Conservatoire National des Arts et Métiers

Polycopié de cours

Electronique C3

Version provisoire du mardi 18 septembre 2002

Télévision numérique et multimédia : 2ème partie

C.ALEXANDRE

1 Compression sans pertes

1.1 Rappels de théorie de l'information

1.1.1 Définitions

Soit un message de longueur l symboles pris dans un alphabet de N symboles différents. On définit les grandeurs suivantes :

- z est le nombre fini d'états possibles du message. (ex : une page texte ayant 40 lignes de 80 lettres avec 26 lettres possibles $\Rightarrow$ l = 3200, N = 26, $z = 26^{3200} = 10^{3200.\log(26)} = 10^{4526}$).

- I est la quantité d'information [unité : Shannon] transportée par le message (ex : I = 15041,4 sh).

$$I = \log_2(z) \quad \text{avec} \quad \log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}$$

  Pourquoi ? Soit deux messages M1 et M2 (avec I1 I2 et z1 z2), on veut que la quantité d'information de la réunion des deux messages en un message M (avec I et z) soit la somme I1+I2. Comme on a z = z1.z2, cela implique que I = log(z). En effet, I = log(z) = log(z1.z2) = log(z1)+log(z2) = I1 + I2. On utilise le log en base 2 parce qu'on travaille le plus souvent en binaire.

- H est l'entropie d'un message. C'est la quantité d'information par symbole [unité : Shannon/symbole].

$$H(M) = \frac{I}{l} = -\sum_{i=0}^{N-1} p_i.\log_2(p_i)$$

  avec l'alphabet composé des symboles $\{S_0, S_1, ..., S_{N-1}\}$ ayant les probabilités d'apparition $\{p_0, p_1, ..., p_{N-1}\}$ tel que $\sum_{i=0}^{N-1} p_i = 1$. (ex : 26 symboles équiprobables (p_i = 1/26) $\Rightarrow$ $p_i.\log_2(p_i) = -0.1808 \Rightarrow$ H = 4.7004 sh/sym, H = I/l = 4.7004 sh/sym).

Pourquoi? Voir démonstration en bases de transmission numérique.

Propriété : l'entropie est maximale quand les symboles sont équiprobables (en binaire, 0 et 1 équiprobables $\Rightarrow$ H = 1 sh/bit).

- La redondance est le rapport entre la quantité d'information portée par un signal et celle qu'il transporterait si tous les symboles qui le constituent étaient équiprobables. Elle traduit le degré d'inefficacité du code utilisé pour traduire le message en signal.

$$R = 1 - \frac{H(M)}{H_{max}(M)}$$

Par exemple, en anglais, on connaît la probabilité d'apparition des 26 lettres de l'alphabet. En supposant les symboles indépendants (ce qui n'est pas vrai), on peut calculer H = 2,8 sh/lettre. En supposant l'équiprobabilité, on a vu que H = 4,7 sh/lettre $\Rightarrow$ R = 0,4. (ex : la page texte de départ. On a 10^{4526} combinaisons possibles, mais elles ne donnent pas toutes un message en français correct. De nombreuses combinaisons ne portent pas d'information, il y a redondance).

Propriété : un signal aléatoire a une redondance nulle (puisque les symboles sont équiprobables par définition).

1.1.2 Le message

Il est généralement vu comme une suite de symboles mis en série.

$$M = S_0S_1S_2S_3S_4...S_{N-1}$$

L'information contenue dans le message peut être de nature mono-dimensionnelle (fichier informatique) ou bidimensionnelle (image). L'image doit donc être convertie en signal mono-dimensionnel par analyse ligne à ligne de gauche à droite et de haut en bas des pixels qui la composent.

S_0	S_1	S_2	S_3	S_4
S_5	S_6	S_7	S_8	S_9
S_{10}	S_{11}	S_{12}	S_{13}	S_{14}
S_{15}	S_{16}	S_{17}	S_{18}	S_{19}
S_{20}	S_{21}	S_{22}	S_{23}	S_{24}

Dans le message mono-dimensionnel $M = S_0S_1S_2S_3S_4...S_{N-1}$, il y a une corrélation variable entre les symboles. La probabilité de trouver dans un fichier texte une suite « LES » est plus grande que d'avoir une suite « LZX ». Plus la distance entre deux symboles est grande, plus la corrélation est faible.

Dans une image, il y a corrélation entre les pixels voisins dans les 8 directions. Plus la distance entre deux symboles est grande, plus la corrélation est faible.

1.1.3 La compression sans perte d'informations (lossless)

L'objectif de la compression sans perte d'informations est la réduction de la redondance R. La décompression permet de retrouver exactement le message originel. Pour comprimer un message, on pourra considérer :

- chaque symbole indépendamment. On ne tient pas compte de la corrélation entre les symboles, c'est moins efficace.
- des ensembles de symboles. On essaye de prendre en compte la corrélation entre les symboles, c'est plus efficace.

Propriétés :

1. Le message comprimé est composé de symboles équiprobables (si on considère chaque symbole indépendamment).
2. plus la compression est efficace, et plus le signal portant le message devient aléatoire (R tend vers 0).

Conséquence : si le signal portant le message est aléatoire (ou pseudo-aléatoire), alors la compression est impossible ou encore, plus la différence de probabilité entre les symboles est grande dans le message à comprimer et plus la compression sera élevée.

Le taux de compression relatif (en %) est défini par :

$$\text{taux} = \left(1 - \frac{\text{taille du message compressé}}{\text{taille du message originel}}\right) \text{x}100$$

1.1.4 Le codage par plage (RLC : Run Length Coding)

C'est une méthode élémentaire permettant de prendre en compte la redondance entre symboles. Elle est efficace quand le message est composé de suites de symboles identiques. Au lieu de coder indépendamment chaque symbole, on détermine des couples (nombre de symboles S consécutifs, S).

AABBBDDDEFF = 2A3B3D1E2F

Cette méthode est particulièrement efficace dans le cas d'une image composée de pixels noirs (N) ou blancs (B) comme dans le cas du télécopieur. Avec trois bits, on peut par exemple coder :

N	000
BN	001
BBN	010
BBBN	011
BBBBN	100
BBBBBN	101
BBBBBBN	110
BBBBBBBN	111

Si P est la probabilité d'un pixel noir, le taux de compression est inférieur ou égal à 1-3.P. Par exemple, si P = 0.1, on ne pourra pas espérer dépasser 70 % de taux de compression. Il faut pour que ce codage soit efficace que P soit petit (P < 1/3), sinon, il faut coder les blancs. Si $P_{blanc} \approx P_{noir}$, alors ce codage augmente la redondance au lieu de la diminuer.

1.1.5 Particularité des codes binaires

Il existe des codes binaires appartenant aux familles suivantes :

- Les codes à décodage unique. Il ne doit pas y avoir de source de confusion dans le décodage d'un message.

  Exemple : soient le code $S_0 = 0$, $S_1 = 10$, $S_2 = 00$, $S_3 = 01$ et le message M = 010100. Ce code n'est pas à décodage unique car il y a 4 messages décodés possibles (M1 = 0 | 10 | 10 | 0 = S_0 S_1 S_1 S_0, M2 = 01 | 0 | 10 | 0 = S_3 S_0 S_1 S_0, M3 = 01 | 01 | 00 = S_3 S_3 S_2, M4 = 01 | 01 | 0 | 0 = S_3 S_3 S_0 S_0).

  Définition : Un code est à décodage unique si les messages codés correspondants à deux suites distinctes de n symboles sont distincts quel que soit n.

  Propriétés : un code dont les symboles ont même longueur est à décodage unique.

- Les codes à décodage instantané. Chaque symbole doit être décodé instantanément, c'est à dire indépendamment des autres symboles.

  Exemple : soient les deux codes à décodage unique suivants.

  code 1 $S_0 = 0$, $S_1 = 01$, $S_2 = 011$, $S_3 = 0111$

  code 2 $S_0 = 0$, $S_1 = 10$, $S_2 = 110$, $S_3 = 1110$

  Le code 1 n'est pas instantané. Il faut attendre le début du symbole suivant pour pouvoir un symbole. Le code 2 est instantané. Le bit à 0 sert de marqueur de fin de symbole.

  Définition : Un code est à décodage instantané si aucun des symboles composant ce code n'est le préfixe d'un autre symbole.

  Voyons maintenant comment synthétiser un code à décodage instantané. La méthode suivante produit des codes instantanés à décodage unique. Soit une source S = {S_0, S_1, ..., S_{N-1}} composée de N symboles.

1. On divise S en deux sous-ensembles $S_a = \{S_0, S_1, ..., S_k\}$ et $S_b = \{S_{k+1}, S_{k+2}, ..., S_{N-1}\}$.
2. On affecte le bit 0 aux symboles composant S_a et le bit 1 aux symboles composant S_b.
3. On répète les opérations 1 et 2 jusqu'à l'obtention de sous-ensembles ne contenant qu'un seul symbole.

Exemple : soit un alphabet à 7 symboles.

<table>
<tr><td>S_0</td><td>S_1</td><td>S_2</td><td>S_3</td><td>S_4</td><td>S_5</td><td>S_6</td></tr>
<tr><td colspan="2">0</td><td colspan="5">1</td></tr>
<tr><td>0</td><td>1</td><td colspan="3">0</td><td colspan="2">1</td></tr>
<tr><td></td><td></td><td>0</td><td colspan="2">1</td><td>0</td><td>1</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>1</td><td></td><td></td></tr>
</table>

Les codes obtenus sont les suivants : $S_0 = 00$, $S_1 = 01$, $S_2 = 100$, $S_3 = 1010$, $S_4 = 1011$, $S_5 = 110$, $S_6 = 111$.

- Les codes compacts. Soit une source $S = \{S_0, S_1, ..., S_{N-1}\}$ composée de N symboles et $P = \{P_0, P_1, ..., P_{N-1}\}$ l'ensemble des probabilités des symboles de la source. A chaque symbole S_i correspond un mot-code C_i de longueur l_i (en nombre de bits). La longueur moyenne d'un symbole (donc le nombre de bits utilisé pour coder un symbole) exprimée en bits/symbole est donnée par :

$$\overline{L} = \sum_{i=0}^{N-1} p_i . l_i$$

Le code le plus compact est celui ayant la longueur moyenne la plus faible.

Propriété : La plus petite longueur moyenne d'un symbole $\overline{L}$ possible est égale à l'entropie H de la source (1 bit = 1 Shannon), $\overline{L} = H$.

1.2 Codage statistique

1.2.1 les méthodes de Huffman et Shannon-Fano (symboles indépendants)

On cherche à minimiser la longueur moyenne du symbole. Puisque les probabilités $P = \{P_0, P_1, ..., P_{N-1}\}$ sont fixées par la source une fois pour toute, on va affecter aux symboles les plus probables les mots-codes les plus courts (voir : on essaye d'égaliser les $p_i.\log_2(p_i)$; pour avoir $\overline{L} = H$, on doit obtenir $l_i = -\log_2(p_i)$). C'est le principe de l'alphabet de Morse. On travaille pour l'instant sur chaque symbole pris indépendamment.

La méthode de Shannon-Fano est l'application directe de la synthèse de codes à décodage instantané.

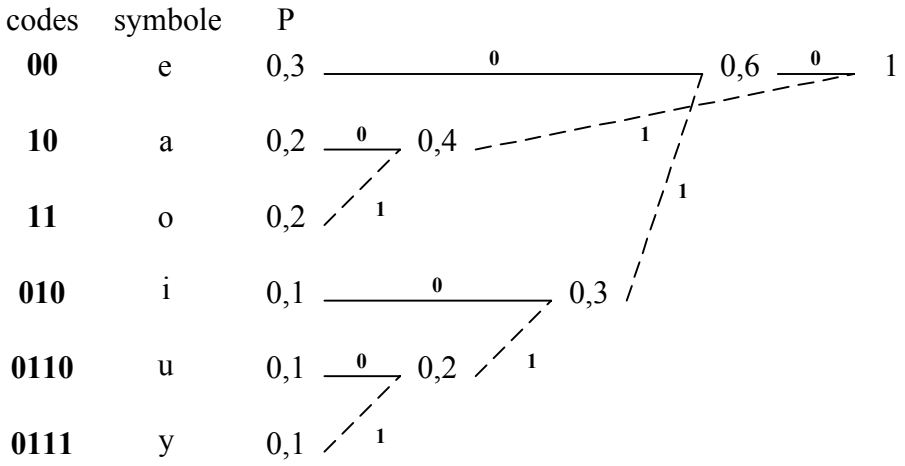
1. On établit la liste des symboles à coder et leur probabilité d'apparition.
2. On les classe par ordre décroissant (par pure commodité).
3. On les sépare en deux sous-ensembles ayant approximativement la même probabilité.
4. On affecte le bit 0 aux symboles composant le premier sous-ensemble et le bit 1 aux symboles composant le deuxième sous-ensemble.
5. On répète les opérations 3 et 4 jusqu'à l'obtention de sous-ensembles ne contenant qu'un seul symbole.

symbole	e	a	o	i	u	y
probabilité	0,3	0,2	0,2	0,1	0,1	0,1
étape 1	**0** (*0,5*)		**1** (*0,5*)			
étape 2	**0**	**1**	**0** (*0,3*)		**1** (*0,2*)	
étape 3			**0**	**1**	**0**	**1**
codes	**00**	**01**	**100**	**101**	**110**	**111**

Résultats : l'entropie H de la source est égale à 2,45 sh/sym. Avec un code de longueur fixe 3 bits, on a $\overline{L} = 3$ bits/sym. Avec le code de Shannon-Fano, on obtient $\overline{L} = 2,5$ bits/sym ce qui est proche de l'entropie.

La méthode de Huffman est assez similaire à la méthode précédente.

1. On établit la liste des symboles à coder et leur probabilité d'apparition.
2. On les classe par ordre décroissant (par pure commodité).
3. On remplace les deux symboles ayant la probabilité la plus faible par un nouveau symbole dont la probabilité est la somme des deux probabilités précédentes.
4. On répète les opérations 2 et 3 jusqu'à l'obtention d'un symbole de probabilité égale à 1.
5. On forme le code en parcourant l'arbre ainsi créé en allant de la racine (symbole de probabilité égale à 1) vers les feuilles (symboles de départ). On attribue 0 aux branches horizontales et 1 aux branches obliques.



Résultats : l'entropie H de la source est égale à 2,45 sh/sym. Avec un code de longueur fixe 3 bits, on a $\overline{L} = 3$ bits/sym. Avec le code de Huffman, on obtient $\overline{L} = 2{,}5$ bits/sym, ce qui est proche de l'entropie.

Remarques : les codes de Shannon-Fano et Huffman sont à décodage unique et instantané. La méthode de Shannon-Fano donne des codes légèrement plus efficaces que ceux de la méthode de Huffman.

1.2.2 La méthode de Huffman (sur plusieurs symboles consécutifs)

Il est apparu rapidement que le codage était plus efficace en codant des paires de symboles (ou des triplets, ou des quartets ...) plutôt que chaque symbole indépendamment. Par exemple, pour un fichier texte, on considère des motifs de plusieurs caractères car la redondance d'une langue se manifeste surtout sur la répétition syllabique. De cette manière, on prend en compte la corrélation entre symboles. L'exemple suivant donne le taux de compression pour un

fichier binaire et un fichier texte en prenant en considération un octet (caractères indépendants) ou deux octets (deux caractères consécutifs).

fichier	un octet	deux octets
binaire	19 %	31 %
texte	40 %	49 %

Il faut alors calculer la table des probabilités par groupes de symboles. La taille de cette table croit fortement avec le nombre de symboles et la taille du groupe. Par exemple, avec 256 symboles (table ASCII), on a les tailles suivantes :

taille du groupe	1	2	3	N
taille de la table	256	65536	16777216	256^N

Problème 1 : pour pouvoir décoder le message, le décodeur doit connaître la table des probabilités. Il faut donc soit insérer une entête contenant cette table dans le message compressé, soit que le codeur et le décodeur utilise toujours la même table. La deuxième solution est très inefficace puisque la table doit être calculée en fonction de la probabilité des symboles composant le message.

Puisqu'il faut insérer une entête, il faut que sa taille soit négligeable devant la taille du fichier comprimé. En codant plusieurs symboles consécutifs, l'amélioration du taux de compression est entièrement perdue à cause de l'accroissement de la table. Au-delà d'un certain point, le message compressé avec la table devient plus grand que le message originel.

Problème 2 : le codage du message nécessite la détermination des probabilités des symboles ou groupes de symboles (ou alors, il faut toujours utiliser la même table). Il y a donc deux lectures du fichier dans le cas général :

1. première lecture, détermination des probabilités des symboles.
2. deuxième lecture, compression du message.

Le codage en temps réel est donc difficile à implémenter. Il faut trouver une solution pour laquelle le codeur et le décodeur calculent en synchronisme la même table de probabilité et si

possible, que cette table évolue en fonction de la statistique de la source (si les probabilités des symboles changent avec le temps). C'est le but du codage de Huffman adaptatif.

1.2.3 Le codage de Huffman adaptatif

Objectifs : l'algorithme adaptatif doit pouvoir (sans se transformer en usine à gaz) :

- supprimer les deux lectures du fichier originel.
- supprimer l'entête du message compressé.
- adapter le codage aux variations statistiques de la source.

La première idée la plus évidente est la suivante :

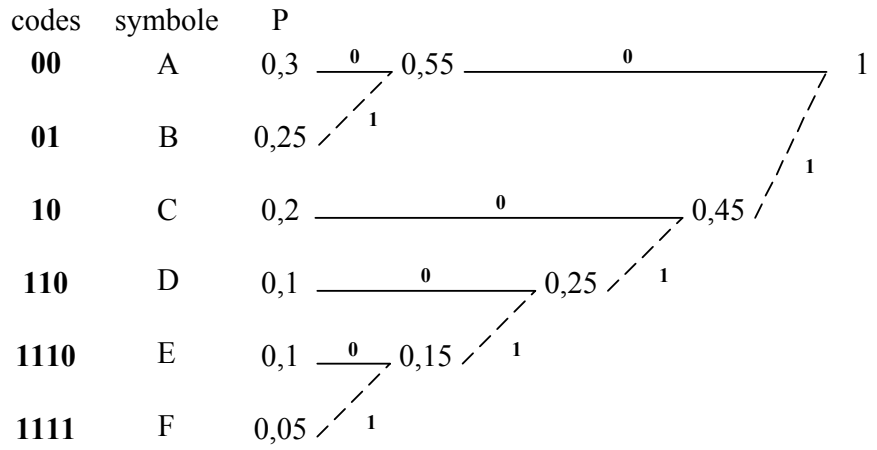
1. on initialise une table des fréquences d'apparition des symboles (équivalente à la table de probabilité) avec la valeur 1 (symboles équiprobables).
2. on calcule un arbre de codage.
3. on code le premier symbole.
4. on augmente la fréquence du symbole émis de 1 puis on recommence les étapes 2, 3 et 4.

Cette méthode est très inefficace en temps de calcul car on doit recalculer un arbre de codage à chaque nouveau symbole émis. Il existe des algorithmes qui permettent de recalculer partiellement l'arbre de codage et donc diminue la charge de calculs. Ils sont utilisés dans les programmes de compression comme PKZIP ou GZIP.

Il existe un exemple simple quoique moins efficace d'algorithme adaptatif sans recalcul de l'arbre de codage. Il faut pour cela construire dynamiquement la table des fréquences d'apparition des symboles en conservant le même arbre de décodage. Les étapes suivantes sont nécessaires :

1. Initialiser la table des fréquences à 0 et déterminer un code de Huffman correspondant au nombre de symboles (en faisant une moyenne sur les différents messages à comprimer par exemple).
2. lire un symbole du message.
3. émettre le code du symbole.
4. incrémenter la fréquence correspondant à ce symbole.
5. classer par ordre de fréquence les symboles, et recommencer les étapes 2, 3, 4 et 5.

Exemple : on prend le code de Huffman suivant comme référence.



On veut émettre le message M = ABC et EEDEFEDFFD. Il y a un changement des propriétés statistiques du message après les 3 premiers caractères (les caractères les plus probables A, B et C deviennent D, E et F).

code	message émis		A	B	C	E	E	D	E	F	E	D	F	F	D
00		A0	**A1**	A1	A1	A1	**E2**	E2	**E3**	E3	**E4**	E4	E4	E4	E4
01		B0	B0	**B1**	B1	B1	A1	A1	A1	A1	A1	**D2**	D2	**F3**	F3
10		C0	C0	C0	**C1**	C1	B1	B1	B1	B1	B1	A1	**F2**	D2	**D3**
110		D0	D0	D0	D0	**E1**	C1	C1	C1	C1	C1	B1	A1	A1	A1
1110		E0	E0	E0	E0	D0	D0	**D1**	D1	D1	D1	C1	B1	B1	B1
1111		F0	F0	F0	F0	F0	F0	F0	F0	**F1**	F1	F1	C1	C1	C1
	code émis		00	01	10	1110	110	1110	00	1111	00	1110	1111	10	10

Avec le codage de Huffman statique, la séquence est composée de 43 bits. Avec cette méthode, elle est composée de 37 bits.

Remarques :

- On ne construit pas réellement d'arbre de codage, on se sert d'un modèle initial que l'on suppose proche de la source.
- Lorsque les fréquences deviennent élevées, l'adaptation est très lente (ou alors, il faut ré-initialiser fréquemment le système).
- La réversibilité de l'algorithme assure un décodage facile du message compressé.
- La table initiale doit être connue (ou transmise) du décodeur.

1.2.4 Le codage arithmétique

C'est un code statistique comme Huffman, mais de conception plus récente. Le principe général a été énoncé par Elias en 1963. Il donne, en général, des résultats supérieurs d'environ 10 % au codage de Huffman au prix d'une complexité algorithmique beaucoup plus élevée. Les seuls algorithmes utilisables pour implémenter le codage arithmétique, les algorithmes du Skew-Soder et du Q-Coder, ont été développés et brevetés par IBM en 1988. Leur utilisation doit donc faire l'objet de paiement de royalties.

L'algorithme de codage utilise le principe suivant :

1. Déterminer les probabilités P_i de chaque symbole S_i.
2. Calcul de l'intervalle $[a_i, b_i[$ associé à chaque symbole S_i, ou a_i est la probabilité cumulée des S_{i-1} symboles et b_i celle de S_i symboles.
3. Codage du message selon la procédure suivante :

   min = 0; max = 1; i = 0;

   Répéter {

   s = max - min;

   max = min + s.b_i;

   min = min + s.a_i;

   i=i+1;

   } tant qu'on n'a pas atteint la fin du message;

Exemple :

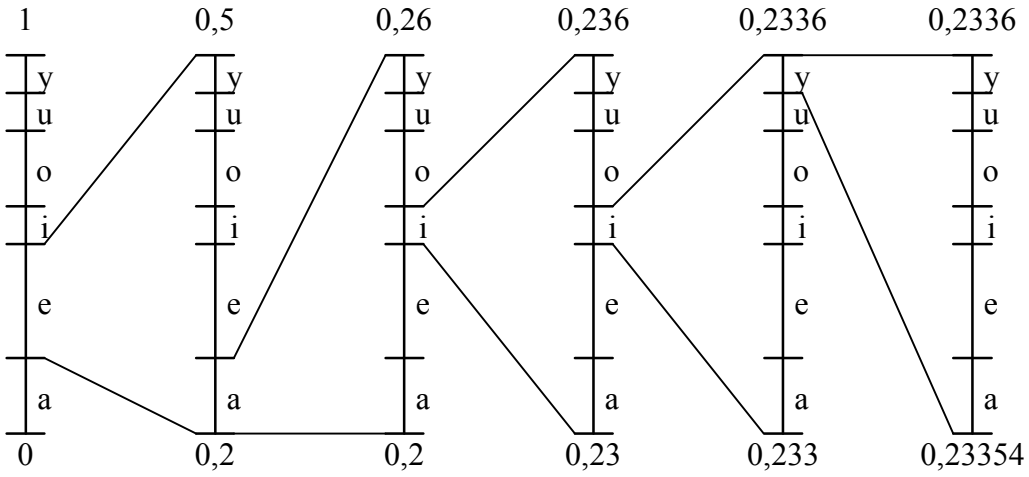
Détermination des probabilités P_i de chaque symbole S_i et calcul de l'intervalle $[a_i, b_i[$.

Symbole	Probabilité	Intervalle
a	0,2	[0, 0.2[
e	0,3	[0.2, 0.5[
i	0,1	[0.5, 0.6[
o	0,2	[0.6, 0.8[
u	0,1	[0.8, 0.9[
y	0,1	[0.9, 1[

Codage du message « eaiiy » par exemple.

caractère	Intervalle	s	Max	Min
-	-	-	1	0
e	[0.2, 0.5[	1	0.5	0.2
a	[0, 0.2[	0.3	0.26	0.2
i	[0.5, 0.6[	0.06	0.236	0.23
i	[0.5, 0.6[	0.006	0.2336	0.233
y	[0.9, 1[	0.0006	0.2336	0.23354

Le message compressé est représenté par les bornes Min et Max finalement obtenues. Il n'est pas obligatoire de transmettre ces deux bornes, une seule suffit ou bien n'importe quelle valeur entre ces deux bornes. On peut aussi voir le processus de codage de la manière suivante :



Remarques :

- Il faudra 5 décimales pour coder le message (il n'y a pas assez de symboles dans ce message pour mettre en évidence la compression H = 2,45 sh/sym).
- Il n'est pas nécessaire d'ordonner la table des symboles par probabilité.
- On ne commence à transmettre le message comprimé qu'à la fin du codage. Il faut utiliser le codage arithmétique incrémental.
- La précision du calcul des bornes Min et Max croit avec le nombre de symboles à comprimer. Cet algorithme n'est donc pas réalisable tel quel.

Principe du décodage du message. L'algorithme de décodage utilise le principe suivant :

1. La variable décimale N représente le message codé (borne Min dans cet exemple).
2. Décodage du message selon la procédure suivante :

   Répéter {

   trouver le ième symbole tel que $N \in [a_i, b_i[$;

   le sauver;

   $s = b_i - a_i$;

   $N = N - a_i$;

   $N = N / s$;

   } tant $N \neq 0$;

Exemple : reprenons l'exemple précédent.

N	i	Caractère décodé	intervalle	s
0.23354 0.03354 ($N=N-a_i$) 0.1118 ($N=N/s$)	2	e	[0.2, 0.5[	0.3
0.1118 0,1118 ($N=N-a_i$) 0.559 ($N=N/s$)	1	a	[0, 0.2[	0.2
0.559 0,059 ($N=N-a_i$) 0.59 ($N=N/s$)	3	i	[0.5, 0.6[	0.1
0.59 0,09 ($N=N-a_i$) 0.9 ($N=N/s$)	3	i	[0.5, 0.6[	0.1
0.9 0 ($N=N-a_i$) 0 ($N=N/s$)	6	y	[0.9, 1[	0.1

Le codage arithmétique incrémental est une évolution du principe précédent. On n'attend pas la fin du codage pour transmettre le message comprimé. En effet, après quelques itérations, les premières décimales ne sont plus modifiées.

caractère	Max	Min	décimale transmise
-	1	0	-
e	0.5	0.2	0
a	0.26	0.2	2
i	0.236	0.23	3
i	0.2336	0.233	3
y	0.2336	0.23354	54

Il est possible d'effectuer les calculs avec un nombre de décimales limité et en utilisant des entiers pour coder les bornes. On obtient alors le codage arithmétique binaire. Voyons son principe sur un exemple :

- Les probabilités des symboles sont codées en puissance de 2 : avec 4 bits, on atteint une précision de $2^{-4} = 0{,}0625$.
- On utilise une source à 4 symboles :

symbole	Probabilité P	Intervalle	P codée en binaire	Intervalle codé en binaire
S_1	1/2	[0, 0.5[	0.1	[0, 0.1[
S_2	1/4	[0.5, 0.75[	0.01	[0.1, 0.11[
S_3	1/8	[0.75, 0.875[	0.001	[0.11, 0.111[
S_4	1/8	[0.875, 1[	0.001	[0.111, 1[

- On veut coder la chaîne « S_1 S_1 S_2 S_3 ».

symbole	Intervalle	Intervalle codé en binaire
S_1	[0, 1/2[	[0, 0.1[
S_1	[0, 0.1/4[	[0, 0.01[
S_2	[1/8, 3/16[	[0.001, 0.0011[
S_3	[11/64, 23/128[	[0.0010110, 0.0010111[

Le codage incrémental est possible (attention à la propagation de la retenue), mais le nombre de bits augmente avec la précision sur l'intervalle. Il faut renormaliser les intervalles. C'est ce que font les algorithmes du Skew-Soder et du Q-Coder (avec en plus un codage adaptatif).

1.2.5 Compression à base de dictionnaire

Cette nouvelle méthode de compression n'est plus basée sur les statistiques des symboles. Elle ne travaille pas au niveau du symbole (généralement un caractère ≡ un octet), mais au niveau de la chaîne de caractères. Voyons le principe général sur un exemple.

On a un dictionnaire contenant 2000 mots de 6 lettres adressables avec un mot de 11 bits. chaque lettre est codée avec 8 bits. L'envoi d'un mot coûte 48 bits alors que l'envoi de son adresse ne coûte que 11 bits. Il y a donc bien compression.

Le premier algorithme utilisable a été conçu par Abraham Lempel et Jacob Ziv et se nomme LZ77. A partir de cet algorithme, de très nombreuses variantes ont été mises au point telles que LZ78, LZSS, LZH, LZW, ... En 1984, M. Welch a introduit une modification dans l'algorithme LZ77 pour obtenir l'algorithme LZW. Cet algorithme est le plus utilisé pour la compression de données sans pertes. On le retrouve notamment :

- dans les modems (norme V42bis),

- dans les compacteurs de fichiers informatiques en association avec le code de Huffman adaptatif (PKZIP, GZIP, COMPRESS, ...),

- dans les formats d'image (GIF, TIFF, ...).

Cet algorithme fait l'objet d'un copyright détenu par Compuserve et Unisys. Il a été pendant très longtemps considéré comme étant libre de droits. En fait, une bataille juridique a eu lieu notamment au sujet du format GIF créé par Compuserve pour transmettre les images sur le réseau. Le format GIF, utilisé dans 98 % des cas sur le réseau, a faillit disparaître. Un accord est finalement intervenu spécifiant que l'utilisation de l'algorithme LZW est libre de droits pour un particulier, mais payante pour un professionnel.

Les caractéristiques de l'algorithme LZW sont les suivantes :

1. Il travaille généralement sur des octets.
2. Le dictionnaire n'est pas enregistré dans le fichier comprimé, mais il est reconstruit automatiquement à la décompression.

3. C'est un algorithme à mémoire fonctionnant sur un mode d'apprentissage. Toute nouvelle séquence de symboles est ajoutée au dictionnaire.
4. Le compactage quasiment en temps réel ne nécessite qu'une seule lecture du fichier.

Voyons l'algorithme de codage. Soit un dictionnaire D dont la taille augmente d'une puissance de 2 chaque fois qu'il est plein. Il contient les séquences de symboles (codés sur N bits) connues repérées par leur adresse. Les adresses 0 à 2^{N-1} contiennent les symboles élémentaires, les adresses supérieures à 2^{N-1} contiennent les chaînes de caractères. Le processus de codage est le suivant :

1. Lecture du premier caractère s.

2. tant que la fin du message n'est pas arrivé

```
{
        t = symbole suivant s;
        u = s ⊕ t;  (⊕ ≡ concaténation)
        si u ∈ D alors
                s = u;
        sinon
                émettre l'adresse de s;
                ajouter u à D;
                s = t;
}
```

Exemple : l'alphabet est composé de 16 caractères (de 0 à F). Le message à transmettre M = 20FAC1D0AFAC1D0F. Les 16 premières adresses du dictionnaire valent :

adresse	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
contenu	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Le tableau suivant nous donne l'évolution des variables internes du codeur :

s	t	u	adresse	bits émis	contenu du dictionnaire	à l'adresse
2	0	20	2	0010	20	10
0	F	0F	0	0000	0F	11
F	A	FA	F	1111	FA	12
A	C	AC	A	1010	AC	13
C	1	C1	C	1100	C1	14
1	D	1D	1	0001	1D	15
D	0	D0	D	1101	D0	16
0	A	0A	0	0000	0A	17
A	F	AF	A	1010	AF	18
F	A	FA	-	-	FAC	19
FA	C	FAC	12	10010	C1D	1A
C	1	C1	-	-	D0F	1B
C1	D	C1D	14	10100		
D	0	D0	-	-		
D0	F	D0F	16	10110		
F	-	-	F	01111		

Remarques :

- Avec un codage à longueur fixe de 4 bits, on aurait émis 64 bits pour coder ces 16 caractères alors que l'on en a utilisé 56 avec cet algorithme, soit un taux de compression de 14 % et ce avec un message très court.
- Il n'y a compression que si la chaîne de caractère existe dans le dictionnaire. Cela ne se produit qu'au dixième, douzième et quatorzième caractère.
- La séquence FAC n'a été vue qu'au cours de son deuxième passage.
- La taille des adresses croit lorsque la taille du dictionnaire dépasse une puissance de 2. Il faut donc signaler au décodeur le passage d'une adresse codée sur N bits à une adresse codée sur N+1 bits, puis continuer à coder toutes les adresse avec N+1 bits (jusqu'au changement de taille du dictionnaire). La signalisation du changement de taille peut se faire soit dans l'entête, soit grâce à un code spécial de changement de longueur.

L'algorithme de décodage est le suivant. La lecture des premières adresses se fait sur N bits. Les 2^N premières adresses du dictionnaire D sont remplies avec les caractères isolés (comme à

l'émission). Le reste du dictionnaire est construit au fur et à mesure de la décompression. Le processus de décodage est le suivant :

1. Lecture de la première adresse a; lecture du caractère correspondant s.
2. tant que la fin du message n'est pas arrivé

{

b = adresse suivante;

si b ∈ D alors

s = [b]; (s = caractère à l'adresse b)

sinon

s = [b] ⊕ t; (⊕ ≡ concaténation)

lecture de s;

t = premier caractère de s;

ajouter [a] ⊕ t à D;

a = b;

}

Exemple : reprenons l'exemple précédent.

a	b	s	t	contenu du dictionnaire	à l'adresse
2	-	2	-	-	-
-	0	0	0	20	10
0	F	F	F	0F	11
F	A	A	A	FA	12
A	C	C	C	AC	13
C	1	1	1	C1	14
1	D	D	D	1D	15
D	0	0	0	D0	16
0	A	A	A	0A	17
A	12	FA	F	AF	18
12	14	C1	C	FAC	19
14	16	D0	D	C1D	1A
16	F	F	F	D0F	1B

1.2.6 Comparaisons et domaines d'utilisation

Le codage de Huffman est utilisé dans JPEG et MPEG pour la compression des images fixes et animées. En effet, les propriétés statistiques des symboles à coder varient peu avec les images à compresser. On a donc des tables de codes fixes par défaut et on prévoit la possibilité de communiquer au décodeur d'autres jeux de tables de codage. De plus, l'implémentation d'un codeur et d'un décodeur dans un circuit intégré est facilité par la faible complexité du processus. Les codes de Huffman, dans leur version adaptative, sont aussi utilisés pour la compression de fichiers informatiques.

Le codage arithmétique est plus performant d'environ 10 % que le codage de Huffman. Son utilisation est prévue dans JPEG, mais elle est limitée par le brevet d'IBM et par la complexité élevée des processus de codage et de décodage.

Le codage LZW, malgré un certain flou au niveau des droits d'utilisation, est très utilisé pour la compression sans pertes de données des fichiers informatiques ainsi que pour le modem. Il est, en moyenne, plus performant que les deux précédents à condition que le message à coder soit suffisamment long. Son apparente simplicité masque toutefois une complexité plus élevée que pour l'algorithme de Huffman.

A titre d'exemple, le tableau suivant indique les taux de compression (dans le cas de fichiers informatiques connus) obtenus à l'aide de différents algorithmes. PKZIP et ARJ utilisent un codage LZW associé à un codage de Huffman (pour comprimer les adresses).

nom du fichier	taille originel	Huffman	PKZIP	ARJ
INSTALL.EXE	205152	168907	95216	89833
INSTALL.HLP	118651	86616	53695	46168
SYSTEM.INI	1174	951	775	648
WINHELP.EXE	208624	161163	89266	82639
WINFILE.EXE	107632	82376	50557	47112
ECHECS.BMP	153718	46011	17296	14506
WIN.INI	2737	2174	1681	1452
TAUX	0 %	31 %	61 %	64 %

1.3 Le codage de Huffman dans JPEG

1.3.1 Codage de la position du bit de poids fort

Nous allons maintenant considérer des symboles représentant une valeur numérique positive ou négative. Si le symbole est codé sur 16 bits, il faudrait une table de Huffman comportant 65536 codes à longueur variables. Cela pose deux problèmes :

1. la taille de la table que le codeur doit envoyer au décodeur est beaucoup trop grande,
2. plus la table est grande, plus la vitesse du codage et du décodage est faible.

Il faut rappeler que le code de Huffman n'est efficace que quand il y a des différences élevées de probabilité entre les valeurs à coder. Si les valeurs à coder sont équiprobables (distribution aléatoire), son efficacité est nulle. Plutôt que de coder tous les bits de l'échantillon, on ne va coder avec Huffman que la position du bit de poids fort. Les bits restants sont supposés à distribution aléatoire et donc codés en binaire naturel (ou complément à 2, CA2).

Exemple : soit un coefficient que l'on appellera DIFF codé sur 12 bits (-2047, +2047). On lui assigne un code de Huffman, représentant la position du bit de poids fort, suivi de ses bits de poids faibles qui spécifient le signe et la valeur exacte de son amplitude. Les valeurs de DIFF, en CA2, sont groupées en 12 catégories.

catégorie SSSS	valeurs de DIFF
0	0
1	-1,1
2	-3,-2,2,3
3	-7..-4,4..7
4	-15..-8,8..15
5	-31..-16,16..31
6	-63..-32,32..63
7	-127..-64,64..127
8	-255..-128,128..255
9	-511..-256,256..511
10	-1023..-512,512..1023
11	-2047..-1024,1024..2047

Quand DIFF est positif, les SSSS bits de poids faibles de DIFF (les *extras-bits*) sont mis à la suite du code de Huffman. Quand DIFF est négatif, les SSSS bits de poids faibles de (DIFF-1) suivent le code de Huffman. Prenons l'exemple suivant :

- DIFF = $(85)_d$ ==> catégorie = 7 (exemple de code de Huffman : 11110)
  $(85)_d = (1010101)_b$ ==> code = 11110 1010101

- DIFF = $(-85)_d$ ==> (DIFF-1) = $(-86)_d = (10101010)_b$ en CA2
  ==> code = 11110 0101010

Le décodage s'effectue très rapidement. La reconnaissance du mot code de Huffman permet d'identifier la catégorie et le nombre d'extras-bits de la valeur à décoder. Si le MSB des extras-bits vaut 1, alors DIFF est positif et DIFF est égal à la valeur de ces extras-bits. Si le MSB des extras-bits vaut 0, alors DIFF est négatif et DIFF vaut, tous calculs effectués en CA2 : DIFF = limite basse de la catégorie + extras-bits. Prenons l'exemple suivant :

- mot-code reçu = 11110 1010101 ==> extras-bits = 1010101 catégorie = 7
  le MSB des extras-bits vaut 1 ==> DIFF > 0
  ==> DIFF = $(1010101)_b = (85)_d$

- mot-code reçu = 11110 0101010 ==> extras-bits = 0101010 catégorie = 7
  le MSB des extras-bits vaut 0 ==> DIFF < 0
  limite basse de la catégorie 7 = $(-127)_d = (10000001)_b$

$$\begin{array}{rr} & 10000001 \\ + & 0101010 \\ \hline & 10101011 \end{array}$$

D'où on tire DIFF = $(-85)_d$.

1.3.2 Spécification d'une table

Nous allons maintenant étudier la méthode permettant de générer les tables de Huffman. La génération de ces tables nécessite la connaissance préalable des probabilités d'apparition des

valeurs à coder. Dans JPEG, la méthode de Huffman traditionnelle a été adaptée pour répondre aux contraintes suivantes :

- les codes générés ne doivent pas être composés uniquement de 1,
- la longueur des codes est limitée à 16 bits,
- le codeur doit pouvoir envoyer au décodeur les tables en utilisant le moins de bits possible,
- les probabilités des valeurs à coder doivent être non nulles. Si l'analyse statistique préalable donne une probabilité nulle à une valeur, il faut la changer en une probabilité minimum (sinon, aucun code ne sera généré).

La détermination d'une table de Huffman de N valeurs se fait en trois étapes :

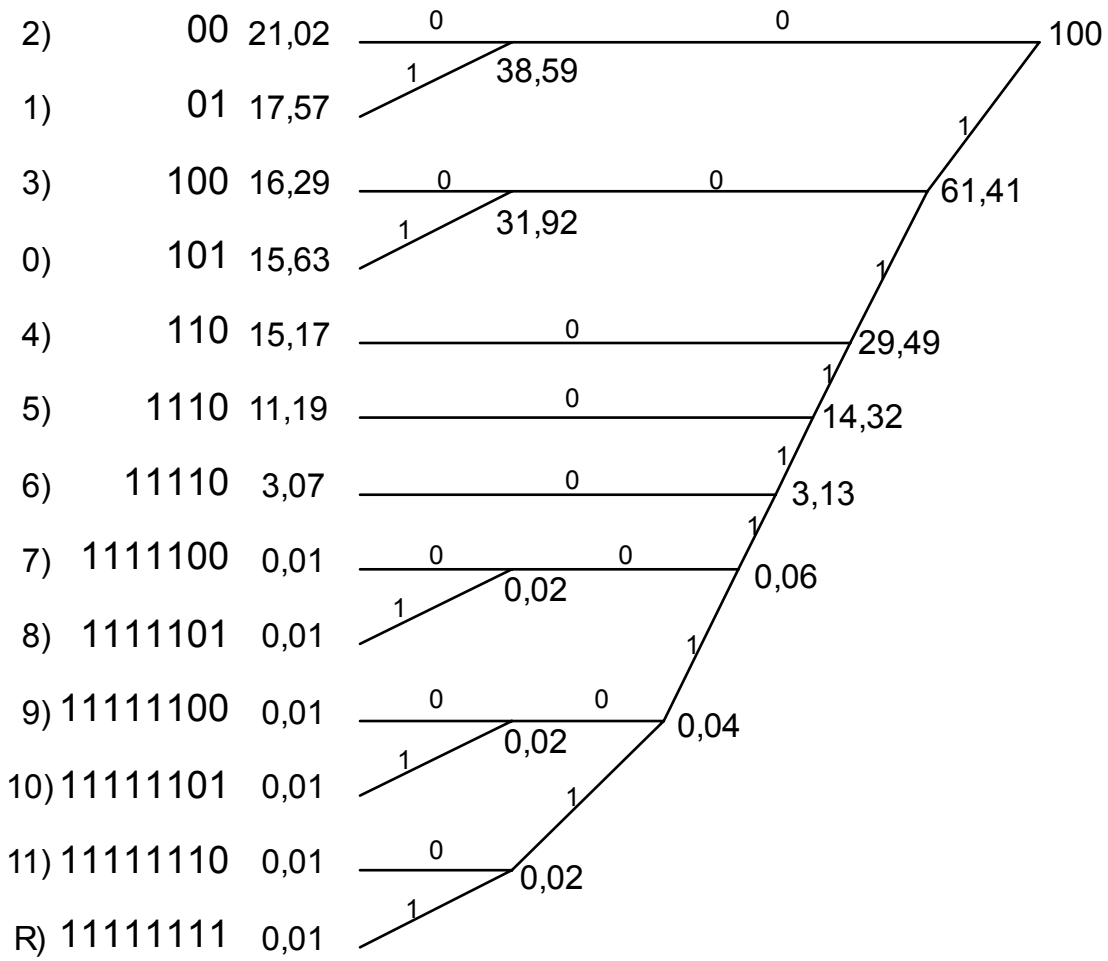
1. Par la méthode du tableau de Huffman, on trouve les codes et on génère la variable CODESIZE(0...N) qui contient la longueur des différents codes classés dans l'ordre des valeurs à coder. La valeur N de CODESIZE (qui contient N+1 codes) est la valeur de code composée uniquement de 1 qui a été associée à une valeur à coder factice de probabilité minimum. Elle ne sera pas utilisée par la suite.
2. On limite la longueur des codes à 16 bits et on génère la variable BITS(1...16) qui contient le nombre de codes existant pour chacune des 16 longueurs de code possibles.
3. On génère la variable HUFFVAL(0...N-1) qui indique à quelles valeurs à coder doivent être associés les codes de Huffman.

Les trois variables générées ont une largeur d'un octet par case. Nous verrons plus loin que les deux variables BITS et HUFFVAL permettent de spécifier complètement le code. Ce sont ces deux variables que le codeur envoie au décodeur. Si on envoyait au décodeur une table de codes (limités à 16 bits) de N valeurs de la manière la plus simple possible, cela prendrait 2 octets pour une valeur de code plus 1 octet pour sa longueur, soit au total 3N octets. En envoyant BITS et HUFFVAL, on ne transmet que N+16 octets.

Exemple 1: calcul des valeurs de CODESIZE, BITS et HUFFVAL pour une valeur DIFF. Soient les catégories suivantes à coder :

catégories	0	1	2	3	4	5	6	7	8	9	10	11
probabilités	15,63	17,57	21,02	16,29	15,17	11,19	3,07	0,01	0,01	0,01	0,01	0,01

Il faut noter que les probabilités des catégories 7 à 11 étaient nulles lors de l'analyse statistique, mais que l'on a quand même souhaité y associer des codes. On a donc mis ces probabilités au minimum. On détermine la table de Huffman par une méthode classique. La treizième valeur R est factice. Son mot-code est 11111111 et ne sera plus utilisé par la suite.



CODESIZE contient 13 valeurs de longueur de code classées dans l'ordre des catégories. Par exemple, la catégorie 5 a un code de longueur 4.

CODESIZE = 3, 2, 2, 3, 3, 4, 5, 7, 7, 8, 8, 8, 8

Avec seulement 12 catégories à coder, tous les codes ont des longueurs inférieures à 16 bits. Il n'y a donc pas de réduction à faire. BITS contient 16 valeurs de nombre de code par longueur. Par exemple, il y a 2 codes de longueur 2 et 2 codes de longueur 7.

BITS = 0, 2, 3, 1, 1, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0

HUFFVAL contient la catégorie de la valeur à coder (de 0 à 11) auquel doivent être associé les codes classés par longueur croissante. Par exemple, le troisième code doit être associé à la catégorie 0.

code de Huffman	HUFFVAL
00	1
01	2
100	0
101	3
110	4
1110	5
11110	6
1111100	7
1111101	8
11111100	9
11111101	10
11111110	11

Les codes ne sont pas nécessairement attribués dans l'ordre de la méthode de Huffman comme le montre le tableau suivante. Par contre, les longueurs de code correspondent et la longueur moyenne d'un symbole reste la même. L'entropie vaut 2,68 [sh/sym], alors que la longueur moyenne d'un symbole vaut 2,7869 bits.

catégorie	attribution Huffman	attribution JPEG
0	101	100
1	01	00
2	00	01
3	100	101
4	110	110
5	1110	1110
6	11110	11110
7	1111100	1111100
8	1111101	1111101
9	11111100	11111100
10	11111101	11111101
11	11111110	11111110

Exemple 2: réduction à 16 bits d'une table. Avant réduction, on a une variable BITS contenant le nombre de code pour des longueurs comprises entre 1 et 32, 32 étant une limite arbitraire.

BITS(1...32) = 0,2,2,1,2,3,6,3,5,6,4,7,0,1,5,1,0,0,0,0,0,13,102,0,0,0,0,0,0,0,0,0

Pour pouvoir réduire la longueur des codes, il va falloir récupérer des codes de longueur inférieure pour gagner de la place. L'efficacité du code de Huffman est donc diminuée dans l'opération. Après réduction, BITS contient le nombre de codes pour des longueurs comprises entre 1 et 16.

BITS(1...16) = 0,2,2,1,2,3,6,3,5,6,3,2,0,0,0,127

On remarque que la réduction nous a fait perdre un code de longueur 11 bits, 5 codes de longueur 12 bits, un code de longueur 14 bits et 5 codes de longueur 15 bits. En contrepartie, on a gagné 126 codes de longueur 16 bits. L'entropie vaut 3,3572 [sh/sym], la longueur moyenne d'un symbole sans réduction vaut 3,3944 bits, alors que la longueur moyenne d'un symbole avec réduction est égale à 3,4025 bits. La perte de compression est d'environ 0,25%, ce qui est négligeable.

1.3.3 Le codage

Nous allons maintenant voir que la variable BITS suffit pour déterminer les codes, et que la variable HUFFVAL suffit pour associer ces codes aux valeurs à coder. On peut retrouver la liste des codes et des longueurs de code classée par ordre de longueur croissante en deux étapes :

1. à partir de BITS, on génère la variable HUFFSIZE(1...16) qui contient les longueurs des codes dans l'ordre croissant,
2. on détermine, à partir de HUFFSIZE, la variable HUFFCODE(1...16) qui contient les codes classés par ordre de longueur croissante. Les codes sont calculés de la manière suivante :
   - Le premier code vaut 0,
   - Si le code suivant est de même longueur, on le détermine en ajoutant 1 au code précédent . Si la longueur du code suivant vaut M bits de plus, on le calcule en ajoutant 1 au code précédent puis en faisant un décalage logique M fois à gauche.
   - On recommence cette opération autant de fois qu'il y a de codes à calculer.

Pour pouvoir coder les différentes valeurs, on va ordonner les codes et les longueurs de code dans l'ordre des valeurs à coder grâce à HUFFVAL. Cela nous donnera deux variables EHUFCO et EHUFSI qui seront utilisées dans le codeur.

Reprenons l'exemple 1 vu précédemment :

BITS = 0, 2, 3, 1, 1, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0

HUFFVAL = 1, 2, 0, 3, 4, 5, 6, 7, 8, 9, 10, 11

La variable HUFFSIZE se déduit facilement de BITS

HUFFSIZE = 2, 2, 3, 3, 3, 4, 5, 7, 7, 8, 8, 8

La variable HUFFCODE se calcule en suivant la méthode exposée dans le tableau suivant :

opérations à effectuer	codes binaires générés : HUFFCODE
code 1 : 0 et de longueur 2	00
+1	01
+1 puis décalage à gauche 1 fois	100
+1	101
+1	110
+1 puis décalage à gauche 1 fois	1110
+1 puis décalage à gauche 1 fois	11110
+1 puis décalage à gauche 2 fois	1111100
+1	1111101
+1 puis décalage à gauche 1 fois	11111100
+1	11111101
+1	11111110

De HUFFVAL, HUFFCODE et HUFFSIZE, on déduit les variables EHUFCO et EHUFSI contenant les codes et les longueurs des codes classés par ordre des catégories croissantes :

catégorie	EHUFSI	EHUFCO
0	3	100
1	2	00
2	2	01
3	3	101
4	3	110
5	4	1110
6	5	11110
7	7	1111100
8	7	1111101
9	8	11111100
10	8	11111101
11	8	11111110

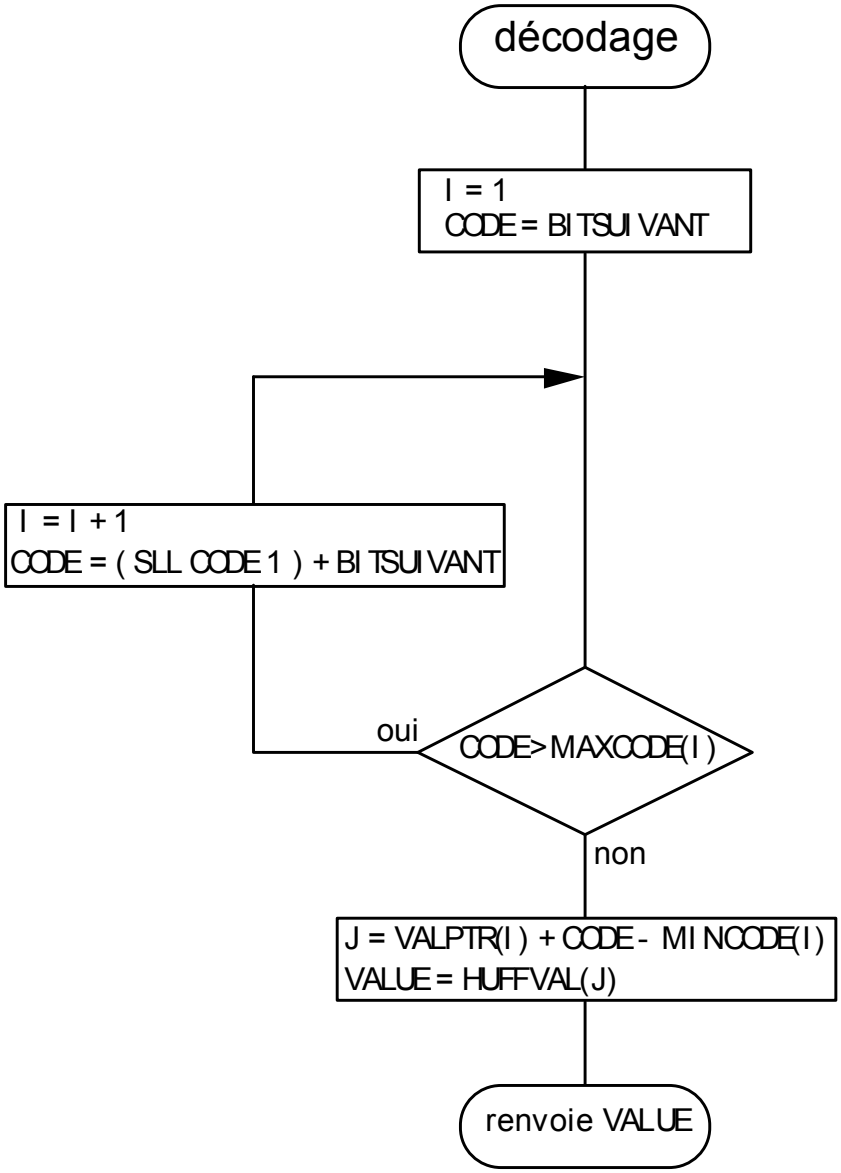
1.3.4 Le décodage

Le décodage d'un code de Huffman est une partie délicate. Il faut vérifier dans la liste des codes, à chaque fois que l'on reçoit un bit supplémentaire, si on termine un mot-code. En effet, le code est à décodage unique et de longueur variable et on ne connaît pas la longueur de ce code avant de l'avoir décodé. Si la table contient un grand nombre de valeurs, le temps de recherche devient prohibitif et on doit utiliser la technique des arbres binaires. Cet algorithme de recherche est très efficace, mais il est assez lourd à mettre en œuvre, surtout en langage assembleur. Nous allons voir dans ce chapitre que l'on peut procéder au décodage grâce à une simple boucle à partir des variables HUFFCODE et BITS.

Le décodage s'effectue en deux étapes :

1. à partir de HUFFCODE et BITS, on génère trois variables :
   - VALPTR(1...16) contient les pointeurs qui indique l'emplacement du début des codes de longueur n (n allant de 1 à 16) dans HUFFCODE. S'il n'y a pas de code de longueur n, alors VALPTR = 0.
   - MINCODE(1...16) contient la valeur minimum des codes de longueur n. S'il n'y a pas de code de longueur n, alors MINCODE = 0.
   - MAXCODE(1...16) contient la valeur maximum des codes de longueur n. S'il n'y a pas de code de longueur n, alors MAXCODE = -1 = $(FFFF)_{hca2}$.

2. avec ces trois tables, le décodage se fait très facilement grâce à l'algorithme de la page suivante. La procédure BITSUIVANT extrait le bit suivant de la chaîne des données comprimées. La fonction SLL *variable* 1 indique qu'il faut faire un décalage logique à gauche sur *variable* une fois. La variable VALUE contient le rang du code dans la liste des valeurs codées.



Reprenons l'exemple 1 vu précédemment. La détermination des trois tables (voir : tableau suivant) ne pose pas de problèmes particuliers. On voit que :

- les codes de longueur 2 commencent en position 0 dans la liste des codes classés par ordre des longueurs croissantes. La valeur minimum vaut 00, la valeur maximum vaut $(01)_b$.
- il n'y a pas de codes de longueur 6 (MAXCODE = -1).

- il n'y a qu'un code de longueur 5 (MINCODE = MAXCODE = $(11110)_b$) qui se trouve en position 6 dans HUFFCODE.

longueur	BITS	VALPTR	MINCODE	MAXCODE
1	0	0	0	-1
2	2	0	00	01
3	3	2	100	110
4	1	5	1110	1110
5	1	6	11110	11110
6	0	0	0	-1
7	2	7	1111100	1111101
8	3	9	11111100	11111110
9	0	0	0	-1
10	0	0	0	-1
11	0	0	0	-1
12	0	0	0	-1
13	0	0	0	-1
14	0	0	0	-1
15	0	0	0	-1
16	0	0	0	-1

Prenons, par exemple, le mot-code 101 à décoder. En suivant l'algorithme de décodage, on obtient la séquence suivante :

CODE = 1 > MAXCODE(1) = -1

CODE = 10 > MAXCODE(2) = 01

CODE = 101 < MAXCODE(3) = 110

==> J = VALPTR(3) + CODE - MINCODE(3) = 2 + 5 - 4 = 3

⇨ catégorie = HUFFVAL(3) = 3

2 La compression du son

2.1 Introduction

Avec l'émergence des nouveaux marchés de l'audiovisuel numérique et du multimédia, le développement des techniques de traitement numérique du signal devient un enjeu économique essentiel pour les entreprises. Or le traitement numérique du signal a très vite été confronté à un problème d'encombrement trop important du signal numérisé dans les chaînes de transmission (encombrement spectral) ou sur les supports de stockage (encombrement en quantité de bits). Pour économiser les ressources des supports de transmission ou de stockage, il est donc nécessaire de réduire le débit numérique en ne conservant que les informations utiles.

Pour le codage sonore, les codeurs « perceptifs » permettent une réduction de débit numérique importante. Leur principe consiste à utiliser un modèle psycho-acoustique qui évalue, pour le récepteur final qu'est l'oreille, les informations inutiles et le degré de redondance des informations présentes dans le signal sonore. Le modèle psycho-acoustique se réfère au comportement du système auditif de l'Homme qui fait que la suppression de certaines informations sonores ne modifie pas la qualité subjective finale de l'écoute. Le modèle psycho-acoustique détermine, dans le domaine fréquentiel, le seuil de masquage en dessous duquel une composante est estimée inaudible. Ce seuil est utilisé aussi dans une étape de quantification pour déterminer le niveau maximal de bruit de quantification que l'on peut générer sans que celui-ci soit audible.

Après un bref rappel de l'anatomie de l'oreille chez l'homme, nous décrirons le comportement acoustique du système auditif. Pour déterminer ce comportement, il faut faire appel à l'appréciation humaine. Des expériences psycho-acoustiques, au cours desquelles un auditeur doit détecter un son en présence d'un autre, ont été élaborées pour mettre en évidence les phénomènes liés à l'audition. Nous exposerons dans ce chapitre les résultats de ces expériences et les interprétations qui en découlent. Nous aborderons ensuite la modélisation mathématique de ces phénomènes complexes et leurs exploitations dans le domaine du codage sonore, en particulier pour les codeurs fréquentiels ou dit « perceptifs ». Nous présenterons comme codeur « perceptif » la norme MPEG-1 audio qui met en œuvre deux modèles psycho-acoustiques basés sur les propriétés physiologiques et mécaniques de

l'oreille. La couche 3 de MPEG1 audio, qui n'est pas utilisée en télévision numérique, ne sera pas abordé dans ce cours.

2.2 L'audition

L'oreille se décompose en 3 parties, l'oreille externe, l'oreille moyenne et l'oreille interne avec chacune un rôle spécifique dans le système auditif (figure 2-1).

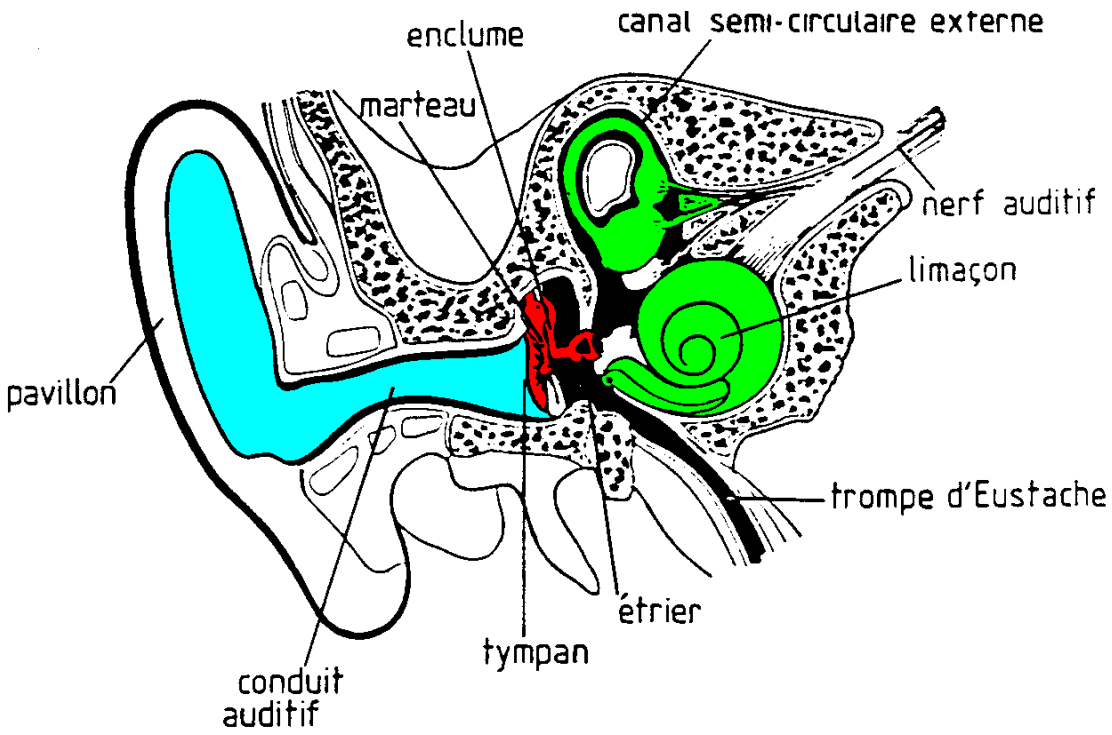


Figure 2-1 : Structure du système auditif chez l'homme

2.2.1 L'oreille externe

L'oreille externe se compose du pavillon et du canal auditif :

- Le pavillon est la partie visible de l'oreille externe. Il canalise les ondes sonores vers le conduit auditif et nous permet de localiser dans l'espace les sources sonores.
- Le canal auditif (ou méat) est le siège de résonances renforçant les fréquences comprises entre 2000 Hz et 5000 Hz avec un maximum autour de 3700 Hz. Il joue un rôle d'amplificateur de pression acoustique. Associé au pavillon, il contribue au filtrage du signal acoustique dans la bande audible.

2.2.2 L'oreille moyenne

L'oreille moyenne est constituée du tympan, de la chaîne des osselets et de la trompe d'Eustache (figure 2-2).

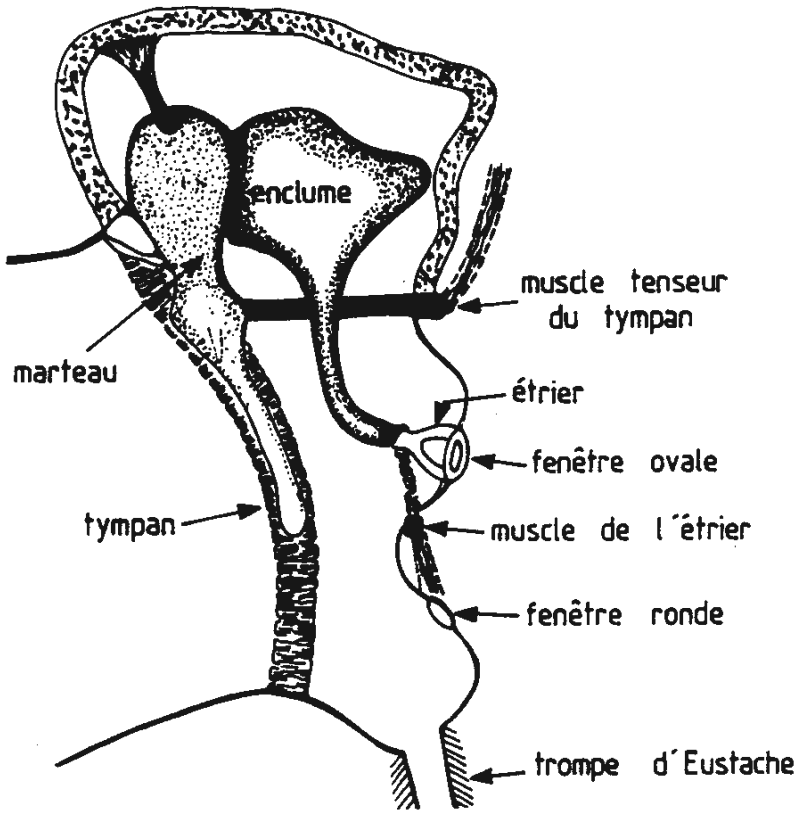


Figure 2-2 : L'oreille moyenne

- Le tympan est une membrane élastique très mince séparant le canal auditif de l'oreille moyenne. Il sert de transducteur mécanique pour les vibrations acoustiques provenant du conduit auditif.
- La chaîne des osselets se compose du marteau solidaire du tympan, de l'enclume et de l'étrier. Son rôle est de transmettre les vibrations du tympan à la fenêtre ovale, entrée de l'oreille interne. Les osselets améliorent la transmission des vibrations entre deux milieux (gaz et liquide) et deux sections différentes (tympan et fenêtre ovale).
- La trompe d'Eustache assure le rôle d'égalisateur de la pression acoustique sur les deux faces du tympan

La combinaison de l'oreille externe et moyenne se comporte comme un filtre passe bande. La figure 2-3 donne le gabarit de ce filtre.

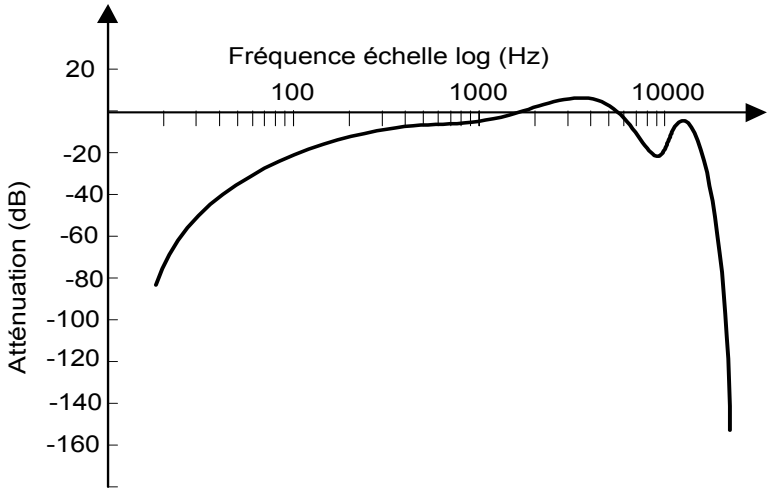


Figure 2-3 : Gabarit du filtre passe bande modélisant la combinaison de l'oreille externe et moyenne

2.2.3 L'oreille interne

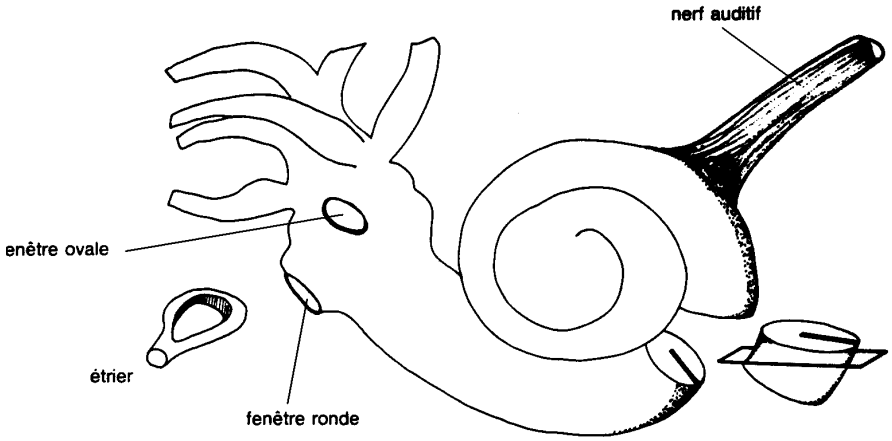
L'oreille interne est une cavité osseuse (os du rocher) remplie de liquide constituée du vestibule et du limaçon ou cochlée :

- Le vestibule est la cavité centrale contenant les organes de l'équilibre (canaux circulaires) qui ne jouent aucun rôle dans l'audition.
- Le limaçon ou cochlée est une spirale de 30 mm faisant deux tours et demi autour d'un cône osseux appelé la columelle. La cochlée est le dernier organe du système auditif jouant un rôle essentiel dans le mécanisme de l'audition.

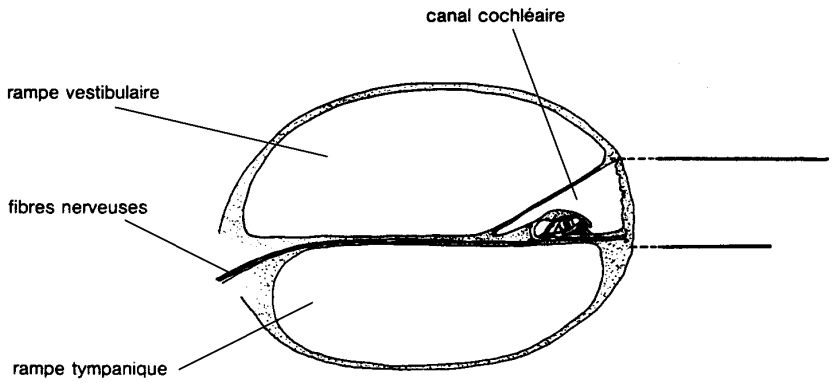
2.2.4 La cochlée

A l'extrémité basale de la cochlée se trouvent deux fenêtres recouvertes de fines membranes flexibles. L'ouverture supérieure est appelée fenêtre ovale et l'étrier se rattache à sa membrane. La seconde ouverture est appelée fenêtre ronde et compense la pression appliquée par l'étrier à la fenêtre ovale évitant ainsi une compression dangereuse. La coupe longitudinale de la cochlée (Figure 2-4) montre 3 rampes. Les rampes vestibulaires et tympaniques contiennent un fluide peu compressible, la périlymphe, et la rampe centrale ou canal cochléaire est empli d'un autre fluide, l'endolymphe. Les mouvements de la périlymphe sous l'action de la pression sur la membrane de la fenêtre ovale provoquent des déformations sur le canal cochléaire. La membrane basilaire, support de l'organe de Corti qui contient une multitude de cellules ciliées sensibles, provoque par l'intermédiaire de la membrane tectoriale une excitation des cils sous l'action des déformations du canal cochléaire. Ces cils stimulent à

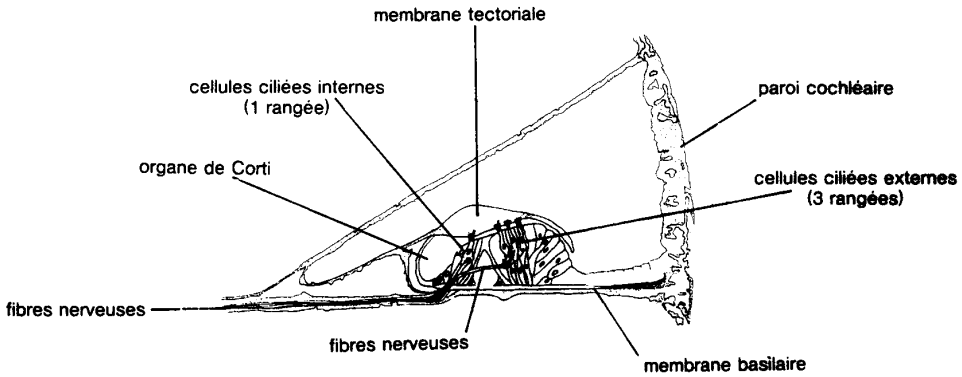
leurs tours les fibres du nerf auditif et transmettent la hauteur du son (tonie) et sa force (sonie) au cerveau selon la position et l'amplitude de la zone de vibration.



COCHLÉE



COUPE DE LA COCHLÉE



LE CANAL COCHLÉAIRE

Figure 2-4 : La cochlée (vue, coupe et détail)

2.2.5 Principe du mécanisme de l'audition

Les vibrations sonores parviennent au tympan sous la forme de variations de pression captées par l'oreille externe. Ces variations de pression sont transformées en vibrations mécaniques par la chaîne des osselets et transmises au fluide de la cochlée à travers la fenêtre ovale. La membrane basilaire ainsi que l'organe de Corti solidaire de la membrane sont affectés par ces vibrations mécaniques et ce sont les cellules ciliées de l'organe de Corti qui remplissent le rôle de transducteur, transformant l'énergie mécanique en énergie électrique. Les figures 2-5 et I-6 montrent la propagation d'une onde le long de la membrane basilaire et les enveloppes résultantes pour quatre fréquences différentes. Pour les basses fréquences, le déplacement maximal se situe près de l'extrémité apicale de la cochlée, l'hélicotrème et pour les hautes fréquences, il se rapproche de l'extrémité basale. Dans le cas de sons complexes, la membrane basilaire est soumise à des maxima en différents points se déplaçant dans le temps.

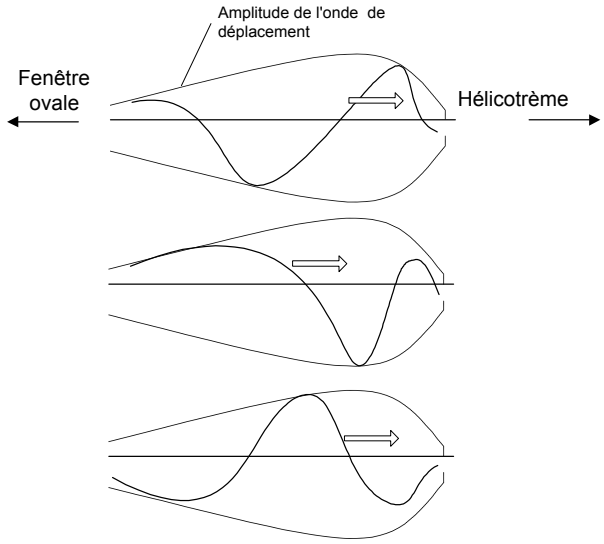


Figure 2-5 : Propagation d'une onde le long de la membrane basilaire

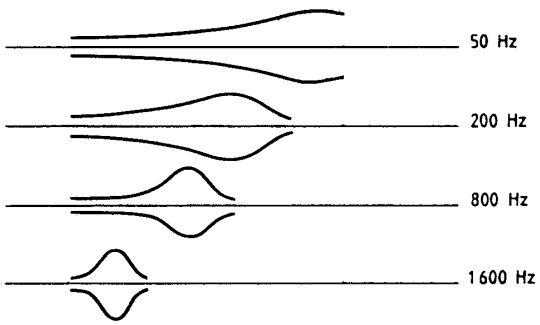


Figure 2-6 : Evolutions de la position des ondes en fonction de la fréquence dans la membrane basilaire

La cochlée a un comportement proche de l'analyseur de Fourier. Elle réalise en quelque sorte une analyse fréquentielle du son.

2.3 Les propriétés acoustiques de l'oreille

L'étude des propriétés psycho-acoustiques de l'oreille entreprise depuis les années 1940 par Fletcher et surtout par Zwicker et Feldtkeller a permis de décrire le comportement de l'oreille. Ces études ont mis en évidence les valeurs limites du niveau acoustique et leurs évolutions dans la zone audible, les filtres auditifs amenant à la notion de bandes critiques ainsi que les phénomènes de masquage ou seuil d'audition masqué.

2.3.1 Le seuil d'audition absolu

Le seuil d'audition absolu précise la limite inférieure en niveau des sons audibles ainsi que leurs limites inférieures et supérieures en fréquence. En règle générale, on donne comme valeur de référence pour la plus petite fréquence audible la valeur de 20 Hz et 20 kHz pour la plus grande fréquence audible. Le plus petit niveau acoustique audible correspond à 0 dB. La courbe obtenue est proche du gabarit inversé du filtre passe bande modélisant l'oreille externe et moyenne. La sensibilité maximale est entre 2 kHz et 5 kHz. La limite supérieure en niveau se situe autour d'un niveau acoustique de 130 dB. Le seuil d'audition absolu est obtenu lorsque le son test est la seule stimulation sonore apparaissant dans un silence total. La figure 2-7 montre ces deux courbes.

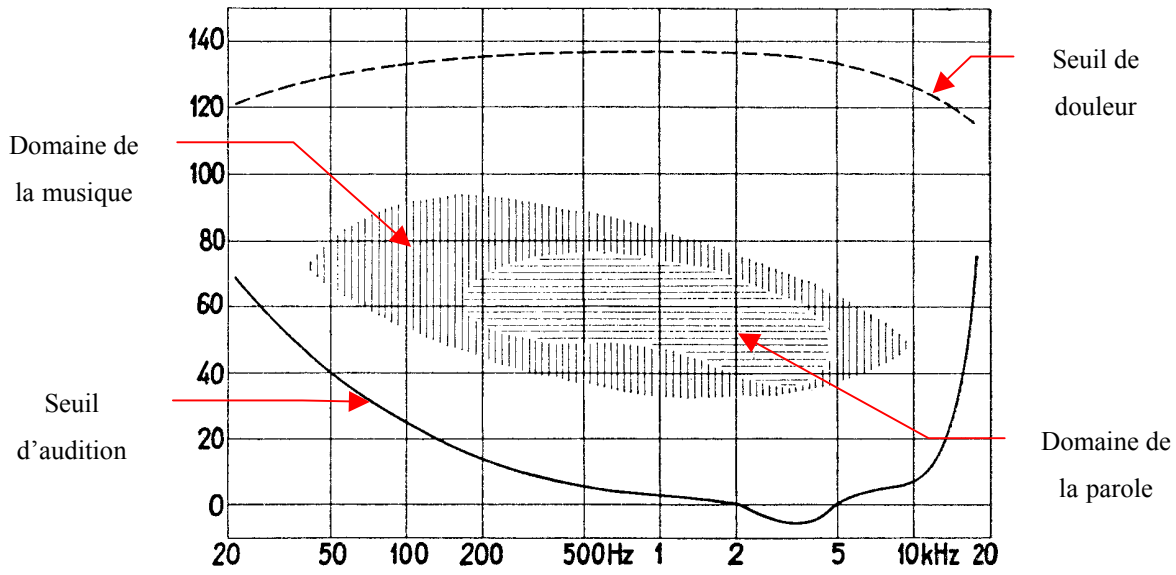


Figure 2-7 : Aire d'audition

2.3.2 Le phénomène de masquage

D'après le fonctionnement de la cochlée, on conçoit que lorsque deux sons sont présents à l'entrée du système auditif, l'un peut être masqué par l'autre selon leurs amplitudes et leurs fréquences respectives. Les expériences psycho-acoustiques ont permis de quantifier le niveau de pression acoustique pour qu'un son test devienne inaudible en présence d'un son parasite ou masquant. Les expériences de masquage ont été réalisées avec comme bruit masquant un bruit blanc à bande étroite.

La figure 2-8 donne les courbes d'effet de masque d'un bruit blanc large bande (densité spectrale constante entre 20 Hz et 20 kHz) de pression acoustique I_{wr} comme son masquant et d'une série de sons purs couvrant la zone audible pour les sons tests. On remarque de ces courbes que les seuils de détection restent constants de 20 Hz à 500 Hz puis croissent de 10 dB/décade jusqu'à atteindre le seuil d'audition absolu. L'allure de ces courbes s'explique par le fait que l'oreille prend en compte la puissance du bruit blanc sur des bandes de fréquences étroites et non pas sur une bande unique s'étalant de 20 Hz à 20 kHz.

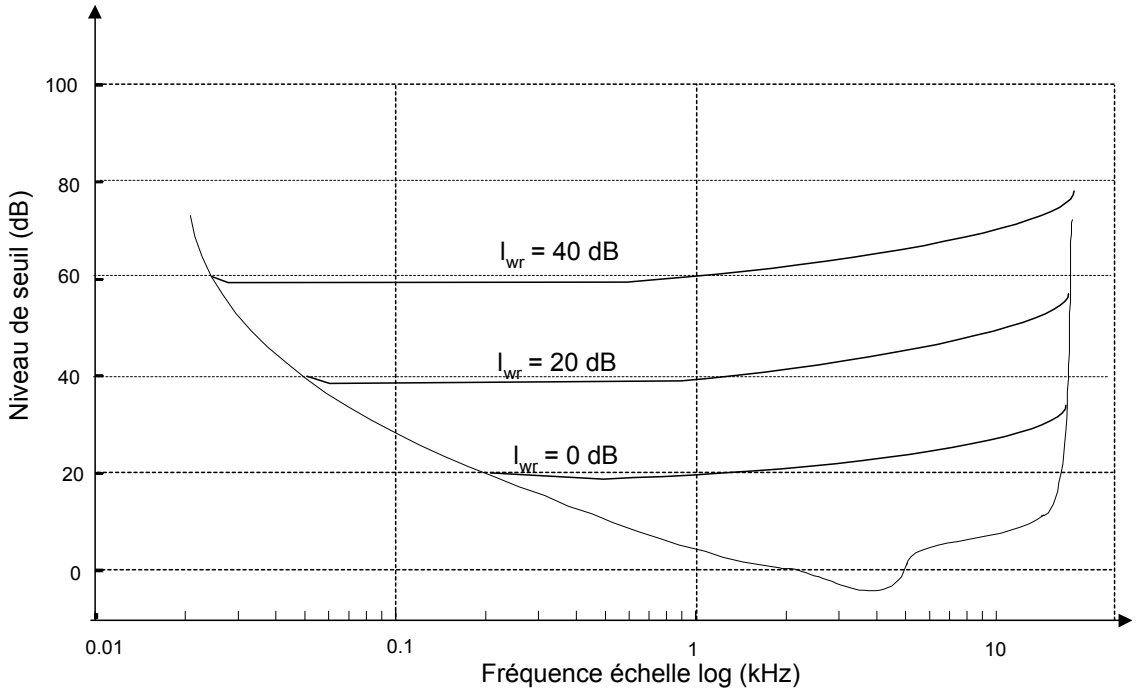


Figure 2-8 : Seuils d'audition en présence de bruit blanc

La figure 2-9 montre les courbes d'effet de masque de plusieurs bruits à bande étroite (100 à 200 Hz de largeur) masquants. Nous remarquons que les courbes présentent un maximum à la fréquence centrale des bruits à bande étroite inférieure de 4 dB au niveau acoustique des bruits masquants. Cette valeur de 4 dB représente la capacité de l'oreille à détecter un son pur dans un bruit et se nomme **taux de masquage**. De plus la pente vers les hautes fréquences dépend du niveau acoustique du son masquant et présente un deuxième maximum pour les forts niveaux acoustiques. Une faible variation de la fréquence centrale du bruit masquant entraîne une translation de la courbe de masquage. Cette expérience montre le comportement non linéaire de l'oreille dans le domaine fréquentiel.

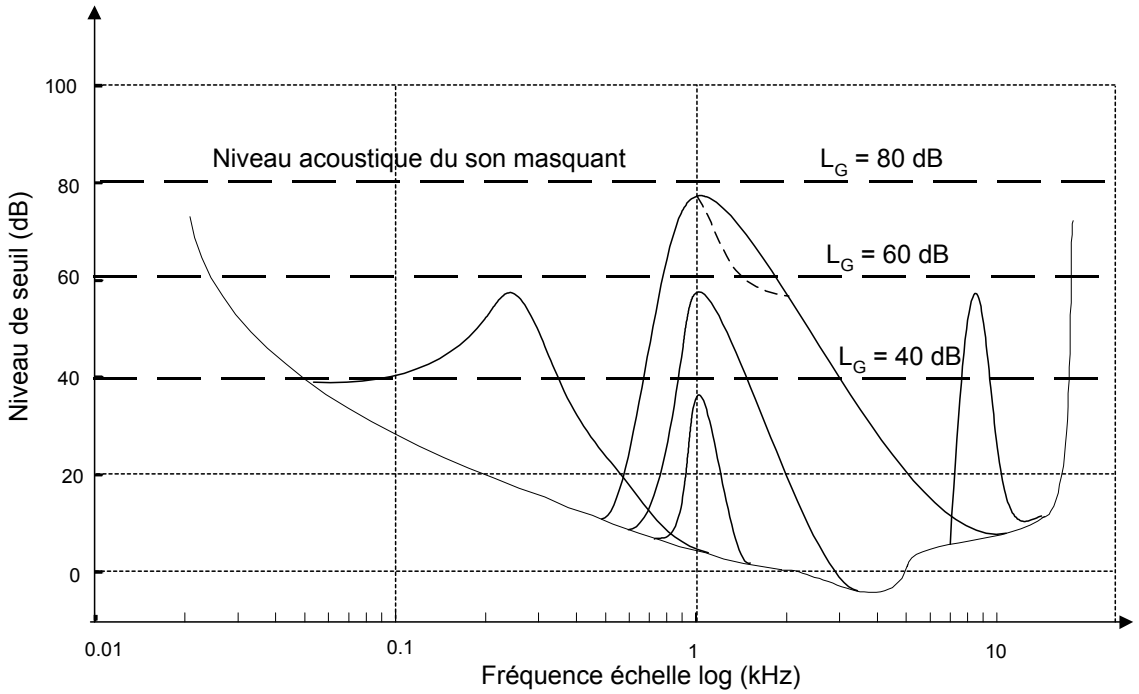


Figure 2- 9 : Courbe d'effet de masque de bruits à bande étroite

2.3.3 Les bandes critiques

Les expériences de masquage ont montré que l'oreille prend en compte le bruit masquant non pas sur l'ensemble de la bande audible mais sur des bandes de fréquences étroites correspondantes aux composantes fréquentielles proches de la fréquence du signal test. Ces bandes de fréquences sont appelées **bandes critiques** et leurs largeurs sont d'environ 100 Hz pour les fréquences centrales inférieures à 500 Hz et augmentent avec la fréquence au-delà de 500 Hz. Le tableau ci-dessous donne 24 bandes critiques adjacentes reconnues.

Numéro de la bande	Fréquence inférieure	Fréquence Supérieure	Largeur de la bande critique
1	20	100	80
2	100	200	100
3	200	300	100
4	300	400	100
5	400	510	110
6	510	630	120
7	630	770	140
8	770	920	150
9	920	1080	160
10	1080	1270	190
11	1270	1480	210
12	1480	1720	240
13	1720	2000	280
14	2000	2320	320
15	2320	2700	380
16	2700	3150	450
17	3150	3700	550
18	3700	4400	700
19	4400	5300	900
20	5300	6400	1100
21	6400	7700	1300
22	7700	9500	1800
23	9500	12000	2500
24	12000	15500	3500

Tableau 2-1 : Table des bandes critiques

En réalité, n'importe quelle fréquence est la fréquence centrale d'une bande critique. Il faut se représenter ce système périphérique comme une série de filtre ayant des fréquences centrales très voisines les unes des autres et dont les bandes passantes se recouvrent largement.

La décomposition du spectre de fréquences en bandes étroites est une propriété fondamentale de l'ouïe. L'ouïe peut former une bande critique en n'importe quel point de l'échelle des fréquences, c'est le son qui détermine le lieu fréquentiel où elles se forment.

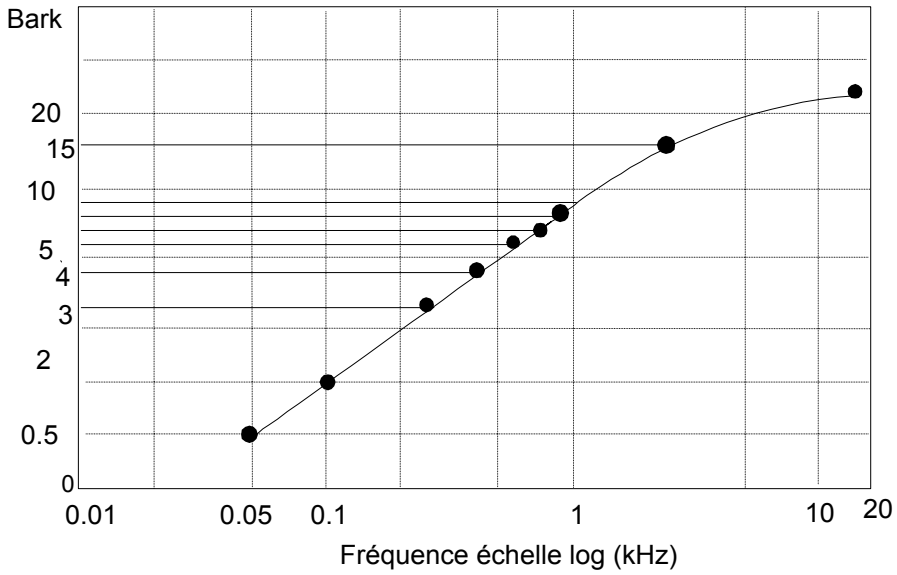


Figure 2-10 : Echelle des bandes critiques en fonction de la fréquence

Du fait de la non-linéarité de l'ouïe dans le domaine des fréquences, il est nécessaire de construire une échelle appelée « échelle basilaire » dans laquelle les bandes critiques sont représentées linéairement et possèdent la même largeur. Il a été défini une unité « **le Bark** » représentant le taux de bandes critiques sur l'échelle des fréquences. La figure 2-10 donne le taux de bandes critiques en fonction de la fréquence. En fait, ce terme de « taux de bandes critiques » est assez trompeur ; il s'agit simplement du nombre de bandes critiques compris entre 0 et la fréquence considérée. Il suffit de lire sur le tableau 2-1 le n° de la bande. Par exemple, à la fréquence de 1080 Hz correspond 9 bark.

Nous pouvons considérer que le système auditif se comporte comme un banc de filtres passe-bande. Ces filtres peuvent se former autour de n'importe quelles fréquences du son. Ils sont appelés « les filtres auditifs », leur bande passante est « la bande critique » et leur échelle est le Bark.

1 Bark = 1 largeur de bande critique

2.3.4 L'excitation

L'excitation est une notion essentielle pour expliquer la perception des variations des amplitudes et des fréquences. Elle correspond à la distribution de la stimulation sur le nerf auditif et représente la quantité d'énergie qui active les connexions nerveuses.

L'excitation se déduit des courbes de masquage en ajoutant le taux de masquage (de -2 à -6 dB pour les bruits à bande étroite et environ -24 dB pour les sons purs). Alors que l'excitation ou la courbe de masquage se modifie lorsqu'on la représente dans une échelle fréquentielle linéaire ou logarithmique, la forme de l'excitation ou de la courbe de masquage ne change pas dans le domaine des Bark quelle que soit la fréquence centrale du signal masquant.

Nous savons que la membrane basilaire est équivalente à un banc de filtres (les filtres auditifs) dont les fréquences centrales sont proches les unes des autres et dont les bandes passantes se recouvrent largement. Cette propriété permet d'expliquer la formation de l'excitation sur la membrane basilaire. Un son pur excite plusieurs filtres et la réponse de ces filtres correspond à l'excitation. La figure 2-11 montre la construction de l'excitation d'une sinusoïde de 1 kHz.

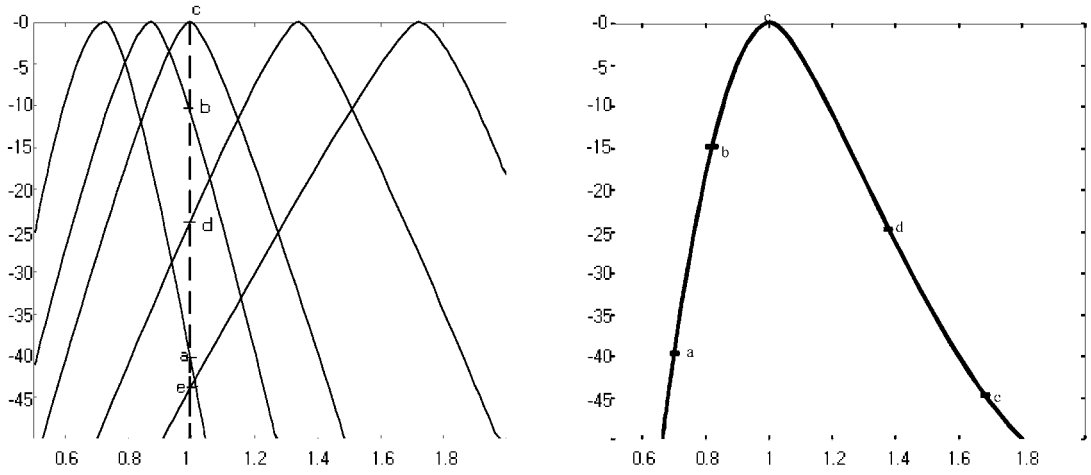
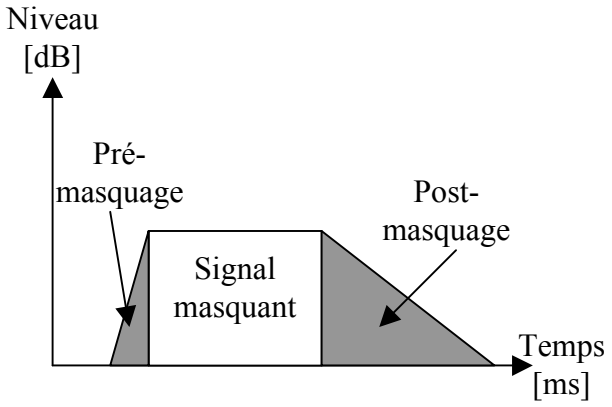


Figure 2- 11 : Construction de l'excitation d'une sinusoïde de 1 kHz

2.3.5 Propriétés temporelles

Les propriétés temporelles de l'oreille sont la résolution temporelle, l'intégration temporelle et le masquage temporel.

- La résolution temporelle du système auditif se réfère à sa capacité à détecter des changements dans les caractéristiques temporelles du signal sonore tels que de brefs silences, des clics ou une modulation d'amplitude. Le seuil de détection et de discrimination est de 2 à 3 ms pour les bruits à large bande. La résolution est de 22 ms à basse fréquence et de 3 ms à haute fréquence.
- L'intégration temporelle est la capacité de sommation d'informations sur une durée pour évaluer la détection et la discrimination d'un stimulus. La durée critique mesurée est de l'ordre de 200 ms.
- Le masquage temporel indique le comportement au cours du temps des courbes de masquage fréquentiel. On distingue deux types de masquage temporel, le masquage antérieur (pré masquage) et le masquage postérieur (post 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 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.



2.4 La norme MPEG

2.4.1 Présentation de la norme

La norme MPEG (Motion Picture Expert Group) constitue une norme internationale ISO (International Standard Organisation) proposant des méthodes de compression de l'image animée (vidéo) et du son associé (audio). La norme MPEG1 a été adoptée en 1992. Elle vise des applications de compression audio et vidéo synchronisées à des débits de 1,5 Mbit/s. La partie audio vise à la fois à réduire le débit et à atteindre la qualité du disque compact audio. Les fréquences d'échantillonnage du signal d'entrée peuvent être égales à 32 kHz (NICAM),

44.1 (Compact disc) et 48 kHz (enregistrement studio, DAT). Quatre modes de transmission sont prévus :

1) Stéréo : codage des deux voies gauche et droite de manière indépendante.
2) Joint stéréo : on exploite la redondance entre les deux voies gauche et droite pour réduire le débit.
3) Dual_channel : deux voies sons indépendantes (par exemple, son bilingue).
4) Mono : une seule voie son.

Selon l'application, différentes couches du système de codage, de complexités et de performances croissantes, peuvent être utilisées :

- La couche I. Elle utilise l'algorithme PASC (Precision Adaptative Subband Coding) développé par Philips pour la cassette audio numérique (DCC). Elle utilise un débit fixe compris entre 32 et 448 kbit/s. La qualité subjective de type compact disc nécessite 192 kbit/s par voie audio, soit 384 kbit/s en stéréo. Elle a pour principal avantage une relative simplicité d'implémentation du codeur et du décodeur, mais elle n'est quasiment plus utilisée aujourd'hui.
- La couche II. Son algorithme est connu sous le nom de MUSICAM, standard retenu pour le DAB (radio numérique). Le débit fixe peut être choisi entre 32 et 192 kbit/s et la qualité compact disc nécessite 128 kbit/s par voie audio, soit 256 kbit/s en stéréo. La complexité du codeur et du décodeur est plus élevée qu'avec la couche I, mais le débit est réduit de 30 à 50 %. C'est cette couche qui est utilisée pour la télévision numérique par satellite en Europe (système DVB).
- La couche III. L'algorithme ASPEC (Advanced SPectre Entropy Coding) est un développement plus récent que les autres algorithmes. Beaucoup plus complexe à implémenter, il utilise un débit variable et la qualité compact disc est obtenue avec 64 kbit/s par voie audio, soit 128 kbit/s en stéréo, c'est à dire un taux de compression environ deux fois plus élevé que la couche II. Il s'agit de la norme utilisée dans les fameux fichiers MP3 que l'on trouve un peu partout sur Internet. C'est là sa principale application.

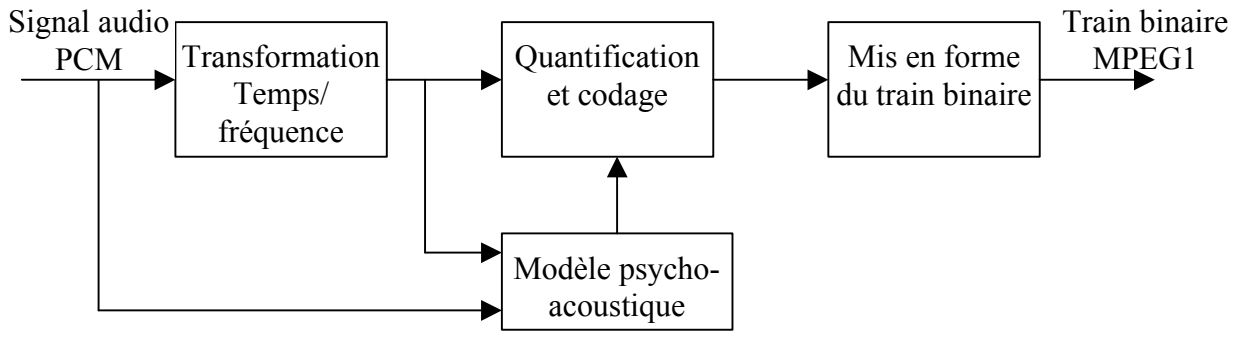
Pour résumé, les taux de compression obtenus pour une qualité subjective de type compact disc sont les suivants :

	débit	Taux de compression
Compact disc	2*44100*16 = 1378 kbit/s	1
MPEG1 couche I	384 kbit/s	3,6
MPEG1 couche II	256 kbit/s	5.4
MPEG1 couche III	128 kbit/s	10,8

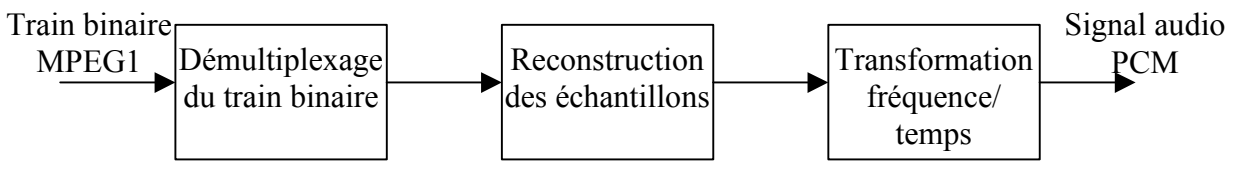
La norme MPEG2, plus récente, a été publiée en 1994. La partie audio, compatible avec MPEG1, apporte :

- le support du multicanal avec 5 voies haute fidélité plus une voie basse fréquence (5.1),
- le support multilinguistique avec 7 voies de commentaires possibles,
- le support des très bas débits jusqu'à 8 kbit/s,
- et l'extension des fréquences d'échantillonnage à 16, 22.05 et 24 kHz.

Il faut encore noter que la société Dolby a mis au point un algorithme concurrent de MPEG2 5.1 qui permet aussi de coder les voies sons (gauche, centre, droite, arrière gauche, arrière droite et basse) dans 384 kbit/s : le dolby digital anciennement appelé dolby AC-3. Les deux systèmes sont utilisés pour le DVD, mais seul MPEG2 5.1 a été retenu pour la télévision numérique en Europe dans le cadre du groupe DVB. La figure suivante représente le synoptique du codeur et du décodeur MPEG-1 audio.



(a)

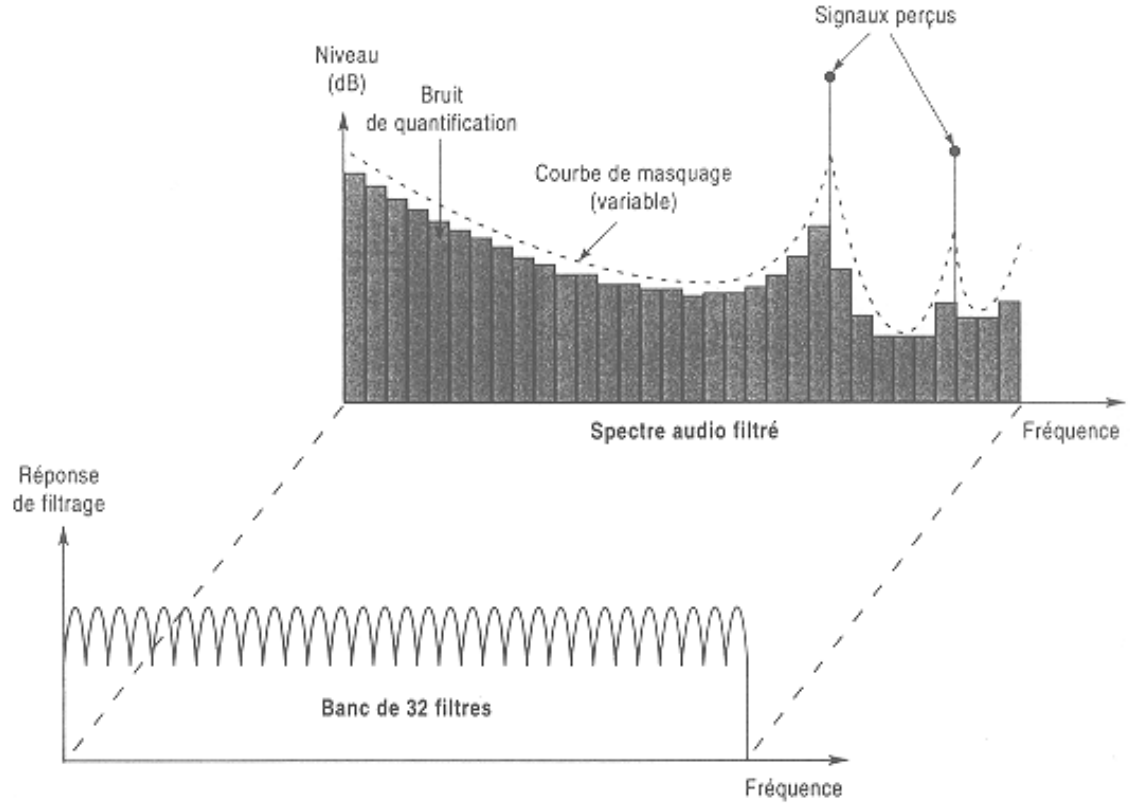


(b)

Figure 2-12 : Synoptique simplifié du codeur (a) et du décodeur (b) MPEG-1 audio

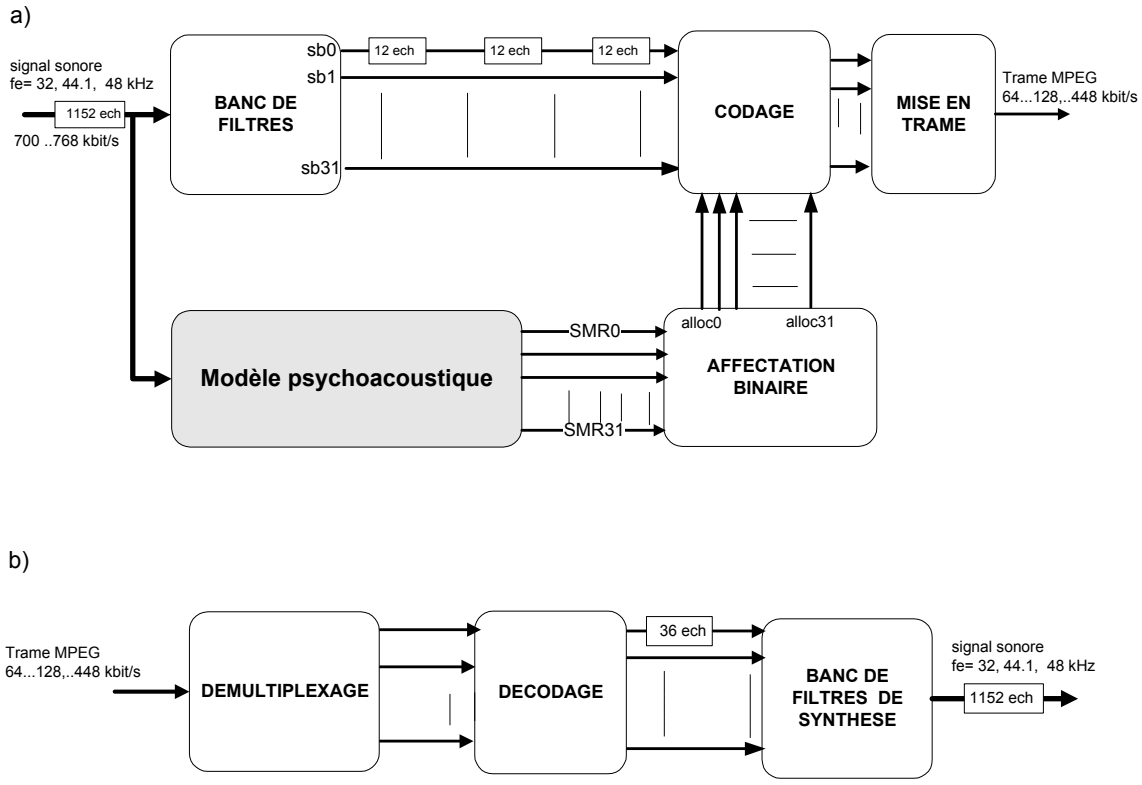
Les quatre parties du codeur sont :

- La transformation temps/fréquence qui permet de diviser le spectre du signal audio en plusieurs sous-bandes de fréquence.
- Le modèle psycho-acoustique qui utilise les propriétés psycho-acoustiques de l'oreille décrites précédemment pour calculer un niveau de masquage dans chacune de ces sous-bandes. Deux modèles psycho-acoustiques, modèle n°1 et modèle n°2, sont proposés par la norme MPEG.
- L'affectation binaire (ou répartition de bruit) qui détermine le nombre de bits alloué à chaque sous-bande à partir de la courbe de masquage afin que le bruit de quantification y reste inférieur au seuil d'audibilité. Les raies dont l'amplitude est inférieure au seuil ne sont pas codées. La figure suivante montre l'exemple de la couche II avec ses 32 sous-bandes :



- Le formatage du train binaire qui code et formate les sorties quantifiées du banc de filtres et ajoute des informations annexes nécessaires aux opérations de décodage.

Dans la suite de ce cours, nous ne nous intéresserons qu'à la couche II et au modèle 1. Le synoptique du codeur et du décodeur MPEG1 audio couche II sont les suivants :



Voyons plus en détail les différentes étapes.

2.4.2 La transformation temps/fréquence

Dans la couche 2 du codeur MPEG, le banc de filtres utilisé est un banc de filtres pseudo-QMF (Quadrature Mirror Filter) ayant 32 sous-bandes d'égales largeurs.

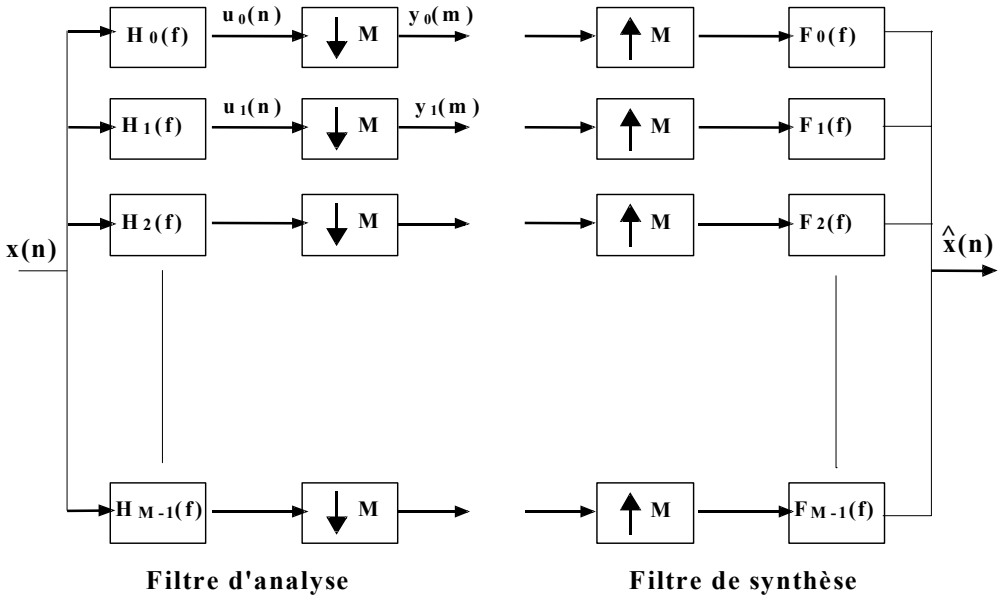


Figure 2-13 : Structure d'un banc de filtre d'analyse et de synthèse

Les 32 sous bandes sont équidistantes et de fréquence d'échantillonnage Fe/32. Ce banc réalise une partition régulière sur l'axe des fréquences par translation en fréquence d'un filtre FIR passe-bas prototype. La figure 2-13 représente la structure du banc de filtres. Le filtre $H_k(f)$ est identique au filtre $F_k(f)$. L'opération de décimation (↓ M) consiste en la suppression de M-1 échantillons sur M. L'opération (↑ M) consiste en l'insertion de M-1 zéros entre chaque échantillon. Les sorties des filtres $F_k(f)$ sont sommées pour former $\hat{x}(n)$. Soit *fe*, la fréquence d'échantillonnage, M le nombre de sous bandes et k le numéro de la sous-bande, nous obtenons :

- Largeur de bande : $\Delta fc = \frac{fe}{4M}$
- Fréquence centrale de chaque filtre : $fc = (2k+1)\frac{fe}{4M}$

Les M réponses en fréquence $H_k(f)$ se déduisent d'un filtre prototype par la relation :

$$H_k(f) = H\left(f - \frac{2k+1}{4M}\right) + H\left(f + \frac{2k+1}{4M}\right)$$

La translation en fréquence est obtenue par la modulation de la réponse impulsionnelle du filtre prototype h(n) par une fonction en cosinus. La réponse impulsionnelle du filtre en sous

bande est alors $h_k(n) = 2h(n)\cos\left(2.\Pi\frac{(2k+1)}{4M}n\right)$. La répartition en fréquence du filtre prototype est représentée par la figure 2-14.

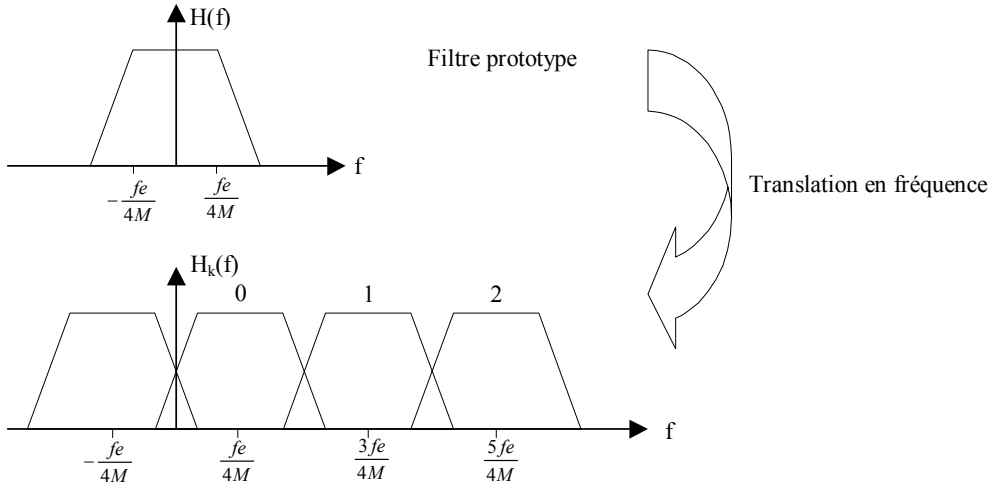


Figure 2-14 : Translation en fréquence du filtre passe-bas prototype

Pour satisfaire à la condition de reconstruction parfaite, c'est-à-dire que le signal après analyse et synthèse soit égal au signal d'entrée à un retard près, on doit utiliser un filtre de réponse en fréquence H(f) et de réponse impulsionnelle h(n) :

$$H(f) = \begin{cases} 1 & \text{pour } -\frac{fe}{4M} < f < \frac{fe}{4M} \\ 0 & \text{ailleurs} \end{cases} \quad \text{et} \quad h(0) = \frac{1}{2M}$$

$$h(n) = \frac{1}{2M}\sin c\left(\frac{n}{2M}\right)$$

Le nombre de coefficient N de la réponse impulsionnelle étant limité à 512, la condition de reconstruction n'est pas remplie. Cette condition n'est pas fondamentale car il suffit que le taux d'ondulation en bande passante soit inférieur à un seuil imposé par la sensibilité de l'oreille à des variations de pression acoustique (0.01 dB) et que la réponse en fréquence dans la bande atténuée soit inférieure au seuil imposé par la dynamique du signal non compressé (-96 dB pour le compact disque). La figure 2-15 montre la réponse globale du banc de filtre.

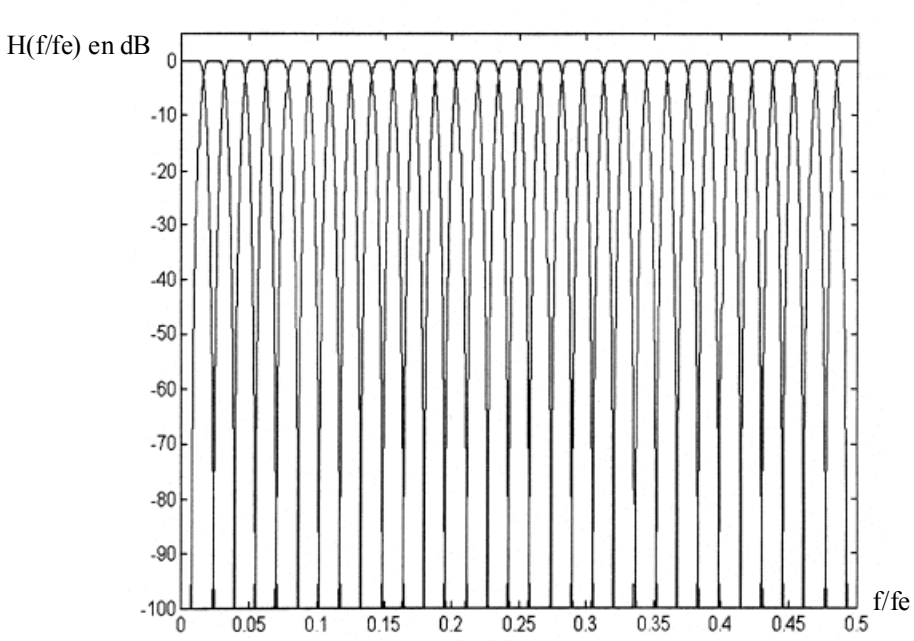
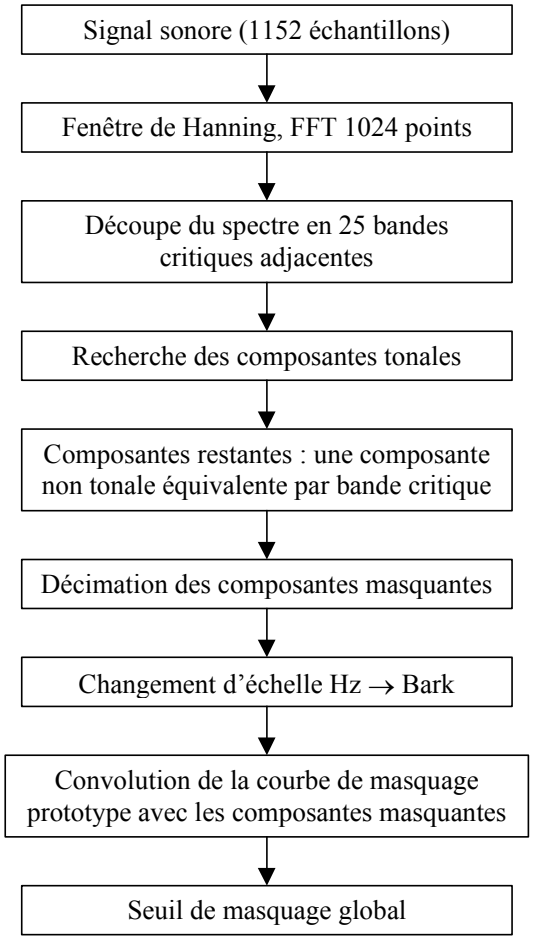


Figure 2-15 : Réponse globale du banc de filtre

Dans la couche II, les échantillons en sortie du banc de filtres d'analyse sont regroupés en 3 groupes de 12 échantillons par sous-bande.

2.4.3 Le modèle psycho-acoustique n°1

Le modèle n°1 est le moins complexe des 2 modèles de MPEG. Son principal avantage est d'avoir un nombre d'opérations limité, donc un temps de calcul faible. Pour cela, de nombreuses approximations sont faîtes, notamment pour la résolution fréquentielle du système, c'est à dire le nombre de composantes prise en compte. Le synoptique suivant résume les différentes étapes nécessaires au calcul du niveau global de masquage.



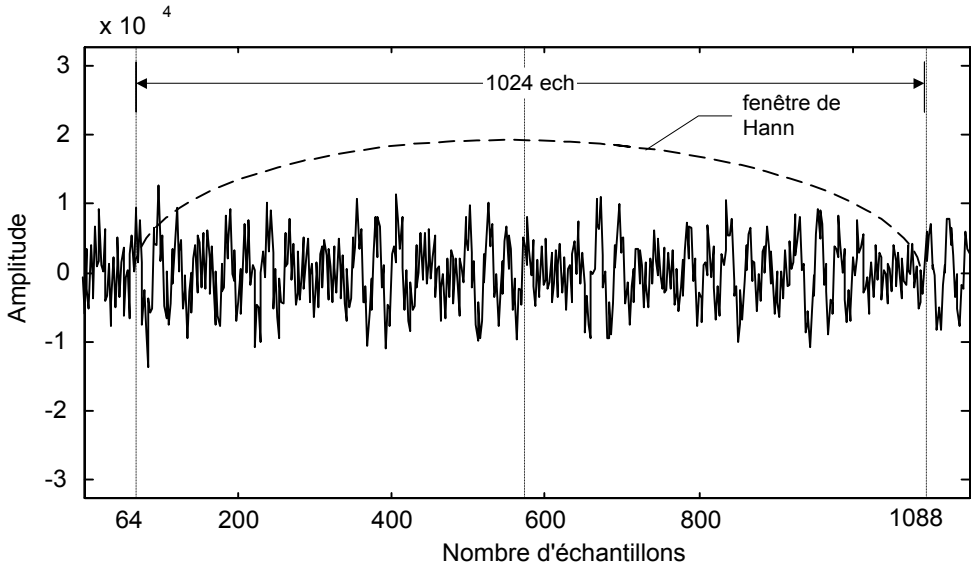
2.4.3.1 Représentation fréquentielle

Comme dans tous les modèles basés sur les propriétés fréquentielles de l'oreille, Il est nécessaire d'effectuer une analyse fréquentielle précise du signal sonore. La première opération consiste donc à estimer la densité spectrale de puissance X(k) du signal sonore. Le spectre est calculé par une FFT (Fast Fourier Transform) sur 1024 échantillons x(n) (512 pour la couche 1) préalablement pondérés par une fenêtre de Hann h(n) pour réduire les effets de blocs. Les résultats sont donnés en Décibel et sont normalisés de façon que le niveau maximum soit égal à 96 dB. Des valeurs relatives suffisent car la partie quantification de ce codeur utilisera les facteurs d'échelle, donc des valeurs normalisées.

Un décalage de 256 échantillons est nécessaire pour compenser le retard introduit par le banc de filtres. De plus la fenêtre de pondération doit être centrée sur le bloc de 1152 échantillons sonores entrant. Les échantillons se trouvant en dehors de la fenêtre d'analyse ne jouent pas un rôle primordial dans l'estimation du spectre. La figure suivante montre la position de la fenêtre d'analyse sur un bloc de 1152 échantillons. Le spectre de densité de puissance est donné de la façon suivante :

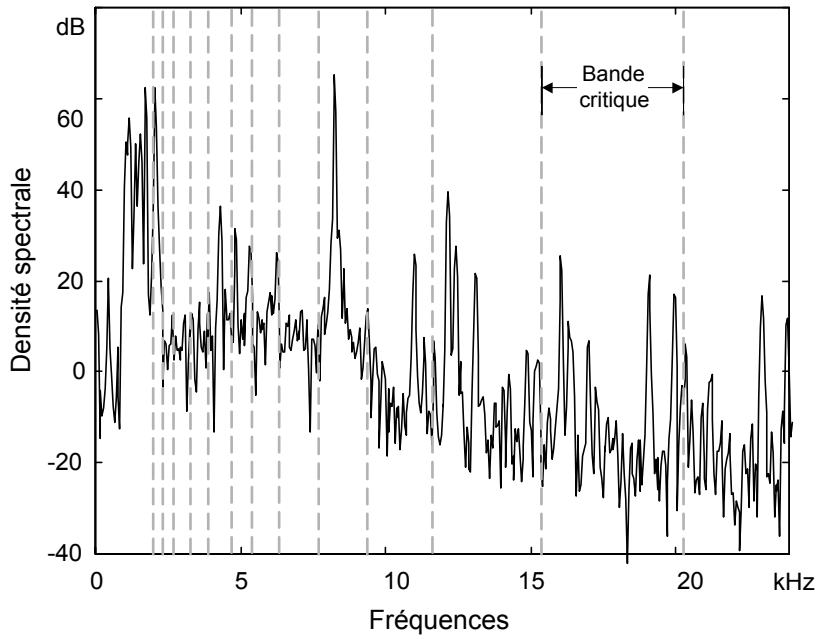
$$X(k) = 10\log\left(\left|\frac{1}{N}\sum_{l=0}^{N-1} h(l)x(l)e^{\frac{-2\pi jkl}{N}}\right|^2\right) \text{ dB} \qquad k = 0.....\left(\frac{N}{2}-1\right)$$

avec $h(n) = \sqrt{\frac{8}{3}}0.5\left(1-\cos\left(\frac{2\pi n}{N-1}\right)\right)$ $0 \le n \le N\text{-}1$. Le terme $\sqrt{\frac{8}{3}}$ permet de compenser la perte d'énergie due à l'atténuation du signal aux bord de la fenêtre de Hann.



Représentation temporelle d'un bloc de 1152 échantillons, position de la fenêtre d'analyse.

La figure suivante montre la densité spectrale de puissance de ce même bloc d'échantillons.



Densité spectrale de puissance calculée par le modèle n°1

La résolution fréquentielle de cette représentation dépend évidemment du nombre de points de la FFT. Dans le cas de la couche 2, la FFT est calculée sur une fenêtre de 1024 points. La résolution fréquentielle pour un signal échantillonné à F_e est égale : $\Delta f = \frac{F_e}{1024}$. (exemple : pour $F_e = 48\text{kHz}$ $\Delta f = \frac{48000}{1024} = 46.8\text{Hz}$).

2.4.3.2 Localisation des composantes tonales

Le masquage est différent selon que le son masquant est tonal (son pur) ou non tonal (bruit) ; il est donc nécessaire de réaliser cette distinction pour toutes les composantes X(k). Les composantes tonales sont repérées à l'aide de règles empiriques. Le modèle n°1 classe comme tonale une composante vérifiant la condition suivante :

Soit X(k), un maximum local, c'est à dire une composante vérifiant :

$$X(k-1) < X(k) \geq X(k+1)$$

et X(k) est tonal s'il vérifie :

$$X(k) - X(k+j) \geq 7 \text{ dB}$$

Cette condition doit être vérifiée dans des bandes de fréquences, correspondant à la résolution fréquentielle de l'oreille, meilleure aux basses fréquences. En fait la résolution en fréquence de l'oreille aux basses fréquences est très supérieure à celle de la FFT avec 1024 points ; mais la valeur empirique de 7 dB a été déterminée dans les mêmes conditions, c'est à dire sur une représentation spectrale similaire et sur ces mêmes bandes de fréquences. Cette condition doit être vérifiée pour tous les j tels que (cas de la couche 2) :

$$\begin{aligned} &j = -2,+2 && \text{pour } 2 < k < 63 \\ &j = -3,-2,+2,+3 && \text{pour } 64 < k < 127 \\ &j = -6,\ldots,-2,+2,\ldots,+6 && \text{pour } 128 < k < 255 \\ &j = -12,\ldots,-2,+2,\ldots,+12 && \text{pour } 256 < k < 500 \end{aligned}$$

Pour $F_e = 48$ kHz, ces bandes de fréquences centrées sur un maximum local ont pour valeur :

$$\begin{aligned} &\Delta f = 93{,}75 \text{ Hz} && \text{pour } 0 < f \leq 3 \text{ kHz} \\ &\Delta f = 140.63 \text{ Hz} && \text{pour } 3 < f \leq 6 \text{ kHz} \\ &\Delta f = 281.25 \text{ Hz} && \text{pour } 6 < f \leq 12 \text{ kHz} \\ &\Delta f = 562.50 \text{ Hz} && \text{pour } 12 < f \leq 24 \text{ kHz} \end{aligned}$$

Si la composantes X(k) est classée tonale, on lui ajoute la puissance des deux composantes voisines pour obtenir la composante masquante tonale $X_{tm}(k)$:

$$X_{tm}(k) = 10\log\left(10^{\frac{X(k-1)}{10}} + 10^{\frac{X(k)}{10}} + 10^{\frac{X(k+1)}{10}}\right) \text{ en dB}$$

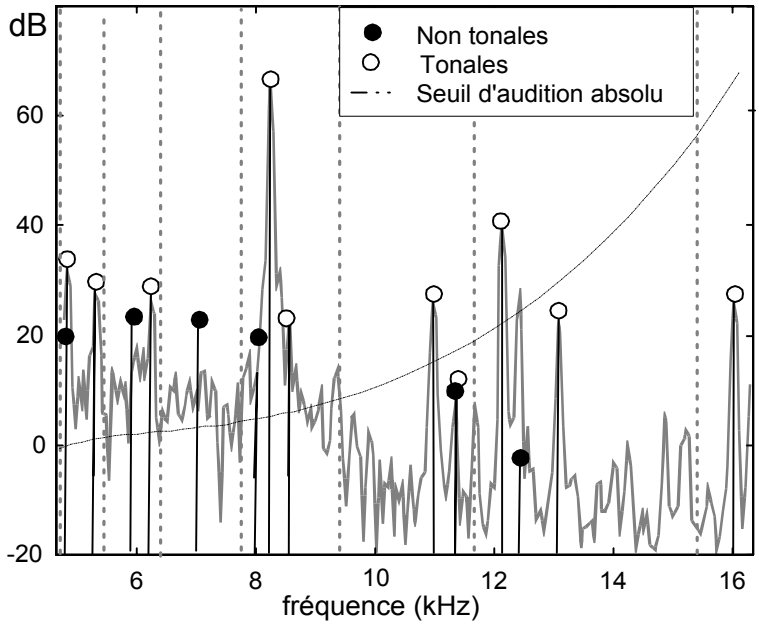
2.4.3.3 Détermination des composantes non tonales

Dans un premier temps, le spectre est découpé en 26 bandes critiques (jusqu'à 24 kHz) artificiellement juxtaposées et dont les limites sont fixées par la norme MPEG. La figure de la page précédente montre la découpe de l'axe des fréquences par ces bandes critiques à partir d'environ 2 kHz pour plus de clarté. Dans ces bandes critiques, toutes les composantes qui ne sont pas classées tonales, sont considérées comme non tonales. A la limite d'audition, l'oreille réalise une intégration des puissances dans une bande de fréquences donnée : la bande critique. La puissance perçue par l'oreille dans une bande critique est égale à la somme de

toutes les puissances des composantes dans cette bande de fréquence. Le modèle n°1 utilise cette propriété en assimilant les composantes non tonales, dans une même bande critique, à une composante unique dont la puissance est la somme des puissances des composantes non tonales de cette bande critique et sa position, la moyenne géométrique de la bande critique concernée. Soit k la position de cette composante, k_{min} et k_{max} sont respectivement la limite inférieure et supérieure de la bande critique, la composante masquante non tonales est :

$$X_{nm}(k) = 10\log\left(\sum_{i=k_{min}}^{k_{max}} 10^{\frac{X(i)}{10}}\right)$$ pour les X(i) non tonales avec $k = \left(k_{min} \cdot k_{max}\right)^{1/2}$

La figure suivante donne la représentation du spectre de puissance à partir de la fréquence 4.4 kHz ainsi que ses composantes tonales et non tonales.



composantes tonales et non tonales

2.4.3.4 Représentation en Bark

La représentation fréquentielle du signal dans l'échelle des Bark (1 Bark = 1 largeur de bande critique) est nécessaire pour l'application des propriétés psycho-acoustiques et surtout pour le calcul du seuil de masquage. Si on exprime la fréquence suivant cette nouvelle unité, la forme de la courbe de masquage générée par une composante, ne dépend pas de la fréquence de

cette composante. La relation entre une fréquence, exprimée en hertz, dans l'intervalle [20 Hz-20 kHz] et une fréquence, exprimée en Bark, dans l'intervalle [1-24 Bark] est donnée par l'expression suivante :

$$f_{Bark} = 13 \cdot \text{arctg}\left[0.76 \cdot f_{kHz}\right] + 3.5 \cdot \text{arctg}\left[\left(\frac{f_{kHz}}{7.5}\right)^2\right].$$

2.4.3.5 Réduction de la complexité de traitement

Décimation des composantes masquantes

Pour réduire la complexité de traitement, certaines composantes parmi les composantes listées (tonales ou non tonales) sont éliminées ; les critères d'élimination des composantes sont basés sur des propriétés psycho-acoustique ou des hypothèses simplificatrices. Ces critères sont les suivants :

- les composantes ayant une puissance inférieure au seuil d'audition absolu LT_q sont éliminées : $X_{tm}(k) \leq LT_q(k)$ et $X_{nm}(k) \leq LT_q(k)$
- Deux composantes tonales séparées de moins de 0.5 Bark (une demi-bande critique) entraînent l'élimination de la moins puissante.

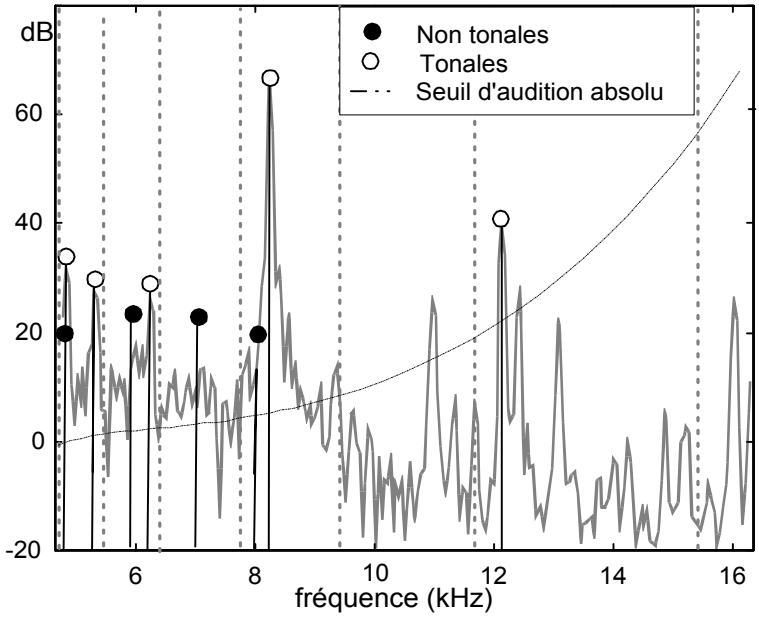
Cette décimation permet d'obtenir un nombre de composantes tonales N_t et un nombre de composantes non tonales N_n tel que : $N_t \leq 24$ et $N_n \leq 24$.

Décimation de l'axe fréquentiel

Il faut réduire davantage la complexité de traitement pour le calcul de la courbe de masquage sur toute la bande audible. On effectue un sous-échantillonnage de l'axe des fréquences de la FFT représenté par $k \in [0\ldots512]$. Un nouvel indice $i \in [0\ldots126]$, correspondant à une version sous-échantillonnée de k, est défini de la manière suivante :

- pour $k \in [0\ldots47]$ (sous-bandes n°0 à n°2 du banc de filtres) aucun sous-échantillonnage n'est appliqué.
- pour $k \in [48\ldots95]$ (sous-bandes n°3 à n°5 du banc de filtres) une raie sur deux est prise en compte.

- pour k ∈ [96…191] (sous-bandes n°6 à n°11 du banc de filtres) une raie sur quatre est prise en compte.
- pour k ∈ [192…511] (sous-bandes n°12 à n°31 du banc de filtres) une raie sur huit est prise en compte.



composantes tonales et non tonales après décimation

2.4.3.6 Calcul de la courbe de masquage individuelle

Lorsque les composantes masquantes sont repérées et listées, il reste à leur appliquer la courbe de masquage à la manière d'une convolution dans l'échelle des Bark. Cette courbe est une somme entre une courbe d'excitation $f(b_m,b)$, dont la forme est indépendante de la fréquence (en Bark) de la composante masquante $X_{tm}(b_m)$ ou $X_{tm}(b_m)$, et un taux de masquage $a_{Vtm}(b_m)$ ou $a_{Vnm}(b_m)$ négatif dépendant de la tonalité et de la fréquence de cette même composante. Les courbes de masquage appliquées aux composantes masquantes tonales et non tonales sont données par les expressions suivantes :

$$\begin{aligned} LT_{tm}(b_m,b) &= \mathbf{X_{tm}}(\mathbf{b_m}) + a_{Vtm}(b_m) + f(b_m,b) \quad \text{dB} \\ LT_{nm}(b_m,b) &= \mathbf{X_{nm}}(\mathbf{b_m}) + a_{Vnm}(b_m) + f(b_m,b) \quad \text{dB} \end{aligned}$$

$X_{tm}(b_m)$ ou $X_{tm}(b_m)$: composantes masquantes.

$a_{Vtm}(b_m)$ ou $a_{Vnm}(b_m)$: taux de masquage.

$f(b_m,b)$: fonction d'étalement de la cochlée (excitation).

b_m : taux de bande critique (Bark) de la composante masquante.

b : taux de bande critique de la composante masquée.

Calcul de l'excitation

La fonction modélisant l'excitation définie par le modèle n°1 de MPEG audio prend en compte l'influence de la puissance de la composante masquante, elle est indépendante de la tonalité de la composante masquante et de son taux de bande critique. Afin de limiter une nouvelle fois la complexité du traitement, l'étendue de la courbe d'excitation le long de la bande audible est limitée à l'intervalle [-3 -- 8 Bark]. Cette courbe d'excitation, modélisée par des segments de droite dans l'échelle des Bark, est donnée par :

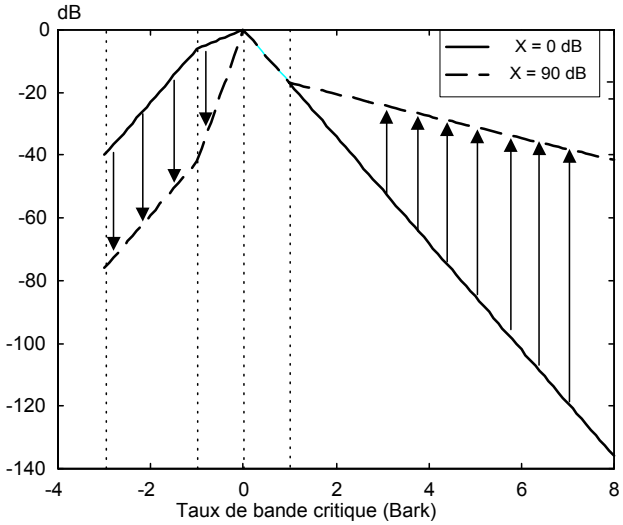
$$f(b_m,b) = 17\cdot(b-b_m+1) - \left[0.4\cdot\mathbf{X}(\mathbf{b_m})+6\right] \text{ dB} \qquad -3 \le b-b_m < -1 \text{ Bark}$$

$$f(b_m,b) = \left[0.4\cdot\mathbf{X}(\mathbf{b_m})+6\right]\cdot(b-b_m) \text{ dB} \qquad -1 \le b-b_m < 0 \text{ Bark}$$

$$f(b_m,b) = 17\cdot(b-b_m) \text{ dB} \qquad 0 \le b-b_m < 1 \text{ Bark}$$

$$f(b_m,b) = \left[17-0.15\cdot\mathbf{X}(\mathbf{b_m})\right]\cdot(b-b_m-1) - 17 \text{ dB} \qquad 1 \le b-b_m < 8 \text{ Bark}$$

La courbe est représentée sur la figure suivante pour deux niveaux de puissances de composantes masquantes.



Courbe d'excitation du modèle n°1 de MPEG Audio

Le taux de masquage

Le taux de masquage dépend de la tonalité de la composante masquante ; il est plus important en valeur absolue pour une tonale que pour une non-tonale. De plus, il dépend de la fréquence de la composante masquante. Il est donné par :

Pour les masquantes tonales : $a_{Vtm} = -1.525 - 0.275(b_m - b) - 4.5$ dB

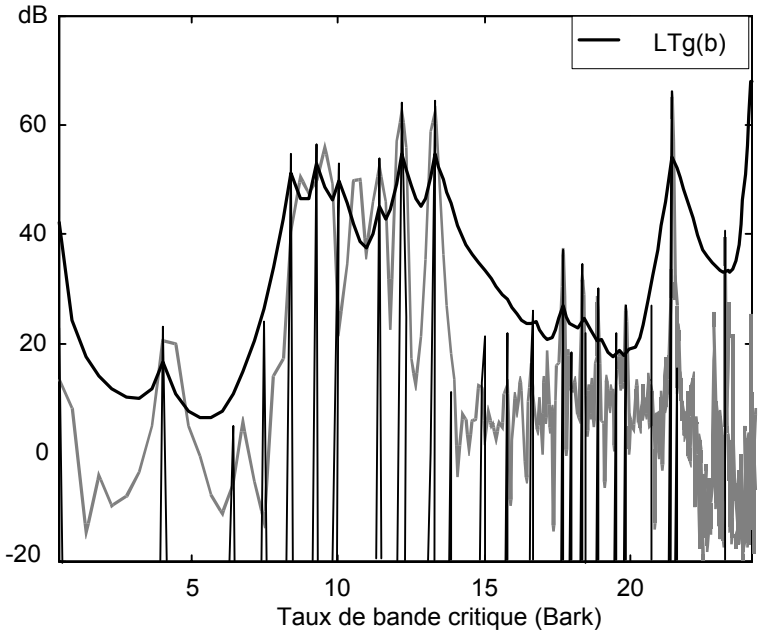
Pour les masquantes non tonales : $a_{Vnm} = -1.525 - 0.175(b_m - b) - 0.5$ dB

2.4.3.7 Le seuil masquage global

Le seuil masquage global est obtenu par la somme des puissances correspondant au seuil individuel et au seuil d'audition absolu.

$$\mathrm{LT_g(i)} = 10\log\left(10^{\frac{\mathrm{LT_q(i)}}{10}} + \sum_{j=1}^{N_t} 10^{\frac{\mathrm{LT_{tm}(j,i)}}{10}} + \sum_{j=1}^{N_n} 10^{\frac{\mathrm{LT_{nm}(j,i)}}{10}}\right)$$

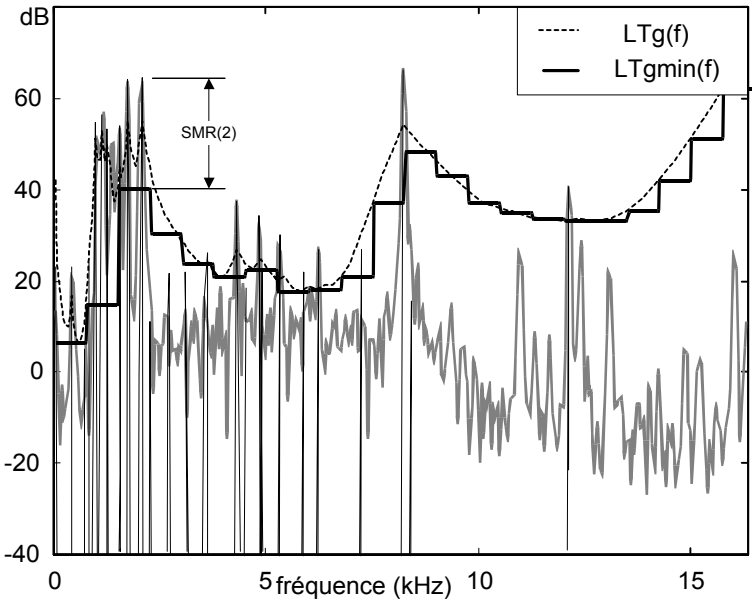
La figure suivante donne un exemple de seuil de masquage global (en Bark) à partir du même bloc d'échantillons que précédemment.



Seuil de masquage global en fonction du taux de bande critique.

2.4.3.8 Le rapport signal à masque (SMR)

Pour réaliser l'allocation optimale de bits, il suffit de connaître le rapport signal à masque dans chacune des sous-bandes. Le spectre de puissance est découpé en sous-bandes du banc de filtres (16 composantes par sous-bande pour la FFT 1024). Dans chaque sous bande n, le niveau minimum de seuil de masquage (Lt_{gmin}) est déterminé.



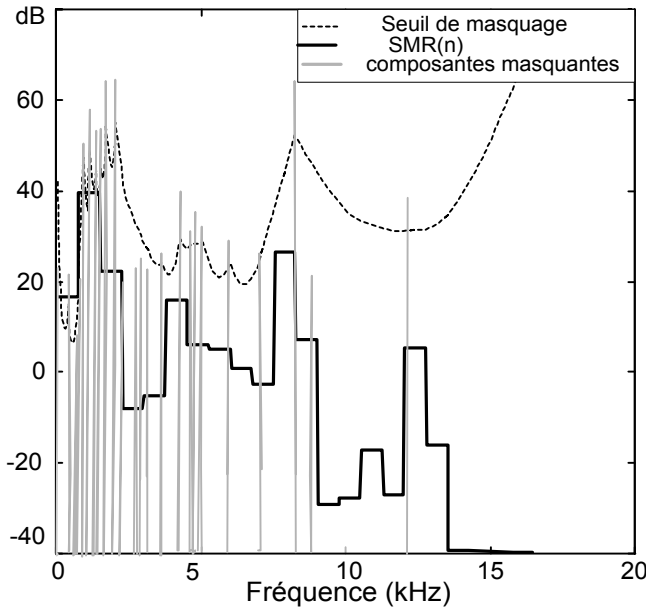
Seuil de masquage minimum pour chaque sous-bande de la même trame.

Le niveau de puissance maximale pour chaque sous-bande est déterminé, en comparant la composante fréquentielle la plus puissante et le facteur d'échelle maximum parmi les 3 de la sous-bande concernée.

$$L_{sb}(n) = \max\left(X(k), 20\log(FE_{max}(n) * 32768) - 10\right) \quad \text{dB}$$

Le terme -10 dB corrige la différence entre le niveau crête et le niveau RMS. Pour le calcul du SMR, on effectue la différence en dB entre le niveau de puissance maximal et le minimum de la courbe de masquage ($LT_{min}(n)$)sur la sous-bande considérée.

$$\boxed{SMR(n) = L_{sb}(n) - LT_{\min}(n) \ \text{dB}}$$



Rapport signal à masque du modèle psychoacoustique n°1

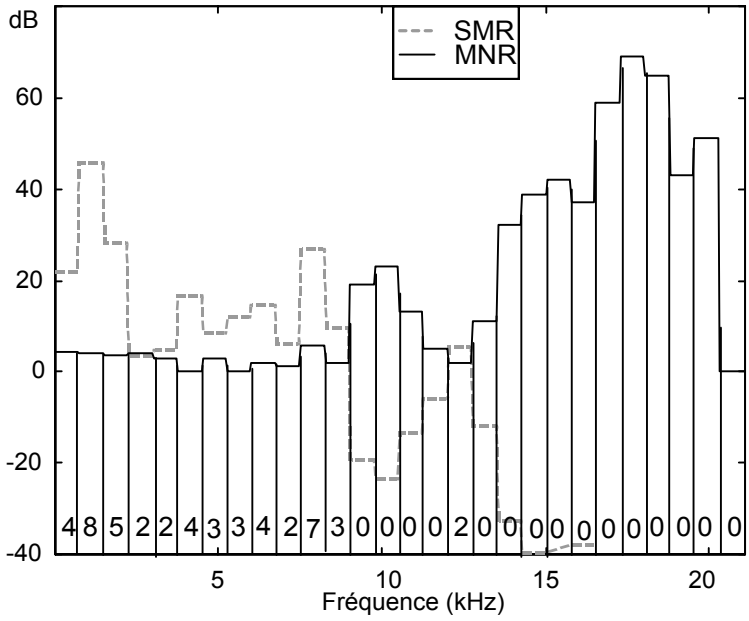
2.4.4 L'affectation binaire

L'affectation binaire consiste en une série d'opérations qui sont :

- Le calcul des facteurs d'échelle (FE1, FE2, FE3) de chaque groupe de 12 échantillons par sous-bande. Le maximum de la valeur absolue de ces douze échantillons est codé avec un mot de 6 bits.
- Le codage des facteurs d'échelle et des informations de sélection des facteurs d'échelle (scfsi) selon une table fournie par la norme. Seuls les scfsi des sous-bandes ayant une affectation binaire non nulle sont transmis. Comme on transmet 3 facteurs d'échelle consécutifs par sous-bande, on effectue un codage différentiel des facteurs d'échelle.
- Un rapport signal à masque (SMR) pour chaque sous-bande est utilisé pour déterminer le nombre de bits assigné à un bloc d'échantillons (3*12 en couche 2). La procédure d'affectation binaire consiste à déterminer le rapport masque à bruit minimum (MNR) pour chaque sous-bande à partir du SMR et du rapport signal sur bruit de quantification (SNR donné par la norme en fonction de l'affectation binaire). Le MNR est déterminé en utilisant une procédure itérative sur l'expression suivante :

$$\boxed{MNR(n) = SMR(n) - SNR} \text{ en dB}$$

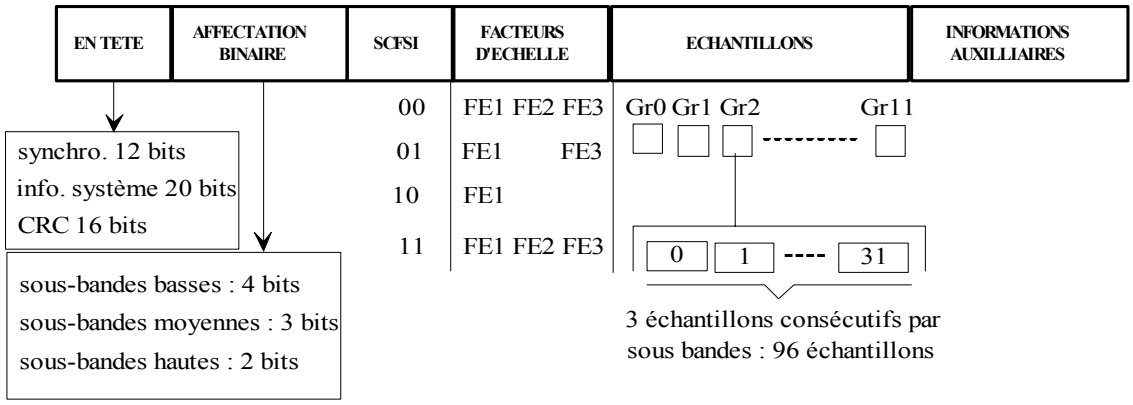
A la première itération, aucun bit n'est alloué dans chacune des 32 sous-bandes, l'algorithme calcule le MNR pour chaque sous-bande à partir du SNR correspondant. A chaque itération, dans la sous-bande où le MNR est le plus faible, le nombre de bits assignés à cette sous-bande est incrémenté dans la limite des bits disponibles. Ce processus est répété tant que tous les bits disponibles ne sont pas tous utilisés. Aucun bit n'est affecté au facteur d'échelle si aucun bit n'est affecté dans la sous-bande correspondante. Cette procédure d'allocation de bits peut allouer 0,2,....15 bits aux sous-bandes basses fréquences mais sur les sous-bandes hautes, le nombre de bits possible est limité en fonction du débit. La norme fournie un tableau montrant les allocations possibles par sous-bande. La figure suivante montre une représentation du SMR, du MNR et de l'affectation binaire correspondante.



- La quantification et le codage des échantillons en sous-bande selon un quantificateur linéaire. Chacun des échantillons en sous-bande est normalisé en divisant sa valeur par le facteur d'échelle pour obtenir X, et quantifié par la formule AX+B, où A et B sont donnés par la norme. Seuls les N bits de poids fort sont pris en compte où N représente le nombre de bits nécessaires pour coder le nombre de pas.
- Le codage de l'affectation binaire et de données auxiliaires.

2.4.5 Le formatage du train binaire

Le format du train binaire est représenté ci-dessous :



Il ne comprend que des codes à longueur fixe. Le débit en sortie du codeur est donc constant.

2.4.6 Performance du codeur MPEG1 audio couche II

L'appréciation des performances du codeur est réalisée par des tests subjectifs. On recherche la transparence entre le signal sonore original et le signal traité par la chaîne codeur/décodeur, le décodeur étant indépendant du modèle psycho-acoustique retenu pour la compression. Au résultat des tests obtenus avec le modèle n°1 pour la radiodiffusion numérique DAB (le codeur est aussi appelé MUSICAM), le CCETT a défini des niveaux de qualité selon le débit binaire par voie monophonique :

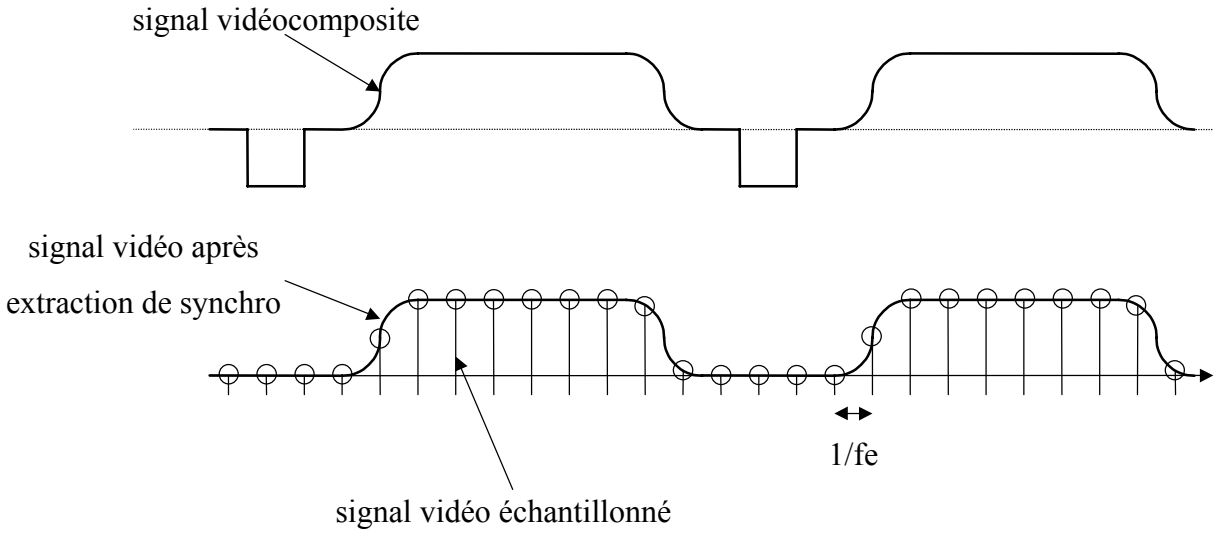
- 192 kbit/s $\Rightarrow$ qualité professionnelle pour studio de production.
- 128 kbit/s $\Rightarrow$ qualité diffusion type compact disque.
- 96 kbit/s $\Rightarrow$ qualité diffusion type compact disque.
- 91 kbit/s $\Rightarrow$ qualité diffusion type MULTISON TVHD.
- 64 kbit/s $\Rightarrow$ qualité commerciale.

3 La numérisation des images

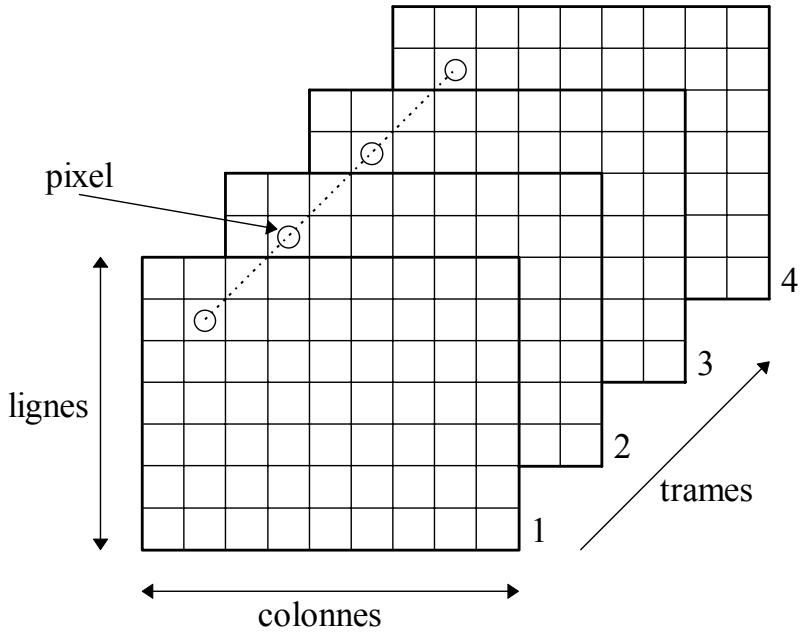
3.1 La norme CCIR601

Depuis de nombreuses années déjà, les professionnels de la vidéo utilisent divers formats numériques dans les studios des chaînes de télévisions pour l'enregistrement, la manipulation, le montage et la copie des signaux vidéos. Afin de faciliter l'interopérabilité des matériels et l'échange des programmes, le CCIR (Comité Consultatif International des Radiocommunications) a normalisé les conditions de numérisation (CCIR601) et d'interface (CCIR656) des signaux vidéos numériques en composantes (Y, Cb, Cr). Les principaux avantages de ces formats numériques normalisés sont de permettre des copies multiples sans dégradation de la qualité des images, de permettre des effets spéciaux irréalisables en analogique, de faciliter les montages de toutes sortes ainsi que de permettre les échanges internationaux indépendamment du standard utilisé pour la diffusion car les fréquences d'échantillonnage sont les mêmes dans tous les pays.

En respectant le théorème de Shannon (fe > 2.fmax), on doit prendre une fréquence d'échantillonnage supérieure à 10 MHz pour la luminance. On souhaite que les points d'échantillonnage soient situés au même emplacement sur toutes les lignes vidéo analogiques :



Cela conduit à réaliser une structure d'échantillonnage orthogonale, c'est à dire une structure fixe des échantillons d'une ligne à l'autre et d'une image à l'autre :



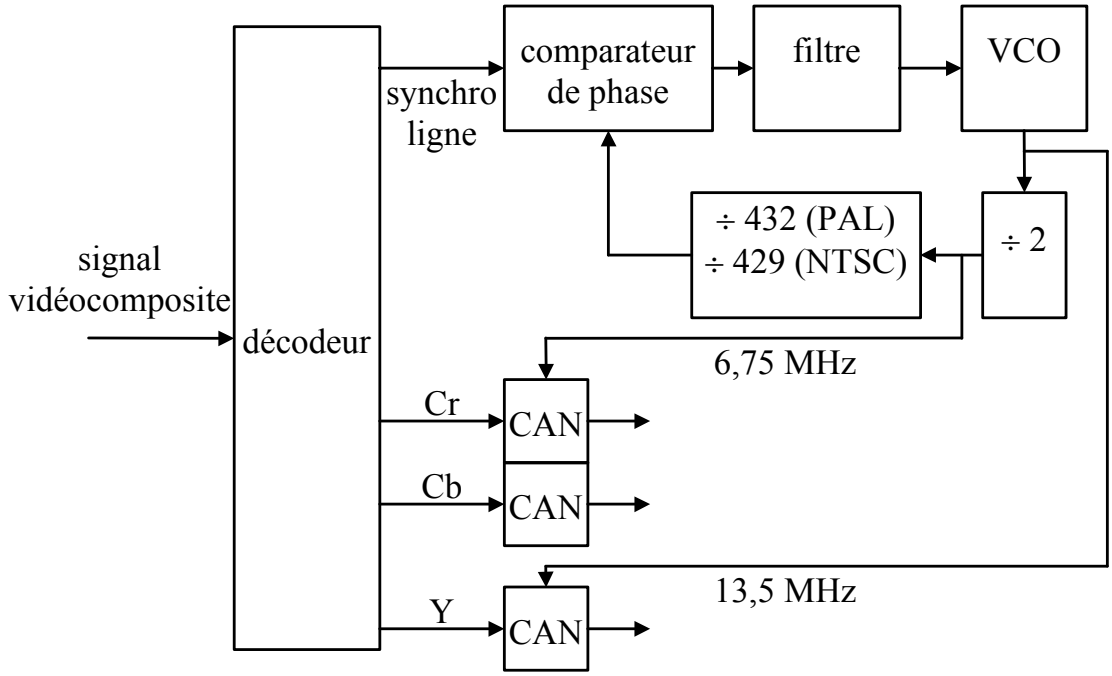
Les échantillons qui forment les pixels de l'image se situent sur une grille rectangulaire. Pour réaliser une telle structure, il faut que la fréquence d'échantillonnage soit un multiple entier de la fréquence ligne du signal vidéo à coder. Il n'en existe que deux valeurs dans le monde, 15625 Hz en PAL et SECAM et 15734 Hz en NTSC. Comme on souhaite une fréquence unique pour les deux systèmes, on choisit le plus petit commun multiple au-delà de 10 MHz : Fe = 13.5 MHz ce qui nous donne 858 échantillons par ligne pour les systèmes à 60 trames/s et 864 échantillons par ligne pour les systèmes à 50 trames/s. On prend 720 pixels pour la durée utile (visible) de la ligne. La fréquence d'échantillonnage 13.5/4 = 3.375 MHz est notée 1, donc la fréquence 13.5 MHz est notée 4.

On souhaite que les échantillons des signaux de chrominance Cr et Cb coïncident avec ceux de la luminance donc on échantillonne Cr et Cb avec une fréquence multiple de 3.375 MHz en phase avec l'horloge échantillonnant la luminance. On parle alors d'une structure d'échantillonnage orthogonale à coïncidence. Toutes les fréquences d'échantillonnage sont dérivées d'une horloge à 27 MHz, mais elles sont verrouillées en phase sur la fréquence ligne.

On appelle format la suite de trois chiffres A:B:C, chaque chiffre correspondant à la fréquence d'échantillonnage des signaux Y:Cb:Cr. Par exemple, le format 4:2:2 veut dire que :

Fe_Y = 13.5 MHz, Fe_{Cb} = 6.75 MHz et Fe_{Cr} = 6.75 MHz.

Le nombre de bits utilisé pour quantifier un échantillon (luminance ou chrominance) est égal à 8 pour la diffusion (S/N = 50 dB) ou 10 bits en studio (pour les trucages notamment). Le débit binaire brut (interface parallèle) est donc égal à 27x8 bits = 216 Mbit/s ou 27x10 bits = 270 Mbit/s.

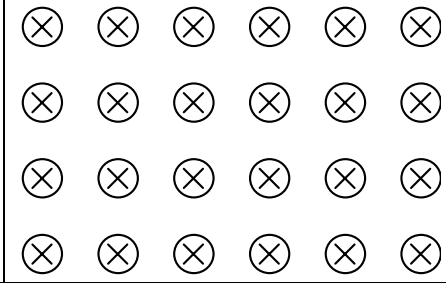
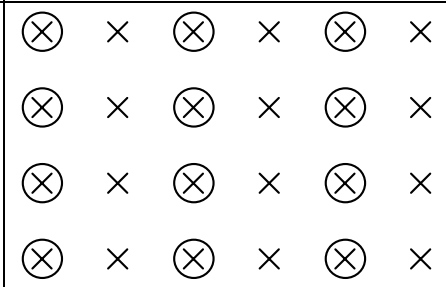
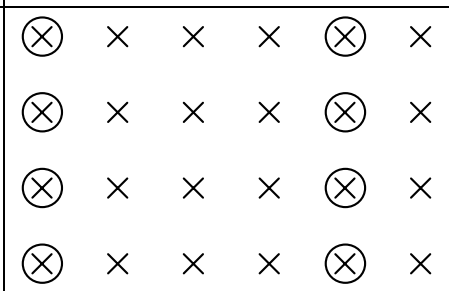
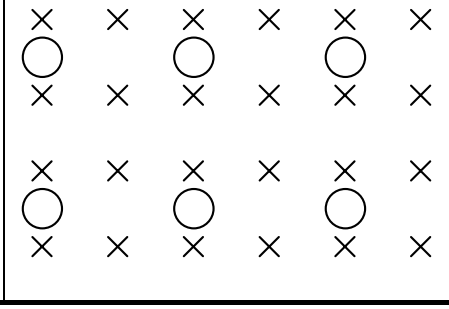


3.2 Les différents formats d'images

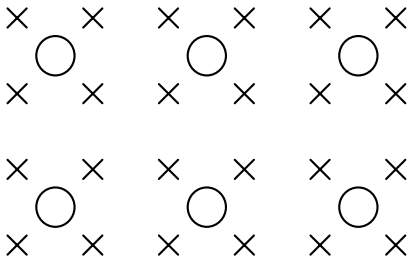
3.2.1 Les formats vidéos normalisés

Le problème majeur des formats d'images est principalement dû à l'absence de normalisation dans le domaine de l'informatique et à la divergence des besoins entre les mondes de la télévision, des télécommunications et du multimédia. Dans le domaine de la télévision et de la vidéoconférence, les choses sont à peu près claires et on peut classer les formats d'images ou de séquences d'images dans les catégories suivantes :

1. La norme CCIR601 a été définie pour la télévision numérique studio. Elle est la source naturelle du processus de codage MPEG-2. Elle prend en compte les standards européens à 625 lignes et américains à 525 lignes au format 4:3. Elle comprend aujourd'hui 4 structures d'échantillonnage appelées 4:4:4, 4:2:2, 4:1:1 et 4:2:0. Le tableau suivant résume les caractéristiques de ces différents formats.

nom	structure d'échantillonnage : × représente un échantillon de luminance ○ représente un échantillon de chrominance	$f_{échantillonnage}$ et dimensions	application
4 :4 :4		Y : 13,5 MHz 720x576 en 625 lignes 720x480 en 525 lignes Dr, Db : 13,5 MHz 720x576 en 625 lignes 720x480 en 525 lignes	qualité haute définition (pour la chrominance)
4 :2 :2		Y : 13,5 MHz 720x576 en 625 lignes 720x480 en 525 lignes Dr, Db : 6,75 MHz 360x576 en 625 lignes 360x480 en 525 lignes	post-production, studio
4 :1 :1		Y : 13,5 MHz 720x576 en 625 lignes 720x480 en 525 lignes Dr, Db : 3,375 MHz 180x576 en 625 lignes 180x480 en 525 lignes	diffusion
4 :2 :0		Y : 13,5 MHz 720x576 en 625 lignes 720x480 en 525 lignes Dr, Db : 6,75 MHz (une ligne sur deux) 360x288 en 625 lignes 360x240 en 525 lignes	diffusion (nécessite une mémoire d'image et un filtrage vertical)

2. La norme H261 de vidéoconférence à px64 Kbit/s a défini le format CIF (Common Intermediate Format) qui est aussi souvent utilisé pour l'affichage des séquence vidéo sur ordinateur. A noter l'existence d'un format QCIF (Quarter CIF) qui représente le quart d'une image CIF. Le format d'image est égal à 4 :3 et la fréquence image est de 30 Hz en balayage non entrelacé.

nom	structure d'échantillonnage : × représente un échantillon de luminance ○ représente un échantillon de chrominance	dimensions	application
CIF		Y : 352x288 Dr, Db : 176x144	vidéoconférence, informatique

3. La norme MPEG-1 a défini le format SIF (Source Input Format) comme source naturelle d'images pour le codeur. La structure d'échantillonnage est la même que pour le CIF mais elle prend en compte les standards européen à 625 lignes et américains à 525 lignes au format 4:3 en balayage non entrelacé (mode progressif).

nom	dimensions SIF625	dimensions SIF525	application
SIF	Y : 352x288 Dr, Db : 176x144	Y : 352x240 Dr, Db : 176x120	vidéo qualité VHS à 1,5 Mbit/s

3.2.2 Les formats informatiques

En ce qui concerne le format de fichiers graphiques, le monde informatique est proche de l'anarchie totale. On compte une bonne centaine de formats différents, aucun n'étant normalisé. Même le fichier contenant des données JPEG utilisé aujourd'hui est une norme de fait mais n'est pas le format officiel de l'ISO. Chaque constructeur d'équipement et de logiciel a créé ou crée encore son propre format, les anciens formats ne disparaissant jamais. On peut essayer de classer ces différents formats dans le tableau suivant :

catégorie	types de fichiers	remarque
langage de commande de dispositifs d'impression	Printer Command Language (PCL) PostScript (PS) Hewlett-Packard Printer Graphics Language (HPGL)	commande d'imprimante et de table traçante
langage de description de page	Encapsulated PostScript (EPS) Portable Document Format (PDF)	le PDF est de plus en plus utilisé sur Internet
format de télécopie	transmission : CCITT groupe 3 et JBIG stockage : formats propriétaires et TIFF classe F	
image bitmap (contient tous les pixels enregistrés dans l'ordre de balayage)	Graphics Interchange Format (GIF) Tag Image File Format (TIFF) Truevision Graphics Adapter (TGA) JPEG File Interchange Format (JPG) Microsoft Windows Bitmap (BMP) ...	Les plus utilisés. 90% des images échangées sur Internet sont au format GIF
image vecteur (contient la description mathématique des éléments de l'image)	Persistence of Vision (POV) Autocad Drawing Interchange Format (DXF) Virtual Reality Modeling Language (VRML) ...	beaucoup moins utilisés sauf VRML pour la 3D
image metafile (contient du format bitmap ou vectoriel)	Hierarchical Data Format (HDF) Macintosh Picture (PICT) Ritch Text Format (RTF) Microsoft Windows Metafile (WMF) ...	utilisés notamment sur Macintosh et sous Windows
multimédia (contient un mélange de texte, de son, d'images fixes et de séquences vidéo)	Intel : Digital Video Interface (DVI) Microsoft : Ressource Interchange File Format (AVI, RIFF, WAV) Apple : Quicktime (QTM) MPEG ...	MPEG est le seul normalisé, mais n'est pas le plus utilisé.

Pour être complet, il faudrait parler des espaces colorimétriques (RGB, YUV, CMY et HSV), du nombre de couleurs (8, 16, 24 ou 32), de la transparence, des possibilités d'animation (pour les images bitmap) ainsi que de l'apparition progressive à l'écran (au lieu de

l'apparition séquentielle). Les formats MPEG et JPEG devraient mettre un peu d'ordre dans ce domaine.

3.3 Critères d'évaluation de la qualité d'une image

3.3.1 Introduction

Pour mettre au point un critère objectif d'évaluation de la qualité d'une image, il est nécessaire d'établir un modèle théorique de l'observateur humain moyen. On connaît aujourd'hui assez bien le système visuel humain, c'est-à-dire tout ce qui concerne l'oeil et le nerf optique. Toutefois, nos connaissances s'arrêtent au cerveau : les divers traitements qui s'y produisent nous sont pour une large part inconnus. La modélisation théorique se révèle trop complexe pour se prêter à une représentation unique car elle met en jeu de nombreux facteurs psychologiques. Le goût personnel, l'humeur du moment, le type de programme le plus familier, la nature même du programme, tout concourt à rendre impossible la définition du mode de perception du « téléspectateur-type ». De plus, la situation est compliquée par les techniques numériques utilisées dans les compressions de type JPEG et MPEG. Elles sont pour la plupart non-linéaires et adaptatives et produisent des défauts dont les caractéristiques sont difficiles à évaluer car ils ne sont pas uniformément répartis sur toute l'image et ne peuvent être assimilés à un bruit blanc. On utilise donc deux critères pour évalué la qualité d'une image (ou d'une séquence d'images) : les critères objectifs basés sur des calculs mathématiques et les tests subjectifs basés sur l'évaluation humaine.

3.3.2 Les critères objectifs

On peut définir plusieurs critères quantitatifs en mesurant l'erreur entre l'image reconstruite notée $\hat{f}(x,y)$ et l'image originale notée $f(x,y)$, toutes deux de dimension MxN et codées sur n bits.

- L'erreur absolue moyenne : $EAM = \frac{1}{M.N}\sum_{x=1}^{M}\sum_{y=1}^{N}\left|f(x,y) - \hat{f}(x,y)\right|$.

- L'erreur quadratique moyenne : $EQM = \sqrt{\frac{1}{M.N}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(f(x,y) - \hat{f}(x,y)\right)^2}$.

- Le rapport signal sur erreur : $SNR = 10.\log\left(\frac{\frac{1}{M.N}\sum_{x=1}^{M}\sum_{y=1}^{N} f(x,y)^2}{EQM^2}\right)$.

- Le rapport signal sur erreur crête : $PSNR = 10.\log\left(\frac{\frac{1}{M.N}\sum_{x=1}^{M}\sum_{y=1}^{N} \left(2^n - 1\right)^2}{EQM^2}\right)$.

L'EAM et l'EQM sont utilisée dans les algorithmes de détection de mouvements pour déterminer le meilleur vecteur de mouvement. Le PSNR est utilisé, sous certaines conditions, pour tester la qualité des images comprimées à la place des tests subjectifs qui sont longs et coûteux à organiser. On utilise le PSNR au lieu du SNR car celui ci varie, à EQM identique, en fonction de la luminosité moyenne de l'image testée. Cette mesure permet une évaluation rapide de la différence entre deux séquences d'images. Elle n'est valable que si l'on souhaite comparer deux mécanismes de compression d'image de même nature, c'est-à-dire utilisant à peu près les mêmes techniques fondamentales. Ce critère a notamment été utilisé pour l'élaboration de la norme MPEG afin de sélectionner les techniques de compression. La méthode suivante a été suivie :

1. On établit un modèle de test utilisant les techniques faisant l'objet d'un certain consensus.

2. On incorpore dans ce modèle de test les techniques en compétition pour être normalisées.

3. Si l'amélioration en PSNR apportée par une méthode nouvelle est supérieure à 1 dB, on l'incorpore dans le modèle de test suivant. Sinon, elle est éliminée.

4. On établit un nouveau modèle et on recommence le processus.

Le PSNR peut être utilisé dans ces conditions plutôt restrictives. En général, il vaut mieux utiliser les tests subjectifs. Dans le cas général, il n'y a aucune garantie qu'une image comprimée ayant un meilleur PSNR qu'une autre soit subjectivement de meilleure qualité car cette mesure n'incorpore pas de modèle de l'observateur (ex : comparaisons entre deux images comprimées par JPEG et par une méthode basée sur les fractales).

3.3.3 Les tests subjectifs

L'autre démarche consiste à mettre en oeuvre des essais subjectifs présentés à un nombre significatif d'observateurs et effectués selon une méthode aussi précise que possible. On procède ensuite à une étude statistique pour calculer les moyennes, les variances, etc. La mise en oeuvre d'un test est délicate car les causes d'ambiguïté et d'incertitude sont nombreuses. Les caractéristiques clés d'une séance d'essais sont les suivantes :

- La taille et le choix de la population testée.

- Le choix des séquences de test :
  - ⇒ type de séquence.
  - ⇒ durée d'une séquence.

- Les conditions d'observation :
  - ⇒ distance d'observation.
  - ⇒ réglage des sources.
  - ⇒ disposition de la salle de test.
  - ⇒ matériel utilisé.

- Le choix de la méthode et de l'échelle de notation :
  - ⇒ échelle de qualité.
  - ⇒ échelle de dégradation.
  - ⇒ échelle de rapport.
  - ⇒ méthode à simple stimulus.
  - ⇒ méthode à double stimulus.

- Le traitement des résultats :
  - ⇒ observations non valables.
  - ⇒ calcul des notes.
  - ⇒ traitement statistique.

Devant la complexité du problème, le CCIR (Comité Consultatif International des Radiocommunications) a émis l'avis 500 qui est destiné à assurer une certaine homogénéité

dans les conditions expérimentales. On souhaite en effet que le même test organisé dans différents laboratoires à travers le monde donne la même note absolue de qualité (ou de dégradation). Cet avis n'est cependant pas assez précis pour organiser complètement une séance de test, aussi l'UER (Union Européenne de Radiodiffusion) a lancé plusieurs recherches pour le compléter. En l'absence de modèles théoriques utilisables, les procédures utilisées évoluent de manière plutôt empirique.

Il faut encore introduire deux notions subjectives utilisées dans la compression d'image :

- le défaut juste perceptible (Just Noticeable Defect (JND)). La compression est optimale quand on atteint le JND pour une distance d'observation donnée.

- la dégradation élégante (graceful degradation). Ce qui est important, ce n'est pas la dégradation quantitative apportée à l'image comprimée (mesurée par le PSNR), mais la manière dont l'observateur perçoit ce défaut. La compression par ondelettes vis à vis de JPEG est sans doute la meilleure illustration possible de cette notion.

4 La compression d'image

4.1 introduction

La numérisation des signaux analogiques est une tendance fondamentale en électronique parce que le signal numérique est plus robuste et plus facile à traiter que son équivalent analogique. Par contre, il occupe beaucoup plus de place. En télévision, le signal vidéocomposite analogique (sans compression) supportant une image au format 4:2:2 (voir : recommandation CCIR 601) occupe une bande passante d'environ 14 MHz. Le signal vidéo numérique correspondant (débit utile 166 Mbit/sec, codage NRZ) doit occuper une bande d'environ 80 MHz pour pouvoir être décodé. Si on souhaite diffuser ce signal dans un canal 8 MHz, compte tenu du son et des codes correcteurs d'erreur, il faudra comprimer le débit numérique vingt fois sans trop dégrader la qualité de l'image.

On peut légitimement s'interroger sur la pertinence des choix de fréquences d'échantillonnage (13.5 MHz et 6.75 MHz) et de nombre de bits par pixels (8 bits) s'il s'avère que seul 5 % de l'information de départ est nécessaire pour obtenir une image de bonne qualité. Il se trouve que les critères choisis pour quantifier une image sont commodes d'un point de vue technique, mais ne sont pas adaptés d'un point de vue physiologique. Il faut donc changer de perspective pour envisager une forte réduction d'information. La combinaison de plusieurs méthodes permet d'obtenir ce résultat :

1. Décorrélation intra-trame : on supprime la part d'information commune entre les pixels voisins à l'intérieur d'une image à l'aide d'une transformation de type TCD (Transformation en Cosinus Discrète).
2. Quantification psychovisuelle : on n'attribue aux fréquences spatiales de l'image que le nombre de bits suffisant pour satisfaire la réponse de l'œil.
3. Codage statistique.
4. Décorrélation inter-trames : on supprime la part d'information commune entre plusieurs images successives grâce à la détection de mouvement.

Les trois premières méthodes sont utilisées dans la norme ISO 10918 plus connue sous l'acronyme JPEG (Joint Photographic Experts Group). En leur ajoutant la décorrélation inter-trames, on obtient le principe des normes ISO 13818 plus connues sous l'acronyme MPEG (Motion Pictures Experts Group). La compression à base d'ondelettes est de même nature que

la compression selon JPEG, mais en remplaçant la TCD par une transformation à base d'ondelettes alors que la compression à base de fractales repose sur un principe totalement différent de ceux utilisés dans JPEG.

Le problème de la compression d'image se pose aussi en informatique car la taille des fichiers d'images non-comprimées est beaucoup trop importante pour le stockage et pour la transmission sur les réseaux. Un autre domaine où la compression d'image est importante concerne les images binaires dans le domaine de la télécopie par exemple. La norme CCITT groupe 3 y est progressivement remplacée par la norme JBIG (Joint Bi-level Image experts Group) crée conjointement par le CCITT et l'ISO.

4.2 Compression d'image fixe

4.2.1 Généralités sur la norme JPEG

4.2.1.1 Introduction

La norme JPEG décrit deux méthodes de compression d'images fixes :

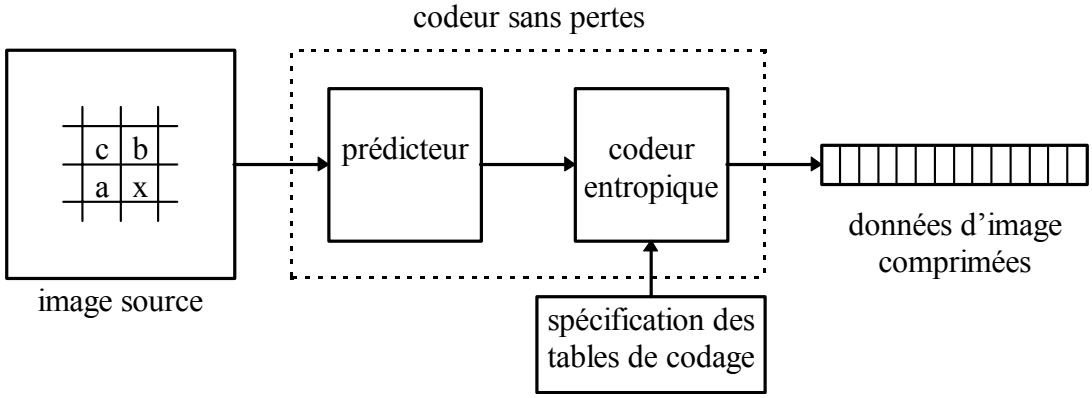
1. la compression sans pertes (lossless) est basée sur la prédiction linéaire (un peu comme dans le cas de la MIC différentielle). L'image décomprimée est identique à l'original.
2. la compression avec pertes (lossy) est basée sur la TCD. L'image décomprimée n'est pas identique à l'original, mais les défauts apportés par la compression sont supposés être invisible à l'œil (pour une distance d'observation donnée).

Ces méthodes sont applicables dans le cas d'images fixes de nature photographique à ton continu (continuous tone still pictures) codées en niveaux de gris ou en couleur. Les images de type binaire ne sont pas concernées et elles ont leur propre norme : JBIG. Il est à noter que la norme JPEG n'a pas été prévue ni mise au point sur des images de nature artificielle (comme une page de texte ou une image de synthèse par exemple, ce qui ne manque pas de poser quelques problèmes pour la télévision). Un deuxième point est assez surprenant dans JPEG comme dans MPEG d'ailleurs ; le codeur n'est pas normalisé. Seuls le format des données comprimées et le décodeur font l'objet de la normalisation. Des exemples de tables de codage sont donnés à titre informatif, mais il n'y a aucune garantie de qualité de compression dans les normes JPEG et MPEG. Il existe d'ailleurs d'importantes différences de qualité d'images entre les codeurs MPEG commercialisés. Par contre, tous les décodeurs

commercialisés doivent être capables de comprendre un train binaire JPEG ou MPEG et de l'afficher de la même manière.

4.2.1.2 Codage sans pertes

La figure suivante montre les principales étapes utilisées dans le processus de codage sans pertes.



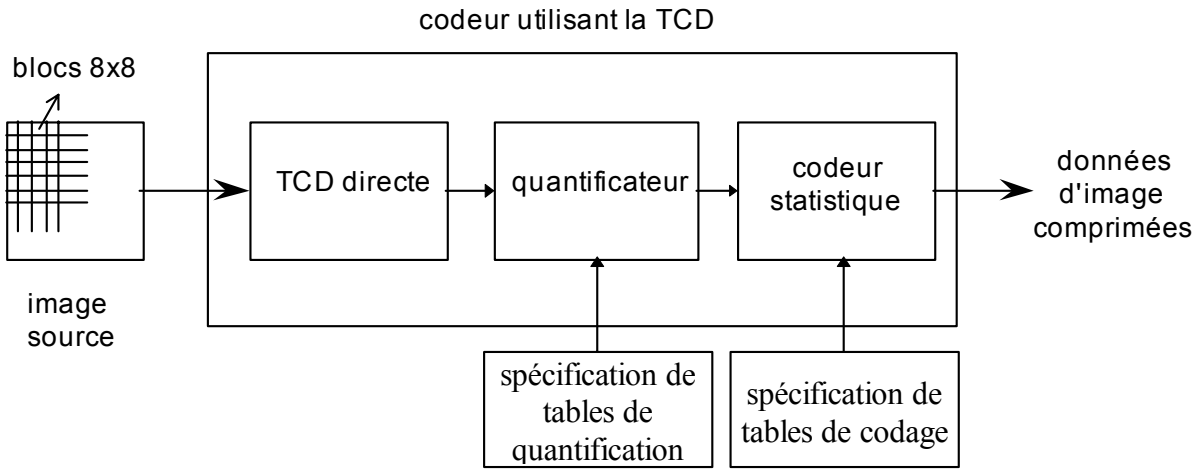
Un prédicteur utilise les valeurs des trois pixels voisins a, b et c pour former une prédiction $\hat{x}$ de l'échantillon à coder x. Cette prédiction est ensuite soustraite de la valeur effective du pixel x et la différence fait l'objet d'un codage entropique sans pertes de type Huffman ou arithmétique. Le codeur sélectionne le meilleur estimateur de x parmi les valeurs suivantes :

- $\hat{x} = a$,
- $\hat{x} = b$,
- $\hat{x} = c$,
- $\hat{x} = a + b - c$,
- $\hat{x} = a + ((b - c)/2)$,
- $\hat{x} = b + ((a - c)/2)$,
- $\hat{x} = (a + b)/2$.

L'image source peut être codée avec une précision allant de 2 à 16 bits par échantillons. Le taux de compression est de l'ordre de 50 % pour des images de complexité moyenne et n'apporte pas d'amélioration nette vis à vis des algorithmes utilisés dans un compresseur de type PKZIP. Cette méthode de compression peut trouver sa place dans les domaines où la perte d'informations est interdite, par exemple dans le cas des images médicales.

4.2.1.3 Codage avec pertes

La figure suivante montre les principales étapes utilisées dans le processus de codage avec pertes. Chaque composante de l'image source subit le même traitement, mais généralement avec des tables de codage différentes pour la luminance et pour la chrominance.



La composante à compresser est tout d'abord découpée en blocs 8x8 pixels. Chaque bloc subit ensuite le traitement suivant :

- Transformation en cosinus discrète (décorrélation intra-trame et passage dans l'espace des fréquences spatiales). On obtient un bloc transformé 8x8.

- Quantification psychovisuelle (adaptation du codage à la réponse de l'œil). Chaque coefficient du bloc transformé est divisé par une valeur contenue dans la table de quantification qui peut être spécifiée dans les données comprimées.

- Codage statistique (adaptation du codage au contenu de l'image). Les coefficients quantifiés sont remis en ordre (transformation en zigzag) puis codés à l'aide d'un codage par plage suivi d'un codage de Huffman ou d'un codage arithmétique. La table de codage peut être spécifiée dans les données comprimées.

La décompression s'effectue en sens inverse, les différents traitements étant réversibles.

Dans le codage à base de DCT, deux modes de fonctionnement sont possibles ; le mode séquentiel et le mode progressif. Dans le mode séquentiel, les blocs 8x8 sont entrés bloc par

bloc de gauche à droite et rangée de blocs par rangée de blocs de bas en haut. Quand un bloc a été quantifié et préparé au codage entropique, il est immédiatement codé et ajouté aux données d'image comprimées. Au décodage, l'image apparaît de la manière suivante :

Séquentiel

En mode progressif, les blocs 8x8 sont également codés dans le même ordre, mais en plusieurs balayage de l'image grâce à une mémoire tampon de coefficients (ayant la taille de l'image) insérée entre le quantificateur et le codeur entropique. Les coefficients quantifiés se trouvant dans ce tampon peuvent alors être partiellement codés à chaque balayage au moyen de deux procédures pouvant être combinées ensembles :

1. la sélection spectrale. On transmet à chaque balayage une bande de coefficients représentant des fréquences basses, moyennes ou hautes.

2. les approximations successives. On transmet à chaque balayage une partie des bits codant les coefficients, en partant du MSB et en affinant vers le LSB.

Au décodage, l'image apparaît progressivement (la résolution s'améliore à chaque balayage) de la manière suivante :

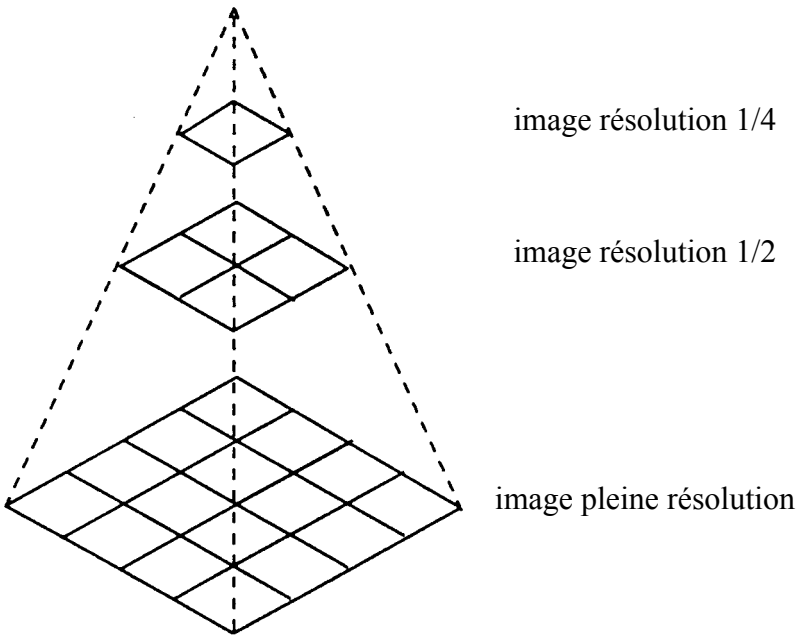
Progressif

L'image source peut être codée avec une précision de 8 bits ou de 12 bits par échantillons. Le taux de compression est de l'ordre de 10 à 30 pour des résultats de bonne qualité en fonction

de la complexité des images. Cette méthode de compression (en mode séquentiel) est à la base de la norme MPEG utilisée en télévision.

4.2.1.4 Mode hiérarchique

C'est un codage de type multi-résolution ou pyramidal. Les images de résolution ½ et ¼ sont calculées par filtrage/décimation afin de former, par exemple, la pyramide hiérarchique à trois niveaux suivante :



Schématiquement, l'algorithme de codage est le suivant :

1. On construit, à partir de l'image original, la pyramide hiérarchique.
2. On code l'image $I_{1/n}$ de plus basse résolution 1/n et on l'ajoute aux données d'image comprimées. On décode l'image 1/n (notée $\hat{I}_{1/n}$).
3. On construit, par interpolation de $\hat{I}_{1/n}$, l'image de résolution 2/n (notée $\hat{I}_{2/n}$). On calcule la différence $I_{2/n} - \hat{I}_{2/n}$, on la code et on l'ajoute aux données d'image comprimées. On décode la différence $I_{2/n} - \hat{I}_{2/n}$ et on reconstruit $\hat{\hat{I}}_{2/n}$.
4. On construit, par interpolation de $\hat{\hat{I}}_{2/n}$ l'image de résolution 4/n (notée $\hat{I}_{4/n}$) et on reprend le même processus qu'à l'étape 2 pour ce niveau de résolution.
5. On traite de la même manière tous les niveaux de la pyramide jusqu'à l'image pleine résolution.

Il faut noter que :

- L'image utilisée pour reconstruire le niveau suivant de la pyramide par interpolation est forcément l'image décodée car c'est la seule connue du décodeur. On évite ainsi la propagation d'erreurs, à chaque itération de la reconstruction, au niveau du décodeur.
- La qualité de l'image reconstruite peut théoriquement être meilleure que par l'algorithme direct à base de TCD.
- Le processus de codage peut être avec ou sans pertes.
- On peut utiliser le mode progressif ou séquentiel.

4.2.1.5 Les processus de codage

Le tableau suivant donne un résumé des caractéristiques essentielles des différents processus de codage définis dans la norme JPEG. Il a été fait abstraction dans ce document des considérations sur l'entrelacement des sources multiples et donc des balayages multiples qui sont utilisés quand on souhaite coder plusieurs images en même temps en les entrelaçant.

processus	caractéristiques	applications
processus de base (requis pour tous les décodeurs basés TCD)	• processus basé TCD. • pixels 8 bits pour chaque composante. • séquentiel. • codage de Huffman.	compression d'images de nature photographique (informatique ou vidéo). à la base de MPEG.
processus basés TCD étendue	• processus basé TCD. • pixels 8 bits ou 12 bits. • séquentiel ou progressif. • codage de Huffman ou codage arithmétique.	peu ou pas utilisé mais potentiellement utile sans le codage arithmétique.
processus sans perte	• processus prédictif. • pixels 2 à 16 bits. • séquentiel. • codage de Huffman ou codage arithmétique.	imagerie médicale (peu ou pas utilisé).
processus hiérarchiques	• trames multiples (différentielles ou non). • utilise des processus basés TCD étendue ou sans perte.	peu ou pas utilisé.

En fait, quand on parle de compression selon la norme JPEG, on sous-entend généralement qu'il s'agit du processus de base car les autres sont très peu (voir pas du tout) utilisés.

4.2.1.6 Les extensions hors norme

Le problème de la normalisation quand les demandes du marché sont très fortes, c'est que le moindre retard produit des normes de fait qui ont tendance à empêcher les vrais normes retardataires de se diffuser. Ce problème n'a pas manqué de se produire pour JPEG, mais beaucoup moins pour MPEG. Trois domaines ont été concernés par ces extensions hors normes :

- Les tests de conformité. La norme JPEG spécifiait deux aspects de la compression d'image fixe ; les processus de décodage et la syntaxe du train binaire, et les tests permettant de vérifier la conformité d'un produit matériel ou logiciel avec la norme. Ces tests de conformité sont essentiels à la bonne diffusion de la norme afin que chacun puisse vérifier la qualité des produits disponibles sur le marché. Hors ces tests sont arrivés avec deux ans de retard pour JPEG et c'est le circuit CL550 de la société C-CUBE qui a servi à tester la conformité des autres produits commercialisés. Pendant quelques temps, l'inter-opérabilité des équipements a posé problème, ce qui a nuit à la diffusion de la norme.

- Le format du fichier d'échange. Pour des raisons diverses, le format SPIFF (Still Picture Interchange File Format) spécifiant le fichier informatique contenant les données JPEG (et autres) est arrivé avec 4 années de retard sur la norme de compression. C'est donc la société C-CUBE qui a définit le format JFIF (JPEG File Interchange Format) qui est utilisé depuis le début sous le vocable JPEG. Il est peu probable que le format SPIFF remplacera brutalement le format JFIF même s'il lui est supérieur.

- La compression de séquences vidéo basée JPEG. La norme MPEG est conçue pour la compression des séquences d'images. A la différence de JPEG, la complexité du processus n'est pas répartie de manière symétrique entre le codeur et le décodeur. La complexité du codeur est élevée (donc son coût est élevé) alors que le décodeur n'est implémenté que dans un seul circuit intégré. De plus, de part sa structure, on ne peut accéder directement à chaque image (accès aléatoire) dans le train binaire ce qui pose un problème pour le montage vidéo. Pour remédier à ces problèmes, des solutions propriétaires de compression

de séquences d'images basées sur JPEG ont été développées : on les regroupe sous l'acronyme MJPEG (Motion JPEG). Chaque image est comprimée indépendamment par un codeur JPEG et une couche syntaxique supplémentaire est ajoutée à la syntaxe JPEG pour former un train binaire vidéo. Un train binaire audio comprimé est généralement rajouté pour former le train binaire MJPEG. On retrouve cette solution dans des cartes de compression vidéo sur PC ou dans des magnétoscopes professionnels comme le Digital Betacam de Sony. Le MJPEG n'étant pas normalisé, il n'y a pas d'inter-opérabilité entre les équipements.

4.2.2 Le processus de base dans JPEG

4.2.2.1 Décorrélation d'un bloc d'image

4.2.2.1.1 Introduction

La corrélation entre deux échantillons x(k) et x(l) d'un signal discret x(n) exprime la similitude existant entre ces échantillons. Si la corrélation est élevée, il y a redondance entre les informations transmises. Pour diminuer le débit d'information du signal x(n) que l'on veut transmettre, il faut réduire sa redondance et donc minimiser cette corrélation. Il y a deux méthodes pour décorréler un signal :

1. On peut, connaissant la valeur de x(k), prédire la valeur de x(l), $\hat{x}(l)$, et ne transmettre que la différence $x(l) - \hat{x}(l)$. On élimine ainsi la part d'information commune entre x(k) et x(l). Cette méthode, appelée codage par prédiction, est à la base de la MIC différentielle.

2. On peut aussi chercher directement une transformation mathématique réversible (pour pouvoir reconstituer le signal d'origine).

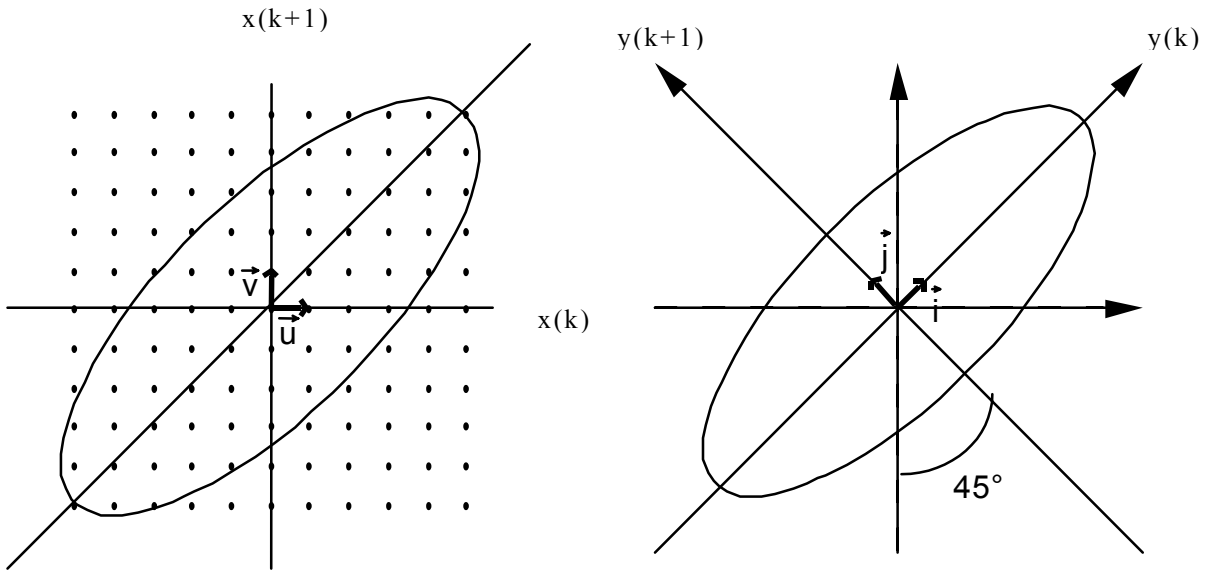
Dans le cas du traitement de l'image, à qualité équivalente, le codage par transformation permet des taux de compression plus élevés que le codage par prédiction. De plus, ce type de traitement est moins sensible aux erreurs de transmission.

Prenons un exemple. Soit un espace vectoriel à deux dimensions E et un repère orthonormé $(\vec{u}, \vec{v})$ dans cet espace. Soit deux échantillons adjacents x(k) et x(k+1) d'un signal x(n). On suppose que l'amplitude de ces échantillons est quantifiée sur 11 niveaux. On va placer les

valeurs de x(k) sur l'axe des $\vec{u}$ et les valeurs de x(k+1) sur l'axe des $\vec{v}$ pour k variant de 0 à N (N grand). La corrélation entre ces échantillons est telle que les combinaisons les plus favorables se situent dans une région proche de la droite x(k+1) = x(k). C'est la région elliptique indiquée sur la figure suivante. On effectue une rotation de 45° sur les axes, ce qui nous donne un nouveau repère $(\vec{i}, \vec{j})$ et deux nouveaux échantillons y(k) et y(k+1). Cette rotation permet de modifier la valeur des variances des deux échantillons. On est passé de:

$$s^2_{x(k)} \cong s^2_{x(k+1)} \quad \Rightarrow \quad s^2_{y(k)} > s^2_{y(k+1)}$$

Le nombre d'états possibles de y(k) est resté le même que celui de x(k). Par contre, la dynamique de y(k+1) est beaucoup plus faible que celle de x(k+1), ce qui implique que le nombre de bits permettant de coder y(k+1) est plus faible que celui nécessaire au codage de x(k+1). Il y a donc bien diminution du débit binaire. La rotation inverse permet de retrouver les échantillons de départ.



Le changement de base effectué dans cet exemple peut facilement être étendu à N échantillons corrélés se trouvant dans un espace vectoriel à N dimensions grâce à une transformation du type :

$$Y = A.X$$

A étant la matrice de changement de base. Dans la pratique, seules les transformations orthogonales sont utilisées parce que A est alors aisément inversible par la relation :

$$A^{-1} = A^{t}$$

Il est donc facile de revenir au vecteur d'origine X. L'orthogonalité de A est vérifiée si ses vecteurs lignes et ses vecteurs colonnes sont orthogonaux deux à deux.

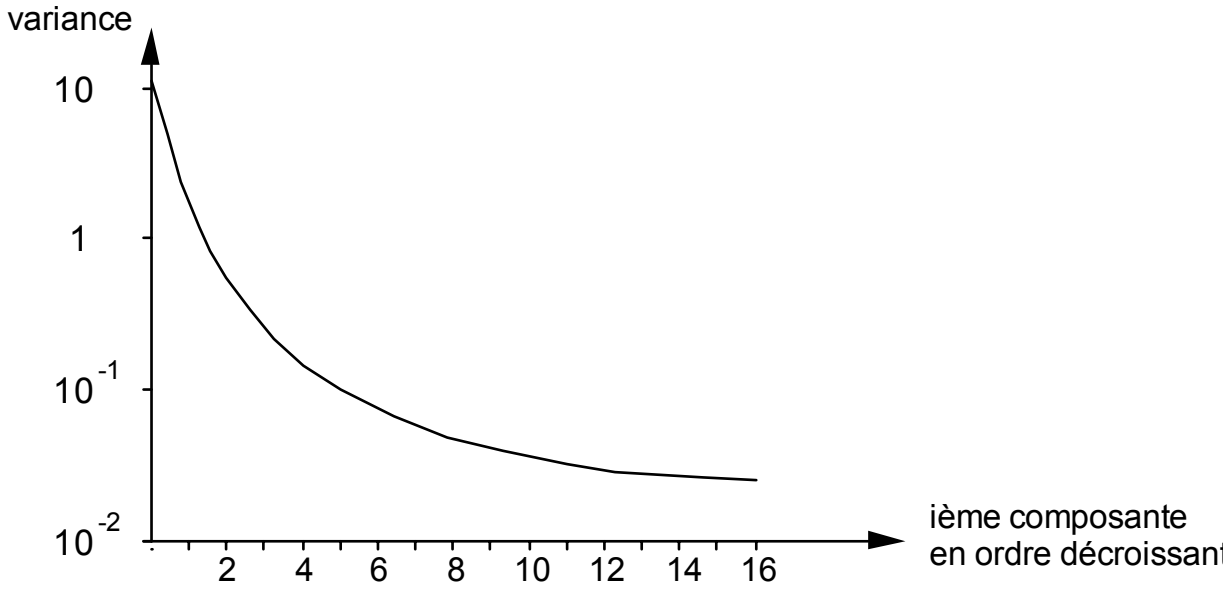
4.2.2.1.2 La base de KARHUNEN et LOEVE

La meilleure base est celle qui permettrait d'obtenir des échantillons indépendants. Hélas, on n'a pas trouvé de transformation inversible permettant d'y aboutir. Le mieux que l'on sache faire est de déterminer une base dans laquelle les échantillons sont non-corrélés. La transformation permettant de passer dans cette base a été élaborée par KARHUNEN et LOEVE dans le domaine des signaux continus. L'extension de cette méthode dans le domaine des signaux discrets s'appelle la transformation de HOTELLING. On peut considérer le vecteur image comme un signal essentiellement aléatoire. Dans une base donnée, ses composantes sont autant de variables aléatoires auxquelles correspond une matrice de variance-covariance R. Pour éliminer la redondance entre les pixels voisins et comprimer le signal, on va diagonaliser R. On peut démontrer que la détermination de la base de KARHUNEN-LOEVE (transformation de HOTELLING) revient à la recherche des vecteurs propres de la matrice de corrélation du vecteur image.

Appelons F_{mn} le vecteur image d'origine et F_{uv} le vecteur image transformé après le changement de base. Pour un u,v donné, le vecteur transformé Fuv est une variable aléatoire dont la variance est égale à luv. Contrairement à Fmn dont les variances sont toutes égales, les luv sont variables. On peut tracer la courbe des luv classées par ordre décroissant et démontrer que la décroissance de cette courbe est maximale dans la base de K&L. La transformation de HOTELLING est dite optimale dans le sens de la compression des données. En effet, comme l'énergie du signal est concentrée sur un nombre minimum de coefficients, on peut annuler les composantes de faible amplitude restantes en provoquant peu de défauts sur l'image. Comme les composantes de faible variance correspondent en général aux fréquences spatiales élevées, on observera alors une perte de résolution.

En première approximation, on modélise souvent le signal d'image par un processus de MARKOV monodimensionnel caractérisé par le coefficient r de corrélation entre deux pixels adjacents. On prend pour r des valeurs comprises entre 0,9 et 0,95. La courbe suivante

représente les variances, classées par ordre décroissant, des composantes d'un vecteur d'image 4x4 issu d'un processus de MARKOV du premier ordre dans la base de K&L avec r = 0,95.



Comme la transformation de HOTELLING est optimale, cette courbe représente un minimum. Le principal avantage de la transformation de HOTELLING est de tenir compte de la statistique de l'image et donc d'être optimale au sens de la compression des données. Cette transformation a aussi de nombreux inconvénients :

1. La matrice de corrélation R n'est pas toujours diagonalisable.
2. Il n'existe pas d'algorithme rapide permettant de traiter ce problème.
3. Il faut recalculer la matrice de transformation pour chaque image.

Ces difficultés rendent la transformation de HOTELLING inutilisable en pratique. On va donc essayer de trouver une base moins performante (sous optimale), mais permettant une réalisation pratique de la décorrélation. La base de KARHUNEN-LOEVE, qui représente la limite théorique à atteindre, servira d'élément de comparaison.

4.2.2.1.3 La transformée en cosinus discrète

Il existe bien des bases orthogonales envisageables. Elles ne présentent un intérêt que lorsque les composantes du vecteur image y sont décorrélées efficacement et que leurs matrices de passage sont faciles à mettre en œuvre. On peut citer :

- La transformation de WALSH-HADAMART.

- La transformation de FOURIER.
- La transformation de HAAR.
- La transformation en cosinus discrète (TCD).

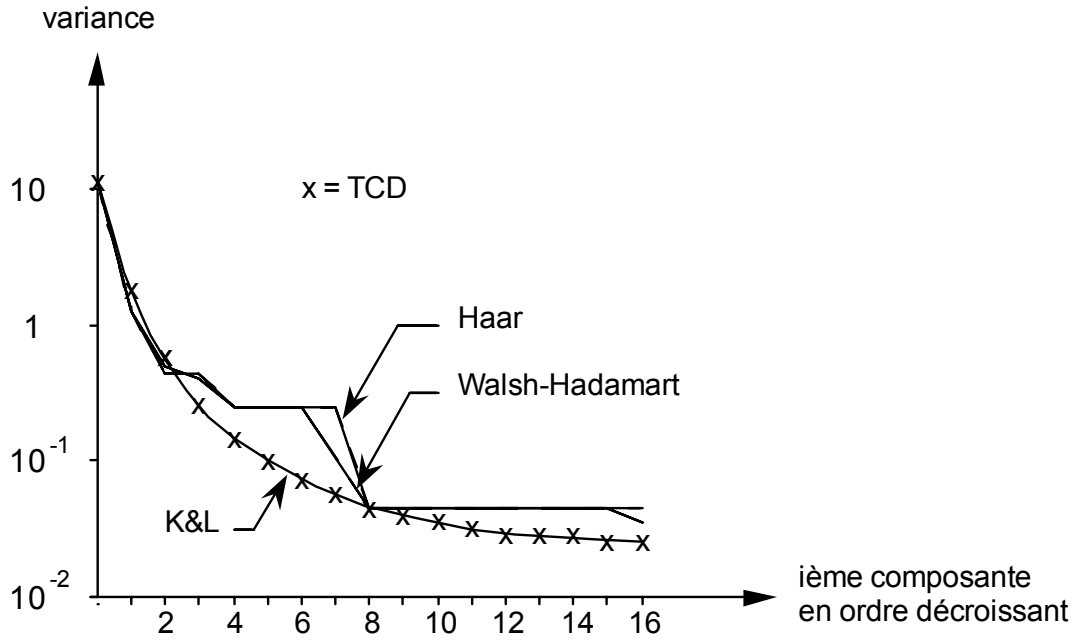
Ces transformations sont dites sous-optimales puisqu'elles tendent, au sens de l'efficacité de la compression de données, vers la limite théorique représentée par la transformation de HOTELLING. Elles ont donné lieu à de nombreux travaux, mais la transformation en cosinus discrète semble être le meilleur compromis vitesse-efficacité.

La TCD a été définie par AHMED, NATARAJAN et RAO, pour un signal x(n), n = 0, 1,..., N-1, par :

- $X(0) = \frac{\sqrt{2}}{N} . \sum_{n=0}^{N-1} x(n)$

- $X(k) = \frac{2}{N} . \sum_{n=0}^{N-1} x(n) . \cos\left(\frac{k . \pi . (2.n + 1)}{2.N} \right)$ avec $k = 1, 2, ..., N - 1$

Elle fournit la meilleure approximation de la base de KARHUNEN-LOEVE, dans le cas où le signal d'image est modélisé par un processus de MARKOV du premier ordre (coefficient de corrélation élevé), pour la décroissance des variances transformées et pour les vecteurs représentant la nouvelle base. Considérons un signal Markovien du premier ordre, de longueur 16 et de coefficient de corrélation 0,95. La figure suivante montre les échantillons, classés par ordre des variances décroissantes, du signal transformé par les méthodes de HAAR, WALSH-HADAMART, K&L et TCD. La TCD concentre l'énergie sur un faible nombre de coefficients aussi bien que la transformation de HOTELLING (base de K&L).



Si on calcule les vecteurs composant la base de K&L et la base de la TCD d'un signal Markovien monodimensionnel de longueur N=8 et de corrélation 0.9, on constate qu'ils sont presque identiques à une différence de phase de π près. La TCD inverse est obtenue en inversant la matrice de passage de la TCD directe. Il vient:

$$x(n) = \frac{1}{\sqrt{2}}.X(0) + \sum_{k=1}^{N-1} X(k).\cos\left(\frac{k.\pi.(2.n+1)}{2.N}\right)$$

La TCD peut être calculée à partir d'une transformation de Fourier discrète (TFD). Il faut prendre le signal x(n) (de taille N) à transformer puis le rendre pair par un effet de miroir (et donc doubler sa taille). Les N premiers points à la sortie de la FFT correspondent en module aux N points calculés par la TCD.

Vis à vis des différentes transformations ayant pour but de comprimer le débit d'informations d'un signal d'image, la TCD offre les avantages suivants :

- c'est la meilleure approximation de la transformation de HOTELLING,
- on peut la calculer avec des algorithmes rapides (même principe que pour la FFT),
- ses composantes sont réelles (donc les calculs sont plus simples que dans le cas de la TFD),

- compte tenu de diverses approximations, le signal transformé représente les fréquences du signal d'origine (dans le cas d'un signal aléatoire comme la voix ou l'image).

Pour toutes ces raisons, la TCD est aujourd'hui utilisée dans beaucoup d'applications ayant pour but la compression de l'image avec perte d'informations. Dans ce cadre, on effectue la transformation sur des blocs d'image de dimension NxN. Il faut donc utiliser la TCD bidimensionnelle (TCD-2D) définie par:

$$X(u,v) = \frac{4.C(u).C(v)}{N^2}.\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n).\cos\left(\frac{\pi.u(2.m+1)}{2.N}\right).\cos\left(\frac{\pi.v(2.n+1)}{2.N}\right)$$

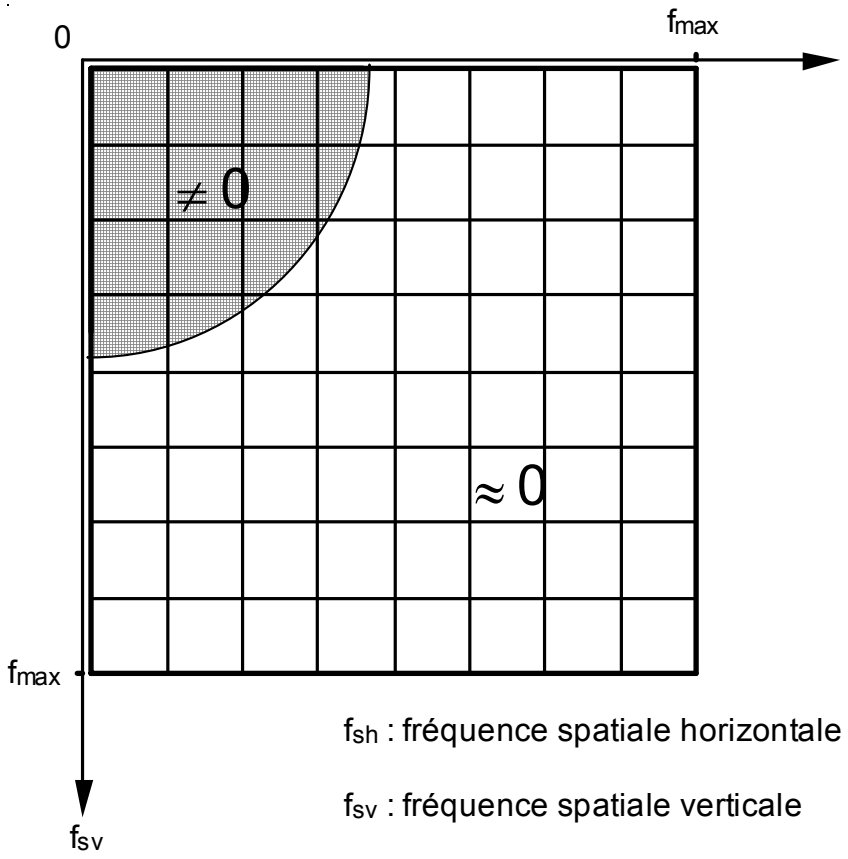
$$\text{avec} \quad C(k) = \frac{1}{\sqrt{2}} \quad \text{si} \quad k = 0, \quad C(k) = 1 \quad \text{si} \quad k \neq 0$$

La TCD-2D inverse vaut:

$$x(m,n) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u).C(v).X(u,v).\cos\left(\frac{\pi.u(2.m+1)}{2.N}\right).\cos\left(\frac{\pi.v(2.n+1)}{2.N}\right)$$

$$\text{avec} \quad C(k) = \frac{1}{\sqrt{2}} \quad \text{si} \quad k = 0, \quad C(k) = 1 \quad \text{si} \quad k \neq 0$$

La taille optimale des blocs d'image traités est égale à 8x8. C'est celle qui est utilisée dans JPEG. Comme la TFD bidimensionnelle, la TCD-2D est séparable. Ainsi, la transformée à deux dimensions peut se calculer en utilisant 2N fois l'algorithme de la transformée monodimensionnelle, une fois sur chaque ligne de la matrice image, puis une fois sur chaque colonne. L'interprétation physique de la TCD-2D est assimilable à celle de la TFD-2D. La matrice issue de la transformation représente les fréquences spatiales de l'image suivant une direction donnée.



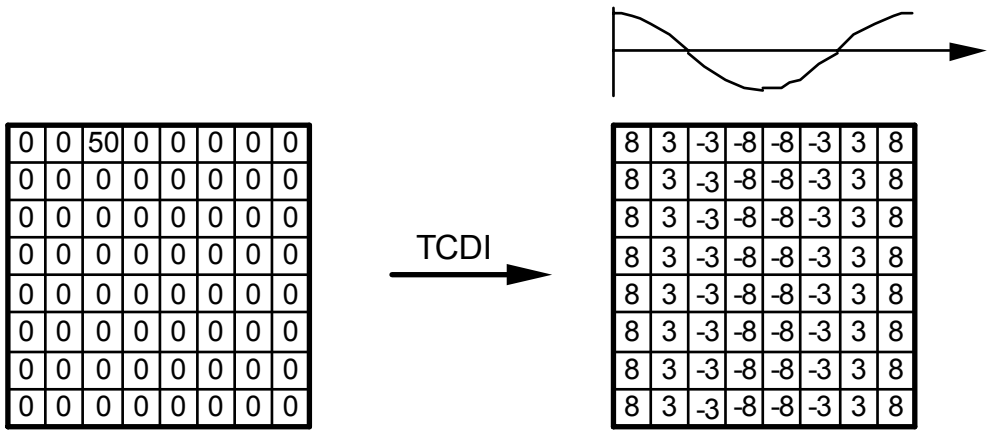
L'intérêt de la TCD-2D (vis à vis de la TFD-2D par exemple) réside dans la concentration de l'énergie sur un nombre minimum de coefficients. Comme le spectre du signal d'image est décroissant en fonction de la fréquence, l'énergie se trouve en général sur les composantes représentant les fréquences spatiales les plus faibles. La TCD-2D est la transformation permettant d'obtenir le plus de coefficients négligeables.

4.2.2.2 Quantification psychovisuelle

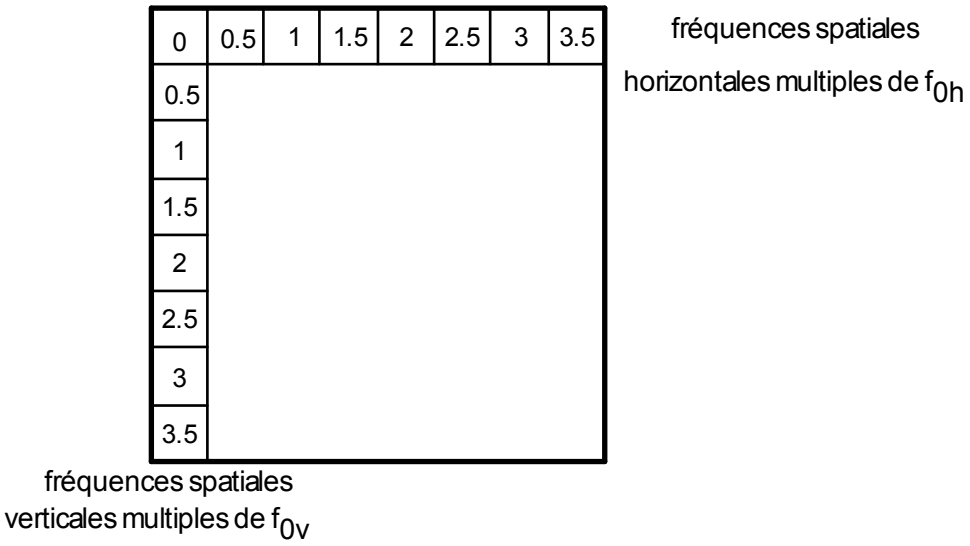
4.2.2.2.1 Seuil de perception des fréquences spatiales

4.2.2.2.1.1 Fréquences spatiales et TCD

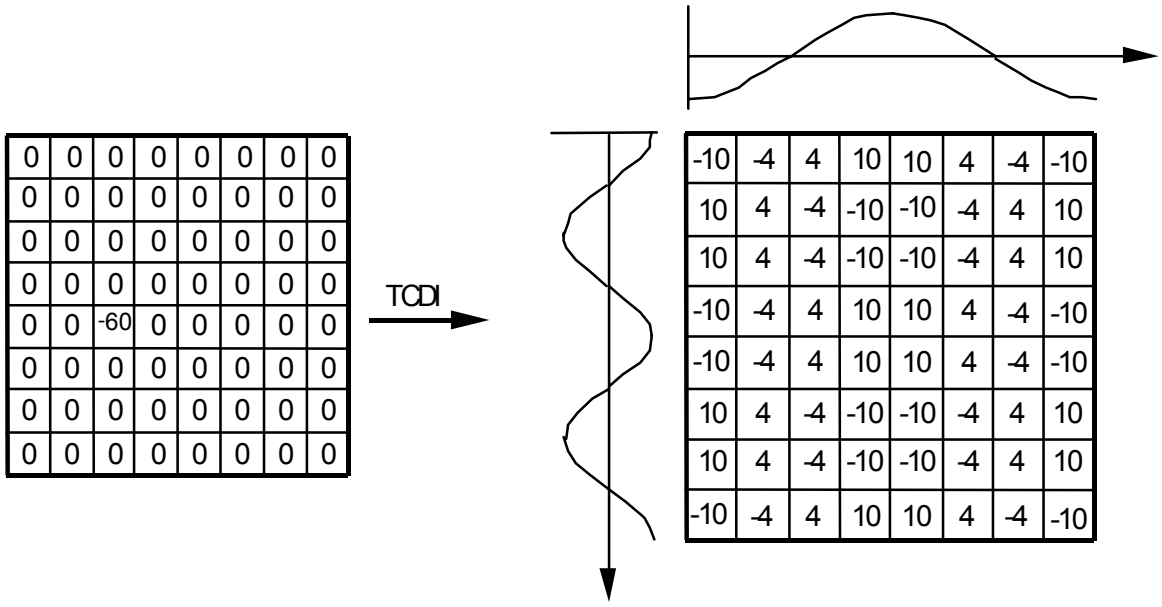
La TCD a permis de décorréler le bloc d'image traité et de passer dans l'espace des fréquences spatiales. Les coefficients de la première ligne de la matrice 8x8 ainsi obtenue correspondent aux fréquences spatiales horizontales, ceux de la première colonne aux fréquences spatiales verticales. Sur l'exemple suivant, on voit qu'au coefficient (0,2) correspond une période spatiale sur 8 pixels, la valeur du coefficient déterminant l'amplitude de l'oscillation. C'est la fréquence fondamentale horizontale f_{0h} .



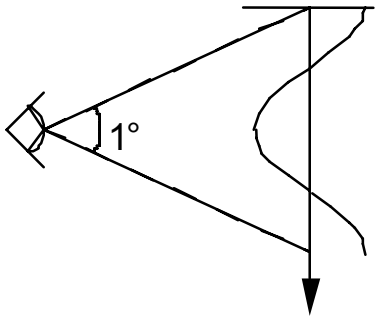
Les autres coefficients de la première ligne (ou de la première colonne) sont des multiples de la fréquence fondamentale.



Par exemple, le coefficient (0,5) correspond à 2,5 périodes spatiales sur 8 pixels. L'élément (0,0) représente la valeur moyenne du bloc d'image. Il faut noter que dans JPEG, les valeurs des pixels ont été décalées de 128 pour obtenir une représentation signée (dynamique : -128 à +127). Quand le coefficient (0,0) est nul, cela veut dire que la valeur moyenne du bloc vaut 128. Les coefficients ne se trouvant ni sur la première ligne, ni sur la première colonne (voir un exemple sur la figure suivante) représentent des combinaisons de fréquences spatiales horizontales (fsh) et verticales (fsv).

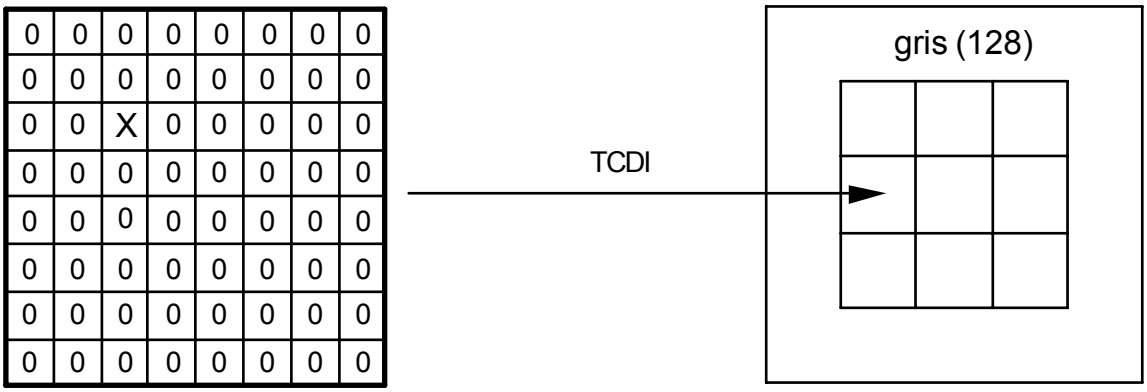


Pour s'affranchir des notions de distance d'observation et de taille de bloc, on utilise le cycle par degré (c/°) comme unité de fréquence spatiale. Un cycle par degré, cela veut dire que l'on voit une période spatiale sous un angle apparent d'un degré



4.2.2.2.1.2 Seuil de perception

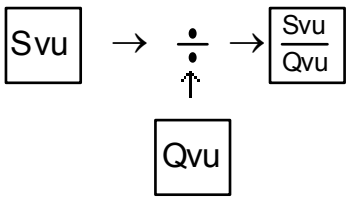
On va déterminer, pour chaque fréquence spatiale, le seuil à partir duquel on voit apparaître une différence juste discernable (jnd : just noticeable difference) entre le bloc uniforme et le bloc comportant une fréquence. Prenons par exemple l'élément (2,2) d'une matrice de luminance. On place sur un fond gris à mi-niveau (128) plusieurs blocs identiques calculés par transformation inverse de la matrice dont tous les éléments sont nuls, sauf le coefficient (2,2).



On augmente X à partir de 0 jusqu'au moment où on voit apparaître une différence entre le fond et le milieu de l'image. La valeur de X obtenue est le seuil de perception de la fréquence spatiale (2,2). Cette opération doit être effectuée avec plusieurs observateurs et selon un mode expérimental rigoureux. Les premiers systèmes de codage utilisant les seuils de perception pour comprimer les images masquaient toutes les valeurs inférieures à ces seuils mais ne modifiaient pas le pas de quantification pour les valeurs restantes.

4.2.2.2.1.3 Quantification par les seuils de perception

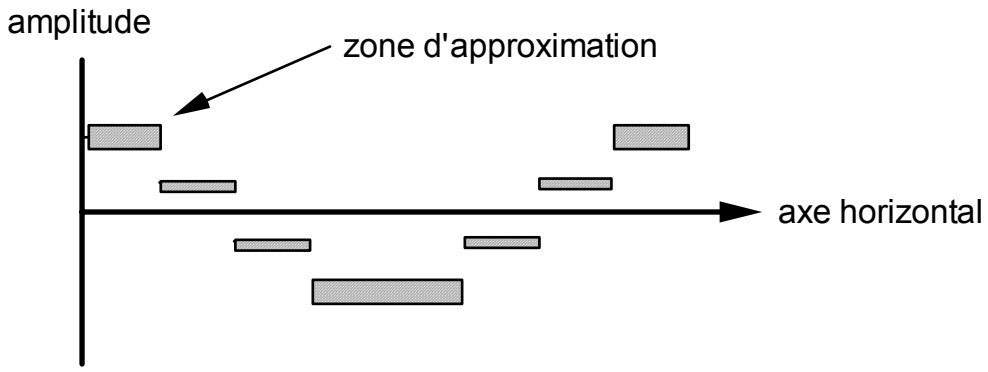
Dans JPEG, les coefficients transformés (Svu) sont quantifiés linéairement par les seuils de perception (Qvu) correspondants, en effectuant une simple division.



Le passage du masquage par les seuils à la quantification linéaire par les seuils suppose que l'écart d'amplitude discernable pour une fréquence donnée soit le même quelle que soit cette amplitude et qu'il soit identique au seuil de perception. Ces hypothèses ne paraissent pas irréalistes à priori. Voyons sur un exemple à quoi correspond exactement la quantification psychovisuelle. Le seuil de perception de la fréquence fondamentale horizontale (coefficient (0,2)) est égal à 10. Les multiples de cette valeur nous donnent les amplitudes d'oscillations suivantes sur le bloc image :

coef (0,2)	première ligne de la matrice image							
10	2	1	-1	-2	-2	-1	1	2
20	3	1	-1	-3	-3	-1	1	3
30	5	2	-2	-5	-5	-2	2	5
40	7	3	-3	-7	-7	-3	3	7
50	8	3	-3	-8	-8	-3	3	8

Ainsi, il y aura approximation par l'amplitude correspondant à la valeur 30 de toutes les amplitudes correspondant aux valeurs comprises entre 25 et 35.



S'il y avait 8 bits attribués à ce coefficient avant quantification (intervalle: -128 à 127), on passe après quantification à un intervalle compris entre -13 et +13, soit une attribution de 5 bits. On est bien arrivé au résultat recherché ; au lieu d'attribuer uniformément un même nombre de bits à toutes les fréquences spatiales, on alloue ces bits en fonction de données physiologiques que nous allons maintenant étudier.

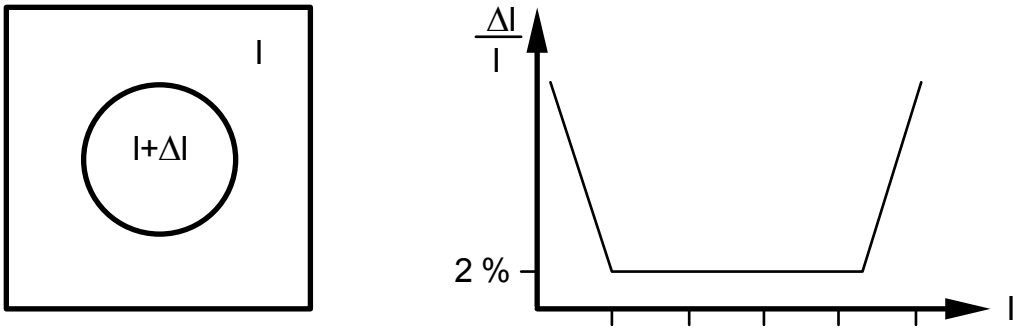
4.2.2.2.2 Modèle de vision des fréquences spatiales

4.2.2.2.2.1 Généralités

Nous allons voir dans ce paragraphe un modèle simple du système de visualisation humain. Il n'est pas question de faire une étude exhaustive de l'état de la recherche dans ce domaine, mais plutôt de synthétiser les résultats concernant la sensibilité au contraste et la perception des fréquences spatiales. La rétine est tapissée de photorécepteurs composés de cônes et de bâtonnets. Seuls les cônes nous intéressent car on se place dans le cas de la vision photopique (luminosité élevée). Nous allons étudier dans un premier temps la sensibilité au contraste.

4.2.2.2.2.2 Sensibilité au contraste

Considérons l'expérience suivante qui mesure le seuil différentiel de luminance :



Le cercle d'intensité I + ΔI est entouré d'un fond d'intensité I. On augmente ΔI jusqu'à ce qu'on observe une différence entre le cercle et le fond. La valeur ΔI / I est appelée fraction de Weber. A l'exception des très faibles et très fortes valeurs de I, elle est constante et vaut environ 0,02. Nous allons supposer que les seuils différentiels pour les signaux de chrominances Dr et Db suivent la même loi que pour la luminance. Nous supposerons aussi que la valeur de la fraction de Weber est à peu près identique en luminance et en chrominance.

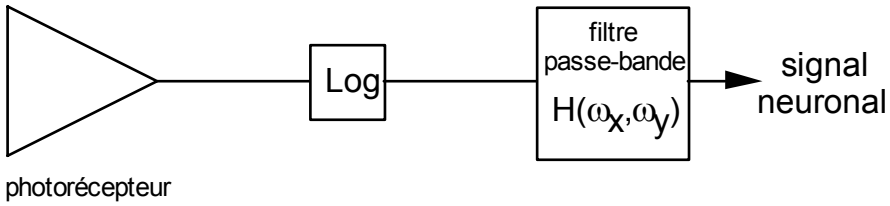
4.2.2.2.2.3 Modèle logarithmique

L'œil a une réponse non-linéaire variant comme le logarithme de l'excitation. Cette non-linéarité intervient juste après les photorécepteurs de la rétine. Pour modéliser simplement la vision monochromatique, on va faire deux hypothèses simplificatrices.

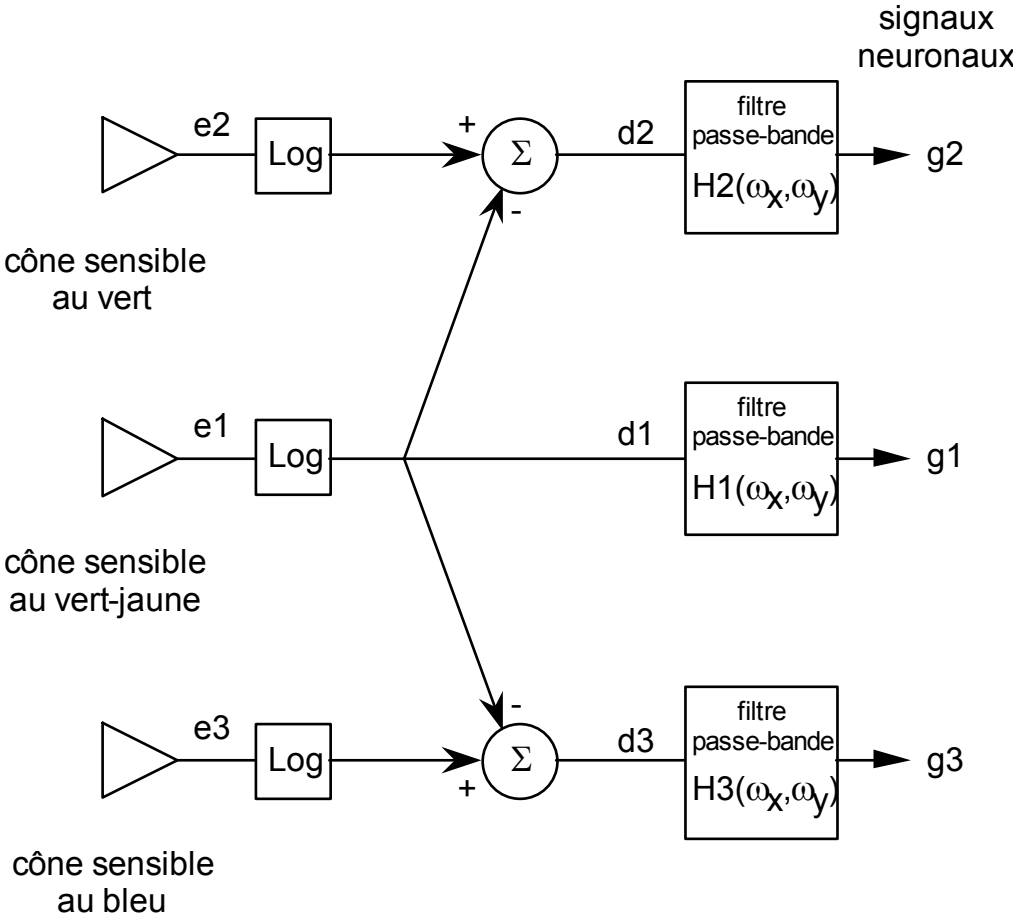
- L'œil réagit indépendamment aux différentes fréquences spatiales. On peut donc modéliser sa réponse par un système linéaire (filtre passe-bande bidimensionnel).

- La sensibilité relative de l'œil est la même pour les fréquences spatiales horizontales et verticales. Elle est diminuée de moitié pour les fréquences spatiales se situant sur la diagonale de l'image. A partir des mesures de seuil des fréquences horizontales, on pourra déterminer la fonction de transfert bidimensionnelle $H(\omega_x, \omega_y)$ du filtre passe-bande.

Ces hypothèses, vérifiées par l'expérience, conduisent au modèle monochromatique de la figure suivante. Ce modèle logarithmique est une approximation correcte du système de vision monochrome si on reste dans la gamme des intensités moyennes et si on ne monte pas trop haut en fréquence.



La théorie trichromatique de la vision des couleurs postule l'existence de trois types de cônes sensibles chacun à une gamme de longueurs d'ondes différentes. L'extension du modèle précédent à cette théorie conduit au diagramme suivant qui offre une analogie remarquable avec les systèmes de télévision.



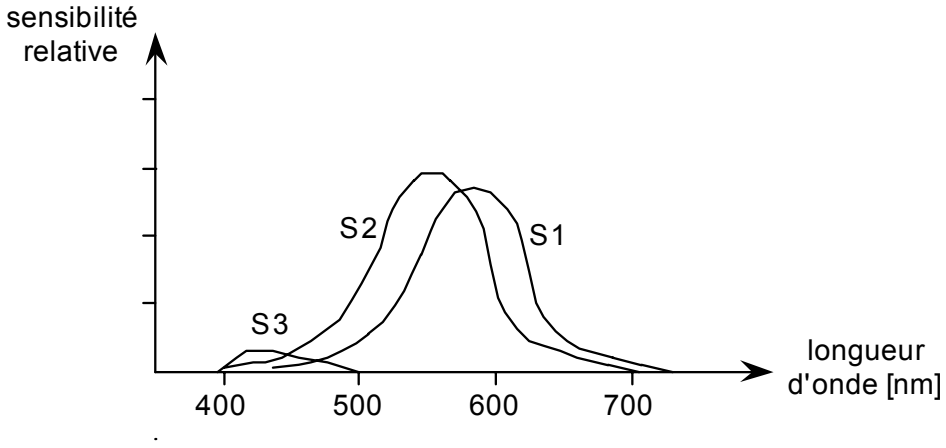
Dans ce modèle proposé par Frei en 1974, les trois sortes de récepteurs ont une sensibilité spectrale S1(l), S2(l), S3(l) et produisent les signaux :

$$e_1 = \int_{visible} C(\lambda) \,.\, S1(\lambda) \,.\, d\lambda$$

$$e_2 = \int_{visible} C(\lambda) \,.\, S2(\lambda) \,.\, d\lambda$$

$$e_3 = \int_{visible} C(\lambda) \,.\, S3(\lambda) \,.\, d\lambda$$

où C(λ) est la distribution spectrale d'énergie de la lumière incidente. Les trois courbes d'absorption spectrale sont difficiles à mesurer précisément. Par le biais d'estimations indirectes, on obtient les courbes suivantes :



Les trois signaux e1, e2, e3 sont alors soumis à une fonction de transfert logarithmique et combinés pour produire les sorties :

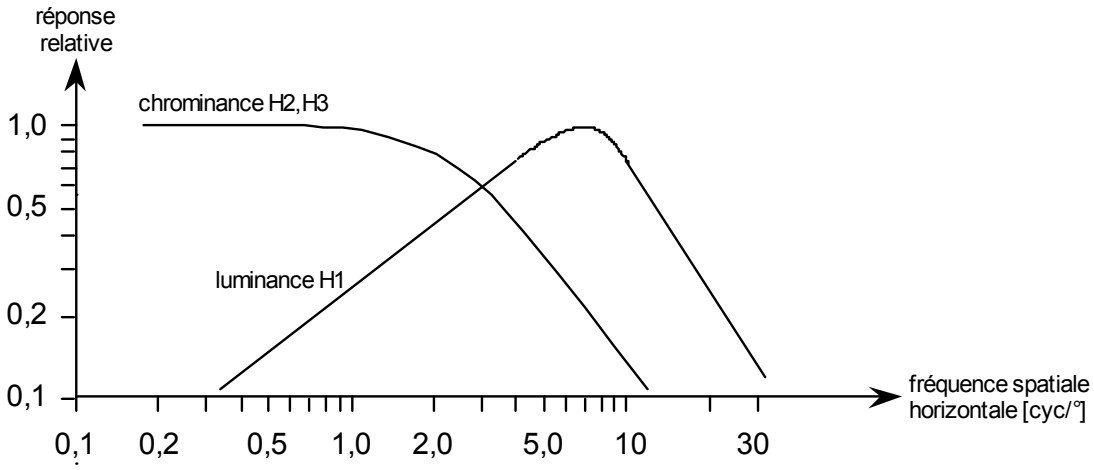
$$d1 = \log(e1)$$

$$d2 = \log(e2) - \log(e1) = \log\left(\frac{e2}{e1}\right)$$

$$d3 = \log(e3) - \log(e1) = \log\left(\frac{e3}{e1}\right)$$

Finalement, d1, d2 et d3 passent à travers des filtres passe-bande pour donner les signaux destinés au cerveau g1, g2, g3. Dans ce modèle, les signaux d2 et d3 sont liés à la chrominance de la lumière incidente, alors que d1 est proportionnel à sa luminance. Ce

modèle prédit un grand nombre de propriétés de la vision des couleurs et respecte les bases de la colorimétrie. Il respecte notamment le phénomène d'invariance de la teinte et de la saturation des couleurs lorsque la luminance de la source varie. On a pu estimer les fonctions de transfert en luminance et en chrominance suivantes :

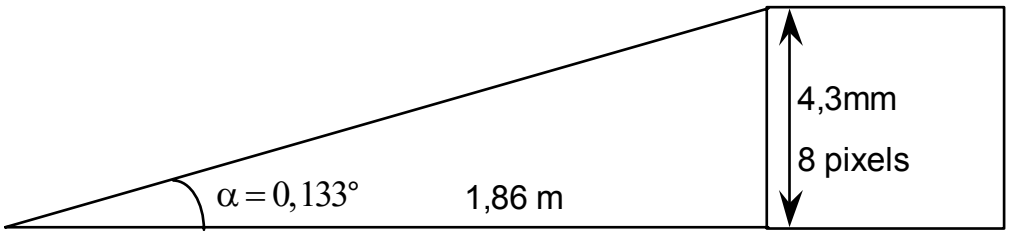


La réponse pour la luminance H1 est la même que la réponse H du modèle monochrome. Les hypothèses qui ont été faites pour ce modèle sont toujours valables. On peut donc déduire les fonctions de transfert bidimensionnelles à partir de la réponse aux fréquences spatiales horizontales.

4.2.2.2.2.4 Détermination des tables de quantification JPEG

L'image de télévision est composée d'un plan luminance Y et de deux plans chrominance Dr et Db. Dans le processus JPEG, on traite indépendamment les trois plans avec une table de quantification pour la luminance et une table de quantification pour la chrominance. Pour relier les résultats expérimentaux au modèle précédent, nous allons assimiler le signal g1 au signal de luminance Y et les signaux g2,g3 aux signaux de chrominance Dr et Db. Les courbes de réponse en fréquence de l'oeil sont donc valables pour Y, Dr et Db.

En considérant une image au format CCIR601 observée à une distance égale à 6 fois la hauteur de l'écran, un bloc de 8x8 pixels apparaît sous un angle de 0,133 °. Sur la figure suivante, les calculs ont été fait avec un moniteur 51 centimètres.

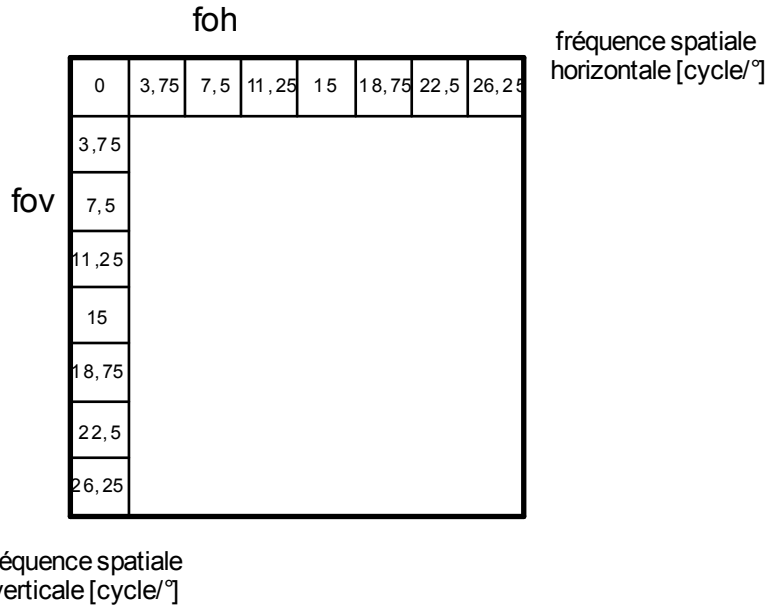


$$\alpha = \text{arctg}\left(\frac{4{,}3 \cdot 10^{-3}}{1{,}86}\right) = 0{,}133\ °$$

La fréquence fondamentale fo est donc de :

$$1 \text{ cycle} \rightarrow 0{,}133° \Rightarrow \text{fo} = 7{,}5 \text{ cycle/degré}$$

Les coefficients de la première ligne et de la première colonne correspondent aux multiples de la fréquence spatiale fondamentale. Cela nous donne :



En exploitant (avec diverses approximations) les courbes de réponse en fréquence de l'œil, on peut donc déduire les tables de quantifications recommandées (mais pas normalisées) dans

JPEG. La table de quantification définie pour la luminance (c'est à dire les seuils de perception) est la suivante :

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

La table de quantification définie pour la chrominance est la suivante:

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

4.2.2.3 Codage entropique

4.2.2.3.1 Généralités

Les étapes précédentes (TCD plus quantification) ont permis d'obtenir une matrice 8x8 ayant un maximum de coefficients nuls. Il faut maintenant appliquer un codage astucieux utilisant cette propriété pour obtenir le taux de compression le plus élevé possible. Nous appellerons coefficient DC l'élément (0,0) de la matrice. Il représente la valeur moyenne du bloc 8x8 pixels de l'image d'origine.

DC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC

Les 63 autres éléments de la matrice représentent les différentes fréquences spatiales du bloc d'image. Ils seront appelés coefficients AC. On fixe un modèle de codage par type de coefficient. Il y a deux types de codage statistique envisagés dans JPEG : le codage de Huffman et le codage arithmétique. Seul le codage de Huffman sera étudié dans ce cours.

4.2.2.3.2 Modèles de codage

4.2.2.3.2.1 Introduction

Pour pouvoir garder des tables de Huffman de taille raisonnable, on n'a pas associé un code à chaque valeur possible des coefficients. En effet, le codeur doit pouvoir spécifier au décodeur les tables qu'il va utiliser pour comprimer une image dans un temps relativement bref. De plus, des tailles de table trop élevées auraient pénalisé la vitesse du processus. Aussi, on n'a associé un code de Huffman qu'à la position du bit de poids fort de la valeur à coder, les bits restants étant supposés à distribution aléatoire et donc codés en binaire naturel (ou complément à 2).

4.2.2.3.2.2 Coefficient DC

Le coefficient DC fait l'objet d'un codage différentiel utilisant un prédicteur à une dimension, qui est la valeur DC du bloc 8x8 le plus récemment codé. La différence DIFF, qui fera l'objet du codage, est obtenue par la relation :

DIFF = coefficient_DC - prédiction

Comme on l'a vu dans le paragraphe sur la compression sans pertes, on assigne à DIFF un code de Huffman, représentant la position du bit de poids fort, suivi de ses bits de poids faibles qui spécifient le signe et la valeur exacte de son amplitude. Les valeurs de DIFF, en complément à 2 (CA2), sont groupées en 12 catégories. Un code de Huffman est associé à chaque catégorie. Quand DIFF est positive, les SSSS bits de poids faibles de DIFF (les extras-bits) sont mis à la suite du code de Huffman. Quand DIFF est négative, les SSSS bits de poids faibles de (DIFF-1) suivent le code de Huffman.

4.2.2.3.2.3 Coefficients AC

La matrice des coefficients DCT quantifiés est ré-ordonnée selon la séquence en zigzag suivante :

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

La lecture de la matrice dans cet ordre revient à classer les coefficients AC par ordre des fréquences spatiales croissantes. Chaque coefficient est ensuite décrit par une valeur composite RS codée sur 8 bits de la forme :

$$RS = (RRRRSSSS)b$$

RRRR indique la longueur de la plage de coefficients nuls entre 2 coefficients non-nuls (codage par plage). SSSS définit la catégorie du coefficient AC qui suit la plage de coefficients à 0 (même principe que pour le coefficient DC).

catégorie SSSS	coefficients AC
1	-1,1
2	-3,-2,2,3
3	-7..-4,4..7
4	-15..-8,8..15
5	-31..-16,16..31
6	-63..-32,32..63
7	-127..-64,64..127
8	-255..-128,128..255
9	-511..-256,256..511
10	-1023..-512,512..1023

De plus, on définit deux valeurs RS spécifiques. Si la longueur de la plage à 0 est supérieure à 15, alors on a RS = (11110000)b qui indique 15 coefficients nuls suivis d'un coefficient nul (ZRL). Si tous les coefficients sont nuls jusqu'au dernier, alors on a RS = (00000000)b qui indique la fin de bloc (EOB). Le tableau suivant synthétise les différentes valeurs de codage :

R \ S	0	1	2	3	4	5	6	7	8	9	10
0	EOB										
1	X										
2	X										
3	X										
4	X										
5	X										
6	X										
7	X										
8	X										
9	X										
10	X										
11	X										
12	X										
13	X										
14	X										
15	ZRL										

VALEURS COMPOSITES

A chaque valeur composite RS est associé un code de Huffman qui est suivi, comme pour le coefficient DC, de bits additionnels spécifiant le signe et l'amplitude exacte du coefficient. Prenons un exemple. Soit la matrice DCT quantifiée suivante :

	5	0	0	0	0	0	0
0	0	10	0	0	0	0	0
0	0	0	0	0	0	0	0
0	-35	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	-12	0	0	0	0

En balayant en zigzag cette matrice, on voit les séquences suivantes :

- 0 zéro suivi d'une valeur égale à 5,
- 5 zéros suivis d'une valeur égale à 10,
- 3 zéros suivis d'une valeur égale à –35,
- 37 zéros suivis d'une valeur égale à –12,
- 14 zéros.

d'où on déduit les valeurs de RS suivantes:

$$RS = (03)h\ ,\ (54)h\ ,\ (36)h\ ,\ (F0)h\ ,\ (F0)h\ ,\ (54)h\ ,\ (00)h$$

Comme pour le coefficient DC, on fait suivre le code de Huffman correspondant à la valeur de RS, par les SSSS bits de poids faibles de l'échantillon non nul que l'on code. Si le code de Huffman correspondant à $(03)_h$ vaut $(100)_b$, on envoie le code :

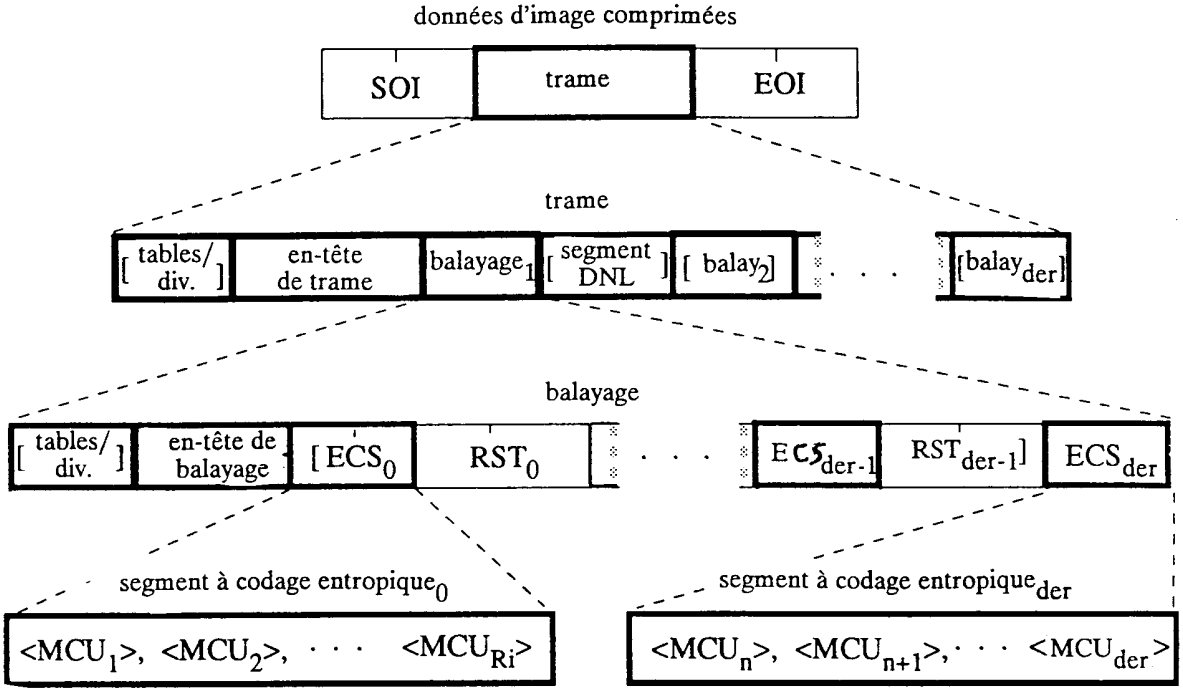
$$\text{mot-code} = (100\ \ 101)_b$$

4.2.2.3.2.4 Spécification des tables de Huffman

Les tables sont calculées et spécifiées dans JPEG comme nous l'avons vu dans le chapitre sur les compressions sans pertes de données.

4.2.2.4 Syntaxe du train binaire

La structure de la syntaxe du train binaire (en mode non-hiérarchique) est la suivante :



Elle est composée de plusieurs niveaux (trame, balayage et segment) précédés d'en-têtes et de tables de codage et/ou de tables de quantification. Nous n'allons pas étudier ici le détail de cette syntaxe, mais mettre l'accent sur deux problèmes importants causés tous les deux par l'utilisation de codes à longueurs variable :

1. La taille des images comprimées varie généralement du simple au triple en fonction de la richesse en détails fins de l'image. Le débit en sortie d'un codeur MJPEG ne peut pas être constant si on comprime un signal de télévision. Il est nécessaire de lui ajouter un mécanisme de régulation de débit agissant en temps réel sur les coefficients de quantification. On n'a donc que deux possibilités :

   - La qualité des images comprimées est constante et le débit varie.

   - Le débit est constant et la qualité des images varie.

2. Dans un code à longueur fixe, une erreur de transmission provoque une erreur de lecture de la donnée en cours, mais l'erreur ne se propage pas sur les données suivantes. Pour un code à longueur variable (VLC), il y a propagation des erreurs de transmission. En effet, le décodeur ne connaît la longueur d'un VLC qu'une fois qu'il a décodé sa valeur. L'inversion d'un bit du code se traduit par une erreur sur la valeur décodée, mais aussi sur la longueur du code. Le décodeur se désynchronise donc en cas d'erreur et se met à lire n'importe quoi. Il est donc impératif d'insérer de temps en temps dans le train binaire des codes de resynchronisation de longueur fixe qui soit uniques (et généralement alignés sur une frontière d'octets). Dans JPEG, ces marqueurs sont codés sur 16 bits comme par exemple le marqueur de début d'image SOI = FFD8.

4.3 La compression d'images animées

4.3.1 introduction

La compression des images animées concerne principalement deux domaines d'application :

1. la vidéoconférence. Elle permet de réaliser, par liaisons téléphoniques ou satellites, des réunions de travail de visu entre équipes d'une même société situées dans plusieurs villes ou pays. Elle a nécessité pendant de nombreuses années des équipements lourds et coûteux (avec une salle dédiée bien souvent). Aujourd'hui, une simple caméra, un microphone et un PC couplé à un accès de base au RNIS (Réseau Numérique à Intégration de Service) permettent de la réaliser. Cette solution permet de plus le partage de fichiers (dataconferencing).
2. la télévision (au sens large). Elle inclut la vidéo sur ordinateur (vidéo sur CDROM ou sur DVD, acquisition, restitution, montage ou effets spéciaux) et la diffusion de télévision numérique (par voie terrestre, par faisceau hertzien, par satellite, par câble ou via un réseau de type ATM par exemple).

4.3.1.1 La vidéoconférence

Le tableau suivant donne un résumé des différentes normes utilisées dans ce domaine. Il faut y ajouter des formats propriétaires que de grands constructeurs informatiques (Intel avec Indeo par exemple) essayent d'imposer par le biais du marché. La vidéoconférence peut être de type point à point (entre deux stations) ou multipoints.

norme générique	norme vidéo	norme audio	normes contrôle et système	application
H.320	H.261	G.711 (50-3600 Hz,48-64 Kbit/s) G.722 (50-6900 Hz,48-64 Kbit/s) G.728 (50-3600 Hz, 16 Kbit/s)	H.221 et H.230 et H.242	RNIS (2.B + D ou 6.B + D)
H.323	H.261 ou H.263	G.711, G.722, G.728, G.723	H.225 et H.245 RTP/RTCP	LAN (sans garantie de bande passante)
H.324	H.261 ou H.263	G.723 (50-3600 Hz, 5.3 & 6.3 Kbit/s)	H.245 et H.223	RTC via un modem V34 (28800 bps)
H.310	H.261 MPEG-2	MPEG-1, MPEG-2, G.7xx	H.221 et H.222 et H.245	ATM (haute résolution)
T.120	Les normes de la famille T.120 sont une couche applicative reposant sur H.310, H.323 et H.324.			partage de données informatiques

Les mécanismes utilisés dans H.261 sont les mêmes que dans MPEG-1, les deux algorithmes se ressemblant d'ailleurs beaucoup. Les niveaux de qualité sont les suivants :

- 30 images au format CIF par seconde (qualité de type VHS) avec un débit de 384 Kbit/s (6.B + D).
- 15 images au format CIF par seconde (qualité dégradée) avec un débit de 128 Kbit/s (2.B + D).

L'algorithme destiné à le remplacer, H.263, permet (au mieux) de doubler le taux de compression.

4.3.1.2 La télévision

Les algorithmes utilisés pour comprimer le signal vidéo existent depuis de nombreuses années, mais ils sont restés inutilisés tant que les circuits intégrés n'ont pas eu assez de puissance de calcul pour pouvoir les implémenter. Dès que ce fut le cas, le processus de normalisation a commencé. La normalisation MPEG (Motion Picture Expert Group) qui concerne les images animées et le son a débuté en même temps que JPEG (image fixe). Dans MPEG, on souhaite obtenir un fort taux de compression tout en préservant une bonne qualité d'image. D'autre part, l'enregistrement numérique des images comprimées nécessite un accès immédiat à chaque image ; c'est l'accès aléatoire. En comprimant individuellement chaque image avec un processus de type JPEG, on a un accès aléatoire puisque chaque image peut être décodée indépendamment des autres, mais la compression n'est pas suffisamment élevée.

Pour l'augmenter, on doit réduire la redondance temporelle grâce à la détection de mouvements. A partir d'une image de référence, on prédit la valeur de l'image à coder puis on soustrait cette prédiction à la valeur réelle de l'image afin d'obtenir l'erreur de prédiction. On code ensuite dans le train binaire :

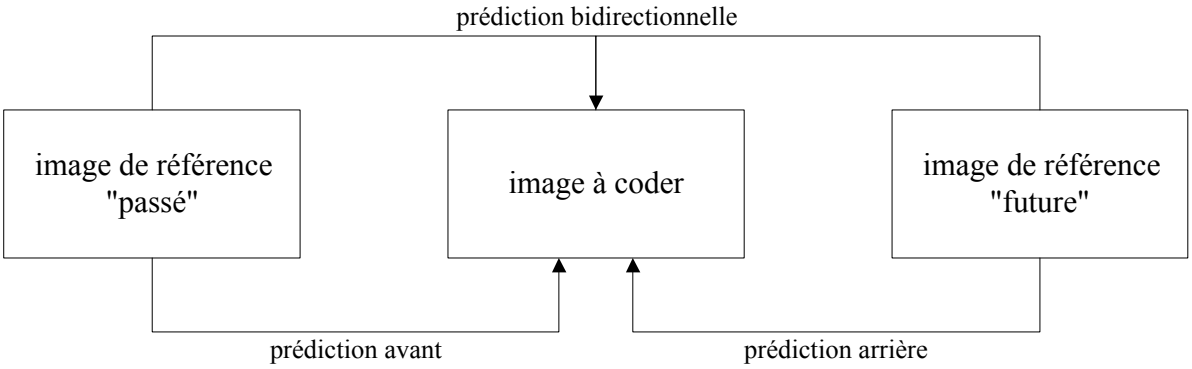
- les vecteurs de mouvement permettant de reconstruire la prédiction à partir de l'image de référence décodée.
- l'image représentant l'erreur de prédiction.

Au décodage, l'accès à cette image n'est pas indépendant car il faut avoir décodé au préalable la ou les images de références pour pouvoir la reconstruire, ce qui limite l'accès aléatoire.

Cependant, pour obtenir un taux de compression élevé, il faut avoir un minimum d'images indépendantes. La norme MPEG est un compromis entre accès aléatoire aux données et fort taux de compression.

Les techniques clés utilisées pour la compression sont les suivantes :

- Une détection et compensation de mouvements. On peut utiliser une détection avant, arrière ou bidirectionnelle.

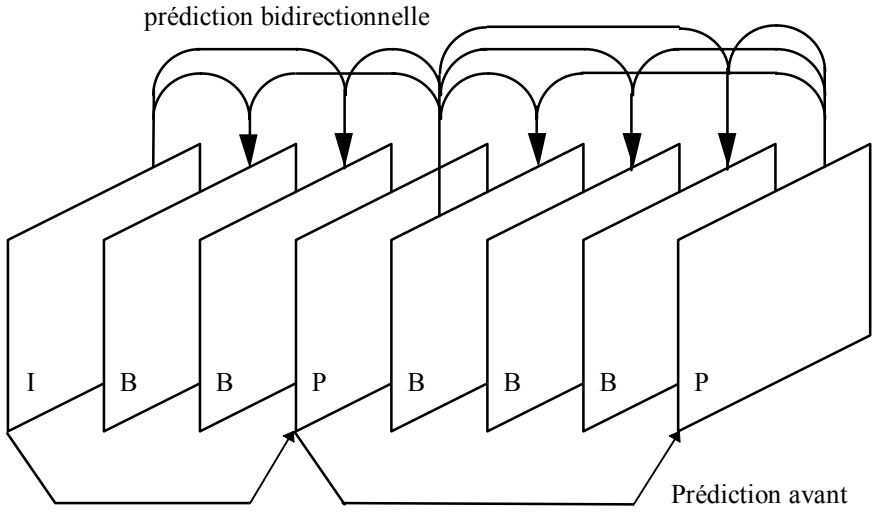


  - L'image à coder ne faisant l'objet d'aucune prédiction s'appelle image I (Intra). Elle sert de référence pour prédire les images B ou les images P. Son taux de compression est le plus faible car l'image fait seulement l'objet d'un codage de type JPEG.

  - L'image à coder faisant l'objet d'une prédiction avant s'appelle image P (Prédite). Elle sert de référence pour prédire des images B ou des images P. La compression est nettement plus importante car on ne code plus que l'erreur de prédiction qui est normalement moins riche en détails fins que l'image d'origine.

  - L'image à coder faisant l'objet d'une prédiction avant et arrière s'appelle image B (Bidirectionnelle). Elle ne peut pas servir de référence pour la prédiction. La compression est encore plus importante car l'erreur de prédiction est plus faible.

La figure suivante illustre les relations possibles entre ces trois types d'images à l'intérieur d'un groupe d'images. Un groupe débute obligatoirement par une image I qui est le point d'accès obligatoire pour pouvoir le décoder.



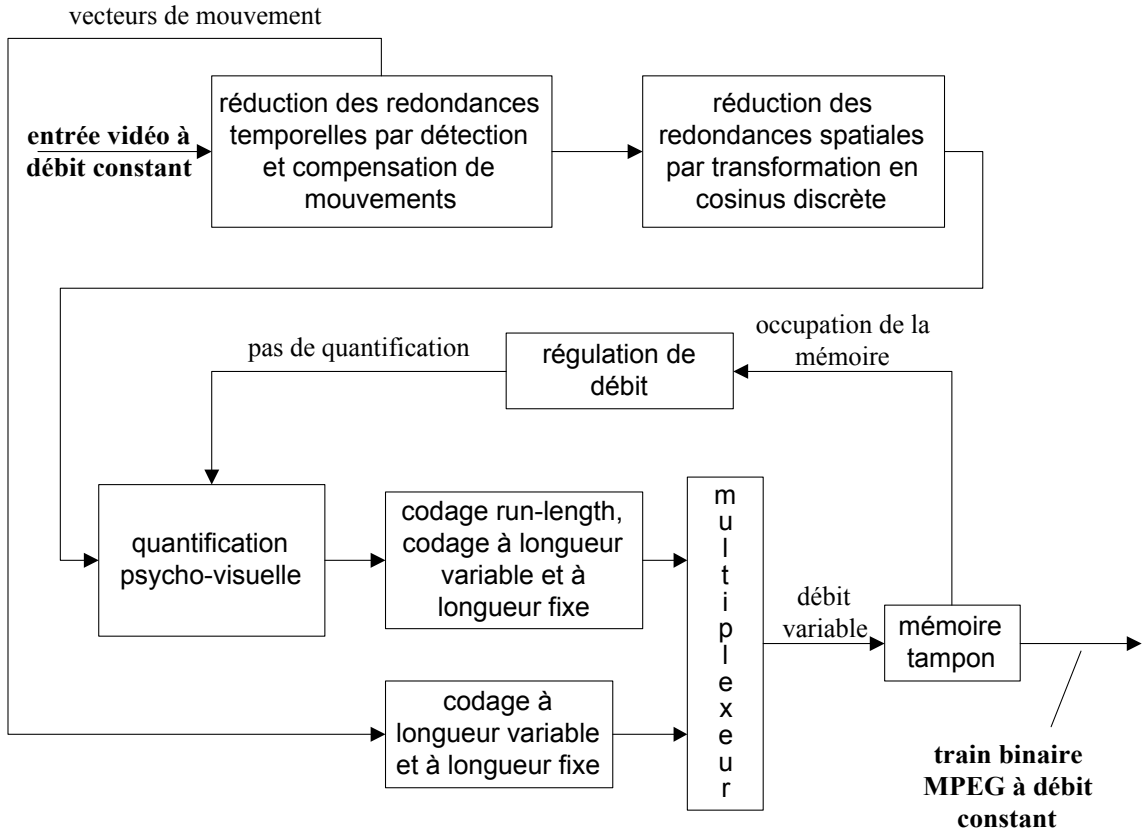
- Une compression de type JPEG qui associe :
  1. la transformation en cosinus discrète (TCD).
  2. la quantification psycho-visuelle. C'est à cette étape qu'à lieu la perte d'information car MPEG est un codage avec pertes. On ne récupère pas au décodage la totalité de l'information de départ, mais l'information perdue est supposée ne pas être utilisée par le système visuel humain. Aux erreurs d'arrondis près, les autres étapes du processus sont sans pertes.
  3. le codage « run-length » suivi d'un codage à longueur variable et/ou d'un codage à longueur fixe.

  Cette compression JPEG s'applique soit sur l'image à coder, soit sur l'erreur de prédiction.

- Un codage à longueur variable et un codage à longueur fixe qui est appliqué sur les vecteurs de mouvements.

- Une régulation de débit. L'utilisation des codes à longueurs variables dans le codage entraîne un débit variable en sortie du codeur. Dans la majorité des applications, il faut que le débit soit constant, aussi on doit ajouter une boucle de régulation de débit dans le codeur. Elle prend en compte le taux d'occupation du tampon de sortie et joue sur le pas de quantification pour réguler le débit. Sa réalisation détermine en partie la qualité des images comprimées.

Le schéma suivant résume le principe simplifié de la compression MPEG :



Au décodage, on reconstitue à l'aide des vecteurs de mouvement les images dites compensées en mouvement puis on leur ajoute l'erreur de prédiction contenue dans le train binaire. Il faut noter que la complexité du codeur est beaucoup plus élevée que la complexité du décodeur ce qui n'est pas gênant puisque c'est le coût du décodeur qui est le plus important. On parle alors de codage asymétrique (à la différence de JPEG ou le codage est symétrique).

Les normes MPEG sont au nombre de deux (nous ne parlerons pas ici de la norme MPEG-4) :

- La norme MPEG-1 a été finalisée en 1992. Son but était de permettre le stockage et la reproduction d'un film sur support CDROM simple vitesse avec un débit de l'ordre de 1,5 Mbit/s. Etant donné le taux de compression à atteindre (plus de 100 avec des images au format CCIR601 4 :2 :2), on réduit d'entrée la résolution de l'image en utilisant le format SIF ce qui donne une qualité d'image de type VHS. On alloue un débit constant de 1,15 Mbit/s à la vidéo, les 350 Kbit/s restant étant utilisés pour le son stéréo et des données auxiliaires. Il faut rappeler que le format SIF concerne des images non-entrelacées.

- La norme MPEG-2 a été finalisée en 1995. Elle concerne toutes les applications de la télévision (notamment la diffusion) avec des niveaux de qualité d'images (entrelacées ou non) allant du format SIF à la haute définition. Comme il y a compatibilité ascendante, cette norme englobe la norme MPEG-1. Certains diffuseur américains ayant commencé la commercialisation de services de télévision numérique par satellite (DirectTV notamment) avant l'élaboration de MPEG-2, une « norme » MPEG-1.5 a été créée (Philips et Thomson) pour prendre en compte l'entrelacement des images en télévision. MPEG-2 comporte 6 profils (profiles) qui déterminent le jeu d'outils de compression utilisé et 4 niveaux (levels) définissant la résolution des images. La signification des niveaux est la suivante :

nom	résolution	application
Low	352x240x30 352x288x25	format SIF (MPEG-1)
Main	720x480x30 720x576x25	format CCIR601 (qualité numérique studio)
High-1440	1440x1080x30 1440x1152x25	format TVHD en 4/3
High	1920x1080x30 1920x1152x25	format TVHD en 16/9

Les profils ont la signification suivante :

nom	utilisation
Simple	Il n'y a pas d'image B. La simplification du codeur et du décodeur est obtenue au détriment de la compression.
Main	C'est le profil standard utilisé en télévision. Il utilise des images de type I, B et P au format 4:2:0 et correspond au meilleur compromis qualité/compression/complexité.
4:2:2	Ce profil a été définit spécialement pour les applications studio. Les images sont au format 4:2:2 et les débits autorisés sont plus élevés.
SNR scalable	Il s'agit d'un profil de codage hiérarchique prévu pour une utilisation ultérieure. Il permet de transmettre une image de base en terme de quantification ainsi que des informations supplémentaires séparées permettant

	d'améliorer ses caractéristiques.
Spatially scalable	Même chose que pour le profil précédent, mais avec une image de base en terme de résolution spatiale. Ces deux profils permettent par exemple : • de transmettre simultanément en définition standard et HD. • de permettre une réception de qualité acceptable en cas de réception difficile et de qualité optimale dans de bonnes conditions (pour la diffusion terrestre).
High	Profil concernant la télévision haute définition (4:2:2 ou 4:2:0).

Les combinaisons de profils et de niveaux suivantes sont possibles avec les débits maximums suivants :

	Simple	Main	4 :2 :2	SNR scalable	Spatially scalable	High
Low		MP@LL 4 Mbit/s		SNRP@LL 4 Mbit/s		
Main	SP@ML 15 Mbit/s	MP@ML 15 Mbit/s	422P@ML 50 Mbit/s	SNRP@ML 15 Mbit/s		HP@ML 20 Mbit/s
High-1440		MP@H14L 60 Mbit/s			SSP@H14L 60 Mbit/s	HP@H14L 80 Mbit/s
High		MP@HL 80 Mbit/s				HP@HL 100 Mbit/s
image B		x	x	x	x	x
4 : 2 : 0	x	x	x	x	x	x
4 : 2 : 2			x			x
SNR				x	x	x
Spatially					x	x

Les normes MPEG se composent de trois parties distinctes :

1. La compression des images animées.

2. La compression du son. Il y quatre algorithme principaux : MPEG-1 couche I, MPEG-1 couche II, MPEG-1 couche III, MPEG-2 5.1 et 7.1 (et Dolby AC-3 qui quoique non normalisé MPEG tend à devenir une norme de fait au Etats-Unis pour le codage 5 voies).
3. La couche système. Elle assure le multiplexage des trains binaires audio et vidéo élémentaire afin de former un train binaire unique. Ce train est appelé « train programme (Program Stream) » quand il est destiné à un médium ne provoquant quasiment aucune erreur de stockage (disque dur, CDROM, DVD). Il est appelé « train transport (Transport Stream) » quand il est destiné à un médium provoquant beaucoup d'erreurs de transmission (diffusion par satellite par exemple).

Le tableau suivant résume les différentes appellations :

	Vidéo	Audio	System
MPEG-1	ISO/IEC 11172-2	ISO/IEC 11172-3 couche I, II et II uniquement	ISO/IEC 11172-1 train programme uniquement
MPEG-2	ISO/IEC 13818-2	ISO/IEC 13818-3	ISO/IEC 13818-1

Nous allons étudier maintenant la principale norme utilisée pour la télévision, MPEG-2 MP@ML.

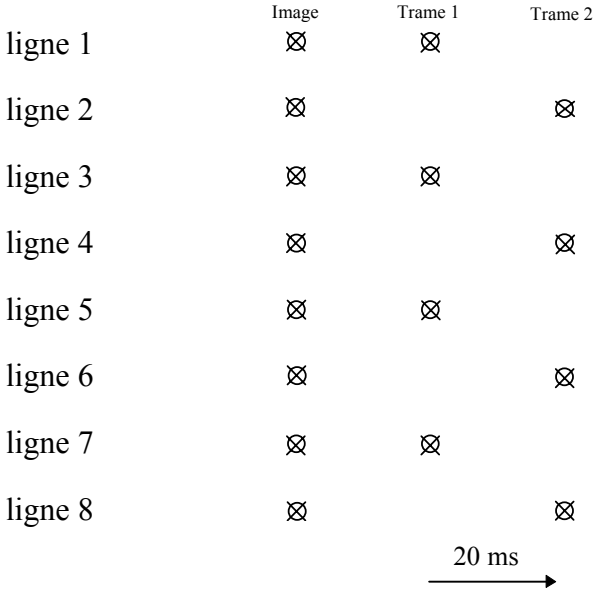
4.3.2 La norme MPEG-2 vidéo MP@ML

4.3.2.1 Codage

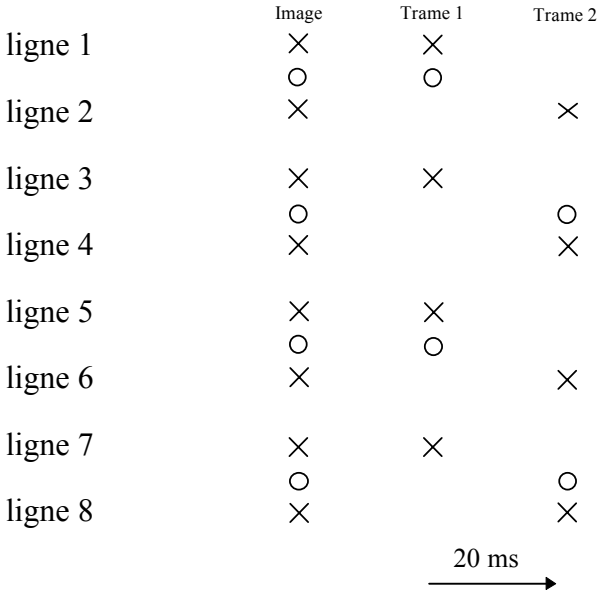
4.3.2.1.1 Pré-traitement

La source d'images utilisée en MP@ML est conforme à la recommandation 601 du CCIR (format 4:2:2). Sa structure d'échantillonnage est orthogonale à coïncidence. L'image est entrelacée, la trame 1 correspondant aux lignes impaires, la trame 2 correspondant aux lignes paires. Les deux trames sont espacées temporellement de 20 ms. Dans MPEG-2, on peut

travailler soit sur l'image 720x576 (les deux trames superposées), soit sur les deux trames 720x288 prises séparément.



La conversion du format 4:2:2 au format 4:2:0 est obligatoire. Dans le premier format, on a un échantillon de chrominance pour deux échantillons de luminance. Dans le format 4:2:0, on a un échantillon de chrominance pour quatre échantillons de luminance, l'échantillon de chrominance ne coïncidant plus avec un échantillon de luminance. La position de l'échantillon de chrominance dans chaque trame est la suivante :



La compression préserve cette qualité d'image avec un débit d'environ 6 Mbit/s. Avec un débit plus faible, par exemple 3,5 Mbit/s, on procède généralement à un filtrage avec sous-échantillonnage de l'image pour travailler avec une définition réduite de 544x576 pixels, ce qui correspond à une qualité de type PAL.

4.3.2.1.2 Séquence vidéo et groupe d'images

Après pré-traitement, on va découper la suite d'images 4:2:0 en séquences, puis en groupe d'images (GOP). Une séquence contient un nombre entier de GOP. Elle commence par un code de début (start_code) et un en-tête et se termine par un code de fin. La durée d'une séquence peut être quelconque. Un groupe d'images (GOP) est composé d'images I, P et B. Sa composition est complètement paramétrable à condition qu'il commence par une image I. On définit, pour le qualifier, deux paramètres qui sont la distance M entre deux images Prédites et la distance N entre deux images Intra. La composition du GOP n'est pas normalisée, toutefois la structure suivante est généralement utilisée pour la télévision au format européen (50 trames/s).

$$M = 3, N = 13 \Rightarrow I\ B\ B\ P\ B\ B\ P\ B\ B\ P\ B\ B\ P$$

4.3.2.1.3 Mise en ordre des images

On a vu au paragraphe précédent une structure de groupe typique. A l'entrée du codeur, on a donc :

GOP n°1													GOP n°2					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
I	B	B	P	B	B	P	B	B	P	B	B	P	**I**	B	B	P	B	...

Les images 1 et 14 sont codées indépendamment en Intra, les images 4, 7, 10, 13 et 17 sont prédites à partir des images 1 ou 14 ou des images prédites passées, alors que les images 2, 3, 5, 6, 8, 9, 11, 12, 15, 16 et 18 sont prédites à partir d'images de références antérieures et postérieures. On s'aperçoit que les images B, pour pouvoir être décodées, doivent obligatoirement être précédées des images I et P qui leur servent de référence. Ainsi, pour permettre le décodage, l'ordre des images dans le train binaire est modifié par rapport à leur séquence naturelle. A la sortie du codeur, dans le train binaire MPEG-2 et à l'entrée du décodeur, on a :

GOP n°1													GOP n°2					
1	4	2	3	7	5	6	10	8	9	13	11	12	14	17	15	16	20	...
I	P	B	B	P	B	B	P	B	B	P	B	B	**I**	P	B	B	P	...

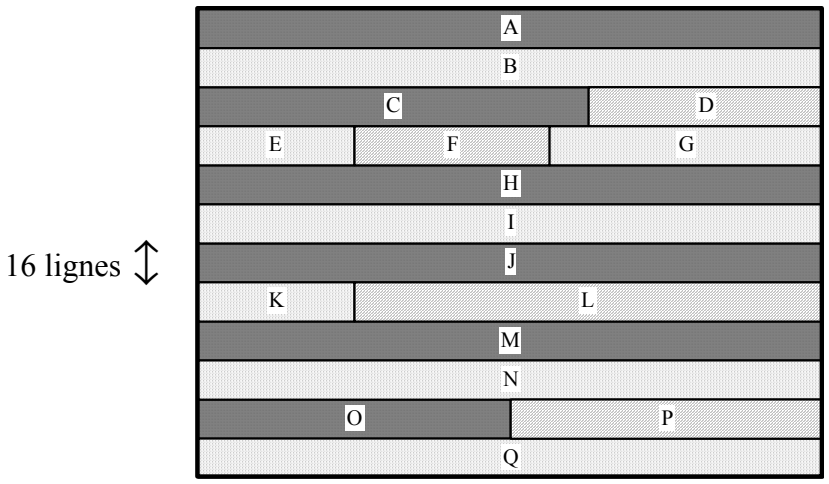
On doit donc remettre en ordre les images au décodage afin d'obtenir à sa sortie :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...

L'inconvénient immédiat qui découle d'un tel système de codage est qu'il ne permet pas de faire du montage (application magnétoscope) car un grand nombre d'images dépend les unes des autres. On ne peut pas accéder indépendamment à une image B ou P. Le point d'accès dans le GOP est la première image du groupe codé ; l'image Intra.

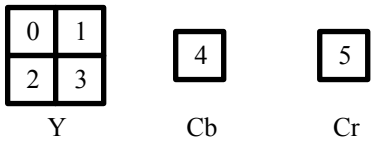
4.3.2.1.4 Slices

Un slice est une suite d'un nombre quelconque de macroblocs (un carré de 16 lignes par 16 pixels). Il doit contenir au moins un macrobloc et ne peut pas déborder sur un autre slice. Le premier et le dernier macrobloc dans un slice doivent se trouver sur la même ligne horizontale de macroblocs. Dans la structure de slice générale, les slices ne sont pas obligés de recouvrir entièrement l'image. Toutefois, en MP@ML, il existe une structure restreinte qui doit être utilisée.



4.3.2.1.5 Macroblocs

Au format 4:2:0, un macrobloc de 16x16 pixels est composé de quatre blocs 8x8 d'échantillons de luminance, et de deux blocs contenant les échantillons de chrominance Cr et Cb correspondant.

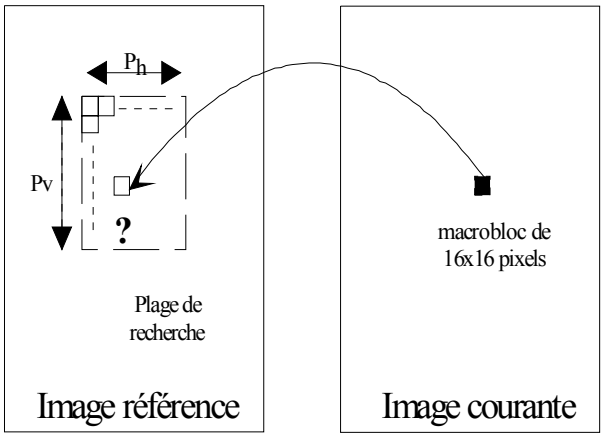


Un macrobloc sauté est un macrobloc pour lequel aucune information n'est codée dans le train binaire.

4.3.2.1.6 Détection et compensation de mouvements

La détection de mouvement a lieu sur le macrobloc. Elle consiste, connaissant un macrobloc de luminance courant, à trouver le macrobloc de luminance qui lui ressemble le plus dans une image de référence. C'est le macrobloc de référence. Connaissant la position des deux macroblocs, on en déduit un vecteur de déplacement. Les critères de ressemblance entre deux macroblocs sont généralement l'erreur quadratique moyenne et l'erreur absolue moyenne.

La recherche du macrobloc de référence se fait à l'intérieur d'une fenêtre dont les dimensions sont fonction des valeurs de deux paramètres, f_code horizontal et f_code vertical. Cette recherche peut se faire en mode image ou en mode trame. En mode image, on a un macrobloc courant de dimension 16x16 et on recherche le meilleur macrobloc dans l'image de référence. On obtient un vecteur ayant deux coordonnées x et y. En mode trame, on a deux macroblocs courants de dimension 16x8 correspondant à chaque trame et on recherche séparément le meilleur macrobloc 16x8 dans chaque trame de l'image de référence. On calcule donc un vecteur pour chaque trame.



Fcode	Ph	Pv	
		(mode image)	(mode trame)
1	16	16	8
2	32	32	16
3	64	64	32
4	128	128	64
5	256	256	128

Pv : plage de recherche verticale
Ph : plage de recherche horizontale

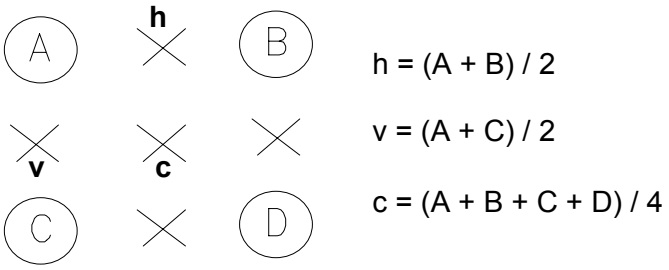
Les dimensions de la fenêtre sont choisies en fonction du déplacement des motifs entre l'image courante et l'image de référence. Il existe, pour chaque séquence vidéo, une taille de fenêtre optimale qu'il n'est pas nécessaire de dépasser. Pour une séquence d'images dont les mouvements sont lents, un f_code 3-2 (horizontal-vertical) est suffisant. Pour une séquence rapide, il est préférable de choisir un f_code 5-4.

L'algorithme estimant le déplacement d'un macrobloc n'est pas normalisé. La méthode FULLSEARCH (recherche intégrale) donne actuellement les meilleurs résultats avec une charge de calcul très importante.

La compensation de mouvement est la phase qui consiste à prendre le macrobloc de référence et à le déplacer de la valeur du vecteur de mouvement correspondant. Par exemple, au décodage, pour obtenir le macrobloc décodé, on compense le macrobloc de référence puis on lui ajoute l'erreur de prédiction.

Dans l'algorithme FULLSEARCH (recherche intégrale), on cherche systématiquement le meilleur macrobloc sur toute la fenêtre. Plusieurs critères de concordance des macroblocs sont utilisables. L'erreur quadratique moyenne donne les meilleurs résultats, mais on préfère utiliser le critère de la différence absolue qui est plus rapide à calculer. Si deux macroblocs, ou plus, dans l'image de référence, donnent la même erreur, on garde le macrobloc qui présente la distance minimale avec le macrobloc courant.

L'estimation du macrobloc trouvé par l'algorithme de recherche (FULLSEARCH ou autre) doit être affinée par une recherche au demi-pixel. Cela revient à déplacer le macrobloc d'un demi-pixel dans les huit directions possibles. On utilise l'interpolation linéaire avec la convention de l'arrondi à l'entier le plus proche.



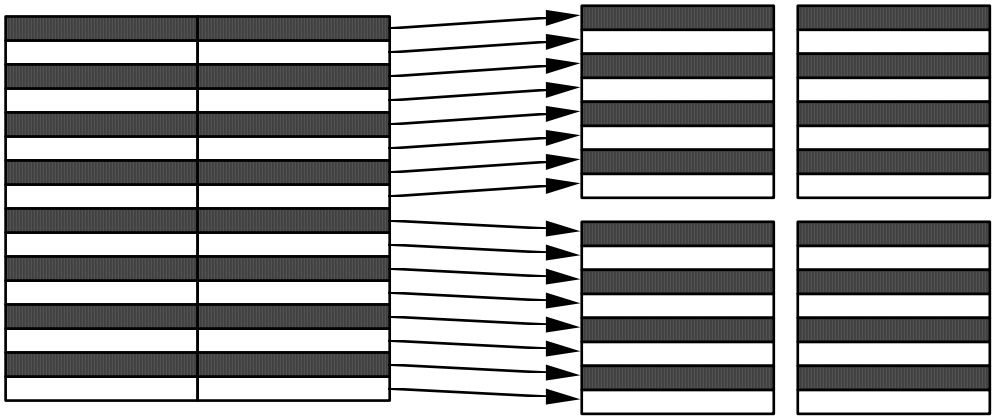
Des critères de choix doivent être définis pour sélectionner le type de recherche utilisé pour compenser un macrobloc. On a les possibilités suivantes :

- pour un macrobloc de type P, on peut faire une détection avant (image de référence passée) en mode image ou en mode trame.

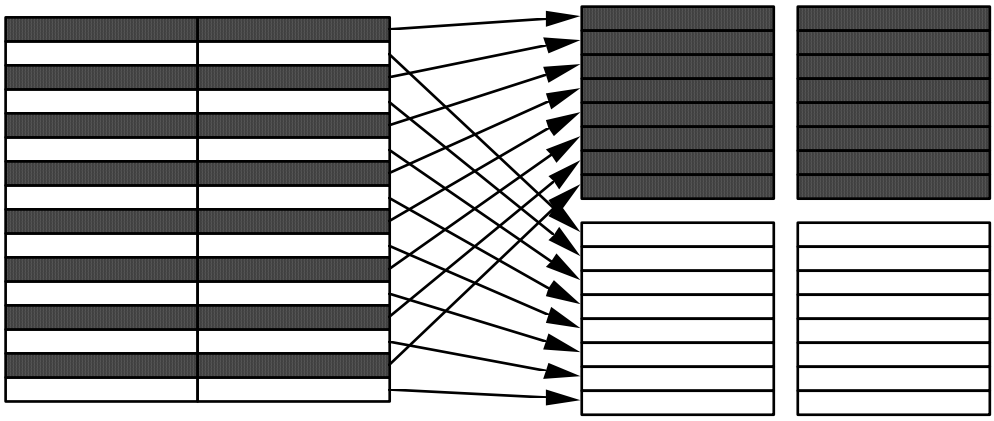
- pour un macrobloc de type B, on peut faire une détection avant (image de référence passée), une détection arrière (image de référence à venir) ou bien une détection bidirectionnelle. Dans ce cas, le macrobloc de référence est calculé en faisant la moyenne des deux macroblocs obtenus en détection avant et arrière. Ces trois estimations peuvent se faire en mode image ou en mode trame.

4.3.2.1.7 Transformation en cosinus discrète

Cette transformation (TCD) permet de réduire la redondance spatiale de l'image. Elle est appliquée soit sur des blocs extraits d'un macrobloc Intra, soit sur des blocs extraits d'un macrobloc représentant l'erreur de prédiction (macrobloc Inter). Les quatre blocs de luminance peuvent être traités en mode trame (chaque trame prise séparément) ou en mode image (les deux trames superposées). Les deux blocs de chrominance ne peuvent être traités qu'en mode image. La figure suivante illustre le découpage du macrobloc de luminance en mode TCD image.



La figure suivante illustre le découpage du macrobloc de luminance en mode TCD trame.



La Transformation en Cosinus Discrète directe bi-dimensionnelle NxN est définie par :

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\frac{(2x+1)u\pi}{2N} \cos\frac{(2y+1)v\pi}{2N}$$

avec u, v, x, y = 0, 1, 2, ... N-1

où x, y sont des coordonnées spatiales.

u, v sont des coordonnées dans le plan transformé.

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u,v = 0 \\ 1 & \text{autrement} \end{cases}$$

Les coefficients spatiaux f(x,y) sont codés sur 9 bits avec une dynamique possible de [-256 ; +255]. Les coefficients transformés F(u,v) sont codés sur 12 bits avec une dynamique de [-2048 ; +2047].

4.3.2.1.8 Quantification psycho-visuelle

Après TCD, on va diviser la valeur de chacun des 64 coefficients d'un bloc de luminance ou de chrominance par l'élément correspondant de la matrice de quantification. La matrice par défaut utilisée pour quantifier les blocs codés en Intra (luminance et chrominance) est :

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

La matrice par défaut utilisée pour quantifier les blocs de luminance et de chrominance non-codés en Intra, c'est-à-dire les erreurs de prédiction, est :

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

En simplifiant, la formule permettant de calculer le coefficient quantifié de cordonnées i, j QAC[i][j] vaut :

$$QAC[i][j] = \frac{16 * AC[i][j]}{Quant * W[i][j]}$$

avec : W = matrice de quantification,
AC = coefficient TCD,
Quant = pas de quantification variant de 1 à 31

4.3.2.1.9 Codages des images

La quantification a fait apparaître un grand nombre de zéros dans le bloc 8x8. Afin d'optimiser le codage, on va changer l'ordre des coefficients en effectuant un balayage en zigzag. Dans le tableau suivant, on a numéroté les 64 coefficients de 0 à 63. La figure de gauche représente l'ordre normal dans un bloc, la figure de droite représente l'ordre après balayage.

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

Ordre normal

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

Ordre après balayage en zigzag

En lisant les coefficients dans leur nouvel ordre, on va appliquer un codage par plage (« run-length ») en codant le nombre de zéros qui précède une valeur non-nulle et la valeur de ce coefficient. Le code correspondant à la valeur [nombre de zéros, valeur du coefficient] est un code de Huffman suivi d'un code de longueur fixe. Le code correspondant à la valeur [zéros jusqu'à la fin du bloc] est un code de Huffman nommé « end of block ». On appelle coefficient DC le coefficient 0 d'un bloc appartenant à un macrobloc Intra. Il correspond à la

valeur moyenne des coefficients de ce bloc. Il fait l'objet d'un codage particulier dans MPEG :

1. On code la différence entre ce coefficient DC et celui du bloc précédent de même type (luminance ou chrominance).
2. Le code correspondant à cette différence est un code de Huffman suivi d'un code de longueur fixe.

4.3.2.1.10 Codage des vecteurs de mouvements

Les vecteurs de mouvement sont utilisés par le décodeur pour reconstituer le macrobloc de référence. Ils ne sont pas codés directement dans le train binaire, mais c'est la différence entre ce vecteur et un vecteur de prédiction que l'on code. Le codage suit les règles suivantes :

- La prédiction est remise à zéro en début de slice ou si le dernier macrobloc a été codé en Intra.
- Chaque vecteur avant ou arrière est codé relativement au prédicteur de même type. Chaque coordonnée est codée indépendamment.
- Le vecteur, après codage, devient prédicteur pour le vecteur suivant.

Le code correspondant à une cordonnée du vecteur de différence ainsi obtenue est composé :

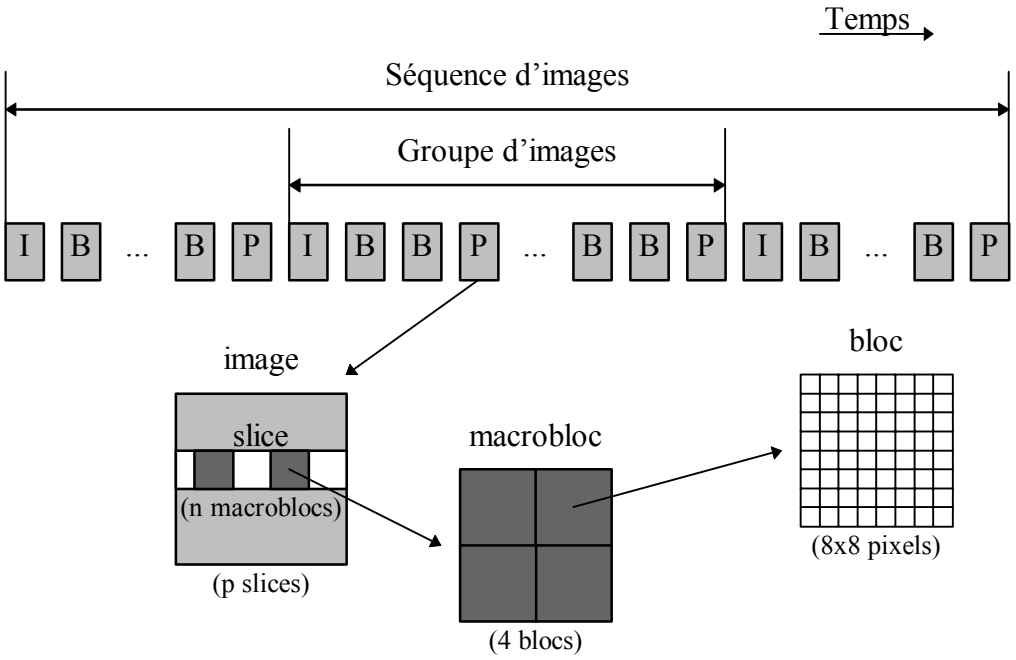
- d'un code à longueur variable allant de -16 à +16
- d'un code de longueur fixe de longueur égale à f_code - 1.

4.3.2.2 Syntaxe

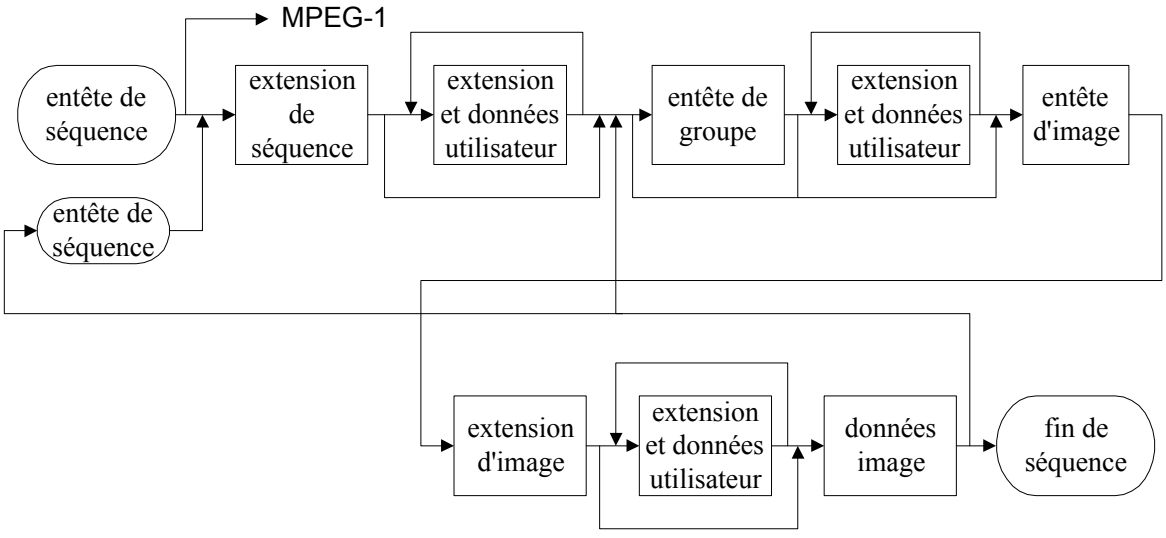
4.3.2.2.1 Organisation générale

Le train binaire vidéo peut être vu comme une structure hiérarchique comprenant six niveaux :

1. La séquence vidéo
2. Le groupe d'image
3. L'image
4. Le slice
5. Le macrobloc
6. Le bloc



Les trois premiers niveaux sont composés d'en-têtes et d'extensions qui contiennent les informations générales nécessaires au décodage du train binaire. La figure suivante nous montre l'organisation de ces trois niveaux. Les trois niveaux suivants contiennent les données images.



Plusieurs règles régissent la syntaxe des trois premiers niveaux :

- si le premier en-tête de séquence n'est pas suivi par une extension de séquence, alors il s'agit d'un train binaire MPEG-1.

- si le premier en-tête de séquence est suivi d'une extension de séquence, alors tous les en-têtes de séquence sont suivis par une extension de séquence.

- s'il y a une extension de séquence dans le train binaire, alors chaque en-tête d'image doit être suivi d'une extension d'image.

- la première image codée qui suit un en-tête de groupe doit être une image Intra.

4.3.2.2.2 Les start_code

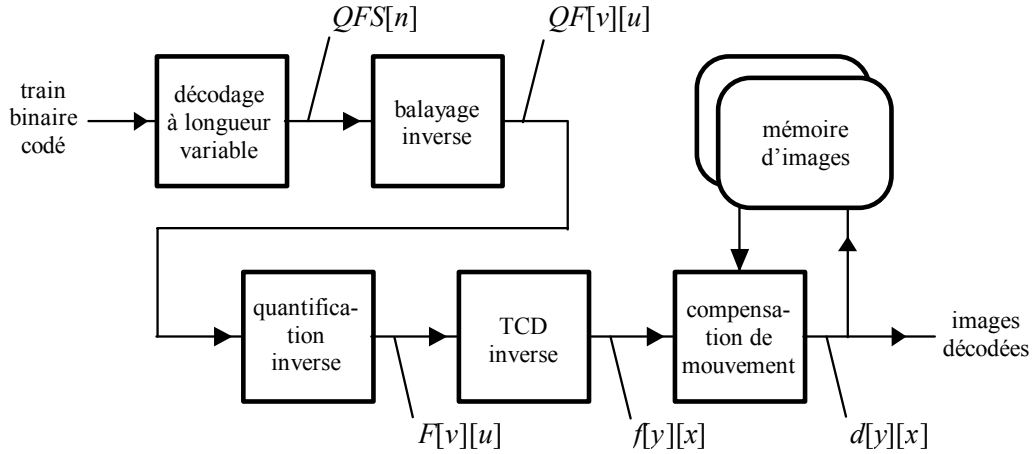
Les start_code sont des chaînes de 32 bits uniques dans le train binaire. Ils se composent d'un préfixe de 23 bits « 0000 0000 0000 0000 0000 0001 » suivi par une valeur d'un octet qui identifie le type du start_code. Cette valeur est toujours la même sauf dans le cas du start_code de début de slice où elle représente sa position verticale. Tous les start_code doivent être alignés sur un début d'octet : on insère au besoin des bits à 0 avant le début du préfixe de telle manière que le premier bit de la chaîne soit le bit de poids fort d'un octet. Le tableau suivant définit les valeurs utilisées pour coder le type de start_code dans le train binaire vidéo.

nom	valeur (hexadecimal)
picture_start_code	00
slice_start_code	01 à AF
reserved	B0
reserved	B1
user_data_start_code	B2
sequence_header_code	B3
sequence_error_code	B4
extension_start_code	B5
reserved	B6
sequence_end_code	B7
group_start_code	B8

4.3.2.3 Décodage

4.3.2.3.1 Organisation générale

On voit, figure suivante, le synoptique simplifié du décodeur Main Profile, Main Level. Les images une fois décodées sont séparées en deux pour former les trames qui sont affichées.



4.3.2.3.2 Décodage à longueur variable

Les codes à longueur variable sont décodés grâce au jeu de tables spécifié par la norme. On code dans le train binaire la différence entre deux coefficients DC successifs de même type (Y, Cr, Cb). La différence obtenue au décodage doit donc être ajoutée à un prédicteur pour retrouver la valeur d'origine du coefficient. Le décodeur utilise un prédicteur par composante

Y, Cr et Cb. A chaque fois qu'un coefficient DC est reconstruit, on assigne sa valeur au prédicteur correspondant. A certains moments, en synchronisme avec le codeur, on initialise les trois prédicteurs avec une valeur particulière.

Les autres coefficients d'un bloc sont composés d'au moins un code à longueur variable. Suivant sa valeur, on a trois possibilités :

- C'est un code « End of Block ». Il indique qu'il n'y a plus que des coefficients nuls dans le bloc courant. On passe au bloc suivant.

- C'est un code normal. Il est donc suivi, en général, d'un code à longueur fixe. On obtient une valeur [run, level] qui indique un nombre de zéros suivi de la valeur non-nulle d'un coefficient.

- C'est un code « Escape ». La valeur n'existe pas dans la table de codes à longueur variable, elle est donc codée par un code ayant une longueur fixe de 18 bits.

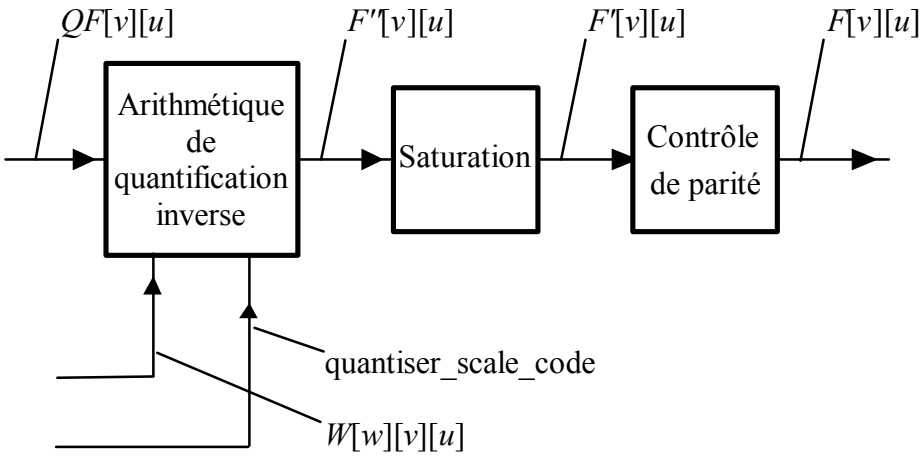
A la fin de cette étape, on connaît QFS[n], n étant compris entre 0 et 63.

4.3.2.3.3 Balayage inverse

On va maintenant convertir QFS[n] en un tableau à deux dimensions QF[v][u], u et v étant compris entre 0 et 7, par le biais d'un balayage inverse. Deux types de balayage existent, le balayage en zigzag et le balayage alterné.

4.3.2.3.4 Quantification inverse

Le tableau à deux dimensions QF[v][u] subit une quantification inverse pour produire les coefficients TCD reconstruits F[v][u]. Ce processus nécessite une matrice de quantification W[w][v][u] et un pas de quantification compris entre 1 et 31. La variable w vaut 0 ou 1 suivant le type du bloc, luminance ou chrominance. Après déquantification, on sature les coefficients F''[v,u] pour qu'ils appartiennent à l'intervalle [-2048 ; +2047] puis on applique un contrôle de parité.



4.3.2.3.5 TCD inverse

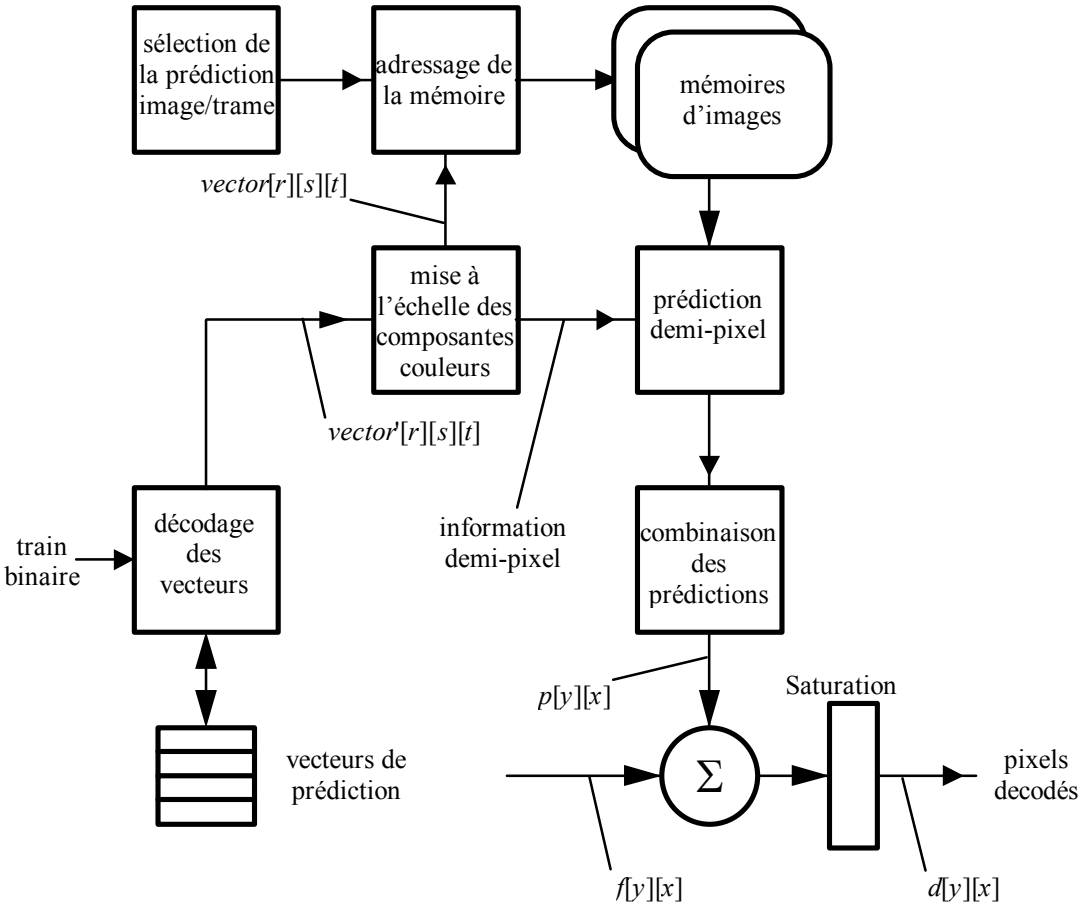
On applique sur les F[v][u] une TCD inverse :

$$f(x,y) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$

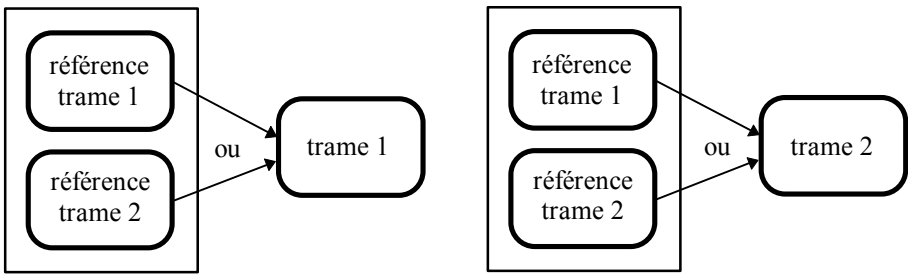
puis on sature le résultat pour que tous les f(x,y) soient compris dans l'intervalle [-256 ; +255]. Nous avons maintenant reconstruit un macrobloc Intra ou une erreur de prédiction. Il nous reste à étudier la compensation de mouvement.

4.3.2.3.6 Compensation de mouvement

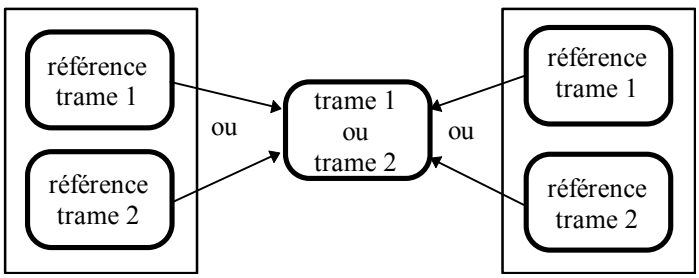
La compensation de mouvement forme des macroblocs de prédictions p[x][y] à partir d'images précédemment décodées. Elles sont combinées aux coefficients décodés f[x][y] issus de la TCD inverse pour obtenir les images décodées. Si le macrobloc est codé en Intra, il n'y a pas de prédiction. Si le macrobloc est sauté, les coefficients décodés f[x][y] sont nuls et le macrobloc final d[x][y] est seulement composé de la prédiction p[x][y].



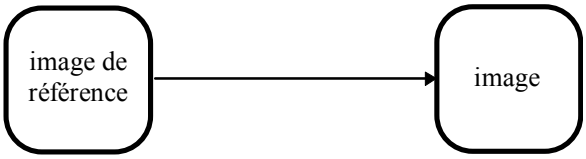
Il existe deux modes de prédiction : le mode trame et le mode image. Les figures suivantes résument les différentes possibilités : prédiction trame en mode avant,



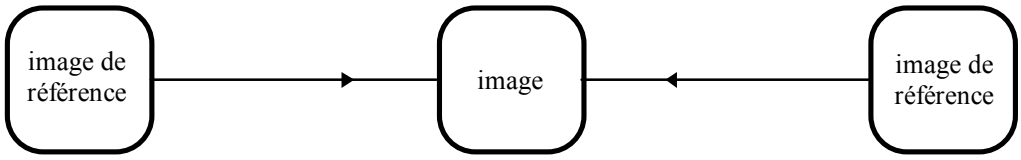
prédiction trame en mode bidirectionnel,



prédiction image en mode avant,



et prédiction image en mode bidirectionnel.



Pour réduire le nombre de bits utilisés, les vecteurs de mouvement sont codés différentiellement par rapport au vecteur précédent de même type. Ainsi, pour pouvoir décoder un vecteur, il existe quatre prédicteurs (ayant chacun une composante horizontale et verticale). Pour chaque macrobloc, on calcule un vecteur pour la luminance, puis on en déduit le vecteur, pour la chrominance. Afin que le décodage soit possible, les coordonnées verticales des vecteurs en mode trame doivent être limitées à la moitié de l'excursion possible prévue par le f_code.

Nous avons maintenant décodé un vecteur. Toutefois, l'algorithme n'a pas forcément modifié les quatre prédicteurs. Une mise à jour supplémentaire peut être nécessaire. Tous les prédicteurs doivent être remis à zéro dans les cas suivants :

- en début de slice.
- quand un macrobloc Intra est décodé.
- quand un macrobloc de type P de mouvement nul est décodé.
- quand un macrobloc est sauté dans une image P.

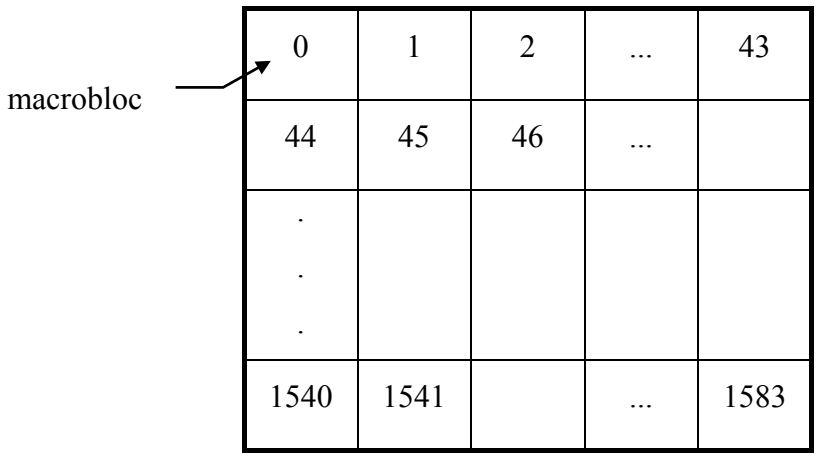
Nous avons fini d'étudier le processus de décodage et de calcul des vecteurs de mouvement. Il nous reste à former la prédiction à partir des pixels de la trame ou de l'image de référence. Un pixel donné est prédit en lisant le pixel de référence correspondant décalé de la valeur du vecteur de mouvement. Les règles suivantes doivent être appliquées :

- Tous les vecteurs sont spécifiés au demi-pixel. Ainsi, si une composante de vecteur est impaire, le pixel de référence doit être reconstitué par interpolation linéaire à partir des pixels voisins.
- En mode trame, une variable signale si la prédiction est formée à partir de la trame de référence 1 ou 2.
- En mode bidirectionnel, le pixel de référence est calculé en faisant la moyenne des pixels de référence avant et arrière.

Une fois formés, les blocs de prédiction p[x][y] sont ajoutés aux coefficients transformés f[x][y]. Le résultat est ensuite saturé pour donner les blocs décodés d[x][y] qui représentent les pixels de l'image décodée par le décodeur MPEG-2.

4.3.2.3.7 Calcul de l'adresse d'un macrobloc, macrobloc sauté

Pour pouvoir reconstruire l'image complète, nous devons calculer l'adresse du macrobloc décodé qui représente la position absolue du macrobloc que l'on veut décoder. Elle prend les valeurs suivantes :



Si la compensation de mouvement est suffisamment efficace, il n'y a pas de données transformées F(u,v) à coder. Les macroblocs se trouvant dans cette situation font l'objet d'un codage spécial et peuvent être sautés. Les règles suivantes sont obligatoires :

- Il n'y a pas de macroblocs sautés dans une image Intra.
- Le premier et le dernier macrobloc d'un slice ne sont pas sautés.
- Dans une image B, il n'y a pas de macrobloc sauté après un macrobloc Intra.

Pour reconstruire ces macroblocs, il n'y a alors ni coefficients TCD, ni vecteurs de mouvement à décoder. Le décodeur doit alors utiliser les règles suivantes :

- dans une image P
  1. La prédiction doit être faite en mode image.
  2. les prédicteurs doivent être remis à zéro.
  3. Le vecteur de mouvement utilisé vaut 0.
- dans une image B
  1. La prédiction doit être faite en mode image.
  2. La direction de la prédiction avant/arrière/bidirectionnelle doit être la même que celle du macrobloc précédent.
  3. Les prédicteurs ne sont pas affectés.
  4. Les vecteurs de mouvement utilisés sont égaux aux prédicteurs correspondants.

Nous sommes maintenant en mesure de décoder entièrement un train binaire Main profile, Main level.

5 Aspects système

5.1 Multiplexage des flux élémentaires

5.1.1 Généralités

Nous sommes maintenant en possession de trains binaires élémentaires à débit constant représentant :

- soit de la vidéo comprimée,
- soit de l'audio comprimée,
- soit des données (guide de programme, télétexte, contrôle d'accès, données privées, ...).

Le but de la couche système est de former un train binaire unique (train binaire série) à partir des affluents élémentaires. Cette couche doit assurer les fonctions suivantes :

1. synchroniser l'horloge du décodeur sur l'horloge du codeur,
2. synchroniser l'image et les sons qui lui sont associés,
3. empaqueter les affluents élémentaires et les identifier,
4. ajouter aux affluents élémentaires les informations de service nécessaires au bon fonctionnement du décodeur (tables systèmes, gestion du contrôle d'accès, ...),
5. initialiser et gérer les mémoires tampons (buffers) nécessaires au bon fonctionnement du codeur et du décodeur.

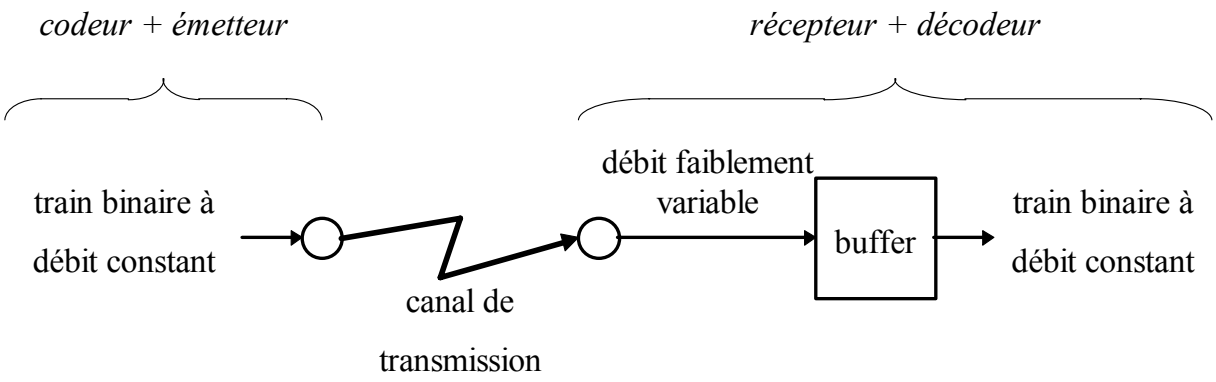
La couche système MPEG-2 permet de produire deux types de train système :

- Le train programme est destiné aux média quasiment sans erreurs de transmission tels que le CD-ROM, le DVD ou un réseau de transmission informatique.
- Le train transport est destiné aux média sujets aux erreurs de transmission comme la diffusion par satellite, par câble ou par voie terrestre.

La spécification système originelle définie par le groupe de normalisation international MPEG-2 a été complétée par le groupe européen DVB (Digital Vidéo Broadcast). Ce groupe d'industriels et de diffuseurs est habilité à proposer des normes « clés en main » à l'organisme européen compétent pour les télécommunications, l'ETSI (« European Telecommunication Standard Institute »). Nous allons commencer par étudier deux problèmes importants que doit résoudre le train système.

5.1.2 Le rôle de la mémoire tampon

Dans tout système de transmission numérique, il existe un buffer à la réception pour tenir compte du jitter introduit par le canal de transmission.

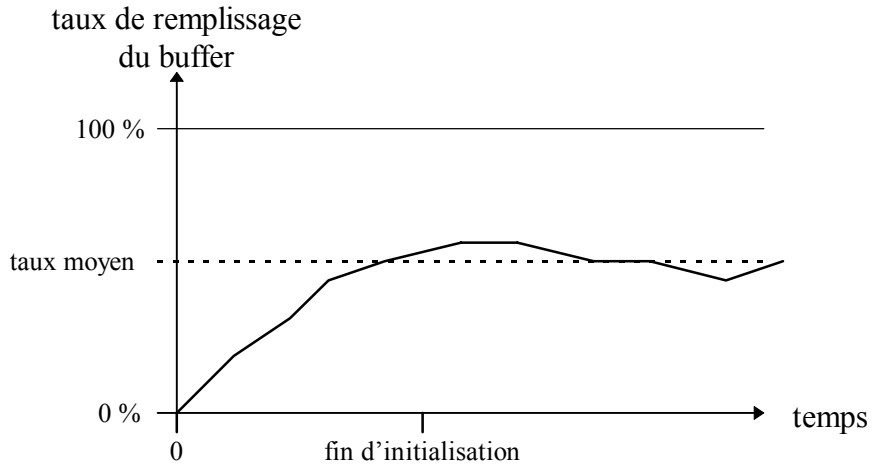


Ce buffer est alimenté par un train binaire à débit faiblement variable et vidé à débit constant. Ce débit doit être impérativement strictement égal au débit généré à l'émission sinon on atteindra les limites du buffer :

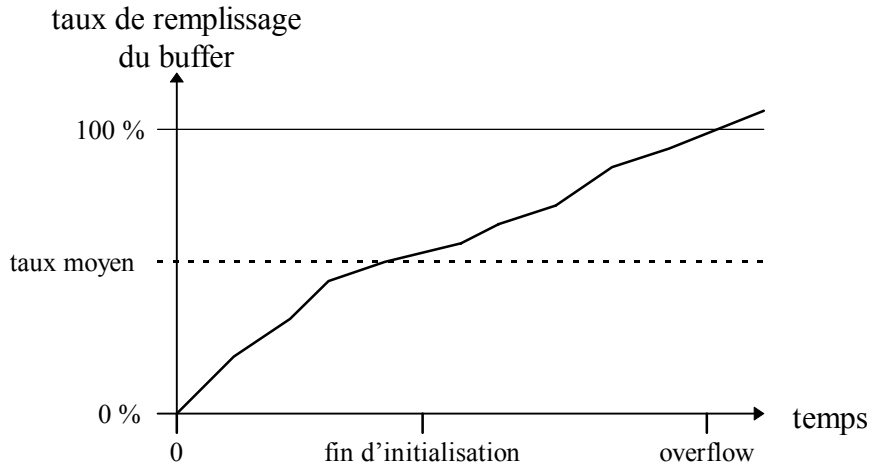
- Si le débit de vidage du buffer à la réception est inférieur au débit à l'émission, le buffer va se remplir jusqu'à sa taille maximum et provoquer une erreur de débordement « overflow » ce qui provoquera une perte de données.

- Si le débit de vidage du buffer à la réception est supérieur au débit à l'émission, le buffer va se vider jusqu'à sa taille minimum et provoquer une erreur de débordement « underflow » ce qui provoquera aussi une perte de données.

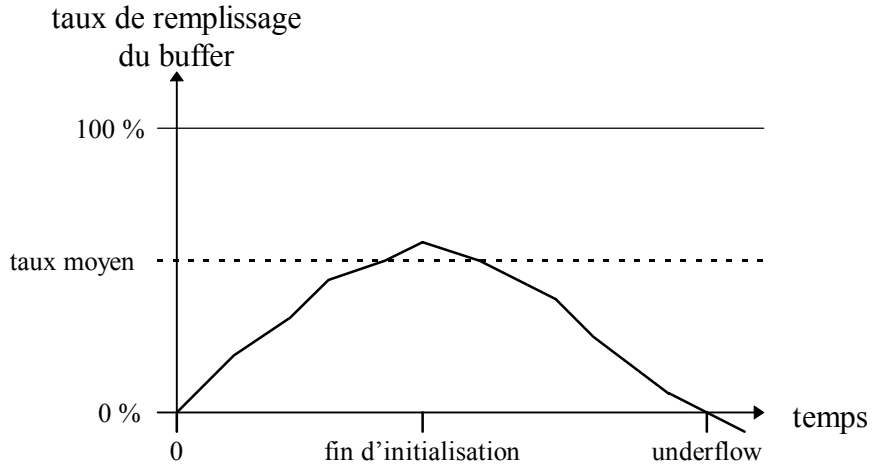
Il faut d'ailleurs considérer le début de la transmission. On ne peut pas commencer à vider le buffer lorsque le premier bit du message est reçu car le jitter de transmission pourrait provoquer accidentellement un underflow. Il faut un délai d'initialisation pour remplir par exemple la moitié du buffer puis commencer le vidage. Les trois figures suivantes représentent l'état de remplissage du buffer à la réception dans les trois cas possibles : F horloge réception = F horloge émission, F horloge réception < F horloge émission et F horloge réception > F horloge émission.



fonctionnement normal



H réception < H émission



H réception > H émission

Le seul moyen permettant une réception correcte sans perte de données est l'asservissement de l'horloge du décodeur sur l'horloge du codeur. Cet asservissement doit se faire dans le cas d'une transmission immédiate (diffusion par satellite) mais aussi dans celui d'une lecture différée (lecture sur un CD-ROM).

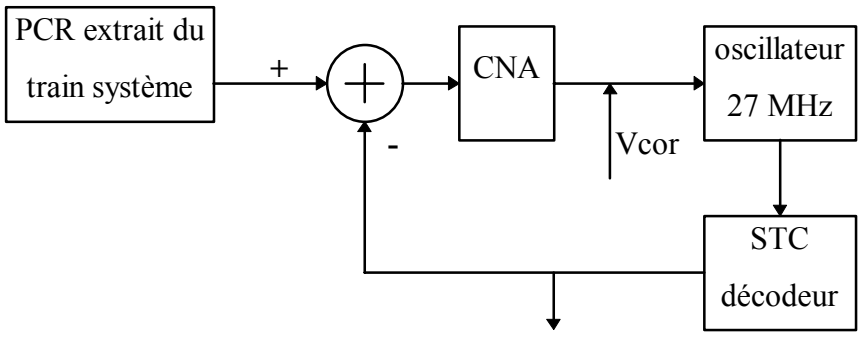
5.1.3 Synchronisation de l'horloge du décodeur

Il existe une horloge système STC (System Time Clock) à 27 MHz (précision ± 30 ppm) qui sert à synchroniser toutes les opérations dans le codeur. Pour obtenir au décodeur une image de la STC codeur, on insère périodiquement dans le train binaire système un champ PCR (Program Clock Reference) qui est égal à la STC au moment de son insertion. La période d'insertion maximale est égale à 100 ms (spécification MPEG-2) ou 40 ms (spécification DVB). Ce champ est codé en deux parties :

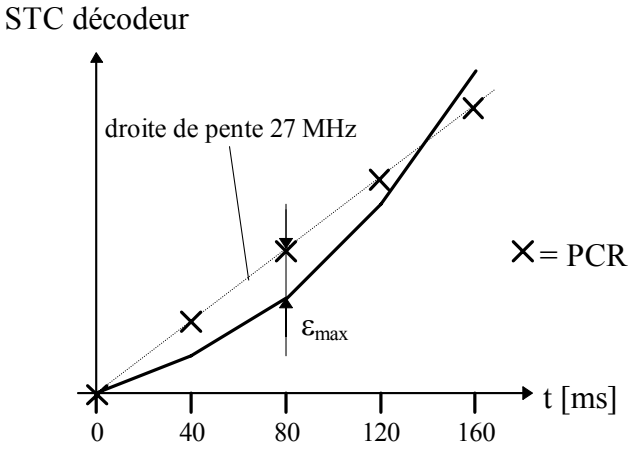
- PCR_base (33 bits) représente un nombre de périodes d'une horloge à 90 kHz. Il est utilisé pour la compatibilité avec MPEG-1.

- PCR_extension (9 bits) associé avec PCR_base représente un nombre de périodes de l'horloge à 27 MHz.

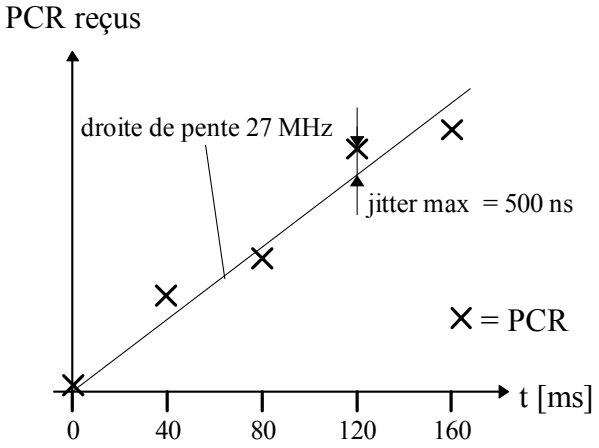
Le schéma simplifié de la STC décodeur est le suivant :



L'évolution de la STC décodeur est la suivante :



L'erreur maximale de poursuite de la boucle ε_{max} n'est pas définie par la norme. C'est au fabricant de décodeur de veiller à la bonne implémentation du système afin de ne pas saturer les différents buffers du décodeur. Par contre, le jitter maximal de la chaîne de traitement comprenant la lecture de la STC codeur, l'insertion du PCR dans le train binaire système, le démultiplexage du PCR dans le décodeur et la mise à jour de sa STC ne doit pas dépasser 500 ns. Le canal de transmission est supposé parfait (ou bien le décodeur est branché directement sur la sortie du codeur). Le schéma suivant illustre cette tolérance :



En fait, il n'y a pas une seule STC pour tout le codeur MPEG-2, mais autant de STC que de codeurs de services, un service étant généralement constitué d'une voie vidéo, de plusieurs voies sons et de données. En effet, chaque codeur de service est asservi en phase et en fréquence sur le signal source vidéo numérique de type CCIR656 (généralement 270 Mbit/s) présent à son entrée. Il faudra donc envoyer un champ PCR par service.

5.1.4 Synchronisation du son et de l'image

La compression du signal vidéo utilise des mécanismes qui produisent un débit variable au cours du temps. Ce débit varie au gré de la nature des images ou des changements de plans. Bien que le codeur MPEG incorpore un mécanisme de régulation de débit, il est nécessaire de placer un buffer de grande taille (1,75 x 1 Mbit = 1835008 bits en MP@ML) après le codeur. En effet, avec un débit de 6 Mbit/s, la taille moyenne des images comprimées toutes les 40 millisecondes vaut (GOP standard M=13, N = 3) :

image I = 66 Ko, image P = 41 Ko, image B = 19 Ko.

avec de fortes variations autour de cette moyenne puisque le train binaire vidéo est composé d'un tiers de codes à longueur fixe et de deux tiers de codes à longueur variable. Ce buffer est alimenté par le débit variable issu du codeur vidéo et vidé à débit constant. Un buffer de même taille doit être inséré avant le décodeur vidéo car le processus de décodage ne se fait pas non plus à débit constant. On peut considérer en première approximation que le processus de décodage est symétrique de celui du codage. Afin de garantir le synchronisme entre le buffer du codeur et le buffer du décodeur (ni overflow, ni underflow), un modèle théorique de décodeur vidéo est incorporé dans le codeur : c'est le VBV (Video Buffer Verifier). Son principe de base est le suivant :

- le VBV est alimenté à débit constant,
- le décodage des images est effectué instantanément.

En ce qui concerne l'audio, le débit est constant (pour les couches I et II) à la sortie du codeur, mais on insère quand même un buffer de petite taille (3584 octets en MP@ML) afin de tamponner les données avant le décodeur audio.

Chaque affluent élémentaire est découpé en paquets de taille variable qui constituent ainsi des PES (Packetized Elementary Stream). Ces paquets sont constitués d'un en-tête de paquet (packet header) suivi des données proprement dites. L'en-tête peut contenir un marqueur de décodage DTS (Decoding Time Stamp) et/ou un marqueur de présentation PTS (Presentation Time Stamp). Ces marqueurs PTS/DTS ont deux fonctions principales :

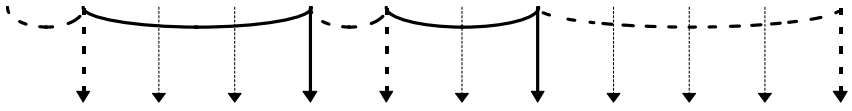
1. Assurer le fonctionnement correct (ni overflow, ni underflow) des buffers de décodage audio et vidéo,

2. Assurer la synchronisation du son et de l'image. On considère que la désynchronisation entre le son et l'image ne doit pas dépasser 40 à 80 ms pour qu'elle ne soit pas perçue par le téléspectateur.

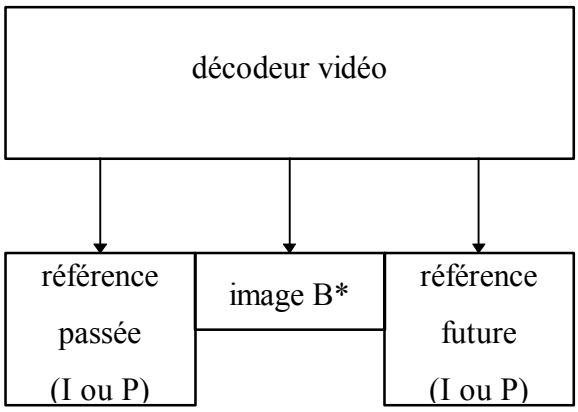
En fait, ces deux fonctions sont réalisées simultanément. Les champs PTS/DTS sont codés sur 33 bits et représentent un nombre de périodes d'horloge à 90 kHz indiquant une heure absolue (la valeur de la STC codeur divisée par 300). Ils sont émis au maximum toutes les 0.7 secondes. Le découpage en PES des affluents élémentaires peut être arbitraire (il n'y a pas nécessairement une image ou une trame audio par PES). Il n'y a pas non plus obligatoirement un PTS/DTS par PES ni même un couple PTS/DTS à chaque envoi. En réalité, on n'envoie un DTS que si le PTS est présent et que la valeur du PTS est différente de la valeur du DTS.

La valeur des PTS et DTS correspond aux heures suivantes : DTS = heure absolue de vidage du buffer du décodeur audio ou vidéo, PTS = heure absolue de présentation de l'image ou de la trame audio en sortie du décodeur. Ces heures s'appliquent à l'arrivée du premier octet du premier Picture Start Code trouvé dans le PES ou bien à l'arrivée du premier octet de la trame audio.

Ces deux variables sont redondantes, la valeur de l'une pouvant être déduite de la valeur de l'autre. Dans le cas du train binaire audio, on a PTS = DTS car le décodage d'une trame audio implique sa présentation puisqu'il n'y a pas de buffer à la sortie du décodeur. Dans le cas du train binaire vidéo, la situation est plus complexe car l'ordre des images comprimées n'est pas le même que l'ordre des images présentées à la sortie du décodeur. Voyons l'ordre des images à différents endroits de la chaîne de transmission ainsi que la différence entre les valeurs de PTS et de DTS :

Source	I0	B1	B2	P3	P4	B5	P6	B7	B8	B9	P10	P11
Codage / décodage	I0	P3	B1	B2	P4	P6	B5	P10	B7	B8	B9	P11
												
Présentation		I0	B1	B2	P3	P4	B5	P6	B7	B8	B9	P10
PTS = DTS + X [ms]		40	0	0	120	40	0	80	0	0	0	160

L'explication des différences entre PTS et DTS est simple si l'on a en mémoire la structure d'un décodeur vidéo ainsi que le rôle des différents types d'image :



mémoires d'images extérieures

* les images de référence future et passée doivent être entièrement mémorisées. On peut décoder les images B en ne mémorisant que leur trame 1 (on a donc obligatoirement présentation = décodage).

référence passée	image B	référence future	présentation
I0	-	-	-
I0	-	P3	I0
I0	B1	P3	B1
I0	B2	P3	B2
P3*	-	P4	P3
P4	-	P6	P4
P4	B5	P6	B5
P6	-	P10	P6
P6	B7	P10	B7
P6	B8	P10	B8
P6	B9	P10	B9
P10	-	P11	P10

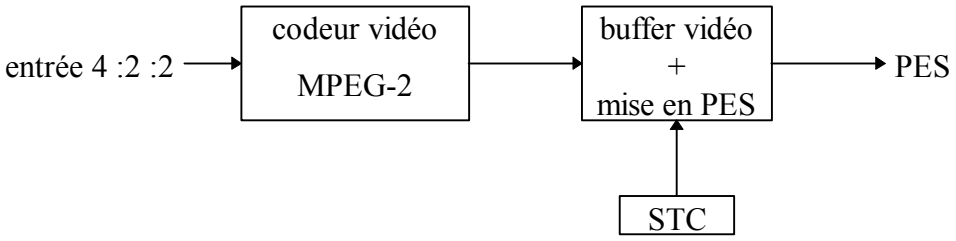
* en réalité, la référence future n'est pas copiée dans la référence passée, seul le pointeur change.

Si on considère le temps de codage (ou de décodage) comme étant nul (modèle VBV), on doit avoir un ordre de vidage du buffer (ou ordre de décodage) toutes les 40 ms. Le décodeur ne recevant pas un DTS à chaque image, il doit :

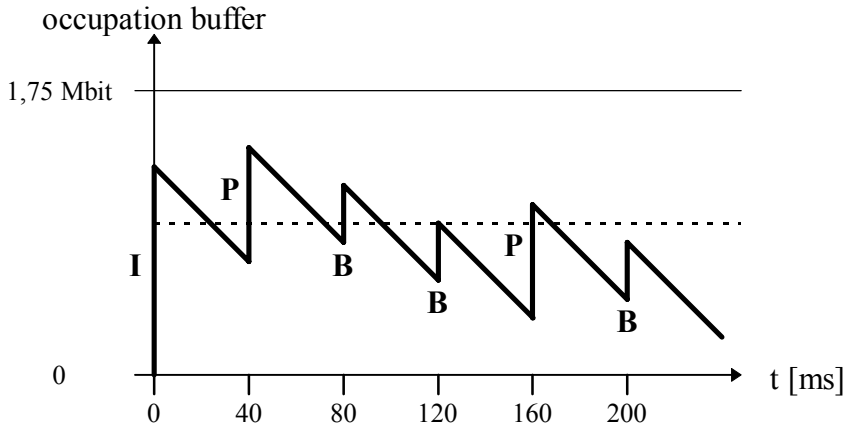
1. mémoriser le DTS reçu et décoder une image,
2. décoder une nouvelle image toutes les périodes de références (40 ms) s'il ne reçoit pas de nouveau DTS.

Le mécanisme est identique dans le cas de l'audio, avec une période de référence égale à la durée d'une trame (36 ms à 32 kHz, 26.1 ms à 44.1 kHz et 24 ms à 48 kHz). Le PTS peut être calculé localement pour la vidéo en utilisant la règle vue précédemment lors de la remise en ordre des images. On peut déduire le DTS de la valeur du PTS en faisant l'opération inverse. Nous en avons fini avec la synchronisation du son et de l'image, vérifions que le bon fonctionnement des buffers est aussi assuré. L'insertion du PES contenant le DTS est

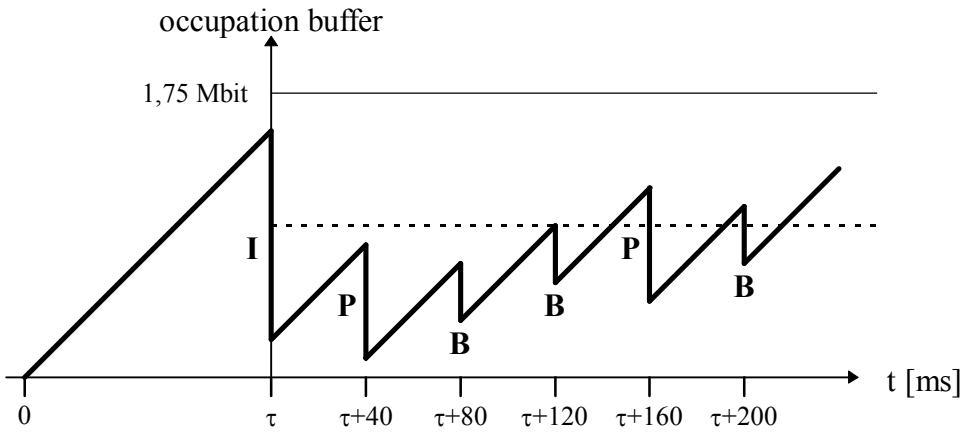
fonctionnellement assuré derrière le buffer du codeur vidéo. En fait, l'insertion d'une donnée dans le train binaire augmente le débit et nécessite donc un mécanisme de buffer. Il est donc plus commode d'insérer directement le PES dans le buffer du codeur :



L'analyse du remplissage des buffers du codeur et du décodeur implique obligatoirement la relation DTS = STC codeur + τ.



buffer vidéo codeur



buffer vidéo décodeur

Un raisonnement par l'absurde suffit à se convaincre de la nécessité du retard τ. Si τ était nul, on devrait commencer à décoder une image dont on aurait reçu que le premier octet (le DTS correspond à l'heure de vidage du buffer appliqué au premier octet du Picture Start Code). Le retard est donc indispensable pour s'assurer que les données à décoder instantanément sont bien présentes dans le buffer du décodeur. L'ordre de grandeur de τ est d'une centaine de millisecondes. L'envoi (ou la reconstitution) d'un DTS toutes les 40 ms assure automatiquement la synchronisation des buffers codeur et décodeur et évite l'underflow et l'overflow.

5.1.5 Train programme et transport

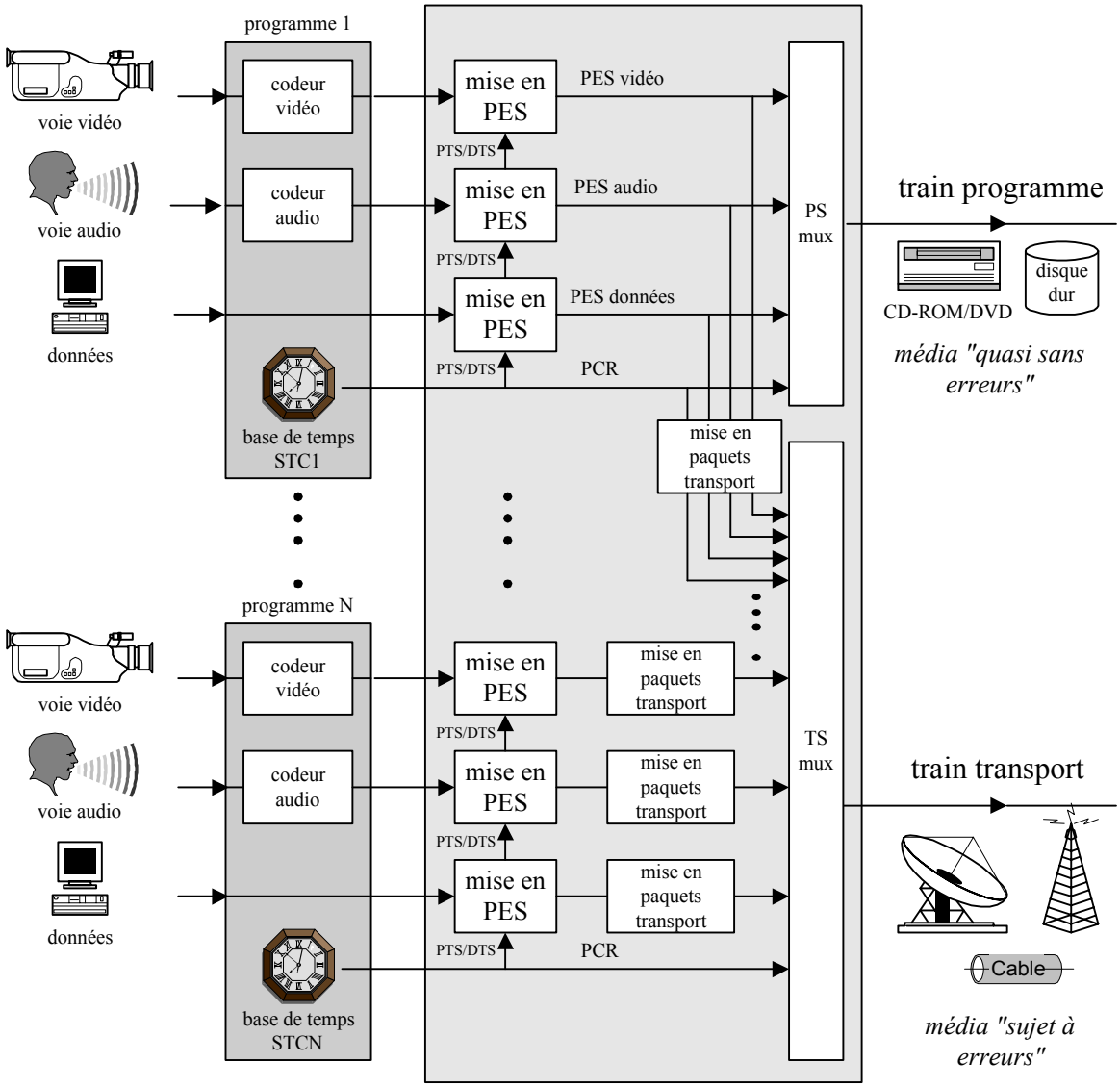
Comme on l'a vu au paragraphe précédent, le premier niveau d'empaquetage est la mise en PES. Un train programme, destiné à un média sans perte, est composé d'un faible nombre de PES (généralement un PES vidéo, quelques PES audio (pour le doublage) et quelques PES de données (pour le sous-titrage)). Ce mélange des PES pour former le train programme est appelé multiplexage (PS mux). La spécification du program stream MPEG-2 reprend en grande partie la spécification MPEG-1. L'identification des affluents élémentaires se fait grâce au champ stream_ID qui permet d'identifier 16 voies vidéo, 32 voies audio et 3 voies de données. Destiné à l'origine au CD-ROM, le program stream sera certainement utilisé pour le DVD.

Le program stream MPEG-1 n'était absolument pas prévu pour la diffusion. En effet, celle-ci provoque de nombreuses erreurs de transmission et il est nécessaire de protéger le train binaire par de puissants codes détecteurs et correcteurs d'erreurs tels que les codes convolutionnels ou Reed-Solomon. Mais ce dernier ne peut être appliqué que sur des paquets de petites tailles (quelques centaines d'octets) pour des raisons matérielles. Il faut donc découper à nouveau le train binaire en paquets de longueur 188 octets appelés paquets de transport. Cette opération de mise en paquets de transport se fait sur le train programme vu précédemment pour former le train transport (transport stream). Le train transport est donc un sur-ensemble du train programme ce qui autorise les conversions entre les deux types de programme.

Le train transport incorpore un mécanisme différent d'identification des affluents élémentaires (ES : Elementary Stream) basé sur les tables PSI (Program Specific Information)

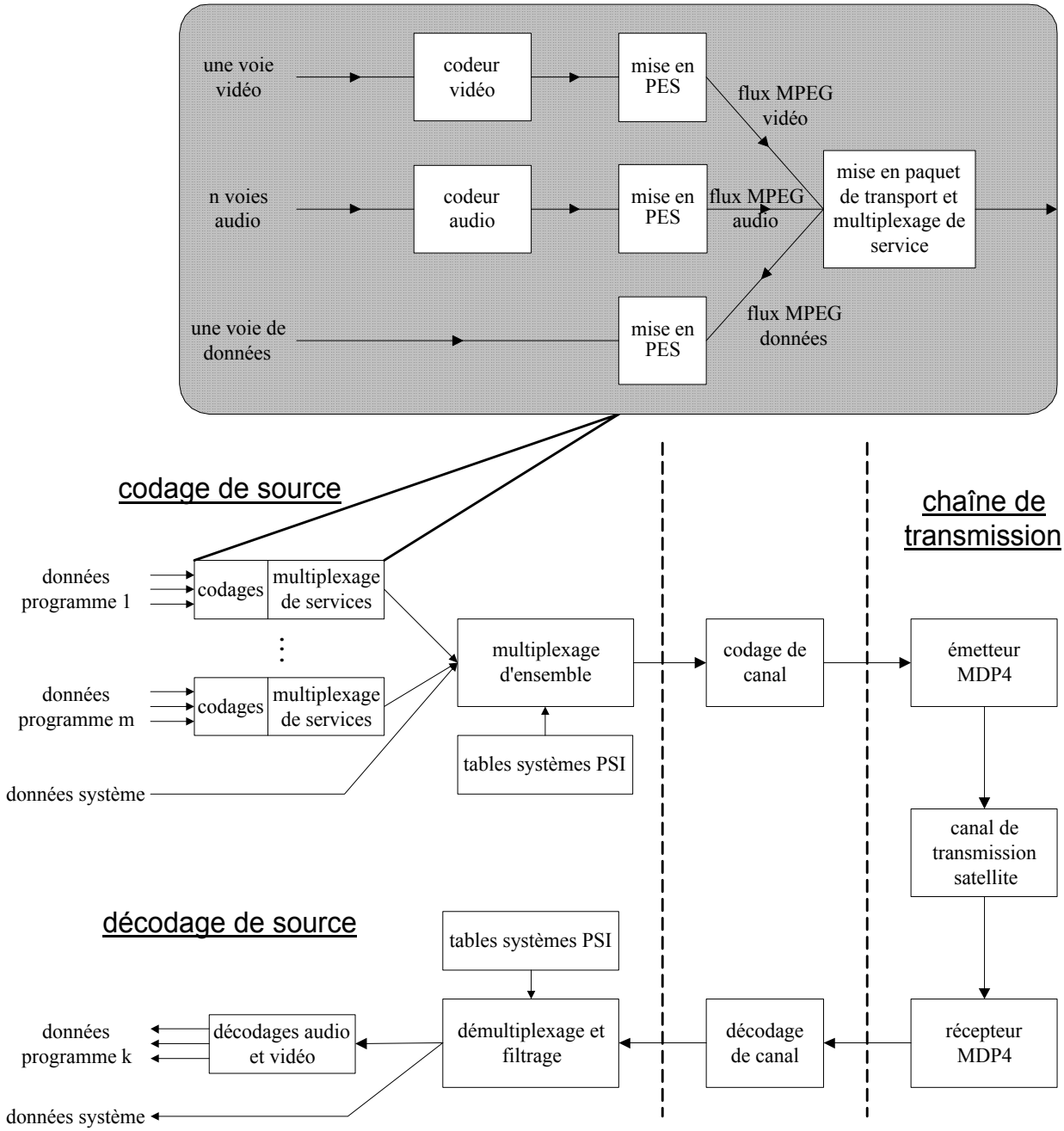
normalisées en partie par MPEG et en partie par DVB. Le champ stream_ID dans l'en-tête du PES n'est alors plus utilisé pour identifier les ES, mais il faut quand même assurer sa cohérence (ne pas utiliser un stream_ID audio pour identifier un ES vidéo par exemple).

Le champ PCR, nommé SCR en program stream, est soit inclus dans l'entête de PES, soit inséré directement dans un paquet de transport.



5.1.6 Le train transport MPEG2

La figure suivante est une vue d'ensemble de la chaîne DVB satellite. On notera les deux niveaux de multiplexage dans le codeur : le multiplexage de service et le multiplexage d'ensemble.



Un paquet de transport se compose (voir : figure suivante) d'un en-tête de paquet de 4 octets et d'une charge utile (payload) éventuellement précédée (ou entièrement remplacée) d'un champ d'adaptation de longueur variable. La charge utile est constituée des PES transportant les programmes TV transmis par le canal, ainsi qu'un certain nombre de données auxiliaires permettant au décodeur de se retrouver dans le train transport.

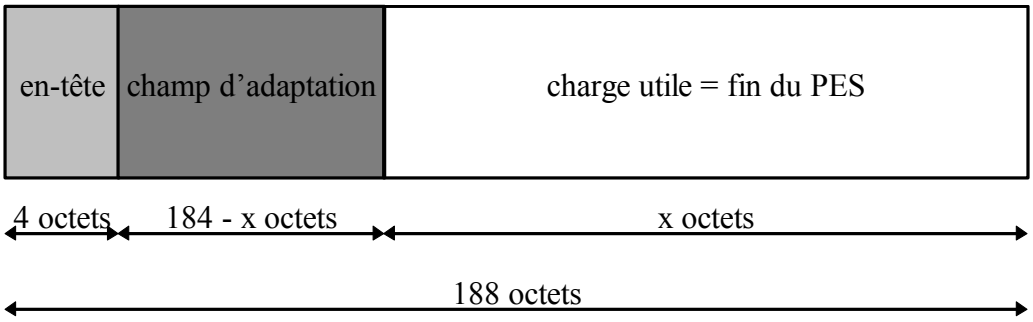
sync byte	transport packet error indicator	payload unit start indicator	transport priority	PID	transport scrambling control	adaptation field control	continuity counter	données ou champ d'adaptation
8 bits	1 bit	1 bit	1 bit	13 bits	2 bits	2 bits	4 bits	jusqu'a 184 octets

L'en-tête du paquet est composé des champs suivants :

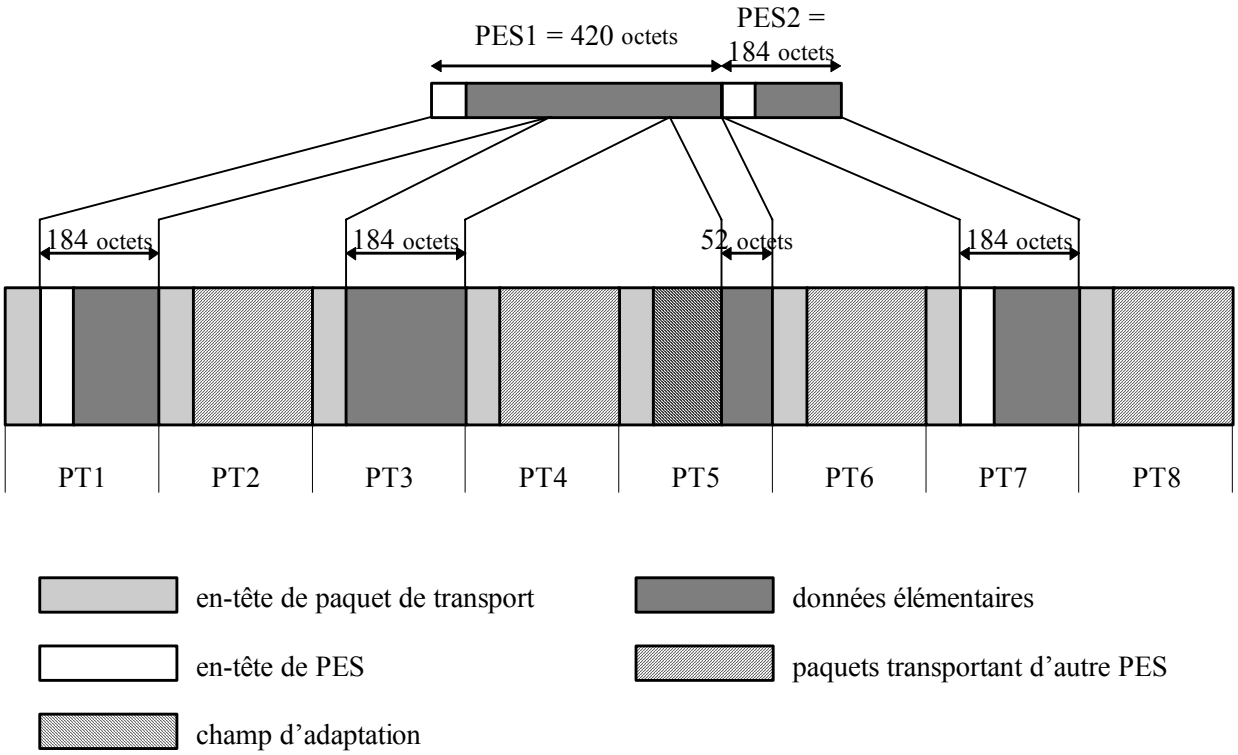
- sync byte. octet de synchronisation 47h indiquant le début du paquet.
- transport packet error indicator. Ce bit est positionné par le décodage de canal. Il indique que le paquet contient des erreurs (entre 8 et 16) qui n'ont pas pu être corrigées.
- payload unit start indicator. La signification de ce bit varie selon que les données d'un PES sont présentes dans le paquet de transport ou non :
  - Données PES présentes. Un '1' indique que le paquet débute par un en-tête de PES, '0' sinon.
  - Données PSI présentes. Un '1' indique que le premier octet du paquet donne l'adresse du premier octet d'une section PSI. '0' sinon.
- transport priority. Indicateur de priorité sur le paquet en cours.
- PID. L'identificateur de paquet indique à quelle voie est attribué le paquet.
- transport scrambling control. La valeur 00 indique que le paquet est en clair, les autres valeurs indiquent quel mot de contrôle est utilisé pour l'embrouillage.
- adaptation field control. Ces deux bits renseignent sur le contenu du paquet (champ d'adaptation, données ou champ d'adaptation et données).
- continuity counter. Ce compteur tournant de 0 à 15 s'incrémente avec chaque paquet de transport de même PID sauf s'il contient seulement un champ PCR. Ce compteur permet de détecter un paquet perdu ou un paquet mal identifié.

Un paquet de transport ne peut transporter que des données issues d'un seul PES. De plus, un paquet PES débute obligatoirement au début d'un paquet de transport et se termine obligatoirement à la fin d'un paquet de transport. Comme le découpage des paquets PES (et donc leur longueur) est indépendant du découpage en paquets de transport, il n'y a pas de raisons pour que la longueur des paquets PES soit un multiple de 184 octets. Ainsi, le dernier paquet de transport d'un paquet PES devra débuter par un champ d'adaptation (AF :

Adaptation field) dont la longueur sera le complément à 184 du nombre d'octets x restant à transmettre pour terminer ce paquet PES.



Outre ce rôle de bourrage, le champ d'adaptation est aussi utilisé pour la transmission du champ PCR. Dans ce cas particulier (ainsi que pour des données privées), le paquet de transport peut n'être constitué que d'un champ d'adaptation. La figure suivante illustre la formation d'un train transport :



Le PES1 est transporté par les paquets PT1, PT3 et PT5. Le PES2 tient exactement dans PT7.

5.1.7 Les tables SI

Un multiplex transport MPEG-2 peut transporter plusieurs programmes (de l'ordre de la dizaine pour un répéteur satellite) composés chacun d'un ou plusieurs trains élémentaires. Pour que le décodeur puisse se retrouver dans ce flot d'informations, MPEG-2 définit 4 types de tables (3 réellement indispensables) dont l'ensemble constitue l'information spécifique aux programmes (PSI : Program Specific Information).

1. La PAT, « Program Association Table ». Cette table, dont la présence est obligatoire, est transportée par les paquets dont le PID vaut 0x0000. Elle associe, pour chaque programme convoyé par le train transport, le numéro de programme (de 0 à 65535) et le PID des paquets transportant la PMT (carte du programme). La PAT est toujours transmise en clair, même si tous les programmes sont embrouillés.
2. La PMT, « Program Map Table ». Il y en a une par programme présent dans le multiplex. Elle indique en clair les PID des trains élémentaires constituant le programme ainsi que le PID du paquet portant le PCR (qui peut être celui d'un des trains élémentaires constituant le programme). Elle peut aussi donner les PID des données privées (qui peuvent être embrouillées) relatives au programme comme par exemple les ECM pour le contrôle d'accès.
3. La CAT, « Control Access Table ». Cette table, dont la présence est obligatoire dès qu'un programme est à accès conditionnel, est transportée par les paquets dont le PID vaut 0x0001. Elle indique les PID des paquets transportant les EMM pour un ou plusieurs systèmes de contrôle d'accès.
4. La PT, « Private Table ». Ces tables privées contiennent des données privées utilisées par exemple pour le contrôle d'accès (EMM et ECM).

A ces tables, la norme DVB ajoute des informations complémentaires (DVB-SI, Service Information) permettant au récepteur de se configurer automatiquement et à l'utilisateur de naviguer dans les nombreux services offerts au moyen du guide électronique de programmes (EPG, Electronique Program Guide). Ces informations se composent de 4 tables principales :

1. La NIT, « Network Information Table ». Cette table transporte des informations spécifiques relatives à un réseau constitué de plusieurs canaux de transmission (donc de plusieurs trains transports indépendants), telles que les fréquences et/ou les numéros de canaux du réseau utilisés lors de la configuration du décodeur. Cette table constitue par définition le programme 0 du multiplex.

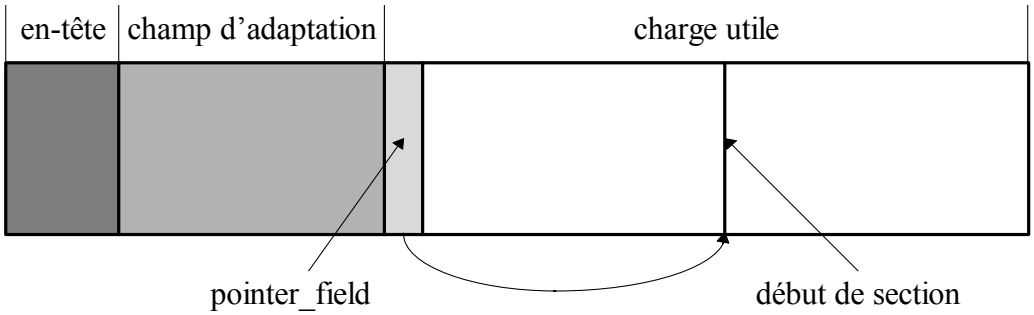
2. La SDT, « Service Description Table ». Cette table liste les noms et d'autres paramètres associés à chaque service d'un même multiplex.
3. L'EIT, « Event Information Table ». Cette table est utilisée pour la transmission d'informations relatives aux événements en cours ou à venir dans le multiplex et éventuellement sur d'autres multiplex.
4. La TDT, « Time and Date Table ». Cette table est utilisée pour la remise à l'heure de l'horloge interne du récepteur.

et de trois tables facultatives :

1. La BAT, « Bouquet Association Table ». Cette table est utilisée pour grouper la présentation, à l'utilisateur, de bouquets de services associés.
2. La RST, « Running status Table ». Cette table est transmise pour la mise à jour rapide d'un ou de plusieurs événements au moment où un changement se produit (à la différence des autres tables qui sont transmises de manière répétitive).
3. La ST, « Stuffing Tables ». Ces tables de bourrage sont utilisées par exemple pour invalider des tables devenues inutiles.

La fréquence de répétition des tables n'est pas imposée par la norme, mais elle doit être suffisante (10 à 50 fois par seconde) pour permettre au décodeur d'accéder rapidement au programme recherché. Chaque table est constituée d'une ou de plusieurs sections (au maximum 256), chacune de longueur maximale 1024 octets (sauf pour les tables privées où elle peut atteindre 4096 octets).

Contrairement aux PES, les sections ne débutent pas au début d'un paquet de transport et ne finissent pas à la fin d'un paquet de transport. Lorsqu'une section débute dans un paquet, le bit d'en-tête payload_unit_start_indicator est mis à 1. Le paquet peut commencer par la fin d'une autre section, précédée ou non d'un champ d'adaptation. Le premier octet de la charge utile, appelé pointer_field, donne le décalage du début de la nouvelle section par rapport à cet octet.



5.1.8 Décodage du multiplex MPEG2

Les étapes suivantes sont effectuées (sans contrôle d'accès) pour rechercher un programme dans un multiplex transport MPEG-2. Dès l'accord sur le canal réalisé, il faut :

1. filtrer le PID 0 pour acquérir les paquets transportant les sections de la PAT et construire cette table,
2. puis présenter le choix de programmes disponibles à l'utilisateur.

Quand l'utilisateur a choisi son programme, il faut :

1. filtrer le PID correspondant à la PMT de ce programme et construire cette table,
2. filtrer le paquet indiqué par le champ PCR_PID, extraire le PCR et synchroniser la STC décodeur,
3. et s'il y a plusieurs PID audio et vidéo pour ce programme, présenter le choix à l'utilisateur.

Quand le choix est fait, il faut :

1. filtrer les PID correspondants et décoder les affluents élémentaires.

La partie visible par l'utilisateur de ce processus est la présentation interactive du guide électronique des programmes (EPG) généralement associé au réseau au moyen des informations fournies par les tables PSI et DVB-SI pour lui permettre de naviguer facilement dans les dizaines de programmes et services mis à sa disposition. Toutefois, le zapping n'est pas aussi rapide avec la télévision numérique qu'avec les émissions analogiques, car le processus d'acquisition du programme est relativement long (de l'ordre de la seconde pour le service par satellite) en raison des opérations à effectuer (synchronisation RF, acquisition du programme et attente de la première image I).

5.2 Embrouillage et contrôle d'accès

5.2.1 Introduction

Il est certain que la plupart, sinon la totalité, des services de télévision numérique seront payants (au moins au début). Le problème de l'embrouillage, et donc du contrôle d'accès aux services embrouillés est crucial pour tous les opérateurs et a donc fait l'objet d'une normalisation par le groupe DVB. La norme prévoit le transport de données de contrôle d'accès que l'on retrouve au moyen de la table de contrôle d'accès CAT et de la PMT. La norme définit aussi un algorithme commun d'embrouillage (Common Scrambling Algorithm CSA) pour lequel un compromis coût/complexité a été choisi avec un objectif de résistance aux pirates suffisamment long (2010).

Par contre, l'accès conditionnel lui-même n'est pas spécifié par la norme car chaque opérateur préfère avoir son propre système, même s'il existe entre eux de grandes similitudes dues notamment à leur filiation avec le système eurocrypt. Afin d'éviter un empilage de boîtiers décodeurs chez l'abonné désireux d'avoir accès à plusieurs réseaux, deux options ont été prévues : le simulcrypt et le multicrypt.

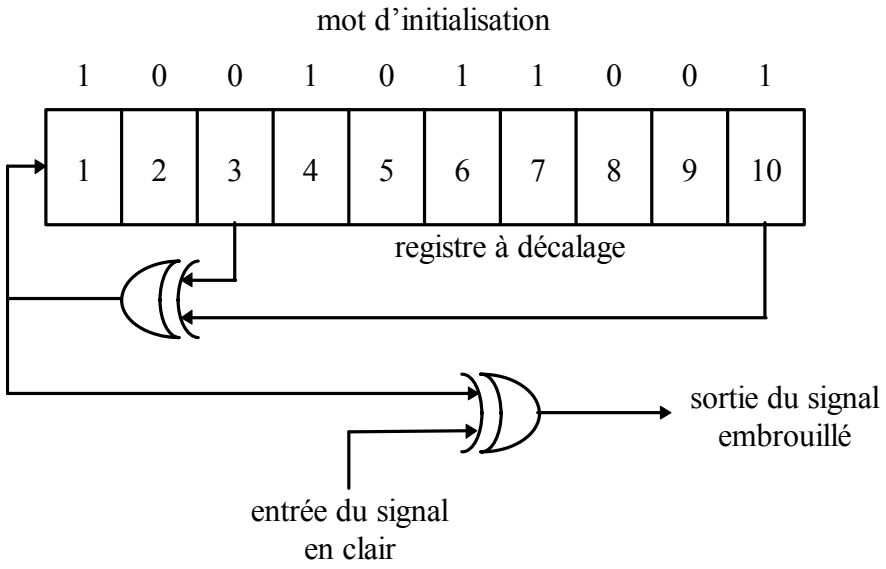
Avant d'aborder les divers mécanismes en œuvre dans le contrôle d'accès, il convient de définir deux termes :

- L'embrouillage. C'est l'opération destinée à transformer un signal numérique en un signal numérique aléatoire ou pseudo-aléatoire de même débit binaire, en vue d'en faciliter la transmission ou de le rendre inintelligible. En télévision à péage (radiodiffusée ou distribuée), le signal émis est embrouillé de telle sorte que, pour le désembrouiller, il soit nécessaire de fournir au désembrouilleur (de structure connue) des indications (mots de contrôle) dont la détention est assujettie au paiement.
- Le chiffrement ou cryptage. Le cryptage est l'art de coder un message de façon à le rendre incompréhensible sauf pour son destinataire. Tout système de cryptage est composé d'un algorithme de codage plus ou moins compliqué utilisant ou non une ou plusieurs clés de sécurité et il est, en principe, conçu de manière à être inviolable. Les deux codes le plus utilisés dans les années 1990, car considérés comme pratiquement inviolables, sont le DES (*data encryption standard*), développé en 1977 par I.B.M. et approuvé par le National Bureau of Standards des Etats-Unis pour son utilisation par l'administration américaine, et

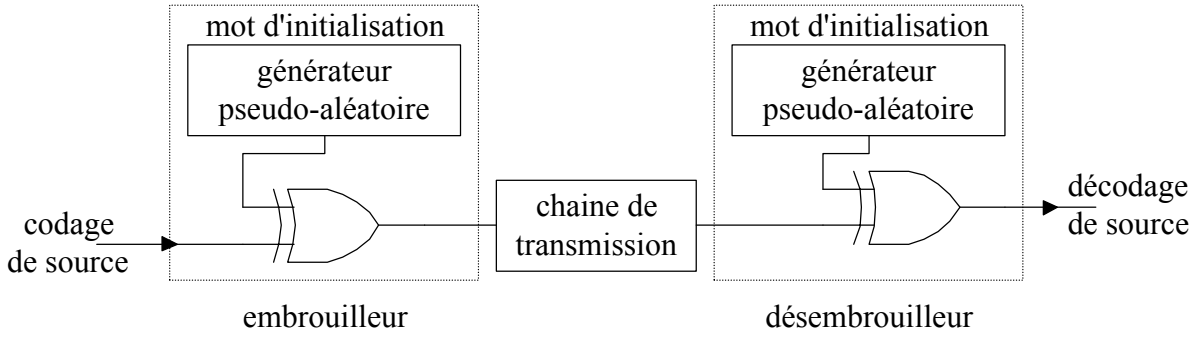
le RSA (d'après Rivest, Shamir et Adleman, les auteurs du code), qui est de plus en plus employé car le codage se fait à l'aide d'une clé publique tandis que le décodage se fait par une clé connue du seul destinataire du message codé.

5.2.2 L'embrouillage

La spécification de l'algorithme CSA utilisé par la norme DVB n'est bien entendu pas publique. Le principe général de l'embrouillage n'en demeure pas moins simple. Un exemple d'embrouilleur est représenté sur le schéma suivant :



Le même générateur pseudo-aléatoire existe à la réception (le désembrouilleur) pour former la chaîne complète :



Les mots d'initialisation changent périodiquement, en synchronisme à l'émission et à la réception.

Dans DVB, l'algorithme d'embrouillage est beaucoup plus complexe tout en restant assez facilement implantable dans un circuit intégré. Le nombre de paquets ayant un PID différent et pouvant être désembrouillés en parallèle dépend de l'implémentation matériel du démultiplexeur (environ une dizaine). Le CSA consiste en un chiffrement à deux couches indépendantes dont chacune pallie aux éventuelles faiblesses de l'autre. Il utilise un mot d'initialisation, appelé mot de contrôle, de longueur 64 bits. Il est prévu, dans la norme MPEG-2, d'utiliser deux mots de contrôle alternativement (avec une période d'environ 10 secondes) pour embrouiller le signal : le mot de contrôle pair (CWE « Control Word Even ») et le mot de contrôle impair (CWO « Control Word Odd »).

L'embrouillage dans MPEG peut porter sur deux niveaux selon que l'on utilise un train programme ou un train transport :

- Au niveau PES (train programme). L'embrouillage a généralement lieu avant multiplexage, les deux bits du champ de l'en-tête PES_scrambling_control indiquant le mode utilisé :

PES_scrambling_control	signification
00	pas d'embrouillage
01	pas d'embrouillage
10	embrouillage avec mot pair
11	embrouillage avec mot impair

  L'embrouillage doit se faire sur des tronçons de 184 octets (pour permettre la conversion du program stream en transport stream), l'en-tête du PES n'étant pas embrouillé.

- Au niveau paquet de transport (train transport). L'embrouillage est réalisé après multiplexage sur l'ensemble de la charge utile du paquet de transport. Le PCR contenu dans un champ d'adaptation est en clair. Le début des en-têtes de PES n'est pas embrouillé, mais les champs PTS/DTS le sont. Les deux bits du champ

transport_scrambling_control (dans l'en-tête du paquet de transport) indiquent le mode utilisé :

transport _scrambling_control	signification
00	pas d'embrouillage
01	embrouillage avec mot par défaut
10	embrouillage avec mot pair
11	embrouillage avec mot impair

5.2.3 Les mécanismes de contrôle d'accès

Le contrôle d'accès utilise deux tables SI, la PMT et la CAT, ainsi que deux types de messages spécifiques : les EMM « Entitlement Management Messages » et les ECM « Entitlement Control Messages ». Ces messages sont contenus dans des tables privées avec plusieurs identificateurs de table TID possibles. La syntaxe des données privées est particulière à chaque système de contrôle d'accès et ne fait l'objet d'aucune normalisation. C'est le principal motif d'incompatibilité entre deux décodeurs (TPS et Canal satellite par exemple).

Il y a dans le mécanisme de contrôle d'accès deux clés de chiffrement : la clé utilisateur et la clé de service. La clé utilisateur permet de déchiffrer la clé de service qui permet de déchiffrer les mots de contrôle. Un système de contrôle d'accès peut comprendre plusieurs opérateurs indépendant ayant chacun leur base de données gérant les droits des abonnés. La syntaxe des EMM et des ECM reste la même, mais les clés de services sont différentes. Quand on a plusieurs systèmes de contrôles d'accès, la syntaxe des EMM et des ECM est différente ainsi que les bases de données gérant les droits des abonnés.

Les données privées des ECM sont générées à partir des mots de contrôles et de la clé de service. Les données privées des EMM sont générées à partir de la clé de service et de la clé utilisateur. Il y a dans le décodeur un lecteur qui lit les cartes à puce (MSD « Module de Sécurité Détachable »). Le MSD possède un numéro et contient les droits de l'utilisateur sous la forme d'une carte binaire (bitmap).

Un ECM (quelques dizaine d'octets) contient les deux mots de contrôle (pair/impair) chiffrés d'une composante (affluent élémentaire). Chaque ECM est envoyé 5 à 10 fois par seconde. Il y a un ou plusieurs ECM par composante. On trouve :

- un ECM par mode d'accès et par opérateur,
- une famille d'ECM par système de contrôle d'accès.

Les EMM permettent la gestion des droits de l'utilisateur grâce au numéro de MSD qui permet l'accès à une carte utilisateur particulière. Ils autorisent la mise à jour ou la suppression des droits de l'abonné. Ces EMM sont contenus dans une base de données gérée par l'opérateur. Celle-ci est périodiquement transmise dans sa totalité. Avec un débit réservé de 500 Kbit/s, il faut environ 1/2 heure pour transmettre une base de données de 500000 abonnés. Dans le cas d'un bouquet satellite, le même signal est transmis sur tous les répéteurs. On trouve plusieurs familles d'EMM tels que :

- les EMM_M (message). Ils permettent d'envoyer un message (de type texte) sur tous les MSD (comme, par exemple, de ne pas oublier de se réabonner).
- les EMM_G (groupe). Grâce à ces EMM, on peut définir des groupes d'utilisateurs qui seront mis à jour simultanément. Ce mécanisme permet, par exemple, d'empêcher la diffusion d'un événement sportif dans une région géographique donnée.
- Les EMM_U (utilisateur). Ils permettent la gestion d'un utilisateur donné grâce au numéro de MSD qu'ils contiennent.

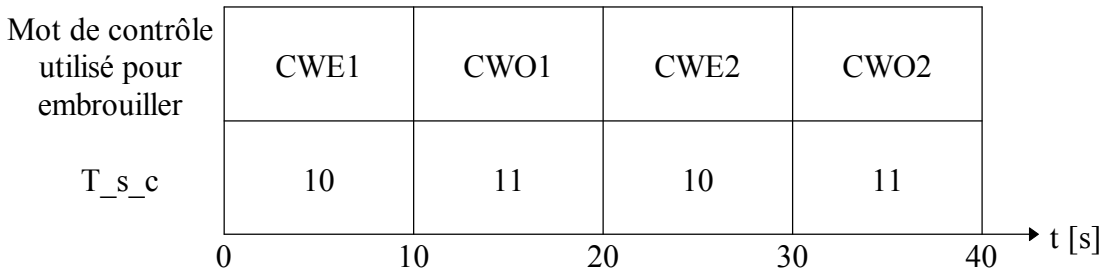
Le MSD permet, grâce à la clé utilisateur, de déchiffrer les EMM et donc de mettre à jour les droits. Puis il déchiffre, grâce à la clé de service de l'opérateur, les mots de contrôle contenus dans les ECM qui sont mis en mémoire dans le décodeur (pour une durée de 10 secondes). Les tables PAT, CAT et PMT sont transmises en clair. On y trouve les données suivantes :

- La PAT (PID = 0) contient (entre autre) le PID de la PMT du programme considéré.
- On a, dans la PMT, les PID des composantes du programme ainsi que ceux des ECM associés.
- La CAT (PID = 1) contient les PID des EMM d'un (ou plusieurs) opérateurs d'un (ou plusieurs) systèmes de contrôle d'accès.

5.2.4 Désembrouillage du multiplex MPEG2

Nous allons maintenant nous placer dans le cas d'un train transport. Le temps moyen (non-interruptible) pris par le MSD pour déchiffrer un ECM est de l'ordre de 50 ms. Or les ECM d'une composante sont émises 5 à 10 fois par seconde pour faciliter la vitesse du zapping. Afin de ne pas bloquer en permanence le décodeur, le MSD ne doit déchiffrer l'ECM que lors du premier changement (une fois toutes les 10 secondes) de mot de contrôle. Afin de repérer

facilement ce changement, le TID de la table privée indique le mot de contrôle (pair ou impair) qui change dans l'ECM. Lors du premier changement de TID sur le PID concerné, l'ECM est envoyé au MSD et les deux mots de contrôle sont déchiffrés puis mis en mémoire dans le décodeur. Seul le mot de contrôle qui n'est pas en cours d'utilisation change (sinon le décodage est impossible) de façon à être utilisé lors du prochain changement pair/impair dans le codeur. Le schéma suivant indique l'état des mots de contrôle, des bits transport_scrambling_control, du TID et le contenu de l'ECM au codage et au décodage pour une composante.



A l'émission

A la réception

Voyons maintenant de manière synthétique les différentes étapes effectuées quand un abonné change de chaîne :

1. A l'aide de la PMT, le décodeur fait l'acquisition des PID des composantes du programme ainsi que les ECM associés.
2. Le démultiplexeur filtre les paquets de transport correspondant aux PID des composantes et présente une liste d'ECM associés au MSD.

3. Grâce à son bitmap (qui représente les droits de l'abonné), le MSD va essayer de déchiffrer les ECM dans la liste. C'est la phase de pré-filtrage qui permet de gagner en vitesse de fonctionnement.
4. Si l'opération est réussie, le démultiplexeur est définitivement programmé pour filtrer les PID des ECM. Il présente alors au MSD uniquement les ECM dont le TID vient de changer.
5. L'embrouillage démarre et le programme apparaît à l'écran.

5.2.5 Multicrypt et Simulcrypt

Revenons aux deux options prévues par la norme DVB en cas de systèmes de contrôle d'accès multiples. Dans le cas du simulcrypt, le multiplex transport devra comporter les EMM et les ECM correspondant à chaque système de contrôle d'accès utilisant l'algorithme commun d'embrouillage (CSA). Chaque famille de décodeur peut décoder le train transport puisque toutes les informations nécessaires sont dans le train binaire. Cette technique nécessite un accord entre les opérateurs de télévision payante.

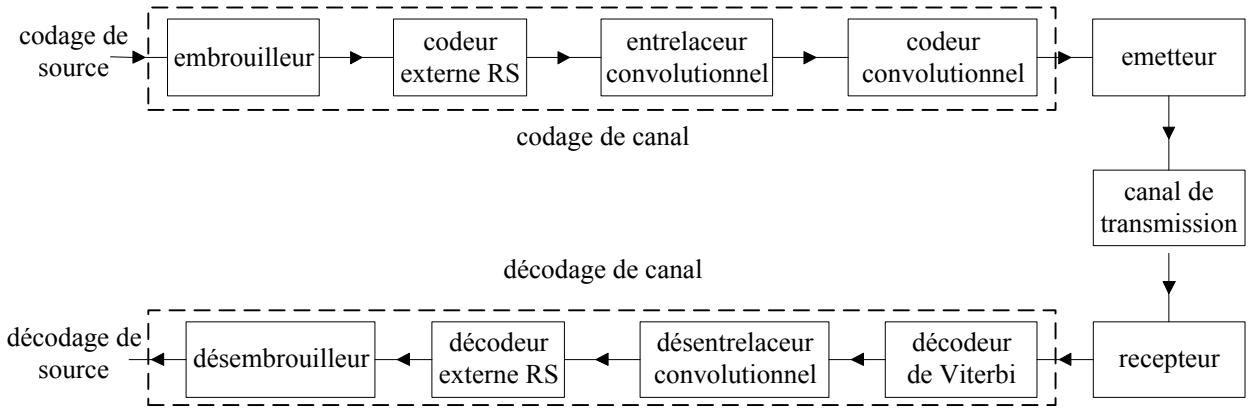
L'autre méthode est appelée multicrypt. Elle utilise un module de contrôle d'accès et de désembrouillage détachable au format PCMCIA inséré sur le chemin des données reçues au niveau transport via une interface normalisée (Common Interface DVB-CI). Elle comporte un bus permettant au module, qui peut inclure un lecteur de carte à puce, de communiquer avec le décodeur. Dans cette approche, un même décodeur dispose de plusieurs slot PCMCIA. L'utilisateur peut donc connecter un module pour chaque réseau utilisant un système de contrôle d'accès et/ou d'embrouillage différent. Cette technique ne nécessite pas d'accord entre les opérateurs de télévision payante. Chaque opérateur vend son module et sa carte à puce à l'abonné, mais le décodeur est standard et peut être acheté séparément. C'est la solution la plus élégante, mais la normalisation de l'interface DVB-CI est encore très récente.

Le procédé simulcrypt est le plus facile à réaliser techniquement comme le démontre l'accord récent passé entre AB-SAT et Canal-plus. Le système multicrypt est lui plus délicat à mettre en œuvre techniquement mais il devrait se généraliser à l'avenir.

6 Le codage de canal

6.1 Introduction

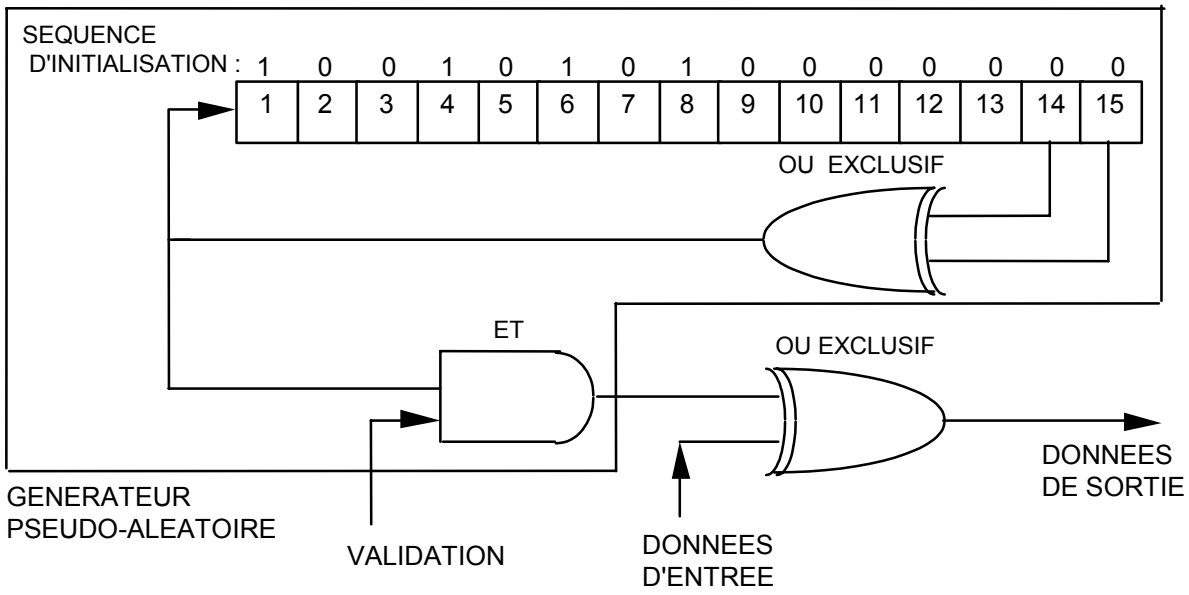
Le rôle de la chaîne de transmission est de diffuser le train binaire sur une zone géographique donnée avec un minimum d'erreurs. Son synoptique est le suivant :



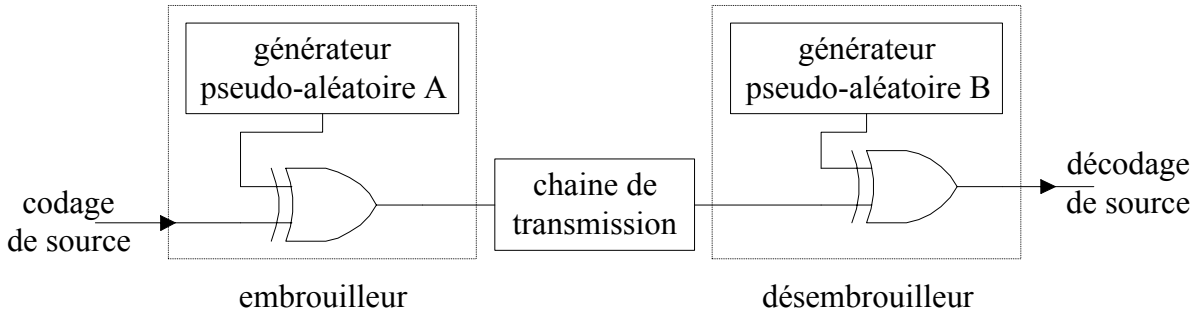
Elle est formée d'un embrouilleur qui rend le train binaire pseudo-aléatoire et d'un code correcteur d'erreurs concaténé formé par la mise en cascade d'un code Reed-Solomon, d'un entrelacement et d'un code convolutionnel. Ce code concaténé sert à minimiser les erreurs de transmission. Finalement, on trouve l'émetteur qui module le signal et le canal de transmission qui assure la diffusion. Au décodage, les opérations inverses sont effectuées pour retrouver le train binaire émis.

6.2 L'ensemble embrouilleur/désembrouilleur

Le train binaire issu du codage de source peut contenir des périodicités. Elles produisent des raies parasites dans le spectre du signal qui nuisent à l'efficacité de la modulation. Afin d'éliminer ces raies, un embrouilleur rend le train binaire aléatoire. Il utilise un générateur pseudo-aléatoire dont le polynôme générateur est $P(x) = 1 + x^{14} + x^{15}$. Le schéma de l'embrouilleur et du désembrouilleur est le suivant :

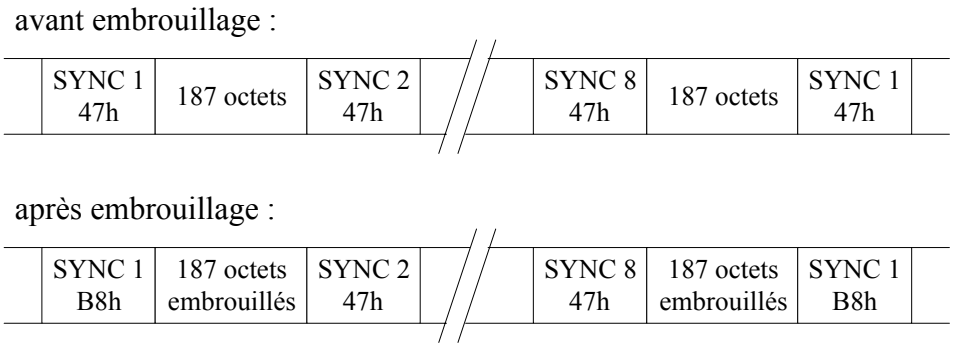


L'embrouillage (ou brassage) est effectué avant le codage de canal, l'opération inverse étant réalisée à la réception après décodage de canal afin de retrouver la séquence d'origine. Pour cela, Les générateurs A et B doivent être identiques et maintenus synchronisés.



Le chargement de la suite "100101010000000" dans le registre du générateur de séquence pseudo-aléatoire doit être effectué au début de chaque groupe de 8 paquets de transport. Afin de synchroniser le désembrouilleur, l'octet de synchro du premier paquet de chaque groupe est inversé (on transmet $\overline{\text{SYNC}}$ = B8 au lieu de SYNC = 47 hex). Le premier bit à la sortie du générateur de séquence pseudo-aléatoire est combiné avec le premier bit du premier octet suivant l'octet $\overline{\text{SYNC}}$.

Afin de faciliter la synchronisation, les 7 autres octets de synchronisation SYNC du groupe ne sont pas embrouillés: le générateur continue à fonctionner normalement alors que l'entrée VALIDATION est inactivée. La période du générateur de séquence pseudo-aléatoire est donc égale à 1503 octets soit (8x188)-1 octets. On obtient finalement l'organisation suivante :



6.3 Code Reed-Solomon

Les codes cycliques sont des codes en blocs linéaires particuliers ; parmi ceux-ci on trouve les codes de Golay, B.C.H. et les codes de Reed-Solomon. Comme tous les codes en blocs, chaque mot de code de longueur N symboles est composé de K symboles issus de la source d'information et de N-K symboles de contrôle. Dans le cas particulier des codes Reed-Solomon, les mots de code sont q aires où q est une puissance de 2 ($q = 2^m$).

Les paramètres des codes de Reed-Solomon sont les suivants :

- longueur des mots : $N = q-1 = 2^m-1$.
- nombre de bits par symbole : m.
- nombre de symboles de contrôle : $N-K = 2t$.
- distance minimale entre les mots du code : $2t+1$.

Les codes de Reed-Solomon peuvent corriger t symboles de m bits parmi les N symboles d'un mot de code. Ces codes sont donc bien adaptés à la correction de paquets d'erreurs à condition que la longueur du paquet d'erreurs ne soit pas supérieure à t symboles.

Le code de Reed-Solomon retenu par le groupe DVB est du type RS(204,188) qui comprend 16 octets de contrôle. Comme la longueur des mots codes doit être égale à 2^m-1, on réalise ce code à partir d'un code original RS(255,239). Pour cela, on ajoute 51 octets nuls aux 188 octets d'un paquet de transport afin d'obtenir 239 octets, puis on y applique le codage RS(255,239). Après codage, on supprime les 51 octets excédentaires pour obtenir un paquet de 204 octets comprenant 16 octets de contrôle. On effectue la même opération au décodage. Comme la longueur des mots est égale à 2^8-1, les symboles seront des octets. Ce code permet donc de corriger 8 octets erronés par paquet de transport et d'en détecter 16.

Les performances du code peuvent être facilement calculées dans le cas d'octets erronés répartis de manière aléatoire. En définissant Pe le taux d'erreurs octets en entrée du décodeur et Ps le taux d'erreurs octets après décodage, on a (t = 8) :

Ps = P(d'avoir t+1 octets erronés par mot de code)
+ P(d'avoir t+2 octets erronés par mot de code)
+ ...
+ P(d'avoir N octets erronés par mot de code)

Après développement on obtient:

$$P_s = \sum_{J=t+1}^{N} C_J^N . P_e^{\,J} . (1 - P_e)^{N-J} . \frac{J}{N}$$

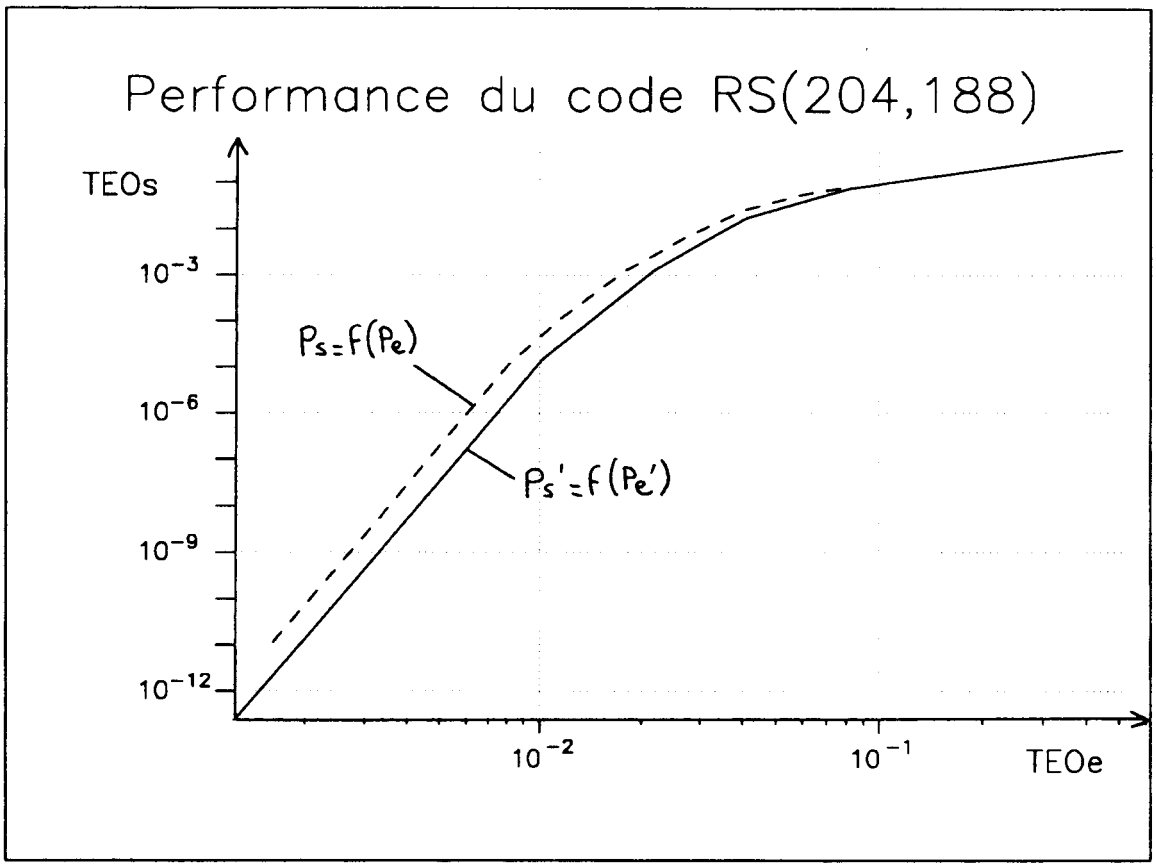
Le fait de réaliser un codeur Reed-Solomon RS(204,188) à partir d'un codeur RS(255,239) modifie sensiblement la relation ci-dessus. En effet en définissant respectivement Pe' et Ps' les taux d'erreurs octets en entrée et en sortie du décodeur RS(204,188) on peut facilement vérifier les relations :

$$P'_e = \frac{204+51}{204} . P_e \quad \text{et} \quad P'_s = \frac{188+51}{188} . P_s$$

La relation ci-dessus s'exprime alors :

$$P'_s = 1{,}27127.P_s = 1{,}27127.\sum_{J=t+1}^{N} C_J^N.(0{,}8.P'_e)^J.(1-0{,}8.P'_e)^{N-J}.\frac{J}{N}$$

Les relations $P_s = f(P_e)$ et $P'_s = f(P'_e)$ sont représentées sur la figure suivante :



A partir de ces caractéristiques, on peut vérifier que le code RS(204,188) pour un taux d'erreurs octets en entrée TEOe donne un taux d'erreurs octets en sortie TEOs plus faible que le code RS(255,239). Ceci s'explique car la redondance apportée par le code RS(204,188) est supérieure à celle apportée par le code RS(255,239).

Si le taux d'erreurs en entrée est supérieur à 4.10^{-2} le code RS(204,188) n'est pas efficace car le nombre d'erreurs par paquet de 204 octets est supérieur à 8 en moyenne : le décodeur de Reed-Solomon ne parvient plus alors à corriger les erreurs.

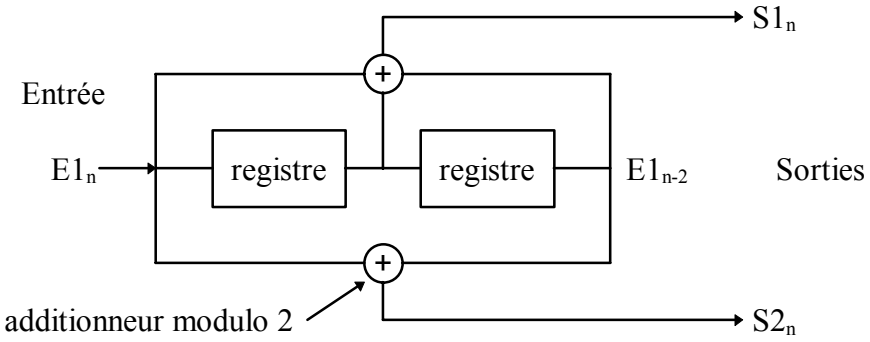
Le décodage des codes de Reed-Solomon utilise en général un algorithme découvert par Berlekamp.

6.4 Code convolutionnel

6.4.1 Codage

Pour un code en blocs linéaires, chaque mot de code C = $\{c_0, c_1, ..., c_{N-1}\}$ dépend d'un message unique M = $\{m_0, m_1, ..., m_{K-1}\}$. Les codes convolutionnels sont une extension des codes en blocs linéaires, chaque mot de code C dépendant de plusieurs messages. Pour un message de K bits présenté à l'entrée du codeur, N bits en sortent qui dépendent de ce message et des (m-1) messages précédents. Le rendement R du codeur est défini par le rapport K/N, m est appelé la longueur de contrainte.

Pour étudier le fonctionnement du codeur convolutionnel nous allons utiliser un exemple simple: R=1/2 et m=3. Le schéma de principe de ce codeur convolutionnel est le suivant :



L'opération effectuée par le codeur consiste à passer la séquence d'entrée E1n à travers un registre à décalage de longueur finie K(m-1). E1n est alors combinée avec les différentes sorties du registre à décalage pour obtenir les sorties S1n et S2n. La matrice génératrice G=[g1,g2] définit le codage et par conséquence le câblage des deux additionneurs modulo2. On a ici:

- g1=[111]
- g2=[101]

g1 et g2 sont appelés les polynômes générateurs.

Le fonctionnement d'un codeur convolutionnel peut être analysé par 3 diagrammes :

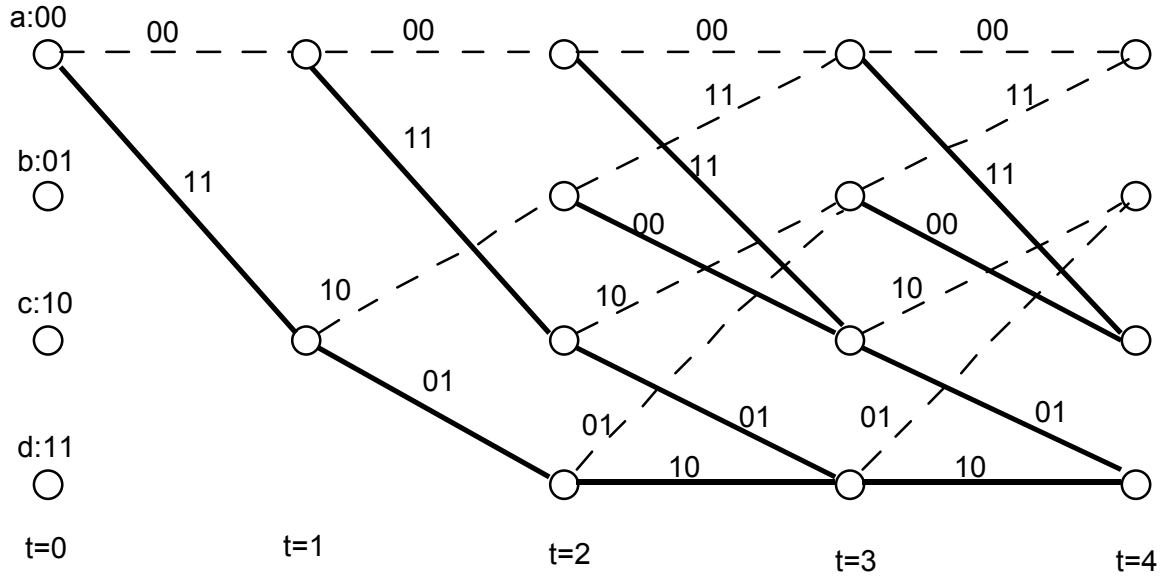
1. le diagramme en arbre
2. le diagramme d'état
3. le diagramme en treillis

Nous ne nous intéresserons ici qu'au diagramme en treillis.

Un codeur de longueur de contrainte m possède $2^{K(m-1)}$ états internes. Dans notre exemple K=1 et m=3, le codeur possède donc 4 états internes a, b, c et d:

état interne	$E1_{n-1}$	$E1_{n-2}$
a	0	0
b	0	1
c	1	0
d	1	1

Le diagramme en treillis de ce codeur est le suivant :



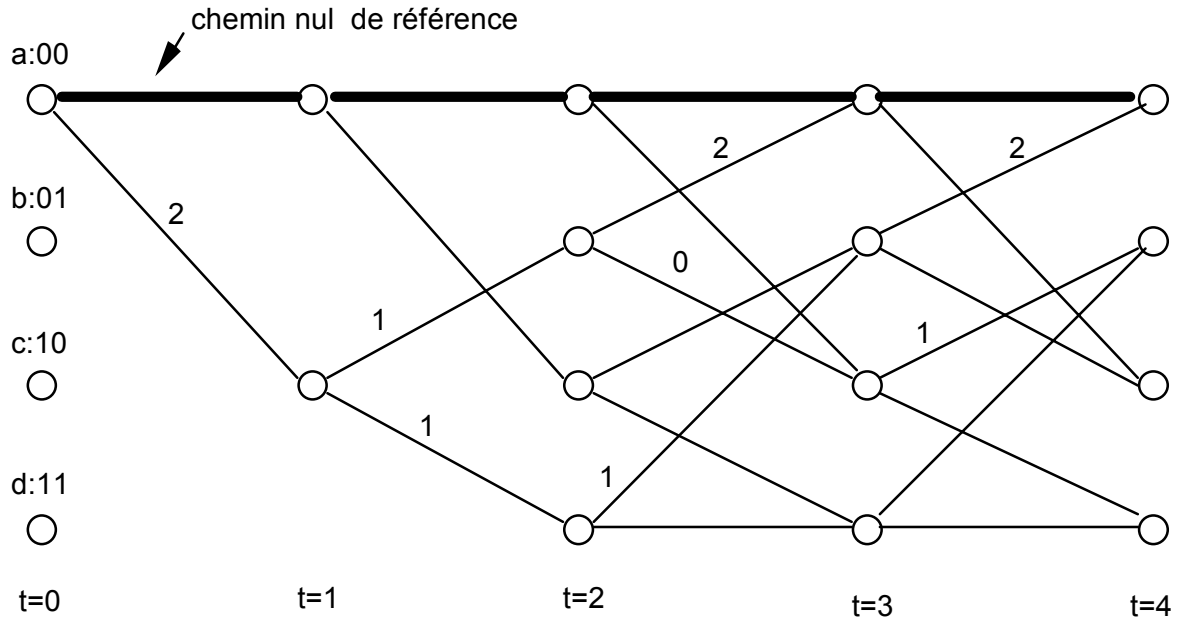
Sur chaque branche est noté l'état des deux sorties S1n et S2n.

Dans ce diagramme en treillis, les traits gras correspondent à un état 1 sur l'entrée E1n alors que les traits pointillés correspondent à un état 0. On peut constater sur ce diagramme

qu'après une période transitoire, le treillis contient 4 nœuds à chaque étape qui correspondent aux 4 états internes du codeur. A chaque nœud arrivent et partent deux chemins (à partir de t=3) : le treillis se répète alors à chaque étape.

Pour un code convolutionnel, la distance entre les différents chemins du treillis a un rôle important sur le pouvoir de correction du code. La distance minimale entre deux chemins qui divergent puis convergent est appelée distance libre du codeur. Pour la déterminer, on peut utiliser le diagramme en treillis.

On choisit comme chemin de référence le chemin qui correspond à l'émission d'une suite de zéros en entrée du codeur (chemin en gras dans le diagramme en treillis ci-dessous).

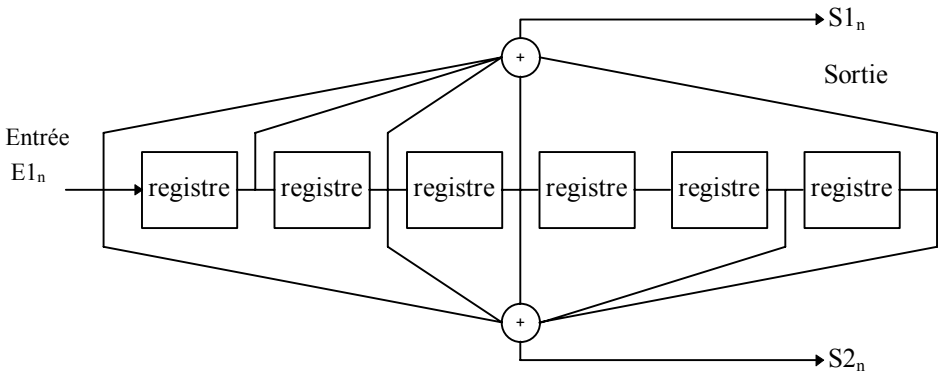


De ce diagramme on constate qu'un seul chemin (a,c,b,a) est à la distance 5 du chemin de référence. Il est à noter qu'il existe deux chemins (a,c,d,b,a) et (a,c,b,c,b,a) à la distance 6 du chemin de référence : dans notre cas la distance libre du codeur est donc égale à 5.
Le code retenu par le groupe DVB est un code de rendement 1/2 et de longueur de contrainte m=7. Les polynômes générateurs g1 et g2 permettent d'obtenir la meilleure distance libre pour ce code soit 10 bits :

- $g1=[1111001]_{bin}=[171]_{oct}$

- $g2=[1011011]_{bin}=[133]_{oct}$

Le schéma de principe du codeur est le suivant :



6.4.2 Décodage des codes convolutionnels

Il existe plusieurs algorithmes de décodage :

- l'algorithme de Viterbi

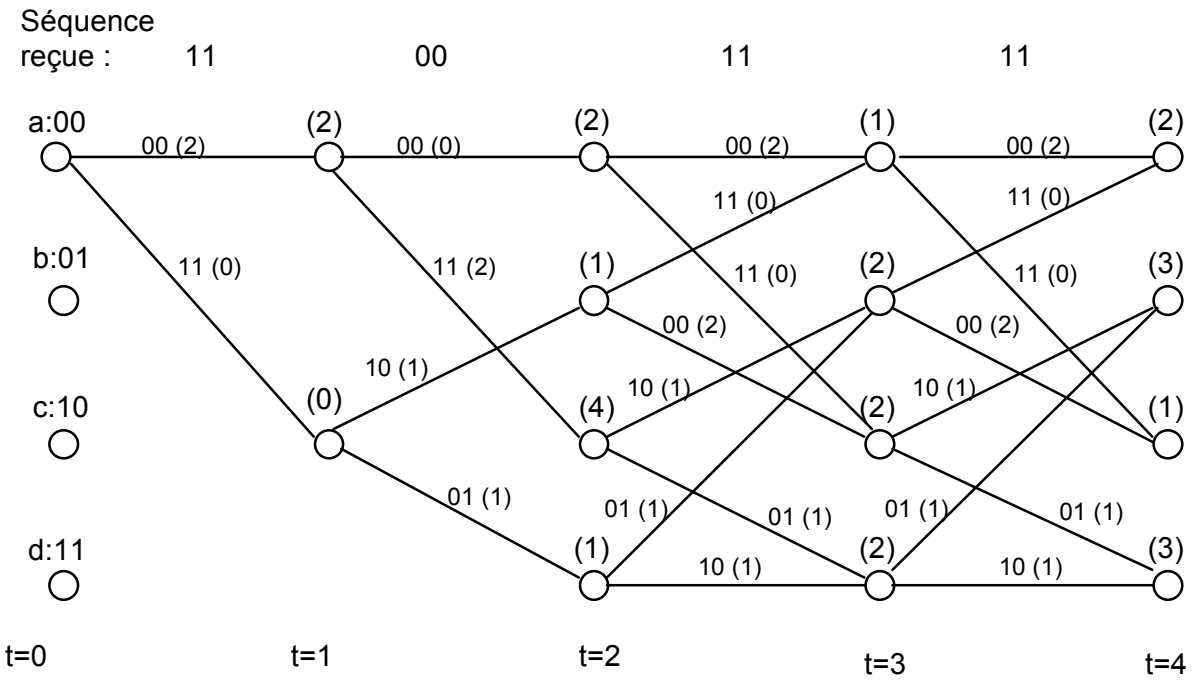
- l'algorithme à décodage réfléchi

- l'algorithme à décodage séquentiel de Fano

En pratique, on utilise l'algorithme de Viterbi pour les codes à longueur de contrainte inférieure ou égale à 10. Pour les autres codes, la complexité du décodeur étant trop importante, on utilise l'algorithme à décodage séquentiel. Nous ne traiterons ici que l'algorithme de Viterbi.

Cet algorithme applique la règle de décision du maximum de vraisemblance : règle qui consiste à choisir comme séquence décodée celle qui, parmi les séquences pouvant avoir été émises par le codeur, est à la distance (de Hamming ou Euclidienne) la plus faible de la séquence reçue. L'algorithme de Viterbi utilise le diagramme en treillis pour déterminer cette séquence.

Pour comprendre l'algorithme de Viterbi, nous allons reprendre l'exemple utilisé au §1.1.3.1. Le décodage convolutionnel est ici à décision dure car on utilise la distance de Hamming pour déterminer la séquence la plus probable. Supposons qu'à l'instant t = 0, le codeur soit dans l'état a et que la séquence en entrée du codeur soit 1001. La séquence en sortie est alors 11101111. On considère que la séquence reçue est 11001111 (soit une erreur sur le 3ième bit). Le diagramme en treillis associé est le suivant:



A chaque étape, l'algorithme de Viterbi calcule pour chacune des branches la distance de Hamming ou métrique entre le couple de bits reçu et le couple de bits associé à la branche considérée.

Ensuite pour chaque nœud, on ne conserve qu'un seul chemin baptisé survivant, le survivant étant le chemin qui est à la distance minimale de la séquence reçue.

Exemple :

A l'instant t=3, deux chemins convergent vers l'état a :

- le chemin (a,c,b,a) à la distance 0+1+0=1 de la séquence reçue.

- le chemin (a,a,a,a) à la distance 2+0+2=4 de la séquence reçue.

  Le survivant en ce nœud sera donc le chemin (a,c,b,a).

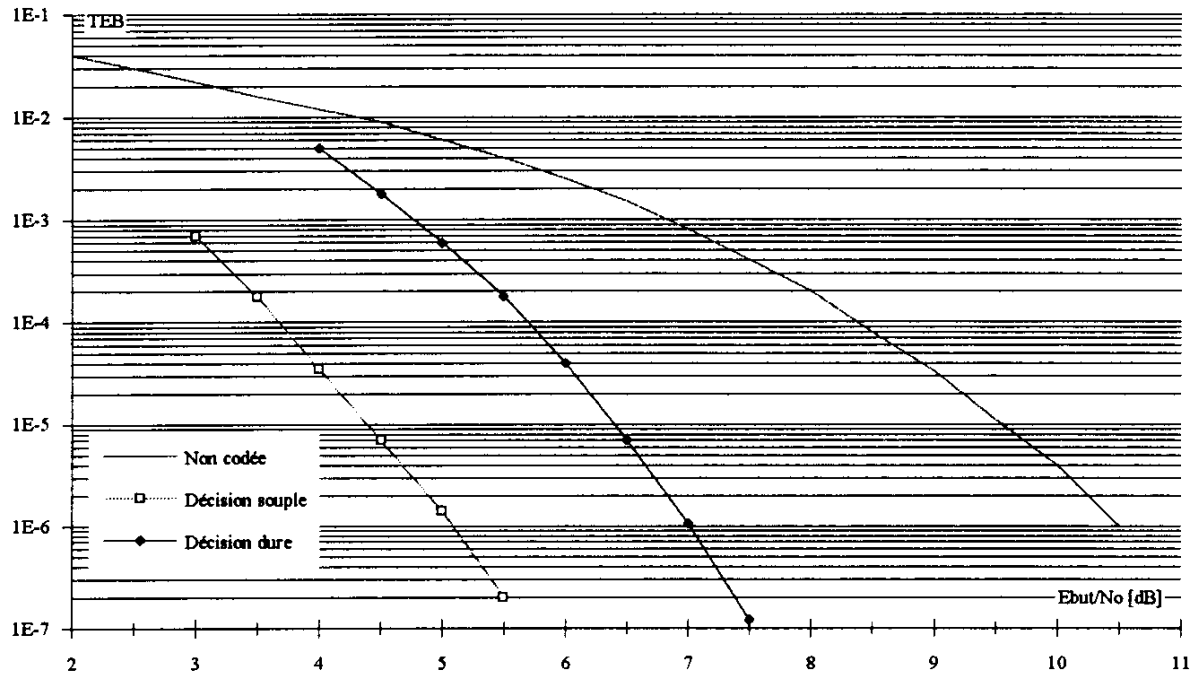
  A l'instant t=4, il reste 4 chemins survivants qui convergent respectivement : vers l'état a avec une distance de 2, vers l'état b avec une distance de 3, vers l'état c avec une distance de 1 et enfin vers l'état d avec une distance de 3.

  La séquence la plus vraisemblablement émise par le codeur est donc celle qui correspond au chemin survivant qui converge vers l'état c à t=4 c'est à dire le chemin (a,c,b,a,c).

  Ce chemin correspond à l'émission par le codeur de la séquence 11101111 soit la séquence 1001 à l'entrée du codeur : le décodeur de Viterbi a corrigé l'erreur survenue dans la transmission.

L'algorithme peut être utilisé aussi dans le cas d'un canal analogique: le décodeur travaille alors en décision souple ou douce. L'algorithme est le même que celui présenté précédemment ; la seule différence concerne le calcul de métrique qui utilise la distance euclidienne au lieu de la distance de Hamming. Le décodage en décision souple apporte en théorie une amélioration des performances d'environ 2 dB par rapport au décodage en décision dure. En pratique, on effectue une quantification sur 2^n niveaux avec n=3 ou 4.

Le calcul des performances théoriques des codes convolutionnels avec décodage à décision dure ou souple est difficile. Ce calcul utilise en effet la fonction de transfert du code convolutionnel qui est déterminée à partir du diagramme en treillis et de la distance libre du code. Nous nous contenterons de donner la courbe théorique Pe = f(Ebut/No).



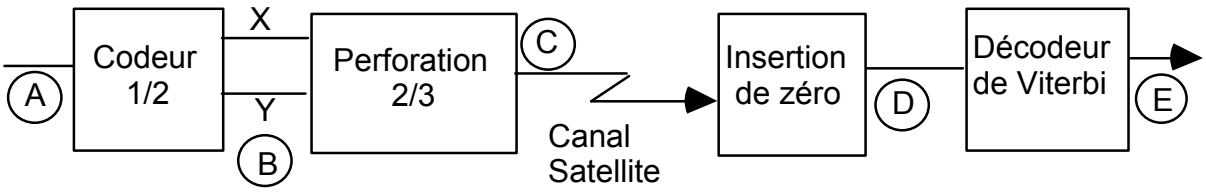
Pour un code convolutionnel K/N et de longueur de contrainte m, on a 2^{m-1} états internes : l'utilisation de l'algorithme de Viterbi nécessite de conserver à chaque étape $2^{K(m-1)}$ survivants et métriques associées. A chaque étape, 2^K chemins convergent vers chaque nœud ; pour ces 2^K chemins, un calcul de métrique est effectué afin de déterminer le survivant. En conséquence le nombre de calculs de métrique $2^{K(m-1)}2^K$ à chaque étape croit exponentiellement avec K et m ce qui limite l'utilisation de l'algorithme de Viterbi à de faibles valeurs de K et m.

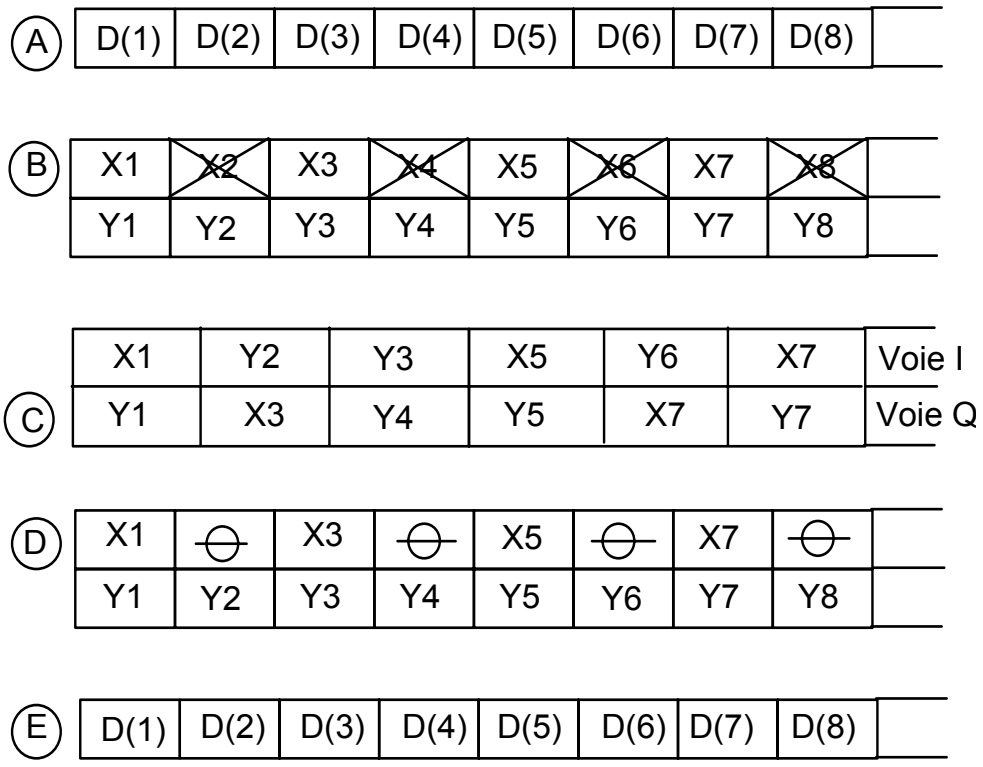
En pratique, on ne peut pas attendre que l'ensemble des symboles émis par le codeur soit reçu pour commencer le décodage : en fait on montre que l'on peut décoder un symbole émis à l'instant t = n après avoir parcouru le treillis jusqu'à t = n+x avec x longueur de troncature égale à au moins 5 fois la longueur de contrainte.

6.4.3 Perforation

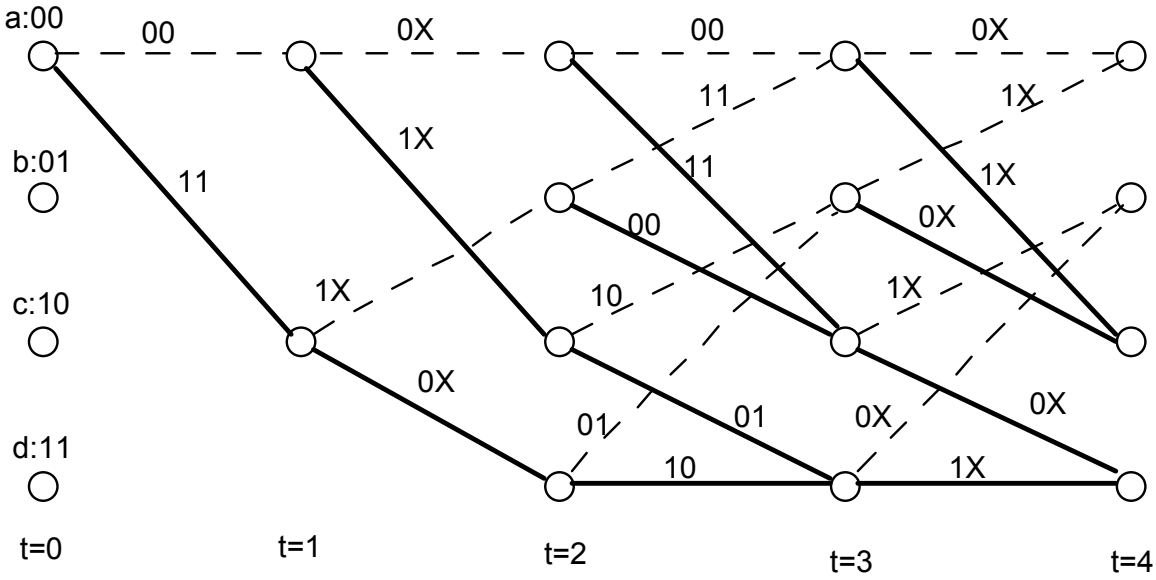
Il est possible à partir d'un code convolutionnel 1/2 de réaliser des codes convolutionnels de rendement $R = \frac{n-1}{n}$ en supprimant certains symboles en sortie du codeur. Cette technique permet d'augmenter le rendement du code convolutionnel sans augmenter la complexité du

décodeur, la robustesse du code étant évidemment diminuée. Prenons par exemple une perforation 2/3 :





La perforation s'obtient en supprimant un symbole sur 4 en sortie du codeur 1/2. A la réception on rajoute à la place de chaque symbole manquant un symbole nul ($\varnothing$). Le diagramme en treillis de ce codeur perforé de rendement 2/3 et de longueur de contrainte 3 est représenté sur la figure suivante :



La distance libre du codeur convolutionnel 1/2 était égal à 5 ; elle est réduite à 3 dans le cas du codeur perforé de rendement 2/3. Il faut cependant noter qu'il n'existe pas de code de rendement 2/3 et de longueur de contrainte 3 ayant une distance libre supérieure à 3.

La recommandation du groupe DVB définit la façon de perforer le code convolutionnel 1/2 afin d'obtenir les rendements 2/3, 3/4, 5/6 et 7/8.

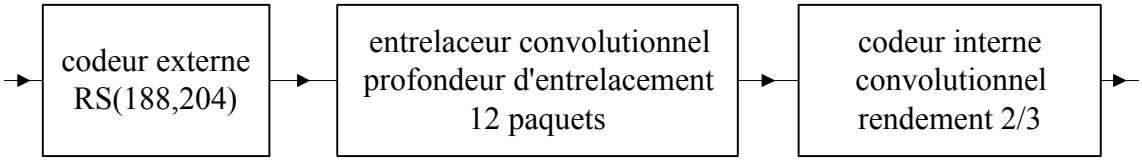
2/3		3/4		5/6		7/8	
P	D Libre	P	D Libre	P	D Libre	P	D Libre
X:10 Y:11 $I=X_1Y_2Y_3$ $Q=Y_1X_3Y_4$	6	X:101 Y:110 $I=X_1Y_2$ $Q=Y_1X_3$	5	X:10101 Y:11010 $I=X_1Y_2Y_4$ $I=Y_1X_3X_5$	4	X:1000101 Y:1111010 $I=X_1Y_2Y_4Y_6$ $I=Y_1Y_3X_5X_7$	3

Ces codes convolutionnels perforés ont été découverts par Yasuda et permettent d'obtenir la distance libre maximale.

6.5 Concaténation des codes

6.5.1 Principe

La concaténation de codes permet d'améliorer les performances du codage de canal. Elle repose sur la mise en cascade de deux codes aux propriétés complémentaires, séparés par un entrelaceur.

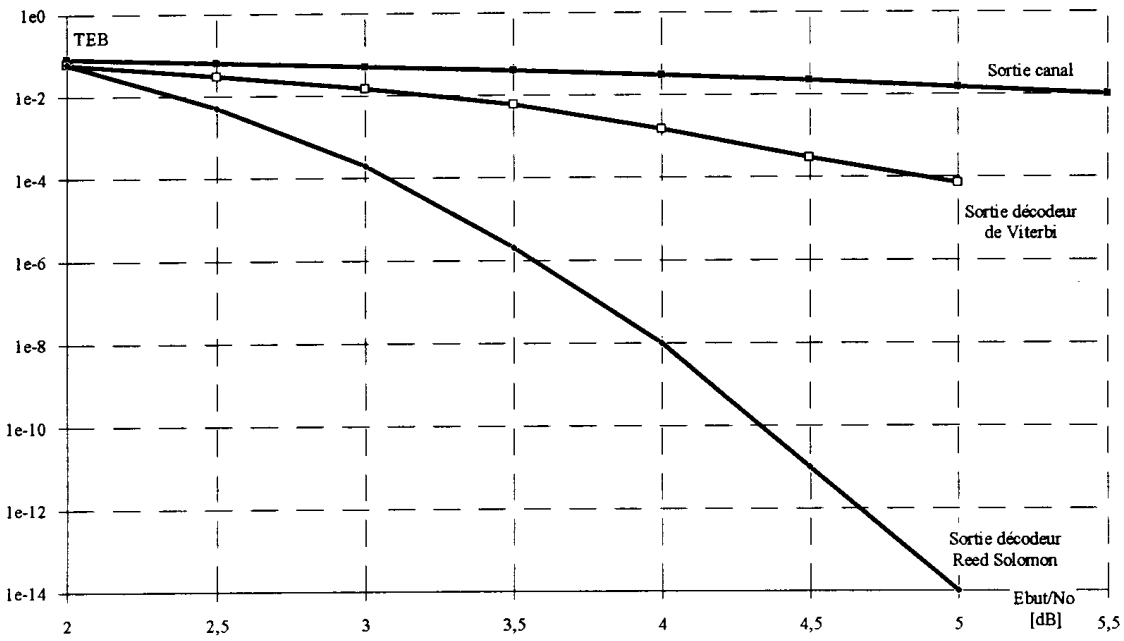


On peut rappeler les caractéristiques des codes utilisés :

- code convolutionnel.
  - bon fonctionnement avec un canal fortement bruité.
  - le bruit en entrée du décodeur doit être blanc et gaussien pour obtenir de bonnes performances.
  - les erreurs sont groupées en paquets à la sortie du décodeur.

- code Reed-Solomon (188,204)
  - ce code est très efficace si le taux d'erreurs est inférieur à 4.10^{-2}.
  - bonne correction des erreurs en paquet.
  - les erreurs en entrée du décodeur peuvent être isolée ou groupées en courts paquets.

Comme le canal satellite est fortement bruité par un bruit blanc gaussien, le code convolutionnel est généralement utilisé dans le codeur interne. L'objectif étant d'obtenir un taux d'erreur d'environ 10^{-11} en sortie du décodeur, seul un code en blocs puissant comme le code de Reed-Solomon peut servir comme code externe. On peut ainsi obtenir les courbes de taux d'erreurs binaires suivantes en différents points de la chaîne de transmission :



6.5.2 Entrelacement

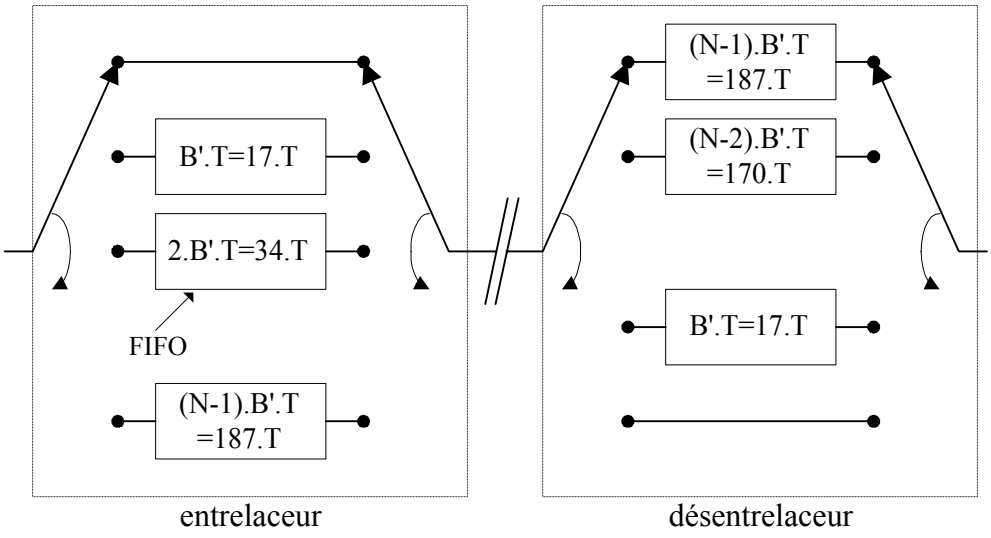
Afin d'améliorer les performances des codes concaténés, on intercale entre le codeur externe et le codeur interne un étage entrelaceur. A la réception, un étage désentrelaceur est placé entre le décodeur interne et le décodeur externe pour réaliser l'opération inverse.

L'entrelacement permet de décorréler les erreurs en sortie du décodeur interne (décodeur de Viterbi). En effet, comme nous l'avons vu précédemment, les erreurs en sortie du décodeur de Viterbi sont groupées en paquets qui peuvent être d'une longueur supérieure au nombre de symboles que peut corriger le décodeur Reed-Solomon. L'entrelacement disperse donc les erreurs qui peuvent alors être considérées comme isolées.

Le groupe DVB a choisi d'utiliser un entrelaceur convolutionnel (BxN) avec les caractéristiques :

B=204, N=12 et B'=17

Le schéma de principe de l'ensemble entrelaceur-désentrelaceur est le suivant :



T est égal à la durée de transmission de N=12 symboles. Le temps de propagation à l'intérieur d'une branche est donc un multiple de la durée de transmission d'un paquet de transport. Par conséquent, si la transmission provoque deux octets erronés consécutifs, ils se retrouvent dans deux paquets de transport différents après désentrelacement.

La profondeur d'entrelacement est égale à 12 paquets puisque les octets se trouvant dans un paquet pendant la transmission sont répartis dans 12 paquets successifs après désentrelacement. Tant que la longueur du bloc d'octets erronés consécutifs est inférieure ou égale à 12x8=96, le nombre d'octets erronés dans un paquet de transport après désentrelacement ne dépassera pas 8.

Les octets de synchro et l'octet de synchro inversé sont toujours routés dans la branche 0 (retard nul) de l'entrelaceur. La synchronisation du désentrelaceur est donc simplement effectuée en routant le premier octet de synchro détecté dans sa branche 0.

7 Notions de filtrage

7.1 Premier critère de Nyquist

La largeur de bande du canal de transmission est limitée. Il est donc nécessaire de filtrer le signal à transmettre s(n.T) afin d'en limiter l'occupation spectrale. La question essentielle du filtrage est donc la suivante : quel filtre doit-on utiliser pour pouvoir reconstituer parfaitement les échantillons émis à la réception, c'est-à-dire pour ne pas avoir d'interférences inter-symboles (ISI) ?

Le premier critère de Nyquist impose, pour ne pas avoir d'interférences inter-symboles, de filtrer le signal s(n.T) avec un filtre passe bas de fréquence de coupure $\frac{1}{2T}$, T étant la période symbole du signal considéré. De cette manière, la réponse impulsionnelle du filtre vaut 1 sur l'échantillon filtré et s'annule sur tous les autres échantillons (voir Fig.2.1). Vous noterez que le critère de Nyquist s'applique à un signal échantillonné qui n'est pas nécessairement quantifié : il reste valable si l'amplitude des échantillons est analogique. La conséquence de ce critère est très importante :

La bande minimale occupée par un signal échantillonné ne dépend que de la fréquence symbole. Elle ne dépend en aucun cas du nombre d'états que peut prendre un symbole et n'est donc pas directement reliée au débit binaire.

Si on prend un signal en bande de base de fréquence symbole 1 Msymb/s, il faudra une bande passante théorique minimale de 500 kHz pour pouvoir le transmettre. Avec un bit par symbole (2 états par échantillon), cela donnera un débit binaire de 1 Mbit/s, avec 6 bits par symbole (64 états par échantillon), on aura 6 Mbit/s. L'augmentation du nombre d'états par symbole n'influe pas sur la bande passante, mais elle joue sur le TEB, le signal étant d'autant plus sensible au bruit que le nombre d'états est élevé.

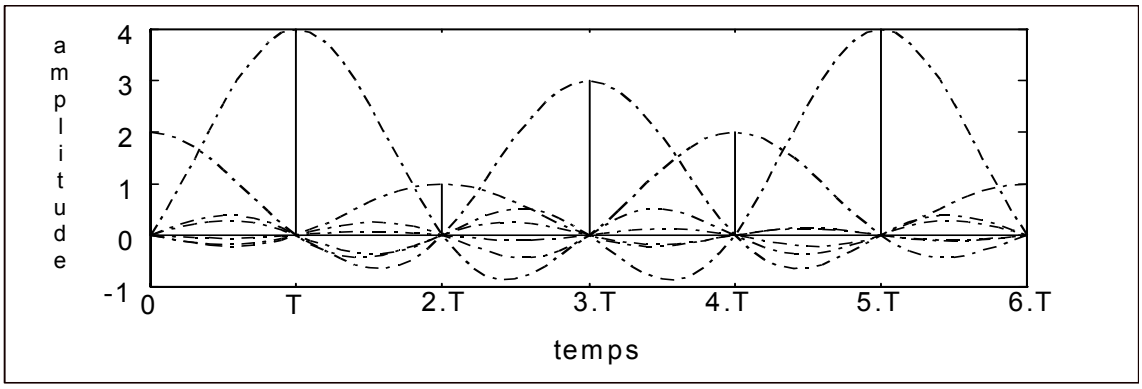


Figure 7-1 : s(n.T) convolué avec la réponse impulsionnelle du filtre de Nyquist

Le signal s'(n.T) après filtrage est égal à la somme des réponses impulsionnelles convoluées avec les échantillons de s(n.T) (voir : Fig.6-2).

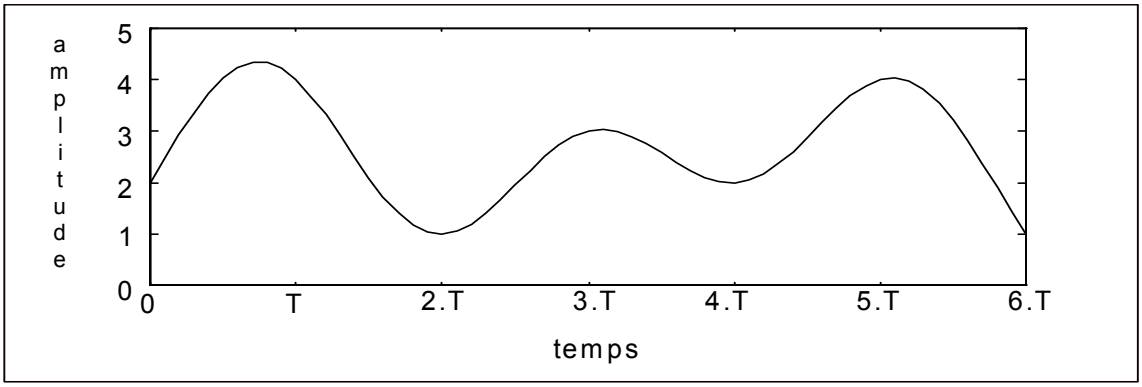


Figure 7-2 : s(n.T) après filtrage

A la réception, si on sait placer le point d'échantillonnage au bon endroit, on peut reconstituer le signal d'origine (voir : Fig.6-3).

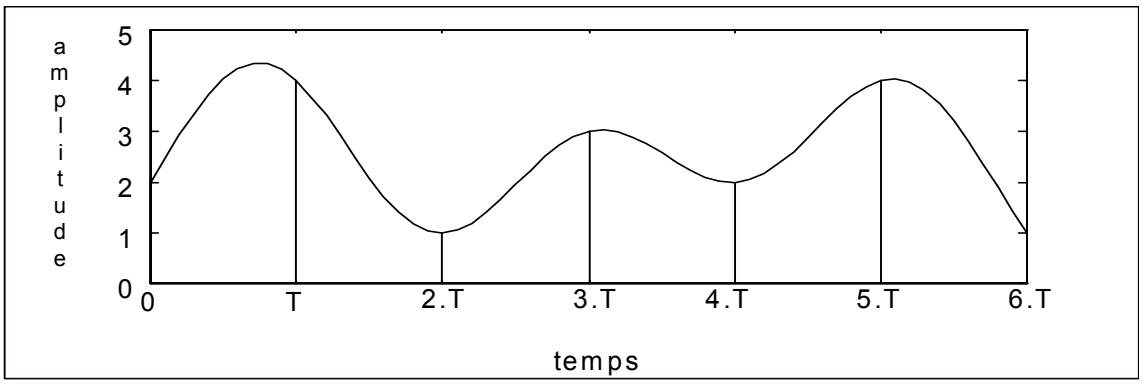


Figure 7-3 : reconstruction parfaite de s(n.T) à la réception

Il faut pour cela récupérer la fréquence rythme puis trouver le point optimal de ré-échantillonnage (voir : boucle de Costas).

7.2 Deuxième critère de Nyquist

Le deuxième critère de Nyquist indique qu'il existe une famille de filtres qui respecte le premier critère : les filtres en cosinus surélevé (Fig.2.4) dont l'équation est :

$$H(f) = \begin{cases} 1 & \text{si} \quad 0 < f < (1-\alpha).f_n \\ \cos^2\left\{\dfrac{\pi}{4\alpha}\left(\dfrac{f}{f_n} - \left[1-\alpha\right]\right)\right\} & \text{si} \quad (1-\alpha).f_n < f < (1+\alpha).f_n \\ 0 & \text{si} \quad f > (1+\alpha).f_n \end{cases}$$

$$\text{avec } f_n = \frac{1}{2T}$$

α, le facteur d'arrondi (« roll-off ») peut être compris entre 0 et 1. On fixe généralement cette valeur entre 0,35 et 0,15.

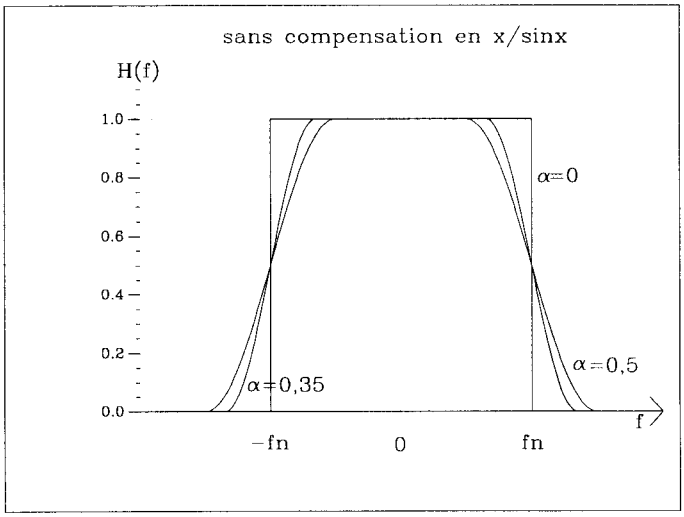


Figure 7-4 : réponse en fréquence du filtre en cosinus surélevé

7.3 Diagramme de l'œil

Le diagramme de l'œil est une mesure essentielle en transmission numérique. Elle permet de voir les interférences inter-symboles et de repérer le point optimal d'échantillonnage. Dans ce diagramme, on superpose les transitions $0 \rightarrow 1 \rightarrow 0$ et $1 \rightarrow 0 \rightarrow 1$. La figure 7-5 donne un

exemple d'œil sans ISI. La ligne en pointillés matérialise le point idéal d'échantillonnage. Toutes les transitions se rejoignant en deux points d'amplitude 0 et 1, cela signifie bien que l'on retrouve exactement le signal d'origine.

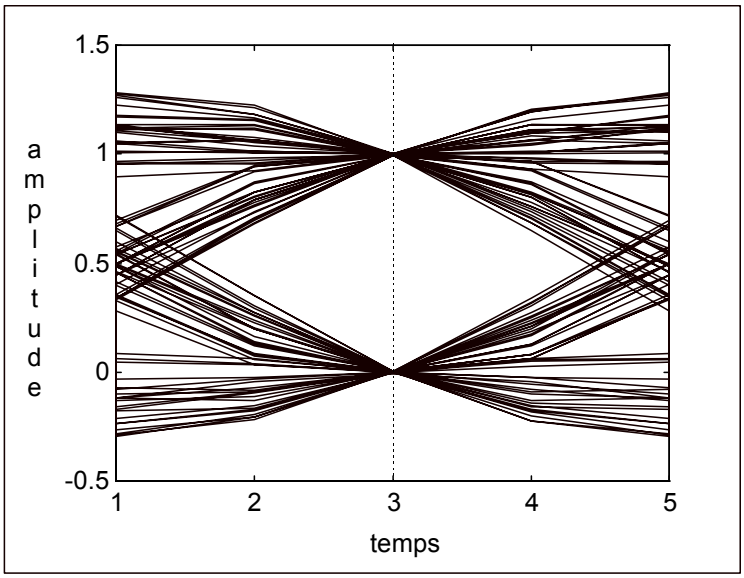


Figure 7-5 : diagramme de l'œil sans ISI

Par contre, sur la figure 7-6, on voit qu'il y a ISI puisqu'il n'existe aucun endroit où les transitions se rejoignent pour ne former qu'un point. Par rapport à l'œil précédent, le taux d'erreur sera plus élevé en présence de bruit.

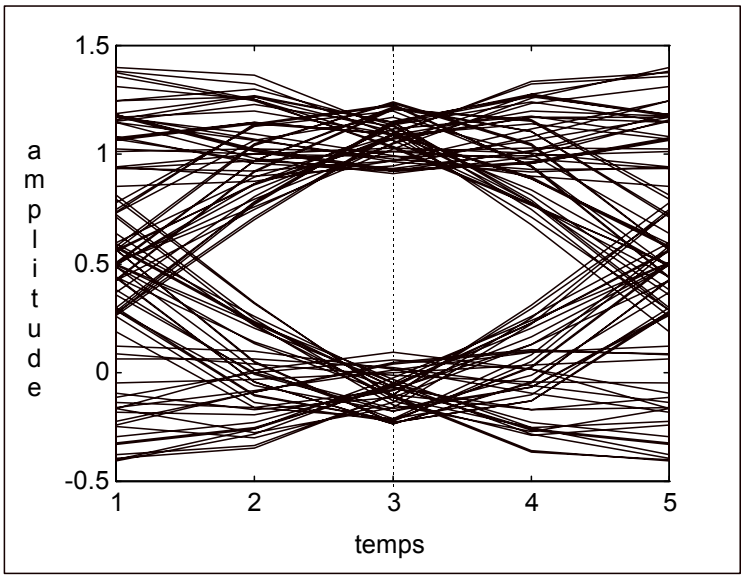


Figure 7-6 : diagramme de l'œil avec ISI

7.4 Blanchisseur de spectre

Le filtre en cosinus surélevé n'entraîne pas d'interférences inter-symboles si le signal est sous la forme d'une suite d'échantillons. Par contre, il ne fonctionne pas dans le cas d'un signal NRZ car ce signal a un spectre en sin(x)/x au lieu d'avoir un spectre blanc. Pour obtenir dans les deux cas le même signal après filtrage, il est nécessaire d'ajouter au filtre de Nyquist une compensation en x/sin(x) (cela rend le spectre du signal NRZ blanc dans la bande passante). On obtient alors l'équation :

$$H(f) = \begin{cases} \dfrac{\pi f T}{\sin(\pi f T)} & \text{si} \quad 0 < f < (1-\alpha).f_n \\ \dfrac{\pi f T}{\sin(\pi f T)} . \cos^2\left\{ \dfrac{\pi}{4\alpha}\left(\dfrac{f}{f_n} - [1-\alpha] \right) \right\} & \text{si} \quad (1-\alpha).f_n < f < (1+\alpha).f_n \\ 0 & \text{si} \quad f > (1+\alpha).f_n \end{cases}$$

Ce qui nous donne la réponse en fréquence suivante :

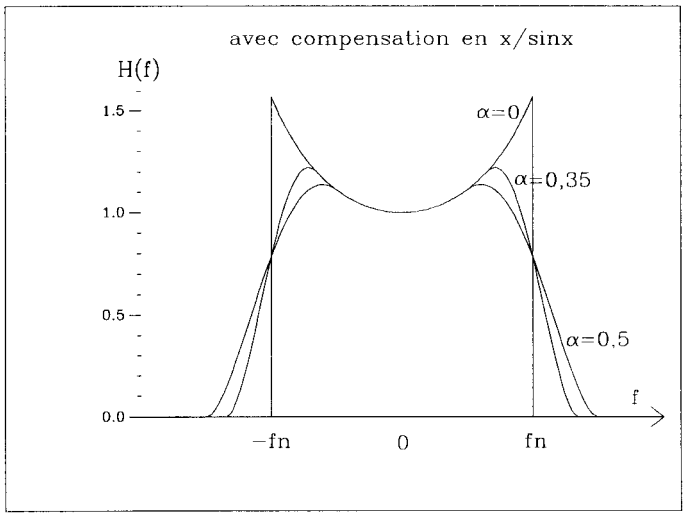


Figure 7-7 : réponse en fréquence du filtre en cosinus surélevé avec compensation

En fait, il faut limiter l'occupation du signal à l'émission car la bande passante du canal de transmission est coûteuse et filtrer au plus étroit le signal à la réception afin de limiter le bruit entrant dans le décodeur et donc minimiser le taux d'erreur. Le filtre en cosinus surélevé est donc séparé en deux filtres : un filtre en racine de cosinus surélevé à l'émission et un filtre en

racine de cosinus surélevé à la réception. La compensation en x/sin(x) n'est appliquée qu'à l'émission.

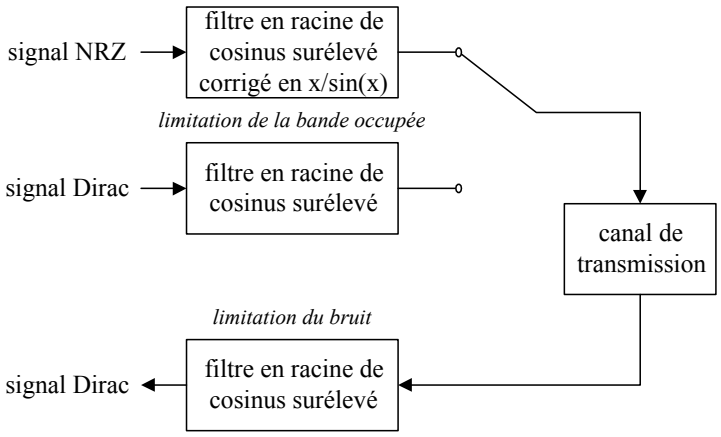


Figure 7-8 : chaîne de transmission

7.5 Sur-échantillonnage et interpolation

Avec un roll-off supérieur à 0, la largeur du filtre de Nyquist dépasse la demi-fréquence d'échantillonnage. Il est donc nécessaire de sur-échantillonner le signal pour respecter Shannon. En effet, le sur-échantillonnage permet d'augmenter la fréquence d'échantillonnage tout en conservant le spectre d'origine du signal. On peut le voir sur la figure 7-9 avec un signal s(n.T) égal à une suite de diracs.

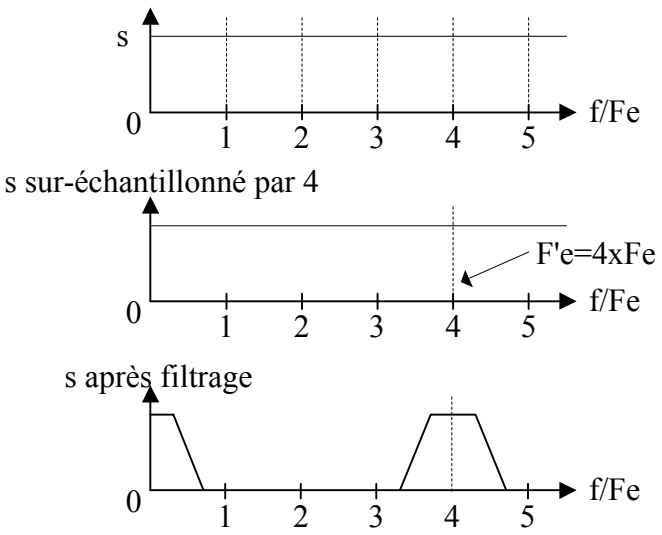


Figure 7-9 : effet fréquentiel du sur-échantillonnage

Voyons sur un exemple (Fig.6-10) comment on réalise un sur-échantillonnage facteur 4. On insère 3 zéros entre chaque échantillon (1), on multiplie l'amplitude des échantillons par 4 (pour garder la même puissance)(2) puis on filtre (3).

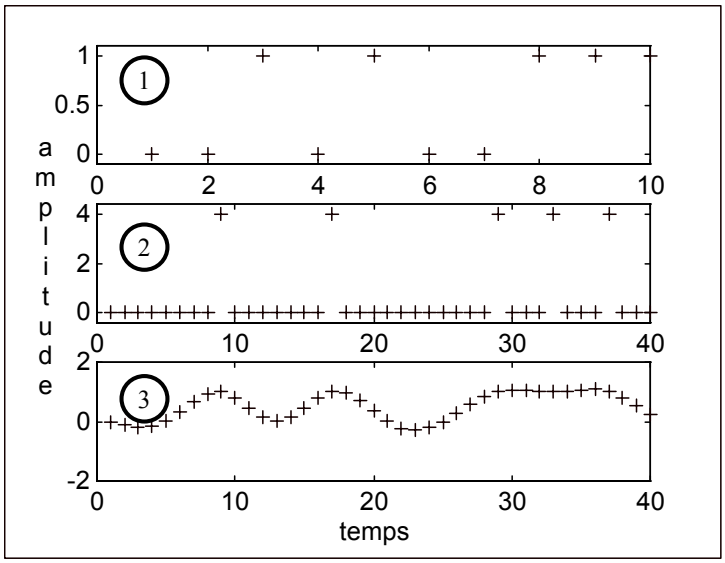


Figure 7-10 : réalisation d'un sur-échantillonnage avec un facteur 4

Le filtrage du signal par un filtre de Nyquist nécessite au moins un sur-échantillonnage facteur 2. Cependant, ce facteur est aussi égal au nombre d'échantillons composant le diagramme de l'œil. Afin que ce dernier soit lisible, on utilisera un facteur supérieur ou égal à 4. Tel est le cas pour les figures 6-5 et 6-6, mais les échantillons sont reliés entre eux par des traits afin de rendre la lecture plus commode. Si on ne relie pas les points entre eux, on obtient le diagramme suivant :

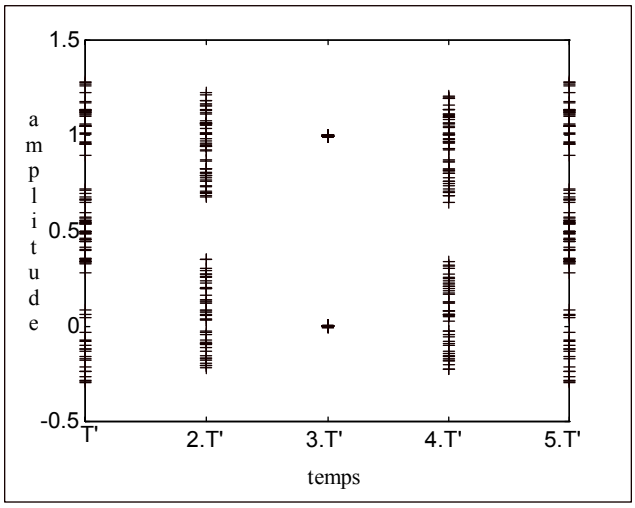
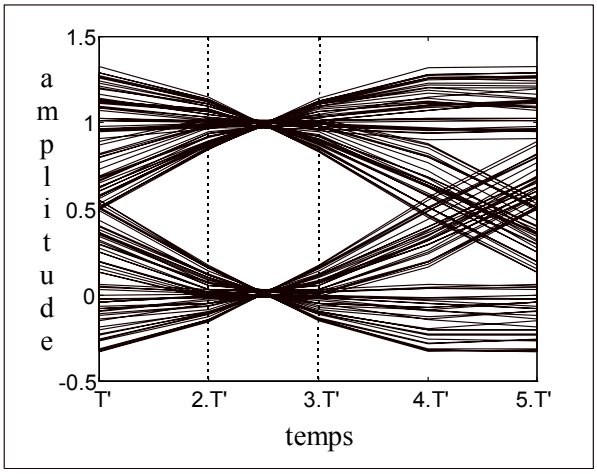


Figure 7-11 : diagramme de l'œil en mode échantillons

T' est égal à 1/F'e, F'e étant la nouvelle fréquence d'échantillonnage. Le point 5.T' est identique au point T'. Il apparaît dans le diagramme pour une simple question de symétrie. On retrouve bien 4 points dans le diagramme de l'œil avec F'e =4.Fe.

Il reste encore un problème à traiter. En effet, rien ne garantit que le point optimal d'échantillonnage du signal reçu se trouve sur un échantillon. En fait, dans le cas général, ce point ce trouve entre deux échantillons (Fig.6-12).



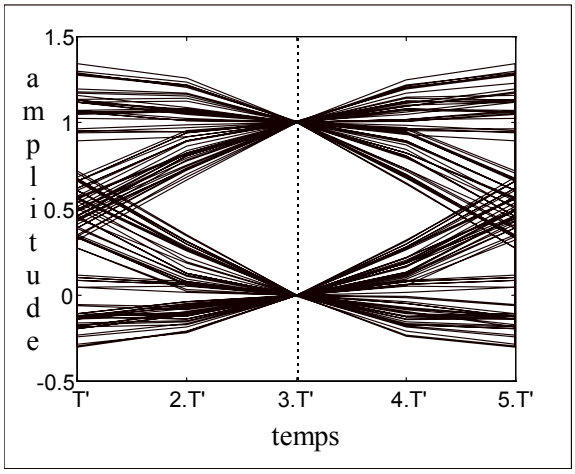


Figure 7-12 : point optimal entre deux échantillons

Figure 7-13 : point optimal après décalage

Il faut donc prévoir un dispositif permettant d'interpoler le signal puis de le décaler dans le temps d'une durée inférieure à un échantillon le point d'échantillonnage (Fig.6-13). C'est le diagramme de l'œil qui permet de calculer cette durée.

7.6 Mesures du TEB

L'expression du TEB en fonction du S/N et de Eb/No n'est pas compliquée à condition que les définitions soient précises. Si tel n'est pas le cas, on calcule à peu près n'importe quoi. Il y a deux aspects du problème qui doivent être précisé :

1. Les puissances de bruit et de signal sont généralement calculées dans une résistance de 1 Ω à l'entrée du décodeur. La courbe de TEB ne dépend que de l'écart-type du bruit et de la distance Δ entre deux points de la modulation. Elle est donc indépendante de la valeur moyenne du signal V_{moy}. Or, quand on calcule S la puissance moyenne du signal, on l'exprime à partir de Δ mais aussi de V_{moy}. Le TEB=f(S/N) dépend donc aussi de la valeur

moyenne du signal, ce qui peut paraître surprenant quand on connaît le principe physique mis en jeu. C'est uniquement la calcul de S qui établit cette dépendance. En fait, dans le domaine des transmissions, on travaille surtout avec des signaux à valeur moyenne nulle.

2. La courbe de TEB=f(S/N) est très dépendante des conventions utilisées. Elle rend impossible la comparaison entre différents types de modulation si on ne précise pas explicitement les conditions de mesures (ou de simulation). Pour rendre possible ces comparaisons, on définit deux nouvelles grandeurs :

- L'énergie par bit Eb. Elle est égale à S multipliée par la durée d'un bit T_b.

$$Eb = C.Tb = \frac{C}{fb} \quad \text{avec} \quad f_b = \frac{1}{Tb} = \text{fréquence bit}$$

- La densité spectrale de puissance monolatérale dans un Hz de bande après filtrage idéal de Nyquist.

$$No = \frac{N}{Bw} \text{ avec Bw = bande équivalente de bruit unilatérale.}$$

Les courbes de TEB sont généralement exprimées en fonction de Eb/No (en dB) afin de s'affranchir de la fréquence rythme et de la largeur de la bande de bruit. Ces courbes sont aussi dépendantes de la valeur moyenne du signal puisqu'elles sont calculées à partir de S.

Nous pouvons maintenant définir les grandeurs utilisées dans les courbes de TEB :

- Rapport signal sur bruit C/N.
  C est la puissance du signal avant le filtre de réception, N est la puissance de bruit avant le filtre de réception.

- Eb/No. D'après les définitions précédentes, on obtient :

$$\frac{Eb}{No} = \frac{C}{N}.\frac{Bw}{fb}$$

Avec par exemple une modulation MDP4, on a une efficacité spectrale de 2 bit/s/Hz. La largeur de bande minimale du filtre passe-bas de réception est égale à fb/2. On a alors :

$$\frac{Eb}{No} = \frac{C}{N}.\frac{1}{2}$$

- Ebu/No. Afin de comparer les performances des codes correcteurs d'erreurs ayant des rendements différents, on définit l'énergie par bit utile transmis. Pour un code de rendement R, on a :

$$Ebu = C.Tbu = \frac{Eb}{R}$$

Ce qui nous donne finalement :

$$\frac{Ebu}{No} = \frac{1}{R}.\frac{Eb}{No}$$

Avec par exemple un code RS de rendement $\frac{188}{204}$ et un code convolutionnel de rendement $\frac{2}{3}$, on trouve (en dB) :

$$\frac{Ebu}{No} = \frac{Eb}{No} + 10.\log(\frac{3}{2}) + 10.\log(\frac{204}{188}) = \frac{Eb}{No} + 2{,}11\,\text{dB}$$

Comme on a, en dB :

$$\frac{Eb}{No} = \frac{C}{N} - 3\,dB$$

On obtient finalement :

$$\frac{Ebu}{No} = \frac{C}{N} - 0{,}9\ dB$$

8 La chaîne de diffusion par satellite

8.1 Présentation

Une chaîne de diffusion par satellite est composée :

- d'un émetteur de forte puissance,
- d'un lien montant,
- d'un répéteur satellite,
- d'un lien descendant,
- d'un récepteur.

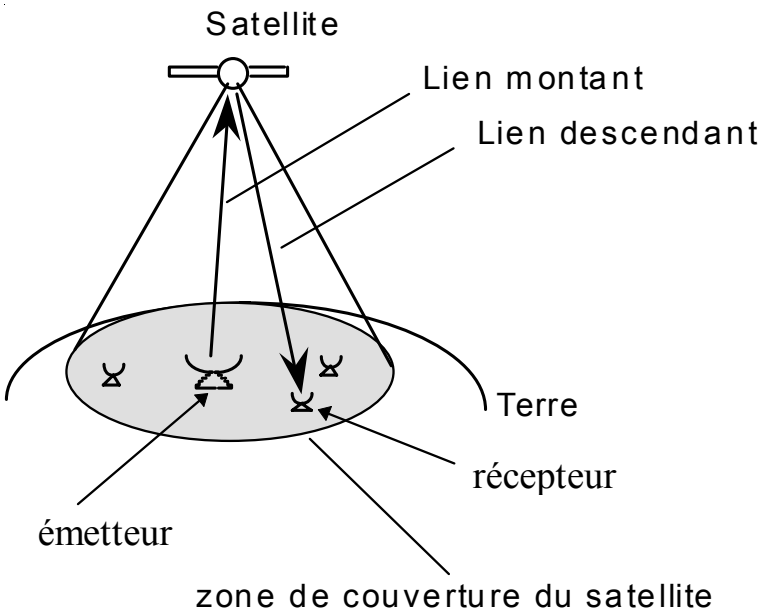


Figure 8-1 : Chaîne de transmission par satellite

Le satellite est géostationnaire et se situe à une distance de 36000 km de la Terre. La fréquence d'émission utilisée pour le lien montant est d'environ 14 GHz. Le satellite comporte un amplificateur de type tube à ondes progressives chargé d'amplifier le signal reçu avant son émission vers la Terre. Sa puissance est limitée car la seule source énergétique dont dispose le satellite est l'énergie solaire.

L'émetteur étant de forte puissance, nous négligerons le bruit introduit dans le lien montant. Par contre, nous considérerons qu'un bruit blanc et gaussien s'additionne au signal dans le lien descendant (cas du « bruit blanc additif gaussien ») étant donné la faible puissance reçue au niveau du récepteur.

Le choix de la modulation retenue est un compromis entre l'efficacité spectrale et la résistance aux distorsions. Compte tenu des non-linéarités du canal de transmission, on montre que la modulation par saut de phase à 4 états est la plus appropriée. Elle permet la transmission de 2 bit/s/Hz.

8.2 Emetteur

La modulation par saut de phase à 4 états (MDP4) est une modulation bidimensionnelle où chaque état de phase Φ(t) représente deux éléments binaires. L'expression en fonction du temps du signal modulé est la suivante :

$$s(t) = A(t).\cos(2.\pi.f_c.t + \Phi(t)) \text{ avec } \Phi(t) = \frac{\pi(2.n+1)}{4}, \; n=0, 1, 2, 3$$

A(t) est l'enveloppe du signal s(t), Φ(t) est sa phase et f_c est la fréquence porteuse. Le signal s(t) peut aussi s'exprimer à partir des composantes en quadrature I(t) et Q(t) par :

$$s(t) = I(t).\cos(2.\pi.f_c.t) - Q(t).\sin(2.\pi.f_c.t) = \Re\left\{(I(t) + jQ(t)).e^{j.2.\pi.fc.t}\right\}$$

avec :

$$A(t) = \sqrt{I^2(t) + Q^2(t)} \text{ et } \Phi(t) = \text{arctg}\frac{Q(t)}{I(t)}$$

I(t)+jQ(t) est l'enveloppe complexe du signal s(t). Le schéma de principe de l'émetteur MDP4 est le suivant :

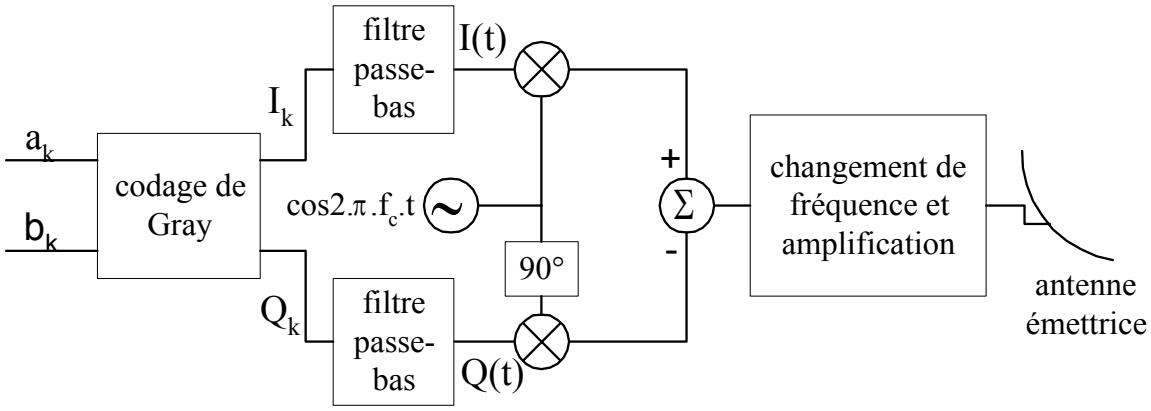


Figure 8-2 : Synoptique de l'émetteur

Les signaux I(t) et Q(t) sont obtenus à partir des éléments binaires a_k et b_k après codage de Gray et filtrage passe-bas. Après modulation, le signal s(t) subit plusieurs changements de fréquence avant d'être amplifié puis émis.

n	a_k	b_k	I_k	Q_k	$\Phi(t)$
0	0	0	1	1	$\pi/4$
1	1	0	-1	1	$3\pi/4$
2	1	1	-1	-1	$5\pi/4$
3	0	1	1	-1	$7\pi/4$

Tableau 8-1 : Valeurs possibles des signaux

Le codage de Gray permet d'éviter qu'une erreur sur le premier bit n'entraîne une erreur supplémentaire sur le second bit du symbole. On peut voir en effet au tableau 8-1, qu'entre deux états de phase adjacents, il n'y a qu'un des deux bits du symbole qui diffère. Ce tableau peut être représenté sous la forme d'une constellation :

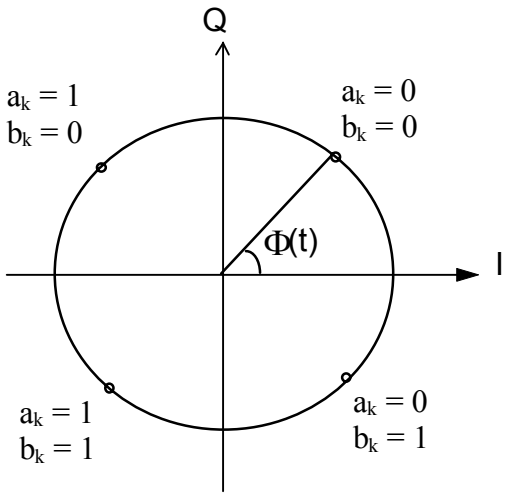


Figure 8-3 : Constellation en MDP4

Pour limiter la largeur spectrale de s(t), il est nécessaire de filtrer les signaux Ik et Qk car ils ont un spectre de largeur infini. La recommandation du groupe DVB préconise l'utilisation d'un filtre de Nyquist avec un facteur d'arrondi (« roll-off ») , α, égal à 0,35.

Il est à noter que le filtrage des signaux I_k et Q_k a une répercussion sur l'enveloppe constante A(t) du signal s(t). Les variations de l'enveloppe après filtrage sont gênantes car l'amplificateur du répéteur satellite est non-linéaire.

8.3 Satellite

Le schéma de principe d'un répéteur satellite est le suivant :

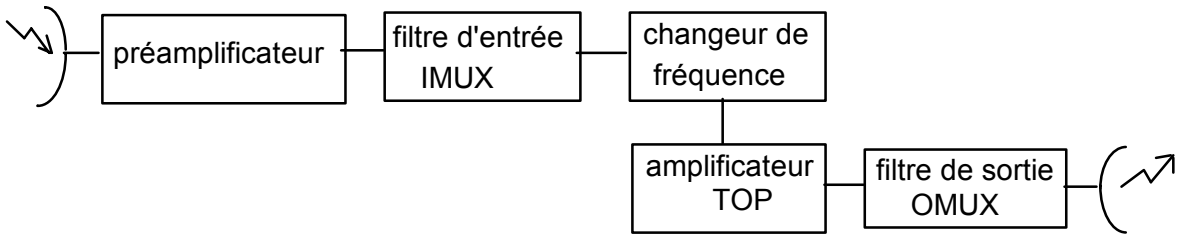


Figure 8-4 : Synoptique d'un répéteur satellite

Le filtre IMUX est un filtre passe-bande centré autour de la fréquence porteuse. Il limite la bande de bruit et les brouillages issus des canaux adjacents. Il est généralement compensé en temps de groupe.

Le tube à ondes progressives est un amplificateur de puissance à haut rendement qui fonctionne à des fréquences très élevées (jusqu'à plusieurs dizaines de GHz) et dans une bande de fréquence très large. Les modèles les plus performants délivrent une puissance supérieure à 200W. En règle générale, le point de fonctionnement du TOP se situe au point de saturation ou légèrement en dessous pour fournir le maximum de puissance en sortie. Il est alors dans un régime non linéaire (non-linéarités de phase et d'amplitude). Ses caractéristiques sont définies de la manière suivante :

Figure 8-5 : Caractérisation du TOP

A(r) définit la non-linéarité d'amplitude (« AM-AM »). Φ(r) définit la non-linéarité de phase (« AM-PM »).

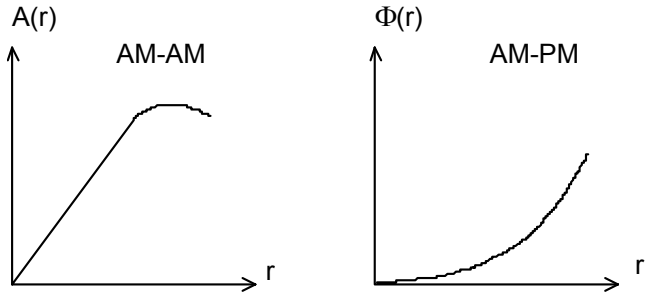


Figure 8-6 : Caractéristiques du TOP

On définit le recul de sortie par la valeur en décibel du rapport entre la puissance de sortie au point de saturation et la puissance de sortie au point de fonctionnement. Selon le type de modulation, le recul de sortie est compris entre 0 et 5 dB. Dans le cas de la modulation MDP4, le recul de sortie est généralement égal à 0 dB.

Plusieurs méthodes permettent à partir des caractéristiques pratiques AM-AM et AM-PM de modéliser le TOP. Ces représentations doivent être suffisamment précises au voisinage de la saturation afin d'effectuer une bonne estimation des performances.

Le fonctionnement en régime non linéaire du TOP entraîne un élargissement du spectre du signal de sortie. Afin de ne pas brouiller les canaux adjacents, il faut limiter ce spectre : c'est la fonction essentielle du filtre OMUX. Le temps de groupe du filtre OMUX n'est pas compensé afin de ne pas diminuer la puissance disponible en sortie du satellite.

Le débit symbole Rs est une caractéristique essentielle du système. Il conditionne le débit utile de la chaîne de transmission. On le compare généralement à la bande passante à -3 dB du canal satellite Bp(-3 dB) qui est déterminée par les filtres IMUX et OMUX. Pour cela, on définit le rapport :

$$\beta = \frac{Bp(-3dB)}{Rs}$$

Il est inversement proportionnel à l'efficacité spectrale (en bits/s/Hz) et permet de s'affranchir des valeurs réelles de débit symbole et de bande passante. La valeur minimale, $\beta=1$, permet d'obtenir l'efficacité maximale de la modulation MDP4 (2 bits/s/Hz). En pratique, β est compris entre 1,1 et 1,4. Avec un code convolutionnel de rendement 2/3, la valeur $\beta=1{,}15$ nous permet d'obtenir le débit binaire souhaité.

$$\textit{débit utile} = \frac{36 \times 2}{1{,}15} \times \frac{2}{3} \times \frac{188}{204} = 38{,}4 \text{ Mbits/s}$$

8.4 Récepteur

Le synoptique suivant présente les divers éléments du récepteur :

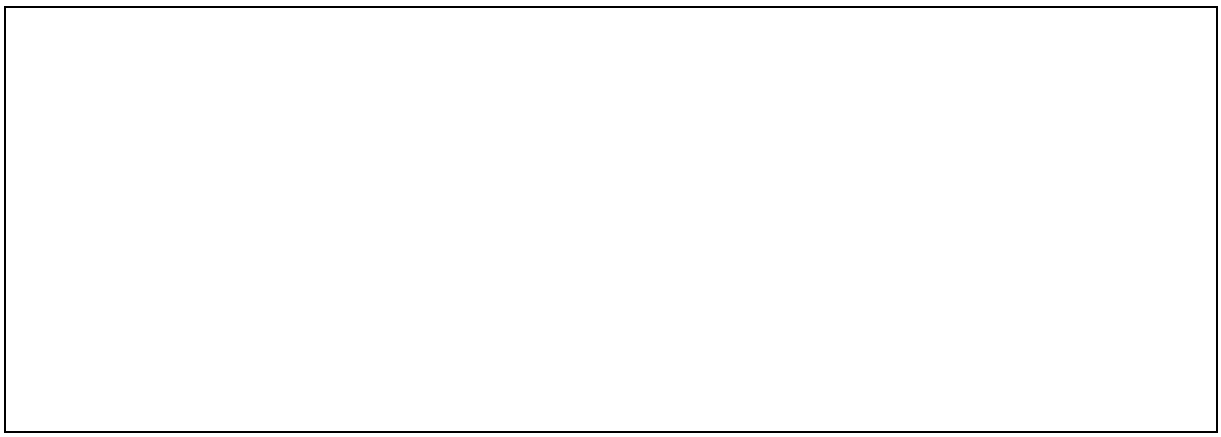


Figure 8-7 : Schéma synoptique du récepteur

On utilise une démodulation cohérente pour démoduler le signal, la récupération de la porteuse s'effectuant directement à partir du signal modulé.

Le signal reçu est de la forme :

$$r(t) = \Re\left\{\left[I(t-\tau) + jQ(t-\tau)\right].e^{j2\pi f_c t}.e^{j\theta}\right\} + n(t)$$

n(t) est le bruit superposé au signal utile. L'objectif du récepteur est, après avoir déterminé les paramètres τ et θ, d'extraire les I_k et Q_k à partir des signaux démodulés I(t) et Q(t).

8.5 Le bilan de liaison

Afin d'évaluer le rapport C/N après l'antenne de réception, il est nécessaire de faire un bilan de puissance de la liaison. En général, ce bilan ne prend pas en compte les dégradations apportées par le trajet montant. En effet, l'amplificateur en sortie de l'émetteur terrestre étant de puissance élevée, le rapport signal sur bruit au niveau de l'antenne réceptrice du satellite est important malgré l'atténuation que subit le signal émis. On ne s'intéressera donc pour effectuer ce bilan qu'à l'ensemble « émetteur satellite - lien descendant - récepteur terrestre ».

8.5.1 Définitions

8.5.1.1 PIRE

Une antenne isotropique est une antenne idéale qui rayonne de façon uniforme dans toutes les directions. Par définition, le gain d'une antenne isotropique vaut 1. Pour cette antenne, l'intensité des radiations I_r est donnée par la relation suivante : $I_r = \frac{P}{4.\pi}$ où P est la puissance injectée dans l'antenne. L'augmentation relative de la puissance rayonnée est obtenue en focalisant les radiations dans une direction privilégiée. Cette augmentation traduit le gain de l'antenne G et on a :

$$G = \frac{\text{Valeur maximale des radiations dans la direction privilégiée}}{\text{Valeur des radiations fournies par une antenne isotropique } (I_r)}$$

La Puissance Isotropique Rayonnée Equivalente (PIRE) est définie par la relation suivante :

$$PIRE = P_E.G_E \text{ [W]} = 10.\log (P_E.G_E) \text{ [dBW]}$$

avec P_E : puissance d'émission

G_E : gain de l'antenne d'émission dans la direction privilégiée de propagation.

8.5.1.2 Densité de flux reçu

Le flux (puissance surfacique) est la quantité de puissance reçue avec une antenne ayant une surface de 1 m^2 située à une distance d (en mètre) de l'antenne d'émission. La surface d'une sphère étant égale à $4.\pi.d^2$, on a la relation :

$$\Phi = \frac{PIRE}{4.\pi.d^2} \ [W/m^2]$$

En France, la distance moyenne entre un satellite géostationnaire et l'antenne de réception est égale à 38000 km. En pratique, il faut multiplier cette relation par A, l'affaiblissement atmosphérique par temps clair (0,3 dB). On exprime alors le flux en dBW/m^2 avec la relation :

$$\Phi \ [dBW/m^2] = PIRE \ [dBW] - 10.\log (4.\pi.d^2) - A \ [dB]$$

8.5.1.3 Affaiblissement atmosphérique à 12 GHz

La pluie est le principal élément perturbateur s'opposant à la propagation des ondes dans l'atmosphère. L'intensité des précipitations et la longueur de leur trajet dans l'atmosphère (angle d'élévation) sont les paramètres qui déterminent l'intensité de la perturbation. Ces précipitations ont deux influences :

1. diminution de la puissance reçue (affaiblissement atmosphérique).

2. augmentation de la température de bruit.

La diminution du rapport C/N à la réception est égale à la somme des deux valeurs correspondantes.

Pour la France, les résultats suivants ont été obtenus pour une station réceptrice ayant un facteur de bruit du LNB égal à 1,2 dB :

pourcentage du temps du mois le plus défavorable	99 %	99,9%
affaiblissement atmosphérique (ΔC)	1,2 dB	4 dB
augmentation de la température de bruit (ΔN)	1,5 dB	3 dB
diminution du C/N (ΔC + ΔN) (par rapport à la réception par temps clair)	2,7 dB	7 dB

Tableau 8-2 : Valeurs de diminution de C/N

On lit, par exemple, que pendant 99 % du temps du mois le plus défavorable, la diminution du C/N par rapport à une réception par temps clair a été inférieure à 2,7 dB (1,2 + 1,5). La qualité de service d'une diffusion par satellite n'est jamais garantie 100 % du temps.

8.5.1.4 Facteur de qualité de l'installation de réception

Pour une antenne parabolique en réception, on définit le gain Gi par la formule approchée :

$$\mathrm{Gi} = \eta\left(\frac{\pi.\mathrm{D}}{\lambda}\right)^2$$

avec η : rendement de l'antenne (compris entre 0,5 et 0,8),

D : diamètre de l'antenne de réception,

λ : longueur d'onde du signal reçu.

Le facteur de qualité de l'installation de réception est défini par la formule :

$$\frac{G}{T} = \frac{\alpha.\beta.Gi}{\alpha.T_A + (1-\alpha).T_0 + (F-1).T_0} \quad [^\circ K^{-1}]$$

avec α : pertes de couplage (proche de 1),

β : pertes dues à l'erreur de pointage et au vieillissement (proche de 1),

Gi : gain de l'antenne de réception,

T_A [°K] : température d'antenne,

T_0 : température ambiante (T_0 = 290 °K),

F : facteur de mérite du LNB (Low Noise Block convertor) en général exprimé en dB.

Ce facteur de qualité s'exprime généralement en dB :

$$\frac{G}{T}\ [\text{dB/°K}] = 10.\log(\alpha.\beta.Gi) - 10.\log(\alpha.T_A) - 10.\log(1-\alpha).T_0 - 10.\log(F-1).T_0$$

On appelle $(1-\alpha).T_0$ le bruit de couplage et $(F-1).T_0$ la température de bruit du LNB.

8.5.1.5 Rapport porteuse à bruit

C'est le rapport entre la puissance de la porteuse et la puissance de bruit totale ramenée à l'entrée du LNB.

$$\frac{C}{N} = \Phi.\frac{G}{T}.\frac{\lambda^2}{4.\pi.k.B_W}$$

avec Φ : flux reçu par l'antenne,

$\frac{G}{T}$: facteur de mérite de l'installation de réception,

k : constante de Boltzmann = $1{,}38.10^{-23}$ J/°K,

B_W : bande passante du canal satellite.

Cette formule s'exprime aussi en dB :

$$\text{C/N [dB]} = \Phi\ [\text{dBW/m}^2] + \frac{G}{T}\ [\text{dB/°K}] + 10.\log\left(\frac{\lambda^2}{4.\pi.k.B_W}\right)$$

8.5.2 Exemple de calcul des paramètres d'une liaison par satellite

8.5.2.1 Paramètres de la liaison

Les paramètres techniques de la liaison entre le satellite et le décodeur de télévision numérique sont les suivants :

1. Caractéristiques du satellite HOT BIRD 1 (EURELSAT).

- PIRE à Paris : 51 dBW,
- Largeur d'un canal : B_W = 36 MHz,
- Débit symbole : 27,5 Mbauds,
- Débit binaire brut : 55 Mbit/s,

- fréquence du signal reçu : f = 12,521 GHz.

2. Caractéristiques de la modulation.

- Modulation MDP4,
- Roll-off des filtres de Nyquist : $\alpha = 0.35$,
- Rendement du code convolutif : r = 3/4.

3. Caractéristiques de l'installation de réception.

- Facteur de bruit du LNB : F = 1,1 dB,
- Rendement de l'antenne : $\eta = 70$ %,
- Température de bruit d'antenne : $T_A = 30$ °K,
- Pertes de couplage négligeables,
- Pertes dues à l'erreur de pointage et au vieillissement : $\beta = -0,5$ dB.

Afin de compenser les principales distorsions de la liaison par satellite, on tiendra compte des marges suivantes :

- Marge due à la non-linéarité du répéteur satellite utilisé à saturation : 0,8 dB,
- Marge de réalisation du démodulateur : 0,8 dB,
- Marge due aux brouillages ACI et CCI : 0,4 dB,
- Marge due à l'augmentation de bruit provenant du codage de Reed-Solomon : $10.\log\left(\frac{188}{204}\right) = 0,36$ dB,
- Marge due aux limites de bande passante des filtres de démultiplexage : 0,2 dB.

8.5.2.2 Détermination du C/N

On va maintenant déterminer la valeur du C/N correspondant à une continuité de service assurée pendant 99,9 % du temps du mois le plus défavorable. On cherche à obtenir une qualité d'image sans défaut, c'est-à-dire obtenue avec un canal quasiment sans erreurs. Cette qualité de service est obtenue avec un taux d'erreur égal à 2.10^{-4} après le décodeur de Viterbi. Dans le cas d'une modulation MDP4 associée avec un code convolutionnel de rendement 3/4, on obtient :

$$\frac{E_b}{N_0} = 4 \text{ dB}.$$

Le débit utile (hors codes correcteurs d'erreurs) est égal à :

$$\text{débit utile} = 55.10^6.\frac{188}{204}.\frac{3}{4} = 38 \text{ Mbit / s}.$$

Or on sait que

$$\frac{C}{N} = \frac{Eb}{No}.\frac{\text{Débit utile}}{\text{bande passante canal}}$$

ce qui donne en dB :

$$\frac{C}{N} = 4 + 0{,}2 = 4{,}2 \text{ dB}$$

Comme on souhaite une continuité de service assurée pendant 99,9 % du temps du mois le plus défavorable, il est nécessaire d'ajouter une marge de 7 dB. La valeur minimale du C/N par temps clair doit donc être égale à :

$$\frac{C}{N} = 4{,}2 + 7 = 11{,}2 \text{ dB}$$

Il faut ensuite ajouter les marges dues aux distorsions de la liaison par satellite :

$$M = 0{,}8 + 0{,}8 + 0{,}4 + 0{,}36 + 0{,}2 = 2{,}56 \text{ dB}$$

Au total, la valeur minimale du C/N par temps clair doit donc être égale à :

$$\frac{C}{N} = 11{,}2 + 2{,}56 \cong 13{,}8 \text{ dB}$$

8.5.2.3 Facteur de qualité de l'installation de réception

On rappelle que :

$$C/N\ [dB] = \Phi\ [dBW/m^2] + \frac{G}{T}\ [dB/°K] + 10.\log\left(\frac{\lambda^2}{4.\pi.k.B_W}\right)$$

d'où :

$$\frac{G}{T}\ [dB/°K] = C/N\ [dB] - \Phi\ [dBW/m^2] - 10.\log\left(\frac{\lambda^2}{4.\pi.k.B_W}\right)$$

- Calculons le flux reçu. On a :

$$\Phi\ [dBW/m^2\ ] = PIRE\ [dBW] - 10.\log\left(4.\pi.d^2\right) - A\ [dB]$$
$$51 - 10.\log(4.\pi.3800000^2) - 0.3$$

Ce qui donne : $\Phi = -111{,}9$ dB W/m^2.

- Calculons la quantité (avec $\lambda = c / f$) :

$$10.\log\left(\frac{\lambda^2}{4.\pi.k.B_W}\right) = 109{,}6\ dB$$

On obtient finalement le facteur de qualité :

$$\frac{G}{T} = 13{,}8 + 111{,}9 - 109{,}6 = 16{,}1\ dB/°K$$

8.5.2.4 Diamètre de l'antenne de réception

On sait que l'on a :

$$\frac{G}{T}[dB] = G\ [dB] - T\ [dB] \quad \Rightarrow \quad G\ [dB] = \frac{G}{T}[dB] + T\ [dB]$$

avec $T = \alpha.T_A + (1-\alpha).T_0 + (F-1).T_0$. Comme les pertes de couplages sont considérées comme négligeables ($\alpha = 1$), on a :

$$T = T_A+(F-1).T_0 = 30 + (10^{1,1/10} - 1).290 = 113,6\ °K = 20,5\ dB$$

$$G = 16,1 + 20,5 = 36,6\ dB$$

Or $G = \beta.Gi$, d'où $Gi\ [dB] = G\ [dB] - \beta\ [dB] = 36,6 - (- 0,5) = 37,1\ dB$

$$\Rightarrow Gi = 5128,6$$

Comme $Gi = \eta\left(\frac{\pi.D}{\lambda}\right)^2$, on tire :

$$D = \frac{\lambda}{\pi}.\sqrt{\frac{Gi}{\eta}} = 65\ cm$$

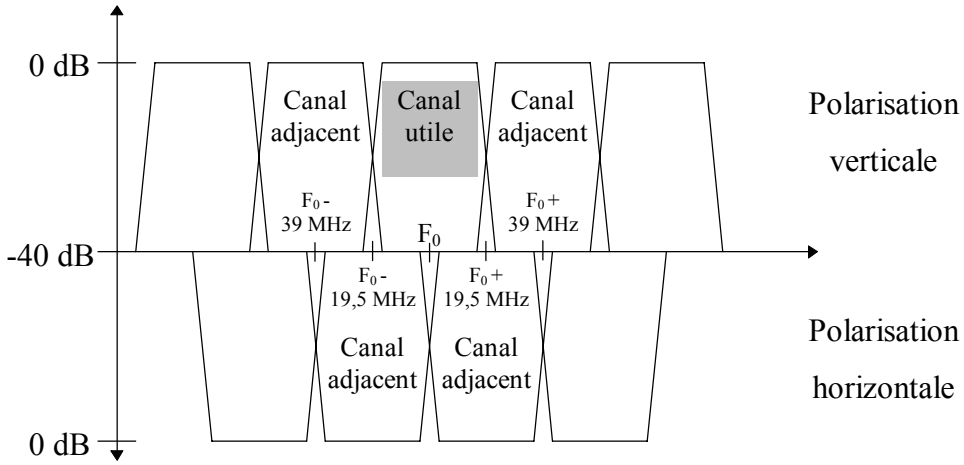
8.6 brouillage entre canaux

Le signal modulé est souvent perturbé par des brouillages causés par des signaux parasites dont l'influence se fait sentir dans la bande occupée par le signal utile. Ces signaux parasites, baptisés brouilleurs, entraînent une dégradation des performances de la chaîne de transmission. Le brouilleur est caractérisé par le rapport C/I :

C : puissance moyenne du signal modulé utile après le filtre de réception,

I : puissance moyenne du brouilleur après le filtre de réception.

Dans le cas d'une chaîne de transmission de type répéteur satellite, les brouilleurs sont liés à l'utilisation de plans de fréquence qui définissent la répartition des différents signaux modulés. Pour obtenir une occupation spectrale minimale, on réduit au maximum l'écart entre les fréquences porteuses ce qui a pour effet d'augmenter le brouillage entre le canal satellite utile et les canaux adjacents (brouillage ACI). Le plan de fréquence le plus souvent utilisé en Europe (satellite ASTRA, EUTELSAT et TELECOM) est le suivant :



L'emploi de polarisations orthogonales permet de doubler le rendement spectrale en bit/s/Hz. Cependant, le découplage entre les deux polarisations n'étant pas parfait, ceci constitue une cause supplémentaire de brouillage ACI.

De plus, certains satellites utilisent les mêmes canaux ainsi que les mêmes polarisations. Dans ce cas, si l'antenne réceptrice n'est pas suffisamment sélective, un brouillage entre canaux est à prévoir (brouillage co-canaux CCI).

9 La transmission point à point par faisceau hertzien

9.1 Description du système

Le groupement européen DVB a retenu un système de diffusion de télévision numérique, le DVB-MS, basé sur la norme éditée pour la diffusion de télévision numérique par satellite. Ce système, appelé MVDS (Multipoint Video Distribution Systems), permet la transmission de programmes à partir d'une antenne installée sur le toit d'un bâtiment élevé et leur réception par des récepteurs situés dans la ligne de vue. Ces programmes pourront donc être captés par des décodeurs satellites en remplaçant la parabole satellite par un convertisseur MVDS.

Ce système est basé sur le principe du MMDS (Multichannel Microwave Distribution System) surnommé "câble hertzien", qui permet de constituer des réseaux de distribution hyperfréquence. Il est adapté du système utilisé aux Etats-Unis depuis le début des années soixante.

La figure suivante représente les différentes étapes nécessaires pour adapter les caractéristiques d'un signal de télévision en sortie d'un multiplex de transport MPEG2 à celles du "canal MVDS".

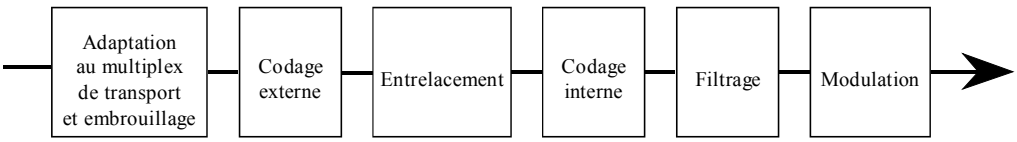


Figure 9-1 : Emetteur

Pour des raisons de compatibilité et de coût, les différentes fonctions du système MVDS sont identiques à celles utilisées pour la diffusion par satellite. Les principaux paramètres du système sont les suivants :

Largeur du canal (-3 dB)	33 MHz
Modulation	QPSK
Code convolutionnel	rendement de 1/2 à 7/8
Code Reed-Solomon	(204,188) T=8
Bits par symbole	2
Débit Utile Maximum	de 23.75 Mbps à 41.57 Mbps
C/N	de 4.1 dB à 8.4 dB

Tableau 9-1 : Paramètres du système MVDS

On voit sur la figure suivante un exemple de réseau MVDS. On regroupe sur la tête de réseau les différents programmes satellites, les chaînes hertziennes ainsi que d'éventuelles chaînes locales et on les distribue à l'utilisateur par le biais de faisceau micro-ondes.

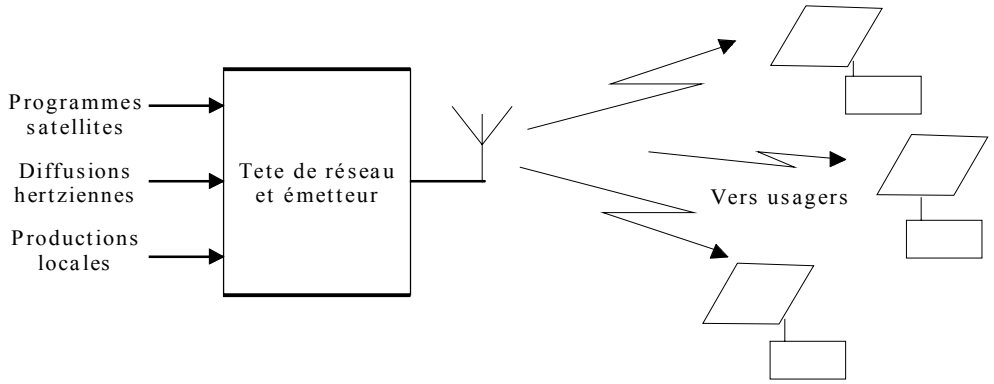


Figure 9-2 : Application du MVDS

Pour la mise en place d'une chaîne locale ou régionale, le coût global et la difficulté de mise en oeuvre élimine le satellite et le coût d'installation et d'exploitation d'un réseau câblé est élevé. Le système MMDS apporte alors une solution simple et peu coûteuse.

9.2 Zone de couverture

La portée du système est sensible au relief. Elle est aussi fonction de la puissance de l'émetteur et de la taille de l'antenne de réception. A titre d'exemple, avec un émetteur de 1 Watt fonctionnant à la fréquence de 12 GHz et une antenne de réception de 10 cm, le signal

sera reçu dans un rayon de 30 km. Le même émetteur et une antenne de réception de 28 cm porteront l'acquisition du signal à une distance de plus de 100 km. La couverture peut être étendue grâce à l'emploi de relais. L'opérateur crée ainsi un réseau de type cellulaire, sans générer d'interférence entre les canaux.

Pour une zone de couverture équivalente, la puissance d'un émetteur UHF est de l'ordre du kilowatt alors qu'elle n'est que d'environ 30 watts pour MMDS (bande 2.5 GHz). Dans la bande des 2 GHz, il n'y a pas de problème particulier de propagation. Les atténuations dues à la pluie ou à la neige sont inférieures à 0.05 dB/km. Les échos créés par la réflexion sur le relief sont sévèrement atténués à cause du niveau d'absorption à ces fréquences et à la faible puissance (relative) d'émission utilisée. Par contre dans la bande des 30 GHz cette atténuation est très importante, jusqu'à 20 dB/km. La couverture maximum disponible pour ces émetteurs n'excède alors pas plus de 3 ou 4 km de diamètre.

On peut toutefois considérer l'affaiblissement comme un avantage pour créer des couvertures locales en utilisant les bandes de fréquences supérieures. Mais cela se fait au détriment d'une puissance d'émission plus importante.

9.3 Réglementation

En France, le contexte réglementaire empêche à ce jour l'émergence de cette solution. Le MMDS a été conçu, au départ, pour regrouper les avantages du câble, du satellite et de la diffusion hertzienne sans en présenter les inconvénients.

Cette situation hybride aurait dû en assurer le succès. En fait, elle a constitué un frein à son développement, car il est difficile de situer clairement ce nouveau mode de communication. La France a opté pour un rattachement à la distribution par câble. Les possibilités sont limitées aux seules liaisons, internes aux réseaux câblés, en excluant la réception individuelle.

Les conséquences d'un tel choix sont importantes car le MMDS ne peut pas, dans les zones rurales, se substituer au câble, toujours utilisé pour la distribution finale au téléspectateur. Il ne constitue pas un moyen de diffusion terrestre, il se limite à remplacer certains tronçons de câble.

Les fréquences traditionnellement utilisées par le MMDS dans les pays qui ont développé ce service appartiennent à la bande des 2.5 GHz. En France métropolitaine, cette bande est affectée aux forces armées. Le CSA a souhaité voir se développer ce système de diffusion dans la bande 3.6-3.8 GHz. La DGPT (Direction Générale des Postes et Télécommunications) demande que le MMDS utilise une bande affectée à la radiodiffusion. La seule qui soit disponible est la bande des 40 GHz qui ne permet pas des couvertures dépassant quelques kilomètres.

A terme, il est probable que le MMDS se développe dans les zones rurales avec la possibilité d'atteindre directement le téléspectateur, devenant ainsi un complément de la télévision terrestre. Il peut de plus être un concurrent du réseau câblé pour les petites agglomérations, comme aux Etats-Unis, où il est considéré comme un moyen de diffusion.

A plus petite échelle et au-delà de l'application purement télévisuelle, le système peut permettre également la transmission audio et de données informatiques. Nous avons alors un véritable outil multimédia. L'interactivité est assurée, soit par le réseau téléphonique, soit par la voie de retour MDS ou encore une liaison radio type GSM. Ce nouveau concept est appelé réseau d'accès large bande LMDS (Local Multipoint Distribution System).

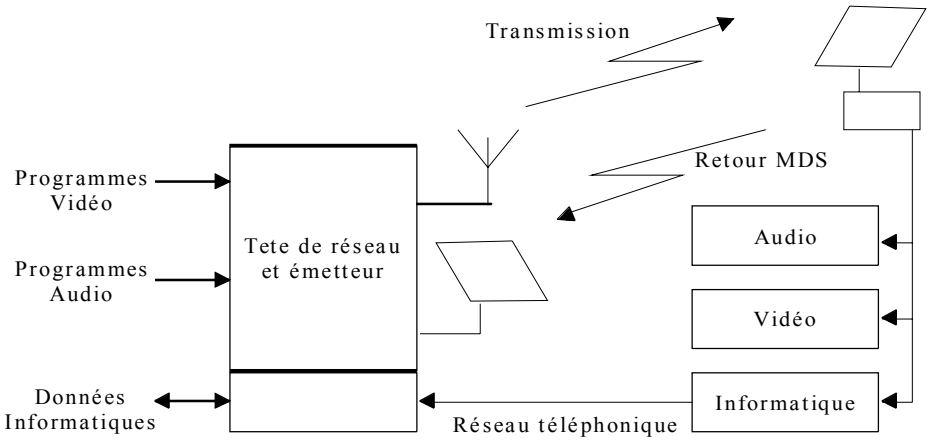


Figure 9-3 : Réseau LMDS

10 La diffusion sur un réseau de distribution collective par câble

Un réseau de distribution collective par câble permet de transporter des signaux TV à partir d'une tête de réception satellite commune vers les différentes prises d'usager de l'immeuble (ou de la résidence collective). Ce système de distribution collective est appelé SMATV (Satellite Master Antenna Television). Sa réalisation doit respecter certaines conditions définies par des organismes de normalisation :

- UTE (Union Technique de l'Electricité),
- CENELEC (Comité Européen de Normalisation Electrotechnique),
- CEI (Commission Electrotechnique Internationale).

10.1 Structure de réseau

Dans la plupart des cas, les réseaux de distribution collective installés dans les immeubles ont soit une structure en étoile soit une structure arborescente.

10.1.1 Réseau en structure étoile

Cette structure utilise une seule armoire de distribution placée de préférence au pied de chaque colonne montante de l'immeuble. Regroupant les câbles de branchement, elle convient à un immeuble de quelques étages.

Dans cette configuration, la répartition du niveau du signal est fonction de la longueur du câble coaxial utilisée pour relier chaque logement. En effet, les sorties du PRD ("Point de Regroupement de Distribution") ont des atténuations variables, ce qui permettra d'ajuster les différents niveaux. Par conséquent, les logements les plus éloignés auront un niveau de signal plus important à la sortie du PRD après amplification, afin de compenser la perte due à la longueur du câble. La figure 10-1 présente l'exemple d'un immeuble en structure étoile.

10.1.2 Réseau en structure arborescente

Cette structure utilise plusieurs coffrets de distribution répartis sur différents niveaux dans chaque colonne montante de l'immeuble. Dans cette configuration, la répartition du niveau du signal ne dépend plus de la longueur du câble utilisé par chaque logement. En effet, les sorties du PRD ont une sortie principale à très faible perte et plusieurs sorties dérivées à atténuation constante. Par conséquent, le niveau du signal autour d'un PRD sera sensiblement identique.

On retrouve cette structure dans la plupart des grands immeubles ou l'absence de gaine technique dans l'immeuble ne permettre pas d'adopter une distribution en étoile.

Dans les deux structures de réseau présentées, il est conseillé de prévoir un point de desserte collectif (PDC), placé de préférence au pied de l'immeuble pour permettre éventuellement une interconnexion avec le réseau câblé de la ville. La figure 10-2 présente l'exemple d'un immeuble en structure arborescente.

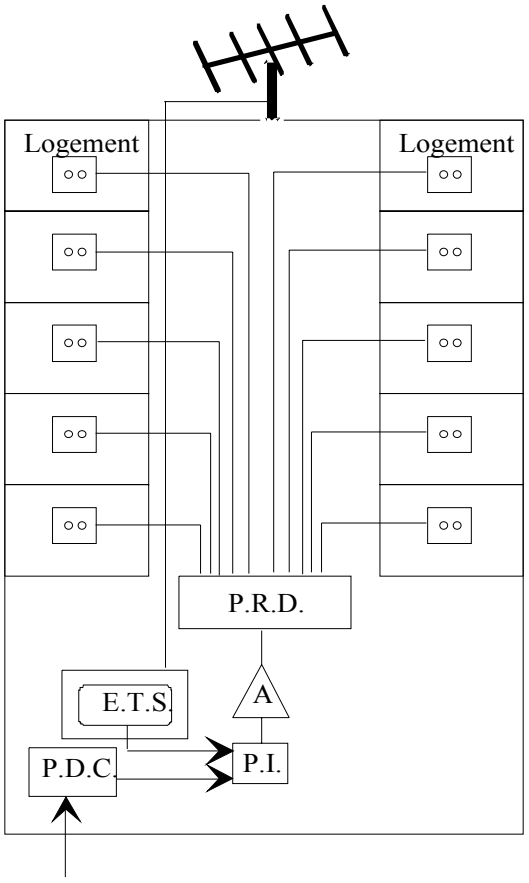


Figure 10-1 : Structure étoile.

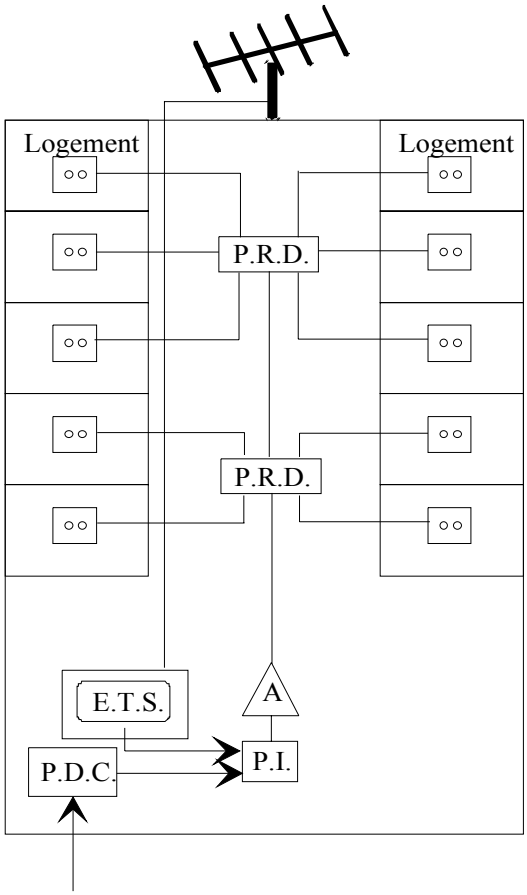


Figure 10 -2 : Structure arborescente.

P.D.C. : Point de Desserte Collectif.
P.R.D. : Point de Regroupement de Distribution.
E.T.S. : Equipement de Traitement local des Signaux.
P.I. : Point d'Interface.
A : Amplificateur.

Le choix de la structure de distribution peut être différent suivant la structure de l'immeuble.

Le tableau 10-1 donne la recommandation sur le choix de la structure de distribution.

	Structure étoilée	Structure arborescente
Immeuble ≤ Rez-de-chaussée + 7 étages Nombre de logements par colonne montante ≤ 32	*	
Immeuble > Rez-de-chaussée + 7 étages Nombre de logements par colonne montante ≤ 48		*
Autre cas		*

Tableau 10-1 : Recommandation sur le choix de la structure de distribution.

10.2 Comportement du réseau de distribution collectif

Un réseau de distribution SMATV est constitué par des câbles coaxiaux, des répartiteurs et dérivateurs, un ou plusieurs amplificateurs à large bande. Du fait que les éléments utilisés dans le réseau ne sont pas parfaits, ils apportent successivement des dégradations dans le réseau. Ces dégradations peuvent être classées dans les catégories suivantes :

- Perte de transmission des répartiteurs et dérivateurs,
- pertes linéiques des câbles coaxiaux,
- distorsion non-linéaires dans les amplificateurs (problèmes d'intermodulation),
- distorsions d'amplitude et de phase causés par des échos.

C'est cette dernière catégorie de défauts qui aura le plus d'influence sur le signal de télévision numérique. Dans un immeuble, la distribution des signaux de télévision utilise dans la plupart des cas des câbles coaxiaux et la répartition des signaux est assurée par des dérivateurs et des répartiteurs. La liaison entre les logements peut générer des échos en raison des désadaptations d'impédance entre les connexions. On peut classer les échos en plusieurs catégories :

- les échos entre étages,
- l'écho entre la tête de réseau et le premier point de branchement,
- l'écho entre un point de branchement et une prise d'usager.

Une campagne de mesure a été effectuée dans le projet DIGSMATV sur des réseaux collectifs existant. Les résultats de ces mesures ont permis de modéliser un réseau de référence. Le modèle représentant les échos de ce réseau de référence est le suivant :

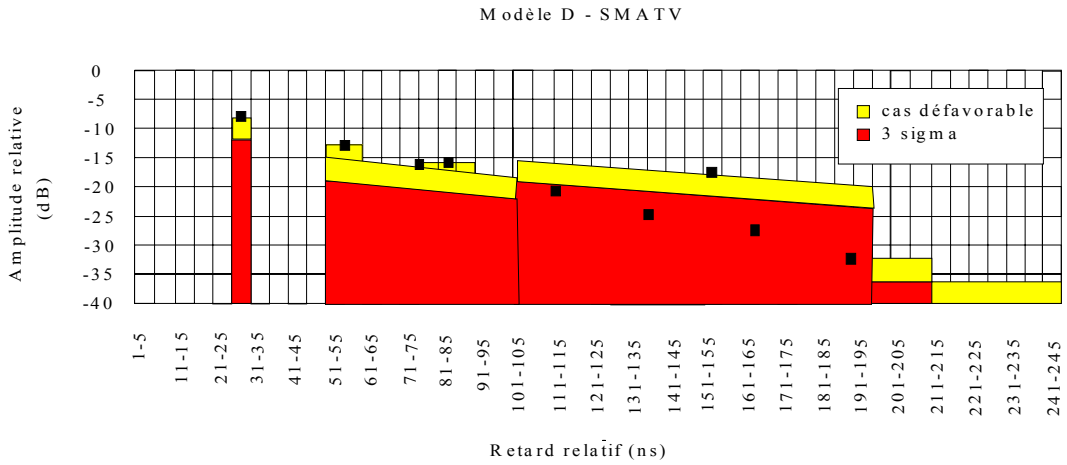


Figure 10-3 : accumulation des échos

On peut donc modéliser le canal câble par le schéma suivant :

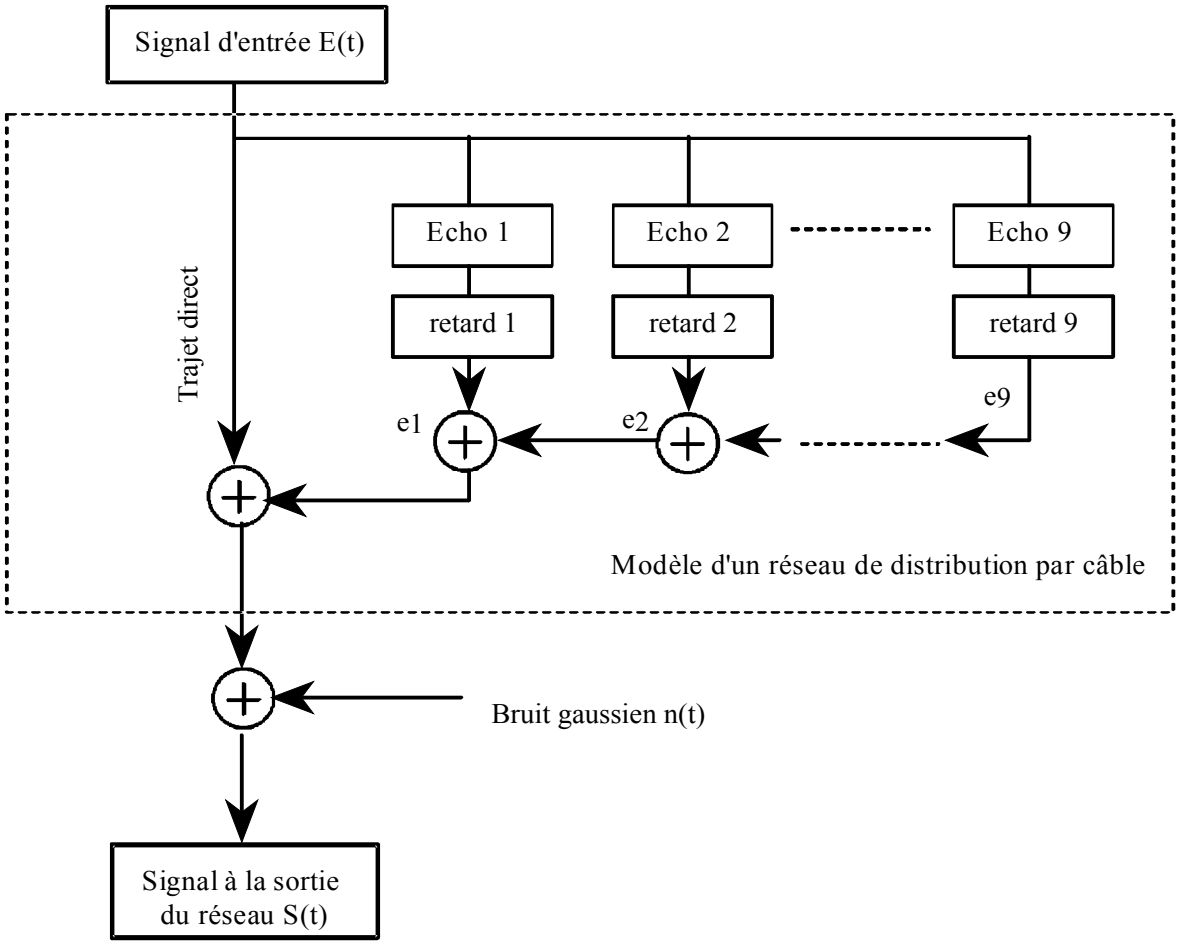


Figure 10-4 : accumulation des échos

10.3 Systèmes de distribution des signaux TV numérique en collectivité

10.3.1 Introduction

La modulation et le système de codage de canal utilisés dans les réseaux SMATV ont été définis par l'ETSI (European Telecommunications Standards Institute) dans le projet DVB-SMATV. Afin de minimiser le coût de l'étude, ce projet propose de conserver le même système de codage de canal que celui utilisé pour la distribution directe par satellite.

10.3.2 Les modulations

La norme ETS 300 473 définit deux types de modulation : soit une modulation QAM (Modulation d'amplitude en quadrature), soit une modulation QPSK (modulation par déplacement de phase en quadrature). On appelle I et Q les coordonnées dans l'espace des signaux d'un point de la constellation retenue. Les constellations suivantes sont utilisées :

- $I \in \{\pm 1\}$ et $Q \in \{\pm 1\}$ correspond à 4 états,
- $I \in \{\pm 1 \pm 3\}$ et $Q \in \{\pm 1 \pm 3\}$ correspond à 16 états,
- $I \in \{\pm 1 \pm 3 \pm 5\}$ et $Q \in \{\pm 1 \pm 3 \pm 5\}$ correspond à 32 états,
- $I \in \{\pm 1 \pm 3 \pm 5 \pm 7\}$ et $Q \in \{\pm 1 \pm 3 \pm 5 \pm 7\}$ correspond à 64 états.

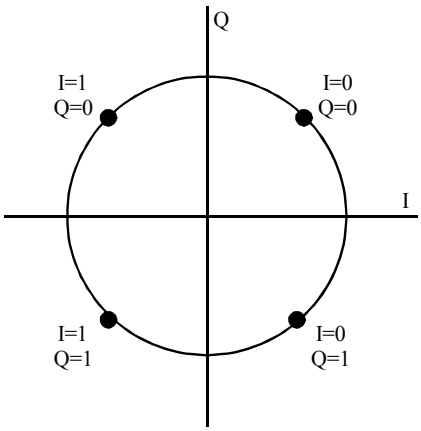


Figure 10-5 : Constellation en QPSK.

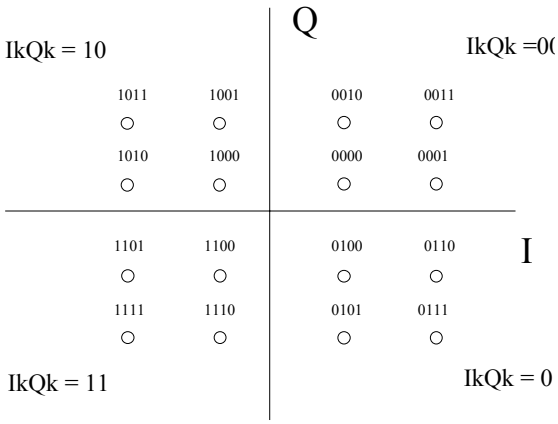


Figure 10-6 : Constellation d'une QAM à 16 états.

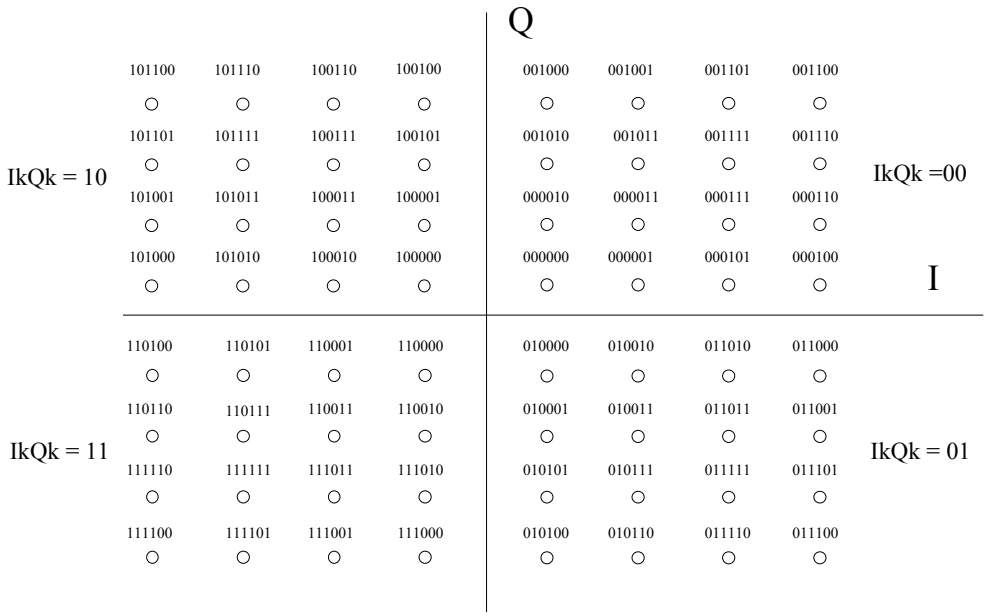


Figure 10-7 : Constellation d'une QAM à 64 états.

Les taux d'erreurs binaires théoriques espérés sont les suivants :

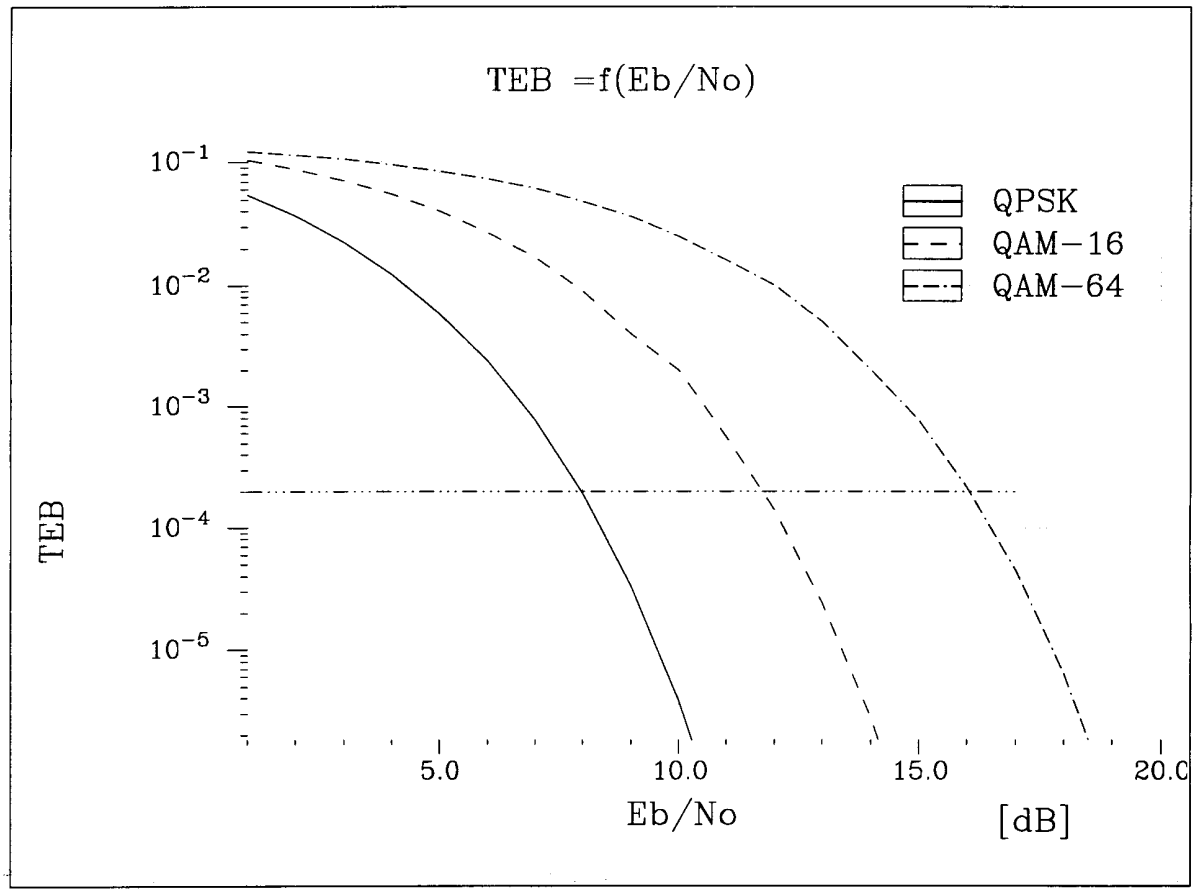


Figure 10-8 : TEB pour différentes modulations

10.3.3 Techniques de distribution

Dans les installations collectives, deux systèmes de distribution des signaux TV numériques sont envisagés. Le système A permet de respecter la bande passante des canaux terrestres, le système B permet d'avoir une distribution directe à un coût minimum.

- Système A : Transmodulation. Pour des raisons de compatibilité avec la largeur de bande des canaux de télévision analogique, les signaux QPSK en provenant de la tête de réception sont décodés puis recodés en format QAM à multiples niveaux (QAM16, 32 ou 64). En fonction du nombre de niveaux utilisés, la modulation QAM permettra d'augmenter la capacité de transmission tout en respectant la bande passante. La largeur des canaux utilisés dans les collectivités est de 8 MHz.
- Système B : Distribution directe. Le principe de ce système consiste à distribuer les signaux TV numérique de la manière la plus directe possible aux utilisateurs. Dans ce cas, les signaux QPSK ne nécessitent pas de décodage et recodage intermédiaire, ce qui permet d'assurer une distribution à un coût minimum.

10.3.3.1 Système A : Transmodulation

Principe : l'approche de ce système consiste à effectuer une transmodulation. Le signal QPSK en provenance du satellite occupe une largeur de bande assez importante (36 MHz à -3 dB). Afin de pouvoir distribuer ce signal dans les réseaux de distribution par câble dont la largeur de bande est limitée à 8 MHz, le signal QPSK d'origine est transcodé en un signal QAM (Quadrature Amplitude Modulation). La figure 10-9 représente la configuration du système A par transmodulation.

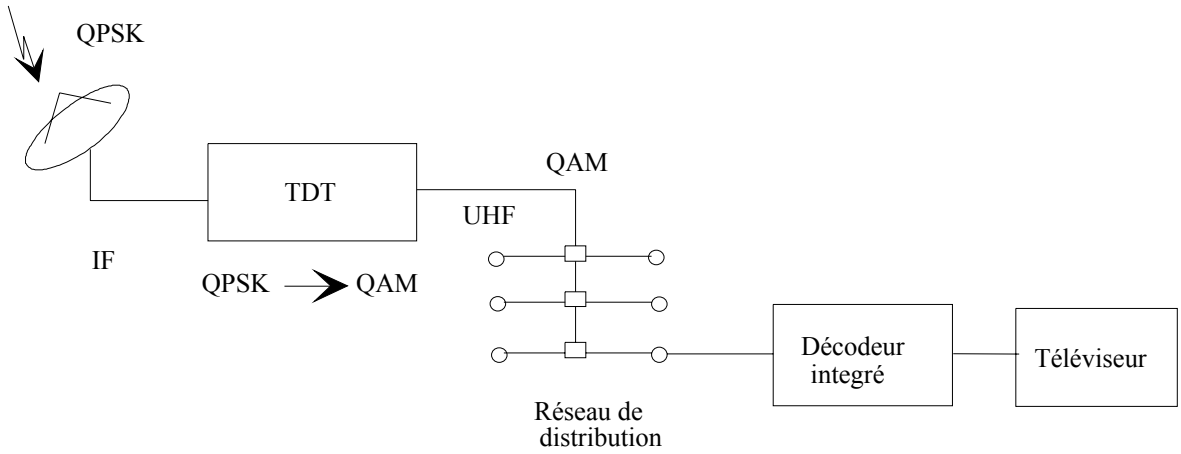


Figure 10-9 : Configuration du système A par transmodulation.

Ce système est aussi appelé SMATV-DTM (SMATV system based on Digital Transmodulation). Il est composé par les éléments suivants :

- Une unité de transmodulation TDT (Transparent Digital Transmodulation) : Cette unité se situe à la tête de distribution. Elle assure la démodulation du signal QPSK et la modulation du signal QAM pour être compatible avec la largeur du canal. Elle effectue les codages nécessaires pour la distribution dans les réseaux de collectifs par câble.
- Un réseau de distribution UHF : c'est une structure de câble coaxial utilisée pour distribuer les signaux aux utilisateurs.
- Un décodeur intégré (récepteur) : cet ensemble réalise la démodulation et le décodage du signal QAM et éventuellement l'égalisation nécessaire pour compenser la distorsion du canal utilisé.

Le diagramme fonctionnel de l'unité de transmodulation est représenté par la figure 10-12. On reconnaît dans ce diagramme des éléments du codage de canal utilisés pour la diffusion par satellite. Seules les fonctions suivantes sont à étudier :

- Convertisseur octet-mot : Il réalise une conversion des octets en mots de m bits. 2^m représente les niveaux des modulations QAM16, QAM32 et QAM64 respectivement pour m = 4, 5 et 6. La figure suivante représente le cas d'une QAM 64 (m = 6, k = 3 et n = 4).

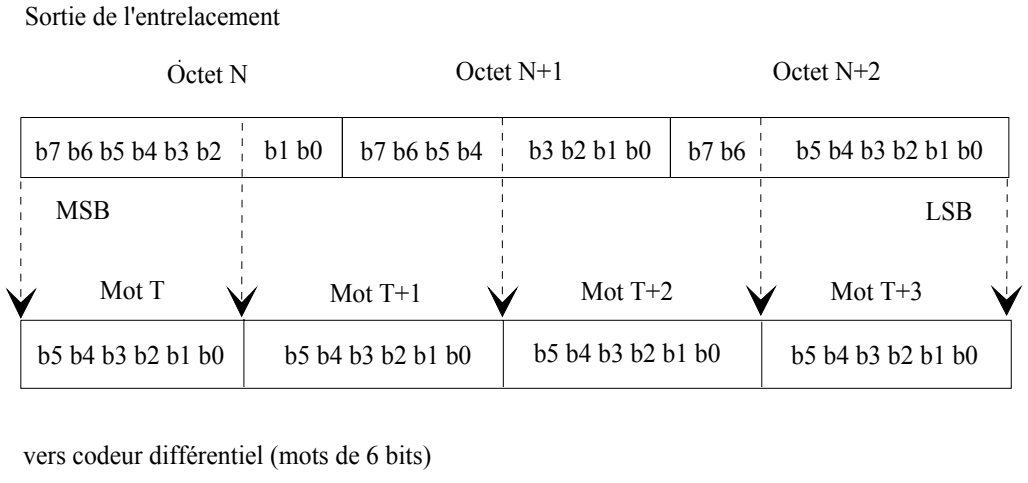


Figure 10-10 : Conversion octet-mot pour QAM 64.

- Codeur différentiel : Les deux bits de poids fort de chaque symbole subissent un codage différentiel selon la configuration suivante :

Quadrant	2 bits de poids forts (MSB)	Rotation des 4 bits restants (LSB)
1	00	0
2	10	+ π/2
3	11	+ π
4	01	+ 3π/2

Tableau 10-2 : Codage différentiel.

La figure 10-11 représente le schéma fonctionnel de la conversion, le codage et le placement des points dans la constellation :

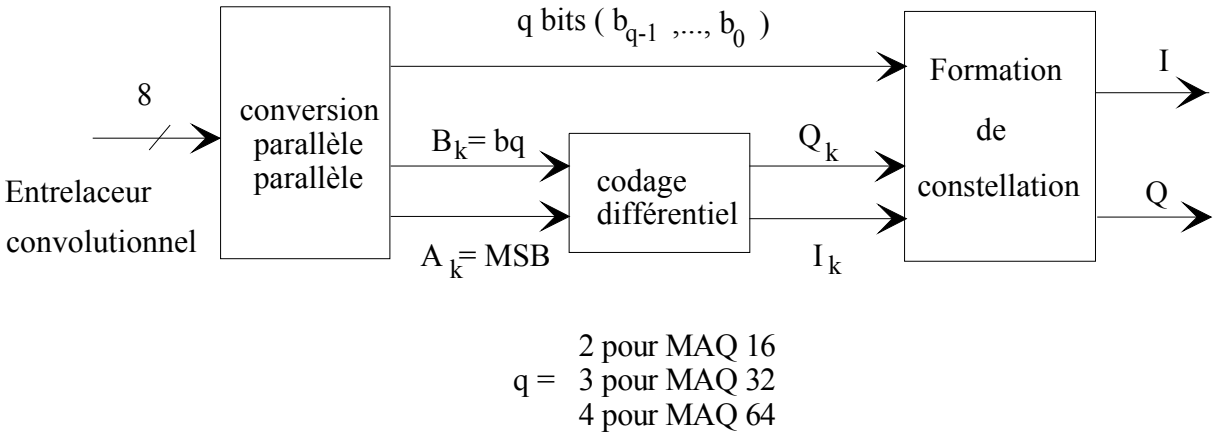


Figure N° 10-11 : Schéma fonctionnel de conversion octet-mot et codage différentiel.

- Filtre de Nyquist : cette partie réalise un filtrage en racine de cosinus surélevé sur des signaux I et Q en utilisant un facteur d'arrondi de 0,15 définit par la norme.

- Modulation QAM et interface physique : cette partie effectue une modulation des signaux I et Q dans une des fréquences de la bande VHF/UHF du réseau de la collectivité.

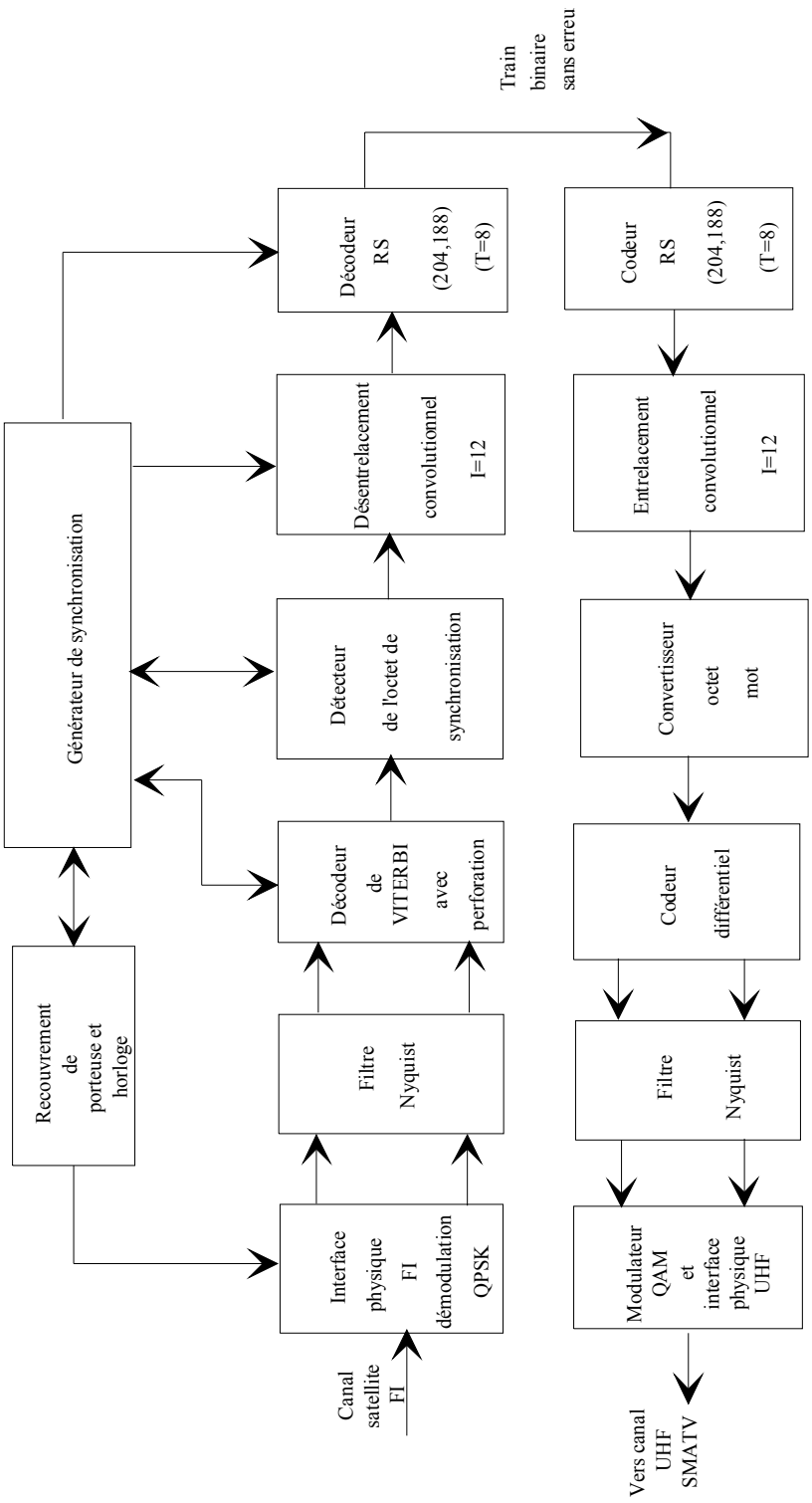


Figure 10-12 : Diagramme fonctionnel de la transmodulation QPSK en QAM.

Objectif : les signaux de TV numérique en provenance du satellite sont à l'origine en modulation QPSK. Le débit brut du signal QPSK correspondant à un canal satellite de type ASTRA est égal à 56,3 Mbit/s, pour un rapport de BW/Rs = 1,28 et une bande passante de 36

MHz. Pour pouvoir distribuer ces signaux dans le réseau collectif, il faut faire appel à une transmodulation QPSK → QAM qui effectue d'abord un décodage QPSK puis un recodage en QAM. Ce qui correspond à un débit brut en QAM de 37,5 Mbit/s (après codage de Reed Solomon) pour une bande passante de 8 MHz. Le tableau suivant permet de comparer le débit brut et la largeur de bande occupée entre un système de distribution par satellite et une distribution par câble.

Modulation	QPSK (satellite)	QAM (câble)
Débit brut	56,3 Mbit/s	37,5 Mbit/s
Bande passante	36 MHz	8 MHz

Tableau 10-3 : débit brut et la largeur de bande occupée

Contraintes : Certes, l'utilisation de la modulation QAM à multiples niveaux permet de distribuer les signaux de TV numérique dans les réseaux de distribution collectifs, tout en respectant la limitation de largeur des canaux terrestres. Cependant le débit binaire utile (R_u) varie suivant le nombre de niveaux appliqué. Le tableau 10-4 montre la variation du débit binaire utile, du débit symbole et de la largeur de bande occupée en fonction des différentes modulations QAM16 à QAM64 utilisées pour la distribution par câble.

Type de modulation	Débit binaire utile en sortie du multiplexeur MPEG-2 [Mbit/s]	Débit symbole câble [MBaud]	Bande occupée [MHz]	Largeur du canal [MHz]
QAM16	21,7	5,88	6,77	7
QAM32	25,2	5,47	6,29	7
QAM64	33,4	6,04	6,95	7
QAM16	**25,2**	**6,84**	**7,86**	**8**
QAM32	**31,9**	**6,92**	**7,96**	**8**
QAM64	**38,1**	**6,89**	**7,92**	**8**

Tableau 10-4 : Variation du débit binaire utile, du débit symbole et de la largeur de bande occupée en fonction des différentes modulations QAM16 à QAM64.

Nous constatons que seule la modulation QAM64 qui possède un débit utile de 38,1 Mbit/s est capable de faire passer la totalité du débit brut (37,5 Mbit/s) dans un canal de 8 MHz. En revanche, pour les modulations QAM16 et QAM32, les débits de transmission utilisés dans une même largeur de bande seront plus faibles (25,2 Mbit/s et 31,9 Mbit/s). Ce qui signifie que le nombre de programmes TV numérique transmis sera moins important.

10.3.3.2 Système B : Distribution directe

Ce système applique le principe de réception directe par satellite. A la place des signaux analogiques, on reçoit un signal numérique au format QPSK. Ce signal en provenance du satellite conservera sa bande passante de 36 MHz. Seule la fréquence porteuse du signal QPSK est transposée dans une bande de fréquence plus faible. Cette nouvelle fréquence se situe soit dans le plan de fréquence VHF-UHF utilisé par le réseau collectif, soit dans la bande des fréquences intermédiaire satellite (BIS) réservée pour la diffusion des programmes de TV analogique par satellite (950 MHz - 2050 MHz). Ce système profite d'une structure déjà installée pour la réception par satellite de la TV analogique. Deux configurations peuvent être considérées :

- SMATV-FI.
- SMATV-S.

Dans ces deux configurations, les signaux arrivant au récepteur ne subissent aucun changement de codage ou de modulation.

10.3.3.2.1 SMATV-FI

Cette configuration permet de distribuer directement des signaux QPSK à l'utilisateur. A la sortie du convertisseur LNB (Low Noise Block), les signaux QPSK en fréquence intermédiaire (FI) sont distribués directement au réseau de distribution collectif terrestre dans la bande réservée pour la diffusion par satellite (950 MHz - 2050 MHz). Arrivé à la prise d'usager, le récepteur QPSK de l'utilisateur s'accordera sur la fréquence correspondante au programme sélectionné pour effectuer la démodulation et le décodage du signal QPSK.

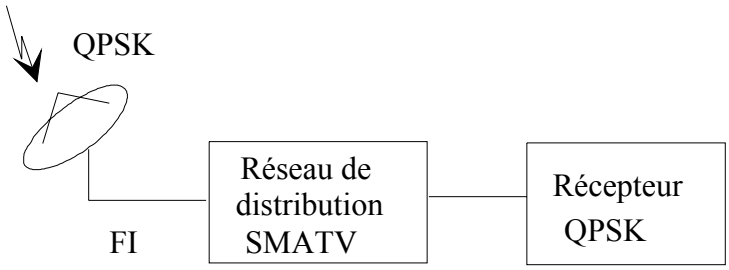


Figure 10-13 : Configuration du système SMATV-FI.

10.3.3.2.2 SMATV-S

Dans cette configuration, la fréquence porteuse FI du signal QPSK qui se trouve dans la bande de fréquence intermédiaire satellite B.I.S (entre 950 MHz et 2050 MHz) est d'abord transposée en bande S (entre 230 MHz et 470 MHz) avant d'être distribuée dans un réseau de distribution par câble. La figure suivante représente la configuration du système SMATV-S.

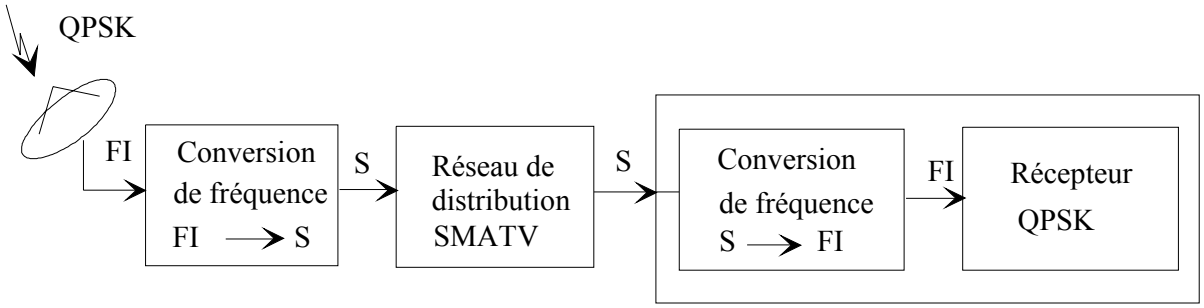


Figure 10-14 : Configuration du système SMATV-S.

La procédure de conversion inverse (bande S en bande B.I.S.) peut être intégrée dans les récepteurs QPSK.