



Cours Réseaux Multimédias



Fatiha El Hatmi & Mayssa Trabelsi

2^{ème} Licence Informatique et multimédia (IM)

2024-2025



Plan du cours

- ☐ Chapitre 1 : Généralités sur le son
- ☒ **Chapitre 2 : L'image**
- ☐ Chapitre 3 : Internet et les applications temps réel
- ☐ Chapitre 4 : QoS
- ☐ Chapitre 5 : Voix sur IP Vop
- ☐ Chapitre 6 : Insuffisance de l'IP
- ☐ Chapitre 7 : Routage multicast
- ☐ Chapitre 8 : Protocoles de transports multimédia
- ☐ Chapitre 9 : Diffusion de la vidéo : la vidéo à la demande VoD



Plan du chapitre II

■ Numérisation de l'image

- ▶ Numérisation
- ▶ Résolution

■ Compression

- ▶ Compression sans perte
- ▶ Algorithmes de compression SP
 - Codage RLE
 - Codage Huffman

■ Annexes

Compression avec perte

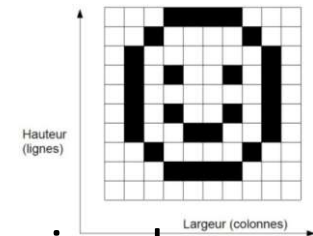


L'image numérique

□ **L'image numérique:** est toute image acquise ou créée et stockée sous forme binaire, composée d'unités élémentaires, appelées pixels (picture element).

D'un point de vue **mathématique:**

- Une image est une matrice de nombres représentant un signal.
- Plusieurs outils permettent de manipuler ce signal.



□ **Pixel (de l'anglais Picture element):** point coloré élémentaire, petit constituant de base pour former les images numériques sur ordinateur. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image.

□ **Numérisation:** la numérisation est la conversion du signal électrique continu (dans l'espace ou temps) en un signal numérique discret (image numérique).



L'image numérique

- Les valeurs de $f(x,y)$ sont la réponse du capteur au phénomène observé
- Les valeurs de $f(x,y)$ sont des valeurs de «voltage» continu (x et $y \in \mathbb{R}$)
- Les valeurs de $f(x,y)$ doivent être converties vers le domaine numérique
- Conversion Analogique/Numérique (A/N)
- Deux procédés sont impliqués pour numériser une image :

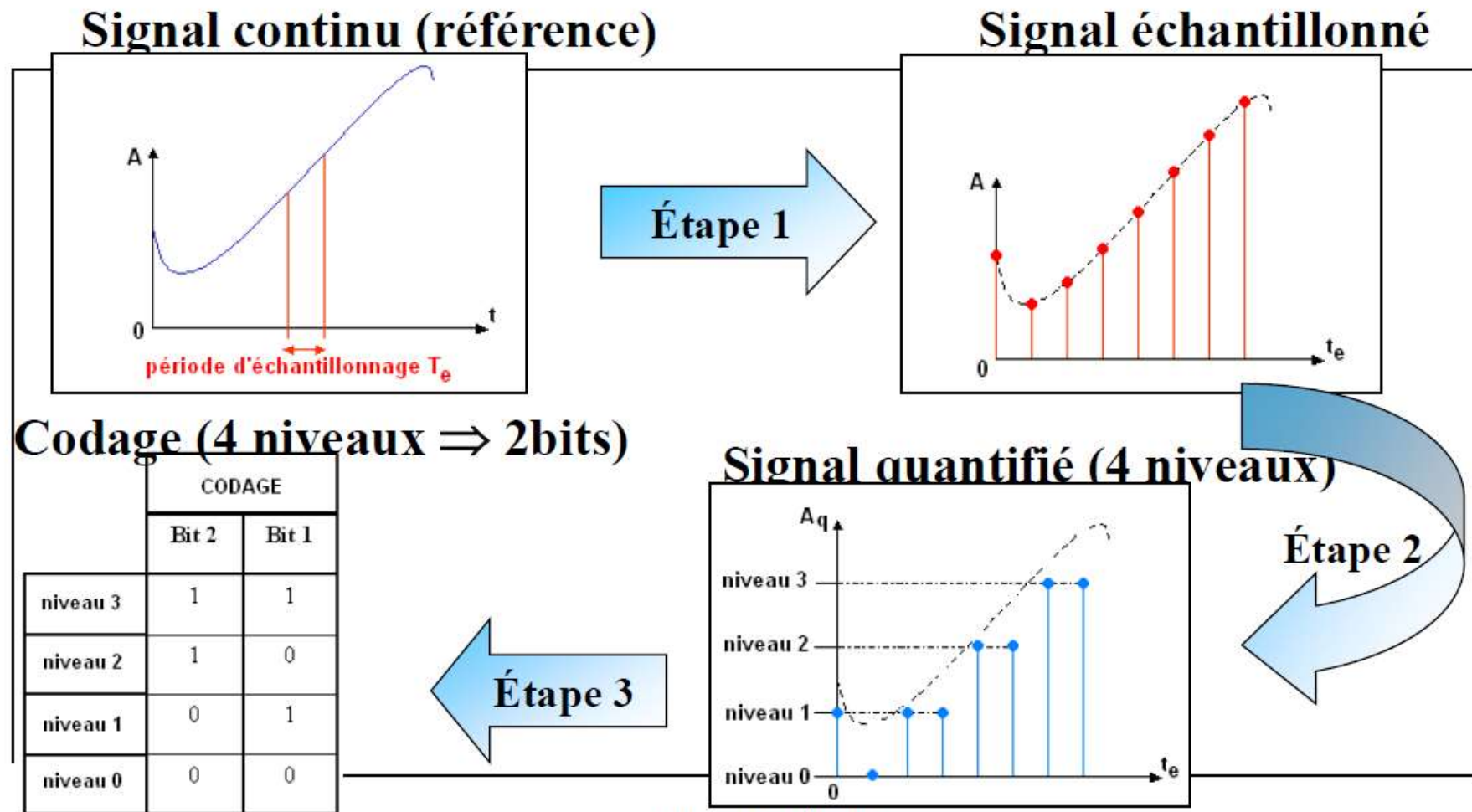
Numérisation = Échantillonnage + Quantification

- L'**échantillonnage** est le procédé de discrétisation spatiale d'une image consistant à associer à chaque zone rectangulaire $R(x,y)$ d'une image continue une unique valeur $I(x,y)$.
- La **quantification** des luminances désigne la limitation du nombre de valeurs différentes que peut prendre $I(x,y)$, c'est la discrétisation des intensités de l'image réelle.



L'image numérique

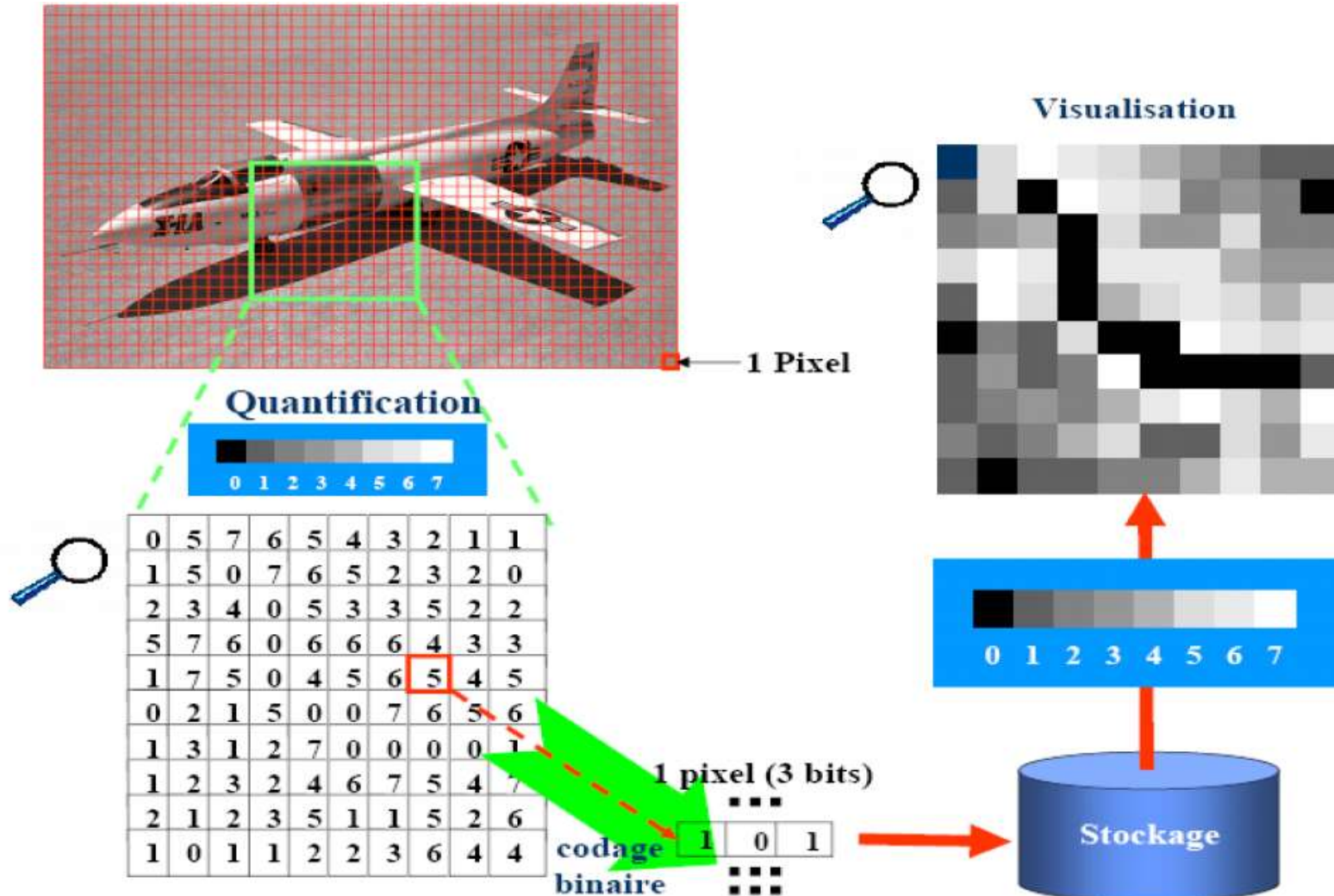
□ Chaîne de numérisation d'un signal temporel (1-D)





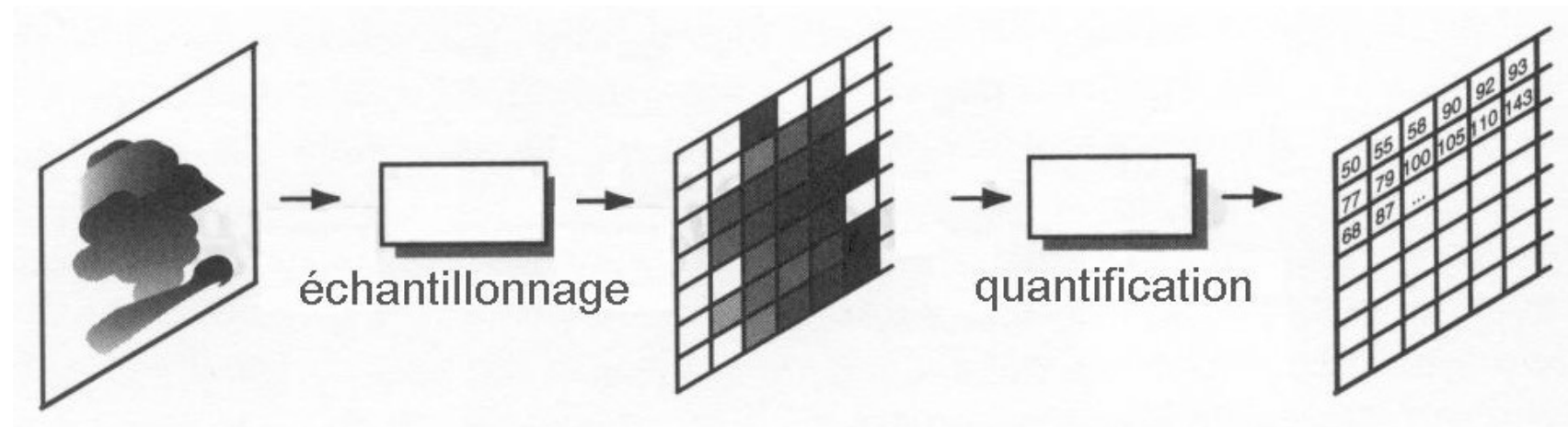
Echantionnage et quantification

Avec un capteur à matrice :





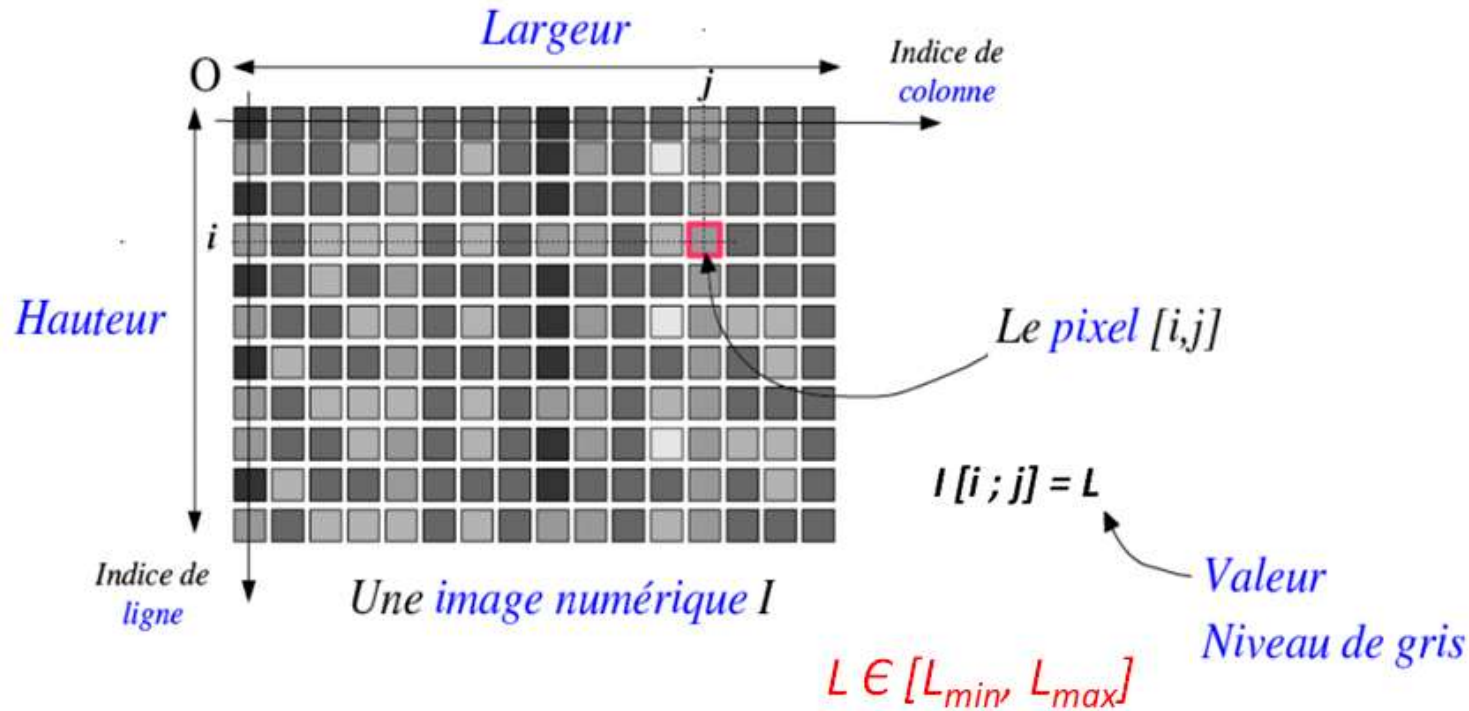
Représentation des images



- Matrice de dimension $M * N$
- Chaque élément a une valeur entière dans l'intervalle $[L_{min}, L_{max}]$
- Le nombre de «bits» requis pour représenter les niveaux de gris dans l'intervalle « L » est « K »
- La relation entre « K » et « L » est : $L = 2^K = (L_{max} - L_{min}) + 1$
- Le nombre de bits pour entreposer une image est donc: $b = M \times N \times K$



Représentation des images



$$L_{max} - L_{min} + 1 = \text{nombre de niveaux de gris}$$

$[i; j]$ les coordonnées spatiales d'un élément de l'image (ligne i , colonne j) après numérisation ;



Echantillonnage et résolution

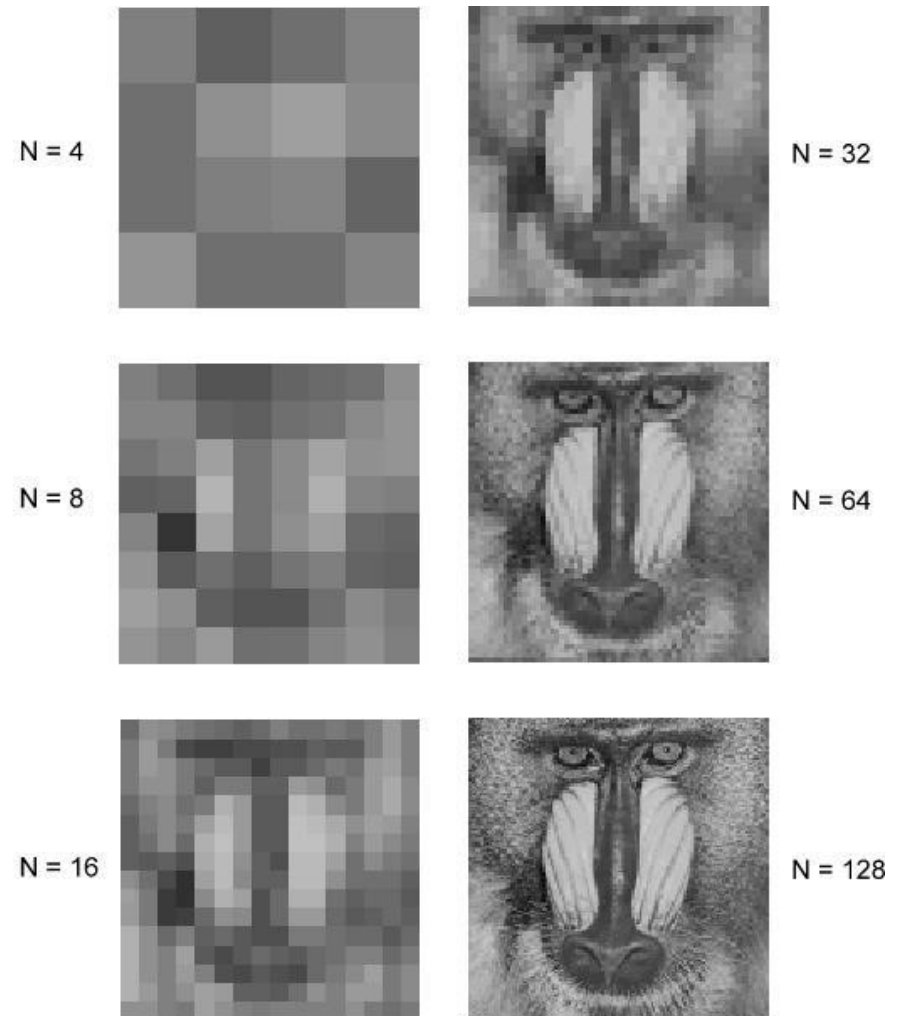
■ Résolution spatiale

Le plus petit détail discernable

■ Résolution de tons de gris

Le plus petit changement discernable

Une image a donc une résolution spatiale de **M x N** pixels et une résolution de tons de gris de **K** bits ou de **L** niveaux ou tons.



Résolution spatiale



Discrétisation de l'amplitude (quantification des niveaux de gris)

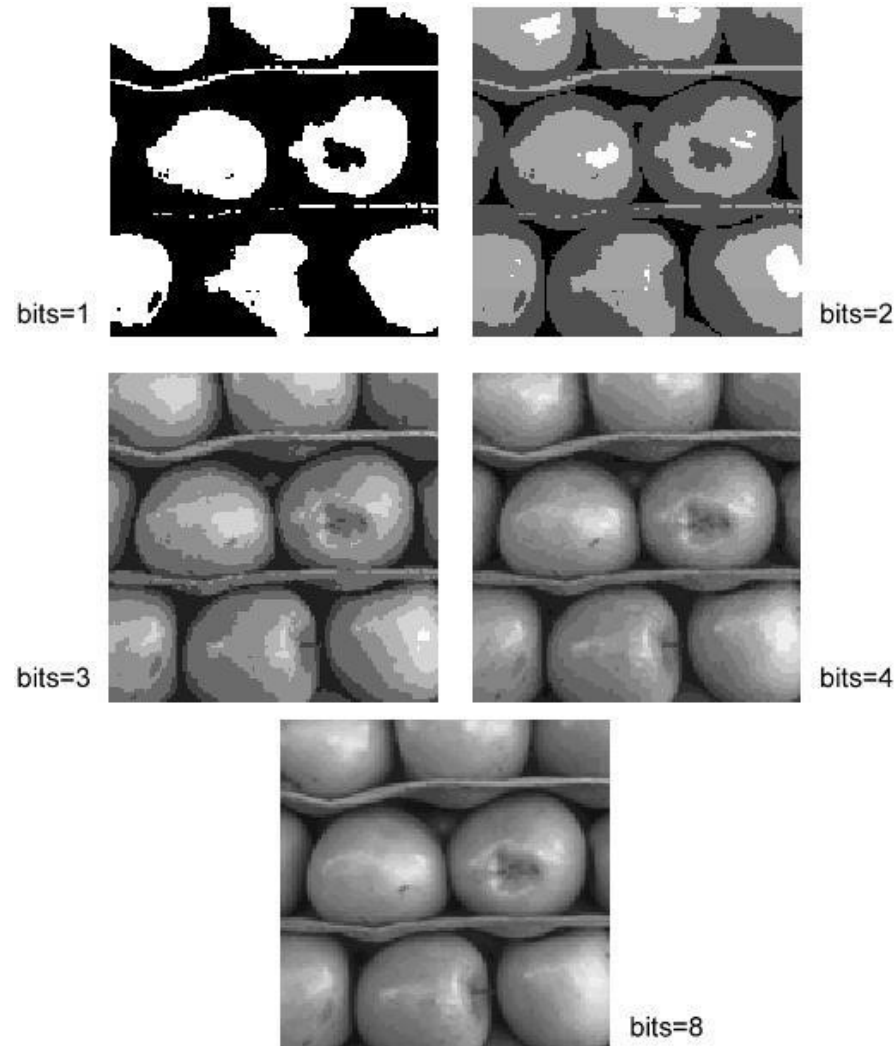
- La valeur de $f(i,j)$ est quantifiée sur k bits et peut ainsi prendre 2^k valeurs entières.

Quantification = représentation sur 2^k niveaux de gris.

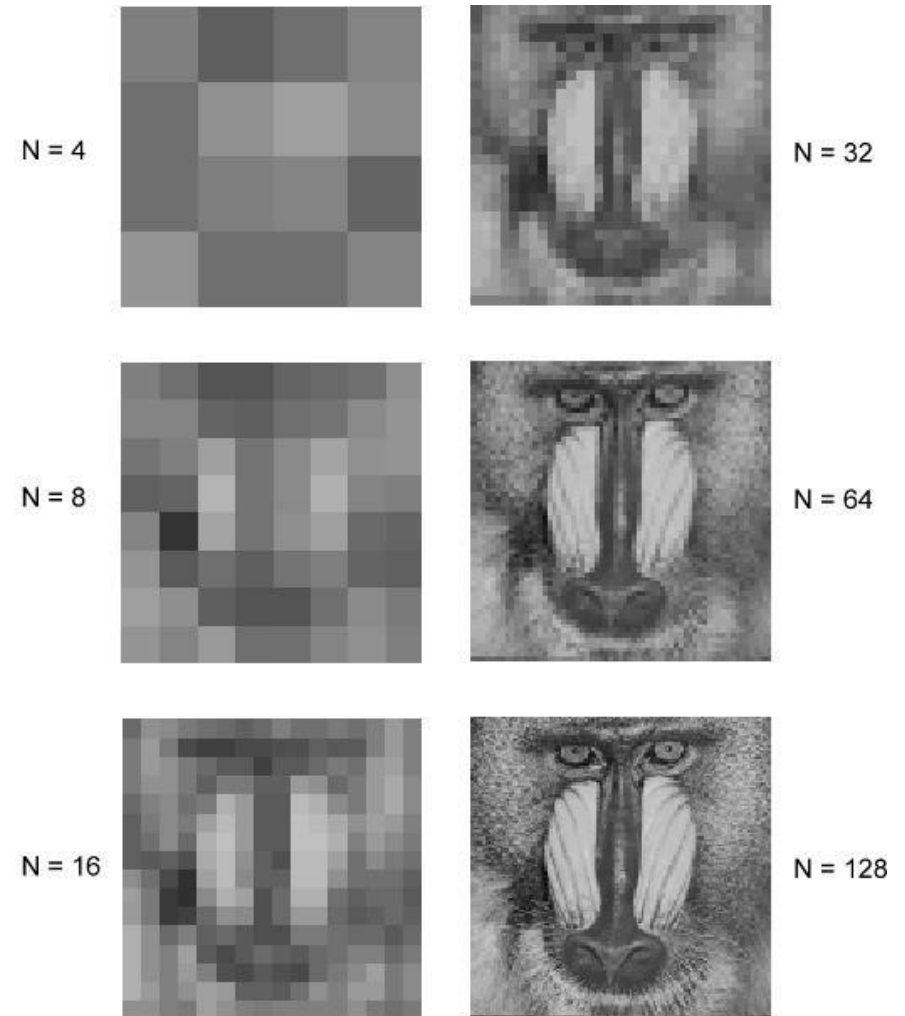
- $K = 1$ bit: $0 \leq I(i,j) \leq 1$ (image binaire)
- $K = 8$ bits: $0 \leq I(i,j) \leq 255$ (image en niveaux de gris)
- $K = 16$ bits: $0 \leq I(i,j) \leq 65\,535$
- Quand chaque pixel est choisi soit noir soit blanc, cela donne une image **monochrome** ;
- Il peut aussi être choisi parmi les intermédiaires entre ces extrêmes, cela donne une image en **niveaux de gris** ;
- Enfin, il peut être choisi parmi les couleurs visibles par notre œil, cela donne une **image en couleurs**.
- Ces valeurs possibles attribuées à un pixel représentent le résultat de la quantification du pixel (en traitement du signal).



Quantification des niveaux de gris



Résolution des niveaux de gris



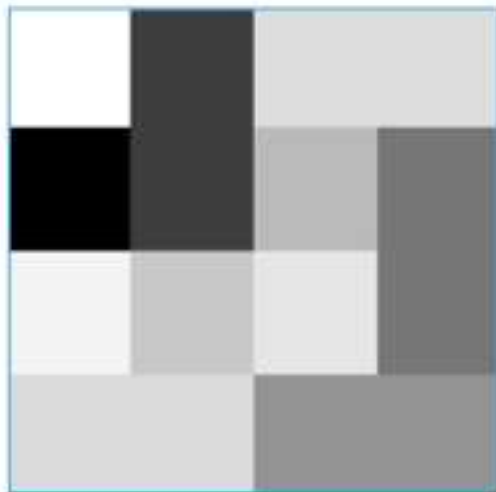
Résolution spatiale



Quantification des niveaux de gris

□ Image en niveaux de gris

- *Chaque pixel est associé à un niveau de gris généralement codé sur 8 bits (de 0 à 255)*

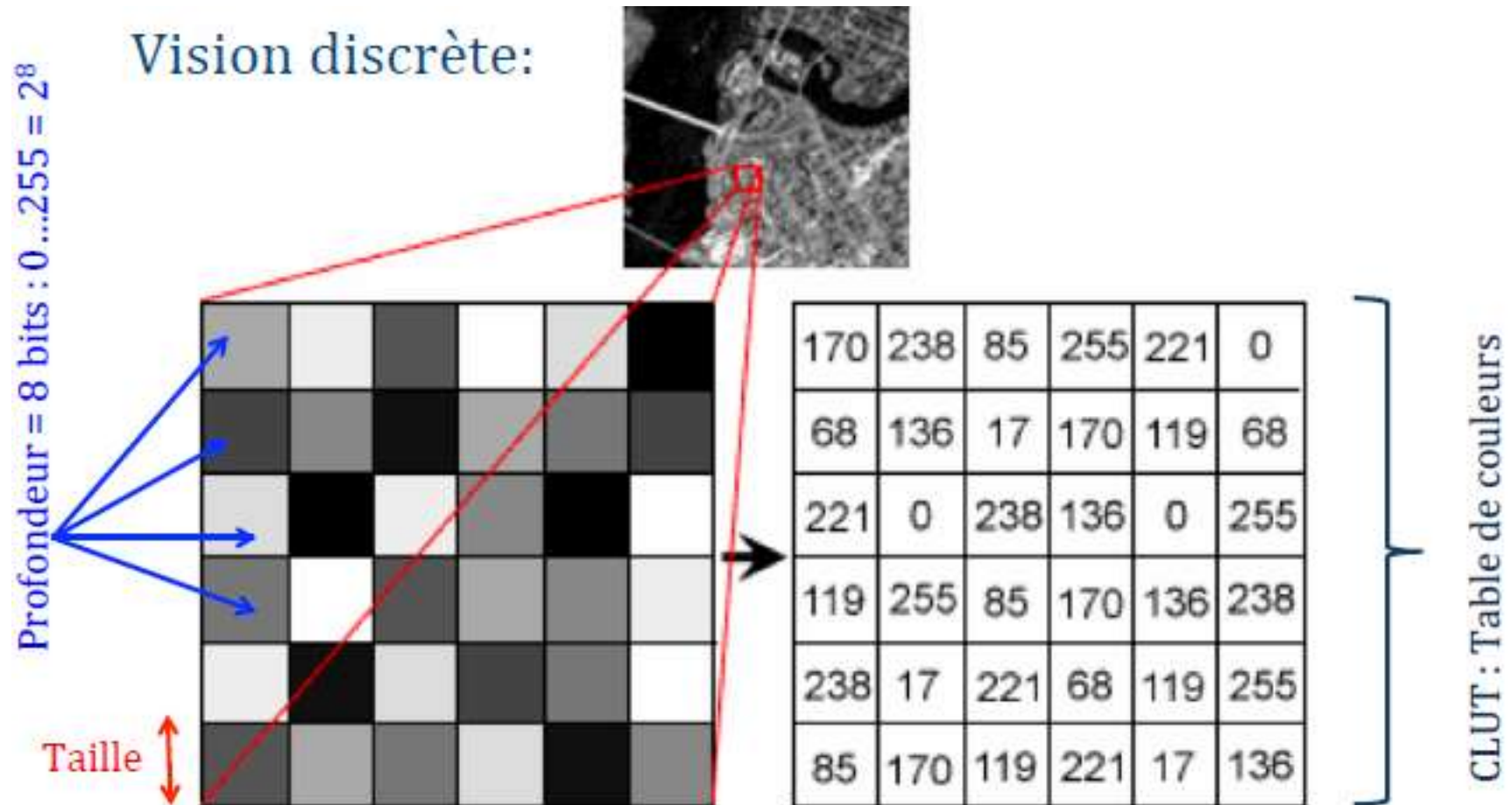


255	83	205	205
0	83	147	88
232	179	209	88
200	200	111	111

Niveau = 0 -> Noir
Niveau = 255 -> Blanc



Quantification des niveaux de gris





Plan du chapitre II

■ Numérisation de l'image

- ▶ Numérisation
- ▶ Résolution

■ Compression

- ▶ Compression sans perte
- ▶ Algorithmes de compression SP
 - Codage RLE
 - Codage Huffman
- ▶ Compression avec perte



Compression

□ Pourquoi compresser :

□ La taille de stockage

$$90_{\text{min}} \times 25_{\text{frame/s}} \times 576_{\text{line/frame}} \times 720_{\text{pixel/line}} \times 24_{\text{bit/pixel}}$$

$$= 168 \text{ GB}$$



36 DVDs



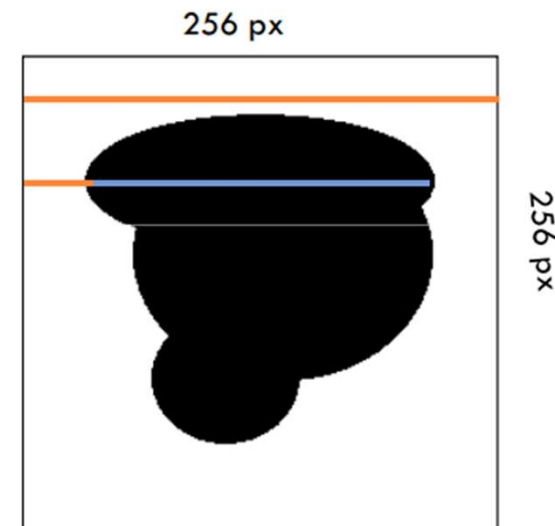
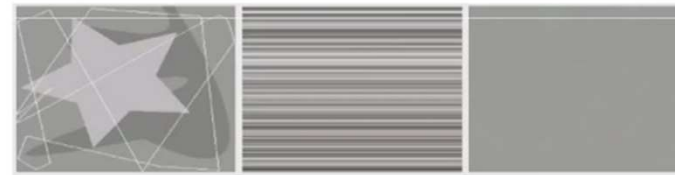
1 DVD



Compression

□ Principes de compression:

- La présence de redondance
- Le nombre de couleurs limité
- Le gains des informations non exploitées
- Run Length Coding
- Efficace pour les images binaires
- On peut coder par deux valeurs uniques si on a un seul objet
 - Exp.: pour une image de 256 x 256 px
 - On a
 - 256;
 - ...
 - 20, 50;
 - ...





Compression

● Objectifs :

- ▶ Réduire le volume de données nécessaire au codage d'un signal numérique
- ▶ Pour faciliter son stockage ou sa transmission par réseau
- ▶ Gagner de la place
- ▶ Diminuer le temps de transmission

● Principe :

- ▶ Détection de redondances dans le signal.
- ▶ Un algorithme de compression permet le codage réduit du signal. Un algorithme (inverse) de décompression permet d'exploiter le signal .

● Types de compressions

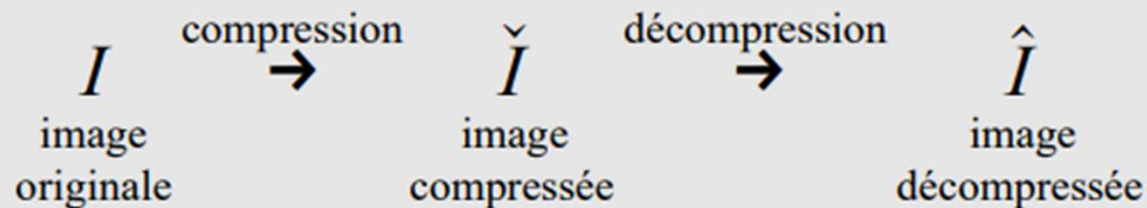
- ▶ Compression sans perte (ou non-destructive, i.e. codage ou compactage) : Le signal obtenu après décompression est strictement identique à l'original : RLE, Huffman, LZW, Arithmétique...
 - Utilisation : Fichier exécutable, fichier texte.
- ▶ Compression avec perte (ou destructive, ou avec dégradation) : Le signal obtenu après décompression diffère (légèrement) de l'original
 - Utilisation : image, son, vidéo. : on sacrifie des informations pour compresser plus efficacement : DCT, ondelettes, prédictif.



Compression

□ Performances :

• Notations



• Mesures de performance

→ Taux de compression

$$\tau := \frac{\text{volume}(\check{I})}{\text{volume}(I)}, \text{ souvent noté en ratio. Ex.: } \tau = \frac{2 \text{ Mo}}{10 \text{ Mo}} = 0,2 \text{ noté } 1:5 \text{ ("1 pour 5")}$$

Confusion commune avec le **quotient** (ou *ratio*) de compression $q := \frac{\text{volume}(I)}{\text{volume}(\check{I})} = \frac{1}{\tau}$

→ Mesure objective de distorsion (cas avec perte)

- Erreur quadratique moyenne
(ang. « Mean Square Error »)

$$MSE = \frac{1}{3MN} \sum_{k=R,G,B} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I_{m,n}^k - \hat{I}_{m,n}^k)^2$$

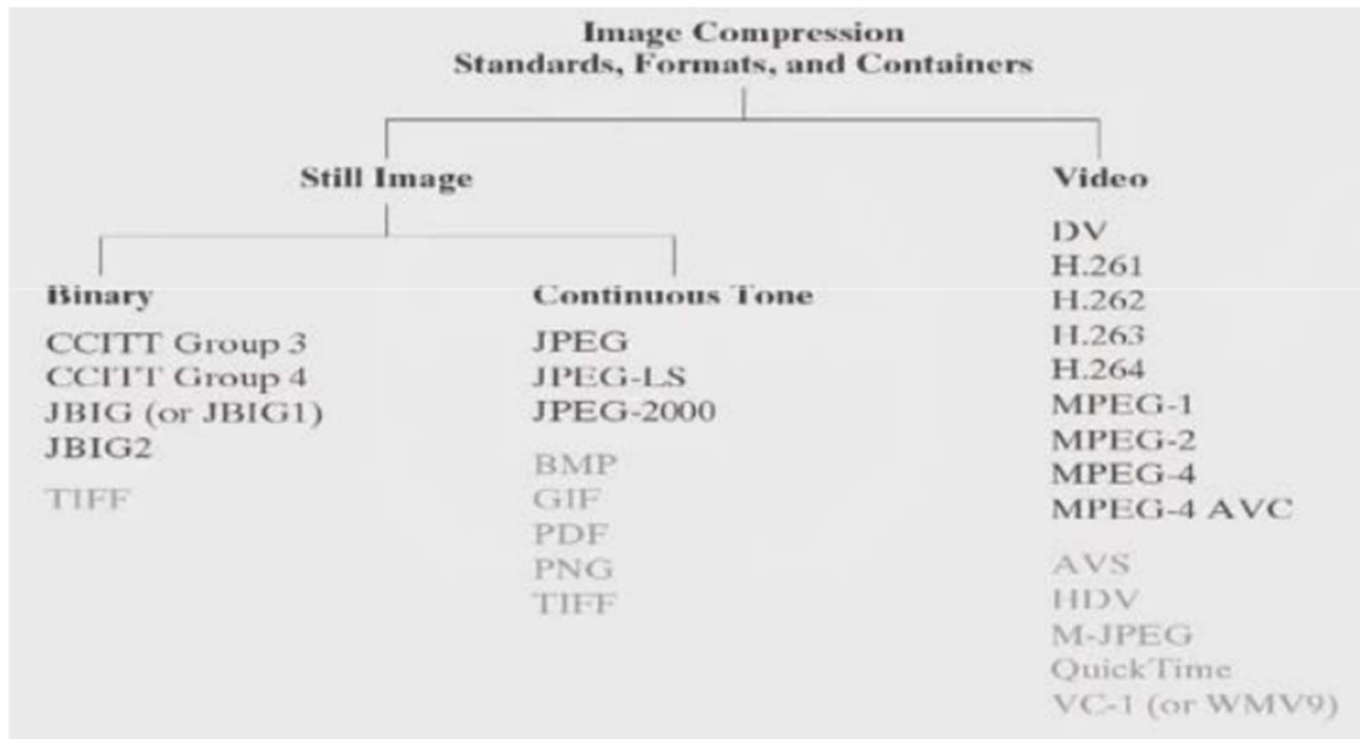
- Rapport signal sur bruit pic-à-pic
(ang. « Peak Signal-Noise Ratio »)
 d = valeur max. possible (ex. 255)

$$PSNR = 10 \cdot \log_{10} \left(\frac{d^2}{MSE} \right)$$



Formats d'images

□ Création des standards de compression :



□ Images: JPEG-LS fidèle à la sources

□ Vidéos: Les MPEG 1~4 sont les plus fameuses



Formats d'images

□ Format JPEG :

- Les images JPEG ont l'extension ".jpg", ".jpeg", ".jpe" ou ".jfif", les quatre notations sont justes.
- Le mot JPEG veut dire Joint Photographic Expert Group représentant le nom d'un groupe d'experts qui s'est chargé de créer un format de compression pour les images photographiques.
- Le format JPEG est adapté aux images comportant plus de 256 couleurs, c'est-à-dire en fait les photographies en couleurs, surtout si elles sont riches en nuances. Mais la compression JPEG induit des pertes de données car elle procède par approximations.
- À la compression elle calcule, puis recalcule à la décompression, les couleurs s'approchant des couleurs d'origine, en fonction d'un taux de compression choisi par l'utilisateur. Plus le taux est élevé, plus léger sera le fichier mais plus l'image sera dégradée



Formats d'images

❑ Format JPEG :

- Voici une image non compressée. Elle fait 300 x 225 pixels, pèse environ **200 ko** et mettra dans les **60 secondes** à être téléchargée en très bas débit (28 kb/sec).
- La même, comportant le même nombre de pixels mais très fortement compressée en JPEG : elle ne pèse plus que **6 ko** et prendra théoriquement moins de **2 secondes** de téléchargement à débit égal...



Image originale



Image très fortement compressée en JPEG



Formats d'images

❑ Format JPEG :

► L'image a subi des altérations, comme on peut le voir sur l'image elle-même et mieux encore sur ce détail agrandi.



Image très fortement compressée
en JPEG : Zoom



Types de compression

□ Types de compressions avec ou sans perte :

	Compressions sans perte (codages)	Compressions avec perte
Exemples	<ul style="list-style-type: none">• delta• codes à longueur variable : VLC préfixé, Shannon-Fano, Huffman• codage arithmétique• à base de dictionnaire : Lempel-Ziv (LZ77, LZW)• par décorrélation : Run-Length Encoding (RLE), codage prédictif sans perte	<ul style="list-style-type: none">• par moyennage de blocs• par transformation linéaire optimale ou de Karhunen-Loeve (KLT)• par transformée en cosinus discrète (DCT) : JPEG• par transformée en ondelettes discrète (DWT) : JPEG 2000• par quantification (scalaire ou vectorielle)• par détection de motifs images redondants (fractale)
Remarques	<ul style="list-style-type: none">• Taux de compression limité• Aucune perte d'information	<ul style="list-style-type: none">• Bon taux de compression• Perte d'information



Types de compression

□ Types de compressions avec ou sans perte :

- Autres critères de qualité possibles :
 - Vitesse de compression/décompression
 - Accessibilité de l'algo : sous licence ou libre



Types de compression

□ Compression des principaux formats d'images bitmaps :

Format	Espaces couleur	Compression(s)	C. α	Domaines d'utilisation, rem.
TIFF (.tif)	<i>RGB</i> , CIE $L^*a^*b^*$, <i>CMYB</i> , couleurs indexées, ndg	Aucune Sans perte (LZW, Huffman) Avec perte (JPEG)	Oui	PAO, Infographie, bureautique Très flexible, mais nombreuses variantes pas toujours supportées
BMP (.bmp)	<i>RGB</i> , couleurs indexées, ndg	Aucune Sans perte (RLE)	Non	Bureautique sous Windows Compression peu efficace
GIF (.gif)	couleurs indexées (2 à 256)	Sans perte (LZW)	Oui	Pages web Animations possibles
JFIF (.jpg)	<i>RGB</i> , <i>CMYB</i> , ndg	Avec perte (JPEG)	Non	Pages web, photographie Compr. efficace mais destructive
PNG (.png)	<i>RGB</i> , ndg, 256 couleurs indexées	Sans perte (deflate=LZ77+Huffman)	Oui	Pages web, photo. sans perte Format libre. Jusqu'à 48 bits.

- TIFF = Tag(ged) Image File Format ; GIF=Graphics Interchange Format ; JFIF = JPEG File Interchange Format ; PNG = Portable Network Graphics.
- *CMYB* (fr. *CMJN*) = Cyan, Magenta, Jaune, Noir ; ndg=niveaux de gris.
- Canal α : permet la transparence.



Types de compression

□ Compression des principaux formats d'images bitmaps :

- Algorithmes sans perte (*lossless*)
 - Reconstruction parfaite
 - Basés sur la redondance statistique
 - Faible rapport de compression
- Algorithmes avec perte (*lossy*)
 - Image reconstruite \neq image originale
 - Basés sur la quantification
 - Redondance psychovisuelle : “visually lossless”
 - Rapport de compression élevé



Compression sans pertes

❑ Codage RLE (Running Length Encoding) :

• Principe

- Codage par plage (*ang.* « Running Length Encoding »)
- Recherche de séquences de données redondantes (*ex.* niveaux identiques).
- Codage de la valeur et du nombre de répétitions :

0	0	0	11	11	67	121	121	98	98	98	32	37	37	37	37	37	37	2	0
↓																			
0	3	11	2	67	1	121	2	98	3	32	1	37	6	2	1	0	1		

• Avantage

- Algorithmes de compression et décompression très simples et rapides.

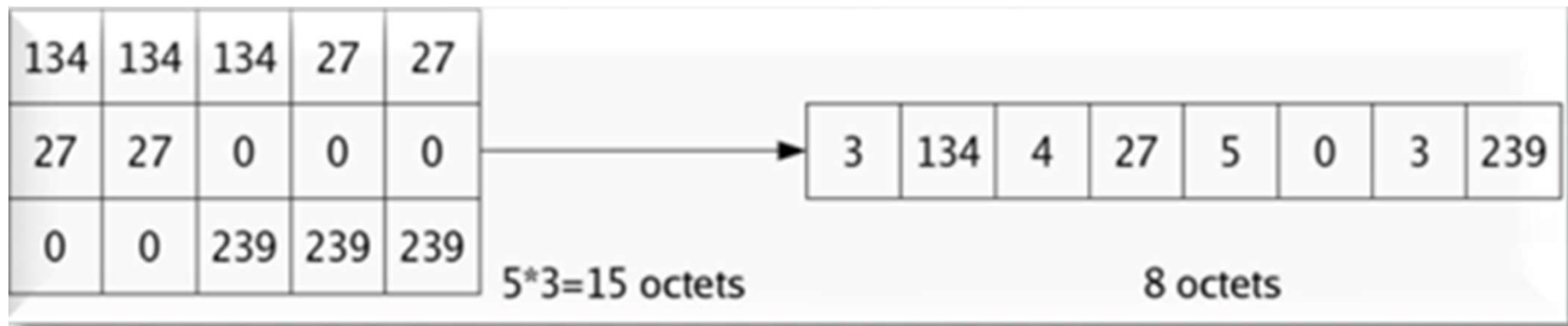
• Limites

- Efficace seulement pour de nombreuses et longues plages constantes.
 - Cas des images de synthèse simples ; peu adapté aux photos.
 - Utilisé *ponctuellement* dans de nombreux formats (BMP, JPG, TIFF, PCX, ...).
- Nécessite de fixer un maximum pour la longueur des plages (*ex.* 255).



Compression sans pertes

❑ Codage RLE (Exemple) :



► Problème :

Inefficace pour des images très texturées!!



Compression sans pertes

□ Codage de Huffman (1/3) :

• Principe

- **Coder les valeurs avec un nombre de bits différent.**
 - Code (utilisant des mots) à **longueur variable** (*ang.* « Variable Length Coding »),
 - dit aussi **codage entropique** (*ang.* « Entropy coding »).
- **Plus une valeur apparaît fréquemment, plus le nombre de bits utilisés pour la coder est petit (*i.e.* plus son code est court).**

• Algorithme de Huffman : codage

- **Phase 1 : Construction de l'arbre.**
 1. Trier les différentes valeurs par ordre décroissant de fréquence d'apparition
⇒ table de **poids**.
 2. Fusionner les deux poids minimaux dans un arbre binaire et affecter leur somme à la racine.
 3. Réordonner la table de poids par poids décroissants.
 4. Recommencer en 2. jusqu'à obtenir un seul arbre.
- **Phase 2 : Construction du code** à partir de l'arbre obtenu dans la phase 1.
 - À partir de la racine, attribuer des 0 aux sous-arbres de gauche et des 1 à droite.



Compression sans pertes

□ Codage de Huffman (2/3) :

Exemple de codage

→ Construction de l'arbre

1. Table des poids

10	15	15	15	15
10	90	100	100	15
10	90	180	100	15
10	90	180	90	15
10	10	10	10	10



10 ₉	15 ₇	90 ₄	100 ₃	180 ₂
-----------------	-----------------	-----------------	------------------	------------------

2. Fusion des poids minimaux

10 ₉	15 ₇	90 ₄	<div> <div>•</div> <div>/ \</div> <div>5</div> <div>100₃ 180₂</div> </div>
-----------------	-----------------	-----------------	--

3. Réordonnancement

10 ₉	15 ₇	<div> <div>•</div> <div>/ \</div> <div>5</div> <div>100₃ 180₂</div> </div>	90 ₄
-----------------	-----------------	--	-----------------

4. Itérations

10 ₉	<div> <div>•</div> <div>/ \</div> <div>9</div> <div> <div>•</div> <div>/ \</div> <div>5</div> <div>100₃ 180₂</div> </div> <div>90₄</div> </div>	15 ₇
-----------------	--	-----------------

<div> <div>•</div> <div>/ \</div> <div>16</div> <div> <div>•</div> <div>/ \</div> <div>9</div> <div> <div>•</div> <div>/ \</div> <div>5</div> <div>100₃ 180₂</div> </div> <div>90₄</div> </div> </div> <td>10₉</td>	10 ₉
---	-----------------

<div> <div>•</div> <div>/ \</div> <div>25</div> <div> <div>•</div> <div>/ \</div> <div>16</div> <div> <div>•</div> <div>/ \</div> <div>9</div> <div> <div>•</div> <div>/ \</div> <div>5</div> <div>100₃ 180₂</div> </div> <div>90₄</div> </div> <div>15₇</div> </div> </div> <td>10₉</td>	10 ₉
--	-----------------



Compression sans pertes

□ Codage de Huffman (3/3)

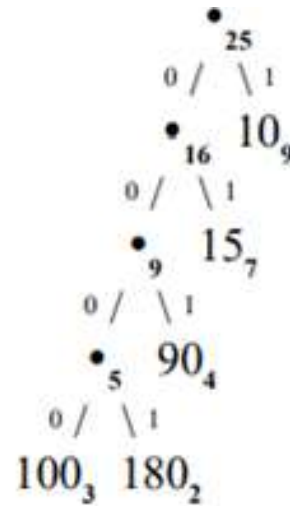
• Exemple de codage (suite)

→ Construction du code

Affectation de valeurs
binaires aux arcs

→ Image codée (en lignes)

10101010110010000 ...
soit 55 bits vs. $25 \times 8 = 200$ bits



⇒ code

Valeur	Code
10	1
15	01
90	001
100	0000
180	0001

10	15	15	15	15
10	90	100	100	15
10	90	180	100	15
10	90	180	90	15
10	10	10	10	10



Comparaison méthodes de compression

- Sans compression (BMP) : Format très lourd
- Compression sans perte :
 - RLE (PCX) : Poids plus faible : élimination des redondances par répétition
 - LZW (GIF / TIFF) : Utilisation d'un dictionnaire : plus efficace que RLE
 - Huffman
- Compression avec pertes :
 - Compression avec DCT (JPEG) : Réduction du coût mémoire et taux de compression réglable
 - Compression avec ondelettes (JPEG 2000) :
 - Meilleur rapport taux de compression / qualité d'image
 - Gestion multirésolution de l'image
 - Compression avec ou sans perte
 - Compression prédictive (WebP)
 - Meilleur taux de compression que JPEG, JPEG2000 et PNG



Annexes : Compression avec pertes

❑ Transformée en cosinus discrète TCD

× Transformée en cosinus discret – DCT

+ C'est une représentation fréquentielle

× **Composante continue**
au coin en haut à gauche

× **Basses fréquences**
en haut à gauche

× **Hautes fréquences**
en bas à droite



1.0e+003 *

1.4195	-0.0078	-0.0140	-0.0113	-0.0507	-0.0280	-0.0359	0.0002
0.0404	-0.0401	-0.0302	0.0259	-0.0183	0.0003	-0.0001	-0.0001
0.0402	-0.0183	-0.0109	-0.0342	0.0004	-0.0000	-0.0000	0.0004
-0.0301	0.0125	0.0005	0.0199	-0.0002	0.0001	-0.0001	-0.0001
-0.0647	0.0300	0.0001	-0.0004	0.0005	-0.0001	0.0002	-0.0000
0.0848	-0.0250	-0.0390	-0.0001	0.0000	0.0000	0.0002	-0.0004
0.0004	0.0005	-0.0000	0.0002	0.0004	0.0003	0.0004	-0.0001
-0.0501	0.0001	-0.0002	-0.0004	0.0001	0.0001	0.0004	0.0006

Fréquence : importance et rapidité d'un changement de couleur.

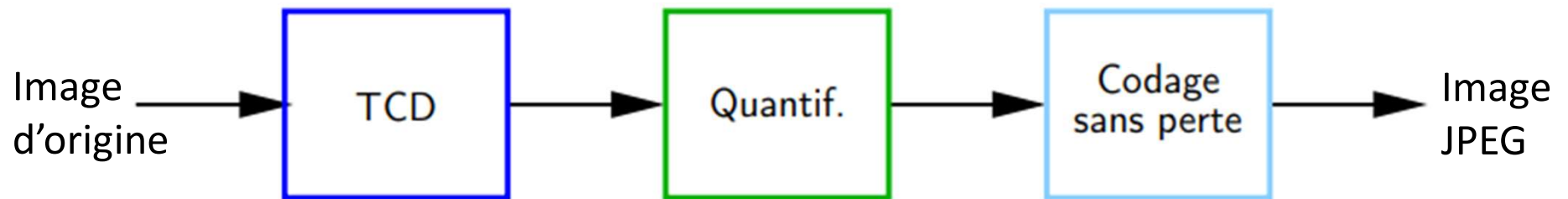
Réduire la quantité d'information de haute fréquence car l'œil humain est plus sensible aux basses fréquences.

Principale opération destructive dans JPEG.



Annexes : Compression avec pertes

□ De la TCD à la JPEG



- L'image est préalablement découpée en blocs 8×8
- On soustrait 128 aux valeurs de luminance
- Les blocs sont codés indépendamment



Annexes : Compression avec pertes

□ Compression JPEG

■ Blocs JPEG :

- Codage 8x8 pixels
- Directe si en Niveaux de gris
- Les canaux Rouge, Vert, Bleu sont corrélés fortement
- → Passage Y (luminance) Cb Cr (couleurs) par le moyen d'une matrice 3x3 inversible
- Sur chaque sous image, on applique la Transformée de Cosinus Discrète (TCD)

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

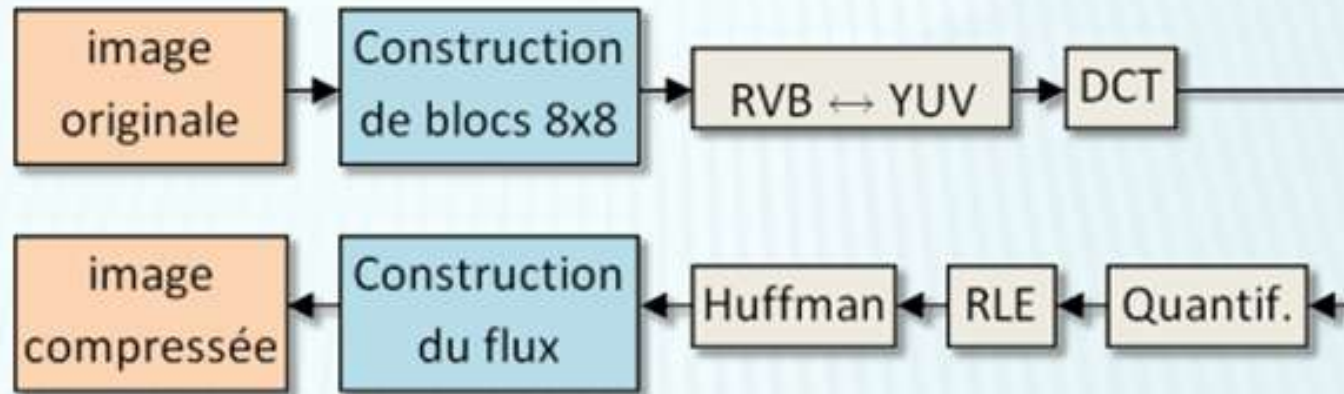
- *L'œil humain est moins sensible aux détails de l'information de couleur (chrominance) que ceux de l'intensité (luminance).*



Annexes : Compression avec pertes

□ Compression JPEG

× Chaîne de traitement du format JPEG



× Remarques

- + Chaque image est décomposée en blocs 8x8
- + Sur les bords, on peut compléter les derniers blocs
 - × Par des zéros
 - × Par des prolongements continus
 - × Par une symétrie
- + Chaque bloc est une 'petite' image



Annexes : Compression avec pertes

□ Compression JPEG

