Master: INFORMATIQUE
Parcours: VICO Visual Computing

UE: Multimedia Communication

Image and video coding

vincent.ricordel@univ-nantes.fr

Basis on image/video coding

Outlines:

- Source / channel coding
- Lossless / lossy coding
- DPCM principle
- Transform principle
- Quantization principle
- Errors assessment

Introduction

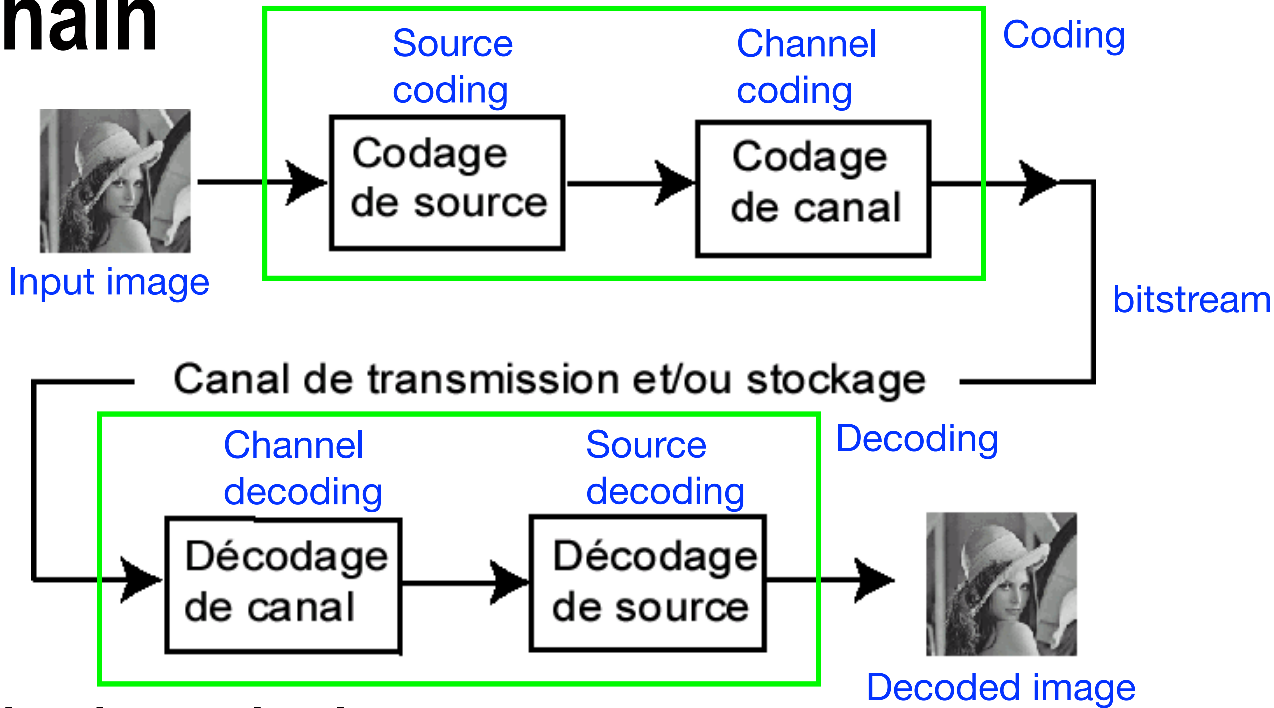
- Context : digital image coding

- Goal : reduce the binary amount of data that represents the image information
➔ Source coding

$$Coding\ ratio. = \frac{Info.\ Qty.\ original\ .\ ima.\ [bit]}{Info.\ Qty.\ compressed\ .\ ima.\ [bit]}$$

- Application : storage or transmission

Coding chain

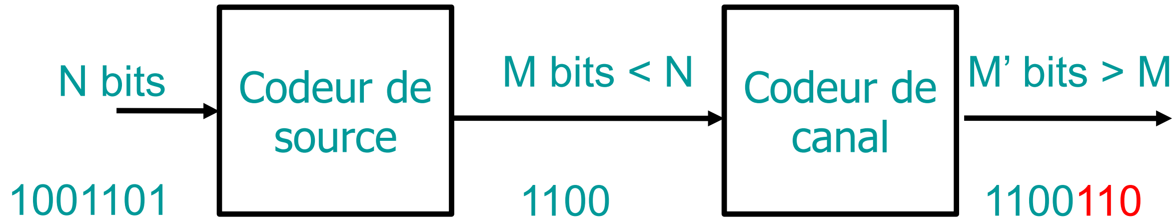


- Transmission chain:
  - Source coding / Channel coding
  - Dual operations for the decoding

- Channel :
  - Physical link between transmitter and receptor (cable, fiber, air...)
    => Noise, interferences...
- Channel coding:
  - Goal :
    - Low error ratio : protection of the message
      - Packet loss, and errors (bit or packet)
      - => error correction (and error detection)
    - Adaptation of the signal to channel (modulation ...)

  => through an error free transmission

  « cost » = the message size increases



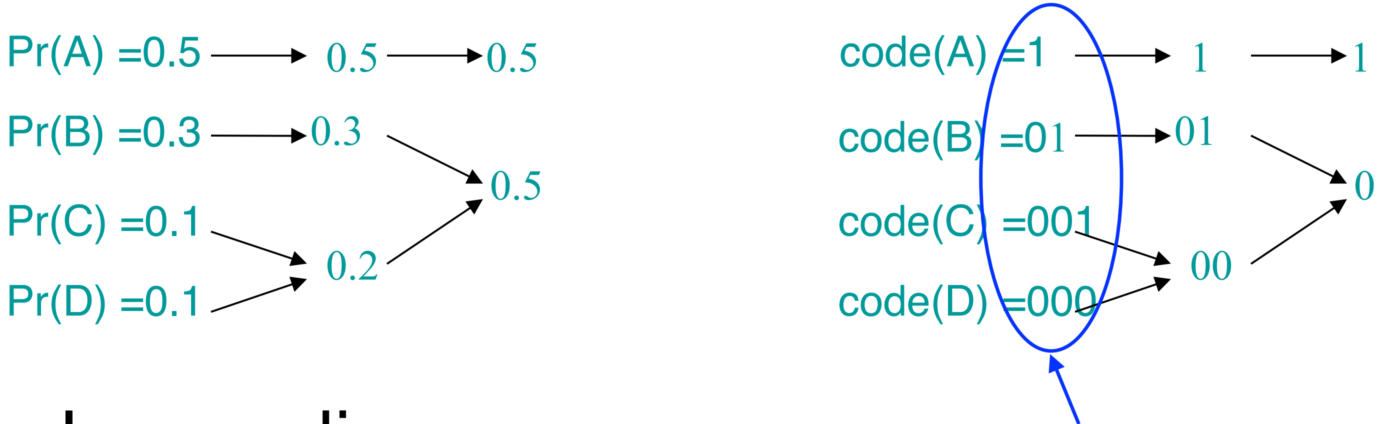
- Channel coding => increasing of the redundancies
  - Ex :
    - basic : duplication
      - n times the same bit : decision = majority
      - If n = 8, for 0 :
        - 0000 0000 sent
        - 0110 0001 received => 0 chosen
    - Bit of parity
    - CRC
    - Hamming, Reed Salomon, ...

- Source coding, 2 types :
  - Lossless coding (entropic coding)
    Ex. : Huffman, Lempel-Ziv, Arithmetic …
    ➔ Low compression ratio (1.5 à 2)
  - Lossy coding :
    No redundancies
    The redundant information :
    - Predictable
    - No perceived by the human visual system (HVS)

    ➔ High compression ratio (>10)

Lossless Coding

- Ex 1 : RLC « Run Lenght Coding »
  - Source:
    M = “ 00000000100000000011000000000000 ”
  - Transmitted info.:
    RLC(M) = “ 01000,1,01001,1,1,01100 ”
    - couple (“ nb_0 ” , 1)
    - the ‘,’ are virtual
    - 0 = separators
    - Ex. a serie of 75 ‘0’ is coded by 20 bits :
      ‘01111 01111 01111 01111 01111’



These codewords are uniquely decodable

- **Lossless coding**
  - Ex 2 : Huffman
    - The code size depend on the occurring probability of the symbol in the data
      => the decoder has to compute these probabilities
    - Ex. : AABCDABABC => 20 bits
      (2 bits/character * 10)
      - pr(A) = 0.5, pr(B) =0.3, pr(C) =0.1, pr(D)=0.1
      - Design of a tree
      - Finally :
        size = 5*1 + 3*2 + 3*1 + 3*1 = 17 [bits]

- Lossless Coding « with codebook »
  - Ex 3 : GIF " Graphics Interchange Format "
    - Design of a codebook containing words (the repeat) :
      - For each word a code
      - The codebook is transmitted or no
  - Ex 4 : LZW (ZIP)
    - Dynamic design of the codebook
    - ...

Input image =
Gray level image :
8 bits/pixel

GIF : low efficiency with natural images

6.8 bit/pixel
Compression ratio = 8/6.8 = 1.17

- Opposition between
  Source coding / Channel Coding

  - Source coding :
    decreasing of the source redundancies

  - Channel :
    increasing of the redundancies to protect the information

    => dual source & channel coding

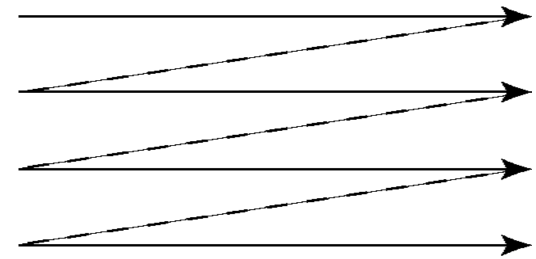
    Take into account when encoding the source of the constraints linked with the channel

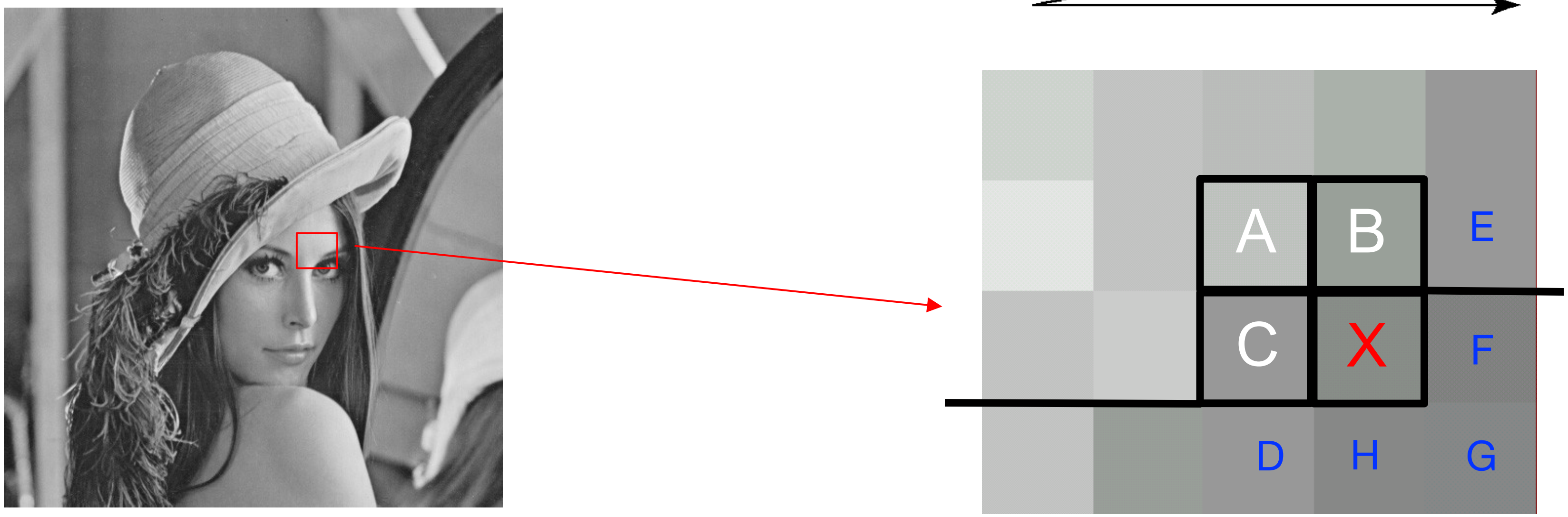
Source coding

- Lossy coding, methods :
  - Prediction / transform / quantization
- Ex. Prediction
  - Image transmission using a « Zigzag scan »
    - Image = « matrix of pixels »
    - Pixel = « picture element »
    - Pixel coded with 1 byte (2^8 =256)
      - 8 bits ➔ 256 gray levels
      - « 0 » ➔ black
      - « 255 » ➔ white

Gray level image :
8 bits/pixel

- « Zigzag scan »





- Gray level of a pixel depends on the levels of its neighbors
- Base : predict X by using A, B or C (DPCM coding basis)

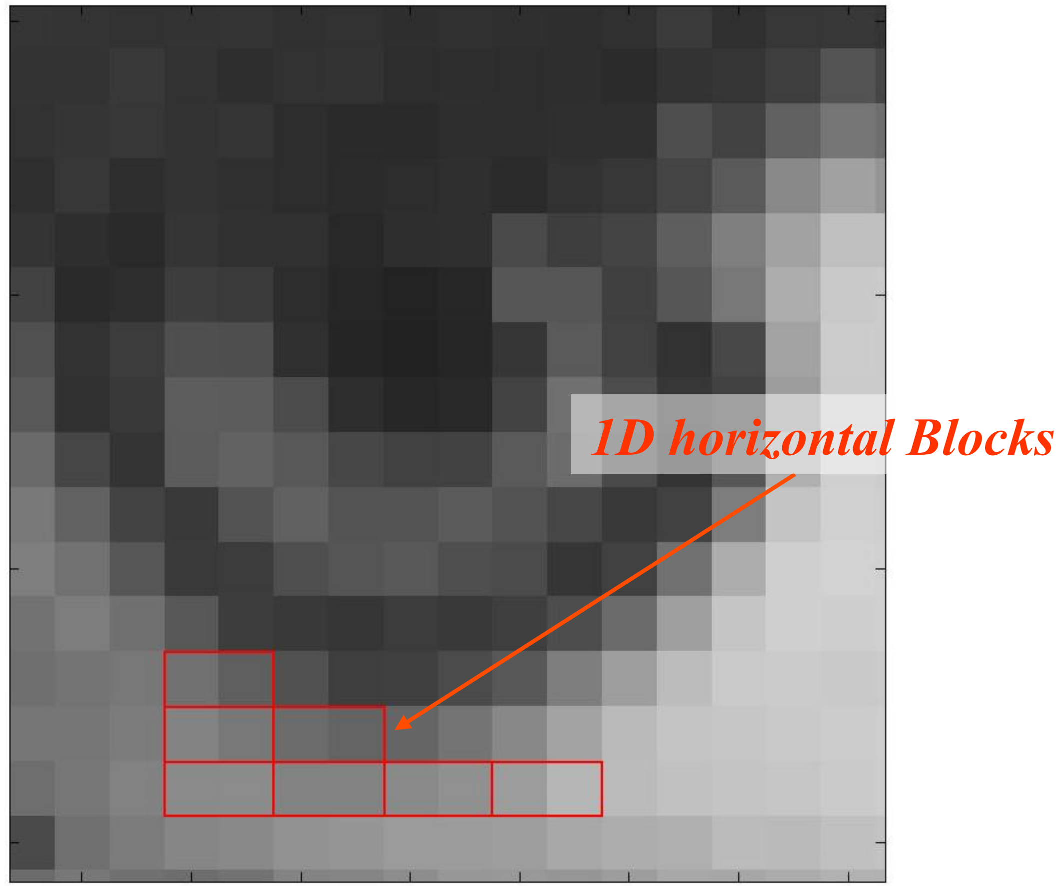
➔ <u>analysis</u> of the image

Predict X by using :
« past » pixels (coded/decoded) that are closed spatially.

To subdivide the image into 1×2 blocks

- ***To subdivide the original image into 1×2 1D horizontal blocks***

Image: *Lena*



Two neighboring pixels have often close luminance values. How are these luminance values correlated?

Correlation between two neighboring pixels

- For each block, we plot the point (x_1, x_2) that represents the luminance value of the left pixel (x_1) with respect to the luminance value of the right pixel (x_2) :

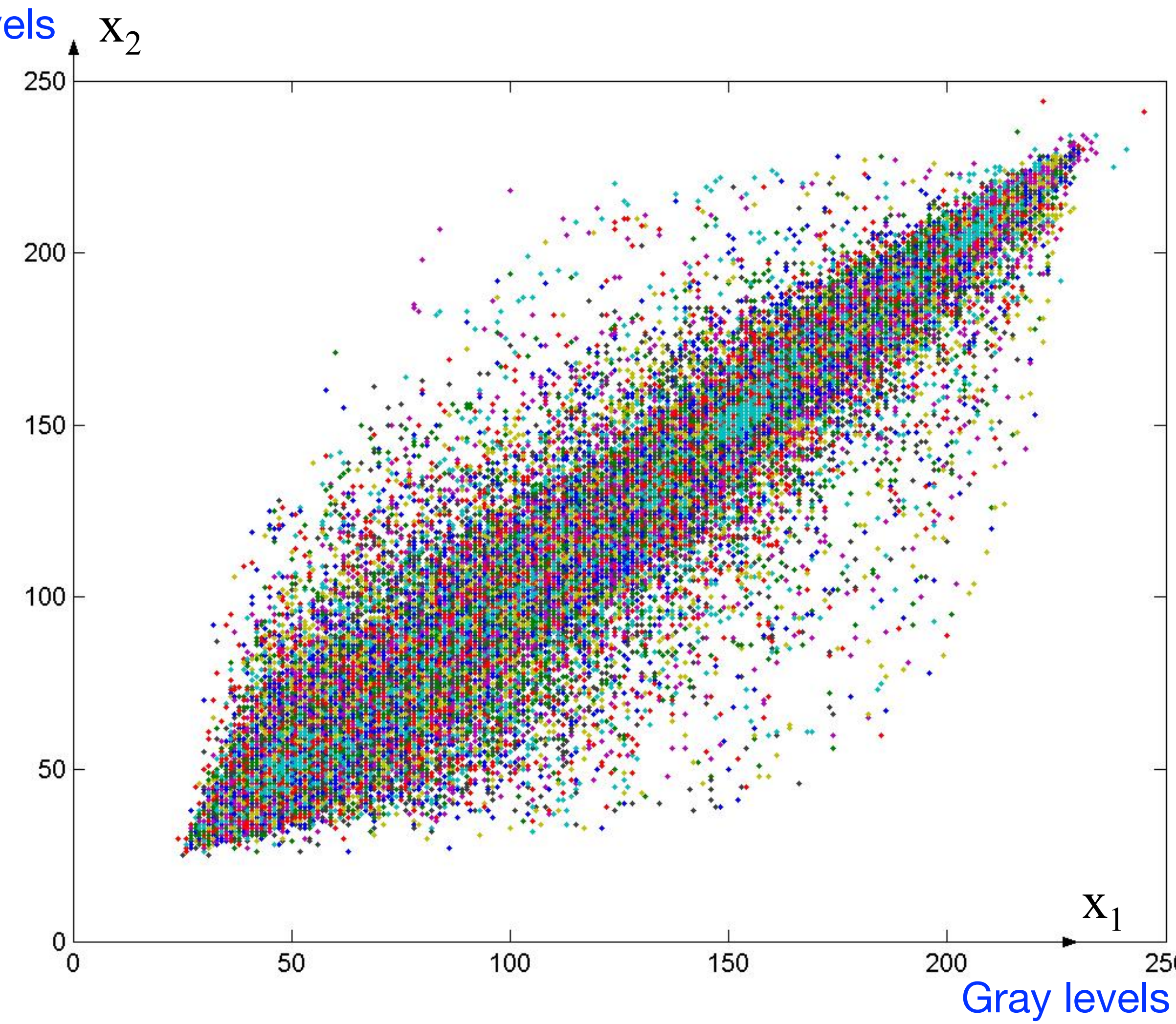
Pixels are concentrated around the line $x_2 = x_1$

↓

The luminance values of two neighboring pixels are close

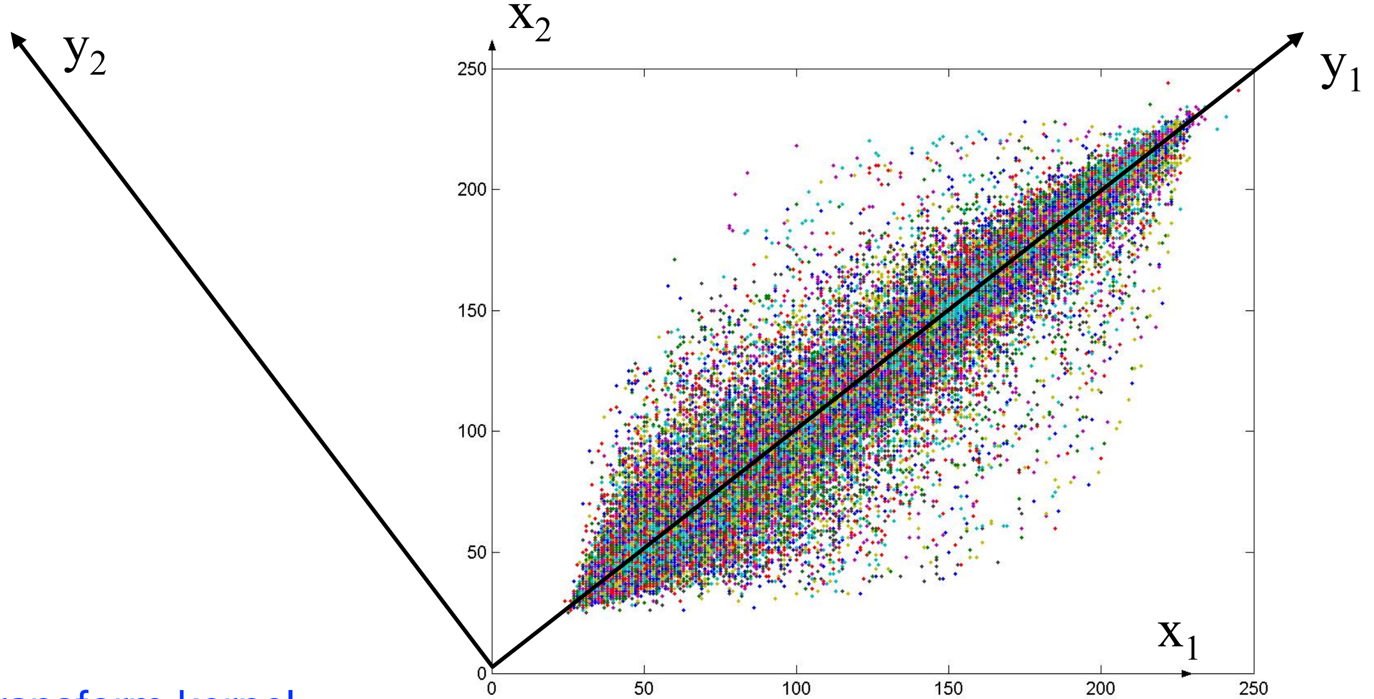
↓

The neighboring pixels are strongly correlated



Decorrelation of two neighboring pixels

❖ To decorrelate the image pixels, we look for an application $\mathcal{T}$ that transforms the luminance value x_1 of a block into a transformed value y_1 that is uncorrelated with the neighboring transformed value y_2:

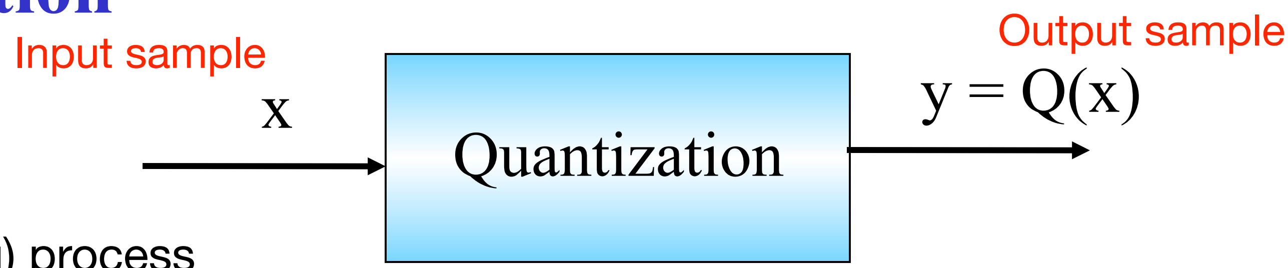


Transform kernel

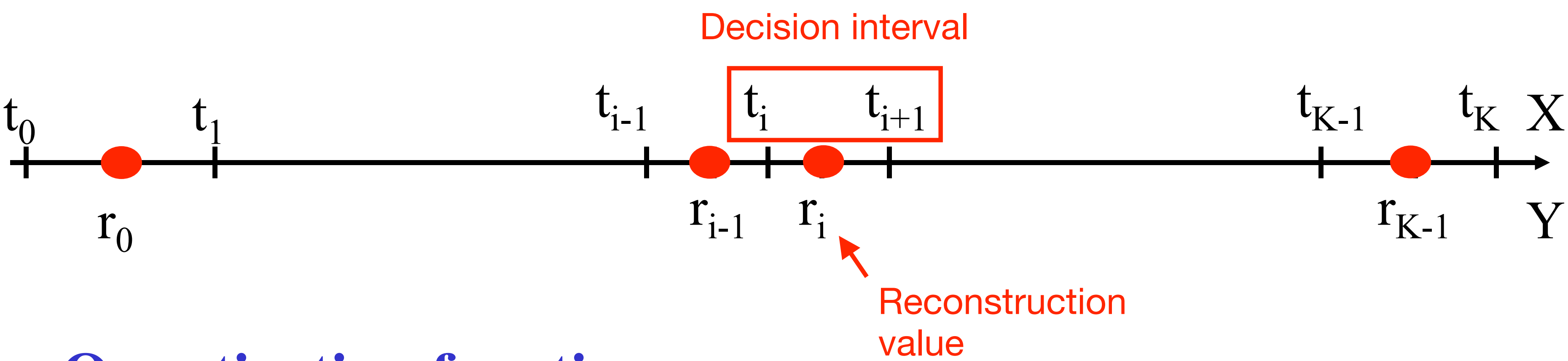
Here: $\mathcal{T} = \begin{bmatrix} \cos(\pi/4) & \sin(\pi/4) \\ -\sin(\pi/4) & \cos(\pi/4) \end{bmatrix}$ and $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathcal{T} \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ Transform

Quantization

- **Scalar quantization**



A « simplification » (rounding) process



- **Quantization function**

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

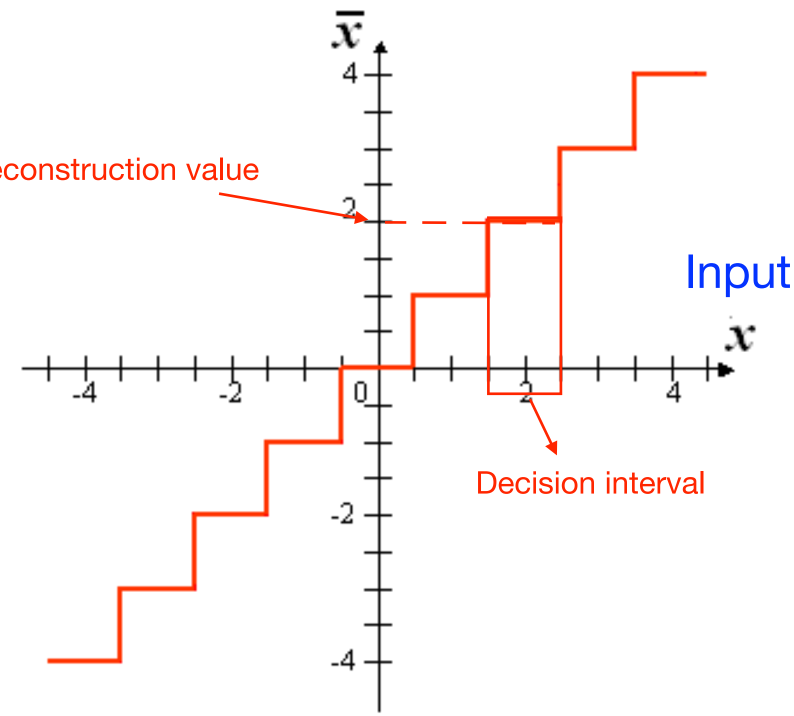
and $\quad t_0 = x_{min} \; ; \; t_K = x_{max}$

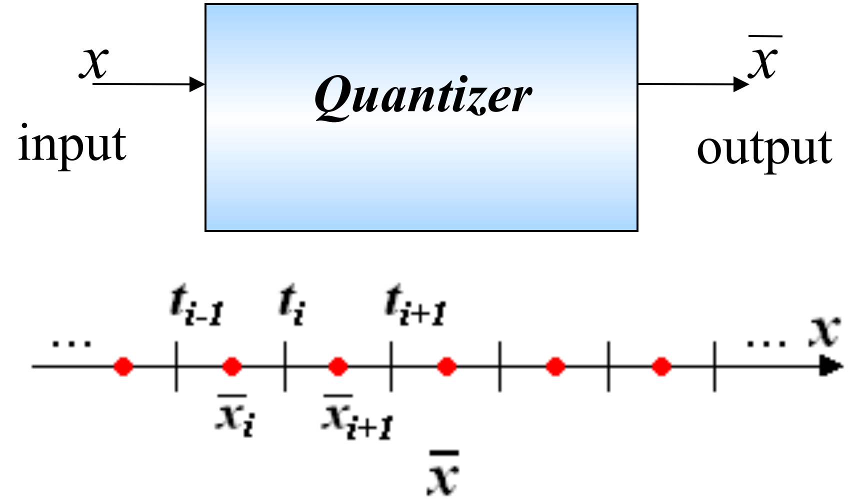
Scalar Image Quantization

- Usual because of its simplicity

$$Q: \quad \Re \rightarrow C = \{\overline{x}_1, \ldots, \overline{x}_i, \overline{x}_{i+1}, \ldots, \overline{x}_L\}$$

$$x \rightarrow \overline{x} = Q(x)$$



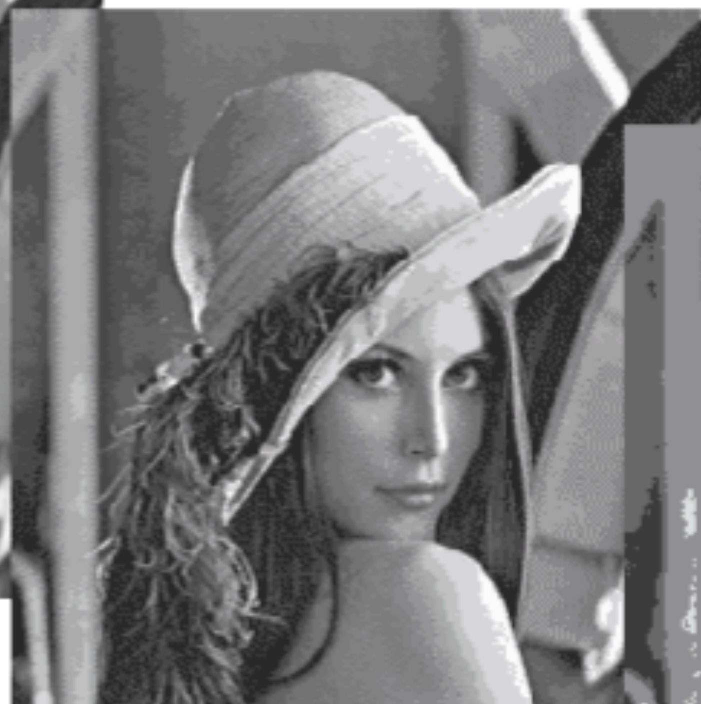


Example of a 9-level uniform quantizer

Example: quantization of an image

8 bits / pixel

256 – level quantization



4 bits / pixel

16 - level quantization

2 bits / pixel

4 - level quantization

Image *Lena*

Appréciation des erreurs

- Cumul des distorsions, ex. 1 :
  - EQM « Erreur Quadratique Moyenne »
    - MSE « Mean Square Error »
    - Erreur : $e(i,j) = I_{réf}(i,j) - I_{dég}(i,j)$
      Error
      (Iréf: Original image; Idég: Distorted image)
    - Cumul des erreurs : $\sum_{i=1:N} \sum_{j=1:M} e^2(i,j)$
      Cumulative errors
      - Normalisation : $(1/(M.N)) \times \sum_{i=1:N} \sum_{j=1:M} e^2(i,j)$

        Image size: MxN pixels

Appréciation des erreurs

- Cumul des distorsions, ex. 2 :
  - PSNR « Peak Signal-to-Noise Ratio »

MSE

- $PSNR_{[dB]} = 10 \cdot \log_{10} \left(255^2 / \left(\left(1/(M.N) \times \sum_{i=1:N} \sum_{j=1:M} e^2(i,j) \right) \right) \right)$

- *PSNR* > 35 *dB* => très bonne qualité Good quality
- *PSNR* < 20 *dB* => qualité médiocre Poor quality

« In fact, poorly correlated with the human judgement »

Appréciation des erreurs

Pseudo-image of the errors

- Image des erreurs :

Uniform quantizer

  - Quantification uniforme puis LUT telle que :
    - *e(i,j) = 0* : NdG = 127
    - *e(i,j) max* (typ. + 255) : NdG = 255
    - *e(i,j) min* (typ. - 255) : NdG = 0

Display of the errors as gray levels