

Compression des images

Représentation fréquentielle

F. Dupont



SOMMAIRE

- ✗ Introduction
- ✗ Rappels de théorie de l'information
- ✗ Méthodes de codage sans perte
- ✗ Compression avec pertes
 - + Réduction de la redondance spatiale
 - ✗ Méthodes spatiales
 - ✗ Méthodes par transformées
 - ✗ Méthodes fractales
 - + Réduction de la redondance couleur
 - + Quantification
 - ✗ Quantification scalaire
 - ✗ Quantification vectorielle (QV)
- ✗ Standard JPEG
- ✗ JPEG2000

INTRODUCTION

✗ Image numérique : une première compression

Image numérique obtenue à partir d'une image physique

- ✗ échantillonnage dans les 2 directions spatiales
- ✗ quantification de l'intensité lumineuse par des niveaux de gris (ou de couleur)
 - codage des valeurs quantifiées par des mots binaires de longueur fixe

✗ Besoins en compression des images

+ en transmission

- ✗ Séquences d'images de qualité moyenne (vidéo conférence)
- ✗ Séquences de haute qualité
TVHD : 1280x720 progressif (50 images/s) → ~ 50 Mpixels/s
ou 1920x1080 entrelacé (25 images/s) → ~ 50 Mpixels/s
- ✗ images satellitaires
SPOT : jusqu'à 12000 x 12000 pixels → ~ 140 Mpixels

+ en stockage

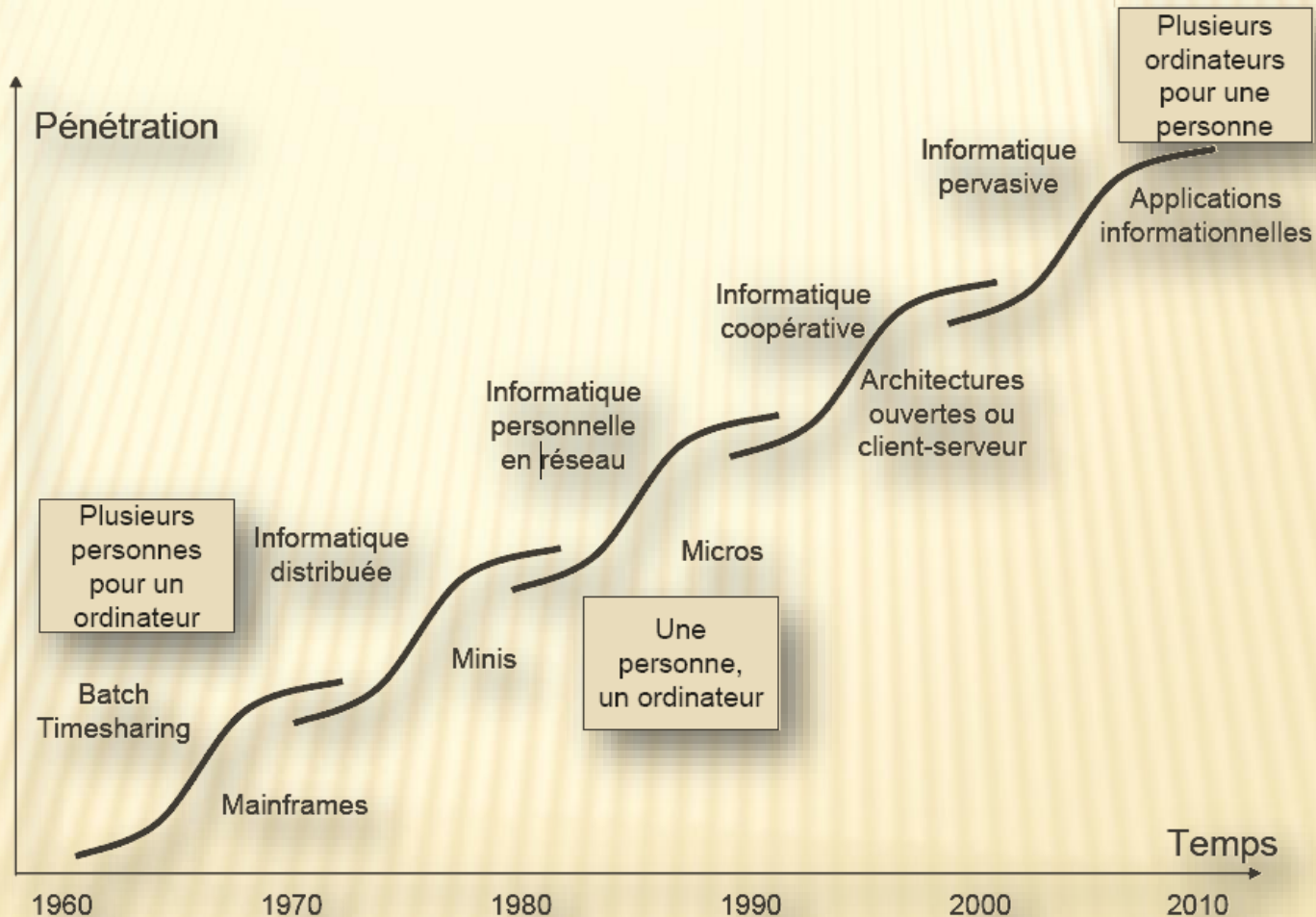
- ✗ une photo numérique : 3 à 10 Mpixels
- ✗ séquences d'images (angiographie 10 images/seconde, 5s → ~ 30 Mpxiels)
- ✗ Documents audiovisuels : l'INA a numérisé 400 000 heures d'archives audiovisuelles (soit 600 millions d'images....)

+ Applications

- ✗ médical,
- ✗ cartographie,
- ✗ multimedia,
- ✗ particuliers...

EVOLUTION

✗ accroissement des moyens de calcul et de stockage



EVOLUTION

✗ Accroissement des demandes

+ Documents multimédia à distribuer et à partager

- ✗ Images
- ✗ Vidéos
- ✗ Audio
- ✗ Documents
- ✗ Objets 3D, ...

+ Besoins

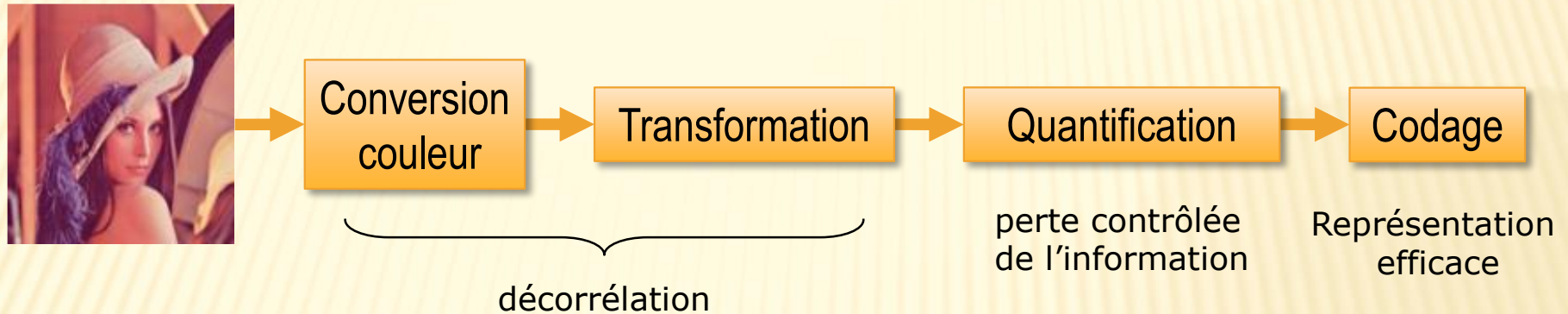
- ✗ Compression
- ✗ Indexation
- ✗ Tatouage
- ✗ Gestion BD

✗ Des évolutions pour :

- + Une amélioration de la qualité
- + De nouvelles fonctionnalités pour de nouveaux usages



UN SCHÉMA GÉNÉRAL



× Deux catégories de méthodes

- + Codage sans perte → réduction de redondance (pas d'étape de quantification)
- + Compression avec pertes → dégradation contrôlée

✗ Compression : analyse préliminaire

- + Quel taux de compression ?
- + Quelle qualité ?
 - ✗ sans perte (sans distorsion)
 - ✗ avec pertes (combien ?)
- + Quelle complexité de codage, de décodage ?

→ Fonction de l'application

CLASSIFICATION DES MÉTHODES

✖ Approche 1

- + méthodes réversibles (sans distorsion ou sans perte)
- + méthodes irréversibles (avec distorsion)

✖ Approche 2

- + méthodes spatiales : pixels
- + méthodes par transformation : coefficients

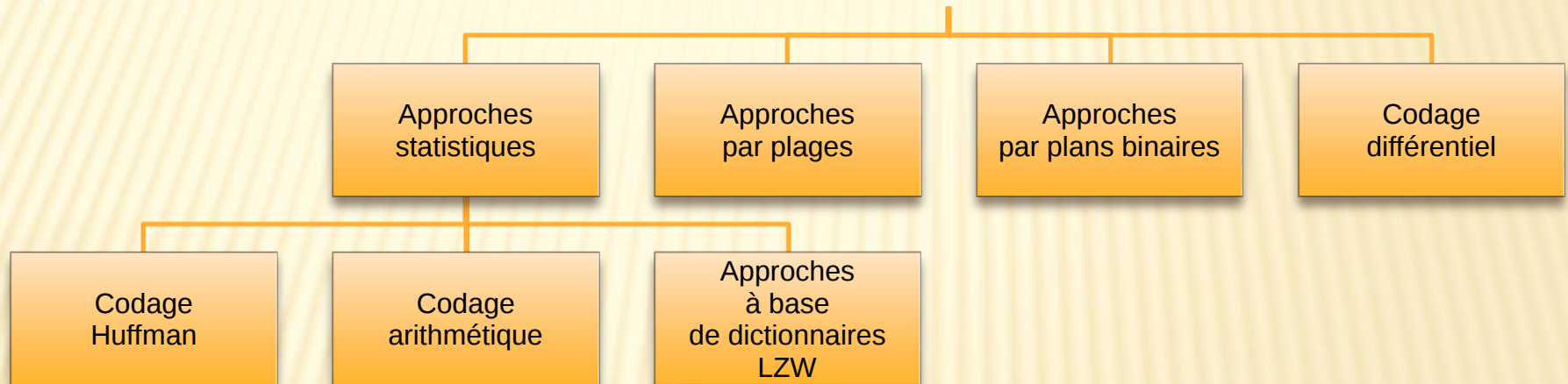
✖ Approche 3 (choix du voisinage)

- + pixel, ligne, bloc, imagerie,...

CLASSIFICATION DES MÉTHODES

✗ Méthodes réversibles (sans distorsion)

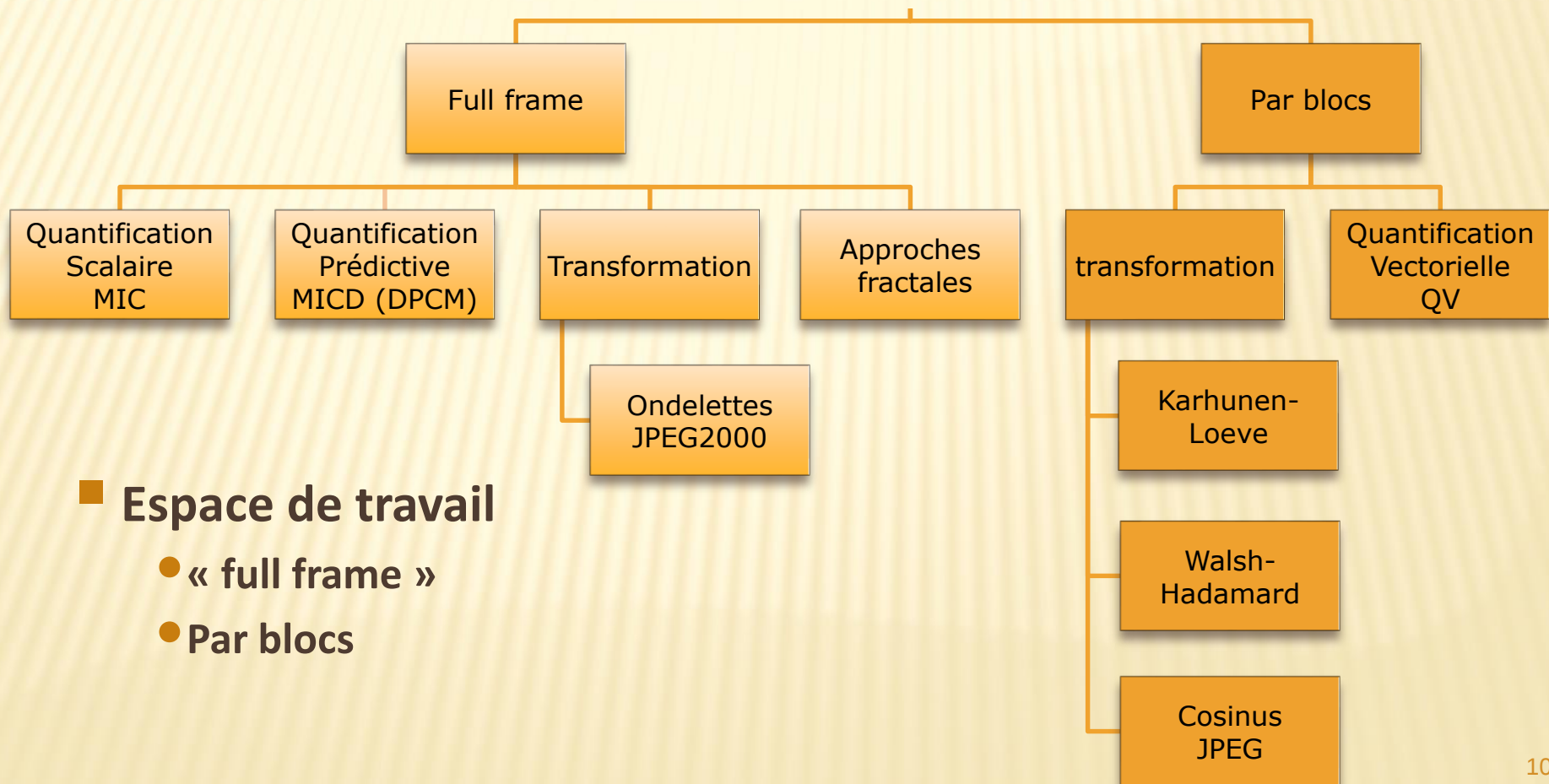
+ taux de compression faible (< 3)



CLASSIFICATION DES MÉTHODES

✗ Méthodes irréversibles (avec perte contrôlée)

+ taux de compression élevé (> 8)



■ Espace de travail

- « full frame »
- Par blocs

CRITÈRES D'ÉVALUATION

✗ Taux de compression

$$T_c = \frac{\text{Taille de l'image originale}}{\text{Taille du fichier comprimé}}$$

✗ Qualité de l'image reconstruite

+ critères quantitatifs

✗ Erreur quadratique moyenne (= mean square error)

$$\text{EQM(MSE)} = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [X(i,j) - \hat{X}(i,j)]^2$$

✗ Rapport signal/bruit

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

CRITÈRES D'ÉVALUATION

+ critères qualitatifs :

- × évaluation subjective
- × Consensus, ROC (experts)

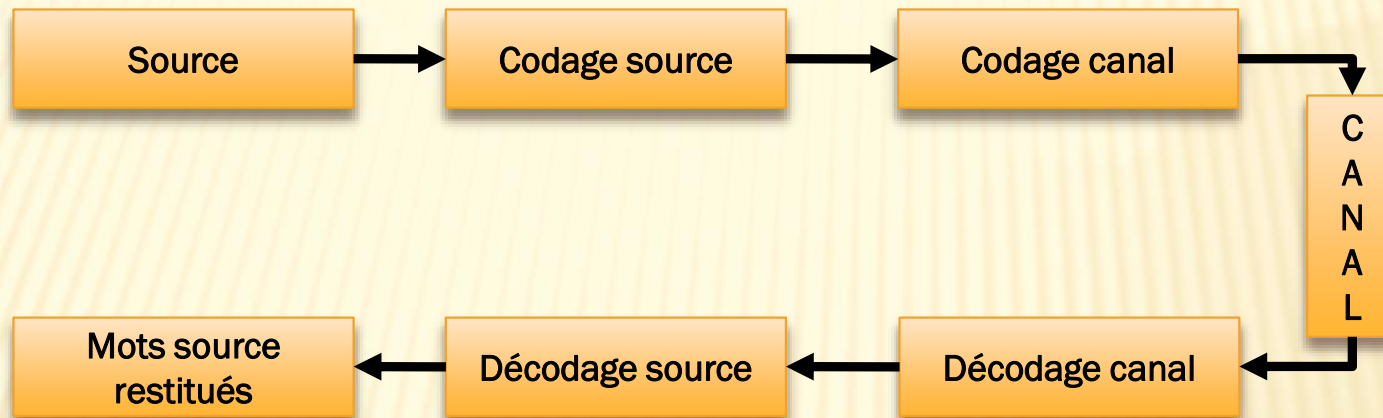
× **Degré de complexité, type d'application**

THÉORIE DE L'INFORMATION : RAPPELS

- ✗ 1920 : premières tentatives de définition de mesure de l'information
- ✗ à partir de 1948 : travaux de Shannon → Théorie de l'information
 - discipline fondamentale qui s'applique dans le domaine des communications
- ✗ développements constants car les exigences actuelles s'orientent toujours vers une augmentation de l'information à stocker ou à transmettre

- ✗ Objectifs
 - + déterminer les limites imposées par les lois de la nature lorsqu'on doit stocker ou transmettre le contenu d'une source (d'information)
 - + proposer des dispositifs permettant d'atteindre ou d'approcher ces limites

✗ Schéma d'une chaîne de communication



- + Le codage de source consiste à éliminer les redondances de la source afin d'en réduire le débit binaire
- + Le codage de canal a un rôle de protection contre les erreurs (dues à la transmission sur le canal) qui est assuré en ajoutant de la redondance (codes correcteurs d'erreurs)
- + Les points de vue codage de source et codage de canal sont donc fondamentalement différents

DÉFINITION DE L'«INFORMATION»

- ✗ Incertitude d'un événement ou self-information ou quantité d'information
- ✗ Une information est un élément de connaissance qui peut être conservé (mis en mémoire), traité (traitement de l'information) ou transmis
- ✗ La difficulté rencontrée pour définir la quantité d'information relative à un événement est liée au caractère subjectif de l'information effectivement apportée par la réalisation de cet événement.
- ✗ La quantité d'information est relative à un contenu statistique : elle est plus grande si on ne s'attend pas à ce que l'évènement se réalise.

DÉFINITION

- ✗ Soit un événement E , une probabilité d'apparition $p(E)$, la quantité d'information $I(E)$ liée à l'apparition de E s'exprime :

$$I(E) = -\log [P(E)]$$

- ✗ Propriétés :

- + $I(E)$ est d'autant plus grande que $p(E)$ est petite ;
- + Si E est toujours réalisé $I(E)=0$;
- + Si E est très rare, $p(E)$ est proche de 0, $I(E)$ tend vers l'infini ;
- + $p(E) \leq 1$ donc $I(E) \geq 0$.

- ✗ **Logarithme**

- + népérien pour les signaux continus
- + de base 2 pour les signaux discrets
(l'unité est le bit, abréviation de binary unit).

× Exemples

- + Un événement qui a une chance sur 2 de se produire conduit à une quantité d'information de

$$-\log_2 (1/2) = 1 \text{ bit}$$

- + Une source peut délivrer 8 événements équiprobables :

$$I_{\text{total}} = -8 * \log_2 (1/8) = 24 \text{ bits}$$

× Interprétation

$I(E)$ peut être interprétée :

- × *a priori*, par l'incertitude qui règne sur la réalisation de E;
- × *a posteriori*, par l'information apportée par la réalisation de E.

ENTROPIE

✗ Entropie d'une variable aléatoire discrète

- + Soit X une variable aléatoire à valeurs dans $\{x_1, x_2, \dots, x_n\}$ telle que $p_i = p(X = x_i)$ pour i de 1 à n
- + L'entropie de X notée $H(X)$ est la moyenne des incertitudes calculée sur les événements $\{X = x_i\}$

$$H(X) = - \sum_{i=1}^{i=n} p_i \log p_i$$

- ✗ L'entropie d'une source discrète est donc la quantité moyenne d'information par symbole et est exprimée en bits par symbole.

ENTROPIE, DÉBIT

✗ Remarques

- + $H(X)$ dépend de la loi de probabilité de X mais n'est pas fonction des valeurs prises par X
- + $H(X)$ correspond à l'espérance mathématique de la variable aléatoire incertitude $I(X)$ définie par
$$I(X) = -\log P(X)$$
- + Exprimée en Shannons, $H(X)$ représente le nombre moyen de bits nécessaires à la codification binaire des différentes réalisations de X

- ✗ **Le taux (débit) d'information est le nombre de symboles par unité de temps :**
 $T = n H(X)$ en bits par seconde.

EXEMPLE

✗ On extrait au hasard une carte d'un jeu de 32 cartes

+ On a $H(X) = -32 \times 1/32 \times \log_2 1/32 = \log_2 32 = 5$ Sh.

+ Pour savoir quelle carte a été extraite, on peut demander si sa couleur est rouge ou noire, s'il s'agit d'un cœur ou d'un carreau, etc.
Les réponses à cinq questions peuvent être résumées par cinq bits ('1' pour oui et '0' pour non).

+ Une autre façon de modéliser le problème consiste à attribuer un numéro (de 0 à 31) à chaque carte. L'écriture de ces numéros en base deux requiert $\log_2 32 = \log_2 2^5 = 5$ bits.

PROPRIÉTÉS DE L'ENTROPIE

- ✗ L'entropie d'une variable aléatoire X à n valeurs possibles est maximum et vaut $\log(n)$ lorsque la loi de X est uniforme.

En effet, l'incertitude sur X est la plus grande si toutes les valeurs possibles ont la même probabilité de se réaliser

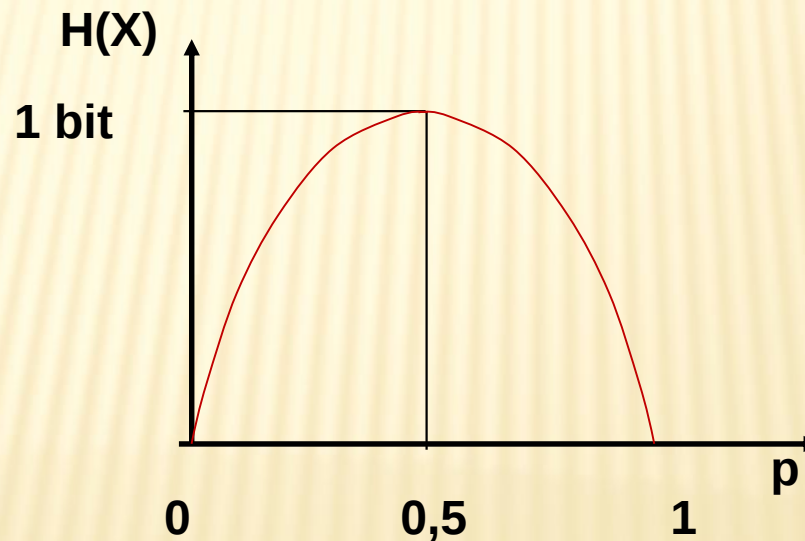
- ✗ L'entropie augmente lorsque le nombre de valeurs possibles augmente.

EXEMPLE

✗ Soit une source binaire à 2 symboles : $\begin{matrix} |X| \\ |P| \end{matrix} = \begin{matrix} |x_1 & x_2| \\ p & 1-p \end{matrix}$

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

- $p=0$ (x_2 toujours réalisé) $\rightarrow H(x)=0$
- $p=1$ (x_1 toujours réalisé) $\rightarrow H(x)=0$
- $p=0,5$ (x_1 et x_2 équiprobables) $\rightarrow H(x)=1$



CODAGE DE SOURCE

- ✗ Soit S une source discrète à N valeurs possibles et d'entropie $H(S) < \log_2 N$

alors l'entropie relative $H_r(S) = H(S)/\log_2 N$ est < 1 ,
donc la redondance $r = 1 - H_r(S)$ est différente de zéro

→ Codage source = élimination de la partie redondante de l'information délivrée par la source = réduction du débit binaire de la source tout en conservant l'information (compression sans perte d'information)

Exemple : soient 256 symboles d'une image à niveaux de gris (8 bits par pixel)

- + en réalité $H \leq H_{\max} = 8$ (exemple : Image Lena → $H = 7,3$ b/p)
- + redondance : $R = H_{\max} - H$ (exemple : $H=5$ bits par pixel → $R=3$ bits par pixel)
→ chaque pixel peut être codé sur 5 bits en moyenne sans perte d'information)

MÉTHODES DE CODAGE SANS PERTE

✗ Généralités sur les codes

- + Deux types de codes peuvent être utilisés :
 - ✗ Codes à longueur fixe
Tous les mots ont même longueur (même nombre de symboles)
 - ✗ Codes à longueur variable
La longueur des mots varie en fonction de leur fréquence d'apparition
Un mot sera d'autant plus long que sa probabilité d'apparition sera petite
- + On dira qu'un code est uniquement déchiffrable (ou à décodage unique) si toute suite de mots code ne peut être interprétée (décodée) que d'une seule manière.
- + Exemples
 - ✗ {0,11,010} est uniquement déchiffrable.
 - ✗ {0,1,101} n'est pas uniquement déchiffrable car 101 peut être interprété comme 1-0-1 ou 101.

MÉTHODES DE CODAGE SANS PERTE

- + Un code préfixe est un code pour lequel aucun mot n'est le début d'un autre mot. Ce code est uniquement déchiffrable et est dit à décodage instantané.

Construction d'un code préfixe : construction d'un arbre dont chaque branche représente une lettre de l'alphabet élémentaire (alphabet de référence).

Il faut alors choisir les mots de telle sorte que pour tout couple de mots, le chemin permettant de conduire à l'un des mots ne soit pas inclus dans le chemin menant à l'autre mot.

Exemples :

- + $\{0,10,110,111\}$ est un code préfixe.
- + $\{0,11,110,111\}$ n'est pas un code préfixe car le chemin menant à 11 est inclus dans les chemins menant à 110 et 111.

CODAGE HUFFMAN (1952)

- **code à longueur variable : codes longs pour les symboles de probabilités faibles**

- ✗ **Algorithme de génération d'un codage optimal symbole par symbole.**
 - + Extraction des probabilités
 - + Création de l'arbre
 - + Création de la table d'Huffman
 - + Codage

- ✗ **Transmission/stockage**
 - + table + codes en binaire

- ✗ **Décodage**
 - + Lecture de la table d'Huffman
 - + Création de l'arbre de décodage
 - + Lecture séquentielle et décodage

CODE DE HUFFMAN

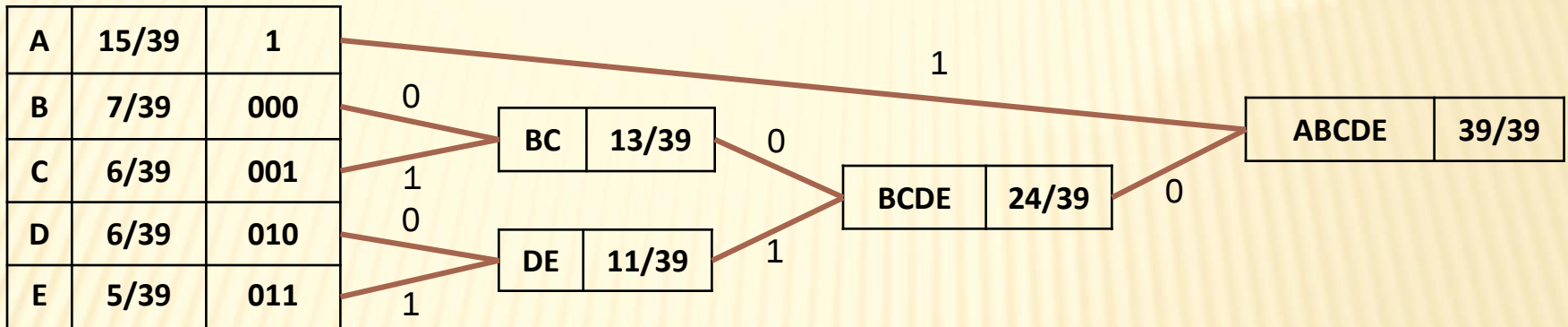
La construction de l'arbre permettant d'affecter aux n lettres source un mot code s'effectue de façon ascendante.

✗ Algorithme

- + Classement des lettres source suivant l'ordre décroissant de leur probabilité → on les considère comme les feuilles d'un arbre
- + Regroupement des deux lettres source de probabilités les plus faibles, en un nouveau groupe (nœud) de probabilité égale à la somme des probabilités des deux lettres source avec affectation des symboles «0» et «1» à chacune des branches
- + Considérant les $n-1$ lettres source restantes, itération à la première étape jusqu'à ce qu'il ne reste plus que 2 lettres source

CODE DE HUFFMAN- EXEMPLE

✗ Même source que l'exemple précédent



La longueur moyenne d'un mot code est :

$$1 \times 15/39 + 3 \times (1 - 15/39) = 2,23 \text{ bits}$$

➤ Entropie de la source (2,18 Sh) = limite

CODAGE SOURCE - REMARQUES

- ✗ **Le codage de Huffman nécessite la connaissance des probabilités *a priori* des lettres source.**
Sinon estimation en mesurant les fréquences d'apparition dans le message à transmettre → transmission de l'arbre des codes pour que le décodage soit possible.
- ✗ **Pour améliorer les performances, élever l'ordre d'extension de la source (ordre n = prendre n symboles à la fois pour les coder).**
Inconvénient : l'arbre des codes à transmettre plus long.
Exemple : si la source émet des symboles du code ASCII étendu, il y a 256 caractères.
Extension d'ordre 2 = d'une table de probabilités de $256^2 = 65536$ nombres!
→ inutile pour des messages courts.
- **Pour pallier cet inconvénient, on a introduit le code de Huffman adaptatif dont le principe repose sur la mise à jour des statistiques sur les symboles au fur et à mesure de la lecture du message.**

CODAGE ARITHMÉTIQUE (1976)

Huffman $\Rightarrow$ 1 symbole = 1 mot-code

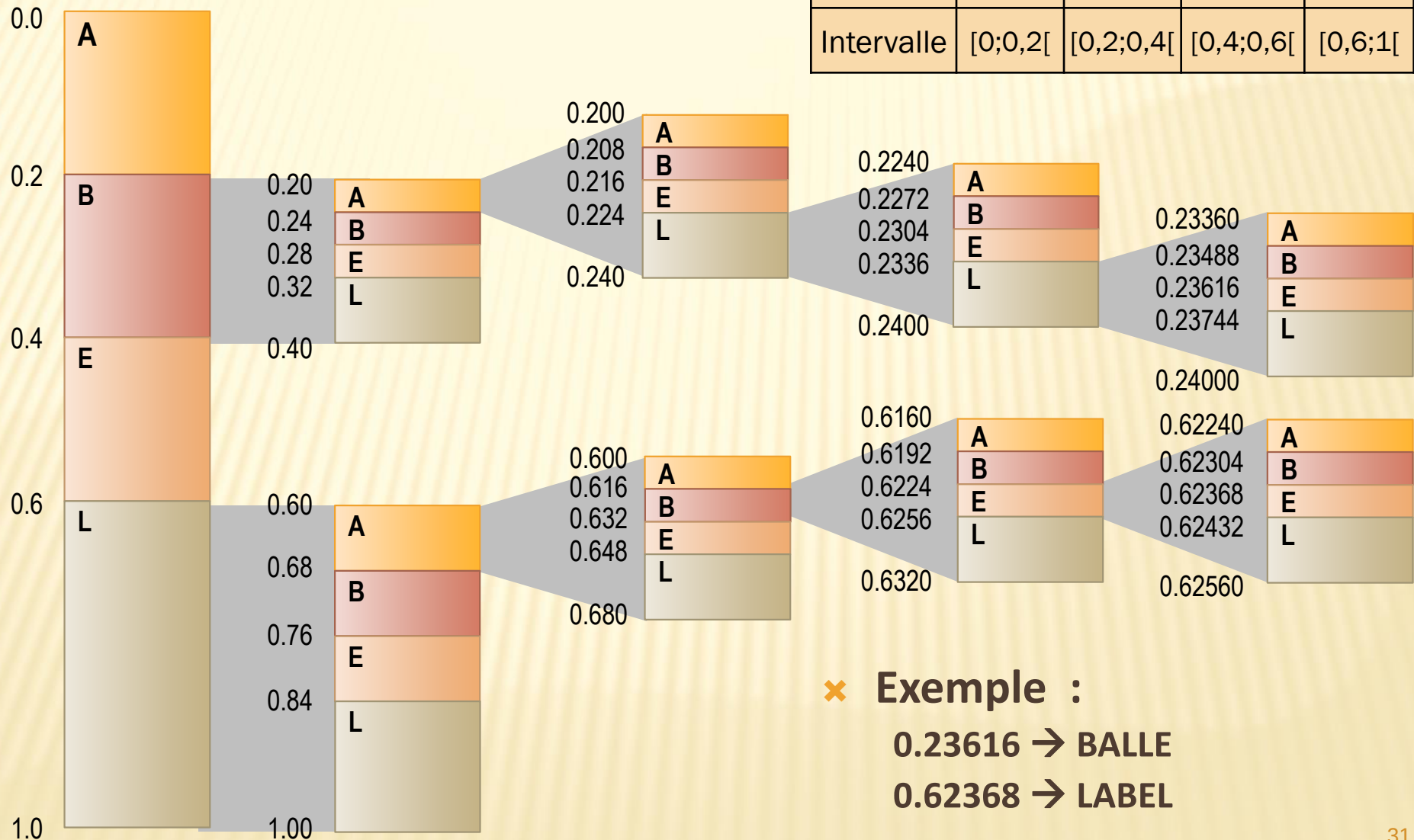
Arithmétique $\Rightarrow$ 1 flot de symboles = nombre en virgule flottante

✗ Principe

attribuer à l'ensemble du message un seul nombre en virgule flottante

Symbole	A	B	E	L
P_i	1/5	1/5	1/5	2/5
Intervalle	[0;0,2[	[0,2;0,4[	[0,4;0,6[	[0,6;1[

Exemple :



✗ **Exemple :**
 0.23616 → BALLE
 0.62368 → LABEL

DÉCODAGE ARITHMÉTIQUE

Symbole	A	B	E	L
P_i	1/5	1/5	1/5	2/5
Intervalle	$[0;0,2[$	$[0,2;0,4[$	$[0,4;0,6[$	$[0,6;1[$

1. **0,23616** $\rightarrow$ B **$(0,23616 - 0,2)/0,2 = 0,1808$**
2. **0,1808** $\rightarrow$ A **$(0,1808 - 0)/0,2 = 0,904$**
3. **0,904** $\rightarrow$ L **$(0,904 - 0,6)/0,4 = 0,76$**
4. **0,76** $\rightarrow$ L **$(0,76 - 0,6)/0,4 = 0,4$**
5. **0,4** $\rightarrow$ E

➤ **Le décodeur doit connaître les intervalles associés aux symboles**

CODAGE PAR LONGUEUR DE PLAGES (RUN LENGTH ENCODING)

✗ Objectif

- + coder une séquence binaire sans perte

✗ exemple

- + ligne d'image de 256 pixels : 0 0 0 0 0 0 0 1 1 1 1 1 0 0 ... 0

- + chaque plage est codée par

- ✗ la valeur de la plage
- ✗ la longueur des plages successives (sur 8 bits)

0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 1 1 1 1 0 1 0 0
7 5 256-12=244

→ 25 bits → taux de 10

- + il faut un faible nombre de plages
- + il faut de longues plages

CODAGE PAR LONGUEUR DE PLAGES (RUN LENGTH ENCODING)

✗ Codage du nombre de symboles identiques

0000011111000000000000000000	⇒	5w5b17w
0000000000001111100000000000	⇒	11w5b11w
A B C C C C C A B C A B C	⇒	A B !6C A B C A B C

✗ CCITT, Fax groupe III

↳ Huffman sur les plages de 0 précédant les 1

✗ JPEG

↳ Huffman sur les plages de 0 précédant les coeff. DCT

MOVE TO FRONT

✗ Modification dynamique de la table de codage

- + Lorsqu'un caractère apparaît dans le message, il est codé par sa position dans la table, puis déplacé au début de celle-ci
- Les plages de symboles identiques se transforment en plages de zéros
les codes correspondant aux symboles fréquents se retrouvent en début de table

+ Exemple : A A A C C B A A D D

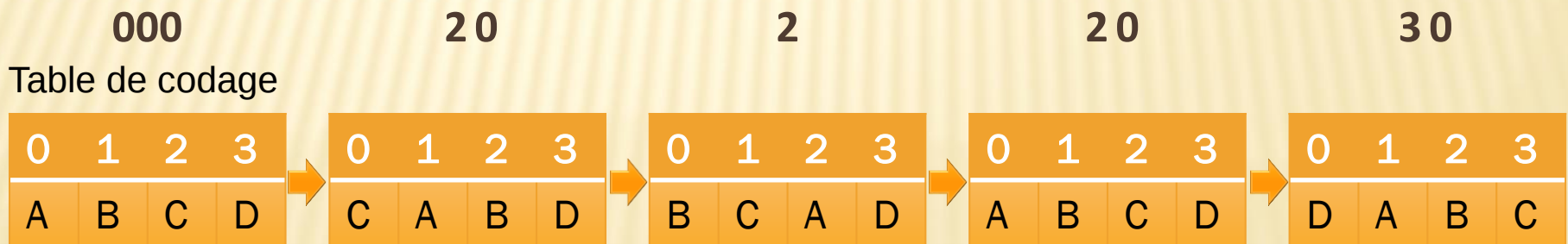
✗ Avec une table fixe :

0	1	2	3
A	B	C	D



code : 0 0 0 2 2 1 0 0 3 3

✗ Move-to-front :



TRANSFORMATION DE BURROWS-WHEELER (BWT)

✗ Réorganisation des données pour un codage plus efficace

→ message de meilleure entropie que le message initial

+ Génération d'un tableau à partir de tous les décalages possibles

+ Tri selon la première colonne

+ La dernière colonne + le numéro de la ligne contenant le premier caractère du message initial constitue le message à coder

→ La dernière colonne est pratiquement triée

TRANSFORMATION DE BURROWS-WHEELER (BWT)

+ Exemple : COMPRESSION

C O M P R E S S I O N

tri

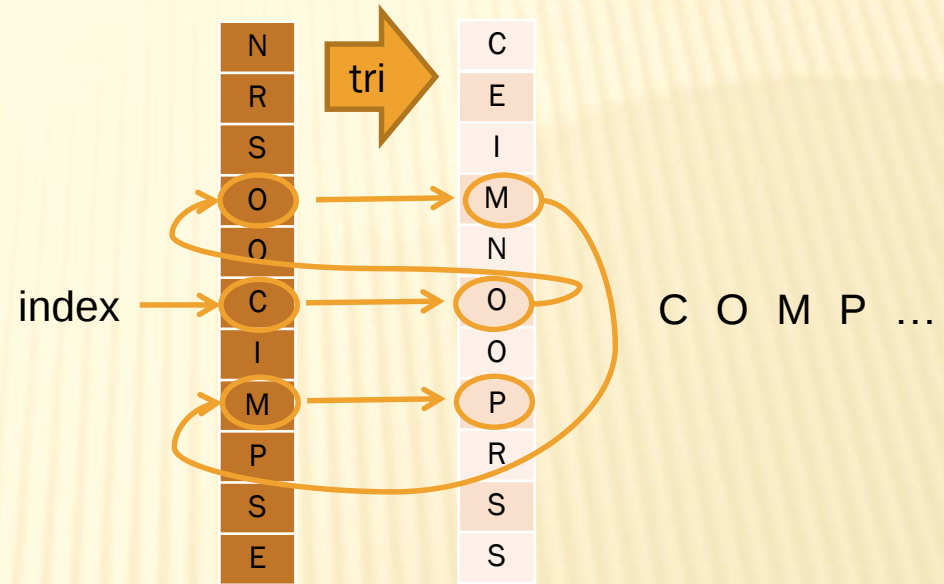
C	O	M	P	R	E	S	S	I	O	N
E	S	S	I	O	N	C	O	M	P	R
I	O	N	C	O	M	P	R	E	S	S
M	P	R	E	S	S	I	O	N	C	O
N	C	O	M	P	R	E	S	S	I	O
O	M	P	R	E	S	S	I	O	N	C
O	N	C	O	M	P	R	E	S	S	I
P	R	E	S	S	I	O	N	C	O	M
R	E	S	S	I	O	N	C	O	M	P
S	I	O	N	C	O	M	P	R	E	S
S	S	I	O	N	C	O	M	P	R	E

→ Message à coder : NRSOOCIMPSE + 5

TRANSFORMATION DE BURROWS-WHEELER (BWT)

✗ Décodage

- + Le message codé fournit la dernière colonne du tableau trié
- + Le tri du message redonne la première colonne
- + L'index du premier caractère donne accès au caractère suivant



- Transformation préalable à un codage type move-to-front et RLE
- Très efficace
- Utilisée dans BZIP2

CODAGE À BASE DE DICTIONNAIRE

- ✗ **Premier algorithme publié par Lempel et Ziv en 1977 : LZ77**
 - + Utilisé par le format PNG à partir de 1995 (alternative libre au format GIF)
- ✗ **Algorithme LZW (Lempel Ziv Welch),**
 - + Variante de LZ77 publiée par T. Welch et brevetée par Unisys en 1985
 - + Compress, Unix
 - + PDF, .., GIF

CODAGE À BASE DE DICTIONNAIRE : LZW

- ✗ **Codage Huffman → modèle sans mémoire (ordre 0)**
- ✗ **Modèle avec mémoire (ordre supérieur) :**
 - + probabilité de « u » en français : 0,05
 - + probabilité de « u » sachant la lettre précédente « q » : 0,95
- ✗ **On part d'une table de symboles qu'on enrichit au fur et à mesure pendant le codage**
- ✗ **Définitions :**
 - + symbole
 - + séquence de symbole
 - + dictionnaire : table des séquences de symboles
 - + code émis : index ou adresse des séquences de symboles

CODAGE À BASE DE DICTIONNAIRE : LZW

✗ Algorithme de codage

Entrée et Motif sont initialement vides

- + lire **Entrée** (un symbole élémentaire)
- + si **Motif + Entrée** est présent dans le dictionnaire
 - ✗ **Motif = Motif + Entrée**
- + sinon
 - ✗ ajouter **Motif + Entrée** dans le dictionnaire à l'index disponible suivant
 - ✗ émettre l'index du **Motif** comme code en sortie
 - ✗ **Motif = Entrée**
- + reboucler tant que les index restent inférieurs à 4096 (2^{12} si 12 bits)

✗ Décodage

- + Le dictionnaire est reconstruit dynamiquement au moment du décodage

CODAGE À BASE DE DICTIONNAIRE : LZW

✗ exemple

soit à coder une image avec 256 valeurs par pixel

valeurs : 8 8 255 160 8 8 8 240

La table T est initialisée de 0 à 255 (256 entrées)

Entrée et Motif : vides

Entrée	Motif _n	Motif _n + Entrée	T(index, séq. symbole)	Code	Motif _{n+1}
8	-	8 ∈ T	-	-	8
8	8	8 8 ∉ T	(256,8 8)	8	8
255	8	8 255 ∉ T	(257,8 255)	8	255
160	255	255 160 ∉ T	(258,255 160)	255	160
8	160	160 8 ∉ T	(259,160 8)	160	8
8	8	8 8 ∈ T	-	-	8 8
8	8 8	8 8 8 ∉ T	(260,8 8 8)	256	8
240	8	8 240 ∉ T	(261,8 240)	8	240

..

Indices codés sur de 9 bits jusqu'a 12 bits

CODAGE À BASE DE DICTIONNAIRE : LZW

- **LZW15VC : amélioration de LZW capable de coder les adresses jusqu'à 15 bits.**
 - + commence à coder sur 9 bits et utilise un bit supplémentaire après avoir codé 256 valeurs supplémentaires
 - + le dictionnaire est vidé lorsqu'il est complètement rempli
 - + ceci permet de le rendre aussi performant que d'autres codes pour les messages courts.

QUELQUES FORMATS DE CODAGE SANS PERTE

- + **BMP : BitMap Windows/IBM crée pour Windows 1.0 (1985)**
 - × Pixels codés sur 1, 4, 8, 24 bits
 - × Compression RLE pour les images sur 4 et 8 bits
 - × Pas de compression des images couleur

- + **GIF : Graphic Interchange Format (1987)**
 - × limité à 256 couleurs !!!!!
 - × compression LZW
 - × Copyright par COMPUSERVE

- + **PCX : ZSoft/Microsoft (1990), (lié au logiciel PC Paintbrush)**
 - × images sur 1, 4, 8 ou 24 bits
 - × ne peut gérer 16 ou 32 bits de codage pour les pixels
 - × Compression RLE sur les bandes RGB d'une ligne

QUELQUES FORMATS DE CODAGE SANS PERTE

+ PNG : Portable Network Graphics (1995)

- × Compression LZ77
- × Alternative libre au format GIF
- × Format souple incluant gamma et transparence
- × Prend en charge des images NG 16 bits, 256 couleurs indexées, vraies couleurs jusqu'à 48 bits/pixel

+ TIFF :

- × format « poids lourd »
- × compression LZW
- × format complet mais complexe : beaucoup de champs
- × pas beaucoup d'applications capables de gérer toutes les possibilités et subtilités de ce format
- × Dans la version 6.0 de 1992, prise en charge des images de tous types (binaires, niveaux de gris, couleurs indexées, RGB, YCbCr, $L^*a^*b^*$, transparence...) et de différentes techniques de compression (RLE, CCITT, LZW, JPEG)

RÉSUMÉ : MÉTHODES DE CODAGE SANS PERTE

- ✗ **RLE (Run Length Encoding)**

- ✗ **Codage de Huffman**

au moins 1 bit par symbole

proba=0,9 → information = 0,15 bits, mais au moins 1 bit avec Huffman

- ✗ **Codage arithmétique**

moins d'1 bit par symbole, mais sans mémoire

- ✗ **Codage à base de dictionnaire**

modèle avec mémoire

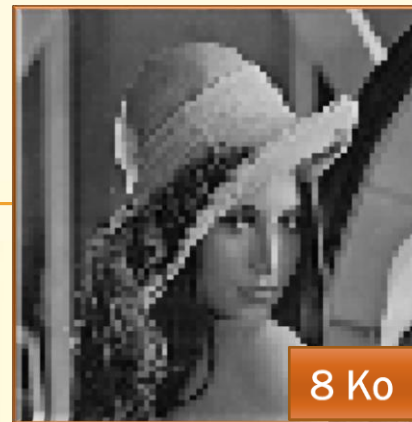
COMPRESSION AVEC PERTES

✗ Réfléchir avant d'agir !

Lena originale
256x256x8 bits



Sous-échantillonnage
par 2,8 ($2,8^2=8$)



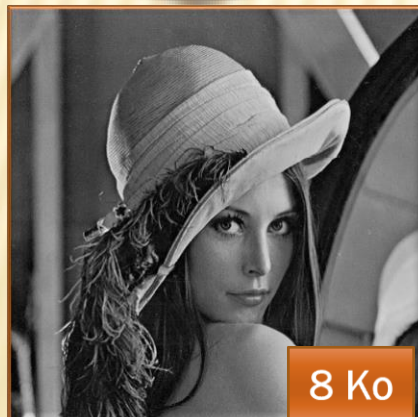
Quantification
sur 1 bit



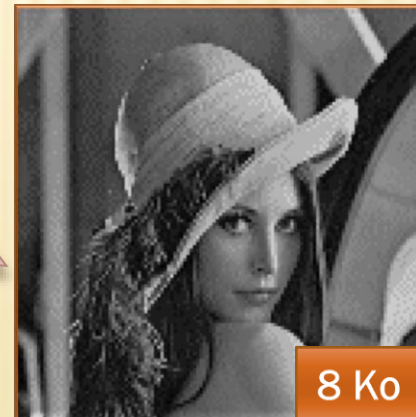
Diffusion d'erreur



JPEG
taux=8



Sous-échantillonnage par 2 et
quantification sur 4 bits



COMPRESSION AVEC PERTES

× Réduction de redondance

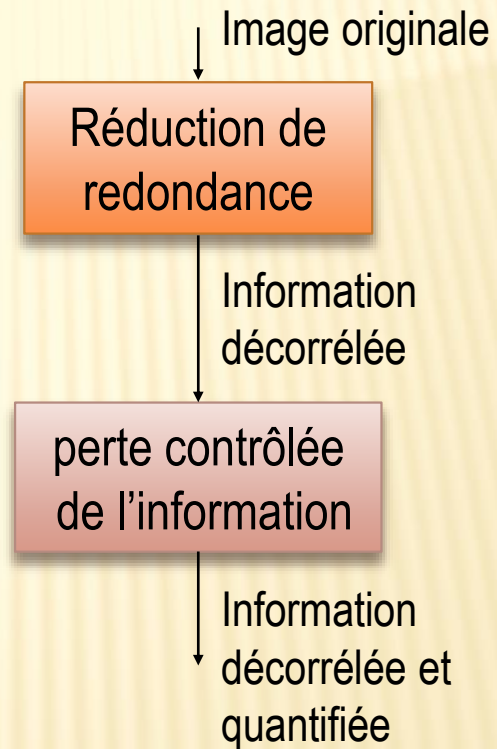
+ Redondance spatiale

× Méthodes spatiales

★ Méthodes prédictives

× Méthodes par transformées

× Méthodes fractales



RÉDUCTION DE REDONDANCE

✕ Méthodes spatiales

→ Méthodes prédictives

MICD : modulation par impulsion et codage différentiel

DPCM : Differential Pulse Code Modulation

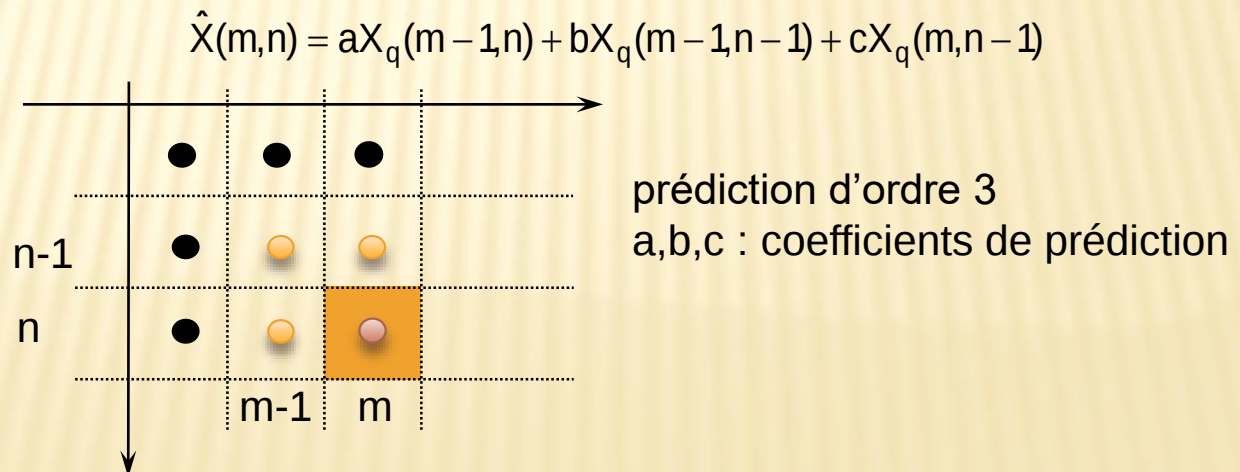
+ Objectif

réduire la redondance avec une prédiction du pixel futur par un ensemble de n pixels voisins connus : DPCM d'ordre n

+ Hypothèse

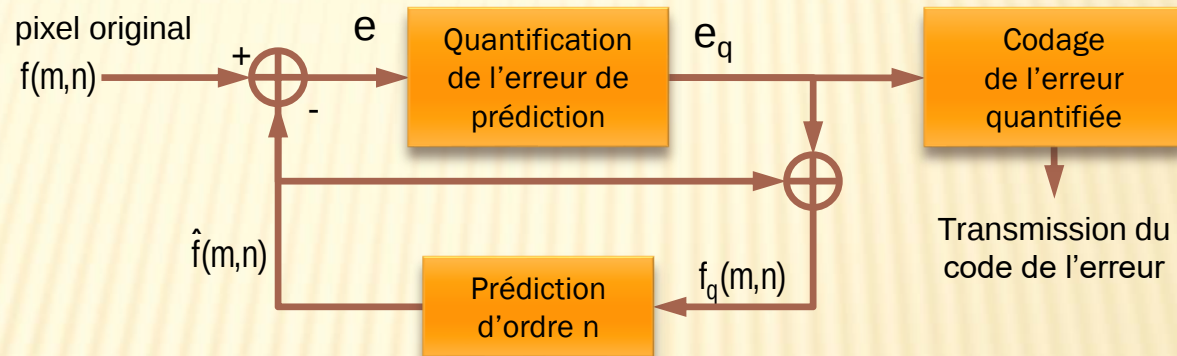
image naturelle, donc variation lisse des NG

→ ressemblance ou corrélation entre les pixels voisins

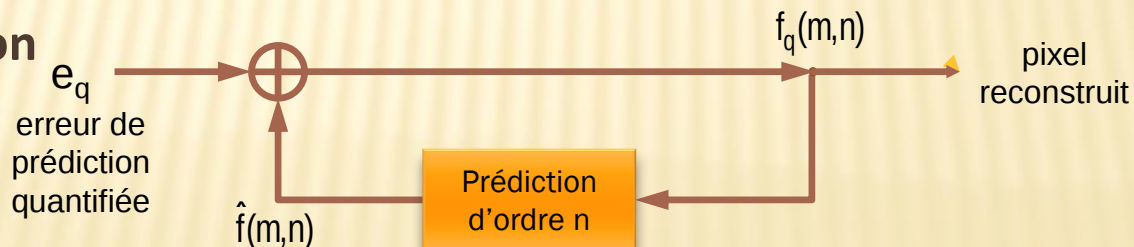


✖ Transmission

- + n pixels originaux pour commencer la prédiction
- + n coefficients de prédiction
- + erreur de prédiction



✖ Décompression



+ Désavantage :

- × Sensibilité aux erreurs de transmission**

l'erreur se répercute et s'amplifie au fur et à mesure → rafraîchir par des pixels originaux

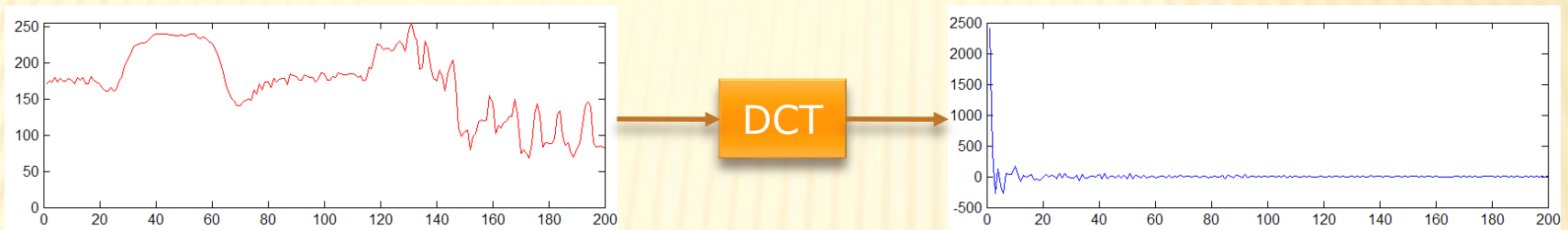
+ Estimation des coefficients de prédiction

- × mesure de la corrélation entre les pixels voisins, mais une image n'est pas stationnaire**
- × adaptation locale des coefficients → mais il faut les transmettre !!**

✗ Méthodes par transformées

→ Changement de base de l'image pour une représentation plus compacte

+ Exemple : deux représentations du même signal



+ Karhunen-Loeve

✗ décorrélation parfaite des coefficients transformés :

$$\text{Cor}(S(k), S(l)) = 0 \quad k, l = 0, \dots, N-1 \text{ et } k \neq l$$

donc aucune redondance dans S

✗ très intéressant pour l'analyse de données et la compression

mais la base dépend de l'image (vecteurs propres de la matrice d'auto-covariance)

+ Walsh-Hadamard

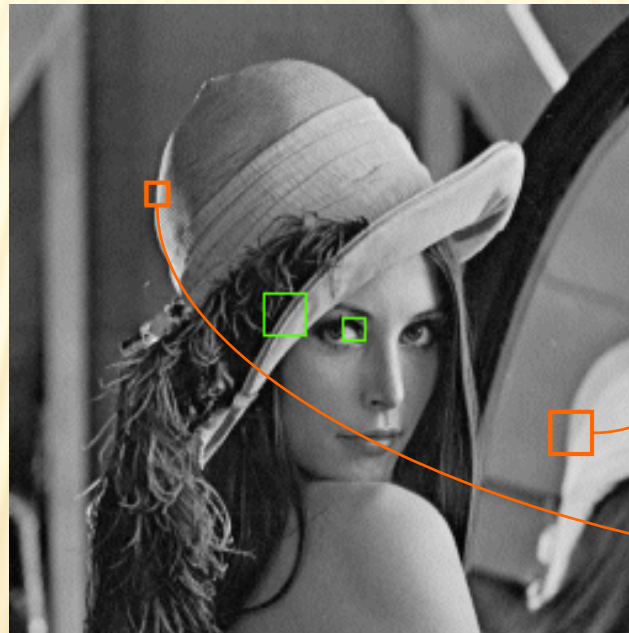
+ Transformée Cosinus discrète (→ JPEG)

+ Ondelettes (→ JPEG2000)

✗ Méthodes fractales

→ recherche d'auto-similarités

Une référence : la méthode de Jacquin (1989)



Ajustement de luminance



Réduction



× Double partition de l'image



- × Pour chaque bloc de destination, recherche du meilleur bloc source et des transformations (rotation, retournement, réduction de la moyenne, ...)
- × Codage des couples et des transformations

+ Redondance couleur

→ Espace de couleur permettant une bonne décorrélation

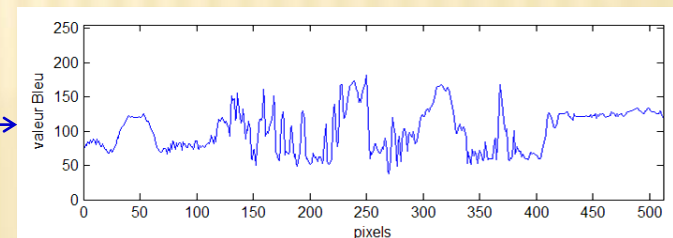
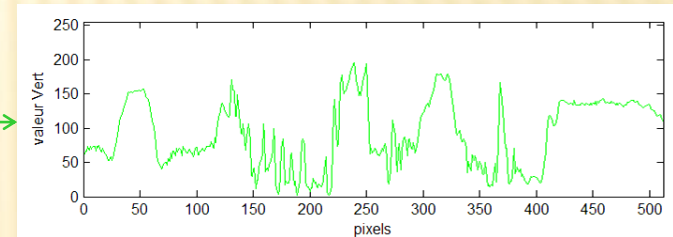
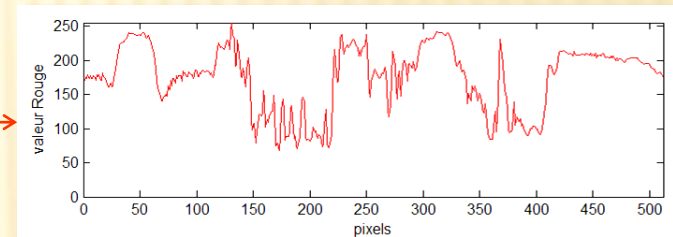
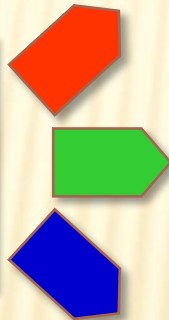
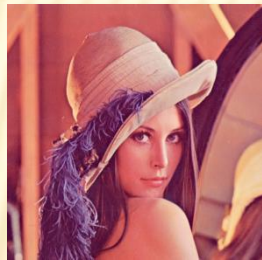
+ espace adapté à la quantification (contrôle de la dégradation)

?

+ Exemple

Espace RVB

→ fortes corrélations



+ Redondance couleur

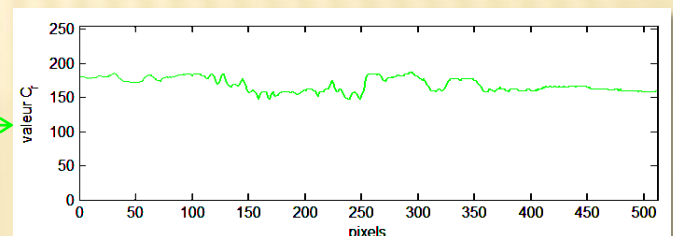
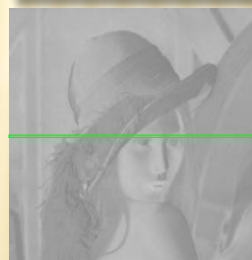
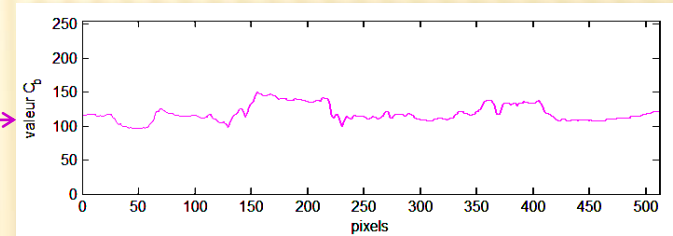
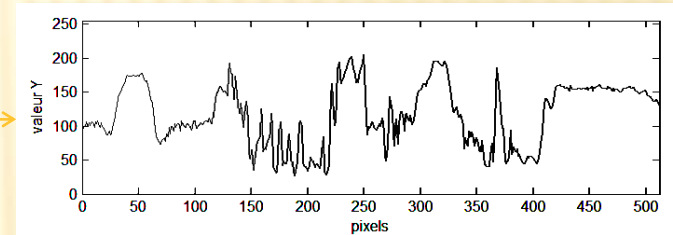
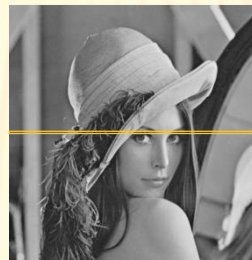
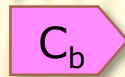
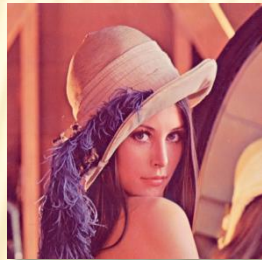
→ Espace de couleur permettant une bonne décorrélation

+ espace adapté à la quantification (contrôle de la dégradation)

+ Exemple

Espace $Y C_b C_r$

→ bonne décorrélation



+ Espaces adaptés au contrôle des distorsions visuelles

→ L^*a^*b ou L^*u^*v

+ Prise en compte de la perception des textures et des effets de masquage

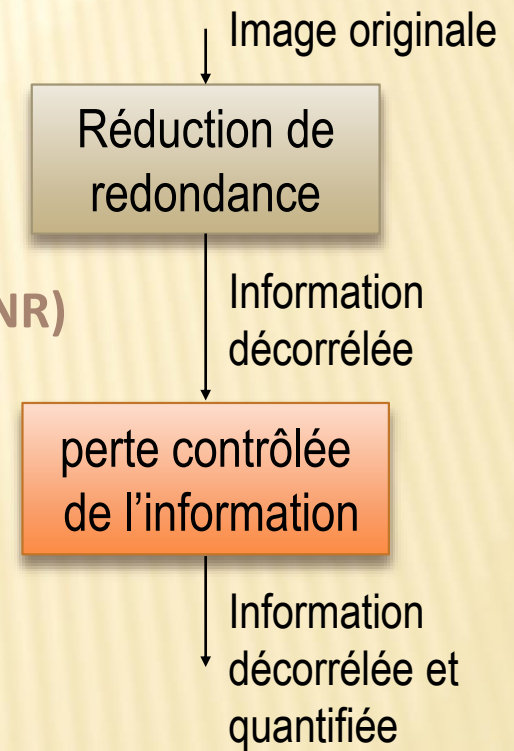
→ Modèle de Poirson et Wandell

→ Modèle modifié de Nadenau et Reichel

QUANTIFICATION

✗ Quantification scalaire

- ✗ Dynamique de la source (de $V_{\min}$ à $V_{\max}$)
- ✗ Distribution de la source ($p(x)$)
- ✗ Quel débit (pour s'adapter à un canal) ? (bpp)
- ✗ Quelle distorsion (quelle qualité visuelle) ? (EQM, PSNR)



QUANTIFICATION

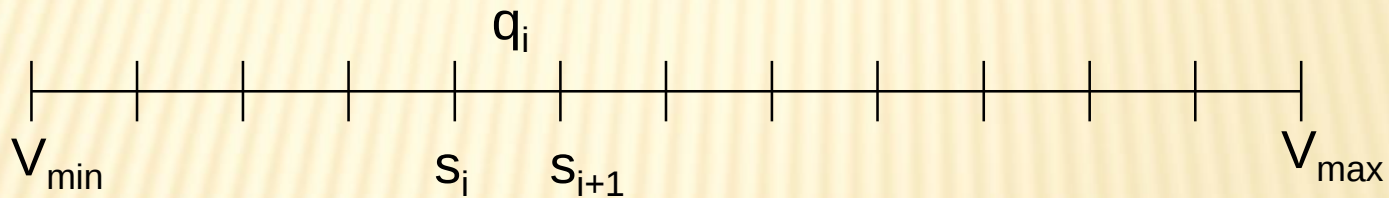
✗ Quantification scalaire uniforme linéaire

→ intervalles égaux

+ seuils de quantification s_i espacés régulièrement entre $V_{\min}$ et $V_{\max}$ (répartition uniforme)

+ valeurs de quantification q_i espacées régulièrement entre s_i et s_{i+1} (répartition uniforme)

$$q_i = \frac{s_i + s_{i+1}}{2}$$



→ ne prend pas en compte la densité de probabilité

QUANTIFICATION

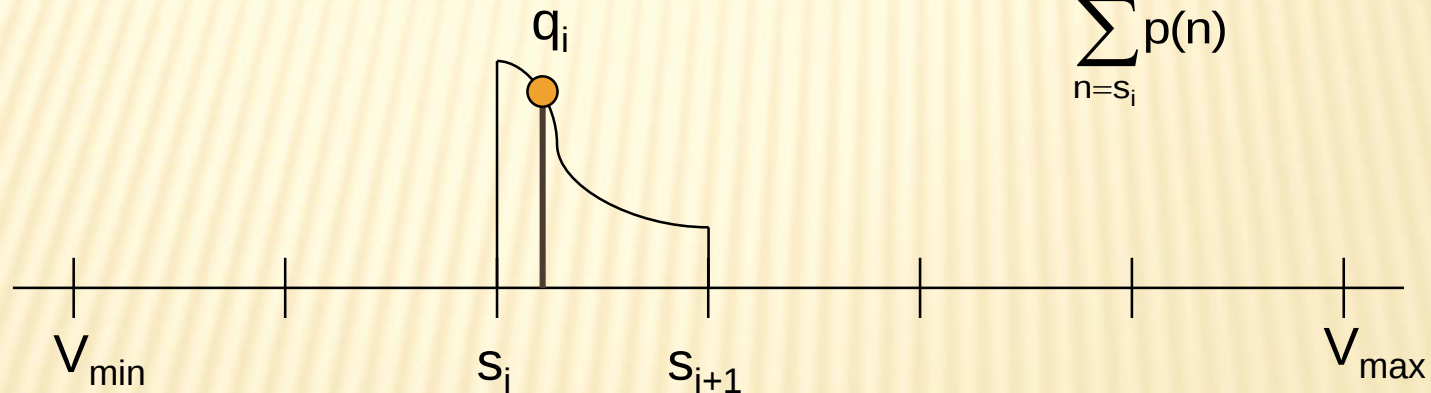
✗ Quantification scalaire uniforme optimale

→ dépend de la densité de probabilité

+ Répartition uniforme des seuils de quantification s_i entre $V_{\min}$ et $V_{\max}$

+ q_i : barycentre de $p(x)$ entre s_i et s_{i+1}

$$q_i = \frac{\sum_{n=s_i}^{s_{i+1}} n.p(n)}{\sum_{n=s_i}^{s_{i+1}} p(n)}$$



QUANTIFICATION

✗ Quantification non uniforme

- ✗ Prise en compte du SVH (quantification non linéaire du contraste)
- ✗ Minimisation d'erreur (Lloyd-Max)
 - ★ Calcul des seuils et des valeurs de quantification avec l'hypothèse d'un modèle de distribution → répartition non uniforme des s_i et des q_i
 - ★ Modèle a priori : généralement Gaussien
 - ★ Contrainte de débit : b bits par pixel
 - ★ Minimisation de la distorsion globale D → détermination des s_i et q_i par des algorithmes itératifs :

$$D = \sum_{i=0}^{2^b-1} \sum_{n=s_i}^{s_{i+1}} (n - q_i)^2 p(n)$$

QUANTIFICATION

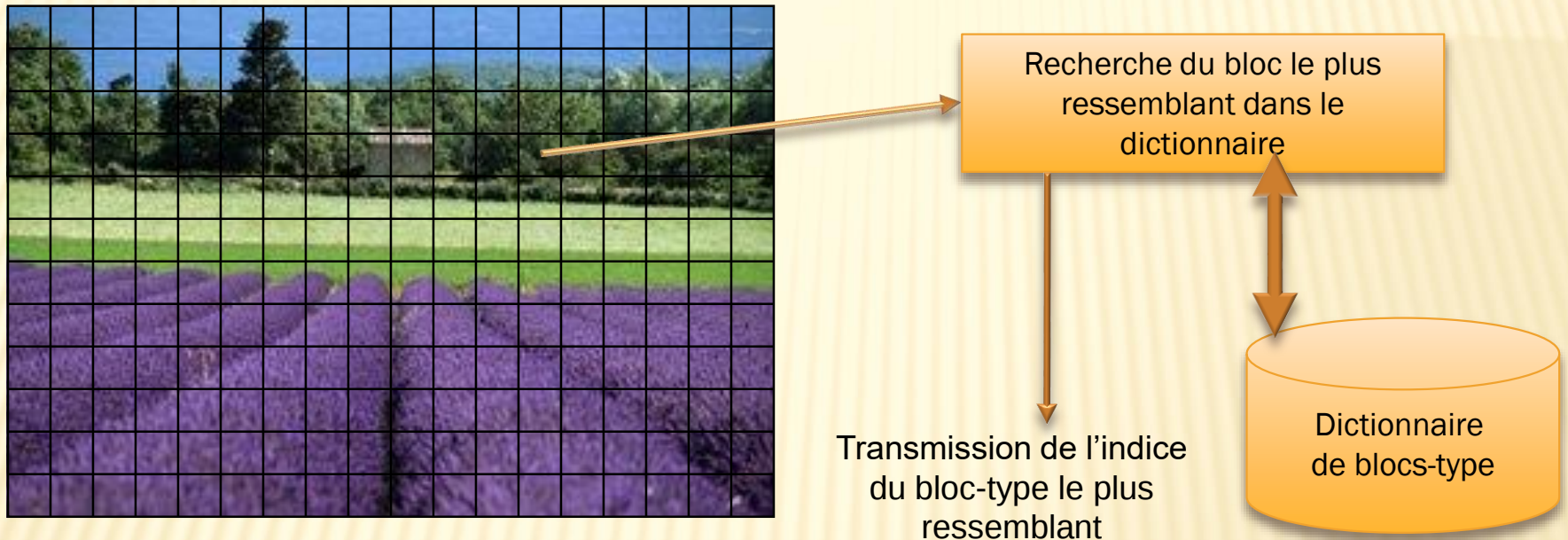
✗ Quantification vectorielle (QV) (Vector quantization VQ)

+ Dictionnaire de vecteurs types

- ✗ blocs de pixels (quantification spatiale)
- ✗ Transformées de blocs (quantification vectorielle à transformée)
- ✗ Composantes couleur

+ Codage par l'index du bloc le plus ressemblant dans le dictionnaire

→ recherche la similarité entre les blocs de pixels ou de coefficients



- Bloc-type (codeword) : bloc de pixels $n \times n$ ou vecteur de pixels $1 \times n^2$ qui représente une classe de blocs similaires
- Dictionnaire (codebook) : ensemble des blocs-type
- Indice du bloc-type : n° du bloc-type codé sur k bits si le dictionnaire comporte 2^k blocs-type

QUANTIFICATION VECTORIELLE QV

✖ Hypothèses

- + bloc 4x4 (ou vecteur de taille 16)
- + 256 codewords dans le codebook (k=8)
- + pixel original codé sur b=8 bits

✖ Taux de compression par bloc

$$T_c = \frac{n.n.b}{k} = \frac{4.4.8}{8} = 16$$

→ Débit par bloc = Débit pour toute l'image = 0.5 bpp
Débit fixe imposé par les 3 hypothèses de départ

✖ Détermination du dictionnaire

Problème délicat, car de la qualité du dictionnaire dépend la qualité de l'image codée

QUANTIFICATION VECTORIELLE QV

- ✗ **Détermination du dictionnaire : algorithme LBG (1980) (Linde Buzo Gray)**
 - + connu sous le nom de la méthode des nuées dynamiques en analyse de données.
- + C'est une méthode par apprentissage : besoin d'une base de données (base de blocs $n \times n$)
 - ✗ Constitution de la base d'apprentissage : ensemble d'images ressemblantes divisées en blocs
 - ✗ Problème : répartir la base en C classes de blocs
 1. Initialisation du dictionnaire par C codewords
 2. Pour chaque vecteur V_i , faire
 - a. calcul de distance $d(V_i, C_k)$ (mesure de similarité) entre V_i et chaque codeword $C_k \rightarrow C$ distances
 - b. classer V_i dans la classe k qui minimise $d(V_i, C_k)$
 3. Pour chaque classe k obtenue, faire
 - a. calculer le barycentre de la classe
 - b. remplacer C_k par ce barycentre
 4. Si les nouveaux C_k sont tous les mêmes que les anciens, alors fin du calcul, sinon aller à l'étape 2

QUANTIFICATION VECTORIELLE QV

- × **Détermination du dictionnaire**

- long et fastidieux, mais fait une seule fois (en temps différé)

- × **Compression**

- associer l'indice de la classe k au bloc V_i à coder si $d(V_i, C_k)$ est minimale : assez rapide si le dictionnaire n'est pas très gros

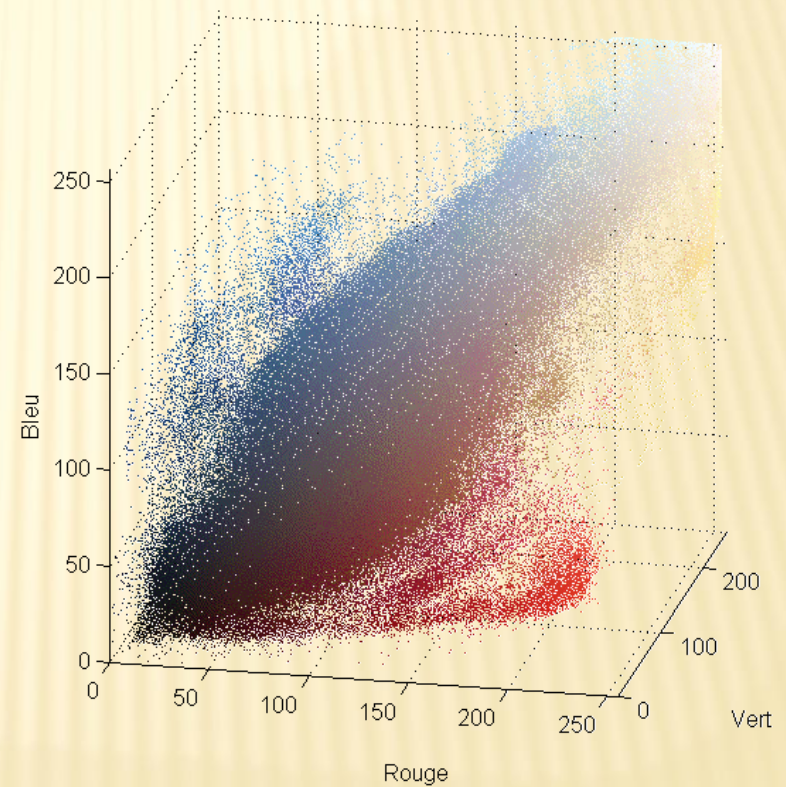
- × **Décompression**

- construire l'image décodée avec les blocs-type associés aux indices reçus : très rapide

QUANTIFICATION COULEUR

+ Exemple

- × 5 308 416 pixels
- × 194931 couleurs sur 16 777 216 possibles



QUANTIFICATION COULEUR

- ✗ **Uniforme** → boîtes de taille et formes constantes
 - ne prend pas en compte la répartition des couleurs
- ✗ **Non uniforme**
 - Méthodes de classification et d'apprentissage
 - + Méthodes scalaires séquentielles
 - sur chaque axe à partir des histogrammes marginaux
 - + Partitionnement 3D
 - ✗ Mode descendant : division
 - ✗ Mode ascendant : fusion
 - ✗ Mode mixte : division-fusion
 - ✗ Algorithme LBG (Linde-Buzo-Gray)
 - ✗ Approches neuronales
 - + Recherche dans le dictionnaire
 - ✗ Exhaustive
 - ✗ Arborescente
 - Taille et organisation du dictionnaire
 - Temps de codage/décodage

STANDARD JPEG

✕ Historique des standards

- + JPEG pour la compression d'images fixes en couleur (1980 -1993)
- + MPEG1 pour l'enregistrement support CD-ROM, multimédia (1988 - 1992)
- + Norme H.261 pour visiophone et visioconférence sur le RNIS (1990)
- + MPEG2 pour la TV de distribution (TV,TVHD) (1990 -1994)
- + MPEG4 pour les applications multimédia (1994 - 1998)
- + MPEG7 pour les applications multimédia (1998 - fin 2001)

STANDARD JPEG

✗ **JPEG : Joint Picture Expert Group**

première réunion en mars 86

✗ **Objectif**

Codage d'images fixes couleur avec ou sans perte

+ 3 méthodes en compétition en 87 :

- ✗ DCT (transformée cosinus discrète)
- ✗ DPCM (méthode prédictive)
- ✗ BTC (block truncation coding)

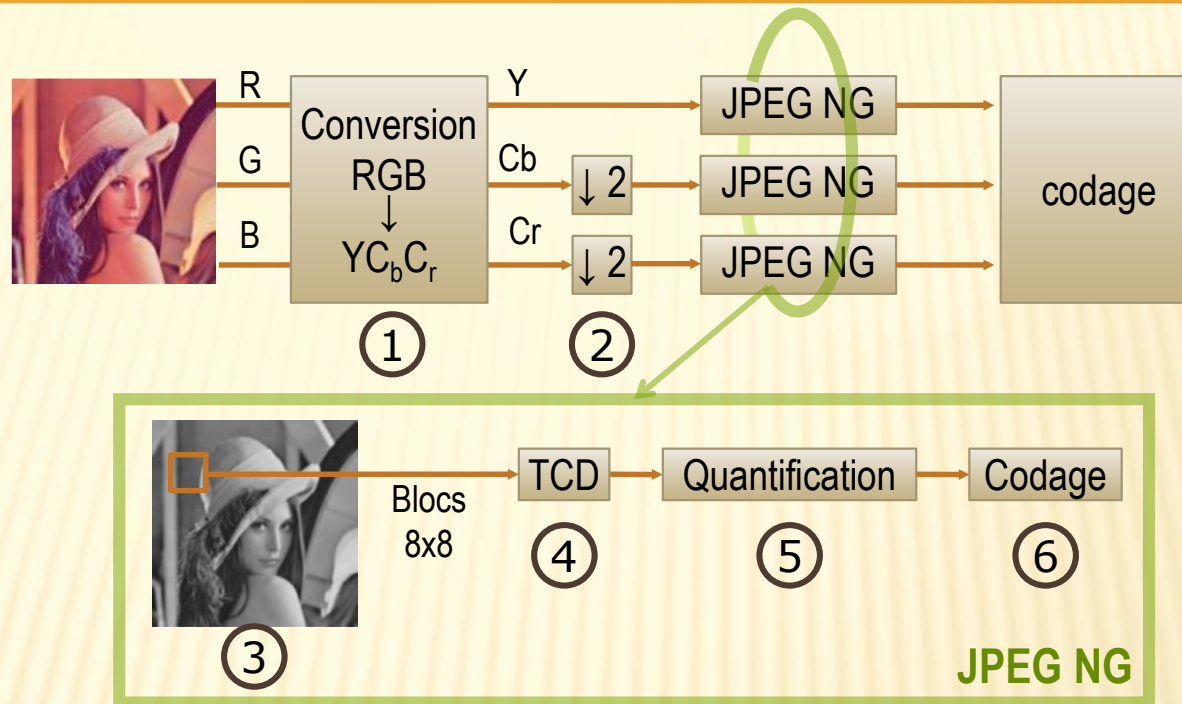
+ Choix de la DCT en janvier 88

+ JPEG devient un standard international (IS) en 93

+ Le standard spécifie entièrement le codeur et le décodeur

JPEG

× Principe



- ① Décorrélation des composantes couleur
- ② Sous-échantillonnage de C_b et C_r
- ③ Découpage en blocs
- ④ Transformation DCT pour décorrélérer et compacter l'information
- ⑤ Quantification pour éliminer les coefficients peu significatifs
- ⑥ Codage pour générer le fichier binaire (codage entropique VLC)

✖ Conversion couleur

→ de RGB à $Y C_b C_r$

$$\begin{cases} Y = 0.299R + 0.587V + 0.114B \\ C_b = 0.564(B - Y) \\ C_r = 0.713(R - Y) \end{cases}$$

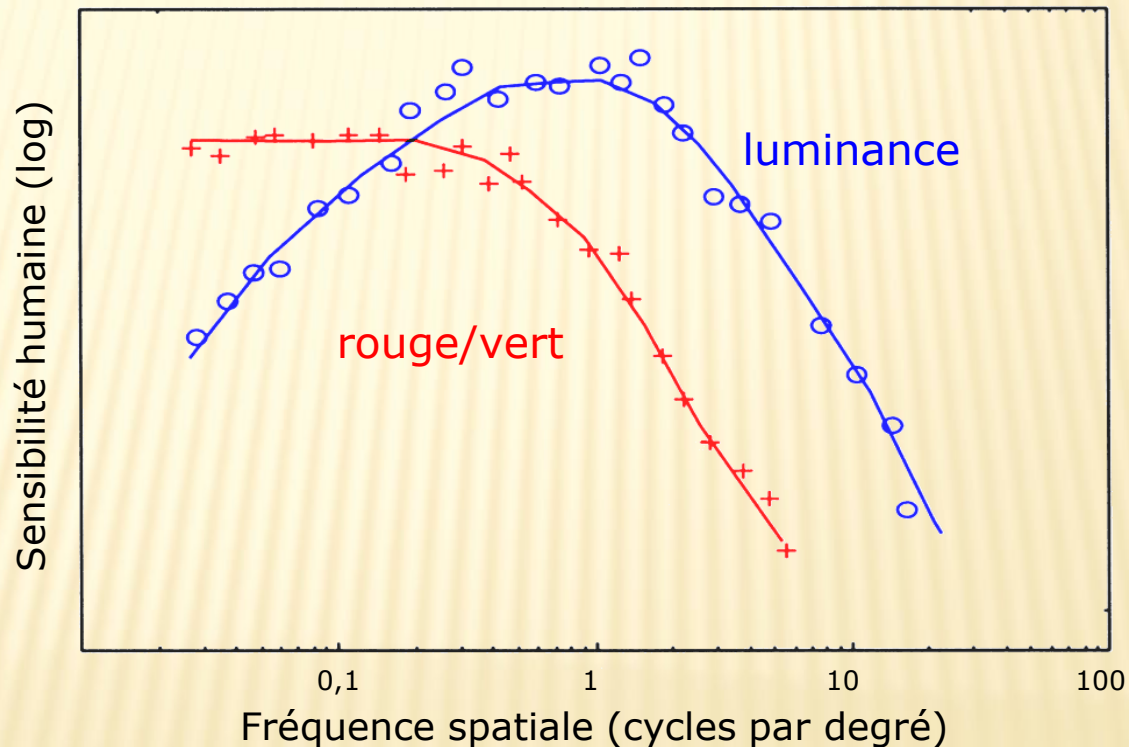
soit

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ V \\ B \end{bmatrix}$$

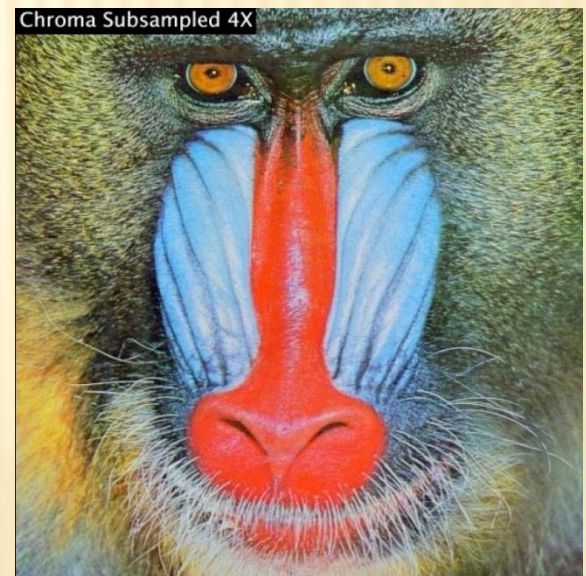
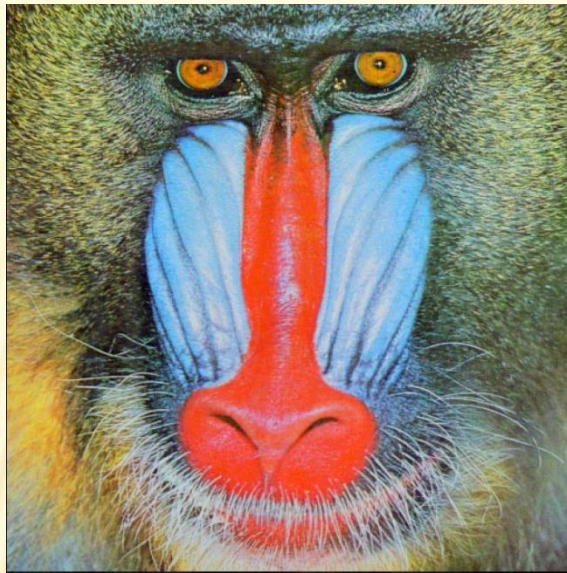
$$\begin{bmatrix} R \\ V \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.4021 \\ 1 & -0.3441 & -0.7142 \\ 1 & 1.7718 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix}$$

✗ Sous-échantillonnage de C_b et C_r

car l'œil est de l'ordre de trois fois plus sensible aux variations de luminance par rapport aux variations de chrominance



× Exemple



✗ Découpage en blocs 8x8

- + Traitement indépendant des blocs
- + Adaptabilité locale
- + Réduction de mémoire pour le traitement
- + Attention aux effets de bloc

✗ Transformée cosinus discrète

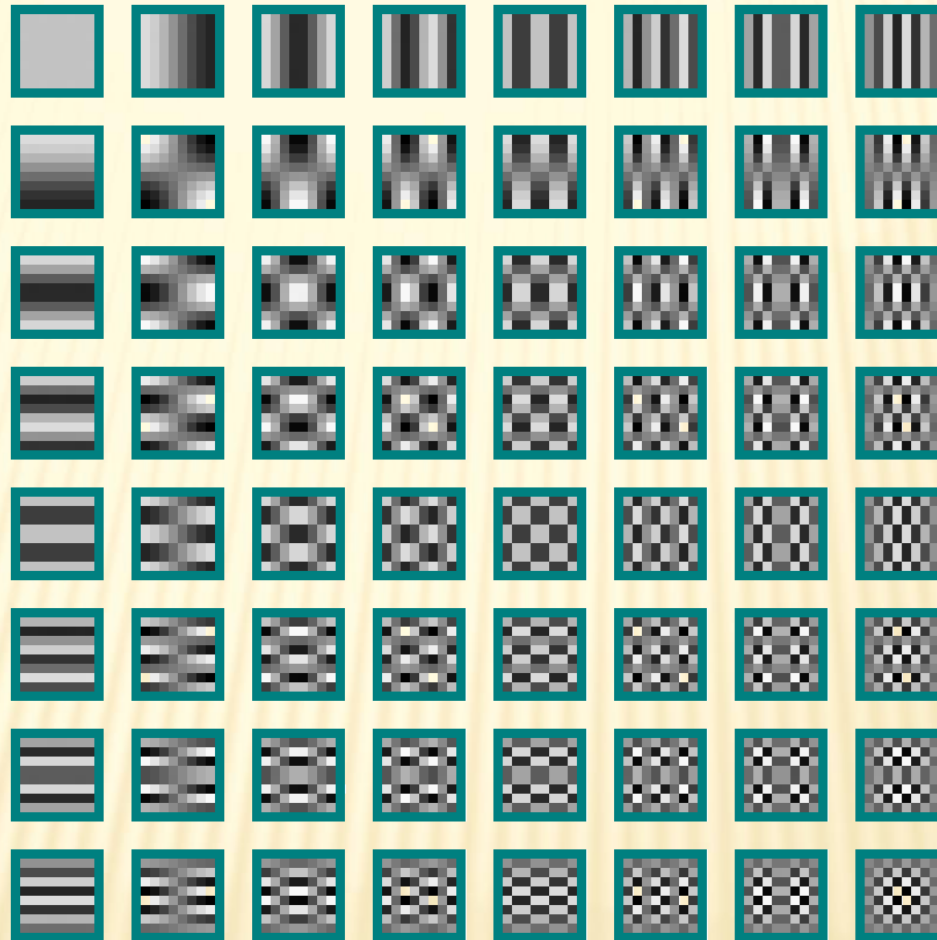
$$X_{\text{DCT}}(k,l) = \frac{C(k)C(l)}{4} \sum_{i=0}^7 \sum_{j=0}^7 X(i,j) \cos\left(\frac{\pi k(2i+1)}{16}\right) \cos\left(\frac{\pi l(2j+1)}{16}\right) \quad \text{DCT}$$

$$X(i,j) = \frac{1}{4} \sum_{k=0}^7 \sum_{l=0}^7 C(k)C(l) X_{\text{DCT}}(k,l) \cos\left(\frac{\pi k(2i+1)}{16}\right) \cos\left(\frac{\pi l(2j+1)}{16}\right) \quad \text{DCT inverse}$$

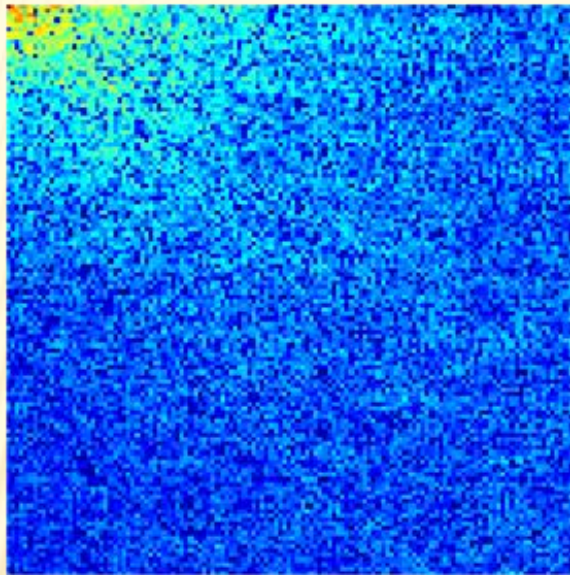
$$C(k) = C(l) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } k = l = 0 \\ 1 & \text{sinon} \end{cases}$$

- ✗ Coefficients DCT réels
- ✗ Valeur DCT à l'origine proportionnelle à la moyenne de l'image
- ✗ Transformation linéaire
- ✗ Conservation de l'énergie

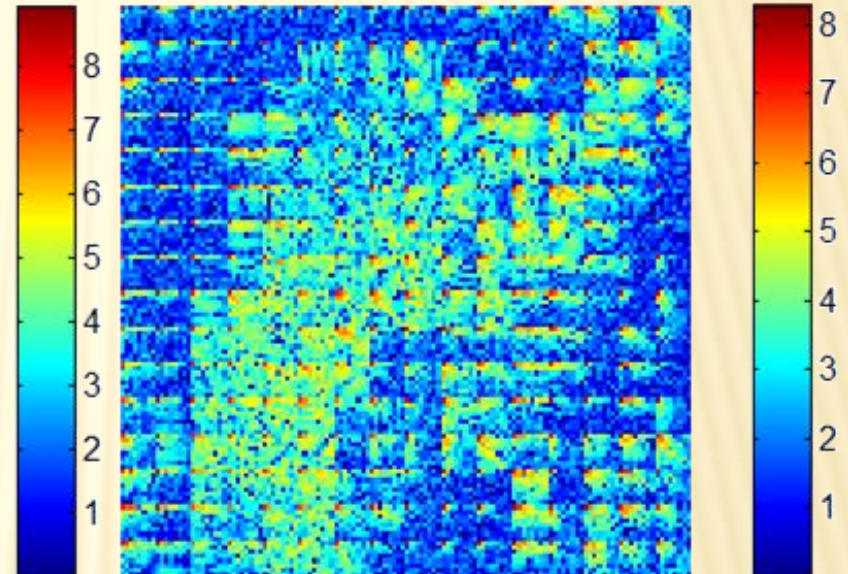
DCT par blocs 8x8 : matrices de base



✕ Exemples



DCT full frame de Lena



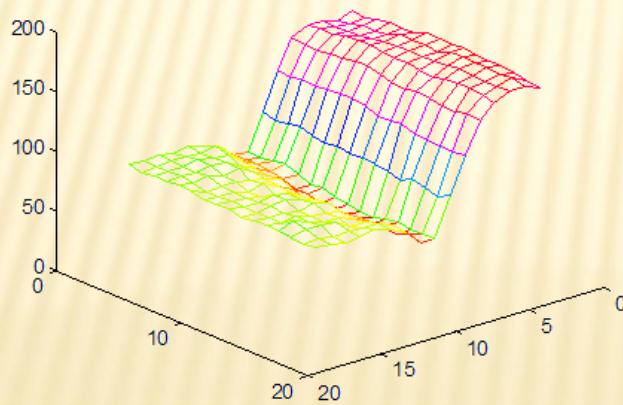
DCT par bloc 8x8

- + coefficients décorrélés
- + compacité de l'information

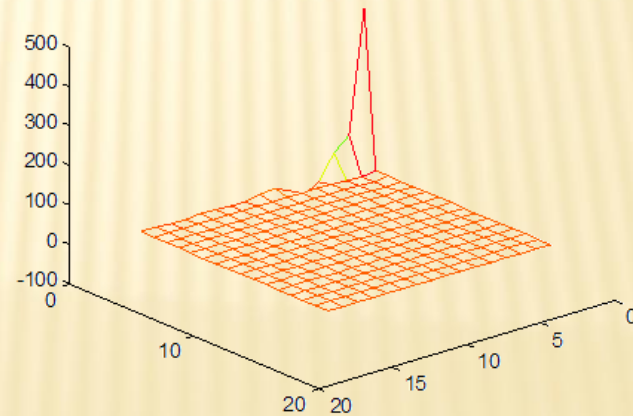
✗ Exemple : Transformée cosinus d'un bloc image



Bloc image



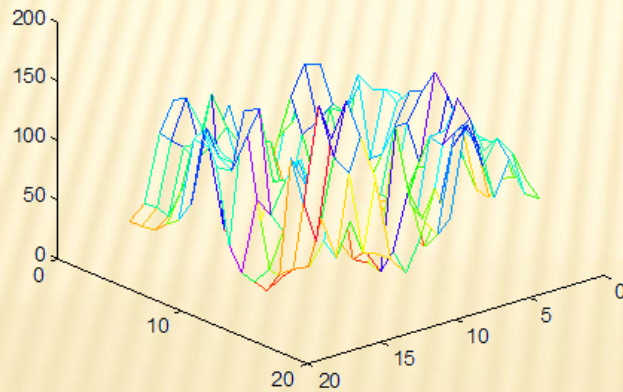
Spectre DCT du bloc



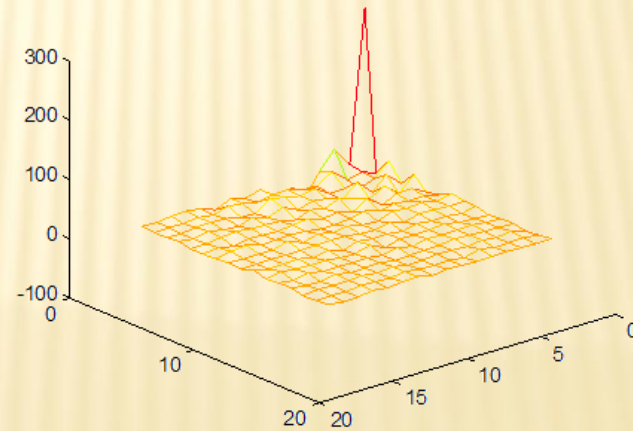
✗ Exemple : Transformée cosinus d'un bloc image



Bloc image



Spectre DCT du bloc



✗ Quantification des blocs

63	46	24	14	22	40	56	64	416	-32	60	32	48	0	0	0	16	11	10	16	24	40	51	61
56	39	31	43	57	61	60	60	12	-24	-56	0	0	0	0	0	12	12	14	19	26	58	60	55
52	35	42	78	95	79	63	63	-42	14	80	24	-40	0	0	0	14	13	16	24	40	57	69	56
55	37	50	90	100	72	59	73	-56	17	44	-29	0	0	0	0	14	17	22	29	51	87	80	62
60	39	43	68	66	41	47	77	18	-1	0	0	0	-1	0	1	18	22	37	56	68	109	103	77
68	42	29	33	28	21	40	73	-1	0	0	0	0	-1	0	0	24	35	55	64	81	104	113	92
83	52	22	14	20	33	52	69	0	0	0	0	0	0	0	0	49	64	78	87	103	121	120	101
98	63	24	12	32	58	71	70	-1	0	0	0	0	0	-1	0	72	92	95	98	112	100	103	99

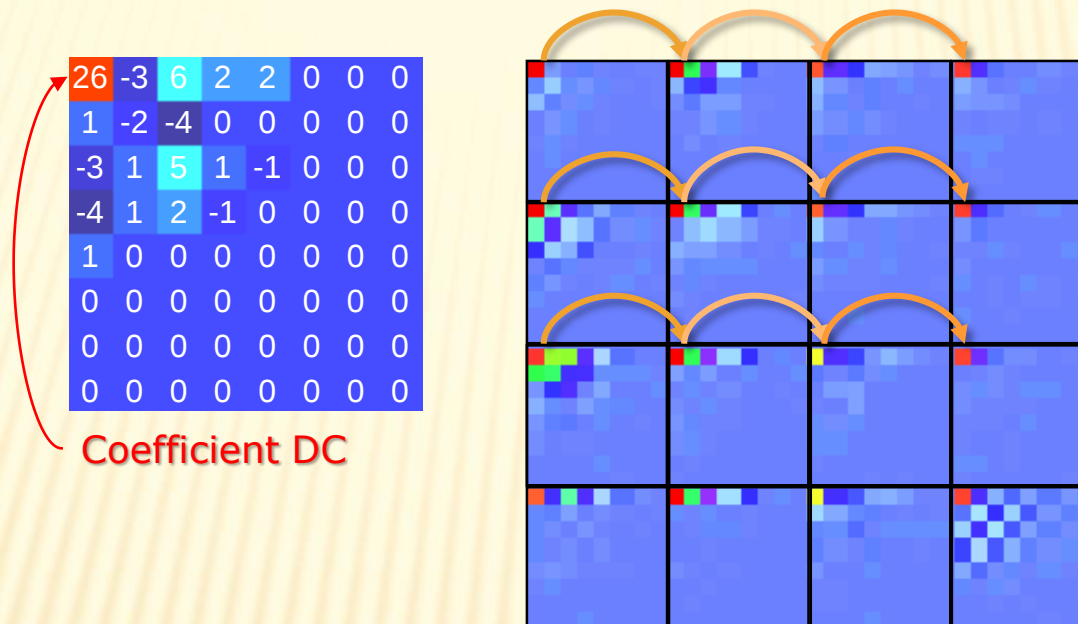
Bloc image
coefficients DCT du bloc
 $K \star$
QM

+ seuillage et quantification en une seule opération

$$X_{\text{DCTQ}}(k,l) = \text{NINT} \left(\frac{X_{\text{DCT}}(k,l)}{\text{QM}(k,l)} \right)$$

- ✗ QM : matrice de normalisation dont les éléments indiquent l'importance visuelle relative des coefficients DCT
- ✗ NINT : opération d'arrondi à l'entier le plus proche
- ✗ K : valeur de réglage

✗ Codage des coefficients quantifiés

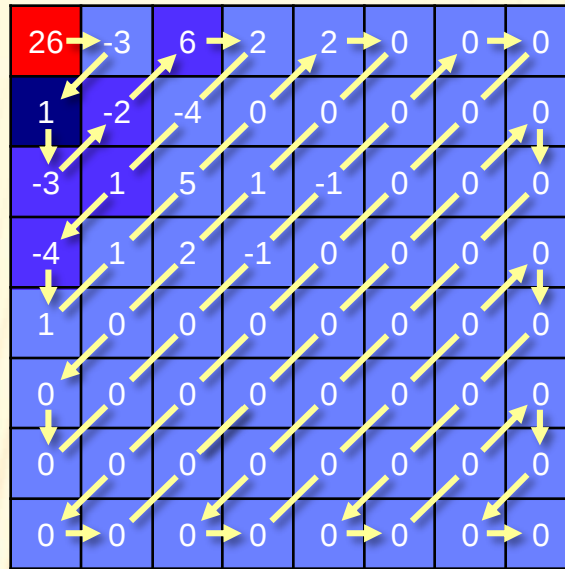


+ Coefficients DC («valeur moyenne»)

- ✗ codés à part par une approche DPCM
- ✗ Prédiction d'ordre 1 en utilisant le coefficient DC du bloc voisin 8x8 de gauche

+ Autres coefficients

Parcours
en
ZIGZAG



→ Transformation du bloc 8x8 en un vecteur de 64 éléments

26 -3 1 -3 -2 6 2 -4 1 -4 1 1 5 0 2 0 0 1 2 0 0 0 0 0 -1 -1 EOB

- × Minimisation du nombre de plages de zéros
- × Maximisation de la longueur des plages

Codage des zéros : par plages

→ Pour chaque coefficient, on code le couple :

(longueur de plage de zéros, amplitude) = (run, level)

avec un codage entropique (Huffman) : Variable Length Coding VLC

26 -3 1 -3 -2 6 2 -4 1 -4 1 1 5 0 2 0 0 1 2 0 0 0 0 0 -1 -1 EOB

(0,-3);(0,1);....;(1,2);(2,1);(0,2);(5,-1);(0,-1);EOB

Couple (Run, Level)	VLC
(1,1)	00
(1,-1)	011
(1,2)	101
(1,-2)	0101
.	.
.	.
(1,12)	0100001
(1,-12)	1100000
.	.

× exemple



Image originale



T=10:1, PSNR=34.9 dB

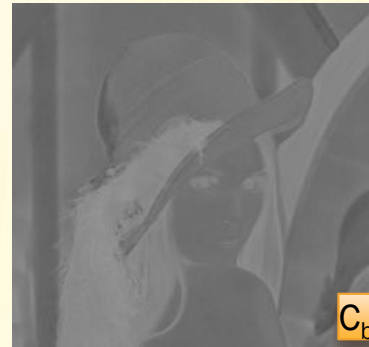


T=20, PSNR=32.2 dB

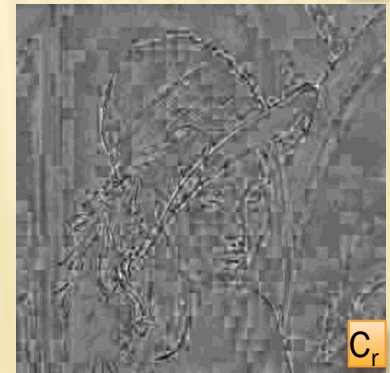
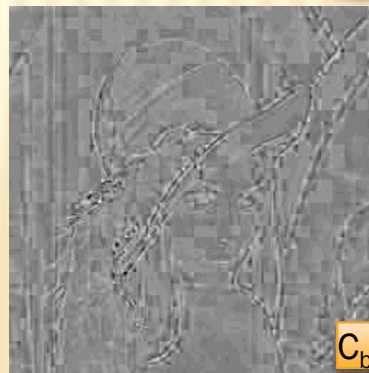
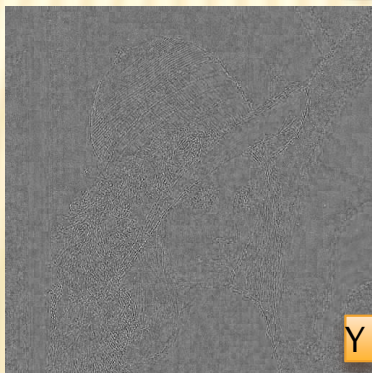


T=60, PSNR=27.3 dB

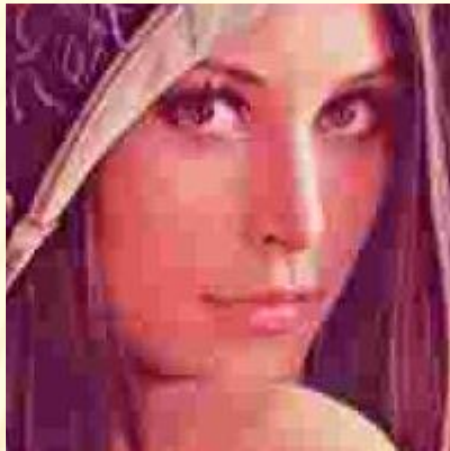
× Exemple



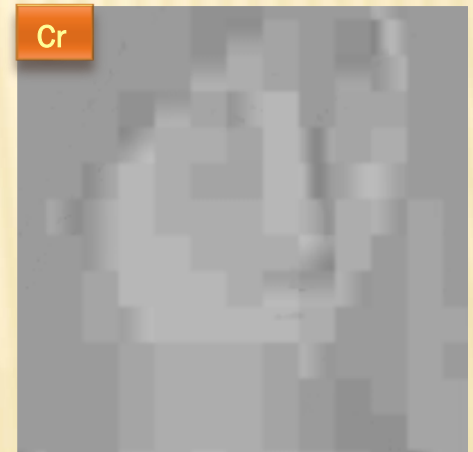
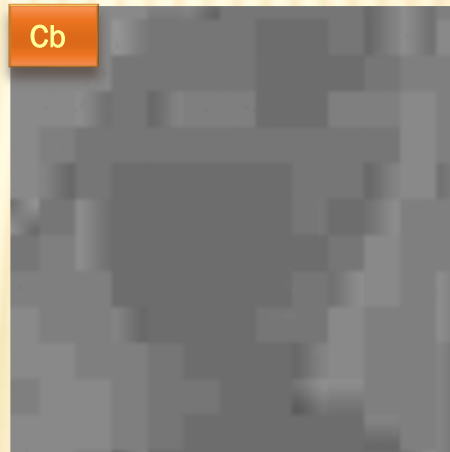
Images d'erreur
recadrées



✗ Exemple



Détail de l'image Lena
après compression JPEG
avec un taux de 82:1
→ moins de 0.29 bit/pixel



✗ Conclusion sur JPEG

- + Format très répandu
 - 80% des images sur le *Web* seraient codées avec JPEG
- + Outil universel de compression plus ou moins adapté à tout type d'images naturelles
- + Taux de compression :
 - ✗ jusqu'à 25-30 sur des images naturelles 24 bits, sans dégradation majeure
 - ✗ Jusqu'à 8-10 en monochrome
- + Résolution unique
- + Qualité unique
- + Débit non maîtrisé
- + Pas de possibilité sans perte
- + Pas de gestion de ROI, de codage d'une partie de l'image
- + Effets de bloc pour des taux élevés
- + Pas de robustesse aux erreurs de transmission

JPEG2000

Projet défini en 1996

✖ Objectifs

- Codage de différents types d'images (binaire, NG, couleur, multi-composante)
- Codage d'images de nature différente (naturelle, médicale, satellitaire, texte, graphiques, images de synthèse)
- Débits très faibles avec une bonne qualité subjective

✖ Fonctionnalités visées

- + Codage NG et binaire
- + Codage avec ou sans perte
- + Transmission progressive en résolution et en qualité
- + Accès aléatoire aux régions de l'image
- + Robustesse aux erreurs de transmission (embedded bitstream)
- + Sécurité (accès à l'image, identification de l'image)
- + Codage ROI
- + Ajout de données à l'image codée (texte, son, adresse WEB,...)

✖ Une norme en 12 parties

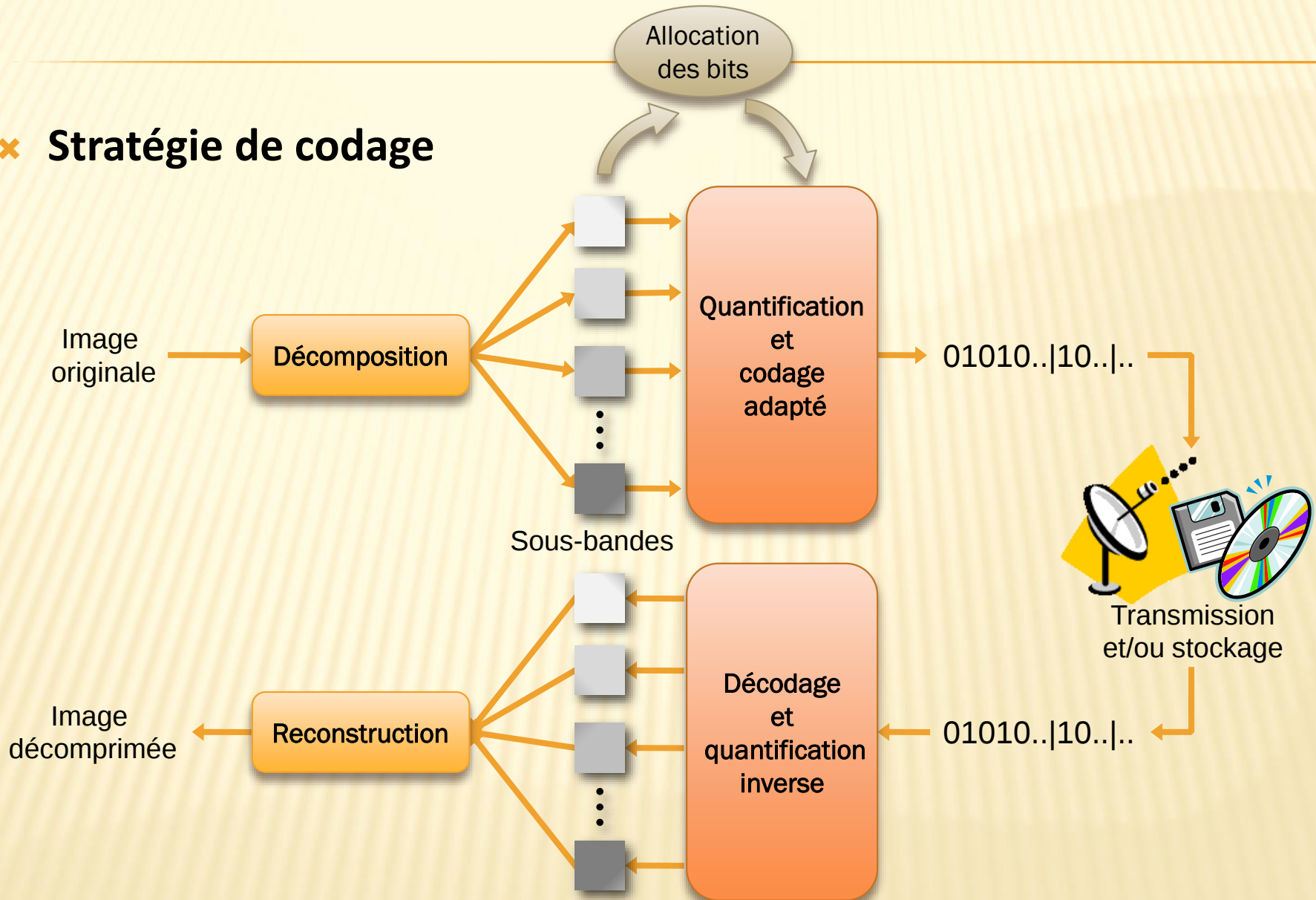
- + Part 1, Core coding system (*intended as royalty and license-free - NB NOT patent-free*)
- + Part 2, Extensions (*adds more features and sophistication to the core*)
- + Part 3, Motion JPEG 2000
- + Part 4, Conformance
- + Part 5, Reference software (*Java and C implementations are available*)
- + Part 6, Compound image file format (*document imaging, for pre-press and fax-like applications, etc.*)
- + Part 7 has been abandoned
- + Part 8, JPSEC (*security aspects*)
- + Part 9, JPIP (*interactive protocols and API*)
- + Part 10, JP3D (*volumetric imaging*)
- + Part 11, JPWL (*wireless applications*)
- + Part 12, ISO Base Media File Format (*common with MPEG-4*)

➔ Partie 1 : format de base .jp2

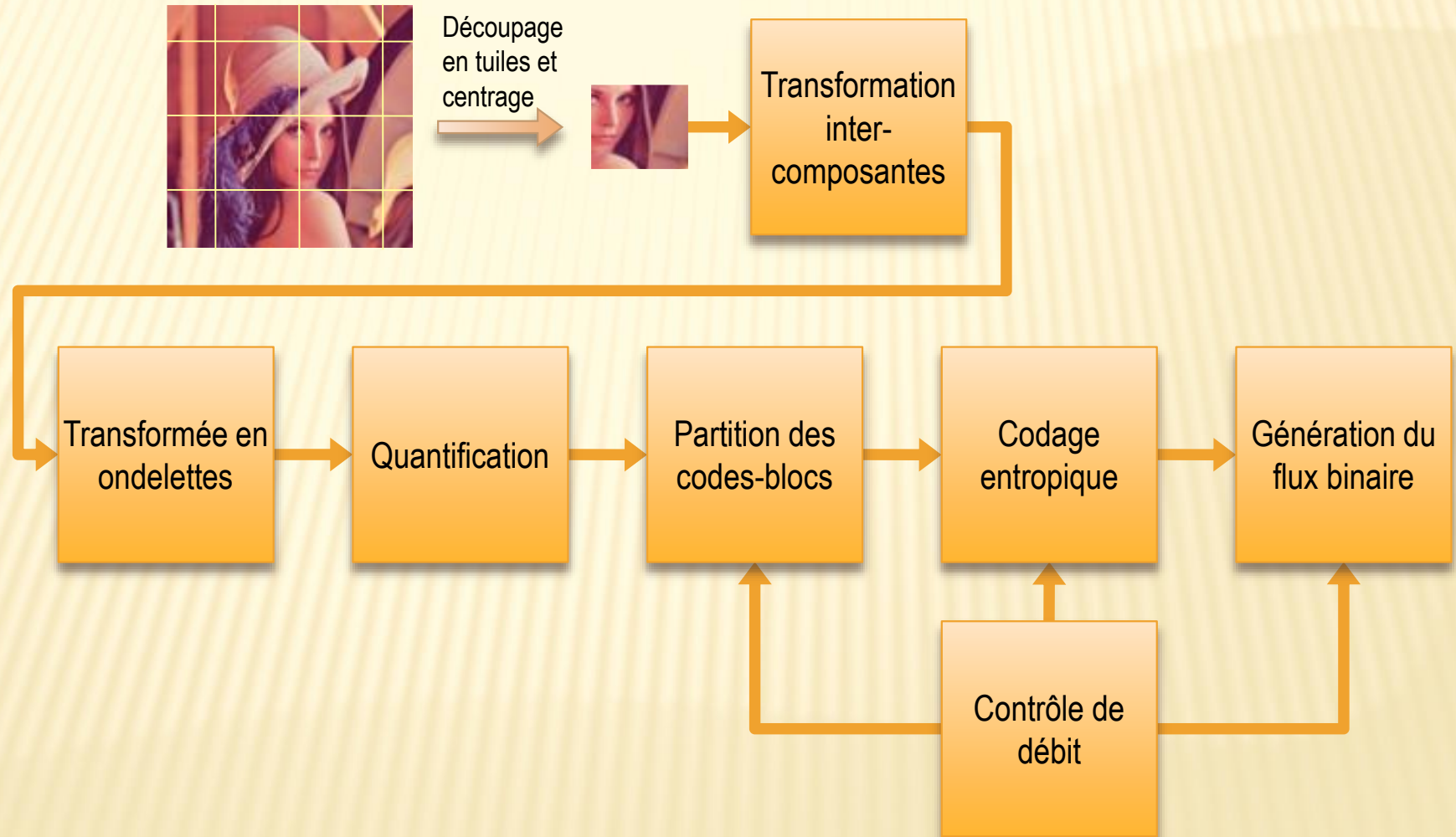
✗ Evolution des principales méthodes

- + Woods, 1986 : 1ère méthode de codage sous bandes
- + Shapiro, 1993 : ZWT : Zero wavelet trees
- + Said & Pearlman, 1996 : SPIHT : Set Partitioning In Hierarchical Trees
- + Taubman, 2000 : EBCOT : Embedded Block Coding with Optimized Truncation
 - méthode adoptée pour JPEG 2000

✖ Stratégie de codage



✗ Schéma général



✗ Découpage en tuiles et centrage

- + Découpage en sous-images rectangulaires disjointes traitées indépendamment
- Réduction de la complexité
- Possibilité de décodage d'une partie quelconque de l'image
- Artefacts au niveau des raccords



- + Ajout d'une composante continue pour centrer la gamme dynamique autour de zéro



✗ Transformation inter-composantes

→ Passage de RVB à $Y C_b C_r$

✗ RCT : Reversible Color Transform à valeurs entières pour le codage sans perte

$$\begin{cases} Y = \left\lfloor \frac{1}{4}(R + 2G + B) \right\rfloor \\ C_b = B - G \\ C_r = R - G \end{cases} \quad \begin{cases} G = Y - \left\lfloor \frac{1}{4}(C_b + C_r) \right\rfloor \\ R = C_r + G \\ B = C_b + G \end{cases}$$

- × ICT : Irreversible Color Transform pour la compression avec perte
(transformation YC_bC_r classique)

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ V \\ B \end{bmatrix}$$

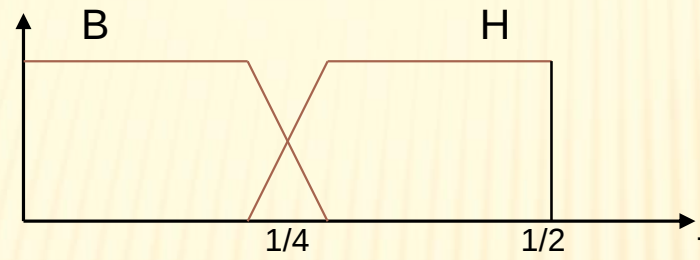
$$\begin{bmatrix} R \\ V \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.4021 \\ 1 & -0.3441 & -0.7142 \\ 1 & 1.7718 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix}$$

-
- + **Usage non obligatoire**
 - + **Si aucune transformation n'est utilisée, les différentes composantes ne sont pas nécessairement échantillonnées de la même manière**
 - + **Remarques**
 - Le format jp2 accepte uniquement des images RVB ou indexées.**
 - Il autorise l'enregistrement de profils ICC**



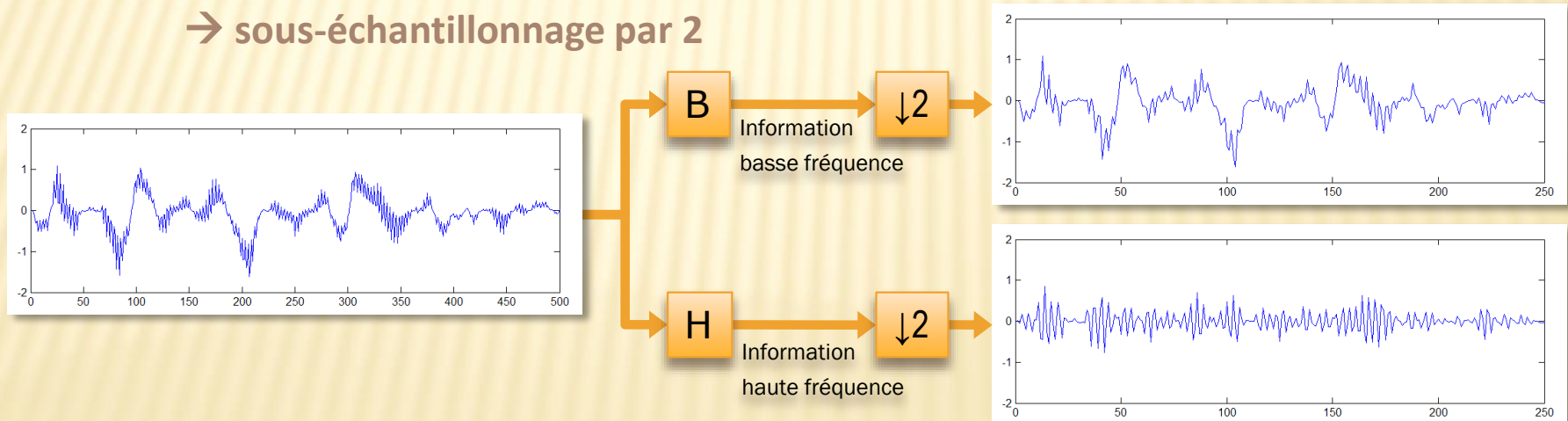
✗ Transformée en ondelettes : principe 1D

→ décomposition passe-haut, passe-bas



→ Deux signaux à bande deux fois plus étroite

→ sous-échantillonnage par 2



THÉORIE DES ONDELETTES

✗ Ψ : ondelette mère

$\int \Psi(t)dt = 0$, la fonction $\Psi(t)$ oscille et s'amortit, d'où le nom d'ondelette.

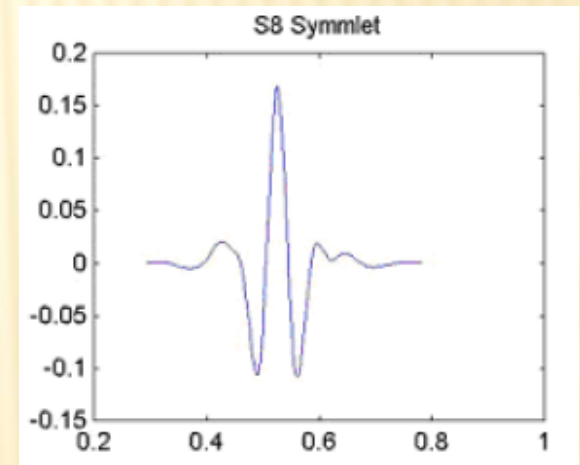
✗ Ψ_{ab} : famille d'ondelettes

$$\Psi_{ab} = |a|^{-1/2} \Psi\left(\frac{t-b}{a}\right)$$

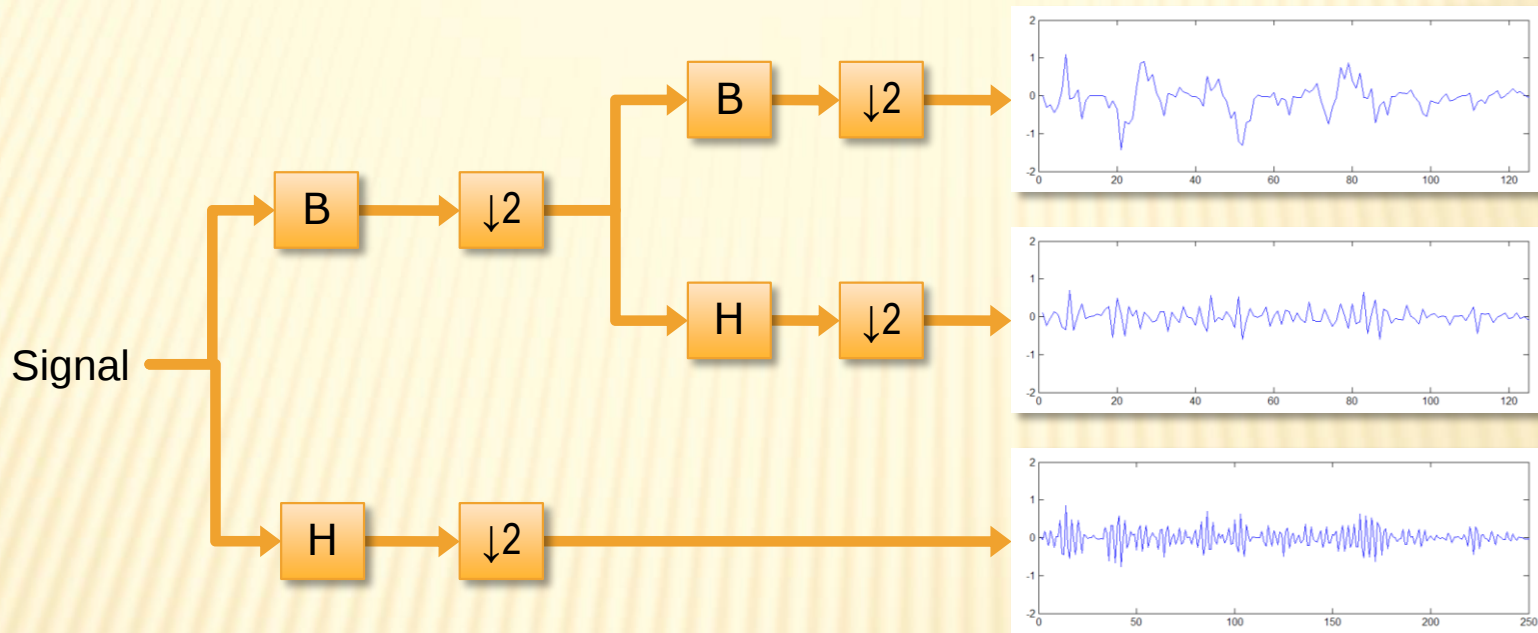
✗ Transformée en ondelettes

$$W_f(a, b) = \langle \Psi_{ab}, f \rangle$$

$$= |a|^{-1/2} \int_{\mathbb{R}} f(t) \Psi\left(\frac{t-b}{a}\right) dt$$



- + Le processus peut être appliqué de nouveau sur le signal basse-fréquence



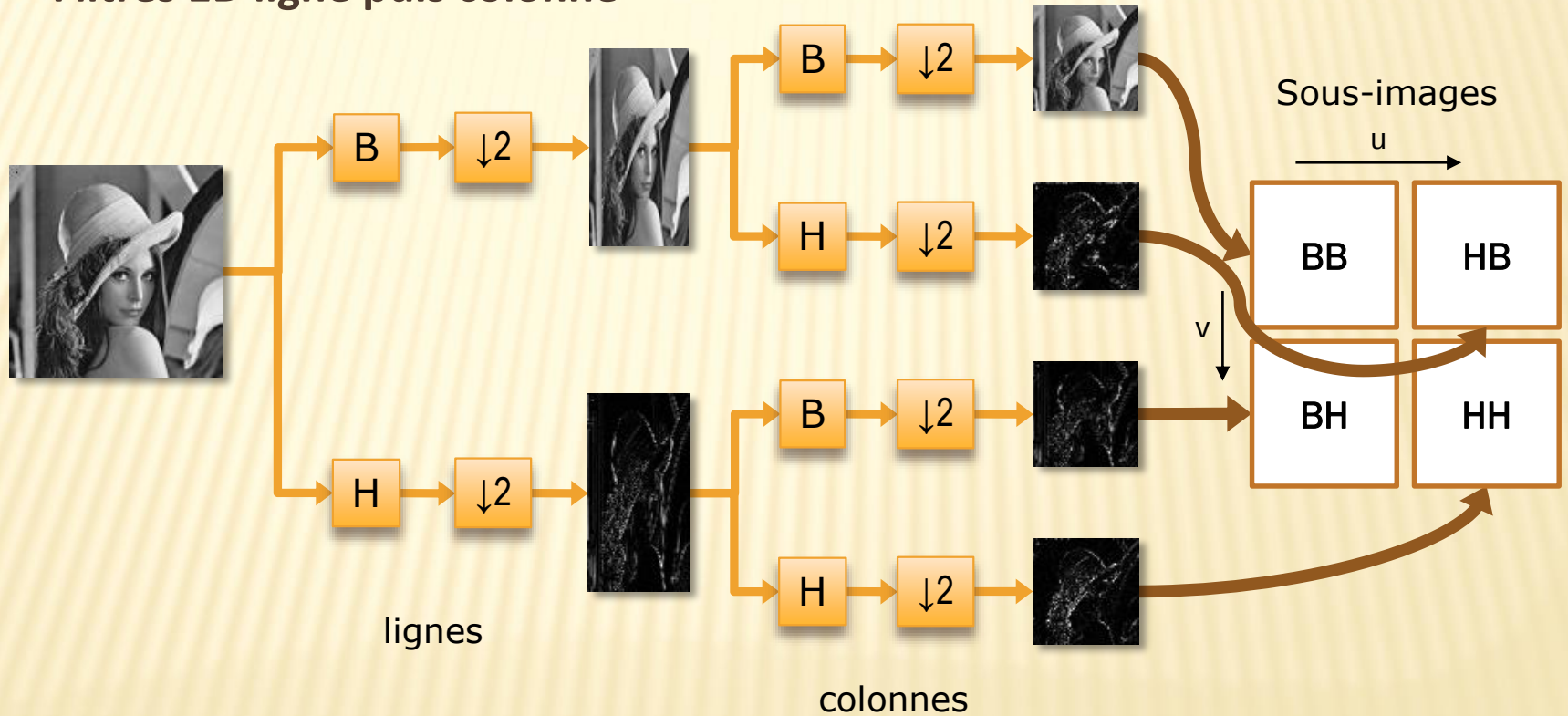
→ Décomposition pyramidale par ondelettes

- + Reconstruction parfaite en appliquant le schéma inverse

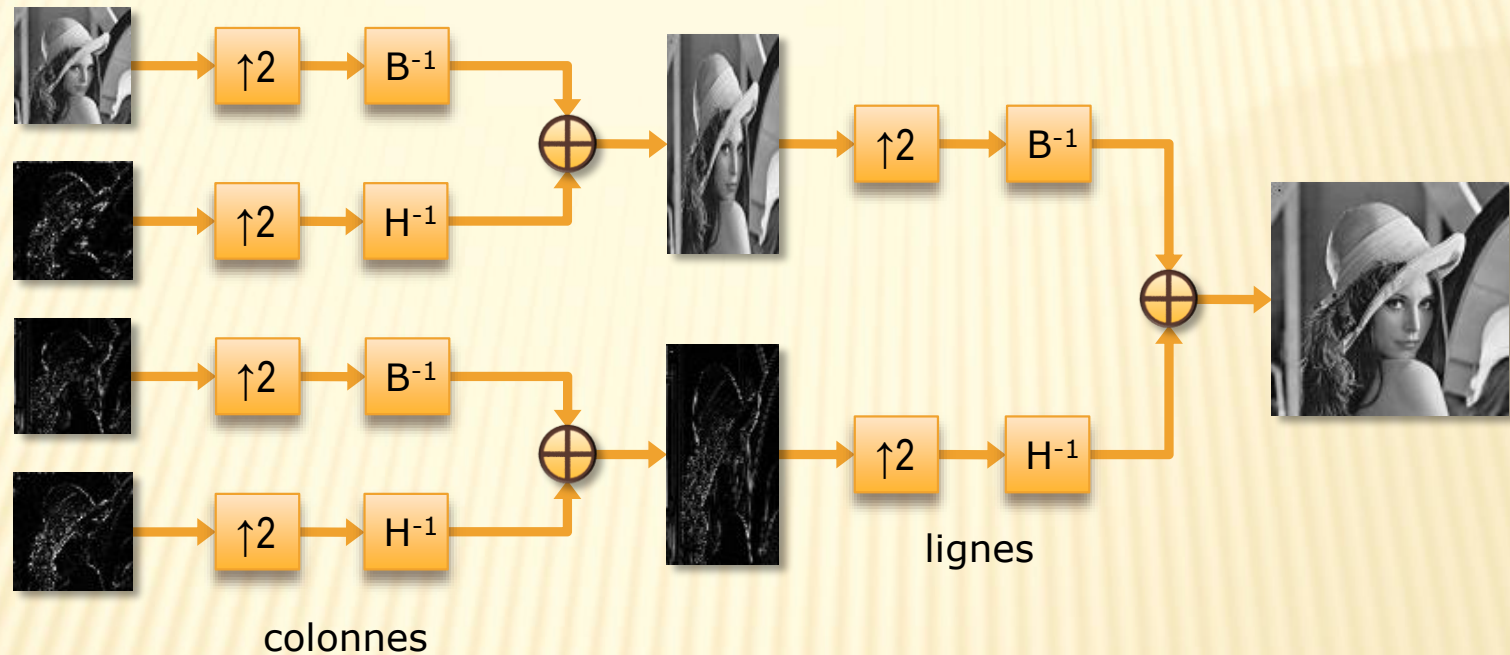
+ Transformée en ondelettes 2D

Esteban/Galland 1977 - Woods/O'Neil 1986 - ... - Mallat (1989)

Filtres 1D ligne puis colonne



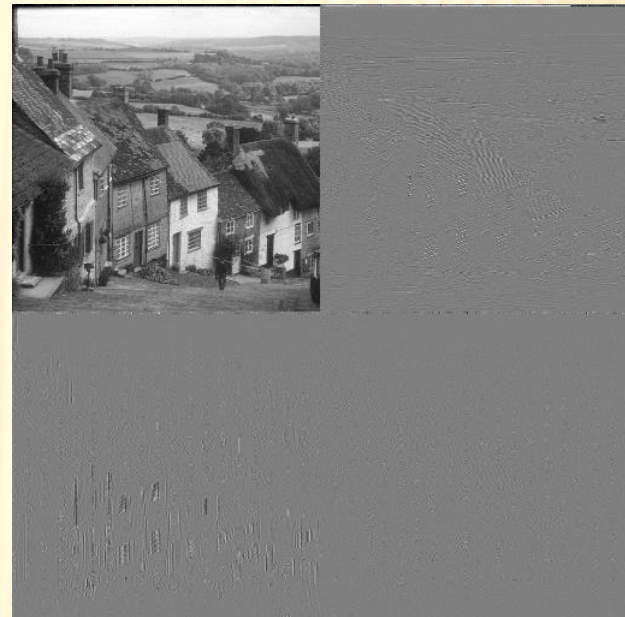
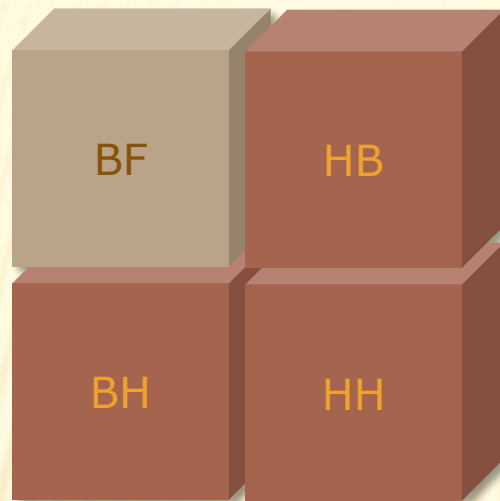
✗ Reconstruction



✗ Deux possibilités dans JPEG2000

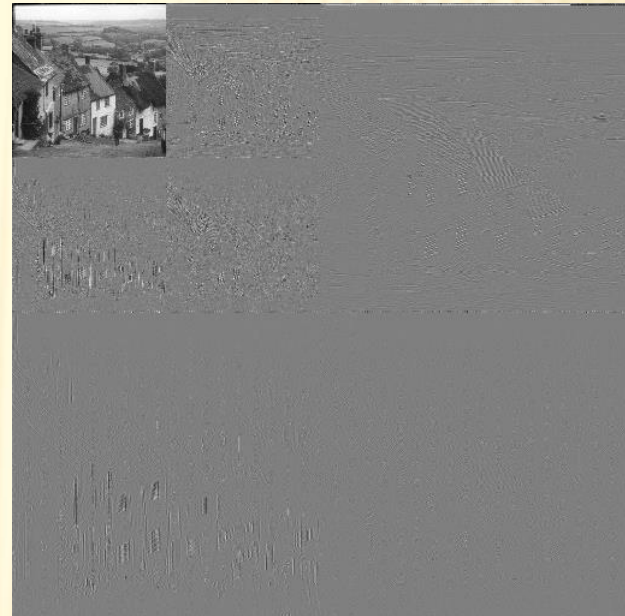
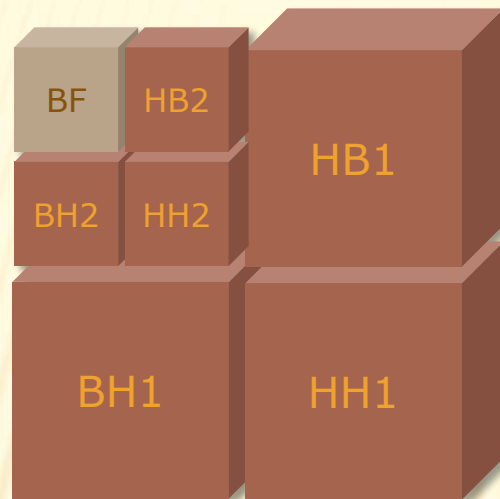
- + ondelette 9/7 à coefficients réels de Daubechies pour la compression avec pertes
- + ondelette 5/3 à coefficients entiers de Le Gall principalement pour le codage sans perte

× Exemple



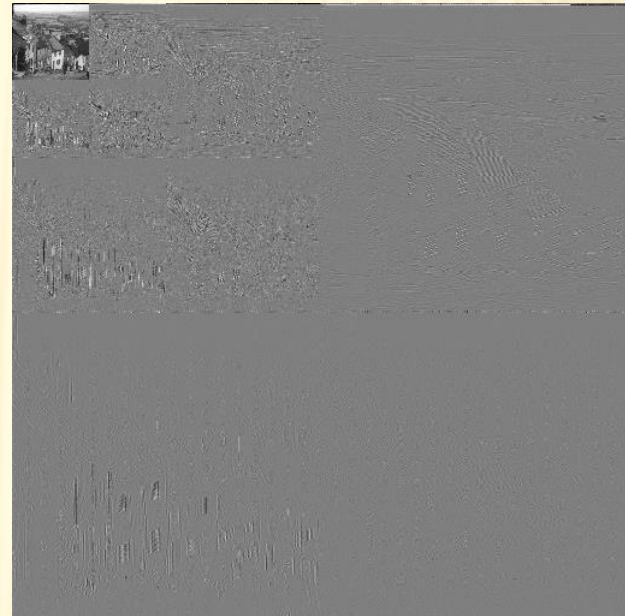
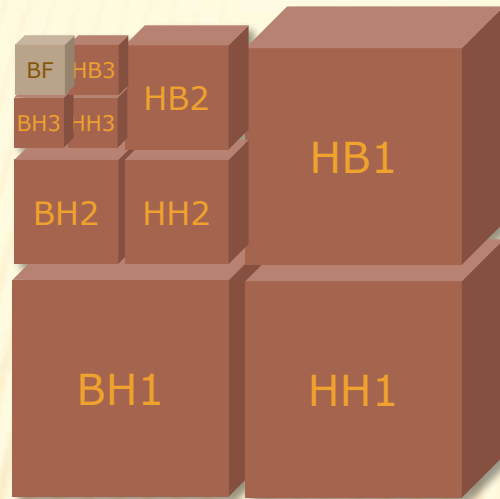
Niveau 1

× Exemple



Niveau 2

× Exemple



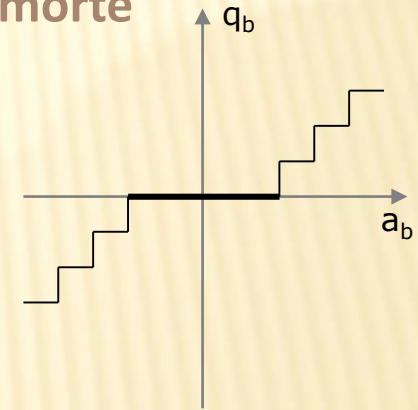
Niveau 3



✗ Quantification

+ Quantification scalaire des coefficients avec zone morte

$$q_b(u,v) = \text{sign}(a_b(u,v)) \left\lfloor \frac{|a_b(u,v)|}{\Delta_b} \right\rfloor$$



+ Pas de quantification Δ_b

- ✗ transmis pour chaque sous bande
- ✗ égal à 1 pour le codage sans perte
- ✗ dépend de la dynamique de la sous bande à quantifier

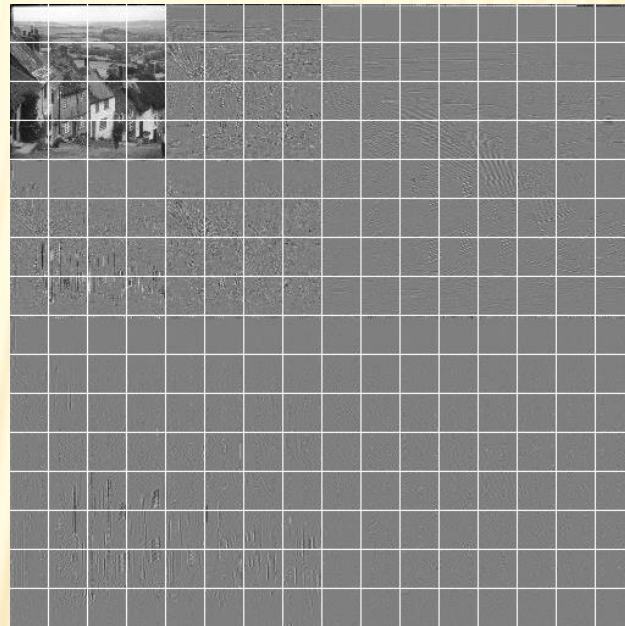
+ Zone morte

→ Seuil sur les coefficients d'ondelette



✗ Partition des codes-blocs

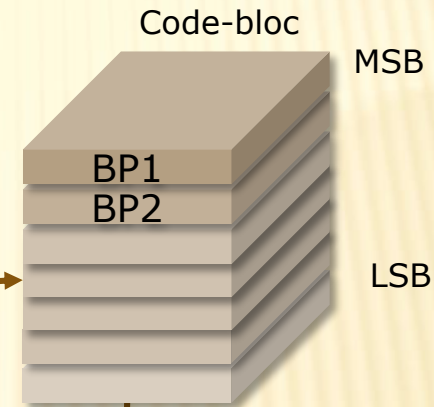
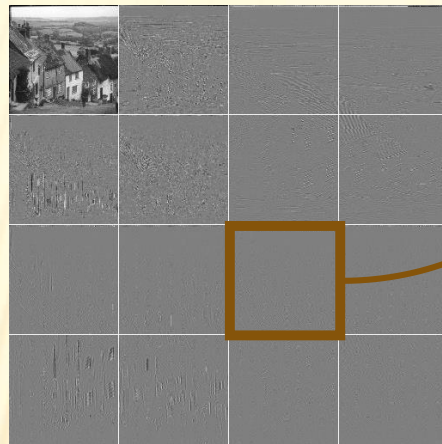
- + Création de blocs rectangulaires (codes-blocs) dont la taille est une puissance de 2 (typiquement 32x32 ou 64x64)





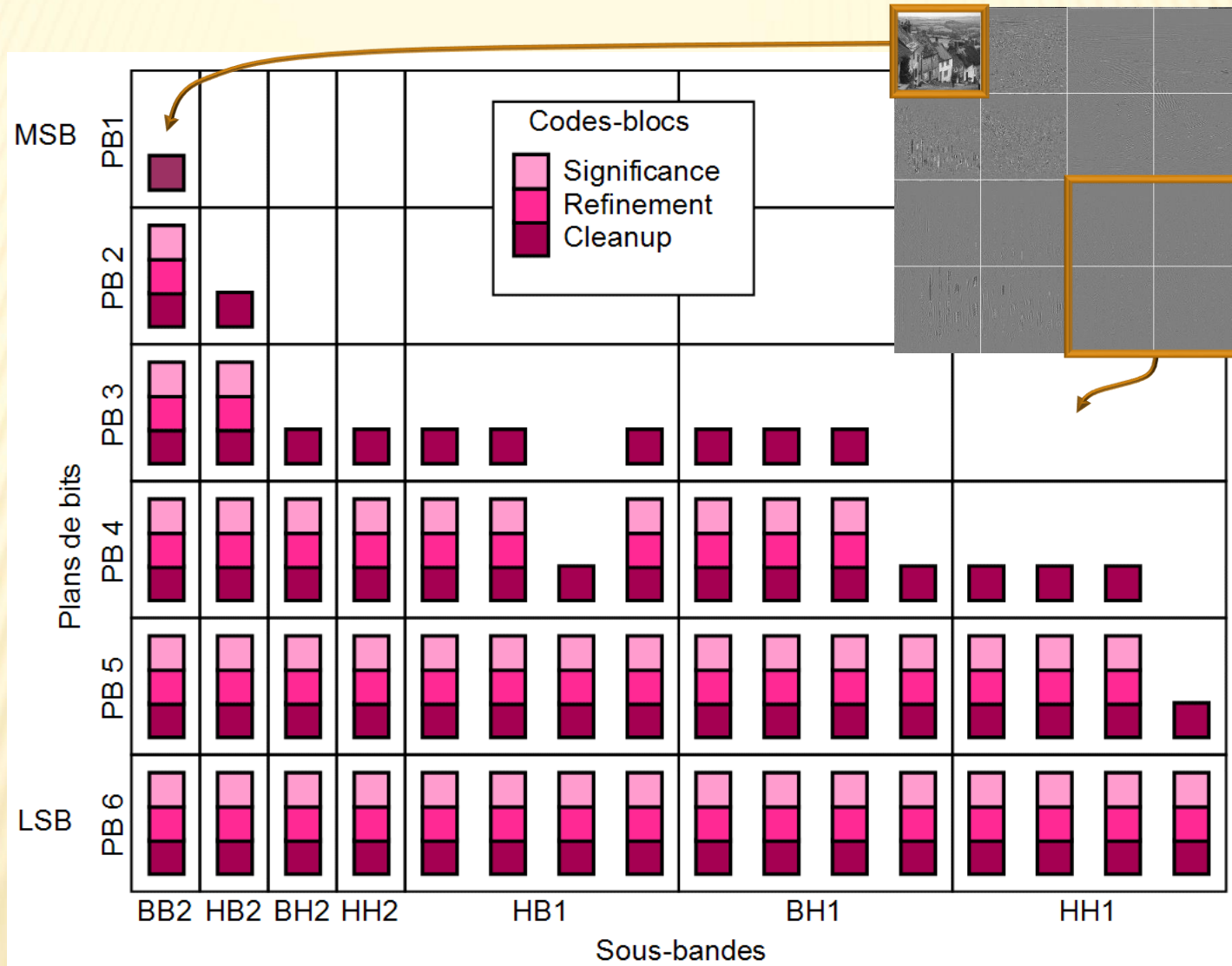
✗ Codage individuel des codes-blocs

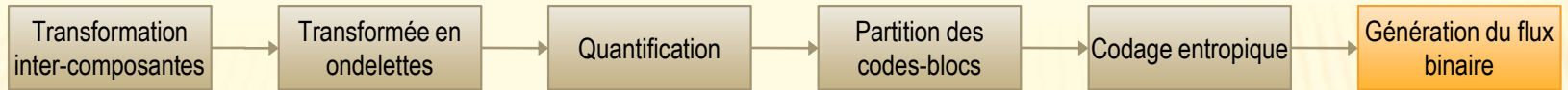
Sous-bandes quantifiées
Coefficients regroupés en code-blocs



Codage en trois passes
par codage arithmétique contextuel

d_0	v_0	d_1
h_0	x	h_1
d_2	v_1	d_3





✗ Flux binaire

+ Organisation en couches

- différents niveaux de résolution
- transmission indépendante

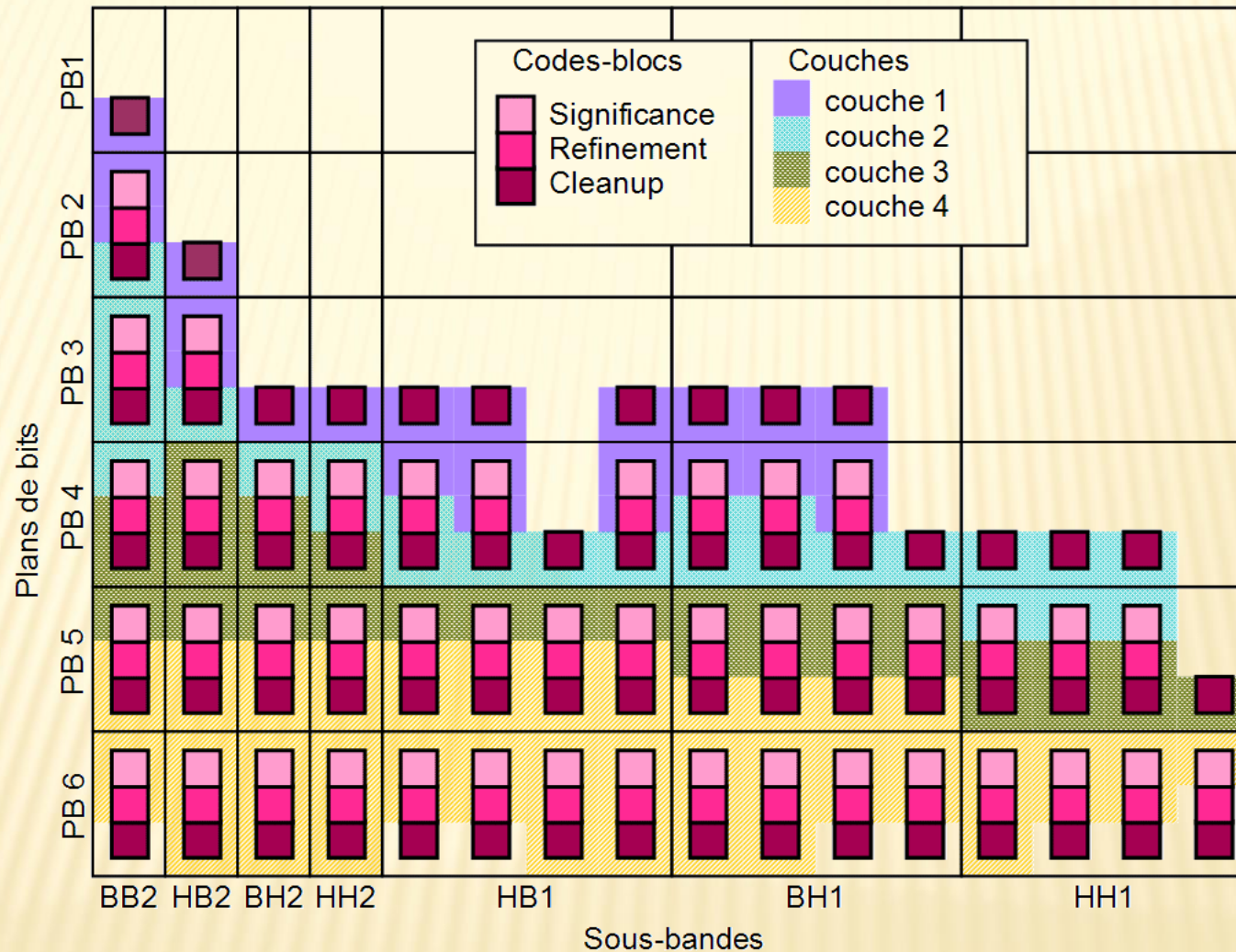
+ Chaque couche est constituée de :

- ✗ un certain nombre de passes de codage par plan binaire
- ✗ un certain nombre de plans binaires d'un code-bloc
- ✗ pour tous les code-blocs d'une sous-bande
- ✗ pour toutes les sous-bandes associées à l'image ou à la tuile de l'image

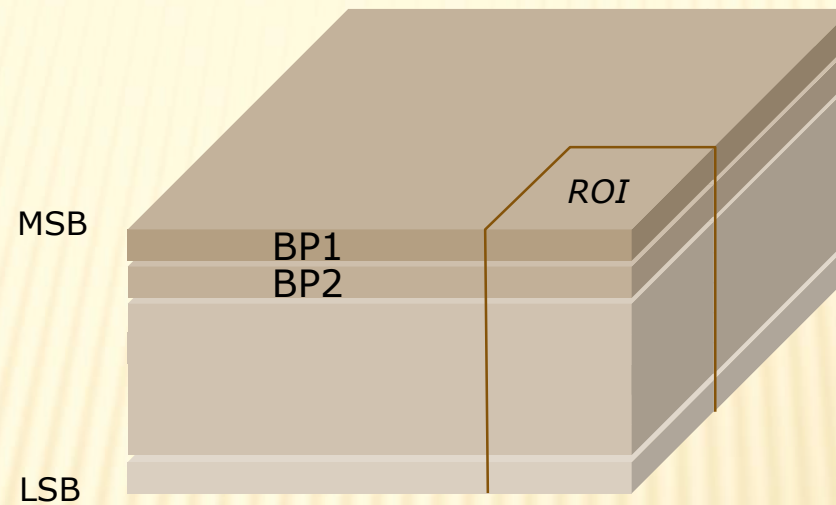
+ les couches sont indépendantes les unes des autres

+ le décodage peut se faire à chaque couche

✕ Exemple

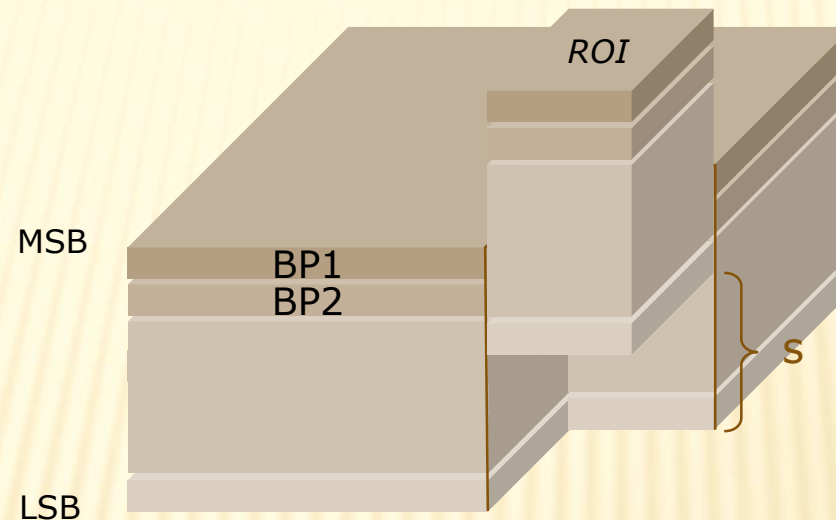


✗ Codage des régions d'intérêt



✗ Codage des régions d'intérêt

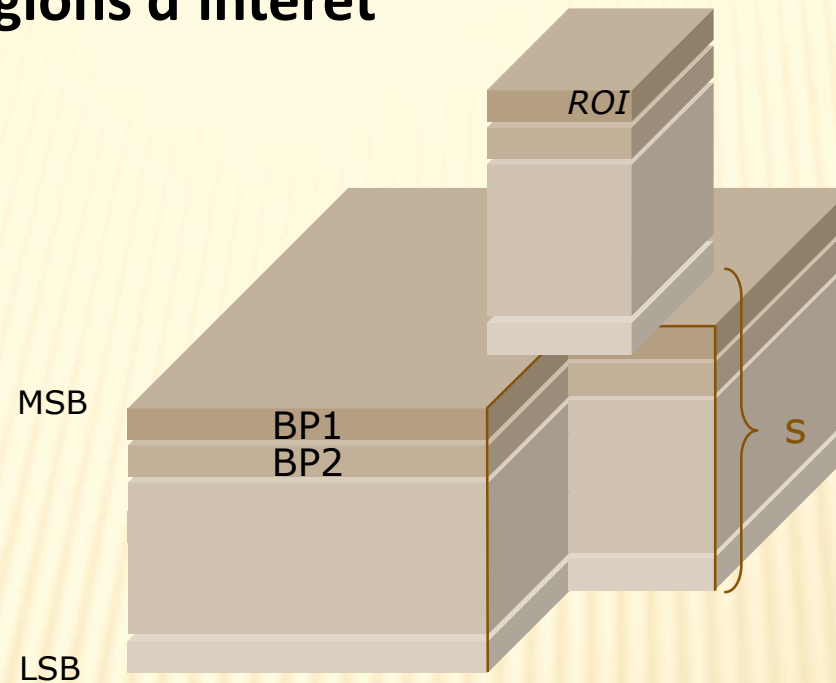
+ « General scaling »



- ✗ Le décalage s contrôle l'écart de qualité ROI/fond
- ✗ Possibilité de décalages différents pour les différentes régions
- ✗ Nécessité de transmettre les masques des régions

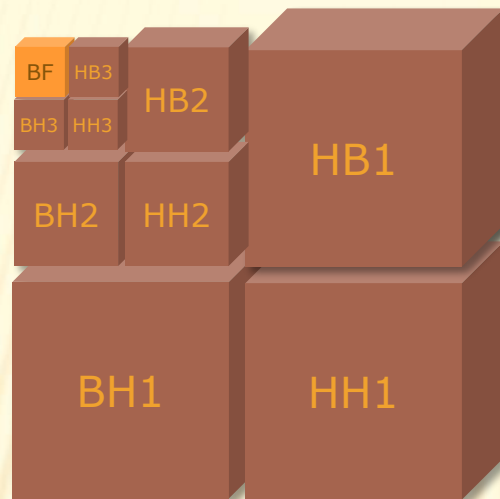
✗ Codage des régions d'intérêt

+ « Max shift »



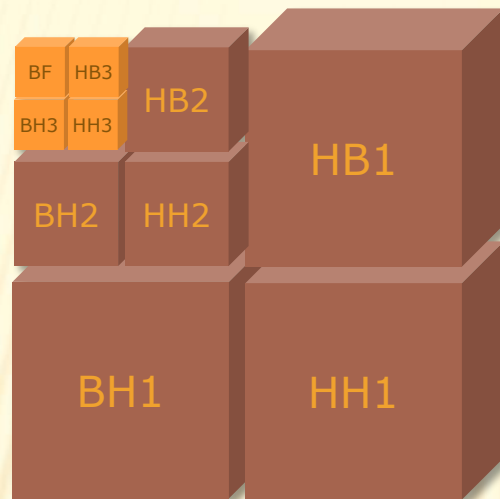
- ✗ s supérieur au plus grand des coefficients du fond
- ✗ → Plus besoin de transmettre les masques des régions

✖ Exemple : Transmission progressive par résolution



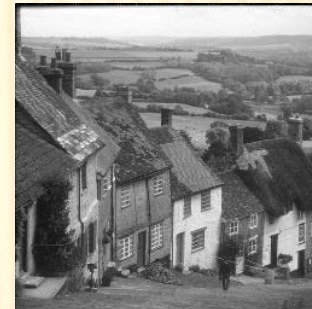
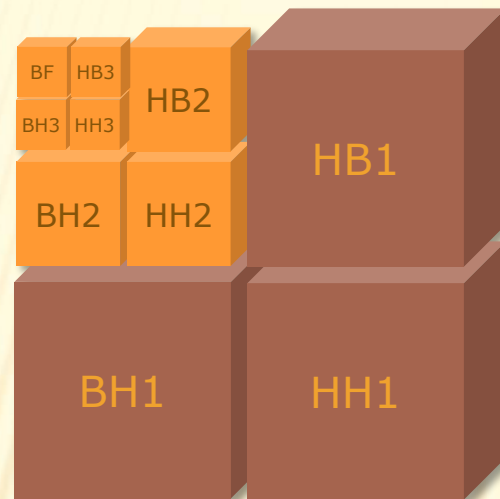
Résolution 3 : image 64x64

✗ Exemple : Transmission progressive par résolution



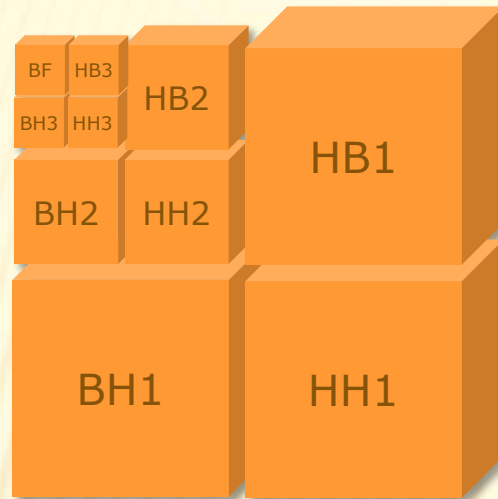
Résolution 2 : image 128x128

✖ Exemple : Transmission progressive par résolution



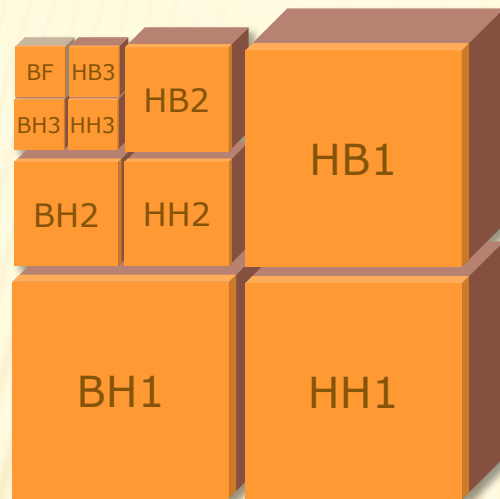
Résolution 1 : image 256x256

✗ Exemple : Transmission progressive par résolution



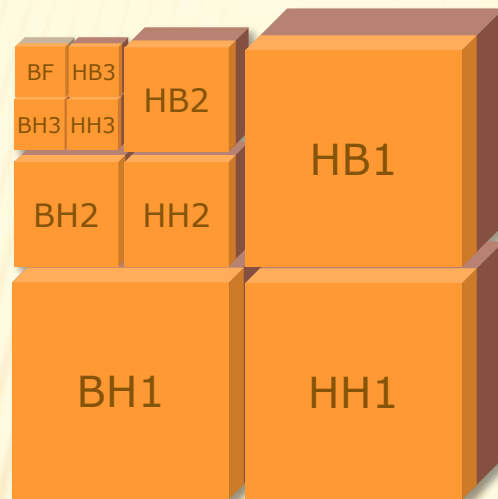
Résolution haute : image originale 512x512

✗ Exemple : Transmission progressive par qualité



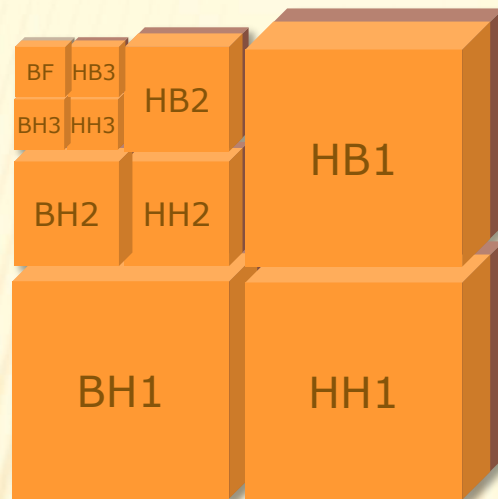
Qualité 1 : 0.03 bits par pixel

✗ Exemple : Transmission progressive par qualité



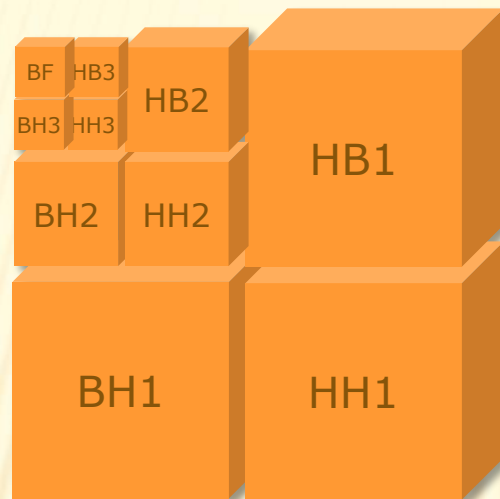
Qualité 2 : 0.15 bits par pixel

✖ Exemple : Transmission progressive par qualité



Qualité 3 : 0.45 bits par pixel

✗ Exemple : Transmission progressive par qualité

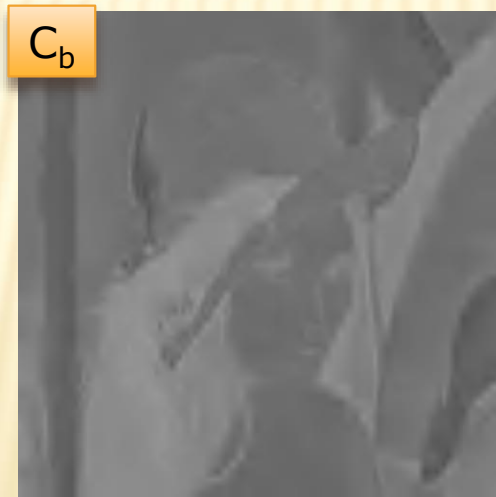
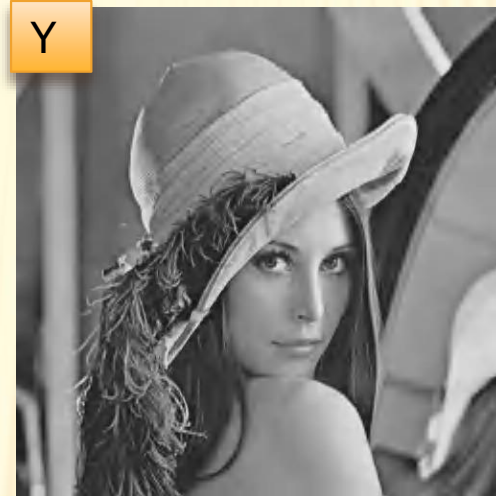


Qualité maximale : 4,84 bits par pixel

✕ Exemple : Régions d'intérêt



✕ Exemple : composantes couleur



0.23 bits
par pixel

✕ Exemple : comparaison JPEG ↔ JPEG2000



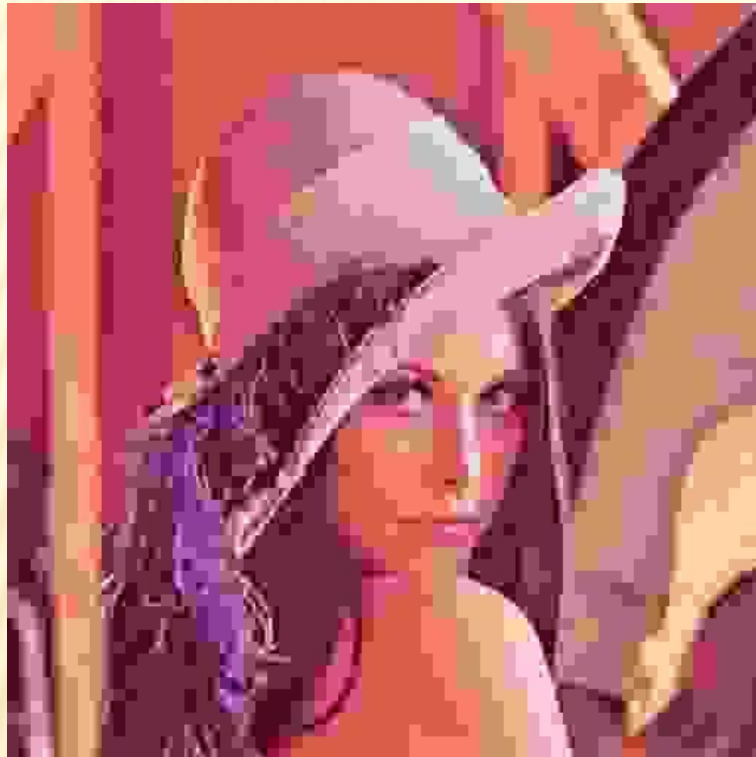
JPEG



JPEG2000

0.23 bit par pixel

✖ Exemple : comparaison JPEG ↔ JPEG2000



JPEG



JPEG2000

0.17 bit par pixel

✗ Quelques extensions

+ JPEG2000 Partie 2

- Davantage de flexibilité dans les décompositions en ondelettes
- Davantage de possibilités pour le codage des régions d'intérêt

→ Un autre format : .jpx prenant en charge

- ✗ Des images composées de différentes couches
- ✗ Des animations
- ✗ Des espaces de couleurs et des fonctions de conversion plus nombreux (nombre arbitraire de composantes, ...)
- ✗ Des meta-données plus riches

+ JPEG2000 Partie 3

- Séquences d'images : format .mj2

- + JPEG2000 Partie 6

- Documents multi-pages : format .jpm

- + JPEG2000 Partie 9

- Interactivité : protocole client-serveur JPIP

Parties 6,9, 10, 11 toujours en évolution

CONCLUSION

✗ Sur JPEG2000

- + un format complexe, mais flexible et offrant de multiples possibilités
- + Les services autour de la norme devront voir le jour
- + Les utilisateurs devront apprendre les multiples fonctionnalités

✗ Sur la compression des images couleur

- + Le plus souvent, les composantes couleurs sont quantifiées séparément
→ la spécificité des images couleur n'est pas réellement prise en compte
- + Marge de progression avec
 - ✗ Des approches de quantification plus adaptées (quantification vectorielle, par exemple)
 - ✗ Une meilleure prise en compte des critères psycho-visuels complets (textures, masquage, ...)

POUR TESTER ...

✕ <http://www.kakadusoftware.com/>

+ kdu_show → visualisation d'image compressées JPEG2000

+ kdu_compress → compression JPEG2000

Compression, codage et transmission d'objets 3D

F. Dupont



PRÉAMBULE

✗ Objectif de la compression 3D :

- Stockage et transmission plus rapide et moins cher
- Faire pour les données 3D, ce que JPEG, MPEG, MP3 et autres standards sont devenus pour l'image, la vidéo et le son
 - Passe par des étapes de standardisation
- Domaines d'applications
 - ✗ Jeux vidéo
 - ✗ Animations
 - ✗ Modélisation 3D (CAO)
 - ✗ E-commerce

SOURCES DE DONNÉES 3D

Application dans le domaine médical

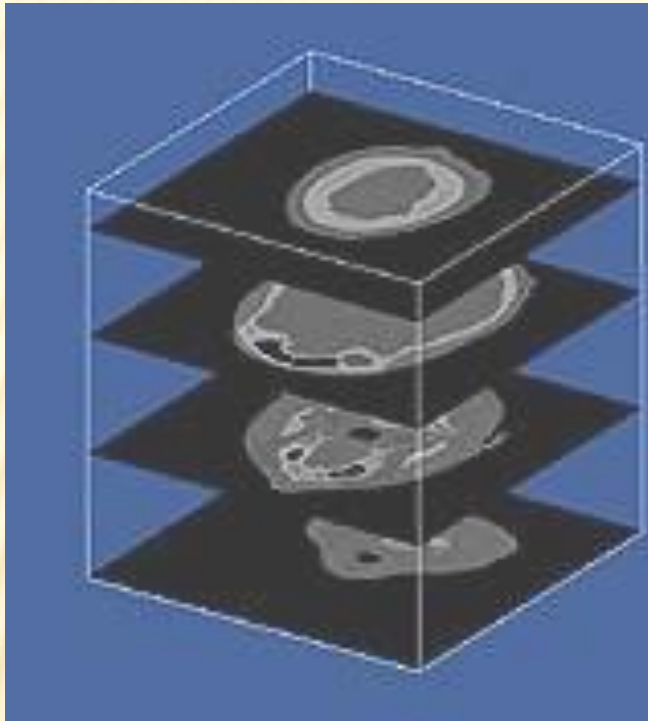
- ✗ Radiographie (tomo-densitométrie ou Scanner)
- ✗ Échographie (échotomographie ou échographie Doppler)
- ✗ RMN (résonance magnétique nucléaire)
- ✗ Imagerie nucléaire (scintigraphie, PET ou SPECT)

Application dans d'autres domaines

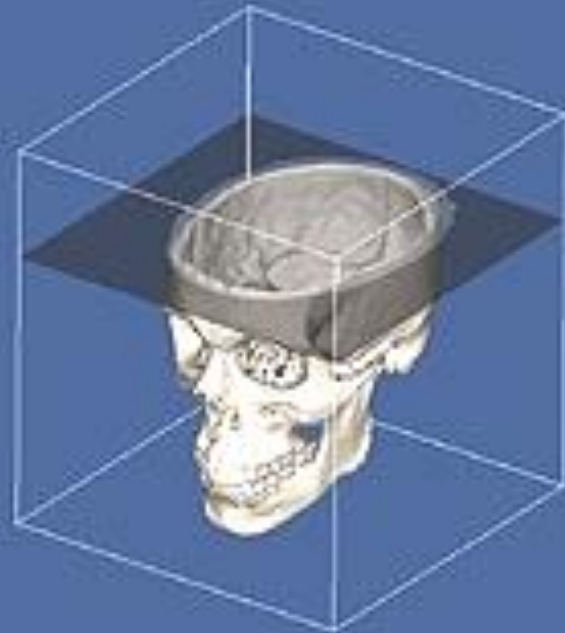
- ✗ Domaines des sciences de la Terre : Océanographie, géophysique, sismologie ou la météorologie.
- ✗ Industrie : techniques de contrôle non destructif (CND) (qualité de soudure, uniformité de conductivité).
- ✗ Industrie : conception assistée par ordinateur

SOURCES DE DONNÉES 3D

Empilement de coupes

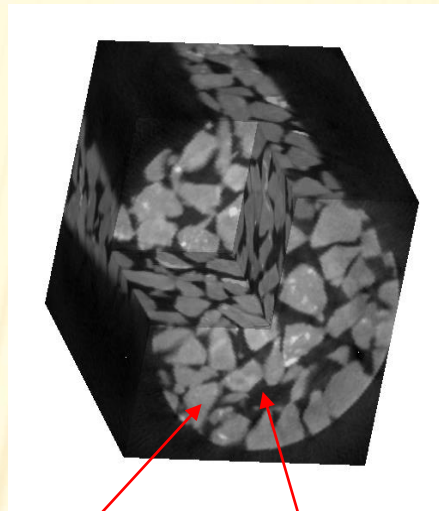
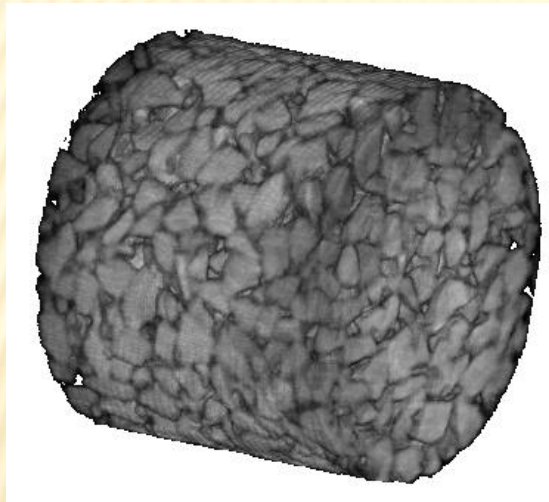


Isosurface



SOURCES DE DONNÉES 3D

Carotte cylindrique d'enrobé drainant

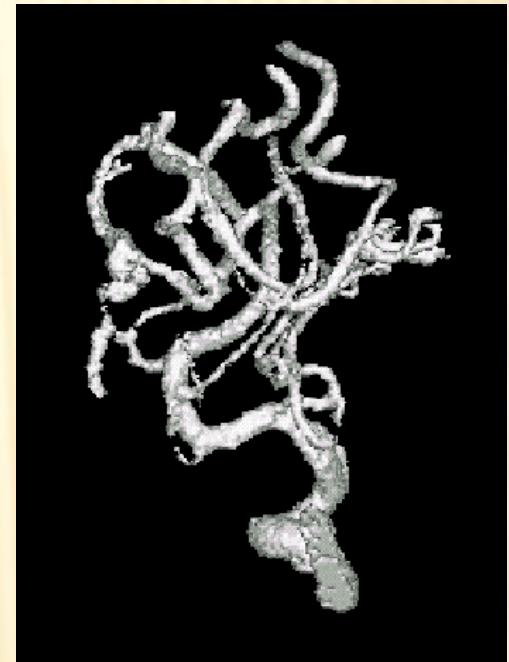


Granulats

Vide

Rendus volumique

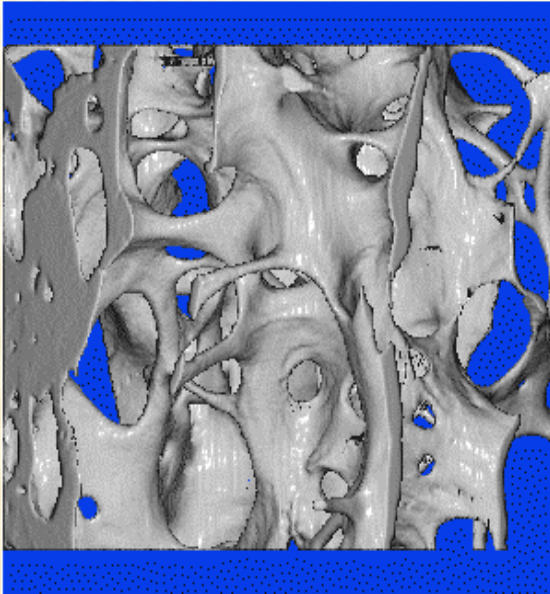
Réseau vasculaire



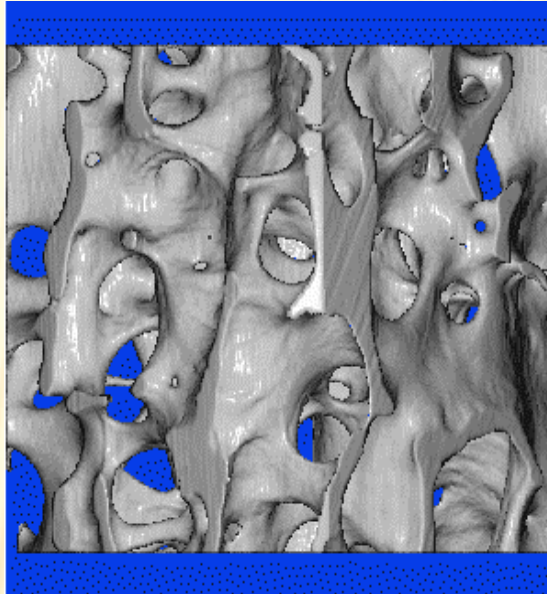
**Segmentation
+
Rendu surfacique**

SOURCES DE DONNÉES 3D : ESRF (SYNCHROTRON)

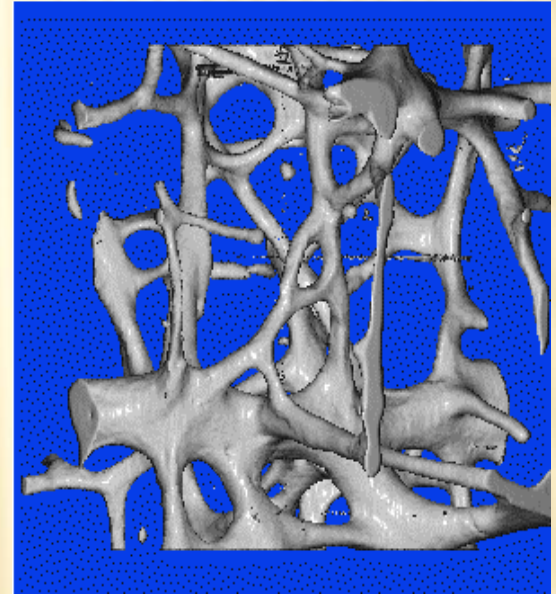
Échantillons d'os trabéculaire
Images de Vertèbres, précision de qq microns



33 ans

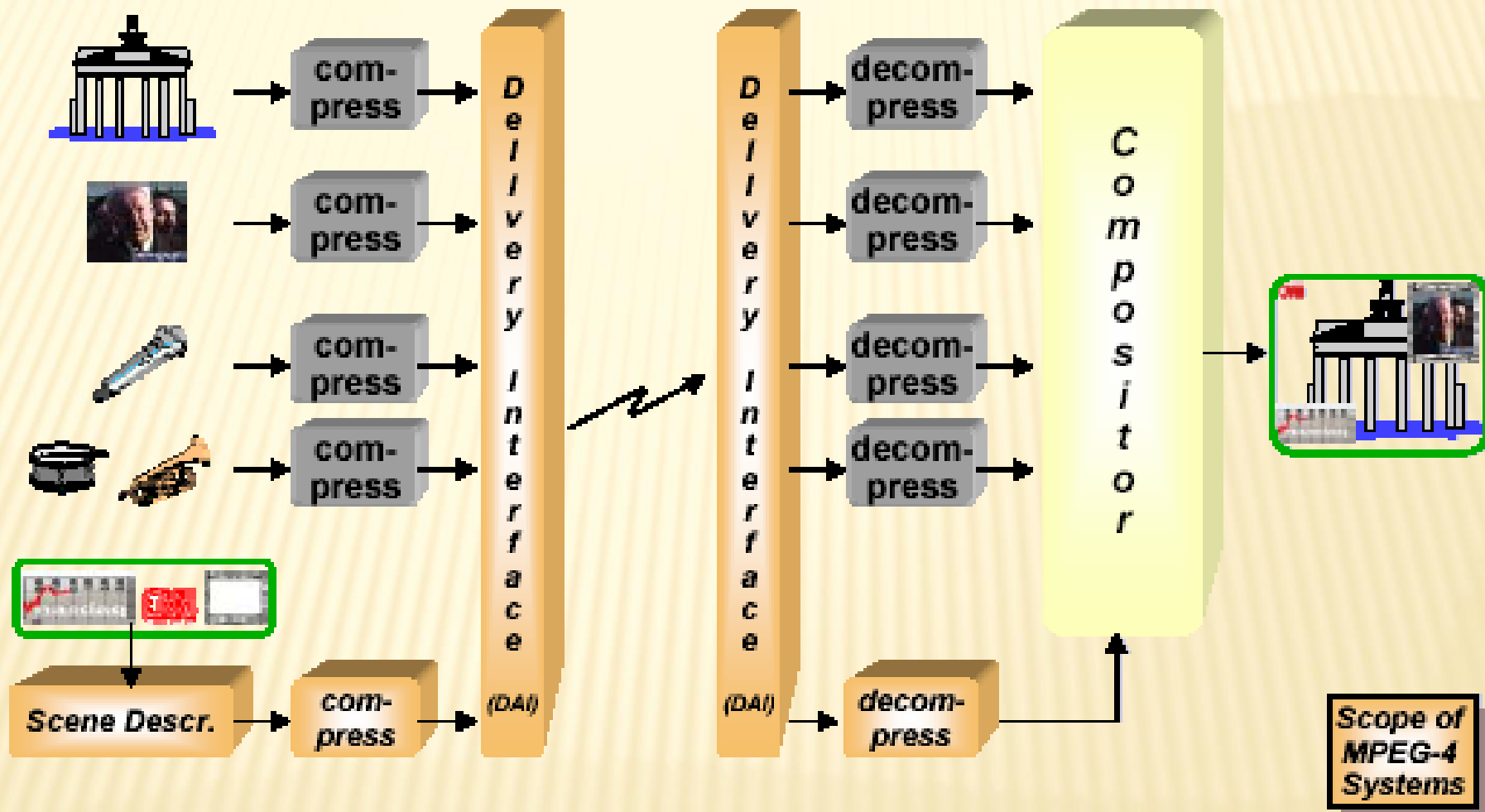


55 ans



72 ans

MPEG-4 COMPRESSION BASÉE CONTENU



MPEG-4 MIXAGE 2D-3D

- ✕ **Une scène MPEG-4 peut contenir des éléments 2D et 3D**

- + Primitives 2D dans une scène 3D

- ✕ Interface 2D dans un monde 3D

- + Composition de plusieurs scènes 2D et 3D

- ✕ Objet 3D dans une scène 2D

- + Mapping de scènes 2D ou 3D sur des surfaces quelconques

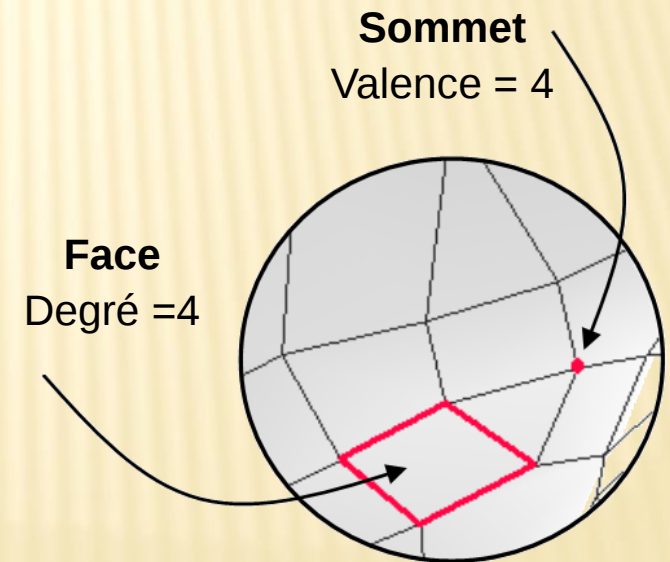
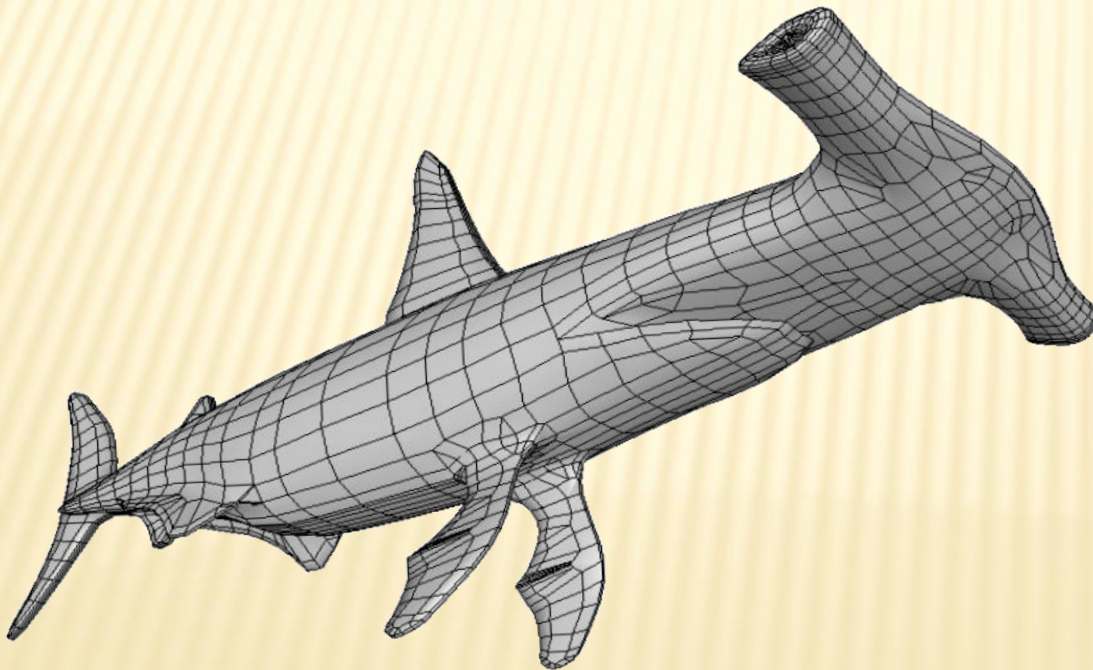
- ✕ Effets spéciaux

MPEG-4 COMPRESSION 3D

- ✗ **MPEG-4 V2 fournit des outils pour la compression 3D**
 - + Codage générique de maillage 3-D
 - + Scalabilité - Level of Detail (LOD)
 - ✗ Le décodeur peut faire un rendu d'une version simplifiée du maillage
 - ✗ Pour objets distants, avec des terminaux peu puissants
 - + Transmission progressive
 - ✗ Pour recevoir un objet avec une qualité qui s'améliore
 - ✗ 2 approches :
 - ★ Transmission de LOD successifs
 - ★ Codage de maillage progressif

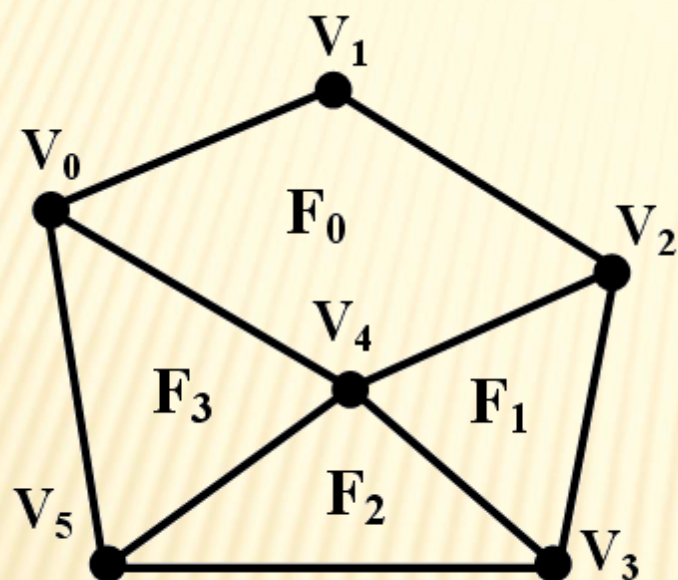
LES MAILLAGES POLYGONAUX

- ✗ **Le maillage polygonal, représentation 3D la plus utilisée :**
 - + Scan 3D
 - + CAO
 - + Rendu réaliste...
- ✗ **Visualisation surfacique → pas besoin des données intérieures**



LES MAILLAGES POLYGONAUX

✖ Il faut coder la **géométrie** et la **connectivité**



Géométrie				Connectivité			
4 octets × 3 coordonnées				4 octets × 3 indices			
V_1	x_1	y_1	z_1	F_1	4	2	3
V_2	x_2	y_2	z_2				
V_3	x_3	y_3	z_3				
V_4	x_4	y_4	z_4				
V_5	x_5	y_5	z_5				

Maillage triangulaire
36 octets / sommet
288 bits / sommet

COMPRESSION DE MAILLAGES 3D

✗ Objectif:

Réduire la taille mémoire pour améliorer / faciliter la transmission, le stockage, la manipulation des objets 3D

✗ 3 Classes de méthodes:

- + Simplification
- + Codage mono-résolution
- + Codage multi-résolution

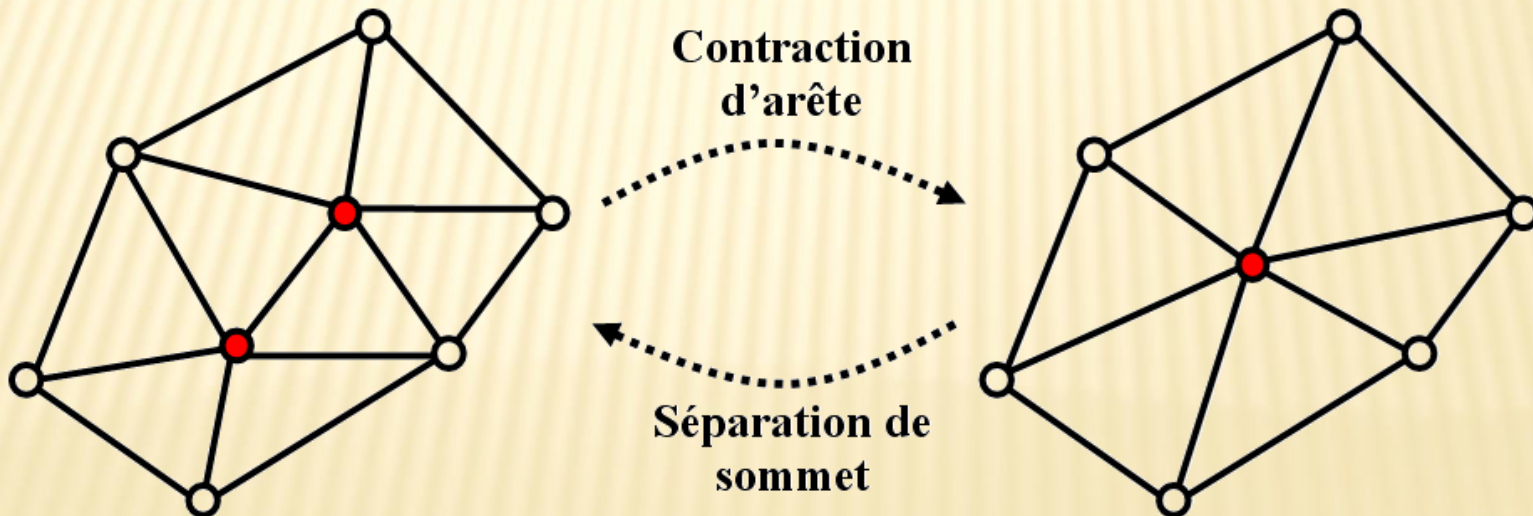
SIMPLIFICATION DE MAILLAGES 3D

✖ Objectif:

Diminuer le nombre de faces et de sommets en gardant un aspect visuel acceptable

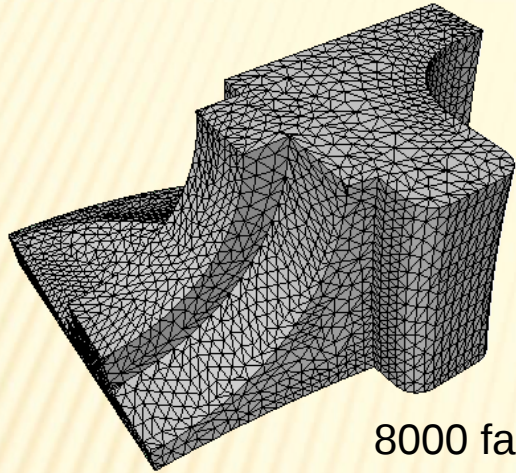
✖ Plusieurs approches

- + Fusion de faces
- + Suppression de sommets (par contraction d'arête)
- + Remaillage (on échantillonne de nouveaux points sur l'objet)

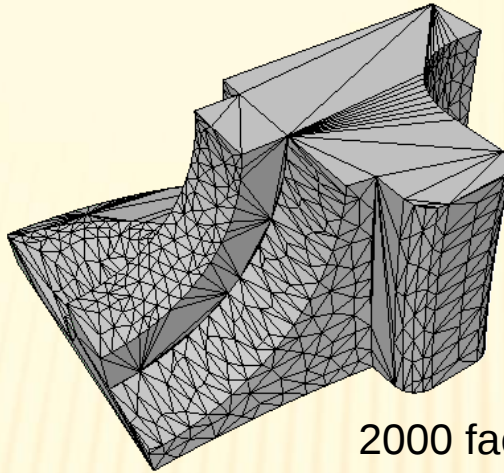


SIMPLIFICATION DE MAILLAGES 3D

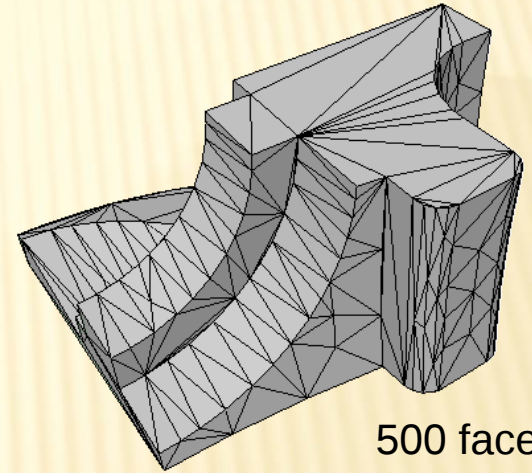
Exemple de simplification du maillage *Fandisk* [Garland, 1997]



8000 faces

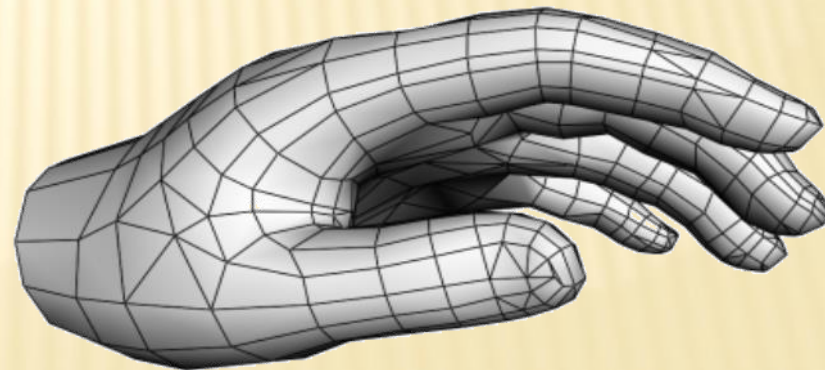
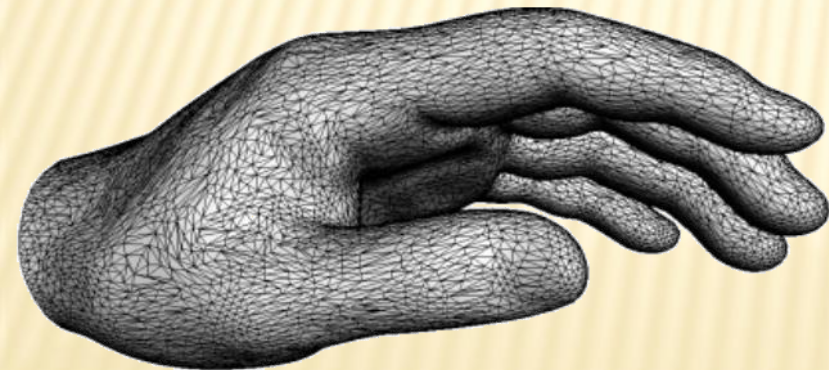


2000 faces



500 faces

Exemple de remaillage du modèle *Hand* [Alliez, 2003]



COMPRESSION MONO-RESOLUTION

✗ Il faut coder la géométrie et la connectivité

+ Codage de la géométrie

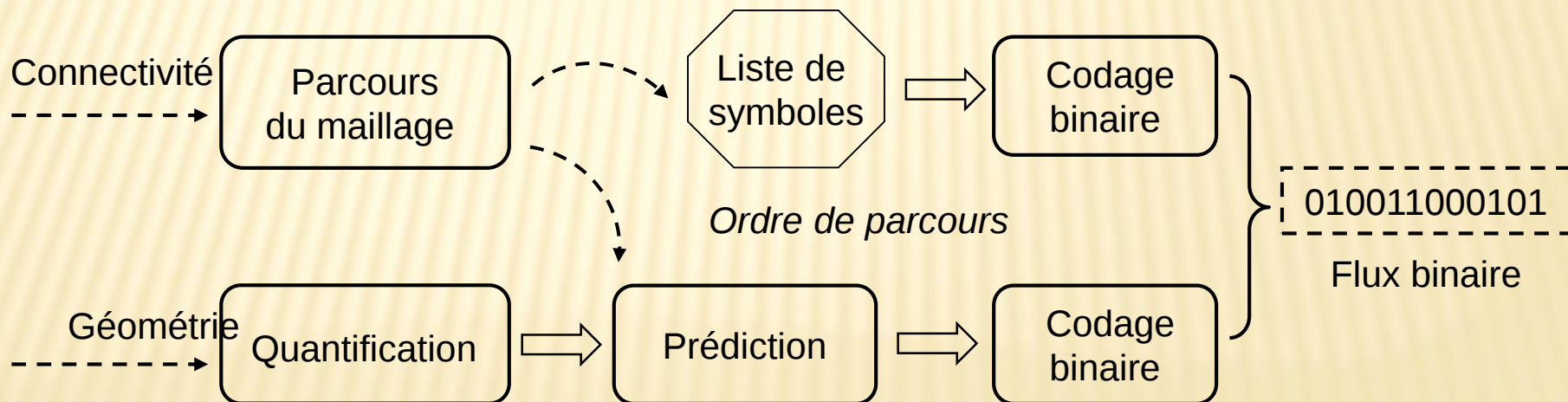
- ✗ Quantification des coordonnées

- ✗ Prédiction

+ Codage de la connectivité

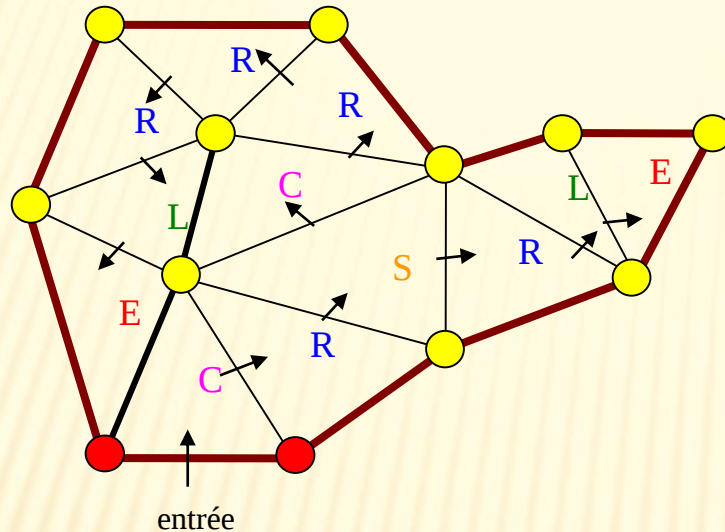
- ✗ Génération d'un parcours sur le graphe du maillage

- ✗ Codage des symboles correspondants



CODAGE DE LA CONNECTIVITÉ

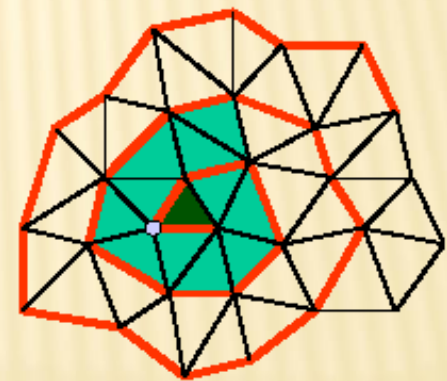
✗ Approche par croissance de région [Rossignac, 1998]



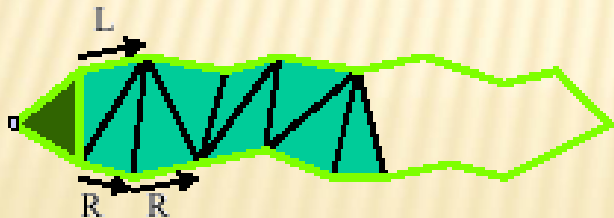
CRSRLECRRRLE

En sortie, on obtient
une liste de symboles
=> Codage entropique

✗ Spirales de triangles [Bajaj, 1999]

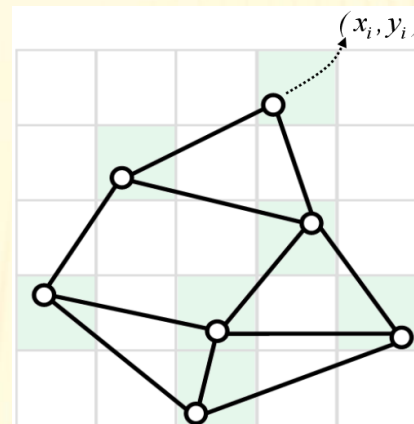
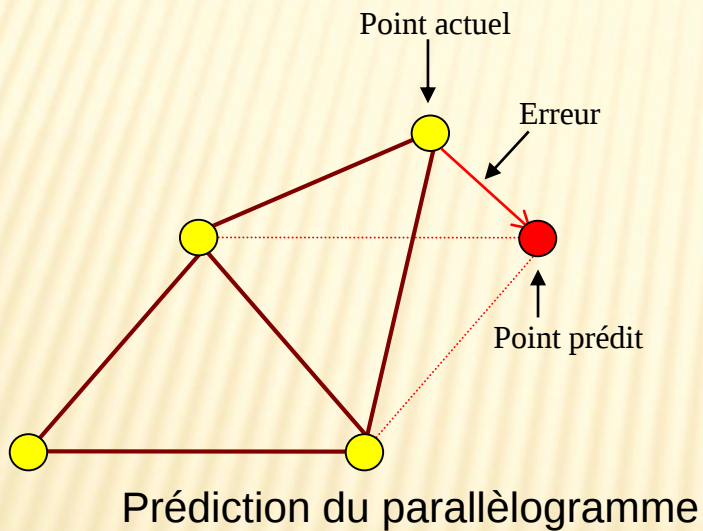


✗ Ensemble de Bandes de triangles [Deering, 1995]

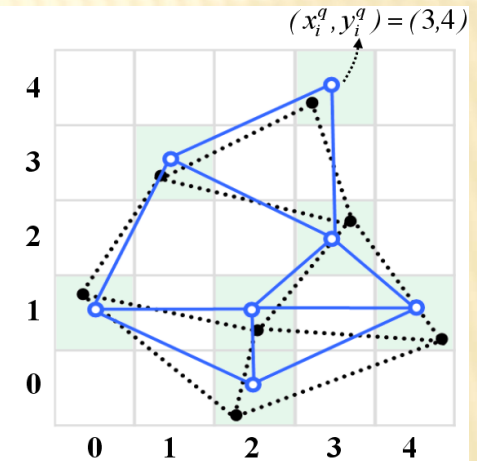


CODAGE DE LA GÉOMÉTRIE

- ✗ Codage de la géométrie
 - ✗ Quantification
 - ✗ Prédiction
 - ✗ Codage entropique



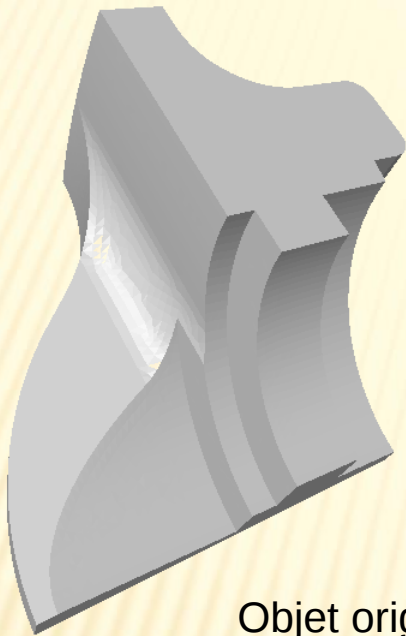
Maillage original



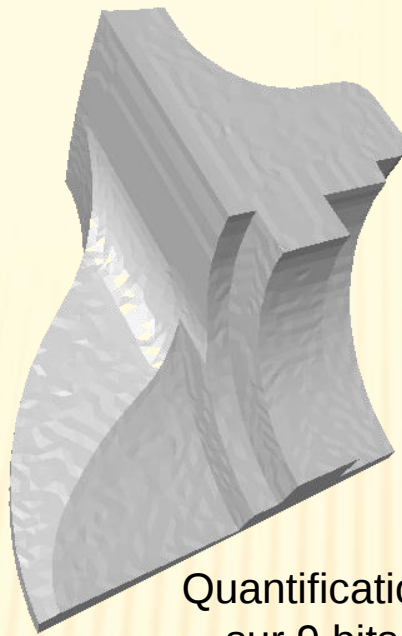
Maillage quantifié

Quantification uniforme sur une grille 5×5

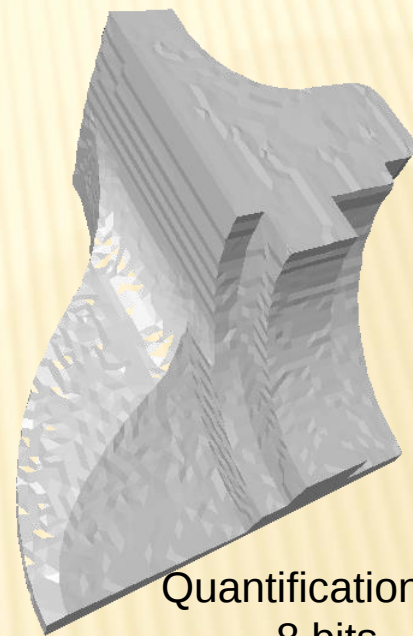
QUELQUES EXEMPLES DE RÉSULTATS



Objet original



Quantification
sur 9 bits



Quantification sur
8 bits

Taille binaire sans codage : 234 Ko

Tailles compressées:

Quantification 12 bits: 19 Ko => 91.7 %

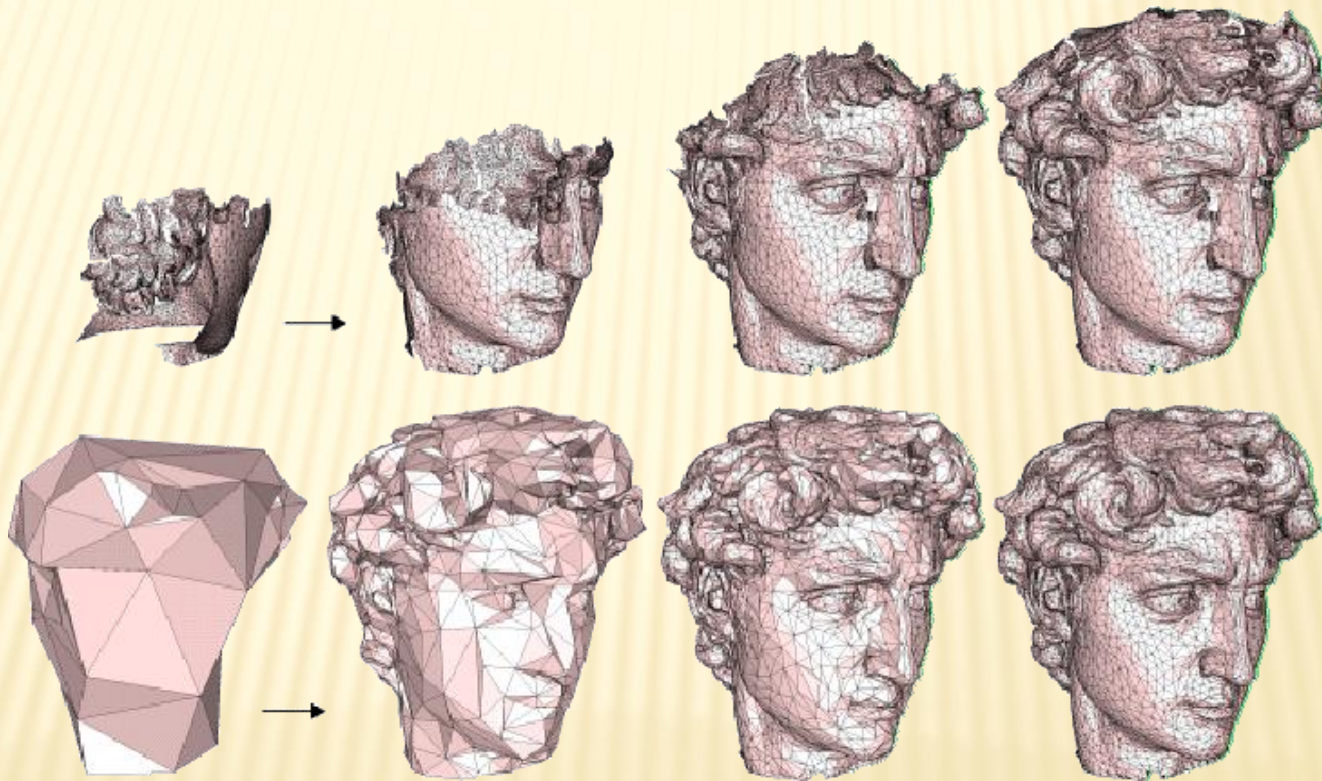
Quantification 10 bits: 14 Ko => 94.2 %

Quantification 8 bits: 9 Ko => 96.3 %

COMPRESSION MULTI-RESOLUTION

✗ Principe:

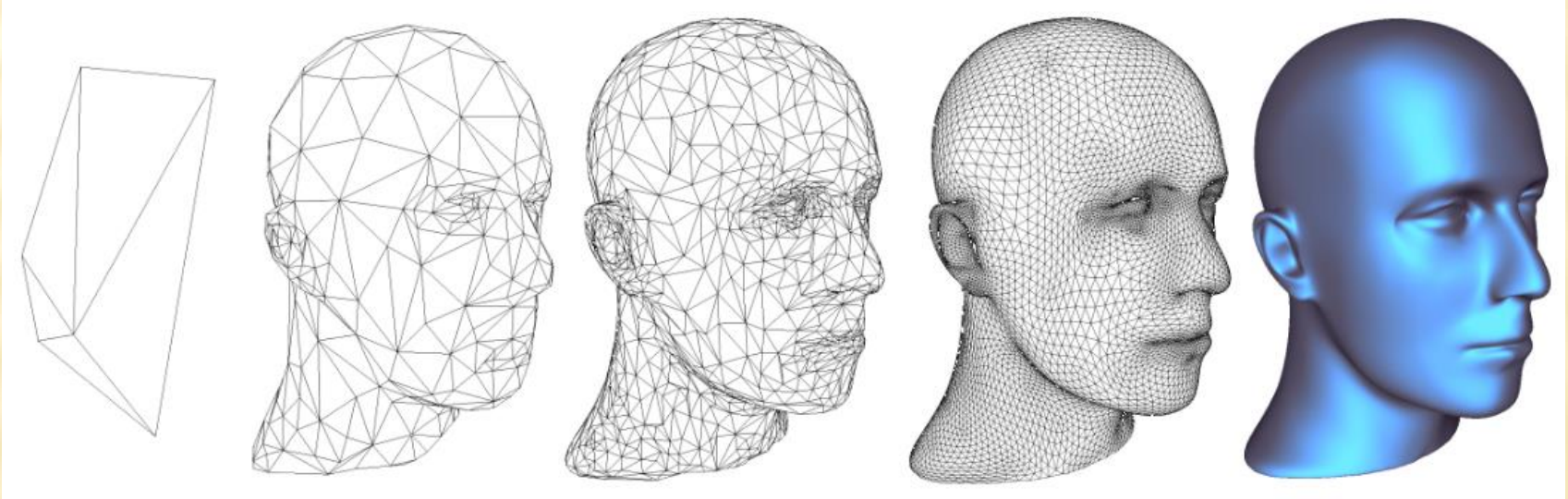
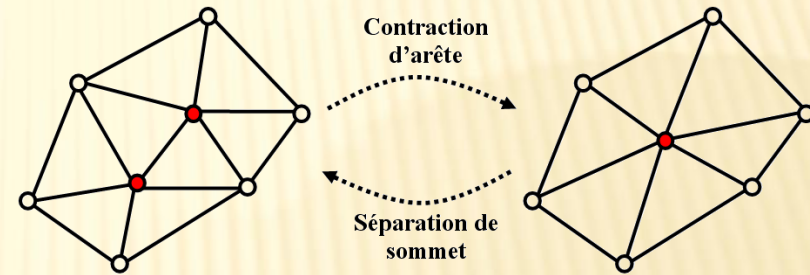
Transmettre d'abord un maillage simple, puis une séquence d'information de raffinement permettant de reconstruire la pleine résolution.



COMPRESSION MULTI-RESOLUTION

✗ Par simplification:

Le maillage est simplifié itérativement et les informations nécessaires à l'inversion du processus sont mémorisées

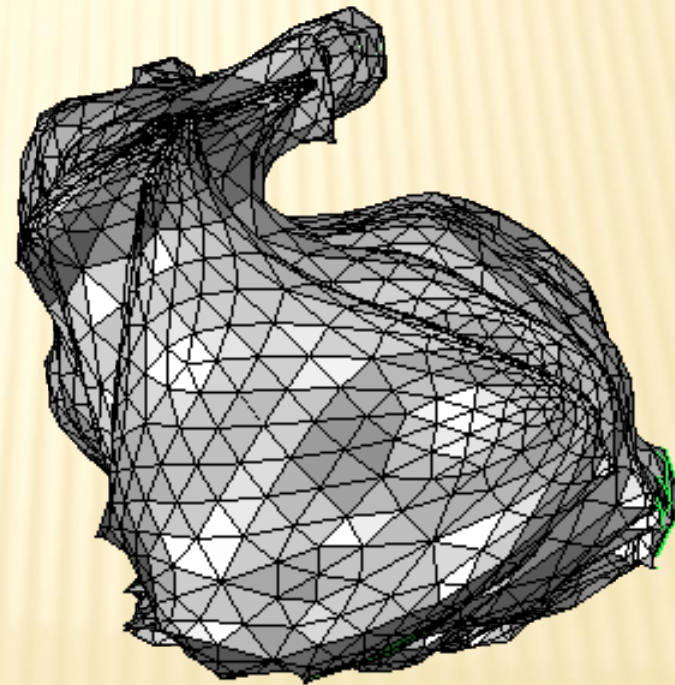
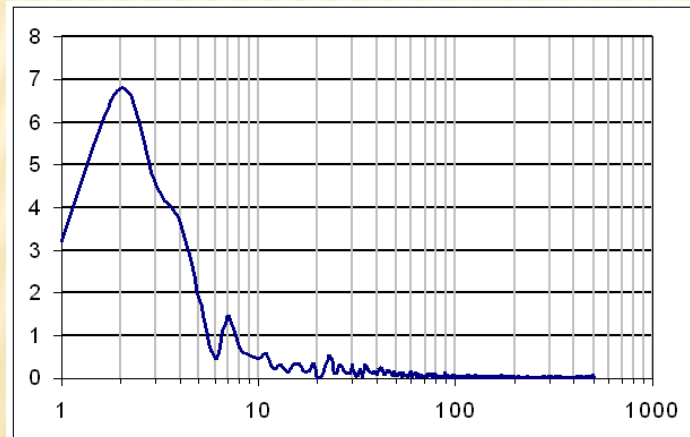
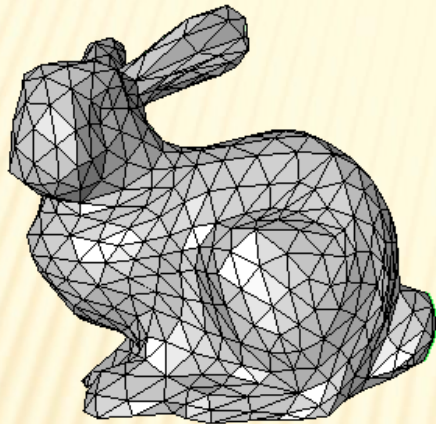


[Alliez, 2001]

COMPRESSION MULTI-RESOLUTION

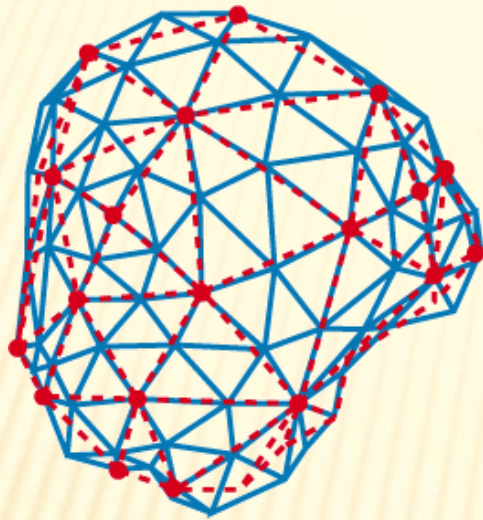
✗ Par décomposition spectrale

On transmet d'abord les coefficients basses fréquences puis les hautes fréquences au fur et à mesure [Karni, Gotsman, 2000]

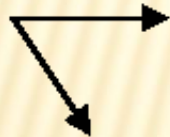
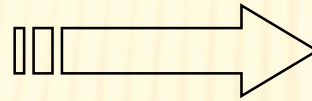


COMPRESSION MULTI-RESOLUTION

- ✗ Par décomposition sur une base d'ondelettes



Décomposition
multi-échelles



D^5



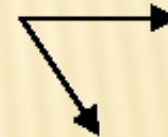
M^5



D^4



M^1



D^0



M^0

[Valette, 2004]