Chapter 1

MULTIMEDIA SIGNAL PROCESSING

Image digitalization

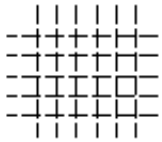
Sampling and Quantization

Sampling and Quantization

• Computers handle discrete data.

• Sampling

- Sample the value of the image at the nodes of a regular grid on the image plane.

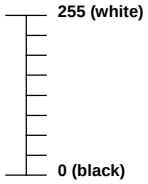


- A pixel (*picture element*) at (m, n) is the image intensity value at grid point indexed by the integer coordinate (m, n)

• Quantization

- Is a process of transforming a real valued sampled image to one taking only a finite number of distinct values.

- Each sampled value in a 256-level grayscale image is represented by 8 bits.



In the previous resource (Image processing – Examples of applications) we briefly mentioned on the fact that digitalization implies a double discretization:

- Firstly from the domain of the image definition: the 2D spatial domain for still images (respectively the 2D spatial domain and time for moving images): this is ***sampling***.
- Secondly from the image signal amplitude: the luminance signal in the case of monochrome images, the three color signals in the case of color images (representing “Red Green Blue” (RGB) or "Luminance, Chromatic1, Chromatic2": Y, Cr, Cb): this is ***quantization***.

We can only then speak of digital (or digitalized) images once this double discretization has been carried out, in space and in amplitude. The basic (or canonical) representation of an image corresponds to a 2D table, in which each element corresponds to a pixel. For a monochrome image, each pixel is encoded on 8 bits, for example, and in this way could take 256 different values (effect of the quantization). In the case of a color image, the pixel will have three components that will represent the “Red Green Blue” (RGB) components or "Luminance, Chromatic1, Chromatic2" (YCrCb)) according to the chosen model.

Note: if R, G, B each take their value from [0, 255], and Y too, the two chromatic components Cr and Cb take their initial value from [-128, +127]. In practice, these are encoded by adding an offset of 128 so they have both a range dynamic of [0, 255].

A Typical Image Processing Scenario

Recall: What is an Image?

– A function I(x,y) over two spatial coordinates of a plane

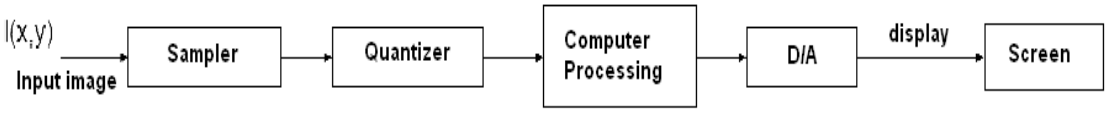
– To obtain digital data for digital processing:

• *Sampling (spatially) and Quantizing the luminance values*

• *Can be done by single chip*

i.e. Charge-coupled Device (capteur CCD)

Typical image processing scenario



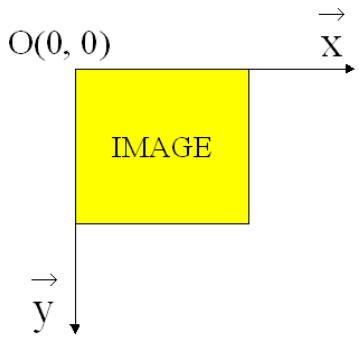
The figure above shows the complete chain used to carry out any type of image processing:

- The image is represented by a function of two continuous variables, these variables corresponding to spatial coordinates (x, y).
- The first function is *Sampling*: we sample the continuous image signal (discretization of the spatial coordinates).
- The result is then injected into the *Quantization* function: for a monochrome image, we often choose 256 luminance levels.
- The image is now digitalized, so we can apply the required image processing (*cf. Processing*), which could be binarization, contrast enhancement, etc.
- The processed digital image is finally sent to a digital-to-analog converter (DAC) to obtain a signal that can be displayed on a monitor.

The typical functional chain for image processing involves an operation for digitalizing the images that we will then process. This operation is sampling followed by quantization. These two notions were reviewed in connection with signal processing and electronics. However the design and achievement of these two stages in imaging are quite different because images are 2D spatial signals and their representation for color images requires us to consider to digitalize the three components.

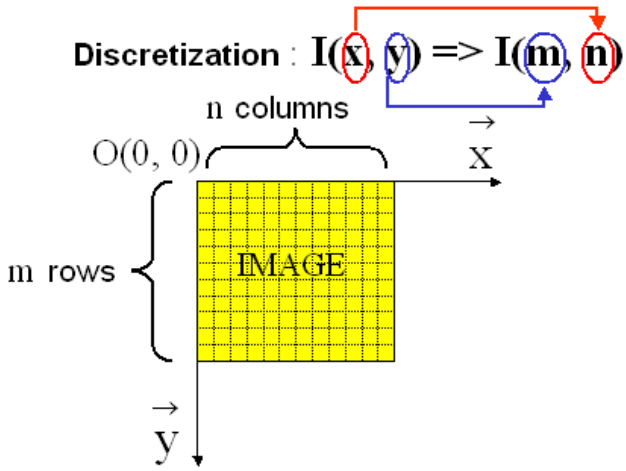
Two important remarks:

1 – The input signal I(x, y) is a 2D image. In images, the axis (Ox) and (Oy) are represented and oriented as shown below:



The (Oy) axis is directed from the top of the image towards the bottom.

2 – Once these two axis are discrete, we obtain « m » rows and « n » columns:

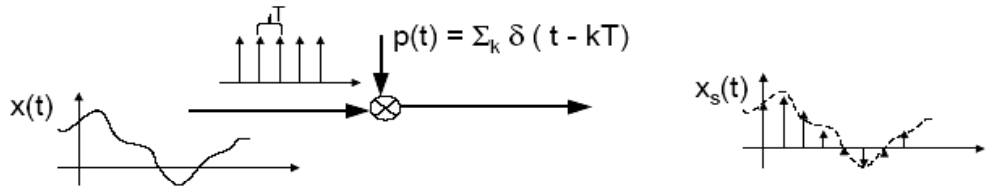


In image processing, **we need to speak of an image I(m, n)** and not I(n, m). **The coordinate order is modified** because « m » corresponds to the discretization according to (Oy) and « n » according to (Ox).

Recall: 1-D Sampling

- **Time domain**

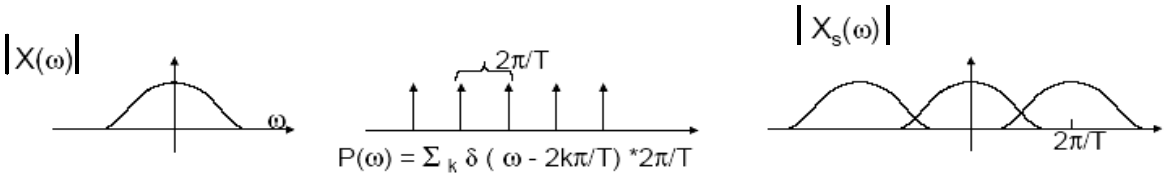
  - Multiply continuous-time signal with periodic impulse train



- **Frequency domain**

  - Duality: sampling in one domain ⇔ tiling in another domain

    - the Fourier Transform of an impulse train is an impulse train (proper scaling & stretching)



Before moving on to sampling a 2D signal, we should recall some results about 1D signal sampling. The most widespread type of sampling in electronics and signal processing is time-based sampling. The figure above presents the various results issued by such processing in the time and frequency domains.

Note: $\omega=2\pi f$, with ω: pulsation in rad.s^{-1} et f: temporal frequency in Hz

- In the time domain: starting from a signal « x » depending on time, you must choose a sampling period « T ». The sampled signal $x_S(t)$ will thus be represented by its values at instants t = T, t = 2T, t = 3T, … Practically, to create such a sampling, you must carry out the product of convolution of the analog signal x(t) with a series of impulses p(t). These impulses correspond to the sum of the Dirac pulse (Dirac delta function) at time kT, k being an integer.

- In the frequency domain: there is a duality between the time domain and the frequency domain. The fact of sampling a signal in one of these domains causes the periodization of the same signal in the other domain. Here X(ω) (resp. P(ω)) represents the frequency spectrum of x(t) (resp. p(t)). The sampling period T in the time domain has thus created the periodization (period 1/T) of the spectrum in the frequency domain (see the representation of $X_S(\omega)$).

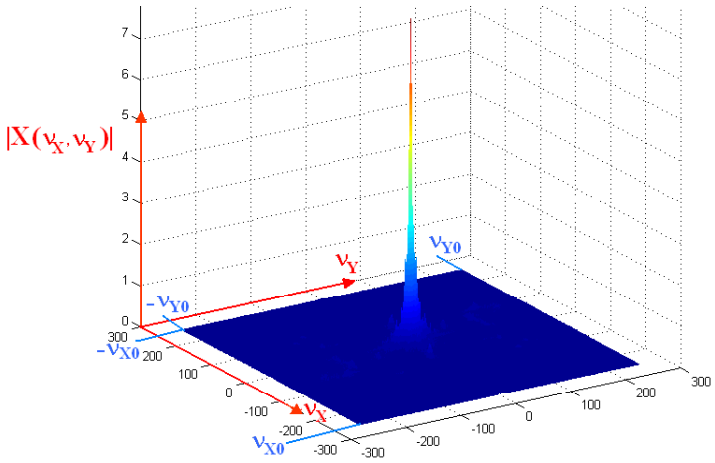
This duality between the two domains can create aliasing effects due to spectrum overlapping, so certain constraints must be met.

Extension to 2-D sampling (1)

- Bandlimited 2-D signal

  - its Fourier transform is zero outside a bounded region is spatial frequency domain: $|\nu_X| < \nu_{X0}$ et $|\nu_Y| < \nu_{Y0}$

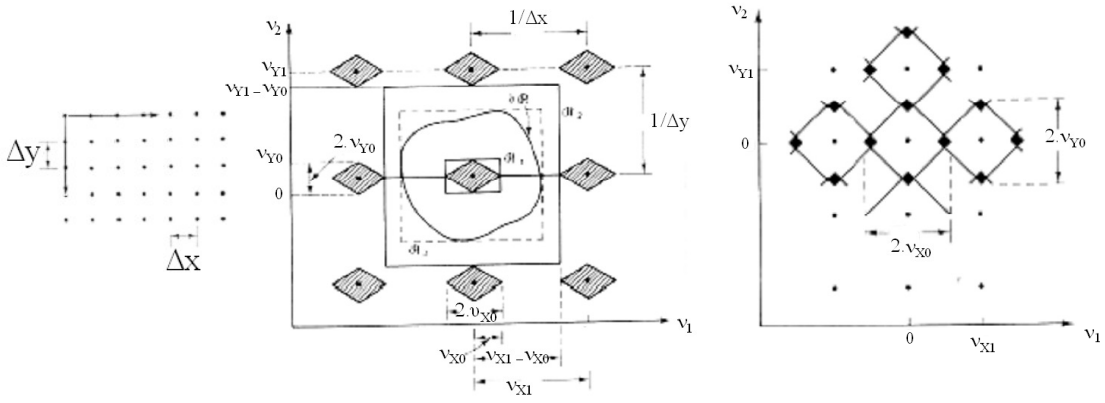
Example : the image *Barbara* and its spectrum



The problems caused by the sampling of two-dimensional signals are most likely similar to those of one-dimensional signals. The Nyquist criterion is applied in the same way. However, an interpretation in a two-dimensional space is necessary to better understand the phenomena linked to sampling, such as the effects on the sampling pattern geometry.
The figure above represents an example using a Fourier transform from a 2D limited bandwidth function (image *Barbara*). In this case we can work with a bounded spectrum. The spectrum spreads over a spatial frequency domain defined by two basis vectors.

Extension to 2-D sampling (2)

- 2-D Comb function
  - $\mathrm{comb}(x, y; \Delta x, \Delta y) = \sum_{m,n} \delta(x - n.\Delta x, y - m.\Delta y)$
  - $\mathrm{FT} \Rightarrow \mathrm{COMB}(\nu_X, \nu_Y) = \mathrm{comb}(\nu_X, \nu_Y; 1/\Delta x, 1/\Delta y)/\Delta x.\Delta y$
- Sampling vs. Replication (tiling)
  - Nyquist rates ($2.\nu_{X0}$ et $2.\nu_{Y0}$) and Aliasing



The variations according to $\vec{x}$ (Δx) and the variations according to $\vec{y}$ (Δy) are going to define the plane on which the 2D image is located. The sampling structure is regular (figure on the left) so the arrangement of sampled points (pixels) is regular in the plane (x, y). The position of the point at coordinates (m, n) is given by: nΔx + mΔy.

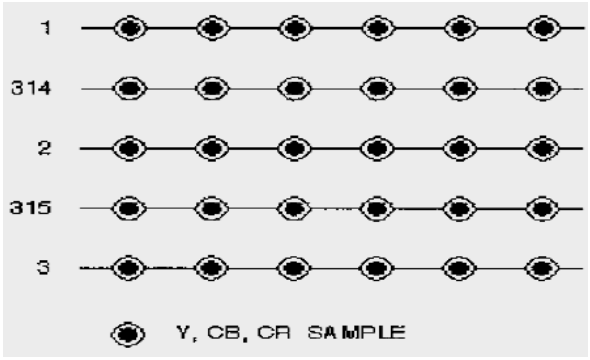
In the frequency domain, points are defined by their "frequential" coordinates ν_X et ν_Y. From a function *f*(x, y), to which we associate the amplitude spectrum $F(\nu_X, \nu_Y,)$, we will obtain after sampling:

- $f_e(x, y) = f(x,y).\sum_m \sum_n \delta(x - n.\Delta x,\ y - m.\Delta y)$
- $F_e(\nu_x, \nu_y) = \left(\frac{1}{\Delta x.\Delta y}\right) \sum_m \sum_n F(\nu_x - n.\Delta\nu_x, \nu_y - m.\Delta\nu_y)$

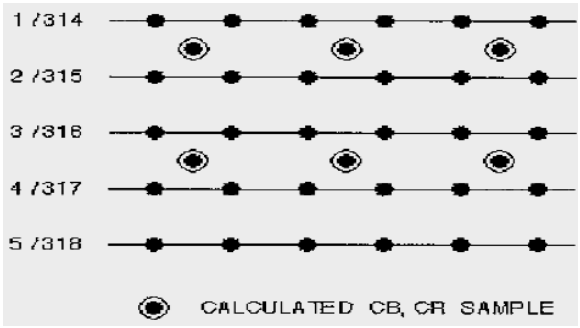
For a bounded spectrum image signal, we can reconstruct the original (non-sampled) image, from the ideally sampled image if the original signal spectra do not overlap (central figure in opposition with the image on the right). To do this, we use an interpolator R of constant spatial frequential gain (spatial frequencies), equal to ΔxΔy in the F domain and zero outside of F, so that this has no overlapping with any of the translates of F. We then have the ideal two-dimensional linear interpolator: the Nyquist spatial interpolation filter.

Sampling Lattice in Color Images and Videos

4 : 4 : 4 Sampling Structure



4 : 2 : 0 Sampling Structure



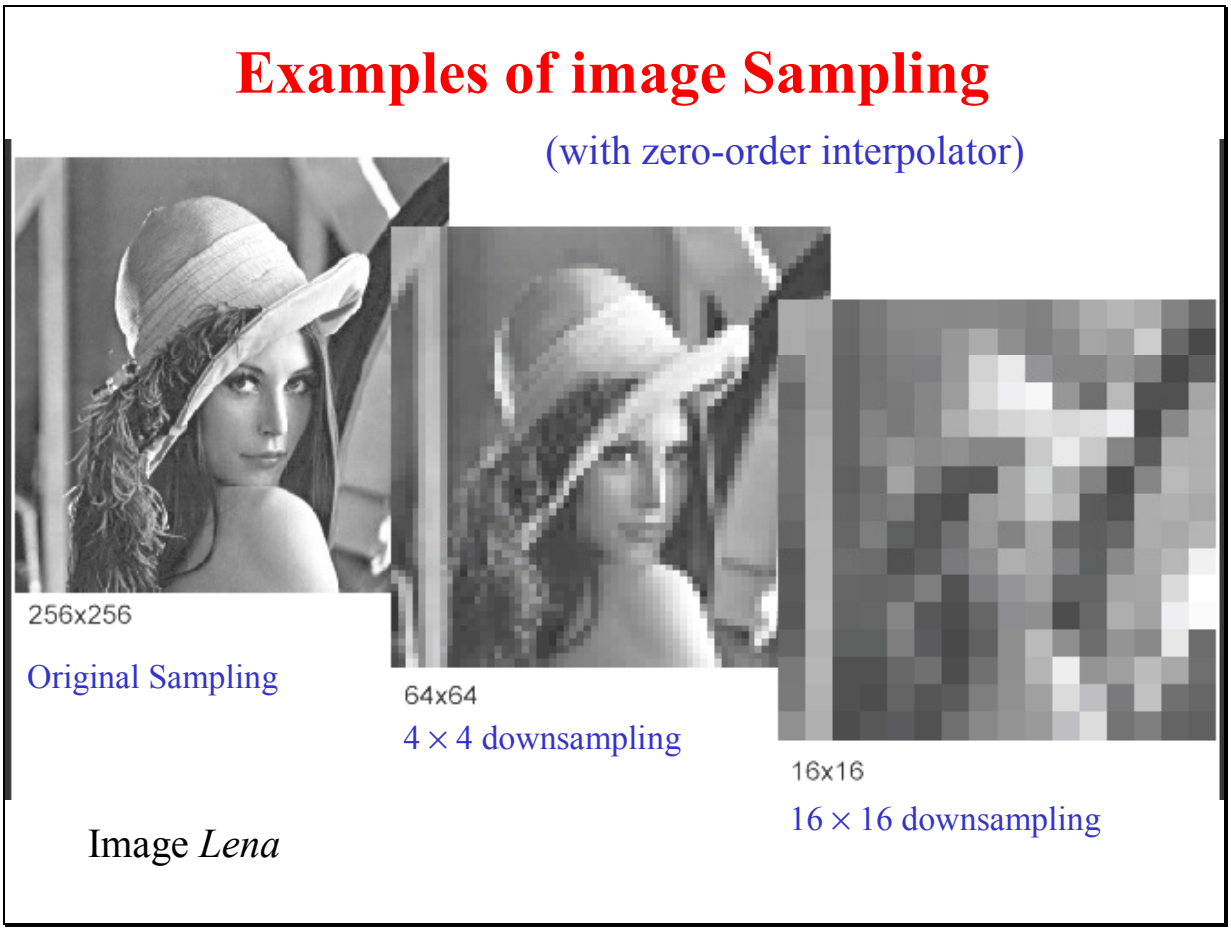
Orthogonal sampling structures

The figure above presents two common structures for orthogonal sampling of color images:

- A 4 : 4 : 4 structure: For each pixel, we have a luminance component and two chromatic components, Cr and Cb. In this case there is no sub-sampling of the chromatic components.

- A 4 : 2 : 0 structure: For each pixel, we have a luminance component (Y) but the two chromatic components, Cr and Cb are only for a group of 2×2 luminance pixels. The chromatic resolution is divided by a 2 factor along the two spatial dimensions. This format (used in DVDs) reduces the vertical and horizontal resolutions.

For these sampling structures, we transformed the RGB representation of the image into the YCrCb representation. The reduction in the spatial sampling frequency of the two chromatic components is made possible by the fact that, in natural images, the two chromatic components have a spatial frequency bandwidth much narrower than that of the luminance component.



In this figure, we have represented the image "*Lena*" sampled with two different sampling structures. The image on the left is the reference image (spatial dimensions: 256×256 pixels). The second image is sample with a sampling frequency four times lower for each of the two spatial dimensions. This means it is 64×64 pixels. For display purposes, it has been brought to the same size as the original using a zoom. This is in fact an interpolation of zero-order (each pixel is duplicated 4×4 times, so that on the screen it displays a square of 4×4 identical pixels). A pixelization (staircase) effect clearly appears, but it is still possible to distinguish the different elements in the image (the face, the hat etc.). The third image (on the right) shows the result after a downsampling of 16 in relation to the original, following the two spatial directions. Again in order to see it at the same size, we have used zoom factor 16 (interpolation of zero-order creating for each sample a reconstruction by 16×16 pixels of the same amplitude). The image is strongly pixelized and we can hardly distinguish the objects in the scene.

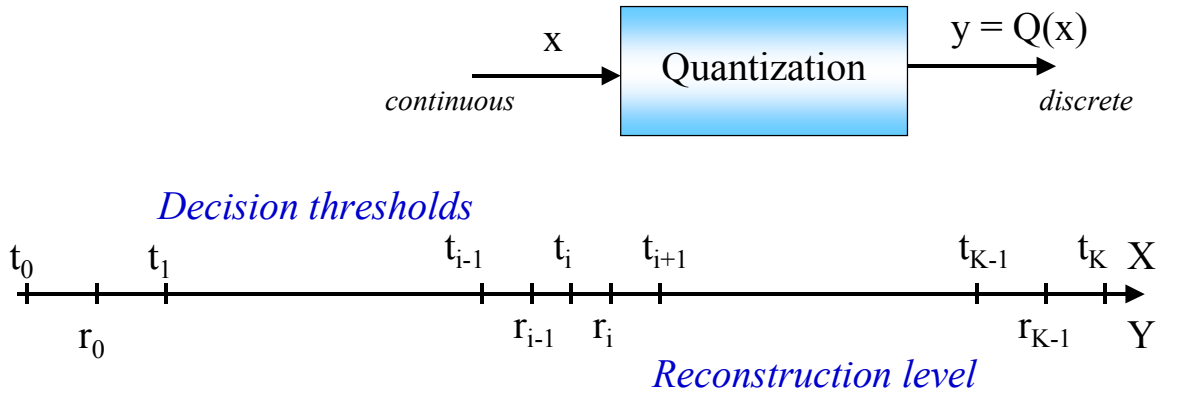
Now that we have looked at the two-dimensional sampling of a 2D signal (in this case an image), we now have to determine how to proceed with the quantization of the sampled image in order to get a digital image.

Video and Image digitalization

Quantization

Quantization

- **Scalar quantization**



- **Quantization function**

for $t_j \le x < t_{j+1}$ then $y = r_j \quad \forall j = 0, \ldots, K-1$

and $t_0 = x_{min}$; $t_K = x_{max}$

The image is now sampled in the spatial domain. It is made up of a set of pixels. We now have to carry out a quantization so that each analog amplitude of the image signal is represented by a discrete value. We choose a set of predefined values called reconstruction levels, spread over the dynamic range $[x_{min}, x_{max}]$ of the signal x to quantize.
On the figure above, these reconstruction levels are defined by the sequence $(r_i)_{i\in[0,K-1]}$ made up of K values. These levels are associated with decision thresholds defined by the sequence $(t_i)_{i\in[0,K]}$ with $t_0=x_{min}$ et $t_K=x_{max}$. Let's take for example the case of a continuous magnitude signal « x » as input into the quantizer; the quantized signal « y » output will be defined by the condition:

$$\forall i \in [0..K-1], \text{ for } t_i \le x < t_{i+1}, \text{ then } y = r_i \text{ and } t_0 = x_{min} ; t_K = x_{max}$$

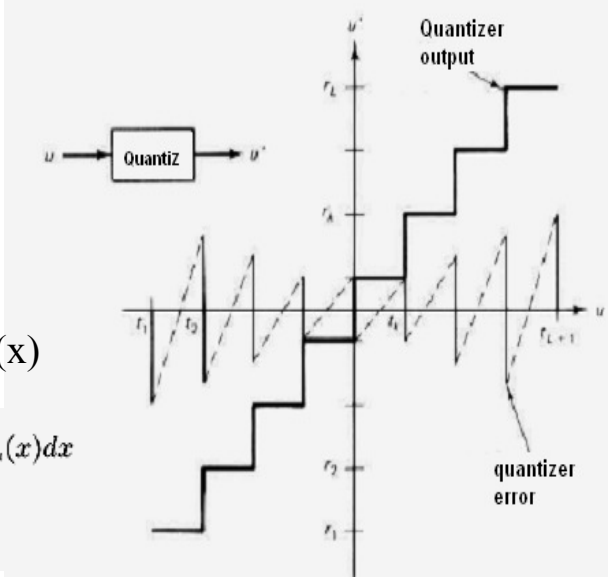
There are several ways of quantizing. The way of choosing the reconstruction levels and the decision thresholds is not necessarily a uniformly distributed type on $[x_{min}, x_{max}]$, and strongly depends on the characteristics of the image signal being quantized.

Example of a quantization law

- Multiple-to-one mapping
- Want to minimize error

- **Mean-square-error (ε)**
  - Let an input « u » with the probability density function $p_u(x)$

$$\begin{aligned}\mathcal{E} &= E[(u-u')^2] = \int_{t_1}^{t_{L+1}} (x-u'(x))^2 p_u(x)dx \\ &= \sum_{i=1}^{L} \int_{t_i}^{t_{i+1}} (x-r_i)^2 p_u(x)dx\end{aligned}$$



The chart above gives an example of a quantization law. The quantizer input corresponds to « u », and the output to the signal « u' ». The decision thresholds are given by the sequence $(t_i)_i$ and the reconstruction levels are given by the sequence $(r_i)_i$. Quantization is a process that causes irreversible loss unlike certain forms of sampling that we have looked at previously. However, depending on how you configure the quantization, the human eye may not necessarily notice these errors. We must then try to minimize them. Typically, we seek to minimize the mean square error. By definition, this square error ε^2 is given by the relation:

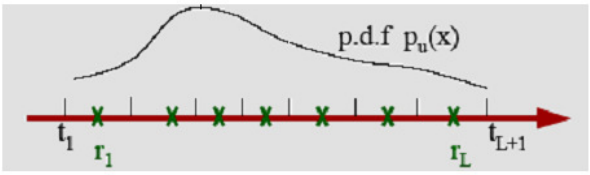
$$\varepsilon^2 = E\left[(u-u')^2\right] = \int_{t_1}^{t_{L+1}} (x-u'(x))^2 . p_u(x).dx$$

Where: E stands for the statistical mean value and $p_u(x)$ stands for the probability density of u. By decomposing the integral, we have finally:

$$\varepsilon^2 = \sum_{i=1}^{L} \int_{t_i}^{t_{i+1}} (x-r_i)^2 . p_u(x).dx$$

Quantization must be chosen so as to minimize the mean square error but we will see that other criteria linked to the image characteristics also are interesting.

Which kind of quantization law to use? 0



- Allocate more values in more probable ranges
- Minimize error in a probability sense
  - **MMSE (minimum mean square error):**

$$\begin{aligned}\mathcal{E} &= E[(u-u')^2] = \int_{t_1}^{t_{L+1}} (x-u'(x))^2 p_u(x)dx \\ &= \sum_{i=1}^{L} \int_{t_i}^{t_{i+1}} (x-r_i)^2 p_u(x)dx\end{aligned}$$

    - assign high penalty to large error
    - square error gives convenience in math.:differential, etc.
  - An optimization problem
    - **What $\{t_k\}$ and $\{r_k\}$ to use?**
    - **Last time: necessary conditions by setting partial differential to zero**

The question asked here is to know what criteria must we take into account in order to correctly choose a quantization law. The figure above represents a quantizer Q with an input signal « u », and an output signal « u' ». The decision thresholds are given by the sequence $(t_i)_i$ and the reconstruction levels are given by the sequence $(r_i)_i$. the probability density function $p_u(x)$ is also represented. In this example, the quantizer is no longer linear: the thresholds and reconstruction levels are not regularly placed. Actually, by looking carefully at the probability density of « u », we can see that the function slope is gentler (or steeper) for certain values of t_i than for others. When the slope is gentle, we can reasonably think that you need few reconstruction levels to represent the range of values described. Inversely, when the curve's slope becomes steeper, numerous values are affected in a short time and so more levels of reconstruction are needed. We have also seen that to choose a quantization law, we also need to minimize the mean square error whose expression directly depends on the parameters (t_i) and (r_i). Optimizing quantization is achieved through the choice of decision thresholds and reconstruction levels. These parameters can be directly determined by minimizing the mean square error (calculation of partial derivatives etc.).

Optimal Quantization (1)

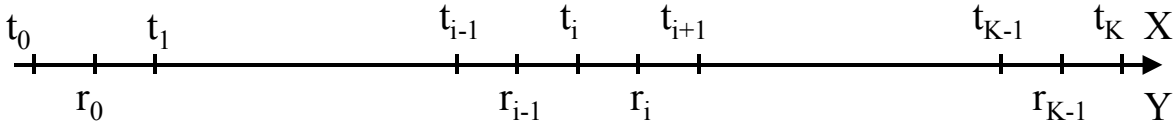
- Criteria of optimal quantization
  - Mean Square Error ($y = Q(x)$)

$$\varepsilon^2 = E\{(X-Y)^2\} = \int_{-\infty}^{+\infty} (x-y)^2 \, p_x(x)\, dx$$

$$\varepsilon^2 = \sum_i \int_{t_i}^{t_{i+1}} (x-y_i)^2 \, p_x(x)\, dx$$

minimization of ε^2:

$$\begin{cases} t_{i+1} = (r_i + r_{i+1}) / 2 \\ r_i = E\{X / x \in [t_i, t_{i+1}]\} \end{cases}$$



- **Special case**: uniform probability density function: $p_x(x)$ is constant

then $t_{i+1} = (r_i + r_{i+1}) / 2$ and $r_i = (t_i + t_{i+1}) / 2$

The optimal quantization criteria is linked to the minimization of the mean square error. The quantizer input corresponds to the signal X, and the output to signal Y. The reconstruction levels are given here by (r_i), and the decision thresholds by (t_i). We are going to try to determine the relations ensuing from this optimization and minimalization criterion: To minimize ε^2, we calculate its partial derivatives:

- In relation to t_i : $\frac{\partial \varepsilon^2}{\partial t_i} = p_X(t_i).(t_i - r_{i-1})^2 - p_X(t_i).(t_i - r_i)^2$
- In relation to r_i : $\frac{\partial \varepsilon^2}{\partial r_i} = -2.\int_{t_i}^{t_{i+1}} p_X(x).(x - r_i).dx$

By cancelling the first partial derivative, we obtain the relation: $t_{i+1} = \frac{r_i + r_{i+1}}{2}$

By cancelling the second partial derivative, we obtain:

$$\int_{t_i}^{t_{i+1}} p_X(x).(x - r_i).dx = 0 \Leftrightarrow \int_{t_i}^{t_{i+1}} p_X(x).x.dx = \int_{t_i}^{t_{i+1}} p_X(x).r_i.dx$$

$$\Leftrightarrow E\{X; x \in [t_i, t_{i+1}[\} = r_i . \int_{t_i}^{t_{i+1}} p_X(x).dx = r_i$$

This means that minimizing the mean square error (optimal quantization) implies that the decision thresholds d_i and the reconstruction levels q_i be given by the relations: $\begin{cases} d_{i+1} = \frac{q_i + q_{i+1}}{2} \\ q_i = E\{X; x \in [d_i, d_{i+1}[\} \end{cases}$.

These are the two relations obtained by Max. These show that the optimal quantizer for a signal using a non-uniform probability density function is not a linear quantizer.
In the particular case of an input signal whose probability density is constant over [x_{min}, x_{max}], we have : $p_X(x) = \frac{1}{(x_{max} - x_{min})}$, and we obtain the relations:

$$\begin{cases} r_i = \frac{t_i + t_{i+1}}{2} \\ t_{i+1} = \frac{r_i + r_{i+1}}{2} \end{cases}$$

The optimal quantizer is linear in this case: $\Delta t_i = \frac{x_{max} - x_{min}}{K}$.

For a signal normalized over [0, 1], the minimized square error is worth: $\varepsilon^2 = \frac{1}{12.K^2}$.

We have seen that, in order to optimize the quantizer, we must minimize the mean square error of quantization with this criterion. Naturally, other criteria may be used, in particular those concerning the visual appearance.

Optimal Quantization (2)

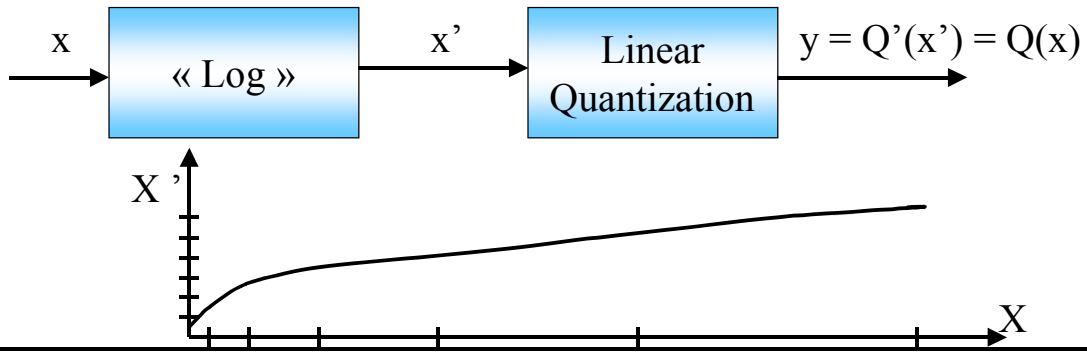
- **Maximum entropy quantization** : H(Y) maximum

$$\Rightarrow \quad p_i = \int_{t_i}^{t_{i+1}} p_x(x)\,dx = \text{constant} \quad \forall i = 1, \ldots\ldots, K$$

- **Psycho-visual quantization of images**

  - Perception of small stimuli on a background: $\Delta L / L = cst$

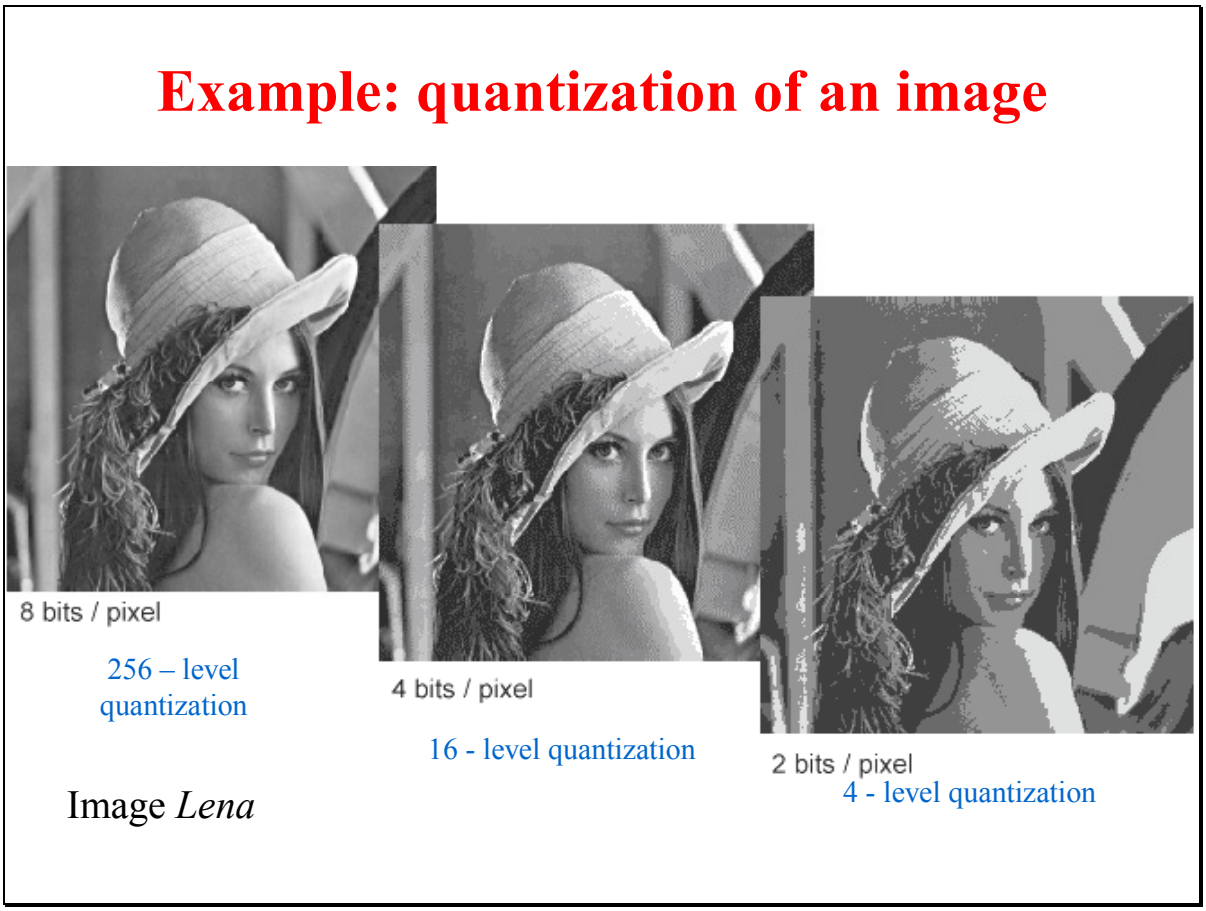
  - $Cst \cong 0.01$ in luminance TV range (given by Weber law)



The characteristics of the human system of vision can be used to design the quantization law. For a usual stimulus of a given form and fixed ΔL amplitude, the visibility threshold ΔL / L is more or less constant (L being the luminance). In the TV luminance range, it varies slightly when L goes from black to white. For a quantization at the visibility threshold, we would have r_{i+1}-r_i=ΔL. It becomes possible to use a uniform quantization on a compressed signal by a non-linearity « f » (see diagram above). You would of course need to carry out the reverse operation coming out of the quantizer with an expansion function.

This non-linear compression function is defined by $f(x) = \int_{L_0}^{x} \Delta L(x).dx \approx \lambda.Log(x / L_0)$. For television, compression « f » is already used: it is the gamma correction given by $f(x) = x^{\frac{1}{\gamma}}$ that compensates for the non-linear nature of CRT-type TV screens. We then apply a linear quantization over 256 leveks. In color television, we usually quantize the luminance and chromatic signals separately. From a perceptual point of view, the YCrCb space is more uniform than the RGB space.

We can see by looking at television-type applications that despite the interesting properties of the optimal quantizer, it can sometimes be a better idea to use a non-linear compression function and then to use a linear quantization (even if the signal's probability density function is not uniform). The "compression function – quantization – expansion function" series would be chosen in order to approximate the optimal quantization law.



This illustration shows examples of a quantization carried out on the image Lena:

- For the image on the left: quantization is followed by a natural binary coding with 8 bits per pixel. There are $2^8 = 256$ reconstruction levels to represent the magnitude of each pixel. It is the typical case of a monochrome image (only in gray scales).
- For the middle image: quantization is carried out with a 4 bits per pixel coding, giving $2^4 = 16$ reconstruction levels. Contours are well rendered but textures are imprecise in some cases. These are areas in the signal with a weak spatial variation, which suffer more visually due to the appearance of false contours (loss on the face and the shoulder).
- For the image on the right: quantization is carried out with a 2 bits per pixel coding, so we have $2^2 = 4$ reconstruction levels. The deterioration seen on the previous image is even more flagrant here.

We have now seen the various steps of image digitalization: double discretization by spatial sampling and then by quantization. The images are now digital and are ready to be processed with appropriate techniques, according to the required application.f