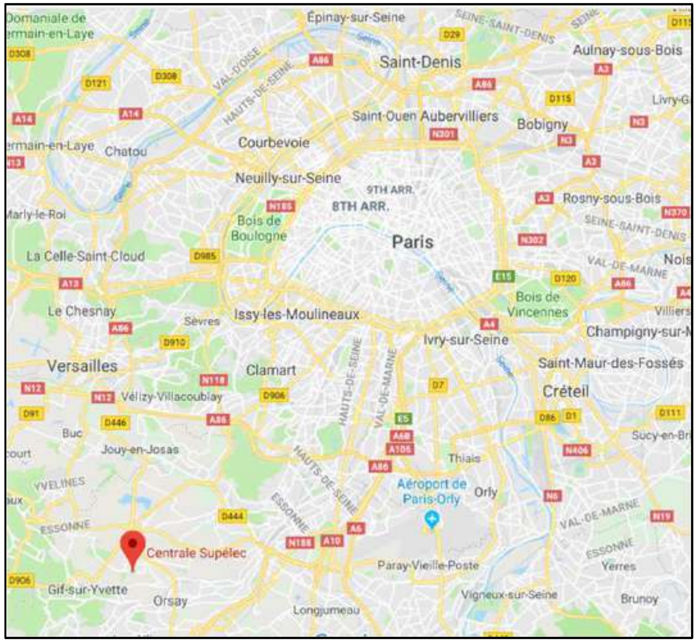
INTRODUCTION TO CODING THEORY
APPLICATION TO IMAGE CODING

Frederic Dufaux

frederic.dufaux@l2s.centralesupelec.fr

Signals and Systems Laboratory – L2S

- **Signals and Systems Laboratory (L2S)**
  - Univ. Paris-Saclay, CNRS, CentraleSupelec
  - 100 researchers and 150 PhD students / postdocs
  - **Fundamental** and **applied** aspects of **mathematics** in signal processing, information theory, communications, and control feedback theory



- **Three departments**
  - Signals and Statistics
  - Systems and Control
  - Telecoms and Networks

My Research Activities

- **Common theme**
  - Hyper-realistic and immersive visual communications for enhanced Quality of Experience
- **Research interest**
  - image and video coding, video transmission over wireless network
  - volumetric video, point cloud, light field, digital holography
  - high dynamic range imaging
  - visual quality assessment, image aesthetics
  - image and video analysis
- **Links**
  - https://www.l2s.centralesupelec.fr/u/dufaux-frederic/
  - https://scholar.google.com/citations?user=ziqjbTIAAAAJ&hl=en

Context and Trends

- 350 Million photos are uploaded every day
- 100 million hours of video watched every day

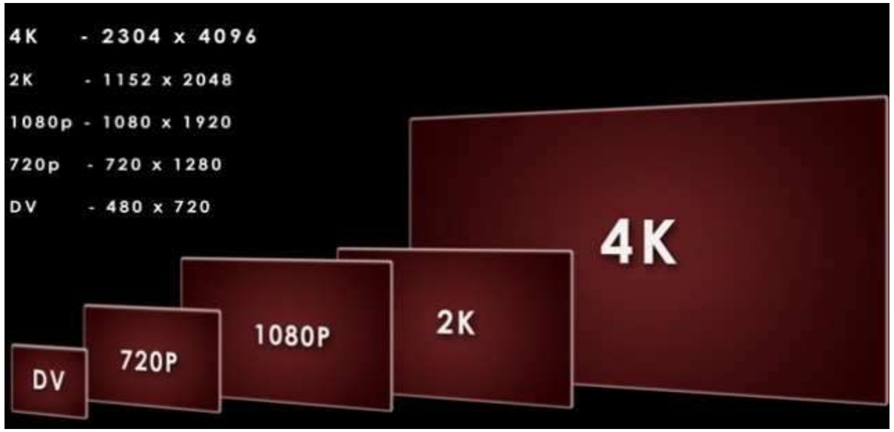
- 500 hours of video uploaded every minute
- 1 billion hours of video watched every day

- Video represented 75% of all Internet traffic in 2017
- Expected to account for 82% in 2022

Images / videos are omnipresent !!

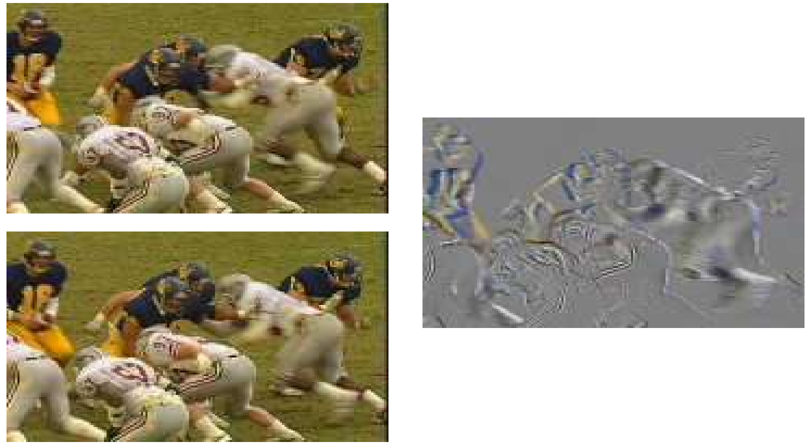
Context and Trends

- **More, more, more…**
- **Higher spatial and temporal resolutions**
  - Ultra High Definition (UHD), 4K, 8K
  - High Frame Rate (HFR)
- **Higher pixel depth**
  - up to 14 bits per component
  - High dynamic range
- **More colors**
  - 4:4:4 color sampling
  - Wide color gamut
- **More views**
  - 3D, multi-view, free viewpoint

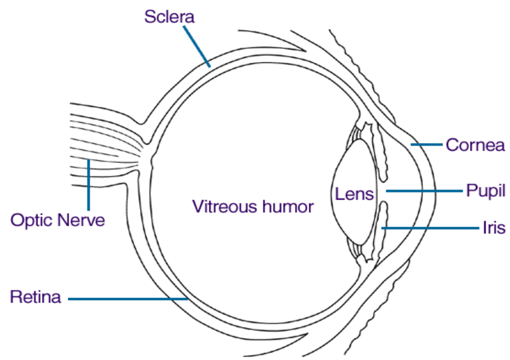


Compression: fundamental concepts

- Exploit the correlation in the data
  - Reduce redundancies
  - Lossless



- Exploit the human visual system
  - Remove imperceptible data
  - Introduce (hardly) noticeable distortions



Outline

- **Preliminaries**
- **Source Models**
  - Memoryless source, Markov source
- **Source Coding**
  - Uniquely decodable, instantaneous, prefix
  - Huffman Code, Run-length Encoding
- **Rate-distortion theory**
- **Transform Coding**
  - Discrete Cosine Transform (DCT)
  - Optimal bit allocation
- **Predictive Coding**
- **JPEG**

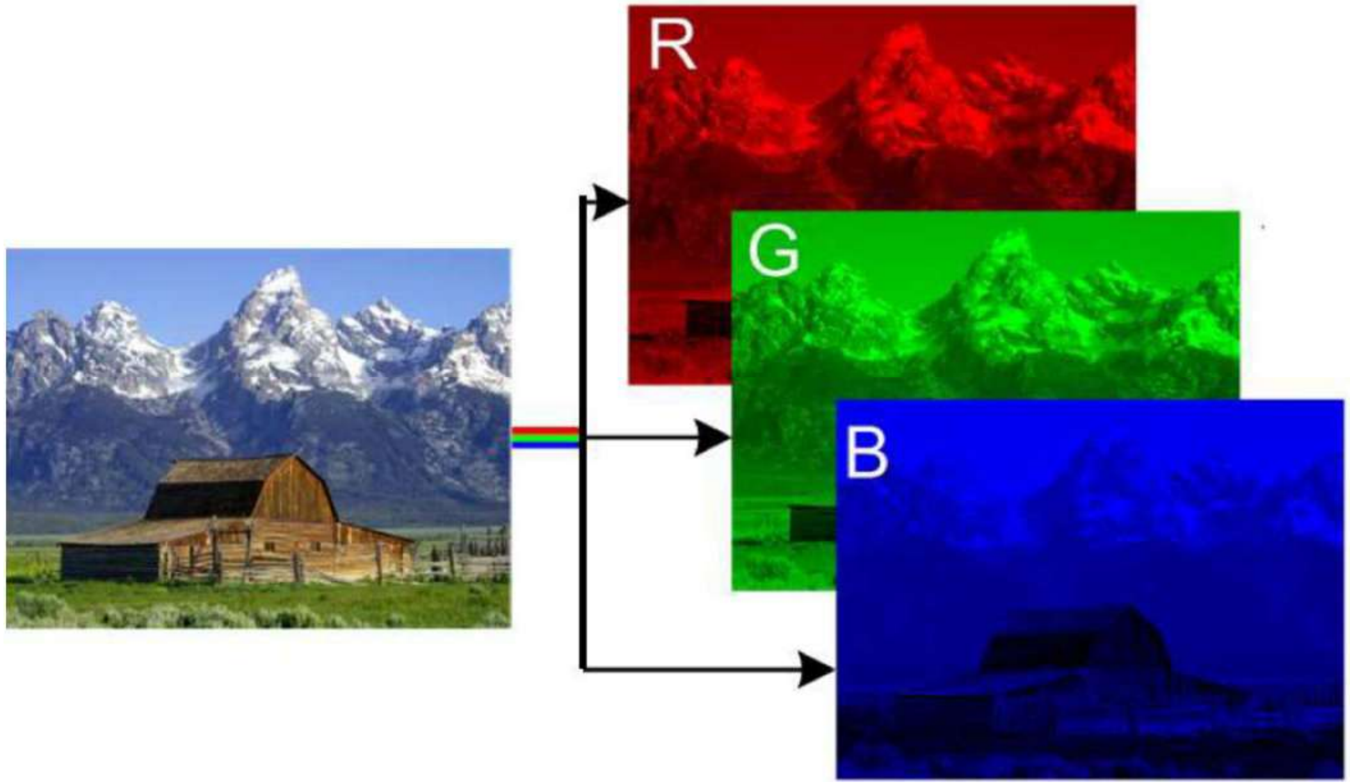
PRELIMINARIES

Preliminaries

- **Consider sources made from discrete values**
  - their elements belong to a finite alphabet
  - images, video, text, audio (music or speech), etc.
- **These sources are generally redundant**
- **Source coding intends to remove this redundancy**

Representation of digital video/images

- **A color may be represented as a mixture of Red (R), Green (G) and Blue (B)**
  - Represent a color image with three separate arrays: R, G, and B
  - Inspired by trichromatic human vision



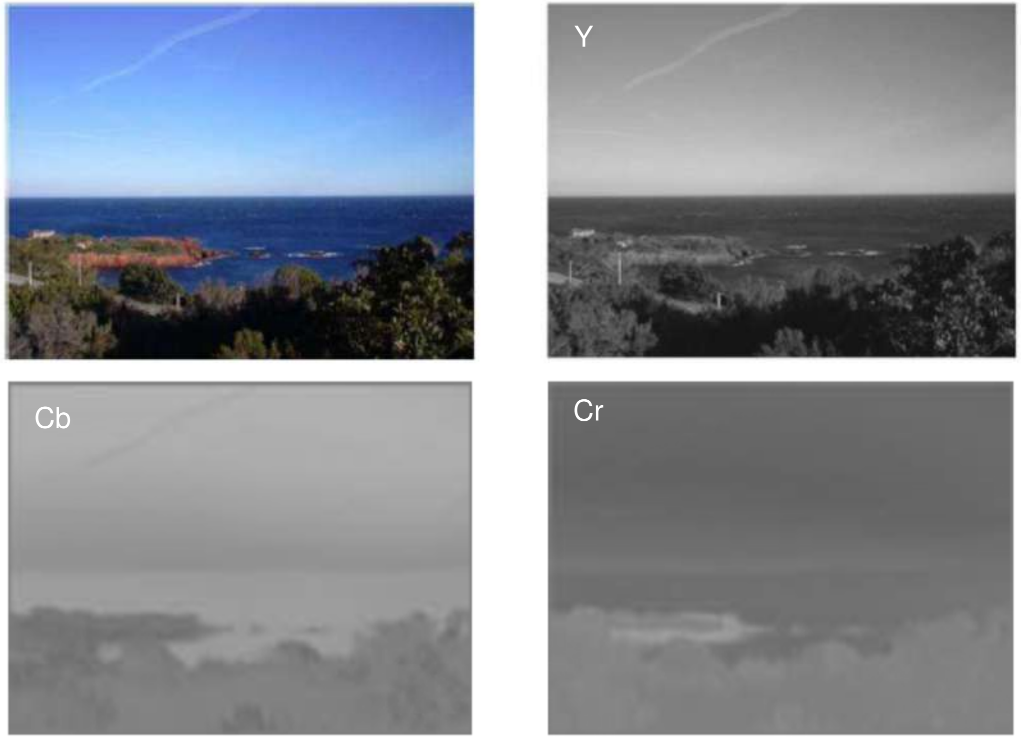
  - Typically, each pixel value is represented with 8 bits: [0…255]

Representation of digital video/images

- **Luminance-chrominance color space**
  - YCbCr
  - Decorrelate the data

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 1.4021 \\ 1.0 & -0.3441 & -0.7142 \\ 1.0 & 1.7718 & 0.0 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix}$$



  - Chroma subsampling
    - *4:2:2 horizontal subsampling by 2*
    - *4:2:0 horizontal & vertical subsampling by 2*

SOURCE MODELS

Source Models

- **Source**
  - The source signal (image, audio, text) is modeled as the realization of a random sequence:

$$X_1, X_2, X_3, \ldots, X_{n-2}, X_{n-1}, X_n$$

- **Models**
  - Stationary or non-stationary
  - Dependence between the successive elements :
    - *memoryless discrete source*
    - *discrete Markov source of order 1*
    - *discrete Markov source of order m*

Memoryless Discrete Source

- **Memoryless discrete source**
  - The symbols generated by X_l, l=1, …, n belong to a set (alphabet) $\mathcal{X}=\{a_1,\ldots,a_J\}$
  - For each sample, a realization of X_l is drawn
    → probability $\Pr(X_l = a_i) = p_i$
    → independent with respect to previous symbols
    → stationarity (implicit) : p_i does not depend on time.

We have:

$$\Pr(X_n = a_{i_n} \,|\, X_{n-1} = a_{i_{n-1}}, \ldots, X_1 = a_{i_1}) = \Pr(X_n = a_{i_n})$$

$$\Pr(X_1 = a_{i_1}, \ldots, X_n = a_{i_n}) = \prod_{l=1}^{n} \Pr(X_l = a_{i_l})$$

Memoryless Discrete Source

– Given a memoryless discrete source with probability distribution

$$\mathbf{p} = (p_1, \dots, p_J)$$

– The information associated with each symbol is

$$I(p_i) = \log_2\left(\frac{1}{p_i}\right) = -\log_2(p_i) \quad \text{(in bit/symbol)}$$

– The *entropy* of a source *X* is the average amount of information carried by a symbol

$$H(X) = -\sum_{i=1}^{J} p_i \log_2(p_i) \qquad \text{(in bit/symbol)}$$

Memoryless Discrete Source

– For an alphabet with J symbols

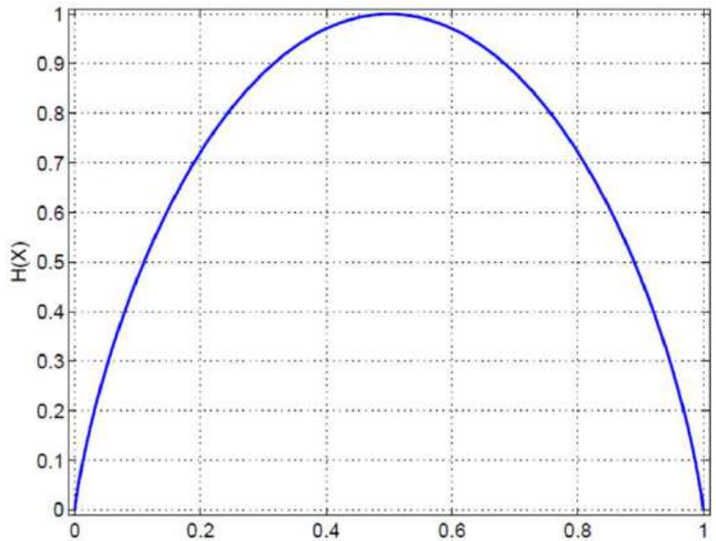
$$H(X) \leq \log_2(J)$$

with equality when all symbols have the same probability

$$p = P\{X = 0\}$$

$$q = P\{X = 1\} = 1 - p$$

$$H(X) = -p\log(p) - (1-p)\log(1-p)$$



SOURCE CODING

Source Coding

Consider source X with values in $\mathcal{X} = \{A,B,C,D\}$

Different codes can be assigned to each symbol

Natural binary code, fixed length:	A: 00	B: 01	C: 10	D: 11
Variable length code:	A: 0	B: 10	C: 110	D: 111

Questions :

– Why choose a given code rather than another one ?
– What properties should a code have ?
– Is there an optimal code ?

Source Coding

- **Definition**

  A source code C for a source X is a mapping function from $\mathcal{X}$ to $\mathcal{D}^*$, the set of codewords from a D–ary alphabet (D values).

  Notations :

  $C(x)$: code associated with $x \in \mathcal{X}$

  $l(x)$: length of $C(x)$

  In the binary case:

  $\mathcal{D}^* = \{0, 1, 00, 01, 000, 001, 010, \ldots\}$

  with alphabet

  $\mathcal{D} = \{0, 1\}$

- **Definition**

  Average length of a memoryless source

$$L(C) = \sum_{x \in \mathcal{X}} \Pr(X = x)\, l(x)$$

Source Coding

- **Example**

  Consider source X with the associated code

$\Pr(X = 1) = 1/2$	codeword $C(1) = 0$
$\Pr(X = 2) = 1/4$	codeword $C(2) = 10$
$\Pr(X = 3) = 1/8$	codeword $C(3) = 110$
$\Pr(X = 4) = 1/8$	codeword $C(4) = 111$

  $H(X) = 1.75$ bits and $L(C) = 1.75$ bits $\rightarrow L(C) = H(X)$!

  Any sequence of bits can be uniquely decoded, e.g.

  $0110111100110 \quad \rightarrow \quad 134213$

  Consider source Y with the associated code

$\Pr(Y = 1) = 1/3$	codeword $C(1) = 0$
$\Pr(Y = 2) = 1/3$	codeword $C(2) = 10$
$\Pr(Y = 3) = 1/3$	codeword $C(3) = 11$

  $H(Y) = 1.58$ bits and $L(C) = 1.66$ bits $\rightarrow L(C) > H(Y)$!

- **Definition**

  A code is *non singular* if each element of source *X* is associated with a different element of $\mathcal{D}^*$,

$$x_1 \neq x_2 \rightarrow C(x_1) \neq C(x_2)$$

  This is obviously a useful property, but not sufficient

- **Definition**

  The extension *C** of a code *C* is the function associating a sequence of source elements (belonging to $\mathcal{X}$) to a sequence of codewords (belonging to $\mathcal{D}^*$) :

$$C(x_1, x_2, \ldots x_n) = C(x_1)\, C(x_2) \ldots C(x_n)$$

  with $C(x_1)\, C(x_2) \ldots C(x_n)$ denoting the concatenation of the codes associated with x_1, $x_2, \ldots, x_n$

- **Definition**

  A code is said to be *uniquely decodable* if its extension is non singular

  In other words, any encoded sequence of codewords has only one possible sequence of symbols producing it. However, one may have to look at the entire coded stream to determine even the first symbol of the source.

- **Definition**

  A code is said to be *instantaneously decodable* if it can be decoded without reference to future codewords. In other words, the symbol x_i can be decoded as soon as we come to the end of the corresponding codeword.

- **Definition**

  A code is denoted as *prefix* if no codeword is the prefix of another codeword

  All prefix codes are instantaneously decodable.

  However, all instantaneously decodable codes are not prefix, but none has shorter average length. Hence, one can concentrate on prefix codes.

- **Example**

X	Singular	Non-singular, but not uniquely decodable	Uniquely decodable, but not instantaneous	Prefix, instantaneous
A	0	0	10	0
B	0	010	00	10
C	0	01	11	110
D	0	10	110	111

- **Kraft inequality**

  A prefix code defined on an alphabet of size D with associated lengths l_1, l_2,…, l_J has the following property

$$\sum_{i=1}^{J} D^{-l_i} \leq 1$$

In the binary case, D=2, the Kraft inequality becomes

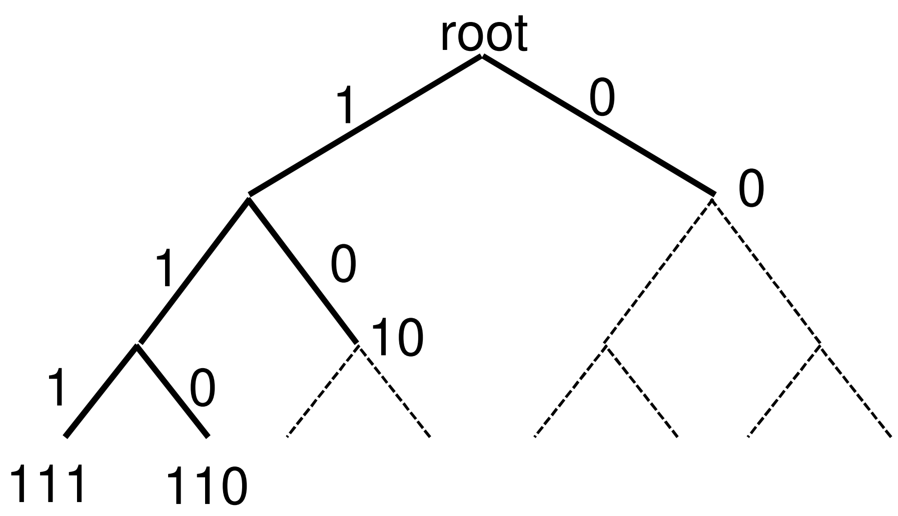
$$\sum_{i=1}^{J} 2^{-l_i} \leq 1$$

Conversely, given codewords of length l_1, l_2,…, l_J satisfying Kraft inequality, one can build a prefix code with these word lengths.

- **Kraft inequality - demonstration**

  The demonstration requires organizing the codewords of the prefix code as a tree

  Example (binary case):



Prefix condition implies that each codeword eliminates its descendants as possible codewords

- **Kraft inequality - demonstration**

  Let l_{max} be the length of the longest codeword
  The total number of nodes at level l_{max} is $2^{l_{\max}}$

  A codeword at level l_i corresponds to $2^{l_{\max}-l_i}$ descendants at level l_{max}

  Summing over all the codewords, we have

$$\sum 2^{l_{\max}-l_i} \leq 2^{l_{\max}}$$

  Hence

$$\sum 2^{-l_i} \leq 1$$

- **Optimal code**

  We are looking for a prefix code associated to some source X characterized by the probabilities **p** = (p_1,…,p_J) minimizing

$$L = \sum p_i l_i$$

where the $l_1,\ldots, l_J$ characterizing the codewords lengths (relaxing the integer constraint) satisfy (binary case)

$$\sum 2^{-l_i} \leq 1$$

Write the Lagrangian

$$\mathcal{J} = \sum p_i l_i + \lambda\left(\sum 2^{-l_i} - 1\right)$$

- **Optimal code**

  The minimum is reach when

$$\frac{\partial \mathcal{J}}{\partial l_i} = p_i - \lambda \ln 2 \; 2^{-l_i^*} = 0$$

$$\sum p_i - \lambda \ln 2 \sum 2^{-l_i^*} = 0$$

$$\lambda = \frac{1}{\ln 2}$$

$$p_i = 2^{-l_i^*} \;\; \rightarrow \;\; l_i^* = -\log_2 p_i$$

  Therefore, the average length

$$L^* = \sum p_i l_i^* = -\sum p_i \log_2 p_i = H(X)$$

Bounds on the Optimal Codelength

If we re-introduce the constraint that the l_i are integers

- **Theorem**

  The expected length of any instantaneous binary code for a random variable X is greater or equal to the entropy $H(X)$

$$L \geq H(X)$$

with equality iff $2^{-l_i} = p_i$

Bounds on the Optimal Codelength

- **Theorem**

  For any discrete source X with pdf $\boldsymbol{p}$, there exists a prefix binary code such that the average length L of the codewords statisfies

$$L < H(X) + 1$$

HUFFMAN CODE

Huffman Code

- **Consider a discrete source *X***
  - Emitting symbols with values in $\mathcal{X}$
  - pdf **p**
  - Entropy $H(X)$ bits/symbol
- **Algorithm to construct an optimal code for a given source distribution**
  - Variable Length Code (VLC)
  - Prefix code (no codeword is the prefix of another one)
    - *Instantaneously decodable*

Huffman Code

- **There exists an optimal instantaneous code (with minimum expected length) that satisfies the following properties**
  1. If $p_j > p_k$ then $l_j \leq l_k$
  2. The two longest codewords have the same length
  3. The two longest codewords, corresponding to the two least likely symbols, differ only in the last bit (least significant bit)

Huffman Code

- **Consider some source *X* with *J* symbols and we want to build the associated minimal code *C*.**
  1. Take the two least probable symbols a_j and a_k and combine them in a fake symbol a_{jk} of probability $p_j + p_k$

     → *new source with J - 1 symbols*
  2. Find the minimal code C_{J-1} for this new source
  3. The codewords associated to a_j and a_k must differ only by their LSB, hence their values are obtained by adding 0 or 1 to the codeword associated to the fake symbol a_{jk}

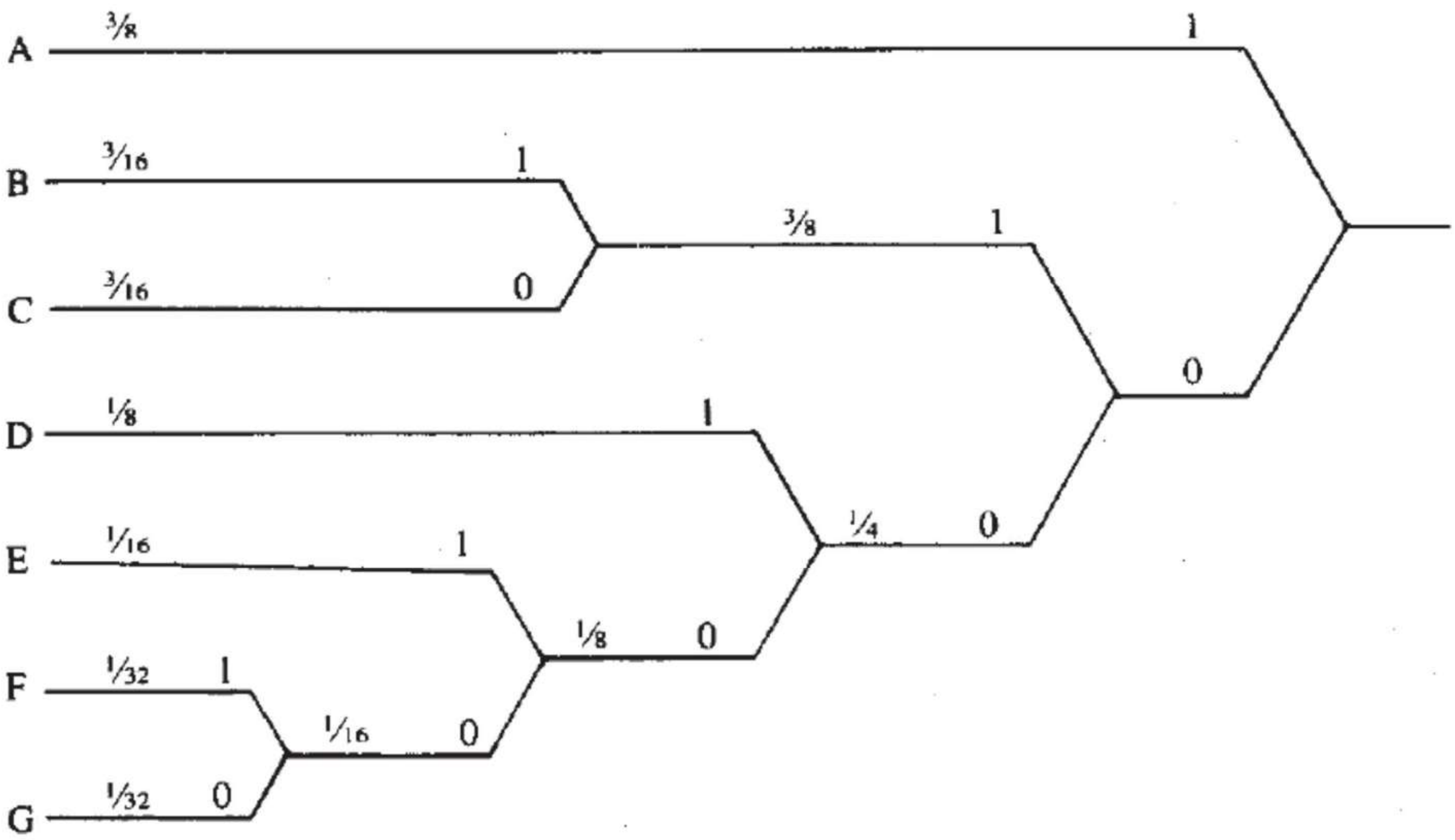
     → *C is thus very simply obtained from C_{J-1}.*

- **If C_{J-1} is minimal for the source with *J* - 1 symbols, then *C* is minimal for the source with *J* symbols.**

- **Example**
  - Consider a source emitting symbols belonging to {A,B,C,D,E,F,G}
  - with pdf

Symbol	Probability
A	$3/8$
B	$3/16$
C	$3/16$
D	$1/8$
E	$1/16$
F	$1/32$
G	$1/32$

Huffman Code



Huffman Code

Symbol	Probability	codeword	codeword length
A	$3/8$	1	1
B	$3/16$	011	3
C	$3/16$	010	3
D	$1/8$	001	3
E	$1/16$	0001	4
F	$1/32$	00001	5
G	$1/32$	00000	5

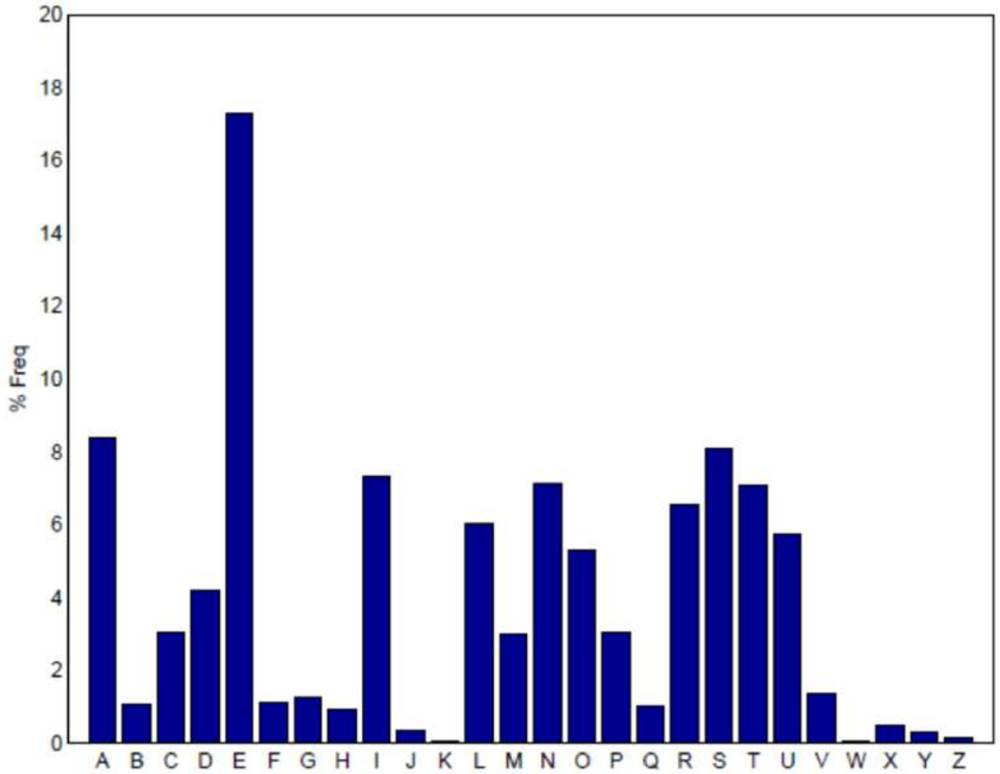
Huffman Code

- **We obtain a VLC code with**
  - Average length of code $\bar{l} = \sum_i p_i l_i = 2.44$
  - Entropy $H(X) = 2.37$

$$\rightarrow \quad \bar{l} > H(X)$$

Compression for French Language

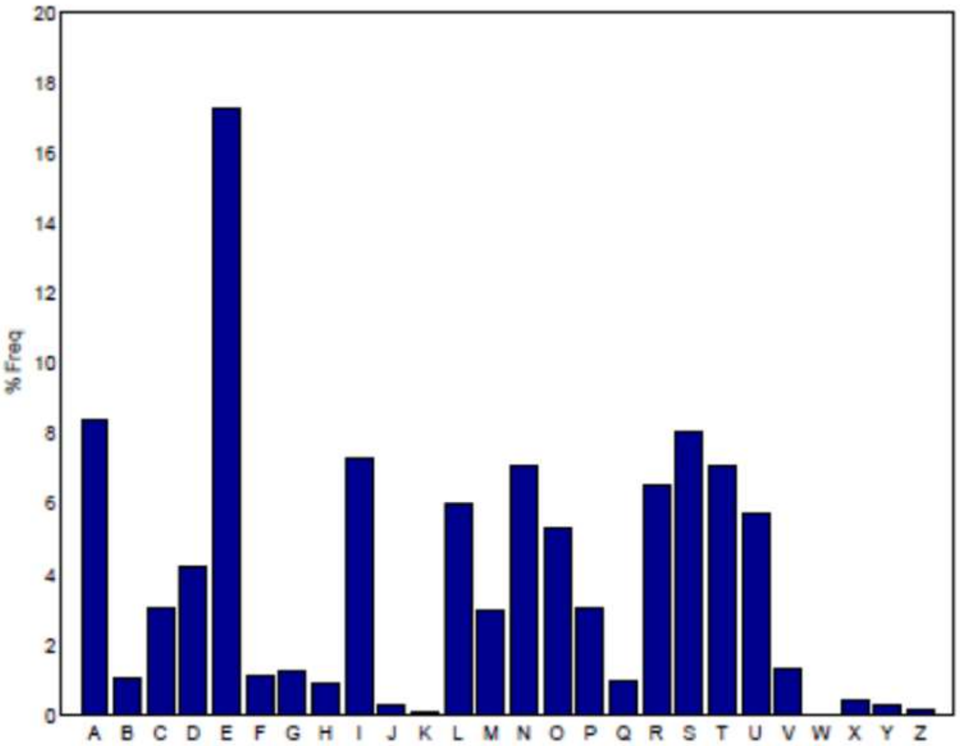
– 26 symbols → 5 bits/symbol without compression
– Entropy: $H(X) = 3.999$ bits/symbol



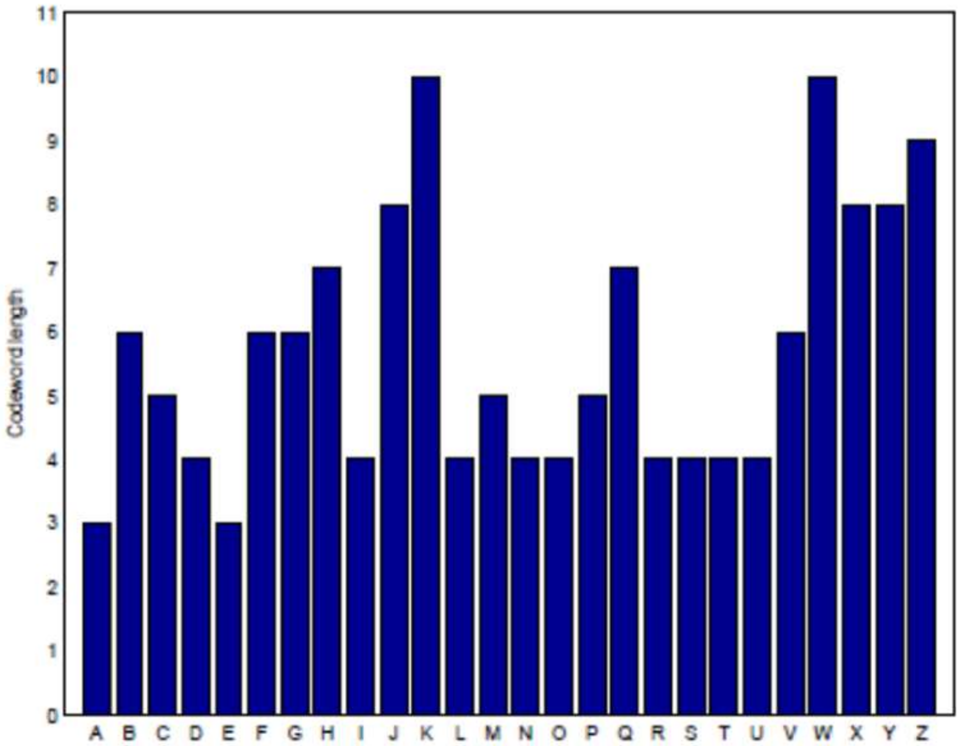
Symbol probabilities

Compression for French Language

– Entropy: $H(X)$ = 3.999 bits/symbol
– Huffman codes: $\bar{l} = 4.041$ bits/symbols



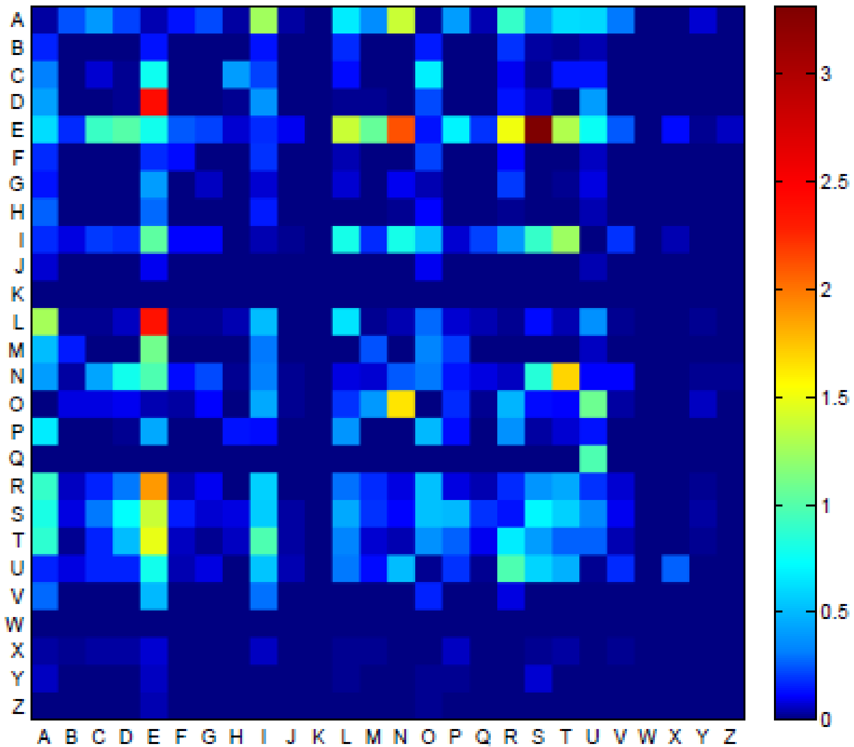
Symbol probabilities



Huffman codeword lengths

Compression for French Language

K=1: Entropy of letters	3.999 bits/blocks	3.999 bits/symbols
K=2: Entropy of digrams	7.440 bits/blocks	3.720 bits/symbols
K=3: Entropy of trigrams	9.452 bits/blocks	3.151 bits/symbols



Digram distribution

ait 1.59%	ent 1.25%	les 0.94%
lle 0.78%	des 0.72%	ant 0.70%
que 0.67%	our 0.63%	ien 0.60%

Most frequent trigrams

RUN-LENGTH ENCODING

- **Run-Length Encoding**
  - Simple form of lossless data compression
  - Runs of data (i.e. sequences in which the same data value occurs in many consecutive data elements)
  - Most useful when the data contains many runs:
    - *Binary images, graphics, line drawings*
    - *Many 1's (white) and few 0's (black)*

Run-Length Encoding

- **Example**

input:

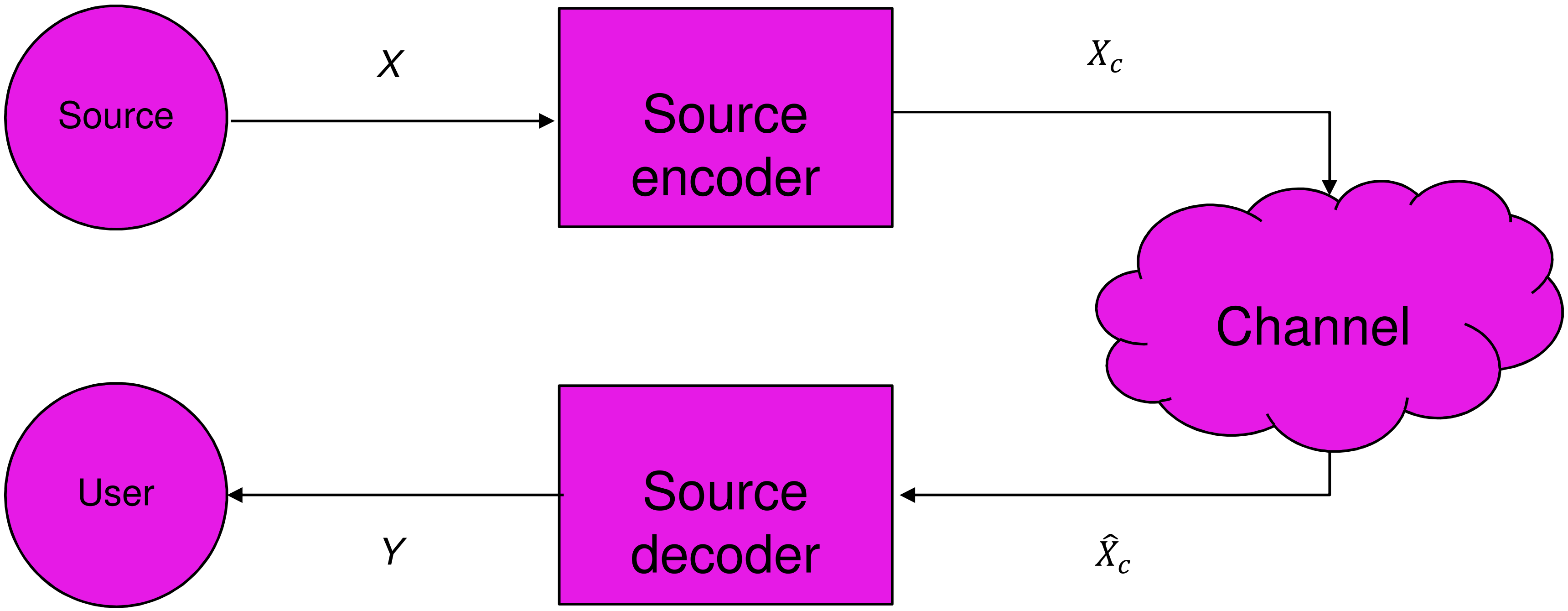
1111111111110111111111111000111111111111111111111111011111111111111

run-length:

(12,1) (1,0) (12,1) (3,0) (24,1) (1,0) (14,1)

RATE-DISTORTION THEORY

- **Lossy source coding scheme**



– when the transmission channel is perfect (no noise), $\hat{X}_c = X_c$

Distortion

- **Lossy coding**
  - the received signal will be different from the initial one.
  - requires some measure of the quality of the reconstructed signal
- **Ideal tool**
  - representative set of human observers who can evaluate the quality
- **Realistic tool**
  - computable criterion

$\Rightarrow$ **distortion measure**

Distortion

- **If we have a set of values**
  - Mean Square Error (MSE)

$$\sigma_d^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2$$

  - Signal to Noise Ratio (SNR): $\mathrm{SNR} = \frac{\sigma_x^2}{\sigma_d^2}$
  - Signal to Noise Ratio in dB: $\mathrm{SNR(dB)} = 10\log_{10}\frac{\sigma_x^2}{\sigma_d^2}$
  - Peak Signal to Noise Ratio (PSNR): $\mathrm{PSNR} = 10\log_{10}\frac{\sigma_{max}^2}{\sigma_d^2}$

- **Theorem (Shannon, 1959)**
  - The minimum amount of information to represent a discrete memoryless Gaussian source of variance σ_x^2 for a given distortion *D* is

$$R(D) = \frac{1}{2}\log_2\left({}^{\sigma_x^2}/_{D}\right)$$

  - *D* being evaluated via a quadratic distortion measure

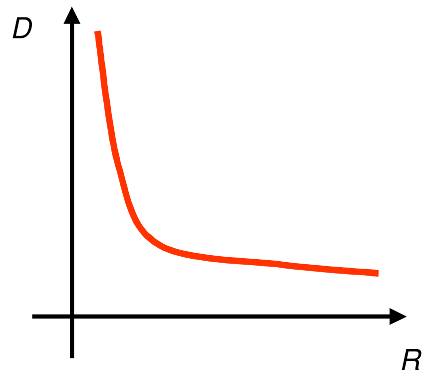
Rate-Distortion

rate distortion curve $R(D)$

$\Leftrightarrow$

lower bound of the rate R attainable for a certain distortion D

Invert the curve $\Rightarrow$ distortion-rate curve



$$D(R) = 2^{-2R}\sigma_x^2$$

and taking the log

$$10\log_{10}D(R) = -6R + 10\log_{10}\sigma_x^2$$

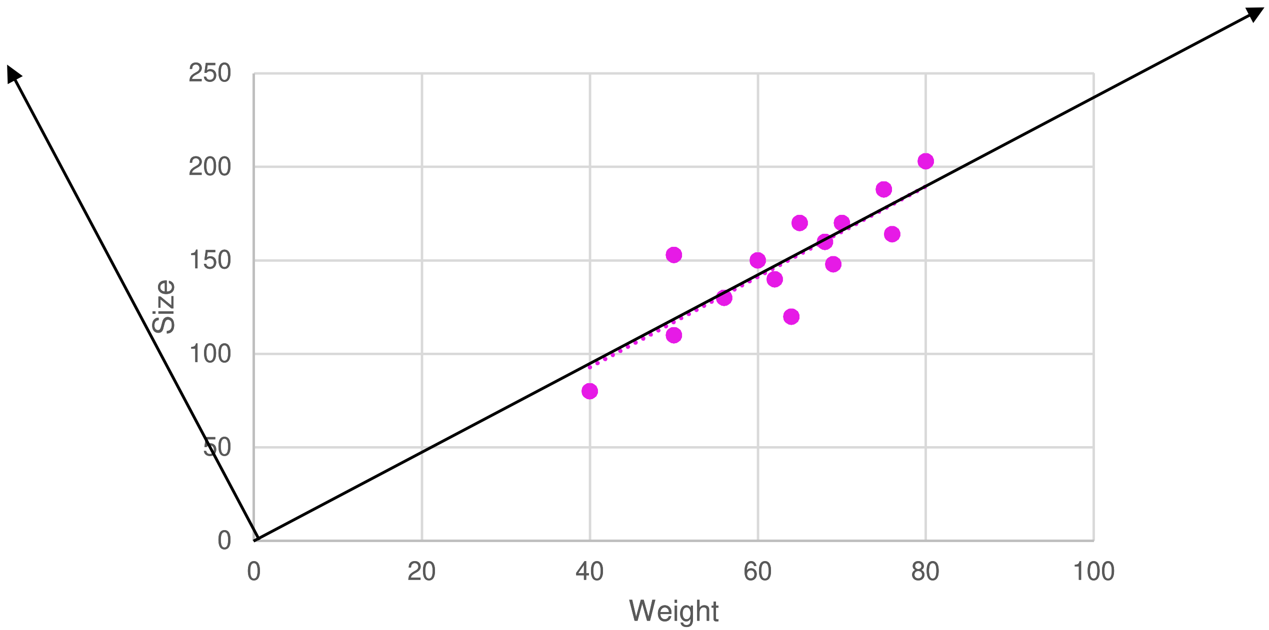
$\Rightarrow$ the distortion (MSE) decreases of 6 dB/bit.

TRANSFORM CODING

- **Strong correlation between neighbor samples**
  - Image coding: neighbor pixels

Transform Coding

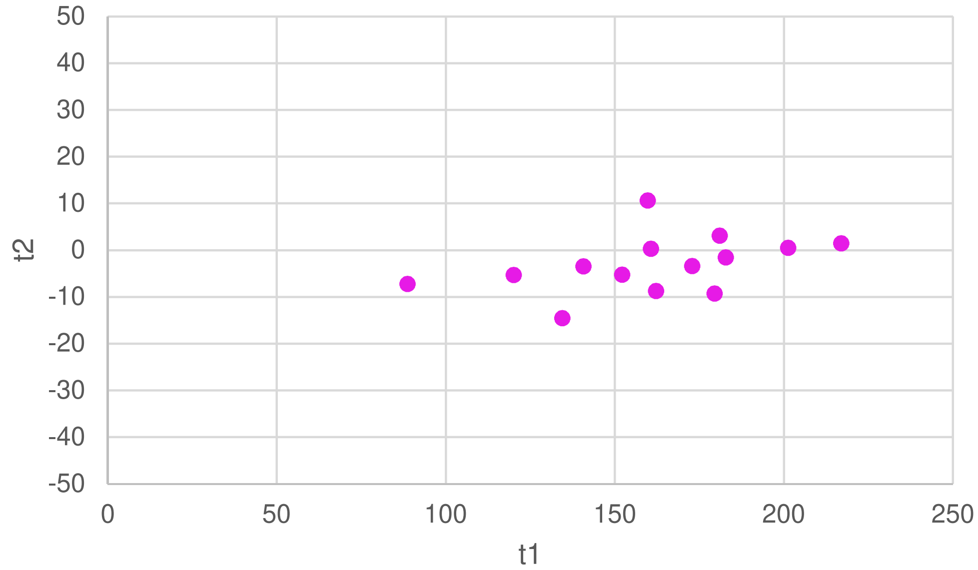
Weight	Size
65	170
75	188
60	150
70	170
56	130
80	203
68	160
50	110
40	80
50	153
69	148
62	140
76	164
64	120



Transform Coding

Weight	Size
65	170
75	188
60	150
70	170
56	130
80	203
68	160
50	110
40	80
50	153
69	148
62	140
76	164
64	120

t1	t2
181,025	3,075
201,349	0,523
160,710	0,270
182,880	-1,540
140,766	-3,458
217,049	1,473
172,908	-3,404
120,080	-5,340
88,680	-7,240
159,769	10,613
162,203	-8,779
152,222	-5,286
179,568	-9,304
134,504	-14,552



- **Transform coding**

  – Express the samples (pixels) in a new base

    ⇒ Reduction of the spatial / temporal correlation

    ⇒ Energy compaction

  – After the transform, the signal is ‘easier’ to quantize

    - Few coefficients are ‘significant’
    - Most coefficients are ‘insignificant’

- **Transform coding**
  - Optimal transform (Karhunen Loeve, KLT)
    - ⇒ optimal compaction
    - ⇒ depends on the signal statistics
  - Sub-optimal transforms (Discrete cosine transform, DCT)
    - ⇒ compaction close to optimal
    - ⇒ does not depend on the signal statistics
  - Other transforms
    - ⇒ Less powerful, but suited to specific applications

- **Linear transforms**
  - Consider some ***base*** made of
    - vectors in $\mathbb{R}^n$
    - matrices of $\mathbb{R}^{n \times n}$

$$\mathbf{U} = [\mathrm{u}_1, \ldots, \mathrm{u}_N] \ ,$$

with

- for the vectors $N = n$
- for the matrices $N = n \times n$

- **Inverse transform**
  - Express any vector $\boldsymbol{s} \in \mathbb{R}^N$ as a linear combination of basis vectors

$$\mathbf{s} = \sum_{k=1}^{N} t_k \mathbf{u}_k = \mathbf{U}\mathbf{t}$$

- **Forward transform**
  - Express the contribution (weight) of each basis vector for a given vector $\boldsymbol{s} \in \mathbb{R}^N$

$$\mathbf{t} = \mathbf{U}^{-1}\mathbf{s} = \mathbf{V}\mathbf{s}$$

- **Unitary transform**
  - The basis is orthonormal

    ⇒ each basis vector is

    - orthogonal to all the other ones
    - of unit norm

$$\langle \mathbf{u}_k , \mathbf{u}_l \rangle = \mathbf{u}_k^T \, \mathbf{u}_l = \sum_{m=1}^{N} \mathrm{u}_{k,m} \, \mathrm{u}_{l,m} = \delta_{k,l} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases}$$

$$\Downarrow$$

$$\mathbf{U}^T \mathbf{U} = \mathbf{U} \, \mathbf{U}^T = \mathbf{I}_N$$

- **Energy conservation**
  - Let **U** be an orthonormal transform matrix, $\mathbf{s} \in \mathbb{R}^N$ and $\mathbf{t} = \mathbf{U}^T\mathbf{s}$

$$\langle \mathbf{s}, \mathbf{s} \rangle = \langle \mathbf{t}, \mathbf{t} \rangle$$

  - Demonstration

$$\begin{aligned} \langle \mathbf{t}, \mathbf{t} \rangle &= \mathbf{t}^T\mathbf{t} \\ &= (\mathbf{U}^T\mathbf{s})^T(\mathbf{U}^T\mathbf{s}) \\ &= \mathbf{s}^T\mathbf{U}\,\mathbf{U}^T\mathbf{s} \\ &= \mathbf{s}^T\mathbf{s} = \langle \boldsymbol{s}, \boldsymbol{s} \rangle \end{aligned}$$

    as

$$\mathbf{U}\,\mathbf{U}^T = \mathbf{I}_N$$

  - Distortion in signal domain = distortion in transform domain
  - Fundamental property to perform ‘bit allocation’ in the transform domain

- **Separable transform**
  - Consider the case where the signal to be processed is a matrix (a block)
  - An orthonormal transform is said to be *separable* if the transform of some block $\mathbf{s} \in \mathbb{R}^{n \times n}$ can be expressed as

$$\mathbf{t} = \mathbf{U}\mathbf{s}\mathbf{U}^{T}$$

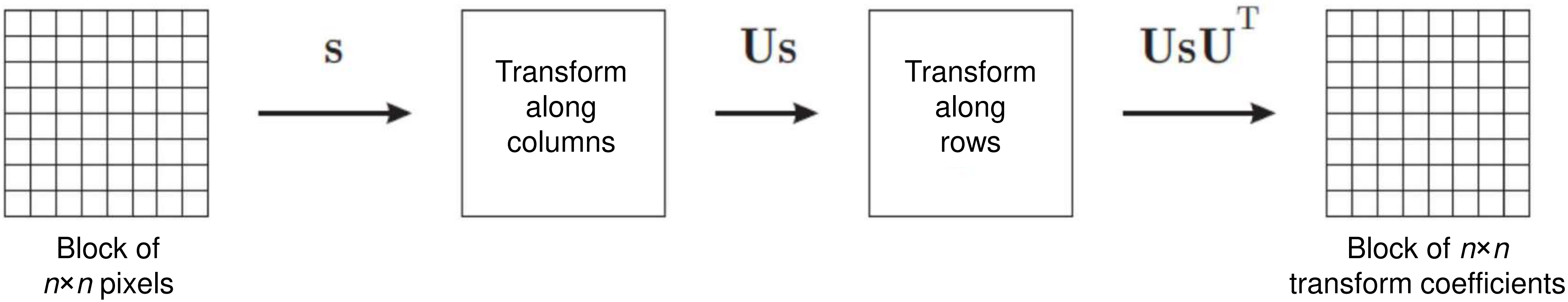
  - where $\mathbf{t} \in \mathbb{R}^{n \times n}$ is the transformed block and $\mathbf{U} \in \mathbb{R}^{n \times n}$ an orthonormal transformation matrix

  - Inverse transform

$$\mathbf{s} = \mathbf{U}^{T}\mathbf{t}\mathbf{U}$$

Transform Coding

- **Separable transform**



  - Of practical interest
    - non-separable transform : multiplication of a $1 \times n^2$ vector by an $n^2 \times n^2$ matrix
    - separable transform : two multiplications by matrices of size $n \times n$

complexity $O(n^4) \Rightarrow$ complexity $O(n^3)$

DISCRETE COSINE TRANSFORM (DCT)

- **Discrete Cosine Transform (DCT)**
  - DCT is sub-optimal
    - Less energy compaction than KLT
    - But good approximation for highly correlated Markov process
  - The transform matrix $\mathbf{U}_{\mathrm{DCT}}$ has the following components

$$U_{i,k} = \alpha_i \cos\frac{\pi(2k+1)i}{2N} \qquad i,k = 0,\dots,N-1$$

with

$$\alpha_0 = \sqrt{\frac{1}{N}} \quad \text{and} \quad \alpha_i = \sqrt{\frac{2}{N}} \ \text{for}\ \ i > 0$$

Discrete Cosine Transform (DCT)

- **Separable transform**
  - DCT is a separable transform

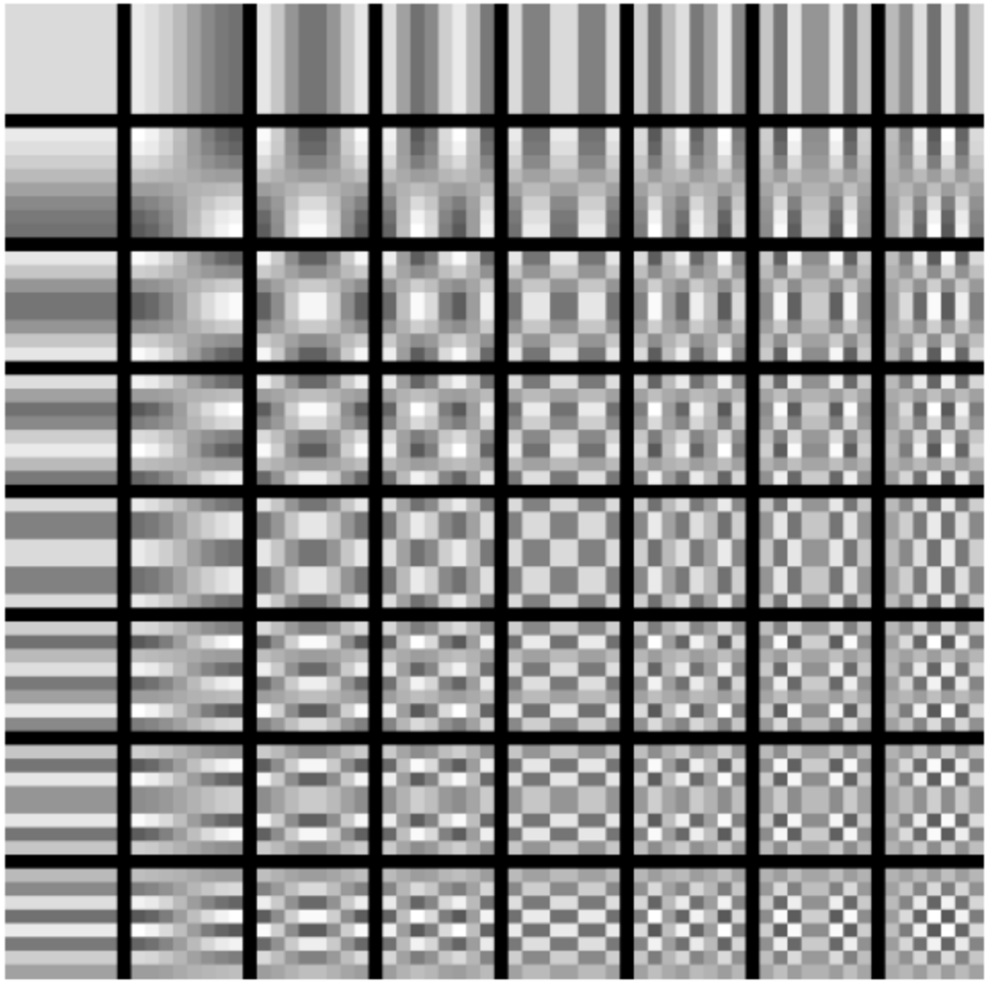
  - Given a block $\boldsymbol{s} \in \mathbb{R}^{n \times n}$
  - Its transform $\mathbf{t} \in \mathbb{R}^{n \times n}$ is given by

$$\mathbf{t} = \mathbf{U}\mathbf{s}\mathbf{U}^{\boldsymbol{T}}$$

⇒ straightforward to obtain a 2D-DCT

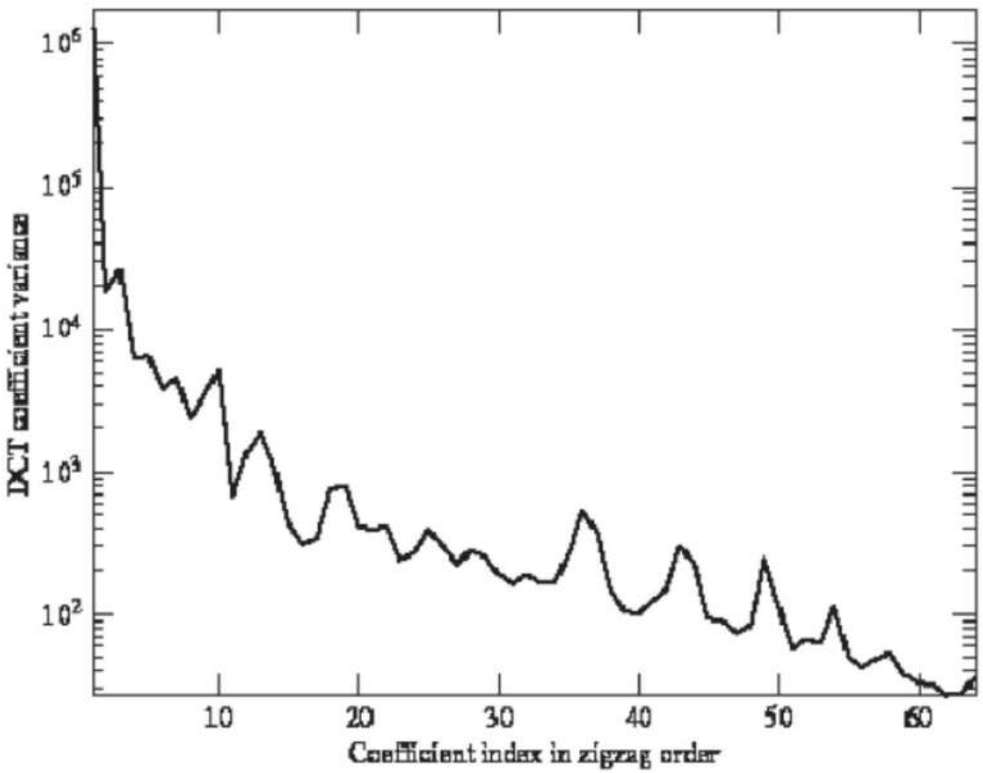
Discrete Cosine Transform (DCT)

- **2D DCT basis matrices of size 8×8**



Discrete Cosine Transform (DCT)

- **Variance of the transformed coefficients**



- **Application to image compression**

Original image: 24 bpp

0.748 bpp (CR 32.080) PSNR 33.45 dB

0.207 bpp (CR 115.959) PSNR 29.50 dB

  - DCT is usually applied on small size blocks ⇒ block artifacts

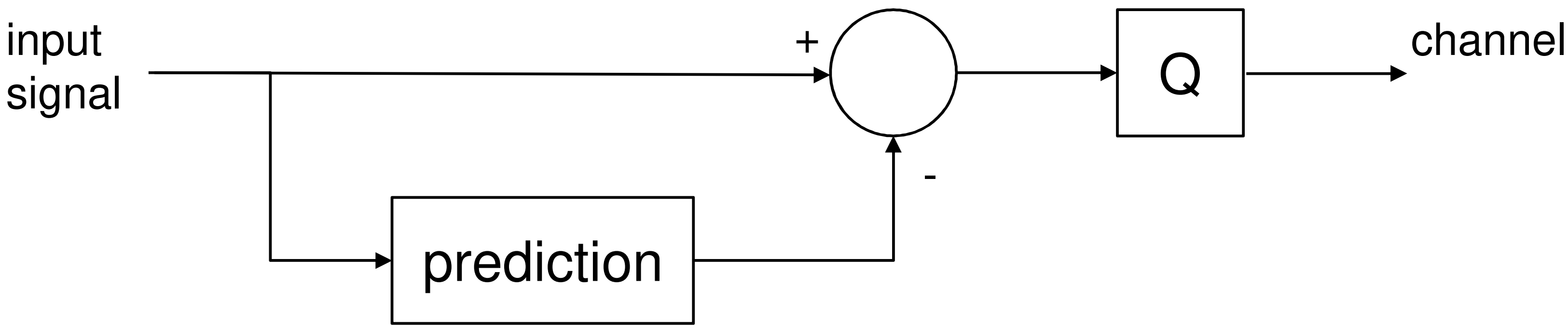
PREDICTIVE CODING

- **DPCM Coding**
  - DPCM : differential pulse code modulation
  - This is NOT a modulation technique, but a source coding technique
  - Exploit the fact that successive samples are redundant
  - Rate distortion curve

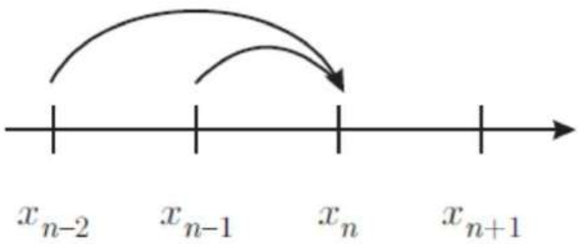
$$D(R) = \varepsilon_X \sigma_X^2 \, 2^{-2R}$$

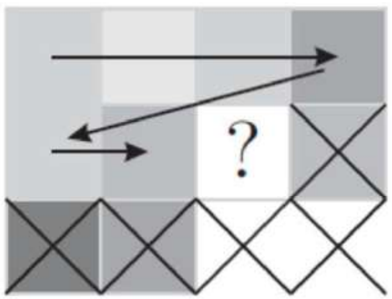
  - Decreasing $\sigma_X^2 \Rightarrow$ Reducing the distortion $D(R)$

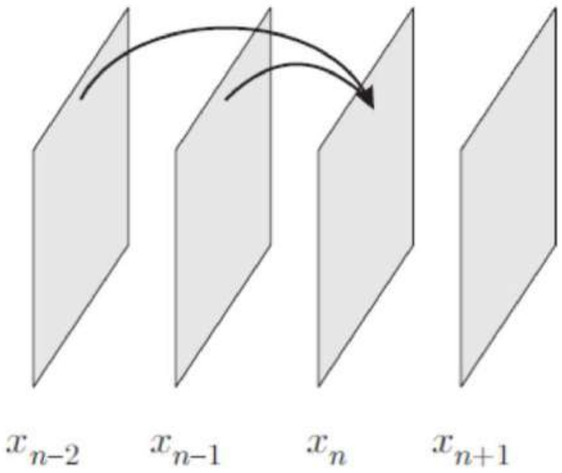
- **DPCM basic principle**
  1. Compute $\hat{x}_n$, a prediction of the sample x_n
  2. Evaluate the prediction error $e_n = x_n - \hat{x}_n$
  3. Quantize the prediction error and transmit it $\tilde{e}_n = Q(e_n)$



- **Prediction must be *causal***
  - past samples (1D)
  - already processed pixels (2D)
  - past images (2D+t)







- **Simple example : one step prediction**
  - Assume a 1D signal, stationary, with zero mean $(E(x_n) = 0)$ and autocorrelation $\gamma_n = E(x_k x_{k-n})$
  - Assume
$$\hat{x}_n = a_1 x_{n-1}$$
  - Problem :

How should we choose a_1 in such a way that the prediction error is minimized?

Predictive Coding

– Prediction error

$$e_n = x_n - \hat{x}_n = x_n - a_1 x_{n-1}$$

– Variance of the prediction error (MSE)

$$E(e_n^2) = E\left((x_n - a_1 x_{n-1})^2\right)$$

$$= E(x_n^2) - 2a_1 E(x_n x_{n-1}) + a_1^2 E(x_{n-1}^2)$$

$$= \gamma_0 - 2a_1\gamma_1 + a_1^2\gamma_0$$

Predictive Coding

- Minimization of the MSE

$$\frac{dE(e_n^2)}{da_1} = 0$$

$$\frac{dE(e_n^2)}{da_1} = -2\gamma_1 + 2a_1\gamma_0 = 0$$

- Therefore

$$a_1 = \frac{\gamma_1}{\gamma_0}$$

- In practice: autocorrelation estimated from signal samples

$$\hat{\gamma}_n = \frac{1}{N-n}\sum_{i=1}^{N-n} x_i x_{i+n} \qquad n = 0,1$$

Predictive Coding

- **Prediction on *p*-steps**
  - Assume a 1D signal, stationary, with zero mean $(E(x_n) = 0)$ and autocorrelation $\gamma_n = E(x_k x_{k-n})$
  - Exploit *p* past samples to predict the current one

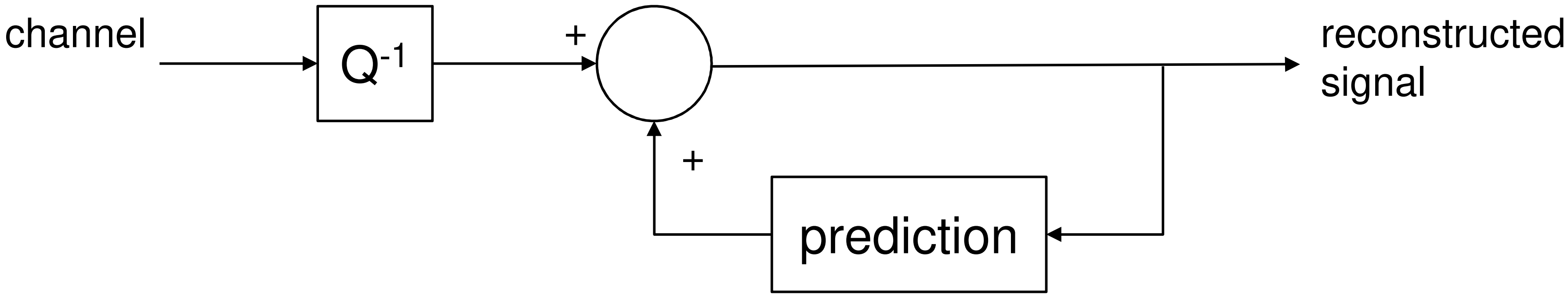
$$\hat{x}_n = \sum_{i=1}^{p} a_i x_{n-i}$$

  - Problem :

How should we choose a_i in such a way that the prediction error is minimized?

Predictive Coding

- **Decoder side: signal reconstruction**



which corresponds to

$$x_n^r = \tilde{e}_n + a_1 x_{n-1}^r$$

Predictive Coding

Accumulation of the errors

$$x_n - x_n^r = (e_n + a_1 x_{n-1}) - (\tilde{e}_n + a_1 x_{n-1}^r)$$

$$= (e_n - \tilde{e}_n) + a_1 (x_{n-1} - x_{n-1}^r)$$

– The reconstructed signal error depends on two terms

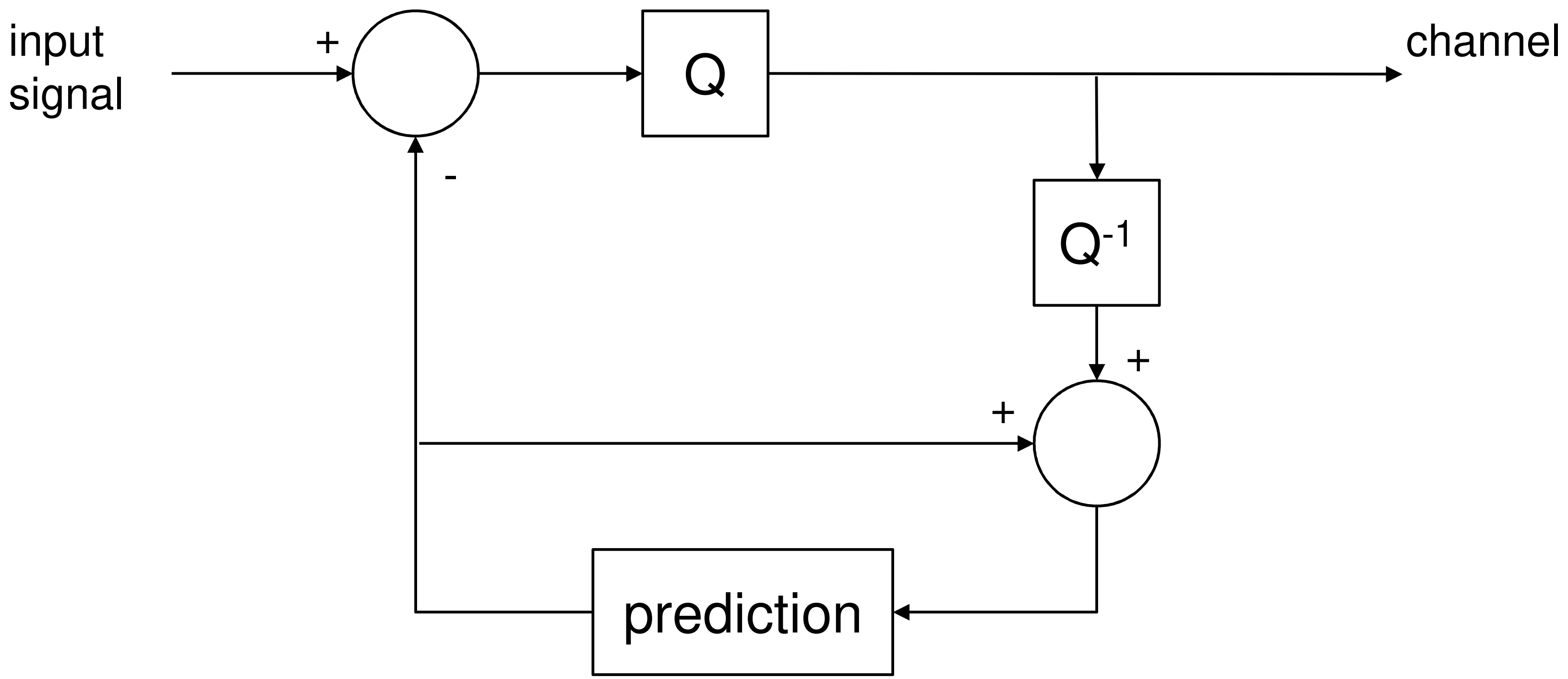
The quantization error

$$(e_n - \tilde{e}_n)$$

The quantization error of the previous sample

$$a_1 (x_{n-1} - x_{n-1}^r)$$

- **DPCM**
  - Introduction of a decoder at the encoder side
  - Prediction at the encoder is now obtained from samples which will also be available at the decoder



Predictive Coding

Prediction

$$x_n = e_n + \sum_{i=1}^{p} a_i x_{n-i}^r$$

Reconstructed samples at the encoder/decoder side

$$x_n^r = \tilde{e}_n + \sum_{i=1}^{p} a_i x_{n-i}^r$$

Predictive Coding

Reconstruction error

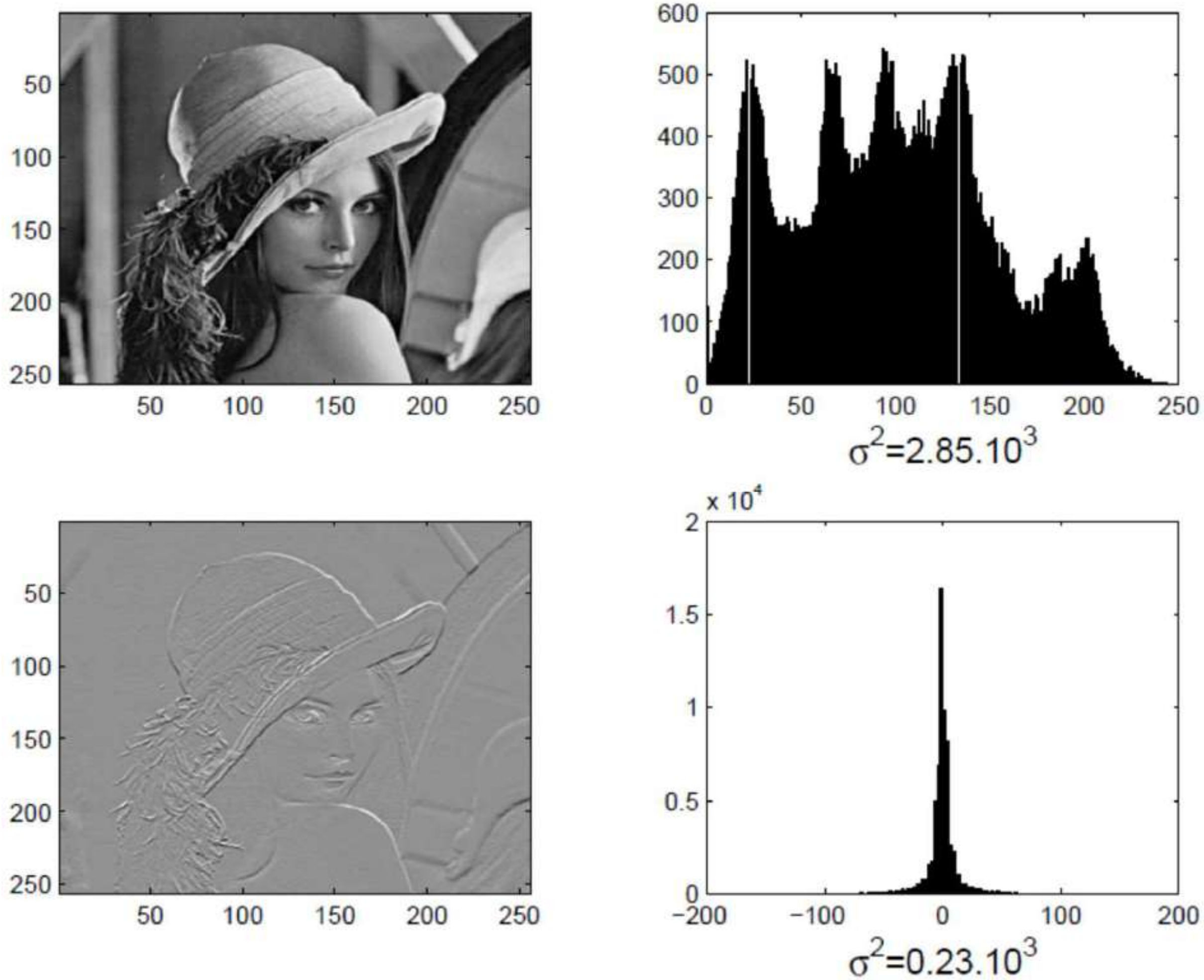
$$x_n - x_n^r = \left(e_n + \sum_{i=1}^{p} a_i x_{n-i}^r\right) - \left(\tilde{e}_n + \sum_{i=1}^{p} a_i x_{n-i}^r\right)$$

$$= (e_n - \tilde{e}_n) + \sum_{i=1}^{p} a_i (x_{n-i}^r - x_{n-i}^r)$$

The reconstruction error only depends on the error introduced by quantization (in the absence of transmission error)

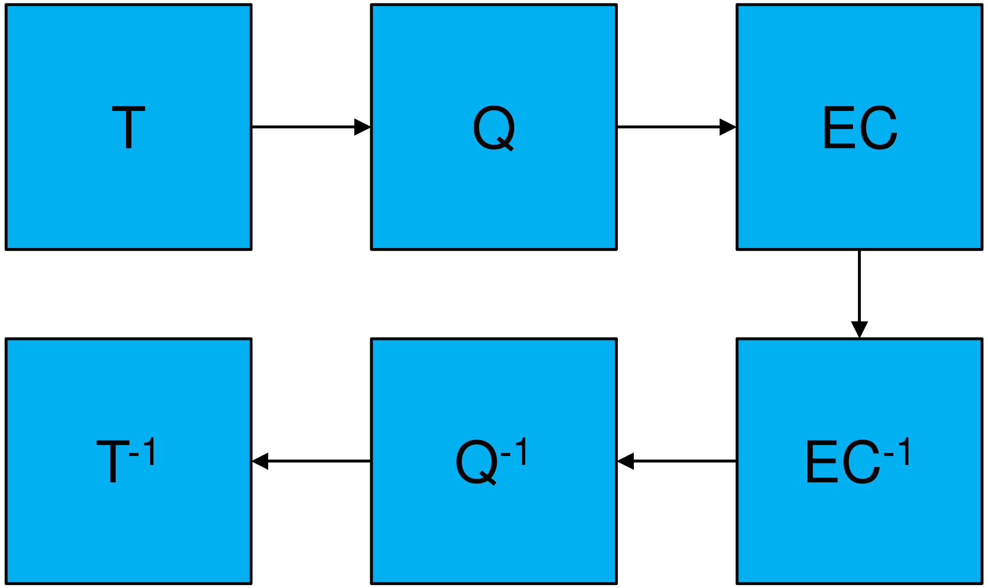
$$x_n - x_n^r = (e_n - \tilde{e}_n)$$

- **Example: DPCM on an image**
  - DPCM by predicting the samples by their left neighbor



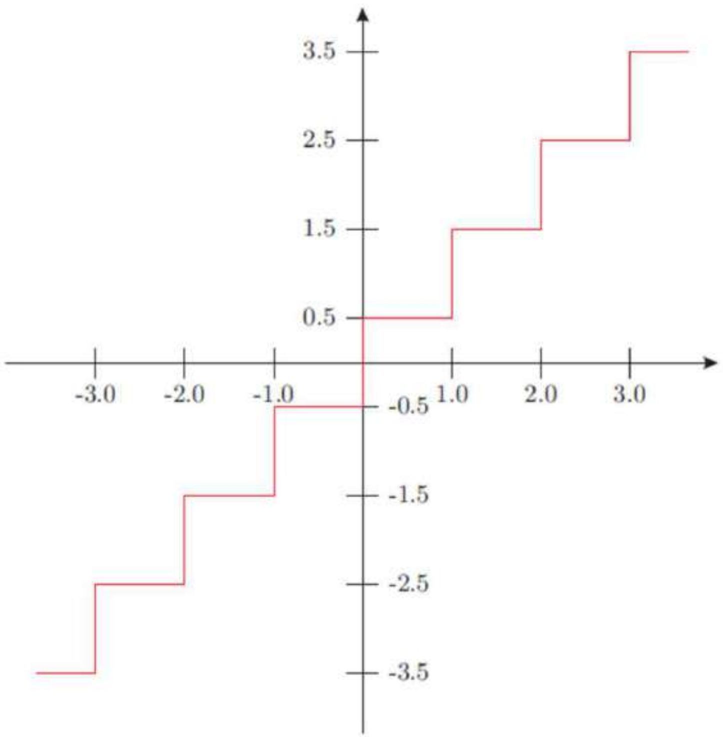
SCALAR QUANTIZATION

Scalar quantization



– Quantization is done after linear transform

- **Uniform scalar quantization**
  - Decision intervals $[b_{i-1}, b_i]$, $i = 1 \ldots M$ of fixed length Δ.
  - Consider the following quantizers (for given y_i)



Mid-rise quantizer

Values 0 or close to 0 are reconstructed as -0.5 or 0.5

EXAMPLE: JPEG

JPEG

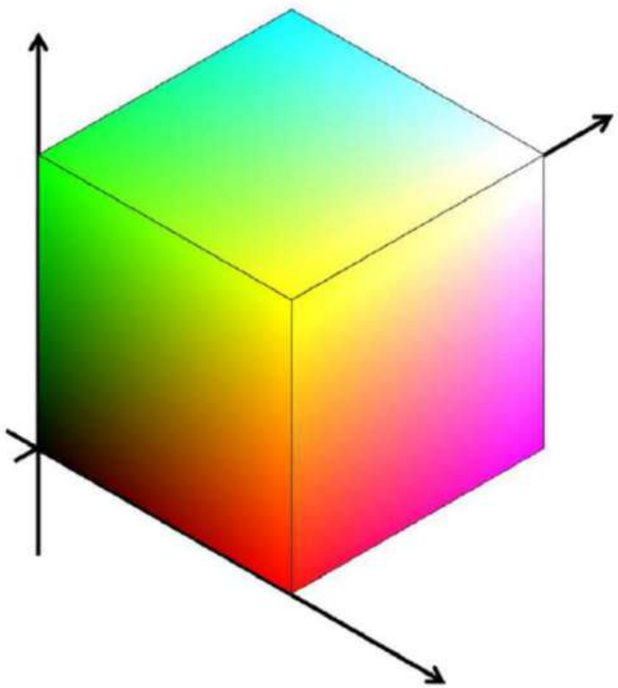
- Joint Photographic Experts Group (JPEG)
- ISO/IEC 10918-1
- Baseline standard issued in 1992

- Applications
  - Digital photography
  - Internet
  - Printing
  - Medical imaging

- Reference: http://www.jpeg.org

Representation of digital video/images

- **RGB color space**
  - Red-Green-Blue
  - Inspired by trichromatic human vision



Representation of digital video/images

- **Luminance-chrominance color space**
  - YCbCr
  - Decorrelate the data

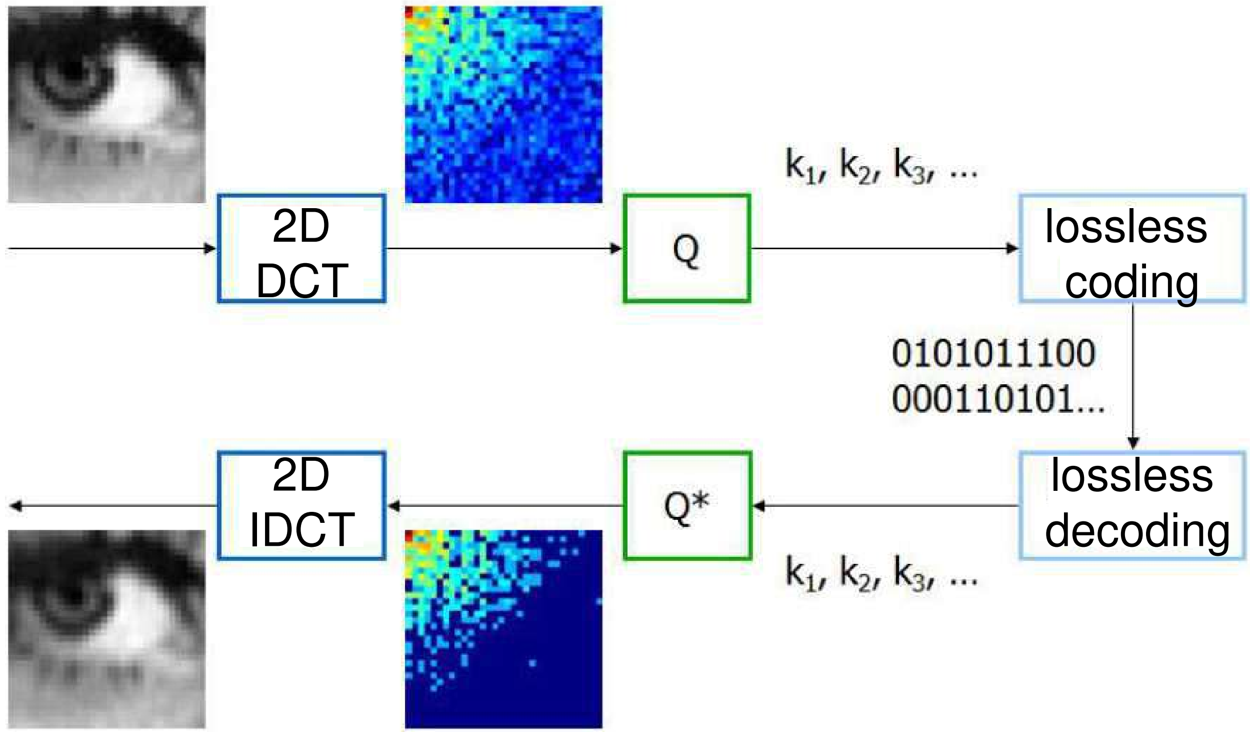
$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 1.4021 \\ 1.0 & -0.3441 & -0.7142 \\ 1.0 & 1.7718 & 0.0 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix}$$

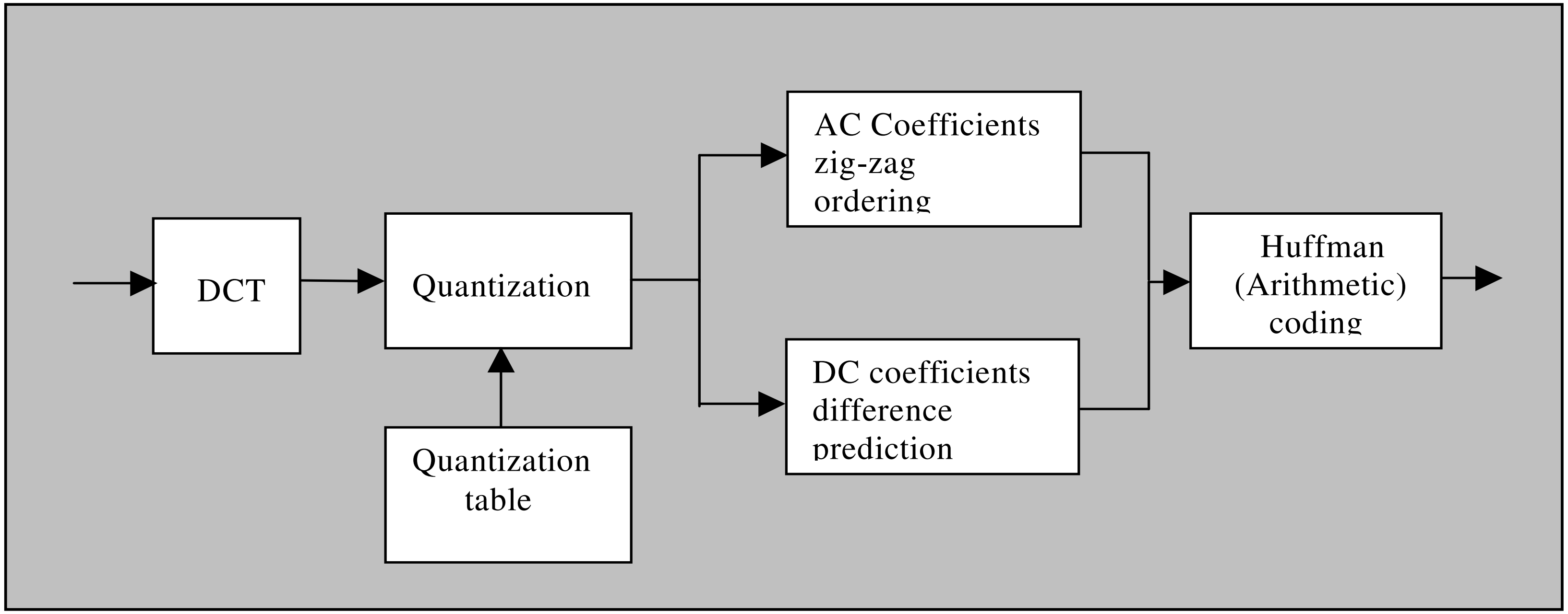
  - Chroma subsampling
    - *4:2:2 horizontal subsampling by 2*
    - *4:2:0 horizontal & vertical subsampling by 2*

JPEG

- Image partitioned into blocks of 8x8 pixels
  - *Small block → stationary signal*
  - *Large block → exploit correlation across a larger scale*
- Transform coding:
  - *2D DCT*
- Weighted scalar quantization
  - *Takes into account Human Visual System and its sensitivity to different frequencies*



JPEG



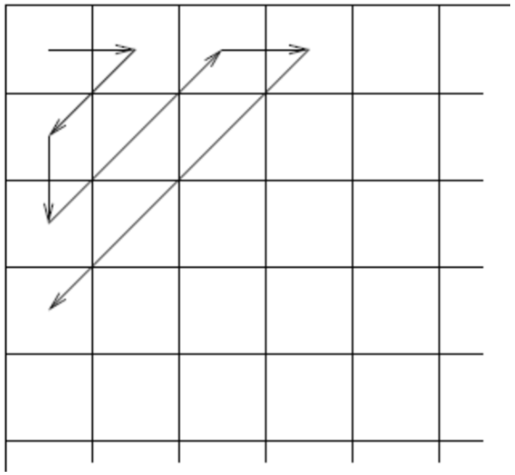
- Zig-zag scan
- DC coefficients coded by DPCM
- AC coefficients coded by Huffman and run-length coding

- **Run-length coding**

Quantized 8x8 DCT block

61	16	3	0	0	0	0	0
3	3	0	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Zig-zag scan



AC coefficients: run-length of zeros + Huffman coding

DC	0,16	0,3	1,3	0,3	7,-1	EOB

JPEG

JPEG at 0.5 bpp (48:1)

JPEG

JPEG at 0.25 bpp (96:1)

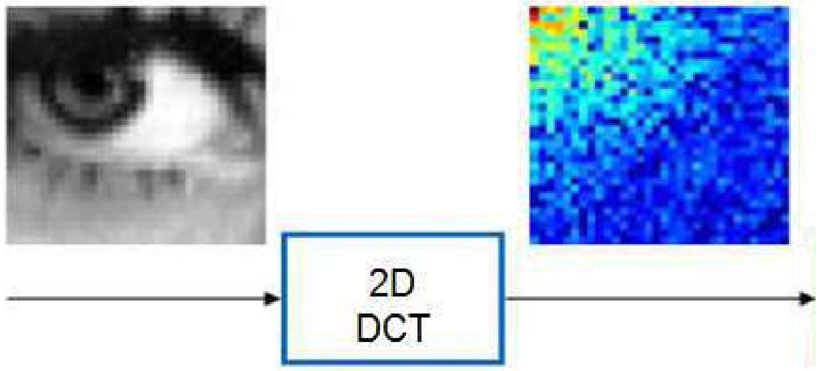
JPEG

JPEG at 0.125 bpp (192:1)

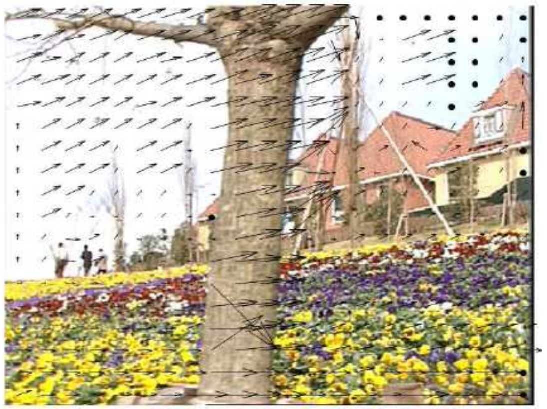
EXTENSION TO VIDEO CODING

Compression: hybrid video coding

- Hybrid video coding
  - Transform coding: compact the energy of the signal



  - Predictive coding:
  - Spatial: prediction from previously encoded blocks
  - Temporal: motion compensated prediction



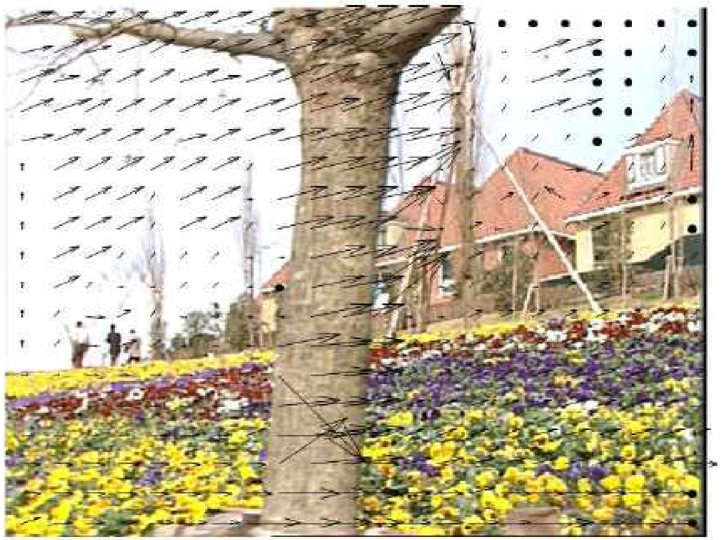
  - Entropy coding: bit-stream based on probabilities of symbols

Temporal motion compensated prediction

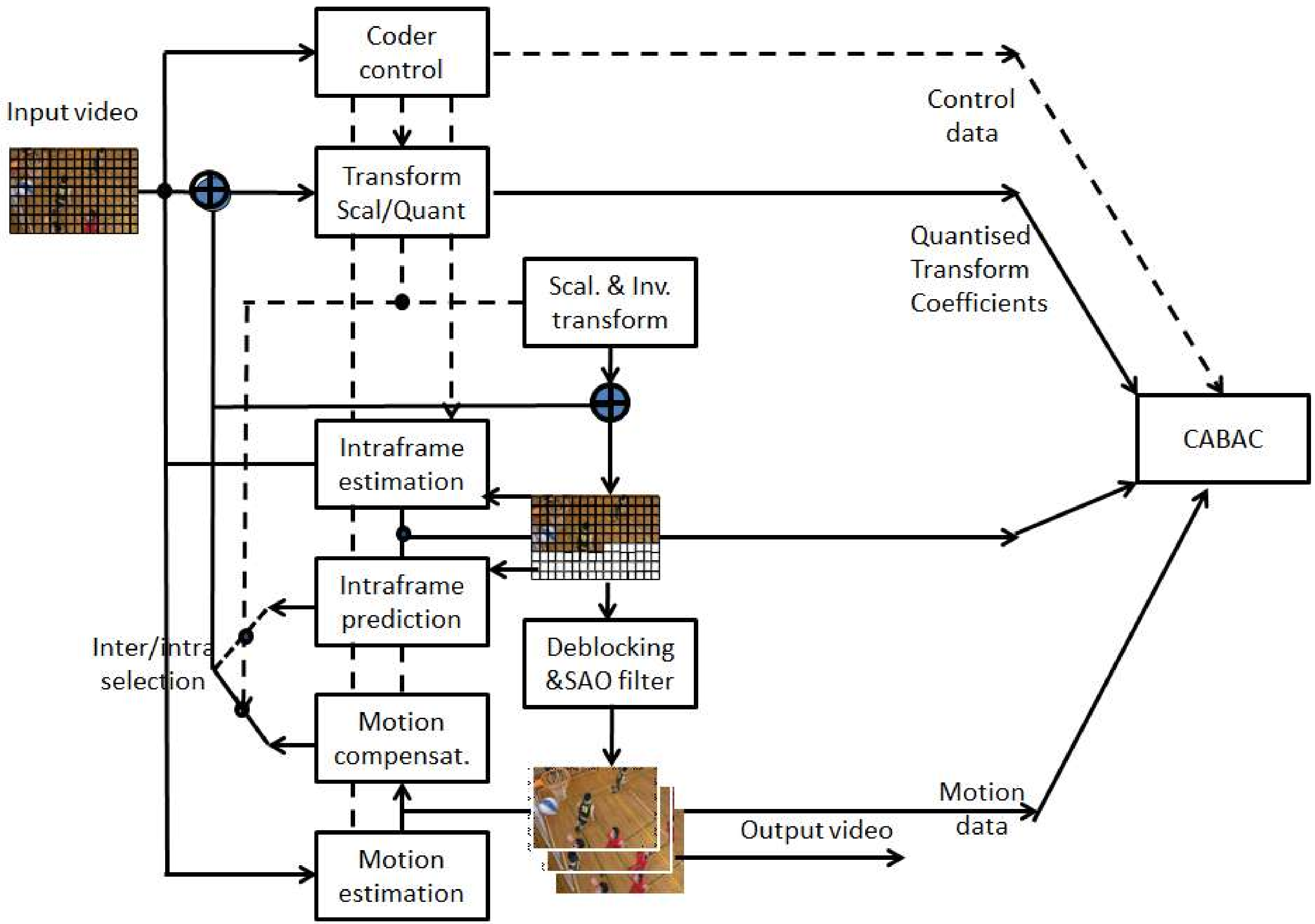
- For every block in a current frame, the most similar block in a reference frame is searched for

- Block matching motion estimation

$$\min \sum_{B} \left\| f(x, y, t) - f(x - d_x, y - d_y, t - \Delta t) \right\|$$



HEVC hybrid video coding scheme

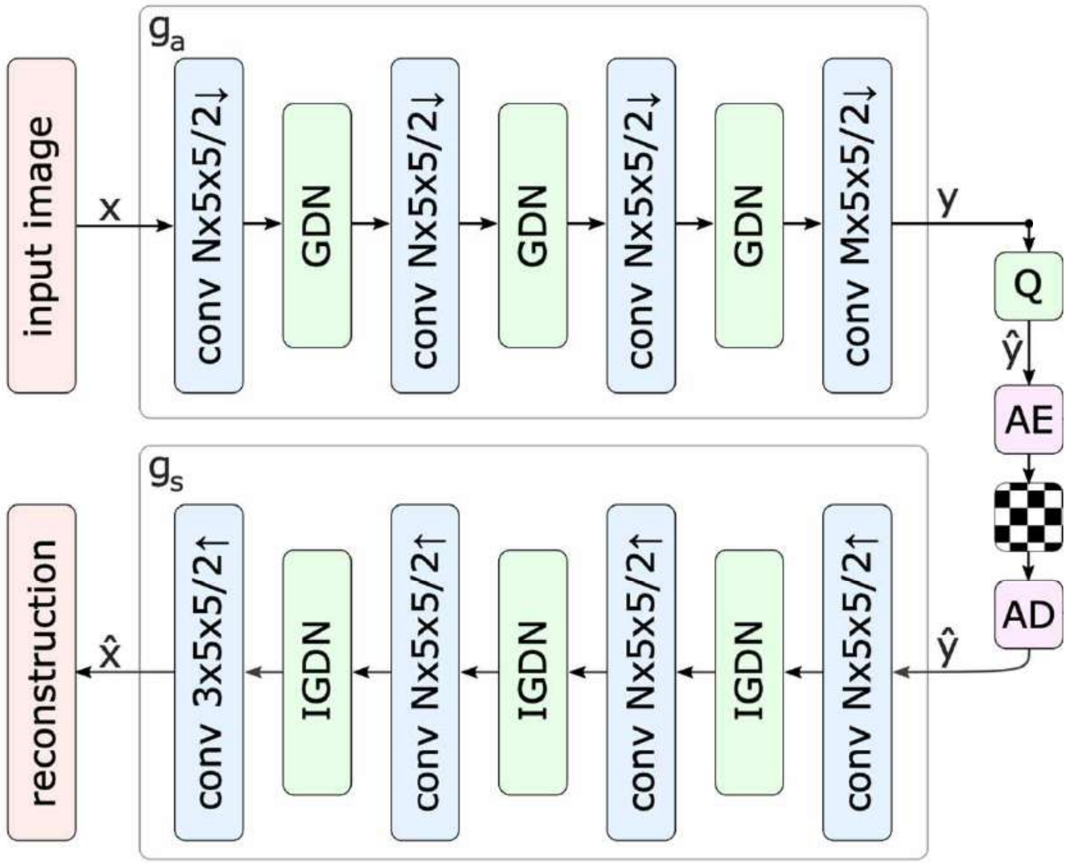


- **Hybrid video coding**
  - Transform coding: DCT-like integer transform
  - Predictive coding: Intra- or Inter- prediction
  - Entropy coding: context-adaptive binary arithmetic coding (CABAC)

LEARNING-BASED (LOSSY) IMAGE COMPRESSION

Learning-based image compression

- **Variational auto-encoder (VAE)-based compression**



- Optimized end-to-end
- Quantization
  - Non differentiable
  - Backward pass (in training):
    $$\hat{y}_i = y_i + \mathcal{U}\left[-\frac{1}{2}, \frac{1}{2}\right]$$
  - Inference:
    $$\hat{y}_i = \mathrm{round}(y_i)$$
- Entropy coding
  - Differential entropy for training
- What is learned:
  - Analysis transform
  - Synthesis transform
  - Probability distribution of latent variables

Ballé, J., Laparra, V., Simoncelli, E., "End-to-end optimized image compression". In *International Conference on Learning Representations*. 2016

OBJECTIVE (PERCEPTUAL) QUALITY METRICS

MSE and PSNR

- **Mean Square Error (MSE)**

$$\mathcal{D} = \frac{1}{N\,M} \sum_{n=1}^{N} \sum_{m=1}^{M} (f_{n,m} - \tilde{f}_{n,m})^2$$

- **Peak Signal to Noise Ratio (PSNR)**

$$\mathrm{PSNR} = 10\,\log_{10}\left(\frac{255^2}{\mathcal{D}}\right)$$

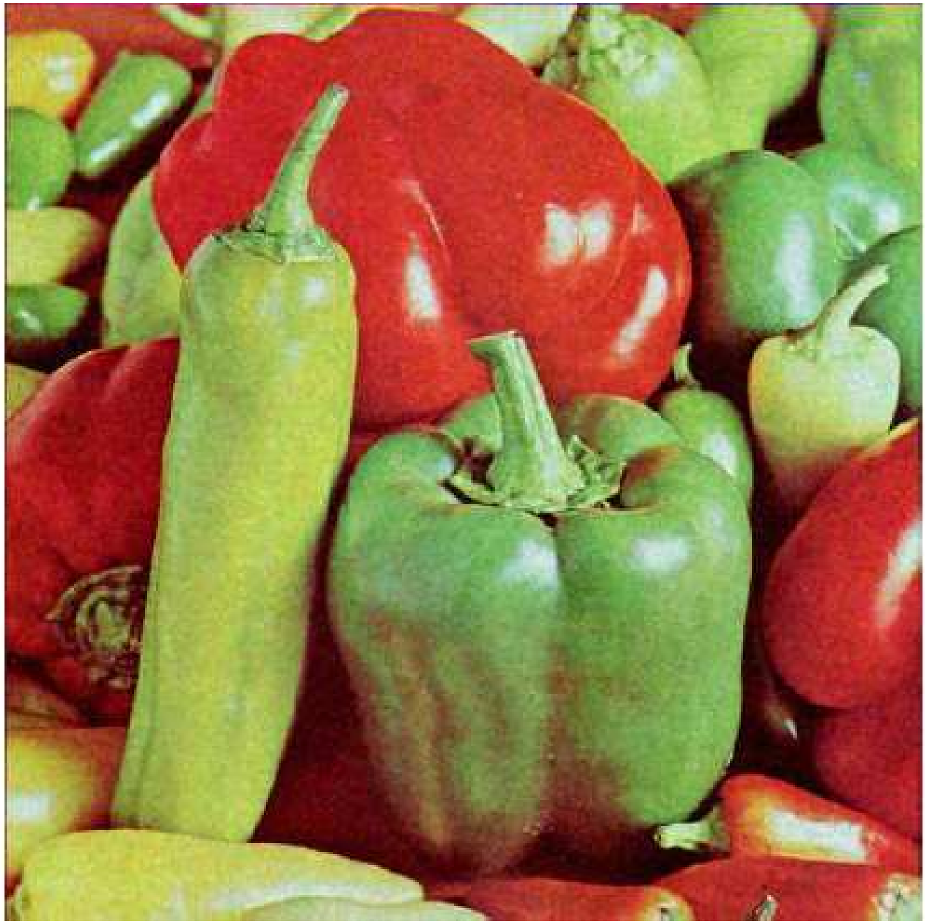
- **Widely used**
  – Simple to use and to understand
  – Allows analytical expressions for optimization
- **Not always correlated with perceptual quality!**

Examples

- **White noise, σ=4**

original image

MSE=16



Examples

- **Concentrated noise, 100x100 patch**

original image

MSE=16

Examples

- **Noise on edges (Sobel filter)**

edges

MSE=16

Examples

- **Noise in high spatial frequencies**

original image

MSE=16

Examples

- **Chroma components subsampling**

original image

MSE=21.27

REFERENCES

References

- T. Cover et J. Thomas, Elements of Information Theory, 2nd ed., Wiley, 2006.
- K. Sayood, Introduction to data compression, 2nd ed., Morgan Kaufman, 2005.
- J. B. Anderson, Digital Transmission Engineering, Prentice Hall, Upper Saddle River, 1998.
- R. E. Blahut, Digital Transmission of Information, Addison-Wesley, Reading, 1990.
- R.G. Gallager, Information Theory and Reliable Communication, McGraw-Hill, New-York, 1968.
- J. B. Anderson and S. Mohan, Source and Channel Coding, Kluwer, Boston, 1993.
- A. K. Jain, Fundamentals of Digital Image Processing, Prentice Hall, Englewood Cliffs, 1989
- J. G. Proakis, Digital Communication, 4th edition, Mc Graw Hill, Boston, 2000
- P. P. Vaidyanathan, Multirate systems and filter banks, Prentice Hall, Englewood Cliffs, 1993

References

- **D. Taubman and M. Marcellin, "JPEG 2000: Image Compression Fundamentals, Standards and Practice", Kluwer Academic Publishers, 2002.**
- **V. Sze, M. Budagavi, G. J. Sullivan, High Efficiency Video Coding (HEVC), Springer, 2014**
- **M. Wien, High Efficiency Video Coding: Coding Tools and Specification, Springer, 2015**