

## Chapitre 5 :

### Techniques de compression d'image (Norme JPEG)

(La source de ce chapitre : [http://igm.univ-mlv.fr/~dr/XPOSE2013/La\\_compression\\_de\\_donnees/jpeg.html](http://igm.univ-mlv.fr/~dr/XPOSE2013/La_compression_de_donnees/jpeg.html))

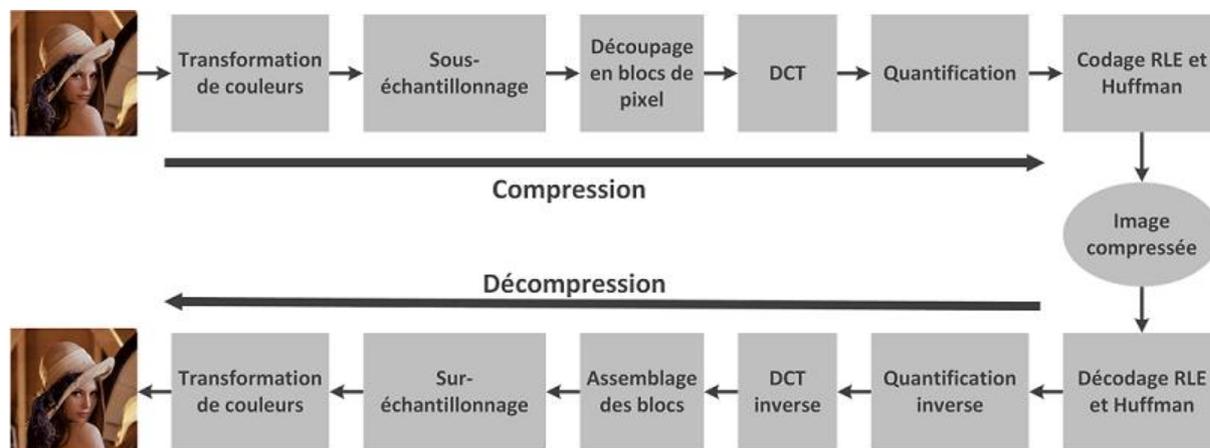
#### Introduction :

Le format JPEG (**J**oint **P**hotographic **E**xperts **G**roup) est un format d'image compressée qui offre une bonne compression pour une qualité très correcte. Ces deux avantages en font l'un des formats d'image les plus répandus, particulièrement sur le web où les problématiques de transfert et de stockage sont importantes.

Le format JPEG est représenté par les fichiers .jpeg ou .jpg et offre des taux de compressions allant de 3 à 100 (autrement dit, le fichier est 3 à 100 fois plus petit que l'image d'origine).

#### 1. Principe de la Compression/Décompression JPEG

La compression d'une image au format JPEG suit un certain nombre d'étapes visant à réduire la place occupée à l'aide de diverses méthodes. Ces étapes sont illustrées dans le schéma ci-dessous :



Les étapes du codage d'une image en utilisant le standard JPEG sont :

1. L'image en base RGB est convertie en base YCbCr et les composantes de chrominance (Cb et Cr) sont sous-échantillonnées. Ensuite les trois composantes sont traitées séparément.
2. Pour chaque composante, on groupe les valeurs par blocs de 8x8 valeurs. Pour chaque bloc on applique un codage par transformée orthogonale avec la transformée en cosinus discrète, DCT (Discrete cosine transform),

3. La matrice obtenue par DCT est quantifiée (ce qui induit une compression destructrice de l'image).
4. Pour chaque composante, on entreprend une compression sans perte des matrices quantifiées en utilisant un codage entropique : celui-ci consiste à parcourir les blocs de l'image en zigzag puis à utiliser une compression RLE et un codage de Huffman (ou arithmétique).

Le stockage de données compressées par la méthode JPEG dans un fichier est défini par la norme JFIF (JPEG File Interchange Format).

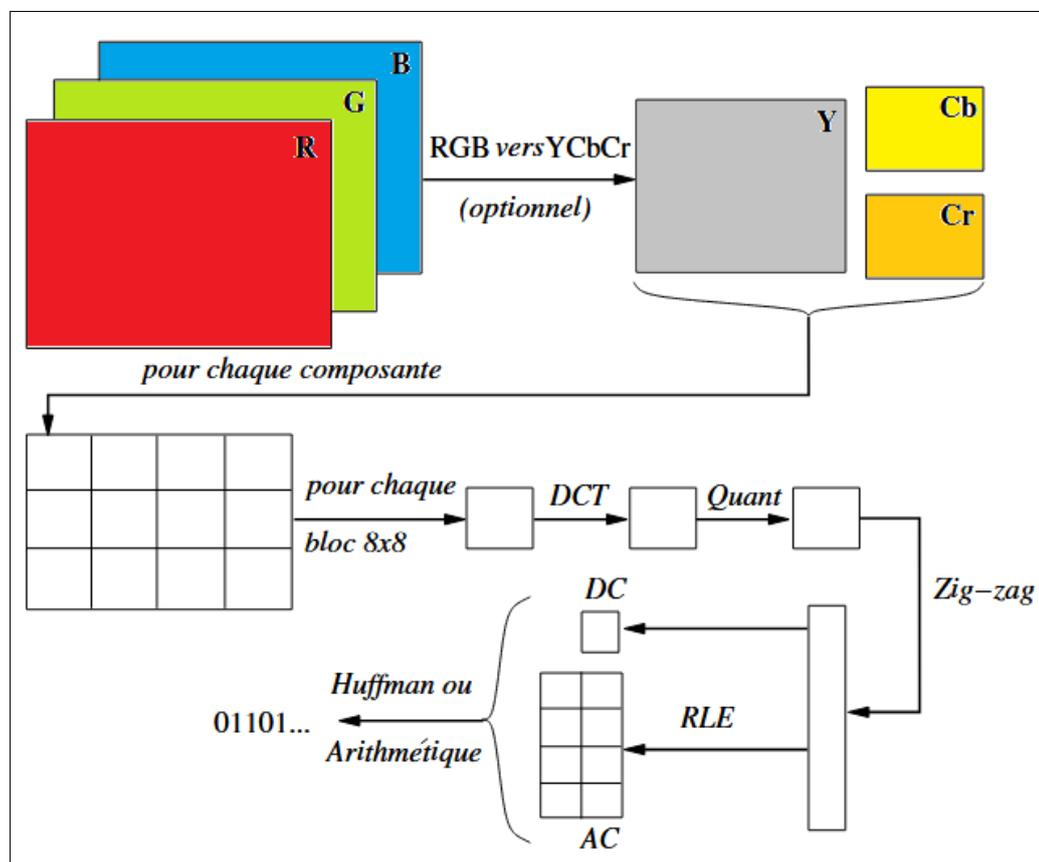


Figure. Compression JPEG

**Remarque :** Le coefficient **DC** (Direct Current) nous donne la valeur moyenne des niveaux du bloc, et les 63 autres coefficients **AC** (Alternative Current) représentent les amplitudes des fréquences spatiales horizontales et verticales dans le bloc.

## 2. Transformation de couleurs

La première étape consiste à passer du modèle initial des couleurs de l'image (souvent RVB) en modèle de type chrominance/luminance (YCrCb).



Ce modèle définit un pixel en fonction de sa luminance (intensité de la couleur) et de deux informations de chrominance (la couleur du pixel). Cette étape va permettre de préparer la prochaine étape : le sous-échantillonnage.

**La transformation affine RGB  $\rightarrow$  Y CbCr :**

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.144 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \times \begin{pmatrix} R \\ V \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

L'ajout de 128 à Cb et Cr permet d'obtenir des valeurs entre 0 et 255.

**La transformation affine retour Y CbCr  $\rightarrow$  RGB :**

$$\begin{pmatrix} R \\ V \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.402 \\ 1 & -0.344 & -0.714 \\ 1 & 1.772 & 0 \end{pmatrix} \begin{pmatrix} Y \\ (Cb - 128) \\ (Cr - 128) \end{pmatrix}$$

## 3. Le sous-échantillonnage

L'objectif ici va être de réduire l'information occupée par la chrominance. Si l'œil humain est très sensible à la luminance, elle distingue par contre très mal la chrominance. Ainsi, une perte d'information sur ce point passera relativement inaperçu.

On distingue 3 types majeurs de sous-échantillonnages :

**Le 4:4:4** est un sous-échantillonnage qui consiste à ne rien faire. Aucune compression n'est effectuée et aucune perte de qualité ne peut être enregistrée.

Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							
Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							
Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							
Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							
Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							
Y	Y	Y	Y	Y	Y	Y	Y
Cb							
Cr							

**Le 4:2:2** est un sous-échantillonnage qui consiste à supprimer les informations de la chrominance sur une colonne sur deux.

Y		Y		Y		Y	
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	
Y		Y		Y		Y	
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	
Y		Y		Y		Y	
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	
Y		Y		Y		Y	
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	
Y		Y		Y		Y	
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	
Y		Y		Y		Y	
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	

**Le 4:2:0** est un sous-échantillonnage qui effectue le même processus que le 4:2:2 mais en effectuant aussi cette opération sur les lignes. On divise donc ici par 4 l'information de la chrominance.

Y		Y		Y		Y	
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	
Y	Y	Y	Y	Y	Y	Y	Y
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	
Y	Y	Y	Y	Y	Y	Y	Y
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	
Y	Y	Y	Y	Y	Y	Y	Y
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	
Y	Y	Y	Y	Y	Y	Y	Y
Cb	Y	Cb	Y	Cb	Y	Cb	Y
Cr		Cr		Cr		Cr	

#### 4. Le découpage en bloc de pixels

À partir d'ici, la compression va travailler sur des blocs de pixels seulement. Ces blocs sont des matrices de pixels 8\*8 soient 64 pixels.

$$f = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix}$$

#### 5. DCT

La DTC (**D**iscrete **C**osine **T**ransform) est une étape qui permet d'évaluer l'amplitude des changements d'un pixel à l'autre afin d'identifier les hautes et basses fréquences. Les basses fréquences, très présentes dans une image, sont des zones unies où les couleurs sont proches les unes des autres. A l'inverse, les hautes fréquences sont des zones de contraste, de changement rapide dans les couleurs.

Cette étape consiste à appliquer à notre matrice de pixels la formule mathématique du DCT afin d'obtenir une matrice des fréquences. Cette matrice sera utilisée dans la prochaine étape : la quantification.

$$f = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix} \rightarrow F = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

#### 6. La Quantification

La quantification a pour objectif d'atténuer les hautes fréquences d'une image qui ont été mis en évidence par la DCT. L'œil humain distingue mal les zones de contrastes (les hautes fréquences) et la quantification va permettre de diminuer l'importance de ces informations superflues.

La technique est simple. Il suffit de diviser notre matrice des fréquences avec la matrice de quantification pour obtenir notre matrice quantifiée (il faut **arrondi** les valeurs de  $F^*$ ).

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

Matrice de quantification

$$F^*(i, j) = Round(F(i, j)/Q(i, j))$$

Cette matrice a un avantage qui sera utilisée dans la prochaine étape : les valeurs non nulles ne sont toutes situées dans le bord haut-gauche de la matrice.

$$F = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix} \quad \rightarrow \quad F^* = \begin{bmatrix} 79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

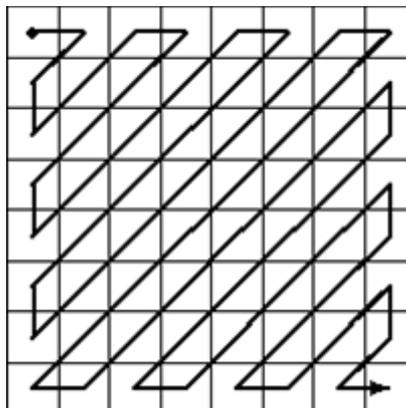
Matrice DCT

Matrice quantifiée

### 7. Le codage

La dernière étape de la compression est le codage. L'objectif va être d'encoder notre bloc de pixel à l'aide du codage **RLE** suivi du codage de **Huffman**.

Comme nous ne savons pas encoder une matrice, nous devons la transformer en chaîne de nombres. Pour cela, nous la parcourons de la façon suivante :



Nous obtenons alors la chaîne "79, 0, -2, -1, -1, -1, 0, 0, -1, 0...0" auquel nous appliquons un codage **RLE** uniquement sur les nombre 0 pour obtenir "79, 0, -2, 3[-1], 2[0], -1, 55[0]"

Nous finissons alors par un codage de **Huffman** pour terminer le codage de ce bloc de pixel. Une fois l'opération appliquée à tous les blocs de pixels, nous obtenons un fichier compressé JPEG.

### **8. Autre méthodes de compression d'images à base de la DWT**

Exemples : EZW ou SPIHT ou JPEG2000 ...