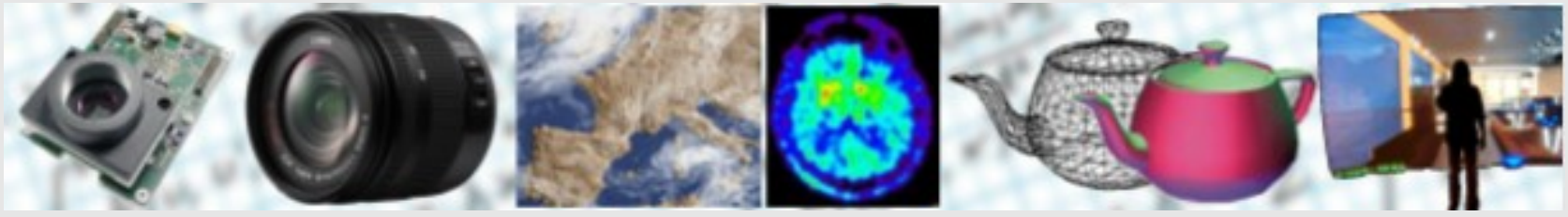


TI – Traitement d'Images
Semaine 12 : Compression d'images

Olivier Losson

Master ASE : http://master-ase.univ-lille1.fr
Master Informatique : http://www.fil.univ-lille1.fr
Spécialité IVI : http://master-ivi.univ-lille1.fr

Plan du cours

- 1 – Introduction
  - Généralités sur la compression de données
  - Mesurer la compression
  - Types de compression et formats d'images
- 2 – Compression sans perte (codage)
  - Codage RLE
  - Codage de Huffman
- 3 – Compression avec pertes
  - Transformée en cosinus discrète (DCT)
  - Compression JPEG
- Sélection de références

Généralités sur la compression de données

- **Objectif**
  - **Réduire le volume de données nécessaire au codage d'un signal numérique**
  - **Pour faciliter son stockage ou sa transmission par réseau**
- **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
    - 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
    - Que perdre ?

Mesurer la compression

- **Notations**

$$I \xrightarrow{\text{compression}} \check{I} \xrightarrow{\text{décompression}} \hat{I}$$

image originale — image compressée — image décompressée

- **Mesures de performance**
  - **Taux de compression**

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

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

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

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

Types de compressions (1/2)

- **Avec ou sans perte**

	Compressions sans perte (codages)	Compressions avec perte
Exemples	• 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	• par moyennage de blocs • par transformation linéaire optimale ou de Karhunen-Loeve (KLT) • par transformée en cosinus discrète : JPEG • par transformée en ondelettes : JPEG 2000 • par quantification : quantification scalaire ou vectorielle • par détection de motifs images redondants (fractale)
Remarques	• Taux de compression limité • Aucune perte d'information	• Bon taux de compression • Perte d'information

Types de compressions (2/2)

Compression des principaux formats d'images bitmaps

Format	Espaces couleur	Compression(s)	C. α	Domaines d'utilisation, rem.
TIFF (.tif)	*RGB*, CIE $L^*a^*b^*$, *CMYB*, 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)	*RGB*, 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)	*RGB*, *CMYB*, ndg	Avec perte (JPEG)	Non	Pages web, photographie Compr. efficace mais destructive
PNG (.png)	*RGB*, 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.

Codage RLE (1/3)

- **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).**

Codage RLE (2/3)

- **Amélioration : décomposition en plans de bits**
  - **Les plages *apparemment* uniformes présentent en fait de faibles variations**

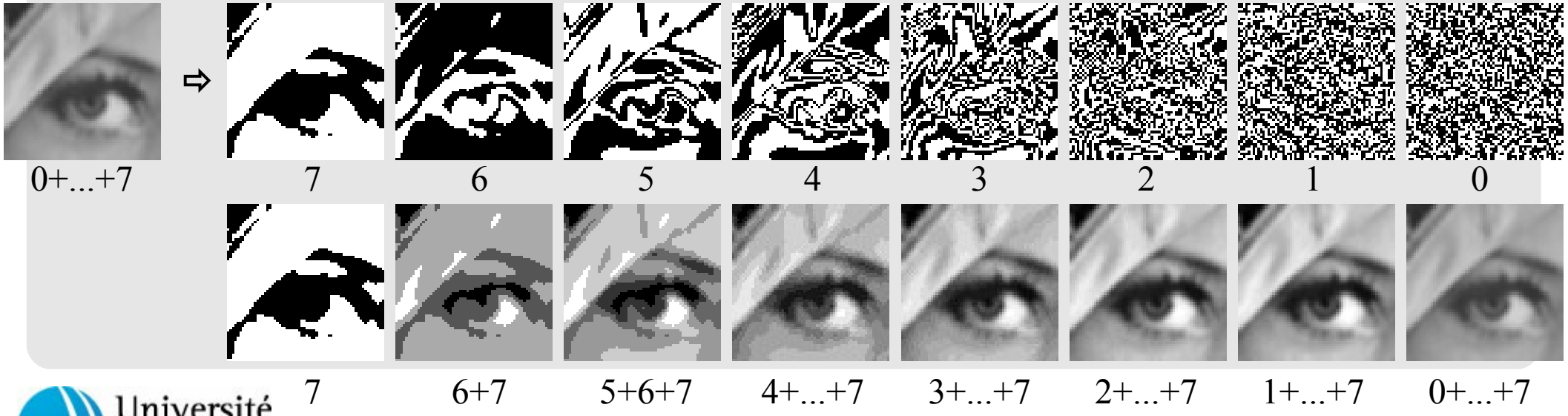
100	99	100	101	100	101	100	99

Plan 7	0	0	0	0	0	0	0	0	1 plage <0,8>
Plan 6	1	1	1	1	1	1	1	1	1 plage <1, 8>
Plan 5	1	1	1	1	1	1	1	1	1 plage <1,8>
Plan 4	0	0	0	0	0	0	0	0	1 plage <0,8>
Plan 3	0	0	0	0	0	0	0	0	1 plage <0,8>
Plan 2	1	0	1	1	1	1	1	0	4 plages <1,1><0,1><1,5><0,1>
Plan 1	0	1	0	0	0	0	0	1	4 plages <0,1><1,1><0,5><1,1>
Plan 0	0	1	0	1	0	1	0	1	8 plages unitaires

  - **Forte cohérence entre pixels voisins au niveau des plans de bits.**

Codage RLE (3/3)

- **Amélioration : décomposition en plans de bits (suite)**
  - **Décomposition en 8 plans de bits ⇨ plages constantes plus longues ⇨ codage RLE sélectif**
    - Efficace sur plans de poids fort
    - Inutile sur plans 0 et 1
  - **Contenu informatif des différents plans**
    - plans de poids faibles (0..2) : surtout du bruit
    - plans médians (3..5) : (très approximativement) information de contours, textures
    - plans de poids forts (6..7) : information de contraste



Codage de Huffman (1/3)

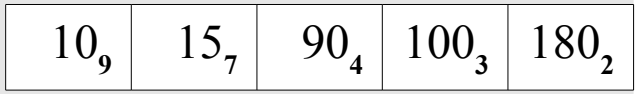
- **Principe**
  - **Coder les valeurs avec un nombre de bits différent.**
    - Code (utilisant des mots) à longueur variable (*ang.* « Variable Length Coding »),
    - dit aussi codage entropique (*ang.* « Entropy coding »).
  - **Plus une valeur apparaît fréquemment, plus le nombre de bits utilisés 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.

Codage de Huffman (2/3)

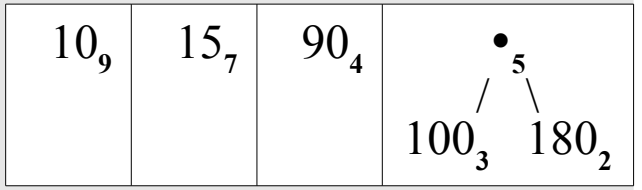
- **Exemple de codage**
  - **Construction de l'arbre**

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

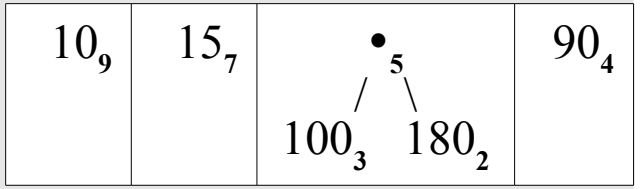
1. Table des poids



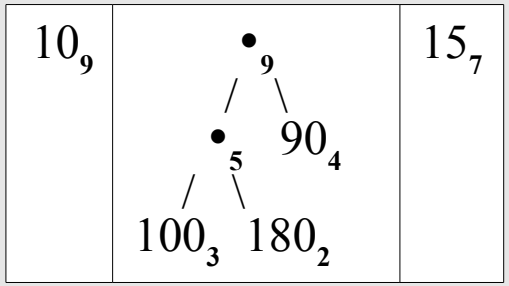
2. Fusion des poids minimaux

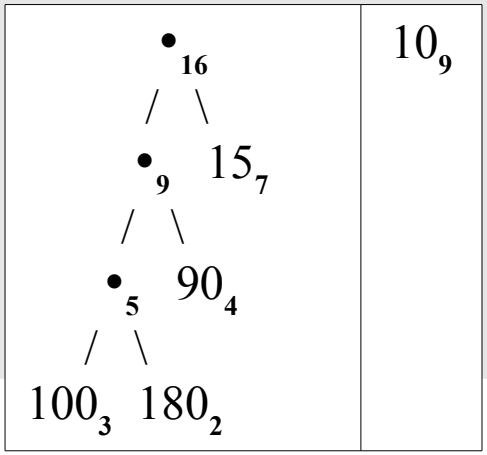


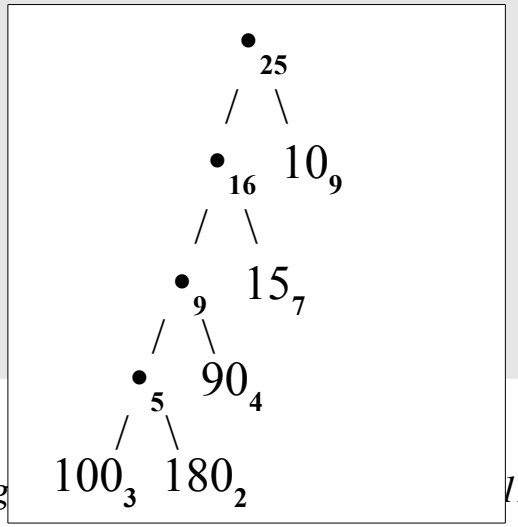
3. Réordonnancement



4. Itérations





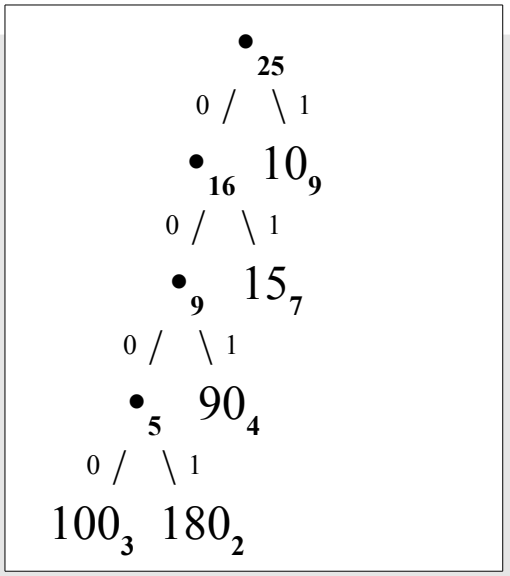


Codage de Huffman (3/3)

- **Exemple de codage (suite)**
  - **Construction du code**

    Affectation de valeurs binaires aux arcs
  - **Image codée (en lignes)**

    1010101011001000 0...
    soit 55 bits vs. 25x8=200 bits



⇨ code

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

- **Décodage**
  - **Propriété du préfixe unique : aucun code n'est le préfixe d'un autre**
    ⇨ décodage non ambigu
  - **Le décodeur**
    - doit connaître la table de codage (entête) ;
    - extrait les valeurs au plus tôt :

Entrée	Action	Buffer	Émission
1	Identification de 10	vide	10
0	Bufferise et attend	0	rien
1	Identification de 15	vide	15
0	Bufferise et attend	0	rien
1	Identification de 15	vide	15
…	…	…	…

Transformée en cosinus discrète (1/6)

Transformée de Fourier discrète 2D (*ang.* DFT) (*rappel*)

Notations

Domaine spatial (pixels)

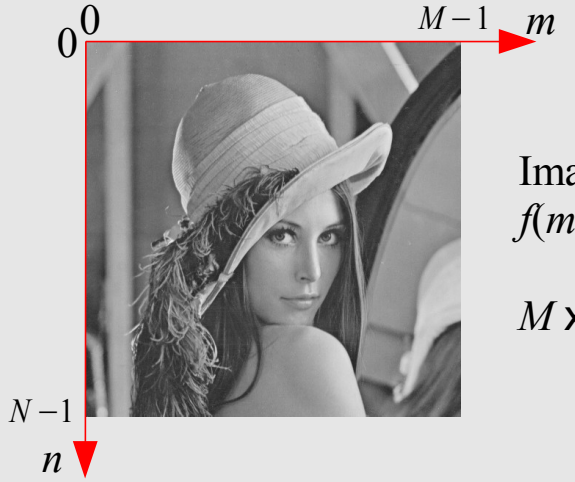


Image $f(m,n)$

$M \times N$

DFT

IDFT

Transformée $F(u,v)$

$M \times N$

Domaine fréquentiel (cycles/pixel)

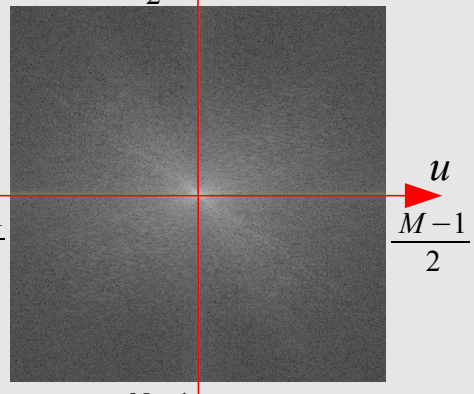


Image du spectre $\log|F(u,v)|$

DFT et DFT inverse

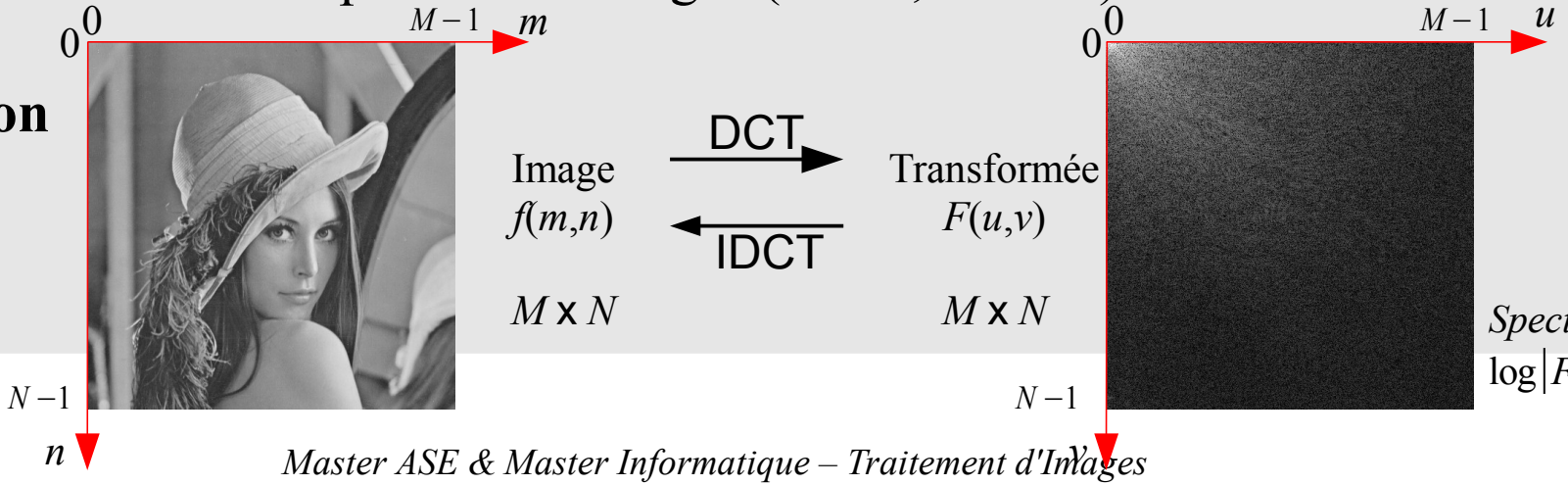
$$F(u,v) := \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\, e^{-j2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)}$$

$$= \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \cdot \left[\underbrace{\cos 2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)}_{C_{m,n}^{M,N}(u,v)} - j \cdot \underbrace{\sin 2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)}_{S_{m,n}^{M,N}(u,v)} \right]$$

$$f(m,n) := \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\, e^{+j2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)}$$

Transformée en cosinus discrète (2/6)

- **Introduction à la transformée en cosinus discrète (*ang.* DCT)**
  - **Inconvénients de la DFT : sur un signal *f* réel,**
    - produit un signal *F* de spectre symétrique ; seule la moitié des coefficients spectraux a donc besoin d'être calculée ;
    - produit un signal *F* complexe, sans que sa partie réelle ou imaginaire seule permette de représenter (donc de reconstruire) le signal *f*.
  - **La DCT est une transformation spectrale (parmi d'autres) qui**
    - possède les mêmes propriétés que la DFT ;
    - s'applique uniquement sur les signaux réels ;
    - est définie par des fonctions de base en *cosinus* seulement ;
    - est utilisée en compression d'images (JPEG, MPEG).
  - **Repré-sentation**



Transformée en cosinus discrète (3/6)

DCT et DCT inverse

En 1D

$$F(u) := \sqrt{\frac{2}{M}}\, c(u) \sum_{m=0}^{M-1} f(m) \underbrace{\cos\left(\pi \frac{(2m+1)u}{2M}\right)}_{D_m^M(u) = D_u^M(m)}$$

$$f(m) := \sqrt{\frac{2}{M}} \sum_{u=0}^{M-1} c(u) F(u) \overbrace{\cos\left(\pi \frac{(2m+1)u}{2M}\right)}$$

Coef. de normalisation :

$$c(\alpha) := \begin{cases} 1/\sqrt{2} & \text{si } \alpha = 0, \\ 1 & \text{si } \alpha \neq 0. \end{cases}$$

pour $\alpha \in \{u, v\}$

En 2D

$$F(u,v) := \frac{2}{\sqrt{MN}}\, c(u) c(v) \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \underbrace{\cos\left(\pi \frac{(2m+1)u}{2M}\right)}_{D_m^M(u) = D_u^M(m)} \underbrace{\cos\left(\pi \frac{(2n+1)v}{2N}\right)}_{D_n^N(v) = D_v^N(n)}$$

$$f(m,n) := \frac{2}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} c(u) c(v) F(u,v) \overbrace{\cos\left(\pi \frac{(2m+1)u}{2M}\right)} \overbrace{\cos\left(\pi \frac{(2n+1)v}{2N}\right)}$$

Transformée en cosinus discrète (4/6)

DCT et DCT inverse (suite)

Lien avec la DFT

- Par rapport à la DFT, la résolution fréquentielle du spectre est doublée dans la DCT : *cf.* fonctions de base
  - DFT *(cos seul)* $C_u^M(m) := \cos\left(2\pi \frac{m\,u}{M}\right)$
  - DCT $D_u^M(m) := \cos\left(\pi \frac{(2m+1)u}{2M}\right) = \cos\left(2\pi \frac{(m+0{,}5)\,u}{\mathbf{2}\,M}\right)$

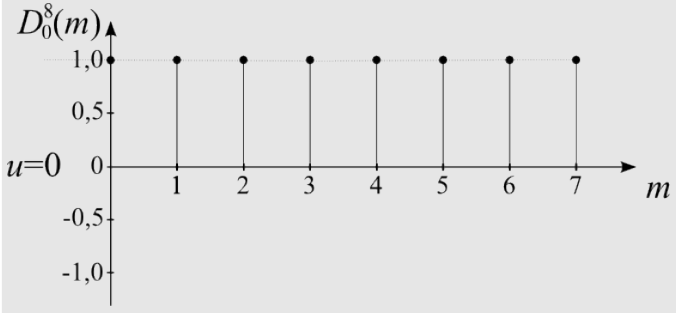
Séparabilité

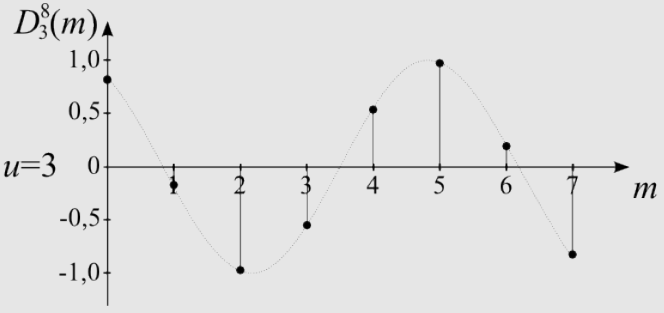
- Comme la DFT 2D, la DCT 2D peut être séparée en deux transformées 1D
  - DFT $F(u,v) := \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \left[\overbrace{\frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} f(m,n)\left(C_u^M(m) - j\,S_u^M(m)\right)}^{\text{DFT 1D de } f(:,n) \text{ (ligne } n\text{)}} \right] \cdot \left(C_v^N(n) - j\,S_v^N(n)\right)$
  - DCT $F(u,v) := \sqrt{\frac{2}{N}}\,c(v) \sum_{n=0}^{N-1} \left[\overbrace{\sqrt{\frac{2}{M}}\,c(u) \sum_{m=0}^{M-1} f(m,n)\,D_u^M(m)}^{\text{DCT 1D de } f(:,n) \text{ (ligne } n\text{)}} \right] \cdot D_v^N(n)$

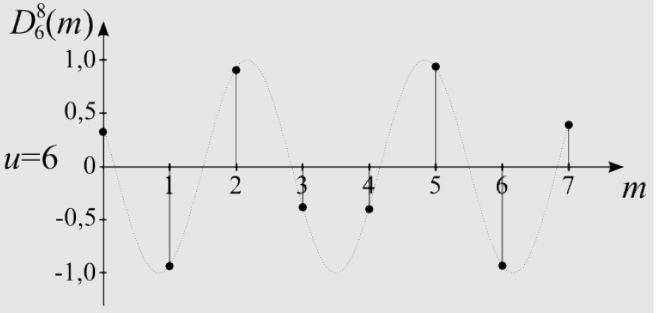
Transformée en cosinus discrète (5/6)

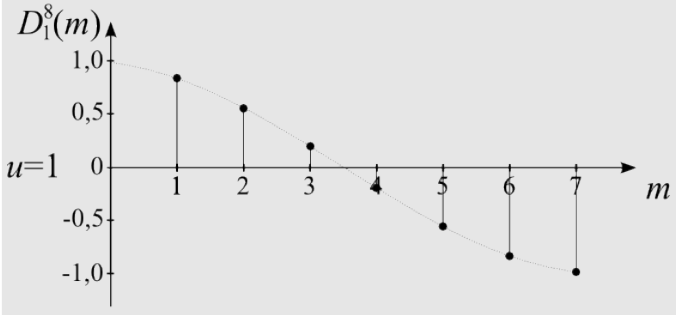
- **Fonctions de base**
  - **En 1D pour *M*=8**

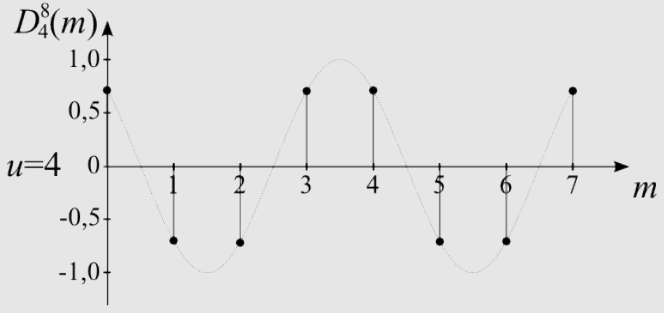
$$D_u^8(m) := \cos\left(\pi \frac{(2m+1)u}{16}\right)$$

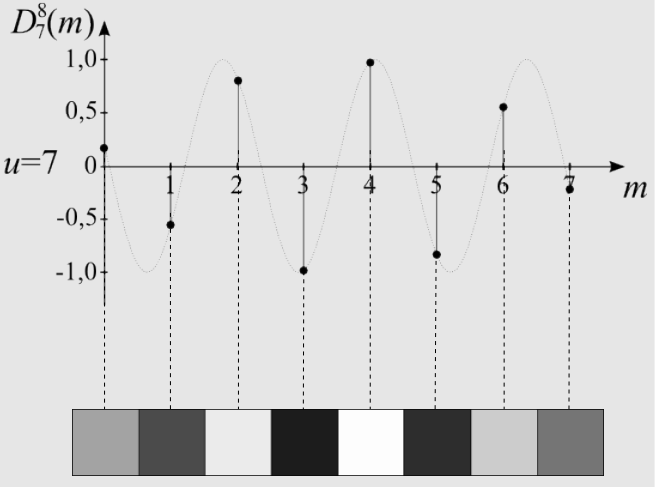


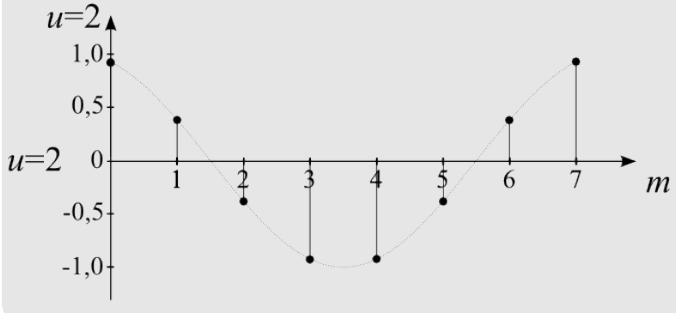


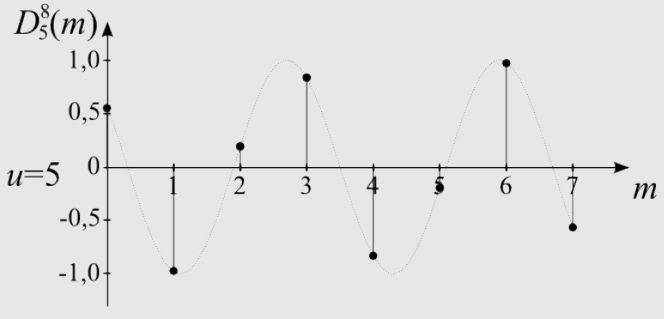








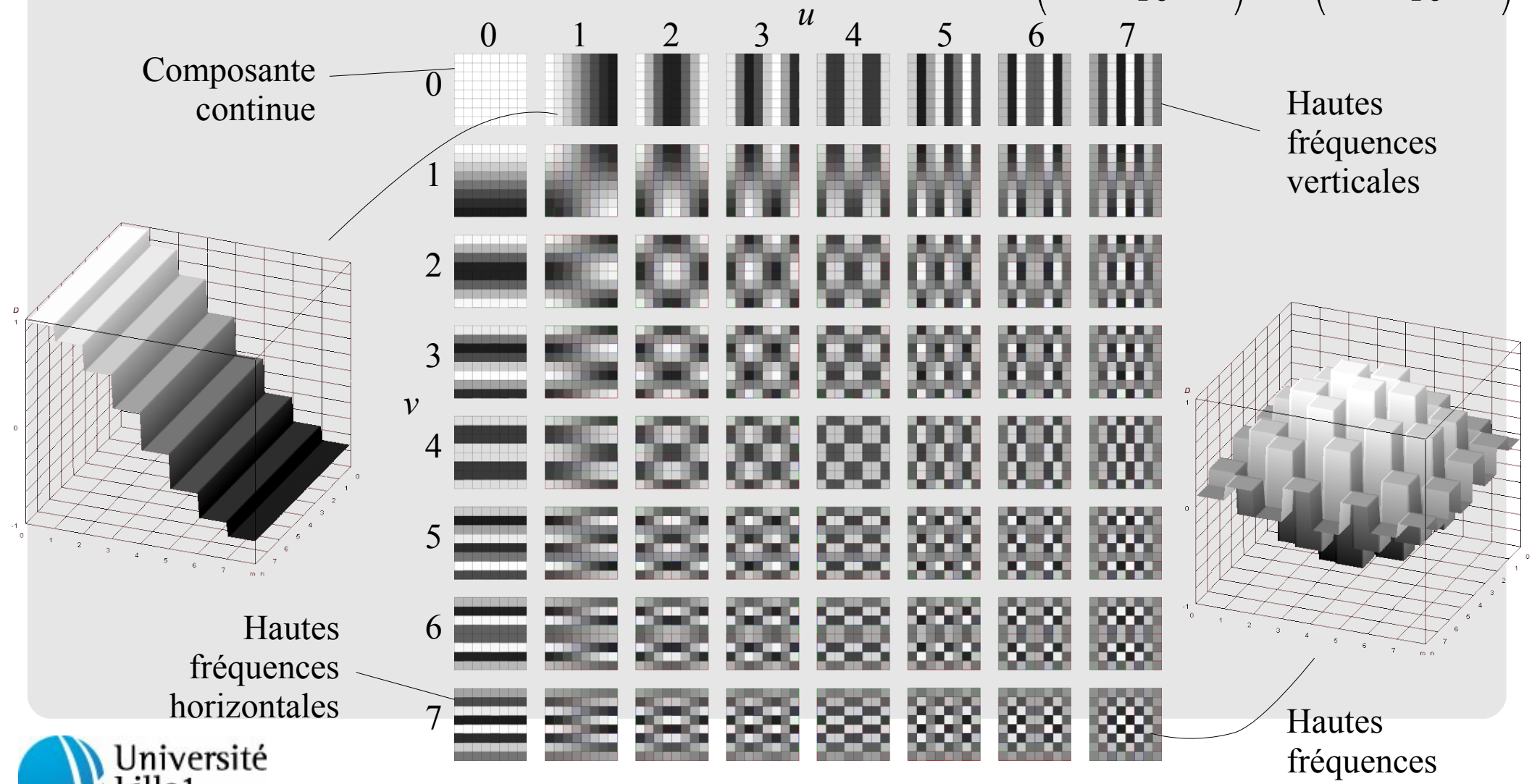




Représentation en niveaux de gris

Transformée en cosinus discrète (6/6)

- **Fonctions de base**
  - **En 2D pour *M*=*N*=8** $D^{8,8}_{u,v}(m,n) = D^8_u(m)\,D^8_v(n) := \cos\left(\pi\frac{(2m+1)u}{16}\right)\cos\left(\pi\frac{(2n+1)v}{16}\right)$

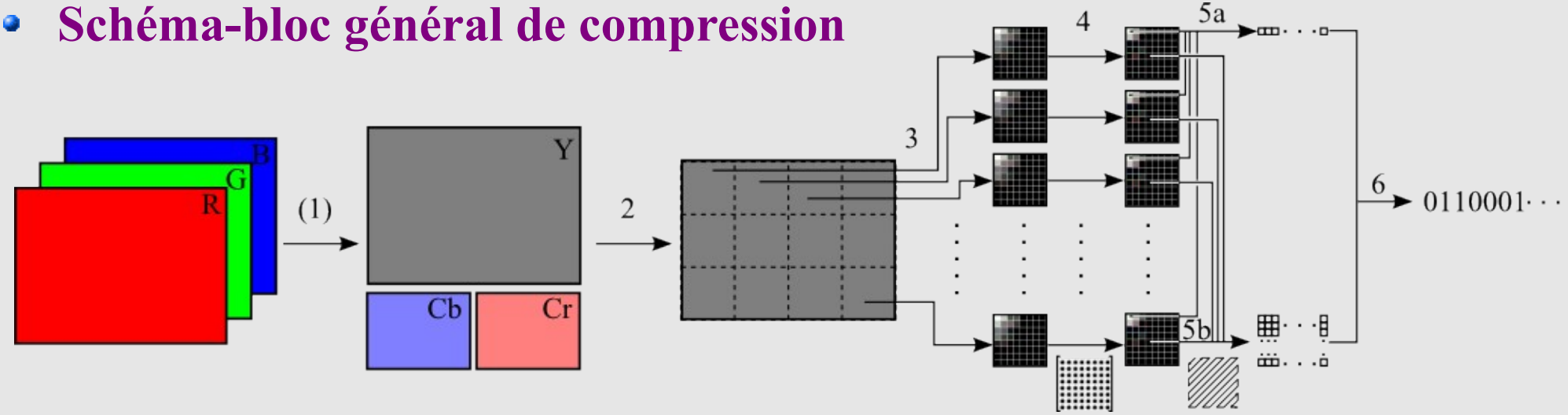


Compression JPEG (1/7)

- **Généralités**
  - **JPEG (*Joint Photographic Expert Group*) : standard depuis 1992.**
    - Images en ndg et couleur jusqu'à 24 bits, de qualité photographique.
    - Nombreux domaines d'applications : photo/vidéo en MM, astronomie, ...
  - **Méthode basée sur une transformation (DCT 2D).**
  - **Ratio de compression nettement plus élevé que sans perte** (25:1 acceptable)**.**
- **Distorsion**
  - **Perte irréversible ⇨ artefacts de compression**
  - **Minimiser la distorsion perceptible**
    - Choix de perte basés sur des expériences psychovisuelles
    - Dégradation *uniforme* de l'image
    - Pas de limite à la compression (choix utilisateur fonction de l'application)
  - **Sources de perte lors de la compression JPEG**
    - **Quantification** des coefficients de la DCT (+ éventuellement des couleurs)
    - **Arrondis** de nombres réels en entiers

Compression JPEG (2/7)

- **Principe fondamental**
  - **Application de la DCT sur des blocs de 8x8 pixels (*M*=*N*=8).**
  - **Quantification : les coefficients les moins significatifs (de hautes fréquences) sont représentés avec moins de précision, voire éliminés.**
- **Schéma-bloc général de compression**



(1) : Changement d'espace couleur et sous-échantillonnage de la chrominance (*facultatif*)

2 : Découpage de chaque plan Y,C_b,C_r en blocs de 8x8 pixels

3 : DCT sur chaque bloc

4 : Quantification de chaque bloc suivant table

5a : Codage différentiel des coefficients DC

5b : Codage RLE des coef. AC en parcours zigzag

6 : Codage entropique (de Huffman ou arithmétique) des séquences DC et AC ⇨ Signal compressé

Compression JPEG (3/7)

- **Transformation (facultative) RGB → YC_bC_r**
  - **Avantage :**
    - L'œil humain est moins sensible aux détails de l'information de couleur (**chrominance**) que de ceux de l'intensité (**luminance**).
    - ⇨ Possibilité de
      - compresser davantage la chrominance sans perte visible de qualité (JPEG) ;
      - réserver une bande passante plus étroite (1/4) à la chrominance (signal TV).
  - **Principe :**
    - Séparation des informations de luminance (Y) et de chrominance (C_b et C_r).
    - Codage des ***différences*** de composantes : $C_b \propto B\text{-}Y$ et $C_r \propto R\text{-}Y$.

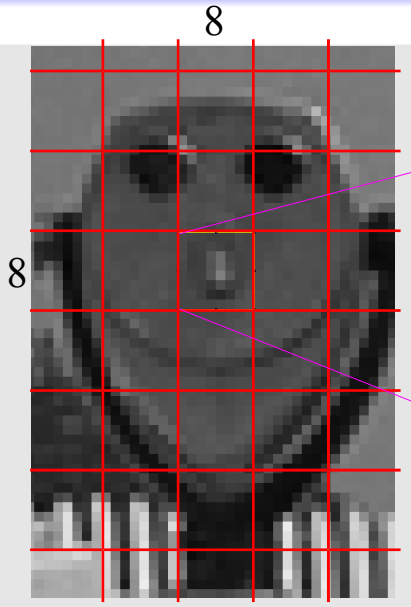
$$\begin{cases} Y &= w_R \cdot R + (1 - w_B - w_R) \cdot G + w_B \cdot B \\ C_b &= \dfrac{0{,}5}{1 - w_B} \cdot (B - Y) \\ C_r &= \dfrac{0{,}5}{1 - w_R} \cdot (R - Y) \end{cases}$$

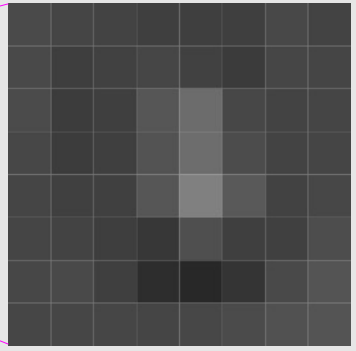
ITU : $w_R = 0{,}299 \quad w_G = 0{,}587 \quad w_B = 0{,}114$

Compression JPEG (4/7)

- **DCT sur blocs 8x8**
  - **Découpage en *P* blocs 8x8**

    Exemple :





$$\begin{bmatrix} 74 & 69 & 67 & 63 & 63 & 63 & 72 & 67 \\ 73 & 62 & 65 & 70 & 65 & 59 & 70 & 69 \\ 75 & 60 & 63 & 86 & 108 & 71 & 67 & 69 \\ 71 & 60 & 63 & 83 & 109 & 76 & 67 & 69 \\ 69 & 64 & 64 & 86 & 128 & 89 & 66 & 71 \\ 69 & 67 & 62 & 55 & 79 & 63 & 64 & 77 \\ 71 & 73 & 63 & 45 & 40 & 52 & 73 & 84 \\ 70 & 70 & 70 & 69 & 71 & 75 & 81 & 84 \end{bmatrix}$$

  - **DCT**
    - Après centrage des valeurs autour de 0 par soustraction de $2^{profBits-1}$ (*ex.* $2^{8-1}=128$).

$$\begin{bmatrix} -54 & -59 & -61 & -65 & -65 & -65 & -56 & -61 \\ -55 & -66 & -63 & -58 & -63 & -69 & -58 & -59 \\ -53 & -68 & -65 & -42 & -20 & -57 & -61 & -59 \\ -57 & -68 & -65 & -45 & -19 & -52 & -61 & -59 \\ -59 & -64 & -64 & -42 & 0 & -39 & -62 & -57 \\ -59 & -61 & -66 & -73 & -49 & -65 & -64 & -51 \\ -57 & -55 & -65 & -83 & -88 & -76 & -55 & -44 \\ -58 & -58 & -58 & -59 & -57 & -53 & -47 & -44 \end{bmatrix} \begin{matrix} \xrightarrow{m} \\ \\ \\ \\ \\ \\ \\ \downarrow n \end{matrix}$$

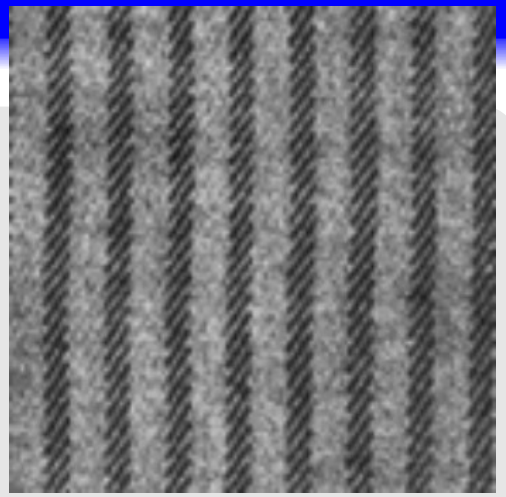
$f(m,n)$

DCT $\longrightarrow$ arrondi à l'entier

$$\begin{bmatrix} \mathbf{-457} & -15 & -8 & 14 & 29 & -12 & -7 & 17 \\ 1 & 10 & -21 & 8 & 11 & 9 & -3 & 2 \\ -24 & 7 & 49 & -24 & -30 & 20 & 3 & -11 \\ -16 & 5 & 16 & -2 & -10 & -7 & 2 & 1 \\ 24 & -7 & -26 & 12 & 4 & -4 & 0 & 0 \\ -14 & 5 & 19 & -7 & -6 & 3 & -1 & -1 \\ 9 & -1 & -12 & 7 & 7 & -6 & -1 & 4 \\ 11 & -1 & -10 & 5 & 4 & -4 & -2 & 0 \end{bmatrix} \begin{matrix} \xrightarrow{u} \\ \\ \\ \\ \\ \\ \\ \downarrow v \end{matrix}$$

$F(u,v)$ { **Coefficient DC (1 valeur)** ; Coefficients AC (63 valeurs)

Compression JPEG (5/7)



- **Quantification (*ang.* « quantization »)**
  - **Principe**
    - Réduire la quantité d'information de hautes fréquences.
    - Car l'œil humain est plus sensible aux basses fréquences.
    - **Principale opération destructive** dans JPEG.
  - **Implémentation**
    - Division de chaque coefficient fréquentiel issu de la DCT, puis arrondi à l'entier.
    - Les diviseurs sont donnés dans une **matrice de quantification** Q.
    - Le standard JPEG fournit une matrice pour la luminance et pour la chrominance.
    - *Exemple* :

$$F = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Q

quantification → arrondi à l'entier

$$\begin{bmatrix} -29 & -1 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & 1 & 3 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

La plupart des coef. de hautes fréquences sont annulés

$$F^*(u,v) = \text{round}\left(\frac{F(u,v)}{Q(u,v)}\right)$$

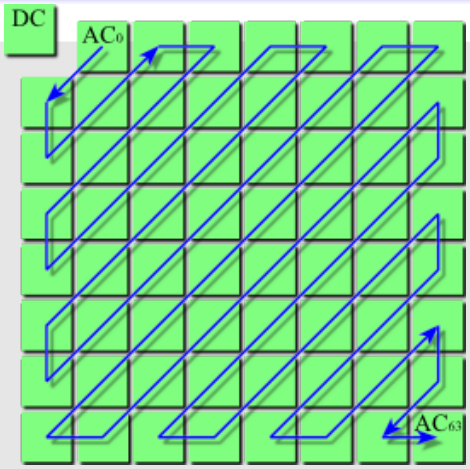
Compression JPEG (6/7)

Codage entropique

Formation des séquences intermédiaires (cas *baseline*)

- Le premier coef. (DC) de chaque bloc i, qui concentre la majeure partie de l'énergie, varie peu d'un bloc à l'autre.
  La séquence $(DC^i)_{0 \leqslant i < P}$ est codée par codage différentiel :

DC^0	DC^1-DC^0	DC^2-DC^1	...	$DC^{P-2}-DC^{P-1}$



- Les autres coef. (AC) de chaque bloc i sont lus en zigzag, formant une séquence globalement décroissante, dont une plage finale de coef. nuls :

$$F^*(u,v) = \begin{bmatrix} -29 & -1 & -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & 1 & 3 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

zigzag $\longrightarrow$ sur AC

$$-1 \quad 0 \quad -2 \quad 1 \quad -1 \quad 1 \quad -1 \quad 1 \quad -1 \quad 1 \quad 0 \quad 3 \quad 0 \quad 1 \quad 0 \quad 0 \quad -1 \quad 1 \quad 0 \quad -1 \quad 0 \quad 0 \quad -1 \quad 0 \quad -1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \ldots$$

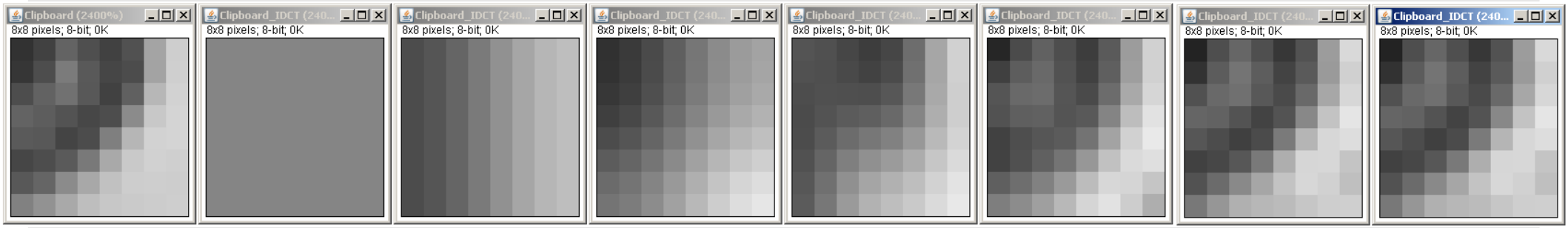
La séquence $(AC_k^i)_{0 \leqslant k < 63,\ 0 \leqslant i < P}$ est ensuite codée bloc par bloc par codage RLE.

Codage entropique (de Huffman ou arithmétique) de ces 2 séquences .

Compression JPEG (7/7)

Influence des coefficients DCT

- **Prise en compte d'un nombre croissant de coefficients *(parcours zigzag)***



Original 1 coef. 2 coef. 4 coef. 8 coef. 16 coef. 32 coef. 64 coef.

- **Application d'un « facteur de qualité » *q* sur la matrice *Q* :** $Q' = \frac{100}{q} \cdot Q$

q = 7 q = 5 q = 2 q = 10

petits diviseurs *grands diviseurs*

Sélection de références

- **Livres**
  - **Éric Incerti, *Compression d'image – Algorithmes et standards*, Vuibert 2003**
  - **Gilles Burel, *Introduction au traitement d'images – Simulation sous Matlab*, Hermès 2001 (chapitre 8)**
- **Sites web**
  - **Cours de P. Nerzic (U. Rennes)**
    http://perso.univ-rennes1.fr/pierre.nerzic/IN/Cours/S3P3%20-%20Codages%20et%20compression.pdf
  - **Page wikipédia sur JPEG** (*en anglais, plus complète que celle en français*)
    http://en.wikipedia.org/wiki/JPEG
  - **Basics of DCT and Entropy Coding, par Nimrod Peleg**
    www.lokminglui.com/J4DCT-Huff2009.pdf
  - **Applet de démo, par C. G. Jennings**
    http://cgjennings.ca/toybox/hjpeg/index.html
  - **Cours de D. Marshall (U. Cardiff)**
    http://www.cs.cf.ac.uk/Dave/Multimedia/PDF/ (*cf.* chapitres 9 et 10)