

Université Badji-Mokhtar. Annaba
Faculté des sciences de l'ingénierie
Département d'Informatique



Chapitre 4

Compression image JPEG

S3- 2024/2025

Introduction

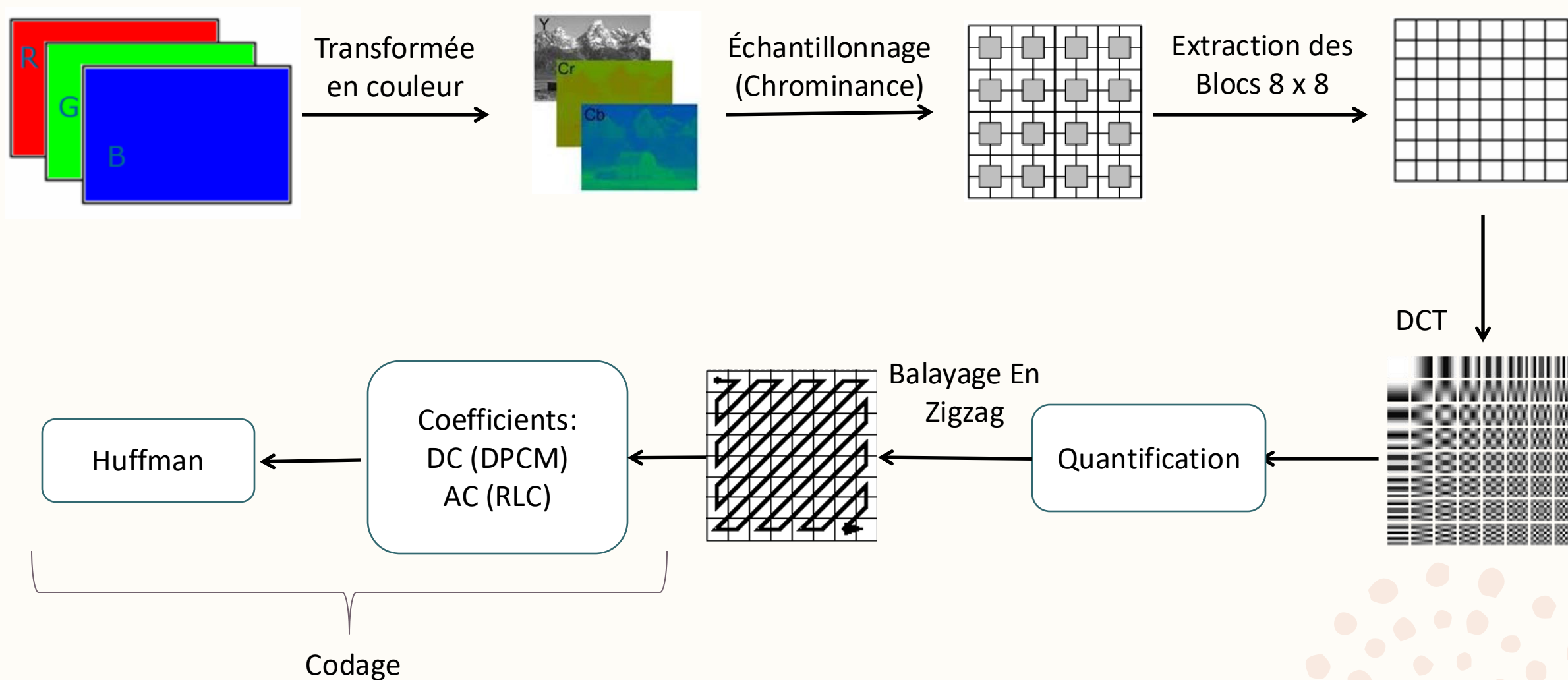
- **La méthode de compression avec pertes:** élimine certaines données pour réduire la taille total des données. Cela peut entrainer une dégradation de la qualité.
- Les objectifs de la compression avec pertes sont d'éliminer les données non pertinentes.
- Utilisé dans le domaine des fichiers multimédias: images, audio, vidéo.
- **Exemples:** JPEG, MPEG, MP3.

Compression JPEG

- **JPEG** est une norme de compression d'image qui a été développée par le « **Joint Photographic Experts Group** » en 1992.
- JPEG est une méthode de compression d'image avec perte.
- Cela permet d'obtenir des fichiers de taille nettement plus petite avec peu ou pas d'impact perceptible sur la qualité et la résolution de l'image.
- JPEG fonctionne en supprimant les informations qui ne sont pas facilement visibles à l'œil humain tout en conservant les informations que l'œil humain est capable de percevoir.

Compression JPEG

- Les étapes de compression JPEG



Compression JPEG

1. La transformée en couleur :

- ✓ La vision humaine: manque de sensibilité couleurs et forte sensibilité aux variations des intensités lumineuses.
- ✓ JPEG modifie l'espace colorimétrique de l'image de RGB à YCbCr.
- ✓ Le mode YCrCb sépare la luminosité du pixel
 - **Y** représente la luminance qui décrit le degré de clarté ou d'obscurité d'un pixel, capturant essentiellement la version en niveaux de gris d'une image.
 - **Cb** représente la composante de chrominance liée à la différence de couleur bleue.
 - **Cr** représente la composante de chrominance liée à la différence de couleur rouge.

Compression JPEG

1. La transformée en couleur :

✓ Changement d'espace colorimétrique

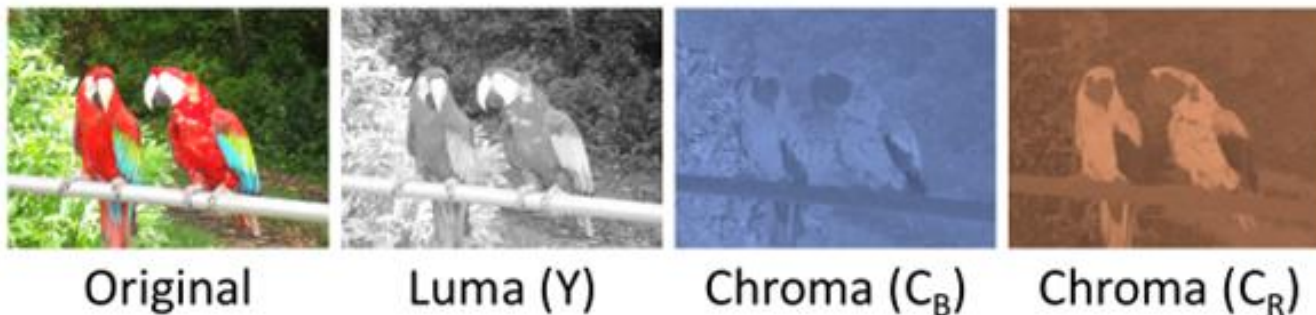
■ RGB -> YCrCb

Luminance : $Y = 0.299 R + 0.587 G + 0.114 B$

Chrominance : $C_b = -0.1687 R - 0.3313 G + 0.5 B$

$C_r = 0.5 R - 0.4187 G - 0.0813 B$

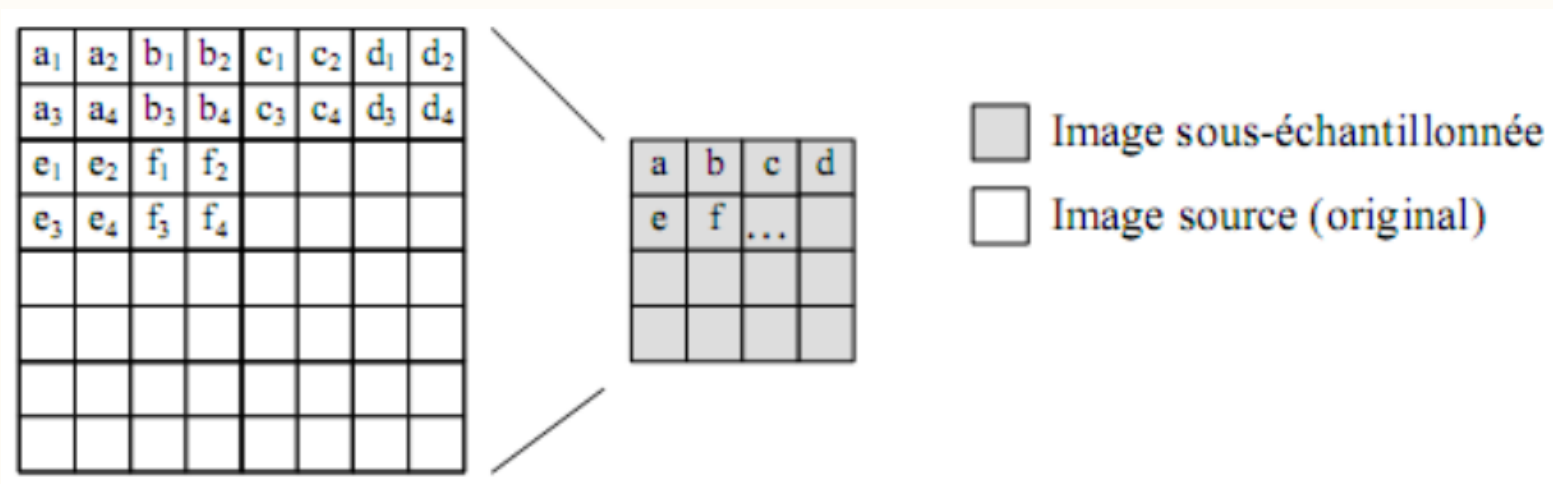
■ Images Chrominance : **moins d'information -> fort taux de compression**



Compression JPEG

2. Sous échantillonnage:

- ✓ L'œil ne distingue pas bien la différence de couleurs entre pixels voisins, donc :
- ✓ Coder la luminance de chaque pixel
- ✓ Sous échantillonnage de la chrominance (consiste à réduire la taille d'une image ou bien suppression de pixels)

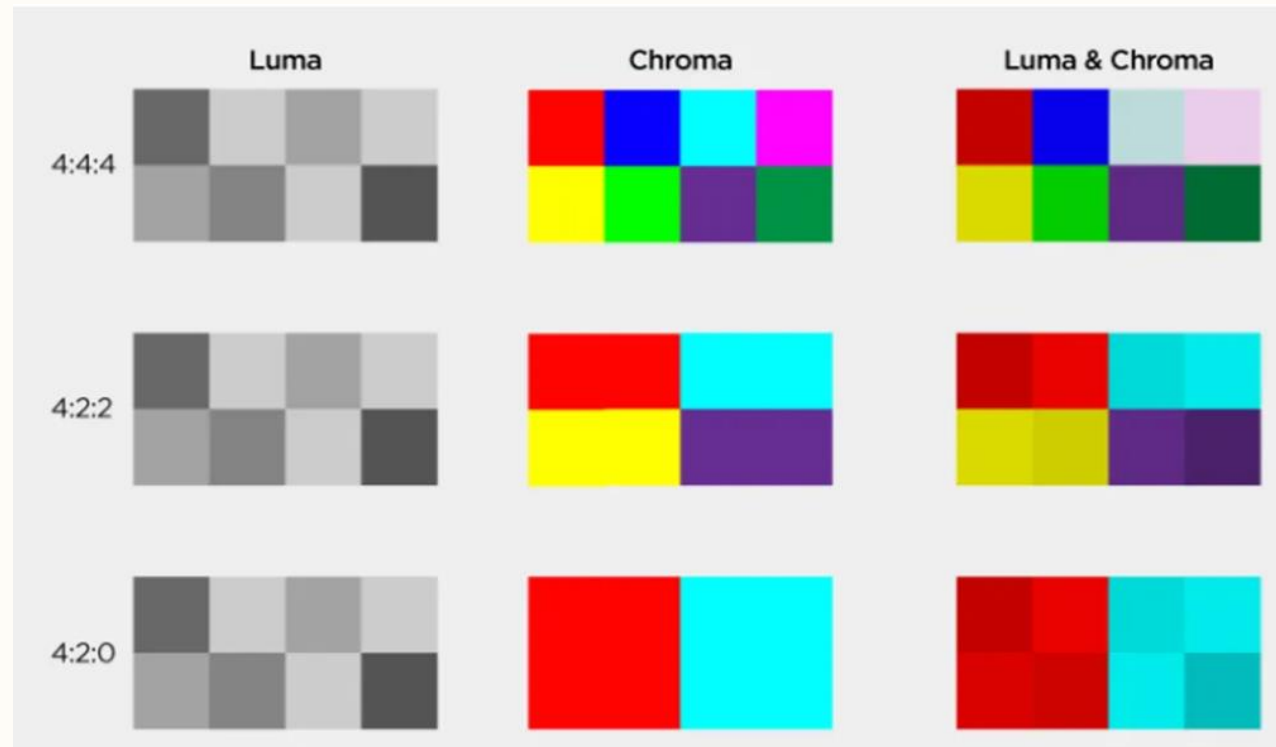


Compression JPEG

2. Sous échantillonnage:

Les schémas de sous-échantillonnage sont :

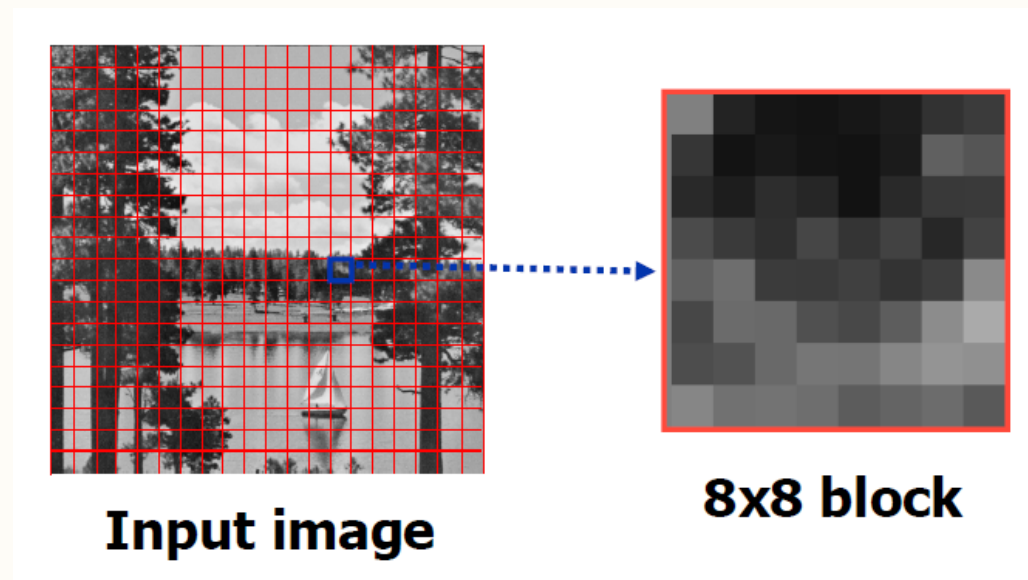
- ✓ 4:4:4: pas de sous-échantillonnage
- ✓ 4:2:2: sous-échantillonnage chroma horizontal
- ✓ 4:2:0: sous-échantillonnage chroma horizontal et vertical



Compression JPEG

3. Extraction des Blocs 8 x 8:

- ✓ JPEG divise l'image en blocs de 8x8 pixels.
- ✓ Pour chaque bloc de 8x8, les données de l'image (pour les composants Y, Cb et Cr) sont traitées séparément.



Compression JPEG

4. La transformation DCT:

- ✓ Pour chaque bloc de 8x8 pixels, JPEG applique une opération mathématique appelée transformée en cosinus discrète (DCT).
- ✓ La DCT convertit les informations du domaine spatial (valeurs des pixels) en informations du domaine fréquentiel.
 - ✓ **Les coefficients basse fréquence** représentent les informations générales de l'image (changements progressifs et fluides de couleur et de luminosité).
 - ✓ **Les coefficients haute fréquence** représentent les détails fins, les contours et le bruit.



Compression JPEG

4. La transformation DCT:

- ✓ La transformation calcule les coefficients $F(u,v)$ par :

$$F(u, v) = \frac{2}{N} c(u)c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{Img}(x, y) \cos \left[\frac{\pi}{N} u \left(x + \frac{1}{2} \right) \right] \cos \left[\frac{\pi}{N} v \left(y + \frac{1}{2} \right) \right]$$

- $\text{Img}(x,y)$: valeur de pixel à la position (x,y)
- N : taille du bloc (généralement $N=8$ pour JPEG).
- u, v : indices de fréquence (0 à $N-1$).
- $c(u), c(v)$ facteurs d'échelle:

$$\text{ou} \begin{cases} c(0) = (2)^{-1/2} \\ c(w) = 1 \text{ pour } w = 1, 2, \dots, N - 1 \end{cases}$$

Compression JPEG

4. La transformation DCT:

- ✓ Exemple:
- ✓ On considère la Matrice x des valeurs $\text{img}(x,y)$ des pixels d'un bloc

$\text{img}(x,y) =$

212	220	216	216	220	216	220	220
208	216	216	220	220	220	216	220
172	204	212	216	220	216	220	220
144	156	180	208	216	216	216	216
136	140	144	164	196	212	216	216
132	136	140	144	160	184	208	216
132	132	136	136	140	148	168	192
136	136	132	136	140	144	148	160

Compression JPEG

4. La transformation DCT:

- ✓ Exemple:
- ✓ On obtient après transformation la matrice DCT:

$F(u,v) =$

1471	-90	-1	-7	0	-5	-3	0
160	21	-17	-1	-3	-1	-2	0
-16	38	13	-4	1	-1	0	0
-3	-15	15	6	-1	1	1	1
-6	-3	-9	8	3	0	0	0
-3	-1	-2	-8	2	2	0	1
4	-1	1	-1	-5	-1	-2	1
-2	1	2	1	-1	-3	-1	1

Compression JPEG

5. La quantification:

- ✓ Représente la phase non conservatrice de compression JPEG
- ✓ Elle réduit le nombre de bits nécessaires (JPEG supprime certaines de ces informations haute fréquence qui sont moins perceptibles à l'œil humain sans affecter la qualité perçue de l'image).
- ✓ Les matrices de quantification doivent être connues du décodeur, donc transmises avec l'image.
- ✓ Des matrices ont été proposées par les membres du comité JPEG
- ✓ Chaque valeur de la matrice DCT est divisée par un nombre, puis arrondie à l'entier le plus proche.

Compression JPEG

5. La quantification:

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

La table de quantification de chrominance

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

la table de quantification de luminance

Compression JPEG

5. La quantification:

✓ Exemple:

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

$Q(u, v)$

1471	-90	-1	-7	0	-5	-3	0
160	21	-17	-1	-3	-1	-2	0
-16	38	13	-4	1	-1	0	0
-3	-15	15	6	-1	1	1	1
-6	-3	-9	8	3	0	0	0
-3	-1	-2	-8	2	2	0	1
4	-1	1	-1	-5	-1	-2	1
-2	1	2	1	-1	-3	-1	1

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

92	-8	0	0	0	0	0	0
13	2	-1	0	0	0	0	0
-1	3	1	0	0	0	0	0
0	-1	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\hat{F}(u, v)$

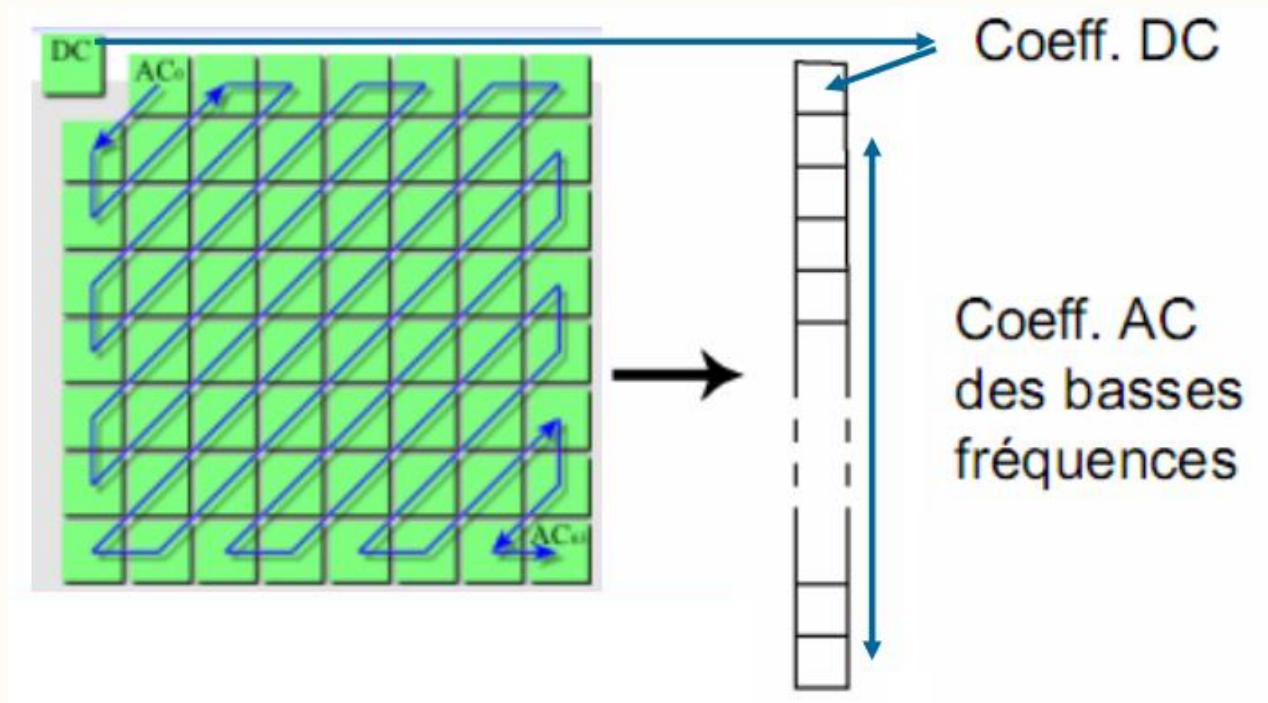
$F(u, v)$

Compression JPEG

6. Codage :

- ✓ Balayage en «zig-zag» (d'un bloc):

Former un vecteur où les coefficients relatifs aux basses fréquences sont regroupés



Compression JPEG

6. Codage :

✓ Differential Pulse Code Modulation (DPCM):

- DPCM est utilisé pour encoder le **coefficient DC**, qui est la première valeur dans l'ordre en zigzag du bloc DCT quantifié.
- Le coefficient DC représente la luminosité moyenne du bloc.
- Les blocs voisins d'une image ont souvent des niveaux de luminosité similaires, de sorte que la différence entre les coefficients DC consécutifs (delta) est généralement faible.
- Le codage de cette différence (au lieu de la valeur DC réelle) réduit le nombre de bits requis.

$$\Delta DC = DC_{\text{actuel}} - DC_{\text{précédent}}$$

Compression JPEG

6. Codage :

✓ Differential Pulse Code Modulation (DPCM):

■ Exemple:

- Supposons que les coefficients DC pour trois blocs soient : 120, 123, 124.
- Différences (ΔDC) : $\Delta DC = [120 - 0, 123 - 120, 124 - 123] = [120, 3, 1]$

Compression JPEG

6. Codage :

✓ Differential Pulse Code Modulation (DPCM):

■ Table Huffman:

- ΔDC est catégorisé en fonction de la plage de ses valeurs.
- Chaque catégorie est codée à l'aide d'un code Huffman.

• Exemple:

- Pour $\Delta DC = 3$:
- Il appartient à la catégorie 2 (plage de 2 à 3).
- Code Huffman : 011.
- Bits supplémentaires : 11 (représentation binaire de 3).
- Valeur codée finale : 01111.

Category	Range	Huffman Code
0	0	00
1	-1 to 1	010
2	-3 to -2, 2 to 3	011
3	-7 to -4, 4 to 7	100
4	-15 to -8, 8 to 15	101
5	-31 to -16, 16 to 31	110
6	-63 to -32, 32 to 63	1110
7	-127 to -64, 64 to 127	11110
8	-255 to -128, 128 to 255	111110
9	-511 to -256, 256 to 511	1111110

Compression JPEG

6. Codage :

✓ Run-Length Coding (RLC):

- RLC est utilisé pour coder les **coefficients AC**, qui représentent les variations du bloc après la suppression de la composante DC.
- Après quantification, de nombreux coefficients AC sont nuls, en particulier pour les composants haute fréquence.
- Au lieu de stocker tous les zéros, RLC encode efficacement les séries de zéros consécutifs.
- Chaque coefficient AC différent de zéro est représenté par une paire (**Run, Value**)
 - **Run** : le nombre de zéros consécutifs avant ce coefficient.
 - **Value** : le coefficient différent de zéro réel.
- Un symbole spécial « **EOB** » est utilisé pour indiquer que toutes les valeurs restantes dans le bloc sont nulles (**1011**).

Compression JPEG

6. Codage :

✓ Run-Length Coding (RLC):

■ Exemple:

- Exemple de données : 4,0,0,0,-2,0,0,3,0,0,0,0,0,-1,0,5, 0,...0
- Compression RLC (0,4) ; (3,-2) ; (2,3) ; (5,-1) ; (1,5) ; EOB.

Compression JPEG

6. Codage :

✓ Run-Length Coding (RLC):

■ Table Huffman :

■ Coefficients AC: **(2,-3)**

■ Run = 2 (2 zéros avant -3).

■ Taille = 2 (-3 nécessite 2 bits)

■ Code de Huffman pour (2, 2) : **111010**.

■ Bits supplémentaires pour -3 : **10** (représentation binaire de -3).

■ Valeur codée finale : **11101010**.

(Run, Size)	Huffman Code
(0, 1)	00
(0, 2)	01
(0, 3)	100
(0, 4)	1010
(0, 5)	10110
(0, 6)	101110
(1, 1)	1100
(1, 2)	11010
(1, 3)	111010
(2, 1)	11100
(2, 2)	111010
(3, 1)	1111000
(3, 2)	1111100
(4, 1)	11111000
(5, 1)	11111100
EOB (End of Block)	1011

Compression JPEG

6. Codage :

✓ Run-Length Coding (RLC):

■ Table Huffman : Exemple 2

- Compression RLC (0,4) ; (3,-2) ; (2,3) ; (5,-1) ; (1,5) ; EOB.
- (Run, Taille): (0,3) ; (3,2) ; (2,2) ; (5,1) ; (1,3) ; EOB.
- Code Huffman: 100, 1111100,1111000,11111100,111010, 1011
- Codage de magnitude 4(100), -2 (01), 3(11), -1(0), 5(101)
- Ajouter les bits de magnitude:

100**100** 1111100**01** 1111000**11** 11111100**0** 111010**101** 1011

(Run, Size)	Huffman Code
(0, 1)	00
(0, 2)	01
(0, 3)	100
(0, 4)	1010
(0, 5)	10110
(0, 6)	101110
(1, 1)	1100
(1, 2)	11010
(1, 3)	111010
(2, 1)	11100
(2, 2)	111010
(3, 1)	1111000
(3, 2)	1111100
(4, 1)	11111000
(5, 1)	11111100
EOB (End of Block)	1011

Décompression JPEG

La décompression est l'inverse du processus de compression :

- ✓ Lit les données compressées.
- ✓ Applique le décodage Huffman pour restaurer les longueurs d'exécution et les coefficients DCT.
- ✓ Inverse l'étape de quantification (bien que ce ne soit pas exact, car la quantification élimine certaines informations).
- ✓ Effectue la DCT inverse (IDCT) pour revenir au domaine spatial.

$$Img(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u)c(v)F(u, v) \cos \left[\frac{\pi}{N} u \left(x + \frac{1}{2} \right) \right] \cos \left[\frac{\pi}{N} v \left(y + \frac{1}{2} \right) \right]$$

- ✓ Reconstitue l'image RGB en inversant la conversion de l'espace colorimétrique YCbCr en RGB.