

Projet Image Mpeg

Anaïs Chibani, Lauriane Kiersnowski , Flavie Kolb

Janvier 2023

1 Introduction

Le format Jpeg est un format d'image compressée. Il offre un bon compromis entre niveau de compression et niveau de qualité. C'est l'un des formats les plus répandus sur le Web actuellement. Voyons les étapes de cette compression.

2 Étapes de la compression JPEG

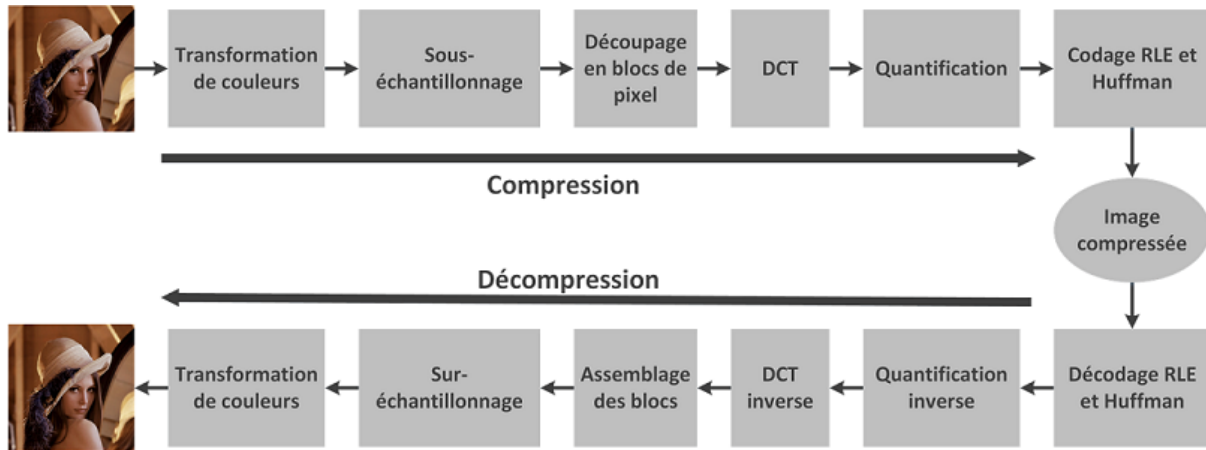


FIGURE 1 – Organisation de la transformation JPEG aboutissant à une image reconstituée [5]

1. Changement d'espace de couleur : La première étape est un changement de la base de couleurs de RGB vers YCbCr. La nouvelle base est composée de la luminance Y (noir et blanc : l'intensité) et de deux chrominances Cb Cr. L'œil humain n'a pas la même sensibilité à la luminance et à la chrominance : il est plus réceptif aux contrastes entre obscurité et luminosité sur une image, et donc à la luminance. Ainsi, en utilisant un espace de couleur différent, certains détails de l'image peuvent être représentés de manière plus efficace, ce qui permet de réduire la taille du fichier.
2. Sous-échantillonnage de chrominance (Chrominance downsampling) : L'œil humain étant moins sensible aux chrominances qu'à la luminance, cette étape consiste à diminuer la taille des chrominances en moyennant la valeur de plusieurs pixels voisins. Cela va créer un lissage de couleur et donc une légère perte d'information, à laquelle l'œil ne devrait pas être trop sensible cependant.
3. Découpage en blocs : Pour la suite des étapes, nous travaillons avec des sous-blocs généralement de taille 8*8 pixels.
4. DCT : Transformée en Cosinus Discrète : L'application de la DCT crée une matrice des fréquences qui évalue l'amplitude des changements de couleurs entre les pixels, elle représente les mêmes données dans une base différente. Cela va permettre d'isoler les fréquences les plus importantes (en bas à droite, équivalent aux parties de l'image avec des détails grossiers et des transitions brusques de couleur) et les plus négligeables (en haut à gauche, équivalent aux parties de l'image avec des détails fins et des transitions douces de couleur) , comme l'illustre la figure 2.

En prenant tous les coefficients i.e. en utilisant tous les blocs de la base, on reconstitue exactement le bloc 8*8 initial. En revanche, en ne prenant seulement les blocs dans le coin supérieur gauche, on s'approche aussi de l'image initiale. Voici le lien d'une animation interactive du fonctionnement de la DCT : [DCT_interactive](#)

En dimension 2 :

DCT :

$$DCT(w, h) = \frac{2}{\sqrt{N * N}} C(m) C(n) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} pixel(i, j) \cos \frac{(2i+1)w\pi}{2N} \cos \frac{(2j+1)h\pi}{2N}$$

Avec $C(k) = \frac{1}{\sqrt{2}}$ quand $k = 0$ et $C(k) = 1$ sinon.

Avec N la largeur/la hauteur du bloc : 8*8.

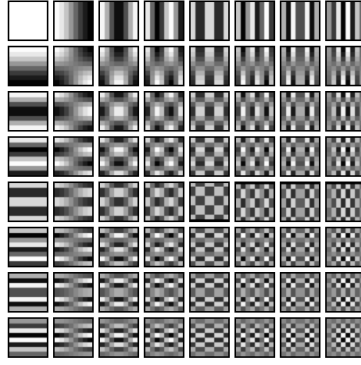


FIGURE 2 – 64 blocs de taille 8*8 de la base après DCT

5. Quantification : Le produit avec la matrice de quantification Q réduit la valeur des nombres obtenus après la DCT afin de gagner de la place et de les stocker sur moins de bits. Elle est appliquée à chaque bloc 8*8 (arrondi des divisions terme à terme). Cette étape crée une perte d'information en enlevant les informations superflues lors de l'arrondi. Les coefficients diminuent plus sur les hautes fréquences peu visibles à l'oeil nu : de nombreux coefficients en bas à droite deviennent donc nuls.

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

Remarque : Une trop grosse quantification ou plusieurs quantifications successives peuvent entraîner de trop grosses pertes et donc la dégradation de l'image. Ce phénomène est appelé artefacts. Sur la figure 3, la quantification à gauche est 5 fois plus importante que celle du milieu, on distingue alors aisément des différences avec l'image originale.

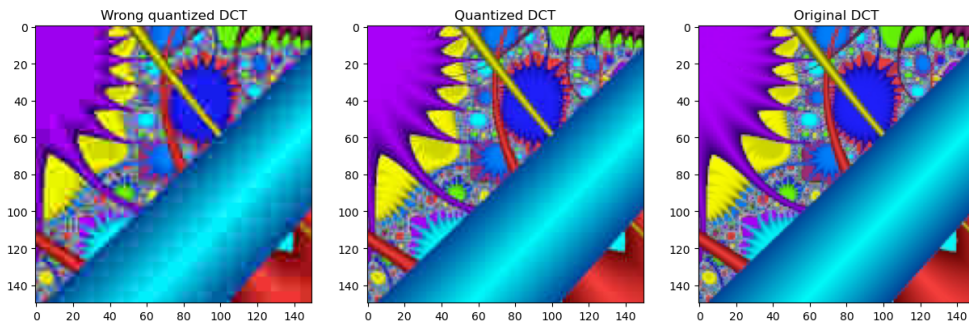


FIGURE 3 – Illustration de l'apparition d'artefacts

6. Lecture en ZigZag : Grâce à l'étape précédente de quantification, nous créons une matrice avec de nombreux 0 et les coefficients importants regroupés en haut à gauche. La lecture en zigzag des matrices permet de regrouper les zéros et de les coder en RLE (Run length encoding), c'est à dire qu'au lieu de stocker une séquence de 0 de taille N dans un vecteur, on crée un tuple : $(N, [0])$. Par exemple, la séquence $[1, 1, 1, 2, 2, 2, 3, 3, 3]$ peut être codée comme $[(3, 1), (3, 2), (3, 3)]$, ce qui représente une économie de place importante si les séquences sont longues.
7. Codage de Huffman : Lors de l'étape du codage de Huffman, chaque élément est associé à une séquence binaire variable. Cette séquence est choisie de telle sorte que les nombres apparaissant le plus fréquemment soient représentés avec les codes les plus courts. Le code de Huffman est un code optimal en terme de longueur moyenne, c'est-à-dire que parmi tous les codes possibles pour représenter les symboles d'un

message, le code de Huffman possède la longueur moyenne la plus courte. Dans la compression Jpeg, une séquence est : (coefficient de la matrice, nombre de 0 précédant cette séquence). Le codage de Huffman est un codage sans perte qui n'altère pas les données.

L'**entropie** du codage de Huffman représente le nombre moyen de bits pour coder un pixel. Elle peut être calculée en utilisant la formule suivante :

$$H = \sum (p_i * l_i)$$

Avec :

- H : entropie du code de Huffman
- p_i : probabilité d'un symbole ('1', '01', '11', '10', '00', '010',...)
- l_i : longueur du code de Huffman associé à ce symbole. (1, 2, 2, 2, 2, 3, ...)

Estimation du gain de stockage :

Usuellement, sur une image composée de 3 canaux, chaque pixel est codé sur 3*8bits : 8 bits pour chacun des coefficients entre 0 et 255.

Le poids de stockage est donc : $8*3 * \text{taille_h} * \text{taille_w}$.

avec taille_h la résolution verticale et taille_w la résolution horizontale.

Avec le codage de Huffman le poids de stockage est $3*H * \text{taille_h} * \text{taille_w}$.

Références

- [1] La compression JPEG : <https://compression.fiches-horaires.net/la-compression-avec-perte-1/le-compression-jpeg/> - 10/12/2022
- [2] À la découverte de la compression JPEG : <https://zestedesavoir.com/articles/1532/la-compression-jpeg/> - 15/11/2022
- [3] JPEG - Définition et Explications : <https://www.techno-science.net/glossaire-definition/JPEG-page-2.html> - 15/11/2022
- [4] Vidéo How are Images Compressed ? [46MB 4.07MB] JPEG In Depth, Branch Education <https://youtu.be/Kv1Hiv3ox8I> - 21/12/2021
- [5] La compression de données : http://igm.univ-mlv.fr/dr/XPOSE2013/La_compression_de_donnees/jpeg.html - 30/12/2022