

Chapitre 3

Les techniques de compression vidéo et audio

3.1. Introduction

Apparue au début des années 1980, la numérisation des images vidéo conformément à la norme 4:2:2 donne naissance à des fichiers gigantesques, donc à des débits extrêmement élevés au regard de la capacité relativement limitée des supports de stockage et de transmission classiques de l'époque.

Pourquoi compresser la vidéo ?

Soit une vidéo de résolution 720x576 (format PAL), dont chaque pixel est codé sur 24 bits, possédant une cadence de 25 images par seconde. Le débit brut nécessaire pour l'écriture (et la lecture) du fichier est de : $720 \times 576 \times 24 \times 25 = 237 \text{ Mbits/s}$

Pour une vidéo haute définition (HDTV), de résolution 1920x1080 : $1920 \times 1080 \times 24 \times 25 = 1.15 \text{ Gbits/s} !!!$

Ces débits ont longtemps dépassé et dépassent encore les débits des différents supports de stockage. Par exemple, un disque dur est capable de lire aujourd'hui à des débits moyens d'environ 600 Mbits/s, un lecteur de DVD d'ordinateur sait lire à un débit (très) maximum de 150 Mbits/s

La numérisation d'une vidéo analogique pose le problème du stockage et de la transmission. Ces deux problématiques ont longtemps été un frein au développement du numérique en audiovisuel.

D'où la solution par la compression numérique qui permet des économies en matière de stockage et de transmission, et ouvre la voie à des fonctionnalités totalement nouvelles dans les applications courantes (cryptage, pay per view).

Les compressions numériques éliminent les données redondantes, simplifient ou suppriment certaines informations peu importantes en fonction de l'application visée.

3.2. Calcul des débits de transmission vidéo numérique

Débit d'une vidéo au format 4:2:2

La taille d'une image au format 4:2:2 8 bits 1440 échantillons avec une fréquence de 25 images/s est :

$$1440 \times 8 \times 575 = 6,3 \text{ Mbits/image} = 6,3 \times 25 = 156 \text{ Mbits/s} = 20 \text{ Mo/s}$$

Le débit numérique d'une vidéo au format 4:2:2 est 156 Mbits/s

Ordre de grandeur :

- Pour transmettre une vidéo numérique au format 4:2:2 à plein débit il faudrait 160 lignes ADSL à 1Mbits/s
- Un CD-Rom de 650 Mo ne peut contenir que 30 secondes de vidéo au format 4:2:2/ 8 bits.
- Un espace mémoire de 1Go peut contenir 50 secondes d'images 4:2:2 sur 8 bits.

La réduction du débit du signal vidéo est un passage obligé pour autoriser le stockage d'une grande quantité d'images sur un support de stockage numérique et pour permettre la diffusion de ces images numériques dans un canal de fréquence conventionnel.

3.3. Principe de la compression numérique

Les algorithmes de compression reposent sur les deux principes suivants :

- Ne pas transmettre des informations redondantes.
- Ne pas transmettre des informations non ou peu perceptibles par l'œil ou l'oreille humaine.

Cela est d'autant plus difficile que le débit final que l'on cherche à obtenir est faible. Il existe deux grandes catégories de compression numérique :

3.3.1. Les algorithmes sans pertes (lossless)

Ils effectuent un traitement totalement transparent, permettant de retrouver intégralement les données d'origine après décompression. Dans ce cas, l'information est conservée mais le taux de compression est faible ce qui les rendent non adaptés pour les vidéos.

3.3.2. Les algorithmes «avec pertes» (lossy)

Ils imposent de négliger certaines informations. Si cela se fait dans des proportions limitées, l'élimination de ces informations peut passer inaperçues. Ils se caractérisent par un taux de compression plus élevé et pertes d'information plus ou moins visibles ou audibles.

3.3.3. Mesure de la performance d'un algorithme de compression

Le taux de compression est une mesure de la performance d'un algorithme de compression de données numériques. Il existe plusieurs définitions du taux de compression dont la plus courante est le rapport entre le volume de données non compressées et le volume de données compressées. Un taux de compression de 40:1 appliquée sur une vidéo au format 4:2:2/ 8 bits permet de descendre le débit de transmission de 156Mbits/s à 4 Mbits/s ($156 \div 40$).

Un algorithme de compression (respectivement décompression) fait appel à toute une série de mécanismes paramétrés de manière très précise. Un algorithme de compression/décompression est décrit par un **codec** (COdage / DECodage ou COmpression / DECompression ou COmpresseur / DECompresseur).

Pour lire correctement un fichier compressé il faut **absolument** le codec utilisé pour la génération de ce fichier. Il existe d'innombrables algorithmes de compression et par conséquent d'innombrables codecs tels que :

Codec d'images : PNG, JPEG, GIFF, TIFF

Codec audio : MP3, Ogg Vorbis,

Codec vidéo : MPEG -1,-2,-4

3.3.4. Quelques ordres de grandeur

Dans le cas d'une **image fixe**, on peut typiquement supprimer jusqu' à 70% de ses informations sans toucher à sa qualité. On dit qu'un taux de compression de 3:1 est totalement transparent. Si l'on accepte quelques pertes, généralement peu décelables par un œil non averti, le taux de compression peut atteindre 10:1.

Dans le cas d'une **séquence vidéo animée**, on peut obtenir des facteurs de compression nettement supérieurs en tenant compte de la forte parenté entre les images contiguës. Par exemple, pour les applications de diffusion, il est courant d'utiliser des taux de compression compris entre 15:1 et 40:1. **En studio** cependant, les exigences plus sévères en matière de qualité du signal imposent de limiter le taux de compression à **5:1**.

Dans le cas d'une **séquence audio**, on peut obtenir des facteurs de compression de l'ordre de 4:1 à 14:1 suivant la qualité du fichier numérique final souhaité.

Rappel : le débit audio qualité CD est égal à **1,3Mbits/s** donc **1 minute d'audio** en qualité **CD** c'est environ **10Mo**.

3.4. Les redondances de l'image vidéo

Une image vidéo présente quatre types de redondance :

3.4.1. La redondance spatiale

Toute plage uniforme sur une image renferme des pixels identiques. Il est donc inutile de coder séparément chacun de ces pixels puisqu'un seul peut les caractériser tous. Il suffit alors de transmettre deux données, l'une représentant la valeur du pixel, l'autre indiquant le nombre de fois qu'elle se répète. Nous verrons comment une technique comme la Transformée en Cosinus Discrète (souvent noté DCT) peut mettre en évidence cette redondance spatiale à l'intérieur de chaque image.

3.4.2. La redondance temporelle

Dans une séquence vidéo, il existe une très forte corrélation entre les images successives. Entre deux images successives, les différences peuvent être minimales. Les techniques d'estimation de mouvement permettent de coder une image par rapport à sa voisine, en ne transmettant que les informations relatives au déplacement de ses composantes. L'élimination des redondances temporelles peut conduire à des taux de compression très élevés. En revanche, le codage s'applique non plus à des images isolées, mais à des groupes d'images rendues indissociables les unes des autres. Il est donc bien adapté à la diffusion d'un flux continu.

3.4.3. La redondance subjective

L'exploitation de la redondance subjective fait appel à la notion de codage perceptuel, tirant parti des faiblesses de la vision ou de l'audition humaine. Elle consiste à coder avec un nombre réduit de bit les éléments de l'image jugés les moins significatifs.

3.4.4. La redondance statistique

Il s'agit d'une notion purement mathématique : si certains codes reviennent plus fréquemment que d'autres, autant leur réserver les mots numériques les plus courts. Cette opération, appelée codage entropique, n'entraîne aucune perte. Exemple : codage Huffman

3.5. Techniques de compression vidéo (source coding)

Comment compresser ?

- **1ère idée** : on compresse de manière indépendante chacune des images constituant le flux vidéo. C'est le format MJPEG (Motion JPEG), qui est parfois utilisé lorsqu'on a besoin de pouvoir accéder de manière aléatoire à chacune des images (montage vidéo). On utilise donc ici uniquement la redondance spatiale de l'information

- **2nde idée** : deux images qui se suivent dans une séquence vidéo sont quasiment identiques. On parle alors de redondance temporelle. Le but est de stocker uniquement ce qui est modifié d'une image à une autre.

Aujourd'hui, une séquence vidéo ne se compose pas uniquement d'une succession d'image, mais inclue très souvent également du son. Lors de la compression, il y a donc deux flux à gérer.

Le comité d'expert JPEG (Joint Photographic Expert Group) a édité un standard pour la compression des images fixes. Dans sa version avec pertes (JPEG lossy), l'algorithme JPEG offre des facteurs de compression supérieurs à 10.

Les algorithmes MPEG (Motion Pictures Expert Group) pour la compression vidéo ont été largement inspirés par le standard JPEG.

Le standard MPEG-1 (1992) vise à reproduire un programme vidéo (incluant le son stéréo) sur un support de type CD (videodisc, CD-I), c'est-à-dire à réduire le taux global d'information (audio stereo + video) à 1.5 Mbits/s maximum ! Cette réduction drastique de l'information ne peut se faire qu'au détriment de la qualité de l'image.

Le standard MPEG-2 (1994) vise une qualité meilleure, à des bits rates de 4 à 9 Mbits/s. Les applications visées sont le DVD (Digital Versatile Disc) et le DVB (Digital Video Broadcasting).

La compression vidéo nécessite deux types de compression :

- Compression intra-images : Prise en compte de la similarité dans les zones de la même image.
- Compression inter-images : Prise en compte de la similitude d'une image et celles qui l'entourent dans la séquence (précédente, suivante), on ne tiendra en compte que la différence entre elles.

3.6. La compression intra-images

Elle s'appelle aussi « **compression spatiale** » et utilise sur les images la technique propre à la norme JPEG. C'est aujourd'hui une des méthodes de compression les plus utilisées, notamment grâce aux forts taux de compression dont elle est capable. Ainsi, des taux de 10:1 à 20:1 peuvent être obtenus sans dégradations visibles de l'image. JPEG a été conçu comme une méthode de compression destinée aux **images fixes**.

Le schéma de la figure 3.1, montre les étapes du codage d'une image en utilisant le standard JPEG.

- 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) L'image en format SIF est tout d'abord discrétisée en blocs de 8x8 pixels. Dans chaque bloc de luminance (ou de chrominance), on applique la transformée en cosinus discrète à la luminance (ou aux deux signaux de chrominance). Pour chaque composante, on groupe les valeurs par blocs de 8x8 valeurs. Pour chaque bloc on applique un codage par transformée orthogonale qui est la transformée en cosinus discrète, DCT (Discrete cosine transform).

Considérons l'exemple d'un bloc de luminance (gris dégradé) et sa matrice DCT comme suit :

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

1080	0	0	0	0	0	0	0
-546.5	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

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

Bloc de luminance

Coefficients DCT

Coefficients DCT quantifiés

NB : Il est évident que, pour la plupart des blocs, tant la luminance que la chrominance seront plus ou moins constantes, ce qui se traduira par :

- des coefficients de la DCT de valeur importante, pour les ordres faibles (basses fréquences),
- des coefficients peu significatifs pour les ordres élevés (hautes fréquences).

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	93	95	98	112	100	103	99

Quantification de luminance

68 est l'arrondi de $1080 \div 16$
-46 est l'arrondi de $-546.5 \div 12$
-4 est l'arrondi de $-57.1 \div 14$
-1 est l'arrondi de $-17 \div 24$
0 est l'arrondi de $-4.3 \div 72$

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

Quantification de chrominance

- 3) La matrice obtenue par DCT est quantifiée (ce qui induit une compression destructrice de l'image).

JPEG définit des tables de quantification par défaut, dont les valeurs sont le résultat de nombreuses mesures réalisées par le comité. Chacun des 64 coefficients obtenus précédemment est divisé par un entier $Q(i,j)$ appelé coefficient de quantification, dont la valeur varie selon la fréquence considérée. La valeur obtenue est ensuite arrondie. Cette table de coefficients peut être définie par l'utilisateur, mais les standards du JPEG recommandent 2 tables pour la luminance et la chrominance.

Il est également possible de définir sa propre table de quantification, par exemple avec :

$$Q(i,j) = 1 + (i + j)R$$

4) Codage entropique

Pour chaque composante, on réalise une compression sans perte des matrices quantifiées en utilisant un codage près de l'entropie :

Un train de bits série est obtenu par lecture «zig-zag» des coefficients, comme le montre la figure 3.6.

Cette procédure 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 défini par la norme JFIF (JPEG File Interchange Format)

1	3	5	7	9	11	13	15
3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29

R=2

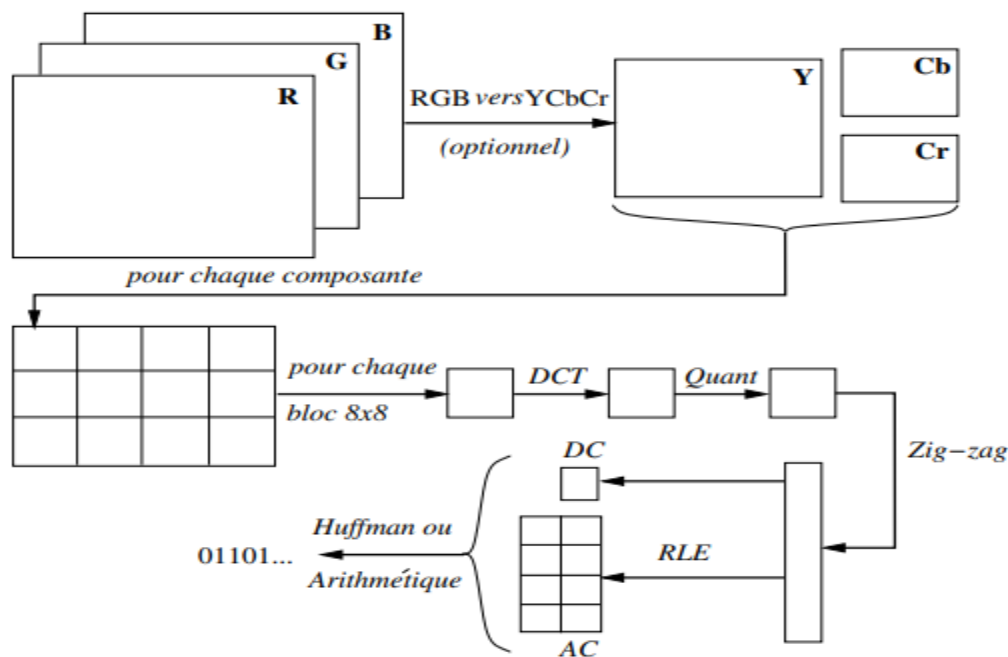


Fig. 3.2 Etapes JPEG par composante

La DCT (discrete cosine transform) fonctionne à la manière de la transformée de Fourier :

Elle s'effectue sur des blocs de 8 pixels de l'image. Si les dimensions ne sont pas multiples de 8, les bords de l'image sont répétés autant de fois que nécessaire.

$$C_{ij} = \frac{1}{4} K_i K_j \sum_{x=0}^7 \sum_{y=0}^7 p_{xy} \cos \left[\frac{(2x+1)i\pi}{16} \right] \cos \left[\frac{(2y+1)j\pi}{16} \right]$$

Remarque : Ces valeurs ne concernent pas l'exemple précédent.

Enfin, la dernière étape consiste à écrire dans le fichier les paramètres utilisés pour la compression (en-tête) et les données compressées.

Il existe un mode progressif permettant, lors du décodage, l'affichage progressif de l'image. Dans ce mode, c'est l'ordre d'enregistrement des fréquences qui est modifié de sorte qu'une lecture partielle du fichier permet déjà d'avoir une idée de l'image (avec approximation de pixels manquants).

Chacune des valeurs continues (ou leur différence) est codée selon le code de Huffman représenté dans le tableau ci-dessus :

On code donc la séquence 1118, -4, 5 par : (code_ligne | numéro colonne) (111111111110 | 01110100010) , (1110 | 011), (1110 | 101)

Soit par exemple la séquence 1118, 2, 0, -2, 0 (13 fois), -1, 0 ...

L'encodeur s'intéresse au nombre Z de zéros précédant une valeur x non nulle :

- trouver x dans le 1er tableau, et mémoriser son emplacement (R,C).
- la paire (R,Z) est utilisée comme (ligne,colonne) pour le second tableau.
- Enfin, le code de Huffman trouvé dans ce tableau est concaténé à C.

Pour 1118 **2** 0 -2 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 ... Le premier coefficient x=2 se situe sur la ligne R = 2 et C=2, et il y a Z=0 zéros avant.

0	0						0
1	-1	1					10
2	-3	-2	2	3			110
3	-7	-6	-5	-4	...	7	1110
...	
15	-32767	-32766	-32765	32767	11111...0
16	32768						11111...1

R	Z=0	Z=1	...	Z=15
0:	1010			1111111001'
1:	00'	1100	...	111111110100'
2:	01'	11011	...	111111110110'
3:	100	111001	...	111111110111'
4:	1011	11110110	...	111111111000'
...				

On s'intéresse donc au code situé en (R=2,Z=0) du second tableau : On code : 01 | 10

1118 2 0 **-2** 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 ... Le second coefficient x=-2 se situe sur la ligne R = 2 et C=1, et il y a Z=1 zéros avant.

0	0						0
1	-1	1					10
2	-3	-2	2	3			110
3	-7	-6	-5	-4	...	7	1110
...	
15	-32767	-32766	-32765	32767	11111...0
16	32768						11111...1

R	Z=0	Z=1	...	Z=15
0:	1010			1111111001'
1:	00'	1100	...	111111110100'
2:	01'	1101	...	111111110110'
3:	100	111001	...	111111110111'
4:	1011	11110110	...	111111111000'
...				

On s'intéresse donc au code situé en (R=2,Z=1) du second tableau : 11011 et on code 11011|01

1118 2 0 -2 0 0 0 0 0 0 0 0 0 0 0 **-1** 0 ... Le coefficient x=-1 se situe maintenant sur la ligne R = 1 et C=0, et il y a Z=13 zéros avant.

R	Z=0	Z=1	...	Z=15
0:	1010			1111111001'
1:	00'	1100	...	111111110100'
2:	01'	11011	...	111111110110'
3:	100	111001	...	111111110111'
4:	1011	11110110	...	111111111000'
...				

R	Z=0	Z=1	...	Z=15
0:	1010			1111111001'
1:	00'	1100	...	111111110100'
2:	01'	11011	...	111111110110'
3:	100	111001	...	111111110111'
4:	1011	11110110	...	111111111000'
...				

On s'intéresse donc au code situé en (R = 1 , Z = 1 3) du second tableau : 1110101 et on code 1110101 | 0

1118 2 0 -2 0 0 0 0 0 0 0 0 0 0 0 0 -1 0

R	Z=0	Z=1	...	Z=15
0:	1010			1111111001'
1:	00'	1100	...	111111110100'
2:	01'	1101	...	1111111110110'
3:	100	111001	...	11111111110111'
4:	1011	11110110	...	11111111111000'
...				

R	Z=0	Z=1	...	Z=15
0:	1010			1111111001'
1:	00'	1100	...	111111110100'
2:	01'	1101	...	1111111110110'
3:	100	111001	...	11111111110111'
4:	1011	11110110	...	11111111111000'
...				

Enfin, la séquence de 0 se code simplement 1010

Au final, la séquence : 1118 2 0 -2 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 ...

Se code : 1111111111101110100010 01101101110111010101010

Ces 46 bits codent une seule composante couleur d'un bloc 8x8. Imaginons que les autres plans couleur soient également codés avec un mot de 46 bits.

Dans ce cas, la compression est de : $(64*3*8)/(46*3) = 11.13$!

En fait, le taux est même meilleur puisque les composantes continues (qui occupent ici 23 des 46 bits) des autres blocs sont codées à partir des différences !

JPEG définit un "tableau 2" différent pour la luminance et pour la chrominance. Cependant, la norme laisse le choix de ce tableau, et il est possible de spécifier son propre code.

Des variantes du JPEG utilisent plutôt du codage arithmétique adaptatif, appelé codeur QM, qui ne nécessite donc pas les tableaux 1 et 2. En pratique, le gain obtenu est autour de 5 à 10%. Mais son utilisation reste rare.

3.7. La compression inter-images

La compression temporelle exploitation de la redondance temporelle entre images adjacentes dans une vidéo mais non identiques.

3.7.1. Structure vidéo pour MPEG 1/2/4

Dans une séquence vidéo, les données sont hiérarchisées d'une manière bien précise. Tout d'abord, en termes de représentation des couleurs (figure suivante).

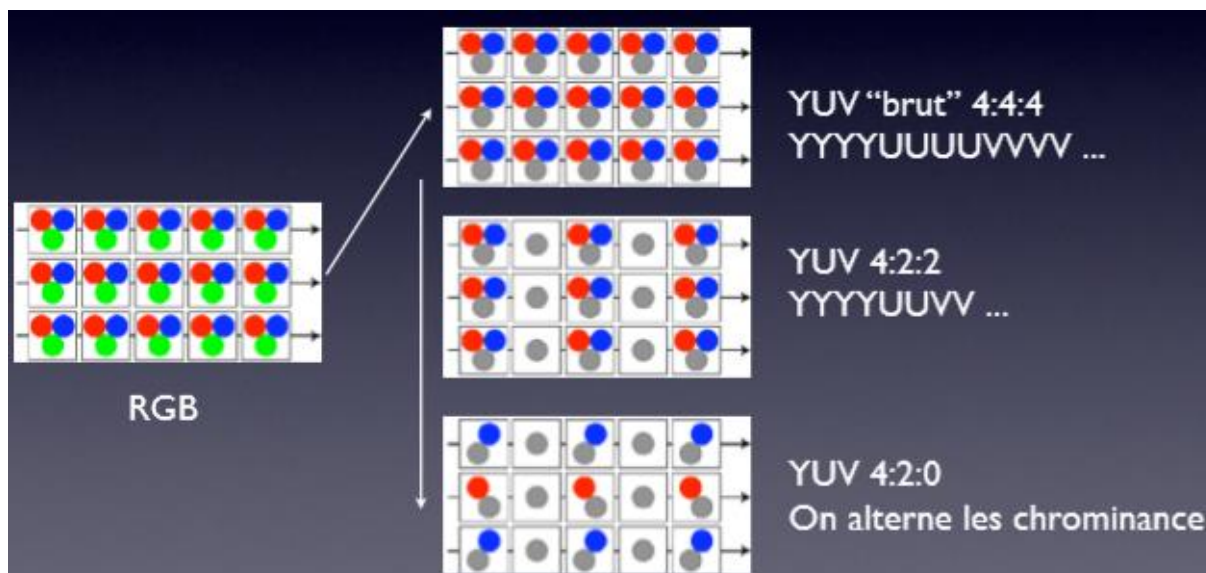


Figure 3.4 : Formats de numérisation des signaux(Y,U,V ou Y,Cr,Cb)

Les images sont tout d'abord numérisées dans un format SIF (format de moindre résolution que le 4 : 2 : 2 utilisé dans JPEG). Ce format consiste en fait à ne conserver :

- qu'un pixel sur deux en luminance, dans le sens horizontal et vertical (résolution 360x288 pixels utiles au lieu de 720x576),
- qu'un pixel sur quatre en chrominance (180x144).

Ensuite, la hiérarchie concerne aussi le découpage (voir la figure suivante) des images à traiter.

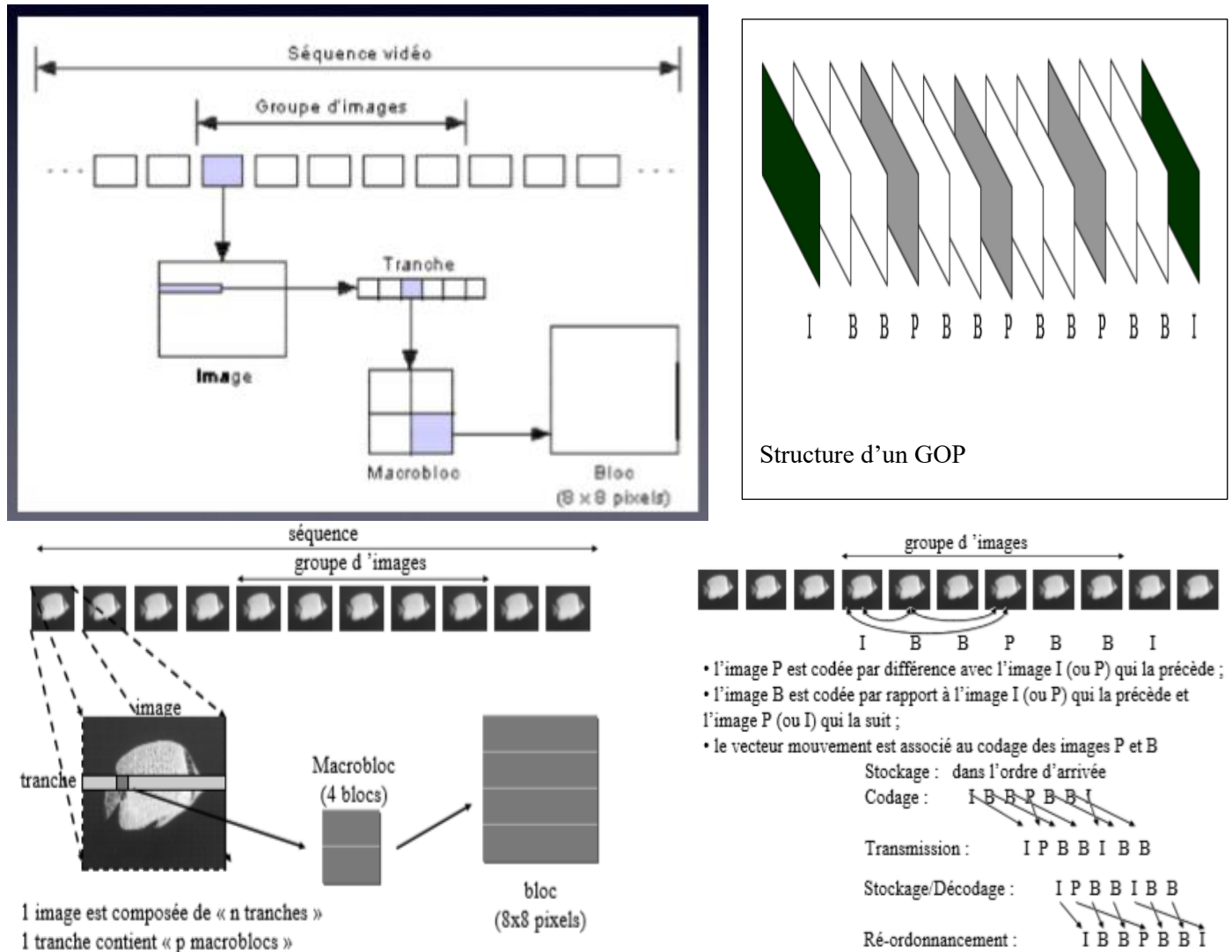


Figure 3.5 : Découpage d'une séquence vidéo

La compression MPEG utilise 2 types de compression, selon que l'on traite la redondance spatiale ou temporelle des séquences d'image. Pour cela, on est amené à analyser une séquence vidéo (figure 3.5) comme suit :

- La séquence vidéo est constituée de plusieurs groupes d'images(GOP). Une image est composée de n tranches. Chaque tranche contient p macrobloccs dont chacun est formé par 4 bloccs de luminance et 2 bloccs de chrominance.
- Le GOP(Group Of Pictures) :**

Il contient 3 types d'images : Les images I (intra-codées), les images P (prédictives), et les images B (bidirectionnelles).Un GOP) est défini par 2 valeurs :

- le nombre d'intervalles séparant 2 images I et noté **N**

- le nombre d'intervalles séparant 2 images P ou une image P et une image I, noté **M**

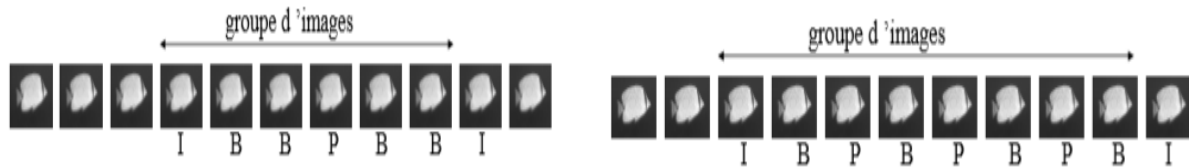


Figure 3.6 : Groupe d'images (6,3) : $N=6, M=3$

Groupe d'images (8,2) : $N=6, M=3$

- Les images I : il s'agit d'images complètes codées selon l'algorithme JPEG vu précédemment. Typiquement, une image I est intercalée dans le flux toutes les 10 à 15 images, c'est à dire qu'on trouve 2 à 3 images I par seconde dans un flux MPEG. Elles ont pour objectif :

1) d'assurer une diffusion multi-destinataire (broadcasting) : un utilisateur peut se connecter à tout moment au flux.

2) d'assurer qu'en cas d'erreur, il soit possible quand même de continuer la lecture. Il faut pour cela attendre la prochaine image I.

3) d'assurer la lecture avant ou retour rapide, sans que le décodeur n'ait besoin de décoder l'ensemble des images du flux.

Les images I, comme leur nom l'indique, sont codées indépendamment des autres, à la manière d'une image fixe (proche de JPEG). C'est le codage spatial de l'information.

- **Les images P** : Une image P est prédite à partir de l'estimation du mouvement entre l'image I précédente et cette image, ou entre l'image P précédente et cette image. On a besoin pour cela d'un algorithme prédisant les mouvements d'une image à l'autre (estimation de mouvement : Motion Estimator). Ces images sont codées par rapport à l'image précédente. Elles ne codent donc que la différence bloc par bloc de l'image précédente. Pour maximiser le taux de compression, on cherche donc dans l'image précédente un macrobloc identique ou semblable.

En conséquence, les images P peuvent être vues comme une mosaïque :

- de blocs composés d'un vecteur (pour indiquer où se situait le bloc dans l'image précédente)
- et de blocs complets correspondant aux zones inexistantes dans l'image précédente

Trouver le vecteur de déplacement d'un bloc d'une image à l'autre est appelé compensation de mouvement.

- **Les images B** : sont prédites à partir d'une interpolation bidirectionnelle entre images I et P, ou P et P voisines.

Pour les images P et B, on codera les informations de déplacements (vecteurs de mouvement : Motion Vectors) et la différence entre l'image prédite et l'image réelle.

Pour les images B, les vecteurs de déplacements sont interpolés temporellement, et l'image est prédite soit à partir de l'image I ou P précédente (prédiction avant : Forward), soit à partir de l'image I ou P suivante (prédiction arrière : Backward), soit de manière bidirectionnelle à partir des deux I ou P voisines. Parmi les trois techniques, l'image prédite la plus proche de l'image réelle est retenue, et la différence est encodée comme pour les images P.

Les images de type B peuvent se référer à une image de type I ou P, mais jamais à une autre image B.

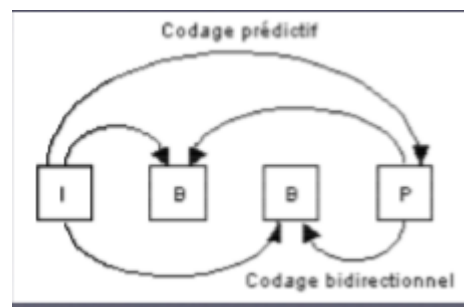
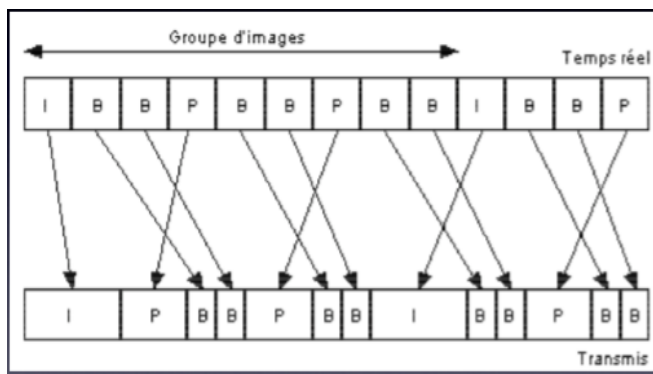


Figure 3.7 : Composition du flux vidéo

Cette figure montre également que les places respectives occupées par les images de type I, P ou B.

3.7.2. Estimation du mouvement

Les images sont discrétisées en macroblocs : un macrobloc contient 16x16 pixels de luminance (c'est-à-dire 4 blocs voisins utilisés pour la DCT), et donc 8x8 (la moitié) pixels de chrominance (dans le format SIF).

Pour un macrobloc de l'image P courante, on recherche dans l'image (I ou P) précédente le macrobloc voisin le plus semblable, en termes de luminances et de chrominances. Une fois identifié, on peut évaluer le « vecteur déplacement » (motion vector) du macrobloc qui sera appliqué à tous les pixels du macrobloc et aux trois composantes (Y, Cb,Cr).

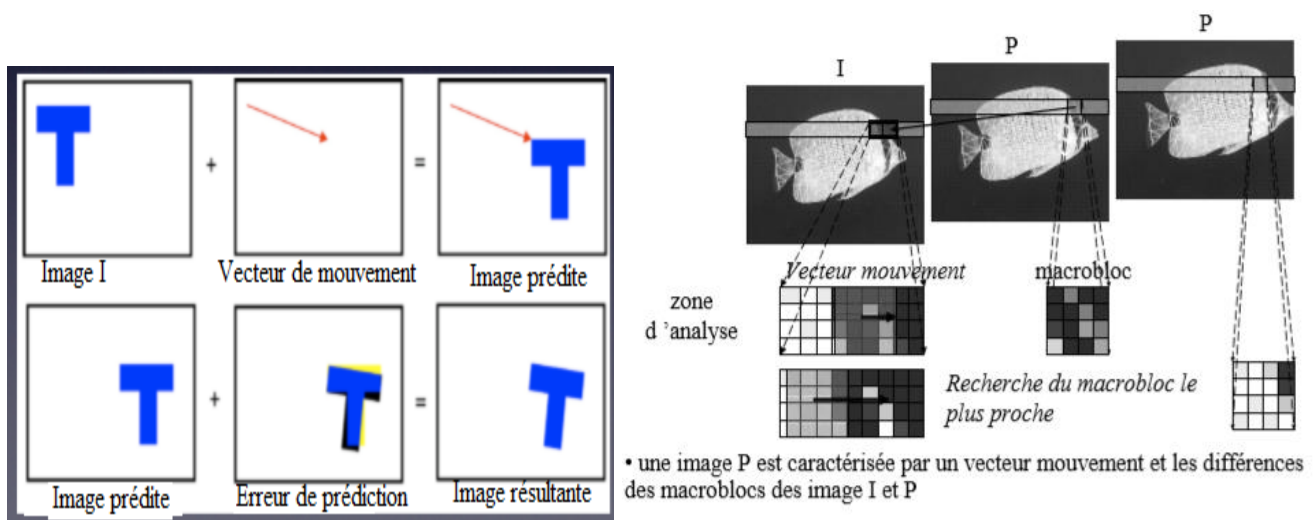


Figure 3.8 : vecteurs et estimation de mouvement

Le codage de l'image prédite P (à partir de l'image I ou P précédente) concerne :

- Les vecteurs de mouvement codés,
- La différence entre l'image réelle P et l'image prédite par l'ensemble des les vecteurs de mouvement, cette différence (ou résidu) étant codée en intra (DCT, etc..) : voir figure suivante.

On peut imaginer la complexité de l'encodeur (Figure 3.9), notamment en relation avec l'estimateur de mouvement. Le codec est dit asymétrique, car le décodeur est moins complexe que l'encodeur (même chose qu'en compression audio).

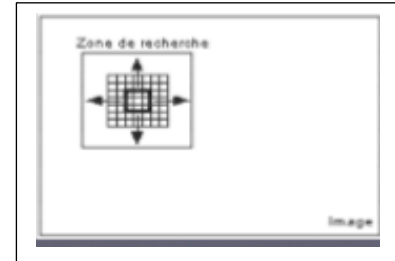
3.7.3. Compensation de mouvement

La compression par compensation de mouvement consiste à considérer que la vidéo est en réalité un fond statique (background) et un ensemble d'objets en mouvement (foreground) :

- Coder la première image en JPEG et l'utiliser comme référence
- Diviser la prochaine image en blocs et comparer chacun d'eux avec le bloc correspondant dans l'image de référence.
- Coder les différences entre les blocs
- Continuer avec les autres images.

La norme MPEG ne spécifie pas comment faire la recherche de blocs similaires entre 2 image, ni où la faire dans l'image, ni le seuil de ressemblance à atteindre. Il existe différentes méthodes de recherche traditionnellement utilisées :

- similitude de blocs (block matching),
- similitude de gradients (gradient matching),
- similitude de phase (phase correlation)



• **Similitude de blocs (block matching) :**

1/ la recherche de blocs semblables n'est lancée que si les 2 blocs dans les 2 images à la même position sont suffisamment différents.

2/ les blocs semblables sont recherchés dans un voisinage, et seulement sur la luminance

3.7.4. Prédiction de l'erreur : les mouvements dans l'image sont souvent plus complexes que de simples translations. Ainsi, la prédiction obtenue par compensation de mouvement n'est pas suffisante. De ce fait, on code également l'erreur de prédiction.

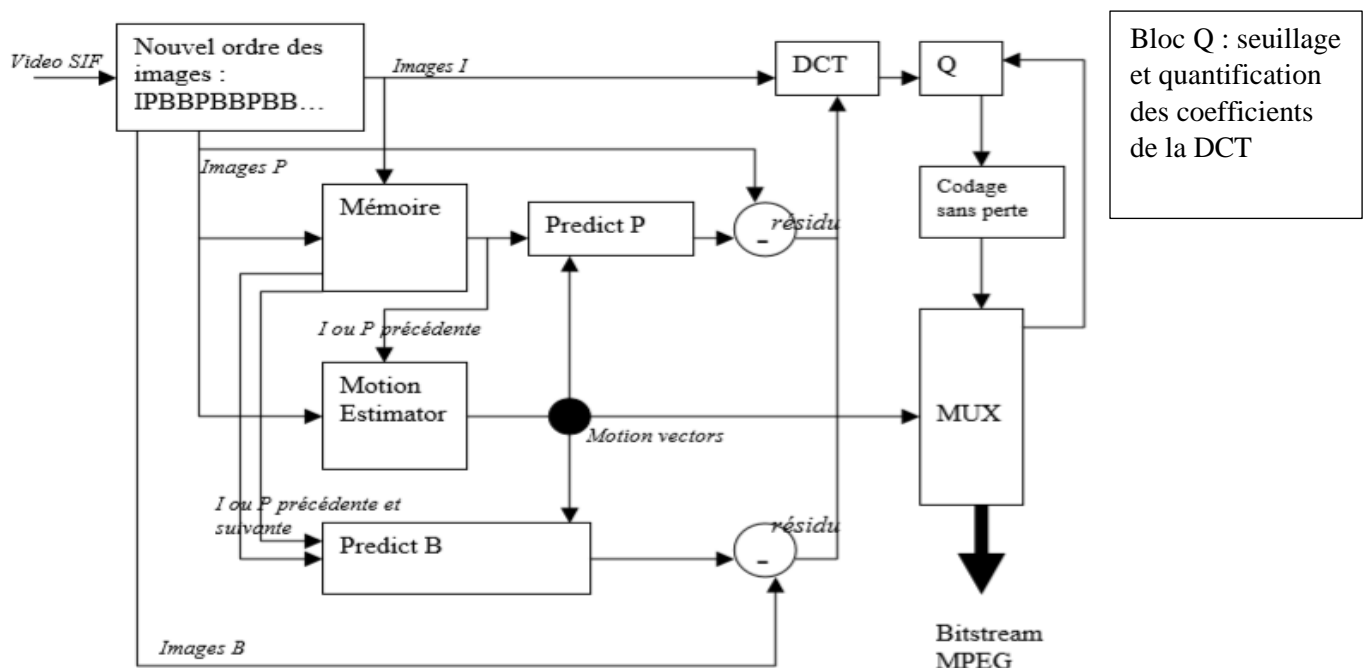


Figure 3.9 : Schéma global de l'encodeur MPEG

Comme on l’observe sur la figure 3.9, l’ordre initial des images IBP est modifié de manière à faciliter les opérations de prédiction. En général, on retrouve 2 images B entre deux images P, et 11 images entre deux images I.

Une boucle de régulation est prévue après le multiplexeur (MUX) : elle agit sur le niveau de quantification des coefficients de la DCT de manière à assurer un bit rate en sortie constant.

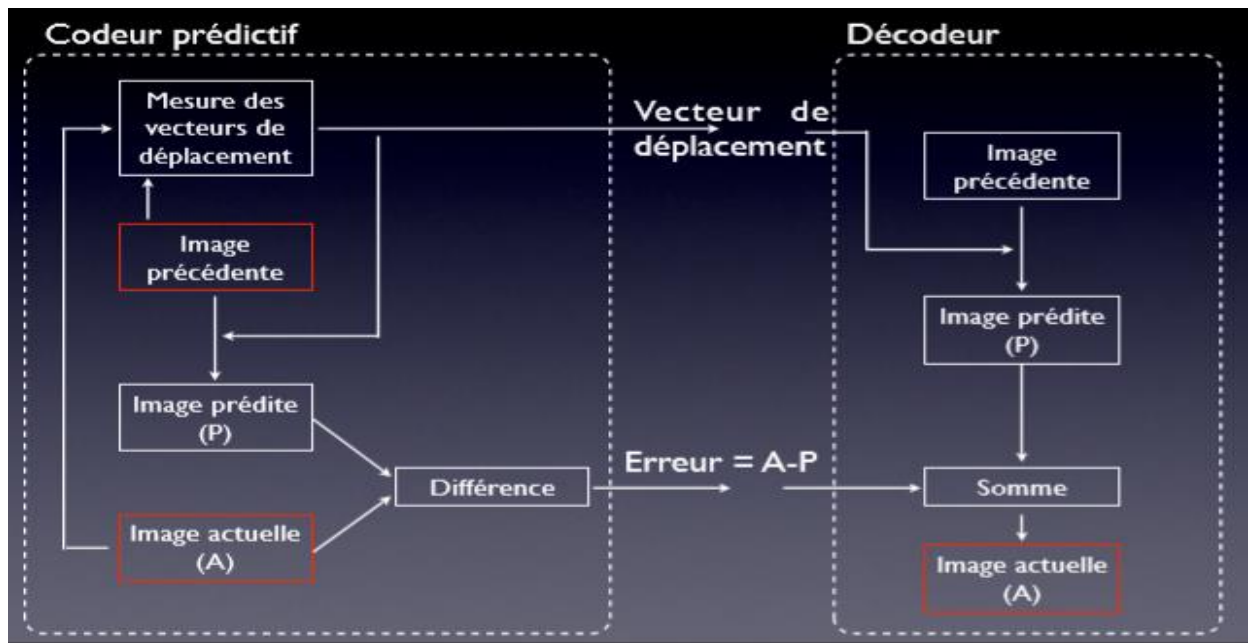


Figure 3.10 : Codeur/décodeur vidéo

3.8. Les normes de compression vidéo

3.8.1. MPEG-1

C’est le premier système numérique de qualité VHS, élaboré par le groupe MPEG en 1990. Son principe est décrit précédemment, avec une résolution de 320x240, avec un maximum de 30 images/s, les images devant être en mode progressif (format plein), couleurs en 4:2:0. La partie audio est codé en MPEG-1 Audio Layer I, II ou III (MP3 pour ce dernier).

3.8.2. MPEG-2

En 1994, le groupe MPEG définit la norme de seconde génération MPEG-2. Cette norme est aujourd’hui la plus utilisée, depuis les DVD Vidéo jusqu’à la télévision Haute Définition (HDTV). Elle permet la compression d’images entrelacées (télévision numérique), et autorise l’emploi de 4 résolutions (352x288 jusqu’à 1920x1152). MPEG-2 définit également des profils, sélectionnant des compressions sans images B, des choix différents pour les couleurs, ou des débits différents. Le multiplexage de données est plus général également : sous-titres, etc.

L’apport de MPEG-2 par rapport à MPEG-1 concerne principalement la création de différents niveaux et différents profils.

Quatre niveaux	Cinq profils
Low : SIF, MPEG-1 (360x288)	Simple : pas d’image B
Main : standard 4 :2 :0 (720x576)	Main : I,B,P
High-1140 : résolution (1440 x 1152)	SNR scalable (pour mémoire)
High : résolution (1920 x 1152)	Spatially scalable (pour mémoire)
High : 4 :2 :0 ou 4 :2 :2	pour TV haute définition

Pour (presque) chaque combinaison, il existe des algorithmes de compression proposés dans le standard MPEG-2. La plus courante est l’association Main level at Main profile (MP@ML), c’est-à-dire celle qui

permet d'atteindre des taux d'information de 4 à 9 Mb/s et que l'on retrouve notamment dans le standard DVD. 4 Mb/s correspond plutôt à une qualité vidéo analogique PAL et 9 Mb/s correspond plutôt à une qualité vidéo numérique CCIR-601.

En outre, dans MPEG-2, trois modes existent pour l'estimation des mouvements dans les signaux vidéo avec entrelacement : les blocs et macroblocs peuvent ainsi être créés sur l'image complète (formée par deux trames successives) ou sur chaque trame séparément (une trame = une demi-image). Donc l'unité de compression MPEG est le macrobloc.

Estimation de mouvement en MPEG-2

- calcul sur 1 bloc du signal luminance ;
- appliqué aux 4 blocs du macrobloc ;
- appliqué aux signaux R-Y et B-Y ;
- si incalculable, bloc codé en image I ;
- vecteur mouvement codé sur 12 bits ;
- un décodeur dispose en permanence de 4 images en mémoire ;
- un codeur dispose en permanence de 8 images en mémoire ;
- un décodeur est paramétré pour un profil donné ;

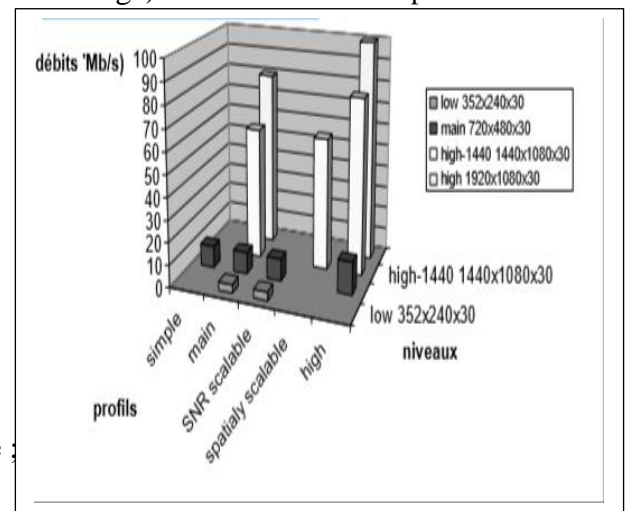


Figure 3.11 : Profils MPEG-2

La norme permet la définition de la résolution de l'image à partir de 5 profils et 4 niveaux.

3.8.3. MPEG-4

Le standard MPEG-4 (aussi appelé H.264) fait son apparition en 1998 : Il est aujourd'hui utilisé dans le format Blu-Ray Disc, ou prochainement exploité pour la TNT HD. Cette norme ajoute un aspect multimédia interactif et se veut plus générique qu'une simple méthode de compression vidéo. Elle définit des objets audiovisuels.

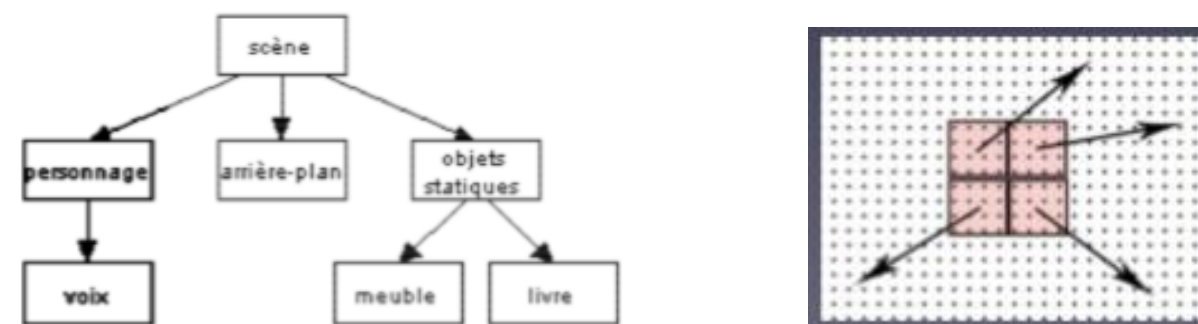


Figure 3.12 : Aspect multimédia interactif du MPEG-4

Techniques spécifiques au MPEG-2/4 et améliorations :

- Quartel-pel : le mouvement d'une image à une autre n'est pas forcément décrit par un nombre entier de pixel. L'idée est donc de travailler avec des quart de pixels extrapolés selon un algorithme bilinéaire ou bi-cubique.
- Global Motion Compensation (GMC) : l'objectif ici est de chercher un effet global dans la vidéo (translations, changement de perspective, rotations, etc.). La GMC permet de définir des points d'ancrage liés au mouvement de l'image.
- Quantification adaptative : la table de quantification est adaptée à chaque bloc à traiter.
- 4MV : pour le MPEG-4, les macroblocs 16x16 sont subdivisés en 4 blocs 8x8, chacun possédant son propre vecteur de compensation.

3.8.4. Le MPEG4 part 10 (AVC)

La norme MPEG 4 regroupe plusieurs parties définissant chacune un contexte précis d'utilisation. Par exemple, la partie 3 est une norme de compression pour le codage perceptuel et les signaux audio ; elle spécifie notamment le format audio AAC. La partie 10 est une norme avancée de compression vidéo appelée aussi H.264 ou AVC (Advanced Video Codec).

Fin 2003 : H.264/AVC (MPEG-4 Part 10) adopté par le 3GPP et HD-DVD, comprend de nombreuses techniques nouvelles qui lui permettent de compresser beaucoup plus efficacement les vidéos que les normes précédentes (MPEG-1, MPEG-2...) et fournit plus de flexibilité aux applications dans un grand nombre d'environnements réseau.

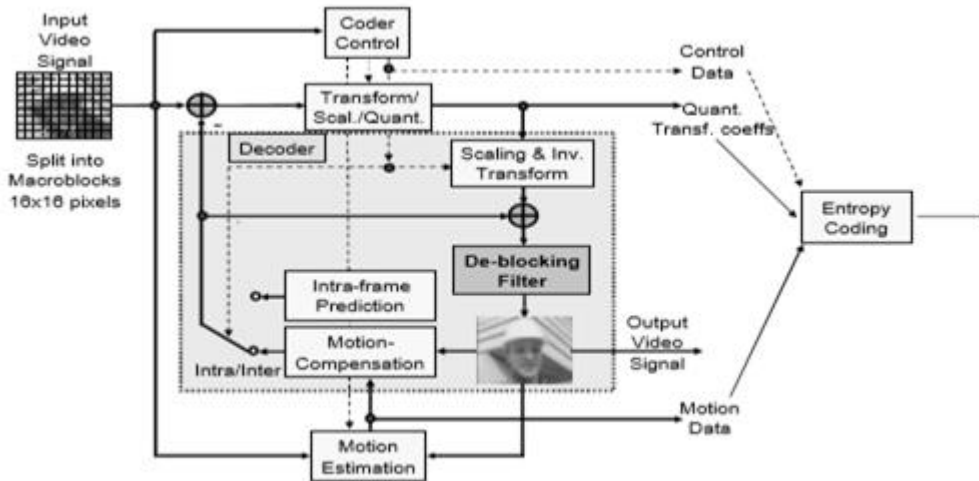


Figure 3.13 :Codec de base du H.264

H.264 ajoute une étape de déblocage (deblocking) après la quantification et un **Codage temporel** qui permet l'identification des cycles /périodicités des mouvements. En outre, la mémoire d'images est plus importante.



1) Sans Filtre

2) avec Filtre H264/AVC Deblocking

Figure 3.14 : effet du déblocage

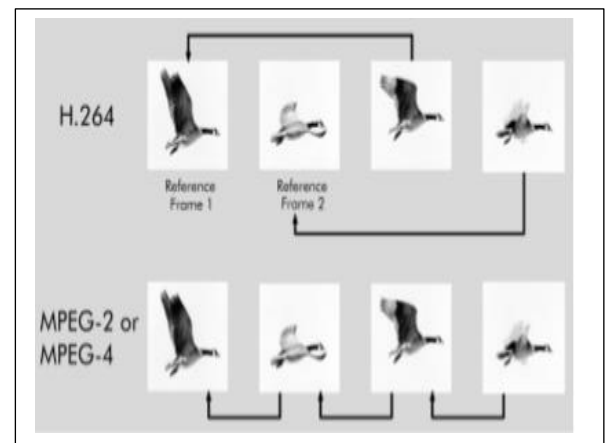
Mesure de la Qualité Vidéo : PSNR

Soit une séquence vidéo de K images et de résolution spatiale MxN

- PSNR : Peak To Signal Noise Ratio
- RMSE : Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{N \cdot M \cdot K} \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^M [x(i,j,k) - \hat{x}(i,j,k)]^2}$$

$$PSNR = 20 \cdot \log_{10} \frac{255}{RMSE}$$



PSNR_A = 37 dB

PSNR_B = 37 dB

Figure 3.15 : Inconvénient du PSNR

3.8.5. MPEG HEVC / H265 (High Efficiency Video Coding)

Nouveaux profils

- Main 12, Main 4:2:2 10, Main 4:2:2 12, Main 4:4:4, Main 4:4:4 10, Main 4:4:4 12, Monochrome 12, Monochrome 16, Main 12 Intra, Main 4:2:2 10 Intra, Main 4:2:2 12 Intra, Main 4:4:4 Intra, Main 4:4:4 10 Intra, Main 4:4:4 12 Intra, Monochrome 12 Intra et Monochrome 16 Intra

- prise en charge de nombreux formats (de la visio à la TVUHD 8K)

- 100, 120 et 150 images par secondes

- performances :

- blocs de taille variable

- GOP de taille variable

- taux de compression de l'ordre de 1/100

3.9. Les techniques de compression MPEG audio

3.9.1 Principes du MPEG audio

Là encore, les limites de l'oreille humaine seront exploitées afin de réduire la quantité d'informations nécessaires pour encoder les signaux audio sans altérer de manière perceptible la qualité du son à reproduire.

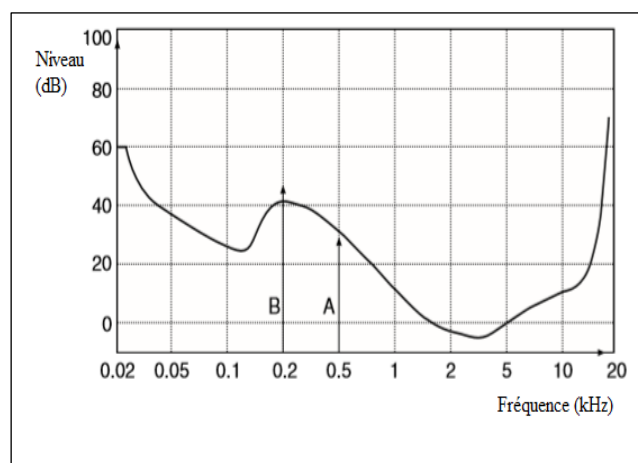
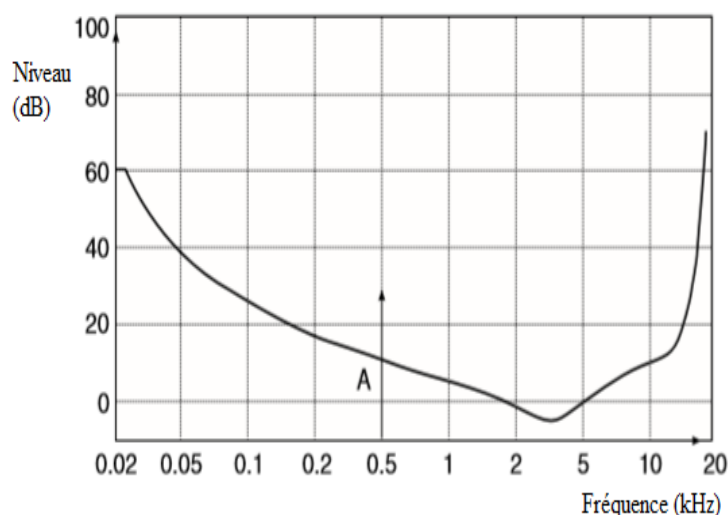


Figure 3.16 : Sensibilité de l'oreille en fonction de la fréquence. Figure 3.17 : Effet de masquage fréquentiel.

Depuis longtemps, on sait que l'oreille humaine a une sensibilité maximale pour les fréquences audio allant de 1 à 5 kHz. La courbe de sensibilité, qui représente le seuil d'audibilité ou de perception en fonction de la fréquence en l'absence de tout signal «perturbateur», est représentée sur la figure 3.16, où l'on voit que le signal A est audible, car il dépasse le seuil d'audibilité.

a) Masquage de fréquentiel

Plus récemment, il a été suggéré que cette courbe est modifiée en présence de signaux multiples ; par exemple, dans le cas de deux signaux de fréquences relativement proches, le signal le plus fort augmente le seuil de perception au voisinage de sa fréquence. Cet effet est illustré sur la figure 3.17, où l'on peut voir que le signal A, auparavant audible, est maintenant masqué (le signal A est inaudible) par le signal B qui est plus puissant que A. Cet effet est appelé **masquage de fréquence**.

b) Masquage de temporel

Un son de forte amplitude masque également les sons qui le précèdent immédiatement ou le suivent dans le temps, comme illustré sur la figure 3.18.

On distingue deux types de masquage temporel :

- le masquage antérieur (pré masquage) : Le masquage antérieur où le signal test précède le signal masquant ne s'exerce que dans les 30 à 40 ms qui précèdent le masquant.
- le masquage postérieur (post masquage) : Le masquage postérieur où le signal test suit le signal masquant s'exerce pour les retards inférieurs à 1 ms avec un niveau de seuil identique au masquage fréquentiel ; puis ce seuil décroît jusqu'à la valeur de seuil de 200 ms.

3.9.2 Codage et compression du signal audio

Le modèle psychoacoustique de l'audition humaine a été utilisé à la base de la conception de l'encodeur conceptuel, caractérisé par une courbe de masquage et une quantification de signaux variant en fonction des signaux à coder.

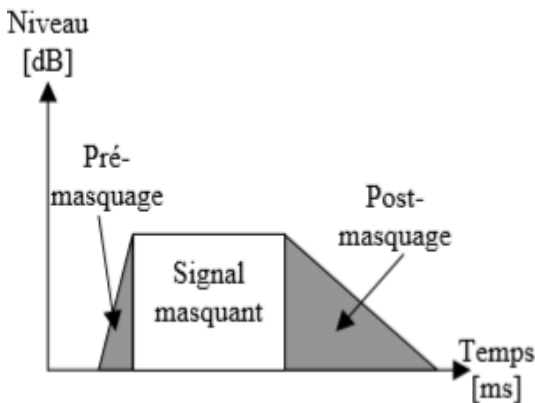


Figure 3.18 : Effet du masquage temporel.

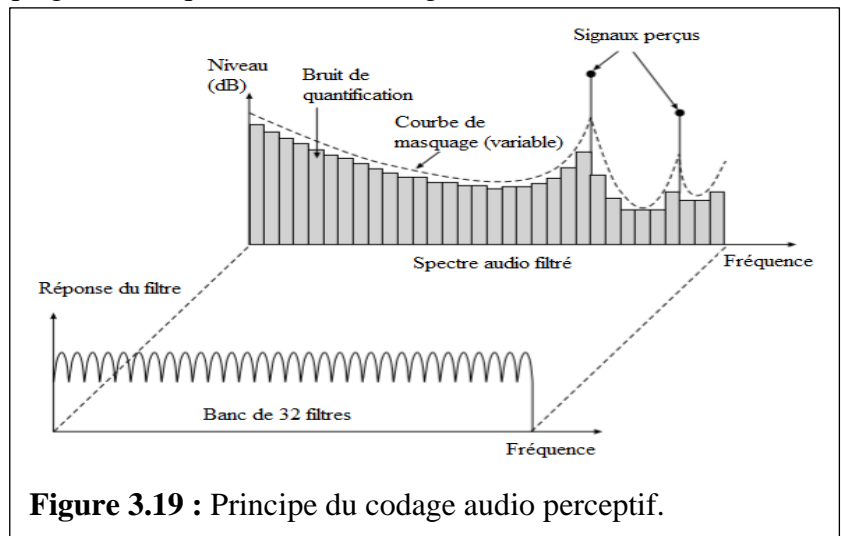
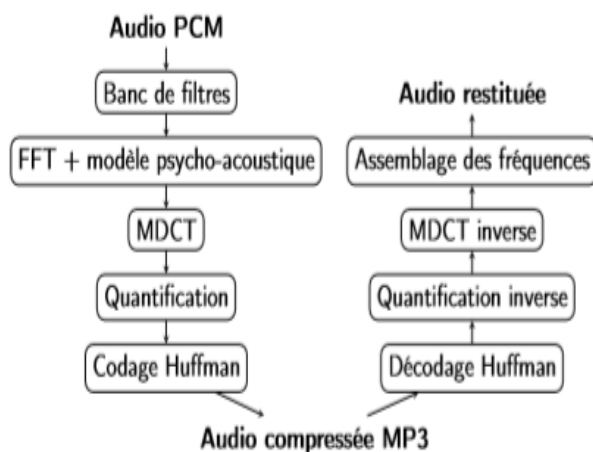


Figure 3.19 : Principe du codage audio perceptif.

Le principe du processus de codage consiste à d'abord diviser la bande de fréquences audio en 32 sous-bandes d'égale largeur au moyen d'un banc de filtres polyphasés. Le signal de sortie d'un filtre de sous-bande correspondant à une durée de 32 échantillons PCM est appelé un échantillon de sous-bande. Le principe du codage perceptif est illustré à la figure 3.19.

Ce modèle psycho-acoustique à masque variable est caractérisé par :

- Un banc de 32 filtres dits polyphasé ;
- Echantillonnage à 32KHz, 44,1KHz, 48KHz ;
- Quantification du masquage sur une trame de 12x32 échantillons ;
- Quantification sur 16 bits avec facteur d'échelle ;
- Codage multicanal en MPEG-2 (Left, Center, Right, Left Surround, Right Surround).



- Regroupement en trame de 1152 échantillons (18, 26 ou 36 ms)
- banc de 32 filtres
 - différents des bandes-critiques mais simples
 - ré-amplification par bandes
- FFT et utilisation du modèle psycho-acoustique
 - FFT réalisée sur 1024 point
 - Identification des masqueurs
 - déduction de la courbe de masquage
- MDCT (Modified Discrete Cosinus Transform)
 - découpe en 18 sous-sous-bandes → 576 bandes-étroites (28 à 55 Hz)
 - permet de s'approcher des bandes-critiques
- quantification non linéaire
 - suppression des composantes totalement masquées
- codage de Huffman

3.10. Multiplexage des signaux

3.10.1. Multiplexage MPEG-1

Le multiplexage permet d'assurer la transmission des signaux audio/vidéo vers les unités de stockage. Le multiplexeur :

- Reçoit du Codec les trains élémentaires de données (Elementary Streams) ;
- Découpe les trains élémentaires (ES) en paquets (Packetized Elementary Stream) ;
- Regroupe les paquets en packs dont l'en-tête contient les informations de temps et de débits. Chaque pack contient :
 - 0 à 16 paquets vidéo ;
 - 0 à 32 paquets audio ;
 - 0 à 2 paquets de données privées

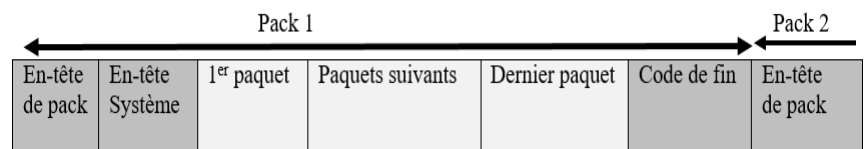


Figure 3.20 : Flux de données en MPEG-1

3.10.2. Multiplexage MPEG-2

Pour le stockage des données audio/vidéo, la norme MPEG-2 reprend le multiplexage de la norme MPEG-1. La norme prévoit un flux de transport (Transport Stream) pour la transmission sur les média des programmes destinés à la diffusion.

Le multiplexeur :

- Découpe les flux élémentaires (ES) en paquets de 188 octets ;
- Constitue des trains de données pouvant combiner plusieurs programmes.

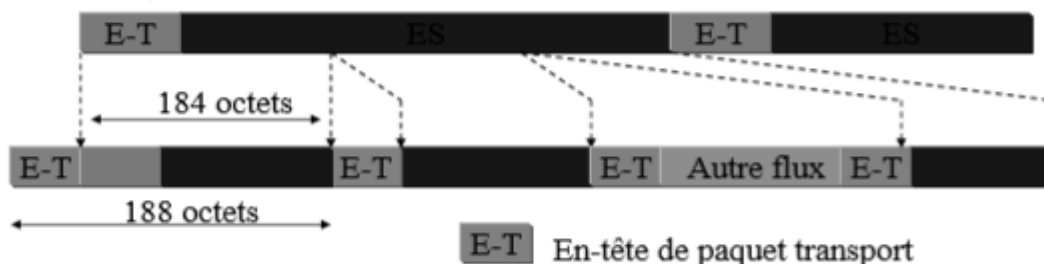


Figure 3.21 : Paquet de flux élémentaires pour MPEG-2

Exercices

Exemple 1 : Transmission par télécopie

Une page à transmettre est constituée de points blancs et noirs. Chaque point est représenté par un élément binaire ("1" si la couleur est noire, "0" si elle est blanche).

Les dimensions de la page sont de 8,5x11 pouces. Sachant que la résolution est de 200 points par pouce, le nombre d'éléments binaires nécessaires pour représenter une page est donc :

$$(8,5 \times 200) \times (11 \times 200) = 3,74 \text{ Mbits}$$

Si on utilise un modem au débit de 14,4 kbits/s, le temps nécessaire à la transmission d'une page serait de $3,74 \cdot 10^6 / 14,4 \cdot 10^3 = 259 \text{ s}$.

Grâce aux méthodes de compression, cette durée est réduite à 17s !

Exemple 2 : Stockage d'une séquence vidéo sur un CD-ROM

Une image de télévision numérique est formée de 720 points pour chacune des 480 lignes. L'œil étant moins sensible à la couleur qu'à la luminance, on estime qu'il est suffisant que Cb et Cr ne soient présents qu'un point sur deux.

Dans ce cas, le nombre d'éléments binaires décrivant une image est de :

$$(480 \times 720) + (480 \times 720 \times 1/2) + (480 \times 720 \times 1/2) = 691200 \text{ octets}$$

La transmission d'une séquence vidéo de 25 images/s nécessite donc un débit de :

$$691200 \times 25 = 17,28 \text{ Mo/s}$$

Ainsi, sur un CD de 650 Mo, on peut stocker l'équivalent de :

$$650/17,28 = 38 \text{ s de vidéo !}$$

Grâce aux méthodes de compression, cette durée est très nettement plus importante !

Exemple 3 : Fichiers musicaux

Considérons un signal analogique stéréo. Sa numérisation en "qualité CD" requiert une fréquence d'échantillonnage de 44,1kHz, et une quantification des échantillons sur 16 éléments binaires (2 octets).

Le nombre d'éléments binaires nécessaires à la représentation d'une seconde de musique (stéréophonique) est :

$$44,1 \cdot 10^3 \times 2 \times 2 = 176,4 \text{ ko/s}$$

Avec un tel débit, un CD de 650 Mo peut stocker environ $650/0,1764 = 61$ minutes de musique.

Grâce aux méthodes de compression, on peut stocker sur ce même CD plus de 10 heures de musique !

Débit du signal numérique

Quel que soit la quantification utilisée, la numérisation correcte d'un signal analogique de bande passante comprise entre 0 et Fmax, sur N bits, nécessite un débit binaire en sortie de quantificateur D tel que :

$$D = N \cdot F_e \geq 2 \cdot N \cdot F_{\max}$$

On obtient aussi le volume d'informations V en multipliant le débit D par la durée de la capture d :

$$V = D \cdot d.$$

Exemple 4 :

Le codage d'un signal vocal dont la bande de fréquences est comprise entre 0 et 4 kHz, peut être échantillonné à 8 kHz. Une quantification PCM sur 8 bits nous donne un débit de 64 Kb/s.

Exemple 5 :

Une transmission d'images vidéo par voie téléphonique se fait à 360 pixels par ligne, 288 lignes par image vidéo, 12 bits/pixel et une fréquence de 10 im/s. Le débit numérique est donc :

$$360 \times 288 \times 12 \times 10 = \mathbf{12,44 \text{ Mbit/s}}$$
 (des millions de bits par seconde).

Exemple 6 :

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 de quantification : 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 de-quantifiée

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

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

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

Exercice 2 : Considérons les schémas transformés de deux exemples « 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.

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

- 1) Calculer les DCT pour chaque cas en utilisant les équations données au cours pour vérifier les résultats trouvés.
- 2) Les deux matrices originales, à quelles composantes (Y,Cb,Cr) du signal vidéo correspondent-elles ? justifier votre réponse.
- 3) Donner les matrices DCT quantifiées.
- 4) Donner le codage de Huffman concernant la suite des coefficients non-nuls rencontrés au parcours zigzag du schéma quantifié.

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

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

[illegible]