

Théorie des codes - TP 2

ZZ3 F5 - Année 2013-2014

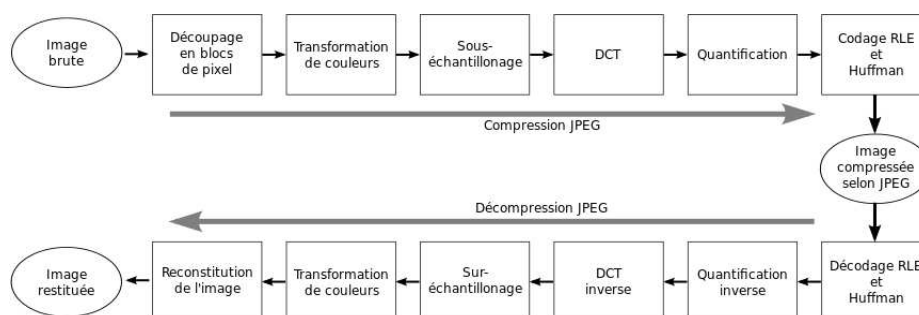
Codage par transformée : Compression par DCT (JPEG)

La norme JPEG est une norme qui définit le format et l'algorithme de décodage pour une image compressée avec pertes. JPEG est l'acronyme de *Joint Photographic Experts Group* qui est un comité d'experts qui édite des normes de compression. La norme, dite JPEG, a pour vrai nom ISO/CEI 10918-1 | UIT-T Recommendation T.81. Ces travaux ont débuté dans les années 1978 à 1980 avec des premiers essais en laboratoire, puis une trentaine d'experts internationaux ont spécifié la norme en 1991. La norme officielle et définitive a été adoptée en 1992.



Introduction La compression JPEG

Le processus de compression et de décompression JPEG¹ comporte six étapes principales représentées ci-dessous :



- **Découpage en blocs** : Le format JPEG commence par découper l'image en blocs carrés de 64 pixels (8×8 pixels).
- **Transformation de couleurs** : JPEG code les couleurs où les meilleurs taux de compression sont obtenus avec des codages de couleur de type luminance/chrominance car l'œil humain est assez sensible à la luminance (la luminosité) mais peu à la chrominance (la teinte) d'une image. Afin de pouvoir exploiter cette propriété, l'algorithme convertit l'image d'origine depuis son modèle colorimétrique initial (en général RVB) vers le modèle de type chrominance/luminance YCbCr. Dans ce modèle, Y est l'information de luminance, et Cb et Cr sont deux informations de chrominance, respectivement le bleu moins Y et le rouge moins Y.
- **Sous-échantillonnage de la chrominance** : Pour exploiter cette faible sensibilité de l'œil humain à la chrominance, on peut procéder à un sous-échantillonnage de ses signaux. Le principe de l'opération est de réduire la taille des blocs de chrominance alors que les blocs de luminance (blocs Y) ne sont pas modifiés (4:4:4, 4:2:0, 4:2:2, ...).
- **DCT** : La transformée DCT (*Discrete Cosine Transform*, en français transformée en cosinus discrète), est une transformation numérique qui est appliquée à chaque bloc. Cette transformée est une variante de la transformée de Fourier. Elle décompose un bloc, considéré comme une fonction numérique à deux variables, en une somme de fonctions cosinus oscillant à des fréquences différentes. Chaque bloc est ainsi décrit en une carte de fréquences et en amplitudes plutôt qu'en pixels et coefficients de couleur. La valeur d'une fréquence reflète l'importance et la rapidité d'un changement, tandis que la valeur d'une amplitude correspond à l'écart associé

1. <http://fr.wikipedia.org/wiki/JPEG>

à chaque changement de couleur. L'application de la DCT est une opération théoriquement sans perte d'informations (aux erreurs d'arrondis près).

- **Quantification** : La quantification est l'étape de l'algorithme de compression JPEG au cours de laquelle se produit la majeure partie de la perte d'information (et donc de la qualité visuelle), mais c'est aussi celle qui permet de gagner le plus de place (contrairement à la DCT, qui ne compresse pas). La DCT a retourné, pour chaque bloc, une matrice de 8×8 nombres. La quantification consiste à diviser cette matrice par une autre, appelée matrice de quantification, et qui contient 8×8 coefficients spécifiquement choisis par le codeur. Le but est ici d'atténuer les hautes fréquences, c'est-à-dire celles auxquelles l'œil humain est très peu sensible. Ces fréquences ont des amplitudes faibles, et elles sont encore plus atténuées par la quantification ; certains coefficients sont même souvent ramenés à 0.
- **Codage RLE et Huffman** : Le codage des coefficients AC et DC sont différents (cf annexe). Le codage des coefficients AC s'effectue en zigzag et se termine par un caractère de fin. Ce résultat est ensuite compressé selon un algorithme RLE basé sur la valeur 0 (le codage RLE intervient uniquement sur cette dernière), puis un codage entropique de type Huffman ou arithmétique. Les deux premières étapes permettent de diminuer l'entropie de la source et ainsi rendre plus performant le codage entropique.

Exercice Transformée et Quantification dans la norme JPEG

Dans un premier temps, nous allons réaliser un encodeur JPEG "light" sans le codage entropique des données. Nous allons découper une image en niveau de gris par des blocs de 8×8 pixels, appliquer une transformée en cosinus discrète, puis quantifier ces valeurs. Afin de visualiser et quantifier la dégradation de l'image, nous allons ensuite la décoder et calculer la distorsion.

1. Réaliser l'algorithme d'encodage (transformée + quantification) de l'image en niveau de gris (plus de détails dans l'annexe).
2. Réaliser l'algorithme de décodage.
3. Réaliser des tests en faisant évoluer le facteur de qualité, c'est-à dire des tests qualitatifs (observation visuelle de l'image reconstruite) et quantitatifs (calcul et tracé de la distorsion).

Exercice Heuristiques et codage entropique dans la norme JPEG

Finalement pour compresser les données, les coefficients quantifiés sont codés par un codage entropique de Huffman. Pour que cet algorithme donne les meilleurs résultats, il y a en amont des heuristiques de diminution entropique : parcours en zig-zag et codage RLE.

1. Réaliser le parcours en zig-zag des coefficients.
2. Réaliser le codage des coefficients AC (plus de détails dans l'annexe).
3. Réaliser le codage des coefficients DC (plus de détails dans l'annexe).
4. Calculer le taux de compression par rapport au facteur de qualité. Comparer ce calcul à la théorie.

Appendix JPEG Encoding

The JPEG encoding process is based on the DCT (discrete cosine transform). The original image should be grouped into 8×8 blocks first. Each 8×8 block is transferred by forward DCT into a set of 64 values referred to as DCT coefficients. Each of the 64 coefficients is then quantized using a quantization table with 64 corresponding values. Before Huffman encoding, the entropy encoding, the two-dimensional image data is converted to one-dimensional zigzag sequences.

1. DCT

The first step in JPEG encoding is level shifting of image by 128, i.e., image data minus 128. After which the forward DCT is applied onto the image data to transform the original two-dimensional data into a set of coefficients. One of these values is referred as the DC coefficient and the other 63 as the AC coefficients. The forward DCT equation is :

$$Y(i, j) = \frac{2}{N} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I(x, y) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2y+1)j\pi}{2N} \right]$$

$$\text{où } C(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } i = 0 \\ 1 & \text{pour } i > 0 \end{cases}$$

The inverse DCT equation is :

$$I(x, y) = \frac{2}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i) C(j) Y(i, j) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2y+1)j\pi}{2N} \right]$$

2. Quantization

Each of the 64 coefficients is then quantized using one of 64 corresponding values from a quantization table. The quantized DCT coefficients are computed with

$$Y_Q(i, j) = \text{round} \left(\frac{Y(i, j)}{Q(i, j)} \right) \text{ for } i = 0, 1, 2, \dots, 7; j = 0, 1, 2, \dots, 7$$

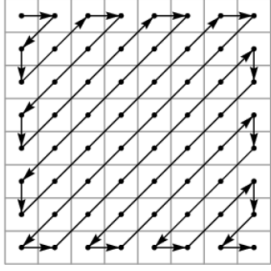
A common quantization matrix is :

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

We can modify this quantization matrix by multiply with a quality coefficient.

3. Zig-zag sequence

After quantization, the DC coefficient and the 63 AC coefficients are prepared for entropy coding, which is Huffman encoding. The previous quantized DC coefficient is used to predict the current quantized DC coefficient, and the difference is encoded. The 63 quantized AC coefficients undergo no such differential encoding, but are converted into a one-dimensional zig-zag sequence ZZ , as shown in the following figure, where the numbers are the zig-zag orders.

$$\mathbf{Z} = \begin{bmatrix} 0 & 1 & 5 & 6 & 14 & 15 & 27 & 28 \\ 2 & 4 & 7 & 13 & 16 & 26 & 29 & 42 \\ 3 & 8 & 12 & 17 & 25 & 30 & 41 & 43 \\ 9 & 11 & 18 & 24 & 31 & 40 & 44 & 53 \\ 10 & 19 & 23 & 32 & 39 & 45 & 52 & 54 \\ 20 & 22 & 33 & 38 & 46 & 51 & 55 & 60 \\ 21 & 34 & 37 & 47 & 50 & 56 & 59 & 61 \\ 35 & 36 & 48 & 49 & 57 & 58 & 62 & 63 \end{bmatrix}$$


4. Huffman Encoding

Encoding DC coefficients :

The DC coefficients are coded differentially, using a one-dimensional predictor, PRED, which is the quantized DC value from the most recently coded 8×8 block from the same component (Y, U, V). The difference, DIFF, is obtained from : $DIFF = ZZ(0) - PRED$. At the beginning of the scan and at the beginning of each restart interval, the prediction for the DC coefficient prediction is initialized to 0. The twos-complement difference magnitudes are grouped into 12 categories, SSSS, as shown the following figure. A Huffman code is created for each of the 12 difference magnitude categories by Huffman DC tables :

SSSS	DIFF Values	Category	Code Length	Code Word
0	0	0	2	00
1	-1,1	1	3	010
2	-3,-2,2,3	2	3	011
3	-7,...,-4,4,...,7	3	3	100
4	-15,...,-8,8,...,15	4	3	101
5	-31,...,-16,16,...,31	5	3	110
6	-63,...,-32,32,...,63	6	4	1110
7	-127,...,-64,64,...,127	7	5	11110
8	-255,...,-128,128,...,255	8	6	111110
9	-511,...,-256,256,...,511	9	7	1111110
10	-1023,...,-512,512,...,1023	10	8	11111110
11	-2047,...,-1024,1024,...,2047	11	9	111111110

Difference Magnitude Categories for DC Coding Table for Luminance DC Difference

Example : Current quantized DC index : 2. Previous block DC index : 4. Prediction error : -2. The prediction error is coded in two parts. Which category it belongs to (Table of JPEG Coefficient Coding Categories), and code using a Huffman code (JPEG Default DC Code). Prediction error = -2 is in category "2", with a codeword "011". Which position it is in that category, using a fixed length code, length=category number. "-2" is the number 1 (starting from 0) in category 2, with a fixed length code of "01". The overall codeword is "01101".

Encoding AC coefficients :

Since many coefficients are zero, runs of zeros are identified and coded efficiently. In addition, if the last part of ZZ is entirely zero, this is coded explicitly as an end-of-block (EOB). Each nonzero AC coefficient in ZZ is described by a composite 8-bit value, R/S, of the form : $R/S = \text{binary "RRRR/SSSS"}$. The four bits, "SSSS", define a category for the amplitude of the next nonzero coefficient in ZZ. The "SSSS" is defined by category table. The four bits, "RRRR", give the position of the coefficient in ZZ relative to the previous nonzero coefficient. The Huffman AC code table consists of one Huffman code (maximum length of 16 bits) for each possible composite value. The Huffman table for luminance AC coefficients and chrominance AC coefficients are shown in a table (H.txt).

Example : Symbol (0/5). The value "5" is represented in two parts. Which category it belongs to (Table of JPEG Coefficient Coding Categories), and code the "(runlength, category)" using a Huffman code (JPEG Default AC Code). AC=5 is in category "3". Symbol (0/5) has codeword "100". Which position it is in that category, using a fixed length code, length=category number "5" is the number 4 (starting from 0) in category 3, with a fixed length code of "011". The overall codeword for (0/5) is "100011".