead of ***"file". "Stream" is a more general term*** because the compressed data may be transmitted directly to the decoder, instead of being written to a file and saved.

Also, the data to be compressed may be downloaded from a network instead of being input from a file.

3. For the original input stream, we use the terms ***unencoded , raw , or original data***. The contents of the final, compressed, stream is considered the encoded or compressed data. The term ***bitstream*** is also used in the literature to indicate the compressed stream.
4. ***Compression performance*****:** Several measures are commonly used to express the
5. performance of a compression method.

1. The *compression ratio* is defined as

$$\text{Compression ratio} = \frac{\text{size of the output stream}}{\text{size of the input stream}}.$$

A value of 0.6 means that the data occupies 60% of its original size after compression. Values greater than 1 imply an output stream bigger than the input stream **(negative compression).** The compression ratio can also be called bpb (bit per bit), since it equals the number of bits in the compressed stream needed, on average, to compress one bit in the input stream.

Two more terms should be mentioned in connection with the compression ratio. The term bitrate (or "bit rate") is a general term for bpb and bpc. Thus, the main goal of data compression is to represent any given data at low bit rates. The term bit budget refers to the functions of the individual bits in the compressed stream. Imagine a compressed stream where 90% of the bits are variable-size codes of certain symbols, and the remaining 10% are used to encode certain tables. The bit budget for the tables is 10%.

2. The inverse of the compression ratio is called the compression factor :