

Cours/TD 4 Compression par transformée. Codage JPEG

1 Compression par transformée : transformée, quantification, codage près de l'entropie

1.1 Transformées. Bases orthogonales

Definition. Soit E un espace vectoriel muni d'une base $\vec{b}_1, \dots, \vec{b}_n \dots$. Soient \vec{u} et \vec{v} deux vecteurs de E .

$$\vec{u} = \sum_{n \in N} x_n \vec{b}_n \text{ et } \vec{v} = \sum_{n \in N} y_n \vec{b}_n$$

On appelle produit scalaire de \vec{u} et \vec{v} relatif à la base $\vec{b}_1, \dots, \vec{b}_n \dots$, et on note $\langle \vec{u}, \vec{v} \rangle$ la somme des produits deux à deux des coordonnées.

$$\langle \vec{u}, \vec{v} \rangle = \sum_{n \in N} x_n y_n .$$

Definition. Une matrice A est orthogonale si et seulement si :

- A est inversible et son inverse est égale à sa transposée : $A^{-1} = A^t$.
- tous ses vecteurs colonne sont orthogonaux entre eux et de norme 1. Ainsi une matrice orthogonale représente une base orthonormale.

1.2 Codage par transformée

Soit $\{b_n\}_{n \in N}$ une base orthonormée de l'ensemble des images et I l'une de ces images, la relation entre I et ses coefficients c_n dans la base $\{b_n\}_{n \in N}$ est donnée par la formule de décomposition dans une base orthonormée

$$I = \sum_{n \in N} c_n b_n \text{ avec } c_n = \langle I, b_n \rangle$$

C'est un changement de base vers une base plus adaptée. Une base orthonormée permet un contrôle simple sur la norme L_2 ayant la propriété de conservation de l'énergie suivante :

$$\|I\|^2 = \langle I, I \rangle = \sum_{m \in N} \sum_{n \in N} c_m \cdot c_n \langle b_m, b_n \rangle = \sum_{n \in N} c_n^2 \text{ avec } c_n = \langle I, b_n \rangle$$

Il en résulte que si on souhaite approcher la fonction I par I_M , i.e. minimiser $\|I - I_M\|^2$

$$I_M = \sum_{m \in M} c_m b_m \text{ avec } c_m = \langle I, b_m \rangle$$

ont doit sélectionner dans I_M les M plus grands produits scalaires $\langle I, b_m \rangle$.

On souhaite choisir une base telle que l'erreur entre I et son approximation non-linéaire I_M avec M coefficients I_M décroisse rapidement pour une large classe de fonctions. Pour une classe donnée, cette décroissance dépend de la base utilisée et permet de classer les performances d'approximation des différentes bases. Elle est mesurée à l'aide d'un exposant k reliant l'erreur au nombre de coefficients : $\|I - I_M\|^2 \leq kM^{-k}$.

La quantification suit le principe de l'approximation sur une base orthonormée. Elle consiste à utiliser une partition de l'espace des images et à remplacer chaque image par un représentant de la classe à laquelle elle appartient. L'image est alors reconstruite à l'aide des coordonnées quantifiées.

Le codage par transformée est obtenu en quantifiant les coefficients par un quantificateur Q_n connu et en codant, sans perte cette fois, les coefficients quantifiés par des méthodes similaires à celles de la section précédente. L'image dégradée \tilde{I} effectivement compressée est alors donnée par :

$$\tilde{I} = \sum_{n \in N} Q_n(c_n) b_n$$

Pour le standard JPEG la base choisie est dérivée des bases de Fourier : l'image est découpée en blocs de taille 8x8 et chacun de ces sous-blocs est décomposé dans une base de cosinus locaux.

Dans la compression on choisit les produits scalaires qui après l'application de la matrice de quantification sont différents de zero. Ces coefficients sont alors quantifiés et codés à l'aide d'un codage statistique (Huffman ou codage arithmétique) dans lequel les coefficients sont modélisés comme indépendants. La transformée a un effet décorrélant qui justifie cette approximation. Un modèle psycho-visuel est de plus utilisé pour quantifier les coefficients selon leur importance visuelle. Cet algorithme est très efficace pour une large gamme de taux de compression mais présente l'inconvénient de faire apparaître ces blocs 8x8 à fort taux de compression.

Dans un codeur par transformée très simple : l'image I est transformée en ses coefficients dans la base b_n , ceux-ci sont quantifiées de manière uniforme à l'aide d'un pas Δ et l'on code alors les valeurs quantifiées par deux listes : une liste binaire disant pour chaque coefficient s'il est nul ou non et une liste des valeurs quantifiées non nuls. Cette stratégie s'explique par le caractère particulier des coefficients quantifiés à 0.

En effet, en reprenant les notations précédentes, l'erreur introduite par la compression est donnée par la différence entre I et \tilde{I} , et se mesure aisément en norme quadratique :

$$\|I - \tilde{I}\|^2 = \sum_{\substack{n \in N \\ |c_n| \leq \Delta/2}} c_n^2 + \sum_{\substack{n \in N \\ |c_n| > \Delta/2}} (c_n - Q(c_n))^2$$

et en introduisant M_Δ le nombre de coefficients c_n tel que $|c_n| \geq \Delta/2$

$$\|I - \tilde{I}\|^2 \leq \|I - I_{M_\Delta}\|^2 + M_\Delta \Delta^2/4$$

où I_{M_Δ} est obtenu à partir de I en conservant les M_Δ plus grands coefficients en valeurs absolues.

L'étude du terme $\|I - I_{M_\Delta}\|^2$ est le cœur de la théorie de l'approximation dans les bases orthonormées. Pour approcher une fonction (une image) avec M coefficients à choisir librement pour minimiser l'erreur quadratique de reconstruction, la bonne stratégie est de conserver les M plus grands coefficients en valeurs absolues. L'un des objets de la théorie de l'approximation est d'étudier les possibilités d'approximations de classes de fonctions dans une base donnée. Ceci s'exprime bien souvent par une relation entre l'erreur d'approximation $\|I - I_M\|^2$, M et Δ , le seuil associé, de la forme : $\|I - I_M\|^2 \leq CM^{-\gamma}$ et $M_\Delta \leq C'\Delta^{-2\gamma/(\gamma+1)}$, où α est lié à une forme de régularité propre à la classe. L'erreur de compression satisfait alors :

$$\|I - \tilde{I}\|^2 \leq C''M_\Delta^{-\gamma}$$

La taille du code nécessaire pour spécifier les coefficients quantifiés est également liée à cette quantité M_Δ . La proportion de coefficients quantifiés non nuls est de M_Δ/N . L'entropie de la carte binaire de non nullité des coefficients est donc, si les coefficients sont considérés comme indépendants, de $-M_\Delta \log_2(M_\Delta/N) - (N - M_\Delta) \log_2(1 - M_\Delta/N)$. Enfin, chacun des M_Δ coefficients non quantifiés à 0 requiert au plus $C - \log_2 \Delta$ bits pour être spécifiés. Il en résulte, après un développement limité en M_Δ/N , que le nombre total R de bits nécessaire pour coder l'image satisfait $R \simeq M_\Delta(1 + \log_2(M_\Delta/N) + C - \log \Delta) \simeq M_\Delta(C' + \log_2(M_\Delta/N))$.

La combinaison des estimations de $\|I - \tilde{I}\|^2$ et R donne alors :

$$\|I - \tilde{I}\|^2 \leq C R^{-\gamma} \log_2^{\gamma+2}(R)$$

de sorte que l'efficacité de cet algorithme de codage est lié à une performance d'approximation non linéaire dans la théorie de l'approximation. Les performances de l'algorithme JPEG sont ainsi liées à la capacité de la base de Fourier à approcher les fonctions régulières. La base de Fourier n'est cependant pas optimale pour les images et les bases d'ondelettes, introduites plus récemment, possèdent de meilleures propriétés. C'est donc tout naturellement qu'elles ont été utilisées dans le nouveau standard JPEG 2000. Les ondelettes, introduites par S. Mallat en 1989, s'adaptent à la taille des structures des images. La base comprend des fonctions de supports variés : de fonctions à large support pour les grandes tendances à

des fonctions à support très petits pour des détails très précis en passant par les situations intermédiaires. Elle est orthonormée et la technique de codage par transformée s'applique.

Cependant pour des images géométriques, ni les bases issues de Fourier ni les ondelettes ne sont optimales.

2 Méthodes de compression par transformée

Dans ces méthodes, l'image de dimension $N \times N$ est subdivisée en sous images ou blocs de taille réduite (la quantité de calcul demandée pour effectuer la transformation sur l'image entière est très élevée). Chaque bloc subit une transformation mathématique orthogonale inversible linéaire du domaine spatial vers le domaine fréquentiel, indépendamment des autres blocs. Les coefficients obtenus sont alors quantifiés et codés. Parmi les transformations linéaires existantes :

- Transformation de Karhunen-Loeve (TKL)
- Transformation de Fourier discrète (TFD)
- Transformation de Hadamard (TH)
- Transformation par ondelettes (de Haar (THA))
- Transformation en cosinus discrète (TCD)

L'hypothèse sous-jacente à une décomposition en blocs est l'indépendance statistique entre les différents blocs. Cette hypothèse est fautive. Ainsi, un pixel donné est considéré comme indépendant des pixels du bloc voisin qui lui sont proches, tandis qu'il est considéré comme corrélé avec tous les pixels du bloc auquel il appartient y compris ceux qui en sont le plus éloignés.

3 Codage JPEG (Joint Photographic Expert Group)

La **compression JPEG** est une **compression avec pertes**. Elle a un des meilleurs taux de compression (20 :1 à 25 :1) sans perte notable de qualité. La compression JPEG est plus efficace sur les images photographiques (comportant de nombreux pixels de couleurs différentes) et moins sur des images géométriques (à la différence de la compression LZW) car sur ces dernières les différences de nuances dues à la compression sont très visibles. Il existe une forme de compression JPEG sans perte utilisée dans l'imagerie médicale ou spatiale, mais elle est moins efficace (approximativement 2 :1).

3.1 Numérisation des couleurs

Le trichromatisme de l'œil humain

Trois types de récepteurs sur la rétine :

- A : max à 535 nm (dans le vert)
- B : max à 570 nm (dans le jaune/rouge)

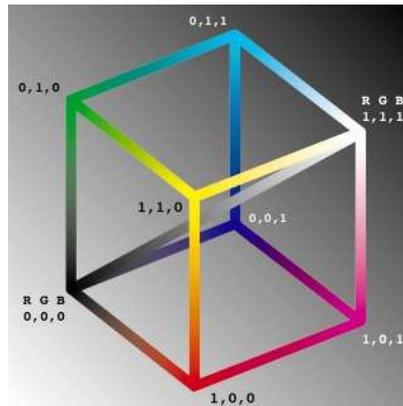


FIG. 1 – Cube RGB des couleurs

- C : max à 445 nm (dans le violet/bleu)

Du point de vue sensibilité, le type A est dominant (30 fois celle du type C). Le cerveau fait la synthèse des couleurs à partir (du degré) de l'excitation des récepteurs.

Le trichromatisme des pixels d'un écran.

Une image informatique peut être assimilée à un tableau de pixels, organisé en lignes et colonnes, dont chaque élément a une valeur. Un pixel est codé suivant la qualité de l'image :

- image en noir et blanc (image binaire) : un seul bit suffit pour coder le point (0 pour noir, 1 pour blanc)
- image en 256 nuances de gris : chaque point est représenté par un octet (8 bits)
- image en couleur : toutes les couleurs peuvent être exprimées comme des combinaisons linéaires de trois couleurs de base, par exemple Rouge(R), Vert(V), Bleu(B), avec $0 \leq R, V, B \leq 255$ (RGB en anglais)

Il y a d'autres systèmes de couleurs : YUV ou YCbCr.

Ces systèmes exploitent le fait que le cerveau traduit le signal trichromatique perçu par l'oeil, comme un signal composé de trois composantes, dont l'une est achromatique : la luminance. Elle permet d'éclaircir ou d'assombrir une couleur en ajustant la quantité de noir.

La méthode JPEG profite des imperfections de la perception d'une image par l'oeil humain pour compresser sans dégradation apparente. Le système YCbCr facilite les traitements à effectuer sur une image fixe en tenant compte que l'oeil humain :

- est plus sensible à l'intensité (luminance) qu'à la couleur. On peut sous-échantillonner les composantes couleur (i.e. chrominance) avant leur compression.

- perçoit mieux les contrastes sur les faibles que sur les fortes intensités, il est peu sensible aux variations en haute fréquence. On peut donc ”quantifier” les variations de couleurs.(Quantifier un signal consiste à réduire sa précision en le discrétisant.)

3.2 Les transformations affines dans l’espace de couleurs : RVB YCbCr

La transformation affine $RGB \mapsto YCbCr$:

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

L’ajout de 128 à Cb et Cr permet d’obtenir des valeurs entre 0 et 255. Les valeurs Cb et Cr s’appellent les valeurs de chrominance.

La valeur $Y = 0.299 \cdot R + 0.587 \cdot V + 0.114 \cdot B$ de la luminance était employée par les moniteurs en noir et blanc pour représenter une couleur de RVB. Autrement dit : Le passage d’une image en couleurs vers sa version en noir et blanc correspond à poser simplement $Cb=Cr=0$.

La transformation affine retour $YCbCr \mapsto RGB$:

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

Quelle est la source discrète ?

Les données de la luminance, chrominance composant l’image : une série statistique bivariée (les deux dimensions correspondent à l’axe horizontal et à l’axe vertical)

Pour les images naturelles, le signal composant l’image présente une forte corrélation spatiale entre pixels proches de l’image, dans les zones lisses de l’image, et dans les textures. L’information est essentiellement contenue dans les zones de ”rupture” statistique (contours, détails non répétitifs).

3.3 Echantillonnage

La norme de JPEG tient compte du fait que le système visuel humain est moins sensible aux composantes chromatiques Cb et Cr qu’à la luminance, en sous-échantillonnant horizontalement et verticalement les composantes chromatiques avant leur compression. La luminance est prise en chaque pixel tandis que la chrominance est prise comme une valeur moyenne pour un bloc de pixels. Plusieurs formats ont ainsi été définis :

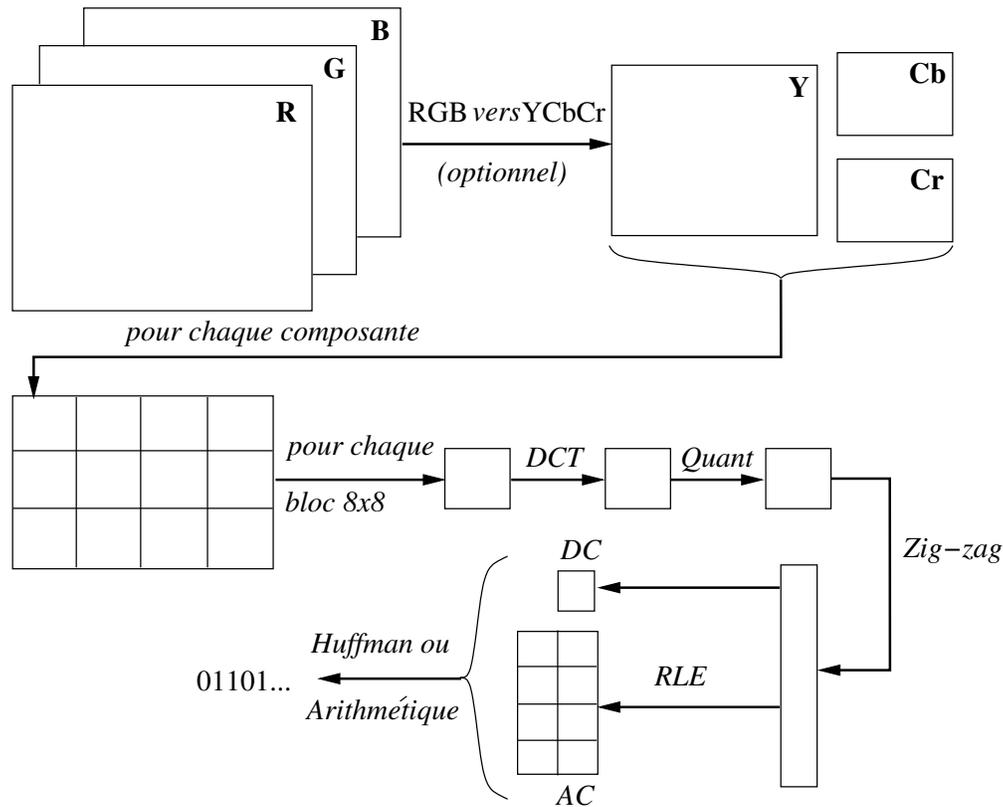


FIG. 2 – Etapes JPEG par composante

- Le format 4 : 4 : 4 est le format de base où les composantes chromatiques n'ont pas été sous-échantillonnées.
- Le format 4 : 2 : 2 où les composantes Cb et Cr ont été échantillonnées d'un facteur deux verticalement.
- Le format 4 : 2 : 0 qui est le plus commun où les composantes Cb et Cr ont été échantillonnées d'un facteur deux verticalement et horizontalement.
- Le format 4 : 0 : 0 ne contient aucune information chromatique. Il s'agit donc d'une image en niveaux de gris, seule la luminance Y est conservée.

Le format 4 : 2 : 0 Avant sous-échantillonnage, pour chaque pixel on avait 3 informations (Y, Cb et Cr); pour un bloc de 4 pixels, on avait donc 12 informations différentes. Après le sous-échantillonnage, nous avons par bloc de 4 pixels 2 informations Cb et Cr et 4 informations Y.

3.4 Processus de codage JPEG

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

1. L'image en base RVB 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 pres de l'entropie : celui-ci consiste à parcourir les blocs de l'image en zigzag puis à utiliser une compression RLE et un codage de Huffman (voir un codage arithmétique).

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

3.5 DCT : Transformée en Cosinus Discrète (DCT Discrete Cosinus transform)

Interprétation : La transformée prend en entrée un signal (constitué dans le cas du J.P.E.G. de 64 pixels) fonction des dimensions spatiales x et y et le transforme pour décrire chaque bloc en une carte de fréquences et en amplitudes plutôt qu'en pixels et couleurs. La DCT décompose ce signal en 64 signaux. Chaque signal contient une fréquence spatiale. La DCT donne l'amplitude de ces signaux. 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 à chaque changement de couleur.

Dans l'image transformée la redondance entre les pixels voisins est réduite, voir éliminée. La transformée produit des coefficients non-corrélés qui peuvent être encodés indépendamment. Elle est à la base de JPEG, MPEG-1, MPEG-2 (pour TV et video), H.263 (video-telephones).

La Transformation en Cosinus Discrète unidimensionnelle DCT-II

La transformation en Cosinus Discrète unidimensionnelle (DCT-II) est :

$$z_k = \frac{q(k)}{2} \sum_{n=0}^{N-1} x_n \cos \left[\frac{(2n+1)\pi k}{2N} \right] \quad (3.1)$$

où

$$q(k) = \begin{cases} \sqrt{2} & (k = 0) \\ 1 & (k > 0) \end{cases}$$

Le nombre d'opérations demandés par l'application de la DCT-II augmente non linéairement avec la taille de la matrice. Donc, on découpe l'image en matrices de taille plus petite (8 × 8).

Si la dimension de la composante n'est pas divisible par 8, l'encodeur remplit les lignes en recopiant la dernière ligne de la composante.

En réécrivant la transformée DCT-II sous forme matricielle pour $N = 8$ nous avons :

$$X = \begin{pmatrix} x_{00} & x_{01} & \dots & x_{07} \\ x_{10} & x_{11} & \dots & x_{17} \\ \vdots & \vdots & & \vdots \\ x_{70} & x_{71} & \dots & x_{77} \end{pmatrix} \longrightarrow Z = \begin{pmatrix} z_{00} & z_{01} & \dots & z_{07} \\ z_{10} & z_{11} & \dots & z_{17} \\ \vdots & \vdots & & \vdots \\ z_{70} & z_{71} & \dots & z_{77} \end{pmatrix}$$

ou équivalent :

$$\begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ z_7 \end{pmatrix} = C_8 \cdot \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_7 \end{pmatrix} \text{ ou } C_8 \in \mathbb{R}^{8 \times 8}$$

$$C_8 = \frac{1}{2} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos \frac{1}{16} \pi & \cos \frac{3}{16} \pi & \cos \frac{5}{16} \pi & \cos \frac{7}{16} \pi & \cos \frac{9}{16} \pi & \cos \frac{11}{16} \pi & \cos \frac{13}{16} \pi & \cos \frac{15}{16} \pi \\ \cos \frac{2}{16} \pi & \cos \frac{6}{16} \pi & \cos \frac{10}{16} \pi & \cos \frac{14}{16} \pi & \cos \frac{18}{16} \pi & \cos \frac{22}{16} \pi & \cos \frac{26}{16} \pi & \cos \frac{30}{16} \pi \\ \cos \frac{3}{16} \pi & \cos \frac{9}{16} \pi & \cos \frac{15}{16} \pi & \cos \frac{21}{16} \pi & \cos \frac{27}{16} \pi & \cos \frac{33}{16} \pi & \cos \frac{39}{16} \pi & \cos \frac{45}{16} \pi \\ \cos \frac{4}{16} \pi & \cos \frac{12}{16} \pi & \cos \frac{20}{16} \pi & \cos \frac{28}{16} \pi & \cos \frac{36}{16} \pi & \cos \frac{44}{16} \pi & \cos \frac{52}{16} \pi & \cos \frac{60}{16} \pi \\ \cos \frac{5}{16} \pi & \cos \frac{15}{16} \pi & \cos \frac{25}{16} \pi & \cos \frac{35}{16} \pi & \cos \frac{45}{16} \pi & \cos \frac{55}{16} \pi & \cos \frac{65}{16} \pi & \cos \frac{75}{16} \pi \\ \cos \frac{6}{16} \pi & \cos \frac{18}{16} \pi & \cos \frac{30}{16} \pi & \cos \frac{42}{16} \pi & \cos \frac{54}{16} \pi & \cos \frac{66}{16} \pi & \cos \frac{78}{16} \pi & \cos \frac{90}{16} \pi \\ \cos \frac{7}{16} \pi & \cos \frac{21}{16} \pi & \cos \frac{35}{16} \pi & \cos \frac{49}{16} \pi & \cos \frac{63}{16} \pi & \cos \frac{77}{16} \pi & \cos \frac{91}{16} \pi & \cos \frac{105}{16} \pi \end{pmatrix}$$

Remarque. Pour les applications on simplifie la forme de la matrice C_8 en utilisant

$$\cos(\pi - x) = -\cos(x).$$

$$C_8 = \frac{1}{2} \cdot \begin{pmatrix} c_4 & c_4 \\ c_1 & c_3 & c_5 & c_7 & -c_7 & -c_5 & -c_3 & -c_1 \\ c_2 & c_6 & -c_6 & -c_2 & -c_2 & -c_6 & c_6 & c_2 \\ c_3 & -c_7 & -c_1 & -c_5 & c_5 & c_1 & c_7 & -c_3 \\ c_4 & -c_4 & -c_4 & c_4 & c_4 & -c_4 & -c_4 & c_4 \\ c_5 & -c_1 & c_7 & c_3 & -c_3 & -c_7 & c_1 & -c_5 \\ c_6 & -c_2 & c_2 & -c_6 & -c_6 & c_2 & -c_2 & c_6 \\ c_7 & -c_5 & c_3 & -c_1 & c_1 & -c_3 & c_5 & -c_7 \end{pmatrix}$$

où

$$\begin{pmatrix} c_1 = \cos \frac{\pi}{16} = 0.9808 & c_2 = \cos \frac{\pi}{8} = 0.9239 & c_3 = \cos \frac{3\pi}{16} = 0.8315 & c_4 = \cos \frac{\pi}{4} = 0.7071 \\ c_5 = \cos \frac{5\pi}{16} = 0.5556 & c_6 = \cos \frac{3\pi}{8} = 0.3827 & c_7 = \cos \frac{7\pi}{16} = 0.1951 \end{pmatrix}$$

Propriétés caractéristiques de la DCT-II unidimensionnelle

- la DCT-II est réversible : $(C_8)^{-1} = C_8^t$. (parce que elle est une transformée orthogonale).
- la DCT-II conserve la norme Euclidéenne des matrices (l'énergie du signal) $\sum x_{ij}^2 = \sum z_{kr}^2$ (parce que elle est une transformée orthogonale).
- (Pour JPEG et MPEG) on réécrit la transformée sous forme matricielle en factorisant la décomposition. Le nombre de multiplications scalaires nécessaires est ainsi réduit. L'algorithme de Chen :

$$\begin{pmatrix} z_0 \\ z_2 \\ z_4 \\ z_6 \end{pmatrix} = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ c_2 & c_6 & -c_6 & -c_2 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_6 & -c_2 & c_2 & -c_6 \end{bmatrix} \cdot \begin{pmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{pmatrix}$$

$$\begin{pmatrix} z_1 \\ z_3 \\ z_5 \\ z_7 \end{pmatrix} = \begin{bmatrix} c_1 & c_3 & c_5 & c_7 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_5 & -c_1 & c_7 & c_3 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} \cdot \begin{pmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{pmatrix}$$

Lors de cette étape, la seule perte vient des erreurs d'arrondi due à l'utilisation de l'ordinateur. Des encodeurs/décodeurs de JPEG peuvent être trouvés chez www.ijg.org.

La Transformée en Cosinus Discrète bidimensionnelle

[Pratt, 1978] [Gonzalez and Wintz, 1977]

La transformée bi-dimensionnelle 2d DCT-II d'une image ou d'une matrice est séparable, étant simplement la DCT-II unidimensionnelle, d'au-dessus, appliquée le long des lignes et après le long des colonnes (ou inversement). C'est-à-dire, la 2d DCT-II est donné sous forme matricielle par la formule (omettant la normalisation) :

La transformée

$$T : Y \mapsto X = C_8 \cdot Y \cdot C_8^t$$

$$X = \begin{pmatrix} x_{00} & x_{01} & \dots & x_{07} \\ x_{10} & x_{11} & \dots & x_{17} \\ \vdots & \vdots & & \vdots \\ x_{70} & x_{71} & \dots & x_{77} \end{pmatrix} \longrightarrow Y = \begin{pmatrix} y_{00} & y_{01} & \dots & y_{07} \\ y_{10} & y_{11} & \dots & y_{17} \\ \vdots & \vdots & & \vdots \\ y_{70} & y_{71} & \dots & y_{77} \end{pmatrix}$$

Où équivalent pour $N = 8$ nous avons :

$$\mathbf{y}(i, j) = \frac{1}{4} q(i) q(j) \sum_{k=0}^7 \sum_{r=0}^7 \mathbf{x}(k, r) \cos \left[\frac{(2k+1)i\pi}{16} \right] \cos \left[\frac{(2r+1)j\pi}{16} \right] \quad (3.2)$$

où

$$q(k) = \begin{cases} \sqrt{2} & (k = 0) \\ 1 & (k > 0) \end{cases}$$

Propriétés caractéristiques de la DCT bidimensionnelle

- la 2d DCT-II est réversible : $T^{-1} : Y \mapsto X = C_8^t \cdot Y \cdot C_8$. (parce que C_8 est une transformée orthogonale et $(C_8)^{-1} = C_8^t$).
- la DCT-II conserve la norme Euclidienne des matrices (l'énergie du signal) $\sum_{i=0}^7 \sum_{j=0}^7 \mathbf{x}_{ij}^2 = \sum_{k=0}^7 \sum_{r=0}^7 \mathbf{y}_{kr}^2$ (parce que elle est une transformée orthogonale).
- $y_{00} = 8m$ où $m = \frac{1}{64} \sum_{i=0}^7 \sum_{j=0}^7 \mathbf{x}_{ij}$ **la moyenne** des 8×8 coefficients de \mathbf{X}

Exercice. Une unité de données constante sera diagonalisée (au sens de l'algèbre linéaire) :

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \mapsto \begin{pmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

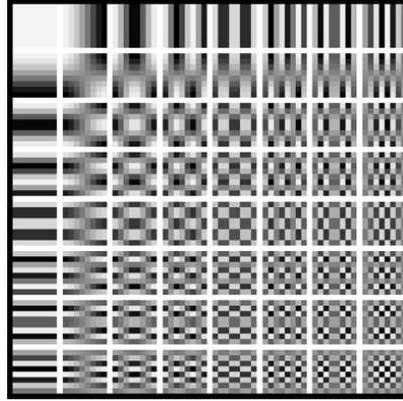


FIG. 3 – La base de cosinus locaux

Consequences : AC-DC

Les entrées de la matrice X sont corrélées. Considérons la décomposition du X suivante :

$$\begin{pmatrix} x_{00} & x_{01} & \dots & x_{07} \\ x_{10} & x_{11} & \dots & x_{17} \\ \vdots & \vdots & & \vdots \\ x_{70} & x_{71} & \dots & x_{77} \end{pmatrix} = \begin{pmatrix} m & m & \dots & m \\ m & m & \dots & m \\ \vdots & \vdots & & \vdots \\ m & m & \dots & m \end{pmatrix} + \begin{pmatrix} \delta_{00} & \delta_{01} & \dots & \delta_{07} \\ \delta_{10} & \delta_{11} & \dots & \delta_{17} \\ \vdots & \vdots & & \vdots \\ \delta_{70} & \delta_{71} & \dots & \delta_{77} \end{pmatrix}$$

où m est la moyenne des x_{ij} , et où les δ_{ij} sont les déviations de cette moyenne. Après la transformation $\mathbf{X} \rightarrow \mathbf{Y}$ on obtient la décomposition :

$$\begin{pmatrix} y_{00} & y_{01} & \dots & y_{07} \\ y_{10} & y_{11} & \dots & y_{17} \\ \vdots & \vdots & & \vdots \\ y_{70} & y_{71} & \dots & y_{77} \end{pmatrix} = \begin{pmatrix} 8m & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} + \begin{pmatrix} 0 & y_{01} & \dots & y_{07} \\ y_{10} & y_{11} & \dots & y_{17} \\ \vdots & \vdots & & \vdots \\ y_{70} & y_{71} & \dots & y_{77} \end{pmatrix}$$

Le coefficient $y_{00} = 8m$ du schéma transformé ne dépend que de la moyenne des valeurs de gris. Il est appelé coefficient **DC** (*direct current*). Les 63 autres coefficients ne dépendent que des déviations de cette moyenne. Ils s'appellent coefficients **AC** (*alternative current*). Graphiquement la composante DC nous donne la valeur moyenne des niveaux du bloc, et les composantes AC représentent les amplitudes des fréquences spatiales horizontales et verticales dans le bloc.

Consequences : Petits coefficients AC

Parce que la *DCT* préserve la moyenne quadratique, c.a.d. la norme Euclidéenne des matrices (l'énergie du signal), une image (un signal) "numériquement tranquille" va produire de petits coefficients AC.

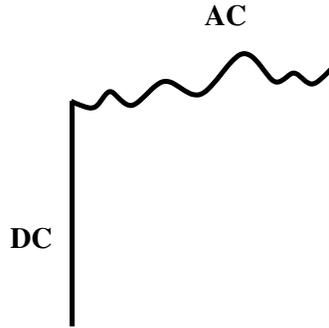


FIG. 4 – Graphe des signaux AC et DC

Dans la matrice, après application de la DCT, les basses fréquences se trouvent en haut à gauche et les hautes fréquences en bas à droite. Les hautes fréquences représentent les zones à forts contrastes dans l'image, i.e. changements rapides d'intensité des pixels. Une image classique admet une grande continuité entre les valeurs des pixels. Généralement les coefficients AC numériquement importants se trouvent dans le carré 4×4 **autour du coefficient DC**. C'est le carré des "basses fréquences spatiales" (les valeurs gris ne montrent pas d'oscillations rapides). Ainsi on représente l'intégralité de l'information de l'image sur très peu de coefficients, correspondant à des fréquences plutôt basses.

Exemple. *Considérons le schéma transformé d'un exemple de luminance d'un dégradé :*

$$\begin{pmatrix} 30 & 30 & 30 & 30 & 30 & 30 & 30 & 30 \\ 60 & 60 & 60 & 60 & 60 & 60 & 60 & 60 \\ 90 & 90 & 90 & 90 & 90 & 90 & 90 & 90 \\ 120 & 120 & 120 & 120 & 120 & 120 & 120 & 120 \\ 150 & 150 & 150 & 150 & 150 & 150 & 150 & 150 \\ 180 & 180 & 180 & 180 & 180 & 180 & 180 & 180 \\ 210 & 210 & 210 & 210 & 210 & 210 & 210 & 210 \\ 240 & 240 & 240 & 240 & 240 & 240 & 240 & 240 \end{pmatrix} \mapsto \begin{pmatrix} 1080 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -546.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -57.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Exemple. *Considérons le schéma transformé d'un exemple "à haute fréquence spatiale" :*

$$\begin{pmatrix} 0 & 250 & 0 & 250 & 0 & 250 & 0 & 250 \\ 250 & 0 & 250 & 0 & 250 & 0 & 250 & 0 \\ 0 & 250 & 0 & 250 & 0 & 250 & 0 & 250 \\ 250 & 0 & 250 & 0 & 250 & 0 & 250 & 0 \\ 0 & 250 & 0 & 250 & 0 & 250 & 0 & 250 \\ 250 & 0 & 250 & 0 & 250 & 0 & 250 & 0 \\ 0 & 250 & 0 & 250 & 0 & 250 & 0 & 250 \\ 250 & 0 & 250 & 0 & 250 & 0 & 250 & 0 \end{pmatrix} \mapsto \begin{pmatrix} 1000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -32.5 & 0 & -38.3 & 0 & -57.4 & 0 & -163.3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -38.3 & 0 & -45.2 & 0 & -67.6 & 0 & -192.6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -57.4 & 0 & -67.6 & 0 & -101.2 & 0 & -288.3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -163.3 & 0 & -192.6 & 0 & -288.3 & 0 & -821.1 \end{pmatrix}$$

Exemple.

$$\begin{pmatrix} 43 & 44 & 46 & 51 & 57 & 65 & 72 & 76 \\ 44 & 45 & 46 & 50 & 56 & 63 & 69 & 72 \\ 47 & 47 & 46 & 49 & 55 & 60 & 64 & 65 \\ 53 & 53 & 52 & 52 & 54 & 57 & 59 & 59 \\ 62 & 62 & 59 & 58 & 58 & 58 & 57 & 55 \\ 75 & 73 & 71 & 68 & 67 & 66 & 62 & 60 \\ 85 & 85 & 83 & 80 & 77 & 74 & 72 & 70 \\ 92 & 91 & 91 & 87 & 85 & 83 & 81 & 80 \end{pmatrix} \mapsto \begin{pmatrix} 512.3 & -15.2 & 6.6 & 2.9 & -1.8 & -0.5 & -0.3 & 0 \\ -79.3 & -51.8 & 7.5 & 0.6 & 0.6 & 0 & -1 & -0.6 \\ 45 & -10.5 & 0.8 & -2.5 & 0.7 & 0.4 & 0.6 & 0.3 \\ -6.6 & 5.3 & -1.1 & 0.2 & 0.1 & 0.1 & 0.2 & -0.6 \\ 0.8 & -0.7 & 0.3 & -1.1 & -0.3 & 0.3 & 0.1 & 0 \\ 0 & 0.4 & 0.1 & -0.7 & 0.1 & 0.2 & -0.3 & 0 \\ 0.3 & -0.8 & 0.1 & 1.5 & 0.3 & -0.1 & 0.8 & 0.2 \\ -1.2 & 0.1 & -0.3 & 0.3 & 0.1 & -0.2 & -1.1 & -0.1 \end{pmatrix}$$

L'reversibilité de la DCT bidimensionnelle

C_8 est une matrice **orthogonale** : la matrice inverse de C_8 est la matrice transposée : $C_8^{-1} = C_8^t$. Donc, on a pour la transformation inverse :

$$T^{-1} : Y \mapsto X = C_8^t \cdot Y \cdot C_8$$

Si

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_7 \end{pmatrix} = C_8 \cdot \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_7 \end{pmatrix} \cdot C_8^t,$$

alors

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_7 \end{pmatrix} = C_8^t \cdot \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_7 \end{pmatrix} \cdot C_8.$$

3.6 Quantification—quantifie les variations de couleurs

Quantifier un signal consiste à réduire sa précision en le discrétisant en quantités discontinues ou quanta. Un signal prenant valeur dans un alphabet de n valeurs sera représenté par un nombre inférieur m de valeurs. Le signal ainsi quantifié pourra être décrit avec moins de bits que la version initiale. Il existe donc, de fait, une erreur de quantification : on introduit des pertes, ce qui rend ainsi le processus irréversible. On distingue plusieurs types de quantification : la quantification scalaire, vectorielle et la quantification en treillis.

Pour chaque composante (luminance, chrominance) la quantification consiste à diviser la matrice obtenue après avoir appliqué la DCT, par une autre, appelée matrice de quantification, et qui contient 8×8 coefficients. Cette division atténue les hautes fréquences auxquelles l'oeil humain est très peu sensible. Les coefficients leur correspondant sont souvent ramenés à 0.

La situation picturale la plus fréquente correspond à une matrice transformée de la forme

$$\begin{pmatrix} A & B & B & C & C & D & D & D \\ B & B & C & C & D & D & D & D \\ B & C & C & D & D & D & D & D \\ C & C & D & D & D & D & D & D \\ C & D & D & D & D & D & D & D \\ D & D & D & D & D & D & D & D \\ D & D & D & D & D & D & D & D \\ D & D & D & D & D & D & D & D \end{pmatrix}$$

où A, B, C, D désignent, en ordre décroissant, le poids numérique de la position. Un codage de Huffman performant exigera une invasion de zéros sur les positions D. Pour ne pas perdre l'information qui décrit des ruptures picturales, "l'annulation" des coefficients négligeables se fait par passage aux coefficients quantifiés :

$$\tilde{y}_{jk} = \left\lfloor \frac{y_{jk}}{q_{jk}} \right\rfloor$$

où la table de quantification Q_{50} en luminance est :

$$\begin{pmatrix} q_{00} & q_{01} & \dots & q_{07} \\ q_{10} & q_{11} & \dots & q_{17} \\ \vdots & \vdots & & \vdots \\ q_{70} & q_{71} & \dots & q_{77} \end{pmatrix} = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

la table de quantification Q_{50} en chrominance est

$$\begin{pmatrix} q_{00} & q_{01} & \dots & q_{07} \\ q_{10} & q_{11} & \dots & q_{17} \\ \vdots & \vdots & & \vdots \\ q_{70} & q_{71} & \dots & q_{77} \end{pmatrix} = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

Exercice. Soit les coefficients d'un dégradé :

$$\begin{pmatrix} 30 & 30 & 30 & 30 & 30 & 30 & 30 & 30 \\ 60 & 60 & 60 & 60 & 60 & 60 & 60 & 60 \\ 90 & 90 & 90 & 90 & 90 & 90 & 90 & 90 \\ 120 & 120 & 120 & 120 & 120 & 120 & 120 & 120 \\ 150 & 150 & 150 & 150 & 150 & 150 & 150 & 150 \\ 180 & 180 & 180 & 180 & 180 & 180 & 180 & 180 \\ 210 & 210 & 210 & 210 & 210 & 210 & 210 & 210 \\ 240 & 240 & 240 & 240 & 240 & 240 & 240 & 240 \end{pmatrix} \mapsto \begin{pmatrix} 1080 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -546.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -57.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Trouver le schéma quantifié correspondant.

$$\begin{pmatrix} 68 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -46 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \leftarrow \begin{pmatrix} 68 \text{ est l'arrondi de } \frac{1080}{16} \\ -46 \text{ est l'arrondi de } \frac{-546.6}{12} \\ -4 \text{ est l'arrondi de } \frac{-57.1}{17} \\ -1 \text{ est l'arrondi de } \frac{-17}{24} \\ 0 \text{ est l'arrondi de } \frac{-4.3}{72} \end{pmatrix}$$

3.7 Reordonnement en zig-zag des 64 coefficients de DCT

La matrice 8×8 , Y obtenue après avoir appliqué la transformée D.C.T. à X est traversé en zigzag pour construire un vecteur avec 64 coefficients (0..63). Ordre de codage défini par la norme JPEG.

La première valeur dans le vecteur (à index 0) correspond à la plus basse fréquence spatiale de l'image. Elle s'appelle DC *courant continu* ("direct current" en anglais). En incrémentant l'index dans le vecteur, nous obtenons des valeurs correspondant à des fréquences plus élevées. Ces 63 coefficients s'appellent AC (*courant alternatif*) qui sont - en général - sensiblement plus petits en module que le coefficient dominant DC. Le schéma de parcours zigzag est :

<i>DC</i>	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

Le codage de Huffman concerne **la suite des coefficients non-nuls** rencontrés au parcours zigzag du schéma quantifié. Évidemment, les blocs séparateurs de zéros doivent être pris en compte.

Remarque. *Il y a donc deux symboles particuliers à coder :*

1. *(EOB) \equiv end of block qui indique la fin des coefficients non-nuls dans la suite des 63 coefficients à coder.*
2. *(ZRL) \equiv zero run list qui code un bloc de 16 entiers 0.*

La table pour le codage des valeurs **luminance** (transformées et quantifiées) a été établie empiriquement par le groupe JPEG (*Joint Photographic Experts Group*), à la base d'un grand nombre de tests avec des images à précision 8 bits. Elle n'est pas vraiment une table par défaut, mais communément acceptée en tant que telle.

0 – tête/cat.	mot code	0 – tête/cat.	mot code
0/0(<i>EOB</i>)	1010	3/9	111111110010100
0/1	00	3/ <i>a</i>	111111110010101
0/2	01	4/1	111011
0/3	100	4/2	111111000
0/4	1011	4/3	111111110010110
0/5	11010	4/4	111111110010111
0/6	1111000	4/5	111111110011000
0/7	11111000	4/6	111111110011001
0/8	1111110110	4/7	111111110011010
0/9	111111110000010	4/8	111111110011011
0/ <i>a</i>	111111110000011	4/9	111111110011100
1/1	1100	4/ <i>a</i>	111111110011101
1/2	11011	5/1	1111010
1/3	1111001	5/2	1111110111
1/4	111110110	5/3	111111110011110
1/5	11111110110	5/4	111111110011111
1/6	111111110000100	5/5	111111110100000
1/7	111111110000101	5/6	111111110100001
1/8	111111110000110	5/7	111111110100010
1/9	111111110000111	5/8	111111110100011
1/ <i>a</i>	111111110001000	5/9	111111110100100
2/1	11100	5/ <i>a</i>	111111110100101
2/2	11111001	6/1	1111011
2/3	1111110111	6/2	111111110110
2/4	111111110100	6/3	111111110100110
2/5	111111110001001	6/4	111111110100111
2/6	111111110001010	6/5	111111110101000
2/7	111111110001011	6/6	111111110101001
2/8	111111110001100	6/7	111111110101010
2/9	111111110001101	6/8	111111110101011
2/ <i>a</i>	111111110001110	6/9	111111110101100
3/1	111010	6/ <i>a</i>	111111110101101
3/2	111110111	7/1	11111010
3/3	111111110101	7/2	111111110111
3/4	111111110001111	7/3	111111110101110
3/5	111111110010000	7/4	111111110101111
3/6	111111110010001	7/5	111111110110000
3/7	111111110010010	7/6	111111110110001
3/8	111111110010011	7/7	111111110110010

0 – tête/cat.	mot code	0 – tête/cat.	mot code
7/8	111111110110011	<i>b</i> /7	111111111010101
7/9	111111110110100	<i>b</i> /8	111111111010110
7/ <i>a</i>	111111110110101	<i>b</i> /9	111111111010111
8/1	111111000	<i>b</i> / <i>a</i>	111111111011000
8/2	11111111000000	<i>c</i> /1	1111111010
8/3	111111110110110	<i>c</i> /2	111111111011001
8/4	111111110110111	<i>c</i> /3	111111111011010
8/5	111111110111000	<i>c</i> /4	111111111011011
8/6	111111110111001	<i>c</i> /5	111111111011100
8/7	111111110111010	<i>c</i> /6	111111111011101
8/8	111111110111011	<i>c</i> /7	111111111011110
8/9	111111110111100	<i>c</i> /8	111111111011111
8/ <i>a</i>	111111110111101	<i>c</i> /9	111111111100000
9/1	111111001	<i>c</i> / <i>a</i>	111111111100001
9/2	111111110111110	<i>d</i> /1	11111111000
9/3	111111110111111	<i>d</i> /2	111111111100010
9/4	111111111000000	<i>d</i> /3	111111111100011
9/5	111111111000001	<i>d</i> /4	111111111100100
9/6	111111111000010	<i>d</i> /5	111111111100101
9/7	111111111000011	<i>d</i> /6	111111111100110
9/8	111111111000100	<i>d</i> /7	111111111100111
9/9	111111111000101	<i>d</i> /8	111111111101000
9/ <i>a</i>	111111111000110	<i>d</i> /9	111111111101001
<i>a</i> /1	111111010	<i>d</i> / <i>a</i>	111111111101010
<i>a</i> /2	111111111000111	<i>e</i> /1	111111111101011
<i>a</i> /3	111111111001000	<i>e</i> /2	111111111101100
<i>a</i> /4	111111111001001	<i>e</i> /3	111111111101101
<i>a</i> /5	111111111001010	<i>e</i> /4	111111111101110
<i>a</i> /6	111111111001011	<i>e</i> /5	111111111101111
<i>a</i> /7	111111111001100	<i>e</i> /6	111111111110000
<i>a</i> /8	111111111001101	<i>e</i> /7	111111111110001
<i>a</i> /9	111111111001110	<i>e</i> /8	111111111110010
<i>a</i> / <i>a</i>	111111111001111	<i>e</i> /9	111111111110011
<i>b</i> /1	1111111001	<i>e</i> / <i>a</i>	111111111110100
<i>b</i> /2	111111111010000	<i>f</i> /0(<i>ZRL</i>)	11111111001
<i>b</i> /3	111111111010001	<i>f</i> /1	111111111110101
<i>b</i> /4	111111111010010	<i>f</i> /2	111111111110110
<i>b</i> /5	111111111010011	<i>f</i> /3	111111111110111
<i>b</i> /6	111111111010100	<i>f</i> /4	111111111111000

Le code du $n^{\circ}3$ dans la catégorie 3, i.e. codage sur 3 bits est : le complément à 2 de 100 : 011

Le code du $n^{\circ}0$ dans la catégorie 1, i.e. codage sur 1 bit est : le complément à 2 de 1 : 0

Le mot code est "1111111110000100 010001 1111111110100110 011 111111010 0 1010". Les espaces dans l'écriture sont pour aider le lecteur. Le taux de compression : à 512 bits = 64 octets binaires (les 8 fois 8 valeurs de luminance) un mot code de longueur 55. Donc le taux de compression est : $t = 1 - \frac{55}{512} = 0.89$.

La déquantification : la perte d'information de la décompression

Le mot code permet de reconstruire la matrice des valeurs quantifiées (le coefficient DC quantifié vient d'ailleurs) par multiplication avec le schéma des diviseurs utilisés pour la quantification. Dans notre cas, on obtiendra la matrice déquantifiée

$$\begin{pmatrix} 68 \cdot 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -46 \cdot 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4 \cdot 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 \cdot 24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 \cdot 72 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1088 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -552 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -56 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

La retransformation de la matrice des valeurs déquantifiées, suivi d'un arrondi (on était parti d'octets binaires) termine la décompression. On obtient :

$$\begin{pmatrix} 30 & 30 & 30 & 30 & 30 & 30 & 30 & 30 \\ 61 & 61 & 61 & 61 & 61 & 61 & 61 & 61 \\ 91 & 91 & 91 & 91 & 91 & 91 & 91 & 91 \\ 119 & 119 & 119 & 119 & 119 & 119 & 119 & 119 \\ 153 & 153 & 153 & 153 & 153 & 153 & 153 & 153 \\ 181 & 181 & 181 & 181 & 181 & 181 & 181 & 181 \\ 211 & 211 & 211 & 211 & 211 & 211 & 211 & 211 \\ 242 & 242 & 242 & 242 & 242 & 242 & 242 & 242 \end{pmatrix}$$

Exercice. : Soit trois matrices 8×8 de valeurs de luminance, ainsi que les matrices des valeurs transformées et quantifiées. Déterminer dans chaque cas le codage de Huffman associé et calculer le taux de compression.

1.

$$\begin{pmatrix} 159 & 152 & 142 & 134 & 133 & 140 & 149 & 155 \\ 176 & 170 & 162 & 156 & 157 & 163 & 171 & 177 \\ 132 & 129 & 123 & 120 & 121 & 126 & 132 & 136 \\ 72 & 71 & 69 & 68 & 69 & 70 & 72 & 74 \\ 69 & 70 & 72 & 73 & 73 & 71 & 69 & 67 \\ 123 & 126 & 131 & 134 & 133 & 129 & 123 & 119 \\ 157 & 163 & 171 & 177 & 176 & 170 & 162 & 156 \\ 132 & 139 & 149 & 157 & 158 & 151 & 142 & 135 \end{pmatrix} \mapsto \begin{pmatrix} 64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2.

$$\begin{pmatrix} 83 & 89 & 91 & 84 & 73 & 68 & 75 & 83 \\ 96 & 98 & 96 & 86 & 78 & 82 & 99 & 114 \\ 82 & 85 & 84 & 75 & 67 & 69 & 83 & 97 \\ 88 & 99 & 108 & 107 & 99 & 94 & 97 & 104 \\ 88 & 101 & 115 & 118 & 112 & 108 & 111 & 116 \\ 95 & 103 & 110 & 107 & 99 & 97 & 105 & 114 \\ 122 & 130 & 136 & 131 & 120 & 114 & 117 & 124 \\ 96 & 111 & 127 & 131 & 121 & 109 & 105 & 105 \end{pmatrix} \mapsto \begin{pmatrix} 50 & 0 & 0 & -3 & 0 & 0 & 0 & 0 \\ -9 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

3.

$$\begin{pmatrix} 65 & 61 & 68 & 59 & 69 & 60 & 67 & 63 \\ 61 & 72 & 52 & 78 & 50 & 76 & 56 & 67 \\ 68 & 52 & 81 & 44 & 84 & 47 & 76 & 60 \\ 59 & 78 & 44 & 88 & 40 & 84 & 50 & 69 \\ 69 & 50 & 84 & 40 & 88 & 44 & 78 & 59 \\ 60 & 76 & 47 & 84 & 44 & 81 & 52 & 68 \\ 67 & 56 & 76 & 50 & 78 & 52 & 72 & 61 \\ 63 & 67 & 60 & 69 & 59 & 68 & 61 & 65 \end{pmatrix} \mapsto \begin{pmatrix} 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$