Chapter 1

MULTIMEDIA SIGNAL PROCESSING

Examples of Image Processing

Images and Videos Analysis

The four major domains of application for image processing have been globally presented: what are the issues? what are the objectives? what methods to use? and which tools? We will now look more closely at some concrete applications derived from these different domains.

Image and video analysis

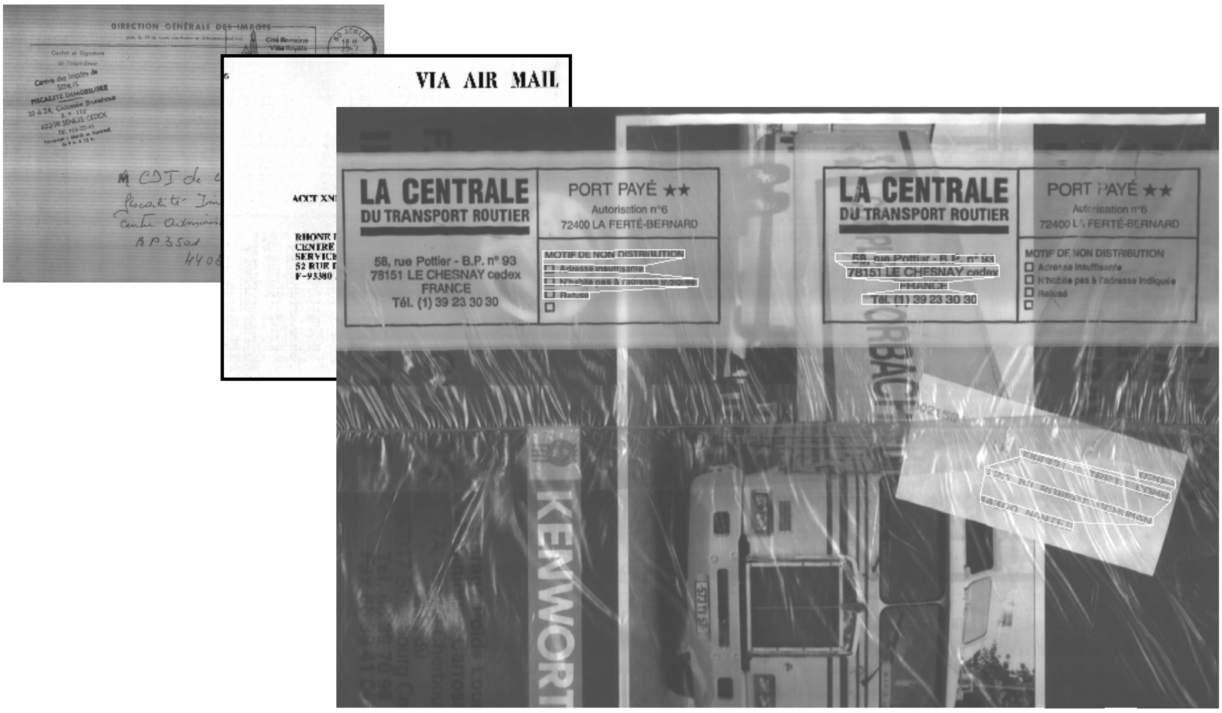
- **Objectives**
  - Objects detection and extraction (segmentation)
    Objects of interest
  - Pattern recognition
    Object classification, object identification
  - Scene analysis and interpretation
    - Relational
    - Quantitative description
    - Qualitative description

The first domain that interests us is image analysis. This covers a large number of potential applications and is probably the domain for which there is the greatest variety of examples (medical analysis, classification of chromosomes, automatic reading, signature verification, remote sensing, segmentation and classification of geographical areas, etc.).
We can however suggest a typical image analysis cycle, which consists of

- ***Detecting*** the different objects present in a scene and extracting them (segmentation, slicing into areas).
- ***Characterizing*** the objects in the picture (size, location, etc.).
- ***Recognizing*** objects and classifying them.
- ***Analyzing and interpreting*** the scene according to the previous results. The scene could for example be described quantitatively and/or qualitatively.

Let us remember though that the applications are quite varied and that the global approach given here may sometimes require changes in the stages of processing, according to the difficulties encountered.

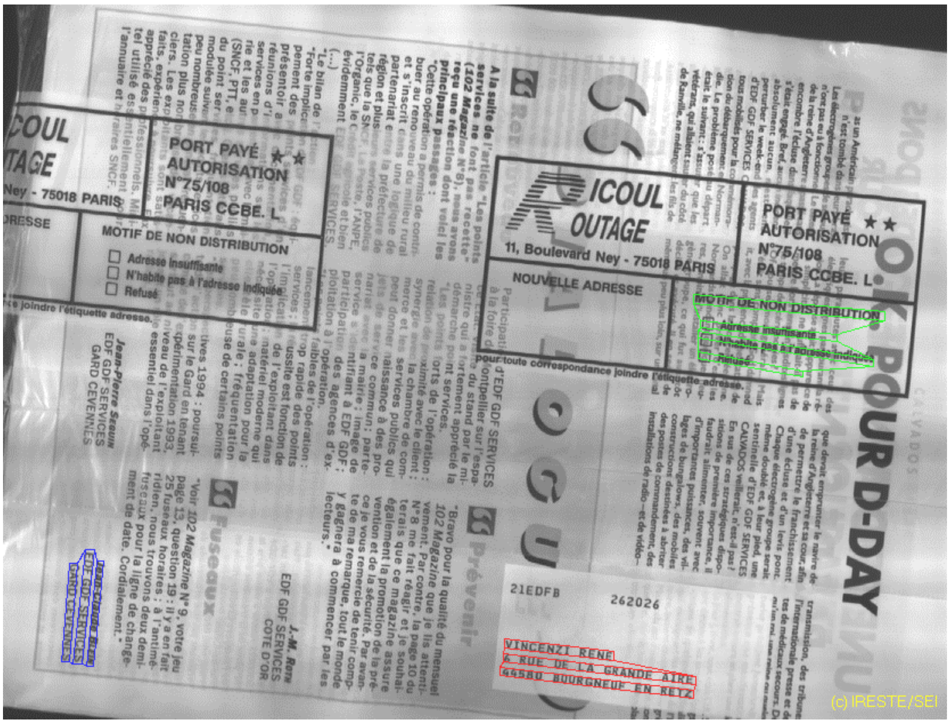
Address block detection and extraction



The example presented in the image above highlights detection and extraction of a delivery address. The application would be used for automatic sorting whose nature would depend on the postal objects and how they were posted. Most industrialized nations use automatic sorting and obstacles vary from one country to another due to differences that exist between postal codes (among other factors); are the postcodes numbers or other characters ? what is their size ? etc.

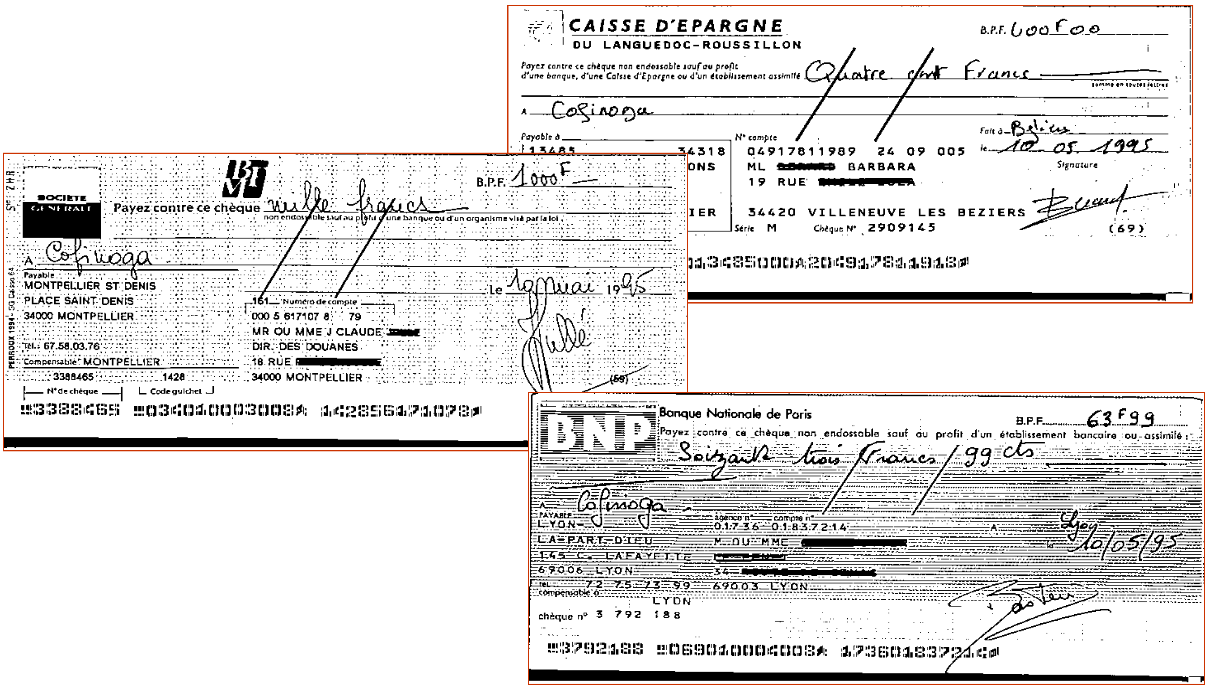
Various processing algorithms exist for this. We can for example notice that the characters of the text form a regular texture containing vertical contours arranged horizontally. By applying directional filters, it is possible to highlight the pixels of the image that are linked to text. We could for example carry out binarization followed by post-processing so we can extract the rectangles surrounding text boxes. Once the text is separated from the image, we could analyze the neighboring components to find those that correspond to text boxes. Once the global processing of the image was carried out, you would then need to apply local processes and consider various constraints (for example, geometrical) to avoid errors. These different operations are described in the next chapters.

Address block detection and extraction



The image above is also the result of an analyzing process. The objective is once again to detect and extract from the image the text boxes linked to the recipient's address. However, the image above presents new difficulties; the background is much less even than the one given previously, as it is made up of text. We must also be able to distinguish, from all the overlapping characters, which ones are linked to the "sender's address block" and which are linked to the rest of the image. However the background text and the address text are perpendicular to each other. It would seem possible to segment the image by applying adapted directional filters. Then we could take up the Characterization – Recognition – Interpretation chain from an image analysis process so we could obtain, as on the picture shown, exact detection of the recipient's address block.

Extracting and reading the amount written on a cheque



The last example presented for image analysis concerns extracting and reading the amount written on a cheque. This is particularly important, considering the industrial and economic potential. Many publications and methods exist concerning this subject, however potential users of this kind of automatic recognition seem to want more reliable results. Because of this, the problem remains open-ended. With this technology, figures and handwritten letters need to be recognized in an environment whose complexity demands the use of appropriate techniques. Apart from the enormous variation in how individuals may form characters, other problems arise such as the non-standardized format of cheques (field position, type of background) and overlapping of the items that make up the amount (figures, baseline, decimal separator).

The main products or prototypes available in France come from SRTP (the post office) with Dassault AT (cheque processing automat that reads only the numerical amount), Matra (MSI) and their LireCheques (cheque reader) product and finally the product called InterCheque from a2ia that can be found in some banks (Société Générale, Banque Populaire) whose recognition rate approaches 70% thanks to a technology based on neural networks and Markov models.

Examples of Image Processing

Image and Video Coding with Data Compression

After these few examples of image analysis applications, we are going to look now at applications linked to image encoding for data compression.

Context

- Extension of digital techniques to many domains of information processing , information capture , storage and transmission

⇒ only one way to store, process and transmit information units independently of their meaning is preferred

- Audio-visual signals have specific properties which require dedicated techniques for representing efficiently them:
  - ➢ data are representation of analog signals

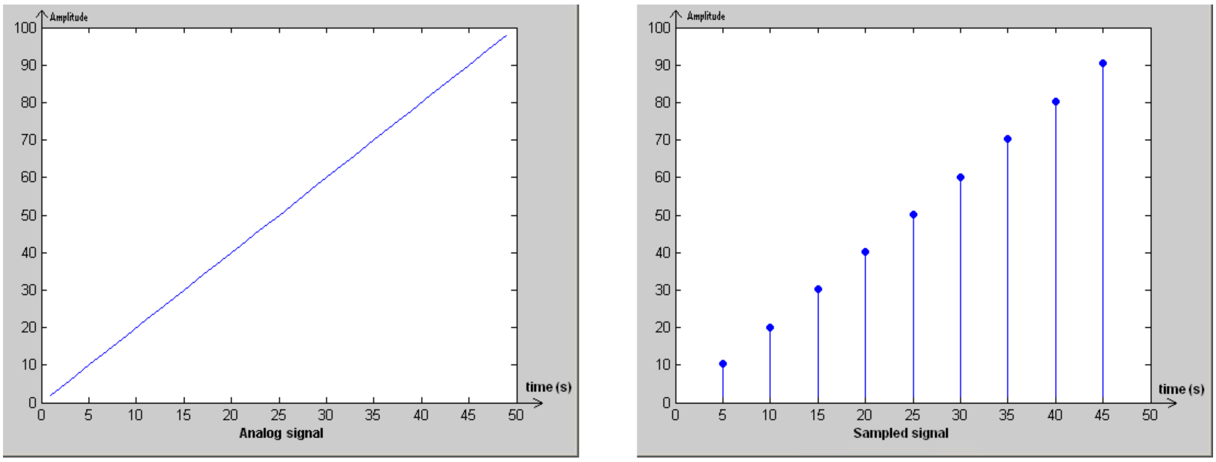
    ⇒ sampling + quantization

  - ➢ large amount of data to represent a small part of the original source of information (e.g. 1s of audio/video)

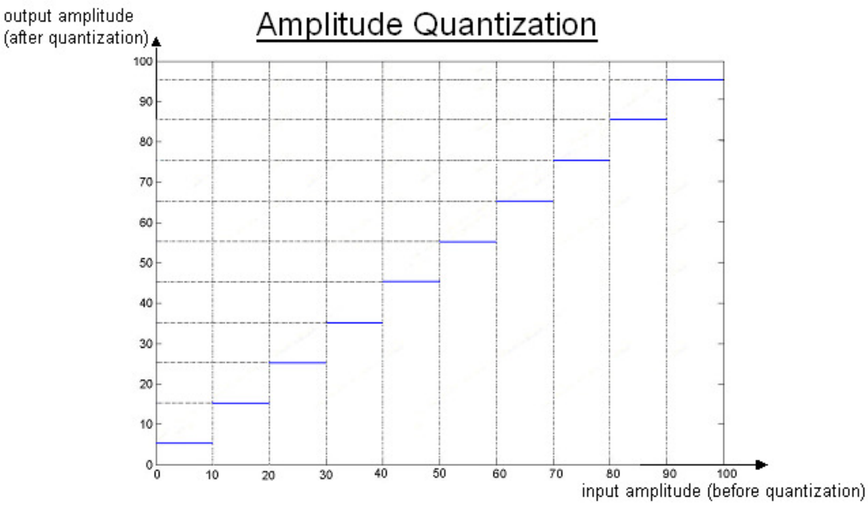
As this was mentioned in the presentation, the problem posed here is the need to greatly reduce the quantity of binary information elements in order to represent images destined for storage or transmission. Although the images are quite varied, meaning that the information they contain may all be very different (color, light, objects), it is still preferable to put in place a unique procedure for information compression encoding, which can be applied to each image, independent of its characteristics.

Furthermore, audiovisual signals are analog (continuous) signals whose representation requires a very important quantity of data (for example for a single second of video footage, you need to be able to store a sequence of 25 images). The effective representation of such signals needs adequate methods to be used.

The binary representation of an analog signal is only possible if the signal has undergone sampling (classically temporal, spatial, etc.) so that one can work on a finite number of samples from this signal. The amplitudes linked to an analog signal also stretch over a continuous interval that will need to be discretized according to different criteria: this is signal quantization. These two stages are presented here briefly for a 1D signal but they will be examined in more detail in the sequel to this module for a 2D image signal.



The figure above represents a ramp type signal before and after sampling. The sampling frequency equals 0.2Hz ($\frac{1}{5}s^{-1}$). The amplitude of this signal varies continuously over the interval [0, 100], so a quantification will be necessary.



The figure above represents an example of quantification: if a sample of the signal has an amplitude included in the interval [10k ; 10(k+1)[, « k » being a natural integer, the amplitude value would be brought by quantification to 10k+5.

After quantization, the data have a discrete representation that it will be possible to encode. Typically we would use binary encoding.

Large amount of data for representing audio-visual information

- Data size requirements
  - **One Teletext page = 25 rows × 40 char** **=> 1 Kbyte**
  - **One A4 text page = 50 rows × 80 char** **=> 4 Kbytes**
  - **One second of telephone (300 Hz - 3400 Hz)** **=> 8 Kbytes**
  - **One second of stereo hi-fi audio** **=> 176.4 Kbytes**
  - **One A4 fax** **=> 1.1 Mbytes**
  - **One image (color-slide quality: 2000 × 3000)** **=> 18 Mbytes**
  - **one second of TV program (video signal)** **=> 21 Mbytes**
  - **one second of HDTV program** **=> 230 Mbytes**

Need data compression

To illustrate the problem that a large quantity of data are required to represent a little information from an image signal, the document above gives an indication of the data sizes required to store and transmit different images, that we regularly find in our environment. These use Anglo-Saxon units, so remember that 1 byte = 1 octet = 8 bits.
For example, a single faxed sheet of A4 paper needs a capacity of 1.1 Mb to transmit the basic image. These different examples show how important it is to make data compression encoding before sending or storing an image.

Contex

- Need to mix in the same document different types of information
  - Text (common ; intrinsically structured ; semantic level)
  - Data tables (common ; intrinsically structured)
  - Graphics (vectorized data ; structured information)
  - Audio (list of many temporal samples ; low level representation (non structural & non semantic))
  - Image (list of scalar/vector samples issue from a scanning of 2D plane ; low level representation (non structural & non semantic))
  - Video (set of Image data along the time axis; same properties as Image data)

Apart from the fact that a large quantity of data are required to represent a little information from an image signal, the representation of this signal is also made difficult but the very nature of the image signal. This signal mixes many and varied types of information. Consider that an image may contain text boxes, data tables, pictures and so on.

The "Compression size/Quality" compromise also directly depends on the image content. If we consider for example a geographical map accompanied by a legend, it is not possible to simply recognize shapes and the borders of the various continents, seas and oceans, once decoding has occurred. The writing must not be too deteriorated so that legends and names on the map can still be easily read by any user.

Main aspects of image compression

- Type of images to compress ? which services ?
- Image quality:
  - lossless image coding
  - or lossy image coding

    in that case: with or without visible distortions?
- ***Compression rate / Quality***
- Codec complexity : coder end , decoder end
- Error sensitivity (error robustness):

  effects of transmission errors on the decoded image
- Fixed or variable compression rate:

  $\Rightarrow$ ***transmission requirements***

Before applying data compression encoding to an image, one must ask several, often trivial question but which have an important influence on the results obtained after compression:

- What type of image are we dealing with? What will it be used for?
- Do you need an exact encoding (reversible but highly constrained which gives a low compression rate)? Otherwise, will any deterioration be visible to the naked eye (here again, the compression rate is affected)?
- The compression rate must be as strong as possible to save sufficient storage space and transmission time as possible, but as that affects the quality, what type of balance are we striving for?
- A rise in the compression rate increases the decoder complexity and consequently the calculation cost for the system. Here again a suitable compromise must be found depending on the intended usage.
- How well does it need to stand up to errors? Transmission errors may have strong, direct repercussions on the decoded image.
- Will the compression rate be fixed or variable? What will the transmission conditions be like?

When it comes to the crunch, the required usage of the encoding is as important as the encoding itself. Before starting, you must define the compression rate and depending on your objectives, choose the compression size/quality ratio that you want to obtain.

In Summary

✠ **Very « hot » applications**

· huge volume of information
· wide spread of applications and services

digital photography (MMDB)
↓
HDTV

✠ **Multiple dependences**

· which images (≈ types & services)
· which resolution, frame rate/s, ...
· which quality required (*acceptance level*)
· which communication network

✠ **Multiple requirement scalabilities**

(multiscale, multiple qualities ...)

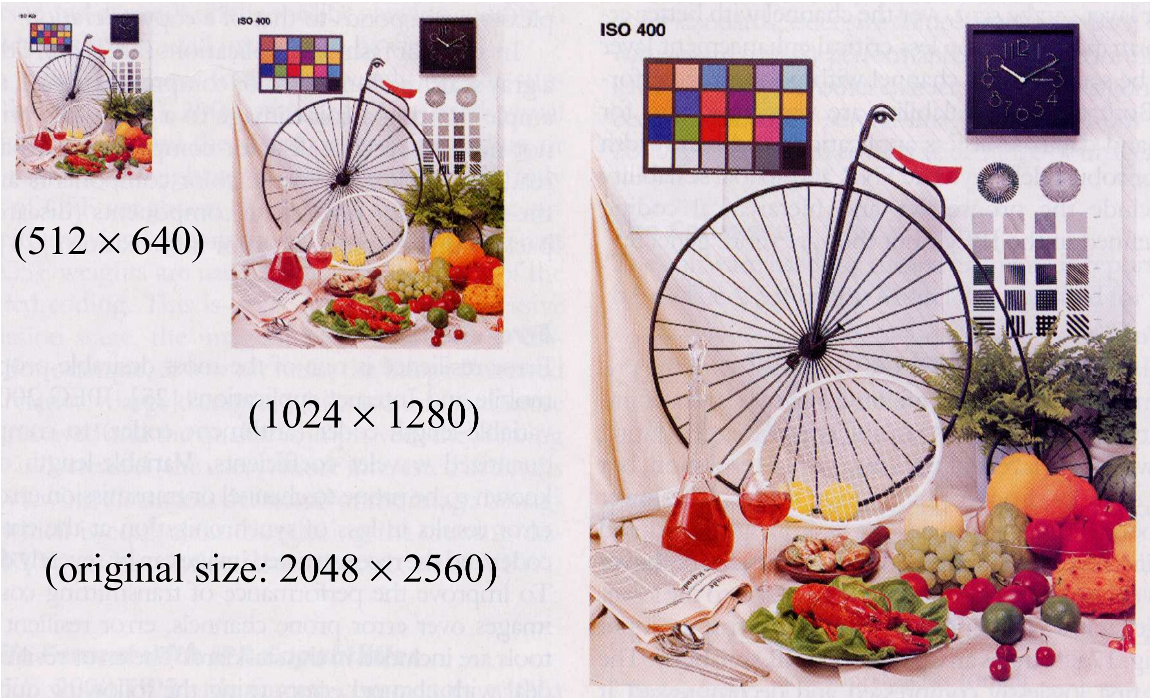
■ Other functionalities than pure coding: object manipulation,...

To sum up, what we have put forward leads to the conclusion that setting up image compression encoding is a complex procedure. On the one hand, you must consider a very large volume of information and a wide range of applications and services. On the other hand, the encoding itself will depend on a number of parameters: the image being processed (type, content, usage), its resolution, the quality required for the intended usage and the accepted amount of deterioration, the transmission network to be used, and so on. In addition to these difficulties, there may be complexity, linked to features other than the encoding itself (object manipulation, etc.).

The next section presents the case of encoding for a high definition image; by its nature this type of image creates a constraint that has a direct effect on the compression complexity.

Example of HD image: Image *Bike*

(decoded at 3 scales from the same encoding process)



The figure above presents a high-definition (HD) image, "Bike" which, after compression encoding, has been decoded to 3 resolution levels and different sizes, but using the same process. The presentation is in a pyramid form, from the highest resolution (to the left of the picture) to the finest (on the right). The image with the highest resolution appears to be sharp, and colors and details are clearly rendered. Deterioration does not seem to be perceptible by the human eye in this case. However, for an HD image such as "Bike" with a 2048×2560 resolution, new compression encoding problems appear.

Note: be careful in relation to the representation given on the illustration above, because if the display does not keep the same resolution as the original, there may be under-sampling.

Problems

- Digitalization of audiovisual signals
  - specific problems with HD images / video
    - image sensors (multi-spectral sensors, resolution, sensitivity, high speed luminance capture)
    - quantization (high precision for audio / some imagery systems, high speed conversion)
- High data compression rate but quality saving !

  for storage and for transmission (reduced bandwidth)
- Others requirements
  - multiple adaptations (transmission rate, scalability, errors robustness, packetization)
  - access to partial contents, content description, …

→ ***JPEG2000, MPEG4, MPEG7 standards***

When we digitize audiovisual signals, problems specific to HD image appear:

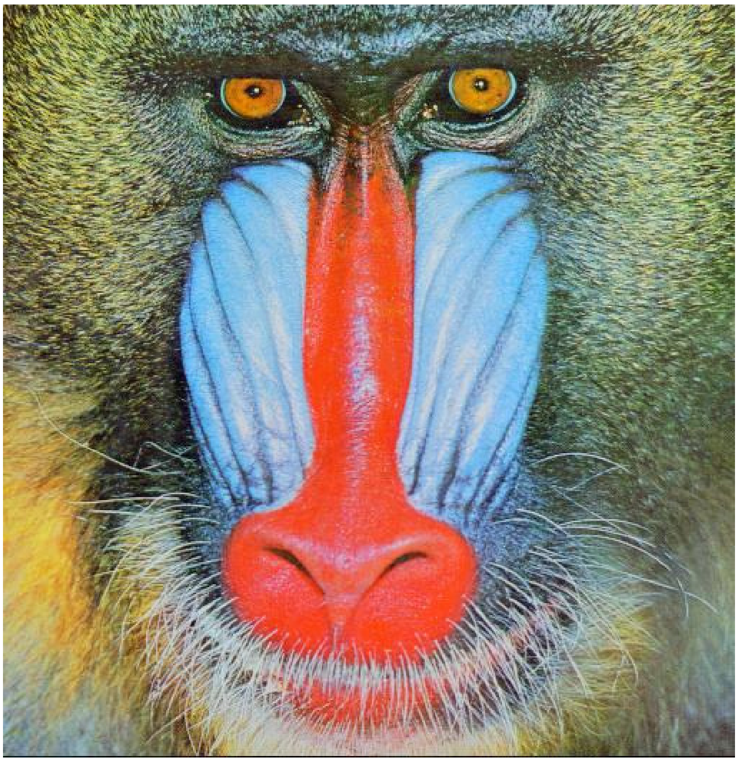
- the type of capturing device used (multi-spectral types, resolution, sensitivity, high lighting speed capture, etc.)
- The way in which quantification is chosen.

Furthermore, we are seeking to attain high compression rates, considering the intended applications. However, as this is a high definition image, we wish to keep a high degree of quality. The compromise between compression rate and quality is limited. Finally, by taking into consideration other constraints, such as the necessity to adapt the encoding (according to error resilience, transmission mode, etc.), it has become necessary to create compression standards that may meet the different constraints caused by these problems and which we can apply to images independent of their nature. The most common standards for pictures and video are JPEG, GIF, JPEG2000, and MPEG4.
The next section presents an illustration of different results obtained for a picture called "Mandrill" which was encoded in JPEG with different parameters and compression rates.

Example of JPEG coding : original of *Mandrill*

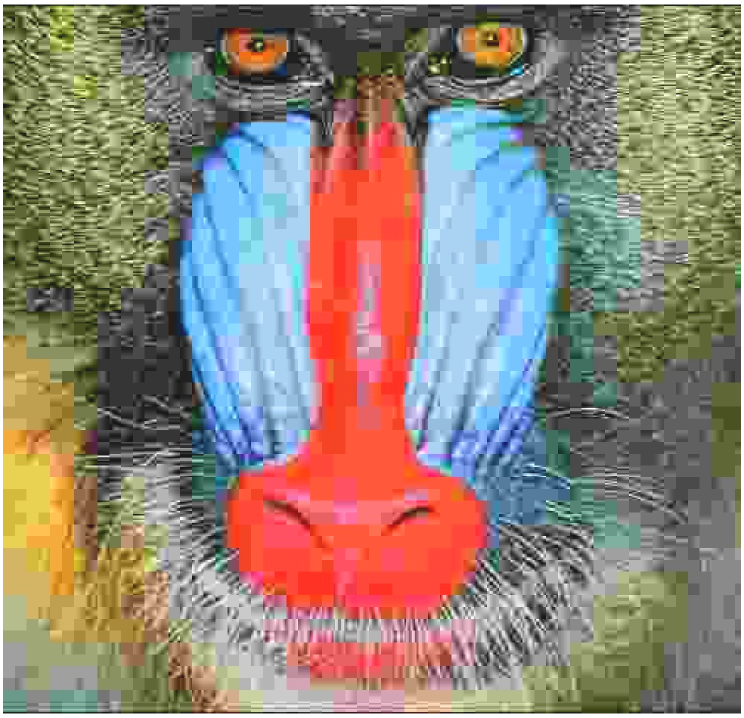
Original:
512×512
$\Rightarrow$ 24 bpp

$\Rightarrow$ **768 Ko**



Above is the original "Mandrill" picture with a size of 512×512 pixels. A pixel is coded on 24 bits (three times 8 bits because the initial representation is in full RGB colour, so there are 2^{24} possible levels for each pixel). The image is thus represented by a volume of 512×512×24 = 768 Ko. We are going to put this image through a series of JPEG encodings, with different parameters, to compare the visual quality of the results and the compression rates obtained.

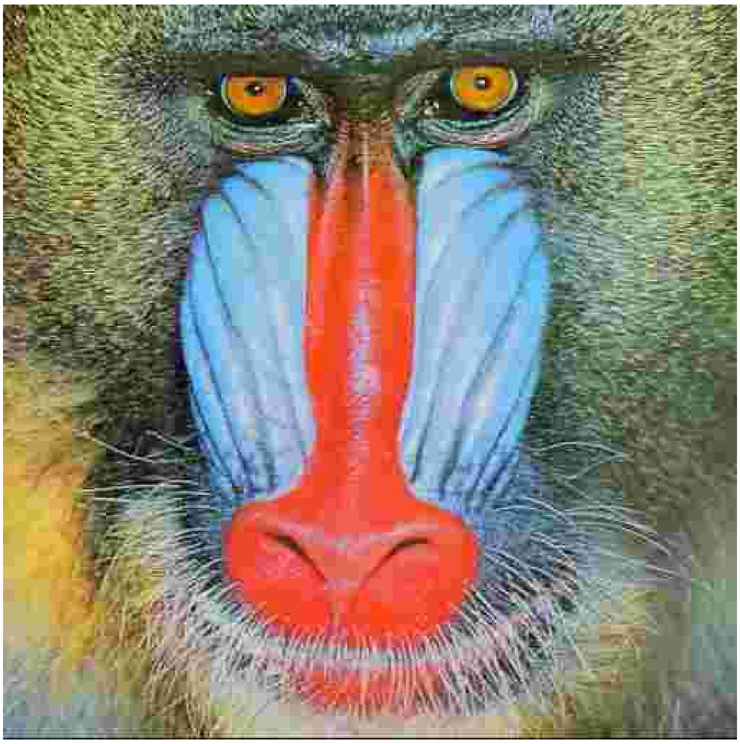
Example of JPEG coding (q = 5 ; 9 Ko)



τ_C =768/9
≈ 85

Above is the "Mandrill" picture that has undergone a JPEG encoding. Quantization was carried out with a low quality factor q (q=5). The compression gain is relatively high. The size of the data describing the image is only 9Kb. There is a high compression rate τ_C, worth 768/9=85. However, the image has lost much of its visual quality. Colors and shapes are rougher than on the original image and a blockiness effect quite clearly appears over the whole image (blocks of pixels are represented by practically the same color).

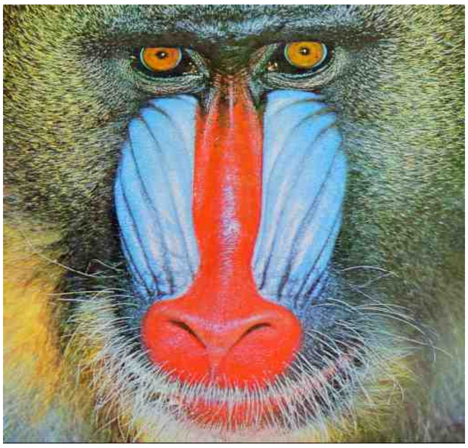
Example of JPEG coding (q = 10 ; 17 Ko)



τ_C =768/17
≈ 45

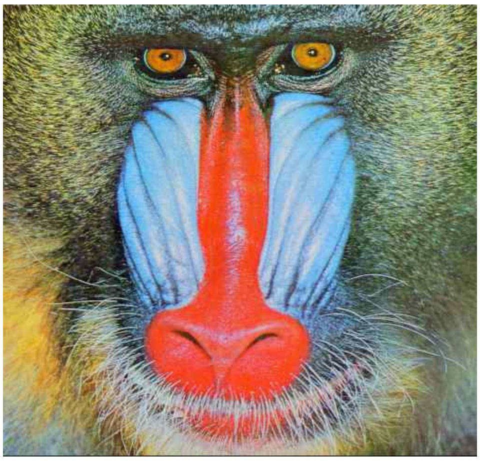
Another example of a result with a higher quality factor, although it remains quite low (q=10). The compression rate is lower than the previous example (τ_C = 45). The image quality has improved and the blockiness effect is less apparent, although it can still be seen on the vivid reds and blues of the baboon's face. The range of colors is lower than the original image, but it gives a better representation of the monkey's fur than the previous encoding. Note that by doubling the quality factor q, the compression rate is divided by 1.88, which is nearly half.

Example of JPEG coding (q = 25 ; 32 Ko)



τ_C =768/32 ≈ 24

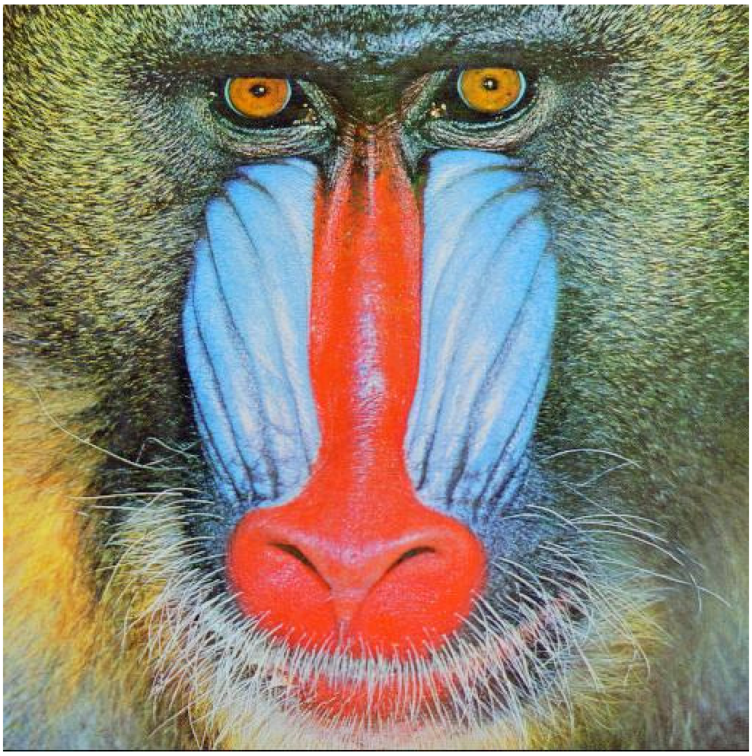
Example of JPEG coding (q = 35 ; 40 Ko)



τ_C =768/40 ≈ 19.2

Two more results obtained by giving precedence to image quality rather than the compression rate. However, this time, the compression rate was divided by only 1.25. The evolution of the compression rate according to the quantization is generally not a linear function.

Example of JPEG coding (q = 75 ; 76 Ko)



τ_C =768/76 ≈ 10.1

Finally, the image above presents a result where the image quality is paramount over the compression. The number of bits required to store the encoded image is 10 less than what is required to store the original image. It is however difficult for the human eye to detect the loss of picture quality: it is hard to differentiate between this and the original image in terms of the color range and the sharpness of colors and textures.

Here we have presented image processing under its different aspects. It was created from new requirements made by information technology, and in turn uses those same technologies to ensure its rapid development. With its expansion, four wide domains of application have appeared: restoration, encoding, analysis and image synthesis.

In this course, the domains of analysis and encoding have been accompanied by examples of concrete applications, in order to give an initial view of the possibilities and results offered by the various image processing.

However, in the real world, the images that surround us cannot generally be processed directly and they are not digital. They need to be digitized in order to be processed with adapted algorithms and systems. This digitizing stage will be looked at in the next resource (Image digitization).