

## Traitement Numérique des Images : Compression et tatouage



J-M. MOUREAUX

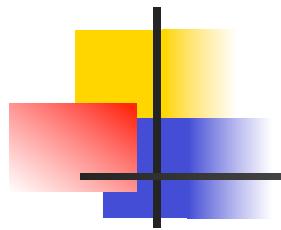
[jean-marie.moureaux@univ-lorraine.fr](mailto:jean-marie.moureaux@univ-lorraine.fr)



TELECOM Nancy

CRAN (Centre de Recherche en Automatique de Nancy) – CNRS UMR 7039 – Université de Lorraine

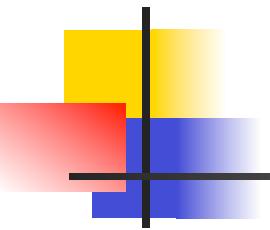




# Plan du Cours

---

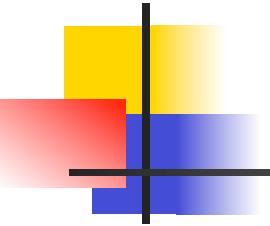
1. Introduction
2. Outils mathématiques de base
3. Stratégie de compression
4. Transformée
5. Quantification
6. Codage entropique
7. Compression d'images fixes : de JPEG à JPEG2000
8. Compression de vidéos : de MPEG I à MPEG IV
9. Transmission de documents confidentiels et sécurité



# Plan du Cours

---

1. Introduction
2. Outils mathématiques de base
3. Stratégie de compression
4. Transformée
5. Quantification
6. Codage entropique
7. Compression d'images fixes : de JPEG à JPEG2000
8. Compression de vidéos : de MPEG I à MPEG IV
9. Transmission de documents confidentiels et sécurité



# **Un peu d'histoire ...**

## **Quelques dates importantes ... dans l'histoire des communications**

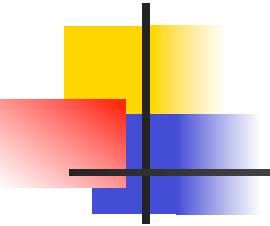
**1880** : Maxwell formalise les lois de l'électromagnétisme

**1888** : Hertz en déduit la notion de propagation

**1894** : Marconi invente la radio (première transmission sans fil)

**Depuis, les communications n'ont cessé de se développer ...**

**Aujourd'hui : mobiles, Internet, WAP, transmission par satellite, réseaux hauts débits, ADSL, ...**



# Encore un peu d'histoire ...

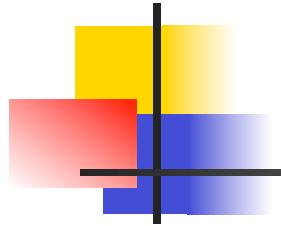
## Quelques dates importantes ... dans l'histoire des codes

**1820** : Braille met au point un code (6 bits) basé sur l'occurrence des mots et des caractères, avec un caractère spécial pour indiquer si le symbole suivant est un mot ou un caractère ⇒ réduction d'environ 20% de l'espace occupé.

**1843** : Morse met au point un code basé sur l'occurrence des caractères pour la transmission par télégraphe.

**Depuis, les codes n'ont cessé de se développer ...**

*Huffman, arithmétique ...*

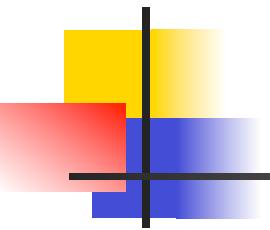


# Les enjeux de la compression

Un bref retour en arrière ... dans les années 70...

...à l'époque on savait s'amuser ...



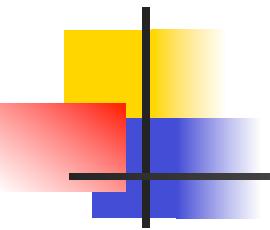


# Les enjeux de la compression

Aujourd'hui une certaine candeur a fait place à un certain réalisme...



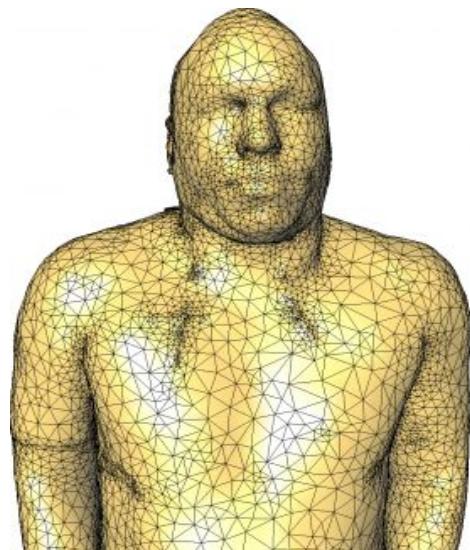
Source : [http://www.jeuxvideo.com/screenshots/00013/00013149\\_062.htm](http://www.jeuxvideo.com/screenshots/00013/00013149_062.htm)



# Les enjeux de la compression

Des objets 3D dont le volume de données atteint aisément  
**plusieurs Gigaoctets !**

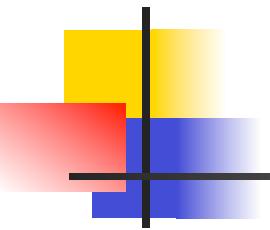
... donc très lourds à manipuler, stocker, transmettre.



(source : <http://shapes.aimatshape.net/>)



Disney · PIXAR  
**MONSTERS, INC.**

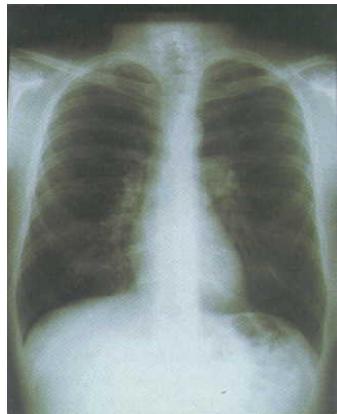


# Les enjeux de la compression

Une imagerie radiologique de plus en plus précise ...

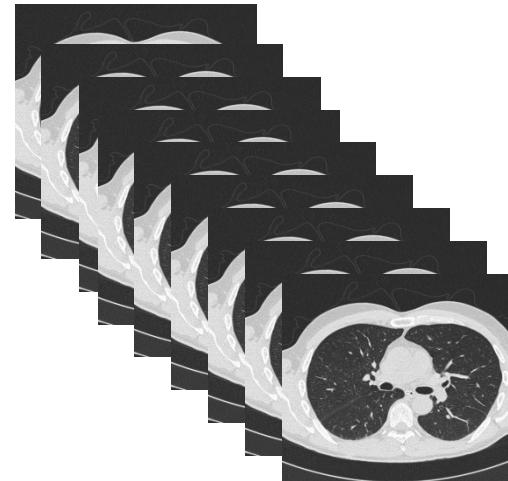
... donc très lourde à manipuler, stocker, transmettre.

Hier (et encore aujourd'hui)



Radiographie des poumons  
(source : <http://stsp.creteil.iufm.fr/article29.html>)

Aujourd'hui



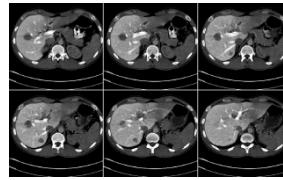
Scanner de poumons

Un examen, c'est environ  
200 Moctets à stocker  
(ou à transmettre) !

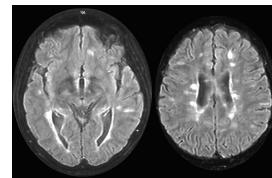
# Contexte de l'imagerie radiologique

Images de plus en plus précises

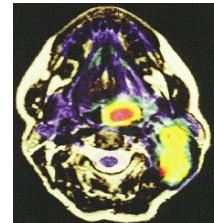
Scanner



IRM



PET



PACS



Diagnostic



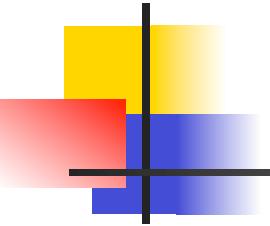
Traitements



Consultations postérieures

PACS : Picture archiving communication system.

→ : Echanges de données numériques



# Les enjeux de la compression

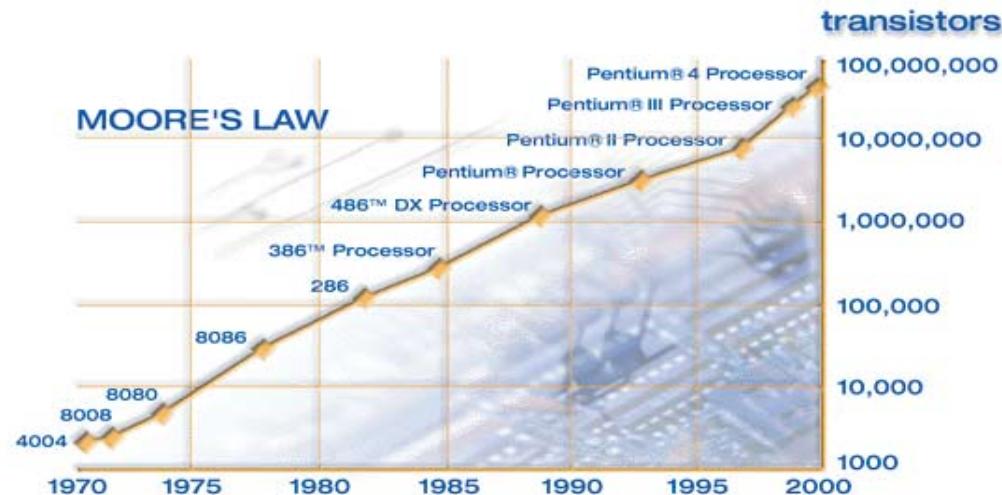
On estime qu'environ **2 Teraoctets** (soit plus de 2000 CD-roms) sont nécessaires pour décrire un long métrage **d'1h30 de cinéma numérique...**

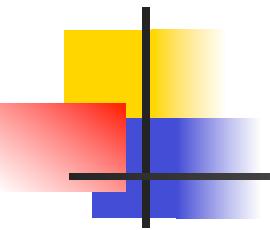


# La compression et les lois

Cyril Northcote Parkinson a établi que **les volumes de données augmenteraient toujours jusqu'à remplir l'espace de stockage disponible.**

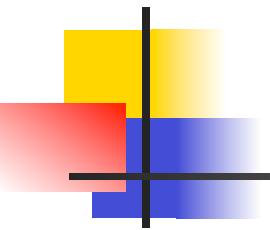
Or la loi de Moore nous permet de savoir que l'espace de stockage et la capacité de traitement des données stockées doublent tous les 18 mois. Les experts de l'industrie prévoient donc que, **d'ici à la fin du 21e siècle, chaque personne sur terre disposera d'un téraoctet de données stockées.** Parkinson est également connu pour sa loi sur l'absorption de la bande passante : « **Le trafic réseau augmente jusqu'à occuper la largeur de bande passante disponible** »





# Compression : les problématiques

- ✓ **Des réseaux hétérogènes** : *informatiques, téléphoniques, radiomobiles, capteurs,...*
- ✓ **Des capacités de stockage qui augmentent moins vite que les besoins** : *lois de Parkinson*
- ✓ **Des données de plus en plus riches** : *images médicales DICOM sur 12 bits,...*
- ✓ **Des standards multiples** : *JPEG, JPEG2000, MPEG2, MPEG4, H264, H265...*
- ✓ **De nouvelles fonctionnalités** : *progressivité, interactivité,...*
- ✓ **Des traitements additionnels** : *tatouage, post-traitements médicaux,...*



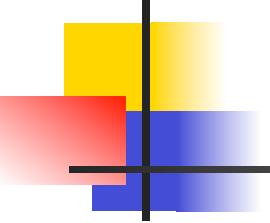
# Pourquoi compresser ?

**Pour transmettre ou pour stocker !**

## **Quelque exemples pour s'en convaincre :**

- standard de télévision NTSC (Etats-Unis) : image numérique 720x480x2 pixels (chacun codé sur 24 bits) environ 30 images/seconde  $\Rightarrow$  environ 249 MHz de bande passante nécessaire or bande passante disponible 4,2 MHz !!!  $\Rightarrow$  nécessité de compresser d'un facteur 59 !!!
- DVD : image 720x576 pixels (chacun codé sur 12 bits)  $\Rightarrow$  4,9 Mbits/image, soit 125 Mbits/s (en pratique débit moyen 4,5 Mbits/s)
- imagerie médicale 3D : 1 scanner poumon 500 images de 512x512 pixels (chacun codé sur 12 bits) soit 187,5 Moctets à stocker (ou à transmettre) !!!
- archives nationales américaines :  $12,5 \cdot 10^9$  Moctets !!!

*capacité du cerveau humain ...  $12,5 \cdot 10^7$  Moctets !!!*
- séquence vidéo composée d'images couleur 512x512 pixels (chacun codé sur 24 bits) transmise sur une ligne téléphonique à 9600 bauds : 11 minutes / image !!!



# Compresser Pour Transmettre...

... à travers les réseaux

Informatiques (Internet) :

*fichiers texte, images, son, vidéo...*

Téléphoniques :

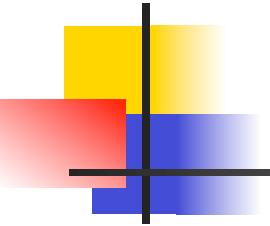
*voix numérisée, minitel, multimedia (ADSL)*

Radio-mobiles :

*GSM, UMTS, GSM de 3ième génération...*

Satellites :

*Sondes spatiales, télévision à haute définition...*



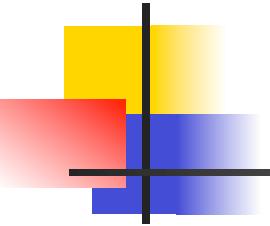
# Compresser Pour Stocker...

**... sur des supports du type :**

Disques durs, clés USB, disquettes : *fichiers*

CD : *Sons, images*

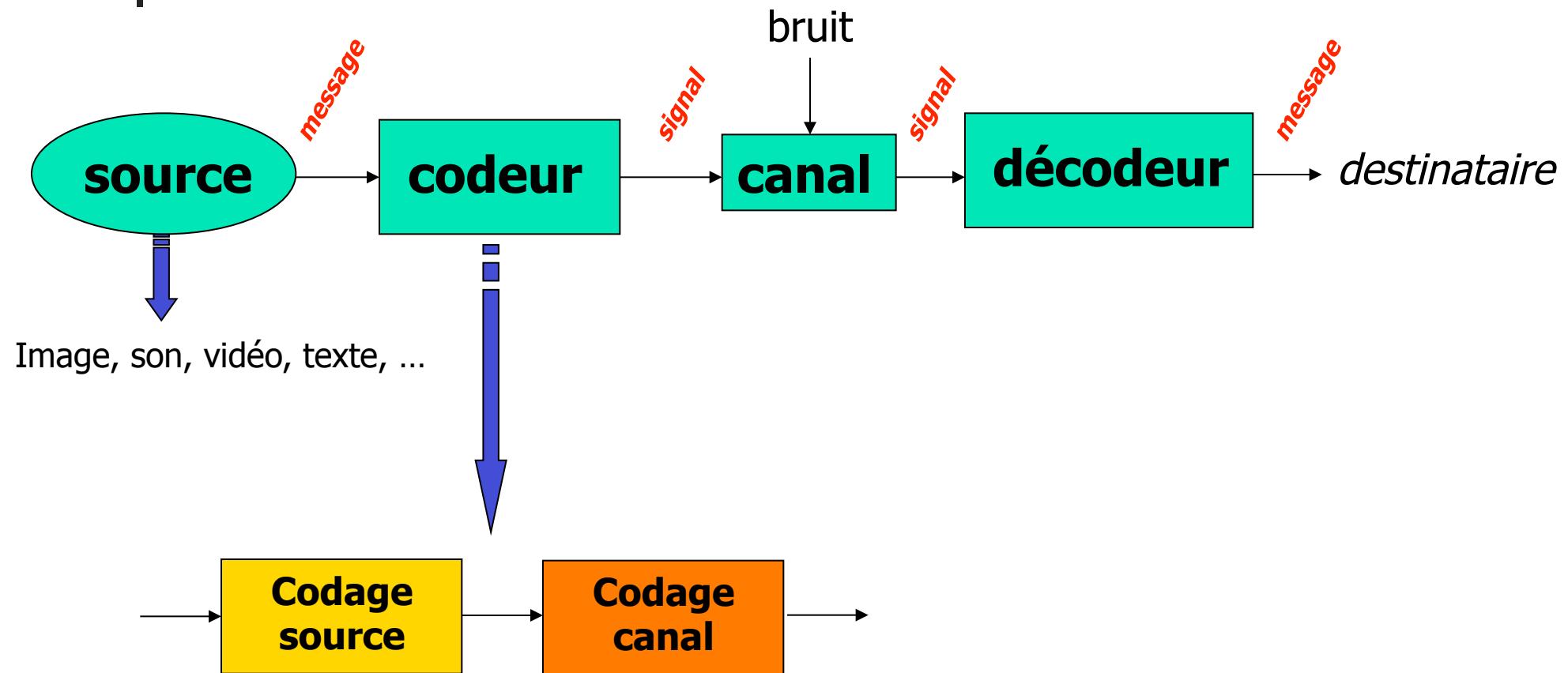
DVD : *Vidéo*

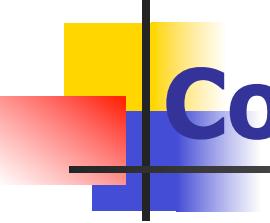


# Principaux outils et standards

- Vidéo : **MPEG 1, 2, 4 ...**
- Audio : **ADPCM (32 Kbits/s) ...**
- Image fixe : **GIF, JPEG, JPEG2000, JBIG, Facsimile ...**
- Téléconférence : **H.261, H.263, H.264, H.265, ...**

# Modèle de communication de Shannon

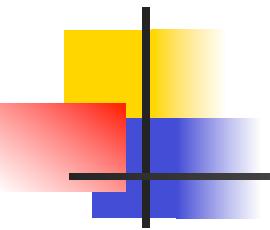




# Compression sans perte vs avec perte

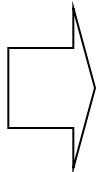
## En fonction de l'application :

- Compression **sans perte** (l'intégrité des données est préservée)  
*imagerie médicale, fichiers texte, ...*
- Compression **avec perte** (les données sont dégradées)  
*images grand public, vidéo, ...*



# Applications et Contraintes

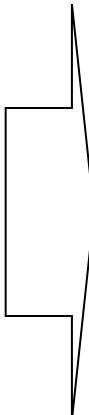
## « Temps réel »



*Téléphone, vidéo*

COMPRESSION / DECOMPRESSION RAPIDES

## « Temps différé »

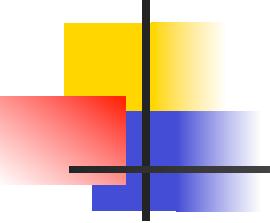


*Stockage sur disque (CD, CD ROM, DVD...)*

COMPRESSION LENTE / DECOMPRESSION RAPIDE

*Imagerie satellitaire ou embarquée*

COMPRESSION RAPIDE / DECOMPRESSION LENTE



# Applications et Contraintes

## Médical

pas d'artefact (erreur de diagnostic)

## Militaire

- conservation des détails (déttection de cibles)
- aspect mouvement (suivi de mobiles)

## Vidéo

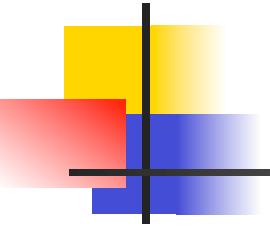
### « grand public »

effet de masquage de l'œil  
(espace et temps)

## Vision

### par ordinateur

Détection des contours  
(guidage d'un robot...)



# Performances

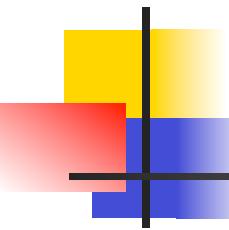
Les performances d'un système de compression **avec perte** sont :

- **le taux de compression :**  
(débit initial / débit après compression)
- **la qualité du signal comprimé :**
  - > critère subjectif (visuel)
  - > critères objectifs (SNR...)
- **la complexité du système**  
(coût calcul, mémoire requise)

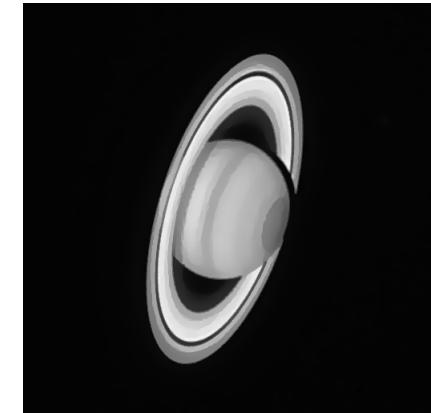
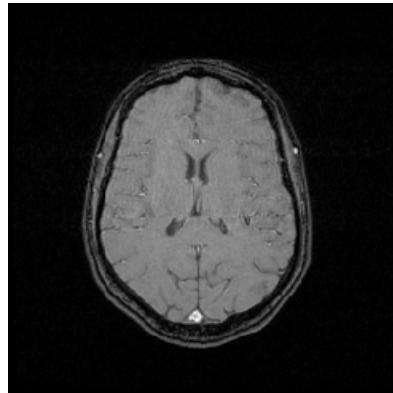
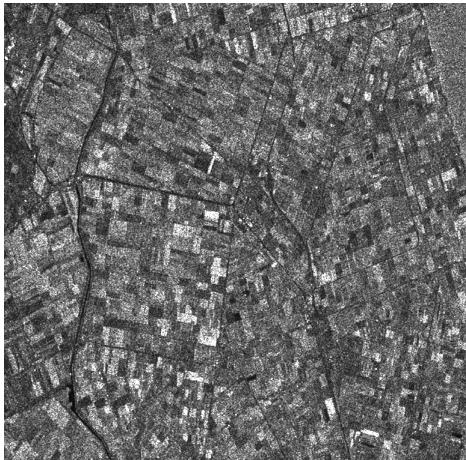


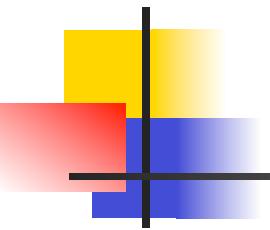
## **PROBLEME :**

Optimiser ces 3 facteurs en même temps !...

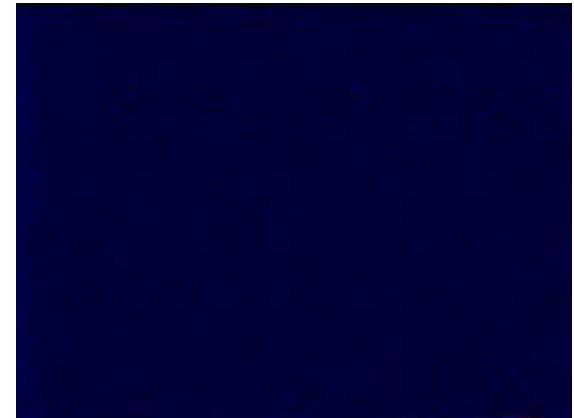
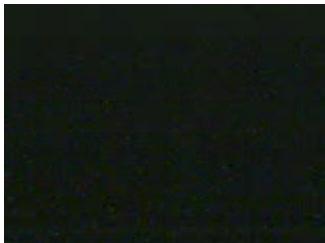


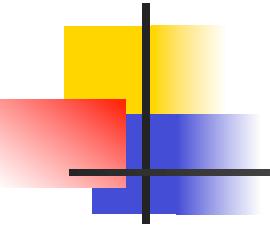
# Compresser : facile ou difficile ?





# Compresser : facile ou difficile ?

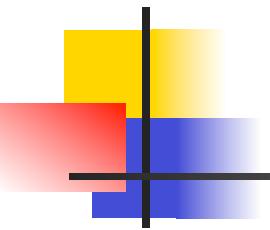




# Compresser : facile ou difficile ?

**Difficile car données numériques de natures très différentes**

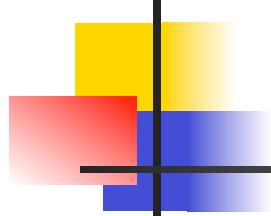
Performances de compression liées au contenu informatif de ces données et à la sensibilité de l'application



# Plan du Cours

---

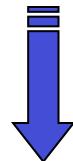
1. Introduction
2. Outils mathématiques de base
3. Stratégie de compression
4. Transformée
5. Quantification
6. Codage entropique
7. Compression d'images fixes : de JPEG à JPEG2000
8. Compression de vidéos : de MPEG I à MPEG IV
9. Transmission de documents confidentiels et sécurité



# Notion de source (définition)

**Source d'information**  $\Leftrightarrow$  **Système (pouvant prendre plusieurs états)**

Source

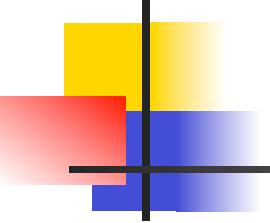


**langage (ou alphabet)**

Représenter la suite des états sous une forme particulière : message

Alphabet source : ensemble de symboles caractérisant les états du système

*Exemples :*  $S = \{a, b, c, d\}, S = \{0, 5\}, \dots$



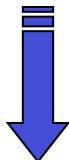
# Notion de source (définition)

source discrète : alphabet discret et généralement fini (cas du numérique)

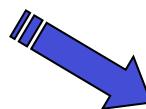
source sans mémoire : les états sont indépendants

source avec mémoire : les états sont dépendants entre eux

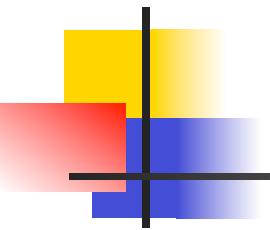
Caractériser et compter les états (loi de probabilité de la source)



Mesurer la quantité d'information produite par la source



Codage en vue d'une transmission ou d'un stockage efficaces



# Source : cas du son numérique

Son = signal 1D échantillonné + numérisé

↓  
représentation sur un nombre fini de niveaux (= quantification)

- **parole** transmise en bande téléphonique : fréquence d'échantillonnage = 8 KHz

Sur le réseau téléphonique public :

- norme UIT-T G.711 (1972) : codage à 64 Kbits/s (chaque échantillon étant codé sur 8bits)
- évolution vers 32 Kbits/s, puis 16 Kbits/s et même 8 Kbits/s !
- NB : GSM de l'ordre de 13 Kbits/s en Europe

- **musique** : fréquence d'échantillonnage généralement = 44,1 KHz

Sur un CD-Rom :

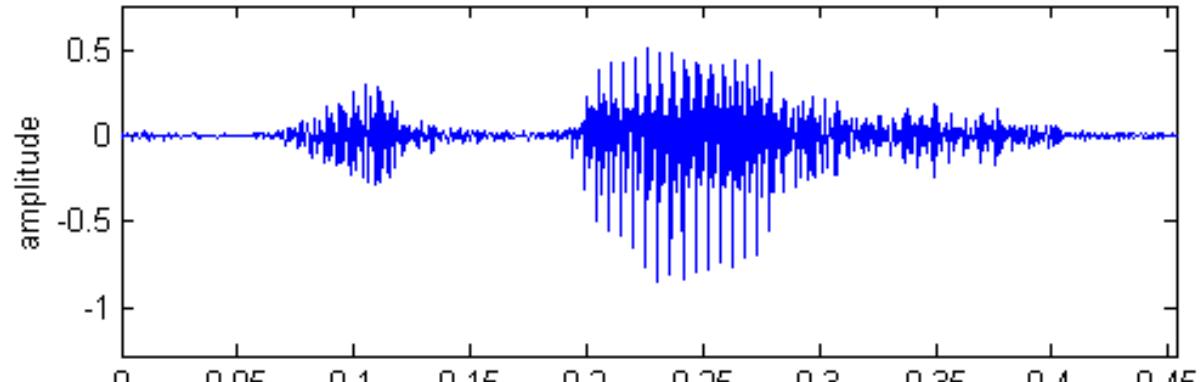
- 700 Kbits/s par voie (chaque échantillon étant codé sur 16 bits), soit 1,4 Mbits/s (stéréo)

# Source : cas du son numérique

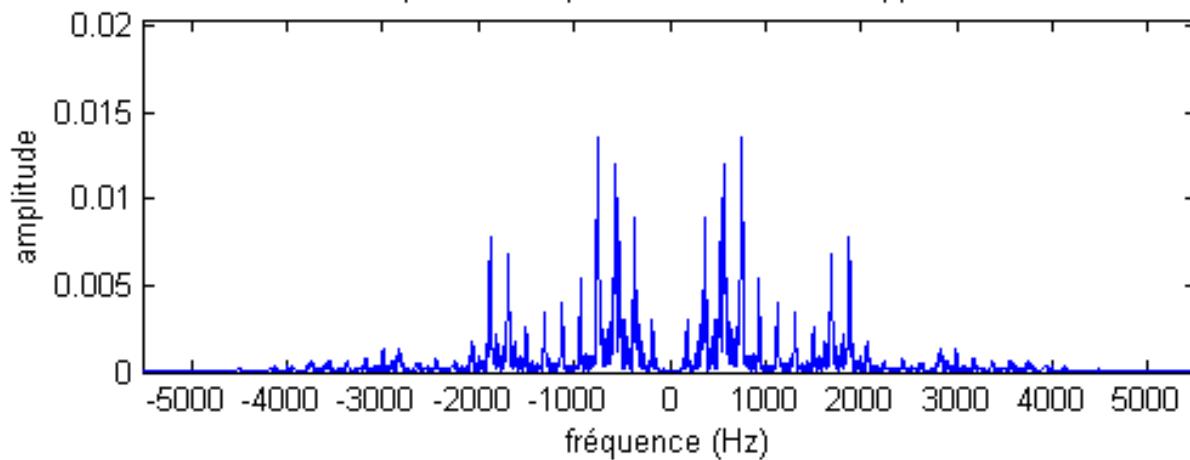


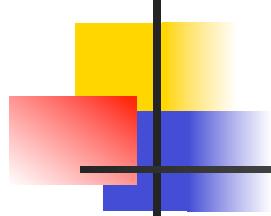
« developped »

signal temporel du mot : developped - fréquence d'échantillonnage 11025 Hz - codage 64 bits



spectre d'amplitude du mot : developped



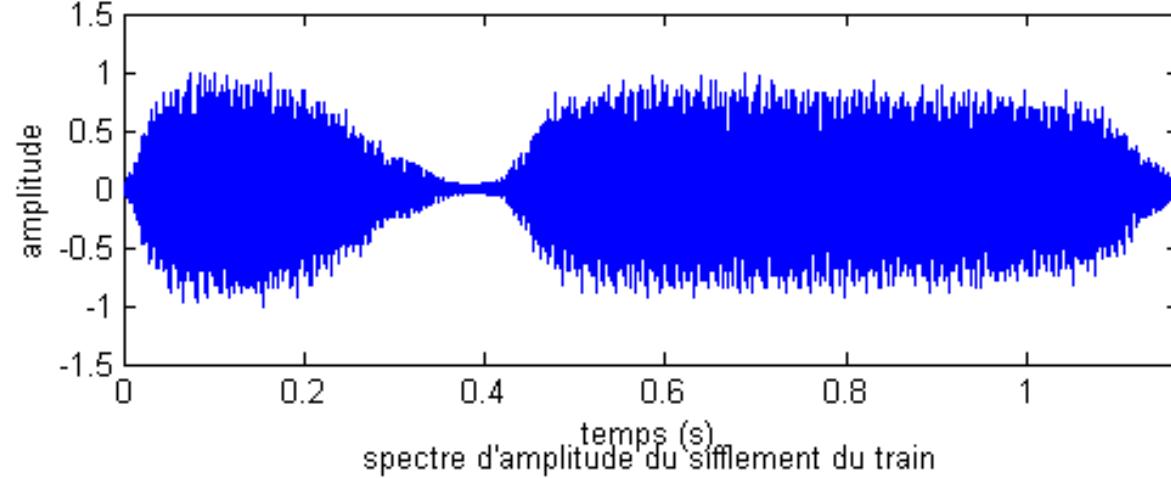


# Source : cas du son numérique

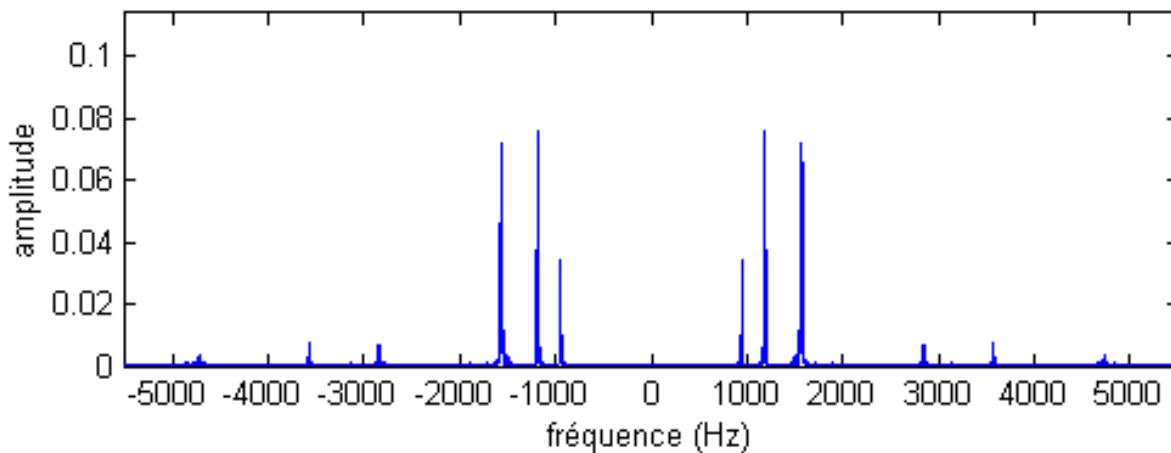


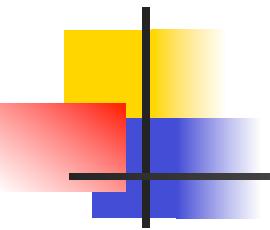
« train »

signal temporel du sifflement du train - fréquence d'échantillonnage 8 kHz - codage 16 bits



spectre d'amplitude du sifflement du train





# Source : cas de l'image numérique

Image échantillonnée + numérisée



représentation sur un nombre fini de niveaux (= quantification)

1 échantillon = 1 pixel (picture element)

## Exemples :

- image 256 niveaux de gris

dynamique de 0 (noir) à 255 (blanc) : alphabet  $S = \{0,1,2,\dots,255\}$

chaque niveau est représenté par 8 éléments binaires (0 ou 1)

Code binaire :  $C = \{00000000,00000001,\dots,11111111\}$



**--> 8 bits/pixel**

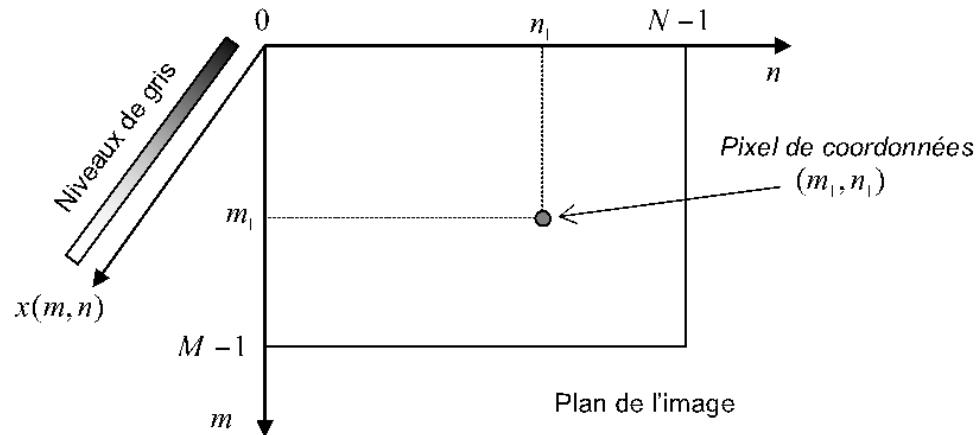
- Cas général  $b$  niveaux de gris

dynamique de 0 (noir) à  $2^b - 1$  (blanc)  $S = \{0,1,\dots,2^b - 1\}$

chaque niveau est représenté par  $b$  éléments binaires (0 ou 1)

# Source : cas de l'image numérique

Image numérique à niveaux de gris de taille  $M \times N$  pixels = matrice  $M \times N$



Intensité lumineuse (niveau de gris)  
du pixel de coordonnées (1,5)

$M$  lignes

$N$  colonnes

234	0	12	25	65	82
126	10	56	8	78	34
35	69	16	5	45	90
87	53	21	78	2	0
90	1	76	54	43	9



# Source : cas de l'image numérique

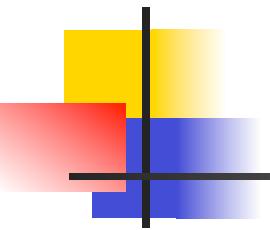
Image numérique **couleur** de taille  $M \times N$  pixels = 3 matrices  $M \times N$

		$N$ colonnes					
$M$ lignes		CANAL R					
		234	0	12	25	65	82
126	10	56	8	78	34		
35	69	16	5	45	90		
87	53	21	78	2	0		
90	1	76	54	43	9		

		$N$ colonnes					
$M$ lignes		chrominances du pixel de coordonnées (1,5)					
		3	0	12	25	0	82
12	10	65	8	87	18		
35	69	16	5	45	90		
87	0	21	8	2	0		
9	1	7	21	43	97		

		$N$ colonnes					
$M$ lignes		CANAL B					
		17	0	12	25	55	2
12	10	56	8	78	4		
35	69	61	5	45	95		
0	0	21	78	2	0		
90	1	76	5	79	214		



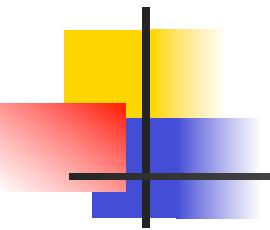


# Source : cas de l'image numérique

*Une image* = réalisation d'un **processus aléatoire bidimensionnel**  $I(m, n)$   
 $(m, n)$  sont les coordonnées spatiales d'un pixel

*L'intensité d'un pixel* = **variable aléatoire**  $X(m, n)$  qui prend  
ses valeurs  $x(m, n)$  dans l'intervalle  $[0, 2^b[$

**Hypothèses simplificatrices** : stationnarité et ergodicité de  $I(m, n)$



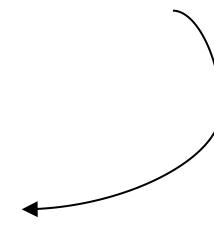
# Source : cas de l'image numérique

**Sous les hypothèses précédentes, possibilité de calculer :**

- probabilité d'apparition du niveau  $x$

$$p_X(x) = \text{probabilité}\{X = x\} = \frac{\text{nombre de pixels égaux à } x}{\text{nombre total de pixels dans l'image}}$$

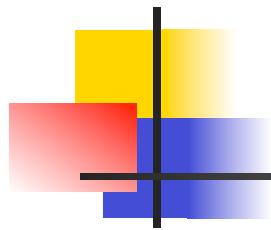
- moyenne empirique

$$\mu = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n)}{M.N}$$


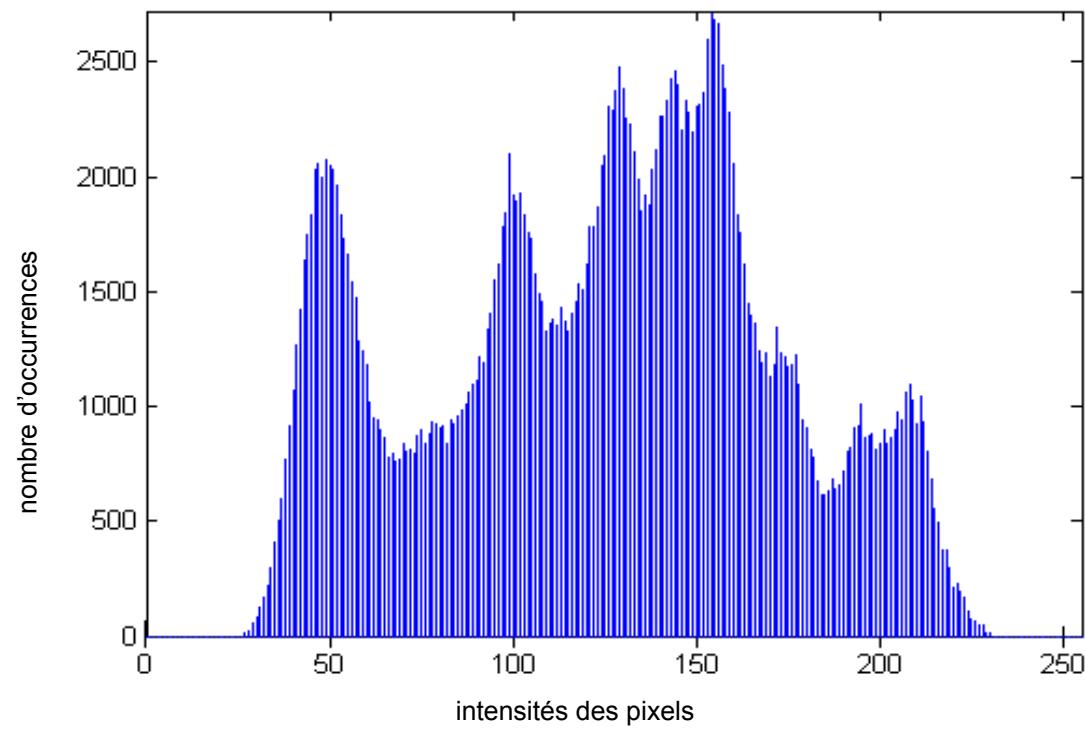
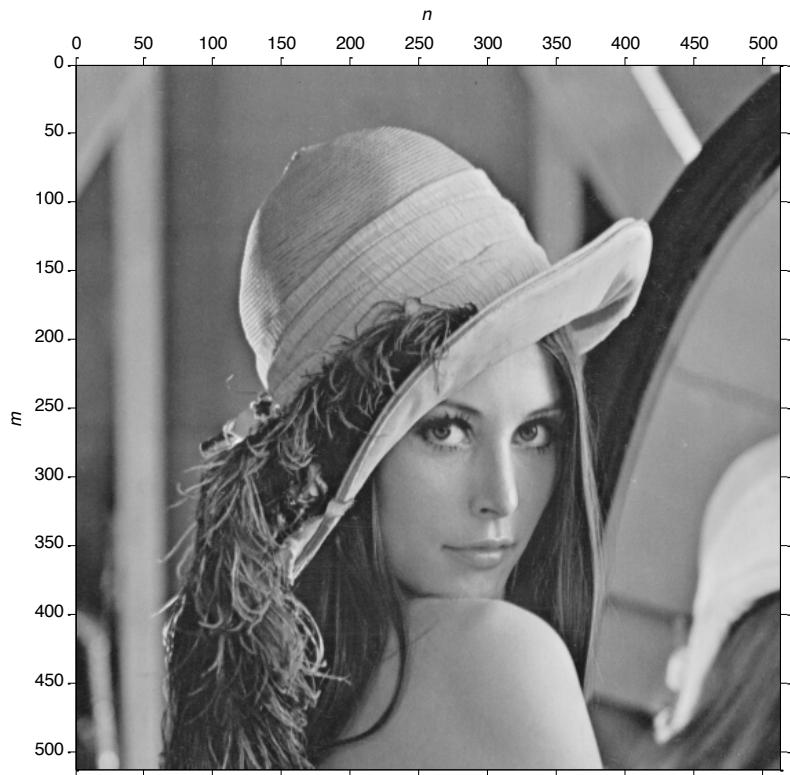
- variance empirique

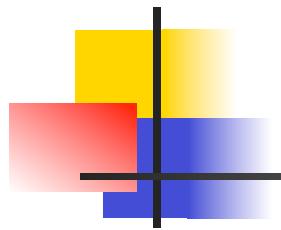
$$\sigma^2 = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [x(m, n) - \mu]^2}{M.N}$$

- ...



# Histogramme





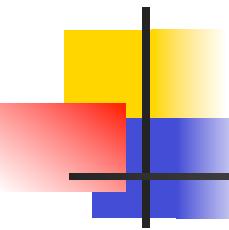
# Distribution de la source

Quelques lois utiles pour modéliser une distribution

$$f_X(x) = \frac{1}{b-a} \quad \text{avec } x \in [a, b] \quad \text{uniforme}$$

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \quad \text{gaussienne}$$

$$f_X(x) = \frac{1}{\sqrt{2\sigma}} e^{-\frac{\sqrt{2}}{\sigma}|x-\mu|} \quad \text{laplacienne}$$



# Entropie (rappel)

**Entropie (Shannon)** : quantité d'information moyenne minimale contenue dans une source

*Unité* : bits/échantillon (ou bits/pixel)

## Entropie d'ordre zéro :

Pour une source  $S$  indépendante prenant ses valeurs dans un ensemble de  $K$  symboles de probabilité d'apparition  $p_k \quad k \in \{1, \dots, K\}$

$$H(S) = - \sum_{k=1}^K p_k \log_2 p_k \quad \text{bits/pixel}$$

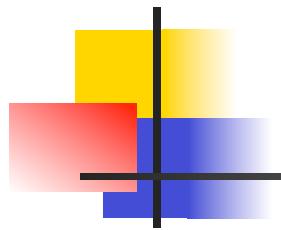
Exemple : image  $I$  codée sur 8 bits/pixel avec  $H(I)=6,5$  bits/pixel

## Entropie conjointe : les échantillons sont des groupes de pixels

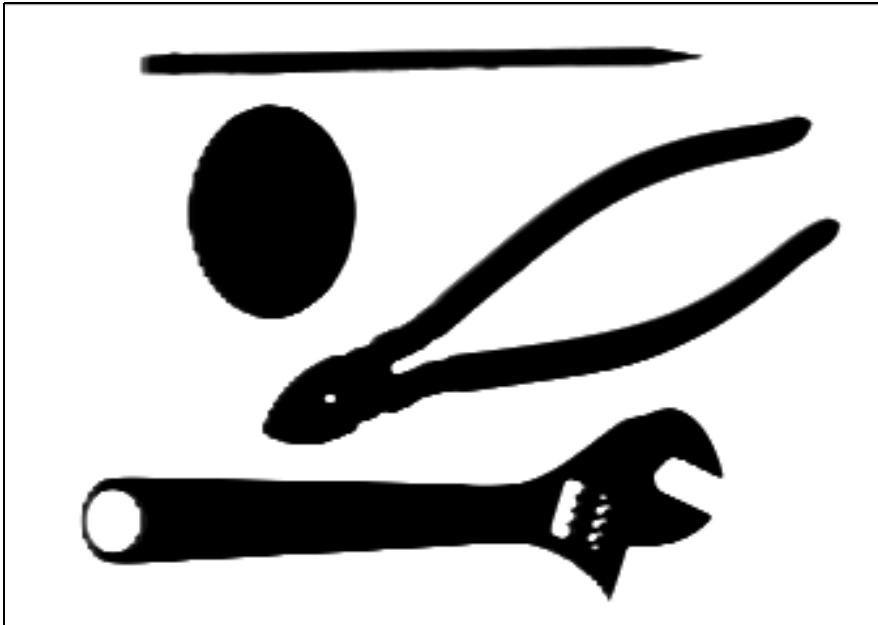
*Permet de prendre en compte la corrélation entre pixels*

## Entropie conditionnelle : les échantillons sont des pixels ou des groupes de pixels

*Permet de prendre en compte le passé*



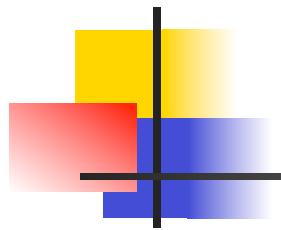
# Quelques entropies



$H=1,22$  bits/pixel

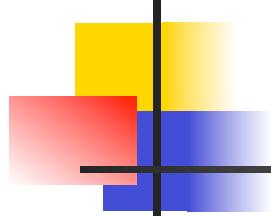


$H=7,4$  bits/pixel

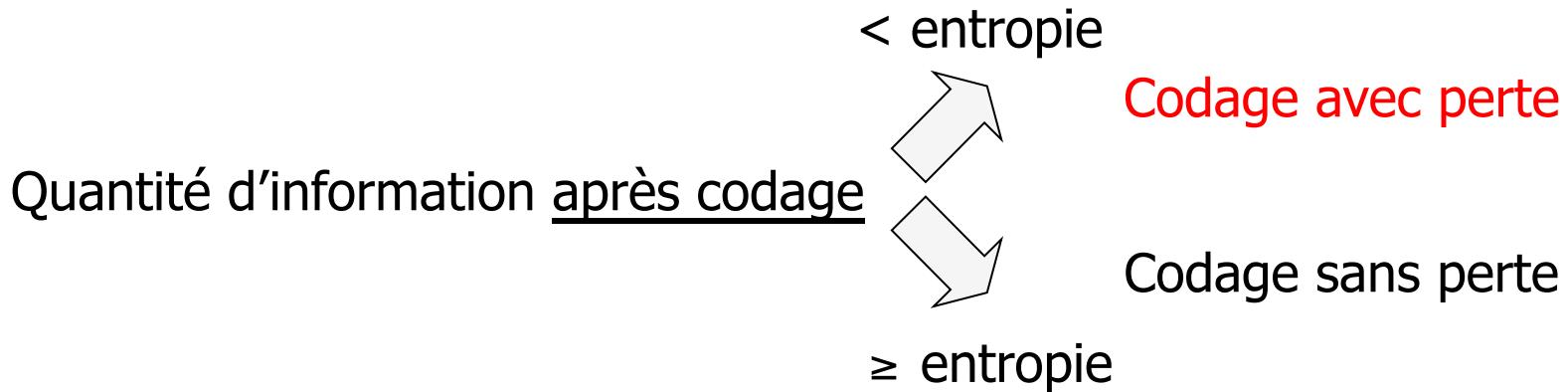


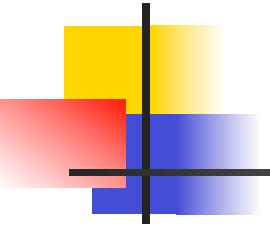
# Histogramme et entropie

exercice



# Codage sans perte vs avec perte





# Notion de qualité

**Qualité :** Subjective ou objective ?

*Distorsion moyenne :* 
$$D = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (x(m, n) - \hat{x}(m, n))^2$$

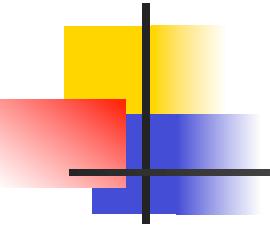
*Rapport Signal / Bruit :*  
*(Signal to Noise Ratio)*

$$SNR = 10 \log_{10} \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x^2(m, n)}{M.N.D} \text{ dB}$$

$x$  pixel original

$\hat{x}$  pixel compressé

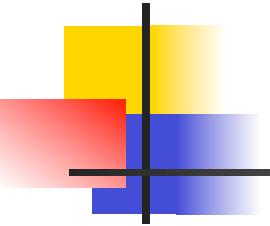
$$PSNR = 10 \log_{10} \frac{(2^b - 1)^2}{D} \text{ dB}$$



# Qualité visuelle et PSNR

Images « Lena » et « Cornouaille »





# Qualité visuelle et PSNR

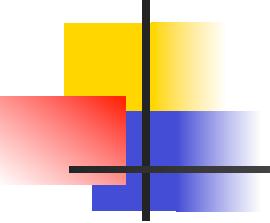
Images « Lena » et « Cornouaille » - Compression JPEG2000 - Taux de compression 64:1



PSNR = 30,82 dB



PSNR = 27,97 dB



# PSNR : les limites ...

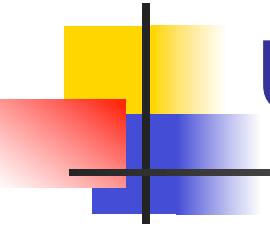


Bruit Gaussien dans une région de 300 pixels



Bruit Gaussien sur toute l'image

$\text{PSNR}=11,06 \text{ dB}$  pour chaque image !



# Une autre limite ...

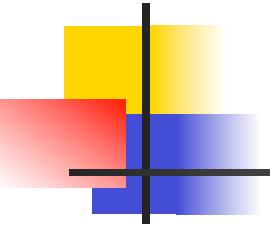


***Aucune mesure  
pour décrire la  
préservation du  
contenu !!!!!!!***



NMSE = 10.7 %

NMSE = 64 %



# Qualité visuelle et PSNR

Zoom sur l'image « lena » - Taux de compression 64:1

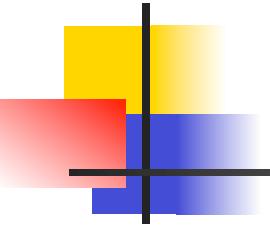


Original

Quant. Vect.  
(30,90 dB)

JPEG2000  
(30,82 dB)

SPIHT  
(31,08 dB)



# Perception humaine

Œil agit comme un filtre spatial passe-bas

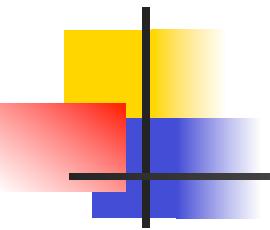
Œil sensible à l'intensité lumineuse

| bonne qualité d'image : PSNR > 30 dB

| œil sensible à des variations de l'ordre de 1 dB

Oreille entend des sons dans une gamme allant de 20 Hz à 20 kHz

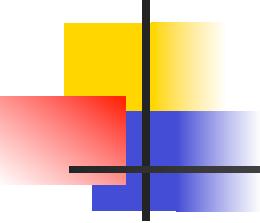
Oreille sensible à l'intensité des sons (dépend de la fréquence)



# Plan du Cours

---

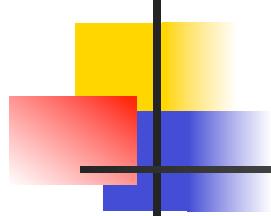
1. Introduction
2. Outils mathématiques de base
3. Stratégie de compression
4. Transformée
5. Quantification
6. Codage (cf cours CDCCE (TRS))
7. Compression d'images fixes : de JPEG à JPEG2000
8. Compression de vidéos : de MPEG I à MPEG IV
9. Transmission de documents confidentiels et sécurité



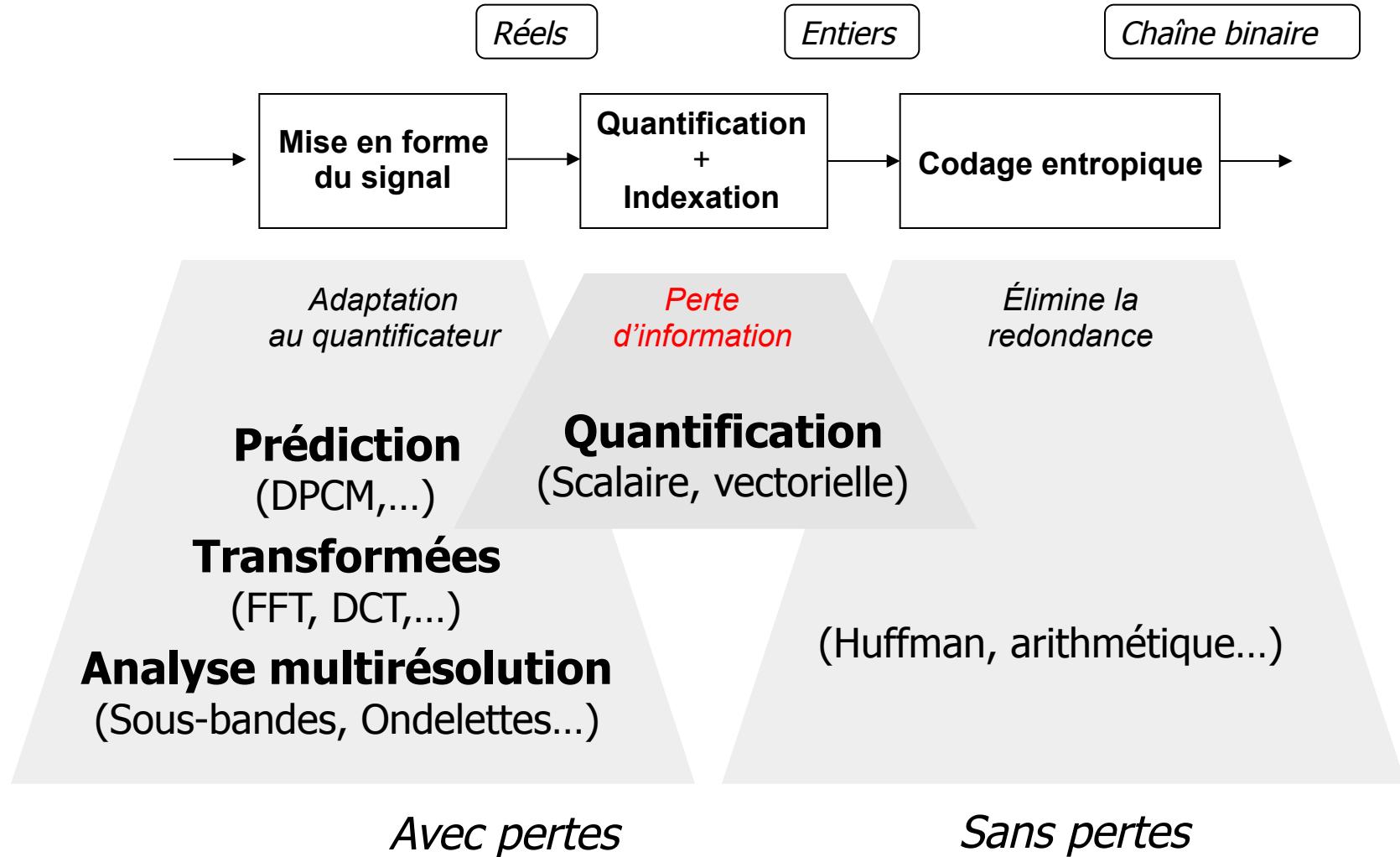
# Stratégie de compression

Diviser pour mieux régner !

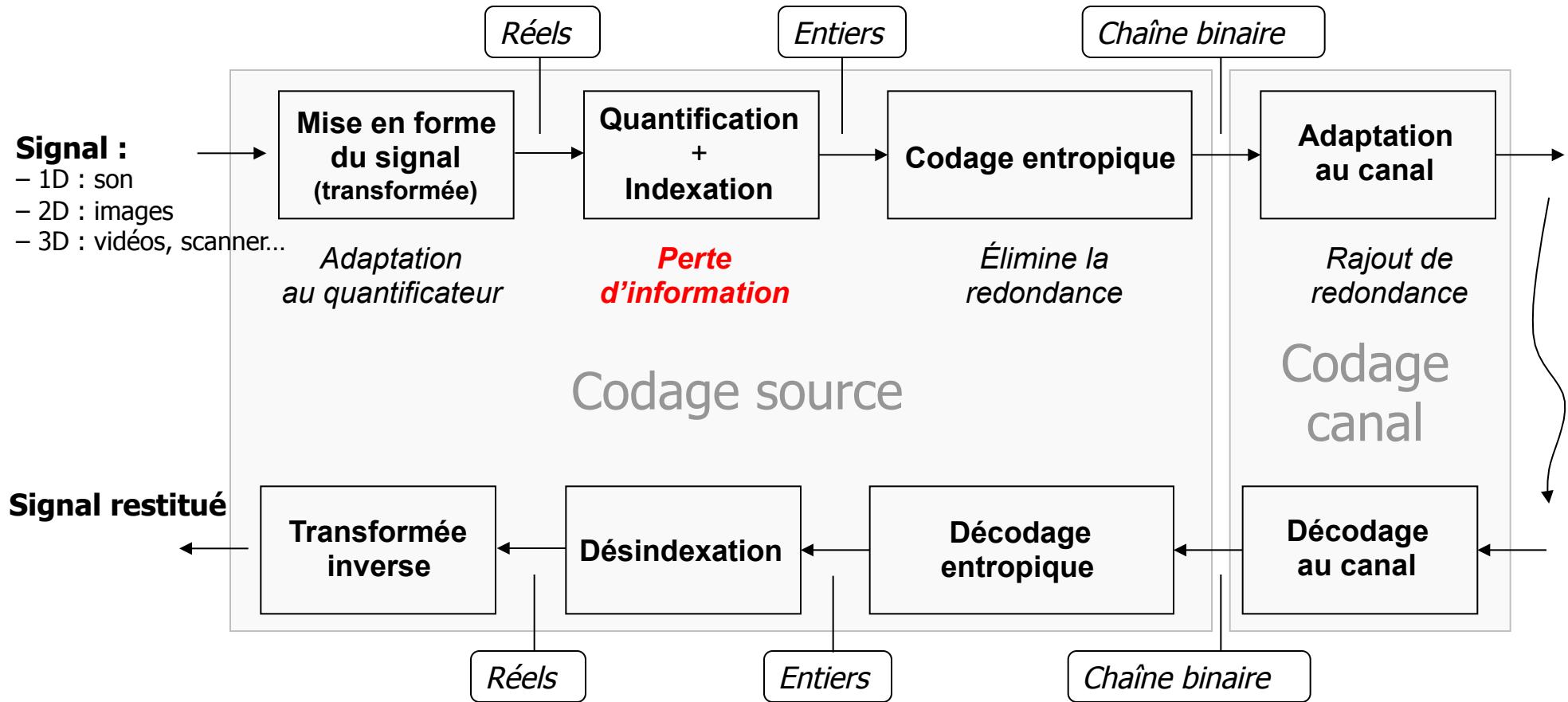


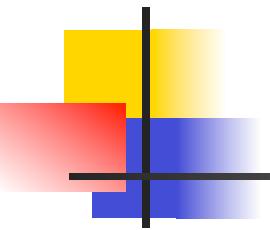


# La Chaîne De Compression



# La Chaîne De Compression

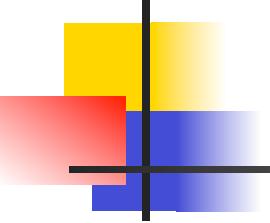




# Plan du Cours

---

1. Introduction
2. Outils mathématiques de base
3. Stratégie de compression
4. Transformée
5. Quantification
6. Codage entropique
7. Compression d'images fixes : de JPEG à JPEG2000
8. Compression de vidéos : de MPEG I à MPEG IV
9. Transmission de documents confidentiels et sécurité



# Changement d'espace

## Objectifs du changement d'espace de représentation :

- *Passer du domaine spatial au domaine fréquentiel*

- *Réorganiser l'information*

exemple : séparer les basses fréquences (zones homogènes)  
des hautes fréquences (contours nets).

- *Compackter l'énergie*

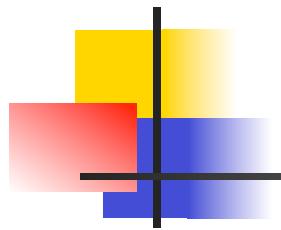
répartir l'énergie du signal d'origine sur peu de coefficients.

## Principales méthodes :

Transformée : *Karhunen Loeve, Hadamard, DCT, FFT,...*

Sous-bande : *bancs de filtres*

Analyse multirésolution : *ondelettes*



# Changement d'espace

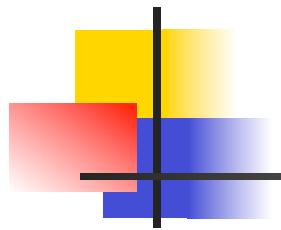
$I(m, n)$  réalisation d'un processus aléatoire       $T$  transformée (base orthonormale  $B$ )

*Coefficients de la transformée :*       $V = T \cdot I$

- *inversibilité* ( $T$  doit être bijective)     $I = T^{-1} \cdot V$

- *orthogonalité* :     $T^T T = T^{-1} T = Id$

- *unitarité* :             $\|V\|^2 = \|I\|^2$



# Changement d'espace

Exemple :  $I$  contenant 2 pixels  $I(0) = 205$      $I(1) = 206$     8 bits/pixel

transformée :  $T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$V$  ?

conservation d'énergie ?

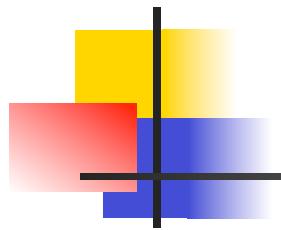
$V_q$  ?

nombre bits/pixel ?

$\hat{I}$  ?

$\|V - V_q\|^2$  ?

$\|I - \hat{I}\|^2$  ?



# Changement d'espace

$$V = T \cdot I = \begin{pmatrix} 290,6 \\ -0,7 \end{pmatrix}$$

*conservation d'énergie :*

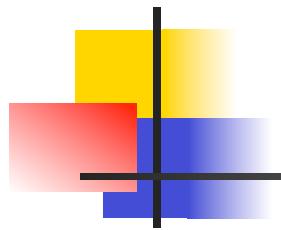
$$\|V\|^2 = \frac{1}{2} (411^2 + 1^2) = 84461$$
$$\|I\|^2 = (205^2 + 206^2) = 84461$$

$$V_q = \begin{pmatrix} 291 \\ 0 \end{pmatrix} \quad \textbf{9 bits au lieu de 16 au départ !!!}$$

$$\hat{I} = T^{-1} \cdot V_q = \begin{pmatrix} 205,8 \\ 205,8 \end{pmatrix}$$

$$\|V - V_q\|^2 = (-0,4)^2 + (0,7)^2 = 0,65$$

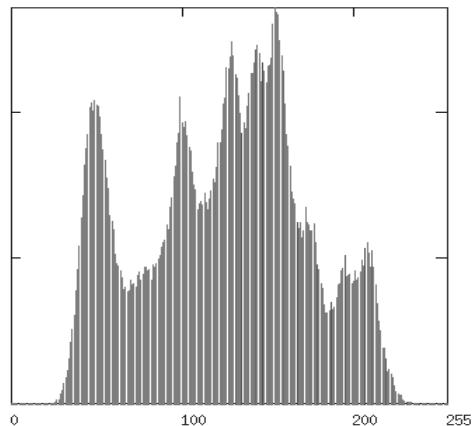
$$\|I - \hat{I}\|^2 = (-0,8)^2 + (0,2)^2 = 0,68$$



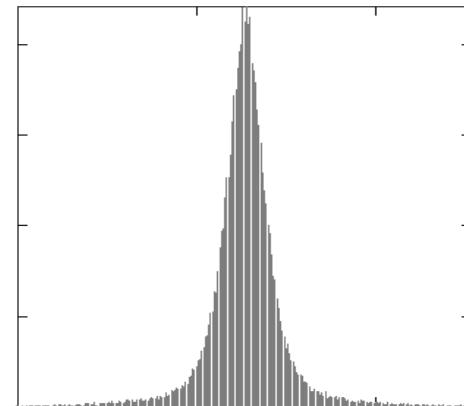
# Avantages

## Avantages du changement d'espace de représentation :

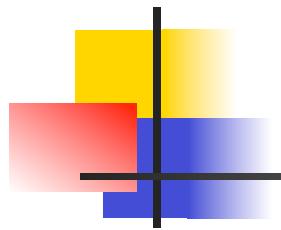
plus facile à coder



histogramme de l'image



histogramme typique des coefficients de la transformation

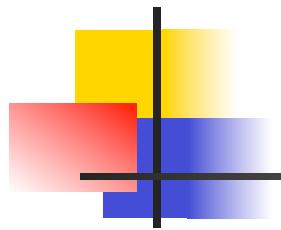


# Méthode par transformée : DCT

**DCT** : Discrete Cosine Transform (Transformée en cosinus discrète).

La transformée d'une fonction  $x(m,n)$  vaut pour le point  $(u,v)$  :

$$\left\{ \begin{array}{l} X(u,v) = \frac{2}{\sqrt{MN}} \cdot C(u) \cdot C(v) \cdot \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \cos\left[\frac{(2m+1)\pi}{2M} u\right] \cdot \cos\left[\frac{(2n+1)\pi}{2N} v\right] \cdot x(m,n) \\ \text{avec: } \begin{cases} (u,v) \in [0, M-1] \times [0, N-1] \\ C(0) = 1/\sqrt{2} \quad \text{et} \quad \forall \alpha \neq 0 \quad C(\alpha) = 1 \end{cases} \end{array} \right.$$



# Méthode par transformée : DCT

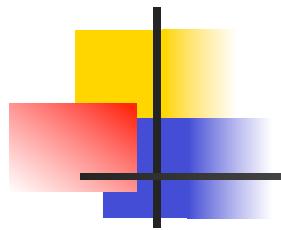
**DCT** : Discrete Cosine Transform (Transformée en cosinus discrète).

La transformée d'une fonction  $x(m,n)$  vaut pour le point  $(u,v)$  :

$$\left\{ \begin{array}{l} X(u,v) = \frac{2}{\sqrt{MN}} \cdot C(u) \cdot C(v) \cdot \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \cos\left[\frac{(2m+1)\pi}{2M} u\right] \cdot \cos\left[\frac{(2n+1)\pi}{2N} v\right] \cdot x(m,n) \\ \text{avec: } \begin{cases} (u,v) \in [0, M-1] \times [0, N-1] \\ C(0) = 1/\sqrt{2} \quad \text{et} \quad \forall \alpha \neq 0 \quad C(\alpha) = 1 \end{cases} \end{array} \right.$$

La transformée inverse d'une fonction  $X(u,v)$  vaut pour le point  $(m,n)$  :

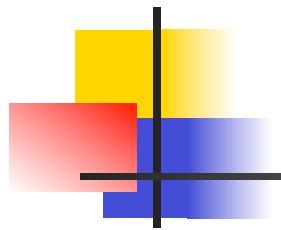
$$x(m,n) = \frac{2}{\sqrt{MN}} \cdot \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} C(u).C(v).\cos\left[\frac{(2m+1)\pi}{2M} u\right] \cdot \cos\left[\frac{(2n+1)\pi}{2N} v\right] \cdot X(u,v)$$



# Méthode par transformée : DCT

	<i>n</i>							
<i>m</i>	32	33	33	36	46	85	150	177
	31	31	38	48	95	132	179	196
	32	37	59	104	145	175	192	177
	48	61	119	159	186	182	163	138
	87	118	168	194	185	158	130	113
	130	170	193	177	154	125	116	115
	168	191	173	155	136	113	115	135
	185	156	139	125	120	121	140	174

Bloc 8x8 pixels extrait de l'image Lena (le coin en haut à gauche se situe à la position (184,280)).



# Méthode par transformée : DCT

	v							
u	999,75	-164,78	-22,84	-25,80	1,00	1,02	-3,53	7,57
	-191,35	-245,32	29,34	27,23	-9,53	13,64	-4,15	4,52
	-85,80	4,12	169,35	-11,50	6,10	-5,26	-3,90	-1,30
	-14,02	94,81	-14,39	-46,94	-12,20	-5,34	-4,11	-1,48
	-9,50	-3,79	-16,15	1,82	11,25	11,67	8,04	-0,18
	-3,38	14,78	-8,48	-2,43	-2,68	-0,53	-4,21	-3,26
	-3,97	-5,86	4,60	-0,57	-2,26	7,33	-0,35	-1,88
	1,98	4,44	-2,82	1,43	2,39	0,93	-5,50	-7,21

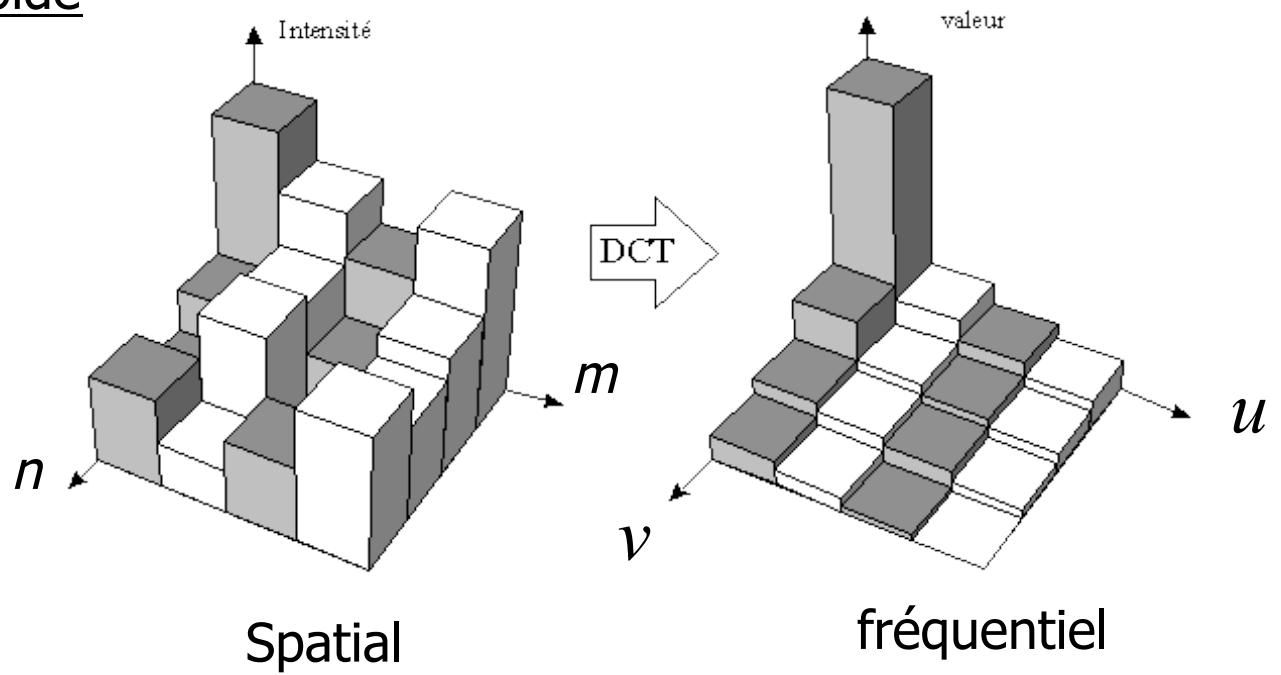
DCT du bloc 8x8 pixels précédent

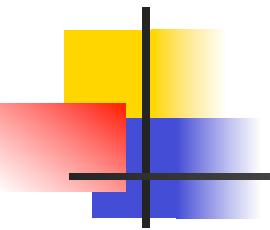
# Méthode par transformée : DCT

La DCT (Discrete Cosine Transform) est inadaptée aux signaux non-stationnaires

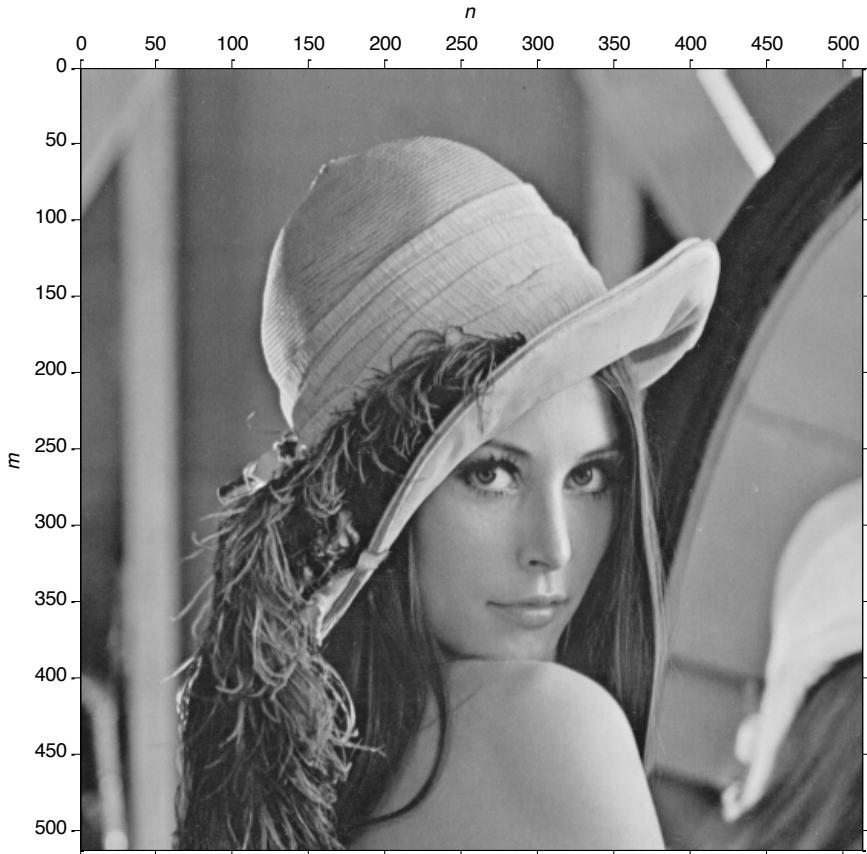
- ➡ Découpage de l'image en blocs 8x8 pixels (+ ou - stationnaires)
- ➡ Effets de blocs après quantification
- ➡ Algorithme rapide

**Ex. : Bloc 4x4 pixels**

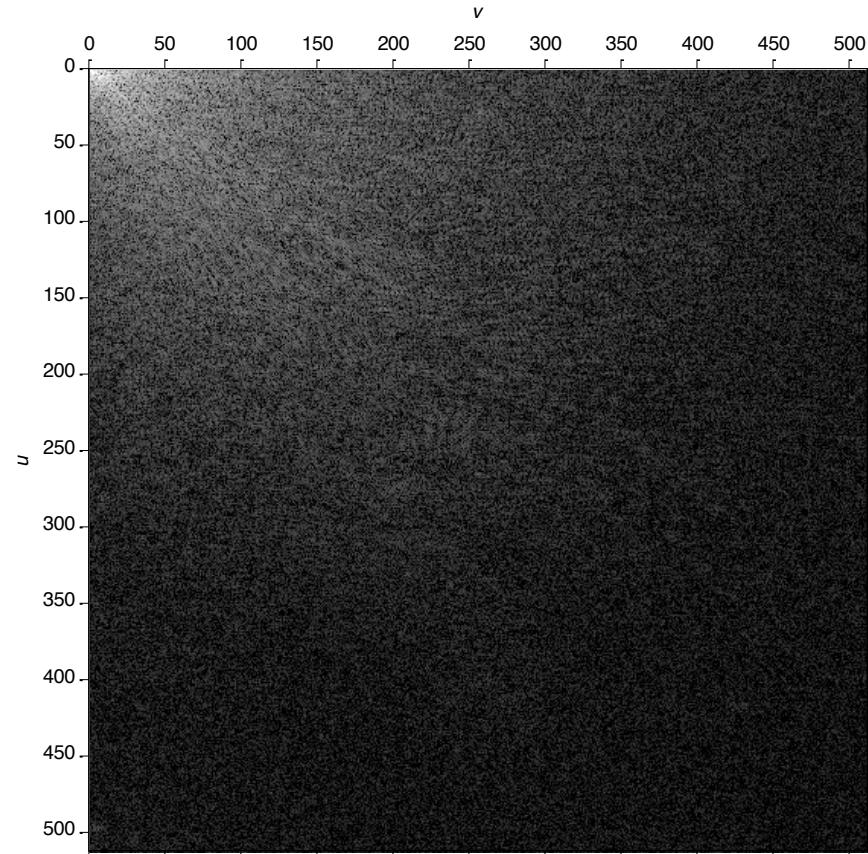




# Méthode par transformée : DCT



(a) : intensité des pixels  $x(m,n)$



(b) : valeurs des coefficients de la DCT  $X(u,v)$

(les niveaux sombres représentent les coefficients proches de zéro)

# Analyse Multirésolution: Ondelette

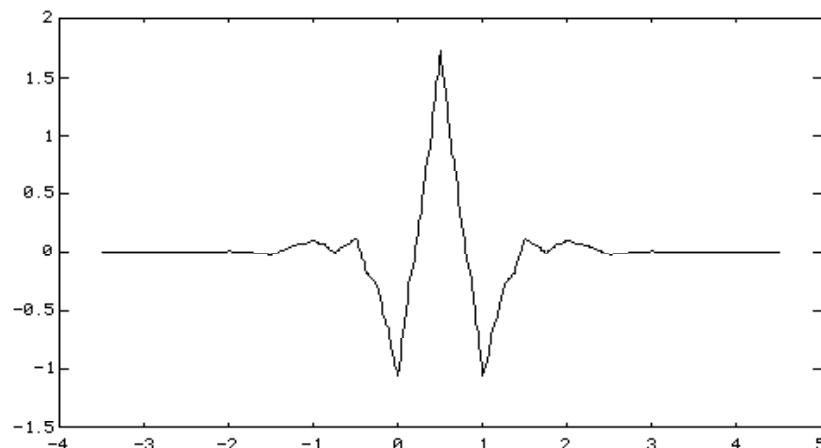
**Définition :**

- a facteur d'échelle
- b facteur de translation

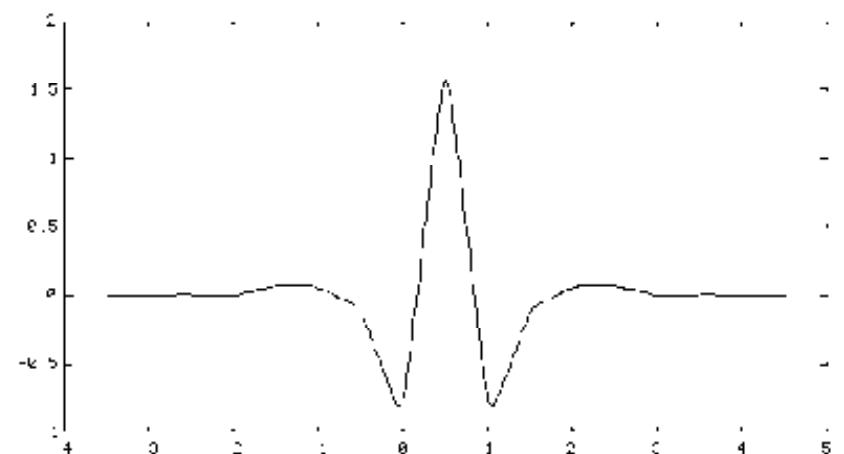
$$\psi_{a,b}(x) = |a|^{-1/2} \cdot \psi\left(\frac{x - b}{a}\right)$$

**Ondelettes orthogonales :**

$$\psi_{m,n}(x) = 2^{-m/2} \psi(2^{-m}x - n) \quad (m, n) \in \mathbb{Z}^2$$



exemples  
d'ondelettes

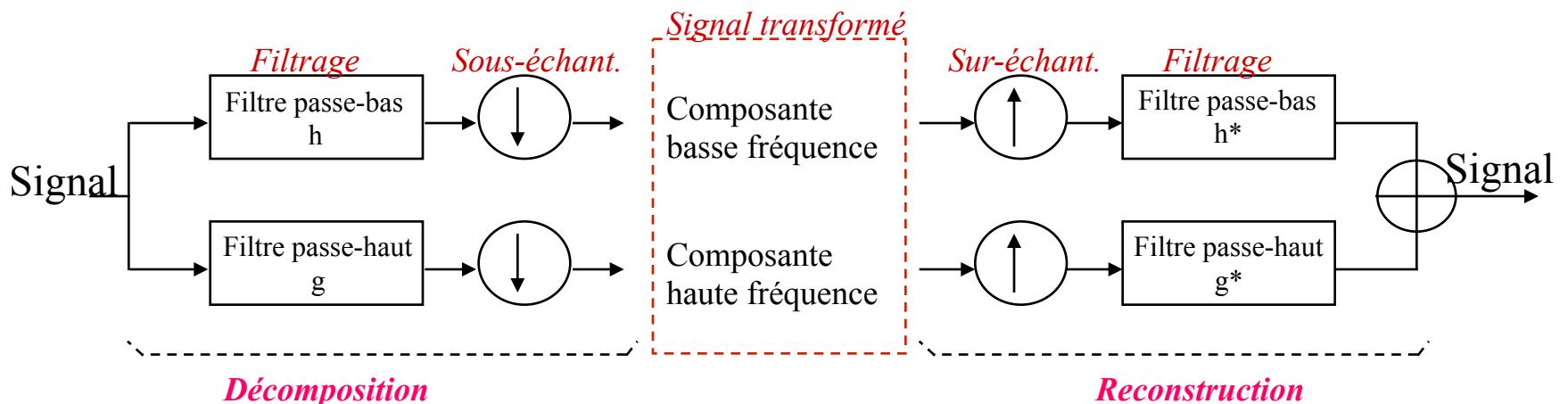


# Analyse Multirésolution: Ondelette

## Calcul des Coefficients :

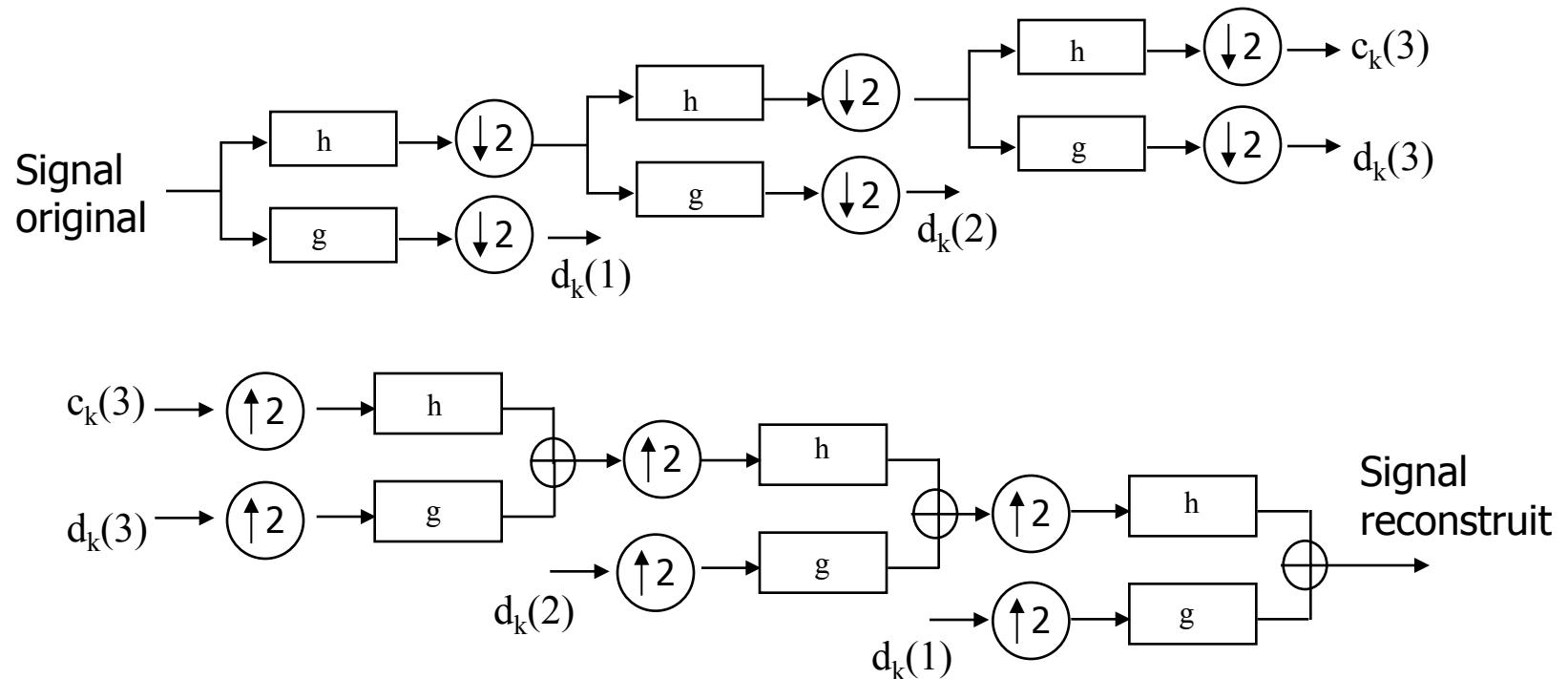
$$\left\{ \begin{array}{l} c_{m,n}(f) = \langle f, \psi_{m,n} \rangle = \int f(x) \overline{\psi}_{m,n}(x) dx \\ f(x) = \sum_{m,n} c_{m,n}(f) \psi_{m,n}(x) \end{array} \right.$$

## Mise en œuvre par filtrage :



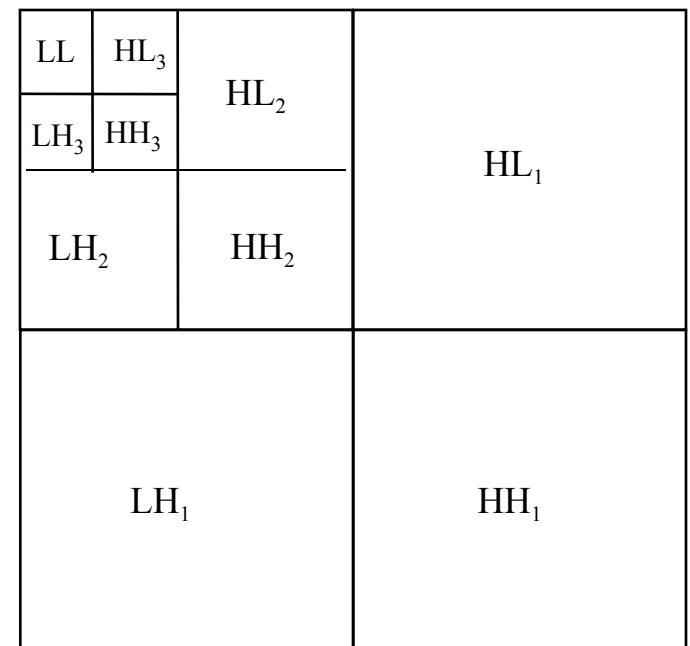
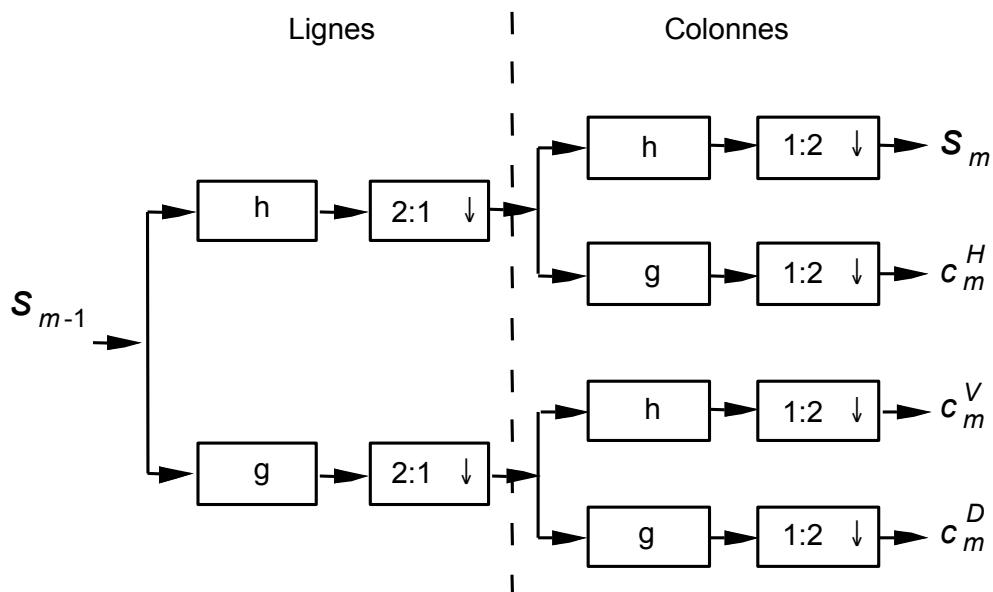
# Analyse Multirésolution: Ondelette

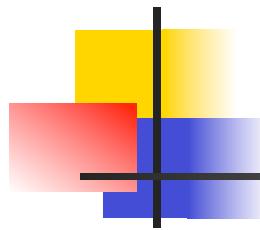
Décomposition multirésolution 1D :



# Analyse Multirésolution: Ondelette

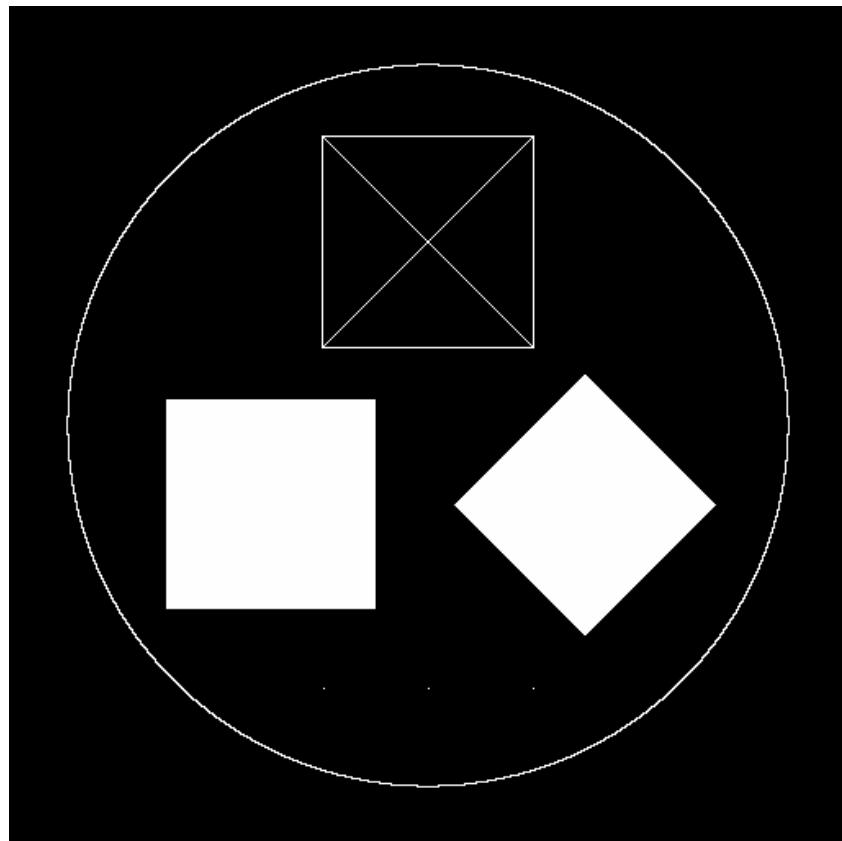
Décomposition multirésolution 2D ou séparable :



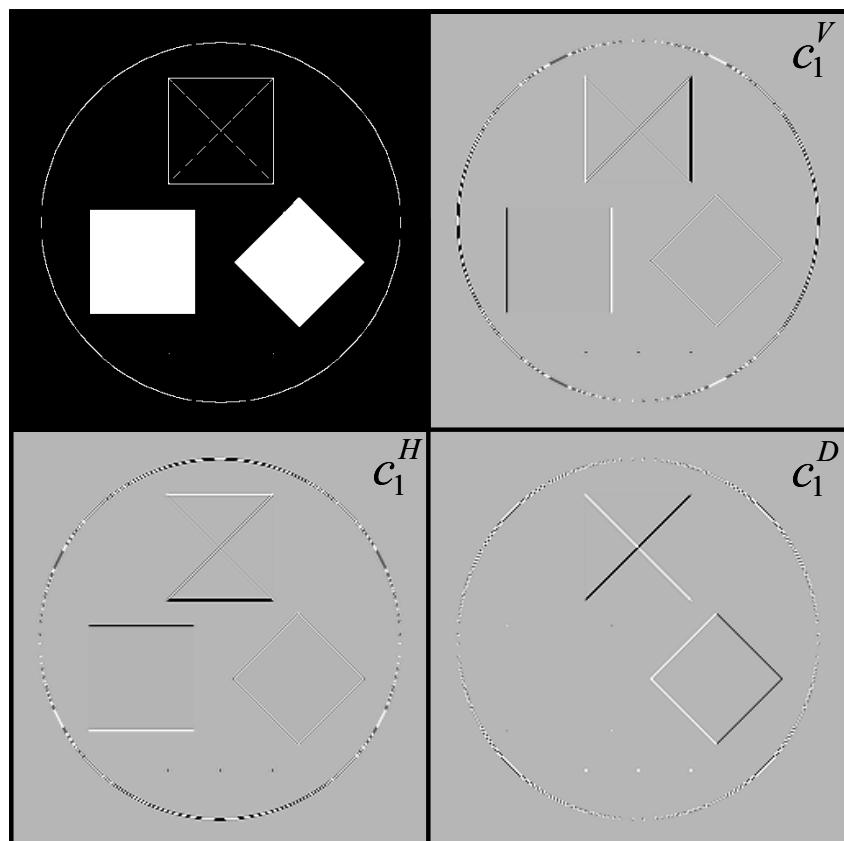


# Analyse Multirésolution: Ondelette

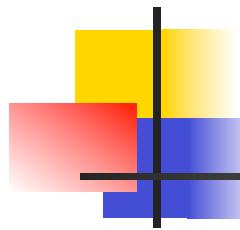
Décomposition multirésolution 2D ou séparable :



(a)

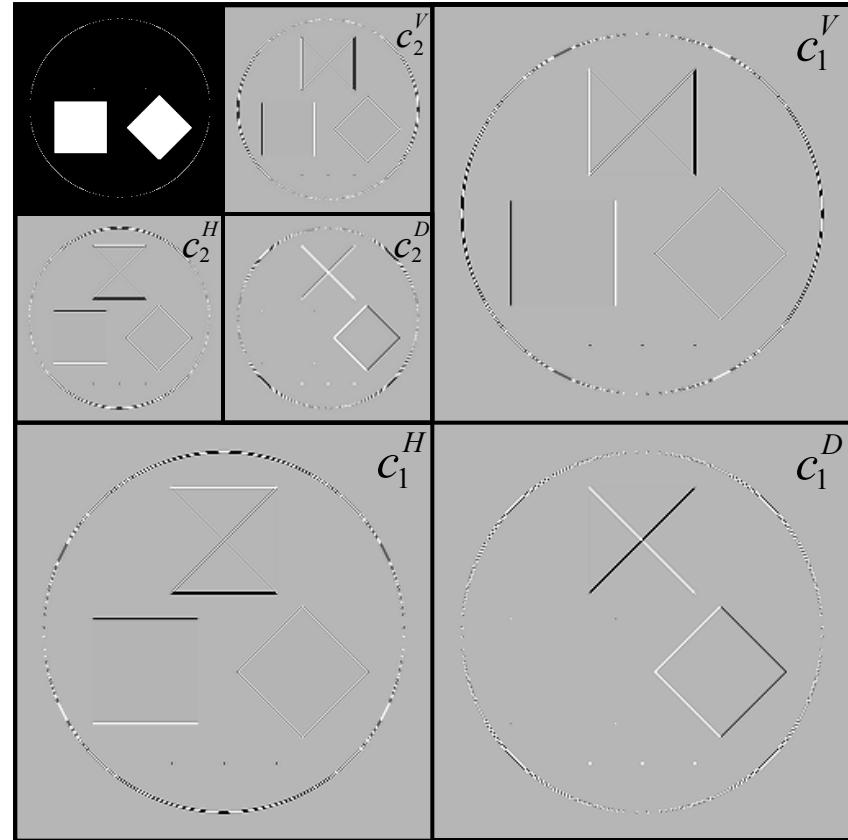
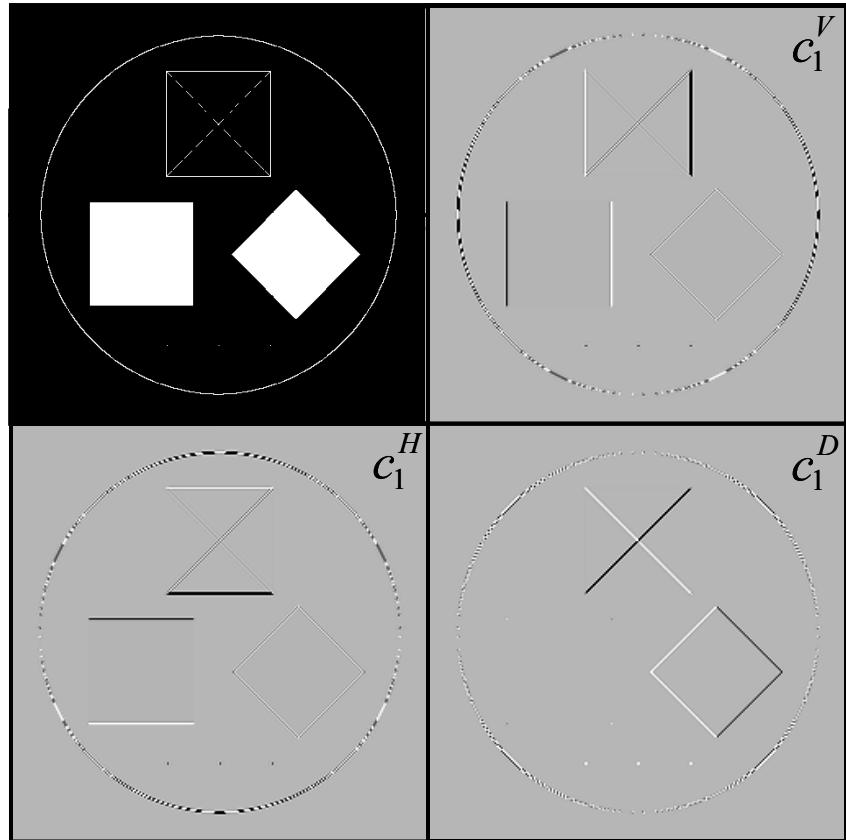


(b)



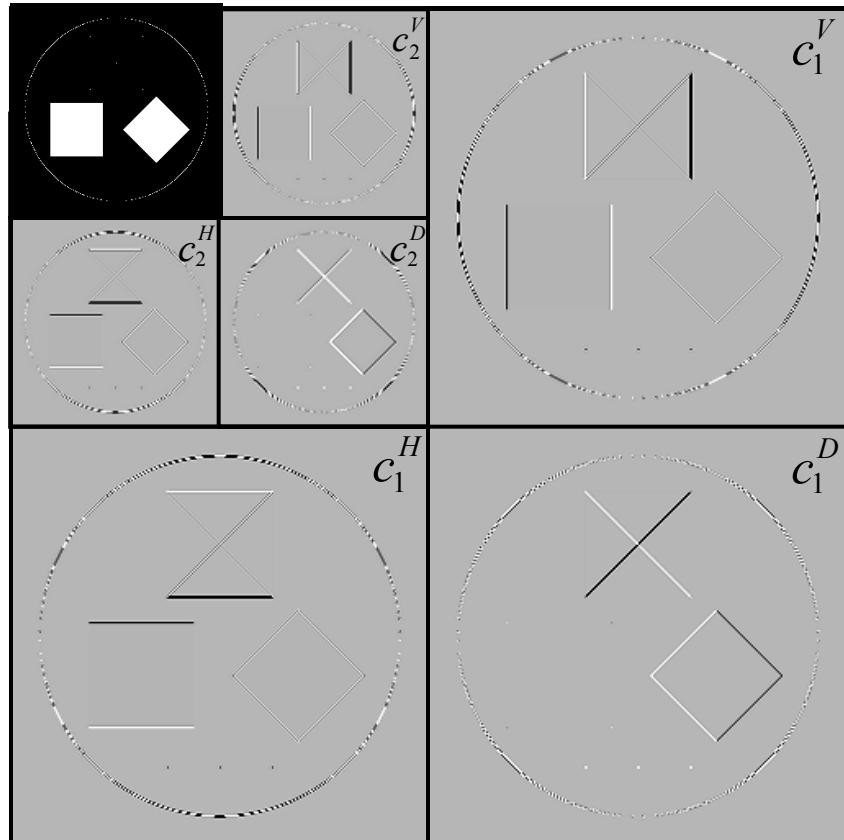
# Analyse Multirésolution: Ondelette

Décomposition multirésolution 2D ou séparable :

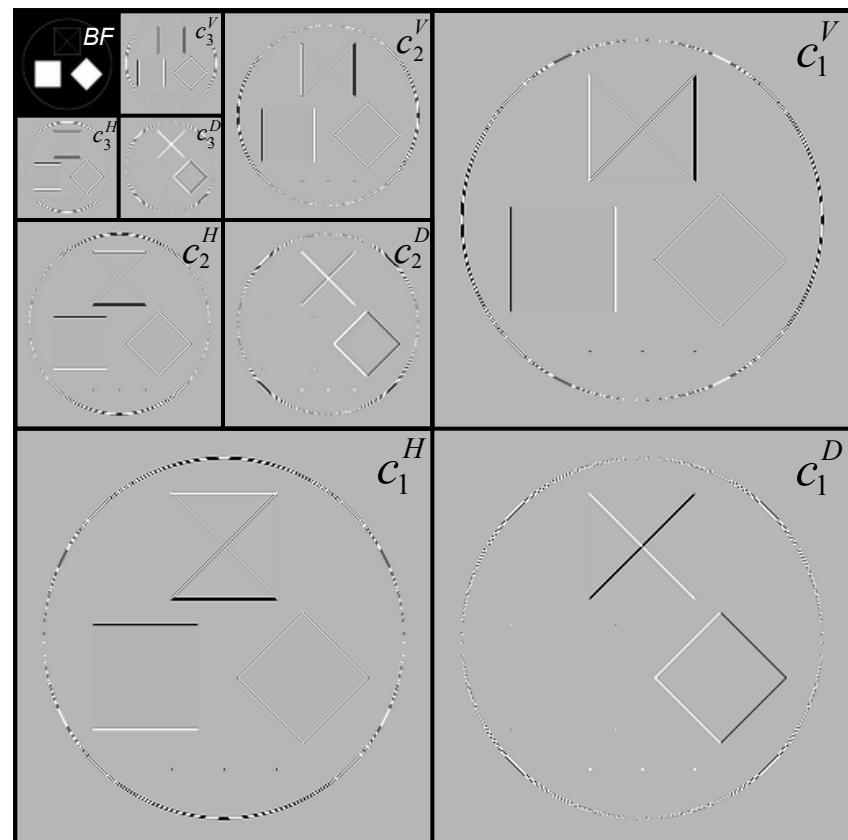


# Analyse Multirésolution: Ondelette

Décomposition multirésolution 2D ou séparable :



(e)

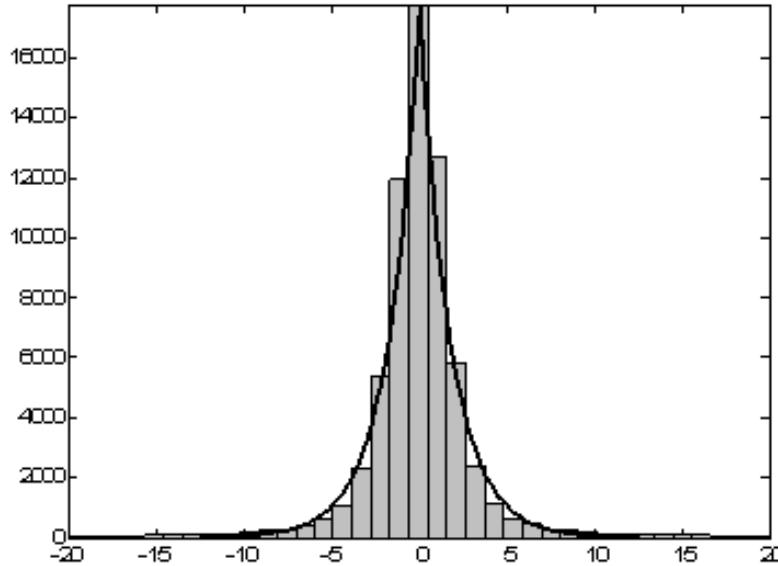


(f)

# Analyse Multirésolution: Ondelette



(a)



(b)

Gaussienne généralisée

**Modèle de loi des coefficients:**

$$a = \frac{b \cdot p}{2 \cdot \Gamma(1/p)} \quad b = \frac{1}{\sigma} \sqrt{\frac{\Gamma(3/p)}{\Gamma(1/p)}}$$

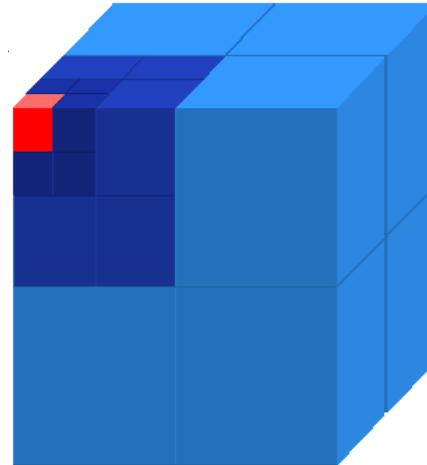
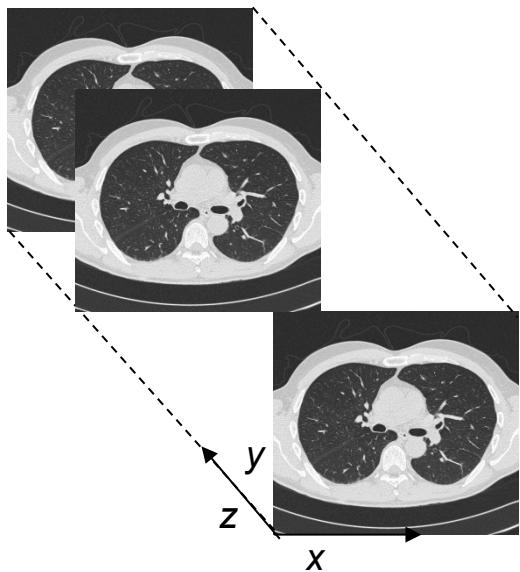
$$f_X(x) = a \cdot e^{-|b \cdot x|^p} \quad 0 < p \leq 2$$

$$\Gamma(n+1) = n! \quad \Gamma(1/2) = \sqrt{\pi}$$

$$\Gamma\left(m + \frac{1}{2}\right) = \frac{1 \cdot 3 \cdot 5 \dots (2m-1)}{2^m} \sqrt{\pi}, \quad m = 1, 2, 3, \dots$$

# Changement d'espace : Ondelettes

## Cas des images volumiques

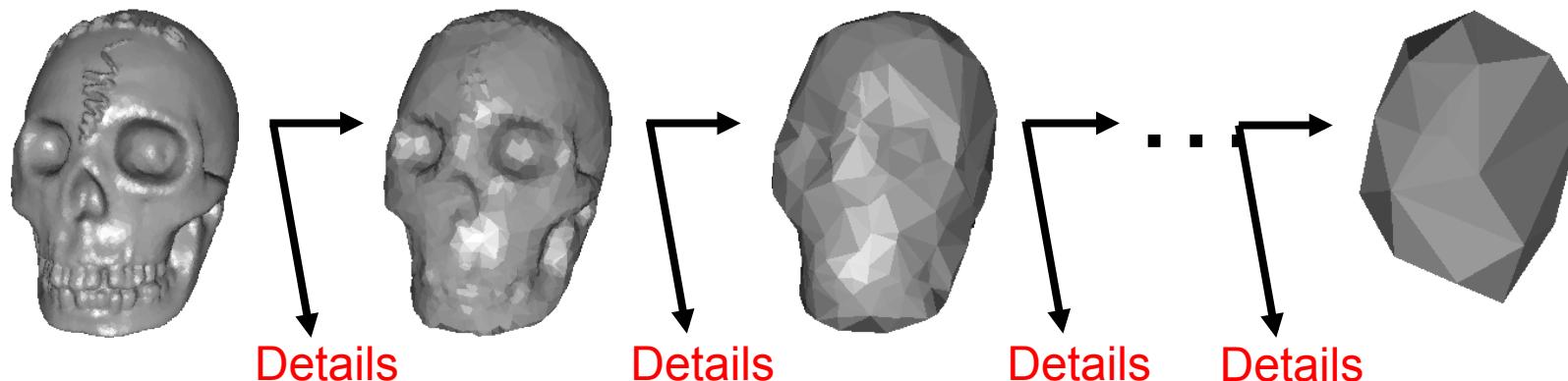


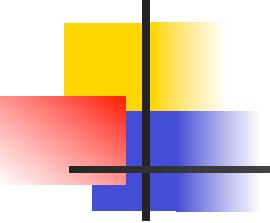
**Volume de coefficients d'ondelettes**

# Changement d'espace : Ondelettes

## Cas des objets 3D

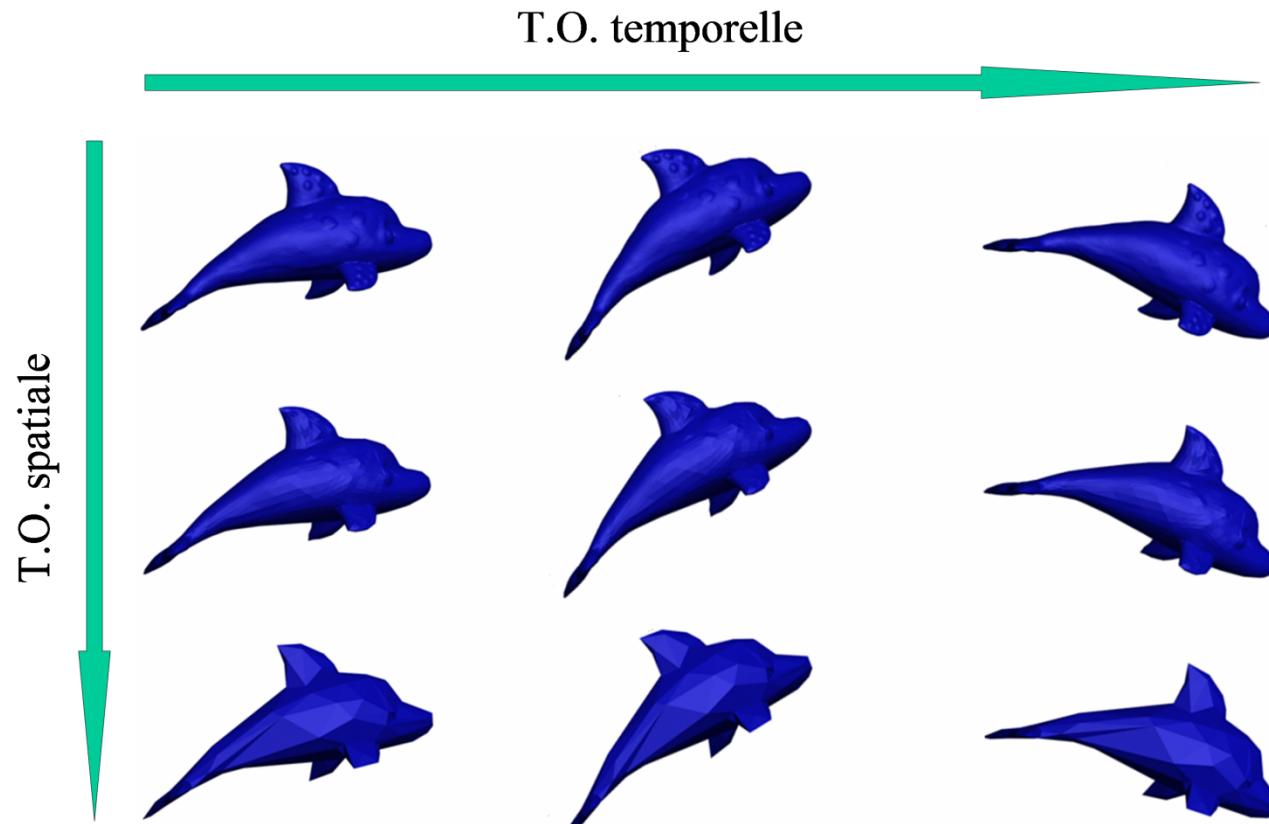
- Représentation Multirésolution
  - 1 maillage basses-fréquences
  - N sous-ensembles de détails (coefficients d'ondelettes) Vecteurs tridimensionnels





# Changement d'espace : ondelettes

Comment appliquer une transformée en ondelettes sur les **maillages animés** ?



# Analyse Multirésolution: Ondelette

- est bien adaptée aux signaux non-stationnaires
- permet une décomposition spatio-fréquentielle de l'image
- permet une décomposition multirésolution
- pas d'effets de bloc
- permet la transmission progressive

coefficients d'ondelettes  
(détails perdus entre 2 résolutions)

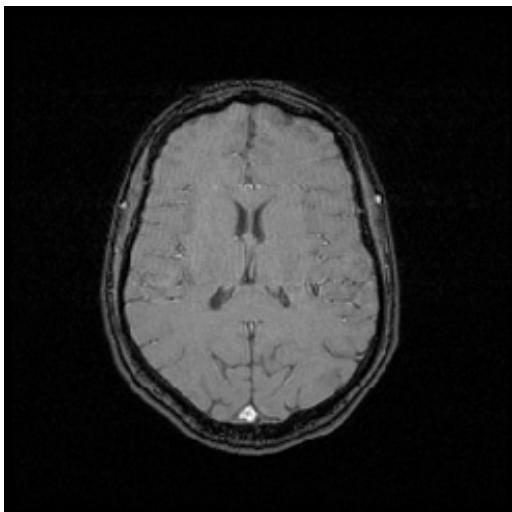


Image originale

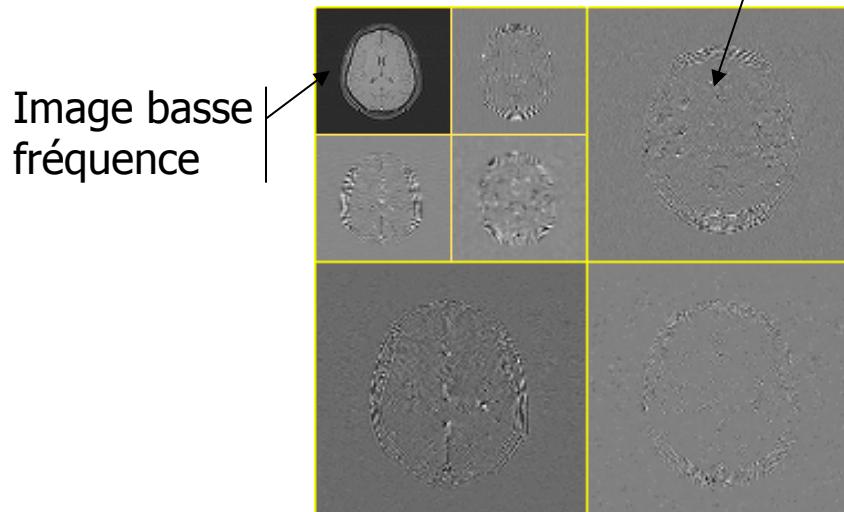
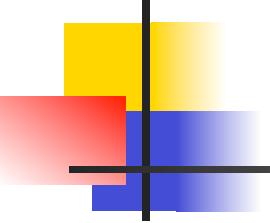
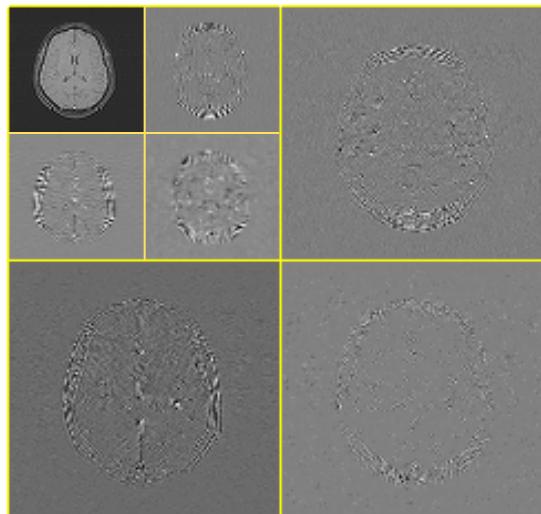


Image transformée



# Analyse Multirésolution

2 approches : **interbandes** ou **intrabandes**



on code les **dépendances**  
entre les sous-images.

chaque sous-image est codée  
de manière **indépendante**.

# Analyse Multirésolution

## Approche interbandes

Codage des coefficients non significatifs  
(arbres de zéros)

1 zéro parent dans la BF (résolution  $L$ )

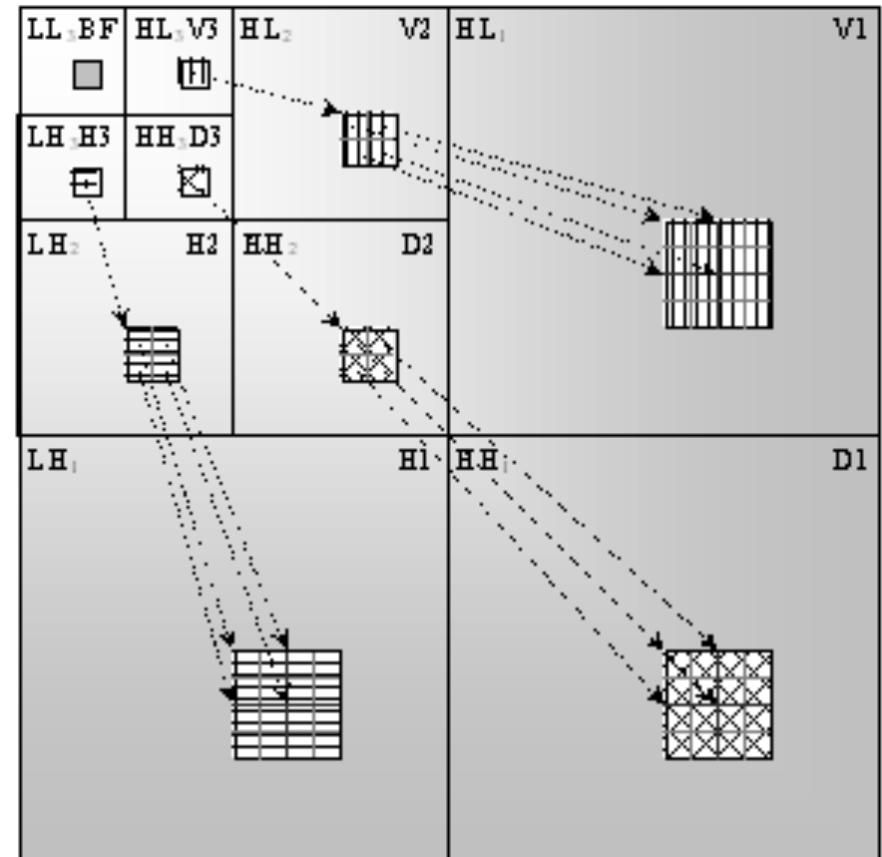
$\downarrow$   
 $4^L$  coefficients au total

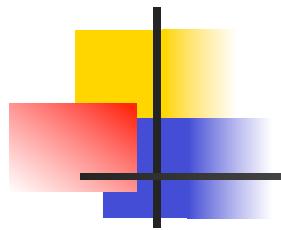
Passes successives pour déterminer le seuil  
qui permettra d'atteindre le débit cible

Avantages : simplicité, efficacité

Inconvénient : sensibilité aux erreurs de transmission

Algorithmes EZW, SPIHT



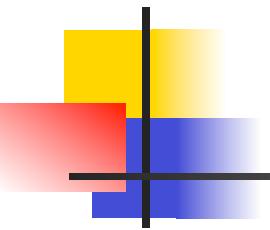


# Analyse Multirésolution

## Approche interbandes

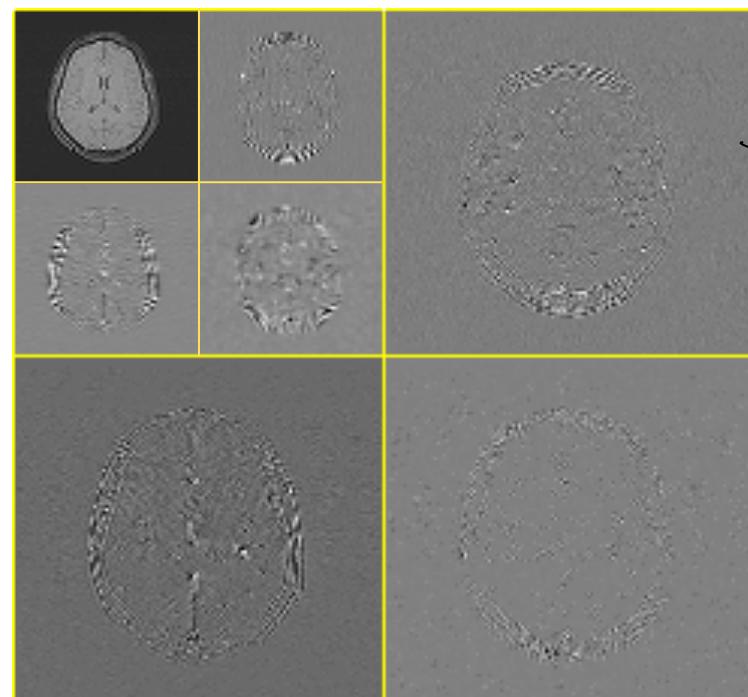
Transmission progressive (bits significatifs + raffinement)





# Analyse Multirésolution

## Approche intrabandes



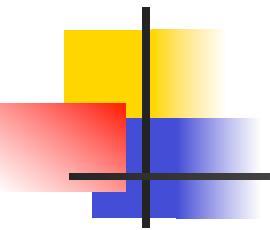
Nécessité de définir un débit pour chaque sous-bande

Sous-bande  $k$  :  $R_k$

Au total :  $R = \sum_k R_k = R_{cible}$

Problème : obtenir la distorsion totale la plus faible

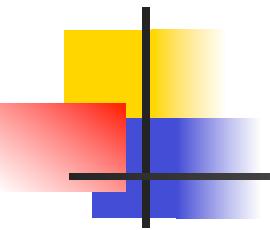
(cf paragraphe JPEG2000)



# Plan du Cours

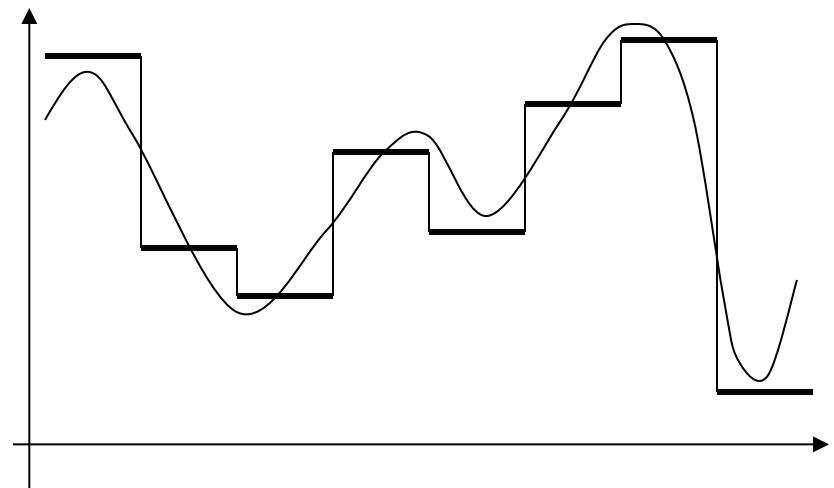
---

1. Introduction
2. Outils mathématiques de base
3. Stratégie de compression
4. Transformée
5. Quantification
6. Codage entropique
7. Compression d'images fixes : de JPEG à JPEG2000
8. Compression de vidéos : de MPEG I à MPEG IV
9. Transmission de documents confidentiels et sécurité

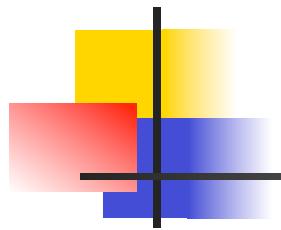


# Quantification : Principe

**BUT :** Représenter un signal numérisé sur  $L_1$  niveaux par  $L_2$  niveaux     $L_2 < L_1$



Adaptation optimale  
des niveaux au signal



# Quantification : Principe



**DICTIONNAIRE**  $C = \{\mathbf{y}_i\}_{i \in \mathfrak{S}} \subset \mathbf{R}^n$  avec  $\mathfrak{S}$  ensemble d'index

**QUANTIFICATION**

$$Q = \beta \circ \alpha$$

avec  $\alpha: \mathbf{R}^n \rightarrow \mathfrak{S}$  et  $\beta: \mathfrak{S} \rightarrow C$

$$\mathbf{x} \mapsto i \qquad \qquad i \mapsto \mathbf{y}_i$$

On a donc : 
$$Q(\mathbf{x}) = \mathbf{y}_i$$

# Quantification : Principe

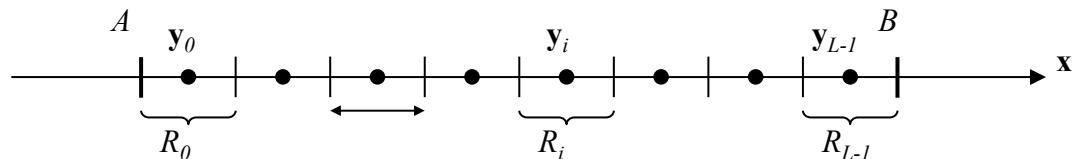
**DICTIONNAIRE**



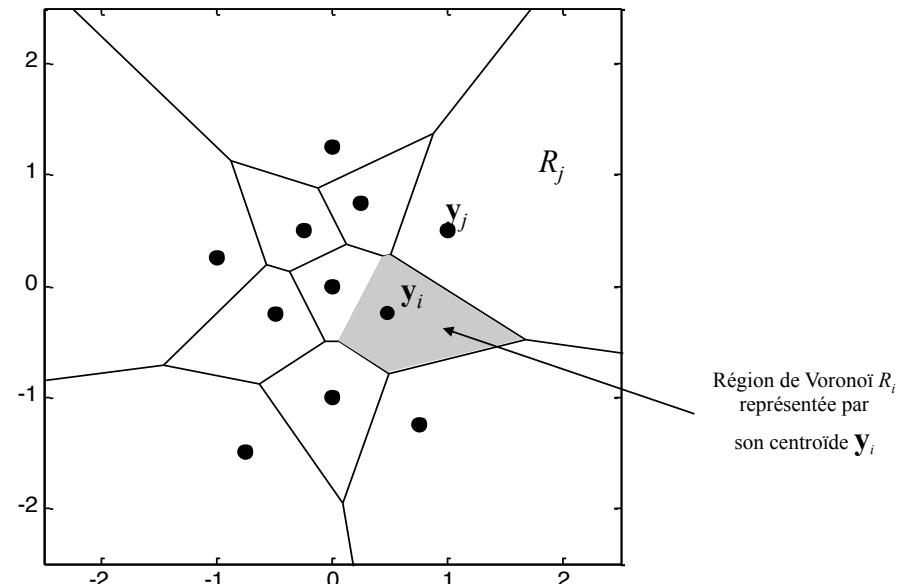
Partitionnement de l'espace en régions de Voronoï

$$\mathfrak{R}_i = \left\{ \mathbf{x} \in \mathbf{R}^n \mid \alpha(\mathbf{x}) = i \right\} = \left\{ \mathbf{x} \in \mathbf{R}^n \mid Q(\mathbf{x}) = \mathbf{y}_i \right\}$$

- cas scalaire ( $n = 1$ )



- cas vectoriel ( $n > 1$ )



# Quantification scalaire uniforme

QSU : le plus simple qui existe

$$Q(x) = y_i \text{ si } x_{i-1} \leq x < x_i \text{ pour } i = 1, 2, \dots, L$$

$$\mathfrak{R}_i = [x_{i-1}, x_i[ = \Delta \quad \text{pas de quantification}$$

Implantation :

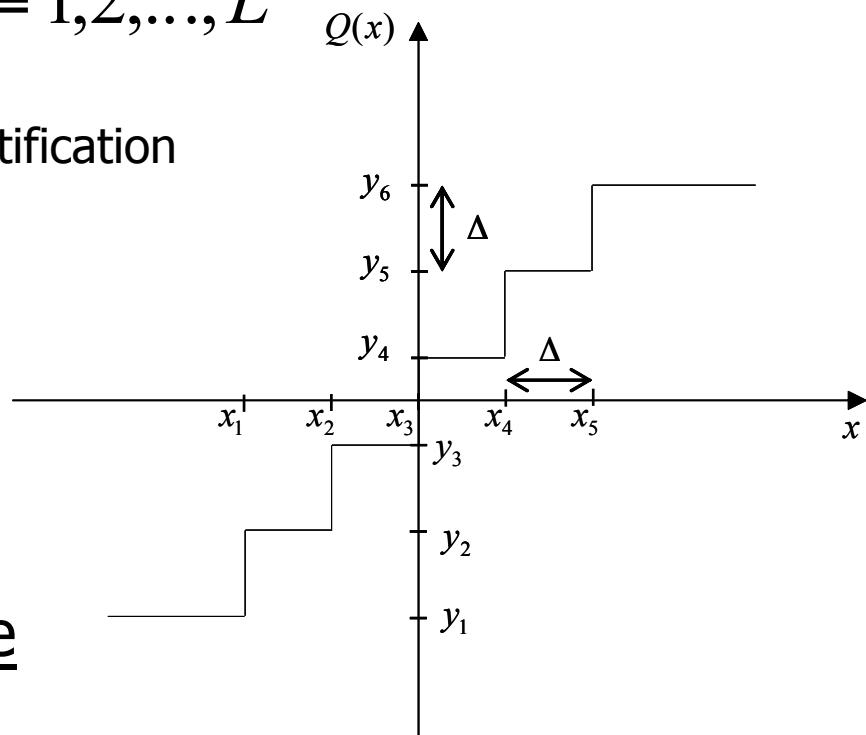
$$\alpha(x) = \left\lfloor \frac{x}{\Delta} \right\rfloor \quad \beta(i) = \left( i + \frac{1}{2} \right) \cdot \Delta$$

QSU optimal pour une source uniforme

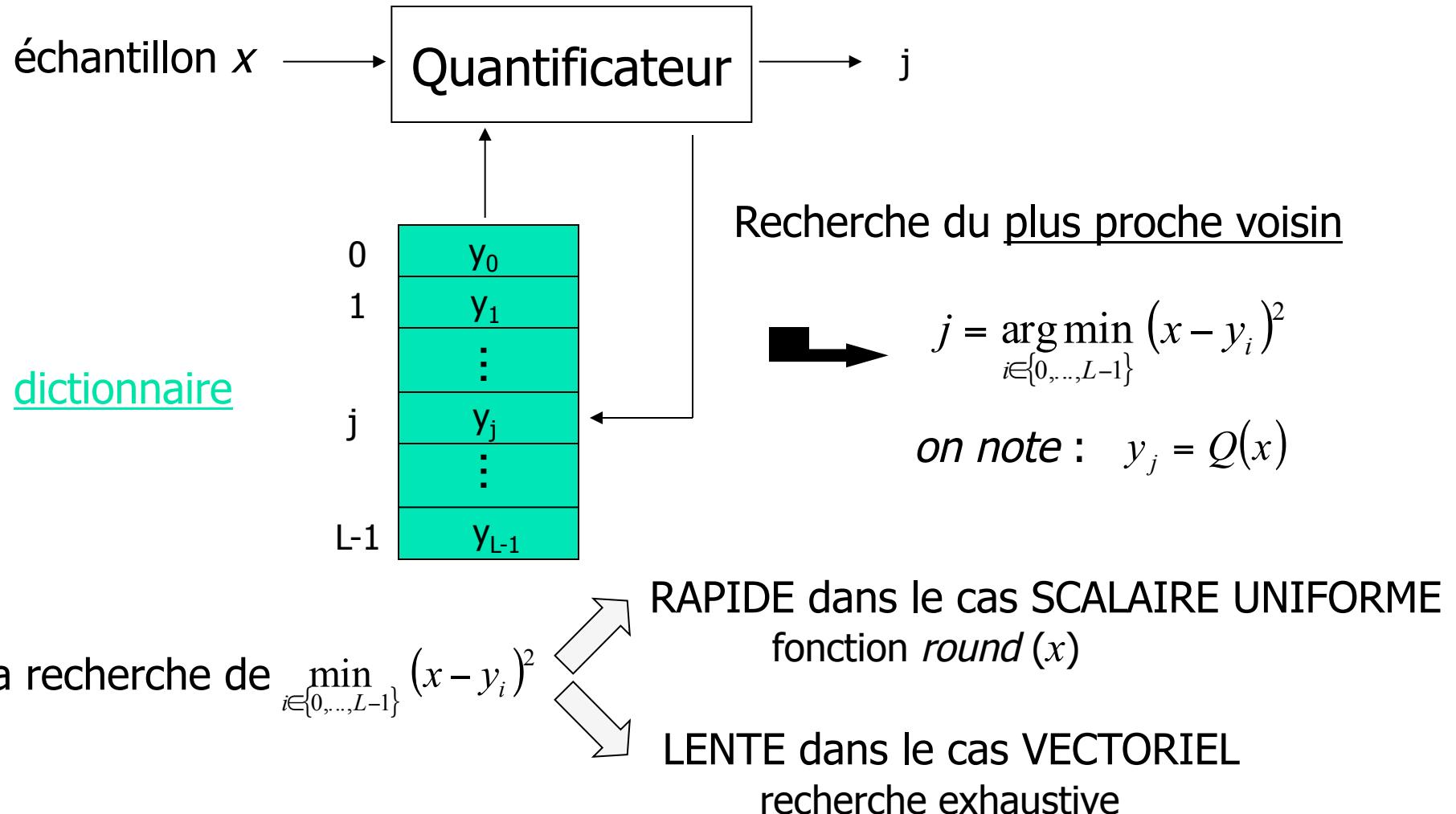
pour une source non uniforme : *QS optimal* (Max-Lloyd)

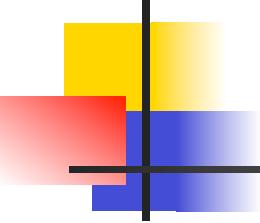


« pas » variable (s'adapte à la statistique)



# Compression par quantification





# Caractéristiques de quantification

**Débit binaire** (dictionnaire contenant  $L$  représentants)

Maximal :  $R_{\max} = \log_2 L$  bits / échantillon

Entropique :  $R_e = -\sum_{k=1}^L p_k(y_k) \log_2 p_k(y_k)$  bits / échantillon  
 $R_e \leq R_{\max}$

Exemple :



$$\text{Taux de compression} = \frac{8 \text{ bits / pixel}}{4 \text{ bits / pixel}} = 2 : 1$$

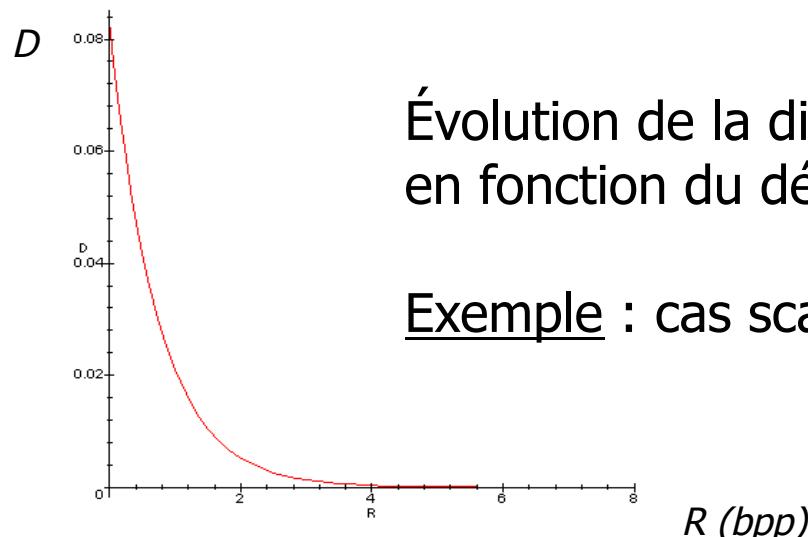
# Caractéristiques de quantification

## Distorsion moyenne

$$D = \int_{-\infty}^{+\infty} \|X - Q(X)\|^2 f_X(x) dx$$

$f_X(x)$  : densité de probabilité de la variable aléatoire d'entrée  $X$   
(intensité du pixel)

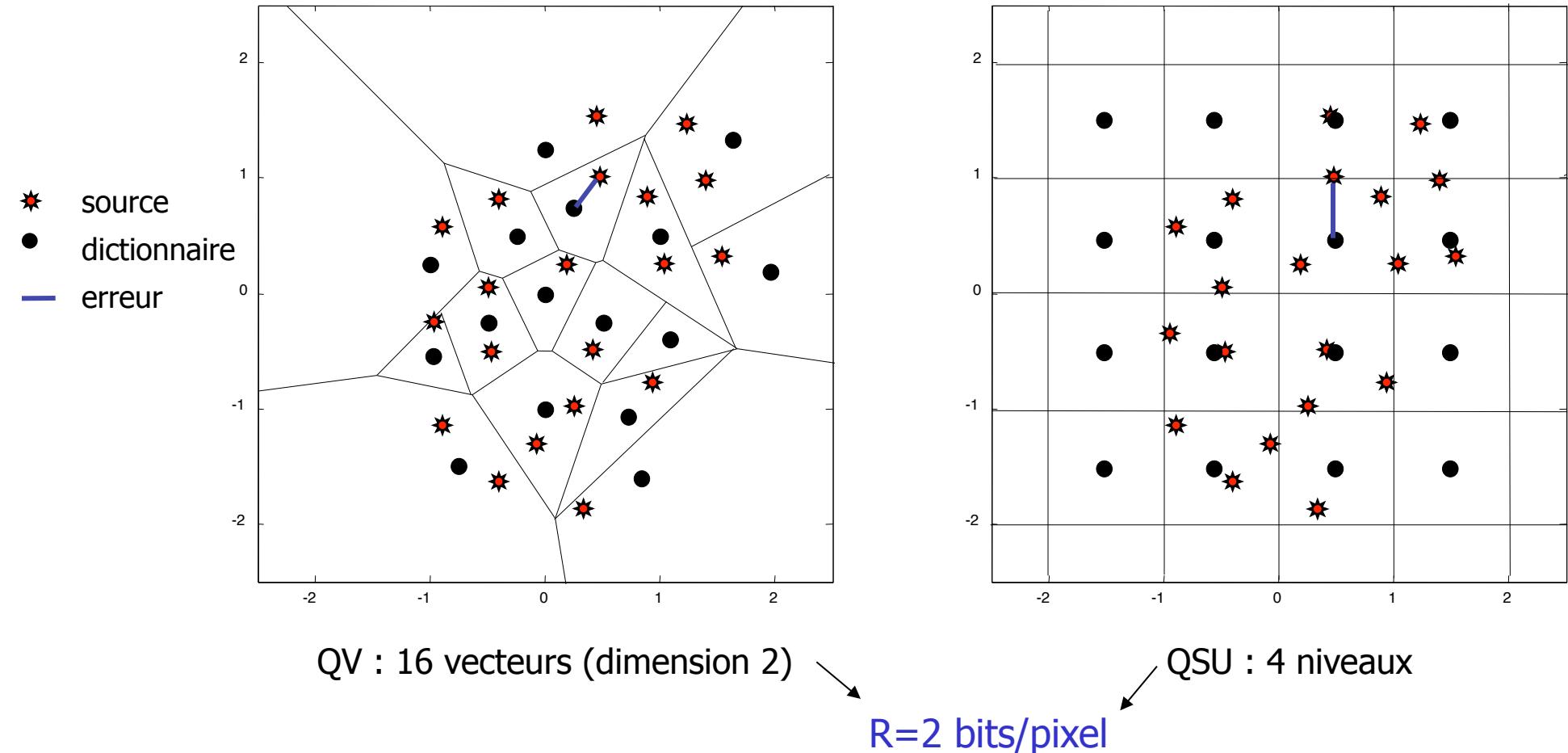
avec  $f_X(x) dx = p_X(x) = \text{probabilité}\{X = x\}$



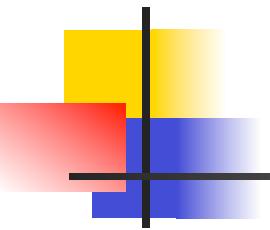
Évolution de la distorsion  
en fonction du débit

Exemple : cas scalaire uniforme  $D(R) = \frac{1}{12} 2^{-2R}$

# Avantage vectoriel/scalaire



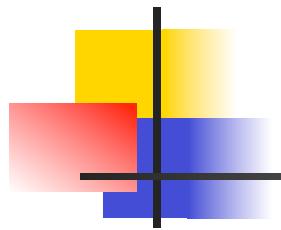
**QV : moins d'erreur – tous les vecteurs du dictionnaire sont utiles**



# Plan du Cours

---

1. Introduction
2. Outils mathématiques de base
3. Stratégie de compression
4. Transformée
5. Quantification
6. Codage entropique
7. Compression d'images fixes : de JPEG à JPEG2000
8. Compression de vidéos : de MPEG I à MPEG IV
9. Transmission de documents confidentiels et sécurité



# Codage sans perte (notions)

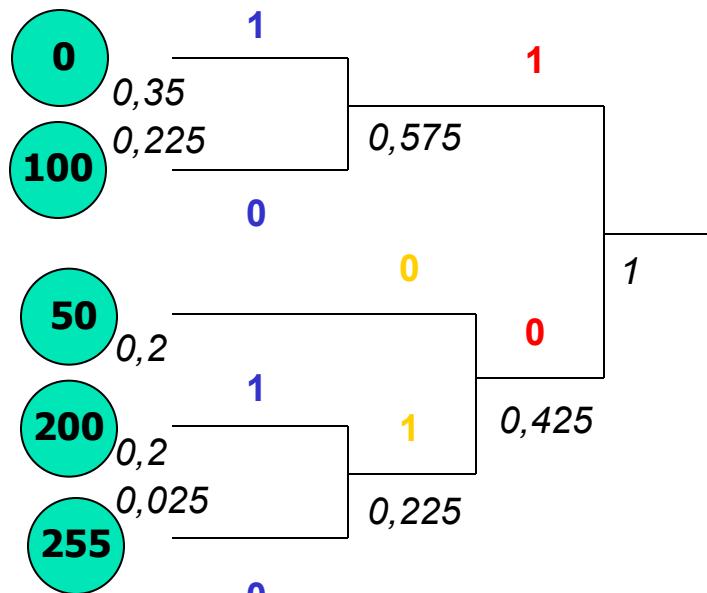
- longueur fixe / longueur variable
- décodage unique (sans ambiguïté)
- code préfixe

# Code de Huffman

Code préfixe mis au point par D.A. Huffman en 1952

Exemple : Soient les symboles 0,50,100,200,255 à coder.

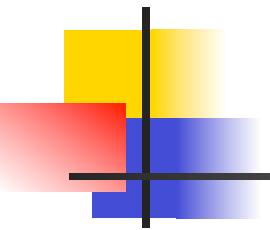
k	I(k)	p <sub>k</sub>
0	0	0,35
1	50	0,2
2	100	0,225
3	200	0,2
4	255	0,025



k	I(k)	Code c(k)
0	0	11
1	50	00
2	100	10
3	200	011
4	255	010

avantages : faible complexité, code instantané, plus petit  $\bar{l}$  parmi tous les codes instantanés  
inconvénients : performances faibles, nécessite de connaître toute la source

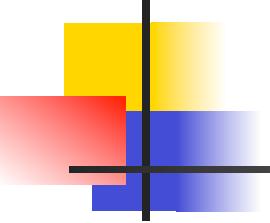
Version adaptative du code de Huffman



# Plan du Cours

---

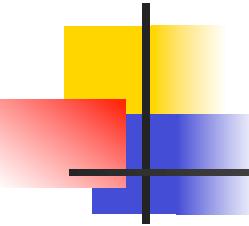
1. Introduction
2. Outils mathématiques de base
3. Stratégie de compression
4. Transformée
5. Quantification
6. Codage entropique
7. Compression d'images fixes : de JPEG à JPEG2000
8. Compression de vidéos : de MPEG I à MPEG IV
9. Transmission de documents confidentiels et sécurité



# Historique

## Recherches et Normes en Compression d'images

- 1964      FFT      Transformée de Fourier Discrète
- 1974      DCT      Transformée en cosinus discrète
- 1990-92    DWT      Ondelettes Bi-orthogonales
- 1992      Norme JPEG
- 2000      Norme JPEG 2000



# JPEG (Joint Picture Expert Group)

## Objectifs :

- Comprimer des images fixes (couleur ou niveaux de gris)
- Normalisation en 1992 par deux groupes d'experts : ISO<sup>1</sup> et CCITT<sup>2</sup>
- Version **avec pertes** (JPEG - 1992) ou **sans pertes** (JPEG LS - 1998)
- Version pour image avec un nombre limité de niveaux (JBIG)
- Qualité d'image réglable

## Principales applications :

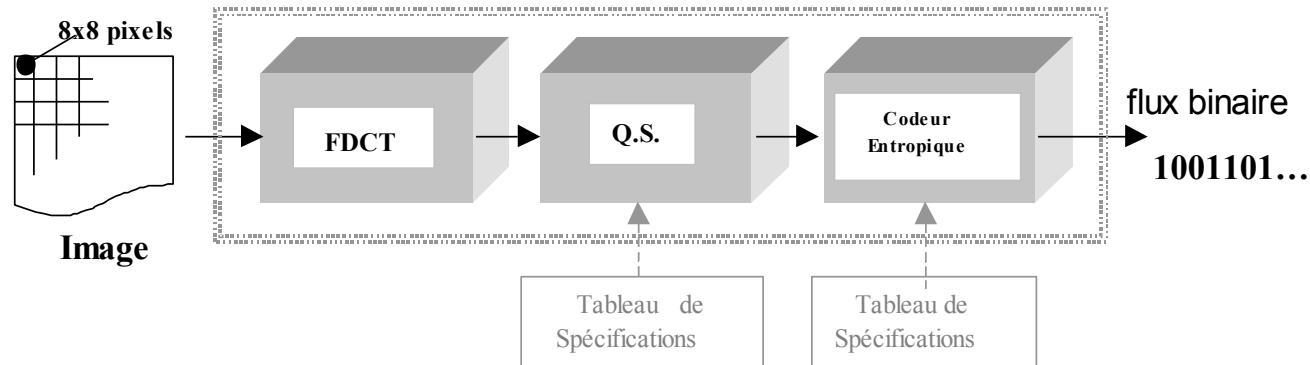
- Fax, imprimantes, Internet, appareils photos numériques,...

1- International Standard Organisation

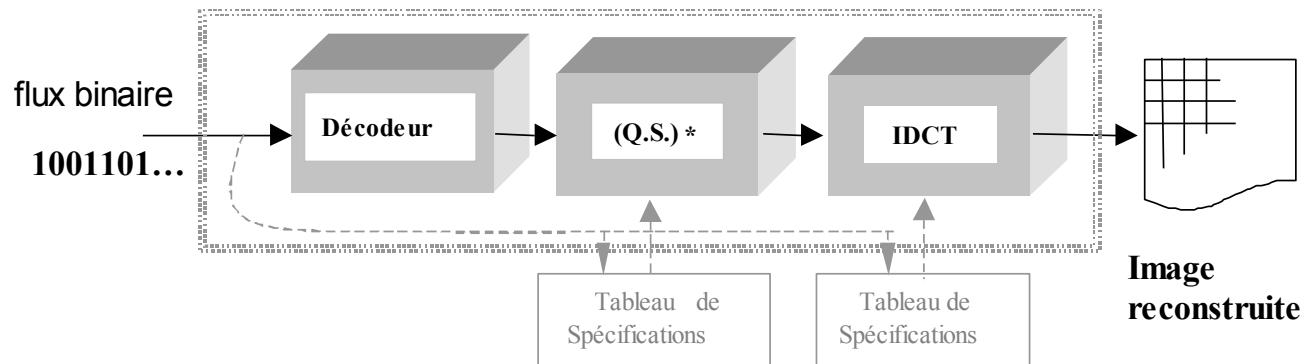
2- Comité Consultatif International Télégraphique et Téléphonique

# JPEG : Principe

## CODEUR



## DECODEUR



Performances (taux de compression / qualité) :

*Images couleur* : jusqu'à 50:1

-> peu de dégradation

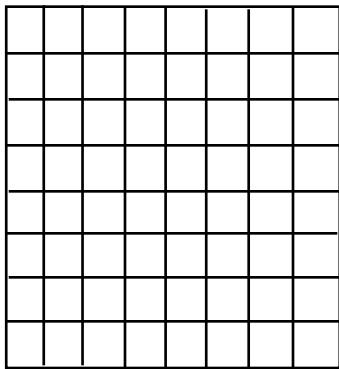
*Images niveaux de gris* : au-delà de 20:1

-> dégradations visibles

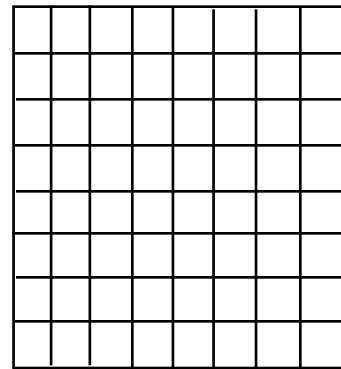
# JPEG : Principe

Traitement JPEG sur chaque bloc 8x8 d'une image :

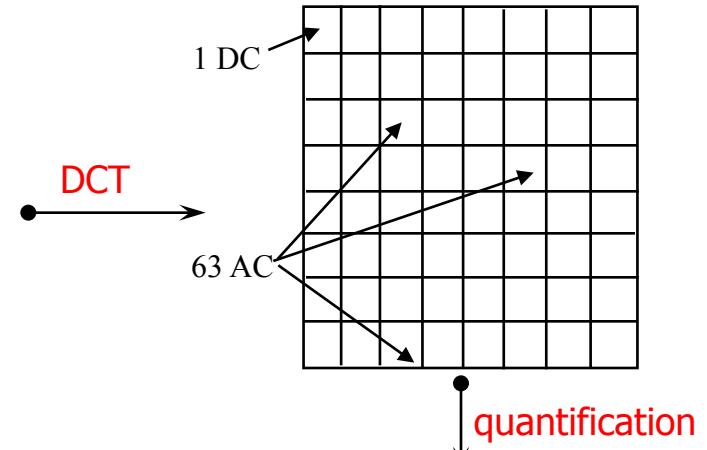
**8x8 échantillons 8 bits  
(entiers entre 0 et 255)**



**8x8 échantillons 8 bits  
(entiers entre -128 et +127)**



**8x8 coefficients DCT  
(réels entre -1023.0 et +1024.0)**

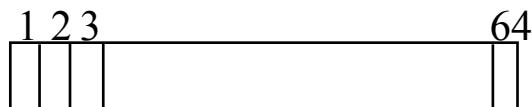


quantification

1000101010111...

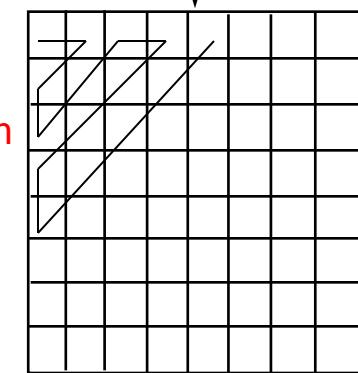
séquence de bits

Huffman

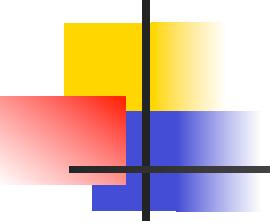


**64 coefficients ordonnés  
basses fréquences -> hautes fréquences**

zigzag scan



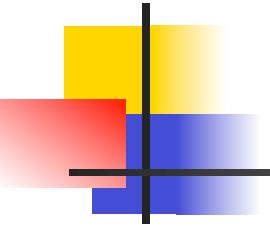
**8x8 coefficients quantifiés**



# Quelques Résultats

Image d'origine

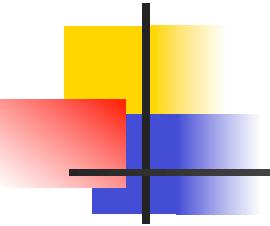




# Quelques Résultats

TC = 10:1

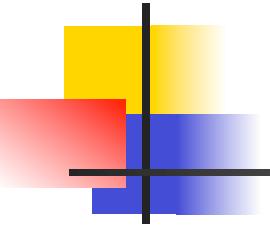




# Quelques Résultats

TC = 20:1

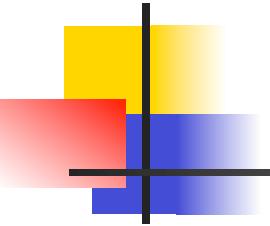




# Quelques Résultats

TC = 30:1

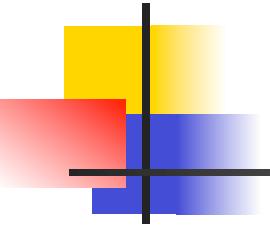




# Quelques Résultats

TC = 40:1

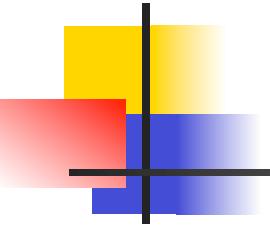




# Quelques Résultats

TC = 60:1

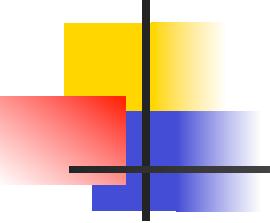




# Quelques Résultats

TC = 80:1

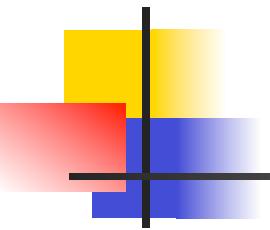




# Quelques Résultats

TC = 120:1





# Avantage Et Inconvénients

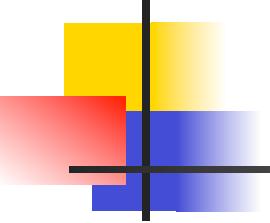
- **AVANTAGE** : Gros succès de JPEG
  - 80% des images sur le web seraient encodées JPEG ;
  - Appareils photos numériques.
- **MAIS** :
  - Efficacité de codage limitée ;
  - Effets visuels de blocs à forte compression ;
  - Les applications d'imagerie demandent de nouvelles fonctionnalités non supportées par JPEG.
- Souhait du comité JPEG de définir une nouvelle norme pour répondre à ces 3 problèmes : **JPEG 2000**.



# Le Futur : JPEG 2000

## Critères exigés (« requirements ») :

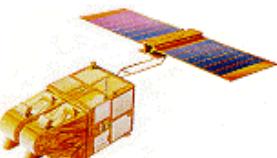
- Excellent rapport distorsion / débit (30% « meilleur » que JPEG)
- Gestion de 2 à 16 millions de couleurs sur la même architecture ;
- Compression avec ou sans pertes ;
- Transmission progressive par résolution et par raffinement ;
- Faible complexité algorithmique ( euh !!! ) ;
- Accès aléatoire dans le fichier compressé pour extraction de régions ; d'intérêt (ROI - Regions Of Interest) ;
- Robustesse aux erreurs de transmission ;
- Protection des informations pour l'exploitation correcte de l'image.



# Le Futur : JPEG 2000

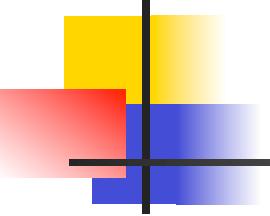


## Applications visées par JPEG 2000



- Internet
- Appareils photo numériques
- Imprimantes
- Scanners
- Télécopie
- Images médicales
- Télécommunications mobiles
- Images Satellites





# Le Futur : JPEG 2000



## Les besoins diffèrent selon les applications



### ***Médical :***

- sans pertes (visuelles au moins)
- région d'intérêt (ROI)
- 12 bits de profondeur au moins



### ***Mobiles :***

- robustesse aux erreurs de transmission



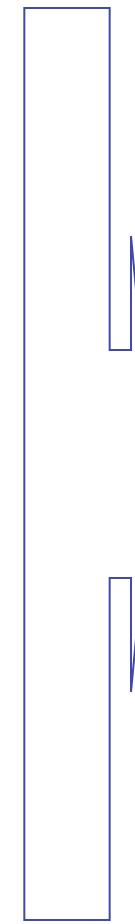
### ***Satellite :***

- capacité de stocker des images énormes
- traitement au « fil de l'eau »



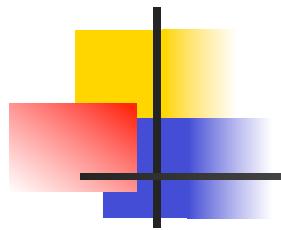
### ***Appareil photo numérique :***

- codage temps réel
- faible complexité



**Analyse multirésolution**

**Ondelettes**

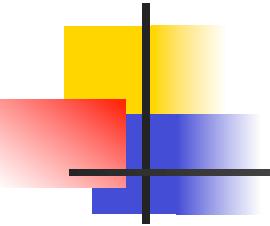


# Processus De Normalisation

- Projet défini en 1996 ;
- Appel à contribution lancé en Mars 1997 ;
- 22 algorithmes candidats sont présentés ;
- Tests objectifs (mesure qualité) et subjectifs (visuels).

## Structure de base retenue

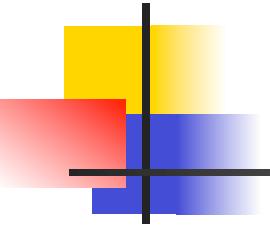
1. Transformée en ondelettes (Filtres 9-7)
2. Codeur par plan de bits
3. Codeur Entropique



# Processus De Normalisation

- Tous les 3 mois, réunion d'environ 100 experts internationaux ;
- Principe des « core experiments » qui sélectionnent les techniques les plus intéressantes ;
- Le « Verification Model » intègre ces technologies et permet de les comparer,

=> PLATEFORME DE TESTS

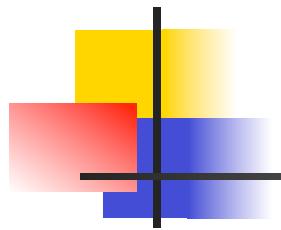


# Processus De Normalisation

- En décembre 1999 : « Working Draft » ;
- « Committee Draft » adopté en mars 2000 ;
- Version finale (« International Standard ») fin 2000.

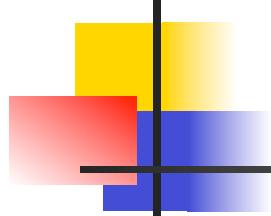
## Qu'est-ce qui est normalisé ?

- Seuls la syntaxe et le décodeur sont normalisés ;
  - Le codeur est seulement informatif.

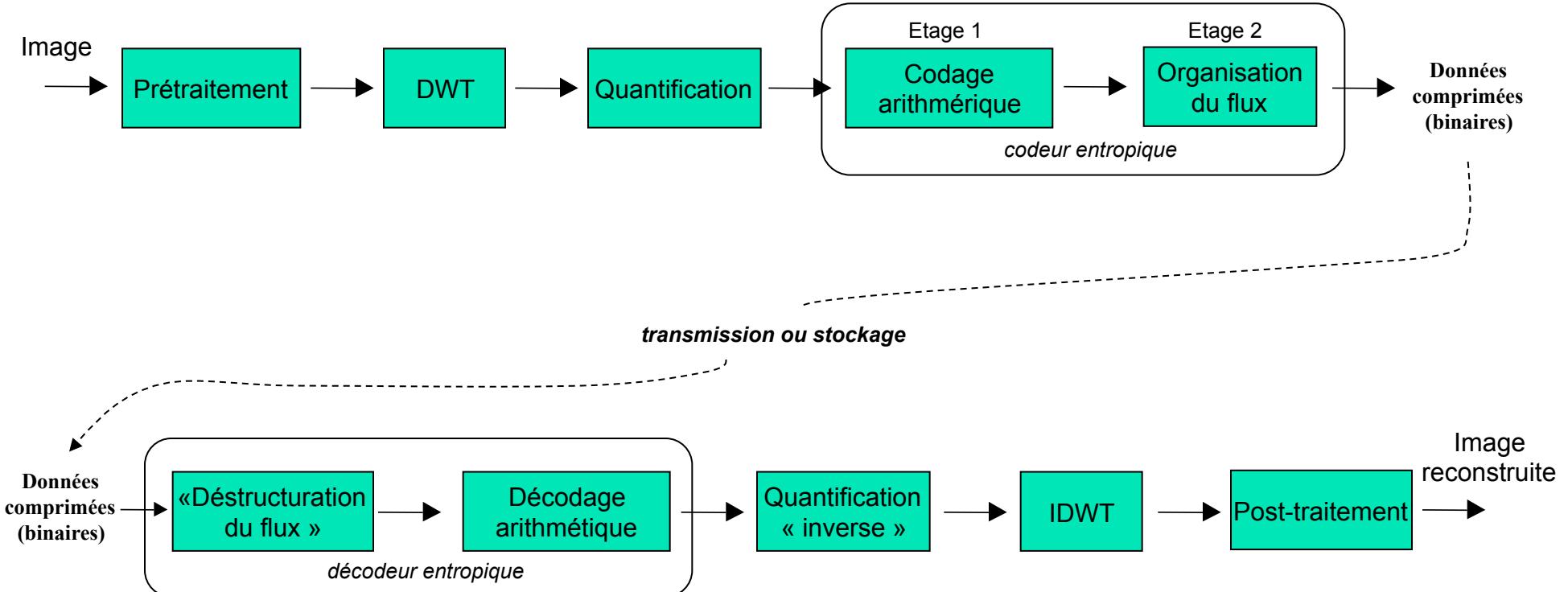


# Caractéristiques De JPEG 2000

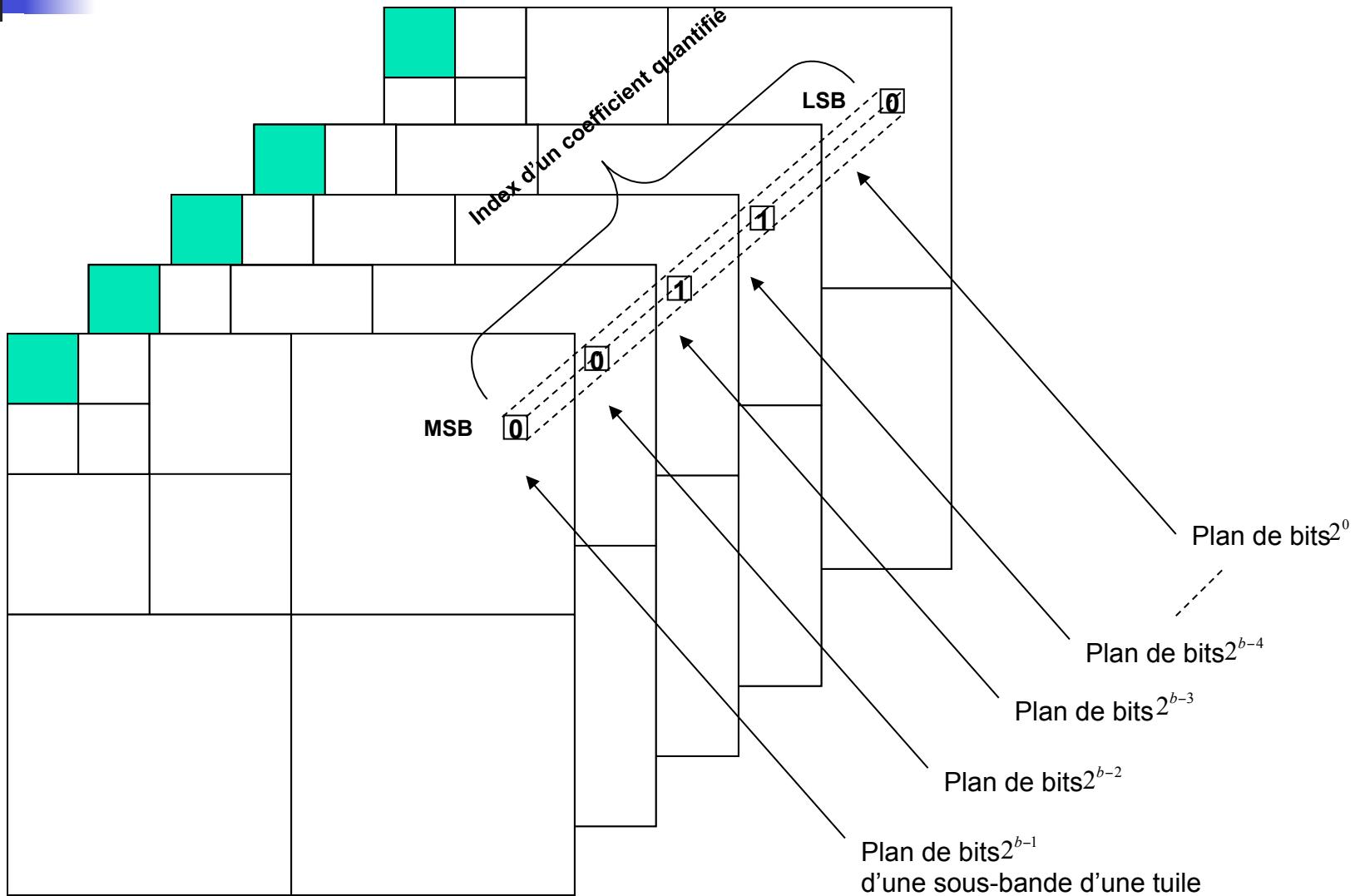
- Gestion :
  - des images multi-composantes (ex.: couleur) ;
  - des dynamiques de 1 à 32 bits ;
- Découpage de l'image en « tuiles » et transformation de chaque « tuile » ;
- Choix de transformées en ondelettes (lifting ou convolution). Filtres pré-implémentés ou utilisateurs ;
- Multirésolution : Nombre de niveaux de décomposition variable et choix de l'arbre de décomposition ;
- Codage par blocs uniformes de 64x64 coefficients transformés.

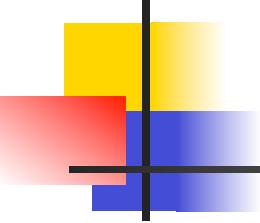


# JPEG 2000 : schéma



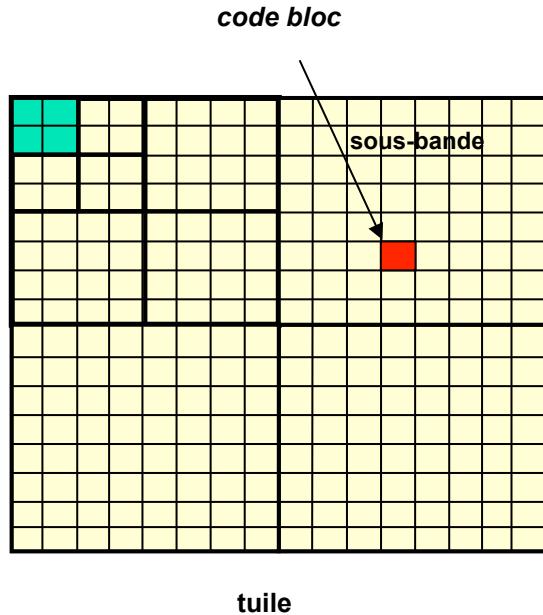
# JPEG 2000 : plans de bits





# JPEG 2000 : éléments traités

Image (ou composante d'image)



Chaque code bloc est codé de manière indépendante  
(codage arithmétique)

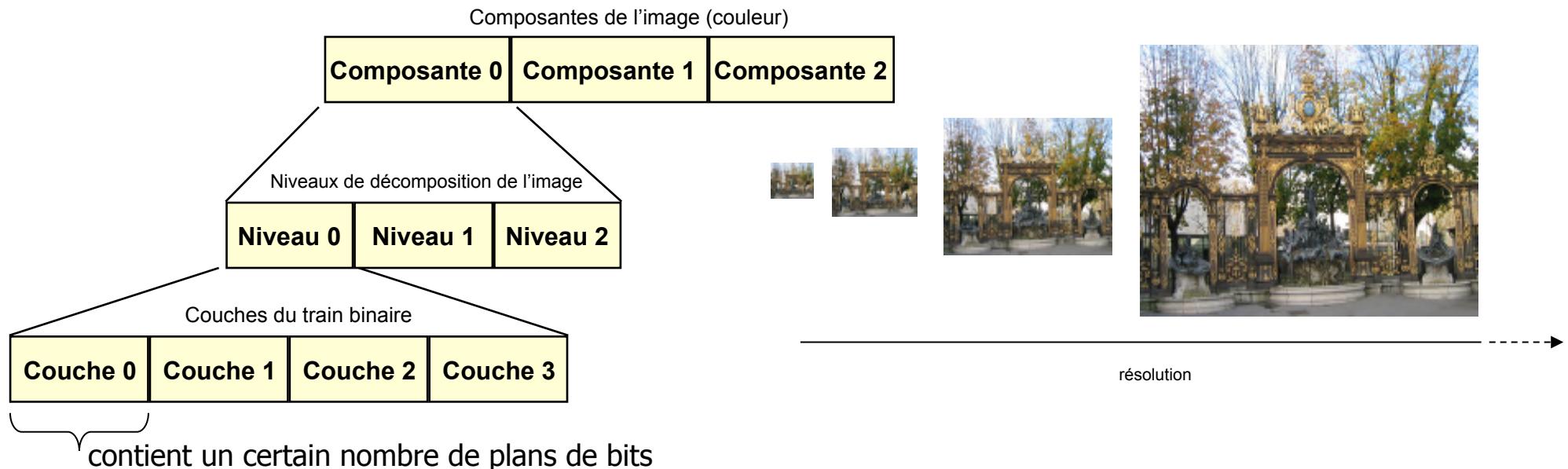


Plus de flexibilité que dans EZW  
(accès aléatoire, manipulations géométriques,  
parallélisation des calculs)

# Organisation du train binaire

## Etage 2

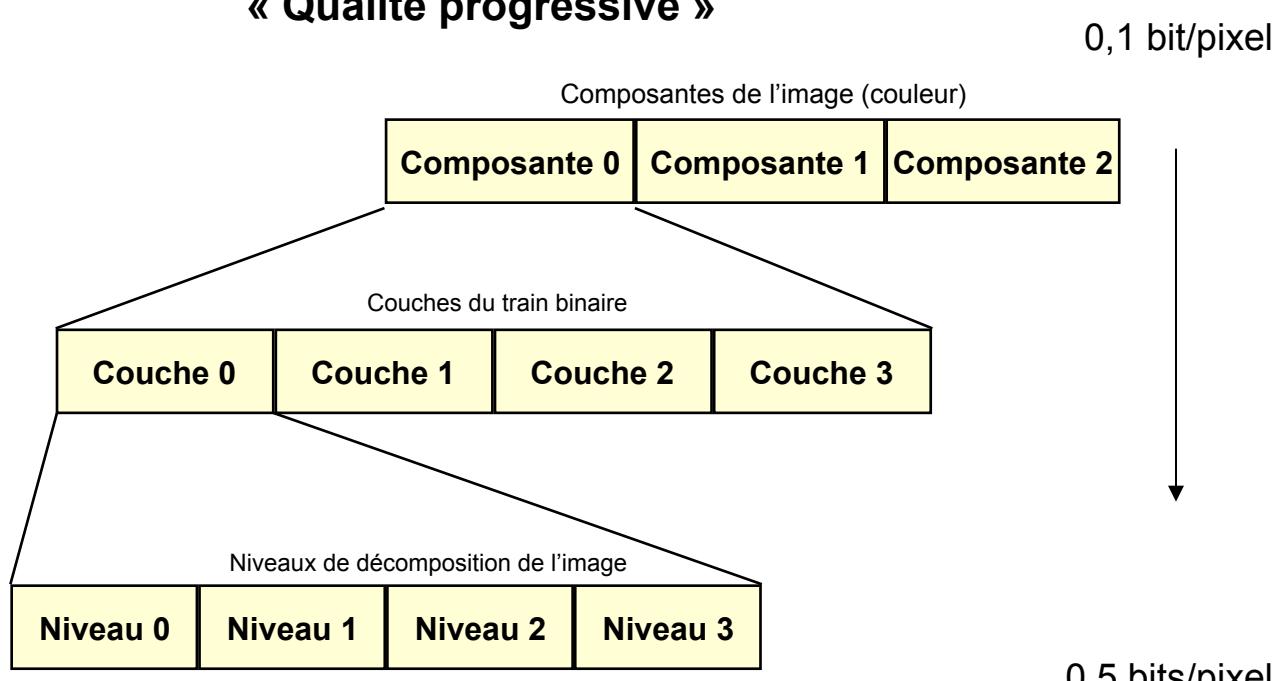
### « Résolution Progressive »



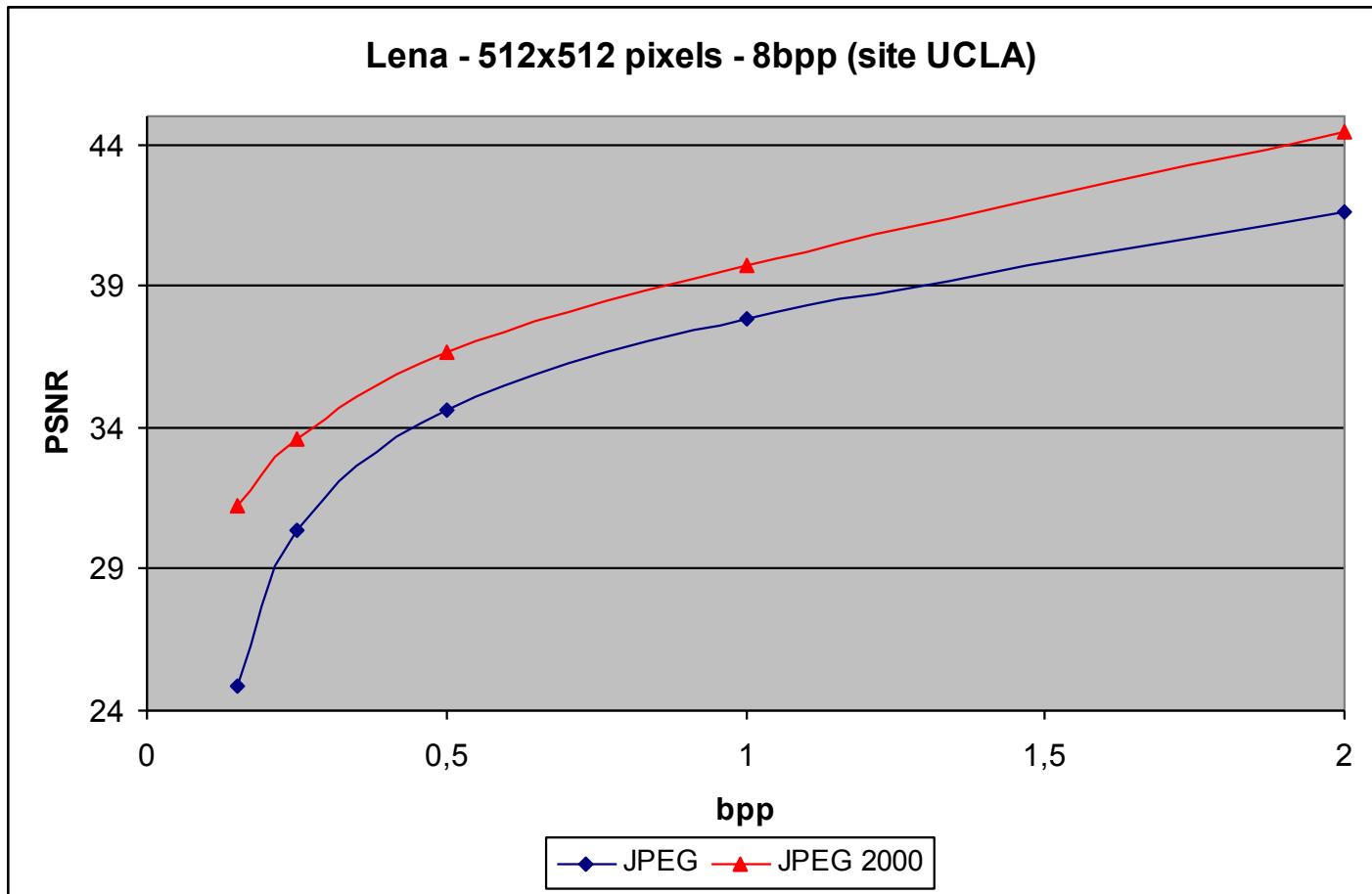
# Organisation du train binaire

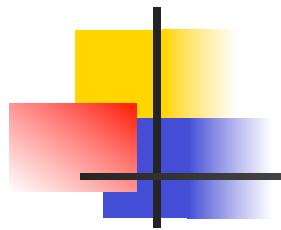
Etage 2

« Qualité progressive »



# JPEG vs JPEG 2000





# JPEG vs JPEG 2000

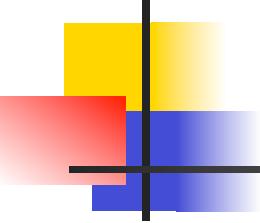
JPEG (DCT)



Ondelettes (JPEG-2000)



Taux de Compression 80:1



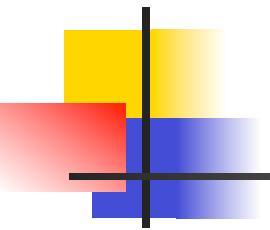
# Les Sites Internet

**Le site officiel JPEG :**

<http://www.jpeg.org/>

**Un modèle de vérification en JAVA est disponible à l'adresse :**

<http://www.jj2000.epfl.ch/>

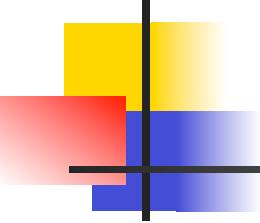


# Plan du Cours

1. Introduction
2. Outils mathématiques de base
3. Stratégie de compression
4. Transformée
5. Quantification
6. Codage entropique
7. Compression d'images fixes : de JPEG à JPEG2000
8. **Compression de vidéos : de MPEG I à MPEG IV**
9. Transmission de documents confidentiels et sécurité

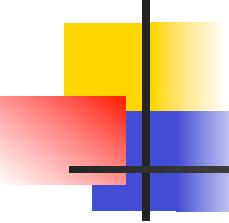
# La Famille MPEG





# La Norme MPEG

- MPEG est le nom donné au groupe de travail de l'ISO chargé de définir une norme de compression de séquences vidéo (définition exacte : *d'une succession d'images dans le temps accompagnée d'une bande sonore* ).
- En réalité MPEG signifie *Moving Pictures Expert Group* dont le nom ISO est  
**ISO/IEC JTC1 SC29 WG11**
  - ISO : International Organization for Standardisation
  - IEC : International Electro-Technical Commission
  - JTC1 : Joint Technical Committee 1
  - SC29 : Sub-Committee 29
  - WG11 : Worg Group 11 (Moving Pictures with Audio)
- MPEG définit des méthodes de compression, les éditeurs sont libres de les implanter comme bon leur semble pour commercialiser leurs produits.



# Les 3 Constituants MPEG

La première version définitive de MPEG (appelée MPEG phase I (abrégé MPEG1)) définit un flot de bits pour des signaux audio et vidéo compressés de manière **optimisée pour être relu à 1.5 Mbps**

## Les 3 Constituants

MPEG est divisé en 3 parties qui possèdent chacune sa propre définition comprenant la largeur de bande passante qui lui est accordée.

### Vidéo :

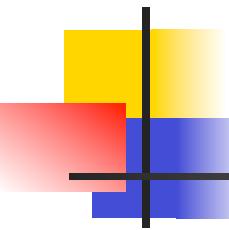
Définition de la méthode de compression des images animées dans le temps (1,15 MBps)

### Audio :

Compression des séquences audio (256 Kbit/s)

### System :

Synchronisation et multiplexage des séquences audio et vidéo.



# MPEG-1 En Chiffres

## MPEG Vidéo

### Résolution

- 352 x 240 x 30 images par seconde (aux US)
- 352 x 288 x 25 images par seconde (en Europe - CIF)

### Taux de compression

- 26 (taux maximum)

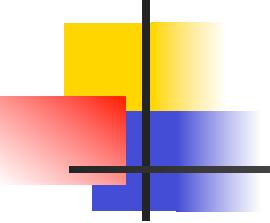
## MPEG Audio

### Numérisation, Signal/bruit

- 44,1 KHz et 8 bits

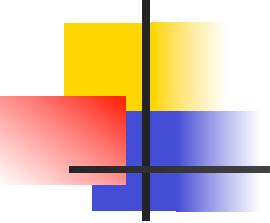
### Taux de compression

- 6 à 7 (Compression constatée, Max. théorique 22)



# Principes De MPEG Vidéo

- Le grand principe du codage repose sur **l'exploitation des redondances** qui existent entre les images successives d'une séquence ;
- Une fois les redondances temporelles déterminées, on utilise la DCT et plus précisément la norme JPEG pour comprimer celles-ci;
- Les informations obtenues : Coefficients DCT, vecteur déplacement, et paramètres de quantification sont ensuite codés grâce à un codage entropique de Huffman.



# Principes du codage vidéo

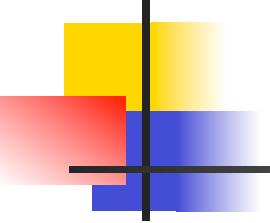
exploiter les **redondances** entre les images successives d'une séquence



coder l'erreur entre deux images qui se ressemblent

2 types d'information dans un flux MPEG :

- image
- mouvement



# Prédictions Temporelles

Il existe 3 types de Codage pour les redondances

## I (Intraframes)

Codage comme une image fixe, nécessaire car la séquence nécessite un commencement.

## P (Predicted Frames)

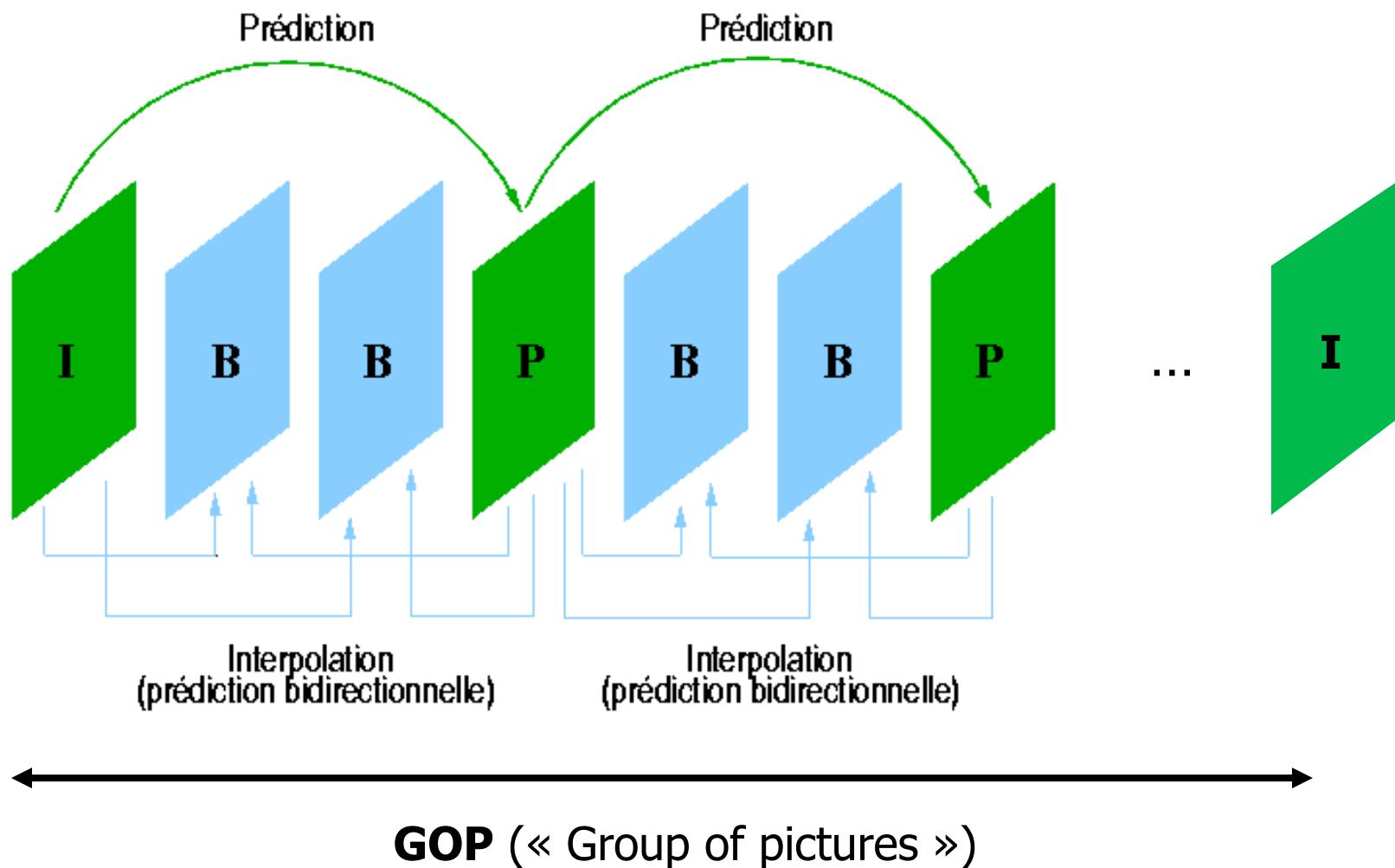
Image calculée à partir de l'image de type P ou I la plus récemment calculée (chaque bloc d'une image peut être codé par des méthodes différentes)

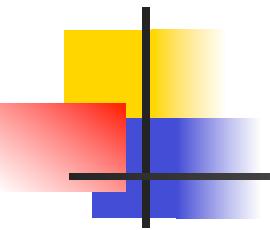
## B (Bidirectionnal Frames)

Image codée à partir des 2 images de type I ou P les plus récentes l'une dans le passé, l'autre dans le futur. 3 calculs pour connaître le meilleur codage possible :

- *à partir de l'image antérieure* ;
- *à partir de l'image future* ;
- *à partir de la moyenne des deux images*.

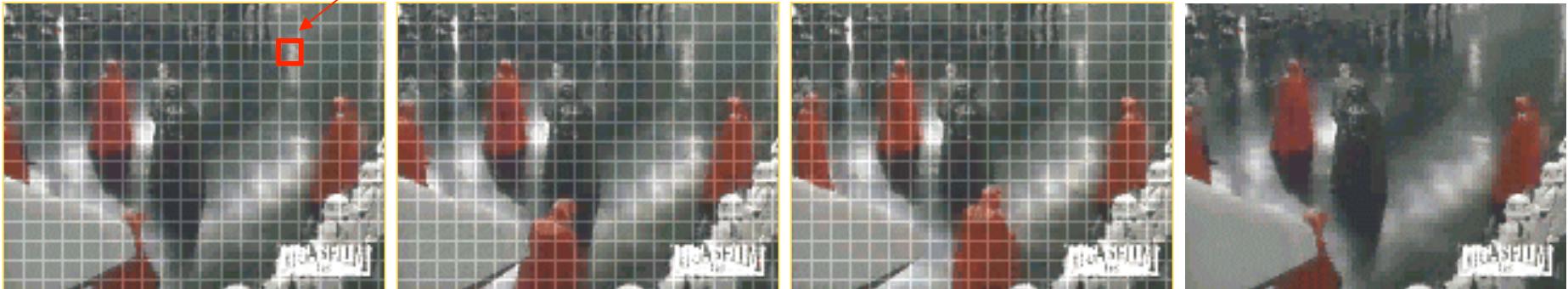
# Prédictions Temporelles



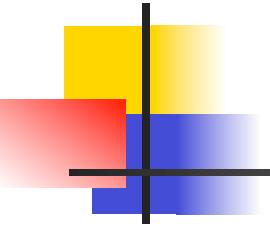


# La Compensation De Mouvement

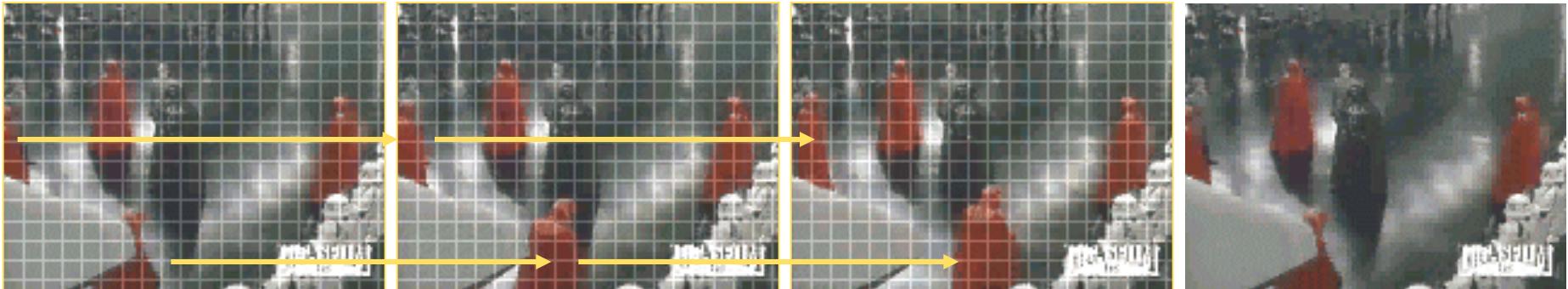
L'image est découpée en macroblocs de taille 16x16 pixels



Ici nous voyons trois frames consécutives qui possèdent le même décor, mais diffèrent dans la position de deux personnes.

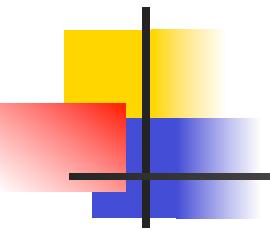


# La Compensation De Mouvement

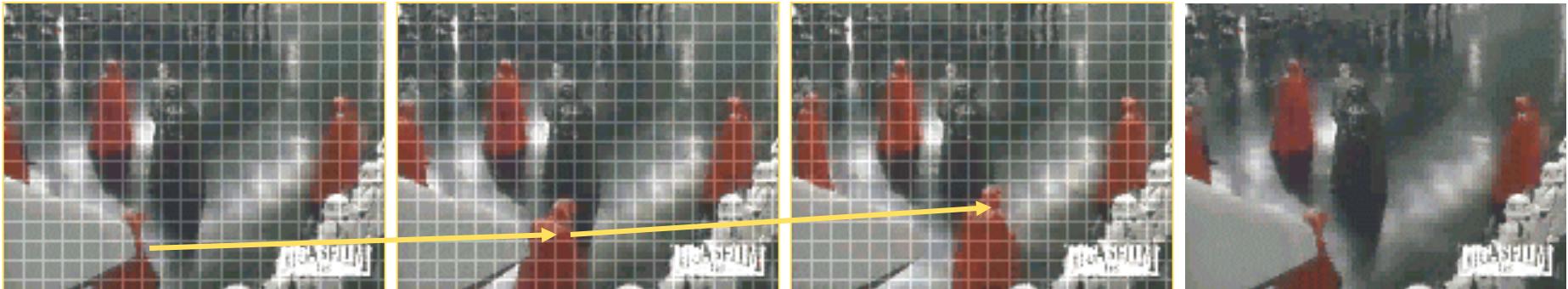


Ici nous voyons trois frames consécutives qui possèdent le même décor, mais diffèrent dans la position de deux personnes.

- Les macroblocs **contenant le décor** vont correspondre exactement ;

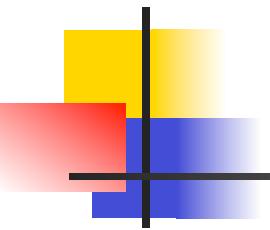


# La Compensation De Mouvement



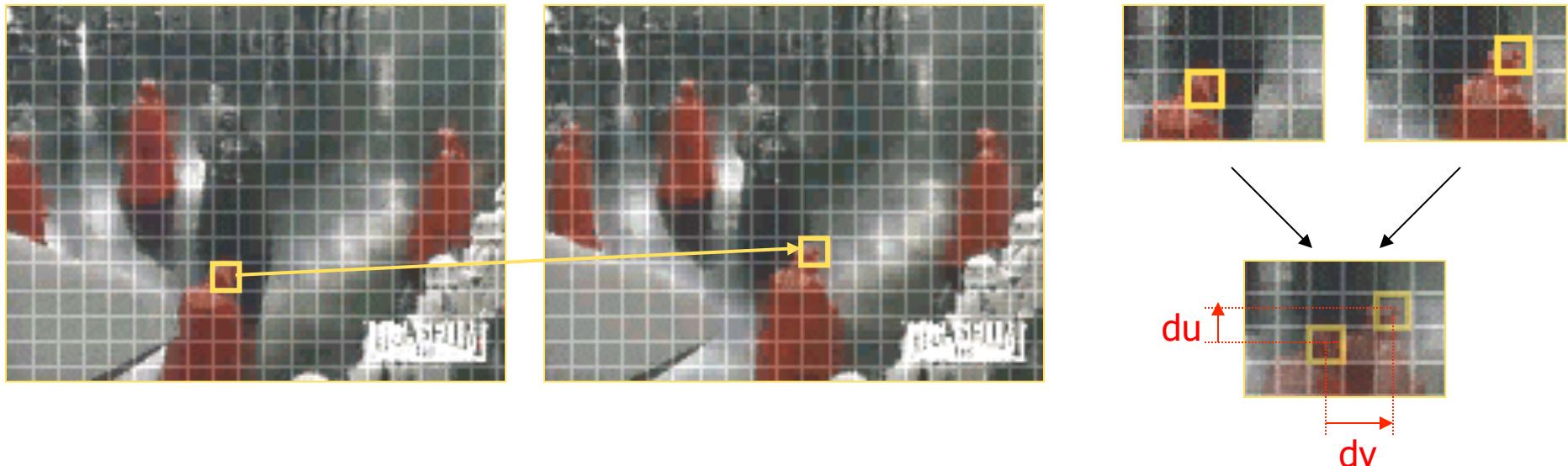
Ici nous voyons trois frames consécutives qui possèdent le même décor, mais diffèrent dans la position de deux personnes.

- Les macroblocs contenant le décors vont correspondre exactement ;
- Les macroblocs **contenant les deux personnes** en mouvement vont être décalés en position par une certaine quantité inconnue et vont devoir être dépistés : « block matching ».



# « Block Matching »

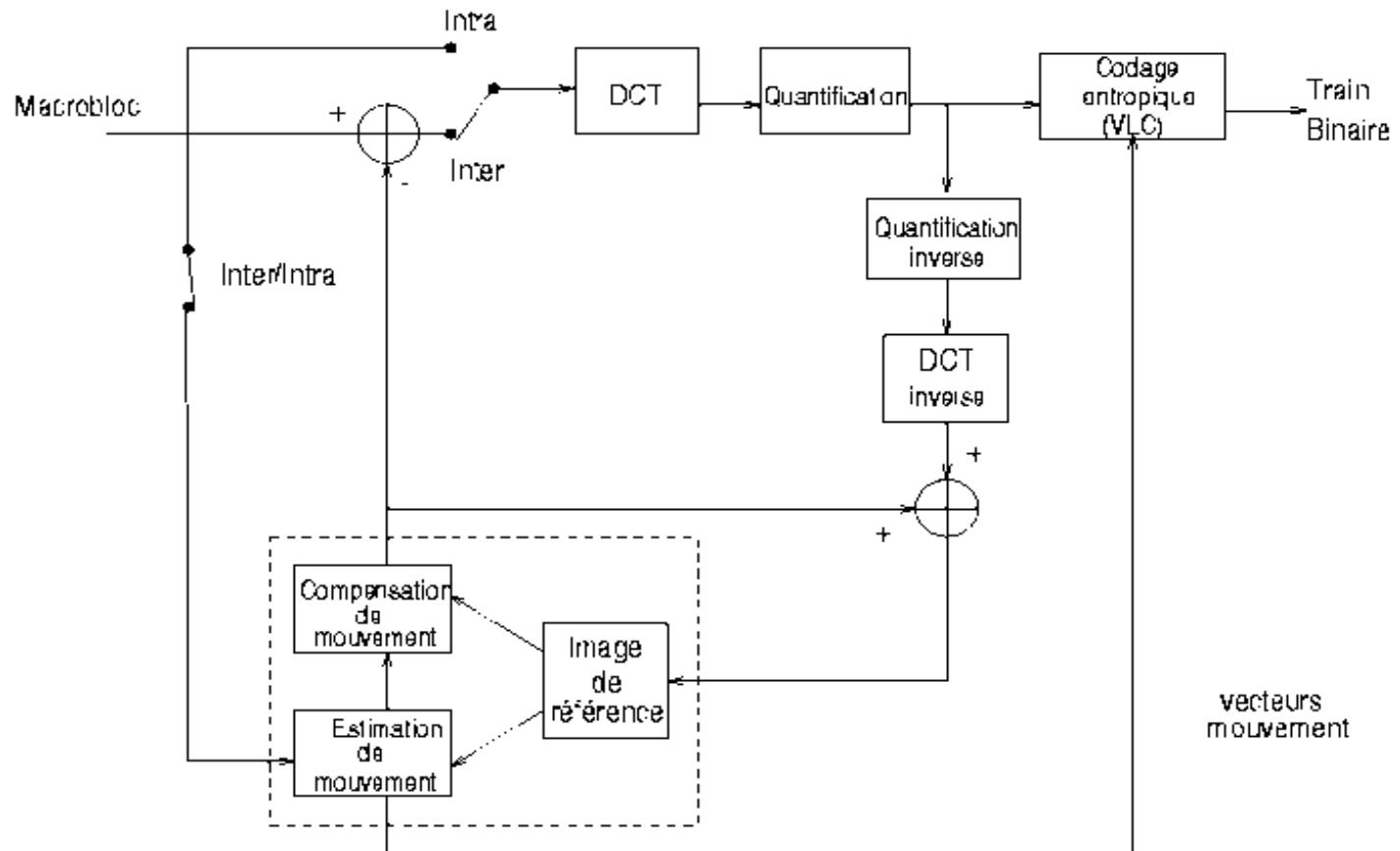
- Méthode non normalisée ;
- Consiste à chercher le vecteur déplacement ( $du, dv$ ) entre 2 blocs consécutifs.



## Méthodes de recherche du :

- maximum de corrélation entre 2 macroblocs
- minimum d'erreur quadratique entre 2 macroblocs
- ...

# Schéma Global



# Exemple MPEG-1 (I-B-P)

Originale

MPEG-1  
Séquence : IBBPBBPBBPBBPBBPBB

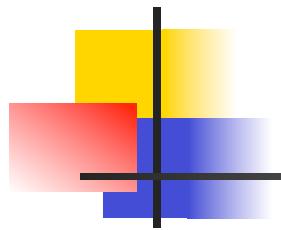


14,5 Mbits/s



1,48 Mbits/s

Facteur 10 environ



# Exemple MPEG-1 (I-P)

Originale



MPEG-1

Séquence : IPPPPPPPPPPPPPPP

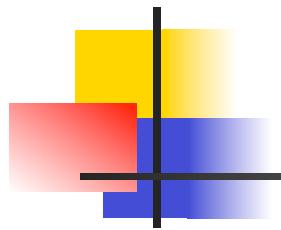


14,5 Mbits/s



1,48 Mbits/s

Facteur 10 environ



# Exemple MPEG-1 (I)

Originale



MPEG-1

Séquence : I => Motion-JPEG

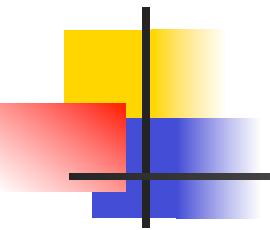


14,5 Mbits/s



1,48 Mbits/s

Facteur 10 environ



# MPEG-2 Vidéo

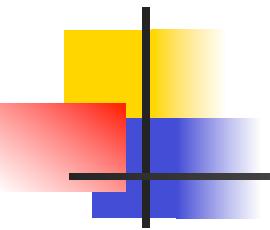
L'encodage MPEG-2 est fondamentalement similaire à l'encodage MPEG-1, avec les frames-I, les frames-P, et les frames-B.

## Les différences essentielles sont :

- La DCT a une dimension de 10x10 à la place de 8x8, pour donner 50% de coefficients en plus, améliorant bien mieux la qualité ;
- Comme MPEG-2 est orienté vers l'émission TV aussi bien que pour les applications CD-ROM : il supporte les images progressives et entrelacées, alors que MPEG-1 ne supporte que les images progressives ;
- D'autres détails mineurs aussi diffèrent entre les deux standards.

Ex. : MPEG2 peut supporter 4 niveaux de résolution :

1. **LL (Low Level)** : niveau bas (352x240), prévu pour les **VCRs** (Video Cassette Recorders), et assurer la compatibilité backward avec MPEG-1.
2. **ML (Main Level)** : niveau principal (720x480), définition normale pour la retransmission NTSC.
3. **H-14 (High-1440)** : niveau haut-1440 (1440x1152), pour la HDTV.
4. **HL (High Level)** : niveau haut (1920x1080), aussi pour la HDTV.



# Vers MPEG-4

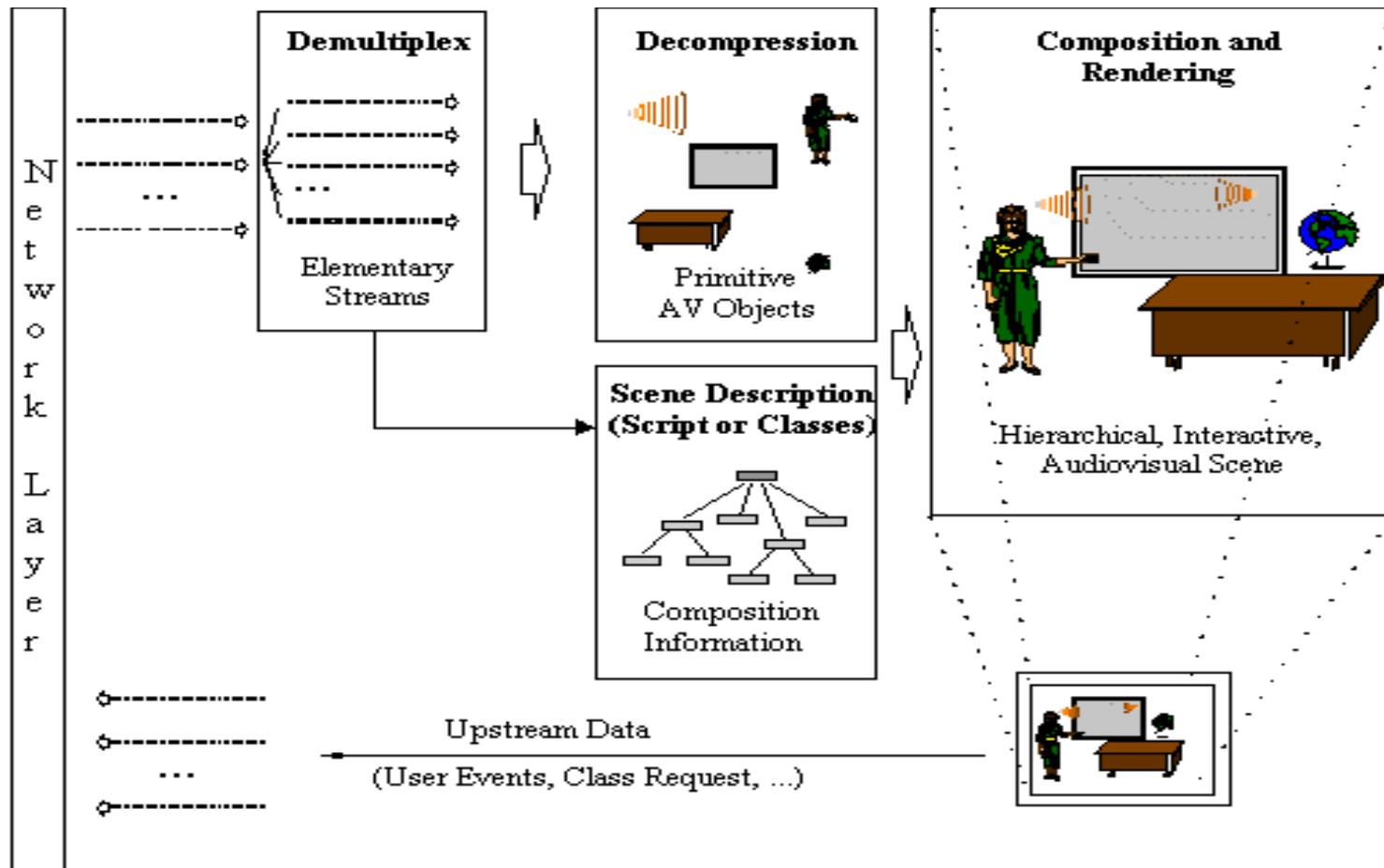
MPEG-1 a constitué une forte base pour la nouvelle norme, mais MPEG-2 aura une incidence plus importante sur le consommateur que cela n'a été le cas pour son prédecesseur.

MPEG-3, que l'on avait conçue à l'origine pour la télévision à haute définition, a été finalement intégrée à la MPEG-2.

**La norme MPEG-4, en cours de normalisation,** est prévue pour des applications comme le télé-enseignement, la télé-surveillance, la visio-conférence ou encore le télé-achat, qui font appel à des méthodes de transmission plus lentes, comme les lignes téléphoniques classiques. De plus MPEG-4 doit être robuste aux erreurs de transmission.

# Vers MPEG-4

## Une approche orientée « objets en mouvement »



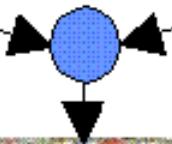
# Vers MPEG-4

## Idée fondamentale : la SEGMENTATION SPATIO-TEMPORELLE

Arrière plan de la scène

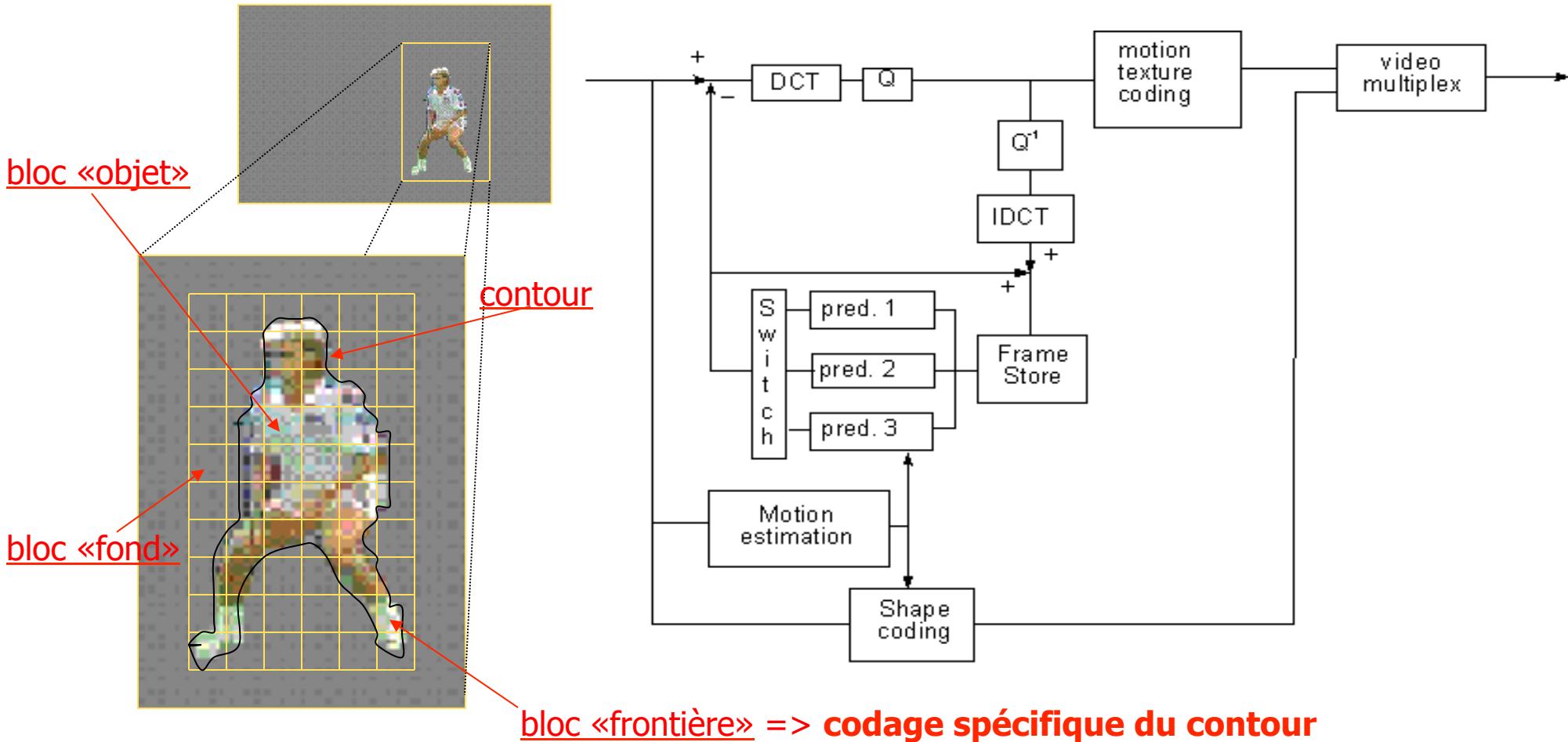


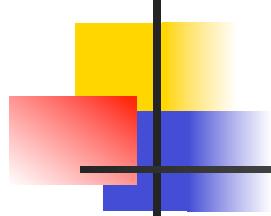
Objet en mouvement (VOP)  
« Video Object Plane »



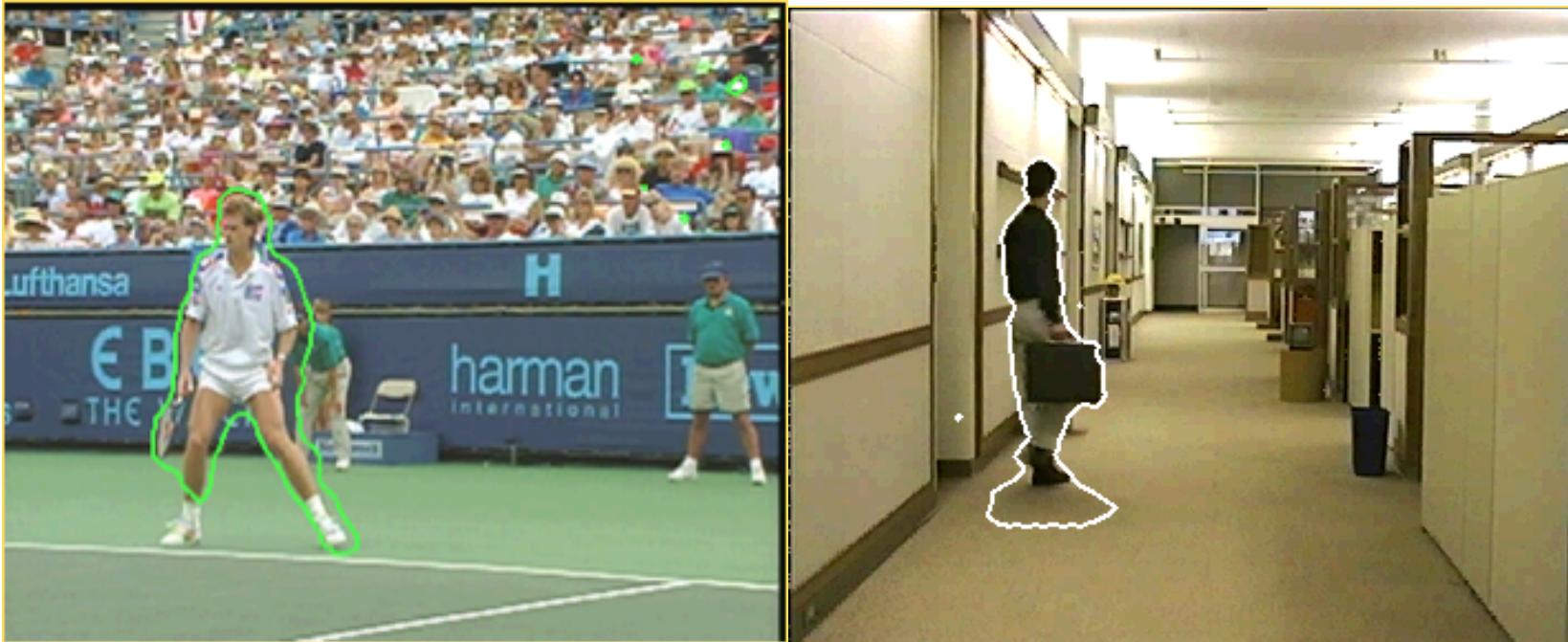
# Vers MPEG-4

## Codage des objets en mouvement (VOP)



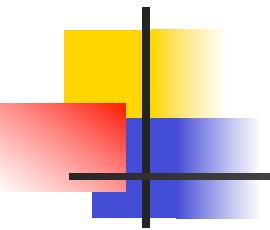


# Exemples De Segmentation



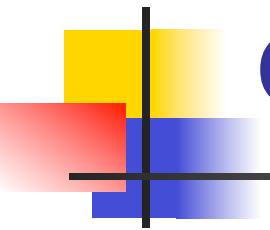
## Problème difficile.

Les éditeurs sont **libres d'implanter la méthode de segmentation** comme bon leur semble pour commercialiser leurs produits.



# Plan du Cours

1. Introduction
2. Outils mathématiques de base
3. Stratégie de compression
4. Transformée
5. Quantification
6. Codage entropique
7. Compression d'images fixes : de JPEG à JPEG2000
8. Compression de vidéos : de MPEG I à MPEG IV
9. Transmission de documents confidentiels et sécurité



# Cas de la vidéo H264

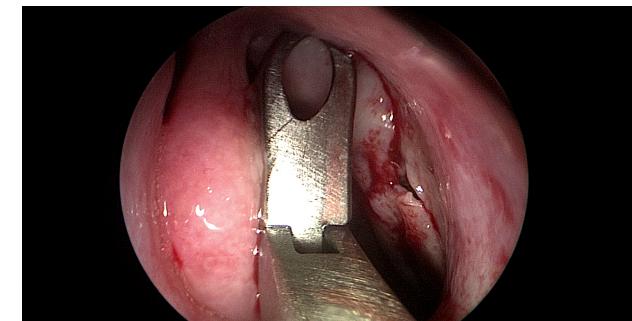
exploiter les **redondances** entre les images successives d'une séquence



1



2

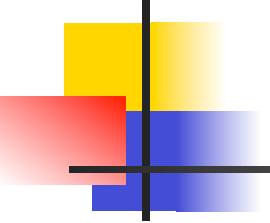


17

coder l'erreur entre deux images qui se ressemblent

2 types d'information dans un flux MPEG :

- image
- mouvement



# Prédictions Temporelles

Il existe 3 types de Codage pour les redondances

## I (Intraframes)

Codage comme une image fixe, nécessaire car la séquence nécessite un commencement.

## P (Predicted Frames)

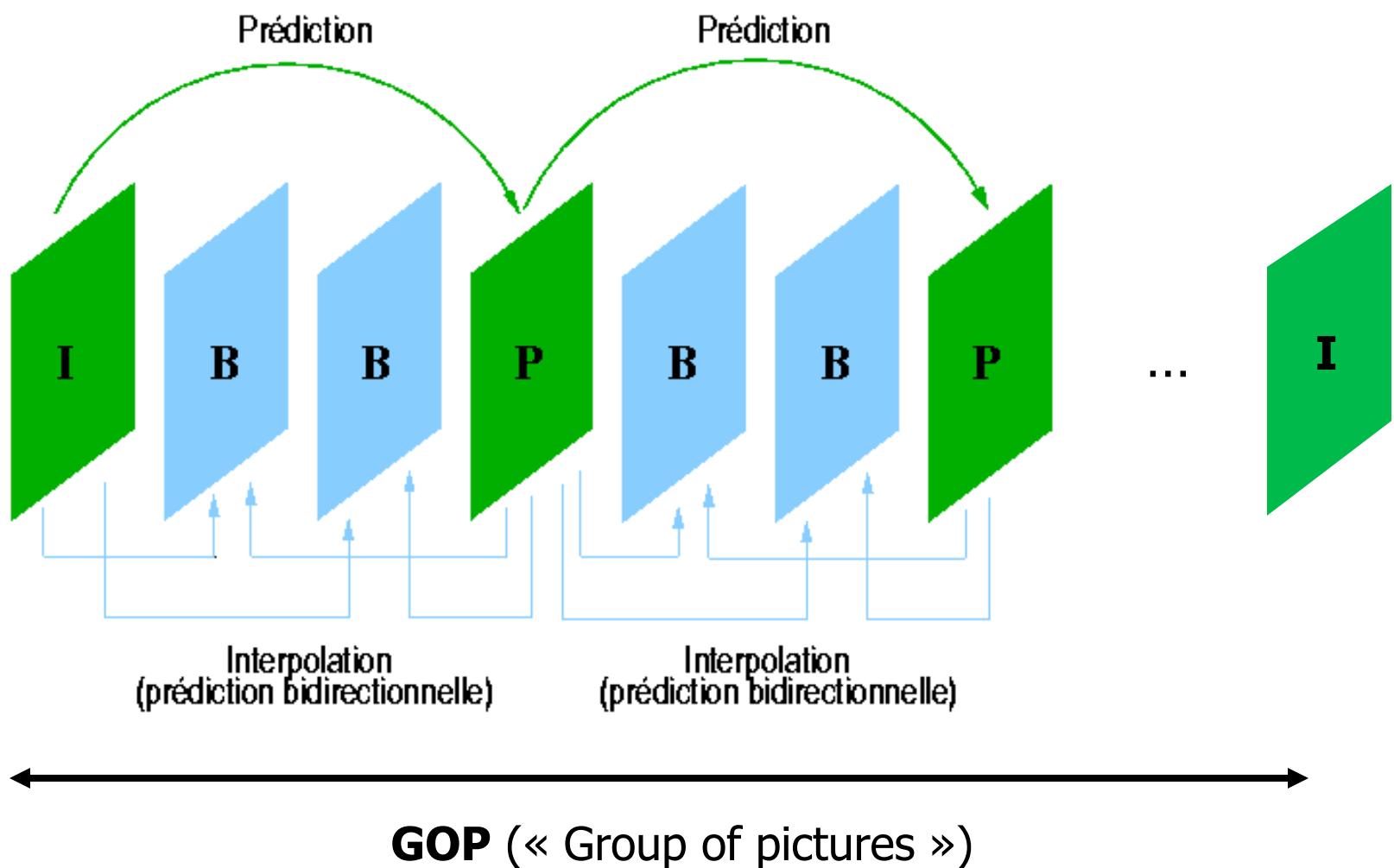
Image calculée à partir de l'image de type P ou I la plus récemment calculée (chaque bloc d'une image peut être codé par des méthodes différentes)

## B (Bidirectionnal Frames)

Image codée à partir des 2 images de type I ou P les plus récentes l'une dans le passé, l'autre dans le futur. 3 calculs pour connaître le meilleur codage possible :

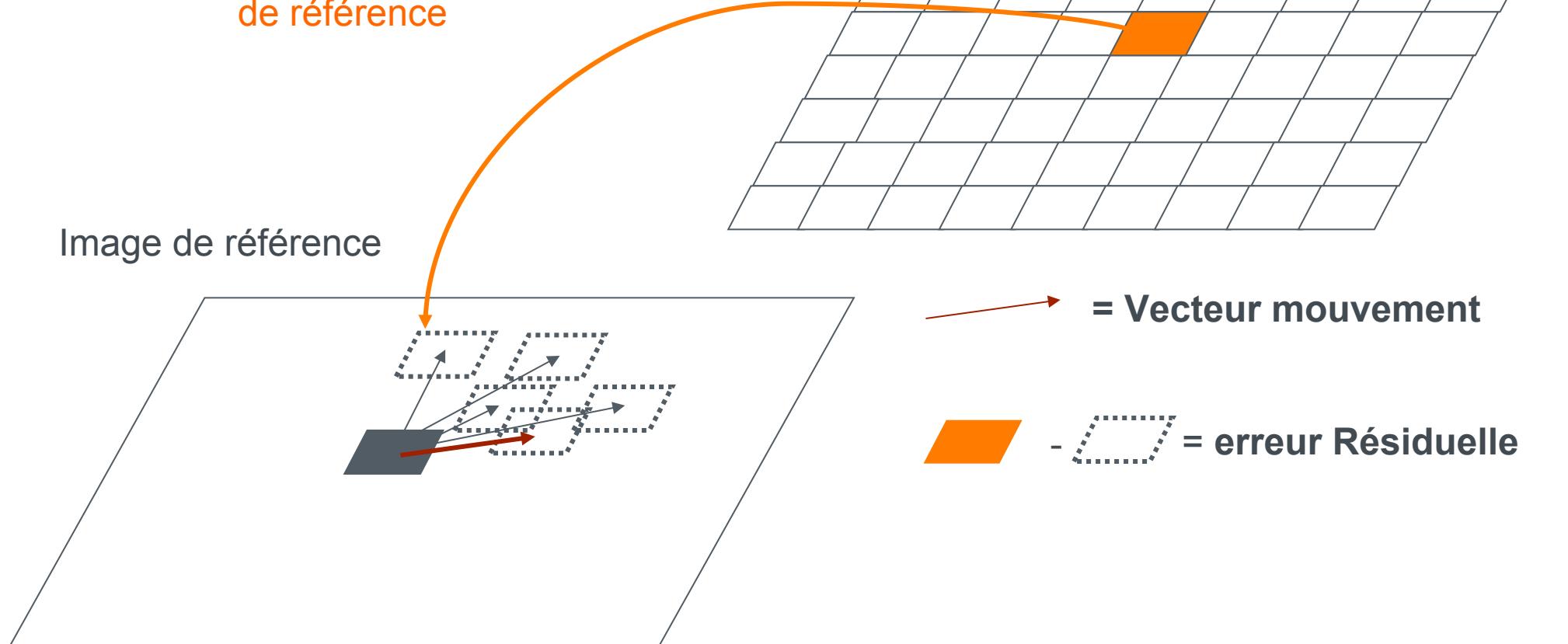
- *à partir de l'image antérieure* ;
- *à partir de l'image future* ;
- *à partir de la moyenne des deux images*.

# Prédictions Temporelles

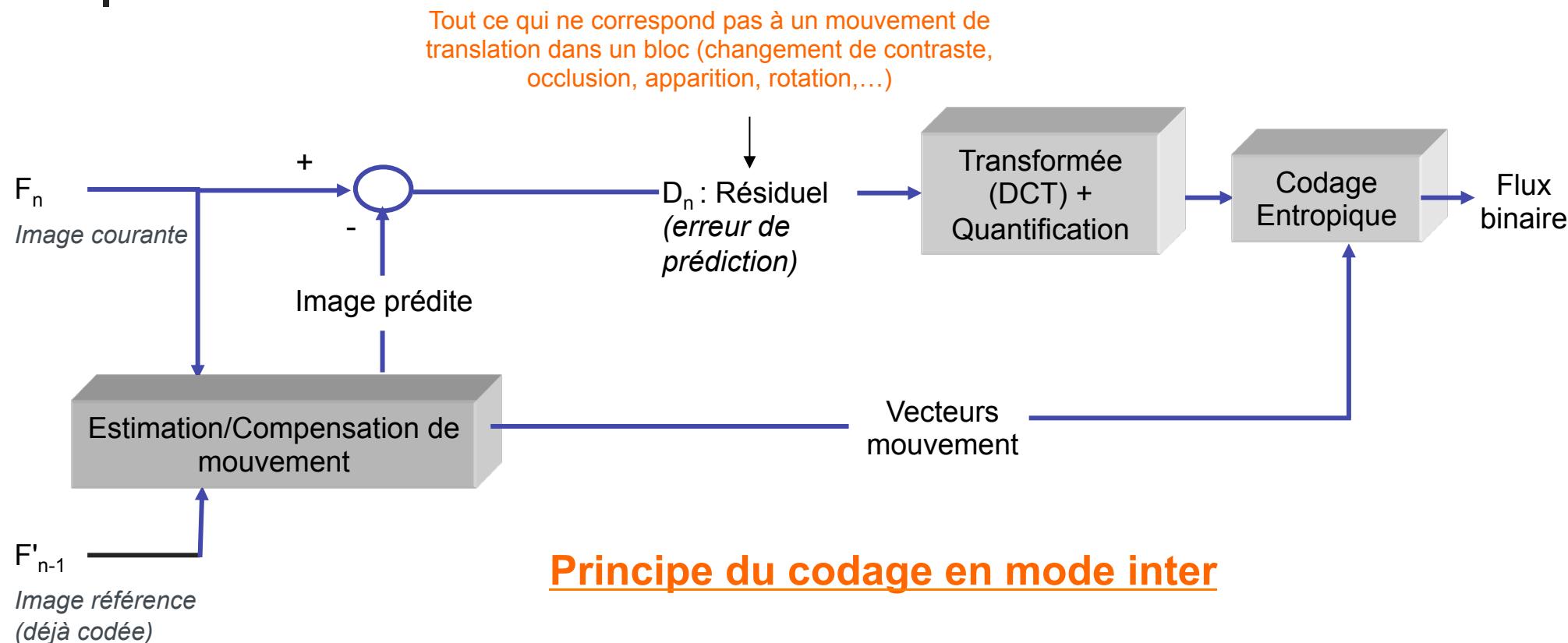


# Estimation/compensation de mouvement (E/CM)

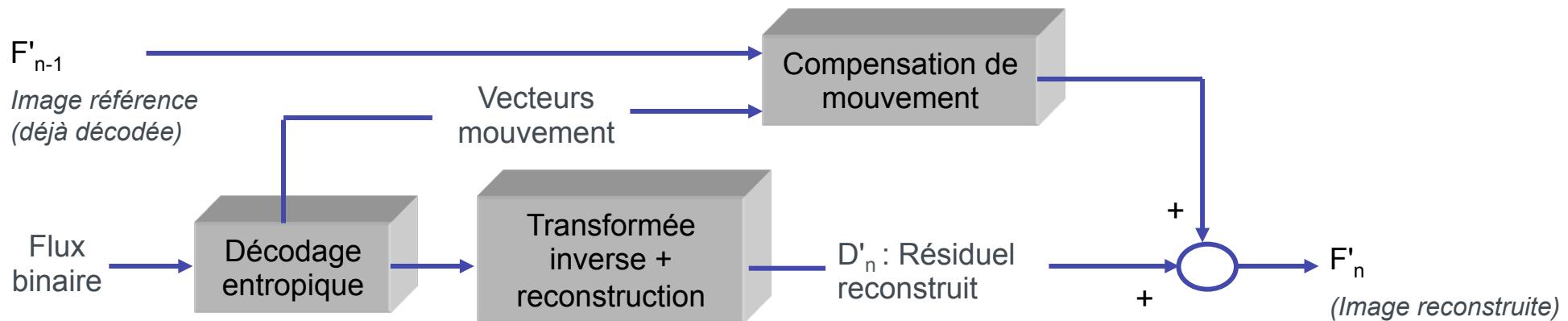
Estimation de mouvement par bloc =  
recherche du bloc similaire dans l'image  
de référence



# Principes de bases des standards



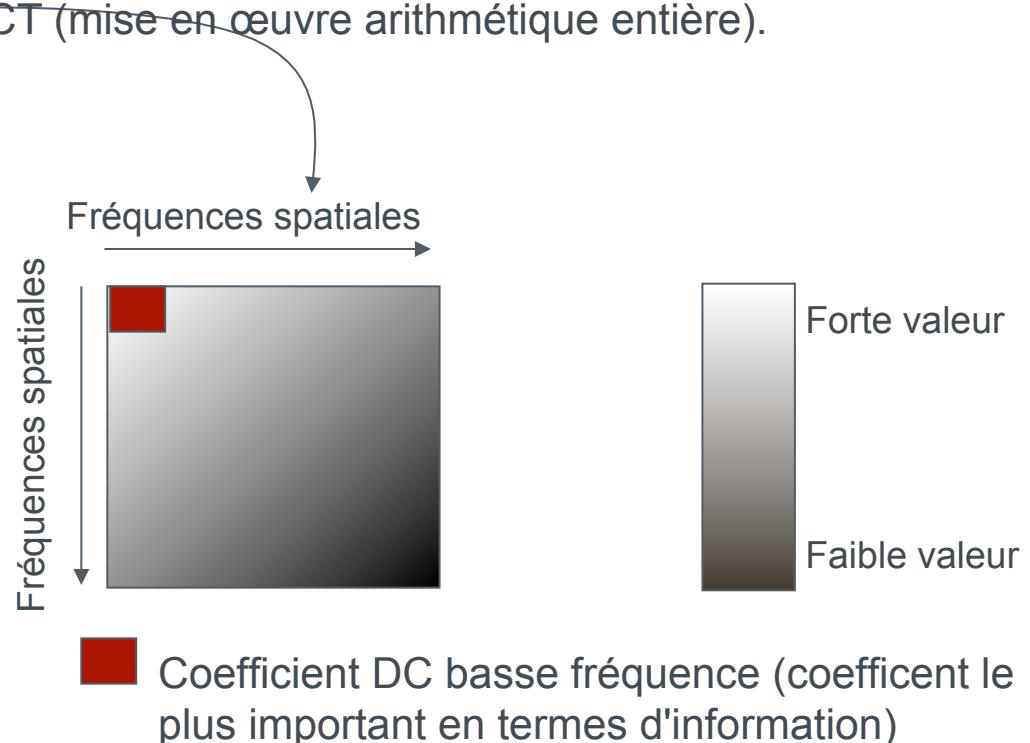
# Estimation/compensation de mouvement : décodage



Décodage en mode inter

# Codage par transformée

- L'étape de transformation est présente en mode intra & en mode inter sur les résiduels
- MPEG1, 2, 4 visual : Discrete Cosinus Transform (DCT)
- H264 : transformée entière dérivée de la DCT (mise en œuvre arithmétique entière).



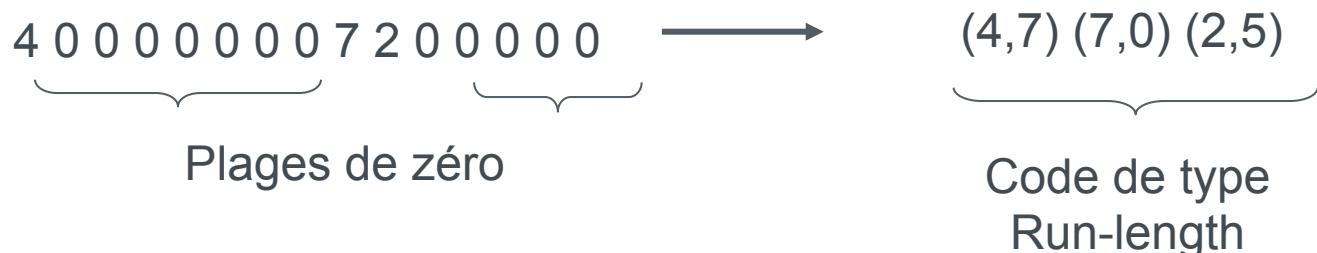
Appliqué sur des blocs de taille 2\*2 (cas particulier H264) à 8\*8

# Codage entropique

Utilisé pour coder : les coefficients de la DCT, les vecteurs mouvements, les éléments de syntaxe (+ loin).

3 grandes familles de codes sont utilisées :

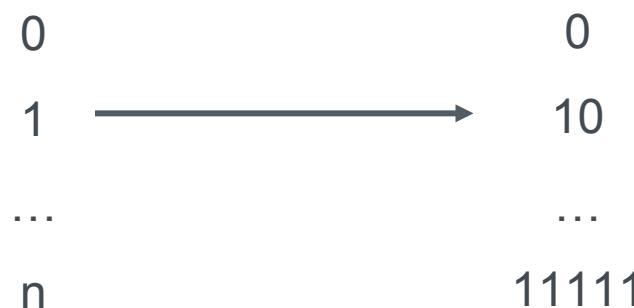
1 Codes Run-level :



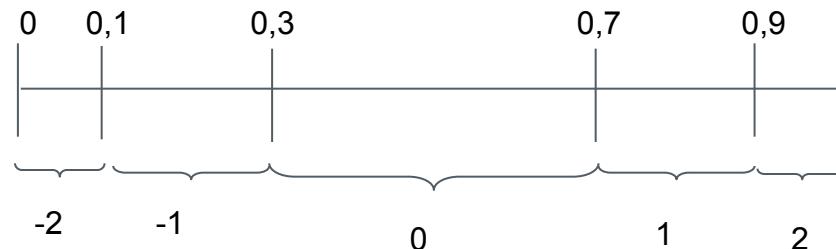
2 Variable Length Coder (VLC) :

Mots à coder (du + probable au – probable) :

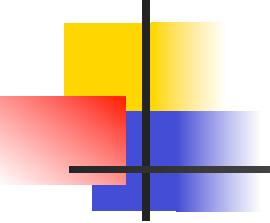
Mots du code :



3 Code arithmétique



Partitionnement de l'intervalle [0, 1] : longueur de chaque partie = proba. d'un symbole à coder.



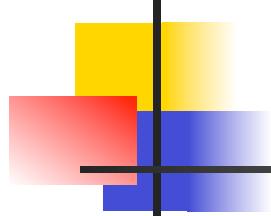
# Codage entropique

VLC :

- + coût de calcul, mise en oeuvre
- Efficacité (ne permet pas de s'approcher de moins d'1 bit/symbole de la borne entropique)

Arithmétique :

- + Efficacité
- Implantation + délicate que les VLC, coût de calcul



# Standard H264 : généralités

MPEG 4 → couvrir un large panel d'applications

H264 → « surpasser » les performances de compression

Conséquences : Perfectionnement de chaque étape de la chaîne de compression

⇒ Multiplication des modes de prédiction, de codage possibles

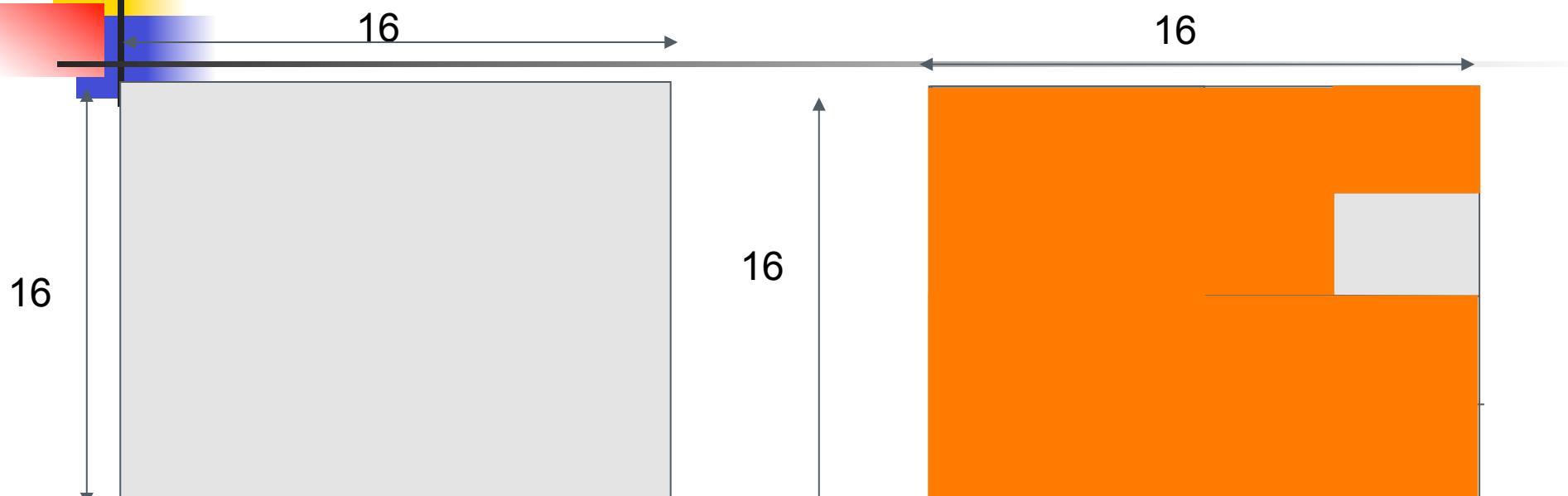
⇒ Complexité et potentielle diversité des implantations du codeur

- L'atome du codeur H264 : ~~frame~~ MACROBLOC.
- Les macroblocs sont regroupés en SLICES.
- Un slice peut contenir des macroblocs codés différemment :
  - I : contient des macroblocs codés en mode intra
  - P : contient des macroblocs codés en mode inter & intra



Mise en garde!

# Estimation Mouvement : Taille des blocs (1)



MPEG 2 : taille fixe

MPEG 4 visual, H264 : + large  
éventail de taille de bloc

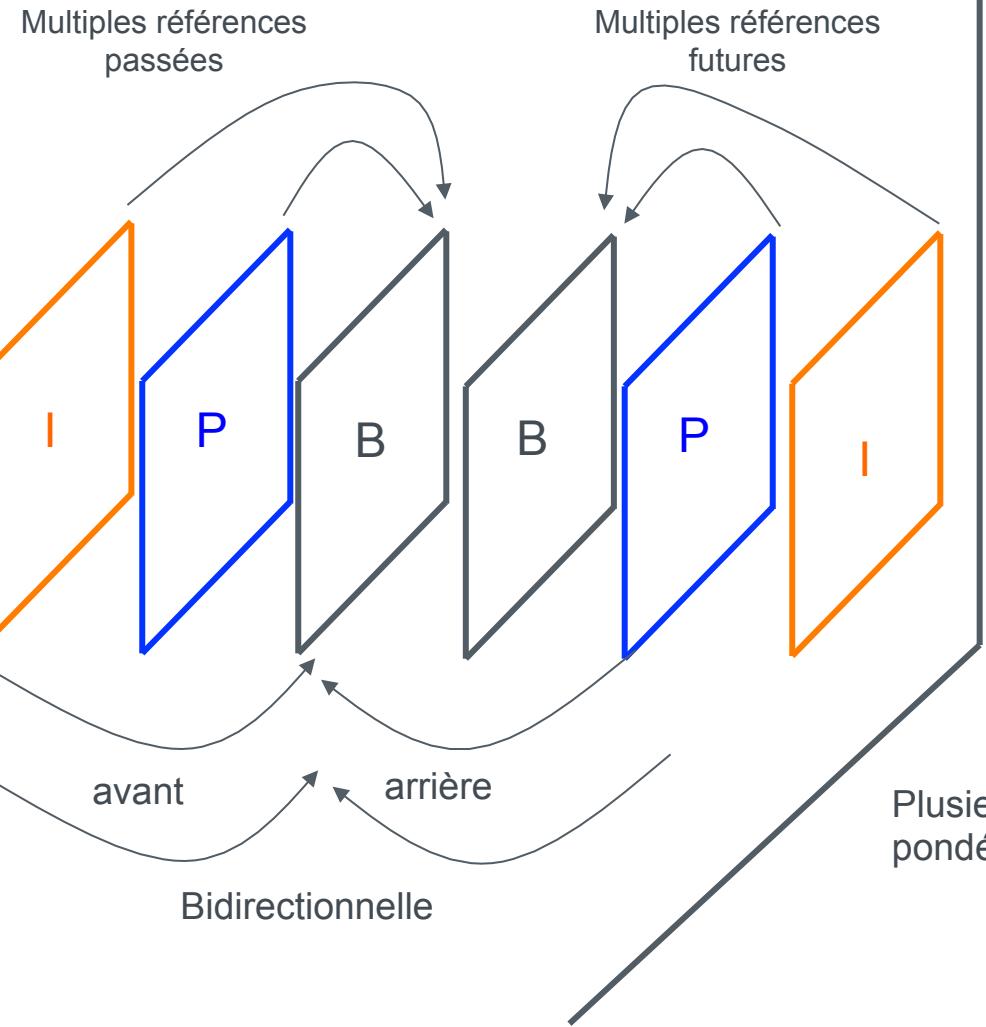
**Idée clef :** Blocs + petits = estimation + précise

⇒ Erreur résiduelle faible

⇒ + de vecteurs à transmettre, + de vecteurs à calculer ( complexité).

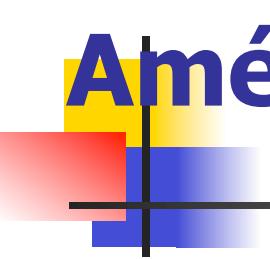
Compromis à faire entre grands et petits blocs : critère choix du mode non spécifié par le standard.

# Améliorations EM : prédiction par rapport à plusieurs références (3)



- **Prédiction avant** : par rapport à une image I (ou P) passée.
- **Prédiction arrière** : par rapport à une image future (I ou P).
- **Références multiples** (H264) : prédiction effectuée à partir de plusieurs références (choix de l'image de référence en fonction de la plus faible erreur résiduelle).
- **Prédiction pondérée** (H264) : permet de pénaliser les références de la liste en fonction de leur éloignement temporel.
- + **Bidirectionnelle (MPEG2, 4, H264)** : par rapport à une image passée et une image future.

Plusieurs modes de bidirectionnels, réf., mult., prediction pondérée : **critère de choix non spécifié par la norme**



# Améliorations EM : Méthodes de block matching

- H264 = explosion des modes possibles pour l'estimation de mouvement (taille bloc, interpolation, type de prédiction et mode de calcul )
- + Techniques de recherche du bloc dans l'image de référence (**block matching**) :
  - Présente au codage et non au décodage.
  - Choix de la technique : **non spécifié par le standard**.
  - **Grand nombre de stratégies possibles** : 3 pas, logarithmique, orthogonale, en croix, spirale,...
- **Estimation de mouvement** = Partie extrêmement coûteuse au codage & Influence sur complexité décodage :
  - Interpolation effectuée au décodage
  - Blocs sautés (blocs éliminés car mouvement nul) => pas d'appel de la transformée.
  - Taille bloc : influence sur le type de transformée utilisée.

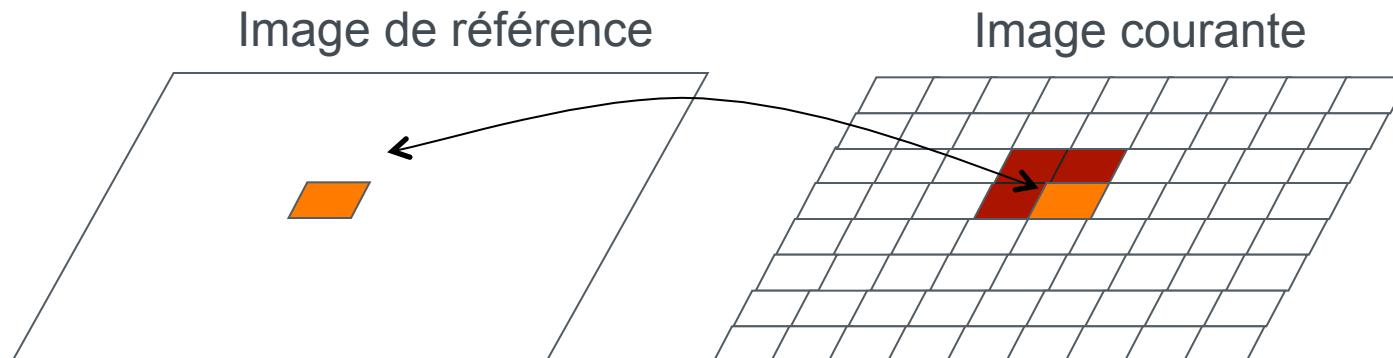
# Prédiction tout azimut

## ▪ Corrélations temporelles :

⇒ Estimation de mouvement

- Existence de fortes corrélations spatiales au sein d'une image :  
⇒ Prédiction dans le codage intra.

La prédiction est généralisée dans H264 i.e. un macroblock peut être codé en mode :

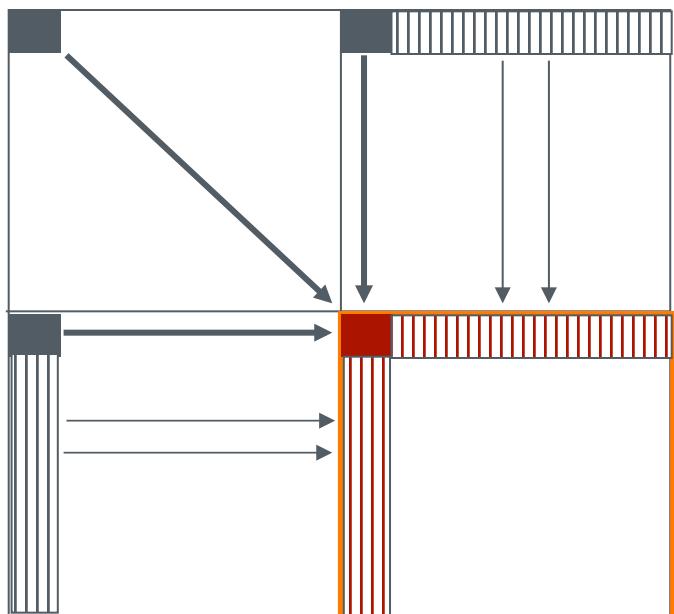


Prédiction **inter** /  
macroblocks image de  
référence (déjà codée)

Prédiction **intra** / macroblocs déjà codés

# Prédiction tout azimut : codage intra

Dans MPEG 4 visual : Prédiction des coefficients DC (et AC) après DCT.

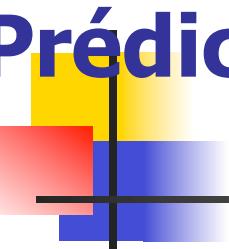


$$\text{■} - \text{■} = \text{Erreurs résiduelles codées}$$

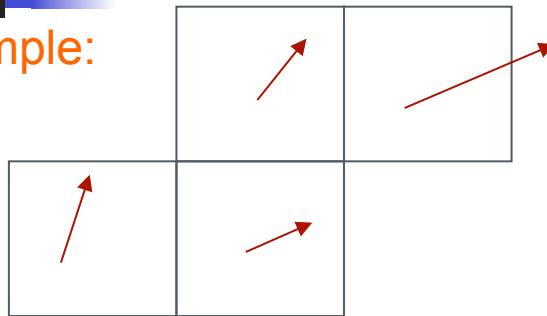
On code la différence entre les coefficients de DCT et le coefficient courant.

(utilisé pour le coefficient DC, optionnel pour les autres)

# Prédiction tout azimut: vecteurs mouvements!

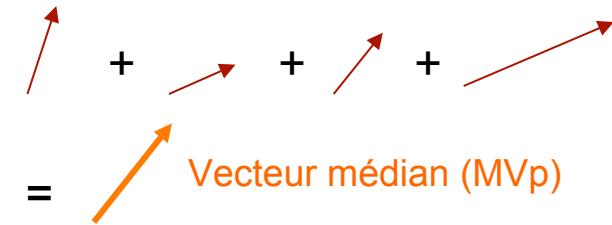


Exemple:



Vecteurs mouvements de blocs voisins

Codage différentiel  
→



Ce qui est codé : différence entre le vecteur médian et vecteur courant

Plusieurs modes différents (!! ) => critère de décision non spécifié par le standard

Ex. d'un mode de prédiction. On calcule :

1. La moyenne des 4 vecteurs (**MVp**).
2. La différence entre MVp et le vecteur courant, (l'erreur résiduelle).
3. On transmet MVp, le résiduel est **codé**.
4. Transmission du mode de prédiction utilisé.

# Transformées utilisées dans H264

H264 utilise plusieurs (!!!) transformées (utilisations spécifiées par la norme)

## 1. Transformée 4\*4 appliquée sur tous les types de résiduels : (mode intra et inter)

- Transformée dérivée de la DCT
- Transformée **"core"** entière (arithmétique entière)
- Implantation à l'aide uniquement d'additions et de décalages.

DCT d'un bloc X :

$$Y = AxA^T =$$

a	a	a	a
b	c	-c	-b
a	-a	-a	a
c	-b	b	-c

X

a	b	a	c
a	c	-a	-b
a	-c	-a	b
a	-b	a	-c

Avec a, b et c  
les coefficients  
(réels) de la  
DCT

# Transformées utilisées dans H264

Transformée  $4 \times 4$  appliquée sur tous les résiduels (intra & inter) :

$Y =$

1	1	1	1
2	1	-1	-2
1	-1	-1	1
1	-2	2	-1

$\times$

1	2	1	1
1	1	-1	-2
1	-1	-1	2
1	-2	1	-1

transformée "core" :  
coefficients entiers



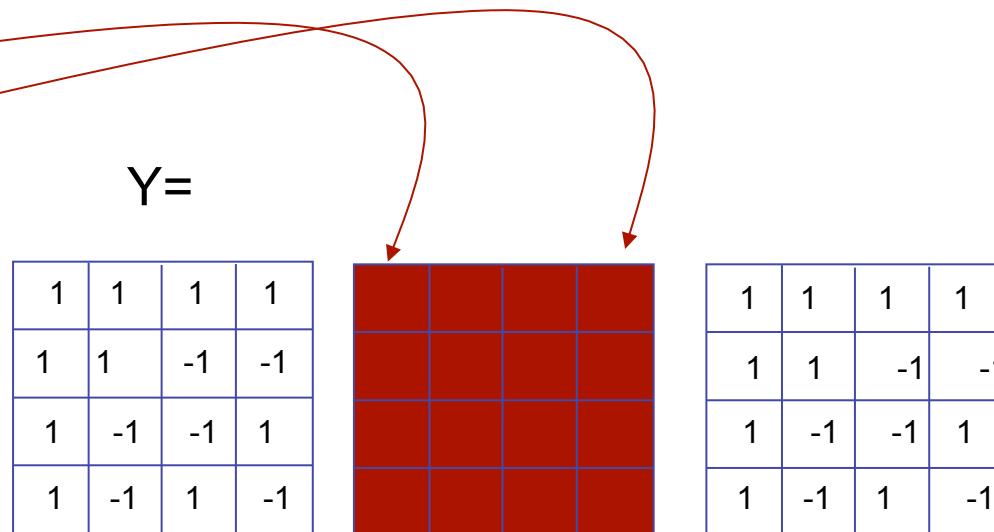
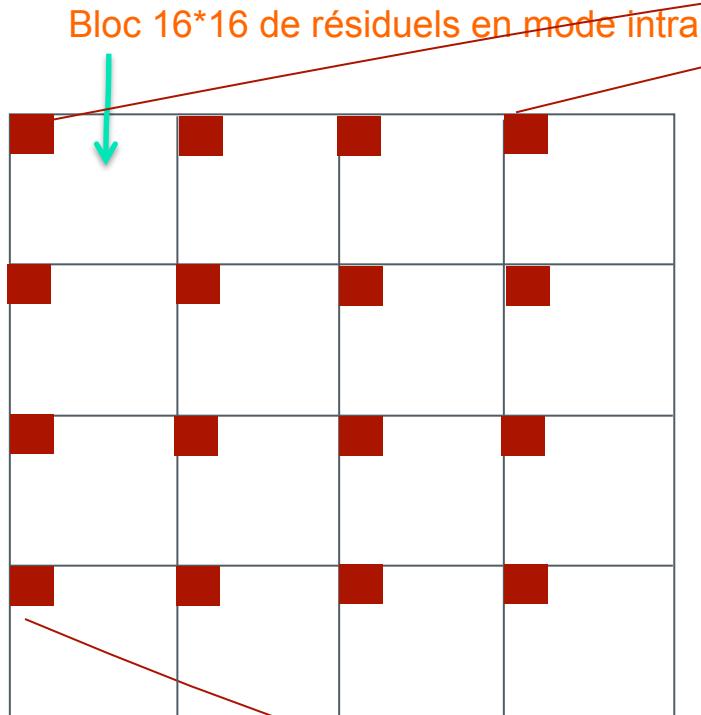
Matrice de scaling (multiplication  
terme à terme)

$a^2$	$ab/2$	$a^2$	$ab/2$
$ab/2$	$b^2/4$	$ab/2$	$b^2/4$
$a^2$	$ab/2$	$a^2$	$ab/2$
$ab/2$	$b^2/4$	$ab/2$	$b^2/4$

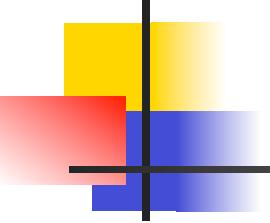
- Les multiplications matricielles ne nécessitent que des opérations d'arithmétique entière.
- Renormalisation (scaling) intégrée dans l'étape de quantification

# Transformées utilisées dans H264

Pour les résiduels intra issus d'une prédiction par bloc 16\*16 :  
Transformée de *Hadamard* 4\*4 sur les coefficients DC (basse fréquence)

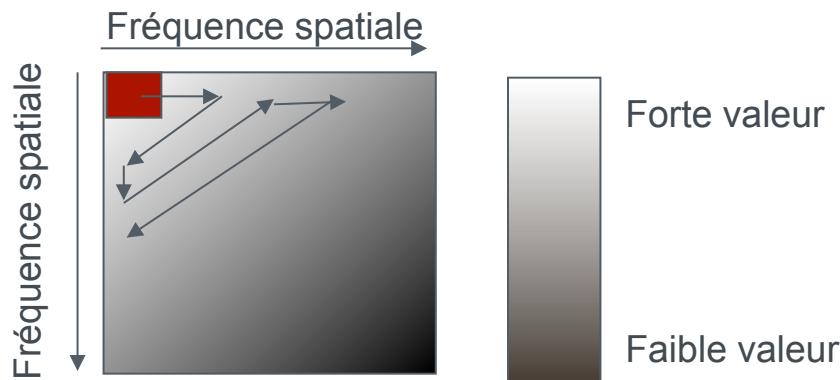


/2



# Codage entropique

- Données résiduelles après transformation : codage adapté au contexte (CAVLC et CABAC)
  - Exploite le caractère creux des coefficients (bcp de zéros et faibles valeurs)
  - Exploite corrélations entre coefficients : coefficients parcourus en zigzag => **valeurs non nulles au début puis plages de 0.**



Numéro symbole      Mot code

Numéro symbole	Mot code
0	1
1	010
2	011
3	00100
4	00100
5	00110
6	00111
7	0001001

- NB : D'autres données sont codées : **Exponentiel-Golomb**
  - Mode de prédiction pour chaque bloc
  - Paramètres de quantification
  - Numéro image référence
  - **Erreur de prédiction des vecteurs mouvement**



Codeur Exp-Golomb  
d'ordre 0

# Codage exponentiel- Golomb

Algorithme pour un entier positif (code exponentiel-Golomb d'ordre k) :

- 1- Prendre le nombre en binaire hormis les k derniers chiffres et lui ajouter 1. Notez-le.
- 2- Compter le nombre de bits écrits et soustraire un.
- 3- Ecrire un nouveau nombre binaire en commençant par le nombre de 0 déterminé dans l'étape précédente.
- 4- Concaténer les k derniers bits en binaire soit le nombre binaire noté dans la première étape.

Intérêt : coder les symboles fréquents avec un code proche du 0 et ne pas avoir à transmettre de table (Golomb connu au décodeur)

Numéro symbole      Mot code

0	1
1	010
2	011
3	00100
4	00100
5	00110
6	00111
7	0001001

Codeur Exp-Golomb  
d'ordre 0

# Context adaptated Variable Length Coder (CAVLC)

0	3	-1	0
0	-1	1	0
1	0	0	0
0	0	0	0

Zigzag → 0 3 0 1 -1 -1 0 1 0 0 0 0 0 0 0 0

Pour comprendre : [http://iris.ee.iisc.ernet.in/web/Courses/mm\\_2012/pdf/CAVLC\\_Example.pdf](http://iris.ee.iisc.ernet.in/web/Courses/mm_2012/pdf/CAVLC_Example.pdf)

**Etape 1 :** Codage du nombre de coeff. non nuls et du nombre de 1 (redondance de +1)

- NbNonNul = 5
  - NbUn = 3 (nombre max)
  - Infos codées à l'aide d'une LUT. Choix parmi plusieurs LUT possibles dépendant des blocs voisins. ( "context adaptative")
- ⇒ 0000100

# Context adaptated Variable Length Coder (CAVLC)

Etape 2 : Codage du signe des coefficients dont la valeur absolue = 1

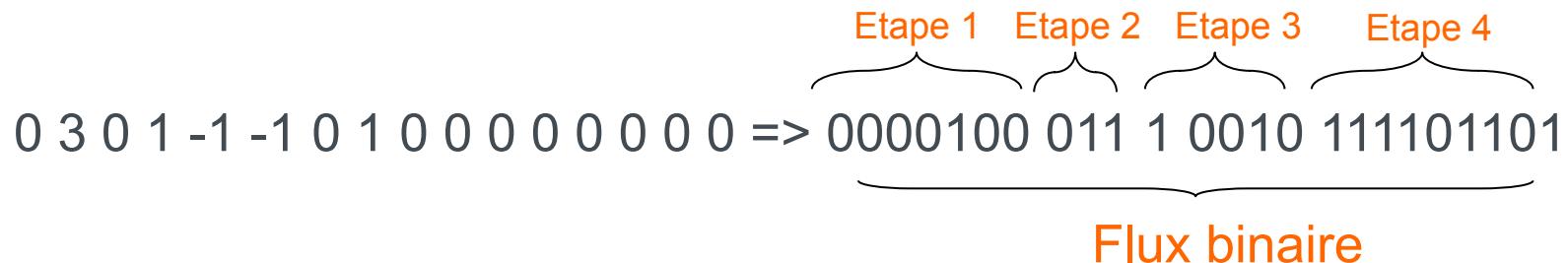
- + -> bit 0
- - -> bit 1
- => 0 1 1 pour la séquence (1 -1 -1)

Etape 3 : Codage du niveau de chaque coefficient non nul

- +1 codé 1
- +3 codé 001,0

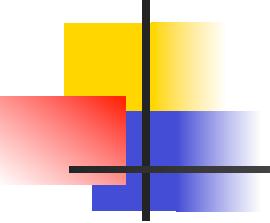
Etape 4 : Codage des plages de zéro

TotalZeros	3	111
run_before(4)	ZerosLeft=3; run_before=1	10
run_before(3)	ZerosLeft=2; run_before=0	1
run_before(2)	ZerosLeft=2; run_before=0	1
run_before(1)	ZerosLeft=2; run_before=1	01
run_before(0)	ZerosLeft=1; run_before=1	No code required; last coefficient.



# Context based Adaptative Binary Arithmetic Coding (CABAC) :

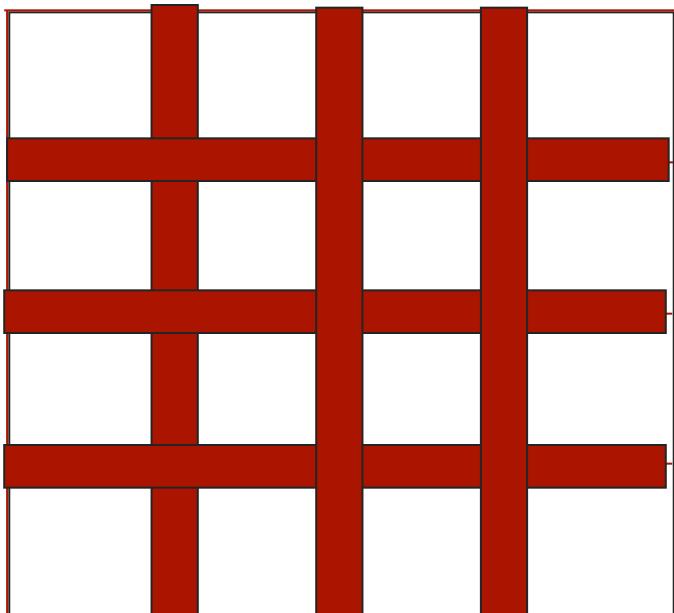
- Alternative de CAVLC (plus performant mais plus coûteux)
  - Absent du profile de base (baseline), *possible* dans les profiles main et high
1. Tous les coefficients sont exprimés dans le système binaire : codage Exp-Golomb
  2. Choix d'un **modèle de contexte** : probabilités des bits 0 et 1
  3. Codage arithmétique à partir des intervalles définis en 2.
  4. Le modèle de contexte est mis à jour.  
**Près de 400 modèles de contexte différents pour coder la syntaxe du flux H264!!**



# Filtre de deblocking

Permet de réduire les effets de bloc dans chaque macroblock

- Appliqué au **codage** : images destinées à être utilisées comme référence.
- Appliqué au **décodage** avant la reconstruction des macroblocks.



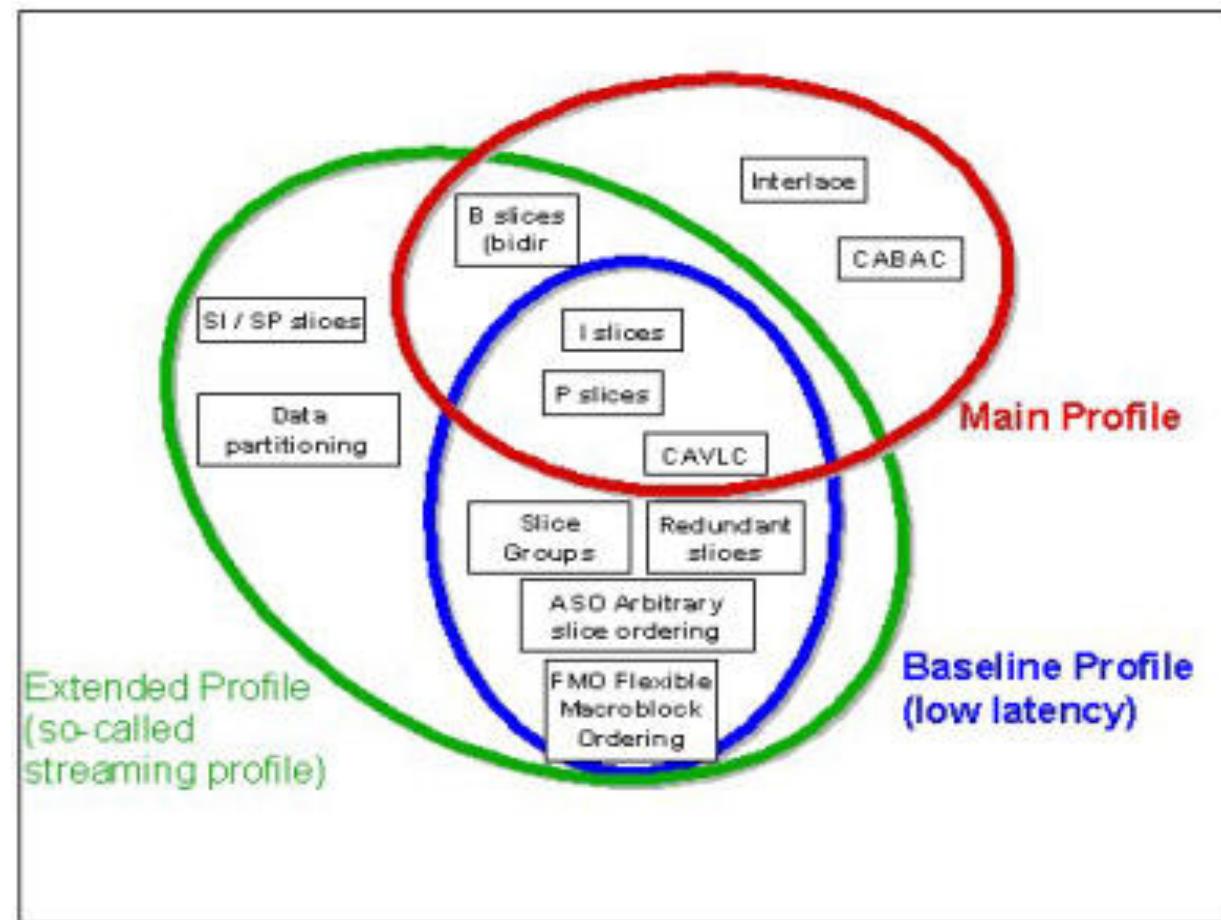
Filtre appliqué sur 3 pixels de part et d'autre de la frontière

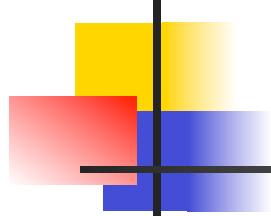
# « Profiles » de H264

4 profiles développés dans le standard H264 :

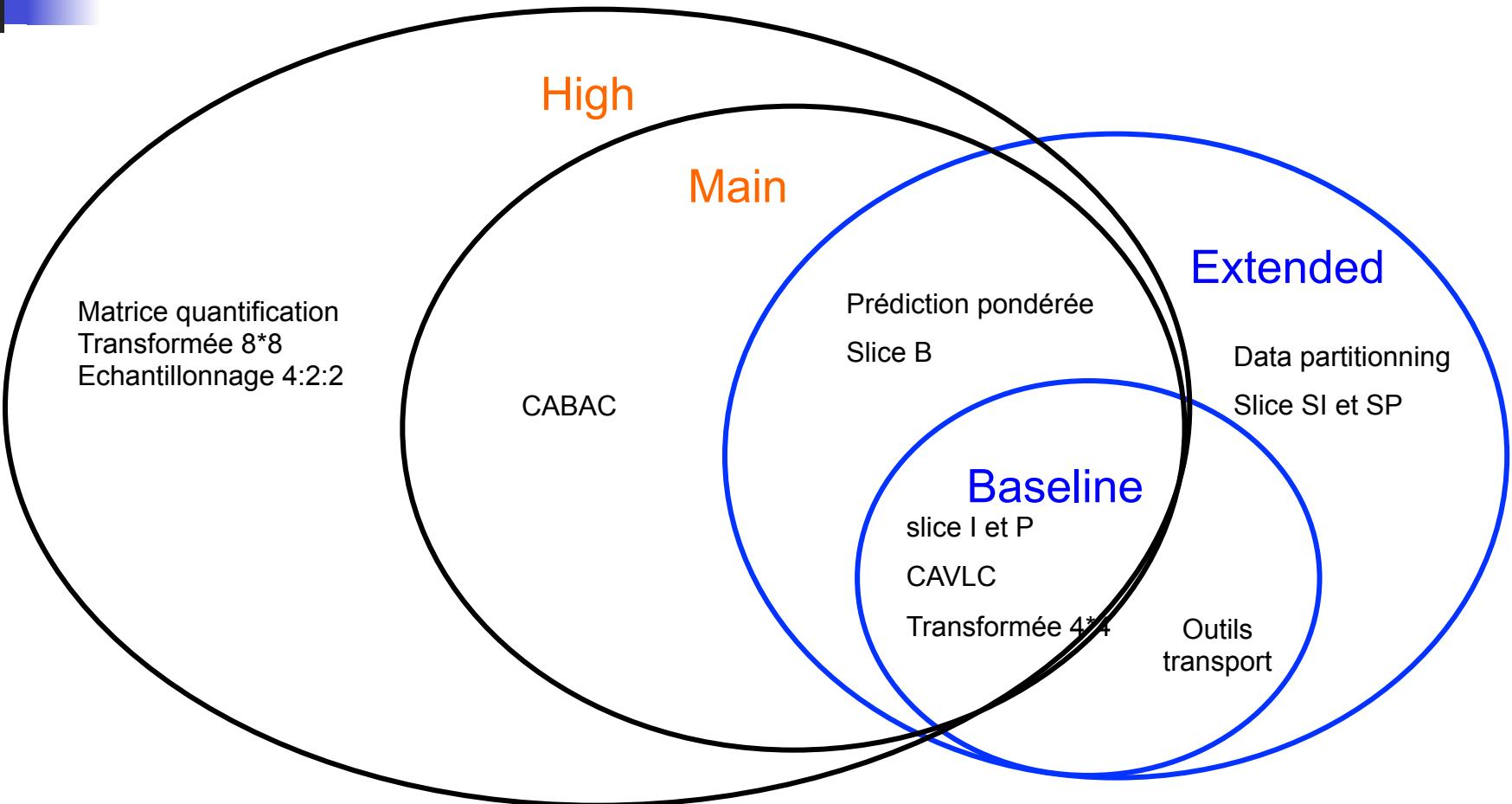
- **Baseline** (visio, comm. Wireless,...)
- **Main** (TV, stockage)
- **Extended** (streaming)
- **High** (TVHD)

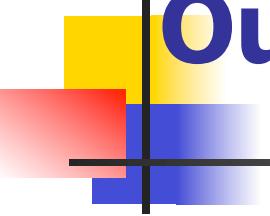
Outils utilisés selon le « profile »





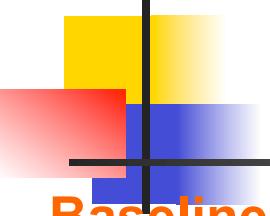
# Profiles de H264





# Outils communs à tous les profiles (H264 core)

- Codage inter : prédiction (P)
- Compensation de mouvement : bloc de taille 16\*16 à 4\*4
- Précision : pixel -> ¼ pixel (interpolation)
- Prédiction intra et vecteurs mouvements
- Transformées Hadamard et DCT modifiée
- VLC Syntaxes et paramètres :
  - Exp-Golomb
  - CAVLC



# « Profiles »

**Baseline** : robustesse du flux compressé face aux **erreurs de transmission** :

- **ASO (Arbitrary Slice order)** :
- **Redondant Slice** : Contient des informations supplémentaires pour reconstruire l'image en cas d'erreur de transmission.
- **Slice Group** : Regroupement de slice pour traiter les vidéos entrelacées, les erreurs de transmission (slices dans un même groupe ne sont pas voisins)

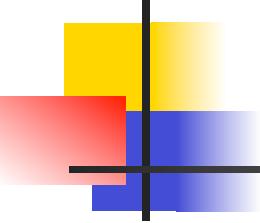
**Extended** : version améliorée du baseline

- Prédiction : Bi-directionnelle et pondérée
- Slices SP et SI : permet de switcher d'un débit à un autre pour s'adapter à la bande passante (internet).
- Data Partitionned slice : outils pour sécuriser la transmission
  - + **Tous les outils mis en œuvre dans le baseline**

*A priori « profile » peu utilisé*

**Main** : outils avancés pour TV et stockage

- **Prédiction** : bi-directionnelle et pondérée (réduction de 10 à 15% du débit).
- **Vidéo entrelacée**
- **Codage** : CAVLC ou CABAC (réduction de 10 à 15% du débit).



# « Profiles »

**Profiles High** : extension ultérieure du standard H264 dédiées à la Haute Définition.

- Permet de traiter les pixels codés sur 10-12 bits
- Echantillonnage 4:2:2 et 4:4:4
- Switching adaptatif des transformées 4\*4 et 8\*8
- Matrices de quantification avec critères perceptuels.
- Codage inter sans perte.
- Transformée couleur additionnelle

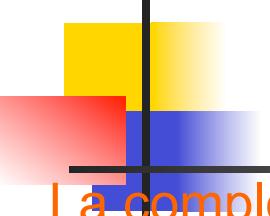
Plusieurs "sous"-profiles :

- **High Profile : DVD et Blue-ray Disc + DVB TVHD**
  - High 10 Profile (Hi10P)
  - High 4:2:2 Profile (Hi422P) : studio
  - High 4:2:2 Profile (Hi422P)
- } Studios de production

# Levels

Level déterminé en fonction des **caractéristiques quantitatives** de la vidéo : résolution écran, fréquence image et débit en ligne.

Numéro Level	Résolution max	Fréq. Max (frame/s)	Débit vidéo max (kbits/s )	Taux comp. (pour du 4:2:0)
1	<b>QCIF (176×144)</b>	<b>15</b>	<b>64</b>	<b>71,28</b>
1,1	CIF (352 × 288)	7,57	192	48
1,2	CIF (352 × 288)	15,15	384	48
1,3	CIF (352 × 288)	30	768	92,16
2	<b>CIF (352 × 288)</b>	<b>30</b>	<b>2000</b>	<b>18,24768</b>
2,1	512×396	25	4000	15,2064
2,2	<b>SDTV (720*576)</b>	<b>12,5</b>	<b>4000</b>	<b>15,552</b>
3	<b>SDTV (720*576)</b>	<b>25</b>	<b>10000</b>	<b>12,4416</b>
3,1	ATV (1280×720)	30	14000	23,69828571
3,2	SXVGA (1280×1024)	42,18	20000	33,1776
4	<b>UXVGA (1600×1280)</b>	<b>30</b>	<b>20000</b>	<b>37,748736</b>
4,1	UXVGA (1600×1280)	30	50000	15,0994944
4,2	UXVGA (1600×1280)	60	50000	30,1989888
5	<b>2760×2048</b>	<b>26,71</b>	<b>135000</b>	<b>13,4217728</b>
5,1	4608×2048	26,66	240000	12,582912



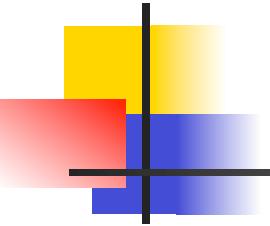
# Complexité et configurations

La complexité du décodage dépend de plusieurs paramètres :

- **Level** : Résolution, fréquence image. Débit
- **Profile**
- **Options sélectionnées** par le codeur. Exemple: profile main avec ou sans CABAC, avec ou sans image B.
- Interpolation, décodage entropique, transformée inverse, filtre deblocking
- Caractéristiques de la vidéo & de l'encodeur.

La complexité du codage dépend de surcroît des briques algorithmiques non spécifiées par le standard :

- Estimation de mouvement
- **Mode décision** (choix taille des blocs, type de prédition, interpolation, etc...) *totalement libre*
- Transformée, codage entropique, filtre deblocking
- **Allocation des débits** (appel répété du codeur => explosion de la complexité ) :
  - Débit variable (pour une qualité constante).
  - Débit constant pour s'adapter à la bande passante.



# Complexité et performances

Seul le décodage est spécifié dans la norme

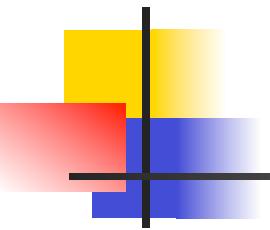
⇒ Grand espace de liberté pour l'implémentation de la norme

⇒ Exemples de recherche sur H264

- Fast multiframe motion estimation by...
- Low delay rate control for Real Time H264 video coding

⇒ Exemple de marge de progression : MPEG 2

- A qualité = le débit a été divisé par 2 entre 1995 et 2001
- On parle parfois de 5 générations d'encodeurs



# Conclusion et perspectives



## Enjeux industriels très importants

Imagerie satellitaire, médicale, vidéoconférence,...



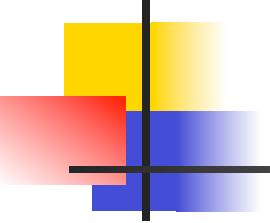
## Problèmes

- Compromis : taux de compression / qualité / complexité
- Normalisation
- Erreurs de transmission
- Coexistence avec d'autres traitements (tatouage,...)



## Perspectives

- Taux de compression élevés (*bas débits*)
- Compression / Classification (*Zones d'intérêt*)
- Prise en compte de la capacité des futurs réseaux
  - ° *Codage combiné Source / canal*
  - ° « *Scalabilité* »



# Bibliographie

- K. Sayood, « Introduction to data compression », Second Edition, *Morgan Kaufman Publishers*, 2000.
- A. Mostefaoui, F. Prêteux, V. Lecuire, JM. Moureaux, « Gestion des données Multimédias », *Traité IC2*, Editions Hermès Lavoisier, 2004.
- A. Gersho, R.M. Gray, « Vector Quantization and Signal Compression», *Kluwer Academic Publishers*, 1992.
- N. Moreau, « Techniques de Compression des Signaux », *Ed. Masson*, 1995.
- M. Antonini, T. Gaidon, M. Barlaud, P. Mathieu, « Wavelet Transform and Image Coding », *Wavelets in Image Communication*, *Ed. Elsevier*, 1994.