

## Chapitre III : Éléments de théorie de l'information (entropie), MICD, compression de type JPEG/JPEG2000, psycho visuel.

### 1- L'entropie

La théorie de l'information permet d'évaluer la quantité d'information dans une image. Pour cela, chaque point d'une image est considéré comme une variable aléatoire. Une image est alors considérée comme une suite de  $W \cdot H$  variables aléatoires.

L'entropie est une grandeur qui caractérise la quantité d'informations que contient une image. Par exemple une image dont tous les pixels ont la même valeur contient très peu d'informations car elle est extrêmement redondante, son entropie est faible. En revanche, une image dont tous les pixels ont une valeur aléatoire contient beaucoup d'informations; son entropie est forte.

En pratique, l'entropie d'une image numérique est inversement liée à la probabilité d'apparition des niveaux de gris dans l'image. Plus une valeur de gris  $k$  est rare, plus sa probabilité d'apparition  $p(k)$  est faible, et cela contribue à une entropie globale plus grande.

Par définition, l'entropie d'ordre zéro  $H_0$  est donnée par :

$$H_0 = - \sum_{k=0}^{2^R-1} p(k) \cdot \log_2 p(k) \quad \text{unité bpp}$$

Avec:  $p(k)$  est la probabilité d'apparition des niveaux de gris dans l'image,  $k$  est la valeur de gris et  $R$  est le nombre de bits par pixel.  $k$  désigne un point d'une image en niveaux de gris, c'est une variable aléatoire dont les valeurs sont des entiers de l'intervalle  $[0 \ 255]$  ( $2^{R-1}$ ) où 255 est la valeur maximale de niveau de gris. L'entropie est exprimée en bits par pixel et le logarithme est en base 2 car un bit peut avoir 2 valeurs différentes.

L'utilisation du logarithme de base 2, fait de  $H_0$  le nombre moyen de bits par pixel nécessaire pour coder toute l'information contenue dans l'image. Une image codée avec  $R$  bits par pixels a en fait presque toujours une entropie d'ordre zéro  $H_0$  inférieure à  $R$ .

Par conséquent, l'entropie  $H_0$  d'une image originale fournit le débit minimal qu'il est possible d'atteindre par compression pixel par pixel sans dégrader l'image, et par là même un taux de compression sans perte maximal.

Shannon définit également l'entropie d'un message, d'une manière comparable à la notion utilisée en thermodynamique. Dans cette discipline, il s'agit d'une grandeur physique qui permet d'évaluer la dégradation de l'énergie d'un système ; nous disons souvent que l'entropie mesure le degré de désordre d'un système. Dans la théorie de l'information, l'entropie d'un message indique le poids d'information qu'il porte, et donc sa " compressibilité " théorique.

### 2- Le codeur MICD

Le codeur MICD (en Anglais DPCM pour *Differential Pulse Code Modulation*) est un système simple, appartenant à la famille des codeurs prédictifs. Il en existe deux versions :

MICD : MIC Différentiel,

MICDA : MICD Adaptatif (celui auquel nous nous intéressons dépend du contenu des données image).

Les diagrammes en blocs du codeur et du décodeur sont présentés sur la figure 3.1, avec les notations suivantes :

$X(i, j)$  : signal d'entrée.

$X_p(i, j)$  : prédiction du signal d'entrée.

$E(i, j)$  : erreur de prédiction.

$\hat{E}(i, j)$  : erreur de prédiction quantifiée.

$\hat{X}(i, j)$  : signal codé-décodé (signal reconstruit qui sert à la prédiction de l'étape suivante).

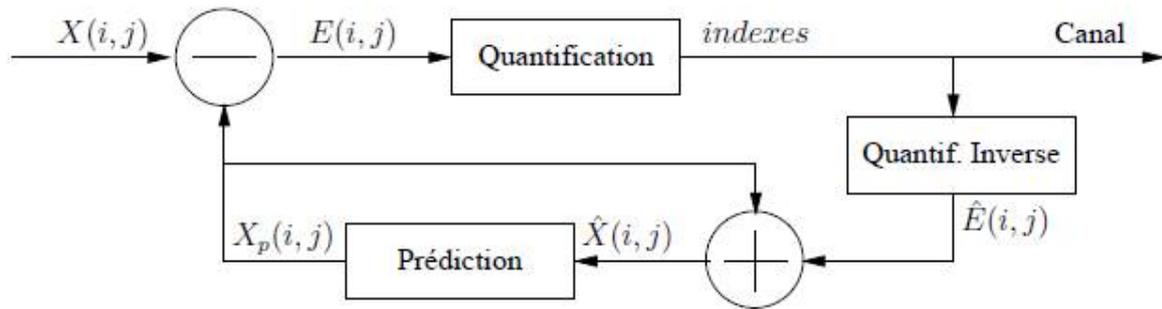
Un exemple que nous pouvons citer est le codage des signaux de télévision à traiter ou transmettre sous forme numérique est effectué en utilisant la modulation d'impulsions codées différentielles (MICD).

La motivation de ces codeurs appliqués en imagerie vient du fait que les pixels voisins (spatialement) ont une intensité lumineuse voisine (numériquement). Dans le schéma de codage, il s'agit de quantifier la différence  $E(i, j)$  entre le signal  $X(i, j)$  et une prédiction  $X_p(i, j)$  de ce signal. Cette prédiction est réalisée à partir des valeurs passées de  $\hat{E}$  et de  $X_p$ . Si nous notons  $Q$  le bruit de quantification, nous avons :

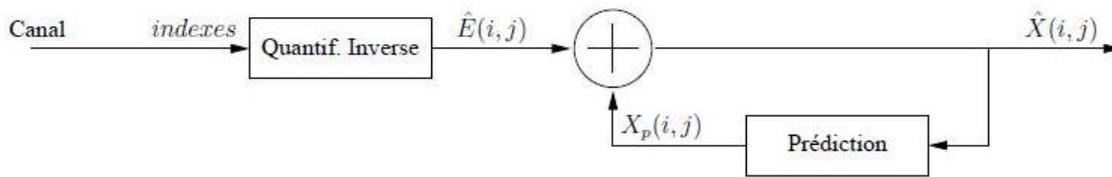
$$\hat{E}(i, j) = E(i, j) + Q(i, j)$$

En suivant la chaîne du codage, il vient :

$$\begin{aligned}\hat{X}(i, j) &= X_p(i, j) + \hat{E}(i, j) \\ &= X_p(i, j) + E(i, j) + Q(i, j) \\ &= X(i, j) + Q(i, j)\end{aligned}$$



(a) Schéma du codeur



(b) Schéma du décodeur

Figure 3.1 : Diagrammes du schéma de codage / décodage

Concernant la prédiction, Les pixels sont des valeurs d'intensité lumineuse et sont ici comprises entre 0 et 255. Pour les deux codeurs (MICD et MICDA), les pixels voisins du pixel à coder (noté X) sont notés comme suit :

Ligne précédente	B	C	D
Ligne courante	A	X	

Nous allons étudier deux prédictions fixes pour le MICD :

$P(X) = A$  (mono-dimensionnelle).

$P(X) = A+C / 2$  (bi-dimensionnelle).

### 3- Compression de type JPEG/JPEG2000, psycho visuel

#### 3-1 Pourquoi vouloir compresser une image ?

De nos jours, la puissance des processeurs augmente plus vite que les capacités de stockage, et énormément plus vite que la bande passante des réseaux, car cela demande d'énormes changements dans les infrastructures de télécommunications. Ainsi, pour pallier ce manque, il est courant de réduire la taille des données en exploitant la puissance des processeurs plutôt qu'en augmentant les capacités de stockage et de transmission des données.

La compression d'images peut s'employer avec des contraintes et des attentes très différentes, selon l'utilisation à laquelle les images sont destinées. Nous pouvons vouloir réduire le nombre de bits d'une image avec une contrainte sur la capacité de stockage, la vitesse de transmission, le temps d'accès depuis un médium de stockage, etc... Nous pouvons prévoir d'utiliser l'image décompressée pour des applications de loisir, du traitement d'image, de la recherche visuelle rapide dans une base d'images, du diagnostic médical, etc...

Nous donnons quelques raisons de l'utilisation de la technologie de la numérisation en compression comme solution aux problèmes de stockage et de transmission de l'image :

- Une image au standard VGA+ (640 x 480, 256 couleurs) occupe un octet par pixel soit 300 ko. Dans cette définition, qui reste limitée par rapport aux standards audiovisuels, 25 images (soit une seconde de vidéo) occuperaient plus de 7 Mo, sans parler du son. Il est aisé de comprendre que ces valeurs sont incompatibles avec la plupart des supports informatiques actuels, et même avec les contraintes des réseaux publics ou privés envisageables à moyen terme.
- L'image consomme une quantité impressionnante d'octets quand elle est numérisée. Aujourd'hui, nous parlons de " qualité megapixel " pour les appareils photo numériques ; cela signifie que chaque image comporte environ un million de pixels dont chaque pixel nécessite trois octets pour les composantes RVB (rouge, vert, bleu). Donc, sans compression, cela représenterait un peu plus de 3 Mo pour une seule photographie.
- Une image de 800x800 pixels en 16 millions de couleurs (24 bits par pixel), correspondant à un fond d'écran, occupe 1 million 400 000 octets.
- La transmission d'une séquence d'images couleur au format QCIF échantillonnée à 30 Hertz représente un débit de 9.12 M bits par seconde, et de 36.40 M bits par seconde au format CIF (le format de QCIF [Quart de CIF]).
- La transmission d'une séquence vidéo 512x512, 8 bpp, 3 couleurs sur une ligne téléphonique avec modem à 9600 bauds (Le baud est une unité de mesure utilisée dans le domaine des télécommunications en général, et dans le domaine informatique en particulier, notamment dans le contexte de communications avec certains périphériques externes par exemple via un modem), nécessite 11 minutes par image.
- Une image de télévision basse résolution contient 3 composantes ou couleurs, 512x512 pixels par couleur et 8 bits par pixel, soit un total de  $6 \times 10^6$  bits.
- Les encyclopédies les plus complètes comportent environ 25 000 pages contenant du texte et des images. Nous pouvons considérer chaque page comme une image qui, numérisée à 300 dpi, occupe un volume de 1 Mo. Au total, la numérisation de l'ouvrage conduit à 25 Go. Il est clair que ce volume est impossible à stocker. Il faut donc effectuer une "compression" des données afin d'en réduire le volume.
- Dans d'autres domaines professionnels tels que l'imagerie médicale, des masses gigantesques de données sont acquises chaque jour. Nous chiffrons à environ 10 Teraoctets la masse de données d'un service radiologique d'un hôpital dans un pays industrialisé.

Pour remédier à ces contraintes, il n'y a qu'une solution : comprimer les images. Les chercheurs ont imaginé de nombreuses méthodes de compression, que nous les classons en deux catégories : celles qui se contentent de comprimer les données sans les altérer, et celles qui les compactent en les modifiant. Les premières, dites non destructives, permettent de reconstituer, au bit près, le fichier dans l'état où il était avant la compression.

Pour réduire le volume global des images tout en conservant l'image originale, un moyen consiste en la compression des images avec le minimum de dégradation et le maximum d'efficacité possible.

### **3.2 Définition de la compression d'image**

A cet effet, la compression et le codage consiste en la réduction de la taille physique d'un bloc d'information en réduisant le nombre de bits par pixel à stocker ou à transmettre suite à l'exploitation de la redondance informationnelle dans l'image. Nous parlons alors de réduction de redondance. La redondance peut être occasionnée par une représentation non efficace des données; son élimination dans ce cas, n'empêche pas la restitution de ces données.

Trois sortes de redondances sont exploitées dans la compression d'images :

- La redondance spatiale entre pixels ou blocs voisins dans l'image ;
- La redondance temporelle entre images successives dans une séquence vidéo ;
- La redondance spectrale entre plan de couleur ou bandes spectrales.

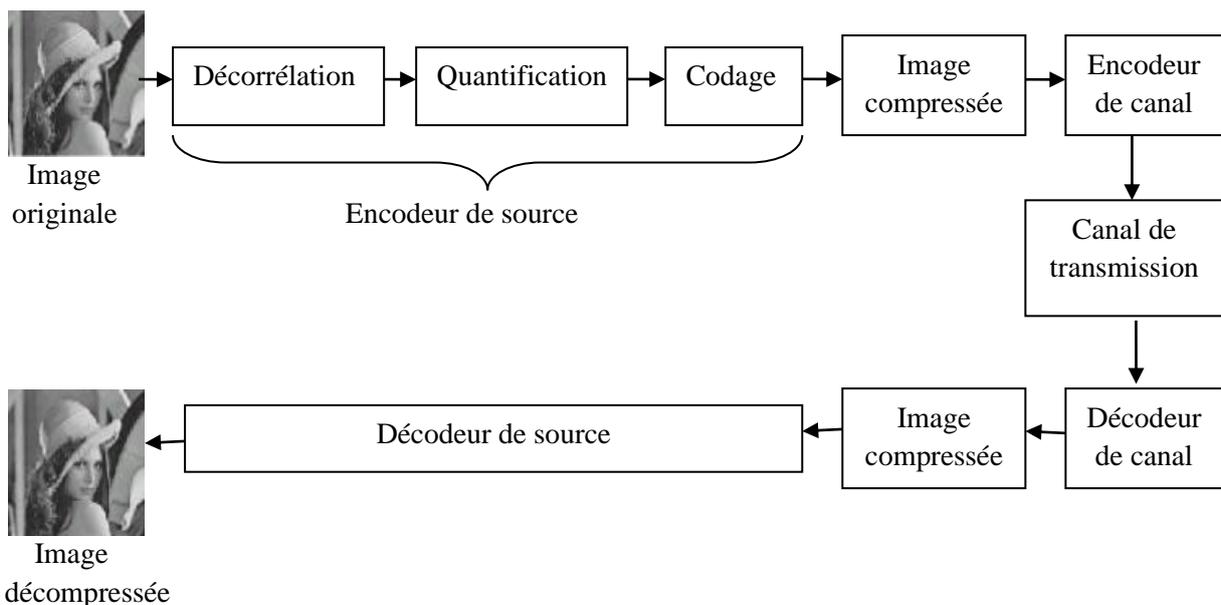
En fait, l'ensemble des techniques de compression des images exploite la corrélation entre des pixels voisins pour produire une " valeur probable " du prochain pixel à coder. Le produit de cette prédiction étant statistiquement proche de la valeur réelle, il est alors possible d'effectuer le codage de manière plus économique, en codant uniquement la différence avec la valeur prédite. Les ordres de grandeur des valeurs à transmettre sont alors statistiquement plus faibles, ce qui permet d'utiliser un nombre plus faible de bits.

### 3.3 La redondance psychovisuelle

L'œil humain perçoit une image de manière particulière; il sait reconnaître des détails qui portent plus d'information que d'autres. Par suite, nous pouvons éliminer les détails qui ont très peu d'importance sans dégrader de manière significative la qualité de l'image; cette opération est une quantification (opération irréversible). La codification IGS (Improved Gray Scale) est une méthode de codage permettant de prendre en compte les préoccupations développées ci-dessus. Supposons que l'on ait une image originale à 256 niveaux de gris. Nous pouvons réduire le volume de l'image en codant sur 16 niveaux; malheureusement ces méthodes appliquées brutalement font apparaître des effets quelquefois inesthétiques ou purement artificiels (passages brusques d'un niveau de gris à un autre, faux contours notamment). La codification IGS est un moyen d'adoucir le procédé. Le principe est le suivant : le codage à 256 niveaux s'effectue sur 8 bits séparables en poids forts (4 bits) et poids faibles (4 bits); pour des pixels successifs, le codage IGS à 16 niveaux (donc sur 4 bits) s'obtient en tenant compte du codage sur 8 bits des pixels précédents et en ne retenant que les bits de poids forts.

### 3.4 Système de compression

La figure ci-dessous précise la chaîne d'opérations relatives à la compression/décompression d'images.



**Figure 3.2 :** Schéma d'un système de compression

Un système de compression comporte un encodeur à une extrémité du processus (source) et un décodeur à l'autre extrémité du processus (utilisation). Entre le codage de l'information source et le décodage pour utilisation existe un canal qui est la cause principale de l'opération de compression; par exemple, le canal pourra être un réseau de transmission de données.

### **3.4.1 L'encodeur de source :**

L'objectif de l'encodeur de source est de représenter le message avec le moins de bits possibles.

Pour ce faire, il cherche à éliminer toute la redondance contenue dans le message de la source. Nous avons mentionné auparavant que l'entropie d'une image désigne la quantité moyenne d'informations, le but de la réduction de l'entropie est par conséquent de transformer la distribution des valeurs de l'information à transmettre. L'encodeur de source (ou la réduction de l'entropie) est composé de trois étapes : la décorrélation, la quantification et le codage entropique.

Nous distinguons les étapes suivantes pour l'encodage de source :

- **Décorrélation.** la dépendance existante entre chacun des pixels et ses voisins (la luminosité varie très peu d'un pixel à un pixel voisin) traduit une corrélation très forte sur l'image. Nous essayons donc de tirer partie de cette corrélation, pour réduire le volume d'information en effectuant une opération de décorrélation des pixels ou réduction des redondances interpixels. La décorrélation consiste à transformer les pixels initiaux en un ensemble de coefficients moins corrélés, c'est une opération réversible.

- **Quantification.** la quantification des coefficients a pour but de réduire le nombre de bits nécessaires pour leurs représentations. Elle représente une étape clé de la compression visant la réduction des redondances psychovisuelles.

- **Codage.** une fois les coefficients quantifiés, ils sont codés. Un codeur doit satisfaire à priori les conditions suivantes :

- Unicité : deux messages différents ne doivent pas être codés de la même façon,
- Déchiffrabilité : deux mots de codes successifs doivent être distingués sans ambiguïté.

Nous distinguons 2 types de codage de source:

1- Codage sans perte (« entropique »)

Ex: Huffman, Lempel-Ziv, Arithmétique, ...

=> Taux de compression faible (1.5 à 2).

2- Codage avec pertes (« irréversible ») :

Suppression des redondances ou de l'information inutile car prévisible et invisible par le système visuel humain (SVH).

=> Taux de compression élevés (>10).

Ex: image transmise en la balayant (« Zigzag scan »).

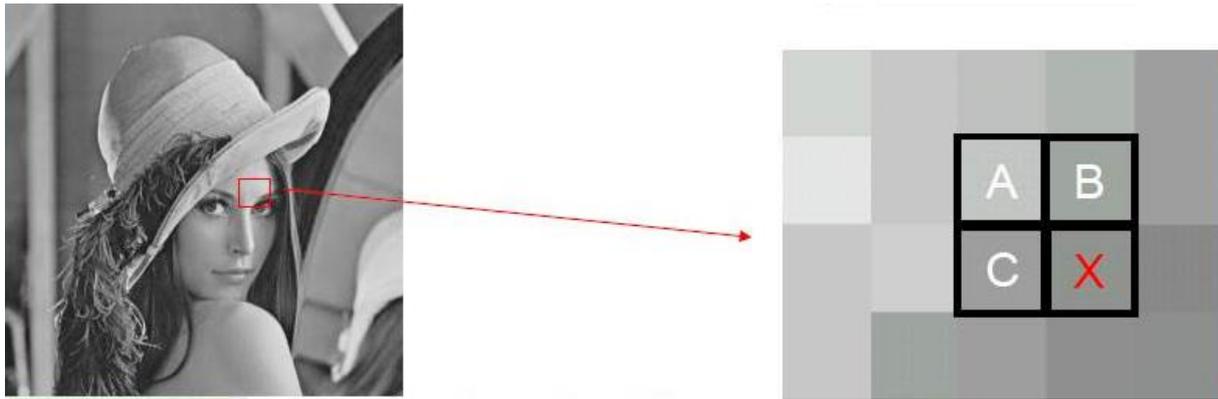


Figure 3.3 : Principe du codage MICD

Nous utilisons le codage MICD (Modulation d'Impulsions Codées Différentielles): le niveau de gris d'un pixel dépend souvent de celui de ses voisins.

=> prédire X en utilisant A, B ou C.

Notons que de ces trois opérations, seule la quantification est une opération irréversible (puisque des données sont abandonnées).

### 3.4.2 L'encodeur de canal :

Le rôle du codage de canal est de protéger le message des perturbations du canal de transmission en ajoutant de la redondance au message compressé. En effet, l'encodeur de canal est nécessaire quand le canal est " bruiteux "; donc; les données sont codées dans un format permettant de réduire au maximum les erreurs dues au passage de l'information dans le canal, c'est à dire d'accroître l'immunité au bruit. Pour les canaux du type réseau de transmission de données, les codes de Hamming sont les plus connus et les plus utilisés. Ils permettent de détecter les erreurs et de les auto-corriger.

### 3.4.3 Le décodeur de canal :

Nous le trouvons à l'autre extrémité de la chaîne, il effectue l'opération inverse de l'encodeur de canal.

### 3.4.4 Le décodeur de source :

Le décodeur de source se compose de deux étages :

- 1- Le décodage qui rétablit les redondances de codage.
- 2- La décorrélation inverse qui rétablit les redondances interpixel.

La quantification inverse évidemment n'a pas de sens, puisque l'opération directe est irréversible.

Les systèmes de compression utilisés réellement ne comportent pas nécessairement tous les blocs fonctionnels de la figure précédente. Nous pouvons d'ailleurs distinguer 2 catégories de systèmes de compression :

- Si  $I'(x,y) = I(x,y)$ , le système de compression est idéal ; il correspond à la compression sans pertes appelée quelquefois compactage.
- Si  $I'(x,y) <> I(x,y)$ , le système de compression admet une distorsion; il correspond à la compression avec pertes.

### 3.5 La compression JPEG

Les méthodes par transformation figurent parmi les techniques de compression les plus employées. Elles permettent d'obtenir des taux de compression élevés tout en conservant une bonne qualité d'image. Ce sont des méthodes qui font appel successivement à plusieurs principes de compression. JPEG est un ensemble de spécifications et algorithmes de compression des images fixes. Leur statut de norme officielle constitue une évolution essentielle.

JPEG est originellement prévu pour des images en 16 millions de couleurs (24 bits par pixel). Tous les logiciels de traitement d'images exploitent actuellement le format JPEG, en importation comme en exportation. Il en va de même des appareils photos numériques. Des processeurs spécialisés sont également disponibles sur le marché.

Le principe de la compression par transformation est de décomposer les pixels fortement corrélés de l'image en ensembles de coefficients spectraux partiellement décorrélés, dont l'énergie est concentrée dans un nombre restreint de coefficients. Ce compactage de l'énergie permet d'affecter en priorité aux coefficients spectraux les plus énergétiques un nombre de bits plus élevé qu'aux autres. Les méthodes par transformation suivent le schéma de fonctionnement présenté dans la figure suivante :

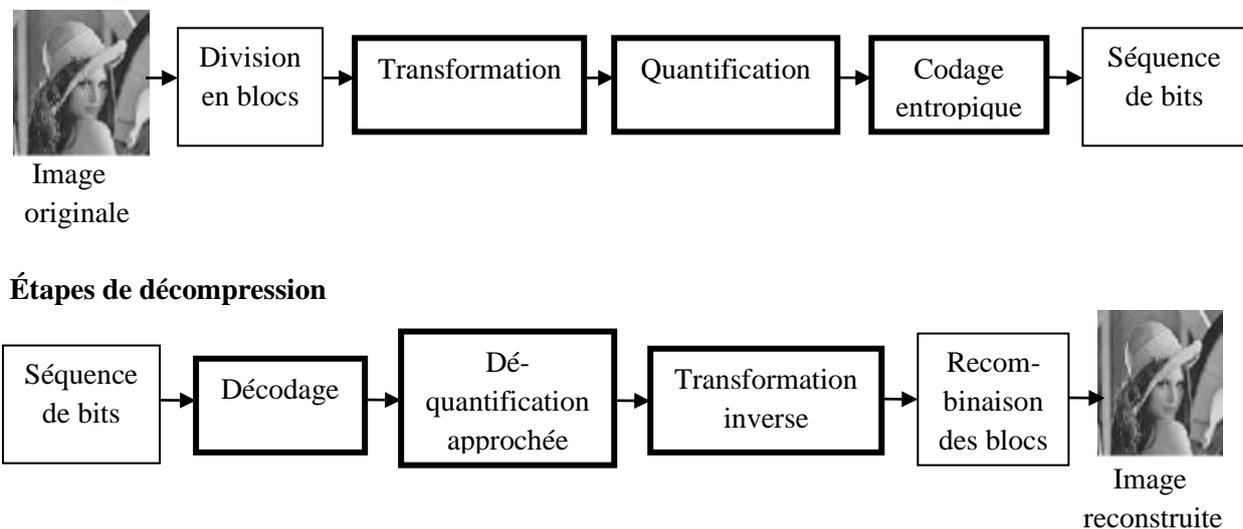


Figure 3.4 : Principe général de la compression/décompression JPEG.

#### 3.5.1 Division en blocs

La première étape consiste à diviser l'image en blocs sur lesquels vont s'appliquer indépendamment les étapes suivantes. La principale raison de ce découpage est de limiter le nombre de pixels à traiter à la fois pour diminuer les temps de calcul et la complexité des circuits électroniques. Il peut résulter de cette division un effet visuel appelé effet de « blocs » : à des taux élevés la frontière des blocs devient visible car ils ont été comprimés indépendamment. La taille des blocs est variable selon les méthodes. Elle est de 8x8 pour JPEG et MPEG.

#### 3.5.2 Transformation

La deuxième étape consiste à appliquer une transformation mathématique à chaque bloc. Les transformations utilisées en compression d'image sont des transformations orthogonales, leur but est de décorréler les pixels, ce qui a pour effet en général de redistribuer l'énergie de l'image dans un

nombre restreint des coefficients transformés. De cette façon, un grand nombre de coefficients transformés ont des très faibles valeurs, et peuvent être supprimés ou se voir allouer un nombre très faible de bits lors de l'étape suivante de quantification. La transformation fait passer d'un espace de nombres entiers, les pixels, à un espace de nombres flottants (voire de complexes) qui sont les coefficients du plan des fréquences, aussi appelés coefficients spectraux.

Les 2 motivations principales à l'utilisation d'une transformation sont :

- L'obtention d'une représentation de l'image qui se prête bien à la quantification et au codage.
- La possibilité d'ajuster les erreurs de quantification selon la sensibilité au système visuel humain.

Les algorithmes JPEG de compression non conservative sont fondés sur les propriétés de la DCT (« Discrete Cosine Transformation » ou « Transformation Discrète en Cosinus »).

Les images couleur sont généralement codées comme trois images monochromes. L'image est tout d'abord découpée en zones de dimensions réduites (8 x 8 pixels), puis chaque pavé est traité par la DCT. La compression, selon un taux laissé au choix de l'utilisateur, est effectuée par quantification des coefficients issus de cette décomposition spectrale. Cette étape est la seule à l'origine des dégradations. Enfin, les données obtenues sont codées sous une forme elle-même compressée au moyen des algorithmes conservatifs (Huffman).

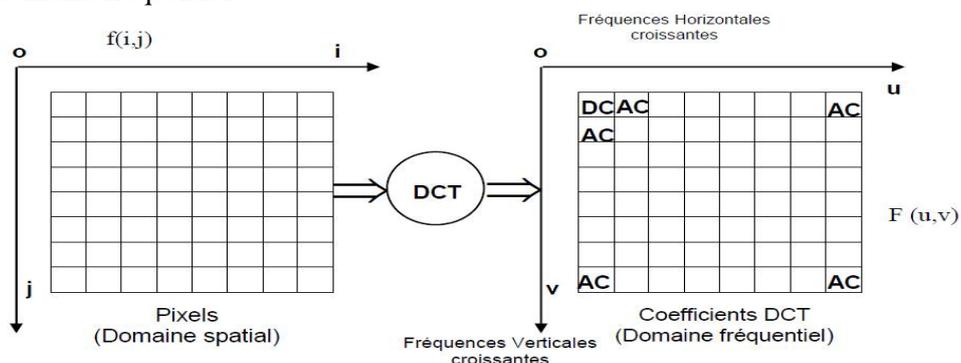
### 3.5.3 Quantification et codage

La troisième étape est la quantification des coefficients transformés, afin de se ramener à un nombre limité de valeurs différentes. La quantification est souvent précédée d'une pondération psychovisuelle des coefficients, afin de préserver ceux auxquels l'œil est le plus sensible. La quantification est la seule étape irréversible de tout le schéma de compression par transformation. Souvent, un quantificateur scalaire uniforme est employé. Un codage entropique est effectué sur les coefficients quantifiés, pour aboutir au flot binaire de sortie. En général, c'est le degré de quantification qui détermine le niveau de compression obtenu. Une quantification grossière donne une compression importante mais introduit une forte dégradation.

### 3.5.4 Illustration de l'algorithme JPEG avec perte

Nous présentons dans ce qui suit, les étapes de l'algorithme JPEG :

Étape1 : l'image est divisée en blocs de 8x8 pixels, une valeur optimale de taille de blocs pour JPEG est estimée à 8. Dans le codage JPEG, l'image est ainsi codée par blocs de 8x8 pixels. Nous effectuons alors la transformée en cosinus discrète (DCT) de chacun de ces blocs afin de passer du domaine spatial au domaine fréquentiel.



**Figure 3.5 :** Représentation de la DCT d'un bloc de  $8 \times 8$  pixels.

La DCT permet schématiquement de changer l'échelle de mesure, en passant d'une échelle définissant un pixel en fonction de sa position en x et en y à une échelle définissant la fréquence d'apparition de ce pixel dans un bloc de pixels, en effet, il est dès lors possible de supprimer des informations sans pour autant altérer le résultat final, contrairement à un bloc de pixels où la disparition brute de plusieurs éléments est immédiatement visible.

La formule de transformation DCT utilisée est :

$$X_c(k,l) = \frac{c(k,l)}{4} \sum_{m=0}^7 \sum_{n=0}^7 x(m,n) \cos \left[ \pi \frac{(2m+1)k}{16} \right] \cos \left[ \pi \frac{(2n+1)l}{16} \right]$$

Avec  $k=0, 1, \dots, 7$

$l=0, 1, \dots, 7$

Nous allons prendre le bloc suivant extrait à partir de l'image LENA :

$$x(m,n) = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix}$$

Dans ce qui suit, nous présentons la transformée du bloc, après avoir arrondi les coefficients fréquentiels à l'entier le plus proche :

$$X_c(k,l) = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

Nous voyons que la DCT a pour effet de concentrer l'information en très peu de coefficients fréquentiels correspondant aux basses fréquences, et que les autres coefficients sont de haute fréquence.

Dans la matrice, suite à l'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  $4 \times 4$  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.

Le coefficient 1260 sur la composante (0,0) représente " la moyenne " de la grandeur d'ensemble de la matrice d'entrée et 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.

Typiquement, les valeurs des pixels varient lentement de point au point adjacent à travers une image, en particulier pour des images qui ne présentent pas de discontinuités brutales. Par conséquent, les coefficients des basses fréquences, correspondant à des plages spatiales uniformes, ont des valeurs plus grandes que les coefficients des hautes fréquences. L'énergie du signal se trouve donc concentrée sur les fréquences spatiales inférieures, alors que la majorité des autres fréquences sont nulles ou presque.

Étape2 : les coefficients fréquentiels sont normalisés. Une matrice de normalisation composée d'entiers sur 8 bits est utilisée pour pondérer les coefficients. Chaque coefficient de la matrice transformée est divisé par la valeur correspondante de la matrice de normalisation. Les coefficients normalisés sont obtenus par la formule :

$$\hat{X}(k,l) = \text{int} \left[ \frac{X(k,l)}{q \cdot Q(k,l)} \right]$$

Où :

$\hat{X}(k,l)$  est le coefficient DCT normalisé.

$\text{Int}(X)$  est la fonction d'arrondi à l'entier le plus proche.

$X(k,l)$  est le coefficient DCT de position (k,l) du bloc.

q est le pas de quantification.

$Q(k,l)$  est la matrice de normalisation.

La normalisation est équivalente à une quantification scalaire. 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  $8 \times 8$  coefficients. Cette division atténue les hautes fréquences auxquelles l'œil humain est très peu sensible. Les coefficients leur correspondant sont souvent ramenés à 0.

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 : nous introduisons des pertes, ce qui rend ainsi le processus irréversible.

Le pas de quantification est une valeur entière qui sert à régler le degré de quantification du bloc. Plus le pas est grand, plus grandes sont la compression ainsi que la distorsion. Dans les applications courantes du standard, les valeurs de q utilisées en pratique sont comprises entre 1 et environ 25. Le pas peut être modifié d'un bloc à l'autre afin d'adapter localement la compression.

La matrice de normalisation sert à pondérer les coefficients DCT afin d'adapter le degré de quantification de chaque coefficient en fonction de la sensibilité du système visuel au bruit de

quantification. Ainsi, l'œil nu est peu sensible au bruit introduit dans les hautes fréquences et les valeurs de  $Q(k,l)$  correspondant aux fréquences élevées seront plus fortes que pour les basses fréquences. Prenons l'exemple d'une image couleur RGB+L, 4 matrices différentes de normalisation différentes peuvent être utilisées pour les composantes couleurs et luminance.

Une des matrices de normalisation de la luminance (respectivement quantification) fournie par le standard est :

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

Sur l'exemple que nous avons utilisé, les coefficients normalisés sont :

$$\bar{X}_c(k,l) = \begin{bmatrix} 79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

La même matrice de normalisation est appliquée à tous les blocs de l'image. Elle est transmise avec les coefficients comprimés en tant que frais généraux de codage. Le choix de la matrice de normalisation est laissé libre à l'utilisateur. Le comité JPEG a effectué un travail considérable en vue de trouver expérimentalement des matrices qui fournissent les meilleurs résultats possibles visuellement en pondérant les coefficients selon leur importance perceptuelle.

Étape3 : le coefficient DC normalisé est codé.

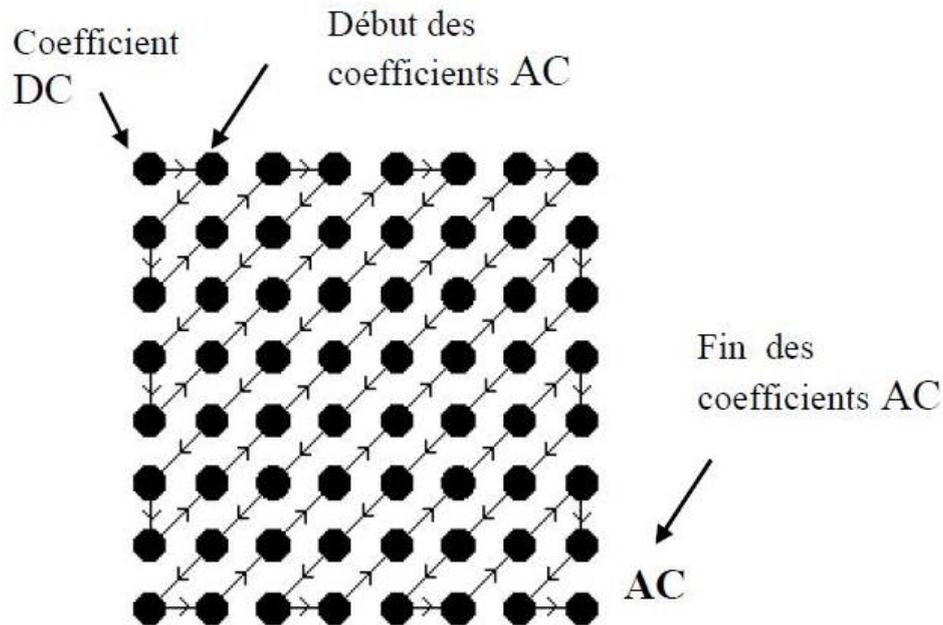
Par un système de DPCM, le coefficient DC est codé en utilisant la valeur DC du bloc précédent.

Étape4 : les coefficients AC sont réarrangés par une méthode zigzag.

L'énergie est concentrée dans les coefficients en haut à gauche de la matrice transformée. Nous réordonnons les coefficients afin d'obtenir un vecteur 1D qui range approximativement les coefficients par ordre décroissant d'énergie. Comme de nombreux coefficients de  $\bar{X}$  correspondant aux fréquences élevées sont nuls nous obtenons des grandes plages de zéros, qui se prêtent bien au codage RLC.

Au parcours ligne par ligne, nous préférons le Zig-zag scanning qui consiste à parcourir le bloc 8x8 de façon à lire en premier les coefficients forts puis de terminer par les plus faibles et d'augmenter la probabilité d'obtenir un nombre important de zéros consécutifs.

Le réarrangement zig-zag se fait comme mentionné sur la figure suivante :



**Figure 3.6 :** Réarrangement zig-zag des blocs normalisés JPEG

Dans l'exemple utilisé, le réarrangement zig-zag aboutit au vecteur suivant :

[79 0 -2 -1 -1 -1 0 0 -1 EOB]

Le symbole EOB est placé juste après la dernière valeur non nulle. Comme les blocs ont toujours 64 coefficients, il n'est pas nécessaire de préciser la longueur de la plage de zéro finissant le bloc réarrangé.

Étape5 : Le codage entropique

La suite des opérations est plus complexe : nous effectuons un codage particulier pour le coefficient le plus important (coefficient DC), puis un autre codage pour les autres coefficients (coefficients AC) :

intervalle	catégorie DC	catégorie AC
0	0	N/A
-1,1	1	1
-3,-2,2,3	2	2
-7,...,-4,4,...,7	3	3
-15,...,-8,8,...,15	4	4
-31,...,-16,16,...,31	5	5
-63,...,-32,32,...,63	6	6
-127,...,-64,64,...,127	7	7
-255,...,-128,128,...,255	8	8
-511,...,-256,256,...,511	9	9
-1023,...,-512,512,...,1023	A	A
-2047,...,-1024,1024,...,2047	B	B
-4095,...,-2048,2048,...,4095	C	C
-8191,...,-4096,4096,...,8191	D	D
-16383,...,-8192,8192,...,16383	E	E
-32767,...,-16384,16384,...,32767	F	N/A

a) table des catégories

catégorie	code de base	longueur
0	10	3
1	11	4
2	100	5
3	0	5
4	101	7
5	110	8
6	1110	10
7	11110	12
8	111110	14
9	1111110	16
A	11111110	18
B	111111110	20

b) table pour coefficients DC

1) Nous calculons la différence entre le coefficient DC : 79 et celui de la sous image précédemment traitée (par exemple 89); le résultat, -10, permet de déterminer, à partir de la table JPEG des catégories (a), la catégorie à prendre en compte (4) pour obtenir, dans la table JPEG de luminance (b), le code en binaire (101) et la longueur totale du code (7). Il faut donc compléter le code de base à 3 bits par 4 bits complémentaires; ceux-ci sont générés par la règle suivante :

- si la différence  $DC_i - DC_{i-1}$  est positive, on prend les K bits (où K est la catégorie précédemment déterminée) les moins significatifs de  $DC_i$ .
- si la différence  $DC_i - DC_{i-1}$  est négative, on prend le K bits les moins significatifs de  $DC_i$  et on retranche 1. Nous sommes ici dans le second cas :  $-11 = (0101)_2$ . On a donc finalement pour le code de DC : 1010101.

Il est utile de mentionner quelques remarques :

Un coefficient non zéro dans la lecture séquentielle d'un schéma quantifié est caractérisé par :

1. Le nombre des zéros qui le séparent de son prédécesseur non-nul : 0-tête.
2. Sa catégorie i qui donne le nombre de bits nécessaires pour le codage du coefficient.
3. Son numéro dans la catégorie (les numéros commencent par zéro).

Pour effectuer le codage de Huffman de la lecture séquentielle des coefficients quantifiés, il nous manque seulement une table pour les mots code des symboles du type 0-tête/catégorie.

Parce que les zéros peuvent être de chaînes de plus en plus longues, le nombre 0-tête est considéré un nombre binaire en base hexadécimale : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f.

Si le nombre 0-tête est plus grand que 16 on rajoute un symbole zero run list pour chaque bloc de 16 zéros et on compte la suite de zéros.

4. Il y a donc deux symboles particuliers à coder :

(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.

Représentation en binaire d'un nombre négatif est une représentation en complément à 2 :

- Partir de la représentation positive de ce nombre.
- Inverser les valeurs de tous les bits : les 1 deviennent des 0 et les 0 des 1.
- Ajouter 1 au résultat de l'opération précédente, ou bien rajouter un bit de "signe" avant.

En utilisant les catégories faites selon le nombre minimal de bits nécessaires au codage nous n'avons plus besoin d'utiliser le bit supplémentaire. Pourquoi ?

Le code du n°17 dans la catégorie 6, i.e. codage sur 6 bits est : le complément à 2 de 101110 : 010001

Le code du n°3 dans la catégorie 3, i.e. codage sur 3 bits est : le complément à 2 de 100 : 011

Le code du n°0 dans la catégorie 1, i.e. codage sur 1 bit est : le complément à 2 de 1 : 0

0 – tête/cat.	mot code	0 – tête/cat.	mot code
0/0( <i>EOB</i> )	1010	3/9	1111111110010100
0/1	00	3/ <i>a</i>	1111111110010101
0/2	01	4/1	111011
0/3	100	4/2	1111111000
0/4	1011	4/3	1111111110010110
0/5	11010	4/4	1111111110010111
0/6	1111000	4/5	1111111110011000
0/7	11111000	4/6	1111111110011001
0/8	1111110110	4/7	1111111110011010
0/9	1111111110000010	4/8	1111111110011011
0/ <i>a</i>	1111111110000011	4/9	1111111110011100
1/1	1100	4/ <i>a</i>	1111111110011101
1/2	11011	5/1	1111010
1/3	1111001	5/2	11111110111
1/4	111110110	5/3	1111111110011110
1/5	11111110110	5/4	1111111110011111
1/6	1111111110000100	5/5	1111111110100000
1/7	1111111110000101	5/6	1111111110100001
1/8	1111111110000110	5/7	1111111110100010
1/9	1111111110000111	5/8	1111111110100011
1/ <i>a</i>	1111111110001000	5/9	1111111110100100
2/1	11100	5/ <i>a</i>	1111111110100101
2/2	11111001	6/1	1111011
2/3	1111110111	6/2	111111110110
2/4	111111110100	6/3	1111111110100110
2/5	1111111110001001	6/4	1111111110100111
2/6	1111111110001010	6/5	1111111110101000
2/7	1111111110001011	6/6	1111111110101001
2/8	1111111110001100	6/7	1111111110101010
2/9	1111111110001101	6/8	1111111110101011
2/ <i>a</i>	1111111110001110	6/9	1111111110101100
3/1	111010	6/ <i>a</i>	1111111110101101
3/2	111110111	7/1	11111010
3/3	111111110101	7/2	111111110111
3/4	1111111110001111	7/3	1111111110101110
3/5	1111111110010000	7/4	1111111110101111
3/6	1111111110010001	7/5	1111111110110000
3/7	1111111110010010	7/6	1111111110110001
3/8	1111111110010011	7/7	1111111110110010

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

