Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 10, Number 6 (2014), pp. 917-928

http://www.ripublication.com

A Qualitative Interpretation Of The Sampling Theorem In The Time Domain

Luca Grilli[1] **and Angelo Sfrecola**[2]

Dipartimento di Economia
Università degli Studi di Foggia

Abstract

The signals coming from the real world around us are analog and, in order to treat these quantities with a computer, it is necessary to perform a conversion from analog to digital signals.
In this paper we show how the analog signals can be processed digitally. In order to achieve such result we introduce the main definitions necessary for a more comfortable reading of the paper and then we present the process of signal conversion from analog to digital. The conversion of signals from analog to digital is proposed giving a qualitative interpretation of the sampling theorem in the time domain instead of, as happens more often, in the frequency domain. The reason behind the choice of presenting the qualitative analysis in the time domain, is twofold: on one side this approach does not require the introduction of the concept of Fourier Transform and spectrum of a signal and, on the other hand, it allows us to give a more intuitive interpretation of the sampling theorem.
At the end of the paper we present some of the best known digital formats, both compressed and uncompressed, highlighting the necessity of compression to reduce the size of a file containing audio and digital video. Finally we present some examples of compression techniques such as the RLE (Run Length Encoding) and Huffman coding.

Keyworks: multimedia, analog-digital conversion, sampling theorem, file compression.

AMS subject classification: 68P30, 94A20, 94A12, 94A09.

[1] Corresponding Author: Dipartimento di Economia, Università degli Studi di Foggia, Largo Giovanni Paolo II 1, I-71100 Foggia, Italy e-mail: luca.grilli@unifg.it.
[2] This paper has been prepared during the period in which the author was a temporary professor at Faculty of Economics at University of Foggia.

1. Main definitions.

In this section we present some definitions that will be useful in the rest of the paper. We recall that with the term *information* we mean anything that can be communicated and that reduces the uncertainty of the receiver of a signal, however a *signal* can be considered as a set of signs leading information; such signs may be graphical symbols, values of physical quantities, etc. An electrical signal is a signal that transfers information through a sequence of values of an electrical quantity, for example a voltage.

A signal is said analog when its amplitude can assume any value within two extreme values. The following figure shows an example of an analog signal, the amplitude that it can assume can be any of the values between the maximum value X_1 and the maximum value X_2.

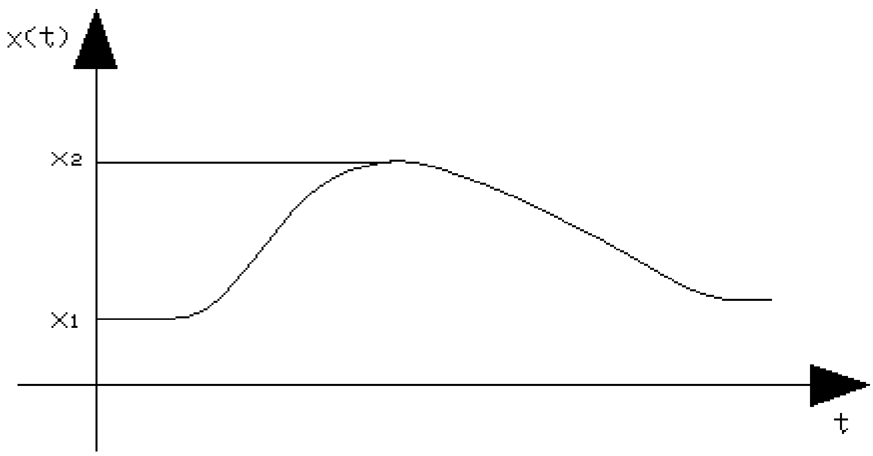

Figure 1: Analog Signal.

A signal is said *digital* if it can only assume one value among a certain finite number of values. The following figure shows an example of digital signal, this signal can assume only one of the four values A, B, C, D. For example, it cannot assume a value between A and B or between B and C or between C and D.

A *binary signal* is a digital signal that can assume only two values. A binary signal is in fact a special case of digital signal.

Note that a binary signal does not necessarily have a zero value and it is worth considering that not always, in digital systems, the lowest value of the binary signal is associated with the logic zero and the highest value is associated with the logic one, the opposite can happen that is the highest value is associated with the logic zero while the lowest value is associated with the logic one.

The following figure shows an example of a binary signal, it may take only two values, either the value A (which in the example is considered zero) or the value B.

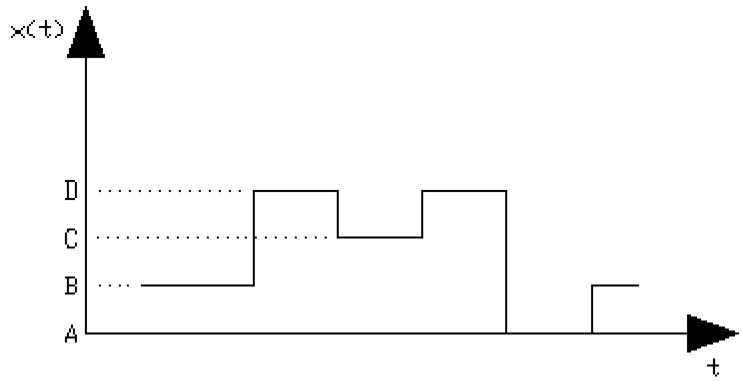

Figure 2: Digital Signal.

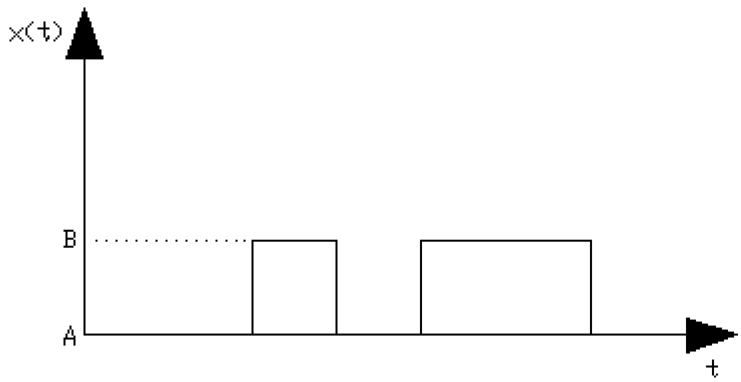

Figure 3: Binary Signal.

2. Analog-Digital signal conversion.

As previously mentioned, the signals that surround us are analog signals, in order to be processed by a computer it is necessary to convert the analog signals into digital or better binary signals.

The following steps in the following order are necessary to complete the analog-digital conversion:

1. Sampling.
2. Quantization.
3. Encoding

Sampling.

The sampling procedure consists in selecting parts of the signal, called *samples*, in predetermined time instants. The following figure shows a schematic diagram that can replicate the sampling procedure.

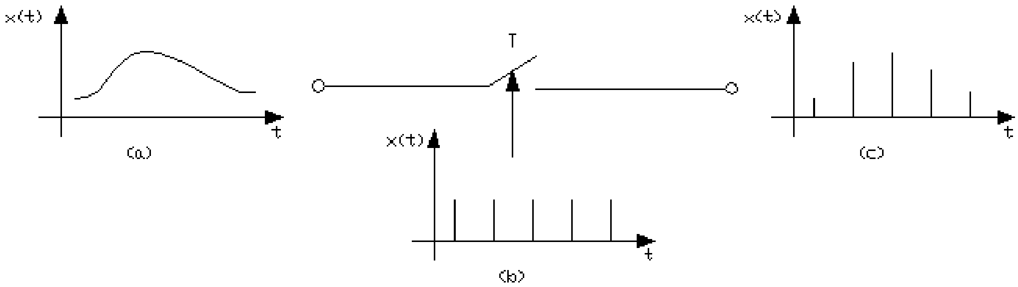

Figure 4: Sampling.

In Figure 5b it is represented a pulses train which closes the switch T at fixed time intervals. During the instants of time in which the switch is closed the input signal represented in figure 5a can pass out; while for all the time period in which the switch is open the output signal is zero. The final result of such operation is the pulses train shown in Figure 5c (the sampled signal).

The Sampling Theorem.

Let us consider the case in which the analog signal represents a sound. In order to be able to listen a sound converted into a digital signal it is necessary to convert it back into an analog signal; to be sure of being able to convert a digital signal into an analog signal it is necessary that, during the sampling operation, the conditions of the sampling theorem are satisfied.

A correct and complete presentation of the sampling theorem would require the introduction of the concepts of Fourier transform and signal spectrum and, in literature, the sampling theorem is presented in the frequency domain. In this paper, we present a simple graphic representation of the sampling theorem in the time domain. This representation allows us to present the sampling theorem in a rather comprehensive and exhaustive way without resorting to the concepts of Fourier transform and signal spectrum.

Let us suppose to consider an analog signal represented graphically in Figure 6a. This signal shows a peak, if we require that the sampled signal replicate this peak it is necessary that the sampling frequency is adequately high.

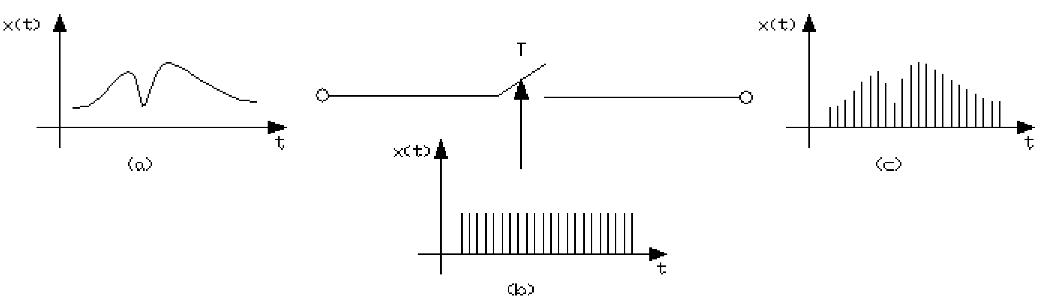

Figure 5: Sampling with adequately high frequency .

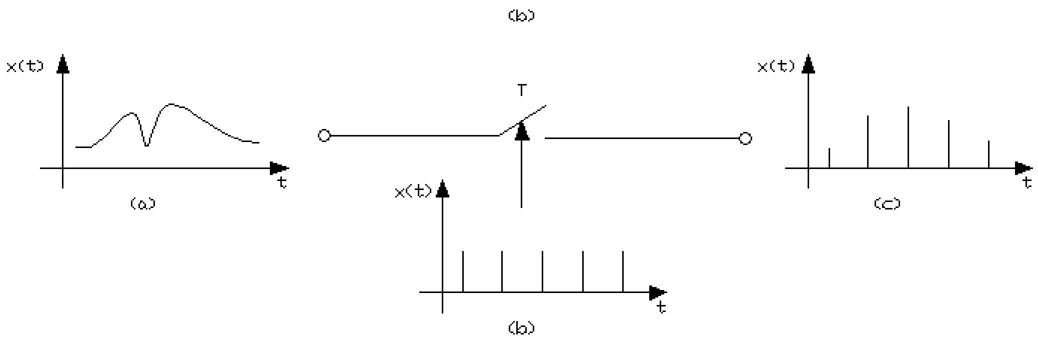

Figure 6: Sampling with low frequency.

In Figure 5, the sampling frequency (Figure 5b), is sufficiently high and, as seen intuitively, the envelope of the sampled signal (Figure 5c), follows the characteristics of the starting analog signal (figure 5a).

On the contrary in Figure 6 the sampling frequency is too low, it follows that the envelope of the sampled signal (Figure 6c), does not follow the characteristics of the starting analog signal.

As a result, a correct sampling that allows us to reconstruct the analog signal starting from its samples, requires that the sampling frequency is sufficiently high in relation to the characteristics of the signal to be sampled.

Quantization.

After the sampling procedure is completed the amplitudes of the samples of the sampled signal assume the same amplitude of the analog signal if considered exactly at the sampling time (that is the time instant in which the switch closes -Figure 4-). The successive encoding operation (better clarified in the following) requires that the samples assume one among a number of possible values.

The quantization consists in associating to each sample, which, as mentioned, may still assume an arbitrary amplitude, an exact amplitude selected among a limited number of possible values.

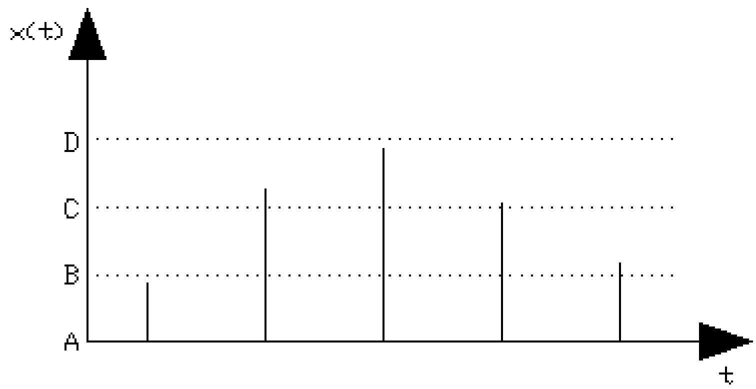

Figure 7: Quantization.

Thus, with reference to Figure 7, the first sample is associated with a sample of size B, since the amplitude B best approximates the amplitude of the sample, the second sample is associated with a sample of size C, the third sample is associated with a sample of size D, the fourth sample is associated with a sample of size C, and so on.

It is worth noting that the quantization process introduces an error known as the quantization error. The quantization error is always present in an analog-digital conversion and can never be eliminated. It should be noted, however, that if the sampling levels are sufficiently close the error decreases and assumes acceptable values.

In figure 8 the quantization error (*e)* is represented, it is computed for each of the samples of the example in Figure 7.

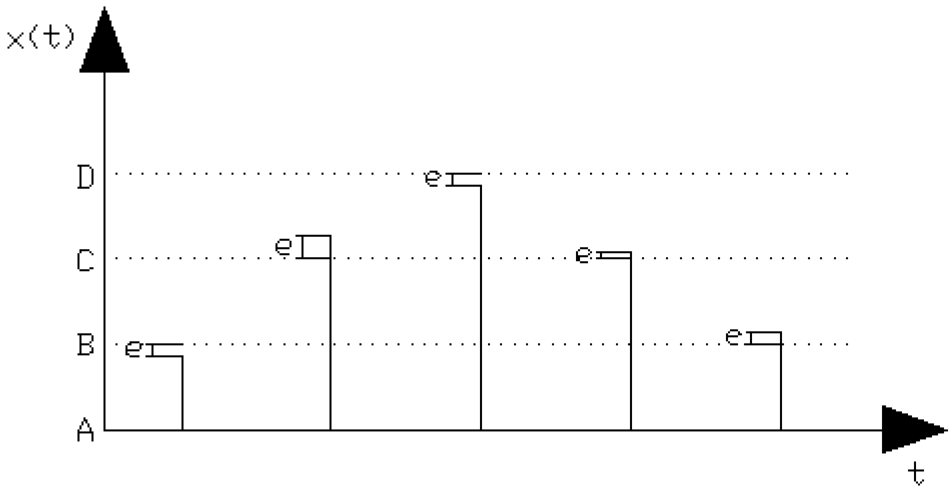

Figure 8: Quantization error

The closeness between the sampling levels depends on the number of bits used for the encoding process, the greater is the number of bits used for encoding the greater is the number of quantization levels and closer will be the quantization levels. With a sufficient number of coding bits quantization levels can be obtained that are close enough to make the quantization error acceptable.

Encoding.

The last step to be performed in the conversion from analog to digital is encoding.

Encoding consists of associating a code to each sample. The more bits used for encoding the greater are the quantization levels that can be represented. However, the number of bits used for encoding must be finite, and therefore can represent only a finite number of values of the amplitudes of the signal, and this is the reason why, as was mentioned before about the quantization, the samples must have finite number of amplitudes and must assume one among a finite number of values and the quantization process is necessary.

Let us consider the case represented in the previous figures. A possible encoding can be the following: samples of amplitude A are associated with the code 00, samples of amplitude B are associated with the code 01, samples of amplitude C are associated with the code 10 and the samples of amplitude D are associated with the code 11.

A	00
B	01
C	10
D	11

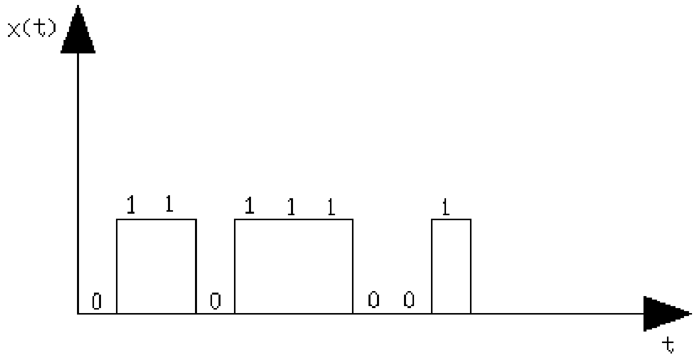

Figure 9: Encoding.

The signal sample that we are converting to binary is a sequence of quantized samples BCDCB which is associated to the code 0110111001.

The following figures show the total absence of similarity between the starting analog signal and the final (converted) binary signal.

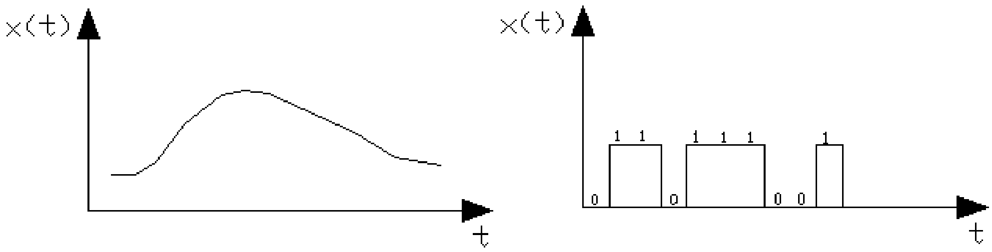

Figure 10: Analog vs Digital.

In the case of an audio signal, in order to listen a sound (analog) converted into a digital signal it is necessary to convert it back into an analog signal, by performing the reverse process, namely the conversion from digital to analog.

3. Analog-Digital signal conversion and application to multimedia data.

Digital Audio

The audio files used in multimedia can be of two types, synthesized sounds live (MIDI file), or audio data files (files WAVE *Waveform Audio File Format*, MP3 files).

Let us consider the files of synthesized sounds live, such as MIDI files. In this case the stored sounds have undergone the process of conversion from analog to digital, but, in these types of files, are stored commands that are interpreted by an appropriate synthesizer that generates sound, this is the reason why MIDI files occupy very little space, about a few tens of kilobytes per minute of music produced. The extension of these files is .mid.

The audio data file presents sounds that are stored after the process of conversion from analog to digital.

When the user executes these files, digital sounds undergo the reverse process of conversion (digital to analog) and it is possible to reproduce the analog starting sound.

An example of the format of these sound files is the WAVE format. The Wave format is an uncompressed format, a WAVE file occupies about 10MB per minute of reproduced sound, (if we consider a sampling at 44100 samples per second, a quantization at 16 bits per sample and that the signal is stereo (2-channel)). A 4 minutes song in Wave format occupies about 40MB. WAVE files have the extension .wav.

Another file format for digital audio is the well known MP3 format. The MP3 format is a compressed format for storing digital sounds that will be described later, at the moment we simply observe that an MP3 file occupies about 1MB per minute of reproduced sound. A 4 minutes song in MP3 format takes up about 4 MB. Files of this type have the extension .mp3.

Images.

The images processed by a computer can be of two types: bitmap or vector graphics images.

A bitmap image consists of a matrix of pixels, each pixel can take a certain colour, the enlargement of the image affects the quality of the image. The bitmap format is uncompressed. For example: a 640x480 image at 24 bits per pixel is an image composed of 640 pixels horizontally and 480 pixels vertically which uses 24 bits to describe the color of each pixel, and so each pixel can assume one of 2^{24}=16.777.216 colors; that image occupies 640 x 480 x 24 = 7.372.800, or 7.372.800/8 = 921.600 Byte = 900kByte.

Some bitmap file formats are: BMP, (uncompressed format that displays up to 16 million colors, 24 bits per pixel, the file extension is .bmp); GIF (Graphic Interchange Format - a compressed format that displays up to 256 colors, 8 bits per pixel, the file extension .gif); JPEG (Joint Picture Experts Group - a compressed format that can display up to 16 million colors, the file extension is .jpg).

The figures below show the comparison between the three sizes just presented, indicating, for each of the images, size and format.

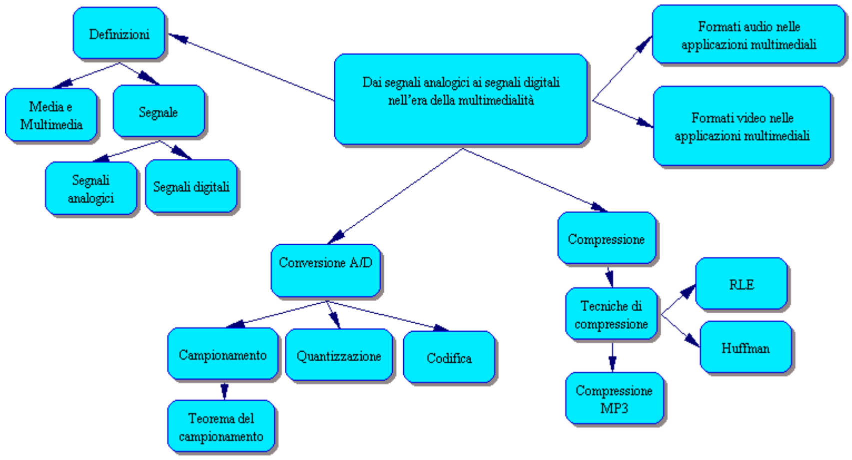

Figure 11: BMP, file 1.18 MB

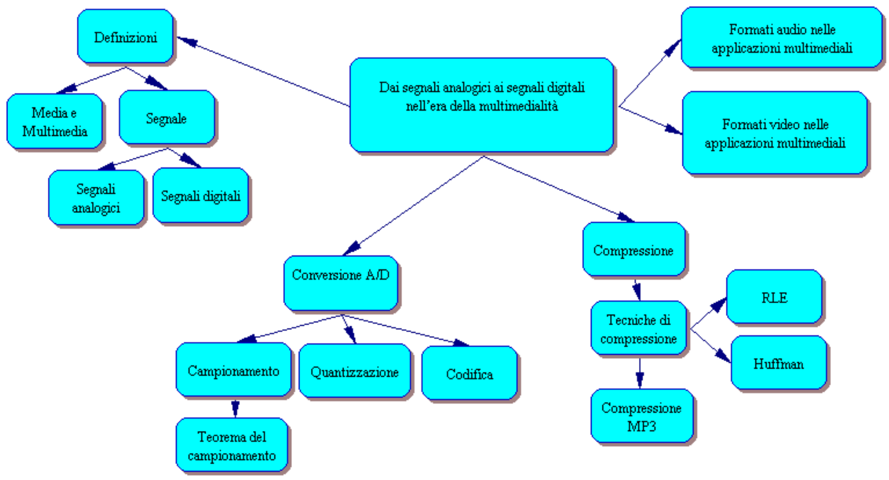

Figure 12: GIF, file 10kB

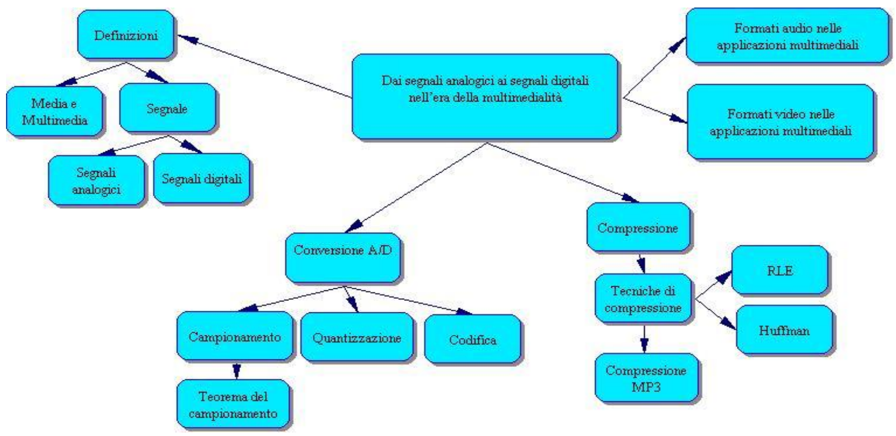

Figure 13: JPG, file 51.8 kB

The vector graphics programs draw images using a set of graphic objects called primitives such as lines, rectangles, arcs, etc.

Objects are stored using numeric coordinates or mathematical formulas.

The enlargement of the image does not affect the quality of the image. Among the advantages we have: more accurate control of line and contour, possibility of manipulating objects (rotate, translate, enlarge…), possibility of changing the colour shades of the objects and text around objects can be inserted.

Animations.

Animations are obtained by projecting a series of successive images called frames. The size of animations files would be prohibitive in absence of compression techniques, for example, considering image frames 640x480 at 24 bits per pixel, each frame will occupy 921,600 Byte (nearly a MB). A video sequence of 30 frames per second, with no compression, will occupy about 30MB per second. Consequently a CD-ROM (700MB) could store about a 30 seconds video, it results that it is essential to use compression techniques.

Data Compression.

Data compression consists in encoding data in a format that allows you to encode information using fewer bits than the original.

In order to compress data a compressor is used while to decompress data it is necessary to use a decompressor hence the name: *codec* (compressor-decompressor).

A compression algorithm is a set of rules that allows to convert a sequence of bits into a sequence of shorter length, so that, starting from such a compressed sequence, it is possible (by means of a second decompression algorithm) to reconstruct the original sequence.

There are two types of compression: lossless and lossy compression.

The lossless compression is used, for example, in the compression of text or data, satellite or medical images, and whenever it is not acceptable to have a loss of information due to the compression procedure. The WinZip for example operates without conversion losses, has numerous algorithms and, depending on the file to be compressed, it selects the algorithm which allows to obtain a higher compression ratio.

The lossless compression procedure achieves lower compression levels, higher levels of compression require the compromise of losing part of information during compression.

The lossy compression is to be used when the lightness of the file is preferred compared to quality. As in the case of compressed formats JPEG, MPEG, MP3.

The video compression techniques are:

- interframe space compression: it eliminates duplicate data that appears within the same frame.
- intraframe time compression: it deletes data that is repeated in successive frames.

RLE (Run Length Encoding).

An example of lossless compression is the RLE technique. It is based on the idea of representing bits of information in compact form, when the same data is repeated several times.

The following example aims to clarify this concept: let suppose to store the string "*ciaooooooo !!!!!!*"; it requires the storage of 16 characters.

The previous string can be stored using the sequence of characters "*cia*7o*6!*" that are only nine (after *cia* there are seven *o* followed by six exclamation marks). The * indicates the number of repetitions of the same character.

This encoding can be profitably used in the representation of images, in fact, many images often contain large portions of the same color, therefore the pixels corresponding to these parts can be compacted by the technique RLE.

Huffman coding.

Huffman coding is an encoding technique based on the idea of representing in a more compact form data that is repeated frequently deriving a table based on the estimated probability or frequency of occurrence (weight) for each possible value of the source symbol.

The Huffman coding is therefore an encoding technique of variable length data.

Thus, for example, if we suppose to have a text consisting of 1000 characters where A repeats 700 times, B 200 times, C 70 times and D 30 times, using a fixed length coding such as the following:

A	00
B	01
C	10
D	11

It is necessary to use 2 bits for each symbol, and then, the number of bits to be used in order to represent the original text will be: *1000 x 2 = 2000 bits.*

Let us consider a variable length coding such as the following:

A	0
B	10
C	110
D	111

In this case the symbol A (the most frequent) is represented by a single bit, B by two bits, C and D by three bits (the less frequent), the total number of bits to be used to represent the text of 1000 characters will be: *700 x 1 + 200 x 2 + 70 x 3 + 30* *x 3 =1200 bit.*

Note that in order for the text to be decoded it is necessary that representation of a character is the prefix of the representation of another character.

References.

1. P.k. Andleigh, k. Thakkar *"Multimedia Systems Design"* - Prentice Hall, 1996.
2. Davis Yen Pen *"Digital Audio Compression"* – Digital Technical Journal Vol.5 No. 2, 1993.
3. Didier Legall *"MPEG - A Video Compression Standard for Multimedia Applications"* - Communications of the ACM, vol. 34, No. 4, 1991.
4. Foley, vanDam, Feiner, Hughes, Phillips *"Introduction to Computer Graphics"* - Addison-Wesley, 1994.
5. D.A. Huffman, *"A Method for the Construction of Minimum-Redundancy Codes"*, Proceedings of the I.R.E., September 1952, pp 1098–1102.
6. C. D. Manning, P. Raghavan, and H. Schtze *"Introduction to Information Retrieval*". Cambridge University Press, 2008.
7. R.J. Marks II, *Handbook of Fourier Analysis and Its Applications,* Oxford University Press, 2009.